MAD – Data Analysis & Biostatistics in R

Exploratory Data Analysis and Inference

James R. Hunter, Ph.D.

DIPA, EPM, UNIFESP

18 September 2020

MAD

Section 1

But, First, 2 More Important Data Munging Functions

Section 2

Long vs. Wide Data

What This Means

- Spreadsheets usually present data in wide format
  - Each case has a number of variables
- For some analyses, we need to combine some of these variables
  - This would make the format of the data long

Example Database

- From State of São Paulo database (SEADE), table of comorbidities
- Randomized set of 300 cases of demographic and comorbidity info
- Dataset already "tidy"

```r
sp_comorb <- readRDS(here::here("seade_comorb_sample.rds")) %>%
  mutate(pacid = 1:nrow(.), .before = 1) # add pacid to make what is happening clear
glimpse(sp_comorb)
```

```
## Rows: 300
## Columns: 10
## $ pacid       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ city        <chr> "Itaquaquecetuba", "Sorocaba", "Sao Paulo", "Sao Paulo"...
## $ age         <dbl> 58, 62, 78, 65, 59, 68, 67, 83, 61, 58, 73, 67, 77, 77,...
## $ sex         <fct> male, male, female, female, male, female, female, femal...
## $ death       <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, T...
## $ cardiopathy <fct> true, true, true, true, false, true, true, true, true, ...
## $ diabetes    <fct> true, NA, true, true, true, NA, NA, NA, false, NA, true...
## $ obesity     <fct> false, NA, NA, NA, false, true, NA, NA, false, NA, fals...
## $ neuro       <fct> false, NA, NA, NA, false, NA, NA, NA, false, NA, false,...
## $ kidney      <fct> false, NA, true, NA, false, NA, NA, true, false, NA, fa...
```

Change Format for Current Analysis

- For current analysis, want to study comorbidities as a group
  - Not as individual conditions
- In this case . . .
  - Each comorbidity variable is not really a variable in itself
    - They are values of two new variables
    - `comorbid`: the name of the comorbidity (its *key*)
    - `value`: presence or absence of condition (its *value*)
- *key:value* pair

Function `tidyr::pivot_longer()`

- `cols` = columns that will be combined into key:value pair
- `names_to` = give a name to the variable that will hold the keys
- `values_to` = give a name to the variable for the values

```r
sp_comorb_long <- sp_comorb %>%
  pivot_longer(cols = cardiopathy:kidney, names_to = "comorbid",
               values_to = "value")
glimpse(sp_comorb_long)
```

```
## Rows: 1,500
## Columns: 7
## $ pacid    <int> 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4...
## $ city     <chr> "Itaquaquecetuba", "Itaquaquecetuba", "Itaquaquecetuba", "...
## $ age      <dbl> 58, 58, 58, 58, 58, 62, 62, 62, 62, 62, 78, 78, 78, 78, 78...
## $ sex      <fct> male, male, male, male, male, male, male, male, male, male...
## $ death    <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE,...
## $ comorbid <chr> "cardiopathy", "diabetes", "obesity", "neuro", "kidney", "...
## $ value    <fct> true, true, false, false, false, true, NA, NA, NA, NA, tru...
```

Long Now about Comorbidities

- Patients not basic units in this format
  - Each `pacid` appears 5 times
  - 1 for each comorbidity

Can Take a Long Tibble and Make It Wide

- `tidyr::pivot_wider()`
- Values of *key* variable become names of wider variables
- Values of *value* variable become values related to these

```r
sp_comorb_wide <- sp_comorb_long %>%
  pivot_wider(names_from = "comorbid",
              values_from = "value")
glimpse(sp_comorb_wide)
```

```
## Rows: 300
## Columns: 10
## $ pacid       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ city        <chr> "Itaquaquecetuba", "Sorocaba", "Sao Paulo", "Sao Paulo"...
## $ age         <dbl> 58, 62, 78, 65, 59, 68, 67, 83, 61, 58, 73, 67, 77, 77,...
## $ sex         <fct> male, male, female, female, male, female, female, femal...
## $ death       <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, T...
## $ cardiopathy <fct> true, true, true, true, false, true, true, true, true, ...
## $ diabetes    <fct> true, NA, true, true, true, NA, NA, NA, false, NA, true...
## $ obesity     <fct> false, NA, NA, NA, false, true, NA, NA, false, NA, fals...
## $ neuro       <fct> false, NA, NA, NA, false, NA, NA, NA, false, NA, false,...
## $ kidney      <fct> false, NA, true, NA, false, NA, NA, true, false, NA, fa...
```

