

An Adventure in Statistics

Andy Field

University of Sussex

2019-09-25

ANDY FIELD



Key information

Teaching

- 1 × weekly lecture (2 in weeks 1 and 2) on theory
- 1 × weekly practical class (2 in weeks 1 and 2) about R/ Rstudio

Assessment

- 2 × 24-hour Take Away Papers (TAPs)
 - TAP 1: 30%
 - TAP 2: 30%
- 1 × Report (40%)



Part 1: Introducing R and RStudio





R Console

/Volumes/Alpha Lacertae/Documents/Academic/data/
[REDACTED] RESTORED FROM /Users/andyfield/.rapp.rhistory

```
> library(tidyverse)
— Attaching packages
tidyverse 1.2.1 —
✓ ggplot2 3.2.1    ✓ purrr  0.3.2
✓ tibble  2.1.3    ✓ dplyr   0.8.3
✓ tidyr   0.8.3    ✓ stringr 1.4.0
✓ readr   1.3.1    ✓forcats 0.4.0
— Conflicts
tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()   masks stats::lag()
> library(here)
here() starts at /Users/andyfield
> getwd()
[1] "/Volumes/Alpha Lacertae/Documents/Academic/data/teaching_data/ais_data"
> zombie_tib <- readr::read_csv("ais_14_zombie_taser.csv")
Parsed with column specification:
cols(
  id = col_double(),
  immobility = col_double(),
  r_tms = col_double(),
  taser = col_double()
)
> |
```







Studio®

ANDY FIELD





R and R Studio

R

- Free software environment for statistical analysis and graphics

R Studio

- A free integrated development environment (IDE) for R
- You use R Studio as a way to interact with R
 - Install R and forget about it until such time that you need to update it
 - Install RStudio
 - Set up RStudio to suit your personality and enjoy your status as rad data scientist hipster



Why R?

