An Adventure in Statistics

Andy Field

University of Sussex

2019-09-19

Key information

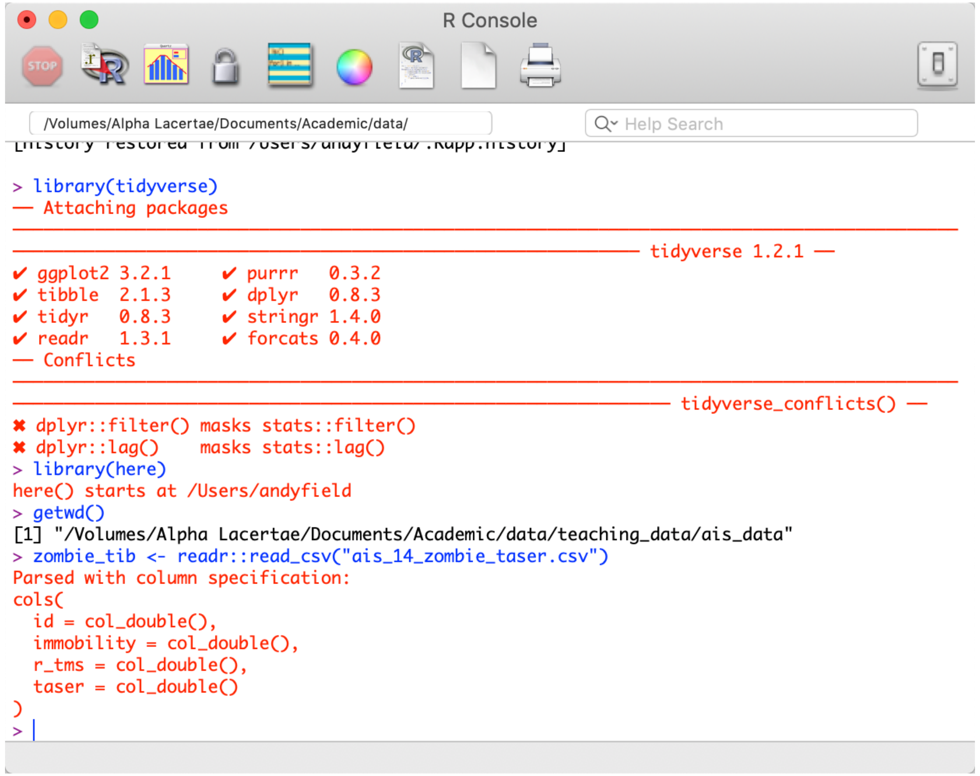
Teaching

- 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/
Help Search
> 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()
)
>


RStudio®

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

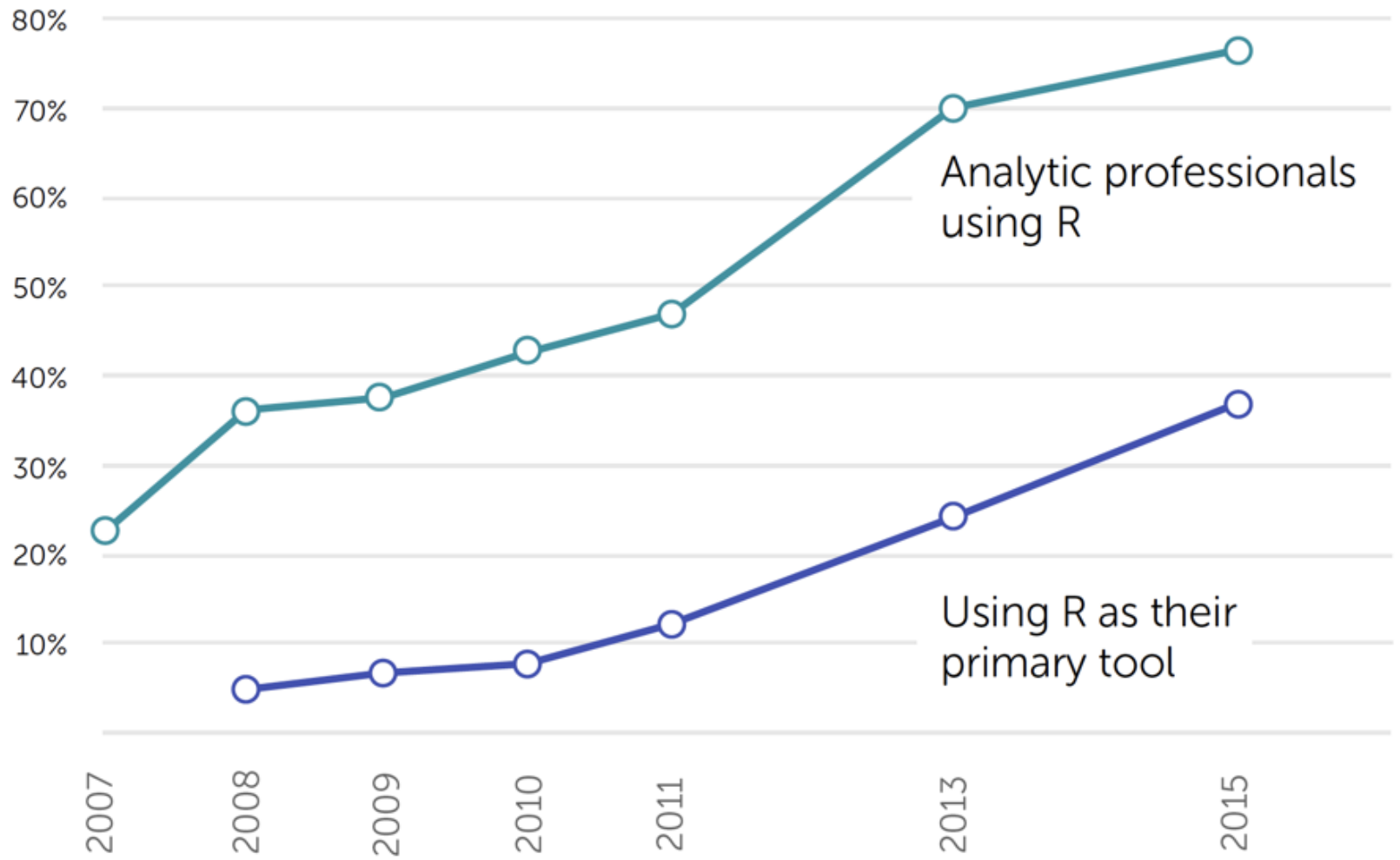
R is the most widely used data analysis software

