R Notebook

# First Session

**Understanding the data**

To design an effective visualization with a clear purpose, you must thoroughly understand your data. The following points will help you find the information you want to convey with your data:

* What kind of data is it? Nominal, ordinal, interval or ratio data?
* How do the different parts of the data relate to each other?
* Can you organize the data in a way that makes it easy for you to create your visualizations?
* What do you want to communicate with your data?

## The ggplot2 package

The grammar of graphics (ggplot)

To contruct a graph, we need

* data: a dataframe in tidyformat
* some mapping to include aesthetics elements
* a geometric representation of the aesthetics elements

The ggplot allows to manage those elements and adding to your graph as layers

First install pacman that will allow to install other packages

# install.packages("pacman")

And now I will install the packages that we will use

pacman::p\_load(tidyverse, # several datascience packages  
 palmerpenguins, # data  
 gapminder, #data,  
 gtsummary) # for tables)

We have everything ready

head(penguins)

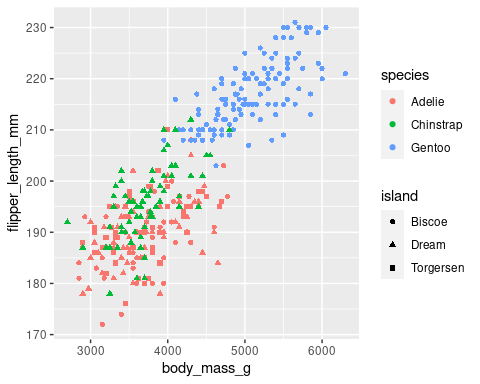
## # A tibble: 6 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_… body\_mass\_g sex   
## <fct> <fct> <dbl> <dbl> <int> <int> <fct>  
## 1 Adelie Torge… 39.1 18.7 181 3750 male   
## 2 Adelie Torge… 39.5 17.4 186 3800 fema…  
## 3 Adelie Torge… 40.3 18 195 3250 fema…  
## 4 Adelie Torge… NA NA NA NA <NA>   
## 5 Adelie Torge… 36.7 19.3 193 3450 fema…  
## 6 Adelie Torge… 39.3 20.6 190 3650 male   
## # … with 1 more variable: year <int>

View(penguins)

## My first graph

penguins %>%   
 ggplot(aes(x = body\_mass\_g,   
 y = flipper\_length\_mm,   
 color = species,   
 shape = island)) +  
 geom\_point()

## Warning: Removed 2 rows containing missing values (geom\_point).



covid\_latvia <- read\_csv("https://covid.ourworldindata.org/data/owid-covid-data.csv")

## Rows: 135581 Columns: 67

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): iso\_code, continent, location, tests\_units  
## dbl (62): total\_cases, new\_cases, new\_cases\_smoothed, total\_deaths, new\_dea...  
## date (1): date

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Second session

## Packages

Run the first lines

## Explore the dataset

### Know your variables

* continuous: num or int
* categorical: fct or chr
* date/time

### head

### str & glimpse

### summary

Summary of all the dataset

Summary of a specific variable (or vector in R vocabulary)

species

island

flipper\_length

### dim

Give the **dim**ensions of the dataset in rows and columns

## Summary tables

#### With tables

By sex

By sex and species

by species and island

#### With gtsummary

all

by sex

by species

by island

only sex, species and body\_mass\_g (new: select)

## Visualize the dataset

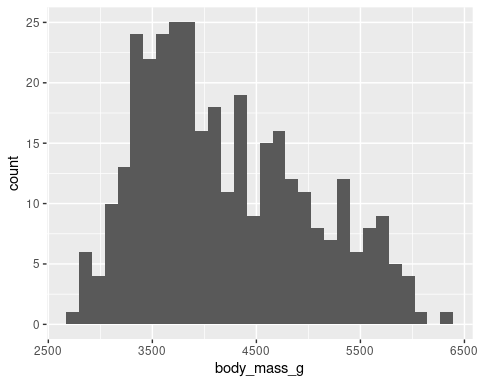
### One variable

#### continuous: num or int

penguins %>%   
 ggplot(aes(x = body\_mass\_g)) +   
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing non-finite values (stat\_bin).

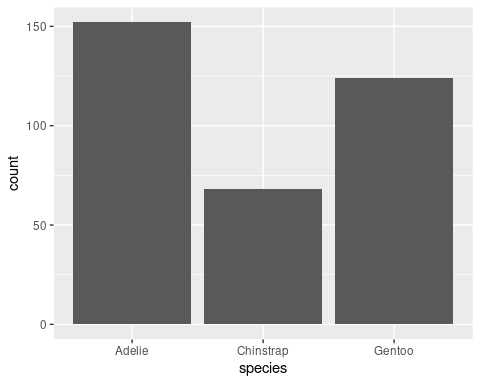


by flipper\_length

only one specie (new: filter)

#### categorical: fct or chr

penguins %>%   
 ggplot(aes(x = species)) +   
 geom\_bar()



By island

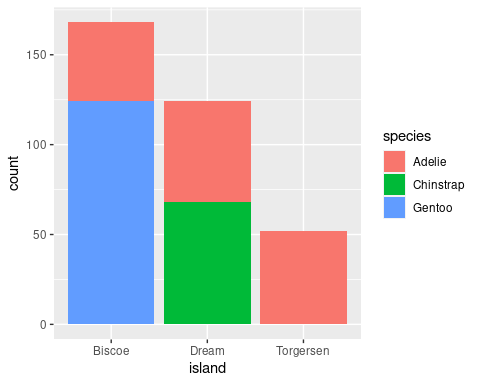
### Two variables

#### number vs categorical

species vs body\_mass\_g

#### categorical vs categorical

penguins %>%   
 ggplot(aes(x = island,   
 fill = species)) +   
 geom\_bar()



### Time

See <https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations>

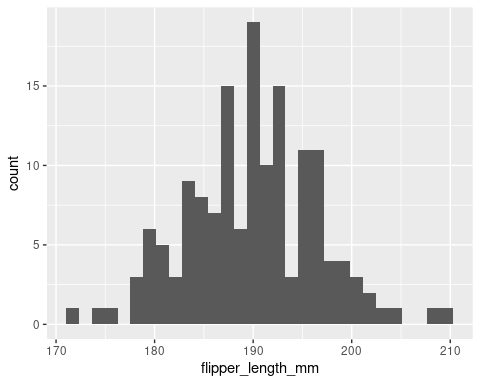
## Second session last part, delete this header afterwards

### One variable

penguins %>%   
 filter(species == "Adelie") %>%   
 ggplot(aes(x = flipper\_length\_mm)) +   
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat\_bin).



penguins %>%   
 filter(species == "Adelie") %>%   
 ggplot(aes(x = body\_mass\_g)) +   
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat\_bin).

