

# Package: geodk (via r-universe)

December 25, 2024

**Type** Package

**Title** Access Danish Geospatial Data

**Version** 0.0.0.9000

**Date** 2024-11-25

**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 (>= 3.5.0)

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

**Suggests** testthat (>= 3.0.0), vdiff

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**X-schema.org-isPartOf** <http://ropengov.org/>

**X-schema.org-keywords** ropengov

**Repository** <https://ropengov.r-universe.dev>

**RemoteUrl** <https://github.com/rOpenGov/geodk>

**RemoteRef** HEAD

**RemoteSha** d2a0a65f0a81343373ae231a91fcc9ae277d134a

## Contents

get_levels . . . . .	2
municipalities . . . . .	2
plot_denmark . . . . .	3
plot_municipalities . . . . .	4
plot_regions . . . . .	5
regions . . . . .	5
<b>Index</b>	<b>7</b>

---

get_levels	<i>Get all available levels</i>
------------	---------------------------------

---

### Description

Get all available levels

### Usage

```
get_levels()
```

### Value

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

### Examples

```
get_levels()
```

---

municipalities	<i>Danish municipalities</i>
----------------	------------------------------

---

### Description

A sf dataset providing geometry for the danish municipalities

### Usage

```
municipalities
```

**Format**

municipalities:

An object of classes `sf` and `data.frame` with 99 rows and 9 columns:

**code** Unique code

**name** The Danish name of the municipality

**region\_code** The unique code of the region that this municipality is located in

**region\_name** The name of the region that this municipality is located in

**source\_changed** When was the source last changed on the data-provider's side

**visualcenter\_x, visualcenter\_y** Visual center of the polygon in WGS84

**geometry** Polygons for each municipality

**last\_update** When was the data in the package last updated by maintainers

**Source**

Provided by The Danish Agency of Climate Data <https://dawadocs.dataforsyningen.dk/dok/om#vilkaar>.  
Terms and conditions apply.

---

plot_denmark	<i>Plot a map of Denmark with given borders</i>
--------------	---

---

**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 <code>get_levels()</code>
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-Brandø",
    "Ringkøbing-Skjern", "Hedensted", "Morsø", "Skive", "Thisted", "Viborg",
    "Brønderslev", "Frederikshavn", "Vesthimmerlands", "Læsø", "Rebild",
    "Mariagerfjord", "Jammerbugt", "Aalborg", "Hjørring")
)
```

## Arguments

municipality     Municipality to plot. Multiple is supported.

## Value

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

## Examples

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

---

plot_regions	<i>Plot selected (or all) regions</i>
--------------	---------------------------------------

---

**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"))
```

---

regions	<i>Danish regions</i>
---------	-----------------------

---

**Description**

A sf dataset providing geometry for the danish regions

**Usage**

```
regions
```

**Format**

regions:

An object of classes sf and data.frame with 5 rows and 8 columns:

**code** Unique code

**name** The Danish name of the region

**nuts2** EU nuts division

**source\_changed** When was the source last changed on the data-provider's side

**visualcenter\_x, visualcenter\_y** Visual center of the polygon in WGS84

**geometry** Polygons for each region

**last\_update** When was the data in the package last updated by maintainers

**Source**

Provided by The Danish Agency of Climate Data <https://dawadocs.dataforsyningen.dk/dok/om#vilkaar>.  
Terms and conditions apply.

# Index

## \* datasets

- municipalities, [2](#)
- regions, [5](#)

[get\\_levels](#), [2](#)

[municipalities](#), [2](#)

[plot\\_denmark](#), [3](#)

[plot\\_municipalities](#), [4](#)

[plot\\_regions](#), [5](#)

[regions](#), [5](#)