MAD – Data Analysis & Biostatistics in R

Inference & Regression

James R. Hunter, Ph.D.

DIPA, EPM, UNIFESP

14 January 2025

Section 1

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

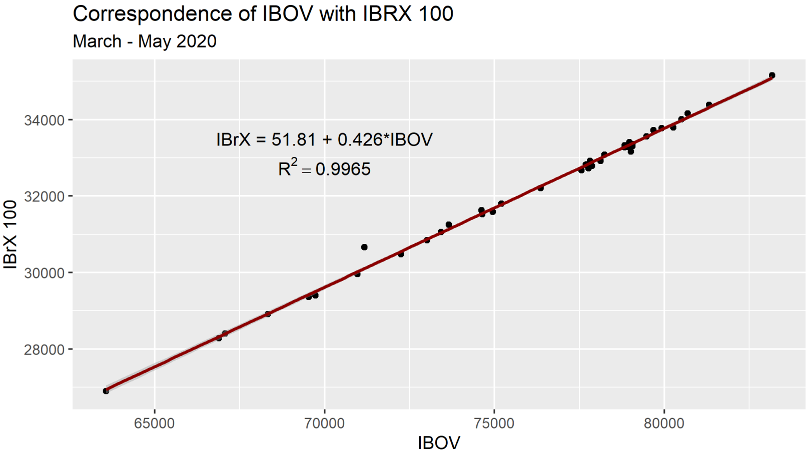
- Solve problems of regression with the *Least Squares* method
- Invented by Carl Friedrich Gauss (1777 - 1855)
- Method minimizes the differences between predicted linear values and the values based on the data
- Achieves the best relation between the real dependent variable and the predicted values 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

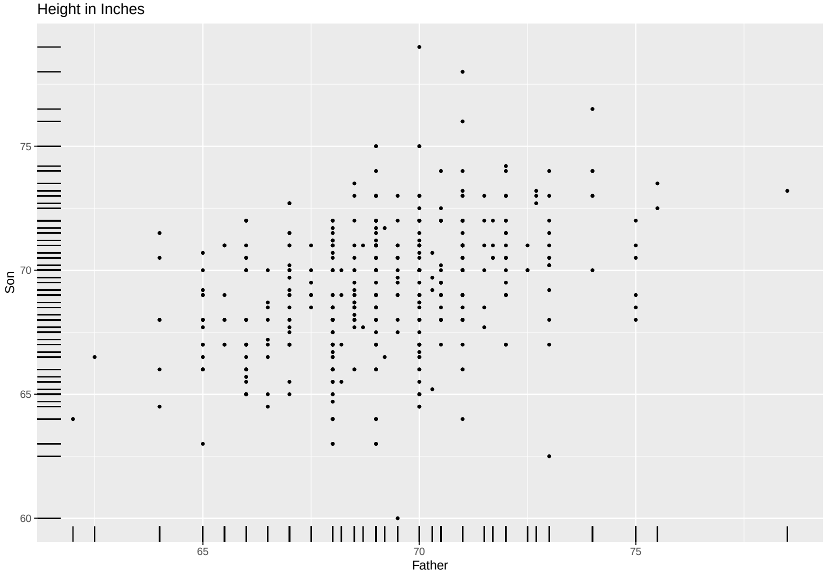
Focus on Fathers and Sons

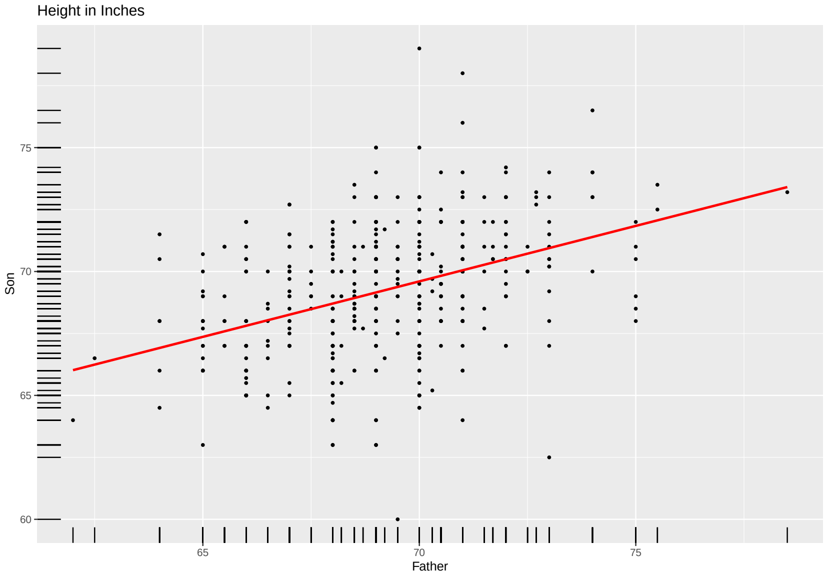
```r
boys <- galton %>%
  filter(sex == "M") %>%
  select(-family, -mother, -sex, -nkids)
glimpse(boys)
```

```
## 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, 74.~
## $ height <dbl> 73.2, 73.5, 72.5, 71.0, 70.5, 68.5, 72.0, 69.0, 68.0, 76.5, 74.~
```

- `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
60
65
70
75
Father
65
70
75

What Have We Learned from the Scatterplot?