Section 3

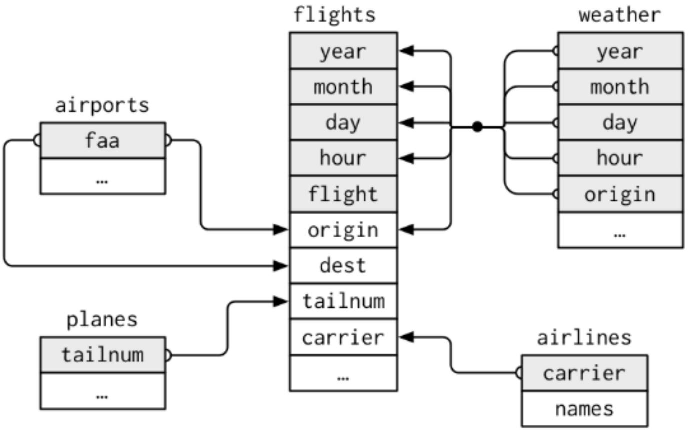
Joining Data from Different Tibbles

Joining Data from Different Tibbles

- Data for an analysis frequently comes from more than 1 table
- Especially true for data from *relational data bases* like SQL
- `join...` functions to integrate data frames based on common keys

Data for Joins

- Data on flights leaving any of NYC commerical airports in 2013
  - Package `nycflights13`
  - Tables for
    - Names of airlines serving the airports
    - Airports flights went to
    - Planes - models and tail numbers
    - Weather at airports
    - Flights - central component of system


flights
year
month
day
hour
flight
origin
dest
tailnum
carrier
...
weather
year
month
day
hour
origin
...
airports
faa
...
planes
tailnum
...
airlines
carrier
names

Select Sample of 10 Flights

```r
library(nycflights13)
data(flights)
# select a set of 10 flights
flights <- flights %>%
  slice_sample(n = 10) %>%
  select(year:day, flight, origin, dest, carrier) # select subset of vars
flights
```

```
## # A tibble: 10 x 7
##     year month   day flight origin dest  carrier
##    <int> <int> <int>  <int> <chr>  <chr> <chr>
##  1  2013     2    15   1871 LGA    MIA   AA
##  2  2013     5     2   4131 EWR    RIC   EV
##  3  2013     3    20   2083 EWR    DFW   AA
##  4  2013     7    13   1585 LGA    MCO   DL
##  5  2013     7     8   1223 EWR    DFW   AA
##  6  2013     4     1    119 JFK    MSY   B6
##  7  2013     4     6    161 EWR    LAX   UA
##  8  2013     4    28   1524 EWR    ORD   UA
##  9  2013     7    31    575 EWR    ATL   DL
## 10  2013     2    12   3719 LGA    RIC   9E
```

Only Carrier ID, No Carrier Name (airlines)

```
# load the airlines list
data(airlines)
head(airlines)
```

```
## # A tibble: 6 x 2
##   carrier name
##   <chr>   <chr>
## 1 9E      Endeavor Air Inc.
## 2 AA      American Airlines Inc.
## 3 AS      Alaska Airlines Inc.
## 4 B6      JetBlue Airways
## 5 DL      Delta Air Lines Inc.
## 6 EV      ExpressJet Airlines Inc.
```

Joining the Carrier Name to the Flights

- Both tables have variable `carrier`
  - Carrier 2-digit code
- `left_join()`
  - Join the data in the right hand base to the data in the left-hand base
  - Using common variable
  - Only show columns related to problem at hand

```r
# join the airline names to the flights
flights_mod <- flights %>%
  left_join(airlines, by = "carrier")
flights_mod[, 4:8]
```

```
## # A tibble: 10 x 5
##    flight origin dest  carrier name
##     <int> <chr>  <chr> <chr>   <chr>
## 1    1871 LGA    MIA   AA      American Airlines Inc.
## 2    4131 EWR    RIC   EV      ExpressJet Airlines Inc.
## 3    2083 EWR    DFW   AA      American Airlines Inc.
## 4    1585 LGA    MCO   DL      Delta Air Lines Inc.
## 5    1223 EWR    DFW   AA      American Airlines Inc.
## 6     119 JFK    MSY   B6      JetBlue Airways
## 7     161 EWR    LAX   UA      United Air Lines Inc.
## 8    1524 EWR    ORD   UA      United Air Lines Inc.
## 9     575 EWR    ATL   DL      Delta Air Lines Inc.
## 10   3719 LGA    RIC   9E      Endeavor Air Inc.
```

Types of Joins

- Mutating joins like `left_join`
  - Change the left-hand data frame
    - Can remove rows from left frame
  - Drawing data from right hand frame
  - Leaving right hand frame alone
- Other mutating joins
  - `right_join()`
    - Roles of right and left bases reversed
  - `full_join()`
    - Retains all records on left whether exists corresponding key in right or not
  - `inner_join()` Only retains records with key value in both bases

Note on Keys in Joins

- If left and right keys have different names, need different `by =`
- Case of key = `a` on left and `b` on right
- `by = c("a" = "b")`
  - Note use of `c()` function
  - Note use of quotation marks

Section 4

Exploratory Data Analysis

Initial Exploration of Data

- Place where we try to find "what the data are saying"
- Series of measures and graphs that display the variables
- Exploration of variables
  - One at a time (univariate)
  - Crosstabulations of sets of variables
- On the lookout for weird data values

Why Do We Do This?