Transferable skills

R is the most widely used data analysis software

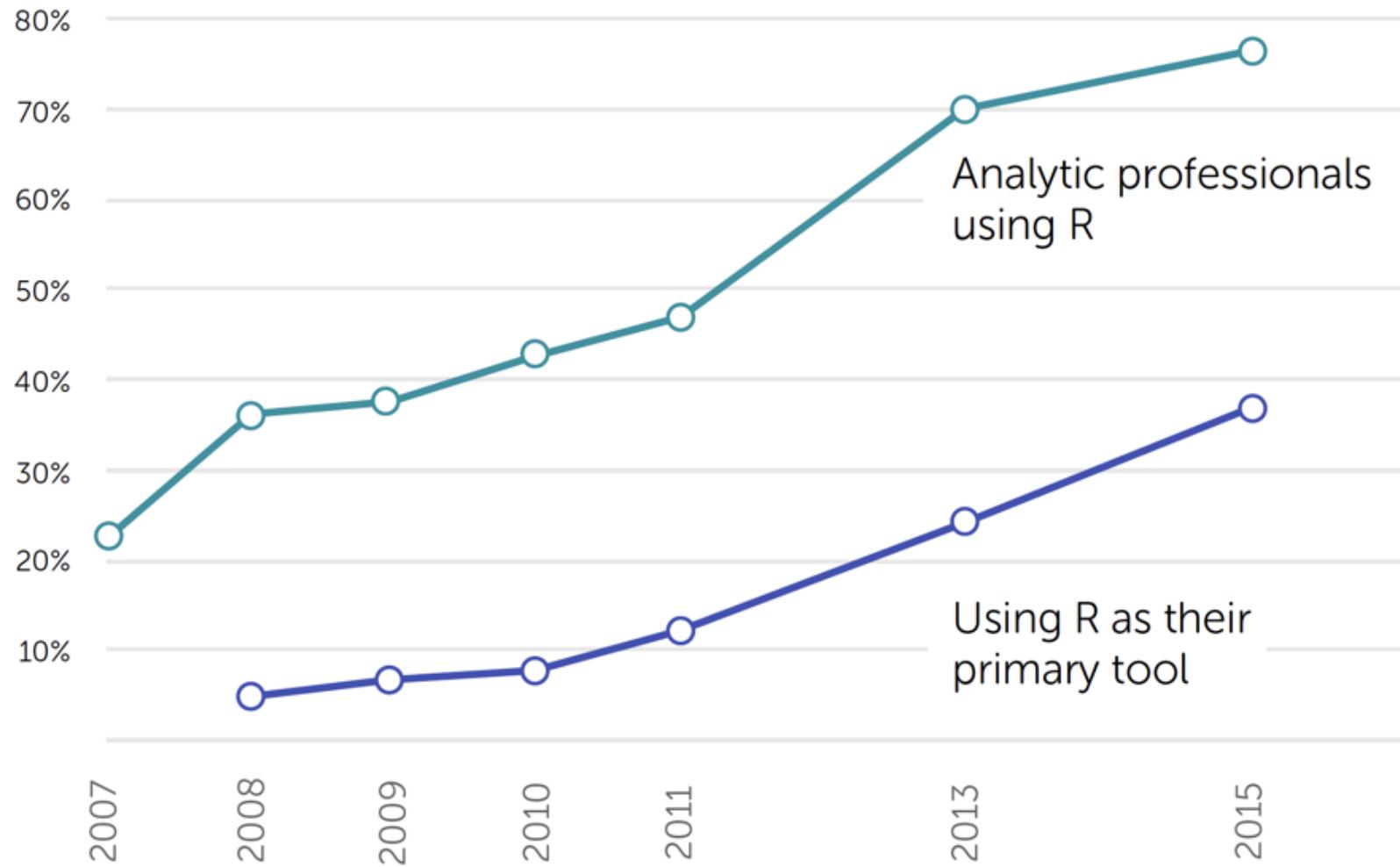
Reproducible science

Cutting edge

One stop shop

RStudio is amazeballs

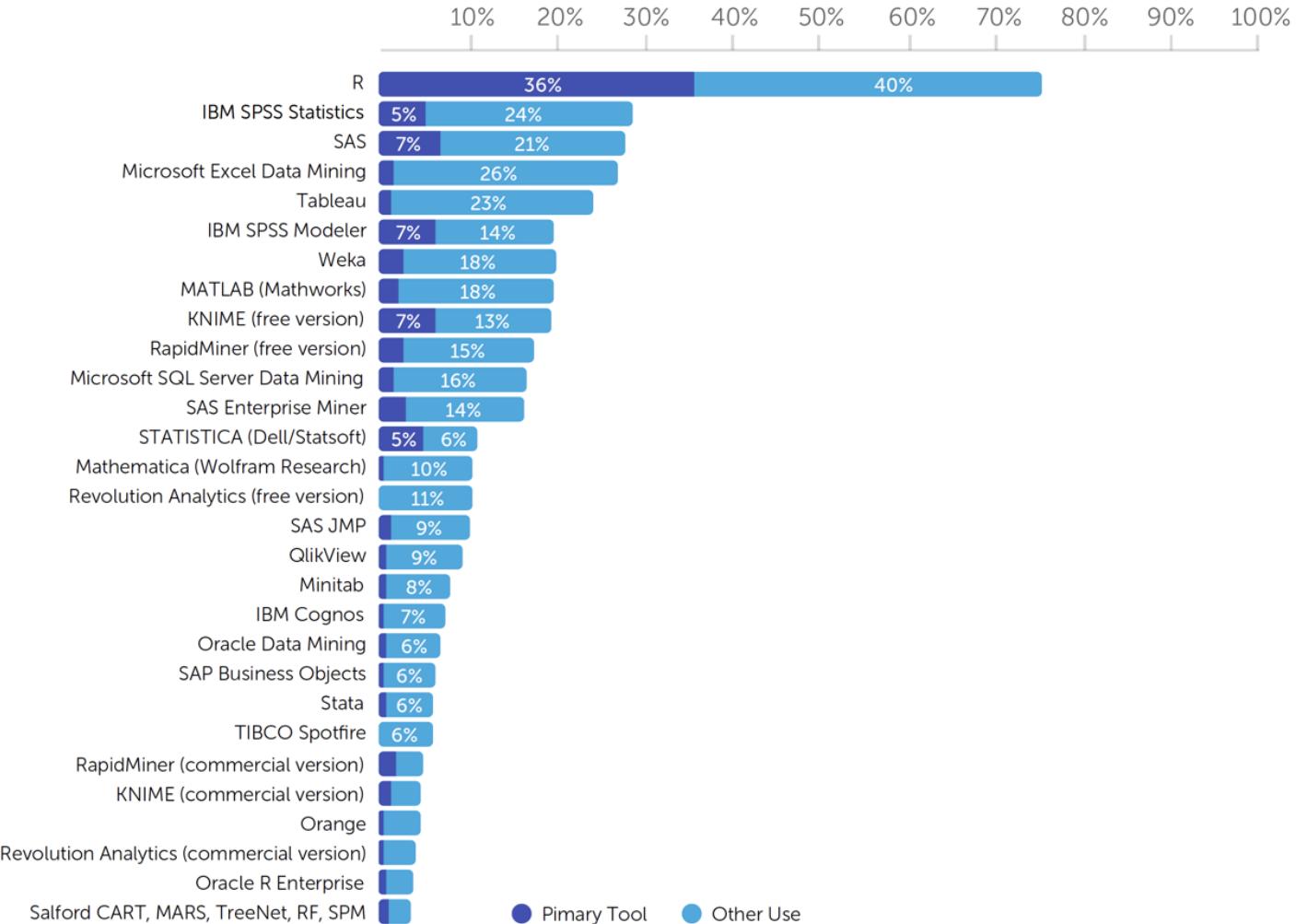




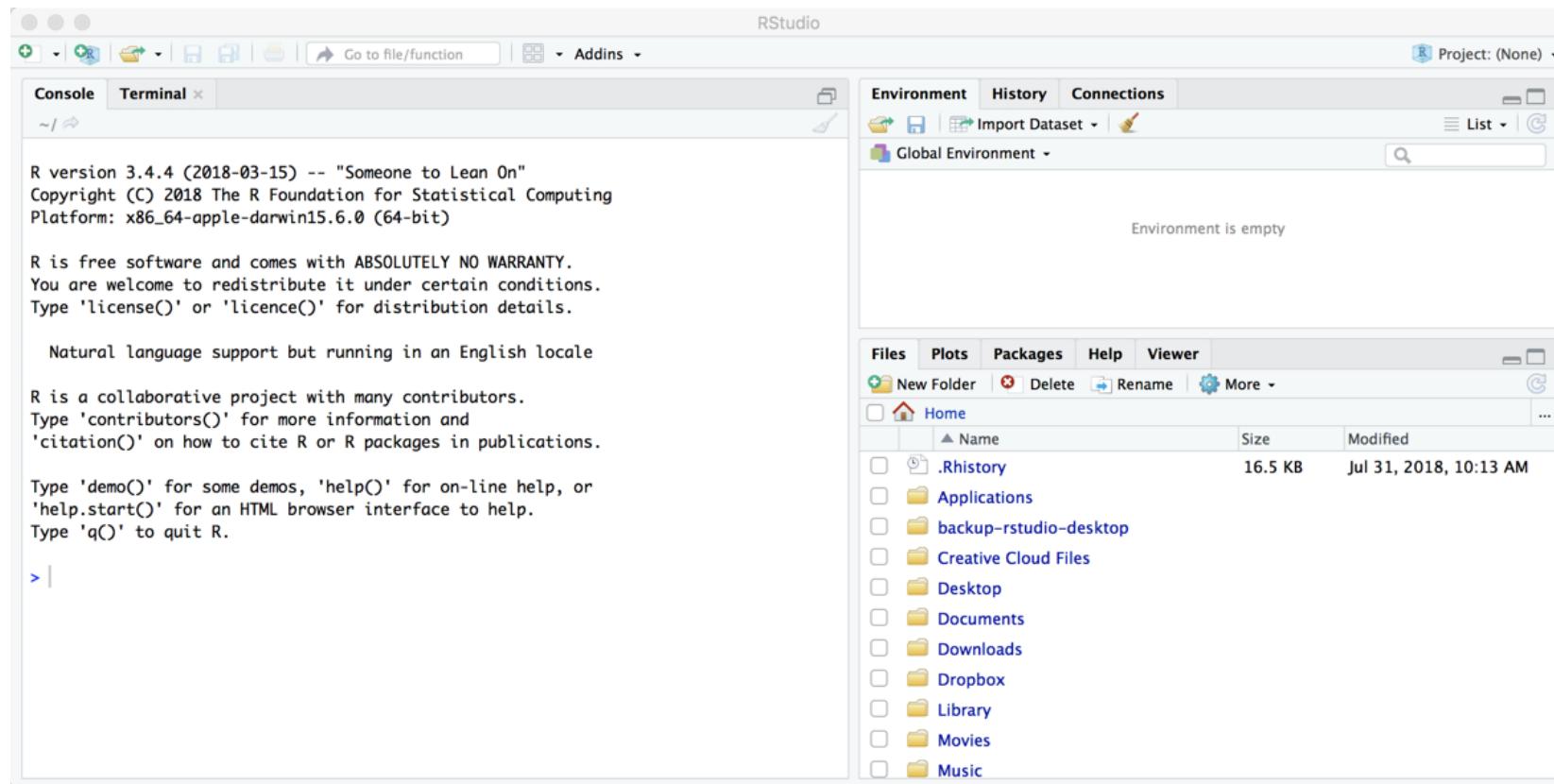
