

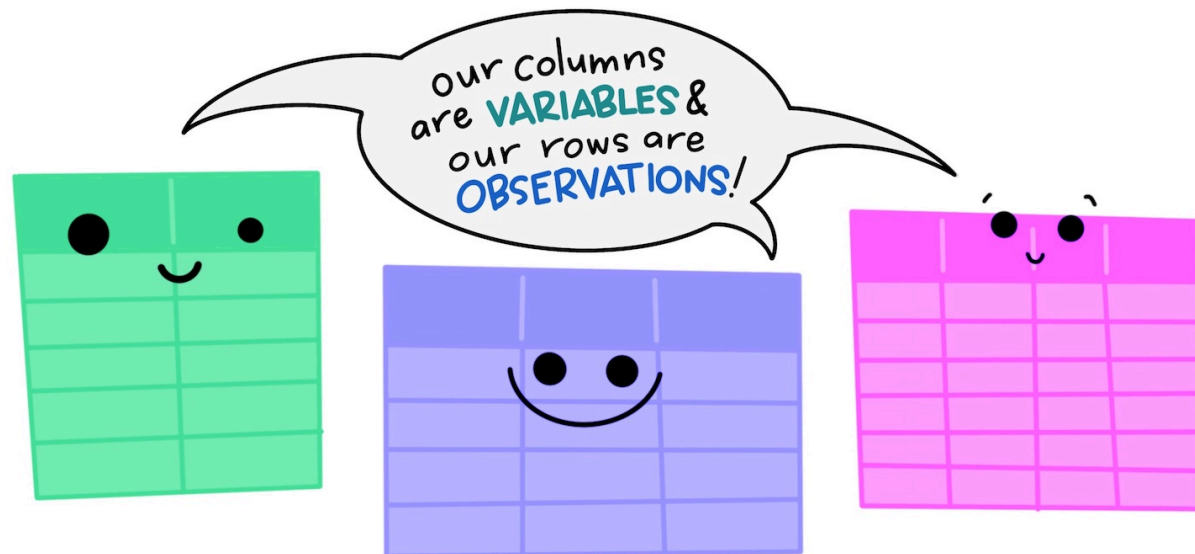
Data cleaning

Melbourne Statistical Consulting Platform

University of Melbourne

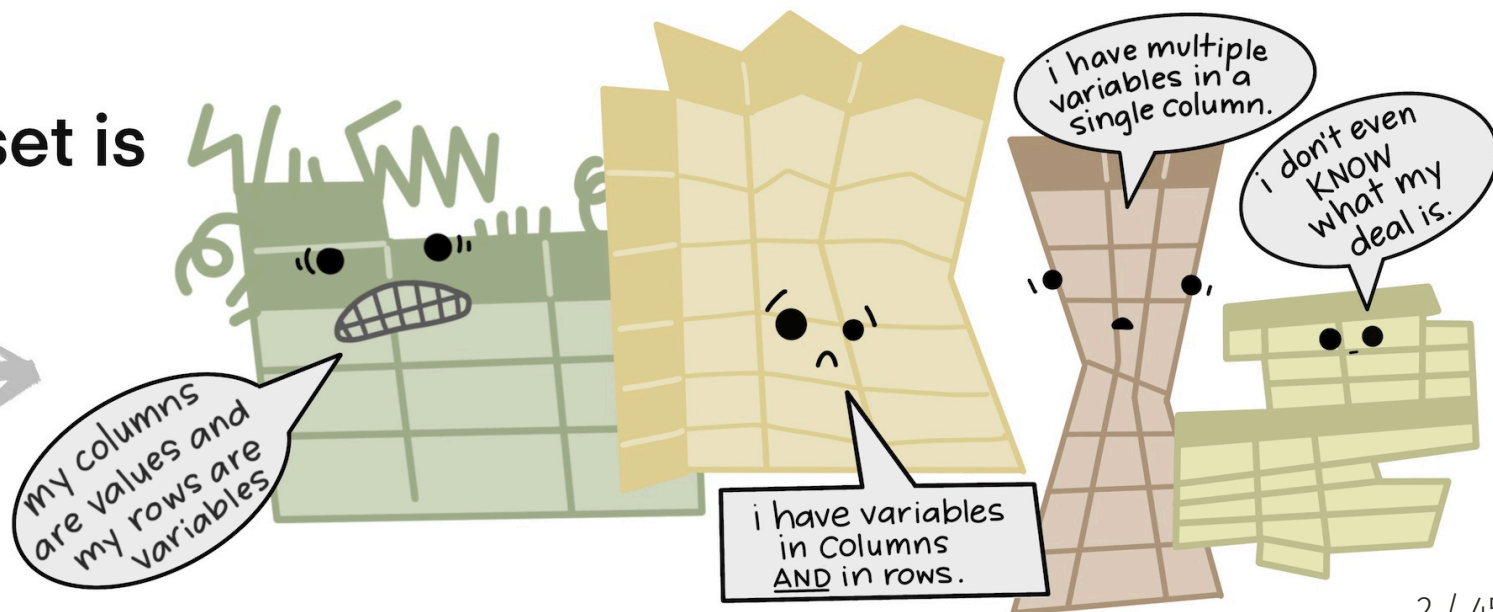
April 2024

The standard structure of tidy data means that
"tidy datasets are all alike..."



