ETC1010: Data Modelling and Computing

Week of Tidy Data

Professor Di Cook & Dr. Nicholas Tierney
EBS, Monash U.

2019-08-07

What is this song?

Recap

Hopefully a lot of you have done the Survey

- Traffic Light System: Green = "good!" ; Red = "Help!"
- R + Rstudio
- Tower of babel analogy for writing R code
- We are using , *not* for ETC1010?
- Functions are _
- columns in data frames are accessed with _ ?
- packages are installed with _ ?
- packages are loaded with _ ?
- Why do we care about Reproducibility?
- Output + input of rmarkdown
- I have an assignment group
- If I have an assignment group, have recorded my assignment group in the ED survey

Today: Outline

- An aside on learning
- Tidy Data
- Terminology of data
- Different examples of data
- Steps in making data tidy
- Lots of examples

A note on difficulty

- This is not a programming course - it is a course about **data, modelling, and computing**.

- At the moment, you might be sitting there, feeling a bit confused about where we are, what are are doing, what R is, and how it even works.
- That is OK!

A note on difficulty

- The theory of this class will only get you so far
- The real learning happens from doing the data analysis - the **pressure of a deadline can also help.**

An aside on learning

- I want to take the first 15 minutes of class to discuss some ideas on learning, and how this ties into the course.
- [new slides]

Tidy Data

> You're ready to sit down with a newly-obtained dataset, excited about how it will open a world of insight and understanding, and then find you can't use it. You'll first have to spend a significant amount of time to restructure the data to even begin to produce a set of basic descriptive statistics or link it to other data you've been using.
>
> --John Spencer

> "Tidy data" is a term meant to provide a framework for producing data that conform to standards that make data easier to use. Tidy data may still require some cleaning for analysis, but the job will be much easier.
>
> --John Spencer

Source: Measure Evaluation

Example: US graduate programs

- Data from a study on US grad programs.
- Originally came in an excel file containing rankings of many different programs.
- Contains information on four programs:
  1. Astronomy
  2. Economics
  3. Entomology, and
  4. Psychology

Example: US graduate programs

```
library(tidyverse)
grad <- read_csv("data/graduate-programs.csv")
grad %>% top_n(10)
# A tibble: 10 x 16
   subject Inst  AvNumPubs AvNumCits PctFacGrants PctCompletion
   <chr>   <chr>     <dbl>     <dbl>        <dbl>         <dbl>
 1 econom… BOST…      0.49      2.66         36.9          34.2
 2 econom… UNIV…      0.79      2.68         71.4          42.6
 3 econom… UNIV…      0.61      3.44         54.8          62.5
 4 econom… UNIV…      0.61      1.81         50.4          37.9
 5 psycho… NORT…      0.92      1.32         45.8          32.1
 6 psycho… UNIV…      1.42      4.45         72            48.9
 7 psycho… UNIV…      1.15      3.47         69.5          46.8
 8 psycho… UNIV…      1.05      1.72         65.6          34.2
 9 psycho… UNIV…      1.39      3.3          57.2          28.1
10 psycho… UNIV…      2.02      2.89         69.9          39
# … with 10 more variables: MedianTimetoDegree <dbl>,
#   PctMinorityFac <dbl>, PctFemaleFac <dbl>, PctFemaleStud <dbl>,
#   PctIntlStud <dbl>, AvNumPhDs <dbl>, AvGREs <dbl>, TotFac <dbl>,
#   PctAsstProf <dbl>, NumStud <dbl>
```

Example: US graduate programs

What's good about the format?