Analytic professionals
using R

Using R as their
primary tool



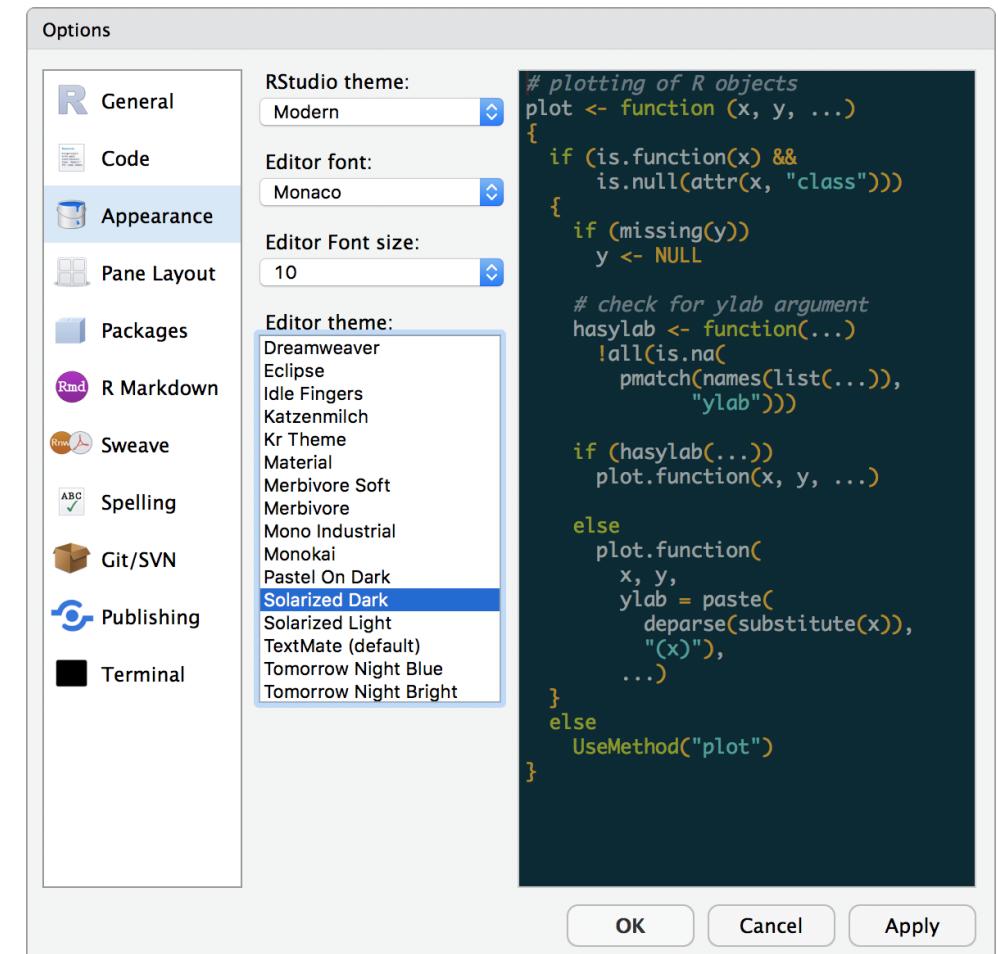


Starting up R Studio

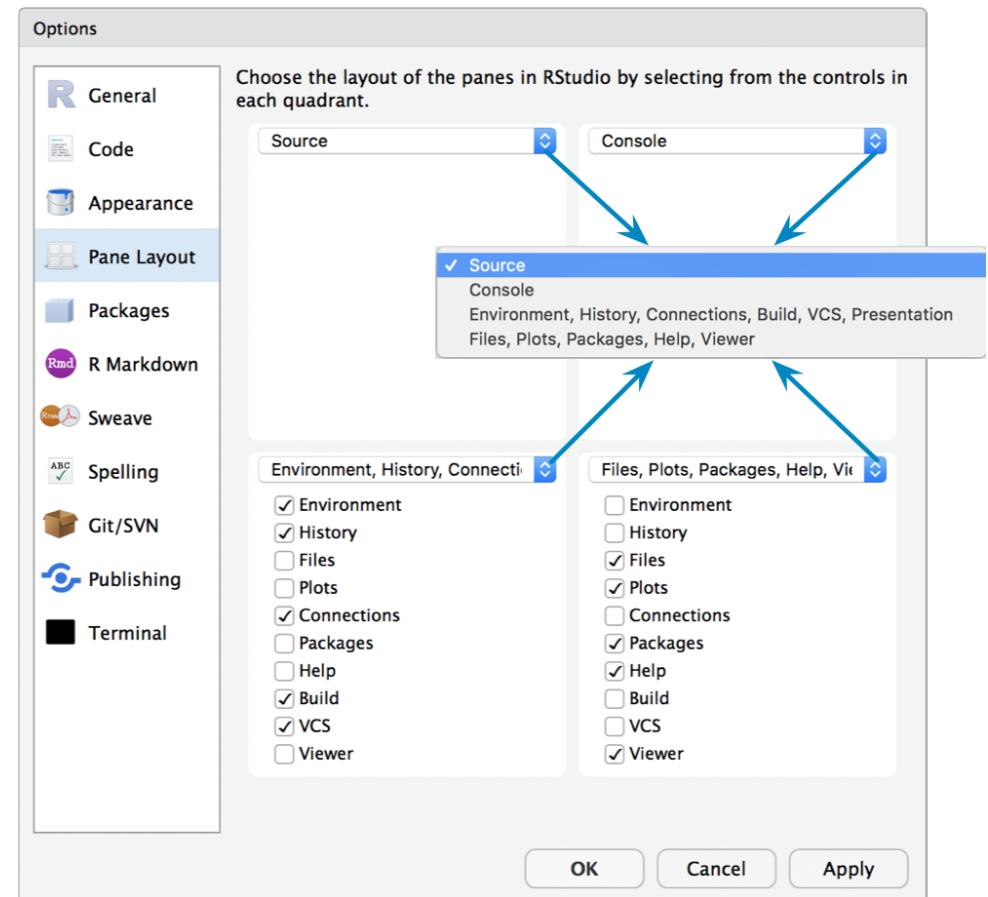


Try it! Colour scheme

- Windows
 - Tools > Options
- MacOS
 - Tools > Global Options
 - RStudio > Preferences



