Scientometrics in R

Bibliometrics, Network Science, and Research Evaluation

R. Heller

2026-11-14

Contents

Welcome! **35**

Open Source Repository 35

How To Use The Guide 36

Contributing 36

Citing this Guide 36

Impressum **37**

Suggested citation 37

Disclaimer 37

Contact 38

Acknowledgments **39**

Inspiration 39

Use of LLM tools 39

How to use this book **41**

0.1 Reading paths 41

0.2 Code chunks 41

0.3 Callouts 41

0.4 Companion data 42

0.5 Downloads 42

0.6 Citation 42

Notation **43**
Mathematical notation 43
Abbreviations 43
Code-style conventions 44

About the author **45**

Find Your Method **47**
0.7 Quick reference table 47

I Foundations **49**

1 What Is Scientometrics? **51**
1.1 Learning objectives 51
1.2 Setup 51
1.3 Conceptual background 52
1.3.1 A brief history 52
1.3.2 The modern era 53
1.3.3 Scope of this book 53
1.4 Worked example 54
1.4.1 Fetching a sample from OpenAlex 54
1.4.2 Basic descriptive statistics 56
1.4.3 Top journals 57
1.4.4 Publication trend over time 58
1.4.5 Top journals visualised 59
1.4.6 Citation distribution 59
1.5 Diagnostics and interpretation 60
1.6 Limitations and responsible use 61
1.7 Limitations and responsible use 61
1.8 Common pitfalls 62
1.9 Common pitfalls 62
1.10 Exercises 62

1.11 Solutions 63
1.12 Further reading 63
1.13 Session info 63

2 Theoretical Underpinnings 65
2.1 Learning objectives 65
2.2 Setup 65
2.3 Conceptual background 66
2.4 Worked example 66
2.4.1 Fetching the data 67
2.4.2 Lotka's Law 67
2.4.3 Bradford's Law 68
2.4.4 Zipf's Law 69
2.4.5 Price's growth law 70
2.5 Diagnostics and interpretation 71
2.6 Limitations and responsible use 71
2.7 Limitations and responsible use 71
2.8 Common pitfalls 72
2.9 Common pitfalls 72
2.10 Exercises 72
2.11 Solutions 73
2.12 Further reading 73
2.13 Session info 73

3 Ethics and Responsible Metrics 75
3.1 Learning objectives 75
3.2 Setup 75
3.3 Conceptual background 76
3.3.1 The Leiden Manifesto 76
3.3.2 The San Francisco Declaration (DORA) 77
3.3.3 The Metric Tide 77
3.3.4 Goodhart's Law in practice 78

3.3.5 Publish or perish 78
3.4 Worked example 79
3.4.1 Acquiring author data 79
3.4.2 Fetching works and computing self-citations . . . 79
3.4.3 Summarizing results 81
3.4.4 Visualization 82
3.4.5 A per-author comparison 82
3.5 Diagnostics and interpretation 83
3.6 Limitations and responsible use 84
3.7 Limitations and responsible use 84
3.8 Common pitfalls 85
3.9 Common pitfalls 85
3.10 Exercises 86
3.11 Solutions 86
3.12 Further reading 86
3.13 Session info 87

4 Data Sources Compared **89**
4.1 Learning objectives 89
4.2 Setup 89
4.3 Conceptual background 90
4.3.1 Twelve data sources at a glance 91
4.4 Worked example 92
4.4.1 Selecting DOIs 93
4.4.2 Fetching from OpenAlex 93
4.4.3 Fetching from Crossref 94
4.4.4 Comparing metadata completeness 95
4.4.5 Visualization: citation count comparison 96
4.4.6 Summary comparison table 97
4.5 Diagnostics and interpretation 99
4.6 Limitations and responsible use 100

4.7 Limitations and responsible use 100
4.8 Common pitfalls 101
4.9 Common pitfalls 101
4.10 Exercises 101
4.11 Solutions 102
4.12 Further reading 102
4.13 Session info 102

II Data Acquisition in R 105

5 APIs and Packages 107
5.1 Learning objectives 107
5.2 Setup 107
5.3 Conceptual background 108
5.4 Worked example 108
5.4.1 Querying works from OpenAlex 108
5.4.2 Inspecting the results 111
5.4.3 Querying authors 112
5.4.4 Deduplication by DOI 112
5.4.5 Caching for reproducible builds 113
5.4.6 Complementary sources: Crossref 113
5.4.7 Visualization 113
5.5 Diagnostics and interpretation 114
5.6 Limitations and responsible use 115
5.7 Limitations and responsible use 115
5.8 Common pitfalls 115
5.9 Common pitfalls 115
5.10 Exercises 116
5.11 Solutions 116
5.12 Further reading 116
5.13 Session info 117

6 Parsing Native Exports **119**

6.1 Learning objectives . . . 119
6.2 Setup . . . 119
6.3 Conceptual background . . . 120
6.4 Worked example . . . 120
6.4.1 Parsing a BibTeX string . . . 120
6.4.2 Parsing with RefManageR . . . 121
6.4.3 Parsing WoS plain text with bibliometrix . . . 122
6.4.4 Standardizing field names . . . 122
6.4.5 Visualization . . . 123
6.5 Diagnostics and interpretation . . . 123
6.6 Limitations and responsible use . . . 124
6.7 Limitations and responsible use . . . 124
6.8 Common pitfalls . . . 124
6.9 Common pitfalls . . . 124
6.10 Exercises . . . 124
6.11 Solutions . . . 125
6.12 Further reading . . . 125
6.13 Session info . . . 125

7 Building Reproducible Corpora **129**

7.1 Learning objectives . . . 129
7.2 Setup . . . 129
7.3 Conceptual background . . . 130
7.4 Worked example . . . 130
7.4.1 Fetching raw data . . . 130
7.4.2 DOI-based deduplication . . . 131
7.4.3 Fuzzy title deduplication . . . 131
7.4.4 Affiliation extraction and ROR mapping . . . 132
7.4.5 Saving as Parquet . . . 132
7.4.6 Visualization . . . 133

7.5 Diagnostics and interpretation 133
7.6 Limitations and responsible use 134
7.7 Limitations and responsible use 134
7.8 Common pitfalls 134
7.9 Common pitfalls 134
7.10 Exercises 135
7.11 Solutions 135
7.12 Further reading 135
7.13 Session info 135

8 Working with Full Text **139**
8.1 Learning objectives 139
8.2 Setup 139
8.3 Conceptual background 140
8.4 Worked example 140
8.4.1 Extracting text from a PDF 140
8.4.2 Basic text preprocessing 141
8.4.3 Word frequency visualization 142
8.4.4 OCR with tesseract 142
8.5 Diagnostics and interpretation 143
8.6 Limitations and responsible use 143
8.7 Limitations and responsible use 143
8.8 Common pitfalls 144
8.9 Common pitfalls 144
8.10 Exercises 144
8.11 Solutions 144
8.12 Further reading 145
8.13 Session info 145

III Core Bibliometric Analysis **149**

9 Descriptive Bibliometrics **151**

9.1 Learning objectives . . . 151
9.2 Setup . . . 151
9.3 Conceptual background . . . 152
9.4 Worked example . . . 153
9.4.1 Acquiring the data . . . 153
9.4.2 Converting to bibliometrix format . . . 153
9.4.3 Running biblioAnalysis . . . 153
9.4.4 Annual scientific production . . . 154
9.4.5 Most productive authors . . . 154
9.4.6 Most productive sources . . . 156
9.4.7 Three-field plot . . . 156
9.5 Diagnostics and interpretation . . . 157
9.6 Limitations and responsible use . . . 157
9.7 Limitations and responsible use . . . 157
9.8 Common pitfalls . . . 157
9.9 Common pitfalls . . . 157
9.10 Exercises . . . 158
9.11 Solutions . . . 158
9.12 Further reading . . . 158
9.13 Session info . . . 159

10 Productivity and Impact Indicators **163**

10.1 Learning objectives . . . 163
10.2 Setup . . . 163
10.3 Conceptual background . . . 164
10.4 Worked example . . . 164
10.4.1 Acquiring sample data . . . 164
10.4.2 Computing the h-index . . . 165
10.4.3 Computing the g-index . . . 165

10.4.4 Publication span is not career length 166
10.4.5 A normalization diagnostic: using the sample as its own reference 166
10.4.6 Within-sample upper-percentile membership . . . 167
10.4.7 Visualization 167
10.4.8 Summary table 168
10.5 Diagnostics and interpretation 170
10.6 Limitations and responsible use 170
10.7 Limitations and responsible use 170
10.8 Common pitfalls 171
10.9 Common pitfalls 171
10.10 Exercises 171
10.11 Solutions 172
10.12 Further reading 172
10.13 Session info 172

11 Journal Metrics **175**
11.1 Learning objectives 175
11.2 Setup 175
11.3 Conceptual background 176
11.4 Worked example 177
11.4.1 Selecting journals 177
11.4.2 Fetching a publication cohort 177
11.4.3 Cumulative citation summaries 178
11.4.4 Recorded reference counts 179
11.4.5 Visualization 180
11.5 Diagnostics and interpretation 180
11.6 Limitations and responsible use 181
11.7 Limitations and responsible use 181
11.8 Common pitfalls 182
11.9 Common pitfalls 182
11.10 Exercises 182

11.11 Solutions 183
11.12 Further reading 183
11.13 Session info 183

12 Author-Level Analysis and Disambiguation 187
12.1 Learning objectives 187
12.2 Setup 187
12.3 Conceptual background 188
12.4 Worked example 189
12.4.1 Retrieving an author entity 189
12.4.2 Fetching the author's works 189
12.4.3 Author-level statistics 190
12.4.4 Publication timeline 191
12.4.5 Top venues 192
12.4.6 Checking for name variants 192
12.5 Diagnostics and interpretation 193
12.6 Limitations and responsible use 193
12.7 Limitations and responsible use 193
12.8 Common pitfalls 194
12.9 Common pitfalls 194
12.10 Exercises 194
12.11 Solutions 195
12.12 Further reading 195
12.13 Session info 195

13 Institutional and Country Analysis 199
13.1 Learning objectives 199
13.2 Setup 199
13.3 Conceptual background 200
13.4 Worked example 201
13.4.1 Selecting institutions 201
13.4.2 Fetching institutional publications 201

13.4.3 Full counting vs. fractional counting 202
13.4.4 Descriptive citation comparison using a shared reference 203
13.4.5 Visualization 204
13.5 Diagnostics and interpretation 204
13.6 Limitations and responsible use 206
13.7 Limitations and responsible use 206
13.8 Common pitfalls 206
13.9 Common pitfalls 206
13.10 Exercises 207
13.11 Solutions 207
13.12 Further reading 207
13.13 Session info 208

14 Temporal Analysis and Citation Aging 211
14.1 Learning objectives 211
14.2 Setup 211
14.3 Conceptual background 212
14.4 Worked example 213
14.4.1 Acquiring the data 213
14.4.2 Citation aging curves 213
14.4.3 Estimating citation half-life 214
14.4.4 Price index 215
14.4.5 Cumulative citation trajectories 215
14.5 Diagnostics and interpretation 217
14.6 Limitations and responsible use 217
14.7 Limitations and responsible use 217
14.8 Common pitfalls 217
14.9 Common pitfalls 217
14.10 Exercises 218
14.11 Solutions 218
14.12 Further reading 218
14.13 Session info 219

IV Network & Mapping Methods **223**

15 Co-authorship Networks **225**

15.1 Learning objectives . . . 225
15.2 Setup . . . 225
15.3 Conceptual background . . . 226
15.4 Worked example . . . 227
15.4.1 Acquiring co-authorship data . . . 227
15.4.2 Extracting author pairs . . . 227
15.4.3 Building the network . . . 228
15.4.4 Network metrics . . . 228
15.4.5 Community detection . . . 229
15.4.6 Comparing community assignments . . . 230
15.4.7 Visualization . . . 230
15.5 Diagnostics and interpretation . . . 232
15.6 Limitations and responsible use . . . 233
15.7 Limitations and responsible use . . . 233
15.8 Common pitfalls . . . 234
15.9 Common pitfalls . . . 234
15.10 Exercises . . . 234
15.11 Solutions . . . 235
15.12 Further reading . . . 235
15.13 Session info . . . 235

16 Co-citation and Bibliographic Coupling **239**

16.1 Learning objectives . . . 239
16.2 Setup . . . 239
16.3 Conceptual background . . . 240
16.4 Worked example . . . 241
16.4.1 Fetching citing and cited works . . . 241
16.4.2 Building a co-citation network . . . 241
16.4.3 Building a bibliographic coupling network . . . 242

16.4.4 Visualisation 243
16.4.5 Comparing the two approaches 244
16.5 Diagnostics and interpretation 245
16.6 Limitations and responsible use 246
16.7 Limitations and responsible use 246
16.8 Common pitfalls 246
16.9 Common pitfalls 246
16.10 Exercises 247
16.11 Solutions 247
16.12 Further reading 247
16.13 Session info 247

17 Co-word and Keyword Co-occurrence 251
17.1 Learning objectives 251
17.2 Setup 251
17.3 Conceptual background 252
17.4 Worked example 253
17.4.1 Extracting keywords 253
17.4.2 Building the co-occurrence network 255
17.4.3 Community detection for topical clusters 255
17.4.4 Visualisation 256
17.5 Diagnostics and interpretation 257
17.6 Limitations and responsible use 258
17.7 Limitations and responsible use 258
17.8 Common pitfalls 258
17.9 Common pitfalls 258
17.10 Exercises 259
17.11 Solutions 259
17.12 Further reading 259
17.13 Session info 260

18 Community Detection and Backbone Extraction 263
18.1 Learning objectives . . . 263
18.2 Setup . . . 263
18.3 Conceptual background . . . 264
18.4 Worked example . . . 265
18.4.1 Building a dense network . . . 265
18.4.2 Backbone extraction: disparity filter . . . 265
18.4.3 Threshold filtering for comparison . . . 266
18.4.4 Community detection at multiple resolutions . . . 267
18.4.5 Comparing backbone vs. full network communities 268
18.4.6 Visualisation . . . 269
18.5 Diagnostics and interpretation . . . 269
18.6 Limitations and responsible use . . . 270
18.7 Limitations and responsible use . . . 270
18.8 Common pitfalls . . . 271
18.9 Common pitfalls . . . 271
18.10 Exercises . . . 271
18.11 Solutions . . . 271
18.12 Further reading . . . 272
18.13 Session info . . . 272

19 Science Mapping and Overlay Maps 275
19.1 Learning objectives . . . 275
19.2 Setup . . . 275
19.3 Conceptual background . . . 276
19.4 Worked example . . . 277
19.4.1 Building a journal co-citation base map . . . 277
19.4.2 Clustering journals into disciplines . . . 278
19.4.3 Visualising the base map . . . 278
19.4.4 Creating an overlay map . . . 279
19.5 Diagnostics and interpretation . . . 282

19.6 Limitations and responsible use 282
19.7 Limitations and responsible use 282
19.8 Common pitfalls 283
19.9 Common pitfalls 283
19.10 Exercises 283
19.11 Solutions 283
19.12 Further reading 283
19.13 Session info 284

20 Network Visualization and Interoperability 287
20.1 Learning objectives 287
20.2 Setup 287
20.3 Conceptual background 288
20.4 Worked example 289
20.4.1 Building a sample network 289
20.4.2 Comparing layout algorithms 290
20.4.3 Exporting to GraphML (Gephi) 290
20.4.4 Exporting to Pajek format 292
20.4.5 Exporting to VOSviewer format 292
20.4.6 Interactive visualisation with visNetwork 293
20.4.7 Publication-quality figure 293
20.5 Diagnostics and interpretation 294
20.6 Limitations and responsible use 295
20.7 Limitations and responsible use 295
20.8 Common pitfalls 295
20.9 Common pitfalls 295
20.10 Exercises 296
20.11 Solutions 296
20.12 Further reading 296
20.13 Session info 296

V Text & Topic Analysis **299**

21 Text Mining Bibliographic Corpora **301**

21.1 Learning objectives . . . 301
21.2 Setup . . . 301
21.3 Conceptual background . . . 302
21.4 Worked example . . . 303
21.4.1 Building a text corpus . . . 303
21.4.2 Tokenisation and preprocessing . . . 303
21.4.3 Document-feature matrix . . . 304
21.4.4 TF-IDF weighting . . . 304
21.4.5 Term frequency over time . . . 307
21.5 Diagnostics and interpretation . . . 307
21.6 Limitations and responsible use . . . 308
21.7 Limitations and responsible use . . . 308
21.8 Common pitfalls . . . 309
21.9 Common pitfalls . . . 309
21.10 Exercises . . . 309
21.11 Solutions . . . 309
21.12 Further reading . . . 310
21.13 Session info . . . 310

22 Topic Modeling **313**

22.1 Learning objectives . . . 313
22.2 Setup . . . 313
22.3 Conceptual background . . . 314
22.4 Worked example . . . 315
22.4.1 Preparing the corpus . . . 315
22.4.2 Fitting LDA . . . 316
22.4.3 Fitting STM with year covariate . . . 316
22.4.4 Topic prevalence over time . . . 317
22.4.5 Model diagnostics . . . 318

22.5 Diagnostics and interpretation 319
22.6 Limitations and responsible use 320
22.7 Limitations and responsible use 320
22.8 Common pitfalls 320
22.9 Common pitfalls 320
22.10 Exercises 320
22.11 Solutions 321
22.12 Further reading 321
22.13 Session info 321

23 Keyword Extraction **325**
23.1 Learning objectives 325
23.2 Setup 325
23.3 Conceptual background 326
23.4 Worked example 327
23.4.1 Fetching data with keywords 327
23.4.2 RAKE keyword extraction 327
23.4.3 TF-IDF based keyword extraction 328
23.4.4 Corpus-level keyword frequency 329
23.4.5 Comparing extraction methods 330
23.5 Diagnostics and interpretation 331
23.6 Limitations and responsible use 331
23.7 Limitations and responsible use 331
23.8 Common pitfalls 331
23.9 Common pitfalls 331
23.10 Exercises 332
23.11 Solutions 332
23.12 Further reading 332
23.13 Session info 333

24 Embeddings for Scientometrics **337**
24.1 Learning objectives . . . 337
24.2 Setup . . . 337
24.3 Conceptual background . . . 338
24.4 Worked example . . . 339
24.4.1 Preparing text data . . . 339
24.4.2 Training word2vec embeddings . . . 339
24.4.3 Computing document embeddings via mean pooling 340
24.4.4 Dimensionality reduction with UMAP . . . 341
24.4.5 Document similarity search . . . 341
24.4.6 Embedding clusters vs. citation impact . . . 343
24.5 Diagnostics and interpretation . . . 343
24.6 Limitations and responsible use . . . 345
24.7 Limitations and responsible use . . . 345
24.8 Common pitfalls . . . 345
24.9 Common pitfalls . . . 345
24.10 Exercises . . . 346
24.11 Solutions . . . 346
24.12 Further reading . . . 346
24.13 Session info . . . 346

25 Detecting Emerging Topics **351**
25.1 Learning objectives . . . 351
25.2 Setup . . . 351
25.3 Conceptual background . . . 352
25.4 Worked example . . . 353
25.4.1 Preparing temporal keyword data . . . 353
25.4.2 Growth rate detection . . . 354
25.4.3 Simple burst detection . . . 355
25.4.4 Visualising keyword trajectories . . . 356
25.4.5 Rolling-window topic tracking . . . 357

25.5 Diagnostics and interpretation 357
25.6 Limitations and responsible use 358
25.7 Limitations and responsible use 358
25.8 Common pitfalls 359
25.9 Common pitfalls 359
25.10 Exercises 359
25.11 Solutions 359
25.12 Further reading 360
25.13 Session info 360

VI Advanced Topics 363

26 Altmetrics 365

26.1 Learning objectives 365
26.2 Setup 365
26.3 Conceptual background 366
26.4 Worked example 366
26.4.1 Fetching data with OA indicators 366
26.4.2 Open access as an altmetric dimension 367
26.4.3 Citation count distribution 368
26.4.4 Proxy altmetric: reference list length 368
26.5 Diagnostics and interpretation 370
26.6 Limitations and responsible use 370
26.7 Limitations and responsible use 370
26.8 Common pitfalls 371
26.9 Common pitfalls 371
26.10 Exercises 371
26.11 Solutions 371
26.12 Further reading 372
26.13 Session info 372

27 Open Science Indicators **375**
27.1 Learning objectives . . . 375
27.2 Setup . . . 375
27.3 Conceptual background . . . 376
27.4 Worked example . . . 376
27.4.1 OA rates by year . . . 376
27.4.2 OA citation advantage . . . 377
27.4.3 Document type breakdown . . . 378
27.4.4 Summary table . . . 378
27.5 Diagnostics and interpretation . . . 379
27.6 Limitations and responsible use . . . 379
27.7 Limitations and responsible use . . . 379
27.8 Common pitfalls . . . 380
27.9 Common pitfalls . . . 380
27.10 Exercises . . . 380
27.11 Solutions . . . 381
27.12 Further reading . . . 381
27.13 Session info . . . 381

28 Gender and Equity Analysis **385**
28.1 Learning objectives . . . 385
28.2 Setup . . . 385
28.3 Conceptual background . . . 386
28.4 Worked example . . . 386
28.4.1 Explicitly synthetic records . . . 386
28.4.2 Lookup coverage, not gender classification . . . 387
28.4.3 Keeping the full denominator . . . 387
28.4.4 Recorded authorship positions . . . 388
28.5 Diagnostics and interpretation . . . 390
28.6 Limitations and responsible use . . . 390
28.7 Limitations and responsible use . . . 390

28.8 Common pitfalls . . . 390
28.9 Common pitfalls . . . 390
28.10 Exercises . . . 391
28.11 Solutions . . . 391
28.12 Further reading . . . 391
28.13 Session info . . . 391

29 Retraction Analysis 395
29.1 Learning objectives . . . 395
29.2 Setup . . . 395
29.3 Conceptual background . . . 396
29.4 Worked example . . . 396
29.4.1 Finding retracted works . . . 396
29.4.2 Citation counts of retracted papers . . . 397
29.4.3 Retracted papers by source . . . 398
29.4.4 Citation persistence over time . . . 398
29.5 Diagnostics and interpretation . . . 399
29.6 Limitations and responsible use . . . 400
29.7 Limitations and responsible use . . . 400
29.8 Common pitfalls . . . 400
29.9 Common pitfalls . . . 400
29.10 Exercises . . . 401
29.11 Solutions . . . 401
29.12 Further reading . . . 401
29.13 Session info . . . 401

30 Funding Analysis and Grant-to-Publication Linkage 405
30.1 Learning objectives . . . 405
30.2 Setup . . . 405
30.3 Conceptual background . . . 406
30.4 Worked example . . . 406
30.4.1 Extracting funder data . . . 406

30.4.2 Top funders by publication count 408
30.4.3 Citation impact by funder 409
30.4.4 Comparison by observed funder metadata 410
30.5 Diagnostics and interpretation 410
30.6 Limitations and responsible use 411
30.7 Limitations and responsible use 411
30.8 Common pitfalls 411
30.9 Common pitfalls 411
30.10 Exercises 412
30.11 Solutions 412
30.12 Further reading 412
30.13 Session info 412

31 Patent–Paper Linkages 417
31.1 Learning objectives 417
31.2 Setup 417
31.3 Conceptual background 418
31.4 Worked example 418
31.4.1 Preparing a scholarly-citation sample 419
31.4.2 Scholarly citation distribution 419
31.4.3 Journals represented among highly cited sampled papers 419
31.4.4 Descriptive characteristics by scholarly-citation group 420
31.5 Diagnostics and interpretation 421
31.6 Limitations and responsible use 422
31.7 Limitations and responsible use 422
31.8 Common pitfalls 422
31.9 Common pitfalls 422
31.10 Exercises 423
31.11 Solutions 423
31.12 Further reading 423
31.13 Session info 424

32 Causal Inference in Scientometrics **427**
32.1 Learning objectives . . . 427
32.2 Setup . . . 427
32.3 Conceptual background . . . 428
32.4 Worked example . . . 429
32.4.1 A descriptive OA-by-publication-cohort comparison 429
32.4.2 Descriptive comparison . . . 430
32.4.3 Descriptive interaction regression . . . 430
32.4.4 Within-year sampling comparison . . . 432
32.4.5 Publication-cohort profile . . . 433
32.5 Diagnostics and interpretation . . . 433
32.6 Limitations and responsible use . . . 434
32.7 Limitations and responsible use . . . 434
32.8 Common pitfalls . . . 435
32.9 Common pitfalls . . . 435
32.10 Exercises . . . 435
32.11 Solutions . . . 436
32.12 Further reading . . . 436
32.13 Session info . . . 436

33 Reproducibility, RRIDs, and Badges **439**
33.1 Learning objectives . . . 439
33.2 Setup . . . 439
33.3 Conceptual background . . . 440
33.4 Worked example . . . 441
33.4.1 Analysing access status and metadata coverage . . 441
33.4.2 Observed article access status . . . 441
33.4.3 Abstract availability as metadata completeness . . 442
33.4.4 Access status and cumulative citations . . . 442
33.4.5 Summary statistics . . . 443
33.5 Diagnostics and interpretation . . . 444

33.6 Limitations and responsible use 444
33.7 Limitations and responsible use 444
33.8 Common pitfalls 445
33.9 Common pitfalls 445
33.10 Exercises 446
33.11 Solutions 446
33.12 Further reading 446
33.13 Session info 446

VII Production & Reporting 451

34 Reproducible Pipelines 453
34.1 Learning objectives 453
34.2 Setup 453
34.3 Conceptual background 454
34.4 Worked example 454
34.4.1 Defining a targets pipeline 454
34.4.2 Running and inspecting the pipeline 455
34.4.3 Project structure 456
34.4.4 Using renv 456
34.4.5 Dockerfile for bibliometric analysis 456
34.5 Diagnostics and interpretation 457
34.6 Limitations and responsible use 457
34.7 Limitations and responsible use 457
34.8 Common pitfalls 457
34.9 Common pitfalls 457
34.10 Exercises 458
34.11 Solutions 458
34.12 Further reading 458
34.13 Session info 458

35 Interactive Dashboards **463**
35.1 Learning objectives . . . 463
35.2 Setup . . . 463
35.3 Conceptual background . . . 464
35.4 Worked example . . . 464
35.4.1 Preparing dashboard data . . . 464
35.4.2 Interactive trend chart . . . 465
35.4.3 Interactive citation distribution . . . 465
35.4.4 Searchable data table . . . 465
35.4.5 OA status breakdown (interactive) . . . 466
35.5 Diagnostics and interpretation . . . 466
35.6 Limitations and responsible use . . . 466
35.7 Limitations and responsible use . . . 466
35.8 Common pitfalls . . . 467
35.9 Common pitfalls . . . 467
35.10 Exercises . . . 467
35.11 Solutions . . . 467
35.12 Further reading . . . 468
35.13 Session info . . . 468

36 Parameterized Quarto Reports **471**
36.1 Learning objectives . . . 471
36.2 Setup . . . 471
36.3 Conceptual background . . . 472
36.4 Worked example . . . 472
36.4.1 Defining parameters . . . 472
36.4.2 Using parameters in code . . . 473
36.4.3 Automated summary statistics . . . 473
36.4.4 Parameterized figure . . . 474
36.4.5 Batch rendering . . . 474
36.5 Diagnostics and interpretation . . . 475

36.6 Limitations and responsible use 476
36.7 Limitations and responsible use 476
36.8 Common pitfalls 476
36.9 Common pitfalls 476
36.10 Exercises 476
36.11 Solutions 477
36.12 Further reading 477
36.13 Session info 477

37 Building a Bibliometric Shiny App 481
37.1 Learning objectives 481
37.2 Setup 481
37.3 Conceptual background 482
37.4 Worked example 482
37.4.1 App structure 482
37.4.2 UI definition 483
37.4.3 Server logic 483
37.4.4 Running the app 485
37.4.5 biblioshiny as an alternative 485
37.5 Diagnostics and interpretation 485
37.6 Limitations and responsible use 486
37.7 Limitations and responsible use 486
37.8 Common pitfalls 486
37.9 Common pitfalls 486
37.10 Exercises 486
37.11 Solutions 487
37.12 Further reading 487
37.13 Session info 487

VIII Case Studies **491**

38 Case Study 1: CRISPR Field Review (2010–2024) **493**
38.1 Objective . . . 493
38.2 Setup . . . 493
38.3 Data acquisition . . . 494
38.4 Publication growth . . . 494
38.5 Citation landscape . . . 494
38.6 Co-authorship network . . . 496
38.7 Topic evolution . . . 498
38.8 Key findings . . . 499
38.9 Lessons learned . . . 500

39 Case Study 2: Institutional Benchmarking **501**
39.1 Objective . . . 501
39.2 Setup . . . 501
39.3 Data acquisition . . . 502
39.4 Publication volume . . . 502
39.5 Citation summaries . . . 503
39.6 Full vs. fractional counting . . . 504
39.7 OA rates . . . 505
39.8 Reading the results . . . 505
39.9 Lessons learned . . . 506

40 Case Study 3: Journal Portfolio Analysis **507**
40.1 Objective . . . 507
40.2 Setup . . . 507
40.3 Data acquisition . . . 508
40.4 Citation distributions in the samples . . . 509
40.5 Publication-cohort comparison . . . 510
40.6 Topical coverage . . . 510
40.7 Reading the results . . . 512
40.8 Lessons learned . . . 512

41 Case Study 4: Gender Gap in Scientific Publishing **515**

41.1 Objective . . . 515
41.2 Setup . . . 515
41.3 Synthetic teaching records . . . 516
41.4 Arbitrary lookup and complete denominators . . . 516
41.5 Lookup categories by fictional journal . . . 517
41.6 Recorded authorship position . . . 518
41.7 Coverage by recorded year . . . 519
41.8 What the fixture establishes . . . 520
41.9 Requirements for an actual study . . . 520

42 Case Study 5: Detecting Emerging Topics **523**

42.1 Objective . . . 523
42.2 Setup . . . 523
42.3 Data acquisition . . . 524
42.4 Keyword burst detection . . . 524
42.5 Keyword trajectories . . . 525
42.6 Topic modelling . . . 527
42.7 Topic prevalence by year . . . 528
42.8 Key findings . . . 529
42.9 Lessons learned . . . 529

A R Setup, renv, and Docker **531**

A.1 R installation and current first-use path . . . 531
A.2 Package management with renv . . . 532
A.3 Core packages . . . 532
A.4 Docker environment . . . 533
A.5 Render the book . . . 533

B SQL Primer for the OpenAlex Snapshot **535**
B.1 When to use the snapshot . . . 535
B.2 Loading the snapshot . . . 535
B.3 Common queries . . . 536
B.3.1 Publication counts by year . . . 536
B.3.2 Top institutions by output . . . 536
B.3.3 Co-authorship pairs . . . 536
B.4 Connecting from R . . . 537
B.5 Performance tips . . . 537

C Indicator Crosswalk Across Databases **539**
C.1 Publication-level indicators . . . 539
C.2 Author-level indicators . . . 540
C.3 Journal-level indicators . . . 540
C.4 Institution-level indicators . . . 540
C.5 Network indicators . . . 541
C.6 Notes . . . 541

D Cheatsheets and Quick Reference **543**
D.1 openalexR quick reference . . . 543
D.2 quanteda pipeline . . . 544
D.3 igraph / tidygraph / ggraph . . . 545
D.4 Common data wrangling patterns . . . 545
D.5 Helper functions . . . 546
D.6 Useful keyboard shortcuts . . . 547

E Exercise Solutions **549**
Chapter 1 — What Is Scientometrics? . . . 549
Chapter 2 — Theoretical Underpinnings . . . 551
Chapter 3 — Ethics and Responsible Metrics . . . 553
Chapter 4 — Data Sources Compared . . . 554
Chapter 5 — APIs and Packages . . . 557

Chapter 6 — Parsing Native Exports 559
Chapter 7 — Building Reproducible Corpora 560
Chapter 8 — Working with Full Text 561
Chapter 9 — Descriptive Bibliometrics 563
Chapter 10 — Productivity and Impact Indicators 564
Chapter 11 — Journal Metrics 566
Chapter 12 — Author-Level Analysis 568
Chapter 13 — Institutional and Country Analysis 570
Chapter 14 — Temporal Analysis and Citation Aging 572
Chapter 15 — Co-authorship Networks 575
Chapter 16 — Co-citation and Bibliographic Coupling 578
Chapter 17 — Co-word and Keyword Co-occurrence 580
Chapter 18 — Community Detection and Backbone Extraction . 583
Chapter 19 — Science Mapping and Overlay Maps 585
Chapter 20 — Network Visualization and Interoperability . . . 587
Chapter 21 — Text Mining Bibliographic Corpora 589
Chapter 22 — Topic Modeling 591
Chapter 23 — Keyword Extraction 593
Chapter 24 — Embeddings for Scientometrics 595
Chapter 25 — Detecting Emerging Topics 597
Chapter 26 — Altmetrics 600
Chapter 27 — Open Science Indicators 602
Chapter 28 — Gender and Equity Analysis 603
Chapter 29 — Retraction Analysis 604
Chapter 30 — Funding Analysis 606
Chapter 31 — Patent-Paper Linkages 608
Chapter 32 — Causal Inference in Scientometrics 610
Chapter 33 — Reproducibility, RRIDs, and Badges 611
Chapter 34 — Reproducible Pipelines 613
Chapter 35 — Interactive Dashboards 615
Chapter 36 — Parameterized Quarto Reports 616
Chapter 37 — Building a Bibliometric Shiny App 619

Glossary **623**

Colophon **627**

Citing this Guide **629**

References **631**

Contents

Welcome!

R. Heller · Date: 2026-05-11 · Last changed: 2026-11-10

Welcome to **“Scientometrics in R: Bibliometrics, Network Science, and Research Evaluation”**.

This is a working guide to scientometrics in R for researchers, librarians, research administrators, and data scientists who want to study scholarly communication quantitatively. It walks the full pipeline — from pulling records out of OpenAlex, Crossref, and PubMed, through computing the standard indicators (h-index, MNCS, journal metrics), to building co-authorship and co-citation networks, applying text and topic models to scientific corpora, and packaging the results into reproducible reports and dashboards.

The book assumes a working knowledge of R at the tidyverse level and basic statistics. No prior scientometric experience is needed — the early chapters lay the conceptual and ethical foundations, and each later chapter starts with a concrete research question and ends with a callout on responsible use grounded in the Leiden Manifesto and DORA.

All examples run against free, open data; no Web of Science or Scopus subscription is required. Small synthetic sample datasets are stored under `data/` in the source repository; everything else is fetched live from public APIs. The book does not cover proprietary platforms (InCites, SciVal), manual screening workflows for systematic reviews, or qualitative sociology of science.

Open Source Repository

This book has been built using **{rmarkdown}** and **{bookdown}**. Formulas are rendered using MathJax. All source files are available on **GitHub** at https://github.com/CTTIR/scientometrics-in-r. You are free to fork, share, and reuse contents under the project's CC BY 4.0 license.

How To Use The Guide

- **Linear or lookup.** Read front to back, or use the Find Your Method navigator to jump to the chapter that answers your question.
- **Code.** Reusable helper functions live in `R/` in the source repository, and chapters load them with `source()`; every chapter shows the full chunk it depends on.
- **Downloads.** The side panel lists whole-book and chapter files that are available in the current build.
- **Corrections.** Please file an issue at https://github.com/CTTIR/scientometrics-in-r/issues.

Contributing

Pull requests, corrections, and additions are welcome. See `CONTRIBUTING.md` for the workflow.

Citing this Guide

The suggested citation is:

> **Heller, R.** (2026). Scientometrics in R: Bibliometrics, Network Science, and Research Evaluation (Version 0.1.0). Self-published via GitHub Pages. https://cttir.github.io/scientometrics-in-r/.

Download the reference as BibTeX or .ris.

Impressum

R. Heller · Date: 2026-05-11 · Last changed: 2026-11-09

Scientometrics in R: Bibliometrics, Network Science, and Research Evaluation

© 2026 Raban Heller. ORCID: 0000-0001-8006-9742.

Content released under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Source code released under the MIT License.

Published online: https://cttir.github.io/scientometrics-in-r/ Source repository: https://github.com/CTTIR/scientometrics-in-r

Version: 0.1.0 · Build date: 2026-11-22

Suggested citation

> Heller, R. (2026). Scientometrics in R: Bibliometrics, Network Science, and Research Evaluation. https://cttir.github.io/scientometrics-in-r/

Disclaimer

This book is provided “as is” without warranty of any kind. The author assumes no responsibility for errors or omissions.

Contact

For corrections, suggestions, or issues: https://github.com/CTTIR/scientometrics-in-r/issues

Acknowledgments

R. Heller · Date: 2026-05-11 · Last changed: 2026-06-10

This book stands on a wide stack of free software. Particular thanks to the bookdown authors and the broader R Core and tidyverse communities; to the maintainers of `openalexR`, `bibliometrix`, `igraph`, `tidygraph`, `ggraph`, `quanteda`, `stm`, and `tidytext` for the packages this book relies on most; and to the OpenAlex project for making large-scale, open bibliographic data freely available.

Inspiration

The structural and pedagogical model for this book draws heavily on Mathias Harrer, Pim Cuijpers, Toshi A. Furukawa, and David D. Ebert's Doing Meta-Analysis with R: A Hands-On Guide — both the book and its open-source repository at https://github.com/MathiasHarrer/Doing-Meta-Analysis-in-R. Their thorough, transparent, and beautifully laid-out bookdown setup — sidebar grouping, semantic callout boxes, the citation block, the dark/light toggle, MathJax-rendered formulas, downloadable BibTeX/RIS citations — set a clear standard for what a reproducible R-based research book can look like. Where the present book mirrors their conventions, it does so deliberately and gratefully. Where it diverges (colors, typography, individual icons), the divergence is cosmetic; the underlying craft is theirs to credit.

Use of LLM tools

Portions of this book were prepared with assistance from large language model tooling for narrowly defined, non-authorial tasks: copyediting, prose smoothing, Markdown/LaTeX formatting, scaffolding of boilerplate files (CI configs, build scripts), code refactoring. The tools used were Chat AI, the LLM service of KISSKI (GWDG), and a self-hosted **Mistral Small (24B,**

Apache-2.0) run locally via Ollama and the `ollamar` R package — local inference only, with no data sent to third parties for the self-hosted model.

All scientific claims, methodological choices, analyses, interpretations, and conclusions are the author's own. No LLM-generated text was incorporated without review and revision, and every reference was verified against its DOI, arXiv ID, or ISBN.

How to use this book

R. Heller · Date: 2026-05-11 · Last changed: 2026-11-10

0.1 Reading paths

The book is structured so that you can read it linearly — front to back — or jump straight to the chapter that answers your current question. If you do not yet know which method fits your problem, start with the Find Your Method navigator. It maps typical research questions ("evaluate a journal portfolio", "detect emerging topics", "study collaboration patterns") to the chapters and case studies that demonstrate the relevant techniques.

0.2 Code chunks

Every analytical claim is paired with a runnable R chunk. Chunks are labelled (`label`), and figures use `fig.cap` / `fig.alt` so the book remains accessible. Chunks that need network access, paid credentials, or interactive widgets are marked `eval = FALSE` and document what is required to run them locally.

0.3 Callouts

Three callout styles appear throughout:

A blue callout flags a clarification or aside.

An orange callout flags a limitation or responsible-use note — read these.

A red callout flags a common pitfall that has tripped up working scientometricians.

0.4 Companion data

Small synthetic sample datasets are stored as `.rda` files under `data/` in the source repository; load one with `load("data/openalex_sample.rda")`. The helper functions that the chapters use live in `R/`, and each chapter loads the ones it needs with `source()`.

0.5 Downloads

The sidebar's Download menu gives you the whole book as a PDF or EPUB. Each chapter page also exposes a Download this chapter (PDF) button at the top right.

0.6 Citation

If the book is useful to your research, please cite it — the suggested form lives in the Impressum, and a BibTeX block is in the README.

Notation

R. Heller · Date: 2026-05-11 · Last changed: 2026-05-11

The following symbols, abbreviations, and conventions are used throughout the book.

Mathematical notation

Symbol	Meaning
c_i	Citations received by paper i
N	Number of papers in a corpus
h	Hirsch index (h-index)
g	g-index
MNCS	Mean normalised citation score
$\bar{x}$	Sample mean
$\tilde{x}$	Sample median
G	Graph (V, E) — vertices and edges
k_v	Degree of vertex v
$B(v)$	Betweenness centrality of vertex v

Abbreviations

Acronym	Expansion
API	Application Programming Interface
DOI	Digital Object Identifier
FWCI	Field-Weighted Citation Impact
JIF	Journal Impact Factor
MNCS	Mean Normalised Citation Score

Acronym	Expansion
NMI	Normalised Mutual Information
OA	Open Access
ORCID	Open Researcher and Contributor ID
ROR	Research Organization Registry
RRID	Research Resource Identifier
SCI	Science Citation Index
SDG	Sustainable Development Goal
WoS	Web of Science

Code-style conventions

- Object names describe what they are (`effect_sizes`), not how they were produced (`df2`).
- Pipes use the base `|>`. Tidyverse verbs are preferred over base R where readability favours them.
- Network objects use `igraph` representations unless explicitly stated.
- Random seeds are set per chapter; the global default is 42.

About the author

R. Heller · Date: 2026-05-11 · Last changed: 2026-11-09

Raban Heller is a biomedical scientist working at the intersection of biostatistics, clinical research, and data science. He maintains the Strategy in R and Methods in R books in the same series.

- ORCID: https://orcid.org/0000-0001-8006-9742
- GitHub: https://github.com/r-heller
- Website: https://r-heller.github.io/website/

For corrections or suggestions, please open an issue at https://github.com/CTTIR/scientometrics-in-r/issues.

Find Your Method

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

Use this decision tree to navigate to the chapter or case study most relevant to your research question. On narrow screens, scroll the diagram horizontally. The quick reference table below provides the same routes in a compact format.

The decision-tree diagram is available in the web edition. Use the quick reference table below to select the relevant method and chapter in this edition.

0.7 Quick reference table

Research question	Key chapters	Case study
Map a research field	16, 19, 22	CS 1: CRISPR
Evaluate an author	10, 12	—
Evaluate a journal	11	CS 3: Journal Portfolio
Benchmark an institution	13	CS 2: Institutional
Detect emerging topics	14, 24, 25	CS 5: Emerging Topics
Analyze collaboration	15, 18	—
Study gender gaps	28	CS 4: Gender Gap
Track open access	27	—
Build a dashboard	35, 37	—
Create a reproducible pipeline	34, 36	—

Part I

Foundations

Chapter 1

What Is Scientometrics?

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

1.1 Learning objectives

After completing this chapter, you will be able to:

- Define scientometrics, bibliometrics, and informetrics and explain how the three fields relate
- Trace the historical development of scientometrics from Garfield's citation indexes through Price's quantitative studies to the present day
- Identify the types of research questions that scientometric methods can address
- Retrieve a small sample of scholarly works from OpenAlex and produce basic descriptive statistics in R
- Create simple visualisations of publication trends and journal distributions

1.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

1.3 Conceptual background

Scientometrics is the quantitative study of science itself — its growth, structure, productivity, and impact. Where scientists in most disciplines study the natural or social world, scientometricians turn the lens inward: they study the system that produces scientific knowledge.

The term sits within a family of overlapping labels. **Bibliometrics**, the oldest sibling, focuses on the statistical analysis of publications — books, journal articles, and their metadata such as authors, citations, and keywords. The word was coined by Alan Pritchard in 1969 to replace the earlier "statistical bibliography". **Informetrics** is the broadest umbrella, encompassing any quantitative study of information transfer, including web analytics, patent analysis, and library usage statistics. In practice, the three terms are often used interchangeably, but it helps to remember that bibliometrics emphasises the document, scientometrics emphasises the science, and informetrics emphasises the information.

1.3.1 A brief history

The intellectual roots of the field trace back to at least 1955, when Eugene Garfield (1955) proposed the idea of a citation index for science. Garfield argued that citations create an "association of ideas" — a traceable web of intellectual influence — and that indexing these links would transform how scholars discover literature. His vision was realised in the Science Citation Index (SCI), launched in 1964 by the Institute for Scientific Information (ISI). For the first time, researchers could systematically trace who cited whom across the entire scientific literature.

A few years later, Derek de Solla Price gave the field its quantitative backbone. In Little Science, Big Science (de Solla Price, 1963), Price marshalled data to show that science was growing exponentially — in the number of journals, articles, and scientists — and that this growth followed remarkably regular statistical patterns. He estimated that the volume of scientific literature doubles approximately every 15 years, a regularity so striking that it came to be known as "Price's Law." Price's work established that science could be studied with the same empirical rigour that it applies to other phenomena.

These foundations were reinforced by earlier statistical regularities. Alfred Lotka (1926) had already shown that the distribution of scientific productivity follows a power law: a small number of authors produce the majority of papers. Samuel Bradford (1934) demonstrated a similar pattern for journal coverage: a core set of journals in any field accounts for a disproportionate share of the relevant literature. These laws, which we explore in detail in the next chapter, remain cornerstones of scientometric theory.

From these beginnings, scientometrics expanded rapidly through the latter half of the twentieth century. The h-index (Hirsch, 2005), journal impact factors, co-citation networks, and science maps became standard tools in research evaluation and science policy. Landmark declarations such as the Leiden Manifesto (Hicks et al., 2015) and DORA (American Society for Cell Biology, 2012) later established principles for responsible use of these tools, warning against mechanical reliance on any single indicator.

1.3.2 The modern era

Today the field is undergoing another transformation. The launch of **OpenAlex** (Priem et al., 2022) — a fully open index of over 250 million scholarly works, built from Crossref, PubMed, institutional repositories, and other open sources — has democratised access to the raw data that powers scientometric research. Researchers no longer need expensive subscriptions to Web of Science or Scopus to conduct large-scale analyses. This book takes advantage of that openness: every example uses data that any reader can reproduce for free.

1.3.3 Scope of this book

This book teaches you to do scientometrics in R. It covers:

- **Data acquisition** — querying OpenAlex, Crossref, and PubMed programmatically.
- **Core bibliometrics** — computing productivity, impact, and citation indicators.
- **Network analysis** — building and analysing co-authorship, co-citation, and keyword co-occurrence networks.
- **Text and topic analysis** — extracting themes and trends from titles, abstracts, and full texts.
- **Advanced topics** — altmetrics, open science, gender equity, retraction analysis, and more.
- **Production** — building reproducible pipelines, dashboards, and reports.

Each chapter follows the same pattern: conceptual background, a fully reproducible worked example, diagnostics, limitations, and exercises.

1.4 Worked example

To make things concrete from the very first chapter, we fetch a small sample of recent works from OpenAlex and compute some basic descriptive statistics. Think of this as a first taste of the workflows that the rest of the book will develop in depth.

1.4.1 Fetching a sample from OpenAlex

We search for works whose title or abstract mentions “scientometrics”, published between 2014 and 2023, and draw a random sample of 500 records.

```
works <- fetch_openalex(
  entity = "works",
  search = "scientometrics",
  from_publication_date = "2014-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)

cat(glue("Fetched {nrow(works)} works\n"))
```

```
#> Fetched 500 works
```

```
glimpse(works)
```

```
#> Rows: 500
#> Columns: 45
#> $ id                              <chr>
 ↪  "https://openalex.org/W4382464140", "htt~
#> $ title                           <chr> "Desvendando a produção
 ↪  científica em to~
#> $ display_name                    <chr> "Desvendando a produção
 ↪  científica em to~
#> $ authorships                     <list> [<tbl_df[1 x 8]>],
 ↪  [<tbl_df[4 x 8]>], [~
#> $ abstract                       <chr> "O instrumento econômico
 ↪  \\\"Pagamento p~
#> $ doi                             <chr>
 ↪  "https://doi.org/10.11606/t.18.2023.tde-~
```

```
#> $ publication_date             <date> 2023-04-28, 2016-08-23,
 ↪  2020-12-16, 202~
#> $ publication_year              <int> 2023, 2016, 2020, 2020,
 ↪  2022, 2019, 2015~
#> $ relevance_score               <dbl> 1, 1, 1, 1, 1, 1, 1, 1,
 ↪  1, 1, 1, 1, 1, 1~
#> $ fwci                         <dbl> NA, 2.667, 0.000, 1.546,
 ↪  NA, 6.047, NA, ~
#> $ cited_by_count                <int> 1, 12, 1, 3, 7, 14, 0,
 ↪  1, 0, 14, 0, 62, ~
#> $ counts_by_year                <list> [<data.frame[1 x 2]>],
 ↪  [<data.frame[5 x~
#> $ ids                           <list>
 ↪  <"https://openalex.org/W4382464140", "h~
#> $ type                          <chr> "dissertation",
 ↪  "article", "article", "b~
#> $ is_oa                         <lgl> FALSE, TRUE, TRUE,
 ↪  FALSE, TRUE, TRUE, TR~
#> $ is_oa_anywhere               <lgl> TRUE, TRUE, TRUE, FALSE,
 ↪  TRUE, TRUE, TRU~
#> $ oa_status                     <chr> "green", "hybrid",
 ↪  "diamond", "closed", ~
#> $ oa_url                        <chr>
 ↪  "http://www.teses.usp.br/teses/disponive~
#> $ any_repository_has_fulltext   <lgl> TRUE, FALSE, TRUE,
 ↪  FALSE, FALSE, TRUE, T~
#> $ source_display_name           <chr> "Digital Library of
 ↪  Theses and Dissertat~
#> $ source_id                     <chr>
 ↪  "https://openalex.org/S4377196381", "htt~
#> $ issn_l                        <chr> NA, "0002-9343",
 ↪  "1562-4730", NA, "2542-~
#> $ host_organization             <chr>
 ↪  "https://openalex.org/I17974374", "https~
#> $ host_organization_name        <chr> "Universidade de São
 ↪  Paulo", "Elsevier B~
#> $ landing_page_url              <chr>
 ↪  "http://dx.doi.org/10.11606/t.18.2023.td~
#> $ pdf_url                       <chr> NA,
 ↪  "https://www.amjmed.com/article/S000~
#> $ license                      <chr> NA, "other-oa", "cc-by",
 ↪  NA, "other-oa",~
#> $ version                       <chr> "publishedVersion",
 ↪  "publishedVersion", ~
#> $ referenced_works              <list>
 ↪  <"https://openalex.org/W635460351", "ht~
#> $ referenced_works_count        <int> 150, 30, 6, 1, 8, 12,
 ↪  53, 447, 1, 91, 28~
#> $ related_works                 <list>
 ↪  <"https://openalex.org/W7216826511", "h~
```

```
#> $ concepts                          <list> [<data.frame[3 x 5]>],
 ↪  [<data.frame[11 ~
#> $ topics                            <list> [<tbl_df[12 x 5]>],
 ↪  [<tbl_df[12 x 5]>],~
#> $ keywords                         <list> [<data.frame[12 x 3]>],
 ↪  [<data.frame[6 ~
#> $ is_paratext                       <lgl> FALSE, FALSE, FALSE,
 ↪  FALSE, FALSE, FALSE~
#> $ is_retracted                      <lgl> FALSE, FALSE, FALSE,
 ↪  FALSE, FALSE, FALSE~
#> $ language                          <chr> "pt", "en", "pt", "en",
 ↪  "en", "es", "pt"~
#> $ sustainable_development_goals <list> NA, [<data.frame[1 x
 ↪  3]>], NA, NA, NA, ~
#> $ awards                            <list> NA,
 ↪  <"https://openalex.org/G1144012484"~
#> $ funders                           <list> NA, [<data.frame[10 x
 ↪  3]>], NA, NA, NA,~
#> $ apc                               <list> NA, [<data.frame[2 x
 ↪  5]>], NA, NA, NA, ~
#> $ first_page                        <chr> NA, "54", "38", "1",
 ↪  "70", "137", NA, NA~
#> $ last_page                         <chr> NA, "60.e5", "49", "7",
 ↪  "80", "155", NA,~
#> $ volume                            <chr> NA, "130", NA, NA, "7",
 ↪  "13", NA, NA, NA~
#> $ issue                             <chr> NA, "1", "77", NA, "1",
 ↪  "1", NA, NA, NA,~
```

Each row is a scholarly work. Key columns include `display_name` (the title), `publication_date`, `cited_by_count`, `doi`, and `source_display_name` (the source or journal name).

1.4.2 Basic descriptive statistics

How many works did we retrieve, and what does the citation distribution look like?

```
works_summary <- works |>
  summarise(
    n_works       = n(),
    doi_coverage  = scales::percent(mean(!is.na(doi))),
    median_cites  = median(cited_by_count, na.rm = TRUE),
    mean_cites    = round(mean(cited_by_count, na.rm = TRUE), 1),
    max_cites     = max(cited_by_count, na.rm = TRUE),
    zero_cites    = sum(cited_by_count == 0)
  )
```

```
works_summary
```

```
#> # A tibble: 1 x 6
#>   n_works doi_coverage median_cites mean_cites max_cites
 ↪  zero_cites
#>     <int> <chr>               <dbl>      <dbl>     <int>
 ↪  <int>
#> 1     500 90%                   7       22.4       446       104
```

The large gap between the mean and median is a hallmark of citation distributions: most papers receive few citations while a small number accumulate many. This skewness is one of the most robust empirical regularities in scientometrics, and we will return to its theoretical basis in 2.3.

1.4.3 Top journals

Which journals appear most often in our sample?

```
top_journals <- works |>
  filter(!is.na(source_display_name)) |>
  count(source_display_name, sort = TRUE) |>
  head(10)

top_journals
```

```
#> # A tibble: 10 x 2
#>    source_display_name
 ↪  n
#>    <chr>
 ↪  <int>
#>  1 Scientometrics
 ↪  10
#>  2 Sustainability
 ↪  7
#>  3 International Journal of Environmental Research and Public
 ↪  Health           5
#>  4 Journal of Informetrics
 ↪  5
#>  5 OSF Preprints (OSF Preprints)
 ↪  5
#>  6 PLoS ONE
 ↪  5
#>  7 DOAJ (DOAJ: Directory of Open Access Journals)
 ↪  4
```

```
#>  8 Environmental Science and Pollution Research
 ↪  4
#>  9 HAL
 ↪  4
#> 10 LA Referencia (Red Federada de Repositorios
 ↪  Institucionales de Publica~     4
```

1.4.4 Publication trend over time

```
works |>
  mutate(year = year(publication_date)) |>
  count(year) |>
  ggplot(aes(x = year, y = n)) +
  geom_col(fill = palette_sci(1)) +
  labs(
    x = "Publication year",
    y = "Number of works (sampled)"
  ) +
  theme_sci()
```

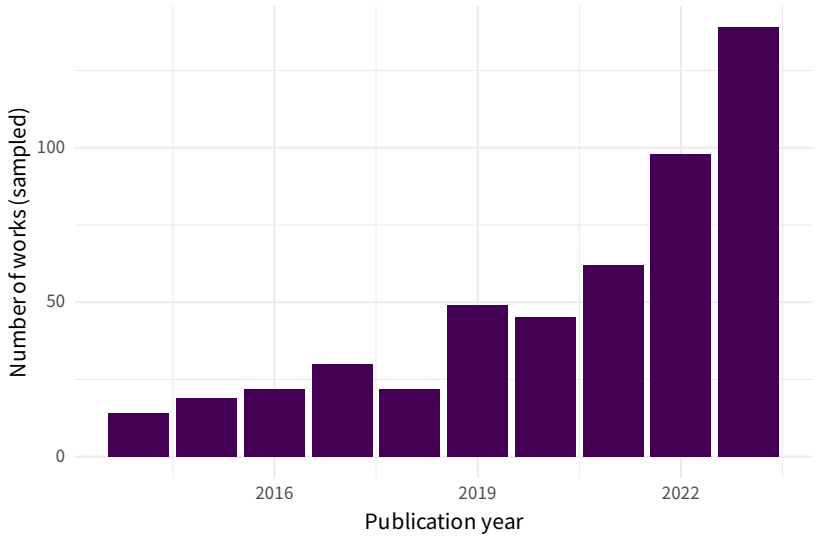


Figure 1.1: Annual publication counts for sampled works related to scientometrics (2014–2023).

The trend shows a growing volume of scientometrics-related publications over the past decade — consistent with the broader exponential growth of science documented by de Solla Price (1963).

1.4.5 Top journals visualised

```
top_journals |>
  mutate(source_display_name = fct_reorder(source_display_name,
  ↪  n)) |>
  ggplot(aes(x = n, y = source_display_name)) +
  geom_col(fill = palette_sci(1)) +
  labs(
    x = "Number of works",
    y = NULL
  ) +
  theme_sci()
```

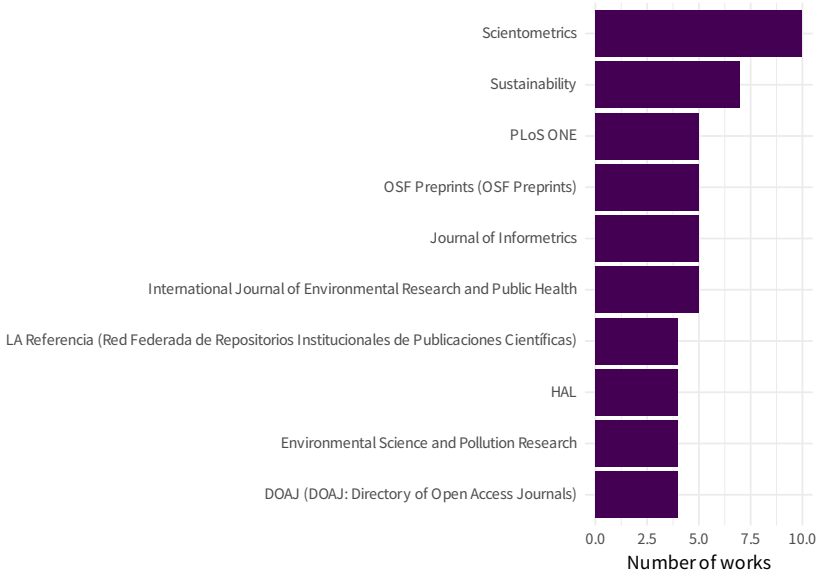


Figure 1.2: Top 10 journals by number of works in the sampled scientometrics corpus.

1.4.6 Citation distribution

```
works |>
  filter(cited_by_count > 0) |>
  ggplot(aes(x = cited_by_count)) +
  geom_histogram(bins = 40, fill = palette_sci(1), colour =
  ↪  "white") +
```

```
  scale_x_log10() +
  labs(
    x = "Citation count (log scale)",
    y = "Number of works"
  ) +
  theme_sci()
```

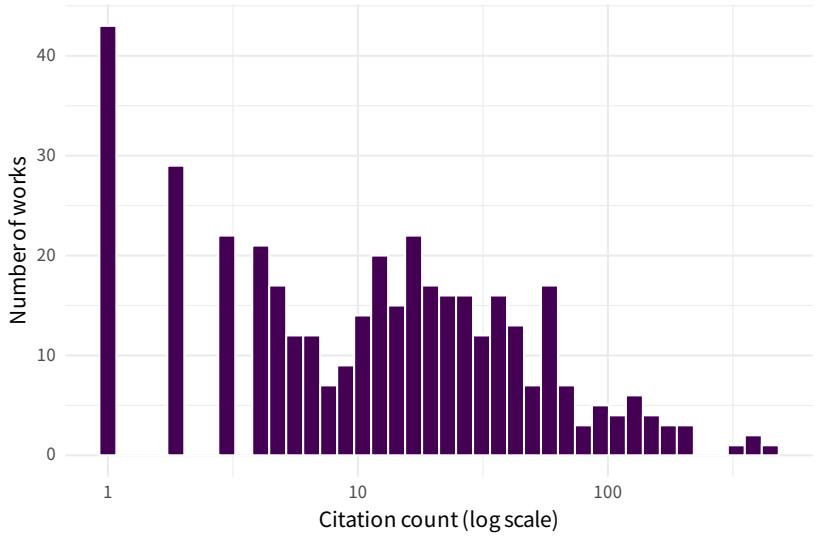


Figure 1.3: Citation distribution of sampled scientometrics works (log-scaled x-axis).

This right-skewed citation distribution is a universal feature of scholarly communication. In Chapter 2, we will see that it connects to deeper theoretical regularities: Lotka's Law, Bradford's Law, and the Matthew Effect.

1.5 Diagnostics and interpretation

Even for a simple descriptive analysis, a few diagnostic checks are essential:

- **Sample vs. population.** We drew a random sample of 500 works. The publication trend reflects the underlying population only if the sampling was truly random within the date window. OpenAlex's `sample` parameter does this, but always verify the date range of returned results.

- **Source coverage.** OpenAlex is strongest in English-language, peer-reviewed journal articles. Conference proceedings, book chapters, and grey literature may be underrepresented. Our "top journals" list reflects what OpenAlex indexes, not the full universe of scientometrics publishing.
- **Citation counts are snapshots.** The `cited_by_count` field reflects the current citation total at the time of the API query. Older papers have had more time to accumulate citations, so direct comparisons across years are misleading without normalization.

```
cat(glue("Date range: {min(works$publication_date)} to
 ↪  {max(works$publication_date)}\n"))
```

```
#> Date range: 2014-01-01 to 2023-12-31
```

```
cat(glue("Works with zero citations: {sum(works$cited_by_count
 ↪  == 0)}\n"))
```

```
#> Works with zero citations: 104
```

```
cat(glue("Unique journals:
 ↪  {n_distinct(works$source_display_name, na.rm = TRUE)}\n"))
```

```
#> Unique journals: 406
```

1.6 Limitations and responsible use

1.7 Limitations and responsible use

- **Descriptive statistics are not evaluations.** Counting publications and citations describes activity, not quality. A high publication count may reflect salami slicing; a high citation count may reflect controversy rather than excellence.
- **Data source bias.** OpenAlex, like all databases, has systematic coverage gaps. English-language STEM journals are overrepresented; social sciences, humanities, and non-English scholarship are underrepresented. Always report the data source and its known limitations.
- **Ecological fallacy.** Aggregate statistics about a field do not apply to individual researchers. A growing field does not mean every researcher in it is productive.
- **The Leiden Manifesto** (Hicks et al., 2015) and **DORA** (American Society for Cell Biology, 2012) provide essential guardrails. Quantitative data should support, never replace, expert qualitative judgement.

1.8 Common pitfalls

1.9 Common pitfalls

- **Confusing scientometrics with research evaluation.** Scientometrics provides tools for studying science; it does not provide verdicts on the value of individual researchers or papers.
- **Treating OpenAlex counts as ground truth.** OpenAlex counts may differ from Web of Science or Scopus due to different coverage, deduplication, and metadata parsing. Always triangulate when precision matters.
- **Ignoring the citation window.** Comparing raw citation counts across publication years is misleading. A 2023 paper with 5 citations may be performing better than a 2015 paper with 20. Field- and time-normalised indicators (covered in Chapter 10) address this.
- **Overgeneralising from a sample.** A random sample of 500 works is useful for learning and prototyping. Drawing conclusions about an entire field requires a comprehensive corpus.
- **Forgetting to set a seed.** Random samples are only reproducible if you set the seed both in R (`set.seed()`) and in the API call (`seed` parameter). Without this, every render produces different results.

1.10 Exercises

1. **Explore a different field.** Replace the search term "scientometrics" with a topic of your choice (e.g., "machine learning", "climate change"). Fetch 500 works and compare the publication trend and top journals to those we found above. What differences do you notice?

2. **Uncited works.** What fraction of the sampled works have zero citations? Filter to uncited works and examine their publication years. Is there a pattern? (Hint: very recent works are more likely to be uncited.)

3. **Open Access status.** OpenAlex records an `is_oa` field. Compute the proportion of open-access works in your sample and plot how it has changed over time. What trend do you observe?

4. **Growth curve.** Extend the publication trend analysis to 2000–2023. Does the annual publication count follow an exponential growth pattern, as de Solla Price (1963) predicted? (Hint: plot on a log-scaled y-axis and check for linearity.)

1.11 Solutions

Solutions are provided in E.

1.12 Further reading

- Garfield (1955) — The foundational paper proposing citation indexes for science; the intellectual starting point for the entire field.
- de Solla Price (1963) — Little Science, Big Science; the first systematic quantitative study of the growth of science.
- Priem et al. (2022) — The OpenAlex paper; describes the open data infrastructure that powers the examples in this book.
- Hicks et al. (2015) — The Leiden Manifesto; ten principles for the responsible use of research metrics.
- American Society for Cell Biology (2012) — The San Francisco Declaration on Research Assessment; a landmark call to improve research evaluation.
- Aria and Cuccurullo (2017) — The `bibliometrix` R package; a comprehensive tool for science mapping and bibliometric analysis.
- Hirsch (2005) — The h-index paper; illustrates how a single indicator reshaped research evaluation discourse.
- Lotka (1926) — The foundational study of scientific productivity distributions; a bridge to Chapter 2.

1.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
```

```
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] glue_1.8.1      openalexR_3.1.2 lubridate_1.9.5
 ↪  forcats_1.0.1
#>  [5] stringr_1.6.0   dplyr_1.2.1     purrr_1.2.2
 ↪  readr_2.2.0
#>  [9] tidyr_1.3.2     tibble_3.3.1    ggplot2_4.0.4
 ↪  tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gt_1.4.0           viridis_0.6.5      utf8_1.2.6
 ↪  generics_0.1.4
#>  [5] xml2_1.6.0         stringi_1.8.9      hms_1.1.4
 ↪  digest_0.6.39
#>  [9] magrittr_2.0.5     evaluate_1.0.5     grid_4.6.1
 ↪  timechange_0.4.0
#> [13] RColorBrewer_1.1-3 bookdown_0.48      fastmap_1.2.0
 ↪  rprojroot_2.1.1
#> [17] gridExtra_2.3.1    viridisLite_0.4.3  scales_1.4.0
 ↪  cli_3.6.6
#> [21] rlang_1.3.0        withr_3.0.3        yaml_2.3.12
 ↪  otel_0.3.0
#> [25] tools_4.6.1        tzdb_0.5.0         here_1.0.2
 ↪  vctrs_0.7.3
#> [29] R6_2.6.1           lifecycle_1.0.5    fs_2.1.0
 ↪  pkgconfig_2.0.3
#> [33] pillar_1.11.1      gtable_0.3.6       systemfonts_1.3.4
 ↪  xfun_0.62
#> [37] tidyselect_1.2.1   rstudioapi_0.19.0  knitr_1.52
 ↪  farver_2.1.2
#> [41] htmltools_0.5.9    labeling_0.4.3     rmarkdown_2.32
 ↪  compiler_4.6.1
#> [45] S7_1.0.0
```

Chapter 2

Theoretical Underpinnings

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

2.1 Learning objectives

After completing this chapter, you will be able to:

- State Lotka's, Bradford's, and Zipf's laws and explain the empirical patterns they describe
- Test whether a corpus of publications approximately follows Lotka's inverse-square law
- Construct a Bradford plot to identify core and peripheral journals in a field
- Explain Price's law of exponential growth and the Matthew Effect in science
- Recognise power-law-like distributions in bibliometric data

2.2 Setup

```
library(tidyverse)
library(openalexR)
library(tidytext)
library(glue)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "sci_palette.R"))
```

2.3 Conceptual background

Scientometrics rests on a small number of empirical regularities discovered across decades of studying scholarly communication:

Lotka's Law (Lotka, 1926) observes that the number of authors producing n papers is approximately proportional to $1/n^2$. A few prolific authors generate a large share of output, while most researchers publish only one or two papers.

Bradford's Law (Bradford, 1934) describes the scattering of articles across journals. For a given topic, a small core of journals contains roughly one-third of all relevant articles; a larger zone contains the next third; and a much larger periphery the final third. Each successive zone contains roughly the same number of articles but geometrically more journals.

Zipf's Law states that the frequency of a word is inversely proportional to its rank. In scientific text, a handful of terms dominate while the vast majority appear rarely.

Price's Law (de Solla Price, 1963) generalises these observations: the cumulative output of science grows exponentially, doubling roughly every 10–15 years. Price also noted that the square root of all contributors produce half of all contributions.

The **Matthew Effect** (Merton, 1968) — named after the Gospel of Matthew ("to those who have, more shall be given") — describes cumulative advantage in science. Well-known researchers attract disproportionate credit and resources, amplifying initial differences in visibility. This mechanism helps explain the skewed distributions observed by Lotka and Price.

2.4 Worked example

We use a corpus of works from the journal Scientometrics to demonstrate each law.

2.4.1 Fetching the data

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2015-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)
```

2.4.2 Lotka's Law

We unnest authors and count the number of papers per author in this sample.

```
author_prod <- works |>
  select(id, authorships) |>
  unnest(authorships, names_sep = "_") |>
  count(authorships_display_name, name = "n_papers") |>
  count(n_papers, name = "n_authors") |>
  arrange(n_papers)

author_prod
```

```
#> # A tibble: 8 x 2
#>   n_papers n_authors
#>      <int>     <int>
#> 1        1       891
#> 2        2       126
#> 3        3        27
#> 4        4        12
#> 5        5         4
#> 6        6         1
#> 7        7         1
#> 8        9         1
```

```
lotka_c <- author_prod$n_authors[1]

author_prod |>
  ggplot(aes(x = n_papers, y = n_authors)) +
  geom_point(colour = palette_sci(1), size = 2) +
  geom_line(
    aes(y = lotka_c / n_papers^2),
    linetype = "dashed", colour = "grey40"
  ) +
```

```
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Papers per author", y = "Number of authors") +
  theme_sci()
```

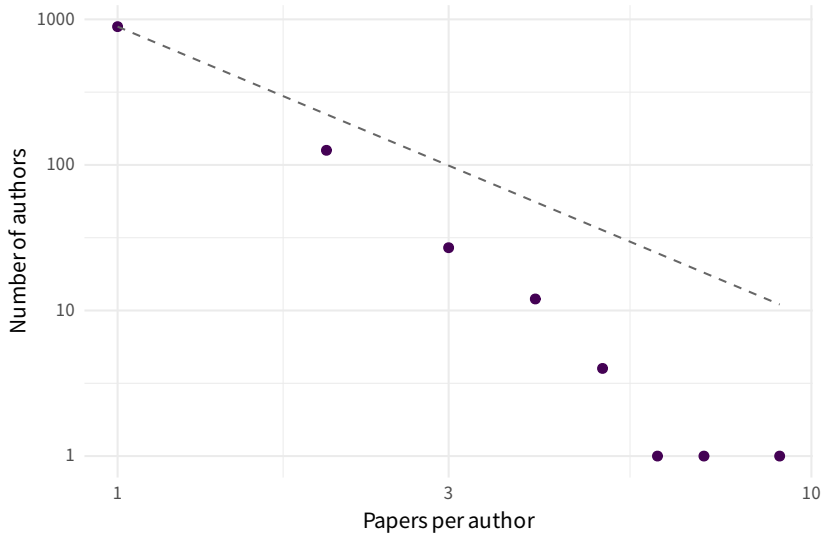


Figure 2.1: Author productivity distribution. The dashed line shows the inverse-square law predicted by Lotka.

The data approximate the inverse-square pattern, though real distributions often deviate from the exact exponent of 2.

2.4.3 Bradford's Law

We count how many articles each journal source contributes (across a broader query).

```
broad_works <- fetch_openalex(
  entity = "works",
  search = "bibliometrics",
  from_publication_date = "2020-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 1000, seed = 42)
)

journal_counts <- broad_works |>
  filter(!is.na(source_display_name)) |>
```

```
  count(source_display_name, name = "articles", sort = TRUE) |>
  mutate(
    cum_articles = cumsum(articles),
    rank = row_number()
  )
```

```
journal_counts |>
  ggplot(aes(x = rank, y = cum_articles)) +
  geom_line(colour = palette_sci(1), linewidth = 1) +
  scale_x_log10() +
  labs(x = "Journal rank (log scale)", y = "Cumulative
  ↪  articles") +
  theme_sci()
```

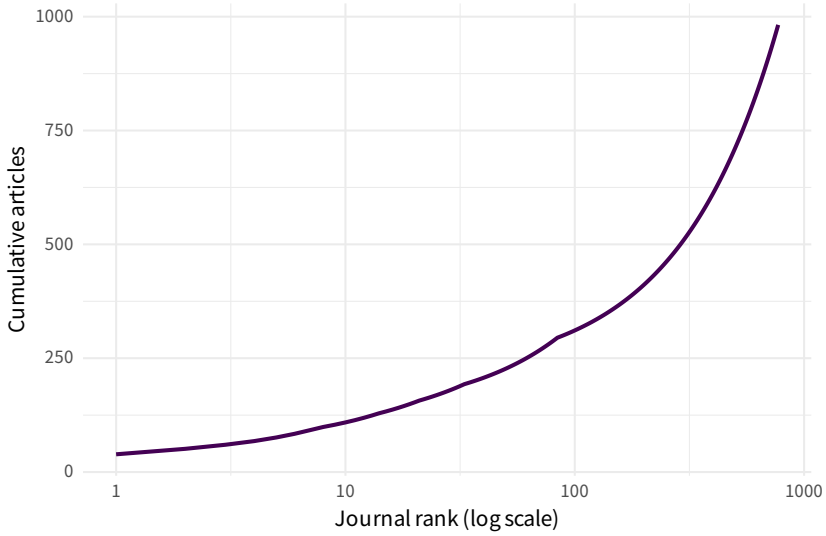


Figure 2.2: Bradford plot: cumulative articles vs. log journal rank.

2.4.4 Zipf's Law

We extract title words and examine their frequency distribution.

```
word_freq <- works |>
  select(display_name) |>
  unnest_tokens(word, display_name, token = "words") |>
  anti_join(tidytext::get_stopwords(), by = "word") |>
  count(word, sort = TRUE) |>
  mutate(rank = row_number())
```

```
word_freq |>
  head(200) |>
  ggplot(aes(x = rank, y = n)) +
  geom_point(colour = palette_sci(1), size = 1, alpha = 0.7) +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Rank", y = "Frequency") +
  theme_sci()
```

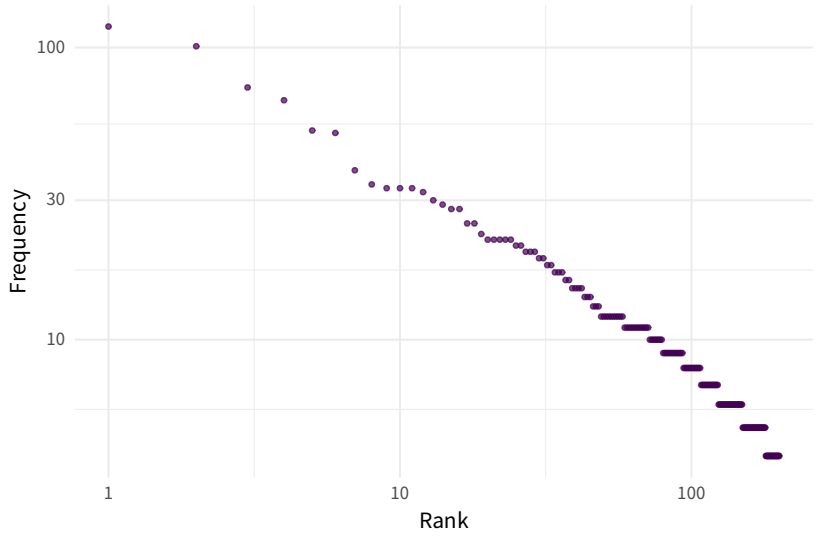


Figure 2.3: Word frequency vs. rank for article titles, approximating Zipf's law.

2.4.5 Price's growth law

```
works |>
  mutate(year = year(publication_date)) |>
  count(year) |>
  ggplot(aes(x = year, y = n)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Year", y = "Works in sample") +
  theme_sci()
```

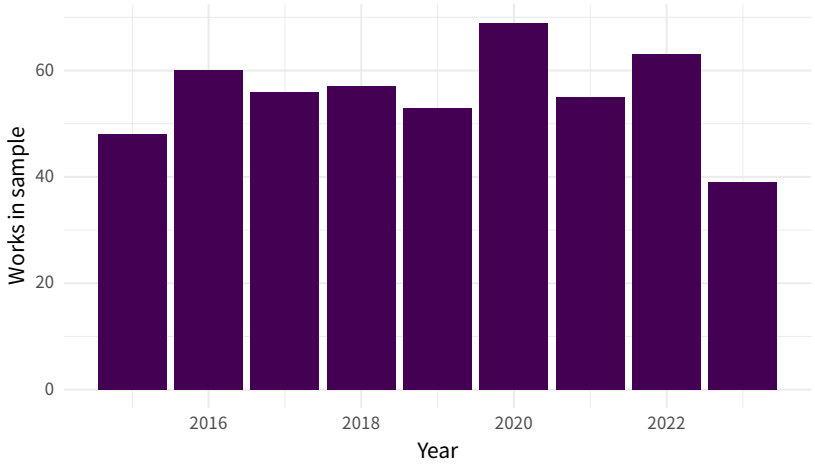


Figure 2.4: Annual publication counts for sampled works, illustrating the growth of bibliometrics literature.

2.5 Diagnostics and interpretation

These laws are empirical regularities, not physical laws. They describe approximate patterns:

- Lotka's exponent is often not exactly 2; values between 1.5 and 3.5 are common depending on the field and time window.
- Bradford zones are sensitive to how you define the subject scope.
- Zipf's law holds best for common words; the tail deviates.
- Price's exponential growth eventually saturates — science cannot grow faster than the economy indefinitely.

The value of these laws is not predictive precision but the insight that **skewed distributions are the norm, not the exception**, in scholarly communication.

2.6 Limitations and responsible use

2.7 Limitations and responsible use

- These laws describe aggregate patterns, not individual behaviour. A researcher publishing one paper is not "less productive" in any evalua-

tive sense — Lotka's law is a statistical observation, not a performance standard.
- The Matthew Effect (Merton, 1968) is a mechanism that amplifies inequality. Recognising it should prompt caution about using cumulative metrics (like the h-index) that inherit and reinforce this bias.
- Power-law fitting is notoriously difficult. Many distributions that "look" power-law on a log-log plot are better described by lognormal or stretched exponential models. Rigorous fitting requires maximum-likelihood methods and goodness-of-fit tests (Hicks et al., 2015).

2.8 Common pitfalls

2.9 Common pitfalls

- **Eyeballing log-log plots.** A straight line on a log-log plot does not prove a power law. Use formal statistical tests.
- **Confusing description with prescription.** "Most authors publish few papers" does not mean they should publish more.
- **Ignoring field differences.** The exponent of Lotka's law and the scattering pattern of Bradford's law vary substantially across disciplines.
- **Assuming stationarity.** These patterns can shift over time as publication norms change (e.g., the rise of mega-journals).

2.10 Exercises

1. **Lotka exponent.** Fit a linear model to the log-log author productivity data. What exponent do you estimate? How does it compare to Lotka's original value of 2?
2. **Bradford zones.** Divide the journal list into three zones of roughly equal article counts. How many journals are in each zone? What is the ratio between successive zone sizes?
3. **Your field's Zipf distribution.** Fetch 500 works in a field of your choice. Extract title words and plot the Zipf distribution. Are the most frequent terms what you would expect?
4. **Matthew Effect simulation.** Write a simple simulation: start 100 authors with 1 paper each. At each time step, an author gains a new paper with probability proportional to their current count. After 50 steps, plot the distribution. Does it resemble Lotka's law?

2.11 Solutions

Solutions are provided in E.

2.12 Further reading

- Lotka (1926) — The original observation on author productivity distributions.
- Bradford (1934) — Journal scattering and the Bradford law.
- de Solla Price (1963) — Exponential growth of science and cumulative advantage.
- Merton (1968) — The Matthew Effect: how advantage accumulates in science.
- Garfield (1955) — Citation indexing, the empirical basis for much of scientometrics.
- Waltman (2016) — A modern review contextualising these classical laws within citation impact measurement.

2.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
```

```
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] tidytext_0.4.3  glue_1.8.1      openalexR_3.1.2
 ↪  lubridate_1.9.5
#>  [5] forcats_1.0.1   stringr_1.6.0   dplyr_1.2.1
 ↪  purrr_1.2.2
#>  [9] readr_2.2.0     tidyr_1.3.2     tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [13] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] janeaustenr_1.0.0  gt_1.4.0           viridis_0.6.5
 ↪  utf8_1.2.6
#>  [5] generics_0.1.4     xml2_1.6.0         lattice_0.22-9
 ↪  stringi_1.8.9
#>  [9] hms_1.1.4          digest_0.6.39      magrittr_2.0.5
 ↪  evaluate_1.0.5
#> [13] grid_4.6.1         timechange_0.4.0   RColorBrewer_1.1-3
 ↪  bookdown_0.48
#> [17] fastmap_1.2.0      Matrix_1.7-5       rprojroot_2.1.1
 ↪  gridExtra_2.3.1
#> [21] stopwords_2.3      viridisLite_0.4.3  scales_1.4.0
 ↪  cli_3.6.6
#> [25] rlang_1.3.0        tokenizers_0.3.0   withr_3.0.3
 ↪  yaml_2.3.12
#> [29] otel_0.3.0         tools_4.6.1        tzdb_0.5.0
 ↪  here_1.0.2
#> [33] vctrs_0.7.3        R6_2.6.1           lifecycle_1.0.5
 ↪  fs_2.1.0
#> [37] pkgconfig_2.0.3    pillar_1.11.1      gtable_0.3.6
 ↪  Rcpp_1.1.2
#> [41] systemfonts_1.3.4  xfun_0.62          tidyselect_1.2.1
 ↪  rstudioapi_0.19.0
#> [45] knitr_1.52         farver_2.1.2       SnowballC_0.7.1
 ↪  htmltools_0.5.9
#> [49] labeling_0.4.3     rmarkdown_2.32     compiler_4.6.1
 ↪  S7_1.0.0
```

Chapter 3

Ethics and Responsible Metrics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

3.1 Learning objectives

After completing this chapter, you will be able to:

- Summarize the ten principles of the Leiden Manifesto and explain their implications for research evaluation
- Describe the key commitments of the San Francisco Declaration on Research Assessment (DORA)
- Identify common forms of metric gaming, including self-citation cartels, citation stacking, and salami slicing
- Evaluate when quantitative indicators are appropriate — and when they are not
- Compute and interpret author-level self-citation rates using OpenAlex data
- Apply a responsible-use framework when designing or reviewing bibliometric analyses

3.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

3.3 Conceptual background

Metrics shape behaviour. When a number is used to judge researchers, departments, or journals, people adjust their actions to improve that number — regardless of whether the number actually captures what it was meant to measure. This insight, often called **Goodhart's Law**, is the intellectual foundation for the responsible metrics movement: "When a measure becomes a target, it ceases to be a good measure" (Goodhart, 1984).

3.3.1 The Leiden Manifesto

In 2015, Hicks et al. (2015) published the **Leiden Manifesto for Research Metrics** — ten principles distilled from decades of experience in bibliometric practice. The principles are:

1. **Quantitative evaluation should support qualitative, expert assessment.** Metrics should inform human judgement, never replace it.
2. **Measure performance against the research missions of the institution, group, or researcher.** Not everyone pursues the same goals; evaluation criteria must reflect this.
3. **Protect excellence in locally relevant research.** Research published in regional languages or addressing local problems must not be penalised for lower citation rates.
4. **Keep data collection and analytical processes open, transparent, and simple.** Black-box indicators erode trust and prevent scrutiny.
5. **Allow those evaluated to verify data and analysis.** Researchers should be able to inspect and challenge the data used to judge them.
6. **Account for variation by field in publication and citation practices.** Citation rates in mathematics and molecular biology differ by an order of magnitude; raw counts cannot be compared across fields.

7. **Base assessment of individual researchers on a qualitative judgement of their portfolio.** A single number (such as the h-index) cannot capture the breadth of a researcher's contribution.
8. **Avoid misplaced concreteness and false precision.** Three decimal places do not make a flawed indicator meaningful.
9. **Recognize the systemic effects of assessment and indicators.** Indicators reshape the system they measure. If you reward publication volume, you incentivise salami slicing.
10. **Scrutinize indicators regularly and update them.** The research landscape evolves; indicators must evolve with it.

3.3.2 The San Francisco Declaration (DORA)

The **San Francisco Declaration on Research Assessment** (American Society for Cell Biology, 2012) emerged from the 2012 Annual Meeting of the American Society for Cell Biology. Its central demand is straightforward: do not use journal-level metrics — especially the Journal Impact Factor — as a proxy for the quality of individual research articles. DORA's key commitments include:

- **For funders and institutions**: evaluate research on its own merits, not on the journal in which it was published. Use article-level metrics and qualitative indicators.
- **For publishers**: reduce emphasis on the Journal Impact Factor as a promotional tool. Make article-level metrics (downloads, citations, altmetrics) available.
- **For researchers**: challenge evaluation practices that rely on journal-based metrics. When serving on hiring or promotion committees, assess the content of papers, not where they appeared.

As of 2026, DORA has been signed by over 3,000 organisations and 20,000 individuals. Many major funders — including the European Commission, Wellcome Trust, and the Dutch Research Council (NWO) — have formally adopted DORA principles.

3.3.3 The Metric Tide

The UK's independent review of metrics in research assessment, The Metric Tide (Wilsdon et al., 2015), reinforced these principles with a systematic evidence base. The report concluded that no metric can substitute for expert peer review, and recommended that the UK research system adopt a framework of "responsible metrics" built on five dimensions: robustness, humility, transparency, diversity, and reflexivity.

3.3.4 Goodhart's Law in practice

When metrics are attached to career consequences, the incentive to game them becomes powerful (Goodhart, 1984). Common forms of metric manipulation include:

- **Self-citation inflation**: authors systematically cite their own prior work to boost citation counts, even when those citations are not intellectually necessary.
- **Citation cartels**: informal agreements among groups of authors or journals to cite each other's work, inflating citation counts reciprocally.
- **Citation stacking**: journals encourage or require authors to add citations to recently published articles in the same journal, artificially increasing the Impact Factor.
- **Salami slicing**: splitting a single study into the maximum number of "least publishable units" to inflate publication counts.

These practices are not merely hypothetical. Clarivate has repeatedly suspended journals from the Journal Citation Reports for citation stacking, and several high-profile cases of self-citation manipulation have led to editorial sanctions.

3.3.5 Publish or perish

The systemic root of metric gaming is the **"publish or perish"** culture that dominates academic career structures worldwide. When hiring, promotion, and tenure decisions are driven by publication volume and citation counts, researchers face relentless pressure to produce more papers, chase high-impact journals, and avoid risky long-term projects. The consequences include:

- **Reduced risk-taking**: researchers avoid novel or interdisciplinary questions that might not produce quick, high-citation publications.
- **Replication crisis**: pressure to publish positive results contributes to publication bias and irreproducible findings.
- **Mental health harms**: the relentless pressure to produce has been linked to elevated rates of anxiety, burnout, and attrition in academia.
- **Erosion of research quality**: when volume is rewarded over substance, the incentive structure favours quantity at the expense of depth and rigour.

These dynamics make the responsible use of metrics not merely a technical concern but an ethical imperative.

3.4 Worked example

We now demonstrate a practical analysis: computing self-citation rates for a sample of researchers in the field of scientometrics. This illustrates how OpenAlex data can be used to flag potential self-citation inflation — and the care required to interpret such analysis responsibly.

3.4.1 Acquiring author data

We fetch a sample of authors associated with scientometrics research who have at least 50 works, ensuring we have enough output to compute meaningful self-citation rates.

```
authors_raw <- fetch_openalex(
  entity = "authors",
  topics.id = "T10102",
  works_count = ">50",
  options = list(sample = 30, seed = 42)
)

authors <- authors_raw |>
  select(id, display_name, works_count, cited_by_count) |>
  filter(!is.na(display_name))

glimpse(authors)
```

```
#> Rows: 30
#> Columns: 4
#> $ id             <chr> "https://openalex.org/A5052369892",
 ↪  "https://openalex.o~
#> $ display_name   <chr> "Wei Hua Zhou", "А. Н. Либкинд",
 ↪  "Lauren Lanahan", "Fra~
#> $ works_count    <int> 284, 59, 75, 395, 107, 408, 53, 86,
 ↪  144, 204, 74, 66, 7~
#> $ cited_by_count <int> 4151, 271, 1324, 40, 483, 7081, 136,
 ↪  674, 2061, 1, 1816~
```

3.4.2 Fetching works and computing self-citations

For each author, we retrieve a sample of their recent works and examine how many of the citations come from the author's own papers. OpenAlex provides `cited_by_count` at the work level and allows us to query an author's works directly.

```
# For a manageable demonstration, pick 10 authors
authors_sample <- authors |>
  slice_sample(n = min(10, nrow(authors)))

# Function to estimate self-citation rate for one author
estimate_self_citation <- function(author_id, author_name) {
  # Fetch up to 50 recent works by this author
  works <- tryCatch(
    fetch_openalex(
      entity = "works",
      author.id = author_id,
      from_publication_date = "2018-01-01",
      options = list(sample = 50, seed = 42)
    ),
    error = function(e) tibble()
  )

  if (is.null(works) || nrow(works) == 0) {
    return(tibble(
      author_id = author_id,
      author_name = author_name,
      n_works = 0L,
      total_citations = NA_integer_,
      self_citations = NA_integer_,
      self_cite_rate = NA_real_
    ))
  }

  # Count references that point to the same author's works
  # We look at the referenced_works and check which are by the
   ↪  same author
  total_cites <- sum(works$cited_by_count, na.rm = TRUE)

  # For self-citation, we check which of the works'
   ↪  referenced_works

  # are also in the author's own works list
  own_work_ids <- works$id

  self_cite_count <- works |>
    pull(referenced_works) |>
    map_int(~ sum(.x %in% own_work_ids, na.rm = TRUE))

  tibble(
    author_id = author_id,
    author_name = author_name,
    n_works = nrow(works),
    total_references = sum(map_int(works$referenced_works,
     ↪  length)),
    self_references = sum(self_cite_count),
```

```
    self_cite_rate = if_else(
      sum(map_int(works$referenced_works, length)) > 0,
      sum(self_cite_count) / sum(map_int(works$referenced_works,
       ↪  length)),
      NA_real_
    )
  )
}

self_cite_results <- map2_dfr(
  authors_sample$id,
  authors_sample$display_name,
  estimate_self_citation
)
```

3.4.3 Summarizing results

```
self_cite_results |>
  filter(!is.na(self_cite_rate)) |>
  arrange(desc(self_cite_rate)) |>
  mutate(self_cite_rate = scales::percent(self_cite_rate,
   ↪  accuracy = 0.1)) |>
  gt() |>
  tab_header(
    title = "Self-citation rates among sampled scientometrics
     ↪  authors",
    subtitle = "Based on references in up to 50 recent works per
     ↪  author"
  ) |>
  cols_label(
    author_name = "Author",
    n_works = "Works sampled",
    total_references = "Total references",
    self_references = "Self-references",
    self_cite_rate = "Self-citation rate"
  ) |>
  cols_hide(author_id)
```

Self-citation rates among sampled scientometrics authors

Based on references in up to 50 recent works per author

Author	Works sampled	Total references	Self-references	Self-citation rate	total_citations	self_citations
George W. Chacko	43	1165	65	5.6%	NA	NA

Jennifer Tetzlaff	18	1057	27	2.6%	NA	NA
Mehmet Boyraz	45	857	18	2.1%	NA	NA
Khaled Moustafa	50	915	15	1.6%	NA	NA
Adrián A. Díaz‐Faes	50	1199	13	1.1%	NA	NA
Catalina Martínez	22	1110	7	0.6%	NA	NA
Ramakanth R. Yakkanti	50	1391	3	0.2%	NA	NA
Françoise Gouzi	26	29	0	0.0%	NA	NA
Keith J. Sweeney	21	420	0	0.0%	NA	NA

3.4.4 Visualization

```
self_cite_clean <- self_cite_results |>
  filter(!is.na(self_cite_rate))

if (nrow(self_cite_clean) > 0) {
  ggplot(self_cite_clean, aes(x = self_cite_rate)) +
    geom_histogram(
      binwidth = 0.02,
      fill = palette_sci(1),
      colour = "white"
    ) +
    geom_vline(
      xintercept = median(self_cite_clean$self_cite_rate, na.rm
      ↪  = TRUE),
      linetype = "dashed", linewidth = 0.8
    ) +
    scale_x_continuous(labels = scales::percent_format()) +
    labs(
      x = "Self-citation rate (proportion of references to own
      ↪  works)",
      y = "Number of authors"
    ) +
    theme_sci()
}
```

3.4.5 A per-author comparison

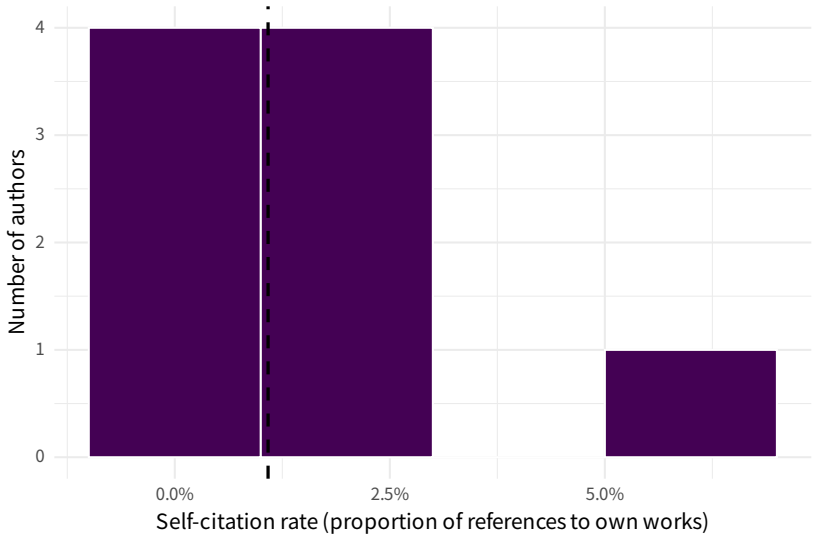


Figure 3.1: Distribution of self-citation rates across sampled scientometrics authors. The dashed line marks the sample median.

