

Introduction to R Workshop



Introductions Introduction to R Workshop

Dr Jeromy Anglim
Deakin University
jeromy.anglim@deakin.edu.au

Internet / Wi-Fi: See "Guest WiFi Deakin"
If you have not already done so, files for workshop can be downloaded from:
<http://bit.ly/rworkshop2015>
Or by going to
<https://github.com/jeromyanglim/introduction-to-r-one-day-workshop>
and clicking "Download ZIP"

Overview of the day

Time	Activity
10:00 – 11:30 (90 minutes)	Workshop: Morning session
11:30 – 11:45 (15 minutes)	Break: Morning
11:45 – 1:00 (75 minutes)	Workshop: Noon session
1:00 – 1:45 (45 minutes)	Break: Lunch
1:45 – 3:15 (90 minutes)	Workshop: Afternoon session
3:15 – 3:30 (15 minutes)	Break: Afternoon
3:30 – 5:00 (90 minutes)	Workshop: Final session

- Housekeeping
 - Toilets
 - Internet / Wi-Fi: See "Guest WiFi Deakin"
 - Power
 - Food / Drinks
 - Laptop security
 - Feel free to ask questions

Aims of workshop

- By the end of this workshop you should have a basic understanding of
 - What R can do
 - Whether you want to use R in the future
 - How the language works
 - How to perform standard data analysis activities with R such as importing, manipulating, graphing, and modelling data
 - How to combine knitr and ProjectTemplate to implement a reproducible workflow for a data analysis project

Overview of content

- Sessions 1 and 2
 - Introductions
 - What can R do? Why and when would you want to use R?
 - How to get help; Working with RStudio
 - Core Language
 - Data types, functions, operations, loading data, saving data
 - R Packages (installing, loading, using)
 - Data manipulation
 - Graphs (base, lattice, ggplot2)
- Session 3
 - Standard statistical functions: Descriptive statistics, correlations, linear regression
 - Overview of what is possible
 - Generalised linear models, multilevel modelling, structural equation modelling, Bayesian analysis, bootstrapping, meta-analysis
- Session 4
 - Workflow
 - How to organise a complete data analysis project with ProjectTemplate
 - How to use knitr to create reproducible data analysis documents that process raw data and produce statistical output

Workshop format

- Slides
 - I will present some slides
 - Most content will be covered in the interactive presentations
- Interactive presentations
 - I will introduce many concepts by demonstrating functionality in R and RStudio
 - You are encouraged to open these script files and run them on your laptop as I work through them
- Exercises
 - There will be regular points where you are given brief exercises where you asked to apply the concepts presented

What is R?

- "R is a free software environment for statistical computing and graphics" - <http://www.r-project.org>
- It is a full-featured statistical analysis package (e.g., like SAS, Stata, SPSS, etc.) that allows you to:
 - Import and export data in a wide range of formats
 - Manipulate data
 - Analyse data
 - Graphically represent data
- It is a full-featured programming language designed for data analysis
- Why the name "R"?
 - First letter of two originators: Ross Ihaka and Robert Gentleman
 - Built on a earlier language called "S"

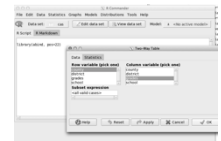
Introduction to R Workshop

Why use R?

- R is free to use
- R is open source
- R runs on Windows, OSX, Linux
- R has a huge library of user-contributed packages (over 6,000 on CRAN)
 - Functionality not available in other software
 - Only need to learn the new bits of the specific package; i.e., you can use the same graphics, data manipulation, data simulation, programming tools across analysis tasks
- R is flexible
- R facilitates reproducible research
- R makes data analysis fun
- Popular in academia and industry: lots of free online resources

Challenges of using R?

- R involves writing scripts; it does not have a GUI like SPSS, SAS, Stata, etc.
 - Although see Rcmdr (<http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>) – GUI training wheels rather than dedicated analysis tool
 - If you like data mining, see <http://rattle.togaware.com/>
 - See RStudio for a user friendly Integrated Developer Environment (IDE)
- R is more interactive
 - In SPSS and SAS you choose a command and get piles of output which you wade through
 - R is a conversation: You interactively request relevant output
- Mental model for R has links with formal statistics, unix, and programming



