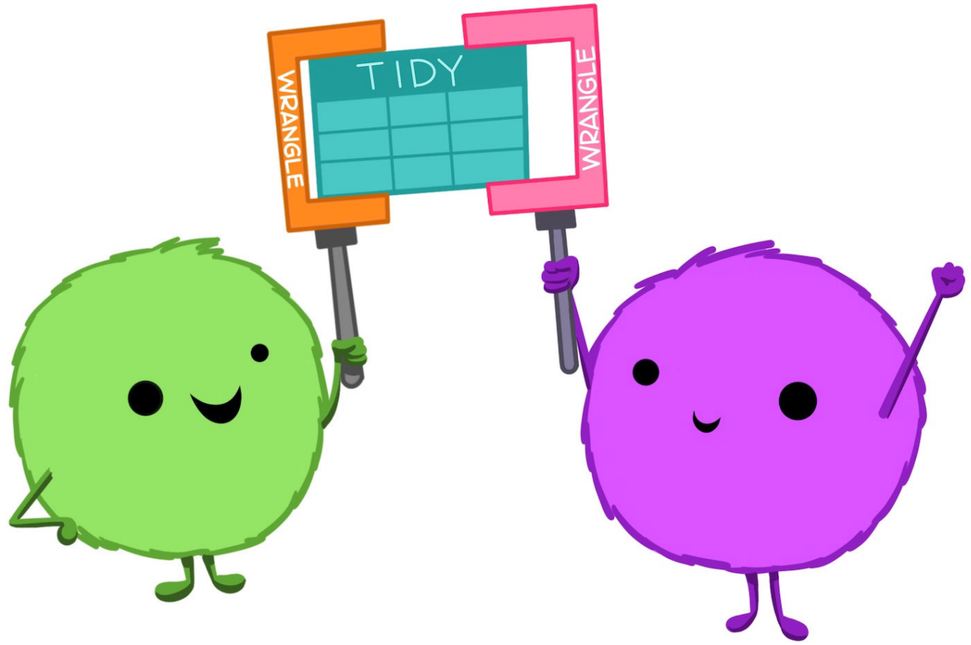
Effective Data Wrangling

Data Exploration and Visualisation

Dr Zak Varty

What is Data Wrangling?

- You have raw data, now what?
- Subset, summarise, create, merge…
- wrangling = manipulation = munging
- Fly-by tour, focusing on common operations



Source: Openscapes blog by J Lowndes and A Horst.