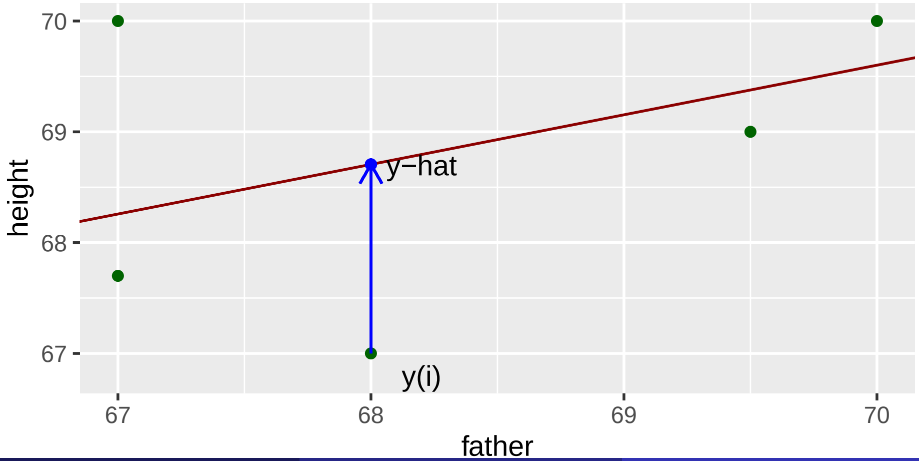
- **Seems** that taller the fathers, taller the sons
- Descriptive statistics of the 2 variables
  - And, correlation

```
## 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



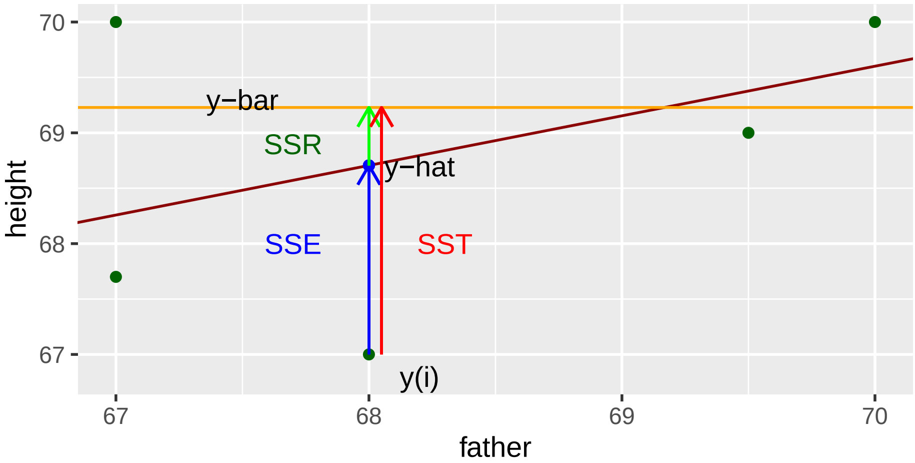
SSE – A Component of Total Sum of Squares (SST)

$$SST = SSE + SSR$$

- 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

```r
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 2

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?

