My Electronic R Lab Notebook for WILS 2016

Miles Benton

Last updated: 18 November, 2016

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

This document is an example template which demonstrates the utility and power of R/RStudio to provide useful tools for not only data exploration and analysis but also electronic reporting and disemination. The main use case of this document is to show the specific use of R Notebooks as an excellent alternative to a hand written lab book.

There are many benefits to using an electronic 'lab book' in the R/RStudio ecosystem:

- while students are working on there 'experiements' they are also writing notes and recording exactly what they are doing
- code and data is explorable and evaluated within the same document as text
- inline citation and bibliography management
- inline citation of figure and tables
- the notebook is easily synced to version control (backup systems) such as GitHub, so there is no chance of ever losing your work
- the notebook can be exported out to numerous formats including: Word docx, pdf, HTML, tex, …

Here is a list of required packages to load:

```
knitr::opts_chunk$set(dpi = 150)
require(knitr)
require(knitcitations)  # used to create citations and bibliography
require(printr)
options("citation_format" = "pandoc")
```

Friday 18th November 2016

What follows is a template for the basic structure laid out in the WILS Lab Book pdf manual.

Title

Using R Notebooks for electronic lab book documentation in Bioinformatics (and generally any other field you want!)

Aims

Explore the use of the R Notebook environment for creating a reproducible and dynamic reporting system for use in the field of Bioinformatics.

Introduction

It has been well established that the tools contained within the RStudio software have made creating and diseminating electronic reports much easier. By using RMarkdown users are able to generate notes/written content in the same ernvironment as working with their data and producing statistics and results.

Materials

For this demonstration we need to download and install several pieces of software.

First would be `R`, which can be downloaded here: https://www.r-project.org/

Next we'll want to grab `RStudio`, here is the link: https://www.rstudio.com/products/rstudio/download/preview/

If you want to read more about RMarkdown: http://rmarkdown.rstudio.com/

More information on R Notebooks: http://rmarkdown.rstudio.com/r_notebooks.html

Methods

We will be exploring a range of different methods and functions from both R [1] and RStudio, and see how they are implemented in R Notebooks and how that can be leveraged to help us create an electronic record, or lab book.

Results

Here are some of the many features that are available when using R Notebooks.

Display code

We are able to write and evaluate code using *chunks*, a concept implemented in RMarkdown - allowing us to both write text and code within the same document.

To genetate a new chunk press *ctrl+alt+i*, within the chunk you can write and evaluate R code:

```
1+1
```

```
## [1] 2
```

Evaluate statistics/models and display in text

```
lm(cars)
```

```
##
## Call:
## lm(formula = cars)
##
## Coefficients:
## (Intercept)         dist
##      8.2839       0.1656
```

```
summary(lm(cars))
```

```
##
## Call:
## lm(formula = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5293 -2.1550  0.3615  2.4377  6.4179
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.28391    0.87438   9.474 1.44e-12 ***
## dist         0.16557    0.01749   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.156 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Can even include plots

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
plot(cars, pch = 16, col = "cadetblue")
```

