

Package ‘rfriend’

July 23, 2025

Type Package

Title Provides Batch Functions and Visualisation for Basic Statistical Procedures

Version 1.0.0

Description Designed to streamline data analysis and statistical testing, reducing the length of R scripts while generating well-formatted outputs in 'pdf', 'Microsoft Word', and 'Microsoft Excel' formats. In essence, the package contains functions which are sophisticated wrappers around existing R functions that are called by using 'f_' (user f_riently) prefix followed by the normal function name. This first version of the 'rfriend' package focuses primarily on data exploration, including tools for creating summary tables, f_summary(), performing data transformations, f_boxcox() in part based on 'MASS/boxcox' and 'rcompanion', and f_bestNormalize() which wraps and extends functionality from the 'bestNormalize' package. Furthermore, 'rfriend' can automatically (or on request) generate visualizations such as boxplots, f_boxplot(), QQ-plots, f_qqnorm(), histograms f_hist(), and density plots. Additionally, the package includes four statistical test functions: f_aov(), f_kruskal_test(), f_glm(), f_chisq_test for sequential testing and visualisation of the 'stats' functions: aov(), kruskal.test(), glm() and chisq.test. These functions support testing multiple response variables and predictors, while also handling assumption checks, data transformations, and post hoc tests. Post hoc results are automatically summarized in a table using the compact letter display (cld) format for easy interpretation. The package also provides a function to do model comparison, f_model_comparison(), and several utility functions to simplify common R tasks. For example, f_clear() clears the workspace and restarts R with a single command; f_setwd() sets the working directory to match the directory of the current script; f_theme() quickly changes 'RStudio' themes; and f_factors() converts multiple columns of a data frame to factors, and much more. If you encounter any issues or have feature requests, please feel free to contact me via email.

Note When loading, both MuMIn and rstatix are imported. Since rstatix internally depends on broom, this may trigger a warning about S3 method overwrites, specifically for nobis.fidistr and nobis.multinom. These warnings are harmless and do not affect functionality.

License GPL-3

Encoding UTF-8

Depends R (>= 4.4.0)

Imports bestNormalize, crayon, DHARMA, emmeans, ggplot2, grDevices, knitr, magick, multcomp, multcompView, MuMIn, nortest, pander, rmarkdown, rstatix, rstudioapi, stringr, this.path, writexl, xfun

RoxygenNote 7.3.2

SystemRequirements Pandoc (>= 3.2)

NeedsCompilation no

Author Sander H. van Delden [aut, cre]

Maintainer Sander H. van Delden <plantmind@proton.me>

Repository CRAN

Date/Publication 2025-07-16 15:40:02 UTC

Contents

f_aov	3
f_bestNormalize	6
f_boxcox	8
f_boxplot	12
f_chisq_test	15
f_clear	17
f_conditional_round	18
f_corplot	20
f_factors	23
f_glm	25
f_hist	29
f_kruskal_test	31
f_load_packages	34
f_model_comparison	34
f_open_file	37
f_pander	38
f_qqnorm	39
f_rename_columns	41
f_rename_vector	43
f_setwd	44
f_summary	45
f_theme	48
plot.f_bestNormalize	49
plot.f_boxcox	50
print.f_summary	51

Index	52
--------------	-----------

f_aov	<i>Perform multiple aov() functions with optional data transformation, inspection and Post Hoc test.</i>
-------	--

Description

Performs an Analysis of Variance (ANOVA) on a given dataset with options for (Box-Cox) transformations, normality tests, and post-hoc analysis. Several response parameters can be analysed in sequence and the generated output can be in various formats ('Word', 'pdf', 'Excel').

Usage

```
f_aov(
  formula,
  data = NULL,
  norm_plots = TRUE,
  ANCOVA = FALSE,
  transformation = TRUE,
  alpha = 0.05,
  adjust = "sidak",
  aov_assumptions_text = TRUE,
  close_generated_files = FALSE,
  open_generated_files = TRUE,
  output_type = "off",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE
)
```

Arguments

formula	A formula specifying the model to be fitted. More response variables can be added using - or + (e.g., response1 + response2 ~ predictor) to do a sequential aov() for each response parameter.
data	A data frame containing the variables in the model.
norm_plots	Logical. If TRUE, plots are included in the output files. Default is TRUE.
ANCOVA	Logical. If TRUE, prevents automatic conversion of predictors to factors, allowing for Analysis of Covariance (ANCOVA). Default is FALSE.
transformation	Logical or character string. If TRUE, or if "bestnormalize", applies bestNormalize() transformation if residuals are not normal. If "boxcox" applies a boxcox transformation. If FALSE no transformation will be applied. Default is TRUE.
alpha	Numeric. Significance level for ANOVA, post hoc tests, and Shapiro-Wilk test. Default is 0.05.
adjust	Character string specifying the method used to adjust p-values for multiple comparisons. Available methods include:

"tukey" Tukey's Honest Significant Difference method, appropriate for all pair-wise comparisons. Controls family-wise error rate.

"sidak" Šidák correction that controls the family-wise error rate. Less conservative than Bonferroni.

"bonferroni" Conservative adjustment that multiplies p-values by the number of comparisons.

"none" No adjustment. Equivalent to Fisher's LSD method.

"fdr" False Discovery Rate adjustment, controls the expected proportion of false positives among significant results.

Default is "sidak".

aov_assumptions_text

Logical. If TRUE, includes a short explanation about ANOVA assumptions in the output file. Default is TRUE.

close_generated_files

Logical. If TRUE, closes open 'Excel' or 'Word' files depending on the output format. This to be able to save the newly generated file by the f_aov() function. 'Pdf' files should also be closed before using the function and cannot be automatically closed. Default is FALSE.

open_generated_files

Logical. If TRUE, Opens the generated output files ('pdf', 'Word' or 'Excel') files depending on the output format. This to directly view the results after creation. Files are stored in tempdir(). Default is TRUE.

output_type

Character string specifying the output format: "pdf", "word", "excel", "rmd", "console" or "off" (no file generated). The option "console" forces output to be printed. Default is "off".

output_file

Character string specifying the name of the output file. Default is "dataname_aov_output".

output_dir

Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.

save_in_wdir

Logical. If TRUE, saves the file in the working directory Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.

Details

The function performs the following steps:

- Check if all specified variables are present in the data.
- Ensure that the response variable is numeric.
- Perform Analysis of Variance (ANOVA) using the specified formula and data.
- If shapiro = TRUE, check for normality of residuals using the Shapiro-Wilk test.
- If residuals are not normal and transformation = TRUE apply a data transformation.
- If significant differences are found in ANOVA, proceed with post hoc tests using estimated marginal means from emmeans() and Sidak adjustment (or another option of adjust =).

More response variables can be added using - or + (e.g., `response1 + response2 ~ predictor`) to do a sequential `aov()` for each response parameter captured in one output file.

Outputs can be generated in multiple formats ("pdf", "word", "excel" and "rmd") as specified by `output_type`. The function also closes any open 'Word' files to avoid conflicts when generating 'Word' documents. If `output_type = "rmd"` is used it is advised to use it in a chunk with `{r, echo=FALSE, results='asis'}`

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder. (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "/usr/local/bin". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "/usr/bin" or "/usr/local/bin") or manually, and ensure the directory containing Pandoc is in your PATH.
- If Pandoc is not found, this function may not work as intended.

Value

An object of class 'f_aov' containing results from `aov()`, normality tests, transformations, and post hoc tests. Using the option "output_type", it can also generate output in the form of: R Markdown code, 'Word', 'pdf', or 'Excel' files. Includes print and plot methods for 'f_aov' objects.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Make a factor of Species.
iris$Species <- factor(iris$Species)

# The left hand side contains two response variables,
# so two aov's will be conducted, i.e. "Sepal.Width"
# and "Sepal.Length" in response to the explanatory variable: "Species".
f_aov_out <- f_aov(Sepal.Width + Sepal.Length ~ Species,
  data = iris,
  # Save output in MS Word file (Default is console)
  output_type = "word",
  # Do boxcox transformation for non-normal residual (Default is bestnormalize)
  transformation = "boxcox",
  # Do not automatically open the file.
  open_generated_files = FALSE
)

# Print output to the console.
print(f_aov_out)
```

```
# Plot residual plots.
plot(f_aov_out)

#To print rmd output set chunk option to results = 'asis' and use cat().
f_aov_rmd_out <- f_aov(Sepal.Width ~ Species, data = iris, output_type = "rmd")
cat(f_aov_rmd_out$rmd)
```

f_bestNormalize	<i>f_bestNormalize: Automated Data Normalization with bestNormalize</i>
-----------------	---

Description

Applies optimal normalization transformations using 'bestNormalize', provides diagnostic checks, and generates comprehensive reports.

Usage

