MAD – Data Analysis & Biostatistics in R

Inference & Regression

James R. Hunter, Ph.D.

DIPA, EPM, UNIFESP

2 October 2020

MAD

Section 1

Inference with Categorical Variables

Focus on Categorical Variables Only

- Case of categorical x numeric variables
  - Covered last week
  - Case of penguin body mass x species
  - Use of t-test
- First, single categorical variable
  - Example from Psychology
  - D. Navarro, LSR, v0.6
- Second, two-way table of categorical variables
  - From SEADE comorbidity tables

Single Variable – Goodness of Fit Test

- Experiment to see if people can choose randomly
- 200 students asked to pick a card from a virtual deck (*baralho*) at random
- Then asked to pick a 2nd card from the virtual deck
- We are only interested in the suits (*naipes*) they chose
- Initial focus on 1st card they chose (`choice_1`)
- If they had chosen purely randomly, all the probabilities should be equal

Look at the data

- Get a table of the observed values $(O_1 ... O_{200})$

```
naipes <- readRDS(here("naipes.rds"))
head(naipes) # View initial cases
```

```
##      id choice_1 choice_2
## 1 subj1  espadas     paus
## 2 subj2    ouros     paus
## 3 subj3    copas     paus
## 4 subj4  espadas     paus
## 5 subj5    copas  espadas
## 6 subj6     paus    copas
```

```
(observed <- table(naipes$choice_1))
```

```
## 
##     paus   ouros   copas espadas 
##       35      51      64      50
```

Hypothesis Test

- Null hypothesis

$$H_0: \text{ All four suits chosen with equal probability}$$

- Another, more mathematical, format

$$H_0 : P = (0.25, 0.25, 0.25, 0.25)$$

- Alternative hypothesis

$$H_0: \text{ At least one suit has a probability that is not 0.25}$$

$$H_0 : P \neq (0.25, 0.25, 0.25, 0.25)$$

Probabilities and Expected Values

- We have a set of probabilities we want to test $P = (0.25, 0.25, 0.25, 0.25)$
  - Store them in a vector

```
(probs <- c(copas = 0.25, ouros = 0.25,
            paus = 0.25, espadas = 0.25))
```

```
##   copas   ouros    paus espadas 
##    0.25    0.25    0.25    0.25
```

Calculating Goodness of Fit

- *Goodness of fit* means how close are the observed data to the null hypothesis
- Need to translate the probabilities of null hypothesis to an expected frequency (E) for each suit (N = 200)

$$E_i = N \times P_i$$

- Then calculate how far the observed value (O_i) for each case differs from the expected value (E_i): $(O_i - E_i)$

Do This in R

```r
n <- 200
expected <- n * probs
diff <- observed - expected
df <- tibble(suit = names(observed),
             expected,
             observed,
             diff)
df
```

```
## # A tibble: 4 x 4
##   suit    expected observed diff
##   <chr>      <dbl> <table>  <table>
## 1 paus          50 35       -15
## 2 ouros         50 51         1
## 3 copas         50 64        14
## 4 espadas       50 50         0
```

Not Quite There

- Remember when we measured difference from mean in variance
  - Difference always equals 0
  - Differences: -15, 1, 14, 0
  - Sum of Differences: 0
- Avoid that consequence by *squaring* the difference

```
df$diff_sq <- df$diff^2
df
```

```
## # A tibble: 4 x 5
##   suit    expected observed diff    diff_sq
##   <chr>      <dbl> <table>  <table> <table>
## 1 paus          50 35       -15     225
## 2 ouros         50 51         1       1
## 3 copas         50 64        14     196
## 4 espadas       50 50         0       0
```

Goodness of Fit Statistic - X^2

- Next step: divide differences squared by *expected*
- Finally: sum the adjusted squared differences
- k = number of categories (naipes)

$$X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

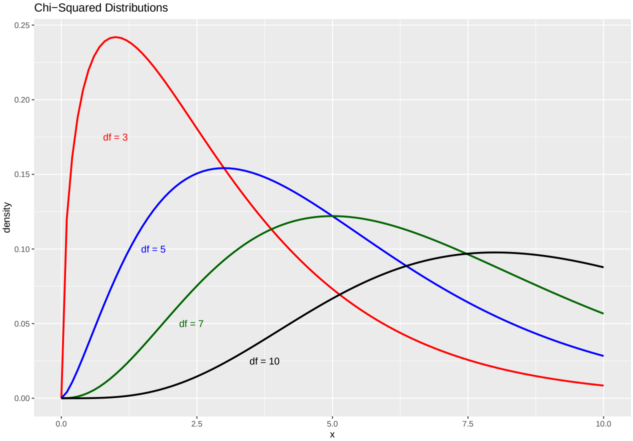
- If X^2 is small, unlikely to reject null hypothesis - Differences will be too small to justify this

```r
(X_sq <- sum((observed - expected)^2/expected))
```

```
## [1] 8.44
```

Sampling Distribution of X^2

- This statistic (X^2) follows a chi-squared distribution
  - *Qui-quadrado*
  - χ^2
  - with $(k-1)$ degrees of freedom



What is Going On – Statistical Sidebar

- χ^2 Distribution is quite common
- If you have multiple variables that are normally distributed
  - Square their values
  - Sum the squares
- The result will have a χ^2 distribution

Mathematical Demonstration

- 3 normally distributed variables

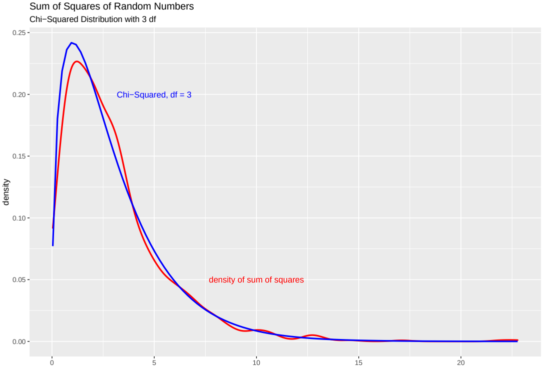
```
norm1 <- rnorm(n = 1000)
norm2 <- rnorm(n = 1000)
norm3 <- rnorm(n = 1000)
```

- Sum the squares of the values

```
final_dist <- norm1^2 + norm2^2 + norm3^2
```

Graph of Results

```
tibble(final_dist) %>%
  ggplot(aes(x = final_dist)) +
  geom_density(color = "red", size = 1) +
  stat_function(fun = dchisq, args = list(df = 3), color = "blue", size = 1) +
  labs(title= "Sum of Squares of Random Numbers",
       subtitle = "Chi-Squared Distribution with 3 df",
       x = "") +
  annotate("text", x = 10, y = .05, label = "density of sum of squares", color = "red") +
  annotate("text", x = 5, y = .2, label = "Chi-Squared, df = 3", color = "blue")
```



Complete Hypothesis Test

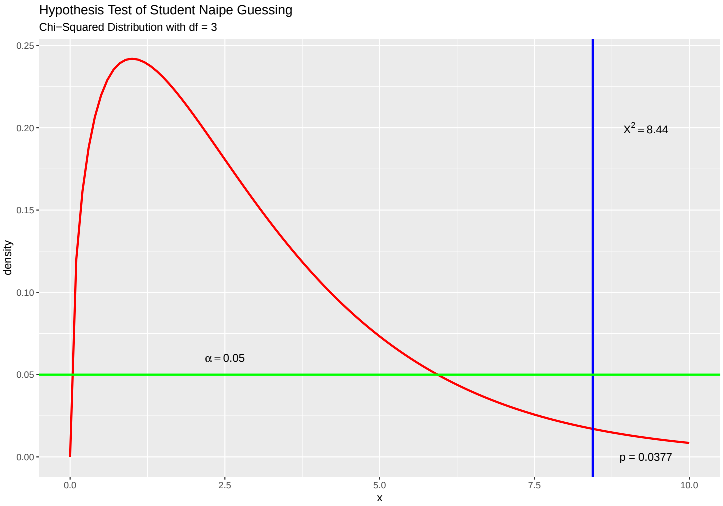
- Compare our X^2 to where the 95th percentile of the χ^2 distribution with 3 degrees of freedom (4 naipes - 1)
- `qchisq(p = .95, df = 3)` $= 7.8147279$
  - Quantile function for the chi-squared distribution
- If our X^2 is larger than 7.8147279, we should reject the null hypothesis
- $X^2 = 8.44$; $\chi^2 = 7.8147279$
- $\therefore$ we should reject null hypothesis
  - People cannot guess randomly
- Exact probability can be shown with `pchisq()`
  - Takes the quantile (X^2) as its `p` argument

```
pchisq(q = X_sq, df = 3, lower.tail = FALSE) ## only calculat
```

```
## [1] 0.03774185
```

```
1-pchisq(q = X_sq, df = 3) ## calculate 1 - lower tail
```

```
## [1] 0.03774185
```


Hypothesis Test of Student Naipe Guessing

Chi−Squared Distribution with df = 3

Running the Test in R

- Doesn't get easier

```
chisq.test(x = observed)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 8.44, df = 3, p-value = 0.03774
```

Section 2

χ^2 Test of Independence

Return to SEADE Comorbidity Data

- Question: Are people with more than 60 years more likely to die from COVID-19?
- Exists a numeric age variable
- Create a categorical variable for ages: `age_group`
  - `elderly`: 60 years old or older
  - `others` : less than 60 years
    - Note: only 1 child in sample (newborn)
- This is a 2 x 2 test, but could have any dimensions
  - **All** cells in test *should* have at least **5** cases

Table

```r
comorbid <- readRDS(here("seade_comorbid_sample.rds")) %>%
  mutate(age_group = ifelse(age < 60, "other", "elderly"))
summarytools::ctable(comorbid$death, comorbid$age_group, prop = "c")
```

```
## Cross-Tabulation, Column Proportions
## death * age_group
## Data Frame: comorbid
##
## ------- ---------- ------------- ------------- -------------
##           age_group       elderly         other         Total
##   death
##   FALSE                107 ( 59.4%)   91 ( 75.8%)  198 ( 66.0%)
##    TRUE                 73 ( 40.6%)   29 ( 24.2%)  102 ( 34.0%)
##   Total                180 (100.0%)  120 (100.0%)  300 (100.0%)
## ------- ---------- ------------- ------------- -------------
```