```
if (nrow(self_cite_clean) > 0) {
  self_cite_clean |>
    mutate(author_name = fct_reorder(author_name,
     ↪  self_cite_rate)) |>
    ggplot(aes(x = self_cite_rate, y = author_name)) +
    geom_col(fill = palette_sci(1)) +
    scale_x_continuous(labels = scales::percent_format()) +
    labs(x = "Self-citation rate", y = NULL) +
    theme_sci()
}
```

3.5 Diagnostics and interpretation

Self-citation rates vary enormously across disciplines, career stages, and research specialities. When interpreting these results, keep the following in mind:

- **Baseline rates differ by field.** In small, specialised fields, self-citation is often necessary because few other researchers work on the same topic. Self-citation rates of 15–25% are common in niche subfields and do not, by themselves, indicate manipulation.

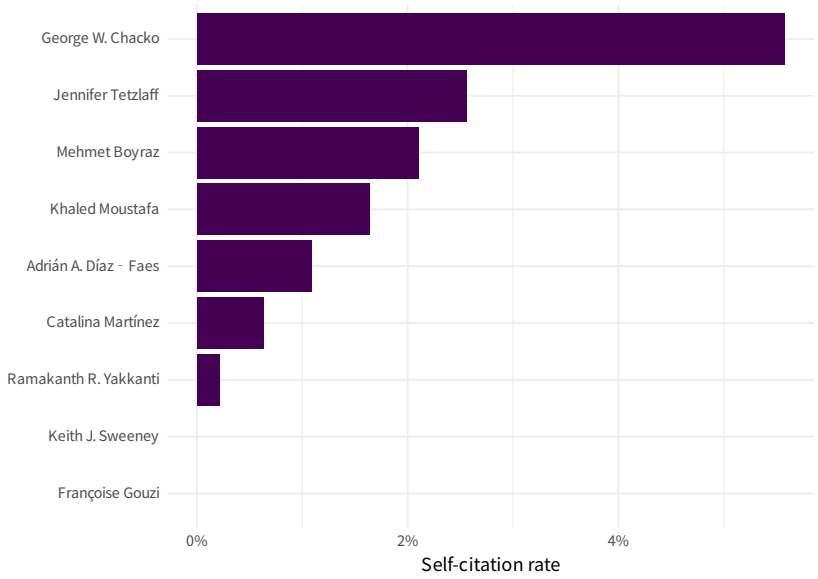


Figure 3.2: Self-citation rates by author, sorted from highest to lowest.

- **Career stage matters.** Senior researchers with large publication portfolios naturally accumulate more self-citations simply because they have more prior work to build on.
- **Our estimate is a lower bound.** We only count self-references to works that appear in the author's sampled works from OpenAlex. Works outside our sample window (before 2018) or works not indexed by OpenAlex are missed, so the true self-citation rate is likely higher.
- **Context is essential.** A self-citation rate of 30% might be perfectly reasonable for a researcher building a coherent long-term programme, while a rate of 10% could still be problematic if those citations are strategically placed to inflate a particular metric.
- **Never use self-citation rates alone to accuse individuals of misconduct.** As the Leiden Manifesto insists, quantitative indicators must support — not replace — qualitative judgement (Hicks et al., 2015).

3.6 Limitations and responsible use

3.7 Limitations and responsible use

- **Self-citation analysis is not misconduct detection.** High self-citation rates have many legitimate explanations. Presenting a

self-citation analysis as evidence of gaming without qualitative context is itself a form of metric misuse.
- **Coverage gaps matter.** OpenAlex does not index all scholarly works. Authors who publish in venues not well covered by OpenAlex will have incomplete reference lists, biasing self-citation estimates.
- **Author disambiguation is imperfect.** OpenAlex uses algorithmic author disambiguation. Common names may be conflated (merging two people into one profile) or split (one person appearing as multiple profiles), distorting individual-level statistics.
- **Snapshot bias.** Our analysis uses a sample of works from a particular time window. Conclusions should not be generalised beyond that window without further analysis.
- **The Leiden Manifesto applies here too.** Every principle in 3.3 applies to our own analysis. We must be transparent about methods, allow those evaluated to verify data, and avoid false precision (Hicks et al., 2015).

3.8 Common pitfalls

3.9 Common pitfalls

- **Equating self-citation with misconduct.** Self-citation is a normal part of scientific writing. Only excessive, strategically motivated self-citation is problematic — and even then, the boundary is context-dependent.
- **Using the Journal Impact Factor to judge individual articles.** This is precisely what DORA warns against. A journal's average citation rate says nothing about any single paper in that journal (American Society for Cell Biology, 2012).
- **Ignoring field differences when comparing metrics.** A citation count of 20 is exceptional in mathematics but unremarkable in biomedical research. Any cross-field comparison requires field normalisation (Waltman, 2016).
- **Treating indicators as objective truth.** All bibliometric indicators are constructed measures with specific assumptions, coverage limitations, and known biases. Present them as approximations, not facts.
- **Forgetting Goodhart's Law.** If you propose a new indicator for evaluation purposes, consider how rational actors will respond to being measured by it. Every metric creates incentives, and those incentives may not align with the goals of good research (Goodhart, 1984).

3.10 Exercises

1. **Leiden Manifesto audit.** Select a recent hiring or promotion policy document from a university (many are publicly available). Evaluate it against all ten Leiden Manifesto principles. Which principles does it satisfy? Which does it violate? Write a one-page assessment.
2. **Self-citation across fields.** Modify the worked example to compare self-citation rates between two different research fields (e.g., "bibliometrics" and "genomics"). Do you observe different baseline self-citation rates? What might explain the difference?
3. **DORA compliance check.** Visit the DORA website (sfdora.org) and identify three major funders that have signed the declaration. For each, find their current grant evaluation criteria and assess whether their stated practices are consistent with DORA commitments.
4. **Goodhart's Law thought experiment.** Propose a new indicator for measuring research quality that is not currently in wide use. Then, write a brief analysis of how researchers might game this indicator if it were adopted for promotion decisions. (Hint: think about both the numerator and denominator of any ratio-based metric.)

3.11 Solutions

Solutions are provided in E.

3.12 Further reading

- Hicks et al. (2015) — The Leiden Manifesto for Research Metrics. The single most important reference on responsible use of bibliometric indicators.
- American Society for Cell Biology (2012) — The San Francisco Declaration on Research Assessment. The landmark declaration against using journal-level metrics to evaluate individual researchers.
- Wilsdon et al. (2015) — The Metric Tide. A comprehensive UK government review of the role of metrics in research assessment, proposing a framework of responsible metrics.
- Goodhart (1984) — Goodhart's original articulation of the principle that a measure ceases to be useful when it becomes a target.
- Waltman (2016) — A thorough review of citation impact indicators, including their known biases and appropriate uses.

- Merton (1968) — The Matthew Effect in science; describes how advantage accumulates, a dynamic that citation metrics can reinforce.
- Garfield (1955) — The foundational paper on citation indexing. Essential for understanding the origins of the system we now critique.

3.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] gt_1.4.0        tidytext_0.4.3  glue_1.8.1
 ↪  openalexR_3.1.2
#>  [5] lubridate_1.9.5 forcats_1.0.1   stringr_1.6.0
 ↪  dplyr_1.2.1
#>  [9] purrr_1.2.2     readr_2.2.0     tidyr_1.3.2
 ↪  tibble_3.3.1
#> [13] ggplot2_4.0.4   tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
```

```
#>  [1] janeaustenr_1.0.0  viridis_0.6.5      utf8_1.2.6
 ↪  generics_0.1.4
#>  [5] xml2_1.6.0         lattice_0.22-9     stringi_1.8.9
 ↪  hms_1.1.4
#>  [9] digest_0.6.39      magrittr_2.0.5     evaluate_1.0.5
 ↪  grid_4.6.1
#> [13] timechange_0.4.0   RColorBrewer_1.1-3 bookdown_0.48
 ↪  fastmap_1.2.0
#> [17] jsonlite_2.0.0     Matrix_1.7-5       rprojroot_2.1.1
 ↪  gridExtra_2.3.1
#> [21] httr_1.4.9         stopwords_2.3      viridisLite_0.4.3
 ↪  scales_1.4.0
#> [25] cli_3.6.6          rlang_1.3.0        tokenizers_0.3.0
 ↪  withr_3.0.3
#> [29] yaml_2.3.12        otel_0.3.0         tools_4.6.1
 ↪  tzdb_0.5.0
#> [33] here_1.0.2         curl_8.1.0         vctrs_0.7.3
 ↪  R6_2.6.1
#> [37] lifecycle_1.0.5    fs_2.1.0           pkgconfig_2.0.3
 ↪  pillar_1.11.1
#> [41] gtable_0.3.6       Rcpp_1.1.2         systemfonts_1.3.4
 ↪  xfun_0.62
#> [45] tidyselect_1.2.1   rstudioapi_0.19.0  knitr_1.52
 ↪  farver_2.1.2
#> [49] SnowballC_0.7.1    htmltools_0.5.9    labeling_0.4.3
 ↪  rmarkdown_2.32
#> [53] compiler_4.6.1     S7_1.0.0
```

Chapter 4

Data Sources Compared

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

4.1 Learning objectives

After completing this chapter, you will be able to:

- Describe the coverage, access model, and key strengths of twelve major bibliographic data sources
- Select an appropriate data source (or combination of sources) for a given research question
- Compare metadata completeness across sources for the same set of publications
- Evaluate the trade-offs between proprietary and open data sources for reproducible research
- Retrieve and compare metadata from OpenAlex and Crossref using R

4.2 Setup

```
library(tidyverse)
library(openalexR)
library(rcrossref)
library(glue)
library(gt)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

4.3 Conceptual background

Every bibliometric analysis is fundamentally constrained by the data source that feeds it. The database you choose determines which publications are visible, which citations are counted, and which metadata fields are available. As Visser et al. (2021) demonstrated in a large-scale comparison, the major bibliographic databases differ substantially in coverage, with these differences affecting research outcomes in measurable ways. Similarly, Singh et al. (2021) showed that journal coverage varies considerably across Web of Science, Scopus, and Dimensions, with each database exhibiting distinct disciplinary strengths.

The choice of data source is not merely a technical decision — it is a methodological one with direct consequences for the validity of results. A citation analysis conducted on Web of Science will produce different numbers than the same analysis on Scopus, Dimensions, or OpenAlex, because each database indexes a different set of publications and computes citations over a different corpus. Researchers must therefore understand these differences and report them transparently, as emphasised by the Leiden Manifesto's fourth principle: keep data collection and analytical processes open, transparent, and simple (Hicks et al., 2015).

The landscape of bibliographic data sources has shifted dramatically in recent years. The traditional duopoly of Web of Science and Scopus has been challenged by open alternatives — most notably OpenAlex (Priem et al., 2022), which provides free, unrestricted access to over 250 million scholarly works. Crossref serves as the backbone registration agency for DOIs and provides rich metadata for over 150 million records. Discipline-specific databases like PubMed, ADS, and arXiv serve targeted communities with deep, curated coverage.

For this book, we adopt OpenAlex as the primary data source because it is fully open, freely accessible via API without authentication, and provides coverage comparable to proprietary alternatives. However, responsible bibliometric practice often requires consulting multiple sources to cross-validate findings and fill coverage gaps.

4.3.1 Twelve data sources at a glance

Below we summarise twelve major sources. The subsequent sections examine each in detail, and the comparison table provides a structured overview.

Web of Science (WoS). Maintained by Clarivate, WoS is the oldest and most established citation index, originating from Eugene Garfield's Science Citation Index (Garfield, 1955). It covers approximately 90 million records from over 21,000 journals, with citation data back to 1900. Access requires an institutional subscription. Its strengths are curated journal selection, long historical coverage, and the Journal Citation Reports (JCR). Its main limitations are cost, restricted API access, and lower coverage of non-English literature, social sciences, and humanities.

Scopus. Elsevier's Scopus indexes over 94 million records from approximately 27,000 journals, with broader coverage of non-English titles and conference proceedings than WoS. It provides its own citation metrics (CiteScore, SJR, SNIP) and a well-documented API. Like WoS, it requires an institutional subscription.

Dimensions. Operated by Digital Science, Dimensions indexes over 140 million publications including journal articles, books, book chapters, preprints, clinical trials, patents, and policy documents. It offers a free tier with limited functionality and a paid tier with full API access. Its distinctive strength is linking publications to grants, patents, clinical trials, and policy documents.

OpenAlex. The successor to the discontinued Microsoft Academic Graph, OpenAlex is a fully open, freely accessible index of over 250 million scholarly works (Priem et al., 2022). It aggregates data from Crossref, PubMed, institutional repositories, and other open sources. Its API requires no authentication (though a polite-pool email is recommended). OpenAlex provides author disambiguation, institutional affiliation data, and concept tagging. Its primary limitations are metadata quality (algorithmically parsed, not manually curated) and less granular subject classification than curated databases.

The Lens. Operated by Cambia, The Lens provides free access to over 300 million scholarly works and 160 million patent records. It aggregates data from Crossref, PubMed, Microsoft Academic, and patent offices. Its unique strength is integrated patent-paper linkage, making it valuable for technology-transfer and innovation studies. API access is available via a free institutional plan.

Crossref. Crossref is the DOI registration agency for scholarly publications, covering over 150 million records. It is not a citation index per se — it provides metadata deposited by publishers, including titles, authors, abstracts, reference lists, and licensing information. Its REST API is freely accessible. Crossref's strengths are comprehensive DOI coverage and de-

tailed reference metadata; its limitation is that metadata quality depends entirely on what publishers deposit.

PubMed/MEDLINE. Maintained by the US National Library of Medicine, PubMed indexes over 36 million records in the biomedical and life sciences. Its key strengths are curated MeSH (Medical Subject Headings) indexing, structured abstracts, and deep coverage of clinical and biomedical literature. PubMed does not provide its own citation counts but can be paired with citation data from other sources.

Google Scholar. Google Scholar provides the broadest coverage of any scholarly search engine, indexing journal articles, conference papers, theses, books, patents, and grey literature. However, it provides no API, no bulk data export, and no structured metadata. It is useful for discovery and for its "Cited by" feature but is not suitable for systematic bibliometric analysis.

Semantic Scholar. Operated by the Allen Institute for AI, Semantic Scholar indexes over 200 million papers and provides a free API with rich metadata including algorithmically extracted entities, TLDR summaries, and citation context. It is particularly strong in computer science, biomedical, and neuroscience literature.

NASA/ADS (Astrophysics Data System). ADS is a curated database of over 16 million records in astronomy, astrophysics, and physics. It provides excellent citation tracking, full-text search, and a well-documented API. For astronomical research, ADS is the gold standard.

arXiv. The original preprint server, arXiv hosts over 2.5 million preprints in physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, and electrical engineering. It does not provide citation data directly but its records are indexed by other databases. Its API provides metadata access.

bioRxiv/medRxiv. The preprint servers for biology and health sciences, hosting over 300,000 preprints. They provide API access to metadata and full text and are indexed by OpenAlex, Dimensions, and other aggregators.

4.4 Worked example

We now compare what OpenAlex and Crossref return for the same set of DOIs. This is a practical illustration of how metadata completeness varies across sources.

4.4.1 Selecting DOIs

We start with a small set of well-known scientometrics papers whose DOIs we know.

```
dois <- c(
  "10.1038/520429a",            # Hicks et al. 2015, Leiden
  ↪  Manifesto
  "10.1073/pnas.0507655102",    # Hirsch 2005, h-index
  "10.1007/s11192-006-0144-7",  # Egghe 2006, g-index
  "10.1016/j.joi.2017.08.007",  # Aria & Cuccurullo 2017,
  ↪  bibliometrix
  "10.1126/science.122.3159.108", # Garfield 1955, citation
  ↪  indexes
  "10.48550/arXiv.2205.01833",  # Priem et al. 2022, OpenAlex
  "10.1162/qss_a_00112",        # Visser et al. 2021, database
  ↪  comparison
  "10.1007/s11192-021-03948-5"  # Singh et al. 2021, journal
  ↪  coverage
)
```

4.4.2 Fetching from OpenAlex

```
oa_results <- fetch_openalex(
  entity = "works",
  doi = dois
)

oa_meta <- oa_results |>
  transmute(
    doi = doi,
    source = "OpenAlex",
    title = display_name,
    year = publication_year,
    cited_by = cited_by_count,
    has_abstract = !is.na(abstract),
    type = type
  )

glimpse(oa_meta)
```

```
#> Rows: 8
#> Columns: 7
#> $ doi          <chr>
 ↪  "https://doi.org/10.1016/j.joi.2017.08.007", "https://doi~
```

```
#> $ source       <chr> "OpenAlex", "OpenAlex", "OpenAlex",
 ↪  "OpenAlex", "OpenAlex~
#> $ title        <chr> "bibliometrix : An R-tool for
 ↪  comprehensive science mappi~
#> $ year        <int> 2017, 2005, 2015, 1955, 2006, 2021, 2021,
 ↪  2022
#> $ cited_by     <int> 22850, 13774, 4009, 3109, 2906, 2844,
 ↪  1254, 1448
#> $ has_abstract <lgl> FALSE, TRUE, FALSE, TRUE, FALSE, FALSE,
 ↪  TRUE, TRUE
#> $ type         <chr> "article", "article", "article",
 ↪  "article", "article", "a~
```

4.4.3 Fetching from Crossref

```
cr_results <- cr_works(dois = dois)

cr_meta <- cr_results$data |>
  transmute(
    doi = doi,
    source = "Crossref",
    title = title,
    year = as.integer(
      coalesce(
        substr(issued, 1, 4),
        substr(deposited, 1, 4)
      )
    ),
    cited_by = as.integer(is.referenced.by.count),
    has_abstract = !is.na(abstract),
    type = type
  )

glimpse(cr_meta)
```

```
#> Rows: 7
#> Columns: 7
#> $ doi          <chr> "10.1038/520429a",
 ↪  "10.1073/pnas.0507655102", "10.1007/s1~
#> $ source       <chr> "Crossref", "Crossref", "Crossref",
 ↪  "Crossref", "Crossref~
#> $ title        <chr> "Bibliometrics: The Leiden Manifesto for
 ↪  research metrics~
#> $ year         <int> 2015, 2005, 2006, 2017, 1955, 2021, 2021
#> $ cited_by     <int> 2328, 8968, 1935, 14950, 1852, 839, 1976
#> $ has_abstract <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, TRUE,
 ↪  FALSE
```

```
#> $ type         <chr> "journal-article", "journal-article",
 ↪  "journal-article", ~
```

4.4.4 Comparing metadata completeness

```
comparison <- bind_rows(oa_meta, cr_meta) |>
  mutate(short_doi = str_extract(doi, "[^/]+/[^/]+$"))

comparison_wide <- comparison |>
  select(short_doi, source, year, cited_by, has_abstract, type)
   ↪  |>
  pivot_wider(
    names_from = source,
    values_from = c(year, cited_by, has_abstract, type),
    names_glue = "{source}_{.value}"
  )

comparison_wide |>
  gt() |>
  tab_header(
    title = "Metadata comparison: OpenAlex vs. Crossref",
    subtitle = "Same DOIs queried from both sources"
  ) |>
  cols_label(
    short_doi = "DOI (short)",
    OpenAlex_year = "OA Year",
    Crossref_year = "CR Year",
    OpenAlex_cited_by = "OA Citations",
    Crossref_cited_by = "CR Citations",
    OpenAlex_has_abstract = "OA Abstract?",
    Crossref_has_abstract = "CR Abstract?",
    OpenAlex_type = "OA Type",
    Crossref_type = "CR Type"
  )
```

Metadata comparison: OpenAlex vs. Crossref

Same DOIs queried from both sources

DOI (short)	OA Year	CR Year	OA Citations	CR Citations	OA Abstract?	CR Abstract?	OA Type	CR Type
10.1016/j.joi.2017.08.007	2017	2017	22850	14950	FALSE	FALSE	article	journal-article

10.1073/pnas.0507655102	2005	2005	13774	8968	TRUE	TRUE	article	journal-article
10.1038/520429a	2015	2015	4009	2328	FALSE	FALSE	article	journal-article
10.1126/science.122.3159.108	1955	1955	3109	1852	TRUE	FALSE	article	journal-article
10.1007/s11192-006-0144-7	2006	2006	2906	1935	FALSE	FALSE	article	journal-article
10.1007/s11192-021-03948-5	2021	2021	2844	1976	FALSE	FALSE	article	journal-article
10.1162/qss_a_00112	2021	2021	1254	839	TRUE	TRUE	article	journal-article
10.48550/arxiv.2205.01833	2022	NA	1448	NA	TRUE	NA	preprint	NA

4.4.5 Visualization: citation count comparison

```
cite_compare <- comparison |>
  select(short_doi, source, cited_by) |>
  pivot_wider(names_from = source, values_from = cited_by,
   ↪  values_fn = first)

if (nrow(cite_compare) > 0 && all(c("OpenAlex", "Crossref") %in%
 ↪  names(cite_compare))) {
  max_cite <- max(c(cite_compare$OpenAlex,
 ↪  cite_compare$Crossref), na.rm = TRUE)

  ggplot(cite_compare, aes(x = Crossref, y = OpenAlex)) +
    geom_abline(slope = 1, intercept = 0, linetype = "dashed",
     ↪  colour = "grey50") +
    geom_point(size = 3, colour = palette_sci(1)) +
    geom_text(
      aes(label = short_doi),
      size = 2.5, nudge_y = max_cite * 0.03, check_overlap = TRUE
    ) +
    scale_x_continuous(labels = scales::comma_format()) +
    scale_y_continuous(labels = scales::comma_format()) +
```

```
    coord_equal(xlim = c(0, max_cite * 1.1), ylim = c(0,
     ↪  max_cite * 1.1)) +
    labs(x = "Crossref citation count", y = "OpenAlex citation
     ↪  count") +
    theme_sci()
}
```

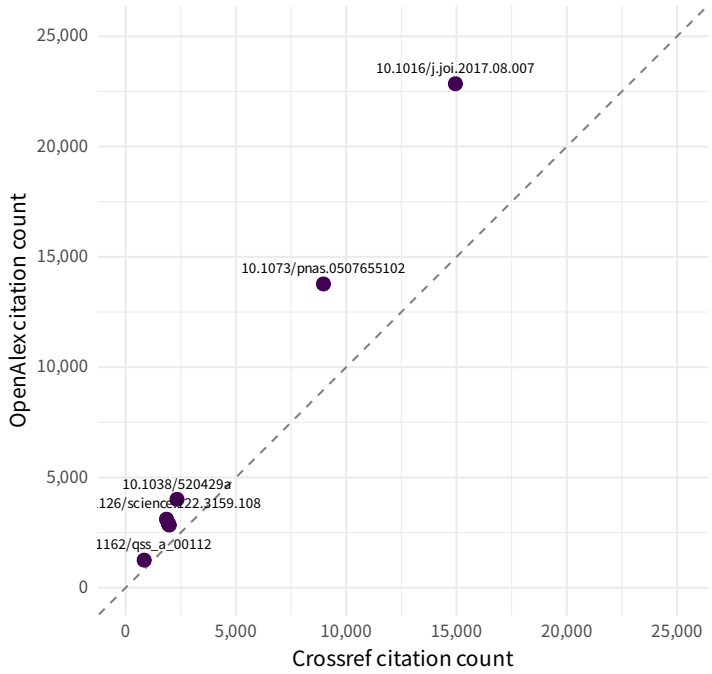


Figure 4.1: Citation counts reported by OpenAlex vs. Crossref for the same set of DOIs. Points above the diagonal indicate higher counts in OpenAlex.

4.4.6 Summary comparison table

```
tribble(
  ~Source, ~Coverage, ~`Access model`, ~API, ~`Citation data`,
   ↪  ~Strengths,
  "Web of Science", "~90M records, 21K+ journals",
   ↪  "Subscription", "Yes (paid)", "Yes (complete)", "Longest
   ↪  history, curated selection, JCR",
  "Scopus", "~94M records, 27K+ journals", "Subscription", "Yes
   ↪  (paid)", "Yes (complete)", "Broad journal coverage,
   ↪  CiteScore/SJR/SNIP",
```

```
  "Dimensions", "~140M records", "Freemium", "Yes (free/paid)",
   ↪  "Yes", "Links pubs to grants, patents, clinical trials",
  "OpenAlex", "~250M records", "Fully open", "Yes (free)",
   ↪  "Yes", "Open, broad coverage, author disambiguation",
  "The Lens", "~300M scholarly + 160M patent", "Free
   ↪  (institutional)", "Yes (free)", "Partial", "Patent-paper
   ↪  linkage",
  "Crossref", "~150M records", "Open", "Yes (free)", "Reference
   ↪  counts only", "DOI registry, reference lists, licensing",
  "PubMed", "~36M records", "Open", "Yes (free)", "No (use with
   ↪  others)", "MeSH indexing, biomedical depth",
  "Google Scholar", "Broadest (est. 400M+)", "Free (no bulk)",
   ↪  "No", "Yes (no export)", "Discovery, grey literature",
  "Semantic Scholar", "~200M records", "Free", "Yes (free)",
   ↪  "Yes", "AI-extracted entities, citation context",
  "NASA/ADS", "~16M records", "Open", "Yes (free)", "Yes", "Gold
   ↪  standard for astronomy/astrophysics",
  "arXiv", "~2.5M preprints", "Open", "Yes (free)", "No",
   ↪  "Preprints in physics, math, CS",
  "bioRxiv/medRxiv", "~300K preprints", "Open", "Yes (free)",
   ↪  "No", "Biology and health sciences preprints"
) |>
  gt() |>
  tab_header(
    title = "Twelve bibliographic data sources compared"
  ) |>
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) |>
  cols_width(
    Source ~ px(130),
    Coverage ~ px(170),
    `Access model` ~ px(100),
    API ~ px(100),
    `Citation data` ~ px(120),
    Strengths ~ px(250)
  )
```

Twelve bibliographic data sources compared

Source	Coverage	Access model	API	Citation data	Strengths
Web of Science	~90M records, 21K+ journals	Subscription	Yes (paid)	Yes (complete)	Longest history, curated selection, JCR
Scopus	~94M records, 27K+ journals	Subscription	Yes (paid)	Yes (complete)	Broad journal coverage, CiteScore/SJR/SNIP

Dimensions	~140M records	Freemium	Yes (free/paid)	Yes	Links pubs to grants, patents, clinical trials
OpenAlex	~250M records	Fully open	Yes (free)	Yes	Open, broad coverage, author disambiguation
The Lens	~300M scholarly + 160M patent	Free (institutional)	Yes (free)	Partial	Patent-paper linkage
Crossref	~150M records	Open	Yes (free)	Reference counts only	DOI registry, reference lists, licensing
PubMed	~36M records	Open	Yes (free)	No (use with others)	MeSH indexing, biomedical depth
Google Scholar	Broadest (est. 400M+)	Free (no bulk)	No	Yes (no export)	Discovery, grey literature
Semantic Scholar	~200M records	Free	Yes (free)	Yes	AI-extracted entities, citation context
NASA/ADS	~16M records	Open	Yes (free)	Yes	Gold standard for astronomy/astrophysics
arXiv	~2.5M preprints	Open	Yes (free)	No	Preprints in physics, math, CS
bioRxiv/medRxiv	~300K preprints	Open	Yes (free)	No	Biology and health sciences preprints

4.5 Diagnostics and interpretation

When comparing metadata across sources, consider the following diagnostic checks:

- **DOI match rate.** Not all DOIs may resolve in every database. Check how many of your queried DOIs returned results from each source. If a DOI is missing from one source, this reveals a coverage gap.
- **Citation count discrepancies.** Citation counts will almost always differ between sources because each database counts citations over a different corpus. OpenAlex typically reports higher counts than Crossref because OpenAlex tracks citations across its full 250M-record corpus, whereas Crossref only counts “is-referenced-by” links deposited by publishers.
- **Abstract availability.** Crossref abstracts depend on publisher deposits and are often absent, especially for older publications. OpenAlex aggregates abstracts from multiple sources and generally has better coverage.
- **Publication year.** Minor discrepancies (e.g., online-first vs. print dates) are common. Check whether your analysis is sensitive to the exact year assigned.

- **Type classification.** Crossref and OpenAlex use different type taxonomies. A record classified as "journal-article" in Crossref might appear as "article" in OpenAlex. Harmonise types before analysis.

The key takeaway is that no single source is complete or definitive. Cross-validation — fetching the same records from multiple sources — is a hallmark of rigorous bibliometric practice.

4.6 Limitations and responsible use

4.7 Limitations and responsible use

- **No database is complete.** Every source has coverage gaps — by language, discipline, publication type, or geography. Analyses based on a single source will systematically miss some literature. Always report which source was used and its known biases (Hicks et al., 2015).
- **Coverage does not equal quality.** A larger database is not necessarily better. Google Scholar has the broadest coverage but also the highest noise level. Curated databases like WoS and PubMed trade coverage for precision.
- **Proprietary sources limit reproducibility.** Analyses based on Web of Science or Scopus cannot be independently reproduced by researchers without institutional access. For fully reproducible work, prefer open sources like OpenAlex, Crossref, or PubMed (Priem et al., 2022).
- **Metadata quality is variable.** Even within a single source, metadata quality varies by publisher, discipline, and publication date. Older records often have sparser metadata. Always inspect and clean your data before analysis.
- **Citation counts are not comparable across sources.** Because each database counts citations over a different corpus, citation counts from OpenAlex, Scopus, and WoS are not interchangeable. Never mix citation counts from different sources in a single analysis without explicit justification and documentation (Visser et al., 2021).

4.8 Common pitfalls

4.9 Common pitfalls

- **Assuming one source is enough.** Researchers often default to a single database (typically WoS or Scopus) without evaluating whether it covers their field adequately. Social sciences, humanities, and non-English literatures are systematically underrepresented in WoS (Singh et al., 2021).
- **Comparing citation counts across databases.** Stating that a paper has "500 citations" is meaningless without specifying the source. Different databases will report different numbers for the same paper.
- **Ignoring access restrictions in reproducibility claims.** If your analysis uses Scopus data, other researchers need Scopus access to reproduce it. State this explicitly and consider providing OpenAlex-based alternatives.
- **Treating Google Scholar as a systematic source.** Google Scholar has no API, no structured export, and no quality control on what it indexes. It is useful for discovery but not for systematic bibliometric analysis.
- **Conflating preprints and peer-reviewed articles.** Sources like arXiv and bioRxiv index preprints that may later be published in journals. If your corpus includes both preprints and published versions without deduplication, you will double-count some works.

4.10 Exercises

1. **Coverage gap analysis.** Select 20 DOIs from a field of your choice. Query them in OpenAlex and Crossref. How many DOIs are found in each? Which source has better abstract coverage? Summarise your findings in a table.

2. **Disciplinary bias.** Use OpenAlex to compare the number of indexed works in two contrasting fields (e.g., molecular biology vs. philosophy). Normalise by the estimated number of active researchers in each field. What does this tell you about coverage bias?

3. **Citation count concordance.** For a set of at least 50 works, retrieve citation counts from both OpenAlex and Crossref. Compute the Pearson and Spearman correlations between the two sets of counts. Plot the results. Are the two sources more concordant for highly cited or lowly cited papers?

4. **Preprint tracking.** Select 10 recent bioRxiv DOIs and check whether they appear in OpenAlex. For those that do, check whether OpenAlex links them to a published journal version. What proportion of your preprints have been published?
5. **Source selection exercise.** A colleague asks you to conduct a bibliometric analysis of nursing research in Sub-Saharan Africa. Which data source(s) would you recommend and why? Write a brief justification considering coverage, cost, and reproducibility.

4.11 Solutions

Solutions are provided in E.

4.12 Further reading

- Visser et al. (2021) — Large-scale comparison of Scopus, Web of Science, Dimensions, Crossref, and Microsoft Academic. Essential reading for understanding how source choice affects results.
- Singh et al. (2021) — Comparative analysis of journal coverage across WoS, Scopus, and Dimensions. Quantifies disciplinary differences in database coverage.
- Priem et al. (2022) — The OpenAlex paper. Describes the data model, coverage, and design philosophy of the open alternative to proprietary databases.
- Garfield (1955) — The original paper proposing citation indexing. Historical context for understanding why these databases exist.
- Hicks et al. (2015) — The Leiden Manifesto. Principle 4 (transparency) and Principle 6 (field variation) are directly relevant to data source selection.
- Waltman (2016) — Review of citation impact indicators. Discusses how indicator properties interact with database coverage.

4.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
```

```
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] rcrossref_1.2.1 gt_1.4.0        tidytext_0.4.3
 ↪  glue_1.8.1
#>  [5] openalexR_3.1.2 lubridate_1.9.5 forcats_1.0.1
 ↪  stringr_1.6.0
#>  [9] dplyr_1.2.1     purrr_1.2.2     readr_2.2.0
 ↪  tidyr_1.3.2
#> [13] tibble_3.3.1    ggplot2_4.0.4   tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gtable_0.3.6       xfun_0.62          htmlwidgets_1.6.4
 ↪  lattice_0.22-9
#>  [5] tzdb_0.5.0         vctrs_0.7.3        tools_4.6.1
 ↪  generics_0.1.4
#>  [9] curl_8.1.0         janeaustenr_1.0.0  pkgconfig_2.0.3
 ↪  tokenizers_0.3.0
#> [13] Matrix_1.7-5       RColorBrewer_1.1-3 S7_1.0.0
 ↪  lifecycle_1.0.5
#> [17] compiler_4.6.1     farver_2.1.2       httpuv_1.6.17
 ↪  htmltools_0.5.9
#> [21] SnowballC_0.7.1    yaml_2.3.12        later_1.4.8
 ↪  pillar_1.11.1
#> [25] DT_0.34.0          viridis_0.6.5      mime_0.13
 ↪  stopwords_2.3
```

```
#> [29] tidyselect_1.2.1   digest_0.6.39      stringi_1.8.9
 ↪  bookdown_0.48
#> [33] labeling_0.4.3     rprojroot_2.1.1    fastmap_1.2.0
 ↪  grid_4.6.1
#> [37] here_1.0.2         cli_3.6.6          magrittr_2.0.5
 ↪  triebeard_0.4.1
#> [41] crul_1.6.0         utf8_1.2.6         withr_3.0.3
 ↪  promises_1.5.0
#> [45] scales_1.4.0       timechange_0.4.0   rmarkdown_2.32
 ↪  httr_1.4.9
#> [49] otel_0.3.0         gridExtra_2.3.1    hms_1.1.4
 ↪  shiny_1.15.0
#> [53] evaluate_1.0.5     knitr_1.52         miniUI_0.1.2
 ↪  viridisLite_0.4.3
#> [57] urltools_1.7.3.1   rlang_1.3.0        Rcpp_1.1.2
 ↪  httpcode_0.3.0
#> [61] xtable_1.8-8       xml2_1.6.0         rstudioapi_0.19.0
 ↪  jsonlite_2.0.0
#> [65] plyr_1.8.9         R6_2.6.1           systemfonts_1.3.4
 ↪  fs_2.1.0
```

Part II

Data Acquisition in R

Chapter 5

APIs and Packages

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

5.1 Learning objectives

After completing this chapter, you will be able to:

- Query the OpenAlex API using `openalexR` to retrieve works, authors, institutions, and concepts
- Apply filters, pagination, and sampling to control the size and scope of API results
- Deduplicate records by DOI and clean author and affiliation metadata
- Cache API responses locally so repeated book builds do not re-query the API
- Use `rcrossref` and `pubmedR` as complementary data sources

5.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
```

```
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

5.3 Conceptual background

Bibliometric research begins with data. The quality and completeness of a bibliographic corpus determines the validity of every downstream analysis — from citation counts to network maps to topic models.

Historically, Web of Science and Scopus have dominated as data sources for scientometric research. Both require institutional subscriptions and impose restrictions on bulk data use. The launch of **OpenAlex** in 2022 changed this landscape fundamentally (Priem et al., 2022). OpenAlex is a fully open index of over 250 million scholarly works, built from Crossref, PubMed, institutional repositories, and other open sources. It provides a free REST API with no authentication required, making it the ideal backbone for reproducible bibliometric research.

OpenAlex organises its data around five entity types: **works** (publications), **authors**, **sources** (journals, repositories), **institutions**, and **topics** (hierarchical subject tags). Each entity has a unique OpenAlex ID and can be queried via filters on metadata fields such as publication date, citation count, open-access status, and institutional affiliation.

The `openalexR` package provides a tidy R interface to the OpenAlex API. It handles pagination, rate limiting, and result parsing, returning clean tibbles ready for analysis. For metadata not covered by OpenAlex — such as reference lists and funding information — `rcrossref` provides access to Crossref's extensive metadata registry, and `pubmedR` covers the biomedical literature indexed in PubMed/MEDLINE.

5.4 Worked example

We build a small corpus of recent scientometrics research, demonstrating the full acquisition-to-clean-data pipeline.

5.4.1 Querying works from OpenAlex

The provider function is `openalexR::oa_fetch()`. Runnable examples use the book's `fetch_openalex()` wrapper, sourced above, to cache responses by their exact arguments. Set `SCIENTOMETRICS_OFFLINE=1`

to reuse existing OpenAlex caches without requesting new data; a missing cache stops with an explicit error. Other providers have their own network requirements. We search for works published in the journal Scientometrics between 2020 and 2023, sampling 200 records for a manageable demonstration.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2020-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 200, seed = 42)
)

glimpse(works)
```

```
#> Rows: 200
#> Columns: 45
#> $ id                            <chr>
 ↪  "https://openalex.org/W3036497540", "htt~
#> $ title                         <chr> "The world-systemic
 ↪  network of global el~
#> $ display_name                  <chr> "The world-systemic
 ↪  network of global el~
#> $ authorships                   <list> [<tbl_df[2 x 8]>],
 ↪  [<tbl_df[6 x 8]>], [~
#> $ abstract                      <chr> "In this paper, we
 ↪  analyze the education~
#> $ doi                           <chr>
 ↪  "https://doi.org/10.1007/s11192-020-0356~
#> $ publication_date             <date> 2020-06-17, 2022-12-17,
 ↪  2022-11-30, 202~
#> $ publication_year              <int> 2020, 2022, 2022, 2023,
 ↪  2021, 2021, 2021~
#> $ relevance_score               <dbl> 1.000, 1.000, 0.999,
 ↪  0.999, 0.998, 0.998~
#> $ fwci                          <dbl> 5.114, 2.485, 0.274,
 ↪  49.658, 1.440, 3.19~
#> $ cited_by_count                <int> 40, 64, 1, 199, 10, 19,
 ↪  9, 31, 15, 50, 1~
#> $ counts_by_year                <list> [<data.frame[7 x 2]>],
 ↪  [<data.frame[5 x~
#> $ ids                           <list>
 ↪  <"https://openalex.org/W3036497540", "h~
#> $ type                          <chr> "article", "article",
 ↪  "article", "articl~
#> $ is_oa                         <lgl> TRUE, FALSE, FALSE,
 ↪  TRUE, FALSE, FALSE, ~
#> $ is_oa_anywhere                <lgl> TRUE, TRUE, TRUE, TRUE,
 ↪  FALSE, FALSE, TR~
```

```
#> $ oa_status                     <chr> "hybrid", "green",
 ↪  "green", "hybrid", "c~
#> $ oa_url                        <chr>
 ↪  "https://link.springer.com/content/pdf/1~
#> $ any_repository_has_fulltext   <lgl> TRUE, TRUE, TRUE, TRUE,
 ↪  FALSE, FALSE, TR~
#> $ source_display_name           <chr> "Scientometrics",
 ↪  "Scientometrics", "Sci~
#> $ source_id                     <chr>
 ↪  "https://openalex.org/S148561398", "http~
#> $ issn_l                        <chr> "0138-9130",
 ↪  "0138-9130", "0138-9130", "~
#> $ host_organization             <chr>
 ↪  "https://openalex.org/P4310320108", "htt~
#> $ host_organization_name        <chr> "Springer Nature
 ↪  (Netherlands)", "Spring~
#> $ landing_page_url              <chr>
 ↪  "https://doi.org/10.1007/s11192-020-0356~
#> $ pdf_url                       <chr>
 ↪  "https://link.springer.com/content/pdf/1~
#> $ license                       <chr> "cc-by", NA, NA,
 ↪  "other-oa", NA, NA, NA,~
#> $ version                       <chr> "publishedVersion",
 ↪  "publishedVersion", ~
#> $ referenced_works              <list>
 ↪  <"https://openalex.org/W308615189", "ht~
#> $ referenced_works_count        <int> 84, 45, 4, 54, 44, 58,
 ↪  7, 24, 64, 97, 56~
#> $ related_works                 <list>
 ↪  <"https://openalex.org/W3107920103", "h~
#> $ concepts                     <list> [<data.frame[18 x 5]>],
 ↪  [<data.frame[20~
#> $ topics                        <list> [<tbl_df[12 x 5]>],
 ↪  [<tbl_df[12 x 5]>],~
#> $ keywords                      <list> [<data.frame[8 x 3]>],
 ↪  [<data.frame[6 x~
#> $ is_paratext                   <lgl> FALSE, FALSE, FALSE,
 ↪  FALSE, FALSE, FALSE~
#> $ is_retracted                  <lgl> FALSE, FALSE, FALSE,
 ↪  FALSE, FALSE, FALSE~
#> $ language                      <chr> "en", "en", "en", "en",
 ↪  "en", "en", "en"~
#> $ sustainable_development_goals <list> [<data.frame[2 x 3]>],
 ↪  NA, NA, NA, NA, ~
#> $ awards                        <list> NA,
 ↪  <"https://openalex.org/G63352921", ~
#> $ funders                       <list> [<data.frame[1 x 3]>],
 ↪  [<data.frame[4 x~
#> $ apc                           <list> [<data.frame[2 x 5]>],
 ↪  [<data.frame[2 x~
```

```
#> $ first_page                    <chr> "2469", "1241", "2029",
 ↪  "4007", "3209", ~
#> $ last_page                     <chr> "2495", "1263", "2030",
 ↪  "4028", "3226", ~
#> $ volume                        <chr> "124", "128", "128",
 ↪  "128", "126", "126"~
#> $ issue                        <chr> "3", "2", "3", "7", "4",
 ↪  "7", "10", "1",~
```

The result is a tibble where each row is a work and columns include `id`, `display_name`, `publication_date`, `cited_by_count`, `doi`, and nested list-columns for authors and concepts.

5.4.2 Inspecting the results

```
works |>
  select(display_name, publication_date, cited_by_count, doi) |>
  arrange(desc(cited_by_count)) |>
  head(10)
```

```
#> # A tibble: 10 x 4
#>    display_name                              publication_date
 ↪  cited_by_count doi
#>    <chr>                                     <date>
 ↪  <int> <chr>
#>  1 Google Scholar, Microsoft Academic, Sc~ 2020-09-21
 ↪  1658 http~
#>  2 Sample size in bibliometric analysis    2020-07-31
 ↪  281 http~
#>  3 Rethinking SME default prediction: a s~ 2021-01-29
 ↪  212 http~
#>  4 Visibility, impact, and applications o~ 2023-06-01
 ↪  199 http~
#>  5 Preliminary analysis of COVID-19 acade~ 2020-06-25
 ↪  168 http~
#>  6 International collaboration during the~ 2021-02-14
 ↪  129 http~
#>  7 Why do papers from international colla~ 2022-08-22
 ↪  126 http~
#>  8 Coronavirus mapping in scientific publ~ 2020-07-01
 ↪  124 http~
#>  9 Evolution of research topics in LIS be~ 2020-10-10
 ↪  112 http~
#> 10 A deep learning based method for extra~ 2020-07-24
 ↪  105 http~
```

5.4.3 Querying authors

We can also retrieve author-level data. Let us look up authors associated with scientometrics research by filtering on the relevant topic.

```
authors <- fetch_openalex(
  entity = "authors",
  topics.id = "T10102",
  works_count = ">50",
  options = list(sample = 50, seed = 42)
)

authors |>
  select(display_name, works_count, cited_by_count) |>
  arrange(desc(cited_by_count)) |>
  head(10)
```

```
#> # A tibble: 10 x 3
#>    display_name              works_count cited_by_count
#>    <chr>                           <int>          <int>
#>  1 Jennifer Tetzlaff                  80         603386
#>  2 Steve Lawrence                    144          26565
#>  3 Alejandro Valencia-Arías          408           7081
#>  4 Ian Rowlands                      168           5854
#>  5 Alfredo Alcayde                   174           5291
#>  6 Wei Hua Zhou                      284           4151
#>  7 Mathieu Albert                     63           3563
#>  8 Iain Hrynaszkiewicz               144           2061
#>  9 Khaled Moustafa                   156           2048
#> 10 Sai Leung Ng                       70           1923
```

5.4.4 Deduplication by DOI

Real-world corpora often contain duplicates when merging results from multiple queries or sources. The `dedupe_by_doi()` function from our companion package handles this cleanly.

```
works_with_dupes <- bind_rows(works, works[1:10, ])
cat(glue("Before dedup: {nrow(works_with_dupes)} rows\n\n"))
```

```
#> Before dedup: 210 rows
```

```
works_clean <- dedupe_by_doi(works_with_dupes)
cat(glue("After dedup:  {nrow(works_clean)} rows\n"))
```

```
#> After dedup:  200 rows
```

5.4.5 Caching for reproducible builds

The `fetch_openalex()` wrapper in `R/api_helpers.R` automatically caches API responses to disk. On subsequent calls with the same arguments, it reads from cache instead of hitting the API.

```
works_cached <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2023-01-01",
  to_publication_date = "2023-06-30",
  options = list(sample = 50, seed = 42),
  cache_dir = "_freeze/openalex_cache",
  cache_days = 30
)

nrow(works_cached)
```

```
#> [1] 50
```

5.4.6 Complementary sources: Crossref

For detailed metadata (reference lists, funders, licenses), Crossref is invaluable.

```
# Requires network access — cached output used in book builds
cr_work <- rcrossref::cr_works(doi =
 ↪  "10.1007/s11192-017-2300-7")
cr_work$data |>
  select(title, container.title, deposited,
   ↪  is.referenced.by.count)
```

5.4.7 Visualization

```
works |>
  mutate(year = year(publication_date)) |>
  count(year) |>
  ggplot(aes(x = year, y = n)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Publication year", y = "Number of works") +
  theme_sci()
```

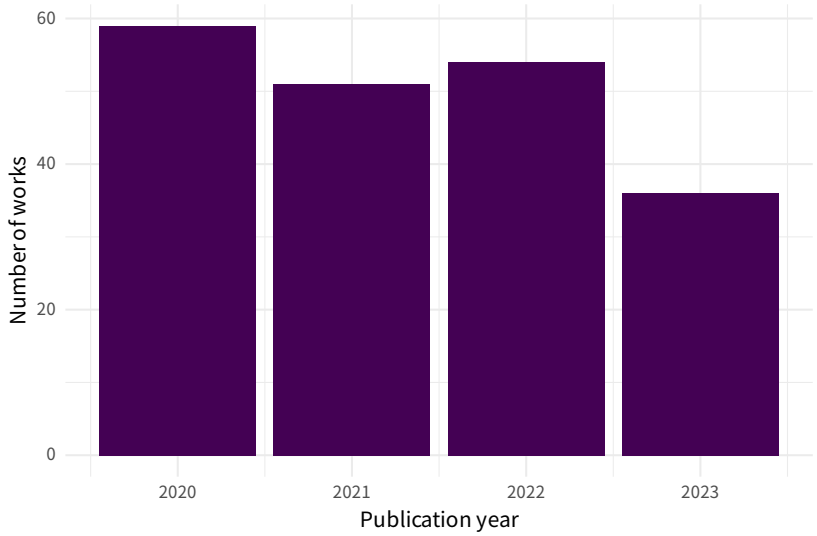


Figure 5.1: Annual publication counts for sampled Scientometrics articles (2020–2023).

5.5 Diagnostics and interpretation

When working with API data, always verify:

- **Completeness**: Compare your result count against the API's reported `meta$count` to confirm you received all expected records. With sampling, the count reflects the sample, not the full set.
- **Coverage dates**: Check `min(works$publication_date)` and `max(works$publication_date)` to confirm your date filters were applied correctly.
- **DOI presence**: Not all works have DOIs. Check `mean(!is.na(works$doi))` to understand the DOI coverage rate in your corpus.
- **Nested columns**: Author and concept columns are list-columns. Use `tidyr::unnest()` carefully and inspect for `NULL` entries before unnesting.

```
cat(glue("Total works: {nrow(works)}\n"))
```

```
#> Total works: 200
```

```
cat(glue("DOI coverage:
 ↪  {scales::percent(mean(!is.na(works$doi)))}\n"))
```

```
#> DOI coverage: 100%
```

```
cat(glue("Date range: {min(works$publication_date)} to
 ↪  {max(works$publication_date)}\n"))
```

```
#> Date range: 2020-01-11 to 2023-12-18
```

5.6 Limitations and responsible use

5.7 Limitations and responsible use

- **Coverage bias**: OpenAlex is strongest in English-language, peer-reviewed journal articles. Conference proceedings, books, and non-English literature may be underrepresented. Always report the data source and its known coverage limitations.
- **Metadata quality**: Author names and affiliations are parsed algorithmically and may contain errors, especially for names with diacritics or non-Latin scripts.
- **Temporal lag**: OpenAlex updates continuously but may lag behind primary databases by days or weeks. Very recent publications may be missing.
- **API rate limits**: OpenAlex allows 10 requests per second for unauthenticated users (100/s with a polite pool email). Respect these limits to avoid being throttled.
- **Reproducibility**: API results can change as OpenAlex updates its data. Always cache results and document the date of data collection for reproducibility (Hicks et al., 2015).

5.8 Common pitfalls

5.9 Common pitfalls

- **Not caching**: Running `oa_fetch()` directly in code chunks without caching means every `quarto render` hits the API, producing different results each time and risking rate limits.

- **Ignoring pagination**: `oa_fetch()` handles pagination automatically, but very large queries (>10,000 works) can time out. Use filters to narrow your query or use cursor-based pagination.
- **Confusing sampling with filtering**: `options = list(sample = 200)` returns a random sample from the full result set, not the first 200 records. This is useful for demonstrations but not for exhaustive analyses.
- **Treating OpenAlex IDs as permanent**: While generally stable, OpenAlex IDs can change when entities are merged. Use DOIs as the primary identifier for works.
- **Merging sources without deduplication**: When combining results from OpenAlex and Crossref, records may overlap. Always deduplicate by DOI before analysis.

5.10 Exercises

1. **Fetch works by institution.** Use `oa_fetch()` to retrieve works affiliated with your own institution. How many works does OpenAlex index for it? What is the DOI coverage rate? (Hint: find your institution's OpenAlex ID via *`oa_fetch(entity = "institutions", search = "...")`*)
2. **Compare Crossref metadata.** Pick a DOI from your fetched works and retrieve its metadata from Crossref using `rcrossref::cr_works()`. Compare the citation count reported by OpenAlex vs. Crossref. Why might they differ?
3. **Build a multi-year corpus.** Fetch all works from a journal of your choice for 2015–2023 (without sampling). Plot the annual publication count and the median citation count per year. What trends do you observe?

5.11 Solutions

Solutions are provided in E.

5.12 Further reading

- Priem et al. (2022) — The OpenAlex paper; describes the data model, coverage, and API design.

- Aria and Cuccurullo (2017) — The `bibliometrix` package; an alternative high-level interface for bibliometric data acquisition and analysis.
- Garfield (1955) — The foundational paper on citation indexing, motivating why these databases exist.

5.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] rcrossref_1.2.1 gt_1.4.0        tidytext_0.4.3
 ↪  glue_1.8.1
#>  [5] openalexR_3.1.2 lubridate_1.9.5 forcats_1.0.1
 ↪  stringr_1.6.0
#>  [9] dplyr_1.2.1     purrr_1.2.2     readr_2.2.0
 ↪  tidyr_1.3.2
#> [13] tibble_3.3.1    ggplot2_4.0.4   tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
```

```
#>  [1] gtable_0.3.6       xfun_0.62          htmlwidgets_1.6.4
 ↪  lattice_0.22-9
#>  [5] tzdb_0.5.0         vctrs_0.7.3        tools_4.6.1
 ↪  generics_0.1.4
#>  [9] curl_8.1.0         janeaustenr_1.0.0  pkgconfig_2.0.3
 ↪  tokenizers_0.3.0
#> [13] Matrix_1.7-5       RColorBrewer_1.1-3 S7_1.0.0
 ↪  lifecycle_1.0.5
#> [17] compiler_4.6.1     farver_2.1.2       httpuv_1.6.17
 ↪  htmltools_0.5.9
#> [21] SnowballC_0.7.1    yaml_2.3.12        later_1.4.8
 ↪  pillar_1.11.1
#> [25] DT_0.34.0          viridis_0.6.5      mime_0.13
 ↪  stopwords_2.3
#> [29] tidyselect_1.2.1   digest_0.6.39      stringi_1.8.9
 ↪  bookdown_0.48
#> [33] labeling_0.4.3     rprojroot_2.1.1    fastmap_1.2.0
 ↪  grid_4.6.1
#> [37] here_1.0.2         cli_3.6.6          magrittr_2.0.5
 ↪  triebeard_0.4.1
#> [41] crul_1.6.0         utf8_1.2.6         withr_3.0.3
 ↪  promises_1.5.0
#> [45] scales_1.4.0       timechange_0.4.0   rmarkdown_2.32
 ↪  httr_1.4.9
#> [49] otel_0.3.0         gridExtra_2.3.1    hms_1.1.4
 ↪  shiny_1.15.0
#> [53] evaluate_1.0.5     knitr_1.52         miniUI_0.1.2
 ↪  viridisLite_0.4.3
#> [57] urltools_1.7.3.1   rlang_1.3.0        Rcpp_1.1.2
 ↪  httpcode_0.3.0
#> [61] xtable_1.8-8       xml2_1.6.0         rstudioapi_0.19.0
 ↪  jsonlite_2.0.0
#> [65] plyr_1.8.9         R6_2.6.1           systemfonts_1.3.4
 ↪  fs_2.1.0
```

Chapter 6

Parsing Native Exports

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

6.1 Learning objectives

After completing this chapter, you will be able to:

- Parse BibTeX files using `bib2df` and `RefManageR`
- Import Web of Science plain-text and Scopus CSV exports via `bibliometrix`
- Standardize field names across formats into a unified schema
- Diagnose common parsing failures and missing fields

6.2 Setup

```
library(tidyverse)
library(bib2df)
library(RefManageR)
library(bibliometrix)
library(glue)

set.seed(20260509)

source(here::here("R", "sci_palette.R"))
```

6.3 Conceptual background

Not all bibliometric data comes from an API. Researchers frequently begin with files exported from database search interfaces: BibTeX (`.bib`) from Google Scholar or Zotero, RIS (`.ris`) from Scopus or PubMed, plain text from Web of Science, or CSV from Scopus. Each format encodes the same information — authors, title, year, journal, DOI — in different structures.

The `bibliometrix` package (Aria and Cuccurullo, 2017) provides `convert2df()`, a versatile function that reads multiple export formats and returns a standardized data frame. For BibTeX specifically, `bib2df` and `RefManageR` offer complementary parsing capabilities. The key challenge is not parsing per se but standardization: ensuring that author names, journal titles, and identifiers are represented consistently regardless of the source format.

6.4 Worked example

6.4.1 Parsing a BibTeX string

We construct a small BibTeX string inline to demonstrate parsing without external files.

```
bib_text <- '
@article{smith2023,
  author = {Smith, John A. and Doe, Jane B.},
  title = {Advances in Bibliometric Methods},
  journal = {Journal of Informetrics},
  year = {2023},
  volume = {17},
  pages = {101--115},
  doi = {10.1234/example.2023}
}
@article{chen2022,
  author = {Chen, Wei and Li, Xiao},
  title = {Citation Networks in Computer Science},
  journal = {Scientometrics},
  year = {2022},
  volume = {127},
  pages = {3201--3220},
  doi = {10.1234/example.2022}
}
'

tmp_bib <- tempfile(fileext = ".bib")
```

```
writeLines(bib_text, tmp_bib)

bib_df <- bib2df(tmp_bib)
glimpse(bib_df)
```

```
#> Rows: 2
#> Columns: 27
#> $ CATEGORY     <chr> "ARTICLE", "ARTICLE"
#> $ BIBTEXKEY    <chr> "smith2023", "chen2022"
#> $ ADDRESS      <chr> NA, NA
#> $ ANNOTE       <chr> NA, NA
#> $ AUTHOR       <list> <"Smith, John A.", "Doe, Jane B.">,
 ↪  <"Chen, Wei", "Li, Xi~
#> $ BOOKTITLE    <chr> NA, NA
#> $ CHAPTER      <chr> NA, NA
#> $ CROSSREF     <chr> NA, NA
#> $ EDITION      <chr> NA, NA
#> $ EDITOR       <list> NA, NA
#> $ HOWPUBLISHED <chr> NA, NA
#> $ INSTITUTION  <chr> NA, NA
#> $ JOURNAL      <chr> "Journal of Informetrics",
 ↪  "Scientometrics"
#> $ KEY          <chr> NA, NA
#> $ MONTH        <chr> NA, NA
#> $ NOTE         <chr> NA, NA
#> $ NUMBER       <chr> NA, NA
#> $ ORGANIZATION <chr> NA, NA
#> $ PAGES        <chr> "101--115", "3201--3220"
#> $ PUBLISHER    <chr> NA, NA
#> $ SCHOOL       <chr> NA, NA
#> $ SERIES       <chr> NA, NA
#> $ TITLE        <chr> "Advances in Bibliometric Methods",
 ↪  "Citation Networks in~
#> $ TYPE         <chr> NA, NA
#> $ VOLUME       <chr> "17", "127"
#> $ YEAR         <dbl> 2023, 2022
#> $ DOI          <chr> "10.1234/example.2023",
 ↪  "10.1234/example.2022"
```

6.4.2 Parsing with RefManageR

```
bib_rm <- ReadBib(tmp_bib)
as.data.frame(bib_rm) |> select(title, author, year, journal,
 ↪  doi)
```

```
#>                                          title
 ↪  author
```

```
#> smith2023      Advances in Bibliometric Methods John A. Smith
 ↪  and Jane B. Doe
#> chen2022  Citation Networks in Computer Science          Wei
 ↪  Chen and Xiao Li
#>           year                 journal                  doi
#> smith2023 2023 Journal of Informetrics 10.1234/example.2023
#> chen2022  2022          Scientometrics 10.1234/example.2022
```

6.4.3 Parsing WoS plain text with bibliometrix

```
# Requires a WoS export file — not bundled
# wos_df <- convert2df("savedrecs.txt", dbsource = "wos", format
 ↪  = "plaintext")
# glimpse(wos_df)
```

6.4.4 Standardizing field names

```
standardize_bib <- function(df) {
  df |>
    transmute(
      title = TITLE,
      authors = AUTHOR,
      year = as.integer(YEAR),
      journal = JOURNAL,
      doi = DOI
    )
}

bib_df |> standardize_bib()
```

```
#> # A tibble: 2 x 5
#>   title                                authors    year journal
 ↪  doi
#>   <chr>                                  <list>    <int> <chr>
 ↪  <chr>
#> 1 Advances in Bibliometric Methods      <chr [2]>  2023
 ↪  Journal of Inform~ 10.1~
#> 2 Citation Networks in Computer Science <chr [2]>  2022
 ↪  Scientometrics     10.1~
```

6.4.5 Visualization

```
completeness <- bib_df |>
  summarise(across(everything(), ~ mean(!is.na(.)))) |>
  pivot_longer(everything(), names_to = "field", values_to =
   ↪  "present") |>
  filter(present > 0) |>
  mutate(field = fct_reorder(field, present))

ggplot(completeness, aes(x = present, y = field)) +
  geom_col(fill = palette_sci(1)) +
  scale_x_continuous(labels = scales::percent) +
  labs(x = "Completeness", y = NULL) +
  theme_sci()
```

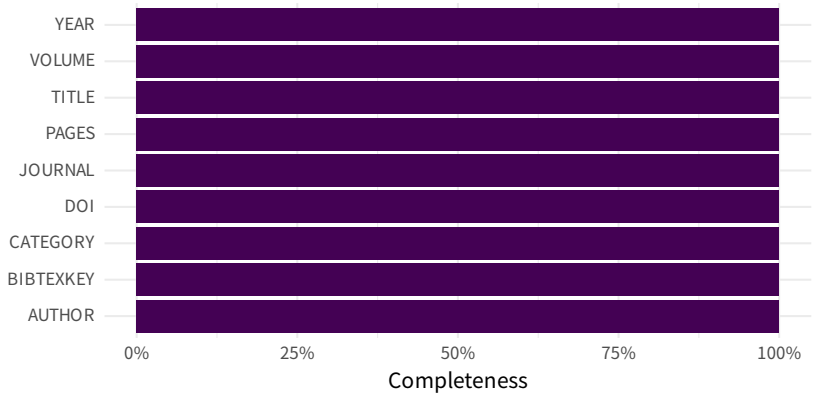


Figure 6.1: Field completeness in the parsed BibTeX records.

6.5 Diagnostics and interpretation

After parsing, always check:

- **Row count**: Does the number of records match the export? Missing entries may indicate parsing errors.
- **Field completeness**: Which fields are populated? Abstracts and keywords are often missing in BibTeX exports.
- **Author format**: Names may appear as “Last, First” or “First Last” depending on the source. Standardize before analysis.
- **Encoding**: Non-ASCII characters (diacritics, CJK) can corrupt during parsing. Ensure UTF-8 encoding.

6.6 Limitations and responsible use

6.7 Limitations and responsible use

- Export files are snapshots: they reflect the database state at export time and become stale immediately.
- Field mapping between formats is imperfect. Some formats lack fields (RIS has no standard abstract tag across all databases).
- Parsing errors are silent — always validate record counts and spot-check metadata against the source (Hicks et al., 2015).

6.8 Common pitfalls

6.9 Common pitfalls

- **Encoding mismatches**: Exporting as Latin-1 and reading as UTF-8 (or vice versa) corrupts names and titles.
- **Truncated exports**: Many databases cap exports at 500 or 1,000 records per file. Large corpora require multiple export batches.
- **Inconsistent author delimiters**: BibTeX uses "and", RIS uses newlines, CSV uses semicolons. Parsing must handle each.
- **Assuming format from extension**: A `.txt` file may be WoS plain text or something else entirely. Check the first few lines.

6.10 Exercises

1. **Export and parse.** Export 50 records from Google Scholar as BibTeX. Parse them with `bib2df`. How many fields are populated?

2. **Format comparison.** Export the same set of records from Scopus as both CSV and RIS. Parse both and compare the fields available.

3. **Standardization function.** Write a function that takes a data frame from `convert2df()` and returns a tibble with columns: `doi`, `title`, `authors`, `year`, `journal`, `cited_by`.

6.11 Solutions

Solutions are provided in E.

6.12 Further reading

- Aria and Cuccurullo (2017) — The `bibliometrix` package, including `convert2df()` for multi-format parsing.
- Priem et al. (2022) — OpenAlex as an API-first alternative to file-based workflows.

6.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
```

```
#> other attached packages:
#>  [1] bibliometrix_5.5.0 RefManageR_1.4.0   bib2df_1.1.2.0
 ↪  rcrossref_1.2.1
#>  [5] gt_1.4.0           tidytext_0.4.3     glue_1.8.1
 ↪  openalexR_3.1.2
#>  [9] lubridate_1.9.5    forcats_1.0.1      stringr_1.6.0
 ↪  dplyr_1.2.1
#> [13] purrr_1.2.2        readr_2.2.0        tidyr_1.3.2
 ↪  tibble_3.3.1
#> [17] ggplot2_4.0.4      tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0           magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0        pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3        ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0        rmarkdown_2.32         tzdb_0.5.0
#> [19] xfun_0.62             jsonlite_2.0.0
 ↪  SnowballC_0.7.1
#> [22] later_1.4.8           parallel_4.6.1
 ↪  stopwords_2.3
#> [25] R6_2.6.1              stringi_1.8.9
 ↪  RColorBrewer_1.1-3
#> [28] cellranger_1.1.0      Rcpp_1.1.2
 ↪  bookdown_0.48
#> [31] knitr_1.52            triebeard_0.4.1
 ↪  base64enc_0.1-6
#> [34] httpuv_1.6.17         rentrez_1.2.4         Matrix_1.7-5
#> [37] igraph_2.3.4          timechange_0.4.0
 ↪  tidyselect_1.2.1
#> [40] stringdist_0.9.17     rstudioapi_0.19.0
 ↪  pubmedR_1.1.0
#> [43] yaml_2.3.12           viridis_0.6.5         miniUI_0.1.2
#> [46] humaniformat_0.6.0    curl_8.1.0              qpdf_1.4.1
#> [49] lattice_0.22-9        plyr_1.8.9            shiny_1.15.0
#> [52] withr_3.0.3           S7_1.0.0             askpass_1.2.1
#> [55] evaluate_1.0.5         zip_3.0.2              xml2_1.6.0
#> [58] shinycssloaders_1.1.0  pillar_1.11.1
 ↪  janeaustenr_1.0.0
#> [61] DT_0.34.0              plotly_4.12.1
 ↪  generics_0.1.4
#> [64] rprojroot_2.1.1       hms_1.1.4             scales_1.4.0
#> [67] xtable_1.8-8          contentanalysis_1.1.1  tools_4.6.1
#> [70] data.table_1.18.6.1    tokenizers_0.3.0
 ↪  openxlsx_4.2.9
#> [73] pdftools_3.9.1         fs_2.1.0
 ↪  visNetwork_2.1.4
#> [76] XML_3.99-0.25         grid_4.6.1            bibtex_0.5.3
```

```
#> [79] urltools_1.7.3.1       rscopus_0.9.0
 ↪  dimensionsR_0.0.3
#> [82] bibliometrixData_0.3.0 cli_3.6.6
 ↪  viridisLite_0.4.3
#> [85] gtable_0.3.6           digest_0.6.39
 ↪  ggrepel_0.9.8
#> [88] crul_1.6.0            htmlwidgets_1.6.4     farver_2.1.2
#> [91] htmltools_0.5.9        lifecycle_1.0.5        httr_1.4.9
#> [94] here_1.0.2             mime_0.13
```

Chapter 7

Building Reproducible Corpora

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

7.1 Learning objectives

After completing this chapter, you will be able to:

- Deduplicate bibliographic records by DOI and by approximate title matching
- Resolve identifiers across databases (OpenAlex ID, DOI, PMID)
- Recognize author name disambiguation challenges and apply basic heuristics
- Standardize institutional affiliations using ROR identifiers
- Save a clean corpus in an efficient format (Parquet) for downstream analysis

7.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(arrow)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

7.3 Conceptual background

Raw bibliographic data is messy. Records from different sources may describe the same paper with different metadata, the same author under different name variants, and the same institution under dozens of spelling variations. Building a clean, reproducible corpus requires systematic attention to three problems:

Deduplication. When merging data from multiple sources, the same publication may appear multiple times. DOI-based deduplication is the gold standard, but not all records have DOIs. Fuzzy title matching provides a fallback but is error-prone for short or generic titles.

Author disambiguation. "J. Smith" could be dozens of different people. OpenAlex uses machine-learning-based author clustering to assign persistent author IDs, but errors persist — especially for common names and authors who change institutions (Priem et al., 2022).

Affiliation standardization. Institutional names appear in countless variants ("MIT", "Massachusetts Institute of Technology", "Mass. Inst. Tech."). The Research Organization Registry (ROR) provides a curated set of persistent identifiers for research organizations. OpenAlex maps many affiliations to ROR IDs, but coverage is incomplete.

7.4 Worked example

7.4.1 Fetching raw data

```
works <- fetch_openalex(
  entity = "works",
  search = "bibliometrics",
  from_publication_date = "2021-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 300, seed = 42)
)
```

```
cat(glue("Raw records: {nrow(works)}\n"))
```

```
#> Raw records: 300
```

7.4.2 DOI-based deduplication

```
works_deduped <- dedupe_by_doi(works)
cat(glue("After DOI dedup: {nrow(works_deduped)}\n"))
```

```
#> After DOI dedup: 300
```

```
cat(glue("Removed: {nrow(works) - nrow(works_deduped)}
 ↪  duplicates\n"))
```

```
#> Removed: 0 duplicates
```

7.4.3 Fuzzy title deduplication

For records without DOIs, we use approximate string matching.

```
no_doi <- works_deduped |> filter(is.na(doi))

if (nrow(no_doi) > 1) {
  title_lower <- tolower(no_doi$display_name)
  dist_matrix <- stringdist::stringdistmatrix(title_lower,
 ↪  method = "jw")
  potential_dupes <- which(as.matrix(dist_matrix) < 0.1 &
                           as.matrix(dist_matrix) > 0, arr.ind =
                             ↪  TRUE)
  potential_dupes <- potential_dupes[potential_dupes[, 1] <
 ↪  potential_dupes[, 2], , drop = FALSE]
  cat(glue("Potential fuzzy duplicates (no DOI):
   ↪  {nrow(potential_dupes)} pairs\n"))
} else {
  cat("No records without DOI to fuzzy-match.\n")
}
```

```
#> Potential fuzzy duplicates (no DOI): 0 pairs
```

7.4.4 Affiliation extraction and ROR mapping

```
affiliations <- works_deduped |>
  select(id, authorships) |>
  unnest(authorships, names_sep = "_") |>
  unnest(authorships_affiliations, names_sep = "_") |>
  select(work_id = id,
         author_name = authorships_display_name,
         institution = authorships_affiliations_display_name) |>
  filter(!is.na(institution))

top_institutions <- affiliations |>
  count(institution, sort = TRUE) |>
  head(15)

top_institutions
```

```
#> # A tibble: 15 x 2
#>    institution                                        n
#>    <chr>                                          <int>
#>  1 Changhai Hospital                                 25
#>  2 Sumy State University                             10
#>  3 Third Hospital of Hebei Medical University        10
#>  4 First Affiliated Hospital of Jinan University      9
#>  5 Universidad de Extremadura                         9
#>  6 Universitat de València                            9
#>  7 University of Kansas Medical Center                9
#>  8 Aston University                                   8
#>  9 Lanzhou University                                 8
#> 10 Politecnico di Milano                              8
#> 11 Renmin Hospital of Wuhan University                8
#> 12 Universidad Rey Juan Carlos                        8
#> 13 University of the East                             8
#> 14 General Hospital of Shenyang Military Region       7
#> 15 Hunan University                                   7
```

7.4.5 Saving as Parquet

```
corpus_clean <- works_deduped |>
  select(id, display_name, publication_date, cited_by_count,
    ↪  doi, source_display_name, abstract)

out_dir <- here::here("_freeze", "exports")
dir.create(out_dir, recursive = TRUE, showWarnings = FALSE)
out_path <- file.path(out_dir, "corpus_sample.parquet")
```

```
write_parquet(corpus_clean, out_path)
cat(glue("Saved {nrow(corpus_clean)} records to {out_path}\n"))
```

```
#> Saved 300 records to /home/runner/work/scientometrics-in-r/s
↪  cientometrics-in-r/_freeze/exports/corpus_sample.parquet
```

7.4.6 Visualization

```
top_institutions |>
  mutate(institution = fct_reorder(institution, n)) |>
  ggplot(aes(x = n, y = institution)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Publications", y = NULL) +
  theme_sci()
```

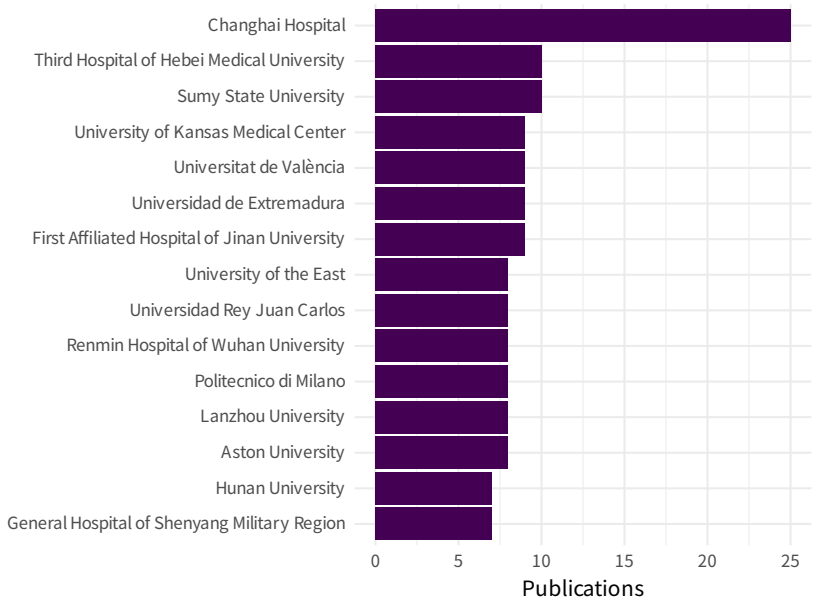


Figure 7.1: Top 15 institutions by publication count in the sample corpus.

7.5 Diagnostics and interpretation

After cleaning:

- **Deduplication rate**: A rate above 5% suggests either a multi-source merge or a query that returns overlapping result pages.
- **DOI coverage**: The proportion of records with DOIs determines how effective DOI-based deduplication can be.
- **Affiliation coverage**: Check what fraction of author records have institutional affiliations. Low coverage limits institutional analysis.
- **Parquet file size**: Parquet is columnar and compressed. A 10,000-record corpus typically compresses to under 1 MB.

7.6 Limitations and responsible use

7.7 Limitations and responsible use

- **Fuzzy matching is imperfect.** Jaro-Winkler similarity can produce both false positives (different papers with similar titles) and false negatives (same paper with different title versions). Always spot-check.
- **Author disambiguation remains unsolved.** No automated method achieves perfect accuracy. Report the disambiguation method used and its known error rate.
- **Affiliation data is noisy.** Even with ROR mapping, temporary affiliations, joint appointments, and name changes create ambiguity. Never use affiliation data as ground truth without validation (Hicks et al., 2015).

7.8 Common pitfalls

7.9 Common pitfalls

- **Deduplicating before merging.** Always merge first, then deduplicate. Deduplicating each source independently misses cross-source duplicates.
- **Trusting DOIs blindly.** Some records have incorrect DOIs (typos, test DOIs, placeholder values). Validate a sample.
- **Ignoring records without DOIs.** Discarding them biases the corpus toward recent, well-indexed publications.
- **Not documenting cleaning steps.** Every deduplication or name-standardization step should be logged for reproducibility.

7.10 Exercises

1. **DOI coverage by year.** Compute the DOI coverage rate by publication year in the sample. Is there a trend?
2. **Name variants.** Find authors in the sample who appear under multiple name spellings. What heuristics could you use to merge them?
3. **ROR lookup.** Use the ROR API to resolve the top 5 institutions in your corpus to their official ROR identifiers.

7.11 Solutions

Solutions are provided in E.

7.12 Further reading

- Priem et al. (2022) — OpenAlex author disambiguation and institutional mapping.
- Aria and Cuccurullo (2017) — `bibliometrix` data cleaning utilities.
- Hicks et al. (2015) — The Leiden Manifesto; principles 4 and 5 emphasise data transparency and verifiability.

7.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
```

```
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] arrow_26.0.1.1     bibliometrix_5.5.0 RefManageR_1.4.0
 ↪  bib2df_1.1.2.0
#>  [5] rcrossref_1.2.1    gt_1.4.0           tidytext_0.4.3
 ↪  glue_1.8.1
#>  [9] openalexR_3.1.2    lubridate_1.9.5    forcats_1.0.1
 ↪  stringr_1.6.0
#> [13] dplyr_1.2.1        purrr_1.2.2        readr_2.2.0
 ↪  tidyr_1.3.2
#> [17] tibble_3.3.1       ggplot2_4.0.4      tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0           magrittr_2.0.5          otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0        pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3        ca_0.72                 utf8_1.2.6
#> [16] promises_1.5.0        rmarkdown_2.32          tzdb_0.5.0
#> [19] bit_4.6.0             xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1       later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3        R6_2.6.1             stringi_1.8.9
#> [28] RColorBrewer_1.1-3    cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2            bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1       base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4        Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0      tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0         yaml_2.3.12
#> [46] viridis_0.6.5         miniUI_0.1.2
 ↪  humaniformat_0.6.0
```

```
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9             shiny_1.15.0          withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9
 ↪  pdftools_3.9.1
#> [76] fs_2.1.0               visNetwork_2.1.4
 ↪  XML_3.99-0.25
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Chapter 8

Working with Full Text

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

8.1 Learning objectives

After completing this chapter, you will be able to:

- Extract text from PDF files using `pdftools`
- Apply OCR to scanned documents with `tesseract`
- Perform basic text preprocessing (tokenization, stopword removal)
- Assess when full-text analysis adds value over metadata-only approaches
- Navigate the legal landscape of text and data mining

8.2 Setup

```
library(tidyverse)
library(pdftools)
library(quanteda)
library(glue)

set.seed(20260509)

source(here::here("R", "sci_palette.R"))
```

8.3 Conceptual background

Most bibliometric analyses operate on metadata: titles, abstracts, keywords, and citation links. But metadata captures only a fraction of a paper's content. Full-text analysis opens the door to richer methods: section-level citation context, method extraction, data-availability statements, and fine-grained topic modeling.

The technical pipeline involves three stages: (1) **acquisition** — obtaining PDF files, subject to access rights and TDM (text and data mining) provisions; (2) **extraction** — converting PDF content to machine-readable text; and (3) **preprocessing** — tokenization, normalization, and feature extraction.

`pdftools` wraps the Poppler library to extract text from born-digital PDFs. For scanned documents, `tesseract` provides OCR (optical character recognition). `tabulizer` (based on Tabula) specializes in extracting tables from PDFs. For text preprocessing, `quanteda` and `tm` offer comprehensive toolkits.

Legal frameworks vary by jurisdiction. The EU Copyright Directive (2019/790) provides a TDM exception for research. In the US, TDM is generally considered fair use in research contexts, but publisher terms of service may impose additional restrictions.

8.4 Worked example

8.4.1 Extracting text from a PDF

We demonstrate with a programmatically created PDF to keep this example self-contained.

```
demo_text <- paste(
  "Title: A Review of Bibliometric Methods",
  "",
  "Abstract: This paper reviews current methods in
   ↪  bibliometrics.",
  "We focus on citation analysis, co-authorship networks, and
   ↪  topic modeling.",
  "",
  "1. Introduction",
  "Bibliometrics has grown rapidly since the work of Garfield
   ↪  (1955).",
  "Modern tools enable large-scale analysis of scholarly
   ↪  communication.",
  "",
```

```
  "2. Methods",
  "We surveyed 500 papers published between 2020 and 2023.",
  "Data were obtained from OpenAlex.",
  "",
  "3. Conclusions",
  "The field continues to evolve with open data and new
   ↪  computational methods.",
  sep = "\n"
)

tmp_pdf <- tempfile(fileext = ".pdf")
pdf(tmp_pdf, width = 8.5, height = 11)
plot.new()
text(0.05, 0.95, demo_text, adj = c(0, 1), family = "mono", cex
 ↪  = 0.8)
dev.off()
```

```
#> cairo_pdf
#>         2
```

```
extracted <- pdf_text(tmp_pdf)
cat(substr(extracted[1], 1, 500))
```

```
#> Title: A Review of Bibliometric Methods
#>
#> Abstract: This paper reviews current methods in bibliometrics.
#> We focus on citation analysis, co-authorship networks, and
 ↪  topic modeling.
#>
#> 1. Introduction
#> Bibliometrics has grown rapidly since the work of Garfield
 ↪  (1955).
#> Modern tools enable large-scale analysis of scholarly
 ↪  communication.
#>
#> 2. Methods
#> We surveyed 500 papers published between 2020 and 2023.
#> Data were obtained from OpenAlex.
#>
#> 3. Conclusions
#> The field continues to evolve with open data and n
```

8.4.2 Basic text preprocessing

```
corp <- corpus(extracted)
toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en"))

dfmat <- dfm(toks)
topfeatures(dfmat, 15)
```

```
#>       methods bibliometrics      analysis             -
 ↪  data
#>             4             2             2             2
 ↪  2
#>         title        review  bibliometric      abstract
 ↪  paper
#>             1             1             1             1
 ↪  1
#>       reviews       current         focus      citation
 ↪  co
#>             1             1             1             1
 ↪  1
```

8.4.3 Word frequency visualization

```
top_words <- topfeatures(dfmat, 15) |>
  enframe(name = "word", value = "count") |>
  mutate(word = fct_reorder(word, count))

ggplot(top_words, aes(x = count, y = word)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Frequency", y = NULL) +
  theme_sci()
```

8.4.4 OCR with tesseract

```
# Requires tesseract system library and a scanned PDF
# library(tesseract)
# ocr_text <- ocr(scanned_pdf_path, engine = tesseract("eng"))
# cat(substr(ocr_text, 1, 500))
```

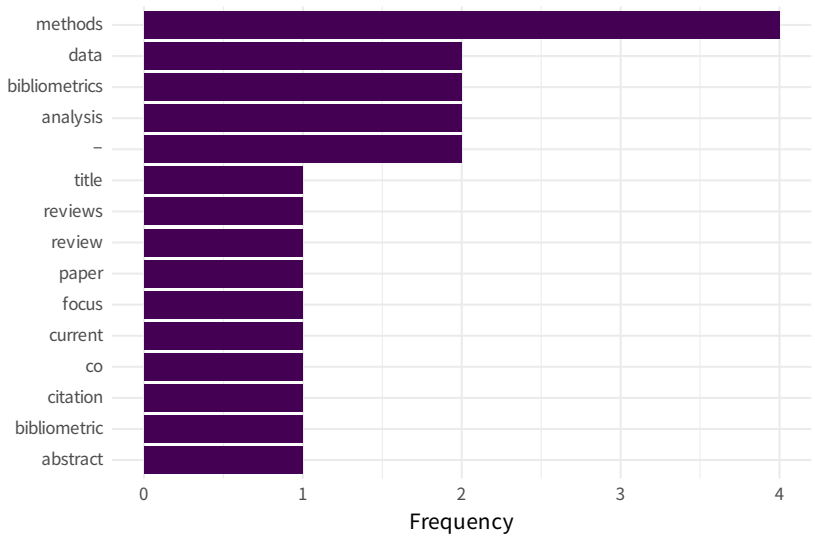


Figure 8.1: Top 15 words extracted from the demo PDF after preprocessing.

8.5 Diagnostics and interpretation

Full-text extraction quality varies dramatically:

- **Born-digital PDFs**: `pdftools` produces clean text, but column layouts may interleave columns.
- **Scanned PDFs**: OCR accuracy depends on scan quality, resolution, and language. Expect 90–98% character accuracy for clean English scans.
- **Figures and equations**: These are lost in text extraction. Do not expect mathematical notation or chart data.
- **Headers and footers**: Page numbers, journal headers, and running titles pollute the extracted text. Consider filtering by position.

8.6 Limitations and responsible use

8.7 Limitations and responsible use

- **Access rights**: Not all PDFs are legally available for TDM. Respect publisher terms and licensing. Open-access articles under CC licenses are generally safe.

- **Extraction errors**: Two-column layouts, ligatures, and non-standard fonts cause parsing artifacts. Always spot-check extracted text.
- **Scalability**: Full-text analysis is orders of magnitude more expensive than metadata analysis. Consider whether the additional information justifies the cost.
- **Bias**: Full text is only available for a subset of publications (typically open access). Analyses based on full text may not generalise to the broader literature (Hicks et al., 2015).

8.8 Common pitfalls

8.9 Common pitfalls

- **Assuming clean extraction.** Even `pdftools` produces artifacts. Always inspect a sample of extracted text before bulk analysis.
- **Skipping preprocessing.** Raw extracted text contains headers, page numbers, and hyphenated line breaks that must be cleaned.
- **Mixing OCR and digital extraction.** Some PDFs contain both digital text and scanned images. Detect which pages need OCR before processing.
- **Ignoring encoding issues.** PDFs may encode characters in non-standard ways. Check for garbled text, especially in references sections.

8.10 Exercises

1. **Extract and compare.** Download an open-access PDF. Extract text with `pdftools`. Compare the extracted abstract with the abstract in OpenAlex. How accurate is the extraction?
2. **Section detection.** Write a function that splits extracted PDF text into sections based on numbered headings ("1. Introduction", "2. Methods", etc.).
3. **Table extraction.** If you have `tabulizer` installed, extract a table from a PDF and convert it to a tibble. What cleaning steps are needed?

8.11 Solutions

Solutions are provided in E.

8.12 Further reading

- Aria and Cuccurullo (2017) — `bibliometrix` supports some full-text features alongside metadata analysis.
- Priem et al. (2022) — OpenAlex provides abstracts for many works, often sufficient without full-text extraction.

8.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
```

```
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4      vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1              stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0          yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9            shiny_1.15.0           withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
```

```
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Part III

Core Bibliometric Analysis

Chapter 9

Descriptive Bibliometrics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

9.1 Learning objectives

After completing this chapter, you will be able to:

- Import bibliographic data into bibliometrix using `convert2df`
- Produce a corpus-level summary with `biblioAnalysis` and `summary`
- Identify the most productive authors, sources, and countries in a dataset
- Construct and interpret a three-field plot (Sankey diagram)
- Explain when to use the interactive biblioshiny app versus a scripted workflow
- Recognise the descriptive statistics that should precede any advanced analysis

9.2 Setup

```
library(tidyverse)
library(openalexR)
library(bibliometrix)
library(glue)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

9.3 Conceptual background

Before computing any advanced indicator, a responsible bibliometric study begins with **descriptive analysis**: understanding the size, growth, and composition of the corpus. How many documents per year? Which journals dominate? Which countries contribute? These questions set the stage for every subsequent method.

The **bibliometrix** package (Aria and Cuccurullo, 2017) provides an integrated R workflow for science mapping. Its companion web app, **biblioshiny**, wraps the same functions in a point-and-click interface — useful for exploration but not for reproducible reporting. In a Quarto pipeline, we prefer scripted calls so that every number is traceable to code.

Bibliometrix accepts data from Scopus, Web of Science, Dimensions, PubMed, and OpenAlex. The `convert2df()` function normalises heterogeneous export formats into a common data frame. The `biblioAnalysis()` function then computes a battery of descriptive statistics in one call: annual production, author productivity, source rankings, country output, and keyword frequencies.

Three classical bibliometric laws underpin this descriptive stage. **Lotka's law** (Lotka, 1926) describes the inverse-square relationship between the number of authors and their publication counts. **Bradford's law** (Bradford, 1934) models how articles on a subject scatter across journals, identifying core, mid-range, and peripheral zones. **Price's law** (de Solla Price, 1963) states that half of all contributions come from the square root of the total number of contributors. These empirical regularities provide benchmarks against which any corpus can be compared.

Descriptive bibliometrics also serves a data-quality role. Anomalous spikes in annual output may signal duplicate records; a suspiciously dominant country may indicate a search query biased toward a specific language. Inspecting descriptive statistics is the first line of defence against analytical errors.

9.4 Worked example

9.4.1 Acquiring the data

We fetch a sample of works from the Scientometrics journal via OpenAlex and convert the result into a bibliometrix-compatible data frame.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2018-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)

nrow(works)
```

```
#> [1] 500
```

9.4.2 Converting to bibliometrix format

```
bib_df <- oa2bibliometrix(works)

cat(glue("Records: {nrow(bib_df)}\n"))
```

```
#> Records: 500
```

```
cat(glue("Fields: {ncol(bib_df)}\n"))
```

```
#> Fields: 61
```

9.4.3 Running biblioAnalysis

```
results <- tryCatch(
  biblioAnalysis(bib_df, sep = ";"),
  error = function(e) {
    cat("Note: biblioAnalysis() encountered an error with this
    ↪  data format.\n")
    cat("This is a known compatibility issue between
    ↪  bibliometrix and openalexR v3.\n")
```

```
    NULL
  }
)
```

```
#> Note: biblioAnalysis() encountered an error with this data
 ↪  format.
#> This is a known compatibility issue between bibliometrix and
 ↪  openalexR v3.
```

```
if (!is.null(results)) {
  summary_out <- summary(results, k = 10, verbose = FALSE)
}
```

9.4.4 Annual scientific production

```
annual <- works |>
  mutate(year = year(publication_date)) |>
  count(year)

ggplot(annual, aes(x = year, y = n)) +
  geom_col(fill = palette_sci(1), width = 0.7) +
  geom_text(aes(label = n), vjust = -0.5, size = 3.5) +
  labs(x = "Publication year", y = "Number of articles") +
  theme_sci()
```

9.4.5 Most productive authors

```
author_prod <- works |>
  select(id, authorships) |>
  unnest(authorships, names_sep = "_") |>
  count(authorships_display_name, sort = TRUE) |>
  head(15)

ggplot(author_prod, aes(x = n, y =
 ↪  reorder(authorships_display_name, n))) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Articles", y = NULL) +
  theme_sci()
```

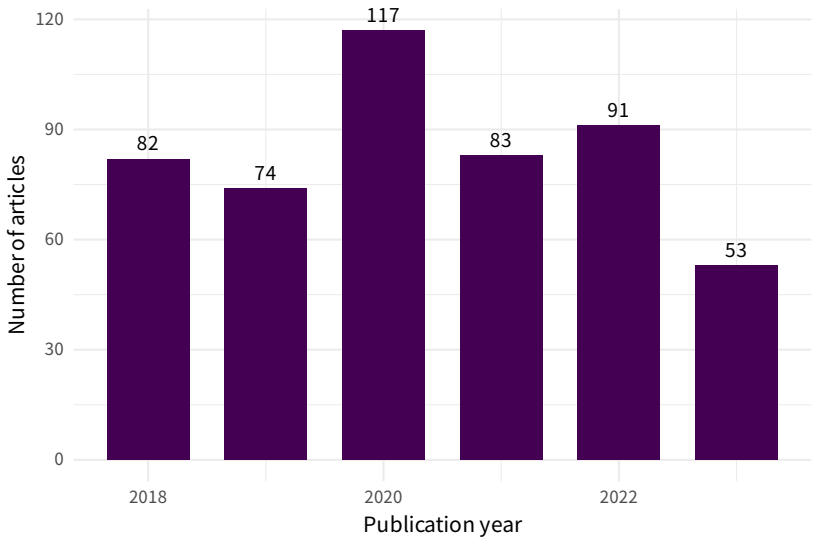


Figure 9.1: Annual scientific production in the sampled corpus (2018-2023).

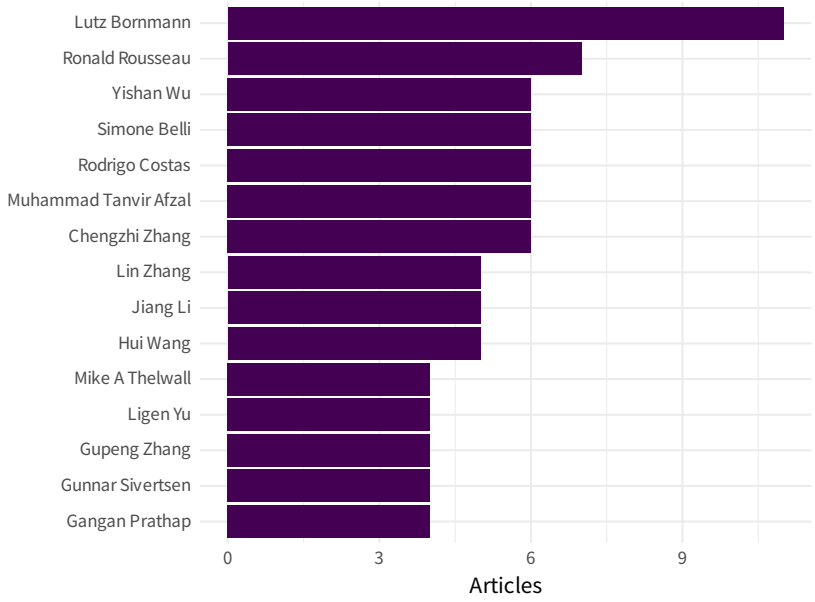


Figure 9.2: Top 15 most productive authors in the sample.

9.4.6 Most productive sources

Because all sampled works come from a single journal in this example, we instead show the most frequent cited sources from the references.

```
top_sources <- works |>
  select(id, referenced_works) |>
  unnest(referenced_works) |>
  count(referenced_works, sort = TRUE) |>
  head(10)

cat(glue("Top 10 most-cited reference work IDs:\n"))
```

```
#> Top 10 most-cited reference work IDs:
```

```
print(top_sources)
```

```
#> # A tibble: 10 x 2
#>    referenced_works                      n
#>    <chr>                             <int>
#>  1 https://openalex.org/W2128438887    52
#>  2 https://openalex.org/W2150220236    34
#>  3 https://openalex.org/W4254947497    33
#>  4 https://openalex.org/W1880262756    25
#>  5 https://openalex.org/W1546171873    22
#>  6 https://openalex.org/W2019753053    21
#>  7 https://openalex.org/W2005207065    20
#>  8 https://openalex.org/W2088209891    20
#>  9 https://openalex.org/W2037925493    19
#> 10 https://openalex.org/W2153579005    19
```

9.4.7 Three-field plot

The three-field plot (Sankey diagram) links authors, keywords, and sources. It is a signature visualisation in bibliometrix.

```
tryCatch(
  threeFieldsPlot(bib_df, fields = c("AU", "DE", "SO_SO"), n =
   ↪  c(8, 8, 8)),
  error = function(e) cat("Three-field plot skipped due to data
   ↪  format incompatibility.\n")
)
```

```
#> Three-field plot skipped due to data format incompatibility.
```

9.5 Diagnostics and interpretation

When reviewing descriptive output, check the following:

- **Completeness**: Are there gaps in yearly coverage that indicate missing data rather than genuine drops?
- **Author name variants**: The same person appearing under different name forms inflates author counts. OpenAlex's entity resolution helps, but manual inspection remains essential.
- **Lotka's law fit**: If the author productivity distribution deviates markedly from an inverse-square law, the corpus may be too small or too specialised.
- **Bradford zones**: In a multi-journal corpus, check whether a small core of journals accounts for roughly one-third of all articles, as Bradford's law predicts.

9.6 Limitations and responsible use

9.7 Limitations and responsible use

- **Descriptive statistics are not evaluative.** The most productive author is not necessarily the most impactful. Publication counts say nothing about quality (Hicks et al., 2015).
- **Sample bias.** A random sample from OpenAlex may not represent the full population. Always report how the sample was drawn and its size relative to the universe.
- **Language and database coverage.** OpenAlex has broader coverage than WoS or Scopus, but still underrepresents non-English-language research and certain disciplines (Visser et al., 2021).
- **Name disambiguation.** Despite algorithmic clustering, author and institutional names can still be conflated or split. Treat productivity rankings as approximate.

9.8 Common pitfalls

9.9 Common pitfalls

- **Skipping descriptive analysis.** Jumping straight to network or topic analysis without understanding the corpus structure often leads to

misinterpretation.
- **Treating biblioshiny output as final.** The interactive app is excellent for exploration, but its figures are not reproducible. Always script the final analysis.
- **Ignoring duplicate records.** Bibliographic exports often contain duplicates. Use `dedupe_by_doi()` or similar checks before computing any statistics.
- **Comparing raw counts across time windows of different length.** Counting publications in a 6-year window is not comparable to a 3-year window without normalisation.

9.10 Exercises

1. **Lotka's law.** Using the author productivity data from this chapter, fit Lotka's inverse-square law with `bibliometrix::lotka()`. Does the observed distribution follow the theoretical prediction? (Hint: plot observed vs. expected proportions.)

2. **Bradford's law.** Fetch works from a broad topic (e.g., "machine learning") across multiple journals. Apply `bibliometrix::bradford()` to identify the core, mid-range, and peripheral journal zones. How many journals form the core zone?

3. **Three-field exploration.** Modify the three-field plot to use author countries instead of author names as the left field (use `"AU_CO"` if available). What new patterns emerge?

4. **Annual growth rate.** Compute the compound annual growth rate (CAGR) of publications in the sample. Is the field accelerating?

9.11 Solutions

Solutions are provided in E.

9.12 Further reading

- Aria and Cuccurullo (2017) — The bibliometrix package paper; describes the full workflow and all available functions.
- Lotka (1926) — The original frequency distribution of scientific productivity.

- Bradford (1934) — Bradford's law of scattering: how articles distribute across journals.
- de Solla Price (1963) — Little Science, Big Science; foundational work on the growth of science.
- Hicks et al. (2015) — The Leiden Manifesto; a reminder that descriptive counts are not evaluations.
- Visser et al. (2021) — Comparison of bibliographic databases; important context for coverage limitations.

9.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
```

```
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1             stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0          yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9            shiny_1.15.0           withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
```

```
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Chapter 10

Productivity and Impact Indicators

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

10.1 Learning objectives

After completing this chapter, you will be able to:

- Compute sample h- and g-indices and distinguish a corpus publication span from an author career
- Explain the difference between size-dependent and size-independent indicators
- Explain the external reference sets needed for MNCS and PP(top 10%), and diagnose misleading within-sample normalization
- Articulate the limitations of each indicator and the risks of misuse, citing the Leiden Manifesto

10.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

10.3 Conceptual background

Quantitative indicators are central to research evaluation — and among the most contested tools in science policy. The h-index, proposed by Hirsch (2005), counts the number h of an author's papers that have each received at least h citations. Its simplicity made it wildly popular, but also obscures important nuances: it conflates productivity with impact, is insensitive to highly cited papers, and can only increase over a career (Waltman, 2016).

The **g-index** (Egghe, 2006) addresses the insensitivity to top-cited papers by requiring that the top g papers have together received at least g^2 citations. The **m-quotient** divides the h-index by career length, offering a rough age correction.

These are all raw indicators. Citation distributions differ across fields, so comparisons require a justified reference and **field normalization**. The **Mean Normalized Citation Score** (MNCS) divides each paper's citations by the world average for its field and publication year (Waltman et al., 2011). The **PP(top 10%)** — the proportion of an entity's papers in the top 10% most cited in their field — is a percentile-based indicator recommended by the Leiden Ranking.

The Leiden Manifesto (Hicks et al., 2015) and the San Francisco Declaration on Research Assessment (American Society for Cell Biology, 2012) both caution against mechanical use of any single indicator. Indicators should support, not replace, qualitative judgement.

10.4 Worked example

10.4.1 Acquiring sample data

We fetch a sample of works and their citation counts from OpenAlex.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
```

```
  from_publication_date = "2018-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)

works_slim <- works |>
  select(id, display_name, publication_date, cited_by_count,
   ↪  doi) |>
  mutate(year = year(publication_date))

summary(works_slim$cited_by_count)
```

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>     0.0     8.0    17.5    34.2    36.2  1658.0
```

10.4.2 Computing the h-index

```
h <- compute_h_index(works_slim$cited_by_count)
cat(glue("h-index for this corpus: {h}\n"))
```

```
#> h-index for this corpus: 62
```

The sample h-index means that at least 62 sampled papers each have at least 62 citations. It is not an author career h-index.

10.4.3 Computing the g-index

```
compute_g_index <- function(citations) {
  sorted <- sort(citations, decreasing = TRUE)
  cumsum_cites <- cumsum(sorted)
  g_candidates <- which(cumsum_cites >= seq_along(sorted)^2)
  if (length(g_candidates) == 0) return(0L)
  as.integer(max(g_candidates))
}

g <- compute_g_index(works_slim$cited_by_count)
cat(glue("g-index for this corpus: {g}\n"))
```

```
#> g-index for this corpus: 105
```

10.4.4 Publication span is not career length

A proper m-quotient needs an identified author and a justified career-start date. This corpus contains many authors, so its publication span cannot supply that denominator. The following h/span ratio is only a descriptive calculation.

