reportRx & R Markdown

Princess Margaret Biostatistics

2021-05-27

Contents

1 Introduction

R Markdown is an alternative to R Sweave for literate programming and reproducible research. It can output to html, Word and PDF.

reportRx is a package to facilitate and standardise the presentation of common statistical analyses.

There are four basic types of functions:

- tabular reporting functions (`covsum`,`uvsum`,`mvvsum`,`ordsum`)
- wrappers for printing in Sweave (`pcovsum`,`puvsum`,`pmvsum`)
- wrappers for printing in RMarkdown (`rm_covsum`,`rm_uvsum`,`rm_mvsum`)
- plotting functions (`ggkmcif`,`plotunivariate`,`forestplot`,`riskplot`)

R Markdown + reportRx = More time on stats and less time on formatting

A demo RMarkdown document show casing the reportRx functions is available here.

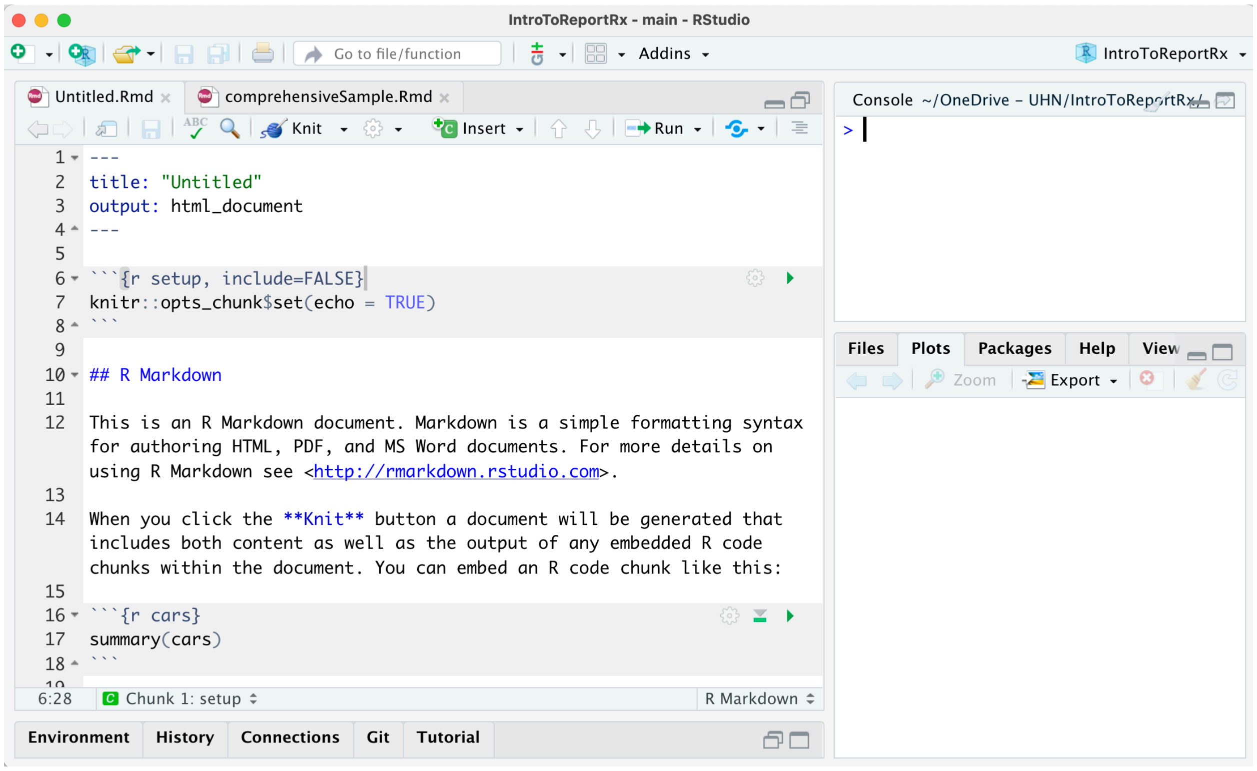
# R Markdown

R Markdown is a package designed for literate R programming and is part of the RStudio IDE.

Like Sweave documents, R Markdown documents contain both text and R code.

R Markdown documents contain three main parts:

1. The YAML header, which is akin to the header in a Latex document.
2. Code chunks, just like Sweave.
3. Text, just like Sweave, but written in markdown, not Latex.



~~~
---
title: "Untitled"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax
for authoring HTML, PDF, and MS Word documents. For more details on
using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

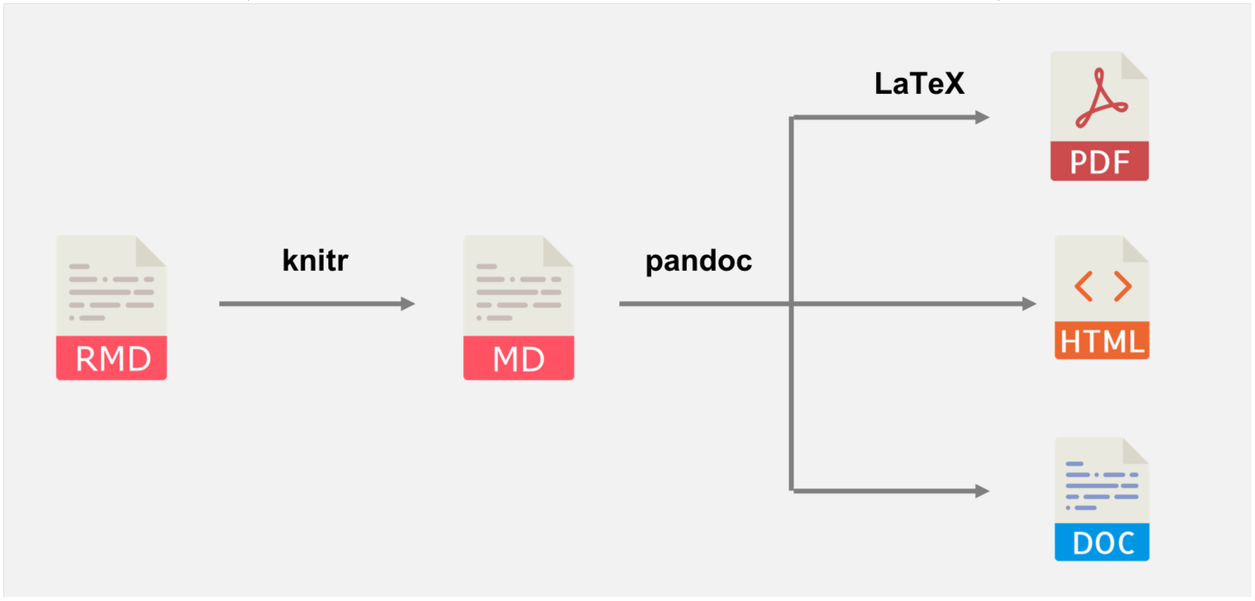
```{r cars}
summary(cars)
```
```

There are excellent tutorials available for learning R Markdown.
```

## 1.1 From R Markdown to PDF/Word/HTML

With reportRx + Sweave .Rnw documents are first made into pdf documents, and then converted to Word documents.

With R Markdown, an .Rmd document is converted to .md and then to PDF, HTML or WORD:



## 1.2 Why Use R Markdown instead of Sweave?

- No need to learn Latex!
  - Formatting is a little quicker: `# Section Header` instead of `\section{Section Header}`
- Easily output to multiple formats, quickly tweak a pdf report to make slides
- Lots of development work is being done to support document creation in R Markdown:
  - Journal Articles & Handouts https://blog.rstudio.com/2016/03/21/r-markdown-custom-formats/
  - Xaringan Presentations https://slides.yihui.org/xaringan/#1
  - Teaching Materials with GitBook https://cjvanlissa.github.io/gitbook-demo/
  - Websites with Blogdown https://alison.rbind.io/post/new-year-new-blogdown/

  If you know Sweave, picking up Markdown is easy!

### 1.3 YAML Header

The top of an R Markdown document is called the YAML header (yet another markup language).

```
---
  title: "Untitled"
author: "Your Name Here"
date: "27 May, 2021"
output:
  pdf_document:
  latex_engine: xelatex
word_document:
  fig_height: 5
fig_width: 7
---
```

This is where you can specify:

- Title
- Author
- Date
- theme (html) or reference style document (Word)
- Table of Contents options
- figure default heights & widths
- bibliography/csl files
- … and lots of other things

Having your own *.Rmd template file with the options you most often use is helpful.

A simple example is available here and a more comprehensive example is here

The ymlthis package can help you to write more complicated YAML headers.

1.4 Code Chunks

Like Sweave, R Markdown has code chunks and the options are very similar. In R Markdown, chunk options can be set globally in the `setup` chunk. These options are over-ridden by any locally set options.

For example, to hide the code and suppress warnings and messages **in the output** you can set the options like this:

```
knitr::opts_chunk$set(echo = FALSE,warning=FALSE,message = FALSE)
library(reportRx)
```

Useful chunk options:

- `eval = FALSE` prevents evaluation of the code. This can be useful for chunks of code that take a long time to run and you may want to run once and save to an R file. It replaces the need to source a separate R script for some code.

- `echo = FALSE` hides the code, but prints any output.

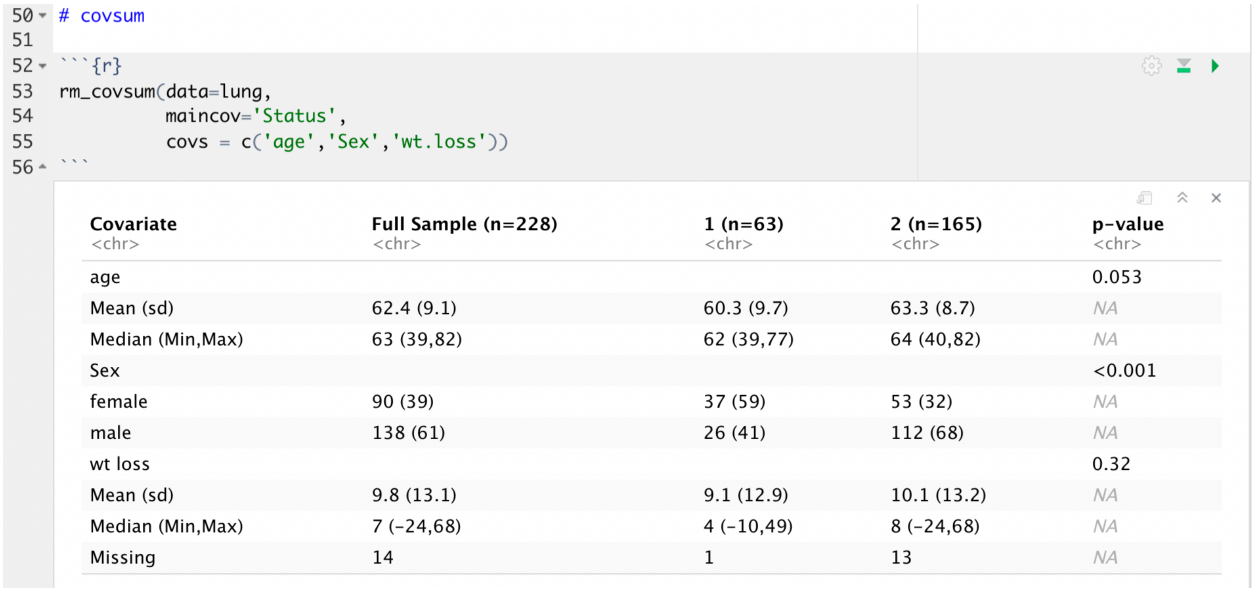
- `warning = FALSE` and `message = FALSE` suppress warnings and messages.
- `results = "hide"` hides the output (can be used with echo=TRUE).
- `results = "asis"` treats the output of your R code as literal Markdown. This is useful if you want to generate text from your R code, for example using `cat`.
- `fig.width = 5` and `fig.height = 5` set the height and width of figures (in inches).
- `fig.cap ='A nice caption'` will print a caption beneath a figure.

For other options see https://yihui.name/knitr/options.

1.5 Running Chunks

You can run section of code with Cmd/Ctrl + Shift + Enter, like Sweave, or execute all the code with the green arrow at the top-right corner of the chunk.

Unlike Sweave, by default code chucks are executed inline, so that you see the output immediately beneath the code chuck, like this:



~~~
# covsum

```{r}
rm_covsum(data=lung,
          maincov='Status',
          covs = c('age','Sex','wt.loss'))
```
~~~

Covariate <chr>	Full Sample (n=228) <chr>	1 (n=63) <chr>	2 (n=165) <chr>	p-value <chr>
age				0.053
Mean (sd)	62.4 (9.1)	60.3 (9.7)	63.3 (8.7)	NA
Median (Min,Max)	63 (39,82)	62 (39,77)	64 (40,82)	NA
Sex				<0.001
female	90 (39)	37 (59)	53 (32)	NA
male	138 (61)	26 (41)	112 (68)	NA
wt loss				0.32
Mean (sd)	9.8 (13.1)	9.1 (12.9)	10.1 (13.2)	NA
Median (Min,Max)	7 (-24,68)	4 (-10,49)	8 (-24,68)	NA
Missing	14	1	13	NA

If you prefer to have the output sent to the Console and the Plots pane instead, then you can you can change this option.

In the RStudio menu: Tools > Global Options > R Markdown > Show output inline for all R Markdown Documents. **Uncheck This**

## Formatting

R Markdown is designed for quick markup. Inserting page breaks, sections, sub-sections, figures, tables and images is very easy.

Formatting is very simple. More extensive formatting is possible and described here.

These are the basic formatting options that will work with any output format:

- `_italic_` or `*italic*` for *italic*
- `__bold__` or `**bold**` for **bold**
    - `<!-- Text Comments -->` for comments that won't appear in the output
- `[PMH Biostatistics](https://www.biostatspm.com/about-us)` for hyperlinks PMH Biostatistics
- `![cute logo](images/logo.png)` for images

1.6 Sections & Breaks

- `# Section Header` will start a new section. The space after the `#` is important.
- `## Subsection Header` for subsections, `### subsubsection` and so on.
- `## unnumbered subsection {-}` add `{-}` for unnumbered headings
- Control how many sections in the table of contents (if you want a TOC) in the yaml header:

```
output:
word_document:
  toc: yes
  toc_depth: 3
```

Pagebreaks (in Word/PDF) can be created with the `\newpage` command.

Horizontal lines can be created with three hyphens (`---`) on a separate line.

1.7 Lists

Create ordered or unordered lists easily. Note that there must be an empty line before the list.

```
* Unordered list
* Item 2
    * Nested bullets need a 4-space indent.
    * Item 2b

1. Ordered list
1. Use `1` for each item and they will be numbered automatically in the output.
1. Handy for when you need to add something in the middle!
```

1.8 Figures and Tables

Plots created in code chucks will be produced in the output, as will Tables formatted by reportRx.

Images not created by R can be inserted with this code: `![Optional Caption](filename.png)`

If you need a table where the data **is not** in a R you can use this (colons determine alignment):

```
| Right | Left | Default | Center |
|------:|:-----|---------|:------:|
|   12  |  12  |    12   |    12  |
|  123  |  123 |   123   |   123  |
|     1 |     1 |      1  |      1  |
```

… but that is a pain, better to have the data in R and use reportRx.

1.9 Formulas

Regular Latex-style formulas can be easily written in R markdown. Use single `$` for inline expressions and `$$` for enclosing standalone expressions.

Examples:

$\beta_0 = 1.0$' for embedded inline formulas like this $\beta_0 = 1.0$

```
$$OR = \frac{Odds_{cases}}{Odds_{controls}}$$
```

$$OR = \frac{Odds_{cases}}{Odds_{controls}}$$

More details and examples of how to write matrices are available here.

1.10 References

In addition to the regular `word_document` and `pdf_document` output types there is a very popular package called `bookdown` that has word and pdf output formats that enable easier cross-referencing. ReportRx will produce tables that are properly formatted in all output formats. You can learn more about bookdown here

To reference a figure, the chunk needs a name, and for fig.caption to be set and can be referenced using `\@ref(fig:speed-plot)` where `speed-plot` is the name of the chunk containing the figure. Example: Figure 1 is an example of a boxplot.