Hypotheses

- H_0: elderly and others have equal chance of dying from COVID-19
  - H_0: p(`elderly` & `death` = TRUE) = p(`others` & `death` = TRUE) $= P_1$
- H_1: elderly and others have different chances of dying from COVID-19
  - H_1: p(`elderly` & `death` = TRUE) = p(`others` & `death` = TRUE) $\neq P_1$

Calculate Expected Values

- Here a little more tricky
    - Must estimate the probability from the data
- Multiply the column total x the probability of the row
    - Probability of row is the proportion for that row of the total
    - $\hat{P}_i = \frac{R_i}{N}$
- Estimated values = product of margins * N

$$E_{ij} = \frac{R_i \times C_j}{N}$$

```
## Cross-Tabulation, Column Proportions
## death * age_group
## Data Frame: comorbid
##
## ------- ----------- -------------- -------------- --------------
##           age_group        elderly          other          Total
##   death
##   FALSE                107 ( 59.4%)    91 ( 75.8%)   198 ( 66.0%)
##    TRUE                 73 ( 40.6%)    29 ( 24.2%)   102 ( 34.0%)
##   Total                180 (100.0%)   120 (100.0%)   300 (100.0%)
## ------- ----------- -------------- -------------- --------------
```

Estimated Values

```
estim <- data.frame(death = integer(length = 2),
                    age_group = integer(length = 2))
ct <- tab_comorb$cross_table
n <- nrow(comorbid)
for(i in 1:2){
  for(j in 1:2){
    estim[i,j] <- sum(ct[i,1:2]) * sum(ct[1:2,j]) / n
  }
}
rownames(estim) <- c("FALSE", "TRUE")
colnames(estim) <- c("elderly", "other")
estim
```

```
##       elderly other
## FALSE   118.8  79.2
## TRUE     61.2  40.8
```

X^2 Statistic for Independence

- Same as for Goodness of Fit
- Need double summation over rows and columns both
- Need to adjust degrees of freedom for both dimensions
  - $(r-1)(c-1)$

$$X^2_{df} = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(E_{ij} - Oij)}{Eij}$$

Executing Test in R

- Three ways with different amounts of information
- 1st: `summarytools::ctable()`

```
summarytools::ctable(comorbid$death, comorbid$age_group, prop = "c", chisq = TRUE)
```

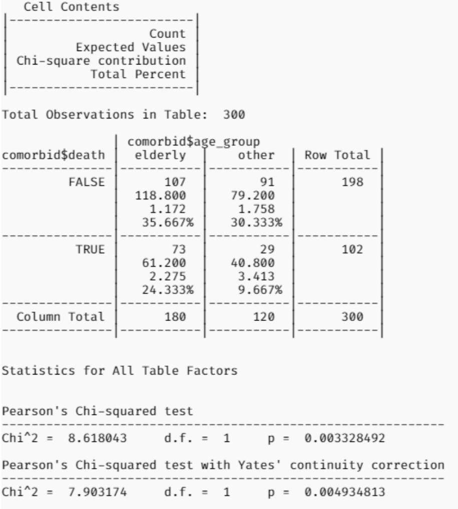
```
## Cross-Tabulation, Column Proportions
## death * age_group
## Data Frame: comorbid
##
##
## ------- ---------- ------------- ------------- -------------
##           age_group       elderly         other         Total
##   death
##   FALSE               107 ( 59.4%)   91 ( 75.8%)  198 ( 66.0%)
##    TRUE                73 ( 40.6%)   29 ( 24.2%)  102 ( 34.0%)
##   Total               180 (100.0%)  120 (100.0%)  300 (100.0%)
## ------- ---------- ------------- ------------- -------------
##
## ----------------------------
##  Chi.squared   df   p.value
## ------------- ---- ---------
##     7.903      1    0.0049
## ----------------------------
```

- Simple presentation

Results with gmodels::CrossTable()

- Shows results in format similar to SPSS and SAS

```r
gmodels::CrossTable(comorbid$death, comorbid$age_group, expected = TRUE, chisq = TRUE,
                    format = "SPSS")
```



```
   Cell Contents
|-------------------------|
|                   Count |
|         Expected Values |
| Chi-square contribution |
|           Total Percent |
|-------------------------|

Total Observations in Table:  300

               | comorbid$age_group
comorbid$death |  elderly  |     other  | Row Total |
---------------|-----------|-----------|-----------|
         FALSE |      107  |       91  |      198  |
               |  118.800  |   79.200  |           |
               |    1.172  |    1.758  |           |
               |  35.667%  |  30.333%  |           |
---------------|-----------|-----------|-----------|
          TRUE |       73  |       29  |      102  |
               |   61.200  |   40.800  |           |
               |    2.275  |    3.413  |           |
               |  24.333%  |   9.667%  |           |
---------------|-----------|-----------|-----------|
  Column Total |      180  |      120  |      300  |
---------------|-----------|-----------|-----------|


Statistics for All Table Factors


Pearson's Chi-squared test
------------------------------------------------------------
Chi^2 =  8.618043     d.f. =  1     p =  0.003328492

Pearson's Chi-squared test with Yates' continuity correction
------------------------------------------------------------
Chi^2 =  7.903174     d.f. =  1     p =  0.004934813
```

Results with `lsr::associationTest()`

- Adjunct to **Learning Statistics with R** book
- Uses formula interface

```
comorbid <- comorbid %>%
  mutate(age_group = factor(age_group),
         death = factor(death))
lsr::associationTest(~ age_group + death, data = comorbid)
```

```
##
##      Chi-square test of categorical association
##
## Variables:   age_group, death
##
## Hypotheses:
##    null:        variables are independent of one another
##    alternative: some contingency exists between variables
##
## Observed contingency table:
##          death
## age_group FALSE TRUE
##   elderly   107   73
##   other      91   29
##
## Expected contingency table under the null hypothesis:
##          death
## age_group FALSE TRUE
##   elderly 118.8 61.2
##   other    79.2 40.8
##
## Test results:
##    X-squared statistic:  7.903
##    degrees of freedom:  1
##    p-value:  0.005
##
## Other information:
##    estimated effect size (Cramer's v):  0.162
##    Yates' continuity correction has been applied
```

Notes on χ^2 Tests: Yates Continuity Corretion and Cramér's V

- Yates continuity correction
  - When you have a 2 x 2 table, the X^2 statistic tends to be too big
  - Yates proposed a *hack* that subtracts 0.5 from all the deviations (before squaring)

$$X^2 = \sum_{i=1} \frac{(|O_i - E_i| - 0.5)^2}{E_i}$$

- Cramér's V
  - Measures correlation between two categorical variables
  - Also can be called "effect size"
  - Varies between 0 and 1
  - Our 0.162 suggests overall association between variables not very strong
    - Although clearly not 0
    - Because they are not independent

Section 3

Machine Learning Models

Flavors *Machine Learning*

- If there is a dependent variable
  - **Supervised**
    - Supervised because the model's results can be evaluated in terms of the dependent variable
  - 2 subtypes
    - **Classification** - Put each case in a group based on values of the independent variables
    - **Regression** - Determine a dependent value based on a combination of the independent variables
- If there is *not* a dependent variable
  - **Unsupervised**
    - Explore the structure among the cases and try to group them in a *cluster* of cases
    - **Cluster Analysis**

Section 4

Simple Linear Regression

Regression – History

- Term comes from eugenics (*eugenismo*) proposed by Sir Francis Galton.
- Studied heights on individuals within families
- Observed that children of
  - Children of tall parents tended to be shorter than the parents
  - Children of shorter parents tended to be taller than the parents
- Called this trend **regression to the mean**

Method of Least Squares

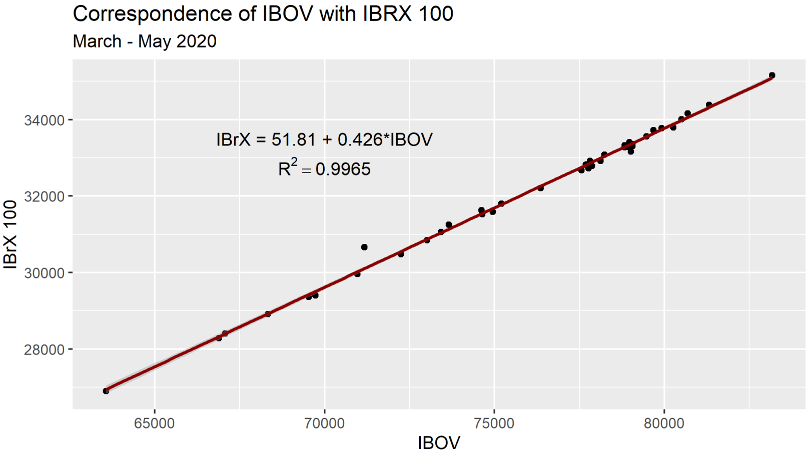
- Solve problems of regression with the *Least Squares* method
- Invented by Carl Friedrich Gauss (1777 - 1855)
- Method minimizes the differences between predicted linear values and the values based on the data
- Achieves the best relation between the real dependent variable and the predicted values of the variable
- In this course, focus on linear model forms
  - Many other types of regression exist

Purpose

Predict a result on a dependent variable based on one or more indepedent variables

- One – *simple* linear regression
- More – *multiple* linear regression

Visualização de Regressão



Straight Line

$$y = \beta_0 + \beta_1 x$$

- β_1 = **Slope** of the line
- β_0 = **Intercept** of the line (where it crosses the y axis)
- Two parameters of regression
- Optimizing these parameters, Least Squares finds the straight line
- *Best* predicts the value of the dependent variable (y) based on the value of the independent variable (x)

Does “Best” Mean “Good”?