- Even after we munged the dataset
  Never trust anything you have not directly observed in a data set.
  "Everybody lies." - Dr. House

Data: fute_mod.rds

- Database of football (soccer) related injuries in US

```
library(tidyverse)
fm <- readRDS(here::here("fute_mod_2020.rds")) %>%
  mutate(age_grp = factor(case_when(
    age < 18 ~ "youth",
    age < 60 ~ "adult",
    TRUE ~ "elderly"
  ))) %>%
  mutate(age_grp = fct_relevel(age_grp, c("youth", "adult", "elderly")))
glimpse(fm)
```

```
## Rows: 7,603
## Columns: 10
## $ case_num    <chr> "160102033", "160106032", "160107304", "160109914", "16...
## $ trmt_date   <date> 2016-01-02, 2016-01-02, 2016-01-01, 2016-01-01, 2016-0...
## $ age         <dbl> 27, 14, 9, 16, 17, 33, 12, 16, 12, 50, 10, 15, 17, 11, ...
## $ sex         <fct> Male, Male, Male, Female, Female, Male, Male, Female, M...
## $ body_part   <fct> Foot, Knee, Toe, Wrist, Wrist, Knee, Finger, Head, Fing...
## $ diag        <fct> "Contusion Or Abrasion", "Fracture", "Fracture", "Strai...
## $ disposition <fct> Released, Released, Released, Released, Released, Relea...
## $ psu         <fct> 63, 61, 8, 20, 73, 61, 58, 61, 63, 61, 20, 20, 20, 17, ...
## $ narrative   <chr> "27YOM PLAYING SOCCER COLLIDED WITH ANOTHER PLAYER CONT...
## $ age_grp     <fct> adult, youth, youth, youth, youth, adult, youth, youth,...
```

age Variable

```
summarytools::descr(fm$age)
```

```
## Descriptive Statistics
## fm$age
## N: 7603
##
##                           age
## ----------------- ---------
##              Mean     16.38
##           Std.Dev      8.92
##               Min      0.00
##                Q1     11.00
##            Median     14.00
##                Q3     17.00
##               Max     85.00
##               MAD      4.45
##               IQR      6.00
##                CV      0.54
##          Skewness      2.22
##       SE.Skewness      0.03
##          Kurtosis      6.60
##           N.Valid   7603.00
##         Pct.Valid    100.00
```

Minimum = 0.00 ?

```
summarytools::descr(fm$age)
```

```
## Descriptive Statistics
## fm$age
## N: 7603
##
##                           age
## ----------------- ---------
##              Mean     16.38
##           Std.Dev      8.92
##               Min      0.00
##                Q1     11.00
##            Median     14.00
##                Q3     17.00
##               Max     85.00
##               MAD      4.45
##               IQR      6.00
##                CV      0.54
##          Skewness      2.22
##       SE.Skewness      0.03
##          Kurtosis      6.60
##           N.Valid   7603.00
##         Pct.Valid    100.00
```

Who Is That Person with Age = 0?

- UNK AGE MALE WAS HEADBUTTED BY ANOTHER PLAYER WHILE PLAYING SOCCERDX NOSE FX
- Not a baby; Person of unknown age
- Change age = 0 to `NA`
- Are there more cases with age = 0 or near it?

How Many Cases Below 5 Years

- Year in which American kids start school

```
fm %>%
  filter(age < 5) %>%
  summarise(n = n())
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    82
```

Section 5

Measures of Central Tendency

Interest in People Who Play Football

- What kind of injuries occur in **amateurs** playing soccer
- Eliminate cases with ages less than 5

```
fm_mk2 <- fm %>%
  filter(age >= 5)
summarytools::descr(fm_mk2$age)
```

```
## Descriptive Statistics
## fm_mk2$age
## N: 7521
##
##                          age
## ----------------- ---------
##              Mean     16.53
##           Std.Dev      8.86
##               Min      5.00
##                Q1     12.00
##            Median     14.00
##                Q3     17.00
##               Max     85.00
##               MAD      4.45
##               IQR      5.00
##                CV      0.54
##          Skewness      2.28
```

Means of Two Distributions

- Mean of `fm` (with small kids): 16.3786225
- Mean of `fm_mk2` (no small kids): 16.5261268
- If we removed 82 cases, why isn't the difference bigger?

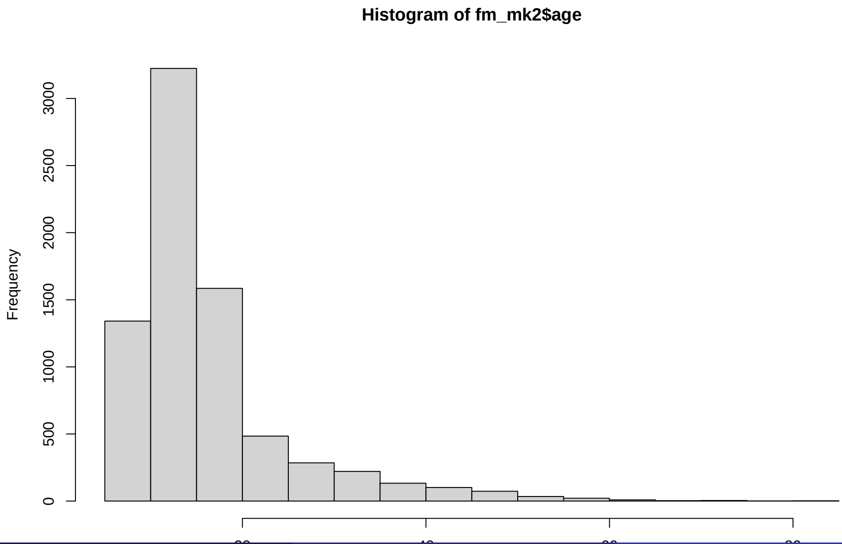
What is a Mean?