```
{r speed-plot,fig.cap='Speed and distance.',fig.height=2.5}
data(cars)
plot(x=cars$dist,y=cars$speed)
```

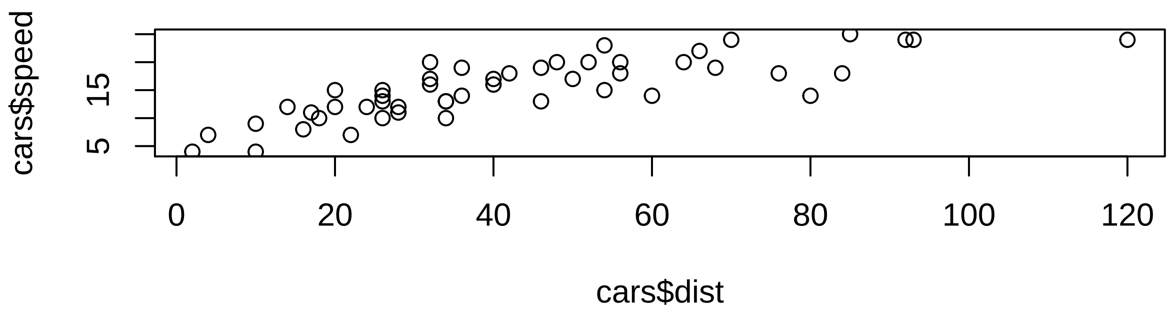


Figure 1: Speed and distance.

Table 1: Summary sample statistics.

Covariate	n=50
speed	
Mean (sd)	15.4 (5.3)
Median (Min,Max)	15 (4,25)
dist	
Mean (sd)	43.0 (25.8)
Median (Min,Max)	36 (2,120)

To reference tables is a similar, use `\@ref(tab:speed-tab)`. If outputting to Word, `chunk_label` needs to be specified as an argument. For HTML/PDF this can be omitted. Hopefully this will be automated for Word one day.

Example: Tab 1 is an example of a table.

```
{r speed-tab}
rm_covsum(data=cars,covs = c('speed','dist'),
          chunk_label = 'speed-tab')
```

1.11 Citations

To add citations use `[@bibindex]`. For example `[@Ensor2014]` will produce a citation in the document like this: (1), and will be added to the bibliography with the following entry:

1. Ensor JE. Biomarker validation: Common data analysis concerns. The Oncologist. 2014;19(8):886–91.

‘

R packages can be referenced like this: `[@R-reportRx]` and R like this: `[@R-base]`. To cite more than one article or package separate them with semicolons: `[@R-base;@R-reportRx]`.

1.12 Creating a bibliography

For R Markdown to create the bibliography, a bibliography file (in the biblatex format) needs to be specified in the YAML. Optionally, a csl file can also be specified. Note that if no filepath is specified, then the directory of the .Rmd file is used. In this example, bibfile.bib is in the local directory, but the csl is stored in a central location.

```
bibliography:  bibfile.bib
csl: ../../../csl/Vancouver.csl
```

There is now a reportRx function, `rmdBibfile` that will read through an R Markdown document and extract all the references from a master bibfile and write them to a local file along with and all the R packages referenced in the document.

`reportRx::rmdBibfile('../library.bib','bibfile.bib')` will use the file library.bib in the parent directory to extract references from and write a smaller bib file in the local directory stored as bibfile.bib with all the references in the current document, including R packages.

The master bibfile can be the .bib file produced by programs like Mendeley or Zotero.

1.13 Moving References

By default, references are at the end of the document which can be annoying if you would like to put them before an Appendix.

Use this command to place the references (this works for all output types).

```
# References
<div id="refs"></div>`
```

2 reportRx

reportRx has reporting functions, printing functions and plotting functions. The actual reporting functions are not called directly, but are called by the printing functions.

Reporting Functions:

- covsum Used to produced Table 1 style output
- etsum To summarise output from a coxph model
- uvsum To combine several univariate models into a single table
- [ordsum] Like `uvsum` for ordinal output. May be added into uvsum one day
- mvsum To summarise multivariable model output

Printing functions for **Sweave** use the prefix `p` (ie. `pcovsum()`).

Printing functions for **R Markdown** use the prefix `rm_` (ie `rm_mvsum()`).

2.1 installing reportRx

To install `reportRx` from the github repository you need to install the `devtools` package:

```
install.packages('devtools')
```

Our most recent version, with the features presented here can then be directly downloaded from github:

```
devtools::install_github("biostatsPMH/reportRx", ref="devel")
```

For the older stable version you can use this:

```
devtools::install_github("biostatsPMH/reportRx", ref="master")
```

2.2 lung data

The sample code will use data from the `lung` dataset in the `survival` package. To run the code samples shown here requires that the lung data be loaded and mutated as follows:

```
library(tidyverse)
library(survival)
library(reportRx)
data(lung)
lung <- lung %>%
  mutate(
    Status=factor(status-1),
    Sex = as.character(factor(sex,labels = c('Male','Female'))),
    sex = factor(sex),
    OneLevelFactor = factor(x='one level')
  )

lung$Sex[sample(1:nrow(lung),size=10)] <- NA
```

2.3 covsum

`covsum` can be used to generate Table 1 style output for either the entire sample, or by specifying the `maincov` argument, by subgroups. Note the use of `rm_covsum`.

Typing `?covsum` will provide a complete description of the covsum arguments. Here the most recent changes are highlighted.

