# Package 'gutenbergr'

May 8, 2026

**Title** Download and Process Public Domain Works from Project Gutenberg

**Version** 0.5.1

**Description** Download and process public domain works in the Project Gutenberg collection <https://www.gutenberg.org/>. Includes metadata for all Project Gutenberg works, so that they can be searched and retrieved.

**License** GPL-2

**URL** https://docs.ropensci.org/gutenbergr/, https://ropensci.r-universe.dev/gutenbergr, https://github.com/ropensci/gutenbergr

**BugReports** https://github.com/ropensci/gutenbergr/issues

**Depends** R (>= 4.1)

**Imports** cli, dplyr, glue, purrr, readMDTable, readr, rlang, stringr, tibble, urltools

**Suggests** curl, devtools (>= 2.4.5), fs (>= 1.6.6), ggplot2, here (>= 1.0.2), knitr, lubridate (>= 1.9.4), rmarkdown, testthat (>= 3.0.0), textdata, tidyr, tidytext, usethis (>= 3.2.1), withr, xml2

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-US

**LazyData** TRUE

**LazyDataCompression** xz

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jordan Bradford [aut, cre] (ORCID: <https://orcid.org/0009-0000-8570-3474>), Jon Harmon [aut] (ORCID: <https://orcid.org/0000-0003-4781-4346>), Myfanwy Johnston [aut], David Robinson [aut, cph]

**Maintainer** Jordan Bradford <jrdnbradford@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-05-03 00:30:12 UTC

# Contents



---

`gutenberg_add_sections` *Add a section column to a Gutenberg tibble*

---

## Description

Identifies section markers (chapters, cantos, letters, etc.) in Project Gutenberg texts and adds a column indicating which section each line belongs to. Sections are forward-filled, so all text between markers belongs to the previous section.

## Usage

```
gutenberg_add_sections(
  data,
  pattern,
  ignore_case = TRUE,
  format_fn = NULL,
  group_by = "auto",
  section_col = "section"
)
```

## Arguments

| | |
|---|---|
| data | A tibble::tibble with a text column containing the text to analyze. Typically data should be piped from gutenberg_download and contain a gutenberg_id column, but this is not required. |
| pattern | A regex pattern to identify headers. Must match the specific formatting of your book. See Details and Examples for common patterns. |
| ignore_case | Logical; should pattern matching be case-insensitive? Default is TRUE. |
| format_fn | Optional function to format section text. Receives the matched text and returns formatted text. Common options include stringr::str_to_title and stringr::str_to_upper but a custom function can also be provided. |
| group_by | Character vector of column names to group by before filling sections, or NULL to disable grouping. Defaults to "auto", which automatically uses "gutenberg_id" if that column exists. Set to NULL to treat the entire dataset as one document, or specify custom column names for grouping (e.g., group_by = "book_title"). |
| section_col | Character string specifying the name of the section column to create. Defaults to "section". |

## Details

### Common Section Patterns for Project Gutenberg Books:

Different books use different formatting for their section markers. Here are patterns for common formats:

- Chapters with Roman numerals: `"^Chapter [IVXLCDM]+"`
- Chapters with Arabic numerals: `"^Chapter [0-9]+"`
- Plays with both Roman and Arabic numerals: `"^(ACT|SCENE) [IVXLCDM0-9]+"`
- Books (e.g., *Paradise Lost*): `"^BOOK [IVXLCDM]+"`
- Cantos (e.g., *Dante's Inferno*): `"^CANTO [IVXLCDM]+"`
- Staves (e.g., *A Christmas Carol*): `"^STAVE [IVXLCDM]+"`
- Multiple formats (e.g., *Frankenstein*): `"^(Letter|Chapter) [0-9]+"`

Use gutenberg_works() to search for books and examine a few lines with gutenberg_download() to determine the exact format before writing your pattern.

## Value

A tibble::tibble with an added column named according to section_col, containing the section marker for each row. Rows before the first section marker will have NA.

