MONASH University

MONASH
BUSINESS
SCHOOL

Statistical Thinking (ETC2420/ETC5242)

Associate Professor Catherine Forbes

Week 2: Introduction to data

Week 1 Learning Goals

The learning goals for Week 1:

- Learn how to set up **R** and **RStudio** on your own device.
- Learn to install and load **R** packages.
- Learn what are **R Markdown** files and reproducible research.
- Learn what is ‘the tidyverse’.
- Learn some basic **R** commands to manipulate and plot data.

Week 2 Learning Goals

The learning goals for Week 2:

- Identify types of variables, summarise them appropriately, and characterise relationships between them.
- Describe scientific data collection principles.
- Classify variables as being numerical or categorical.
- Illustrate ‘tidy data’ organisational principles.
- Produce descriptive summaries of numerical and categorical data using appropriate ggplot2, tidyr and dplyr functions.

Assigned reading

Assigned reading for Week 2:

- Chapter 1 in ISRS (prescribed textbook: *Introductory Statistics with Randomization and Simulation*)
- Chapters 1, 3 and 12 in *R for Data Science*, especially:
  - Sections 1.1 - 1.2
  - Sections 3.1 - 3.4
  - Section 3.6
  - Sections 12.1 - 12.3

What is statistical thinking?

- A way of understanding a complex world. . .
  - using simple terms for essential structure,
  - acknowledging and assessing degree of uncertainty,
  - based on foundations from maths, stats, computer science, psychology, and more
- Using data to challenge intuition

“*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 major things:

- **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 all of these from the perspective of:

- **Frequentist thinking** based on randomisation and simulation
- **Bayesian thinking** using subjective probabilities

But first, let's just focus on the **data**

Upcoming video book chapters

- First 3 learning goals covered by videos produced by the ISRS authors
  - short and clear
  - follow the textbook sections in Chapter 1
  - correspond to Week 2 video book chapters 2 through 8
- Summary points provided on upcoming slides
- After these, we will pick up again with a focus on the final two learning goals

ISRS Sections 1.1 - 1.2

1.1 Case study - using stents to prevent strokes

- Stents and risk of stroke
- Treatment and control groups
- A data table
- Summary statistics
- Random fluctuation

1.2: Data basics

- Observations, variables and data matrices
- Types of variables
- Relationships between variables

ISRS Section 1.3 - 1.4

1.3 Data collection principles

- Populations and samples
- Sampling from a population
- Explanatory and response variables
- Observational studies and experiments

1.4 Observational studies and sampling strategies

- Observational studies
- Simple random sampling
- Stratified sampling

ISRS Sections 1.5 - 1.6

1.5: Experiments

- Control groups and treatment groups
- Randomisation and replication
- Blocking
- Blinding and placebos

1.6 Examining numerical data

- Scatterplots for paired data
- Dot plots and the mean
- Histograms and shape
- Variance and standard deviation
- Box plots, the median, and robust statistics

ISRS Section 1.7

1.7 Considering categorical data

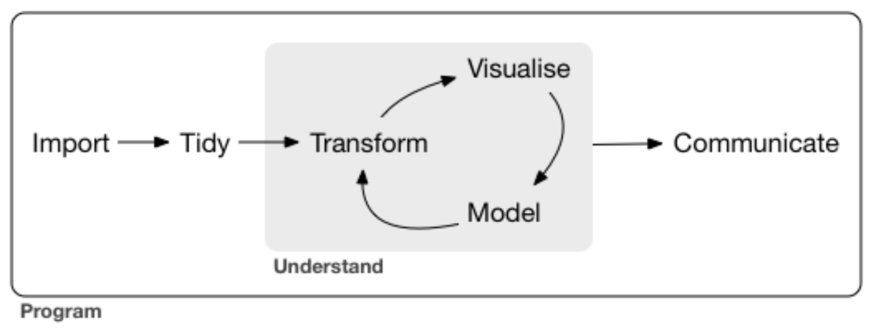
- Contingency tables and bar plots
- Row and column proportions
- Comparing numerical data across groups

Introduction to tidy data

- Recall 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: categorial data functions

Why tidy data?

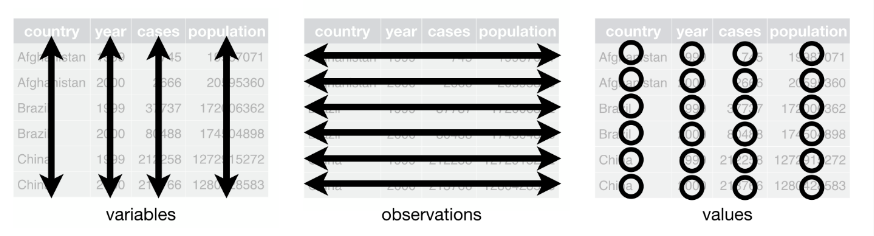
- "A typical data science project" (R4ds)



- Put (rectangular) data in a standard format

Tidy format

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



Which data set is in a ‘tidy’ format?

