Statistics_R

Ulises Jose Bustamante Mora

2023-10-22

Math

```
x = 5
print(tan(x))
```

```
## [1] -3.380515
```

```
print(cos(x))
```

```
## [1] 0.2836622
```

```
print(sin(x))
```

```
## [1] -0.9589243
```

```
print(sqrt(x))
```

```
## [1] 2.236068
```

```
print(log(x))
```

```
## [1] 1.609438
```

```
print(exp(x))
```

```
## [1] 148.4132
```

```
print(factorial(x))
```

```
## [1] 120
```

```
y1 = 5.4
y2 = 5.5

print(floor(y1))
```

```
## [1] 5
```

```r
print(floor(y2))
```

```
## [1] 5
```

```r
print(ceiling(y1))
```

```
## [1] 6
```

```r
print(ceiling(y2))
```

```
## [1] 6
```

```r
print(trunc(y1))
```

```
## [1] 5
```

```r
print(trunc(y2))
```

```
## [1] 5
```

```r
print(round(y1),0)
```

```
## [1] 5
```

```r
print(round(y2),0)
```

```
## [1] 6
```

Measures of Centrality and Variability

```r
library(DescTools)
salaries = c(1000,18000,2550,3365,8874,2589,5248,2550)

print(paste0("The mean of salaries is: ", mean(salaries)))
```

```
## [1] "The mean of salaries is: 5522"
```

```r
print(paste0("The median of salaries is: ", median(salaries)))
```

```
## [1] "The median of salaries is: 2977"
```

```r
print(paste0("The mode of salaries is: ", Mode(salaries)))
```

```
## [1] "The mode of salaries is: 2550"
```

```r
print(paste0("The quatiles of salaries is: ", quantile(salaries)))
```

```
## [1] "The quatiles of salaries is: 1000"   "The quatiles of salaries is: 2550"
## [3] "The quatiles of salaries is: 2977"   "The quatiles of salaries is: 6154.5"
## [5] "The quatiles of salaries is: 18000"
```

```r
print(paste0("The standard deviation of salaries are: ", sd(salaries)))
```

```
## [1] "The standard deviation of salaries are: 5585.44861991535"
```

```r
summary(salaries)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    2550    2977    5522    6154   18000
```

Simple Random Sampling

Create the sample for the Iris dataset

```r
sampleForIris = sample(c(0,1), 150, replace = TRUE, prob = c(0.7,0.3))
summary(as.factor(sampleForIris))
```

```
##   0   1
## 103  47
```

Now we are going to separate the rows that has the value of 1 and let the columns without modification

```r
sampleIris = iris[sampleForIris==1,]
head(sampleIris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
## 7          4.6         3.4          1.4         0.3  setosa
## 9          4.4         2.9          1.4         0.2  setosa
```

We compared both outputs

```r
summary(as.factor(sampleForIris))
```

```
##   0   1
## 103  47
```

```r
dim(sampleIris)
```

```
## [1] 47  5
```

Stratified Sampling

```r
library("sampling")

#We create a variable with the number of examples that we want for each example.
portion = 25

#From the iris dataset, we separate by Species, and extract 25 examples
#for the three different category
# the method is srxwor. so it is a simple sample without replacement

sampleIris = strata(data = iris, stratanames = c("Species"),
                    size=c(rep(portion,3)), method = "srswor")
summary(sampleIris)
```

```
##        Species      ID_unit            Prob        Stratum
##  setosa    :25   Min.   :  1.00   Min.   :0.5   Min.   :1
##  versicolor:25   1st Qu.: 41.00   1st Qu.:0.5   1st Qu.:1
##  virginica :25   Median : 74.00   Median :0.5   Median :2
##                  Mean   : 75.59   Mean   :0.5   Mean   :2
##                  3rd Qu.:110.50   3rd Qu.:0.5   3rd Qu.:3
##                  Max.   :150.00   Max.   :0.5   Max.   :3
```

```r
#Same example but with different recollection metrics

sampleInfer = strata(data=infert, stratanames = c("education"), size= c(5,48,47), method="srswor")
summary(sampleInfer)
```

```
##    education     ID_unit            Prob           Stratum
##  0-5yrs : 5   Min.   :  1.00   Min.   :0.4000   Min.   :1.00
##  6-11yrs:48   1st Qu.: 51.75   1st Qu.:0.4000   1st Qu.:2.00
##  12+ yrs:47   Median :120.00   Median :0.4052   Median :2.00
##               Mean   :119.38   Mean   :0.4033   Mean   :2.42
##               3rd Qu.:181.50   3rd Qu.:0.4052   3rd Qu.:3.00
##               Max.   :248.00   Max.   :0.4167   Max.   :3.00
```

Systematic Sampling

```
## Loading required package: dplyr
```

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

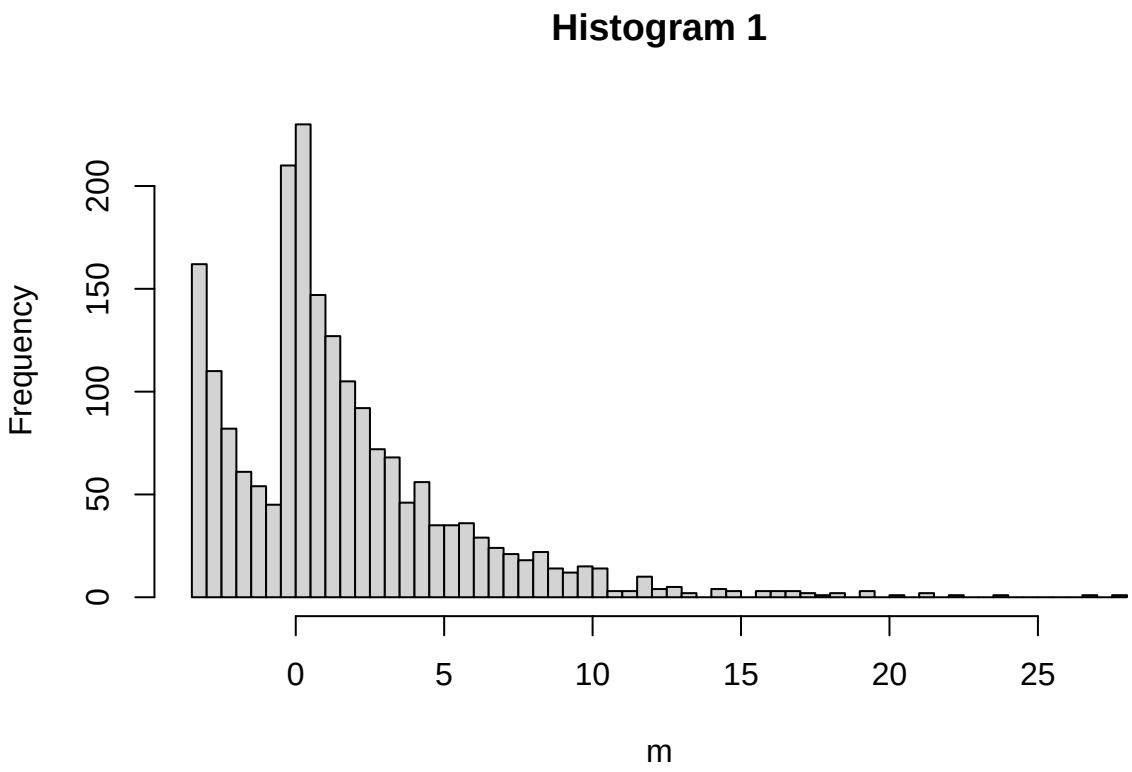
```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: magrittr

## [1] 15  1

##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 4           4.6         3.1          1.5         0.2     setosa
## 14          4.3         3.0          1.1         0.1     setosa
## 24          5.1         3.3          1.7         0.5     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 54          5.5         2.3          4.0         1.3 versicolor
```