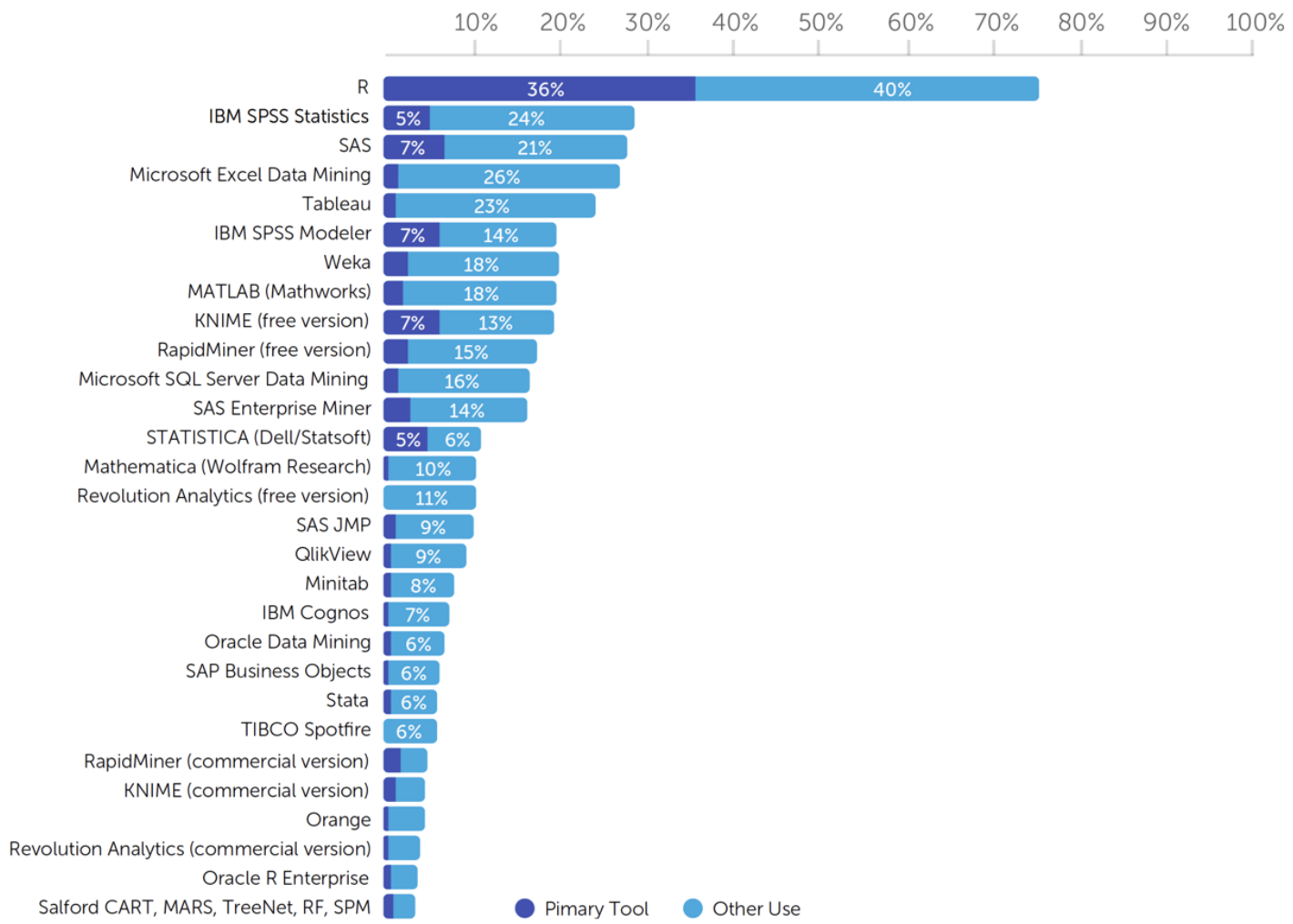
Reproducible science

Cutting edge

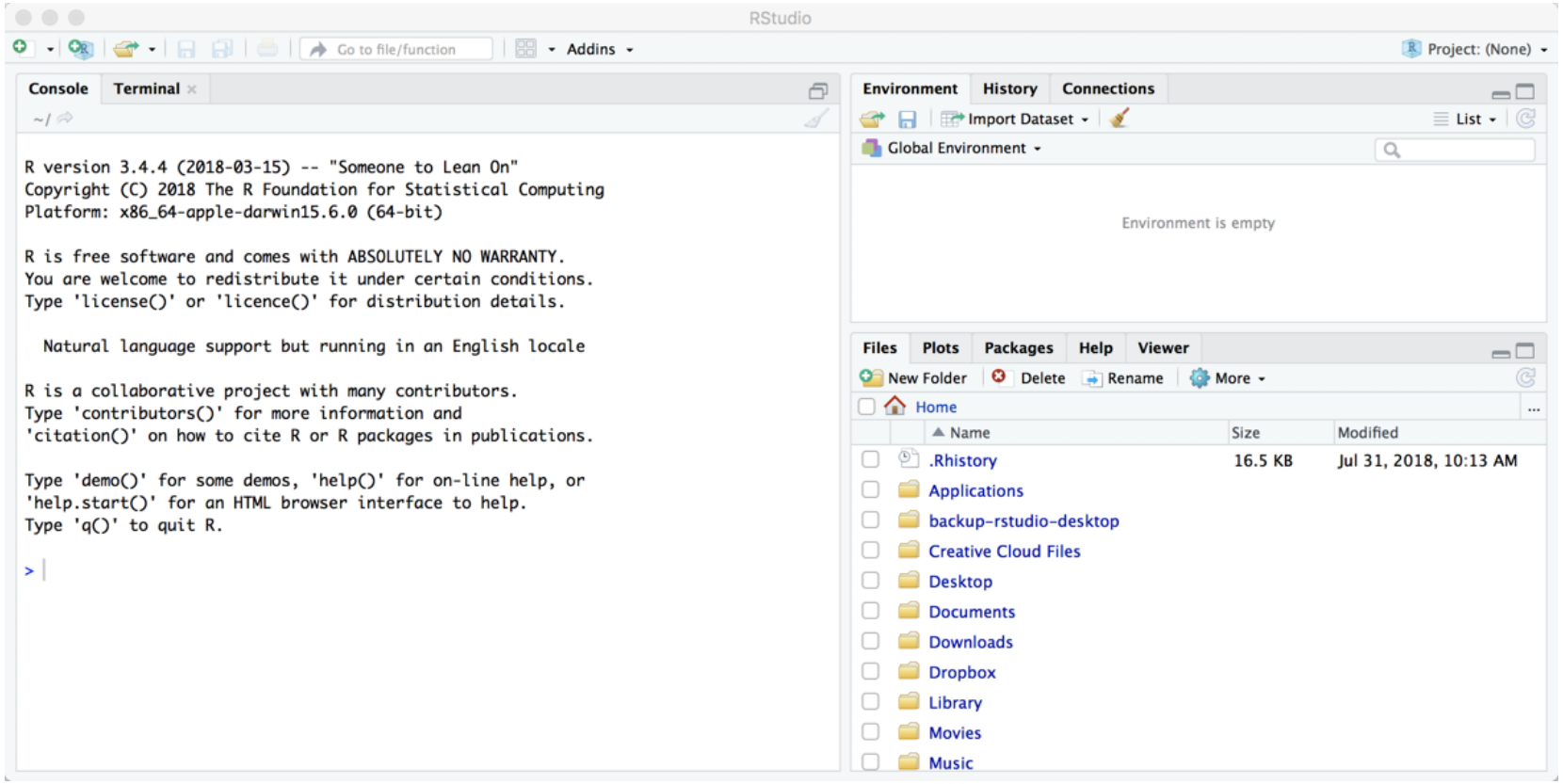
One stop shop

RStudio is amazeballs


80%
70%
60%
50%
40%
30%
20%
10%
Analytic professionals using R
Using R as their primary tool
2007
2008
2009
2010
2011
2013
2015

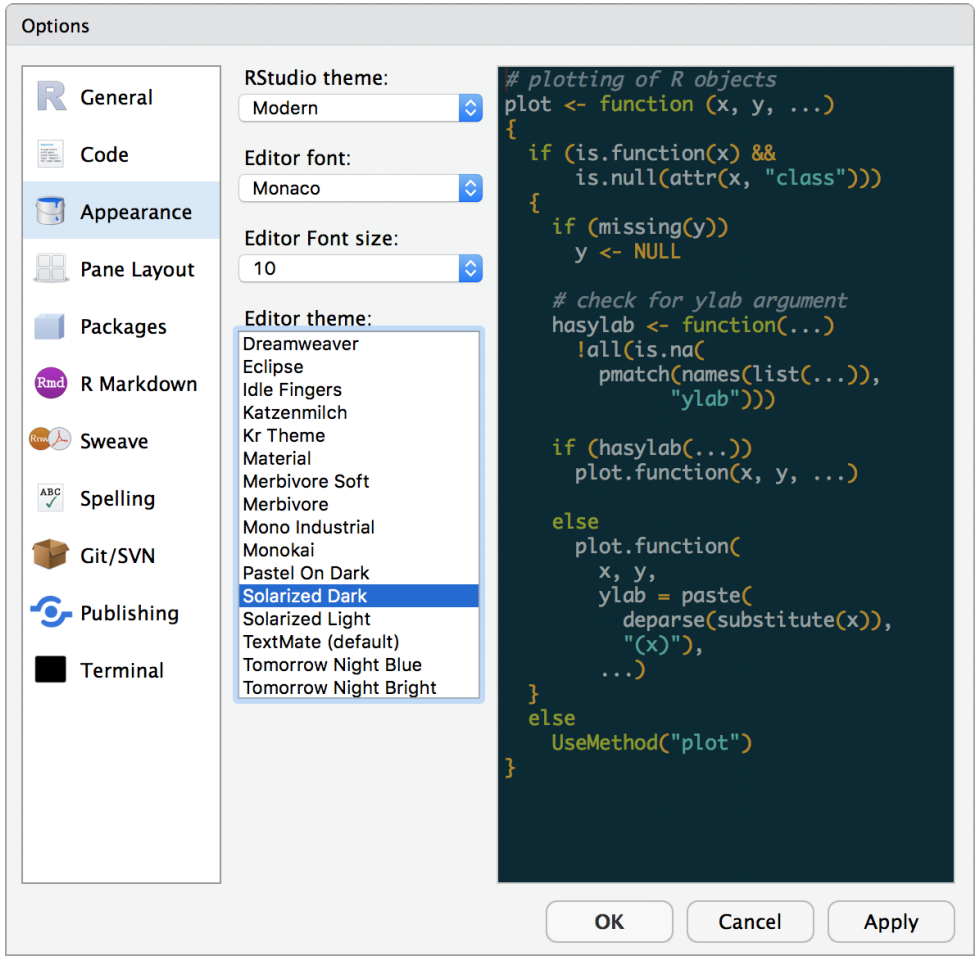

10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
R
36%
40%
IBM SPSS Statistics
5%
24%
SAS
7%
21%
Microsoft Excel Data Mining
26%
Tableau
23%
IBM SPSS Modeler
7%
14%
Weka
18%
MATLAB (Mathworks)
18%
KNIME (free version)
7%
13%
RapidMiner (free version)
15%
Microsoft SQL Server Data Mining
16%
SAS Enterprise Miner
14%
STATISTICA (Dell/Statsoft)
5%
6%
Mathematica (Wolfram Research)
10%
Revolution Analytics (free version)
11%
SAS JMP
9%
QlikView
9%
Minitab
8%
IBM Cognos
7%
Oracle Data Mining
6%
SAP Business Objects
6%
Stata
6%
TIBCO Spotfire
6%
RapidMiner (commercial version)
KNIME (commercial version)
Orange
Revolution Analytics (commercial version)
Oracle R Enterprise
Salford CART, MARS, TreeNet, RF, SPM
Pimary Tool
Other Use

