Package ‘reportRmd’

October 17, 2022

Title  Tidy Presentation of Clinical Reporting
Version  0.0.1
Description  Streamlined statistical reporting in 'Rmarkdown' environments.
Facilitates the automated reporting of descriptive statistics, multiple
univariate models, multivariable models and tables combining these outputs.
Plotting functions include customisable survival curves, forest plots from
logistic and ordinal regression and bivariate comparison plots.
License  MIT + file LICENSE
Suggests  rmarkdown, testthat (>= 3.0.0)
Config/testthat/edition  3
Encoding  UTF-8
RoxygenNote  7.2.1
Imports  aod, cmprsk, geepack, ggplot2, ggpubr, gridExtra, kableExtra,
        knitr, MASS, pander, plyr, rlang, scales, survival
Collate  'helper.R' 'main.R' 'globals.R' 'data.R'
Depends  R (>= 2.10)
LazyData  true
VignetteBuilder  knitr
NeedsCompilation  no
Author  Lisa Avery [cre, aut],
        Ryan Del Bel [aut],
        Osvaldo Espin-Garcia [aut],
        Tyler Pittman [aut] (0000-0002-5013-6980),
        Yanning Wang [ctr],
        Jessica Weiss [aut],
        Wei Xu [aut]
Maintainer  Lisa Avery <lisa.avery@uhn.ca>
Repository  CRAN
Date/Publication  2022-10-17 06:10:02 UTC
R topics documented:

addspace ................................................................. 3
boxcoxfitRx ............................................................. 3
cap ................................................................. 4
covsum ................................................................. 4
crrRx ................................................................. 6
ctDNA ................................................................. 7
eancelCol ............................................................... 7
forestplot2 ............................................................ 8
formatp ................................................................. 9
geoR_boxcoxfit ....................................................... 9
ggkmcf ................................................................. 10
ggkmcf_paste .......................................................... 14
hbld ................................................................. 15
lbld ................................................................. 15
lpvalue ................................................................. 16
mvsum ................................................................. 16
nestTable .............................................................. 17
nicename .............................................................. 19
niceNum ............................................................... 19
outTable ............................................................... 20
pembrolizumab ......................................................... 21
plotuv ................................................................. 22
psthre ............................................................... 24
pstprn ............................................................... 24
pvalue ................................................................. 24
rmds ................................................................. 25
rm_covsum ............................................................ 25
rm_mvsum ............................................................ 28
rm_survdiff ........................................................... 30
rm_survsum ........................................................... 31
rm_survtime .......................................................... 33
rm_uvsum ............................................................ 35
rm_uv_mv .............................................................. 38
sanitizestr ............................................................. 39
testData .............................................................. 40
uvsum ................................................................. 40

Index 43
addspace

Description
Add spaces to strings in LaTeX. Returns appends \texttt{~~~} before the string

Usage
addspace(x)

Arguments
x string

boxcoxfitRx

fit box cox transformed linear model

Description
Wrapper function to fit fine and gray competing risk model using function crr from package cmprsk

Usage
boxcoxfitRx(f, data, lambda = FALSE)

Arguments
f formula for the model. Currently the formula only works by using the name of the column in a dataframe. It does not work by using $ or [] notation.
data dataframe containing data
lambda boolean indicating if you want to output the lambda used in the boxcox transformation. If so the function will return a list of length 2 with the model as the first element and a vector of length 2 as the second.

Value
a list containing the linear model (lm) object and, if requested, lambda
cap

**Capitalize a string**

**Description**
Calitalize a string

**Usage**
cap(x)

**Arguments**
x string

covsum

**Get covariate summary dataframe**

**Description**
Returns a dataframe corresponding to a descriptive table.

**Usage**
covsum(
  data,
  covs,
  maincov = NULL,
  digits = 1,
  numobs = NULL,
  markup = TRUE,
  sanitize = TRUE,
  nicenames = TRUE,
  IQR = FALSE,
  all.stats = FALSE,
  pvalue = TRUE,
  show.tests = FALSE,
  excludeLevels = NULL,
  full = TRUE,
  digits.cat = 0,
  testcont = c("rank-sum test", "ANOVA"),
  testcat = c("Chi-squared", "Fisher"),
  include_missing = FALSE,
  percentage = c("column", "row")
)

Arguments

data    dataframe containing data
covs    character vector with the names of columns to include in table
maincov    covariate to stratify table by
digits    number of digits for summarizing mean data, does not affect p-values
numobs    named list overriding the number of people you expect to have the covariate
markup    boolean indicating if you want latex markup
sanitize    boolean indicating if you want to sanitize all strings to not break LaTeX
nicenames    boolean indicating if you want to replace . and _ in strings with a space
IQR    boolean indicating if you want to display the inter quantile range (Q1,Q3) as opposed to (min,max) in the summary for continuous variables
all.stats    boolean indicating if all summary statistics (Q1,Q3 + min,max on a separate line) should be displayed. Overrides IQR.
pvalue    boolean indicating if you want p-values included in the table
show.tests    boolean indicating if the type of statistical used should be shown in a column beside the pvalues. Ignored if pvalue=FALSE.
excludeLevels    a named list of covariate levels to exclude from statistical tests in the form list(varname=c('level1','level2')). These levels will be excluded from association tests, but not the table. This can be useful for levels where there is a logical skip (ie not missing, but not presented). Ignored if pvalue=FALSE.
full    boolean indicating if you want the full sample included in the table, ignored if maincov is NULL
digits.cat    number of digits for the proportions when summarizing categorical data (default: 0)
testcont    test of choice for continuous variables, one of rank-sum (default) or ANOVA
testcat    test of choice for categorical variables, one of Chi-squared (default) or Fisher
include_missing    Option to include NA values of maincov. NAs will not be included in statistical tests
percentage    choice of how percentages are presented, one of column (default) or row

Details

Comparisons for categorical variables default to chi-square tests, but if there are counts of <5 then the Fisher Exact test will be used and if this is unsuccessful then a second attempt will be made computing p-values using MC simulation. If testcont='ANOVA' then the t-test with unequal variance will be used for two groups and an ANOVA will be used for three or more. The statistical test used can be displayed by specifying show.tests=TRUE.

The number of decimals places to display the statistics can be changed with digits, but this will not change the display of p-values. If more significant digits are required for p-values then use tableOnly=TRUE and format as desired.

See Also

fisher.test, chisq.test, wilcox.test, kruskal.test, and anova
Description

Wrapper function to fit fine and gray competing risk model using function crr from package cmprsk

Usage

crrRx(f, data)

Arguments

f   formula for the model. Currently the formula only works by using the name of the column in a dataframe. It does not work by using $ or [] notation.

data  dataframe containing data

Value

a competing risk model with the call appended to the list

See Also

crr

Examples

# From the crr help file:
set.seed(10)
ftime <- rexp(200)
fstatus <- sample(0:2,200,replace=TRUE)
cov <- matrix(runif(600),nrow=200)
dimnames(cov)[[2]] <- c('x1','x2','x3')
df <- data.frame(ftime,fstatus,cov)
m1 <- crrRx(as.formula('ftime+fstatus~x1+x2+x3'),df)
# Nicely output to report:
rm_mvsum(m1,data=df,showN = TRUE,vif=TRUE)
ctDNA

Tumour size change over time Longitudinal changes in tumour size since baseline for patients by changes in ctDNA status (clearance, decrease or increase) since baseline.

