Package 'cspp'

October 17, 2021

Type Package

Title A Tool for the Correlates of State Policy Project Data

Version 0.3.2

Author Caleb Lucas (https://caleblucas.com/) and Joshua McCrain (http://joshuamccrain.com/)

Maintainer Caleb Lucas <calebjlucas@gmail.com>

Description A tool that imports, subsets, visualizes, and exports the Correlates of State Policy Project dataset assembled by Marty P. Jordan and Matt Grossmann (2020) <http://ippsr.msu.edu/public-policy/correlates-state-policy>. The Correlates data contains over 2000 variables across more than 100 years that pertain to state politics and policy in the United States. Users with only a basic understanding of R can subset this data across multiple dimensions, export their search results, create map visualizations, export the citations associated with their searches, and more.

License GPL-3

Encoding UTF-8

LazyData true

Language en-US

Depends R (>= 2.10), dplyr(>= 1.0.0)

Imports stringr, readr, tidyselect, ggplot2, mapproj, rlang, haven, purrr, csppData, ggcorrplot

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, testthat, ggraph, igraph

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-10-16 23:30:19 UTC

R topics documented:

`corr_plot` *Create correlation plots of CSPP data*

Description

`corr_plot` takes CSPP data from `get_cspp_data` and returns either a correlation matrix or correlation plot.

Usage

```
corr_plot(
  data = NULL,
  vars = NULL,
  summarize = TRUE,
  labels = TRUE,
  label_size = 3,
  colors = c("#6D9EC1", "#FFFFFF", "#E46726"),
  cor_matrix = FALSE
)
```

Arguments

`data`	A dataframe. If data is generated by `get_cspp_data` function, the function can automatically parse the dataframe. Otherwise, this function will attempt to make a correlation plot or matrix from all numeric variables within the passed dataframe.
`vars`	Default is NULL. If left NULL, uses all variables within the passed dataframe. Otherwise, must be a character vector. The dataframe is subset based on variables listed.
`summarize`	Default is TRUE. If TRUE, and if the variable `st` is present, the function will create state specific averages for each variable in the dataframe. If FALSE, the function will generate the correlation matrix and plot for all values in the dataset.
`labels`	Default is TRUE. If TRUE, the correlation plot will include labels for the correlation value. If FALSE, no labels will be present.
`label_size`	Default is 3. Controls the size of the font for labels.

colors	Specify the colors to be used in the correlation plot. Must include three values in a character vector format. The default values are ‘c("#6D9EC1", "#FFFFFF", "#E46726")‘.
cor_matrix	Default is FALSE. If set to TRUE, instead of returning a ggplot object that is a correlation plot, returns a correlation matrix. This is particularly useful if you want to customize the output with ggcorrplot.

Details

This function is a wrapper that passes a dataframe to the ggcorrplot::ggcorrplot function which generates correlation heat plots.

Value

ggplot2 object or correlation matrix

See Also

ggcorrplot

Examples

```
corr_plot(data = get_cspp_data(), vars = c("pollib_median",
 "innovatescore_boehmkeskinner", "citi6013", "ranney4_control", "h_diffs"),
 cor_matrix = FALSE)
```

---

generate_map *Generate map visualizations (choropleths) of CSPP data*

Description

generate_map takes CSPP data from get_cspp_data and plots the values of numeric variables on the map of the U.S. It can also plot individual states or sets of states.

Arguments

cspp_data	Dataframe generated by get_cspp_data which must include the variable state. If there are multiple years of data per state, by default the most recent year is used in creating the map unless average_years is set to TRUE. Default is NULL and returns the most recent year's poptotal data as an example map.
var_name	Specify the variable from the dataset passed to cspp_data to plot on the map. If left blank, the first variable that is not "year", "st", "state", "state_fips", or "state_icspr" is used. Default is NULL.
average_years	Default is FALSE. If TRUE, averages over all of the years per state in the dataframe to produce a value to plot on the map. If the type of the variable in var_name is not numeric, will reset this parameter to FALSE.

`drop_NA_states` Choose whether to drop states at the map generating stage which have NA values. Default is `FALSE` and states with missing data will be filled grey. If set to `TRUE`, states will have no fill in the plot.

If you're passing a dataframe subset to certain states, set this to TRUE.