Starting up R Studio

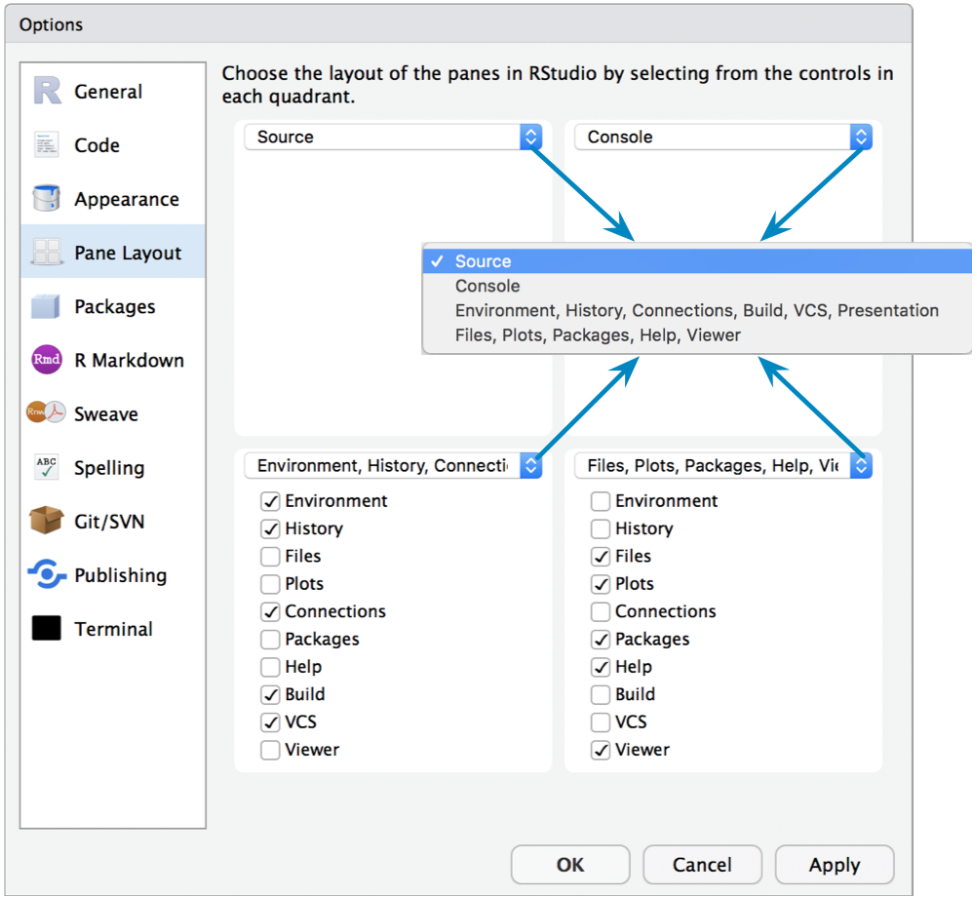


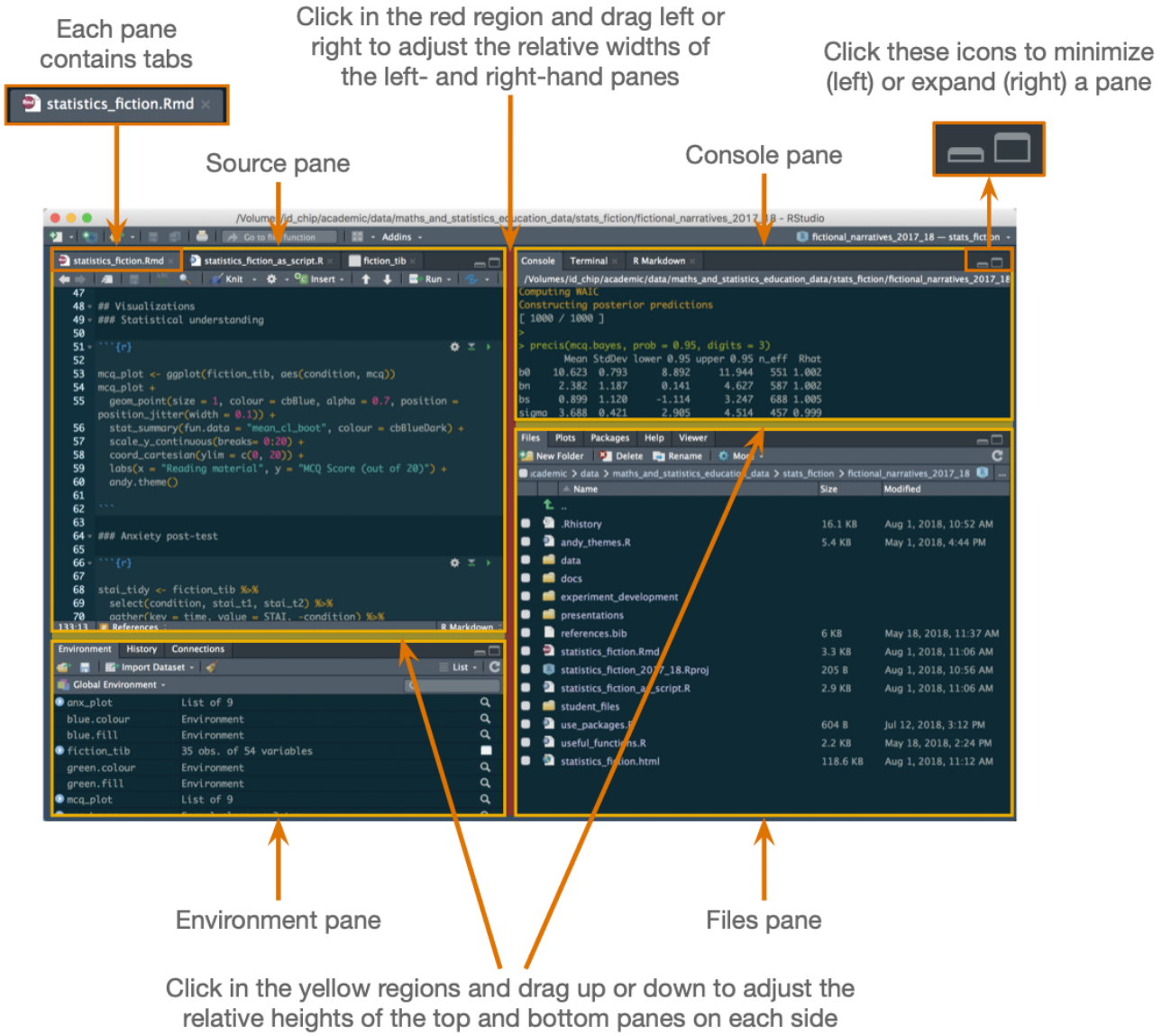
Try it! Colour scheme

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



Try it! Pane locations




Each pane contains tabs
statistics_fiction.Rmd
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
Source pane
Console pane
Environment pane
Files 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 .Renviron
- Save it to the Documents folder on the N: drive