Description

Tumour size change over time
Longitudinal changes in tumour size since baseline for patients by changes in ctDNA status (clearance, decrease or increase) since baseline.

Usage

ctDNA

Format

A data frame with 270 rows and 5 variables:

- **id**: Patient ID
- **cohort**: Study Cohort: A = Squamous cell carcinoma of soft palate, B = Triple negative breast cancer, C = Ovarian, high grade serous, D = Melanoma, E = Other Solid Tumor
- **ctdna_status**: Change in ctDNA since baseline
- **time**: Number of weeks on treatment
- **size_change**: Percentage change in tumour measurement

Source

https://www.nature.com/articles/s43018-020-0096-5

---

excelCol

Retrieve columns number from spreadsheet columns specified as unquoted letters

Description

Retrieve columns number from spreadsheet columns specified as unquoted letters

Usage

excCol(...)

Arguments

... unquoted excel column headers (i.e. excCol(A,CG,AA)) separated by commas
Value

a numeric vector corresponding to columns in a spreadsheet

Examples

## Find the column numbers for excel columns AB, CE and BB
excelCol(AB, CE, bb)

forestplot2 Create a forest plot using ggplot2

Description

This function will accept a log or logistic regression fit from glm, and display the OR or RR for each variable on the appropriate log scale.

Usage

forestplot2(
    model,
    conf.level = 0.95,
    orderByRisk = TRUE,
    colours = "default",
    showEst = TRUE,
    rmRef = FALSE,
    logScale = TRUE,
    nxTicks = 5
)

Arguments

model an object output from the glm function, must be from a logistic regression
conf.level controls the width of the confidence interval
orderByRisk logical, should the plot be ordered by risk
colours can specify colours for risks less than, 1 and greater than 1.0. Default is red, black, green
showEst logical, should the risks be displayed on the plot in text
rmRef logical, should the reference levels be removed for the plot?
logScale logical, should OR/RR be shown on log scale, defaults to TRUE. See https://doi.org/10.1093/aje/kwr156 for why you may prefer a linear scale.
nxTicks Number of tick marks supplied to the log_breaks function to produce

Value

a plot object
### Examples

```r
glm_fit = glm(orr~change_cdtdna_group+sex+age+l_size, 
data=pembrolizumab,family = 'binomial')
forestplot2(glm_fit)
```

### `formatp`

**Specific p-value formatting**

- If \( p < 0.001 \) returns "<0.001", if \( p < 0.01 \) returns \( p \) to 3 decimal places otherwise returns \( p \) to 2 decimal places.

**Usage**

```r
formatp(pvalues)
```

**Arguments**

- `pvalues`: a vector of \( p \) values

### `geoR_boxcoxfit`

**Parameter Estimation for the Box-Cox Transformation**

This function is copied from the geoR package which has been removed from the CRAN repository.

**Usage**

```r
geoR_boxcoxfit(object, xmat, lambda, lambda2 = NULL, add.to.data = 0)
```

**Arguments**

- `object`: a vector with the data
- `xmat`: a matrix with covariates values. Defaults to `rep(1, length(y))`.
- `lambda`: numerical value(s) for the transformation parameter lambda. Used as the initial value in the function for parameter estimation. If not provided default values are assumed. If multiple values are passed the one with highest likelihood is used as initial value.
- `lambda2`: logical or numerical value(s) of the additional transformation (see DETAILS below). Defaults to `NULL`. If `TRUE` this parameter is also estimated and the initial value is set to the absolute value of the minimum data. A numerical value is provided it is used as the initial value. Multiple values are allowed as for `lambda`.
- `add.to.data`: a constant value to be added to the data.
Details

For more information see: https://cran.r-project.org/web/packages/geoR/index.html

---

**ggkmcif**

*Plot KM and CIF curves with ggplot*

---

**Description**

This function will plot a KM or CIF curve with option to add the number at risk. You can specify if you want confidence bands, the hazard ratio, and p-values, as well as the units of time used.

**Usage**

```r
ggkmcif(
  response,
  cov = NULL,
  data,
  type = NULL,
  pval = TRUE,
  HR = FALSE,
  HR_pval = FALSE,
  conf.curves = FALSE,
  conf.type = "log",
  table = TRUE,
  times = NULL,
  xlab = "Time",
  ylab = NULL,
  main = NULL,
  stratalabs = NULL,
  strataname = nicename(cov),
  stratalabs.table = NULL,
  strataname.table = strataname,
  median.text = FALSE,
  median.lines = FALSE,
  median.CI = FALSE,
  set.time.text = NULL,
  set.time.line = FALSE,
  set.time = 5,
  set.time.CI = FALSE,
  censor.marks = TRUE,
  censor.size = 3,
  censor.stroke = 1.5,
  fsize = 10,
  nsize = 3,
  lsize = 1,
  psize = 3.5,
)```
median.size = 3, median.pos = NULL, median.lsize = 1,  
set.size = 3, set.pos = NULL, set.lsize = 1,  
ylim = c(0, 1), col = NULL,  
linetype = NULL, xlim = NULL,  
legend.pos = NULL, pval.pos = NULL,  
plot.event = 1, event = c("col", "linetype"),  
flip.CIF = FALSE, cut = NULL,  
eventlabs = NULL, event.name = NULL,  
Numbers_at_risk_text = "Numbers at risk", HR.digits = 2,  
HR.pval.digits = 3, pval.digits = 3,  
median.digits = 3, set.time.digits = 3,  
returns = FALSE, print.n.missing = TRUE  
)

Arguments

response character vector with names of columns to use for response

cov String specifying the column name of stratification variable

data dataframe containing your data

type string indicating the type of univariate model to fit. The function will try and  
   guess what type you want based on your response. If you want to override this  
   you can manually specify the type. Options include "KM", and 'CIF'

pval boolean to specify if you want p-values in the plot (Log Rank test for KM and  
   Gray's test for CIF)

HR boolean to specify if you want hazard ratios included in the plot

HR_pval boolean to specify if you want HR p-values in the plot

conf.curves boolean to specify if you want confidence interval bands

conf.type One of "none"(the default), "plain", "log", "log-log" or "logit". Only enough of  
   the string to uniquely identify it is necessary. The first option causes confidence  
   intervals not to be generated. The second causes the standard intervals curve  
   + k *se(curve), where k is determined from conf.int. The log option calculates  
   intervals based on the cumulative hazard or log(survival). The log-log option
bases the intervals on the log hazard or log(-log(survival)), and the logit option on log(survival/(1-survival)).

