Package ‘reportRx’

February 20, 2015

Title  Tools for automatically generating reproducible clinical report

Description  reportRx is a set of tools that integrates with LaTeX and knitr to automatically generate reproducible clinical reports. Functions to automatically produce descriptive tables, outcome summaries, univariate and multivariate analysis results and more are included.

Version  1.0

License  GPL-3

Author  Ryan Del Bel
        <rdelbel@gmail.com>,
        Wei Xu
        <wxu@uhnres.utoronto.ca>

Maintainer  Ryan Del Bel <rdelbel@gmail.com>

Imports  survival, xtable, aod, cmprsk, geoR, stringr,

Depends  reshape

Collate  'helper.R' 'main.R' 'misc.R'

NeedsCompilation  no

Repository  CRAN

Date/Publication  2013-12-17 23:18:16

R topics documented:

  addspace .......................................................... 2
  boxcoxfitRx .................................................. 2
  cap ............................................................... 3
  citime ............................................................ 3
  covsum .............................................................. 4
  crrRx ............................................................... 4
  etsum ............................................................... 5
  forestplot .......................................................... 5
  lbld ................................................................. 6
  lpvalue ............................................................. 6
addspace  

Add spaces to strings in LaTeX

Description

Add spaces to strings in LaTeX. Returns appends ~~~ 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 = F)
Arguments

\texttt{f} \quad \text{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 $\text{	extdollar}$ or $\text{[]}$ notation.}

\texttt{data} \quad \text{dataframe containing data}

\texttt{lambda} \quad \text{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.}

\textbf{Description}

\textit{Capitalize a string}

\textbf{Usage}

\texttt{cap(x)}

\textbf{Arguments}

\texttt{x} \quad \text{string}

\textbf{citime} \quad \textit{Get CI confidence interval}

\textbf{Description}

Returns the confidence interval of a CI at a specified time. Currently not very powerful. Only works on single strata.

\textbf{Usage}

\texttt{citime(data, response, times, units = "Years", outcomes = NULL, decimals = 2)}

\textbf{Arguments}

\texttt{data} \quad \text{dataframe containing data}

\texttt{response} \quad \text{character vector of response}

\texttt{times} \quad \text{numeric vector specifying single time to get CI for}

\texttt{units} \quad \text{string specifying the unit of times}

\texttt{outcomes} \quad \text{character vector specifying names of competing outcomes. Leave NULL if there is only one outcome}

\texttt{decimals} \quad \text{positive integer corresponding to the number of decimals}
covsum

Get covariate summary dataframe

Description

Returns a dataframe corresponding to a descriptive table

Usage

covsum(data, covs, maincov = NULL, numobs = NULL, markup = T, sanitize = T, nicenames = T)

Arguments

data dataframe containing data
covs character vector with the names of columns to include in table
maincov covariate to stratify table by
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

crrRx

fit crr model

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
etsum

Get event time summary dataframe

Description
This function will output a dataframe with useful summary statistics from a coxph model.

Usage
etsum(data, response, group = 1, times = c(12, 24))

Arguments
- **data**: dataframe containing data
- **response**: character vector with names of columns to use for response
- **group**: string specifying the column name of stratification variable
- **times**: numeric vector of times you want survival time probabilities for.

Examples
```r
require(survival)
lung$sex<-factor(lung$sex)
etsum(lung,c("time","status"),"sex")
etsum(lung,c("time","status"))
etsum(lung,c("time","status"),"sex",c(1,2,3))
```

forestplot

Create a forest plot

Description
Create a forest plot. All entries with cutoff=T will be plotted with an NA rather than their original value.

Usage
```r
forestplot(data, xlab = NULL, ylab = NULL, main = NULL, space = 0, bool = F, xlim = NULL)
```
Arguments

data  dataframe containing data
xlab  String corresponding to xlabel. By default is set to names(data)[2]
ylab  String corresponding to ylabel. By default is set to names(data)[1]
main  String corresponding to main title. By default is set to "Forest plot for subgroup analysis"
space  numeric corresponding to offset of y label. Should be positive if y label is on top of the names of the y axis
bool  A boolean vector. All entries with T will be invisible in the plot
xlim  vector of length 2 corresponding to limits of x-axis. Default to NULL.

