R for Biologists: a powerful statistical tool in research

Afaf Saaidi, June 2021

Objective of this training: To introduce biologists to the use of R for their statistical analysis. We provide an overall review of statistical measurement tools (mean, median . . .) and the appropriate hypothesis test to use when comparing samples.

#Time allocation: 4 sessions of 50' each.

Table of contents

The artcile Ali and Bhaskar (2016) was used as the basis for the statistical content.

1. Get initiated to R

Rstudio

How to set-up Rstudio? Rstudio is a friendly user environment when one can use R code. Two options are available to you to acquire it:

Option1: Local installation:

- Install R software: https://cran.rstudio.com/
- Install RStudio: https://www.rstudio.com/products/rstudio/download/#download

Option2: Use the cloud version:

- Create an account for free on RStudio cloud https://rstudio.cloud

Four windows appear: top left (code), bottom left console, top right (environment with every objects we create) and top bottom (plots) .

Why R and principle of packages:

```
-R is an open-source and *freely* available
-R is a strong scripting language for statistical and quantitative analysis
-R allows to import data from Excel, mysql...
-R is flexible and grows quickly (As of November 2020, more than 16,000 free packages are available).
```

Manipulate data

Existing datasets Several R Datasets are available at: https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html

```
###### We will use an existing database " PlantGrowth" that  Results from an experiment to compare yiel

## assign the dataset to Mydata, 30 obs. with two variables
Mydata1 <- PlantGrowth
## Get first instances of the dataframe with the variable labels
head(Mydata1)
```

```
##   weight group
## 1   4.17  ctrl
## 2   5.58  ctrl
## 3   5.18  ctrl
## 4   6.11  ctrl
## 5   4.50  ctrl
## 6   4.61  ctrl
```

```

## specify the file name, be sure to upload it to the working directory
filename="RmrpProfiling.csv"
## Assign content to Mydata2 dataframe
Mydata2<-read.csv(filename)
## Get first instances of the dataframe with the variable labels
head(Mydata2)
```

Import files

```
##   Helix opening closing length frequency RNAprofile Nature
## 1  P12b     122     184      7         0 NotSampled Native
## 2    P7      80     195      3         0 NotSampled Native
## 3    P4      75     247      5         0 NotSampled Native
## 4    P9     100     115      6      1000   Featured Native
## 5    H1       3     263      9      1000   Featured Native
## 6   P3b      37      52      5       981   Featured Native
```

```

## Specify the dataframe in x, and specify the file name in file.
write.table(x = Mydata1, file = "Mydata1.csv")
```

Export files

2- Descriptive statistics

Variables

We remind that the variables on which we can do statistical analysis are *quantitative variables*.

- Quantitative or numerical data are subdivided into **discrete** (counted integer) and **continuous** measurements.

Measurments

Descriptive statistics allow to describe the relationship between variables in a sample or in a population by measuring how the data is organized around a central location (central tendency) and the spread to extremes (degree of dispersion).

Descriptive statistics provide a summary of data in the form of **mean**, **median**, **mode**…

```
## Get statistics for all variables in the dataframe
summary(Mydata1)
```

```
##      weight       group   
##  Min.   :3.590   ctrl:10  
##  1st Qu.:4.550   trt1:10  
##  Median :5.155   trt2:10  
##  Mean   :5.073            
##  3rd Qu.:5.530            
##  Max.   :6.310            
```

```
## Focus on the quantitative variable "weight":

Poids<-Mydata1$weight

Poids
```

```
##  [1] 4.17 5.58 5.18 6.11 4.50 4.61 5.17 4.53 5.33 5.14 4.81 4.17 4.41 3.59 5.87
## [16] 3.83 6.03 4.89 4.32 4.69 6.31 5.12 5.54 5.50 5.37 5.29 4.92 6.15 5.80 5.26
```

```
##1- Measures of the center:
#1-1 mean is the average of all values in the sample

mean(Poids)
```

```
## [1] 5.073
```

```
# 1-2 median is the central value: arrange values in ascending or descending, get the middle value.

median(Poids)
```

```
## [1] 5.155
```

```
# 1-3 mode is the most common value

mode(Poids)
```

```
## [1] "numeric"
```

```
##2-Measures of the spread
#2-1 Range =max-min

range(Poids)
```

```
## [1] 3.59 6.31
```

```
#2-2 Quartiles devide your dataset into 4 parts. Q1 (25%),Q2 (50%)...
quantile(Poids)
```

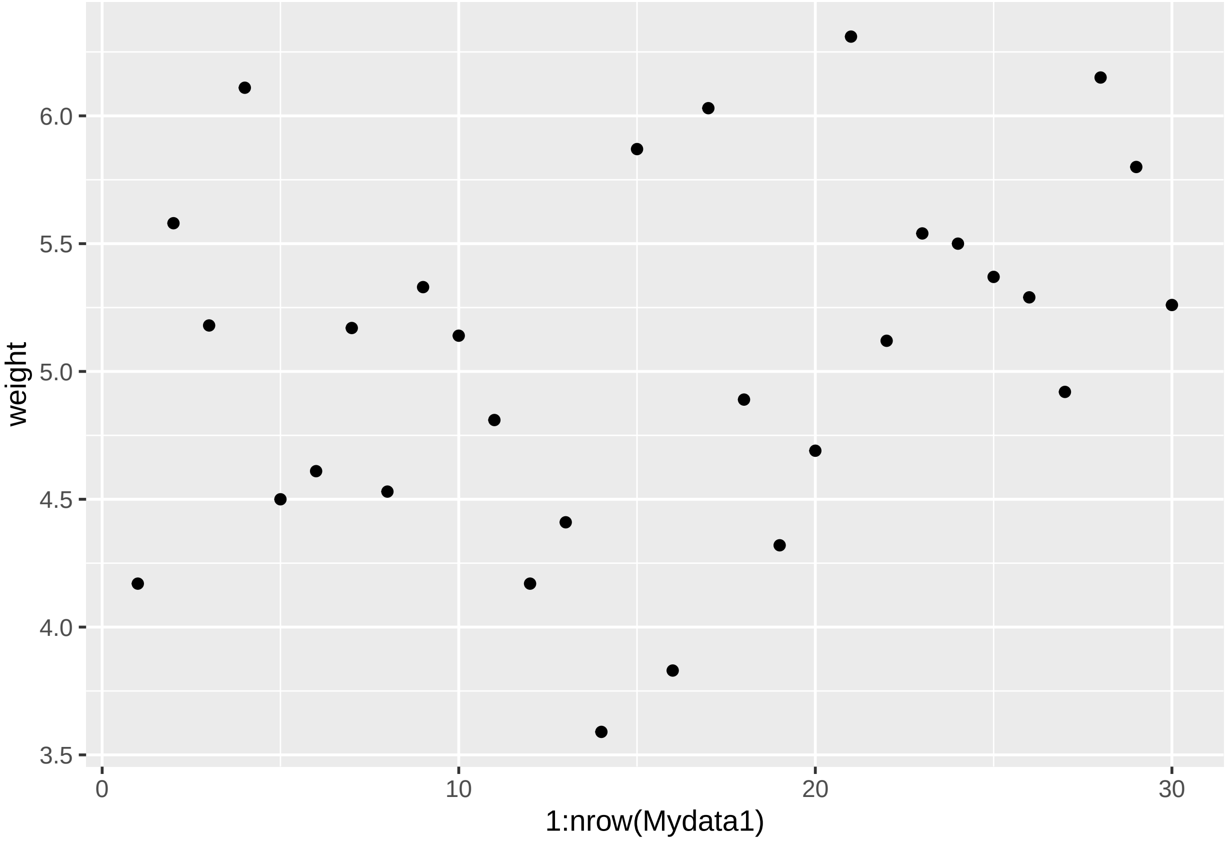
```
##    0%   25%   50%   75%  100% 
## 3.590 4.550 5.155 5.530 6.310 
```

