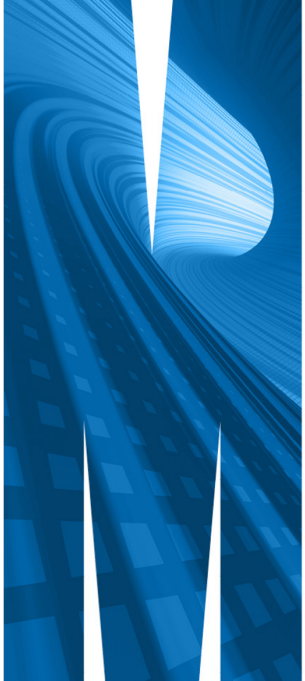
MONASH University

Introduction

Statistical Thinking (ETC2420 / ETC5242)

Week 1, Semester 2, 2025



Outline

1. What is statistics?
2. Info about ETC2420 / ETC5242
3. R, RStudio, tidy data

Outline

What is statistics?

Some examples:

- Weather forecasts: Bureau of Meteorology
- Poll aggregation: FiveThirtyEight (R.I.P. March 2025)
- Climate change modelling: Australian Academy of Science
- Discovery of the Higgs Boson (the ‘God Particle’): van Dyk (2014)
- Smoking leads to lung cancer: Doll & Hill (1945)
- A/B testing for websites

Damjan’s examples:

- Statistical genomics
- Web analytics
- Skin texture image analysis

Goals of statistics

- Answer questions using **data**
- Evaluate **evidence**
- Make **decisions**

And, importantly:

- Clarify **assumptions**
- Quantify **uncertainty**
- Optimise **study design**

Why study statistics?

"The best thing about being a statistician is that you get to play in everyone's backyard."
—John W. Tukey (1915–2000)

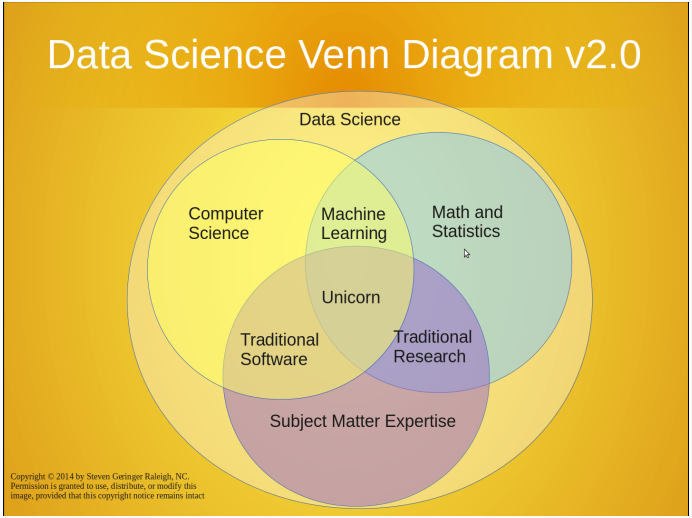
"I keep saying the sexy job in the next ten years will be **statisticians**. . . The ability to take data – to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it's going to be a hugely important skill in the next decades. . . "
—Hal Varian, Google's Chief Economist, Jan 2009

Empoyment opportunity

Highest-Paying Business Majors in 2025:

1. HR Management
2. **Business Analytics**
3. Entrepreneurship
4. Finance
5. ...

Data science, analytics, machine learning, AI,…



Data science is a ‘team sport’

Outline

1. What is statistics?
2. Info about ETC2420 / ETC5242
3. R, RStudio, tidy data

Teaching team

Chief Examiner & Lecturer

- Damjan Vukcevic

Head tutor

- Lachlan Macquarie

Tutors

- Jayani Piyadi Gamage
- Maliny Po
- Nolan Pham
- Thomas Nguyen

Classes

Seminars (“Lectures”)

- Learn new concepts and techniques
- See examples of these in action
- Discuss and ask questions

Workshops

- See more examples
- Start doing some exercises
- Discuss and ask questions

Tutorials

- Tackle more exercises
- Work in small groups
- Get help from your tutor

Tutorials

- Attend your **allocated** class only
- To change classes, use Allocate+
- Teaching staff cannot change your timetable

Moodle

- We will make announcements on Moodle (so check regularly)
- All content and info will be on Moodle
- Updated throughout the semester

Discussion forum ('Ed Discussion')

Find the link on Moodle.

