Data Analytics
CS301
Basic Stats

Fall 2018
Oliver Bonham-Carter

Writing Functions

```
functionName <- function(arg1, arg2, arg3=2, ...) {
  newVar <- sin(arg1) + sin(arg2) # do useful stuff
  newVar / arg3  # Return value }
```

```
functionName(2,3,1) # run function with inputs
```

- **functionName**: is the function's name
- **args**: arguments of the function, also called formals to import data into a function. No limit to the number for a function.
- **Return value**: The last line of the code is the value that will be returned by the function. It is not necessary that a function return anything

Example of Function

```
#Return the sum of squares:
sumOfSquares <- function(x,y) {
  x^2 + y^2
 }
#run sumOfSquares () with x=2 and y=4
sumOfSquares(2,4) # returns 20
```

Another Simple Example

```
# function to plot points on the canvas
redPlot <- function(x, y, ...) {
        plot(x, y, col="red")
        }
# run the function
redPlot(2,4) # plot a red point
```

Yet, Another Example: Using An If-Else Statement

```
GimmeAtLeastFive <- function(inNum){
  if(inNum >= 5){
      print("That is at least five")
    }
  else{
      print("not enough")
  }
}
```

Basic Stats

- We will spend some time looking at different types of statistical tests so that they can be implemented in code.

Putting Things Together: Find Some Basic Stats

```
library(dplyr) # and load tidyverse too!
data_people <- tibble::tribble(
  ~EyeColour, ~Height, ~Weight, ~Age,
  "Blue",         1.8, 110L, 18L,
  "Brown",        1.9, 150L, 34L,
  "Blue",         1.7, 207L, 28L,
  "Brown",        1.9, 170L, 21L,
  "Blue",         1.9, 164L, 29L,
  "Brown",        1.9, 183L, 31L,
  "Brown",        1.9, 175L, 20L,
  "Blue",         1.9, 202L, 27L
)
```

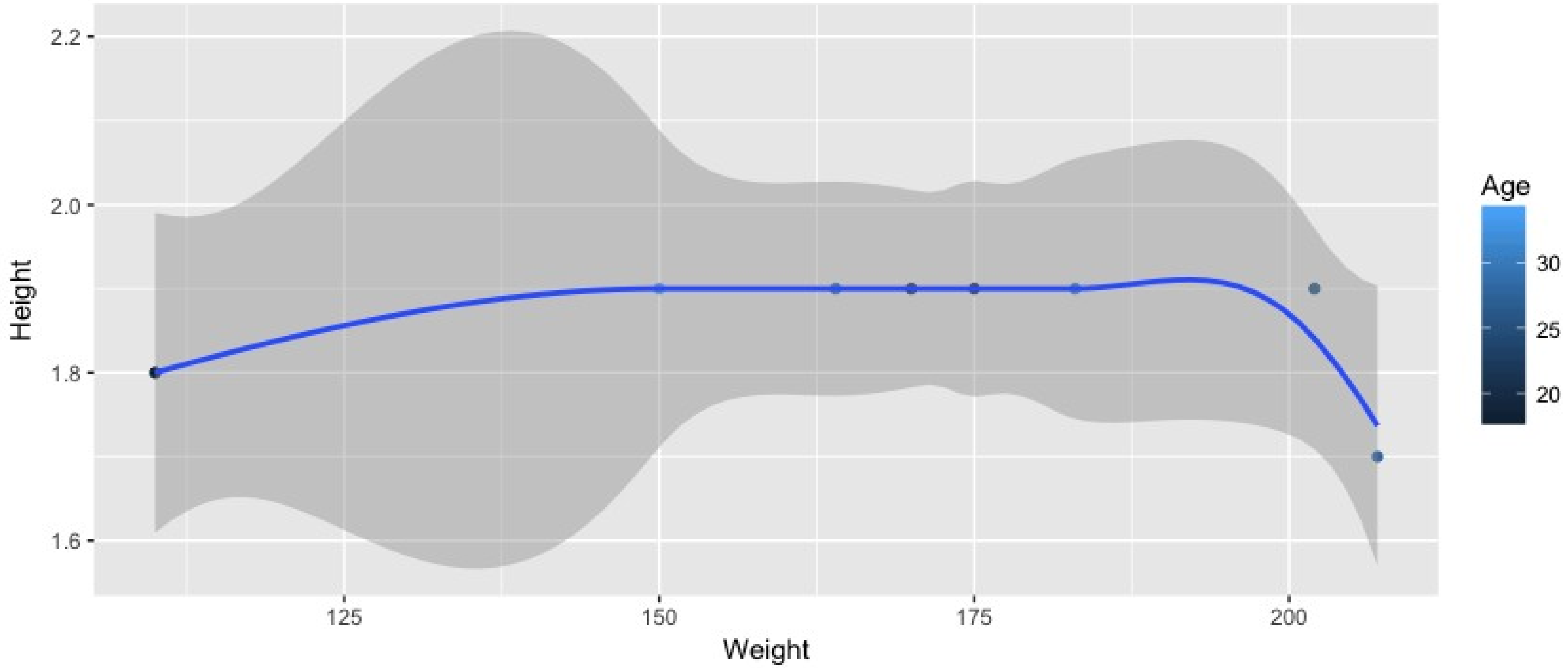
Calculate the Body Mass Index (BMI)

```
# Find the average BMI of people with blue eyes using piping

# Note: BMI = (height / (weight * weight))

data_people %>% select(EyeColour, Height, Weight) %>%
filter(EyeColour=="Blue") %>% mutate(BMI = Weight / Height^2)
%>% summary(averageBMI == mean(BMI))
```

```
> data_people %>% select(EyeColour, Height, Weight) %>% filter(EyeColour=="Blue") %>%
mutate(BMI = Weight / Height^2) %>% summary(averageBMI == mean(BMI))
  EyeColour              Height          Weight            BMI
 Length:4           Min.   :1.700   Min.   :110.0   Min.   :33.95
 Class :character   1st Qu.:1.775   1st Qu.:150.5   1st Qu.:42.56
 Mode  :character   Median :1.850   Median :183.0   Median :50.69
                    Mean   :1.825   Mean   :170.8   Mean   :51.74
                    3rd Qu.:1.900   3rd Qu.:203.2   3rd Qu.:59.87
                    Max.   :1.900   Max.   :207.0   Max.   :71.63
```

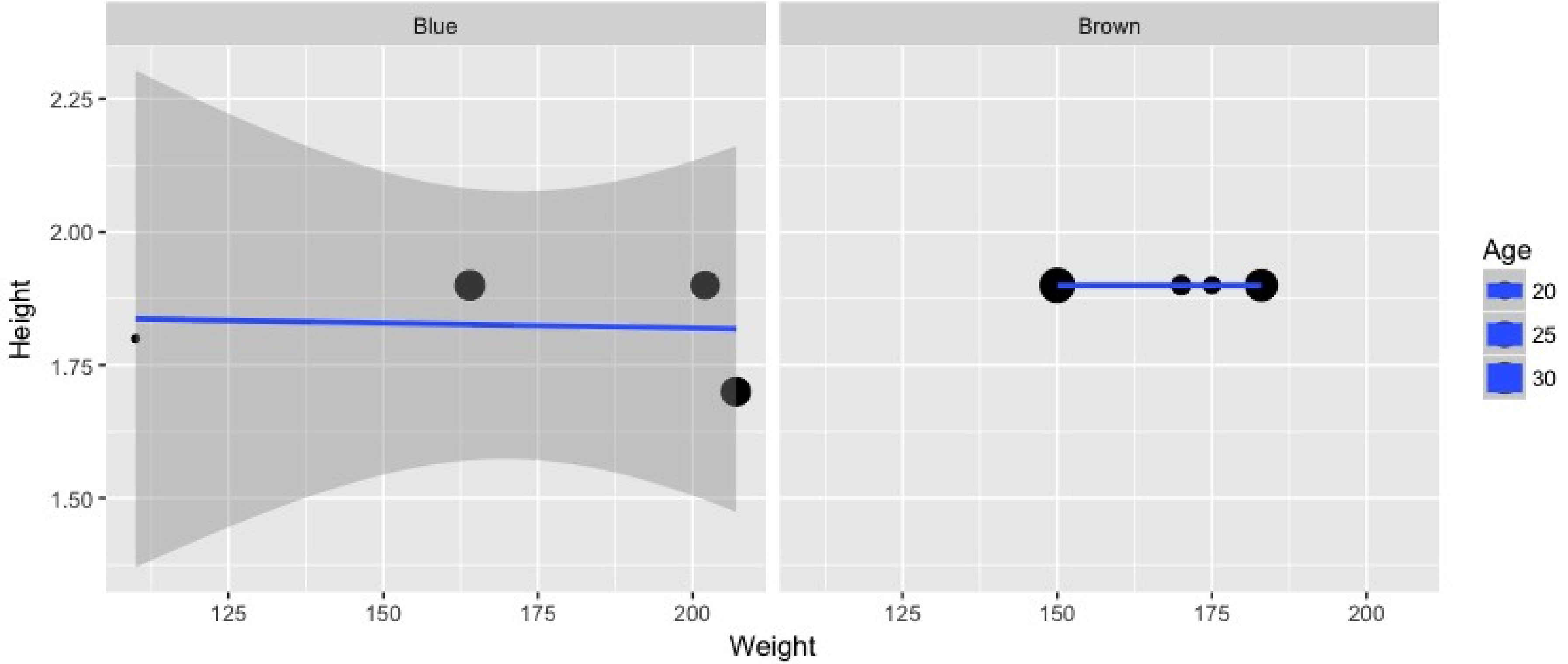
Ggplot!

```
data_people %>% filter(Height, Weight) %>%
ggplot(aes(x = Weight, y = Height, col = Age))
+ geom_point() + geom_smooth()

# Try playing with the settings!!
```



More With Ggplot!

```
data_people %>% filter(Height, Weight) %>%
ggplot(aes(x = Weight, y = Height, size = Age, col =
Age)) + geom_point() + geom_smooth(method = lm) +
facet_wrap(~EyeColour)

#Note: geom_smooth applies a linear model
```



Basic Stats:
Working with *p*-values

- Suppose: We are the producers of two kinds o drinks: green and purple. Each drink comes in a bottle and we would like to know whether the green and the purple drink are filled to the same levels.
- We randomly select 9 bottles from our entire set of 100000 bottles