- **Rows** contain information about the institution
- **Columns** contain types of information, like average number of publications, average number of citations, % completion,

Example: US graduate programs

Easy to make summaries:

```
grad %>% count(subject)
# A tibble: 4 x 2
  subject        n
  <chr>      <int>
1 astronomy     32
2 economics    117
3 entomology    27
4 psychology   236
```

Example: US graduate programs

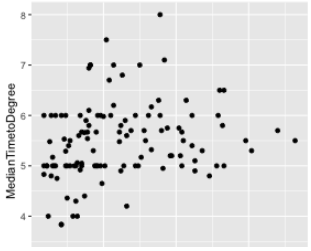
Easy to make summaries:

```
grad %>%
  filter(subject == "economics") %>%
  summarise(mean = mean(NumStud),
            s = sd(NumStud))
# A tibble: 1 x 2
   mean     s
  <dbl> <dbl>
1  60.7  39.4
```

Example: US graduate programs

Easy to make plot

```
grad %>%
  filter(subject == "economics") %>%
  ggplot(aes(x = NumStud,
             y = MedianTimetoDegree)) +
  geom_point() +
  theme(aspect.ratio = 1)
```



Your Turn: Go to the rstudio.cloud + open Lecture 2A.

- Notice the `data/` directory with many datasets!
- flag a tutor with the traffic light system if you need a hand.
- Open the `graduate-programs.Rmd` file
- Can you answer these questions?
  - "What is the average number of graduate students per economics program?:"
  - "What is the best description of the relationship between number of students and median time to degree?"

```
countdown(minutes = 3)
```

Terminology of data: Variable

- A quantity, quality, or property that you can measure.
- For the grad programs, these would be all the column headers.

```
grad
# A tibble: 412 x 16
   subject Inst  AvNumPubs AvNumCits PctI
   <chr>   <chr>     <dbl>     <dbl>
 1 econom… ARIZ…       0.9       1.57
 2 econom… AUBU…       0.79      0.64
 3 econom… BOST…       0.51      1.03
 4 econom… BOST…       0.49      2.66
 5 econom… BRAN…       0.3       3.03
 6 econom… BROW…       0.84      2.31
 7 econom… CALI…       0.99      2.31
 8 econom… CARN…       0.43      1.67
 9 econom… CITY…       0.35      1.06
10 econom… CLAR…       0.47      0.7
# … with 402 more rows, and 10 more vari
#   PctMinorityFac <dbl>, PctFemaleFac <
#   PctIntlStud <dbl>, AvNumPhDs <dbl>,
#   PctAsstProf <dbl>, NumStud <dbl>
```

Terminology of data: Observation

- A set of measurements made under similar conditions
- Contains several values, each associated with a different variable.
- For the grad programs, this is institution, and program, uniquley define the observation.

```
grad
# A tibble: 412 x 16
   subject Inst  AvNumPubs AvNumCits PctI
   <chr>   <chr>     <dbl>     <dbl>
 1 econom… ARIZ…       0.9       1.57
 2 econom… AUBU…       0.79      0.64
 3 econom… BOST…       0.51      1.03
 4 econom… BOST…       0.49      2.66
 5 econom… BRAN…       0.3       3.03
 6 econom… BROW…       0.84      2.31
 7 econom… CALI…       0.99      2.31
 8 econom… CARN…       0.43      1.67
 9 econom… CITY…       0.35      1.06
10 econom… CLAR…       0.47      0.7
# … with 402 more rows, and 10 more vari
#   PctMinorityFac <dbl>, PctFemaleFac <
#   PctIntlStud <dbl>, AvNumPhDs <dbl>,
#   PctAsstProf <dbl>, NumStud <dbl>
```

