

Package ‘inlpubs’

July 7, 2026

Title USGS INL Project Office Publications

Version 1.4.0

Description Provides bibliographic information and term-frequency text analysis tools for publications of the U.S. Geological Survey (USGS) Idaho National Laboratory (INL) Project Office. Includes datasets of publications, authors, and term frequencies, along with functions to search terms, build word clouds, and extract text and cover images from publication documents.

Depends R (>= 4.1)

Imports checkmate, knitr, kableExtra, parallel, stats, stringi, tm, utils

Suggests callr, chromote, cli, connectapi, covr, cyclocomp, graphics, htmltools, htmlwidgets, jsonlite, lintr, magick, markdown, pdftools (>= 2.0), pkgbuild, pkgdown, pkgload, png, rcmdcheck, reactable, renv, rmarkdown, rsconnect, RWeka, tesseract, textutils, tinytest, webshot2, wordcloud2

License CC0

URL <https://rconnect.usgs.gov/INLP0/inlpubs-main/>,
<https://code.usgs.gov/inl/inlpubs>

BugReports <https://code.usgs.gov/inl/inlpubs/-/issues>

Copyright This software is in the public domain because it contains materials that originally came from the United States Geological Survey (USGS), an agency of the United States Department of Interior. For more information, see the official USGS copyright policy at <https://www.usgs.gov/information-policies-and-instructions/copyrights-and-credits>

Encoding UTF-8

SystemRequirements Complete functionality necessitates Amazon Corretto (win), and default-jre, pandoc, libxml2-dev, libpoppler-cpp-dev, libmagick++-dev, optipng, libtesseract-dev, libleptonica-dev, tesseract-ocr-eng (deb)

LazyData true

LazyDataCompression xz
Config/roxygen2/version 8.0.0
NeedsCompilation no
Author Jason C. Fisher [aut, cre] (ORCID:
 <<https://orcid.org/0000-0001-9032-8912>>),
 Kerri C. Treinen [aut] (ORCID: <<https://orcid.org/0000-0003-0645-6810>>),
 Allison R. Trecka [aut] (ORCID: <<https://orcid.org/0000-0001-8498-4737>>)
Maintainer Jason C. Fisher <jfisher@usgs.gov>
Repository CRAN
Date/Publication 2026-07-06 22:40:02 UTC

Contents

add_content	2
authors	4
get_pdf_image	6
get_pdf_text	7
make_wordcloud	9
mine_text	10
plot_pubs	12
pubs	13
search_terms	14
terms	16
Index	17

add_content	<i>Add Content from PDF Documents</i>
-------------	---------------------------------------

Description

Extract the text or cover image from publication PDF documents and save the result to disk, for incorporation into the **inlpubs** package corpus. Publications are selected by identifier or by year, and each must already be present in the pubs dataset before this function is run.

Usage

```
add_content(  
  pub_id,  
  year,  
  type = c("text", "image"),  
  ...,  
  srcdir = "archive",  
  destdir = tempdir(),  
  ignore = NULL,
```

```

    pubs = inlpubs::pubs,
    overwrite = FALSE
)

```

Arguments

pub_id	'character' vector. Unique identifier(s) of the publication(s) to process. Publications may instead, or additionally, be selected with year.
year	'integer' vector. Year(s) of publication. All publications from the given year(s) are selected, in addition to any specified by pub_id.
type	'character' string. Type of content to extract from each PDF, either "text" (the default) or "image" (a cover image).
...	Additional arguments passed to the extraction function: get_pdf_text when type = "text", or get_pdf_image when type = "image".
srcdir	'character' string. Source directory containing the publication PDFs, organized into year subdirectories (see Details). Defaults to "archive" in the working directory.
destdir	'character' string. Destination directory for the extracted output files. Created if it does not already exist. Defaults to the session's temporary directory.
ignore	'character' vector. Publication identifier(s) to skip. Useful for documents that are known to fail automatic extraction.
pubs	'pub' table. Publications of the INLPO; see the pubs dataset for the expected data format.
overwrite	'logical' flag. Whether to overwrite an existing output file. Defaults to FALSE, in which case publications with an existing output are skipped.

