

Package: gendertext (via r-universe)

June 19, 2026

Title Detect Gendered Words in Text and Suggest Neutral Alternatives

Version 0.1.0

Description Identifies gendered words and phrases in text using a built in dictionary of more than two hundred gendered terms paired with gender neutral alternatives. Reports the share of gendered language in a text, lists every gendered term found together with its suggested neutral replacement, and can rewrite a text in gender neutral form. Plain text files are read with base R, while other document formats such as PDF and Word are supported through the optional 'readtext' package. The dictionary is informed by published guidance on gender inclusive language, including the United Nations guidelines <<https://www.un.org/en/gender-inclusive-language/>> and the European Parliament guidance on gender neutral language.

License MIT + file LICENSE

URL <https://github.com/mashrur-ayon/gendertext>

BugReports <https://github.com/mashrur-ayon/gendertext/issues>

Depends R (>= 3.5)

Suggests knitr, readtext, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Repository <https://mashrur-ayon.r-universe.dev>

Date/Publication 2026-06-11 14:34:00 UTC

RemoteUrl <https://github.com/mashrur-ayon/gendertext>

RemoteRef HEAD

RemoteSha d9dfac3303a199d4ff2b796ebdf99b8ced33fc1d

RemoteSubdir gendertext

Contents

gender_dictionary	2
gender_replace	3
gender_score	4
gender_suggestions	6
read_text	7

Index	8
--------------	----------

gender_dictionary	<i>Dictionary of Gendered Terms and Gender Neutral Alternatives</i>
-------------------	---

Description

A curated dictionary of gendered English words and phrases, each paired with a suggested gender neutral alternative. The dictionary covers gendered occupational titles (for example "chairman" and "stewardess"), gendered pronouns, forms of address, family and relationship terms, and common idioms and compounds built on gendered words. It powers `gender_score()`, `gender_suggestions()`, and `gender_replace()`.

Usage

```
gender_dictionary
```

Format

A data frame with 208 rows and 2 variables:

gendered Character. A gendered word or phrase, in lower case.

neutral Character. The suggested gender neutral alternative.

Details

All entries are stored in lower case. Matching in the package functions is case insensitive and tolerant of possessive forms, so "Chairman's" in a text is matched by the entry "chairman".

The selection of terms and replacements is informed by published guidance on gender inclusive language, including the United Nations guidelines for gender inclusive language in English and the European Parliament guidance on gender neutral language.

Source

Curated by the package authors, informed by the United Nations guidelines for gender inclusive language (<https://www.un.org/en/gender-inclusive-language/>) and the European Parliament guidance on gender neutral language (https://www.europarl.europa.eu/cmsdata/151780/GNL_Guidelines_EN.pdf).

Examples

```
data(gender_dictionary)
head(gender_dictionary)
nrow(gender_dictionary)
```

gender_replace	<i>Rewrite Text with Gender Neutral Alternatives</i>
----------------	--

Description

Replaces gendered terms and phrases in a text or in a file with the gender neutral alternatives from the built in dictionary [gender_dictionary](#) or from a user supplied dictionary. Longer phrases are replaced before shorter ones and matching is case insensitive. The capitalisation of each replacement follows the matched text: an all caps match yields an all caps replacement and a match starting with a capital letter yields a capitalised replacement.

Usage

```
gender_replace(text = NULL, path = NULL, dictionary = NULL)
```

Arguments

text	A character string containing the text to rewrite. Optional if path is provided.
path	A character string giving a file path (txt, pdf, docx, and other formats supported by read_text()). Optional if text is provided.
dictionary	Optional data frame with character columns gendered and neutral to use instead of the built in gender_dictionary .

Details

The function performs plain dictionary substitution. It does not adjust the surrounding grammar, so a replacement such as "they" for "he" may require manual revision of verb forms. It is intended as a drafting aid, not a fully automatic rewriter.

Value

A length one character string containing the rewritten text.

See Also

[gender_suggestions\(\)](#) to preview the replacements, [gender_score\(\)](#) for an overall share, and [gender_dictionary](#) for the built in dictionary.

Examples

```
gender_replace(text = "The chairman called the policeman.")

# Capitalisation is preserved
gender_replace(text = "Chairman Smith spoke. THE FIREMAN AGREED.")

# Use a custom dictionary
my_dict <- data.frame(gendered = "dude", neutral = "person")
gender_replace(text = "Hey dude!", dictionary = my_dict)
```

gender_score	<i>Gendered Language Score</i>
--------------	--------------------------------

Description

Computes the share of gendered language in a text or in a file, based on the built in dictionary [gender_dictionary](#) or on a user supplied dictionary. Multi word phrases are matched before single words and each piece of text is counted at most once, so a phrase such as "ladies and gentlemen" is never counted again as "ladies" plus "gentlemen".