```
rm_covsum(data=lung,
       covs=c('Status','wt.loss','OneLevelFactor'),
       digits=2,
```

Table 2: Summary sample statistics by Sex.

Covariate	Full Sample (n=228)	Female (n=88)	Male (n=130)	NA (n=10)
Status				
0	63	37 (59)	22 (35)	4 (6)
1	165	51 (31)	108 (65)	6 (4)
wt loss				
Mean (sd)	9.83 (13.14)	7.80 (13.26)	11.06 (12.77)	12.67 (15.81)
Median (Q1,Q3)	7.00 (0.00,15.75)	4 (0,11)	8 (1,18)	5 (0,23)
Range (min, max)	(-24,68)	(-24,52)	(-13,68)	(-5,38)
Missing	14	3	10	1
OneLevelFactor				
one level	228	88 (39)	130 (57)	10 (4)

```
    maincov = 'Sex',
    all.stats=TRUE,
    include_missing=T,
    percentage='row',
    pvalue=FALSE)
```

New Functionality & Bug Fixes

- `digits` number of digits for summarizing numeric data
- `pvalue` boolean indicating if you want p-values included in the table
- `full` boolean indicating if the full sample column should be displayed
- `include_missing` prints the number of values of the maincov missing and excluded from the table
- `percentage` choice of how percentages are presented ,one of *column* (default) or *row*
- `show.tests` option to display the statistical tests performed
- `all.tests` option of showing both the IQR and the range on separate lines
- `excludeLevels` option to exclude levels from covariates from association tests
- Added functionality to test for small counts in contingency table and perform Fisher.exact if req'd
- If testcont=T will perform unequal variance t-test for two groups
- character variables are automatically converted to factors
- function will now work with data imported from SPSS using `haven` package
- function works as expected with factors containing only a single level
- if median, Q1,Q3 (or Min/Max) are all integers decimals are not reported
- NaN is no longer displayed, instead cells are empty
- silenced the try function

2.4 uvsum

`uvsum` will produce a single table with many univariate results

Table 3: Univariate analysis of predictors of Status.

Covariate	OR(90%CI)	p-value	N
wt loss	1.01 (0.99,1.03)	0.61	214
Sex		**<0.001**	**218**
Female	Reference		88
Male	3.56 (2.11,6.01)		130
ph ecog	**2.17 (1.50,3.15)**	**<0.001**	**227**
meal cal	1.00 (1.00,1.00)	0.75	181
age	**1.04 (1.01,1.07)**	**0.025**	**228**

```
rm_uvsum(response = 'Status',
         covs=c('wt.loss','Sex','ph.ecog','meal.cal','age'),
         data=lung,
         CIwidth=.9)
```

New Functionality & Bug Fixes

- **showN** is an option to show the sample size for each variable/factor level
- **CIwidth** allows for differ confidence intervals to be produced
- output
- proper t-distribution confidence intervals for means are produced
- function will check for variables and produce useful warnings
- character variables are automatically converted to factors
- function will now work with data imported from SPSS using `haven` package

2.5 mvsum

`mvsum` will return a consistently formatted table for different types of mutivariable models

```
glm_fit = glm(Status~Sex+age+wt.loss,data=lung,family = 'binomial')
rm_mvsum(glm_fit)
```

Covariate	OR(95%CI)	Global p-value
Sex	**3.47 (1.81,6.64)**	**<0.001**
age	**1.04 (1.01,1.08)**	**0.016**
wt loss	1.00 (0.98,1.03)	0.93

New Functionality & Bug Fixes

- `showN` is an option to show the sample size for each variable/factor level
- `CIwidth` allows for differ confidence intervals to be produced
- automatically retrieve data from the model if not specified
- fixed bug with variable names embedded in level names

- fixed bug with centering in models
- added support for polr ordinal regression models
- updated CIs to use t-test instead of Z-test for means
- fixed treatment of glm objects to properly handle linear, binomial and poisson models

2.6 ggsurv

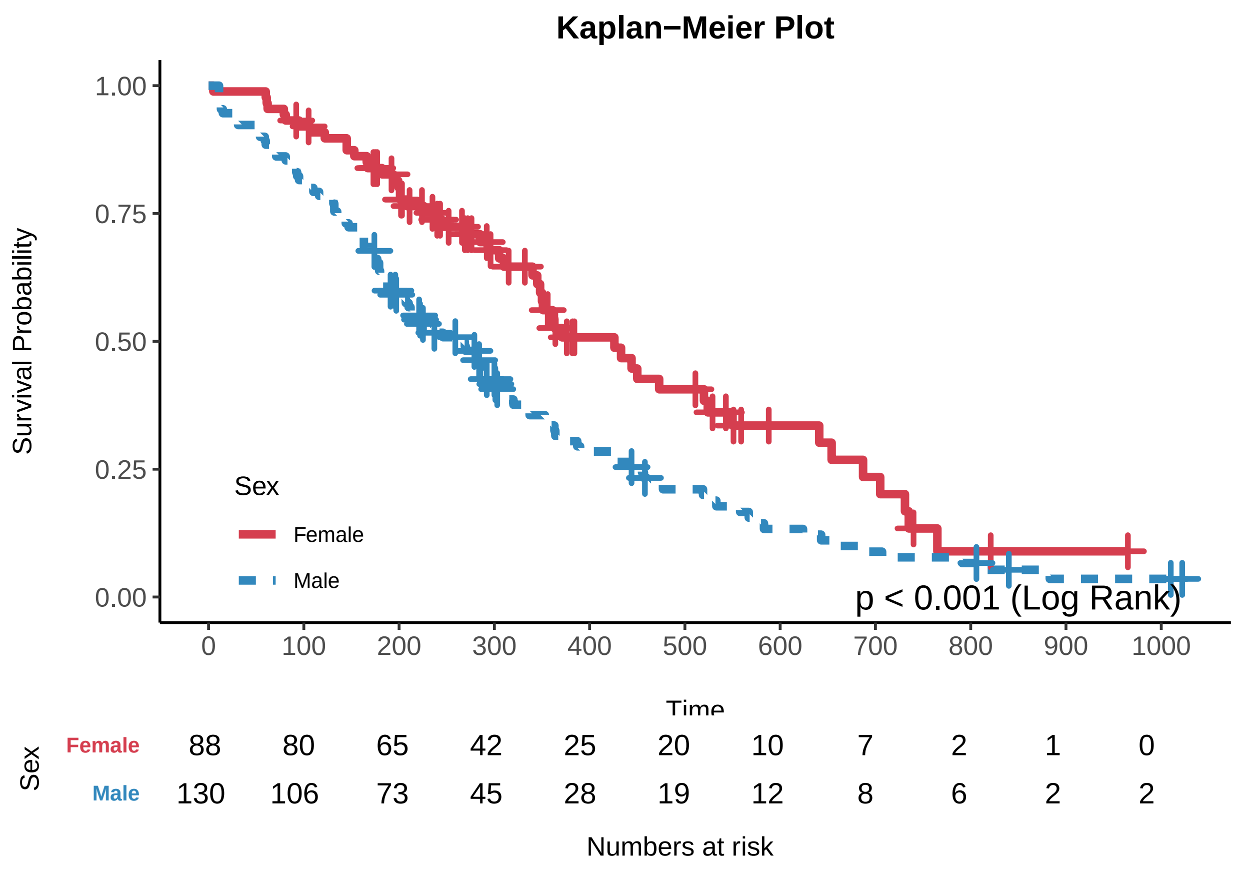
… this function has been replaced by `ggkmcif` and may be deprecated.

2.7 ggkmcif

Plot KM and CI curves using ggplot2 with the usual reportRx arguments and a lot of options for customisation for publication-ready plots.