"...but every messy dataset is
messy in its own way."

—HADLEY WICKHAM



Motivation

Garbage in, garbage out



- the majority of the science of data is (unfortunately) data cleaning
- no easy fixes and every data set is different
- R has smart ways to explore the mess and clean it, often in reproducible ways
- We will focus on some of the most common

Issues with individual variables

- Variable name issues
- Obvious errors
- Missing data
- Different date and time formats

Variable name issues

Some biosecurity screening data from incoming flights to Australia:

```
screening_unclean <-  
  read_csv("../5 data/airport_screening_unclean.csv")
```

```
glimpse(screening_unclean)
```

Rows: 200

Columns: 24

\$ BRM

\$ sex

\$ age

\$ date

\$ P_C

\$ `period of stay`

\$ port...7

\$ airline

\$ port...9

\$ passport_country

\$ number_trips

\$ eggs

\$ other_flower_seed

\$ unidentified_seed

\$ beef

\$ other_fruit_vegetable_seed

\$ pork

\$ other_seeds

\$ hides_skins

\$ sesame seed

<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0

<chr> "F", "F", "M", "M", "M",

<dbl> 42, 1, 64, 28, 40, 35, 45

<chr> "16/12/2014", "16/10/2013

<chr> "P", "P", "P", "P", "P",

<dbl> 2884, 8, 5, 288, 63, 89,

<chr> "OOL", "BNE", "BNE", "MEI

<chr> "JQ", "FJ", "CZ", "SQ", "

<chr> "CHC", "NAN", "CAN", "SIM

<chr> "A", "B", "B", "D", "B",

<dbl> 5, 2, 32, 21, 16, 7, 6, 1

<chr> "0", "0", "0", "0", "0",

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0

<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0

Duplicate variable names

```
screening_unclean %>%  
  tabyl(port...7) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

port...7	n	percent	valid_percent
ADL	8	4.0%	4.1%
BNE	26	13.0%	13.2%
DRW	5	2.5%	2.5%
MEL	60	30.0%	30.5%
OOL	5	2.5%	2.5%
PER	8	4.0%	4.1%
SYD	85	42.5%	43.1%
NA	3	1.5%	-

```
screening_unclean %>%  
  tabyl(port...9) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

port...9	n	percent
AKL	35	17.5%
ATH	1	0.5%
AUH	1	0.5%
BGW	1	0.5%
BKI	1	0.5%
BKK	9	4.5%
BLR	1	0.5%
BOM	1	0.5%
BWN	2	1.0%
CAN	4	2.0%
CDG	1	0.5%

Spaces in variable names

```
screening_unclean %>%  
  summarise(Mean = mean(period of stay),  
            SD = sd(period of stay),  
            Min = min(period of stay),  
            Max = max(period of stay))%>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Error: <text>:2:32: unexpected symbol

1: screening_unclean %>%

2: summarise(Mean = mean(period of
 ^

```
screening_unclean %>%  
  summarise(Mean = mean(`period of stay`),  
            SD = sd(`period of stay`),  
            Min = min(`period of stay`),  
            Max = max(`period of stay`))%>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Mean	SD	Min	Max
159.0	492.7	1.0	3,310.0

Unclear variable names

```
screening_unclean %>%  
  tabyl(P_C) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

P_C	n	percent
C	8	4.0%
P	192	96.0%

Changing variable names

```
screening_unclean %>%  
  rename(arrival_port = port...7,  
         check_in_port = port...9,  
         period_of_stay = `period of stay`,  
         passenger_or_crew = P_C)
```

```
glimpse(screening_unclean)
```

Rows: 200

Columns: 24

\$ BRM	<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, ...
\$ sex	<chr> "F", "F", "M", "M", "M", "M", "M"...
\$ age	<dbl> 42, 1, 64, 28, 40, 35, 45, 31, 24...
\$ date	<chr> "16/12/2014", "16/10/2013", "10/1...
\$ P_C	<chr> "P", "P", "P", "P", "P", "P", "P"...
\$ `period of stay`	<dbl> 2884, 8, 5, 288, 63, 89, 43, 2, 8...
\$ port...7	<chr> "OOL", "BNE", "BNE", "MEL", "SYD"...
\$ airline	<chr> "JQ", "FJ", "CZ", "SQ", "MU", "CZ...
\$ port...9	<chr> "CHC", "NAN", "CAN", "SIN", "PVG"...
\$ passport_country	<chr> "A", "B", "B", "D", "B", "B", "B"...
\$ number_trips	<dbl> 5, 2, 32, 21, 16, 7, 6, 18, 3, 34...
\$ eggs	<chr> "0", "0", "0", "0", "0", "yes", "..."
\$ other_flower_seed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ unidentified_seed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

Changing variable names