Usage

```
gender_score(
  text = NULL,
  path = NULL,
  unit = c("tokens", "matches"),
  dictionary = NULL
)
```

Arguments

text	A character string containing the text to analyse. Optional if path is provided.
path	A character string giving a file path (txt, pdf, docx, and other formats supported by read_text()). Optional if text is provided.
unit	A character string giving the counting unit: <ul style="list-style-type: none"> • "tokens" (default): reports gendered tokens as a share of all tokens in the cleaned text. • "matches": reports the total number of dictionary matches only (useful for quick detection).
dictionary	Optional data frame with character columns gendered and neutral to use instead of the built in gender_dictionary .

Details

The reported neutral share is a proxy. It is the proportion of tokens that are not matched by any dictionary entry, not a comprehensive linguistic measure of neutrality. When `unit = "tokens"`, a matched multi word phrase contributes one gendered unit for every word it spans, so the gendered and neutral percentages always sum to 100.

Value

A data frame with one row and the following columns:

total_units Total number of units counted (tokens or matches, depending on unit).

gendered_units Number of gendered units detected.

neutral_units For `unit = "tokens"`, the difference `total_units - gendered_units`. Otherwise NA.

gendered_percent Percentage of gendered units.

neutral_percent Percentage of unmatched (proxy neutral) units, or NA for `unit = "matches"`.

See Also

[gender_suggestions\(\)](#) to list the terms behind the score, [gender_replace\(\)](#) to rewrite the text, and [gender_dictionary](#) for the built in dictionary.

Examples

```
# Direct text input
gender_score(text = "The chairman said he will call the policeman.")

# Count matches only
gender_score(text = "The chairman spoke.", unit = "matches")

# Analyse a file shipped with the package
txt <- system.file("extdata", "test.txt", package = "gendertext")
gender_score(path = txt)

# Use a custom dictionary
my_dict <- data.frame(
  gendered = c("dude"),
  neutral = c("person")
)
gender_score(text = "Hey dude!", dictionary = my_dict)
```

gender_suggestions *Gender Neutral Suggestions for Detected Gendered Terms*

Description

Identifies the gendered terms and phrases that occur in a text or in a file, using the built in dictionary [gender_dictionary](#) or a user supplied dictionary, and returns the suggested gender neutral alternative for each detected term, optionally with occurrence counts.

Usage

```
gender_suggestions(
  text = NULL,
  path = NULL,
  include_counts = TRUE,
  dictionary = NULL
)
```

Arguments

text	A character string containing the text to analyse. Optional if path is provided.
path	A character string giving a file path (txt, pdf, docx, and other formats supported by read_text()). Optional if text is provided.
include_counts	Logical; if TRUE (default), the result includes the number of occurrences found for each detected term.
dictionary	Optional data frame with character columns gendered and neutral to use instead of the built in gender_dictionary .

Value

A data frame with one row per detected term, sorted by decreasing count and then alphabetically:

gendered Detected gendered term or phrase from the dictionary.

suggested_neutral Suggested gender neutral replacement.

count Number of occurrences in the text (only when `include_counts = TRUE`).

See Also

[gender_score\(\)](#) for an overall share, [gender_replace\(\)](#) to apply the suggestions, and [gender_dictionary](#) for the built in dictionary.

Examples

```
gender_suggestions(text = "Our chairman said he will email the mailman.")

# Without counts
gender_suggestions(
  text = "The fireman and the policeman arrived.",
  include_counts = FALSE
)

# Analyse a file shipped with the package
txt <- system.file("extdata", "test.txt", package = "gendertext")
head(gender_suggestions(path = txt))
```

read_text	<i>Read Text from a File</i>
-----------	------------------------------

Description

Reads text content from a document and returns it as a single character string for downstream analysis in [gender_score\(\)](#), [gender_suggestions\(\)](#), and [gender_replace\(\)](#). Plain text files (extensions txt, text, md, and rmd) are read with base R. Other formats such as pdf, docx, rtf, odt, csv, and json are read with the suggested 'readtext' package when it is installed.

Usage

```
read_text(path)
```

Arguments

path A character string giving the path to a file. The file must exist.

Value

A length one character string containing the extracted text.

See Also

[gender_score\(\)](#), [gender_suggestions\(\)](#), [gender_replace\(\)](#)

Examples

```
txt <- system.file("extdata", "test.txt", package = "gendertext")
substr(read_text(txt), 1, 60)

if (requireNamespace("readtext", quietly = TRUE)) {
  pdf <- system.file("extdata", "test.pdf", package = "gendertext")
  substr(read_text(pdf), 1, 60)
}
```

Index

* datasets

- gender_dictionary, 2

- gender_dictionary, 2, 3–6
- gender_replace, 3
- gender_replace(), 2, 5–7
- gender_score, 4
- gender_score(), 2, 3, 6, 7
- gender_suggestions, 6
- gender_suggestions(), 2, 3, 5, 7

- read_text, 7
- read_text(), 3, 4, 6