

Reproducible Data Visualisations

Andrew Stewart
University of Manchester

Twitter: @ajstewart_lang
GitHub: github.com/ajstewartlang

xx/xx/xx

Reproducible Data Visualisations

Andrew Stewart
University of Manchester

Twitter: @ajstewart_lang
GitHub: github.com/ajstewartlang

xx/xx/xx

Science that can be replicated vs. science that can be reproduced

Replicable Science is when someone else can run a study the same as or conceptually equivalent to your one, and find a similar pattern of effects.

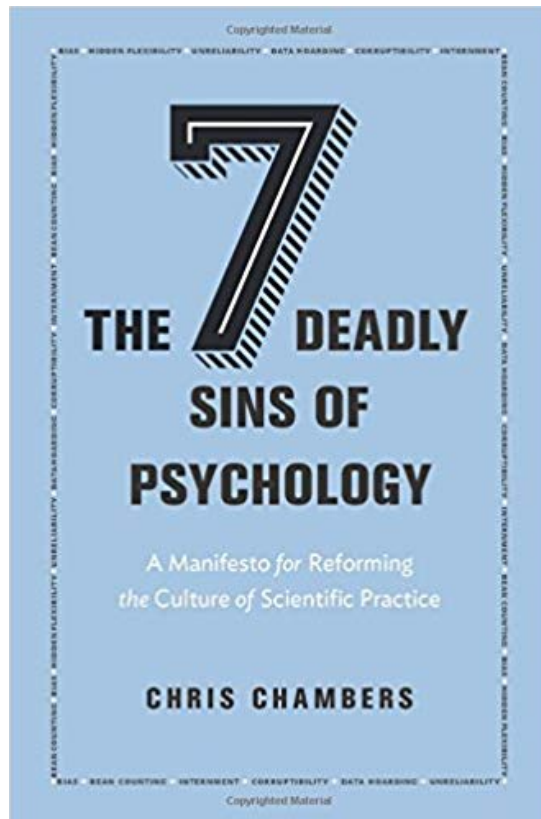
Reproducible Science is when someone else can take your data and your analysis code, run it and then find the same effects that you have reported.

How do we make our science more replicable?

How do we make our science more reproducible?

A move towards open science...

You really should read this book!



Sins include *p*-hacking, lack of power, HARKing, failing (refusal) to share data and code, too many researcher degrees of freedom...



<http://www.stat.columbia.edu/~gelman/>

Andrew Gelman gives the following recommendations to researchers:

- Analyze all your data.
- Present all your comparisons.
- Put in the effort to take accurate measurements (low bias, low variance, and a large enough sample size).
- Do repeated-measures comparisons where possible.
- Make your data public.

But it's not just the data you need to make public, but also your **code**!

What role can R play in Open and Reproducible Science?

- R scripts are easy to share allowing for reproducibility and easy public sharing of data and code.
- R is free, open source software that is much more flexible and powerful than SPSS.
- There is an active R community continuously updating statistical tests and packages that run in R.
- As R is a programming language, it forces you to **know** your data.

R vs. SPSS

“SPSS is like a bus - easy to use for the standard things, but very frustrating if you want to do something that is not already pre-programmed.

R is a 4-wheel drive off-roader, with a bike on the back, a kayak on top, good walking and running shoes in the passenger seat, and mountain climbing and spelunking gear in the back.

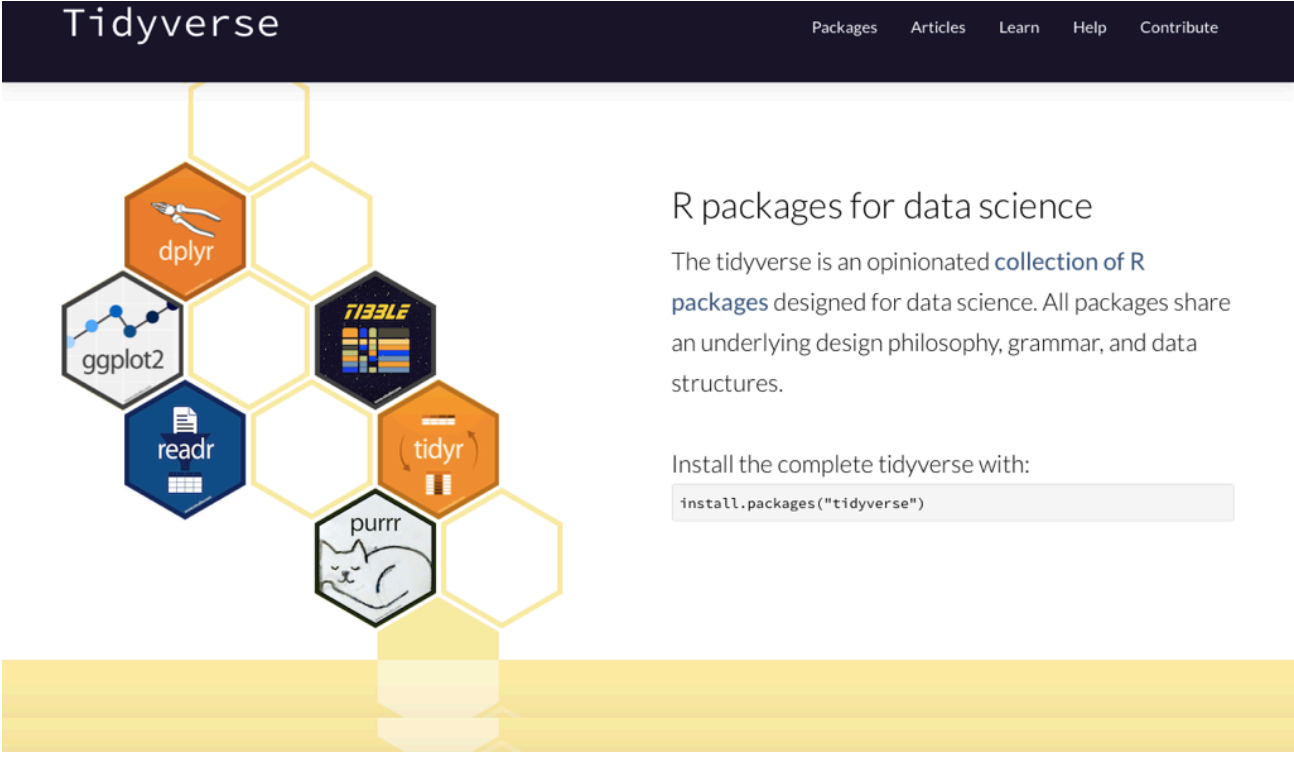
R can take you anywhere you want to go if you take time to learn how to use the equipment, but that is going to take longer than learning where the bus stops are in SPSS.” (*Greg Snow, 2010, stackoverflow.com*)

In meme form...

If statistics programs/languages were cars...



A workflow for reproducible science in the Tidyverse



The screenshot shows the top portion of the Tidyverse website. At the top is a dark blue navigation bar with the word "Tidyverse" on the left and links for "Packages", "Articles", "Learn", "Help", and "Contribute" on the right. Below the navigation bar is a large graphic on the left consisting of a cluster of hexagons. Some hexagons contain icons and labels for Tidyverse packages: "dplyr" (orange hexagon with a bird icon), "ggplot2" (grey hexagon with a network icon), "readr" (blue hexagon with a document icon), "tidyr" (orange hexagon with a document icon), "purrr" (grey hexagon with a cat icon), and "TIBBLE" (dark blue hexagon with a bar chart icon). To the right of this graphic, the text "R packages for data science" is followed by a paragraph: "The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures." Below this paragraph, it says "Install the complete tidyverse with:" followed by a code block containing the command `install.packages("tidyverse")`. The entire content area has a light yellow background with a subtle gradient.

Tidyverse

Packages Articles Learn Help Contribute

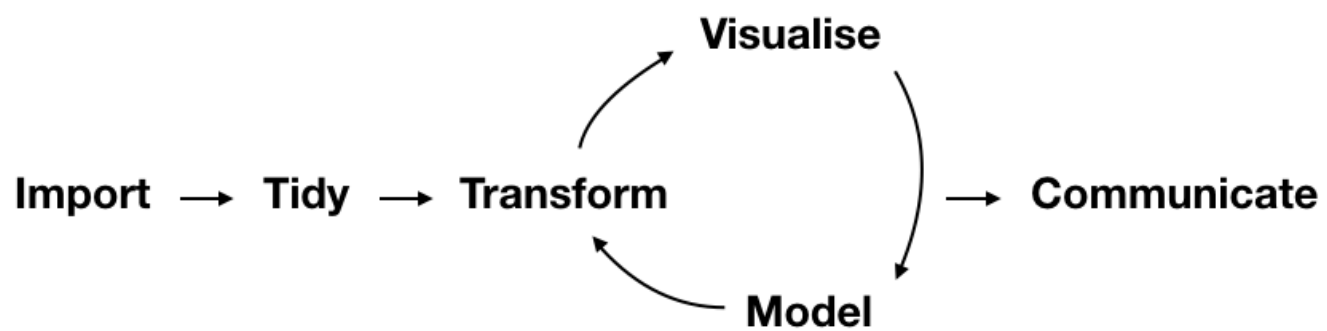
R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