Use it for:

- Questions about **content** (statistical concepts, R, etc.)
- Questions about **admin** (tutorials, consultations, due dates, etc.)
- Particularly useful for discussing any issues with R or RStudio
- Feel free to answer others' questions
- If you find something cool in R, share it there
- Share any interesting links (e.g. news articles) that are relevant to the topics we are learning.

Contact / Questions

General questions:

- Post on the **discussion forum** on Moodle
- Do not email such questions to staff
- Using the forum allows everyone to benefit from the discussion.

Personal questions (only):

- Use the unit email address: etc2420.clayton-x@monash.edu
- Do **not** send emails directly to teaching staff.

Unit overview

Let’s look on Moodle…

Group assignment

- Assignment 2 will be done in **groups**
- You will be able to select your own group
- But groups must be **within** your tutorial class (as per Allocate+)
- We will include a peer-review component to encourage everyone to contribute.
- (More details later in the semester)

Computing

- You are *expected* to use R and RStudio
- Complete the StartR module and ask questions
- Install the latest versions of both R and RStudio
- Install the most recent version of all packages
- Quickest resource for R help: look online (Google, Stack Exchange, etc.)
- Also: discussion forum and consultations

Quarto

- Use Quarto for tutorial exercises and assessment tasks
- All assignments must be submitted in PDF, which should be generated from Quarto

What is “statistical thinking”?

Some initial answers:

- Using data to answer questions
- Understanding the world through statistical models
- Acknowledging and assessing uncertainty

“Perhaps H. G. Wells was right when he said ‘statistical thinking will one day be as necessary for efficient citizenship as the ability to read and write’!”
—Samuel S Wilks, President of the American Statistical Association in 1951

What can statistics do for us?

Three common goals:

- **Describe**:
  Characterise complex and noisy data using simpler terms.
- **Decide**:
  When uncertain, use data to support decisions.
- **Predict**:
  Anticipate relative chance for potential outcomes of a future random event, based on past data.

We'll consider two broad approaches:

- Frequentist/classical inference
- Bayesian inference

We will learn about...

- Randomness
- Using computers to do statistics
- Using computational tools to **understand** statistical properties
- How to communicate and present statistical results
- Using R and RStudio to carry these out.

Mathematics and computation

- Statistics uses both **mathematics** and **computation**
- Need to learn and use both
- Traditional teaching emphasises mathematics
- Modern teaching emphasises computation
- We follow the second approach
- This means you will need to do some programming (with our help!)
- **Important:** this will require some self-study, the best way to learn programming
- You will also need some mathematical concepts from first year or high school

Programming / Coding

- R is a programming language as well as a data analysis tool.
- Our focus is using R to analyse data and report our findings in a reproducible way.
- This involves writing R code.
- While our focus is not on programming itself, you will nevertheless write a lot of code.
- Will cover some basic programming concepts/techniques.

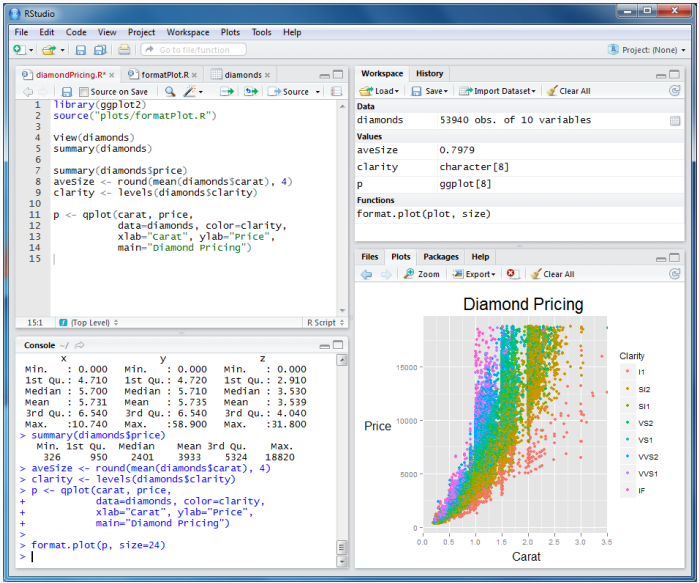
Learning to code (in R)