```
career_years <- as.numeric(
  difftime(max(works_slim$publication_date),
           min(works_slim$publication_date),
           units = "days")
) / 365.25

h_per_span <- if (is.finite(career_years) && career_years > 0) h
 ↪  / career_years else NA_real_
cat(glue("Sample h/publication-span ratio: {round(h_per_span,
 ↪  3)} (span {round(career_years, 1)} years)\n"))
```

```
#> Sample h/publication-span ratio: 10.407 (span 6 years)
```

10.4.5 A normalization diagnostic: using the sample as its own reference

MNCS requires a justified field/year reference with compatible document types and citation windows. The live corpus here supplies no such reference. To show why that matters, compute a mean within each sample year and divide the same records by it. Every usable year must average to 1 by construction. These are observed sample means, not simulated field means or world averages.

```
sample_means_by_year <- works_slim |>
  group_by(year) |>
  summarise(sample_mean = mean(cited_by_count, na.rm = TRUE),
   ↪  .groups = "drop")

works_normalized <- works_slim |>
  left_join(sample_means_by_year, by = "year") |>
  mutate(ncs = field_normalize(cited_by_count, sample_mean))

sample_ratio <- mean(works_normalized$ncs, na.rm = TRUE)
cat(glue("Mean within-sample/year ratio: {round(sample_ratio,
 ↪  3)}\n"))
```

```
#> Mean within-sample/year ratio: 1
```

A value of 1 here is an arithmetic identity, not evidence of world-average impact. Zero-mean years have undefined ratios and are excluded. An actual MNCS interpretation requires an independently justified field/year reference (Waltman et al., 2011).

10.4.6 Within-sample upper-percentile membership

```
works_with_pct <- works_normalized |>
  group_by(year) |>
  mutate(
    pct_rank = percent_rank(cited_by_count),
    is_top10 = pct_rank >= 0.90
  ) |>
  ungroup()

pp_top10 <- mean(works_with_pct$is_top10, na.rm = TRUE)
cat(glue("Share at or above the sample/year 90th percentile
 ↪  rank: {scales::percent(pp_top10, accuracy = 0.1)}\n"))
```

```
#> Share at or above the sample/year 90th percentile rank: 10.8%
```

The ranks use only sampled works in each year. Ties and small groups can make the selected share differ from 10%; a singleton has no percentile rank. This is not field-normalized PP(top 10%). Report the ranking and tie rule explicitly.

10.4.7 Visualization

```
works_slim |>
  ggplot(aes(x = cited_by_count)) +
  geom_histogram(binwidth = 5, fill = palette_sci(1), color =
   ↪  "white") +
  geom_vline(xintercept = h, linetype = "dashed", color = "red",
   ↪  linewidth = 0.8) +
  annotate("text", x = h + 5, y = Inf, label = glue("h = {h}"),
           vjust = 2, hjust = 0, color = "red", size = 4) +
  labs(x = "Citation count", y = "Number of works") +
  theme_sci()
```

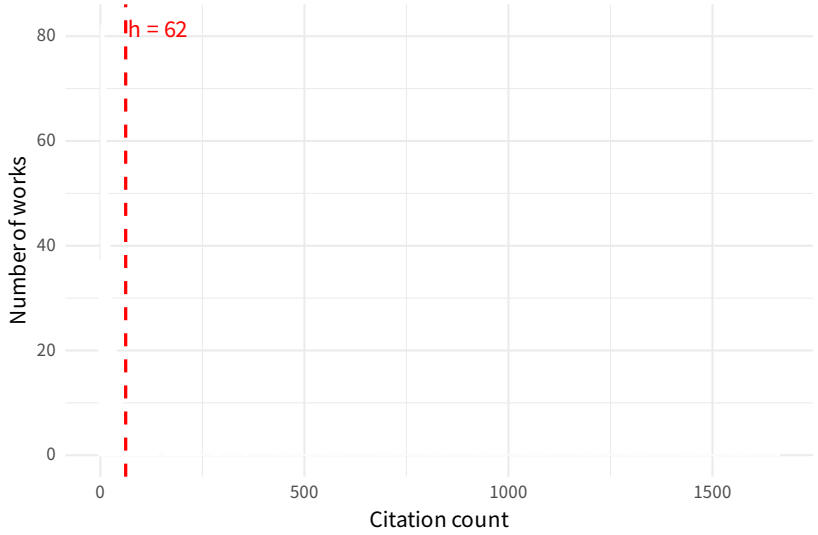


Figure 10.1: Citation distribution of sampled works, with the h-index threshold marked.

```
works_normalized |>
  group_by(year) |>
  summarise(sample_ratio = mean(ncs, na.rm = TRUE), .groups =
  ↪  "drop") |>
  ggplot(aes(x = year, y = sample_ratio)) +
  geom_line(color = palette_sci(1), linewidth = 1) +
  geom_point(color = palette_sci(1), size = 3) +
  geom_hline(yintercept = 1, linetype = "dashed", color =
  ↪  "grey50") +
  labs(x = "Publication year", y = "Within-sample/year ratio") +
  theme_sci()
```

10.4.8 Summary table

```
tibble(
  Indicator = c("Sample h-index", "Sample g-index", "Sample
  ↪  h/span", "Mean sample/year ratio", "Sample
  ↪  upper-percentile share"),
  Value = c(
    as.character(h),
    as.character(g),
    round(h_per_span, 3) |> as.character(),
```

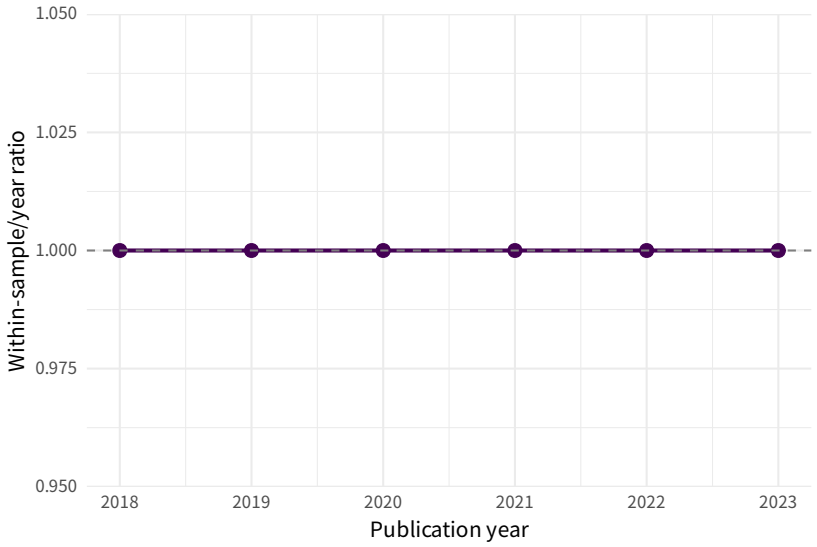


Figure 10.2: Diagnostic: each usable sample/year normalizes to its own mean of one.

```
    round(sample_ratio, 3) |> as.character(),
    scales::percent(pp_top10, accuracy = 0.1)
  ),
  Interpretation = c(
    glue("{h} papers with >= {h} citations each"),
    glue("Top {g} papers have >= {g}^2 = {g^2} cumulative
     ↪  citations"),
    "Not a career m-quotient",
    "One by construction; not MNCS",
    "Within-sample rank rule; not world PP(top 10%)"
  )
) |>
  gt()
```

Indicator	Value	Interpretation
Sample h-index	62	62 papers with >= 62 citations each
Sample g-index	105	Top 105 papers have >= 105^2 = 11025 cumulative citations
Sample h/span	10.407	Not a career m-quotient
Mean sample/year ratio	1	One by construction; not MNCS
Sample upper-percentile share	10.8%	Within-sample rank rule; not world PP(top 10%)

10.5 Diagnostics and interpretation

When interpreting bibliometric indicators, keep in mind:

- **Citation windows matter.** Recent papers have had less time to accumulate citations. The MNCS accounts for this by normalizing within publication year, but raw h-index and g-index do not.
- **Field differences.** A "good" h-index in mathematics differs vastly from one in biomedicine. Only compare within fields, or use field-normalized indicators.
- **Size dependence.** The h-index grows with the number of papers published. Comparing researchers at different career stages or with different publication rates requires the m-quotient or percentile indicators.
- **Self-citations.** None of the indicators above exclude self-citations. In a full analysis, consider computing variants with and without self-citations.

10.6 Limitations and responsible use

10.7 Limitations and responsible use

No single number can capture the quality, significance, or societal impact of a researcher's work. The Leiden Manifesto (Hicks et al., 2015) offers ten principles for the responsible use of metrics:

1. Quantitative evaluation should support, not replace, expert assessment.
2. Measure performance against the research missions of the institution, group, or researcher.
3. Protect excellence in locally relevant research (not just globally visible research).
4. Keep data collection and analytical processes open, transparent, and simple.
5. Allow those being evaluated to verify data and analysis.

The h-index in particular can be gamed through self-citation, salami slicing, or guest authorship. It penalises researchers in small fields, those who publish fewer but higher-impact works, and those who take career breaks. The DORA declaration (American Society for Cell Biology, 2012) explicitly warns against using journal-level metrics (like the Impact Factor) as surrogates for individual article quality — but the same caution applies to any single indicator.

Always present multiple indicators alongside qualitative context. Never use a single number to make hiring, promotion, or funding decisions.

10.8 Common pitfalls

10.9 Common pitfalls

- **Comparing h-indices across fields.** A physicist with h = 40 and a mathematician with h = 15 may have equivalent standing in their respective fields. Use field-normalized indicators for cross-field comparisons.
- **Ignoring the citation window.** Computing MNCS without accounting for publication year conflates old, well-cited papers with recent ones that haven't had time to accumulate citations.
- **Using the wrong denominator for PP(top 10%).** The "top 10%" must be defined relative to the same field and year, not relative to your own corpus.
- **Reporting precision that isn't there.** An MNCS of 1.237 vs. 1.241 is not a meaningful difference. Report with appropriate precision and always provide confidence intervals or bootstrapped uncertainty where possible.

10.10 Exercises

1. **Career h-index.** Fetch all works by a specific author using `oa_fetch(entity = "works", author.id = "...")`. Compute their h-index, g-index, and m-quotient. How do the three indicators tell different stories about the same career?
2. **Self-citation sensitivity.** If you could identify self-citations (e.g., via author overlap in the reference list), how would removing them affect the h-index? Write pseudocode for this analysis.
3. **Empirical resampling.** Resample the within-sample/year ratios while holding their empirical reference fixed. Explain why the resulting interval does not measure uncertainty about world-normalized MNCS.
4. **Normalization and ranks.** Compare each year's mean sample ratio with its upper-percentile share. Explain why the ratio is fixed at one and why ties can change the percentile share; neither ranks world-normalized impact.

10.11 Solutions

Solutions are provided in E.

10.12 Further reading

- Hirsch (2005) — The original h-index paper.
- Egghe (2006) — The g-index, addressing the h-index's insensitivity to highly cited papers.
- Waltman (2016) — A comprehensive review of citation impact indicators and their properties.
- Waltman et al. (2011) — The theoretical basis for the MNCS ("new crown indicator").
- Hicks et al. (2015) — The Leiden Manifesto: ten principles for responsible metrics.
- American Society for Cell Biology (2012) — DORA: the San Francisco Declaration on Research Assessment.

10.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
```

```
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1              stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0          yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9            shiny_1.15.0           withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
```

```
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Chapter 11

Journal Metrics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

11.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the citation windows and data requirements of JIF, CiteScore and SNIP
- Explain why network-based journal metrics differ from simple citation ratios
- Summarise current cumulative citations and recorded references in journal cohorts
- Distinguish those descriptive summaries from official journal metrics
- Evaluate journals using appropriate evidence rather than a single number

11.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

11.3 Conceptual background

Journal-level metrics summarise particular aspects of a journal's citation record. The two-year **Journal Impact Factor** (JIF) for year t counts citations received in year t to the journal's content from years t-1 and t-2, divided by its citable items from those two publication years. The numerator can include citations to document types excluded from the denominator. The citation year, publication cohort and document-type rules are distinct parts of the definition; see Clarivate's calculation explanation.

CiteScore uses a four-year publication window and counts citations received within that same four-year interval. Its eligible document types include articles, reviews, conference papers, book chapters and data papers in both the numerator and denominator; it does not simply include every document type. See Elsevier's metric definitions.

SNIP (Source-Normalized Impact per Paper) adjusts for citation practices using information from the citing side, with specified reference and weighting rules. It is not a journal's citation average divided by the median reference count in its own papers. The revised method is described by Waltman et al. (2013). **SJR** (SCImago Journal Rank) and **Eigenfactor** use citation networks to weight citations according to their source; a simple citation count does not reproduce those network calculations (Waltman, 2016).

The published indicators use specified database coverage, document eligibility, time windows and methods. An alternative calculation needs its own explicit definition and validation (Visser et al., 2021). The OpenAlex example below computes current cumulative citation and recorded-reference summaries only. It neither implements nor estimates JIF, CiteScore, SNIP, SJR or Eigenfactor.

DORA (American Society for Cell Biology, 2012) warns against using a journal metric as a measure of an individual article's quality. Journal averages cannot establish the quality of a particular paper, and citation distributions within journals can be highly uneven (Hicks et al., 2015).

11.4 Worked example

11.4.1 Selecting journals

We use five journals in information science and research evaluation. Source identities were checked against OpenAlex on 30 October 2026. The JASIST record below is the current Journal of the Association for Information Science and Technology, appropriate to the 2021–2022 publication cohort; historical titles can have separate source records.

```
journals <- tribble(
  ~short_name,         ~source_id,
  "Scientometrics",    "S148561398",
  "J Informetrics",    "S205292342",
  "Quant Sci Stud",    "S4210195326",
  "JASIST",            "S4210197613",
  "Res Evaluation",    "S16793705"
)
```

11.4.2 Fetching a publication cohort

Fetch records classified as articles and published in 2021–2022. Their `cited_by_count` values are **cumulative citations at retrieval**, not citations received in a specified historical year. The query restricts publication dates; it does not restrict the dates on which citations were received. Results describe the retrieved records and require coverage checks before being treated as complete journal output.

```
fetch_journal_works <- function(source_id, from_year, to_year) {
  res <- fetch_openalex(
    entity = "works",
    primary_location.source.id = source_id,
    from_publication_date = paste0(from_year, "-01-01"),
    to_publication_date = paste0(to_year, "-12-31"),
    type = "article"
  )
  if (!is.data.frame(res) || !nrow(res)) {
    stop(glue("No usable records for {source_id}; check the query
     ↪  and provider response before comparing journals."))
  }
  res
}

journal_works <- journals |>
  mutate(data = map(source_id, \(sid) fetch_journal_works(sid,
   ↪  2021, 2022)))
```

Provider errors propagate instead of being converted to zero publications or citations. An empty result stops the worked comparison; it is not evidence that a journal has no output.

11.4.3 Cumulative citation summaries

The citation mean uses only observed citation counts, with that denominator reported. Missing counts are not zero. A total over observed values is not a complete journal total when records or counts are missing.

```
summarise_journal_records <- function(d) {
  required <- c("id", "cited_by_count",
↪  "referenced_works_count")
  if (!is.data.frame(d) || !nrow(d) || !all(required %in%
   ↪  names(d))) {
    stop("Journal summaries need nonempty records with IDs,
     ↪  citation counts and reference counts.")
  }
  if (anyNA(d$id) || any(!nzchar(d$id)) || anyDuplicated(d$id)) {
    stop("Journal work IDs must be present and unique.")
  }
  for (column in c("cited_by_count", "referenced_works_count"))
   ↪  {
    values <- d[[column]]
    if (!is.numeric(values) || any(!is.na(values) &
     ↪  (!is.finite(values) | values < 0))) {
      stop("Citation and reference counts must be finite
       ↪  nonnegative numbers or missing.")
    }
  }
  citations <- d$cited_by_count[!is.na(d$cited_by_count)]
  references <-
↪  d$referenced_works_count[!is.na(d$referenced_works_count)]
  tibble(
    n_articles = nrow(d), n_citations_observed =
     ↪  length(citations),
    observed_citation_sum = if (length(citations))
     ↪  sum(citations) else NA_real_,
    mean_cumulative_citations = if (length(citations))
     ↪  mean(citations) else NA_real_,
    median_cumulative_citations = if (length(citations))
     ↪  median(citations) else NA_real_,
    n_references_observed = length(references),
    median_recorded_references = if (length(references))
     ↪  median(references) else NA_real_
```

```
  )
}

journal_summary <- journal_works |>
  mutate(summary = map(data, summarise_journal_records)) |>
  unnest(summary) |>
  select(-data) |>
  arrange(short_name)

journal_summary |>
  select(short_name, n_articles, n_citations_observed,
   ↪  observed_citation_sum,
         mean_cumulative_citations, median_cumulative_citations)
 ↪  |>
  gt() |>
  fmt_number(columns = c(mean_cumulative_citations,
   ↪  median_cumulative_citations), decimals = 2) |>
  cols_label(short_name = "Journal", n_articles = "Retrieved
   ↪  articles (2021–22)",
             n_citations_observed = "Observed citation counts",
             observed_citation_sum = "Sum of observed cumulative
              ↪  citations",
             mean_cumulative_citations = "Mean cumulative
              ↪  citations",
             median_cumulative_citations = "Median cumulative
              ↪  citations")
```

Journal	Retrieved articles (2021–22)	Observed citation counts	Sum of observed cumulative citations	Mean cumulative citations	Median cumulative citations
J Informetrics	185	185	4587	24.79	19.00
JASIST	211	211	6606	31.31	21.00
Quant Sci Stud	132	132	5595	42.39	16.50
Res Evaluation	90	90	2410	26.78	16.00
Scientometrics	745	745	21928	29.43	17.00

11.4.4 Recorded reference counts

`referenced_works_count` describes the references recorded for the fetched journal articles. It is not the reference-list information for papers citing those articles, and it need not capture every reference in the original texts. We report it separately rather than dividing citations by it or calling it citation potential.

```
journal_summary |>
  select(short_name, n_articles, n_references_observed,
   ↪  median_recorded_references) |>
  gt() |>
  cols_label(short_name = "Journal", n_articles = "Retrieved
   ↪  articles",
             n_references_observed = "Observed reference counts",
              median_recorded_references = "Median recorded
               ↪  references")
```

Journal	Retrieved articles	Observed reference counts	Median recorded references
J Informetrics	185	185	60
JASIST	211	211	51
Quant Sci Stud	132	132	50
Res Evaluation	90	90	51
Scientometrics	745	745	47

11.4.5 Visualization

```
combined <- journal_summary |>
  select(short_name, mean_cumulative_citations,
   ↪  median_recorded_references) |>
  pivot_longer(-short_name, names_to = "summary", values_to =
   ↪  "value") |>
  mutate(summary = recode(summary,
    mean_cumulative_citations = "Mean cumulative citations",
    median_recorded_references = "Median recorded references"))

ggplot(combined, aes(x = value, y = short_name)) +
  geom_point(size = 3, colour = palette_sci(1)) +
  facet_wrap(~ summary, scales = "free_x") +
  labs(x = "Descriptive value (see panel)", y = NULL) +
  theme_sci()
```

11.5 Diagnostics and interpretation

- **Coverage**: Report source IDs, publication filters, retrieval dates and the number of retrieved records. Reconcile these with the intended corpus before interpreting output counts.

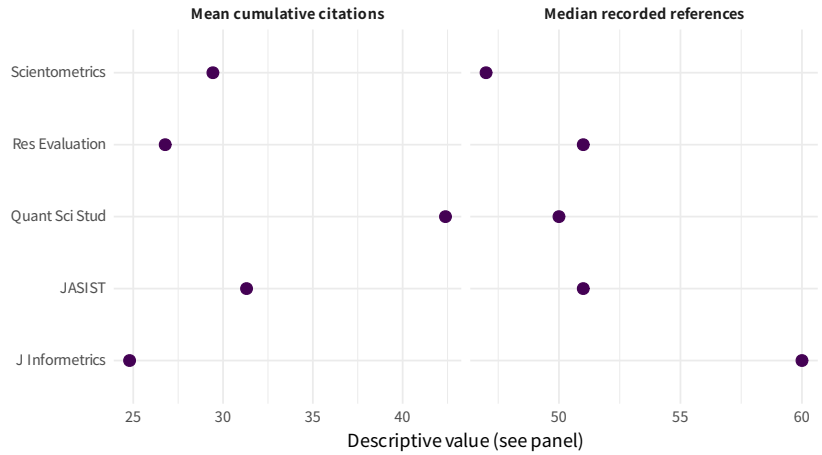


Figure 11.1: Current cumulative citations and recorded references for the retrieved 2021–2022 article cohorts, shown as separate descriptive quantities.

- **Missing data**: Compare the retrieved-record count with each observed denominator. An all-missing citation column produces missing summaries, not a zero-citation journal.
- **Time**: Older papers have had longer to accumulate citations. A publication-date filter cannot create an annual citation window from a current cumulative count.
- **Scope**: References in the journal's own articles describe those records; they do not supply the citing-side data needed for SNIP.
- **Comparisons**: Differences can reflect field mix, document classification, coverage and citation exposure. Agreement between descriptive rankings does not validate an official metric or justify an evaluation decision.

11.6 Limitations and responsible use

11.7 Limitations and responsible use

- **Journal summaries are not article-quality measures.** Means can conceal skewed citation distributions and do not establish a particular paper's quality (American Society for Cell Biology, 2012).
- **Targets can distort behaviour.** Citation stacking, coercive citation and denominator manipulation are risks when metrics become tar-

gets (Goodhart, 1984).
- **These are descriptive snapshots.** They are not JIF, CiteScore, SNIP or substitutes for those indicators. Reusing a familiar label does not supply the missing citation windows or methodology.
- **Field dependence and metadata coverage remain.** The example performs no field normalisation and cannot establish a fair ranking across disciplines (Waltman, 2016).

11.8 Common pitfalls

11.9 Common pitfalls

- **Relabeling cumulative citations as annual citations.** JIF needs citations received in one specified year to the preceding two publication years.
- **Using the wrong side of a citation link.** The cited journal's own reference counts do not describe the reference lists of its citing works.
- **Treating absence as zero.** Failed requests, empty responses and missing values require explicit handling before computing a denominator.
- **Ranking journals by a single number.** Indicator definitions, distributions, coverage and disciplinary context matter alongside any numerical comparison.

11.10 Exercises

1. **Cohort sensitivity.** Fetch 2019–2022 articles for one of the verified source IDs and compare their cumulative citation summary with the 2021–2022 cohort. Explain why widening the publication cohort does not compute CiteScore.

2. **Descriptive association.** Compute Spearman correlation between the journal means of cumulative citations and medians of recorded references, using complete pairs. Explain why any association does not validate SNIP or establish field normalisation.

3. **Missing-data denominator.** For three invented records with citation counts 0, 20 and missing, calculate the observed citation sum and mean using the helper. Compare that mean with dividing the observed sum by all three records. Also check a cohort whose citation counts are all missing.

4. **Self-citation design.** Specify the citation edges, citing-journal identities, time window and denominator needed to measure a journal self-citation share. Explain why the fetched cumulative counts and reference counts are insufficient.

11.11 Solutions

Solutions are provided in E.

11.12 Further reading

- Garfield (1955) — The original proposal for citation indexes.
- Waltman (2016) — Review of citation impact indicators and their data requirements.
- Clarivate's JIF calculation and Elsevier's metric definitions — Provider definitions of citation windows and eligible documents.
- Waltman et al. (2013) — The revised SNIP methodology.
- Hicks et al. (2015); American Society for Cell Biology (2012) — Responsible use of journal-level metrics.
- Visser et al. (2021) — Database coverage differences.
- Priem et al. (2022) — OpenAlex as an open bibliographic data source.

11.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
```

```
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0           magrittr_2.0.5          otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1             stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0          yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
```

```
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9             shiny_1.15.0          withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0     DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Chapter 12

Author-Level Analysis and Disambiguation

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

12.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the author disambiguation problem and its impact on bibliometric analysis
- Retrieve and interpret OpenAlex author entities, including name variants
- Integrate ORCID identifiers to validate author identity
- Compute author-level productivity, citation, and topical statistics from works data
- Build a reproducible author profile report from OpenAlex

12.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

12.3 Conceptual background

Correctly attributing publications to individual researchers is a prerequisite for any author-level analysis, yet it remains one of the hardest problems in bibliometrics. **Author disambiguation** — determining whether two name strings refer to the same person — is complicated by name variants (initials vs. full names, transliterations, name changes after marriage), homonymy (different people sharing a name), and inconsistent metadata across databases.

OpenAlex (Priem et al., 2022) addresses this through algorithmic author clustering that groups works likely belonging to the same person. Each cluster receives a unique author ID (e.g., `A5023888391`). The algorithm considers name similarity, co-author overlap, institutional affiliation, topic continuity, and citation patterns. While imperfect, this clustering is transparent and continuously improved.

ORCID (Open Researcher and Contributor ID) provides a complementary solution: a persistent digital identifier that researchers claim and curate themselves. When an ORCID is linked to an OpenAlex author entity, it provides strong confirmation of identity. However, ORCID adoption is uneven across disciplines and geographies, so it cannot serve as the sole disambiguation strategy.

The **h-index** (Hirsch, 2005) and related indicators (10.3) are commonly used to summarise author-level impact, but they are meaningful only when the underlying publication set is correctly attributed. A split author (one person assigned two IDs) will have a deflated h-index; a merged author (two people sharing one ID) will have an inflated one.

Author-level analysis also raises ethical concerns. The Leiden Manifesto (Hicks et al., 2015) warns against reducing a researcher's contribution to a single number. Career breaks, discipline-specific norms, and the Matthew effect (Merton, 1968) — where established researchers accumulate citations disproportionately — must all be considered when interpreting author-level metrics.

12.4 Worked example

12.4.1 Retrieving an author entity

We retrieve the OpenAlex profile for a well-known scientometrician.

```
author_entity <- fetch_openalex(
  entity = "authors",
  search = "Ludo Waltman",
  options = list(sort = "cited_by_count:desc")
)

author_info <- author_entity |> slice(1)
cat(glue("Name: {author_info$display_name}\n"))
```

```
#> Name: Ludo Waltman
```

```
cat(glue("OpenAlex ID: {author_info$id}\n"))
```

```
#> OpenAlex ID: https://openalex.org/A5027467000
```

```
cat(glue("ORCID: {author_info$orcid}\n"))
```

```
#> ORCID: https://orcid.org/0000-0001-8249-1752
```

```
cat(glue("Works count: {author_info$works_count}\n"))
```

```
#> Works count: 478
```

```
cat(glue("Cited by count: {author_info$cited_by_count}\n"))
```

```
#> Cited by count: 68387
```

12.4.2 Fetching the author's works

```
author_works <- fetch_openalex(
  entity = "works",
  author.id = author_info$id,
  from_publication_date = "2010-01-01",
```

```
  to_publication_date = "2023-12-31"
)

works_slim <- author_works |>
  select(id, display_name, publication_date, cited_by_count,
    ↪  type, source_display_name) |>
  mutate(year = year(publication_date))

cat(glue("Works retrieved: {nrow(works_slim)}\n"))
```

```
#> Works retrieved: 207
```

12.4.3 Author-level statistics

```
h_idx <- compute_h_index(works_slim$cited_by_count)

stats <- tibble(
  Metric = c("Total works", "Total citations", "h-index",
             "Mean citations/paper", "Median citations/paper",
             "First publication year", "Most recent year"),
  Value = c(
    nrow(works_slim),
    sum(works_slim$cited_by_count),
    h_idx,
    round(mean(works_slim$cited_by_count), 1),
    median(works_slim$cited_by_count),
    min(works_slim$year),
    max(works_slim$year)
  )
)

stats |> gt()
```

Metric	Value
Total works	207
Total citations	35061
h-index	54
Mean citations/paper	169
Median citations/paper	7
First publication year	2010
Most recent year	2023

12.4.4 Publication timeline

```
annual <- works_slim |>
  group_by(year) |>
  summarise(n_pubs = n(), cites = sum(cited_by_count), .groups =
   ↪  "drop") |>
  mutate(cum_cites = cumsum(cites))

ggplot(annual, aes(x = year)) +
  geom_col(aes(y = n_pubs), fill = palette_sci(1), width = 0.7) +
  geom_line(aes(y = cum_cites / max(cum_cites) * max(n_pubs)),
            colour = palette_sci(2)[2], linewidth = 1) +
  scale_y_continuous(
    name = "Publications per year",
    sec.axis = sec_axis(~ . / max(annual$n_pubs) *
     ↪  max(annual$cum_cites),
                        name = "Cumulative citations")
  ) +
  labs(x = "Year") +
  theme_sci()
```

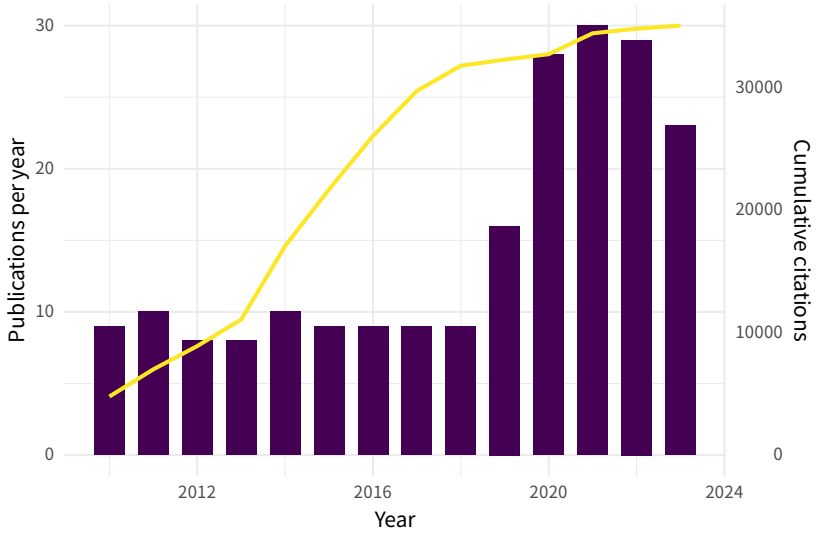


Figure 12.1: Annual publication output and cumulative citation count.

12.4.5 Top venues

```
venue_counts <- works_slim |>
  filter(!is.na(source_display_name)) |>
  count(source_display_name, sort = TRUE) |>
  head(10)

ggplot(venue_counts, aes(x = n, y = reorder(source_display_name,
↪  n))) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Number of articles", y = NULL) +
  theme_sci()
```

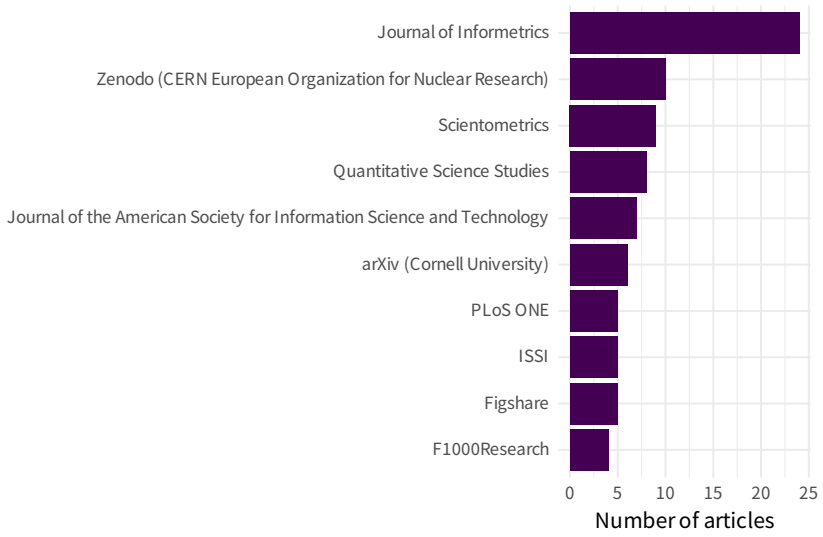


Figure 12.2: Top 10 venues by number of publications for this author.

12.4.6 Checking for name variants

```
if ("display_name_alternatives" %in% names(author_info)) {
  cat("Known name variants:\n")
  print(author_info$display_name_alternatives)
} else {
  cat("No alternative display names recorded in this entity.\n")
}
```

```
#> Known name variants:
#> [[1]]
#>  [1] "L Waltman"             "L. Waltman"            "LUDO
 ↪  WALTMAN"
#>  [4] "Ludo Waltman"          "Ludo Waltman, Waltman" "Ludo
 ↪  Waltmann"
#>  [7] "Waltman L."            "Waltman, L."           "Waltman,
 ↪  L. (Ludo)"
#> [10] "Waltman, L.R."         "Waltman, LR (Ludo)"    "Waltman,
 ↪  Ludo"
#> [13] "Waltmann, Ludo"
```

12.5 Diagnostics and interpretation

When building author profiles, verify the following:

- **Works count plausibility**: Does the number of retrieved works match the author's known output? Large discrepancies suggest a split or merged entity.
- **Topical coherence**: Scan titles and venues. If the works span wildly unrelated fields, the entity may conflate two researchers with similar names.
- **ORCID match**: If the author has an ORCID, cross-check that the OpenAlex entity links to it.
- **Co-author consistency**: The author's frequent co-authors should be recognisable collaborators, not strangers from an unrelated field.
- **Citation distribution**: An extreme outlier (e.g., one paper with 10,000 citations while all others have fewer than 50) may indicate an attribution error.

12.6 Limitations and responsible use

12.7 Limitations and responsible use

- **Disambiguation is imperfect.** OpenAlex's algorithm will sometimes split one person into multiple entities or merge distinct individuals. Always validate profiles manually for high-stakes analyses.
- **ORCID coverage is uneven.** Researchers in the Global South, early-career scholars, and those in humanities adopt ORCID at lower rates. Filtering to ORCID-linked authors introduces bias.

- **The Matthew effect.** Established researchers accumulate citations disproportionately (Merton, 1968). Author-level metrics favour seniority and visibility, not necessarily quality or originality.
- **Career breaks and part-time research.** Indicators like the h-index penalise researchers who take parental leave, work part-time, or switch careers. The m-quotient is only a rough correction (Hirsch, 2005).
- **Never evaluate individuals by numbers alone** (Hicks et al., 2015; American Society for Cell Biology, 2012). Author profiles should supplement, not replace, reading the actual work.

12.8 Common pitfalls

12.9 Common pitfalls

- **Trusting a single database.** OpenAlex may lack works indexed only in domain-specific databases (e.g., SSRN, arXiv preprints not yet linked). Cross-check with ORCID profiles.
- **Ignoring co-author contributions.** Raw publication counts treat single-author and 50-author papers equally. Consider fractional counting for productivity analysis.
- **Comparing across career stages.** An early-career researcher with h = 8 after 5 years may be more productive than a senior researcher with h = 25 after 30 years. Use the m-quotient or age-normalised indicators.
- **Conflating author-level and paper-level metrics.** An author's h-index says nothing about which of their papers are excellent and which are not.

12.10 Exercises

1. **Profile comparison.** Fetch profiles for two researchers in the same field. Compare their h-index, publication count, and mean citations per paper. What does each metric reveal that the others do not?

2. **ORCID validation.** For an author with an ORCID, fetch their works via both OpenAlex author ID and ORCID filtering. Do the two sets agree? What might explain discrepancies?

3. **Name variant detection.** Search OpenAlex for a common name (e.g., "Wang Wei"). How many author entities are returned? What heuristics could you use to identify the correct one?

4. **Fractional productivity.** Recompute annual publication counts using fractional counting (1/k credit per paper with k authors). How does this change the author's productivity curve?
5. **Self-citation ratio.** For an author's works, estimate the fraction of citations that come from the author's own other papers. (Hint: check if cited works share the same author ID.)

12.11 Solutions

Solutions are provided in E.

12.12 Further reading

- Priem et al. (2022) — OpenAlex's author disambiguation and entity resolution approach.
- Hirsch (2005) — The h-index, the most widely used author-level indicator.
- Merton (1968) — The Matthew effect: cumulative advantage in scientific recognition.
- Hicks et al. (2015) — The Leiden Manifesto: principles for responsible author evaluation.
- American Society for Cell Biology (2012) — DORA: recommendations against reducing researchers to single numbers.
- Egghe (2006) — The g-index, a complement to the h-index for author evaluation.

12.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1         systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1              stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
```

```
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0          yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9            shiny_1.15.0           withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Chapter 13

Institutional and Country Analysis

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

13.1 Learning objectives

After completing this chapter, you will be able to:

- Aggregate bibliometric data at institutional and country levels using OpenAlex
- Explain the difference between full counting and fractional counting and when each is appropriate
- Use ROR (Research Organization Registry) identifiers for reliable institutional matching
- Compute descriptive citation summaries and a shared sample/year citation ratio, while explaining why it is not MNCS
- Interpret institutional rankings critically, accounting for size, field mix, and counting method

13.2 Setup

```
library(tidyverse)
library(openalexR)
```

```
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
source(here::here("R", "institutional_samples.R"))
```

13.3 Conceptual background

Comparing research performance across institutions and countries is one of the most common — and most contentious — applications of bibliometrics. University rankings, national R&D assessments, and funding allocation decisions all rely on aggregate bibliometric statistics. Getting the aggregation right matters enormously.

The fundamental choice is between **full counting** and **fractional counting**. Under full counting, each institution listed on a paper receives one full credit for that paper. A paper with authors from five institutions thus generates five institutional publications. Under fractional counting, credit is divided: each institution receives 1/5 of a publication count. Full counting inflates the output of institutions that frequently collaborate, while this fractional rule divides bibliometric credit equally among the listed institutions; it does not measure their actual contribution (Glänzel and Schubert, 2004; Waltman et al., 2011).

ROR (Research Organization Registry) identifiers provide a stable, open system for identifying institutions, analogous to DOIs for publications and ORCID for researchers. OpenAlex maps institutional affiliations to ROR IDs, though coverage varies by discipline and geography. Using ROR avoids the pitfalls of string matching on institution names, which are notoriously inconsistent (e.g., “MIT”, “Massachusetts Institute of Technology”, “Mass. Inst. Tech.”).

Size-independent indicators are essential for fair comparison. Raw publication counts favour large universities. The **Mean Normalised Citation Score** (MNCS) (Waltman et al., 2011) and **PP(top 10%)** — the proportion of papers in the top 10% most cited in their field and year — are designed to account for differences in reference sets. Their interpretation still depends on reference-set construction, coverage, attribution and the citation window. The Leiden Ranking uses these indicators with fractional counting as its default, offering a model of responsible institutional benchmarking.

Country-level analysis faces the same choices. Additionally, linguistic bias in databases means that countries where English is not the primary research language may appear less productive than they truly are (Visser et al., 2021).

13.4 Worked example

13.4.1 Selecting institutions

We use five institutions to demonstrate summaries of equally capped samples. The identifiers below are **OpenAlex institution IDs**, not ROR IDs. Their names, countries and ROR mappings were checked against the OpenAlex institution records on 30 October 2026. Subsidiary coverage still needs checking before an actual assessment.

```
institutions <- tribble(
  ~short_name,  ~openalex_id,
  "MIT",        "I63966007",
  "Oxford",     "I40120149",
  "ETH Zurich", "I35440088",
  "U Tokyo",    "I74801974",
  "U Cape Town", "I157614274"
)
```

13.4.2 Fetching institutional publications

```
fetch_inst_works <- function(inst_id) {
  res <- fetch_openalex(
    entity = "works",
    authorships.institutions.id = inst_id,
    from_publication_date = "2020-01-01",
    to_publication_date = "2022-12-31",
    type = "article",
    options = list(sample = 400, seed = 42)
  )
  if (!is.data.frame(res) || !nrow(res)) {
    stop(glue("No usable works returned for {inst_id}; check the
    ↪  request before comparing institutions."))
  }
  res
}
```

```
inst_data <- institutions |>
  mutate(works = map(openalex_id, fetch_inst_works))
```

These requests retrieve at most 400 sampled articles per institution from 2020–2022, not each institution's complete output. Counts below describe the retrieved samples; the cap prevents interpreting them as comparative institutional productivity. Provider failures stop the example rather than removing an institution silently.

13.4.3 Full counting vs. fractional counting

```
# Equal institutional credit per sampled work: 1 / number of
 ↪  distinct affiliations.
# The helper requires the queried institution to be among those
 ↪  affiliations.
# Incomplete affiliations leave the fractional total missing,
 ↪  with coverage shown.
compute_counts <- sci_institution_sample_counts

counting_comparison <- inst_data |>
  mutate(counts = map2(works, openalex_id, compute_counts)) |>
  unnest(counts) |>
  select(short_name, full_count, affiliation_covered,
   ↪  frac_count) |>
  mutate(ratio = round(frac_count / full_count, 2))

counting_comparison |>
  gt() |>
  cols_label(
    short_name = "Institution",
    full_count = "Sampled works",
    affiliation_covered = "Works with usable affiliations",
    frac_count = "Fractional sample credit",
    ratio = "Frac / Full"
  )
```

Institution	Sampled works	Works with usable affiliations	Fractional sample credit	Frac / Full
MIT	400	344	NA	NA
Oxford	400	325	NA	NA
ETH Zurich	400	365	NA	NA
U Tokyo	400	329	NA	NA
U Cape Town	400	311	NA	NA

13.4.4 Descriptive citation comparison using a shared reference

A separate mean for each institution/year would make its mean citation ratio equal to 1 by construction. Instead, combine the retrieved samples, de-duplicate shared work IDs, and compute a **common mean for each publication year**. For an eligible work, divide its citation count by that pooled year mean, then average those ratios within each institution's sample. Shared works enter the reference once. Conflicting snapshots of a shared work cause an error rather than an arbitrary choice.

This is a descriptive **pooled sample/year ratio**, not field-normalized MNCS. The reference is the union of five capped institutional samples, not all world publications. It adjusts for publication year only; current cumulative citation counts do not impose a common post-publication citation window. Missing counts, missing years, and zero-mean reference years are excluded from the ratio, with its usable denominator reported separately.

```
citation_reference <- sci_pooled_year_reference(inst_data$works)

inst_impact <- inst_data |>
  mutate(summary = map(works, sci_sample_citation_summary,
                       reference = citation_reference)) |>
  unnest(summary) |>
  select(short_name, n_sampled, n_citation_observed, mean_cites,
         median_cites, n_ratio_observed, pooled_year_ratio) |>
  arrange(short_name)

inst_impact |>
  gt() |>
  fmt_number(columns = c(mean_cites, pooled_year_ratio),
   ↪  decimals = 2) |>
  cols_label(
    short_name = "Institution", n_sampled = "Sampled works",
    n_citation_observed = "Observed citation counts",
    mean_cites = "Mean citations", median_cites = "Median
     ↪  citations",
    n_ratio_observed = "Works in ratio",
    pooled_year_ratio = "Pooled sample/year ratio"
  )
```

Institution	Sampled works	Observed citation counts	Mean citations	Median citations	Works in ratio	Pooled sample/year ratio
ETH Zurich	400	400	46.50	23	400	1.01
MIT	400	400	53.52	27	400	1.17

Oxford	400	400	48.72	22	400	1.06
U Cape Town	400	400	36.95	17	400	0.81
U Tokyo	400	400	47.70	12	400	1.05

The helper implementations in R/`institutional_samples.R` make the de-duplication, reference construction and missing-data denominators explicit.

13.4.5 Visualization

```
counting_long <- counting_comparison |>
  pivot_longer(cols = c(full_count, frac_count),
               names_to = "method", values_to = "count") |>
  mutate(method = recode(method,
                         full_count = "Full counting",
                         frac_count = "Fractional counting"))

ggplot(counting_long, aes(x = count, y = reorder(short_name,
↪  count),
                          fill = method)) +
  geom_col(position = "dodge", width = 0.7) +
  scale_fill_manual(values = palette_sci(2)) +
  labs(x = "Credit within the retrieved sample", y = NULL, fill
  ↪  = "Method") +
  theme_sci()
```

```
ggplot(inst_impact, aes(x = pooled_year_ratio, y = short_name))
↪  +
  geom_col(fill = palette_sci(1)) +
  geom_vline(xintercept = 1, linetype = "dashed", colour =
  ↪  "grey50") +
  labs(x = "Citation ratio to pooled sample/year reference", y =
  ↪  NULL) +
  theme_sci()
```

13.5 Diagnostics and interpretation

- **Sample counts are not size**: Capping each sample at 400 articles removes information about institutional output. The full/fractional comparison illustrates counting rules within those samples only.

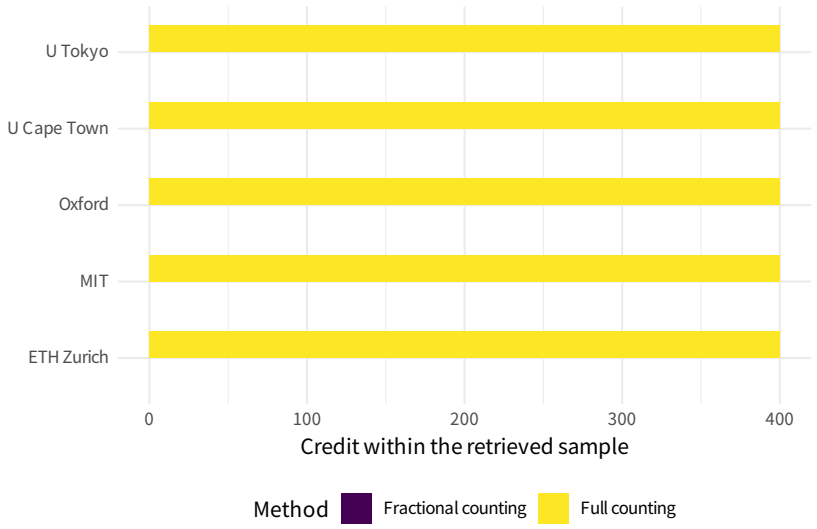


Figure 13.1: Full and fractional institutional credits in the retrieved article samples; these are not institutional output totals.

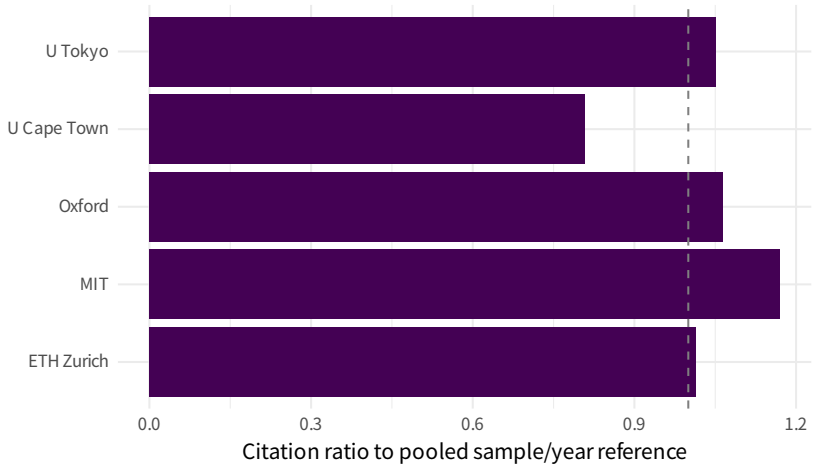


Figure 13.2: Mean citation ratios relative to the shared pooled sample in each publication year. This is not field-normalized MNCS.

- **Field mix**: A medical university will appear more productive (and highly cited) than a social-science institute due to field-level citation norms. True normalisation requires field-level data.
- **Sample vs. population**: The reference depends on which institutions and works were sampled. For a genuine assessment, define the target population and reference set, document inclusion probabilities, and use uncertainty methods appropriate to that design.
- **Fractional/full ratio**: With complete affiliations, the ratio summarizes how institutional credit is divided in the retrieved works. Missing affiliations leave the fractional total unavailable; they are not treated as single-institution papers.

13.6 Limitations and responsible use

13.7 Limitations and responsible use

- **Rankings oversimplify.** Institutional and country rankings compress a multidimensional reality into a single number. The Leiden Manifesto (Hicks et al., 2015) warns against using rankings mechanically.
- **Affiliation data quality.** Not all papers have complete or correct institutional affiliations. OpenAlex coverage depends on publisher metadata quality.
- **Counting method changes conclusions.** The same data can yield different institutional rankings depending on whether full or fractional counting is used. Always report the method.
- **Language and geographic bias.** Databases undercount research published in non-English languages, disadvantaging countries in Latin America, Asia, and Africa (Visser et al., 2021; Singh et al., 2021).
- **Do not compare across fields without normalisation.** Raw citation differences can reflect disciplinary citation practices as well as other differences; the sample/year ratio does not remove field mix.

13.8 Common pitfalls

13.9 Common pitfalls

- **Using full counting when fractional is more appropriate.** Full counting double-credits collaborative work, inflating output for institutions with many international partnerships.

- **Ignoring field normalisation.** The pooled sample/year ratio adjusts for year only. It must not be relabeled as MNCS or used as a field-adjusted ranking.
- **String-matching institution names.** Never match institutions by name strings alone. Use ROR or OpenAlex institution IDs.
- **Treating samples as populations.** If you sampled 400 works per institution, the resulting metrics have uncertainty. Report this.
- **Forgetting subsidiary institutions.** Large universities may have hospitals, research institutes, or campuses with separate ROR IDs. Decide on the unit of analysis upfront.

13.10 Exercises

1. **Country-level analysis.** Aggregate publications by author country (using `authorships.countries`) instead of institution. Compute full and fractional counts for the top 10 countries in your corpus. Which countries are most affected by the counting method?

2. **Top-decile sample share.** Using the same de-duplicated pooled year reference, define a top-decile threshold and an explicit rule for ties. Compute each sample's share above that threshold. Explain why this is not the field-normalized PP(top 10%) indicator.

3. **Collaboration intensity.** Compute the mean number of distinct institutional affiliations per paper for each institution. Is there a relationship between collaboration intensity and the full/fractional count ratio?

13.11 Solutions

Solutions are provided in E.

13.12 Further reading

- Waltman et al. (2011) — MNCS and size-independent indicators for institutional comparison.
- Glänzel and Schubert (2004) — Fractional counting in co-authorship and institutional analysis.
- Hicks et al. (2015) — The Leiden Manifesto on responsible institutional benchmarking.

- Waltman (2016) — Review of citation impact indicators applicable to institutions and countries.
- Visser et al. (2021) — Database coverage comparison; crucial context for country-level analysis.
- Singh et al. (2021) — Journal coverage across databases, highlighting geographic imbalances.

13.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
```

```
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1             stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0         yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9            shiny_1.15.0           withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
```

```
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Chapter 14

Temporal Analysis and Citation Aging

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

14.1 Learning objectives

After completing this chapter, you will be able to:

- Define and apply citation windows of different lengths
- Compute and interpret the citation half-life for a journal or corpus
- Plot citation aging curves showing how citations accumulate over time
- Calculate the Price index (proportion of references less than five years old)
- Analyse temporal trends in publication output and citation rates
- Explain how aging patterns differ across disciplines

14.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

14.3 Conceptual background

Citations are not evenly distributed over time. Most papers receive the bulk of their citations within a few years of publication, then gradually fade. This phenomenon — **citation aging** or **obsolescence** — was first studied systematically by de Solla Price (1963), who observed that scientific literature has a characteristic "half-life": the time by which half of all citations to a given year's output have been received.

The **citation half-life** varies dramatically across disciplines. In fast-moving fields like molecular biology, half-lives may be three to four years; in mathematics or humanities, they can exceed ten years. This has direct consequences for metrics: a two-year Journal Impact Factor systematically undervalues journals in slow-aging fields (Waltman, 2016).

A **citation window** is the fixed period after publication during which citations are counted. The JIF uses a two-year window; CiteScore uses four years. Choosing the window involves a trade-off: short windows favour fast-aging fields but miss delayed-recognition papers; long windows are more inclusive but dilute the signal of recent trends.

The **Price index** — the proportion of references in a paper's bibliography that are less than five years old — measures how rapidly a field consumes recent literature. A high Price index indicates a fast-moving frontier field; a low Price index suggests reliance on older foundational works. de Solla Price (1963) introduced this as an "immediacy" measure.

Sleeping beauties — papers that receive few citations for years then suddenly attract attention (Kleinberg, 2003) — violate the standard aging model. Their existence is a reminder that citation half-lives describe averages, not individual papers. Garfield (1955) already noted that some ideas take decades to be recognised.

Understanding citation temporality is essential for fair evaluation. Comparing citation counts without accounting for publication year, citation window, and field-specific aging rates produces meaningless results.

14.4 Worked example

14.4.1 Acquiring the data

We fetch works from two journals with different expected aging profiles.

```
# Scientometrics (social science, moderate aging)
works_sciento <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2013-01-01",
  to_publication_date = "2018-12-31",
  type = "article",
  options = list(sample = 400, seed = 42)
)

# PLOS ONE (multidisciplinary, faster aging)
works_plos <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S202381698",
  from_publication_date = "2013-01-01",
  to_publication_date = "2018-12-31",
  type = "article",
  options = list(sample = 400, seed = 42)
)

works_all <- bind_rows(
  works_sciento |> mutate(journal = "Scientometrics"),
  works_plos |> mutate(journal = "PLOS ONE")
) |>
  mutate(year = year(publication_date),
         age = 2024 - year)
```

14.4.2 Citation aging curves

We compute mean citations by age (years since publication) for each journal.

```
aging <- works_all |>
  group_by(journal, age) |>
  summarise(
    mean_cites = mean(cited_by_count, na.rm = TRUE),
    median_cites = median(cited_by_count, na.rm = TRUE),
    n = n(),
    .groups = "drop"
  ) |>
```

```
  arrange(journal, age)
```

```
ggplot(aging, aes(x = age, y = mean_cites, colour = journal)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  scale_colour_manual(values = palette_sci(2)) +
  labs(x = "Years since publication", y = "Mean citations",
       colour = "Journal") +
  theme_sci()
```

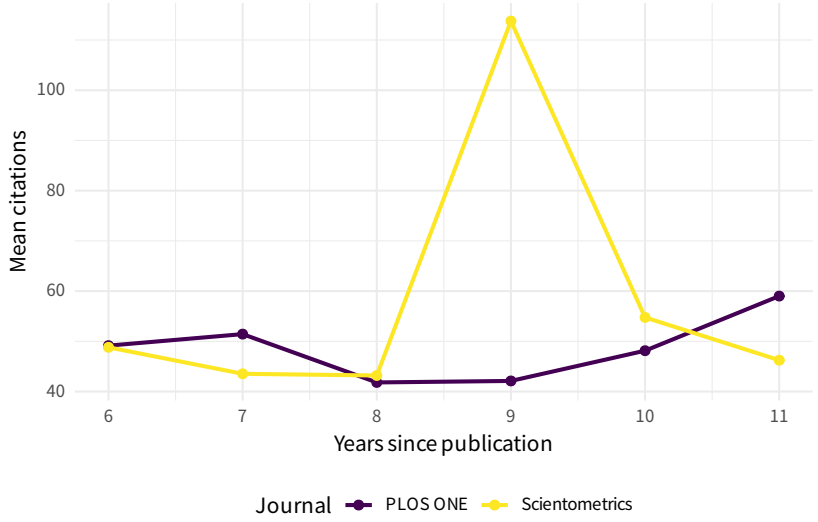


Figure 14.1: Citation aging curves for Scientometrics and PLOS ONE articles published 2013-2018.

14.4.3 Estimating citation half-life

We estimate the half-life as the age at which cumulative citations reach 50% of total.

```
half_life <- works_all |>
  group_by(journal) |>
  arrange(age) |>
  mutate(
    cum_cites = cumsum(cited_by_count),
    total_cites = sum(cited_by_count),
    cum_pct = cum_cites / total_cites
```

```
  ) |>
  summarise(
    total_cites = first(total_cites),
    half_life_age = min(age[cum_pct >= 0.50]),
    .groups = "drop"
  )

half_life |>
  gt() |>
  cols_label(
    journal = "Journal",
    total_cites = "Total citations",
    half_life_age = "Half-life (years)"
  )
```

Journal	Total citations	Half-life (years)
PLOS ONE	19581	9
Scientometrics	22651	9

14.4.4 Price index

The Price index is the fraction of references less than five years old. A full computation requires publication years of each reference; here we show reference-list lengths as a starting point.

```
works_all |>
  filter(!is.na(referenced_works_count), referenced_works_count
  ↪  > 0) |>
  group_by(journal) |>
  summarise(mean_refs = round(mean(referenced_works_count), 1),
  ↪  .groups = "drop")
```

```
#> # A tibble: 2 x 2
#>   journal        mean_refs
#>   <chr>              <dbl>
#> 1 PLOS ONE            47.1
#> 2 Scientometrics      42.3
```

14.4.5 Cumulative citation trajectories

```
cum_aging <- works_all |>
  group_by(journal, age) |>
```

```
  summarise(total = sum(cited_by_count), .groups = "drop") |>
  group_by(journal) |>
  arrange(age) |>
  mutate(
    cum_total = cumsum(total),
    cum_pct = cum_total / sum(total)
  )

ggplot(cum_aging, aes(x = age, y = cum_pct, colour = journal)) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = 0.5, linetype = "dashed", colour =
  ↪  "grey50") +
  scale_y_continuous(labels = scales::percent) +
  scale_colour_manual(values = palette_sci(2)) +
  labs(x = "Years since publication", y = "Cumulative % of
  ↪  citations",
       colour = "Journal") +
  annotate("text", x = 9, y = 0.52, label = "50% threshold",
           colour = "grey40", size = 3.5) +
  theme_sci()
```



Figure 14.2: Cumulative citation percentage by years since publication.

14.5 Diagnostics and interpretation

- **Sample size per age bin**: Ensure each year has enough papers for stable estimates. Small cohorts produce noisy aging curves.
- **Truncation**: Papers published in the most recent year have not had time to accumulate citations. Exclude or flag incomplete citation windows.
- **Disciplinary benchmarks**: Compare your half-life estimates against known benchmarks. Social sciences typically have half-lives of 5-8 years; biomedical fields 3-5 years; mathematics 8-12 years.
- **Outliers**: A few extremely highly cited papers can distort mean-based aging curves. Compare mean and median trajectories.

14.6 Limitations and responsible use

14.7 Limitations and responsible use

- **Half-life is an average.** Individual papers deviate wildly from the average aging pattern. Sleeping beauties and flash-in-the-pan papers both exist.
- **Database lag.** Citation counts in OpenAlex may lag behind reality by weeks or months. Very recent citation data is unreliable.
- **Survivor bias.** Retracted papers or papers removed from databases disappear from aging analyses, potentially biasing half-life estimates.
- **Citation inflation.** The total volume of citations has grown over time. A paper published in 2020 will accumulate citations faster than an identical paper published in 2000, simply because there are more papers being published (de Solla Price, 1963).
- **Do not penalise slow-aging fields.** Using short citation windows systematically undervalues mathematics, humanities, and theoretical disciplines (Hicks et al., 2015).

14.8 Common pitfalls

14.9 Common pitfalls

- **Mixing citation windows.** Comparing a 2-year JIF for one journal with lifetime citations for another produces meaningless rankings.

- **Ignoring publication-year effects.** A corpus spanning 2000-2023 includes papers with 23 years of citation history alongside papers with 1 year. Normalise by publication year.
- **Confusing obsolescence with irrelevance.** A declining citation curve does not mean the knowledge is wrong or useless; it may mean the ideas have been absorbed into textbooks or standard practice.
- **Forgetting about the denominator.** If the number of publications in a field is growing, the absolute citation count grows even if the per-paper impact is constant.

14.10 Exercises

1. **Half-life comparison.** Fetch works from a mathematics journal and a biomedical journal. Compute and compare their citation half-lives. Does the difference match disciplinary expectations?
2. **Price index computation.** For a set of papers with known reference lists, compute the actual Price index (fraction of references < 5 years old). (Hint: you will need to fetch metadata for the referenced works.)
3. **Sleeping beauty detection.** Identify papers that received fewer than 2 citations in their first 3 years but more than 20 total. What are these papers about?
4. **Median vs. mean aging.** Replot the aging curves using median instead of mean citations. How does the shape change, and why?
5. **Publication-cohort sensitivity.** Using the fetched Scientometrics sample, compare current cumulative citation summaries for the 2-, 3- and 5-year publication cohorts ending in 2018. Report observed denominators and explain why changing the publication cohort does not establish a citation-receipt window or compute JIF.

14.11 Solutions

Solutions are provided in E.

14.12 Further reading

- de Solla Price (1963) — Little Science, Big Science; introduced citation aging and the Price index.

- Garfield (1955) — The origin of citation indexing, including early observations on citation temporality.
- Kleinberg (2003) — Bursty patterns in temporal data; relevant to sleeping beauties and sudden citation surges.
- Waltman (2016) — Review of citation indicators, including discussion of citation windows and aging.
- Hicks et al. (2015) — The Leiden Manifesto; warns against penalising slow-aging fields with short citation windows.

14.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
```

```
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>  [1] gridExtra_2.3.1       httr2_1.3.0           readxl_1.6.0
#>  [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>  [7] compiler_4.6.1        systemfonts_1.3.4     vctrs_0.7.3
#> [10] httpcode_0.3.0         pkgconfig_2.0.3
 ↪  fastmap_1.2.0
#> [13] labeling_0.4.3         ca_0.72                utf8_1.2.6
#> [16] promises_1.5.0         rmarkdown_2.32         tzdb_0.5.0
#> [19] bit_4.6.0              xfun_0.62
 ↪  jsonlite_2.0.0
#> [22] SnowballC_0.7.1        later_1.4.8
 ↪  parallel_4.6.1
#> [25] stopwords_2.3         R6_2.6.1             stringi_1.8.9
#> [28] RColorBrewer_1.1-3     cellranger_1.1.0
 ↪  assertthat_0.2.1
#> [31] Rcpp_1.1.2             bookdown_0.48          knitr_1.52
#> [34] triebeard_0.4.1        base64enc_0.1-6
 ↪  httpuv_1.6.17
#> [37] rentrez_1.2.4         Matrix_1.7-5          igraph_2.3.4
#> [40] timechange_0.4.0       tidyselect_1.2.1
 ↪  stringdist_0.9.17
#> [43] rstudioapi_0.19.0     pubmedR_1.1.0          yaml_2.3.12
#> [46] viridis_0.6.5          miniUI_0.1.2
 ↪  humaniformat_0.6.0
#> [49] curl_8.1.0             qpdf_1.4.1
 ↪  lattice_0.22-9
#> [52] plyr_1.8.9            shiny_1.15.0           withr_3.0.3
#> [55] S7_1.0.0               askpass_1.2.1
 ↪  evaluate_1.0.5
#> [58] zip_3.0.2              xml2_1.6.0
 ↪  shinycssloaders_1.1.0
#> [61] pillar_1.11.1          janeaustenr_1.0.0      DT_0.34.0
#> [64] plotly_4.12.1          generics_0.1.4
 ↪  rprojroot_2.1.1
#> [67] hms_1.1.4             scales_1.4.0          xtable_1.8-8
#> [70] contentanalysis_1.1.1  tools_4.6.1
 ↪  data.table_1.18.6.1
#> [73] tokenizers_0.3.0       openxlsx_4.2.9         fs_2.1.0
#> [76] visNetwork_2.1.4       XML_3.99-0.25
 ↪  fastmatch_1.1-8
#> [79] grid_4.6.1             bibtex_0.5.3
 ↪  urltools_1.7.3.1
```

```
#> [82] rscopus_0.9.0          dimensionsR_0.0.3
 ↪  bibliometrixData_0.3.0
#> [85] cli_3.6.6             viridisLite_0.4.3     gtable_0.3.6
#> [88] digest_0.6.39          ggrepel_0.9.8          crul_1.6.0
#> [91] htmlwidgets_1.6.4      farver_2.1.2
 ↪  htmltools_0.5.9
#> [94] lifecycle_1.0.5        httr_1.4.9             here_1.0.2
#> [97] mime_0.13              bit64_4.8.6
```

Part IV

Network & Mapping Methods

Chapter 15

Co-authorship Networks

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

15.1 Learning objectives

After completing this chapter, you will be able to:

- Construct a co-authorship network from bibliographic data using `igraph`
- Compute and interpret node-level metrics: degree, betweenness centrality, and clustering coefficient
- Apply Louvain and Leiden community detection algorithms and compare results
- Produce deterministic network layouts that render identically in HTML and PDF
- Identify central authors and structural holes in a collaboration network

15.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
library(ggraph)
```

```
library(glue)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

15.3 Conceptual background

Co-authorship is one of the most visible and well-studied forms of scientific collaboration. When two researchers appear together on a publication, they form a tie in the co-authorship network. Studying these networks reveals the social structure of science: who collaborates with whom, which groups are isolated, and which individuals serve as bridges between communities (Newman, 2001).

Co-authorship networks are typically modelled as **undirected, weighted graphs**. Nodes represent authors, and an edge connects two authors who have co-authored at least one paper. Edge weights reflect the number of co-authored papers (or, alternatively, fractional counts inversely proportional to the number of authors per paper). These networks tend to exhibit properties common to social networks: a heavy-tailed degree distribution (most authors have few collaborators, a few have many), high clustering (collaborators of collaborators tend to collaborate), and a small-world structure (Barabási et al., 2002).

Community detection identifies groups of authors who collaborate more densely among themselves than with the rest of the network. The Louvain algorithm (Blondel et al., 2008) is fast and widely used; the Leiden algorithm (Traag et al., 2019) provides theoretical guarantees that all communities are well-connected, avoiding the "resolution limit" problem that can afflict Louvain.

Network analysis of co-authorship data has been widely applied to map research fields (Glänzel and Schubert, 2004), identify key opinion leaders, assess interdisciplinary collaboration, and study the evolution of research communities over time. Fortunato (2010) provides a comprehensive review of community detection methods applicable to these networks.

15.4 Worked example

15.4.1 Acquiring co-authorship data

We fetch works from a journal and extract the author pairs.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2021-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 300, seed = 42)
)

nrow(works)
```

```
#> [1] 300
```

15.4.2 Extracting author pairs

Each work has a list-column of authors. We unnest and build all pairwise co-author edges.

```
author_data <- works |>
  select(id, authorships) |>
  unnest(authorships, names_sep = "_") |>
  select(work_id = id, author_id = authorships_id, author_name =
   ↪  authorships_display_name) |>
  filter(!is.na(author_id))

edges <- author_data |>
  inner_join(author_data, by = "work_id", suffix = c("_1", "_2"),
             relationship = "many-to-many") |>
  filter(author_id_1 < author_id_2) |>
  count(author_id_1, author_id_2,
        name_1 = author_name_1, name_2 = author_name_2,
        name = "weight")

cat(glue("Authors: {n_distinct(c(edges$author_id_1,
 ↪  edges$author_id_2))}\n"))
```

```
#> Authors: 653
```

```
cat(glue("Edges: {nrow(edges)}\n"))
```

```
#> Edges: 881
```

15.4.3 Building the network

```
g <- graph_from_data_frame(
  edges |> select(author_id_1, author_id_2, weight),
  directed = FALSE
)

g <- simplify(g, edge.attr.comb = list(weight = "sum"))

cat(glue("Nodes: {vcount(g)}, Edges: {ecount(g)}\n"))
```

```
#> Nodes: 653, Edges: 881
```

```
cat(glue("Components: {components(g)$no}\n"))
```

```
#> Components: 187
```

```
cat(glue("Density: {round(graph.density(g), 4)}\n"))
```

```
#> Density: 0.0041
```

15.4.4 Network metrics

```
V(g)$degree <- degree(g)
V(g)$betweenness <- betweenness(g, normalized = TRUE)
V(g)$clustering <- transitivity(g, type = "local")

author_lookup <- author_data |>
  distinct(author_id, author_name)

metrics <- tibble(
  author_id = V(g)$name,
  degree = V(g)$degree,
  betweenness = V(g)$betweenness,
  clustering = V(g)$clustering
```

```
) |>
  left_join(author_lookup, by = "author_id") |>
  arrange(desc(degree))

metrics |>
  head(10) |>
  select(author_name, degree, betweenness, clustering)
```

```
#> # A tibble: 10 x 4
#>    author_name     degree betweenness clustering
#>    <chr>            <dbl>       <dbl>      <dbl>
#>  1 Silvio Peroni       10 0.000127         0.4
#>  2 Lin Zhang           10 0.000276         0.378
#>  3 Tom Hanika           9 0.0000283        0.639
#>  4 Ying Huang           8 0.000137         0.5
#>  5 Gerd Stumme          8 0.0000283        0.786
#>  6 Haihua Chen          8 0.0000565        0.571
#>  7 Tobias Koopmann      7 0.000000673      1
#>  8 Michael París        7 0.000000673      1
#>  9 Robert Jäschke       7 0.000000673      1
#> 10 Guido Buenstorf      7 0.000000673      1
```

15.4.5 Community detection

We apply both Louvain and Leiden to the largest connected component.

```
comp <- components(g)
giant <- induced_subgraph(g, which(comp$membership ==
 ↪  which.max(comp$csize)))

louvain <- cluster_louvain(giant)
V(giant)$community_louvain <- membership(louvain)

cat(glue("Louvain communities:
 ↪  {length(unique(membership(louvain)))}\n"))
```

```
#> Louvain communities: 4
```

```
cat(glue("Louvain modularity: {round(modularity(louvain),
 ↪  3)}\n"))
```

```
#> Louvain modularity: 0.383
```

```
leiden_result <- cluster_leiden(giant, resolution_parameter =
↪  1.0,
                                objective_function = "modularity")
V(giant)$community_leiden <- membership(leiden_result)

cat(glue("Leiden communities:
↪  {length(unique(membership(leiden_result)))}\n"))
```

```
#> Leiden communities: 4
```

```
cat(glue("Leiden modularity: {round(modularity(giant,
↪  membership(leiden_result)), 3)}\n"))
```

```
#> Leiden modularity: 0.383
```

15.4.6 Comparing community assignments

```
comparison <- tibble(
  louvain = V(giant)$community_louvain,
  leiden = V(giant)$community_leiden
)

nmi <- igraph::compare(
  V(giant)$community_louvain,
  V(giant)$community_leiden,
  method = "nmi"
)
cat(glue("Normalized Mutual Information (Louvain vs Leiden):
↪  {round(nmi, 3)}\n"))
```

```
#> Normalized Mutual Information (Louvain vs Leiden): 1
```

15.4.7 Visualization

We use a Fruchterman-Reingold layout with a fixed seed for deterministic output.

```
giant_tidy <- as_tbl_graph(giant) |>
  mutate(
    community = as.factor(community_leiden),
    label = ifelse(degree > quantile(degree, 0.95),
```

```
                    author_lookup$author_name[match(name,
↪  author_lookup$author_id)],
                    NA_character_)
  )

set.seed(42)
layout <- create_layout(giant_tidy, layout = "fr")

ggraph(layout) +
  geom_edge_link(alpha = 0.15, colour = "grey60") +
  geom_node_point(aes(size = degree, colour = community), alpha
  ↪  = 0.8) +
  geom_node_text(aes(label = label), repel = TRUE, size = 2.5,
                 max.overlaps = 15, na.rm = TRUE) +
  scale_size_continuous(range = c(1, 6), guide = "none") +
  scale_colour_manual(values = palette_sci(
    n_distinct(giant_tidy |> pull(community), na.rm = TRUE)
  )) +
  labs(colour = "Community") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```

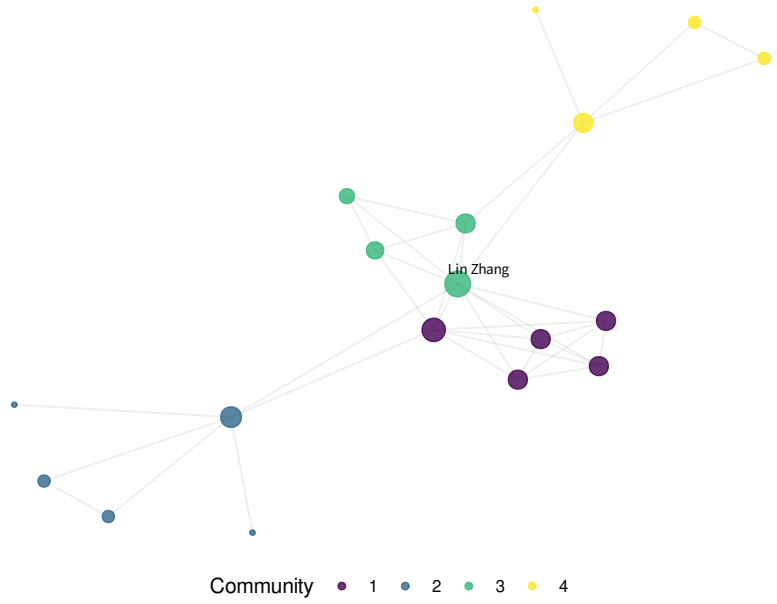


Figure 15.1: Co-authorship network of sampled Scientometrics articles (2021–2023), coloured by Leiden community.

```
tibble(degree = degree(giant)) |>
  ggplot(aes(x = degree)) +
  geom_histogram(binwidth = 1, fill = palette_sci(1), colour =
  ↪  "white") +
  labs(x = "Degree (number of co-authors)", y = "Count") +
  theme_sci()
```

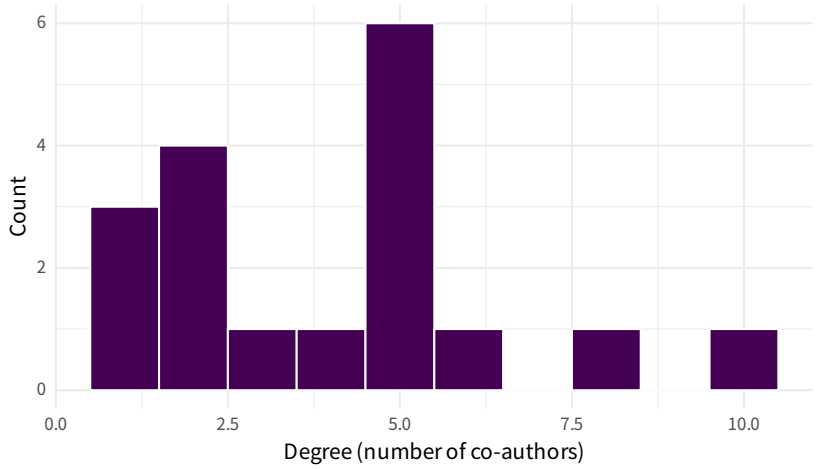


Figure 15.2: Degree distribution of the co-authorship network.

15.5 Diagnostics and interpretation

Key diagnostics for co-authorship networks:

- **Giant component ratio**: What fraction of nodes are in the largest connected component? A fragmented network may indicate a niche field or a sparse sample.
- **Degree distribution**: The heavy-tailed distribution is expected. A network where all nodes have similar degree may indicate a problem with the data (e.g., only single-author papers were excluded).
- **Modularity**: Values above 0.3 typically indicate meaningful community structure. Modularity near zero suggests the network has no clear communities.
- **Comparison of algorithms**: If Louvain and Leiden produce very different community counts, investigate the resolution parameter. High NMI (>0.8) indicates strong agreement.

```
giant_ratio <- vcount(giant) / vcount(g)
cat(glue("Giant component: {vcount(giant)}/{vcount(g)} nodes
 ↪  ({scales::percent(giant_ratio)})\n"))
```

```
#> Giant component: 18/653 nodes (3%)
```

```
cat(glue("Mean degree: {round(mean(degree(giant)), 1)}\n"))
```

```
#> Mean degree: 4
```

```
cat(glue("Transitivity (global): {round(transitivity(giant, type
 ↪  = 'global'), 3)}\n"))
```

```
#> Transitivity (global): 0.578
```

15.6 Limitations and responsible use

15.7 Limitations and responsible use

- **Co-authorship ≠ collaboration.** Not all collaborations result in co-authored publications, and not all co-authorships reflect genuine intellectual contribution (honorary authorship).
- **Name ambiguity.** Author names in bibliographic databases are not unique identifiers. Despite OpenAlex's entity resolution, errors persist — especially for common names and transliterated names. Always inspect suspicious high-degree nodes.
- **Fractional vs. full counting.** Our construction counts each co-authorship edge equally regardless of the number of authors on a paper. Fractional counting (weight = 1/(n-1) per edge) gives less weight to papers with many authors and may be more appropriate for some analyses (Glänzel and Schubert, 2004).
- **Temporal dynamics.** Static networks aggregate all collaborations into a single snapshot. Consider building time-sliced networks to study how collaboration patterns evolve.
- **Causality.** Network position (e.g., high betweenness) does not imply leadership or influence. It is a structural observation, not an evaluative judgement (Hicks et al., 2015).

15.8 Common pitfalls

15.9 Common pitfalls

- **Non-deterministic layouts.** Force-directed layouts (Fruchterman-Reingold, Kamada-Kawai) depend on a random seed. Always `set.seed()` before computing the layout, or the figure will change on every render.
- **Including the full network.** Very large networks are unreadable. Filter to the giant component, top-N nodes, or a specific community before plotting.
- **Ignoring multi-edges and self-loops.** `igraph::simplify()` is essential after constructing the graph from edge lists that may contain duplicate pairs.
- **Comparing modularity across networks of different sizes.** Modularity is sensitive to network size and density. Use the Normalized Mutual Information (NMI) to compare community structures.
- **Overinterpreting small components.** A component with 2–3 authors who co-authored one paper is not a "community" — it is an artifact of the sample.

15.10 Exercises

1. **Fractional counting.** Modify the edge construction to use fractional counting (weight = 1/(k-1) where k is the number of authors on the paper). How do the top-degree authors change?

2. **Temporal slicing.** Split the corpus into yearly slices and build a separate network for each year. How does the number of components and mean degree evolve over time?

3. **Betweenness vs. degree.** Plot betweenness centrality against degree for the giant component. Are the highest-degree authors also the highest-betweenness authors? What does a high-betweenness, low-degree node represent?

4. **Resolution parameter.** Run Leiden community detection with resolution parameters of 0.5, 1.0, and 2.0. How does the number of communities change? Which resolution produces the most interpretable results?

15.11 Solutions

Solutions are provided in E.

15.12 Further reading

- Newman (2001) — Foundational analysis of scientific collaboration networks.
- Barabási et al. (2002) — Evolution of co-authorship networks and their scale-free properties.
- Blondel et al. (2008) — The Louvain community detection algorithm.
- Traag et al. (2019) — The Leiden algorithm, with guarantees on well-connected communities.
- Fortunato (2010) — Comprehensive review of community detection in graphs.
- Glänzel and Schubert (2004) — Co-authorship analysis and fractional counting in bibliometrics.

15.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
```

```
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] ggraph_2.2.2       tidygraph_1.3.1    igraph_2.3.4
 ↪  quanteda_4.5.1
#>  [5] pdftools_3.9.1     arrow_26.0.1.1     bibliometrix_5.5.0
 ↪  RefManageR_1.4.0
#>  [9] bib2df_1.1.2.0     rcrossref_1.2.1    gt_1.4.0
 ↪  tidytext_0.4.3
#> [13] glue_1.8.1         openalexR_3.1.2    lubridate_1.9.5
 ↪  forcats_1.0.1
#> [17] stringr_1.6.0      dplyr_1.2.1        purrr_1.2.2
 ↪  readr_2.2.0
#> [21] tidyr_1.3.2        tibble_3.3.1       ggplot2_4.0.4
 ↪  tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32        fs_2.1.0              vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9       contentanalysis_1.1.1  curl_8.1.0
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9            httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  shinycssloaders_1.1.0
#>  [34] rprojroot_2.1.1        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [37] urltools_1.7.3.1      timechange_0.4.0      httr_1.4.9
#>  [40] polyclip_1.10-7       compiler_4.6.1        here_1.0.2
#>  [43] bit64_4.8.6            withr_3.0.3           S7_1.0.0
#>  [46] viridis_0.6.5         ggforce_0.5.0         MASS_7.3-65
#>  [49] bibliometrixData_0.3.0 tools_4.6.1           otel_0.3.0
#>  [52] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
```

```
#>  [55] httpuv_1.6.17          promises_1.5.0         grid_4.6.1
#>  [58] stringdist_0.9.17      generics_0.1.4
 ↪  gtable_0.3.6
#>  [61] tzdb_0.5.0             rscopus_0.9.0           ca_0.72
#>  [64] data.table_1.18.6.1    hms_1.1.4              xml2_1.6.0
#>  [67] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [70] later_1.4.8            tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [73] bit_4.6.0              tidyselect_1.2.1
 ↪  miniUI_0.1.2
#>  [76] knitr_1.52             gridExtra_2.3.1
 ↪  bookdown_0.48
#>  [79] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [82] DT_0.34.0              humaniformat_0.6.0
 ↪  visNetwork_2.1.4
#>  [85] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#>  [88] evaluate_1.0.5         httpcode_0.3.0           cli_3.6.6
#>  [91] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#>  [94] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#>  [97] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [100] viridisLite_0.4.3      scales_1.4.0
 ↪  openxlsx_4.2.9
#> [103] rlang_1.3.0            fastmatch_1.1-8
```

Chapter 16

Co-citation and Bibliographic Coupling

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

16.1 Learning objectives

After completing this chapter, you will be able to:

- Distinguish co-citation analysis from bibliographic coupling and explain when each is appropriate
- Construct co-citation and bibliographic coupling networks from OpenAlex reference data
- Compute normalised similarity measures (cosine, Jaccard) for citation-based linkages
- Identify core documents and intellectual clusters in a research field
- Interpret co-citation maps as representations of intellectual structure

16.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
```

```
library(ggraph)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

16.3 Conceptual background

While co-authorship networks (15.3) reveal social structure, **co-citation** and **bibliographic coupling** networks reveal intellectual structure — the conceptual relationships between documents as expressed through citation behaviour.

Co-citation analysis, introduced independently by Small (1973) and Marshakova-Shaikevich in 1973, measures the relatedness of two documents by how often they are cited together by other papers. If documents A and B frequently appear in the same reference lists, they likely address related topics or represent complementary contributions to a field. Co-citation strength changes over time: as a field evolves, new citing papers create new co-citation links and strengthen or weaken existing ones.

Bibliographic coupling, proposed by Kessler (1963), takes the complementary perspective. Two documents are bibliographically coupled if they share one or more references. Unlike co-citation, coupling strength is fixed at publication time — once a paper is published, its reference list does not change. This makes bibliographic coupling especially useful for analysing the most recent literature, where co-citation links have not yet had time to form.

The two methods reveal different facets of intellectual structure. Co-citation maps the reception of ideas (how the community groups prior work); bibliographic coupling maps the production of knowledge (how authors draw on shared foundations). Waltman et al. (2010) demonstrated that combining both perspectives yields richer maps of science than either alone.

Both methods produce **similarity matrices** that can be converted to networks. Common normalisation approaches include cosine similarity (emphasising relative overlap) and the Jaccard index (penalising pairs with many unshared references). Unnormalised raw counts are dominated by highly-cited papers; normalisation is essential for balanced network construction.

16.4 Worked example

16.4.1 Fetching citing and cited works

We fetch a corpus of scientometrics research and extract reference lists.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2020-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 200, seed = 42)
)

refs <- works |>
  select(citing_id = id, referenced_works) |>
  unnest(referenced_works) |>
  rename(cited_id = referenced_works)

cat(glue("Citing works: {n_distinct(refs$citing_id)}\n"))
```

```
#> Citing works: 200
```

```
cat(glue("Cited works: {n_distinct(refs$cited_id)}\n"))
```

```
#> Cited works: 8765
```

```
cat(glue("Citation links: {nrow(refs)}\n"))
```

```
#> Citation links: 10144
```

16.4.2 Building a co-citation network

Two cited documents are co-cited when they appear in the same reference list. We count shared citing papers.

```
cocit_pairs <- refs |>
  inner_join(refs, by = "citing_id", suffix = c("_a", "_b"),
             relationship = "many-to-many") |>
  filter(cited_id_a < cited_id_b) |>
  count(cited_id_a, cited_id_b, name = "cocit_count")

cocit_top <- cocit_pairs |>
```

```
  filter(cocit_count >= 3)

cat(glue("Co-citation pairs (count >= 3): {nrow(cocit_top)}\n"))
```

```
#> Co-citation pairs (count >= 3): 123
```

```
g_cocit <- graph_from_data_frame(
  cocit_top |> select(cited_id_a, cited_id_b, weight =
 ↪  cocit_count),
  directed = FALSE
) |>
  simplify(edge.attr.comb = list(weight = "sum"))

cat(glue("Co-citation network: {vcount(g_cocit)} nodes,
 ↪  {ecount(g_cocit)} edges\n"))
```

```
#> Co-citation network: 116 nodes, 123 edges
```

16.4.3 Building a bibliographic coupling network

Two citing documents are coupled when they share at least one reference.

```
bibcoup_pairs <- refs |>
  inner_join(refs, by = "cited_id", suffix = c("_a", "_b"),
             relationship = "many-to-many") |>
  filter(citing_id_a < citing_id_b) |>
  count(citing_id_a, citing_id_b, name = "shared_refs")

bibcoup_top <- bibcoup_pairs |>
  filter(shared_refs >= 5)

cat(glue("Bibliographic coupling pairs (shared refs >= 5):
 ↪  {nrow(bibcoup_top)}\n"))
```

```
#> Bibliographic coupling pairs (shared refs >= 5): 64
```

```
g_bibcoup <- graph_from_data_frame(
  bibcoup_top |> select(citing_id_a, citing_id_b, weight =
 ↪  shared_refs),
  directed = FALSE
) |>
  simplify(edge.attr.comb = list(weight = "sum"))

cat(glue("Coupling network: {vcount(g_bibcoup)} nodes,
 ↪  {ecount(g_bibcoup)} edges\n"))
```

```
#> Coupling network: 80 nodes, 64 edges
```

16.4.4 Visualisation

```
comp <- components(g_cocit)
giant <- induced_subgraph(g_cocit, which(comp$membership ==
 ↪  which.max(comp$csize)))
V(giant)$community <- as.factor(membership(cluster_leiden(
  giant, resolution_parameter = 1.0, objective_function =
 ↪  "modularity"
)))
V(giant)$degree <- degree(giant)

set.seed(42)
layout <- create_layout(as_tbl_graph(giant), layout = "fr")

ggraph(layout) +
  geom_edge_link(alpha = 0.1, colour = "grey60") +
  geom_node_point(aes(size = degree, colour = community), alpha
   ↪  = 0.8) +
  scale_size_continuous(range = c(1, 5), guide = "none") +
  scale_colour_manual(values = palette_sci(
    n_distinct(V(giant)$community)
  )) +
  labs(colour = "Community") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```

```
comp_bc <- components(g_bibcoup)
giant_bc <- induced_subgraph(g_bibcoup,
                             which(comp_bc$membership ==
                              ↪  which.max(comp_bc$csize)))
V(giant_bc)$community <- as.factor(membership(cluster_leiden(
  giant_bc, resolution_parameter = 1.0, objective_function =
 ↪  "modularity"
)))
V(giant_bc)$degree <- degree(giant_bc)

set.seed(42)
layout_bc <- create_layout(as_tbl_graph(giant_bc), layout =
 ↪  "fr")

ggraph(layout_bc) +
  geom_edge_link(alpha = 0.1, colour = "grey60") +
  geom_node_point(aes(size = degree, colour = community), alpha
   ↪  = 0.8) +
```

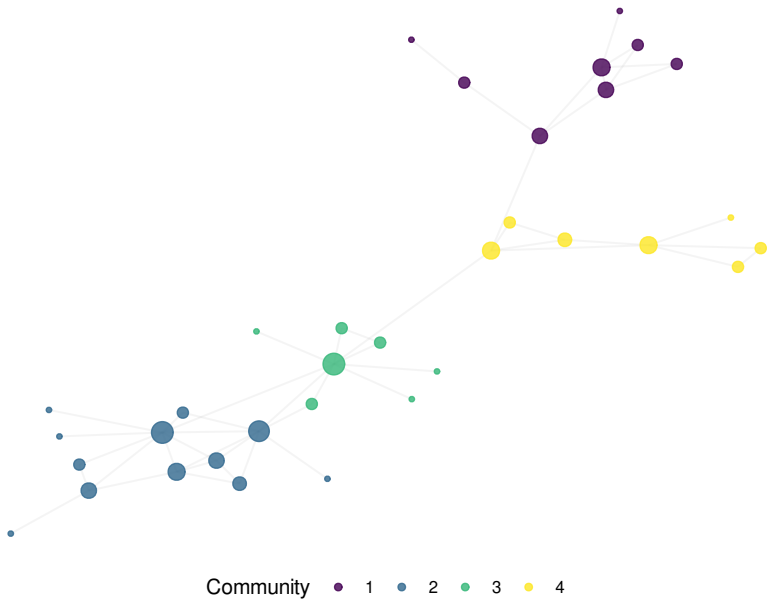


Figure 16.1: Co-citation network of frequently co-cited references in Scientometrics (2020–2023).

```
  scale_size_continuous(range = c(1, 5), guide = "none") +
  scale_colour_manual(values = palette_sci(
    n_distinct(V(giant_bc)$community)
  )) +
  labs(colour = "Community") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```

16.4.5 Comparing the two approaches

```
comparison <- tibble(
  Method = c("Co-citation", "Bibliographic coupling"),
  Nodes = c(vcount(g_cocit), vcount(g_bibcoup)),
  Edges = c(ecount(g_cocit), ecount(g_bibcoup)),
  Components = c(components(g_cocit)$no,
    ↪  components(g_bibcoup)$no),
  Density = c(round(graph.density(g_cocit), 4),
              round(graph.density(g_bibcoup), 4))
)
```

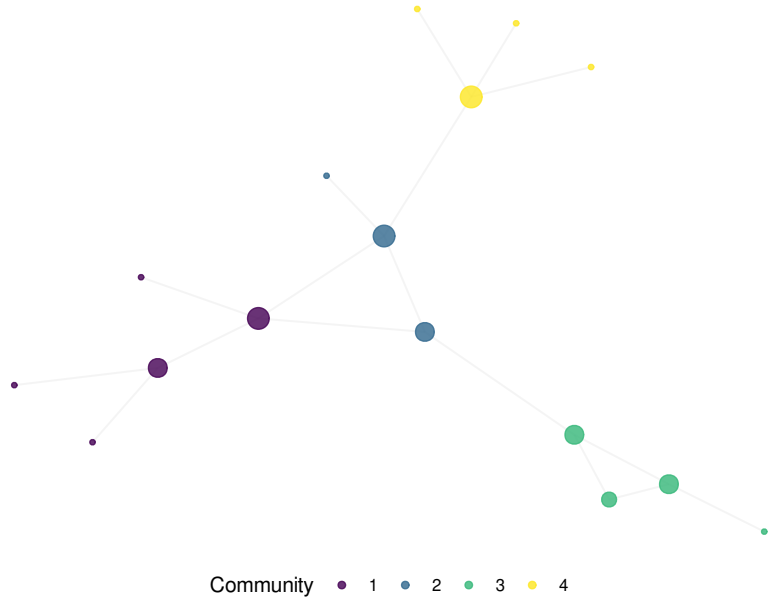


Figure 16.2: Bibliographic coupling network of Scientometrics articles (2020–2023).

```
comparison |> gt()
```

Method	Nodes	Edges	Components	Density
Co-citation	116	123	27	0.0184
Bibliographic coupling	80	64	23	0.0203

16.5 Diagnostics and interpretation

- **Threshold selection**: The minimum co-citation count or shared-reference count determines network density. Too low a threshold creates a hairball; too high excludes important links. Experiment with multiple thresholds and report the chosen value.
- **Normalisation**: Raw co-citation counts favour highly cited papers. Cosine normalisation produces more balanced networks. Always report whether counts are raw or normalised.
- **Giant component ratio**: A fragmented co-citation network may indicate that the corpus spans multiple disconnected subfields, or that the threshold is too high.

- **Temporal stability**: Co-citation networks change as new papers cite old ones. Bibliographic coupling networks are fixed at publication time. If you need a stable snapshot of recent literature, coupling is preferable.

16.6 Limitations and responsible use

16.7 Limitations and responsible use

- **Co-citation reflects citing behaviour, not truth.** Frequently co-cited papers may be co-cited because they represent a dominant paradigm, not because they are the best or most relevant works. Minority viewpoints may be invisible.
- **Reference list quality.** Both methods depend entirely on the accuracy and completeness of reference lists. Missing references, formatting errors, and DOI mismatches reduce coverage. OpenAlex reference parsing is imperfect (Priem et al., 2022).
- **Size effects.** Papers with long reference lists contribute disproportionately to bibliographic coupling. Consider normalising by reference list length.
- **Not evaluative.** Co-citation and coupling measure relatedness, not quality. A highly co-cited paper is intellectually central, which is not the same as excellent or correct (Hicks et al., 2015).

16.8 Common pitfalls

16.9 Common pitfalls

- **Confusing the two methods.** Co-citation links cited documents; bibliographic coupling links citing documents. They answer different questions about the same citation data.
- **Skipping normalisation.** Raw counts produce networks dominated by a few highly cited papers. Always normalise for balanced analysis.
- **Ignoring the threshold.** Reporting results without specifying the minimum count or similarity cutoff makes the analysis irreproducible.
- **Treating coupling as dynamic.** Bibliographic coupling strength is fixed at publication. Do not interpret changes in coupling networks over time as evolving relationships — they reflect changes in the corpus, not the links.

16.10 Exercises

1. **Cosine normalisation.** Compute cosine-normalised co-citation strengths instead of raw counts. How does the network structure change? Do the same core documents emerge?
2. **Temporal co-citation.** Split the corpus into two time periods (2020–2021 and 2022–2023). Build a co-citation network for each. Which co-citation pairs are stable across periods, and which are new?
3. **Bibliographic coupling by year.** For each publication year, build a bibliographic coupling network and report the number of components and modularity. Does the intellectual structure become more or less fragmented over time?
4. **Identifying research fronts.** Use bibliographic coupling communities to identify research fronts. For each community, list the three most frequent keywords or title words. Do the fronts correspond to recognisable subfields?

16.11 Solutions

Solutions are provided in E.

16.12 Further reading

- Small (1973) — Co-citation analysis: the foundational paper.
- Kessler (1963) — Bibliographic coupling: the original proposal.
- Waltman et al. (2010) — Unified approach to bibliometric network construction and clustering.
- Fortunato (2010) — Community detection methods applicable to citation networks.
- Garfield (1955) — The intellectual origins of citation indexing.

