



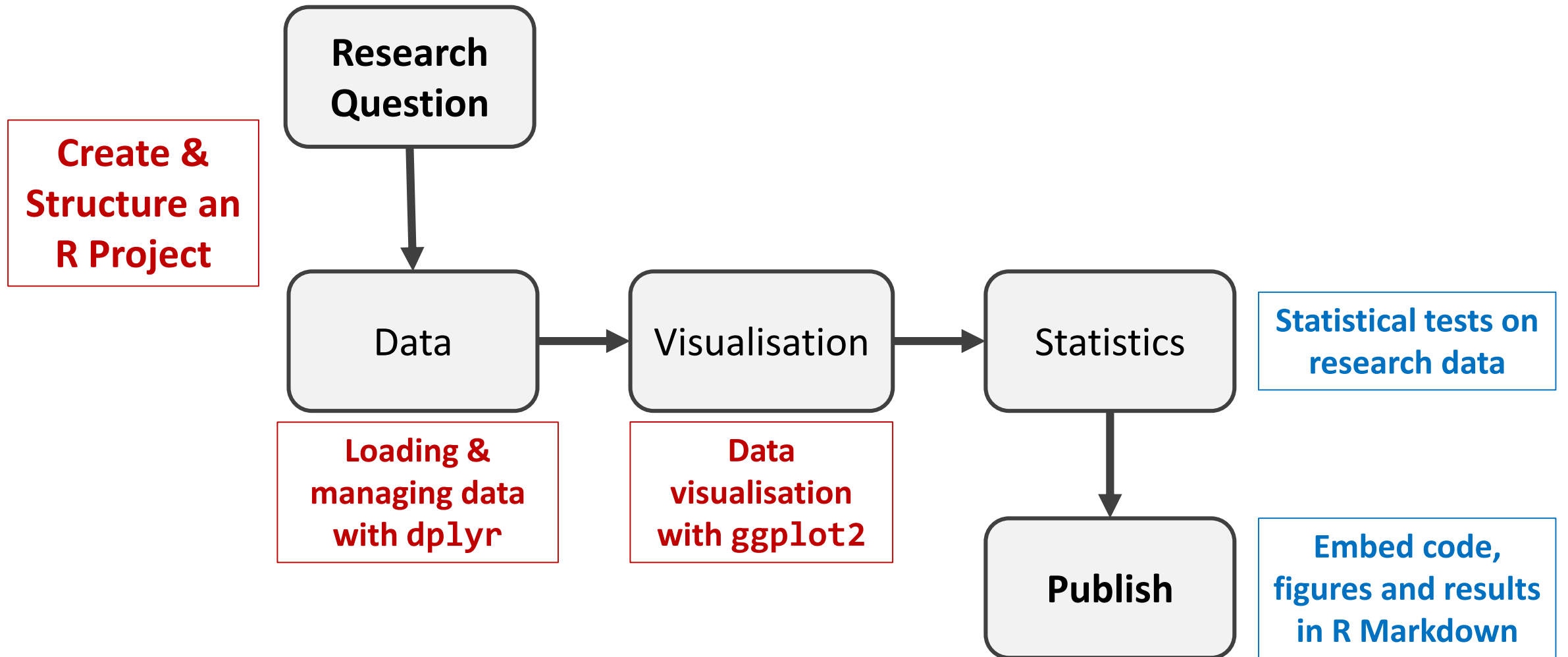
Using R as a Research Tool.

NERC E3 DTP Training – 17th November 2020

Dr Susan Johnston, Institute of Evolutionary Biology

Demonstrator: Lucy Peters

Using R as a Research Tool: Overview



Today's session:

- Carry out basic statistics and visualisations, including:
 - Chi-squared test with `chisq.test()`
 - 2-sample t-test with `t.test()`
 - Linear regression with `lm()`
- Write, embed and render code and results into an HTML document.

Report writing

Using R as a Research Tool.

Dr Susan Johnston: Susan.Johnston@ed.ac.uk

Demonstrators: Gergana Dalaskova, John Godlee.
Hat-Tips to Kyle Dexter, The Coding Club and R4all.

November 6, 2017

1 Introduction

1.1 What is R?

R began its life in New Zealand in 1993 as a language and environment for statistical computing and graphics. It is an interpreted programming language, meaning that rather than pointing and clicking, the user types in commands. It is **free** and works across all platforms.

1.2 Why use R?

R Base Graphics: An Idiot's Guide

Comments (-)

Share

Hide Toolbars

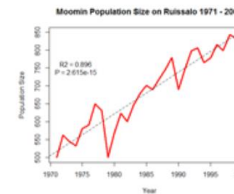
One of the most powerful functions of R is its ability to produce a wide range of graphics to quickly and easily visualise data. Plots can be replicated, modified and even publishable with just a handful of commands.