```
f_bestNormalize(
  data,
  alpha = 0.05,
  plots = FALSE,
  data_name = NULL,
  output_type = "off",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  close_generated_files = FALSE,
  open_generated_files = TRUE,
  ...
)
```

Arguments

data	Numeric vector or single-column data frame.
alpha	Numeric. Significance level for normality tests (default = 0.05).
plots	Logical. If TRUE, plots Q-Q plots and Histograms of the original and transformed data. Default is FALSE.
data_name	A character string to manually set the name of the data for plot axis and reporting. Default extracts name from input object. data.
output_type	Character. Output format: "console", "pdf", "word", "rmd", or "off". The option "console" forces output to be printed. Default is "off".
output_file	Character. Custom output filename (optional).
output_dir	Character. Output directory (default = tempdir()).
save_in_wdir	Logical. Save in working directory (default = FALSE).

close_generated_files
 Logical. If TRUE, closes open 'Word' files. This to be able to save the newly generated file by the f_bestNormalize() function. 'Pdf' files should also be closed before using the function and cannot be automatically closed. Default is FALSE.

open_generated_files
 Logical. If TRUE, Opens the generated output file, this to directly view the results after creation. Files are stored in tempdir(). Default is TRUE.

...
 Additional arguments passed to bestNormalize.

Details

This is a wrapper around the 'bestNormalize' package. Providing a fancy output and the settings of 'bestNormalize' are tuned based on sample size n . If $n < 100$, `loo = TRUE`, `allow_orderNorm = FALSE` and `r` doesn't matter as `loo = TRUE`. If $100 \leq n < 200$, `loo = FALSE`, `allow_orderNorm = TRUE` and `r = 50`. If $n \geq 200$, `loo = FALSE`, `allow_orderNorm = TRUE`, `r = 10`. These setting can be overwritten by user options.

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "/usr/local/bin". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "/usr/bin" or "/usr/local/bin") or manually, and ensure the directory containing Pandoc is in your PATH.
- If Pandoc is not found, this function may not work as intended.

Value

Returns an object of class 'f_bestNormalize' containing:

- transformed_data Normalized vector.
- bestNormalize Full bestNormalize object from original package.
- data_name Name of the analyzed dataset.
- transformation_name Name of selected transformation.
- shapiro_original Shapiro-Wilk test results for original data.
- shapiro_transformed Shapiro-Wilk test results for transformed data.
- norm_stats Data frame of normality statistics for all methods.
- rmd Rmd code if outputtype = "rmd".

Also generates reports in specified formats, when using output to console and `plots = TRUE`, the function prints QQ-plots, Histograms and a summary data transformation report.

#' @return An object of class 'f_bestNormalize' containing results from "bestNormalize", the input data, transformed data, Shapiro-Wilk test on original and transformed data. Using the option "output_type", it can also generate output in the form of: R Markdown code, 'Word', or 'pdf' files. Includes print and plot methods for objects of class 'f_bestNormalize'.

Author(s)

Sander H. van Delden <plantmind@proton.me>

References

Peterson, C. (2025). **bestNormalize**: Flexibly calculate the best normalizing transformation for a vector. Available at: <https://cran.r-project.org/package=bestNormalize>

Examples

```
# Create some skewed data (e.g., using a log-normal distribution).
skewed_data <- rlnorm(100, meanlog = 0, sdlog = 1)

# Use set.seed to keep the outcome of bestNormalize stable.
set.seed(123)

# Transform the data and store all information in f_bestNormalize_out.
f_bestNormalize_out <- f_bestNormalize(skewed_data)

# Print the output.
print(f_bestNormalize_out)

# Show histograms and QQplots.
plot(f_bestNormalize_out)

# Directly store the transformed_data from f_bestNormalize and force to show
# plots and transformation information.
transformed_data <- f_bestNormalize(skewed_data, output_type = "console")$transformed_data

# Any other transformation can be choosen by using:
boxcox_transformed_data <- f_bestNormalize(skewed_data)$bestNormalize$other_transforms$boxcox$x.t
# and substituting '$boxcox' with the transformation of choice.

#To print rmd output set chunk option to results = 'asis' and use:
f_bestNormalize_rmd_out <- f_bestNormalize(skewed_data, output_type = "rmd")
cat(f_bestNormalize_rmd_out$rmd)
```

f_boxcox

f_boxcox: A User-Friendly Box-Cox Transformation

Description

Performs a Box-Cox transformation on a dataset to stabilize variance and make the data more normally distributed. It also provides diagnostic plots and tests for normality. The transformation is based on code of MASS/R/boxcox.R. The function prints λ to the console and returns (output) the transformed data set.

Usage

```
f_boxcox(
  data = data,
  lambda = seq(-2, 2, 1/10),
  plots = FALSE,
  transform.data = TRUE,
  interp = (plots && (length(lambda) < 100)),
  eps = 1/50,
  xlab = expression(lambda),
  ylab = "log-Likelihood",
  alpha = 0.05,
  open_generated_files = TRUE,
  close_generated_files = FALSE,
  output_type = "off",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  ...
)
```

Arguments

data	A numeric vector or a data frame with a single numeric column. The data to be transformed.
lambda	A numeric vector of λ values to evaluate for the Box-Cox transformation. Default is <code>seq(-2, 2, 0.1)</code> .
plots	Logical. If TRUE, plots log-likelihood of the Box-Cox transformation, Histograms and Q-Q plots of the original and transformed data. Default is FALSE.
transform.data	Logical. If TRUE, returns the transformed data. Default is TRUE.
interp	Logical. If TRUE and fewer than 100 λ values are provided, interpolates for smooth plotting. Default is determined by log-likelihood of the Box-Cox transformation and the length of λ .
eps	A small positive value used to determine when to switch from the power transformation to the log transformation for numerical stability. Default is 1/50.
xlab	Character string. Label for the x-axis in plots. Default is an expression object representing λ .
ylab	Character string. Label for the y-axis in plots. Default is "log-Likelihood".
alpha	Numeric. Significance level for the Shapiro-Wilk test of normality. Default is 0.05.
open_generated_files	Logical. If TRUE, opens the generated output files ('pdf', 'Word' or 'Excel') files depending on the output format. This to directly view the results after creation. Files are stored in <code>tempdir()</code> . Default is TRUE.
close_generated_files	Logical. If TRUE, closes open 'Word' files depending on the output format. This to be able to save the newly generated files. 'Pdf' files should also be closed before using the function and cannot be automatically closed.

output_type	Character string specifying the output format: "pdf", "word", "rmd", "off" (no file generated) or "console". The option "console" forces output to be printed. Default is "off".
output_file	A character string specifying the name of the output file (without extension). If NULL, a default name based on the dataset name is generated.
output_dir	Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.
save_in_wdir	Logical. If TRUE, saves the file in the working directory Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.
...	Additional arguments passed to plotting functions.

Details

The function uses the following formula for transformation:

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log(y), & \lambda = 0 \end{cases}$$

where (y) is the data being transformed, and (λ) the transformation parameter, which is estimated from the data using maximum likelihood. The function computes the Box-Cox transformation for a range of λ values and identifies the λ that maximizes the log-likelihood function. The beauty of this transformation is that, it checks suitability of many of the common transformations in one run. Examples of most common transformations and their λ value is given below:

λ -Value	Transformation
-2	$\frac{1}{x^2}$
-1	$\frac{1}{x}$
-0.5	$\frac{1}{\sqrt{x}}$
0	$\log(x)$
0.5	\sqrt{x}
1	x
2	x^2

If the estimated transformation parameter closely aligns with one of the values listed in the previous table, it is generally advisable to select the table value rather than the precise estimated value. This approach simplifies interpretation and practical application.

The function provides diagnostic plots: a plot of log-likelihood against λ values and a Q-Q plot of the transformed data. It also performs a Shapiro-Wilk test for normality on the transformed data if the sample size is less than or equal to 5000.

Note: For sample sizes greater than 5000, Shapiro-Wilk test results are not provided due to limitations in its applicability.

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "/usr/local/bin". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "/usr/bin" or "/usr/local/bin") or manually, and ensure the directory containing Pandoc is in your PATH.
- If Pandoc is not found, this function may not work as intended.

Value

An object of class 'f_boxcox' containing, among others, results from the boxcox transformation, lambda, the input data, transformed data, Shapiro-Wilk test on original and transformed data. Using the option "output_type", it can also generate output in the form of: R Markdown code, 'Word', or 'pdf' files. Includes print and plot methods for 'f_boxcox' objects.

Author(s)

- Sander H. van Delden <plantmind@proton.me>
- Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>
- W. N. Venables and B. D. Ripley

References

The core of calculating λ and the plotting was taken from:
file MASS/R/boxcox.R copyright (C) 1994-2004 W. N. Venables and B. D. Ripley

- <https://r-coder.com/box-cox-transformation-r/>
- <https://CRAN.R-project.org/package=MASS>

Some code to present the result was taken and modified from file:
rcompanion/R/transformTukey.r. (Developed by Salvatore Mangiafico)

- https://rcompanion.org/handbook/I_12.html

The explanation on BoxCox transformation provided here was provided by r-coder:

- <https://r-coder.com/box-cox-transformation-r/>

See Also

[boxcox](#)

Examples

```
# Create non-normal data in a data.frame or vector.
df <- data.frame(values = rlnorm(100, meanlog = 0, sdlog = 1))

# Store the transformation in object "bc".
bc <- f_boxcox(df$values)

# Print lambda and Shaprio.
print(bc)

# Plot the QQ plots, Histograms and Lambda Log-Likelihood estimation.
plot(bc)

# Or Directly use the transformed data from the f_boxcox object.
df$values_transformed <- f_boxcox(df$values)$transformed_data
print(df$values_transformed)
```

f_boxplot

*Generate a Boxplot Report of a data.frame***Description**

Generates boxplots for all numeric variables in a given dataset, grouped by factor variables. The function automatically detects numeric and factor variables. It allows two output formats ('pdf', 'Word') and includes an option to add a general explanation about interpreting boxplots.

Usage