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

A workflow for reproducible science in the Tidyverse

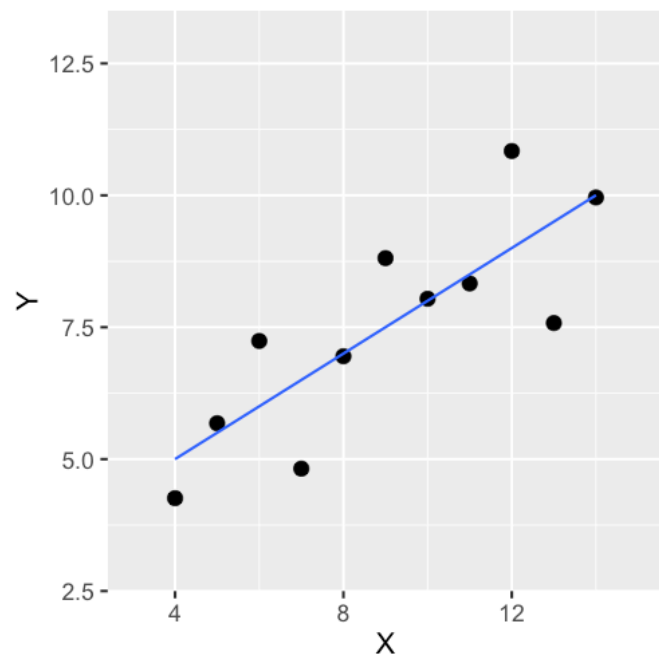


<https://www.tidyverse.org>

Why Data Visualisation is Important

Anscombe's Quartet

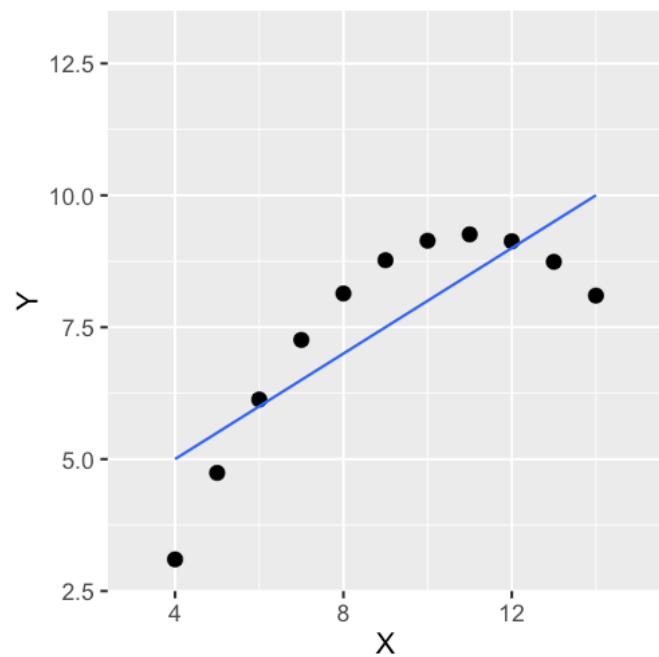
Plot 1



```
## [1] "Mean of X is: 9"
## [1] "SD of X is: 3.32"
## [1] "Mean of Y is: 7.5"
## [1] "SD of Y is: 2.03"
```

```
## [1] "Pearson's r is 0.82"
```

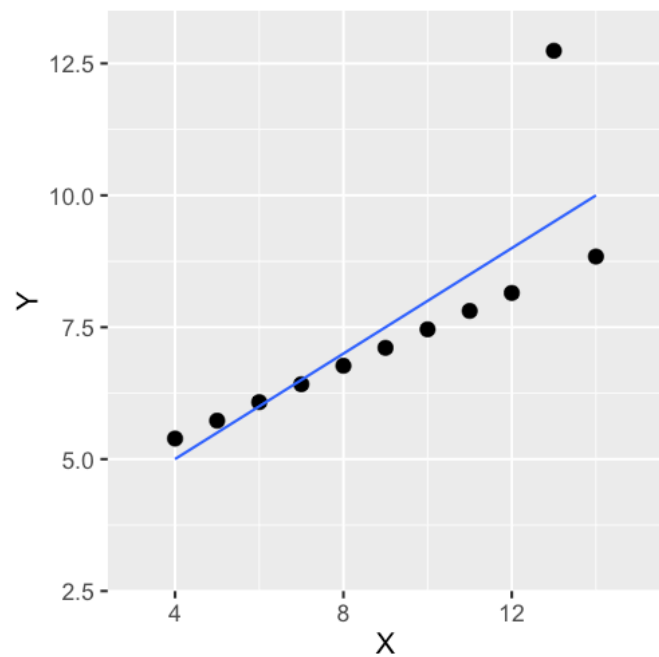
Plot 2



```
## [1] "Mean of X is: 9"
## [1] "SD of X is: 3.32"
## [1] "Mean of Y is: 7.5"
## [1] "SD of Y is: 2.03"
```

```
## [1] "Pearson's r is 0.82"
```

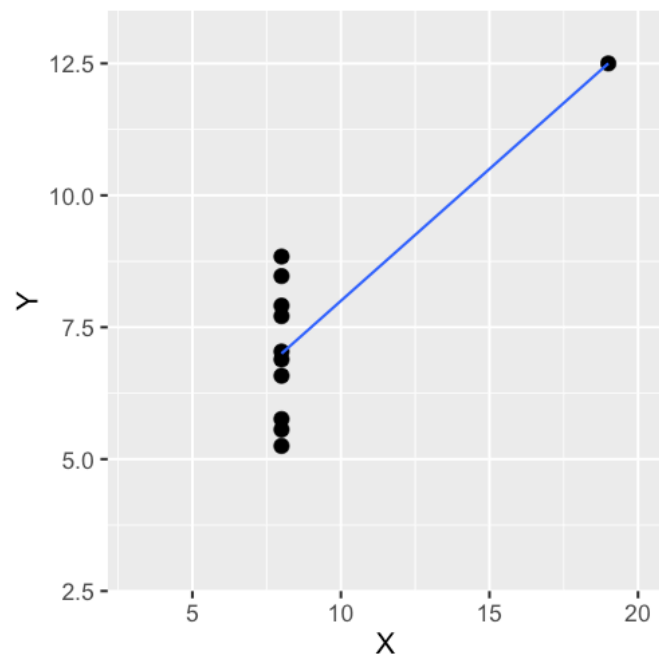
Plot 3



```
## [1] "Mean of X is: 9"
## [1] "SD of X is: 3.32"
## [1] "Mean of Y is: 7.5"
## [1] "SD of Y is: 2.03"
```

```
## [1] "Pearson's r is 0.82"
```

Plot 4

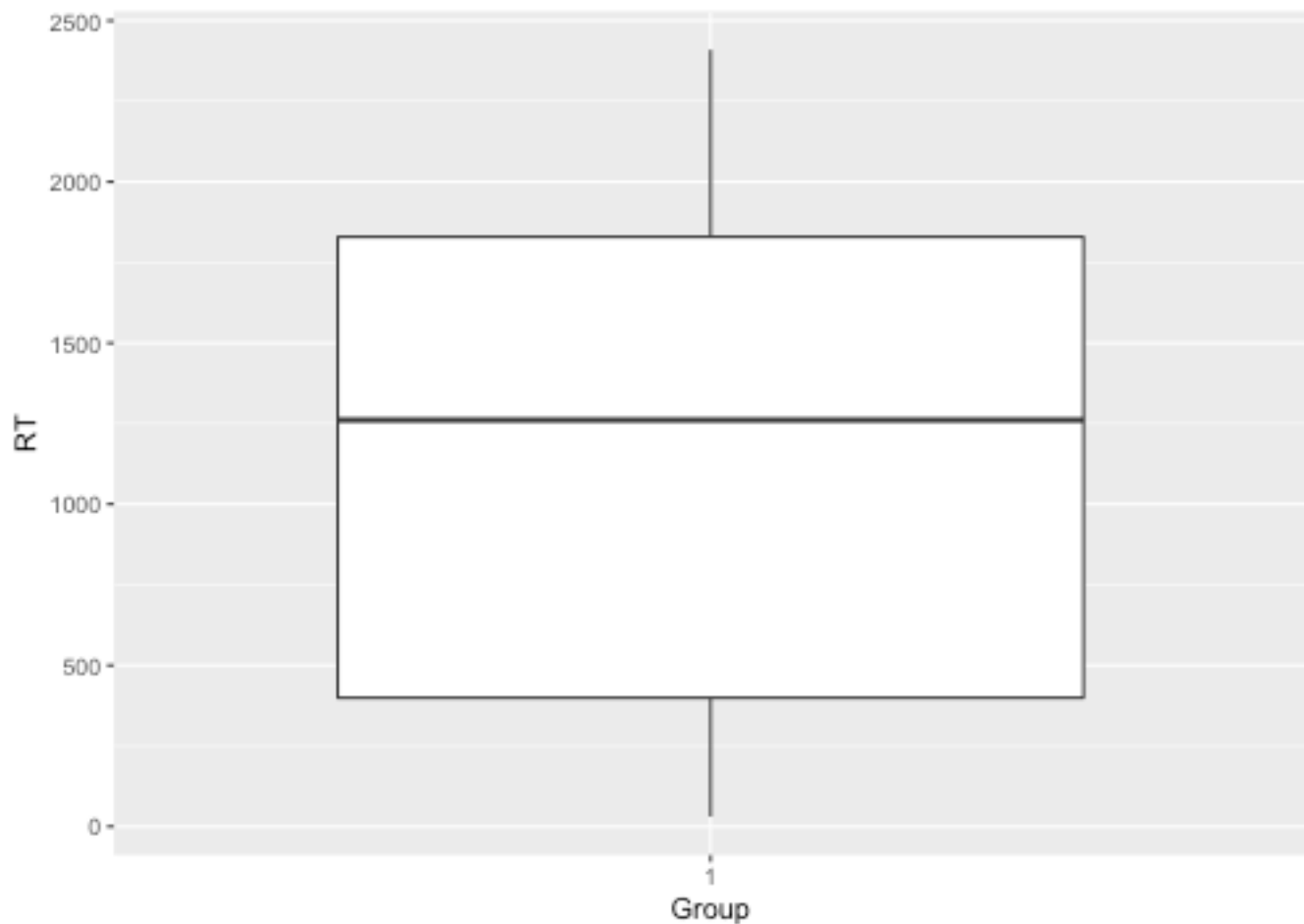


```
## [1] "Mean of X is: 9"
## [1] "SD of X is: 3.32"
## [1] "Mean of Y is: 7.5"
## [1] "SD of Y is: 2.03"
```

