ETC5521: Exploratory Data Analysis

Initial data analysis

Lecturer: *Emi Tanaka*

ETC5521.Clayton-x@monash.edu

Week 3 - Session 1

Data Analysis

> **Data analysis** is a process of cleaning, transforming, inspecting and modelling data with the aim of extracting information.

- Data analysis includes:
  - exploratory data analysis,
  - confirmatory data analysis, and
  - *initial data analysis*.
- Confirmatory data analysis is focussed on statistical inference and includes processes such as testing hypothesis, model selection, or predictive modelling... but today's focus will be on ***initial data analysis***.

Initial Data Analysis (IDA)

- There are various definitions of IDA, much like there are numerous definitions for EDA.
- Some people would be practicing IDA without realising that it is IDA.
- Or other cases, a different name is used to describe the same process, such as Chatfield (1985) referring to IDA also as ***"initial examination of data"*** and Cox & Snell (1981) as ***"preliminary data anlysis"*** and Rao (1983) as ***"cross-examination of data"***.

So what is IDA?

Chatfield (1985) The Initial Examination of Data. *Journal of the Royal Statistical Society. Series A (General)* **148**
Cox & Snell (1981) Applied Statistics. *London: Chapman and Hall.*
Rao (1983) Optimum balance between statistical theory and application in teaching. *Proc. of the First Int Conference on Teaching Statistics* 34-49

What is IDA?

> The two **main objectives for IDA** are:
>
> 1. **data description**, and
> 2. **model formulation**.

- ***IDA differs from the main analysis*** (i.e. usually fitting the model, conducting significance tests, making inferences or predictions).
- ***IDA is often unreported*** in the data analysis reports or scientific papers due to it being "uninteresting" or "obvious".
- The role of ***the main analysis is to answer the intended question(s) that the data were collected for***.
- Sometimes IDA alone is sufficient.

❶ Data Description Part 1/2

- Data description should be one of the first steps in the data analysis to ***assess the structure and quality of the data***.
- We refer them to occasionally as ***data sniffing*** or ***data scrutinizing***.
- These include using common or domain knowledge to check if the recorded data have sensible values. E.g.
  - Are positive values, e.g. height and weight, recorded as positive values with a plausible range?
  - If the data are counts, do the recorded values contain non-integer values?
  - For compositional data, do the values add up to 100% (or 1)? If not is that a measurement error or due to rounding? Or is another variable missing?

❶ Data Description Part 2/2

- In addition, numerical or graphical summaries may reveal that there is unwanted structure in the data. E.g.,
  - Does the treatment group have different demographic characteristics to the control group?
  - Does the distribution of the data imply violations of assumptions for the main analysis?
- *Data sniffing* or *data scrutinizing* is a process that you get better at with practice and have familiarity with the domain area.
- Aside from checking the *data structure* or *data quality*, it's important to check how the data are understood by the computer, i.e. checking for *data type* is also important. E.g.,
  - Was the date read in as character?
  - Was a factor read in as numeric?

Next we'll see some *illustrative* examples and cases *based on real data* with some R codes

- Note: that there are a variety of ways to do IDA & EDA and you don't need to prescribe to what we show you.

Example 1 Checking the data type Part 1/2

`lecture3-example.xlsx`

	A	B	C	D
1	id	date	loc	temp
2	1	3/1/10	New York	42
3	2	3/2/10	New York	41.4
4	3	3/3/10	New York	38.5
5	4	3/4/10	New York	41.1
6	5	3/5/10	New York	39.8

```r
library(readxl)
library(here)
df <- read_excel(here("data/lecture3-example.xlsx"))
df

## # A tibble: 5 x 4
##      id date                loc       temp
##   <dbl> <dttm>              <chr>    <dbl>
## 1     1 2010-01-03 00:00:00 New York  42
## 2     2 2010-02-03 00:00:00 New York  41.4
## 3     3 2010-03-03 00:00:00 New York  38.5
## 4     4 2010-04-03 00:00:00 New York  41.1
## 5     5 2010-05-03 00:00:00 New York  39.8
```

Any issues here?

Example 1 Checking the data type Part 2/2

```r
library(lubridate)
df %>%
  mutate(id = as.factor(id),
         day = day(date),
         month = month(date),
         year = year(date)) %>%
  select(-date)
```

```
## # A tibble: 5 x 6
##   id    loc       temp   day month  year
##   <fct> <chr>    <dbl> <int> <dbl> <dbl>
## 1 1     New York  42       3     1  2010
## 2 2     New York  41.4     3     2  2010
## 3 3     New York  38.5     3     3  2010
## 4 4     New York  41.1     3     4  2010
## 5 5     New York  39.8     3     5  2010
```

- `id` is now a `factor` instead of `integer`
- `day`, `month` and `year` are now extracted from the `date`
- Is it okay now?
- In the United States, it's common to use the date format MM/DD/YYYY (gasps) while the rest of the world commonly use DD/MM/YYYY or YYYY/MM/DD.
- It's highly probable that the dates are 1st-5th March and not 3rd of Jan-May.
- You can validate this with other variables, say the temperature here.