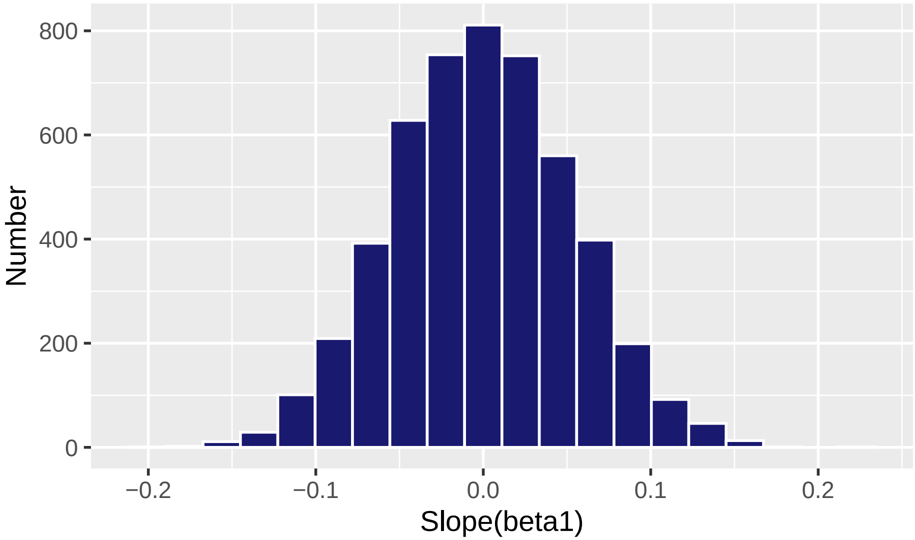
$$Y_i = \beta_0 + \epsilon_i$$

- 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

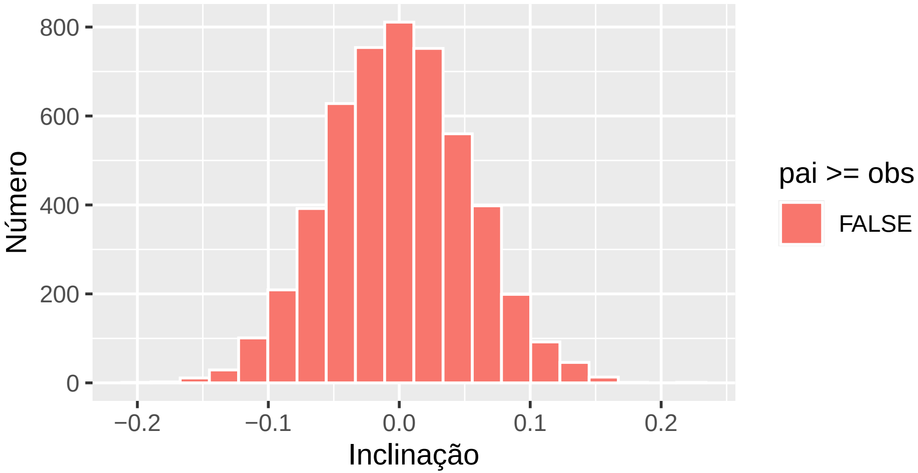
- We will make a simulation of the null hypothesis
- If we do not reject the null, any son's height could have occurred for any father's height
- We can calculate the regression model 5,000 times shuffling around the son's heights
- As a result, we can focus on the values of the slope, β_1
- 2nd, we will compare our observed value of β_1 (0.4477479) 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 3