```
f_boxplot(
  data = NULL,
  formula = NULL,
  fancy_names = NULL,
  output_type = "pdf",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  close_generated_files = FALSE,
  open_generated_files = TRUE,
  boxplot_explanation = TRUE,
  detect_factors = TRUE,
  jitter = FALSE,
  width = 8,
  height = 7,
  units = "in",
  res = 300,
  las = 2
)
```

Arguments

data	A data.frame containing the data to be used for creating boxplots.
formula	A formula specifying the factor to be plotted. More response variables can be added using - or + (e.g., response1 + response2 ~ predictor) to generate multiple boxplots. If the formula is omitted and only data is provided all data will be used for creating boxplots.
fancy_names	An optional named vector mapping column names in data to more readable names for display in plots (name map). Defaults to NULL.
output_type	Character string, specifying the output format: "pdf", "word", "rmd" or "png". Default is "pdf".
output_file	A character string, specifying the name of the output file (without extension). If NULL, a default name based on the dataset is generated.
output_dir	Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.
save_in_wdir	Logical. If TRUE, saves the file in the working directory Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.
close_generated_files	Logical. If TRUE, closes open 'Word' files depending on the output format. This to be able to save the newly generated files. 'Pdf' files should also be closed before using the function and cannot be automatically closed.
open_generated_files	Logical. If TRUE, Opens the generated output files ('pdf', 'Word' or 'png') files depending on the output format. This to directly view the results after creation. Files are stored in tempdir(). Default is TRUE.
boxplot_explanation	A logical value indicating whether to include an explanation of how to interpret boxplots in the report. Defaults to TRUE.
detect_factors	A logical value indicating whether to automatically detect factor variables in the dataset. Defaults to TRUE.
jitter	A logical value, if TRUE all data per boxplot is shown, if FALSE (default) individual data points (except for outliers) are omitted.
width	Numeric, png figure width default 8 inch
height	Numeric, png figure height default 7 inch
units	Character string, png figure units default "in" = inch, other options are: "px" = Pixels, "cm" = centimeters, "mm" = millimeters.
res	Numeric, png figure resolution default 300 dpi
las	An integer (0 t/m 3), las = 0: Axis labels are parallel to the axis. las = 1: Axis labels are always horizontal. las = 2: Axis labels are perpendicular to the axis. (default setting). las = 3: Axis labels are always vertical.

Details

The function performs the following steps:

- Detects numeric and factor variables in the dataset.
- Generates boxplots for each numeric variable grouped by each factor variable.
- Outputs the report in the specified format ('pdf', 'Word' or 'Rmd').

If `output_type = "rmd"` is used it is advised to use it in a chunk with `{r, echo=FALSE, results='asis'}`

If no factor variables are detected, the function stops with an error message since factors are required for creating boxplots.

This function will plot all numeric and factor candidates, use the function `subset()` to prepare a selection of columns before submitting to `f_boxplot()`.

Note that there is an optional `jitter` option to plot all individual data points over the boxplots.

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

Windows: Install Pandoc and ensure the installation folder (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.

macOS: If using Homebrew, Pandoc is typically installed in `/usr/local/bin`. Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.

Linux: Install Pandoc through your distribution's package manager (commonly installed in `/usr/bin` or `/usr/local/bin`) or manually, and ensure the directory containing Pandoc is in your PATH.

If Pandoc is not found, this function may not work as intended.

Value

Generates a report file ('pdf' or 'Word') with boxplots and, optionally, opens it with the default program. Returns NULL (no R object) when generating 'pdf' or 'Word' files. Can also return R Markdown code or 'PNG' files depending on the output format.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Example usage:
data(iris)

new_names = c(
  "Sepal.Length" = "Sepal length (cm)" ,
  "Sepal.Width" = "Sepal width (cm)",
  "Petal.Length" = "Petal length (cm)",
  "Petal.Width" = "Petal width (cm)",
  "Species" = "Cultivar"
)
```

```
# Use the whole data.frame to generate a pdf report and don't open the pdf.
f_boxplot(iris, fancy_names = new_names, output_type = "pdf", open_generated_files = FALSE) #

# Use a formula to plot several response parameters (response 1 + response 2 etc)
# and generate a rmd output without boxplot_explanation.
data(mtcars)
f_boxplot(hp + disp ~ gear*cyl,
          data=mtcars,
          boxplot_explanation = FALSE,
          output_type = "word",
          open_generated_files = FALSE) # Do not automatically open the 'Word' file.
```

f_chisq_test	<i>Chi-squared Test with Post-hoc Analysis</i>
--------------	--

Description

Performs a chi-squared test `chisq.test`, then automatically conducts post-hoc analysis if the test is significant. The function provides adjusted p-values for each cell in the contingency table using a specified correction method.

Usage

```
f_chisq_test(
  x,
  y,
  p = NULL,
  method = "bonferroni",
  digits = 3,
  alpha = 0.05,
  force_posthoc = FALSE,
  ...
)
```

Arguments

x	A numeric vector (or factor), or a contingency table in matrix or table form. If a data frame is entered the function will try to convert it to a table.
y	A numeric vector; ignored if x is a matrix, table or data.frame. If x is a factor, y should be a factor of the same length.
p	A vector of probabilities of the same length as x. Default is NULL. An error is given if any entry of p is negative.
method	Character string specifying the adjustment method for p-values. Default is "bonferroni". Other options include "holm", "hochberg", "hommel", "BH", "BY", "fdr", and "none".

digits	Integer specifying the number of decimal places for rounding. Default is 3.
alpha	Numeric threshold for significance. Default is 0.05.
force_posthoc	Logical indicating whether to perform post-hoc tests even if the chi-squared test is not significant. Default is FALSE.
...	Additional arguments passed to <code>chisq.test</code> .

Details

The function first performs a chi-squared test using `chisq.test`. If the test is significant ($p < \alpha$) or if `force_posthoc = TRUE`, it conducts post-hoc analysis by examining the standardized residuals. The p-values for these residuals are adjusted using the specified method to control for multiple comparisons.

If the input is a data frame, the function attempts to convert it to a table and displays the resulting table for verification.

Value

An object of class `f_chisq_test` containing:

- `chisq_test_output`: The output from `chisq.test`.
- `adjusted_p_values`: Matrix of adjusted p-values (for table/matrix input).
- `observed_vs_adj_p_value`: Interleaved table of observed values and adjusted p-values.
- `stdres_vs_adj_p_value`: Interleaved table of standardized residuals and adjusted p-values.
- `adj_p_values`: Vector of adjusted p-values (for vector input).
- `posthoc_output_table`: Data frame with observed values, expected values, standardized residuals, and adjusted p-values (for vector input).

Author(s)

Sander H. van Delden <plantmind@proton.me>

References

This function implements a post-hoc analysis for chi-squared tests inspired by the methodology in: Beasley, T. M., & Schumacker, R. E. (1995). Multiple Regression Approach to Analyzing Contingency Tables: Post Hoc and Planned Comparison Procedures. The Journal of Experimental Education, 64(1), 79-93.

The implementation draws inspiration from the '`chisq.posthoc.test`' package by Daniel Ebbert.

Examples

```
# Chi.square on independence: Association between two variables.
# Create a contingency table.
my_table <- as.table(rbind(c(100, 150, 50), c(120, 90, 40)))
dimnames(my_table) <- list(Gender = c("Male", "Female"),
                           Response = c("Agree", "Neutral", "Disagree"))
```



```

# Perform chi-squared test with post-hoc analysis.
f_chisq_test(my_table)

# Use a different adjustment method.
f_chisq_test(my_table, method = "holm")

# Other forms still work like Goodness-of-Fit: Match to theoretical distribution.
# Observed frequencies of rolling with a die 1 - 6.
observed <- c(2, 2, 10, 20, 15, 11)

# Expected probabilities under a fair die.
expected_probs <- rep(1/6, 6)

# Chi-Square Goodness-of-Fit Test.
f_chisq_test(x = observed, p = expected_probs)

```

f_clear

*f_clear: Clear Various Aspects of the R Environment***Description**

Provides a convenient way to clear different components of the R environment, including the console, memory, graphics, and more. It also offers the option to restart the R session. This can come in handy at the start of an R script.

Usage

```
f_clear(env = TRUE, gc = TRUE, console = TRUE, graph = TRUE, restart = FALSE)
```

Arguments

env	Logical. If TRUE, all objects in the global environment are removed. Default is TRUE.
gc	Logical. If TRUE, garbage collection is performed to free up memory. Default is TRUE.
console	Logical. If TRUE, the R console is cleared. Default is TRUE.
graph	Logical. If TRUE, all open graphics devices are closed. Default is TRUE.
restart	Logical. If TRUE, the R session is restarted using 'RStudio's' API. Default is FALSE.

Details

- Console Clearing: Clears the console output.
- Garbage Collection: Performs garbage collection to free memory from unreferenced objects.
- Graph Clearing: Closes all open graphics devices.
- Environment Clearing: Removes all objects from the global environment.
- Session Restart: Restarts the R session (only available in 'RStudio').

Value

No return value, called for side effects, see details.

