COMP824 2023 Week 6
Exploratory Data Analysis

Department of Mathematical Sciences
Auckland University of Technology

Overview

The Process of Analytics

Importing Data

Exploratory Data Analysis

Workflow: R Projects

Reading

Chapter 8, 11 Wickham and Grolemund (2020), R for Data Science https://r4ds.had.co.nz/

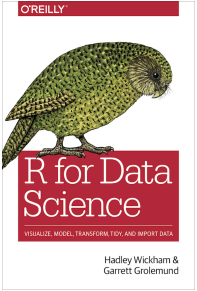
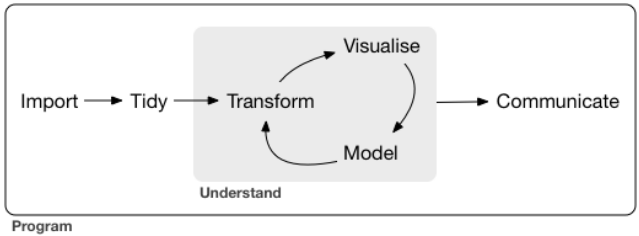


Figure 1: http://r4ds.had.co.nz/

The Process of Analytics



Learning objectives

- Know how to import datasets from a range of file types into R using tidyverse
- Undertake exploratory data analysis using tidyverse
- Understand and apply the key principles of using R projects

Importing data into R

Avoid having to type data into R manually!!

Note: If it seems like a tedious waste of time, it probably is. Someone has probably written an R package to streamline the process and ease your pain).

There are several packages available for reading data into R.

R packages for importing data: `tidyverse` and `readr`

Figure 3: https://https://www.tidyverse.org/

R packages for importing data: `tidyverse` and `readr`

Importing data in tidyverse with readr:

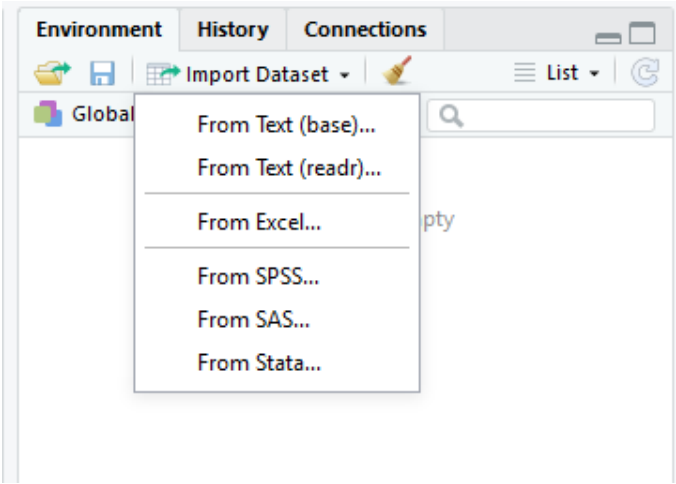
- `read_csv()` comma separated files
- `readr::read_tsv()` tab separated files
- `readr::read_delim()` files with other delimiters
- `readr::read_fwf()` fixed width files
- `readr::read_table()` fixed width files, columns sep. by white space

R packages for importing data: other packages

A selection of other R packages:

- Base R: `read.csv()`, `read.table()`, `read.delim()`
- `scan()`
- `haven` SPSS, SAS and Stata
- `readxl::read_excel()` .xls and .xlsx files
- `DBI` databases
- `xml2` XML

Importing data using R studio (Import Wizard)



Importing csv files with `read_csv`

Basic syntax: `read_csv("myfile.csv")`

```
read_csv("a,b,c
1,2,3
4,5,6")
```

```
# A tibble: 2 x 3
      a     b     c
  <dbl> <dbl> <dbl>
1     1     2     3
2     4     5     6
```

Usually the first row contains the column names

Customising `read_csv`: Skip rows

```
read_csv("The first line of metadata
  The second line of metadata
  x,y,z
  1,2,3", skip = 2)
```

```
# A tibble: 1 x 3
      x     y     z
  <dbl> <dbl> <dbl>
1     1     2     3
```

Customising `read_csv` : Skip comments

```
read_csv("# A comment I want to skip
  x,y,z
  1,2,3", comment = "#")
```

```
# A tibble: 1 x 3
      x     y     z
  <dbl> <dbl> <dbl>
1     1     2     3
```