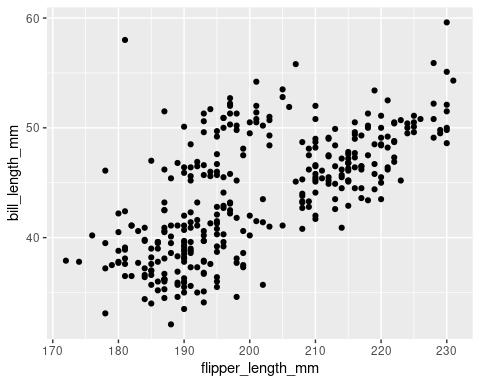


Try adjusting the bins

### Two variables: num vs num

penguins %>%   
 ggplot(aes(x = flipper\_length\_mm,   
 y = bill\_length\_mm)) +   
 geom\_point()

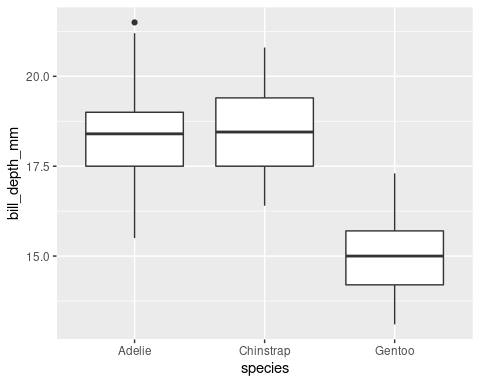
## Warning: Removed 2 rows containing missing values (geom\_point).



### Two variable num vs categorical

penguins %>%   
 ggplot(aes(x = species,   
 y = bill\_depth\_mm)) +   
 geom\_boxplot()

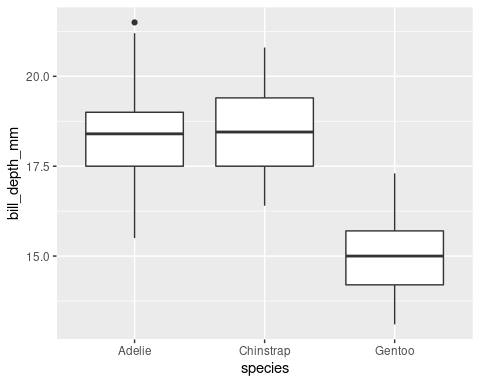
## Warning: Removed 2 rows containing non-finite values (stat\_boxplot).



Try with jitter

penguins %>%   
 ggplot(aes(x = species,   
 y = bill\_depth\_mm)) +   
 geom\_boxplot()

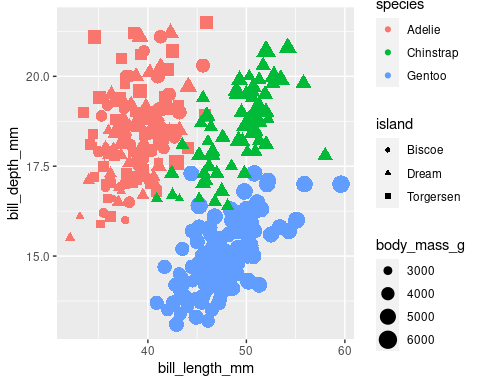
## Warning: Removed 2 rows containing non-finite values (stat\_boxplot).



add color, size, shape

penguins %>%   
 ggplot(aes(x = bill\_length\_mm,   
 y = bill\_depth\_mm,   
 color = species,   
 size = body\_mass\_g,   
 shape = island)) +   
 geom\_point()

## Warning: Removed 2 rows containing missing values (geom\_point).



# Third session

head(penguins)

## # A tibble: 6 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_… body\_mass\_g sex   
## <fct> <fct> <dbl> <dbl> <int> <int> <fct>  
## 1 Adelie Torge… 39.1 18.7 181 3750 male   
## 2 Adelie Torge… 39.5 17.4 186 3800 fema…  
## 3 Adelie Torge… 40.3 18 195 3250 fema…  
## 4 Adelie Torge… NA NA NA NA <NA>   
## 5 Adelie Torge… 36.7 19.3 193 3450 fema…  
## 6 Adelie Torge… 39.3 20.6 190 3650 male   
## # … with 1 more variable: year <int>

head(relig\_income)

## # A tibble: 6 × 11  
## religion `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k` `$50-75k` `$75-100k`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Agnostic 27 34 60 81 76 137 122  
## 2 Atheist 12 27 37 52 35 70 73  
## 3 Buddhist 27 21 30 34 33 58 62  
## 4 Catholic 418 617 732 670 638 1116 949  
## 5 Don’t kn… 15 14 15 11 10 35 21  
## 6 Evangeli… 575 869 1064 982 881 1486 949  
## # … with 3 more variables: $100-150k <dbl>, >150k <dbl>,  
## # Don't know/refused <dbl>

gapminder\_wide <- read\_csv("https://bit.ly/gapminder-rsu")

## Rows: 142 Columns: 38

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): continent, country  
## dbl (36): gdpPercap\_1952, gdpPercap\_1957, gdpPercap\_1962, gdpPercap\_1967, gd...