- All data sets contain WHO counts of TB cases during 1999 and 2000 with population size in 3 countries
- Which data set is in a ‘tidy’ format?

Same data set four ways (table1)

```
# A tibble: 6 x 4
  country      year  cases population
  <chr>       <int>  <int>      <int>
1 Afghanistan  1999    745   19987071
2 Afghanistan  2000   2666   20595360
3 Brazil       1999  37737  172006362
4 Brazil       2000  80488  174504898
5 China        1999 212258 1272915272
6 China        2000 213766 1280428583
```

Which data set is in a ‘tidy’ format?

Same data set four ways (table2)

```
# A tibble: 12 x 4
   country      year type            count
   <chr>       <int> <chr>           <int>
 1 Afghanistan  1999 cases             745
 2 Afghanistan  1999 population   19987071
 3 Afghanistan  2000 cases            2666
 4 Afghanistan  2000 population   20595360
 5 Brazil       1999 cases           37737
 6 Brazil       1999 population  172006362
 7 Brazil       2000 cases           80488
 8 Brazil       2000 population  174504898
 9 China        1999 cases          212258
10 China        1999 population 1272915272
11 China        2000 cases          213766
12 China        2000 population 1280428583
```

Which data set is in a ‘tidy’ format?

Same data set four ways (table3)

```
# A tibble: 6 x 3
  country      year rate
* <chr>       <int> <chr>
1 Afghanistan  1999 745/19987071
2 Afghanistan  2000 2666/20595360
3 Brazil       1999 37737/172006362
4 Brazil       2000 80488/174504898
5 China        1999 212258/1272915272
6 China        2000 213766/1280428583
```

Which data set is in a ‘tidy’ format?

Table4a

```
# A tibble: 3 x 3
  country     ‘1999‘ ‘2000‘
* <chr>        <int>  <int>
1 Afghanistan    745   2666
2 Brazil       37737  80488
3 China       212258 213766
```

Table 4b

```
# A tibble: 3 x 3
  country         ‘1999‘     ‘2000‘
* <chr>            <int>      <int>
1 Afghanistan   19987071   20595360
2 Brazil       172006362  174504898
3 China       1272915272 1280428583
```

Advantages of tidy data

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

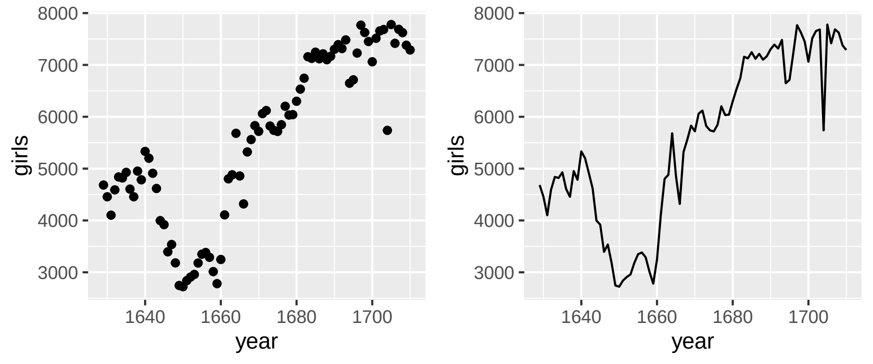
1. Consistent workflow
   - tools have an underlying uniformity
2. Computational benefits
   - mathematical operations on vectors (variables in columns) are fast
3. All the other tidyverse packages work with tidy data
   - tibble
   - ggplot2
   - dplyr

ggplot2

- Making beautiful plots is “easy” with ggplot2
  - you have already used ggplot2 in Lab 1
- Create data visualisations (plots) based on “The Grammar of Graphics”
- You provide
  - the tibble containing the data
  - the mapping of variables to aesthetics
  - desired geom(s), to define the type of plot
  - additional layers: stats, scales, coordinate systems, faceting, position adjustments, labels, and legends
- Produces an object that can be stored, with
  - layers added later
  - printing executed later
  - used as input into other function

A simple ggplot2 example

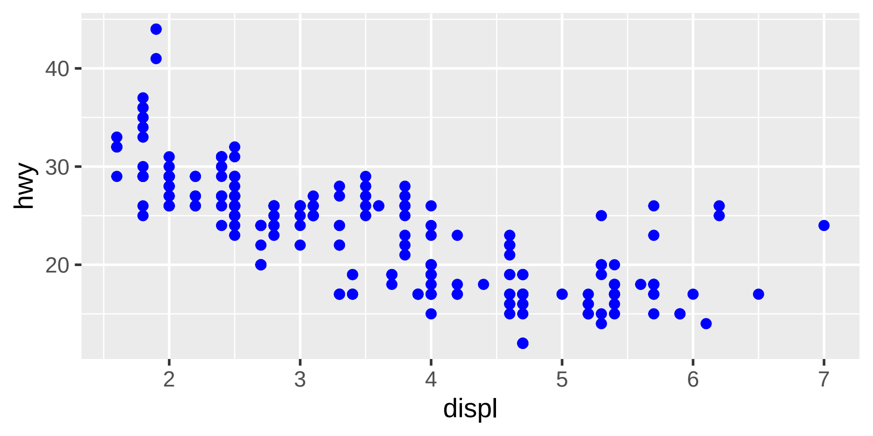
```
library(openintro)
library(gridExtra)
data(arbuthnot)
p1 <- ggplot(data = arbuthnot, aes(x = year, y = girls)) + geom_point()
p2 <- ggplot(data = arbuthnot, aes(x = year, y = girls)) + geom_line()
grid.arrange(p1, p2, ncol = 2)
```