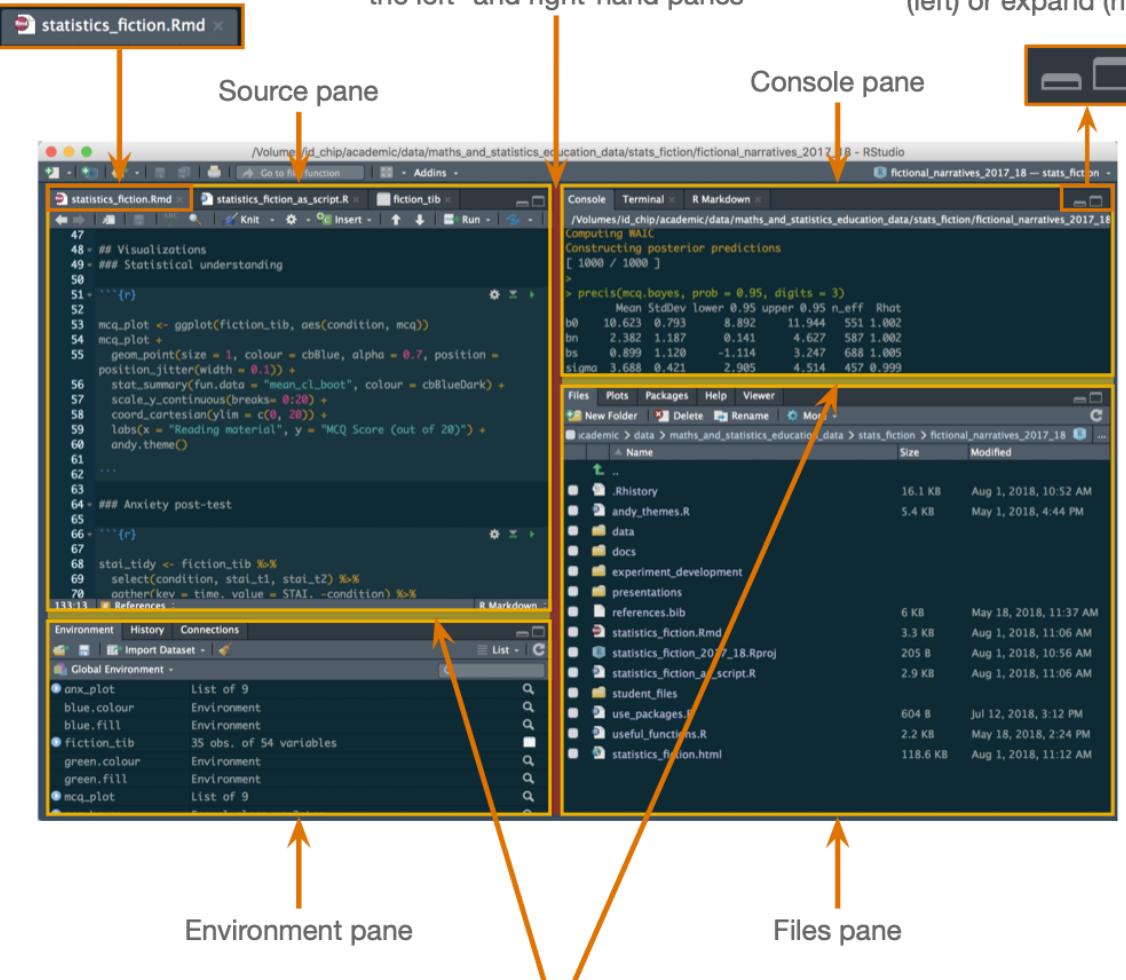
Try it! Pane locations



Each pane contains tabs

Click in the red region and drag left or right to adjust the relative widths of the left- and right-hand panes

Click these icons to minimize (left) or expand (right) a pane



Click in the yellow regions and drag up or down to adjust the relative heights of the top and bottom panes on each side



Part 2: Workflow in RStudio



Before we begin

University computers are networked and use something called UNC paths

This messes a few things up

There's a fix:

- On CANVAS go to **Files > system_files**
- Download the file **.Renvironment**
- Save it to the Documents folder on the N: drive

NOTE: you don't need to do this on your home machine/laptop





ANDY FIELD



R Studio project files

A file created by R Studio with the extension **.Rproj**

- Stores information about the containing folder
- Restores the previous state of the project (i.e. what documents/tabs were open)

Opening a project file sets the working directory to the folder containing the project file

- You can use relative file paths
- The project will work on any machine you care to use
- You can share your project folder with others and it'll work for them

Use project files!

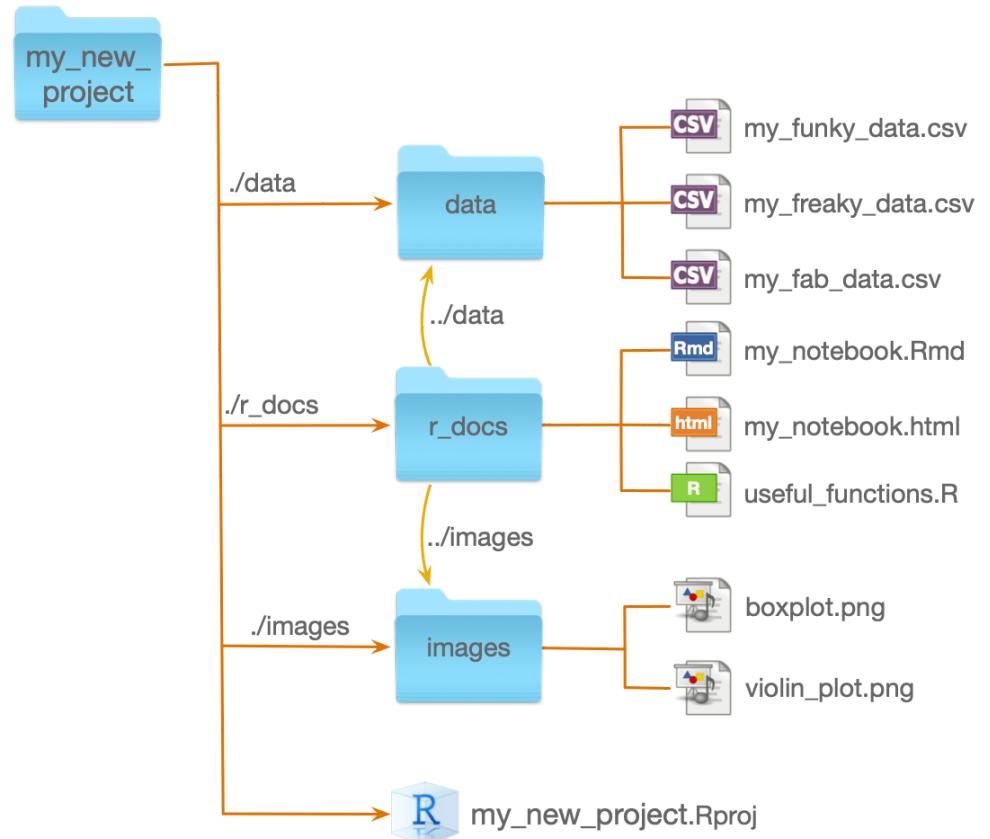


Get organized!