```
table  Logical value. If TRUE, includes the number at risk table
times Numeric vector of times for the x-axis
xlab String corresponding to xlab. By default is "Time"
ylab String corresponding to ylab. When NULL uses "Survival probability" for KM cuves, and "Probability of an event" for CIF
main String corresponding to main title. When NULL uses Kaplan-Meier Plot s, and "Cumulative Incidence Plot for CIF"
stratalabs string corresponding to the labels of the covariate, when NULL will use the levels of the covariate
strataname String of the covariate name default is nicename(cov)
stratalabs.table String corresponding to the levels of the covariate for the number at risk table, when NULL will use the levels of the covariate. Can use a string of "." when the labels are long
strataname.table String of the covariate name for the number at risk table default is nicename(cov)
median.text boolean to specify if you want the median values added to the legend (or as added text if there are no covariates), for KM only
median.lines boolean to specify if you want the median values added as lines to the plot, for KM only
median.CI boolean to specify if you want the 95\% interval with the median text (Only for KM)
set.time.text string for the text to add survival at a specified time (eg. year OS)
set.time.line boolean to specify if you want the survival added as lines to the plot at a specified point
set.time Numeric values of the specific time of interest, default is 5 (Multiple values can be entered)
set.time.CI boolean to specify if you want the 95\% interval with the set time text
censor.marks logical value. If TRUE, includes censor marks (only for KM curves)
censor.size size of censor marks, default is 3
censor.stroke stroke of censor marks, default is 1.5
fsize font size
nsize font size for numbers in the numbers at risk table
lsize line size
psize size of the pvalue
median.size size of the median text (Only when there are no covariates)
median.pos vector of length 2 corresponding to the median position (Only when there are no covariates)
median.lsize line size of the median lines
ggkmcif

`set.size`  
size of the survival at a set time text (Only when there are no covariates)

`set.pos`  
vector of length 2 corresponding to the survival at a set point position (Only when there are no covariates)

`set.lsize`  
line size of the survival at set points

`ylim`  
vector of length 2 corresponding to limits of y-axis. Default to NULL

`col`  
vector of colours

`linetype`  
vector of line types

`xlim`  
vector of length 2 corresponding to limits of x-axis. Default to NULL

`legend.pos`  
Can be either a string corresponding to the legend position ("left", "top", "right", "bottom", "none") or a vector of length 2 corresponding to the legend position (uses normalized units (ie the c(0.5,0.5) is the middle of the plot))

`pval.pos`  
vector of length 2 corresponding to the p-value position

`plot.event`  
Which event(s) to plot (1, 2, or c(1,2))

`event`  
String specifying if the event should be mapped to the colour, or linetype when plotting both events to colour = "col", line type

`flip.CIF`  
boolean to flip the CIF curve to start at 1

`cut`  
numeric value indicating where to divide a continuous covariate (default is the median)

`eventlabs`  
String corresponding to the event type names

`event.name`  
String corresponding to the label of the event types

`Numbers_at_risk_text`  
String for the label of the number at risk

`HR.digits`  
Number of digits printed of the hazard ratio

`HR.pval.digits`  
Number of digits printed of the hazard ratio pvalue

`pval.digits`  
Number of digits printed of the Gray's/log rank pvalue

`median.digits`  
Number of digits printed of the median pvalue

`set.time.digits`  
Number of digits printed of the probability at a specified time

`returns`  
Logical value returns a list with all ggplot objects in a list

`print.n.missing`  
Logical, should the number of missing be shown! Needs to be checked

Details

Note that for proper pdf output of special characters the following code needs to be included in the first chunk of the rmd knitr::opts_chunk$set(dev="cairo_pdf")

Value

Nothing is returned unless returns = TRUE is used. With returns = TRUE, if table = TRUE (the default) a table style graphic with survival plot and number at risk table is returned. Otherwise a plot with the survival curves is returned.
Examples

# Simple plot without confidence intervals
ggkm cif(response = c('os_time', 'os_status'),
         cov = 'cohort',
         data = pembrolizumab)

# Plot with median survival time
ggkm cif(response = c('os_time', 'os_status'),
         cov = 'sex',
         data = pembrolizumab,
         median.text = TRUE, median.lines = TRUE, conf.curves = TRUE)

# Plot with specified survival times and log-log CI
ggkm cif(response = c('os_time', 'os_status'),
         cov = 'sex',
         data = pembrolizumab,
         median.text = FALSE, set.time.text = 'mo OS',
         set.time = c(12, 24), conf.type = 'log-log', conf.curves = TRUE)

# KM plot with 95% CI and censor marks
ggkm cif(c('os_time', 'os_status'), 'sex', data = pembrolizumab, type = 'KM',
         HR = TRUE, HR_pval = TRUE, conf.curves = TRUE, conf.type = 'log-log',
         set.time.CI = TRUE, censor.marks = TRUE)

---

ggkm cif_paste

Plot KM and CIF curves with ggplot

Description

This function puts together a survival curve, and a number at risk table

Usage

ggkm cif_paste(list_gg)

Arguments

list_gg list containing the results of ggkm cif

Value

a gtable with three elements, the survival curve, a spacer and the number at risk table

Examples

plot <- ggkm cif(response = c('pfs_time', 'pfs_status'),
                 data = pembrolizumab, returns = TRUE)

# Highlighting a section of the curve
plot[[1]] <- plot[[1]] +
  ggplot2::geom_rect(xmin=4,xmax=8,ymin=0.15,ymax=0.4,alpha=0.01,fill="yellow")

# Putting the curve back together
ggkmcif_paste(plot)

---

### hbld

**Bold strings in HTML**

**Description**

Bold strings in HTML.

**Usage**

hbld(strings)

**Arguments**

- **strings**: A vector of strings to bold.

---

### lbld

**Bold strings in LaTeX**

**Description**

Bold strings in LaTeX.

**Usage**

lbld(strings)

**Arguments**

- **strings**: A vector of strings to bold.
### lpvalue

**Formats p-values for LaTeX**

**Description**

Returns <0.001 if pvalue is <0.001. Else rounds the pvalue to specified significant digits. Will bold the p-value if it is <= 0.05

**Usage**

lpvalue(x, sigdigits = 2)

**Arguments**

- **x**: an integer
- **sigdigits**: number of significant digit to report

### mvsum

**Get multivariate summary dataframe**

**Description**

Returns a dataframe with the model summary and global p-value for multi-level variables.

**Usage**

mvsum(
  model,
  data,
  digits = 2,
  showN = FALSE,
  markup = TRUE,
  sanitize = TRUE,
  nicenames = TRUE,
  CIwidth = 0.95,
  vif = TRUE
)

**Arguments**

- **model**: fitted model object
- **data**: dataframe containing data
- **digits**: number of digits to round to
- **showN**: boolean indicating sample sizes should be shown for each comparison, can be useful for interactions
nestTable

markup boolean indicating if you want latex markup
sanitize boolean indicating if you want to sanitize all strings to not break LaTeX
nicenames boolean indicating if you want to replace . and _ in strings with a space.
CIwidth width for confidence intervals, defaults to 0.95
vif boolean indicating if the variance inflation factor should be included. See details

Details

Global p-values are likelihood ratio tests for lm, glm and polr models. For lme models an attempt
is made to re-fit the model using ML and if successful LRT is used to obtain a global p-value. For
coxph models the model is re-run without robust variances with and without each variable and a
LRT is presented. If unsuccessful a Wald p-value is returned. For GEE and CRR models Wald
global p-values are returned.

If the variance inflation factor is requested (VIF=T) then a generalised VIF will be calculated in the
same manner as the car package.

VIF for competing risk models is computed by fitting a linear model with a dependent variable
comprised of the sum of the model independent variables and then calculating VIF from this linear
model.

References


nestTable

Combine two table columns into a single column with levels of one
nested within levels of the other.

Description

This function accepts a data frame (via the data argument) and combines two columns into a single
column with values from the head_col serving as headers and values of the to_col displayed under-
neath each header. The resulting table is then passed to outTable for printing and output, to use the
grouped table as a data frame specify tableOnly=TRUE. By default the headers will be bolded and
the remaining values indented.