- Despite being the best way to predict y,
    - Possible that it does **not** describe y well
- **Good** depends on the data
- **Best** depends on the algorithm

Regression Equation

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- $Y_i =$ value of the dependent variable
- $\beta_0 =$ intercept
- $\beta_1 =$ slope of the regression line
- $X_i =$ value of the independent variable
- $\epsilon_i =$ error term for each case

Regression Equation - Estimation

$$\hat{Y}_i = b_0 + b_1 X_i + e_i$$

- $\hat{Y}_i$ = value of the dependent variable (estimated)
- b_0 = intercept (estimated)
- b_1 = slope of the regression line (estimated)
- X_i = value of the independent variable
- e_i = error term for each case

“Error” Term (ϵ)

- Also called **residual**
- Responsible for variability in y the the line cannot explain
- Does not mean “wrong”
- Only means “difference from a mean”
- Similar to what we saw with hypothesis tests

Least Squares

- Makes the calculation that minimizes the *error sum of squares*
- Errors = residuals = differences between the *observed* value and the *expected* value

$$min \sum (y_i - \hat{y}_i)^2$$

- y_i = observed value of the dependent variable
- $\hat{y}_i$ = estimated value of the dependent variable

Example

- Data set of Galton about height in families
- Question is if children are taller or shorter than their parents
- He measured 898 sons/daughters in 197 families
- Original data records are in University College, London (UCL)

Variables

```r
galton <- readRDS(here::here("galton.rds"))
str(galton)
```

```
## 'data.frame':    898 obs. of  6 variables:
##  $ family: Factor w/ 197 levels "1","10","100",..: 1 1 1 1 108 108 108 108 123 1
##  $ father: num  78.5 78.5 78.5 78.5 75.5 75.5 75.5 75.5 75 75 ...
##  $ mother: num  67 67 67 67 66.5 66.5 66.5 66.5 64 64 ...
##  $ sex   : Factor w/ 2 levels "F","M": 2 1 1 1 2 2 1 1 2 1 ...
##  $ height: num  73.2 69.2 69 69 73.5 72.5 65.5 65.5 71 68 ...
##  $ nkids : int  4 4 4 4 4 4 4 4 2 2 ...
```

- `height`, `father`, `mother` – all are height in inches

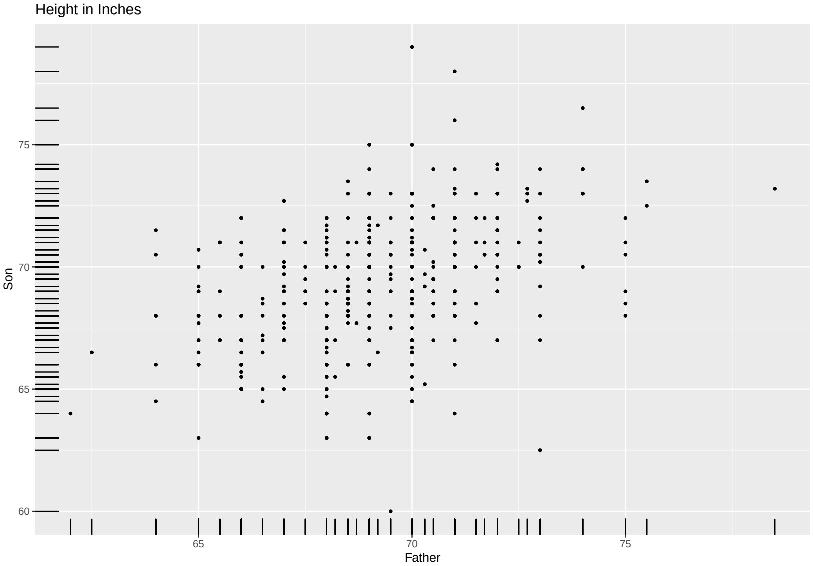
Focus on Fathers and Sons

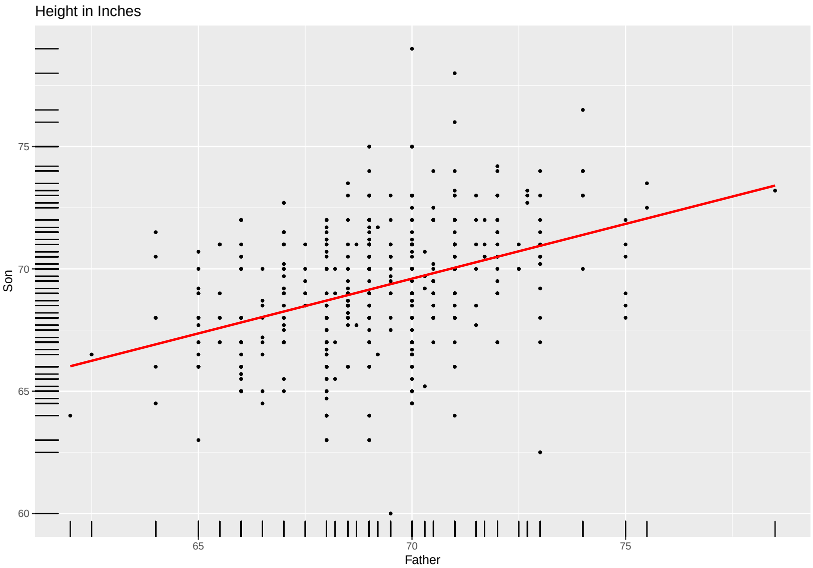
```r
boys <- galton %>%
  filter(sex == "M") %>%
  select(-family, -mother, -sex, -nkids)
glimpse(boys)
```

```
## Rows: 465
## Columns: 2
## $ father <dbl> 78.5, 75.5, 75.5, 75.0, 75.0, 75.0, 75.0, 75.0, 75.0, 74.0, ...
## $ height <dbl> 73.2, 73.5, 72.5, 71.0, 70.5, 68.5, 72.0, 69.0, 68.0, 76.5, ...
```

- `father` is the independent variable
- `height` is the dependent variable
- We want to see if the height of the father predicts the height of the son

Father/Son – Scatterplot




Height in Inches
Son
Father
75
70
65
60
65
70
75

What Have We Learned from the Scatterplot?

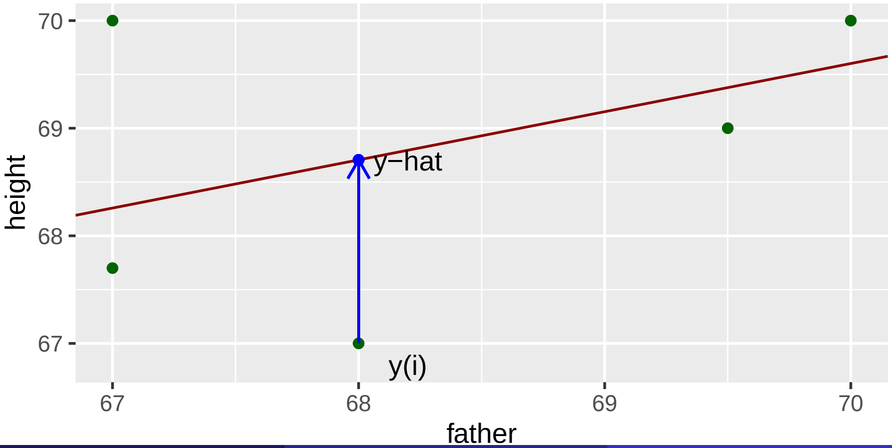
- **Seems** that taller the fathers, taller the sons
- Descriptive statistics of the 2 variables
  - And, correlation

```
## Descriptive Statistics
## boys
## N: 465
##
##                   Mean   Std.Dev     Min       Q1   Median       Q3     Max    IQR     CV
## ------------ ------- --------- ------- ------- -------- ------- ------- ------ ------
##       father   69.17      2.30   62.00   68.00    69.00   70.50   78.50   2.50   0.03
##       height   69.23      2.63   60.00   67.50    69.20   71.00   79.00   3.50   0.04

## [1] "Correlation Coefficient: 0.391"
```

How Do We Calculate the Regression Line?

- A line that minimizes the difference between y_i and $\hat{y}$
- Need to work with squared differences
  - To not end up with a sum of 0
- SSE - Error Sum of Squares



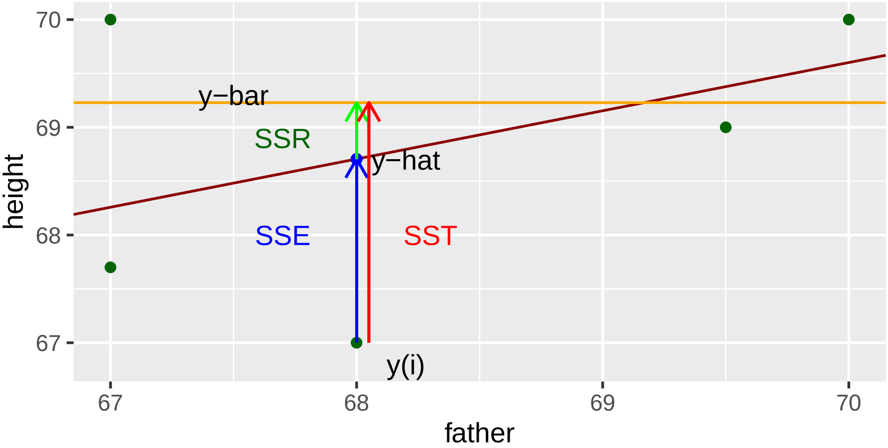
SSE – A Component of Total Sum of Squares (SST)

$$SST = SSE + SSR$$

- SST – Total - SSE – Related to errors/residuals - SSR – Related to/Explained by regression

SST – What Does It Represent?