Terminology of data: Value

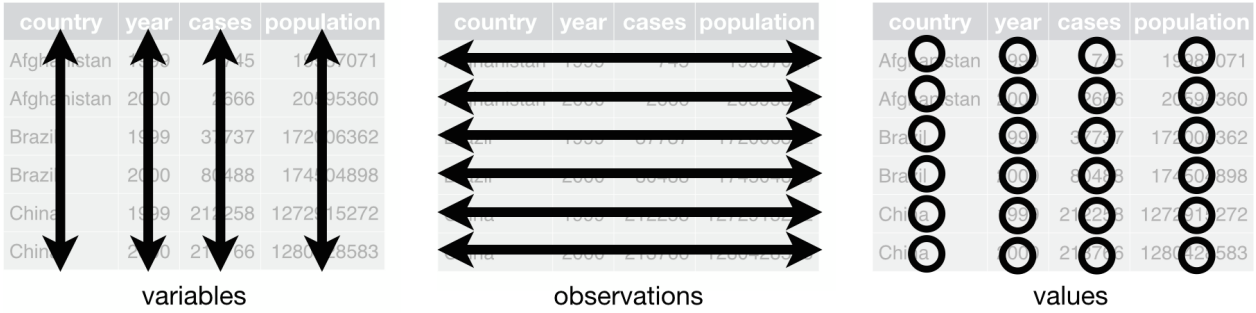
- Is the state of a variable when you measure it.
- The value of a variable typically changes from observation to observation.
- For the grad programs, this is the value in each cell

```
grad
# A tibble: 412 x 16
   subject Inst  AvNumPubs AvNumCits PctI
   <chr>   <chr>     <dbl>     <dbl>
 1 econom… ARIZ…       0.9       1.57
 2 econom… AUBU…       0.79      0.64
 3 econom… BOST…       0.51      1.03
 4 econom… BOST…       0.49      2.66
 5 econom… BRAN…       0.3       3.03
 6 econom… BROW…       0.84      2.31
 7 econom… CALI…       0.99      2.31
 8 econom… CARN…       0.43      1.67
 9 econom… CITY…       0.35      1.06
10 econom… CLAR…       0.47      0.7
# … with 402 more rows, and 10 more vari
#   PctMinorityFac <dbl>, PctFemaleFac <
#   PctIntlStud <dbl>, AvNumPhDs <dbl>,
#   PctAsstProf <dbl>, NumStud <dbl>
```

Tidy tabular form

Tabular data is a set of values, each associated with a variable and an observation. Tabular data is **tidy** iff (if and only if):

- Each variable in its own column,
- Each observation in its own row,
- Each value is placed in its own `cell`.


country
year
cases
population
variables
observations
values

The grad program

Is in **tidy** tabular form.

```
grad
# A tibble: 412 x 16
   subject Inst  AvNumPubs AvNumCits PctFacGrants PctCompletion
   <chr>   <chr>     <dbl>     <dbl>        <dbl>         <dbl>
 1 econom… ARIZ…      0.9       1.57         31.3          31.7
 2 econom… AUBU…      0.79      0.64         77.6          44.4
 3 econom… BOST…      0.51      1.03         43.5          46.8
 4 econom… BOST…      0.49      2.66         36.9          34.2
 5 econom… BRAN…      0.3       3.03         36.8          48.7
 6 econom… BROW…      0.84      2.31         27.1          54.6
 7 econom… CALI…      0.99      2.31         56.4          83.3
 8 econom… CARN…      0.43      1.67         35.2          45.6
 9 econom… CITY…      0.35      1.06         38.1          27.9
10 econom… CLAR…      0.47      0.7          24.7          37.7
# … with 402 more rows, and 10 more variables: MedianTimetoDegree <dbl>,
#   PctMinorityFac <dbl>, PctFemaleFac <dbl>, PctFemaleStud <dbl>,
#   PctIntlStud <dbl>, AvNumPhDs <dbl>, AvGREs <dbl>, TotFac <dbl>,
#   PctAcctProf <dbl>, NumStud <dbl>
```

Different examples of data

For each of these data examples, **let's try together to identify the variables and the observations** - some are HARD!