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

head(gapminder\_wide)

## # A tibble: 6 × 38  
## continent country gdpPercap\_1952 gdpPercap\_1957 gdpPercap\_1962 gdpPercap\_1967  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Africa Algeria 2449. 3014. 2551. 3247.  
## 2 Africa Angola 3521. 3828. 4269. 5523.  
## 3 Africa Benin 1063. 960. 949. 1036.  
## 4 Africa Botswana 851. 918. 984. 1215.  
## 5 Africa Burkina… 543. 617. 723. 795.  
## 6 Africa Burundi 339. 380. 355. 413.  
## # … with 32 more variables: gdpPercap\_1972 <dbl>, gdpPercap\_1977 <dbl>,  
## # gdpPercap\_1982 <dbl>, gdpPercap\_1987 <dbl>, gdpPercap\_1992 <dbl>,  
## # gdpPercap\_1997 <dbl>, gdpPercap\_2002 <dbl>, gdpPercap\_2007 <dbl>,  
## # lifeExp\_1952 <dbl>, lifeExp\_1957 <dbl>, lifeExp\_1962 <dbl>,  
## # lifeExp\_1967 <dbl>, lifeExp\_1972 <dbl>, lifeExp\_1977 <dbl>,  
## # lifeExp\_1982 <dbl>, lifeExp\_1987 <dbl>, lifeExp\_1992 <dbl>,  
## # lifeExp\_1997 <dbl>, lifeExp\_2002 <dbl>, lifeExp\_2007 <dbl>, …

glimpse(gapminder\_wide)

## Rows: 142  
## Columns: 38  
## $ continent <chr> "Africa", "Africa", "Africa", "Africa", "Africa", "Afri…  
## $ country <chr> "Algeria", "Angola", "Benin", "Botswana", "Burkina Faso…  
## $ gdpPercap\_1952 <dbl> 2449.0082, 3520.6103, 1062.7522, 851.2411, 543.2552, 33…  
## $ gdpPercap\_1957 <dbl> 3013.9760, 3827.9405, 959.6011, 918.2325, 617.1835, 379…  
## $ gdpPercap\_1962 <dbl> 2550.8169, 4269.2767, 949.4991, 983.6540, 722.5120, 355…  
## $ gdpPercap\_1967 <dbl> 3246.9918, 5522.7764, 1035.8314, 1214.7093, 794.8266, 4…  
## $ gdpPercap\_1972 <dbl> 4182.6638, 5473.2880, 1085.7969, 2263.6111, 854.7360, 4…  
## $ gdpPercap\_1977 <dbl> 4910.4168, 3008.6474, 1029.1613, 3214.8578, 743.3870, 5…  
## $ gdpPercap\_1982 <dbl> 5745.1602, 2756.9537, 1277.8976, 4551.1421, 807.1986, 5…  
## $ gdpPercap\_1987 <dbl> 5681.3585, 2430.2083, 1225.8560, 6205.8839, 912.0631, 6…  
## $ gdpPercap\_1992 <dbl> 5023.2166, 2627.8457, 1191.2077, 7954.1116, 931.7528, 6…  
## $ gdpPercap\_1997 <dbl> 4797.2951, 2277.1409, 1232.9753, 8647.1423, 946.2950, 4…  
## $ gdpPercap\_2002 <dbl> 5288.0404, 2773.2873, 1372.8779, 11003.6051, 1037.6452,…  
## $ gdpPercap\_2007 <dbl> 6223.3675, 4797.2313, 1441.2849, 12569.8518, 1217.0330,…  
## $ lifeExp\_1952 <dbl> 43.077, 30.015, 38.223, 47.622, 31.975, 39.031, 38.523,…  
## $ lifeExp\_1957 <dbl> 45.685, 31.999, 40.358, 49.618, 34.906, 40.533, 40.428,…  
## $ lifeExp\_1962 <dbl> 48.303, 34.000, 42.618, 51.520, 37.814, 42.045, 42.643,…  
## $ lifeExp\_1967 <dbl> 51.407, 35.985, 44.885, 53.298, 40.697, 43.548, 44.799,…  
## $ lifeExp\_1972 <dbl> 54.518, 37.928, 47.014, 56.024, 43.591, 44.057, 47.049,…  
## $ lifeExp\_1977 <dbl> 58.014, 39.483, 49.190, 59.319, 46.137, 45.910, 49.355,…  
## $ lifeExp\_1982 <dbl> 61.368, 39.942, 50.904, 61.484, 48.122, 47.471, 52.961,…  
## $ lifeExp\_1987 <dbl> 65.799, 39.906, 52.337, 63.622, 49.557, 48.211, 54.985,…  
## $ lifeExp\_1992 <dbl> 67.744, 40.647, 53.919, 62.745, 50.260, 44.736, 54.314,…  
## $ lifeExp\_1997 <dbl> 69.152, 40.963, 54.777, 52.556, 50.324, 45.326, 52.199,…  
## $ lifeExp\_2002 <dbl> 70.994, 41.003, 54.406, 46.634, 50.650, 47.360, 49.856,…  
## $ lifeExp\_2007 <dbl> 72.301, 42.731, 56.728, 50.728, 52.295, 49.580, 50.430,…  
## $ pop\_1952 <dbl> 9279525, 4232095, 1738315, 442308, 4469979, 2445618, 50…  
## $ pop\_1957 <dbl> 10270856, 4561361, 1925173, 474639, 4713416, 2667518, 5…  
## $ pop\_1962 <dbl> 11000948, 4826015, 2151895, 512764, 4919632, 2961915, 5…  
## $ pop\_1967 <dbl> 12760499, 5247469, 2427334, 553541, 5127935, 3330989, 6…  
## $ pop\_1972 <dbl> 14760787, 5894858, 2761407, 619351, 5433886, 3529983, 7…  
## $ pop\_1977 <dbl> 17152804, 6162675, 3168267, 781472, 5889574, 3834415, 7…  
## $ pop\_1982 <dbl> 20033753, 7016384, 3641603, 970347, 6634596, 4580410, 9…  
## $ pop\_1987 <dbl> 23254956, 7874230, 4243788, 1151184, 7586551, 5126023, …  
## $ pop\_1992 <dbl> 26298373, 8735988, 4981671, 1342614, 8878303, 5809236, …  
## $ pop\_1997 <dbl> 29072015, 9875024, 6066080, 1536536, 10352843, 6121610,…  
## $ pop\_2002 <dbl> 31287142, 10866106, 7026113, 1630347, 12251209, 7021078…  
## $ pop\_2007 <dbl> 33333216, 12420476, 8078314, 1639131, 14326203, 8390505…

How many countries?

gapminder\_wide %>%   
 count(country)

## # A tibble: 142 × 2  
## country n  
## <chr> <int>  
## 1 Afghanistan 1  
## 2 Albania 1  
## 3 Algeria 1  
## 4 Angola 1  
## 5 Argentina 1  
## 6 Australia 1  
## 7 Austria 1  
## 8 Bahrain 1  
## 9 Bangladesh 1  
## 10 Belgium 1  
## # … with 132 more rows

n\_distinct(gapminder\_wide$country)