- The total variance is the difference between the model value for each value of X and the mean of the values of the dependent variable ($\hat{y}$)



Sum of Squares

- Refer to the sum of squares we want to minimize as the **SSE**
  - Error sum of squares
- SSE is a component of the total sum of squares (SST)como componente da soma dos quadrados total
- SSE -– the of the squares related to the residuals
- SSR -– sum of squares related to the regression
- Expression for the SSE

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

$$SSE = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

To Determine the Formula for β_0 & β_1

- To minimize the SSE (determine the most efficient line), we need to use calculus cálculo
- Set the partial derivatives of the SSE with respect to β_0 and β_1

$$\frac{\partial}{\partial \beta_0} SSE = \frac{\partial}{\partial \beta_1} SSE = 0$$

- Called the normal equations
- We let the software calculate the parameters of the equation

Function in R

- Function `lm()` (“linear model”)
- `lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)`
- Important arguments are `formula`, `data`, `subset`, `weights`, `na.action`
  - `formula`: where you show which variables you are modelling
    - Dependent variable comes first
    - Separated from the independent by “ $\sim$ ”
  - For the boys: `height ~ father`
  - `data`: data frame or tibble that contains the variables
  - `subset`, `weights`: parameters that permit customization of the variables
  - `na.action`: how you will deal with missing data in the model variables

Function Applied to Fathers and Sons

- Function `lm` produces a list of 12 items in a special format

```
fit1 <- lm(height ~ father, data = boys)
summary(fit1)
```

```
##
## Call:
## lm(formula = height ~ father, data = boys)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3774 -1.4968  0.0181  1.6375  9.3987
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.25891    3.38663   11.30   <2e-16 ***
## father       0.44775    0.04894    9.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 463 degrees of freedom
## Multiple R-squared:  0.1531, Adjusted R-squared:  0.1513
## F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

What Does This Model Say?

$$\hat{y} = 38.259 + 0.448x$$

- If a father had 0 height, the son would be 38.259 inches tall
  - Doesn't make practical sense
  - Establishes a base for the height calculation
  - For each incremental inch on the father's height, the son would be 0.448 inches taller

Extract the Coefficient Values

- Option 1: use `broom::tidy`
  - Automatically extracts the key information and puts in a tibble

```
broom::tidy(fit1) %>% knitr::kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	38.2589122	3.3866340	11.297032	0
father	0.4477479	0.0489353	9.149788	0

- Option 2: use `coef`

```
coef(fit1)
```

```
## (Intercept)      father
##  38.2589122   0.4477479
```

Predictions of New Values

- You can use the model parameters to predict new values of the heights of sons
- Use `broom::augment`
- How tall would the son of a 72 inch father be?

```
fit1 %>% broom::augment(newdata = data_frame(father = 72))
```

```
## # A tibble: 1 x 2
##   father .fitted
##    <dbl>   <dbl>
## 1     72    70.5
```

Section 5

What Does the Model Mean? How to Interpret It?

Does There Exist a Relationship between the Independent and Dependent Variables?

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- If β_1 (slope of the line) were 0, what would be the equation?

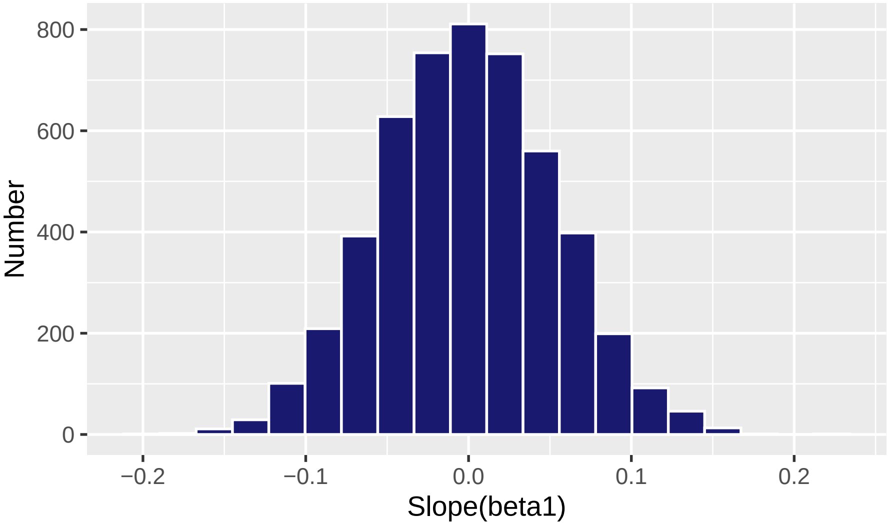
$$Y_i = \beta_0 + \epsilon_i$$

- X disappears
- There would be no relationship between X and Y
  - Only an intercept and an error term
- Makes possible an efficient test of the existence of a relationship between X & Y (or not)
- Create a null hypothesis $H_0 : \beta_1 = 0$

Test of the Null Hypothesis

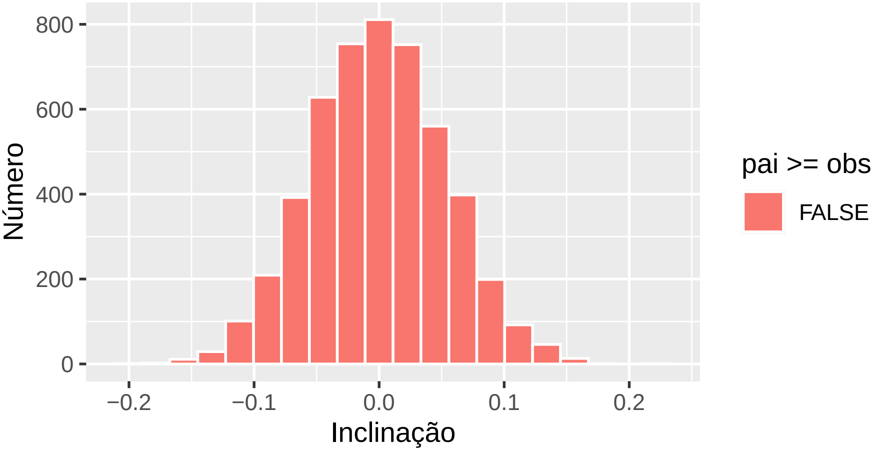
- We will make a simulation of the null hypothesis
- If we do not reject the null, any son's height could have occurred for any father's height
- We can calculate the regression model 5,000 times shuffling around the son's heights
- As a result, we can focus on the values of the slope, β_1
- 2nd, we will compare our observed value of β_1 ('r coef(fit1)[2]) to see where it falls in the simulated values

Histogram of the Slopes of the Simulated Models



Histogram with Values Above and Below Observed Slope

```
## Número de simulações com beta1 >= obs:  0
```



The p-value of the Slope (β_1)

- Because **none** of the simulations produced a value higher than our observed value (0.448)
- We can conclude that the p-value of this test is 0
- There is **no** chance that the slope $= 0$
- Thus, we reject the null hypothesis and conclude that a linear relationship does exist between the heights of fathers and sons

Section 6

Assumptions of Linear Regression and How to Test Them

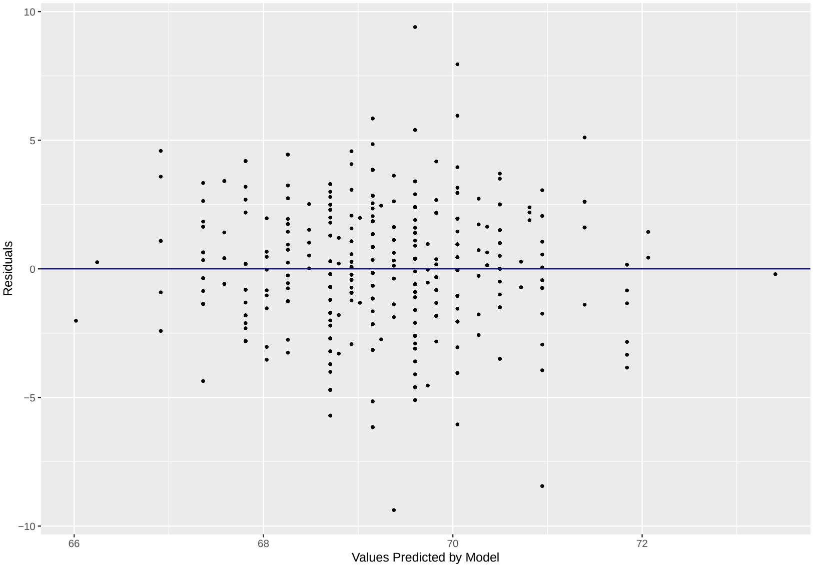
Assumptions of Linear Regression

1. All independent variables must have the same variance
    - Graph of residuals should avoid patterns when looking from left to right
2. All the observations, residuals and independent variables must be independent of each other
    - Graph of residuals should not show a sinuous pattern
3. Residuals should have a near-normal distribution
    - Q-Q graph of the standardized residuals should be a straight line
    - Shows that the variables have a multivarite normal distribution
4. Independent variables should avoid *multicollinearity*
    - They should not have high correlations between them

Residuals Graph

- Graph that shows the value predicted by the model (“fitted value”) vs. the residual
- Use the function `broom::augment()`
  - Extracts efficiently the values used in the model tests

```
mods <- broom::augment(fit1)
residgr <- ggplot(data = mods, mapping = aes(x = .fitted, y = .resid))
residgr <- residgr + geom_point(shape = 20)
residgr <- residgr + geom_hline(yintercept = 0, color = "midnightblue")
residgr <- residgr + labs(x = "Values Predicted by Model",
                          y = "Residuals")
```


10
5
0
−5
−10
Residuals
66
68
70
72
Values Predicted by Model

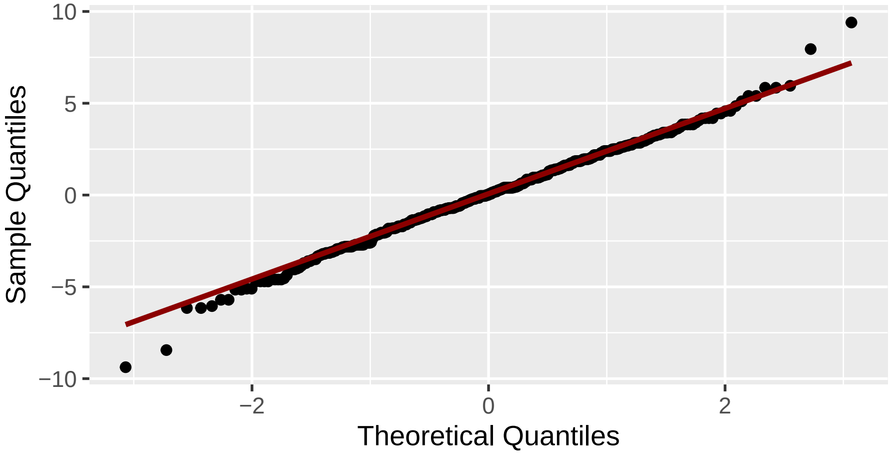
Importance of Residuals

- Can use the residuals to verify if the model respects the assumptions of regression
- Should not show any linear trend