Details

For each selected publication, the function locates its primary PDF (the first file listed in the publication's files entry) within the source directory, then delegates extraction to [get_pdf_text](#) (for type = "text") or [get_pdf_image](#) (for type = "image"). Extra arguments passed through ... are forwarded to whichever of those functions is used.

Source documents are expected to follow the archive layout `srcdir/<year>/<pub_id>/<file>`, for example `archive/2005/Knobel0thers2005/ofr20051223.pdf`. Output files are written to `destdir` and named `pub-<pub_id>`, with the appropriate extension (`.txt` or `.jpg`) supplied by the extraction function.

Publications whose files entry is empty, and those listed in `ignore`, are skipped with a message. By default, existing output files are left in place; set `overwrite = TRUE` to regenerate them.

Extraction failures (including renderer crashes or timeouts on problematic PDFs) are caught per publication: the offending document is reported and skipped, and processing continues with the next one. The identifiers of any failed documents are returned in the "failed" attribute of the result.

Processing the full archive can take a considerable amount of time. For routine use, run this function interactively and one publication year at a time so that progress can be monitored.

Value

Invisibly returns a 'character' vector of paths to the saved output files, carrying a "failed" attribute with the identifiers of any publications whose extraction failed. Returns NULL (invisibly) when no publications remain to process.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[get_pdf_text](#) and [get_pdf_image](#) for the underlying extraction functions, and the [pubs](#) dataset for publication metadata.

Examples

```
## Not run:
# Extract text for a single publication into the corpus folder
add_content("Knobel0thers2005", destdir = "data-raw/corpus")

# Extract text for every publication from a given year
add_content(year = 2005, destdir = "data-raw/corpus")

# Extract cover images into the vignettes folder
add_content(year = 2005, type = "image", destdir = "vignettes")

# Process several years, skipping a document that fails extraction
add_content(year = 2003:2005, ignore = "Mazurek2004", destdir = "data-raw/corpus")

## End(Not run)
```

authors

Contributing Authors to INLPO Publications

Description

Authors who have contributed to publications by the U.S. Geological Survey (USGS), Idaho Water Science Center, Idaho National Laboratory Project Office (INLPO).

Usage

authors

Format

An object of class 'author' that inherits from the 'data.frame' class and contains 267 rows and the following 10 columns:

author_id Unique identifier for the author, composed of the author's first initial and last name in lowercase (for example, "jfisher").

name Author name, surname first, followed by initials or given name.

person A named list of [person](#) objects holding contact details such as email address and **ORCID** identifier. Names match **author_id**.

pub_id A named list of character vectors giving the identifier(s) of the publication(s) the author contributed to; each value is a foreign key referencing the **pub_id** primary key of the [pubs](#) table.

total_pub Total number of publications.

single_authored Number of single-authored publications.

multi_authored Number of multi-authored publications.

first_authored Number of multi-authored publications in which the author appears as the first author.

first_year First year in which the author published.

last_year Most recent year in which the author published.

Source

Curated by INLPO staff.

See Also

[pubs](#) for the publications referenced by **pub_id**.

Examples

```
# Subset Jason Fisher's information and display its structure:
author <- authors["jfisher", ]
str(author, max.level = 3, width = 75, strict.width = "cut")

# Print the author's given name:
author$person |> format(include = "given")
```

get_pdf_image

Obtain Image from a PDF Document

Description

Render a single page of a PDF document to a JPEG image. Typically used to capture a publication's cover (the first page) as a thumbnail. Requires that the **pdftools**, **magick**, and **callr** packages are available.

Usage

```
get_pdf_image(
  input,
  output = tempfile(fileext = ".jpg"),
  page = 1,
  width = 300,
  depth = 8,
  quality = 70,
  timeout = 120
)
```

Arguments

input	'character' string. File path to the PDF document.
output	'character' string. File path for the JPEG image to write. If the path lacks a .jpg or .jpeg extension, .jpg is appended. Defaults to a temporary file.
page	'integer' number. Page number to render. Defaults to 1 (the cover page).
width	'integer' number. Output image width in pixels. Height is set automatically to preserve the page's aspect ratio. Defaults to 300.
depth	'integer' number. Image color depth, either 8 or 16. Defaults to 8.
quality	'integer' number between 0 and 100. JPEG compression quality. Defaults to 70.
timeout	'numeric' number. Maximum time, in seconds, to allow the PDF page to render before the operation is aborted.

