2-Data Wrangling

Felipe Melo

Nottingham Trent University - UK

You should know today

- Why data wrangling?
- How to plan data wrangling
- Basic skills
- Application with an example

Before we begin

- R and Rstudio installed
- Don’t panic
- Everything is reproducible
- You’ll have to train to fix the content

Tibbles <- click on the title


```
 1 library(tidyverse)
 2 library(palmerpenguins)
 3
 4 data("penguins")
 5 penguins %>%
 6   select(1:5)
```

```
# A tibble: 344 × 5
   species island    bill_length_mm bill_depth_mm flipper_length_mm
   <fct>   <fct>              <dbl>         <dbl>             <int>
 1 Adelie  Torgersen           39.1          18.7               181
 2 Adelie  Torgersen           39.5          17.4               186
 3 Adelie  Torgersen           40.3          18                 195
 4 Adelie  Torgersen           NA            NA                  NA
 5 Adelie  Torgersen           36.7          19.3               193
 6 Adelie  Torgersen           39.3          20.6               190
 7 Adelie  Torgersen           38.9          17.8               181
 8 Adelie  Torgersen           39.2          19.6               195
 9 Adelie  Torgersen           34.1          18.1               193
10 Adelie  Torgersen           42            20.2               190
# ℹ 334 more rows
```

Your turn

1- Load tidyverse

2- load “palmerpenguins” dataset

3- call the data

- type penguins

02:15

The PIPE

more about the package magrittr

- "Take what is on the left and use it as the first argument on what comes next

```
1 penguins %>% # take the opbject penguins
2   select(1:3) # then, select the columns 1 to 3
```

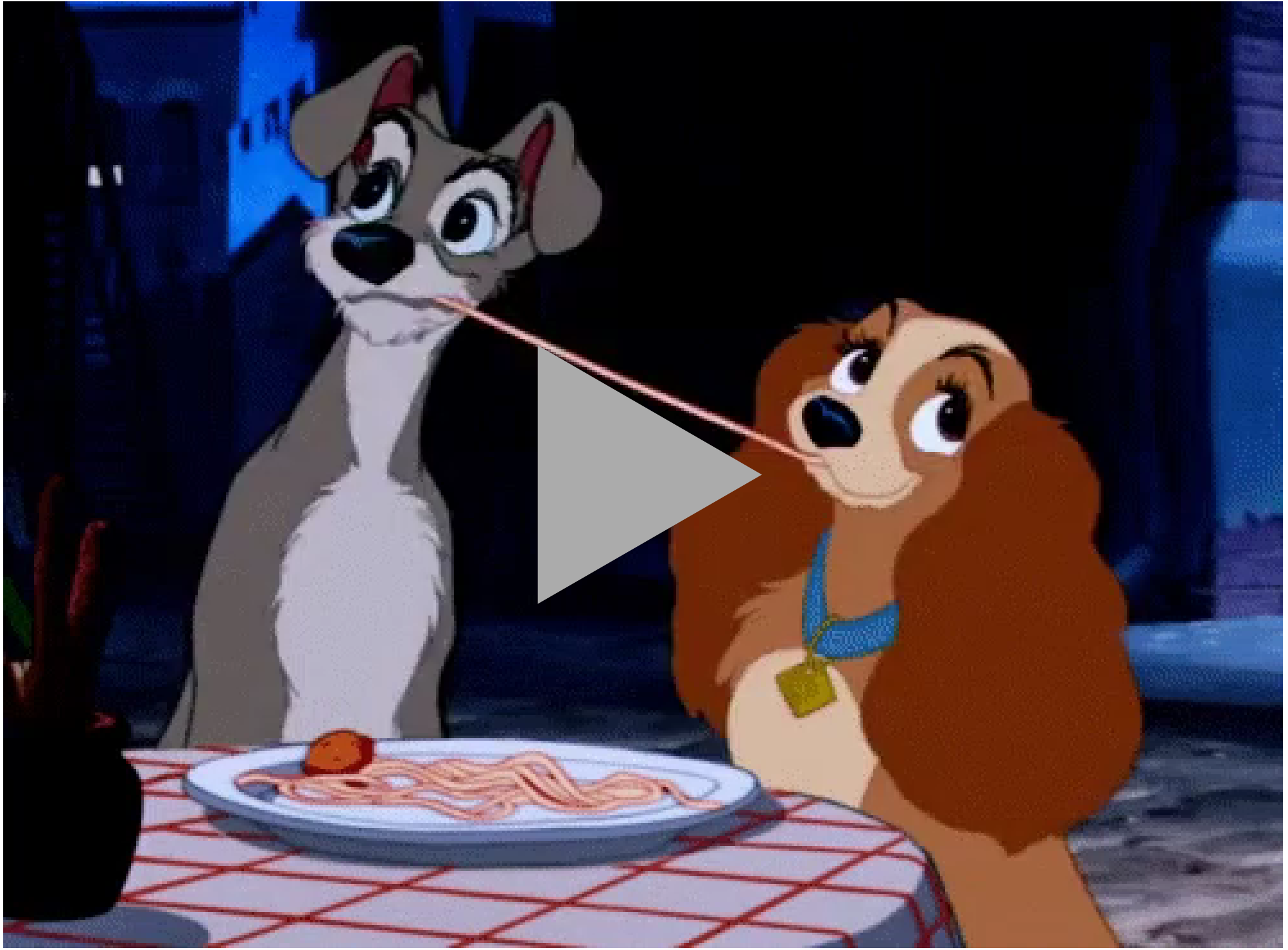
```
# A tibble: 344 × 3
   species island    bill_length_mm
   <fct>   <fct>              <dbl>
 1 Adelie  Torgersen           39.1
 2 Adelie  Torgersen           39.5
 3 Adelie  Torgersen           40.3
 4 Adelie  Torgersen           NA
 5 Adelie  Torgersen           36.7
 6 Adelie  Torgersen           39.3
 7 Adelie  Torgersen           38.9
 8 Adelie  Torgersen           39.2
 9 Adelie  Torgersen           34.1
10 Adelie  Torgersen           42
# ℹ 334 more rows
```