Note

The restart parameter requires 'RStudio' and its API package ('rstudioapi') to be installed and available.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Clear console, memory, graphs, and for example NOT the environment.
f_clear(env = FALSE)
```

f_conditional_round *Conditional Rounding for Numeric Values*

Description

Conditionally formats numeric values based on their magnitude. Values that are very small or very large are formatted using scientific notation, while other values are rounded to a specified number of decimal places. Integers are preserved without decimal places. When applied to a data frame, only numeric columns are processed. All output is character string.

Usage

```
f_conditional_round(
  x,
  threshold_small = 0.01,
  threshold_large = 9999,
  digits = 3,
  replace_na = TRUE,
  detect_int_col = TRUE
)
```

Arguments

x A numeric vector or data frame containing numeric columns to be formatted.

threshold_small Numeric value. Values with absolute magnitude smaller than this threshold will be formatted using scientific notation. Default is 0.01.

threshold_large	Numeric value. Values with absolute magnitude larger than or equal to this threshold will be formatted using scientific notation. Default is 9999.
digits	Integer. Number of significant digits to use in formatting. Default is 3.
replace_na	Logical. If TRUE, NA values will be replaced with empty strings ("") in the output. Default is TRUE.
detect_int_col	Logical. If TRUE, columns in a data.frame containing only integers will be displayed without decimal digits. Columns containing a mix of integers and decimal values will display all values with the specified number of digits. If FALSE, each individual cell is evaluated: integer values are displayed without digits, and numbers containing digits with the specified number of digits. Default is TRUE.

Details

The function applies the following formatting rules:

- Values smaller than `threshold_small` or larger than `threshold_large` are formatted in scientific notation with `digits` significant digits.
- Integer values are formatted without decimal places.
- Non-integer values that don't require scientific notation are rounded to `digits` decimal places.
- NA values are replaced with empty strings if `replace_na = TRUE`.
- Empty strings in the input are preserved.
- For data frames, only numeric columns are processed; other columns remain unchanged.

Value

- If input is a vector: A character vector of the same length as the input, with values formatted according to the specified rules.
- If input is a data frame: A data frame with the same structure as the input, but with character columns formatted according to the specified rules.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Vector examples.
f_conditional_round(c(0.0001, 0.5, 3, 10000))
# Returns: "1.000e-04" "0.500" "3" "1.000e+04".

f_conditional_round(c(0.0001, 0.5, 3, 10000, NA), replace_na = TRUE)
# Returns: "1.000e-04" "0.500" "3" "1.000e+04" ""

# Data frame example.
df <- data.frame(
  name = c("A", "B", "C"),
```

```

    small_val = c(0.0001, 0.002, 0.5),
    integer = c(1, 2, 3),
    integer_mix = c(10, 20, 30.1),
    large_val = c(10000, 5000, NA)
  )

  # Show only two digits.
  f_conditional_round(df, digits = 2)

  # To keep Integers as Integers (no digits)
  # in columns with mixed data (Integers and digits)
  # set detect_int_col = FALSE
  f_conditional_round(df, detect_int_col = FALSE)

```

f_corplot

Correlation Plots with Factor Detection and Customization

Description

Creates correlation plots for numeric variables in a data frame, optionally incorporating factors for coloring and shaping points. It supports automatic detection of factors, customization of plot aesthetics, and the generation of separate legend files.

Usage

```

f_corplot(
  data,
  detect_factors = TRUE,
  factor_table = FALSE,
  color_factor = "auto",
  shape_factor = "auto",
  print_legend = TRUE,
  fancy_names = NULL,
  width = 15,
  height = 15,
  res = 600,
  pointsize = 8,
  legendname = NULL,
  close_generated_files = FALSE,
  open_generated_files = TRUE,
  output_type = "word",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE
)

```

Arguments

data	A data.frame containing the dataset to be visualized. Must include at least two numeric variables.
detect_factors	Logical. If TRUE, the function automatically detects factor variables in the dataset for coloring and shaping points. Defaults to TRUE.
factor_table	Logical. If TRUE, prints a detailed table about the properties of the converted factors to the console. Default is FALSE, so no property table will be printed to the console.
color_factor	Character. The name of the factor variable to use for point colors. If set to "auto", it is automatically determined based on detected factors. Defaults to "auto".
shape_factor	Character. The name of the factor variable to use for point shapes. If set to "auto", it is automatically determined based on detected factors. Defaults to "auto".
print_legend	Logical. If TRUE, a separate legend file is created and displayed. Defaults to TRUE.
fancy_names	Named character vector or NULL. Optional mapping of column names to more readable names for display in plots and legends.
width	Numeric. The width of the output plot in centimeters (default 15 cm).
height	Numeric. The height of the output plot in centimeters (default 15 cm).
res	Numeric. The resolution (in dots per inch) for the output plot image (defaults 1000 dpi).
pointsize	Numeric. The base font size for text in the plot image. Defaults to 8.
legendname	Character string or NULL. The name of the file (omit extension) where the legend will be saved. If NULL, a default filename is generated based on the dataset name (dataname_legend_correlation_plot). Defaults to NULL.
close_generated_files	Logical. If TRUE, closes open 'Word' files depending on the output format. This to be able to save the newly generated files. 'Pdf' files should also be closed before using the function and cannot be automatically closed. Default is FALSE.
open_generated_files	Logical. If TRUE, Opens the generated 'Word' output files. This to directly view the results after creation. Files are stored in tempdir(). Default is TRUE.
output_type	Character string specifying the output format: "pdf", "word", "png" or "rmd". Default is "word".
output_file	Character string or NULL. The name of the file (omit extension) where the cor_plot will be saved. If NULL, a default filename is generated based on the dataset name (dataname_correlation_plot). Defaults to NULL.
output_dir	Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.
save_in_wdir	Logical. If TRUE, saves the file in the working directory Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.

Details

- **Factor Detection:** If `detect_factors` is enabled, up to two factors are automatically detected from the dataset and used for coloring (`color_factor`) and shaping (`shape_factor`) points in the plot.
- **Customization:** Users can manually specify which factors to use by setting `color_factor` and/or (`shape_factor`). Non-factor variables are converted into factors automatically, with a message indicating this conversion.
- **Legend Generation:** A separate legend file is created when factors are used or if `print_legend` is explicitly set to `TRUE`.

The function uses numeric variables in the dataset for scatterplots and computes Pearson correlations displayed in the upper triangle of the correlation matrix.

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder (e.g., "`C:/Users/your_username/AppData/Local/Pandoc`") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "`/usr/local/bin`". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "`/usr/bin`" or "`/usr/local/bin`") or manually, and ensure the directory containing Pandoc is in your PATH.
- If Pandoc is not found, this function may not work as intended.

Value

Output is a 'Word' document with:

- A correlation plot (`output_file`).
- A legend (`legendname`) if applicable.

Using the option "`output_type`", it can also generate output in the form of: R Markdown code, 'pdf', or 'PNG' files. No value is returned to the R environment; instead, files are saved, and they are opened automatically if running on Windows.

Note

- At least two numeric variables are required in the dataset; otherwise, an error is thrown.
- If more than two factors are detected, only the first two are used with a warning message.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Example usage:
data("mtcars")

mtcars_sub <- subset(mtcars, select = -c(am, qsec, vs))
# Customizing factors:
f_corplot(mtcars_sub,
          shape_factor = "cyl",
          color_factor = "gear",
          output_type = "png",
          open_generated_files = FALSE
        )

# Output to MS Word and add fancy column names, only adjusting two of the four variable names.
data(iris)
fancy_names <- c(Sepal.Length = "Sepal Length (cm)", Sepal.Width = "Sepal Width (cm)")
f_corplot(iris,
          fancy_names = fancy_names,
          output_type = "word",
          open_generated_files = FALSE
        )
```

f_factors

Convert multiple columns to Factors in a data frame

Description

Converts multiple specified columns of a data frame into factors. If no columns are specified, it automatically detects and converts columns that are suitable to be factors. The function returns the entire data frame including non factor columns and reports the properties of this new data frame in the console.

Usage

```
f_factors(
  data,
  select = NULL,
  exclude = NULL,
  console = FALSE,
  force_factors = FALSE,
  unique_num_treshold = 8,
  repeats_threshold = 2,
  ...
)
```

Arguments

data	A data frame containing the columns to be converted.
select	A character vector specifying the names of the columns to convert into factors. If NULL, the function automatically detects columns that should be factors based on their data type and unique value count. Default is NULL.
exclude	A character vector specifying the names of the columns NOT to convert into factors. If NULL, no columns are excluded. Default is NULL.
console	Logical. If TRUE, prints a detailed table about the properties of the new data frame to the console. Default is TRUE, if FALSE no property table will be printed to the console.
force_factors	Logical. If TRUE all columns in the data.frame will be converted to factors except for the excluded columns using exclude.
unique_num_treshold	Numeric. A threshold of the amount of unique numbers a numeric column should have to keep it numeric, i.e. omit factor conversion. Default 8.
repeats_threshold	Numeric. A threshold of the minimal number of repeats a numeric cols should have to keep convert it to a factor. Default 2.
...	Additional arguments passed to the factor() function of baseR.

Details

- If select is NULL, the function identifies columns with character data or numeric data with fewer than 8 unique values as candidates for conversion to factors.
- The function checks if all specified columns exist in the data frame and stops execution if any are missing.
- Converts specified columns into factors, applying any additional arguments provided.
- Outputs a summary data frame with details about each column, including its type, class, number of observations, missing values, factor levels, and labels.

Value

Returns the modified data frame with the specified (or all suitable) columns converted to factors. Can also force a print of a summary of the data frame's structure to the console (console = TRUE).

Author(s)

Sander H. van Delden <plantmind@proton.me>

See Also

[factor](#)

Examples