Installing R and RStudio

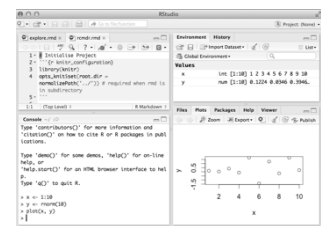
- Install base R
 - <http://www.r-project.org>
- Install RStudio
 - <http://www.rstudio.com>
- Install additional R packages
 - To be discussed later, but generally use RStudio or
 - `install.packages("packagename")`
- Some R functionality requires other (free) software to be installed
 - Various compilers, command-line tools, etc.
 - RTools for Windows <http://cran.r-project.org/bin/windows/rtools>
 - Xcode for OSX <http://developer.apple.com/xcode>
 - Python: <http://www.python.org/downloads>
 - Java SE JDK: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - Perl: <http://www.perl.org/get.html>
 - pandoc: <http://pandoc.org/installing.html>
 - TeX/LaTeX: <http://latex-project.org/ftp.html>



RStudio

- There are many interfaces to R
 - http://www.sciviews.org/_rgui
- RStudio
 - Best general interface to R
 - Improving rapidly
 - Free and open source

• <http://www.rstudio.com>



Instructional materials on R

- Official manuals
 - <http://cran.r-project.org/manuals.html>
 - Fairly technical
- Contributed Documentation
 - <http://cran.r-project.org/other-docs.html>
 - Many useful tutorial style documents from a range of different perspectives
- Many paper books on R
 - <http://www.r-project.org/doc/bib/R-books.html>
- Task Views
 - <http://cran.r-project.org/web/views>
 - Curated overviews of packages relevant to particular domain

Getting Help Online

- Quick-R website:
 - Excellent website with code templates for common tasks
 - <http://www.statmethods.net>
- Reference Cards: See particularly Tom Short's
 - Print out and gradually expand your R vocabulary
 - See particularly Tom Short's <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>
 - RStudio cheatsheets <http://www.rstudio.com/resources/cheatsheets/>
- Google usually has the answer
 - Generally adding "with R" or "in R" to a statement of what you want to do should suffice: e.g., "multiple regression with R"
 - Or use dedicated R search engine: <http://rseek.org>



Introduction to R Workshop

RStudio Projects

- It is good practice to store all files related to a particular analysis project within a single directory on your computer that stores only files related to that project
 - Such files include scripts, data files, configuration files, figures, exported tables, etc.
 - You may also store such files in subdirectories (e.g., store data in a subdirectory called `data`)
- If you work with Rstudio it helpful to make this folder an Rstudio Project (Go to: `File – New Project`)
- This will generate a file with an ".Rproj" extension (e.g., `myproject.Rproj`)
- You can then double click on this file to open the project

Benefits of RStudio Projects

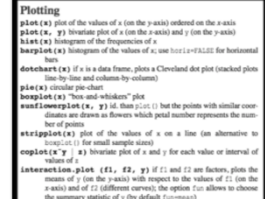
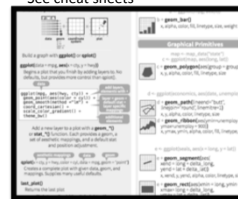
- Ensures that the working directory is the same as the project directory. Thus, when you load or save files, you can have confidence that your script will work.
- The file pane is opened in the working directory, so it's easy to access other files in the project
- RStudio will re-open previous open scripts

Core Language Features

- Review RStudio User Interface
- Open `1-core-language.rmd`
- Topics
 - Assignment
 - Workspaces
 - Data types
 - Getting Help
 - Packages
 - Missing Data
 - Data Summaries
 - Functions
 - Importing / Export Data
 - Random Variables
 - Functions

Graphics and Data Manipulation

- Review RStudio User Interface
- Open `2-graphics-data-manipulation.rmd`
- Topics
 - Base graphics
 - Lattice graphics
 - ggplot2
- See cheat sheets



Statistical models

- Open `3-statistical-models.rmd`
- Topics
 - Correlations
 - Regression models
 - Formula notation
 - Factors: Categorical predictors
 - Illustration of applications
 - Generalized linear model / logistic regression
 - Multilevel modelling
 - Structural equation modelling
 - Meta analysis
 - Bootstrapping



Session 4: Reproducible Research and Workflow with RMarkdown and ProjectTemplate

Introduction to R Workshop

Dr Jeromy Anglim
Deakin University
jeromy.anglim@deakin.edu.au

Introduction to R Workshop

Motivation: How to create documents?