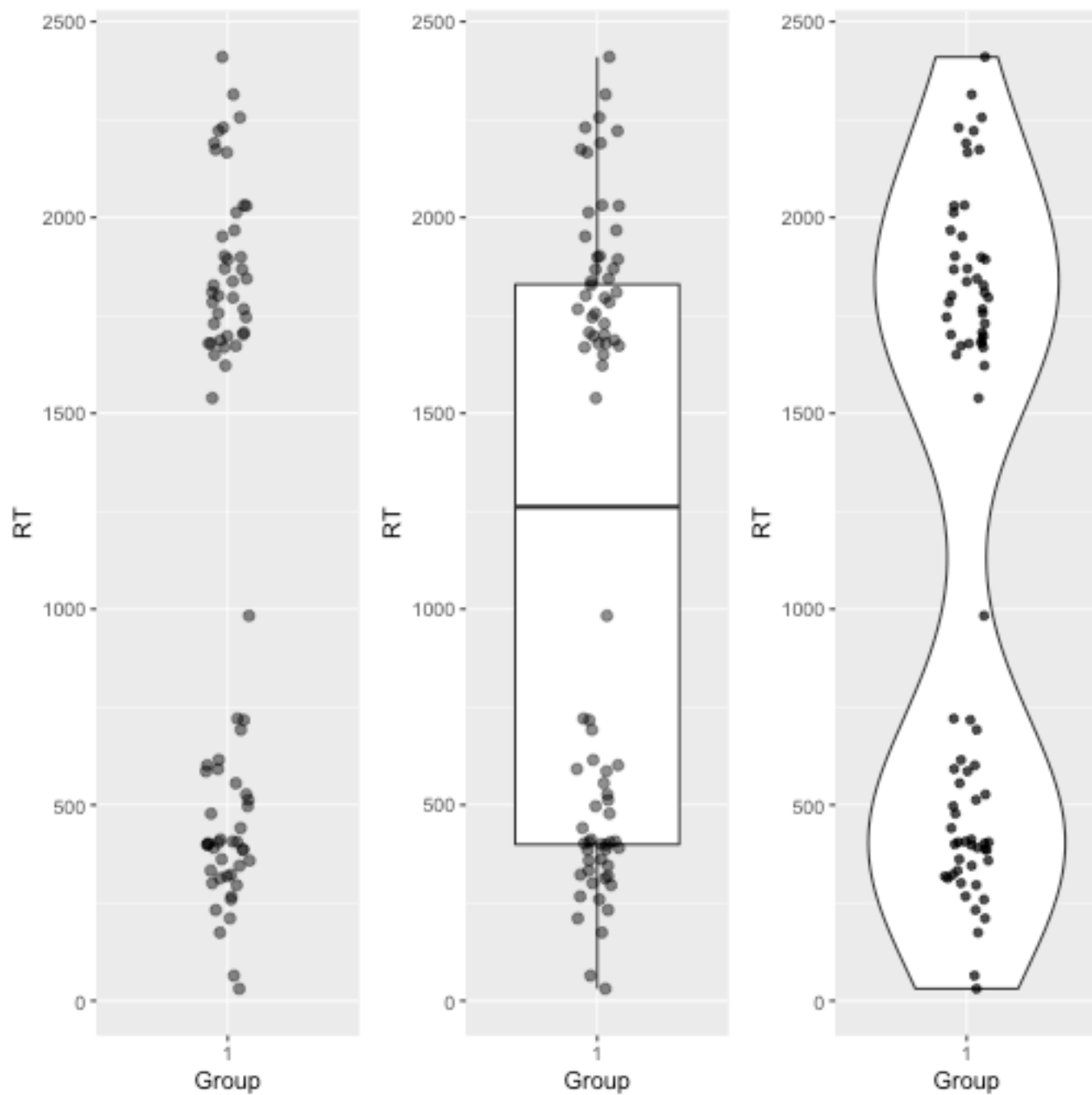
```
## [1] "Pearson's r is 0.82"
```


Plots Based on Aggregated Data Can Mislead...

```
ggplot(data1, aes(x = Group, y = RT)) + geom_boxp
```



But look more closely at the actual data...



The distribution of data matters

The data on the previous slide are clearly bimodal with no data point near the mean. Distribution shape matters and we need to capture that in our data visualisations.

If we only plotted and reported information related to aggregated data, we wouldn't be being honest about what our data look like.

Reasons for visualising data

For yourself - once you have collected your data, you should visualise it before you build any statistical models - does the data look (roughly) as expected with the right number of data points?

For others - when you present your work in a talk, on a poster, or in a published paper you want the viewer to be able to quickly and unambiguously extract the intended meaning from your visualisation.

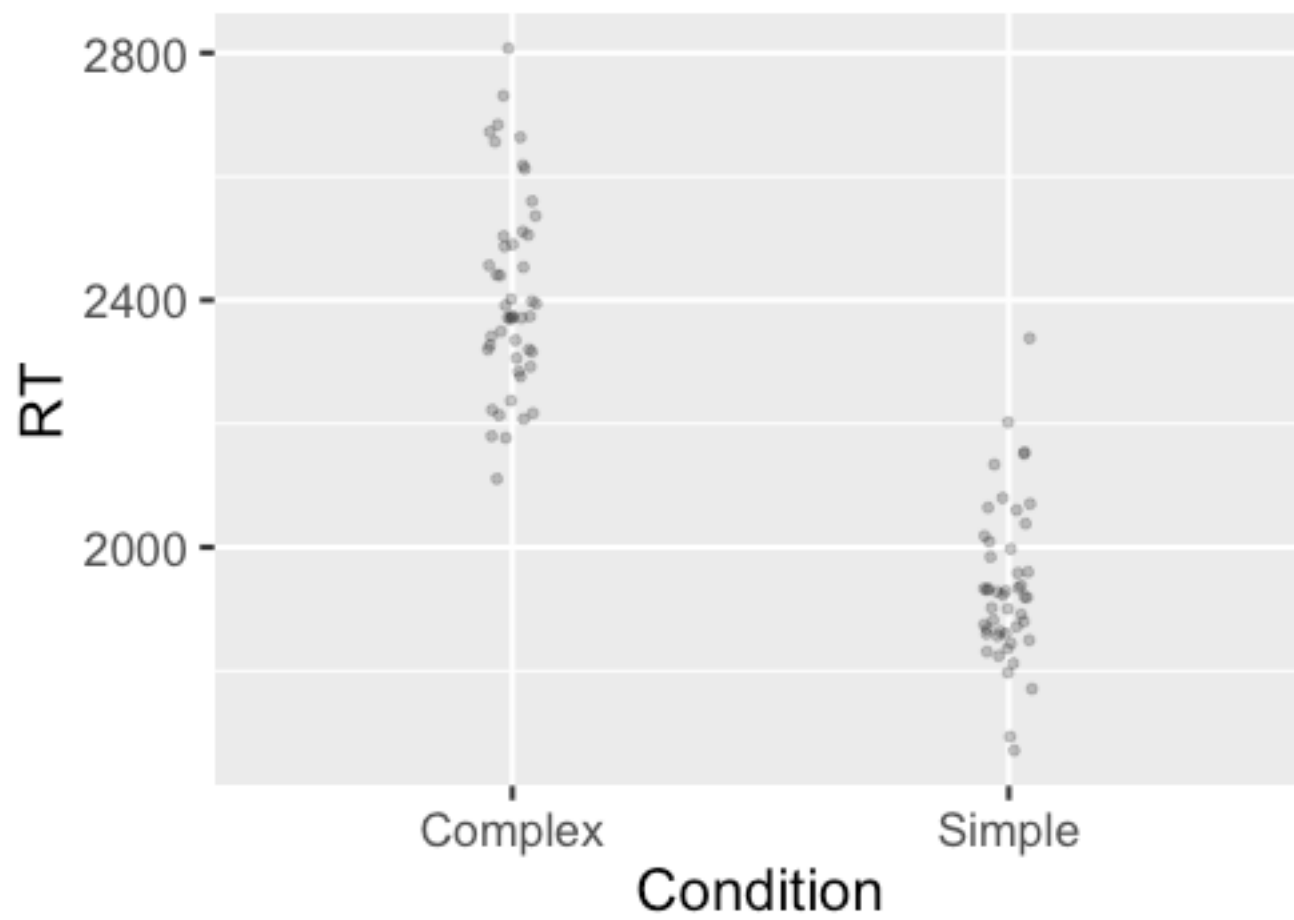
ggplot2

The ggplot2 package is part of the Tidyverse and is based around the Grammar of Graphics (Wickham, 2010):

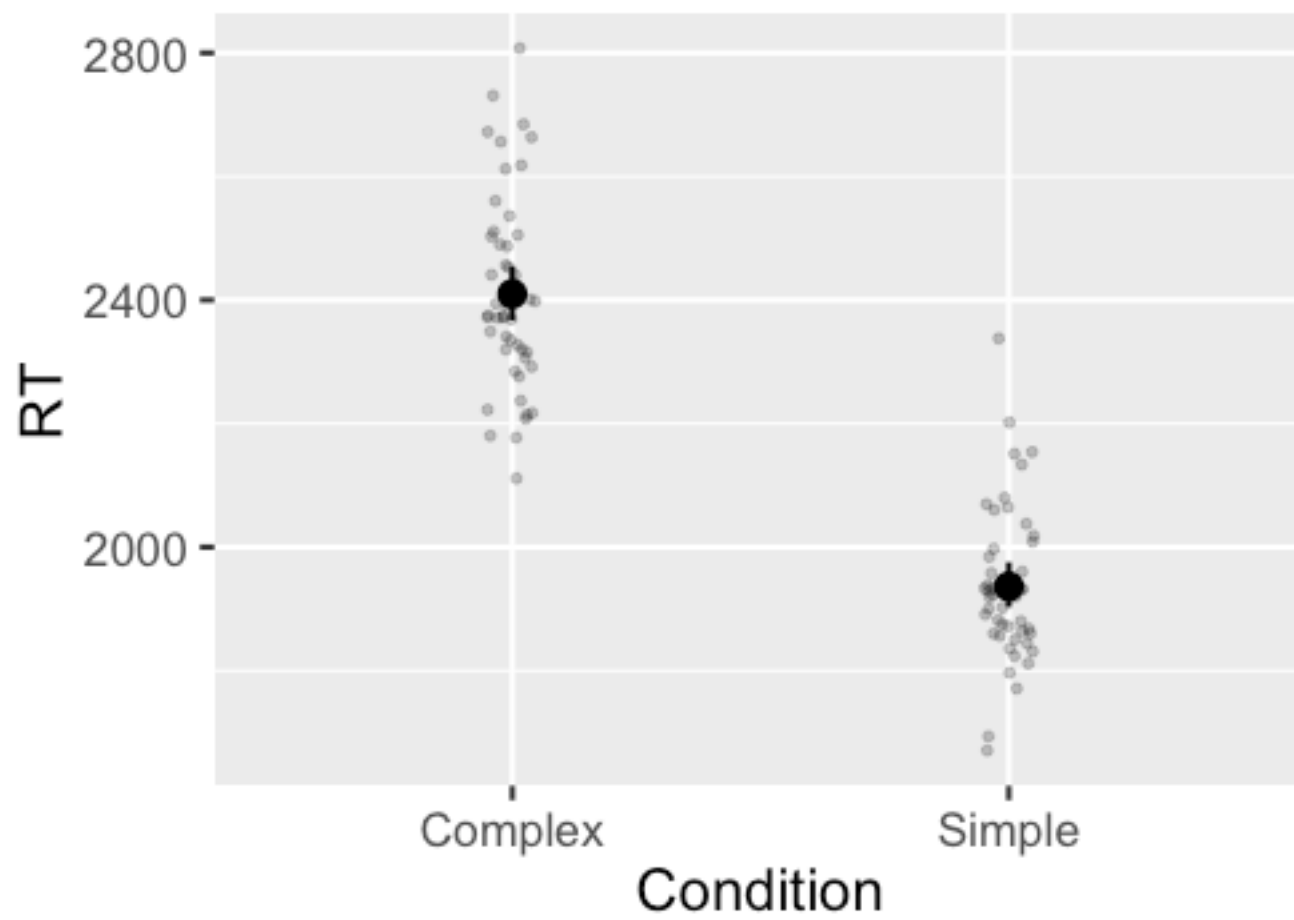
https://byrneslab.net/classes/biol607/readings/wickham_layered-grammar.pdf

Start with defining your data and aesthetics of the plot, before adding geometric objects (geoms), information about labelling, faceting etc.