Example 1 Checking the data type with R Part 1/3

- You can robustify your workflow by ensuring you have a check for the expected data type in your code.

```
xlsx_df <- read_excel(here("data/lecture3-example.xlsx"),
                      col_types = c("text", "date", "text", "numeric")) %>%
  mutate(id = as.factor(id),
         date = as.character(date),
         date = as.Date(date, format = "%Y-%d-%m"))
```

- `read_csv` has a broader support for `col_types`

```
csv_df <- read_csv(here("data/lecture3-example.csv"),
                   col_types = cols(
                        id = col_factor(),
                        date = col_date(format = "%m/%d/%y"),
                        loc = col_character(),
                        temp = col_double()))
```

- The checks (or coercions) ensure that even if the data are updated, you can have some confidence that any data type error will be picked up before further analysis.

Example 1 Checking the data type with R Part 2/3

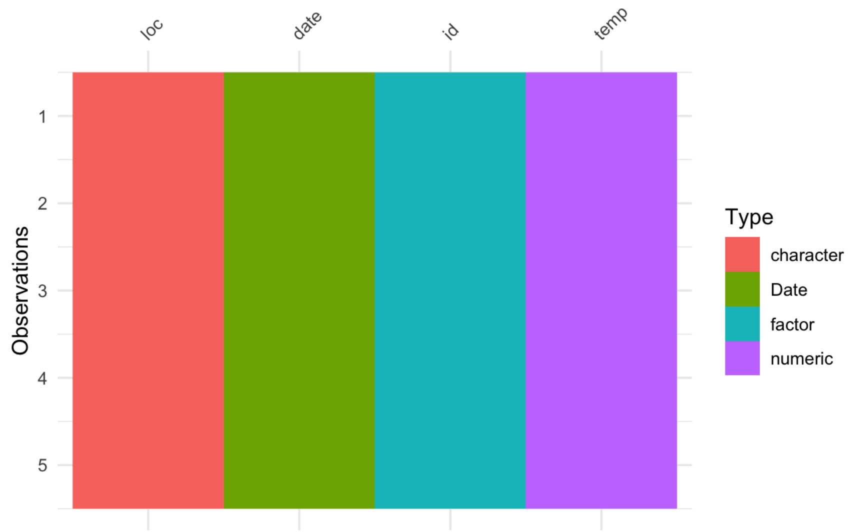
You can have a quick glimpse of the data type with:

```
dplyr::glimpse(xlsx_df)

## Rows: 5
## Columns: 4
## $ id   <fct> 1, 2, 3, 4, 5
## $ date <date> 2010-03-01, 2010-03-02, 2010-03-03, 2010-03-04, 2010-03-05
## $ loc  <chr> "New York", "New York", "New York", "New York", "New York"
## $ temp <dbl> 42.0, 41.4, 38.5, 41.1, 39.8

dplyr::glimpse(csv_df)

## Rows: 5
## Columns: 4
## $ id   <fct> 1, 2, 3, 4, 5
## $ date <date> 2010-03-01, 2010-03-02, 2010-03-03, 2010-03-04, 2010-03-05
## $ loc  <chr> "New York", "New York", "New York", "New York", "New York"
## $ temp <dbl> 42.0, 41.4, 38.5, 41.1, 39.8
```

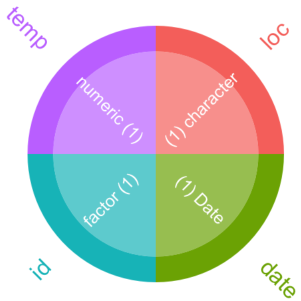
Example 1 Checking the data type with R Part 3/3

You can also visualise the data type with:

```r
library(visdat)
vis_dat(xlsx_df)
```




```r
library(inspectdf)
inspect_types(xlsx_df) %>%
  show_plot()
```



Example 2 Checking the data quality

```
df2 <- read_csv(here("data/lecture3-example2.csv"),
    col_types = cols(id = col_factor(),
                     date = col_date(format = "%m/%d/%y"),
                     loc = col_character(),
                     temp = col_double()))
df2
```

```
## # A tibble: 9 x 4
##   id    date       loc        temp
##   <fct> <date>     <chr>     <dbl>
## 1 1     2010-03-01 New York   42
## 2 2     2010-03-02 New York   41.4
## 3 3     2010-03-03 New York   38.5
## 4 4     2010-03-04 New York   41.1
## 5 5     2010-03-05 New York   39.8
## 6 6     2020-03-01 Melbourne  30.6
## 7 7     2020-03-02 Melbourne  17.9
## 8 8     2020-03-03 Melbourne  18.6
## 9 9     2020-03-04 <NA>       21.3
```

- Numerical or graphical summaries or even just eye-balling the data helps to uncover some data quality issues.
- Any issues here?