Q-Q Graph

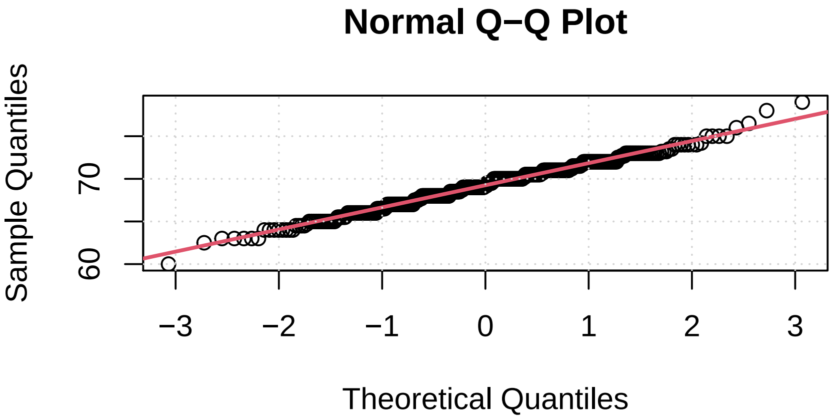
- Verifies the normality of the residuals
  - Closer the curve to a straight line, the better the “fit” with a normal distribution

```r
grqq <- ggplot(data = mods, aes(sample = .resid))
grqq <- grqq + stat_qq()
grqq <- grqq + stat_qq_line(color = "darkred", size = 1)
grqq <- grqq + labs(x = "Theoretical Quantiles",
                    y = "Sample Quantiles")
```


10
5
0
−5
−10
Sample Quantiles
−2
0
2
Theoretical Quantiles

Q-Q Graphs Also Directly Available in Base R

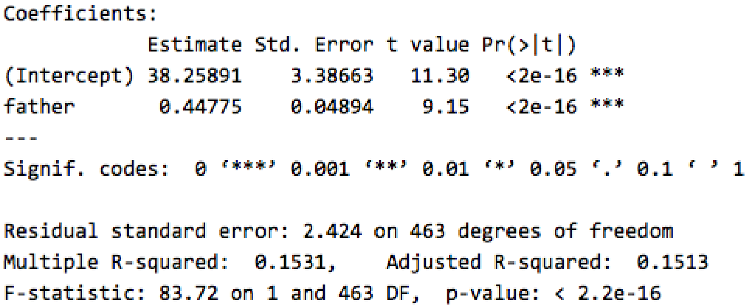
```r
qqnorm(boys$height)
qqline(boys$height, col = 2, lwd = 2)
grid()
```


Normal Q−Q Plot
Sample Quantiles
60
70
−3
−2
−1
0
1
2
3
Theoretical Quantiles

F-Test of Model Variance

- F-Test is a test that verifies that the variances of variables are close to equal
- Uses the F Distribution
  - With 2 degrees of freedom as parameters
- Serves as a test of significance for the model as a whole
- Shown in the `summary()` function output for the `lm()` function

F-Test for the Son-Father Heights Model



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 38.25891    3.38663   11.30   <2e-16 ***
father       0.44775    0.04894    9.15   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424 on 463 degrees of freedom
Multiple R-squared:  0.1531,    Adjusted R-squared:  0.1513 
F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

Summary of the Sum of Squares

- Total Sum of Squares

$$SST = \sum (y_i - \bar{y})^2$$

- Error Sum of Squares

$$SSE = \sum (y_i - \hat{y})^2$$

- Regression Sum of Squares

$$SSR = \sum (\hat{y}_i - \bar{y})^2 = SST - SSE$$

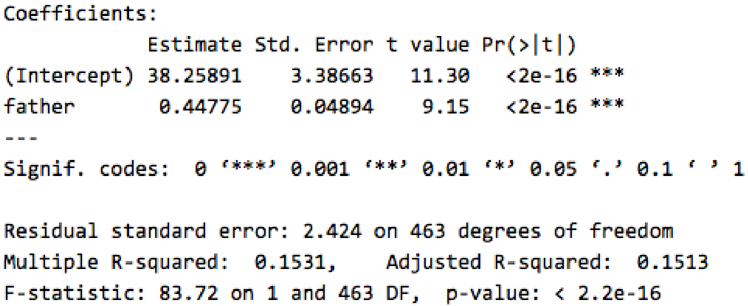
R^2 – Coefficient of Determination

- Measure of how much the regression line explains the variance in Y
- Ratio of SSR to SST

$$R^2 = \frac{SSR}{SST}$$

- Calculated by `lm()`
- Appears in `summary(lm)`
- Varies between 0 and 1
- $\sqrt{R^2} = r$ (correlation coefficient)

R^2



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 38.25891    3.38663   11.30   <2e-16 ***
father       0.44775    0.04894    9.15   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424 on 463 degrees of freedom
Multiple R-squared:  0.1531,    Adjusted R-squared:  0.1513 
F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

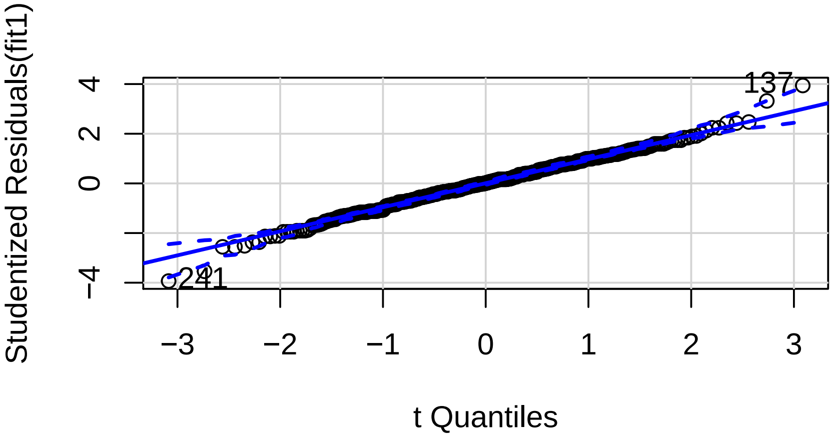
Importance of R^2

- If 100% of the variance in Y can be explained by the regression
- $SSR = SST$
- $\therefore\ R^2 = SST/SST = 1$
- Variance completely explained by the regression
  - Means there is no error
- In general, the degree to which the regression explains the model variance

Section 7

More Advanced Graph

qqPlot() Function from the car Package



```
## [1] 137 241
```

Section 8

Multiple Linear Regression - **MLR**

Multiple Linear Regression - **MLR**

- Regression with more than 1 independent variable
- Now we can also call the independent variables “covariates”
- 1st real machine learning model
- Change in the Equation of the Regression Model

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i + ... + \beta_k X_i + \epsilon_i$$

Types of *Machine Learning*

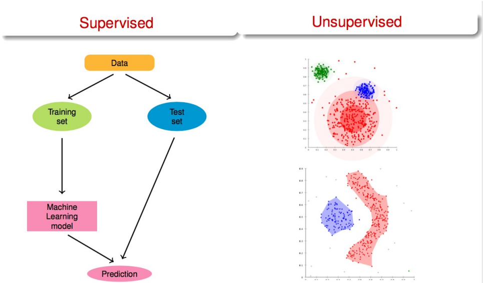
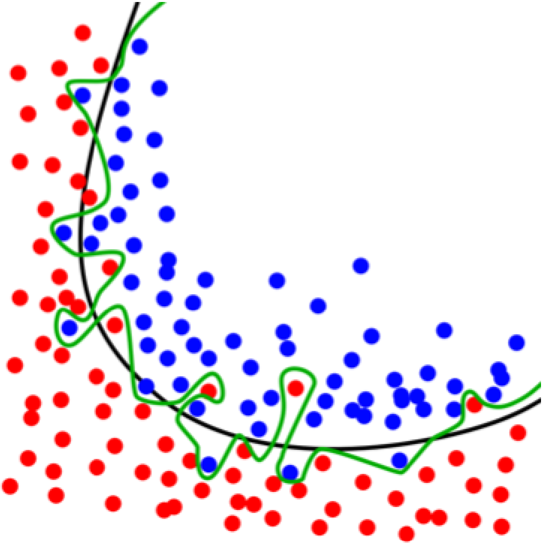


Image Source: Wikipedia

Training x Testing of Models

- Divide data frames into different parts
- To avoid *overfitting*
- **NEVER, EVER, USE THE SAME CASES FOR TESTING THAT YOU USED FOR TRAINING A MODEL**

Overfitting



Model Characteristics

- Covariates
- How many are sufficient for construction of a model?
  - Too few – model does not describe the condition being modelled
  - Too many – overfitting

Strengthening a Model

- Bootstrapping
- k-fold Cross Validation
  - Pull out a group (fold) from the training group
  - Train the model
  - Test the model with the training cases
  - Do the same with all the other groups
- Use as the final model that which shows the best performance

Machine Learning in Biological/Medical Modelling

- Typically, projects with “big data”
- Model can provide information quickly and correctly
    - Clinicians can use the information to design treatments or diagnostics
- Applications in personalized or precision medicine
- Example:
    - Diagnosis of breast cancer with help from a computer model

Can We Have Confidence in Machine Learning Models?