- Types and distinctions
 - Formal Documents**
 - Journal articles, books, book chapters, theses, consulting reports, etc.
 - Informal documents**
 - preliminary analyses, statistical homework,
 - Online content**
 - web pages, blog posts, forum posts Browser metaphor versus page/slide-based metaphor
- Context
 - When to use reproducible analysis?
 - When to use knitr with R Markdown or LaTeX?

What is reproducible analysis?

- Reproducibility varies on a continuum
- One particular form:
 - code transforms raw data and meta-data into processed data,
 - code runs analyses on the data, and
 - code incorporates analyses into a report
- Ideally, the process involves a one-click build
- Public sharing of document, code, and data is optional, but forms part of gold standard of scientific openness
- Goes by many names, particularly “reproducible research”, but I prefer “reproducible analysis”.

Aims of reproducible analysis

- Ability to reproduce analysis
- Increase accuracy
 - Ability to verify analyses are consistent with intentions
 - Ability to review analysis choices
- Increase clarity of communication
- Increased trustworthiness
 - Increased accuracy +
 - Ability for others to verify
- Extensibility
 - Ability to easily modify or re-use existing analyses

Reproducible analysis in R

- Combine R and plain text file format to produce documents (e.g., pdfs, HTML documents, etc.)
- Popular instances
 - Sweave
 - knitr

What is markdown?

- Simple, readable, intuitive, light-weight markup Convert to HTML
- Raw HTML can be interspersed to add functionality
- Various extensions and flavours of markdown
- Popular on websites: e.g., StackOverflow, GitHub, Reddit
- see also: <http://daringfireball.net/projects/markdown/>

RStudio RMarkdown cheat sheet	
3. Markdown Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.	
syntax	becomes
<code># first text</code> <code># Header 1</code> <code>## Header 2</code> <code>### Header 3</code> <code>#### Header 4</code> <code>##### Header 5</code> <code>##### Header 6</code> <code>##### Header 7</code> <code>##### Header 8</code> <code>##### Header 9</code> <code>##### Header 10</code> <code>##### Header 11</code> <code>##### Header 12</code> <code>##### Header 13</code> <code>##### Header 14</code> <code>##### Header 15</code> <code>##### Header 16</code> <code>##### Header 17</code> <code>##### Header 18</code> <code>##### Header 19</code> <code>##### Header 20</code> <code>##### Header 21</code> <code>##### Header 22</code> <code>##### Header 23</code> <code>##### Header 24</code> <code>##### Header 25</code> <code>##### Header 26</code> <code>##### Header 27</code> <code>##### Header 28</code> <code>##### Header 29</code> <code>##### Header 30</code> <code>##### Header 31</code> <code>##### Header 32</code> <code>##### Header 33</code> <code>##### Header 34</code> <code>##### Header 35</code> <code>##### Header 36</code> <code>##### Header 37</code> <code>##### Header 38</code> <code>##### Header 39</code> <code>##### Header 40</code> <code>##### Header 41</code> <code>##### Header 42</code> <code>##### Header 43</code> <code>##### Header 44</code> <code>##### Header 45</code> <code>##### Header 46</code> <code>##### Header 47</code> <code>##### Header 48</code> <code>##### Header 49</code> <code>##### Header 50</code> <code>##### Header 51</code> <code>##### Header 52</code> <code>##### Header 53</code> <code>##### Header 54</code> <code>##### Header 55</code> <code>##### Header 56</code> <code>##### Header 57</code> <code>##### Header 58</code> <code>##### Header 59</code> <code>##### Header 60</code> <code>##### Header 61</code> <code>##### Header 62</code> <code>##### Header 63</code> <code>##### Header 64</code> <code>##### Header 65</code> <code>##### Header 66</code> <code>##### Header 67</code> <code>##### Header 68</code> <code>##### Header 69</code> <code>##### Header 70</code> <code>##### Header 71</code> <code>##### Header 72</code> <code>##### Header 73</code> <code>##### Header 74</code> <code>##### Header 75</code> <code>##### Header 76</code> <code>##### Header 77</code> <code>##### Header 78</code> <code>##### Header 79</code> <code>##### Header 80</code> <code>##### Header 81</code> <code>##### Header 82</code> <code>##### Header 83</code> <code>##### Header 84</code> <code>##### Header 85</code> <code>##### Header 86</code> <code>##### Header 87</code> <code>##### Header 88</code> <code>##### Header 89</code> <code>##### Header 90</code> <code>##### Header 91</code> <code>##### Header 92</code> <code>##### Header 93</code> <code>##### Header 94</code> <code>##### Header 95</code> <code>##### Header 96</code> <code>##### Header 97</code> <code>##### Header 98</code> <code>##### Header 99</code> <code>##### Header 100</code>	Header 1 Header 2 Header 3 Header 4 Header 5 Header 6 Header 7 Header 8 Header 9 Header 10 Header 11 Header 12 Header 13 Header 14 Header 15 Header 16 Header 17 Header 18 Header 19 Header 20 Header 21 Header 22 Header 23 Header 24 Header 25 Header 26 Header 27 Header 28 Header 29 Header 30 Header 31 Header 32 Header 33 Header 34 Header 35 Header 36 Header 37 Header 38 Header 39 Header 40 Header 41 Header 42 Header 43 Header 44 Header 45 Header 46 Header 47 Header 48 Header 49 Header 50 Header 51 Header 52 Header 53 Header 54 Header 55 Header 56 Header 57 Header 58 Header 59 Header 60 Header 61 Header 62 Header 63 Header 64 Header 65 Header 66 Header 67 Header 68 Header 69 Header 70 Header 71 Header 72 Header 73 Header 74 Header 75 Header 76 Header 77 Header 78 Header 79 Header 80 Header 81 Header 82 Header 83 Header 84 Header 85 Header 86 Header 87 Header 88 Header 89 Header 90 Header 91 Header 92 Header 93 Header 94 Header 95 Header 96 Header 97 Header 98 Header 99 Header 100