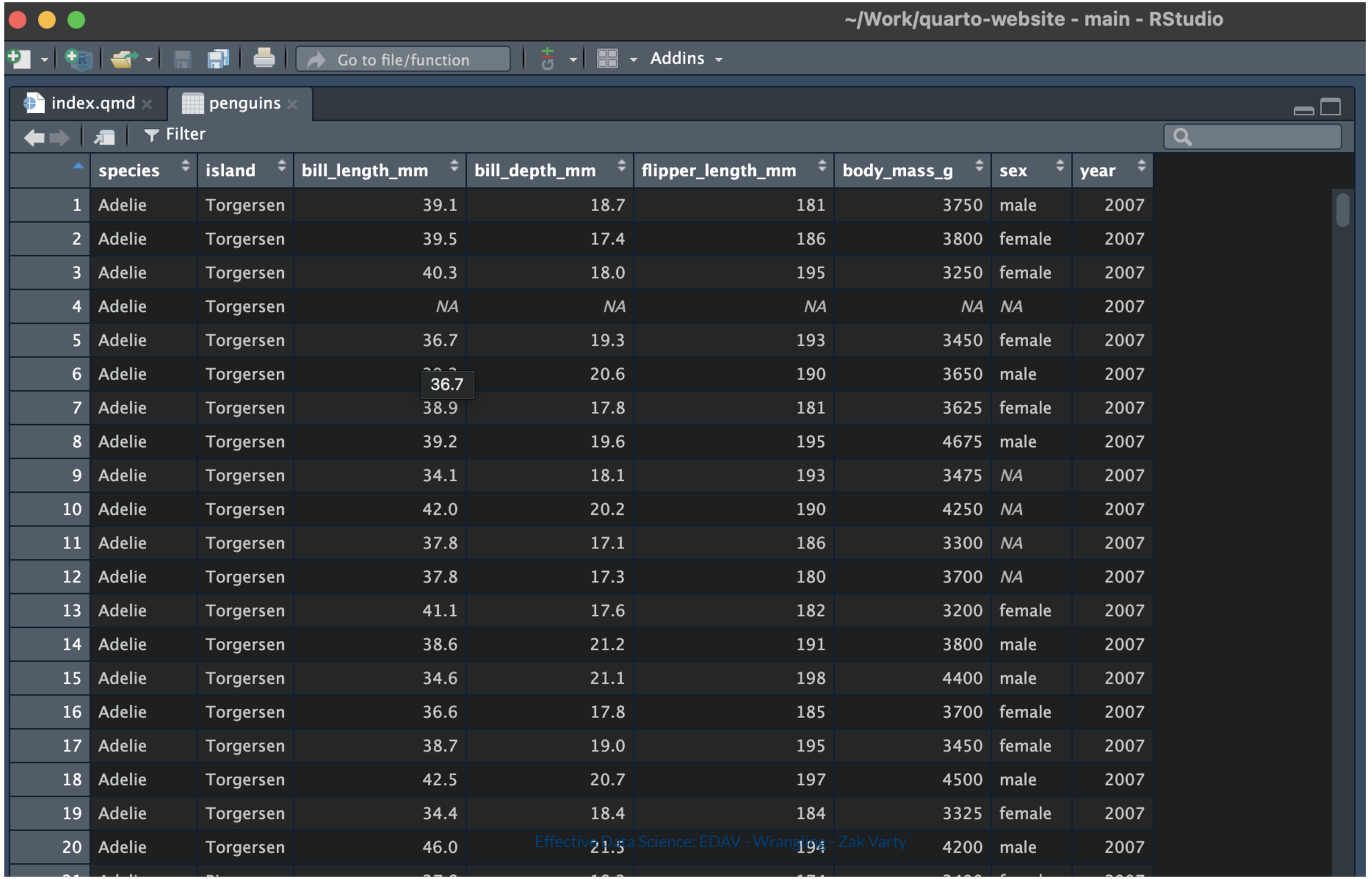
Example Data Sets


```
1 library(palmerpenguins)
2 pengins <- palmerpenguins::penguins
3 cars <- datasets::mtcars
```

Viewing Your Data

View()

```
1 View(penguins)
```



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
1	Adelie	Torgersen	39.1	18.7	181	3750	male	2007
2	Adelie	Torgersen	39.5	17.4	186	3800	female	2007
3	Adelie	Torgersen	40.3	18.0	195	3250	female	2007
4	Adelie	Torgersen	*NA*	*NA*	*NA*	*NA*	*NA*	2007
5	Adelie	Torgersen	36.7	19.3	193	3450	female	2007
6	Adelie	Torgersen	[illegible]	20.6	190	3650	male	2007
7	Adelie	Torgersen	38.9	17.8	181	3625	female	2007
8	Adelie	Torgersen	39.2	19.6	195	4675	male	2007
9	Adelie	Torgersen	34.1	18.1	193	3475	*NA*	2007
10	Adelie	Torgersen	42.0	20.2	190	4250	*NA*	2007
11	Adelie	Torgersen	37.8	17.1	186	3300	*NA*	2007
12	Adelie	Torgersen	37.8	17.3	180	3700	*NA*	2007
13	Adelie	Torgersen	41.1	17.6	182	3200	female	2007
14	Adelie	Torgersen	38.6	21.2	191	3800	male	2007
15	Adelie	Torgersen	34.6	21.1	198	4400	male	2007
16	Adelie	Torgersen	36.6	17.8	185	3700	female	2007
17	Adelie	Torgersen	38.7	19.0	195	3450	female	2007
18	Adelie	Torgersen	42.5	20.7	197	4500	male	2007
19	Adelie	Torgersen	34.4	18.4	184	3325	female	2007
20	Adelie	Torgersen	46.0	21.5	194	4200	male	2007

head()

```
1 head(x = pengins, n = 7)
```

```
# A tibble: 7 × 8
  species island    bill_length_mm bill_depth_mm flipper_l…¹ body_…² sex    year
  <fct>   <fct>              <dbl>         <dbl>       <int>   <int> <fct> <int>
1 Adelie  Torgersen           39.1          18.7         181    3750 male   2007
2 Adelie  Torgersen           39.5          17.4         186    3800 fema…  2007
3 Adelie  Torgersen           40.3          18           195    3250 fema…  2007
4 Adelie  Torgersen           NA            NA            NA      NA <NA>   2007
5 Adelie  Torgersen           36.7          19.3         193    3450 fema…  2007
6 Adelie  Torgersen           39.3          20.6         190    3650 male   2007
7 Adelie  Torgersen           38.9          17.8         181    3625 fema…  2007
# … with abbreviated variable names ¹flipper_length_mm, ²body_mass_g
```