## [1] 142

Count continents

gapminder\_wide %>%   
 count(continent)

## # A tibble: 5 × 2  
## continent n  
## <chr> <int>  
## 1 Africa 52  
## 2 Americas 25  
## 3 Asia 33  
## 4 Europe 30  
## 5 Oceania 2

n\_distinct(gapminder\_wide$continent)

## [1] 5

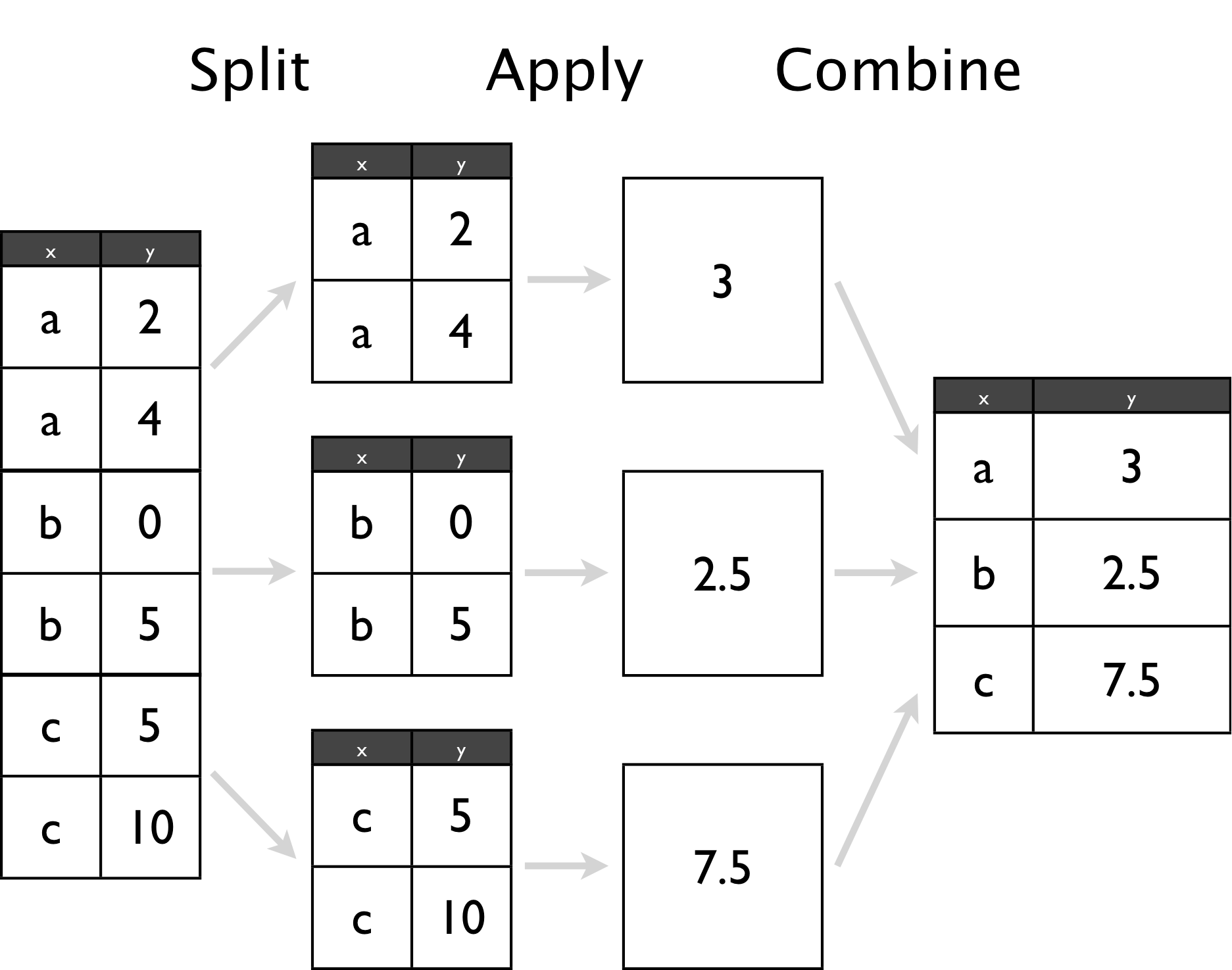
## Wide vs long formats

### Let’s look at the columns: how many variables?

gdpPercap\_1952:gdpPercap\_2007

pop\_1952:pop\_2007

## Split-apply-combine



### Question: how has GDP per continent evolved per year?

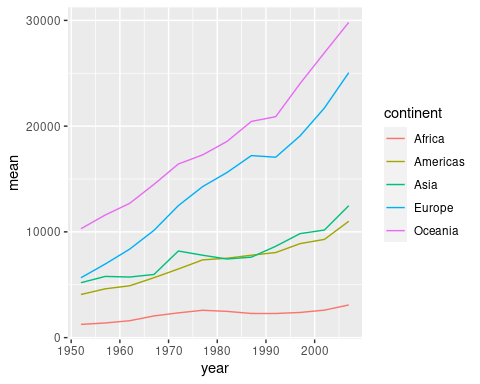
gapminder %>%   
 # split  
 group\_by(continent, year) %>%   
 # apply & combine  
 summarise(mean = mean(gdpPercap))

## `summarise()` has grouped output by 'continent'. You can override using the `.groups` argument.

## # A tibble: 60 × 3  
## # Groups: continent [5]  
## continent year mean  
## <fct> <int> <dbl>  
## 1 Africa 1952 1253.  
## 2 Africa 1957 1385.  
## 3 Africa 1962 1598.  
## 4 Africa 1967 2050.  
## 5 Africa 1972 2340.  
## 6 Africa 1977 2586.  
## 7 Africa 1982 2482.  
## 8 Africa 1987 2283.  
## 9 Africa 1992 2282.  
## 10 Africa 1997 2379.  
## # … with 50 more rows

gapminder %>%   
 # split  
 group\_by(continent, year) %>%   
 # apply & combine  
 summarise(mean = mean(gdpPercap)) %>%   
 ggplot(aes(x = year,   
 y = mean,   
 color = continent)) +   
 geom\_line()