knitr, Rmarkdown, RStudio

- knitr**
 - R Package developed by Yihui Xie for weaving R (and other languages) with various markup languages including markdown
- RMarkdown**
 - A file format that combines R code chunks and markdown text which is converted by knitr into markdown, and other formats (e.g., HTML, pdf, etc.).
- RStudio**
 - RStudio facilitates the application of knitr to RMarkdown to produce HTML, pdfs, and so on
 - Syntax highlighting; Easy to run and create code chunks; Single button to convert RMarkdown to complete documents; useful debugging information
- pandoc**
 - General purpose tool used convert between document formats; included in Rstudio
- LaTeX**
 - If you want to be able to convert to PDF, you may need a LaTeX installation
 - <http://latex-project.org/ftp.html>

Introduction to R Workshop

Example: 1-rmarkdown-overview

- Refer to the excellent RStudio RMarkdown Cheat Sheet
- General workflow
 - Create an RMarkdown File
 - either use Rstudio File – New File – RMarkdown or just create a file with an empty text file with the .rmd extension
 - Optionally add a header
 - Options to specify output format (see RMarkdown cheat sheet)
 - Write the main document in Markdown
 - Embed R code chunks
 - R code chunks can be customised to control output

Example: 2-knitr-regression-example

- General points and RMarkdown
 - It is a good strategy to put library, options, data importing and data processing at the start of the RMarkdown file.
 - Or as we'll see later, have a single command like in ProjectTemplate that sets up your project at the start (library, options, data importing, data processing, loading scripts, etc.) of your RMarkdown document.
 - RMarkdown can be a nice way to organise analysis code even if you have no interest in compiling documents.
 - It's easy to organise code into blocks.
 - It's easy to run code in blocks
 - You can intersperse commentary

Exercise 4-1

- Create a new folder on your computer
- Create a new RMarkdown document in RStudio and save to folder
- Type a little text after the header using some headings, dot points, and any other markdown syntax you wish.
- Create a first code chunk; put the following code inside; run the code chunk
 - `library(MASS)`
 - `data(survey)`
- Add a second code chunk where you add some summary information about the `survey` dataset: e.g.,
 - `summary(survey)`
 - `head(survey)`
- Compile the document by clicking the "Knit" button
- Add a third chunk that includes some figures (e.g., a scatterplot of height and pulse)
- Compile the document by clicking the "Knit" button

3-Sweave-Item-Analysis

- This example provides an illustration of LaTeX and Sweave
- LaTeX is a document markup system designed for making beautiful scientific documents
- Pros
 - The typesetting is beautiful
 - It is excellent for mathematics
 - It handles complex documents well (e.g., PhD thesis; Scientific books): Table of contents, cross-references, references,
- Cons
 - If your collaborators and the people who consume your documents (e.g., journal editors, clients, etc.) expect a different system, then you may have an unpleasant time.
 - While it does a lot of things automatically, customising can be difficult, and there is a learning curve (a bit like R)

Example pages of PDF of LaTeX Document



Example 3-Sweave-Item-Analysis

- Both Sweave and knitr are systems that can combine LaTeX and R code chunks
- The notation for indicating R code chunks is slightly different to RMarkdown but the concepts are the same.
- The main difference is the actual LaTeX.
- Note that you don't have to use Sweave to combine R and LaTeX. You can export your images and your LaTeX tables and then manually incorporate them yourself.