1. **StartR**. Self-paced online modules.
2. **Lectures & workshops**. We will show examples of how things work in R, especially in the workshops.
3. **Tutorials**. These will be based around R coding. You will work directly on exercises to learn and practise, with help available from a tutor.
4. **Discussion forum** (via Moodle). Ask questions and get help from your classmates (and us).
5. **Online resources**. Find these using Google, Stack Exchange, etc.

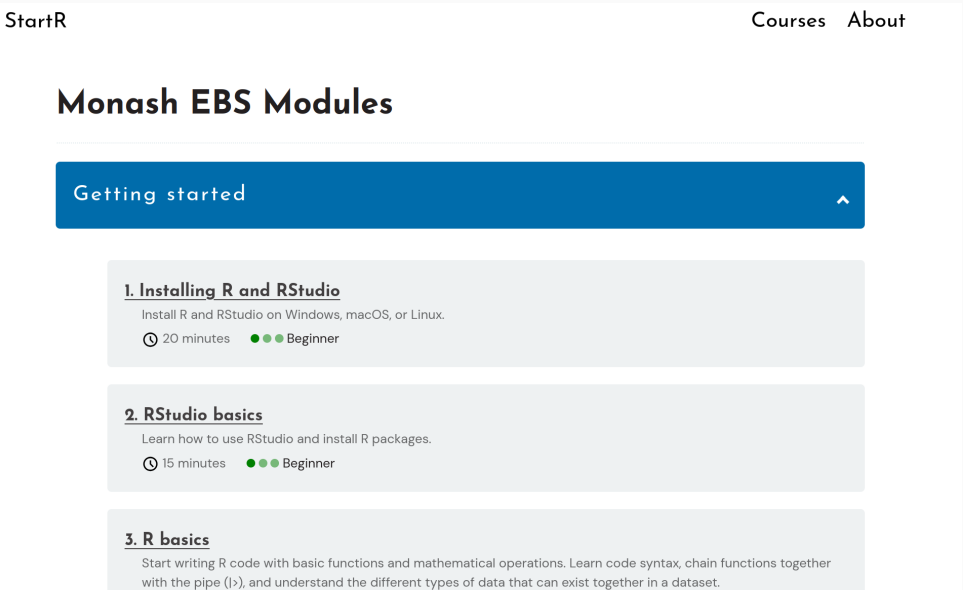
Outline

1. What is statistics?
2. Info about ETC2420 / ETC5242
3. R, RStudio, tidy data

RStudio


RStudio
File Edit Code View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Source
1 library(ggplot2)
2 source("plots/formatPlot.R")
4 view(diamonds)
5 summary(diamonds)
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
11 p <- qplot(carat, price,
12 data=diamonds, color=clarity,
13 xlab="Carat", ylab="Price",
14 main="Diamond Pricing")
15:1 (Top Level) R Script
Console ~/
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
Min. 1st Qu. Median Mean 3rd Qu. Max.
326 950 2401 3933 5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
> format.plot(p, size=24)
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
clarity character[8]
p ggplot[8]
Functions
format.plot(plot, size)
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Carat
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
0
5000
10000
15000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

StartR – https://startr.numbat.space/



Functions in R

- We will introduce functions and loops – these let us repeat the same operations without (re-)writing more code pause
- R has *a lot* of built in functions. They help us do things like computing sums and means and standard deviations.
- Functions have a name and are called using round brackets `name()`.
- Inside the round brackets, you can pass the function **arguments**.