```
screening <-  
  screening_unclean %>%  
    rename(arrival_port = port...7,  
           check_in_port = port...9,  
           period_of_stay = `period of stay`,  
           passenger_or_crew = P_C)
```

The variable names

don't actually change anywhere

unless we save this new data set as an object

Changing variable names

```
screening <-  
  screening_unclean %>%  
    rename(arrival_port = port...7,  
           check_in_port = port...9,  
           period_of_stay = `period of stay`,  
           passenger_or_crew = P_C)  
  
glimpse(screening)
```

Rows: 200

Columns: 24

\$ BRM	<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, ...
\$ sex	<chr> "F", "F", "M", "M", "M", "M", "M"...
\$ age	<dbl> 42, 1, 64, 28, 40, 35, 45, 31, 24...
\$ date	<chr> "16/12/2014", "16/10/2013", "10/1...
\$ passenger_or_crew	<chr> "P", "P", "P", "P", "P", "P", "P"...
\$ period_of_stay	<dbl> 2884, 8, 5, 288, 63, 89, 43, 2, 8...
\$ arrival_port	<chr> "OOL", "BNE", "BNE", "MEL", "SYD"...
\$ airline	<chr> "JQ", "FJ", "CZ", "SQ", "MU", "CZ...
\$ check_in_port	<chr> "CHC", "NAN", "CAN", "SIN", "PVG"...
\$ passport_country	<chr> "A", "B", "B", "D", "B", "B", "B"...
\$ number_trips	<dbl> 5, 2, 32, 21, 16, 7, 6, 18, 3, 34...
\$ eggs	<chr> "0", "0", "0", "0", "0", "yes", "..."
\$ other_flower_seed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ unidentified	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

Changing variable names

```
screening <-  
  screening_unclean %>%  
    rename(arrival_port = port...7,  
           check_in_port = port...9,  
           period_of_stay = `period of stay`,  
           passenger_or_crew = P_C)
```

```
screening <-  
  screening_unclean %>%  
    rename(arrival_port = port...7, check_in_port = port...9, period_of_stay = `period of stay`, passenger_or_crew =
```

Automatically changing variable names

```
screening <-  
  screening_unclean %>%  
    clean_names("snake")  
  
colnames(screening)
```

[1] "brm"	"sex"
[3] "age"	"date"
[5] "p_c"	"period_of_stay"
[7] "port_7"	"airline"
[9] "port_9"	"passport_country"
[11] "number_trips"	"eggs"
[13] "other_flower_seed"	"unidentified_seed"
[15] "beef"	"other_fruit_vegetable_seed"
[17] "pork"	"other_seeds"
[19] "hides_skins"	"sesame_seed"
[21] "millet"	"traditional_medicine_mixed"
[23] "poultry"	"notes"

"Snake case" is `words_separated_by_an_underscore`, and is the default if you just call `clean_names()` with no options.

```
screening <-  
  screening_unclean %>%  
    clean_names("big_camel")  
  
colnames(screening)
```

[1] "Brm"	"Sex"
[3] "Age"	"Date"
[5] "PC"	"PeriodOfStay"
[7] "Port7"	"Airline"
[9] "Port9"	"PassportCountry"
[11] "NumberTrips"	"Eggs"
[13] "OtherFlowerSeed"	"UnidentifiedSeed"
[15] "Beef"	"OtherFruitVegetableSeed"
[17] "Pork"	"OtherSeeds"
[19] "HidesSkins"	"SesameSeed"
[21] "Millet"	"TraditionalMedicineMixed"
[23] "Poultry"	"Notes"

"Big camel case" is `WordsStartingWithAnUppercaseLetter`.

Obvious errors

```
screening_unclean %>%  
  tabyl(passport_country) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

passport_country	n	percent
A	41	20.5%
A ask Cam?	1	0.5%
B	124	62.0%
C	9	4.5%
D	25	12.5%

```
screening_unclean %>%  
  tabyl(eggs) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

eggs	n	percent
0	188	94.0%
1	11	5.5%
yes	1	0.5%

Correcting factor levels

```
screening <-  
  screening_unclean %>%  
    mutate(passport_country =  
      replace(passport_country,  
        passport_country == "A ask Cam?",  
        "A"))  
screening %>%  
  tabyl(passport_country) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

passport_country	n	percent
A	42	21.0%
B	124	62.0%
C	9	4.5%
D	25	12.5%

Correcting numerical values

```
screening <-  
  screening_unclean %>%  
    mutate(eggs = replace(eggs, eggs == "yes", 1),  
           eggs = as.numeric(eggs))  
  
screening %>%  
  tabyl(eggs) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

eggs	n	percent
0	188	94.0%
1	12	6.0%

```
screening <-  
  screening_unclean %>%  
    mutate(eggs = as.numeric(eggs),  
           eggs = replace(eggs, eggs == "yes", 1))
```

Warning: There was 1 warning in `mutate()`.
! In argument: `eggs = as.numeric(eggs)`.
Caused by warning:
! NAs introduced by coercion

```
screening %>%  
  tabyl(eggs) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

eggs	n	percent	valid_percent
0	188	94.0%	94.5%
1	11	5.5%	5.5%
NA	1	0.5%	-

Extension: case_when()

Alternative to replace() when you want to make a number of changes to the same variable at once.

```
screening <-  
  screening_unclean %>%  
    mutate(P_C = replace(P_C, P_C == "P", "Passenger"),  
           P_C = replace(P_C, P_C == "C", "Crew"))
```

```
screening <-  
  screening_unclean %>%  
    mutate(P_C = case_when(P_C == "P" ~ "Passenger",  
                           P_C == "C" ~ "Crew"))
```

Also useful for other kinds of conditional recoding like converting numerical ranges to categories.