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

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

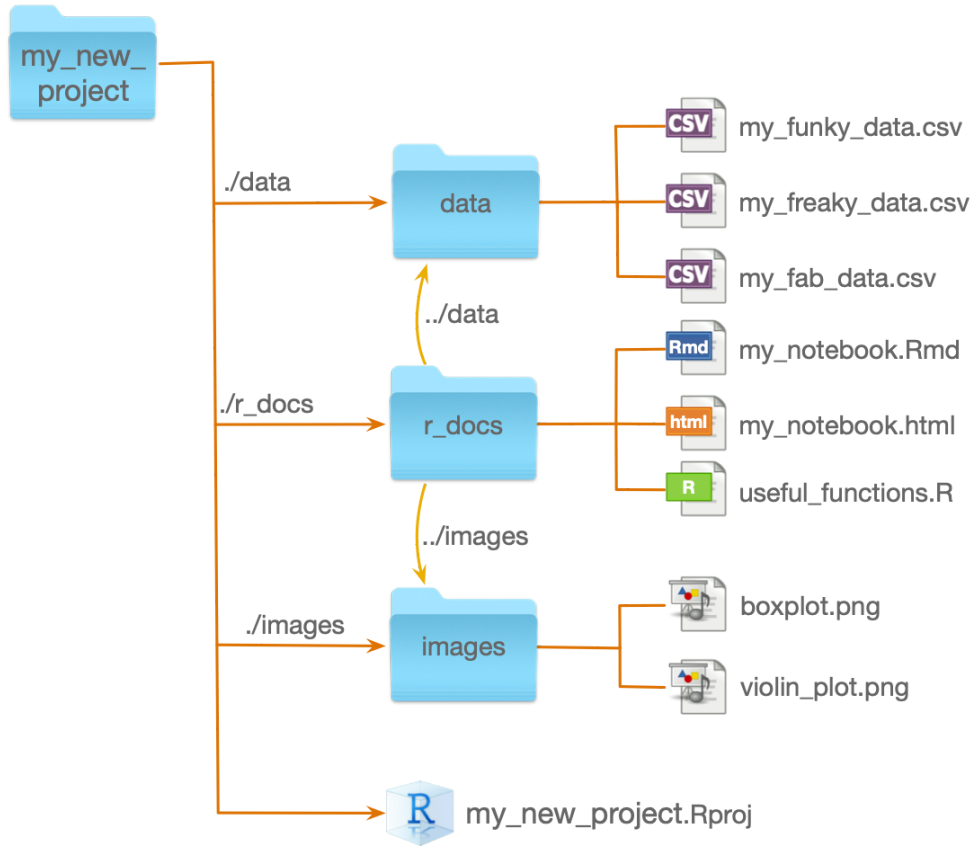
Use project files!

Get organized!

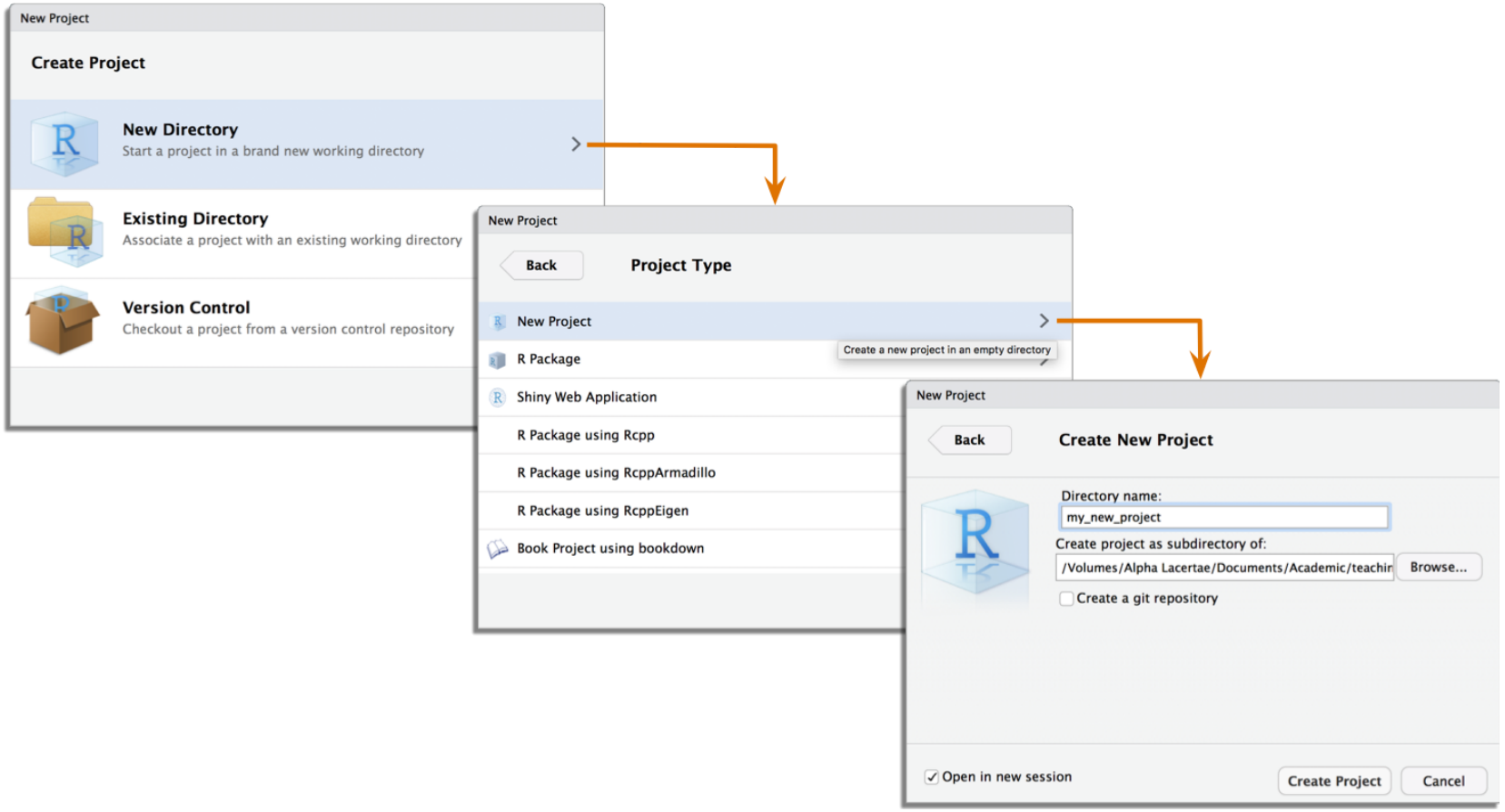
```
C:/Users/andyfield/Documents/my_new_project/data/my_funky_data.csv
```

```
./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!)

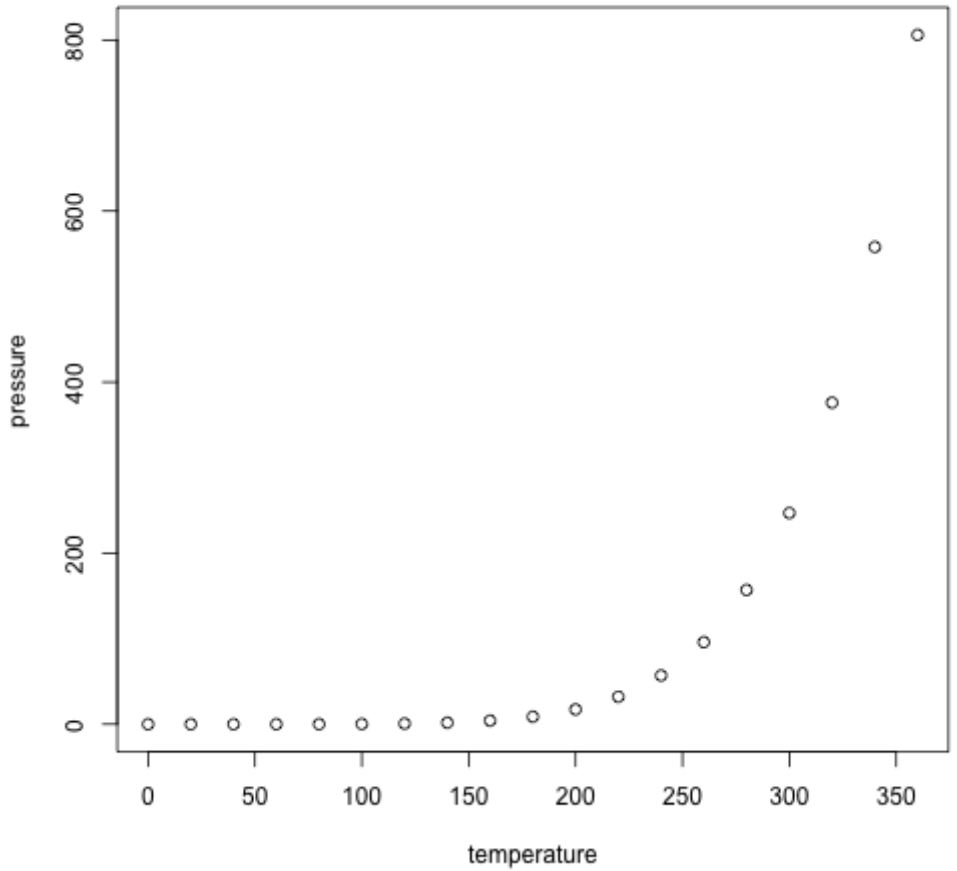
- Great for reproducibility
- Great for integrating analysis into documents
- Great for impressing your friends and loved ones

Try it

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

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