Comparing Populations

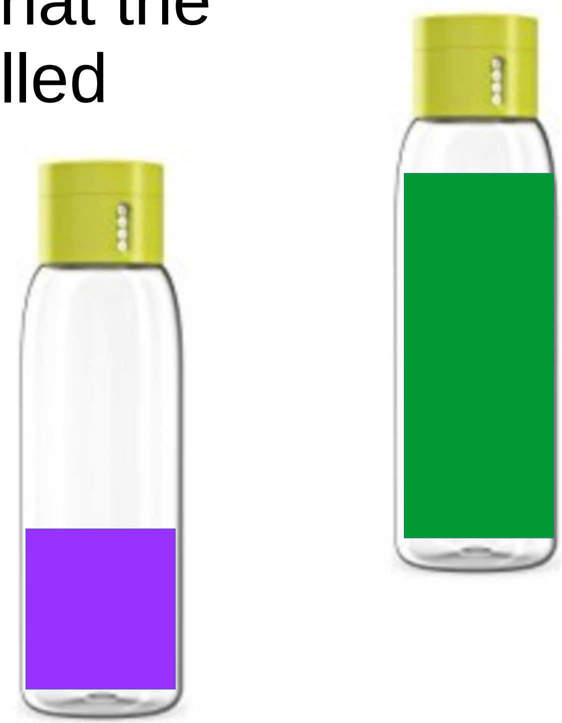
- By inspection,
  - **Purple bottles seem a little under-filled**
  - **Green bottles seem a little over-filled**
- Can we use a statistical test to conclude whether the whole batch is under- or over-filled?

Hypothesis Testing

- We want to know: **Is there a statistically significant difference between the two groups in terms of the average extent to which the bottles are filled?**
  - **Null hypothesis (Ho)**: The bottles are filled the same
  - **Alternative hypothesis (Ha)**: There is a difference between the filling of bottles.
- Remember: we have a sample of ***only nine bottles from the super set of 100000 bottles***.
- Statistics is used to extrapolate from the small set to the larger set.

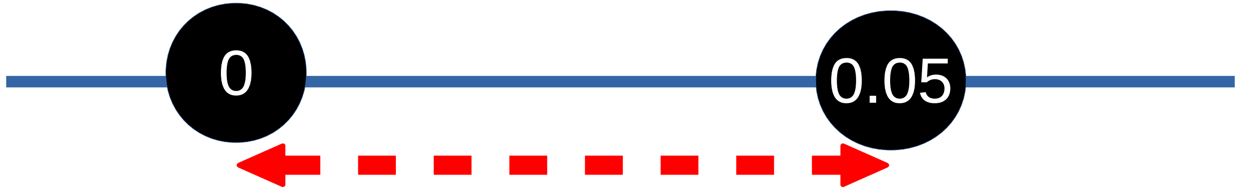
Is Our Sample Telling the Truth?

- We admit that our sample-selection may not necessarily represent our larger stock of bottles:
- The sample selection may still show that the green and purple bottles have been filled differently (*by accident*).



Use *p*-Values

- The p-Value says that we are sure that our sample size that we randomly selected is a good representation of our, larger, superset.
- Use a 95 confidence interval range: Our selected bottles fit within 95 percent of the entire set, meaning, a good representation of the entire set of 100000 bottles.
- **Reject the Null Hypothesis when $p < 0.05$** (when *p* is close to zero)
- Rejecting means that something unnatural is happening.



Is your p-value in here?
Reject H_0

Basic Stats: Run a T-Test

```
data_drinks <- tibble::tribble(
  ~Observation, ~Colour, ~percentFull,
  1,"Green", 70,
  2,"Purple",30,
  3,"Green",50,
  4,"Purple",20,
  5,"Purple",15,
  6,"Green",90,
  7,"Purple",40,
  8,"Green",60,
  9,"Purple",15)
```

Basic Stats: T-Tests

```
data_drinks <- data_drinks %>%
    select(Colour, percentFull) #lose obs. num
#Run the t-test: a comparison of means.
t.test(data = data_drinks, percentFull ~ Colour)
```

 # Check the p-value:

- **If p-val =< alpha = 0.05: reject H0.**
- **If p-val > alpha = 0.05: do not reject H0.**

- **What do we conclude about our *data_drinks*?**

Automate Your T-Test Analysis

```
myOut <- t.test(data = data_drinks, percentFull ~ Colour)
myOut$p.value
rejectOrWhat <- function(pValue){
  if(pValue >= 0.05){
    print("Accept Null Hypothesis")
  }
  else{
    print("Reject Null Hypotheis: something is going on...")
  }}
rejectOrWhat(myOut$p.value)
```

```
#If p-val =< alpha = 0.05: reject H0.
#If p-val > alpha = 0.05: do not reject H0.
```

R’s Built-In Data
Our Built in Data

- R studio (R statistics) has plenty of included data-sets for practicing t-tests work.

```
# find sets
data()
```

```
Data sets in package ‘datasets’:

AirPassengers            Monthly Airline Passenger Numbers 1949-1960
BJsales                  Sales Data with Leading Indicator
BJsales.lead (BJsales)   Sales Data with Leading Indicator
BOD                      Biochemical Oxygen Demand
CO2                      Carbon Dioxide Uptake in Grass Plants
ChickWeight              Weight versus age of chicks on different diets
DNase                    Elisa assay of DNase
EuStockMarkets           Daily Closing Prices of Major European Stock Indices,
                         1991-1998
Formaldehyde             Determination of Formaldehyde
HairEyeColor             Hair and Eye Color of Statistics Students
Harman23.cor             Harman Example 2.3
```

Meta-Data from Data

Choose “AirPassengers” having only one column.

```
View(AirPassengers)
# general meta data
summary(AirPassengers)
```

	AirPassengers
1	112
2	118
3	132
4	129
5	121
6	135
7	148
8	148

```
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  104.0   180.0   265.5   280.3   360.5   622.0
```

What’s in the Summary()

- Min: Minimum value (lower bound)
- Max: Maximum value (upper bound)
- Mean: average value across the set
- Median:
  - The middle number (if num of observations is odd)
  - The average of the middle pair (if num of observations is even)

```
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
```

Medians

Median

First, arrange the observations in an ascending order.

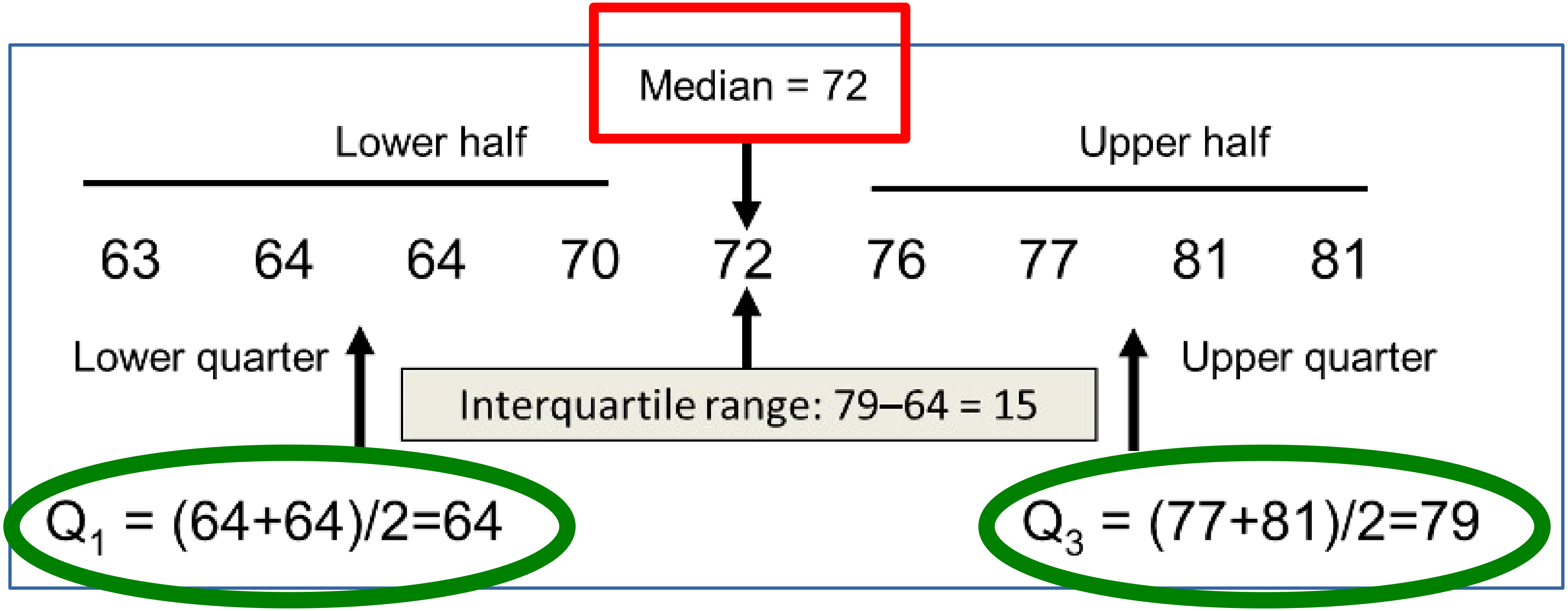
If the number of observations (n) is odd: the median is the value at position

$$\left(\frac{n+1}{2}\right)$$

If the number of observations (n) is even:

1. Find the value at position $\left(\frac{n}{2}\right)$
2. Find the value at position $\left(\frac{n+1}{2}\right)$
3. Find the average of the two values to get the median.