`poly_args` Default is `list(color = "#666666", size = .5)`. Changes the aesthetics of how the states look when plotted. The `fill` of each state can be manually changed through ggplot's `scale_fill_` (see examples). See `geom_polygon` for other options to pass to this argument.

Details

Note: due to complications with plotting Alaska and Hawaii, this package currently does not support plotting these two states.

This function is general in the sense that it will produce a ggplot-style map for any dataframe passed to it with the proper formatting. Any dataframe that has at least three columns, with the first two a numeric 'year' column and a state name as a string, and the final column the value to be plotted, will work with this function.

Value

Returns a `ggplot` object. See examples for how to work with this object.

See Also

`get_cspp_data`, `get_cites`, `get_var_info`

Examples

```
## default map with total population
generate_map()

## pass specific variables
# returns average over all non NA years in the data
generate_map(get_cspp_data(var_category = "demographics"),
             var_name = "pctpopover65")

## add additional ggplot options
generate_map(get_cspp_data(var_category = "demographics"),
             var_name = "pctpopover65",
             poly_args = list(color = "black"),
             drop_NA_states = FALSE) +
 ggplot2::scale_fill_gradient(low = "white", high = "red") +
 ggplot2::theme(legend.position = "none") +
 ggplot2::ggtitle("% Population Over 65")

## plot specific states
# drop_NA_states set to TRUE plots only those states
library(dplyr)
generate_map(get_cspp_data(var_category = "demographics") %>%
```

```
              dplyr::filter(st %in% c("NC", "VA", "SC")),
            var_name = "pctpopover65",
            poly_args = list(color = "black"),
            drop_NA_states = TRUE) +
 ggplot2::scale_fill_gradient(low = "white", high = "red") +
 ggplot2::theme(legend.position = "none") +
 ggplot2::ggtitle("% Population Over 65")

## pass specific variables and years
# returns average over set of years provided
library(dplyr)
generate_map(get_cspp_data(var_category = "demographics") %>%
 dplyr::filter(year %in% seq(2001, 2010)))

# returns average over set of years provided
library(dplyr)
generate_map(get_cspp_data(var_category = "demographics") %>%
 dplyr::filter(year %in% seq(2001, 2010)))
```

`get_cites` *Get citations for CSPP variables*

Description

`get_cites` retrieves citations for variables in the CSPP dataset. Users can print the citations to the console, save them as dataframes, and write them to multiple file types (csv, txt). Citations can be written in one of multiple formats (plaintext, bib). Supply variable names that need to be cited with the `var_names` argument. The function prints user-supplied variable names that do not match any in the CSPP dataset by default (`print_nomatch`). The function also returns the citation for the `cspp` package and the CSPP dataset as a whole. We request you cite both if you use this package for your research.

Usage

```
get_cites(
  var_names,
  write_out = FALSE,
  file_path = NULL,
  format = "bib",
  print_cites = FALSE,
  print_nomatch = TRUE
)
```

Arguments

`var_names`	Default is NULL. Takes a character string. Should be one or more variables from the CSPP dataset. A citation for each variable is returned.

write_out	Default is FALSE. Takes a logical. If FALSE the function does not write the citations out to a file.
file_path	Default is NULL. Takes a character string. If write_out = T then the file will be saved to this filepath.
format	Default is bib. Takes a character string. If write_out = T then the resulting file will be in this format. User must supply "bib", "csv", or "txt".
print_cites	Default is FALSE. Takes a logical value. If TRUE then the function prints the citations to the console.
print_nomatch	Default is TRUE. Takes a logical value. If FALSE then the function does not print variables the user supplied that had no match in CSPP.

See Also

get_cspp_data, get_var_info, generate_map

Examples

```
get_cites("poptotal")

## Not run:
get_cites(var_names = "poptotal",
          write_out = TRUE,
          file_path = "~/path/to/file.csv",
          format = "csv")

## End(Not run)
```

---

get_cspp_data *Load CSPP data into the R environment*

Description

get_cspp_data loads either a full or subsetted version of the full CSPP dataset into the R environment as a dataframe.