str()

```
1 str(penguins)
```

```
tibble [344 × 8] (S3: tbl_df/tbl/data.frame)
 $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
 $ bill_depth_mm    : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
 $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
 $ body_mass_g      : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
 $ sex              : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...
 $ year             : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

names()

```
1 names(penguins)
```

```
[1] "species"           "island"            "bill_length_mm"
[4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
[7] "sex"               "year"
```

```
1 colnames(cars)
```

```
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"
```

```
1 rownames(cars)
```

```
 [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
 [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
 [7] "Duster 360"          "Merc 240D"           "Merc 230"
[10] "Merc 280"            "Merc 280C"           "Merc 450SE"
[13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood"
[16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
[19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
[22] "Dodge Challenger"    "AMC Javelin"         "Camaro Z28"
[25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"
[28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
[31] "Maserati Bora"       "Volvo 142E"
```

Renaming Variables

colnames()

```
cars_renamed <- cars
colnames(cars_renamed)[1] <- "miles_per_gallon"
colnames(cars_renamed)
```

```
[1] "miles_per_gallon" "cyl"              "disp"             "hp"
[5] "drat"             "wt"               "qsec"             "vs"
[9] "am"               "gear"             "carb"
```

dplyr::rename()

```
1 library(dplyr)
2 cars_renamed <- rename(.data = cars_renamed, cylinders = cyl)
3 colnames(cars_renamed)
```

```
[1] "miles_per_gallon" "cylinders"        "disp"             "hp"
[5] "drat"             "wt"               "qsec"             "vs"
[9] "am"               "gear"             "carb"
```

```
1 cars_renamed <- cars_renamed %>%
2   rename(displacement = disp) %>%
3   rename(horse_power = hp) %>%
4   rename(rear_axel_ratio = drat)
5
6 colnames(cars_renamed)
```

```
[1] "miles_per_gallon" "cylinders"        "displacement"     "horse_power"
[5] "rear_axel_ratio"  "wt"               "qsec"             "vs"
[9] "am"               "gear"             "carb"
```

Subsetting

Subsetting with Base R (1)

Subsetting by index

```
# First row
penguins[1, ]

# First Column
penguins[ , 1]

# Rows 2-3 of columns 1, 2 and 4
penguins[2:3, c(1, 2, 4)]
```

Drop rows or columns with negative indices

```
# Drop all but first row
penguins[-(2:344), ]

# Drop all but first column
penguins[ , -(2:8)]
```

Subsetting with Base R (2)

Subset using logical vector

```
1 is_bill_long <- penguins$bill_length_mm > 70
2 pengiuns[is_bill_long, ]
```

Subset using names (note different objects)

```
1 pengins[ , "species"]   # tibble
2 penguins$species       # vector
```

Subsetting with {dplyr} (1)

```r
penguins %>%
  select(species, island, body_mass_g) %>%
  print(n = 4)
```

```
# A tibble: 344 × 3
  species island    body_mass_g
  <fct>   <fct>           <int>
1 Adelie  Torgersen        3750
2 Adelie  Torgersen        3800
3 Adelie  Torgersen        3250
4 Adelie  Torgersen          NA
# … with 340 more rows
```

```r
penguins %>%
  select(species, island, body_mass_g) %>%
  filter(body_mass_g > 6000)
```

```
# A tibble: 2 × 3
  species island body_mass_g
  <fct>   <fct>        <int>
1 Gentoo  Biscoe        6300
2 Gentoo  Biscoe        6050
```

Subsetting with {dplyr} (2)

```
penguins %>%
  select(species, island, body_mass_g) %>%
  filter(!(body_mass_g > 6000)) %>%
  print(n = 4)
```

```
# A tibble: 340 × 3
  species island    body_mass_g
  <fct>   <fct>           <int>
1 Adelie  Torgersen        3750
2 Adelie  Torgersen        3800
3 Adelie  Torgersen        3250
4 Adelie  Torgersen        3450
# … with 336 more rows
```