- There's a missing value in `loc`.
- Temperature is in Farenheit for New York but Celsius in Melbourne (you can validate this again using external sources).

Case study 1 Soybean study in Brazil Part 1/3

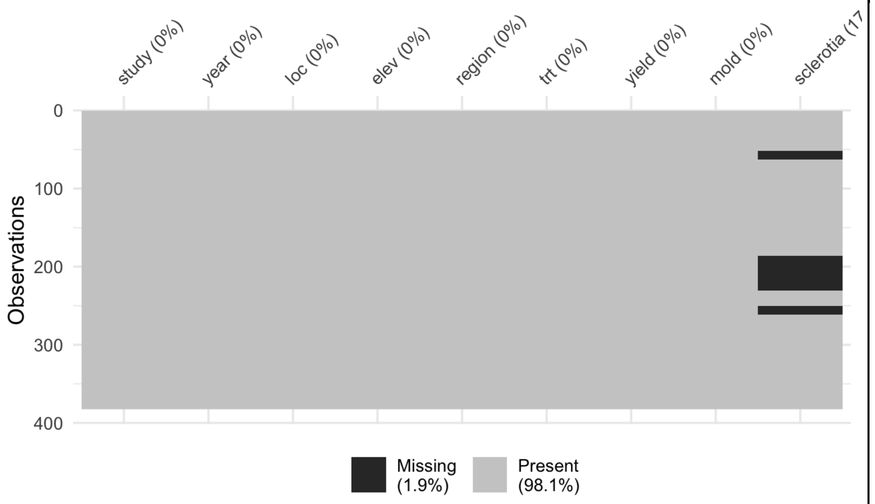
```
data("lehner.soybeanmold", package = "agridat")
skimr::skim(lehner.soybeanmold)

## ── Data Summary ────────────────────────
##                            Values
## Name                       lehner.soybeanmold
## Number of rows             382
## Number of columns          9
## _______________________
## Column type frequency:
##   factor                   4
##   numeric                  5
## ________________________
## Group variables            None
##
```

scroll

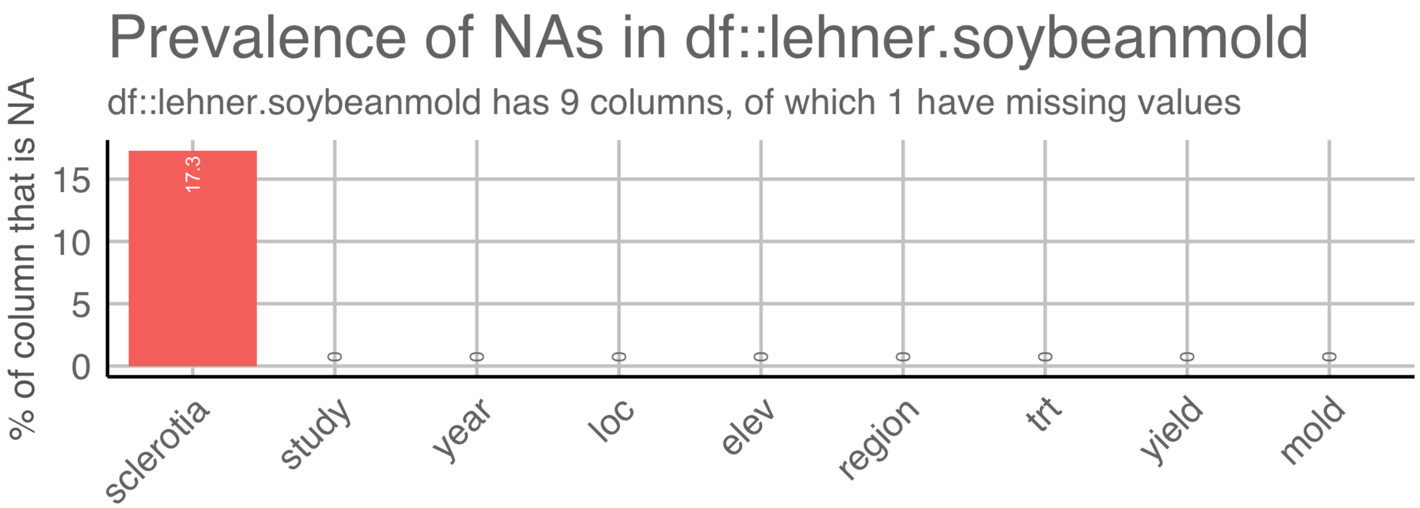
Lehner, M. S., Pethybridge, S. J., Meyer, M. C., & Del Ponte, E. M. (2016). Meta-analytic modelling of the incidence-yield and incidence-sclerotial production relationships in soybean white mould epidemics. *Plant Pathology*. doi:10.1111/ppa.12590

