Package 'LandS'

October 15, 2024

Type Package

Title Biostatistic Tools for DCP

Version 1.0.1

Author Stefano Bergamini and Luca Lalli

Maintainer Nobody OfCourse <stefano.bergamini@istitutotumori.mi.it>

Description This package provides useful functions in daily clinical pratice for biostatistics

License No you can steal it when you want

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Contents

Description

This function creates a list of boxplot

Usage

```
Boxplot_LB(
  data,
  variables,
  group,
  ID_lines = FALSE,
  Posthoc = FALSE,
  Point = F,
  Median_line = F,
  rm.outliers = F,
  alpha_box = 0.1,
  width_box = 0.2,
  size_median_line = 0.8,
  lwd_box = 0.1,
```

```
  lwd_ID_line = 0.2,
  alpha_ID_line = 0.3,
  alpha_point = 0.3,
  size_point = 0.3,
  Test_results = NULL,
  threshold_posthoc = 0.1,
  axis_y_title = NULL,
  axis_x_title = NULL,
  size_axis_x = 6,
  size_axis_y = 6,
  ID = "ID",
  legend_cod = NULL,
  breaks_axis_x = levels(data[, group]),
  labels_axis_x = levels(data[, group]),
  grid = TRUE,
  PPTX = FALSE,
  pptx_width = 7.5,
  pptx_height = 5.5,
  extra = F,
  extra_text = NULL,
  palette_boxplot = c("salmon", "royalblue", "forestgreen", "gold"),
  palette_title = "black",
  size_title = 12,
  target = paste0(path_out, "/Boxplot.pptx"),
  ratio = 1,
  telegram = "none"
)
```

Arguments

data	dataframe
variables	vector containing all variables of interest
group	factor variable splitting the data
ID_lines	whether to print the lines for paired observations
Posthoc	whether to display Posthoc tests
Point	whether to display observation points
Median_line	whether to display the line connecting medians
rm.outliers	whether to remove outliers from display
alpha_box	alpha parameter for the boxes
width_box	box's width
size_median_line	median linewidth
lwd_box	box linewidth
lwd_ID_line	linewidth for paired observations
alpha_ID_line	alpha for paired observations
alpha_point	alpha for points
size_point	size for points
Test_results	dataframe for global and posthoc tests, see cont_var_test_LB

threshold_posthoc	threshold for displaying posthoc tests
size_axis_x	axis y title dimension
size_axis_y	axis x title dimension
ID	ID variable
telegram	

Value

Una lista di boxplot

`cont_var_test_LB` *Test for continuous variables splitted by categories*

Description

The most powerful function ever created. You can perform the 4 major tests and the posthoc tests for Friedman and Kruskal-Wallis. If you are dumb (option dumb = T) you can also perform posthoc tests without correcting for test multiplicity. Please do not try this at home/work and consider asking a statistician before performing any test.

Usage

```
cont_var_test_LB(
  data,
  variables,
  paired = FALSE,
  group,
  dumb = FALSE,
  ID = "ID",
  num_dec = 2,
  excel = F,
  excel_path = paste0(path_out, "/Results.xlsx"),
  telegram = "none"
)
```

Arguments

data	dataframe
variables	vector containing all variables of interest
paired	FALSE/TRUE
group	factor variable splitting the data
dumb	FALSE are you dumb? Hope not
ID	ID variabl (Default = "ID")
num_dec	Decimal number for mean and SD (Default = 2)
excel	export fuction results as multiple Excel sheets
excel_path	path where you want your Excel
telegram	send a telegram message

Value

Una lista con dataset

Examples

```
cont_var_test_LB(data = iris, variables = c("Sepal.Length", "Sepal.Width"), group = "Species", paired = F)
```

`correlazioni_LB` *This function computes the correlation coefficients and prints the pairs from the heightest coefficient*

Description

This function computes the correlation coefficients and prints the pairs from the heightest coefficient

Usage

```
correlazioni_LB(
  dataset,
  lista_vars,
  method = "spearman",
  rho_dec = 3,
  pval_dec = 4
)
```

Arguments

`dataset`	dataframe
`lista_vars`	vector of numeric variables to be computed the correlation
`method`	method to compute the correlation coefficient (Default = "spearman")
`rho_dec`	number of decimal for rho (Default = 3)
`pval_dec`	number of decimal for the pvalue (Default = 4)

Value

Una lista con dataset

filename_LB *This function returns the filename to be outputted*

Description

This function returns the filename to be outputted

Usage

```
filename_LB(
  filename = "Prova",
  extention = ".png",
  output = path_output,
  datetime = F
)
```