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 2 significant digits. Will bold the p-value if it is <= 0.05

Usage

lpvalue(x)

Arguments

x  an integer
makedocx

Convert .TeX to .docx

Description

Converts the knitr-compiled .TeX file to a .docx file

Usage

makedocx(dir, fname, pdwd, imwd = "")

Arguments

dir full path of .TeX file directory
fname .TeX file file name. Do not include extension
pdwd full path to pandoc
imwd full path to image magick. Only include if there is at least one graphic.

mvsum

Get multivariate summary dataframe

Description

Returns a dataframe corresponding to a univariate table

Usage

mvsum(model, data, markup = T, sanitize = T, nicenames = T)

Arguments

model fitted model object
data dataframe containing data
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
### nicename

**Clean 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**

```r
nicename(strings)
```

**Arguments**

- **strings**
  - vector of strings to give a nice name

### pcovsum

**Print covariate summary Latex**

**Description**

Returns a dataframe corresponding to a descriptive table.

**Usage**

```r
pcovsum(data, covs, maincov = NULL, numobs = NULL, 
         Tex = F)
```

**Arguments**

- **data**
  - dataframe containing data
- **covs**
  - character vector with the names of columns to include in table
- **maincov**
  - covariate to stratify table by
- **numobs**
  - named list overriding the number of people you expect to have the covariate
- **Tex**
  - boolean indicating if you want to be able to view extra long tables in the LaTeX pdf. If TeX is T then the table will not convert properly to docx
petsum

**Print LaTeX event time summary**

**Description**

Wrapper for the etsum function that prints paragraphs of text in LaTeX

**Usage**

```r
petsum(data, response, group = 1, times = c(12, 14),
       units = "months")
```

**Arguments**

- `data` : dataframe containing data
- `response` : character vector with names of columns to use for response
- `group` : string specifying the column name of stratification variable
- `times` : numeric vector of times you want survival time probabilities for.
- `units` : string indicating the unit of time. Use lower case and plural.

**Examples**

```r
require(survival)
lung$sex<-factor(lung$sex)
petsum(lung,c("time","status"),"sex")
petsum(lung,c("time","status"))
petsum(lung,c("time","status"),"sex",c(1,2,3),"months")
```

---

plotci

**Plot CI curve**

**Description**

Plots a CI curve. Currently not very powerful. Only plots a single curve

**Usage**

```r
plotci(data, response, group = NULL, units = "months",
       main = "Viral Infections", CI = F, legpos = "topleft",
       xlim = NULL, outcomes = NULL)
```
**Arguments**

- **data**: dataframe containing data
- **response**: character vector or list of character vector. If a list it plot the '1' event for all outcomes on the same plot
- **group**: string of the group want to stratify by
- **units**: units of time
- **main**: String corresponding to title
- **CI**: Bool If True will plot CI and only the '1' event. if F will plot all events except for the final one
- **legpos**: string indicating which position to put legend choices are "topright" etc
- **xlim**: numeric vector corresponding to xlimits. Default is NULL
- **outcomes**: character vector of the names of the different competing outcomes

---

**plotkm**

*Plot KM curve*

---

**Description**

This function will plot a KM curve with possible stratification. You can specify if you want a legend or confidence bands as well as the units of time used.

**Usage**

```r
plotkm(data, response, group = 1, pos = "bottomleft", units = "months", CI = F, legend = T, title = "")
```

**Arguments**

- **data**: dataframe containing your data
- **response**: character vector with names of columns to use for response
- **group**: string specifying the column name of stratification variable
- **pos**: what position you want the legend to be. Current option are bottomleft and topright
- **units**: string specifying what the unit of time is use lower case and plural
- **CI**: boolean to specify if you want confidence intervals
- **legend**: boolean to specify if you want a legend
- **title**: title of plot

**Examples**

```r
require(survival)
lung$sex<-factor(lung$sex)
plotkm(lung,c("time","status"))
plotkm(lung,c("time","status"),"sex")
```
pmvsum

Print multivariate summary LaTeX table

Description

Returns a LaTeX table of the multivariate summary.

