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** (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()
```

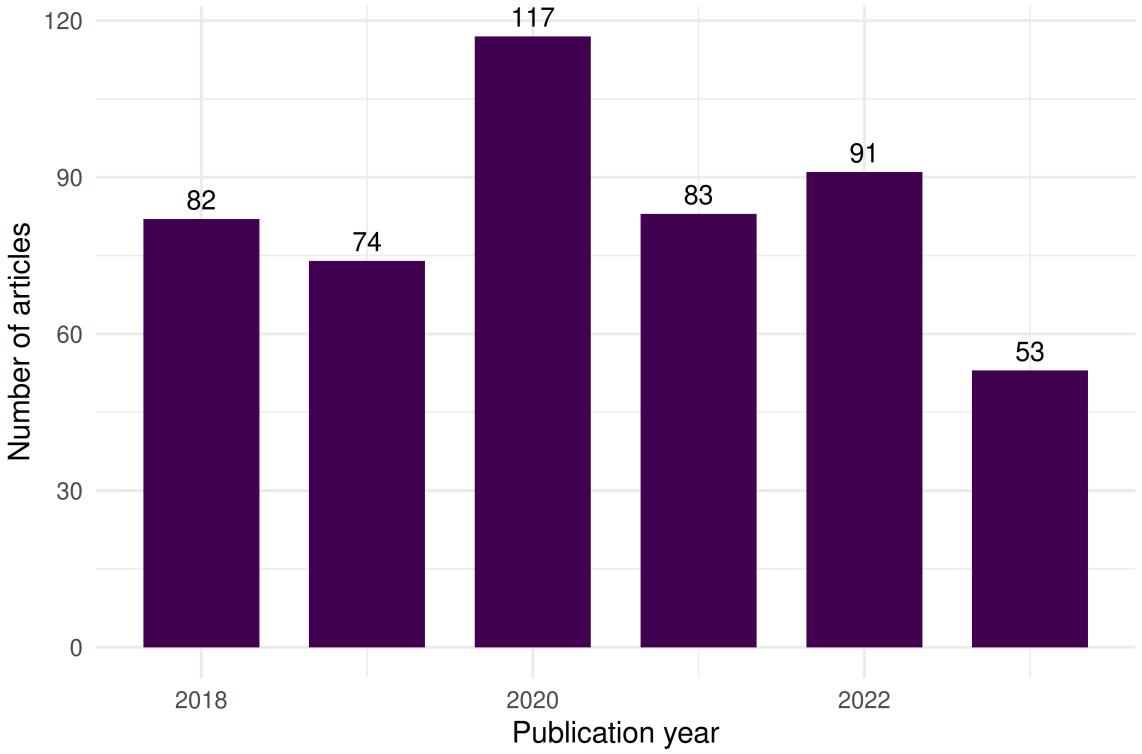


Figure 9.1: Annual scientific production in the sampled corpus (2018-2023).

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()
```