Making the leap from chiefly graphical programmes, such as Excel and Sigmaplot, may seem tricky. However, with a basic knowledge of R, just investing a few hours could completely revolutionise your data visualisation and workflow. Trust me - it's worth it.

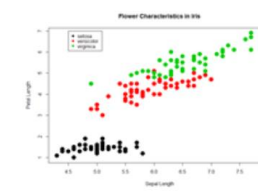
Last year, I presented an informal course on the basics of R Graphics University of Turku. In this blog post, I am providing some of the slides and the full code from that practical, which shows how to build different plot types using the basic (i.e. pre-installed) graphics in R, including:



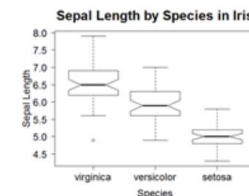
1. Basic Histogram



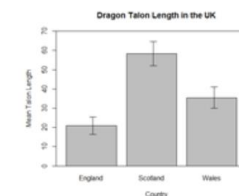
2. Line Graph with Regression



3. Scatterplot with Legend



4. Boxplot with reordered/
formatted axes



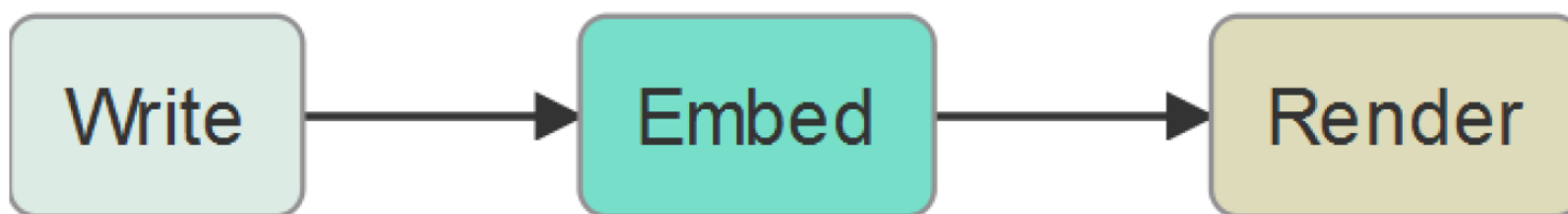
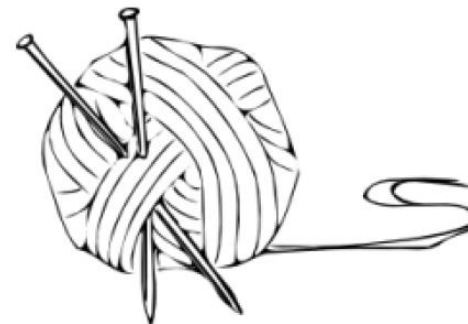
5. Boxplot with Error Bars

LaTeX and R Sweave

knitr to HTML

knitr

Elegant, flexible and fast
dynamic report generation with R



The knitr package allows R code and document templates to be compiled into a single report containing text, results and figures.

Markdown Language

- Lightweight mark-up language
- Basic text formatting, adding images, creating lists, etc.
- Can embed code and results, making it useful for reproducible research.
- Looks very much like regular text with a few extra characters (#, *, etc)

FilesPlotsPackagesHelpViewer

←→🏠📄

🔍

🔄

Markdown Quick Reference ▾Find in Topic

Markdown Quick Reference

R Markdown is an easy-to-write plain text format for creating dynamic documents and reports. See [Using R Markdown](#) to learn more.

Emphasis

```
*italic*  **bold**  
_italic_  __bold__
```

Headers

```
# Header 1  
## Header 2  
### Header 3
```

Lists

Unordered List

```
* Item 1  
* Item 2  
  + Item 2a  
  + Item 2b
```

Ordered List

```
1. Item 1  
2. Item 2  
3. Item 3  
  + Item 3a  
  + Item 3b
```