Extension: case_match()

Alternative to replace() to recode entire factors.

```
screening <-  
  screening_unclean %>%  
    rename(arrival_port = port...7) %>%  
    mutate(arrival_port_fullname = case_match(  
      arrival_port,  
      "ADL" ~ "Adelaide",  
      "BNE" ~ "Brisbane",  
      "DRW" ~ "Darwin",  
      "MEL" ~ "Melbourne",  
      "OOL" ~ "Gold Coast",  
      "PER" ~ "Perth",  
      "SYD" ~ "Sydney"))
```

Missing data

```
screening_unclean %>%  
  tabyl(port...7) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

port...7	n	percent	valid_percent
ADL	8	4.0%	4.1%
BNE	26	13.0%	13.2%
DRW	5	2.5%	2.5%
MEL	60	30.0%	30.5%
OOL	5	2.5%	2.5%
PER	8	4.0%	4.1%
SYD	85	42.5%	43.1%
NA	3	1.5%	-

Missing data

```
screening_unclean %>%  
  summarise(Variable = "Number of trips",  
            Mean = mean(number_trips),  
            SD = sd(number_trips),  
            Min = min(number_trips),  
            Med = median(number_trips),  
            Max = max(number_trips),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Variable	Mean	SD	Min	Med	Max	n
Number of trips	84.0	708.7	2.0	11.0	9,999.0	200

Missing data

```
screening <-  
  screening_unclean %>%  
    mutate(number_trips = na_if(number_trips, 9999))  
  
screening %>%  
  summarise(Variable = "Number of trips",  
            Mean = mean(number_trips),  
            SD = sd(number_trips),  
            Min = min(number_trips),  
            Med = median(number_trips),  
            Max = max(number_trips),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Variable	Mean	SD	Min	Med	Max	n
Number of trips	NA	NA	NA	NA	NA	200

Missing data

```
screening <-  
  screening_unclean %>%  
    mutate(number_trips = na_if(number_trips, 9999))
```

```
screening %>%  
  summarise(Variable = "Number of trips",  
            Mean = mean(number_trips, na.rm = TRUE),  
            SD = sd(number_trips, na.rm = TRUE),  
            Min = min(number_trips, na.rm = TRUE),  
            Med = median(number_trips, na.rm = TRUE),  
            Max = max(number_trips, na.rm = TRUE),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Variable	Mean	SD	Min	Med	Max	n
Number of trips	34.2	76.1	2.0	11.0	551.0	200

Missing data

```
screening <-  
  screening_unclean %>%  
    mutate(number_trips = na_if(number_trips, 9999))
```

```
screening %>%  
  summarise(Variable = "Number of trips",  
            Mean = mean(number_trips, na.rm = TRUE),  
            SD = sd(number_trips, na.rm = TRUE),  
            Min = min(number_trips, na.rm = TRUE),  
            Med = median(number_trips, na.rm = TRUE),  
            Max = max(number_trips, na.rm = TRUE),  
            n = sum(!is.na(number_trips))) %>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Variable	Mean	SD	Min	Med	Max	n
Number of trips	34.2	76.1	2.0	11.0	551.0	199

Missing data

```
screening <-  
  screening_unclean %>%  
    mutate(number_trips = na_if(number_trips, 9999))
```

```
screening %>%  
  drop_na(number_trips) %>%  
  summarise(Variable = "Number of trips",  
            Mean = mean(number_trips),  
            SD = sd(number_trips),  
            Min = min(number_trips),  
            Med = median(number_trips),  
            Max = max(number_trips),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(Mean:Max, decimals = 1)
```

Variable	Mean	SD	Min	Med	Max	n
Number of trips	34.2	76.1	2.0	11.0	551.0	199

Missing data

```
screening_unclean %>%  
  tabyl(port...7) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

port...7	n	percent	valid_percent
ADL	8	4.0%	4.1%
BNE	26	13.0%	13.2%
DRW	5	2.5%	2.5%
MEL	60	30.0%	30.5%
OOL	5	2.5%	2.5%
PER	8	4.0%	4.1%
SYD	85	42.5%	43.1%
NA	3	1.5%	-

```
screening <-  
  screening_unclean %>%  
    mutate(port...7 = replace_na(port...7, "Unknown"))  
screening %>%  
  tabyl(port...7) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

port...7	n	percent
ADL	8	4.0%
BNE	26	13.0%
DRW	5	2.5%
MEL	60	30.0%
OOL	5	2.5%
PER	8	4.0%
SYD	85	42.5%
Unknown	3	1.5%

Removing empty rows or columns