Arguments

filename	name of the file
extention	file extention
output	the main output path
datetime	whether to print the datetime in a cute format

formatz_p *Function to get a formatted p-value for a number o a vector of numbers*

Description

Function to get a formatted p-value for a number o a vector of numbers

Usage

```
formatz_p(value)
```

Arguments

value	a number or a vector of numbers to be formatted

Value

a number or a vector of numbers formatted with 4 digits

Examples

```
formatz_p(c(1.000, 0.75643242, 0.000032431, 0.00214))
```

Kmax_aim_LB — *Function to print the histogram of the AIM::cv.cox.main output*

Description

Function to print the histogram of the AIM::cv.cox.main output

Usage

```
Kmax_aim_LB(kmax.cycle = kmax.cycle)
```

Arguments

kmax.cycle	The vector of values of the best biomarkers

Value

an histogram

KM_LB — *This function allows to create a KM survival curve overall or splitted by a categorical variable*

Description

This function allows to create a KM survival curve overall or splitted by a categorical variable

Usage

```
KM_LB(
  Event = "OS_EVENT",
  tEvent = "OS",
  strata = 1,
  data = data,
  title = "Prova",
  xlab = "Time in months",
  ylab = "Probaility of Surv",
  xlim = c(0, max(data[, tEvent], na.rm = T)),
  breaks_by = 3
)
```

Arguments

Event	Event variable
tEvent	Survival Time Variable
strata	Variable to stratify (Default = 1)
data	dataframe
title	Graph title (Default = "Prova")

xlab	x-axis title (Default = "Time in months")
ylab	y-axis title (Default = "Probaility of Surv")
xlim	limits of x-axis (Default is from 0 to maximum observed time)
breaks_by	breaks of risk table(Default = 3)

Value

a KM graph

`Lineplots_LB` *Function to build the lineplots*

Description

Function to build the lineplots

Usage

```
Lineplots_LB(
  data,
  variables,
  time,
  group = 1,
  split = F,
  lw_reg = 1,
  size_point = 0.6,
  size_title = 12,
  col_title = 1,
  size_axis_x = 5,
  size_axis_y = 6,
  size_title_grid = 7,
  breaks = unique(data[, time]),
  label = unique(data[, time]),
  ylim = c(0.2, 0.8),
  Posthoc = F,
  Friedman = F,
  Test_results = Test_results,
  grid = T,
  ratio = 1,
  extra = F,
  extra_text = NULL,
  PPTX = F,
  pptx_width = 8.5,
  pptx_height = 5.5,
  threshold_posthoc = 0.1,
  check = F,
  target = paste0(path, "/file.pptx"),
  col_lines = c("salmon", "royalblue")
)
```

Arguments

data	dataset
variables	vector of variables
time	x-axis variable
group	factor variable to group
split	whether to split in two windows the lines
lw_reg	lwd of regression line
size_point	size of points
size_title	size of title
col_title	colour of title
size_axis_x	x-axis text size
size_axis_y	y-axis text size
size_title_grid	size of title in the grid
breaks	breaks of x-axis
label	labels of x-axis
ylim	ylim to display in the graph
Posthoc	whether to display posthoc tests
Friedman	friedman overall test dataset
Test_results	posthoc test dataset
grid	whether to build a grid or a pptx
ratio	graph ratio
extra	do you want to add extra option?
extra_text	write your additional options
PPTX	whether to build a pptx or a grid
pptx_width	inch
pptx_height	inch
threshold_posthoc	threshold to display posthoc brackets
check	check the correctness of your graph
target	where do you want your pptx to be saved
col_lines	splitted lines colour

LL_Descrittive — *Function to build, starting from a dataset, the descriptive statistics of every variable*

Description

Function to build, starting from a dataset, the descriptive statistics of every variable

Usage

```
LL_Descrittive(dataset, path = NULL)
```

Arguments

dataset	dataframe
path	where do you want it to be saved

LL_fisher_gt_flex — *Function to build coloumn marginal statistics and Fisher test*

Description

Function to build coloumn marginal statistics and Fisher test

Usage

```
LL_fisher_gt_flex(data, row_var, col_var, label_row_var, label_col_var)
```

Arguments

data	dataframe
row_var	row variable
col_var	column variable
label_row_var	label for row
label_col_var	label for column

LL_Npsurv_format *Function to get a cute format of npsurv output.*

Description

Function to get a cute format of npsurv output.