```
# Make a data.frame:
df <- data.frame(a = c("yes", "no", "yes", "yes", "no",
                        "yes", "yes", "no", "yes"),
                 b = c(1, 2, 3, 1, 2, 3, 1, 2, 3),
                 c = c("apple", "kiwi", "banana", "apple", "kiwi",
                        "banana", "apple", "kiwi", "banana"),
                 d = c(1.1, 1.1, 3.4, 4.5, 5.4, 6.7, 7.8, 8.1, 9.8)
)
str(df)

# Convert specified columns to factors:
df1 <- f_factors(df, select = c("a", "c"))
str(df1)

# Convert all potential factor columns to factor but exclude column "b":
df2 <- f_factors(df, exclude = c("b"))
str(df2)

# Convert all columns to factor but exclude column "b":
df3 <- f_factors(df, exclude = c("b"), force_factors = TRUE)
str(df3)

# Or automatically detect and convert suitable columns to factors.
# In this example obtaining the same results as above automatically
# and storing it in df2:
df4 <- f_factors(df)
str(df4)

# In example above col b was converted to a factor as the number of repeats = 2
# and the amount of unique numbers < 8. In order to keep b numeric we can also
# adjust the unique_num_treshold and/or repeats_threshold:
df5 <- f_factors(df, unique_num_treshold = 2)
str(df5)
```

f_glm

Perform multiple glm() functions with diagnostics, assumption checking, and post-hoc analysis

Description

Performs Generalized Linear Model (GLM) analysis on a given dataset with options for diagnostics, assumption checking, and post-hoc analysis. Several response parameters can be analyzed in sequence and the generated output can be in various formats ('Word', 'pdf', 'Excel').

Usage

```
f_glm(
  formula,
  family = gaussian(),
  data = NULL,
  diagnostic_plots = TRUE,
  alpha = 0.05,
  adjust = "sidak",
  type = "response",
  show_assumptions_text = TRUE,
  dispersion_test = TRUE,
  output_type = "off",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  close_generated_files = FALSE,
  open_generated_files = TRUE,
  influence_threshold = 2,
  ...
)
```

Arguments

formula	A formula specifying the model to be fitted. More response variables can be added using - or + (e.g., response1 + response2 ~ predictor) to do a sequential GLM for each response parameter.
family	The error distribution and link function to be used in the model (default: gaussian()). This can be a character string naming a family function, a family function or the result of a call to a family function. (See family for details of family functions.)
data	A data frame containing the variables in the model.
diagnostic_plots	Logical. If TRUE, plots are included in the output files.
alpha	Numeric. Significance level for tests. Default is 0.05.
adjust	Character string specifying the method used to adjust p-values for multiple comparisons. Available methods include: "tukey" Tukey's Honest Significant Difference method "sidak" Šidák correction "bonferroni" Bonferroni correction "none" No adjustment "fdr" False Discovery Rate adjustment Default is "sidak".
type	specifying the scale on which the emmeans posthoc results are presented, e.g. "link" to show results on the scale for which the variables are linear and "response" when you want to back transform the data to interpret results in the

	units of your original data (e.g., probabilities, counts, or untransformed measurements). Default is "response".
show_assumptions_text	Logical. If TRUE, includes a short explanation about GLM assumptions in the output file.
dispersion_test	Logical for overdispersion test (default: TRUE).
output_type	Character string specifying the output format: "pdf", "word", "excel", "rmd", "off" (no file generated) or "console". The option "console" forces output to be printed. Default is "off".
output_file	Character string specifying the name of the output file. Default is "dataname_glm_output".
output_dir	Character string specifying the name of the directory of the output file. Default is tempdir().
save_in_wdir	Logical. If TRUE, saves the file in the working directory.
close_generated_files	Logical. If TRUE, closes open 'Excel' or 'Word' files depending on the output format. This to be able to save the newly generated file by the f_aov() function. 'Pdf' files should also be closed before using the function and cannot be automatically closed. Default is FALSE.
open_generated_files	Logical. If TRUE, Opens the generated output files ('pdf', 'Word' or 'Excel') files depending on the output format. This to directly view the results after creation. Files are stored in tempdir(). Default is TRUE.
influence_threshold	Leverage threshold (default: 2).
...	Additional arguments passed to glm().

Details

The function first checks if all specified variables are present in the data and ensures that the response variable is numeric.

It performs Analysis of Variance (ANOVA) using the specified formula and data. If shapiro = TRUE, it checks for normality of residuals using the Shapiro-Wilk test and optionally (transformation = TRUE) applies a data transformation if residuals are not normal.

If significant differences are found in ANOVA, it proceeds with post hoc tests using estimated marginal means from emmeans() and Sidak adjustment (or another option of adjust =).

More response variables can be added using - or + (e.g., response1 + response2 ~ predictor) to do a sequential aov() for each response parameter captured in one output file.

Outputs can be generated in multiple formats ("pdf", "word", "excel" and "rmd") as specified by output_type. The function also closes any open 'Word' files to avoid conflicts when generating 'Word' documents. If output_type = "rmd" is used it is advised to use it in a chunk with {r, echo=FALSE, results='asis'}

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "/usr/local/bin". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "/usr/bin" or "/usr/local/bin") or manually, and ensure the directory containing Pandoc is in your PATH.
- If Pandoc is not found, this function may not work as intended.

Value

An object of class 'f_glm' containing results from `glm()`, diagnostics, and post-hoc tests. Using the option "output_type", it can also generate output in the form of: R Markdown code, 'Word', 'pdf', or 'Excel' files. Includes print and plot methods for 'f_glm' objects.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# GLM Binomial example with output to console and MS Word file
mtcars_mod <- mtcars
mtcars_mod$cyl <- as.factor(mtcars_mod$cyl)

glm_bin <- f_glm(vs ~ cyl,
                family = binomial,
                data = mtcars_mod,
                output_type = "word",
                # Do not automatically open the 'Word' file (Default is to open the file)
                open_generated_files = FALSE)

print(glm_bin)

# GLM Poisson example with output to rmd text
data(warpbreaks)

glm_pos <- f_glm(breaks ~ wool + tension,
                data = warpbreaks,
                family = poisson(link = "log"),
                show_assumptions_text = FALSE,
                output_type = "rmd")
cat(cat(glm_pos$rmd))
```

f_hist

*Plot a Histogram with an Overlaid Normal Curve***Description**

This function creates a histogram of the provided data and overlays it with a normal distribution curve.

Usage

```
f_hist(
  data,
  main = NULL,
  xlab = NULL,
  probability = TRUE,
  col = "white",
  border = "black",
  line_col = "red",
  save_png = FALSE,
  open_png = TRUE,
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  width = 8,
  height = 7,
  units = "in",
  res = 300,
  ...
)
```

Arguments

data	A numeric vector of data values to be plotted.
main	A character string specifying the title of the histogram. Default is "Histogram with Normal Curve".
xlab	A character string specifying the label for the x-axis. Default is the name of the data variable.
probability	A logical value indicating whether to plot a probability or frequency histogram. Default is TRUE.
col	A character string specifying the fill color of the histogram bars. Default is "white".
border	A character string specifying the color of the histogram bar borders. Default is "black".
line_col	A character string specifying the color of the normal curve line. Default is "red".

save_png	A logical value default FALSE, if TRUE a png file is saved under the name of the data of under the specified file name.
open_png	Logical. If TRUE, opens generated png files.
output_file	Character string specifying the name of the output file (without extension). Default is the name of the vector or dataframe followed by "_histogram.png".
output_dir	Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.
save_in_wdir	Logical. If TRUE, saves the file in the working directory Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.
width	Numeric, png figure width default 8 inch.
height	Numeric, png figure height default 7 inch.
units	Character string, png figure units default "in" = inch, other options are: "px" = Pixels, "cm" centimeters, "mm" millimeters.
res	Numeric, png figure resolution default 300 dpi.
...	Additional arguments to be passed to the hist function.

Details

The function first captures the name of the input variable for labeling purposes. It then calculates a sequence of x-values and corresponding y-values for a normal distribution based on the mean and standard deviation of the data. The histogram is plotted with specified aesthetics, and a normal curve is overlaid. To increase resolution you can use `png(..., res = 600)` or the 'RStudio' chunk setting, e.g. `dpi=600`.

Value

A histogram plot is created and the function returns this as a recordedplot.

Author(s)

Sander H. van Delden <plantmind@proton.me>

See Also

`hist`, `dnorm`, `lines`

Examples

```
# Example usage:
set.seed(123)
sample_data <- rnorm(100)
f_hist(sample_data)
```

f_kruskal_test	<i>Perform multiple Kruskal-Wallis tests with a user-friendly output file, do data inspection and Dunn's test (of 'rstatix') as post hoc.</i>
----------------	---

Description

Performs the Kruskal-Wallis rank sum test to assess whether there are statistically significant differences between three or more independent groups. It provides detailed outputs, including plots, assumption checks, and post-hoc analyses using Dunn's test. Results can be saved in various formats ('pdf', 'Word', 'Excel', or console only) with customizable output options.