- For example the `rnorm` function draws a **r**andom number from a **norm**al distribution.
- It has up to three arguments: `n`, `mean` and `sd`

```
rnorm(n = 1, mean = 0, sd = 1)
```

```
[1] 0.886855025
```

```
rnorm(1, 0, 1)
```

```
[1] -0.7406252929
```

```
rnorm(1)
```

```
[1] 1.620003025
```

We can make our own functions

```
my_mean <- function(x, n) {
    sum(x)/n
}

my_mean(c(1, 2, 3), 3)
```

```
[1] 2
```

- The `function` command tells us that we are *creating* a function
- The code that is executed every time we call `my_mean()` is inside the *curly braces* `{}`
- (When possible, use the built-in `mean()` function. But this is an example)

Actually there are some problems with `my_mean()`

We can avoid having to input the length of the vector, by computing it *inside the function*.

```
my_mean2 <- function(x) {
    n <- length(x)
    sum(x)/n
}

my_mean2(c(1, 2, 3))
```

```
[1] 2
```

And really...

We don't even need to store the length as a variable

```
my_mean2 <- function(x) {
    sum(x)/length(x)
}

my_mean2(c(1, 2, 3))
```

```
[1] 2
```

Some good practices

1. Indentation is your friend. Keep it consistent.
2. White space is your friend. Put spaces around operators. Later on, we will learn how to let commands go across multiple lines.
3. Good names make code easier to understand.
   (Is `my_mean2()` a good name?)

RStudio will also comment on your *style*

- Coding style is *highly* important
- The code you write will be read by others…and your future self!
- Refer to the tidyverse style guide:
  https://style.tidyverse.org/

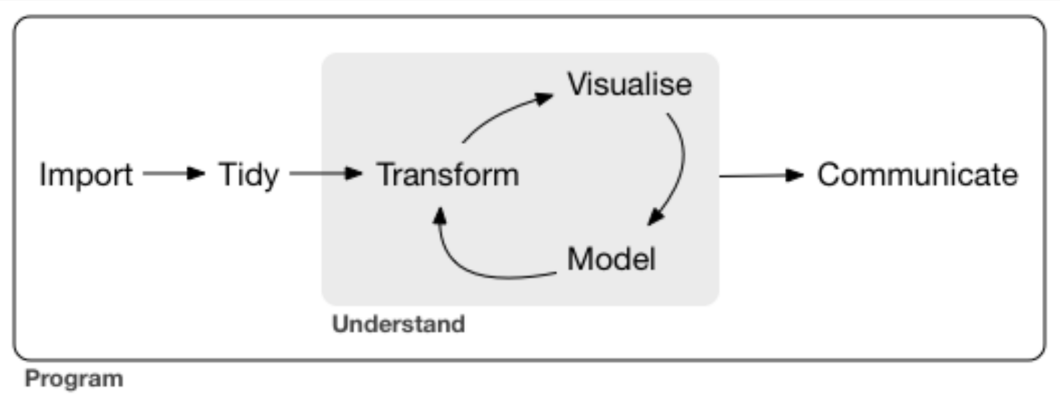
Introduction to tidy data

- The **tidyverse**: an R package, which is itself comprised of many other R packages:
  - ggplot2: data visualisation
  - dplyr: data manipulation
  - tidyr: data organisation
  - readr: data import
  - purrr: function iteration
  - tibble: data storage
  - stringr: string management
  - forcats: categorical data functions

Review the StartR section on tidy data

Why tidy data?