Case study 1 Soybean study in Brazil Part 2/3

```
vis_miss(lehner.soybeanmold)
```



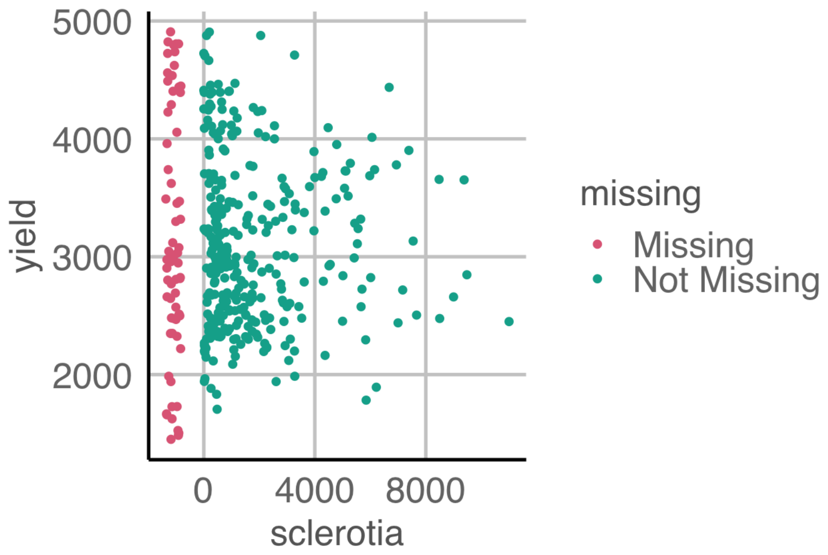

```
inspect_na(lehner.soybeanmold) %>%
  show_plot()
```



Case study 1 Soybean study in Brazil Part 3/3

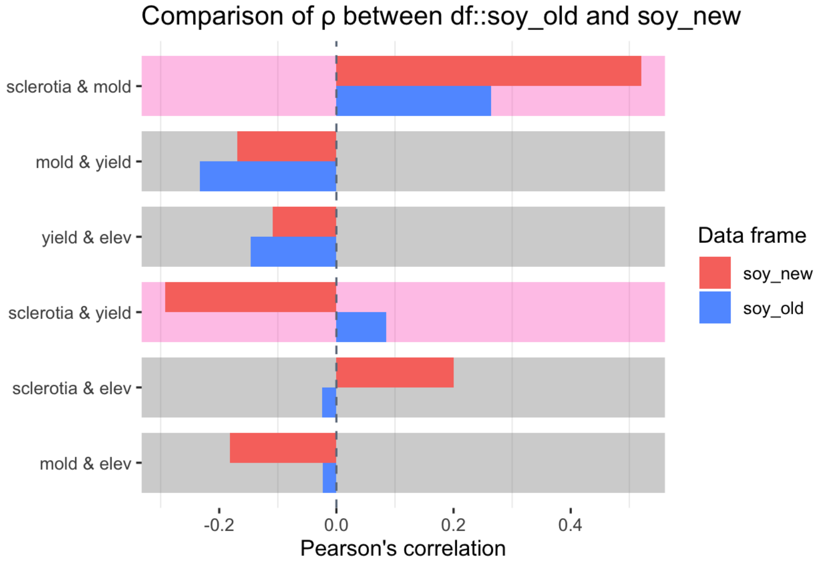
Checking if missing values have different yields:

```
library(naniar)
ggplot(lehner.soybeanmold,
       aes(sclerotia, yield)) +
  geom_miss_point() +
  scale_color_discrete_qualitative()
```



Compare the new with old data:

```
soy_old <- lehner.soybeanmold %>%
  filter(year %in% 2010:2011)
soy_new <- lehner.soybeanmold %>%
  filter(year == 2012)

inspect_cor(soy_old, soy_new) %>%
  show_plot()
```