Why data wrangling

- your data is NEVER ready to analyse
- you need to get to know your data
- do some inspections
- ask some questions

"If you don't reveal some insights soon, I'm going to be forced to slice, dice, and drill!"


Drawing conclusions
Me
Statist



Data exploration

```r
penguins %>% # take the opbject penguins
  write.csv(., "penguins.csv") # then, save as .csv
```

```r
penguins %>%
  summary()
```

```
     species          island    bill_length_mm  bill_depth_mm
Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
                                Mean   :43.92   Mean   :17.15
                                3rd Qu.:48.50   3rd Qu.:18.70
                                Max.   :59.60   Max.   :21.50
                                NA's   :2       NA's   :2
flipper_length_mm  body_mass_g       sex           year
Min.   :172.0     Min.   :2700   female:165   Min.   :2007
1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
Median :197.0     Median :4050   NA's  : 11   Median :2008
Mean   :200.9     Mean   :4202                Mean   :2008
3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
Max.   :231.0     Max.   :6300                Max.   :2009
NA's   :2         NA's   :2
```

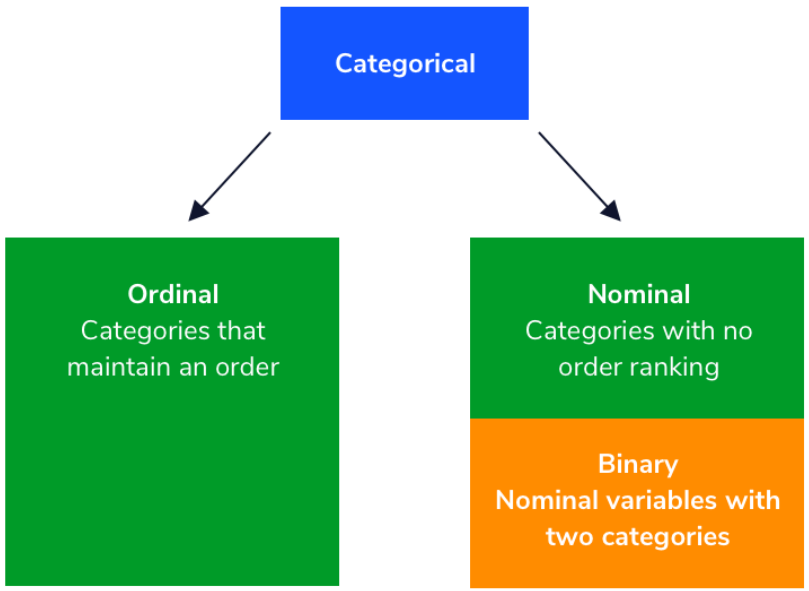
Types of variable

```
penguins %>%
  str()
```