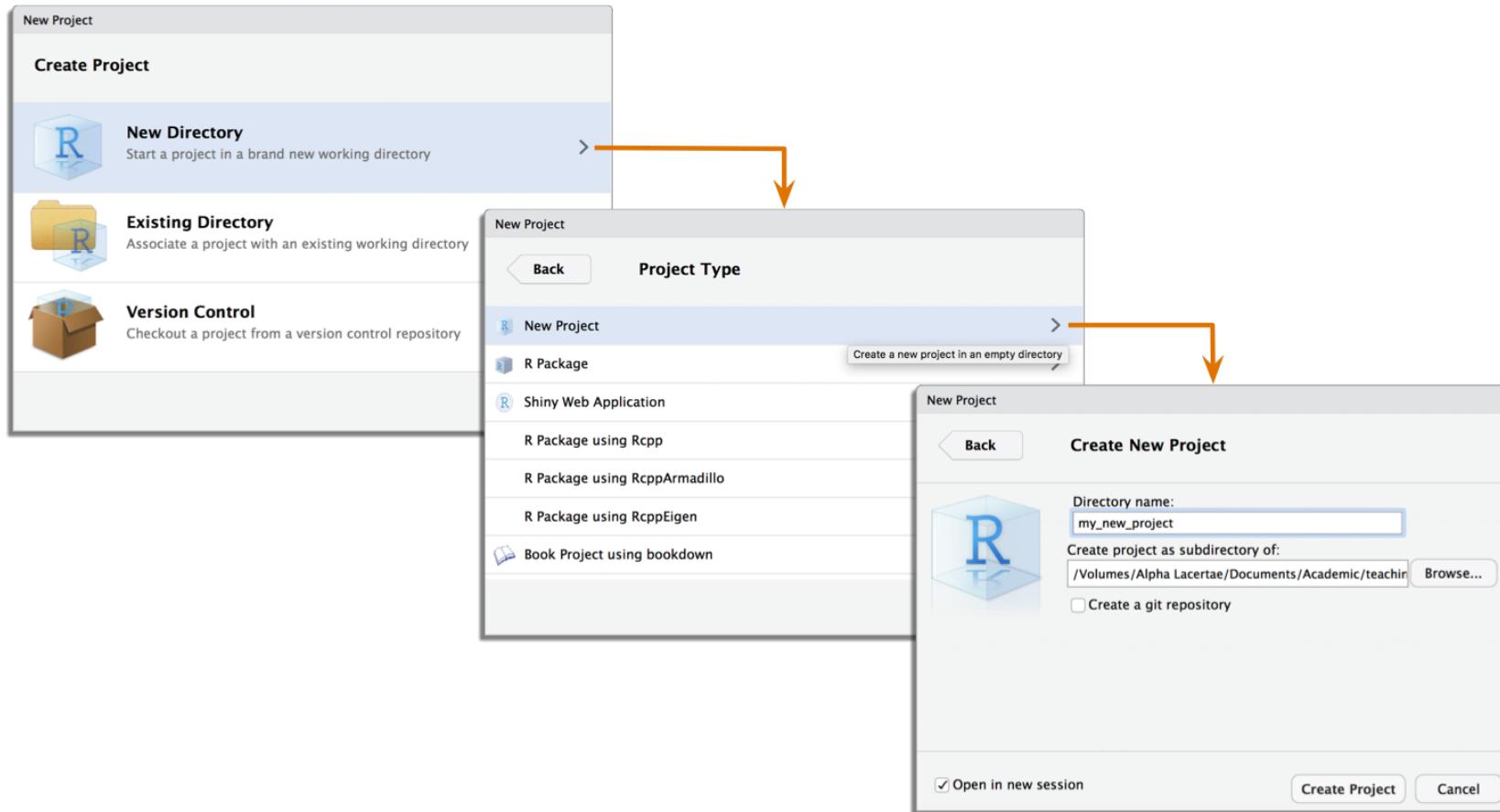
C:/Users/andyfield/Documents/my_new_project/data/my_funky_data.csv

./data/my_funky_data.csv

../data/my_funky_data.csv



Creating a project: File > New Project



Try it!

Create an R Studio project called **my_adventr** on Sussex One Drive

Within it create folders called

- **data**
- **r_docs**

Create a markdown file

- Use the **File > New File > R Markdown ...** menu
- Save it to the **r_docs** folder with the name **sample_tap.Rmd**



Now, let's save some data

On CANVAS go to **Files > data_files**

Download the sample TAP data ([tap_parenting.csv](#))

Copy the file into the data folder of your project



Part 3: Interacting with R



Interacting with R

Direct to the console (boo!)

- Bad for reproducibility/your sanity

Script files (meh!)

- Great for reproducibility
- Bad for integrating analysis into documents

R markdown document (hooray!)

- Great for reproducibility
- Great for integrating analysis into documents
- Great for impressing your friends and loved ones



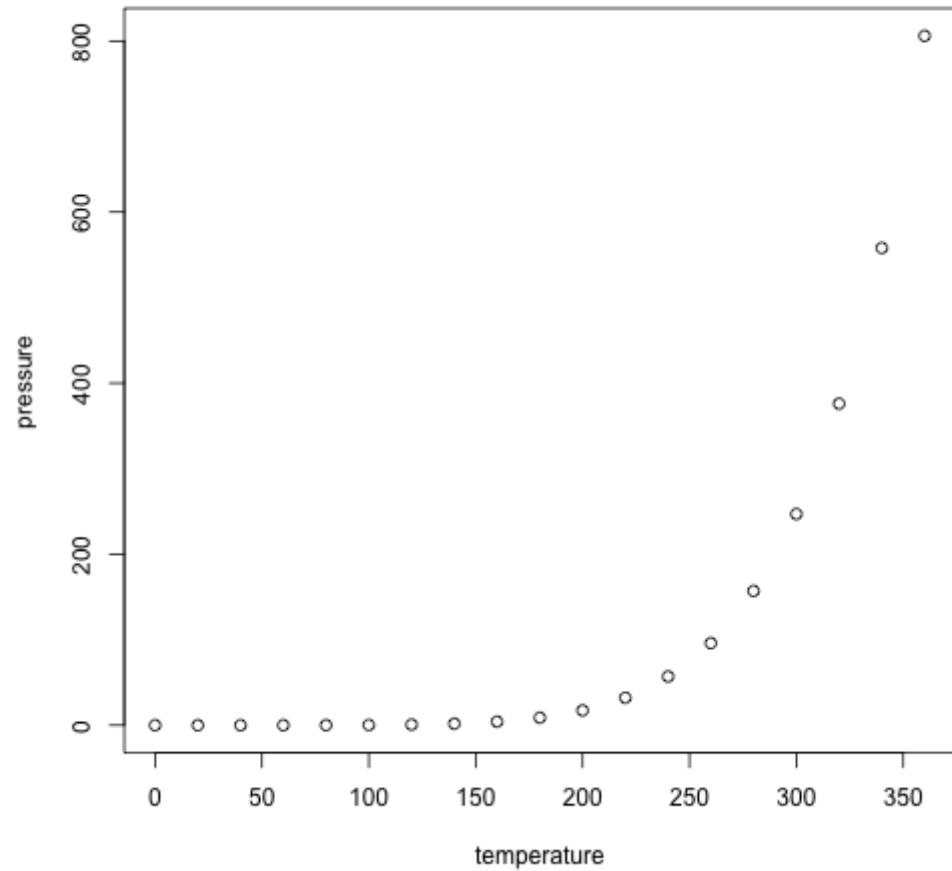
Try it

Type these commands into the console one-by-one and press return

```
plot(pressure)  
cor(pressure)
```



```
plot(pressure)
```



```
cor(pressure)
```

```
##           temperature  pressure
## temperature   1.000000  0.7577923
## pressure      0.7577923 1.0000000
```

