Package: geodk (via r-universe)

October 28, 2024

Type Package

Title Access Danish Geospatial Data

Version 0.0.0.9000

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)

Encoding UTF-8

LazyData true

Suggests testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

Config/testthat/parallel true

Depends R (>= 3.5.0)

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

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://aleksanderbl29.r-universe.dev

RemoteUrl https://github.com/aleksanderbl29/geodk

RemoteRef HEAD

RemoteSha c950499dd7ac01aeb4ffb1e7e55575657d46a552

Contents

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"))
```

Index