Let us plot the variable "Poids":

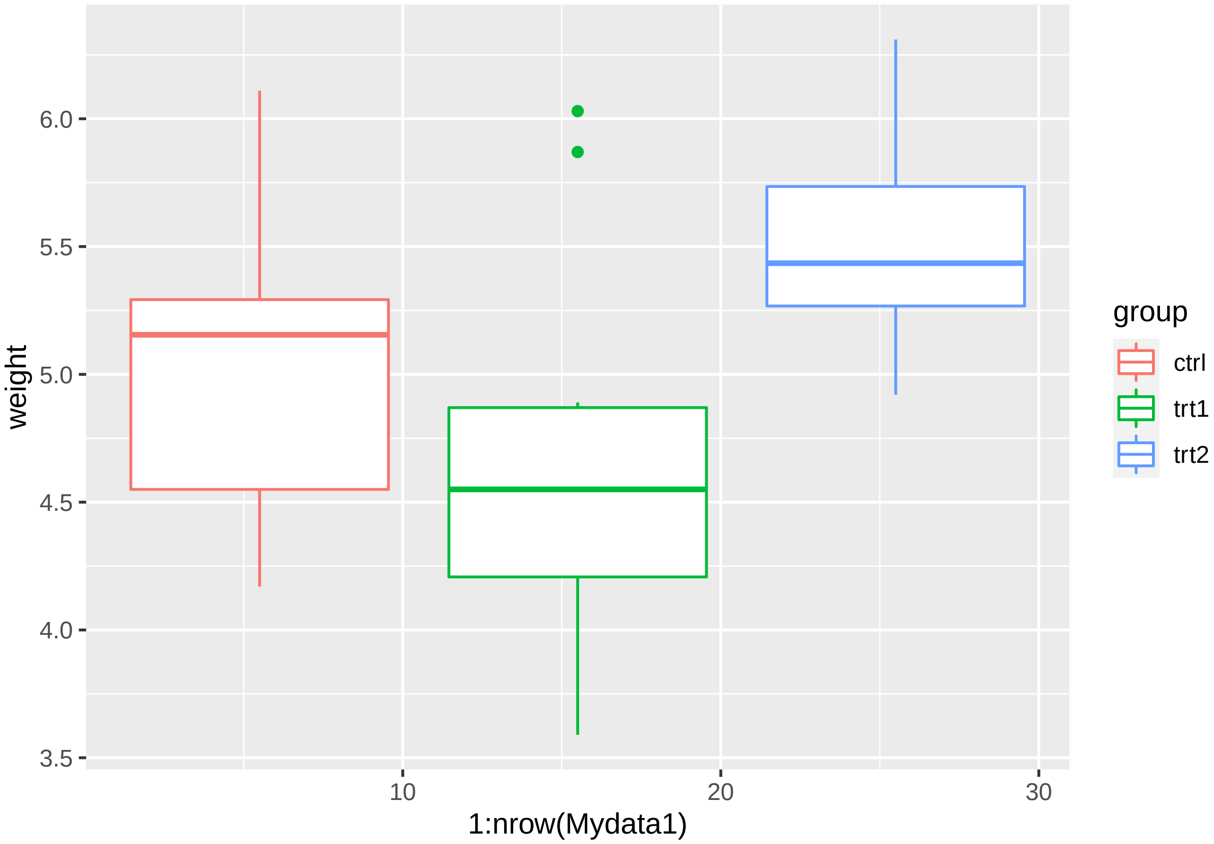
```
## We will be using ggplot2 package, first we should install it than call it:

#install.packages("ggplot2")                              # Install ggplot2 package
library("ggplot2")                                       # Load ggplot2
```

```
## Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame'
## when loading 'dplyr'
```