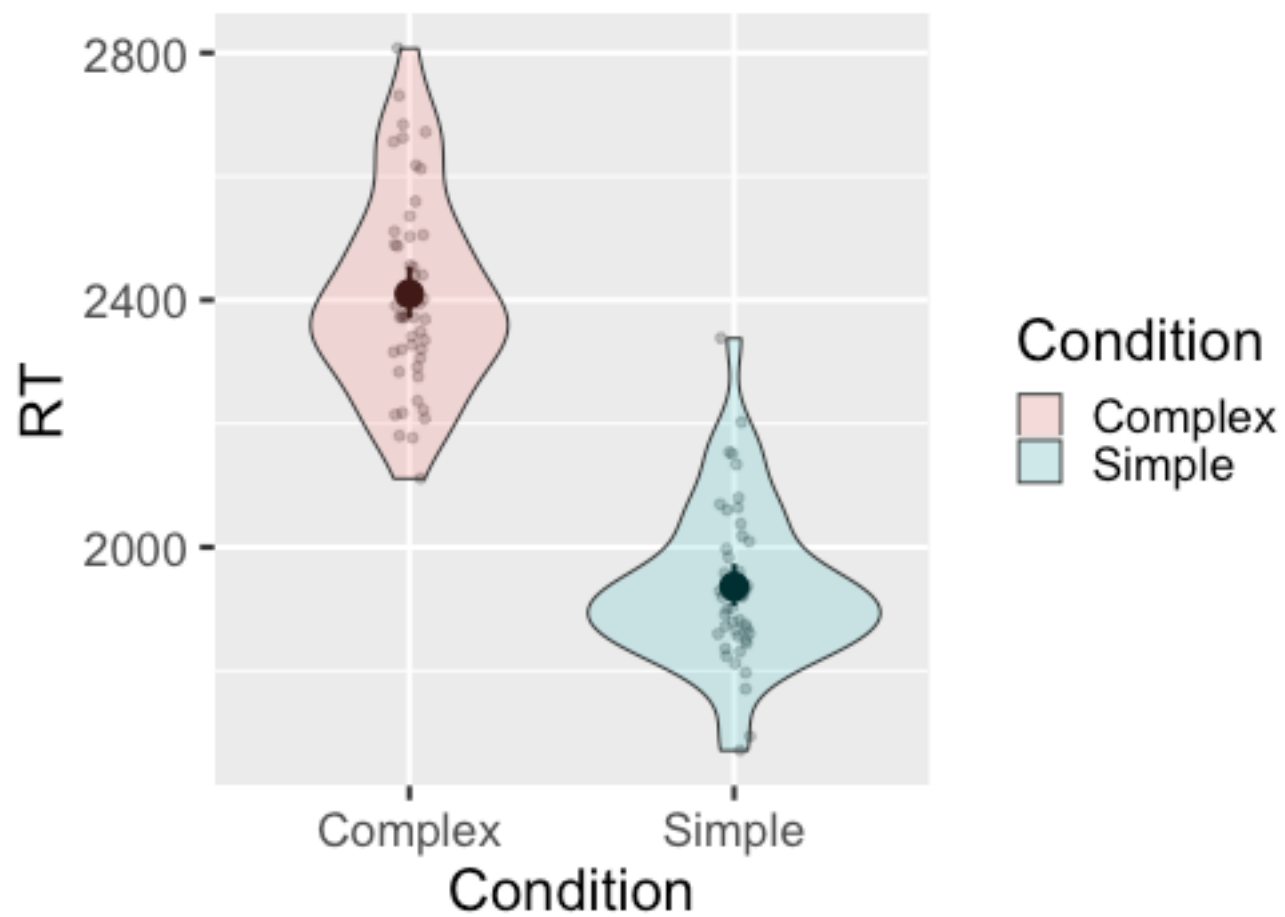
Each plot can be built up gradually, layer by layer like the following:



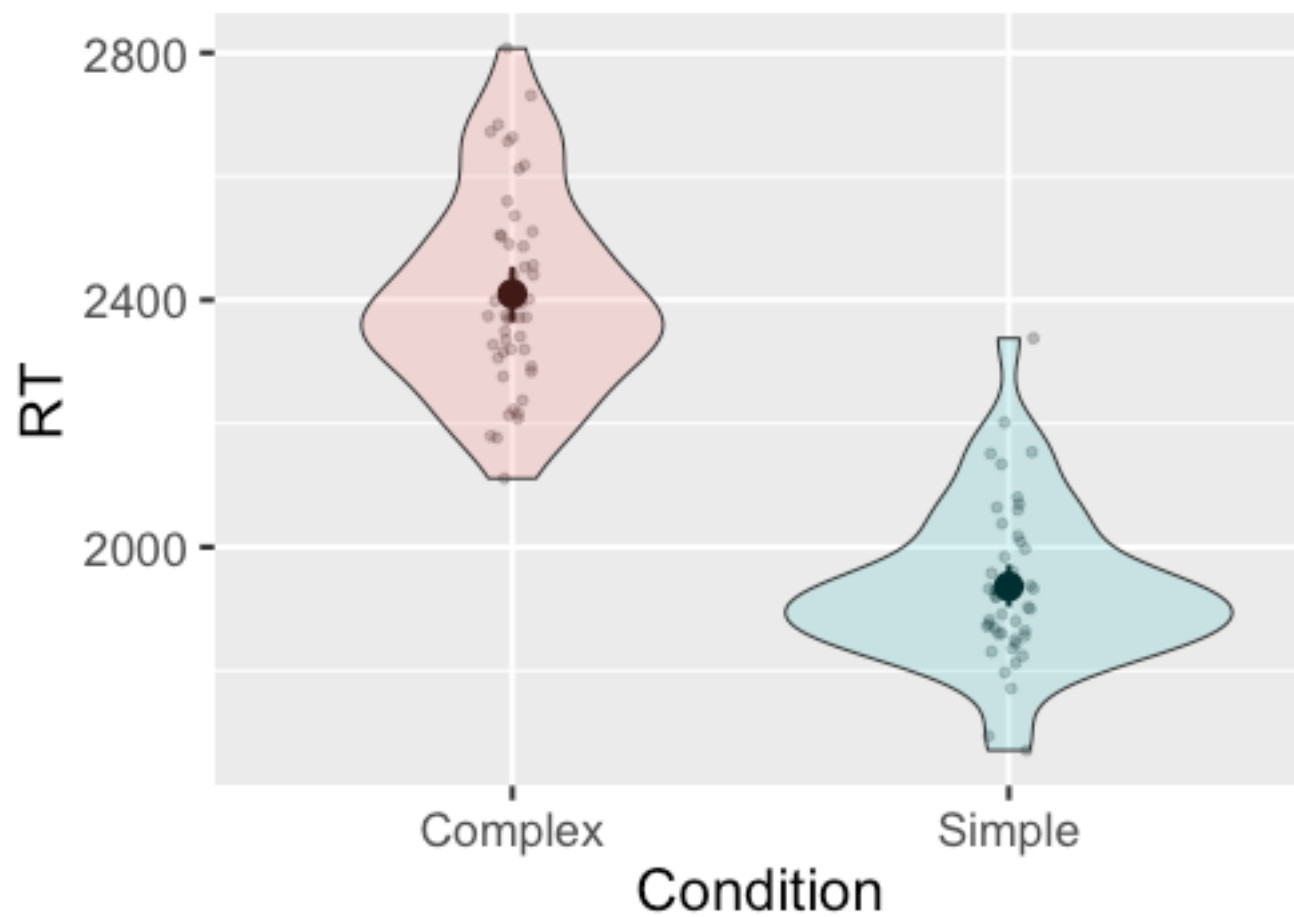
```
ggplot(data_long, aes(x = Condition, y = RT)) +  
  geom_jitter(alpha = .25, position = position_ji
```



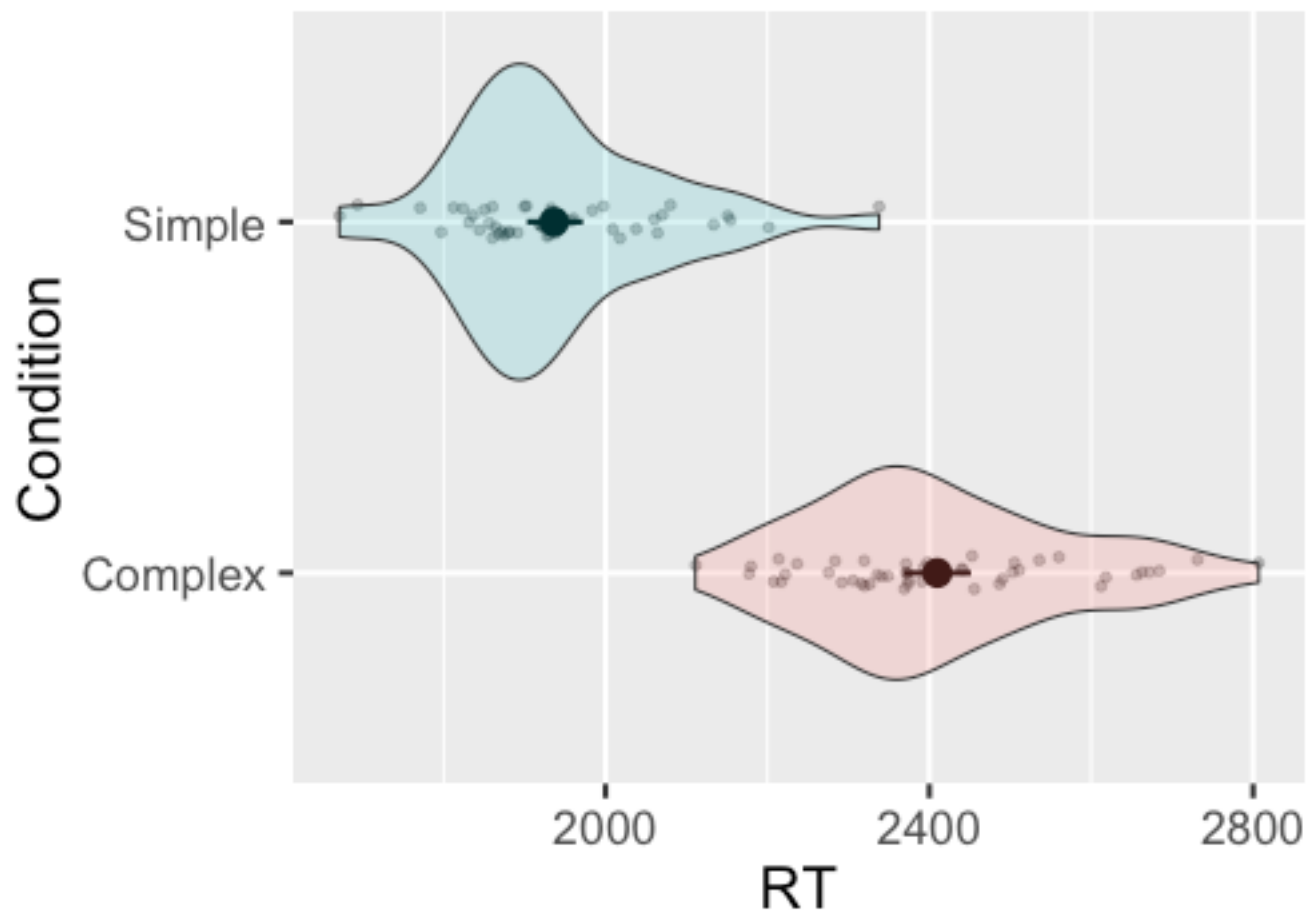
```
ggplot(data_long, aes(x = Condition, y = RT)) +  
  geom_jitter(alpha = .25, position = position_ji  
  stat_summary(fun.data = "mean_cl_boot", colour  
               size = 1)
```



```
ggplot(data_long, aes(x = Condition, y = RT)) +  
  geom_jitter(alpha = .25, position = position_ji  
  stat_summary(fun.data = "mean_cl_boot", colour  
               size = 1) +  
  geom_violin(aes(fill = Condition), alpha = .2)
```

```
ggplot(data_long, aes(x = Condition, y = RT)) +  
  geom_jitter(alpha = .25, position = position_ji  
  stat_summary(fun.data = "mean_cl_boot", colour  
               size = 1) +  
  geom_violin(aes(fill = Condition), alpha = .2)  
  guides(fill = FALSE)
```