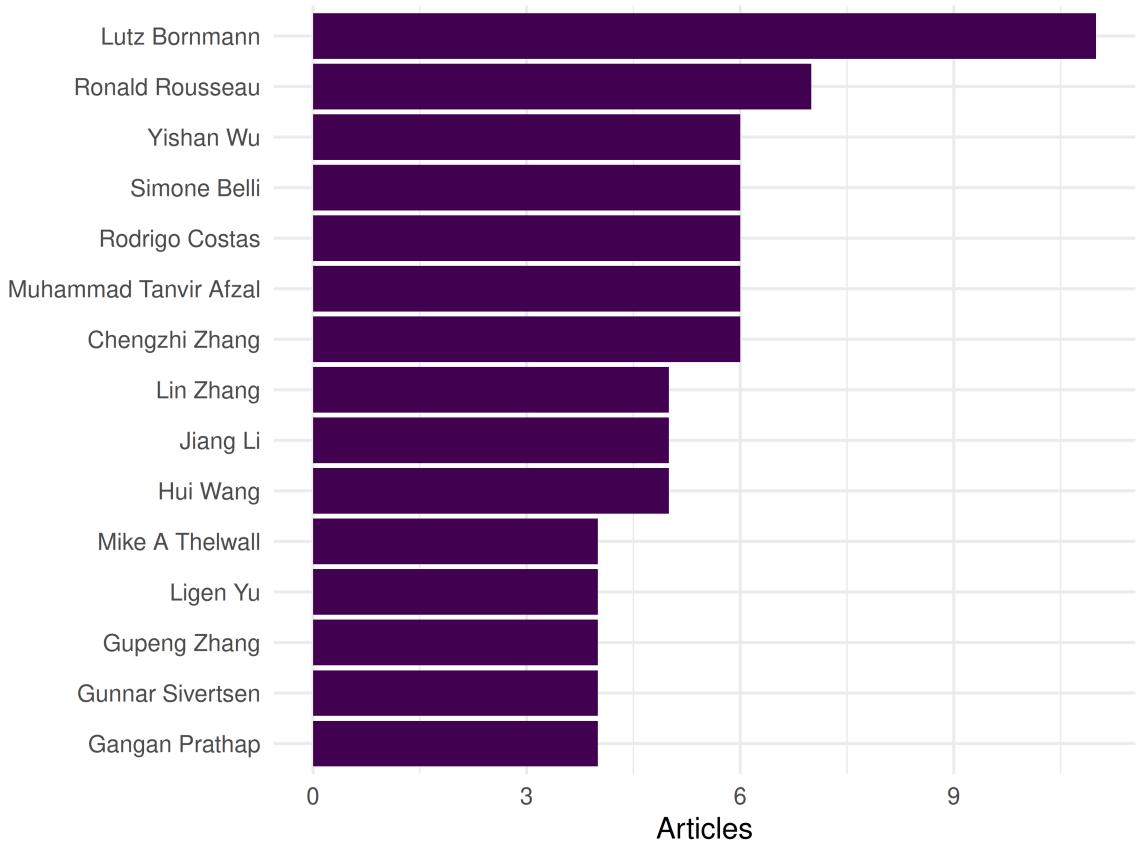


Figure 9.2: Top 15 most productive authors in the sample.

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 × 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.
- 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/libopenblasp-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         vctrs_0.7.3            httpcode_0.3.0
#>  [10] pkgconfig_2.0.3        fastmap_1.2.0          labeling_0.4.3
#>  [13] ca_0.72                utf8_1.2.6             promises_1.5.0
#>  [16] rmarkdown_2.32         tzdb_0.5.0             bit_4.6.0
#>  [19] xfun_0.62              cachem_1.1.0           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               bslib_0.12.0
#>  [28] stringi_1.8.9          RColorBrewer_1.1-3     cellranger_1.1.0
#>  [31] jquerylib_0.1.4        assertthat_0.2.1       Rcpp_1.1.2
#>  [34] bookdown_0.48          knitr_1.52             triebeard_0.4.1
#>  [37] base64enc_0.1-6        rentrez_1.2.4          httpuv_1.6.17
#>  [40] Matrix_1.7-5           igraph_2.3.4           timechange_0.4.0
#>  [43] tidyselect_1.2.1       stringdist_0.9.17      pubmedR_1.1.0
#>  [46] rstudioapi_0.19.0      yaml_2.3.12            viridis_0.6.5
#>  [49] miniUI_0.1.2           humaniformat_0.6.0     curl_8.1.0
#>  [52] qpdf_1.4.1             lattice_0.22-9         plyr_1.8.9
#>  [55] shiny_1.15.0           withr_3.0.3            S7_1.0.0
#>  [58] askpass_1.2.1          evaluate_1.0.5         zip_3.0.2
#>  [61] xml2_1.6.0             shinycssloaders_1.1.0  pillar_1.11.1
#>  [64] janeaustenr_1.0.0      DT_0.34.0              plotly_4.12.1
#>  [67] generics_0.1.4         rprojroot_2.1.1        hms_1.1.4
#>  [70] scales_1.4.0           xtable_1.8-8           contentanalysis_1.1.1
#>  [73] tools_4.6.1            data.table_1.18.6.1    tokenizers_0.3.0
#>  [76] openxlsx_4.2.9         XML_3.99-0.25          fs_2.1.0
#>  [79] visNetwork_2.1.4       fastmatch_1.1-8        grid_4.6.1
#>  [82] bibtex_0.5.3           urltools_1.7.3.1       rscopus_0.9.0
#>  [85] dimensionsR_0.0.3      bibliometrixData_0.3.0 cli_3.6.6
#>  [88] viridisLite_0.4.3      downlit_0.4.5          gtable_0.3.6
#>  [91] sass_0.5.0             digest_0.6.39          ggrepel_0.9.8
#>  [94] crul_1.6.0             htmlwidgets_1.6.4      farver_2.1.2
#>  [97] memoise_2.0.1          htmltools_0.5.9        lifecycle_1.0.5
```

```
#> [100] httr_1.4.9             here_1.0.2             mime_0.13
#> [103] bit64_4.8.6
```