## `summarise()` has grouped output by 'continent'. You can override using the `.groups` argument.



### Question: how has POPULATION per continent evolved per year?

Select only the Baltic countries and explore their changes in gdp and population

Homework: See [<https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations>](https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations)

# Fourth session

compare gapminder wide vs long

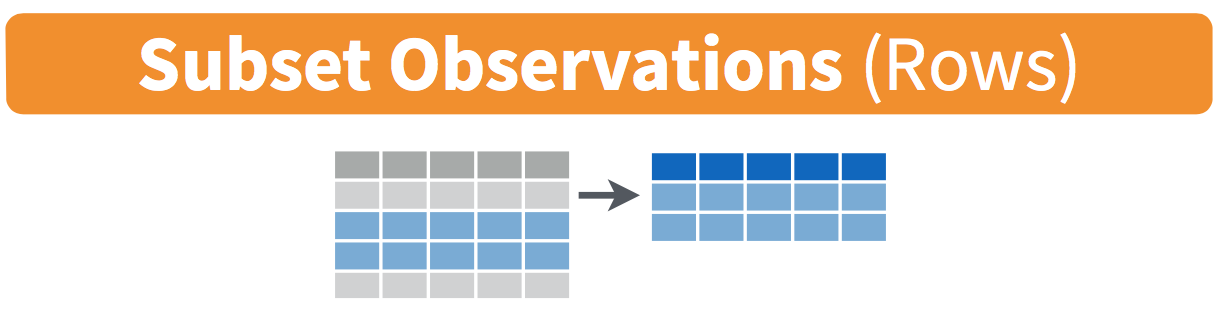
gapminder\_wide <- read\_csv(“[https://bit.ly/gapminder-rsu”)](https://bit.ly/gapminder-rsu%22))

Load the packages

pacman::p\_load(tidyverse, # several datascience packages  
 palmerpenguins, # data  
 gapminder, #data,  
 gtsummary) # for tables)

## Data basics

#### Filtering



penguins %>%   
 filter(species == "Adelie")

## # A tibble: 152 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## <fct> <fct> <dbl> <dbl> <int> <int>  
## 1 Adelie Torgersen 39.1 18.7 181 3750  
## 2 Adelie Torgersen 39.5 17.4 186 3800  
## 3 Adelie Torgersen 40.3 18 195 3250  
## 4 Adelie Torgersen NA NA NA NA  
## 5 Adelie Torgersen 36.7 19.3 193 3450  
## 6 Adelie Torgersen 39.3 20.6 190 3650  
## 7 Adelie Torgersen 38.9 17.8 181 3625  
## 8 Adelie Torgersen 39.2 19.6 195 4675  
## 9 Adelie Torgersen 34.1 18.1 193 3475  
## 10 Adelie Torgersen 42 20.2 190 4250  
## # … with 142 more rows, and 2 more variables: sex <fct>, year <int>

penguins %>%   
 filter(species == "Adelie" & island == "Dream")

## # A tibble: 56 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## <fct> <fct> <dbl> <dbl> <int> <int>  
## 1 Adelie Dream 39.5 16.7 178 3250  
## 2 Adelie Dream 37.2 18.1 178 3900  
## 3 Adelie Dream 39.5 17.8 188 3300  
## 4 Adelie Dream 40.9 18.9 184 3900  
## 5 Adelie Dream 36.4 17 195 3325  
## 6 Adelie Dream 39.2 21.1 196 4150  
## 7 Adelie Dream 38.8 20 190 3950  
## 8 Adelie Dream 42.2 18.5 180 3550  
## 9 Adelie Dream 37.6 19.3 181 3300  
## 10 Adelie Dream 39.8 19.1 184 4650  
## # … with 46 more rows, and 2 more variables: sex <fct>, year <int>

penguins %>%   
 filter(body\_mass\_g < 3000)

## # A tibble: 9 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_… body\_mass\_g sex   
## <fct> <fct> <dbl> <dbl> <int> <int> <fct>  
## 1 Adelie Dream 37.5 18.9 179 2975 <NA>   
## 2 Adelie Biscoe 34.5 18.1 187 2900 fema…  
## 3 Adelie Biscoe 36.5 16.6 181 2850 fema…  
## 4 Adelie Biscoe 36.4 17.1 184 2850 fema…  
## 5 Adelie Dream 33.1 16.1 178 2900 fema…  
## 6 Adelie Biscoe 37.9 18.6 193 2925 fema…  
## 7 Adelie Torge… 38.6 17 188 2900 fema…  
## 8 Chinst… Dream 43.2 16.6 187 2900 fema…  
## 9 Chinst… Dream 46.9 16.6 192 2700 fema…  
## # … with 1 more variable: year <int>

You can use other operators beyond just the == operator that tests for equality:

* > corresponds to “greater than”
* < corresponds to “less than”
* >= corresponds to “greater than or equal to”
* <= corresponds to “less than or equal to”
* != corresponds to “not equal to.” The ! is used in many programming languages to indicate “not.”

[&](https://rdrr.io/r/base/Logic.html) is “and”, [|](https://rdrr.io/r/base/Logic.html) is “or”, and [!](https://rdrr.io/r/base/Logic.html) is “not”.

penguins %>%   
 filter(species == "Adelie" & island != "Torgersen")

## # A tibble: 100 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## <fct> <fct> <dbl> <dbl> <int> <int>  
## 1 Adelie Biscoe 37.8 18.3 174 3400  
## 2 Adelie Biscoe 37.7 18.7 180 3600  
## 3 Adelie Biscoe 35.9 19.2 189 3800  
## 4 Adelie Biscoe 38.2 18.1 185 3950  
## 5 Adelie Biscoe 38.8 17.2 180 3800  
## 6 Adelie Biscoe 35.3 18.9 187 3800  
## 7 Adelie Biscoe 40.6 18.6 183 3550  
## 8 Adelie Biscoe 40.5 17.9 187 3200  
## 9 Adelie Biscoe 37.9 18.6 172 3150  
## 10 Adelie Biscoe 40.5 18.9 180 3950  
## # … with 90 more rows, and 2 more variables: sex <fct>, year <int>

penguins %>%   
 filter(species == "Adelie" | species == "Gentoo")