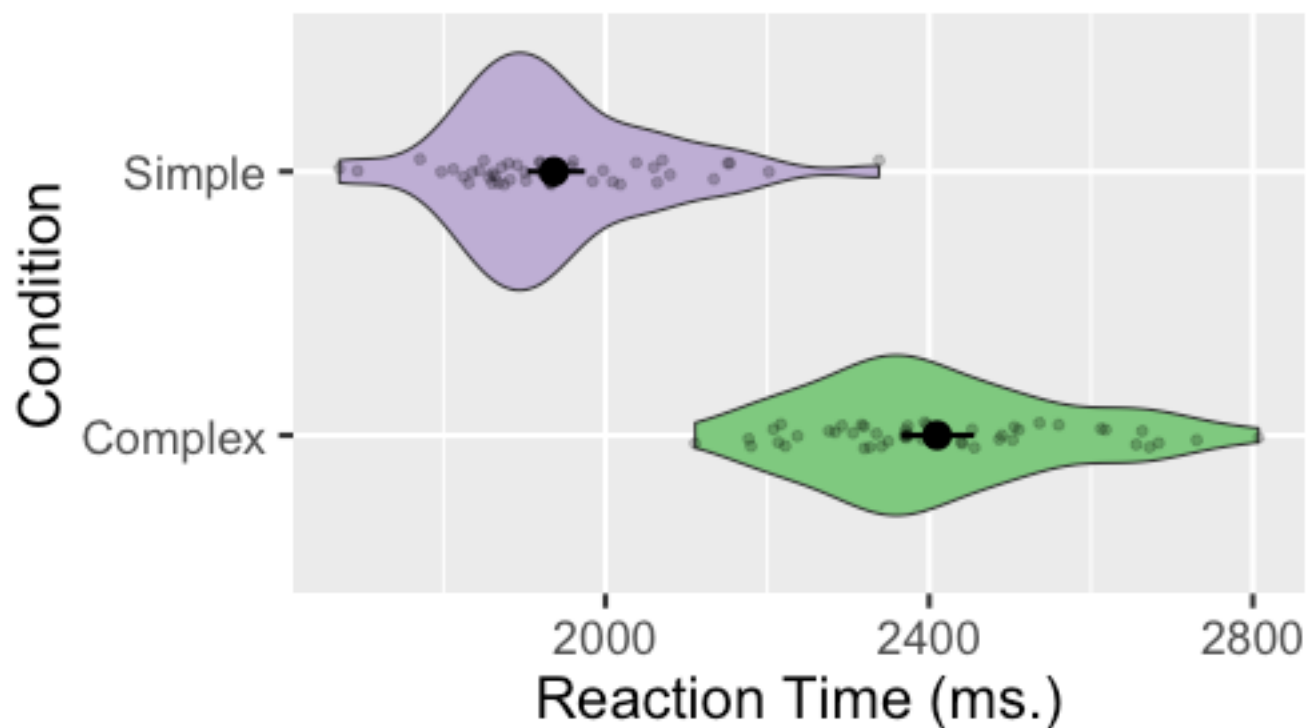


```
ggplot(data_long, aes(x = Condition, y = RT)) +  
  geom_jitter(alpha = .25, position = position_ji  
stat_summary(fun.data = "mean_cl_boot", colour  
              size = 1) +  
  geom_violin(aes(fill = Condition), alpha = .2)  
guides(fill = FALSE) +  
coord_flip()
```

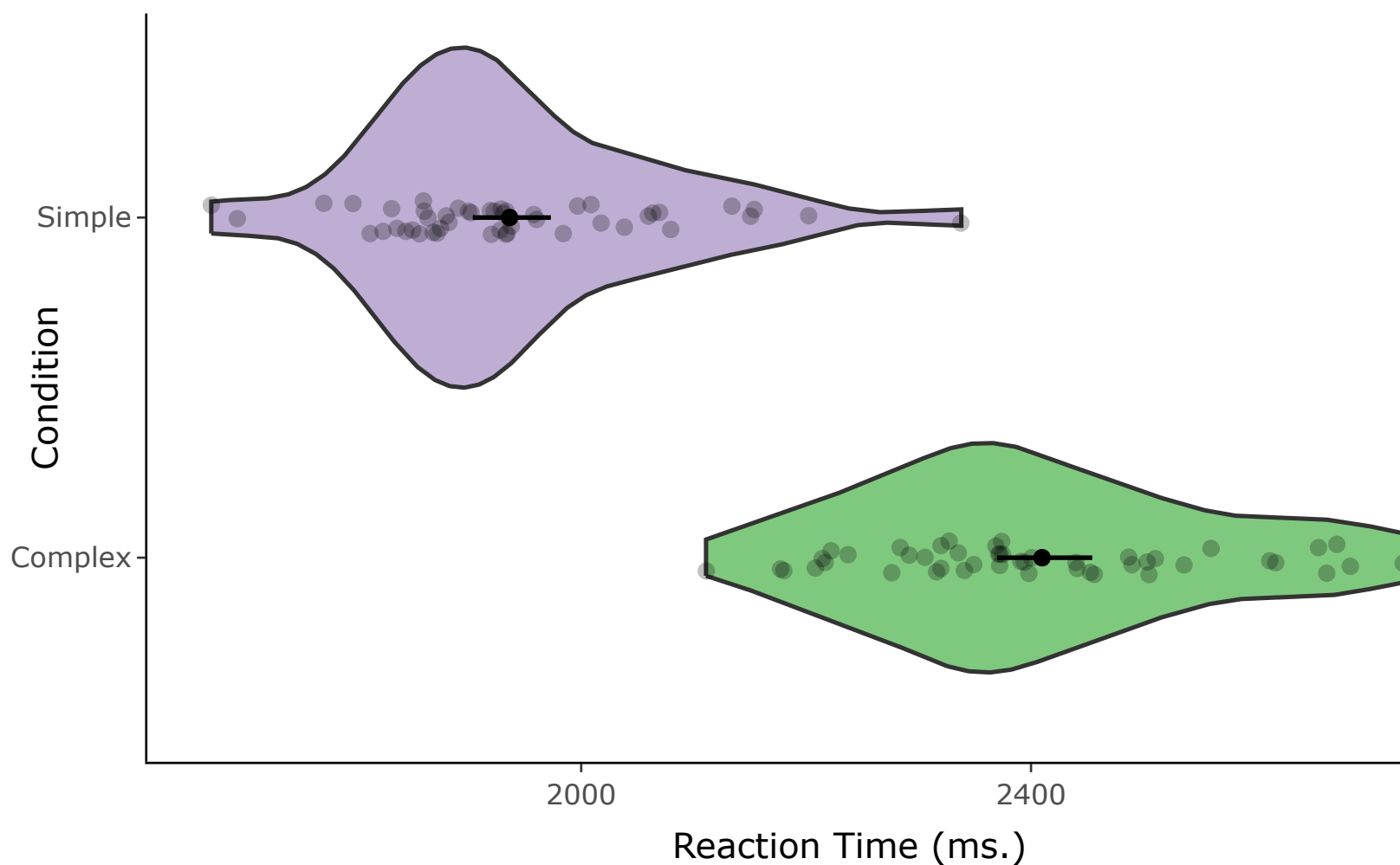
Violin Plots

These are Violin Plots - these are an example of an RDI plot as they capture the Raw data, information about the Distribution, and some Inferential statistics (e.g., Confidence Intervals).

We can modify other characteristics of the plot such as the colour palette we're using, the orientation, and we can also add some labels:



Building interactive visualisations using the plotly package



Raincloud Plots

You might have noticed the violin plots have a little redundant information - the shape of the distribution is plotted, but it's also mirrored (which we don't really need).