Assumptions of Linear Regression and How to Test Them

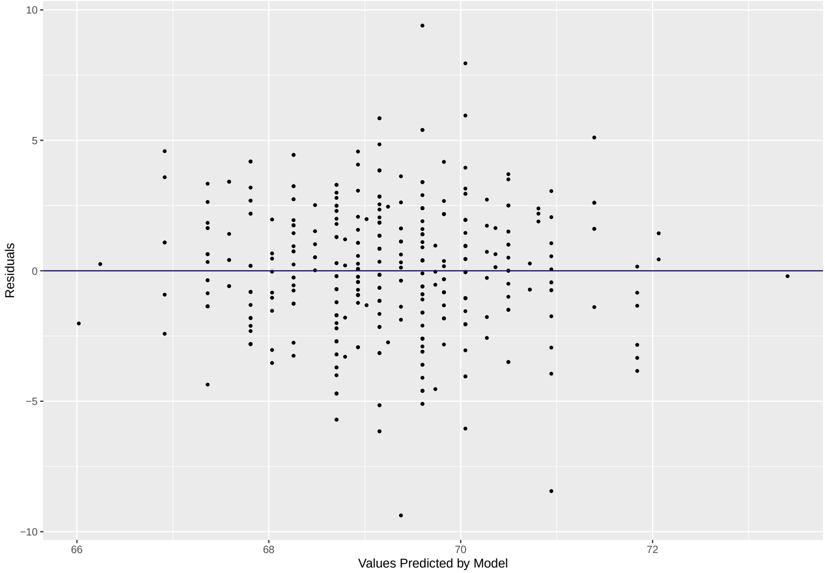
Assumptions of Linear Regression

1. All independent variables must have the same variance
    - Graph of residuals should avoid patterns when looking from left to right
    - **NOT** independent variables all normally distributed
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

```r
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

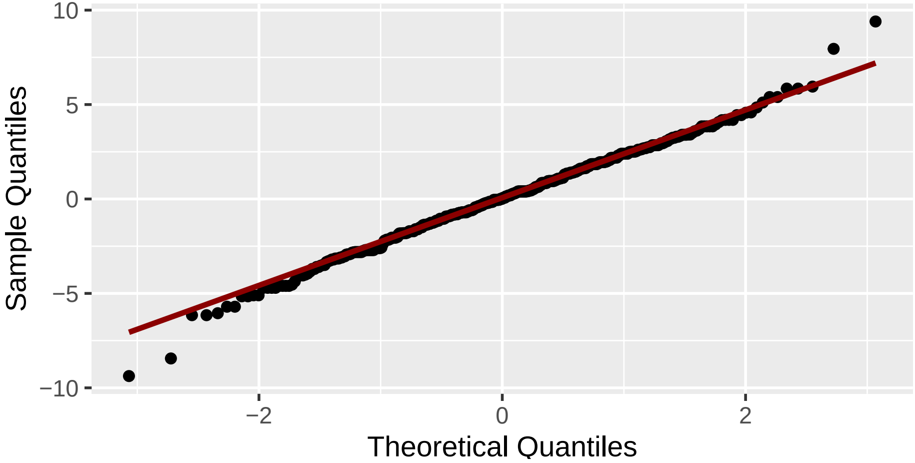
Importance of Residuals

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

Q-Q Graph

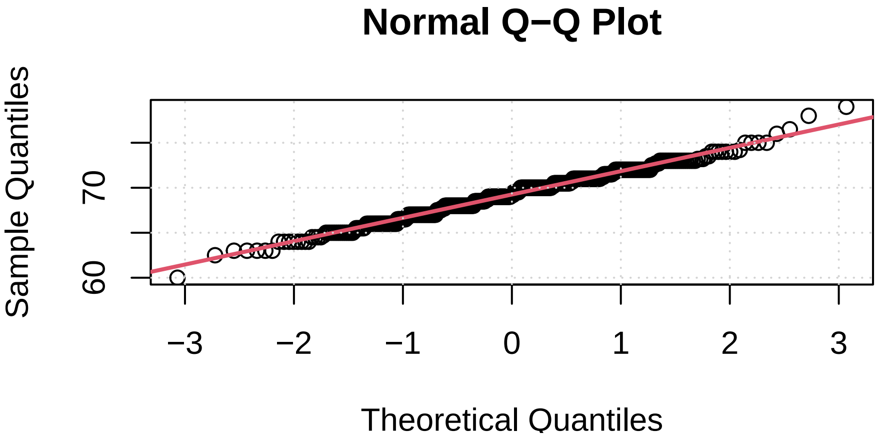
- Verifies the normality of the residuals
  - Closer the curve to a straight line, the better the “fit” with a normal distribution

```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")
```


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

Q-Q Graphs Also Directly Available in Base R

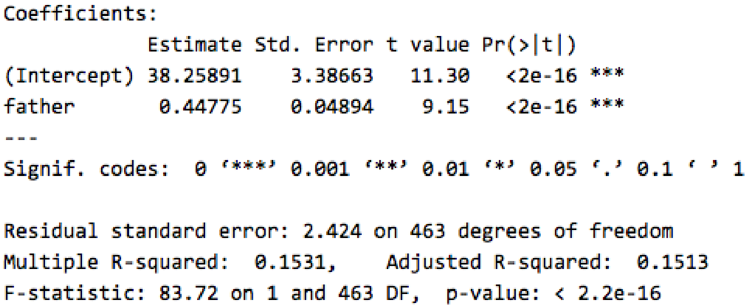
```r
qqnorm(boys$height)
qqline(boys$height, col = 2, lwd = 2)
grid()
```


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

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

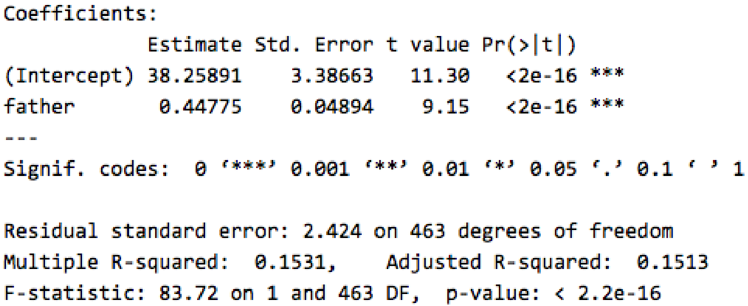
R^2 – Coefficient of Determination

- 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
```