```
screening <-  
  screening_unclean %>%  
    remove_empty()
```

value for "which" not specified, defaulting to c("rows", "cols")

There was previously a "notes" column. It was blank for every row, and is now gone.

Extension: dealing with dates



Source: Allison Horst, 2018

Extension: dealing with dates

- Dates are notoriously problematic for software
- Care must be taken, especially with reference dates and different conventions
- Same is true for time
- the tidyverse `lubridate` package provides useful tools to manage dates

Extension: dealing with dates

- Can easily enter dates with a range of formats and separators

```
ymd("20110604")
```

```
[1] "2011-06-04"
```

```
mdy("06-04-2011")
```

```
[1] "2011-06-04"
```

```
dmy("04/06/2011")
```

```
[1] "2011-06-04"
```

```
dmy("4/6/11")
```

```
[1] "2011-06-04"
```

- For these examples, `lubridate` automatically identified the format. This can also be specified as required.

Extension: dealing with dates

- Various information can be extracted

```
month(mdy("06-04-2011"), label = TRUE)
```

```
[1] Jun
```

```
12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < ... < Dec
```

```
month(mdy("06-04-2011"))
```

```
[1] 6
```

```
wday(mdy("06-04-2011"), label = TRUE)
```

```
[1] Sat
```

```
Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday(mdy("06-04-2011"))
```

```
[1] 7
```

Extension: dealing with dates

- date arithmetic is also possible

```
dmy("04-06-2011") + 1
```

```
[1] "2011-06-05"
```

```
dmy("24-06-2011") - dmy("04-06-2011")
```

Time difference of 20 days

- take care in interpreting the increment: in this case it is days

Extension: dealing with dates

- processing dates for our airport screening example

```
glimpse(screening_unclean$date)
```

```
chr [1:200] "16/12/2014" "16/10/2013" "10/10/2014" "17/10/2014" ...
```

```
screening_unclean %>%  
  tabyl(date) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

date	n	percent
1/02/2013	1	0.5%
1/03/2014	1	0.5%
1/04/2014	1	0.5%
1/06/2013	1	0.5%
1/07/2013	1	0.5%
1/07/2014	1	0.5%
1/09/2013	1	0.5%
1/12/2013	1	0.5%
10/02/2013	1	0.5%
10/04/2014	1	0.5%
10/05/2014	1	0.5%
10/06/2013	1	0.5%
10/06/2014	1	0.5%

Extension: dealing with dates

- processing dates for our airport screening example

```
screening <-  
  screening_unclean %>%  
    mutate(date = dmy(date))
```

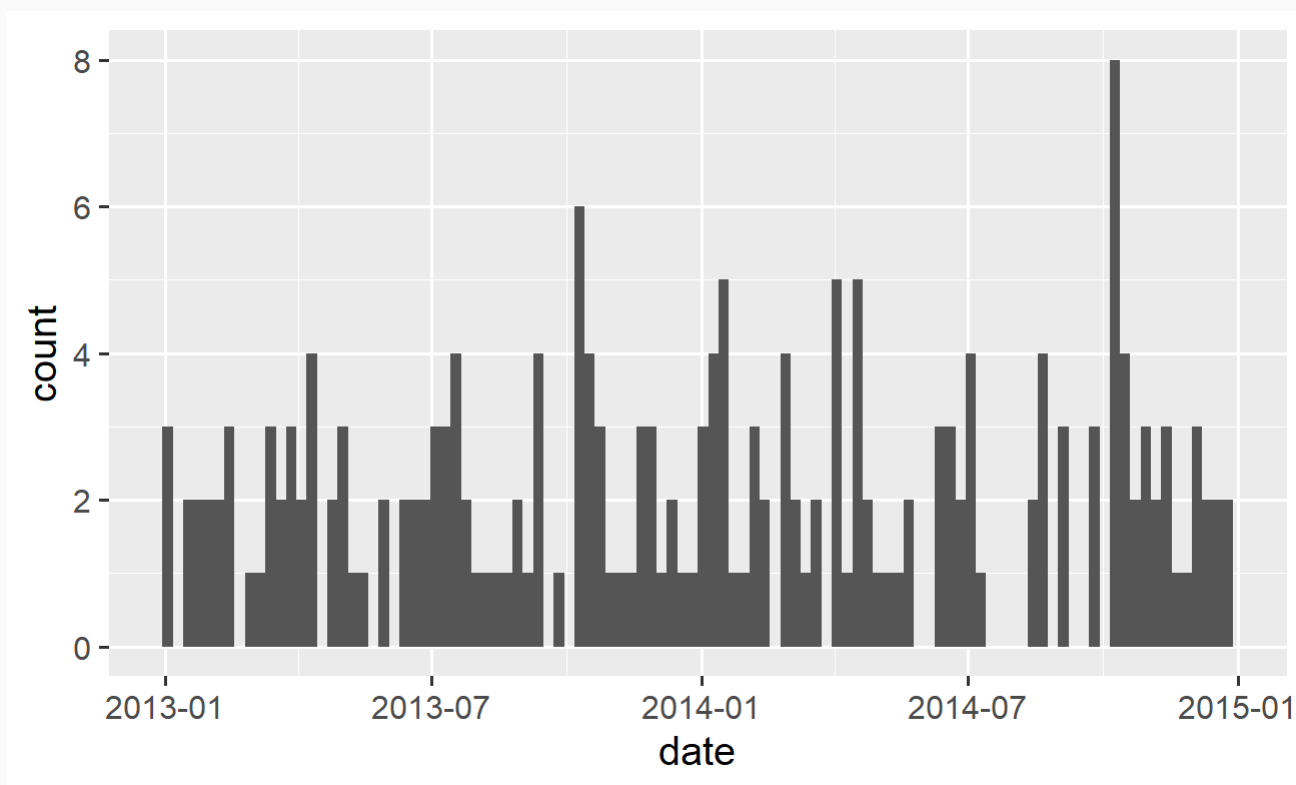
```
screening %>%  
  tabyl(date) %>%  
  adorn_pct_formatting() %>%  
  gt()
```

