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)
```

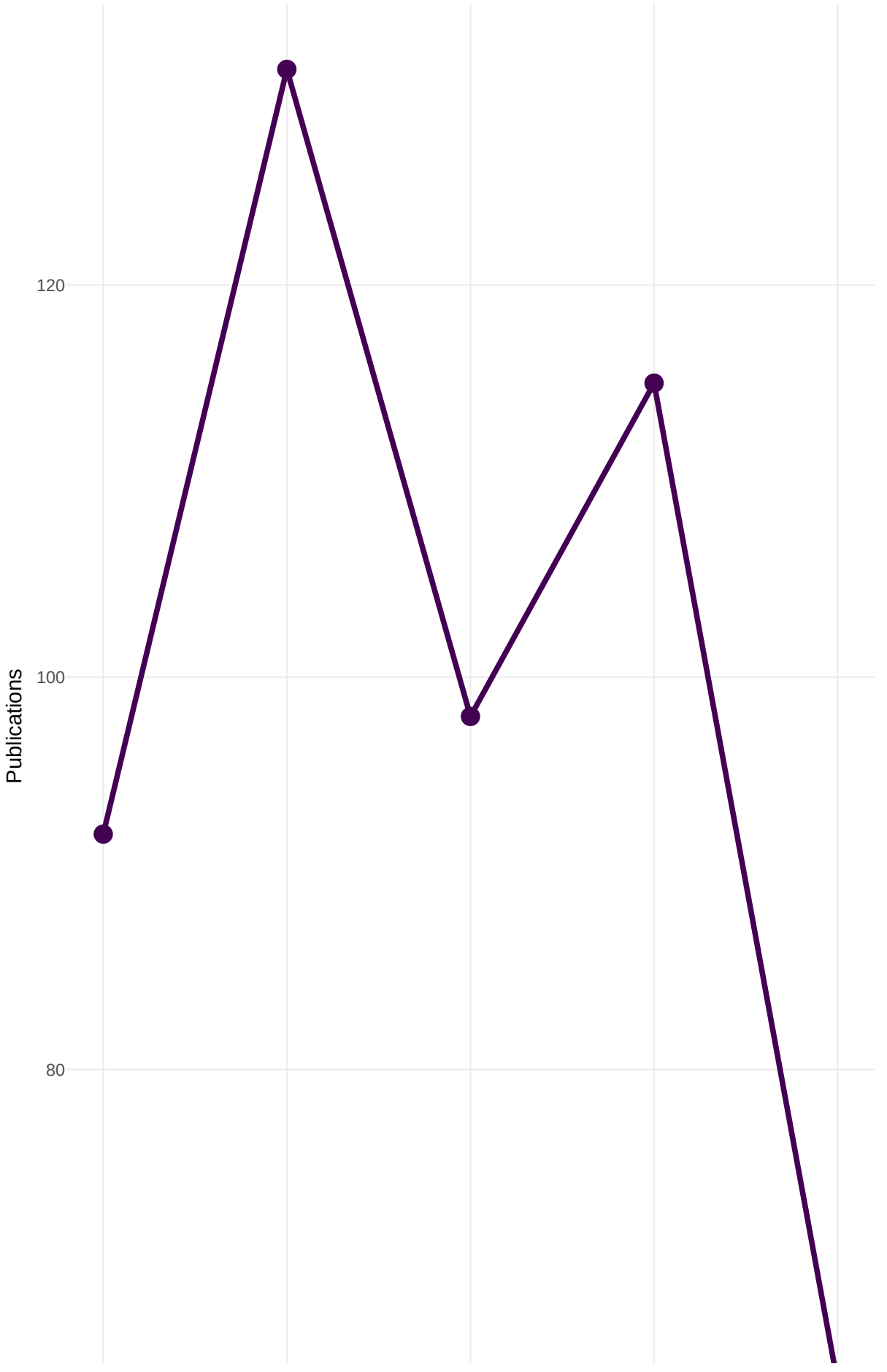

Publications
120
100
80



Figure 35.1: Interactive publication trend with hover details.

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)
```


75
50
Papers
25

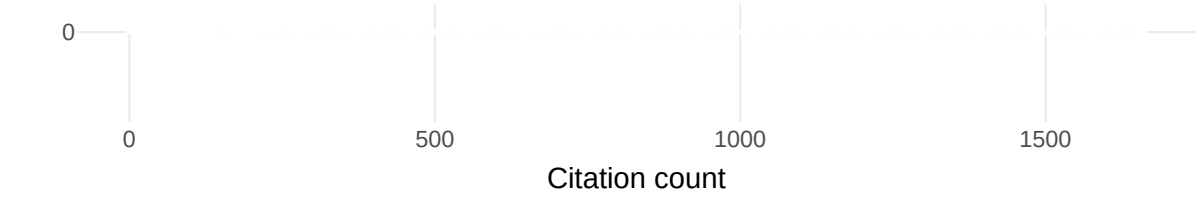


Figure 35.2: Interactive citation distribution with hover details.