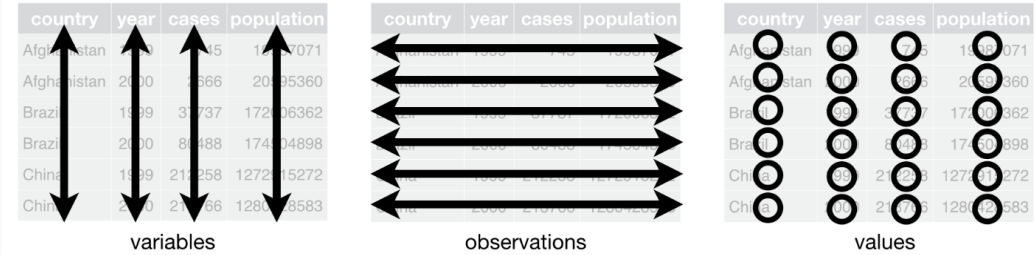
- “A typical data science project” (R4ds)



- Put (rectangular) data in a standard format

Tidy format

- Observations in rows
- Variables in columns
- Values in cells



Advantages of tidy data

Main package is **tidyr**. Advantages of tidy data include:

1. Consistent workflow
   - tools have an underlying uniformity
2. Computational benefits
   - mathematical operations on vectors (variables in columns) are fast
3. All the other tidyverse packages work with tidy data
   - tibble
   - ggplot2
   - dplyr

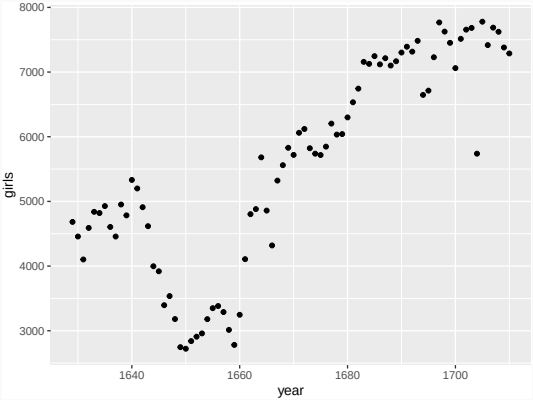
ggplot2

- Making beautiful plots is easy with `ggplot2`
- Create data visualisations based on “The Grammar of Graphics”
- You provide:
  - the data
  - the mapping of variables to “aesthetics”
  - desired geometric objects (“geoms”) to define the type of plot
  - additional layers/options.
- Produces an object that can be stored, printed and even modified later.

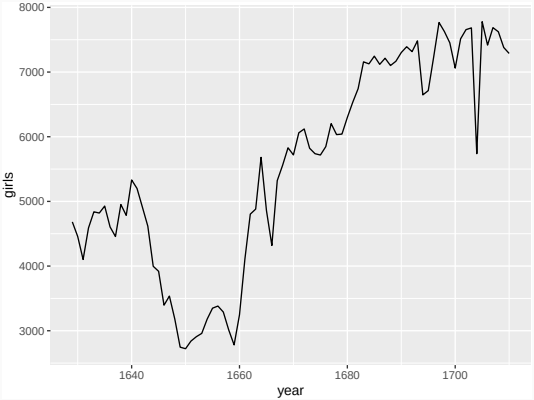
A simple (familiar for some) ggplot2 example

```
p1 <- ggplot(data = arbuthnot, aes(x = year, y = girls)) + geom_point()
p2 <- ggplot(data = arbuthnot, aes(x = year, y = girls)) + geom_line()
```

```
print(p1)
```