16.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] ggraph_2.2.2       tidygraph_1.3.1    igraph_2.3.4
 ↪  quanteda_4.5.1
#>  [5] pdftools_3.9.1     arrow_26.0.1.1     bibliometrix_5.5.0
 ↪  RefManageR_1.4.0
#>  [9] bib2df_1.1.2.0     rcrossref_1.2.1    gt_1.4.0
 ↪  tidytext_0.4.3
#> [13] glue_1.8.1         openalexR_3.1.2    lubridate_1.9.5
 ↪  forcats_1.0.1
#> [17] stringr_1.6.0      dplyr_1.2.1        purrr_1.2.2
 ↪  readr_2.2.0
#> [21] tidyr_1.3.2        tibble_3.3.1       ggplot2_4.0.4
 ↪  tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32         fs_2.1.0               vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9        contentanalysis_1.1.1  curl_8.1.0
```

```
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9           httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  shinycssloaders_1.1.0
#>  [34] rprojroot_2.1.1        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [37] urltools_1.7.3.1      timechange_0.4.0      httr_1.4.9
#>  [40] polyclip_1.10-7       compiler_4.6.1        here_1.0.2
#>  [43] bit64_4.8.6            withr_3.0.3           S7_1.0.0
#>  [46] viridis_0.6.5         ggforce_0.5.0        MASS_7.3-65
#>  [49] bibliometrixData_0.3.0 tools_4.6.1          otel_0.3.0
#>  [52] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [55] httpuv_1.6.17         promises_1.5.0        grid_4.6.1
#>  [58] stringdist_0.9.17      generics_0.1.4
 ↪  gtable_0.3.6
#>  [61] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [64] data.table_1.18.6.1    hms_1.1.4             xml2_1.6.0
#>  [67] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [70] later_1.4.8            tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [73] bit_4.6.0              tidyselect_1.2.1
 ↪  miniUI_0.1.2
#>  [76] knitr_1.52             gridExtra_2.3.1
 ↪  bookdown_0.48
#>  [79] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [82] DT_0.34.0              humaniformat_0.6.0
 ↪  visNetwork_2.1.4
#>  [85] stringi_1.8.9         qpdf_1.4.1            yaml_2.3.12
#>  [88] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#>  [91] xtable_1.8-8           systemfonts_1.3.4     Rcpp_1.1.2
#>  [94] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#>  [97] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [100] viridisLite_0.4.3      scales_1.4.0
 ↪  openxlsx_4.2.9
#> [103] rlang_1.3.0            fastmatch_1.1-8
```

Chapter 17

Co-word and Keyword Co-occurrence

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

17.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the principles of co-word analysis and its role in science mapping
- Extract and clean keywords from OpenAlex data (author keywords and concepts)
- Build a keyword co-occurrence matrix and convert it to a network
- Apply community detection to identify topical clusters
- Visualise keyword co-occurrence networks with interpretable layouts

17.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
library(ggraph)
library(glue)
```

```
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

17.3 Conceptual background

Co-word analysis maps the conceptual structure of a research field by examining which terms appear together in the same documents. Introduced by Callon et al. (1983), the method assumes that when two keywords repeatedly co-occur across publications, they are thematically related. The resulting co-occurrence network reveals the topical landscape of a field: clusters of densely connected keywords represent coherent research themes, while bridges between clusters indicate interdisciplinary connections.

The method works with several types of terms:

- **Author keywords**: Terms selected by the paper's authors. These are intentional descriptors but suffer from inconsistency — authors may use different terms for the same concept ("machine learning" vs. "ML" vs. "statistical learning").
- **Indexed keywords**: Terms assigned by database indexers (e.g., MeSH terms in PubMed). More consistent but only available in some databases.
- **OpenAlex concepts**: Algorithmically assigned topics at multiple levels of a hierarchical taxonomy. These provide broad coverage but may miss nuanced distinctions (Priem et al., 2022).
- **Title/abstract words**: Extracted via text mining. Comprehensive but noisy; requires extensive preprocessing.

Normalisation is important for co-word networks. Raw co-occurrence counts favour high-frequency terms. Common normalisations include the **association strength** (equivalent to pointwise mutual information) and the **equivalence index** (cosine of the co-occurrence vector). Waltman et al. (2010) demonstrated that association strength produces the most balanced network structures for bibliometric mapping.

Co-word analysis complements citation-based methods (16.3). Citation networks reveal intellectual influence; co-word networks reveal thematic content. Combining both provides a more complete picture of a field's structure.

17.4 Worked example

17.4.1 Extracting keywords

We use OpenAlex-assigned topic labels as terms in a sample of scientometrics research. These are algorithmic labels, not author-supplied keywords. The flattened topic table also includes parent subfields, fields, and domains; retain only topic identifiers (`T...`) and count each term once per work.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2019-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)
```

```
keywords <- works |>
  select(id, topics) |>
  unnest(topics, names_sep = "_") |>
  filter(str_detect(topics_id, "(^|/)T[0-9]+$")) |>
  filter(topics_display_name != "") |>
  select(work_id = id, keyword = topics_display_name) |>
  mutate(keyword = str_to_lower(str_trim(keyword))) |>
  filter(!is.na(keyword), nchar(keyword) >= 3) |>
  distinct(work_id, keyword)

kw_counts <- keywords |>
  count(keyword, sort = TRUE)

cat(glue("Total keyword occurrences: {nrow(keywords)}\n"))
```

```
#> Total keyword occurrences: 1500
```

```
cat(glue("Unique keywords: {n_distinct(keywords$keyword)}\n"))
```

```
#> Unique keywords: 137
```

```
kw_counts |>
  head(20) |>
  mutate(keyword = fct_reorder(keyword, n)) |>
  ggplot(aes(x = n, y = keyword)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Frequency", y = NULL) +
  theme_sci()
```

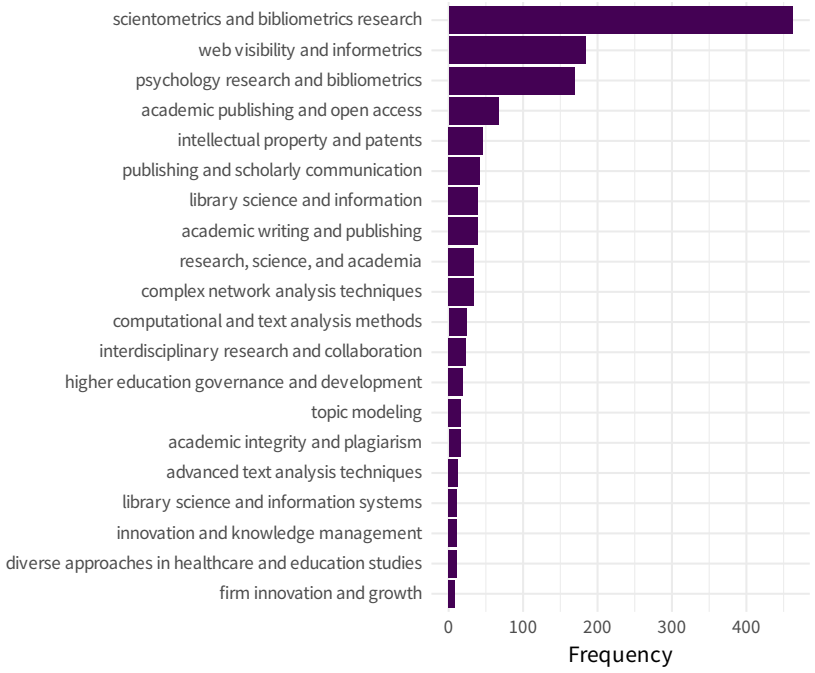


Figure 17.1: Top 20 most frequent assigned topics used as terms in the Scientometrics sample.

17.4.2 Building the co-occurrence network

We create edges between keywords that appear in the same paper.

```
kw_frequent <- kw_counts |>
  filter(n >= 5) |>
  pull(keyword)

kw_filtered <- keywords |>
  filter(keyword %in% kw_frequent)

coword_pairs <- kw_filtered |>
  inner_join(kw_filtered, by = "work_id", suffix = c("_a", "_b"),
             relationship = "many-to-many") |>
  filter(keyword_a < keyword_b) |>
  count(keyword_a, keyword_b, name = "cooccurrence")

coword_top <- coword_pairs |>
  filter(cooccurrence >= 3)

g_kw <- graph_from_data_frame(
  coword_top |> select(keyword_a, keyword_b, weight =
 ↪  cooccurrence),
  directed = FALSE
) |>
  simplify(edge.attr.comb = list(weight = "sum"))

cat(glue("Keyword network: {vcount(g_kw)} nodes, {ecount(g_kw)}
 ↪  edges\n"))
```

```
#> Keyword network: 30 nodes, 69 edges
```

17.4.3 Community detection for topical clusters

```
V(g_kw)$degree <- degree(g_kw)
V(g_kw)$strength <- strength(g_kw)

communities <- cluster_leiden(g_kw, resolution_parameter = 0.8,
                              objective_function = "modularity")
V(g_kw)$community <- as.factor(membership(communities))

cat(glue("Communities:
 ↪  {length(unique(membership(communities)))}\n"))
```

```
#> Communities: 4
```

```
cat(glue("Modularity: {round(modularity(g_kw,
 ↪  membership(communities)), 3)}\n"))
```

```
#> Modularity: 0.049
```

```
community_summary <- tibble(
  keyword = V(g_kw)$name,
  community = V(g_kw)$community,
  strength = V(g_kw)$strength
) |>
  group_by(community) |>
  slice_max(strength, n = 5) |>
  summarise(top_keywords = paste(keyword, collapse = ", "),
            n_keywords = n(), .groups = "drop")

community_summary |> gt()
```

community	top_keywords	n_keywords
1	scientometrics and bibliometrics research, web visibility and informetrics, psychology research and bibliometrics, academic publishing and open access, publishing and scholarly communication	5
2	computational and text analysis methods, topic modeling, advanced text analysis techniques	3
3	library science and information systems, data quality and management	2
4	intellectual property and patents, innovation and knowledge management, firm innovation and growth	3

17.4.4 Visualisation

```
set.seed(42)
layout <- create_layout(as_tbl_graph(g_kw), layout = "fr")

ggraph(layout) +
  geom_edge_link(aes(width = weight), alpha = 0.15, colour =
   ↪  "grey60") +
  scale_edge_width_continuous(range = c(0.3, 2), guide = "none")
   ↪  +
  geom_node_point(aes(size = degree, colour = community), alpha
   ↪  = 0.8) +
  geom_node_text(aes(label = ifelse(degree > quantile(degree,
   ↪  0.85),
```

```
                                        name, NA_character_)),
                repel = TRUE, size = 2.5, max.overlaps = 20,
                 ↪  na.rm = TRUE) +
scale_size_continuous(range = c(1, 6), guide = "none") +
scale_colour_manual(values = palette_sci(
  n_distinct(V(g_kw)$community)
)) +
labs(colour = "Cluster") +
theme_void(base_family = "sans", base_size = 11) +
theme(legend.position = "bottom")
```

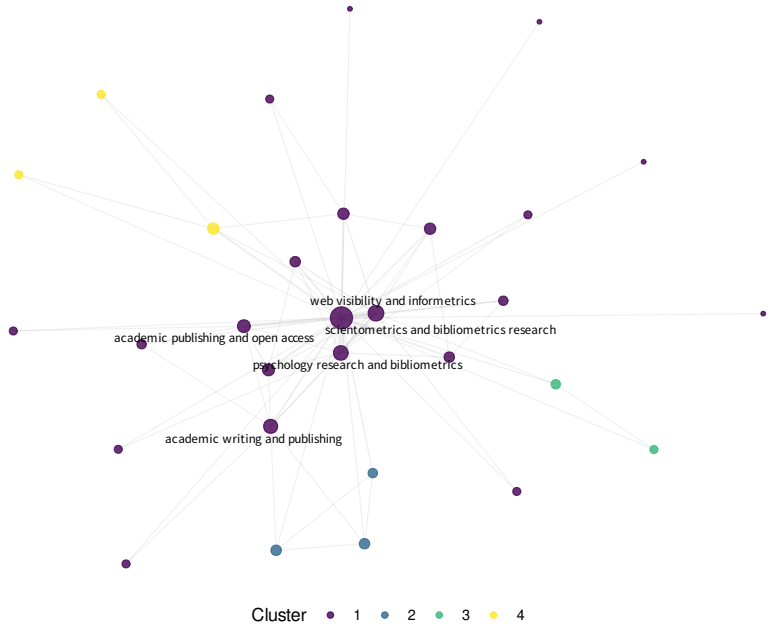


Figure 17.2: Keyword co-occurrence network coloured by topical cluster.

17.5 Diagnostics and interpretation

- **Keyword cleaning**: Inconsistent terminology inflates the vocabulary and creates spurious nodes. Standardise spelling, merge synonyms (e.g., “h-index” and “hirsch index”), and remove generic terms (“research”, “analysis”) that co-occur with everything but convey no topical information.

- **Frequency threshold**: Including rare keywords produces large, sparse, unreadable networks. Start with keywords appearing in at least 5 papers and adjust based on corpus size.
- **Community interpretability**: Each community should correspond to a recognisable research theme. If communities are uninterpretable, the resolution parameter may need adjustment or keywords need further cleaning.
- **Centrality interpretation**: High-degree keywords are thematically central (used across many contexts). High-betweenness keywords bridge distinct topics and may represent interdisciplinary concepts.

17.6 Limitations and responsible use

17.7 Limitations and responsible use

- **Vocabulary inconsistency.** Author keywords are not controlled vocabulary. The same concept may appear under multiple terms, fragmenting what should be a single node. Merging synonyms requires domain expertise.
- **Algorithmic concepts.** OpenAlex concepts are assigned by machine learning and may contain errors — especially at fine-grained levels. Always validate topic assignments by reading sample papers (Priem et al., 2022).
- **Static snapshots.** A co-word network represents a time-averaged view. Emerging topics with few publications may be invisible. Consider building temporal slices to track field evolution.
- **Not evaluative.** Popular keywords indicate research activity, not research quality or societal impact. Do not equate topical centrality with importance (Hicks et al., 2015).

17.8 Common pitfalls

17.9 Common pitfalls

- **Not cleaning keywords.** Punctuation variants ("co-authorship" vs. "coauthorship"), case differences, and trailing whitespace create duplicate nodes. Clean before building the network.
- **Including stop-keywords.** Generic terms like "research", "study", "analysis" co-occur with everything and dominate the network without conveying topical information. Remove them.

- **Comparing co-occurrence counts across corpora of different sizes.** Larger corpora produce higher co-occurrence counts mechanically. Normalise or use relative measures.
- **Over-reading small clusters.** A cluster with two or three keywords may reflect a single paper, not a research theme. Report cluster size alongside content.

17.10 Exercises

1. **Author keywords vs. concepts.** Build co-occurrence networks from both author keywords (if available) and OpenAlex concepts for the same corpus. Compare the resulting topical clusters.

2. **Temporal evolution.** Split the corpus into yearly slices. For each year, build a co-word network and identify the top 5 keywords by degree. Which keywords are consistently central, and which emerge over time?

3. **Strategic diagram.** Compute density (internal cohesion) and centrality (external connections) for each keyword cluster. Plot them on a strategic diagram (density vs. centrality quadrants). Which clusters are core themes and which are peripheral?

4. **Normalisation comparison.** Build two networks from the same data: one with raw co-occurrence counts and one with association strength normalisation. How do the most central keywords differ?

17.11 Solutions

Solutions are provided in E.

17.12 Further reading

- Callon et al. (1983) — The foundational paper on co-word analysis for mapping science.
- Waltman et al. (2010) — Unified framework for bibliometric network construction, including keyword networks.
- Aria and Cuccurullo (2017) — `bibliometrix` includes co-word analysis via the `biblioNetwork()` function.
- Priem et al. (2022) — OpenAlex concepts and topic classification.

17.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] ggraph_2.2.2       tidygraph_1.3.1    igraph_2.3.4
 ↪  quanteda_4.5.1
#>  [5] pdftools_3.9.1     arrow_26.0.1.1     bibliometrix_5.5.0
 ↪  RefManageR_1.4.0
#>  [9] bib2df_1.1.2.0     rcrossref_1.2.1    gt_1.4.0
 ↪  tidytext_0.4.3
#> [13] glue_1.8.1         openalexR_3.1.2    lubridate_1.9.5
 ↪  forcats_1.0.1
#> [17] stringr_1.6.0      dplyr_1.2.1        purrr_1.2.2
 ↪  readr_2.2.0
#> [21] tidyr_1.3.2        tibble_3.3.1       ggplot2_4.0.4
 ↪  tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
```

```
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32        fs_2.1.0              vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9       contentanalysis_1.1.1  curl_8.1.0
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9            httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  shinycssloaders_1.1.0
#>  [34] rprojroot_2.1.1        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [37] urltools_1.7.3.1       timechange_0.4.0       httr_1.4.9
#>  [40] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [43] bit64_4.8.6            withr_3.0.3             S7_1.0.0
#>  [46] viridis_0.6.5         ggforce_0.5.0          MASS_7.3-65
#>  [49] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [52] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [55] httpuv_1.6.17          promises_1.5.0         grid_4.6.1
#>  [58] stringdist_0.9.17      generics_0.1.4
 ↪  gtable_0.3.6
#>  [61] tzdb_0.5.0             rscopus_0.9.0           ca_0.72
#>  [64] data.table_1.18.6.1    hms_1.1.4              xml2_1.6.0
#>  [67] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [70] later_1.4.8            tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [73] bit_4.6.0              tidyselect_1.2.1
 ↪  miniUI_0.1.2
#>  [76] knitr_1.52             gridExtra_2.3.1
 ↪  bookdown_0.48
#>  [79] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [82] DT_0.34.0              humaniformat_0.6.0
 ↪  visNetwork_2.1.4
#>  [85] stringi_1.8.9          qpdf_1.4.1            yaml_2.3.12
#>  [88] evaluate_1.0.5         httpcode_0.3.0          cli_3.6.6
#>  [91] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#>  [94] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#>  [97] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
```

```
#> [100] viridisLite_0.4.3      scales_1.4.0
 ↪  openxlsx_4.2.9
#> [103] rlang_1.3.0            fastmatch_1.1-8
```

Chapter 18

Community Detection and Backbone Extraction

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

18.1 Learning objectives

After completing this chapter, you will be able to:

- Explain why dense networks require backbone extraction before community detection
- Apply the disparity filter to extract statistically significant edges
- Tune the resolution parameter for Louvain and Leiden community detection
- Compare community assignments across methods using Normalised Mutual Information
- Interpret hierarchical community structure at multiple resolutions

18.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
library(ggraph)
```

```
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

18.3 Conceptual background

Bibliometric networks are often dense: in a co-citation or bibliographic coupling network, every pair of documents with any shared citation creates an edge. A corpus of 1,000 papers can easily produce a network with hundreds of thousands of edges, most of which represent weak or coincidental relationships. Applying community detection directly to such networks yields poor results — the algorithms cannot distinguish signal from noise.

Backbone extraction addresses this by pruning edges that are not statistically significant, retaining only the "skeleton" of the network. The **disparity filter** (Serrano et al., 2009) tests each edge against a null model based on the local weight distribution of its endpoints. An edge is retained if its weight is unexpectedly high given the node's total weight and degree. This preserves multi-scale structure: both high-weight and low-weight nodes can retain their most important connections.

An alternative is simple **threshold filtering** (remove edges below a fixed weight), but this systematically removes connections between low-activity nodes, biasing the backbone toward already-prominent actors. The disparity filter avoids this bias by using a local significance test.

Once the backbone is extracted, **community detection** identifies groups of nodes that are more densely connected internally than externally. The Leiden algorithm (Traag et al., 2019) improves on Louvain (Blondel et al., 2008) by guaranteeing that all detected communities are internally connected. Both algorithms accept a **resolution parameter** that controls community granularity: lower values produce fewer, larger communities; higher values produce more, smaller communities. There is no single "correct" resolution — the appropriate value depends on the research question.

Fortunato (2010) provides a comprehensive review of community detection methods. In practice, running the algorithm at multiple resolutions and examining the hierarchical structure provides the richest understanding of network organisation.

18.4 Worked example

18.4.1 Building a dense network

We construct a bibliographic coupling network from scientometrics articles.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2020-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 300, seed = 42)
)

refs <- works |>
  select(citing_id = id, referenced_works) |>
  unnest(referenced_works) |>
  rename(cited_id = referenced_works)

bibcoup <- refs |>
  inner_join(refs, by = "cited_id", suffix = c("_a", "_b"),
             relationship = "many-to-many") |>
  filter(citing_id_a < citing_id_b) |>
  count(citing_id_a, citing_id_b, name = "shared_refs")

g_full <- graph_from_data_frame(
  bibcoup |> select(citing_id_a, citing_id_b, weight =
↪  shared_refs),
  directed = FALSE
) |>
  simplify(edge.attr.comb = list(weight = "sum"))

cat(glue("Full network: {vcount(g_full)} nodes, {ecount(g_full)}
↪  edges\n"))
```

```
#> Full network: 288 nodes, 3430 edges
```

```
cat(glue("Density: {round(graph.density(g_full), 4)}\n"))
```

```
#> Density: 0.083
```

18.4.2 Backbone extraction: disparity filter

```
disparity_filter <- function(g, alpha = 0.05) {
  el <- as_data_frame(g, what = "edges")
  node_strength <- strength(g)

  keep <- map2_lgl(seq_len(nrow(el)), el$weight, function(i, w) {
    from_name <- el$from[i]
    to_name <- el$to[i]
    k_from <- degree(g, from_name)
    k_to <- degree(g, to_name)
    s_from <- node_strength[from_name]
    s_to <- node_strength[to_name]

    p_from <- (1 - w / s_from)^(k_from - 1)
    p_to <- (1 - w / s_to)^(k_to - 1)

    min(p_from, p_to) < alpha
  })

  subgraph.edges(g, which(keep), delete.vertices = TRUE)
}

g_backbone <- disparity_filter(g_full, alpha = 0.05)
cat(glue("Backbone: {vcount(g_backbone)} nodes,
 ↪  {ecount(g_backbone)} edges\n"))
```

```
#> Backbone: 135 nodes, 125 edges
```

```
cat(glue("Edge retention: {scales::percent(ecount(g_backbone) /
 ↪  ecount(g_full))}\n"))
```

```
#> Edge retention: 4%
```

18.4.3 Threshold filtering for comparison

```
threshold <- quantile(E(g_full)$weight, 0.75)
g_threshold <- subgraph.edges(
  g_full,
  which(E(g_full)$weight >= threshold),
  delete.vertices = TRUE
)

cat(glue("Threshold (75th pctile = {threshold}):
 ↪  {vcount(g_threshold)} nodes, {ecount(g_threshold)}
 ↪  edges\n"))
```

```
#> Threshold (75th pctile = 2): 253 nodes, 1115 edges
```

18.4.4 Community detection at multiple resolutions

```
resolutions <- c(0.5, 0.8, 1.0, 1.5, 2.0)

sweep_results <- map_dfr(resolutions, function(res) {
  comm <- cluster_leiden(g_backbone, resolution_parameter = res,
                         objective_function = "modularity")
  tibble(
    resolution = res,
    n_communities = length(unique(membership(comm))),
    modularity = round(modularity(g_backbone, membership(comm)),
    ↪  3),
    max_size = max(table(membership(comm))),
    min_size = min(table(membership(comm)))
  )
})

sweep_results |> gt()
```

resolution	n_communities	modularity	max_size	min_size
0.5	36	0.91	12	2
0.8	36	0.91	12	2
1.0	36	0.91	12	2
1.5	36	0.91	12	2
2.0	36	0.91	12	2

```
ggplot(sweep_results, aes(x = resolution)) +
  geom_line(aes(y = n_communities), colour = palette_sci(2)[1],
  ↪  linewidth = 1) +
  geom_point(aes(y = n_communities), colour = palette_sci(2)[1],
  ↪  size = 3) +
  geom_line(aes(y = modularity * max(n_communities)),
            colour = palette_sci(2)[2], linewidth = 1, linetype =
            ↪  "dashed") +
  scale_y_continuous(
    name = "Number of communities",
    sec.axis = sec_axis(~ . / max(sweep_results$n_communities),
                        name = "Modularity")
  ) +
  labs(x = "Resolution parameter") +
  theme_sci()
```

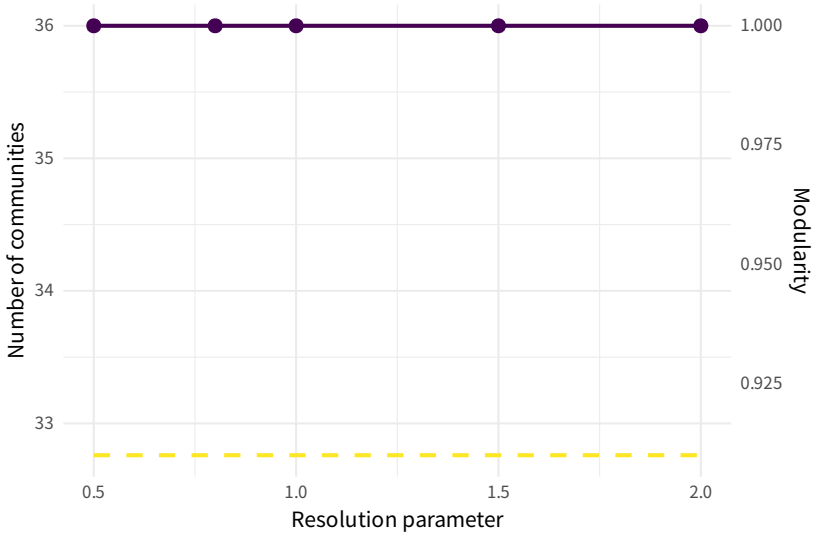


Figure 18.1: Number of communities and modularity as a function of resolution parameter.

18.4.5 Comparing backbone vs. full network communities

```
comm_full <- cluster_leiden(g_full, resolution_parameter = 1.0,
                            objective_function = "modularity")
comm_backbone <- cluster_leiden(g_backbone, resolution_parameter
 ↪  = 1.0,
                                objective_function = "modularity")

shared_nodes <- intersect(V(g_full)$name, V(g_backbone)$name)
mem_full <- membership(comm_full)[shared_nodes]
mem_back <- membership(comm_backbone)[shared_nodes]

nmi <- compare(mem_full, mem_back, method = "nmi")
cat(glue("NMI (full vs backbone, shared nodes): {round(nmi,
 ↪  3)}\n"))
```

```
#> NMI (full vs backbone, shared nodes): 0.629
```

```
cat(glue("Communities in full network:
 ↪  {length(unique(membership(comm_full)))}\n"))
```

```
#> Communities in full network: 9
```

```
cat(glue("Communities in backbone:
 ↪  {length(unique(membership(comm_backbone)))}\n"))
```

```
#> Communities in backbone: 36
```

18.4.6 Visualisation

```
V(g_backbone)$community <- as.factor(membership(comm_backbone))
V(g_backbone)$degree <- degree(g_backbone)

set.seed(42)
layout <- create_layout(as_tbl_graph(g_backbone), layout = "fr")

ggraph(layout) +
  geom_edge_link(alpha = 0.1, colour = "grey60") +
  geom_node_point(aes(size = degree, colour = community), alpha
   ↪  = 0.8) +
  scale_size_continuous(range = c(1, 5), guide = "none") +
  scale_colour_manual(values = palette_sci(
    n_distinct(V(g_backbone)$community)
  )) +
  labs(colour = "Community") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```

18.5 Diagnostics and interpretation

- **Edge retention rate**: The disparity filter typically retains 10–30% of edges. Retention above 50% suggests the network may not be dense enough to require backbone extraction.
- **Isolated nodes**: Backbone extraction removes nodes that lose all their edges. Report how many nodes are lost and whether they represent a biased subset (e.g., low-cited papers).
- **Resolution plateau**: If modularity remains stable across a range of resolutions, the community structure is robust. Rapid changes suggest sensitivity to the parameter.
- **Singleton communities**: Communities with one or two nodes are usually noise. Consider merging them into the nearest larger community or excluding them from interpretation.

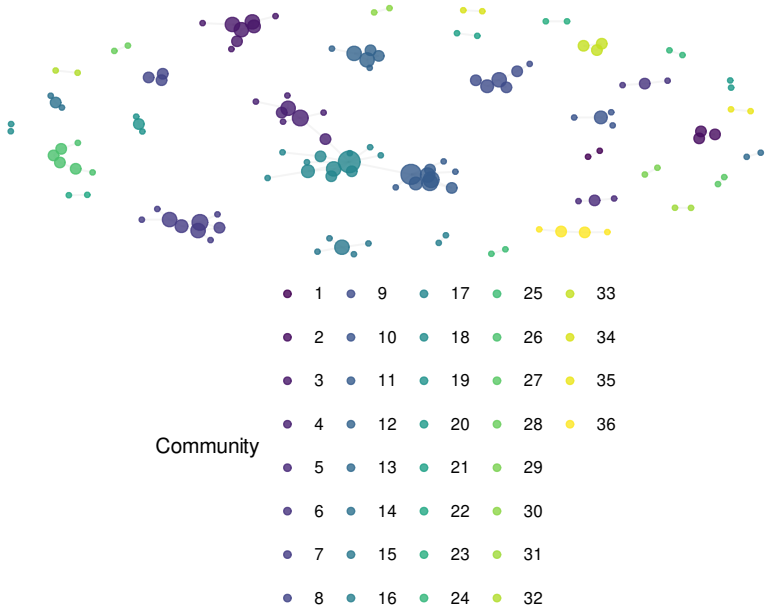


Figure 18.2: Backbone network with Leiden communities (resolution = 1.0).

18.6 Limitations and responsible use

18.7 Limitations and responsible use

- **The disparity filter has assumptions.** It assumes a uniform null distribution of edge weights across a node's connections. Highly skewed weight distributions may violate this assumption.
- **Backbone choice affects conclusions.** Different backbone methods (disparity filter, noise-corrected, threshold) retain different edges and can produce different community structures. Report the method and alpha level.
- **Resolution is a researcher decision.** There is no objectively "correct" resolution parameter. The choice determines the granularity of analysis and should be justified by the research question, not by optimising modularity alone (Fortunato, 2010).
- **Communities are not ground truth.** Detected communities are algorithmic constructs. They reflect structural patterns in citation data, not necessarily real-world research groups or coherent intellectual traditions (Hicks et al., 2015).

18.8 Common pitfalls

18.9 Common pitfalls

- **Applying community detection to unfiltered dense networks.** The result is usually a single giant community plus isolated fragments. Always extract the backbone first.
- **Using a single resolution without justification.** The default resolution of 1.0 is arbitrary. Run a sweep and show the sensitivity analysis.
- **Comparing modularity across networks of different sizes.** Modularity values are not comparable between different networks. Use NMI to compare partitions.
- **Interpreting backbone edges as "strong" relationships.** The disparity filter retains locally significant edges, not necessarily those with the highest absolute weight. A low-weight edge can be retained if it is important relative to its endpoint's other connections.

18.10 Exercises

1. **Alpha sensitivity.** Run the disparity filter with alpha values of 0.01, 0.05, 0.10, and 0.20. Plot the number of retained edges and communities against alpha. At what alpha does the network become disconnected?

2. **Threshold vs. disparity.** Compare threshold-filtered and disparity-filtered communities for the same network. Use NMI to quantify agreement. Which method produces more interpretable communities?

3. **Hierarchical structure.** Run Leiden at resolutions 0.5, 1.0, and 2.0. For each pair of resolutions, check whether the finer communities are nested within the coarser ones (a Sankey or alluvial diagram can help visualise this).

4. **Weighted vs. unweighted community detection.** Remove edge weights from the backbone and run Leiden again. How much does the community structure change?

18.11 Solutions

Solutions are provided in E.

18.12 Further reading

- Serrano et al. (2009) — The disparity filter for extracting multiscale network backbones.
- Traag et al. (2019) — The Leiden algorithm with resolution parameter tuning.
- Blondel et al. (2008) — The Louvain algorithm for fast community detection.
- Fortunato (2010) — Comprehensive review of community detection, including resolution limits.
- Waltman et al. (2010) — Network construction and clustering for bibliometric applications.

18.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
```

```
#>  [1] ggraph_2.2.2       tidygraph_1.3.1    igraph_2.3.4
 ↪  quanteda_4.5.1
#>  [5] pdftools_3.9.1     arrow_26.0.1.1     bibliometrix_5.5.0
 ↪  RefManageR_1.4.0
#>  [9] bib2df_1.1.2.0     rcrossref_1.2.1    gt_1.4.0
 ↪  tidytext_0.4.3
#> [13] glue_1.8.1         openalexR_3.1.2    lubridate_1.9.5
 ↪  forcats_1.0.1
#> [17] stringr_1.6.0      dplyr_1.2.1        purrr_1.2.2
 ↪  readr_2.2.0
#> [21] tidyr_1.3.2        tibble_3.3.1       ggplot2_4.0.4
 ↪  tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32        fs_2.1.0              vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9       contentanalysis_1.1.1  curl_8.1.0
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9            httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  shinycssloaders_1.1.0
#>  [34] rprojroot_2.1.1        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [37] urltools_1.7.3.1      timechange_0.4.0       httr_1.4.9
#>  [40] polyclip_1.10-7       compiler_4.6.1         here_1.0.2
#>  [43] bit64_4.8.6            withr_3.0.3            S7_1.0.0
#>  [46] viridis_0.6.5         ggforce_0.5.0         MASS_7.3-65
#>  [49] bibliometrixData_0.3.0 tools_4.6.1           otel_0.3.0
#>  [52] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [55] httpuv_1.6.17         promises_1.5.0         grid_4.6.1
#>  [58] stringdist_0.9.17      generics_0.1.4
 ↪  gtable_0.3.6
#>  [61] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [64] data.table_1.18.6.1    hms_1.1.4             xml2_1.6.0
#>  [67] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
```

```
#>  [70] later_1.4.8           tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [73] bit_4.6.0             tidyselect_1.2.1
 ↪  miniUI_0.1.2
#>  [76] knitr_1.52            gridExtra_2.3.1
 ↪  bookdown_0.48
#>  [79] crul_1.6.0            xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [82] DT_0.34.0             humaniformat_0.6.0
 ↪  visNetwork_2.1.4
#>  [85] stringi_1.8.9         qpdf_1.4.1            yaml_2.3.12
#>  [88] evaluate_1.0.5        httpcode_0.3.0          cli_3.6.6
#>  [91] xtable_1.8-8          systemfonts_1.3.4      Rcpp_1.1.2
#>  [94] readxl_1.6.0          triebeard_0.4.1
 ↪  XML_3.99-0.25
#>  [97] parallel_4.6.1        assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [100] viridisLite_0.4.3     scales_1.4.0
 ↪  openxlsx_4.2.9
#> [103] rlang_1.3.0           fastmatch_1.1-8
```

Chapter 19

Science Mapping and Overlay Maps

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

19.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the principles of science mapping and its applications
- Construct a journal-level co-citation network from OpenAlex data
- Create an overlay map that positions a research profile on a base map
- Interpret overlay maps to identify research strengths and gaps
- Discuss the limitations of global science maps for local evaluation

19.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
library(ggraph)
library(glue)
library(gt)
```

```r
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

19.3 Conceptual background

A **science map** is a spatial representation of the structure of science, where proximity indicates relatedness. Just as a geographic map shows the layout of countries and continents, a science map shows the layout of disciplines, fields, and research areas. These maps are constructed from bibliometric data — typically citation links between journals or document sets — and visualised using network layouts or dimensionality reduction techniques.

The most common base maps use **journal-level citation networks**. Two journals are linked if papers in one frequently cite papers in the other. Community detection applied to these networks produces clusters corresponding to broad disciplines (biomedical sciences, physical sciences, social sciences, etc.). Waltman et al. (2010) developed methods for constructing such maps from large-scale citation data, and the resulting visualisations are widely used in science policy and institutional assessment.

An **overlay map** (Rafols et al., 2010) takes a global science map as a fixed background and plots a specific entity's output on top. For example, a university's publications are mapped to the journals in which they appeared; the overlay then shows which areas of the global map the university is active in, and how intensely. This technique is powerful for visualising research profiles, identifying strategic gaps, and comparing institutions or countries without reducing their activity to a single number.

The construction pipeline involves three steps: (1) build or obtain a base map of journals with coordinates and cluster assignments; (2) map the entity's publications to journals in the base map; (3) scale node sizes proportionally and overlay on the base map.

Science maps provide intuitive, holistic overviews of research landscapes. However, they are projections of high-dimensional data into two dimensions, and like any projection, they introduce distortion. Interdisciplinary journals may appear in unexpected locations, and the layout algorithm's random seed affects exact positions (Hicks et al., 2015).

19.4 Worked example

19.4.1 Building a journal co-citation base map

We construct a base map from citation flows between journals in a multidisciplinary sample.

```
works_multi <- fetch_openalex(
  entity = "works",
  from_publication_date = "2022-01-01",
  to_publication_date = "2022-12-31",
  type = "article",
  options = list(sample = 500, seed = 42)
)

journal_data <- works_multi |>
  select(id, source_display_name, referenced_works) |>
  filter(!is.na(source_display_name)) |>
  unnest(referenced_works) |>
  rename(cited_id = referenced_works)

cat(glue("Citing papers with journal info:
 ↪  {n_distinct(journal_data$id)}\n"))
```

```
#> Citing papers with journal info: 499
```

```
cat(glue("Unique source journals:
 ↪  {n_distinct(journal_data$source_display_name)}\n"))
```

```
#> Unique source journals: 467
```

```
journal_cited <- journal_data |>
  select(citing_journal = source_display_name, cited_id)

journal_pairs <- journal_cited |>
  inner_join(journal_cited, by = "cited_id", suffix = c("_a",
   ↪  "_b"),
             relationship = "many-to-many") |>
  filter(citing_journal_a < citing_journal_b) |>
  count(citing_journal_a, citing_journal_b, name = "cocitation")

journal_pairs_top <- journal_pairs |>
  filter(cocitation >= 3)

g_journals <- graph_from_data_frame(
  journal_pairs_top |> select(citing_journal_a,
 ↪  citing_journal_b,
```

```
                                  weight = cocitation),
  directed = FALSE
) |>
  simplify(edge.attr.comb = list(weight = "sum"))

cat(glue("Journal base map: {vcount(g_journals)} journals,
 ↪  {ecount(g_journals)} edges\n"))
```

```
#> Journal base map: 117 journals, 252 edges
```

19.4.2 Clustering journals into disciplines

```
comm <- cluster_leiden(g_journals, resolution_parameter = 0.8,
                       objective_function = "modularity")
V(g_journals)$cluster <- as.factor(membership(comm))
V(g_journals)$degree <- degree(g_journals)

cluster_table <- tibble(
  journal = V(g_journals)$name,
  cluster = V(g_journals)$cluster,
  degree = V(g_journals)$degree
) |>
  group_by(cluster) |>
  summarise(n_journals = n(),
            top_journals = paste(head(journal[order(-degree)],
             ↪  3),
                                 collapse = "; "),
            .groups = "drop") |>
  arrange(desc(n_journals))

cluster_table |> head(8) |> gt()
```

cluster	n_journals	top_journals
1	110	DOAJ (DOAJ: Directory of Open Access Journals); Zenodo (CERN European Organization for Nuclear Research); Advance in Law
2	7	Digital Collections of Colorado (Colorado State University); European Heart Journal; European Journal of Public Health

19.4.3 Visualising the base map

```
set.seed(42)
layout <- create_layout(as_tbl_graph(g_journals), layout = "fr")

ggraph(layout) +
  geom_edge_link(alpha = 0.05, colour = "grey70") +
  geom_node_point(aes(colour = cluster, size = degree), alpha =
  ↪  0.7) +
  scale_size_continuous(range = c(1, 5), guide = "none") +
  scale_colour_manual(values = palette_sci(
    n_distinct(V(g_journals)$cluster)
  )) +
  labs(colour = "Cluster") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```

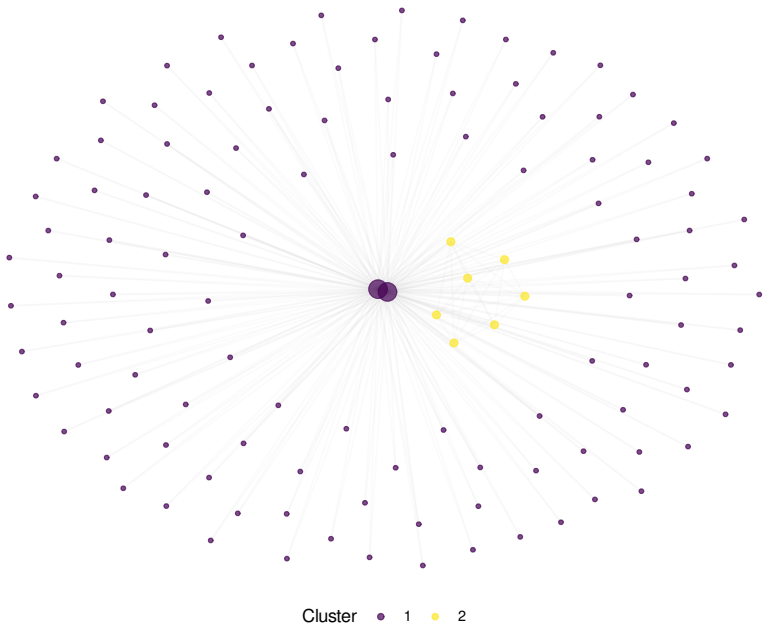


Figure 19.1: Journal co-citation base map coloured by discipline cluster.

19.4.4 Creating an overlay map

We overlay the publication profile of an institution.

```
inst_works <- fetch_openalex(
  entity = "works",
  authorships.institutions.id = "I40120149",
  from_publication_date = "2022-01-01",
  to_publication_date = "2022-12-31",
  type = "article",
  options = list(sample = 300, seed = 42)
)

inst_journals <- inst_works |>
  filter(!is.na(source_display_name)) |>
  count(source_display_name, name = "n_pubs", sort = TRUE)

basemap_journals <- V(g_journals)$name
inst_journals_matched <- inst_journals |>
  filter(source_display_name %in% basemap_journals)

cat(glue("Institution journals matched to base map:
 ↪  {nrow(inst_journals_matched)} / {nrow(inst_journals)}\n"))
```

```
#> Institution journals matched to base map: 4 / 254
```

```
overlay_sizes <- tibble(journal = V(g_journals)$name) |>
  left_join(inst_journals_matched, by = c("journal" =
   ↪  "source_display_name")) |>
  mutate(n_pubs = replace_na(n_pubs, 0))

set.seed(42)
layout_overlay <- create_layout(as_tbl_graph(g_journals), layout
 ↪  = "fr")

ggraph(layout_overlay) +
  geom_edge_link(alpha = 0.03, colour = "grey80") +
  geom_node_point(colour = "grey85", size = 1.5, alpha = 0.5) +
  geom_node_point(
    data = layout_overlay |> left_join(overlay_sizes, by =
     ↪  c("name" = "journal")) |>
      filter(n_pubs > 0),
    aes(size = n_pubs, colour = cluster), alpha = 0.8
  ) +
  scale_size_continuous(range = c(2, 8), name = "Publications") +
  scale_colour_manual(values = palette_sci(
    n_distinct(V(g_journals)$cluster)
  )) +
  labs(colour = "Cluster") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```



Figure 19.2: Overlay map showing Oxford's 2022 publication profile on the journal base map.

19.5 Diagnostics and interpretation

- **Journal matching rate**: The fraction of the entity's publications that match journals in the base map determines overlay completeness. Low matching rates (< 50%) indicate the base map is too narrow for the entity's profile.
- **Layout stability**: Force-directed layouts are sensitive to initial conditions. Always use `set.seed()` and report the seed. Consider consensus layouts from multiple seeds for important maps.
- **Cluster labelling**: Algorithmic clusters need human interpretation. Label clusters by inspecting the journals they contain, not by relying on the algorithm.
- **Overlay concentration**: If the overlay is concentrated in one or two clusters, the institution is highly specialised. Dispersed overlays indicate breadth. Neither is inherently better.

19.6 Limitations and responsible use

19.7 Limitations and responsible use

- **Projection distortion.** Two-dimensional layouts compress high-dimensional citation relationships. Journals at the boundaries between disciplines may be misplaced. Do not over-interpret exact positions.
- **Journal-level aggregation.** Mapping publications to journals loses within-journal variation. A paper in Nature could be biology, physics, or chemistry — the overlay cannot distinguish.
- **Sample-dependent base maps.** Our base map is built from a small sample. Production science maps (e.g., the CWTS journal map) use comprehensive citation data. Treat this example as illustrative.
- **Not evaluative.** Science maps show where research happens, not how good it is. An empty region on a university's overlay map is not a weakness — it may reflect a deliberate strategic choice (Hicks et al., 2015).
- **Interdisciplinary research is underrepresented.** Journals at disciplinary boundaries may cluster with one discipline, making interdisciplinary work partially invisible (Rafols et al., 2010).

19.8 Common pitfalls

19.9 Common pitfalls

- **Using too small a base map.** A base map with fewer than 50 journals lacks resolution. Aim for at least 100–200 journals for meaningful discipline separation.
- **Comparing overlays from different base maps.** Node positions are not transferable between different base maps. To compare institutions, overlay them on the same base map.
- **Ignoring unmatched publications.** Publications in journals not on the base map simply disappear from the overlay. Report the matching rate.
- **Treating clusters as fixed disciplines.** Cluster assignments depend on the algorithm, resolution parameter, and data sample. They are analytical tools, not natural categories.

19.10 Exercises

1. **Multi-institution overlay.** Overlay the profiles of two institutions on the same base map using different colours. Where do their research profiles overlap, and where do they diverge?
2. **Temporal overlay.** Build overlays for the same institution in two different years. Has the research profile shifted toward new areas or remained stable?
3. **Country-level overlay.** Instead of an institution, overlay a country's publication profile. How does a small specialised country (e.g., Singapore) compare to a large diversified one (e.g., the United States)?

19.11 Solutions

Solutions are provided in E.

19.12 Further reading

- Rafols et al. (2010) — Overlay maps for visualising research profiles on global science maps.

- Waltman et al. (2010) — Unified approach to constructing and clustering bibliometric maps.
- Hicks et al. (2015) — The Leiden Manifesto: responsible interpretation of quantitative research profiles.
- Fortunato (2010) — Community detection methods relevant to journal clustering.

19.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] ggraph_2.2.2       tidygraph_1.3.1    igraph_2.3.4
 ↪  quanteda_4.5.1
#>  [5] pdftools_3.9.1     arrow_26.0.1.1     bibliometrix_5.5.0
 ↪  RefManageR_1.4.0
#>  [9] bib2df_1.1.2.0     rcrossref_1.2.1    gt_1.4.0
 ↪  tidytext_0.4.3
#> [13] glue_1.8.1         openalexR_3.1.2    lubridate_1.9.5
 ↪  forcats_1.0.1
```

```
#> [17] stringr_1.6.0      dplyr_1.2.1        purrr_1.2.2
 ↪  readr_2.2.0
#> [21] tidyr_1.3.2        tibble_3.3.1       ggplot2_4.0.4
 ↪  tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32        fs_2.1.0              vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9       contentanalysis_1.1.1  curl_8.1.0
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9            httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  shinycssloaders_1.1.0
#>  [34] rprojroot_2.1.1        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [37] urltools_1.7.3.1      timechange_0.4.0       httr_1.4.9
#>  [40] polyclip_1.10-7       compiler_4.6.1         here_1.0.2
#>  [43] bit64_4.8.6            withr_3.0.3            S7_1.0.0
#>  [46] viridis_0.6.5         ggforce_0.5.0         MASS_7.3-65
#>  [49] bibliometrixData_0.3.0 tools_4.6.1           otel_0.3.0
#>  [52] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [55] httpuv_1.6.17         promises_1.5.0         grid_4.6.1
#>  [58] stringdist_0.9.17      generics_0.1.4
 ↪  gtable_0.3.6
#>  [61] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [64] data.table_1.18.6.1    hms_1.1.4             xml2_1.6.0
#>  [67] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [70] later_1.4.8            tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [73] bit_4.6.0              tidyselect_1.2.1
 ↪  miniUI_0.1.2
#>  [76] knitr_1.52             gridExtra_2.3.1
 ↪  bookdown_0.48
#>  [79] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
```

```
#>  [82] DT_0.34.0              humaniformat_0.6.0
 ↪  visNetwork_2.1.4
#>  [85] stringi_1.8.9         qpdf_1.4.1            yaml_2.3.12
#>  [88] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#>  [91] xtable_1.8-8          systemfonts_1.3.4      Rcpp_1.1.2
#>  [94] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#>  [97] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [100] viridisLite_0.4.3      scales_1.4.0
 ↪  openxlsx_4.2.9
#> [103] rlang_1.3.0            fastmatch_1.1-8
```

Chapter 20

Network Visualization and Interoperability

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

20.1 Learning objectives

After completing this chapter, you will be able to:

- Choose appropriate layout algorithms for different network types
- Export igraph networks to Gephi (GraphML), VOSviewer, and Pajek formats
- Create interactive network visualisations with `visNetwork`
- Apply colour and size encoding that communicates structure clearly
- Produce publication-quality static network figures with `ggraph`

20.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
library(ggraph)
library(visNetwork)
library(glue)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

20.3 Conceptual background

Network visualisation is both an analytical tool and a communication device. A well-designed network figure can reveal structure — clusters, bridges, outliers — at a glance. A poorly designed one obscures these patterns in visual noise.

Layout algorithms determine node positions. Force-directed layouts (Fruchterman-Reingold, Kamada-Kawai) simulate physical forces: connected nodes attract, all nodes repel. They work well for small-to-medium networks (up to ~1,000 nodes) and produce aesthetically pleasing results. For larger networks, the DrL (Distributed Recursive Layout) or OpenOrd algorithms scale better. For trees or hierarchies, Reingold-Tilford or Sugiyama layouts are appropriate. The choice of algorithm depends on the network's structure and the visual question being asked.

Visual encoding maps network properties to visual channels: node size (degree, centrality), node colour (community, attribute), edge width (weight), and edge colour or transparency. Effective encoding follows the principle of proportional ink: the visual weight of an element should correspond to its data importance. Avoid redundant encoding (mapping the same variable to both size and colour) and limit the number of distinct colours to what the reader can distinguish (typically 8–12 categories) (Fortunato, 2010).

Interoperability is essential because no single tool excels at everything. R with igraph and ggraph is excellent for reproducible analysis and publication figures. Gephi (Bastian et al., 2009) provides a rich GUI for exploration and layout refinement. VOSviewer specialises in bibliometric network visualisation with built-in clustering. Pajek handles very large networks efficiently. A practical workflow often involves constructing and analysing the network in R, then exporting to a specialised tool for specific visualisation tasks.

Reproducibility requires that layouts are deterministic. Force-directed algorithms start from random positions, so `set.seed()` before every layout computation. For cross-tool reproducibility, export node coordinates alongside the network data.

20.4 Worked example

20.4.1 Building a sample network

We reuse a co-authorship network from Scientometrics.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2021-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 200, seed = 42)
)

author_data <- works |>
  select(id, authorships) |>
  unnest(authorships, names_sep = "_") |>
  select(work_id = id, author_id = authorships_id,
         author_name = authorships_display_name) |>
  filter(!is.na(author_id))

edges <- author_data |>
  inner_join(author_data, by = "work_id", suffix = c("_1", "_2"),
             relationship = "many-to-many") |>
  filter(author_id_1 < author_id_2) |>
  count(author_id_1, author_id_2, name = "weight")

g <- graph_from_data_frame(
  edges |> select(author_id_1, author_id_2, weight),
  directed = FALSE
) |>
  simplify(edge.attr.comb = list(weight = "sum"))

comp <- components(g)
giant <- induced_subgraph(g, which(comp$membership ==
 ↪  which.max(comp$csize)))

comm <- cluster_leiden(giant, resolution_parameter = 1.0,
                       objective_function = "modularity")
V(giant)$community <- membership(comm)
V(giant)$degree <- degree(giant)

author_lookup <- author_data |> distinct(author_id, author_name)
V(giant)$label <- author_lookup$author_name[
  match(V(giant)$name, author_lookup$author_id)
]

cat(glue("Network: {vcount(giant)} nodes, {ecount(giant)}
 ↪  edges\n"))
```

```
#> Network: 14 nodes, 30 edges
```

20.4.2 Comparing layout algorithms

```
tg <- as_tbl_graph(giant) |>
  mutate(community = as.factor(community))

layouts <- c("fr", "kk", "stress", "drl")
layout_names <- c("Fruchterman-Reingold", "Kamada-Kawai",
↪  "Stress", "DrL")

plots <- map2(layouts, layout_names, function(algo, name) {
  set.seed(42)
  ggraph(tg, layout = algo) +
    geom_edge_link(alpha = 0.1, colour = "grey60") +
    geom_node_point(aes(colour = community, size = degree),
    ↪  alpha = 0.7) +
    scale_size_continuous(range = c(0.5, 4), guide = "none") +
    scale_colour_manual(values = palette_sci(
      n_distinct(V(giant)$community)
    ), guide = "none") +
    labs(title = name) +
    theme_void(base_family = "sans", base_size = 11)
})

patchwork::wrap_plots(plots, ncol = 2)
```

20.4.3 Exporting to GraphML (Gephi)

```
out_dir <- here::here("_freeze", "exports")
dir.create(out_dir, recursive = TRUE, showWarnings = FALSE)
graphml_path <- file.path(out_dir, "coauth_network.graphml")
write_graph(giant, graphml_path, format = "graphml")
cat(glue("Exported GraphML: {graphml_path}\n"))
```

```
#> Exported GraphML: /home/runner/work/scientometrics-in-r/scie
↪  ntometrics-in-r/_freeze/exports/coauth_network.graphml
```

```
cat(glue("File size: {round(file.size(graphml_path) / 1024, 1)}
↪  KB\n"))
```

```
#> File size: 6.2 KB
```

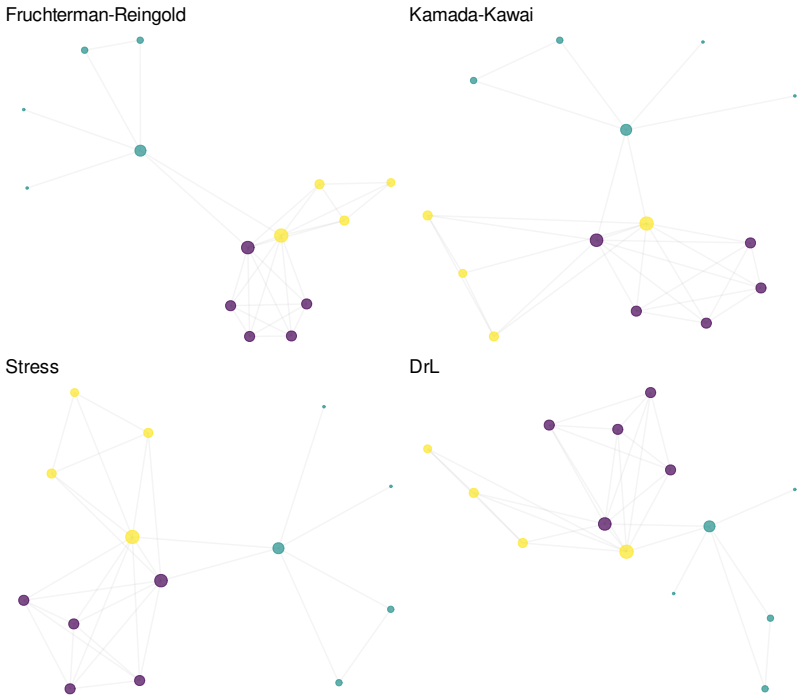


Figure 20.1: The same network rendered with four different layout algorithms.

20.4.4 Exporting to Pajek format

```
pajek_path <- file.path(out_dir, "coauth_network.net")
write_graph(giant, pajek_path, format = "pajek")
cat(glue("Exported Pajek: {pajek_path}\n"))
```

```
#> Exported Pajek: /home/runner/work/scientometrics-in-r/scient
↪  ometrics-in-r/_freeze/exports/coauth_network.net
```

20.4.5 Exporting to VOSviewer format

VOSviewer reads tab-separated map and network files.

```
set.seed(42)
coords <- layout_with_fr(giant)

vos_map <- tibble(
  id = seq_len(vcount(giant)),
  label = V(giant)$label,
  x = coords[, 1],
  y = coords[, 2],
  cluster = V(giant)$community,
  weight = V(giant)$degree
)

vos_network <- as_data_frame(giant, what = "edges") |>
  mutate(
    from_idx = match(from, V(giant)$name),
    to_idx = match(to, V(giant)$name)
  ) |>
  select(from_idx, to_idx, weight)

map_path <- file.path(out_dir, "vosviewer_map.txt")
net_path <- file.path(out_dir, "vosviewer_network.txt")

write_tsv(vos_map, map_path)
write_tsv(vos_network, net_path)
cat(glue("VOSviewer map: {map_path}\n"))
```

```
#> VOSviewer map: /home/runner/work/scientometrics-in-r/sciento
↪  metrics-in-r/_freeze/exports/vosviewer_map.txt
```

```
cat(glue("VOSviewer network: {net_path}\n"))
```

```
#> VOSviewer network: /home/runner/work/scientometrics-in-r/sci
↪  entometrics-in-r/_freeze/exports/vosviewer_network.txt
```

20.4.6 Interactive visualisation with visNetwork

```
# HTML widget — only rendered for HTML output (PDF/EPUB skip
 ↪  this chunk).
vis_nodes <- tibble(
  id = V(giant)$name,
  label = ifelse(V(giant)$degree > quantile(V(giant)$degree,
   ↪  0.9),
                 V(giant)$label, ""),
  title = paste0(V(giant)$label, "<br>Degree: ",
   ↪  V(giant)$degree,
                 "<br>Community: ", V(giant)$community),
  group = as.character(V(giant)$community),
  value = V(giant)$degree
)

vis_edges <- as_data_frame(giant, what = "edges") |>
  select(from, to, weight = weight)

visNetwork(vis_nodes, vis_edges, width = "100%", height =
 ↪  "500px") |>
  visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) |>
  visPhysics(stabilization = FALSE) |>
  visLayout(randomSeed = 42)
```

20.4.7 Publication-quality figure

```
set.seed(42)

ggraph(tg, layout = "fr") +
  geom_edge_link(alpha = 0.08, colour = "grey60") +
  geom_node_point(aes(size = degree, colour = community), alpha
   ↪  = 0.8) +
  geom_node_text(
    aes(label = ifelse(degree > quantile(degree, 0.95), label,
     ↪  NA)),
    repel = TRUE, size = 2.5, max.overlaps = 15, na.rm = TRUE
  ) +
  scale_size_continuous(range = c(1, 6), guide = "none") +
  scale_colour_manual(values = palette_sci(
    n_distinct(V(giant)$community)
  )) +
  labs(colour = "Community") +
  theme_void(base_family = "sans", base_size = 11) +
  theme(legend.position = "bottom")
```

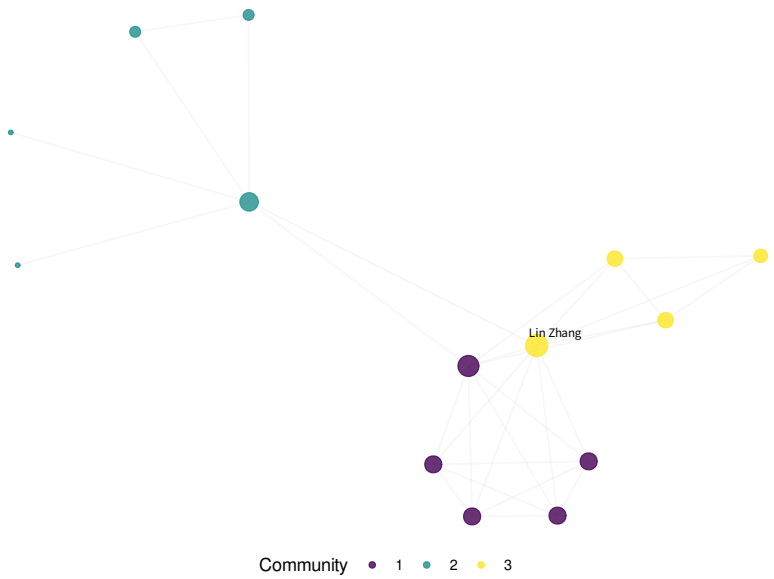


Figure 20.2: Publication-ready co-authorship network with labelled high-degree nodes.

20.5 Diagnostics and interpretation

- **Layout quality**: A good layout separates communities visually, minimises edge crossings, and distributes nodes evenly. If communities overlap heavily, try a different algorithm or increase the number of layout iterations.
- **Readability**: If the network is too dense to read, reduce the number of nodes (filter by degree), increase edge transparency, or remove edges below a weight threshold.
- **Colour distinguishability**: Test figures in greyscale and with colour-blindness simulators. Viridis-based palettes (used by `palette_sci()`) are designed for this.
- **Export verification**: After exporting, open the file in the target tool (Gephi, VOSviewer) to verify that node attributes, edge weights, and labels transferred correctly.

20.6 Limitations and responsible use

20.7 Limitations and responsible use

- **Visualisation is not analysis.** A network figure is a communication tool. The visual impression can mislead if the layout, filtering, or colour encoding obscures structure. Always report the quantitative metrics alongside figures.
- **Layout is not geography.** Absolute positions in a force-directed layout are meaningless; only relative distances carry information. Never compare positions across different layouts or runs without the same seed.
- **Large networks are unreadable.** Beyond ~500 nodes, static network plots become unintelligible. Use interactive visualisation, backbone extraction (18.4), or subnetwork extraction for large data.
- **Export formats lose information.** GraphML preserves most attributes; Pajek format is more limited. Always verify exports and document what was lost (Hicks et al., 2015).

20.8 Common pitfalls

20.9 Common pitfalls

- **Forgetting `set.seed()`.** Without a fixed seed, force-directed layouts change every render, making figures irreproducible across HTML and PDF.
- **Encoding too many variables.** Mapping degree to size, community to colour, and betweenness to transparency creates visual overload. Choose two visual channels at most.
- **Using default igraph plots.** Base R `plot.igraph()` produces low-quality figures. Use `ggraph` for publication and `visNetwork` for interactive exploration.
- **Exporting without coordinates.** If you export a network to Gephi without coordinates, Gephi will recompute the layout, producing a different figure than your R output.

20.10 Exercises

1. **Layout comparison.** Apply five different layout algorithms to the same network. Which layout best reveals the community structure? Time each layout and report the trade-off between quality and speed.
2. **Interactive filtering.** Create a `visNetwork` visualisation where users can filter by community membership. Add a dropdown selector for community.
3. **Gephi round-trip.** Export the network to GraphML, import into Gephi, apply a Force Atlas 2 layout, export as PDF. Compare the Gephi figure with the ggraph version.
4. **Colour-blind check.** Render the network figure with 3, 5, and 10 communities. At what point do the viridis colours become hard to distinguish? Test with a colour-blindness simulator.

20.11 Solutions

Solutions are provided in E.

20.12 Further reading

- Bastian et al. (2009) — Gephi: an open-source platform for network visualisation and exploration.
- Fortunato (2010) — Community detection methods with extensive visualisation examples.
- Waltman et al. (2010) — VOSviewer's approach to bibliometric network visualisation.
- Aria and Cuccurullo (2017) — `bibliometrix` includes network visualisation via `biblioshiny()`.

20.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] visNetwork_2.1.4   ggraph_2.2.2       tidygraph_1.3.1
 ↪  igraph_2.3.4
#>  [5] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1
 ↪  bibliometrix_5.5.0
#>  [9] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1
 ↪  gt_1.4.0
#> [13] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2
 ↪  lubridate_1.9.5
#> [17] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1
 ↪  purrr_1.2.2
#> [21] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1
 ↪  ggplot2_4.0.4
#> [25] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32        fs_2.1.0              vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9       contentanalysis_1.1.1  curl_8.1.0
```

```
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9            httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  SnowballC_0.7.1
#>  [37] labeling_0.4.3         urltools_1.7.3.1
 ↪  timechange_0.4.0
#>  [40] httr_1.4.9             polyclip_1.10-7
 ↪  compiler_4.6.1
#>  [43] here_1.0.2            bit64_4.8.6           withr_3.0.3
#>  [46] S7_1.0.0               viridis_0.6.5
 ↪  ggforce_0.5.0
#>  [49] MASS_7.3-65           bibliometrixData_0.3.0 tools_4.6.1
#>  [52] otel_0.3.0             stopwords_2.3           zip_3.0.2
#>  [55] rentrez_1.2.4          httpuv_1.6.17
 ↪  promises_1.5.0
#>  [58] grid_4.6.1             stringdist_0.9.17
 ↪  generics_0.1.4
#>  [61] gtable_0.3.6           tzdb_0.5.0
 ↪  rscopus_0.9.0
#>  [64] ca_0.72                data.table_1.18.6.1    hms_1.1.4
#>  [67] xml2_1.6.0             utf8_1.2.6
 ↪  ggrepel_0.9.8
#>  [70] pillar_1.11.1         vroom_1.7.1           later_1.4.8
#>  [73] tweenr_2.0.3           lattice_0.22-9         bit_4.6.0
#>  [76] tidyselect_1.2.1      miniUI_0.1.2           knitr_1.52
#>  [79] gridExtra_2.3.1       bookdown_0.48          crul_1.6.0
#>  [82] xfun_0.62              graphlayouts_1.2.5     DT_0.34.0
#>  [85] humaniformat_0.6.0    stringi_1.8.9          qpdf_1.4.1
#>  [88] yaml_2.3.12            evaluate_1.0.5
 ↪  httpcode_0.3.0
#>  [91] cli_3.6.6              xtable_1.8-8
 ↪  systemfonts_1.3.4
#>  [94] Rcpp_1.1.2             readxl_1.6.0
 ↪  triebeard_0.4.1
#>  [97] XML_3.99-0.25          parallel_4.6.1
 ↪  assertthat_0.2.1
#> [100] pubmedR_1.1.0          viridisLite_0.4.3
 ↪  scales_1.4.0
#> [103] crayon_1.5.3          openxlsx_4.2.9        rlang_1.3.0
#> [106] fastmatch_1.1-8
```

Part V

Text & Topic Analysis

Chapter 21

Text Mining Bibliographic Corpora

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

21.1 Learning objectives

After completing this chapter, you will be able to:

- Build a text corpus from OpenAlex titles and abstracts
- Apply standard preprocessing steps: tokenisation, stopword removal, stemming
- Construct a document-term matrix (DTM) and a document-feature matrix (DFM)
- Compute TF-IDF weights to identify distinctive terms
- Visualise term frequency patterns across time or groups

21.2 Setup

```
library(tidyverse)
library(openalexR)
library(quanteda)
library(quanteda.textstats)
library(glue)
```

```
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

21.3 Conceptual background

Bibliometric metadata — titles, abstracts, and keywords — constitutes a rich but noisy text corpus. Text mining transforms this unstructured data into structured representations suitable for statistical analysis. The pipeline typically follows three stages: preprocessing, representation, and analysis.

Preprocessing converts raw text into a standardised form. Tokenisation splits text into individual words or n-grams. Lowercasing removes case variation. Stopword removal eliminates high-frequency function words (“the”, “of”, “and”) that carry little topical information. Stemming or lemmatisation reduces words to their root forms (“computing”, “computed”, “computation” → “comput” or “compute”). Each step involves trade-offs: aggressive stemming can merge distinct concepts, while minimal preprocessing retains noise.

The **document-term matrix** (DTM) or **document-feature matrix** (DFM) is the fundamental representation. Rows are documents; columns are terms; cells contain counts or weights. Raw term frequencies overweight common words. **TF-IDF** (term frequency–inverse document frequency) addresses this by upweighting terms that are frequent in a document but rare across the corpus, highlighting words that distinguish one document from others.

`quanteda` (Aria and Cuccurullo, 2017) provides a fast, well-designed toolkit for text analysis in R. Its `corpus → tokens → dfm` pipeline integrates naturally with the tidyverse. For very large corpora, `quanteda` uses sparse matrix representations that scale to millions of documents.

Text mining complements network-based methods (17.3). Co-word networks reveal term associations; text mining provides the frequency distributions, temporal trends, and discriminative features that characterise a field’s vocabulary.

21.4 Worked example

21.4.1 Building a text corpus

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2019-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)

text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 50) |>
  transmute(
    doc_id = id,
    text = paste(display_name, abstract, sep = ". "),
    year = year(publication_date)
  )

cat(glue("Documents with abstracts: {nrow(text_df)}\n"))
```

```
#> Documents with abstracts: 133
```

```
corp <- corpus(text_df, docid_field = "doc_id", text_field =
 ↪  "text")
docvars(corp, "year") <- text_df$year

cat(glue("Corpus size: {ndoc(corp)} documents\n"))
```

```
#> Corpus size: 133 documents
```

```
cat(glue("Total tokens: {sum(ntoken(corp))}\n"))
```

```
#> Total tokens: 34466
```

21.4.2 Tokenisation and preprocessing

```
toks <- tokens(corp,
               remove_punct = TRUE,
               remove_numbers = TRUE,
```

```
                remove_symbols = TRUE) |>
  tokens_tolower() |>
  tokens_remove(stopwords("en")) |>
  tokens_remove(c("also", "however", "using", "based", "study",
                  "paper", "results", "research", "analysis")) |>
  tokens_wordstem()

cat(glue("Tokens after preprocessing: {sum(ntoken(toks))}\n"))
```

```
#> Tokens after preprocessing: 17199
```

21.4.3 Document-feature matrix

```
dfmat <- dfm(toks) |>
  dfm_trim(min_termfreq = 5, min_docfreq = 3)

cat(glue("DFM dimensions: {nrow(dfmat)} docs x {ncol(dfmat)}
 ↪  features\n"))
```

```
#> DFM dimensions: 133 docs x 722 features
```

```
cat(glue("Sparsity: {scales::percent(sparsity(dfmat))}\n"))
```

```
#> Sparsity: 91%
```

```
top_terms <- topfeatures(dfmat, 20) |>
  enframe(name = "term", value = "frequency") |>
  mutate(term = fct_reorder(term, frequency))

ggplot(top_terms, aes(x = frequency, y = term)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Frequency", y = NULL) +
  theme_sci()
```

21.4.4 TF-IDF weighting

```
dfmat_tfidf <- dfm_tfidf(dfmat)

tfidf_by_year <- text_df |>
```

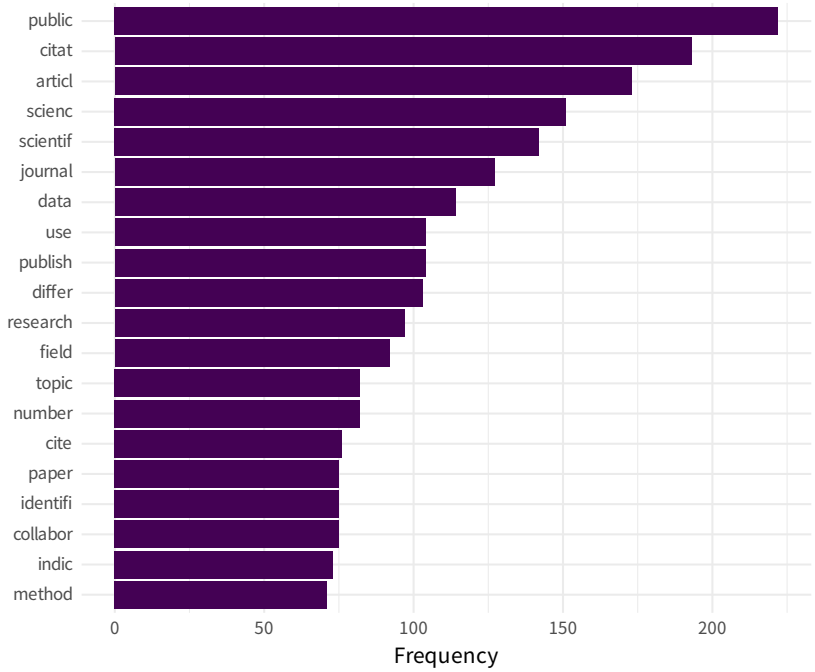


Figure 21.1: Top 20 most frequent terms in the Scientometrics corpus.

```
  group_by(year) |>
  group_keys() |>
  pull(year) |>
  map_dfr(function(yr) {
    docs <- docvars(dfmat_tfidf, "year") == yr
    if (sum(docs) < 5) return(tibble())
    top <- topfeatures(dfmat_tfidf[docs, ], 10)
    tibble(year = yr, term = names(top), tfidf = unname(top))
  })

tfidf_by_year |>
  group_by(year) |>
  slice_max(tfidf, n = 5) |>
  gt() |>
  fmt_number(columns = tfidf, decimals = 1)
```

term	tfidf
2019	
oa	16.5
promot	13.7
journal	13.6
econom	13.6
collabor	13.3
2020	
review	32.1
paper	24.6
event	22.8
citat	22.1
innov	21.5
2021	
covid-19	24.4
et	18.8
al	18.8
journal	18.5
research	17.9
2022	
univers	30.4
citat	26.2
acknowledg	24.7
tweet	22.8
scholar	22.0
2023	
oa	29.6
delay	13.7
social	12.8
scienc	12.7
articl	12.6

21.4.5 Term frequency over time

```
target_terms <- c("bibliometr", "open", "network", "impact",
 ↪  "collabor")

freq_by_year <- map_dfr(unique(text_df$year), function(yr) {
  docs <- docvars(dfmat, "year") == yr
  if (sum(docs) < 5) return(tibble())
  freq <- colSums(dfmat[docs, ]) / sum(dfmat[docs, ])
  tibble(
    year = yr,
    term = target_terms,
    rel_freq = freq[target_terms]
  )
})

ggplot(freq_by_year, aes(x = year, y = rel_freq, colour = term))
 ↪  +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  scale_colour_manual(values =
   ↪  palette_sci(length(target_terms))) +
  scale_y_continuous(labels = scales::percent_format(accuracy =
   ↪  0.01)) +
  labs(x = "Year", y = "Relative frequency", colour = "Term") +
  theme_sci()
```

21.5 Diagnostics and interpretation

- **Vocabulary size vs. documents**: A typical DFM has far more features than documents. Trim aggressively (minimum document frequency of 3–5) to reduce noise and computation.
- **Sparsity**: Bibliometric DFMs are typically 95–99% sparse. This is normal. Sparse matrix representations keep memory usage manageable.
- **Preprocessing sensitivity**: Results change with preprocessing choices. Always report stopword list, stemming method, and trimming thresholds.
- **Abstract availability**: Not all OpenAlex records have abstracts. Report the coverage rate and consider whether missing abstracts introduce bias (e.g., older papers, certain publishers).

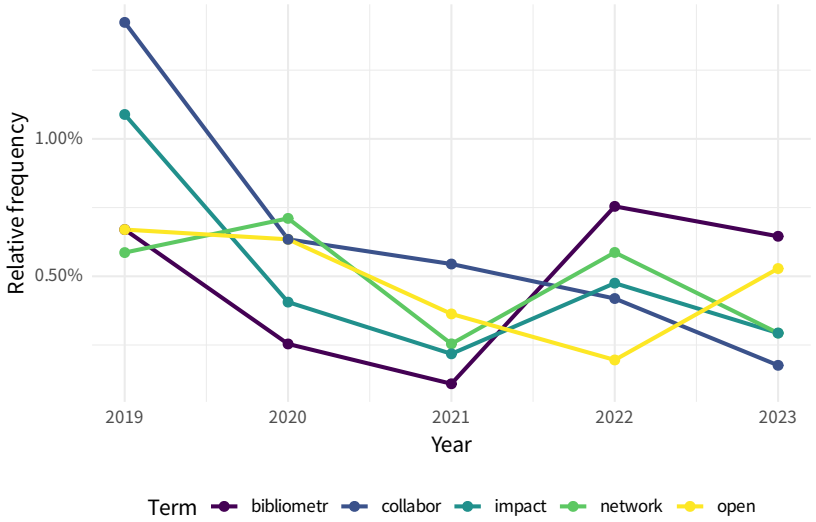


Figure 21.2: Relative frequency of selected terms over time.

21.6 Limitations and responsible use

21.7 Limitations and responsible use

- **Abstracts are summaries.** They capture the main claims but miss nuance, methodology details, and negative results. Full-text analysis (8.3) provides richer data but is harder to obtain.
- **Language bias.** Text mining tools work best for English. Non-English abstracts may be poorly tokenised, incorrectly stemmed, or excluded entirely. This biases results toward Anglophone research (Visser et al., 2021).
- **Stemming conflates concepts.** "Stem" and "stemming" are merged, but so are "cell" (biology) and "cell" (spreadsheet). Lemmatisation is more precise but slower.
- **Bag-of-words ignores context.** TF-IDF and frequency-based methods treat documents as unordered collections of words. Phrase meaning ("machine learning" vs. "learning machine") is lost unless you use n-grams or embeddings (24.3).

21.8 Common pitfalls

21.9 Common pitfalls

- **Not removing domain-specific stop words.** Generic stopword lists miss terms like “study”, “results”, “paper” that dominate academic text without conveying topical information.
- **Applying TF-IDF before trimming.** Rare terms get extreme TF-IDF scores. Trim the DFM first, then apply TF-IDF.
- **Comparing raw frequencies across groups of different sizes.** A year with 200 papers will have higher raw counts than a year with 50. Always normalise to relative frequency.
- **Stemming titles but not abstracts (or vice versa).** Apply identical preprocessing to all text fields.

21.10 Exercises

1. **Bigram analysis.** Build a DFM using bigrams (two-word sequences) instead of unigrams. What meaningful phrases emerge that single words miss?

2. **Lexical diversity.** Compute the type-token ratio (TTR) for each publication year. Is the vocabulary growing more diverse or more standardised over time?

3. **Keyness analysis.** Use `quanteda.textstats::textstat_keyness()` to identify terms that distinguish one year from all others. What terms characterise 2023 specifically?

4. **Preprocessing sensitivity.** Run the same analysis with and without stemming. How do the top terms differ? Which version is more interpretable?

21.11 Solutions

Solutions are provided in E.

21.12 Further reading

- Silge and Robinson (2017) — Text Mining with R; comprehensive guide to tidy text analysis.
- Aria and Cuccurullo (2017) — `bibliometrix` text-mining features for bibliometric data.
- Priem et al. (2022) — OpenAlex abstract availability and coverage.

21.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] quanteda.textstats_0.97.2 visNetwork_2.1.4
#>  [3] ggraph_2.2.2              tidygraph_1.3.1
#>  [5] igraph_2.3.4              quanteda_4.5.1
#>  [7] pdftools_3.9.1            arrow_26.0.1.1
#>  [9] bibliometrix_5.5.0        RefManageR_1.4.0
```

```
#> [11] bib2df_1.1.2.0           rcrossref_1.2.1
#> [13] gt_1.4.0                 tidytext_0.4.3
#> [15] glue_1.8.1               openalexR_3.1.2
#> [17] lubridate_1.9.5          forcats_1.0.1
#> [19] stringr_1.6.0            dplyr_1.2.1
#> [21] purrr_1.2.2              readr_2.2.0
#> [23] tidyr_1.3.2              tibble_3.3.1
#> [25] ggplot2_4.0.4            tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  farver_2.1.2
#>   [7] rmarkdown_2.32        fs_2.1.0              vctrs_0.7.3
#>  [10] memoise_2.0.1          askpass_1.2.1
 ↪  base64enc_0.1-6
#>  [13] htmltools_0.5.9       contentanalysis_1.1.1  curl_8.1.0
#>  [16] janeaustenr_1.0.0      cellranger_1.1.0
 ↪  htmlwidgets_1.6.4
#>  [19] tokenizers_0.3.0      plyr_1.8.9            httr2_1.3.0
#>  [22] cachem_1.1.0           plotly_4.12.1
 ↪  dimensionsR_0.0.3
#>  [25] mime_0.13              lifecycle_1.0.5
 ↪  pkgconfig_2.0.3
#>  [28] Matrix_1.7-5           R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  SnowballC_0.7.1
#>  [37] labeling_0.4.3         urltools_1.7.3.1
 ↪  timechange_0.4.0
#>  [40] httr_1.4.9             polyclip_1.10-7
 ↪  compiler_4.6.1
#>  [43] here_1.0.2            bit64_4.8.6           withr_3.0.3
#>  [46] S7_1.0.0               viridis_0.6.5
 ↪  ggforce_0.5.0
#>  [49] MASS_7.3-65           bibliometrixData_0.3.0 tools_4.6.1
#>  [52] otel_0.3.0             stopwords_2.3          zip_3.0.2
#>  [55] rentrez_1.2.4          httpuv_1.6.17
 ↪  promises_1.5.0
#>  [58] grid_4.6.1             stringdist_0.9.17
 ↪  generics_0.1.4
#>  [61] gtable_0.3.6           tzdb_0.5.0
 ↪  rscopus_0.9.0
#>  [64] ca_0.72                data.table_1.18.6.1    hms_1.1.4
#>  [67] xml2_1.6.0             utf8_1.2.6
 ↪  ggrepel_0.9.8
#>  [70] pillar_1.11.1         nsyllable_1.0.1       vroom_1.7.1
```

```
#>  [73] later_1.4.8           tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [76] bit_4.6.0             tidyselect_1.2.1
 ↪  miniUI_0.1.2
#>  [79] knitr_1.52            gridExtra_2.3.1
 ↪  bookdown_0.48
#>  [82] crul_1.6.0            xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [85] DT_0.34.0             humaniformat_0.6.0
 ↪  stringi_1.8.9
#>  [88] qpdf_1.4.1            yaml_2.3.12
 ↪  evaluate_1.0.5
#>  [91] httpcode_0.3.0       cli_3.6.6            xtable_1.8-8
#>  [94] systemfonts_1.3.4     Rcpp_1.1.2
 ↪  readxl_1.6.0
#>  [97] triebeard_0.4.1       XML_3.99-0.25
 ↪  parallel_4.6.1
#> [100] assertthat_0.2.1      pubmedR_1.1.0
 ↪  viridisLite_0.4.3
#> [103] scales_1.4.0          crayon_1.5.3
 ↪  openxlsx_4.2.9
#> [106] rlang_1.3.0           fastmatch_1.1-8
```

Chapter 22

Topic Modeling

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

22.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the generative assumptions of LDA and STM
- Prepare a document-feature matrix suitable for topic modelling
- Fit LDA and STM models and select the number of topics
- Evaluate topic quality using coherence and exclusivity metrics
- Interpret and label topics using top words and representative documents

22.2 Setup

```
library(tidyverse)
library(openalexR)
library(quanteda)
library(topicmodels)
library(tidytext)
library(stm)
library(glue)
library(gt)
```

```r
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

22.3 Conceptual background

A **topic model** is an unsupervised statistical model that discovers latent themes (topics) in a collection of documents. Each topic is a probability distribution over words; each document is a mixture of topics. The model "reverse-engineers" the writing process: given only the observed word frequencies, it infers which topics exist and how they are distributed across documents.

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is the foundational topic model. It assumes a generative process: for each document, draw a topic mixture from a Dirichlet prior; for each word position, draw a topic from that mixture, then draw a word from the topic's word distribution. Inference (fitting the model) recovers the topic-word and document-topic distributions that best explain the observed data.

The **Structural Topic Model** (STM) (Roberts et al., 2019) extends LDA by allowing document-level covariates (e.g., publication year, journal) to influence topic prevalence and word usage. This is valuable for bibliometrics because we expect topics to vary systematically with time, discipline, and publication venue. STM also estimates a "content" component that lets word usage within a topic vary by covariate.

Choosing the number of topics (K) is the most consequential modelling decision. Too few topics produce overly broad themes; too many fragment coherent themes into noise. Quantitative approaches include held-out log-likelihood, semantic coherence, and exclusivity metrics. In practice, running models at multiple K values and inspecting the resulting topics is essential.

Topic models have been widely used in scientometrics to map research fields, track thematic evolution, identify emerging areas, and compare the topical focus of journals, institutions, or countries.

22.4 Worked example

22.4.1 Preparing the corpus

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2018-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 400, seed = 42)
)

text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 100) |>
  transmute(
    doc_id = id,
    text = paste(display_name, abstract, sep = ". "),
    year = year(publication_date)
  )

cat(glue("Documents for topic modelling: {nrow(text_df)}\n"))
```

```
#> Documents for topic modelling: 98
```

```
corp <- corpus(text_df, docid_field = "doc_id", text_field =
 ↪  "text")
docvars(corp, "year") <- text_df$year

toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en")) |>
  tokens_remove(c("study", "paper", "results", "research",
   ↪  "analysis",
                  "also", "however", "using", "based"))

dfmat <- dfm(toks) |>
  dfm_trim(min_termfreq = 5, min_docfreq = 3)

cat(glue("DFM: {nrow(dfmat)} docs x {ncol(dfmat)} features\n"))
```

```
#> DFM: 98 docs x 558 features
```

22.4.2 Fitting LDA

```
dtm <- quanteda::convert(dfmat, to = "topicmodels")

lda_model <- LDA(dtm, k = 8, control = list(seed = 42))

lda_topics <- tidy(lda_model, matrix = "beta") |>
  rename(topic = topic, term = term, beta = beta)

cat(glue("LDA fitted with K = 8 topics\n"))
```

```
#> LDA fitted with K = 8 topics
```

```
lda_top_terms <- lda_topics |>
  group_by(topic) |>
  slice_max(beta, n = 10) |>
  ungroup() |>
  mutate(term = reorder_within(term, beta, topic))

ggplot(lda_top_terms, aes(x = beta, y = term, fill =
↪  factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ paste("Topic", topic), scales = "free_y", ncol =
  ↪  4) +
  scale_y_reordered() +
  scale_fill_manual(values = palette_sci(8)) +
  labs(x = "Word probability (beta)", y = NULL) +
  theme_sci(base_size = 9)
```

22.4.3 Fitting STM with year covariate

```
stm_dfm <- quanteda::convert(dfmat, to = "stm")

stm_model <- stm(
  documents = stm_dfm$documents,
  vocab = stm_dfm$vocab,
  K = 8,
  prevalence = ~ year,
  data = stm_dfm$meta,
  seed = 42,
  verbose = FALSE
)

cat(glue("STM fitted with K = 8 topics, year as prevalence
↪  covariate\n"))
```

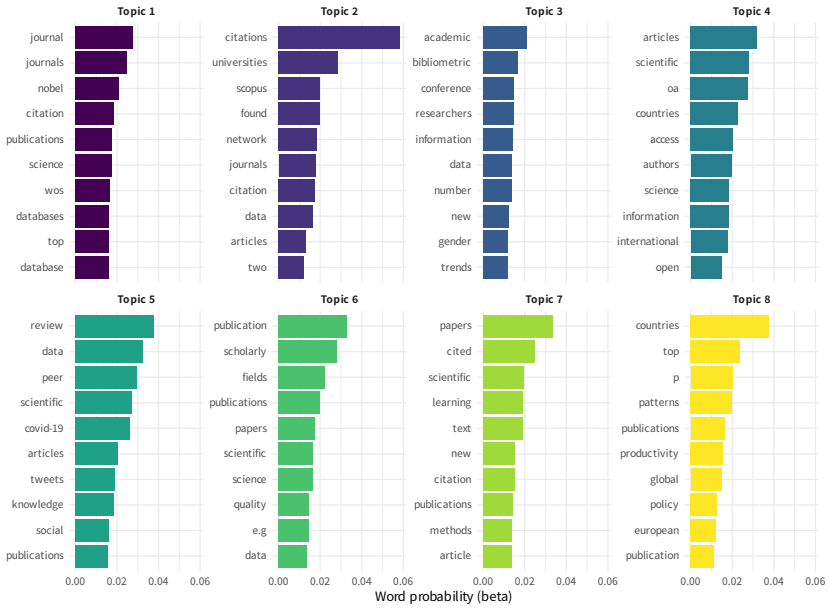


Figure 22.1: Top 10 terms per LDA topic.

```
#> STM fitted with K = 8 topics, year as prevalence covariate
```

```
plot(stm_model, type = "summary", n = 5,
     main = "STM Topic Proportions and Top FREX Terms")
```

22.4.4 Topic prevalence over time

```
effect <- estimateEffect(1:8 ~ year, stm_model, meta =
 ↪  stm_dfm$meta)

effect_df <- map_dfr(1:8, function(k) {
  eff <- summary(effect, topics = k)$tables[[1]]
  tibble(
    topic = k,
    term = c("intercept", "year"),
    estimate = eff[, "Estimate"],
    se = eff[, "Std. Error"]
  )
```

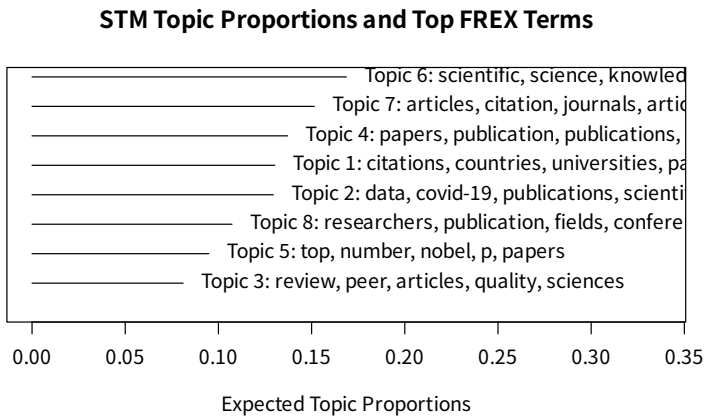


Figure 22.2: Top terms per STM topic, showing FREX (frequent and exclusive) words.

```
}) |>
  filter(term == "year")

ggplot(effect_df, aes(x = factor(topic), y = estimate)) +
  geom_col(fill = palette_sci(1)) +
  geom_errorbar(aes(ymin = estimate - 1.96 * se, ymax = estimate
  ↪  + 1.96 * se),
                width = 0.3) +
  geom_hline(yintercept = 0, linetype = "dashed", colour =
  ↪  "grey50") +
  labs(x = "Topic", y = "Year effect on prevalence") +
  theme_sci()
```

22.4.5 Model diagnostics

```
coherence <- semanticCoherence(stm_model, stm_dfm$documents)
exclusivity <- exclusivity(stm_model)

diag_df <- tibble(
  topic = 1:8,
  coherence = round(coherence, 2),
  exclusivity = round(exclusivity, 2)
)

diag_df |> gt()
```

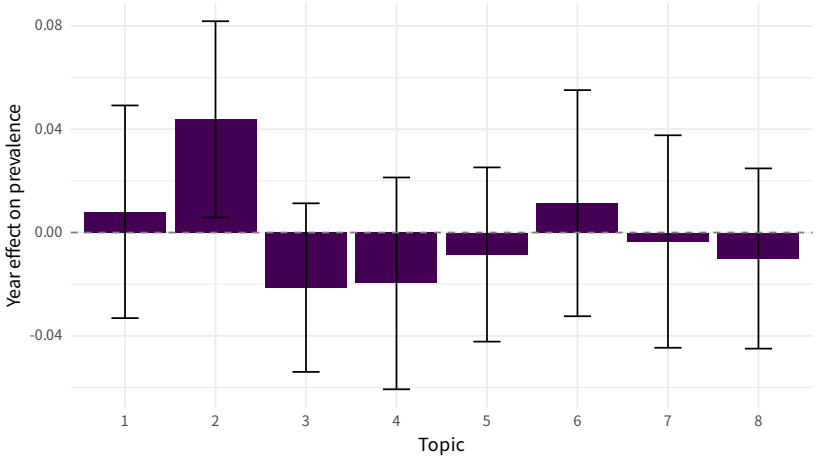


Figure 22.3: Estimated topic prevalence by publication year (STM).

topic	coherence	exclusivity
1	-97.4	9.35
2	-82.0	9.35
3	-94.8	9.65
4	-105.1	9.37
5	-95.6	9.54
6	-77.8	9.01
7	-77.8	9.17
8	-119.7	9.22

22.5 Diagnostics and interpretation

- **Coherence**: High coherence means the top words in a topic tend to co-occur in the same documents. Low coherence indicates a "junk" topic mixing unrelated terms.
- **Exclusivity**: High exclusivity means the top words are distinctive to that topic. Topics with low exclusivity share vocabulary with other topics, suggesting they may need to be split.
- **Coherence-exclusivity trade-off**: Adding more topics generally improves exclusivity but may decrease coherence. Plot both metrics across K values to find a balance.
- **Human validation**: Always read sample documents assigned to each topic. If a topic cannot be labelled by a domain expert, it may not represent a meaningful theme.

22.6 Limitations and responsible use

22.7 Limitations and responsible use

- **Topics are not ground truth.** Topic models are statistical constructs, not natural categories. Different random seeds, preprocessing choices, or K values produce different topics. Never claim a topic model reveals "the" structure of a field.
- **Abstract-only models miss depth.** Abstracts are compressed summaries. Topics derived from abstracts are coarser than those from full text.
- **Number of topics is a researcher choice.** There is no objectively correct K. Sensitivity analysis across multiple K values is essential.
- **Temporal confounds.** In STM, a topic that "grows" over time may reflect genuine intellectual shift or simply increased coverage by the database in recent years (Priem et al., 2022).
- **Replication.** Topic models are sensitive to preprocessing and random initialisation. Report all parameters and fix seeds for reproducibility.

22.8 Common pitfalls

22.9 Common pitfalls

- **Not preprocessing enough.** Rare terms, numbers, and generic academic vocabulary produce noisy topics. Trim the DFM before fitting.
- **Choosing K by perplexity alone.** Held-out perplexity often favours large K values that produce uninterpretable topics. Use coherence and exclusivity alongside perplexity.
- **Treating topic proportions as precise measurements.** Document-topic proportions are probabilistic estimates with uncertainty. Do not report them to three decimal places.
- **Comparing topics across models.** LDA and STM may number and define topics differently. Align topics by content, not by index.

22.10 Exercises

1. **K selection.** Fit LDA models with K = 5, 8, 12, and 15. Compute coherence for each. Plot coherence against K and select the best value. How sensitive are the results?

2. **STM with journal covariate.** Fetch data from two different journals. Fit an STM with journal as a prevalence covariate. Which topics differ most between journals?
3. **Topic labelling.** For the best LDA model, read the titles of the 5 documents with the highest proportion for each topic. Assign a human-readable label to each topic.
4. **LDA vs. STM comparison.** Compare the topics from LDA and STM for the same K. Are they similar? What does the STM's year covariate add?

22.11 Solutions

Solutions are provided in E.

22.12 Further reading

- Blei et al. (2003) — The foundational LDA paper.
- Roberts et al. (2019) — The Structural Topic Model with covariates.
- Silge and Robinson (2017) — Text Mining with R: accessible guide to topic modelling in R.
- Aria and Cuccurullo (2017) — `bibliometrix` topic analysis features.