Customising `read_csv` : No column names

```
read_csv("1,2,3\n4,5,6", col_names = FALSE)
```

```
# A tibble: 2 x 3
     X1    X2    X3
  <dbl> <dbl> <dbl>
1     1     2     3
2     4     5     6
```

Customising `read_csv` : NA

```
read_csv("a,b,c\n1,2,.", na = ".")
```

```
# A tibble: 1 x 3
      a     b c
  <dbl> <dbl> <lgl>
1     1     2 NA
```

Customising read_csv: Max number of rows

```
read_csv("a,b,c\n1,2,3\n1,2,3\n4,5,6", n_max=2)
```

```
# A tibble: 2 x 3
      a     b     c
  <dbl> <dbl> <dbl>
1     1     2     3
2     1     2     3
```

Importing data from Excel

```
library(readxl)
weather <- read_excel("auckland_weather.xlsx",
                      sheet = "weather", skip = 2,
                      n_max = 609)
```

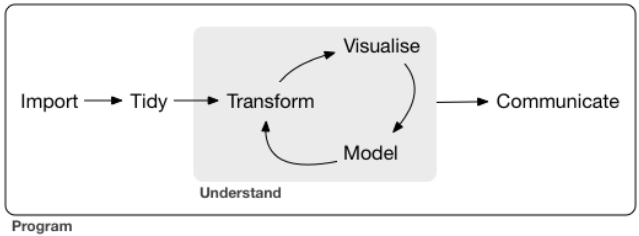
Hint: use the R studio import button to set the correct options.

Importing data using `scan()`

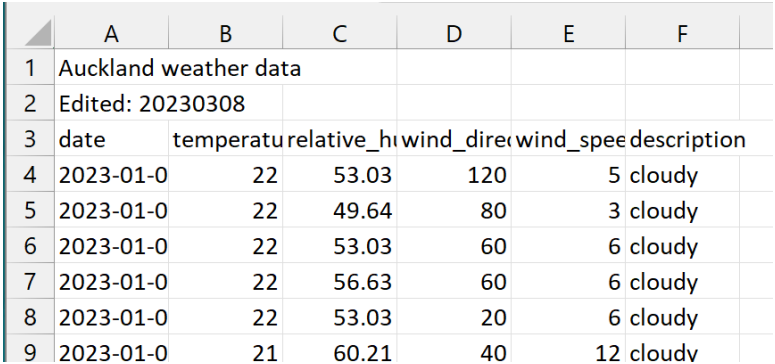
```
x <- scan(text=" 0.98041777  0.43947836 -1.55056151 -0.79728525
                -0.42800126  0.40199984 -2.47297031 -0.37864494
                -2.97528582 -0.65354009 -0.01825684  0.60541110
                 0.75126011 -0.60213081 -0.06856767  0.77608356
                 0.28803778 -0.60501184 -0.73270578  0.79513324
                 0.11889738 -1.61529589  2.06388035 -1.42861423
                -0.54059507 -2.03696668  1.07640270 -0.39991186
                 0.17169290 -0.89070463
           ")
head(x)
```

```
[1]  0.9804178  0.4394784 -1.5505615 -0.7972852
[5] -0.4280013  0.4019998
```

The Process of Analytics



Application 1: Auckland Weather Data



	A	B	C	D	E	F
1	Auckland weather data					
2	Edited: 20230308					
3	date	temperatu	relative_h	wind_dire	wind_spee	description
4	2023-01-0	22	53.03	120	5	cloudy
5	2023-01-0	22	49.64	80	3	cloudy
6	2023-01-0	22	53.03	60	6	cloudy
7	2023-01-0	22	56.63	60	6	cloudy
8	2023-01-0	22	53.03	20	6	cloudy
9	2023-01-0	21	60.21	40	12	cloudy

Figure 5: auckland_weather.csv

Reading a csv file

```
weather <- read_csv("auckland_weather.csv",
                    skip = 2, comment = "#")
```

Reading a csv file - output

```
# A tibble: 3,165 x 6
  date                temperature relati~1 wind_~2
  <dttm>                    <dbl>    <dbl>   <dbl>
1 2023-01-01 00:00:00          22     53.0     120
2 2023-01-01 00:30:00          22     49.6      80
3 2023-01-01 01:00:00          22     53.0      60
# ... with 3,162 more rows, 2 more variables:
#   wind_speed_knots <dbl>, description <chr>,
#   and abbreviated variable names
#   1: relative_humidity, 2: wind_direction_deg
```

Questions of interest

- What time period does the dataset cover?
- What variables are in the dataset?
- What temperatures were observed?
- What is the trend in wind speed, over time?

