Package: geodk (via r-universe)

October 20, 2026

Type Package

Title Access Danish Geospatial Data

Version 0.0.0.9001

Date 2025-03-10

Maintainer Aleksander Bang-Larsen <contact@aleksanderbl.dk>

Description This package provides access to all geospatial data provided by the danish agency called Klimadatastyrelsen. Under the hood it wraps the `{dawaR}` and `{dkdata}` packages which provide access to the agency APIs.

License GPL (>= 3)

URL https://ropengov.github.io/geodk/, https://github.com/rOpenGov/geodk

BugReports https://github.com/rOpenGov/geodk/issues

Depends R (>= 4.1.0)

Imports dawaR (>= 0.2.3), dplyr, ggplot2 (>= 3.4.0), rlang, sf, cli

Suggests testthat (>= 3.0.0), vdiffr, tibble, readr, knitr, rmarkdown, dkstat

Remotes rOpenGov/dkstat

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

X-schema.org-isPartOf http://ropengov.org/

X-schema.org-keywords ropengov

VignetteBuilder knitr

Repository https://ropengov.r-universe.dev

Date/Publication 2025-03-11 14:51:15 UTC

RemoteUrl https://github.com/rOpenGov/geodk

RemoteRef HEAD

RemoteSha 75fe8e0e7f2697ceede42cd5ce552993b7875787

Contents

constituencies . . . 2
geodk_enrich . . . 3
get_geo . . . 3
get_levels . . . 4
municipalities . . . 4
parishes . . . 5
plot_denmark . . . 5
plot_municipalities . . . 6
plot_regions . . . 7
prefill_cache . . . 7
regions . . . 8

Index **9**

`constituencies` *Get constituencies*

Description

Get constituencies

Usage

```
constituencies()
```

Value

Returns a `{sf}` object for the danish constituencies

See Also

Other data retrieval: `get_geo()`, `municipalities()`, `parishes()`, `regions()`

geodk_enrich *Enrich a dataframe with geographic data*

Description

This function makes it easy for you to enrich your dataframe with appropriate geographical information. The main usage, right now at least, is to enrich statistical data from {geodk} with geometries for the appropriate levels.

Usage

```
geodk_enrich(x)
```

Arguments

x	Dataframe. Preferably from {dkstat}

Value

Returns a dataframe of the dimensions cols +1 by rows.

get_geo *Get geogrpahies*

Description

Get geogrpahies

Usage

```
get_geo(section = "postnumre")
```

Arguments

section	The geographic level wanted.

Value

Returns a {sf} object for the specified geography

See Also

Other data retrieval: constituencies(), municipalities(), parishes(), regions()

get_levels *Get all available levels*

Description

Get all available levels

Usage

```
get_levels()
```

Value

This funciton returns all available levels for the `plot_*` functions.

Examples

```
get_levels()
```

municipalities *Get municipalities*

Description

Get municipalities

Usage

```
municipalities()
```

Value

Returns a `{sf}` object for the danish municipalities

See Also

Other data retrieval: `constituencies()`, `get_geo()`, `parishes()`, `regions()`

parishes *Get parishes*

Description

Get parishes

Usage

```
parishes()
```

Value

Returns a {sf} object for the danish parishes

See Also

Other data retrieval: constituencies(), get_geo(), municipalities(), regions()

plot_denmark *Plot a map of Denmark with given borders*

Description

Plot a nice map of Denmark with borders at a given level. Options to fill based on area names or color borders are included. A legend guide is shown if less than 15 areas are plotted.

Usage

```
plot_denmark(level = "regions", fill = NULL, color = NULL)
```

Arguments

level	The desired administrative geographic level for the map. Should be one of get_levels()
fill	The desired color for the areas to be filled with. Could be either a R-friendly color (name or hex code) or names to fill based on the area names.
color	The desired color for the area borders to be colored with. Could be either a R-friendly color (name or hex code) or names to fill based on the area names.

Value

Returns a ggplot object that can be manipulated as any other. The object is automatically printed.

Examples

```
plot_denmark()

plot_denmark(fill = "names")
```

`plot_municipalities` *Plot selected (or all) municipalities*

Description

Plot a vector of municipalities in Denmark. Just provide the name.