Usage

nestTable(
data,
head_col,
to_col,
colHeader = "",


caption = NULL,
indent = TRUE,
boldheaders = TRUE,
hdr_prefix = "",
hdr_suffix = "",
digits = 2,
tableOnly = FALSE
)

Arguments
data dataframe
head_col character value specifying the column name with the headers
to_col character value specifying the column name to add the headers into
colHeader character with the desired name of the first column. The default is to leave this empty for output or, for table only output to use the column name 'col1'.
caption table caption
indent Boolean should the original values in the to_col be indented
boldheaders Boolean should the header column values be bolded
hdr_prefix character value that will prefix headers
hdr_suffix character value that will suffix headers
digits number of digits to round numeric columns to, either a single number or a vector corresponding to the number of numeric columns
tableOnly boolean indicating if the table should be formatted for printing or returned as a data frame

Details
Note that it is possible to combine multiple tables (more than two) with this function.

Value
A character vector of the table source code, unless tableOnly=TRUE in which case a data frame is returned

Examples
## Investigate models to predict baseline ctDNA and tumour size and display together
## (not clinically useful!)
fit1 <- lm(baseline_cdtal~age+l_size+pdl1,data=pembrolizumab)
m1 <- rm_mvsum(fit1,tableOnly=TRUE)
m1$Response = 'ctDNA'
fit2 <- lm(l_size~age+baseline_cdtal+pdl1,data=pembrolizumab)
m2 <- rm_mvsum(fit2,tableOnly=TRUE)
m2$Response = 'Tumour Size'
rbind(m1,m2)
nestTable(rbind(m1,m2),head_col='Response',to_col='Covariate')
nicename

Lean strings for printing

Description

Returns strings with . and _ replaced by a space. This is nice when printing column names of your dataframe in a report.

Usage

nicename(strings, check_numbers = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strings</td>
<td>vector of strings to give a nice name</td>
</tr>
<tr>
<td>check_numbers</td>
<td>boolean indicating if numbers with decimals should be checked for and retained.</td>
</tr>
</tbody>
</table>

niceNum

Round retaining digits

Description

Round retaining digits

Usage

niceNum(x, digits = 2)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>a vector</td>
</tr>
<tr>
<td>digits</td>
<td>numeric</td>
</tr>
</tbody>
</table>
outTable  

Print tables to PDF/Latex HTML or Word

Description

Output the table nicely to whatever format is appropriate. This is the output function used by the `rm_*` printing functions.

Usage

```r
outTable(
  tab,
  row.names = NULL,
  to_indent = numeric(0),
  bold_headers = TRUE,
  rows_bold = numeric(0),
  bold_cells = NULL,
  caption = NULL,
  digits,
  align,
  applyAttributes = TRUE,
  keep.rownames = FALSE,
  fontsize,
  chunk_label
)
```

Arguments

- **tab**: a table to format
- **row.names**: a string specifying the column name to assign to the rownames. If NULL (the default) then rownames are removed.
- **to_indent**: numeric vector indicating which rows to indent in the first column.
- **bold_headers**: boolean indicating if the column headers should be bolded
- **rows_bold**: numeric vector indicating which rows to bold
- **bold_cells**: array indices indicating which cells to bold. These will be in addition to rows bolded by `rows_bold`.
- **caption**: table caption
- **digits**: number of digits to round numeric columns to, either a single number or a vector corresponding to the number of numeric columns in `tab`
- **align**: string specifying column alignment, defaults to left alignment of the first column and right alignment of all other columns. The align argument accepts a single string with 'l' for left, 'c' for centre and 'r' for right, with no separations. For example, to set the left column to be centred, the middle column right-aligned and the right column left aligned use: `align='crl'`
applyAttributes

boolean indicating if the function should use to_indent and bold_cells formatting attributes. This will only work properly if the dimensions of the table output from rm_covsum, rm_uvsum etc haven’t changed.

keep.rownames

should the row names be included in the output

fontsize

PDF/HTML output only, manually set the table fontsize

chunk_label

only used knitting to Word docs to allow cross-referencing

Details

Entire rows can be bolded, or specific cells. Currently indentation refers to the first column only. By default, underscores in column names are converted to spaces. To disable this set rm_ to FALSE

Value

A character vector of the table source code, unless tableOnly=TRUE in which case a data frame is returned

Examples

# To make custom changes or change the fontsize in PDF/HTML
tab <- rm_covsum(data=pembrolizumab,maincov = 'change_ctdna_group',
covs=c('age','sex','pdl1','tmb','l_size'),show.tests=TRUE,tableOnly = TRUE)
outTable(tab, fontsize=7)

# To bold columns with the variable names
rows_bold <- c(1,4,7,10,13)
outTable(tab,rows_bold = rows_bold)

# To bold the estimates for male/female
bold_cells <- as.matrix(expand.grid(5:6,1:ncol(tab)))
outTable(tab,bold_cells= bold_cells)

pembrolizumab

Survival data Survival status and ctDNA levels for patients receiving pembrolizumab

Description

Survival data

Survival status and ctDNA levels for patients receiving pembrolizumab

Usage

cembrolizumab
Format

A data frame with 94 rows and 15 variables:

- **id**: Patient ID
- **age**: Age at study entry
- **sex**: Patient Sex
- **cohort**: Study Cohort: A = Squamous cell carcinoma of soft palate, B = Triple negative breast cancer, C = Ovarian, high grade serous, D = Melanoma, E = Other Solid Tumor
- **l_size**: Target lesion size at baseline
- **pdl1**: PD L1 percent
- **tmb**: log of TMB
- **baseline_ctdna**: Baseline ctDNA
- **change_ctdna_group**: Did ctDNA increase or decrease from baseline to cycle 3
- **orr**: Objective Response
- **cbr**: Clinical Beneficial Response
- **os_status**: Overall survival status, 0 = alive, 1 = deceased
- **os_time**: Overall survival time in months
- **pfs_status**: Progression free survival status, 0 = progression free, 1 = progressed
- **pfs_time**: Progression free survival time in months

Source

https://www.nature.com/articles/s43018-020-0096-5

---

**plotuv**

*Plot multiple bivariate relationships in a single plot*

Description

This function is designed to accompany *uvsum* as a means of visualising the results, and uses similar syntax.