Details

The selected page is rendered at a resolution computed from the requested output width (to avoid rasterizing a needlessly large image), resized with a high-quality Lanczos filter while preserving aspect ratio, flattened onto a white background, converted to an sRGB JPEG at the given color depth, and stripped of metadata before being written.

The Poppler-backed reads and the page rasterization are run in a child process, so a native crash on a problematic PDF (which cannot be caught with tryCatch) surfaces here as a normal R error instead of taking down the session or aborting a batch run.

Value

Returns the path to the image file, invisibly.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[add_content](#) to add cover images to the **inlpubs** package.

Examples

```
input <- system.file("extdata", "test.pdf", package = "inlpubs")
path <- get_pdf_image(input)

unlink(path)
```

get_pdf_text

Obtain Text from a PDF Document

Description

Extract text from a PDF document and write it to a plain-text file, one page per line. Pages with an extractable text layer are read directly; pages without one (e.g. scanned images) fall back to optical character recognition (OCR). The text is normalized to ASCII before writing.

Requires the **pdftools**, **tesseract**, and **callr** packages; image preprocessing additionally requires **magick**.

Usage

```
get_pdf_text(
  input,
  output = tempfile(fileext = ".txt"),
  dpi = 300,
  psm = 1,
  force_ocr = FALSE,
  preprocess = FALSE,
  min_chars = 20,
  cores = NULL,
  timeout = 120
)
```

Arguments

input	'character' string. File path to the PDF document.
output	'character' string. File path for the text file to write. If the path lacks a .txt or .text extension, .txt is appended. Defaults to a temporary file.
dpi	'integer' number between 100 and 1200. Resolution, in dots per inch, at which pages are rendered before OCR. Higher values improve OCR accuracy at the cost of speed; 300 (the default) suits most documents, while 400 to 600 may help with small fonts or poor scans. Ignored for pages read directly from the text layer.
psm	'integer' number between 0 and 13. Tesseract Page Segmentation Mode, describing the expected page layout. The default of 1 segments the page automatically and detects orientation and script, which suits multi-column documents.
force_ocr	'logical' flag. Whether to OCR every page regardless of its text layer. Use this when the embedded text is known to be unreliable (e.g. fonts that extract as gibberish, which min_chars cannot catch because the gibberish is long enough to pass). Defaults to FALSE.
preprocess	'logical' flag. Whether to grayscale, deskew, despeckle, and normalize each page image before OCR, which improves accuracy on scanned documents at a modest per-page cost. Requires the magick package. Applies only to pages that are OCR'd. Defaults to FALSE.
min_chars	'integer' number. Minimum number of non-whitespace characters a page's extracted text must have to be accepted without OCR. Pages below this threshold are sent to OCR. Defaults to 20.
cores	'integer' number. Number of parallel workers to use for OCR. When NULL (the default), the count is taken from the OCR_CORES environment variable if set (useful for honoring a container CPU quota), otherwise from one fewer than the number of detected cores.
timeout	'numeric' number. Maximum time, in seconds, to allow each child-process call to run before it is aborted. Applied independently to the text-extraction call and the page-rendering call. Defaults to 120.

Details

Direct extraction is attempted first, as it is faster and more accurate than OCR for born-digital PDFs; a page is OCR'd only when its text layer yields fewer than min_chars non-whitespace characters, or when force_ocr is TRUE. Poppler reads and page rendering run in isolated child processes, so a renderer crash surfaces as a normal R error rather than aborting the session.

Directly extracted text is reordered into human reading order from word bounding boxes: full-width headers and captions are set aside, then the body is split into columns only on strong evidence of a multi-column layout. OCR uses Tesseract's LSTM engine, run in parallel across cores workers when several pages need it. The result is normalized to strictly ASCII: selected scientific symbols become ASCII tokens, accents are transliterated, line-break hyphenation is rejoined, and quotes, control characters, and repeated punctuation and whitespace are cleaned up.

Value

Returns the path to the text file, invisibly. Each page of the PDF is written as a separate line.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[add_content](#) to add extracted text to the **inlpubs**-package corpus.