## Examples

```
# Dante's "Inferno" - Cantos with Roman numerals
inferno <- gutenberg_download(1001) |>
  gutenberg_add_sections(pattern = "^CANTO [IVXLCDM]+")

# Mary Shelley's "Frankenstein"
# Letters and Chapters with Arabic numerals, normalized to title case
frankenstein <- gutenberg_download(84) |>
```

```
  gutenberg_add_sections(
    pattern = "^(Letter|Chapter) [0-9]+",
    format_fn = stringr::str_to_title
  )

# Classic Brontë sisters' works
# Chapters with Roman numerals, with trailing periods removed from section text
# Consider using `options(gutenbergr_cache_type = "persistent")`
# to prevent redownloading in the future.
bronte_sisters <- gutenberg_download(
  c(
    767,  # "Agnes Grey" by Anne Brontë
    768,  # "Wuthering Heights" by Emily Brontë
    969,  # "The Tenant of Wildfell Hall" by Anne Brontë
    1260, # "Jane Eyre" by Charlotte Brontë
    9182, # "Villette" by Charlotte Brontë
  ),
  meta_fields = c("author", "title")
) |>
  gutenberg_add_sections(
    pattern = "^\\s*CHAPTER [IVXLCDM]+",
    format_fn = function(x) str_remove(x, "\\.$")
  )

# Leo Tolstoy's "War and Peace"
# Add two custom named columns for hierarchical sections
war_and_peace <- gutenberg_download(2600) |>
  gutenberg_add_sections(
    pattern = "^BOOK [A-Z]+",
    section_col = "book"
  ) |>
 gutenberg_add_sections(
   pattern = "^CHAPTER [IVXLCDM]+",
   section_col = "chapter"
 )
```

---

gutenberg_authors *Metadata about Project Gutenberg authors*

---

## Description

Data frame with metadata about each author of a Project Gutenberg work. Although the Project Gutenberg raw data also includes metadata on contributors, editors, illustrators, etc., this dataset contains only people who have been the single author of at least one work.

## Usage

```
gutenberg_authors
```

### Format

A `tibble::tibble()` with one row for each author, with the columns:

**gutenberg_author_id** Unique identifier for the author that can be used to join with the gutenberg_metadata dataset

**author** The `agent_name` field from the original metadata

**alias** Alias

**birthdate** Year of birth

**deathdate** Year of death

**wikipedia** Link to Wikipedia article on the author. If there are multiple, they are "|"-delimited

**aliases** Character vector of aliases. If there are multiple, they are "/"-delimited

### Details

To find the date on which this metadata was last updated, run `attr(gutenberg_authors, "date_updated")`.

### See Also

gutenberg_metadata, gutenberg_subjects

### Examples

```
# See date last updated
attr(gutenberg_authors, "date_updated")
```

---

## gutenberg_cache_clear_all

*Clear all files from the Gutenberg cache*

---

### Description

Deletes all cached `.rds` files in the directory currently returned by `gutenberg_cache_dir()`.

### Usage

```
gutenberg_cache_clear_all(verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `verbose` | Whether to show the status message confirming the path. |

### Value

The number of files deleted (invisibly).

## Examples

```
# Clear entire current cache
gutenberg_cache_clear_all()
```

---

`gutenberg_cache_dir` *Get the active cache directory path*

---

## Description

Calculates the path to the directory where Gutenberg files are stored, based on the current `gutenbergr_cache_type` and `gutenbergr_base_cache_dir` options.

## Usage

```
gutenberg_cache_dir()
```

## Value

A character string representing the path to the cache directory.

## Cache options

The following options control caching behavior:

- `gutenbergr_cache_type`: Character string indicating how downloaded works are cached. Must be either `"session"` (default) or `"persistent"`.
- `gutenbergr_base_cache_dir`: Base directory used for persistent caching when `gutenbergr_cache_type = "persistent"`.

  By default, this is an OS-specific cache directory determined by `tools::R_user_dir("gutenbergr", "cache")`. Advanced users may set this to a custom path.

## Examples

```
# Get current cache directory
gutenberg_cache_dir()
```

## gutenberg_cache_list *List files in the Gutenberg cache*

### Description

Provides a detailed list of files currently stored in the directory returned by `gutenberg_cache_dir()`.

### Usage

```
gutenberg_cache_list(verbose = TRUE)
```

### Arguments

| | |
|---|---|
| verbose | Whether to show the status message showing the cache directory path. |

### Value

A `tibble::tibble()` with the following columns:

**title** The title of the work.

**author** The author(s) of the work.

**file** The filename.

**size_mb** Size of the file in megabytes.

**modified** The last modification time.

**path** The file's absolute path.

### Examples

```
# List all works in the currently set cache
gutenberg_cache_list()

# Suppress the directory path message
gutenberg_cache_list(verbose = FALSE)
```

## gutenberg_cache_remove_ids *Delete specific files from the cache*

### Description

Delete specific files from the cache

### Usage

```
gutenberg_cache_remove_ids(ids, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| ids | A numeric or character vector of Gutenberg IDs to remove from the current cache. |
| verbose | Whether to show the status messages. |

### Value

The number of files successfully deleted (invisibly).