Usage

pmvsum(model, data)

Arguments

model
fitted model object
data
dataframe containing data

psthr

Round and paste with parentheses

Description

Round and paste with parentheses

Usage

psthr(x, y = 2)

Arguments

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

Examples

psthr(c(1.111111, 2.222222, 3.333333))
**pstprn**

*Paste with parentheses*

**Description**

Paste with parentheses

**Usage**

```r
pstprn(x)
```

**Arguments**

|x| a vector |

**Examples**

- `pstprn(c(1,2,3,4,5))`
- `pstprn(c("Hello","Hi",2))`

---

**puvsum**

*Print univariate summary LaTeX table*

**Description**

Returns a LaTeX table of the univariate summary

**Usage**

```r
puvsum(response, covs, data, type = NULL, strata = 1, Tex = T)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>response</code></td>
<td>string vector with name of response</td>
</tr>
<tr>
<td><code>covs</code></td>
<td>character vector with the names of columns to fit univariate models to</td>
</tr>
<tr>
<td><code>data</code></td>
<td>dataframe containing data</td>
</tr>
<tr>
<td><code>type</code></td>
<td>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 in clude &quot;linear&quot;, &quot;logistic&quot;, &quot;coxph&quot;, &quot;crr&quot;, &quot;boxcox&quot;,&quot;logistic&quot;</td>
</tr>
<tr>
<td><code>strata</code></td>
<td>character vector of covariates to stratify by. Only used for coxph and crr</td>
</tr>
<tr>
<td><code>TeX</code></td>
<td>boolean indicating if you want to be able to view extra long tables in the LaTeX pdf. If TeX is T then the table will not convert properly to docx</td>
</tr>
</tbody>
</table>
### pvalue

*Formats p-values*

#### Description

Returns $< 0.001$ if pvalue is $< 0.001$. Else rounds the pvalue to 2 significant digits

#### Usage

```r
pvalue(x)
```

#### Arguments

- `x`: an integer

### 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

```r
sanitizestr(str)
```

#### Arguments

- `str`: a vector of strings to sanitize
uvsum  

Get univariate summary dataframe

Description

Returns a dataframe corresponding to a univariate table

Usage

uvsum(response, covs, data, type = NULL, strata = 1,
       markup = T, sanitize = T, nicenames = T, testing = F)

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
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 in clude "linear", "logistic",
"coxph", "crr", "boxcox","logistic"
strata    character vector of covariates to stratify by. Only used for coxph and crr
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
testing   boolean to indicate if you want to print out the covariates before the model fits.
            This will allow you to see which model is not fitting if the function throws an
            error
Index

*Topic dataframe
  covsum, 4
eتسم, 5
mvsum, 7
puvsum, 12
uvsum, 14

*Topic helper
  cap, 3
nicename, 8
psthr, 11
pstprn, 12

*Topic model
  boxcoxfitRx, 2
crrRx, 4

*Topic plot
  plotkm, 10

*Topic print
  citime, 3
forestplot, 5
makedocx, 7
pcovsum, 8
petsum, 9
plotci, 9
pmvsum, 11

addspace, 2
boxcoxfitRx, 2
cap, 3
citime, 3
covsum, 4
crrRx, 4
etsum, 5
forestplot, 5
lbld, 6
lpvalue, 6

makedocx, 7
mvsum, 7
	nicename, 8

pcovsum, 8
petsum, 9
plotci, 9
plotkm, 10
pmvsum, 11
psthr, 11
pstprn, 12
puvsum, 12
pvalue, 13

sanitizestr, 13
uvsum, 14