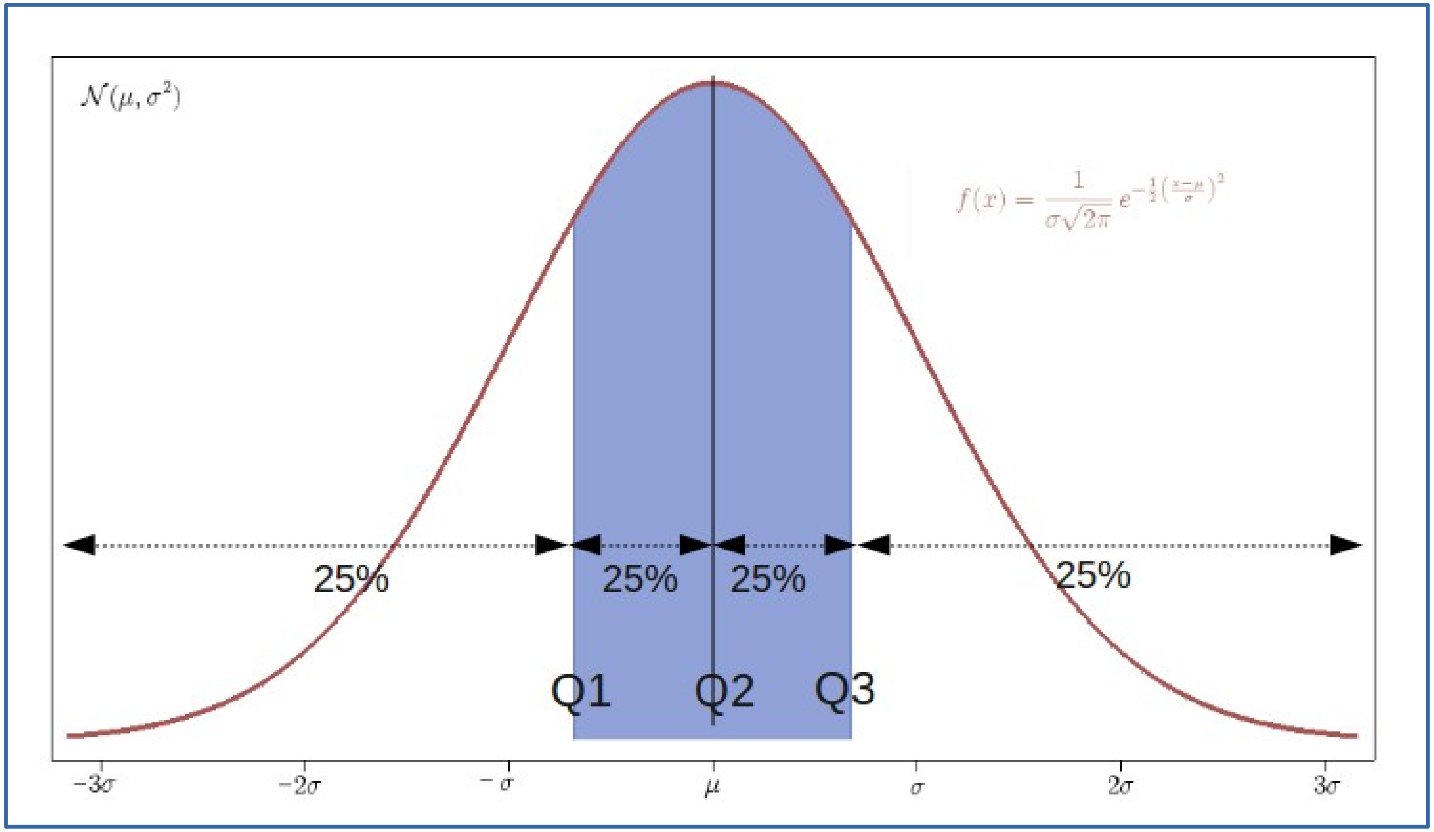
Medians



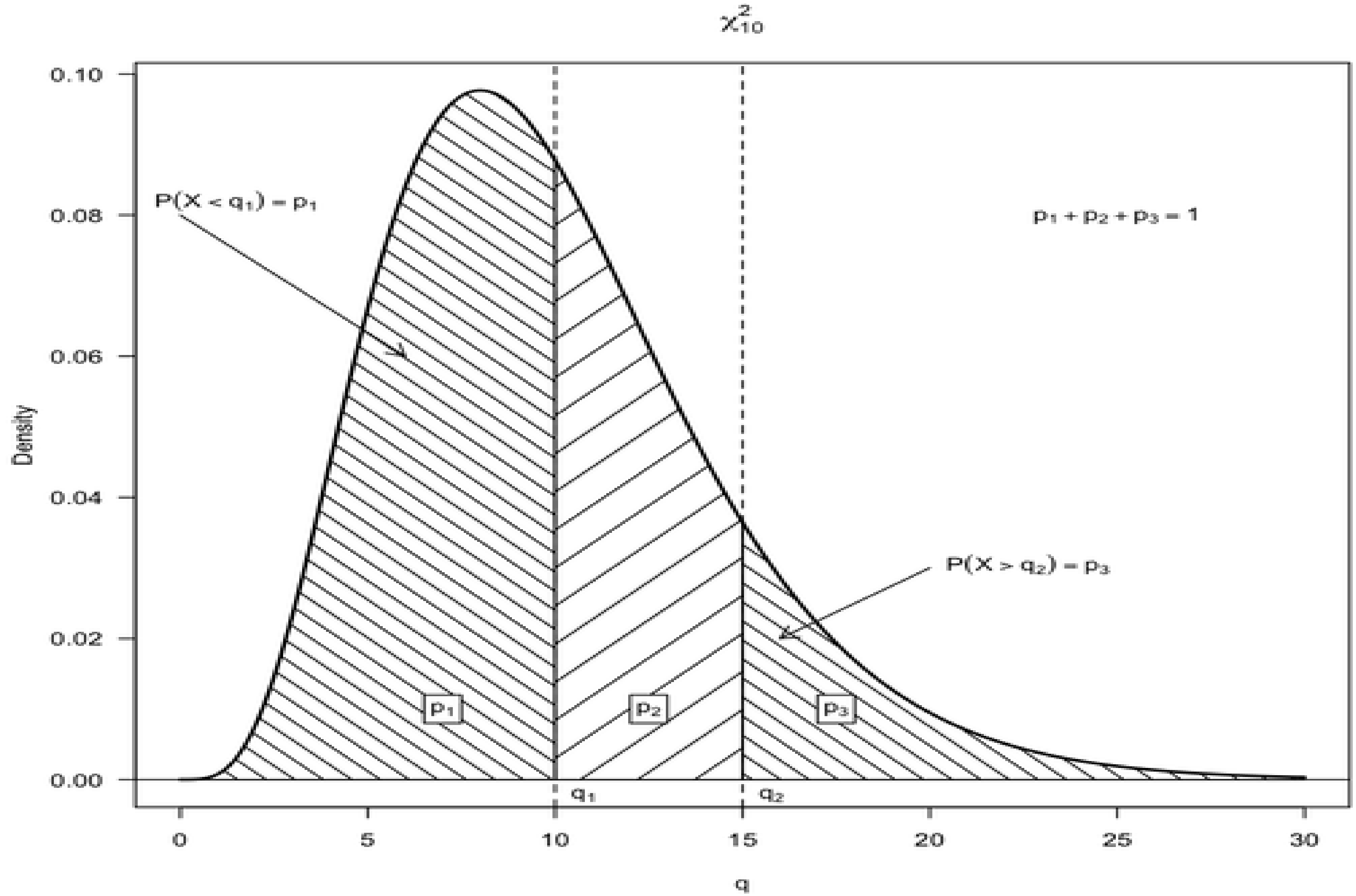
- What does Q1 and Q3 indicate?
  - Quantiles: allow us to determine placements in the set of numbers

Quantiles

- Quantiles are cut-points dividing the range of a probability distribution into contiguous intervals with equal probabilities, or that divide the sample's observations similarly



Quantiles: Help to Study Skews



Quantiles

```r
# find the quantiles of the following set.
qnums <- c(3, 6, 7, 8, 8, 10, 13, 15, 16, 20)
summary(qnums)
```

```
> qnums <- c(3, 6, 7, 8, 8, 10, 13, 15, 16, 20)
> summary(qnums)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    7.25    9.00   10.60   14.50   20.00
```

Finding Quantiles

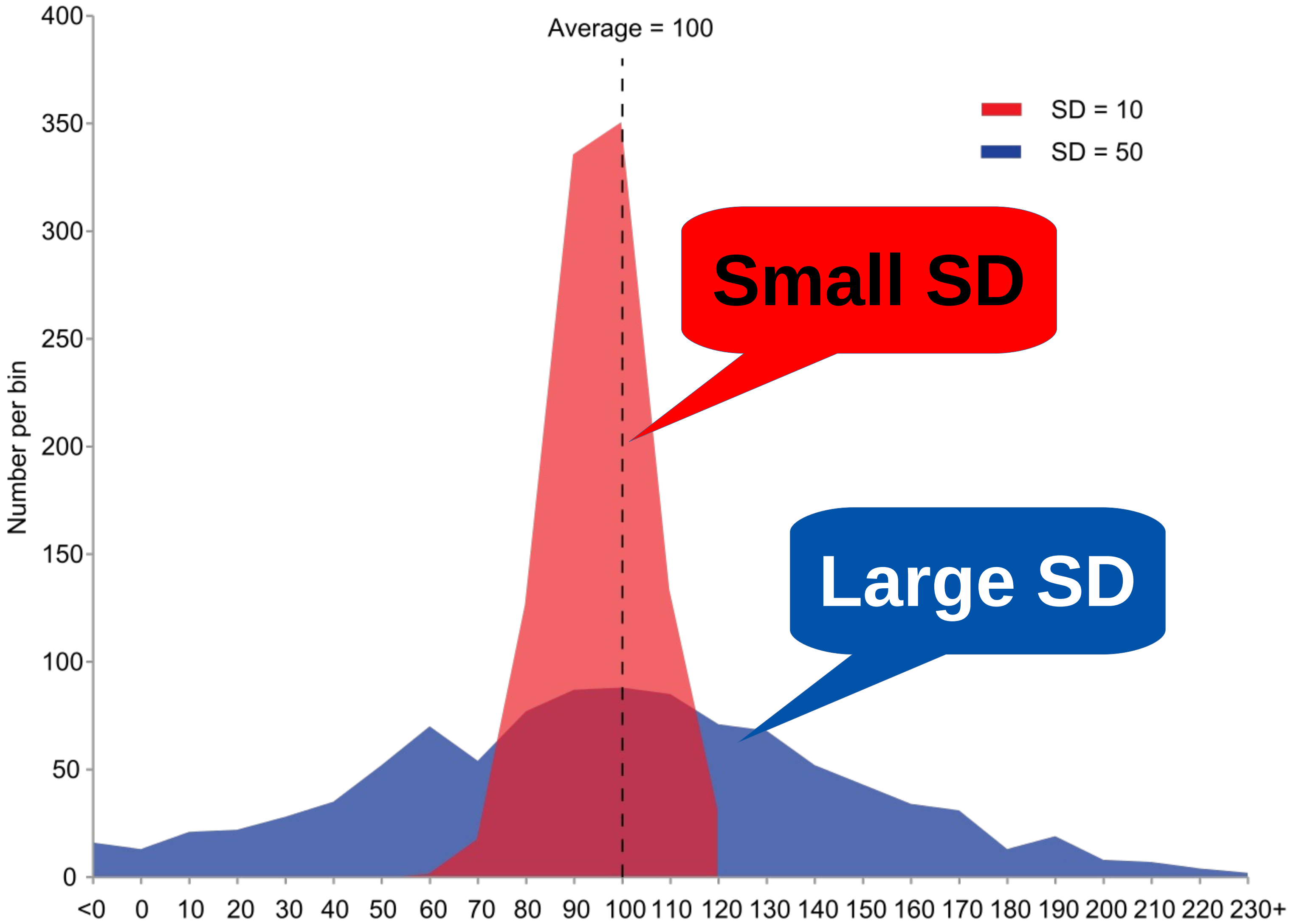
- Finding 1st and 3rd quantiles is to determine the positions at the ¼ and ¾ marks, respectively.