```
penguins %>%
  select(species, island, body_mass_g) %>%
  filter(!(body_mass_g > 6000)) %>%
  select(!c(species, island)) %>%
  print(n = 4)
```

```
# A tibble: 340 × 1
  body_mass_g
        <int>
1        3750
2        3800
3        3250
4        3450
# … with 336 more rows
```

Creating new variables

New Variables with Base R

```r
# add weight column
cars_renamed$weight <- cars_renamed$wt
```

```r
# remove wt column
cars_renamed <- cars_renamed[ ,-which(names(cars_renamed) == "wt")]
head(cars_renamed, n = 3)
```

```
              miles_per_gallon cylinders displacement horse_power
Mazda RX4                 21.0         6          160         110
Mazda RX4 Wag             21.0         6          160         110
Datsun 710                22.8         4          108          93
              rear_axel_ratio  qsec vs am gear carb weight
Mazda RX4                3.90 16.46  0  1    4    4  2.620
Mazda RX4 Wag            3.90 17.02  0  1    4    4  2.875
Datsun 710               3.85 18.61  1  1    4    1  2.320
```

New Variables with `dplyr::mutate()`

```
cars_renamed <- cars_renamed %>%
  mutate(weight_kg = weight * 1000)

cars_renamed %>%
  select(weight, weight_kg) %>%
  head(n = 3)
```

```
               weight weight_kg
Mazda RX4       2.620      2620
Mazda RX4 Wag   2.875      2875
Datsun 710      2.320      2320
```

```
cars_renamed <- cars_renamed %>%
  mutate(cylinder_adjusted_mpg = miles_per_gallon / cylinders)
```

Row names / id to column

```
cars %>%
  tibble::rownames_to_column(var = "model") %>%
  head(n = 3)
```

```
          model  mpg cyl disp  hp drat    wt  qsec vs am gear carb
1     Mazda RX4 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
2 Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
3    Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

```
cars %>%
  tibble::rowid_to_column(var = "row_id") %>%
  head(n = 3)
```

```
  row_id  mpg cyl disp  hp drat    wt  qsec vs am gear carb
1      1 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
2      2 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
3      3 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

Summaries

```
penguins %>%
summarise(average_bill_length_mm = mean(bill_length_mm, na.rm = TRUE))
```

```
# A tibble: 1 × 1
  average_bill_length_mm
                   <dbl>
1                   43.9
```

```
bill_length_mm_summary <- penguins %>%
  summarise(
    mean = mean(bill_length_mm, na.rm = TRUE),
    median = median(bill_length_mm, na.rm = TRUE),
    min = max(bill_length_mm, na.rm = TRUE),
    q_0 = min(bill_length_mm, na.rm = TRUE),
    q_1 = quantile(bill_length_mm, prob = 0.25, na.rm = TRUE),
    q_2 = median(bill_length_mm, na.rm = TRUE),
    q_3 = quantile(bill_length_mm, prob = 0.25, na.rm = TRUE),
    q_4 = max(bill_length_mm, na.rm = TRUE))

bill_length_mm_summary
```

```
# A tibble: 1 × 8
   mean median   min   q_0   q_1   q_2   q_3   q_4
  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  43.9   44.4  59.6  32.1  39.2  44.4  39.2  59.6
```

Grouped Operations

Grouped Operations

```r
penguins %>%
  group_by(species) %>%
  summarise(
    mean = mean(bill_length_mm, na.rm = TRUE),
    median = median(bill_length_mm, na.rm = TRUE),
    min = max(bill_length_mm, na.rm = TRUE),
    q_0 = min(bill_length_mm, na.rm = TRUE),
    q_1 = quantile(bill_length_mm, prob = 0.25, na.rm = TRUE),
    q_2 = median(bill_length_mm, na.rm = TRUE),
    q_3 = quantile(bill_length_mm, prob = 0.25, na.rm = TRUE),
    q_4 = max(bill_length_mm, na.rm = TRUE))
```

```
# A tibble: 3 × 9
  species    mean median   min   q_0   q_1   q_2   q_3   q_4
  <fct>     <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Adelie     38.8   38.8  46    32.1  36.8  38.8  36.8  46
2 Chinstrap  48.8   49.6  58    40.9  46.3  49.6  46.3  58
3 Gentoo     47.5   47.3  59.6  40.9  45.3  47.3  45.3  59.6
```

Multiple Groups