Examples

```
## Not run:
input <- system.file("extdata", "test.pdf", package = "inlpubs")
path <- get_pdf_text(input)

unlink(path)

## End(Not run)
```

make_wordcloud

Create a Word Cloud from Term Frequencies

Description

Create a word cloud from a term-frequency table and save it as a PNG file. The most frequent terms are rendered, laid out within the chosen shape. Requires the **htmltools**, **htmlwidgets**, **magick**, **webshot2**, and **wordcloud2** packages. System dependencies are **ImageMagick** (for displaying the PNG), **OptiPNG** (for PNG compression), and a **Chrome** or Chromium-based browser with support for the Chrome DevTools protocol. Use `chromote::find_chrome()` to locate the browser.

Usage

```
make_wordcloud(
  x,
  max_terms = 200,
  size = 1,
  shape = "circle",
  ellipticity = 0.65,
  ...,
  width = 910,
  output = NULL,
  display = FALSE
)
```

Arguments

<code>x</code>	'data.frame'. Term-frequency table containing at least the columns <code>term</code> and <code>freq</code> .
<code>max_terms</code>	'integer' number. Maximum number of terms to include, taken in order of decreasing frequency. Defaults to 200.

size	'numeric' number. Font-size scaling factor. Defaults to 1.
shape	'character' string. Shape of the cloud. One of "circle" (the default), "cardioid", "diamond", "triangle-forward", "triangle", "pentagon", or "star".
ellipticity	'numeric' number between 0 and 1. Degree of flattening applied to the shape. Defaults to 0.65.
...	Additional arguments passed to <code>wordcloud2::wordcloud2()</code> .
width	'integer' number. Image width in pixels. Defaults to 910. Image height is derived from width and ellipticity.
output	'character' string. Path to the output PNG file. Defaults to a temporary file.
display	'logical' flag. Whether to display the saved PNG in a graphics window. Requires the magick package. Defaults to FALSE.

Value

Invisibly returns the file path to the word-cloud PNG.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[mine_text](#) to build a term-frequency table.

Examples

```
## Not run:
d <- wordcloud2::demoFreq |> head(n = 10)
colnames(d) <- c("term", "freq")
file <- make_wordcloud(d, display = interactive())

unlink(file)

## End(Not run)
```

mine_text	<i>Build a Term-Frequency Table</i>
-----------	-------------------------------------

Description

Build a term-frequency table from document text. A term is a single word or, when n-gram tokenization is enabled, an ordered sequence of words.

Usage

```
mine_text(docs, ngmin = 1, ngmax = ngmin, sparse = NULL)
```

Arguments

docs	'list' or 'character' vector. Document text to analyze. Each element holds the extracted text of a single document and may itself be a character vector (its pieces are concatenated). NA values are dropped. If unnamed, elements are numbered sequentially.
ngmin, ngmax	'integer' numbers. Minimum and maximum number of words per term (n-gram). The default (ngmin = ngmax = 1) counts single words. Values greater than one require the RWeka package and that the JAVA_HOME environment variable point to a Java installation. Recommended for a single document only.
sparse	'numeric' number greater than 0 and less than 1, or NULL. Maximum allowed sparsity: terms appearing in a smaller proportion of documents are removed. For example, sparse = 0.99 removes terms more sparse than 0.99, while sparse = 0.01 retains only terms appearing in nearly every document. The default (NULL) disables sparse-term removal.

Details

Documents are cleaned before counting: URLs, escaped unicode sequences, HTML tags, and selected punctuation are removed; text is lowercased; numbers, English and SMART stopwords, and remaining punctuation are stripped; and whitespace is collapsed. HTML entities are decoded when the **textutils** package is available. When n-grams are requested, terms containing a repeated word are discarded.

Value

A 'data.frame' term-frequency table with columns term, pub_id, and freq, giving the number of times each term occurs in each document. Rows with a zero count are omitted, and the table is sorted by term (ascending) then frequency (descending).

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[search_terms](#) to search within the resulting table, and [make_wordcloud](#) to create a word cloud from it.