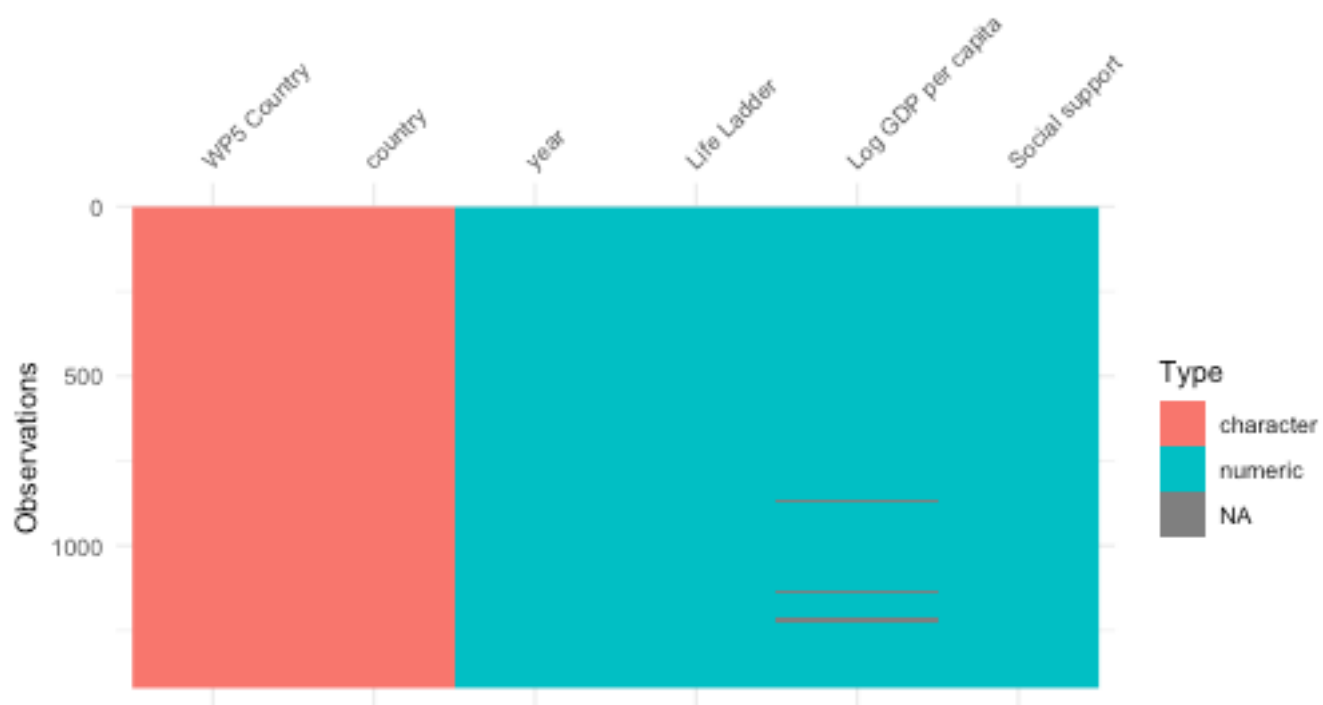
Using different themes

The BBC Cookbook

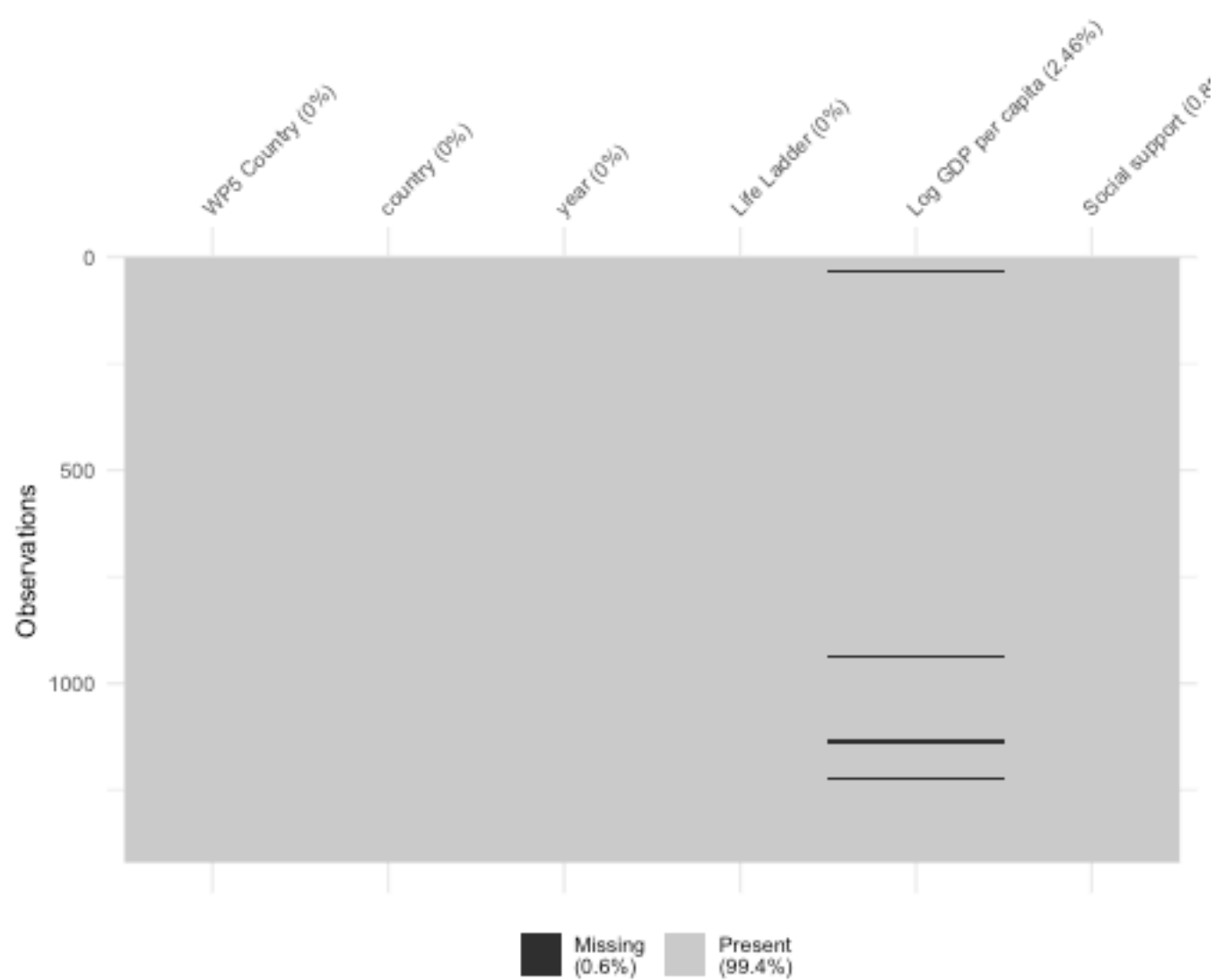
World Happiness Data

We can have a look at the World Happiness dataset that measures Happiness (called Life Ladder) and a bunch of other things (e.g., GDP) over countries over time.

```
vis_dat(happy_data)
```

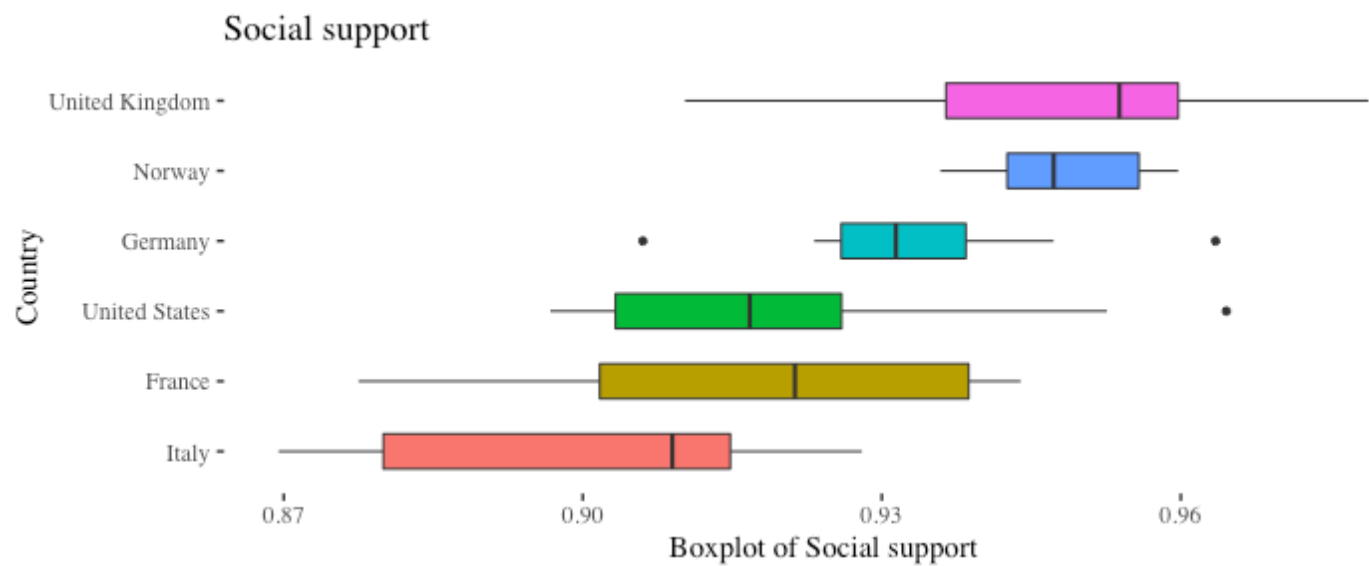



```
vis_miss(happy_data)
```



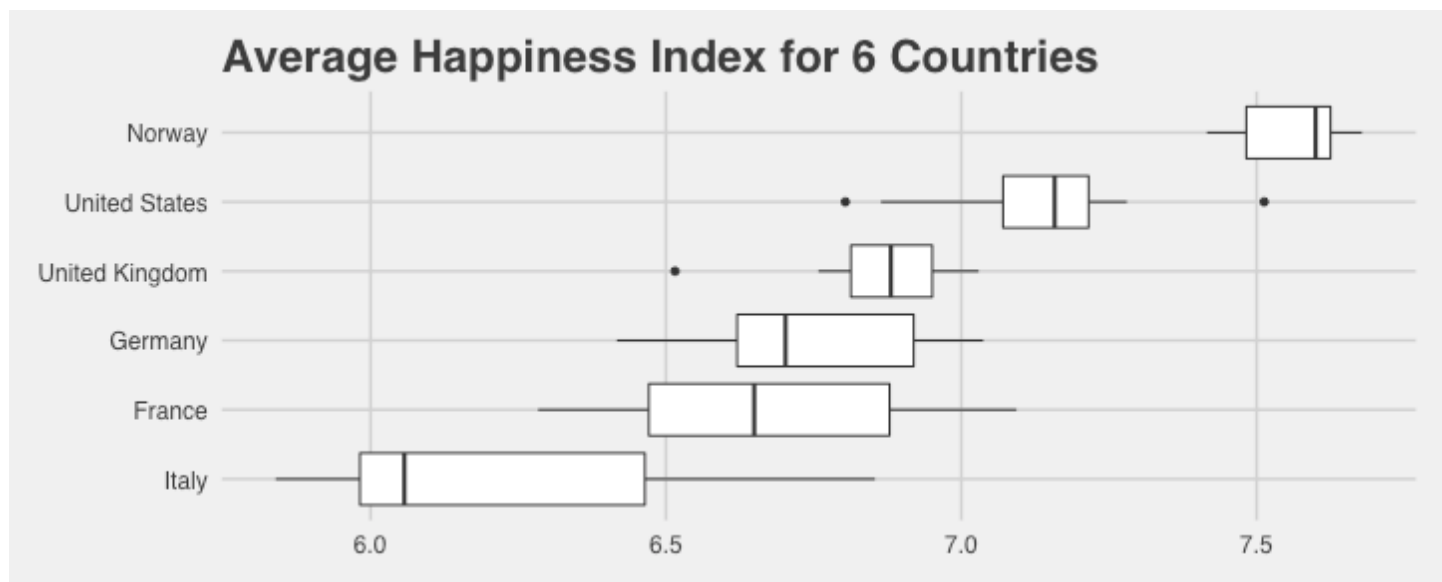


```
happy_data %>%
  group_by(country) %>%
  filter(!is.na(`Life Ladder`) & year == 2016) %>%
  summarise(score = `Life Ladder`) %>%
  mutate(country = reorder(country, score)) %>%
  top_n(20) %>%
  ggplot(aes(x = score, y = country)) +
  geom_point() +
  labs(x = "Happiness Index Score", y = "Country")
  title = "Top 20 Happiest Countries in 2016"
  theme_tufte(base_size = 15)
```