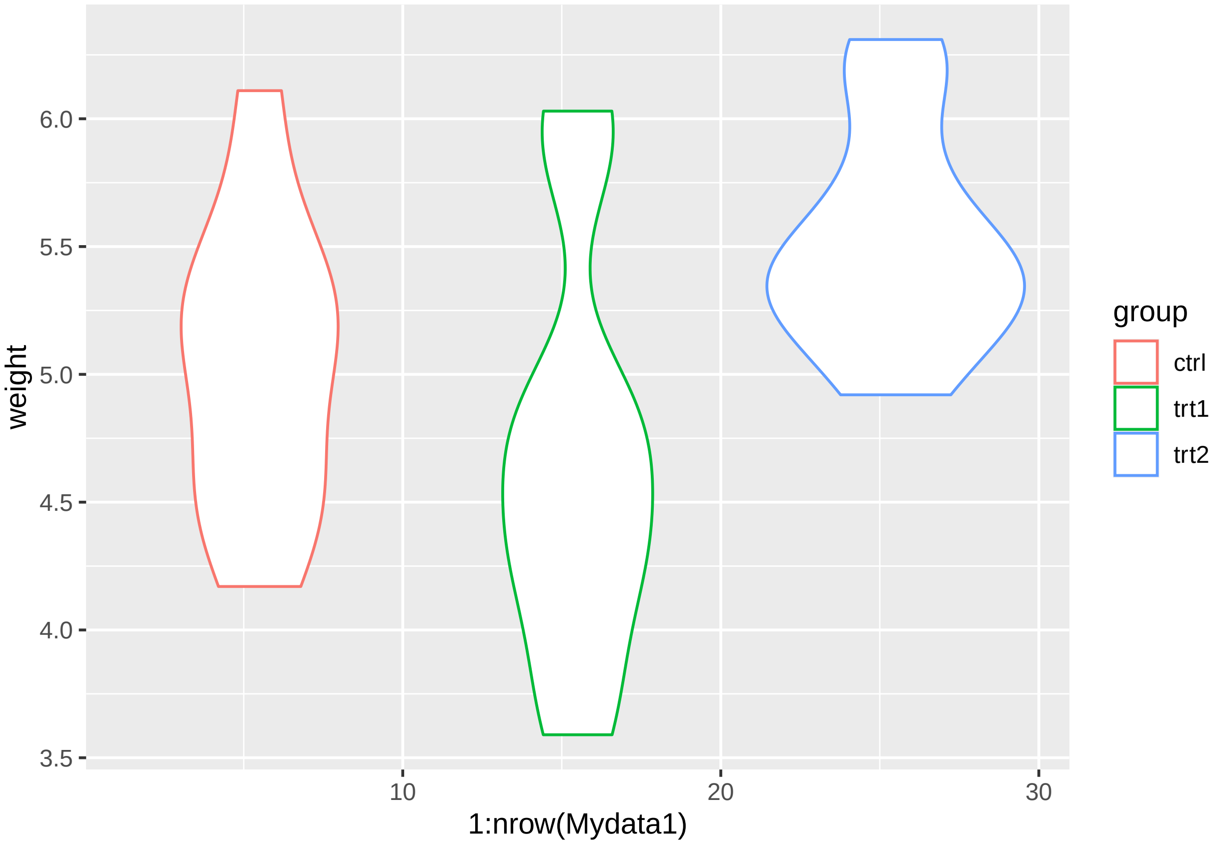
```
## Let s create a variable P1 that will contain the plot parameters

P1 <- ggplot(Mydata1, aes(x = 1:nrow(Mydata1), y =weight))     # We are in the presence of an unidimens

P1 + geom_point()    # plot dots
```



```
P1+ geom_boxplot(aes(y=weight,color=group))  # plot boxplot
```

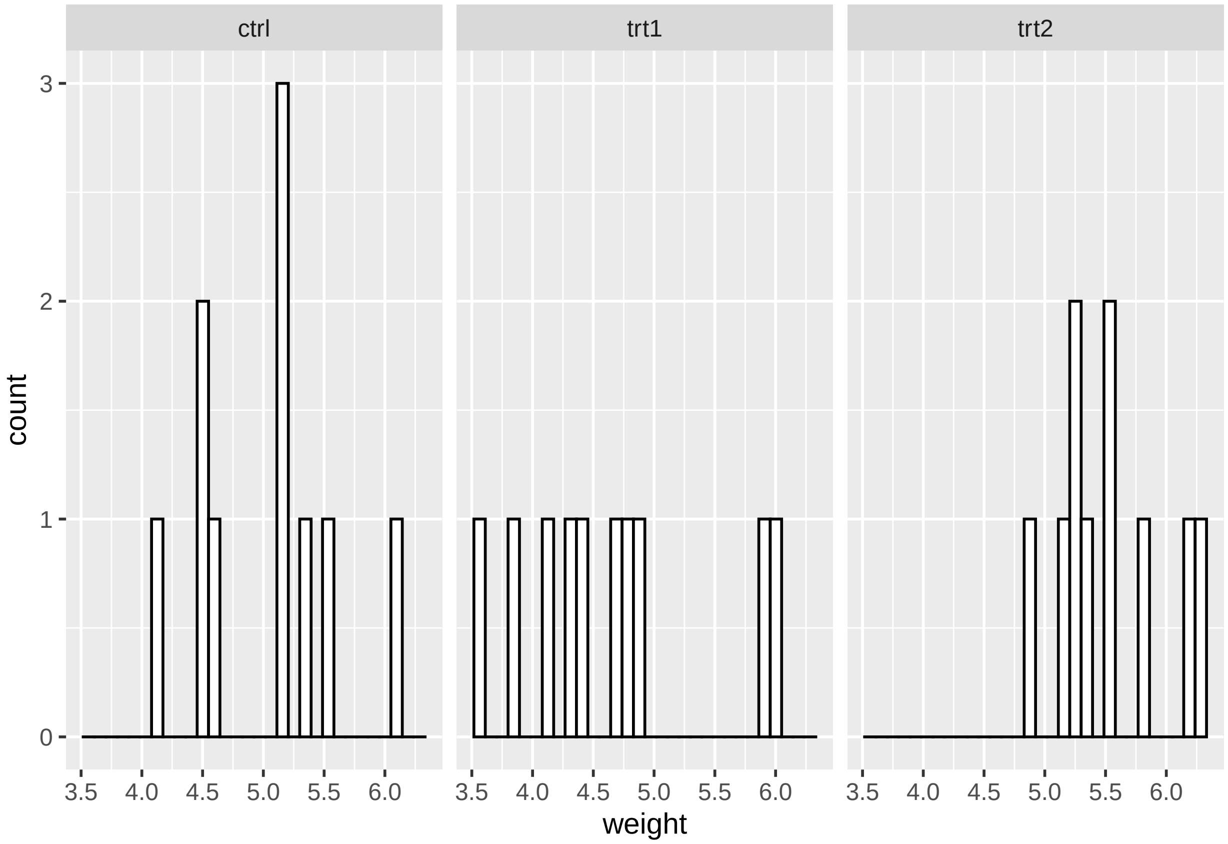


```
P1+geom_violin(aes(y=weight,color=group))  # a more fancy  boxplot showing more the points density
```



```
P2<-ggplot(Mydata1, aes(x=weight))

P2  +
geom_histogram(color="black", fill="white")+   #plot histogram
facet_wrap(~group) # group by group
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



### Dealing with ouliers

2 extreme values figure in trt1 box-plot distribution. What will happen if we discard them from the analysis?