### Examples

```
# Remove specific books from cache
gutenberg_cache_remove_ids(c(1, 2))

# Remove silently
gutenberg_cache_remove_ids(1, verbose = FALSE)
```

---

## gutenberg_cache_set *Set the Gutenberg cache type*

---

### Description

Configures whether the cache should be temporary (per-session) or persistent across sessions.

### Usage

```
gutenberg_cache_set(
  type = getOption("gutenbergr_cache_type", "session"),
  verbose = TRUE
)
```

### Arguments

| | |
|---|---|
| type | Either "session" (default) or "persistent".<br>• "session": Files are stored in a tempdir(). This is the default behavior.<br>• "persistent": Files are stored in an OS-specific user cache directory under works_rds/. These files persist across sessions, preventing redundant downloads of the same files in the future. |
| verbose | Whether to show the status message confirming the path. |

### Value

The active cache path (invisibly).

### Cache options

The following options control caching behavior:

- `gutenbergr_cache_type`: Character string indicating how downloaded works are cached. Must be either `"session"` (default) or `"persistent"`.
- `gutenbergr_base_cache_dir`: Base directory used for persistent caching when `gutenbergr_cache_type = "persistent"`.

  By default, this is an OS-specific cache directory determined by `tools::R_user_dir("gutenbergr", "cache")`. Advanced users may set this to a custom path.

## Examples

```
# Set to persistent (survives R sessions)
gutenberg_cache_set("persistent")

# Set back to session cache (temporary)
gutenberg_cache_set("session")

# Check current cache location
gutenberg_cache_dir()
```

---

`gutenberg_download` *Download one or more works using a Project Gutenberg ID*

---

## Description

Download one or more works by their Project Gutenberg IDs into a data frame with one row per line per work. This can be used to download a single work of interest or multiple at a time. You can look up the Gutenberg IDs of a work using `gutenberg_works()` or the gutenberg_metadata dataset.

## Usage

```
gutenberg_download(
  gutenberg_id,
  mirror = gutenberg_get_mirror(verbose = verbose),
  strip = TRUE,
  meta_fields = character(),
  verbose = TRUE,
  use_cache = TRUE
)
```

## Arguments

| | |
|---|---|
| gutenberg_id | A vector of Project Gutenberg IDs, or a data frame containing a gutenberg_id column, such as from the results of gutenberg_works(). |
| mirror | A mirror URL to retrieve the books from. By default uses the mirror from gutenberg_get_mirror(). |
| strip | Whether to strip suspected headers and footers using gutenberg_strip(). |
| meta_fields | Additional fields describing each book, such as title and author, to add from gutenberg_metadata. |
| verbose | Whether to show messages about the Project Gutenberg mirror that was chosen. |
| use_cache | Whether to use caching. Defaults to TRUE.<br>• See gutenberg_cache_set() for details on configuring caching.<br>• See gutenberg_cache_dir() to check your current cache location.<br>• The files in the cache are .rds files that have already been processed into a tbl_df. |

## Value

A two column tbl_df (see tibble::tibble()) with one row for each line of the text or texts, with columns:

**gutenberg_id** Integer column with the Project Gutenberg ID of each text

**text** A character vector of lines of text

## Examples

```
# Download "The Count of Monte Cristo"
gutenberg_download(1184)

# Download two books: "Wuthering Heights" and "Jane Eyre"
books <- gutenberg_download(c(768, 1260), meta_fields = "title")
books
dplyr::count(books, title)

# Download all books from Jane Austen
austen <- gutenberg_works(author == "Austen, Jane") |>
  gutenberg_download(meta_fields = "title")
austen
dplyr::count(austen, title)
```

## gutenberg_get_all_mirrors *Get all mirror data from Project Gutenberg*

### Description

Get all mirror data from https://www.gutenberg.org/MIRRORS.ALL. This only includes mirrors reported to Project Gutenberg and verified to be relatively stable. For more information on mirroring and getting your own mirror listed, see https://www.gutenberg.org/help/mirroring.html.

### Usage

```
gutenberg_get_all_mirrors()
```

### Value

A tibble::tibble() of Project Gutenberg mirrors and related data, or NULL (invisibly) if the mirror list cannot be retrieved or parsed.

If a tibble::tibble() is returned, it contains:

**continent** Continent where the mirror is located

**nation** Nation where the mirror is located

**location** Location of the mirror

**provider** Provider of the mirror

**url** URL of the mirror

**note** Special notes

### Examples

```
gutenberg_get_all_mirrors()
```

## gutenberg_get_mirror *Get the recommended mirror for Gutenberg files*

### Description

Get the recommended mirror for Gutenberg files and set the global gutenberg_mirror option.