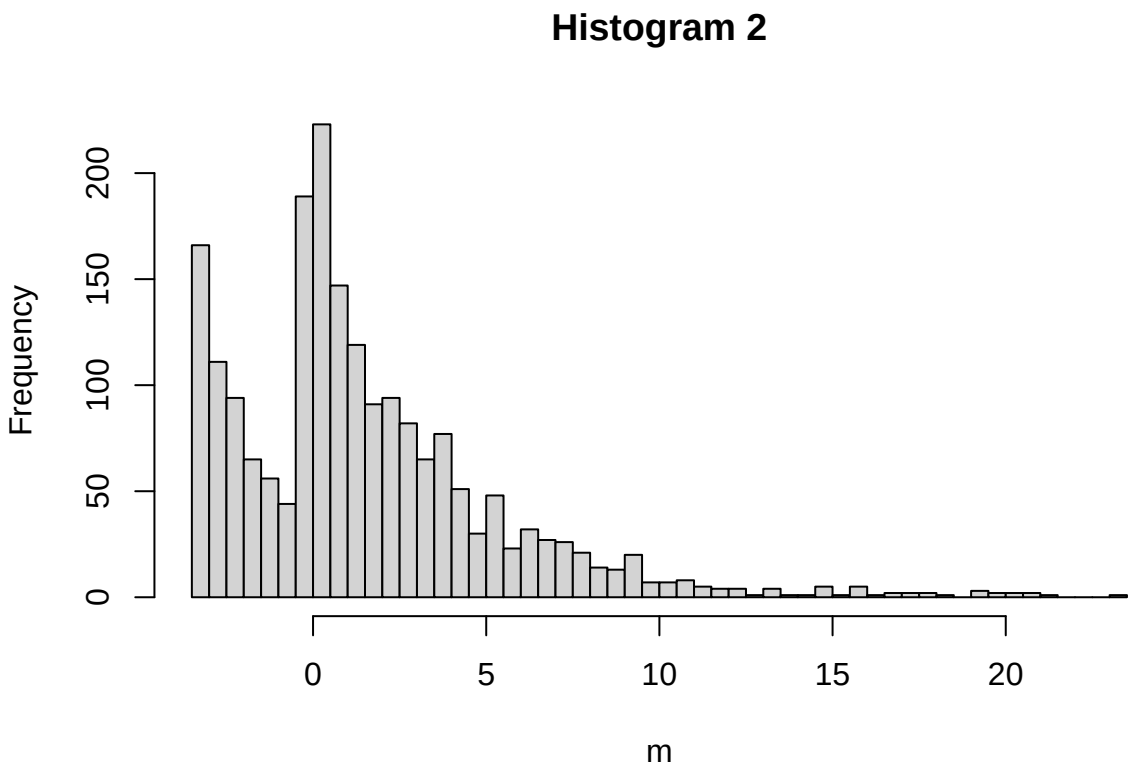
Central Limit Theorem

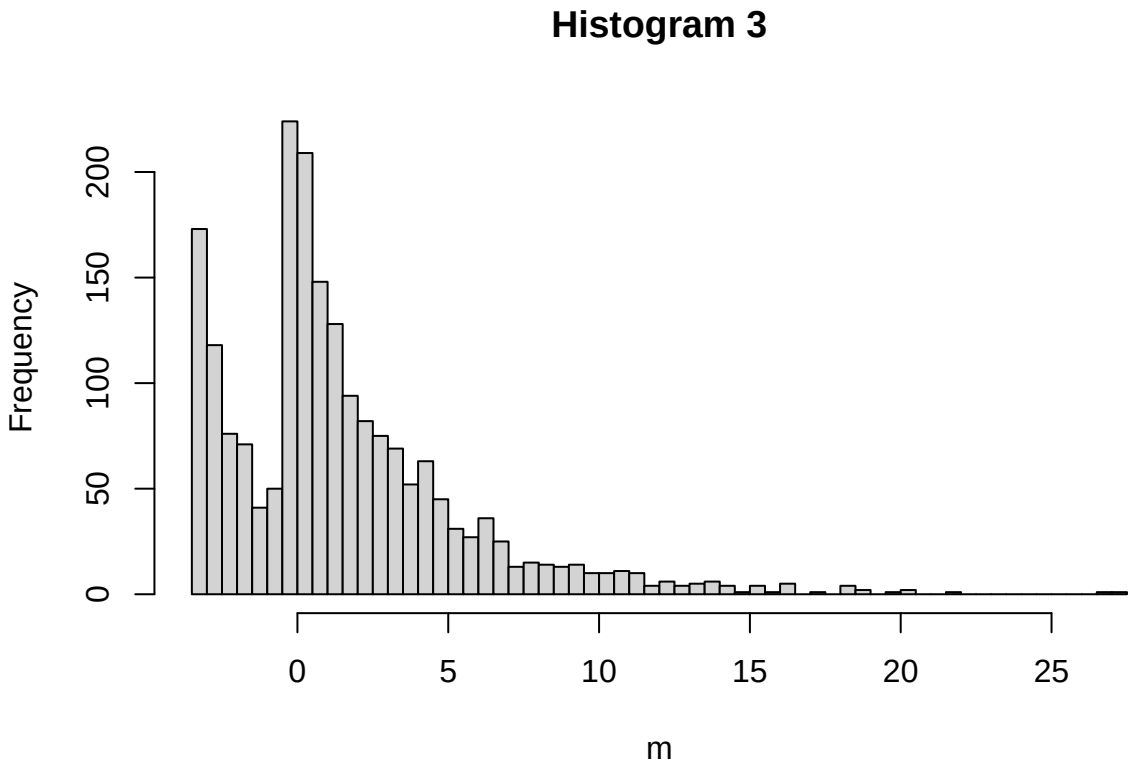
```
library(semTools)
```

```
## Loading required package: lavaan

## This is lavaan 0.6-16
## lavaan is FREE software! Please report any bugs.

##

## ###############################################################################

## This is semTools 0.5-6

## All users of R (or SEM) are invited to submit functions or ideas for functions.

## ###############################################################################
```