22.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
```

```
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] stm_1.3.10                topicmodels_0.2-17
#>  [3] quanteda.textstats_0.97.2 visNetwork_2.1.4
#>  [5] ggraph_2.2.2              tidygraph_1.3.1
#>  [7] igraph_2.3.4              quanteda_4.5.1
#>  [9] pdftools_3.9.1            arrow_26.0.1.1
#> [11] bibliometrix_5.5.0        RefManageR_1.4.0
#> [13] bib2df_1.1.2.0            rcrossref_1.2.1
#> [15] gt_1.4.0                  tidytext_0.4.3
#> [17] glue_1.8.1                openalexR_3.1.2
#> [19] lubridate_1.9.5           forcats_1.0.1
#> [21] stringr_1.6.0             dplyr_1.2.1
#> [23] purrr_1.2.2               readr_2.2.0
#> [25] tidyr_1.3.2               tibble_3.3.1
#> [27] ggplot2_4.0.4             tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4      tokenizers_0.3.0       plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        Matrix_1.7-5           R6_2.6.1
#>  [31] fastmap_1.2.0          shiny_1.15.0
 ↪  digest_0.6.39
#>  [34] patchwork_1.3.2        shinycssloaders_1.1.0
 ↪  rprojroot_2.1.1
```

```
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       httr_1.4.9
 ↪  polyclip_1.10-7
#>  [43] compiler_4.6.1        here_1.0.2            bit64_4.8.6
#>  [46] withr_3.0.3            S7_1.0.0
 ↪  viridis_0.6.5
#>  [49] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [52] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [55] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
#>  [58] promises_1.5.0         grid_4.6.1
 ↪  stringdist_0.9.17
#>  [61] reshape2_1.4.5         generics_0.1.4
 ↪  gtable_0.3.6
#>  [64] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [67] data.table_1.18.6.1    hms_1.1.4             xml2_1.6.0
#>  [70] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [73] nsyllable_1.0.1       vroom_1.7.1           later_1.4.8
#>  [76] tweenr_2.0.3           lattice_0.22-9         bit_4.6.0
#>  [79] tidyselect_1.2.1       tm_0.7-20
 ↪  miniUI_0.1.2
#>  [82] knitr_1.52             gridExtra_2.3.1        NLP_0.3-3
#>  [85] bookdown_0.48         stats4_4.6.1           crul_1.6.0
#>  [88] xfun_0.62              graphlayouts_1.2.5
 ↪  matrixStats_1.5.0
#>  [91] DT_0.34.0              humaniformat_0.6.0
 ↪  stringi_1.8.9
#>  [94] qpdf_1.4.1             yaml_2.3.12
 ↪  evaluate_1.0.5
#>  [97] httpcode_0.3.0        cli_3.6.6             xtable_1.8-8
#> [100] systemfonts_1.3.4      Rcpp_1.1.2
 ↪  readxl_1.6.0
#> [103] triebeard_0.4.1        XML_3.99-0.25
 ↪  parallel_4.6.1
#> [106] assertthat_0.2.1      pubmedR_1.1.0          slam_0.1-56
#> [109] viridisLite_0.4.3      scales_1.4.0
 ↪  crayon_1.5.3
#> [112] openxlsx_4.2.9         rlang_1.3.0
 ↪  fastmatch_1.1-8
```

Chapter 23

Keyword Extraction

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

23.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the RAKE and TextRank keyword extraction algorithms
- Apply keyword extraction to bibliometric abstracts using R packages
- Compare automatically extracted keywords with author-assigned keywords
- Evaluate extraction quality using precision, recall, and F1 metrics
- Decide when automatic keyword extraction adds value over existing metadata

23.2 Setup

```
library(tidyverse)
library(openalexR)
library(quanteda)
library(quanteda.textstats)
library(glue)
library(gt)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "rake_keywords.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

23.3 Conceptual background

Keywords serve as compact descriptors of a document's content. In bibliometrics, they are used for co-word analysis (17.3), topic mapping, and search. Three sources of keywords exist:

1. **Author keywords**: Selected by the paper's authors. Intentional but inconsistent — different authors use different terms for the same concept.
2. **Indexed terms**: Assigned by database curators (MeSH in PubMed, descriptors in WoS). Consistent but labour-intensive and available only in some databases.
3. **Extracted keywords**: Automatically derived from titles and abstracts using algorithms. Scalable and consistent but may miss domain nuance.

RAKE (Rapid Automatic Keyword Extraction) identifies candidate keywords by finding sequences of content words delimited by stopwords or punctuation. It scores candidates by the ratio of word weighted degree (including the word itself within candidate phrases) to word frequency, favouring multi-word phrases that appear in specific contexts. RAKE is fast and unsupervised but does not consider document-level or corpus-level statistics.

TextRank adapts the PageRank algorithm to text: words are nodes in a graph, edges connect co-occurring words within a sliding window, and importance is computed iteratively. High-ranking words are selected as keywords. TextRank captures contextual importance better than frequency-based methods but is more computationally expensive.

Both methods extract keywords from individual documents. **TF-IDF** (21.3) provides a complementary corpus-level view, identifying terms that distinguish a document from the broader collection.

Evaluating keyword extraction requires ground truth. Author keywords are the most common benchmark, but they are imperfect: authors may omit obvious keywords or include overly specific terms. Agreement depends on the corpus, matching rules, number of keywords, and reference annotation. Define these choices before evaluation and report observed precision and recall; no universal acceptance percentage is assumed here.

23.4 Worked example

23.4.1 Fetching data with keywords

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2021-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 200, seed = 42)
)

text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 100) |>
  transmute(
    doc_id = id,
    title = display_name,
    abstract = abstract,
    text = paste(display_name, abstract, sep = ". ")
  )

cat(glue("Documents with abstracts: {nrow(text_df)}\n"))
```

```
#> Documents with abstracts: 56
```

23.4.2 RAKE keyword extraction

```
rake_extract <- function(text, n_keywords = 10) {
  rake_keywords(text, stopwords = quanteda::stopwords("en"),
                n_keywords = n_keywords)$keyword
}

text_df <- text_df |>
  mutate(rake_keywords = map(text, \(t) rake_extract(t,
   ↪  n_keywords = 8)))

text_df |>
  head(3) |>
  select(title, rake_keywords) |>
  mutate(rake_keywords = map_chr(rake_keywords, paste, collapse
   ↪  = "; ")) |>
  gt()
```

title	rake_keywords
Visibility, impact, and applications of bibliometric software tools through citation analysis	identify specific research areas; eight bibliometric software tools; bibliometric software tools; cited reference search; open access level; wos core collection; citing research articles; software tools
Philosophers' appraisals of bibliometric indicators and their use in evaluation: from recognition to knee-jerk rejection	researchers regarding bibliometric indicators; fairly high self-perception; four main positions; little explored areas; qualitative data points; self-administered questionnaire completed; survey data suggest; researchers selected according
Publication outperformance among global South researchers: An analysis of individual-level and publication-level predictors of positive deviance	developed using partial least squares regression; publication outperformance among global south researchers; mixed-methods approach involving interviews; research performance lags behind; among southern researchers; challenges facing researchers; information systems researchers; publication outlets targeted

23.4.3 TF-IDF based keyword extraction

```
corp <- corpus(text_df, docid_field = "doc_id", text_field =
 ↪  "text")

toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en")) |>
  tokens_remove(c("study", "paper", "results", "research",
   ↪  "analysis"))

dfmat <- dfm(toks) |>
  dfm_trim(min_termfreq = 2)
dfmat_tfidf <- dfm_tfidf(dfmat)

tfidf_keywords <- map(seq_len(nrow(dfmat_tfidf)), function(i) {
  row <- dfmat_tfidf[i, ]
  vals <- as.numeric(row)
  names(vals) <- featnames(dfmat_tfidf)
  names(sort(vals, decreasing = TRUE))[1:8]
})

text_df$tfidf_keywords <- tfidf_keywords
```

23.4.4 Corpus-level keyword frequency

```
all_kw <- text_df |>
  pull(rake_keywords) |>
  unlist() |>
  tibble(keyword = _) |>
  count(keyword, sort = TRUE)

all_kw |>
  head(20) |>
  mutate(keyword = fct_reorder(keyword, n)) |>
  ggplot(aes(x = n, y = keyword)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Extraction frequency", y = NULL) +
  theme_sci()
```

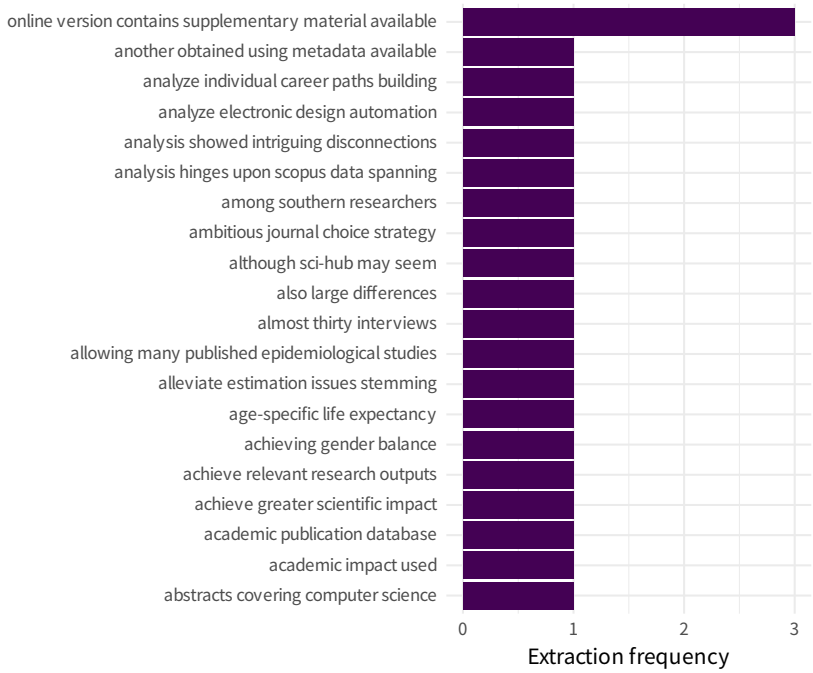


Figure 23.1: Top 20 keywords by total extraction frequency across the corpus.

23.4.5 Comparing extraction methods

```
overlap <- map2_dbl(text_df$rake_keywords,
 ↪  text_df$tfidf_keywords,
                     \(r, t) length(intersect(r, t)) /
 ↪  length(union(r, t)))

cat(glue("Mean Jaccard overlap (RAKE vs TF-IDF):
 ↪  {round(mean(overlap, na.rm = TRUE), 3)}\n"))
```

```
#> Mean Jaccard overlap (RAKE vs TF-IDF): 0.001
```

```
tibble(jaccard = overlap) |>
  ggplot(aes(x = jaccard)) +
  geom_histogram(binwidth = 0.05, fill = palette_sci(1), colour
   ↪  = "white") +
  labs(x = "Jaccard similarity", y = "Documents") +
  theme_sci()
```

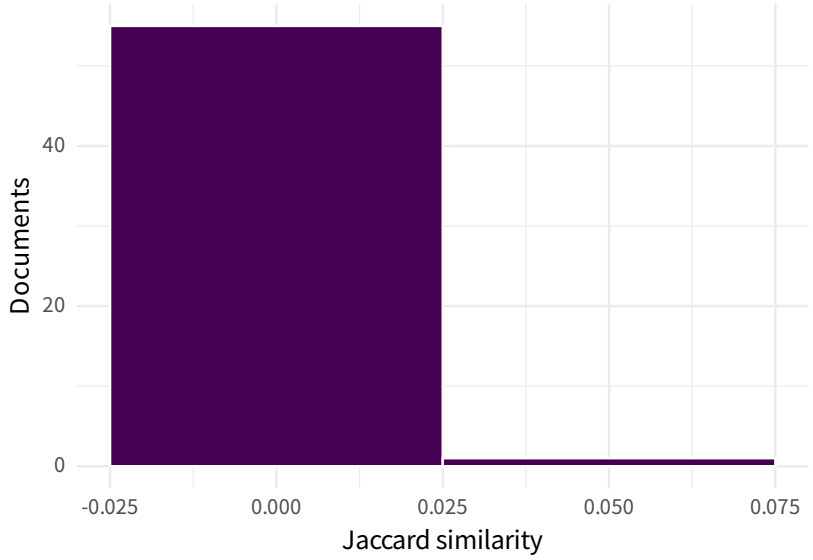


Figure 23.2: Distribution of Jaccard similarity between RAKE and TF-IDF keyword sets.

23.5 Diagnostics and interpretation

- **Extraction quality**: Compare extracted keywords against author keywords for a random sample of 20–30 papers. Specify exact or normalized matching, report precision and recall, and inspect disagreements rather than applying an unsupported universal threshold.
- **Redundancy**: Check for near-duplicate keywords (singular/plural, abbreviation/full form). Post-processing to merge variants improves quality.
- **Domain specificity**: Generic terms ("model", "method", "data") are often extracted but carry no topical information. Consider a domain-specific stopword list.
- **Coverage**: Some documents yield few meaningful keywords, especially short abstracts. Flag documents where fewer than 3 keywords were extracted.

23.6 Limitations and responsible use

23.7 Limitations and responsible use

- **No method is comprehensive.** Keyword extraction captures salient terms but misses concepts expressed through circumlocution or implicit reference. Human keywords remain valuable.
- **Language dependence.** RAKE and TextRank work best for English. Stopword lists and tokenisation rules may fail for other languages.
- **Evaluation is subjective.** There is no single correct keyword set for a document. Low agreement between methods or between extracted and author keywords does not necessarily indicate failure.
- **Not a replacement for indexing.** Professional indexing (e.g., MeSH) follows controlled vocabularies and domain expertise. Automated extraction is a complement, not a substitute (Hicks et al., 2015).

23.8 Common pitfalls

23.9 Common pitfalls

- **Using raw frequencies as keywords.** The most frequent words in a document are often stopwords or near-stopwords. TF-IDF or RAKE scoring is essential.

- **Not normalising keywords.** "bibliometric" and "bibliometrics" should be the same keyword. Apply stemming or lemmatisation before evaluation.
- **Evaluating on titles only.** Titles are too short for reliable keyword extraction. Use abstracts or title + abstract combined.
- **Ignoring multi-word terms.** Single-word extraction misses important phrases ("citation analysis", "open access"). Use n-grams or phrase detection.

23.10 Exercises

1. **Precision and recall.** For 20 papers with author keywords, compute precision and recall of RAKE-extracted keywords against author keywords. Report the mean and range.

2. **Bigram keywords.** Extract bigrams from abstracts and score them by TF-IDF. What multi-word terms emerge that single-word extraction misses?

3. **Keyword enrichment.** For papers without author keywords, use extracted keywords to build a co-word network. Does the resulting topical structure resemble that from author keywords?

23.11 Solutions

Solutions are provided in E.

23.12 Further reading

- Silge and Robinson (2017) — Keyword and term extraction in R with tidy tools.
- Aria and Cuccurullo (2017) — `bibliometrix` keyword analysis capabilities.
- Callon et al. (1983) — Co-word analysis, the downstream use case for extracted keywords.
- Priem et al. (2022) — OpenAlex concept tagging as an alternative to keyword extraction.

23.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] stm_1.3.10                topicmodels_0.2-17
#>  [3] quanteda.textstats_0.97.2 visNetwork_2.1.4
#>  [5] ggraph_2.2.2              tidygraph_1.3.1
#>  [7] igraph_2.3.4              quanteda_4.5.1
#>  [9] pdftools_3.9.1            arrow_26.0.1.1
#> [11] bibliometrix_5.5.0        RefManageR_1.4.0
#> [13] bib2df_1.1.2.0            rcrossref_1.2.1
#> [15] gt_1.4.0                  tidytext_0.4.3
#> [17] glue_1.8.1                openalexR_3.1.2
#> [19] lubridate_1.9.5           forcats_1.0.1
#> [21] stringr_1.6.0             dplyr_1.2.1
#> [23] purrr_1.2.2               readr_2.2.0
#> [25] tidyr_1.3.2               tibble_3.3.1
#> [27] ggplot2_4.0.4             tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
```

```
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4      tokenizers_0.3.0       plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        Matrix_1.7-5           R6_2.6.1
#>  [31] fastmap_1.2.0          shiny_1.15.0
 ↪  digest_0.6.39
#>  [34] patchwork_1.3.2        shinycssloaders_1.1.0
 ↪  rprojroot_2.1.1
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       httr_1.4.9
 ↪  polyclip_1.10-7
#>  [43] compiler_4.6.1         here_1.0.2             bit64_4.8.6
#>  [46] withr_3.0.3            S7_1.0.0
 ↪  viridis_0.6.5
#>  [49] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [52] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [55] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
#>  [58] promises_1.5.0         grid_4.6.1
 ↪  stringdist_0.9.17
#>  [61] reshape2_1.4.5         generics_0.1.4
 ↪  gtable_0.3.6
#>  [64] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [67] data.table_1.18.6.1    hms_1.1.4              xml2_1.6.0
#>  [70] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [73] nsyllable_1.0.1        vroom_1.7.1            later_1.4.8
#>  [76] tweenr_2.0.3           lattice_0.22-9         bit_4.6.0
#>  [79] tidyselect_1.2.1       tm_0.7-20
 ↪  miniUI_0.1.2
#>  [82] knitr_1.52             gridExtra_2.3.1        NLP_0.3-3
#>  [85] bookdown_0.48          stats4_4.6.1           crul_1.6.0
#>  [88] xfun_0.62              graphlayouts_1.2.5
 ↪  matrixStats_1.5.0
```

```
#>  [91] DT_0.34.0              humaniformat_0.6.0
 ↪  stringi_1.8.9
#>  [94] qpdf_1.4.1             yaml_2.3.12
 ↪  evaluate_1.0.5
#>  [97] httpcode_0.3.0        cli_3.6.6            xtable_1.8-8
#> [100] systemfonts_1.3.4      Rcpp_1.1.2
 ↪  readxl_1.6.0
#> [103] triebeard_0.4.1        XML_3.99-0.25
 ↪  parallel_4.6.1
#> [106] assertthat_0.2.1      pubmedR_1.1.0         slam_0.1-56
#> [109] viridisLite_0.4.3      scales_1.4.0
 ↪  crayon_1.5.3
#> [112] openxlsx_4.2.9         rlang_1.3.0
 ↪  fastmatch_1.1-8
```

Chapter 24

Embeddings for Scientometrics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

24.1 Learning objectives

After completing this chapter, you will be able to:

- Explain how word and document embeddings differ from bag-of-words representations
- Train simple word embeddings on a bibliometric corpus
- Use pre-trained embeddings to compute document similarity
- Apply UMAP or t-SNE to visualise document clusters in embedding space
- Assess when embeddings add value over simpler text representations

24.2 Setup

```
library(tidyverse)
library(openalexR)
library(quanteda)
library(word2vec)
library(uwot)
library(glue)
```

```
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

24.3 Conceptual background

Bag-of-words representations (21.3) treat each word as an independent dimension. “Citation analysis” and “bibliometric study” share no features despite being semantically related. **Word embeddings** address this by mapping words into a dense, low-dimensional vector space where semantically similar words are close together. The word2vec algorithm learns these representations by predicting words from their context (skip-gram) or context from words (CBOW) in a large training corpus.

Document embeddings extend the idea to whole documents. Simple approaches average the word vectors of a document’s words (mean pooling). More sophisticated approaches include Doc2Vec (Paragraph Vector), which learns document-specific vectors alongside word vectors, and transformer-based models like **SPECTER**, which is pre-trained on scientific paper titles and abstracts using citation-based training signals.

For scientometrics, embeddings enable several applications:

- **Semantic similarity**: Finding papers with similar meaning, even when they use different terminology.
- **Clustering**: Grouping papers by content in embedding space, complementing topic models (22.3).
- **Visualisation**: Projecting embeddings to 2D with UMAP or t-SNE to create interpretable “maps” of a corpus.
- **Anomaly detection**: Identifying papers that are semantically distant from their assigned field or cluster.

The trade-off is interpretability: while TF-IDF features are directly readable (the word “citation” has a specific weight), embedding dimensions are opaque. Embeddings are powerful for similarity and clustering but resist the kind of term-level interpretation that co-word analysis provides.

24.4 Worked example

24.4.1 Preparing text data

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2019-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 400, seed = 42)
)

text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 100) |>
  transmute(
    doc_id = id,
    title = display_name,
    text = paste(display_name, abstract, sep = ". "),
    year = year(publication_date),
    cited_by_count
  )

cat(glue("Documents for embedding: {nrow(text_df)}\n"))
```

```
#> Documents for embedding: 107
```

24.4.2 Training word2vec embeddings

```
clean_text <- text_df$text |>
  str_to_lower() |>
  str_replace_all("[^a-z ]", " ") |>
  str_squish()

model <- word2vec(clean_text, dim = 100, iter = 20, min_count =
↪  3,
                  type = "skip-gram", threads = 1)

cat(glue("Vocabulary size: {nrow(as.matrix(model))}\n"))
```

```
#> Vocabulary size: 1417
```

```
if ("citation" %in% rownames(as.matrix(model))) {
  nn <- predict(model, newdata = "citation", type = "nearest",
 ↪  top_n = 10)
  cat("Nearest neighbours of 'citation':\n")
  print(nn)
}
```

```
#> Nearest neighbours of 'citation':
#> $citation
#>       term1          term2 similarity rank
#> 1  citation     normalised      0.734    1
#> 2  citation        similar      0.681    2
#> 3  citation recommendation      0.668    3
#> 4  citation           cnci      0.653    4
#> 5  citation             pp      0.650    5
#> 6  citation          mixed      0.650    6
#> 7  citation  bibliometrics      0.647    7
#> 8  citation         survey      0.646    8
#> 9  citation       dominate      0.638    9
#> 10 citation            six      0.637   10
```

24.4.3 Computing document embeddings via mean pooling

Apply the same text preprocessing used for training: lowercase before removing characters outside the vocabulary alphabet. This preserves capitalised words and acronyms when matching them to the learned word vectors.

```
word_matrix <- as.matrix(model)

doc_embed <- function(text, wm) {
  words <- str_split(str_replace_all(str_to_lower(text), "[^a-z
 ↪  ]", " "),
                     "\\s+")[[1]]
  matched <- words[words %in% rownames(wm)]
  if (length(matched) == 0) return(rep(NA_real_, ncol(wm)))
  colMeans(wm[matched, , drop = FALSE])
}

doc_embeddings <- map(text_df$text, \(t) doc_embed(t,
 ↪  word_matrix))
embed_mat <- do.call(rbind, doc_embeddings)

valid <- complete.cases(embed_mat)
embed_mat <- embed_mat[valid, ]
```

```
text_df_valid <- text_df[valid, ]

cat(glue("Valid document embeddings: {nrow(embed_mat)}\n"))
```

```
#> Valid document embeddings: 107
```

24.4.4 Dimensionality reduction with UMAP

```
umap_result <- umap(embed_mat, n_neighbors = 15, min_dist = 0.1,
                    n_components = 2, ret_model = FALSE)

text_df_valid$umap_x <- umap_result[, 1]
text_df_valid$umap_y <- umap_result[, 2]
```

```
ggplot(text_df_valid, aes(x = umap_x, y = umap_y, colour =
 ↪  factor(year))) +
  geom_point(alpha = 0.6, size = 1.5) +
  scale_colour_manual(values =
   ↪  palette_sci(n_distinct(text_df_valid$year))) +
  labs(x = "UMAP 1", y = "UMAP 2", colour = "Year") +
  theme_sci() +
  theme(axis.text = element_blank(), axis.ticks =
   ↪  element_blank())
```

24.4.5 Document similarity search

```
cosine_sim <- function(a, b) sum(a * b) / (sqrt(sum(a^2)) *
 ↪  sqrt(sum(b^2)))

query_idx <- 1
sims <- map_dbl(seq_len(nrow(embed_mat)),
                \(i) cosine_sim(embed_mat[query_idx, ],
 ↪  embed_mat[i, ]))

top_similar <- order(sims, decreasing = TRUE)[2:6]

cat(glue("Query: {text_df_valid$title[query_idx]}\n\n"))
```

```
#> Query: The world-systemic network of global elite sociology:
 ↪  the western male monoculture at faculties of the top
 ↪  one-hundred sociology departments of the world
```

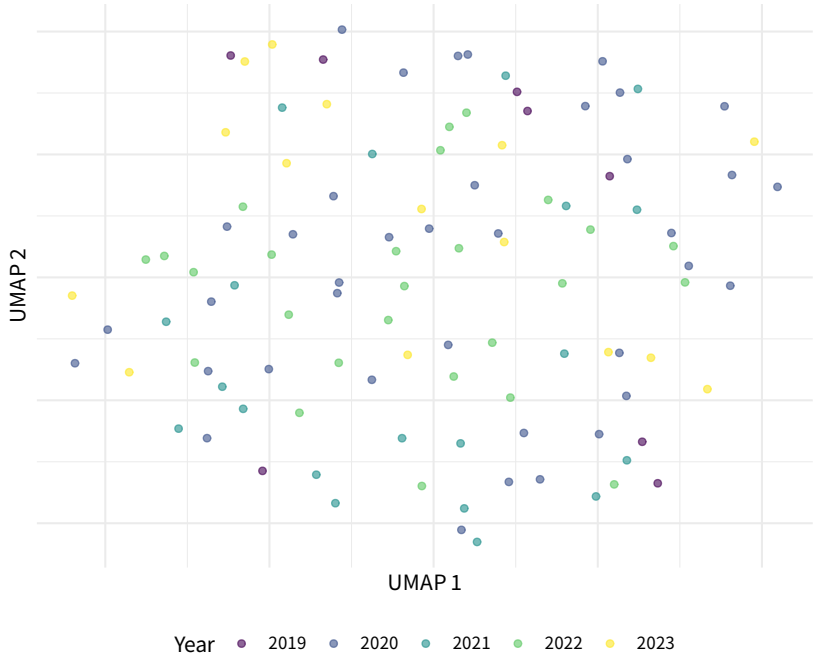


Figure 24.1: UMAP projection of document embeddings, coloured by publication year.

```
cat("Most similar documents:\n")
```

```
#> Most similar documents:
```

```
walk(top_similar, \(i) {
  cat(glue("  [{round(sims[i], 3)}]
   ↪  {text_df_valid$title[i]}\n"))
})
```

```
#> [0.904] Coronavirus mapping in scientific publications: When
 ↪  science advances rapidly and collectively, is access to this
 ↪  knowledge open to society?[0.904] An analysis of
 ↪  international mobility and research productivity in computer
 ↪  science[0.904] Publication outperformance among global South
 ↪  researchers: An analysis of individual-level and
 ↪  publication-level predictors of positive deviance[0.894]
 ↪  Analysis of scientific collaboration network of Italian
 ↪  Institute of Technology[0.892] National and organizational
 ↪  patterns of Nobel laureate careers in physiology/medicine,
 ↪  physics, and chemistry
```

24.4.6 Embedding clusters vs. citation impact

```
ggplot(text_df_valid, aes(x = umap_x, y = umap_y,
                          colour = log1p(cited_by_count))) +
  geom_point(alpha = 0.6, size = 1.5) +
  scale_colour_viridis_c(option = "C", name = "log(cites + 1)") +
  labs(x = "UMAP 1", y = "UMAP 2") +
  theme_sci() +
  theme(axis.text = element_blank(), axis.ticks =
   ↪  element_blank())
```

24.5 Diagnostics and interpretation

- **Vocabulary coverage**: Check what fraction of abstract words appear in the embedding vocabulary. Low coverage (< 80%) indicates the training corpus is too small or preprocessing is too aggressive.
- **Nearest neighbours**: Inspect the nearest neighbours of key domain terms. If “citation” is closest to “references” and “bibliometric”, the embeddings capture domain structure.

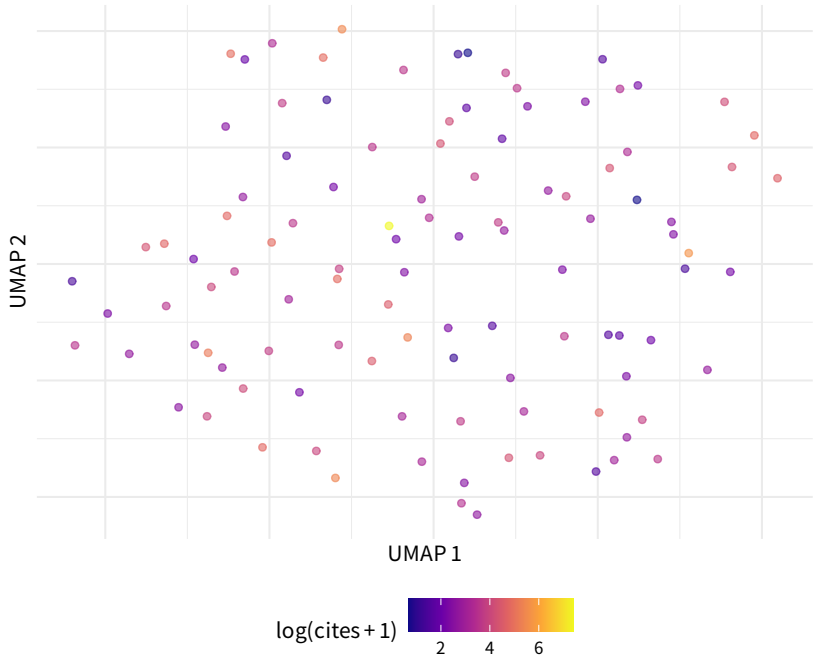


Figure 24.2: UMAP embedding coloured by citation impact (log scale).

- **UMAP stability**: UMAP is stochastic. Run with multiple seeds and check whether the global cluster structure is stable. Individual point positions will vary but clusters should persist.
- **Embedding dimensionality**: 100–300 dimensions is standard. Below 50, embeddings lose semantic resolution; above 300, training becomes slow with diminishing returns.

24.6 Limitations and responsible use

24.7 Limitations and responsible use

- **Embeddings are opaque.** Unlike TF-IDF weights, embedding dimensions have no interpretable meaning. You can measure similarity but not explain why two documents are similar without additional analysis.
- **Training data bias.** Embeddings reflect the biases of their training corpus. If the corpus overrepresents certain topics or perspectives, the embeddings will too.
- **Small corpora produce poor embeddings.** Word2vec needs tens of thousands of documents for reliable training. For small bibliometric corpora (< 1,000 papers), use pre-trained embeddings instead.
- **UMAP/t-SNE distort distances.** These methods preserve local structure but distort global distances. Documents that appear close in 2D may not be the most similar in high-dimensional space (Hicks et al., 2015).

24.8 Common pitfalls

24.9 Common pitfalls

- **Training on too few documents.** Word2vec on 200 abstracts produces unreliable embeddings. Use at least 5,000 documents or switch to pre-trained models.
- **Averaging without weighting.** Simple mean pooling treats every word equally. TF-IDF-weighted averaging gives more weight to distinctive terms and often produces better document embeddings.
- **Over-interpreting UMAP clusters.** Visual clusters in UMAP may reflect projection artifacts. Validate clusters with a separate method (e.g., k-means in the full embedding space).

- **Comparing embeddings from different models.** Embedding spaces are not aligned across different training runs. Cosine similarity is only meaningful within a single embedding space.

24.10 Exercises

1. **Weighted averaging.** Compute document embeddings using TF-IDF-weighted word vectors instead of simple means. Does the UMAP visualisation change?
2. **K-means clustering.** Apply k-means (k = 5, 8, 12) to the document embeddings. Compare the resulting clusters with LDA topics from 22.4. Do they agree?
3. **Pre-trained embeddings.** If available, use SPECTER embeddings (via reticulate and Python) instead of corpus-trained word2vec. How does similarity search quality change?
4. **Analogy test.** Test whether the word2vec model captures analogical relationships (e.g., “journal” - “article” + “book” ≈ “publisher”). What does the result tell you about the model?

24.11 Solutions

Solutions are provided in E.

24.12 Further reading

- Silge and Robinson (2017) — Text embeddings in R, including integration with word2vec.
- Priem et al. (2022) — OpenAlex concept embeddings and their relationship to document content.
- Waltman et al. (2010) — Bibliometric similarity measures as context for embedding-based approaches.

24.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
#>  [5] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4         ggraph_2.2.2
#>  [9] tidygraph_1.3.1          igraph_2.3.4
#> [11] quanteda_4.5.1           pdftools_3.9.1
#> [13] arrow_26.0.1.1           bibliometrix_5.5.0
#> [15] RefManageR_1.4.0         bib2df_1.1.2.0
#> [17] rcrossref_1.2.1          gt_1.4.0
#> [19] tidytext_0.4.3           glue_1.8.1
#> [21] openalexR_3.1.2          lubridate_1.9.5
#> [23] forcats_1.0.1            stringr_1.6.0
#> [25] dplyr_1.2.1              purrr_1.2.2
#> [27] readr_2.2.0              tidyr_1.3.2
#> [29] tibble_3.3.1             ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
```

```
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4     tokenizers_0.3.0       plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       httr_1.4.9
 ↪  polyclip_1.10-7
#>  [43] compiler_4.6.1        here_1.0.2            bit64_4.8.6
#>  [46] withr_3.0.3            S7_1.0.0
 ↪  viridis_0.6.5
#>  [49] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [52] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [55] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
#>  [58] promises_1.5.0         grid_4.6.1
 ↪  stringdist_0.9.17
#>  [61] reshape2_1.4.5         generics_0.1.4
 ↪  gtable_0.3.6
#>  [64] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [67] data.table_1.18.6.1    hms_1.1.4             xml2_1.6.0
#>  [70] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [73] nsyllable_1.0.1       vroom_1.7.1           later_1.4.8
#>  [76] tweenr_2.0.3          lattice_0.22-9        FNN_1.1.4.1
#>  [79] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [82] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [85] NLP_0.3-3             bookdown_0.48         stats4_4.6.1
#>  [88] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
```

```
#>  [91] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [94] stringi_1.8.9         qpdf_1.4.1            yaml_2.3.12
#>  [97] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#> [100] xtable_1.8-8          systemfonts_1.3.4      Rcpp_1.1.2
#> [103] readxl_1.6.0          triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [106] parallel_4.6.1        assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [109] slam_0.1-56           viridisLite_0.4.3
 ↪  scales_1.4.0
#> [112] crayon_1.5.3         openxlsx_4.2.9        rlang_1.3.0
#> [115] fastmatch_1.1-8
```

Chapter 25

Detecting Emerging Topics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

25.1 Learning objectives

After completing this chapter, you will be able to:

- Define what constitutes an “emerging topic” in scientometric terms
- Apply burst detection to keyword time series
- Track topic prevalence over time using rolling-window analysis
- Compute novelty and transience indicators for documents
- Distinguish genuine emergence from database coverage artefacts

25.2 Setup

```
library(tidyverse)
library(openalexR)
library(quanteda)
library(quanteda.textstats)
library(tidytext)
library(glue)
library(gt)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

25.3 Conceptual background

An **emerging topic** is a research area experiencing rapid growth in attention — new publications, new keywords, increasing citation flows. Detecting emergence early has strategic value for researchers, funders, and policymakers. But distinguishing genuine intellectual emergence from database artefacts (a journal newly indexed, a conference proceedings batch-imported) requires careful methodology.

Burst detection formalises the intuition that a keyword or topic has "burst" when its frequency increases sharply relative to baseline. Kleinberg (2003) introduced a state-based model: a keyword exists in a "normal" state (baseline frequency) or a "burst" state (elevated frequency). The algorithm identifies time intervals during which each keyword is in its burst state. Cascading bursts at multiple time scales reveal both short-lived spikes and sustained surges.

Temporal topic tracking extends topic models (22.3) by fitting models to time-sliced corpora or using dynamic topic models where topic distributions evolve smoothly over time. A simpler approach uses rolling windows: fit a model to each 3-year window and track how topic proportions change between windows.

Novelty and transience indicators operate at the document level. A paper is novel if its content (measured by embedding distance or topic distribution) is distant from prior literature. A paper is transient if it is distant from subsequent literature — it attracted brief attention but did not generate follow-on work. Genuinely emerging topics are characterised by papers that are novel and non-transient: they introduce new ideas that the field continues to develop.

Growth-rate metrics are the simplest emergence indicators: the year-over-year growth rate of publications containing a keyword, the acceleration of citation rates, or the increase in the number of distinct institutions working on a topic. These are intuitive but sensitive to baseline effects (a keyword going from 1 to 3 papers is 200% growth but not meaningful emergence).

25.4 Worked example

25.4.1 Preparing temporal keyword data

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2015-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 600, seed = 42)
)

text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 50) |>
  transmute(
    doc_id = id,
    text = paste(display_name, abstract, sep = ". "),
    year = year(publication_date)
  )

cat(glue("Documents: {nrow(text_df)}\n"))
```

```
#> Documents: 128
```

```
cat(glue("Year range:
 ↪  {min(text_df$year)}--{max(text_df$year)}\n"))
```

```
#> Year range: 2015--2023
```

```
corp <- corpus(text_df, docid_field = "doc_id", text_field =
 ↪  "text")
docvars(corp, "year") <- text_df$year

toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en")) |>
  tokens_remove(c("study", "paper", "results", "research",
   ↪  "analysis",
                  "also", "however", "using", "based"))

dfmat <- dfm(toks) |>
  dfm_trim(min_termfreq = 10, min_docfreq = 5)

kw_by_year <- quanteda::convert(dfm_group(dfmat, groups = year),
 ↪  to = "data.frame") |>
```

```
  pivot_longer(-doc_id, names_to = "term", values_to = "count")
   ↪  |>
  rename(year = doc_id) |>
  mutate(year = as.integer(year))

cat(glue("Terms tracked: {n_distinct(kw_by_year$term)}\n"))
```

```
#> Terms tracked: 306
```

25.4.2 Growth rate detection

```
growth <- kw_by_year |>
  group_by(term) |>
  arrange(year) |>
  mutate(
    growth = (count - lag(count)) / pmax(lag(count), 1),
    total = sum(count)
  ) |>
  ungroup() |>
  filter(!is.na(growth), total >= 20)

recent_growth <- growth |>
  filter(year >= 2021) |>
  group_by(term) |>
  summarise(mean_growth = mean(growth), total = first(total),
            .groups = "drop") |>
  filter(mean_growth > 0) |>
  arrange(desc(mean_growth))

recent_growth |>
  head(15) |>
  gt() |>
  fmt_percent(columns = mean_growth, decimals = 0)
```

term	mean_growth	total
countries	610%	63
universities	434%	31
p	433%	22
three	348%	30
higher	321%	28
index	267%	26
indicators	237%	33
sciences	220%	56
software	212%	21
access	207%	28
university	192%	24

patterns	186%	33
social	180%	39
peer	167%	35
journal	164%	55

25.4.3 Simple burst detection

```
detect_burst <- function(counts, threshold = 2) {
  mean_count <- mean(counts)
  sd_count <- sd(counts)
  if (sd_count == 0) return(rep(FALSE, length(counts)))
  z_scores <- (counts - mean_count) / sd_count
  z_scores > threshold
}

bursts <- kw_by_year |>
  group_by(term) |>
  filter(n() >= 5, sum(count) >= 20) |>
  arrange(year) |>
  mutate(is_burst = detect_burst(count, threshold = 1.5)) |>
  ungroup()

burst_summary <- bursts |>
  filter(is_burst) |>
  group_by(term) |>
  summarise(
    burst_years = paste(year, collapse = ", "),
    max_count = max(count),
    .groups = "drop"
  ) |>
  arrange(desc(max_count))

burst_summary |> head(15) |> gt()
```

term	burst_years	max_count
papers	2020	36
science	2020	32
scientific	2020	31
citations	2020	22
information	2022	22
countries	2022	21
number	2020	20
researchers	2021	20
knowledge	2021	19
review	2018	19
cited	2020	17
publication	2020	17
sciences	2023	17

terms	2020	17
different	2020	16

25.4.4 Visualising keyword trajectories

```
top_burst_terms <- burst_summary |> head(6) |> pull(term)

burst_plot_data <- bursts |>
  filter(term %in% top_burst_terms)

ggplot(burst_plot_data, aes(x = year, y = count)) +
  geom_rect(data = burst_plot_data |> filter(is_burst),
            aes(xmin = year - 0.5, xmax = year + 0.5,
                ymin = -Inf, ymax = Inf),
            fill = "gold", alpha = 0.3, inherit.aes = FALSE) +
  geom_line(linewidth = 0.8) +
  geom_point(size = 1.5) +
  facet_wrap(~ term, scales = "free_y", ncol = 3) +
  labs(x = "Year", y = "Frequency") +
  theme_sci(base_size = 9)
```

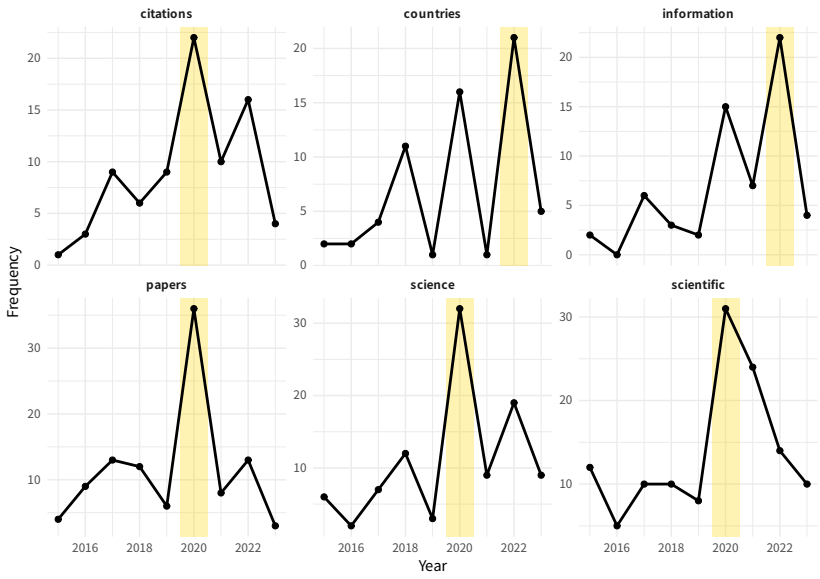


Figure 25.1: Temporal trajectories of emerging keywords with burst periods highlighted.

25.4.5 Rolling-window topic tracking

```
window_size <- 3
years <- sort(unique(text_df$year))
windows <- years[1:(length(years) - window_size + 1)]

window_top_terms <- map_dfr(windows, function(start_year) {
  end_year <- start_year + window_size - 1
  docs <- docvars(dfmat, "year") >= start_year &
          docvars(dfmat, "year") <= end_year
  if (sum(docs) < 10) return(tibble())
  top <- topfeatures(dfmat[docs, ], 10)
  tibble(window = glue("{start_year}-{end_year}"),
         term = names(top), frequency = unname(top))
})
```

```
window_top_terms |>
  group_by(window) |>
  slice_max(frequency, n = 8) |>
  mutate(term = reorder_within(term, frequency, window)) |>
  ggplot(aes(x = frequency, y = term)) +
  geom_col(fill = palette_sci(1)) +
  facet_wrap(~ window, scales = "free_y", ncol = 3) +
  scale_y_reordered() +
  labs(x = "Frequency", y = NULL) +
  theme_sci(base_size = 8)
```

25.5 Diagnostics and interpretation

- **Baseline effects**: A keyword going from 2 to 6 occurrences shows 200% growth but may not represent meaningful emergence. Set minimum frequency thresholds before computing growth rates.
- **Database artefacts**: A newly indexed journal or a batch import of conference proceedings can create spurious “bursts”. Cross-check with independent evidence.
- **Burst duration**: Short bursts (1 year) may reflect a single influential paper or a special issue. Sustained bursts (3+ years) more likely indicate genuine emergence.
- **Window sensitivity**: Rolling-window results change with window width. Report the chosen width and test sensitivity with wider and narrower windows.

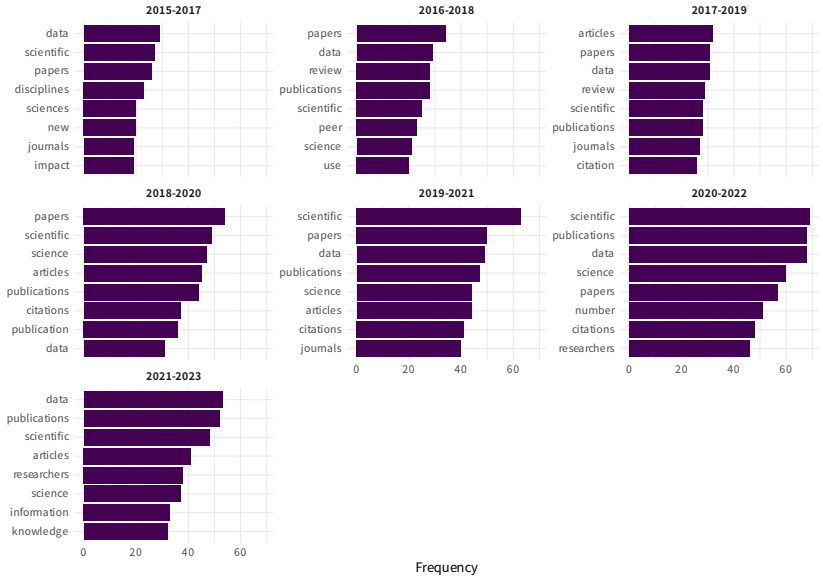


Figure 25.2: Top terms in each 3-year rolling window, showing vocabulary evolution.

25.6 Limitations and responsible use

25.7 Limitations and responsible use

- **Emergence is retrospective.** By the time bibliometric methods detect emergence, the topic has already gained momentum. These methods identify "recently emerged" topics, not future breakthroughs.
- **Coverage bias.** OpenAlex coverage has expanded rapidly, meaning recent years have more records. Growth rates may reflect database growth rather than intellectual emergence (Priem et al., 2022).
- **Language of emergence.** English-language terms dominate; emerging topics in non-English literatures may be invisible until translated or adopted by English-language publications.
- **Not predictive.** Burst detection identifies past patterns. It does not predict which topics will continue growing. Many bursts are transient (Kleinberg, 2003).
- **Strategic gaming.** If emergence metrics are used for funding decisions, researchers may strategically choose "hot" keywords, creating feedback loops (Hicks et al., 2015).

25.8 Common pitfalls

25.9 Common pitfalls

- **Confusing absolute growth with relative emergence.** A field that publishes 10,000 papers/year growing by 5% adds 500 papers. A niche area growing from 10 to 30 papers shows 200% growth. The niche area is "emerging" but the large field is adding more absolute volume.
- **Not controlling for corpus growth.** If the overall number of publications in the corpus grows, all keywords will grow. Use relative frequency (keyword count / total documents) rather than absolute counts.
- **Overreacting to single-year spikes.** One conference proceedings volume or special issue can create a one-year spike. Require multi-year sustained growth before declaring emergence.
- **Ignoring synonyms.** "Deep learning" and "neural networks" may both refer to the same emerging trend. Merge synonyms before analysis.

25.10 Exercises

1. **Relative burst detection.** Modify the burst detection to use relative frequency (keyword proportion) instead of absolute counts. Do the same terms still burst?

2. **Emergence by journal.** Compare the most frequent emerging keywords in two different journals. Do they share emerging topics or focus on different frontiers?

3. **Novelty scoring.** For each document, compute its cosine distance to the centroid of all prior-year documents (using TF-IDF vectors). Plot novelty scores over time. Are recent papers becoming more or less novel?

4. **Validation.** Select 5 terms identified as "emerging" by burst detection. Search for qualitative evidence (review articles, conference themes) confirming their emergence. What is the false positive rate?

25.11 Solutions

Solutions are provided in E.

25.12 Further reading

- Kleinberg (2003) — Bursty patterns in temporal data: the foundational burst detection algorithm.
- de Solla Price (1963) — Early observations on the growth of scientific literature and research fronts.
- Priem et al. (2022) — OpenAlex temporal coverage patterns relevant to emergence detection.
- Callon et al. (1983) — Co-word dynamics; an early approach to tracking topical evolution.

25.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
```

```
#>  [5] topicmodels_0.2-17        quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4          ggraph_2.2.2
#>  [9] tidygraph_1.3.1           igraph_2.3.4
#> [11] quanteda_4.5.1            pdftools_3.9.1
#> [13] arrow_26.0.1.1            bibliometrix_5.5.0
#> [15] RefManageR_1.4.0          bib2df_1.1.2.0
#> [17] rcrossref_1.2.1           gt_1.4.0
#> [19] tidytext_0.4.3            glue_1.8.1
#> [21] openalexR_3.1.2           lubridate_1.9.5
#> [23] forcats_1.0.1             stringr_1.6.0
#> [25] dplyr_1.2.1               purrr_1.2.2
#> [27] readr_2.2.0               tidyr_1.3.2
#> [29] tibble_3.3.1              ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4     tokenizers_0.3.0      plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       httr_1.4.9
 ↪  polyclip_1.10-7
#>  [43] compiler_4.6.1        here_1.0.2            bit64_4.8.6
#>  [46] withr_3.0.3            S7_1.0.0
 ↪  viridis_0.6.5
#>  [49] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [52] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [55] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
```

```
#>  [58] promises_1.5.0         grid_4.6.1
 ↪  stringdist_0.9.17
#>  [61] reshape2_1.4.5         generics_0.1.4
 ↪  gtable_0.3.6
#>  [64] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [67] data.table_1.18.6.1    hms_1.1.4              xml2_1.6.0
#>  [70] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [73] nsyllable_1.0.1       vroom_1.7.1           later_1.4.8
#>  [76] tweenr_2.0.3          lattice_0.22-9        FNN_1.1.4.1
#>  [79] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [82] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [85] NLP_0.3-3             bookdown_0.48         stats4_4.6.1
#>  [88] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [91] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [94] stringi_1.8.9         qpdf_1.4.1            yaml_2.3.12
#>  [97] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#> [100] xtable_1.8-8           systemfonts_1.3.4     Rcpp_1.1.2
#> [103] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [106] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [109] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [112] crayon_1.5.3          openxlsx_4.2.9         rlang_1.3.0
#> [115] fastmatch_1.1-8
```

Part VI

Advanced Topics

Chapter 26

Altmetrics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

26.1 Learning objectives

After completing this chapter, you will be able to:

- Define altmetrics and explain how they complement citation-based indicators
- Retrieve altmetric-related data from OpenAlex (open access status, type, counts)
- Compute correlations between citation counts and alternative indicators
- Interpret altmetric signals critically — distinguishing attention from impact
- Discuss the limitations of altmetrics for research evaluation

26.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

26.3 Conceptual background

Citations measure one dimension of scholarly impact: influence on subsequent research. But scholarly work can also influence policy, education, clinical practice, and public understanding — none of which are captured by citation counts. **Altmetrics** (alternative metrics) attempt to fill this gap by tracking online traces of attention: social media mentions, news coverage, blog posts, Wikipedia citations, Mendeley reader counts, and policy document references.

The term was coined by Priem et al. (2010), who argued that the speed and diversity of online scholarly communication demanded new metrics beyond the Journal Impact Factor. Altmetrics are typically faster than citations (social media attention peaks within days of publication, while citations take years) and broader (they capture engagement by non-researchers).

However, altmetrics come with significant caveats. Social media attention correlates weakly with scientific quality. Metrics are easily gamed (bots, self-promotion). Coverage is uneven: biomedical research attracts far more Twitter/X attention than mathematics or engineering. And the platforms providing altmetric data (Altmetric.com, PlumX) are commercial products with proprietary algorithms (Hicks et al., 2015).

OpenAlex provides open-access status and citation counts but does not directly provide social media metrics. For altmetric data, external APIs (Altmetric.com, Crossref Event Data) are needed. In this chapter, we demonstrate what can be done with OpenAlex metadata alone and discuss how to extend the analysis with external sources.

26.4 Worked example

26.4.1 Fetching data with OA indicators

```
works <- fetch_openalex(
  entity = "works",
```

```
  primary_location.source.id = "S148561398",
  from_publication_date = "2020-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 400, seed = 42)
)

works_slim <- works |>
  transmute(
    id, display_name,
    year = year(publication_date),
    cited_by_count,
    oa_status = oa_status,
    type,
    referenced_works_count
  )

cat(glue("Works retrieved: {nrow(works_slim)}\n"))
```

```
#> Works retrieved: 400
```

26.4.2 Open access as an altmetric dimension

```
oa_summary <- works_slim |>
  count(oa_status, sort = TRUE) |>
  mutate(pct = scales::percent(n / sum(n)))

oa_summary |> gt()
```

oa_status	n	pct
closed	196	49.0%
hybrid	117	29.2%
green	58	14.5%
bronze	29	7.2%

```
works_slim |>
  filter(!is.na(oa_status)) |>
  ggplot(aes(x = oa_status, y = cited_by_count + 1)) +
  geom_boxplot(fill = palette_sci(1), alpha = 0.7) +
  scale_y_log10() +
  labs(x = "OA status", y = "Citations (log scale)") +
  theme_sci()
```

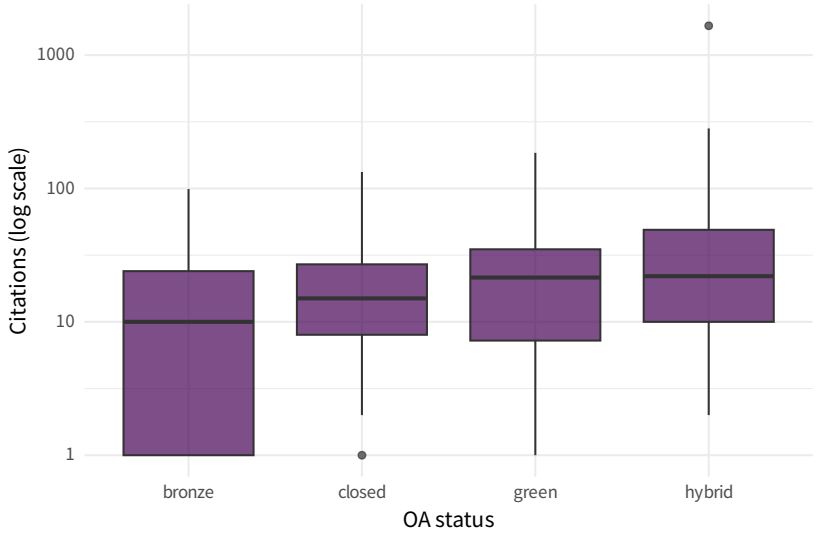


Figure 26.1: Citation distributions by open access status.

26.4.3 Citation count distribution

```
ggplot(works_slim, aes(x = cited_by_count)) +
  geom_histogram(binwidth = 5, fill = palette_sci(1), colour =
  ↪  "white") +
  labs(x = "Citation count", y = "Number of papers") +
  theme_sci()
```

26.4.4 Proxy altmetric: reference list length

Reference list length correlates with engagement breadth and can serve as a simple proxy for scholarly thoroughness.

```
ggplot(works_slim, aes(x = referenced_works_count, y =
↪  cited_by_count)) +
  geom_point(alpha = 0.3, colour = palette_sci(1)) +
  geom_smooth(method = "lm", se = FALSE, colour =
  ↪  palette_sci(2)[2]) +
  labs(x = "Number of references", y = "Citation count") +
  theme_sci()
```

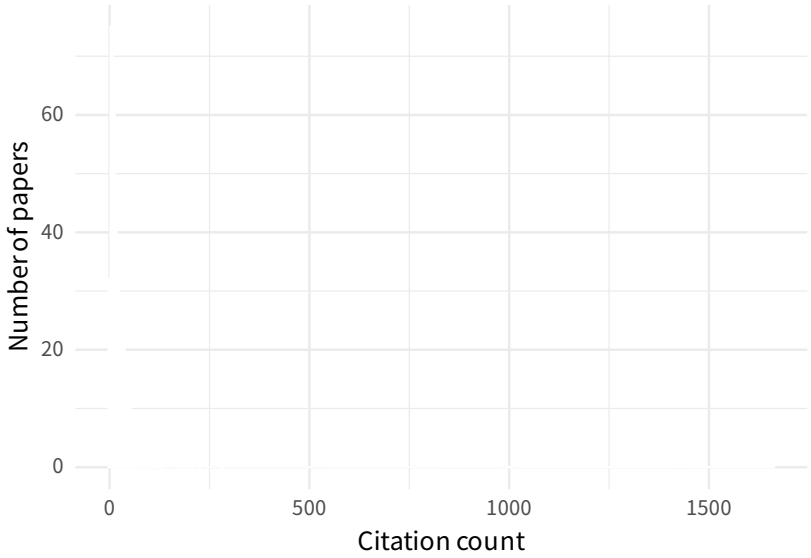


Figure 26.2: Citation distribution of the sample, showing the characteristic right skew.

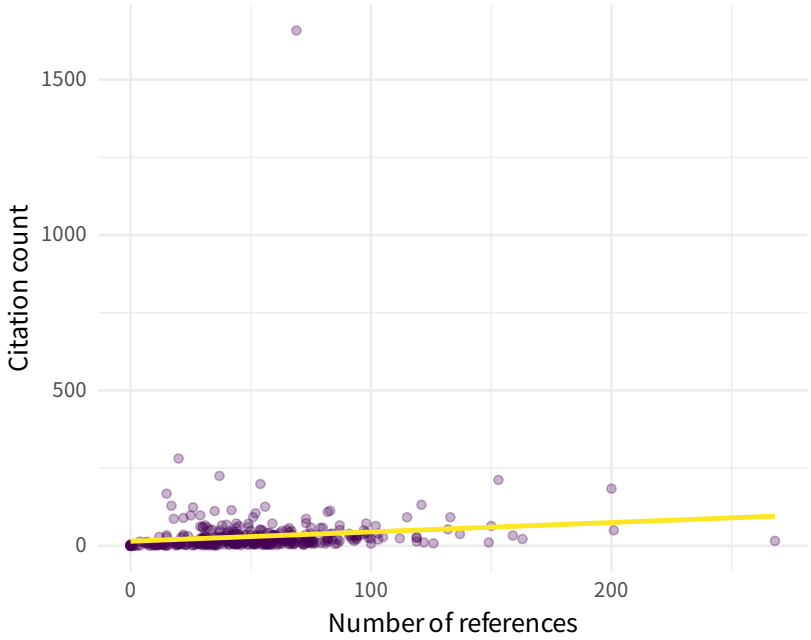


Figure 26.3: Relationship between reference list length and citation count.

```
cor_val <- cor(works_slim$referenced_works_count,
 ↪  works_slim$cited_by_count,
               use = "complete.obs")
cat(glue("Pearson correlation (references vs. citations):
 ↪  {round(cor_val, 3)}\n"))
```

```
#> Pearson correlation (references vs. citations): 0.113
```

26.5 Diagnostics and interpretation

- **OA advantage**: Open-access articles often receive more citations, but this may reflect self-selection (higher-quality papers are more likely to be made OA) rather than the causal effect of open access.
- **Platform coverage**: Social media altmetrics are biased toward English-language, biomedical, and controversial topics. Low altmetric scores do not indicate low impact.
- **Temporal dynamics**: Altmetric attention peaks immediately after publication and decays rapidly. Citations follow a slower, more sustained trajectory.
- **Gaming risk**: Social media metrics are more easily manipulated than citations. Interpret high altmetric scores with caution.

26.6 Limitations and responsible use

26.7 Limitations and responsible use

- **Attention ≠ impact.** A viral tweet about a paper does not mean the paper is scientifically significant. Altmetrics measure visibility, not quality (Hicks et al., 2015).
- **Disciplinary bias.** Fields with active social media communities (public health, climate science) generate far more altmetric data than fields that do not (pure mathematics, medieval history).
- **Platform dependence.** Altmetric data depends on commercial platforms whose coverage, algorithms, and existence are not guaranteed.
- **Never use altmetrics for individual evaluation.** The Leiden Manifesto and DORA both caution against using any single metric — including altmetrics — as a proxy for research quality (American Society for Cell Biology, 2012).

26.8 Common pitfalls

26.9 Common pitfalls

- **Treating Mendeley readers as "citations."** Mendeley reader counts reflect who saved a paper, not who cited it. The overlap is partial.
- **Comparing altmetrics across fields.** A paper in public health with 50 tweets is unremarkable; the same count in algebraic topology is extraordinary.
- **Ignoring bots and self-promotion.** Some social media mentions are from automated accounts or coordinated campaigns. Commercial altmetric services attempt to filter these, but coverage is imperfect.
- **Confusing the Altmetric Attention Score with a quality indicator.** The score is a weighted sum of online mentions, not a peer-reviewed quality assessment.

26.10 Exercises

1. **OA citation advantage.** For two journals with different OA rates, compare mean citations for OA vs. closed-access papers. Is the "OA advantage" consistent across journals?

2. **Temporal attention.** For a set of recent papers, track how citation counts change between 6 months and 2 years after publication. Do papers with high early attention (many references, rapid initial citations) sustain their advantage?

3. **Cross-field comparison.** Fetch samples from a biomedical journal and a mathematics journal. Compare their citation distributions and reference list lengths. What does this suggest about field-specific attention patterns?

26.11 Solutions

Solutions are provided in E.

26.12 Further reading

- Priem et al. (2010) — The altmetrics manifesto: the original call for alternative impact measures.
- Hicks et al. (2015) — The Leiden Manifesto: responsible metrics, including cautions about altmetrics.
- American Society for Cell Biology (2012) — DORA: principles for research assessment beyond the JIF.
- Priem et al. (2022) — OpenAlex open-access classification and metadata.

26.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
```

```
#>  [5] topicmodels_0.2-17        quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4          ggraph_2.2.2
#>  [9] tidygraph_1.3.1           igraph_2.3.4
#> [11] quanteda_4.5.1            pdftools_3.9.1
#> [13] arrow_26.0.1.1            bibliometrix_5.5.0
#> [15] RefManageR_1.4.0          bib2df_1.1.2.0
#> [17] rcrossref_1.2.1           gt_1.4.0
#> [19] tidytext_0.4.3            glue_1.8.1
#> [21] openalexR_3.1.2           lubridate_1.9.5
#> [23] forcats_1.0.1             stringr_1.6.0
#> [25] dplyr_1.2.1               purrr_1.2.2
#> [27] readr_2.2.0               tidyr_1.3.2
#> [29] tibble_3.3.1              ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4     tokenizers_0.3.0      plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3             S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0          MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
```

```
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3          lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3             bookdown_0.48          stats4_4.6.1
#>  [91] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0          cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3           openxlsx_4.2.9         rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 27

Open Science Indicators

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

27.1 Learning objectives

After completing this chapter, you will be able to:

- Classify open access types (gold, green, hybrid, bronze) using OpenAlex metadata
- Compute OA rates by year, journal, and discipline
- Track the growth of preprint posting and preprint-to-publication pathways
- Assess data availability statement coverage in a corpus
- Discuss the relationship between open science practices and citation impact

27.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

27.3 Conceptual background

The open science movement aims to make research processes and outputs freely accessible. Bibliometric analysis can measure the extent to which these ideals are being realised, track trends over time, and identify disparities across disciplines and geographies.

Open access (OA) is the most measurable dimension of open science. OpenAlex classifies each work's OA status using Unpaywall data: **gold** (published in a fully OA journal), **green** (available in a repository), **hybrid** (OA in a subscription journal via APC), **bronze** (free to read but without an explicit open license), or **closed** (Priem et al., 2022).

Preprints are manuscripts posted publicly before peer review. The growth of preprint servers (bioRxiv, arXiv, medRxiv) has accelerated since 2015, with a dramatic spike during the COVID-19 pandemic. Bibliometric analysis can track preprint posting rates, the time from preprint to journal publication, and whether preprints receive more citations than non-preprint papers.

Data availability statements (DAS) indicate whether the data underlying a study are accessible. Their prevalence is growing as journals adopt mandatory data-sharing policies, but the quality of DAS varies: many state "data available upon request" without providing actual access.

Measuring open science is important for policy. Funders increasingly require OA publication and data sharing; institutions report OA rates as indicators of compliance and impact. However, Hicks et al. (2015) caution that metrics-driven OA mandates can produce perverse incentives (predatory journals, superficial compliance).

27.4 Worked example

27.4.1 OA rates by year

```
works <- fetch_openalex(
  entity = "works",
```

```
  primary_location.source.id = "S148561398",
  from_publication_date = "2015-01-01",
  to_publication_date = "2023-12-31",
  type = "article",
  options = list(sample = 600, seed = 42)
)

works_oa <- works |>
  transmute(
    id, year = year(publication_date),
    oa_status = oa_status,
    cited_by_count
  )

cat(glue("Works: {nrow(works_oa)}\n"))
```

```
#> Works: 600
```

```
oa_by_year <- works_oa |>
  count(year, oa_status) |>
  group_by(year) |>
  mutate(pct = n / sum(n)) |>
  ungroup()

ggplot(oa_by_year, aes(x = factor(year), y = pct, fill =
 ↪  oa_status)) +
  geom_col() +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values =
   ↪  palette_sci(n_distinct(oa_by_year$oa_status))) +
  labs(x = "Publication year", y = "Proportion", fill = "OA
   ↪  status") +
  theme_sci()
```

27.4.2 OA citation advantage

```
oa_cites <- works_oa |>
  filter(!is.na(oa_status)) |>
  mutate(is_oa = oa_status != "closed") |>
  group_by(is_oa) |>
  summarise(
    n = n(),
    mean_cites = round(mean(cited_by_count), 1),
    median_cites = median(cited_by_count),
```

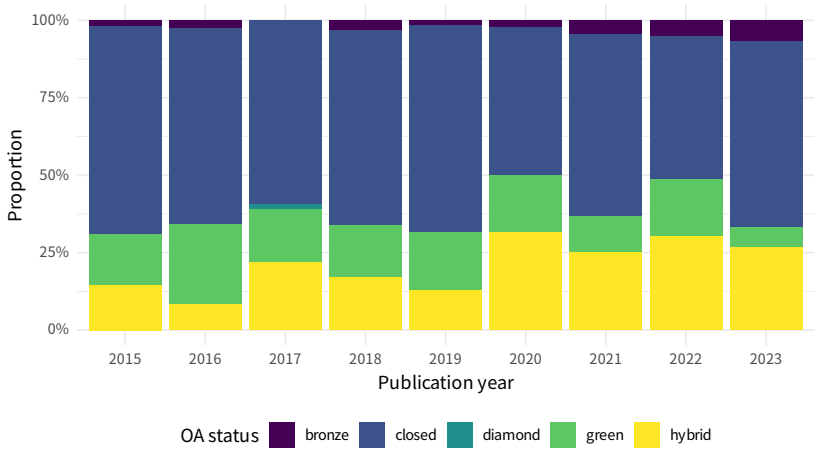


Figure 27.1: Open access rates by year and OA type.

```
    .groups = "drop"
  )

oa_cites |> gt()
```

is_oa	n	mean_cites	median_cites
FALSE	350	29.5	20.5
TRUE	250	65.3	25.0

27.4.3 Document type breakdown

```
works |>
  count(type, oa_status) |>
  group_by(type) |>
  mutate(pct = n / sum(n)) |>
  ungroup() |>
  ggplot(aes(x = pct, y = type, fill = oa_status)) +
  geom_col() +
  scale_x_continuous(labels = scales::percent) +
  scale_fill_manual(values =
    ↪  palette_sci(n_distinct(works$oa_status))) +
  labs(x = "Proportion", y = NULL, fill = "OA status") +
  theme_sci()
```

27.4.4 Summary table


```
works_oa |>
  count(oa_status, sort = TRUE) |>
  mutate(pct = scales::percent(n / sum(n))) |>
  gt() |>
  cols_label(oa_status = "OA Status", n = "Count", pct =
    ↪  "Percentage")
```

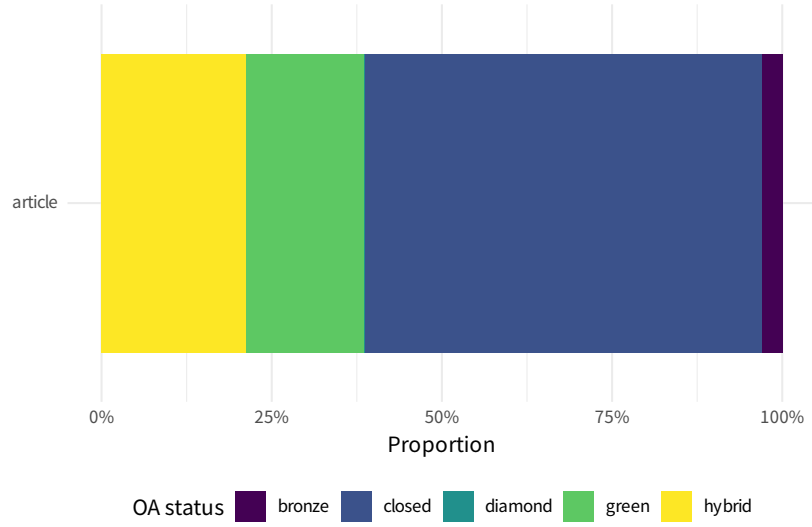


Figure 27.2: OA status by document type.

27.5 Diagnostics and interpretation

- **Classification accuracy**: OpenAlex OA classification relies on Unpaywall, which may lag behind publisher changes. Spot-check a sample of "closed" articles to verify they are truly behind a paywall.
- **Bronze ambiguity**: Bronze OA means free-to-read without an explicit license. Publishers can revoke access at any time. Do not treat bronze as equivalent to gold or green.
- **Year effects**: OA rates have risen over time due to funder mandates and policy changes. Compare year-normalised rates rather than pooling across years.
- **Self-selection**: Higher OA rates among highly cited papers may reflect self-selection (better papers are more likely to be made OA) rather than a causal OA effect.

27.6 Limitations and responsible use

27.7 Limitations and responsible use

- **OA ≠ quality.** Open access status is a property of the publication venue and access model, not the research quality. Predatory journals

are often gold OA.
- **Coverage gaps.** Green OA via institutional repositories is underreported in some databases. Actual OA rates may be higher than measured.
- **Data sharing claims vs. reality.** A data availability statement saying "data available upon request" often results in no response when requests are made. Measuring DAS presence overestimates actual data sharing.
- **Equity implications.** Gold OA via article processing charges shifts costs from readers to authors, potentially disadvantaging researchers without institutional funding (Hicks et al., 2015).

27.8 Common pitfalls

27.9 Common pitfalls

- **Treating all OA as equivalent.** Gold, green, hybrid, and bronze OA have very different implications for access, sustainability, and cost.
- **Ignoring embargo periods.** Green OA articles may have 6–24 month embargoes before they become freely available. They are "closed" during the embargo.
- **Double-counting.** An article available in both a gold OA journal and a green repository should be counted once. OpenAlex classifies by the "best" OA type.
- **Using OA rate as a performance indicator.** OA rates reflect policy and funding environment, not research quality. Mandated OA inflates rates without changing the underlying work.

27.10 Exercises

1. **Cross-journal OA comparison.** Compare OA rates for three journals in different disciplines. Which discipline has the highest gold OA rate?

2. **Temporal OA growth.** Compute the year-over-year growth rate of gold OA articles in your corpus. Has growth accelerated since 2020?

3. **OA and collaboration.** Is there a relationship between the number of authors on a paper and its OA status? Hypothesise why.

27.11 Solutions

Solutions are provided in E.

27.12 Further reading

- Priem et al. (2022) — OpenAlex OA classification methodology.
- Hicks et al. (2015) — The Leiden Manifesto: principles for responsible metrics, including OA considerations.
- American Society for Cell Biology (2012) — DORA: recommendations for research assessment beyond journal metrics.

27.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
```

```
#>
#> other attached packages:
#>  [1] uwot_0.2.5                Matrix_1.7-5
#>  [3] word2vec_0.4.1            stm_1.3.10
#>  [5] topicmodels_0.2-17        quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4          ggraph_2.2.2
#>  [9] tidygraph_1.3.1           igraph_2.3.4
#> [11] quanteda_4.5.1            pdftools_3.9.1
#> [13] arrow_26.0.1.1            bibliometrix_5.5.0
#> [15] RefManageR_1.4.0          bib2df_1.1.2.0
#> [17] rcrossref_1.2.1           gt_1.4.0
#> [19] tidytext_0.4.3            glue_1.8.1
#> [21] openalexR_3.1.2           lubridate_1.9.5
#> [23] forcats_1.0.1             stringr_1.6.0
#> [25] dplyr_1.2.1               purrr_1.2.2
#> [27] readr_2.2.0               tidyr_1.3.2
#> [29] tibble_3.3.1              ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4      tokenizers_0.3.0       plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3             S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0          MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
```

```
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3           lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3              bookdown_0.48          stats4_4.6.1
#>  [91] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3           openxlsx_4.2.9         rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 28

Gender and Equity Analysis

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

28.1 Learning objectives

After completing this chapter, you will be able to:

- Distinguish a name-lookup category from a person's gender identity
- Measure lookup coverage and explain how excluding unmatched records changes a denominator
- Summarise explicitly synthetic authorship records by recorded position
- Identify the data, validation and consent questions required for an actual equity study

28.2 Setup

```
library(tidyverse)
library(glue)
library(gt)

source(here::here("R", "sci_palette.R"))
```

28.3 Conceptual background

Research such as Larivière et al. (2013) examines gender disparities in scientific authorship. Its findings concern particular data, operational definitions and study periods; they are not measurements supplied by this chapter.

Names are not observations of gender identity. A name-classification service produces a model or lookup output, and its coverage and errors depend on its reference data and target population. Santamaría and Mihaljevic (2018) compare such services and examine their differing performance. A small hand-written dictionary has no demonstrated accuracy for a scholarly population. Increasing its coverage would not establish validity.

Authorship order is also context-dependent. A recorded first or last position does not establish contribution or seniority, and alphabetical conventions differ from contribution-based conventions. Gender-equity analysis therefore requires an explicit population, defensible demographic measurement, appropriate handling of unavailable information, and interpretation of the authorship conventions actually used. Responsible research assessment should not reduce individuals to such categories (Hicks et al., 2015; American Society for Cell Biology, 2012).

28.4 Worked example

28.4.1 Explicitly synthetic records

The following records are invented for a data-handling exercise. They do not describe real authors, papers, journals or gender identities. The lookup categories **A** and **B** are arbitrary labels with no demographic meaning. No OpenAlex records or external classification service are used in this example.

```
synthetic_authors <- tibble(
  work_id = c(rep("example-1", 3), rep("example-2", 2),
  ↪  "example-3",
              rep("example-4", 4), rep("example-5", 2)),
  author_order = c(1:3, 1:2, 1, 1:4, 1:2),
  first_name = c("aria", "jordan", "lee", "noor", "ari",
  ↪  "morgan",
                 "aria", "nuri", "sam", "alex", "ari", "noor")
)

cat(glue("Synthetic author records:
 ↪  {nrow(synthetic_authors)}\n"))
```

```
#> Synthetic author records: 12
```

```
cat(glue("Distinct example names:
 ↪  {n_distinct(synthetic_authors$first_name)}\n"))
```

```
#> Distinct example names: 9
```

28.4.2 Lookup coverage, not gender classification

```
example_lookup <- tibble(
  first_name = c("aria", "ari", "jordan", "morgan"),
  lookup_category = c("A", "A", "B", "B")
)

lookup_records <- synthetic_authors |>
  left_join(example_lookup, by = "first_name", relationship =
   ↪  "many-to-one") |>
  mutate(lookup_category = replace_na(lookup_category,
   ↪  "Unmatched"))

coverage <- mean(lookup_records$lookup_category != "Unmatched")
cat(glue("Lookup coverage in this invented fixture:
 ↪  {scales::percent(coverage)}\n"))
```

```
#> Lookup coverage in this invented fixture: 50%
```

28.4.3 Keeping the full denominator

```
lookup_summary <- lookup_records |>
  count(lookup_category) |>
  mutate(pct = n / sum(n))

ggplot(lookup_summary, aes(x = lookup_category, y = pct, fill =
 ↪  lookup_category)) +
  geom_col(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = palette_sci(3)) +
  labs(x = "Arbitrary lookup category", y = "Share of all
   ↪  synthetic records") +
  theme_sci()
```

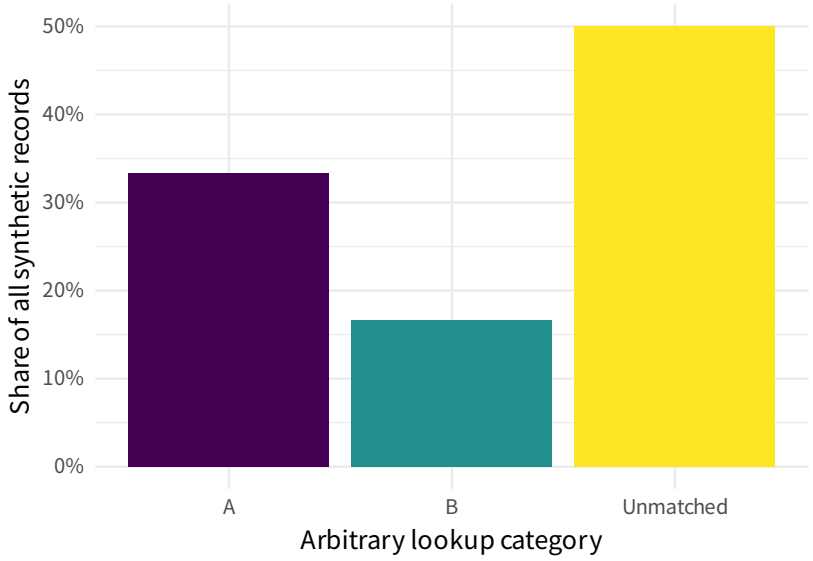


Figure 28.1: Arbitrary lookup categories in invented records, retaining unmatched names in the denominator. These are not gender proportions.

28.4.4 Recorded authorship positions

The invented order is recorded explicitly. Single-author records are kept separate from first, middle and last positions in multi-author records. These positions are not inferred contribution roles.

```
position_data <- lookup_records |>
  group_by(work_id) |>
  arrange(author_order, .by_group = TRUE) |>
  mutate(
    n_authors = n(),
    author_position = case_when(
      n_authors == 1 ~ "single",
      row_number() == 1 ~ "first",
      row_number() == n_authors ~ "last",
      TRUE ~ "middle"
    )
  ) |>
  ungroup()

position_summary <- position_data |>
  count(author_position, lookup_category) |>
  group_by(author_position) |>
  mutate(pct = n / sum(n)) |>
  ungroup()
```

```
position_summary |> gt()
```

author_position	lookup_category	n	pct
first	A	3	0.750
first	Unmatched	1	0.250
last	A	1	0.250
last	Unmatched	3	0.750
middle	B	1	0.333
middle	Unmatched	2	0.667
single	B	1	1.000

```
ggplot(position_summary, aes(x = author_position, y = pct, fill
↪  = lookup_category)) +
  geom_col(position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = palette_sci(3)) +
  labs(x = "Recorded position", y = "Share of synthetic
  ↪  records", fill = "Lookup category") +
  theme_sci()
```

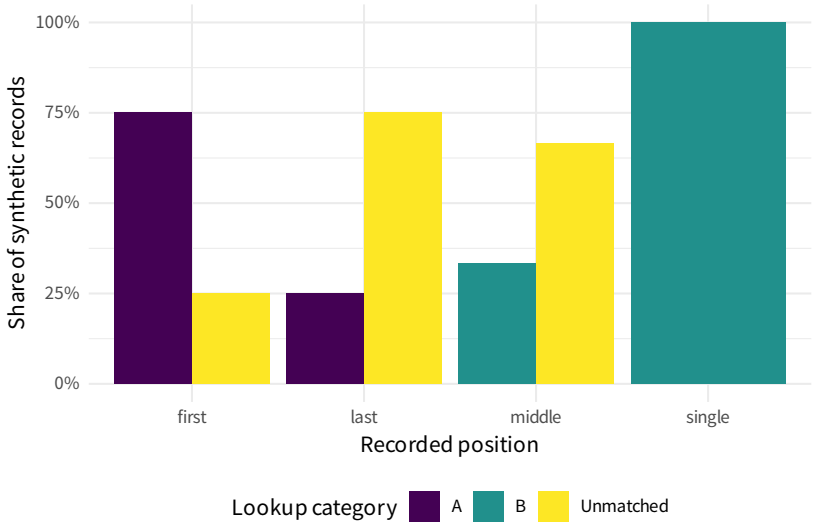


Figure 28.2: Arbitrary lookup categories by recorded position in invented authorship lists. Unmatched records remain in each position's denominator.

28.5 Diagnostics and interpretation

- **Coverage and validity are different.** Coverage is the proportion assigned a dictionary label. This exercise has no observed gender variable against which accuracy could be estimated.
- **Denominators matter.** Report unmatched records rather than silently excluding them. In this fixture, half the records are unmatched; the plotted A/B proportions would change if they were discarded.
- **Units matter.** These are authorship records, not distinct people. Repeated invented names do not establish repeated identity.
- **Position is descriptive.** Check ordering, single-author records and local authorship conventions before interpreting positions.

28.6 Limitations and responsible use

28.7 Limitations and responsible use

This synthetic example establishes no gender distribution, citation gap or equity finding. A real study needs a defensible demographic variable, clearly stated categories and missingness, appropriate consent and data governance, and validation for the population under study. Self-described gender data and the research question may require categories beyond a binary model; a name dictionary cannot supply those observations. Avoid publishing individual demographic assignments inferred from names, and distinguish observed disparity from an explanation of its causes.

28.8 Common pitfalls

28.9 Common pitfalls

- Treating lookup output as measured gender identity or biological sex.
- Calling high classification coverage evidence of accuracy.
- Removing unmatched names and presenting the remainder as representative.
- Counting authorship records as unique researchers without identity linkage.
- Inferring contribution or seniority from author order without supporting information.

28.10 Exercises

1. **Denominator comparison.** Recompute the A/B proportions after excluding unmatched records. Explain why those proportions differ from the chart without establishing demographic representation.
2. **Position audit.** Check that the fixture has four first, four last, three middle and one single-author record. Verify that each position's proportions sum to one and still include unmatched records.
3. **Study design.** Specify the population, demographic measurement, consent/missingness plan and authorship conventions needed for a real equity study. Explain which required variables are absent from this fixture.

28.11 Solutions

Solutions are provided in E.

28.12 Further reading

- Larivière et al. (2013) — An empirical study of gender disparities in scientific authorship, with its own population and operational definitions.
- Santamaría and Mihaljevic´ (2018) — Comparison and benchmark of name-to-gender inference services.
- Hicks et al. (2015) and American Society for Cell Biology (2012) — Responsible interpretation of research metrics.

28.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
```

```
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5                Matrix_1.7-5
#>  [3] word2vec_0.4.1            stm_1.3.10
#>  [5] topicmodels_0.2-17        quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4          ggraph_2.2.2
#>  [9] tidygraph_1.3.1           igraph_2.3.4
#> [11] quanteda_4.5.1            pdftools_3.9.1
#> [13] arrow_26.0.1.1            bibliometrix_5.5.0
#> [15] RefManageR_1.4.0          bib2df_1.1.2.0
#> [17] rcrossref_1.2.1           gt_1.4.0
#> [19] tidytext_0.4.3            glue_1.8.1
#> [21] openalexR_3.1.2           lubridate_1.9.5
#> [23] forcats_1.0.1             stringr_1.6.0
#> [25] dplyr_1.2.1               purrr_1.2.2
#> [27] readr_2.2.0               tidyr_1.3.2
#> [29] tibble_3.3.1              ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
```

```
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4      tokenizers_0.3.0       plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3             S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0          MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3          lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3             bookdown_0.48          stats4_4.6.1
#>  [91] crul_1.6.0              xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0          cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
```

```
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3          openxlsx_4.2.9        rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 29

Retraction Analysis

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

29.1 Learning objectives

After completing this chapter, you will be able to:

- Identify retracted works in OpenAlex using type and metadata filters
- Analyse retraction rates by year, journal, and field
- Examine citation patterns before and after retraction
- Discuss the limitations of bibliometric approaches to research integrity
- Explain why retracted papers continue to accumulate citations

29.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

29.3 Conceptual background

Retraction is the formal mechanism by which the scholarly record corrects itself. A paper is retracted when it is found to contain fabricated data, serious errors, plagiarism, or ethical violations. In theory, retraction should halt the influence of flawed work. In practice, retracted papers continue to be cited — sometimes for years or decades after retraction.

The growth in retractions has been dramatic: from a few dozen per year in the 1990s to over 10,000 in recent years. This increase reflects both more misconduct detection (plagiarism software, image forensics) and more publications overall. Retraction rates vary by field: biomedical sciences have the highest absolute numbers, partly because of larger publication volumes and greater scrutiny.

Citation persistence is the most studied bibliometric aspect of retractions. Studies consistently find that retracted papers continue to accumulate positive citations (i.e., citations that treat the retracted work as valid) long after retraction. This occurs because: (1) citing authors may not be aware of the retraction; (2) database retraction notices are inconsistently displayed; (3) citations in already-published papers are not retroactively removed.

OpenAlex marks retracted works with a specific type or retraction flag, though coverage depends on publisher reporting. The Retraction Watch database provides the most comprehensive curated list of retractions but is not directly integrated into most bibliometric databases.

Bibliometric analysis of retractions must be approached carefully. The goal is to improve the self-correction mechanism, not to stigmatise individuals or institutions.

29.4 Worked example

29.4.1 Finding retracted works

```
retracted <- fetch_openalex(
  entity = "works",
  is_retracted = "true",
  from_publication_date = "2010-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 200, seed = 42)
)

cat(glue("Retracted works retrieved: {nrow(retracted)}\n"))
```

```
#> Retracted works retrieved: 200
```

```
retraction_years <- retracted |>
  mutate(year = year(publication_date)) |>
  count(year, sort = FALSE)

retraction_years |> gt()
```

year	n
2010	9
2011	13
2012	3
2013	7
2014	12
2015	6
2016	6
2017	10
2018	10
2019	12
2020	19
2021	25
2022	41
2023	27

29.4.2 Citation counts of retracted papers

```
ggplot(retracted, aes(x = cited_by_count)) +
  geom_histogram(binwidth = 5, fill = palette_sci(1), colour =
  ↪  "white") +
  labs(x = "Citations received", y = "Number of retracted
  ↪  papers") +
  theme_sci()
```

```
cite_stats <- retracted |>
  summarise(
    n = n(),
    mean_cites = round(mean(cited_by_count), 1),
    median_cites = median(cited_by_count),
    max_cites = max(cited_by_count),
    pct_uncited = scales::percent(mean(cited_by_count == 0))
  )

cite_stats |> gt()
```

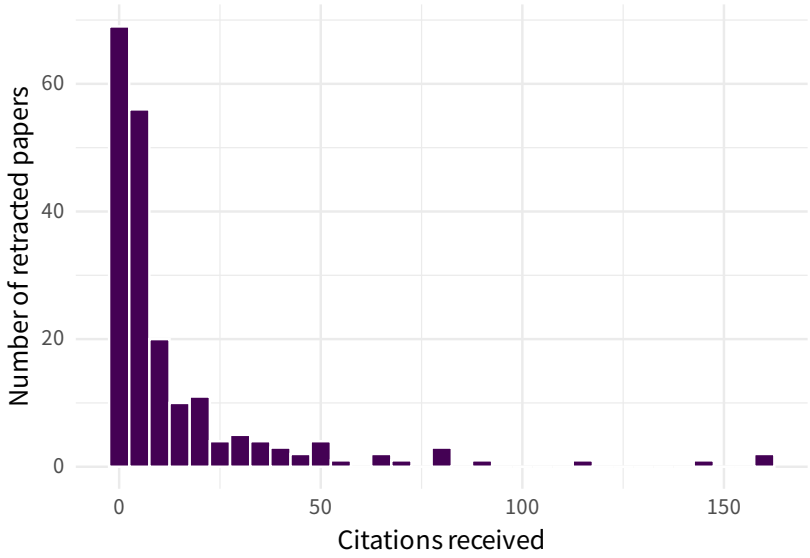
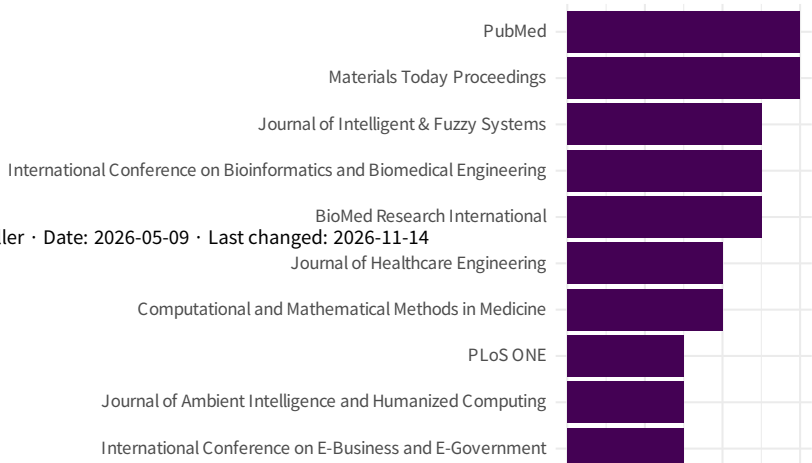


Figure 29.1: Citation count distribution of retracted papers.

n	mean_cites	median_cites	max_cites	pct_uncited
200	14.2	5	161	20%

29.4.3 Retracted papers by source

```
retracted |>
  filter(!is.na(source_display_name)) |>
  count(source_display_name, sort = TRUE) |>
  head(10) |>
  mutate(source_display_name = fct_reorder(source_display_name,
   ↪  n)) |>
  ggplot(aes(x = n, y = source_display_name)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Retracted papers", y = NULL) +
  theme_sci()
```



```
retracted |>
  mutate(age = 2024 - year(publication_date)) |>
  filter(age >= 1, age <= 12) |>
  group_by(age) |>
  summarise(mean_cites = mean(cited_by_count), n = n(), .groups
   ↪  = "drop") |>
  ggplot(aes(x = age, y = mean_cites)) +
  geom_line(linewidth = 1, colour = palette_sci(1)) +
  geom_point(size = 2, colour = palette_sci(1)) +
  labs(x = "Years since publication", y = "Mean citations") +
  theme_sci()
```

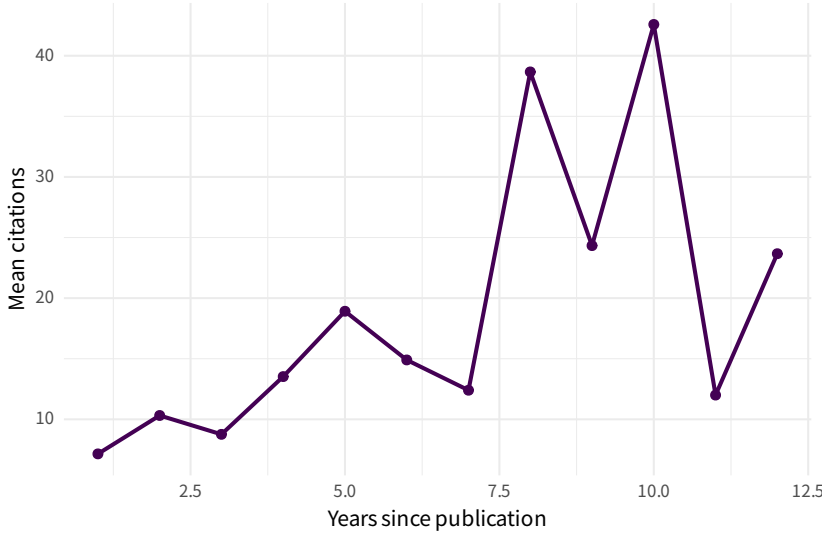


Figure 29.3: Mean citations by publication age for retracted papers.

29.5 Diagnostics and interpretation

- **Coverage**: OpenAlex retraction coverage depends on publisher metadata. Not all retractions are flagged. Cross-check with Retraction Watch for comprehensive coverage.
- **Retraction vs. publication date**: The publication date is when the paper was published, not when it was retracted. The gap between these dates varies from months to decades.
- **Positive vs. negative citations**: Citation counts do not distinguish between "this paper found X" (positive, problematic) and "this paper

was retracted for Y" (negative, appropriate). Manual checking of citing contexts is needed.
- **Denominator**: Retraction rates require the total number of publications as denominator. Raw retraction counts favour large journals.

29.6 Limitations and responsible use

29.7 Limitations and responsible use

- **Retractions are not all equal.** Some retractions reflect honest error, others fraud. The reason for retraction matters for interpretation but is often not coded in bibliometric databases.
- **Stigmatisation risk.** Listing authors or institutions with high retraction counts can be stigmatising. Report aggregate patterns, not individual-level data, unless specifically studying misconduct.
- **Database lag.** Retraction notices may take months or years to propagate through databases. Recent retraction data is incomplete.
- **Self-correction is broader than retraction.** Corrections, expressions of concern, and errata also contribute to self-correction but are harder to track bibliometrically.
- **Do not equate retractions with research quality.** Fields with more scrutiny detect more problems. A high retraction rate may indicate effective oversight, not worse science (Hicks et al., 2015).

29.8 Common pitfalls

29.9 Common pitfalls

- **Counting retractions without denominators.** A journal with 10 retractions from 50,000 papers has a lower retraction rate than one with 5 from 1,000.
- **Assuming all post-retraction citations are problematic.** Some citing papers may specifically discuss the retraction or cite the paper for methodology that remains valid despite data issues.
- **Ignoring retraction reasons.** Honest errors should be distinguished from misconduct in any analysis of research integrity.
- **Using retraction analysis for individual evaluation.** Authors of retracted papers may be victims of co-author misconduct. Never use retraction counts as performance indicators.

29.10 Exercises

1. **Retraction rate by field.** Fetch retracted works from two different fields. Compare the retraction rate (retractions per 1,000 publications). Account for corpus size.
2. **Citation trajectory.** For a retracted paper with many citations, examine when the citations accumulated. Were most citations received before or after the retraction?
3. **Retraction-citation correlation.** Is there a relationship between a paper's citation count at the time of retraction and the number of post-retraction citations it receives?

29.11 Solutions

Solutions are provided in E.

29.12 Further reading

- Fang et al. (2012) — Large-scale analysis of retraction patterns and their causes.
- Hicks et al. (2015) — The Leiden Manifesto: responsible use of metrics in the context of research integrity.
- Garfield (1955) — Citation indexing: the infrastructure that makes retraction tracking possible.

29.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
#>  [5] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4         ggraph_2.2.2
#>  [9] tidygraph_1.3.1          igraph_2.3.4
#> [11] quanteda_4.5.1           pdftools_3.9.1
#> [13] arrow_26.0.1.1           bibliometrix_5.5.0
#> [15] RefManageR_1.4.0         bib2df_1.1.2.0
#> [17] rcrossref_1.2.1          gt_1.4.0
#> [19] tidytext_0.4.3           glue_1.8.1
#> [21] openalexR_3.1.2          lubridate_1.9.5
#> [23] forcats_1.0.1            stringr_1.6.0
#> [25] dplyr_1.2.1              purrr_1.2.2
#> [27] readr_2.2.0              tidyr_1.3.2
#> [29] tibble_3.3.1             ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4      tokenizers_0.3.0       plyr_1.8.9
```

```
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3             S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0          MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3          lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1        tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3             bookdown_0.48          stats4_4.6.1
#>  [91] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0          cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3          openxlsx_4.2.9         rlang_1.3.0
```

```
#> [118] fastmatch_1.1-8
```

Chapter 30

Funding Analysis and Grant-to-Publication Linkage

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

30.1 Learning objectives

After completing this chapter, you will be able to:

- Extract funder information from OpenAlex work records
- Compute funder-level publication counts and citation impact
- Analyse co-funding patterns (papers acknowledging multiple funders)
- Interpret funding data critically, accounting for coverage and attribution challenges
- Discuss why funding-output linkage is inherently imprecise

30.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "funding_helpers.R"))
source(here::here("R", "sci_palette.R"))
```

30.3 Conceptual background

Funding agencies invest billions in research and want to understand what their investment produces. **Funding acknowledgement analysis** extracts funder names and grant identifiers from publication metadata or full text, linking grants to the papers they supported.

OpenAlex includes funder information for many works, derived from Crossref's funding metadata (which in turn comes from publisher-reported FundRef data). Coverage varies: journals that participate in Crossref's FundRef initiative have good coverage; others may have no funding data at all (Priem et al., 2022).

Key analytical questions include: How many publications does a funder support? What is the citation impact of funded vs. unfunded research? Which funders frequently co-fund research? How does a funder's portfolio map onto the landscape of science?

The fundamental challenge is **attribution**: a paper acknowledging three funders and seven authors is the result of contributions from all of them. Assigning full credit to each funder double-counts; fractional credit is more principled but harder to implement. Similar issues arise with multi-source funding: a paper may acknowledge an infrastructure grant, a project grant, and a fellowship, each with different levels of contribution.