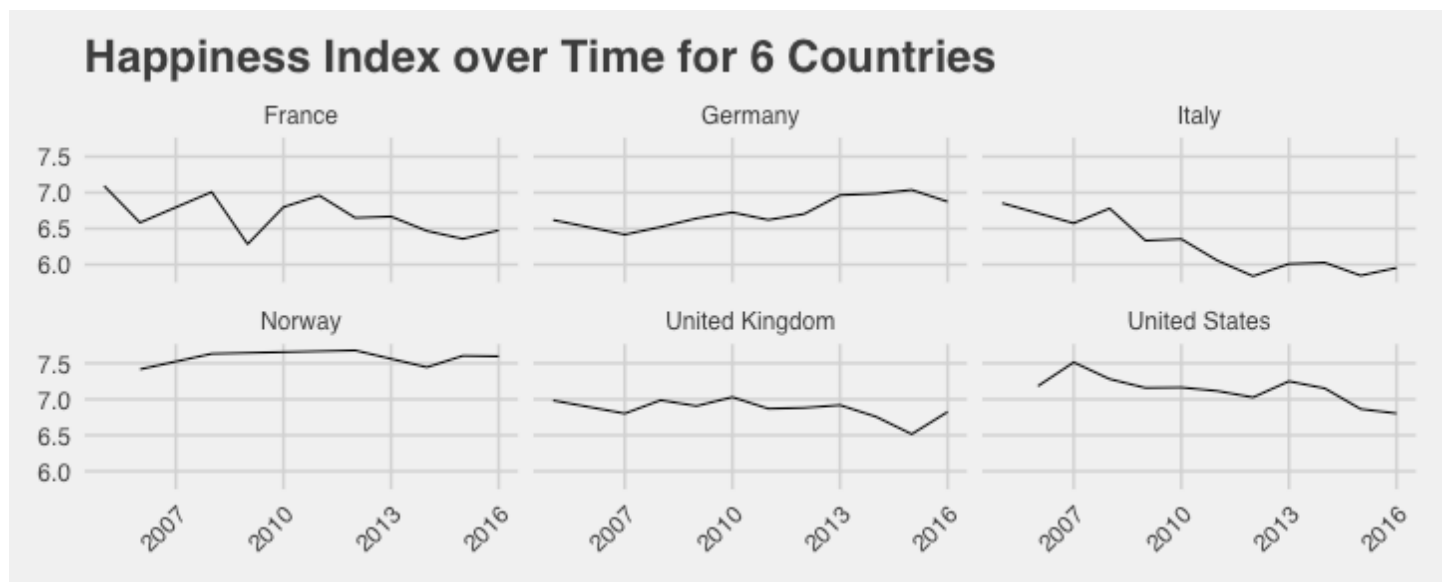


```
country_list <- c("United Kingdom", "France", "Germany",
                  "Italy", "Norway", "United States")

happy_data %>%
  filter(country %in% country_list) %>%
  filter(!is.na(`Social support`)) %>%
  mutate(score = `Social support`) %>%
  mutate(country = reorder(country, score)) %>%
  ggplot(aes(y = score, x = country, fill = country)) +
  geom_boxplot(width = .5) +
  labs(y = "Boxplot of Social support", x = "Country",
       title = "Social support") +
  guides(fill = FALSE) +
  coord_flip() + theme_tufte(base_size = 15)
```



```
happy_data %>%  
  filter(country %in% country_list) %>%  
  group_by(country) %>%  
  mutate(score = `Life Ladder`) %>%  
  ungroup() %>%  
  mutate(country = reorder(country, score)) %>%  
  ggplot(aes(x = country, y = score)) +  
  geom_boxplot() +  
  labs(x = "Country", y = "Happiness Index Score"  
        title = "Average Happiness Index for 6 Cou  
  coord_flip() + theme_fivethirtyeight(base_size
```