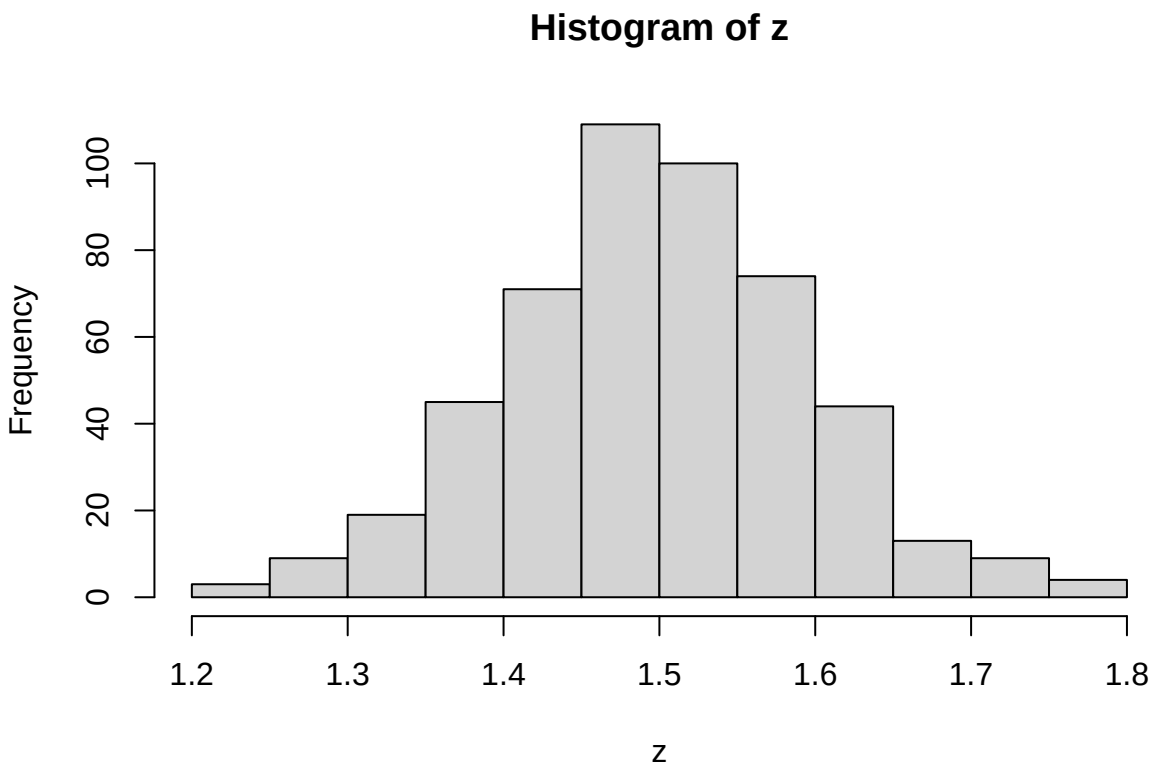
```
#Create a list of 0s 500 times
z = rep(0,500)

#On that list, we are going to add non normalized data of 1000 values
for (i in 1:500){
  m = mvrnonnorm(1000, c(1,2), matrix(c(10,2,2,5), 2,2), skewness = c(5,2), kurtosis = c(3,3))
  z[i] = mean(m)

#We are going to grab the first tree non normalized histograms
  if (i<4){
    hist(m, breaks = 50, main = paste0("Histogram ", i))
  }
}
```