## # A tibble: 276 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## <fct> <fct> <dbl> <dbl> <int> <int>  
## 1 Adelie Torgersen 39.1 18.7 181 3750  
## 2 Adelie Torgersen 39.5 17.4 186 3800  
## 3 Adelie Torgersen 40.3 18 195 3250  
## 4 Adelie Torgersen NA NA NA NA  
## 5 Adelie Torgersen 36.7 19.3 193 3450  
## 6 Adelie Torgersen 39.3 20.6 190 3650  
## 7 Adelie Torgersen 38.9 17.8 181 3625  
## 8 Adelie Torgersen 39.2 19.6 195 4675  
## 9 Adelie Torgersen 34.1 18.1 193 3475  
## 10 Adelie Torgersen 42 20.2 190 4250  
## # … with 266 more rows, and 2 more variables: sex <fct>, year <int>

penguins %>%   
 filter(species %in% c("Adelie", "Gentoo"))

## # A tibble: 276 × 8  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## <fct> <fct> <dbl> <dbl> <int> <int>  
## 1 Adelie Torgersen 39.1 18.7 181 3750  
## 2 Adelie Torgersen 39.5 17.4 186 3800  
## 3 Adelie Torgersen 40.3 18 195 3250  
## 4 Adelie Torgersen NA NA NA NA  
## 5 Adelie Torgersen 36.7 19.3 193 3450  
## 6 Adelie Torgersen 39.3 20.6 190 3650  
## 7 Adelie Torgersen 38.9 17.8 181 3625  
## 8 Adelie Torgersen 39.2 19.6 195 4675  
## 9 Adelie Torgersen 34.1 18.1 193 3475  
## 10 Adelie Torgersen 42 20.2 190 4250  
## # … with 266 more rows, and 2 more variables: sex <fct>, year <int>

TASK

Filter all penguins from Torgersen weighing more than 3500g

#### Mutate

Create new variables

Bill ratio = bill\_length\_mm / bill\_depth\_mm

penguins %>%   
 mutate(bill\_ratio = bill\_length\_mm / bill\_depth\_mm)

## # A tibble: 344 × 9  
## species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## <fct> <fct> <dbl> <dbl> <int> <int>  
## 1 Adelie Torgersen 39.1 18.7 181 3750  
## 2 Adelie Torgersen 39.5 17.4 186 3800  
## 3 Adelie Torgersen 40.3 18 195 3250  
## 4 Adelie Torgersen NA NA NA NA  
## 5 Adelie Torgersen 36.7 19.3 193 3450  
## 6 Adelie Torgersen 39.3 20.6 190 3650  
## 7 Adelie Torgersen 38.9 17.8 181 3625  
## 8 Adelie Torgersen 39.2 19.6 195 4675  
## 9 Adelie Torgersen 34.1 18.1 193 3475  
## 10 Adelie Torgersen 42 20.2 190 4250  
## # … with 334 more rows, and 3 more variables: sex <fct>, year <int>,  
## # bill\_ratio <dbl>

penguins %>%   
 mutate(bill\_ratio = bill\_length\_mm / bill\_depth\_mm) %>%   
 relocate(bill\_ratio, .after = island)

## # A tibble: 344 × 9  
## species island bill\_ratio bill\_length\_mm bill\_depth\_mm flipper\_length\_mm  
## <fct> <fct> <dbl> <dbl> <dbl> <int>  
## 1 Adelie Torgersen 2.09 39.1 18.7 181  
## 2 Adelie Torgersen 2.27 39.5 17.4 186  
## 3 Adelie Torgersen 2.24 40.3 18 195  
## 4 Adelie Torgersen NA NA NA NA  
## 5 Adelie Torgersen 1.90 36.7 19.3 193  
## 6 Adelie Torgersen 1.91 39.3 20.6 190  
## 7 Adelie Torgersen 2.19 38.9 17.8 181  
## 8 Adelie Torgersen 2 39.2 19.6 195  
## 9 Adelie Torgersen 1.88 34.1 18.1 193  
## 10 Adelie Torgersen 2.08 42 20.2 190  
## # … with 334 more rows, and 3 more variables: body\_mass\_g <int>, sex <fct>,  
## # year <int>

TASK

Convert the body\_mass\_g to k

#### Grouping and summarizing data

penguins %>%   
 group\_by(species) %>%   
 summarise(mean\_body\_mass\_g = mean(body\_mass\_g))

## # A tibble: 3 × 2  
## species mean\_body\_mass\_g  
## <fct> <dbl>  
## 1 Adelie NA   
## 2 Chinstrap 3733.  
## 3 Gentoo NA

penguins %>%   
 group\_by(species) %>%   
 summarise(mean\_body\_mass\_g = mean(body\_mass\_g, na.rm = T))

## # A tibble: 3 × 2  
## species mean\_body\_mass\_g  
## <fct> <dbl>  
## 1 Adelie 3701.  
## 2 Chinstrap 3733.  
## 3 Gentoo 5076.

Count

penguins %>%   
 drop\_na() %>%   
 group\_by(species) %>%   
 summarise(mean\_body\_mass\_g = mean(body\_mass\_g),   
 max = max(body\_mass\_g),   
 min = min(body\_mass\_g),   
 sd = sd(body\_mass\_g),   
 n = n())

## # A tibble: 3 × 6  
## species mean\_body\_mass\_g max min sd n  
## <fct> <dbl> <int> <int> <dbl> <int>  
## 1 Adelie 3706. 4775 2850 459. 146  
## 2 Chinstrap 3733. 4800 2700 384. 68  
## 3 Gentoo 5092. 6300 3950 501. 119

penguins %>%   
 drop\_na() %>%   
 group\_by(species) %>%   
 summarise(mean\_body\_mass\_g = mean(body\_mass\_g),   
 max = max(body\_mass\_g),   
 min = min(body\_mass\_g),   
 sd = sd(body\_mass\_g),   
 n = n()) %>%   
 knitr::kable()

| species | mean\_body\_mass\_g | max | min | sd | n |
| --- | --- | --- | --- | --- | --- |
| Adelie | 3706.164 | 4775 | 2850 | 458.6201 | 146 |
| Chinstrap | 3733.088 | 4800 | 2700 | 384.3351 | 68 |
| Gentoo | 5092.437 | 6300 | 3950 | 501.4762 | 119 |

Create a summary table for the flipper\_length\_mm with the median, sd and maximum

## Managing data

### Load data

gap\_wide <- read\_csv("https://bit.ly/gapminder-rsu")

## Rows: 142 Columns: 38

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): continent, country  
## dbl (36): gdpPercap\_1952, gdpPercap\_1957, gdpPercap\_1962, gdpPercap\_1967, gd...