Usage

```r
plotuv(
  response, 
  covs, 
  data, 
  showN = FALSE, 
  showPoints = TRUE, 
  na.rm = TRUE, 
  response_title = NULL, 
  return_plotlist = FALSE, 
  ncol = 2, 
  pMargins = c(0, 0.2, 1, 0.2)
)
```
plotuv

Arguments

- **response**: character vector with names of columns to use for response
- **covs**: character vector with names of columns to use for covariates
- **data**: dataframe containing your data
- **showN**: boolean indicating whether sample sizes should be shown on the plots
- **showPoints**: boolean indicating whether individual data points should be shown when n>20 in a category
- **na.rm**: boolean indicating whether na values should be shown or removed
- **response_title**: character value with title of the plot
- **return_plotlist**: boolean indicating that the list of plots should be returned instead of a plot, useful for applying changes to the plot, see details
- **ncol**: the number of columns of plots to be display in the ggarrange call, defaults to 2
- **p_margins**: sets the TRBL margins of the individual plots, defaults to c(0,0.2,1,.2)

Details

Plots are displayed as follows: If response is continuous
- For a numeric predictor scatterplot
- For a categorical predictor: If 20+ observations available boxplot, otherwise dotplot with median line
If response is a factor
- For a numeric predictor: If 20+ observations available boxplot, otherwise dotplot with median line
- For a categorical predictor barplot
Response variables are shown on the ordinate (y-axis) and covariates on the abscissa (x-axis)

Value

- a list containing plots for each variable in covs
- a plot object

See Also

- ggplot and ggarrange

Examples

```r
## Run multiple univariate analyses on the pembrolizumab dataset to predict cbr and
## then visualise the relationships.
rm_uvsum(data=pembrolizumab,
response='cbr',covs=c('age','sex','l_size','baseline_ctdna'))
plotuv(data=pembrolizumab, response='cbr',
covs=c('age','sex','l_size','baseline_ctdna'),showN=TRUE)
```
pstmt

**Description**
Round and paste with parentheses

**Usage**
pstmt(x, y = 2)

**Arguments**
x: a numeric vector
y: integer corresponding to the number of digits to round by

pstprn

**Description**
Paste with parentheses

**Usage**
pstprn(x)

**Arguments**
x: a vector

pvalue

**Description**
Formats p-values

Returns <0.001 if pvalue is <0.001. Else rounds the pvalue to specified significant digits

**Usage**
pvalue(x, digits)

**Arguments**
x: an integer
digits: the number of significant digits to return
Description

Replace dollar signs with html for proper HTML output

Usage

rmds(s)

Arguments

s a character vector

Description

Outputs a descriptive covariate table

Usage

rm_covsum(
data, 
covs, 
maincov = NULL, 
caption = NULL, 
tableOnly = FALSE, 
covTitle = "", 
digits = 1, 
digits.cat = 0, 
nicenames = TRUE, 
IQR = FALSE, 
all.stats = FALSE, 
pvalue = TRUE, 
unformattedp = FALSE, 
show.tests = FALSE, 
testcont = c("rank-sum test", "ANOVA"), 
testcat = c("Chi-squared", "Fisher"), 
full = TRUE, 
include_missing = FALSE, 
percentage = c("column", "row"), 
excludeLevels = NULL,
Arguments

data dataframe containing data
covs character vector with the names of columns to include in table
maincov covariate to stratify table by
caption character containing table caption (default is no caption)
tableOnly Logical, if TRUE then a dataframe is returned, otherwise a formatted printed object is returned (default).
covTitle character with the names of the covariate (predictor) column. The default is to leave this empty for output or, for table only output to use the column name 'Covariate'.
digits number of digits for summarizing mean data
digits.cat number of digits for the proportions when summarizing categorical data (default: 0)
nicenames boolean indicating if you want to replace . and _ in strings with a space
IQR boolean indicating if you want to display the inter quantile range (Q1,Q3) as opposed to (min,max) in the summary for continuous variables
all.stats boolean indicating if all summary statistics (Q1,Q3 + min,max on a separate line) should be displayed. Overrides IQR.
pvalue boolean indicating if you want p-values included in the table
unformattedp boolean indicating if you would like the p-value to be returned unformatted (ie not rounded or prefixed with ’<‘). Best used with tableOnly = T and outTable function. See examples.
show.tests boolean indicating if the type of statistical used should be shown in a column beside the pvalues. Ignored if pvalue=FALSE.
testcont test of choice for continuous variables, one of rank-sum (default) or ANOVA
testcat test of choice for categorical variables, one of Chi-squared (default) or Fisher
full boolean indicating if you want the full sample included in the table, ignored if maincov is NULL
include_missing Option to include NA values of maincov. NAs will not be included in statistical tests
percentage choice of how percentages are presented, one of column (default) or row
excludeLevels a named list of covariate levels to exclude from statistical tests in the form list(varname =c('level1','level2')). These levels will be excluded from association tests, but not the table. This can be useful for levels where there is a logical skip (ie not missing, but not presented). Ignored if pvalue=FALSE.
rm_covsum

numobs  named list overriding the number of people you expect to have the covariate
markup  boolean indicating if you want latex markup
sanitize boolean indicating if you want to sanitize all strings to not break LaTeX
chunk_label only used if output is to Word to allow cross-referencing

Details

Comparisons for categorical variables default to chi-square tests, but if there are counts of <5 then the Fisher Exact test will be used and if this is unsuccessful then a second attempt will be made computing p-values using MC simulation. If testcont='ANOVA' then the t-test with unequal variance will be used for two groups and an ANOVA will be used for three or more. The statistical test used can be displayed by specifying show.tests=TRUE.

Further formatting options are available using tableOnly=TRUE and outputting the table with a call to outTable.

Value

A character vector of the table source code, unless tableOnly=TRUE in which case a data frame is returned

See Also

covsum, fisher.test, chisq.test, wilcox.test, kruskal.test, anova, and outTable

Examples

```r
rm_covsum(data=pembrolizumab, maincov = 'orr',
covs=c('age','sex','pdl1','tmb','l_size','change_ctdna_group'),
show.tests=TRUE)

# To make custom changes or change the fontsize in PDF/HTML
tab <- rm_covsum(data=pembrolizumab,maincov = 'change_ctdna_group',
covs=c('age','sex','pdl1','tmb','l_size'),show.tests=TRUE,tableOnly = TRUE)
outTable(tab, fontsize=7)

# To return unformatted p-values
tab <- rm_covsum(data=pembrolizumab, maincov = 'orr',
covs=c('age','sex','pdl1','tmb','l_size','change_ctdna_group'),
show.tests=TRUE,unformattedp=TRUE,tableOnly=TRUE)
outTable(tab,digits=5)
outTable(tab,digits=5, applyAttributes=FALSE) # remove bold/indent
```
Format a regression model nicely for 'Rmarkdown'

Description

Multivariate (or univariate) regression models are re-formatted for reporting and a global p-value is added for the evaluation of factor variables.

Usage

```r
rm_mvsum(
  model,
  data,
  digits = 2,
  covTitle = "",
  showN = FALSE,
  CIwidth = 0.95,
  vif = TRUE,
  caption = NULL,
  tableOnly = FALSE,
  p.adjust = "none",
  unformattedp = FALSE,
  chunk_label,
  markup = TRUE,
  sanitize = TRUE,
  nicenames = TRUE
)
```

Arguments

- `model`: model fit
- `data`: data that model was fit on (an attempt will be made to extract this from the model)
- `digits`: number of digits to round estimates to, does not affect p-values
- `covTitle`: character with the names of the covariate (predictor) column. The default is to leave this empty for output or, for table only output to use the column name ‘Covariate’.
- `showN`: boolean indicating sample sizes should be shown for each comparison, can be useful for interactions
- `CIwidth`: width for confidence intervals, defaults to 0.95
- `vif`: boolean indicating if the variance inflation factor should be included. See details
- `caption`: table caption
- `tableOnly`: boolean indicating if unformatted table should be returned
- `p.adjust`: p-adjustments to be performed (Global p-values only)
unformattedp: boolean indicating if you would like the p-value to be returned unformatted (i.e., not rounded or prefixed with '<'). Should be used in conjunction with the digits argument.

chunk_label: only used if output is to Word to allow cross-referencing

markup: boolean indicating if you want LaTeX markup

sanitize: boolean indicating if you want to sanitize all strings to not break LaTeX

nicenames: boolean indicating if you want to replace . and _ in strings with a space

Details

Global p-values are likelihood ratio tests for lm, glm and polr models. For lme models an attempt is made to re-fit the model using ML and if successful LRT is used to obtain a global p-value. For coxph models the model is re-run without robust variances with and without each variable and a LRT is presented. If unsuccessful a Wald p-value is returned. For GEE and CRR models Wald global p-values are returned.