Your Turn: Genes experiment 🤔

```
genes <- read_csv("data/genes.csv")
genes
# A tibble: 3 x 12
  id    `WI-6.R1` `WI-6.R2` `WI-6.R4` `WM-6.R1` `WM-6.R2` `WI-12.R1`
  <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>       <dbl>
1 Gene…      2.18      2.20      4.20      2.63      5.06        4.54
2 Gene…      1.46     0.585      1.86     0.515      2.88        1.36
3 Gene…      2.03     0.870      3.28     0.533      4.63        2.18
# … with 5 more variables: `WI-12.R2` <dbl>, `WI-12.R4` <dbl>,
#   `WM-12.R1` <dbl>, `WM-12.R2` <dbl>, `WM-12.R4` <dbl>
```

Melbourne weather 😦

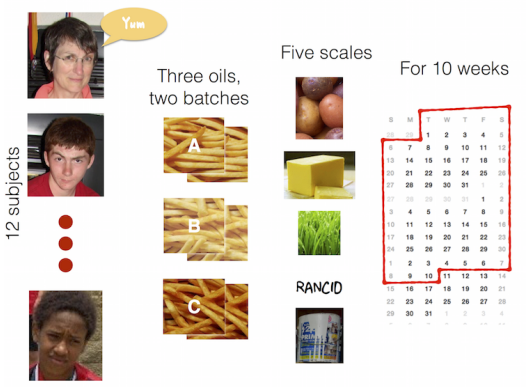
```
melbtemp
# A tibble: 1,593 x 12
   X1         X2 X3    X4       X5    X9   X13   X17   X21   X25   X29   X33
   <chr>   <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 ASN00…  1970 07    TMAX    141   124   113   123   148   149   139   153
 2 ASN00…  1970 07    TMIN     80    63    36    57    69    47    84    78
 3 ASN00…  1970 07    PRCP      3    30     0     0    36     3     0     0
 4 ASN00…  1970 08    TMAX    145   128   150   122   109   112   116   142
 5 ASN00…  1970 08    TMIN     50    61    75    67    41    51    48    -7
 6 ASN00…  1970 08    PRCP      0    66     0    53    13     3     8     0
 7 ASN00…  1970 09    TMAX    168   168   162   162   162   150   184   179
 8 ASN00…  1970 09    TMIN     19    29    62    81    81    55    73    97
 9 ASN00…  1970 09    PRCP      0     0     0     0     3     5     0    38
10 ASN00…  1970 10    TMAX    189   194   204   267   256   228   237   144
# … with 1,583 more rows
```

Tuberculosis notifications data taken from WHO 😫

```
tb
# A tibble: 3,202 x 22
   country  year new_sp_m04 new_sp_m514 new_sp_m014 new_sp_m1524
   <chr>   <dbl>      <dbl>       <dbl>       <dbl>        <dbl>
 1 Afghan…  1997         NA          NA           0           10
 2 Afghan…  1998         NA          NA          30          129
 3 Afghan…  1999         NA          NA           8           55
 4 Afghan…  2000         NA          NA          52          228
 5 Afghan…  2001         NA          NA         129          379
 6 Afghan…  2002         NA          NA          90          476
 7 Afghan…  2003         NA          NA         127          511
 8 Afghan…  2004         NA          NA         139          537
 9 Afghan…  2005         NA          NA         151          606
10 Afghan…  2006         NA          NA         193          837
# … with 3,192 more rows, and 16 more variables: new_sp_m2534 <dbl>,
#   new_sp_m3544 <dbl>, new_sp_m4554 <dbl>, new_sp_m5564 <dbl>,
#   new_sp_m65 <dbl>, new_sp_mu <dbl>, new_sp_f04 <dbl>,
#   new_sp_f514 <dbl>, new_sp_f014 <dbl>, new_sp_f1524 <dbl>,
#   new_sp_f2534 <dbl>, new_sp_f3544 <dbl>, new_sp_f4554 <dbl>,
#   new_sp_f5564 <dbl>, new_sp_f65 <dbl>, new_sp_fu <dbl>
```

French fries

10 week sensory experiment, 12 individuals assessed taste of french fries on several scales (how potato-y, buttery, grassy, rancid, paint-y do they taste?), fried in one of 3 different oils, replicated twice.