- One of a number of measures of central tendency
  - Values that are in the "middle"
  - Popular values
- The arithmetic center of a distribution
- Sensitive to extreme values
- Classic definition of the word average

$$\mu_x = \frac{\sum_{i=1}^{n} x_i}{n}$$

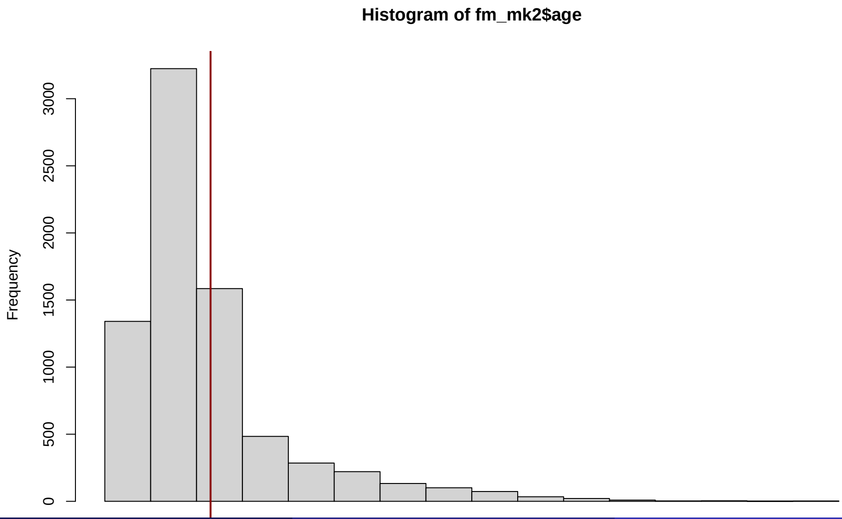
Visualizing the Mean of a Distribution - Histogram

```r
hist(fm_mk2$age)
```



Histogram with the Mean Inserted

```r
hist(fm_mk2$age)
abline(v = mean(fm_mk2$age), col= "darkred", lwd = 2)
```



Section 6

Data Visualization - Graphs

Simple Histogram

- Didn't give us much information
- Terrible Presentation

Grammar of Graphics - `ggplot2`

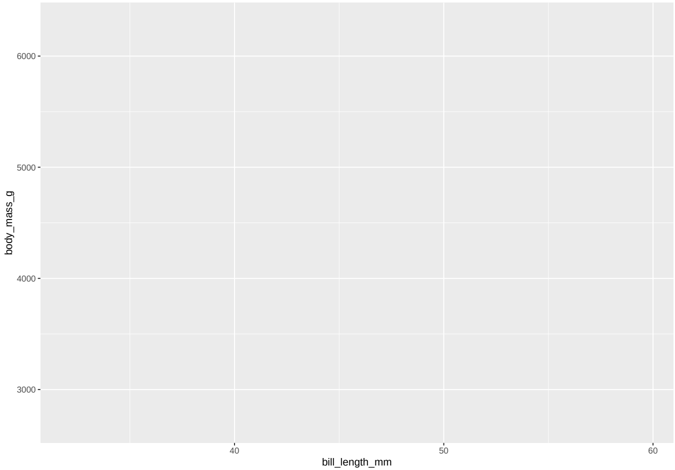
- A system to build graphs (that communicate much better)
- One of Hadley Wickham's first products
- Build your graphs layer by layer
- Begin by specifying a data set `penguin`
  - Variables `bill_length_mm` `body_mass_g`

```
## Rows: 333
## Columns: 3
## $ bill_length_mm <dbl> 39.1, 39.5, 40.3, 36.7, 39.3, 38.9, 39.2, 41.1, 38.6...
## $ body_mass_g    <dbl> 3750, 3800, 3250, 3450, 3650, 3625, 4675, 3200, 3800...
## $ sex            <chr> "male", "female", "female", "female", "male", "femal...
```

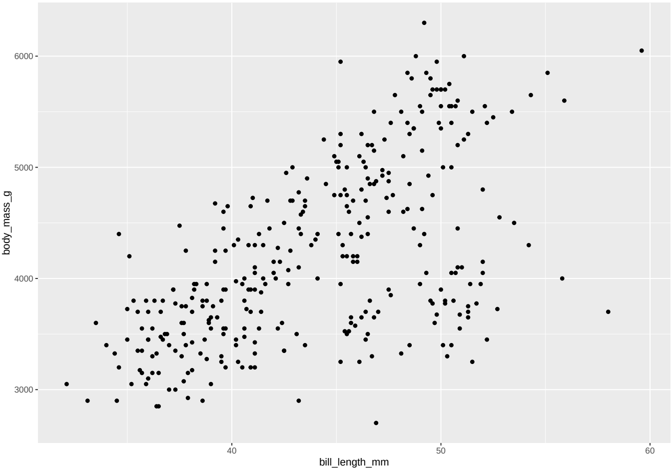
```r
ggplot()
```

```r
ggplot(data = pd)
```