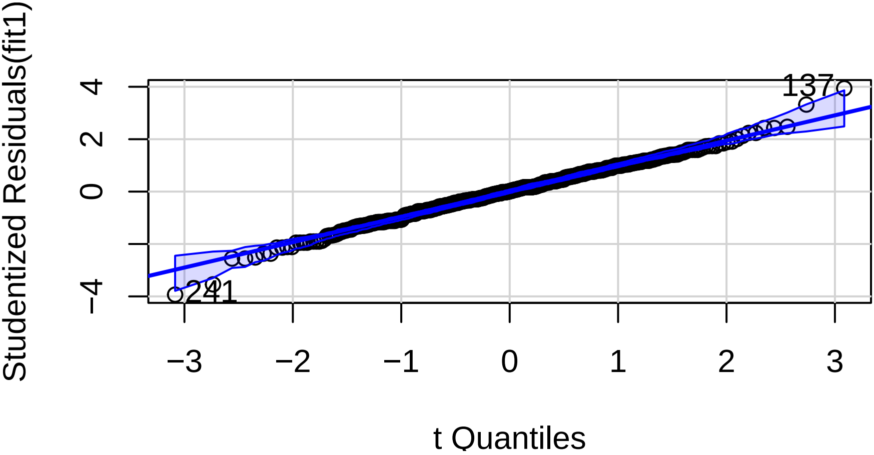
Importance of R^2

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

Section 4

More Advanced Graph

qqPlot() Function from the car Package



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

Section 5

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*

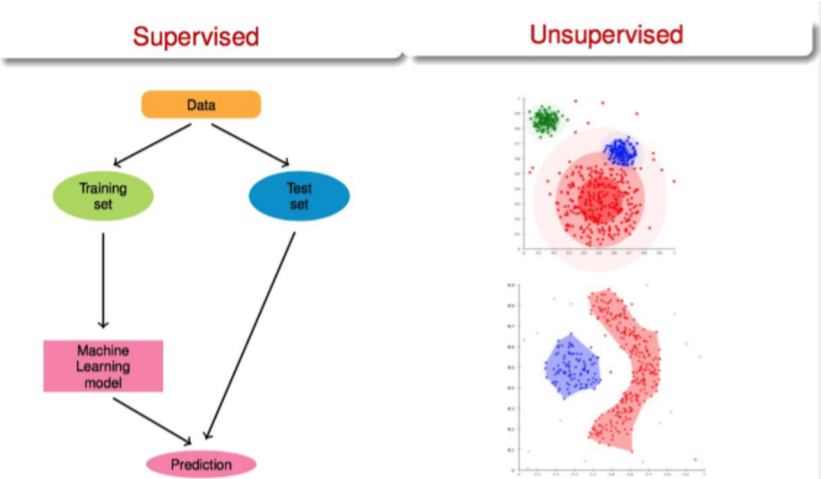
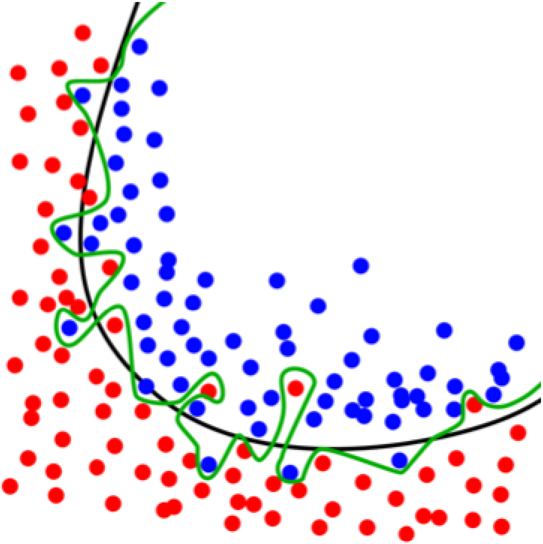


Image Source: Wikipedia

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 6

Mãos na Massa

Data

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

```r
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, 5, ~
## $ father <dbl> 78.5, 78.5, 78.5, 78.5, 75.5, 75.5, 75.5, 75.5, 75.0, 75.0, 75.~
## $ mother <dbl> 67.0, 67.0, 67.0, 67.0, 66.5, 66.5, 66.5, 66.5, 64.0, 64.0, 64.~
## $ sex    <fct> M, F, F, F, M, M, F, F, M, F, M, M, F, F, F, M, M, M, F, F, F, ~
## $ height <dbl> 73.2, 69.2, 69.0, 69.0, 73.5, 72.5, 65.5, 65.5, 71.0, 68.0, 70.~
## $ nkids  <int> 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, ~
```

Section 7

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

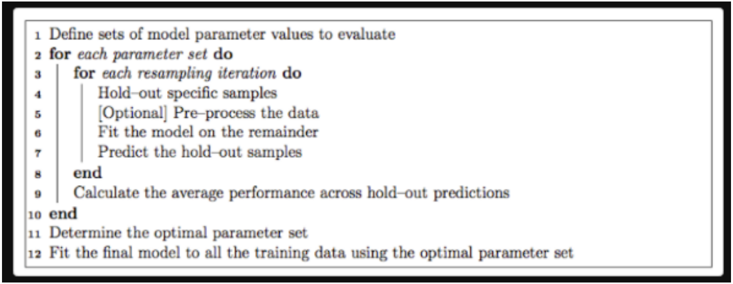
- caret : *Classification And REgression Training*
- ggplot: graphs
- broom : functions for showing and comparing models
- nortest: statistical normality tests
- janitor: help with tables

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

```
##
## The downloaded binary packages are in
##  /var/folders/c7/7ttl8skd5293dgvz_ht79b140000gn/T//RtmpoSA2
```

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 (-)

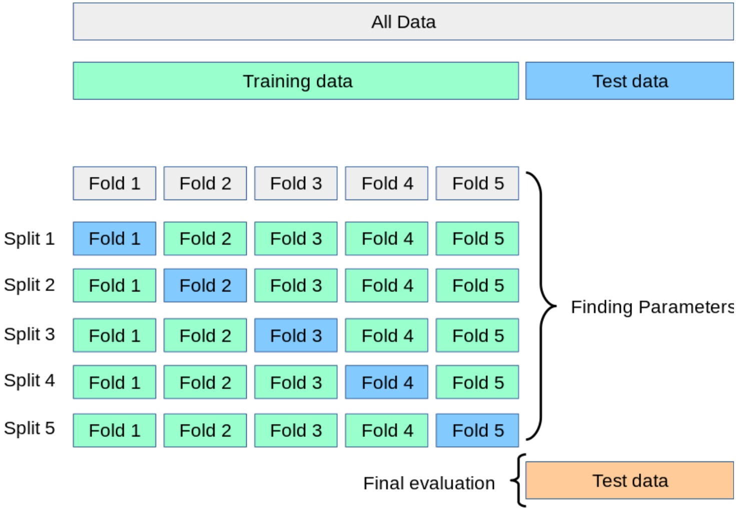
```r
train_data <- galton[indice, ]
test_data <- galton[-indice, ]
```