Sanity check your data

Case study 2 Employment Data in Australia Part 1/3

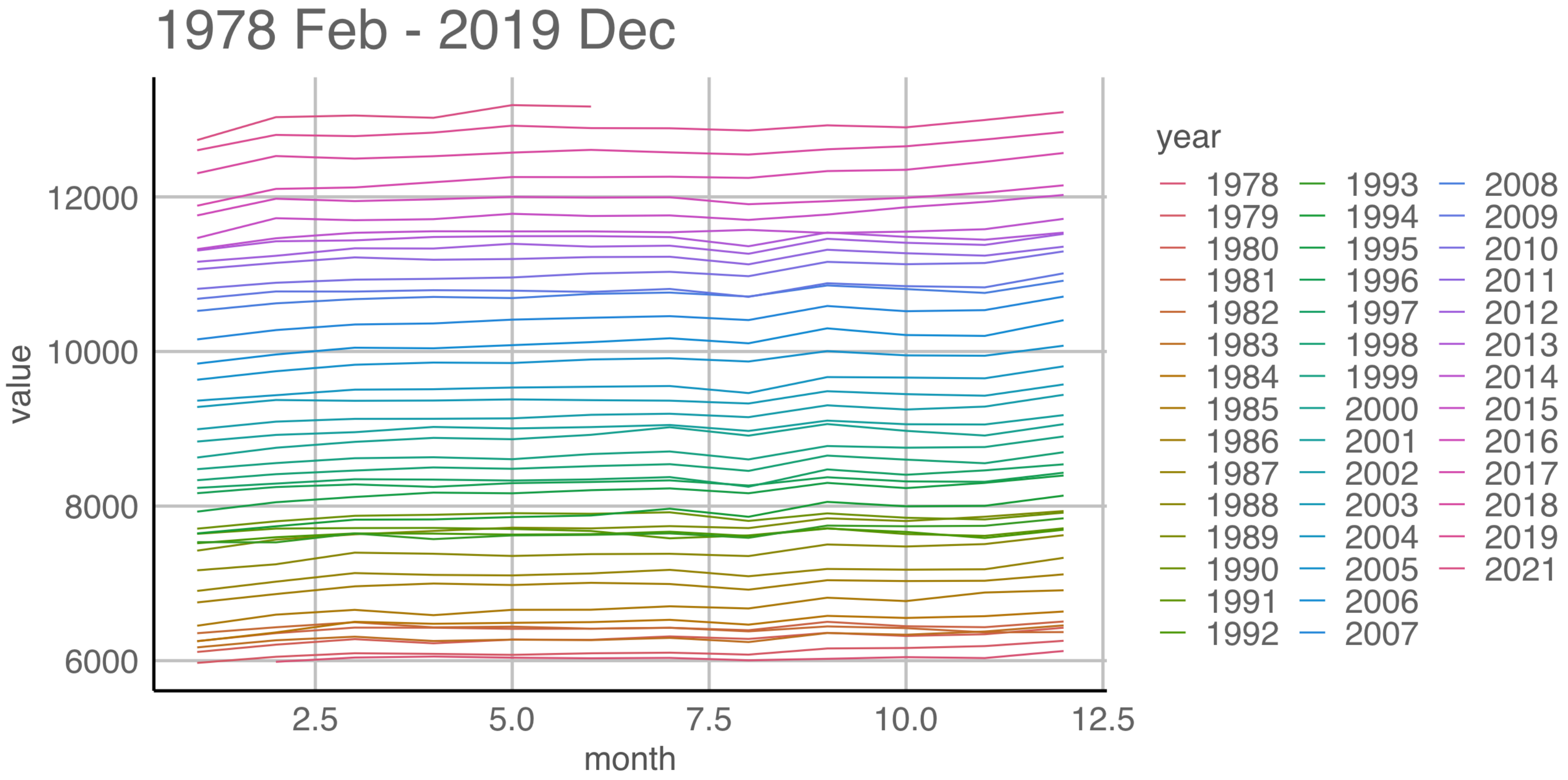
Below is the data from ABS that shows the total number of people employed in a given month from February 1976 to December 2019 using the original time series.

```
glimpse(employed)

## Rows: 509
## Columns: 4
## $ date  <date> 1978-02-01, 1978-03-01, 1978-04-01, 1978-05-01, 1978-06-01, 19
## $ month <dbl> 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9,
## $ year  <fct> 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1978, 197
## $ value <dbl> 5985.660, 6040.561, 6054.214, 6038.265, 6031.342, 6036.084, 600
```

Australian Bureau of Statistics, 2020, Labour force, Australia, Table 01. Labour force status by Sex, Australia - Trend, Seasonally adjusted and Original, viewed 2021-08-09, 🔗