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

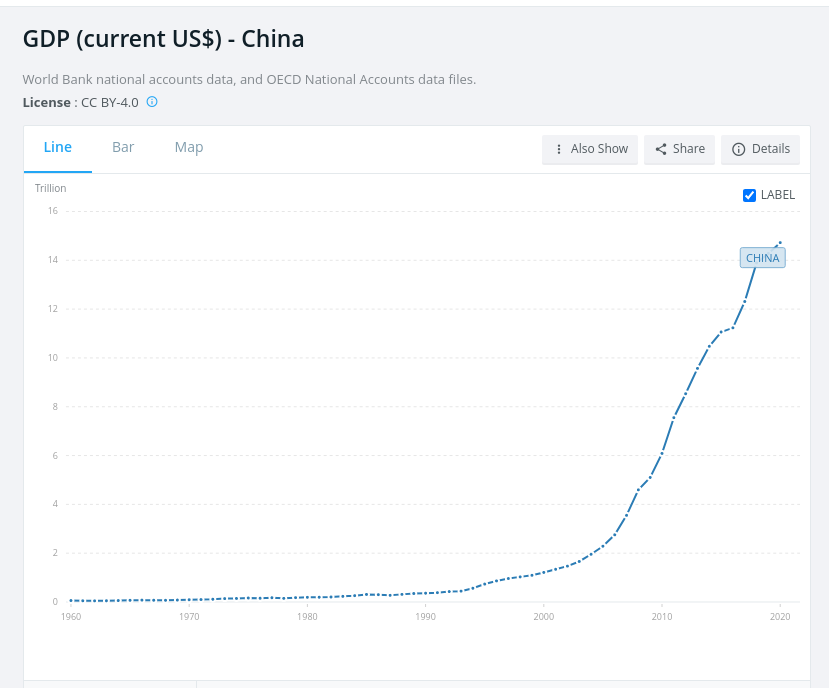
### Explore the data

head(gap\_wide)

## # A tibble: 6 × 38  
## continent country gdpPercap\_1952 gdpPercap\_1957 gdpPercap\_1962 gdpPercap\_1967  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Africa Algeria 2449. 3014. 2551. 3247.  
## 2 Africa Angola 3521. 3828. 4269. 5523.  
## 3 Africa Benin 1063. 960. 949. 1036.  
## 4 Africa Botswana 851. 918. 984. 1215.  
## 5 Africa Burkina… 543. 617. 723. 795.  
## 6 Africa Burundi 339. 380. 355. 413.  
## # … with 32 more variables: gdpPercap\_1972 <dbl>, gdpPercap\_1977 <dbl>,  
## # gdpPercap\_1982 <dbl>, gdpPercap\_1987 <dbl>, gdpPercap\_1992 <dbl>,  
## # gdpPercap\_1997 <dbl>, gdpPercap\_2002 <dbl>, gdpPercap\_2007 <dbl>,  
## # lifeExp\_1952 <dbl>, lifeExp\_1957 <dbl>, lifeExp\_1962 <dbl>,  
## # lifeExp\_1967 <dbl>, lifeExp\_1972 <dbl>, lifeExp\_1977 <dbl>,  
## # lifeExp\_1982 <dbl>, lifeExp\_1987 <dbl>, lifeExp\_1992 <dbl>,  
## # lifeExp\_1997 <dbl>, lifeExp\_2002 <dbl>, lifeExp\_2007 <dbl>, …

names(gap\_wide)

## [1] "continent" "country" "gdpPercap\_1952" "gdpPercap\_1957"  
## [5] "gdpPercap\_1962" "gdpPercap\_1967" "gdpPercap\_1972" "gdpPercap\_1977"  
## [9] "gdpPercap\_1982" "gdpPercap\_1987" "gdpPercap\_1992" "gdpPercap\_1997"  
## [13] "gdpPercap\_2002" "gdpPercap\_2007" "lifeExp\_1952" "lifeExp\_1957"   
## [17] "lifeExp\_1962" "lifeExp\_1967" "lifeExp\_1972" "lifeExp\_1977"   
## [21] "lifeExp\_1982" "lifeExp\_1987" "lifeExp\_1992" "lifeExp\_1997"   
## [25] "lifeExp\_2002" "lifeExp\_2007" "pop\_1952" "pop\_1957"   
## [29] "pop\_1962" "pop\_1967" "pop\_1972" "pop\_1977"   
## [33] "pop\_1982" "pop\_1987" "pop\_1992" "pop\_1997"   
## [37] "pop\_2002" "pop\_2007"