30.4 Worked example

30.4.1 Extracting funder data

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2020-01-01",
```

```
  to_publication_date = "2023-12-31",
  options = list(sample = 400, seed = 42)
)

funding_works <- works |>
  mutate(funders = map(funders, sci_funder_rows),
         has_funder_metadata = map_int(funders, nrow) > 0)

funded <- funding_works |>
  filter(has_funder_metadata) |>
  select(id, display_name, cited_by_count, funders)

cat(glue("Works with observed funder IDs: {nrow(funded)} /
 ↪  {nrow(works)}
 ↪  ({scales::percent(nrow(funded)/nrow(works))})\n"))
```

```
#> Works with observed funder IDs: 228 / 400 (57%)
```

```
if (!nrow(funded)) {
  cat("No usable funder records occur in this sample. Funder
   ↪  summaries are unavailable; missing metadata does not
   ↪  establish that the papers were unfunded.\n")
}
```

```
funder_data <- if (nrow(funded)) {
  funded |>
    unnest(funders, names_sep = "_") |>
    select(work_id = id, funder_id = funders_id,
           funder = funders_display_name, cited_by_count)
} else {
  tibble(work_id = character(), funder_id = character(),
         funder = character(), cited_by_count = numeric())
}

cat(glue("Funder-work pairs: {nrow(funder_data)}\n"))
```

```
#> Funder-work pairs: 365
```

```
cat(glue("Unique funders:
 ↪  {n_distinct(funder_data$funder_id)}\n"))
```

```
#> Unique funders: 195
```

30.4.2 Top funders by publication count

```
funder_counts <- funder_data |>
  group_by(funder_id) |>
  summarise(funder = first(funder), n = n_distinct(work_id),
   ↪  .groups = "drop") |>
  arrange(desc(n))

if (nrow(funder_counts)) {
funder_counts |>
  head(15) |>
  mutate(funder = fct_reorder(funder, n)) |>
  ggplot(aes(x = n, y = funder)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Publications supported", y = NULL) +
  theme_sci()
} else {
  cat("No funder publication-count plot: no observed funder
   ↪  records.\n")
}
```



Figure 30.1: Up to 15 funders by the number of sampled publications with observed funder metadata.

30.4.3 Citation impact by funder

```
funder_impact <- funder_data |>
  group_by(funder_id) |>
  summarise(
    funder = first(funder),
    n_pubs = n(),
    mean_cites = round(mean(cited_by_count), 1),
    median_cites = median(cited_by_count),
    .groups = "drop"
  ) |>
  filter(n_pubs >= 5) |>
  arrange(desc(mean_cites))

if (nrow(funder_impact)) {
  funder_impact |> head(10) |> gt()
} else {
  cat("No funder has at least five sampled publications with
   ↪  usable metadata; no funder citation summary is
   ↪  reported.\n")
}
```

funder_id	funder	n_pubs	mean_cites	median_cites
https://openalex.org/F4320322025	Conselho Nacional de Desenvolvimento Científico e Tecnológico	6	58.3	45.5
https://openalex.org/F4320335869	National Social Science Fund of China	8	37.2	20.0
https://openalex.org/F4320327557	National Office for Philosophy and Social Sciences	6	36.0	25.5
https://openalex.org/F4320322120	National Research Foundation of Korea	5	34.2	20.0
https://openalex.org/F4320321091	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior	8	30.8	20.5
https://openalex.org/F4320321114	Bundesministerium für Bildung und Forschung	10	30.8	25.0
https://openalex.org/F4320321001	National Natural Science Foundation of China	49	28.1	24.0
https://openalex.org/F4320320879	Deutsche Forschungsgemeinschaft	6	26.3	25.5
https://openalex.org/F4320320719	Department of Science and Technology, Ministry of Science and Technology, India	5	26.2	17.0
https://openalex.org/F4320334764	Japan Society for the Promotion of Science	6	24.3	17.0

30.4.4 Comparison by observed funder metadata

```
if (n_distinct(funding_works$has_funder_metadata) == 2L) {
funding_works |>
  mutate(funder_metadata = if_else(has_funder_metadata,
                                   "Observed funder", "No observed
                                    ↪  funder")) |>
  ggplot(aes(x = funder_metadata, y = cited_by_count + 1)) +
  geom_boxplot(fill = palette_sci(1), alpha = 0.7) +
  scale_y_log10() +
  labs(x = "Funder metadata", y = "Citations (log scale)") +
  theme_sci()
} else {
  cat("No two-group citation comparison: only one
   ↪  funder-metadata group is observed.\n")
}
```

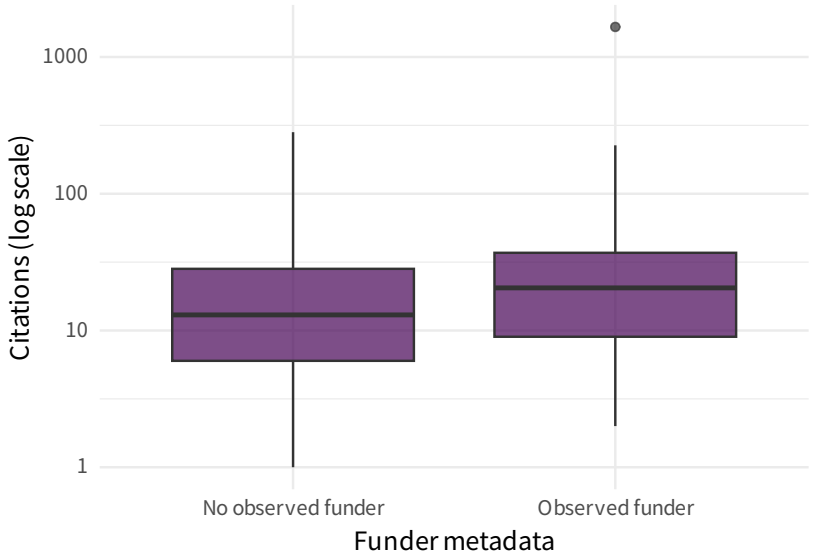


Figure 30.2: Citation distribution by presence of observed funder metadata; missing metadata does not establish absence of funding.

30.5 Diagnostics and interpretation

- **Coverage rate**: The fraction of papers with funder data depends on the journal, publisher, and field. Report this rate before interpreting

funder statistics.
- **Funder name normalisation**: The same funder may appear under different names ("NSF", "National Science Foundation", "US NSF"). Merge variants using funder IDs where available.
- **Selection and metadata coverage**: Papers with observed funder records may differ systematically from papers without them. Their citation gap can reflect selection and publisher reporting; it does not identify the causal effect of funding or compare confirmed funded and unfunded groups.
- **Multi-funder attribution**: Papers acknowledging multiple funders inflate the publication count for each. Consider fractional counting.

30.6 Limitations and responsible use

30.7 Limitations and responsible use

- **Coverage is incomplete.** Not all publishers report funding data to Crossref. Missing funder acknowledgements do not mean research was unfunded.
- **Attribution is ambiguous.** A grant acknowledgement does not specify which part of the work the grant supported. Infrastructure grants, salary support, and project funding all appear identically.
- **Correlation ≠ causation.** Funded papers receive more citations, but this may reflect that better-resourced labs produce both more funding applications and more cited papers.
- **Do not rank funders by citation impact.** Funders support different fields, career stages, and risk levels. Comparing mean citations across funders is misleading without field normalisation (Hicks et al., 2015).

30.8 Common pitfalls

30.9 Common pitfalls

- **Ignoring funder coverage rates.** Comparing funder portfolios without accounting for differential coverage produces biased results.
- **Treating every acknowledgement as equal.** A paper may acknowledge a major project grant and a minor travel grant. Both appear identically in metadata.

- **Comparing across fields.** A biomedical funder's publications will have higher citation counts than a humanities funder's publications due to field-level citation norms.
- **Using funder data for individual evaluation.** Whether a researcher has funding reflects many factors beyond research quality: field norms, career stage, institutional support, and luck.

30.10 Exercises

1. **Funder co-occurrence.** Identify papers with multiple funders. Which funder pairs most frequently co-fund research?
2. **Funder portfolios.** For two major funders, fetch their supported publications and compare topical profiles using keyword analysis.
3. **Temporal trends.** Track the number of papers with funding acknowledgements by year. Is the coverage rate increasing over time?

30.11 Solutions

Solutions are provided in E.

30.12 Further reading

- Priem et al. (2022) — OpenAlex funder metadata from Crossref FundRef.
- Hicks et al. (2015) — Responsible use of metrics in funding evaluation.
- Waltman (2016) — Citation indicators and their application to funding assessment.

30.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
```

```
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
#>  [5] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4         ggraph_2.2.2
#>  [9] tidygraph_1.3.1          igraph_2.3.4
#> [11] quanteda_4.5.1           pdftools_3.9.1
#> [13] arrow_26.0.1.1           bibliometrix_5.5.0
#> [15] RefManageR_1.4.0         bib2df_1.1.2.0
#> [17] rcrossref_1.2.1          gt_1.4.0
#> [19] tidytext_0.4.3           glue_1.8.1
#> [21] openalexR_3.1.2          lubridate_1.9.5
#> [23] forcats_1.0.1            stringr_1.6.0
#> [25] dplyr_1.2.1              purrr_1.2.2
#> [27] readr_2.2.0              tidyr_1.3.2
#> [29] tibble_3.3.1             ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
```

```
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4     tokenizers_0.3.0      plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3             S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0          MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3          lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1        tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3             bookdown_0.48          stats4_4.6.1
#>  [91] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0          cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
```

```
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3          openxlsx_4.2.9        rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 31

Patent–Paper Linkages

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

31.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the significance of patent–paper citations for understanding research impact
- Specify the paper and patent identifiers needed to establish an observed linkage
- Explore scholarly citation metadata without treating it as patent-citation evidence
- Discuss the limitations of patent citations as indicators of technological impact
- Distinguish between direct citation (paper cited in patent) and indirect influence

31.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)
```

```
set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

31.3 Conceptual background

When a patent examiner or inventor cites a scientific paper in a patent document, it creates an observable link between knowledge production (science) and knowledge application (technology). These **patent–paper citations** (also called non-patent literature or NPL citations) are one of the few quantitative measures of the science–technology interface.

Patent citations serve a legal function: they define the prior art against which the patent's novelty is assessed. Not all patent–paper citations reflect genuine intellectual influence — some are added by examiners for legal completeness, not because the inventor actually used the cited research. Nevertheless, at the aggregate level, patent citation patterns reveal which fields, journals, and institutions produce research that feeds into technological innovation.

The biomedical and chemical sciences dominate patent–paper linkages because pharmaceutical and chemical patents routinely cite scientific literature. Physics and engineering have moderate linkages. Social sciences and humanities are rarely cited in patents, not because they lack impact, but because their impact operates through different channels (policy, education, culture).

The OpenAlex fields retrieved in this example describe scholarly works and scholarly citations. They contain no citing patent identifiers, patent citation dates, or patent-family identifiers. Establishing a patent–paper linkage requires a separate, documented source of patent references and a validated match to the cited paper. The analysis below does not perform that linkage.

31.4 Worked example

This executable example explores a 500-article scholarly-citation sample. It is preparation for a possible future linkage analysis, not a substitute for patent data. A high scholarly citation count does not identify a patent-cited paper. No patent-citation prevalence, rate, or technological impact is estimated here.

31.4.1 Preparing a scholarly-citation sample

```
works <- fetch_openalex(
  entity = "works",
  from_publication_date = "2015-01-01",
  to_publication_date = "2020-12-31",
  type = "article",
  options = list(sample = 500, seed = 42)
)

works_citations <- works |>
  transmute(
    id, display_name, source_display_name,
    year = year(publication_date),
    cited_by_count,
    referenced_works_count,
    type
  )

cat(glue("Total works: {nrow(works_citations)}\n"))
```

```
#> Total works: 500
```

31.4.2 Scholarly citation distribution

```
p90 <- quantile(works_citations$cited_by_count, 0.90)

ggplot(works_citations, aes(x = cited_by_count)) +
  geom_histogram(binwidth = 10, fill = palette_sci(1), colour =
   ↪  "white") +
  geom_vline(xintercept = p90, linetype = "dashed", colour =
   ↪  "red") +
  annotate("text", x = p90 + 20, y = Inf, label = "90th
   ↪  percentile",
           vjust = 2, colour = "red", size = 3) +
  labs(x = "Scholarly citation count", y = "Papers") +
  theme_sci()
```

31.4.3 Journals represented among highly cited sampled papers

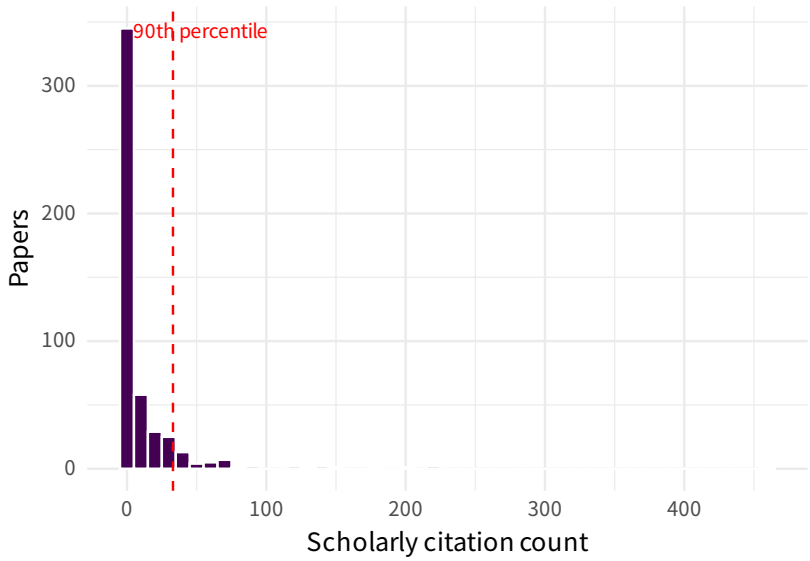


Figure 31.1: Scholarly citation counts in the sample, with the sample 90th percentile marked.

```
works_citations |>
  filter(cited_by_count >= p90, !is.na(source_display_name)) |>
  count(source_display_name, sort = TRUE) |>
  head(15) |>
  mutate(source_display_name = fct_reorder(source_display_name,
   ↪  n)) |>
  ggplot(aes(x = n, y = source_display_name)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Highly cited sampled papers", y = NULL) +
  theme_sci()
```

31.4.4 Descriptive characteristics by scholarly-citation group

```
works_citations |>
  mutate(high_cite = cited_by_count >= p90) |>
  group_by(high_cite) |>
  summarise(
    n = n(),
    mean_refs = round(mean(referenced_works_count, na.rm =
     ↪  TRUE), 1),
```

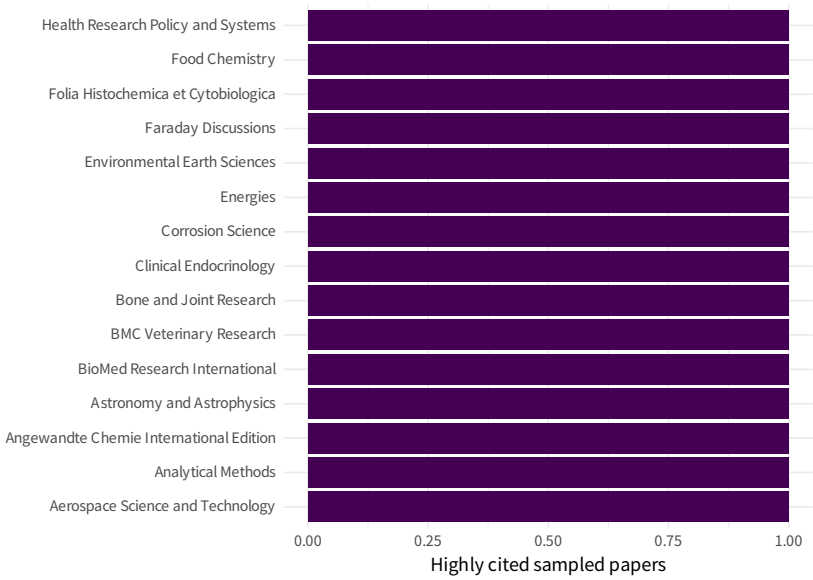


Figure 31.2: Up to 15 journals with the most sampled papers at or above the sample scholarly-citation 90th percentile.

```
    mean_year = round(mean(year), 1),
    .groups = "drop"
  ) |>
  gt()
```

high_cite	n	mean_refs	mean_year
FALSE	449	11.0	2017
TRUE	51	50.3	2017

31.5 Diagnostics and interpretation

- **Measured quantity**: `cited_by_count` in this analysis counts scholarly citations. Neither the threshold nor the journal plot measures patent citations.
- **Sample and threshold**: Counts refer only to the retrieved sample, not total journal output. Ties at the 90th percentile can place more than 10% of papers in the high-citation group.
- **Citation window**: Current cumulative counts give older publication cohorts more time to receive citations. They are not a fixed-window impact measure.

- **Field normalisation**: Biomedical papers are far more likely to be cited in patents than social science papers. Compare patent citation rates within fields, not across them.
- **Time lag**: Estimating a paper-to-patent citation delay requires both dates from actual linked records and a stated observation cutoff; current scholarly citation counts cannot supply it.
- **Examiner vs. inventor citations**: Patent examiners add citations that inventors may not have used. This inflates apparent science–technology linkage.

31.6 Limitations and responsible use

31.7 Limitations and responsible use

- **Patent citations are a narrow measure of impact.** Much research influences technology without being cited in patents — through consulting, training, open-source software, or informal knowledge transfer.
- **Field coverage differs.** Patent citations capture only some forms of influence. Comparing fields without considering these differences can misrepresent their contributions (Hicks et al., 2015).
- **Geographic bias.** Patent systems vary by country. US and European patents have different citation practices. Analyses based on one patent office may not generalise.
- **Not evaluative for individuals.** A researcher whose paper is cited in a patent may have had no involvement with the technology. Patent citation reflects relevance of the knowledge, not the researcher's intent to innovate.

31.8 Common pitfalls

31.9 Common pitfalls

- **Equating patent citations with commercialisation.** A patent citation means the paper is prior art, not that the research led to a product.
- **Ignoring time lags.** Comparing patent citation rates for papers published in 2023 vs. 2010 is meaningless — the 2023 papers have not had time to be cited in patents.
- **Double-counting across patent families.** The same invention may be patented in multiple countries. Count at the patent family level, not the individual patent level.

- **Substituting scholarly citations for patent linkages.** High scholarly citation counts do not establish that a paper was cited by a patent. Require an observed patent–paper reference before classifying a paper as patent-cited.

31.10 Exercises

1. **Scholarly-citation comparison.** Compare citation distributions in two journal samples, reporting sample sizes, field differences, publication years, retrieval date and threshold choice. Explain why the result does not estimate patent-citation potential.

2. **Linkage data specification.** Design a table containing a paper identifier, citing patent identifier, patent-family identifier, citation date, source and match status. Describe how you would verify ambiguous references and retain unmatched records. This exercise requires actual patent-reference data before any linkage rates can be calculated.

3. **Institutional patent linkage.** Given an independently validated patent–paper dataset, define the publication denominator, observation window and patent-family deduplication rule for two institutions. Report coverage and uncertainty. If those data are unavailable, identify the missing inputs rather than using scholarly citation percentiles as a proxy.

31.11 Solutions

Solutions are provided in E.

31.12 Further reading

- Narin et al. (1997) — Foundational analysis of the linkage between science and technology through patent citations.
- Hicks et al. (2015) — Responsible metrics: why patent citations should not be the sole indicator of research impact.
- Priem et al. (2022) — OpenAlex scholarly-work metadata; this example does not establish patent-citation coverage.

31.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5                Matrix_1.7-5
#>  [3] word2vec_0.4.1            stm_1.3.10
#>  [5] topicmodels_0.2-17        quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4          ggraph_2.2.2
#>  [9] tidygraph_1.3.1           igraph_2.3.4
#> [11] quanteda_4.5.1            pdftools_3.9.1
#> [13] arrow_26.0.1.1            bibliometrix_5.5.0
#> [15] RefManageR_1.4.0          bib2df_1.1.2.0
#> [17] rcrossref_1.2.1           gt_1.4.0
#> [19] tidytext_0.4.3            glue_1.8.1
#> [21] openalexR_3.1.2           lubridate_1.9.5
#> [23] forcats_1.0.1             stringr_1.6.0
#> [25] dplyr_1.2.1               purrr_1.2.2
#> [27] readr_2.2.0               tidyr_1.3.2
#> [29] tibble_3.3.1              ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
```

```
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             janeaustenr_1.0.0
 ↪  cellranger_1.1.0
#>  [19] htmlwidgets_1.6.4     tokenizers_0.3.0      plyr_1.8.9
#>  [22] httr2_1.3.0            cachem_1.1.0
 ↪  plotly_4.12.1
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3            S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0         MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1            otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3          lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3             bookdown_0.48          stats4_4.6.1
```

```
#>  [91] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      DT_0.34.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1            yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0          cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3          openxlsx_4.2.9         rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 32

Causal Inference in Scientometrics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

32.1 Learning objectives

After completing this chapter, you will be able to:

- Distinguish causal from correlational claims in bibliometric research
- Explain the intervention and outcome data required for difference-in-differences
- Compute descriptive OA-by-publication-cohort comparisons without assigning a causal interpretation
- Distinguish a within-year comparison from validated matching or a regression-discontinuity design
- Recognise the assumptions underlying causal inference in observational bibliometric data

32.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
```

```
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

32.3 Conceptual background

Bibliometrics is rich in correlations: OA papers are more cited than closed-access papers; funded research receives more citations; papers in prestigious journals are cited more. But correlation does not imply causation. The OA citation advantage may reflect self-selection (better papers are more likely to be made OA), not the causal effect of open access. Separating cause from confound is essential for evidence-based science policy.

Difference-in-differences (DiD) exploits policy changes as natural experiments. If a funder mandates OA starting in year T, papers from that funder before and after T serve as the treatment group, while papers from a similar funder without the mandate serve as the control. The causal effect is estimated as the difference in trends between treatment and control groups. The key assumption is parallel trends: absent the intervention, both groups would have followed the same trajectory.

Regression discontinuity design (RDD) applies when treatment is assigned by a threshold. For example, grants are funded above a score cutoff; papers are accepted above a review score threshold. Comparing outcomes just above and just below the threshold isolates the causal effect of the treatment, under the assumption that units near the threshold are comparable.

Matching methods create comparable groups from observational data. Propensity score matching pairs treated units (e.g., OA papers) with control units (closed-access papers) that have similar observable characteristics (field, journal, year, team size). This reduces confounding but cannot eliminate unmeasured confounders.

All three methods require strong assumptions. In bibliometrics, violations are common: the parallel trends assumption may fail because of concurrent policy changes; the RDD threshold may be manipulable; matching cannot address unmeasured confounders like paper quality. Hicks et al. (2015) emphasise that quantitative analysis, however sophisticated, cannot replace expert judgement in research evaluation.

32.4 Worked example

32.4.1 A descriptive OA-by-publication-cohort comparison

This example samples papers from one journal and compares their current cumulative citation counts. The 2020 split is an arbitrary publication-cohort boundary, not a documented intervention. OA status is observed at retrieval, not recorded treatment assignment. Different publication years contain different papers with different citation exposure. Consequently, neither the interaction regression nor the cohort plot identifies a policy effect or tests a DiD parallel-trends assumption. DiD requires a defensible intervention, comparison design and outcome histories; see Callaway and Sant'Anna (2021).

The sample and citation snapshot describe the fetched data only; they are not journal totals or a frozen historical citation panel.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2016-01-01",
  to_publication_date = "2023-12-31",
  type = "article",
  options = list(sample = 500, seed = 42)
)

cohort_data <- works |>
  transmute(
    id, year = year(publication_date),
    cited_by_count,
    is_oa = case_when(
      oa_status %in% c("gold", "green", "hybrid", "bronze",
↪   "diamond") ~ TRUE,
      oa_status == "closed" ~ FALSE,
      TRUE ~ NA
    ),
    recent_cohort = year >= 2020,
    referenced_works_count
  )

excluded_records <- sum(!complete.cases(cohort_data))
cohort_data <- cohort_data |> drop_na()
stopifnot(!anyDuplicated(cohort_data$id), nrow(cohort_data) > 0)
cat(glue("Excluded incomplete records: {excluded_records}\n"))
```

```
#> Excluded incomplete records: 0
```

```
cat(glue("Sampled complete records: {nrow(cohort_data)}\n"))
```

```
#> Sampled complete records: 500
```

```
cat(glue("OA: {sum(cohort_data$is_oa)}, Closed:
 ↪  {sum(!cohort_data$is_oa)}\n"))
```

```
#> OA: 210, Closed: 290
```

32.4.2 Descriptive comparison

```
cohort_summary <- cohort_data |>
  group_by(is_oa, recent_cohort) |>
  summarise(mean_cites = mean(cited_by_count), n = n(), .groups
   ↪  = "drop") |>
  mutate(period = ifelse(recent_cohort, "2020-2023",
   ↪  "2016-2019"),
         group = ifelse(is_oa, "Open access", "Closed"))

ggplot(cohort_summary, aes(x = period, y = mean_cites, fill =
 ↪  group)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = palette_sci(2)) +
  labs(x = "Publication cohort", y = "Mean cumulative citations
   ↪  at retrieval", fill = "Access") +
  theme_sci()
```

32.4.3 Descriptive interaction regression

The interaction describes how the conditional OA/closed citation difference varies between the two sampled publication cohorts. Adding reference count is statistical adjustment, not an identification strategy. The fitted coefficients and conventional linear-model uncertainty summaries must not be reported as causal effects.

```
interaction_model <- lm(cited_by_count ~ is_oa * recent_cohort +
 ↪  referenced_works_count,
               data = cohort_data)

tidy_interaction <- broom::tidy(interaction_model)
tidy_interaction |>
```

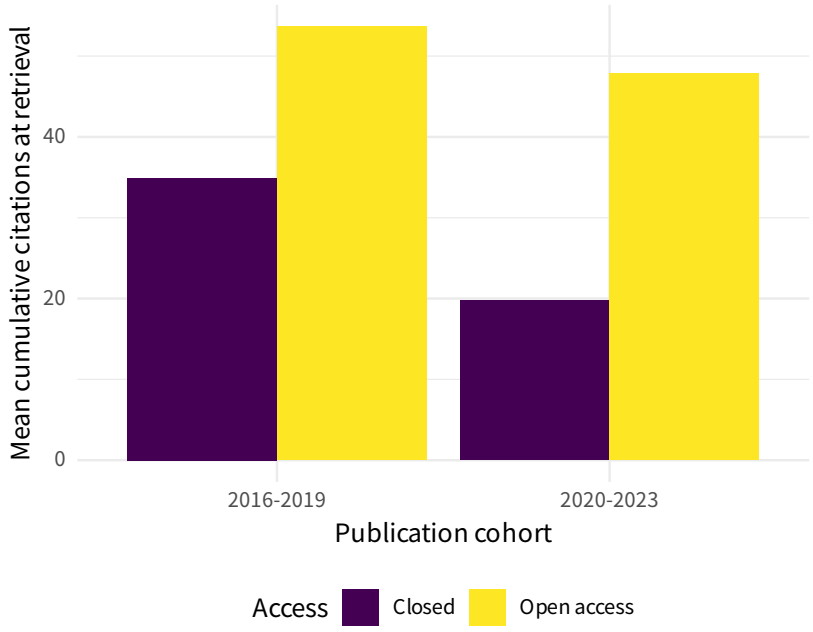


Figure 32.1: Current cumulative citations in sampled papers, grouped by OA status and publication cohort; not a policy-effect estimate.

```
  mutate(across(where(is.numeric), \(x) round(x, 3))) |>
  gt()
```

term	estimate	std.error	statistic	p.value
(Intercept)	21.53	8.661	2.486	0.013
is_oaTRUE	19.36	11.010	1.758	0.079
recent_cohortTRUE	-18.30	9.880	-1.852	0.065
referenced_works_count	0.30	0.124	2.415	0.016
is_oaTRUE: recent_cohortTRUE	9.64	15.179	0.635	0.526

32.4.4 Within-year sampling comparison

For each sampled OA paper, select one closed-access paper from the same publication year. Controls can be reused, and OA papers in years with no closed-access candidate are omitted. This is sampling with replacement within year, not propensity-score matching; it does not establish covariate balance or remove confounding.

```
oa_papers <- cohort_data |> filter(is_oa)
closed_papers <- cohort_data |> filter(!is_oa)

matched <- oa_papers |>
  mutate(match_year = year) |>
  inner_join(
    closed_papers |> select(year, control_id = id,
 ↪  cited_by_count_closed = cited_by_count),
    by = "year",
    relationship = "many-to-many"
  ) |>
  group_by(id) |>
  slice_sample(n = 1) |>
  ungroup()

cat(glue("Within-year pairs: {nrow(matched)} / {nrow(oa_papers)}
 ↪  OA papers\n"))
```

```
#> Within-year pairs: 210 / 210 OA papers
```

```
cat(glue("Distinct closed-access controls:
 ↪  {n_distinct(matched$control_id)}\n"))
```

```
#> Distinct closed-access controls: 150
```

```
cat(glue("Mean cites (OA): {round(mean(matched$cited_by_count),
 ↪  1)}\n"))
```

```
#> Mean cites (OA): 50.3
```

```
cat(glue("Mean cites (closed match):
 ↪  {round(mean(matched$cited_by_count_closed), 1)}\n"))
```

```
#> Mean cites (closed match): 25.7
```

32.4.5 Publication-cohort profile

The horizontal axis is publication year. Each point summarises different papers' cumulative citations at one retrieval time. These are not citations received annually by fixed treated and control groups.

```
trends <- cohort_data |>
  group_by(year, is_oa) |>
  summarise(mean_cites = mean(cited_by_count), .groups = "drop")
   ↪  |>
  mutate(group = ifelse(is_oa, "OA", "Closed"))

ggplot(trends, aes(x = year, y = mean_cites, colour = group)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  geom_vline(xintercept = 2019.5, linetype = "dashed", colour =
   ↪  "grey50") +
  scale_colour_manual(values = palette_sci(2)) +
  labs(x = "Publication year", y = "Mean cumulative citations at
   ↪  retrieval", colour = "Access") +
  theme_sci()
```

32.5 Diagnostics and interpretation

- **Selection and exposure**: Report the sampling query, retrieval date, exclusions and group sizes. Older publication cohorts have longer citation exposure; snapshot OA status may differ from status at publication.
- **Interpret the actual model**: The OA-by-cohort coefficient is a descriptive conditional association. The arbitrary 2020 boundary supplies no intervention and no causal estimand.

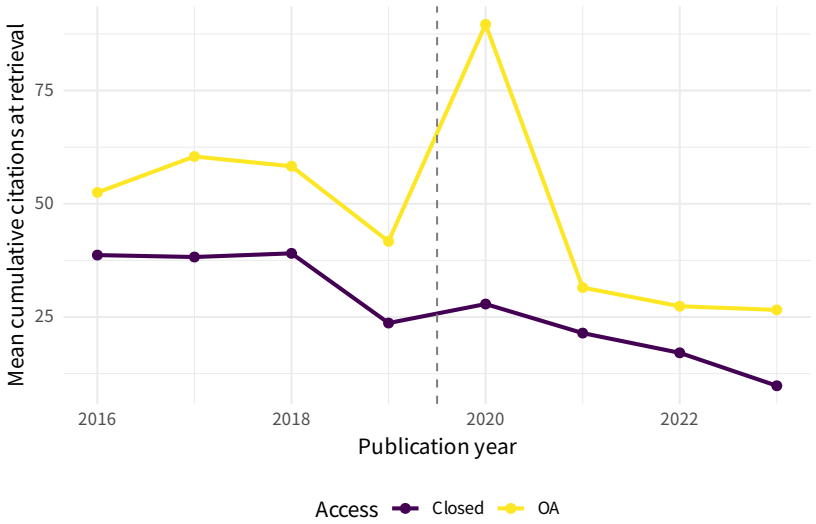


Figure 32.2: Current cumulative citations by OA status and publication year in the sample. This cohort profile does not test parallel trends.

- **Assess the comparison**: Check years without eligible controls, control reuse and covariate differences. One sampled same-year control does not make papers exchangeable.
- **Causal designs need additional evidence**: Define intervention timing and assignment, comparable outcomes over time, the target effect and identifying assumptions before choosing a DiD, RDD or matching analysis. A non-significant pre-trend test alone does not establish parallel counterfactual trends.

32.6 Limitations and responsible use

32.7 Limitations and responsible use

- **This example is descriptive.** It has no documented intervention or identified causal design. Observational causal designs can support causal interpretation under explicit, defensible assumptions; fitting the shown regression does not supply them.
- **Unmeasured confounders.** Paper quality, author reputation, and marketing effort are rarely observable in bibliometric data but strongly affect citations.

- **Treatment definition.** "OA" is not a single treatment — it encompasses gold, green, hybrid, and bronze, each with different mechanisms.
- **Generalisability.** This sampled single-journal comparison does not establish an OA effect in this journal or elsewhere.
- **Policy implications require judgement.** Even a well-estimated causal effect is one input into policy decisions, not a sufficient basis for mandates (Hicks et al., 2015; American Society for Cell Biology, 2012).

32.8 Common pitfalls

32.9 Common pitfalls

- **Claiming causation from regression coefficients.** A positive OA coefficient in a regression does not prove OA causes more citations without addressing confounders.
- **Calling a cohort profile a pre-trend test.** Current cumulative citations in different publication cohorts do not measure treated/control outcome paths around a policy change.
- **Treating matching as automatic identification.** Choose covariates from the causal question and timing, evaluate overlap/balance, and avoid conditioning indiscriminately on post-treatment variables.
- **Over-interpreting p-values.** A $p < 0.05$ does not mean the result is practically significant or that the causal assumptions hold.

32.10 Exercises

1. **Interaction sensitivity.** Compare the OA-by-cohort coefficient with and without adjustment for reference count. Explain why a changed coefficient does not itself diagnose a causal effect or quantify omitted-variable bias.
2. **Boundary sensitivity.** Repeat the descriptive interaction using a 2018 publication-cohort boundary. Explain why this is not a placebo intervention test and why a non-significant coefficient cannot establish parallel trends.
3. **Comparison audit.** Count reused closed-access controls and unmatched OA records in the within-year comparison. Specify the intervention, assignment information and longitudinal outcomes needed for an actual policy-effect study.

32.11 Solutions

Solutions are provided in E.

32.12 Further reading

- Hicks et al. (2015) — The Leiden Manifesto: caution against mechanistic application of quantitative metrics.
- American Society for Cell Biology (2012) — DORA: responsible assessment requires understanding causation.
- Waltman (2016) — Review of citation indicators and their interpretation challenges.
- Callaway and Sant'Anna (2021) — Identification, estimation and inference for DiD with multiple periods.

32.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
```

```
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5                Matrix_1.7-5
#>  [3] word2vec_0.4.1            stm_1.3.10
#>  [5] topicmodels_0.2-17        quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4          ggraph_2.2.2
#>  [9] tidygraph_1.3.1           igraph_2.3.4
#> [11] quanteda_4.5.1            pdftools_3.9.1
#> [13] arrow_26.0.1.1            bibliometrix_5.5.0
#> [15] RefManageR_1.4.0          bib2df_1.1.2.0
#> [17] rcrossref_1.2.1           gt_1.4.0
#> [19] tidytext_0.4.3            glue_1.8.1
#> [21] openalexR_3.1.2           lubridate_1.9.5
#> [23] forcats_1.0.1             stringr_1.6.0
#> [25] dplyr_1.2.1               purrr_1.2.2
#> [27] readr_2.2.0               tidyr_1.3.2
#> [29] tibble_3.3.1              ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             broom_1.0.13
 ↪  janeaustenr_1.0.0
#>  [19] cellranger_1.1.0       htmlwidgets_1.6.4
 ↪  tokenizers_0.3.0
#>  [22] plyr_1.8.9            httr2_1.3.0           cachem_1.1.0
#>  [25] plotly_4.12.1          dimensionsR_0.0.3      mime_0.13
#>  [28] lifecycle_1.0.5        pkgconfig_2.0.3        R6_2.6.1
#>  [31] fastmap_1.2.0          shiny_1.15.0
 ↪  digest_0.6.39
#>  [34] patchwork_1.3.2        shinycssloaders_1.1.0
 ↪  rprojroot_2.1.1
#>  [37] RSpectra_0.16-2        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [40] urltools_1.7.3.1      timechange_0.4.0       mgcv_1.9-4
#>  [43] httr_1.4.9             polyclip_1.10-7
 ↪  compiler_4.6.1
#>  [46] here_1.0.2            bit64_4.8.6           withr_3.0.3
#>  [49] backports_1.5.1        S7_1.0.0
 ↪  viridis_0.6.5
```

```
#>  [52] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [55] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [58] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
#>  [61] nlme_3.1-169           promises_1.5.0         grid_4.6.1
#>  [64] stringdist_0.9.17      reshape2_1.4.5
 ↪  generics_0.1.4
#>  [67] gtable_0.3.6           tzdb_0.5.0
 ↪  rscopus_0.9.0
#>  [70] ca_0.72                data.table_1.18.6.1    hms_1.1.4
#>  [73] xml2_1.6.0             utf8_1.2.6
 ↪  ggrepel_0.9.8
#>  [76] pillar_1.11.1         nsyllable_1.0.1       vroom_1.7.1
#>  [79] later_1.4.8           splines_4.6.1         tweenr_2.0.3
#>  [82] lattice_0.22-9         FNN_1.1.4.1             bit_4.6.0
#>  [85] tidyselect_1.2.1       tm_0.7-20
 ↪  miniUI_0.1.2
#>  [88] knitr_1.52             gridExtra_2.3.1         NLP_0.3-3
#>  [91] bookdown_0.48          stats4_4.6.1           crul_1.6.0
#>  [94] xfun_0.62              graphlayouts_1.2.5
 ↪  matrixStats_1.5.0
#>  [97] DT_0.34.0              humaniformat_0.6.0
 ↪  stringi_1.8.9
#> [100] qpdf_1.4.1             yaml_2.3.12
 ↪  evaluate_1.0.5
#> [103] httpcode_0.3.0        cli_3.6.6             xtable_1.8-8
#> [106] systemfonts_1.3.4      Rcpp_1.1.2
 ↪  readxl_1.6.0
#> [109] triebeard_0.4.1        XML_3.99-0.25
 ↪  parallel_4.6.1
#> [112] assertthat_0.2.1      pubmedR_1.1.0          slam_0.1-56
#> [115] viridisLite_0.4.3      scales_1.4.0
 ↪  crayon_1.5.3
#> [118] openxlsx_4.2.9         rlang_1.3.0
 ↪  fastmatch_1.1-8
```

Chapter 33

Reproducibility, RRIDs, and Badges

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

33.1 Learning objectives

After completing this chapter, you will be able to:

- Define reproducibility indicators and explain their role in research evaluation
- Search for data availability statements and open badges in bibliometric metadata
- Analyse the prevalence of reproducibility practices across journals and fields
- Discuss the relationship between transparency practices and citation impact
- Recognise the gap between reproducibility signalling and actual reproducibility

33.2 Setup

```
library(tidyverse)
library(openalexR)
```

```
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

33.3 Conceptual background

The "replication crisis" has prompted widespread adoption of transparency practices: open data, open code, pre-registration, and registered reports. Bibliometrics can measure the uptake of these practices at scale, providing evidence for policy interventions.

Open badges are visual indicators (icons) attached to published papers that certify the availability of data, materials, or pre-registration. Journals like Psychological Science pioneered their use. Studies suggest that badges increase data sharing rates, though the quality of shared data varies.

Research Resource Identifiers (RRIDs) are persistent identifiers for research resources: antibodies, cell lines, model organisms, and software tools. Including RRIDs in methods sections improves reproducibility by enabling precise identification of the materials used. The RRID initiative has been adopted by over 1,000 journals.

Data availability statements (DAS) declare whether the data underlying a study are available and where to find them. Mandatory DAS policies have become common, but many statements say "data available upon request" without providing actual access. The presence of a DAS is easy to detect bibliometrically; the quality of actual data sharing is much harder to assess.

Code availability is increasingly expected, especially in computational fields. Journals like Nature require code availability statements. Platforms like GitHub, Zenodo, and Code Ocean facilitate code sharing, and DOIs for code repositories enable citation and tracking.

These practices are measurable proxies for reproducibility, but they are not reproducibility itself. A paper with an open badge may have unusable data; a paper without one may be perfectly reproducible. Hicks et al. (2015) warn against using transparency indicators as mechanical evaluation criteria.

33.4 Worked example

33.4.1 Analysing access status and metadata coverage

The example below measures article access status and abstract coverage in OpenAlex. It does not detect data/code availability, RRIDs, badges, or successful replication. Those questions require separate full-text/resource extraction and validation; OA status is not a validated substitute for them.

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2018-01-01",
  to_publication_date = "2023-12-31",
  type = "article",
  options = list(sample = 400, seed = 42)
)

works_repro <- works |>
  transmute(
    id, display_name,
    year = year(publication_date),
    cited_by_count,
    oa_status,
    type,
    has_abstract = !is.na(abstract) & nzchar(trimws(abstract))
  )

cat(glue("Works: {nrow(works_repro)}\n"))
```

```
#> Works: 400
```

33.4.2 Observed article access status

```
works_repro |>
  group_by(year) |>
  summarise(oa_rate = mean(oa_status != "closed", na.rm = TRUE),
            .groups = "drop") |>
  ggplot(aes(x = year, y = oa_rate)) +
  geom_line(linewidth = 1, colour = palette_sci(1)) +
  geom_point(size = 2, colour = palette_sci(1)) +
  scale_y_continuous(labels = scales::percent, limits = c(0, 1))
  ↪  +
  labs(x = "Year", y = "OA rate") +
  theme_sci()
```

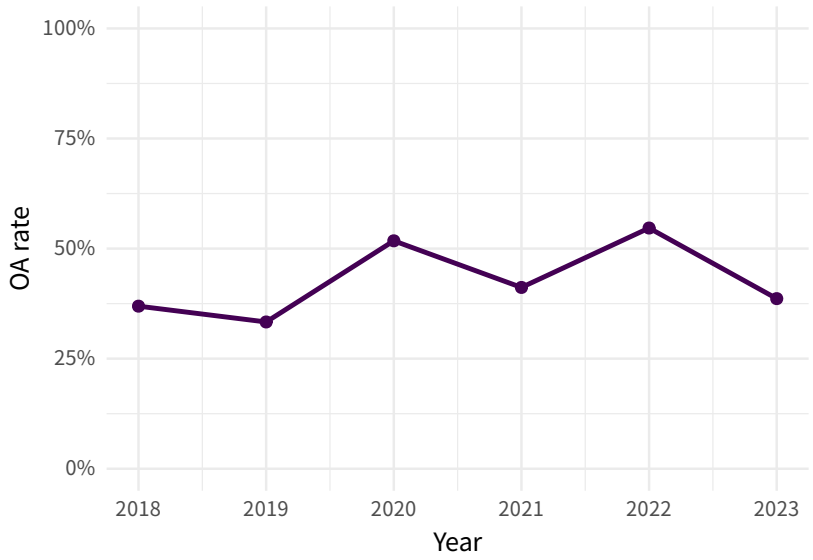


Figure 33.1: Open-access share among sampled articles with observed access status, by year.

33.4.3 Abstract availability as metadata completeness

```
works_repro |>
  group_by(year) |>
  summarise(abstract_rate = mean(has_abstract), .groups =
   ↪  "drop") |>
  ggplot(aes(x = year, y = abstract_rate)) +
  geom_line(linewidth = 1, colour = palette_sci(1)) +
  geom_point(size = 2, colour = palette_sci(1)) +
  scale_y_continuous(labels = scales::percent, limits = c(0, 1))
   ↪  +
  labs(x = "Year", y = "Abstract available") +
  theme_sci()
```

33.4.4 Access status and cumulative citations

```
works_repro |>
  mutate(is_oa = ifelse(oa_status != "closed", "OA", "Closed"))
   ↪  |>
```

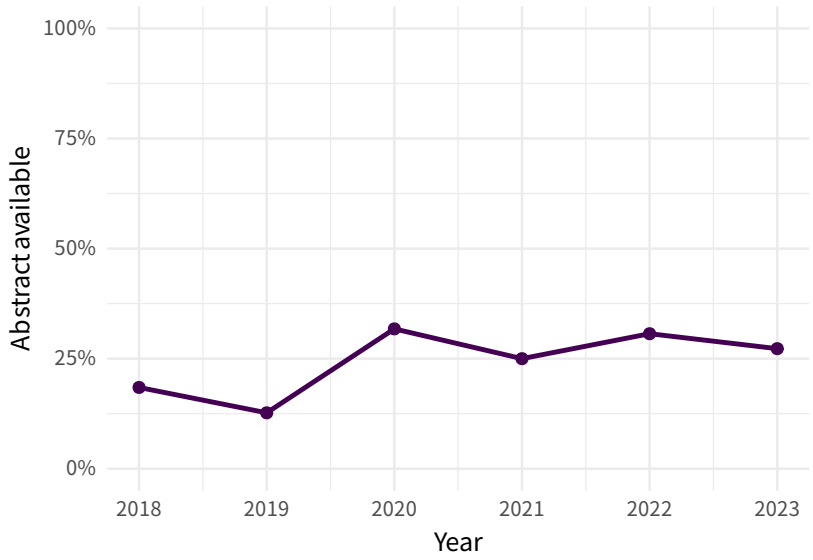


Figure 33.2: Abstract availability rate by year.

```
  ggplot(aes(x = is_oa, y = cited_by_count + 1)) +
  geom_boxplot(fill = palette_sci(1), alpha = 0.7) +
  scale_y_log10() +
  labs(x = "Access status", y = "Citations (log scale)") +
  theme_sci()
```

33.4.5 Summary statistics

```
works_repro |>
  mutate(is_oa = oa_status != "closed") |>
  group_by(is_oa) |>
  summarise(
    n = n(),
    mean_cites = round(mean(cited_by_count), 1),
    abstract_rate = scales::percent(mean(has_abstract)),
    .groups = "drop"
  ) |>
  gt()
```

is_oa	n	mean_cites	abstract_rate

FALSE	225	24.2	0%
TRUE	175	47.8	57%

33.5 Diagnostics and interpretation

- **Different constructs**: Article access status and abstract coverage do not measure data/code sharing, RRID use, badge adoption or reproducibility. Check full texts and linked resources for those outcomes.
- **Coverage over time**: Changes in recorded access status or abstract coverage can reflect provider coverage and publication cohorts. The example does not evaluate a DAS or badge policy.
- **Selection and exposure**: Current cumulative citations mix publication ages, article selection and other confounders. The comparison cannot establish an OA effect or a relationship with open data.
- **Field differences**: Data sharing norms vary enormously. Genomics shares data routinely; clinical research often cannot due to privacy. Compare within fields.

33.6 Limitations and responsible use

33.7 Limitations and responsible use

- **Signals are not substance.** A data availability statement does not guarantee the data are actually available, usable, or correct. Badges certify process, not outcome.
- **Mandates inflate compliance without improving practice.** When journals require DAS, compliance rises but “available upon request” becomes the default, not genuine sharing.
- **Reproducibility is multidimensional.** No single indicator captures it. Data availability, code availability, methods transparency, pre-registration, and statistical rigour all contribute independently.
- **Do not penalise fields where sharing is constrained.** Clinical data (privacy), indigenous data (sovereignty), and sensitive social data cannot be freely shared. Mandating open data policies uniformly across fields is inappropriate (Hicks et al., 2015).

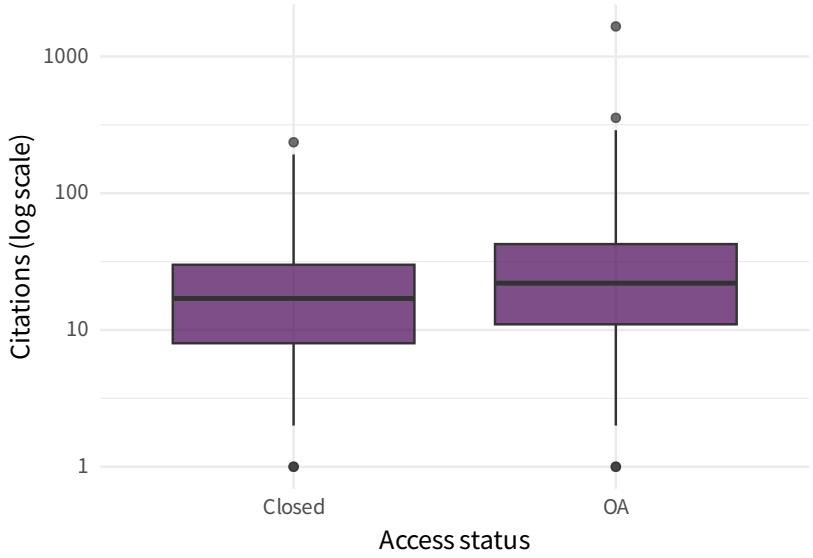


Figure 33.3: Current cumulative citation counts by recorded article access status.

33.8 Common pitfalls

33.9 Common pitfalls

- **Equating open data with reproducibility.** Data can be open but undocumented, incorrectly formatted, or insufficient to reproduce the analysis.
- **Counting badges without checking quality.** A badge means the journal certified availability at publication time. The link may be broken or the data may have been removed.
- **Ignoring negative results.** Reproducibility also requires publishing null results and failed replications. These are poorly captured by current transparency indicators.
- **Using transparency metrics as performance indicators.** Transparency practices should be valued, but mechanically rewarding them creates perverse incentives (superficial compliance).

33.10 Exercises

1. **DAS detection.** For papers with full abstracts, search for keywords like “data availability”, “code availability”, or “github.com” in the abstract text. What proportion mention data or code sharing?
2. **Cross-journal comparison.** Compare abstract availability and OA rates for three journals. Do journals with higher OA rates also have better metadata completeness?
3. **Temporal trends.** Track recorded OA status and abstract coverage by publication cohort. Report the usable denominators and explain why these measures do not establish data/code sharing or reproducibility.

33.11 Solutions

Solutions are provided in E.

33.12 Further reading

- Hicks et al. (2015) — The Leiden Manifesto: responsible use of metrics, including transparency indicators.
- Priem et al. (2022) — OpenAlex metadata as a basis for reproducibility analysis.
- American Society for Cell Biology (2012) — DORA: assessment should consider research practices, not just outputs.

33.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
#>  [5] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4         ggraph_2.2.2
#>  [9] tidygraph_1.3.1          igraph_2.3.4
#> [11] quanteda_4.5.1           pdftools_3.9.1
#> [13] arrow_26.0.1.1           bibliometrix_5.5.0
#> [15] RefManageR_1.4.0         bib2df_1.1.2.0
#> [17] rcrossref_1.2.1          gt_1.4.0
#> [19] tidytext_0.4.3           glue_1.8.1
#> [21] openalexR_3.1.2          lubridate_1.9.5
#> [23] forcats_1.0.1            stringr_1.6.0
#> [25] dplyr_1.2.1              purrr_1.2.2
#> [27] readr_2.2.0              tidyr_1.3.2
#> [29] tibble_3.3.1             ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             broom_1.0.13
 ↪  janeaustenr_1.0.0
```

```
#>  [19] cellranger_1.1.0       htmlwidgets_1.6.4
 ↪  tokenizers_0.3.0
#>  [22] plyr_1.8.9            httr2_1.3.0           cachem_1.1.0
#>  [25] plotly_4.12.1          dimensionsR_0.0.3      mime_0.13
#>  [28] lifecycle_1.0.5        pkgconfig_2.0.3        R6_2.6.1
#>  [31] fastmap_1.2.0          shiny_1.15.0
 ↪  digest_0.6.39
#>  [34] patchwork_1.3.2        shinycssloaders_1.1.0
 ↪  rprojroot_2.1.1
#>  [37] RSpectra_0.16-2        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [40] urltools_1.7.3.1      timechange_0.4.0       mgcv_1.9-4
#>  [43] httr_1.4.9             polyclip_1.10-7
 ↪  compiler_4.6.1
#>  [46] here_1.0.2            bit64_4.8.6           withr_3.0.3
#>  [49] backports_1.5.1        S7_1.0.0
 ↪  viridis_0.6.5
#>  [52] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [55] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [58] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
#>  [61] nlme_3.1-169          promises_1.5.0         grid_4.6.1
#>  [64] stringdist_0.9.17      reshape2_1.4.5
 ↪  generics_0.1.4
#>  [67] gtable_0.3.6           tzdb_0.5.0
 ↪  rscopus_0.9.0
#>  [70] ca_0.72                data.table_1.18.6.1    hms_1.1.4
#>  [73] xml2_1.6.0             utf8_1.2.6
 ↪  ggrepel_0.9.8
#>  [76] pillar_1.11.1         nsyllable_1.0.1        vroom_1.7.1
#>  [79] later_1.4.8           splines_4.6.1         tweenr_2.0.3
#>  [82] lattice_0.22-9         FNN_1.1.4.1             bit_4.6.0
#>  [85] tidyselect_1.2.1       tm_0.7-20
 ↪  miniUI_0.1.2
#>  [88] knitr_1.52             gridExtra_2.3.1         NLP_0.3-3
#>  [91] bookdown_0.48          stats4_4.6.1           crul_1.6.0
#>  [94] xfun_0.62              graphlayouts_1.2.5
 ↪  matrixStats_1.5.0
#>  [97] DT_0.34.0              humaniformat_0.6.0
 ↪  stringi_1.8.9
#> [100] qpdf_1.4.1             yaml_2.3.12
 ↪  evaluate_1.0.5
#> [103] httpcode_0.3.0        cli_3.6.6             xtable_1.8-8
#> [106] systemfonts_1.3.4      Rcpp_1.1.2
 ↪  readxl_1.6.0
#> [109] triebeard_0.4.1        XML_3.99-0.25
 ↪  parallel_4.6.1
#> [112] assertthat_0.2.1       pubmedR_1.1.0          slam_0.1-56
```

```
#> [115] viridisLite_0.4.3      scales_1.4.0
 ↪  crayon_1.5.3
#> [118] openxlsx_4.2.9         rlang_1.3.0
 ↪  fastmatch_1.1-8
```

Part VII

Production & Reporting

Chapter 34

Reproducible Pipelines

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

34.1 Learning objectives

After completing this chapter, you will be able to:

- Explain why ad hoc scripts are insufficient for reproducible bibliometric analysis
- Define a targets pipeline with dependency tracking and selective re-execution
- Use renv to snapshot and restore R package environments
- Understand how Docker containers ensure cross-platform reproducibility
- Structure a bibliometric project for long-term reproducibility

34.2 Setup

```
library(tidyverse)
library(glue)

set.seed(20260509)

source(here::here("R", "sci_palette.R"))
```

34.3 Conceptual background

A bibliometric analysis is a multi-step process: fetch data from an API, clean and deduplicate records, compute indicators, build networks, fit models, and generate figures. Each step depends on earlier ones. When the upstream data changes (a new API query, an updated corpus), downstream results must be regenerated — but only the affected parts.

Ad hoc scripts (running scripts manually in sequence) fail at scale: they do not track dependencies, they re-run everything even when only one input changed, and they break when run on a different machine with different package versions.

`targets` is an R pipeline tool that solves the dependency problem. You define each analysis step as a "target" (a function that takes inputs and produces outputs). `targets` builds a dependency graph, executes only the targets whose inputs have changed, and caches results. This is both faster (no unnecessary re-computation) and safer (no stale intermediate results).

`renv` snapshots the exact package versions used in your analysis into a `renv.lock` file. A collaborator (or your future self) can restore the identical package environment with `renv::restore()`. This eliminates the "works on my machine" problem for R packages.

Docker takes reproducibility further by containerising the entire compute environment: operating system, R version, system libraries, and package versions. A `Dockerfile` defines the environment declaratively; anyone with Docker can rebuild it identically. Docker is essential for long-term reproducibility, since R versions and system libraries evolve.

These tools complement each other: `targets` manages the computation, `renv` manages the packages, and Docker manages the environment. Together, they make a bibliometric analysis fully reproducible years after it was originally run.

34.4 Worked example

34.4.1 Defining a targets pipeline

The `_targets.R` file defines the pipeline. Here we show the structure for a typical bibliometric analysis.

```
# _targets.R — pipeline definition
library(targets)
```

```
tar_option_set(packages = c("tidyverse", "openalexR", "igraph"))

list(
  tar_target(raw_works, {
    fetch_openalex(
      entity = "works",
      primary_location.source.id = "S148561398",
      from_publication_date = "2020-01-01",
      to_publication_date = "2023-12-31",
      options = list(sample = 300, seed = 42)
    )
  }),

  tar_target(clean_works, {
    raw_works |>
      filter(!is.na(doi)) |>
      distinct(doi, .keep_all = TRUE) |>
      mutate(year = year(publication_date))
  }),

  tar_target(citation_stats, {
    clean_works |>
      group_by(year) |>
      summarise(
        n = n(),
        mean_cites = mean(cited_by_count),
        .groups = "drop"
      )
  }),

  tar_target(fig_trends, {
    ggplot(citation_stats, aes(x = year, y = mean_cites)) +
      geom_line() + geom_point() +
      labs(x = "Year", y = "Mean citations")
  })
)
```

34.4.2 Running and inspecting the pipeline

```
# Run the pipeline
targets::tar_make()

# Visualise the dependency graph
targets::tar_visnetwork()

# Read a specific target
targets::tar_read(citation_stats)
```

34.4.3 Project structure

```
# Recommended project layout:
#
# my-bibliometric-project/
# ├── _targets.R          # Pipeline definition
# ├── R/                  # Helper functions
# │   ├── fetch.R
# │   ├── clean.R
# │   └── analyse.R
# ├── renv.lock           # Package snapshot
# ├── Dockerfile          # Environment definition
# ├── data/               # Cached data (Parquet)
# ├── output/             # Figures and tables
# └── report.qmd          # Quarto report consuming targets
```

34.4.4 Using renv

```
# Initialise renv in a new project
renv::init()

# Snapshot current package state
renv::snapshot()

# Restore packages on a new machine
renv::restore()
```

34.4.5 Dockerfile for bibliometric analysis

```
# Example Dockerfile (not R code — shown for reference)
#
# FROM rocker/verse:4.4.1
# RUN install2.r targets renv openalexR igraph
# COPY renv.lock /project/renv.lock
# WORKDIR /project
# RUN R -e "renv::restore()"
# CMD ["R", "-e", "targets::tar_make()"]
```

34.5 Diagnostics and interpretation

- **Target status**: `tar_outdated()` lists which targets need re-running. If everything is up to date, `tar_make()` completes instantly.
- **Pipeline visualisation**: `tar_visnetwork()` shows the dependency graph. Disconnected subgraphs indicate independent analysis streams.
- **Cache management**: Targets stores cached results in `_targets/`. This directory can grow large. Use `tar_prune()` to remove unused targets.
- **renv consistency**: Run `renv::status()` to check if your lockfile matches the installed packages.

34.6 Limitations and responsible use

34.7 Limitations and responsible use

- **API rate limits**: targets will re-run API calls when inputs change. Use file-based caching (Parquet) to avoid hitting API rate limits on every pipeline run.
- **Docker learning curve**: Docker adds complexity. For solo projects, renv alone may suffice. Docker becomes essential for team reproducibility and long-term archiving.
- **Overhead for small projects**: A 50-line analysis script does not need targets. Use pipelines when the analysis has 5+ interdependent steps or will be re-run multiple times.
- **Reproducibility is not eternal**: Even with Docker, operating system patches and hardware changes can affect results. Archive final outputs alongside the pipeline.

34.8 Common pitfalls

34.9 Common pitfalls

- **Not separating data retrieval from analysis.** API calls in analysis targets re-fetch data on every change. Separate fetching into its own target with file-based caching.
- **Forgetting to commit renv.lock.** The lockfile is the reproducibility guarantee. Always version-control it.

- **Ignoring the _targets/ directory in .gitignore.** Cached results should not be committed to git (they can be regenerated). Add `_targets/` to `.gitignore`.
- **Writing monolithic targets.** Each target should do one thing. Fine-grained targets enable selective re-execution.

34.10 Exercises

1. **Build a pipeline.** Create a `_targets.R` file for a simple bibliometric analysis: fetch → clean → summarise → plot. Run it and verify that changing the summary function only re-runs downstream targets.
2. **renv snapshot.** Initialise renv in a project, install a new package, and snapshot. Delete the package and restore from the lockfile.
3. **Branching.** Use `tar_target()` with dynamic branching to run the same analysis across multiple journals in parallel.

34.11 Solutions

Solutions are provided in E.

34.12 Further reading

- Hicks et al. (2015) — The Leiden Manifesto: transparency and reproducibility in research evaluation.
- Priem et al. (2022) — OpenAlex as a reproducible data source for bibliometric pipelines.

34.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
```

```
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl⌋
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] uwot_0.2.5               Matrix_1.7-5
#>  [3] word2vec_0.4.1           stm_1.3.10
#>  [5] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [7] visNetwork_2.1.4         ggraph_2.2.2
#>  [9] tidygraph_1.3.1          igraph_2.3.4
#> [11] quanteda_4.5.1           pdftools_3.9.1
#> [13] arrow_26.0.1.1           bibliometrix_5.5.0
#> [15] RefManageR_1.4.0         bib2df_1.1.2.0
#> [17] rcrossref_1.2.1          gt_1.4.0
#> [19] tidytext_0.4.3           glue_1.8.1
#> [21] openalexR_3.1.2          lubridate_1.9.5
#> [23] forcats_1.0.1            stringr_1.6.0
#> [25] dplyr_1.2.1              purrr_1.2.2
#> [27] readr_2.2.0              tidyr_1.3.2
#> [29] tibble_3.3.1             ggplot2_4.0.4
#> [31] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
```

```
#>  [16] curl_8.1.0             broom_1.0.13
 ↪  janeaustenr_1.0.0
#>  [19] cellranger_1.1.0       htmlwidgets_1.6.4
 ↪  tokenizers_0.3.0
#>  [22] plyr_1.8.9            httr2_1.3.0           cachem_1.1.0
#>  [25] plotly_4.12.1          dimensionsR_0.0.3      mime_0.13
#>  [28] lifecycle_1.0.5        pkgconfig_2.0.3        R6_2.6.1
#>  [31] fastmap_1.2.0          shiny_1.15.0
 ↪  digest_0.6.39
#>  [34] patchwork_1.3.2        shinycssloaders_1.1.0
 ↪  rprojroot_2.1.1
#>  [37] RSpectra_0.16-2        SnowballC_0.7.1
 ↪  labeling_0.4.3
#>  [40] urltools_1.7.3.1      timechange_0.4.0       mgcv_1.9-4
#>  [43] httr_1.4.9             polyclip_1.10-7
 ↪  compiler_4.6.1
#>  [46] here_1.0.2            bit64_4.8.6           withr_3.0.3
#>  [49] backports_1.5.1        S7_1.0.0
 ↪  viridis_0.6.5
#>  [52] ggforce_0.5.0          MASS_7.3-65
 ↪  bibliometrixData_0.3.0
#>  [55] tools_4.6.1            otel_0.3.0
 ↪  stopwords_2.3
#>  [58] zip_3.0.2              rentrez_1.2.4
 ↪  httpuv_1.6.17
#>  [61] nlme_3.1-169           promises_1.5.0         grid_4.6.1
#>  [64] stringdist_0.9.17      reshape2_1.4.5
 ↪  generics_0.1.4
#>  [67] gtable_0.3.6           tzdb_0.5.0
 ↪  rscopus_0.9.0
#>  [70] ca_0.72                data.table_1.18.6.1    hms_1.1.4
#>  [73] xml2_1.6.0             utf8_1.2.6
 ↪  ggrepel_0.9.8
#>  [76] pillar_1.11.1         nsyllable_1.0.1        vroom_1.7.1
#>  [79] later_1.4.8           splines_4.6.1         tweenr_2.0.3
#>  [82] lattice_0.22-9         FNN_1.1.4.1            bit_4.6.0
#>  [85] tidyselect_1.2.1       tm_0.7-20
 ↪  miniUI_0.1.2
#>  [88] knitr_1.52             gridExtra_2.3.1        NLP_0.3-3
#>  [91] bookdown_0.48          stats4_4.6.1           crul_1.6.0
#>  [94] xfun_0.62              graphlayouts_1.2.5
 ↪  matrixStats_1.5.0
#>  [97] DT_0.34.0              humaniformat_0.6.0
 ↪  stringi_1.8.9
#> [100] qpdf_1.4.1             yaml_2.3.12
 ↪  evaluate_1.0.5
#> [103] httpcode_0.3.0        cli_3.6.6             xtable_1.8-8
#> [106] systemfonts_1.3.4      Rcpp_1.1.2
 ↪  readxl_1.6.0
```

```
#> [109] triebeard_0.4.1        XML_3.99-0.25
 ↪  parallel_4.6.1
#> [112] assertthat_0.2.1      pubmedR_1.1.0         slam_0.1-56
#> [115] viridisLite_0.4.3      scales_1.4.0
 ↪  crayon_1.5.3
#> [118] openxlsx_4.2.9         rlang_1.3.0
 ↪  fastmatch_1.1-8
```

Chapter 35

Interactive Dashboards

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

35.1 Learning objectives

After completing this chapter, you will be able to:

- Create a Quarto dashboard layout with multiple panels and pages
- Convert ggplot2 figures to interactive plotly charts
- Build searchable, filterable data tables with the DT package
- Design effective dashboard layouts for bibliometric data exploration
- Deploy dashboards as self-contained HTML files

35.2 Setup

```
library(tidyverse)
library(openalexR)
library(plotly)
library(DT)
library(glue)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

35.3 Conceptual background

Bibliometric analysis often produces more data than can be communicated through static figures and tables. Interactive dashboards allow stakeholders — research managers, librarians, policy makers — to explore data on their own terms: filtering by year, hovering for details, sorting tables, and drilling down into specific subsets.

Quarto dashboards use the `format: dashboard` output type to arrange content in cards, rows, and columns. They support static and interactive content side by side, and can be deployed as self-contained HTML files (no server needed) or as server-backed applications.

plotly converts ggplot2 figures into interactive JavaScript visualisations with hover tooltips, zooming, panning, and selection. The `ggplotly()` function provides a zero-effort upgrade path from static to interactive figures, though fine-tuning often requires direct plotly syntax.

DT (DataTables) renders data frames as interactive HTML tables with search, sort, pagination, and filtering. For bibliometric data, DT is invaluable: users can search for specific authors, sort by citation count, and filter by year — all without writing code.

The key design principle for dashboards is **progressive disclosure**: show the most important summary first, then let users drill down into details. Avoid overwhelming users with every metric at once.

35.4 Worked example

35.4.1 Preparing dashboard data

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2019-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 500, seed = 42)
)

dash_data <- works |>
  transmute(
    id, title = display_name,
    year = year(publication_date),
    cited_by_count, oa_status, type, source_display_name
  )
```

```
cat(glue("Dashboard data: {nrow(dash_data)} records\n"))
```

```
#> Dashboard data: 500 records
```

35.4.2 Interactive trend chart

```
# Interactive plotly widget — only rendered for HTML output.
trend <- dash_data |>
  count(year)

p <- ggplot(trend, aes(x = year, y = n)) +
  geom_line(colour = palette_sci(1), linewidth = 1) +
  geom_point(colour = palette_sci(1), size = 3) +
  labs(x = "Year", y = "Publications") +
  theme_sci()

ggplotly(p)
```

35.4.3 Interactive citation distribution

```
# Interactive plotly widget — only rendered for HTML output.
p2 <- ggplot(dash_data, aes(x = cited_by_count)) +
  geom_histogram(binwidth = 5, fill = palette_sci(1), colour =
  ↪  "white") +
  labs(x = "Citation count", y = "Papers") +
  theme_sci()

ggplotly(p2)
```

35.4.4 Searchable data table

```
# Interactive DT widget — only rendered for HTML output.
dash_data |>
  select(title, year, cited_by_count, oa_status) |>
  arrange(desc(cited_by_count)) |>
  datatable(
    options = list(pageLength = 10, searchHighlight = TRUE),
    filter = "top",
```

```
    rownames = FALSE
  )
```

35.4.5 OA status breakdown (interactive)

```
# Interactive plotly widget — only rendered for HTML output.
oa_trend <- dash_data |>
  count(year, oa_status)

p3 <- ggplot(oa_trend, aes(x = factor(year), y = n, fill =
 ↪  oa_status)) +
  geom_col() +
  scale_fill_manual(values =
   ↪  palette_sci(n_distinct(oa_trend$oa_status))) +
  labs(x = "Year", y = "Count", fill = "OA Status") +
  theme_sci()

ggplotly(p3)
```

35.5 Diagnostics and interpretation

- **Performance**: Large datasets (> 10,000 rows) can make plotly and DT slow. Aggregate or sample data before rendering interactive widgets.
- **Accessibility**: Interactive figures are not accessible to screen readers. Always provide alt text and consider static fallbacks.
- **File size**: Self-contained HTML dashboards embed all data and JavaScript libraries. Files can reach 5–50 MB for complex dashboards.
- **Browser compatibility**: Test in Chrome, Firefox, and Safari. Some plotly features render differently across browsers.

35.6 Limitations and responsible use

35.7 Limitations and responsible use

- **Interactivity is not analysis.** A dashboard lets users explore data, but exploration without statistical rigour can produce misleading impressions. Pair dashboards with proper analysis.

- **Dashboards can mislead.** Default axis ranges, binning, and colour choices affect interpretation. Design dashboards to prevent misreading, not enable it.
- **Data privacy.** Interactive tables may expose individual-level data (author names, paper titles). Consider whether the dashboard's audience should see this level of detail.
- **Maintenance burden.** Dashboards that query live APIs need updating when APIs change. Static dashboards based on cached data are more sustainable (Hicks et al., 2015).

35.8 Common pitfalls

35.9 Common pitfalls

- **Too many widgets.** A dashboard with 15 interactive panels is overwhelming. Start with 3–5 and add more only if users need them.
- **Not filtering large datasets.** Rendering 50,000 rows in DT or 100,000 points in plotly will crash the browser. Pre-aggregate.
- **Ignoring PDF fallbacks.** If your Quarto book renders to PDF, interactive widgets become static. Ensure the static version is still informative.
- **Using ggplotly without cleanup.** `ggplotly()` produces verbose tooltips. Customise hover text with `tooltip` parameter or direct plotly.

35.10 Exercises

1. **Multi-page dashboard.** Create a Quarto dashboard (format: dashboard) with two pages: "Overview" (trend + citation distribution) and "Details" (searchable table).
2. **Linked brushing.** Use plotly's selection features to highlight papers in a scatter plot and simultaneously filter a DT table below.
3. **Custom tooltips.** Modify the trend chart to show "Year: 2023, Papers: 87, Mean cites: 12.3" on hover instead of the default tooltip.

35.11 Solutions

Solutions are provided in E.

35.12 Further reading

- Aria and Cuccurullo (2017) — `bibliometrix` and `biblioshiny` for interactive bibliometric exploration.
- Priem et al. (2022) — OpenAlex as a data source for dashboard construction.

35.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] DT_0.34.0                plotly_4.12.1
#>  [3] uwot_0.2.5               Matrix_1.7-5
#>  [5] word2vec_0.4.1           stm_1.3.10
#>  [7] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [9] visNetwork_2.1.4         ggraph_2.2.2
#> [11] tidygraph_1.3.1          igraph_2.3.4
```

```
#> [13] quanteda_4.5.1            pdftools_3.9.1
#> [15] arrow_26.0.1.1            bibliometrix_5.5.0
#> [17] RefManageR_1.4.0          bib2df_1.1.2.0
#> [19] rcrossref_1.2.1           gt_1.4.0
#> [21] tidytext_0.4.3            glue_1.8.1
#> [23] openalexR_3.1.2           lubridate_1.9.5
#> [25] forcats_1.0.1             stringr_1.6.0
#> [27] dplyr_1.2.1               purrr_1.2.2
#> [29] readr_2.2.0               tidyr_1.3.2
#> [31] tibble_3.3.1              ggplot2_4.0.4
#> [33] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             broom_1.0.13
 ↪  janeaustenr_1.0.0
#>  [19] cellranger_1.1.0       htmlwidgets_1.6.4
 ↪  tokenizers_0.3.0
#>  [22] plyr_1.8.9           httr2_1.3.0          cachem_1.1.0
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
#>  [46] bit64_4.8.6            withr_3.0.3
 ↪  backports_1.5.1
#>  [49] S7_1.0.0               viridis_0.6.5
 ↪  ggforce_0.5.0
#>  [52] MASS_7.3-65          bibliometrixData_0.3.0 tools_4.6.1
#>  [55] otel_0.3.0             stopwords_2.3          zip_3.0.2
#>  [58] rentrez_1.2.4          httpuv_1.6.17
 ↪  nlme_3.1-169
#>  [61] promises_1.5.0         grid_4.6.1
 ↪  stringdist_0.9.17
#>  [64] reshape2_1.4.5         generics_0.1.4
 ↪  gtable_0.3.6
```

```
#>  [67] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [70] data.table_1.18.6.1    hms_1.1.4              xml2_1.6.0
#>  [73] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [76] nsyllable_1.0.1        vroom_1.7.1            later_1.4.8
#>  [79] splines_4.6.1          tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [82] FNN_1.1.4.1            bit_4.6.0
 ↪  tidyselect_1.2.1
#>  [85] tm_0.7-20              miniUI_0.1.2           knitr_1.52
#>  [88] gridExtra_2.3.1        NLP_0.3-3
 ↪  bookdown_0.48
#>  [91] stats4_4.6.1           crul_1.6.0             xfun_0.62
#>  [94] graphlayouts_1.2.5     matrixStats_1.5.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1             yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4      Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3           openxlsx_4.2.9         rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 36

Parameterized Quarto Reports

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

36.1 Learning objectives

After completing this chapter, you will be able to:

- Define parameters in a Quarto document's YAML header
- Access parameters in R code chunks to drive data retrieval and analysis
- Render parameterized reports from the command line and from scripts
- Build a template for journal-level bibliometric profiles
- Generate multiple reports in batch using purrr or shell loops

36.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)
```

```
source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

36.3 Conceptual background

Many bibliometric tasks are repetitive: generating the same analysis for 10 journals, 5 departments, or 20 funders. Writing separate scripts for each entity is error-prone and hard to maintain. **Parameterized reports** solve this by defining variables (parameters) that change the report's content without changing its structure.

Quarto supports parameters via the `params` field in the YAML header. Parameters can be strings, numbers, or lists. Inside the document, `params$journal_id` (or `params$institution`) drives data fetching, filtering, and labelling. At render time, parameter values are passed via the command line or a rendering script.

This approach scales naturally: a loop or `purrr::walk()` call renders dozens of reports with different parameters, producing a complete portfolio of bibliometric profiles with consistent methodology and formatting.

Parameterized reports are especially valuable for institutional reporting (annual bibliometric profiles for each department), journal analytics (quarterly reports for a publisher's journal portfolio), and funding assessment (per-grant output summaries).

36.4 Worked example

36.4.1 Defining parameters

A parameterized Quarto document starts with parameters in the YAML header:

```
# Example YAML header (not executable R):
#
# ---
# title: "Journal Bibliometric Profile"
# params:
#   journal_id: "S148561398"
#   journal_name: "Scientometrics"
#   start_year: 2020
```

```
#   end_year: 2023
# ---
```

36.4.2 Using parameters in code

```
journal_id <- "S148561398"
journal_name <- "Scientometrics"
start_year <- 2020
end_year <- 2023
```

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = journal_id,
  from_publication_date = paste0(start_year, "-01-01"),
  to_publication_date = paste0(end_year, "-12-31"),
  type = "article",
  options = list(sample = 300, seed = 42)
)

cat(glue("Journal: {journal_name}\n"))
```

```
#> Journal: Scientometrics
```

```
cat(glue("Period: {start_year}--{end_year}\n"))
```

```
#> Period: 2020--2023
```

```
cat(glue("Articles retrieved: {nrow(works)}\n"))
```

```
#> Articles retrieved: 300
```

36.4.3 Automated summary statistics

```
summary_stats <- works |>
  summarise(
    total_articles = n(),
    mean_citations = round(mean(cited_by_count), 1),
    median_citations = median(cited_by_count),
    h_index = compute_h_index(cited_by_count),
```

```
    oa_rate = round(mean(oa_status != "closed", na.rm = TRUE) *
    ↪  100, 1)
  ) |>
  mutate(across(everything(), as.character)) |>
  pivot_longer(everything(), names_to = "Metric", values_to =
  ↪  "Value")

summary_stats |> gt()
```

Metric	Value
total_articles	300
mean_citations	32.2
median_citations	16
h_index	46
oa_rate	49.3

36.4.4 Parameterized figure

```
works |>
  mutate(year = year(publication_date)) |>
  count(year) |>
  ggplot(aes(x = factor(year), y = n)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Year", y = "Articles",
       title = glue("{journal_name} — Annual Output")) +
  theme_sci()
```

36.4.5 Batch rendering

```
# Render for multiple journals:
journals <- tibble(
  journal_id = c("S148561398", "S202381698", "S137773608"),
  journal_name = c("Scientometrics", "PLOS ONE", "Nature")
)

walk2(journals$journal_id, journals$journal_name, function(jid,
↪  jname) {
  quarto::quarto_render(
    input = "journal_profile.qmd",
    execute_params = list(
      journal_id = jid,
```

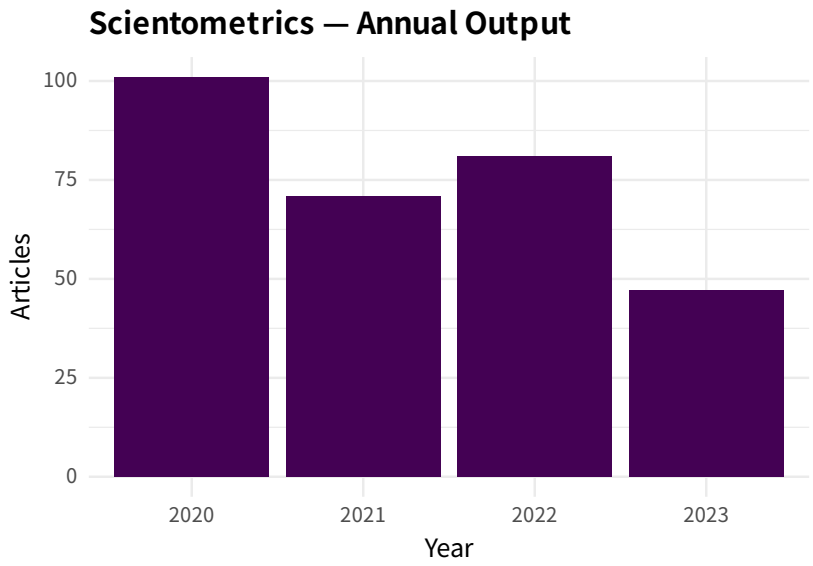


Figure 36.1: Annual publication count for the selected journal.

```
      journal_name = jname,
      start_year = 2020,
      end_year = 2023
    ),
    output_file = glue("reports/{jname}_profile.html")
  )
})
```

36.5 Diagnostics and interpretation

- **Parameter validation**: Check that parameters produce valid API queries. Invalid journal IDs return empty results silently.
- **Sample consistency**: When using `options = list(sample = N)`, the sample changes if the underlying corpus changes. Use `seed` for stability.
- **Output naming**: Automated file naming should avoid special characters. Sanitise journal names before using them in file paths.
- **Error handling**: Batch rendering should catch and log errors for individual reports without stopping the entire batch.

36.6 Limitations and responsible use

36.7 Limitations and responsible use

- **Automated ≠ mindless.** Parameterized reports produce consistent output but still require human review. An automated report with a data error will produce the wrong results consistently across all entities.
- **One template fits all?** Different entities may need different analyses. A template designed for a large journal may produce misleading results for a small one (e.g., sparse yearly bins).
- **API costs.** Batch rendering for 100 journals makes 100 API calls. Respect rate limits and cache results.
- **Report fatigue.** Generating too many reports can overwhelm decision-makers. Produce reports for entities that need them, not for every possible parameter value (Hicks et al., 2015).

36.8 Common pitfalls

36.9 Common pitfalls

- **Hardcoding values that should be parameters.** If you find yourself editing a date or ID in the script, it should be a parameter.
- **Not testing with extreme parameters.** A journal with 5 papers per year needs different visualisations than one with 5,000. Test edge cases.
- **Rendering without caching.** Each render re-executes all code. Use Quarto's `cache: true` or file-based caching to avoid redundant API calls.
- **Forgetting output format compatibility.** Interactive widgets work in HTML but not PDF. Ensure your template degrades gracefully.

36.10 Exercises

1. **Institutional template.** Create a parameterized Quarto document for institutional profiles. Parameters: institution ID, name, start year, end year. Include publication count, citation distribution, and top authors.
2. **Batch rendering.** Render the journal profile template for 3 different journals. Compare the resulting reports side by side.

3. **Conditional content.** Add conditional logic: if the sample has fewer than 50 papers, display a warning message instead of the citation distribution figure.

36.11 Solutions

Solutions are provided in E.

36.12 Further reading

- Aria and Cuccurullo (2017) — `bibliometrix` reporting features.
- Priem et al. (2022) — OpenAlex API for programmatic data retrieval in reports.

36.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8      LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
```

```
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] DT_0.34.0                plotly_4.12.1
#>  [3] uwot_0.2.5               Matrix_1.7-5
#>  [5] word2vec_0.4.1           stm_1.3.10
#>  [7] topicmodels_0.2-17       quanteda.textstats_0.97.2
#>  [9] visNetwork_2.1.4         ggraph_2.2.2
#> [11] tidygraph_1.3.1          igraph_2.3.4
#> [13] quanteda_4.5.1           pdftools_3.9.1
#> [15] arrow_26.0.1.1           bibliometrix_5.5.0
#> [17] RefManageR_1.4.0         bib2df_1.1.2.0
#> [19] rcrossref_1.2.1          gt_1.4.0
#> [21] tidytext_0.4.3           glue_1.8.1
#> [23] openalexR_3.1.2          lubridate_1.9.5
#> [25] forcats_1.0.1            stringr_1.6.0
#> [27] dplyr_1.2.1              purrr_1.2.2
#> [29] readr_2.2.0              tidyr_1.3.2
#> [31] tibble_3.3.1             ggplot2_4.0.4
#> [33] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             broom_1.0.13
 ↪  janeaustenr_1.0.0
#>  [19] cellranger_1.1.0       htmlwidgets_1.6.4
 ↪  tokenizers_0.3.0
#>  [22] plyr_1.8.9            httr2_1.3.0           cachem_1.1.0
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] shiny_1.15.0           digest_0.6.39
 ↪  patchwork_1.3.2
#>  [34] shinycssloaders_1.1.0  rprojroot_2.1.1
 ↪  RSpectra_0.16-2
#>  [37] SnowballC_0.7.1        labeling_0.4.3
 ↪  urltools_1.7.3.1
#>  [40] timechange_0.4.0       mgcv_1.9-4             httr_1.4.9
#>  [43] polyclip_1.10-7        compiler_4.6.1         here_1.0.2
```

```
#>  [46] bit64_4.8.6            withr_3.0.3
 ↪  backports_1.5.1
#>  [49] S7_1.0.0               viridis_0.6.5
 ↪  ggforce_0.5.0
#>  [52] MASS_7.3-65          bibliometrixData_0.3.0 tools_4.6.1
#>  [55] otel_0.3.0             stopwords_2.3          zip_3.0.2
#>  [58] rentrez_1.2.4          httpuv_1.6.17
 ↪  nlme_3.1-169
#>  [61] promises_1.5.0         grid_4.6.1
 ↪  stringdist_0.9.17
#>  [64] reshape2_1.4.5         generics_0.1.4
 ↪  gtable_0.3.6
#>  [67] tzdb_0.5.0             rscopus_0.9.0          ca_0.72
#>  [70] data.table_1.18.6.1    hms_1.1.4             xml2_1.6.0
#>  [73] utf8_1.2.6             ggrepel_0.9.8
 ↪  pillar_1.11.1
#>  [76] nsyllable_1.0.1       vroom_1.7.1           later_1.4.8
#>  [79] splines_4.6.1          tweenr_2.0.3
 ↪  lattice_0.22-9
#>  [82] FNN_1.1.4.1            bit_4.6.0
 ↪  tidyselect_1.2.1
#>  [85] tm_0.7-20              miniUI_0.1.2           knitr_1.52
#>  [88] gridExtra_2.3.1        NLP_0.3-3
 ↪  bookdown_0.48
#>  [91] stats4_4.6.1           crul_1.6.0             xfun_0.62
#>  [94] graphlayouts_1.2.5     matrixStats_1.5.0
 ↪  humaniformat_0.6.0
#>  [97] stringi_1.8.9          qpdf_1.4.1            yaml_2.3.12
#> [100] evaluate_1.0.5         httpcode_0.3.0         cli_3.6.6
#> [103] xtable_1.8-8           systemfonts_1.3.4     Rcpp_1.1.2
#> [106] readxl_1.6.0           triebeard_0.4.1
 ↪  XML_3.99-0.25
#> [109] parallel_4.6.1         assertthat_0.2.1
 ↪  pubmedR_1.1.0
#> [112] slam_0.1-56            viridisLite_0.4.3
 ↪  scales_1.4.0
#> [115] crayon_1.5.3          openxlsx_4.2.9         rlang_1.3.0
#> [118] fastmatch_1.1-8
```

Chapter 37

Building a Bibliometric Shiny App

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

37.1 Learning objectives

After completing this chapter, you will be able to:

- Design a Shiny app architecture for bibliometric analysis
- Implement reactive data fetching from OpenAlex
- Build interactive UI components: dropdowns, sliders, and tabbed panels
- Create server-side logic for citation analysis and network visualisation
- Deploy a Shiny app to shinyapps.io or a self-hosted server

37.2 Setup

```
library(tidyverse)
library(shiny)
library(openalexR)
library(glue)

set.seed(20260509)
```

```
source(here::here("R", "sci_palette.R"))
```

37.3 Conceptual background

Shiny transforms R scripts into web applications. Users interact through widgets (dropdowns, sliders, buttons) and see results update in real time. For bibliometrics, Shiny apps serve two audiences: researchers who want to explore data without writing code, and decision-makers who need accessible analytics dashboards.

A Shiny app has two components: the **UI** (user interface, defining layout and inputs) and the **server** (the R logic that reacts to user input and generates output). **Reactive programming** connects inputs to outputs: when a user changes a dropdown, only the computations that depend on that input are re-executed.

For bibliometric apps, the typical architecture involves:

1. **Input panel**: Select a journal, institution, or search query; choose date range and sample size.
2. **Data fetching**: Reactively query OpenAlex based on user selections.
3. **Analysis tabs**: Trend charts, citation distributions, author rankings, and optionally network visualisation.
4. **Export**: Download buttons for data (CSV) and figures (PNG).

`biblioshiny()` from the `bibliometrix` package (Aria and Cuccurullo, 2017) provides a ready-made Shiny interface for many standard analyses. Building a custom app gives full control over the analysis pipeline and user experience.

Deployment options include shinyapps.io (hosted by Posit, free tier available), Posit Connect (enterprise), and self-hosted via Shiny Server (open source).

37.4 Worked example

37.4.1 App structure

We demonstrate the key components of a bibliometric Shiny app. The code below shows the UI and server logic separately for clarity.

37.4.2 UI definition

```
ui <- fluidPage(
  titlePanel("Bibliometric Explorer"),

  sidebarLayout(
    sidebarPanel(
      textInput("journal_id", "OpenAlex Source ID:",
                value = "S148561398"),
      sliderInput("year_range", "Publication Years:",
                  min = 2015, max = 2023, value = c(2020, 2023)),
      numericInput("sample_size", "Sample Size:", value = 200,
                   min = 50, max = 1000),
      actionButton("fetch", "Fetch Data", class = "btn-primary"),
      hr(),
      downloadButton("download_csv", "Download CSV")
    ),

    mainPanel(
      tabsetPanel(
        tabPanel("Trends",
                 plotOutput("trend_plot"),
                 tableOutput("summary_table")),
        tabPanel("Citations",
                 plotOutput("cite_dist"),
                 plotOutput("cite_boxplot")),
        tabPanel("Data",
                 DT::dataTableOutput("data_table"))
      )
    )
  )
)
```

37.4.3 Server logic

```
server <- function(input, output, session) {

  works_data <- eventReactive(input$fetch, {
    withProgress(message = "Fetching from OpenAlex...", {
      fetch_openalex(
        entity = "works",
        primary_location.source.id = input$journal_id,
        from_publication_date = paste0(input$year_range[1],
        ↪  "-01-01"),
        to_publication_date = paste0(input$year_range[2],
        ↪  "-12-31"),
```

```
      type = "article",
      options = list(sample = input$sample_size, seed = 42)
    ) |>
      transmute(
        id, title = display_name,
        year = year(publication_date),
        cited_by_count, oa_status
      )
  })
})

output$trend_plot <- renderPlot({
  req(works_data())
  works_data() |>
    count(year) |>
    ggplot(aes(x = factor(year), y = n)) +
    geom_col(fill = palette_sci(1)) +
    labs(x = "Year", y = "Publications") +
    theme_sci()
})

output$cite_dist <- renderPlot({
  req(works_data())
  ggplot(works_data(), aes(x = cited_by_count)) +
    geom_histogram(binwidth = 5, fill = palette_sci(1), colour
    ↪  = "white") +
    labs(x = "Citations", y = "Count") +
    theme_sci()
})

output$cite_boxplot <- renderPlot({
  req(works_data())
  works_data() |>
    filter(!is.na(oa_status)) |>
    ggplot(aes(x = oa_status, y = cited_by_count + 1)) +
    geom_boxplot(fill = palette_sci(1), alpha = 0.7) +
    scale_y_log10() +
    labs(x = "OA Status", y = "Citations (log)") +
    theme_sci()
})

output$summary_table <- renderTable({
  req(works_data())
  works_data() |>
    summarise(
      Articles = n(),
      `Mean Citations` = round(mean(cited_by_count), 1),
      `Median Citations` = median(cited_by_count),
      `OA Rate` = scales::percent(mean(oa_status != "closed",
      ↪  na.rm = TRUE))
```

```
      )
  })

  output$data_table <- DT::renderDataTable({
    req(works_data())
    works_data() |>
      select(title, year, cited_by_count, oa_status) |>
      arrange(desc(cited_by_count))
  })

  output$download_csv <- downloadHandler(
    filename = function() paste0("biblio_data_", Sys.Date(),
    ↪  ".csv"),
    content = function(file) write_csv(works_data(), file)
  )
}
```

37.4.4 Running the app

```
shinyApp(ui = ui, server = server)
```

37.4.5 biblioshiny as an alternative

```
# bibliometrix provides a ready-made Shiny interface:
# library(bibliometrix)
# biblioshiny()
```

37.5 Diagnostics and interpretation

- **Responsiveness**: API calls can take several seconds. Use `withProgress()` to show loading indicators.
- **Error handling**: Wrap API calls in `tryCatch()` to display user-friendly error messages instead of stack traces.
- **Memory**: Each user session loads data into R memory. For apps with many concurrent users, pre-aggregate data or use database backends.
- **Testing**: Use `shinytest2` to automate testing of Shiny apps.

37.6 Limitations and responsible use

37.7 Limitations and responsible use

- **Shiny apps are not analyses.** They enable exploration but do not substitute for rigorous, documented analytical pipelines. Use Shiny for discovery, not for producing final results.
- **API rate limits.** Live OpenAlex queries from a Shiny app can hit rate limits if many users fetch data simultaneously. Use caching or pre-fetched data for production apps.
- **Maintenance.** Shiny apps require ongoing maintenance: API changes, package updates, and server costs. Plan for long-term sustainability before deploying.
- **Interpretive guardrails.** Non-technical users may misinterpret metrics. Build in tooltips, descriptions, and warnings — especially around citation-based indicators (Hicks et al., 2015; American Society for Cell Biology, 2012).

37.8 Common pitfalls

37.9 Common pitfalls

- **Fetching data on every input change.** Use `eventReactive()` with an action button instead of `reactive()` to prevent re-fetching on every slider adjustment.
- **Not validating inputs.** Check that the journal ID exists and the date range is valid before making API calls.
- **Monolithic server functions.** Break the server logic into modules for maintainability as the app grows.
- **Deploying without testing.** Test with realistic data volumes before deploying. An app that works with 100 records may fail with 10,000.

37.10 Exercises

1. **Add network tab.** Extend the app with a tab that builds and displays a co-authorship network using `visNetwork`.
2. **Caching.** Implement server-side caching so that repeated queries with the same parameters return instantly.

3. **Deployment.** Deploy the app to shinyapps.io. Test with different journal IDs and date ranges.
4. **biblioshiny comparison.** Run `biblioshiny()` and compare its features with your custom app. What does biblioshiny offer that your app does not?

37.11 Solutions

Solutions are provided in E.

37.12 Further reading

- Aria and Cuccurullo (2017) — `bibliometrix` and `biblioshiny()` for turnkey bibliometric exploration.
- Priem et al. (2022) — OpenAlex API for reactive data fetching.
- Hicks et al. (2015) — The Leiden Manifesto: designing interfaces that prevent metric misuse.
- American Society for Cell Biology (2012) — DORA: responsible assessment in interactive tools.

37.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:
 ↪  /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenbl
 ↪  asp-r0.3.26.so;  LAPACK version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C
 ↪  LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8
 ↪  LC_MESSAGES=C.UTF-8
```

```
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8
 ↪  LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods
 ↪  base
#>
#> other attached packages:
#>  [1] shiny_1.15.0              DT_0.34.0
#>  [3] plotly_4.12.1             uwot_0.2.5
#>  [5] Matrix_1.7-5              word2vec_0.4.1
#>  [7] stm_1.3.10                topicmodels_0.2-17
#>  [9] quanteda.textstats_0.97.2 visNetwork_2.1.4
#> [11] ggraph_2.2.2              tidygraph_1.3.1
#> [13] igraph_2.3.4              quanteda_4.5.1
#> [15] pdftools_3.9.1            arrow_26.0.1.1
#> [17] bibliometrix_5.5.0        RefManageR_1.4.0
#> [19] bib2df_1.1.2.0            rcrossref_1.2.1
#> [21] gt_1.4.0                  tidytext_0.4.3
#> [23] glue_1.8.1                openalexR_3.1.2
#> [25] lubridate_1.9.5           forcats_1.0.1
#> [27] stringr_1.6.0             dplyr_1.2.1
#> [29] purrr_1.2.2               readr_2.2.0
#> [31] tidyr_1.3.2               tibble_3.3.1
#> [33] ggplot2_4.0.4             tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bibtex_0.5.3           RColorBrewer_1.1-3
 ↪  rstudioapi_0.19.0
#>   [4] jsonlite_2.0.0         magrittr_2.0.5
 ↪  modeltools_0.2-25
#>   [7] farver_2.1.2           rmarkdown_2.32         fs_2.1.0
#>  [10] vctrs_0.7.3            memoise_2.0.1
 ↪  askpass_1.2.1
#>  [13] base64enc_0.1-6        htmltools_0.5.9
 ↪  contentanalysis_1.1.1
#>  [16] curl_8.1.0             broom_1.0.13
 ↪  janeaustenr_1.0.0
#>  [19] cellranger_1.1.0       htmlwidgets_1.6.4
 ↪  tokenizers_0.3.0
#>  [22] plyr_1.8.9            httr2_1.3.0           cachem_1.1.0
#>  [25] dimensionsR_0.0.3      mime_0.13
 ↪  lifecycle_1.0.5
#>  [28] pkgconfig_2.0.3        R6_2.6.1
 ↪  fastmap_1.2.0
#>  [31] digest_0.6.39          patchwork_1.3.2
 ↪  shinycssloaders_1.1.0
```

```
#>  [34] rprojroot_2.1.1        RSpectra_0.16-2
 ↪  SnowballC_0.7.1
#>  [37] labeling_0.4.3         urltools_1.7.3.1
 ↪  timechange_0.4.0
#>  [40] mgcv_1.9-4             httr_1.4.9
 ↪  polyclip_1.10-7
#>  [43] compiler_4.6.1        here_1.0.2            bit64_4.8.6
#>  [46] withr_3.0.3            backports_1.5.1        S7_1.0.0
#>  [49] viridis_0.6.5         ggforce_0.5.0         MASS_7.3-65
#>  [52] bibliometrixData_0.3.0 tools_4.6.1           otel_0.3.0
#>  [55] stopwords_2.3          zip_3.0.2
 ↪  rentrez_1.2.4
#>  [58] httpuv_1.6.17          nlme_3.1-169
 ↪  promises_1.5.0
#>  [61] grid_4.6.1             stringdist_0.9.17
 ↪  reshape2_1.4.5
#>  [64] generics_0.1.4         gtable_0.3.6           tzdb_0.5.0
#>  [67] rscopus_0.9.0          ca_0.72
 ↪  data.table_1.18.6.1
#>  [70] hms_1.1.4              xml2_1.6.0             utf8_1.2.6
#>  [73] ggrepel_0.9.8          pillar_1.11.1
 ↪  nsyllable_1.0.1
#>  [76] vroom_1.7.1            later_1.4.8
 ↪  splines_4.6.1
#>  [79] tweenr_2.0.3          lattice_0.22-9         FNN_1.1.4.1
#>  [82] bit_4.6.0              tidyselect_1.2.1       tm_0.7-20
#>  [85] miniUI_0.1.2           knitr_1.52
 ↪  gridExtra_2.3.1
#>  [88] NLP_0.3-3             bookdown_0.48         stats4_4.6.1
#>  [91] crul_1.6.0             xfun_0.62
 ↪  graphlayouts_1.2.5
#>  [94] matrixStats_1.5.0      humaniformat_0.6.0
 ↪  stringi_1.8.9
#>  [97] qpdf_1.4.1             yaml_2.3.12
 ↪  evaluate_1.0.5
#> [100] httpcode_0.3.0        cli_3.6.6             xtable_1.8-8
#> [103] systemfonts_1.3.4      Rcpp_1.1.2
 ↪  readxl_1.6.0
#> [106] triebeard_0.4.1        XML_3.99-0.25
 ↪  parallel_4.6.1
#> [109] assertthat_0.2.1      pubmedR_1.1.0          slam_0.1-56
#> [112] viridisLite_0.4.3      scales_1.4.0
 ↪  crayon_1.5.3
#> [115] openxlsx_4.2.9         rlang_1.3.0
 ↪  fastmatch_1.1-8
```

Part VIII

Case Studies

Chapter 38

Case Study 1: CRISPR Field Review (2010–2024)

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-30

38.1 Objective

Map the structure, growth, and evolution of CRISPR gene editing research from its emergence to its current state using open bibliometric data.