Histogram 1
Frequency
0
50
100
150
200
0
5
10
15
20
25
m


Histogram 2
Frequency
0
50
100
150
200
0
5
10
15
20
m

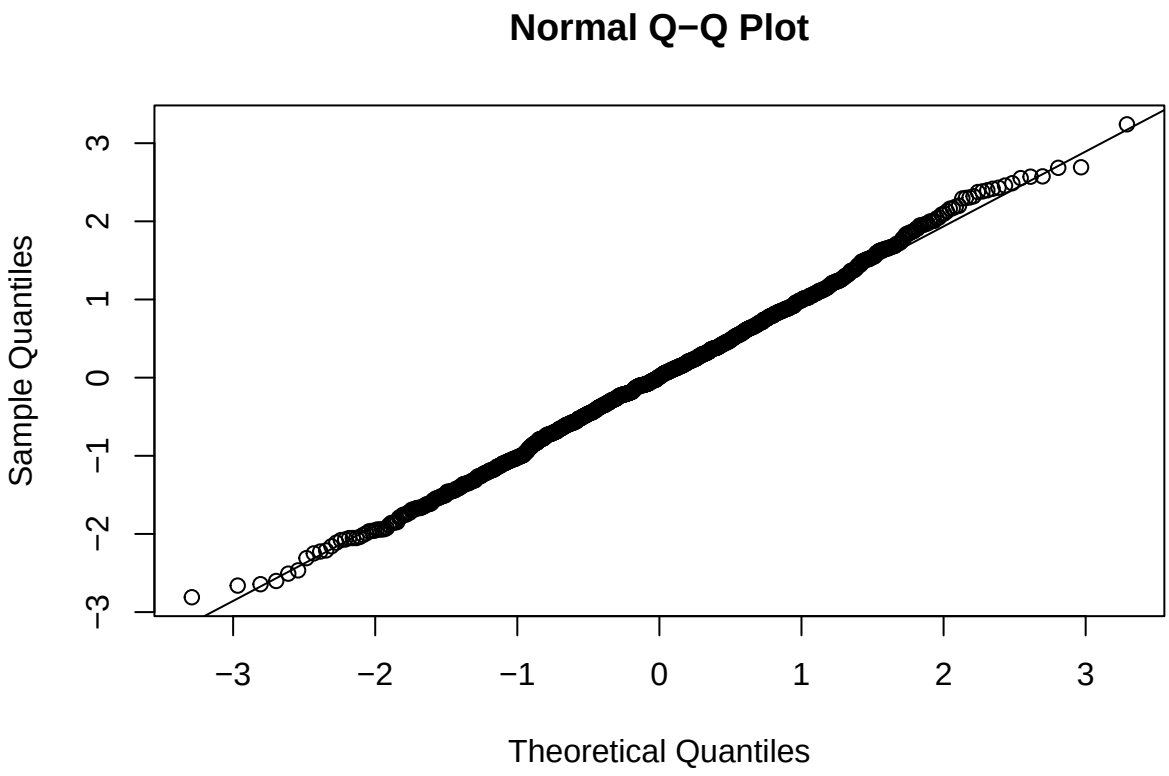



```
#Now with the mean of the dataset with non normalized data, se can see that becomes normalized data.
hist(z)
```



Normaly Test

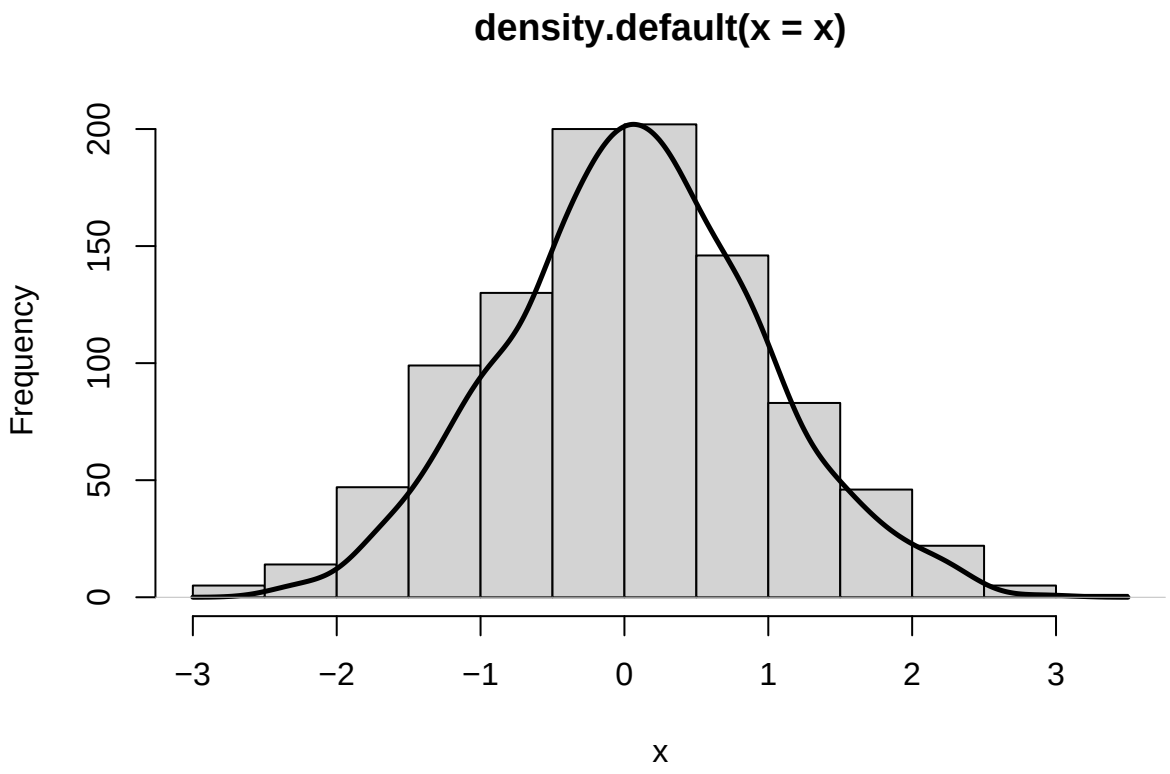
```
#TEST 1
set.seed(123)

x = rnorm(1000)

qqnorm(x)

qqline(x)
```



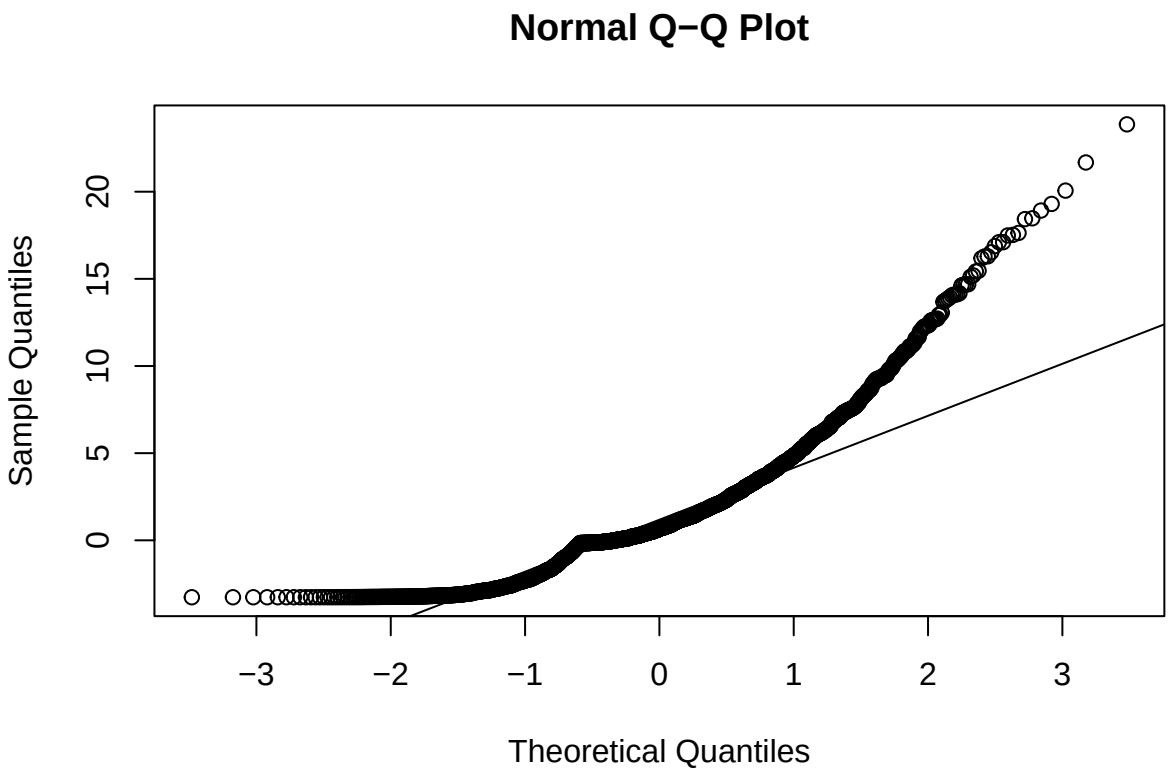

```
#TEST 2
#Shapiro Test
#We notice the the p-value is over 0.05, so our null is correct.
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.99838, p-value = 0.4765
```

```
#TEST 3
#Hist with a density line
hist(x, main="")
par(new = TRUE)
plot(density(x), ylab = "", xlab = "", axes = F, lwd=2.5)
```