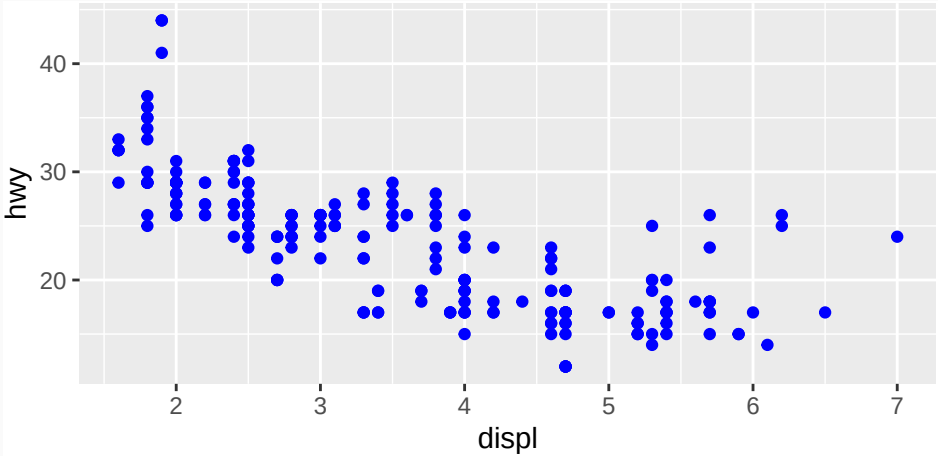



```
print(p2)
```



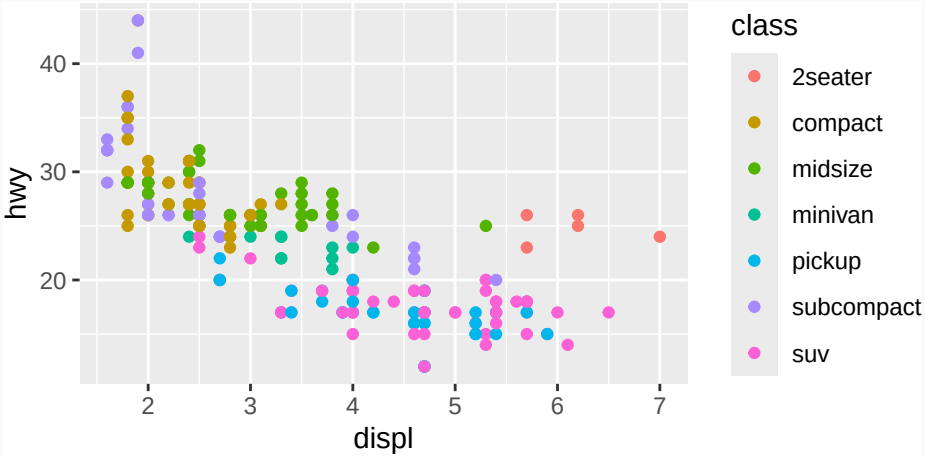
A simple ggplot2 example

```
ggplot(mpg, aes(displ, hwy)) + geom_point(colour = "blue")
```



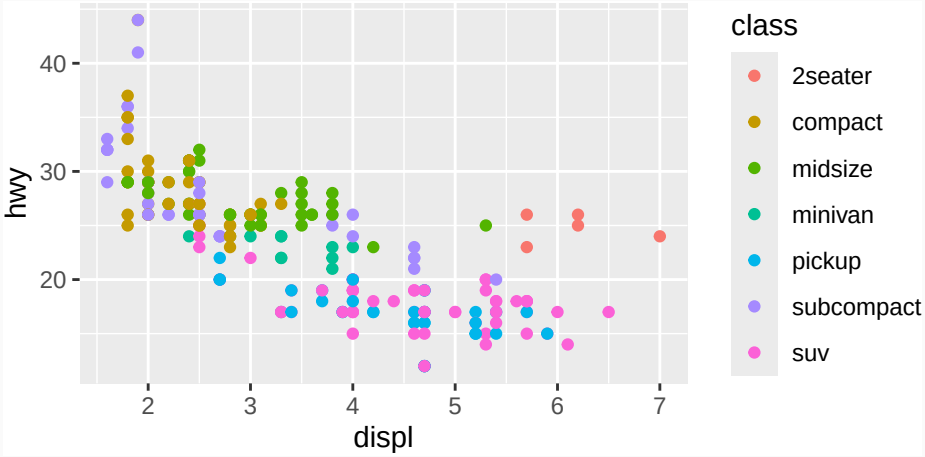
A simple ggplot2 example

```
ggplot(mpg, aes(displ, hwy, colour = class)) + geom_point()
```