```
ggkmcif(c("time","status"),"Sex", data=lung,fsize=10,nsize=4)
```

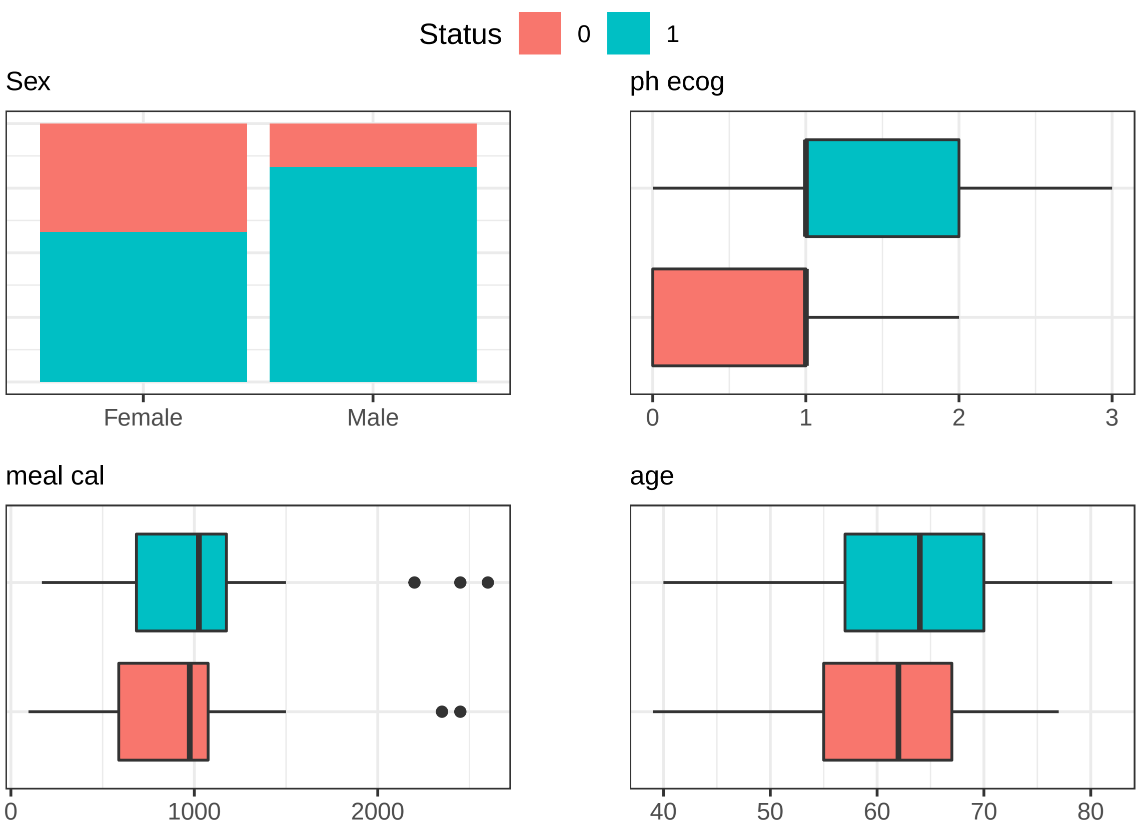
```
## [1] "10 observations have been removed due to missing data"
```



2.8 plot_univariate

Designed to be a visualisation of `uvsum` with arguments supplied in the same manner. This goal is not publication-ready plots but rather a quick means of displaying the data to facilitate the interpretation of the `uvsum` output from.

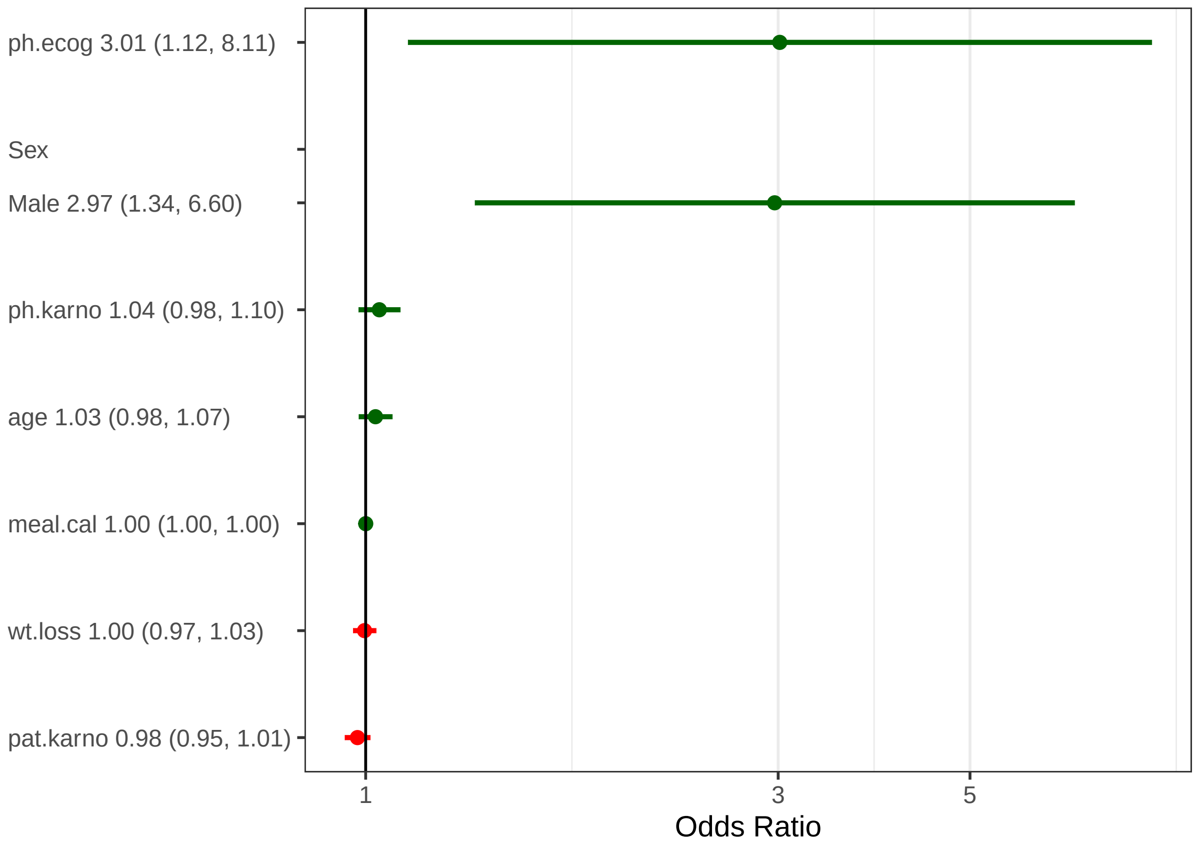
```
plot_univariate(response = 'Status',
                covs=c('Sex','ph.ecog','meal.cal','age'),
                data=lung)
```



2.9 forestplot2

This will produce a forest plot of OR or RR from an object derived from glm. The risks are plotted on a log-scale, ordered from highest to lowest.