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))
```

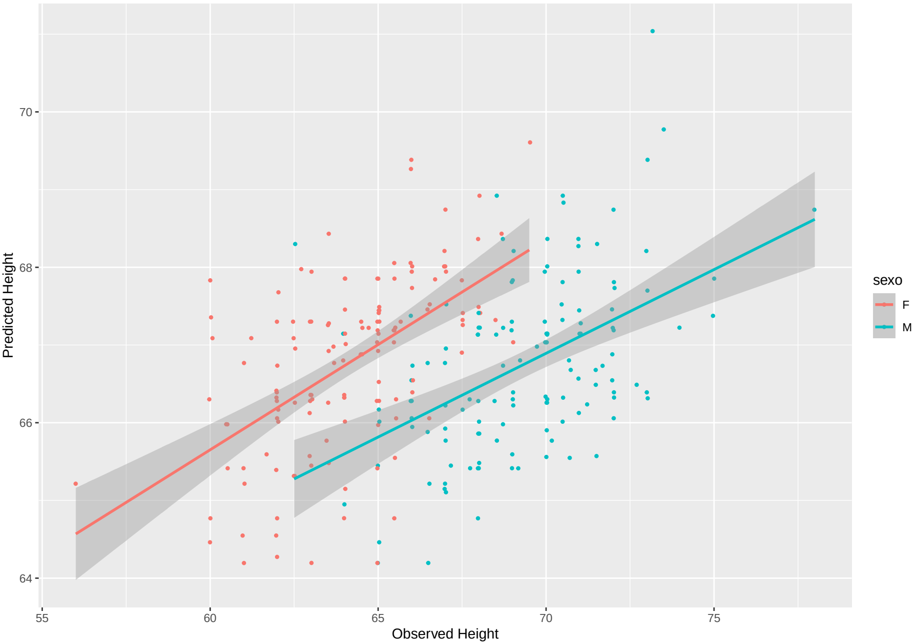
```r
summary(fit_pai_mae)
```

```
## 
## 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

```r
# 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")
```



70
68
66
64
Predicted Height
55
60
65
70
75
Observed Height
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

```r
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

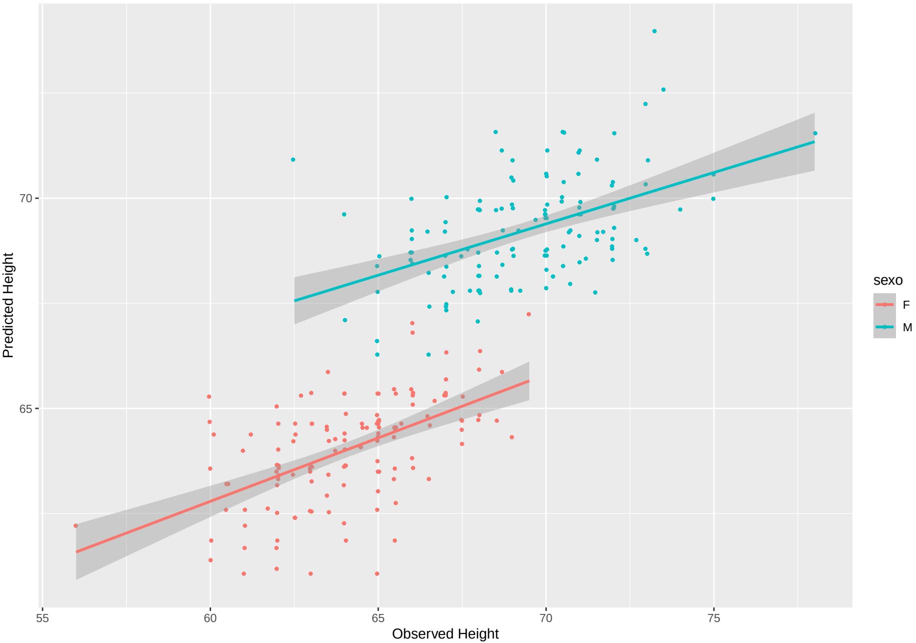
```r
fit_pms <- caret::train(height ~ father + mother + sex,
                        method = "lm",
                        data = train_data,
                        trControl = trainControl(method = "repeatedcv",
                                                 number = 5,
                                                 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")
```


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

How Accurate Was the Model?