Dataset

```
# time period
min(weather$date)
```

```
[1] "2023-01-01 UTC"
```

```
max(weather$date)
```

```
[1] "2023-03-07 22:00:00 UTC"
```

```
# variables
colnames(weather)
```

```
[1] "date"                "temperature"
[3] "relative_humidity"   "wind_direction_deg"
[5] "wind_speed_knots"    "description"
```

Maximum temperature

```
max(weather$temperature)
```

```
[1] 26
```

```
summary(weather$temperature)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13.00   18.00   20.00   19.77   22.00   26.00 
```

Maximum daily “high” temperature

```
library(lubridate)
daily_weather <- weather %>%
  mutate(date_ymd = ymd(date(date))) %>%
  group_by(date_ymd) %>%
  summarise(hi = max(temperature))

daily_weather
```

```
# A tibble: 66 x 2
  date_ymd      hi
  <date>     <dbl>
1 2023-01-01    22
2 2023-01-02    23
3 2023-01-03    23
# ... with 63 more rows
```

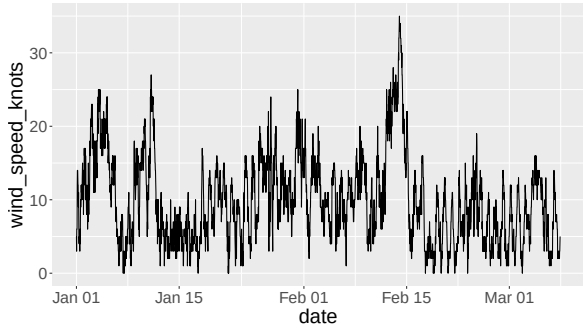
Maximum daily “high” temperature - summary

```
daily_weather$hi %>% summary()
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  19.00   22.25   24.00   23.35   24.00   26.00
```

Wind speed

```
weather %>% ggplot() +
  geom_line(mapping = aes(x = date, y = wind_speed_knots))
```



Application 2: Forbes Richest Atheletes 1990-2020

Source: https://www.kaggle.com/datasets/parulpandey/forbes-highest-paid-athletes-19902019

```
(richest <- read_csv("Forbes_richest_athletes.csv"))
```

```
# A tibble: 301 x 8
   S.NO Name   Natio~1 Curre~2 Previ~3 Sport  Year
  <dbl> <chr>  <chr>     <dbl> <chr>   <chr> <dbl>
1     1 Mike ~ USA           1 <NA>    boxi~  1990
2     2 Buste~ USA           2 <NA>    boxi~  1990
3     3 Sugar~ USA           3 <NA>    boxi~  1990
# ... with 298 more rows, 1 more variable:
#   `earnings ($ million)` <dbl>, and abbreviated
#   variable names 1: Nationality,
#   2: `Current Rank`, 3: `Previous Year Rank`
```

Questions

- What sports are included in the dataset?
- Has earnings increased over time?
- Who earnt the most?

Sports

```
richest %>% count(Sport)
```

```
# A tibble: 29 x 2
  Sport                              n
  <chr>                          <int>
1 American Football                 17
2 American Football / Baseball       1
3 Auto Racing                       10
# ... with 26 more rows
```

Sports - additional data cleaning

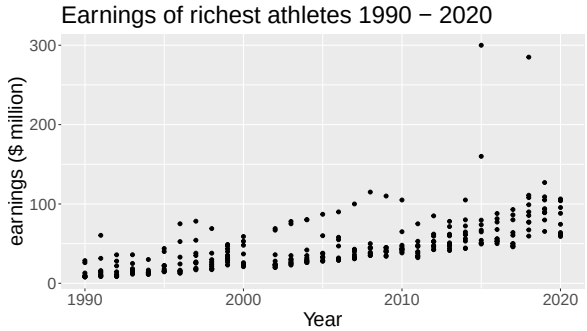
```
richest <- richest %>% mutate(Sport_lower = str_to_lower(Sport))
```

```
(richest %>%
  count(Sport_lower)  %>%
  arrange(-n))
```

```
# A tibble: 20 x 2
  Sport_lower     n
  <chr>       <int>
1 basketball     81
2 boxing         46
3 golf           44
# ... with 17 more rows
```

