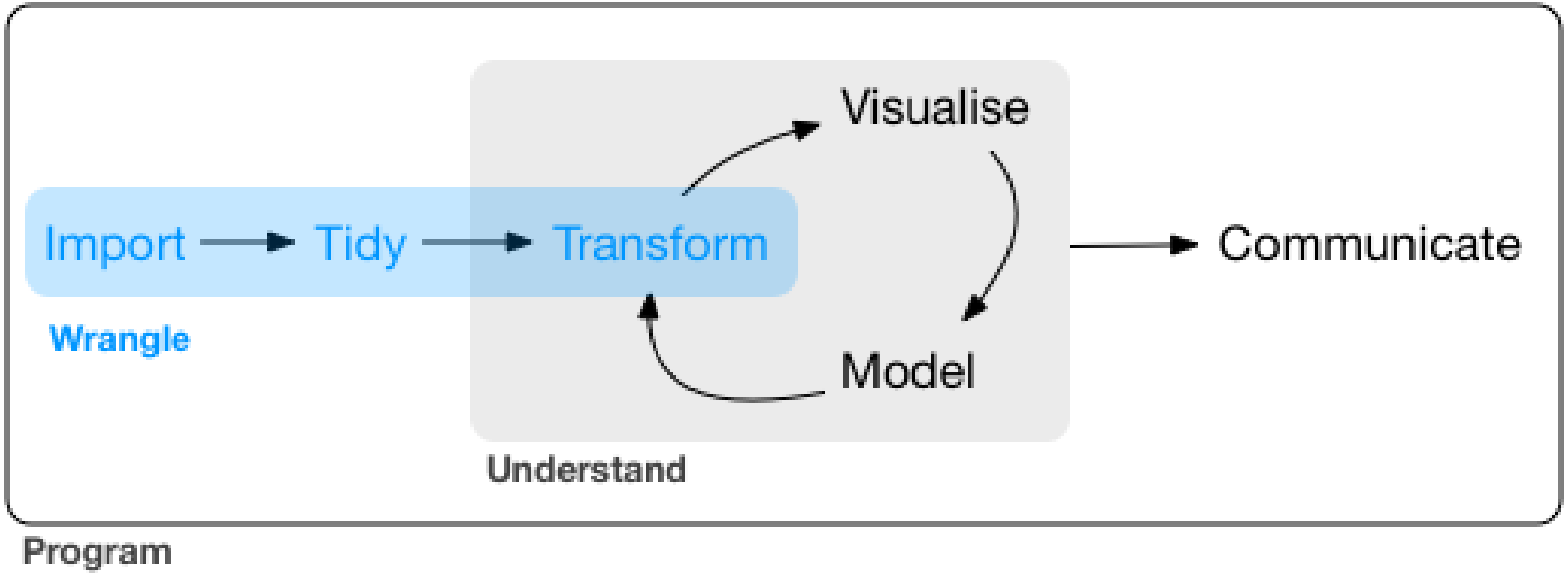
Data Wrangling

TOPIC - 2 | AJAY KOLI

2020-10-20 (updated on 2021-02-11)

Course Progress



🎯 Objectives:

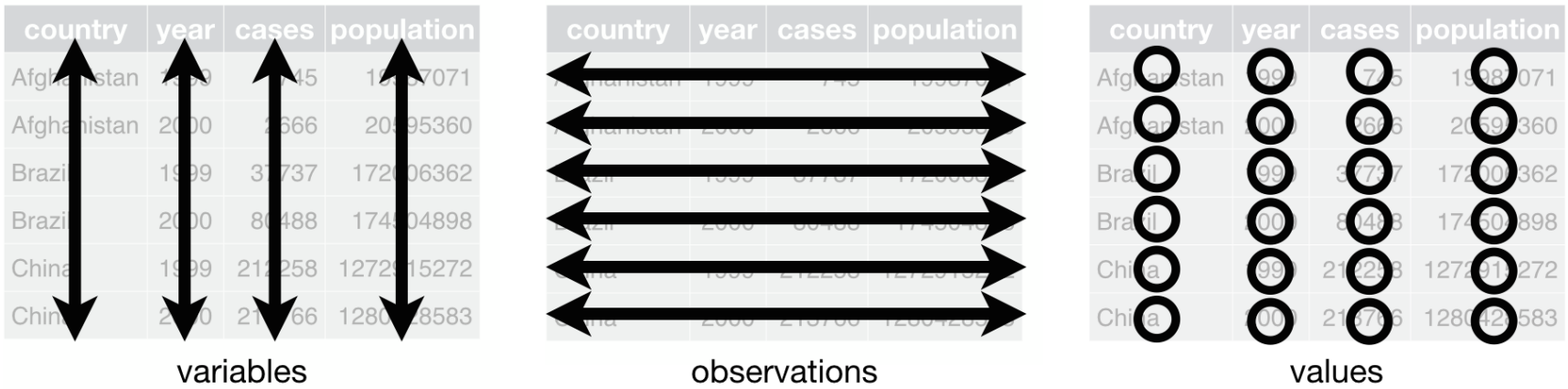
1. To import & export data
2. To perform data manipulation for rows
3. To perform data manipulation for columns
4. To calculate summary statistics for the variables

What is Data wrangling?