A simple ggplot2 example

- Compare *colour=“blue”* in the geom:
- See *mpg* in **R** help for more information

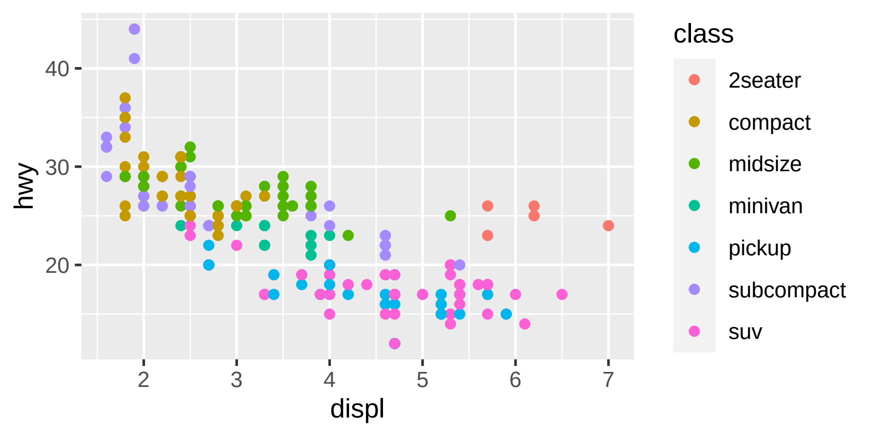
```
ggplot(mpg, aes(displ, hwy)) + geom_point(colour = "blue")
```



A simple ggplot2 example

- With *colour=class* in the aesthetic
  - legend shows automatically
  - notice the *pipe* (%>%)

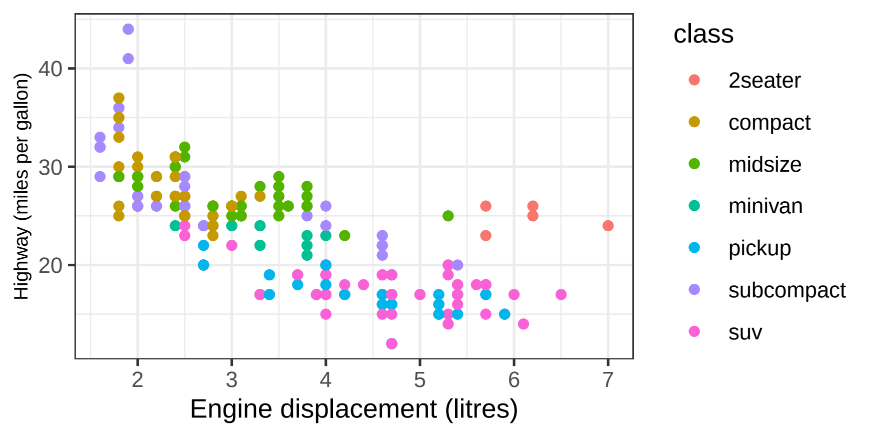
```
mpg %>% ggplot(aes(displ, hwy, colour = class)) + geom_point()
```



A simple ggplot2 example

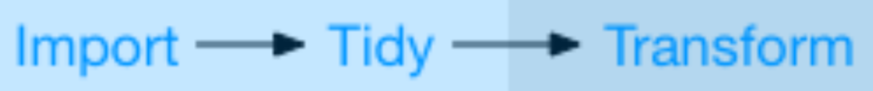
- Change *theme* and size of y-axis label font
- See more options in the R Graphics Cookbook

```
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))
```



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:



To **Import**

- 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

Data wrangling

To **Tidy**

- Put data into a *tibble*

- Use **tidyr** functions to reshape *tibble*
  - *pivot_longer()* can stack columns
  - *pivot_wider()* can unstack columns
  - refer to **vignette("tidy-data")** and **vignette("pivot")** to learn more

- Use **dyplr** "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
  - refer to **vignette("dplyr")** to learn more

- The "pipe" operator (**%>%**) from the **magrittr** package is also useful

Vignettes

Most good **R** packages come with one or more **vignette**

- 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 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

Cheatsheets

RStudio produces **Cheatsheets** for some frequently used activities.

- See `https://rstudio.com/resources/cheatsheets/`
- Also available under the **Help** menu in **RStudio**

You may find these cheatsheets particularly helpful

- Data visualisation (ggplot2)
- Data transformation (dplyr)
- **R Markdown**
- **RStudio**