Package 'describedata'

October 13, 2022

Title Miscellaneous Descriptive Functions

Version 0.1.0

Description Helper functions for descriptive tasks such as making print-friendly bivariate tables, sample size flow counts, and visualizing sample distributions. Also contains 'R' approximations of some common 'SAS' and 'Stata' functions such as 'PROC MEANS' from 'SAS' and 'ladder', 'gladder', and 'pwcorr' from 'Stata'.

Imports dplyr (>= 0.7), forcats, tibble, tidyr, purrr, broom, stringr, haven, ggplot2, lmtest, rlang

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests testthat

NeedsCompilation no

Author Craig McGowan [aut, cre]

Maintainer Craig McGowan <mcgowan.cj@gmail.com>

Repository CRAN

Date/Publication 2019-08-02 11:50:02 UTC

R topics documented:

`bivariate_compare` *Create publication-style table across one categorical variable*

Description

Descriptive statistics for categorical variables as well as normally and non-normally distributed continuous variables, split across levels of a categorical variable. Depending on the variable type, an appropriate statistical test is used to assess differences across levels of the comparison variable.

Usage

```
bivariate_compare(df, compare, normal_vars = NULL,
  non_normal_vars = NULL, cat_vars = NULL, display_round = 2,
  p = TRUE, p_round = 4, include_na = FALSE, col_n = TRUE,
  cont_n = FALSE, all_cont_mean = FALSE, all_cont_median = FALSE,
  iqr = TRUE, fisher = FALSE, workspace = NULL, var_order = NULL,
  var_label_df = NULL)
```

Arguments

`df`	A data.frame or tibble.
`compare`	Discrete variable. Separate statistics will be produced for each level, with statistical tests across levels. Must be quoted.
`normal_vars`	Character vector of normally distributed continuous variables that will be included in the descriptive table.
`non_normal_vars`	Character vector of non-normally distributed continuous variables that will be included in the descriptive table.
`cat_vars`	Character vector of categorical variables that will be included in the descriptive table.
`display_round`	Number of decimal places displayed values should be rounded to
`p`	Logical. Should p-values be calculated and displayed? Default `TRUE`.
`p_round`	Number of decimal places p-values should be rounded to.
`include_na`	Logical. Should `NA` values be included in the table and accompanying statistical tests? Default `FALSE`.
`col_n`	Logical. Should the total number of observations be displayed for each column? Default `TRUE`.
`cont_n`	Logical. Display sample n for continuous variables in the table. Default `FALSE`.
`all_cont_mean`	Logical. Display mean (sd) for all continuous variables. Default `FALSE` results in mean (sd) for normally distributed variables and median (IQR) for non-normally distributed variables. Must be `FALSE` if `all_cont_median == TRUE`.

`all_cont_median`
: Logical. Display median (sd) for all continuous variables. Default `FALSE` results in mean (sd) for normally distributed variables and median (IQR) for non-normally distributed variables. Must be `FALSE` if `all_cont_mean == TRUE`.

`iqr`
: Logical. If the median is displayed for a continuous variable, should interquartile range be displayed as well (`TRUE`), or should the values for the 25th and 75th percentiles be displayed (`FALSE`)? Default `TRUE`

`fisher`
: Logical. Should Fisher's exact test be used for categorical variables? Default `FALSE`. Ignored if `p == FALSE`.

`workspace`
: Numeric variable indicating the workspace to be used for Fisher's exact test. If `NULL`, the default, the default value of `2e5` is used. Ignored if `fisher == FALSE`.

`var_order`
: Character vector listing the variable names in the order results should be displayed. If `NULL`, the default, continuous variables are displayed first, followed by categorical variables.

`var_label_df`
: A data.frame or tibble with columns "variable" and "label" that contains display labels for each variable specified in `normal_vars`, `non_normal_vars`, and `cat_vars`.

Details

Statistical differences between normally distributed continuous variables are assessed using `aov()`, differences in non-normally distributed variables are assessed using `kruskal.test()`, and differences in categorical variables are assessed using `chisq.test()` by default, with a user option for `fisher.test()` instead.