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
  )
```

Show 10 entries Search:

title	year	cited_by_count	oa_status
All	All	All	All
Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations	2020	1658	hybrid
Sample size in bibliometric analysis	2020	281	hybrid
Open peer review: promoting transparency in open science	2020	225	hybrid
Rethinking SME default prediction: a systematic literature review and future perspectives	2021	212	hybrid
Visibility, impact, and applications of bibliometric software tools through citation analysis	2023	199	hybrid
The evolution of scientific disciplines in applied sciences: dynamics and empirical properties of experimental physics	2020	184	green
Preliminary analysis of COVID-19 academic information patterns: a call for open science in the times of closed borders	2020	168	hybrid
Gender gaps in international research collaboration: a bibliometric approach	2019	135	hybrid
Comprehensive metrological and content analysis of the public–private partnerships (PPPs) research field: a new bibliometric journey	2020	132	closed
Do articles in open access journals have more frequent altmetric activity than articles in subscription-based journals? An investigation of the research output of Finnish universities	2019	131	hybrid

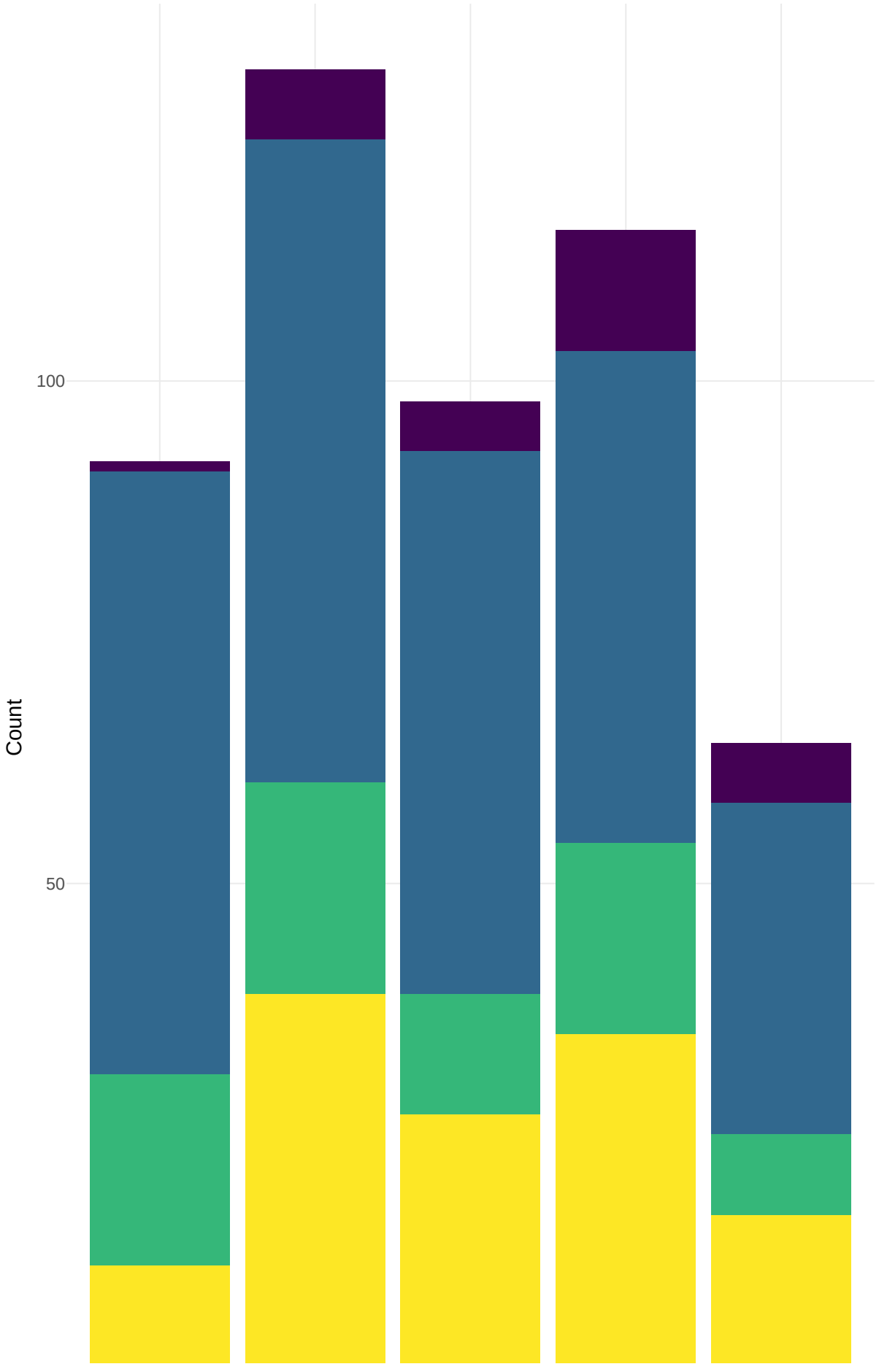
Showing 1 to 10 of 500 entries

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)
```