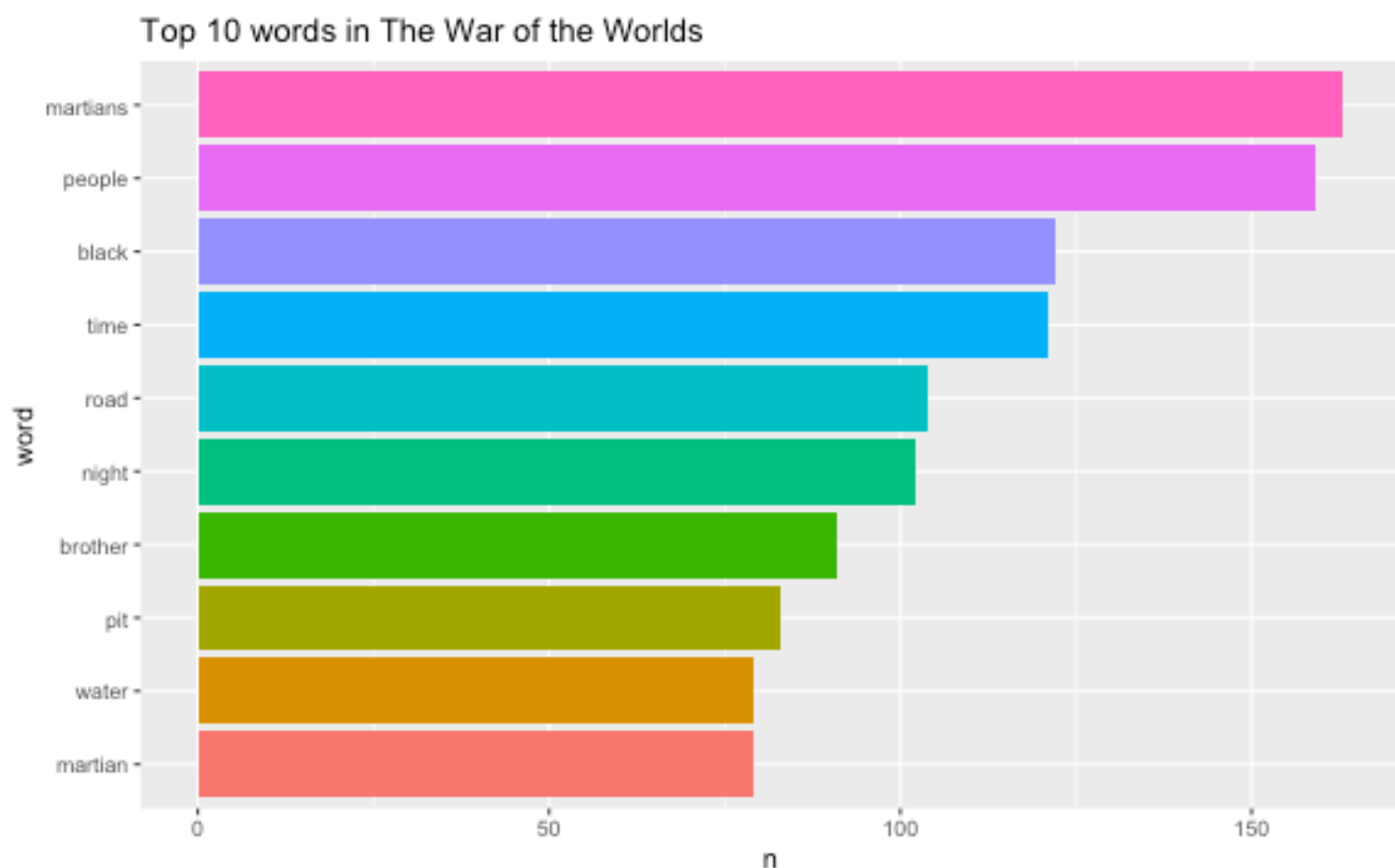


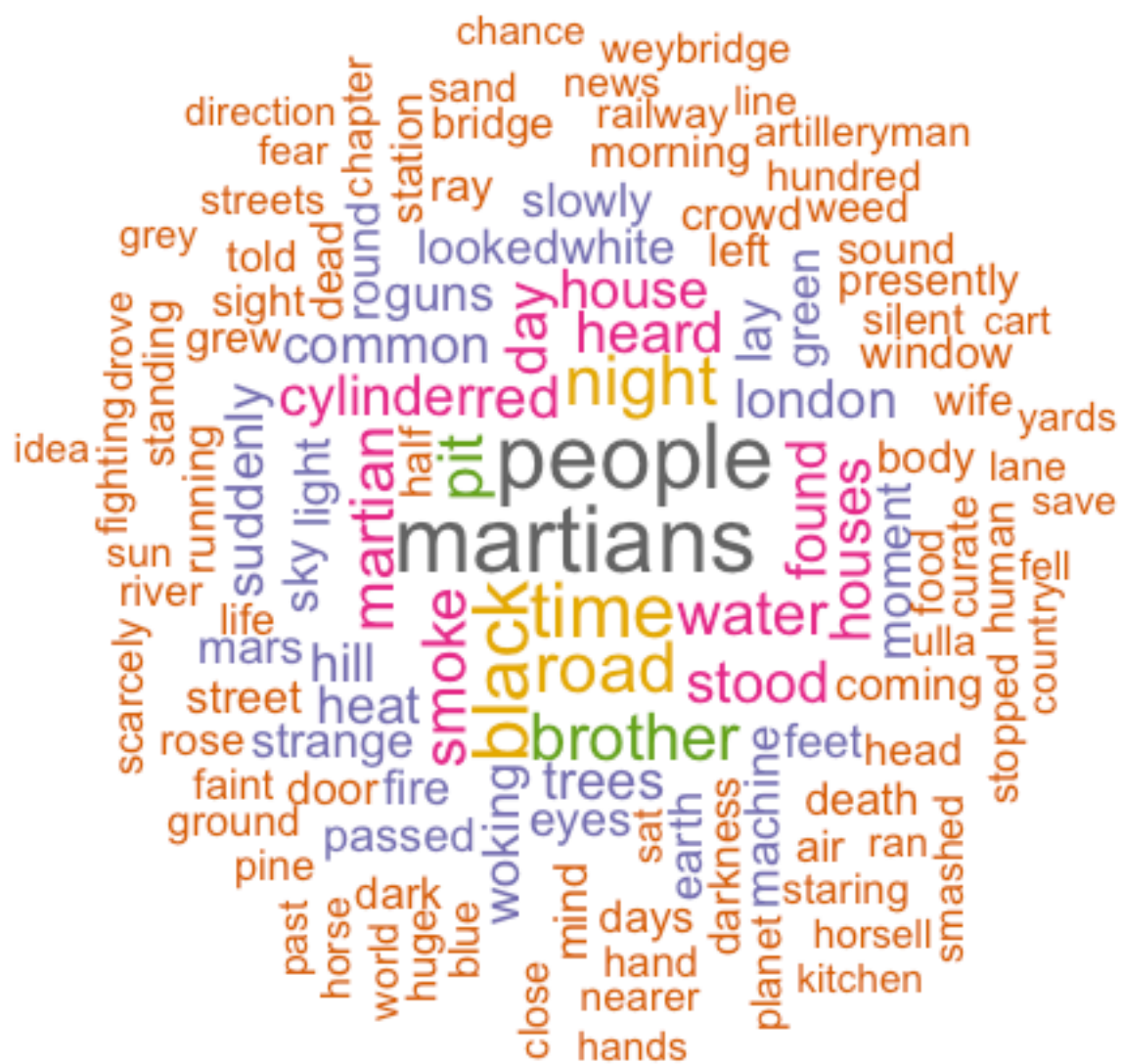
```
country_list <- c("United Kingdom", "France", "Germany",
                  "Italy", "Norway", "United States")

happy_data %>%
  filter(country %in% country_list) %>%
  group_by(year) %>%
  filter(!is.na(`Life Ladder`)) %>%
  ggplot(aes(x = year, y = `Life Ladder`)) +
  geom_line() +
  facet_wrap(~ country) +
  labs(x = "Year", y = "Happiness index",
       title = "Happiness Index over Time for 6 Countries") +
  theme_fivethirtyeight(base_size = 15) +
  theme(axis.text.x = element_text(angle = 45, justify = "center"))
```

Visualising Qualitative Data

Maybe you have lots of qualitative data and are interested in running a content analysis. In the next example, I'm examining all the text in HG Wells' *The War of the Worlds*.





```

# Get 2 HG Wells books ####
titles <- "The War of the Worlds"
books <- gutenbergs_works(title %in% titles) %>%
  gutenbergs_download(meta_fields = "title")

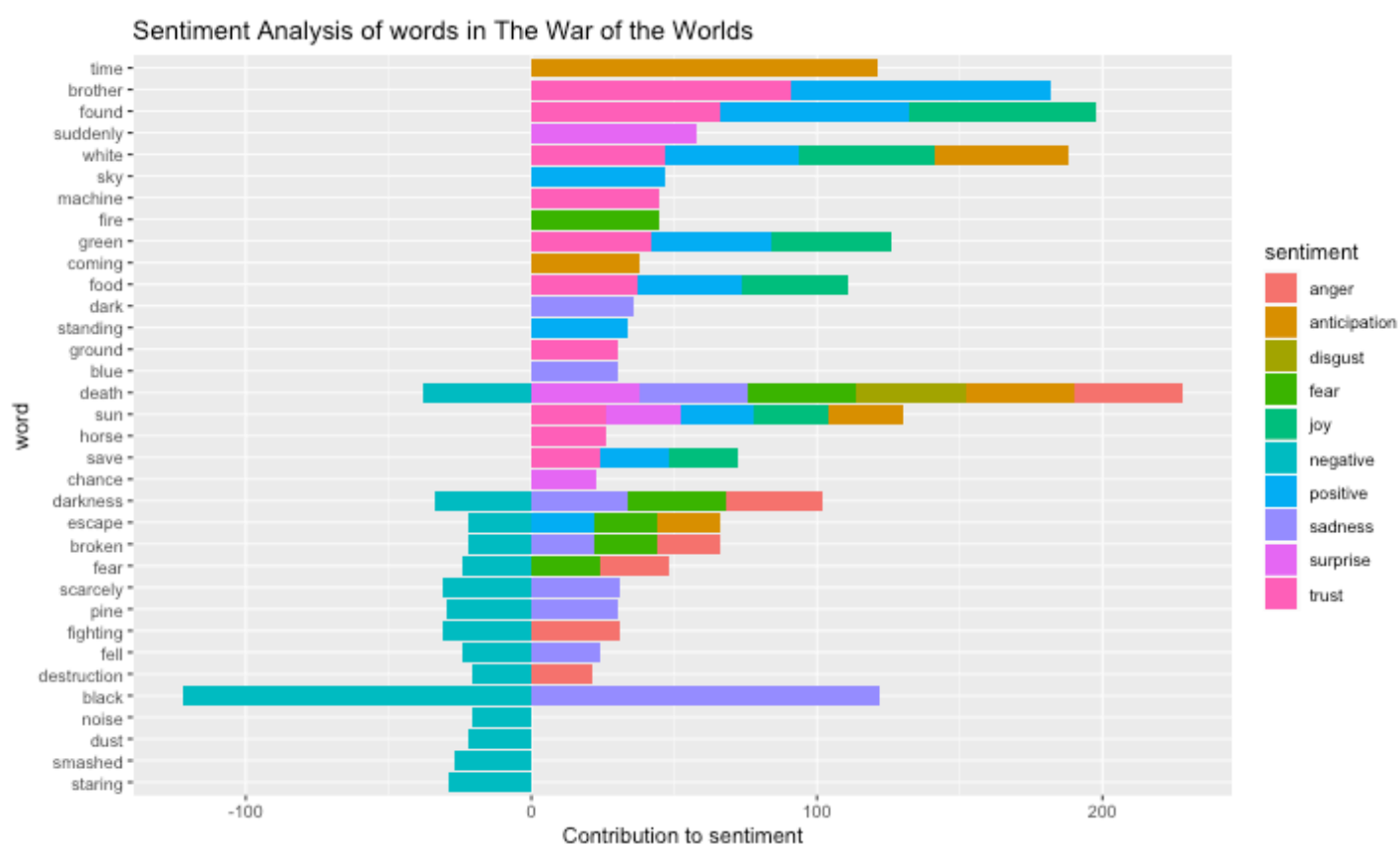
text_waroftheworlds <- books %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)

text_waroftheworlds %>%
  count(word) %>%
  top_n(10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n, fill = word)) +
  geom_col() +
  coord_flip() +
  guides(fill = FALSE) +
  labs(title = "Top 10 words in The War of the Wo

text_waroftheworlds_count <- text_waroftheworlds
  count(word) %>%
  top_n(200)

```


Sentiment Analysis

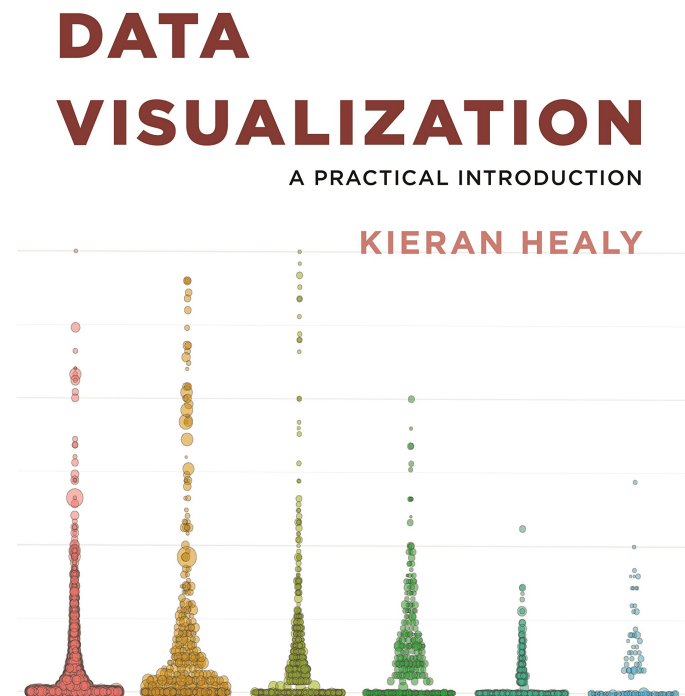
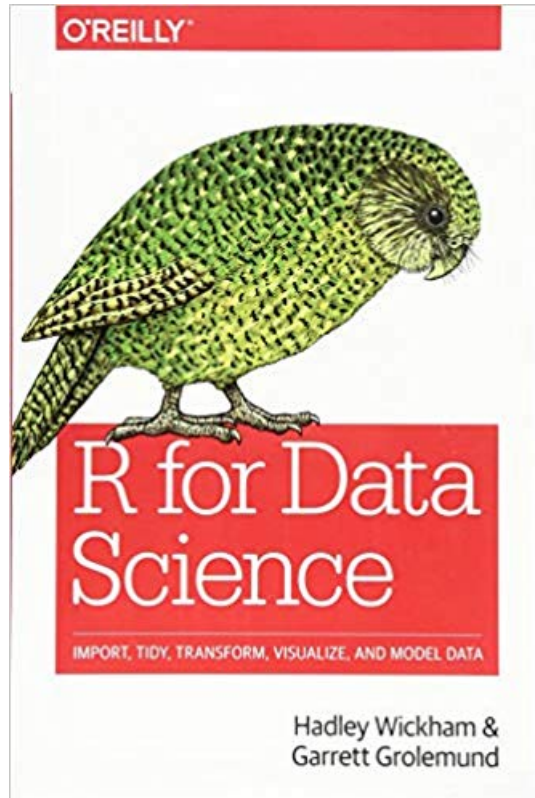


```
sentiments <- get_sentiments("bing")

word_counts <- text_waroftheworlds %>%
  inner_join(sentiments) %>%
  count(word, sentiment, sort = TRUE)
```

```
word_counts %>%
  filter(n > 20) %>%
  mutate(n = ifelse(sentiment == "negative", -n,
    mutate(word = reorder(word, n)) %>%
    ggplot(aes(word, n, fill = sentiment)) +
    geom_col() +
    coord_flip() +
    labs(y = "Contribution to sentiment",
      title = "Sentiment Analysis of words in The W
```


Interested in finding out more?



<https://r4ds.had.co.nz>

<https://kieranhealy.org/publications/dataviz/>

Thanks!



Software
Sustainability
Institute



A Fully Reproducible Talk (just add my accent)

All slides and R code used to generate these slides available here:

XXXXXX



Slides created via the R package **xaringan**, **knitr**, and **R Markdown**.