```
fit = glm(Status~age+Sex+wt.loss+ph.karno+ph.ecog+pat.karno+meal.cal,
          family='binomial',
          data=lung)
forestplot2(fit, rmRef=T)
```



(**Note:** There is an existing forestplot function, which is why this one is named forestplot2)

2.10 etsum

Automatically output text from survival analysis. Set the code chuck option `results='asis'`.

!**Note:** This function may have bugs. Please let us know if you have trouble.

```
rm_etsum(data = lung, response = c("time","Status"), group = 1,
         times=c(365,720,1095), units="days")
```

Will output this:

```
There are 228 patients. There were 165 (72\%) events. The median and range of the follow-up ti
```

2.11 outTable

This function will output a simple table nicely in any of the main R Markdown output formats, including bookdown.

Arguments:

- `to_indent` row indices specifying a small indent in the first column

Table 4: Status by Sex in the Lung data

Status	Female	Male	NA_
0	37	22	4
1	**51**	**108**	**6**

- `to_bold` row indices specifying which rows to bold
- `caption` a caption to print above the table.

```
library(janitor)
tab <- lung %>% tabyl(Status,Sex)
outTable(tab,
         to_indent=1,
         to_bold=2,
         caption= 'Status by Sex in the Lung data' )
```

2.12 excelCol

Retrieve the column numbers corresponding to Excel column headers. This can be useful if variables are referred to by Excel column names in the statistical analysis plan.

Example:

```
excelCol(A,B,BG,CC)
```

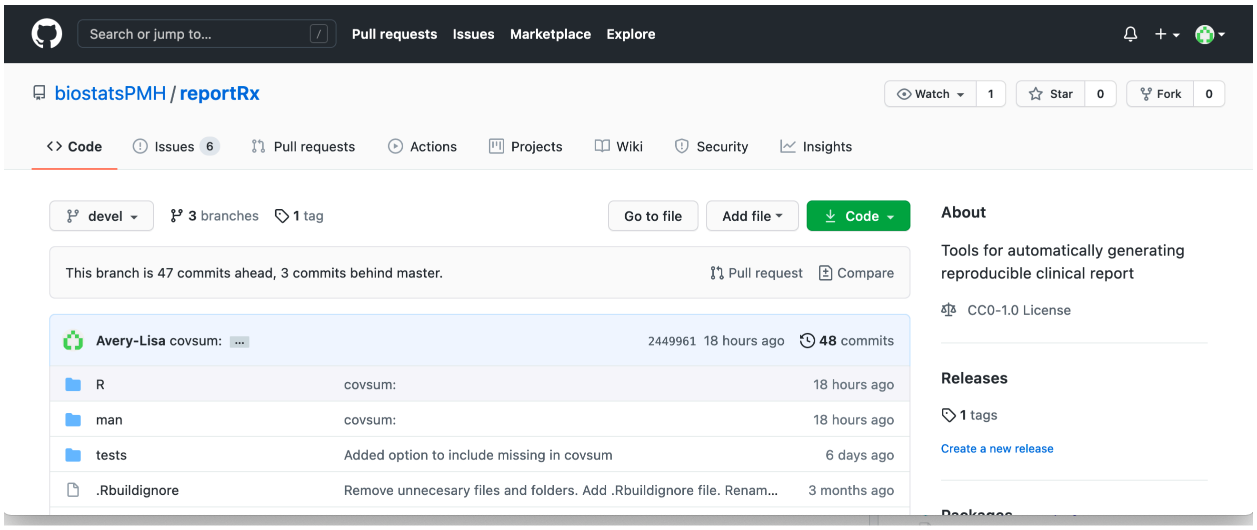
Returns:

```
 A  B BG CC
 1  2 59 81
```

3 Github

reportRx and this documentation are available on the PMH Biostats Github page, along with Jessica's swimmplot package.

Anyone can copy or download code from the git repository, without fear of deleting anything.



This branch is 47 commits ahead, 3 commits behind master.

Avery-Lisa covsum: … 2449961 18 hours ago 48 commits

R	covsum:	18 hours ago
man	covsum:	18 hours ago
tests	Added option to include missing in covsum	6 days ago
.Rbuildignore	Remove unnecesary files and folders. Add .Rbuildignore file. Renam...	3 months ago

About

Tools for automatically generating reproducible clinical report

CC0-1.0 License

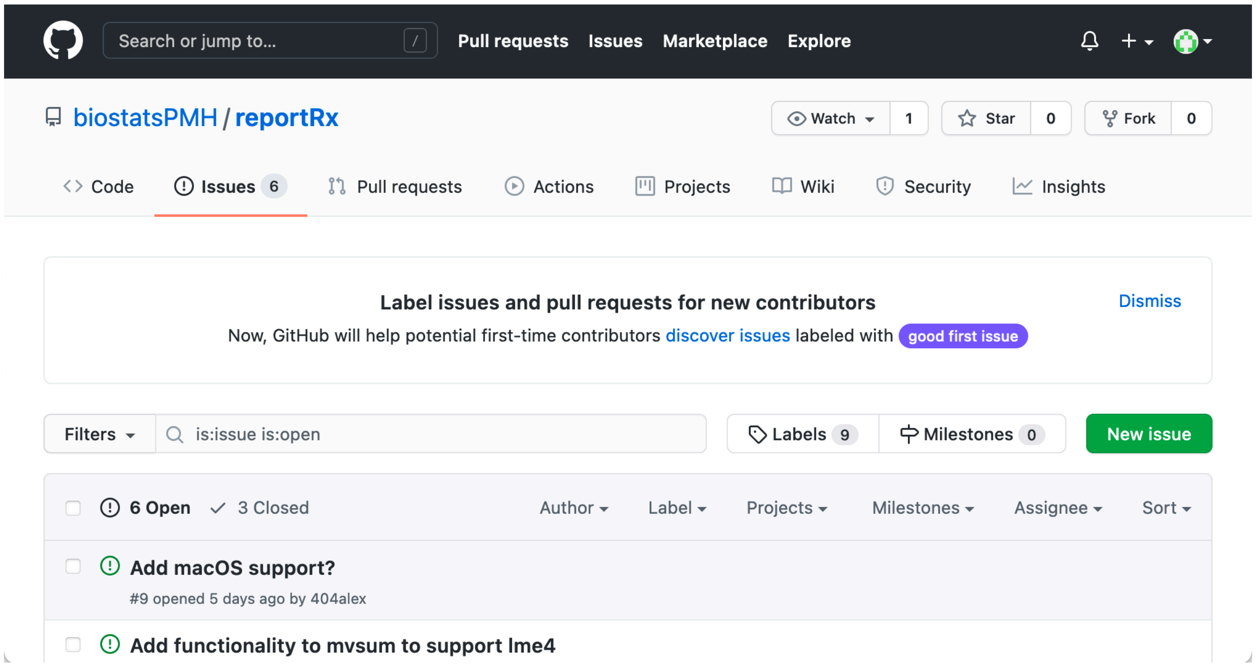
Releases

1 tags

Create a new release

3.1 Help us Improve

If you find a bug or would like to request a new feature you can log an issue



Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

Add macOS support?
#9 opened 5 days ago by 404alex

Add functionality to mvsum to support lme4

3.2 Github Account

If you don't already have a github account you will need to set one up to log issues or access the code.

Please log issues using your **uhnresearch email** (and your real name!). This will help us to prioritise issues arising from within the department.