Count
100
50

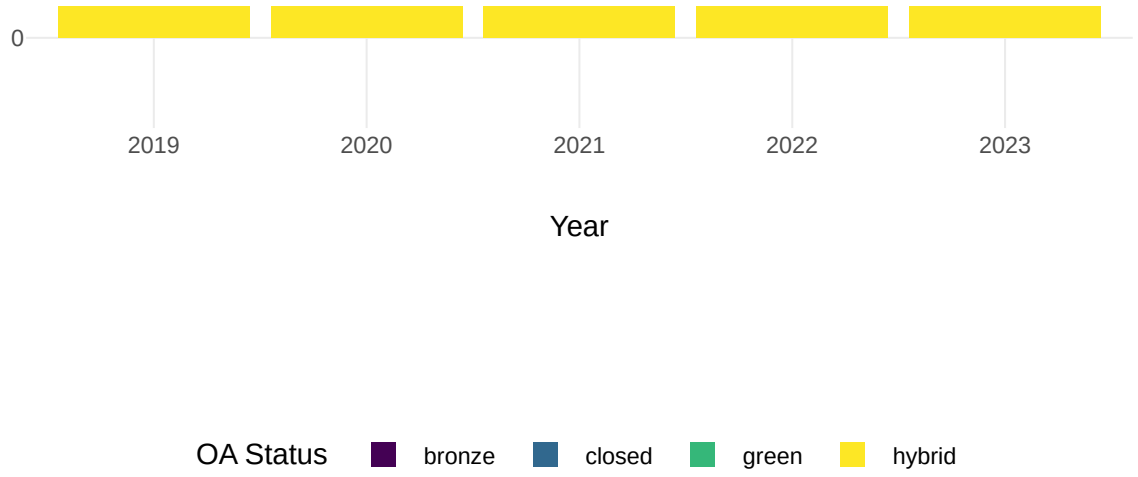


Figure 35.3: Interactive OA status breakdown by year.

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/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] 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       sass_0.5.0             bslib_0.12.0
#>  [22] htmlwidgets_1.6.4      tokenizers_0.3.0       plyr_1.8.9
#>  [25] httr2_1.3.0            cachem_1.1.0           dimensionsR_0.0.3
#>  [28] mime_0.13              lifecycle_1.0.5        pkgconfig_2.0.3
#>  [31] R6_2.6.1               fastmap_1.2.0          shiny_1.15.0
#>  [34] digest_0.6.39          patchwork_1.3.2        shinycssloaders_1.1.0
#>  [37] rprojroot_2.1.1        RSpectra_0.16-2        crosstalk_1.2.2
#>  [40] SnowballC_0.7.1        labeling_0.4.3         urltools_1.7.3.1
#>  [43] timechange_0.4.0       mgcv_1.9-4             polyclip_1.10-7
#>  [46] httr_1.4.9             compiler_4.6.1         here_1.0.2
#>  [49] bit64_4.8.6            withr_3.0.3            backports_1.5.1
#>  [52] S7_1.0.0               viridis_0.6.5          ggforce_0.5.0
#>  [55] MASS_7.3-65            bibliometrixData_0.3.0 tools_4.6.1
#>  [58] otel_0.3.0             stopwords_2.3          zip_3.0.2
#>  [61] httpuv_1.6.17          rentrez_1.2.4          nlme_3.1-169
#>  [64] promises_1.5.0         grid_4.6.1             stringdist_0.9.17
```

```
#>  [67] reshape2_1.4.5          generics_0.1.4          gtable_0.3.6
#>  [70] tzdb_0.5.0              rscopus_0.9.0           ca_0.72
#>  [73] data.table_1.18.6.1     hms_1.1.4               xml2_1.6.0
#>  [76] utf8_1.2.6              ggrepel_0.9.8           pillar_1.11.1
#>  [79] nsyllable_1.0.1         vroom_1.7.1             later_1.4.8
#>  [82] splines_4.6.1           tweenr_2.0.3            lattice_0.22-9
#>  [85] FNN_1.1.4.1             bit_4.6.0               tidyselect_1.2.1
#>  [88] tm_0.7-20               miniUI_0.1.2            downlit_0.4.5
#>  [91] knitr_1.52              gridExtra_2.3.1         NLP_0.3-3
#>  [94] bookdown_0.48           stats4_4.6.1            crul_1.6.0
#>  [97] xfun_0.62               graphlayouts_1.2.5      matrixStats_1.5.0
#> [100] humaniformat_0.6.0      stringi_1.8.9           qpdf_1.4.1
#> [103] yaml_2.3.12             evaluate_1.0.5          httpcode_0.3.0
#> [106] cli_3.6.6               xtable_1.8-8            jquerylib_0.1.4
#> [109] Rcpp_1.1.2              readxl_1.6.0            triebeard_0.4.1
#> [112] XML_3.99-0.25           parallel_4.6.1          assertthat_0.2.1
#> [115] pubmedR_1.1.0           slam_0.1-56             viridisLite_0.4.3
#> [118] scales_1.4.0            crayon_1.5.3            openxlsx_4.2.9
#> [121] rlang_1.3.0             fastmatch_1.1-8
```