Quartile	Calculation	Result
Zeroth quartile	Although not universally accepted, one can also speak of the zeroth quartile. This is the minimum value of the set, so the zeroth quartile in this example would be 3.	3
First quartile	The rank of the first quartile is 10×(1/4) = 2.5, which rounds up to 3, meaning that 3 is the rank in the population (from least to greatest values) at which approximately 1/4 of the values are less than the value of the first quartile. The third value in the population is 7.	7
Second quartile	The rank of the second quartile (same as the median) is 10×(2/4) = 5, which is an integer, while the number of values (10) is an even number, so the average of both the fifth and sixth values is taken—that is (8+10)/2 = 9, though any value from 8 through to 10 could be taken to be the median.	9
Third quartile	The rank of the third quartile is 10×(3/4) = 7.5, which rounds up to 8. The eighth value in the population is 15.	15
Fourth quartile	Although not universally accepted, one can also speak of the fourth quartile. This is the maximum value of the set, so the fourth quartile in this example would be 20. Under the Nearest Rank definition of quantile, the rank of the fourth quartile is the rank of the biggest number, so the rank of the fourth quartile would be 10.	20

Original Data: 3, 6, 7, 8, 8, 10, 13, 15, 16, 20

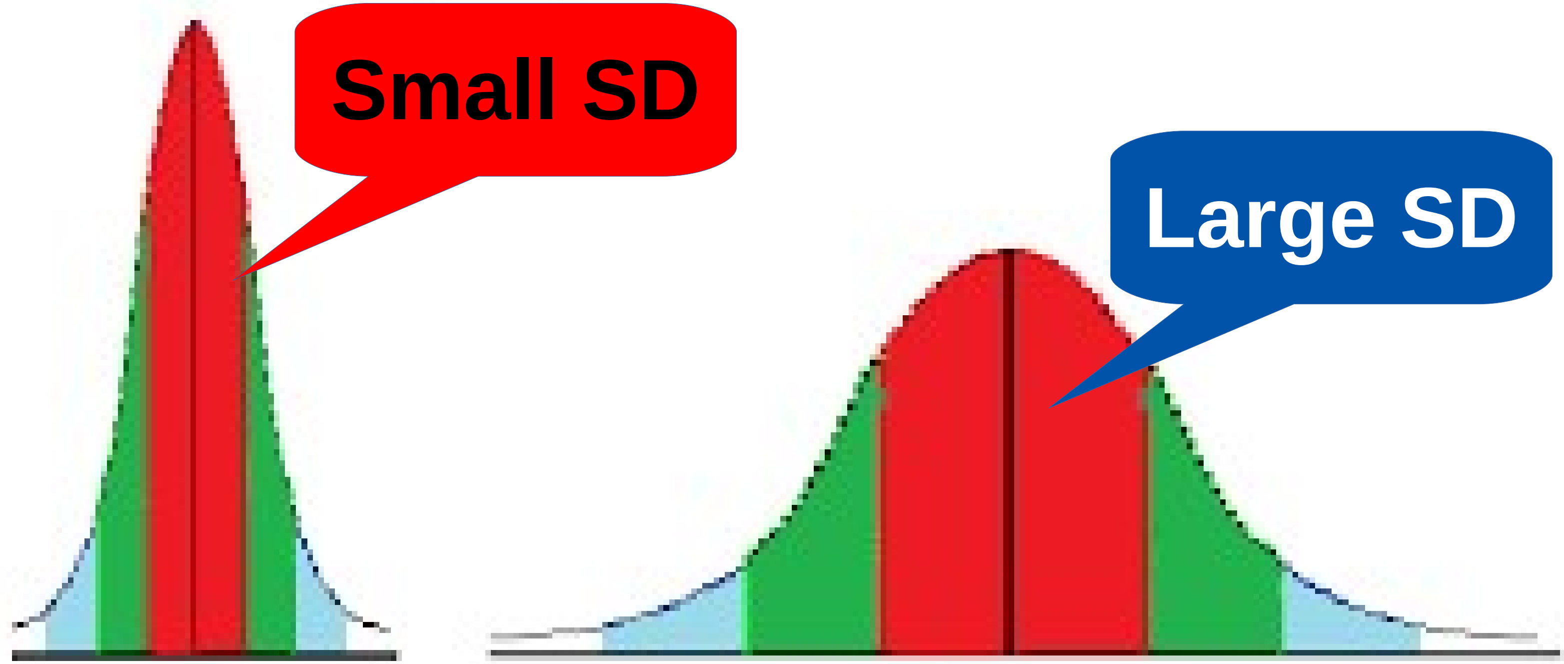
Standard Deviation

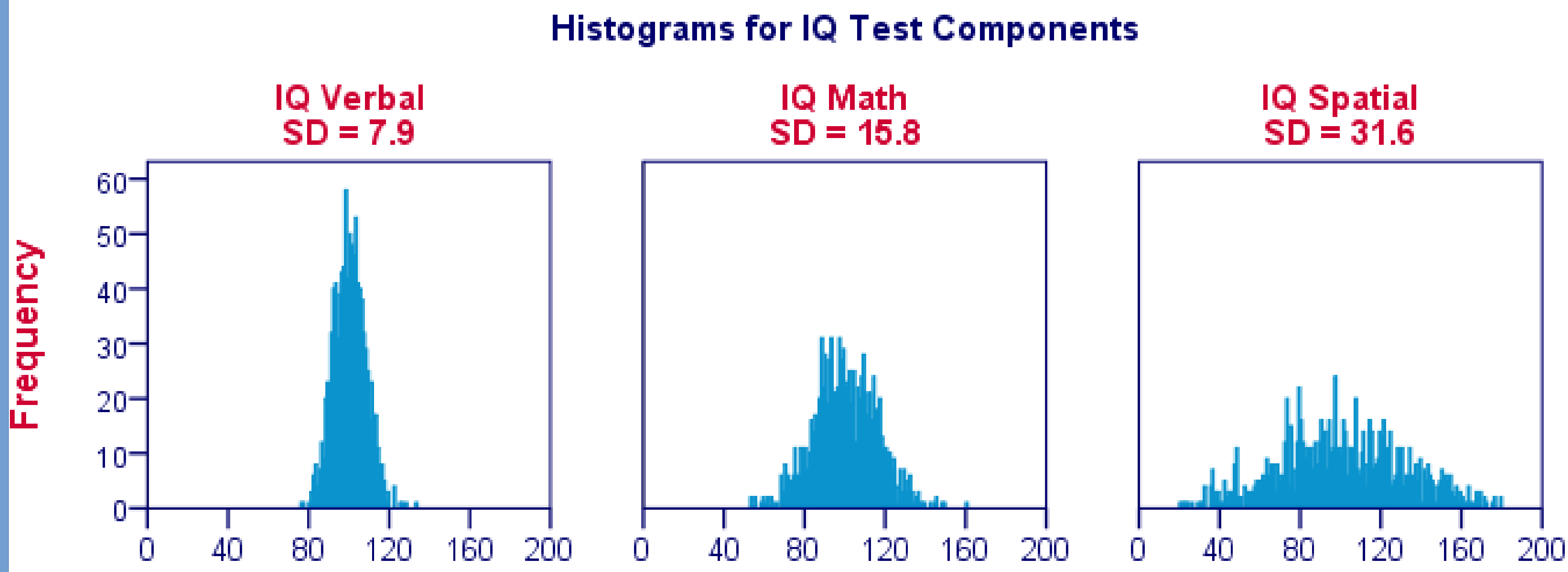
- A quantity calculated to indicate the extent of deviation for a group as a whole.