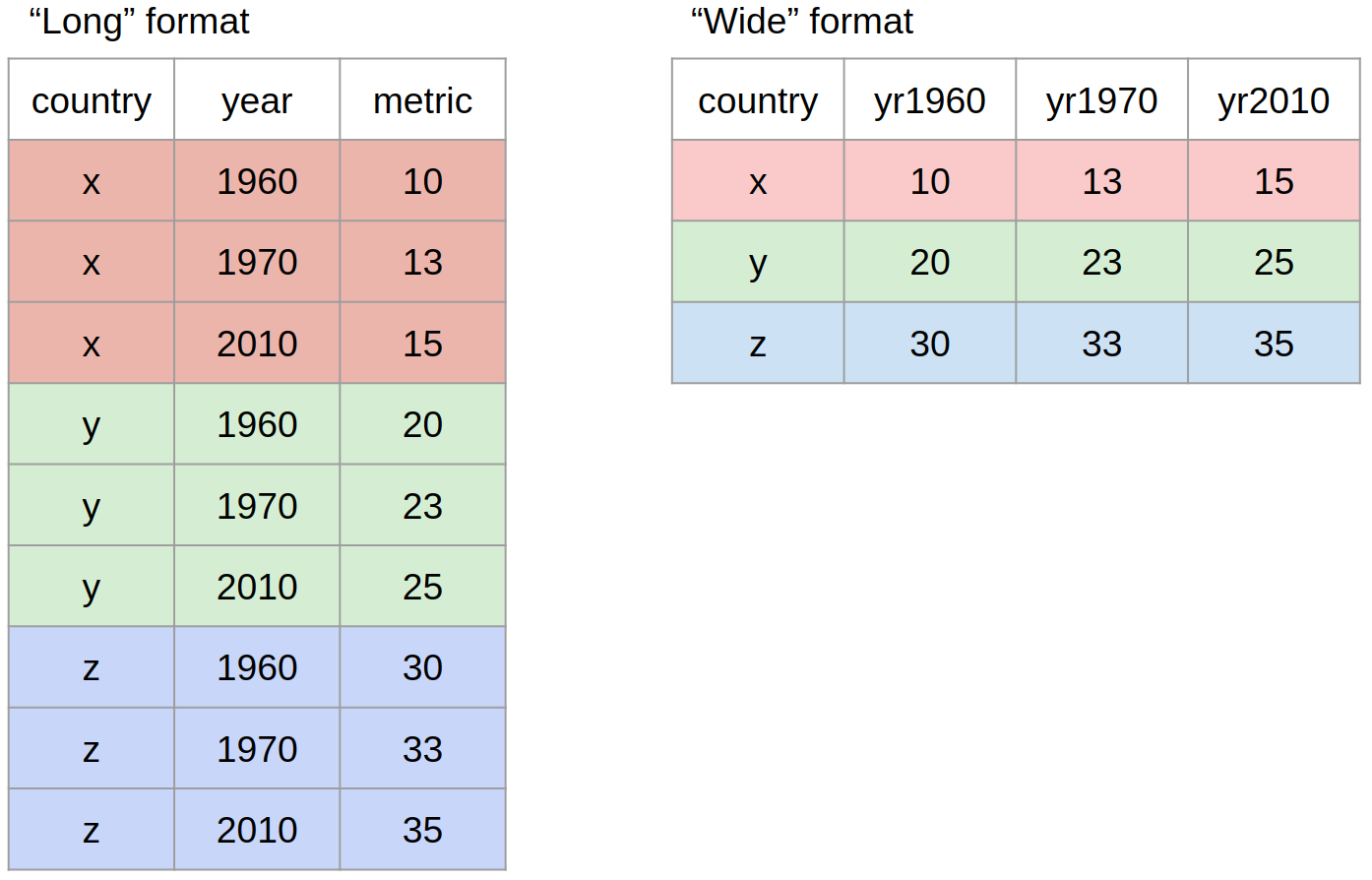


gap\_wide %>%   
 filter(country == "China")

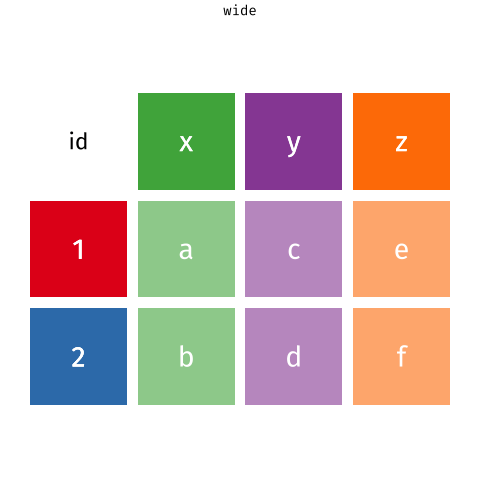
## # A tibble: 1 × 38  
## continent country gdpPercap\_1952 gdpPercap\_1957 gdpPercap\_1962 gdpPercap\_1967  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Asia China 400. 576. 488. 613.  
## # … with 32 more variables: gdpPercap\_1972 <dbl>, gdpPercap\_1977 <dbl>,  
## # gdpPercap\_1982 <dbl>, gdpPercap\_1987 <dbl>, gdpPercap\_1992 <dbl>,  
## # gdpPercap\_1997 <dbl>, gdpPercap\_2002 <dbl>, gdpPercap\_2007 <dbl>,  
## # lifeExp\_1952 <dbl>, lifeExp\_1957 <dbl>, lifeExp\_1962 <dbl>,  
## # lifeExp\_1967 <dbl>, lifeExp\_1972 <dbl>, lifeExp\_1977 <dbl>,  
## # lifeExp\_1982 <dbl>, lifeExp\_1987 <dbl>, lifeExp\_1992 <dbl>,  
## # lifeExp\_1997 <dbl>, lifeExp\_2002 <dbl>, lifeExp\_2007 <dbl>, …

How to plot ?

### Wide vs long



* Wide data is for humans
* Long data is for computers



#### Wide to long

gap\_wide %>%   
 pivot\_longer(gdpPercap\_1952:gdpPercap\_2007,   
 names\_to = )

## # A tibble: 1,704 × 28  
## continent country lifeExp\_1952 lifeExp\_1957 lifeExp\_1962 lifeExp\_1967  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Africa Algeria 43.1 45.7 48.3 51.4  
## 2 Africa Algeria 43.1 45.7 48.3 51.4  
## 3 Africa Algeria 43.1 45.7 48.3 51.4  
## 4 Africa Algeria 43.1 45.7 48.3 51.4  
## 5 Africa Algeria 43.1 45.7 48.3 51.4  
## 6 Africa Algeria 43.1 45.7 48.3 51.4  
## 7 Africa Algeria 43.1 45.7 48.3 51.4  
## 8 Africa Algeria 43.1 45.7 48.3 51.4  
## 9 Africa Algeria 43.1 45.7 48.3 51.4  
## 10 Africa Algeria 43.1 45.7 48.3 51.4  
## # … with 1,694 more rows, and 22 more variables: lifeExp\_1972 <dbl>,  
## # lifeExp\_1977 <dbl>, lifeExp\_1982 <dbl>, lifeExp\_1987 <dbl>,  
## # lifeExp\_1992 <dbl>, lifeExp\_1997 <dbl>, lifeExp\_2002 <dbl>,  
## # lifeExp\_2007 <dbl>, pop\_1952 <dbl>, pop\_1957 <dbl>, pop\_1962 <dbl>,  
## # pop\_1967 <dbl>, pop\_1972 <dbl>, pop\_1977 <dbl>, pop\_1982 <dbl>,  
## # pop\_1987 <dbl>, pop\_1992 <dbl>, pop\_1997 <dbl>, pop\_2002 <dbl>,  
## # pop\_2007 <dbl>, name <chr>, value <dbl>

### Filtering data