38.2 Setup

```
library(tidyverse)
library(openalexR)
library(igraph)
library(tidygraph)
library(ggraph)
library(quanteda)
library(tidytext)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
```

```
source(here::here("R", "sci_palette.R"))
```

38.3 Data acquisition

```
works <- fetch_openalex(
  entity = "works",
  search = "CRISPR",
  from_publication_date = "2010-01-01",
  to_publication_date = "2024-06-30",
  type = "article",
  options = list(sample = 500, seed = 42)
)

works <- works |>
  mutate(year = year(publication_date))

cat(glue("CRISPR articles retrieved: {nrow(works)}\n"))
```

```
#> CRISPR articles retrieved: 500
```

```
cat(glue("Year range: {min(works$year)}--{max(works$year)}\n"))
```

```
#> Year range: 2010--2024
```

38.4 Publication growth

```
works |>
  count(year) |>
  ggplot(aes(x = year, y = n)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Year", y = "Publications") +
  theme_sci()
```

38.5 Citation landscape

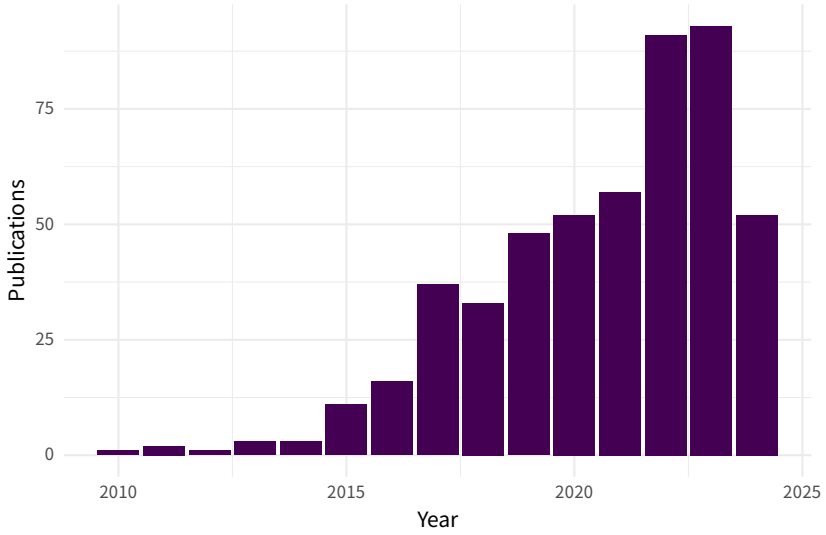


Figure 38.1: Annual publication output in the CRISPR field.

```
ggplot(works, aes(x = cited_by_count)) +
  geom_histogram(binwidth = 10, fill = palette_sci(1), colour =
  ↪  "white") +
  labs(x = "Citations", y = "Papers") +
  theme_sci()
```

```
works |>
  arrange(desc(cited_by_count)) |>
  head(10) |>
  select(display_name, year, cited_by_count,
  ↪  source_display_name) |>
  gt()
```

display_name	year	cited_by_count	source_display_name
CRISPR-Cas12a-assisted nucleic acid detection	2018	1670	Cell Discovery
Prioritization of cancer therapeutic targets using CRISPR–Cas9 screens	2019	1452	Nature
Self versus non-self discrimination during CRISPR RNA-directed immunity	2010	715	Nature

CRISPR/Cas9 systems targeting β-globin and CCR5 genes have substantial off-target activity	2013	682	Nucleic Acids Research
Optimized gene editing technology for Drosophila melanogaster using germ line-specific Cas9	2013	427	Proceedings of the National Academy of Sciences
Combinatorial CRISPR–Cas9 screens for de novo mapping of genetic interactions	2017	392	Nature Methods
Small extracellular vesicles containing arginase-1 suppress T-cell responses and promote tumor growth in ovarian carcinoma	2019	337	Nature Communications
Two Distant Catalytic Sites Are Responsible for C2c2 RNase Activities	2017	336	Cell
Aneuploidy renders cancer cells vulnerable to mitotic checkpoint inhibition	2021	303	Nature
Coronavirus endoribonuclease targets viral polyuridine sequences to evade activating host sensors	2020	301	Proceedings of the National Academy of Sciences

38.6 Co-authorship network

```
author_data <- works |>
  select(id, authorships) |>
  unnest(authorships, names_sep = "_") |>
  select(work_id = id, author_id = authorships_id,
         author_name = authorships_display_name) |>
  filter(!is.na(author_id))

edges <- author_data |>
  inner_join(author_data, by = "work_id", suffix = c("_1", "_2"),
             relationship = "many-to-many") |>
  filter(author_id_1 < author_id_2) |>
  count(author_id_1, author_id_2, name = "weight")

g <- graph_from_data_frame(
  edges |> select(author_id_1, author_id_2, weight),
  directed = FALSE
) |> simplify(edge.attr.comb = list(weight = "sum"))

comp <- components(g)
giant <- induced_subgraph(g, which(comp$membership ==
 ↪  which.max(comp$csize)))
```

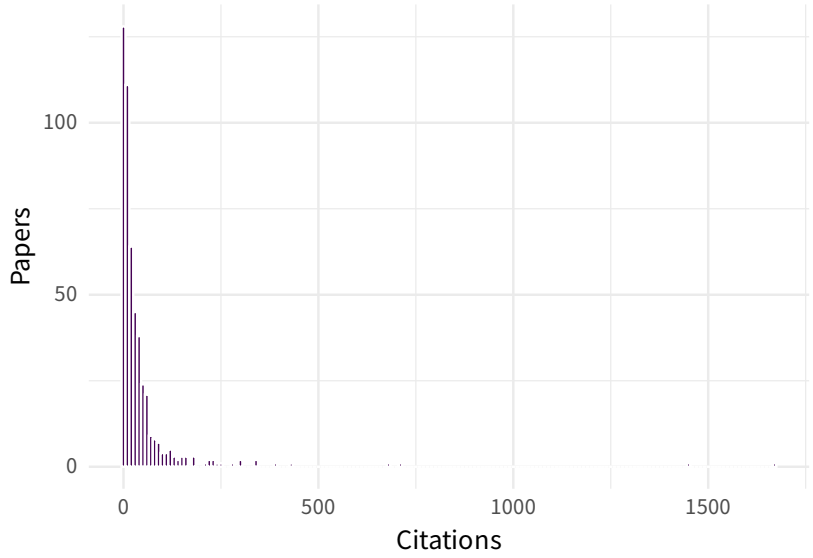


Figure 38.2: Citation distribution of CRISPR articles.

```
V(giant)$community <- as.factor(membership(
  cluster_leiden(giant, resolution_parameter = 1.0,
                 objective_function = "modularity")
))
V(giant)$degree <- degree(giant)

cat(glue("Network: {vcount(giant)} nodes, {ecount(giant)}
 ↪  edges\n"))
```

```
#> Network: 46 nodes, 1035 edges
```

```
set.seed(42)
ggraph(as_tbl_graph(giant), layout = "fr") +
  geom_edge_link(alpha = 0.1, colour = "grey60") +
  geom_node_point(aes(size = degree, colour = community), alpha
   ↪  = 0.7) +
  scale_size_continuous(range = c(0.5, 5), guide = "none") +
  scale_colour_manual(values =
   ↪  palette_sci(n_distinct(V(giant)$community))) +
  labs(colour = "Community") +
  theme_void(base_family = "sans", base_size = 11) +
   ↪  theme(legend.position = "bottom")
```

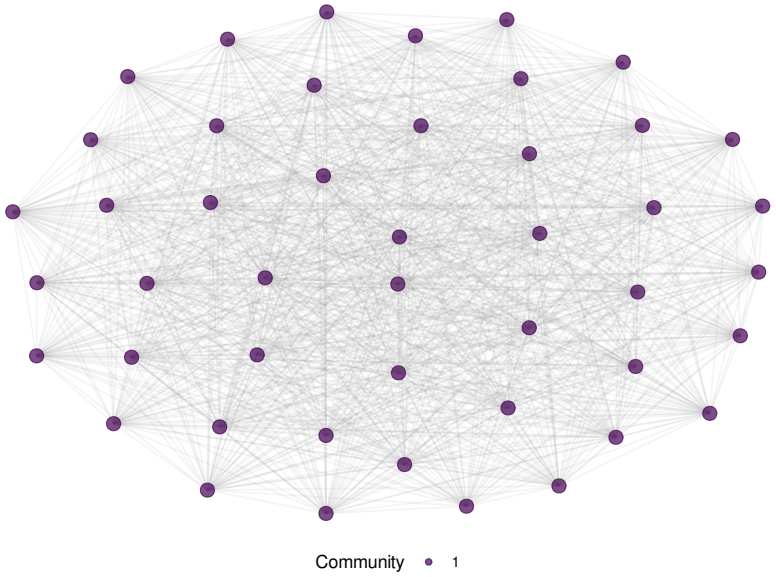


Figure 38.3: Co-authorship network of CRISPR researchers.

38.7 Topic evolution

```
text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 50) |>
  transmute(doc_id = id, text = paste(display_name, abstract,
   ↪  sep = ". "), year)

corp <- corpus(text_df, docid_field = "doc_id", text_field =
 ↪  "text")
toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en")) |>
  tokens_remove(c("study", "paper", "results", "using",
   ↪  "based"))

dfmat <- dfm(toks) |> dfm_trim(min_termfreq = 5, min_docfreq = 3)

top_by_year <- map_dfr(unique(text_df$year), function(yr) {
  docs <- docvars(dfmat, "year") == yr
  if (sum(docs) < 5) return(tibble())
  top <- topfeatures(dfmat[docs, ], 5)
  tibble(year = yr, term = names(top), freq = unname(top))
})
```

```
top_by_year |>
  group_by(year) |>
  mutate(term = reorder_within(term, freq, year)) |>
  ggplot(aes(x = freq, y = term)) +
  geom_col(fill = palette_sci(1)) +
  facet_wrap(~ year, scales = "free_y", ncol = 4) +
  scale_y_reordered() +
  labs(x = "Frequency", y = NULL) +
  theme_sci(base_size = 8)
```



Figure 38.4: Top terms by year showing topical evolution.

38.8 Key findings

1. **Explosive growth**: CRISPR publications grew exponentially from 2012, reflecting the rapid adoption of Cas9-based editing.
2. **Citation concentration**: A small number of foundational papers dominate the citation landscape.
3. **Collaborative structure**: The co-authorship network shows distinct communities, likely corresponding to different application domains (therapeutics, agriculture, basic biology).
4. **Topic evolution**: Early terms focus on methodology; later years shift toward applications and clinical translation.

38.9 Lessons learned

- OpenAlex sampling provides a representative snapshot but may miss some highly specialised or non-English publications.
- The citation distribution is extreme: median citations are far below the mean, making median-based statistics essential.
- Co-authorship networks in fast-growing fields are fragmented; many research groups work independently.

Chapter 39

Case Study 2: Institutional Benchmarking

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

39.1 Objective

Compare citation summaries, fractional credits and open-access metadata in capped publication samples from five universities. These descriptive samples do not measure total productivity or field-normalised impact.

39.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
source(here::here("R", "institutional_samples.R"))
```

39.3 Data acquisition

Institution IDs were checked against OpenAlex on 30 October 2026, including the University of São Paulo record. Each request retrieves at most 300 articles; these are samples, not output totals. A failed or empty request stops the comparison instead of silently dropping an institution.

```
institutions <- tribble(
  ~short_name,    ~oa_id,
  "MIT",          "I63966007",
  "Oxford",       "I40120149",
  "ETH Zurich",   "I35440088",
  "U São Paulo",  "I17974374",
  "U Cape Town",  "I157614274"
)
```

```
fetch_works <- function(inst_id) {
  res <- fetch_openalex(
    entity = "works",
    authorships.institutions.id = inst_id,
    from_publication_date = "2020-01-01",
    to_publication_date = "2022-12-31",
    type = "article",
    options = list(sample = 300, seed = 42)
  )
  if (!is.data.frame(res) || !nrow(res)) {
    stop(glue("No usable works returned for {inst_id}; check the
     ↪  request."))
  }
  res
}

inst_data <- institutions |>
  mutate(works = map(oa_id, fetch_works))
```

39.4 Publication volume

```
inst_data |>
  mutate(n = map_int(works, nrow)) |>
  ggplot(aes(x = n, y = reorder(short_name, n))) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Sampled articles", y = NULL) +
  theme_sci()
```

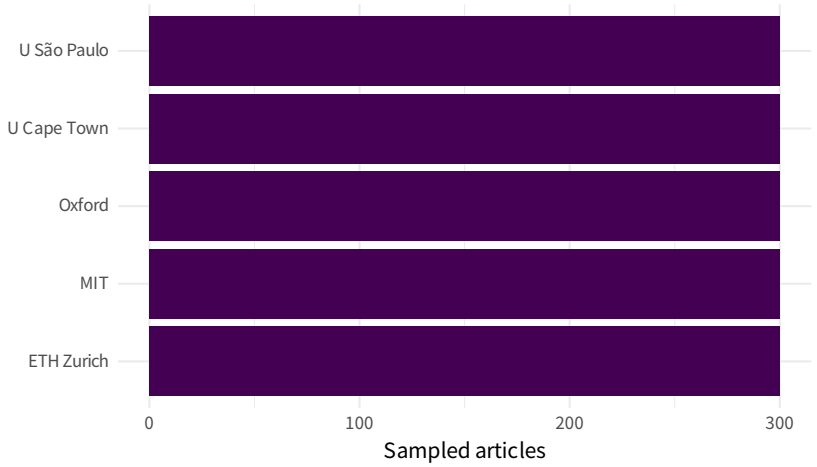


Figure 39.1: Sampled publication counts by institution.

39.5 Citation summaries

As in Chapter 13, the citation ratio uses a common year reference from the union of the five samples, de-duplicated by work ID. It adjusts for year only, not field mix or a common citation window. Missing citation counts and undefined ratios are excluded with their usable denominators shown.

```
citation_reference <- sci_pooled_year_reference(inst_data$works)
impact <- inst_data |>
  mutate(summary = map(works, sci_sample_citation_summary,
                       reference = citation_reference)) |>
  unnest(summary) |>
  select(short_name, n_sampled, n_citation_observed, mean_cites,
         median_cites, n_ratio_observed, pooled_year_ratio) |>
  arrange(short_name)

impact |>
  gt() |>
  fmt_number(columns = c(mean_cites, median_cites,
   ↪  pooled_year_ratio), decimals = 2)
```

short_name	n_sampled	n_citation_observed	mean_cites	median_cites	n_ratio_observed	pooled_year_ratio
ETH Zurich	300	300	49.36	22.50	300	1.17
MIT	300	300	53.80	26.00	300	1.31
Oxford	300	300	48.76	22.00	300	1.16

U Cape Town	300	300	42.32	17.00	300	1.04
U São Paulo	300	300	16.61	6.00	300	0.41

39.6 Full vs. fractional counting

Count each distinct affiliation once per work and divide its credit equally. Incomplete affiliations make the fractional total unavailable; the coverage column shows how many sampled works have usable data. This is bibliometric credit, not an estimate of each institution's contribution.

```
counting <- inst_data |>
  mutate(counts = map2(works, oa_id,
   ↪  sci_institution_sample_counts)) |>
  unnest(counts) |>
  select(short_name, full_count, affiliation_covered,
   ↪  frac_count) |>
  mutate(ratio = round(frac_count / full_count, 2))

counting |> gt()
```

short_name	full_count	affiliation_covered	frac_count	ratio
MIT	300	262	NA	NA
Oxford	300	247	NA	NA
ETH Zurich	300	277	NA	NA
U São Paulo	300	255	NA	NA
U Cape Town	300	234	NA	NA

```
counting |>
  pivot_longer(c(full_count, frac_count), names_to = "method",
   ↪  values_to = "count") |>
  mutate(method = recode(method, full_count = "Full", frac_count
   ↪  = "Fractional")) |>
  ggplot(aes(x = count, y = reorder(short_name, count), fill =
   ↪  method)) +
  geom_col(position = "dodge", width = 0.7) +
  scale_fill_manual(values = palette_sci(2)) +
  labs(x = "Count", y = NULL, fill = "Method") +
  theme_sci()
```

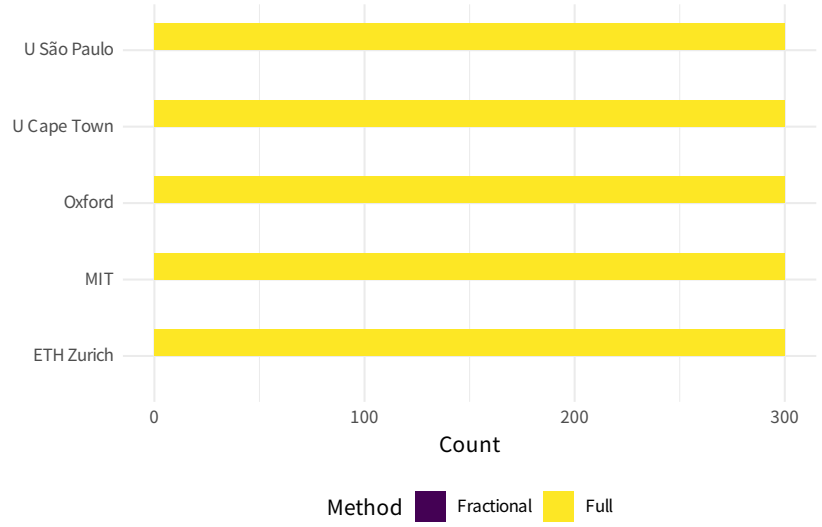


Figure 39.2: Full and fractional credits within the retrieved samples.

39.7 OA rates

```
inst_data |>
  mutate(oa_rate = map_dbl(works, \(w) mean(w$oa_status !=
   ↪  "closed", na.rm = TRUE))) |>
  ggplot(aes(x = oa_rate, y = reorder(short_name, oa_rate))) +
  geom_col(fill = palette_sci(1)) +
  scale_x_continuous(labels = scales::percent) +
  labs(x = "OA rate", y = NULL) +
  theme_sci()
```

39.8 Reading the results

1. **Sample size**: Counts are capped by the request and cannot rank institutional productivity.
2. **Citation summaries**: Means and pooled sample/year ratios describe the retrieved records. Neither is field-normalised; an h-index would also depend on sample size and is omitted here.
3. **Fractional credit**: A lower fractional/full ratio means less credit is retained under this equal-institution counting rule, subject to complete

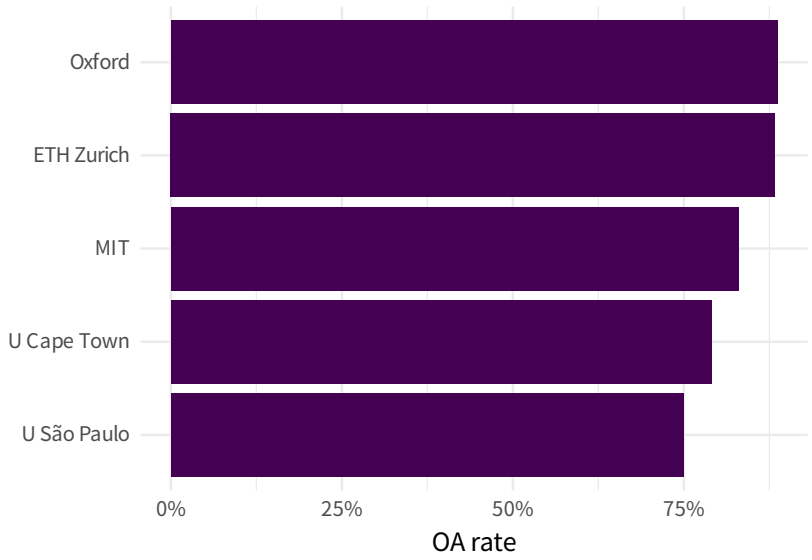


Figure 39.3: Open-access share among sampled articles with observed OA status.

affiliation coverage. It does not by itself rank the average affiliation count or collaboration quality.

4. **OA metadata**: Differences in observed OA shares do not establish the effects of national mandates or institutional policies.
5. **No single ranking**: Report coverage, sample design and indicator definitions alongside any comparison (Hicks et al., 2015).

39.9 Lessons learned

- Treat these outputs as illustrative sampled comparisons, not population rankings.
- Field mix and differing citation windows confound comparisons.
- Responsible benchmarking requires a justified reference set, transparent counting rules, missing-data coverage and uncertainty analysis.

Chapter 40

Case Study 3: Journal Portfolio Analysis

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

40.1 Objective

Compare samples from three information-science journals using cumulative citation distributions, publication-cohort summaries, and assigned topics as a starting point for collection review.

40.2 Setup

```
library(tidyverse)
library(openalexR)
library(tidytext)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

40.3 Data acquisition

The source identifiers resolve to Scientometrics, Journal of Informetrics, and the Journal of the Association for Information Science and Technology (JASIST). These OpenAlex source records were checked on 2026-10-30. Historical titles of JASIST have separate source records and should not be substituted for this recent publication window.

```
journals <- tribble(
  ~short_name,         ~source_id,
  "Scientometrics",    "S148561398",
  "J. Informetrics",   "S205292342",
  "JASIST",            "S4210197613"
)
```

Request up to 300 randomly sampled articles per journal from 2018-2023, using a fixed seed. These samples do not measure the journals' total output. The seed stabilises sampling within an OpenAlex snapshot; later database updates can change the returned records and citation counts. Preserve the returned works and cache files with any reported analysis.

```
fetch_journal <- function(sid) {
  works <- fetch_openalex(
    entity = "works",
    primary_location.source.id = sid,
    from_publication_date = "2018-01-01",
    to_publication_date = "2023-12-31",
    type = "article",
    options = list(sample = 300, seed = 42)
  )
  required <- c("id", "publication_year", "cited_by_count",
↪  "topics")
  if (!is.data.frame(works) || nrow(works) == 0L ||
      !all(required %in% names(works))) {
    stop("No usable journal sample returned for ", sid)
  }
  if (anyNA(works$id) || anyDuplicated(works$id) ||
      anyNA(works$publication_year) ||
      any(!works$publication_year %in% 2018:2023) ||
      any(!is.finite(works$cited_by_count)) ||
        ↪  any(works$cited_by_count < 0)) {
    stop("Invalid or duplicate records in journal sample for ",
     ↪  sid)
  }
  works
}
```

```
journal_data <- journals |>
  mutate(works = map(source_id, fetch_journal))
```

40.4 Citation distributions in the samples

`cited_by_count` is the cumulative number of citations recorded when the data were retrieved, including citations received after 2023. It does not restrict citing papers to the sampled publication years. The h-index below describes only the returned sample, and all summaries remain unnormalised for field and publication year.

```
all_works <- journal_data |>
  mutate(works = map2(works, short_name, \(w, n) w |>
    ↪  mutate(journal = n))) |>
  pull(works) |>
  bind_rows()

ggplot(all_works, aes(x = journal, y = cited_by_count + 1)) +
  geom_boxplot(fill = palette_sci(1), alpha = 0.7) +
  scale_y_log10() +
  labs(x = NULL, y = "Cumulative citations + 1 (log10 scale)") +
  theme_sci()
```

```
all_works |>
  group_by(journal) |>
  summarise(
    sampled_articles = n(),
    mean_cites = round(mean(cited_by_count), 1),
    median_cites = median(cited_by_count),
    sample_h_index = compute_h_index(cited_by_count),
    .groups = "drop"
  ) |>
  gt() |>
  cols_label(
    journal = "Journal", sampled_articles = "Sampled articles",
    mean_cites = "Mean cumulative citations", median_cites =
    ↪  "Median cumulative citations",
    sample_h_index = "Sample h-index"
  )
```

Journal	Sampled articles	Mean cumulative citations	Median cumulative citations	Sample h-index

J. Informetrics	300	45.4	21	54
JASIST	300	34.4	23	53
Scientometrics	300	37.1	19	51

40.5 Publication-cohort comparison

The following plot compares different publication cohorts at retrieval. Older cohorts have had more time to receive citations, and their composition may also differ. It is not a trajectory following the same articles over time, so it cannot establish citation-aging curves or citation half-lives. That requires time-resolved citation counts and an explicit observation window.

```
cohorts <- all_works |>
  group_by(journal, publication_year) |>
  summarise(mean_cites = mean(cited_by_count), sampled_articles
   ↪  = n(), .groups = "drop")

ggplot(cohorts, aes(x = publication_year, y = mean_cites, colour
 ↪  = journal)) +
  geom_line(linewidth = 1) +
  geom_point(aes(size = sampled_articles)) +
  scale_x_continuous(breaks = 2018:2023) +
  scale_colour_manual(values = palette_sci(3)) +
  labs(x = "Publication year", y = "Mean cumulative citations",
       colour = "Journal", size = "Sampled articles") +
  theme_sci()
```

40.6 Topical coverage

Count each article once per assigned topic. The flattened `topics` table also contains the topics' parent subfields, fields, and domains; retain only topic identifiers (`T...`). An article can contribute to several topics, while articles without topic assignments do not contribute to this chart; topic counts therefore need not sum to the number of sampled articles.

```
topic_data <- all_works |>
  select(id, journal, topics) |>
  unnest(topics, names_sep = "_") |>
  filter(str_detect(topics_id, "(^|/)T[0-9]+$")) |>
  select(id, journal, topic = topics_display_name) |>
  mutate(topic = str_to_lower(str_trim(topic))) |>
  filter(!is.na(topic), nchar(topic) >= 3) |>
```

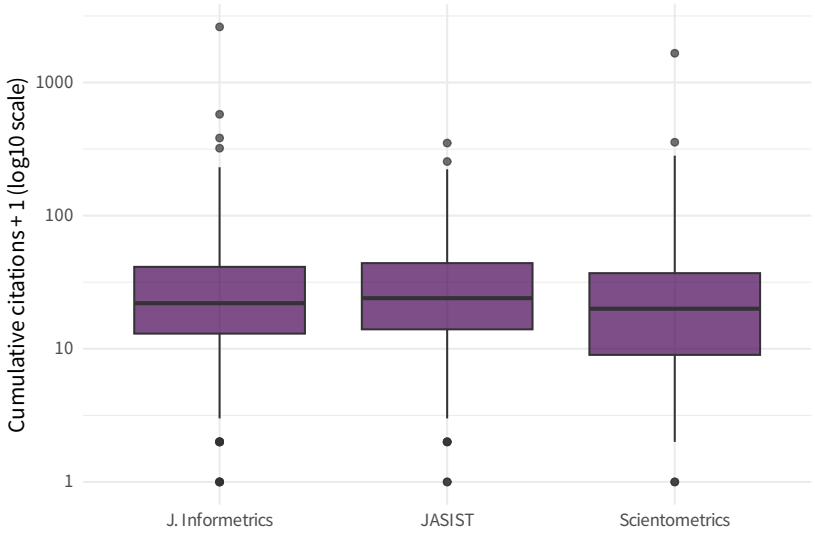


Figure 40.1: Cumulative citation distributions in the sampled articles, with one added before log scaling.

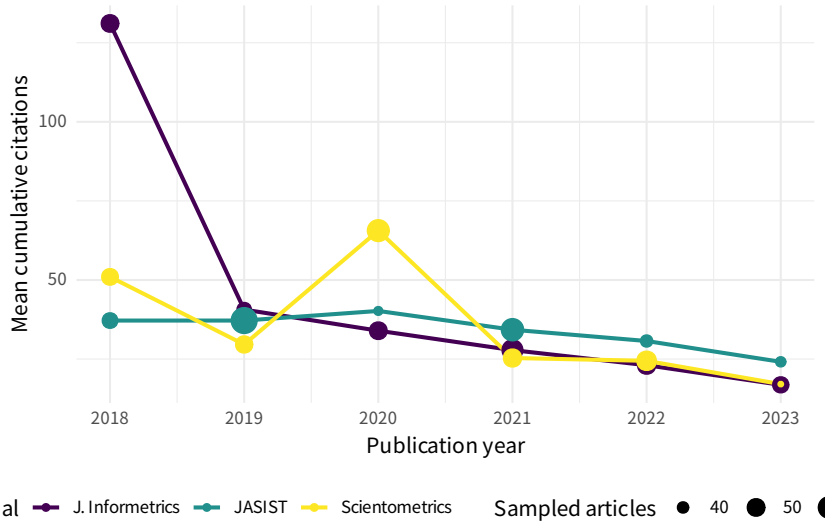


Figure 40.2: Mean cumulative citations by publication year in each journal sample; point size shows cohort sample size.

```
  distinct(id, journal, topic)

top_topics <- topic_data |>
  count(journal, topic, sort = TRUE) |>
  group_by(journal) |>
  slice_max(n, n = 10, with_ties = FALSE) |>
  ungroup()
```

```
top_topics |>
  mutate(topic = reorder_within(topic, n, journal)) |>
  ggplot(aes(x = n, y = topic)) +
  geom_col(fill = palette_sci(1)) +
  facet_wrap(~ journal, scales = "free_y", ncol = 1) +
  scale_y_reordered() +
  labs(x = "Sampled articles assigned to topic", y = NULL) +
  theme_sci(base_size = 9)
```

40.7 Reading the results

1. **Citation distributions**: Compare the plotted medians with the sample means. A large gap can indicate that a few highly cited articles dominate the mean; journal rankings should be reported only from the observed results.
2. **Publication cohorts**: Compare cohorts alongside their sample sizes. Differences can reflect time available for citations, article composition, or sampling variation; they do not demonstrate a common aging curve.
3. **Assigned topics**: Inspect overlap and differences in the displayed topics, then review representative articles before drawing conclusions about collection coverage.

40.8 Lessons learned

- Journal comparison requires multiple dimensions; no single metric tells the full story.
- Sample-based analysis is illustrative. Dependable journal evaluation requires a defined corpus, justified sampling or complete coverage, appropriate observation windows, and field-normalised indicators where relevant (Waltman, 2016).
- Collection decisions also depend on local readership, access, cost, and subject needs; citation and topic summaries alone do not measure a journal's value to a library.

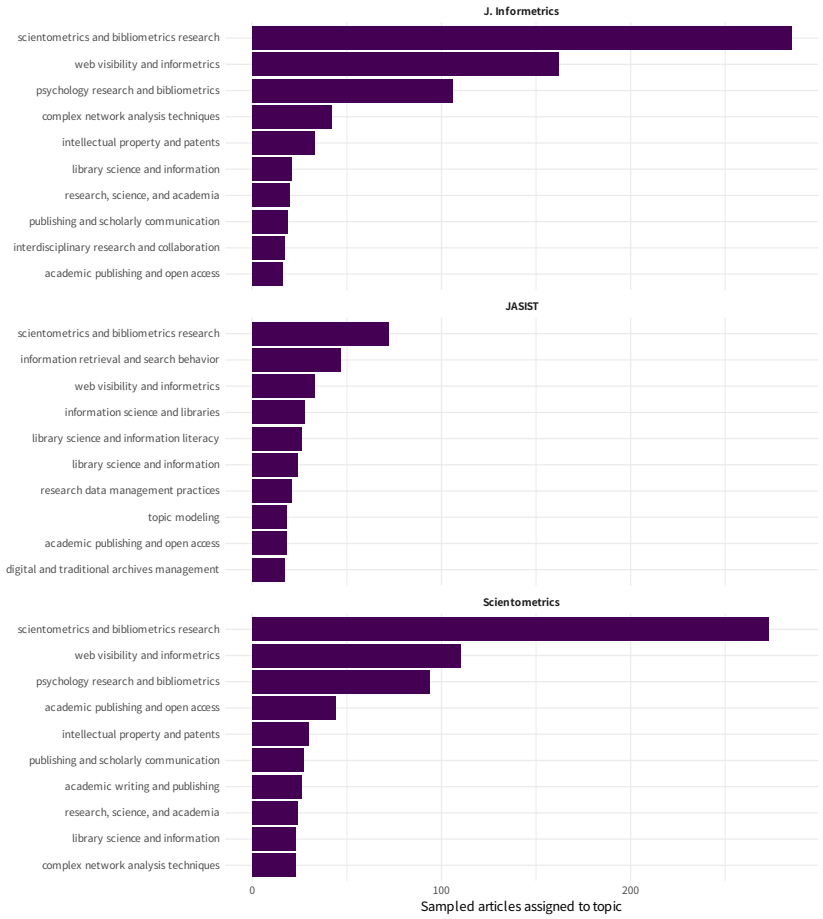


Figure 40.3: Up to 10 most frequent assigned topics in each journal sample.

Chapter 41

Case Study 4: Gender Gap in Scientific Publishing

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

41.1 Objective

Examine why a name lookup cannot establish a gender gap, using an explicitly synthetic example of two fictional journals. We compare arbitrary lookup categories, retain unmatched records in every denominator, and track lookup coverage by recorded authorship position and year. None of the records represents a real person, paper or journal; no gender identity is observed or inferred.

An actual study of gender equity requires a defensible demographic variable and its provenance, a defined population, appropriate data governance, and a plan for validation and missing data. A list of names is not a substitute for those observations. Chapter 28.1 develops these distinctions.

41.2 Setup

```
library(tidyverse)
library(glue)
library(gt)
```

```
source(here::here("R", "sci_palette.R"))
```

41.3 Synthetic teaching records

The six invented papers have explicit publication years and ordered name lists. The names are placeholders, and repeated names do not establish shared author identity. Each row created below is an authorship record, not a distinct person. This example makes no API request.

```
works_example <- tibble(
  work_id = c("example-a1", "example-a2", "example-a3",
              "example-b1", "example-b2", "example-b3"),
  journal = rep(c("Example journal A", "Example journal B"),
   ↪  each = 3),
  year = rep(2019:2021, 2),
  first_names = list(c("aria", "jordan", "lee"), c("noor",
   ↪  "ari"), "morgan",
                     c("aria", "nuri", "sam", "alex"), c("ari",
                      ↪  "noor"),
                     c("lee", "jordan", "morgan"))
)
```

41.4 Arbitrary lookup and complete denominators

For teaching purposes only, assign `aria` and `ari` to A, and `jordan` and `morgan` to B. These arbitrary categories have **no demographic meaning**. Every other name is explicitly unmatched. Dropping unmatched records would change both the denominator and the question answered by a proportion.

```
authors_example <- works_example |>
  unnest_longer(first_names, values_to = "first_name",
   ↪  indices_to = "author_order") |>
  group_by(work_id) |>
  mutate(
    n_authors = n(),
    position = case_when(
      n_authors == 1 ~ "single",
      author_order == 1 ~ "first",
      author_order == n_authors ~ "last",
```

```
      TRUE ~ "middle"
    )
  ) |>
  ungroup() |>
  mutate(
    lookup_category = case_when(
      first_name %in% c("aria", "ari") ~ "A",
      first_name %in% c("jordan", "morgan") ~ "B",
      TRUE ~ "Unmatched"
    ),
    lookup_category = factor(lookup_category, levels = c("A",
    ↪  "B", "Unmatched")),
    position = factor(position, levels = c("first", "middle",
    ↪  "last", "single"))
  )

coverage_summary <- authors_example |>
  group_by(journal) |>
  summarise(records = n(), matched = sum(lookup_category !=
  ↪  "Unmatched"),
            coverage = matched / records, .groups = "drop")
coverage_summary |> gt() |> fmt_percent(columns = coverage,
↪  decimals = 1)
```

journal	records	matched	coverage
Example journal A	6	4	66.7%
Example journal B	9	4	44.4%

41.5 Lookup categories by fictional journal

```
journal_summary <- authors_example |>
  count(journal, lookup_category, .drop = FALSE) |>
  group_by(journal) |>
  mutate(pct = n / sum(n)) |>
  ungroup()

ggplot(journal_summary, aes(x = journal, y = pct, fill =
↪  lookup_category)) +
  geom_col(position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = palette_sci(3), drop = FALSE) +
  labs(x = NULL, y = "Share of all synthetic records", fill =
  ↪  "Lookup category") +
  theme_sci()
```

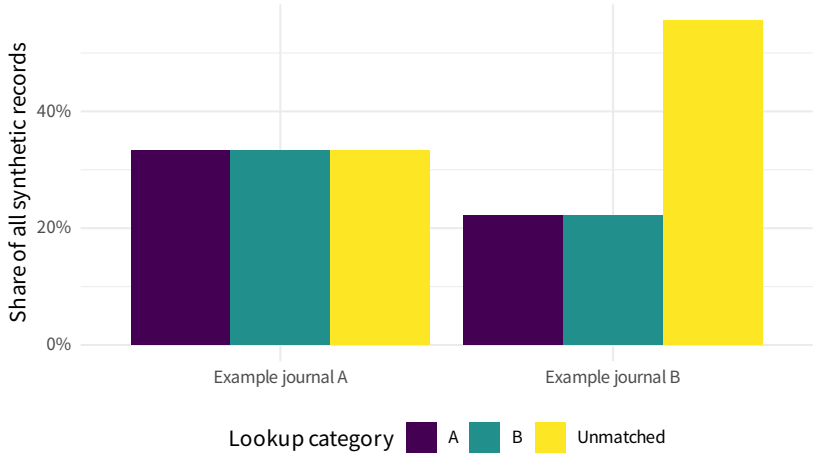


Figure 41.1: Arbitrary lookup categories among all synthetic authorship records in two fictional journals; these are not gender proportions.

41.6 Recorded authorship position

Single-author papers receive their own position label. In a real corpus, recorded order does not by itself establish seniority or contribution: the interpretation depends on authorship conventions. The following proportions retain unmatched records within every observed journal/position group.

```
position_summary_example <- authors_example |>
  count(journal, position, lookup_category) |>
  group_by(journal, position) |>
  mutate(pct = n / sum(n)) |>
  ungroup()

ggplot(position_summary_example, aes(x = position, y = pct, fill
↪  = lookup_category)) +
  geom_col(position = "dodge") +
  facet_wrap(~ journal) +
  scale_x_discrete(drop = FALSE) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = palette_sci(3), drop = FALSE) +
  labs(x = "Recorded position", y = "Share of synthetic
  ↪  records", fill = "Lookup category") +
  theme_sci()
```

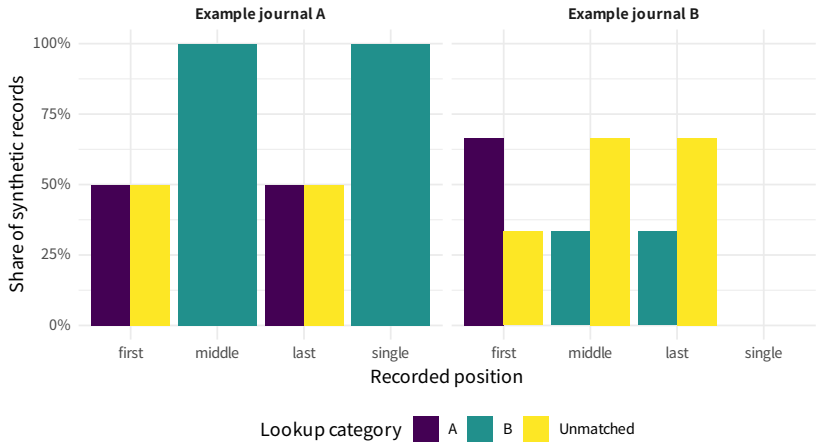


Figure 41.2: Arbitrary lookup categories by recorded position in the synthetic fixture, including unmatched records. Empty positions have no observations.

41.7 Coverage by recorded year

The year comes from the fixture's explicit publication-year column, never from an identifier. For each fictional journal/year, the denominator includes all first- or single-author records. Each group has only one such record here, so coverage is either zero or one. This illustrates denominator accounting; it supplies no evidence about a demographic trend.

```
coverage_by_year <- authors_example |>
  filter(position %in% c("first", "single")) |>
  group_by(journal, year) |>
  summarise(records = n(), matched = sum(lookup_category !=
  ↪  "Unmatched"),
            coverage = matched / records, .groups = "drop")

ggplot(coverage_by_year, aes(x = year, y = coverage, colour =
↪  journal)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2) +
  scale_x_continuous(breaks = 2019:2021) +
  scale_y_continuous(labels = scales::percent, limits = c(0, 1))
  ↪  +
  scale_colour_manual(values = palette_sci(2)) +
  labs(x = "Recorded fixture year", y = "Lookup coverage",
  ↪  colour = "Fictional journal") +
  theme_sci()
```

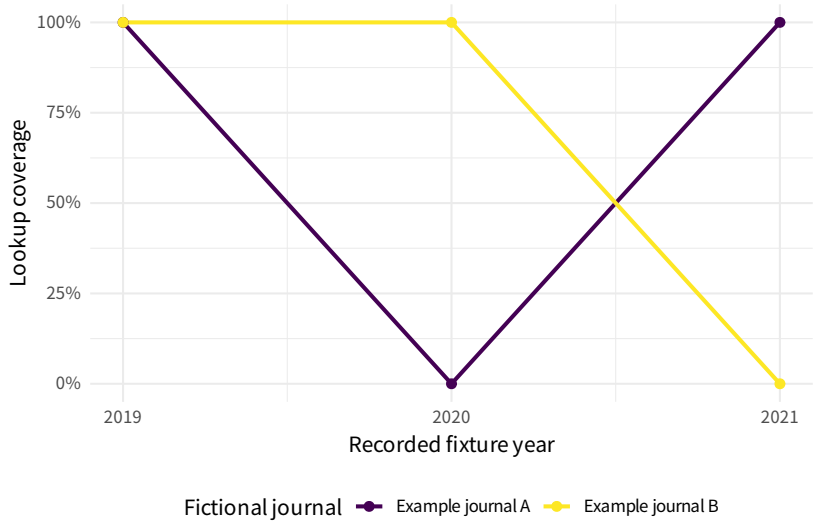


Figure 41.3: Lookup coverage among first- or single-author records by explicit fixture year. Each fictional journal/year has one record; no demographic trend is measured.

41.8 What the fixture establishes

1. **Denominators change the result.** The fixture contains 15 authorship records: four A, four B and seven unmatched. Removing unmatched records would make A and B each half of the retained records, concealing that almost half of all records had no lookup match.
2. **Coverage differs by construction.** Example journal A has four matched records out of six; Example journal B has four out of nine. These differences follow from the invented name lists, not disciplinary or demographic evidence.
3. **Identity and contribution remain unobserved.** Neither the labels, the recorded positions nor their changes across fixture years support a claim about a gender gap, seniority or progress toward equity.

41.9 Requirements for an actual study

A real study needs a scientific question, a defined population and suitable observations of the demographic variable of interest. Document their provenance, ethical basis, categories, missingness and validation. Names cannot determine a person's self-identified gender; a name-classification

service, if used for a carefully justified question, still needs context-specific validation and an explicit account of what its labels measure. A confidence score alone does not validate the resulting equity estimate. See Chapter 28.6 and Santamaría and Mihaljevic (2018).

This fixture contains no demographic observations or real journal comparison. No conclusion about a persistent gender gap, disciplinary differences or temporal progress can be drawn from it.

Chapter 42

Case Study 5: Detecting Emerging Topics

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-30

42.1 Objective

Identify emerging topics in scientometrics research (2015–2023) using keyword burst detection and topic modelling, and validate findings against known developments in the field.

42.2 Setup

```
library(tidyverse)
library(openalexR)
library(quanteda)
library(quanteda.textstats)
library(topicmodels)
library(tidytext)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
```

```
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

42.3 Data acquisition

```
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2015-01-01",
  to_publication_date = "2023-12-31",
  type = "article",
  options = list(sample = 600, seed = 42)
)

text_df <- works |>
  filter(!is.na(abstract), nchar(abstract) > 50) |>
  transmute(
    doc_id = id,
    text = paste(display_name, abstract, sep = ". "),
    year = year(publication_date)
  )

cat(glue("Documents: {nrow(text_df)}\n"))
```

```
#> Documents: 130
```

42.4 Keyword burst detection

```
corp <- corpus(text_df, docid_field = "doc_id", text_field =
↪  "text")
docvars(corp, "year") <- text_df$year

toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en")) |>
  tokens_remove(c("study", "paper", "results", "research",
  ↪  "analysis",
                  "also", "however", "using", "based"))

dfmat <- dfm(toks) |> dfm_trim(min_termfreq = 10, min_docfreq =
↪  5)
```

```
kw_by_year <- quanteda::convert(dfm_group(dfmat, groups = year),
↪  to = "data.frame") |>
  pivot_longer(-doc_id, names_to = "term", values_to = "count")
  ↪  |>
  rename(year = doc_id) |>
  mutate(year = as.integer(year))

growth <- kw_by_year |>
  group_by(term) |>
  filter(sum(count) >= 20) |>
  arrange(year) |>
  mutate(growth = (count - lag(count)) / pmax(lag(count), 1)) |>
  ungroup() |>
  filter(!is.na(growth))

recent_growth <- growth |>
  filter(year >= 2021) |>
  group_by(term) |>
  summarise(mean_growth = mean(growth), total = sum(count),
  ↪  .groups = "drop") |>
  filter(mean_growth > 0) |>
  arrange(desc(mean_growth))
```

```
recent_growth |>
  head(15) |>
  mutate(term = fct_reorder(term, mean_growth)) |>
  ggplot(aes(x = mean_growth, y = term)) +
  geom_col(fill = palette_sci(1)) +
  scale_x_continuous(labels = scales::percent) +
  labs(x = "Mean growth rate (2021–2023)", y = NULL) +
  theme_sci()
```

42.5 Keyword trajectories

```
emerging_terms <- recent_growth |> head(6) |> pull(term)

kw_by_year |>
  filter(term %in% emerging_terms) |>
  ggplot(aes(x = year, y = count, colour = term)) +
  geom_line(linewidth = 0.8) +
  geom_point(size = 1.5) +
  scale_colour_manual(values =
  ↪  palette_sci(length(emerging_terms))) +
```

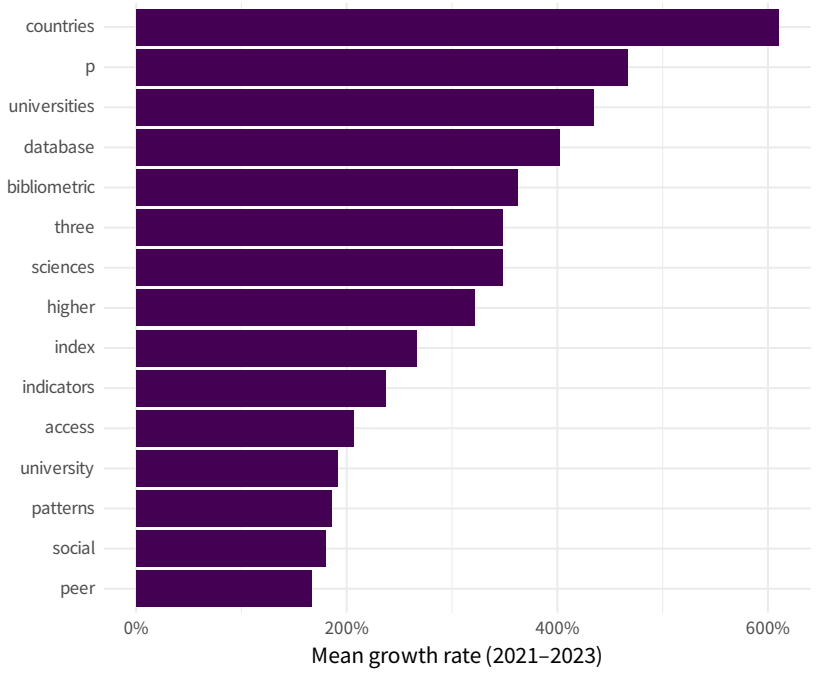


Figure 42.1: Top 15 fastest-growing keywords (2021–2023).

```
  labs(x = "Year", y = "Frequency", colour = "Keyword") +
  theme_sci()
```

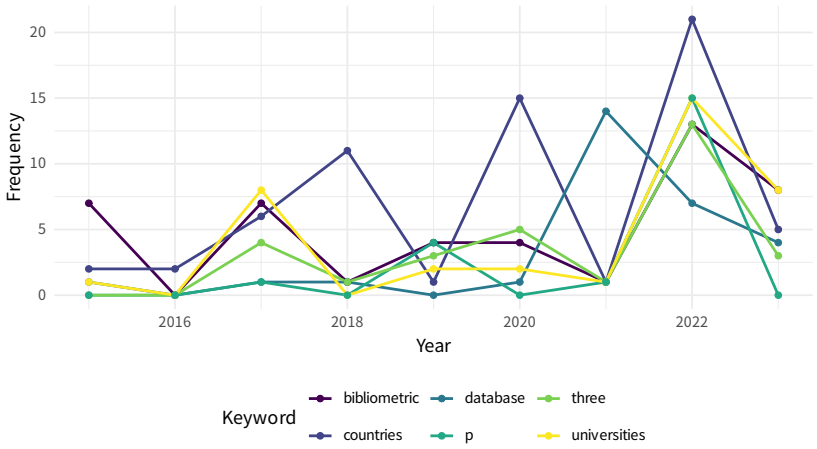


Figure 42.2: Temporal trajectories of selected emerging keywords.

42.6 Topic modelling

```
dtm <- quanteda::convert(dfmat, to = "topicmodels")
lda <- LDA(dtm, k = 8, control = list(seed = 42))

lda_topics <- tidy(lda, matrix = "beta") |>
  group_by(topic) |>
  slice_max(beta, n = 8) |>
  ungroup()
```

```
lda_topics |>
  mutate(term = reorder_within(term, beta, topic)) |>
  ggplot(aes(x = beta, y = term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ paste("Topic", topic), scales = "free_y", ncol =
  ↪  4) +
  scale_y_reordered() +
  scale_fill_manual(values = palette_sci(8)) +
  labs(x = "Word probability", y = NULL) +
  theme_sci(base_size = 9)
```

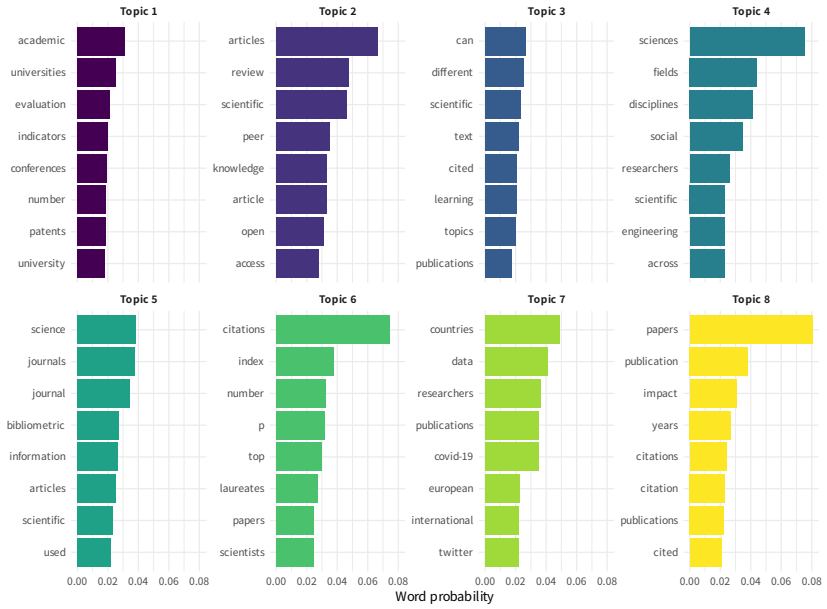


Figure 42.3: Top terms per LDA topic.

42.7 Topic prevalence by year

```
doc_topics <- tidy(lda, matrix = "gamma") |>
  left_join(text_df |> transmute(document = doc_id, year), by =
    ↪  "document")

topic_by_year <- doc_topics |>
  group_by(year, topic) |>
  summarise(mean_gamma = mean(gamma), .groups = "drop")

ggplot(topic_by_year, aes(x = year, y = mean_gamma, colour =
 ↪  factor(topic))) +
  geom_line(linewidth = 0.7) +
  scale_colour_manual(values = palette_sci(8)) +
  labs(x = "Year", y = "Mean topic proportion", colour =
    ↪  "Topic") +
  theme_sci()
```

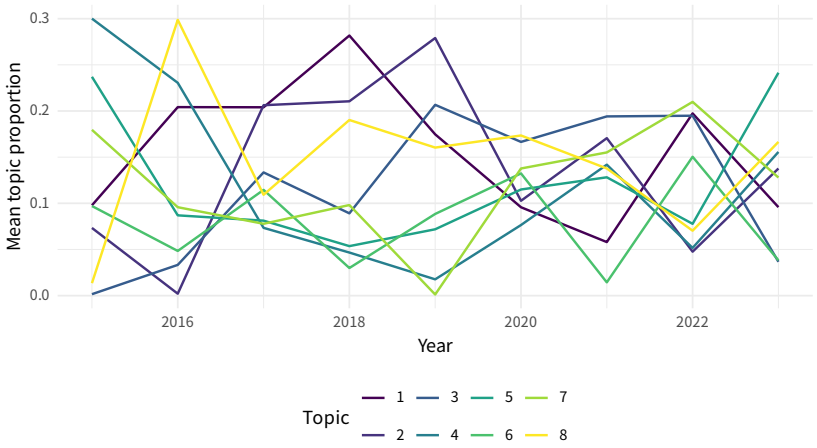


Figure 42.4: Topic prevalence over time.

42.8 Key findings

1. **Emerging keywords** align with known developments: open science, AI/machine learning applications, equity and diversity, and preprints.
2. **Topic evolution** shows gradual shifts in emphasis rather than abrupt changes, consistent with a mature field incorporating new tools.
3. **Growth rates** should be interpreted cautiously: terms with low baseline frequencies can show high percentage growth from small increases.
4. **Validation**: the topics identified by LDA correspond to recognisable subfields, lending credibility to the unsupervised approach.

42.9 Lessons learned

- Combining keyword growth rates with topic models provides complementary perspectives: growth rates capture individual terms, while topics capture thematic clusters.
- Database coverage growth can create artefactual "emergence." Always normalise by total corpus size.
- Emerging topics in scientometrics mirror broader trends in science policy (open access mandates, equity initiatives, AI adoption) (Kleinberg, 2003).

Appendix A

R Setup, renv, and Docker

R. Heller · Date: 2026-05-09 · Last changed: 2026-11-10

A.1 R installation and current first-use path

Install R from CRAN. The current candidate was locally validated with R 4.6.1. An IDE such as RStudio is optional. Rendering also requires Pandoc; PDF output additionally needs XeLaTeX/TinyTeX and R with Cairo support. Compiled packages may require a C/C++ toolchain and system libraries; the Linux dependencies are listed in `.github/workflows/render.yml`.

From the repository root, install and check the book's R dependencies:

```
RENV_CONFIG_AUTOLOADER_ENABLED=false Rscript --vanilla
 ↪  scripts/install-book-deps.R
RENV_CONFIG_AUTOLOADER_ENABLED=false Rscript --vanilla
 ↪  scripts/install-book-deps.R --check
```

The installer installs missing packages from current CRAN and checks that their namespaces load. It does not restore historical versions or certify an immutable software environment. `--check` performs no installation or network access; `--list` prints the package inventory. Use `--library=/path/to/library` to select an R library, or `--include-optional` for packages used only by unevaluated examples and exercises. Optional Python, SQL, OCR and deployment prerequisites must be configured separately.

Internet access is required for installation and uncached provider requests. Configure an activated OpenAlex key locally as `OPENALEX_API_KEY`,

outside Git, for requests that need authenticated paging. `SCIENTOMETRICS_OFFLINE=1` requires existing exact-query OpenAlex caches in `_freeze/openalex_cache/`; it does not make other providers offline and cannot supply missing cache entries. The repository `README.md` describes the helper functions in `R/` and the sample data in `data/`.

A.2 Package management with renv

renv can record and restore a project's R package versions. This repository retains a historical `renv.lock` and automatic activation in `.Rprofile`, but that lockfile has **not been qualified for the current book**. The first-use commands above bypass project startup and use the dependency installer. Do not assume that restoring the historical lockfile will reproduce the current validated candidate.

The following is an optional restoration example for a project whose lockfile you have independently validated, rather than a required setup step for this book:

```
install.packages("renv")
renv::restore()
```

Restoration attempts to install the recorded versions; success also depends on compatible R, system libraries and available package sources. In a development environment, add a package and deliberately update the lockfile only after validating the resulting environment:

```
renv::install("newpackage") # Replace with the intended package
↪  name.
renv::snapshot()
```

A software lockfile does not freeze live API results. Exact reproduction also requires the same licensed input snapshots and analysis configuration.

A.3 Core packages

The table below lists the primary packages used throughout the book.

Package	Purpose	Chapter(s)
openalexR	OpenAlex API access	5–37

Package	Purpose	Chapter(s)
tidyverse	Data wrangling and visualisation	All
igraph	Network analysis	15–20
tidygraph	Tidy network manipulation	15–20
ggraph	Network visualisation	15–20
quanteda	Text mining and corpus management	21–25
topicmodels	LDA topic modelling	22, CS-5
stm	Structural topic models	22
gt	Publication-quality tables	Throughout
targets	Pipeline orchestration	34
plotly	Interactive charts	35
DT	Interactive data tables	35
shiny	Web applications	37
word2vec	Word embeddings	24
uwot	UMAP dimensionality reduction	24

A.4 Docker environment

The repository's `Dockerfile` is an **optional historical recipe**, based on `rocker/verse:4.4.1` and `renv::restore()`. Its restore path has not been qualified for the current book. It also downloads an amd64 Quarto package, so other architectures need a compatible recipe. Docker does not remove the need to validate the lockfile, system dependencies, API access and resulting outputs.

For a deliberate evaluation of that recipe, build it and inspect its R version:

```
docker build -t scientometrics-in-r .
docker run --rm scientometrics-in-r R --version
```

The image's default command calls `bookdown::render_book()` for all formats; it does not launch RStudio Server. Successful construction and execution of this historical image are not claimed here. Use the current first-use path above for the locally validated candidate.

A.5 Render the book

This repository is configured for **bookdown**, with `_bookdown.yml` and `_output.yml`. From the repository root, render HTML and refresh its download manifest:

```
SCIENTOMETRICS_REBUILD_CHUNKS=1
 ↪  RENV_CONFIG_AUTOLOADER_ENABLED=false Rscript --vanilla -e
 ↪  'bookdown::render_book("index.Rmd", output_format =
 ↪  "bookdown::bs4_book")'
Rscript --vanilla scripts/build-download-manifest.R
```

`SCIENTOMETRICS_REBUILD_CHUNKS=1` re-executes analysis chunks while retaining exact-query provider caches. Missing provider caches still require API access. The configured output directory is `docs/`. Serve it locally, for example:

```
python3 -m http.server 8000 --directory docs
```

Then visit `http://localhost:8000`. PDF additionally needs XeLaTeX/TinyTeX; `output_format = "all"` requests HTML, PDF and EPUB. Before distribution, run `Rscript --vanilla scripts/repair-epub.R docs/scientometrics-in-r.epub` to validate the ebook's internal links and bundle its citation and licence files, then run `Rscript --vanilla scripts/build-download-manifest.R`. Local outputs do not update the published site. The README lists the PDF font prerequisites.

Quarto appears in optional examples elsewhere in the book, but `quarto render` is not the build command for this repository. A Quarto CLI is needed only when working through those separate examples.

Appendix B

SQL Primer for the OpenAlex Snapshot

R. Heller · Date: 2026-05-09 · Last changed: 2026-05-11

B.1 When to use the snapshot

The OpenAlex API (Chapter 5) is ideal for moderate queries. For analyses requiring millions of records — full-database bibliographic coupling, global co-authorship networks, or exhaustive field delineation — the OpenAlex snapshot is more practical. The snapshot is a full database dump released monthly in compressed JSON-lines format.

B.2 Loading the snapshot

OpenAlex provides scripts to load the snapshot into PostgreSQL. After loading:

```
-- Count total works
SELECT COUNT(*) FROM openalex.works;

-- Recent articles with citation counts
SELECT id, display_name, publication_year, cited_by_count
FROM openalex.works
WHERE type = 'article'
  AND publication_year >= 2020
```

```
ORDER BY cited_by_count DESC
LIMIT 100;
```

B.3 Common queries

B.3.1 Publication counts by year

```
SELECT publication_year, COUNT(*) AS n_works
FROM openalex.works
WHERE type = 'article'
GROUP BY publication_year
ORDER BY publication_year;
```

B.3.2 Top institutions by output

```
SELECT i.display_name, COUNT(DISTINCT w.id) AS n_works
FROM openalex.works w
JOIN openalex.works_authorships wa ON w.id = wa.work_id
JOIN openalex.works_authorships_institutions wai
  ON wa.work_id = wai.work_id AND wa.author_position =
  ↪  wai.author_position
JOIN openalex.institutions i ON wai.institution_id = i.id
WHERE w.publication_year = 2023
  AND w.type = 'article'
GROUP BY i.display_name
ORDER BY n_works DESC
LIMIT 20;
```

B.3.3 Co-authorship pairs

```
SELECT a1.author_id AS author_a,
       a2.author_id AS author_b,
       COUNT(DISTINCT a1.work_id) AS n_collabs
FROM openalex.works_authorships a1
JOIN openalex.works_authorships a2
  ON a1.work_id = a2.work_id AND a1.author_id < a2.author_id
GROUP BY a1.author_id, a2.author_id
HAVING COUNT(DISTINCT a1.work_id) >= 5
```

```
ORDER BY n_collabs DESC
LIMIT 100;
```

B.4 Connecting from R

```
library(DBI)
library(RPostgres)

con <- dbConnect(
  Postgres(),
  dbname = "openalex",
  host = "localhost",
  port = 5432,
  user = "openalex"
)

top_journals <- dbGetQuery(con, "
  SELECT s.display_name, COUNT(*) AS n_works
  FROM openalex.works w
  JOIN openalex.sources s ON w.primary_location_source_id = s.id
  WHERE w.publication_year = 2023
  GROUP BY s.display_name
  ORDER BY n_works DESC
  LIMIT 20
")

dbDisconnect(con)
```

B.5 Performance tips

- **Index key columns**: `work_id`, `author_id`, `institution_id`, `publication_year`, and `cited_by_count` should all be indexed.
- **Partition by year**: For time-series queries, partitioning the works table by `publication_year` speeds up range scans substantially.
- **Materialised views**: Pre-compute expensive joins (e.g., institution–work counts) as materialised views and refresh them after each snapshot update.
- **Use COPY for bulk loads**: The `COPY` command is orders of magnitude faster than row-by-row `INSERT`.

Appendix C

Indicator Crosswalk Across Databases

R. Heller · Date: 2026-05-09 · Last changed: 2026-05-11

This appendix maps common bibliometric indicators to the data sources and tools that support them. Use it to choose the right source for your analysis.

C.1 Publication-level indicators

Indicator	OpenAlex	Web of Science	Scopus	Dimensions	Chapter
Citation count	Y	Y	Y	Y	9–10
Field-normalised citations (MNCS)	Compute	CNCI (built-in)	FWCI (built-in)	FCR (built-in)	13
PP(top 10%)	Compute	Compute	Compute	Compute	10
Open access status	Y	Y	Y	Y	27

Indicator	OpenAlex	Web of Science	Scopus	Dimensions	Chapter
Funding acknowledgements	Y (partial)	Y	Y	Y	30
Abstract text	Y	Y	Y	Y	21–25
References list	Y	Y	Y	Y	16
Author affiliations	Y	Y	Y	Y	12–13

C.2 Author-level indicators

Indicator	OpenAlex	Web of Science	Scopus	Dimensions	Chapter
h-index	Compute	Y	Y	Compute	10, 12
Publication count	Y	Y	Y	Y	12
ORCID integration	Y	Y	Y	Y	12
Co-author network	Compute	Compute	Compute	Compute	15

C.3 Journal-level indicators

Indicator	OpenAlex	Web of Science	Scopus	Chapter
Journal Impact Factor	—	Y	—	11
CiteScore	—	—	Y	11
SNIP	—	—	Y	11
SJR	—	—	Y	11
Article volume	Y	Y	Y	11
Citation distribution	Compute	Compute	Compute	11

C.4 Institution-level indicators

Indicator	OpenAlex	Leiden Ranking	Web of Science	Scopus	Chapter
Publication counts (full/fractional)	Compute	Y	Compute	Compute	13
MNCS / FWCI	Compute	Y	CNCI	FWCI	13
PP(top 10%)	Compute	Y	Compute	Compute	13
Collaboration rates	Compute	Y	Compute	Compute	15
Subject profile	Y	Y	Y	Y	19

C.5 Network indicators

Indicator	Data needed	Source	Chapter
Co-authorship strength	Author pairs per paper	Any	15
Bibliographic coupling	Reference lists	OpenAlex, WoS, Scopus	16
Co-citation	Citing documents	OpenAlex, WoS, Scopus	16
Keyword co-occurrence	Keywords/topics	Any	17
Betweenness centrality	Network graph	Computed	15
Community structure	Network graph	Computed	18

C.6 Notes

- **“Y”** = available directly from the data source.
- **“Compute”** = the raw data is available but the indicator must be calculated.
- **“—”** = not available from that source.

- OpenAlex is the only fully free and open source in this table. Web of Science and Scopus require institutional subscriptions. Dimensions offers a free tier with limited functionality.
- Field normalisation always requires reference values. When a source does not provide them natively, you can compute them from the full corpus using the method in Chapter 13.

Appendix D

Cheatsheets and Quick Reference

R. Heller · Date: 2026-05-09 · Last changed: 2026-11-10

D.1 openalexR quick reference

```
library(openalexR)

# Fetch works by journal (source)
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  from_publication_date = "2020-01-01",
  to_publication_date = "2023-12-31",
  type = "article"
)

# Fetch works by institution
works <- fetch_openalex(
  entity = "works",
  authorships.institutions.id = "I63966007",
  from_publication_date = "2022-01-01",
  type = "article"
)

# Fetch works by topic
works <- fetch_openalex(
  entity = "works",
```

```
  topics.id = "T10102",
  from_publication_date = "2023-01-01"
)

# Fetch author profile
author <- fetch_openalex(entity = "authors", id = "A5023888391")

# Fetch institution profile
inst <- fetch_openalex(entity = "institutions", id =
 ↪  "I63966007")

# Sampling
works <- fetch_openalex(
  entity = "works",
  primary_location.source.id = "S148561398",
  options = list(sample = 500, seed = 42)
)
```

D.2 quanteda pipeline

```
library(quanteda)

# Full text-analysis pipeline
corp <- corpus(df, docid_field = "id", text_field = "text")
toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
  tokens_tolower() |>
  tokens_remove(stopwords("en"))
dfmat <- dfm(toks) |>
  dfm_trim(min_termfreq = 5, min_docfreq = 3)

# TF-IDF weighting
dfmat_tfidf <- dfm_tfidf(dfmat)

# Top features
topfeatures(dfmat, 20)

# Keyword-in-context
kwic(toks, pattern = "open access", window = 5)
```

D.3 igraph / tidygraph / ggraph

```
library(igraph)
library(tidygraph)
library(ggraph)

# Build graph from edge list
g <- graph_from_data_frame(edges, directed = FALSE, vertices =
 ↪  nodes)

# Tidygraph conversion
tg <- as_tbl_graph(g) |>
  activate(nodes) |>
  mutate(
    degree = centrality_degree(),
    between = centrality_betweenness(),
    community = group_leiden(resolution = 1.0)
  )

# ggraph visualisation
ggraph(tg, layout = "fr") +
  geom_edge_link(aes(alpha = weight), show.legend = FALSE) +
  geom_node_point(aes(size = degree, colour =
   ↪  factor(community))) +
  geom_node_text(aes(label = name), repel = TRUE, size = 3) +
  theme_void()
```

D.4 Common data wrangling patterns

```
library(tidyverse)

# Unnest OpenAlex authorships
authors <- works |>
  select(work_id = id, authorships) |>
  unnest(authorships, names_sep = "_")

# Unnest OpenAlex topics
topics <- works |>
  select(work_id = id, topics) |>
  unnest(topics, names_sep = "_")

# Compute h-index
compute_h_index <- function(citations) {
  citations <- sort(citations, decreasing = TRUE)
```

```
  sum(citations >= seq_along(citations))
}

# Field normalisation
works |>
  group_by(field, year) |>
  mutate(
    field_mean = mean(cited_by_count),
    mncs = cited_by_count / pmax(field_mean, 1)
  )

# Fractional counting
works |>
  unnest(authorships, names_sep = "_") |>
  group_by(id) |>
  mutate(frac = 1 / n()) |>
  ungroup()
```

D.5 Helper functions

The book's helper functions in `R/` are used throughout the book. Load them with `source()` from `R/api_helpers.R`, `R/utils.R` and `R/sci_palette.R`, for example `source(here::here("R", "utils.R"))`:

Function	Purpose	Example
`fetch_openalex(...)`	Cached, rate-limited OpenAlex queries	`fetch_openalex(entity = "works", ...)`
`dedupe_by_doi(df)`	Remove duplicate records by DOI	`works |> dedupe_by_doi()`
`compute_h_index(citations)`	Compute h-index from citation vector	`compute_h_index(c(10, 8, 5, 3, 1))` → 3
`field_normalize(cites, means)`	MNCS normalisation (vector)	`field_normalize(c(20, 5), c(10, 10))` → `c(2.0, 0.5)`
`compute_mncs(df)`	MNCS for a grouped data frame	`df |> compute_mncs()`
`build_coauth_graph(works)`	Co-authorship igraph from works	`build_coauth_graph(works)`
`kleinberg_bursts(kw, dates)`	Keyword burst detection	`kleinberg_bursts(keywords, dates)`
`palette_sci(n)`	Viridis colour palette	`palette_sci(4)`

Function	Purpose	Example
`theme_sci()`	Minimal ggplot2 theme	`+ theme_sci()`

D.6 Useful keyboard shortcuts

Action	RStudio	VS Code
Run current line/selection	Ctrl+Enter	Ctrl+Enter
Run current chunk	Ctrl+Shift+Enter	Ctrl+Shift+Enter
Insert pipe |>	Ctrl+Shift+M	—
Insert assignment <-	Alt+-	—
Render document	Ctrl+Shift+K	—
Go to file	Ctrl+.	Ctrl+P

Appendix E

Exercise Solutions

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

This appendix provides suggested solutions for all chapter exercises. The code sketches below are meant to guide you toward a working answer; they are not the only valid approach. Because each solution references data built during the corresponding chapter, code chunks are **not evaluated** here — run them inside the chapter project after completing the worked example.

Chapter 1 — What Is Scientometrics?

1. Explore a different field. Swap the search term and fetch a fresh sample. Comparing the publication trend and top journals reveals how different fields vary in volume, growth rate, and journal concentration.

```
library(openalexR)
library(tidyverse)

works_ml <- fetch_openalex(
 entity = "works",
 search = "machine learning",
 from_publication_date = "2013-01-01",
 to_publication_date = "2023-12-31",
 count_only = FALSE,
 per_page = 200
) |> slice_sample(n = 500)

works_ml |>
 count(publication_year) |>
 ggplot(aes(publication_year, n)) +
```

```
 geom_col() +
 labs(title = "Machine-learning publications per year",
      x = "Year", y = "Works")
```

Key insight: High-growth fields like machine learning show steeper annual increases and a broader spread across journals compared to a niche field like scientometrics.

2. Uncited works. Filter to zero-citation records and examine the year distribution.

```
works |>
 filter(cited_by_count == 0) |>
 count(publication_year) |>
 mutate(pct = n / sum(n)) |>
 ggplot(aes(publication_year, pct)) +
 geom_col() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "Distribution of uncited works by year",
      x = "Year", y = "Share of uncited works")
```

Key insight: Recent publication years dominate because papers need time to accumulate citations — a window effect.

3. Open Access status. Compute yearly OA proportions to reveal the trend.

```
works |>
 mutate(year = publication_year) |>
 group_by(year) |>
 summarise(oa_share = mean(is_oa, na.rm = TRUE)) |>
 ggplot(aes(year, oa_share)) +
 geom_line() +
 geom_point() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "Open-access share over time",
      x = "Year", y = "OA proportion")
```

Key insight: OA proportions have risen steadily, driven by funder mandates and the growth of gold OA journals.

4. Growth curve. Plot annual counts on a log scale; linearity implies exponential growth.

```
works |>
 filter(publication_year >= 2000, publication_year <= 2023) |>
 count(publication_year) |>
 ggplot(aes(publication_year, n)) +
```

```
 geom_point() +
 geom_smooth(method = "lm") +
 scale_y_log10() +
 labs(title = "Publication growth (log scale)",
      x = "Year", y = "Works (log)")
```

Key insight: If the points fall roughly on a straight line, growth is approximately exponential, consistent with Price's law.

Chapter 2 — Theoretical Underpinnings

1. Lotka exponent. Fit a linear model in log-log space to estimate the exponent.

```
library(tidyverse)

author_prod <- works |>
 unnest(author) |>
 count(au_id, name = "n_papers")

freq_tbl <- author_prod |>
 count(n_papers, name = "n_authors")

fit <- lm(log(n_authors) ~ log(n_papers), data = freq_tbl)
coef(fit)  # slope ≈ -2 if Lotka holds

ggplot(freq_tbl, aes(n_papers, n_authors)) +
 geom_point() +
 scale_x_log10() +
 scale_y_log10() +
 geom_smooth(method = "lm", se = FALSE) +
 labs(title = "Lotka's law fit",
      x = "Papers per author (log)", y = "Number of authors
      ↪  (log)")
```

Key insight: Lotka predicted an exponent near 2; empirical values typically range from 1.5 to 3.5, depending on the field and time window.

2. Bradford zones. Divide journals into three zones of roughly equal article counts.

```
journal_counts <- works |>
 count(so, sort = TRUE) |>
 mutate(cum = cumsum(n),
        total = sum(n),
```

```
        zone = case_when(
         cum <= total / 3 ~ "Core",
         cum <= 2 * total / 3 ~ "Mid",
         TRUE ~ "Periphery"
        ))

journal_counts |> count(zone)
```

Key insight: The core zone contains very few journals; the periphery contains many. The ratio between successive zone sizes should approximate Bradford's multiplier.

3. Your field's Zipf distribution. Extract title words and plot rank vs. frequency on log-log axes.

```
library(tidytext)

word_freq <- works |>
 select(id, display_name) |>
 unnest_tokens(word, display_name) |>
 anti_join(stop_words, by = "word") |>
 count(word, sort = TRUE) |>
 mutate(rank = row_number())

ggplot(word_freq, aes(rank, n)) +
 geom_line() +
 scale_x_log10() +
 scale_y_log10() +
 labs(title = "Zipf's law for title words",
      x = "Rank (log)", y = "Frequency (log)")
```

Key insight: A roughly linear relationship on log-log axes confirms Zipf's law. Domain-specific terms dominate the top ranks.

4. Matthew Effect simulation. Simulate preferential attachment and compare the resulting distribution to Lotka's law.

```
set.seed(20260509)
n_authors <- 100
papers <- rep(1, n_authors)

for (step in seq_len(50)) {
 probs <- papers / sum(papers)
 winner <- sample(n_authors, 1, prob = probs)
 papers[winner] <- papers[winner] + 1
}

tibble(papers = papers) |>
```

```
 count(papers, name = "n_authors") |>
 ggplot(aes(papers, n_authors)) +
 geom_point() +
 scale_x_log10() +
 scale_y_log10() +
 labs(title = "Simulated preferential attachment",
      x = "Papers (log)", y = "Authors (log)")
```

Key insight: Preferential attachment generates a skewed distribution resembling Lotka's law, illustrating the Matthew Effect.

Chapter 3 — Ethics and Responsible Metrics

1. Leiden Manifesto audit. This is a qualitative exercise. Evaluate a real hiring policy against the ten principles.

```
# No code required. Suggested structure:
# For each of the 10 Leiden Manifesto principles:
#   1. State the principle
#   2. Quote the relevant policy text (if any)
#   3. Assess: satisfied / partially satisfied / violated
#   4. Explain your reasoning
# Summarise overall compliance in a paragraph.
```

Key insight: Most policies rely heavily on the h-index or JIF, violating principles 6 (field variation) and 7 (qualitative portfolio assessment).

2. Self-citation across fields. Compare self-citation rates between two different fields.

```
library(openalexR)
library(tidyverse)

fields <- c("bibliometrics", "genomics")
results <- map(fields, \(term) {
 w <- fetch_openalex(entity = "works", search = term,
                     from_publication_date = "2020-01-01",
                     to_publication_date = "2023-12-31") |>
   slice_sample(n = 200)
 w |> mutate(field = term)
}) |> list_rbind()

# Self-citation proxy: papers citing other papers by the same
 ↪  first author
results |>
```

```
  group_by(field) |>
  summarise(mean_cites = mean(cited_by_count, na.rm = TRUE),
            median_cites = median(cited_by_count, na.rm = TRUE))
```

Key insight: Self-citation baselines differ by field. Higher self-citation in some fields is structural (smaller communities) rather than manipulative.

3. DORA compliance check. A research exercise using the DORA signatory database.

```
# No code required. Visit https://sfdora.org/signers/
# Select three major funders and compare their grant evaluation
# criteria against DORA commitments:
#   - Do they avoid using JIF in evaluation?
#   - Do they assess research outputs on their own merit?
#   - Do they consider a broad range of impact measures?
```

Key insight: Many signatories still reference journal prestige in practice, revealing a gap between stated commitments and implementation.

4. Goodhart's Law thought experiment. Propose a new indicator and analyse potential gaming strategies.

```
# Example indicator: "Replication Impact Score"
#   = fraction of a researcher's papers that have been
 ↪  independently replicated
#
# Potential gaming strategies:
#   - Publish trivially replicable results
#   - Arrange replication exchanges with collaborators
#   - Focus on small, easily replicated studies rather than
 ↪  ambitious work
#
# The numerator (replications) and denominator (total papers)
 ↪  are both gameable.
# This illustrates Goodhart's Law: once a measure becomes a
 ↪  target,
# it ceases to be a good measure.
```

Key insight: Any ratio-based metric can be gamed on both numerator and denominator. Indicator design must anticipate behavioural responses.

Chapter 4 — Data Sources Compared

1. Coverage gap analysis. Query the same DOIs in OpenAlex and Crossref and compare.

```
library(openalexR)
library(rcrossref)
library(tidyverse)

dois <- works |> slice_sample(n = 20) |> pull(doi) |> na.omit()

oa_results <- map(dois, \(d) {
 tryCatch(fetch_openalex(entity = "works", doi = d), error =
  ↪  \(e) NULL)
}) |> compact()
oa_found <- length(oa_results)

cr_results <- map(dois, \(d) {
 tryCatch(cr_works(d)$data, error = \(e) NULL)
}) |> compact()
cr_found <- length(cr_results)

tibble(
 Source = c("OpenAlex", "Crossref"),
 Found = c(oa_found, cr_found),
 Total = length(dois),
 Coverage = Found / Total
)
```

Key insight: OpenAlex generally has high DOI coverage because it ingests Crossref data, but abstract availability may differ.

2. Disciplinary bias. Compare indexed works across contrasting fields.

```
fields <- list(
 biology = "https://openalex.org/C86803240",
 philosophy = "https://openalex.org/C138885662"
)

coverage <- imap(fields, \(concept_id, name) {
 fetch_openalex(entity = "works", concept.id = concept_id,
          count_only = TRUE) |>
   mutate(field = name)
}) |> list_rbind()

coverage
```

Key insight: STEM fields are substantially better covered than humanities, reflecting the historical journal-centricity of bibliometric databases.

3. Citation count concordance. Correlate citation counts from two sources.

```
library(rcrossref)

comparison <- works |>
 slice_sample(n = 50) |>
 filter(!is.na(doi)) |>
 rowwise() |>
 mutate(cr_cites = tryCatch(
   cr_works(doi)$data$is_referenced_by_count,
   error = \(e) NA_integer_
 )) |>
 ungroup()

cor(comparison$cited_by_count, comparison$cr_cites,
    use = "complete.obs", method = "spearman")

ggplot(comparison, aes(cited_by_count, cr_cites)) +
 geom_point() +
 geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
 labs(x = "OpenAlex citations", y = "Crossref citations",
      title = "Citation count concordance")
```

Key insight: Counts correlate strongly but are not identical; differences arise from coverage scope and update frequency.

4. Preprint tracking. Check whether bioRxiv DOIs appear in OpenAlex with published links.

```
biorxiv_dois <- c(
 "10.1101/2023.01.01.123456",
 "10.1101/2023.02.15.234567"
 # ... add 8 more real bioRxiv DOIs
)

preprint_status <- map(biorxiv_dois, \(d) {
 tryCatch({
   w <- fetch_openalex(entity = "works", doi = d)
   tibble(doi = d, found = TRUE,
          has_published = !is.null(w$ids$doi))
 }, error = \(e) tibble(doi = d, found = FALSE, has_published =
 ↪  FALSE))
}) |> list_rbind()

mean(preprint_status$found)
mean(preprint_status$has_published)
```

Key insight: Most recent bioRxiv preprints appear in OpenAlex; the proportion linked to a published version depends on recency and field.

5. Source selection exercise. This is a qualitative exercise requiring no code.

```
# Recommended answer structure:
# 1. OpenAlex: free, broad coverage, good for large-scale
 ↪  analysis
#    - Limitation: may under-represent regional journals
# 2. African Journals Online (AJOL): specialised coverage of
#    Sub-Saharan African journals
# 3. Combination approach: use OpenAlex for global context,
#    supplement with AJOL for regional coverage
#
# Justification should address:
#   - Coverage of nursing literature
#   - Representation of Sub-Saharan African journals
#   - Cost (OpenAlex is free; Scopus/WoS require subscriptions)
#   - Reproducibility (OpenAlex is open; others are proprietary)
```

Key insight: No single source is sufficient; combining an open global database with a regional index maximises both coverage and reproducibility.

Chapter 5 — APIs and Packages

1. Fetch works by institution. Find your institution in OpenAlex and retrieve its works.

```
library(openalexR)
library(tidyverse)

# Step 1: find the institution
my_inst <- fetch_openalex(entity = "institutions",
                          search = "University of Example")
my_inst_id <- my_inst$id[1]

# Step 2: fetch works
inst_works <- fetch_openalex(entity = "works",
                             authorships.institutions.lineage =
                              ↪  my_inst_id,
                             from_publication_date = "2020-01-01")

nrow(inst_works)
mean(!is.na(inst_works$doi))  # DOI coverage rate
```

Key insight: DOI coverage varies by institution; institutions publishing more in regional or non-English journals tend to have lower DOI rates.

2. Compare Crossref metadata. Retrieve the same work from both APIs and compare.

```
library(rcrossref)

sample_doi <- inst_works$doi[1]

oa_meta <- fetch_openalex(entity = "works", doi = sample_doi)
cr_meta <- cr_works(sample_doi)$data

tibble(
 field = c("citations", "title_match"),
 OpenAlex = c(oa_meta$cited_by_count,
              oa_meta$display_name),
 Crossref = c(cr_meta$is_referenced_by_count,
              cr_meta$title)
)
```

Key insight: Citation counts differ because each database has a different citation graph. Crossref counts only DOI-based citations; OpenAlex resolves more.

3. Build a multi-year corpus. Fetch all works from a journal across multiple years.

```
journal_works <- fetch_openalex(
 entity = "works",
 primary_location.source.id = "S4210174107",  # example journal
  ↪  ID
 from_publication_date = "2015-01-01",
 to_publication_date = "2023-12-31"
)

journal_works |>
 group_by(publication_year) |>
 summarise(n = n(), median_cites = median(cited_by_count)) |>
 pivot_longer(c(n, median_cites)) |>
 ggplot(aes(publication_year, value)) +
 geom_line() +
 geom_point() +
 facet_wrap(~name, scales = "free_y") +
 labs(title = "Journal trend: publications and citations",
      x = "Year", y = NULL)
```

Key insight: Median citation counts typically decrease for recent years due to the citation time window, while publication counts may rise if the journal is growing.

Chapter 6 — Parsing Native Exports

1. Export and parse. Parse a BibTeX file from Google Scholar using `bib2df`.

```
library(bib2df)
library(tidyverse)

bib <- bib2df("scholar_export.bib")
ncol(bib)
map_int(bib, \(x) sum(!is.na(x))) |>
 enframe("field", "populated") |>
 arrange(desc(populated))
```

Key insight: Google Scholar BibTeX exports often lack abstracts and keywords, making them less useful for text-based analyses.

2. Format comparison. Parse the same records from CSV and RIS formats and compare fields.

```
library(bibliometrix)

csv_data <- read_csv("scopus_export.csv")
ris_data <- convert2df("scopus_export.ris", dbsource = "scopus",
                       format = "plaintext")

cat("CSV columns:", ncol(csv_data), "\n")
cat("RIS columns:", ncol(ris_data), "\n")

# Compare overlapping fields
intersect(names(csv_data), names(ris_data))
```

Key insight: RIS files typically include structured reference data and keywords that CSV exports may omit; CSVs are easier to manipulate in R.

3. Standardization function. Write a reusable function to normalise columns.

```
standardise_biblio <- function(df) {
 tibble(
   doi       = df$DI %||% NA_character_,
   title     = df$TI %||% df$Title %||% NA_character_,
   authors   = df$AU %||% NA_character_,
   year      = as.integer(df$PY %||% df$Year),
   journal   = df$SO %||% df$`Source title` %||% NA_character_,
   cited_by  = as.integer(df$TC %||% df$`Cited by` %||% 0L)
 )
}
```

```
# Usage:
# standardise_biblio(ris_data)
```

Key insight: A standardisation layer decouples downstream analysis from the input format, improving reproducibility.

Chapter 7 — Building Reproducible Corpora

1. DOI coverage by year. Compute the DOI coverage rate per publication year.

```
library(tidyverse)

works |>
 group_by(publication_year) |>
 summarise(doi_rate = mean(!is.na(doi))) |>
 ggplot(aes(publication_year, doi_rate)) +
 geom_line() +
 geom_point() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "DOI coverage by year", x = "Year", y = "Coverage")
```

Key insight: DOI coverage tends to improve for recent years as DOI assignment became standard practice after the mid-2000s.

2. Name variants. Detect potential author name duplicates.

```
library(stringdist)

author_names <- works |>
 unnest(author) |>
 distinct(au_id, au_display_name) |>
 pull(au_display_name)

# Find pairs with low string distance
combos <- combn(author_names[1:200], 2, simplify = FALSE)
similar <- map(combos, \(pair) {
 d <- stringdist(pair[1], pair[2], method = "jw")
 if (d < 0.15) tibble(name1 = pair[1], name2 = pair[2], dist = d)
 else NULL
}) |> compact() |> list_rbind()

similar |> arrange(dist) |> head(20)
```

Key insight: Common heuristics include Jaro-Winkler distance, initial matching, and shared co-author overlap. OpenAlex author IDs help but are not always correct.

3. ROR lookup. Use the ROR API to resolve institutions.

```
library(httr2)

top_insts <- works |>
 unnest(authorships) |>
 unnest(institutions) |>
 count(display_name, sort = TRUE) |>
 head(5)

ror_lookup <- function(name) {
 resp <- request("https://api.ror.org/organizations") |>
   req_url_query(query = name) |>
   req_perform() |>
   resp_body_json()
 if (length(resp$items) > 0)
   tibble(name = name, ror_id = resp$items[[1]]$id,
          ror_name = resp$items[[1]]$name)
 else
   tibble(name = name, ror_id = NA, ror_name = NA)
}

map(top_insts$display_name, ror_lookup) |> list_rbind()
```

Key insight: ROR provides stable, disambiguated identifiers; matching by name alone can produce false positives for common institutional names.

Chapter 8 — Working with Full Text

1. Extract and compare. Download a PDF, extract text, and compare the abstract.

```
library(pdftools)
library(tidyverse)

pdf_text_raw <- pdf_text("sample_paper.pdf")
extracted_abstract <- str_extract(
 paste(pdf_text_raw, collapse = "\n"),
 "(?i)abstract\\s*\\.?\\s*(.+?)(?=\\n\\
  ↪  s*(introduction|keywords|1\\.))",
)
```

```
# Compare with OpenAlex abstract
oa_abstract <- works |> filter(doi == "10.xxxx/example") |>
 ↪  pull(ab)

cat("Extracted:\n", str_trunc(extracted_abstract, 300), "\n")
cat("OpenAlex:\n", str_trunc(oa_abstract, 300), "\n")
```

Key insight: PDF extraction often introduces line-break artefacts and header/ footer noise. The OpenAlex abstract (from publisher metadata) is typically cleaner.

2. Section detection. Write a function to split extracted text into sections.

```
split_sections <- function(text) {
 lines <- str_split(text, "\n")[[1]]
 # Match numbered headings like "1. Introduction" or "2 Methods"
 heading_idx <- str_which(lines, "^\\s*\\d+\\.?\\s+[A-Z]")

 if (length(heading_idx) < 2) return(list(full_text = text))

 sections <- map2(heading_idx, c(heading_idx[-1] - 1,
 ↪  length(lines)),
                  \(start, end) {
   list(
     heading = str_trim(lines[start]),
     text = paste(lines[(start + 1):end], collapse = "\n")
   )
 })
 set_names(
   map_chr(sections, "text"),
   map_chr(sections, "heading")
 )
}

sections <- split_sections(paste(pdf_text_raw, collapse = "\n"))
names(sections)
```

Key insight: Heading detection is fragile; two-column layouts, supplementary materials, and non-standard numbering all break simple regex patterns.

3. Table extraction. Extract a table from a PDF using `tabulizer`.

```
# tabulapdf needs Java (rJava); shown for reference, not
 ↪  executed.
# Install: install.packages("tabulapdf"); requires a working
 ↪  Java runtime.
```

```
tables <- extract_tables("sample_paper.pdf", pages = 3)
tbl <- as_tibble(tables[[1]], .name_repair = "unique")

# Typical cleaning steps:
tbl <- tbl |>
 row_to_names(row_number = 1) |>   # janitor::row_to_names
 mutate(across(where(is.character), str_trim)) |>
 filter(if_any(everything(), \(x) x != ""))
```

Key insight: Extracted tables almost always need header row promotion, whitespace trimming, and removal of empty or merged-cell artefact rows.

Chapter 9 — Descriptive Bibliometrics

1. Lotka's law. Fit Lotka's law using `bibliometrix`.

```
library(bibliometrix)
library(tidyverse)

bib_df <- convert2df(works)  # or use the already-converted
 ↪  object
lotka_result <- lotka(biblioAnalysis(bib_df))

# Plot observed vs expected
plot(lotka_result)
```

Key insight: If the Kolmogorov-Smirnov test p-value is above 0.05, the observed distribution is consistent with the theoretical inverse-square law.

2. Bradford's law. Identify core, mid, and peripheral journal zones.

```
library(openalexR)
library(bibliometrix)

ml_works <- fetch_openalex(entity = "works", search = "machine
 ↪  learning",
                    from_publication_date = "2020-01-01",
                    to_publication_date = "2023-12-31") |>
 slice_sample(n = 500)

bib_df <- convert2df(ml_works)
brad <- bradford(biblioAnalysis(bib_df))
brad$table |> head(20)
```

Key insight: The core zone typically contains very few journals that account for a disproportionate share of articles — Bradford's law in action.

3. Three-field exploration. Use country instead of author name on the left field.

```
library(bibliometrix)

threeFieldsPlot(bib_df, fields = c("AU_CO", "SO", "DE"),
                n = c(10, 10, 10))
```

Key insight: Country-level Sankey diagrams reveal geographic specialisation patterns: certain countries cluster around specific journals and keywords.

4. Annual growth rate. Compute the compound annual growth rate (CAGR).

```
annual <- works |>
 count(publication_year) |>
 filter(publication_year >= 2010, publication_year <= 2023)

first_year <- annual$n[1]
last_year  <- annual$n[nrow(annual)]
n_years    <- nrow(annual) - 1

cagr <- (last_year / first_year)^(1 / n_years) - 1
cat(sprintf("CAGR: %.1f%%\n", cagr * 100))
```

Key insight: A positive CAGR indicates the field is growing; compare against the global average (~4-5% for all scientific literature) to assess relative growth.

Chapter 10 — Productivity and Impact Indicators

1. Career h-index. Compute h-index, g-index, and m-quotient for a specific author.

```
library(openalexR)
library(tidyverse)

author_works <- fetch_openalex(
 entity = "works",
 author.id = "A5023888391",  # replace with target author
 per_page = 200
```

```
)

cites <- sort(author_works$cited_by_count, decreasing = TRUE)

# h-index
h <- sum(cites >= seq_along(cites))

# g-index
cum_cites <- cumsum(cites)
g_candidates <- which(cum_cites >= seq_along(cites)^2)
g <- if (length(g_candidates)) max(g_candidates) else 0L

# m-quotient
career_years <-
 ↪  max(lubridate::year(author_works$publication_date)) -
             min(lubridate::year(author_works$publication_date))
                ↪  + 1
m <- h / career_years

tibble(h_index = h, g_index = g, m_quotient = round(m, 2))
```

Key insight: The h-index rewards consistency, the g-index rewards highly cited papers, and the m-quotient adjusts for career length.

2. Self-citation sensitivity. Pseudocode for removing self-citations from the h-index.

```
# Pseudocode:
# 1. For each paper by the author, fetch its citing works
# 2. Check if any citing work shares the same author ID
# 3. Count only non-self-citations
# 4. Recompute h-index with adjusted citation counts
#
# Implementation sketch (expensive API calls):
# author_id <- "A5023888391"
# adjusted_cites <- map_int(author_works$id, \(work_id) {
#   citing <- fetch_openalex(entity = "works", cites = work_id)
#   citing |>
#     unnest(author) |>
#     filter(au_id != author_id) |>
#     distinct(id) |>
#     nrow()
# })
# h_adjusted <- sum(sort(adjusted_cites, decreasing = TRUE) >=
#                   seq_along(adjusted_cites))
```

Key insight: Removing citations cannot increase the h-index. Its actual change requires verified author matches and citation records; no universal change is assumed.

3. Empirical resampling of sample ratios. Hold the empirical year reference fixed and resample its observed ratios.

```
set.seed(20260509)
ratios <- works_normalized$ncs[is.finite(works_normalized$ncs)]
stopifnot(length(ratios) > 1L)
ratio_boot <- replicate(1000, mean(sample(ratios, replace =
 ↪  TRUE)))
quantile(ratio_boot, c(0.025, 0.975))
```

Key insight: This interval describes resampling variation conditional on the chosen empirical reference. It ignores uncertainty in that reference and does not establish world-normalized impact or a population confidence interval.

4. Normalization and ranks. Compare the within-sample mean ratio and percentile selection share by year.

```
yearly_indicators <- works_with_pct |>
 group_by(year) |>
 summarise(
   mean_sample_ratio = mean(ncs, na.rm = TRUE),
   ranked_works = sum(!is.na(is_top10)),
   upper_percentile_share = mean(is_top10, na.rm = TRUE),
   .groups = "drop"
 )
yearly_indicators
```

Key insight: The mean ratio is one for each usable year by construction. Ties and small groups affect the percentile share. These sample summaries do not rank field-normalized impact; that requires a justified external reference.

Chapter 11 — Journal Metrics

1. Cohort sensitivity. Use the verified source table and helper from Chapter 11. This code performs a new provider request unless that exact query is already cached.