Standard Deviation

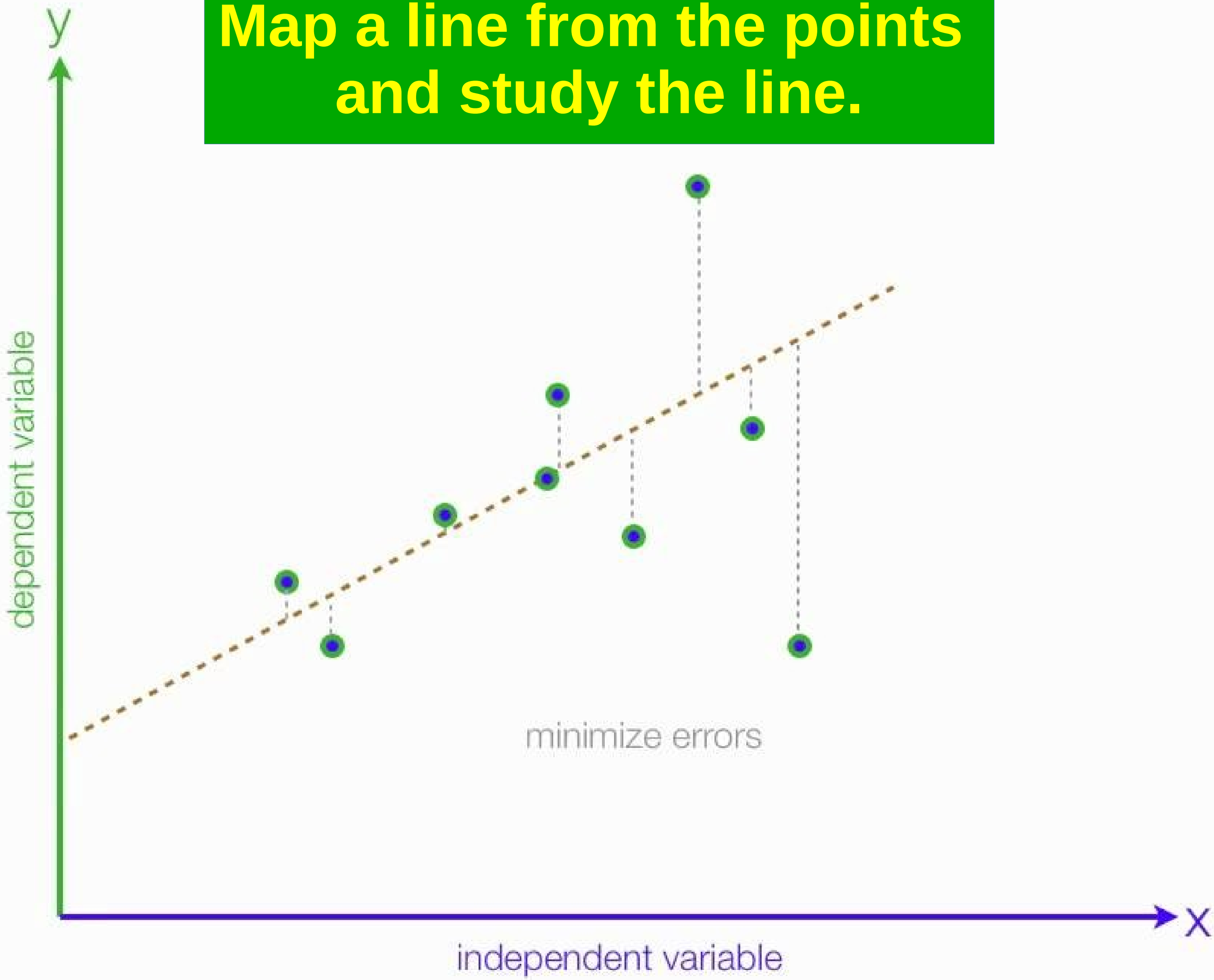
- A measure that is used to quantify the amount of variation or dispersion of a set of data values.
- A **low standard deviation** indicates that the data points tend to be **close to the mean** (also called the expected value) of the set
- A **high standard deviation** indicates that the data points are **spread out** over a wider range of values.


Small SD
Large SD


Histograms for IQ Test Components
IQ Verbal
SD = 7.9
IQ Math
SD = 15.8
IQ Spatial
SD = 31.6
Frequency
0 10 20 30 40 50 60
0 40 80 120 160 200

Linear Regression

Map a line from the points and study the line.



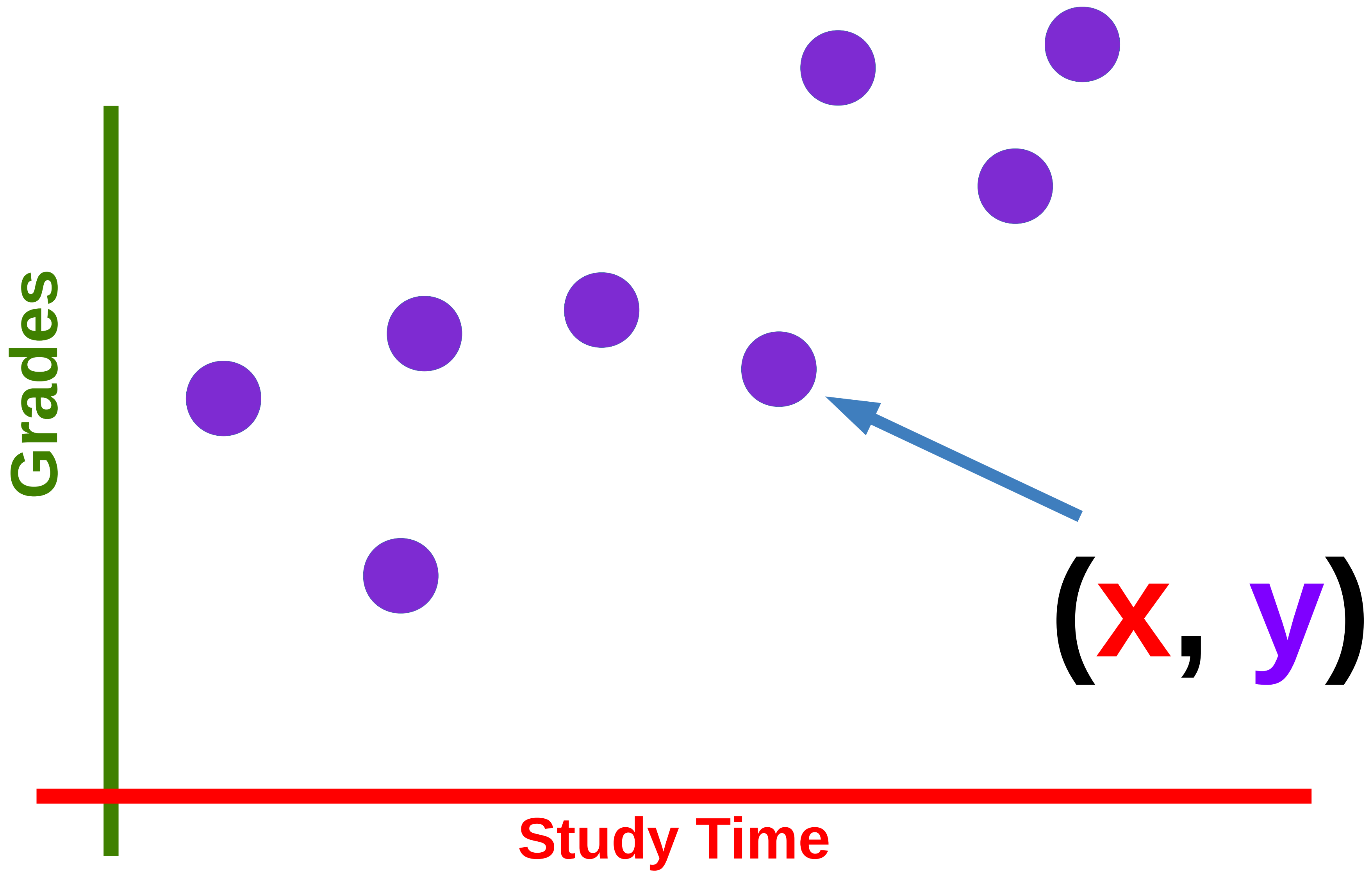
Linear Regression

- Is one thing able to influence another thing?
- A linear approach for modeling the relationship between a scalar dependent variable ***y*** and one or more explanatory variables, or independent variables, denoted by ***x***.
- *Simple linear regression*: Single explanatory variable; **models x and y**
- *Multiple linear regression*: More than one explanatory variable (***y's***); **models x and y1, y2**

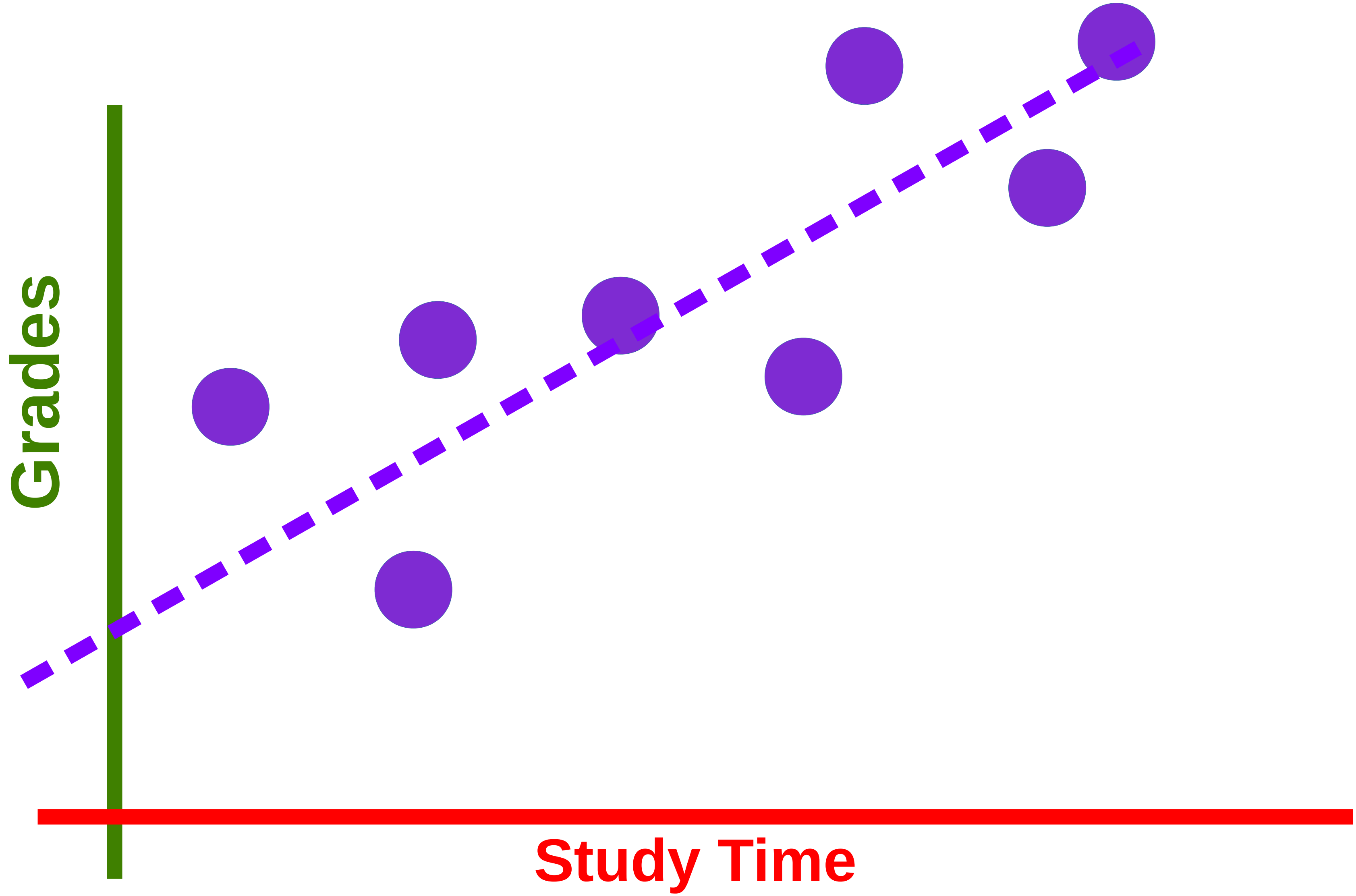
Linear Regression

- A straight line is drawn through a dot cloud.
- As the independent variable progresses, what is the dependent variable doing? Is there a relationship?
- The line has a y-intercept and a slope and can be used to determine the positive or negative relationship

