

# 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 demographic 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
lblld . . . . .	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

<b>Index</b>	<b>15</b>
--------------	-----------

---

addspace	<i>Add spaces to strings in LaTeX</i>
----------	---------------------------------------

---

### Description

Add spaces to strings in LaTeX. Returns appends ~~~ before the string

### Usage

```
addspace(x)
```

### Arguments

x	string
---	--------

---

boxcofitRx	<i>fit box cox transformed linear model</i>
------------	---

---

### Description

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

### Usage

```
boxcofitRx(f, data, lambda = F)
```

**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 lamda 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.

---

cap	<i>Capitalize a string</i>
-----	----------------------------

---

**Description**

Calitalize a string

**Usage**

```
cap(x)
```

**Arguments**

x	string
---	--------

---

citime	<i>Get CI confidence interval</i>
--------	-----------------------------------

---

**Description**

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

**Usage**

```
citime(data, response, times, units = "Years",
        outcomes = NULL, decimals = 2)
```

**Arguments**

data	dataframe containing data
response	character vector of response
times	numeric vector specifying single time to get CI for
units	string specifying the unit of times
outcomes	character vector specifying names of competing outcomes. Leave NULL if there is only one outcome
decimals	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	<i>Get event time summary dataframe</i>
-------	---

---

### Description

This function will output a dataframe with usefull 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 provbabilities for.

### Examples

```
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	<i>Create a forrest plot</i>
------------	------------------------------

---

### Description

Create a forrest plot. All entires with cutoff=T will be plotted with an NA rather than their original value.

### Usage

```
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	<i>Bold strings in LaTeX</i>
------	------------------------------

---

**Description**

Bold strings in LaTeX.

**Usage**

```
lbld(strings)
```

**Arguments**

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

---

lpvalue	<i>Formats p-values for LaTeX</i>
---------	-----------------------------------

---

**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	<i>Convert .TeX to .docx</i>
----------	------------------------------

---

**Description**

Covertes 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	<i>Get multivariate summary dataframe</i>
-------	---

---

**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	boooling indicating if you want to replace . and _ in strings with a space

---

nicename	<i>Clean strings for printing</i>
----------	-----------------------------------

---

**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)
```

**Arguments**

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

---

pcovsum	<i>Print covariate summary Latex</i>
---------	--------------------------------------

---

**Description**

Returns a dataframe corresponding to a descriptive table

**Usage**

```
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

```
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

```
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

```
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 choies 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**

```
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**

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

---

pmvsum	<i>Print multivariate summary LaTeX table</i>
--------	---

---

**Description**

Returns a LaTeX table of the multivariate summary.

**Usage**

```
pmvsum(model, data)
```

**Arguments**

model	fitted model object
data	dataframe containing data

---

psthr	<i>Round and paste with parentheses</i>
-------	---

---

**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**

```
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**

```
puvsum(response, covs, data, type = NULL, strata = 1,
       TeX = 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 include "linear", "logistic", "coxph", "crr", "boxcox", "logistic"
strata	character vector of covariates to stratify by. Only used for coxph and crr
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

---

pvalue	<i>Formats p-values</i>
--------	-------------------------

---

**Description**

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

**Usage**

```
pvalue(x)
```

**Arguments**

x	an integer
---	------------

---

sanitizestr	<i>Sanitizes strings to not break LaTeX</i>
-------------	---

---

**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
-----	---------------------------------

---

`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**

<code>response</code>	string vector with name of response
<code>covs</code>	character vector with the names of columns to fit univariate models to
<code>data</code>	dataframe containing data
<code>type</code>	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", "coxph", "crr", "boxcox", "logistic"
<code>strata</code>	character vector of covariates to stratify by. Only used for coxph and crr
<code>markup</code>	boolean indicating if you want latex markup
<code>sanitize</code>	boolean indicating if you want to sanitize all strings to not break LaTeX
<code>nicenames</code>	boolean indicating if you want to replace . and _ in strings with a space
<code>testing</code>	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
- etsum, 5
- mvsum, 7
- puvsum, 12
- uvsum, 14

## \*Topic **helper**

- cap, 3
- nickname, 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

nickname, 8

pcovsum, 8

petsum, 9

plotci, 9

plotkm, 10

pmvsum, 11

psthr, 11

pstprn, 12

puvsum, 12

pvalue, 13

sanitizestr, 13

uvsum, 14