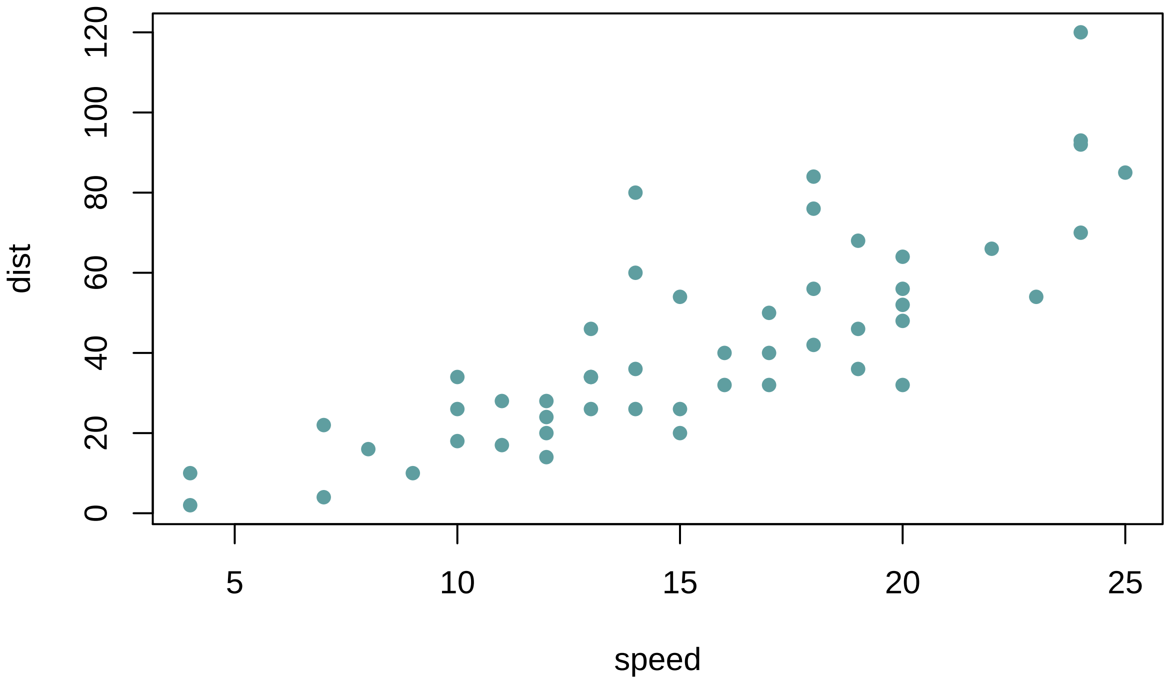


Figure 1: Simple plotting using mtcars data.

A really nifty feature is that plotted figures get asigned a figure number which is able to be referenced in text (see Figure 1).

We can create the same plot as before, but this time fit the regression model that we previously generated. We'll plot the best fit line using the *abline* function.

```r
plot(cars, pch = 16, col = "cadetblue")
abline(lm(cars$dist ~ cars$speed))
```