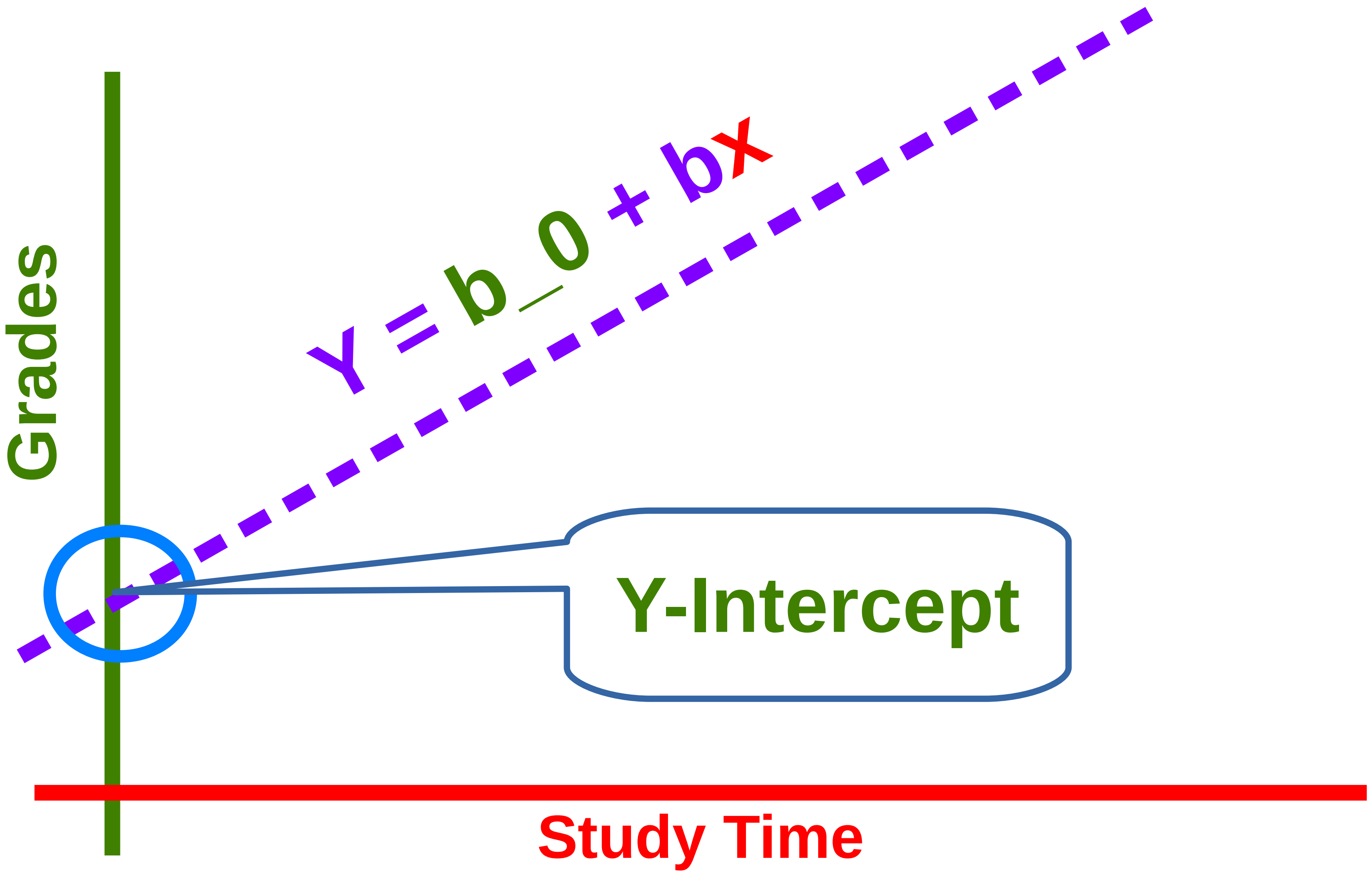
Plot *Study Time* to *Grades* Points



Draw Line Through Points



Intercept and Slope: Positive Relationship



Control and Training groups

Linear Regression

```
Ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
Trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(Ctl, Trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1) # omitting intercept
summary(lm.D9)
```

- **H0: (Null Hyp) there is no relationship between vars, m = 0**
- **Ha: (Alt Hyp) There is a relationship between vars, m!= 0**

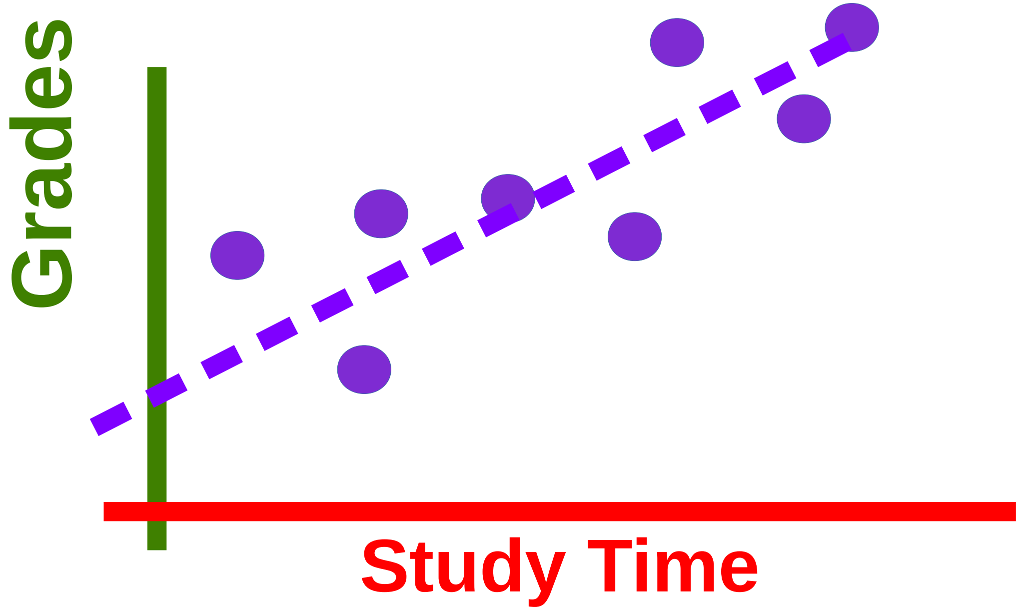
  # Check the p-value:
  - **If p-val =< alpha = 0.05: reject H0.**
  - **If p-val > alpha = 0.05: do not reject H0.**

Regression Assumptions

- The regression has five key assumptions:
  - Linear relationship
  - Multivariate normality
  - No or little multicollinearity
  - No auto-correlation
  - Homoscedasticity

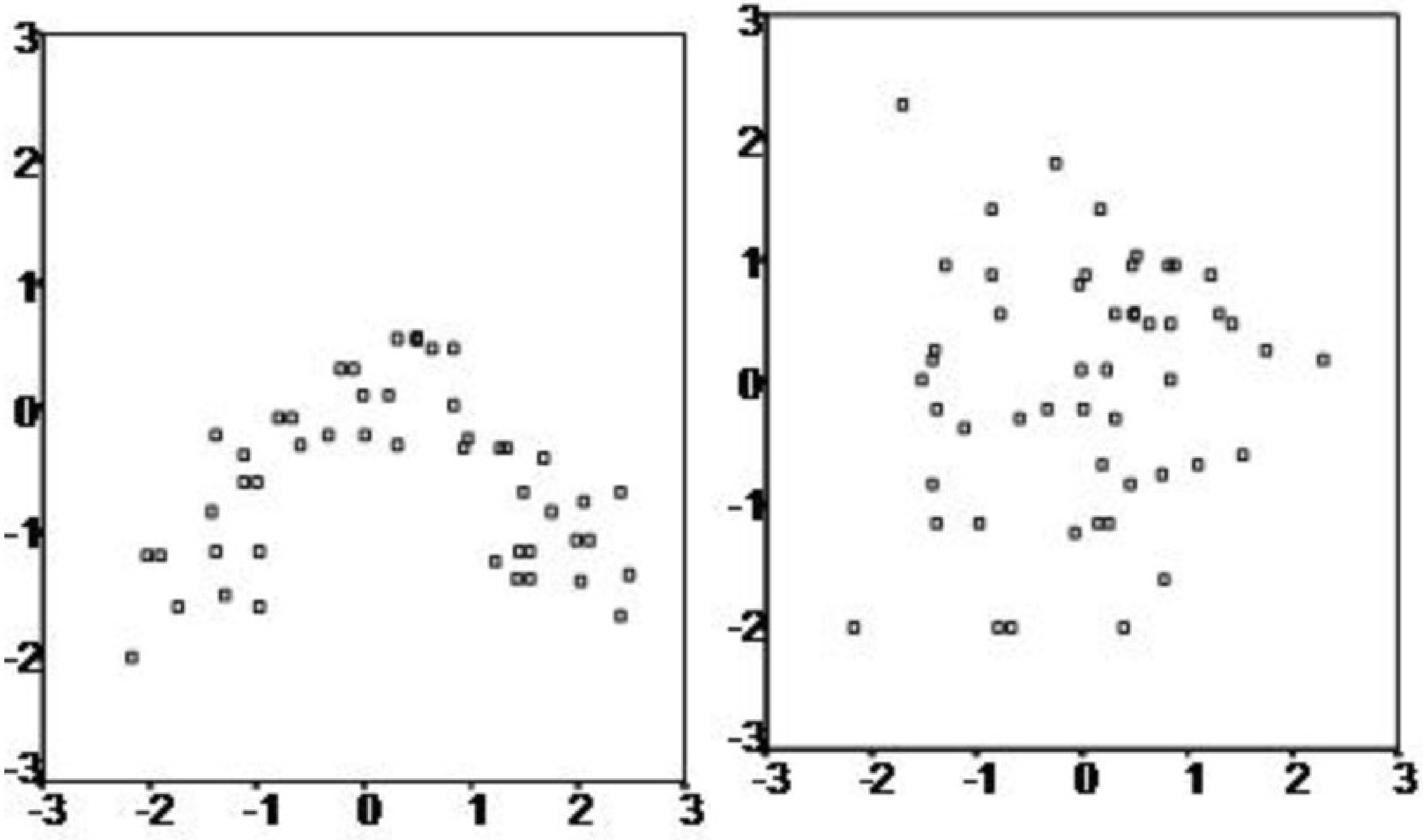
Linear Relationship

- Linear regression needs the relationship between the independent and dependent variables to be *linear*.
- Check for outliers linear regression is sensitive to outlier effects.