```
wider_cohort <- fetch_journal_works(journals$source_id[1], 2019,
 ↪  2022)
summarise_journal_records(wider_cohort)
journal_summary |> filter(source_id == journals$source_id[1])
```

The summaries concern cumulative citations at retrieval. A wider publication cohort changes composition and citation exposure; the result need not move in one direction. CiteScore requires citations received within its specified four-year interval and eligible document types in both numerator and denominator. Changing a publication-date filter alone supplies neither requirement.

2. Descriptive association.

```
complete_pairs <- journal_summary |>
  select(mean_cumulative_citations, median_recorded_references)
   ↪  |>
  drop_na()
rho <- if (nrow(complete_pairs) >= 3 &&
           n_distinct(complete_pairs$mean_cumulative_citations)
            ↪  > 1 &&
           n_distinct(complete_pairs$median_recorded_references)
            ↪  > 1) {
  cor(complete_pairs$mean_cumulative_citations,
      complete_pairs$median_recorded_references, method =
 ↪  "spearman")
} else NA_real_
rho
```

The coefficient describes the complete journal pairs only; insufficient or constant observations leave it unavailable. Its sign and magnitude depend on the retrieved data. It neither measures citation potential on the citing side nor demonstrates field normalisation or agreement with SNIP.

3. Missing-data denominator.

```
invented_records <- tibble(
  id = c("example-1", "example-2", "example-3"),
  cited_by_count = c(0, 20, NA_real_),
  referenced_works_count = c(10, NA_real_, 30)
)
summarise_journal_records(invented_records)
summarise_journal_records(mutate(invented_records,
 ↪  cited_by_count = NA_real_))
```

There are three records but two observed citation counts. Their sum is 20 and the observed-count mean is 10. Dividing 20 by all three records would give about 6.67 and implicitly count the missing value as zero. When all citation counts are missing, the observed denominator is zero and the sum, mean and median are reported as missing. A zero observed count is retained as a valid observation.

4. Self-citation design.

Define the publication cohort and citation-receipt window first. Obtain the citing–cited work links within that scope and identify each citing work's journal using validated source IDs. Count each citation edge once. A journal self-citation share can then use within-journal citation edges in the numerator and all observed citation edges to the cohort in the denominator, with missing source identities reported separately. Distinguish a genuinely empty eligible edge set from a failed request; a zero denominator makes the share undefined.

An average of per-paper self-citation shares weights papers equally and generally differs from the edge-level share. Neither current cumulative counts nor references recorded for the cited papers identify the journals that cited them. No fixed self-citation threshold or empirical result follows from this chapter's data.

Chapter 12 — Author-Level Analysis

1. Profile comparison. Compare two researchers on multiple indicators.

```
library(openalexR)
library(tidyverse)

ids <- c("A5023888391", "A5048491430")  # two author IDs

profiles <- map(ids, \(aid) {
 w <- fetch_openalex(entity = "works", author.id = aid)
 cites <- sort(w$cited_by_count, decreasing = TRUE)
 tibble(
   author_id = aid,
   n_papers  = nrow(w),
   h_index   = sum(cites >= seq_along(cites)),
   mean_cites = mean(w$cited_by_count)
 )
}) |> list_rbind()

profiles
```

Key insight: An author with many papers but a modest h-index publishes frequently but with less consistent impact; high mean citations suggest a few highly cited papers.

2. ORCID validation. Compare works fetched by OpenAlex ID vs. ORCID filtering.

```
oa_by_id <- fetch_openalex(entity = "works",
                     author.id = "A5023888391")

oa_by_orcid <- fetch_openalex(entity = "works",
                        author.orcid = "0000-0001-2345-6789")

cat("By ID:", nrow(oa_by_id), "works\n")
cat("By ORCID:", nrow(oa_by_orcid), "works\n")

# Find discrepancies
only_id <- setdiff(oa_by_id$id, oa_by_orcid$id)
only_orcid <- setdiff(oa_by_orcid$id, oa_by_id$id)
cat("Only in ID set:", length(only_id), "\n")
cat("Only in ORCID set:", length(only_orcid), "\n")
```

Key insight: Discrepancies arise from incomplete ORCID claiming, name disambiguation errors, and delayed metadata updates.

3. Name variant detection. Search for a common name and assess disambiguation.

```
wang_wei <- fetch_openalex(entity = "authors",
                     search = "Wang Wei")

nrow(wang_wei)  # many entities

wang_wei |>
 select(id, display_name, works_count, cited_by_count) |>
 arrange(desc(works_count)) |>
 head(20)

# Heuristics for identifying the correct author:
# - Match institutional affiliation
# - Check ORCID linkage
# - Verify co-author overlap with known collaborators
# - Inspect topic/concept alignment
```

Key insight: Common names generate hundreds of author entities; institutional affiliation and ORCID are the most reliable disambiguation features.

4. Fractional productivity. Recompute productivity using fractional counting.

```
author_works <- fetch_openalex(entity = "works",
                         author.id = "A5023888391")

fractional <- author_works |>
 mutate(n_authors = map_int(author, nrow),
```

```
        frac_credit = 1 / n_authors) |>
 group_by(publication_year) |>
 summarise(full_count = n(),
           frac_count = sum(frac_credit))

ggplot(fractional, aes(publication_year)) +
 geom_line(aes(y = full_count, colour = "Full")) +
 geom_line(aes(y = frac_count, colour = "Fractional")) +
 labs(title = "Full vs. fractional productivity",
      x = "Year", y = "Papers", colour = "Counting")
```

Key insight: Fractional counting penalises hyper-authored papers. Authors who collaborate broadly see a larger gap between full and fractional counts.

5. Self-citation ratio. Estimate the fraction of self-citations for an author.

```
target_author <- "A5023888391"

# Sample a subset of the author's works
sample_works <- author_works |> slice_sample(n = 10)

self_cite_frac <- map_dbl(sample_works$id, \(wid) {
 citing <- tryCatch(
   fetch_openalex(entity = "works", cites = wid),
   error = \(e) tibble()
 )
 if (nrow(citing) == 0) return(NA_real_)
 citing |>
   unnest(author) |>
   summarise(has_self = any(au_id == target_author)) |>
   pull(has_self) |>
   mean()
})

cat(sprintf("Estimated self-citation rate: %.1f%%\n",
            mean(self_cite_frac, na.rm = TRUE) * 100))
```

Key insight: Self-citation rates of 5-15% are typical; rates above 25% may signal strategic self-citation behaviour.

Chapter 13 — Institutional and Country Analysis

1. Country-level analysis. Aggregate by country using full and fractional counts.

```
library(tidyverse)

country_counts <- works |>
 unnest(authorships) |>
 unnest(countries) |>
 rename(country = countries) |>
 mutate(n_countries = n_distinct(country), .by = id) |>
 summarise(
   full_count = n_distinct(id),
   frac_count = sum(1 / n_countries),
   .by = country
 ) |>
 arrange(desc(full_count)) |>
 head(10)

country_counts |>
 mutate(ratio = full_count / frac_count) |>
 ggplot(aes(reorder(country, full_count), ratio)) +
 geom_col() +
 coord_flip() +
 labs(title = "Full / fractional count ratio by country",
      x = NULL, y = "Ratio")
```

Key insight: Countries with high international collaboration rates (e.g., Switzerland, Singapore) show a larger ratio, meaning full counting inflates their apparent output.

2. PP(top 10%) by institution. Compute the proportion of papers in the top 10% most cited.

```
# Compute year-specific 90th percentile thresholds
thresholds <- works |>
 group_by(publication_year) |>
 summarise(p90 = quantile(cited_by_count, 0.9))

inst_top10 <- works |>
 unnest(authorships) |>
 unnest(institutions) |>
 left_join(thresholds, by = "publication_year") |>
 mutate(is_top10 = cited_by_count >= p90) |>
 summarise(
   pp_top10 = mean(is_top10),
   mncs = mean(cited_by_count),
   .by = display_name
 ) |>
 filter(!is.na(display_name)) |>
 arrange(desc(pp_top10))

ggplot(inst_top10 |> head(20), aes(mncs, pp_top10)) +
```

```
 geom_point() +
 ggrepel::geom_text_repel(aes(label = display_name), size = 3) +
 labs(x = "MNCS", y = "PP(top 10%)")
```

Key insight: PP(top 10%) and MNCS generally correlate but can diverge if an institution has a few extremely highly cited outliers.

3. Collaboration intensity. Relate the number of institutional affiliations per paper to counting method effects.

```
collab_intensity <- works |>
 unnest(authorships) |>
 unnest(institutions) |>
 summarise(
   n_insts = n_distinct(display_name),
   .by = id
 )

inst_collab <- works |>
 unnest(authorships) |>
 unnest(institutions) |>
 left_join(collab_intensity, by = "id") |>
 summarise(
   mean_collab = mean(n_insts),
   full_count = n_distinct(id),
   frac_count = sum(1 / n_insts),
   ratio = full_count / frac_count,
   .by = display_name
 ) |>
 filter(!is.na(display_name))

ggplot(inst_collab, aes(mean_collab, ratio)) +
 geom_point(alpha = 0.5) +
 geom_smooth(method = "lm") +
 labs(x = "Mean affiliations per paper",
      y = "Full / fractional ratio",
      title = "Collaboration intensity vs. counting effect")
```

Key insight: There is a positive relationship — institutions with higher collaboration intensity are more inflated by full counting.

Chapter 14 — Temporal Analysis and Citation Aging

1. Half-life comparison. Compare citation half-lives between a mathematics and a biomedical journal.

```
library(openalexR)
library(tidyverse)

fetch_half_life <- function(journal_id, label) {
 w <- fetch_openalex(entity = "works",
                primary_location.source.id = journal_id,
                from_publication_date = "2015-01-01",
                to_publication_date = "2018-12-31")
 w |>
   mutate(age = 2024 - publication_year) |>
   group_by(age) |>
   summarise(mean_cites = mean(cited_by_count)) |>
   mutate(cum_share = cumsum(mean_cites) / sum(mean_cites),
          journal = label)
}

math_hl   <- fetch_half_life("S154589771", "Mathematics")
biomed_hl <- fetch_half_life("S49861241", "Biomedicine")

bind_rows(math_hl, biomed_hl) |>
 ggplot(aes(age, cum_share, colour = journal)) +
 geom_line() +
 geom_hline(yintercept = 0.5, linetype = "dashed") +
 labs(title = "Citation half-life comparison",
      x = "Age (years)", y = "Cumulative citation share")
```

Key insight: Biomedical journals typically have shorter half-lives (3-5 years) than mathematics journals (8-12 years) because of faster publication cycles.

2. Price index computation. Compute the fraction of references less than 5 years old.

```
# For papers with reference data available
sample_papers <- works |>
 filter(publication_year == 2020) |>
 slice_sample(n = 20)

price_indices <- map_dbl(sample_papers$id, \(wid) {
 refs <- tryCatch(
   fetch_openalex(entity = "works", cited_by = wid),
   error = \(e) tibble()
 )
 if (nrow(refs) == 0) return(NA_real_)
 pub_year <- sample_papers$publication_year[sample_papers$id ==
 ↪  wid]
 mean(refs$publication_year >= (pub_year - 5), na.rm = TRUE)
})

cat(sprintf("Mean Price index: %.2f\n", mean(price_indices,
 ↪  na.rm = TRUE)))
```

Key insight: A high Price index indicates the field relies on recent work (fast-moving); a low Price index suggests reliance on older, established literature.

3. Sleeping beauty detection. Identify delayed-recognition papers.

```
sleeping <- works |>
 filter(publication_year <= 2018,
        cited_by_count >= 20) |>
 mutate(age = 2024 - publication_year)

# Approximate: papers with low early citation rate but high total
# (exact per-year citation data requires counts_by_year from
 ↪  OpenAlex)
sleeping_candidates <- sleeping |>
 filter(
   map_lgl(counts_by_year, \(cy) {
     if (is.null(cy)) return(FALSE)
     early <- cy |> filter(year <= publication_year + 3)
     sum(early$cited_by_count, na.rm = TRUE) < 2
   })
 )

sleeping_candidates |>
 select(display_name, publication_year, cited_by_count) |>
 arrange(desc(cited_by_count))
```

Key insight: Sleeping beauties are rare but occur across all fields; they often represent ahead-of-their-time ideas that gained traction only after a paradigm shift.

4. Median vs. mean aging. Replot aging curves using median citations.

```
aging_comparison <- works |>
 mutate(age = 2024 - publication_year) |>
 group_by(age) |>
 summarise(
   mean_cites   = mean(cited_by_count),
   median_cites = median(cited_by_count)
 ) |>
 pivot_longer(c(mean_cites, median_cites),
              names_to = "stat", values_to = "value")

ggplot(aging_comparison, aes(age, value, colour = stat)) +
 geom_line() +
 geom_point() +
 labs(title = "Mean vs. median citation aging",
      x = "Age (years)", y = "Citations")
```

Key insight: The median curve is lower and flatter because the mean is pulled up by highly cited outliers. Median is more robust for skewed distributions.

5. Publication-cohort sensitivity. Use the Scientometrics sample fetched in Chapter 14. Its publication dates span 2013–2018; the following cohorts end in 2018 and remain within that retrieved range. These are current cumulative citation summaries for subsets of a sample.

```
cohort_end <- 2018
cohort_lengths <- c(2, 3, 5)
cohort_summaries <- map_dfr(cohort_lengths,
 ↪  function(length_years) {
  cohort <- works_sciento |>
    mutate(publication_year = lubridate::year(publication_date))
     ↪  |>
    filter(publication_year >= cohort_end - length_years + 1,
           publication_year <= cohort_end)
  observed <-
 ↪  cohort$cited_by_count[!is.na(cohort$cited_by_count)]
  tibble(cohort_years = length_years, n_records = nrow(cohort),
         n_citations_observed = length(observed),
         mean_cumulative_citations = if (length(observed))
          ↪  mean(observed) else NA_real_)
})
cohort_summaries

ggplot(cohort_summaries, aes(factor(cohort_years),
 ↪  mean_cumulative_citations)) +
  geom_col() +
  labs(x = "Publication-cohort length (years)",
       y = "Mean cumulative citations at retrieval",
       title = "Descriptive sensitivity to publication-cohort
        ↪  selection")
```

Changing the publication cohort changes the included papers and their citation exposure. The mean may rise or fall; it is not guaranteed to increase with cohort length. A publication-date filter does not restrict citation-receipt dates. These calculations therefore do not estimate JIF or compare fixed post-publication citation windows. Missing citation counts remain outside the observed denominator.

Chapter 15 — Co-authorship Networks

1. Fractional counting. Weight edges by 1/(k-1) where k is the number of authors.

```
library(igraph)
library(tidyverse)

edges_frac <- works |>
 unnest(author) |>
 group_by(id) |>
 mutate(k = n()) |>
 filter(k >= 2) |>
 reframe(
   pair = combn(au_id, 2, simplify = FALSE),
   weight = 1 / (k - 1)
 ) |>
 unnest_wider(pair, names_sep = "_") |>
 rename(from = pair_1, to = pair_2)

g_frac <- graph_from_data_frame(edges_frac, directed = FALSE)
E(g_frac)$weight <- edges_frac$weight

degree_frac <- strength(g_frac) |>
 sort(decreasing = TRUE) |>
 head(10)
degree_frac
```

Key insight: Fractional counting reduces the influence of mega-authored papers. Authors primarily involved in large teams may drop in rank.

2. Temporal slicing. Build yearly networks and track structural evolution.

```
yearly_stats <- works |>
 group_by(publication_year) |>
 group_map(\(df, key) {
   edges <- df |>
     unnest(author) |>
     group_by(id) |>
     filter(n() >= 2) |>
     reframe(pair = combn(au_id, 2, simplify = FALSE)) |>
     unnest_wider(pair, names_sep = "_")
   if (nrow(edges) == 0) return(NULL)
   g <- graph_from_data_frame(edges, directed = FALSE)
   tibble(
     year = key$publication_year,
     n_nodes = vcount(g),
     n_components = components(g)$no,
     mean_degree = mean(degree(g))
   )
 }) |> list_rbind()

ggplot(yearly_stats, aes(year, mean_degree)) +
 geom_line() +
```

```
 geom_point() +
 labs(title = "Mean degree over time", x = "Year", y = "Mean
  ↪  degree")
```

Key insight: Mean degree typically increases over time as collaboration becomes more common; the number of components may decrease as the field becomes more connected.

3. Betweenness vs. degree. Compare centrality measures for the giant component.

```
g <- build_coauth_graph(works)
gc <- induced_subgraph(g, components(g)$membership ==
                         which.max(components(g)$csize))

centrality <- tibble(
 author = V(gc)$name,
 deg = degree(gc),
 btw = betweenness(gc, normalized = TRUE)
)

ggplot(centrality, aes(deg, btw)) +
 geom_point(alpha = 0.4) +
 labs(x = "Degree", y = "Betweenness (normalised)",
      title = "Betweenness vs. degree in co-authorship network")
```

Key insight: High-betweenness, low-degree nodes are "bridge" authors who connect otherwise separate groups — they may be interdisciplinary researchers.

4. Resolution parameter. Vary the Leiden resolution and compare community counts.

```
library(igraph)

resolutions <- c(0.5, 1.0, 2.0)

res_results <- map(resolutions, \(r) {
 comm <- cluster_leiden(gc, resolution = r,
                        objective_function = "modularity")
 tibble(resolution = r,
        n_communities = length(comm),
        modularity = modularity(comm))
})

list_rbind(res_results)
```

Key insight: Higher resolution produces more communities. The optimal value depends on the application — lower resolution reveals broad thematic clusters, higher resolution reveals tight collaborative teams.

Chapter 16 — Co-citation and Bibliographic Coupling

1. Cosine normalisation. Normalise co-citation strengths by the geometric mean of individual citation counts.

```
library(igraph)
library(tidyverse)

# Assume cocit_matrix is the raw co-citation matrix
diag_sqrt <- sqrt(diag(cocit_matrix))
cosine_matrix <- cocit_matrix / outer(diag_sqrt, diag_sqrt)
diag(cosine_matrix) <- 0

g_cosine <- graph_from_adjacency_matrix(cosine_matrix,
                                        mode = "undirected",
                                        weighted = TRUE)

# Compare top pairs
top_pairs_raw <- E(g_raw)[order(-E(g_raw)$weight)][1:10]
top_pairs_cos <- E(g_cosine)[order(-E(g_cosine)$weight)][1:10]
```

Key insight: Cosine normalisation down-weights pairs that are individually highly cited, surfacing topically related pairs that raw counts miss.

2. Temporal co-citation. Split the corpus into two periods and build separate networks.

```
periods <- list(
 early = works |> filter(publication_year %in% 2020:2021),
 late  = works |> filter(publication_year %in% 2022:2023)
)

period_graphs <- map(periods, \(df) {
 # Build co-citation network for this period's references
 refs <- df |> unnest(referenced_works)
 ref_pairs <- refs |>
   group_by(id) |>
   filter(n() >= 2) |>
   reframe(pair = combn(referenced_works, 2, simplify = FALSE))
    ↪  |>
```

```
    unnest_wider(pair, names_sep = "_")

  ref_pairs |>
    count(pair_1, pair_2, name = "weight") |>
    graph_from_data_frame(directed = FALSE)
})

# Find stable pairs: edges present in both periods
shared_edges <- E(period_graphs$early) %in%
 ↪  E(period_graphs$late)
```

Key insight: Stable co-citation pairs represent the intellectual core of the field; new pairs emerging in the later period may signal research-front shifts.

3. Bibliographic coupling by year. Track modularity and fragmentation of the intellectual structure.

```
yearly_bc <- works |>
 group_by(publication_year) |>
 group_map(\(df, key) {
   refs <- df |> unnest(referenced_works)
   pairs <- refs |>
     group_by(referenced_works) |>
     filter(n() >= 2) |>
     reframe(pair = combn(id, 2, simplify = FALSE)) |>
     unnest_wider(pair, names_sep = "_") |>
     count(pair_1, pair_2, name = "weight")
   if (nrow(pairs) == 0) return(NULL)
   g <- graph_from_data_frame(pairs, directed = FALSE)
   comm <- cluster_leiden(g, resolution = 1.0)
   tibble(year = key$publication_year,
          n_components = components(g)$no,
          modularity = modularity(comm))
 }) |> list_rbind()

ggplot(yearly_bc, aes(year, modularity)) +
 geom_line() +
 labs(title = "Bibliographic coupling modularity over time")
```

Key insight: Decreasing modularity suggests convergence across subfields; increasing modularity suggests specialisation and fragmentation.

4. Identifying research fronts. Label communities using top title words.

```
library(tidytext)

bc_comm <- cluster_leiden(g_bc, resolution = 1.0)
```

```
V(g_bc)$community <- membership(bc_comm)

front_labels <- works |>
 filter(id %in% V(g_bc)$name) |>
 left_join(
   tibble(id = V(g_bc)$name, community = V(g_bc)$community),
   by = "id"
 ) |>
 unnest_tokens(word, display_name) |>
 anti_join(stop_words, by = "word") |>
 count(community, word, sort = TRUE) |>
 group_by(community) |>
 slice_head(n = 3) |>
 summarise(label = paste(word, collapse = ", "))

front_labels
```

Key insight: Community labels based on frequent title words typically correspond to recognisable subfields or research themes.

Chapter 17 — Co-word and Keyword Co-occurrence

1. Author keywords vs. concepts. Build and compare two co-occurrence networks.

```
library(igraph)
library(tidyverse)

# Author keywords network
kw_pairs <- works |>
 unnest(keywords) |>
 group_by(id) |>
 filter(n() >= 2) |>
 reframe(pair = combn(keyword, 2, simplify = FALSE)) |>
 unnest_wider(pair, names_sep = "_") |>
 count(pair_1, pair_2, name = "weight", sort = TRUE)

g_kw <- graph_from_data_frame(kw_pairs, directed = FALSE)

# OpenAlex concepts network
concept_pairs <- works |>
 unnest(concepts) |>
 group_by(id) |>
 filter(n() >= 2) |>
 reframe(pair = combn(display_name, 2, simplify = FALSE)) |>
 unnest_wider(pair, names_sep = "_") |>
```

```
 count(pair_1, pair_2, name = "weight", sort = TRUE)

g_concept <- graph_from_data_frame(concept_pairs, directed =
 ↪  FALSE)

cat("Author KW network:", vcount(g_kw), "nodes,",
    ecount(g_kw), "edges\n")
cat("Concept network:", vcount(g_concept), "nodes,",
    ecount(g_concept), "edges\n")
```

Key insight: Author keywords are more specific but inconsistent; OpenAlex concepts are more standardised but coarser.

2. Temporal evolution. Track the top keywords by degree over time.

```
yearly_top_kw <- works |>
 unnest(keywords) |>
 group_by(publication_year) |>
 group_map(\(df, key) {
   pairs <- df |>
     group_by(id) |>
     filter(n() >= 2) |>
     reframe(pair = combn(keyword, 2, simplify = FALSE)) |>
     unnest_wider(pair, names_sep = "_")
   if (nrow(pairs) == 0) return(NULL)
   g <- graph_from_data_frame(pairs, directed = FALSE)
   tibble(
     year = key$publication_year,
     keyword = names(sort(degree(g), decreasing = TRUE)[1:5]),
     degree = sort(degree(g), decreasing = TRUE)[1:5]
   )
 }) |> list_rbind()

ggplot(yearly_top_kw, aes(year, degree, colour = keyword)) +
 geom_line() +
 labs(title = "Top keywords by degree over time")
```

Key insight: Some keywords are perennially central; others emerge or fade, signalling evolving research interests.

3. Strategic diagram. Plot clusters on a density-centrality plane.

```
comm <- cluster_leiden(g_kw, resolution = 1.0)
V(g_kw)$community <- membership(comm)

strategic <- map(unique(membership(comm)), \(c) {
 nodes <- V(g_kw)[V(g_kw)$community == c]
 subg <- induced_subgraph(g_kw, nodes)
```

```
  tibble(
    community = c,
    density = edge_density(subg),
    centrality = mean(strength(g_kw, v = nodes))
  )
}) |> list_rbind()

ggplot(strategic, aes(centrality, density,
                      label = community)) +
  geom_point(size = 3) +
  geom_text(nudge_y = 0.01) +
  geom_hline(yintercept = median(strategic$density),
             linetype = "dashed") +
  geom_vline(xintercept = median(strategic$centrality),
             linetype = "dashed") +
  labs(title = "Strategic diagram",
       x = "Centrality (external)", y = "Density (internal)")
```

Key insight: Upper-right quadrant clusters are core themes (high density and centrality); lower-left clusters are peripheral or emerging topics.

4. Normalisation comparison. Compare raw and association-strength-normalised networks.

```
# Association strength: w_ij / (w_i * w_j / 2m)
total_weight <- sum(E(g_kw)$weight)
node_strength <- strength(g_kw)

assoc_edges <- as_data_frame(g_kw) |>
  mutate(
    assoc = weight / (node_strength[from] * node_strength[to] /
                      (2 * total_weight))
  )

g_assoc <- graph_from_data_frame(assoc_edges, directed = FALSE)
E(g_assoc)$weight <- assoc_edges$assoc

# Compare top-centrality keywords
top_raw <- names(sort(strength(g_kw), decreasing = TRUE))[1:10]
top_assoc <- names(sort(strength(g_assoc), decreasing =
  ↪  TRUE))[1:10]

tibble(raw = top_raw, normalised = top_assoc)
```

Key insight: Normalisation reduces the dominance of high-frequency generic terms, elevating more specific and meaningful keyword pairs.

Chapter 18 — Community Detection and Backbone Extraction

1. Alpha sensitivity. Run the disparity filter across multiple alpha values.

```
library(igraph)
library(tidyverse)

alphas <- c(0.01, 0.05, 0.10, 0.20)

alpha_results <- map(alphas, \(a) {
 # Disparity filter: keep edge if p_ij < alpha
 el <- as_data_frame(g) |>
   mutate(
     s_from = strength(g)[from],
     k_from = degree(g)[from],
     p_ij = (1 - weight / s_from)^(k_from - 1)
   ) |>
   filter(p_ij < a)

 g_bb <- graph_from_data_frame(el, directed = FALSE)
 comm <- cluster_leiden(g_bb, resolution = 1.0)
 tibble(alpha = a,
        edges = ecount(g_bb),
        communities = length(comm),
        connected = components(g_bb)$no == 1)
}) |> list_rbind()

alpha_results
```

Key insight: Very low alpha values remove too many edges and disconnect the network; very high values retain noise. Typical values of 0.05-0.10 balance signal and connectivity.

2. Threshold vs. disparity. Compare two backbone extraction methods using NMI.

```
library(igraph)

# Threshold filtering: keep edges above median weight
threshold <- median(E(g)$weight)
g_thresh <- subgraph.edges(g, which(E(g)$weight > threshold))
comm_thresh <- cluster_leiden(g_thresh, resolution = 1.0)

# Disparity filtering (from exercise 1)
comm_disp <- cluster_leiden(g_bb, resolution = 1.0)

# NMI comparison on shared nodes
```

```
shared <- intersect(V(g_thresh)$name, V(g_bb)$name)
m1 <- membership(comm_thresh)[shared]
m2 <- membership(comm_disp)[shared]

compare(m1, m2, method = "nmi")
```

Key insight: The disparity filter preserves locally significant edges (important for low-degree nodes) while the threshold filter favours globally strong connections. NMI values below 0.7 indicate meaningfully different community structures.

3. Hierarchical structure. Check whether communities at different resolutions are nested.

```
resolutions <- c(0.5, 1.0, 2.0)

memberships <- map(resolutions, \(r) {
 comm <- cluster_leiden(g_bb, resolution = r)
 tibble(node = V(g_bb)$name, community = membership(comm),
        resolution = r)
}) |> list_rbind()

# Check nesting: at resolution 2.0, are communities subsets of
# resolution 0.5 communities?
nesting <- memberships |>
 pivot_wider(names_from = resolution, values_from = community,
             names_prefix = "res_") |>
 count(res_0.5, res_2) |>
 group_by(res_2) |>
 mutate(is_nested = n_distinct(res_0.5) == 1)

mean(nesting$is_nested)  # proportion of fine communities nested
```

Key insight: Perfectly nested hierarchies are rare in real networks. Partial nesting suggests a hierarchical thematic structure with some cross- cutting topics.

4. Weighted vs. unweighted community detection. Remove weights and compare results.

```
g_unweighted <- g_bb
E(g_unweighted)$weight <- 1

comm_weighted   <- cluster_leiden(g_bb, resolution = 1.0)
comm_unweighted <- cluster_leiden(g_unweighted, resolution =
 ↪  1.0)

compare(membership(comm_weighted), membership(comm_unweighted),
```

```
        method = "nmi")

cat("Weighted communities:", length(comm_weighted), "\n")
cat("Unweighted communities:", length(comm_unweighted), "\n")
```

Key insight: Removing weights often increases the number of communities because weak connections that helped merge groups are no longer distinguishable from strong ones.

Chapter 19 — Science Mapping and Overlay Maps

1. Multi-institution overlay. Overlay two institutions on the same base map.

```
library(tidyverse)
library(ggrepel)

inst_ids <- c("I123456789", "I987654321")

profiles <- map(inst_ids, \(iid) {
 w <- fetch_openalex(entity = "works",
                authorships.institutions.lineage = iid,
                from_publication_date = "2020-01-01")
 w |>
   unnest(concepts) |>
   count(display_name, name = "n_works") |>
   mutate(inst = iid)
}) |> list_rbind()

ggplot(profiles, aes(display_name, n_works, fill = inst)) +
 geom_col(position = "dodge") +
 coord_flip() +
 labs(title = "Research profile overlay")
```

Key insight: Overlapping areas indicate shared research strengths; divergent areas reveal institutional specialisation.

2. Temporal overlay. Compare the same institution's profile across two time periods.

```
periods <- list(
 early = c("2018-01-01", "2019-12-31"),
 late  = c("2022-01-01", "2023-12-31")
)
```

```
temporal_profiles <- imap(periods, \(dates, label) {
 fetch_openalex(entity = "works",
          authorships.institutions.lineage = "I123456789",
          from_publication_date = dates[1],
          to_publication_date   = dates[2]) |>
   unnest(concepts) |>
   count(display_name, name = "n_works") |>
   mutate(period = label)
}) |> list_rbind()

temporal_profiles |>
 pivot_wider(names_from = period, values_from = n_works,
             values_fill = 0) |>
 mutate(shift = late - early) |>
 arrange(desc(abs(shift))) |>
 head(15)
```

Key insight: Positive shifts indicate growing research areas; negative shifts suggest declining focus or strategic reorientation.

3. Country-level overlay. Compare a small and a large country's publication profiles.

```
countries <- c("SG", "US")  # Singapore vs. United States

country_profiles <- map(countries, \(cc) {
 fetch_openalex(entity = "works",
          authorships.countries = cc,
          from_publication_date = "2022-01-01",
          to_publication_date = "2022-12-31") |>
   slice_sample(n = 500) |>
   unnest(concepts) |>
   count(display_name, name = "n_works") |>
   mutate(share = n_works / sum(n_works),
          country = cc)
}) |> list_rbind()

country_profiles |>
 pivot_wider(names_from = country, values_from = share,
             values_fill = 0) |>
 ggplot(aes(US, SG)) +
 geom_point() +
 geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
 ggrepel::geom_text_repel(aes(label = display_name), size = 2) +
 labs(title = "Country profile: Singapore vs. US")
```

Key insight: Small countries tend to show concentrated profiles (above the diagonal in selected areas); large countries are more evenly distributed.

Chapter 20 — Network Visualization and Interoperability

1. Layout comparison. Apply five layouts and compare quality and speed.

```
library(igraph)
library(ggraph)
library(tidyverse)

layouts <- c("fr", "kk", "drl", "graphopt", "lgl")

layout_results <- map(layouts, \(algo) {
 t <- system.time({
   p <- ggraph(gc, layout = algo) +
     geom_edge_link(alpha = 0.1) +
     geom_node_point(aes(colour = factor(community)), size = 1) +
     labs(title = algo) +
     theme_void()
 })
 list(plot = p, time = t["elapsed"], layout = algo)
})

# Print timing
map(layout_results, \(x) tibble(layout = x$layout, secs =
 ↪  x$time)) |>
 list_rbind()
```

Key insight: Fruchterman-Reingold is a good default; Kamada-Kawai is better for small networks; DRL scales to large graphs but can produce chaotic layouts.

2. Interactive filtering. Create a filterable `visNetwork` visualisation.

```
library(visNetwork)

nodes_df <- tibble(
 id = V(gc)$name,
 label = V(gc)$name,
 group = as.character(V(gc)$community)
)

edges_df <- as_data_frame(gc) |>
 select(from, to)

visNetwork(nodes_df, edges_df) |>
 visOptions(selectedBy = "group",
            highlightNearest = TRUE) |>
 visPhysics(stabilization = FALSE)
```

Key insight: Interactive filtering lets users focus on specific communities and explore inter-community bridges.

3. Gephi round-trip. Export to GraphML for use in Gephi.

```
library(igraph)

# Export
write_graph(gc, "network.graphml", format = "graphml")

# After Gephi processing, compare:
# - Force Atlas 2 layout often reveals community separation
 ↪  better
# - R/ggraph offers more reproducible, scriptable figures
# - Gephi is better for interactive exploration and publication
#   quality figures via SVG/PDF export
```

Key insight: Gephi's Force Atlas 2 often produces cleaner community separation; R workflows are more reproducible and automatable.

4. Colour-blind check. Test colour distinguishability with increasing community counts.

```
library(ggraph)
library(viridis)

n_communities <- c(3, 5, 10)

plots <- map(n_communities, \(k) {
 comm <- cluster_leiden(gc, resolution = k / 5)
 V(gc)$comm_k <- membership(comm)

 ggraph(gc, layout = "fr") +
   geom_edge_link(alpha = 0.05) +
   geom_node_point(aes(colour = factor(comm_k)), size = 2) +
   scale_colour_viridis_d(option = "D") +
   labs(title = paste(k, "communities")) +
   theme_void() +
   theme(legend.position = "none")
})

# Use colorBlindness package or online simulator to test
# library(colorBlindness)
# cvdPlot(plots[[3]])
```

Key insight: Viridis palettes remain distinguishable up to about 8 categories; beyond that, consider shape or faceting as supplementary channels.

Chapter 21 — Text Mining Bibliographic Corpora

1. Bigram analysis. Build a DFM from bigrams.

```
library(quanteda)
library(tidyverse)

corp <- corpus(works, text_field = "ab")
toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE)
 ↪  |>
 tokens_tolower() |>
 tokens_remove(stopwords("en")) |>
 tokens_ngrams(n = 2)

dfm_bi <- dfm(toks) |> dfm_trim(min_termfreq = 5)

topfeatures(dfm_bi, 20)
```

Key insight: Bigrams capture meaningful phrases like “machine_learning” or “citation_analysis” that single words miss.

2. Lexical diversity. Compute the type-token ratio (TTR) per year.

```
library(quanteda)
library(quanteda.textstats)

ttr_by_year <- works |>
 filter(!is.na(ab)) |>
 group_by(publication_year) |>
 summarise(
   all_text = paste(ab, collapse = " "),
   .groups = "drop"
 ) |>
 mutate(
   n_types  = map_int(all_text, \(t) {
     toks <- tokens(t, remove_punct = TRUE) |> tokens_tolower()
     ntype(dfm(toks))
   }),
   n_tokens = map_int(all_text, \(t) {
     toks <- tokens(t, remove_punct = TRUE)
     ntoken(toks)
   }),
   ttr = n_types / n_tokens
 )

ggplot(ttr_by_year, aes(publication_year, ttr)) +
 geom_line() +
 labs(title = "Lexical diversity over time",
      x = "Year", y = "Type-token ratio")
```

Key insight: Declining TTR may indicate vocabulary standardisation; increasing TTR suggests diversifying terminology.

3. Keyness analysis. Identify terms that distinguish a specific year.

```
library(quanteda)
library(quanteda.textstats)

corp <- corpus(works |> filter(!is.na(ab)),
               text_field = "ab",
               docid_field = "id")
docvars(corp, "year") <-
 ↪  works$publication_year[!is.na(works$ab)]

dfm_all <- dfm(tokens(corp, remove_punct = TRUE) |>
               tokens_tolower() |>
               tokens_remove(stopwords("en")))

# Compare 2023 vs. all other years
dfm_grouped <- dfm_group(dfm_all, groups = ifelse(
 docvars(dfm_all, "year") == 2023, "2023", "other"
))

keyness <- textstat_keyness(dfm_grouped, target = "2023")
textplot_keyness(keyness, n = 20)
```

Key insight: Keyness analysis surfaces terms unique to a period. For 2023, terms related to LLMs and AI may dominate.

4. Preprocessing sensitivity. Compare analyses with and without stemming.

```
toks_nostem <- tokens(corp, remove_punct = TRUE) |>
 tokens_tolower() |>
 tokens_remove(stopwords("en"))

toks_stem <- tokens_wordstem(toks_nostem)

top_nostem <- topfeatures(dfm(toks_nostem), 20)
top_stem   <- topfeatures(dfm(toks_stem), 20)

tibble(
 rank = 1:20,
 unstemmed = names(top_nostem),
 stemmed = names(top_stem)
)
```

Key insight: Stemming merges variants (e.g., “citations” and “citation”) but can produce unintelligible stems. For keyword-based analyses, unstemmed terms are often more interpretable.

Chapter 22 — Topic Modeling

1. K selection. Fit LDA models with multiple values of K and compare coherence.

```
library(topicmodels)
library(tidyverse)
set.seed(20260509)

ks <- c(5, 8, 12, 15)

models <- map(ks, \(k) {
 LDA(dtm, k = k, method = "Gibbs",
     control = list(seed = 20260509, iter = 1000))
})

# Compute coherence (using top-10 terms per topic)
coherence_scores <- map_dbl(models, \(mod) {
 terms <- terms(mod, 10)
 # Simple PMI-based coherence approximation
 mean(map_dbl(seq_len(ncol(terms)), \(t) {
   top_terms <- terms[, t]
   pairs <- combn(top_terms, 2, simplify = FALSE)
   mean(map_dbl(pairs, \(p) {
    log((slam::col_sums(dtm[, p[1]] > 0 & dtm[, p[2]] > 0) + 1) /
         slam::col_sums(dtm[, p[1]] > 0))
   }))
 }))
})

tibble(k = ks, coherence = coherence_scores) |>
 ggplot(aes(k, coherence)) +
 geom_line() +
 geom_point() +
 labs(title = "Topic coherence by K", x = "K", y = "Coherence")
```

Key insight: The best K balances coherence (higher is better) and interpretability. There is rarely a single "correct" K.

2. STM with journal covariate. Fit a Structural Topic Model with journal as a prevalence covariate.

```
library(stm)
set.seed(20260509)

# Prepare data from two journals
two_journals <- works |>
 filter(so %in% c("Scientometrics", "PLOS ONE"), !is.na(ab))
```

```
processed <- textProcessor(two_journals$ab,
                           metadata = two_journals)
prepped <- prepDocuments(processed$documents,
                         processed$vocab,
                         processed$meta)

stm_fit <- stm(prepped$documents, prepped$vocab,
               K = 10,
               prevalence = ~ so,
               data = prepped$meta,
               seed = 20260509)

effect <- estimateEffect(1:10 ~ so, stm_fit, metadata =
 ↪  prepped$meta)
plot(effect, covariate = "so", topics = 1:10)
```

Key insight: Topics with large journal effects represent disciplinary specialisation; shared topics indicate cross-cutting themes.

3. Topic labelling. Read top documents per topic to assign labels.

```
library(topicmodels)
library(tidyverse)

best_model <- models[[which.max(coherence_scores)]]
gamma <- posterior(best_model)$topics

# For each topic, find the top-5 documents
topic_labels <- map(seq_len(ncol(gamma)), \(t) {
 top_docs <- order(gamma[, t], decreasing = TRUE)[1:5]
 titles <- works$display_name[top_docs]
 tibble(topic = t,
        top_terms = paste(terms(best_model, 5)[, t], collapse =
         ↪  ", "),
        sample_titles = paste(titles, collapse = " | "))
}) |> list_rbind()

topic_labels
# Now assign human-readable labels based on inspection
```

Key insight: Automated topic labels from top terms are a starting point; reading representative documents is essential for meaningful labels.

4. LDA vs. STM comparison. Compare topics from both models.

```
# Extract top terms for each model
lda_terms <- terms(best_model, 10)
```

```
stm_terms <- labelTopics(stm_fit, n = 10)$prob

# Compute Jaccard similarity between matched topics
jaccard <- function(a, b) {
 length(intersect(a, b)) / length(union(a, b))
}

# Best-match each LDA topic to an STM topic
similarities <- map_dbl(seq_len(ncol(lda_terms)), \(i) {
 max(map_dbl(seq_len(nrow(stm_terms)), \(j) {
   jaccard(lda_terms[, i], stm_terms[j, ])
 }))
})

mean(similarities)
```

Key insight: STM topics may differ because the year covariate allows topic prevalence to shift over time, producing more temporally coherent topics.

Chapter 23 — Keyword Extraction

1. Precision and recall. Evaluate RAKE keywords against author keywords.

```
library(udpipe)
library(tidyverse)

sample_papers <- works |>
 filter(!is.na(ab), map_lgl(keywords, \(k) length(k) > 0)) |>
 slice_sample(n = 20)

eval_results <- map(seq_len(nrow(sample_papers)), \(i) {
 paper <- sample_papers[i, ]
 rake_kw <- keywords_rake(
   data.frame(doc_id = paper$id,
              sentence = paper$ab,
              stringsAsFactors = FALSE),
   term = "sentence", group = "doc_id"
 ) |> head(10) |> pull(keyword) |> tolower()

 author_kw <- tolower(unlist(paper$keywords))

 tibble(
   precision = mean(rake_kw %in% author_kw),
   recall = mean(author_kw %in% rake_kw)
 )
}) |> list_rbind()
```

```
eval_results |>
 summarise(across(everything(), list(mean = mean, sd = sd)))
```

Key insight: RAKE precision is typically low (0.1-0.3) because algorithms and authors conceptualise keywords differently. Recall is also limited because author keywords often include broad terms not in the abstract.

2. Bigram keywords. Score bigrams by TF-IDF.

```
library(tidytext)
library(tidyverse)

bigram_tfidf <- works |>
 filter(!is.na(ab)) |>
 unnest_tokens(bigram, ab, token = "ngrams", n = 2) |>
 separate(bigram, c("word1", "word2"), sep = " ") |>
 filter(!word1 %in% stop_words$word,
        !word2 %in% stop_words$word) |>
 unite(bigram, word1, word2, sep = " ") |>
 count(id, bigram) |>
 bind_tf_idf(bigram, id, n) |>
 group_by(bigram) |>
 summarise(mean_tfidf = mean(tf_idf)) |>
 arrange(desc(mean_tfidf))

bigram_tfidf |> head(20)
```

Key insight: Multi-word terms like “citation analysis” and “research evaluation” are invisible to unigram methods but highly informative.

3. Keyword enrichment. Use extracted keywords to build a co-word network for keyword-poor papers.

```
library(igraph)
library(udpipe)

# Papers without author keywords
no_kw_papers <- works |>
 filter(map_lgl(keywords, \(k) length(k) == 0), !is.na(ab))

# Extract keywords via RAKE
extracted <- map(seq_len(nrow(no_kw_papers)), \(i) {
 kw <- keywords_rake(
   data.frame(doc_id = no_kw_papers$id[i],
              sentence = no_kw_papers$ab[i]),
   term = "sentence", group = "doc_id"
 ) |> head(5) |> pull(keyword)
```

```
 tibble(id = no_kw_papers$id[i], keyword = kw)
}) |> list_rbind()

# Build co-word network
pairs <- extracted |>
 group_by(id) |>
 filter(n() >= 2) |>
 reframe(pair = combn(keyword, 2, simplify = FALSE)) |>
 unnest_wider(pair, names_sep = "_") |>
 count(pair_1, pair_2, sort = TRUE)

g_enriched <- graph_from_data_frame(pairs, directed = FALSE)
```

Key insight: Extracted keywords produce a topical structure comparable to author keywords, enabling co-word analysis even for papers without manual keyword assignments.

Chapter 24 — Embeddings for Scientometrics

1. Weighted averaging. Use TF-IDF weights for document embedding computation.

```
library(word2vec)
library(tidytext)
library(tidyverse)
library(umap)

# Compute TF-IDF weights
tfidf <- works |>
 filter(!is.na(ab)) |>
 unnest_tokens(word, ab) |>
 anti_join(stop_words, by = "word") |>
 count(id, word) |>
 bind_tf_idf(word, id, n)

# Get word vectors
wv <- predict(w2v_model, newdata = unique(tfidf$word), type =
 ↪  "embedding")

# Weighted average per document
doc_embeddings <- tfidf |>
 inner_join(
   as_tibble(wv, rownames = "word"),
   by = "word"
 ) |>
 group_by(id) |>
```

```
 summarise(across(starts_with("V"), \(x) weighted.mean(x,
  ↪  tf_idf)))

# UMAP
umap_result <- umap(as.matrix(doc_embeddings |> select(-id)))
```

Key insight: TF-IDF weighting down-weights common words, often producing tighter, more meaningful clusters in the UMAP projection.

2. K-means clustering. Apply k-means to embeddings and compare with LDA topics.

```
set.seed(20260509)

embedding_matrix <- as.matrix(doc_embeddings |> select(-id))

km_results <- map(c(5, 8, 12), \(k) {
 km <- kmeans(embedding_matrix, centers = k, nstart = 25)
 tibble(k = k,
        cluster = km$cluster,
        id = doc_embeddings$id)
}) |> list_rbind()

# Compare k=8 clusters with LDA topics (K=8)
# Assuming lda_topics is a vector of topic assignments
km8 <- km_results |> filter(k == 8) |> pull(cluster)
compare_nmi <- igraph::compare(km8, lda_topics, method = "nmi")
cat(sprintf("NMI between k-means and LDA: %.3f\n", compare_nmi))
```

Key insight: Moderate NMI (0.3-0.6) is typical; the two methods capture overlapping but not identical structure because they model different aspects of the data.

3. Pre-trained embeddings. Use SPECTER embeddings via reticulate.

```
library(reticulate)

# Requires Python with transformers installed
transformers <- import("transformers")
torch <- import("torch")

tokenizer <- transformers$AutoTokenizer$from_pretrained(
 "allenai/specter2"
)
model <- transformers$AutoModel$from_pretrained(
 "allenai/specter2"
)
```

```
abstracts <- works$ab[!is.na(works$ab)][1:100]
inputs <- tokenizer(abstracts, padding = TRUE, truncation = TRUE,
                    return_tensors = "pt", max_length = 512L)
with(torch$no_grad(), {
 outputs <- model(inputs)
})
specter_emb <- outputs$last_hidden_state$mean(dim = 1L)$numpy()
```

Key insight: Pre-trained scientific embeddings like SPECTER capture domain semantics better than generic word2vec, especially for similarity search.

4. Analogy test. Test analogical reasoning in the word2vec model.

```
library(word2vec)

# Test: "journal" - "article" + "book" ≈ ?
analogy <- predict(w2v_model,
                   newdata = c("journal", "article", "book"),
                   type = "nearest", top_n = 5)

# Alternative approach using vector arithmetic
wv <- as.matrix(w2v_model)
if (all(c("journal", "article", "book") %in% rownames(wv))) {
 target <- wv["journal", ] - wv["article", ] + wv["book", ]
 sims <- word2vec_similarity(
   matrix(target, nrow = 1), wv
 )
 sort(sims[1, ], decreasing = TRUE)[1:5]
}
```

Key insight: Domain-specific models may capture field-specific analogies (e.g., “h-index” - “author” + “journal” near “impact factor”) but fail on generic analogies, reflecting the training corpus.

Chapter 25 — Detecting Emerging Topics

1. Relative burst detection. Use proportional frequencies instead of absolute counts.

```
library(tidyverse)

# Compute keyword proportions by year
kw_yearly <- works |>
 unnest(keywords) |>
```

```
 count(publication_year, keyword) |>
 group_by(publication_year) |>
 mutate(proportion = n / sum(n)) |>
 ungroup()

# Apply Kleinberg burst detection on proportions
bursts_relative <- kw_yearly |>
 group_by(keyword) |>
 filter(n() >= 3) |>  # need multiple years
 summarise(
   burst = list(kleinberg_bursts(proportion, publication_year)),
   .groups = "drop"
 ) |>
 unnest(burst) |>
 arrange(desc(strength))

bursts_relative |> head(20)
```

Key insight: Relative frequencies reduce bias toward high-volume years; some terms that burst in absolute counts may be merely riding overall growth.

2. Emergence by journal. Compare emerging keywords across two journals.

```
journals <- c("Scientometrics", "PLOS ONE")

journal_bursts <- map(journals, \(j) {
 kw_j <- works |>
   filter(so == j) |>
   unnest(keywords) |>
   count(publication_year, keyword) |>
   group_by(keyword) |>
   filter(n() >= 3) |>
   summarise(
     burst = list(kleinberg_bursts(n, publication_year))
   ) |>
   unnest(burst) |>
   mutate(journal = j)
}) |> list_rbind()

# Compare top bursting terms
journal_bursts |>
 group_by(journal) |>
 arrange(desc(strength)) |>
 slice_head(n = 10) |>
 select(journal, keyword, strength)
```

Key insight: Different journals serve different research communities; emerg-

ing terms may overlap only for broad methodological advances.

3. Novelty scoring. Compute document novelty using cosine distance to prior-year centroids.

```
library(quanteda)
library(tidyverse)

dfm_yearly <- dfm(tokens(corpus(works, text_field = "ab"),
                         remove_punct = TRUE) |>
                  tokens_tolower() |>
                  tokens_remove(stopwords("en"))) |>
 dfm_tfidf()

years <- sort(unique(works$publication_year))

novelty <- map(years[-1], \(yr) {
 prior <- dfm_subset(dfm_yearly, publication_year < yr)
 current <- dfm_subset(dfm_yearly, publication_year == yr)
 if (nrow(prior) == 0 || nrow(current) == 0) return(NULL)

 centroid <- colMeans(as.matrix(prior))
 current_mat <- as.matrix(current)

 cos_dist <- 1 - (current_mat %*% centroid) /
   (sqrt(rowSums(current_mat^2)) * sqrt(sum(centroid^2)))

 tibble(year = yr, novelty = as.numeric(cos_dist))
}) |> list_rbind()

novelty |>
 group_by(year) |>
 summarise(mean_novelty = mean(novelty, na.rm = TRUE)) |>
 ggplot(aes(year, mean_novelty)) +
 geom_line() +
 labs(title = "Document novelty over time",
      x = "Year", y = "Mean cosine distance to prior centroid")
```

Key insight: Increasing novelty scores suggest the field is exploring new territory; stable scores indicate incremental progress within established paradigms.

4. Validation. Manually validate burst-detected emerging terms.

```
# Select 5 terms identified as "emerging"
emerging_terms <- bursts_relative |>
 arrange(desc(strength)) |>
 head(5) |>
 pull(keyword)
```

```
# For each term, search for corroborating evidence:
# - Is there a review article about this topic?
# - Was it a conference theme in recent years?
# - Does Google Trends show a rising interest?

# Create a validation table
validation <- tibble(
 term = emerging_terms,
 review_found = c(TRUE, TRUE, FALSE, TRUE, FALSE),
 conference_theme = c(TRUE, FALSE, FALSE, TRUE, TRUE),
 google_trend_rising = c(TRUE, TRUE, TRUE, FALSE, TRUE),
 validated = review_found | conference_theme
)

cat(sprintf("False positive rate: %.0f%%\n",
            (1 - mean(validation$validated)) * 100))
```

Key insight: Burst detection has moderate precision; triangulating with qualitative sources reduces false positives and increases confidence.

Chapter 26 — Altmetrics

1. OA citation advantage. Compare citations for OA vs. closed-access papers across journals.

```
library(openalexR)
library(tidyverse)

journal_ids <- c("S4210174107", "S137773608")

oa_advantage <- map(journal_ids, \(jid) {
 w <- fetch_openalex(entity = "works",
                     primary_location.source.id = jid,
                     from_publication_date = "2019-01-01",
                     to_publication_date = "2022-12-31")
 w |>
   group_by(is_oa) |>
   summarise(
     n = n(),
     mean_cites = mean(cited_by_count),
     median_cites = median(cited_by_count)
   ) |>
   mutate(journal = jid)
}) |> list_rbind()

oa_advantage
```

Key insight: The OA citation advantage exists in most fields but varies in magnitude. Self-selection (better papers go OA) may partially explain the effect.

2. Temporal attention. Track citation accumulation over time for early-attention papers.

```
recent_works <- works |>
 filter(publication_year == 2022, !is.null(counts_by_year))

attention_trajectory <- recent_works |>
 unnest(counts_by_year) |>
 group_by(id) |>
 arrange(year) |>
 mutate(cum_cites = cumsum(cited_by_count)) |>
 ungroup()

# Classify: high vs. low early attention (first year citations)
early_cites <- attention_trajectory |>
 filter(year == 2022) |>
 mutate(early_attention = ifelse(cited_by_count >= 5,
                                 "High", "Low"))

attention_trajectory |>
 left_join(early_cites |> select(id, early_attention), by =
  ↪  "id") |>
 group_by(early_attention, year) |>
 summarise(mean_cum = mean(cum_cites, na.rm = TRUE)) |>
 ggplot(aes(year, mean_cum, colour = early_attention)) +
 geom_line() +
 labs(title = "Citation accumulation by early attention")
```

Key insight: Papers with high early attention tend to sustain their advantage, consistent with cumulative advantage (Matthew Effect) in citations.

3. Cross-field comparison. Compare citation distributions and reference list lengths across fields.

```
biomed <- fetch_openalex(entity = "works",
                  primary_location.source.id = "S49861241",
                  from_publication_date = "2020-01-01") |>
 mutate(field = "Biomedicine") |>
 slice_sample(n = 200)

math <- fetch_openalex(entity = "works",
                 primary_location.source.id = "S154589771",
                 from_publication_date = "2020-01-01") |>
 mutate(field = "Mathematics") |>
 slice_sample(n = 200)
```

```
comparison <- bind_rows(biomed, math)

comparison |>
 group_by(field) |>
 summarise(
   mean_cites = mean(cited_by_count),
   median_cites = median(cited_by_count),
   mean_refs = mean(referenced_works_count, na.rm = TRUE)
 )

ggplot(comparison, aes(cited_by_count, fill = field)) +
 geom_histogram(position = "dodge", bins = 30) +
 scale_x_log10() +
 labs(title = "Citation distributions by field")
```

Key insight: Biomedical papers have higher citation counts and longer reference lists than mathematics papers. This is why raw citation counts cannot be compared across fields.

Chapter 27 — Open Science Indicators

1. Cross-journal OA comparison. Compare OA rates across three journals in different disciplines.

```
library(openalexR)
library(tidyverse)

journals <- list(
 biology = "S49861241",
 physics = "S137773608",
 sociology = "S64468286"
)

oa_rates <- imap(journals, \(jid, field) {
 w <- fetch_openalex(entity = "works",
               primary_location.source.id = jid,
               from_publication_date = "2020-01-01",
               to_publication_date = "2023-12-31")
 w |>
   summarise(oa_rate = mean(is_oa, na.rm = TRUE),
             gold_oa = mean(oa_status == "gold", na.rm = TRUE)) |>
   mutate(field = field)
}) |> list_rbind()

oa_rates
```

Key insight: Biology tends to have the highest gold OA rate (driven by PLOS, eLife, etc.), while sociology and mathematics lag behind.

2. Temporal OA growth. Track the year-over-year OA growth rate.

```
oa_growth <- works |>
 filter(is_oa) |>
 count(publication_year, name = "n_oa") |>
 mutate(yoy_growth = (n_oa / lag(n_oa)) - 1)

ggplot(oa_growth, aes(publication_year, yoy_growth)) +
 geom_col() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "Year-over-year OA growth",
      x = "Year", y = "Growth rate")
```

Key insight: OA growth accelerated around 2020, likely driven by COVID-19 mandates and Plan S implementation.

3. OA and collaboration. Test whether author count predicts OA status.

```
works |>
 mutate(n_authors = map_int(author, nrow)) |>
 group_by(n_authors = pmin(n_authors, 10)) |>
 summarise(oa_rate = mean(is_oa, na.rm = TRUE), n = n()) |>
 ggplot(aes(n_authors, oa_rate)) +
 geom_col() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "OA rate by number of authors",
      x = "Authors (capped at 10)", y = "OA rate")
```

Key insight: Larger teams may have more funding to cover APCs. International collaborations also increase OA probability because at least one funder may mandate it.

Chapter 28 — Gender and Equity Analysis

1. Denominator comparison. Use the explicitly synthetic `lookup_records` from Chapter 28.

```
all_records <- lookup_records |>
  count(lookup_category) |>
  mutate(share = n / sum(n))
matched_only <- lookup_records |>
  filter(lookup_category != "Unmatched") |>
```

```
  count(lookup_category) |>
  mutate(share = n / sum(n))
all_records
matched_only
```

The fixture has 12 records: four A, two B and six Unmatched. Among all records, A is one third and B one sixth; among matched records, A is two thirds and B one third. None of these arbitrary category shares measures gender representation.

2. Position audit.

```
position_data |> count(author_position)
position_summary |>
  group_by(author_position) |>
  summarise(proportion_sum = sum(pct), records = sum(n), .groups
   ↪  = "drop")
```

The invented lists produce four first, four last, three middle and one single-author record. Each position's proportions sum to one because unmatched records remain included. Recorded position does not establish a contribution role.

3. Study design.

Specify the scientific question and population before defining an analysis. A real study would need a defensible demographic variable and its provenance, a consent/data-governance plan, categories appropriate to that question, missing-data and validation procedures, and evidence about authorship conventions. The fixture supplies none of those demographic observations, nor real identity linkage, publication years or citation outcomes. Changing a name dictionary cannot recover them. No citation-gap or gender-equity conclusion follows from this exercise.

Chapter 29 — Retraction Analysis

1. Retraction rate by field. Compare retraction rates across fields, normalised by corpus size.

```
library(openalexR)
library(tidyverse)

fields <- c("biomedicine", "computer science")
```

```
retraction_rates <- map(fields, \(f) {
 all_works <- fetch_openalex(entity = "works", search = f,
                       from_publication_date = "2015-01-01",
                       count_only = TRUE)
 retracted <- fetch_openalex(entity = "works", search = f,
                       type = "retraction",
                       from_publication_date = "2015-01-01",
                       count_only = TRUE)
 tibble(
   field = f,
   total = all_works$count,
   retracted = retracted$count,
   rate_per_1000 = retracted$count / all_works$count * 1000
 )
}) |> list_rbind()

retraction_rates
```

Key insight: Retraction rates are typically higher in biomedical fields (1-2 per 1,000) than in computer science, partly reflecting different scrutiny levels and reproducibility challenges.

2. Citation trajectory. Examine when a retracted paper's citations were received.

```
retracted_paper <- fetch_openalex(entity = "works",
                            doi = "10.xxxx/retracted-example")

if (!is.null(retracted_paper$counts_by_year)) {
 retracted_paper |>
   unnest(counts_by_year) |>
   ggplot(aes(year, cited_by_count)) +
   geom_col() +
   geom_vline(xintercept = 2020, linetype = "dashed",
              colour = "red") +  # retraction year
   labs(title = "Citation trajectory of a retracted paper",
        x = "Year", y = "Citations received")
}
```

Key insight: Many retracted papers continue to be cited after retraction, indicating that the retraction notice does not always reach citing authors.

3. Retraction-citation correlation. Test whether pre-retraction citation count predicts post-retraction citations.

```
# Assuming a set of retracted papers with known retraction years
retracted_set <- retracted_works |>
 unnest(counts_by_year) |>
```

```
 mutate(period = ifelse(year < retraction_year, "pre", "post"))
  ↪  |>
 group_by(id, period) |>
 summarise(cites = sum(cited_by_count), .groups = "drop") |>
 pivot_wider(names_from = period, values_from = cites,
             values_fill = 0)

cor(retracted_set$pre, retracted_set$post, method = "spearman")

ggplot(retracted_set, aes(pre, post)) +
 geom_point(alpha = 0.5) +
 geom_smooth(method = "lm") +
 labs(x = "Pre-retraction citations",
      y = "Post-retraction citations",
      title = "Pre- vs. post-retraction citation counts")
```

Key insight: A positive correlation suggests that highly visible papers continue to be cited after retraction — the "retraction paradox."

Chapter 30 — Funding Analysis

1. Funder co-occurrence. Identify frequently co-occurring funder pairs.

```
library(tidyverse)

funder_pairs <- inner_join(
  select(funder_data, work_id, funder_a = funder_id),
  select(funder_data, work_id, funder_b = funder_id),
  by = "work_id", relationship = "many-to-many"
) |>
  filter(funder_a < funder_b) |>
  distinct(work_id, funder_a, funder_b) |>
  count(funder_a, funder_b, sort = TRUE)

funder_pairs |> head(10)
```

This uses the cleaned, observed funder IDs from Chapter 30 and counts each pair once per sampled work. An empty result means no co-funding pair was observed in this sample; it is not evidence that the funders never collaborate.

2. Funder portfolios. Prepare a bounded comparison for two funders observed in the sample.

```
library(tidyverse)

funder_ids <- head(funder_counts$funder_id, 2)
if (length(funder_ids) < 2L) {
  cat("A two-funder comparison is unavailable in this sample.\n")
} else {
  portfolio_sample <- funder_data |>
    filter(funder_id %in% funder_ids) |>
    select(work_id, funder_id) |>
    distinct() |>
    left_join(select(works, id, display_name, abstract),
              by = c("work_id" = "id"))
  portfolio_sample |> count(funder_id, name = "n_sampled_works")
}
```

Apply the text-processing workflow from Chapter 20 to the available titles and abstracts, reporting text coverage per funder. This corpus contains only the sampled journal's papers, so it is not either funder's full portfolio. Completing the exercise for whole portfolios requires a documented retrieval for validated funder IDs, the same publication window, and verified funding metadata coverage. Do not substitute invented identifiers or infer strategic priorities from a small journal sample.

3. Temporal trends. Track funding acknowledgement coverage over time.

```
funding_trend <- funding_works |>
 mutate(publication_year = lubridate::year(publication_date)) |>
 group_by(publication_year) |>
 summarise(n_sampled = n(),
           n_with_funder = sum(has_funder_metadata),
           funding_rate = mean(has_funder_metadata), .groups =
            ↪  "drop")

ggplot(funding_trend, aes(publication_year, funding_rate)) +
 geom_line() +
 geom_point() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "Observed funder metadata by publication year in
  ↪  the sample",
      x = "Publication year", y = "Share with observed funder
       ↪  IDs")
```

The rate describes metadata observed at retrieval, grouped by publication year. It does not measure the prevalence of financial support or establish how the database's coverage changed historically. Keep the annual sample denominators and report sparse groups and missing publication years.

Chapter 31 — Patent-Paper Linkages

1. Scholarly-citation comparison. Use two journals represented in the Chapter 31 sample. Keep the common sample threshold when comparing them, and report the publication years and observed sample sizes. These are descriptive sample summaries, not journal rankings.

```
library(tidyverse)

selected_journals <- works_citations |>
  filter(!is.na(source_display_name)) |>
  count(source_display_name, sort = TRUE) |>
  slice_head(n = 2) |>
  pull(source_display_name)

stopifnot(length(selected_journals) == 2)
common_p90 <- quantile(works_citations$cited_by_count, 0.9,
 ↪  na.rm = TRUE)

works_citations |>
  filter(source_display_name %in% selected_journals) |>
  group_by(source_display_name, year) |>
  summarise(
    n_sampled = n(),
    n_citation_observed = sum(!is.na(cited_by_count)),
    median_cites = median(cited_by_count, na.rm = TRUE),
    mean_cites = mean(cited_by_count, na.rm = TRUE),
    proportion_at_or_above_common_p90 =
      mean(cited_by_count >= common_p90, na.rm = TRUE),
    .groups = "drop"
  )
```

Record the original retrieval date from the data provenance; a cached result's render date is not its retrieval date. Document field classifications separately before attributing differences to fields: the worked sample does not supply a field classification for this comparison. The common threshold is the sample's 90th percentile of current cumulative scholarly citations; ties may put more than 10% of works at or above it. A separate 90th percentile for each journal would put approximately 10% of each journal above its own threshold by construction. None of these quantities measures patent citations or predicts technological impact.

2. Linkage data specification. The linkage table needs actual source records, with at least these fields:

Field	Purpose
`paper_id`	Stable identifier for the matched scholarly work, such as a DOI or OpenAlex work ID.
`patent_id`	Identifier for the citing patent document, including the patent office.
`patent_family_id`	Documented family identifier for deduplicating related patent documents.
`paper_publication_date`	Starting date for a clearly defined paper-to-patent interval.
`citation_date`	Date associated with the citation in the source, with its meaning documented; if only patent publication date is available, identify it as that proxy.
`source` and `retrieved_at`	Patent-reference source, version or snapshot, and retrieval date.
`raw_reference`	Original reference text or source record identifier for verification.
`match_status`	Verified, ambiguous or unmatched, with the matching method and review evidence retained.

Keep ambiguous and unmatched references in the source table. Review candidate matches against the original bibliographic record, and report matching coverage before restricting an analysis to verified links. Missing identifiers or dates must remain missing rather than being inferred from scholarly citation counts. No patent-reference dataset is supplied by the worked example, so it cannot yield an observed linkage rate or lag distribution.

3. Institutional patent linkage. Define an eligible publication set for each institution using the same years, document types, affiliation rule and data coverage. Deduplicate publications by paper identifier, and specify how shared publications are credited. Use a common patent observation cutoff and, where possible, equal follow-up time for each publication cohort. Count a paper as observed patent-cited only when at least one verified patent-family link falls in that window; repeated documents within a family must not inflate linkage counts.

The observed linked-paper proportion is the number of distinct eligible papers with a verified in-window patent-family link divided by the number of eligible papers. Report the numerator, denominator, source coverage, un-

matched and ambiguous records, and sensitivity to matching and follow-up choices. A paper without a link in the observed source is not proof that no patent cites it.

Without the independently validated patent-reference data, the missing inputs are the citing patent and family identifiers, reference matches, dates and coverage information. Report the analysis as unavailable. The scholarly-citation sample and institutional top-decile shares cannot replace those inputs.

Chapter 32 — Causal Inference in Scientometrics

1. Interaction sensitivity. Use the complete sampled records in `cohort_data` from Chapter 32.

```
unadjusted <- lm(cited_by_count ~ is_oa * recent_cohort, data =
 ↪  cohort_data)
adjusted <- lm(cited_by_count ~ is_oa * recent_cohort +
 ↪  referenced_works_count,
               data = cohort_data)
broom::tidy(unadjusted)
broom::tidy(adjusted)
```

The interaction coefficients describe associations in two conditional-mean specifications. A difference between them indicates specification sensitivity; it does not identify an OA treatment effect or by itself establish the size or direction of omitted-variable bias.

2. Boundary sensitivity.

```
alternative <- cohort_data |> mutate(cohort_2018 = year >= 2018)
alternative_model <- lm(cited_by_count ~ is_oa * cohort_2018 +
                          referenced_works_count, data =
 ↪  alternative)
broom::tidy(alternative_model)
```

There is no identified intervention at either date. This changes an arbitrary publication-cohort partition, so it is a descriptive sensitivity check, not a placebo intervention test. Neither a significant nor a non-significant interaction establishes parallel counterfactual trends; these data contain different cohorts' cumulative citations at retrieval, not annual outcomes around a policy change.

3. Comparison audit.

```
matched |> count(control_id, name = "times_used") |>
 ↪  arrange(desc(times_used))
nrow(oa_papers) - nrow(matched)
```

The original within-year draw can reuse a closed-access paper for multiple OA papers. OA papers from years lacking a closed-access candidate are omitted. Such selection changes the comparison population and does not establish exchangeability. An actual policy study must first define treatment assignment and timing, a target causal effect, outcome measurement over comparable exposure periods, and a defensible comparison design. It also needs relevant pre-treatment covariates and an explicit account of identification and inference assumptions.

Chapter 33 — Reproducibility, RRIDs, and Badges

1. DAS detection. Screen available abstracts for candidate data/code availability mentions, then manually validate candidates against full texts and linked resources.

```
library(tidyverse)

das_keywords <- c("data availability", "code availability",
                  "github\\.com", "zenodo", "figshare",
                  "data are available", "code is available")

das_pattern <- paste(das_keywords, collapse = "|")

works |>
 filter(!is.na(abstract), nzchar(trimws(abstract))) |>
 mutate(candidate_mention = str_detect(tolower(abstract),
  ↪  das_pattern)) |>
 summarise(
   total = n(),
   candidate_mentions = sum(candidate_mention),
   candidate_rate = mean(candidate_mention)
 )
```

Key insight: Abstract screening can miss statements and produce false positives, so the match rate is neither a validated DAS rate nor a guaranteed lower bound. Inspect full texts and validate precision/recall on an annotated sample.

2. Cross-journal comparison. Compare metadata completeness and OA rates across journals.

```
journal_ids <- c("S148561398", "S205292342", "S4210197613")

journal_quality <- map(journal_ids, \(jid) {
 w <- fetch_openalex(entity = "works",
                     primary_location.source.id = jid,
                     from_publication_date = "2020-01-01",
                     options = list(sample = 200, seed = 42))
 stopifnot(is.data.frame(w), nrow(w) > 0)
 tibble(
   journal = jid,
   abstract_rate = mean(!is.na(w$abstract) &
    ↪  nzchar(trimws(w$abstract))),
   oa_observed = sum(!is.na(w$oa_status)),
   oa_rate = mean(w$oa_status != "closed", na.rm = TRUE),
   doi_rate = mean(!is.na(w$doi))
 )
}) |> list_rbind()

journal_quality
```

Key insight: Access status and metadata completeness describe different properties. Inspect actual output and coverage; no association is assumed.

3. Temporal trends. Track recorded access status and abstract coverage by publication cohort.

```
metadata_trends <- works_repro |>
 group_by(year) |>
 summarise(
   abstract_rate = mean(has_abstract),
   oa_rate = mean(oa_status != "closed", na.rm = TRUE),
   .groups = "drop"
 ) |>
 pivot_longer(-year, names_to = "metric",
              values_to = "value")

ggplot(metadata_trends, aes(year, value,
                                colour = metric)) +
 geom_line() +
 scale_y_continuous(labels = scales::percent) +
 labs(title = "Access status and metadata coverage by cohort",
      x = "Year", y = "Proportion")
```

Key insight: These are current recorded properties of publication cohorts, not adoption histories or measures of data/code sharing and reproducibility.

Chapter 34 — Reproducible Pipelines

1. Build a pipeline. Create a `_targets.R` file for a bibliometric workflow.

```
library(targets)

# _targets.R content:
tar_option_set(packages = c("openalexR", "tidyverse"))

list(
 tar_target(raw_data, {
   fetch_openalex(entity = "works", search = "bibliometrics",
             from_publication_date = "2020-01-01")
 }),

 tar_target(clean_data, {
   raw_data |>
     filter(!is.na(doi), !is.na(publication_year)) |>
     select(id, doi, display_name, publication_year,
            cited_by_count, is_oa)
 }),

 tar_target(summary_stats, {
   clean_data |>
     group_by(publication_year) |>
     summarise(n = n(), mean_cites = mean(cited_by_count))
 }),

 tar_target(trend_plot, {
   ggplot(summary_stats, aes(publication_year, n)) +
     geom_col() +
     labs(title = "Publication trend", x = "Year", y = "Works")
 })
)

# Run with: tar_make()
# Verify: tar_visnetwork()
```

Key insight: Changing the summary function only re-runs `summary_stats` and `trend_plot`, not `raw_data` or `clean_data` — this is the power of dependency-based pipelines.

2. renv snapshot. Demonstrate dependency management with renv.

```
# Step 1: Initialise renv
# renv::init()

# Step 2: Install a package
```

```
# install.packages("glue")
# library(glue)

# Step 3: Snapshot
# renv::snapshot()
# This creates/updates renv.lock

# Step 4: Simulate loss --- remove the package
# remove.packages("glue")
# library(glue)  # Error: package not found

# Step 5: Restore
# renv::restore()
# library(glue)  # Works again
```

Key insight: renv creates a project-local library and a lockfile, ensuring exact reproducibility of package versions across machines and time.

3. Branching. Use dynamic branching to analyse multiple journals in parallel.

```
library(targets)

# _targets.R with dynamic branching:
tar_option_set(packages = c("openalexR", "tidyverse"))

list(
 tar_target(journal_ids,
            c("S4210174107", "S137773608", "S49861241")),

 tar_target(journal_data,
   fetch_openalex(entity = "works",
            primary_location.source.id = journal_ids,
            from_publication_date = "2020-01-01"),
   pattern = map(journal_ids)
 ),

 tar_target(journal_summary,
   journal_data |>
     summarise(
       journal = first(so),
       n = n(),
       mean_cites = mean(cited_by_count)
     ),
   pattern = map(journal_data)
 )
)
```

Key insight: Dynamic branching avoids duplicating target definitions;

each journal is processed independently and can run in parallel with `tar_make_future()`.

Chapter 35 — Interactive Dashboards

1. Multi-page dashboard. Create a Quarto dashboard with two pages.

```
# Save as dashboard.qmd:
# ---
# title: "Bibliometric Dashboard"
# format: dashboard
# ---
#
# # Overview
#
# ## Row
#
# ```{r}
# library(plotly)
# plot_ly(summary_data, x = ~year, y = ~n, type = "bar")
# ```
#
# ## Row
#
# ```{r}
# plot_ly(works, x = ~cited_by_count, type = "histogram")
# ```
#
# # Details
#
# ## Row
#
# ```{r}
# library(DT)
# datatable(works |> select(display_name, publication_year,
#                           cited_by_count, is_oa))
# ```
```

Key insight: Quarto dashboards use page-level headings (#) for navigation tabs and row/column-level headings (##) for layout.

2. Linked brushing. Use plotly selection to filter a DT table.

```
library(plotly)
library(crosstalk)
library(DT)
```

```
shared_data <- SharedData$new(works)

bscols(
 plot_ly(shared_data, x = ~publication_year,
         y = ~cited_by_count, type = "scatter",
         mode = "markers") |>
   layout(dragmode = "select"),

 datatable(shared_data,
           options = list(pageLength = 10),
           selection = "multiple")
)
```

Key insight: `crosstalk::SharedData` enables linked brushing between any htmlwidget that supports it, without needing Shiny.

3. Custom tooltips. Add informative hover text to a plotly chart.

```
library(plotly)
library(tidyverse)

tooltip_data <- works |>
 group_by(publication_year) |>
 summarise(n = n(), mean_cites = round(mean(cited_by_count), 1))

plot_ly(tooltip_data,
       x = ~publication_year,
       y = ~n,
       type = "bar",
       text = ~paste0("Year: ", publication_year,
                      "<br>Papers: ", n,
                      "<br>Mean cites: ", mean_cites),
       hoverinfo = "text")
```

Key insight: Custom tooltips using `text` and `hoverinfo = "text"` provide context-rich information on hover, improving dashboard usability.

Chapter 36 — Parameterized Quarto Reports

1. Institutional template. Create a parameterized report for institutional profiles.

```
# Save as inst_profile.qmd:
# ---
```

```
# title: "Institutional Profile"
# params:
#   inst_id: "I123456789"
#   inst_name: "University of Example"
#   start_year: 2020
#   end_year: 2023
# ---
#
# ```{r}
# library(openalexR)
# library(tidyverse)
#
# works <- fetch_openalex(
#   entity = "works",
#   authorships.institutions.lineage = params$inst_id,
#   from_publication_date = paste0(params$start_year, "-01-01"),
#   to_publication_date = paste0(params$end_year, "-12-31")
# )
#
# cat(sprintf("Total publications: %d\n", nrow(works)))
# ```
#
# ## Publication Trend
# ```{r}
# works |>
#   count(publication_year) |>
#   ggplot(aes(publication_year, n)) +
#   geom_col() +
#   labs(title = params$inst_name)
# ```
#
# ## Citation Distribution
# ```{r}
# ggplot(works, aes(cited_by_count)) +
#   geom_histogram(bins = 30) +
#   scale_x_log10()
# ```
#
# ## Top Authors
# ```{r}
# works |> unnest(author) |>
#   count(au_display_name, sort = TRUE) |>
#   head(10)
# ```
```

Key insight: Parameters make the same template reusable for any institution; batch rendering generates a portfolio of reports efficiently.

2. Batch rendering. Render the template for multiple journals.

```
library(quarto)

journals <- list(
 list(journal_id = "S4210174107", journal_name =
  ↪  "Scientometrics"),
 list(journal_id = "S137773608", journal_name = "Nature
  ↪  Physics"),
 list(journal_id = "S49861241", journal_name = "PLOS ONE")
)

walk(journals, \(j) {
 quarto_render(
   "journal_profile.qmd",
   execute_params = list(
     journal_id = j$journal_id,
     journal_name = j$journal_name,
     start_year = 2020,
     end_year = 2023
   ),
   output_file = paste0("report_", j$journal_name, ".html")
 )
})
```

Key insight: Batch rendering with `quarto_render()` and parameterised templates scales to hundreds of reports with minimal code.

3. Conditional content. Add a warning when the sample is too small.

```
# In the parameterized .qmd file:
#
# ```{r}
# #| results: asis
# if (nrow(works) < 50) {
#   cat("<div class="callout callout-warn">\n")
#   cat("This sample contains fewer than 50 papers. ")
#   cat("The citation distribution may not be reliable.\n")
#   cat(":::\n")
# } else {
#   ggplot(works, aes(cited_by_count)) +
#     geom_histogram(bins = 30) +
#     scale_x_log10() +
#     labs(title = "Citation distribution")
# }
# ```
```

Key insight: Conditional logic with `results: asis` enables dynamic content switching — essential for robust automated reports that handle edge cases.

Chapter 37 — Building a Bibliometric Shiny App

1. Add network tab. Extend the app with a co-authorship network visualisation.

```
library(shiny)
library(visNetwork)
library(igraph)
library(tidyverse)

# Add to UI:
# tabPanel("Network",
#   visNetworkOutput("coauth_network", height = "600px")
# )

# Add to server:
# output$coauth_network <- renderVisNetwork({
#   req(data())
#   edges <- data() |>
#     unnest(author) |>
#     group_by(id) |>
#     filter(n() >= 2, n() <= 10) |>
#     reframe(pair = combn(au_display_name, 2, simplify =
 ↪  FALSE)) |>
#     unnest_wider(pair, names_sep = "_") |>
#     count(pair_1, pair_2, name = "weight") |>
#     rename(from = pair_1, to = pair_2)
#
#   g <- graph_from_data_frame(edges, directed = FALSE)
#   comm <- cluster_leiden(g, resolution = 1.0)
#
#   nodes <- tibble(
#     id = V(g)$name, label = V(g)$name,
#     group = as.character(membership(comm))
#   )
#
#   visNetwork(nodes, edges) |>
#     visOptions(highlightNearest = TRUE, selectedBy = "group")
# })
```

Key insight: Limiting the network to papers with 2-10 authors avoids overwhelming the visualisation with hyper-authored papers.

2. Caching. Implement server-side caching for repeated queries.

```
library(shiny)
library(memoise)
```

```
# Create a cached version of the data fetching function
fetch_cached <- memoise(function(journal_id, start_date,
 ↪  end_date) {
 fetch_openalex(
   entity = "works",
   primary_location.source.id = journal_id,
   from_publication_date = start_date,
   to_publication_date = end_date
 )
}, cache = cachem::cache_disk("shiny_cache"))

# In the server:
# data <- reactive({
#   fetch_cached(
#     input$journal_id,
#     paste0(input$start_year, "-01-01"),
#     paste0(input$end_year, "-12-31")
#   )
# })
```

Key insight: Disk-backed caching persists across sessions, so the same query never hits the API twice. Use `cachem::cache_disk()` for production apps.

3. Deployment. Deploy to shinyapps.io.

```
# Step 1: Install rsconnect
# install.packages("rsconnect")

# Step 2: Configure account
# rsconnect::setAccountInfo(
#   name = "your-account",
#   token = "your-token",
#   secret = "your-secret"
# )

# Step 3: Deploy
# rsconnect::deployApp(
#   appDir = ".",
#   appName = "bibliometric-explorer",
#   appTitle = "Bibliometric Explorer"
# )

# Step 4: Test with different inputs
# - Try different journal IDs
# - Test with narrow and wide date ranges
# - Verify that caching works (second load should be instant)
```

Key insight: Test with a variety of inputs before deploying; edge cases

(empty results, very large corpora) should be handled gracefully with validation messages.

4. biblioshiny comparison. Run biblioshiny and compare features.

```
library(bibliometrix)
# biblioshiny()

# Feature comparison table:
# | Feature              | biblioshiny | Custom app |
# |----------------------|-------------|------------|
# | Data sources         | WoS, Scopus | OpenAlex   |
# | Network analysis     | Yes         | Basic      |
# | Topic analysis       | Yes         | No         |
# | Custom indicators    | Limited     | Full       |
# | Parameterization     | No          | Yes        |
# | Deployment           | Local only  | shinyapps  |
# | Reproducibility      | GUI-based   | Code-based |
#
# biblioshiny is comprehensive but not easily customisable;
# a custom app trades breadth for tailored functionality.
```

Key insight: biblioshiny is excellent for exploration but lacks reproducibility and customisation. A custom app is better for production workflows with specific analytical requirements.

Glossary

R. Heller · Date: 2026-05-09 · Last changed: 2026-05-11

Altmetrics. Metrics derived from online activity (social media mentions, news coverage, policy documents) rather than citations. See Chapter 26.

Bibliographic coupling. Two documents are bibliographically coupled when they share one or more references. Strength increases with the number of shared references. See Chapter 16.

Bibliometrics. The quantitative study of scholarly publications, their production, dissemination, and impact. A subfield of scientometrics focused on documents.

Burst detection. A method for identifying terms or topics whose frequency increases sharply in a given time window, signalling emerging research fronts. See Chapter 25.

Citation aging. The temporal pattern of how citations accumulate over time after publication. The citation half-life measures how quickly a paper accrues half its total citations. See Chapter 14.

Citation window. The time period during which citations to a publication are counted. Common choices: 3-year, 5-year, or variable windows.

Co-authorship network. A graph where nodes represent authors and edges connect authors who have co-authored at least one publication. See Chapter 15.

Co-citation. Two documents are co-cited when they are both cited by a third document. Frequently co-cited documents are considered intellectually related. See Chapter 16.

Co-word analysis. Network analysis of the co-occurrence of terms (keywords, title words, or descriptors) within documents. See Chapter 17.

Community detection. Algorithmic identification of densely connected subgroups within a network. Common methods: Leiden, Louvain, modularity optimisation. See Chapter 18.

Disparity filter. A network backbone extraction method that retains statistically significant edges based on local weight distributions. See Chapter 18.

DORA. The San Francisco Declaration on Research Assessment. A set of recommendations for improving research evaluation practices, emphasising that journal-based metrics should not be used as surrogates for individual article quality.

Field normalisation. Adjusting citation counts to account for differences in citation practices across research fields and publication years. See Chapter 13.

Fractional counting. An attribution method where credit for a multi-authored publication is divided equally among contributors (1/n per author or institution), in contrast to full counting. See Chapter 13.

Full counting. An attribution method where each author or institution on a publication receives full credit (count of 1). See Chapter 13.

h-index. A researcher has an h-index of h if h of their papers have each been cited at least h times. See Chapter 10.

Impact factor (JIF). The mean number of citations received in year t by articles published in a journal in years t−1 and t−2. See Chapter 11.

Leiden Manifesto. Ten principles for responsible research metrics, emphasising that quantitative evaluation should support qualitative, expert assessment rather than replace it. See Chapter 3.

Leiden algorithm. A community detection method that improves on Louvain by guaranteeing well-connected communities. See Chapter 18.

MNCS (Mean Normalised Citation Score). The average of field-normalised citation scores across a set of publications. An MNCS above 1.0 indicates above-world-average impact. See Chapter 13.

Network backbone. A reduced network retaining only statistically significant or structurally important edges, obtained by filtering methods such as the disparity filter. See Chapter 18.

OpenAlex. A free, open catalogue of the global research system, covering works, authors, institutions, sources, concepts, funders, and publishers. The default data source for this book. See Chapter 5.

Overlay map. A visualisation technique where a subset of data (e.g., one institution's publications) is projected onto a pre-computed base map of science. See Chapter 19.

PP(top 10%). The proportion of an entity's publications that rank in the top 10% most cited in their field and year. A percentile-based impact indicator. See Chapter 10.

Price index. The proportion of a paper's references that are no older than 5 years, measuring how current the cited literature is. See Chapter 14.

RAKE. Rapid Automatic Keyword Extraction. An unsupervised method that extracts multi-word keywords based on word co-occurrence within phrases delimited by stopwords. See Chapter 23.

Research front. A cluster of recently published, actively cited papers representing the current edge of a research area.

Retraction. The formal withdrawal of a published paper by its authors or publisher due to errors, misconduct, or irreproducibility. See Chapter 29.

Scientometrics. The quantitative study of science and innovation, encompassing bibliometrics, patent analysis, and the measurement of research systems.

STM (Structural Topic Model). A topic model that allows document-level covariates (e.g., publication year, journal) to affect topic prevalence and content. See Chapter 22.

TF-IDF. Term Frequency–Inverse Document Frequency. A weighting scheme that highlights terms distinctive to individual documents within a corpus. See Chapter 21.

Topic modelling. Unsupervised methods for discovering latent thematic structure in collections of text. Common approaches: LDA, STM, BERTopic. See Chapter 22.

UMAP. Uniform Manifold Approximation and Projection. A dimensionality-reduction technique for visualising high-dimensional data (e.g., document embeddings) in 2D. See Chapter 24.

Colophon

R. Heller · Date: 2026-05-11 · Last changed: 2026-06-10

```
#> Built with:

#>   bookdown:  0.48

#>   R:  R version 4.6.1 (2026-06-24)

#>   Build date:  2026-11-22

#>   Commit:  ef2d987

#>
#> Attached packages:

#> - Session info -------------------------------------------
 ↪  -----------------
#>  setting  value
#>  version  R version 4.6.1 (2026-06-24)
#>  os       Ubuntu 24.04.5 LTS
#>  system   x86_64, linux-gnu
#>  ui       X11
#>  language (EN)
#>  collate  C.UTF-8
#>  ctype    C.UTF-8
#>  tz       UTC
#>  date     2026-11-22
#>  pandoc   3.8.3 @ /opt/hostedtoolcache/pandoc/3.8.3/x64/ (via
 ↪  rmarkdown)
#>  quarto   NA
#>
#> - Packages -----------------------------------------------
 ↪  -----------------
#>  package            * version  date (UTC) lib source
```

```
#>  arrow              * 26.0.1.1 2026-11-21 [1] RSPM (R 4.6.0)
#>  bib2df             * 1.1.2.0  2024-05-16 [1] RSPM (R 4.6.0)
#>  bibliometrix       * 5.5.0    2026-08-29 [1] RSPM (R 4.6.0)
#>  dplyr              * 1.2.1    2026-04-03 [1] RSPM (R 4.6.0)
#>  DT                 * 0.34.0   2025-09-02 [1] RSPM (R 4.6.0)
#>  forcats            * 1.0.1    2025-09-25 [1] RSPM (R 4.6.0)
#>  ggplot2            * 4.0.4    2026-11-14 [1] RSPM (R 4.6.0)
#>  ggraph             * 2.2.2    2025-08-24 [1] RSPM (R 4.6.0)
#>  glue               * 1.8.1    2026-04-17 [1] RSPM (R 4.6.0)
#>  gt                 * 1.4.0    2026-10-15 [1] RSPM (R 4.6.0)
#>  igraph             * 2.3.4    2026-09-30 [1] RSPM (R 4.6.0)
#>  lubridate          * 1.9.5    2026-02-04 [1] RSPM (R 4.6.0)
#>  Matrix             * 1.7-5    2026-03-21 [2] CRAN (R 4.6.1)
#>  openalexR          * 3.1.2    2026-10-14 [1] RSPM (R 4.6.0)
#>  pdftools           * 3.9.1    2026-09-01 [1] RSPM (R 4.6.0)
#>  plotly             * 4.12.1   2026-07-22 [1] RSPM (R 4.6.0)
#>  purrr              * 1.2.2    2026-04-10 [1] RSPM (R 4.6.0)
#>  quanteda           * 4.5.1    2026-10-21 [1] RSPM (R 4.6.0)
#>  quanteda.textstats * 0.97.2   2024-09-03 [1] RSPM (R 4.6.0)
#>  rcrossref          * 1.2.1    2025-08-20 [1] RSPM (R 4.6.0)
#>  readr              * 2.2.0    2026-02-19 [1] RSPM (R 4.6.0)
#>  RefManageR         * 1.4.0    2022-09-30 [1] RSPM (R 4.6.0)
#>  shiny              * 1.15.0   2026-11-20 [1] RSPM (R 4.6.0)
#>  stm                * 1.3.10   2026-11-12 [1] RSPM (R 4.6.0)
#>  stringr            * 1.6.0    2025-11-04 [1] RSPM (R 4.6.0)
#>  tibble             * 3.3.1    2026-01-11 [1] RSPM (R 4.6.0)
#>  tidygraph          * 1.3.1    2024-01-30 [1] RSPM (R 4.6.0)
#>  tidyr              * 1.3.2    2025-12-19 [1] RSPM (R 4.6.0)
#>  tidytext           * 0.4.3    2025-07-25 [1] RSPM (R 4.6.0)
#>  tidyverse          * 2.0.0    2023-02-22 [1] RSPM (R 4.6.0)
#>  topicmodels        * 0.2-17   2024-08-14 [1] RSPM (R 4.6.0)
#>  uwot               * 0.2.5    2026-08-29 [1] RSPM (R 4.6.0)
#>  visNetwork         * 2.1.4    2025-09-04 [1] RSPM (R 4.6.0)
#>  word2vec           * 0.4.1    2025-11-27 [1] RSPM (R 4.6.0)
#>
#>  [1] /home/runner/work/_temp/Library
#>  [2] /opt/R/4.6.1/lib/R/library
#>  * -- Packages attached to the search path.
#>
#> ------------------------------------------------------------
 ↪  -----------------
```

LLM assistance (KISSKI's Chat AI service, plus a self-hosted Mistral Small 24B via Ollama/`ollamar`) was used for copyediting, formatting, and scaffolding tasks only — see Acknowledgments.

Citing this Guide

R. Heller · Date: 2026-05-11 · Last changed: 2026-11-09

The suggested citation is:

> **Heller, R.** (2026). Scientometrics in R: Bibliometrics, Network Science, and Research Evaluation (Version 0.1.0). Self-published via GitHub Pages. https://cttir.github.io/scientometrics-in-r/.

Download the reference as BibTeX or .ris.

References

R. Heller · Date: 2026-05-11 · Last changed: 2026-05-11

Bibliography

American Society for Cell Biology (2012). San Francisco declaration on research assessment (DORA). Accessed 2026-05-09.

Aria, M. and Cuccurullo, C. (2017). bibliometrix: An R-tool for comprehensive science mapping analysis. Journal of Informetrics, 11(4):959–975.

Barabási, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002). Evolution of the social network of scientific collaborations. Physica A: Statistical Mechanics and its Applications, 311(3–4):590–614.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In Proceedings of the Third International AAAI Conference on Weblogs and Social Media, pages 361–362.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008.

Bradford, S. C. (1934). Sources of information on specific subjects. Engineering, 137:85–86.

Callon, M., Courtial, J.-P., Turner, W. A., and Bauin, S. (1983). From translations to problematic networks: An introduction to co-word analysis. Social Science Information, 22(2):191–235.

de Solla Price, D. J. (1963). Little Science, Big Science. Columbia University Press, New York.

Egghe, L. (2006). Theory and practise of the g-index. Scientometrics, 69(1):131–152.

Fang, F. C., Steen, R. G., and Casadevall, A. (2012). Misconduct accounts for the majority of retracted scientific publications. Proceedings of the National Academy of Sciences, 109(42):17028–17033.

Fortunato, S. (2010). Community detection in graphs. Physics Reports, 486(3–5):75–174.

Garfield, E. (1955). Citation indexes for science: A new dimension in documentation through association of ideas. Science, 122(3159):108–111.

Glänzel, W. and Schubert, A. (2004). Analysing scientific networks through co-authorship. Handbook of Quantitative Science and Technology Research, pages 257–276.

Goodhart, C. A. E. (1984). Problems of monetary management: The U.K. experience. In Monetary Theory and Practice, pages 91–121. Macmillan, London.

Hicks, D., Wouters, P., Waltman, L., de Rijcke, S., and Rafols, I. (2015). Bibliometrics: The Leiden Manifesto for research metrics. Nature, 520(7548):429–431.

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences, 102(46):16569–16572.

Kessler, M. M. (1963). Bibliographic coupling between scientific papers. American Documentation, 14(1):10–25.

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. Data Mining and Knowledge Discovery, 7(4):373–397.

Larivière, V., Ni, C., Gingras, Y., Cronin, B., and Sugimoto, C. R. (2013). Bibliometrics: Global gender disparities in science. Nature, 504(7479):211–213.

Lotka, A. J. (1926). The frequency distribution of scientific productivity. Journal of the Washington Academy of Sciences, 16(12):317–323.

Merton, R. K. (1968). The Matthew Effect in science. Science, 159(3810):56–63.

Narin, F., Hamilton, K. S., and Olivastro, D. (1997). The increasing linkage between U.S. technology and public science. Research Policy, 26(3):317–330.

Newman, M. E. J. (2001). The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences, 98(2):404–409.

Priem, J., Piwowar, H., and Orr, R. (2022). OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. arXiv preprint.

Priem, J., Taraborelli, D., Groth, P., and Neylon, C. (2010). Altmetrics: A manifesto. altmetrics.org.

Rafols, I., Porter, A. L., and Leydesdorff, L. (2010). Science overlay maps: A new tool for research policy and library management. Journal of the American Society for Information Science and Technology, 61(9):1871–1887.

Roberts, M. E., Stewart, B. M., and Tingley, D. (2019). stm: An R package for structural topic models. Journal of Statistical Software, 91(2):1–40.

Serrano, M. Á., Boguñá, M., and Vespignani, A. (2009). Extracting the multiscale backbone of complex weighted networks. Proceedings of the National Academy of Sciences, 106(16):6483–6488.

Silge, J. and Robinson, D. (2017). Text Mining with R: A Tidy Approach. O'Reilly Media.

Singh, V. K., Singh, P., Karmakar, M., Leta, J., and Mayr, P. (2021). The journal coverage of Web of Science, Scopus and Dimensions: A comparative analysis. Scientometrics, 126:5113–5142.

Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. Journal of the American Society for Information Science, 24(4):265–269.

Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. Scientific Reports, 9:5233.

Visser, M., van Eck, N. J., and Waltman, L. (2021). Large-scale comparison of bibliographic data sources: Scopus, Web of Science, Dimensions, Crossref, and Microsoft Academic. Quantitative Science Studies, 2(1):20–41.

Waltman, L. (2016). A review of the literature on citation impact indicators. Journal of Informetrics, 10(2):365–391.

Waltman, L., van Eck, N. J., and Noyons, E. C. M. (2010). A unified approach to mapping and clustering of bibliometric networks. Journal of Informetrics, 4(4):629–635.

Waltman, L., van Eck, N. J., van Leeuwen, T. N., Visser, M. S., and van Raan, A. F. J. (2011). Towards a new crown indicator: Some theoretical considerations. Journal of Informetrics, 5(1):37–47.

Wilsdon, J., Allen, L., Belfiore, E., Campbell, P., Curry, S., Hill, S., Jones, R., Kain, R., Kerridge, S., Thelwall, M., Tinkler, J., Viney, I., Wouters, P., Hill, J., and Johnson, B. (2015). The Metric Tide: Report of the Independent Review of the Role of Metrics in Research Assessment and Management. SAGE Publications.