R Markdown :: CHEAT SHEET

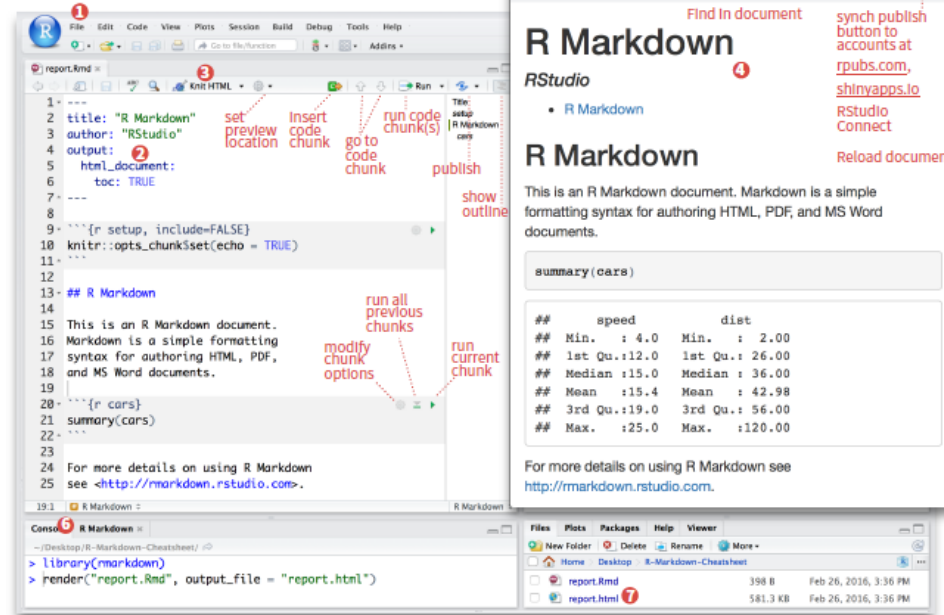
What is R Markdown?

- .Rmd files** - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.
- Reproducible Research** - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.
- Dynamic Documents** - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow



- 1 Open a new .Rmd file at File ► New File ► R Markdown. Use the wizard that opens to pre-populate the file with a template
- 2 Write document by editing template
- 3 Knit document to create report; use knit button or render() to knit
- 4 Preview Output in IDE window
- 5 Publish (optional) to web server
- 6 Examine build log in R Markdown console
- 7 Use output file that is saved along side .Rmd



render

Use `rmarkdown::render()` to render/knit at cmd line. Important args:

input - file to render	output_options - List of render options (as in YAML)	output_file - output file	output_dir - output dir	params - list of params to use	envir - environment to evaluate code chunks in	encoding - of input file
-------------------------------	---	----------------------------------	--------------------------------	---------------------------------------	---	---------------------------------

Embed code with knitr syntax

INLINE CODE

Insert with ``r <code>``. Results appear as text without code.
Built with ``r getRversion()`` → Built with 3.2.3

CODE CHUNKS

One or more lines surrounded with ````{r}` and `````. Place chunk options within curly braces, after `r`. Insert with `[+]`.

```
```{r echo=TRUE}
getRversion()
```
```

GLOBAL OPTIONS

Set with `knitr::opts_chunk$set()`, e.g.

```
knitr::opts_chunk$set(echo = TRUE)
```

IMPORTANT CHUNK OPTIONS

cache - cache results for future knits (default = FALSE)
cache.path - directory to save cached results in (default = "cache/")
child - file(s) to knit and then Include (default = NULL)
collapse - collapse all output into single block (default = FALSE)
comment - prefix for each line of results (default = "##")

dependson - chunk dependencies for caching (default = NULL)
echo - Display code in output document (default = TRUE)
engine - code language used in chunk (default = 'R')
error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)
eval - Run code in chunk (default = TRUE)

fig.align - 'left', 'right', or 'center' (default = 'default')
fig.cap - figure caption as character string (default = NULL)
fig.height, **fig.width** - Dimensions of plots in inches
highlight - highlight source code (default = TRUE)
include - Include chunk in doc after running (default = TRUE)

message - display code messages in document (default = TRUE)
results (default = 'markup')
 'asis' - passthrough results
 'hide' - do not display results
 'hold' - put all results below all code
tidy - tidy code for display (default = FALSE)
warning - display code warnings in document (default = TRUE)

Options not listed above: `R.options`, `aniloops`, `autodep`, `background`, `cache.comments`, `cache.lazy`, `cache.rebuild`, `cache.vars`, `dev`, `dev.args`, `dpl`, `engine.opts`, `engine.path`, `fig.asp`, `fig.env`, `fig.ext`, `fig.keep`, `fig.ip`, `fig.path`, `fig.pos`, `fig.process`, `fig.retina`, `fig.scap`, `fig.show`, `fig.showtext`, `fig.subcap`, `interval`, `out.extra`, `out.height`, `out.width`, `prompt`, `purrr`, `ref.label`, `render`, `size`, `split`, `tidy.opts`



Worked example...

Today's session:

- Carry out basic statistics and visualisations, including:
 - Chi-squared test with `chisq.test()`
 - 2-sample t-test with `t.test()`
 - Linear regression with `lm()`
- Write, embed and render code and results into an HTML document.

