Package 'voteR'

January 22, 2018

Title Variety of Open Tools for Electoral Research

Version 0.0.0.9000

Description This package does this and that vfb stuttgart lalala

Depends R (>= 3.4.1),
ggplot2,
plyr,
tidyverse

License MIT

Encoding UTF-8

URL https://www.prognosophie.de

BugReports https://www.prognosophie.de/impressum

LazyData true

Imports dplyr,
ggplot2,
gtools,
hrbrthemes,
magrittr,
stringr,
tidyr

RoxygenNote 6.0.1

Collate 'create_manual.R'
'gles.R'
'importFrom.R'
'partycolors.R'
'partynames.R'
'polling.R'
'plotting.R'

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`distance_function` *Calculate Distances*

Description

Calculate Distances from different parties/koalitions

Usage

```
distance_function(data_in = "gles2017_out", who = "schwarzgelb",
  issue = "soz")
```

Value

The vector including all coalitions.

Examples

```
distance_function(data_in = "gles2017_out",
                  who = "schwarzgelb",
                  issue = "soz")
```

`gles_recode_partyvar` *gles_recode_partyvar*

Description

Recode a multiparty-variable in GESIS-Datasets such as the GERMAN LONGITUDINAL ELECTION STUDY (GLES)

Usage

```
gles_recode_partyvar(year = 2017, dataset_input = "gles2017",
  dataset_output = "gles2017_out", varname = "q52", own = NULL,
  varlabel = "soz", key = c("a", "b", "c", "d", "e", "f", "g"),
  partynames = c("cdu", "csu", "spd", "linke", "gruene", "fdp", "afd"),
  NAs = "<0", plot = TRUE)
```

Arguments

year	year the GLES-study is from. Defaults to 2017.
dataset_input	Character string of the name of a dataframe containing the raw data.
dataset_output	Character string of the name the output data frame (may already exist or not).
varname	Character string of the name of the original variable.
own	May apply: Different variable name for own position (on left-right scales, e.g.)
varlabel	Character string of the to-be-assigned variable label.
key	Character vector containing original alphabetic party keys.
partynames	Character vector containing shortname party keys.
NAs	Numeric vector containing to-be-assigned NAs/Missing values.
plot	Logical T/F: Show relative frequency barplots while plotting.

Value

A data frame containing output dataframe including newly appended new-variables.

Examples

```
gles_recode_partyvar(year = 2017,
                     dataset_input = "gles2017",
                     dataset_output = "gles2017_out",
                     varname = "q52",
                     varlabel = "soz",
                     key = c("a","b","c","d","e","f","g"),
                     partynames = c("cdu","csu","spd","linke","gruene","fdp","afd"),
                     NAs = c(-97,-98,-99))
```

---

koas *Get coalitions*

Description

Get all available coalitions

Usage

```
koas(year = 2017)
```

Value

The vector including all coalitions.

Examples

```
koas(year = 2017)
```

koa_members *Get coalition members*

Description

Get parties that are member of a certrain coalition

Usage

```
koa_members(koalition)
```

Arguments

coalition	Character string containing the name of the coalition. Options are c("jamaika","schwarzgelb","rotgruen","groko","rotrotgruen","ampel","schwarzgruen").

Value

A vector containing all parties included in the coalition.

Examples

```
koa_members("schwarzgelb")
```

koa_positions *Get coalition members*

Description

Calculate mean koalition issue position and create new variables

Usage

```
koa_positions(data_in = "gles2017_out", coalition = "schwarzgelb",
  issue = "soz")
```

Arguments

data_in	Character string containing the name the dataset.
coalition	Character string containing the name of the coalition.
issue	Character string containing the issue.

Value

The treated dataset.

Examples

```
koa_members(data_in = "gles2017_out",
            coalition = "schwarzgelb",
            issue = "soz")
```

parties *Get parties*

Description

Get main parties for Gles analysis

Usage

```
parties(year = 2017)
```

Value

The vector including all parties

Examples

```
parties(year = 2017)
```

plot_poll *Plot a multiparty poll*

Description

...