- "data exploration and data manipulation" (Jesse Mostipak)
- "tidying and transforming" (Hadley & Garrett)

"Tidying" data means:

- "each column is a variable, and each row is an observation" (Hadley & Garrett)



"Transforming" data means:

- "narrowing in on observations of interest ...
- creating **new variables** that are functions of existing variables ... and
- calculating a set of **summary statistics.**" (Hadley & Garrett)

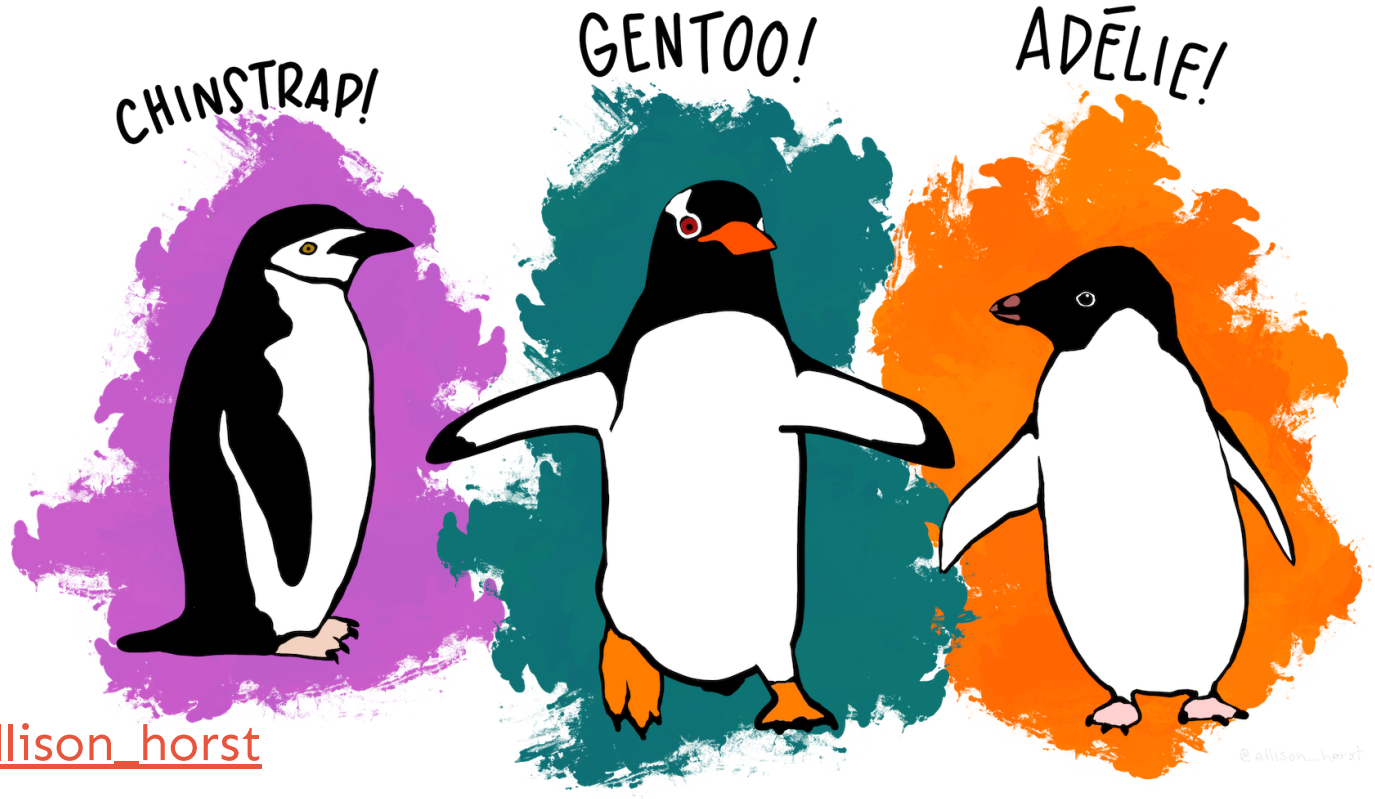
Data

Palmer penguins data

Palmer Staion, Antarctica LTER

Included variables are:

- species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex, year



Artwork by @allison_horst

Included variables are:

- species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex, year

Artwork by @allison_horst

readr package

`readr` pkg

- "to provide a fast and friendly way to **read rectangular** data (like csv, tsv, and fwf)".
- function is `read_csv()`

Source: tidyverse

Codes | Output

```r
# import data
library(readr)

penguins <- read_csv("data/penguins.csv")

penguins
```

Codes | Output

```
# A tibble: 344 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm b
   <chr>   <chr>              <dbl>         <dbl>             <dbl>
 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
# … with 334 more rows
```

Codes | Output

```
summary(penguins)
```

Codes | Output

```
  species          island      bill_length_mm  bill_depth_mm  
Length:344         Length:344   Min.   :32.10   Min.   :13.10  
Class :character   Class :character   1st Qu.:39.23   1st Qu.:15.60  
Mode  :character   Mode  :character   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      
 body_mass_g       sex                 year     
Min.   :2700   Length:344         Min.   :2007  
1st Qu.:3550   Class :character   1st Qu.:2007  
Median :4050   Mode  :character   Median :2008  
Mean   :4202                      Mean   :2008  
3rd Qu.:4750                      3rd Qu.:2009  
Max.   :6300                      Max.   :2009  
NA's   :2                                       
```

Variable types in R:

- `int` stands for integers, like 4, 55, 300.
- `dbl` stands for doubles, or real numbers like 3, 7.45, 1.565, 12.
- `chr` stands for character vectors, or strings like names.
- `dttm` stands for date-times (a date + a time).
- `lgl` stands for logical, vectors that contain only TRUE or FALSE.
- `fct` stands for factors, which R uses to represent **categorical variables** with fixed possible values like occupation: student, professional, government, business.
- `date` stands for dates.