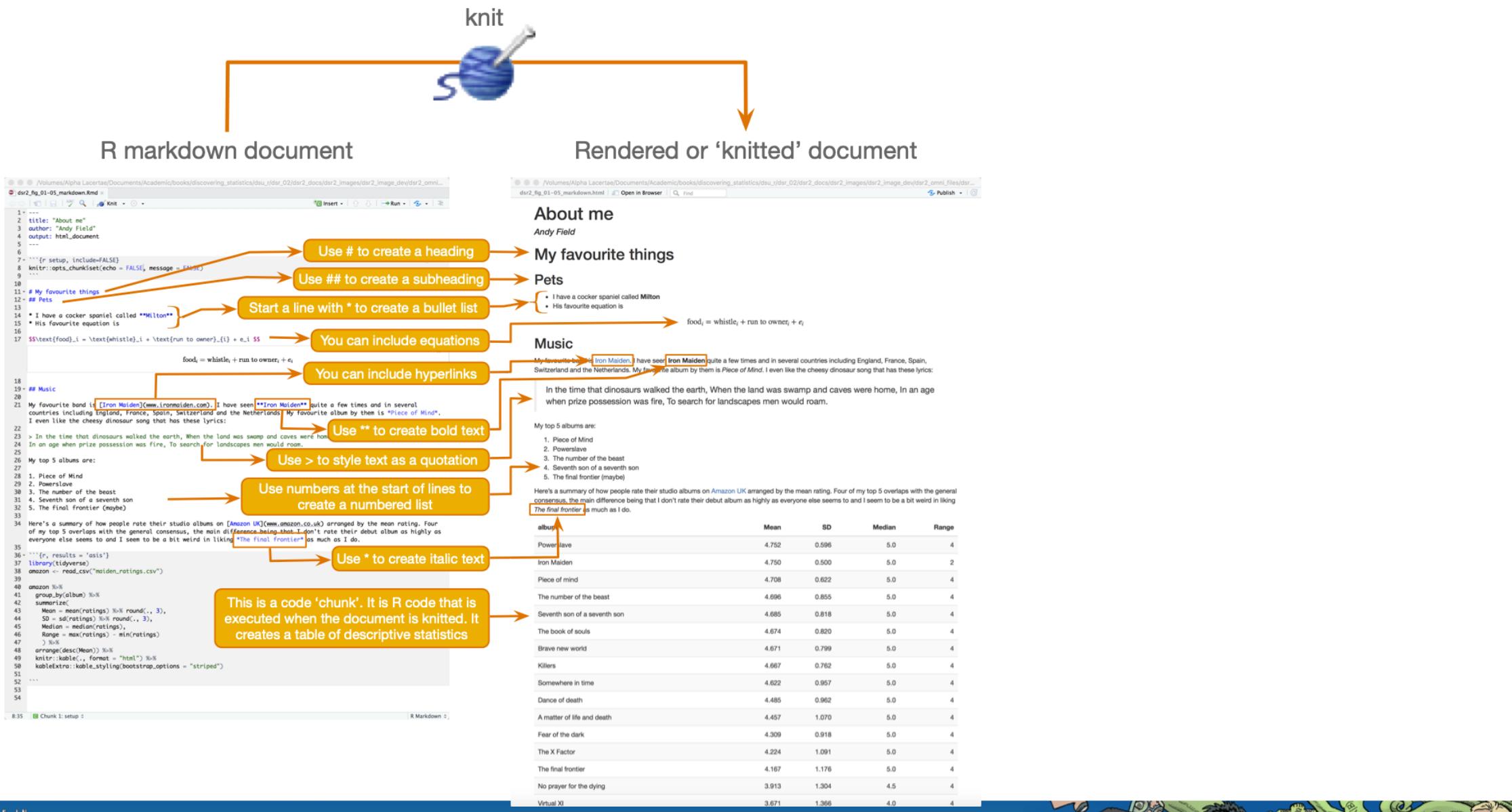




www.rstudio.com

ANDY FIELD





This is a markdown file

This line is the chunk header

I've named this code chunk 'setup'

Options for the chunk

Click here to execute the code within the chunk

This is a code chunk. All R code goes in these chunks

This is R markdown

```
my_markdown.Rmd
```

```
1 ---  
2 title: "My first markdown document"  
3 author: "Andy Field"  
4 date: `r format(Sys.Date(), "%d %B %Y")`  
5 output: html_document  
---  
8 ``{r setup, echo = FALSE}  
9 knitr::opts_chunk$set(warning=FALSE, message=FALSE)  
10  
11 library(here)  
12 library(tidyverse)  
13  
14 funky_tib <- read_csv("../data/my_funky_data.csv")  
15  
16  
17 # Introduction to the project  
18  
19 In this project I am going to use the power of data to bend  
spacetime in such a way that I can bring my dog back to life  
because I miss him. Then I'm going to go back to Long Beach  
Arena in 1984 and watch Iron Maiden on the *World Slavery  
Tour*.
```



Inserting code chunks

Windows

- *ctrl + alt + i*
- **Code > Insert Chunk**

Mac

- *cmd + opt + i* ($\text{⌘} + \text{⌥} + \text{i}$)
- **Code > Insert Chunk**



Try it

In your *sample_tap.Rmd* file

- Create a code chunk, type the first command into it and execute.

```
plot(pressure)
```

- Create a new code chunk, type the second command into it and execute.

```
cor(pressure)
```

- ‘Knit’ the document



Code chunk options

- `echo=FALSE`: the code will be evaluated but not reproduced in the knitted document.
- `eval=FALSE`: the code will not be evaluated.
- `include=FALSE`: RStudio will evaluate the code but neither the code nor output are displayed in the knitted document.
- `results="hide"`: the code will be evaluated and displayed in the knitted document but the output will be omitted.
- Include `fig.width=9` and `fig.height=5` to set the dimensions of an image
 - Obviously replace 9 and 5 with the values you want
- Use `results = 'asis'` when creating tables from code/output



Try it!