date	n	percent
2013-01-02	1	0.5%
2013-01-04	1	0.5%
2013-01-06	1	0.5%
2013-01-16	1	0.5%
2013-01-18	1	0.5%
2013-01-25	2	1.0%
2013-01-29	1	0.5%
2013-02-01	1	0.5%
2013-02-06	1	0.5%
2013-02-10	1	0.5%
2013-02-14	2	1.0%

Extension: dealing with dates

- Now that R recognises these as dates, they can be plotted sensibly.

```
screening %>%  
  ggplot(aes(x = date)) +  
    geom_histogram(binwidth = 7)
```



Putting it all together

```
screening_unclean <-  
  read_csv("../5 data/airport_screening_unclean.csv")
```

```
screening <-  
  screening_unclean %>%  
    rename(arrival_port = port...7, check_in_port = port...9,  
           period_of_stay = `period of stay`, passenger_or_crew = P_C) %>%  
    mutate(passport_country = replace(passport_country, passport_country == "A ask Cam?", "A"),  
           eggs = as.numeric(replace(eggs, eggs == "yes", 1)),  
           number_trips = na_if(number_trips, 9999),  
           arrival_port = replace_na(arrival_port, "Unknown"),  
           date = dmy(date))
```

```
glimpse(screening)
```

Putting it all together


Rows: 200

Columns: 24

\$ BRM	<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, ...
\$ sex	<chr> "F", "F", "M", "M", "M", "M", "M"...
\$ age	<dbl> 42, 1, 64, 28, 40, 35, 45, 31, 24...
\$ date	<date> 2014-12-16, 2013-10-16, 2014-10-...
\$ passenger_or_crew	<chr> "P", "P", "P", "P", "P", "P", "P"...
\$ period_of_stay	<dbl> 2884, 8, 5, 288, 63, 89, 43, 2, 8...
\$ arrival_port	<chr> "OOL", "BNE", "BNE", "MEL", "SYD"...
\$ airline	<chr> "JQ", "FJ", "CZ", "SQ", "MU", "CZ...
\$ check_in_port	<chr> "CHC", "NAN", "CAN", "SIN", "PVG"...
\$ passport_country	<chr> "A", "B", "B", "D", "B", "B", "B"...
\$ number_trips	<dbl> 5, 2, 32, 21, 16, 7, 6, 18, 3, 34...
\$ eggs	<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...
\$ other_flower_seed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ unidentified_seed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ beef	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ other_fruit_vegetable_seed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ pork	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ other_seeds	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ hides_skins	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ sesame_seed	<dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...
\$ millet	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
\$ traditional_medicine_mixed	<dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