### Usage

```
gutenberg_get_mirror(verbose = TRUE)
```

### Arguments

| | |
|---|---|
| verbose | Whether to show messages about the Project Gutenberg mirror that was chosen. |

### Value

A character vector with the URL for the chosen mirror.

### Examples

```
gutenberg_get_mirror()
```

---

## gutenberg_languages *Metadata about Project Gutenberg languages*

---

### Description

Data frame with metadata about the languages of each Project Gutenberg work.

### Usage

```
gutenberg_languages
```

### Format

A `tibble::tibble()` with one row for each work-language pair, with the columns:

**gutenberg_id** Unique identifier for the work that can be used to join with the gutenberg_metadata dataset

**language** Language ISO 639 code. Two letter code if one exists, otherwise three letter.

**total_languages** Number of languages for this work.

### Details

To find the date on which this metadata was last updated, run `attr(gutenberg_languages, "date_updated")`.

### See Also

gutenberg_metadata, gutenberg_subjects

### Examples

```
# See date last updated
attr(gutenberg_languages, "date_updated")
```

gutenberg_metadata *Gutenberg metadata about each work*

## Description

Selected fields of metadata about each of the Project Gutenberg works.

## Usage

```
gutenberg_metadata
```

## Format

A `tibble::tibble()` with one row for each work in Project Gutenberg and the following columns:

**gutenberg_id** Numeric ID, used to retrieve works from Project Gutenberg

**title** Title

**author** Author, if a single one given. Given as last name first (e.g. "Doyle, Arthur Conan")

**gutenberg_author_id** Project Gutenberg author ID

**language** Language ISO 639 code, separated by / if multiple. Two letter code if one exists, otherwise three letter. See https://en.wikipedia.org/wiki/List_of_ISO_639-2_codes

**gutenberg_bookshelf** Which collection or collections this is found in, separated by / if multiple

**rights** Generally one of three options: "Public domain in the USA." (the most common by far), "Copyrighted. Read the copyright notice inside this book for details.", or "None"

**has_text** Whether there is a file containing digits followed by `.txt` in Project Gutenberg for this record (as opposed to, for example, audiobooks). If not, cannot be retrieved with `gutenberg_download()`

## Details

To find the date on which this metadata was last updated, run `attr(gutenberg_metadata, "date_updated")`.

## See Also

`gutenberg_works()`, gutenberg_authors, gutenberg_subjects

## Examples

```
library(dplyr)
library(stringr)

gutenberg_metadata

gutenberg_metadata |>
  count(author, sort = TRUE)

# Look for Shakespeare, excluding collections (containing "Works") and
```

```
# translations
shakespeare_metadata <- gutenberg_metadata |>
  filter(
    author == "Shakespeare, William",
    language == "en",
    !str_detect(title, "Works"),
    has_text,
    !str_detect(rights, "Copyright")
  ) |>
  distinct(title)

# Note that the gutenberg_works() function filters for English
# non-copyrighted works and does de-duplication by default:

shakespeare_metadata2 <- gutenberg_works(
  author == "Shakespeare, William",
  !str_detect(title, "Works")
)

# See date last updated
attr(gutenberg_metadata, "date_updated")
```

---

`gutenberg_strip` *Strip header and footer content from a Project Gutenberg book*

---

## Description

Strip header and footer content from a Project Gutenberg book. This is based on formatting heuristics (regular expression guesses), so it may not be perfect.

## Usage

```
gutenberg_strip(text)
```

## Arguments

| | |
|---|---|
| `text` | A character vector where each element is a line of a book. |

## Details

This function identifies the Project Gutenberg "start" and "end" markers. It also attempts to strip out initial metadata paragraphs (such as "Produced by...", "Transcribed from...", etc.).

Note that this will **not** strip:

- Tables of contents
- Prologues or introductions
- Other author-written text that appears at the start of a book

## Value

A character vector with Project Gutenberg headers and footers removed.

## Examples

```
library(dplyr)

# Download a book without stripping to see the headers
book <- gutenberg_works(title == "Pride and Prejudice") |>
  gutenberg_download(strip = FALSE)

# Look at the raw header and footer
head(book$text, 20)
tail(book$text, 20)

# Manually strip the text
text_stripped <- gutenberg_strip(book$text)

# Check the cleaned results
head(text_stripped, 10)
tail(text_stripped, 10)
```

---

# gutenberg_subjects *Gutenberg metadata about the subject of each work*

---

## Description

Gutenberg metadata about the subject of each work, particularly Library of Congress Classifications (lcc) and Library of Congress Subject Headings (lcsh).