If the variance inflation factor is requested (VIF=T) then a generalised VIF will be calculated in the same manner as the car package.

The number of decimals places to display the statistics can be changed with digits, but this will not change the display of p-values. If more significant digits are required for p-values then use tableOnly=TRUE and format as desired.

Value

A character vector of the table source code, unless tableOnly=TRUE in which case a data frame is returned.

References


Examples

```r
glm_fit = glm(change_ctdna_group~sex:age+baseline_ctdna+l_size, data=pembrolizumab,family = 'binomial')
rm_mvsum(glm_fit)
```

# linear model with p-value adjustment
```r
lm_fit=lm(baseline_ctdna~age+sex+l_size+tmb, data=pembrolizumab)
rm_mvsum(lm_fit,p.adjust = "bonferroni")
```

# Coxph
```r
require(survival)
res.cox <- coxph(Surv(os_time, os_status) ~ sex+age+l_size+tmb, data = pembrolizumab)
rm_mvsum(res.cox, vif=TRUE)
```
**Description**

This is a wrapper function around the survdiff function to display overall event rates and group-specific rates along with the log-rank test of a difference in survival between groups in a single table suitable for markdown output. Median survival times are included by default but can be removed setting median=FALSE

**Usage**

```rrm_survdiff(
  data,
  time,
  status,
  covs,
  strata,
  includeVarNames = FALSE,
  digits = 1,
  showCols = c("N", "Observed", "Expected"),
  CIwidth = 0.95,
  conf.type = "log",
  caption = NULL,
  tableOnly = FALSE
)
```

**Arguments**

- **data**: data frame containing survival data
- **time**: string indicating survival time variable
- **status**: string indicating event status variable
- **covs**: character vector indicating variables to group observations by
- **strata**: string indicating the variable to stratify observations by
- **includeVarNames**: boolean indicating if the variable names should be included in the output table, default is FALSE
- **digits**: the number of digits in the survival rate
- **showCols**: character vector indicating which of the optional columns to display, defaults to c("N","Observed","Expected")
- **CIwidth**: width of the median survival estimates, default is 95%
- **conf.type**: type of confidence interval see `survfit` for details. Default is 'log'.
- **caption**: table caption
- **tableOnly**: should a dataframe or a formatted object be returned
Value

A character vector of the survival table source code, unless tableOnly=TRUE in which case a data
frame is returned

See Also

survdiff

Examples

#' # Differences between sex
# rm_survdiff(data=pembrolizumab,time='os_time',status='os_status',
# covs='sex',digits=1)

# Differences between sex, stratified by cohort
rm_survdiff(data=pembrolizumab,time='os_time',status='os_status',
            covs='sex',strata='cohort',digits=1)

# Differences between sex/cohort groups
rm_survdiff(data=pembrolizumab,time='os_time',status='os_status',
            covs=c('sex','cohort'),digits=1)

Description

Displays event counts, median survival time and survival rates at specified times points for the entire
cohort and by group. The logrank test of differences in survival curves is displayed.

Usage

rm_survsum(
  data,
  time,
  status,
  group = NULL,
  survtimes = NULL,
  survtimeunit,
  survtimesLbls = NULL,
  CIwidth = 0.95,
  unformattedp = FALSE,
  conf.type = "log",
  na.action = "na.omit",
  showCounts = TRUE,
  digits = 2,
  caption = NULL,
  tableOnly = FALSE
)
Arguments

- **data**: data frame containing survival data
- **time**: string indicating survival time variable
- **status**: string indicating event status variable
- **group**: string or character vector indicating the variable(s) to group observations by. If this is left as NULL (the default) then summaries are provided for the entire cohort.
- **survtimes**: numeric vector specifying when survival probabilities should be calculated.
- **survtimeunit**: unit of time to suffix to the time column label if survival probabilities are requested, should be plural
- **survtimesLbls**: if supplied, a vector the same length as survtimes with descriptions (useful for displaying years with data provided in months)
- **CIwidth**: width of the survival probabilities, default is 95%
- **unformattedp**: boolean indicating if you would like the p-value to be returned unformatted (ie not rounded or prefixed with '<'). Should be used in conjunction with the digits argument.
- **conf.type**: type of confidence interval see `survfit` for details. Default is ‘log’.
- **na.action**: default is to omit missing values, but can be set to throw and error using na.action='na.fail'
- **showCounts**: boolean indicating if the at risk, events and censored columns should be output, default is TRUE
- **digits**: the number of digits in the survival rate, default is 2.
- **caption**: table caption for markdown output
- **tableOnly**: should a dataframe or a formatted object be returned

Details

This summary table is supplied for simple group comparisons only. To examine differences in groups with stratification see `rm_survdiff`. To summarise differences in survival rates controlling for covariates see `rm_survtime`.

Value

A character vector of the survival table source code, unless tableOnly=TRUE in which case a data frame is returned

See Also

- `survfit`

Examples

```r
# Simple median survival table
rm_survsum(data=pembrolizumab,time='Var0s_time',status='Var0s_status')

# Survival table with yearly survival rates
```
rm_survtime

rm_survsum(data=pembrolizumab, time='os_time', status='os_status', survtimes=c(12,24), survtimesLbls=1:2, survtimeunit='yr')

# Median survival by group
rm_survsum(data=pembrolizumab, time='os_time', status='os_status', group='sex')

# Survival Summary by cohort, displayed in years
rm_survsum(data=pembrolizumab, time='os_time', status='os_status', group='cohort', survtimes=seq(12,72,12), survtimesLbls=seq(1,6,1), survtimeunit='years')

# Survival Summary by Sex and ctDNA group
rm_survsum(data=pembrolizumab, time='os_time', status='os_status', group=c('sex', 'change_ctdna_group'), survtimes=c(12,24), survtimeunit='mo')
Arguments

data: data frame containing survival data

time: string indicating survival time variable

status: string indicating event status variable

covs: character vector with the names of variables to adjust for in coxph fit

strata: string indicating the variable to group observations by. If this is left as NULL (the default) then event counts and survival rates are provided for the entire cohort.

type: survival function, if no covs are specified defaults to Kaplan-Meier, otherwise the Cox PH model is fit. Use type='PH' to fit a Cox PH model with no covariates.

survtimes: numeric vector specifying when survival probabilities should be calculated.

survtimeunit: unit of time to suffix to the time column label if survival probabilities are requested, should be plural

strata.prefix: character value describing the grouping variable

survtimesLbls: if supplied, a vector the same length as survtimes with descriptions (useful for displaying years with data provided in months)

showCols: character vector specifying which of the optional columns to display, defaults to c('At Risk','Events','Censored')

CIwidth: width of the survival probabilities, default is 95%

conf.type: type of confidence interval see survfit for details. Default is 'log'.

na.action: default is to omit missing values, but can be set to throw and error using na.action='na.fail'

showCounts: boolean indicating if the at risk, events and censored columns should be output, default is TRUE

digits: the number of digits in the survival rate, default is 2.
caption: table caption for markdown output
tableOnly: should a dataframe or a formatted object be returned

Details

If covariates are supplied then a Cox proportional hazards model is fit for the entire cohort and each strata. Otherwise the default is for Kaplan-Meier estimates. Setting type = ‘PH’ will force a proportional hazards model.