Usage

```
plot_municipalities(
  municipality = c("København", "Frederiksberg", "Ballerup", "Brøndby", "Dragør",
     "Gentofte", "Gladsaxe", "Glostrup", "Herlev", "Albertslund", "Hvidovre",
   "Høje-Taastrup", "Lyngby-Taarbæk", "Rødovre", "Ishøj", "Tårnby", "Vallensbæk",
     "Furesø", "Allerød", "Fredensborg", "Helsingør", "Hillerød", "Hørsholm",
   "Rudersdal", "Egedal", "Frederikssund", "Greve", "Køge", "Halsnæs", "Roskilde",
   "Solrød", "Gribskov", "Odsherred", "Holbæk", "Faxe", "Kalundborg", "Ringsted",
     "Slagelse", "Stevns", "Sorø", "Lejre",
      "Lolland", "Næstved", "Guldborgsund",
   "Vordingborg", "Bornholm", "Middelfart", "Christiansø", "Assens", "Faaborg-Midtfyn",
   "Kerteminde", "Nyborg", "Odense", "Svendborg", "Nordfyns", "Langeland", "ærø",
     "Haderslev", "Billund", "Sønderborg", "Tønder", "Esbjerg", "Fanø", "Varde",
    "Vejen", "Aabenraa", "Fredericia", "Horsens", "Kolding", "Vejle", "Herning",
   "Holstebro", "Lemvig", "Struer", "Syddjurs", "Norddjurs", "Favrskov", "Odder",
     "Randers", "Silkeborg", "Samsø", "Skanderborg", "Aarhus",
      "Ikast-Brande",
     "Ringkøbing-Skjern", "Hedensted", "Morsø", "Skive", "Thisted", "Viborg",
     "Brønderslev", "Frederikshavn", "Vesthimmerlands", "Læsø", "Rebild",
     "Mariagerfjord", "Jammerbugt", "Aalborg", "Hjørring")
)
```

Arguments

`municipality`	Municipality to plot. Mutiple is supported.

Value

Returns a {ggplot2} object and prints the plot as well.

Examples

```
plot_municipalities(municipality = c("Aarhus", "Favrskov"))
```

plot_regions *Plot selected (or all) regions*

Description

Plot a vector of regions in Denmark. Just provide the name.

Usage

```
plot_regions(
  region = c("Region Nordjylland", "Region Midtjylland", "Region Syddanmark",
    "Region Hovedstaden", "Region Sjælland")
)
```

Arguments

region	Region(s) to plot

Value

Returns a {ggplot2} object and prints the plot as well.

Examples

```
plot_regions(region = c("Region Nordjylland", "Region Midtjylland"))
```

prefill_cache *Prefill data retrieval cache*

Description

This function prefills the cache of the underlying API-function. By default the sections shown in the example below is prefilled. This is meant to replace the cached dataset that once was in the package. It may be useful to prefill the cache before using geodk_enrich.

Usage

```
prefill_cache(specific_section = NULL)
```

Arguments

specific_section	Enter a seciton name (or vector of section names) you would like to have prefilled. Options can be found in dawaR::avaialable_sections()

Value

Returns nothing. Is called for the side effect of prefilling the cache.

Examples

```
sections <- c(
  "regioner",
  "kommuner",
  "postnumre",
  "sogne",
  "regioner",
  "opstillingskredse",
  "afstemningsomraader"
)
```

`regions` *Get regions*

Description

Get regions

Usage

```
regions()
```

Value

Returns a `{sf}` object for the danish regions

See Also

Other data retrieval: `constituencies()`, `get_geo()`, `municipalities()`, `parishes()`

Index

∗ **data retrieval**
constituencies, 2
get_geo, 3
municipalities, 4
parishes, 5
regions, 8

constituencies, 2, *3–5*, *8*

geodk_enrich, 3
get_geo, *2*, 3, *4, 5*, *8*
get_levels, 4

municipalities, *2, 3*, 4, *5*, *8*

parishes, *2–4*, 5, *8*
plot_denmark, 5
plot_municipalities, 6
plot_regions, 7
prefill_cache, 7

regions, *2–5*, 8