- ML algorithms model interactions among variables
- Interpretation of results of ML models can be difficult
- ML algorithms' "black box" hide how they make choices
  - For some algorithms (e.g. neural networks)
- Thus, *we need models that mean something* to the
  - Builders
  - Users
- "Meaningful Models"

What Makes a Model a “Meaningful Model”

- Being able to generalize based on the model
- Offer an answer to the original motivating question
  - . . . with sufficient precision to be trusted
- The level of precision depends on the nature of the problem

Covariates – *Features*

- The independent variables
- Variables we use to train the model
- Select the **right** variables
- More features not necessarily good
    - Danger of “overfitting”

Section 9

Mãos na Massa

Data

- Continue with the galton data
- Bring the mother's height into the analysis

```
glimpse(galton)
```

```
## Rows: 898
## Columns: 6
## $ family <fct> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, ...
## $ father <dbl> 78.5, 78.5, 78.5, 78.5, 75.5, 75.5, 75.5, 75.5, 75.0, 75.0, ...
## $ mother <dbl> 67.0, 67.0, 67.0, 67.0, 66.5, 66.5, 66.5, 66.5, 64.0, 64.0, ...
## $ sex    <fct> M, F, F, F, M, M, F, F, M, F, M, M, F, F, F, M, M, M, F, F, ...
## $ height <dbl> 73.2, 69.2, 69.0, 69.0, 73.5, 72.5, 65.5, 65.5, 71.0, 68.0, ...
## $ nkids  <int> 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, ...
```

Section 10

The caret Method of Machine Learning

Organized Workflow

- Methodology comes from `caret` package
- Step 1
  - Divide the cases in 2 groups: *training*, *test*
  - Random division
- Train the model with the training data
- After, test the predictions of the model with the values from the test data
- Objective: Make accurate predictions
  - More important than the elegance of the model

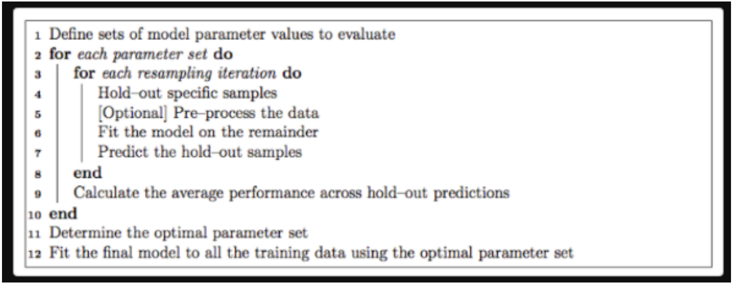
Method Requires a Number of Packages

- `caret` : *Classification And REgression Training*
- `ggplot`: graphs
- `broom` : functions for showing and comparing models
- `nortest`: statistical normality tests
- `janitor`: help with tables

```r
pacman::p_load(caret, ggplot2, broom, nortest, janitor)
```

The caret Process

- An efficient *workflow* for regression and classification problems
- Models built with the function `caret::train`



```
Define sets of model parameter values to evaluate
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples
        [Optional] Pre–process the data
        Fit the model on the remainder
        Predict the hold–out samples
    end
    Calculate the average performance across hold–out predictions
end
Determine the optimal parameter set
Fit the final model to all the training data using the optimal parameter set
```

caret Division of Data

- Function `caret::createDataPartition()`
- Give the function the dependent variable `galton$height`
- Proportion (`p`) that you want in the training sample (70%)
  - Can be between 50% and 70%
  - Higher percentage can cause *overfitting*
- Function returns the *indices* of cases for the training set
- Give it the argument `list = FALSE`

```r
set.seed(42)
indice <- createDataPartition(galton$height, p = 0.70, list = FALSE)
head(indice[, 1], 25)
```

```
##  [1]  2  3  4  6  7  8  9 13 14 15 17 18 20 21 23 24 25 26 27 28 29 30 31 33 34
```

Create train_data and test_data

- **VSS** Remember the comma after the indice
  - Why?
- For the test_data, you want the data that are **NOT** in the train_data
  - Thus, you need to use the minus sign (-)

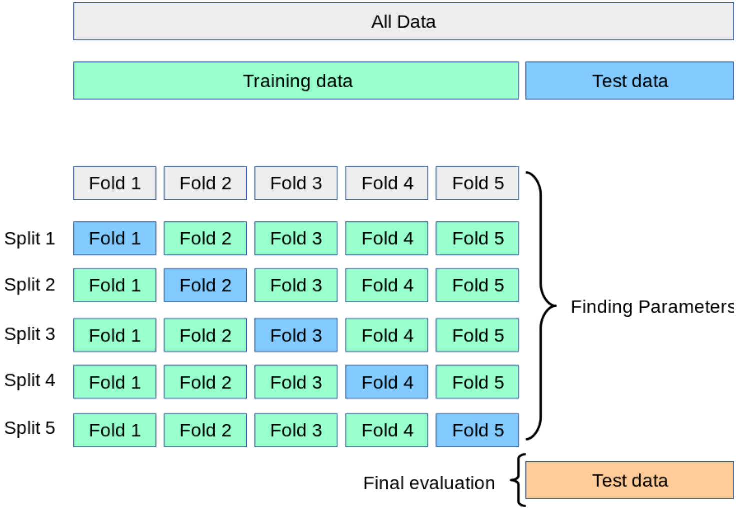
```
train_data <- galton[indice, ]
test_data <- galton[-indice, ]
```

Cross-Validation

- Validation of the calculation of the model parameters
  - Using bits of each case repeatedly
- Mathematical equivalent of amplifying biological samples
- Related to the process of resampling called *bootstrap*
- `caret` selects the model that has the best performance

k-fold Cross-Validation – Process

- Divide the training sample into k equal subgroups
- Train the model with $k - 1$ of the folds
- Software tests this model with the cases of the fold left out
  - Test is of the predictive performance (precision)
- Repeat until you have left out all the folds
- Can repeat the entire process a number of times



All Data

Training data | Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

Final evaluation | Test data

Source: scikit.learn.org

Pre-Processing

- If there are signs that some variables are non-normal
- You can reduce the non-normality of the curves with
  - Centralization (subtract the mean from the value) $(x_i - \bar{x})$
  - Normalization (divide the centralized value by the std. deviation) $\frac{(x_i - \bar{x})}{s}$
- `caret` will perform these for you

train() Heights Model

- caret::train() the function that determines the parameters of the regression model

```
fit_pai_mae <- caret::train(height ~ father + mother,
                        method = "lm",
                        data = train_data,
                        trControl = trainControl(method = "repeatedcv",
                                                 number = 5,
                                                 repeats = 10,
                                                 savePredictions = "none",
                                                 verboseIter = FALSE))
```

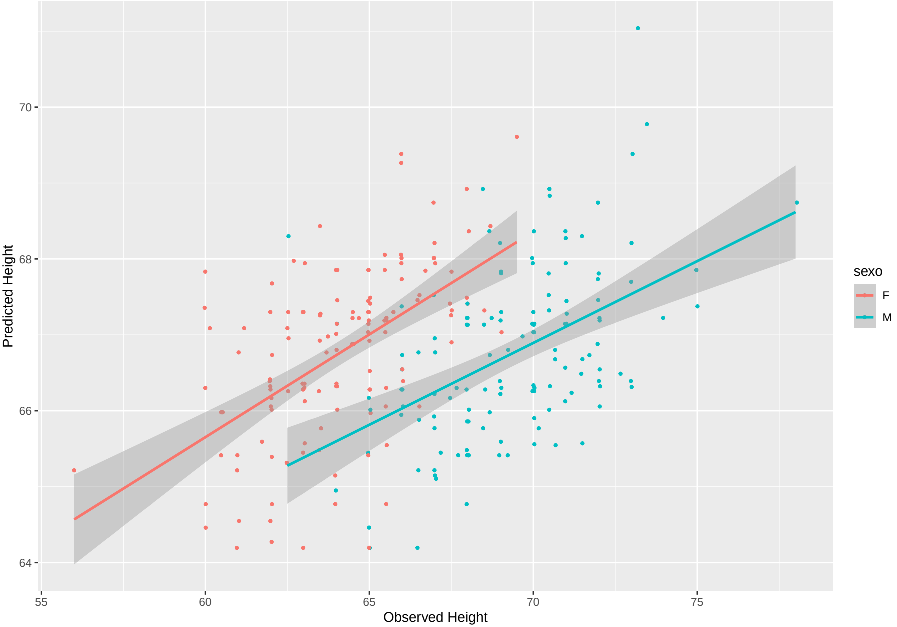
```r
summary(fit_pai_mae)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.480 -2.740 -0.179  2.807 11.699
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.59851    5.08952   4.637 4.31e-06 ***
## father       0.37731    0.05589   6.751 3.34e-11 ***
## mother       0.26601    0.05870   4.532 7.00e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.404 on 628 degrees of freedom
## Multiple R-squared:  0.1052, Adjusted R-squared:  0.1023
## F-statistic:  36.9 on 2 and 628 DF,  p-value: 7.022e-16
```

How Did the Model Do?

- Apply the model to the data from `test_data`
- Until now, the model has not seen these data
- Shows what you can do with any data that measures the same phenomenon
- `predict` calculates the predicted values using the model parameters

```
# previsões
prv <- predict(fit_pai_mae, test_data)
# comparar
gg_pai_mae_1 <- data.frame(obs = test_data$height,
                           previs = prv,
                           sexo = test_data$sex) %>%
  ggplot(aes(x = obs, y = previs, color = sexo)) +
    geom_jitter(shape = 20) +
    geom_smooth(method = "lm") +
    labs(x = "Observed Height", y = "Predicted Height")
```


Predicted Height
Observed Height
70
68
66
64
55
60
65
70
75
sexo
F
M

How Accurate Was the Model?

- Look at the difference between the real (observed) values and the predicted values
- How many of these differences were less than a reasonable standard (? 2 inches)

```
pred <- predict(fit_pai_mae, test_data)
res <- tibble(pred = pred,
                obs = test_data$height,
                dif = obs - pred)
padrao_in <- 2
# teste de bom, ruim
res <- res %>%
  mutate(bomruim = ifelse(abs(dif) <= padrao_in, "bom", "ruim"))
tabyl(res$bomruim) %>% adorn_pct_formatting()
```

```
##  res$bomruim   n percent
##          bom  95   35.6%
##         ruim 172   64.4%
```

Model Is Not Good

- Very low accuracy
  - 36% within our standard of 2 inches
- R^2 very low (0.1023)
  - Only 10% of the variance in the model was explained by the covariates

Can We Do Better?

