

Package ‘gerda’

August 12, 2025

Title German Election Database (GERDA)

Version 0.2.0

Author Hanno Hilbig [aut, cre] (<<https://orcid.org/0000-0001-5849-9172>>)

Description Provides tools to download comprehensive datasets of local, state, and federal election results in Germany from 1990 to 2025. The package facilitates access to data on turnout, vote shares for major parties, and demographic information across different levels of government (municipal, state, and federal). It offers access to geographically harmonized datasets that account for changes in municipal boundaries over time and incorporate mail-in voting districts. Users can easily retrieve, clean, and standardize German electoral data, making it ready for analysis. Data is sourced from <https://german-elections.com>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports stringdist, dplyr, readr, knitr, tibble

Suggests rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/hhilbig/gerda>, <https://german-elections.com>

BugReports <https://github.com/hhilbig/gerda/issues>

VignetteBuilder knitr

Maintainer Hanno Hilbig <hhilbig@ucdavis.edu>

Contents

gerda_data_list	2
load_gerda_web	2
party_crosswalk	3

Index	4
-------	---

gerda_data_list	<i>List of GERDA Data</i>
-----------------	---------------------------

Description

This function lists the available GERDA data sets. The purpose of this function is to quickly provide a list of available data sets and their descriptions.

Usage

```
gerda_data_list(print_table = TRUE)
```

Arguments

print_table	A logical value indicating whether to print the table in the console (TRUE) or return the data as a tibble (FALSE). Default is TRUE.
-------------	--

Value

A tibble containing the available GERDA data with descriptions. When `print_table = TRUE`, the function prints a formatted table to the console and invisibly returns the data tibble. When `print_table = FALSE`, the function directly returns the data tibble.

Examples

```
gerda_data_list()
```

load_gerda_web	<i>Load GERDA Data</i>
----------------	------------------------

Description

This function loads GERDA data from a web source.

Usage

```
load_gerda_web(file_name, verbose = FALSE, file_format = "rds")
```

Arguments

file_name	A character string specifying the name of the file to load. For a list of available data, see gerda_data_list .
verbose	A logical value indicating whether to print additional messages to the console. Default is FALSE.
file_format	A character string specifying the format of the file. Must be either "csv" or "rds". Default is "rds".

Value

A tibble containing the loaded data, or NULL if the data could not be loaded.

Examples

```
# Load harmonized municipal elections data
data_municipal_harm <- load_gerda_web("municipal_harm", verbose = TRUE, file_format = "rds")

# Load federal election data harmonized to 2025 boundaries (includes 2025 election)
data_federal_2025 <- load_gerda_web("federal_muni_harm_25", verbose = TRUE, file_format = "rds")
```

party_crosswalk	<i>GERDA</i>
-----------------	--------------

Description

This function creates a crosswalk between parties and their corresponding names using the ParlGov `view_party` table. In cases where the party name is not found in the `view_party` table, the function returns NA. Note that this function should be run on GERDA party names, and will likely not work on other party naming schemes.

Usage

```
party_crosswalk(party_gerda, destination)
```

Arguments

<code>party_gerda</code>	A character vector containing the GERDA party names to be converted.
<code>destination</code>	The name of the column in the <code>view_party</code> table to map to.

Value

A vector with the mapped party names.

Examples

```
party_crosswalk(c("cdu", "spd", "linke_pds", NA), "left_right")
```

Index

gerda_data_list, [2](#), [2](#)

load_gerda_web, [2](#)

party_crosswalk, [3](#)