- Using the code chunks we have already created
- Change the code chunk headers to include `echo=FALSE` , re-knit the document and compare the result.
- Repeat this process but including `eval=FALSE`



R Markdown: headers

Markdown

- # Level 1 heading
- ## Level 2 heading
- ### Level 3 heading
- ##### Level 4 heading

Knitted text

- **Level 1 heading**
- **Level 2 heading**
- **Level 3 heading**
- **Level 4 heading**



R Markdown: lists

Markdown

```
* This is the first bullet point
  + this is a sub-bullet
  + so is this
* This is the second bullet
  + This is a sub-bullet
    - A third layer of bullet madness
    - It had to be done
* This is the third main bullet
```

Knitted text

- This is the first bullet point
 - this is a sub-bullet
 - so is this
- This is the second bullet
 - This is a sub-bullet
 - A third layer of bullet madness
 - It had to be done
- and this is the third bullet



R Markdown: emphasizing text

Markdown

```
*italic text*
**bold text**
Textsuperscript
Textsubscript
```

Knitted text

italic text

bold text

Text^{superscript}

Text_{subscript}



R Markdown: images

![Caption for image](filepath_for_image)

![Figure 1: my lovely spaniel]
(images/milton_circle_2019.png)



R Markdown: hyperlinks

[text_describing_link](url)

Markdown

The `*adentr*` is a package of interactive tutorials for learning R. Get it from
[milton-the-cat.rocks/home/adentr.html](<http://milton-the-cat.rocks/home/adentr.html>)

Knitted text

The `adentr` is a package of interactive tutorials for learning R. Get it from `milton-the-cat.rocks/home/adentr.html`



R Markdown: equations

Markdown

We can include the linear model in its own paragraph like this:

```
## Y_i = b_0 + b_1X_i + \epsilon_i ##
```

Knitted text

We can include the linear model in its own paragraph like this:

$$Y_i = b_0 + b_1X_i + \epsilon_i$$



Try it!

In your markdown file create a level 1 header that reads 'About me'

Write a sentence or two about yourself using bold and italic somewhere

Create a level two heading 'My favourite things'

Within that section write a list of five of your favoutite things (could be books, movies, bands, songs, statisticians). Include a URL to one of them.



Part 4: Doing things in R



Getting help

To get help, use the `help()` function or `?`

- `help(thing_you_want_help_with)`
- `?thing_you_want_help_with`

Execute help commands at the command line

- Do NOT include `help()` in markdown files

Try accessing the help files for the `mean()` function, by executing:

- `?mean`



R Style Wickham (2014)

R is case sensitive. In code chunks use lower case wherever possible

Variable and function names should be lowercase with an underscore (_) to separate words

- Names should be concise and meaningful
- A variable representing children's' anxiety levels might be named `child_anxiety`
- `scores_on_the_child_manifest_anxiety_scale` is meaningful but too long, and `ca` is concise but not meaningful

Place spaces around all operators (`=`, `+`, `-`, `<-`) to make code easier on the eye!

My practice: use consistent suffixes that identify types of objects

- `_tib` to denote a tibble (more them later), e.g. `anx_tib` for a tibble of data relating to anxiety
- `_mod` to denote a model (e.g., `anx_mod` for a model in which child anxiety is predicted)
- `_out` to denote output (e.g., `anx_out` contains the summary output from the above model)



Functions and objects

```
metallica <- c("Lars", "James", "Jason", "Kirk")
metallica <- metallica[metallica != "Jason"]
metallica <- c(metallica, "Rob")
```

object

instructions

Assignment
operator

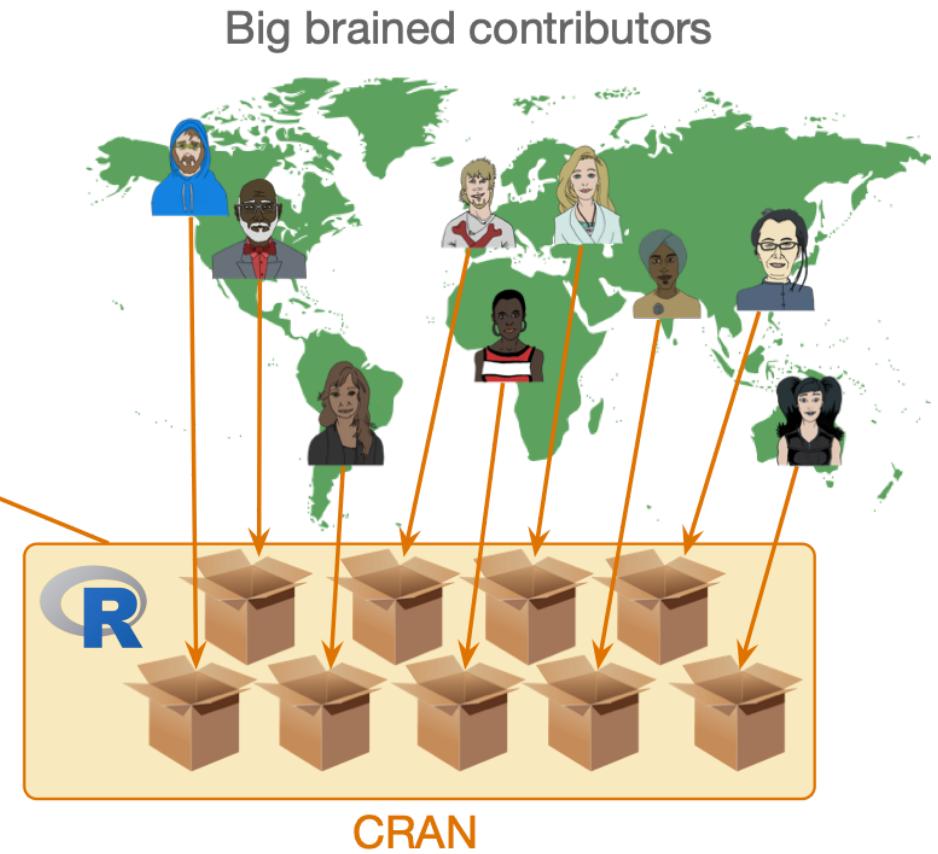
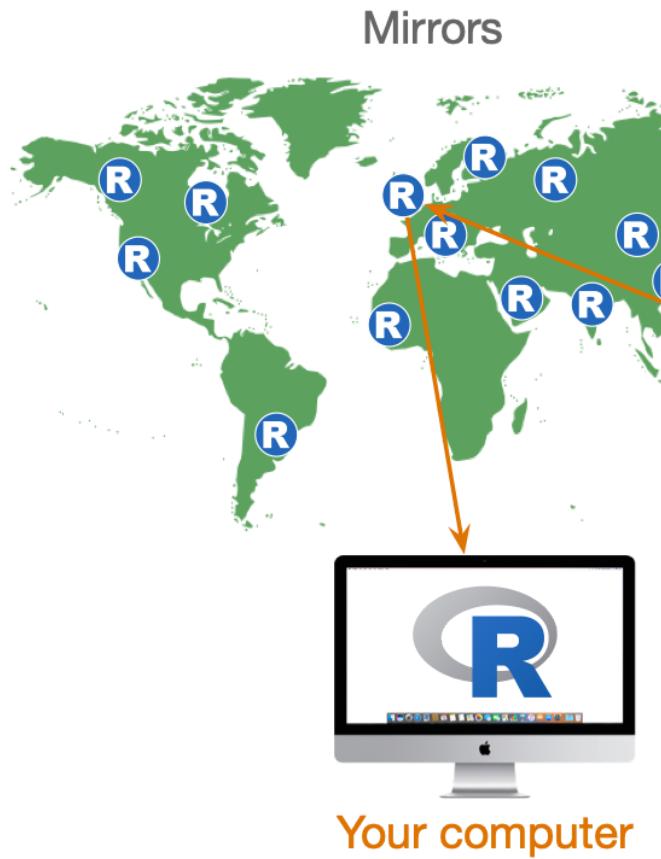


Try It!

```
metallica <- c("Lars", "James", "Kirk", "Rob")
metallica
## [1] "Lars"  "James" "Kirk"  "Rob"
```



Installing and loading packages



Installing and loading packages

Install the package from CRAN

- You need to install the package into R's repository of packages on your computer.
- Every time you update or re-install R you need to re-install packages to use them.
- `install.packages("package_name")`
- **Install packages using the command line. Do NOT include `install.packages()` in markdown files or the package will be installed every time you knit the document**

Load the package

- To use a particular package in a current session load it from the repository.
- `library(package_name)`

Try it!