Case study: reproducible data importing

An Excel file from the ABS

	A	B	C	D	E	F	G	H	I	J	K
1	 Australian Bureau of Statistics	Australian Bureau of Statistics									
2	45190DO001_201819 Recorded Crime – Offenders, Australia, 2018–19										
3	Released at 11.30am (Canberra time) Thurs 6 February 2020										
4	Table 1 Offenders, Principal offence (divisions and selected subdivisions), 2008–09 to 2018–19										
5		Number									
6	Principal offence	2008–09	2009–10	2010–11	2011–12	2012–13	2013–14	2014–15	2015–16	2016–17	2017–18
7	01 Homicide and related offences	816	880	757	661	728	677	712	702	665	665
8	011 Murder	318	326	303	261	296	270	304	276	245	245
9	012 Attempted Murder	232	227	192	168	173	169	172	182	161	161
10	013 Manslaughter and driving causing death	260	322	256	237	258	236	235	247	253	253
11											
12	02 Acts intended to cause injury	72,069	72,280	70,573	68,476	70,371	71,395	72,721	76,206	78,421	78,421
13	021 Assault	69,218	69,079	67,219	64,981	66,406	67,081	68,002	70,986	72,964	72,964
14	029 Other acts intended to cause injury	2,728	3,082	3,317	3,449	3,913	4,311	4,721	5,220	5,460	5,460
15											
16	03 Sexual assault and related offences	6,340	6,412	5,796	6,019	6,173	7,255	7,639	7,898	8,123	8,123
17	031 Sexual assault	4,979	4,889	4,548	4,591	4,475	5,163	5,445	5,530	5,844	5,844
18	032 Non–assaultive sexual offences	1,363	1,517	1,247	1,431	1,696	2,092	2,190	2,370	2,276	2,276
19											
20	04 Dangerous/negligent acts	1,680	1,823	1,801	1,899	1,897	2,079	2,135	2,247	2,465	2,465
21											
22	05 Abduction/harassment	3,598	3,948	3,563	3,488	3,773	3,975	4,592	4,857	4,943	4,943
23											
24	06 Robbery/extortion	3,911	4,070	3,996	3,620	3,634	3,330	3,172	3,194	3,388	3,388
25											
26	07 Unlawful entry with intent	15,887	15,609	15,158	13,934	12,713	12,483	11,742	12,297	12,360	12,360
27											
28	08 Theft	58,941	64,382	64,806	64,587	62,520	47,913	39,424	42,060	42,221	42,221
29	081 Motor vehicle theft	5,438	5,226	5,029	4,929	4,693	4,409	4,539	4,776	4,863	4,863
30	082 Theft (except motor vehicles)	48,203	53,802	54,949	54,873	53,086	38,764	30,164	32,337	32,529	32,529

Contents

Table 1

Table 2

Table 3

Table 4

Table 5

Case study: reproducible data importing

4	Table 1 Offenders, Principal offence (divisions and selected subdivisions), 20			
5				
6	Principal offence	2008–09	2009–10	
7	01 Homicide and related offences	816	880	
8	011 Murder	318	326	
9	012 Attempted Murder	232	227	
10	013 Manslaughter and driving causing death	260	322	
11				
12	02 Acts intended to cause injury	72,069	72,280	
13	021 Assault	69,218	69,079	
14	029 Other acts intended to cause injury	2,728	3,082	
15				
16	03 Sexual assault and related offences	6,340	6,412	
17	031 Sexual assault	4,979	4,889	
18	032 Non–assaultive sexual offences	1,363	1,517	
19				

Case study: reproducible data importing

```
library(readxl)
excel_ex1 <- read_excel("../5 data/1. offenders, australia.xls",
                        sheet = 2, range = "A6:L58")
excel_ex1
```

A tibble: 52 × 12

	`Principal offence`	`2008-09`	`2009-10`	`2010-11`	`2011-12`
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	01 Homicide and related of...	816	880	757	661
2	011 Murder	318	326	303	261
3	012 Attempted Murder	232	227	192	168
4	013 Manslaughter and drivi...	260	322	256	237
5	<NA>	NA	NA	NA	NA
6	02 Acts intended to cause ...	72069	72280	70573	68476
7	021 Assault	69218	69079	67219	64981
8	029 Other acts intended to...	2728	3082	3317	3449
9	<NA>	NA	NA	NA	NA
10	03 Sexual assault and rela...	6340	6412	5796	6019

i 42 more rows

i 7 more variables: `2012-13` <dbl>, `2013-14` <dbl>,

`2014-15` <dbl>, `2015-16` <dbl>, `2016-17` <dbl>,

`2017-18` <dbl>, `2018-19` <dbl>

Case study: reproducible data importing

```
excel_ex2 <- excel_ex1 %>%  
  clean_names()  
excel_ex2
```

```
# A tibble: 52 × 12
```

	principal_offence	x2008_09	x2009_10	x2010_11	x2011_12	x2012_13
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	01 Homicide and relat...	816	880	757	661	728
2	011 Murder	318	326	303	261	296
3	012 Attempted Murder	232	227	192	168	173
4	013 Manslaughter and ...	260	322	256	237	258
5	<NA>	NA	NA	NA	NA	NA
6	02 Acts intended to c...	72069	72280	70573	68476	70371
7	021 Assault	69218	69079	67219	64981	66406
8	029 Other acts intend...	2728	3082	3317	3449	3913
9	<NA>	NA	NA	NA	NA	NA
10	03 Sexual assault and...	6340	6412	5796	6019	6173

```
# i 42 more rows
```

```
# i 6 more variables: x2013_14 <dbl>, x2014_15 <dbl>, x2015_16 <dbl>,
```

```
#   x2016_17 <dbl>, x2017_18 <dbl>, x2018_19 <dbl>
```


Case study: reproducible data importing

```
excel_ex2 <- excel_ex1 %>%  
  clean_names() %>%  
  remove_empty()  
excel_ex2
```