```
 P1 <- ggplot(Mydata1, aes(y=weight,color=group)) +
geom_boxplot()

summary(Mydata1[Mydata1$group=="trt1",])
```

```
##      weight        group   
##  Min.   :3.590   ctrl: 0  
##  1st Qu.:4.207   trt1:10  
##  Median :4.550   trt2: 0  
##  Mean   :4.661            
##  3rd Qu.:4.870            
##  Max.   :6.030            
```

```
 Mydata1filtered<- Mydata1[ Mydata1$group == "trt1" &  Mydata1$weight>5.5, ]

 #install.packages("dplyr")
 library(dplyr)
```

```
## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
 Mydata1 <- Mydata1 %>% anti_join(Mydata1filtered)
```

```
## Joining, by = c("weight", "group")
```

```
P1 <- ggplot(Mydata1, aes(y=weight,color=group)) +
geom_boxplot()

summary(Mydata1[Mydata1$group=="trt1",])
```

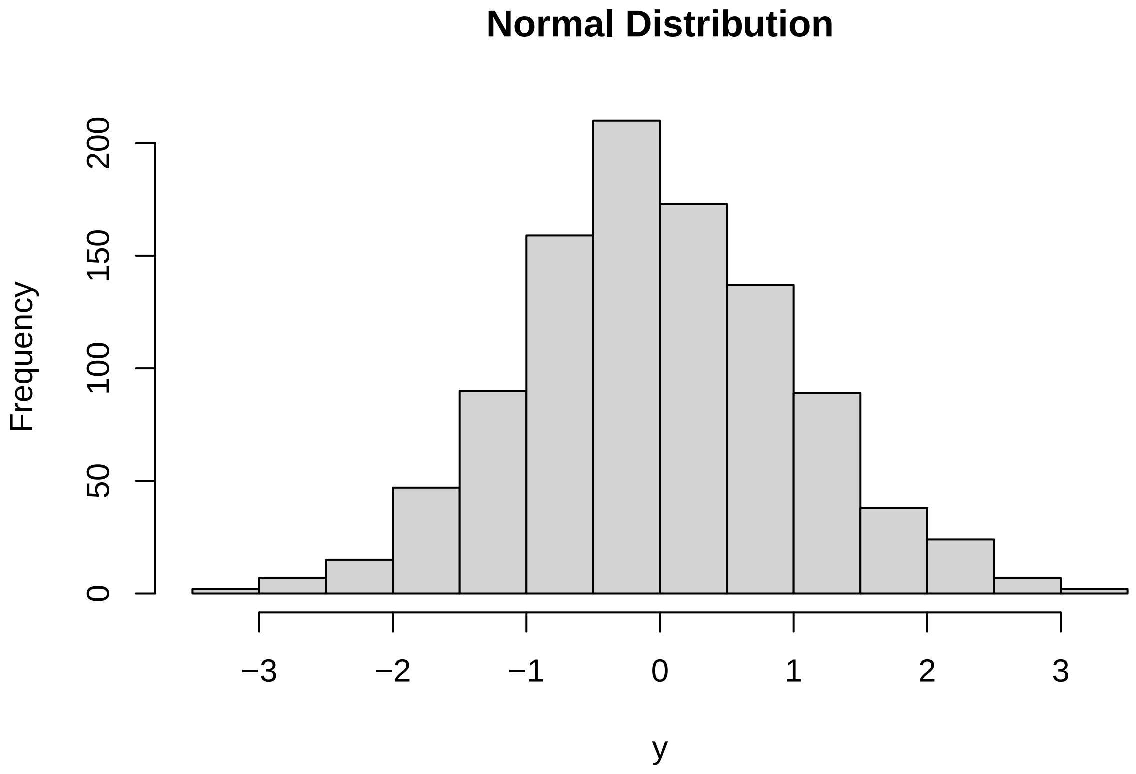
```
##      weight       group
##  Min.   :3.590   ctrl:0
##  1st Qu.:4.085   trt1:8
##  Median :4.365   trt2:0
##  Mean   :4.339
##  3rd Qu.:4.720
##  Max.   :4.890
```

Normal distribution

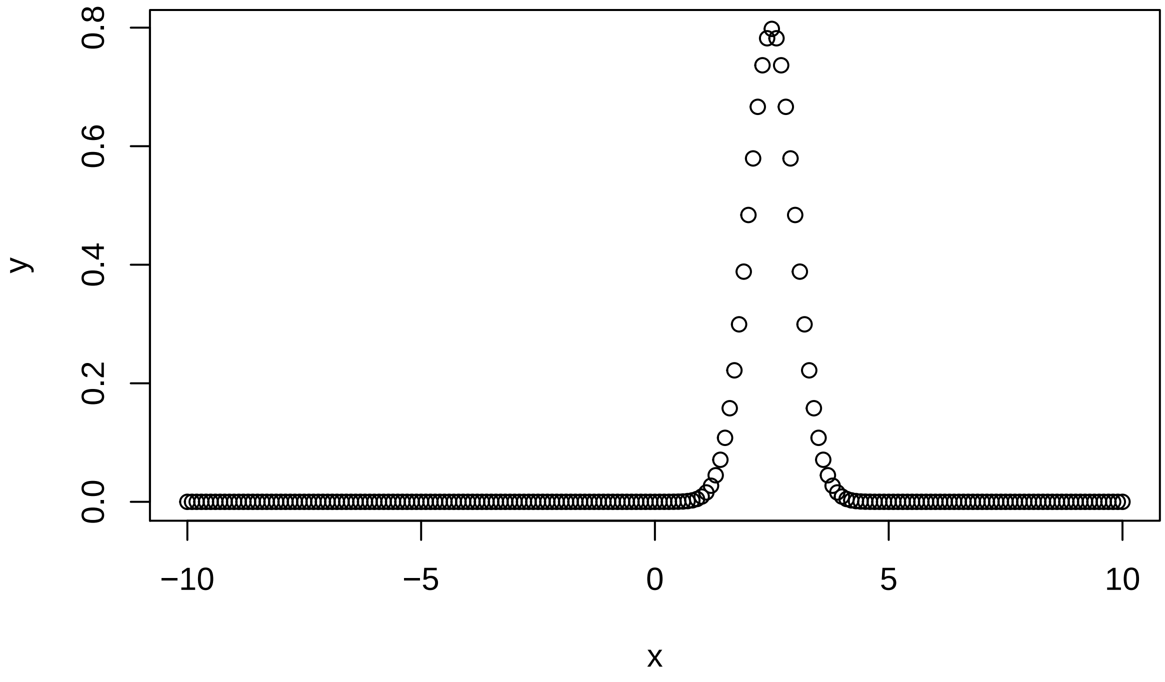
Biological variables are likely to be clustered around a central value (=mean).

The standard normal distribution curve is a curve that is shaped like a bell where the data is symmetrically distributed around the central value.