- Execute `install.packages("tidyverse")`



The tidyverse

A set of packages built upon a common philosophy of data science



ANDY FIELD





The pipe operator (%>%)

Multiple commands:

```
core_members <- subset(metallica, metallica != "Rob")
core_members <- sort(core_members)
```

Nested commands:

```
core_members <- sort(subset(metallica, metallica != "Rob"))
```

Piped commands:

```
core_members <- metallica %>%
  subset(., metallica != "Rob") %>%
  sort()
```



ANDY FIELD



Using a ‘setup’ code chunk

I advise you have a setup chunk that

- Sets global options for your code chunks
- Loads all of the packages you plan to use (in alphabetic order)
- Loads any data that you plan to use

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

library(here)
library(kableExtra)
library(tidyverse)

tap_tib <- readr::read_csv("/Volumes/the_repository/documents/Academic/teaching/an_adventure_in_statistics/my_adve
```



Filepaths made easy (well, easier ...)

Relative paths

If you use an RStudio project, you can use relative paths

- `tap_tib <- readr::read_csv("../data/tap_parenting.csv")`

The `here` package

If you use an RStudio project, `here::here()` returns the project folder

- Text within `here()` returns the folder or file within the project folder that matches the text
- `tap_tib <- here::here("data/tap_parenting.csv") %>% readr::read_csv()`



```
library(here)
here::here()

## [1] "/Volumes/Alpha Lacertae/Documents/Academic/teaching/an_adventure_in_statistics/ais_presentations/ais_intro_r"

here::here("data")

## [1] "/Volumes/Alpha Lacertae/Documents/Academic/teaching/an_adventure_in_statistics/ais_presentations/ais_intro_r/data"

here::here("data/tap_parenting.csv")

## [1] "/Volumes/Alpha Lacertae/Documents/Academic/teaching/an_adventure_in_statistics/ais_presentations/ais_intro_r/data/tap_
```



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

library(kableExtra)
library(tidyverse)

tap_tib <- readr::read_csv("../data/tap_parenting.csv")
```

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

library(here)
library(kableExtra)
library(tidyverse)

tap_tib <- here::here("data/tap_parenting.csv") %>%
  readr::read_csv()
```



Messy vs. Tidy data

Messy data (aka 'wide' data)

What IBM SPSS Statistics uses

- Each row represents a unique case/entity

Tidy data (aka 'long' data)

What many (but not all) R functions require

- Each row represents an instance of the outcome measure
- Rows code information about that 'instance'



Tidy data example

Are invisible people mischievous?

- Placed participants in an enclosed community riddled with hidden cameras
- Measured how many mischievous acts participants performed in a week
- Manipulated whether or not there was access to an invisibility cloak



Independent design

- 12 participants given an invisibility cloak
- 12 participants not given an invisibility cloak
- A random sample of $N = 7$ from each group is shown

id	cloak	mischief
Anupama	No cloak	1
Collin	No cloak	5
Darrell	No cloak	4
Kinaana	No cloak	5
Steven	No cloak	2
Tyler	No cloak	6
Vanessa	No cloak	6
Conan	Cloak	6
Devante	Cloak	7
Jerry	Cloak	5
Kathryn	Cloak	2
Sage	Cloak	3
Shajee'a	Cloak	4
Tamara	Cloak	8



Repeated measures design

- 12 participants given an invisibility cloak for one week
- During a different week they did not have an invisibility cloak
- A random subsample of $N = 7$ participants is displayed

id	cloak	mischief
Anupama	No cloak	1
Anupama	Cloak	3
Collin	No cloak	5
Collin	Cloak	6
Darrell	No cloak	4
Darrell	Cloak	5
Kinaana	No cloak	5
Kinaana	Cloak	5
Steven	No cloak	2
Steven	Cloak	4
Tyler	No cloak	6
Tyler	Cloak	5
Vanessa	No cloak	6
Vanessa	Cloak	8



```
tap_tib <- here::here("data/tap_parenting.csv") %>%  
  readr::read_csv()
```

```
tap_tib
```

tap_parenting.csv

id	intervention	efficacy	did_bhv	id_school	time_cat	intention	time_num	positivity
aa54j	Group	7	1	school_7	baseline	28	0	1
aa54j	Group	7	1	school_7	1 month	46	1	5
aa54j	Group	7	1	school_7	6 months	29	6	2
ac09v	Group	4	1	school_6	baseline	54	0	5
ac09v	Group	4	1	school_6	1 month	48	1	5
ac09v	Group	4	1	school_6	6 months	59	6	5
ad17o	Leaflet	6	1	school_7	baseline	26	0	2
ad17o	Leaflet	6	1	school_7	1 month	28	1	5
ad17o	Leaflet	6	1	school_7	6 months	26	6	2
ad43a	Group	3	1	school_6	baseline	57	0	4



Part 5: Getting the most from practical classes



Practical classes

The practical classes are based on a package of interactive tutorials called [adventr](#) that I wrote

- Students work at your own pace
- You can work with friends/peers to support each other.
- Tutors will wander around giving you one-to-one help when you need it



Installing adventr

```
install.packages("remotes")
library(remotes)
remotes::install_github("profandyfield/adventr")
```

Running a tutorial

```
library(adventr)
learnr::run_tutorial("name_of_tutorial", package = "adventr")
learnr::run_tutorial("adventr_03", package = "adventr")
```



Suggested workflow

Create an RStudio project called **my_advent**

- Within it create folders called **data** and **r_docs**
- Save all of the data files for the tutorials (on Canvas) into the **data** folder
- Save the file **tutorial_template.Rmd** into your **r_docs** folder



Suggested workflow (cont.)

For a given tutorial, have **two** RStudio sessions running simultaneously

- To open a second session select Session > New Session

Session 1: run the tutorial

```
library(adventr)
learnr::run_tutorial("adventr_03", package = "adventr")
```

Session 2: make notes

- Open the file **tutorial_template.Rmd** and save it with a name related to the tutorial
- As you work through the tutorial, copy the code you've written in the tutorial into code chunks in the Rmarkdown file
- Make notes (for example, anything you didn't understand at first, or things to help you remember what you did and why you did it). This will help with your reflective statements
- Save the markdown file for future reference, and/or knit it into an html document