Change variable types in penguin data:

1. `species` : a **factor** denoting penguin species (Adélie, Chinstrap and Gentoo)
2. `island` : a **factor** denoting island in Palmer Archipelago, Antarctica (Biscoe, Dream or Torgersen)
3. `bill_length_mm` : a **number** denoting bill length (millimeters)
4. `bill_depth_mm` : a **number** denoting bill depth (millimeters)
5. `flipper_length_mm` : an **integer** denoting flipper length (millimeters)
6. `body_mass_g` : an **integer** denoting body mass (grams)
7. `sex` : a **factor** denoting penguin sex (female, male)
8. `year` : an **integer** denoting the study year (2007, 2008, or 2009)

Codes | Output

```
penguins <- read_csv("data/penguins.csv",
  col_types = cols(
    species = col_factor(),
    island = col_factor(),
    bill_length_mm = col_double(),
    bill_depth_mm = col_double(),
    flipper_length_mm = col_integer(),
    body_mass_g = col_integer(),
    sex = col_factor(),
    year = col_integer()
  )
)

summary(penguins)
```

Codes | Output

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

dplyr package

`dplyr` pkg

- "dplyr is a grammar of data manipulation"
- "providing a consistent set of verbs that help you solve the most common data manipulation challenges:"
  - `filter()` picks cases based on their values.
  - `select()` picks variables based on their names.
  - `mutate()` adds new variables that are functions of existing variables
  - `arrange()` changes the ordering of the rows.
  - `summarise()` reduces multiple values down to a single summary.

Source: tidyverse

To get a glimpse 👀 of your data

Codes | Output

```
library(dplyr)

glimpse(penguins)
```

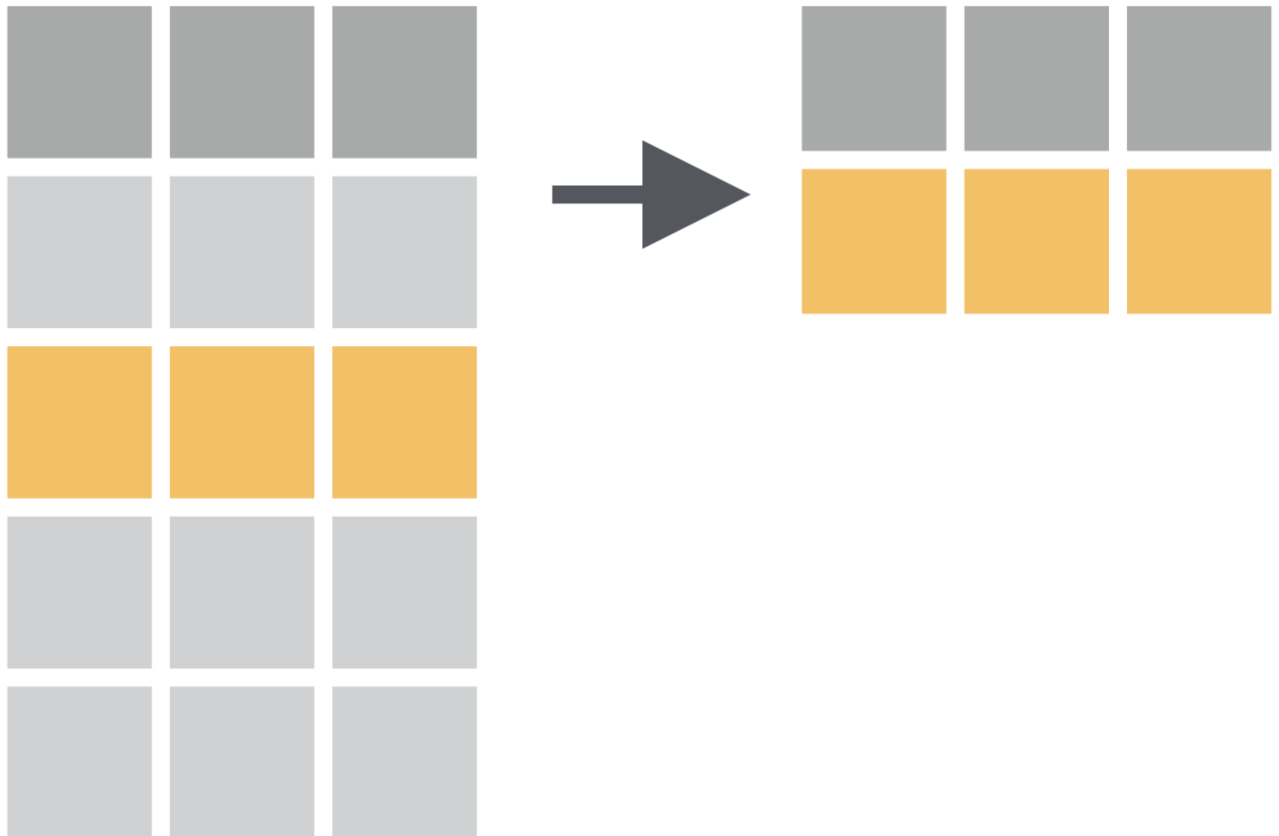
To get a glimpse 👀 of your data

Codes | Output

```
Rows: 344
Columns: 8
$ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Ade
$ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen,
$ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39
$ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19
$ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193,
$ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 467
$ sex               <fct> male, female, female, NA, female, male, fem
$ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2
```

`filter()` function

`filter()` function: Picks cases based on their values.



How to have a data of only Gentoo penguins?