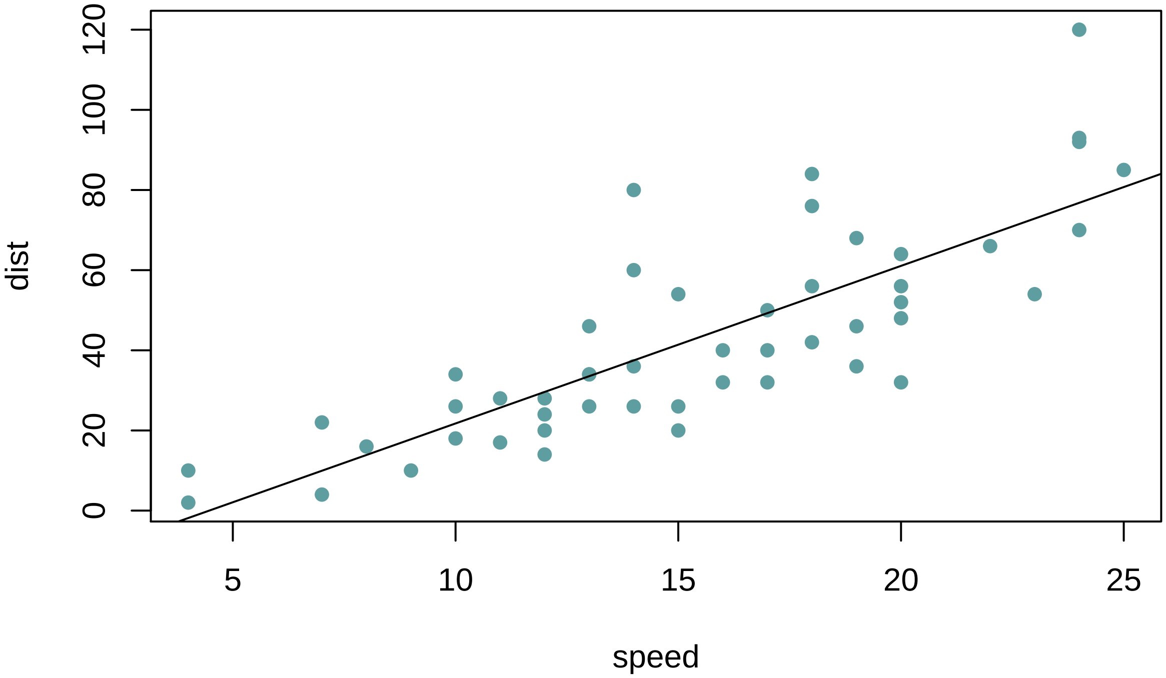


Figure 2: Plotting regression line using mtcars data.

Figure 2 shows the same data we previously plotted, this time with a fitted line from our linear regression.

Tables are also easy

We can use the *kable* function from *knitr*:

```r
n <- 100
x <- rnorm(n)
y <- 2*x + rnorm(n)
out <- lm(y ~ x)
kable(summary(out)$coef, digits=2, caption = 'An example table produced by knitr kable().')
```

Table 1: An example table produced by knitr kable().

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.08	0.11	-0.75	0.45
x	2.12	0.09	22.73	0.00

There is a new package called *printr* which is a companion app to *knitr*, aiming to make tables look nicer. The package is currently not on CRAN, so you will need to install it as below:

```r
# install printr
install.packages(
  'printr',
  type = 'source',
  repos = c('http://yihui.name/xran', 'http://cran.rstudio.com')
)
# load the package
library('printr')
```

The *printr* package aims to make the standard output from `R` a little nicer to look at (and easier to read), have a look at the following:

```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Citation is straight forward

Using the `knitcitations` package makes referencing easy! Just use the *citep* function and provide either a DOI or html link to the citation and R will take care of the rest, including building the reference list at the end of the document.

Example: *glmnet* is a machine learning package with implements an Elastic-net framework. [2]

The embedded RMarkdown code for the above citation looks like this:

```
`r citep("http://www.jstatsoft.org/v33/i01/"`)
```

Output to many popular formats

RStudio uses the power of the `knitr` [3] package and `pandoc` to allow the exporting of your document to several flexible and popular formats.

HTML

By selecting *knit to HTML* an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

Word (.docx)

By selecting *knit to Word* your notebook will be exported to *.docx* format, which will be able to be read by Microsoft Word and other office packages (i.e. Libre Office).

pdf

By selecting *knit to Word* your notebook will be exported to *.pdf*, the universal format.

Discussion

For any heavily based computational or ’informatics program we should be moving towards embracing technology not only in the tools we use for research, but also those for methods of recording and diseminating our work. As researchers in this field are almost exclusively working on their computers it makes sense to have an electronic record rather thantraditional hand written lab books.

This set up is able to run on any OS platform (Linux, MacOS, Windows, etc).

Conclusion

The framework provided by R, RStudio and RMarkdown come together in the form of R Notebooks and make for an excellent form of electronic recording for daily activities/experiments, providing an ideal solution for Bioinformatics lab books.

Monday 21st November 2016

Can easily create bullet lists:

- induction at 10.00am
- supervisor meeting at 12.30pm
- Introduction to R/RStudio workshop at 2.00pm

Also numbered lists:

1. first load the data
2. explore data and QC
3. generate summary statistics

Tuesday 22nd November 2016

1. Explore the lab and get an idea where the data is coming from.

2. Look into chi squared tests in R:

```
## From Agresti(2007) p.39
M <- as.table(rbind(c(762, 327, 468), c(484, 239, 477)))
dimnames(M) <- list(gender = c("F", "M"),
                    party = c("Democrat","Independent", "Republican"))
(Xsq <- chisq.test(M))  # Prints test summary
```

```
##
##  Pearson's Chi-squared test
##
## data:  M
## X-squared = 30.07, df = 2, p-value = 2.954e-07
```

```
Xsq$observed   # observed counts (same as M)
```

gender/party	Democrat	Independent	Republican

gender/party	Democrat	Independent	Republican
F	762	327	468
M	484	239	477

```
Xsq$expected   # expected counts under the null
```

	Democrat	Independent	Republican
F	703.6714	319.6453	533.6834
M	542.3286	246.3547	411.3166

```
Xsq$residuals  # Pearson residuals
```

gender/party	Democrat	Independent	Republican
F	2.198856	0.4113702	-2.843240
M	-2.504669	-0.4685829	3.238673

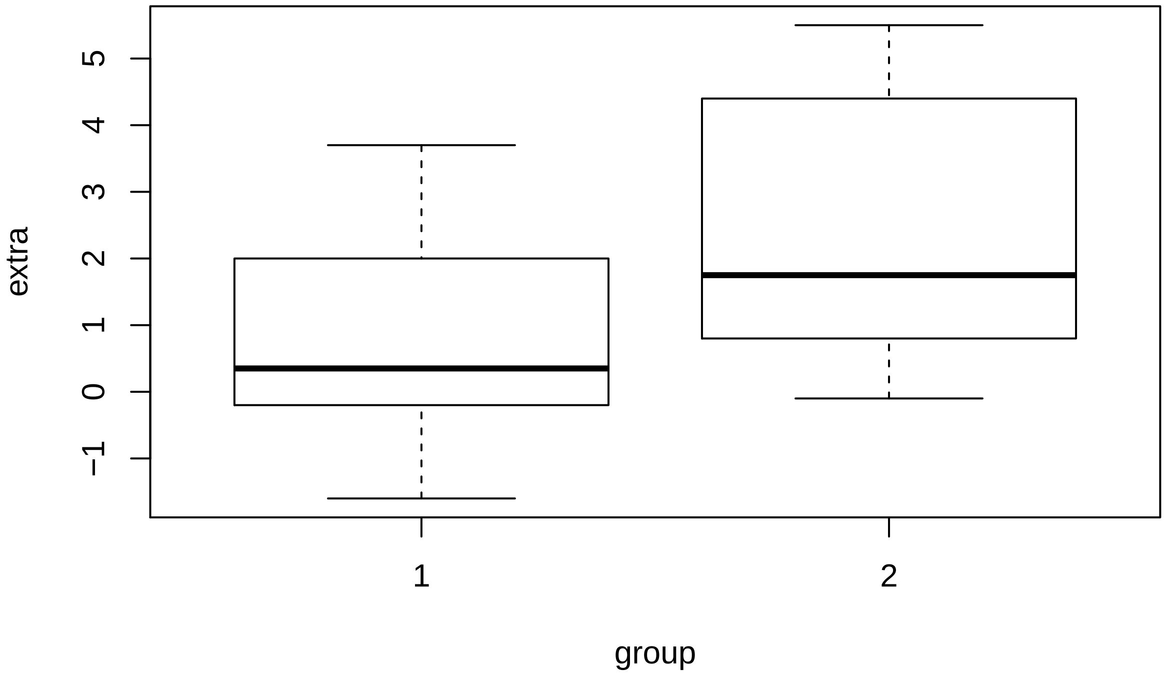
```
Xsq$stdres     # standardized residuals
```

gender/party	Democrat	Independent	Republican
F	4.502053	0.6994517	-5.315945
M	-4.502053	-0.6994517	5.315945

Wednesday 23rd November 2016

Use R to do simple students t-tests:

```
## Classical example: Student's sleep data
plot(extra ~ group, data = sleep)
```



```
## Traditional interface
with(sleep, t.test(extra[group == 1], extra[group == 2]))
```

```
##
##  Welch Two Sample t-test
##
## data:  extra[group == 1] and extra[group == 2]
## t = -1.8608, df = 17.776, p-value = 0.07939
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.3654832  0.2054832
## sample estimates:
## mean of x mean of y
##      0.75      2.33
```

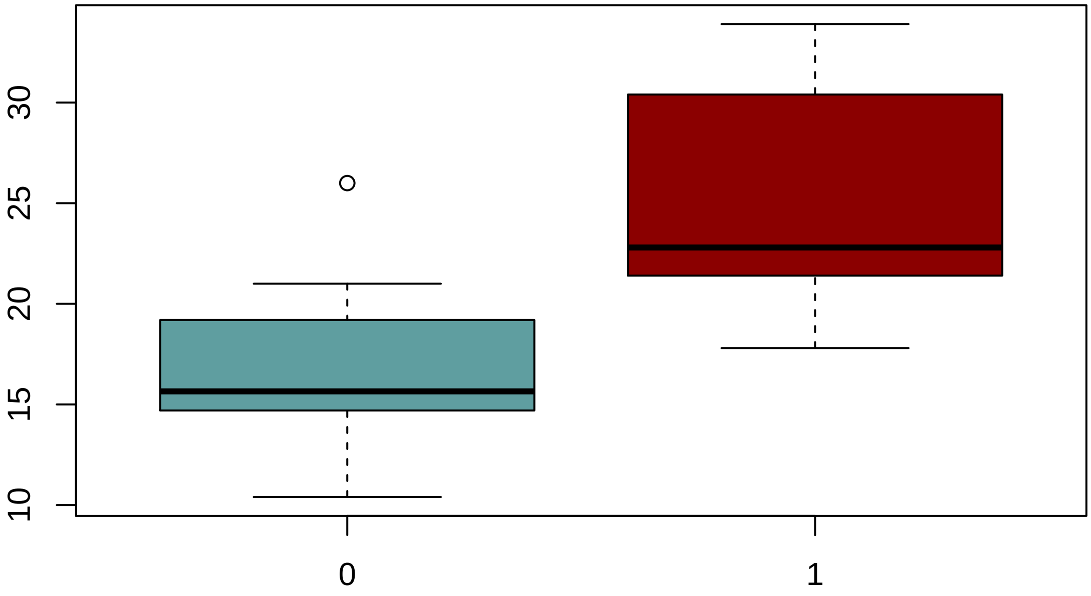
```
## Formula interface
t.test(extra ~ group, data = sleep)
```

```
##
##  Welch Two Sample t-test
##
## data:  extra by group
## t = -1.8608, df = 17.776, p-value = 0.07939
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.3654832  0.2054832
## sample estimates:
## mean in group 1 mean in group 2
##            0.75            2.33
```

Thursday 24th November 2016

Learn how to make box plots in R.

```
boxplot(mtcars$mpg ~ mtcars$vs, col = c('cadetblue', 'darkred'))
```



Too easy!

Friday 25th November 2016

Plan for the day:

- explore integrating lab notebook with a version control service (GitHub)
- look into examples of using the *glmnet* package
- integrate all data sets and learn how to filter data in R
- sit back and realise how awesome R is!! :)

References

1. R Core Team. R: A language and environment for statistical computing [Internet]. 2016. Available: https://www.R-project.org/

2. Friedman JH, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent [Internet]. Journal of Statistical Software. 2010. Available: http://www.jstatsoft.org/v33/i01/

3. Xie Y. Knitr: A comprehensive tool for reproducible research in R. In: Stodden V, Leisch F, Peng RD, editors. Implementing reproducible computational research. 2014. Available: http://www.crcpress.com/product/isbn/9781466561595