Package 'Setviz'

December 10, 2019

Type Package

Title Visualise intersecting sets with weights

Version 0.1.0

Description This package lets you visualise intersecting sets.
It is a wrapper for the upsetR package, and can be used in combination with weighted percentages from teh svydesign package. Its functions take a list of binary variables in a dataframe, and show you what combinations of variables occur in what percentages.

License What license is it under?

Encoding UTF-8

LazyData true

Imports dplyr, roxygen2, UpSetR, magrittr, survey

Suggests hypegrammaR,
testthat,
knitr,
rmarkdown

RoxygenNote 6.1.1

VignetteBuilder knitr

R topics documented:

Index

---

add_set_intersection_to_df

Create a vector containing the weighted percentages in each set

---

Description

Create a vector containing the weighted percentages in each set

Usage

```
add_set_intersection_to_df(data, varnames, exclude_unique = T)
```

Arguments

data	a dataframe containing all the sets for each record, coercible to 1,0
varnames	a vector containing the names of variables to be used in the intersection
exclude_unique	whether the set intersections should include singular sets (i.e. that one variable). Note that if this is set to True, the total set size on the left will be wrong

Value

A vector of the aggregated percent for each intersection

---

expand_to_set_intersections

Expand the binary indicators to set intersections

Description

Expand the binary indicators to set intersections

Usage

```
expand_to_set_intersections(data, varnames)
```

Arguments

data	a dataframe containing all the 1,0 indicators in varnames
varnames	a vector containing the names of variables to be used in the intersection

Value

The dataframe containing the intersections of all variables and the names of those combinations of varnames with '&'

---

plot_set_percentages   *Create a plot from a dataset and variable names combining the make_set_percentages and set_percentage_plot functions*

Description

Create a plot from a dataset and variable names combining the make_set_percentages and set_percentage_plot functions

Usage

```
plot_set_percentages(data, varnames, weight_variable = NULL,
  weighting_function = NULL, nintersects = 12, exclude_unique = T,
  label = NULL)
```

Arguments

`data`	a dataframe containing all the 1,0 indicators in varnames
`varnames`	a vector containing the names of variables to be used in the intersection
`weight_variable`	
	a character string: the name of the variable in the dataset containing the weights, defaults to NULL
`nintersects`	number of intersections to look at, the default being 12
`exclude_unique`	whether the set intersections should include singular sets (i.e. that one variable). Note that if this is set to True, the total set size on the left will be wrong
`label`	the label to be added to the plot

Value

An UpSetR plot object with the different sets

---

set_intersection_plot *Create a plot from the percentages in each set*

Description

Create a plot from the percentages in each set

Usage

```
set_intersection_plot(set_percentages, nintersects = 12, nsets = NULL,
  label = NULL)
```

Arguments

`set_percentages`	
	a names vector with the percentages for each combination
`nintersects`	number of intersections to look at, the default being 12
`nsets`	number of unique sets making up the intersection
`label`	the label to be added to the plot

Value

A plot object

---

svymean_intersected_sets

Create a vector containing the weighted percentages in each set

---

Description

Create a vector containing the weighted percentages in each set

Usage

```
svymean_intersected_sets(data, intersected_names, weight_variable = NULL,
  weighting_function = NULL)
```

Arguments

data   a dataframe containing all the intersected sets

intersected_names

the names of the intersected sets: as combinations of variable names combined with '&'

weight_variable

character string: the name of the variable in the dataset containing the weights

Value

A plot of the intersection

Index