Codes | Output

```r
# three species are Chinstrap, Gentoo, Adelie
penguins %>%
  filter(species == "Gentoo")
```

Codes Output

```
# A tibble: 124 x 8
   species island bill_length_mm bill_depth_mm flipper_length_mm body
   <fct>   <fct>           <dbl>         <dbl>             <int>
 1 Gentoo  Biscoe           46.1          13.2               211
 2 Gentoo  Biscoe           50            16.3               230
 3 Gentoo  Biscoe           48.7          14.1               210
 4 Gentoo  Biscoe           50            15.2               218
 5 Gentoo  Biscoe           47.6          14.5               215
 6 Gentoo  Biscoe           46.5          13.5               210
 7 Gentoo  Biscoe           45.4          14.6               211
 8 Gentoo  Biscoe           46.7          15.3               219
 9 Gentoo  Biscoe           43.3          13.4               209
10 Gentoo  Biscoe           46.8          15.4               215
# … with 114 more rows
```

How to import data file to your computer?

Codes | Output

```
# three species are Chinstrap, Gentoo, Adelie
penguins %>%
  filter(species == "Gentoo") %>%
  write_csv("data/gentoo-penguins.csv")
```

Codes Output

But wait 👮 what the hell is %>%

- this is called **pipe** (%>% = control + shift + m)
- "a powerful tool for clearly expressing a sequence of **multiple operations**"
- interpret/read it as **then**.

```
penguins %>%
  filter(species == "Gentoo")
```

Comparison: Relational Operators

`x < y`

`x > y`

`x <= y`

`x >= y`

`x == y` (equal)

`x != y` (not equal)

How to have a data of penguins with bill length more than 43 mm?

Codes | Output

```
penguins %>%
  filter(bill_length_mm > 43)
```

Codes | Output

```
# A tibble: 188 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm b
   <fct>   <fct>              <dbl>         <dbl>             <int>
 1 Adelie  Torgersen           46            21.5               194
 2 Adelie  Dream              44.1            19.7               196
 3 Adelie  Torgersen          45.8            18.9               197
 4 Adelie  Dream              43.2            18.5               192
 5 Adelie  Biscoe             43.2            19                 197
 6 Adelie  Biscoe             45.6            20.3               191
 7 Adelie  Torgersen          44.1            18                 210
 8 Adelie  Torgersen          43.1            19.2               197
 9 Gentoo  Biscoe             46.1            13.2               211
10 Gentoo  Biscoe             50              16.3               230
# … with 178 more rows
```

How to have a data of Gentoo penguins with bill length more than 50 mm?

Codes | Output

```
penguins %>%
  filter(species == "Gentoo",
         bill_length_mm > 50)
```

Codes Output

```
# A tibble: 22 x 8
   species island bill_length_mm bill_depth_mm flipper_length_mm body
   <fct>   <fct>           <dbl>         <dbl>             <int>
 1 Gentoo  Biscoe           50.2          14.3               218
 2 Gentoo  Biscoe           59.6          17                 230
 3 Gentoo  Biscoe           50.5          15.9               222
 4 Gentoo  Biscoe           50.5          15.9               225
 5 Gentoo  Biscoe           50.1          15                 225
 6 Gentoo  Biscoe           50.4          15.3               224
 7 Gentoo  Biscoe           54.3          15.7               231
 8 Gentoo  Biscoe           50.7          15                 223
 9 Gentoo  Biscoe           51.1          16.3               220
10 Gentoo  Biscoe           52.5          15.6               221
# … with 12 more rows
```

How to have data of non-Gentoo penguins with bill length more than 45 mm and weight more than 4 kg?

Codes | Output

```
penguins %>%
  filter(species != "Gentoo",
         bill_length_mm > 45,
         body_mass_g > 4000)
```

Codes Output

```
# A tibble: 18 x 8
   species   island   bill_length_mm bill_depth_mm flipper_length_… b
   <fct>     <fct>             <dbl>         <dbl>             <int>
 1 Adelie    Torgers…           46            21.5               194
 2 Adelie    Torgers…           45.8          18.9               197
 3 Adelie    Biscoe             45.6          20.3               191
 4 Chinstrap Dream              46            18.9               195
 5 Chinstrap Dream              52            18.1               201
 6 Chinstrap Dream              50.5          19.6               201
 7 Chinstrap Dream              49.2          18.2               195
 8 Chinstrap Dream              52            19                 197
 9 Chinstrap Dream              52.8          20                 205
10 Chinstrap Dream              54.2          20.8               201
11 Chinstrap Dream              51            18.8               203
12 Chinstrap Dream              52            20.7               210
13 Chinstrap Dream              53.5          19.9               205
14 Chinstrap Dream              50.8          18.5               201
15 Chinstrap Dream              49            19.6               212
```

How to have only top or bottom rows from data?

Codes | Output

```
penguins %>%
  filter(species != "Gentoo",
         bill_length_mm > 45,
         body_mass_g > 4000) %>%
  head(3)
```

Codes Output

```
# A tibble: 3 x 8
  species island    bill_length_mm bill_depth_mm flipper_length_mm bo
  <fct>   <fct>              <dbl>         <dbl>             <int>
1 Adelie  Torgersen           46            21.5               194
2 Adelie  Torgersen           45.8          18.9               197
3 Adelie  Biscoe              45.6          20.3               191
```