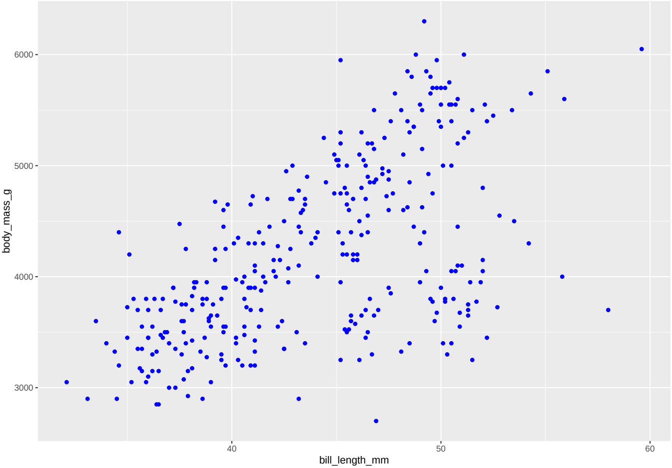
```r
ggplot(data = pd, aes(x = bill_length_mm, body_mass_g ))
```



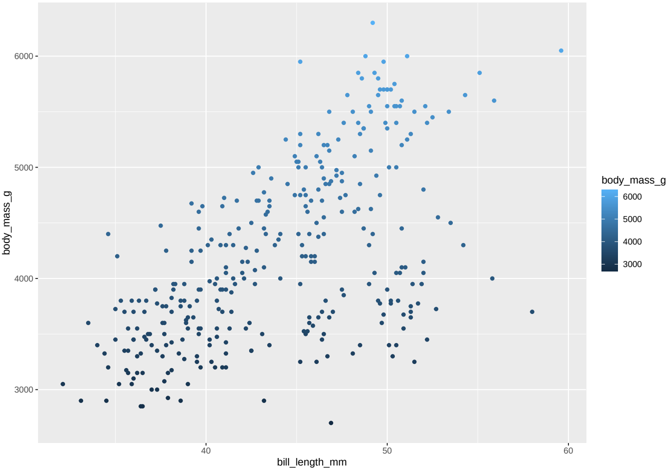
```r
ggplot(data = pd, aes(x = bill_length_mm, body_mass_g )) +
  geom_point()
```



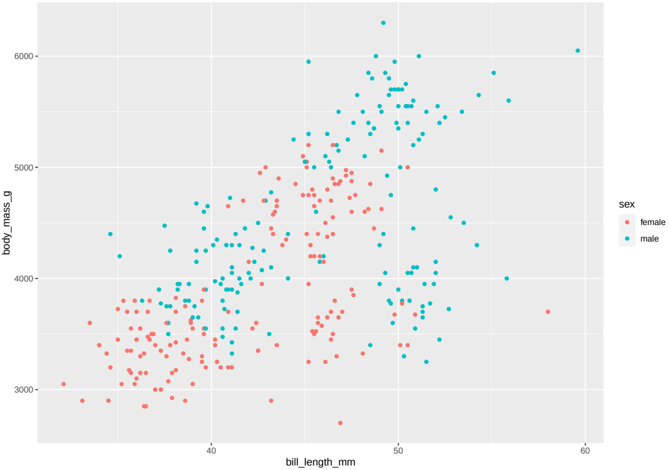
```r
ggplot(data = pd, aes(x = bill_length_mm, body_mass_g )) +
  geom_point(color = "blue")
```



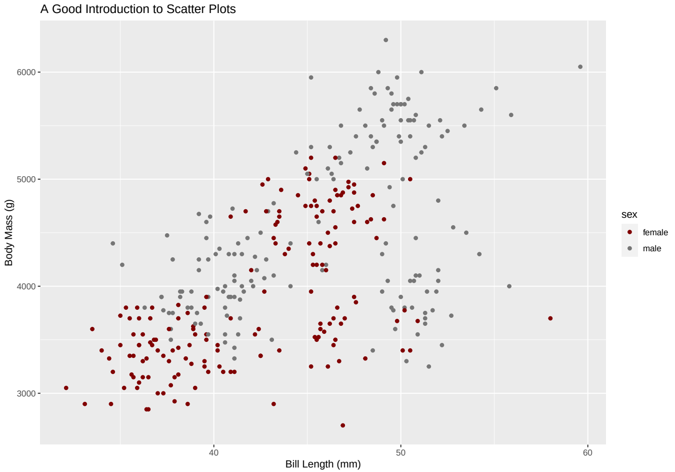
```r
ggplot(data = pd, aes(x = bill_length_mm, body_mass_g, color = body_mass_g)) +
  geom_point()
```



```r
ggplot(data = pd, aes(x = bill_length_mm, body_mass_g, color = sex )) +
  geom_point()
```



```r
ggplot(data = pd, aes(x = bill_length_mm, body_mass_g, color = sex)) +
  geom_point() +
  labs(title = "A Good Introduction to Scatter Plots", x = "Bill Length (mm)",
       y = "Body Mass (g)") +
  scale_colour_manual(values = c("#800000FF", "#767676FF"))
```