```r
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 8

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

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

```
## Rows: 1,704
## Columns: 6
## $ country    <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan",~
## $ continent  <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia,~
## $ year       <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997,~
## $ life_exp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.~
## $ pop        <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 1~
## $ gdp_percap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134,~
```

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)))
```

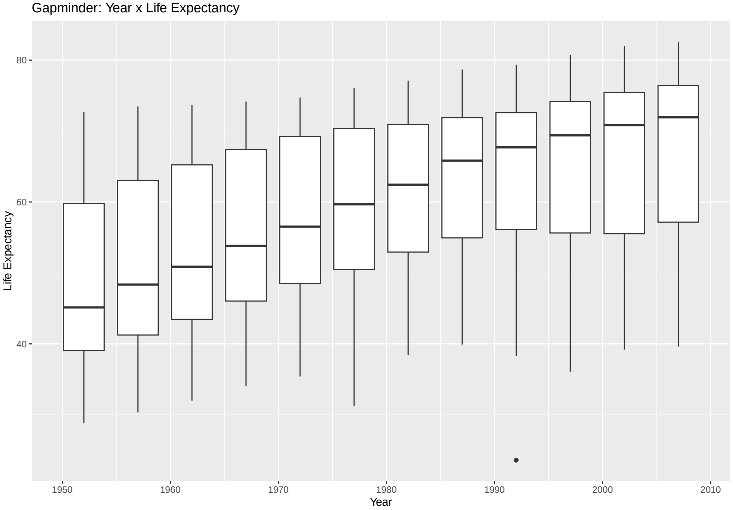
```
## [1] "Correlation Coefficient (life x gdp): 0.584"
```

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

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

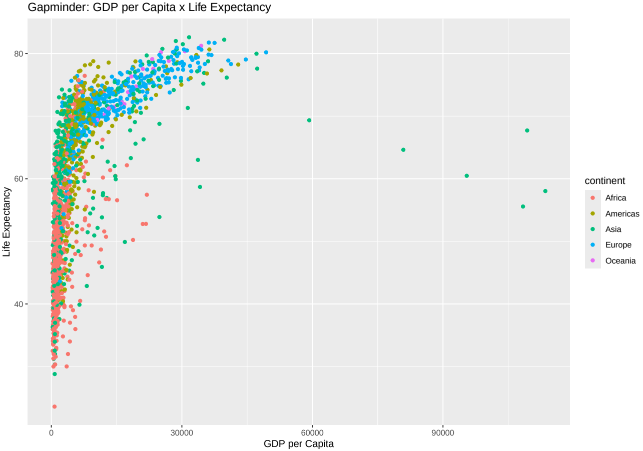
Boxplot of `life_exp` against `year`

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



Scatterplot of life_exp against gdp_percap

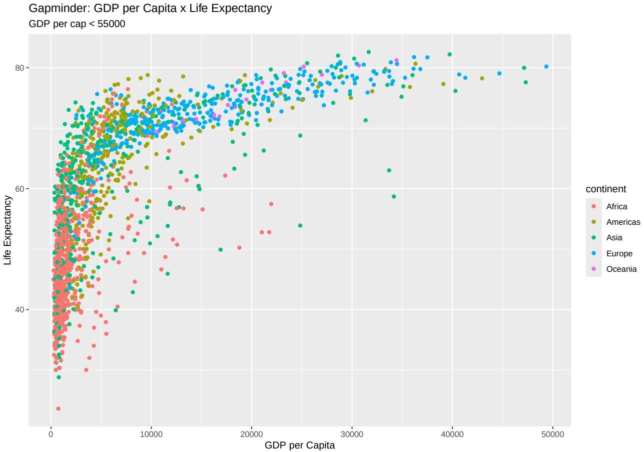
```
ggplot(gm, aes(x = gdp_percap, y = life_exp, color = continent)) +
  geom_point() +
  labs(title = "Gapminder: GDP per Capita x Life Expectancy", x = "GDP per Capita", y = "Life Expectancy")
```



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  3  5  6  8  9 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 30 31
```

```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