Value

A character vector of the survival table source code, unless tableOnly=TRUE in which case a data frame is returned

See Also

survfit
Examples

# Kaplan-Meier survival probabilities with time displayed in years
rm_survtime(data=pembrolizumab,time='os_time',status='os_status',
strata="cohort",type='KM',survtimes=seq(12,72,12),
survtimesLbls=seq(1,6,1),
survtimeUnit='years')

# Cox Proportional Hazards survival probabilities
rm_survtime(data=pembrolizumab,time='os_time',status='os_status',
strata="cohort",type='PH',survtimes=seq(12,72,12),survtimesUnit='months')

# Cox Proportional Hazards survival probabilities controlling for age
rm_survtime(data=pembrolizumab,time='os_time',status='os_status',
covs='age',strata="cohort",survtimes=seq(12,72,12),survtimesUnit='months')

---

Output several univariate models nicely in a single table

Description

A table with the model parameters from running separate univariate models on each covariate. For factors with more than two levels a Global p-value is returned.

Usage

```r
rm_uvsum(
  response,
  covs,
  data,
  digits = 2,
  covTitle = "",
  caption = NULL,
  tableOnly = FALSE,
  removeInf = FALSE,
  p.adjust = "none",
  unformattedP = FALSE,
  chunk_label,
  gee = FALSE,
  id = NULL,
  corstr = NULL,
  family = NULL,
  type = NULL,
  strata = 1,
  nicenames = TRUE,
  showN = TRUE,
  CIWidth = 0.95,
  reflevel = NULL,
)```
returnModels = FALSE
)

Arguments

response
string vector with name of response
covs
character vector with the names of columns to fit univariate models to
data
dataframe containing data
digits
number of digits to round estimates and CI to. Does not affect p-values.
covTitle
character with the names of the covariate (predictor) column. The default is to
leave this empty for output or, for table only output to use the column name
' Covariate'.
caption
character containing table caption (default is no caption)
tableOnly
boolean indicating if unformatted table should be returned
removeInf
boolean indicating if infinite estimates should be removed from the table
p.adjust
p-adjustments to be performed (Global p-values only)
unformattedp
boolean indicating if you would like the p-value to be returned unformatted (ie
not rounded or prefixed with '<'). Should be used in conjunction with the digits
argument.
chunk_label
only used if output is to Word to allow cross-referencing
gge
boolean indicating if gee models should be fit to account for correlated observa-
tions. If TRUE then the id argument must specify the column in the data which
indicates the correlated clusters.
id
character vector which identifies clusters. Only used for geeglm
corstr
character string specifying the correlation structure. Only used for geeglm.
The following are permitted: "independence", "exchangeable", "ar1", "un-
structured" and "userdefined"
family
description of the error distribution and link function to be used in the model.
Only used for geeglm
type
string indicating the type of univariate model to fit. The function will try and
guess what type you want based on your response. If you want to override
this you can manually specify the type. Options include "linear", "logistic",
"poisson", "coxph", "crr", "boxcox", "ordinary", "geeglm"
strata
character vector of covariates to stratify by. Only used for coxph and crr
nicenames
boolean indicating if you want to replace . and _ in strings with a space
showN
boolean indicating if you want to show sample sizes
CIwidth
width of confidence interval, default is 0.95
reflevel
manual specification of the reference level. Only used for ordinal regression
This will allow you to see which model is not fitting if the function throws an
error
returnModels
boolean indicating if a list of fitted models should be returned. If this is TRUE
then the models will be returned, but the output will be suppressed. In addition
to the model elements a data element will be appended to each model so that the
fitted data can be examined, if necessary. See Details
Details

Global p-values are likelihood ratio tests for lm, glm and polr models. For lme models an attempt is made to re-fit the model using ML and if successful LRT is used to obtain a global p-value. For coxph models the model is re-run without robust variances with and without each variable and a LRT is presented. If unsuccessful a Wald p-value is returned. For GEE and CRR models Wald global p-values are returned.

The number of decimals places to display the statistics can be changed with digits, but this will not change the display of p-values. If more significant digits are required for p-values then use tableOnly=TRUE and format as desired.

Value

A character vector of the table source code, unless tableOnly=TRUE in which case a data frame is returned.

See Also

uvsum, lm, glm, crr, coxph, lme, geeglm, polr

Examples

# Examples are for demonstration and are not meaningful
# Coxph model with 90% CI
rm_uvsum(response = c('os_time','os_status'),
covs=c('age','sex','baseline_ctdna','l_size','change_ctdna_group'),
data=pembrolizumab,CIwidth=.9)

# Linear model with default 95% CI
rm_uvsum(response = 'baseline_ctdna',
covs=c('age','sex','l_size','pdl1','tmb'),
data=pembrolizumab)

# Logistic model with default 95% CI
rm_uvsum(response = 'os_status',
covs=c('age','sex','l_size','pdl1','tmb'),
data=pembrolizumab,family = binomial)
# Poisson models returned as model list
mList <- rm_uvsum(response = 'baseline_ctdna',
covs=c('age','sex','l_size','pdl1','tmb'),
data=pembrolizumab, returnModels=TRUE)
# mlist$data will expose the modelled data

# GEE on correlated outcomes
rm_uvsum(response = 'size_change',
covs=c('time','ctdna_status'),
gee=TRUE,
id='id', corstr="exchangeable",
family=gaussian("identity"),
data=ctDNA,showN=TRUE)
**Description**

This function will combine `rm_uvsum` and `rm_mvsum` outputs into a single table. The `tableOnly` argument must be set to `TRUE` when tables to be combined are created. The resulting table will be in the same order as the `uvsum` table and will contain the same columns as the `uvsum` and `mvsum` tables, but the p-values will be combined into a single column. There must be a variable overlapping between the `uvsum` and `mvsum` tables and all variables in the `mvsum` table must also appear in the `uvsum` table.

**Usage**

```r
rm_uv_mv(
  uvsumTable,
  mvsumTable,
  covTitle = "",
  vif = FALSE,
  caption = NULL,
  tableOnly = FALSE,
  chunk_label
)
```

**Arguments**

- `uvsumTable` Output from `rm_uvsum`, with `tableOnly`=TRUE
- `mvsumTable` Output from `rm_mvsum`, with `tableOnly`=TRUE
- `covTitle` character with the names of the covariate (predictor) column. The default is to leave this empty for output or, for table only output to use the column name 'Covariate'.
- `vif` boolean indicating if the variance inflation factor should be shown if present in the `mvsumTable`. Default is FALSE.
- `caption` table caption
- `tableOnly` boolean indicating if unformatted table should be returned
- `chunk_label` only used if output is to Word to allow cross-referencing

**Value**

A character vector of the table source code, unless `tableOnly`=TRUE in which case a data frame is returned

**See Also**

`rm_uvsum`, `rm_mvsum`
Examples