- Gender could be having an effect on height
- Gender is a categorical variable
- Regression compares distributions of numbers
- But, it can include categorical variables

Categorical Variables in Regression

- Divide the variable into a series of "*dummy*" variables
  - 1 *dummy* variable for each level of the categorical variable (less the 1st level)
  - k - 1 dummy variables
- If there are 3 levels (`high`, `medium`, `low`), the system will create 2 new variables
  - `medium` and `low`
  - `high` will be a reference value that represents the case when none of the other variables is present

```
notas <- tibble(x = rep(c("alto", "media", "baixo"), 3),
                y = c(3, 2, 1, 3, 2, 1, 7, 5, 2))
summary(lm(y ~ x, data = notas))
```

```
##
## Call:
## lm(formula = y ~ x, data = notas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3333 -1.0000 -0.3333  0.6667  2.6667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3333     0.9813   4.416  0.00449 **
## xbaixo       -3.0000     1.3878  -2.162  0.07390 .
## xmedia       -1.3333     1.3878  -0.961  0.37377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.7 on 6 degrees of freedom
## Multiple R-squared:  0.4388, Adjusted R-squared:  0.2518
## F-statistic: 2.346 on 2 and 6 DF,  p-value: 0.1767
```

Include sex in the Heights Regression

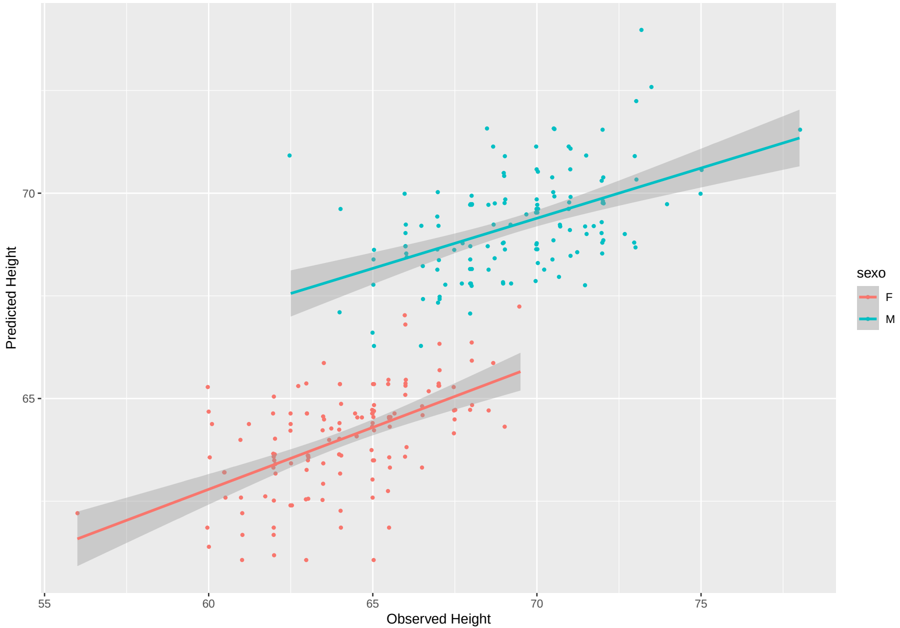
```r
fit_pms <- caret::train(height ~ father + mother + sex,
                        method = "lm",
                        data = train_data,
                        trControl = trainControl(method = "repeatedcv",
                                                 number = 5,
                                                 repeats = 10,
                                                 savePredictions = "none",
                                                 verboseIter = FALSE))
```

```r
summary(fit_pms)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4833 -1.5274  0.0932  1.5369  9.1510
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 15.05115    3.29308   4.571 0.00000586 ***
## father       0.40976    0.03604  11.369    < 2e-16 ***
## mother       0.32157    0.03788   8.489    < 2e-16 ***
## sexM         5.21288    0.17527  29.742    < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.194 on 627 degrees of freedom
## Multiple R-squared:  0.6288, Adjusted R-squared:  0.627
## F-statistic: 354.1 on 3 and 627 DF,  p-value: < 2.2e-16
```

Model Performance

```r
# previsões
prv <- predict(fit_pms, test_data)
# comparar
gg_pms_1 <- data.frame(obs = test_data$height,
                         previs = prv,
                         sexo = test_data$sex) %>%
  ggplot(aes(x = obs, y = previs, color = sexo)) +
    geom_jitter(shape = 20) +
    geom_smooth(method = "lm") +
    labs(x = "Observed Height", y = "Predicted Height")
```



Predicted Height
70
65
55
60
65
70
75
Observed Height
sexo
F
M

How Accurate Was the Model?

```
pred <- predict(fit_pms, test_data)
res_pms <- tibble(pred = pred,
                  obs = test_data$height,
                  dif = obs - pred)
padrao_in <- 2
# teste de bom, ruim
res_pms <- res_pms %>%
  mutate(bomruim = ifelse(abs(dif) <= padrao_in, "bom", "ruim"))
tabyl(res_pms$bomruim) %>% adorn_pct_formatting()
```

```
##  res_pms$bomruim   n percent
##              bom 183   68.5%
##             ruim  84   31.5%
```

Result

- Model predicts 69% of the heights within the standard we set
  - Double the previous model
- R^2 increased to 0.627 (a lot)
- Gender has an important role in determining the heights of the offspring
  - The model captures this characteristic

varImp() Function in caret

- Function evaluates the relative importance of variables in the model
  - Most important - 100%
  - Least important - 0%
- Our Second Model

```
varImp(fit_pms)
```

```
## lm variable importance
##
##        Overall
## sexM    100.00
## father   13.55
## mother    0.00
```

Section 11

Final Example – gapminder

What Is Gapminder?

- R package derived from the site https://www.gapminder.org/
- Monitors socio-economic conditions around the world
- Result of research by Hans Rosling and his family
- They find that poverty in the world can be eliminated by 2030
- Have a look at the video: https://www.gapminder.org/videos/dont-panic-end-poverty/
- Inspiring!

What Can We Learn from This?

- Life Expectancy (`lifeExp`) dependent variable
  - Measured by country
- Our hypothesis is that life expectancy depends on
  - The year surveyed (1952 - 2007 every five years)
    - ★ As time passes (year increases), life expectancy naturally increases
  - Gross domestic product per capita

  Life expectancy as a measure of the health of countries increases based on the economic well being of the population. It has become better over time since the 1950's.

Philosophical Issue

- Objective of Machine Learning models: accurate prediction
  - Niceties of obeying all the assumptions and statistical hypothesis tests not as important
- Objective of Statistical models: relate the data of the sample to a larger truth about a population
  - Assumptions, hypothesis tests, confidence intervals, etc. all very important

Null and Alternative Hypotheses

- If we were building a strictly statistical model, we would first establish a null hypothesis
- H_0: Life expectancy does not vary due to these three variables
  - $H_0 : Y_i = b_0 + \epsilon_i$
- H_1: Life Expectancy has a relationship with at least one of the three covariates

$$H_1 : Y_i = \left(\sum_{k=1}^{K} b_k X_{ik} \right) + b_0 + \epsilon_i$$

Loading Gapminder

```r
gm <- gapminder::gapminder %>%
  janitor::clean_names()
glimpse(gm)
```

```
## Rows: 1,704
## Columns: 6
## $ country    <fct> Afghanistan, Afghanistan, Afghanistan, Afghanistan, Afgh...
## $ continent  <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, As...
## $ year       <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 19...
## $ life_exp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, ...
## $ pop        <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372...
## $ gdp_percap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.11...
```

Descriptive Statistics

```r
gm %>%
  select(year, pop, gdp_percap) %>%
  mutate(pop = log10(pop)) %>%
  descr(stats = c("mean", "sd", "min", "q1", "med", "q3", "max", "iqr", "cv"),
      transpose = TRUE)
```

```
## Descriptive Statistics
## gm
## N: 1704
##
##                         Mean     Std.Dev         Min          Q1      Median          Q3           Max         IQR       CV
## ---------------- --------- --------- --------- --------- --------- --------- ----------- --------- ------
##       gdp_percap   7215.33   9857.45    241.17   1201.92   3531.85   9325.86   113523.13   8123.40   1.37
##              pop      6.85      0.70      4.78      6.45      6.85      7.29        9.12      0.85   0.10
##             year   1979.50     17.27   1952.00   1964.50   1979.50   1994.50     2007.00     27.50   0.01
```

```r
paste("Correlation Coefficient (year x life):", with(gm, round(cor(life_exp, year), 3)))
```

```
## [1] "Correlation Coefficient (year x life): 0.436"
```

```r
paste("Correlation Coefficient (life x gdp):", with(gm, round(cor(life_exp, gdp_percap), 3)))
```

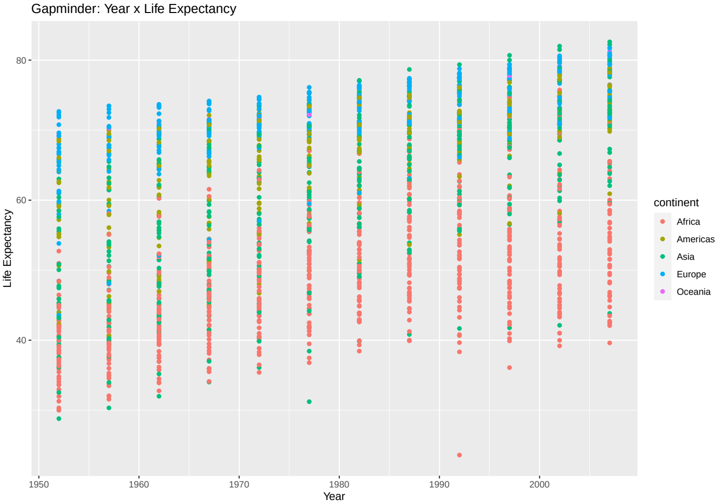
```
## [1] "Correlation Coefficient (life x gdp): 0.584"
```

```r
paste("Correlation Coefficient (gdp x life):", with(gm, round(cor(gdp_percap, year), 3)))
```

```
## [1] "Correlation Coefficient (gdp x life): 0.227"
```

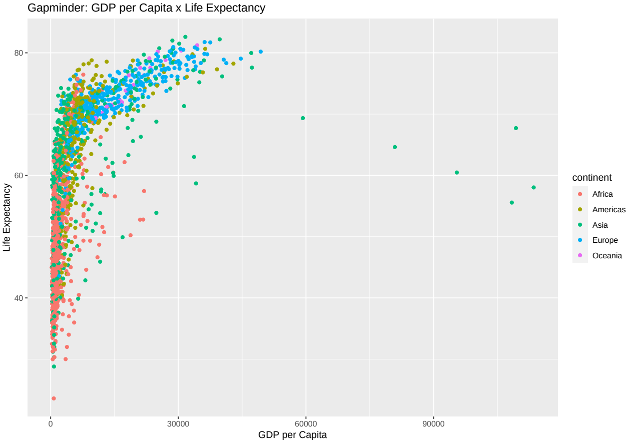
Scatterplot of life_exp against year

```
ggplot(gm, aes(x = year, y = life_exp, color = continent)) +
  geom_point() +
  labs(title = "Gapminder: Year x Life Expectancy", x = "Year", y = "Life Expectancy")
```