```
## rnorm()
# Generate random numbers whose distribution is normal. It takes the sample size as input and generates

y <- rnorm(1000)   # Create a sample of 1000 numbers which are normally distributed.
hist(y, main = "Normal Distribution") # Plot the histogram for this sample.
```



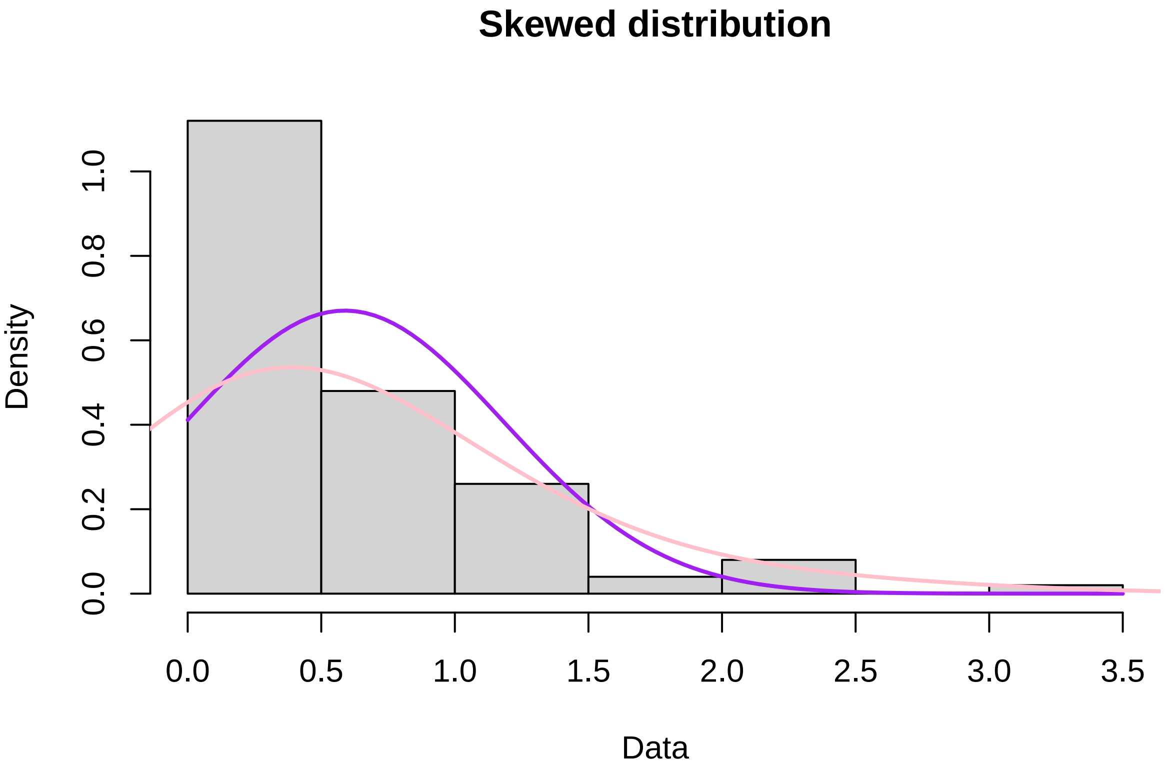
```
##dnorm()
#This function gives height of the probability distribution at each point for a given mean and standard

x <- seq(-10, 10, by = .1) # Create a sequence of numbers between -10 and 10 incrementing by 0.1.

y <- dnorm(x, mean = 2.5, sd = 0.5) # Choose the mean as 2.5 and standard deviation as 0.5.

plot(x,y) # The bell-shape plot
```



Skewed distribution

It is a distribution with an asymmetry of the observations about its mean. We count negatively skewed distribution (longer left tail) and positively skewed distribution (longer right tail).

```r
#Create a nonnormal distribution
n=100                   #number of data point
right <-rexp(n,rate=2)    #exponential distribution (gamma 2

m <-mean(right )    # compute mean
 std<-sd(right )          # and std

title<-("Skewed distribution")