```
plot(pressure)
```

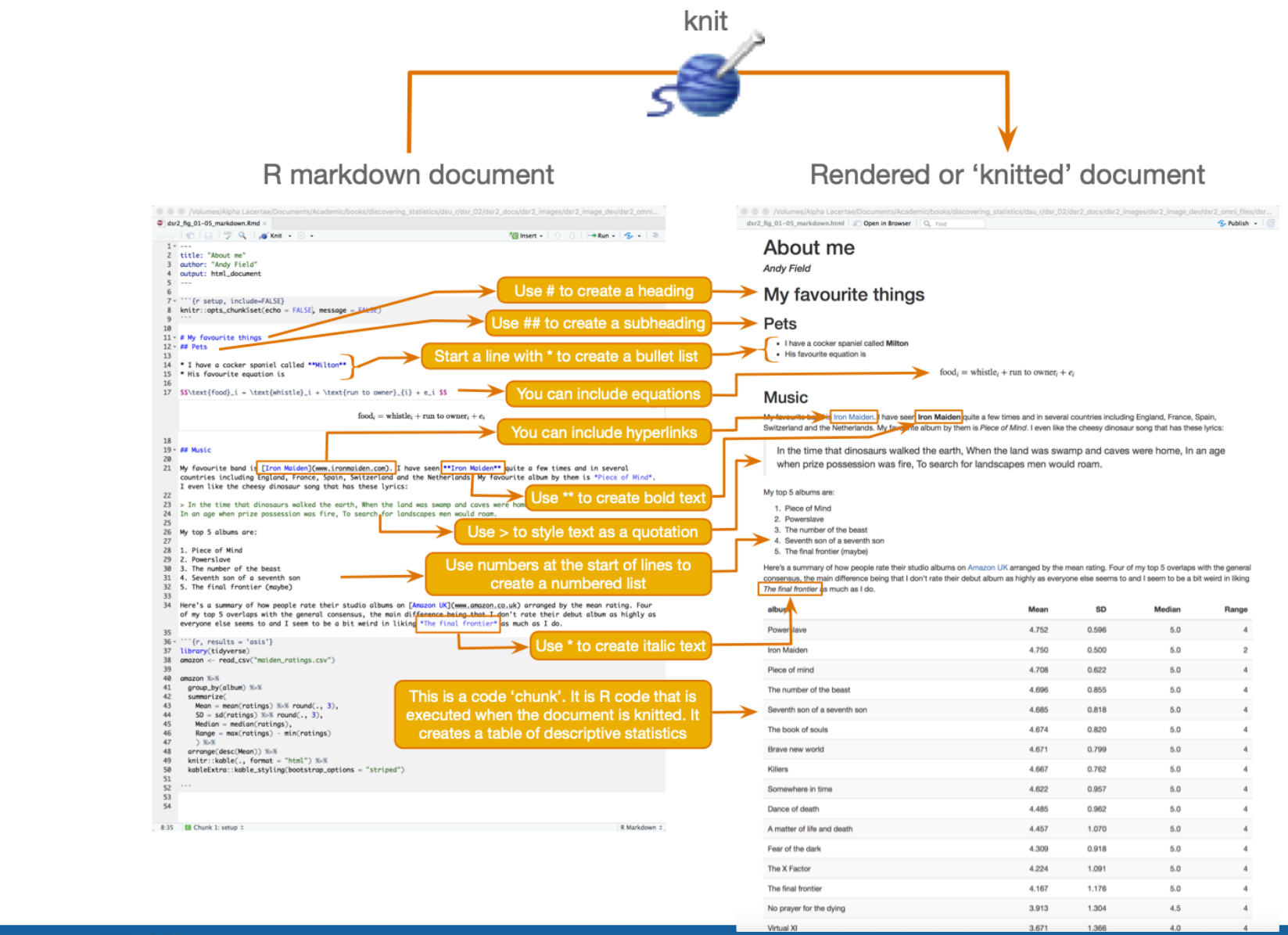


```
cor(pressure)
```

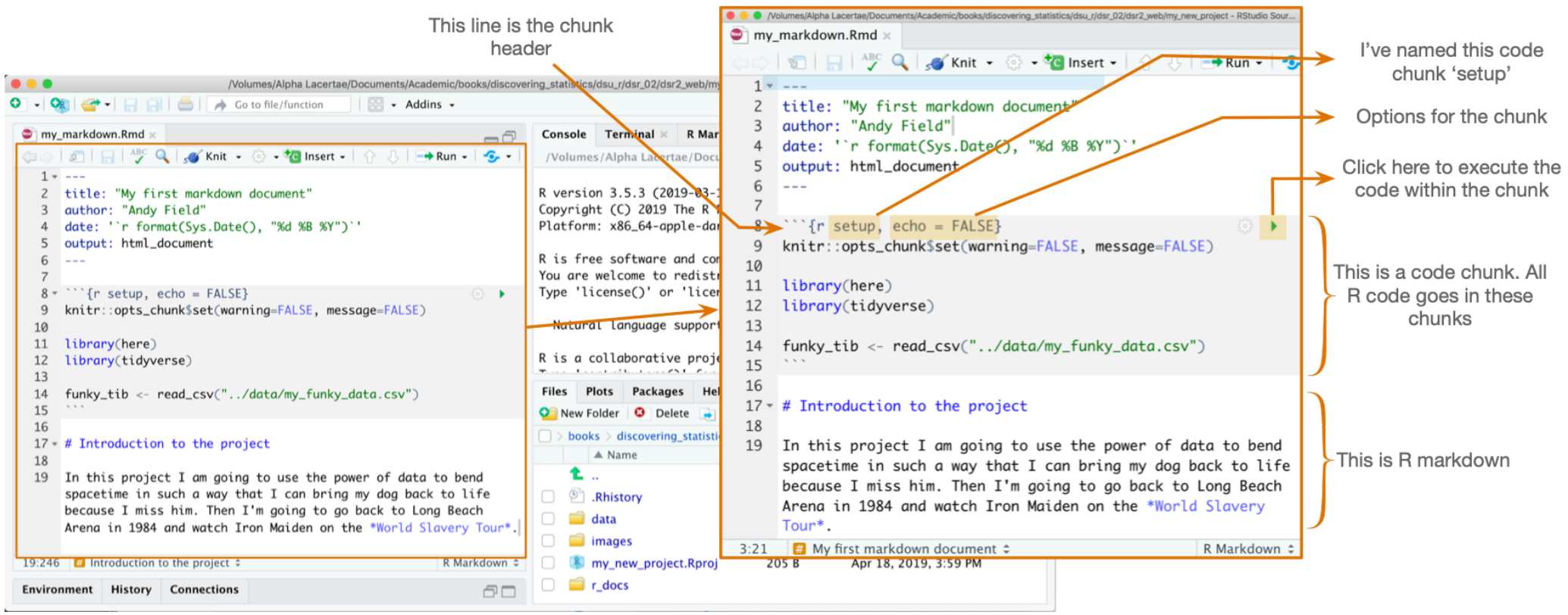
```
##             temperature  pressure
## temperature   1.0000000 0.7577923
## pressure      0.7577923 1.0000000
```



rmarkdown
www.rstudio.com


knit
R markdown document
Rendered or 'knitted' document
Use # to create a heading
Use ## to create a subheading
Start a line with * to create a bullet list
You can include equations
You can include hyperlinks
Use ** to create bold text
Use > to style text as a quotation
Use numbers at the start of lines to create a numbered list
Use * to create italic text
This is a code 'chunk'. It is R code that is executed when the document is knitted. It creates a table of descriptive statistics
About me
Andy Field
My favourite things
Pets
I have a cocker spaniel called Milton
His favourite equation is
Music
My favourite band is Iron Maiden. I have seen Iron Maiden quite a few times and in several countries including England, France, Spain, Switzerland and the Netherlands. My favourite album by them is Piece of Mind. I even like the cheesy dinosaur song that has these lyrics:
In the time that dinosaurs walked the earth, When the land was swamp and caves were home, In an age when prize possession was fire, To search for landscapes men would roam.
My top 5 albums are:
1. Piece of Mind
2. Powerslave
3. The number of the beast
4. Seventh son of a seventh son