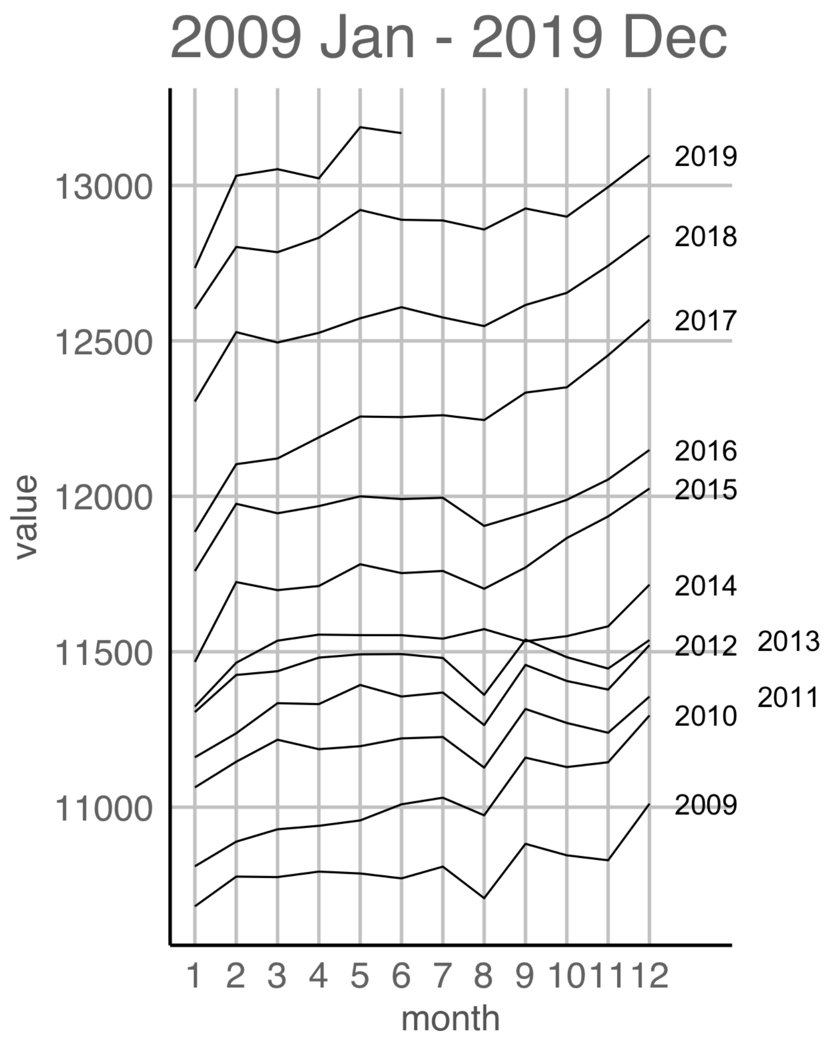
Case study 2 Employment Data in Australia Part 2/3

Do you notice anything?

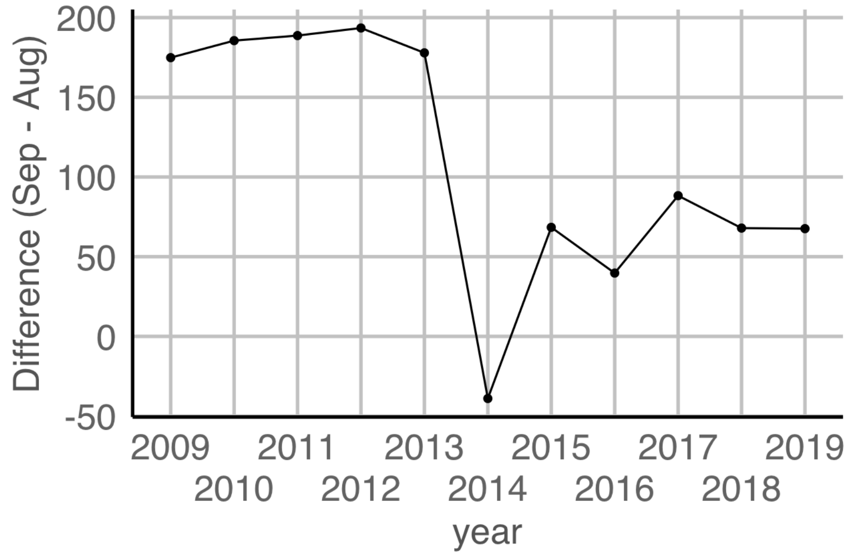


Why do you think the number of people employed is going up each year?

Case study 2 Employment Data in Australia Part 3/3



- There's a suspicious change in August numbers from 2014.



- A potential explanation for this is that there was a *change in the survey from 2014*.