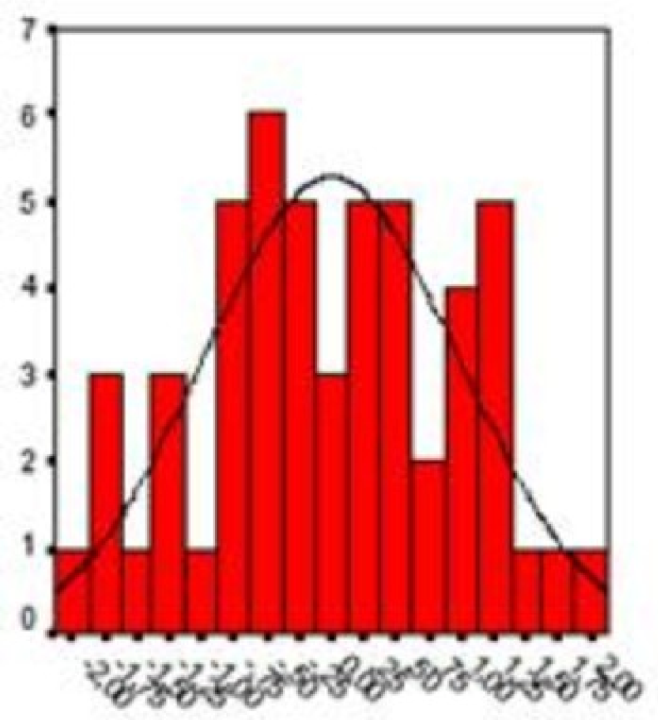
Linear Relationship

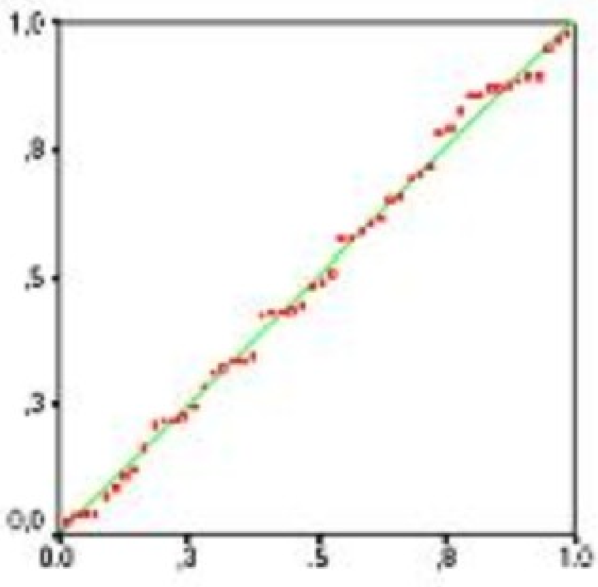
- Scatter plots: See where no and little linearity is present.



Multivariate Normality

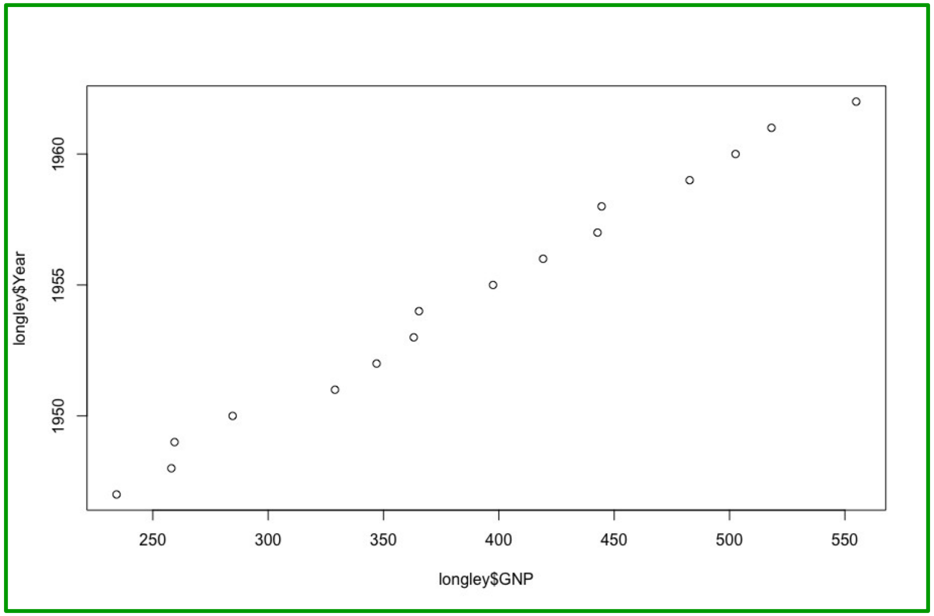
- The data must be of a *normal* distribution
- Check this with a QQ-plot

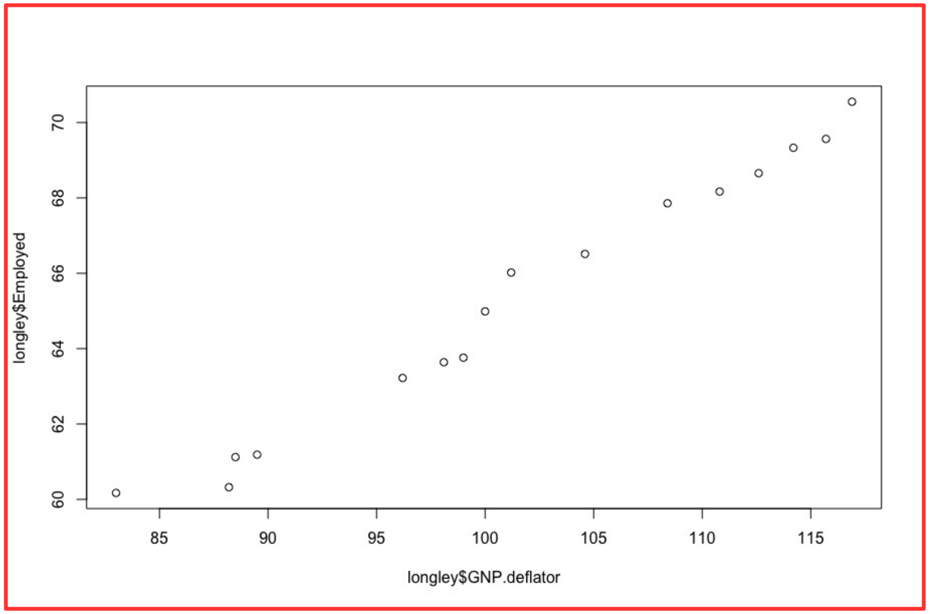




Multivariate Normality

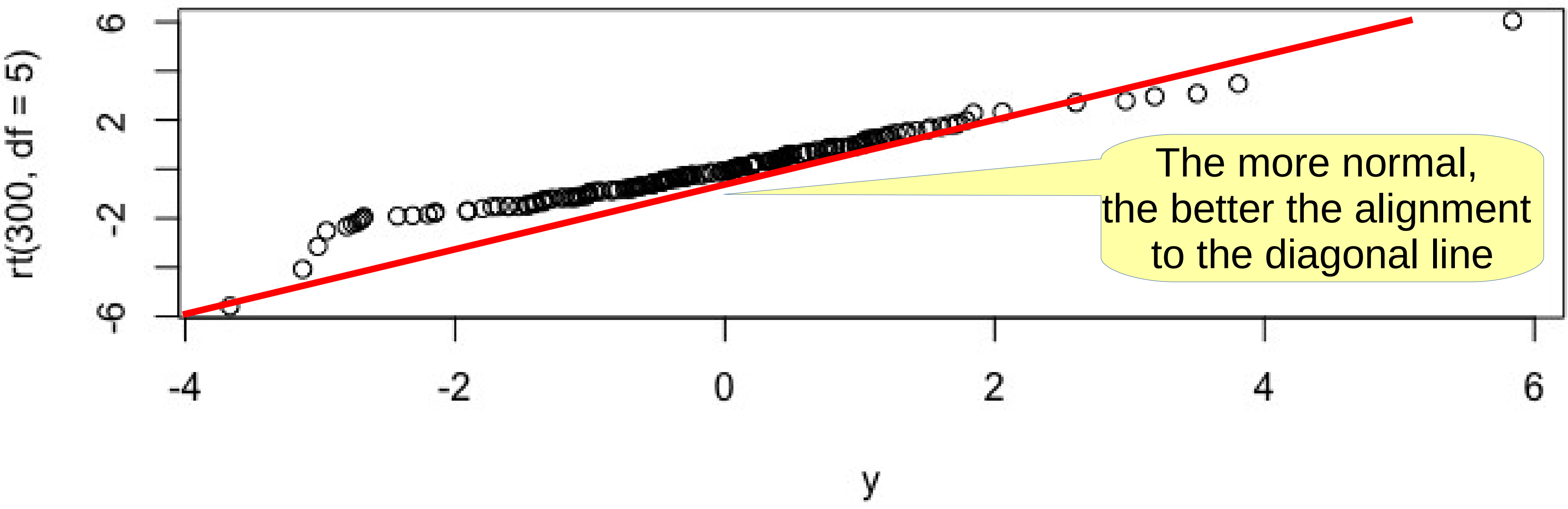
```
# Good
qqplot(x = longley$GNP, y = longley$Year)
# Not so good
qqplot(x = longley$GNP.deflator, y =
longley$Employed)
```