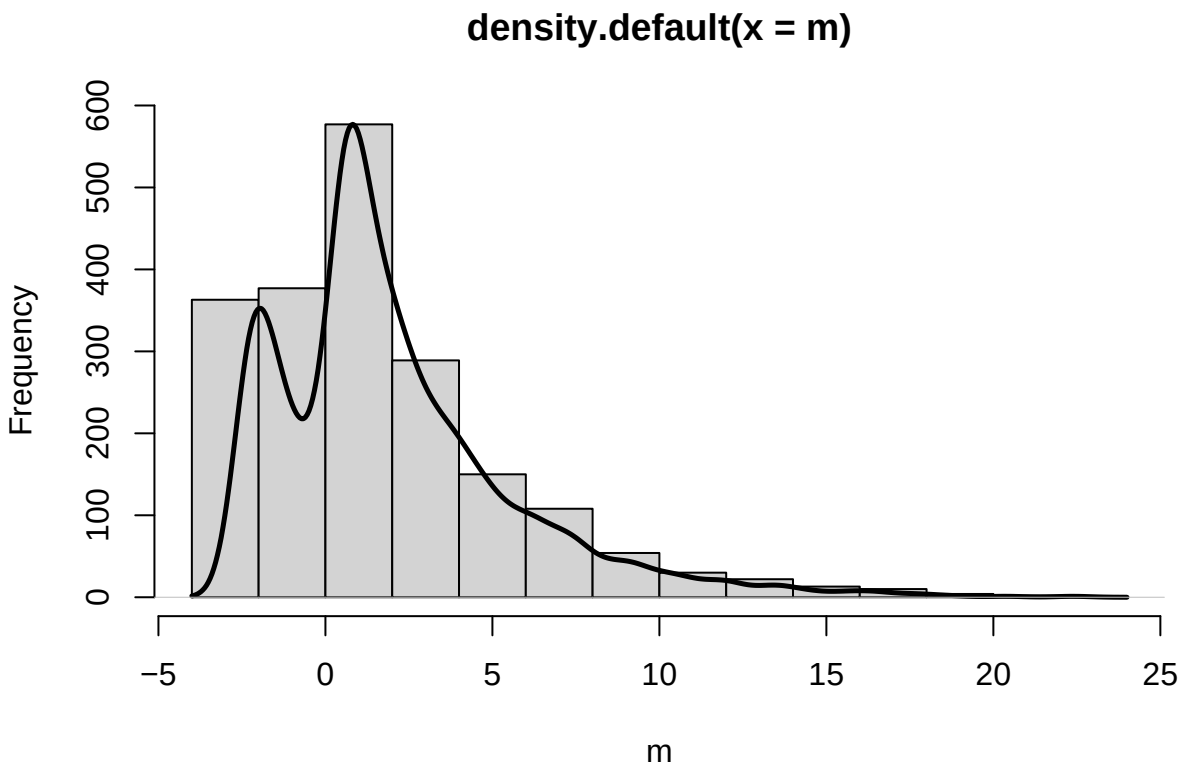

```
#DATA WITHOUT NORMALITY DISTRIBUTION
library(semTools)
m = mvrnonnorm(1000, c(1,2), matrix(c(10,2,2,5), 2,2), skewness = c(5,2), kurtosis = c(3,3))
qqnorm(m)
qqline(m)
```




```
#P value less than 0.05

shapiro.test(m)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m
## W = 0.8857, p-value < 2.2e-16
```

```
hist(m, main="")
par(new = TRUE)
plot(density(m), ylab = "", xlab = "", axes = F, lwd=2.5)
```



Long Term Averages

```
x1 = sample(1:6,6,replace = T)
x2 = sample(1:6, 10000, replace = T)


mean(x1)
```

```
## [1] 4
```

```
mean(x2)
```

```
## [1] 3.4755
```

Confidende Intervals

```
salaries = c(3000,4567,3420,3450,4530,3450,2340,6540,6543,7000)

#CONFIDENT INTERVALS WITH 95%
```

```r
sample = 10
z = 1.96 #95%
average = mean(salaries)
standardDev = sd(salaries)

marginOfError = z*(standardDev/sqrt(sample))

print(paste("The salaries are between:", round(average - marginOfError, 2), "and", round(average + marg
```

```
## [1] "The salaries are between: 3453.57 and 5514.43 .With a confidence of 95%"
```

```r
#CONFIDENT INTERVALS WITH 99%
sample = 10
z = 2.58 #99%
average = mean(salaries)
standardDev = sd(salaries)

marginOfError = z*(standardDev/sqrt(sample))

print(paste("The salaries are between:", round(average - marginOfError, 2), "and", round(average + marg
```

```
## [1] "The salaries are between: 3127.62 and 5840.38 .With a confidence of 99%"
```

```r
#THE SAME PROCESS BUT NOW KNOWING THE VALUE
sample = 1000
z = 1.96
value = 0.7 # 700 people out of 1000 would rather this value

marginOfError = z*sqrt(value*(1-value)/sample)
print(paste("People would choose this value between:", round(value - marginOfError, 2), "and", round(va
```

```
## [1] "People would choose this value between: 0.67 and 0.73 .With a confidence of 95%"
```