Codes | Output

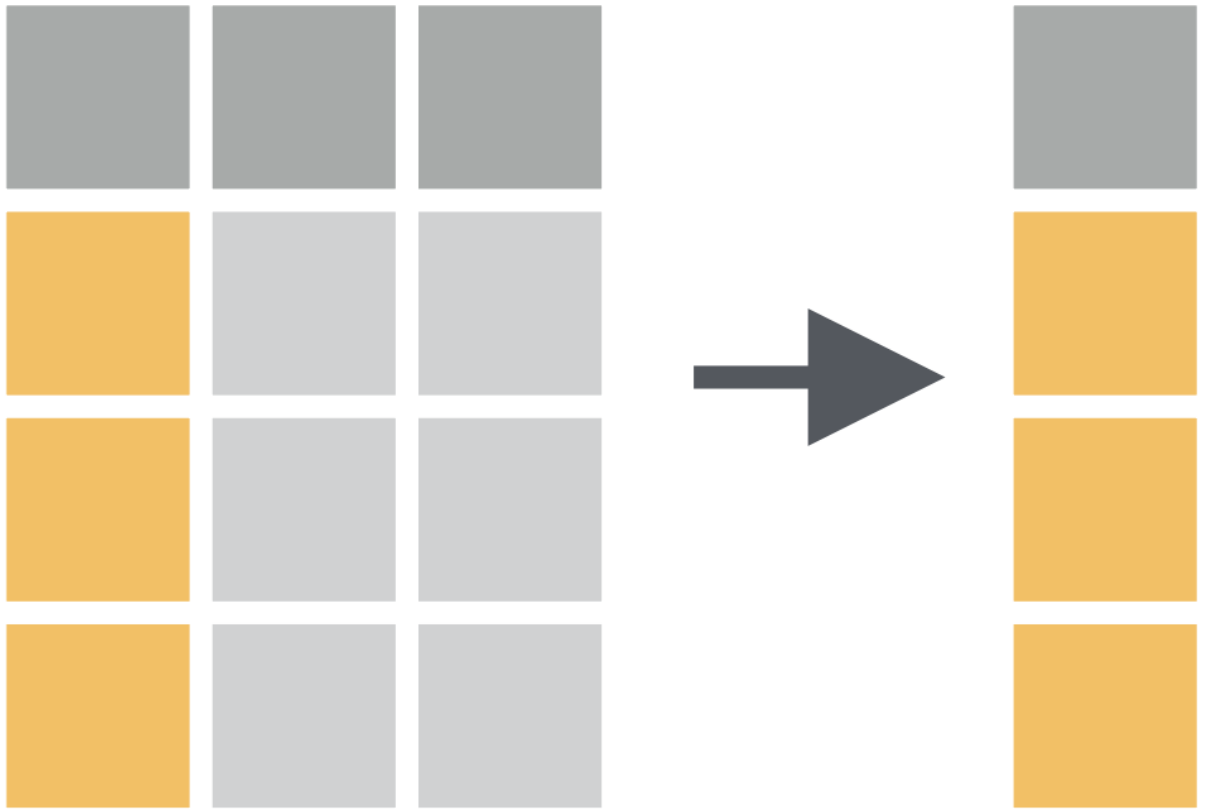
```
penguins %>%
  filter(species != "Gentoo",
         bill_length_mm > 45,
         body_mass_g > 4000) %>%
  tail()
```

Codes Output

```
# A tibble: 6 x 8
  species   island bill_length_mm bill_depth_mm flipper_length_mm boc
  <fct>     <fct>           <dbl>         <dbl>             <int>
1 Chinstrap Dream            53.5          19.9               205
2 Chinstrap Dream            50.8          18.5               201
3 Chinstrap Dream            49            19.6               212
4 Chinstrap Dream            50.7          19.7               203
5 Chinstrap Dream            49.3          19.9               203
6 Chinstrap Dream            50.8          19                 210
```

`select()` function

`select()` function: Chooses rows based on column values.



How to have only `species` variable in data?

Codes | Output

```
penguins %>%
  select(species)
```

Codes Output

```
# A tibble: 344 x 1
   species
   <fct>
 1 Adelie
 2 Adelie
 3 Adelie
 4 Adelie
 5 Adelie
 6 Adelie
 7 Adelie
 8 Adelie
 9 Adelie
10 Adelie
# … with 334 more rows
```

How to have a specific range of variables in data?

Codes | Output

```
penguins %>%
  select(species : bill_depth_mm)
```

Codes Output

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

How to have variables based upon their location in data?

Codes | Output

```
penguins %>%
  select(4:8)
```

Codes | Output

```
# A tibble: 344 x 5
   bill_depth_mm flipper_length_mm body_mass_g sex      year
           <dbl>             <int>       <int> <fct>   <int>
 1          18.7               181        3750 male     2007
 2          17.4               186        3800 female   2007
 3          18                 195        3250 female   2007
 4          NA                  NA          NA <NA>     2007
 5          19.3               193        3450 female   2007
 6          20.6               190        3650 male     2007
 7          17.8               181        3625 female   2007
 8          19.6               195        4675 male     2007
 9          18.1               193        3475 <NA>     2007
10          20.2               190        4250 <NA>     2007
# … with 334 more rows
```

How to have specific variables in data?

Codes | Output

```
penguins %>%
  select(species, body_mass_g, year)
```

Codes | Output

```
# A tibble: 344 x 3
   species body_mass_g  year
   <fct>          <int> <int>
 1 Adelie          3750  2007
 2 Adelie          3800  2007
 3 Adelie          3250  2007
 4 Adelie            NA  2007
 5 Adelie          3450  2007
 6 Adelie          3650  2007
 7 Adelie          3625  2007
 8 Adelie          4675  2007
 9 Adelie          3475  2007
10 Adelie          4250  2007
# … with 334 more rows
```