Also see https://robjhyndman.com/hyndsight/abs-seasonal-adjustment-2/

Check if the *data collection* method has been consistent

Example 3 Experimental layout and data Part 1/2

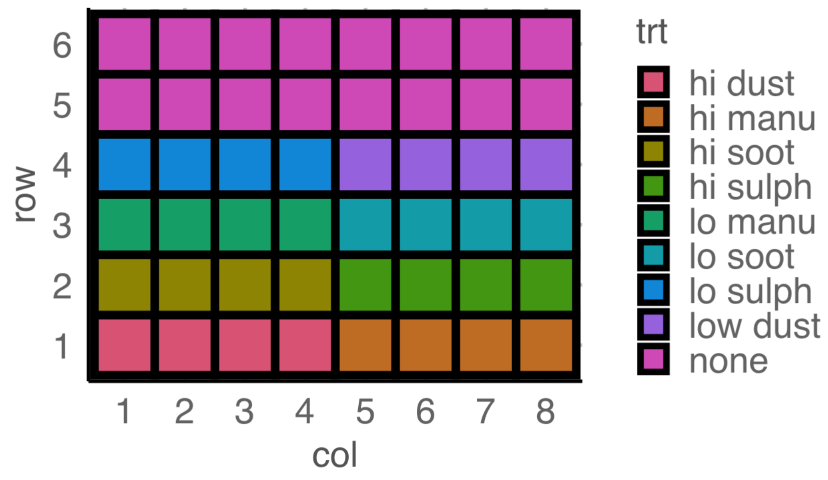
lecture3-example3.csv

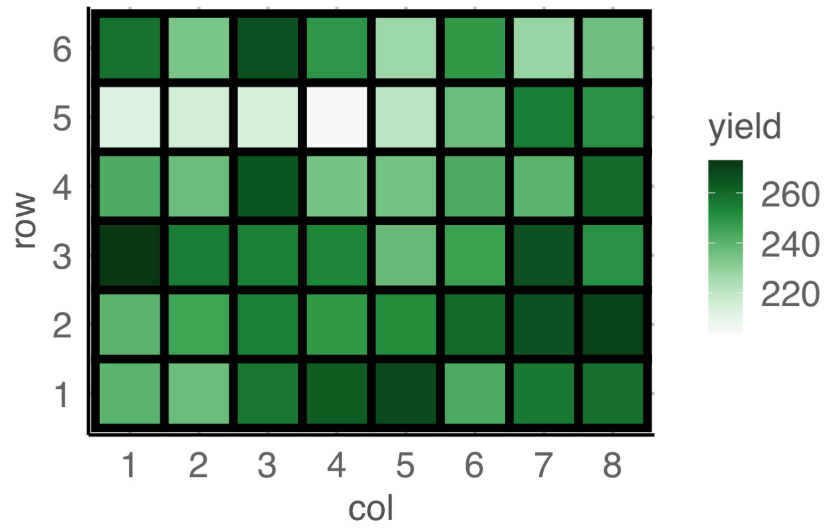
```
df3 <- read_csv(here::here("data/lecture3-example3.csv"),
                col_types = cols(
                  row = col_factor(),
                  col = col_factor(),
                  yield = col_double(),
                  trt = col_factor(),
                  block = col_factor()))
```

```
skimr::skim(df3)
```

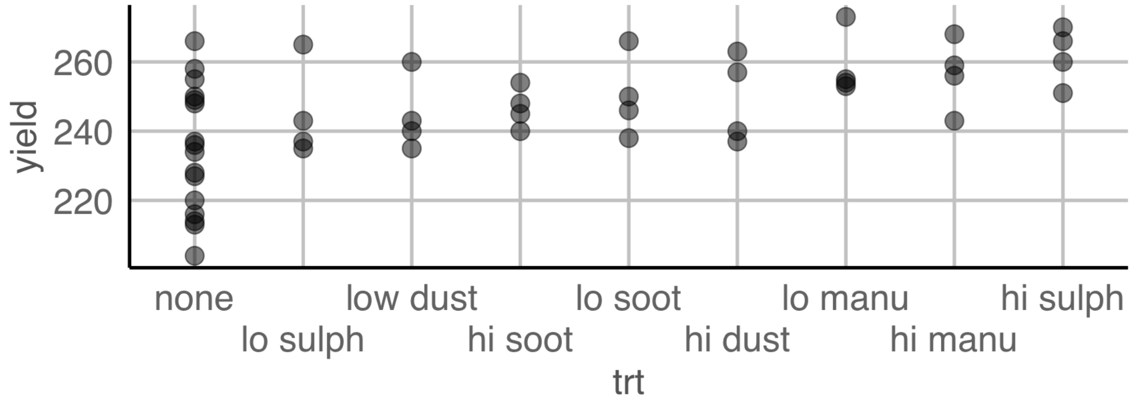
```
## ── Data Summary ────────────────────────
##                            Values
## Name                       df3
## Number of rows             48
## Number of columns          5
## _______________________
## Column type frequency:
```

Example 3 Experimental layout and data Part 2/2





- The experiment tests the effects of 9 fertilizer treatments on the yield of brussel sprouts on a field laid out in a rectangular array of 6 rows and 8 columns.



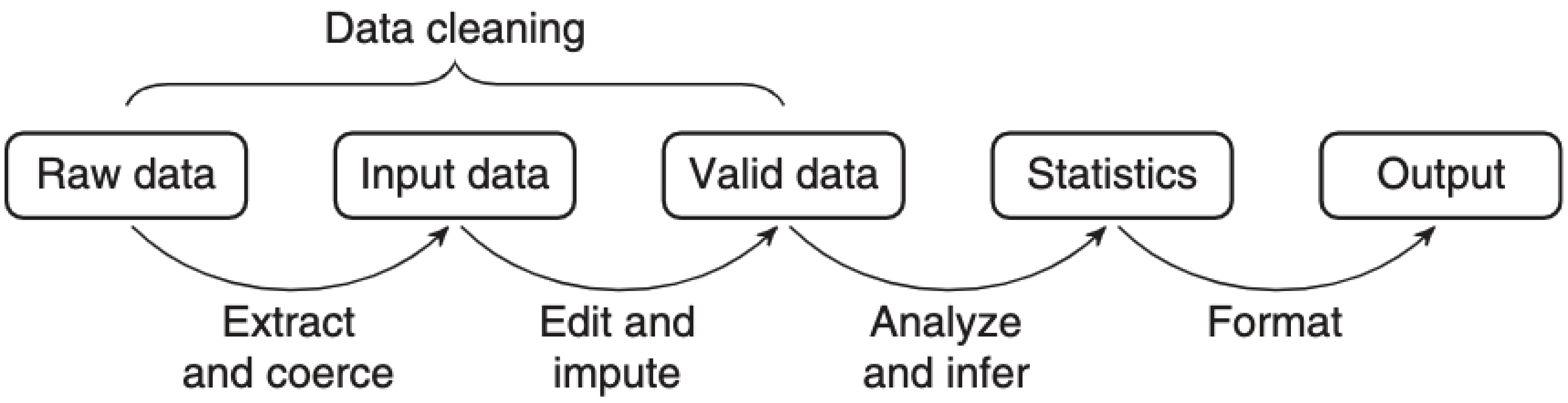
- High sulphur and high manure seems to best for the yield of brussel sprouts.
- Any issues here?