```
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))
```

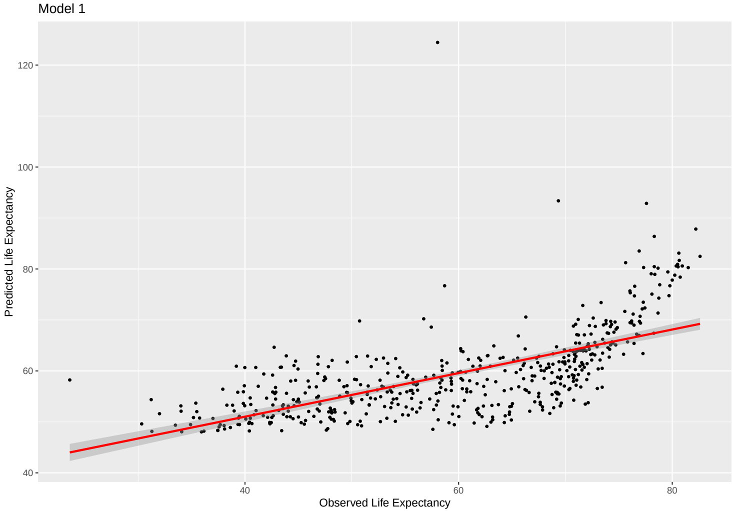
```r
summary(fit_gm_1)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.225  -6.986   1.251   7.754  19.841
##
## Coefficients:
##                  Estimate    Std. Error t value Pr(>|t|)
## (Intercept) -435.89173435   32.86049457  -13.27   <2e-16 ***
## year           0.24780454    0.01662278   14.91   <2e-16 ***
## gdp_percap     0.00066402    0.00002878   23.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.636 on 1193 degrees of freedom
## Multiple R-squared:  0.4454, Adjusted R-squared:  0.4444
## F-statistic:   479 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
40
40
60
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 238   46.9%
##               good 270   53.1%
```

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

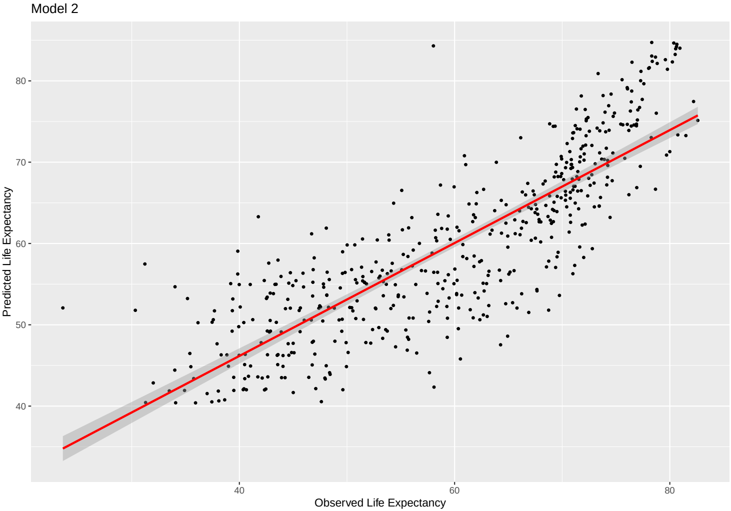
```
fit_gm_2 <- caret::train(life_exp ~ year + gdp_percap + continent,
                         method = "lm",
                         data = gm_train,
                         trControl = trainControl(method = "repeatedcv",
                                                  number = 10,
           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

```r
summary(fit_gm_2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.8105  -4.1216   0.2014   4.3880  19.9893
##
## Coefficients:
##                       Estimate   Std. Error t value Pr(>|t|)
## (Intercept)       -523.9697580   23.3526788  -22.44   <2e-16 ***
## year                 0.2890737    0.0118065   24.48   <2e-16 ***
## gdp_percap           0.0002886    0.0000233   12.39   <2e-16 ***
## continentAmericas   14.2457830    0.5760049   24.73   <2e-16 ***
## continentAsia        9.7975727    0.5376687   18.22   <2e-16 ***
## continentEurope     19.5029043    0.6128062   31.83   <2e-16 ***
## continentOceania    21.0420867    1.7663765   11.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.788 on 1189 degrees of freedom
## Multiple R-squared:  0.7256, Adjusted R-squared:  0.7243
## F-statistic: 524.1 on 6 and 1189 DF,  p-value: < 2.2e-16
```

How Did We Do (This Time)?

```
## 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
80
70
60
50
40
Observed Life Expectancy
40
60
80

Accuracy of Model 2

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

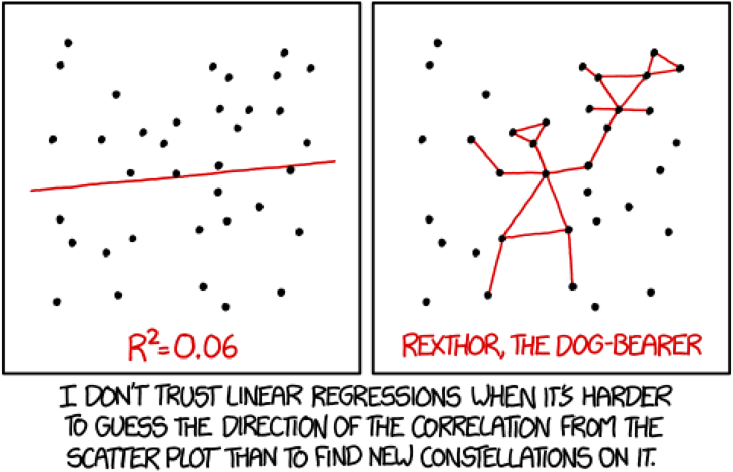
```
# calculate difference between predicted and observed
gm_pred_2 <- gm_pred_2 %>%
  mutate(dif = obs - preds,
         goodbad = ifelse(abs(dif <= 2), "good", "bad"))
tabyl(gm_pred_2$goodbad) %>% adorn_pct_formatting()
```

```
##  gm_pred_2$goodbad   n percent
##                bad 196   38.6%
##               good 312   61.4%
```

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

Danger of Interpretation when R^2 Low


R²=0.06
REXTHOR, THE DOG-BEARER
I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.