```r
penguin_summary_stats <- penguins %>%
  group_by(species, island) %>%
  summarise(
    mean = mean(bill_length_mm, na.rm = TRUE),
    median = median(bill_length_mm, na.rm = TRUE),
    min = max(bill_length_mm, na.rm = TRUE),
    q_0 = min(bill_length_mm, na.rm = TRUE),
    q_1 = quantile(bill_length_mm, prob = 0.25, na.rm = TRUE),
    q_2 = median(bill_length_mm, na.rm = TRUE),
    q_3 = quantile(bill_length_mm, prob = 0.25, na.rm = TRUE),
    q_4 = max(bill_length_mm, na.rm = TRUE))

penguin_summary_stats
```

```
# A tibble: 5 × 10
# Groups:   species [3]
  species   island     mean median   min   q_0   q_1   q_2   q_3   q_4
  <fct>     <fct>     <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Adelie    Biscoe     39.0   38.7  45.6  34.5  37.7  38.7  37.7  45.6
2 Adelie    Dream      38.5   38.6  44.1  32.1  36.8  38.6  36.8  44.1
3 Adelie    Torgersen  39.0   38.9  46    33.5  36.7  38.9  36.7  46
4 Chinstrap Dream      48.8   49.6  58    40.9  46.3  49.6  46.3  58
5 Gentoo    Biscoe     47.5   47.3  59.6  40.9  45.3  47.3  45.3  59.6
```

Ungrouping

By default each `summarise()` will undo one level of grouping.

```
1 class(penguin_summary_stats)
```

```
[1] "grouped_df" "tbl_df"     "tbl"        "data.frame"
```

Use an appropriate number of calls or `ungroup()`.

```
1 penguin_summary_stats %>%
2   summarise_all(mean, na.rm = TRUE)
```

```
# A tibble: 3 × 10
  species   island  mean median   min   q_0   q_1   q_2   q_3   q_4
  <fct>      <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Adelie        NA  38.8   38.7  45.2  33.4  37.0  38.7  37.0  45.2
2 Chinstrap     NA  48.8   49.6  58    40.9  46.3  49.6  46.3  58
3 Gentoo        NA  47.5   47.3  59.6  40.9  45.3  47.3  45.3  59.6
```

```
1 ungroup(penguin_summary_stats)
```

```
# A tibble: 5 × 10
  species   island     mean median   min   q_0   q_1   q_2   q_3   q_4
  <fct>     <fct>     <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Adelie    Biscoe     39.0   38.7  45.6  34.5  37.7  38.7  37.7  45.6
2 Adelie    Dream      38.5   38.6  44.1  32.1  36.8  38.6  36.8  44.1
3 Adelie    Torgersen  39.0   38.9  46    33.5  36.7  38.9  36.7  46
4 Chinstrap Dream      48.8   49.6  58    40.9  46.3  49.6  46.3  58
5 Gentoo    Biscoe     47.5   47.3  59.6  40.9  45.3  47.3  45.3  59.6
```

Reordering Factors (1)

- Factors are stored as integers, ordering can make tables and plots confusing.

```r
tshirts <- tibble::tibble(
  id = 1:12,
  size = as.factor(c("L", NA, "M", "S", "XS", "M", "XXL", "L", "XS", "M", "L", "S"))
)

tshirts %>% group_by(size) %>% summarise(count = n())
```

```
# A tibble: 6 × 2
  size  count
  <fct> <int>
1 L         3
2 M         3
3 S         2
4 XS        2
5 XXL       1
6 <NA>      1
```

Reodrdering Factors (2)

- create new variable with the factor in the correct order.

```r
tidy_tshirt_levels <- c("XS", "S", "M", "L", "XL", "XXL", NA)

tshirts %>%
  mutate(size_tidy = factor(size, levels = tidy_tshirt_levels)) %>%
  group_by(size_tidy, .drop = FALSE ) %>%
  summarise(count = n())
```