Codes | Output

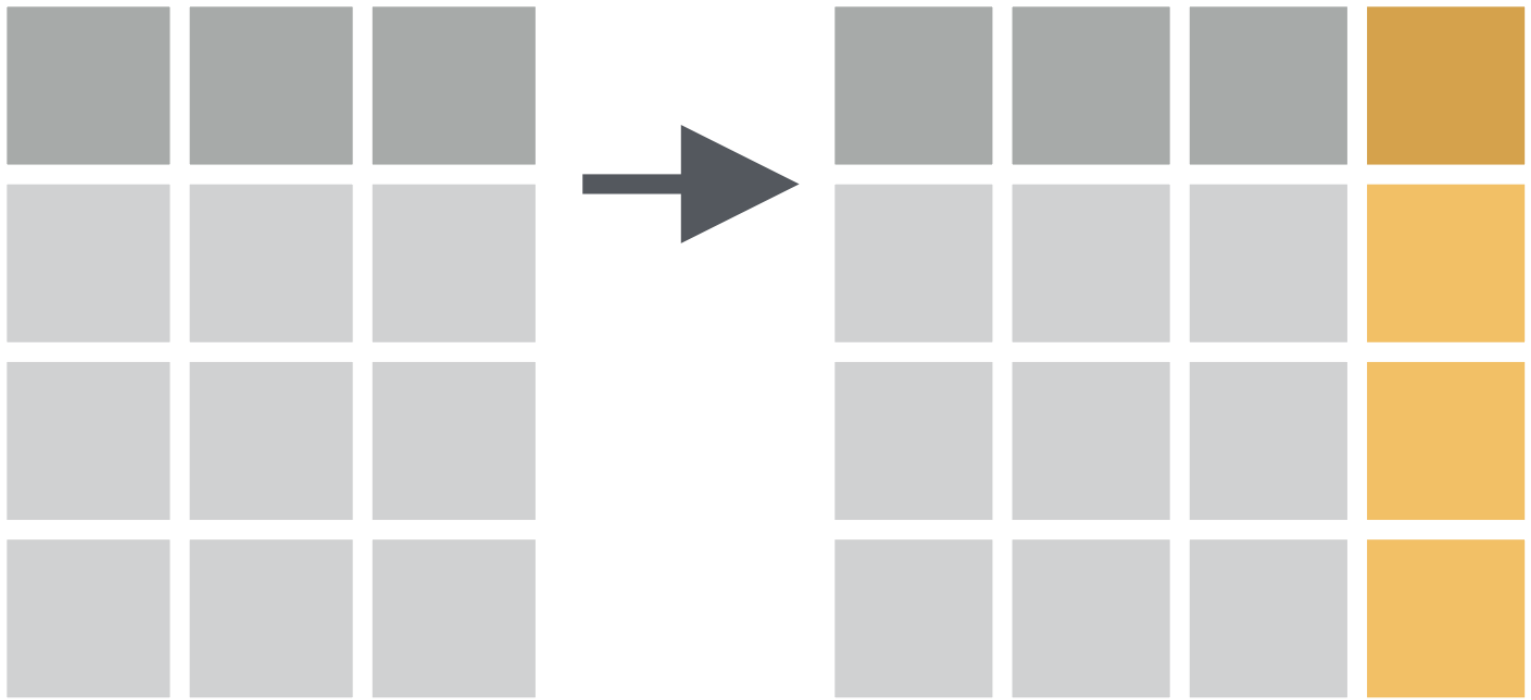
```
penguins %>%
  select(-c(species, body_mass_g, year))
```

Codes Output

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

`mutate()` function

`mutate()` function: Adds new variables that are functions of existing variables



How to convert penguin body mass from grams to kilograms?

Codes | Output

```
penguins %>%
  mutate(body_mass_g = body_mass_g / 1000)
```

Codes | Output

```
# A tibble: 344 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm b
   <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
# … with 334 more rows
```

Codes | Output

```
penguins %>%
  mutate(body_mass_g = body_mass_g / 1000,
         bill = bill_length_mm * bill_depth_mm)
```

Codes Output

```
# A tibble: 344 x 9
   species island bill_length_mm bill_depth_mm flipper_length_… body_
   <fct>   <fct>           <dbl>         <dbl>             <int>
 1 Adelie  Torge…           39.1          18.7               181
 2 Adelie  Torge…           39.5          17.4               186
 3 Adelie  Torge…           40.3          18                 195
 4 Adelie  Torge…           NA            NA                  NA
 5 Adelie  Torge…           36.7          19.3               193
 6 Adelie  Torge…           39.3          20.6               190
 7 Adelie  Torge…           38.9          17.8               181
 8 Adelie  Torge…           39.2          19.6               195
 9 Adelie  Torge…           34.1          18.1               193
10 Adelie  Torge…           42            20.2               190
# … with 334 more rows, and 1 more variable: bill <dbl>
```