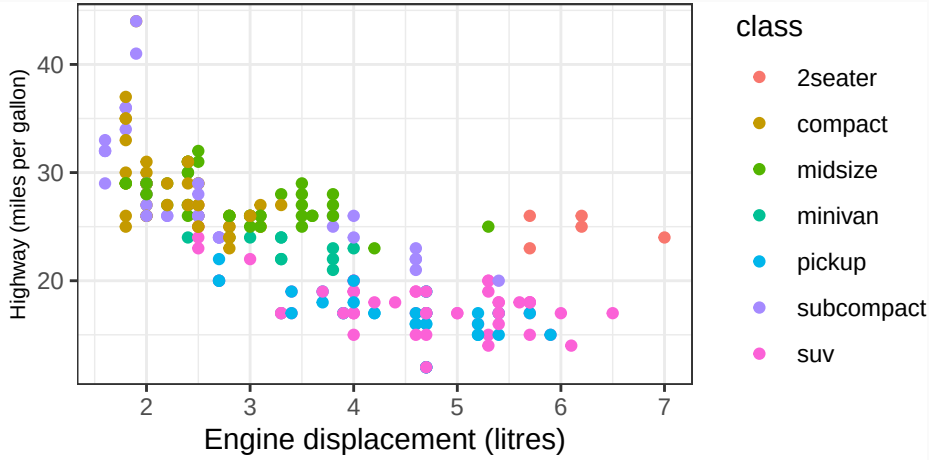
A simple ggplot2 example

```
mpg |>
    ggplot(aes(displ, hwy, colour = class)) + geom_point()
```



A simple ggplot2 example

```
ggplot(mpg, aes(displ, hwy, colour = class)) + geom_point() +
    xlab("Engine displacement (litres)") + ylab("Highway (miles per gallon)") +
    theme_bw() + theme(axis.title.y = element_text(size = 8))
```



A simple ggplot2 example

Did you notice the following?

- Different way to use the `colour` argument
- Automatic legend
- Use of the *pipe* (`|>`)
- Customising axis labels, themes, etc.

See more options in the *R Graphics Cookbook*: https://r-graphics.org/

Data wrangling

- Getting messy data into a standard format is known as **wrangling**
- Having a standard format keeps new variable definitions consistent
- Three main parts to data wrangling: Import $\rightarrow$ Tidy $\rightarrow$ Transform

Importing data

- Use `readr` package functions, e.g. `read_csv()`
- Use `readxl` package (not in **tidyverse**), e.g. `read_excel()`

Many other packages and functions are useful for data that is really messy

- See Chapters 9–16 in R for Data Science for many important tips

“Tidying” up and transforming data

- Put data into a *tibble*
- Use `tidyr` functions to reshape your tibble
  - `pivot_longer()` can stack columns
  - `pivot_wider()` can unstack columns
  - See `vignette("tidy-data")` and `vignette("pivot")` to learn more.
- Use `dplyr` “verb” functions to manipulate data in your tibble
  - `filter()`, `slice()`, `arrange()`, `desc()` work on rows
  - `select()`, `rename()`, `mutate()`, `relocate()` work on columns
  - `summarise()` collapses a group into a single row
  - See `vignette("dplyr")` to learn more.
- The “pipe” operator (`|>`) can neatly tie these into a sequence of steps.

Vignettes

Most good R packages come with one or more **vignettes**.

- A tutorial to help potential users learn how to use a package
- Find list of vignettes for a package from package site:
  - https://cran.r-project.org/web/packages/
  - sort by package name
  - sort by package date
- Consider
  - https://cran.r-project.org/web/packages/tidyverse/
  - https://cran.r-project.org/web/packages/tidyr/
  - https://cran.r-project.org/web/packages/dplyr/

Vignettes

Vignettes will typically include

- An introduction, explaining motivation, or a rational
- Easy-to-replicate examples
- Details of individual functions and available options
- Alerts to conflicts with functions from other packages

Workshop & tutorials

- We will explore some R examples with ggplot and data wrangling
- Practise skills you will need throughout the semester
- These will feature in your first assignment