French fries: Variables? Observations?

```
french_fries <- read_csv("data/french_fries.csv")
french_fries
# A tibble: 696 x 9
    time treatment subject   rep potato buttery grassy rancid painty
   <dbl>     <dbl>   <dbl> <dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
 1     1         1       3     1    2.9     0      0      0      5.5
 2     1         1       3     2   14       0      0      1.1    0
 3     1         1      10     1   11       6.4    0      0      0
 4     1         1      10     2    9.9     5.9    2.9    2.2    0
 5     1         1      15     1    1.2     0.1    0      1.1    5.1
 6     1         1      15     2    8.8     3      3.6    1.5    2.3
 7     1         1      16     1    9       2.6    0.4    0.1    0.2
 8     1         1      16     2    8.2     4.4    0.3    1.4    4
 9     1         1      19     1    7       3.2    0      4.9    3.2
10     1         1      19     2   13       0      3.1    4.3   10.3
# … with 686 more rows
```

Rude Recliners data

This data is collated from this story: 41% Of Fliers Think You're Rude If You Recline Your Seat

What are the variables?

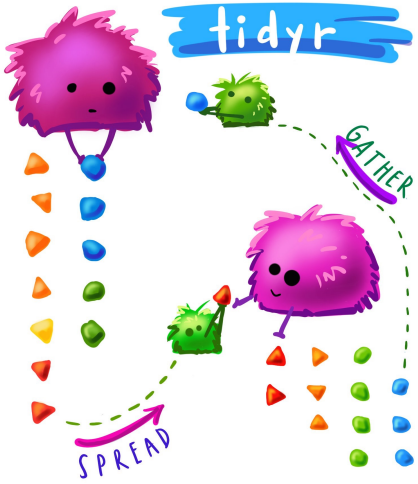
```
recliners <- read_csv("data/recliners.csv")
recliners
# A tibble: 3 x 6
  V1    `V2:Always` `V2:Usually` `V2:About half … `V2:Once in a w…
  <chr>       <dbl>        <dbl>            <dbl>            <dbl>
1 No, …         124          145               82              116
2 Yes,…           9           27               35              129
3 Yes,…           3            3               NA               11
# … with 1 more variable: `V2:Never` <dbl>
```

Messy vs tidy

Messy data is messy in its own way. You can make unique solutions, but then another data set comes along, and you have to again make a unique solution.

Tidy data can be though of as legos. Once you have this form, you can put it together in so many different ways, to make different analyses.

Data Tidying verbs



Source: A drawing made by Alison Horst @allison_horst

Data Tidying Verbs

- `gather`: Specify the **keys** (identifiers) and the **values** (measures) to make long form data. You can also think of this as longer form data.
- `spread`: Variables in columns
- `separate`: Split one column into many

one more time: gather

```
gather(<DATA>,
       <KEY>,
       <VALUE>,
       <COLUMNS TO SELECT>)
```

- **Key** is the name of the variable whose values for the column names.
- **Value** is the name of the variable whose values are spread over the cells.
- **Columns** to select are those that represent values, not variables.

gather: example

```
table4a
# A tibble: 3 x 3
  country     `1999` `2000`
  <chr>        <int>  <int>
1 Afghanistan    745   2666
2 Brazil       37737  80488
3 China       212258 213766
```

```
table4a %>%
  gather(`1999`,
         `2000`,
         key = "year",
         value = "cases")
# A tibble: 6 x 3
  country     year   cases
  <chr>       <chr>  <int>
1 Afghanistan 1999     745
2 Brazil      1999   37737
3 China       1999  212258
4 Afghanistan 2000    2666
5 Brazil      2000   80488
6 China       2000  213766
```

Tidying genes data

Tell me what to put in the following?