Resources for ggplot

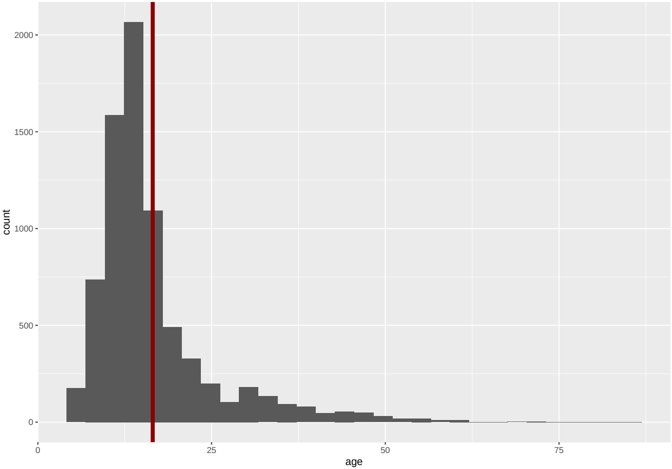
- Winston Chang, **R Graphics Cookbook**, 2Ed., https://r-graphics.org
- Kieran Healy, **Data Visualization: A Practical Introduction**, https://socviz.co
- https://r-graph_gallery.com - examples of many types of graphs with explanations and code
- ggplot cheat sheet

Histogram of age

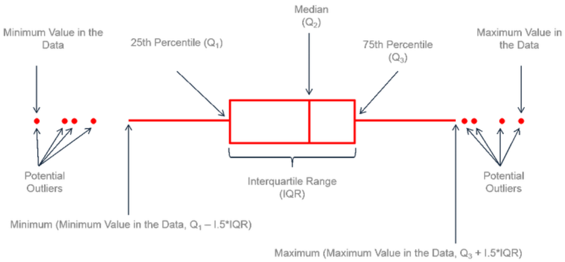
Live Coding

Histogram of age

```
avg_age <- mean(fm_mk2$age)
ggplot(data = fm_mk2, aes(x = age)) +
  geom_histogram(bins = 30) +
  geom_vline(xintercept = avg_age, colour = "darkred", size = 2)
```



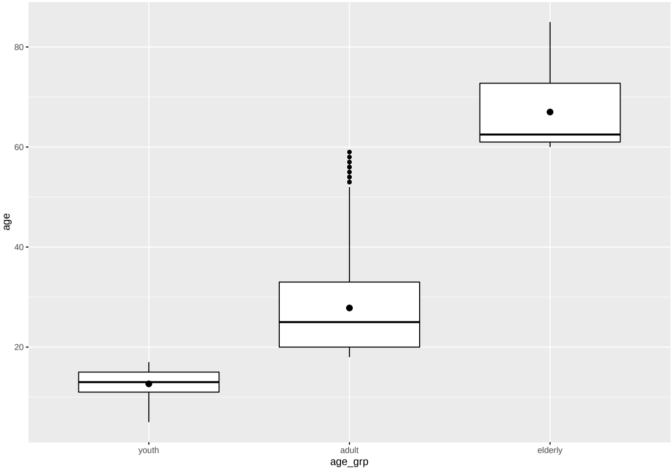
Another Graph that Shows Distribution Clearly - Boxplot



- source: https://r-graph-gallery.com

Boxplot with Our Data

```
fm_mk2 %>%  ggpubr::ggboxplot(x = "age_grp",
                              y = "age",
                              add = "mean",
                              ggtheme = theme_gray())
```



John Tukey on Visualization

The simple graph has brought more information to the data analyst's mind than any other device.

O gráfico simples trouxe mais informações à mente do analista dos dados do que qualquer outro dispositivo.

Section 7

Back to Numbers

Median

- The middle value of the variable
  - Put the values in order from top to bottom
  - Select
    - if odd number: the middle value
    - if even number: the mean of the two middle numbers
  - The line in the middle of a boxplot box
- Robust even when you have extreme values (outliers)
- Function in R: `median()`

Mean vs. Median

- Example data: 10 numbers, 1 far out

```
set.seed(42)
x10 <- c(rnorm(9, mean = 100, sd = 1), 1000)
x10
```

```
##  [1]  101.37096   99.43530  100.36313  100.63286  100.40427   99.89388
##  [7]  101.51152   99.90534  102.01842 1000.00000
```

```
mean(x10)
```

```
## [1] 190.5536
```

```
median(x10)
```

```
## [1] 100.5186
```

Remove the Outlier; See the Change

```
set.seed(42)
x9 <- x10[1:9]
x9
```

```
## [1] 101.37096  99.43530 100.36313 100.63286 100.40427  99.89388 101.51152
## [8]  99.90534 102.01842
```

```
mean(x9)
```

```
## [1] 100.6151
```

```
median(x9)
```

```
## [1] 100.4043
```