## Usage

```
gutenberg_subjects
```

## Format

A `tibble::tibble()` with one row for each pairing of work and subject, with columns:

**gutenberg_id** ID describing a work that can be joined with gutenberg_metadata

**subject_type** Either "lcc" (Library of Congress Classification) or "lcsh" (Library of Congress Subject Headings)

**subject** Subject

## Details

Find more information about Library of Congress Categories here: https://www.loc.gov/catdir/cpso/lcco/, and about Library of Congress Subject Headings here: https://id.loc.gov/authorities/subjects.html.

To find the date on which this metadata was last updated, run `attr(gutenberg_subjects, "date_updated")`.

## See Also

gutenberg_metadata, gutenberg_authors

## Examples

```
library(dplyr)
library(stringr)

gutenberg_subjects |>
  filter(subject_type == "lcsh") |>
  count(subject, sort = TRUE)

sherlock_holmes_subjects <- gutenberg_subjects |>
  filter(str_detect(subject, "Holmes, Sherlock"))

sherlock_holmes_subjects

sherlock_holmes_metadata <- gutenberg_works() |>
  filter(author == "Doyle, Arthur Conan") |>
  semi_join(sherlock_holmes_subjects, by = "gutenberg_id")

sherlock_holmes_metadata


holmes_books <- gutenberg_download(sherlock_holmes_metadata$gutenberg_id)

holmes_books


# See date last updated
attr(gutenberg_subjects, "date_updated")
```

---

# gutenberg_works *Get a filtered table of Gutenberg work metadata*

---

## Description

Get a table of Gutenberg work metadata that has been filtered by some common (settable) defaults, along with the option to add additional filters. This function is for convenience when working with common conditions when pulling a set of books to analyze. For more detailed filtering of the entire Project Gutenberg metadata, use the gutenberg_metadata and related datasets.

## Usage

```
gutenberg_works(
  ...,
  languages = "en",
```

```
  only_text = TRUE,
  rights = c("Public domain in the USA.", "None"),
  distinct = TRUE,
  all_languages = FALSE,
  only_languages = TRUE
)
```

## Arguments

| | |
|---|---|
| `...` | Additional filters, given as expressions using the variables in the gutenberg_metadata dataset (e.g. `author == "Austen, Jane"`). |
| `languages` | Vector of languages to include. |
| `only_text` | Whether the works must have Gutenberg text attached. Works without text (e.g. audiobooks) cannot be downloaded with `gutenberg_download()`. |
| `rights` | Values to allow in the `rights` field. By default allows public domain in the US or "None", while excluding works under copyright. `NULL` allows any value of Rights. |
| `distinct` | Whether to return only one distinct combination of each title and `gutenberg_author_id`. If multiple occur (that fulfill the other conditions), it uses the one with the lowest ID. |
| `all_languages` | Whether, if multiple languages are given, all of them need to be present in a work. For example, if `c("en", "fr")` are given, whether only `en/fr` as opposed to English or French works should be returned. |
| `only_languages` | Whether to exclude works that have other languages besides the ones provided. For example, whether to include `en/fr` when English works are requested. |

## Details

By default, returns:

- English-language works.
- Works that are in text format in Gutenberg (as opposed to audio).
- Works whose text is not under copyright.
- At most one distinct field for each title/author pair.

## Value

A `tibble::tibble()` with one row for each work, in the same format as gutenberg_metadata.

## Examples

```
library(dplyr)

# Default: English, text-based, public domain works
gutenberg_works()

# Filter conditions using ...
```

```
gutenberg_works(author == "Shakespeare, William")

# Language specifications
gutenberg_works(languages = "es") |>
  count(language, sort = TRUE)

# Filter for works that are specifically English AND French
gutenberg_works(languages = c("en", "fr"), all_languages = TRUE)
```

---

`sample_books` *Sample Book Downloads*

---

## Description

A `tibble::tibble()` of book text for two sample books, generated using `gutenberg_download()`.

## Usage

```
sample_books
```

## Format

A `tibble::tibble()` with one row for each line of text from each book, with columns:

**gutenberg_id** Unique identifier for the work that can be used to join with the gutenberg_metadata dataset.

**text** A character vector of lines of text.

**title** The title of this work.

**author** The author of this work.

## Details

This code was used to download the books: `gutenberg_download(c(109, 105), meta_fields = c("title", "author"))`

# Index