- **Key** is the name of the variable whose values for the column names.
- **Value** is the name of the variable whose values are spread over the cells.
- **Columns** to select are those that represent values, not variables.

```
genes
# A tibble: 3 x 12
  id    `WI-6.R1` `WI-6.R2` `WI-6.R4` `WM-6.R1` `WM-6.R2` `WI-12.R1`
  <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
1 Gene…      2.18     2.20       4.20     2.63       5.06       4.54
2 Gene…      1.46     0.585      1.86     0.515      2.88       1.36
3 Gene…      2.03     0.870      3.28     0.533      4.63       2.18
# … with 5 more variables: `WI-12.R2` <dbl>, `WI-12.R4` <dbl>,
#   `WM-12.R1` <dbl>, `WM-12.R2` <dbl>, `WM-12.R4` <dbl>
```

Tidy genes data

```
genes_long <- genes %>%
  gather(key = variable,
         value = expr,
         -id)

genes_long
# A tibble: 33 x 3
   id     variable  expr
   <chr>  <chr>    <dbl>
 1 Gene 1 WI-6.R1  2.18
 2 Gene 2 WI-6.R1  1.46
 3 Gene 3 WI-6.R1  2.03
 4 Gene 1 WI-6.R2  2.20
 5 Gene 2 WI-6.R2  0.585
 6 Gene 3 WI-6.R2  0.870
 7 Gene 1 WI-6.R4  4.20
 8 Gene 2 WI-6.R4  1.86
 9 Gene 3 WI-6.R4  3.28
10 Gene 1 WM-6.R1  2.63
# … with 23 more rows
```

Separate columns

```
genes_long %>%
  separate(col = variable,
           into = c("trt", "leftover"), "-")
# A tibble: 33 x 4
   id     trt   leftover  expr
   <chr>  <chr> <chr>    <dbl>
 1 Gene 1 WI    6.R1     2.18
 2 Gene 2 WI    6.R1     1.46
 3 Gene 3 WI    6.R1     2.03
 4 Gene 1 WI    6.R2     2.20
 5 Gene 2 WI    6.R2     0.585
 6 Gene 3 WI    6.R2     0.870
 7 Gene 1 WI    6.R4     4.20
 8 Gene 2 WI    6.R4     1.86
 9 Gene 3 WI    6.R4     3.28
10 Gene 1 WM    6.R1     2.63
# … with 23 more rows
```

Separate columns

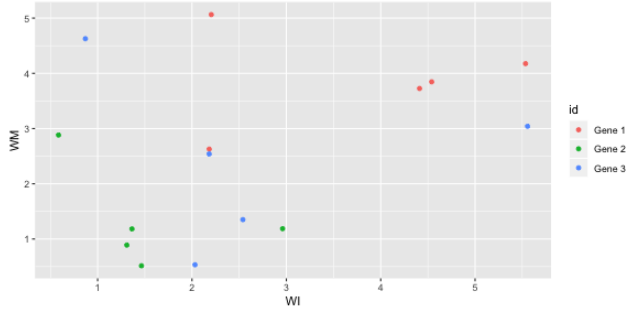
```
genes_long_tidy <- genes_long %>%
  separate(variable, c("trt", "leftover"), "-") %>%
  separate(leftover, c("time", "rep"), "\\.")

genes_long_tidy
# A tibble: 33 x 5
   id     trt   time  rep    expr
   <chr>  <chr> <chr> <chr> <dbl>
 1 Gene 1 WI    6     R1    2.18
 2 Gene 2 WI    6     R1    1.46
 3 Gene 3 WI    6     R1    2.03
 4 Gene 1 WI    6     R2    2.20
 5 Gene 2 WI    6     R2    0.585
 6 Gene 3 WI    6     R2    0.870
 7 Gene 1 WI    6     R4    4.20
 8 Gene 2 WI    6     R4    1.86
 9 Gene 3 WI    6     R4    3.28
10 Gene 1 WM    6     R1    2.63
# … with 23 more rows
```

Now spread to examine different aspects

Examine treatments against each other

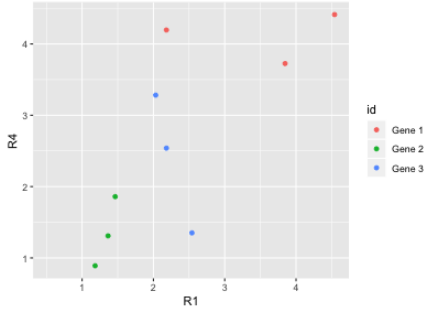
```
genes_long_tidy %>%
  spread(trt, expr) %>%
  ggplot(aes(x=WI, y=WM, colour=id)) + geom_point()
```



Generally, some negative association within each gene, WM is low if WI is high.