Value

A data.frame with columns label, overall, a column for each level of `compare`, and p.value. For `normal_vars`, mean (SD) is displayed, for `non_normal_vars` median (IQR) is displayed, and for `cat_vars` n (percent) is displayed. For p values on continuous variables, a superscript 'a' denotes the Kruskal-Wallis test was used

Examples

```
bivariate_compare(iris, compare = "Species", normal_vars = c("Sepal.Length", "Sepal.Width"))

bivariate_compare(mtcars, compare = "cyl", non_normal_vars = "mpg")
```

`cor.prob` *Calculate pairwise correlations*

Description

Internal function to calculate pairwise correlations and return p values

Usage

```
cor.prob(df)
```

Arguments

df	A data frame or tibble.

Value

A data.frame with columns h_var, v_var, and p.value

`describedata` *describedata: Miscellaneous descriptive and SAS/Stata duplicate functions*

Description

The helpR package contains descriptive functions for tasks such as making print-friendly bivariate tables, sample size flow counts, and more. It also contains R approximations of some common, useful SAS/Stata functions.

Frequency functions

The helper functions `bivariate_compare` and `univar_freq` create frequency tables. `univar_freq` produces simple n and percent for categories of a single variable, while `bivariate_compare` compares continuous or categorical variables across categories of a comparison variable. This is particularly useful for generating a Table 1 or 2 for a publication manuscript.

Sample size functions

`sample_flow` produces tables illustrating how final sample size is determined and the number of participants excluded by each exclusion criteria.

Other helper functions

`nagelkerke` calculates the Nagelkerke pseudo r-squared for a logistic regression model.

Stata replica functions

`ladder`, `gladder`, and `pwcorr` are approximate replicas of the respective Stata functions. Not all functionality is currently incorporated. `stata_tidy` reformats R model output to a format similar to Stata.

SAS replica functions

`proc_means` is an approximate replica of the respective SAS function. Not all functionality is currently incorporated.

gladder *Replica of Stata's gladder function*

Description

Creates ladder-of-powers histograms to visualize nine common transformations and compare each to a normal distribution. The following transformations are included: identity, cubic, square, square root, natural logarithm, inverse square root, inverse, inverse square, and inverse cubic.

Usage

```
gladder(x)
```

Arguments

x	A continuous numeric vector.

Value

A ggplot object with plots of each transformation

Examples

```
gladder(iris$Sepal.Length)
gladder(mtcars$disp)
```

ladder *Replica of Stata's ladder function*

Description

Searches the ladder of powers histograms to find a transformation to make x normally distributed. The Shapiro-Wilkes test is used to assess for normality. The following transformations are included: identity, cubic, square, square root, natural logarithm, inverse square root, inverse, inverse square, and inverse cubic.

Usage

```
ladder(x)
```

Arguments

x	A continuous numeric vector.

Value

A data.frame

Examples

```
ladder(iris$Sepal.Length)
ladder(mtcars$disp)
```

`nagelkerke` *Calculate Nagelkerke pseudo r-squared*

Description

Calculate Nagelkerke pseudo r-squared from a fitted model object.

Usage

```
nagelkerke(mod)
```

Arguments

`mod`	A `glm` model object, usually from logistic regression. The model must have been fit using the `data` option, in order to extract the data from the model object.

Value

Numeric value of Nagelkerke r-squared for the model

`norm_dist_plot` *Create density histogram with normal distribution overlaid*

Description

Plots a simple density histogram for a continuous variable with a normal distribution overlaid. The overlaid normal distribution has the same mean and standard deviation as the provided variable, and the plot provides a visual means to assess the normality of the variable's distribution.

Usage

```
norm_dist_plot(df, vars)
```

Arguments

`df`	A data.frame or tibble.
`vars`	A character vector of continuous variable names.

Value

A ggplot object.