Summary of the Change

```
## # A tibble: 2 x 5
##   vector     n max_value  mean median
##   <chr>  <int>     <dbl> <dbl>  <dbl>
## 1 x10       10      1000  191.   101.
## 2 x9         9      102.  101.   100.
```

Mode

- Most frequently occurring value in a variable
- Useful if you want to find a value that occurs too frequently
- Otherwise, not really useful

Section 8

Measures of Dispersion

Range

- The simplest
- Largest value - smallest value
- `max(x10) - min(x10)`
- Not very useful

```r
max(x10) - min(x10)
```

```
## [1] 900.5647
```

Interquartile Range (IQR)

- Difference between the 25th and 75th percentiles (*quantiles*)
  - Ends of the boxplot box
- The middle 50% of values fall in IQR
- Can get values for quantiles with `quantile()` function
- Can get IQR directly with `IQR()`

```
quantile(x10, probs = c(.25, .75))
```

```
##      25%      75%
## 100.0198 101.4764
```

```
IQR(x10)
```

```
## [1] 1.456593
```

Mean/Median Absolute Deviation (MAD)

- What is *typical* deviation from a given reference point
  - Mean
  - Median (usually used)
- Name describes process of calculating it
  - Determine deviation of every point from reference
  - Take the absolute value of that deviation
  - Find the mean of the absolute deviations

Calculating MAD

- Use first 5 points from `x10` vector

```
val <- x10[1:5]
dev <- val - mean(val) # note vectorization
abs_dev <- abs(dev)
tibble(val = val,
       dev = dev,
       abs_dev = abs_dev) %>%
  knitr::kable()
```

val	dev	abs_dev
101.3710	0.9296545	0.9296545
99.4353	-1.0060021	1.0060021
100.3631	-0.0781755	0.0781755
100.6329	0.1915587	0.1915587
100.4043	-0.0370356	0.0370356

```
paste("Median of 5 values = ", median(val))
```

```
## [1] "Median of 5 values =  100.404268323141"
```

```
paste("Mean Absolute Deviation = ", mean(abs_dev))
```

```
## [1] "Mean Absolute Deviation =  0.448485282412688"
```

Variance & Standard Deviation

- Measures dispersion around the mean
- Idea - measure difference between the mean and each point
- If we do that, what is result?

```
diff_val <- val - mean(val)
diff_val
```

```
## [1]  0.92965452 -1.00600209 -0.07817551  0.19155868 -0.03703560
```

```
round(sum(diff_val), 2)
```

```
## [1] 0
```

Make It More Useful - Square the Differences

- All differences will now be positive

```
diff_val_sq <- diff_val^2
diff_val_sq
```

```
## [1] 0.864257534 1.012040214 0.006111411 0.036694729 0.001371636
```

```
sum(diff_val_sq)
```

```
## [1] 1.920476
```

Look at This in Table Form (like MAD)

```r
sq_dev <- dev^2
tibble(val = val,
       dev = dev,
       sq_dev = sq_dev) %>%
  knitr::kable()
```

val	dev	sq_dev
101.3710	0.9296545	0.8642575
99.4353	-1.0060021	1.0120402
100.3631	-0.0781755	0.0061114
100.6329	0.1915587	0.0366947
100.4043	-0.0370356	0.0013716

```r
paste("Mean of 5 values = ", mean(val))
```

```
## [1] "Mean of 5 values =  100.441303923038"
```

```r
paste("Variance = ", mean(sq_dev))
```

```
## [1] "Variance =  0.384095104622191"
```

Variance

- The sum of the squares ("SST"-`sq_dev`)
  - Useful when we get to regression
- However, what use is 1.92 to interpret the set of numbers it comes from?
- It is the total across all the 5 values
- We want something we can compare to each value
- We can divide the sum by the `n`
  - Gives us the mean of the squared differences

```
sum(diff_val_sq)/length(val)
```

```
## [1] 0.3840951
```

R has a Function for Variance (`var()`)

- `var()` skips all the table calculations

```r
var(val)
```

```
## [1] 0.4801189
```

Why Are the Values Not the Same?????

- R (and all statistical programs) use a different denominator
  - $N - 1$ instead of N
- Our calculation (with N) was for a full population
- $N - 1$ is used when calculating the sample variance

```r
var(val) # sample
```

```
## [1] 0.4801189
```

```r
sum(diff_val_sq)/(length(val) - 1) # sample
```

```
## [1] 0.4801189
```

```r
sum(diff_val_sq)/length(val) # population
```

```
## [1] 0.3840951
```

Samples and Populations

- Use samples to estimate the parameters of populations
  - Population mean

$$\mu_x = \frac{\sum x_i}{n}$$

  - Sample mean

$$\bar{x} = \frac{\sum x_i}{n}$$

Variance Formulas

- Population variance

$$\sigma_x^2 = \frac{\sum (x_i - \mu_x)^2}{n}$$

- Sample variance

$$s_x^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$$

Why the Difference?