T Student

```r
#The average salary for a Data Scientist is $20 per hour. Sample = 9 people
#and Standard Deviation = 10.


#Getting t:
print(paste0("T value is ", (30-20)/10/sqrt(9)))
```

```
## [1] "T value is 0.333333333333333"
```

```r
t = (30-20)/10/sqrt(9)

#eight because degree of freedom is n-1
print(paste0("The probability of getting a salary below than $30 is: " , pt(t, 8)))
```

```
## [1] "The probability of getting a salary below than $30 is: 0.626274507915265"
```

```
#Probability of being higher than $30

print(paste0("The probability of getting a salary higher than $30 is: ", pt(t,8, lower.tail = F)))
```

```
## [1] "The probability of getting a salary higher than $30 is: 0.373725492084735"
```

```
#Another way to do it
1-pt(t,9)
```

```
## [1] 0.3732585
```

```
total = pt(t, 8) + pt(t,8, lower.tail = F)
total
```

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

Hypothesis Test

```
#Hypothesis: An institute claims that on average, 75% of the people vote for Mike.

# Null Hypothesis = 0.75
# Alternative Hypothesis < 0.75

#DATA

n = 100
p = 0.77

Z = (p-0.75)/sqrt(0.75*(1-0.75)/n)
Z
```

```
## [1] 0.4618802
```

```
#The Z value is 0.461. So we need to look on the Z table the P-value.
#In this case is 0.6772. So as this number is higher than 0.05. So the null hypothesis is true

#SECOND TEST

#On Average, six years old children weight 22kg

#Null hypothesis = 22kg
#Alternative hypothesis > 22kg

n = 100
sixYOChildrenMean = 23
standardDeviation = 4

Z = (sixYOChildrenMean-22)/(standardDeviation/sqrt(n))
Z
```

```
## [1] 2.5
```

```
#The Z value is 0.25. So we need to look on the Z table the P-value.
#In this case is 0.9938.
1-0.9938
```

```
## [1] 0.0062
```

```
#So as this number is lower than 0.05. So the null hypothesis is false
```

Chi Square

```
df = matrix(c(19,6,43,32), nrow=2, byrow=T)
rownames(df) = c("Male", "Female")
colnames(df) = c("Present", "Ausent")
df
```

```
##        Present Ausent
## Male        19      6
## Female      43     32
```

```
chisq.test(df) #On this test the p value is 0.15. So we can confirm that there is not a significant dif
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  df
## X-squared = 2.0374, df = 1, p-value = 0.1535
```

```
df = matrix(c(22,3,43,32), nrow=2, byrow=T)
rownames(df) = c("Male", "Female")
colnames(df) = c("Present", "Ausent")
df
```

```
##        Present Ausent
## Male        22      3
## Female      43     32
```

```
chisq.test(df) #On this test the p value is 0.01. So we can confirm that there is a significant differe
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  df
## X-squared = 6.4615, df = 1, p-value = 0.01102
```

Binomial Distribution

```r
# x = events, p = probability, size = number of events

#If you flip a coin five times, what is the probability of tails three times in a row?

dbinom(x = 3, size = 5, prob = 0.5)
```

```
## [1] 0.3125
```

```r
# What is the probability of passing four green lights in a row?

pbinom(q = 4, size= 4 , prob = 0.25) #It is 100% because you sum every result
```

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

```r
pbinom(q = 2, size= 4 , prob = 0.25) # It is 94% because between 3 and 4 times, the probability is too
```

```
## [1] 0.9492188
```

```r
# We are doing a test with 12 questions, by guessing, what is the probability of getting rights 7 quest

dbinom(x = 7, size = 12, prob = 0.25)
```

```
## [1] 0.01147127
```

```r
# What is the probability of passing three green lights or more

pbinom(q = 2, size = 4, prob = 0.25 , lower.tail = F)
```

```
## [1] 0.05078125
```

```r
#Another way to do it
dbinom(x = 3, size = 4, prob = 0.25) + dbinom(x = 4, size = 4, prob = 0.25)
```

```
## [1] 0.05078125
```

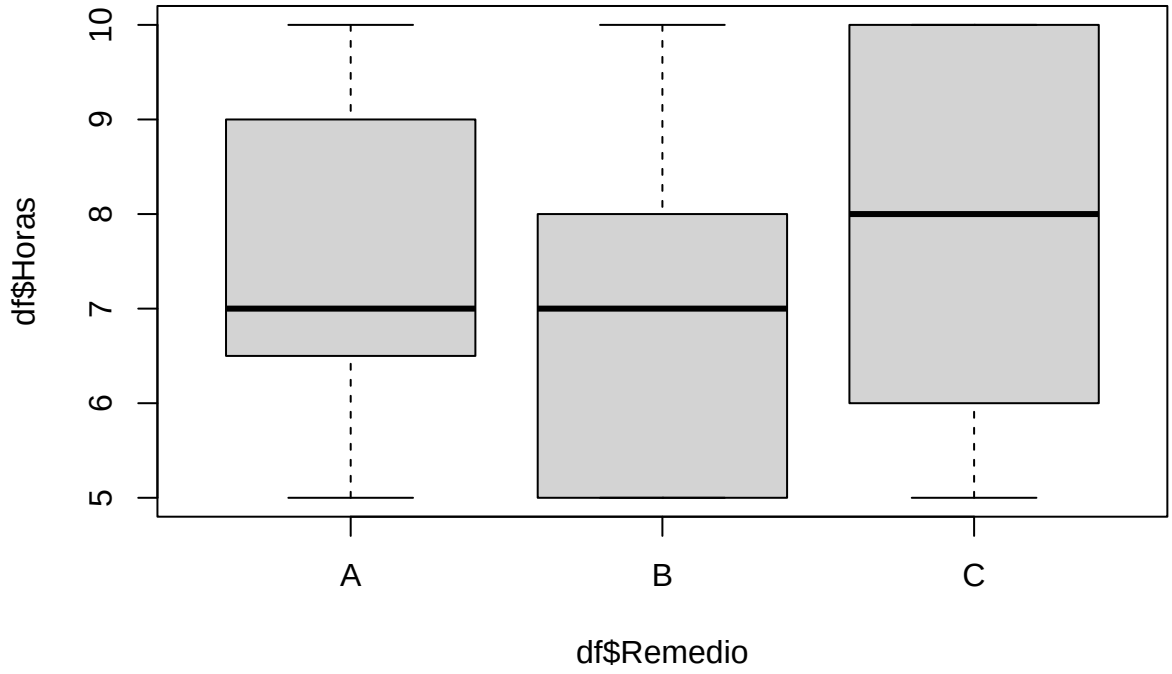
ANOVA

```r
#Importing the dataset
df = read.csv("anova.csv", sep = ";")
df
```

```
##    Sexo Remedio Horas
## 1     F       A     5
## 2     F       A    10
## 3     F       A     7
## 4     F       A     7
```

```
## 5     M       A     7
## 6     M       A     6
## 7     M       A     9
## 8     M       A     9
## 9     F       B     5
## 10    F       B     5
## 11    F       B     5
## 12    F       B     8
## 13    M       B     7
## 14    M       B     8
## 15    M       B    10
## 16    M       B     7
## 17    F       C    10
## 18    F       C    10
## 19    F       C     6
## 20    F       C     6
## 21    M       C    10
## 22    M       C     6
## 23    M       C    10
## 24    M       C     5
```

```
boxplot(df$Horas  ~ df$Remedio)
```




```
#The P value is higher than 0.5 so the null hypothesis is true (Between two variables does not affect w
an = aov(Horas ~ Remedio, data = df)
summary (an)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Remedio      2   4.08   2.042   0.538  0.592
## Residuals   21  79.75   3.798
```