hist(right, xlab="Data", freq= FALSE, main=title)
curve(dnorm(x, mean=m, sd=std), col="purple", lwd=2, add=TRUE) # dnorm for the normal distribuion
lines(density(right,adjust=3),col = "pink", lwd=2)    #  the density curve to show the skewness compare
```



2-Inferential Statistics

Inferential statistics are useful when it is not possible to examine every individual in a population. Inferential statistics make use a random sample to describe and make inferences about the entire population.

The objective is to test the hypotheses where a hypothesis is a suggested explanation for a given phenomenon. This test is performed by measuring the likelihood of the hypothesis to happen.

Null hypothesis (H0) stipulates that there is **no difference** between the studied population variables.

Alternative hypothesis (H1) stipulates that there is **a difference** between the studied population variables.

When performing test-hypothesis we make use of the P-value computed probability). The P value is the probability of the event occurring by chance if the null hypothesis is true. It has values in [0,1] and in function of a significant level chosen by the researcher, the null hypothesis is either retained or rejected.

If the P-value is low as the significant level (usually set to 0.05) => reject the null hypothesis.

Hypothesis testing workflow:

1- State the Hypotheses: null and alternative hypothesis

2- Formulate the analysis: what is the appropriate test to use. what is the significante level to decide.

3- Analyse data: calculate the test statistic

4- Interepret results

PARAMETRIC TESTS

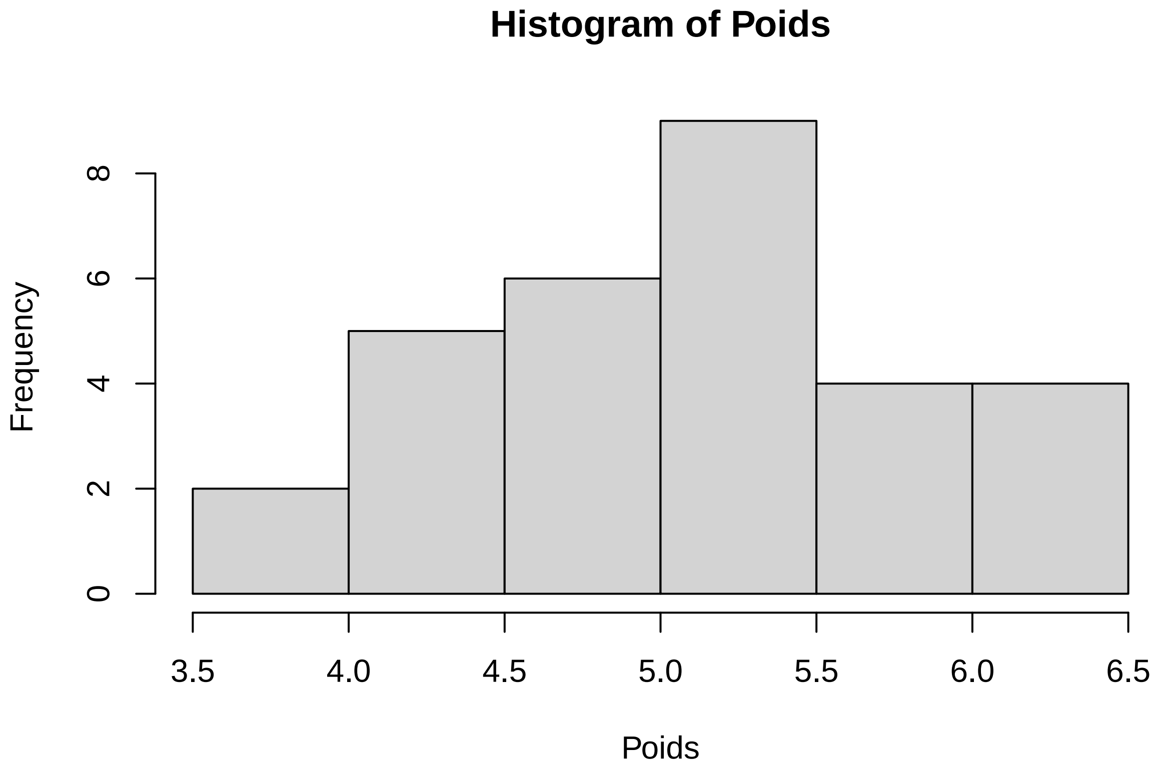
Data that is **normally** distributed is analyzed with **parametric tests**. Two assumptions should be verified:

- The assumption of **normality**: the means of the samples and of the population are normally distributed.
- The assumption of **equal variance** which specifies that the variances of the samples and of their corresponding population are equal.

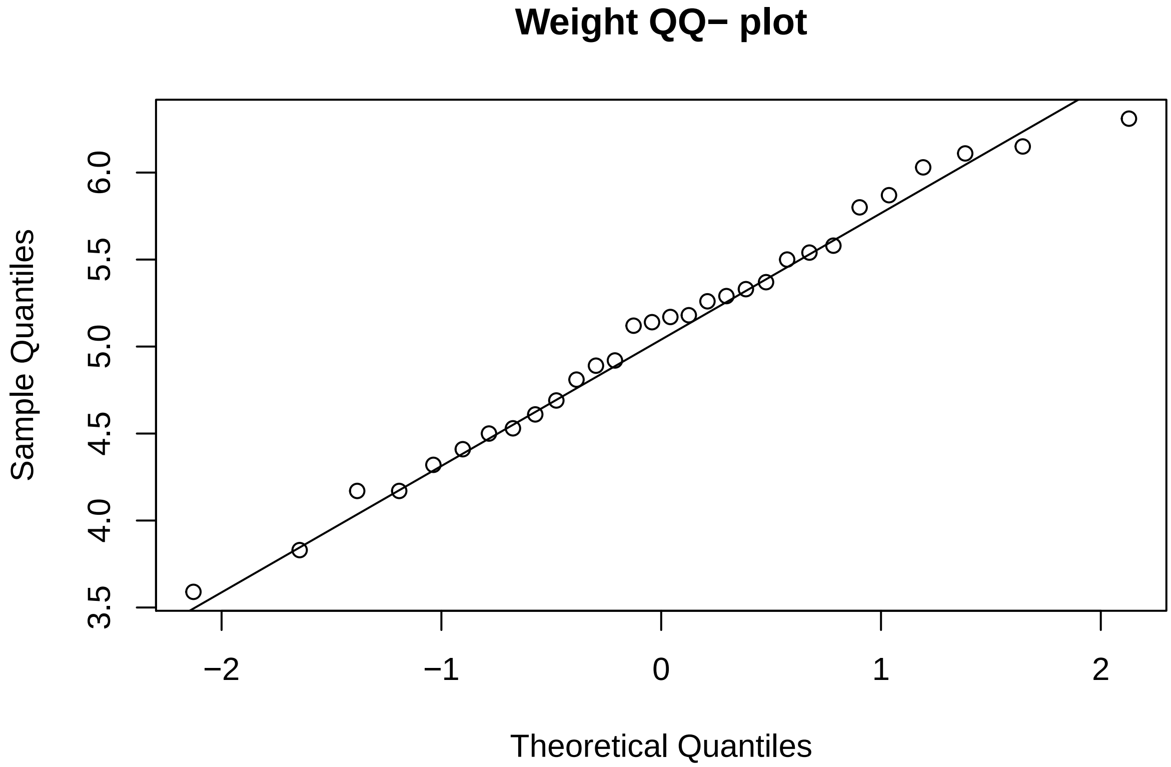
How to test for normality?

The most used tool to check for data normality is the quantile-quantile (QQ) plot. We want to compare a sample with a theoretical sample that comes from a normal distribution.

```
#qqnorm() : function plots your sample against a normal distribution.
#qqline : line that allows you to ocularly assess whether you see a clear deviation from normality.

 #x <- rnorm(100)
 #x <- rgamma(100, 1)
 #hist(x)
 #qqnorm( x, main='QQ- plot')
 #qqline( x)