Check if experimental layout given in the data and the description match

In particular, have a check with a plot to see if treatments are *randomised*.

Statistical Value Chain

> … a ***statistical value chain*** *is constructed by defining a number of meaningful intermediate data products, for which a chosen set of quality attributes are well described …*
>
> *— van der Loo & de Jonge (2018)*



Schema from Mark van der Loo and Edwin de Jonge. 2018. Statistical Data Cleaning with Applications in R. John Wiley and Sons Ltd.

Case study 3 Dutch supermarket revenue and cost Part 1/3

- Data contains the revenue and cost (in Euros) for 60 supermarkets
- Data has been anonymised and distorted

```
## Rows: 60
## Columns: 11
## $ id          <fct> RET01, RET02, RET03, RET04, RET05, RET06, RET07, RET08, R
## $ size        <fct> sc0, sc3, sc3, sc3, sc3, sc0, sc3, sc1, sc3, sc2, sc2, sc
## $ incl.prob   <dbl> 0.02, 0.14, 0.14, 0.14, 0.14, 0.02, 0.14, 0.02, 0.14, 0.0
## $ staff       <int> 75, 9, NA, NA, NA, 1, 5, 3, 6, 5, 5, 5, 13, NA, 3, 52, 10
## $ turnover    <int> NA, 1607, 6886, 3861, NA, 25, NA, 404, 2596, NA, 645, 287
## $ other.rev   <int> NA, NA, -33, 13, 37, NA, NA, 13, NA, NA, NA, NA, 12, NA,
## $ total.rev   <int> 1130, 1607, 6919, 3874, 5602, 25, 1335, 417, 2596, NA, 64
## $ staff.costs <int> NA, 131, 324, 290, 314, NA, 135, NA, 147, NA, 130, 182, 3
## $ total.costs <int> 18915, 1544, 6493, 3600, 5530, 22, 136, 342, 2486, NA, 63
## $ profit      <int> 20045, 63, 426, 274, 72, 3, 1, 75, 110, NA, 9, 220, 34, 8
## $ vat         <int> NA, NA, NA, NA, NA, NA, 1346, NA, NA, NA, NA, NA, NA, 863
```

Case study 3 Dutch supermarket revenue and cost Part 2/3

- Checking for completeness of records

```
library(validate)
rules <- validator(
          is_complete(id),
          is_complete(id, turnover),
          is_complete(id, turnover, profit))
out <- confront(SBS2000, rules)
summary(out)

##   name items passes fails nNA error warning                        expression
## 1   V1    60     60     0   0 FALSE   FALSE                   is_complete(id)
## 2   V2    60     56     4   0 FALSE   FALSE         is_complete(id, turnover)
## 3   V3    60     52     8   0 FALSE   FALSE is_complete(id, turnover, profit)
```

Case study 3 Dutch supermarket revenue and cost Part 3/3

- Sanity check derived variables

```
library(validate)
rules <- validator(
    total.rev - profit == total.costs,
    turnover + other.rev == total.rev,
    profit <= 0.6 * total.rev
)
out <- confront(SBS2000, rules)
summary(out)

##   name items passes fails nNA error warning
## 1   V1    60     39    14   7 FALSE   FALSE abs(total.rev - profit - total.cc
## 2   V2    60     19     4  37 FALSE   FALSE abs(turnover + other.rev - total.
## 3   V3    60     49     6   5 FALSE   FALSE           (profit - 0.6 * total.r
```

Take away messages

- Sanity check your data:
  - by validating the variable types
  - with independent or external sources
  - by checking the data quality
- Check if the data collection method has been consistent
- Check if experimental layout given in the data and the description match
- Consider if or how data were derived for further sanity check of your data

Next we'll have a look at the

2 Model formulation

Lecturer: *Emi Tanaka*

ETC5521.Clayton-x@monash.edu

Week 3 - Session 1