```
# A tibble: 7 × 2
  size_tidy count
  <fct>     <int>
1 XS            2
2 S             2
3 M             3
4 L             3
5 XL            0
6 XXL           1
7 <NA>          1
```

Be Aware!

Be Aware: Factors

Useful `{forcats}` functions:

- `fct_reorder()`: Reordering a factor by another variable.
- `fct_infreq()`: Reordering a factor by the frequency of values.
- `fct_relevel()`: Changing the order of a factor by hand.
- `fct_lump()`: Collapsing the least/most frequent values of a factor into “other”.

Check out the forcats vignette or the factors chapter of R4DS.

Be Aware: Strings

- Working with text data is it's own skill.
- Requires some knowledge of regular expressions.
- `{stringr}` simplifies this, somewhat.
- Learn as you need it: strings section of R4DS.

Be Aware: Dates and Times

Recall ISO standard and proceed with caution

$$YYYY - MM - DD$$

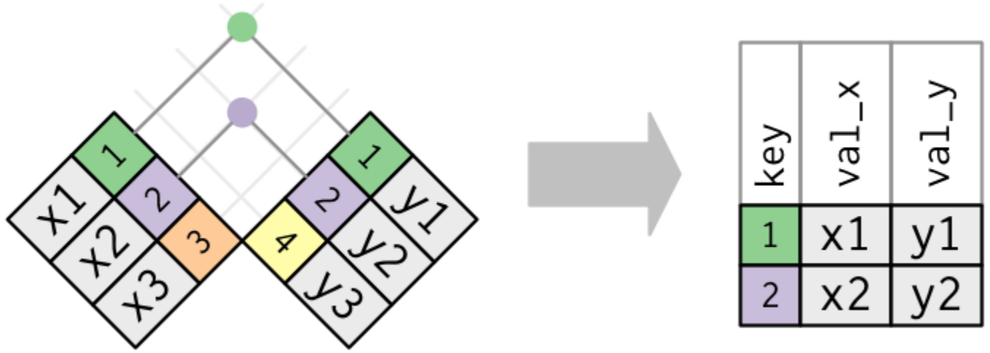
- How many days are there in a year / month? How many hours are there in a day?
- How many seconds are there in a minute?
- What calendar are you using? When does the year begin and how many months?

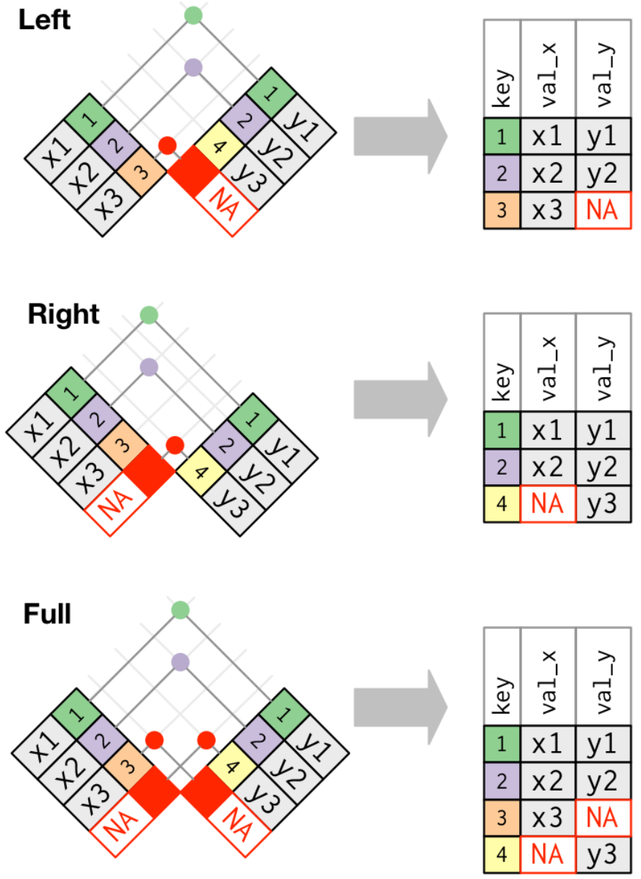
`{lubridate}` makes this easier, see dates and times examples from R4DS.

Be Aware: Relational Data

- Data on the same observational units stored across two or more tables.
- Relational data chapter of R4DS.

Inner Join





Ubiquity of Relational Data

- Relational data base management systems used across data science,
  - More efficient and fewer privacy concerns.
- SQL as a database management language
  - Inspiration for `{dplyr}`, translate with `{dbplyr}`
- More data base theory and efficient queries - big data or introductory SQl books.

Wrapping up

- Learned how to wrangle tabular data in R with base R and `{dplyr}`
- Met the idea of relational data and `{dplyr}`'s relationship to SQL
- Become aware of some tricky data types and packages that can help.