 hist(Poids)
```



```
qqnorm( Poids, main='Weight QQ- plot')
qqline( Poids )
```



#### Student's t-test:

Used to test the null hypothesis that there is **no difference between the means** of **2 groups**.

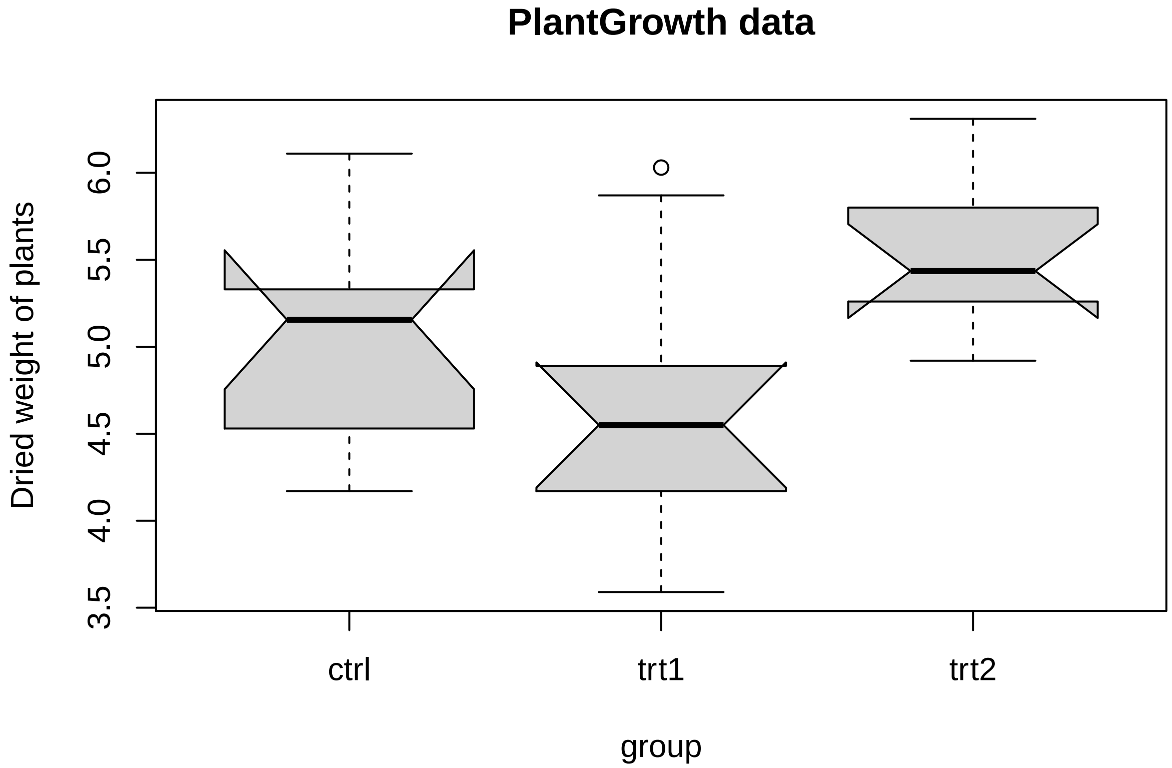
- *one-sample t-test*: useful when comparing two populations X and Y. It tests if the mean of a sample from population X differs significantly from the given mean of population Y.
- *unpaired t-test*: two independent samples from one population. It tests if the population means estimated by two independent samples differ significantly.
- *paired t-test*: two dependent samples from one population. It tests if the population means estimated by two dependent samples differ significantly. (an example of dependent samples could be measurements made on the same subjects before and after a treatment)

Analysis of variance (ANOVA): Used to test the null hypothesis that there is **no difference between the means** of **2 or more groups**.

In ANOVA, two variances are studied: – between-group variability - within-group variability Then, the two estimates of variances are compared using the F-test. F-statistics is computed as the ratio of the mean squares between the groups and the mean squares within groups.

```
## One factor ANOVA example from Dobson's book, cf. Table 7.4:
require(stats); require(graphics)
boxplot(weight ~ group, data = PlantGrowth, main = "PlantGrowth data",
        ylab = "Dried weight of plants", col = "lightgray",
        notch = TRUE, varwidth = TRUE)
```

```
## Warning in bxp(list(stats = structure(c(4.17, 4.53, 5.155, 5.33, 6.11, 3.59, :
## some notches went outside hinges ('box'): maybe set notch=FALSE
```



ANOVA model could be used to help us answer the question whether any group weight means differ from another. ANOVA does not specify which groups differ.

```
anova(lm(weight ~ group, data = PlantGrowth)) #ANOVA for Linear Model Fits
```

```
## Analysis of Variance Table
##
```

```
## Response: weight
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## group      2  3.7663  1.8832  4.8461 0.01591 *
## Residuals 27 10.4921  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Need to add interpretation and to add examples for ANova!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!! interval confidence: Tukey's Honest Significant Differences (HSD). https://bookdown.org/steve_midway/DAR/understanding-anova-in-r.html!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Decription for Pearson correlations!!

```
install.packages("ggpubr")
```

Pearson correlation coefficient

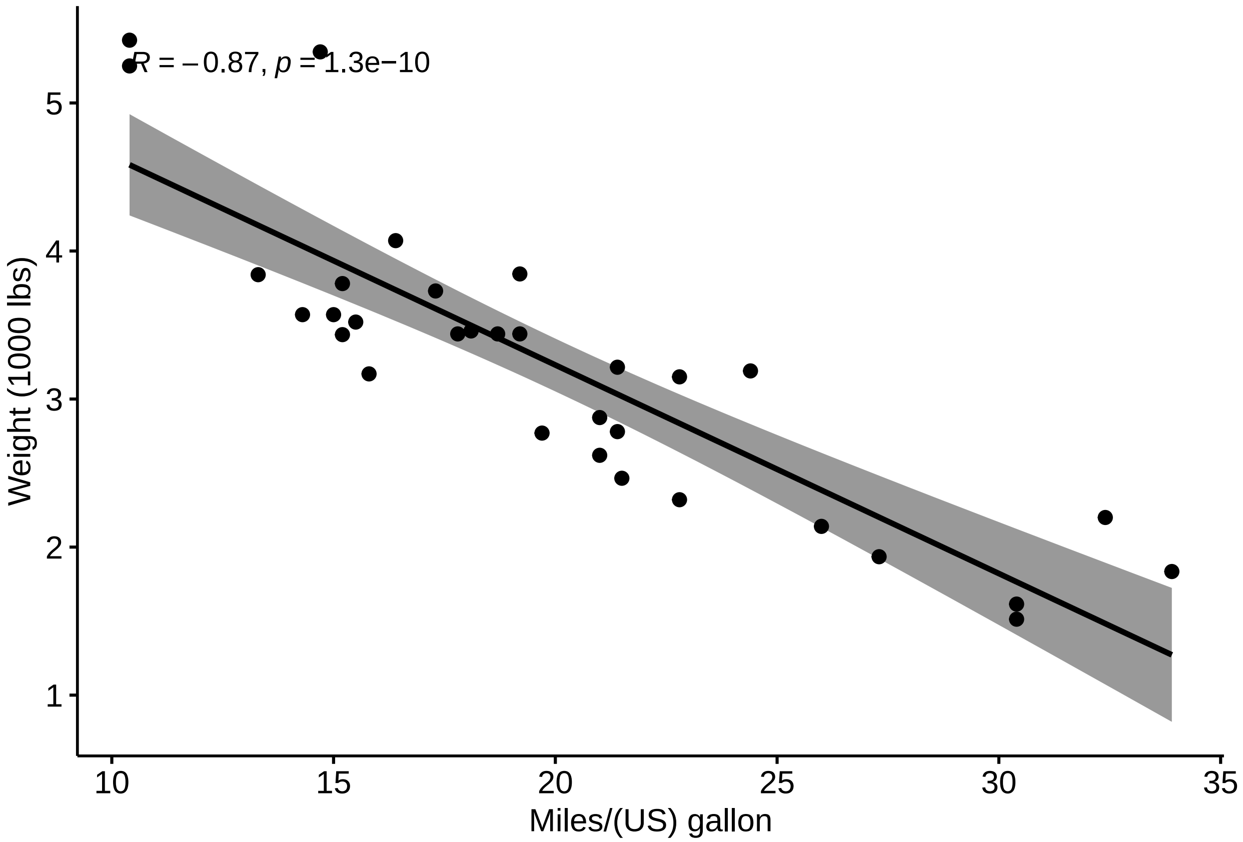
```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library("ggpubr")
my_data <- mtcars
head(my_data, 6)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
ggscatter(my_data, x = "mpg", y = "wt",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Miles/(US) gallon", ylab = "Weight (1000 lbs)")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
res <- cor.test(my_data$wt, my_data$mpg,
                    method = "pearson")
res
```

```
##
##  Pearson's product-moment correlation
##
## data:  my_data$wt and my_data$mpg
## t = -9.559, df = 30, p-value = 1.294e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9338264 -0.7440872
## sample estimates:
##        cor
## -0.8676594
```

```
#    t is the t-test statistic value (t = -9.559),
 #   df is the degrees of freedom (df= 30),
 #   p-value is the significance level of the t-test (p-value = 1.29410^{-10}).
 #   conf.int is the confidence interval of the correlation coefficient at 95% (conf.int = [-0.9338, -0
#    sample estimates is the correlation coefficient (Cor.coeff = -0.87).
```