```
require(survival)

uvTab <- rm_uvsum(response = c('os_time','os_status'),
covs=c('age','sex','baseline_ctdna','l_size','change_ctdna_group'),
data=pembrolizumab,tableOnly=TRUE)

mv_surv_fit <- coxph(Surv(os_time,os_status)~age+sex+
baseline_ctdna+l_size+change_ctdna_group, data=pembrolizumab)

uvTab <- rm_mvsum(mv_surv_fit)

#linear model

uvtab<-rm_uvsum(response = 'baseline_ctdna',
covs=c('age','sex','l_size','pdl1','tmb'),
data=pembrolizumab,tableOnly=TRUE)

lm_fit=lm(baseline_ctdna~age+sex+l_size+tmb,data=pembrolizumab)

mvtab<-rm_mvsum(lm_fit,tableOnly = TRUE)

rm_uv_mv(uvtab,mvtab,tableOnly=TRUE)

#logistic model

uvtab<-rm_uvsum(response = 'os_status',
covs=c('age','sex','l_size','pdl1','tmb'),
data=pembrolizumab,family = binomial,tableOnly=TRUE)

logis_fit<-glm(os_status~age+sex+l_size+pdl1+tmb,data = pembrolizumab,family = 'binomial')

mvtab<-rm_mvsum(logis_fit,tableOnly = TRUE)

rm_uv_mv(uvtab,mvtab,tableOnly=TRUE)
```

---

**sanitizestr**

*Sanitizes strings to not break LaTeX*

**Description**

Strings with special characters will break LaTeX if returned 'asis' by knitr. This happens every time we use one of the main reportRx functions. We first sanitize our strings with this function to stop LaTeX from breaking.

**Usage**

```
sanitizestr(str)
```

**Arguments**

- **str** a vector of strings to sanitize
Funky ctDNA data

There is a weird factor with all one level, the cohort variable contains a cohort level (and Cohort A) and for one cohort all the size changes are missing.

### Usage

testData

### Format

A data frame with 270 rows and 6 variables:

- **id**: Patient ID
- **cohort**: Study Cohort: A = Squamous cell carcinoma of soft pallate, B = Triple negative breast cancer, C = Ovarian, high grade serous, D = Melanoma, E = Other Solid Tumor, cohort -for testing only
- **badfactor**: for testing
- **ctdna_status**: Change in ctDNA since baseline
- **time**: Number of weeks on treatment
- **size_change**: Percentage change in tumour measurement

### Source

[https://www.nature.com/articles/s43018-020-0096-5](https://www.nature.com/articles/s43018-020-0096-5)

---

**uvsum**

Get univariate summary dataframe

### Description

Returns a dataframe corresponding to a univariate regression table.
Usage

uvsum(
    response,
    covs,
    data,
    digits = 2,
    id = NULL,
    corstr = NULL,
    family = NULL,
    type = NULL,
    gee = FALSE,
    strata = 1,
    markup = TRUE,
    sanitize = TRUE,
    nicenames = TRUE,
    showN = TRUE,
    CIwidth = 0.95,
    reflevel = NULL,
    returnModels = FALSE
)

Arguments

response  string vector with name of response
covs      character vector with the names of columns to fit univariate models to
data      dataframe containing data
digits    number of digits to round to
id        character vector which identifies clusters. Used for GEE and coxph models.
corstr    character string specifying the correlation structure. Only used for geeglm. The following are permitted: "independence", "exchangeable", "ar1", "unstructured" and "userdefined"
family    specify details of the model used. This argument does not need to be specified and should be used with caution. By default, gaussian errors are used for linear models, the binomial family with logit link is used for logistic regression and poisson with log link is used for poisson regression. This can be specified with the type argument, or will be inferred from the data type. See family. Ignored for ordinal and survival regression and if the type argument is not explicitly specified.
type      string indicating the type of univariate model to fit. The function will try and guess what type you want based on your response. If you want to override this you can manually specify the type. Options include "linear", "logistic", "poisson", coxph", "crr", "boxcox" and "ordinal"
gee       boolean indicating if gee models should be fit to account for correlated observations. If TRUE then the id argument must specify the column in the data which indicates the correlated clusters.
strata    character vector of covariates to stratify by. Only used for coxph and crr
markdown boolean indicating if you want latex markup
sanitize boolean indicating if you want to sanitize all strings to not break LaTeX
nicenames boolean indicating if you want to replace . and _ in strings with a space
showN boolean indicating if you want to show sample sizes
CIwidth width of confidence interval, default is 0.95
reflevel manual specification of the reference level. Only used for ordinal. This may allow you to debug if the function throws an error.
returnModels boolean indicating if a list of fitted models should be returned.

Details
Univariate summaries for a number of covariates, the type of model can be specified. If unspecified the function will guess the appropriate model based on the response variable.
Confidence intervals are extracted using confint where possible. Otherwise Student t distribution is used for linear models and the Normal distribution is used for proportions.
returnModels can be used to return a list of the univariate models, which will be the same length as covs. The data used to run each model will include all cases with observations on the response and covariate. For gee models the data are re-ordered so that the ids appear sequentially and proper estimates are given.

See Also
lm, glm, crr, coxph, lme, geeglm, polr
Index

* dataframe
  - covsum, 4
  - mvsum, 16
  - rm_covsum, 25
  - uvsum, 40
  - crrRx, 6
  - ctDNA, 7
  - excelCol, 7
  - family, 41
  - fisher.test, 5, 27
  - forestplot2, 8
  - formatp, 9
  - geeglm, 37, 42
  - geoR_boxcoxfit, 9
  - ggarrange, 23
  - ggkmcif, 10
  - ggkmcif_paste, 14
  - ggplot, 23
  - glm, 37, 42
  - hblld, 15
  - kruskal.test, 5, 27
  - lbld, 15
  - lm, 37, 42
  - lme, 37, 42
  - lpvalue, 16
  - mvsun, 16
  - nestTable, 17
  - niceNum, 19
  - outTable, 20, 27
  - pembrolizumab, 21
  - plotuv, 22
  - polr, 37, 42
  - psthr, 24
  - pstprn, 24
  - pvalue, 24

* datasets
  - ctDNA, 7
  - pembrolizumab, 21
  - testData, 40

* helper
  - addspace, 3
  - cap, 4
  - formatp, 9
  - hblld, 15
  - lbld, 15
  - lpvalue, 16
  - niceNum, 19
  - niceName, 19
  - psthr, 24
  - pstprn, 24
  - pvalue, 24
  - rmds, 25
  - sanitizestr, 39

* model
  - boxcoxfitRx, 3
  - crrRx, 6
  - boxcoxfitRx, 3
  - crr, 6, 37, 42

* plot
  - forestplot2, 8
  - plotuv, 22

  - adm space, 3
  - anova, 5, 27
  - boxcoxfitRx, 3
  - cap, 4
  - chisq.test, 5, 27
  - covsum, 4, 27
  - coxph, 37, 42
  - crr, 6, 37, 42

43
rm_covsum, 25
rm_mvsum, 28, 38
rm_survdiff, 30, 32
rm_survsum, 31
rm_survtime, 32, 33
rm_uv_mv, 38
rm_uvsum, 35, 38
rmds, 25

sanitizestr, 39
survdiff, 31
survfit, 30, 32, 34

testData, 40

uvsum, 22, 37, 40

wilcox.test, 5, 27