Scatterplot of `life_exp` against `gdp_percap`

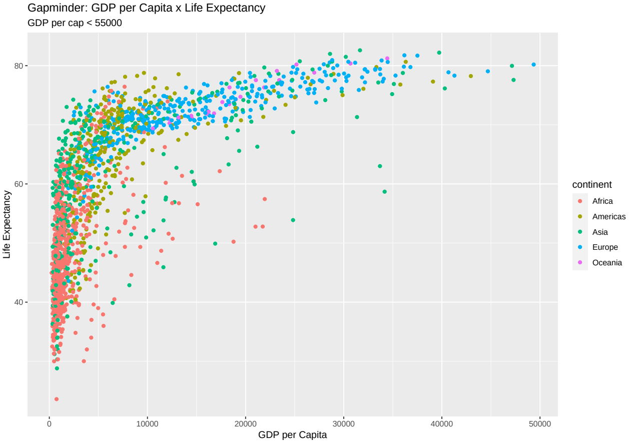
```r
ggplot(gm, aes(x = gdp_percap, y = life_exp, color = continent)) +
  geom_point() +
  labs(title = "Gapminder: GDP per Capita x Life Expectancy", x = "GDP per Capita", y = "Life Expectancy")
```



Scatterplot of `life_exp` against `gdp_percap` - 2

- Get rid of very high GDP's per capita to see mass more clearly

```
gm %>%
  filter(gdp_percap < 55000) %>%
  ggplot( aes(x = gdp_percap, y = life_exp, color = continent)) +
    geom_point() +
    labs(title = "Gapminder: GDP per Capita x Life Expectancy", x = "GDP per Capita", y = "Life Expectancy", su
```



Initializing caret and Related Packages

```
pacman::p_load(caret, tidyverse, broom, nortest, janitor)
```

Setup Training and Test Sets

```r
set.seed = 1946
index <- createDataPartition(gm$life_exp, p = 0.7, list = FALSE)
head(index[, 1], 25)
```

```
##  [1]  1  2  3  5  6  9 10 12 14 15 16 17 18 19 20 21 24 25 26 27 28 30 31 32 33
```

```r
gm_train <- gm[index, ]
gm_test <- gm[-index, ]
```

Plan for Cross-Validation

- Given 142 countries, divide data into 10 folds
    - 14.2 countries per fold
- Repeats of cross-validation
    - Stick with the 10 repeats of heights analysis

train Command to Build Model

```r
fit_gm_1 <- caret::train(life_exp ~ year + gdp_percap,
                         method = "lm",
                         data = gm_train,
                         trControl = trainControl(method = "repeatedcv",
                                                  number = 10,
                                                  repeats = 10,
                                                  savePredictions = "none",
                                                  verboseIter = FALSE))
```

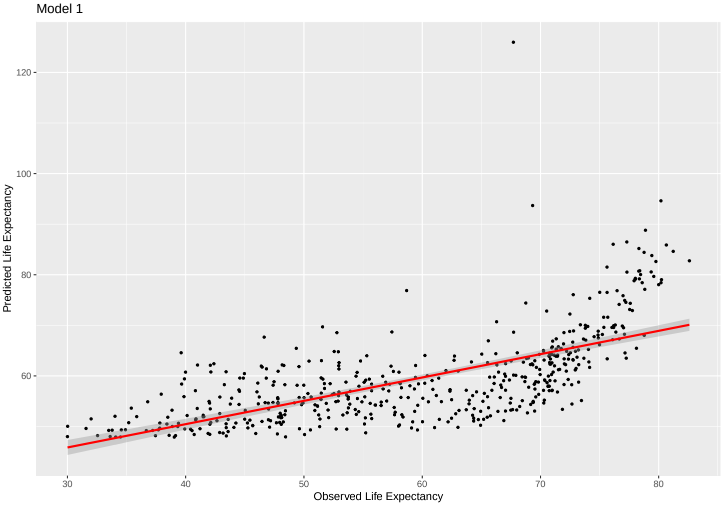
```
summary(fit_gm_1)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.947  -7.035   1.302   7.644  19.946
##
## Coefficients:
##                  Estimate    Std. Error t value Pr(>|t|)
## (Intercept) -444.88582782   32.88263874  -13.53   <2e-16 ***
## year           0.25233666    0.01663001   15.17   <2e-16 ***
## gdp_percap     0.00066984    0.00002919   22.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.691 on 1193 degrees of freedom
## Multiple R-squared:  0.4386, Adjusted R-squared:  0.4377
## F-statistic:   466 on 2 and 1193 DF,  p-value: < 2.2e-16
```

How Did the Model Do?

- Applying model to test data
    - Up to now, model has not seen test data

```r
## predictions
pred_1 <- predict(fit_gm_1, gm_test)
## compare
gm_pred_1 <- data.frame(obs = gm_test$life_exp,
                        preds = pred_1)
gm_1_plot <- ggplot(gm_pred_1, aes(x = obs, y = preds)) +
  geom_jitter(shape = 20) +
  geom_smooth(method = "lm", color = "red")+
  labs(title = "Model 1", x = "Observed Life Expectancy", y = "Predicted Life Expec
gm_1_plot
```


Model 1
Predicted Life Expectancy
120
100
80
60
30
40
50
60
70
80
Observed Life Expectancy

Accuracy of Model 1

- Set an accuracy standard
  - Predicted value 2 years ± observed life expectancy

```r
# calculate difference between predicted and observed
gm_pred_1 <- gm_pred_1 %>%
  mutate(dif = obs - preds,
         goodbad = ifelse(abs(dif <= 2), "good", "bad"))
tabyl(gm_pred_1$goodbad) %>% adorn_pct_formatting()
```

```
##  gm_pred_1$goodbad   n percent
##                bad 231   45.5%
##               good 277   54.5%
```

Conclusion of This Model

- Only captures 47% of the variance in life expectancy
- Continents seem to play a role
  - They all have different slopes
- Add `continent` as a variable to the model
- 58% accuracy not terrific
- Weird outliers with predicted age expectancies above 120 years

Model # 2 – Three Covariates

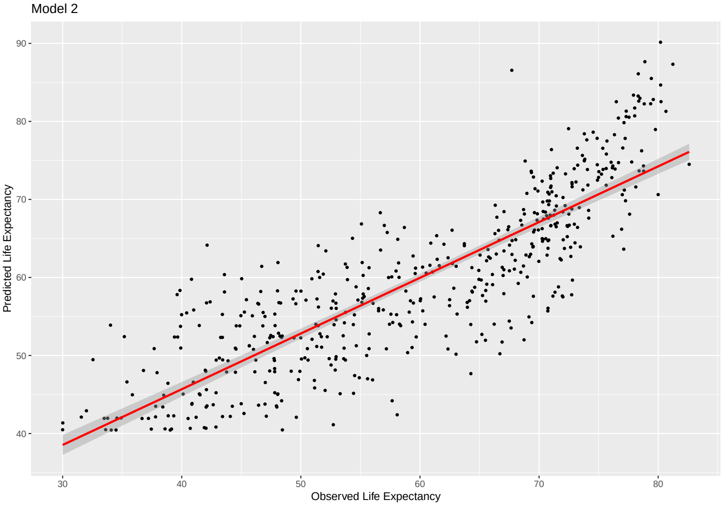
```r
fit_gm_2 <- caret::train(life_exp ~ year + gdp_percap + continent,
                        method = "lm",
                        data = gm_train,
                        trControl = trainControl(method = "repeatedcv",
                                                 number = 10,
                                                 repeats = 10,
                                                 savePredictions = "none",
                                                 verboseIter = FALSE))
```

```r
summary(fit_gm_2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.7132  -4.1337   0.0507   4.4951  19.7608
##
## Coefficients:
##                        Estimate    Std. Error t value Pr(>|t|)
## (Intercept)       -533.19986541   23.47738883  -22.71   <2e-16 ***
## year                 0.29382541    0.01186453   24.77   <2e-16 ***
## gdp_percap           0.00028746    0.00002394   12.01   <2e-16 ***
## continentAmericas   14.34155249    0.58768275   24.40   <2e-16 ***
## continentAsia        8.89348649    0.54472749   16.33   <2e-16 ***
## continentEurope     19.45248558    0.61794077   31.48   <2e-16 ***
## continentOceania    20.91883484    1.90370803   10.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.868 on 1189 degrees of freedom
## Multiple R-squared:  0.7189, Adjusted R-squared:  0.7175
## F-statistic: 506.9 on 6 and 1189 DF,  p-value: < 2.2e-16
```

How Did We Do (This Time)?

```r
## predictions
pred_2 <- predict(fit_gm_2, gm_test)
## compare
gm_pred_2 <- data.frame(obs = gm_test$life_exp,
                        preds = pred_2)
gm_2_plot <- ggplot(gm_pred_2, aes(x = obs, y = preds)) +
  geom_jitter(shape = 20) +
  geom_smooth(method = "lm", color = "red")+
  labs(title = "Model 2", x = "Observed Life Expectancy", y = "Predicted Life Expec
gm_2_plot
```


Model 2
Predicted Life Expectancy
Observed Life Expectancy
90
80
70
60
50
40
30
40
50
60
70
80

Accuracy of Model 2

- Set an accuracy standard
  - Predicted value 2 years ± observed life expectancy

```
# calculate difference between predicted and observed
gm_pred_2 <- gm_pred_2 %>%
  mutate(dif = obs - preds,
         goodbad = ifelse(abs(dif <= 2), "good", "bad"))
tabyl(gm_pred_2$goodbad) %>% adorn_pct_formatting()
```

```
##  gm_pred_2$goodbad   n percent
##                bad 202   39.8%
##               good 306   60.2%
```

Conclusion of This Model

- Better R^2
- Graph shows a clearer trend for accuracy (now 63%)
- Continents seem to play important role
  - Mirrors intuitive thought

Interesting New Machine Learning Site

- https://projects.economist.com/us-2020-forecast/president

The Economist Today Weekly edition ≡ Menu

Forecasting the US elections

The Economist is analysing polling, economic and demographic data to predict America's elections in 2020

→ Read more of our election coverage

President Senate House

Section 12

Next Week's Theme

- Continue with regression
- Focus on Logistic Regression
  - Applying regression to problem of classification
- Remember – project topics to Prof. this week
- Problem Set 2 Available
  - Due 16/10