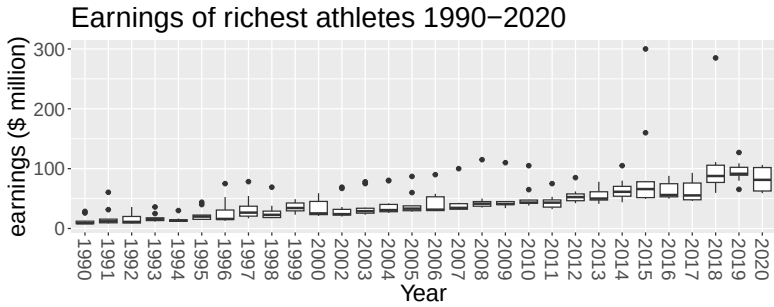
Earnings over time - scatterplot

```
richest %>% ggplot() +
  geom_point(mapping = aes(x = Year,
                           y = `earnings ($ million)`)) +
  labs(title = "Earnings of richest athletes 1990 - 2020" )
```



Earnings over time - boxplot

```
richest %>% ggplot() +
geom_boxplot(mapping = aes(x = as_factor(Year),
                           y = `earnings ($ million)`)) +
scale_x_discrete(breaks = 1990:2020) +
labs(x = "Year", title = "Earnings of richest athletes 1990-2020")+
theme(axis.text.x = element_text(angle = 270, vjust = 0.5, hjust=1))
```



Which athlete earnt the most?

```
richest %>%
  arrange(-`earnings ($ million)`) %>%
  slice_head(n = 1) %>%
  pull(Name)
```

```
[1] "Floyd Mayweather"
```

Which tennis player earnt the most?

```
(tennis <- richest %>%
                filter(Sport_lower == "tennis") %>%
                arrange(-`earnings ($ million)`) %>%
                slice_head(n=1) %>%
                pull(Name))
```

```
[1] "Roger Federer"
```

How much did the highest earning Tennis player earn 2010 - 2020?

```
richest %>%
  filter(Name == tennis,
         between(Year, 2010, 2020)) %>%
  summarise(TotalEarnings = sum(`earnings ($ million)`))
```

```
# A tibble: 1 x 1
  TotalEarnings
          <dbl>
1           746.
```

Workflow: R Projects

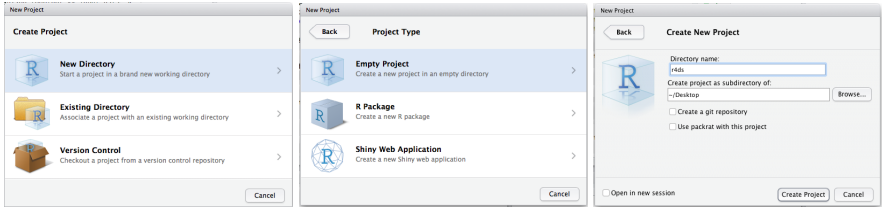
Projects

Key idea: R projects keep all parts of a data analysis project (code, data, results, plots etc) together.

Why?

- Facilitates sharing of project
- Keeps all parts of a project together
- Keeps a project separate from other projects

Create a new project



Source: https://r4ds.had.co.nz/workflow-projects.html

Example: Creating a project

- Create a project
- Check the working directory
- Enter the following code into an R script and save as `diamonds.R`

```
library(tidyverse)

ggplot(diamonds, aes(carat, price)) +
  geom_hex()
ggsave("diamonds.pdf")

write_csv(diamonds, "diamonds.csv")
```

- Run the script
- Inspect the directory where you created the project
- Quit RStudio
- Restart RStudio - notice what opens when you restart

Learning objectives

- Know how to import datasets from a range of file types into R using tidyverse
- Undertake exploratory data analysis using tidyverse
- Understand and apply the key principles of using R projects

References

Wickham, Hadley, and Garrett Grolemund. 2020. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*.