5. The final frontier (maybe)
Here's a summary of how people rate their studio albums on Amazon UK arranged by the mean rating. Four of my top 5 overlaps with the general consensus, the main difference being that I don't rate their debut album as highly as everyone else seems to and I seem to be a bit weird in liking The final frontier as much as I do.
album Mean SD Median Range
Powerslave 4.752 0.596 5.0 4
Iron Maiden 4.750 0.500 5.0 2
Piece of mind 4.708 0.622 5.0 4
The number of the beast 4.696 0.855 5.0 4
Seventh son of a seventh son 4.685 0.818 5.0 4
The book of souls 4.674 0.820 5.0 4
Brave new world 4.671 0.799 5.0 4
Killers 4.667 0.762 5.0 4
Somewhere in time 4.622 0.957 5.0 4
Dance of death 4.485 0.962 5.0 4
A matter of life and death 4.457 1.070 5.0 4
Fear of the dark 4.309 0.918 5.0 4
The X Factor 4.224 1.091 5.0 4
The final frontier 4.167 1.176 5.0 4
No prayer for the dying 3.913 1.304 4.5 4
Virtual XI 3.671 1.366 4.0 4

This is a markdown file



Inserting code chunks

Windows

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

Mac

- *cmd + opt + i* (⌘ + ⌥ + 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**
Text^superscript^
Text~subscript~
```

Knitted text

italic text

bold text

$\text{Text}^{\text{superscript}}$

$\text{Text}_{\text{subscript}}$

R Markdown: images

```
![Caption for image](filepath_for_image)

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

Figure 1: my lovely spaniel

R Markdown: hyperlinks