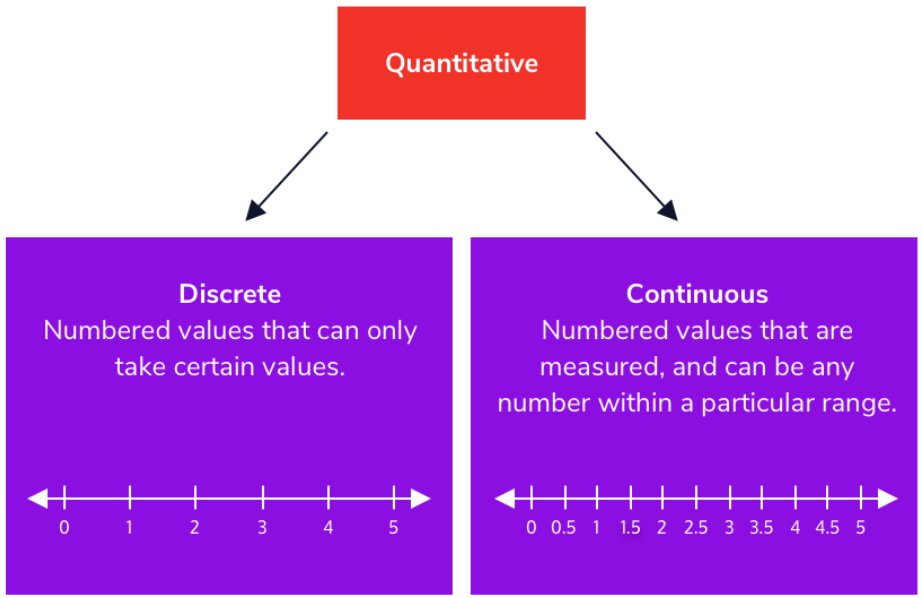
```
tibble [344 × 8] (S3: tbl_df/tbl/data.frame)
 $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1
1 1 1 1 ...
 $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3
3 3 ...
 $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1
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
```

Types of variables

Categorical



Numerical



Summarising data

```
library(vtable)
library(gt)

penguins %>%
  vtable(., lush = TRUE)
```

Summarising data

.

Name	Class	Values	Missing	Sumn
species	factor	'Adelie' 'Chinstrap' 'Gentoo'	0	nuniq
island	factor	'Biscoe' 'Dream' 'Torgersen'	0	nuniq
bill_length_mm	numeric	Num: 32.1 to 59.6	2	mean 43.92 sd: 5.4

Name	Class	Values	Missing	Summ
				nuniq 164
bill_depth_mm	numeric	Num: 13.1 to 21.5	2	mean 17.15 sd: 1.9 nuniq
flipper_length_mm	integer	Num: 172 to 231	2	mean 200.9 sd: 14.06 nuniq

Name	Class	Values	Missing	Summ
body_mass_g	integer	Num: 2700 to 6300	2	mean 4201.7 sd: 801.95 nuniq
sex	factor	'female' 'male'	11	nuniq
year	integer	Num: 2007 to 2009	0	mean 2008.0 sd: 0.8 nuniq

Summarising data

```
1 library(vtable)
2 library(gt)
3
4 penguins %>%
5   group_by(species) %>%
6   na.omit() %>%
7   summarise(mean = mean(bill_lengt
```

```
# A tibble: 3 × 4
  species    mean    sd      n
  <fct>     <dbl> <dbl> <int>
1 Adelie     38.8  2.66   146
2 Chinstrap  48.8  3.34    68
3 Gentoo     47.6  3.11   119
```

More codes here

Your turn

- Try to reproduce
- Create any summary for “penguins”

02:15

This goes on and on…

- Data exploration goes as far and deep as you need
- There is no minimum nor maximum
- The key point is

This needs to make your data make sense to you

Subset data

```r
penguins %>%
  select(body_mass_g)
```

```
# A tibble: 344 × 1
   body_mass_g
         <int>
 1        3750
 2        3800
 3        3250
 4          NA
 5        3450
 6        3650
 7        3625
 8        4675
 9        3475
10        4250
# ℹ 334 more rows
```