Examples

```
norm_dist_plot(df = iris, vars = "Sepal.Width")

norm_dist_plot(df = iris,
               vars = c("Sepal.Width", "Sepal.Length"))
```

---

proc_means *Replica of SAS's PROC MEANS*

Description

Descriptive statistics for continuous variables, with the option of stratifying by a categorical variable.

Usage

```
proc_means(df, vars = NULL, var_order = NULL, by = NULL, n = T,
  mean = TRUE, sd = TRUE, min = TRUE, max = TRUE, median = FALSE,
  q1 = FALSE, q3 = FALSE, iqr = FALSE, nmiss = FALSE,
  nobs = FALSE, p = FALSE, p_round = 4, display_round = 3)
```

Arguments

df	A data frame or tibble.
vars	Character vector of numeric variables to generate descriptive statistics for. If the default (NULL), all variables are included, except for any specified in by.
var_order	Character vector listing the variable names in the order results should be displayed. If the default (NULL), variables are displayed in the order specified in vars.
by	Discrete variable. Separate statistics will be produced for each level. Default NULL provides statistics for all observations.
n	logical. Display number of rows with values. Default TRUE.
mean	logical. Display mean value. Default TRUE.
sd	logical. Display standard deviation. Default TRUE.
min	logical. Display minimum value. Default TRUE.
max	logical. Display maximum value. Default TRUE.
median	logical. Display median value. Default FALSE.
q1	logical. Display first quartile value. Default FALSE.
q3	logical. Display third quartile value. Default FALSE.

iqr	logical. Display interquartile range. Default FALSE.
nmiss	logical. Display number of missing values. Default FALSE.
nobs	logical. Display total number of rows. Default FALSE.
p	logical. Calculate p-value across by groups using aov. Ignored if no by variable specified. Default FALSE.
p_round	Number of decimal places p-values should be rounded to.
display_round	Number of decimal places displayed values should be rounded to

Value

A data.frame with columns variable, by variable, and a column for each summary statistic.

Examples

```
proc_means(iris, vars = c("Sepal.Length", "Sepal.Width"))
proc_means(iris, by = "Species")
```

---

pwcorr *Replica of Stata's pwcorr function*

Description

Calculate and return a matrix of pairwise correlation coefficients. Returns significance levels if method == "pearson"

Usage

```
pwcorr(df, vars = NULL, method = "pearson", var_label_df = NULL)
```

Arguments

df	A data.frame or tibble.
vars	A character vector of numeric variables to generate pairwise correlations for. If the default (NULL), all variables are included.
method	One of "pearson", "kendall", or "spearman" passed on to "cor".
var_label_df	A data.frame or tibble with columns "variable" and "label" that contains display labels for each variable specified in vars.

Value

A data.frame displaying the pairwise correlation coefficients between all variables in vars.

`sample_flow` *Create table illustrating sample exclusions*

Description

Generate a table illustrating sequential exclusion from an analytical sample due to user specified exclusions.

Usage

```
sample_flow(df, exclusions = c())
```

Arguments

`df`	A data.frame or tibble.
`exclusions`	Character vector of logical conditions indicating which rows should be excluded from the final sample. Exclusions occur in the order specified.

Value

A data.frame with columns Exclusion, 'Sequential Excluded', and 'Total Excluded' for display.

`stata_tidy` *Tidy model output into similar format from Stata*

Description

Create a display data frame similar to Stata model output for a fitted R model.

Usage

```
stata_tidy(mod, var_label_df = NULL)
```

Arguments

`mod`	A fitted model object
`var_label_df`	A data.frame or tibble with columns "variable" and "label" that contains display labels for each variable in `mod`.

Value

A data.frame with columns term and display

univar_freq *Univariate statistics for a discrete variable*

Description

Descriptive statistics (N,

Usage

```
univar_freq(df, var, na.rm = FALSE)
```

Arguments

df	A data frame or tibble.
var	A discrete, numeric variable.
na.rm	logical. Should missing values (including NaN) be removed?)

Value

A data.frame with columns var, NObs, and Percent

Examples

```
univar_freq(iris, var = "Species")
univar_freq(mtcars, var = "cyl")
```

Index