Non-PARAMETRIC TESTS

In case the distribution of the sample is skewed or **unknown**, **Non-parametric tests** are used.

####Kolmogorov-Smirnov test

The two-sample Kolmogorov-Smirnov (KS) test is used to test whether **two random samples*** are drawn from the **same distribution**.

The null hypothesis is that both distributions are identical. The statistic of the KS test is a distance between the two empirical distributions (computed as the maximum absolute difference between their cumulative curves).

Need to add example for ks test !!!

Bootstraping techniques:

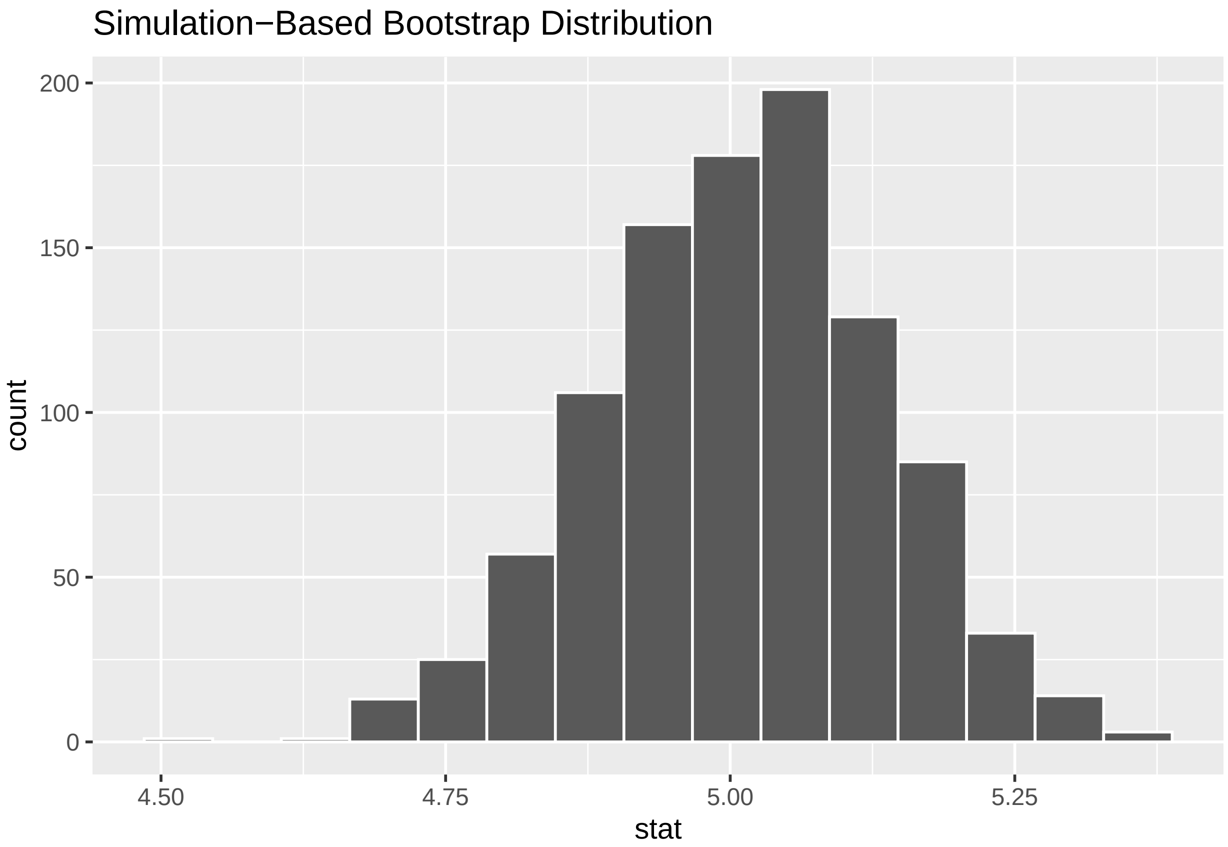
Useful to estimate the mean and the median for the population using statistics from the sample. Assuming that data is representative The Bootstrap population obsr. from the sample appear many times

1- build a bootstrap sample that is a random sample with replacement from and with the same size as the sample. 2- compute the bootstrap statistic (mean, median…) 3- Reapeat 1 and 2 to create the Bootstrap distribution

```
#https://cran.r-project.org/web/packages/infer/infer.pdf
#https://moderndive.github.io/moderndive_labs/static/previous_versions/v0.5.0/9-confidence-intervals.ht

#install.packages("infer")
library(infer)

bootstrap_distribution<-Mydata1 %>%
specify(response = weight)  %>%     #weight as variable of interest
generate(reps = 1000, type = "bootstrap") %>%   # generate bootstrap sample. reps: the number of resamp
calculate(stat = "mean")        # Calculate mean of each bootstrap sample

bootstrap_distribution
```

```
## # A tibble: 1,000 x 2
##    replicate  stat
##        <int> <dbl>
##  1         1  4.98
##  2         2  4.87
##  3         3  4.99
##  4         4  5.27
##  5         5  4.74
##  6         6  4.96
##  7         7  4.92
##  8         8  5.16
##  9         9  4.82
## 10        10  5.06
## # ... with 990 more rows
```

```
bootstrap_distribution %>%
  visualize()
```



# References

Ali, Zulfiqar, and S Bala Bhaskar. 2016. "Basic Statistical Tools in Research and Data Analysis." *Indian Journal of Anaesthesia* 60 (9): 662.