Usage

```
plot_poll(vote = c(cdu = 0.33, spd = 0.2, fdp = 0.11, linke = 0.09, gruene =
  0.09, afd = 0.12, sonstige = 0.05), order = "alphabetical",
  sample_confidence_bounds = TRUE, sample_n = 1000, n_draw = 10000,
  show_quantiles = c(0.05, 0.95), round = 1, xlab = "Party",
  ylab = "Voteshare", title = "Title", subtitle = "Subtitle",
  caption = "Caption", theme_ipsum = FALSE, grid = "Y")
```

Arguments

vote	A labeled party vote share vector.
order	Method to order parties (Default is "alphabetical"; also takes "descending" and "ascending" as well as manual specification of party vector)
sample_confidence_bounds	Logical T/F: add empirical dirichlet quantiles
sample_n	The number of observations in the poll sample.
n_draw	How many samples to draw from the dirichlet distribution.
show_quantiles	Vector of quantiles/confidence boundaries to calculate.
round	Round to k decimals after comma
xlab	x-label string

ylab	y-label string
title	title string
subtitle	subtitle string
caption	caption string
theme_ipsum	Pre-applies nice theme from the hrbrthemes-package.(Attention: possible font-issues when Roboto font is not installed on your computer.)
grid	(Applies only if theme_ipsum == T) Add a grid (options: "none","Y")

Value

A data frame containing n rows of samples for each party.

Warning

Do not operate heavy machinery within 8 hours of using this function.

Examples

```
sample_dirichlet_quantiles(vote = c(cdu = 0.33,....
```

---

sample_dirichlet *Dirichlet-sample of a multinomial election poll*

Description

Calculate a dirichlet-sample of a multinomial election poll

Usage

```
sample_dirichlet(vote = c(cdu = 0.5, spd = 0.4, fdp = 0.1), sample_n = 1000,
  n_draw = 10000)
```

Arguments

vote	A labeled party vote share vector.
sample_n	The number of observations in the poll sample.
n_draw	How many samples to draw from the dirichlet distribution.

Value

A data frame containing n rows of samples for each party.

Warning

Do not operate heavy machinery within 8 hours of using this function.

Examples

```
sample_dirichlet(vote = c(cdu = 0.33,
                          spd = 0.20,
                          fdp = 0.11,
                          linke = 0.09,
                          gruene = 0.09,
                          afd = 0.12,
                          sonstige = 0.05),
                 sample_n = 1000,
                 n_draw = 10000)
```

`sample_dirichlet_quantiles` *Empirical Dirichlet Quantiles from Multinomial Election Poll*

Description

Calculate empirical quantiles from a sample created by `sample_dirichlet` of a multinomial election poll

Usage

```
sample_dirichlet_quantiles(vote = c(cdu = 0.5, spd = 0.4, fdp = 0.1),
  sample_n = 1000, n_draw = 10000, show_mean = TRUE,
  show_quantiles = c(0.05, 0.95), round = 2)
```

Arguments

`vote`	A labeled party vote share vector.
`sample_n`	The number of observations in the poll sample.
`n_draw`	How many samples to draw from the dirichlet distribution.
`show_mean`	Logical T/F: Show sample mean.
`show_quantiles`	Vector of quantiles/confidence boundaries to calculate.
`round`	Logical T/F Round Results to k decimal digits.

Value

A data frame containing `n` rows of samples for each party.

Warning

Do not operate heavy machinery within 8 hours of using this function.

Examples

```
sample_dirichlet_quantiles(vote = c(cdu = 0.33,
                           spd = 0.20,
                           fdp = 0.11,
                           linke = 0.09,
                           gruene = 0.09,
                           afd = 0.12,
                           sonstige = 0.05),
                  sample_n = 1000,
                  n_draw = 10000,
                  show_mean = TRUE,
                  show_quantiles = c(0.05,0.95))
```

Index