Codes | Output

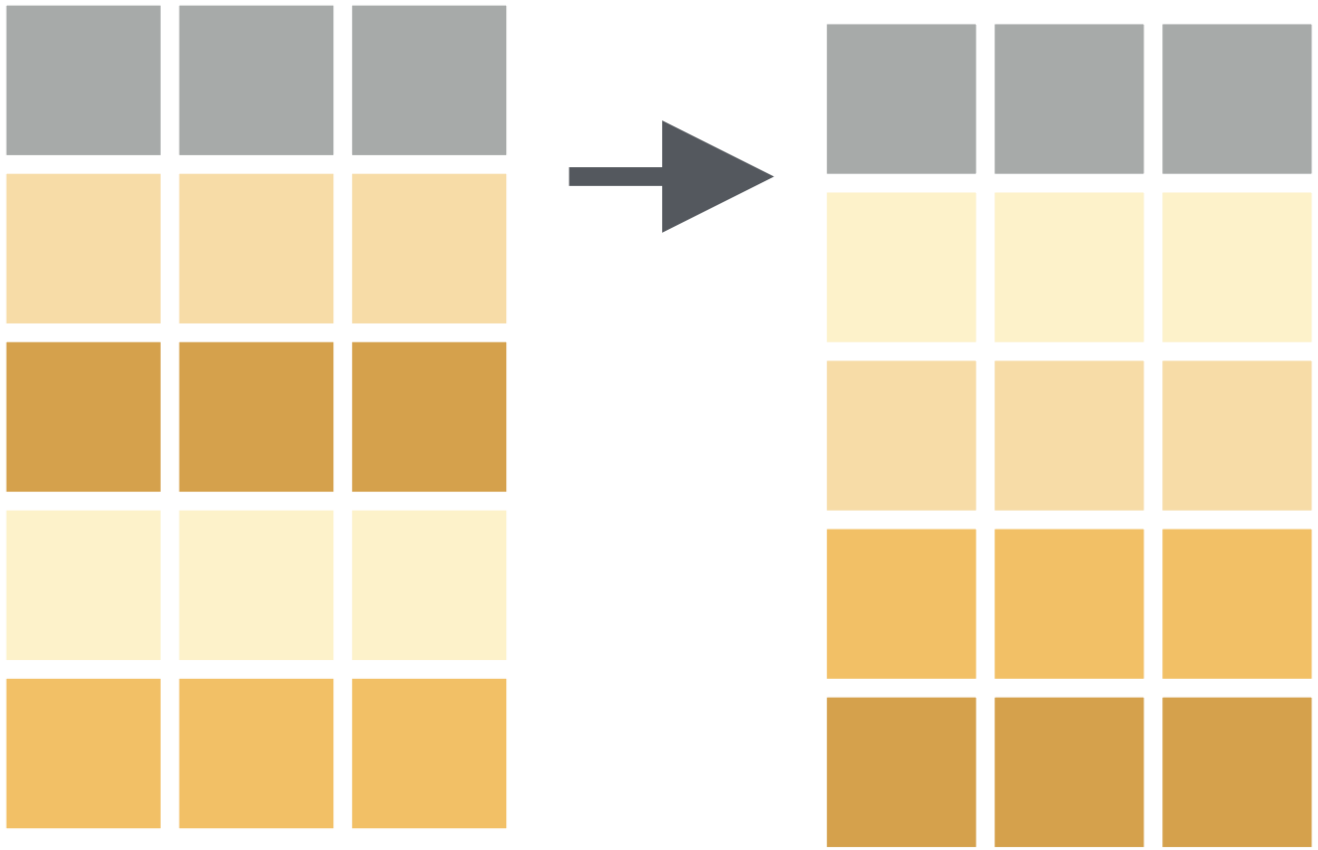
```
penguins %>%
  mutate(body_mass_g = body_mass_g / 1000,
         bill = bill_length_mm * bill_depth_mm) %>%
  select(body_mass_g,
         bill)
```

Codes | Output

```
# A tibble: 344 x 2
   body_mass_g  bill
         <dbl> <dbl>
 1        3.75  731.
 2        3.8   687.
 3        3.25  725.
 4       NA      NA
 5        3.45  708.
 6        3.65  810.
 7        3.62  692.
 8        4.68  768.
 9        3.48  617.
10        4.25  848.
# … with 334 more rows
```

`arrange()` function

`arrange()` function: Changes the order of the rows.



How to have data arranged by the ascending order of bill length of penguins?

Codes | Output

```
penguins %>%
  arrange(bill_length_mm)
```

Codes | Output

```
# A tibble: 344 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm b
   <fct>   <fct>              <dbl>         <dbl>             <int>
 1 Adelie  Dream               32.1          15.5               188
 2 Adelie  Dream               33.1          16.1               178
 3 Adelie  Torgersen           33.5          19                 190
 4 Adelie  Dream               34            17.1               185
 5 Adelie  Torgersen           34.1          18.1               193
 6 Adelie  Torgersen           34.4          18.4               184
 7 Adelie  Biscoe              34.5          18.1               187
 8 Adelie  Torgersen           34.6          21.1               198
 9 Adelie  Torgersen           34.6          17.2               189
10 Adelie  Biscoe              35            17.9               190
# … with 334 more rows
```

Codes | Output

```
penguins %>%
  arrange(desc(bill_length_mm))
```

Codes Output

```
# A tibble: 344 x 8
   species   island bill_length_mm bill_depth_mm flipper_length_mm bo
   <fct>     <fct>           <dbl>         <dbl>             <int>
 1 Gentoo    Biscoe           59.6          17                 230
 2 Chinstrap Dream            58            17.8               181
 3 Gentoo    Biscoe           55.9          17                 228
 4 Chinstrap Dream            55.8          19.8               207
 5 Gentoo    Biscoe           55.1          16                 230
 6 Gentoo    Biscoe           54.3          15.7               231
 7 Chinstrap Dream            54.2          20.8               201
 8 Chinstrap Dream            53.5          19.9               205
 9 Gentoo    Biscoe           53.4          15.8               219
10 Chinstrap Dream            52.8          20                 205
# … with 334 more rows
```