A tibble: 38 × 12

	principal_offence	x2008_09	x2009_10	x2010_11	x2011_12	x2012_13
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	01 Homicide and relat...	816	880	757	661	728
2	011 Murder	318	326	303	261	296
3	012 Attempted Murder	232	227	192	168	173
4	013 Manslaughter and ...	260	322	256	237	258
5	02 Acts intended to c...	72069	72280	70573	68476	70371
6	021 Assault	69218	69079	67219	64981	66406
7	029 Other acts intend...	2728	3082	3317	3449	3913
8	03 Sexual assault and...	6340	6412	5796	6019	6173
9	031 Sexual assault	4979	4889	4548	4591	4475
10	032 Non-assaultive se...	1363	1517	1247	1431	1696

i 28 more rows

i 6 more variables: x2013_14 <dbl>, x2014_15 <dbl>, x2015_16 <dbl>,

x2016_17 <dbl>, x2017_18 <dbl>, x2018_19 <dbl>

Case study: reproducible data importing

```
excel_ex2 <- excel_ex1 %>%  
  clean_names() %>%  
  remove_empty() %>%  
  slice(1, 5, 8, 11:15, 19, 24, 28:30)  
excel_ex2
```

A tibble: 13 × 12

	principal_offence	x2008_09	x2009_10	x2010_11	x2011_12	x2012_13
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	01 Homicide and relat...	816	880	757	661	728
2	02 Acts intended to c...	72069	72280	70573	68476	70371
3	03 Sexual assault and...	6340	6412	5796	6019	6173
4	04 Dangerous/negligen...	1680	1823	1801	1899	1897
5	05 Abduction/harassme...	3598	3948	3563	3488	3773
6	06 Robbery/extortion	3911	4070	3996	3620	3634
7	07 Unlawful entry wit...	15887	15609	15158	13934	12713
8	08 Theft	58941	64382	64806	64587	62520
9	09 Fraud/deception	10138	9651	9046	8690	10126
10	10 Illicit drug offen...	56310	57899	58700	60729	65346
11	11 Weapons/explosives	8962	8654	8820	9651	10524
12	12 Property damage an...	21953	21067	19937	18385	18000
13	13 Public order offen...	65962	75200	73845	69442	73940

i 6 more variables: x2013_14 <dbl>, x2014_15 <dbl>, x2015_16 <dbl>,

x2016_17 <dbl>, x2017_18 <dbl>, x2018_19 <dbl>

Case study: reproducible data importing

```
excel_ex2 <- excel_ex1 %>%  
  clean_names() %>%  
  remove_empty() %>%  
  filter(str_detect(principal_offence, "[0-9][0-9] "))  
excel_ex2
```

A tibble: 15 × 12

	principal_offence	x2008_09	x2009_10	x2010_11	x2011_12	x2012_13
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	01 Homicide and relat...	816	880	757	661	728
2	02 Acts intended to c...	72069	72280	70573	68476	70371
3	03 Sexual assault and...	6340	6412	5796	6019	6173
4	04 Dangerous/negligen...	1680	1823	1801	1899	1897
5	05 Abduction/harassme...	3598	3948	3563	3488	3773
6	06 Robbery/extortion	3911	4070	3996	3620	3634
7	07 Unlawful entry wit...	15887	15609	15158	13934	12713
8	08 Theft	58941	64382	64806	64587	62520
9	09 Fraud/deception	10138	9651	9046	8690	10126
10	10 Illicit drug offen...	56310	57899	58700	60729	65346
11	11 Weapons/explosives	8962	8654	8820	9651	10524
12	12 Property damage an...	21953	21067	19937	18385	18000
13	13 Public order offen...	65962	75200	73845	69442	73940
14	15 Offences against j...	26201	26092	23776	22927	24288
15	16 Miscellaneous offe...	16957	19228	22190	24397	29453

Case study: reproducible data importing

```
excel_ex2 <- excel_ex1 %>%  
  clean_names() %>%  
  remove_empty() %>%  
  slice(1, 5, 8, 11:15, 19, 24, 28:30)  
excel_ex2
```

A tibble: 13 × 12

	principal_offence	x2008_09	x2009_10	x2010_11	x2011_12	x2012_13
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	01 Homicide and relat...	816	880	757	661	728
2	02 Acts intended to c...	72069	72280	70573	68476	70371
3	03 Sexual assault and...	6340	6412	5796	6019	6173
4	04 Dangerous/negligen...	1680	1823	1801	1899	1897
5	05 Abduction/harassme...	3598	3948	3563	3488	3773
6	06 Robbery/extortion	3911	4070	3996	3620	3634
7	07 Unlawful entry wit...	15887	15609	15158	13934	12713
8	08 Theft	58941	64382	64806	64587	62520
9	09 Fraud/deception	10138	9651	9046	8690	10126
10	10 Illicit drug offen...	56310	57899	58700	60729	65346
11	11 Weapons/explosives	8962	8654	8820	9651	10524
12	12 Property damage an...	21953	21067	19937	18385	18000
13	13 Public order offen...	65962	75200	73845	69442	73940

i 6 more variables: x2013_14 <dbl>, x2014_15 <dbl>, x2015_16 <dbl>,

x2016_17 <dbl>, x2017_18 <dbl>, x2018_19 <dbl>

Exercise 2.4.