- Using samples to say something about populations
  - Statistical Inference
- Because the sample is a stand-in for a full population
  - Need to make the variance somewhat larger
  - Compensates for uncertainty about the ability of sample to describe population
  - Also called degrees of freedom
  - Will do more with this later
- For now, use the $n - 1$ formulation
  - `var()` and `sd()` functions

Does Difference Always Exist?

- Yes, but becomes very small when n is very large
- Look at `fm_mk2$age` with 7521 cases
- Because sample size begins to approximate whole population

```
age <- fm_mk2$age
paste("Sample Variance = ", round(var(age), 5))
```

```
## [1] "Sample Variance =  78.41265"
```

```
paste("Population Variance = ", round(sum((age - mean(age))^2/length(age)), 5))
```

```
## [1] "Population Variance =  78.40222"
```

What Are Units of This Square?

- It is in squared units
  - Like areas in comparison to length
  - A room that is 4 meters x 4 meters has an area of 16 sq. m.
- Need to reduce it back to original units
  - Take the square root of variance to get the original scale
  - Standard deviation
  - Function `sd()`

`sd()` of Values We Calculated

- age
  - Variance: var(age) $=$ 78.41265 squared years
  - Standard Deviation: sd(age) $=$ 8.85509 years
- Problem of units solved

Mean and Standard Deviation as Parameters of Normal Distribution

- In next unit, we will see the normal distribution and how to use it to help with inference
- Its formula has 2 parameters: mean and standard deviation
  - Any normal distribution can be described by its mean and standard deviation

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Mean $\pm$ Standard Deviation

- Common notation to show how far around mean 1 standard deviation (s) extends
- Age example
  - 16.5261268 $\pm$ 8.8550917
- Rule of Thumb (Aproximação): 68% of the variance will be included in the range from $\bar{x} - s$ to $\bar{x} + s$
  - For age, 68% of values are between 7.6710352 and 25.3812185
- If you use 2 standard deviations $\bar{x} \pm 2sd$, you will include 95% of all the data
  - For age, 95% of values are between -1.1840565 and 34.2363102

Coefficient of Variation (CV)

- Another way to look at the spread of the variable

$$CV = \frac{s}{\bar{x}}$$

- Shows relative size of mean and sd
- A CV greater than 1 spells trouble that there is too much variance to make useful inferences
- For age, CV $= 0.5358238$

Look Again at summarytools::descr()

```r
summarytools::descr(fm_mk2$age)
```

```
## Descriptive Statistics
## fm_mk2$age
## N: 7521
##
##                           age
## ----------------- ---------
##              Mean     16.53
##           Std.Dev      8.86
##               Min      5.00
##                Q1     12.00
##            Median     14.00
##                Q3     17.00
##               Max     85.00
##               MAD      4.45
##               IQR      5.00
##                CV      0.54
##          Skewness      2.28
##       SE.Skewness      0.03
##          Kurtosis      6.78
##           N.Valid   7521.00
##         Pct.Valid    100.00
```

Look at age for Each sex Separately

- Use of `group_by()` variable of `dplyr`
  - Permits downstream calculations to be done on each gender

```
fm_mk2 %>%
  group_by(sex) %>%
  summarytools::descr(age)
```

```
## Descriptive Statistics
## age by sex
## Data Frame: fm_mk2
## N: 2381
##
##                       Female      Male
## ----------------- --------- ---------
##              Mean     15.37     17.06
##           Std.Dev      7.24      9.46
##               Min      5.00      5.00
##                Q1     12.00     11.00
##            Median     14.00     14.00
##                Q3     16.00     19.00
##               Max     84.00     85.00
##               MAD      2.97      4.45
##               IQR      4.00      8.00
##                CV      0.47      0.55
##          Skewness      3.12      2.02
##       SE.Skewness      0.05      0.03
##          Kurtosis     14.57      5.00
##           N.Valid   2381.00   5140.00
##         Pct.Valid    100.00    100.00
```

Another Example of group_by()

- Ages for each age_grp

```
fm_mk2 %>%
  group_by(age_grp) %>%
  summarytools::descr(age)
```

```
## Descriptive Statistics
## age by age_grp
## Data Frame: fm_mk2
## N: 5658
##
##                         youth     adult   elderly
## ----------------- --------- --------- ---------
##              Mean     12.65     27.82     67.00
##           Std.Dev      2.95      9.42      7.93
##               Min      5.00     18.00     60.00
##                Q1     11.00     20.00     61.00
##            Median     13.00     25.00     62.50
##                Q3     15.00     33.00     73.00
##               Max     17.00     59.00     85.00
##               MAD      2.97      8.90      3.71
##               IQR      4.00     13.00     11.75
##                CV      0.23      0.34      0.12
##          Skewness     -0.47      1.07      0.92
##       SE.Skewness      0.03      0.06      0.49
##          Kurtosis     -0.48      0.42     -0.43
##           N.Valid   5658.00   1841.00     22.00
##         Pct.Valid    100.00    100.00    100.00
```

Homework 1

- GitHub
- 3 Files
  - Lição de Casa 1
  - trplasma.csv
  - pac_demo.xlsx
- October 2 – 2 weeks