Codes | Output

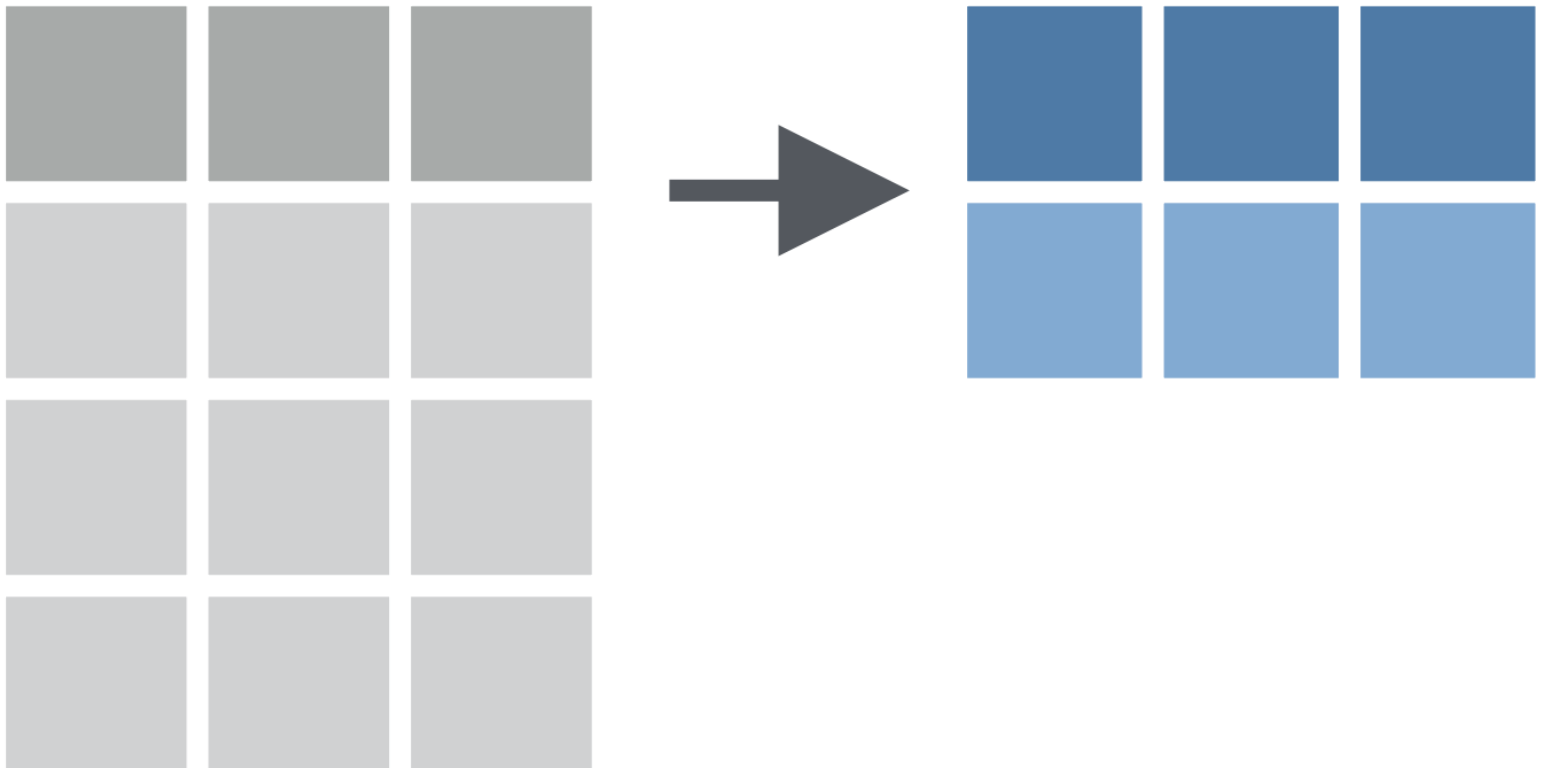
```
penguins %>%
  arrange(species)
```

Codes | Output

```
# A tibble: 344 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm b
   <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
# … with 334 more rows
```

`summarise()` function

`summarise()` function: Chooses rows based on column values.



How to find mean bill length of all penguins?

Codes | Output

```
penguins %>%
  drop_na() %>%
  summarise(mean = mean(bill_length_mm))
```

Codes | Output

```
# A tibble: 1 x 1
   mean
  <dbl>
1  44.0
```

How to find species-wise mean bill length of penguins?

Codes | Output

```
penguins %>%
  drop_na() %>%
  group_by(species) %>%
  summarise(mean = mean(bill_length_mm))
```

Codes Output

```
# A tibble: 3 x 2
  species    mean
  <fct>     <dbl>
1 Adelie     38.8
2 Gentoo     47.6
3 Chinstrap  48.8
```

How to find species-wise mean bill length of penguins and total number of penguins in each species?

Codes | Output

```
penguins %>%
  drop_na() %>%
  group_by(species) %>%
  summarise(mean = mean(bill_length_mm), n = n())
```

Codes Output

```
# A tibble: 3 x 3
  species    mean     n
  <fct>     <dbl> <int>
1 Adelie     38.8   146
2 Gentoo     47.6   119
3 Chinstrap  48.8    68
```

Useful functions

- Center: `mean()`, `median()`
- Spread: `sd()`, `IQR()`, `mad()`
- Range: `min()`, `max()`, `quantile()`
- Position: `first()`, `last()`, `nth()`,
- Count: `n()`, `n_distinct()`
- Logical: `any()`, `all()`

Source: tidyverse.org

`dplyr` functions based upon:

Rows | Columns | Groups of rows

- `filter()` chooses rows based on column values.
- `slice()` chooses rows based on location.
- `arrange()` changes the order of the rows.

`dplyr` functions based upon:

Rows | **Columns** | Groups of rows

- `select()` changes whether or not a column is included.
- `rename()` changes the name of columns.
- `mutate()` changes the values of columns and creates new columns.
- `relocate()` changes the order of the columns.

`dplyr` functions based upon:

Rows | Columns | Groups of rows

- `summarise()` collapses a group into a single row.

Thank you 😄