Examine replicates against each other

```
genes_long_tidy %>%
  spread(rep, expr) %>%
  ggplot(aes(x=R1, y=R4, colour=id)) +
  geom_point() + coord_equal()
```



Very roughly, replicate 4 is like replicate 1, eg if one is low, the other is low. That's a good thing, that the replicates are fairly similar.

Your turn: Demonstrate with koala bilby data (write this as live code)

Here is a little data set to practice gather, spread and separate on.

```
kb <- read_csv("data/koala_bilby.csv")
kb
# A tibble: 5 x 5
  ID    koala_NSW koala_VIC bilby_NSW bi
  <chr>     <dbl>     <dbl>     <dbl>
1 grey         23        43        11
2 cream        56        89        22
3 white        35        72        13
4 black        28        44        19
5 taupe        25        37        21
```

- Read over `koala-bilby.Rmd`
- Gather the data into long form, naming the two new variables, `label` and `count`
- Separate the labels into two new variables, `animal`, `state`
- Spread the long form data into wide form, where the columns are the states.
- Spread the long form data into wide form, where the columns are the animals.

Exercise 1: Rude Recliners

- Open `rude-recliners.Rmd`
- This contains data from the article 41% Of Fliers Think You're Rude If You Recline Your Seat.
- V1 is the response to question: "Is it rude to recline your seat on a plane?"
- V2 is the response to question: "Do you ever recline your seat when you fly?".

```
recliners <- read_csv("data/recliners.csv")
recliners
# A tibble: 3 x 6
  V1    `V2:Always` `V2:Usually` `V2:About half … `V2:Once in a w…
  <chr>       <dbl>        <dbl>            <dbl>            <dbl>
```

Exercise 1: Rude Recliners (15 minutes)

Answer the following questions in the rmarkdown document.

Ex1 A) What are the variables and observations in this data?

Ex 1B) Put the data in tidy long form (using the names `V2` as the key variable, and `count` as the value).

Ex 1C) Use the `rename` function to make the variable names a little shorter.

Exercise 1: Answers

Exercise 2: Tuberculosis Incidence data (15 minutes)

Open: `tb-incidence.Rmd`

Tidy the TB incidence data, using rthe Rmd to orompt questions.

Exercise 3: Currency rates (15 minutes)

- open `currency-rates.Rmd`
- read in `rates.csv`
- Answer the following questions:

1. What are the variables and observations?
2. Gather the five currencies, AUD, GBP, JPY, CNY, CAD, make it into tidy long form.
3. Make line plots of the currencies, describe the similarities and differences between the currencies.

Exercise 4: Australian Airport Passengers (optional extension challenge!)

- Open `oz-airport.Rmd`
- This contains data from the web site Department of Infrastructure, Regional Development and Cities, containing data on Airport Traffic Data 1985–86 to 2017–18.
- Read the dataset, into R, naming it `passengers`
- Tidy the data, to produce a data set with these columns
  - airport: all of the airports.
  - year
  - type_of_flight: DOMESTIC, INTERNATIONAL
  - bound: IN or OUT

Lab quiz

Time to take the lab quiz.

Share and share alike

Learning is where you:

1. Receive information accurately
2. Remember the information (long term memory)
3. In such a way that you can reapply the information when appropriate

Your Turn:

Go to the data source at this link: bit.ly/dmac-noaa-data

- "Which is the best description of the temperature units?"
- "What is the best description of the precipitation units"
- "What does -9999 mean?"