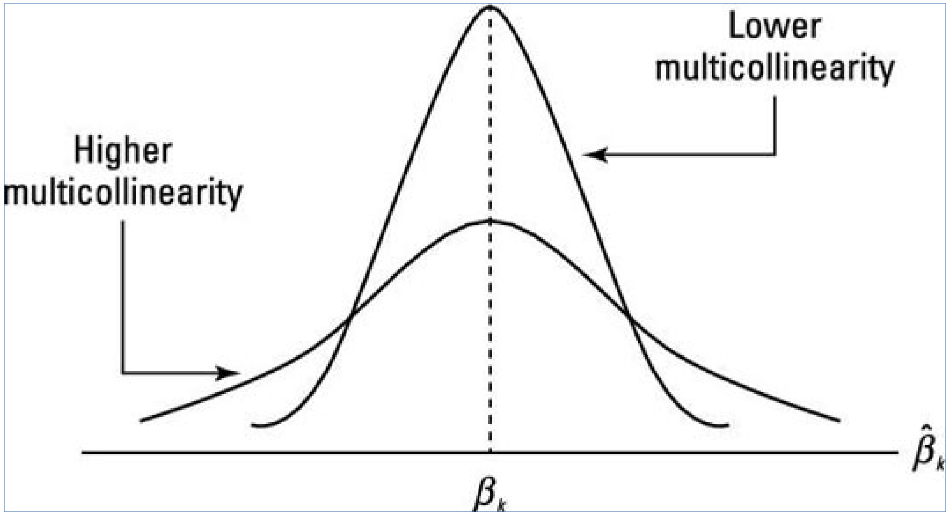
Detecting Normality: QQ-Plot

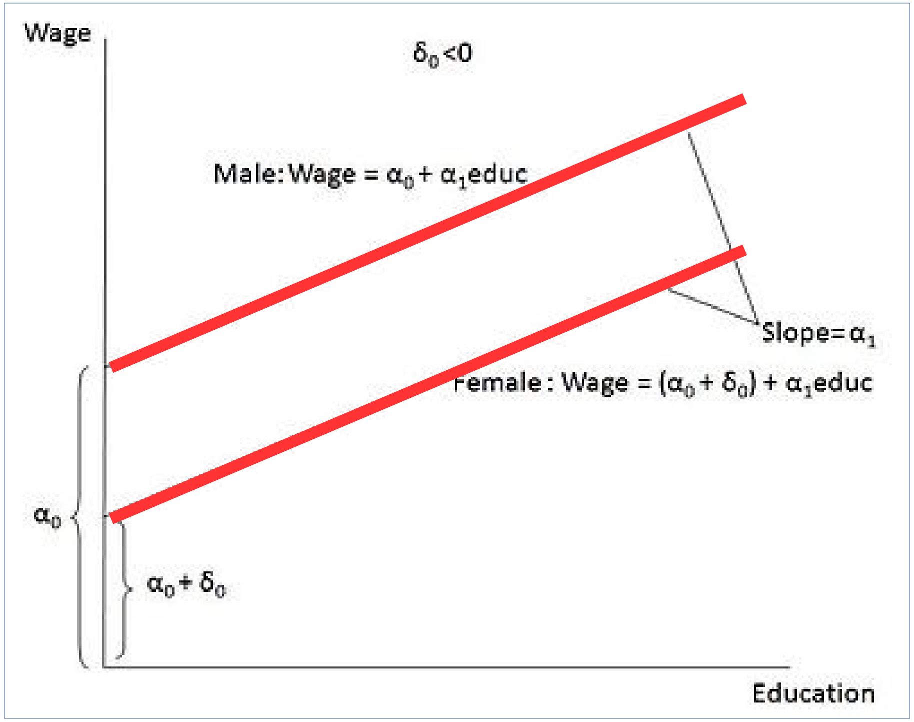
```
y <- rt(200, df = 5) #random
qqnorm(y); qqline(y, col = 2)
qqplot(y, rt(300, df = 5))
```



Multicollinearity

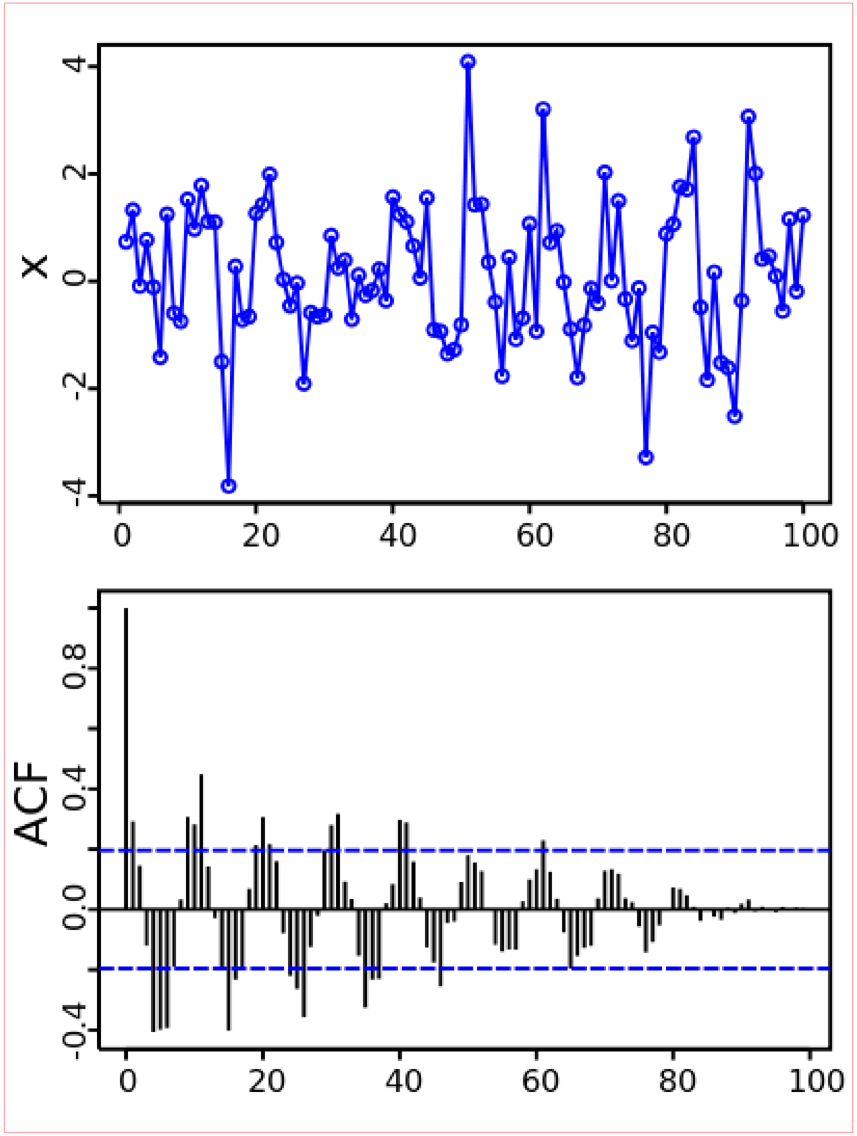
- A phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.
- Same slope; same line





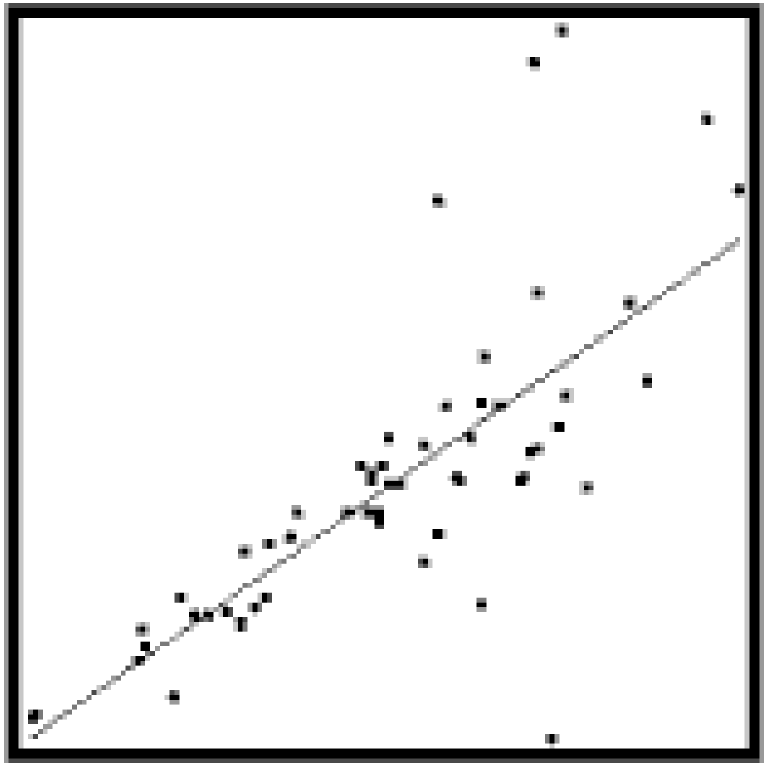
No Auto-correlation

- The correlation of a signal with a delayed copy of itself as a function of delay
- Ex: A plot of a series of 100 random numbers concealing a sine function. The sine function revealed in a correlogram produced by autocorrelation.
- Result: Non random output



Must Have Homoscedasticity

- Data sets in the regression must have the same variance (same quality of being different or divergent)
- This assumption means that the variance around the regression line is the same for all values of the predictor variable (X).
- The plot shows a violation of this assumption. For the lower values on the X-axis, the points are all very near the regression line.



Must Have Homoscedasticity

- Heteroscedasticity examples below
- Differing variance is bad for regression models.