Usage

```
get_cspp_data(
  vars = NULL,
  var_category = NULL,
  states = NULL,
  years = NULL,
  core = FALSE,
  output = NULL,
  path = ""
)
```

Arguments

vars	Default is NULL. If left blank, returns all variables within the dataset. Takes a string or vector of strings. See get_var_info for pulling variable names and get_cites for citations of specific variables and datasets. Names of variables must be exact matches to variables in the dataset.
var_category	Default is NA. If left blank, returns all datasets. Takes a string or vector of strings. Options are one of, or a combination of: "demographics", "economic-fiscal", "government", "elections", "policy_ideology", "criminal justice", "education", "healthcare", "welfare", "rights", "environment", "drug-alcohol", "gun control", "labor", "transportation", "misc. regulation"
states	Default is NULL. If left blank, returns all states. Takes a string or vector of strings of state abbreviations. Use state.abb to load state abbreviations into the R environment.
years	Default is NULL. If left blank, returns all years. Coverage begins at 1900 and runs to 2019. However, coverage depends on the specific variable – see get_var_info. Input can be a vector of years (or a singular year), such as c(2000, 2001, 2002, 2012) or seq(2000, 2012).
core	Default is FALSE. If TRUE, merge the core CSPP data (approximately 70 common and important variables) with the search result.
output	Default is NULL. One of "csv", "dta", "rdata". Optional parameter for writing the resulting dataframe to a file.
path	The directory to write the file to. Default is blank, so writes to working directory. Exclude final slash: e.g., path = "dir1/dir2"

See Also

get_var_info, get_cites, generate_map

Examples

```
## returns full dataset
data <- get_cspp_data()

## use variable names from get_var_info
data <- get_cspp_data(vars = get_var_info(var_names="pctpop")$variable)

## return subsets
# note: this returns the specific variables listed as well as those in the
# var_category argument
data <- get_cspp_data(vars = c("sess_length", "hou_majority", "term_length"),
                      var_category = "demographics",
                      states = c("NC", "VA", "GA"),
                      years = seq(1995, 2004))
```

get_network_data *Get state networks data*

Description

network_data returns a dataframe of the state networks data compiled by the Correlates of State Policy Project. The dataframe is in an edge list format, with each row a state dyad combination. The merge argument allows the direct merging of a dataframe generated by the get_cspp_data function.

Usage

```
get_network_data(category = NULL, merge_data = NULL)
```

Arguments

category	A category within the networks data. Default is NULL. If left blank, returns the full state networks data. Category options are "Distance Travel Migration", "Economic", "Political", "Policy", "Demographic".
merge_data	Default is NULL. Takes a dataframe object in the format generated by get_cspp_data. The function merges this dataframe into the network data by state. If the merge dataframe has multiple observations per state, this function averages over all values per state as long as the variables are numeric. If the dataframe passed has multiple values per state and some are not numeric, only numeric variables are merged.

Details

The network dataframe that results is directed, with variables directed towards the state in the State1 column. For instance, the IncomingFlights variable is the number of flights from State2 with a destination in State1.

Value

A dataframe formatted as an edge list.

See Also

For more information on the construction of the network data as well as a full codebook see http://ippsr.msu.edu/public-policy/state-networks.

Examples

```
# Load full network data:
network.df <- get_network_data()

# Network data for subset of categories:
```

```
network.df <- get_network_data(category = c("Economic", "Political"))

# Merge in data from get_cspp_data()
network.df <- get_network_data(category = "Distance Travel Migration",
                     merge_data  = get_cspp_data(vars = c("sess_length", "hou_majority"),
                                                 years = seq(1999, 2000)))
```

`get_var_info` *Get information regarding the CSPP variables*

Description

`get_var_info` retrieves information regarding variables in the CSPP dataset. The information available includes: the years each variable is observed in the data; a short and long description of each variable; the source and citation for each variable; and a general category that describes each variable.

Usage

```
get_var_info(
  var_names = NULL,
  categories = NULL,
  related_to = NULL,
  exact = FALSE
)
```

Arguments

`var_names`	Default is NULL. Takes a character string. If left blank the function does not subset by variable name.
`categories`	Default is NULL. Takes a character string. If left blank the function does not subset by category.
`related_to`	Default is NULL. Takes a character string. If the user supplies a character string, the function searches the other relevant fields (variable name, short/long description, and source) for string and returns either exact or partial matches depending on the value of the `exact` argument.
`exact`	Default is FALSE. If true, exact matches for the other supplied arguments are used. If TRUE, then partial matches are also returned.

