Package ‘plotDK’

October 1, 2021

Title  Plot Summary Statistics as Choropleth Maps of Danish Administrative Areas

Description  Provides a ggplot2 front end to plot summary statistics on danish provinces, regions, municipalities, and zipcodes. The needed geoms of each of the four levels are inherent in the package, thus making these types of plots easy for the user. This is essentially an updated port of the previously available ‘mapDK’ package by Sebastian Barfort.

Version  0.1.0
Depends  R (>= 4.0)
License  MIT + file LICENSE
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
Imports  dplyr, ggplot2 (>= 3.1.0), mapproj, plotly, purrr, rlang
Suggests  testthat
NeedsCompilation  no
Author  Kristian Stendorff Nielsen [aut, cre]
Maintainer  Kristian Stendorff Nielsen <krisdorff@hotmail.com>
Repository  CRAN
Date/Publication  2021-10-01 08:00:02 UTC

R topics documented:

create_plot_data  .............................................................. 2
municipality  ............................................................... 3
municipality_info  .......................................................... 4
plotDK  ................................................................. 4
province  ................................................................. 6
province_info  ............................................................ 6
region  ................................................................. 7
region_info  ............................................................. 8
zipcodes  ................................................................. 8
zipcode_info  ............................................................ 9
create_plot_data

Description

If data is provided, this function attempts to merge the provided data with the geom-data inherent in the package on the chosen plot-level. If no data is provided, only the geom-data for the chosen plot-level is returned. This data is used to create the DK-plot.

Usage

create_plot_data(data, id, plotlevel, show_missing = FALSE, ...)

Arguments

data: A data.frame containing an ID-variable specifying either a municipality, region, province or zipcode (see id), as well as a value-variable containing any value to be plotted on the chosen level.

id: A character specifying the name of a column in data containing the ID on the chosen level. For municipalities these variables can be either:
• A character-variable with danish municipality names. For accepted values see municipality_info.
• A numeric/integer-variable with official municipality numbers. For accepted values see municipality_info.

For regions these variables can be either:
• A character-variable with danish region names. For accepted values see region_info.
• A numeric/integer-variable with danish region numbers. For accepted values see region_info.

For provinces these variables can be either:
• A character-variable with danish province names. For accepted values see province_info.
• A numeric/integer-variable with danish province numbers. For accepted values see province_info.

For zip-codes these variables can be;
• A numeric/integer-variable with danish zip-codes. For accepted values see zipcode_info.

plotlevel: character, indicating which level to plot. Valid options are "municipality", "region", "province", and "zipcode".

show_missing: logical. Should levels not present in data or with NA-values be printed? This can be used to plot only a subset of entities.

... Further arguments to pass to merge_data
**Value**

A `data.frame` with either raw geom-data or geom-data merged with any data provided.

---

**municipality**

`Municipality data with keys and polygon-geoms for municipalities of Denmark`

---

**Description**

Municipality data with keys and polygon-geoms for municipalities of Denmark

**Usage**

`municipality`

---

**Format**

A data frame with 39,230 rows and 7 columns:

- **long** Longitude coordinates.
- **lat** Latitude coordinates.
- **order** Order of coordinates in geoms.
- **group** Geom groups.
- **id** Name of entity.
- **id_numeric** Number of entity.
- **hole** Indication of a geom hole.

**Source**

Statistics Denmark

---

**municipality_info**

*Information of Valid Municipality Names and Numbers*

---

**Description**

Information of Valid Municipality Names and Numbers

**Usage**

`municipality_info`
Format

A data frame with 99 rows and 2 columns:

- **municipality_names**: Valid names of municipalities.
- **municipality_numbers**: Valid numbers of municipalities.

Source

Statistics Denmark

---

**plotDK**

Generate choropleth maps of Danish Municipalities, Regions, Provinces and Zip-areas

---

Description

Draws a map that highlights any value of interest across either Danish municipalities, regions, provinces or zip-codes. This is essentially a ggplot2-wrapper incorporating geoms of Danish municipalities, regions, provinces and zipcodes. Therefore the output is compatible with further ggplot2 elements.

Usage

```r
plotDK(
  data = NULL,
  id = NULL,
  value = NULL,
  plotlevel = "municipality",
  show_missing = FALSE,
  show_borders = TRUE,
  interactive = FALSE,
  titel = NULL
)
```

Arguments

- **data**: A `data.frame` containing an ID-variable specifying either a municipality, region, province or zipcode (see `id`), as well as a value-variable containing any value to be plotted on the chosen level.
- **id**: A character specifying the name of a column in data containing the ID on the chosen level.

For municipalities these variables can be either:

- A character-variable with Danish municipality names. For accepted values see `municipality_info`.
- A numeric/integer-variable with official municipality numbers. For accepted values see `municipality_info`.