Usage

```
LL_Npsurv_format(fit.npsurv)
```

Arguments

fit.npsurv	A npsurv(Surv(time, event) ~ cov_factor, data) object

Value

A cute format of npsurv output

LL_Tapply_f *Function for an easy application of the tapply*

Description

Function for an easy application of the tapply

Usage

```
LL_Tapply_f(data, var_quant, var_cat, digits = 2)
```

Arguments

data	dataframe
var_quant	quantitative variable
var_cat	categorial variable
digits	digits to display

multivariate_LL *Function to create a multivariate cph model with a vector of variables*

Description

Function to create a multivariate cph model with a vector of variables

Usage

```
multivariate_LL(db, vars, ptime, pevent, dec_HR = 4)
```

Arguments

db	A dataframe
vars	Vector of variables to be included in the multivariate model
ptime	Survival Time variable
pevent	Event variable
dec_HR	digits of HR (Default = 4)

Value

the multivariate model

New_Project_LB *Function to create a new project in the default folder*

Description

Function to create a new project in the default folder

Usage

```
New_Project_LB(name_project, pc = c("Luca", "Stefano"))
```

Arguments

name_project	The name of the Project
pc	Which pc are we operating

Value

Returns a folder in Projects with Analisi, Dati and Output subfolders

`output.aim.f` *Function to print the output of the AIM function with Biomarker, Direction and Cutoff as a data frame model*

Description

Function to print the output of the AIM function with Biomarker, Direction and Cutoff as a data frame model

Usage

```
output.aim.f(res.index, aim.data)
```

Arguments

`res.index`	An output from the AIM package function
`aim.data`	Data where the function was run on

Value

A dataframe-like object

`PDF_print_LB` *Function to print the PDF with the grid.arrange function*

Description

Function to print the PDF with the grid.arrange function

Usage

```
PDF_print_LB(
  plot_list,
  path_print = path_print,
  nrow = 8,
  ncol = 6,
  variables = vett_all_markers
)
```

Arguments

`plot_list`	The list you want to be plotted
`path_print`	The path where you want your PDF to be printed
`nrow`	Rows of your grid
`ncol`	Columns of your grid
`variables`	Number of total graphs to be printed

Value

A pdf in the path_output

`Stringa_LL` *Funzione che riceve in input le posizioni dei nomi di un dataframe e crea una stringa di tali nomi separati da virgola o da altro segno/simbolo*

Description

Funzione che riceve in input le posizioni dei nomi di un dataframe e crea una stringa di tali nomi separati da virgola o da altro segno/simbolo

Usage

```
Stringa_LL(data, vet, sep = ",")
```

Arguments

`data`	datafrane
`vet`	vector of positions for names in the dataset
`sep`	symbol to separate names from each other

`Sys_Time_LB` *Function to get the Sys.time() in a cute and nice format*

Description

Function to get the Sys.time() in a cute and nice format

Usage

```
Sys_Time_LB()
```

Value

The Sys.time() in a cute format

`telegram_mess_LB` *Function to send a Telegram message with BiostatUO9 bot. NB: must create a start_time before running it*

Description

Function to send a Telegram message with BiostatUO9 bot. NB: must create a start_time before running it

Usage

```
telegram_mess_LB(
  process_time = {

   format(lubridate::seconds_to_period(round(as.numeric(difftime(Sys.time(), start_time,
    units = "secs")))), "%H:%M:%S")
 },
  dest = "both",
  script = 0,
  rm_start_time = T
)
```

Arguments

`process_time`	Just don't modify it
`dest`	Who is going to receive the message
`script`	The title of the message
`rm_start_time`	If you want the start_time item to be removed after the message is sent

Value

Nothing

`univariate_LL` *This function allows you to create the univariate regression model for a vector of variables*

Description

This function allows you to create the univariate regression model for a vector of variables

Usage

```
univariate_LL(db, vars, ptime, pevent, dec_HR = 4)
```

Arguments

db	dataframe
vars	vector with variables name
ptime	Survival Time variable
pevent	Event variable
dec_HR	digits of HR (Default = 4)

Value

a dataframe with all univariate models

---

vett.quoted *Funzione che permette partendo da un vettore, di riscrivere quel vettore in varie forme*

Description

Funzione che permette partendo da un vettore, di riscrivere quel vettore in varie forme

Usage

```
vett.quoted(vettore, sym = ", ", quote = T)
```

Arguments

vettore	Starting vector
sym	Symbol of separation (Default ", ")
quote	Vector elements to be quoted or not (Default = T)

Value

Una stringa di elementi formattati al meglio