Examples

```
d <- c(
  "The quick brown fox jumps over the lazy lazy dog.",
  "Pack my brown box.",
  "Jazz fly brown dog."
) |>
  mine_text()

d <- list(
  "A" = "The quick brown fox jumps over the lazy lazy dog.",
```

```

"B" = c("Pack my brown box.", NA, "Jazz fly brown dog."),
"C" = NA_character_
) |>
  mine_text()

```

plot_pubs

Plot Annual Publication Counts

Description

Draw a stacked bar chart of INLPO publication counts by year, with USGS and non-USGS publications distinguished by color.

Usage

```

plot_pubs(
  pubs = inlpubs::pubs,
  author_id = NULL,
  full_range = TRUE,
  legend_loc = "topleft",
  colors = c(USGS = "#4682B4", `Non-USGS` = "#B47846"),
  ...
)

```

Arguments

pubs	'pub' data frame. Publications of the INLPO; see the pubs dataset for the expected data format. Defaults to <code>inlpubs::pubs</code> .
author_id	'character' vector, or NULL. One or more author identifiers used to subset pubs to publications that any of those authors appear on before plotting. The default (NULL) includes all publications (no author filtering).
full_range	'logical' flag. Whether the x-axis spans the year range of all publications in pubs (before any author filtering). If TRUE (the default), the timeline is fixed to the full dataset, making per-author charts directly comparable. If FALSE, the timeline is limited to the year range of the filtered publications. Has no effect when <code>author_id</code> is NULL.
legend_loc	'character' string. Position of the legend, passed as the keyword location to graphics::legend() . One of "topleft" (the default), "topright", "bottomleft", "bottomright", "top", "bottom", "left", "right", or "center".
colors	'character' vector of length 2. Fill colors for the two series, in the order <code>c("USGS", "Non-USGS")</code> . Defaults to steel blue and burnt orange.
...	Additional arguments passed to graphics::barplot() .

Value

Invisibly returns a numeric vector giving the x-axis coordinates of the bar midpoints, as returned by [graphics::barplot\(\)](#).

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[pubs](#) for the publications dataset.

Examples

```
plot_pubs()
```

pubs

Bibliographic Records for INLPO Publications

Description

Bibliographic information for reports, articles, maps, and theses related to scientific monitoring and research conducted by the U.S. Geological Survey (USGS), Idaho Water Science Center, Idaho National Laboratory Project Office (INLPO).

Usage

```
pubs
```

Format

An object of class 'pub' that inherits from the 'data.frame' class and includes the following columns:

pub_id Unique identifier for the publication (the table's primary key), in upper camel case (for example, "FisherOthers2012").

institution Name of the institution that published and/or sponsored the report.

type Type of publication, such as "Open-File Report", "Data Series", or "Manual".

text_ref Text reference (the in-text citation) excluding the year of publication, such as "Dietz" or "Treinen and others".

year Year of publication.

author_id A named list of character vectors giving the identifier(s) of the contributing author(s), in author order; each value is a foreign key referencing the **author_id** primary key of the [authors](#) table. List names match **pub_id**.

title Title of publication.

bibentry A named list of [bibentry](#) objects holding the full bibliographic record, from which formatted citations and BibTeX entries are derived. List names match **pub_id**.

abstract Abstract of publication, or NA when unavailable.

annotation Annotation summarizing the publication, or NA when none is provided.

annotation_src Identifier (**pub_id**) of the publication the annotation was taken from, or NA when not applicable (Knobel and others, 2005; Bartholomay, 2022).

files A named list of character vectors naming the file(s) associated with the publication, primary document first, or NULL when no files are recorded. List names match **pub_id**.

Source

Many of these publications are available through the [USGS Publications Warehouse](#).

References

Bartholomay, R.C., 2022, Historical development of the U.S. Geological Survey hydrological monitoring and investigative programs at the Idaho National Laboratory, Idaho, 2002-2020: U.S. Geological Survey Open-File Report 2022-1027 (DOE/ID-22256), 54 p., [doi:10.3133/ofr20221027](#).

Knobel, L.L., Bartholomay, R.C., and Rousseau, J.P., 2005, Historical development of the U.S. Geological Survey hydrologic monitoring and investigative programs at the Idaho National Engineering and Environmental Laboratory, Idaho, 1949 to 2001: U.S. Geological Survey Open-File Report 2005-1223 (DOE/ID-22195), 93 p., [doi:10.3133/ofr20051223](#).