Usage

```
f_kruskal_test(
  formula,
  data = NULL,
  plot = TRUE,
  alpha = 0.05,
  output_type = "off",
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  kruskal_assumptions_text = TRUE,
  adjust = "bonferroni",
  close_generated_files = FALSE,
  open_generated_files = TRUE
)
```

Arguments

formula	A formula specifying the response and predictor variable (e.g., response ~ predictor). more response variables and predictors can be added using - or + (e.g., response1 + response2 ~ predictor1 + predictor2). The function iterates through these combinations or response and predictors, because the Kruskal-Wallis test itself only allows one response and one predictor combination to be tested simultaneously.
data	A data.frame containing the variables referenced in the formula.
plot	Logical. If TRUE, generates plots (e.g., density plots and boxplots) in the output files. Default is TRUE.
alpha	Numeric. The significance level for the Kruskal-Wallis test and Dunn's test. Default is 0.05.
output_type	Character string. Specifies the output format: "pdf", "word", "excel", "rmd", "off" (no file generated) or "console". The option "console" forces output to be printed. Default is "off".
output_file	Character string. The name of the output file (without extension). If NULL, a default name is generated based on the dataset name. Default is NULL.

<code>output_dir</code>	Character string specifying the name of the directory of the output file. Default is <code>tempdir()</code> . If the <code>output_file</code> already contains a directory name <code>output_dir</code> can be omitted, if used it overwrites the dir specified in <code>output_file</code> .
<code>save_in_wdir</code>	Logical. If TRUE, saves the file in the working directory. Default is FALSE, to avoid unintended changes to the global environment. If the <code>output_dir</code> is specified <code>save_in_wdir</code> is overwritten with <code>output_dir</code> .
<code>kruskal_assumptions_text</code>	Logical. If TRUE, includes a section about Kruskal-Wallis test assumptions in the output document. Default is TRUE.
<code>adjust</code>	Character string. Adjustment method for pairwise comparisons in Dunn's test. Options include "holm", "hommel", "bonferroni", "sidak", "hs", "hochberg", "bh", "by", "fdr" or "none". Default is "bonferroni", if you don't want to adjust the p value (not recommended), use <code>p.adjust.method = "none"</code> .
<code>close_generated_files</code>	Logical. If TRUE, closes open 'Excel' or 'Word' files depending on the output format. This to be able to save the newly generated files. 'Pdf' files should also be closed before using the function and cannot be automatically closed. Default is FALSE.
<code>open_generated_files</code>	Logical. If TRUE, opens the generated output files ('pdf', 'Word' or 'Excel') files depending on the output format. This to directly view the results after creation. Files are stored in <code>tempdir()</code> . Default is TRUE.

Details

This function offers a comprehensive workflow for non-parametric analysis using the Kruskal-Wallis test:

- **Assumption Checks:** Optionally includes a summary of assumptions in the output.
- **Visualization:** Generates density plots and boxplots to visualize group distributions.
- **Post-hoc Analysis:** Conducts Dunn's test with specified correction methods if significant differences are found.

Output files are generated in the format specified by `output_type =` and saved to the working directory, options are "pdf", "word" or "excel". If `output_type = "rmd"` is used it is advised to use it in a chunk with `{r, echo=FALSE, results='asis'}`

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "/usr/local/bin". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "/usr/bin" or "/usr/local/bin") or manually, and ensure the directory containing Pandoc is in your PATH.

- If Pandoc is not found, this function may not work as intended.

Value

An object of class 'f_kruskal_test' containing:

- Kruskal-Wallis test results for each combination of response and predictor variables.
- Dunn's test analysis results (if applicable).
- Summary tables with compact letter displays for significant group differences.

Using the option `output_type`, it can also generate output in the form of: R Markdown code, 'Word', 'pdf', or 'Excel' files. Includes print and plot methods for 'f_kruskal_test' objects.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Example usage:
data(iris)

# Perform Kruskal-Wallis test on Sepal.Length and Sepal.Width by Species
# with "holm" correction for posthoc dunn_test, without showing the output.
output <- f_kruskal_test(
  Sepal.Width + Sepal.Length ~ Species,
  data = iris,
  plot = FALSE,
  output_type = "word",
  adjust = "holm",
  open_generated_files = FALSE
)

# Save Kruskal-Wallis test and posthoc to Excel sheets: Sepal.Width and Sepal.Length.
f_kruskal_out <- f_kruskal_test(
  Sepal.Width + Sepal.Length ~ Species,
  data = iris,
  plot = FALSE,
  output_type = "excel",
  adjust = "holm",
  open_generated_files = FALSE
)
```

`f_load_packages`*Install and Load Multiple R Packages*

Description

Checks if the specified packages are installed. If not, it installs them and then loads them into the global R session.

Usage

```
f_load_packages(...)
```

Arguments

... Unquoted or quoted names of packages to be installed and loaded. These should be valid package names available on CRAN.

Details

The function takes a list or vector indicating package names, installs any that are missing, and loads all specified packages into the global environment of the R session. It uses `requireNamespace()` to check for installation and `library()` to load the packages.

Value

None. The function is called for its side effects of installing and loading packages.

Author(s)

Sander H. van Delden <plantmind@proton.me>

`f_model_comparison`*Compare Two Statistical Models*

Description

Compares two statistical models by calculating key metrics such as AIC, BIC, log-likelihood, R-squared, and others. Supports comparison of nested models using ANOVA tests.

Usage

```
f_model_comparison(model1, model2, nested = NULL, digits = 3)
```

Arguments

model1	The first model object. Supported classes include: "lm", "glm", "aov", "lmerMod", "glmerMod", and "nls".
model2	The second model object. Supported classes include: "lm", "glm", "aov", "lmerMod", "glmerMod", and "nls".
nested	Logical. If TRUE, assumes the models are nested and performs an ANOVA comparison. If NULL (default), the function attempts to automatically determine if the models are nested.
digits	Integer. The number of decimal places to round the output metrics. Defaults to 3.

Details

Calculate various metrics to assess model fit:

- **AIC/BIC:** Lower values indicate better fit.
- **Log-Likelihood:** Higher values (less negative) indicate better fit.
- **R-squared:** Proportion of variance explained by the model.
- **Adjusted R-squared:** R-squared penalized for the number of parameters (for linear models).
- **Nagelkerke R²:** A pseudo-R² for generalized linear models (GLMs).
- **Marginal/Conditional R²:** For mixed models, marginal R² reflects fixed effects, while conditional R² includes random effects.
- **Sigma:** Residual standard error.
- **Deviance:** Model deviance.
- **SSE:** Sum of squared errors.
- **Parameters (df):** Number of model parameters.
- **Residual df:** Residual degrees of freedom.

If the models are nested, an ANOVA test is performed to compare them, and a p-value is provided to assess whether the more complex model significantly improves fit.

Value

A list of class "f_model_comparison" containing:

model1_name	The name of the first model.
model2_name	The name of the second model.
model1_class	The class of the first model.
model2_class	The class of the second model.
metrics_table	A data frame summarizing metrics for both models, their differences, and (if applicable) the ANOVA p-value.
formatted_metrics_table	A formatted version of the metrics table for printing.
anova_comparison	The ANOVA comparison results if the models are nested and an ANOVA test was performed.
nested	Logical indicating whether the models were treated as nested.

Supported Model Classes

The function supports the following model classes:

- Linear models ("lm")
- Generalized linear models ("glm")
- Analysis of variance models ("aov")
- Linear mixed models ("lmerMod")
- Generalized linear mixed models ("glmerMod")
- Nonlinear least squares models ("nls")

Note

- The function supports a variety of model types but may issue warnings if unsupported or partially supported classes are used.
- For GLMs, Nagelkerke's R^2 is used as a pseudo- R^2 approximation.
- For mixed models, the function relies on the 'r.squaredGLMM' function from the 'MuMIn' package for R^2 calculation.
- The idea of this function (not the code), I got from Dustin Fife's function '[model.comparison](#)' in the super cool '[flexplot package](#)'.

Author(s)

Sander H. van Delden <plantmind@proton.me>

See Also

[AIC](#), [BIC](#), [anova](#), [logLik](#), [r.squaredGLMM](#)

Examples

```
# Example with linear models.
model1 <- lm(mpg ~ wt, data = mtcars)
model2 <- lm(mpg ~ wt + hp, data = mtcars)
comparison <- f_model_comparison(model1, model2)
print(comparison)

# Example with GLMs.

model1 <- glm(am ~ wt, data = mtcars, family = binomial)
model2 <- glm(am ~ wt + hp, data = mtcars, family = binomial)
comparison <- f_model_comparison(model1, model2)
print(comparison)

# Example with automatic detection of nested models.
model1 <- lm(mpg ~ wt, data = mtcars)
model2 <- lm(mpg ~ wt + hp, data = mtcars)
```

```
comparison <- f_model_comparison(model1, model2)
print(comparison)
```

f_open_file*Open a File with the Default Application*

Description

Opens a specified file using the default application associated with its file type. It automatically detects the operating system (Windows, Linux, or macOS) and uses the appropriate command to open the file.

Usage

```
f_open_file(filepath)
```

Arguments

filepath	A character string specifying the path to the file to be opened. The path can be absolute or relative.
----------	--

Details

- On **Windows**, the `f_open_file()` function uses `shell.exec()` to open the file. - On **Linux**, it uses `xdg-open` via the `system()` function. - On **macOS**, it uses `open` via the `system()` function.

If an unsupported operating system is detected, the function will throw a message.

Value

Does not return a value; it is called for its side effect of opening a file.

Author(s)

Sander H. van Delden <plantmind@proton.me>

See Also

[`shell.exec()`], [`system()`]

Examples