```
[text_describing_link](url)
```

Markdown

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

Knitted text

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

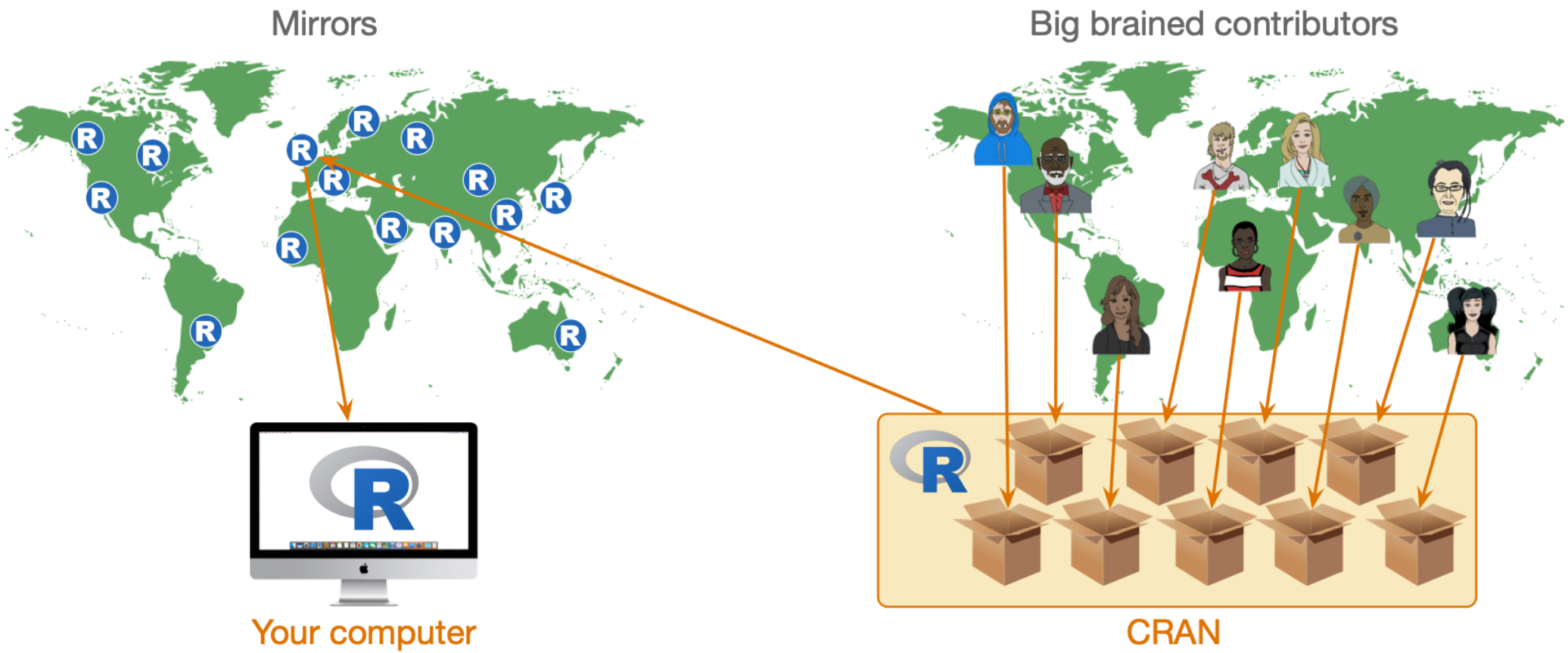
Assignment operator

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

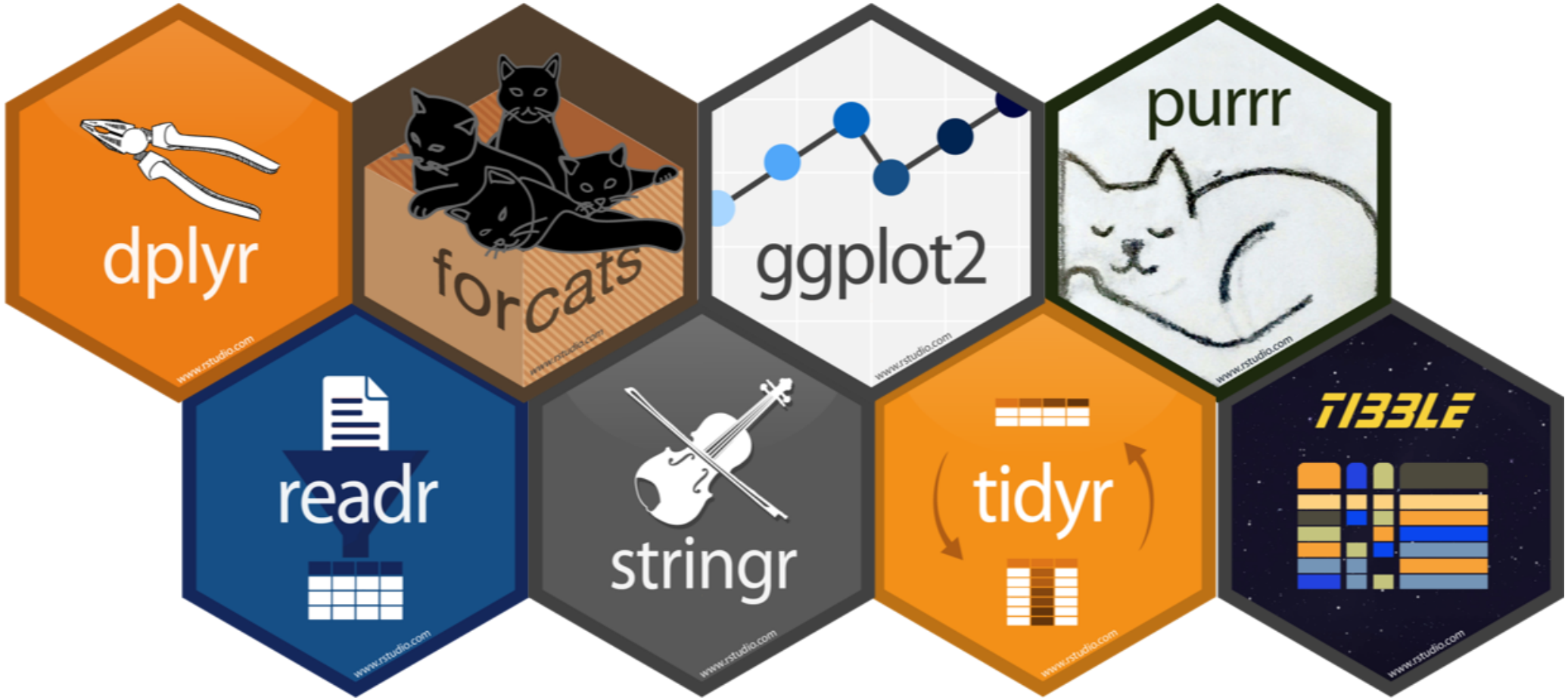
- To use a particular package in a current session load it from the repository.
- `library(package_name)`

Try it!

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

The tidyverse

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


dplyr
forcats
ggplot2
purrr
readr
stringr
tidyr
TIBBLE

The pipe operator (%>%)

Multiple commands:

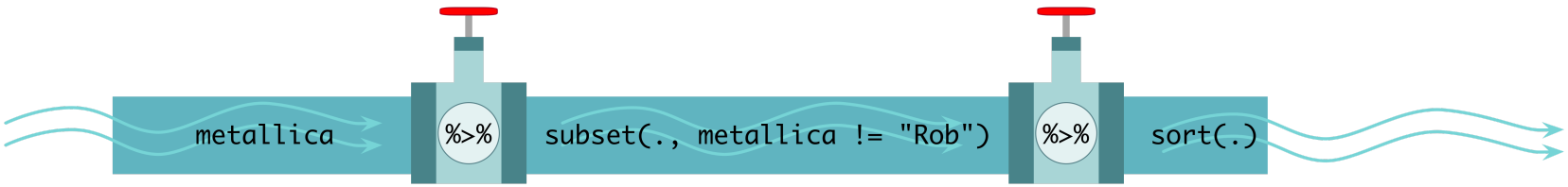
```
core_members <- subset(metallica, metallica != "Rob")
core_members <- sort(core_members)
```

Nested commands:

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

Piped commands:

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



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/the_repository/documents/Academic/teaching/an_adventure_in_statistics/ais_presentations/ais_intro_r"
```

```
here::here("data")
```

```
## [1] "/Volumes/the_repository/documents/Academic/teaching/an_adventure_in_statistics/ais_presentations/ais_intro_r/data"
```

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

```
## [1] "/Volumes/the_repository/documents/Academic/teaching/an_adventure_in_statistics/ais_presentations/ais_intro_r/data/ta
```


```
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'

```
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