See Also

[authors](#) for the contributing authors referenced by author_id, [terms](#) for term frequencies derived from publication text, and [search_terms](#) to query publications by term.

Examples

```
# Subset Fisher and others (2012) and display its structure:
id <- "FisherOthers2012"
pub <- pubs[id, ]
str(pub, max.level = 3, width = 75, strict.width = "cut")

# Print the suggested citation:
attr(unclass(pub$bibentry[[1]])[[1]], which = "textVersion")

# Print the authors' full names:
format(pub$bibentry[[1]]$author, include = c("given", "family"))

# Print the abstract:
pub$abstract
```

search_terms

Search a Term-Frequency Table

Description

Search a term-frequency table by pattern or frequency.

Usage

```
search_terms(
  x,
  data = inlpubs::terms,
  ignore_case = TRUE,
  ...,
```

```

    low_freq = 1,
    high_freq = Inf,
    simplify = TRUE
  )

```

Arguments

x	'character' string. Regular-expression pattern to search for among the table's terms.
data	'term' and 'data.frame' class. Term-frequency table to search, with columns term, pub_id, and freq. Defaults to the term frequencies from the INLPO publications; see the terms dataset for details.
ignore_case	'logical' flag. Whether to ignore character case when matching. Defaults to TRUE.
...	Additional arguments passed to grep() (for example, fixed or perl).
low_freq	'numeric' number. Lower frequency bound; terms occurring fewer than this many times are excluded. Defaults to 1.
high_freq	'numeric' number. Upper frequency bound; terms occurring more than this many times are excluded. Defaults to Inf (no upper limit).
simplify	'logical' flag. Whether to return only the unique publication identifiers (TRUE, the default) rather than the full table of matching rows.

Value

If `simplify = TRUE`, a 'character' vector of unique publication identifiers, ordered by the most frequent matching term first. If `simplify = FALSE`, a 'data.frame' of matching rows sorted by decreasing frequency.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[mine_text](#) to build a term-frequency table.

Examples

```

search_terms("mlms")

out <- search_terms("mlms", simplify = FALSE)
head(out)

```

terms

*Term Frequency from INLPO Publications***Description**

A term-frequency table giving the number of times each term appears in each publication, for publications by the U.S. Geological Survey (USGS), Idaho Water Science Center, Idaho National Laboratory Project Office (INLPO). The table has one row per term-publication pairing.

Usage

terms

Format

An object of class 'term' that inherits from the 'data.frame' class and includes the following columns:

term Term (a single word, or a multi-word n-gram), stored as a factor. Terms are lowercase ASCII.

pub_id Identifier of the publication in which the term appears, stored as a factor; a foreign key referencing the pub_id primary key of the [pubs](#) table.

freq Number of times the term occurs in that publication.

Source

The publication text was sourced from the original PDF documents using the [get_pdf_text](#) function, and term frequencies were extracted from the text using the [mine_text](#) function.

See Also

[mine_text](#) to build a term-frequency table, [search_terms](#) to search within it, and [make_wordcloud](#) to visualize it.

Examples

```
str(terms, max.level = 3, width = 75, strict.width = "cut")
```

```
# Most frequent terms across all publications:
freq_by_term <- tapply(terms$freq, terms$term, FUN = sum)
sort(freq_by_term, decreasing = TRUE) |> head(n = 10)
```

Index

* datasets

- authors, [4](#)
- pubs, [13](#)
- terms, [16](#)

add_content, [2](#), [7](#), [9](#)

authors, [4](#), [13](#), [14](#)

bibentry, [13](#)

chromote::find_chrome(), [9](#)

get_pdf_image, [3](#), [4](#), [6](#)

get_pdf_text, [3](#), [4](#), [7](#), [16](#)

graphics::barplot(), [12](#)

graphics::legend(), [12](#)

grep(), [15](#)

make_wordcloud, [9](#), [11](#), [16](#)

mine_text, [10](#), [10](#), [15](#), [16](#)

person, [5](#)

plot_pubs, [12](#)

pubs, [3–5](#), [12](#), [13](#), [13](#), [16](#)

search_terms, [11](#), [14](#), [14](#), [16](#)

terms, [14](#), [15](#), [16](#)

wordcloud2::wordcloud2(), [10](#)