```
# NOTE: The use of "if(interactive())" prevents this example from running
# during automated CRAN checks. This is necessary because the example
# opens a file, a behavior restricted by CRAN policies for automated
# testing. You don't need to use "if(interactive())" in your own scripts.
if(interactive()) {
  # Open a PDF file.
  f_open_file("example.pdf")

  # Open an image file.
  f_open_file("image.png")

  # Open a text file.
  f_open_file("document.txt")
}
```

f_pander	<i>Fancy Pander Table Output</i>
----------	----------------------------------

Description

Is a wrapper around the pander function from the 'pander' package, designed to produce a fancy table output with specific formatting options.

Usage

```
f_pander(table, col_width = 10, table_width = NULL, ...)
```

Arguments

table	A data frame, matrix, or other table-like structure to be rendered.
col_width	Integer. Specifies the maximum number of characters allowed in table header columns before a line break is inserted. Defaults to 10.
table_width	Integer or NULL. Defines the number of characters after which the table is split into separate sections. Defaults to NULL, meaning no break is applied.
...	Additional arguments passed to the pander function.

Details

This function sets several pander options to ensure that the table output is formatted in a visually appealing manner. The options set include:

- `table.alignment.default`: Aligns all columns to the left.
- `table.alignment.rownames`: Aligns row names to the left.
- `keep.trailing.zeros`: Keeps trailing zeros in numeric values.
- `knitr.auto.asis`: Ensures output is not automatically treated as 'asis'.

- `table.split.table`: Prevents splitting of tables across pages or slides.
- `table.caption.prefix`: Removes the default "Table" prefix in captions.

This function requires [Pandoc](<https://github.com/jgm/pandoc/releases/tag>) (version 1.12.3 or higher), a universal document converter.

- **Windows:** Install Pandoc and ensure the installation folder (e.g., "C:/Users/your_username/AppData/Local/Pandoc") is added to your system PATH.
- **macOS:** If using Homebrew, Pandoc is typically installed in "/usr/local/bin". Alternatively, download the .pkg installer and verify that the binary's location is in your PATH.
- **Linux:** Install Pandoc through your distribution's package manager (commonly installed in "/usr/bin" or "/usr/local/bin") or manually, and ensure the directory containing Pandoc is in your PATH.
- If Pandoc is not found, this function may not work as intended.

Value

None. The function is called for its side effects of setting 'pander' options and creates a pander formatted table in R Markdown.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Example usage of f_pander
df <- data.frame(
  Name = c("Alice", "Bob", "Charlie"),
  Age = c(25, 30, 35),
  Score = c(88.5, 92.3, 85.0)
)

# Render the data frame as a fancy table
f_pander(df)
```

f_qqnorm

Normal Q-Q Plot with Confidence Bands

Description

This function creates a normal Q-Q plot for a given numeric vector and adds confidence bands to visualize the variability of the quantiles.

Usage

```
f_qqnorm(
  x,
  main = NULL,
  ylab = NULL,
  conf_level = 0.95,
  col = NULL,
  pch = NULL,
  cex = NULL,
  save_png = FALSE,
  open_png = TRUE,
  output_file = NULL,
  output_dir = NULL,
  save_in_wdir = FALSE,
  width = 8,
  height = 7,
  units = "in",
  res = 300,
  ...
)
```

Arguments

x	A numeric vector of data values.
main	A character string specifying the title of the histogram. Default is "Histogram with Normal Curve".
ylab	A character string specifying the y-axis label. Default name is "Quantiles of: data_name".
conf_level	Numeric, between 0 and 1. Confidence level for the confidence bands. Default is 0.95 (95% confidence).
col	Numeric, optional parameter for color of point with default 'black'.
pch	Numeric, optional parameter shape of points default pch = 19.
cex	Numeric, optional parameter for graph cex with default cex = 0.6.
save_png	A logical value default FALSE, if TRUE a png file is saved under the name of the data of under the specified file name.
open_png	Logical. If TRUE, opens generated png files.
output_file	Character string specifying the name of the output file (without extension). Default is the name of the vector or dataframe followed by "_histogram.png".
output_dir	Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.
save_in_wdir	Logical. If TRUE, saves the file in the working directory Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.
width	Numeric, png figure width default 8 inch.

height	Numeric, png figure height default 7 inch.
units	Numeric, png figure units default inch.
res	Numeric, png figure resolution default 300 dpi.
...	Additional graphical parameters to be passed to the qqnorm function.

Details

The function calculates theoretical quantiles for a normal distribution and compares them with the sample quantiles of the input data.

It also computes confidence intervals for the order statistics using the Blom approximation and displays these intervals as shaded bands on the plot.

The reference line is fitted based on the first and third quartiles of both the sample data and theoretical quantiles.

To increase resolution you can use `png(..., res = 600)` or the 'RStudio' chunk setting, e.g. `dpi = 600`.

Value

A Q-Q plot is created and the function returns this as a `recordedplot`.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Generate random normal data
set.seed(123)
data <- rnorm(100)

# Create a Q-Q plot with confidence bands
f_qqnorm(data)

# Customize the plot with additional graphical parameters
f_qqnorm(data, conf_level = 0.99, pch = 16, col = "blue")
```

f_rename_columns	<i>Rename Specific Columns in a Data Frame</i>
------------------	--

Description

Renames specific columns in a data frame based on a named vector (`name_map`). It ensures that only the specified columns are renamed, while others remain unchanged.

Usage

```
f_rename_columns(df, name_map)
```

Arguments

df	A data frame whose columns are to be renamed.
name_map	A named vector where the names correspond to the current column names in df, and the values are the new names to assign. All names in name_map must exist in the column names of df. Yet, not all names in the data.frame have to be in name_map. This allows for selective renaming of just one or two columns.

Details

This function is particularly useful when you want to rename only a subset of columns in a data frame. It performs input validation to ensure that:

- name_map is a named vector.
- All names in name_map exist as column names in df.

If these conditions are not met, the function will throw an error with an appropriate message.

Value

A data frame with updated column names. Columns not specified in name_map remain unchanged.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Create a sample data frame.
df <- data.frame(a = 1:3, b = 4:6, c = 7:9)

# Define a named vector for renaming specific columns.
name_map <- c(a = "alpha", c = "gamma")

# Rename columns.
df <- f_rename_columns(df, name_map)

# View updated data frame.
print(df)
```

f_rename_vector	<i>Rename Elements of a Vector Based on a Mapping</i>
-----------------	---

Description

Renames elements of a vector based on a named mapping vector. Elements that match the names in the mapping vector are replaced with their corresponding values, while elements not found in the mapping remain unchanged.

Usage

```
f_rename_vector(vector, name_map)
```

Arguments

vector	A character vector containing the elements to be renamed.
name_map	A named vector where the names correspond to the elements in vector that should be renamed, and the values are the new names to assign.

Details

This function iterates through each element of vector and checks if it exists in the names of name_map. If a match is found, the element is replaced with the corresponding value from name_map. If no match is found, the original element is retained. The result is returned as an unnamed character vector.

Value

A character vector with updated element names. Elements not found in name_map remain unchanged.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Define a vector and a name map.
vector  <- c("Species", "Weight", "L")
name_map <- c(Species = "New_species_name", L = "Length_cm")

# Rename elements of the vector.
updated_vector <- f_rename_vector(vector, name_map)

# View updated vector
print(updated_vector)
```

f_setwd*Set Working Directory Based on Current File or Specified Path*

Description

A wrapper around `setwd()` that sets the working directory to the location of the currently open file in 'RStudio' if no path is provided. If a path is specified, it sets the working directory to that path instead.

Usage

```
f_setwd(path = NULL)
```

Arguments

path	A character string specifying the desired working directory. If <code>NULL</code> (default), the function sets the working directory to the location of the currently open and saved file in 'RStudio'.
------	---

Details

If `path` is not provided (`NULL`), this function uses the `this.path` package to determine the location of the currently open file and sets that as the working directory. The file must be saved for this to work properly.

If a valid `path` is provided, it directly sets the working directory to that path.

Value

None. The function is called for its side effects of changing the working directory.

Note

- The function checks whether the currently open file is saved before setting its location as the working directory.
- If the function is called from an unsaved script or directly from the console, an error will be thrown.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# NOTE: The use of "if(interactive())" prevents this example from running
# during automated CRAN checks. This is necessary because the example
# requires to be run from an R script. You don't need to use
# "if(interactive())" in your own scripts.
if(interactive()) {
  # Store the current working directory, so we can reset it after the example.
  current_wd <- getwd()
  print(current_wd)

  # Run this commando from a saved R script file, or R Notebook to set the working
  # directory to scripts' file location
  f_setwd()

  # Restore your current working directory
  f_setwd(current_wd)
}
```

f_summary

Summarize a Data Frame with Grouping Variables

Description

Computes summary statistics (e.g., mean, standard deviation, median, etc.) for a specified column ("character string") in a data frame, grouped by one or more grouping variables in that data frame ("character strings"). Summary parameters can be customized and the results can be exported to an 'Excel' file.