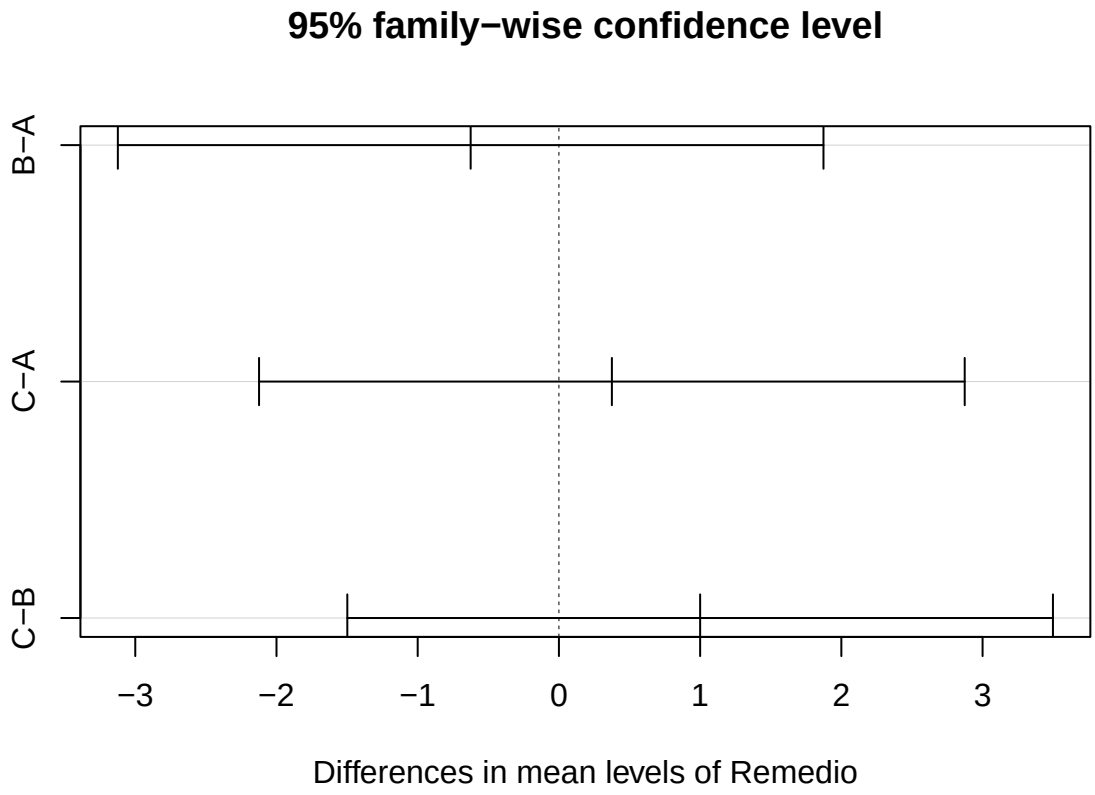
```
#The P value is higher than 0.5 so the null hypothesis is true (Between all variables does not affect w
an = aov(Horas ~ Remedio * Sexo, data = df)
summary(an)
```

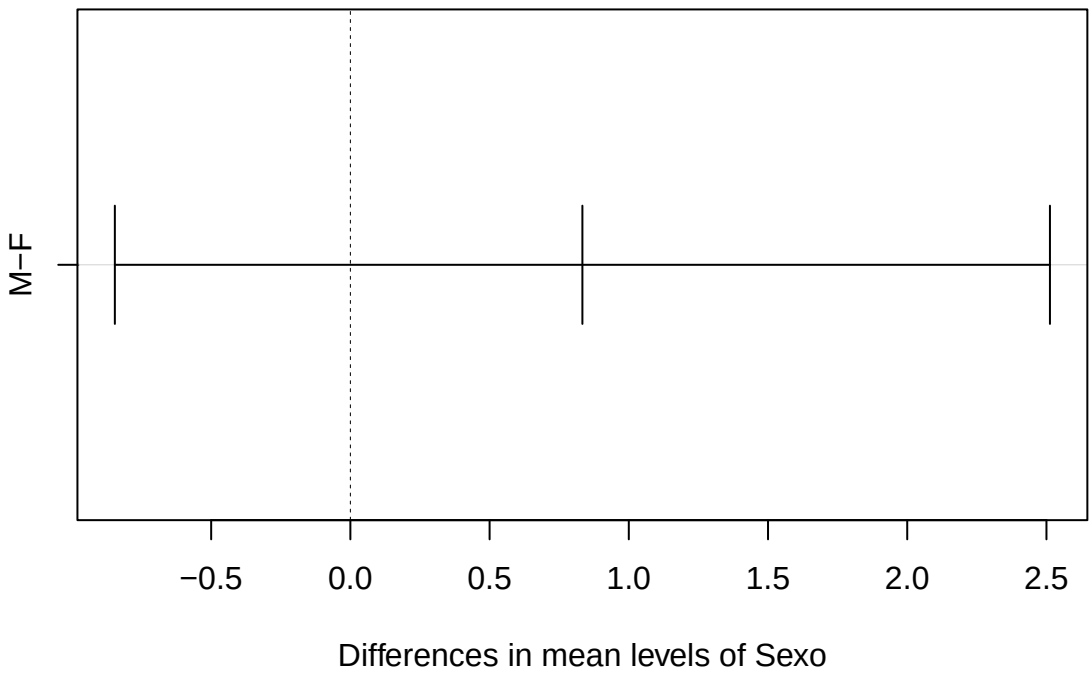
```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Remedio       2   4.08   2.042   0.533  0.596
## Sexo          1   4.17   4.167   1.087  0.311
## Remedio:Sexo  2   6.58   3.292   0.859  0.440
## Residuals    18  69.00   3.833
```