Subset data

```
penguins %>%
  filter(species=="Gentoo",
         bill_length_mm > 50,
         sex=="male") %>%
  select(bill_length_mm,
         bill_depth_mm) %>%
  arrange(bill_depth_mm)
```

```
# A tibble: 21 × 2
   bill_length_mm bill_depth_mm
            <dbl>         <dbl>
 1           51.3          14.2
 2           50.2          14.3
 3           50.1          15
 4           50.7          15
 5           50.4          15.3
 6           52.5          15.6
 7           54.3          15.7
 8           50.8          15.7
 9           50.4          15.7
10           53.4          15.8
# ℹ 11 more rows
```

Add new columns

```r
penguins %>%
  select(bill_length_mm,
         bill_depth_mm) %>%
  mutate(bill_volume=bill_length_mm+bill_depth_mm) %>%
  mutate(log_bill_volume=log(bill_volume)) %>%
  mutate(bill_categ=ifelse(bill_volume<60, "small", "big"))
```

```
# A tibble: 344 × 5
   bill_length_mm bill_depth_mm bill_volume log_bill_volume bill_categ
            <dbl>         <dbl>       <dbl>           <dbl> <chr>
 1           39.1          18.7        57.8            4.06 small
 2           39.5          17.4        56.9            4.04 small
 3           40.3          18          58.3            4.07 small
 4           NA            NA          NA             NA    <NA>
 5           36.7          19.3        56              4.03 small
 6           39.3          20.6        59.9            4.09 small
 7           38.9          17.8        56.7            4.04 small
 8           39.2          19.6        58.8            4.07 small
 9           34.1          18.1        52.2            3.96 small
10           42            20.2        62.2            4.13 big
# ℹ 334 more rows
```

Reshape data Tidyr

Long format

```
penguins %>%
  select(bill_length_mm,
         bill_depth_mm,
         year) %>%
  pivot_longer(col=c(bill_length_mm:bill_depth_mm),
               names_to = "bill_feature", values_to = "value")
```

```
# A tibble: 688 × 3
    year bill_feature   value
   <int> <chr>          <dbl>
 1  2007 bill_length_mm  39.1
 2  2007 bill_depth_mm   18.7
 3  2007 bill_length_mm  39.5
 4  2007 bill_depth_mm   17.4
 5  2007 bill_length_mm  40.3
 6  2007 bill_depth_mm   18
 7  2007 bill_length_mm  NA
 8  2007 bill_depth_mm   NA
 9  2007 bill_length_mm  36.7
10  2007 bill_depth_mm   19.3
# ℹ 678 more rows
```

Reshape data Tidyr

Wide format

```
penguins %>%
  mutate(row = row_number()) %>% # needed to add a rwoo number to identify
  select(row, species, island, body_mass_g) %>%
  pivot_wider(names_from = island, values_from = body_mass_g)
```

```
# A tibble: 344 × 5
     row species Torgersen Biscoe Dream
   <int> <fct>       <int>  <int> <int>
 1     1 Adelie       3750     NA    NA
 2     2 Adelie       3800     NA    NA
 3     3 Adelie       3250     NA    NA
 4     4 Adelie         NA     NA    NA
 5     5 Adelie       3450     NA    NA
 6     6 Adelie       3650     NA    NA
 7     7 Adelie       3625     NA    NA
 8     8 Adelie       4675     NA    NA
 9     9 Adelie       3475     NA    NA
10    10 Adelie       4250     NA    NA
# ℹ 334 more rows
```

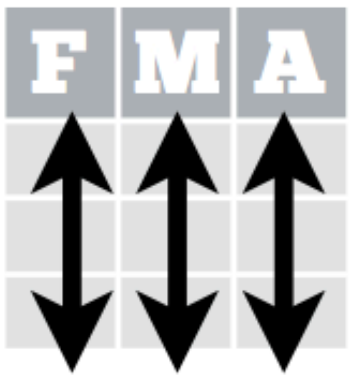
Your turn

- Try to reproduce
- Create any wide and long formats for “penguins”

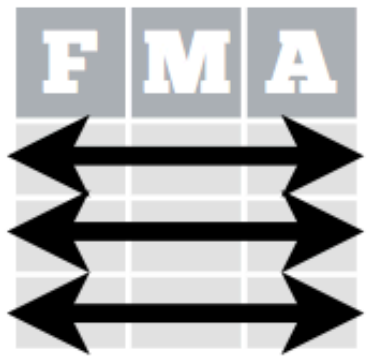
03:15

The correct data format

In a tidy data set:



&



Each **variable** is saved in its own **column**

Each **observation** is saved in its own **row**

End of session on DA