---
For regions these variables can either:
  • A character-variable with danish region names. For accepted values see
    \texttt{region\_info}.
  • A numeric/integer-variable with danish region numbers. For accepted val-
    ues see \texttt{region\_info}.

For provinces these variables can be either:
  • A character-variable with danish province names. For accepted values see
    \texttt{province\_info}.
  • A numeric/integer-variable with danish province numbers. For accepted
    values see \texttt{province\_info}.

For zip-codes these variables can be:
  • A numeric/integer-variable with danish zip-codes. For accepted values see
    \texttt{zipcode\_info}.

\begin{itemize}
  \item \texttt{value} \texttt{numeric}-, \texttt{factor}- or \texttt{character}-variabel to be plotted on the map. Note that \texttt{character-variables} will be naively translated to factors behind the scenes. For full control over levels, pre-convert to a factor.
  \item \texttt{plotlevel} \texttt{character}, indicating which level to plot. Valid options are "municipality", "region", "province", and "zipcode".
  \item \texttt{show\_missing} \texttt{logical}. Should levels not present in data or with NA-values be printed? This can be used to plot only a subset of entities.
  \item \texttt{show\_borders} \texttt{logical}. Should geom borders be drawn?
  \item \texttt{interactive} \texttt{logical}. Should the plot be converted to an interactive plotly plot?
  \item \texttt{titel} \texttt{character}. Optional plot title.
\end{itemize}

\textbf{Value}

A \texttt{ggplot} object.

\textbf{Examples}

\begin{verbatim}
## Empty plot
plotDK()

province_data <- data.frame(
  province_name = c(
    "nordjylland",
    "østjylland",
    "vestjylland",
    "fyn",
    "sydjylland",
    "vest-ogsydsjælland",
    "østsjælland",
    "københavnsomegn",
    "byenkøbenhavn",
    "bornholm",
    "nordsjælland"
})
\end{verbatim}
province  

Province data with keys and polygon-geoms for provinces of Denmark

Description
Province data with keys and polygon-geoms for provinces of Denmark

Usage
province

Format
A data frame with 4,083 rows and 7 columns:

- **long**: Longitude coordinates.
- **lat**: Latitude coordinates.
- **order**: Order of coordinates in geoms.
- **group**: Geom groups.
- **id**: Name of entity.
- **id_numeric**: Number of entity.
- **hole**: Indication of a geom hole.

@source Statistics Denmark

province_info  

Information of Valid Province Names and Numbers

Description
Information of Valid Province Names and Numbers

Usage
province_info
Format

A data frame with 11 rows and 2 columns:

- **province_names** Valid names of provinces.
- **province_numbers** Valid numbers of provinces.

Source

Statistics Denmark

---

<table>
<thead>
<tr>
<th>region</th>
<th>Region data with keys and polygon-geoms for regions of Denmark</th>
</tr>
</thead>
</table>

Description

Region data with keys and polygon-geoms for regions of Denmark

Usage

region

Format

A data frame with 32,522 rows and 7 columns:

- **long** Longitude coordinates.
- **lat** Latitude coordinates.
- **order** Order of coordinates in geoms.
- **group** Geom groups.
- **id** Name of entity.
- **id_numeric** Number of entity.
- **hole** Indication of a geom hole.

Source

Statistics Denmark
### region_info

**Description**

Information of Valid Region Names and Numbers

**Usage**

`region_info`

**Format**

A data frame with 5 observations and 2 columns:

- `region_names` Valid names of regions.
- `region_numbers` Valid numbers of regions.

**Source**

Statistics Denmark

---

### zipcodes

**Description**

Zipcode data with keys and polygon-geoms for zipcodes of Denmark

**Usage**

`zipcodes`

**Format**

A data frame with 49,322 rows and 7 columns:

- `long` Longitude coordinates.
- `lat` Latitude coordinates.
- `order` Order of coordinates in geoms.
- `group` Geom groups.
- `id` Name of entity.
- `id_numeric` Number of entity.
- `hole` Indication of a geom hole.

**Source**

Statistics Denmark
Description

Information of Valid Zipcodes

Usage

zipcode_info

Format

A data frame with 598 rows and 1 column:

zipcode_numbers  Valid numbers of zipcodes.

Source

Statistics Denmark
Index

* datasets
  municipality, 3
  municipality_info, 3
  province, 6
  province_info, 6
  region, 7
  region_info, 8
  zipcode_info, 9
  zipcodes, 8

create_plot_data, 2

municipality, 3
municipality_info, 2, 3, 4

plotDK, 4
province, 6
province_info, 2, 5, 6

region, 7
region_info, 2, 5, 8

zipcode_info, 2, 5, 9
zipcodes, 8