Usage

```
f_summary(
  data,
  data.column,
  ...,
  show_n = TRUE,
  show_mean = TRUE,
  show_sd = TRUE,
  show_se = TRUE,
  show_min = TRUE,
  show_max = TRUE,
  show_median = TRUE,
  show_Q1 = TRUE,
  show_Q3 = TRUE,
  digits = 2,
  export_to_excel = FALSE,
  close_generated_files = FALSE,
```

```

    open_generated_files = TRUE,
    output_file = NULL,
    output_dir = NULL,
    save_in_wdir = FALSE,
    open_excel = TRUE,
    check_input = TRUE,
    eval_input = FALSE,
    digits_excel = NULL,
    detect_int_col = TRUE
  )

```

Arguments

<code>data</code>	A 'data.frame', 'data.table' or 'tibble', i.e. input data to be summarized.
<code>data.column</code>	A character string, vector or list with characters. The name of the column(s) in data for which summary statistics will be calculated.
<code>...</code>	One or more character strings specifying the grouping variables in data. At least one grouping variable must be provided.
<code>show_n</code>	Logical. If TRUE, the summary results n will be included in the output.
<code>show_mean</code>	Logical. If TRUE, the summary results mean will be included in the output.
<code>show_sd</code>	Logical. If TRUE, the summary results sd will be included in the output.
<code>show_se</code>	Logical. If TRUE, the summary results se will be included in the output.
<code>show_min</code>	Logical. If TRUE, the summary results min will be included in the output.
<code>show_max</code>	Logical. If TRUE, the summary results max will be included in the output.
<code>show_median</code>	Logical. If TRUE, the summary results median will be included in the output.
<code>show_Q1</code>	Logical. If TRUE, the summary results Q1 will be included in the output.
<code>show_Q3</code>	Logical. If TRUE, the summary results Q3 will be included in the output.
<code>digits</code>	Integer. Round to the number of digits specified. If digits = NULL no rounding is applied (default is digits = 2). Note that this rounding is independent of the rounding in the exported excel file.
<code>export_to_excel</code>	Logical. If TRUE, the (unrounded values) summary results will be exported to an 'Excel' file. Default is FALSE.
<code>close_generated_files</code>	Logical. If TRUE, closes open 'Excel' files. This to be able to save the newly generated file. Default is FALSE.
<code>open_generated_files</code>	Logical. If TRUE, Opens the generated 'Excel' files. This to directly view the results after creation. Files are stored in tempdir(). Default is TRUE.
<code>output_file</code>	Character string specifying the name of the output file. Default is "dataname_summary.xlsx".
<code>output_dir</code>	Character string specifying the name of the directory of the output file. Default is tempdir(). If the output_file already contains a directory name output_dir can be omitted, if used it overwrites the dir specified in output_file.

save_in_wdir	Logical. If TRUE, saves the file in the working directory. Default is FALSE, to avoid unintended changes to the global environment. If the output_dir is specified save_in_wdir is overwritten with output_dir.
open_excel	Logical. If TRUE and export_to_excel is also TRUE, the generated 'Excel' file will be opened automatically. Default is TRUE.
check_input	If TRUE, checks the input and stops the function if the input is incorrect (default is TRUE).
eval_input	Logical. If TRUE, the function evaluates the third function argument. This should be a character vector with the group by columns. Default is FALSE, which allows group by columns to be written without quotes.
digits_excel	Integer. Round cells in the excel file to the number of digits specified. If digits_excel = NULL no rounding is applied (default is digits_excel = NULL). Note to preserve formatting numbers will be stored as text.
detect_int_col	Logical. If TRUE, columns in a data.frame containing only integers will be displayed without decimal digits. Columns containing a mix of integers and decimal values will display all values with the specified number of digits. If FALSE, each individual cell is evaluated: integer values are displayed without digits, and numbers containing digits with the specified number of digits. Default is TRUE.

Details

The function computes the following summary statistics for the specified column:

- n: number of observations
- mean: mean
- sd: standard deviation
- se: standard error of the mean
- min: minimum value
- max: maximum value
- median: median
- Q1: first quartile
- Q3: third quartile

Each of these summary statistics can be removed by setting e.g. show_n = FALSE. The results are grouped by the specified grouping variables and returned as a data frame. If export_to_excel is set to TRUE, the results are saved as an 'Excel' file in the working directory with a dynamically generated filename.

Value

A data frame containing the computed summary statistics, grouped by the specified variables. This data frame can be automatically saved as an 'Excel' file using export_to_excel = TRUE.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# Example usage:
# Create a summary of mtcars for data column hp grouped by cyl and gear,
# and remove Q1 and Q3 from the output.
# Note that variable can be written as "hp" or as hp. Only data.frame must be data (no quotes)
summary_mtcars <- f_summary(mtcars, "hp", "cyl", "gear", show_Q1 = FALSE, show_Q3 = FALSE)
print(summary_mtcars)

# Create a summary for iris
summary_iris <- f_summary(iris, Sepal.Length, Species)

# Print the a table with column width of 10 characters and table length of 70 characters
print(summary_iris, col_width = 10, table_width = 70)
```

f_theme

Apply a black or white 'RStudio' Theme and Zoom Level

Description

This comes in hand when teaching, the function allows users to apply a "black" or "white" 'RStudio' theme and adjust the zoom level in the 'RStudio' IDE. It includes error handling for invalid inputs.

Usage

```
f_theme(color = "black", zlevel = 0)
```

Arguments

color	A character string. The theme color to apply. Must be either "black" (dark theme) or "white" (light theme). Default is "black".
zlevel	A numeric value. The zoom level to apply, ranging from 0 (default size) to 4 (maximum zoom). Default is 0.

Details

The function performs the following actions:

1. Applies the specified 'RStudio' theme:
 - "black": Applies the "Tomorrow Night 80s" dark theme.
 - "white": Applies the "Textmate (default)" light theme.
2. Adjusts the zoom level in 'RStudio':
 - zlevel = 0: Resets to default zoom level.
 - zlevel = 1: Zooms in once.
 - zlevel = 2: Zooms in twice.
 - zlevel = 3: Zooms in three times.

- `zlevel = 4`: Zooms in four times.

The function includes error handling to ensure valid inputs:

- `color` must be a character string and one of "black" or "white".
- `zlevel` must be a numeric value, an integer, and within the range of 0 to 4. If a non-integer is provided, it will be rounded to the nearest integer with a warning.

Value

None. The function is called for its side effects of changing the 'RStudio' theme or Zoomlevel.

This function does not return a value. It applies changes directly to the 'RStudio' IDE.

Author(s)

Sander H. van Delden <plantmind@proton.me>

Examples

```
# NOTE: This example will change your RStudio theme hence the dont run warning.
## Not run:
# Apply a dark theme with with zoom level 2:
f_theme(color = "black", zlevel = 2)

# Apply a black theme with maximum zoom level:
f_theme(color = "black", zlevel = 4)

# Apply the default light theme default zoom level:
f_theme(color = "black", zlevel = 0)

## End(Not run)
```

`plot.f_bestNormalize` *Plot an f_bestNormalize object*

Description

Plots diagnostics for an object of class `f_bestNormalize`.

Usage

```
## S3 method for class 'f_bestNormalize'
plot(x, which = 1:2, ask = FALSE, ...)
```

Arguments

x	An object of class f_bestNormalize.
which	Integer determining which graph to plot. Default is 1:2.
ask	Logical. TRUE waits with plotting each graph until <Return> is pressed. Default is FALSE.
...	Further arguments passed to or from other methods.

Details

Plot method for f_bestNormalize objects

Value

This function is called for its side effect of generating plots and does not return a useful value. It invisibly returns 'NULL'.

plot.f_boxcox	<i>Plot an f_boxcox object</i>
---------------	--------------------------------

Description

Create diagnostic plots of an object of class f_boxcox.

Usage

```
## S3 method for class 'f_boxcox'
plot(x, which = 1:3, ask = FALSE, ...)
```

Arguments

x	An object of class f_boxcox.
which	Integer determining which graph to plot. Default is 1:2.
ask	Logical. TRUE waits with plotting each graph until <Return> is pressed. Default is FALSE.
...	Further arguments passed to or from other methods.

Details

Plot method for f_boxcox objects

Value

This function is called for its side effect of generating plots and does not return a useful value. It invisibly returns 1.

print.f_summary	<i>Print method for f_summary objects</i>
-----------------	---

Description

This function prints f_summary objects.

Usage

```
## S3 method for class 'f_summary'  
print(x, col_width = 6, table_width = 90, ...)
```

Arguments

x	Object of class f_summary
col_width	Integer. Specifies the maximum number of characters allowed in table header columns before a line break is inserted. Defaults to 10.
table_width	Integer or NULL. Defines the number of characters after which the table is split into separate sections. Defaults to NULL, meaning no break is applied.
...	Additional arguments passed to the pander function.

Value

This function is called for its side effect of printing a formatted output to the console and does not return a useful value. It invisibly returns 1.

Index

AIC, [36](#)
anova, [36](#)

BIC, [36](#)

chisq.test, [15](#), [16](#)

f_aov, [3](#)
f_bestNormalize, [6](#)
f_boxcox, [8](#)
f_boxplot, [12](#)
f_chisq_test, [15](#)
f_clear, [17](#)
f_conditional_round, [18](#)
f_corplot, [20](#)
f_factors, [23](#)
f_glm, [25](#)
f_hist, [29](#)
f_kruskal_test, [31](#)
f_load_packages, [34](#)
f_model_comparison, [34](#)
f_open_file, [37](#)
f_pander, [38](#)
f_qqnorm, [39](#)
f_rename_columns, [41](#)
f_rename_vector, [43](#)
f_setwd, [44](#)
f_summary, [45](#)
f_theme, [48](#)
family, [26](#)

logLik, [36](#)

plot.f_bestNormalize, [49](#)
plot.f_boxcox, [50](#)
print.f_summary, [51](#)

r.squaredGLMM, [36](#)