Introduction to R Workshop

ProjectTemplate

- <http://projecttemplate.net/>
- Why use ProjectTemplate
 - Systematic place to store configuration and package loading settings
 - Automatically load r-script files
 - Automatically load data files stored in data directory
 - Automate running initial data manipulation code
- Installation
 - `install.project("ProjectTemplate", dep = TRUE)`

Standard Process for Creating a ProjectTemplate Project

- Create the folder structure
 - `library('ProjectTemplate')`
`create.project('myproject')`
- Review `config/global.dcf`
 - Choose settings
 - Specify packages to load
- Add data for auto-loading to data directory
- Add any additional R support functions to the `lib` directory
- Load the project
 - `library('ProjectTemplate')`
`load.project()`
- Write any initial data manipulation code and place in the `munge` directory
- Create data analysis files (e.g., r-scripts, RMarkdown, Sweave Files) in home or reports directory
 - Include the load project commands above at the top of each such file

Customise your own version of ProjectTemplate

- Once you start using ProjectTemplate, you find that there are many customisations that you always make to a new project
 - libraries that you always use
 - settings that you prefer over the defaults (e.g., `as_factors: FALSE`)
 - particular ways that you generate analysis scripts
 - Integration with RStudio project structure
- Save this customised version to a special folder on your computer
- To create a new project
 - Make a copy of your customised folder structure
 - Rename the project
 - You only need to complete the project specific customisations
- Over time you may recognise features that you want to add to your customised project

My Customised ProjectTemplate

- Basic description
 - <http://jeromyanglim.blogspot.com.au/2014/05/customising-projecttemplate-in-r.html>
- Overview of files
 - <https://github.com/jeromyanglim/AnglimModifiedProjectTemplate>
- Zip file of Template
 - <https://github.com/jeromyanglim/AnglimModifiedProjectTemplate/archive/master.zip>

Customised ProjectTemplate Workflow

- Setup ProjectTemplate Folder Structure
 - Download the zip file (I have it bookmarked) and unzip it
 - <https://github.com/jeromyanglim/AnglimModifiedProjectTemplate/archive/master.zip>
 - Rename the folder and the RStudio Project file
- Data
 - Ensure that raw data is roughly in the right format
 - Place data files in `data` folder with the names you want the `data.frames` to have in R (e.g., `mydata.csv` becomes `mydata` in R)
- Additional script files
 - Functions that get created during the project or functions that need to be imported get put in `.r` script files in the `lib` folder (e.g., "myfunctions.r")
- Data manipulation
 - Before analysing data, it is usually necessary to clean the data, create new variables, merge data, and so on.
 - This all goes in scripts in the `munge` folder.
 - Run `library("ProjectTemplate"); load.project()` to load the data and then write any data manipulation code.

Configuration

- Configuration settings are stored in `config/global.dcf`
- `data_loading`, `munging`, `load_libraries`: indicate which aspects of ProjectTemplate should run
- `libraries`: Specify which packages you want to use
- `as_factors`: Specifies whether by default strings should be imported as factors

```
1 data_loading: TRUE
2 cache_loading: FALSE
3 munging: TRUE
4 logging: FALSE
5 load_libraries: TRUE
6 libraries: personalityfacets, gdata, gtools, xtable, car, psych, GPArotation,
7   nFactors, vegan, digest, bootstrap, MASS, boot, QuantPsyc
8 as_factors: FALSE
9 data_tables: FALSE
0 version: 0.6
1 recursive_loading: FALSE
2 attach_internal_libraries: TRUE
3
```

Introduction to R Workshop

Customised ProjectTemplate Workflow

- Analyses
 - Store analyses (i.e., code to generate summary statistics, models, tables, figures, etc.) in Rmarkdown files
 - You need a code chunk before any analysis that loads the project with the following code
 - `library("ProjectTemplate"); load.project()`
 - It can be useful to have multiple RMarkdown files: e.g., for exploratory analyses, final analyses and so on.
 - If you store your RMarkdown files in a subfolder, you also need to include the following in the first code chunk:

```
library(knitr)
opts_knit$set(root.dir = normalizePath('../'))
```
 - Alternatively, just put file in the working directory

Running ProjectTemplate

- What happens when you run

```
library("ProjectTemplate"); load.project()
```

 - Configuration file is loaded
 - Options are set
 - Scripts in the `lib` file are loaded
 - Packages specified in the configuration file are loaded
 - Data in the `data` folder is loaded into R
 - Data manipulations specified in the `munge` folder are run
- The benefits
 - Thus, after running a single command you are now ready to analyse your data, or perform new analyses.