Details

Users can request this information regarding specific variables or all the variables within a specific category. Users can request exact matches of their supplied arguments or allow partial matches with the `exact` argument. Users can also search all these relevant fields (variable name, short/long description, source) for a keyword/s with the supply a string `related_to` argument to identify variables related to a topic of interest.

Specifying no arguments returns all the information for all the variables in the CSPP dataset.

See Also

get_cspp_data, get_cites, generate_map

Examples

```
# returns all variable information
get_var_info()

# searches all columns for non-exact matches of "pop" and "fem"
get_var_info(related_to = c("pop","femal"))

get_var_info(categories = "demographics")

# returns non-exact matches for variables with "pop" and that have "femal" anywhere in the row
get_var_info(var_names = "pop",
             related_to = "femal")
```

---

map_example *Sample Dataset for Working with generate_map()*

Description

A dataframe to create a sample map using the generate_map function. The variable plotted is population.

Usage

```
map_example
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 3 columns.

Details

@name map_example

@docType data

@usage data(map_example)

@keywords datasets

network_data *State Network data (IPPSR)*

Description

The State Networks dataset is a compilation of many state-to-state relational variables, including measures of shared borders, travel and trade between states, and demographic characteristics of state populations collected by Shayla Olson (2020) and Marty P. Jordan and Matt Grossmann (2020) <http://ippsr.msu.edu/public-policy/state-networks>.

Usage

```
network_data
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2550 rows and 120 columns.

Details

@name network_data

@docType data

@usage data(network_data)

@keywords datasets

network_vars *State Network (IPPSR) Dataset Variable Names*

Description

A dataset of the the names of the variables in the IPPSR state networks data.

Usage

```
network_vars
```

Format

An object of class `data.frame` with 118 rows and 2 columns.

Details

@name network_vars

@docType data

@usage data(network_vars)

@keywords datasets

plot_panel *Generate time series plots of CSPP data*

Description

plot_panel takes CSPP data from get_cspp_data and plots the values of the passed variable name in a time series (grid or line) format.

Usage

```
plot_panel(
  cspp_data = NULL,
  var_name = NULL,
  years = NULL,
  colors = c("#b3a4a4", "#8f3838", "#dbdbdb")
)
```

Arguments

cspp_data	Dataframe generated by get_cspp_data which must include the variable st.
var_name	Specific variable within the dataframe passed to 'cspp_data' to plot. If left NULL, will automatically plot the first variable after state identifiers.
years	Specify years within the passed dataframe to plot. If left NULL, will plot all years for which not all observations have missing values. Takes a vector of years.
colors	Specify the colors to be used in a grid plot. Must include three values in a character vector format. The default values are 'c("#b3a4a4", "#8f3838", "#dbdbdb")'. If the variable plotted is dichotomous, the first color is the non-treated value and the second color is the treated value. The third color is the value for NA. If plotting a continuous variable, the first color is the low end of the gradient and the second value is the high end of the gradient. See scale_fill_gradient.

Details

This function will take any dataframe consisting of the variables 'year' and 'st' plus one other variable.

Value

ggplot2 object

See Also

get_var_info, get_cites, generate_map

Examples

```
# dichotomous variable
cspp <- get_cspp_data(vars = c("drugs_medical_marijuana"))
plot_panel(cspp)

# change colors and years
plot_panel(cspp, colors = c("white", "blue", "black"),
                 years = seq(1980, 2000))

# continuous variable with missing data:
continuous_data <- get_cspp_data(vars = c("h_diffs"))

plot_panel(continuous_data, colors = c("white", "dodgerblue", "#eeeeee"))

# add ggplot2 features
library(ggplot2)
plot_panel(continuous_data, colors = c("white", "dodgerblue", "#eeeeee")) +
  theme(legend.position = "none") +
  ggplot2::ggtitle("Continuous variable")
```

`var_names_db` *Correlates of State Policy Project Dataset (IPPSR) Variable Names*

Description

A dataframe of all variable names, their descriptions, and sources in the Correlates of State Policy Project Dataset.

Usage

```
var_names_db
```

Format

An object of class `data.frame` with 2179 rows and 3 columns.

Details

@name var_names_db

@docType data

@usage data(var_names_db)

@keywords datasets

Index