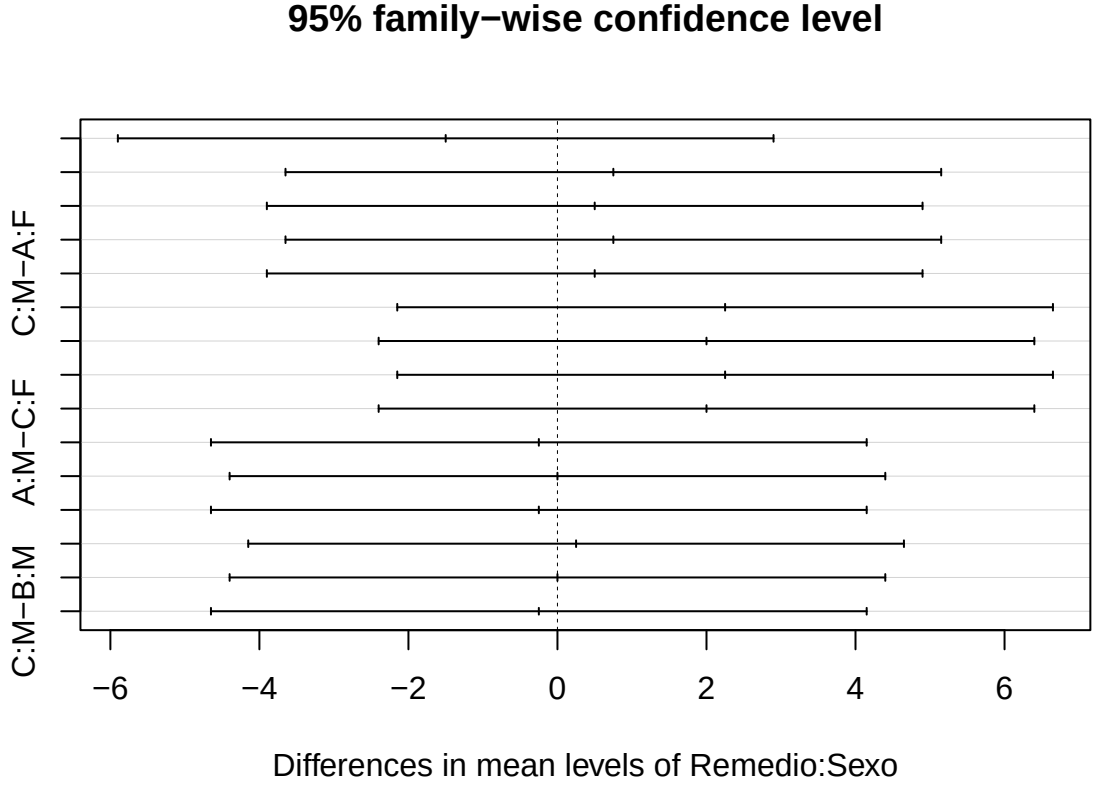
```
#The p adj value is higher than 0.5, so the null hypothesis is true
tukey = TukeyHSD(an)
tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Horas ~ Remedio * Sexo, data = df)
##
## $Remedio
##       diff       lwr      upr     p adj
## B-A -0.625 -3.123427 1.873427 0.8011051
## C-A  0.375 -2.123427 2.873427 0.9226431
## C-B  1.000 -1.498427 3.498427 0.5732500
##
## $Sexo
##          diff        lwr      upr     p adj
## M-F 0.8333333 -0.8459446 2.512611 0.3109477
##
## $`Remedio:Sexo`
##                   diff      lwr     upr     p adj
## B:F-A:F -1.500000e+00 -5.89979 2.89979 0.8816496
## C:F-A:F  7.500000e-01 -3.64979 5.14979 0.9935270
## A:M-A:F  5.000000e-01 -3.89979 4.89979 0.9990466
## B:M-A:F  7.500000e-01 -3.64979 5.14979 0.9935270
## C:M-A:F  5.000000e-01 -3.89979 4.89979 0.9990466
## C:F-B:F  2.250000e+00 -2.14979 6.64979 0.5936233
## A:M-B:F  2.000000e+00 -2.39979 6.39979 0.7010347
## B:M-B:F  2.250000e+00 -2.14979 6.64979 0.5936233
## C:M-B:F  2.000000e+00 -2.39979 6.39979 0.7010347
## A:M-C:F -2.500000e-01 -4.64979 4.14979 0.9999681
## B:M-C:F  8.881784e-16 -4.39979 4.39979 1.0000000
## C:M-C:F -2.500000e-01 -4.64979 4.14979 0.9999681
## B:M-A:M  2.500000e-01 -4.14979 4.64979 0.9999681
## C:M-A:M  0.000000e+00 -4.39979 4.39979 1.0000000
## C:M-B:M -2.500000e-01 -4.64979 4.14979 0.9999681
```

```
plot(tukey)
```


95% family−wise confidence level
B−A
C−A
C−B
−3
−2
−1
0
1
2
3
Differences in mean levels of Remedio


95% family−wise confidence level
M−F
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Differences in mean levels of Sexo


95% family−wise confidence level
C:M−A:F
C:M−C:F
A:M−B:M
−6
−4
−2
0
2
4
6
Differences in mean levels of Remedio:Sexo