Manual integration with MS Word

- Results Text
 - Keep the code that generates the relevant script (".r", ".rmd", etc.)
 - Run the code and read off the values
 - or export the output using one of several options:
 - Running a knitr document so that input and output is displayed
 - Running the script in batch mode from the command-line: `R CMD BATCH yourfile.r`
- Tables
 - Prepare content of table in R as a data.frame
 - Save as a csv file

```
write.csv(mytable, file="output/mytable.csv")
```
 - Double click csv file to open in Excel and copy contents into a data processing Excel file
 - Use Excel to add lines, adjust alignment, etc.
 - Paste formatted table into MS Word
- Figures
 - Export figures to a folder

```
pdf(file="output/filename.pdf")
plot.code.here
dev.off()
```
 - Drag and drop image file into relevant place in Word document

4-project-template-personality-example

- Example of analyses for a journal article performed using ProjectTemplate
- Things to note
 - Metadata
 - It can be useful to import metadata (e.g., how to score tests, variable labels, etc.)
 - Variable lists
 - I often refer to large variable sets in analyses. Thus, storing these variable sets in a list can make it easier to refer to them.
 - Exporting tables, figures, plain text
 - I use RMarkdown as a way of organising analyses where specific results are exported for inclusion in another document.

Exercise 4-2

- Go to "exercise-project-template/raw-materials" unzip the Customised version of ProjectTemplate
- Give the folder and the rstudio project file an appropriate name
- Put `cas.sav` into the `data` folder (California Schools Data)
- Open the Rstudio project file in RStudio
- Open "reports/explore.rmd" and run

```
library(ProjectTemplate); load.project()
```
- Add a few basic analyses of `cas` to the next R code chunk
- Go to "munge/01-munge.R" and add a new variable to `cas` (e.g., create a variable called `performance` which is the sum of `cas$math` and `cas$english`)
- Return to "reports/explore.rmd" and add another code chunk. Create a histogram of `cas$performance`.
- Now imagine that you are exiting RStudio and then returning again. i.e., Quit RStudio and then reload the Rstudio Project file
- Open "reports/explore.rmd" and run

```
library(ProjectTemplate); load.project()
```
- You should see that your histogram code for `cas$performance` still runs



Concluding Remarks

Introduction to R Workshop

Dr Jeromy Anglim
Deakin University
jeromy.anglim@deakin.edu.au

Introduction to R Workshop

Reflections on the day

- The aim of this training has been to get you oriented with R.
- By installing R, RStudio, and related packages you overcame the first obstacle.
- R Commander provides a basic starting point
 - It's a good option if you're used to SPSS-type tools but need a free option.
 - But ultimately, the power of R comes from learning how to write R scripts.
- We have gone over the nuts and bolts of the language
- Data manipulation and graphics are common tasks across all data analysis projects
- We have illustrated R's functionality using common statistical models.
- We've also highlighted a number of good options for managing an R workflow

Future Use of R

- Hopefully, at this point you have a sense of whether R might be something you can use. Perhaps one of these use cases will fit:
 - You may realise that **R is not for you**. It's not for everyone.
 - You may be in an environment, where you don't have access to paid statistics packages and **you need a free tool**.
 - R may be a **tool that you occasionally use** when you need particularly functionality not offered by your regular statistics package.
 - R can be **your primary statistics package** for data analysis.
- R is a tool that becomes more useful over time.
 - You acquire scripts that you can re-use.
 - Learning new statistical models becomes easier over time as you learn the conventions in R.
- R ultimately makes data analysis more fun.

Next Steps

- If you are keen to make R your main platform for data analysis, then there are lots of options for learning more.
 - Start applying R to a project that you are working on
 - Have a read through some of the contributed documentation
 - <http://cran.r-project.org/other-docs.html>
 - Get a book on R relevant to your area
 - <http://www.r-project.org/doc/bib/R-books.html>
 - Check out video tutorials on Coursera and YouTube
 - Follow recent developments in R
 - <https://twitter.com/hashtag/rstats?src=hash>
 - <http://www.r-bloggers.com>

Thank you for coming...

- Please consider filling out the Workshop Feedback form