Chi-squared (χ^2) contingency table

- Analysis of count data

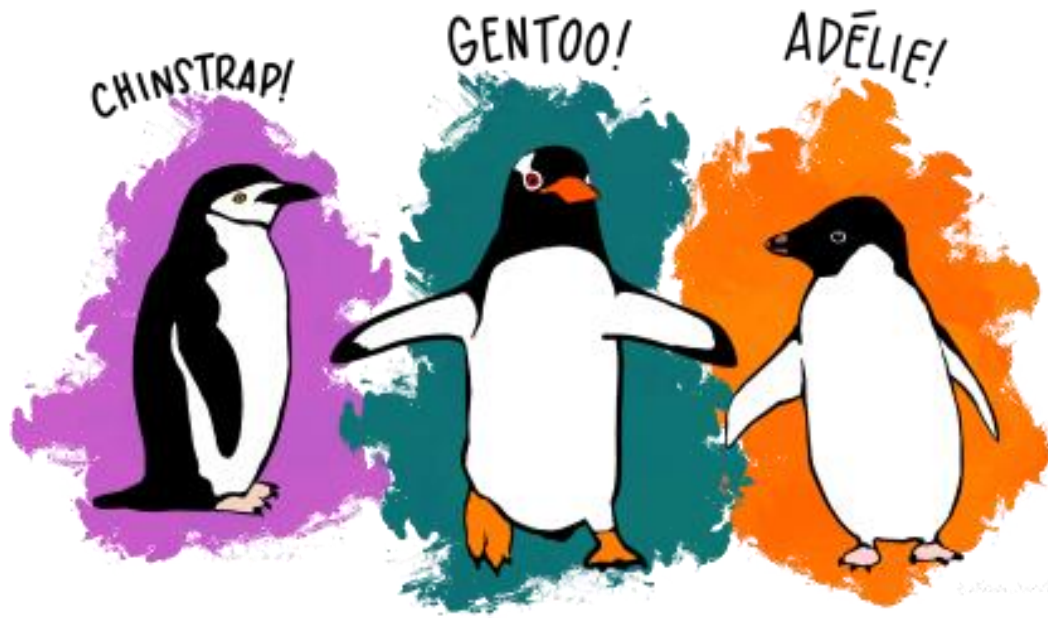


Adalia bipunctata

Are dark ladybirds more likely to live in industrial (dark) backgrounds?

Two sample t-test

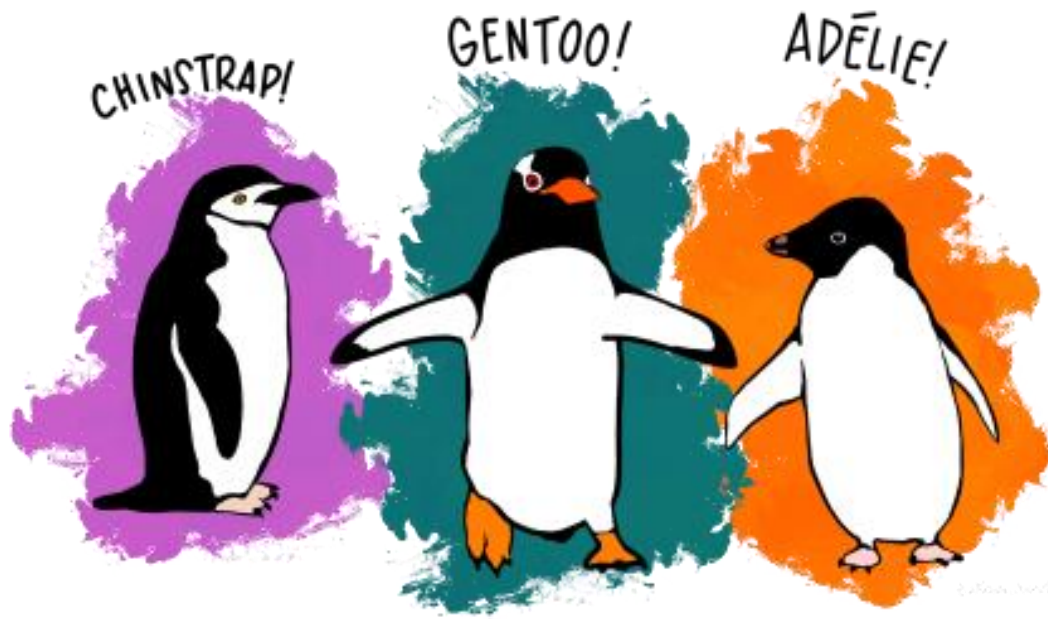
- Compares the mean of two groups.



Does body weight differ significantly between Adélie and Gentoo penguins?

Simple linear regression

- Summarise and study relationships between two continuous variables.



Does flipper length vary relative to body weight in penguins?

Need help?

- **ASK US!**
- Use `?`, `str()`, `glimpse()` to explore functions and objects.
- Coding Club tutorials: <https://ourcodingclub.github.io/tutorials/>
- Google & StackOverflow
- Feel free to write as a normal R script!

