

# NORMALIDAD

Etna Carolina Cortés Martínez

22/11/2021

```
install.packages("stats")
```

```
library(stats)
```

```
install.packages("nortest")
```

```
library(nortest)
```

## IMPORTACIÓN DE LA MATRIZ

1.- Lectura de la matriz

```
BD3<-read.csv("BD3_penguins.csv")
```

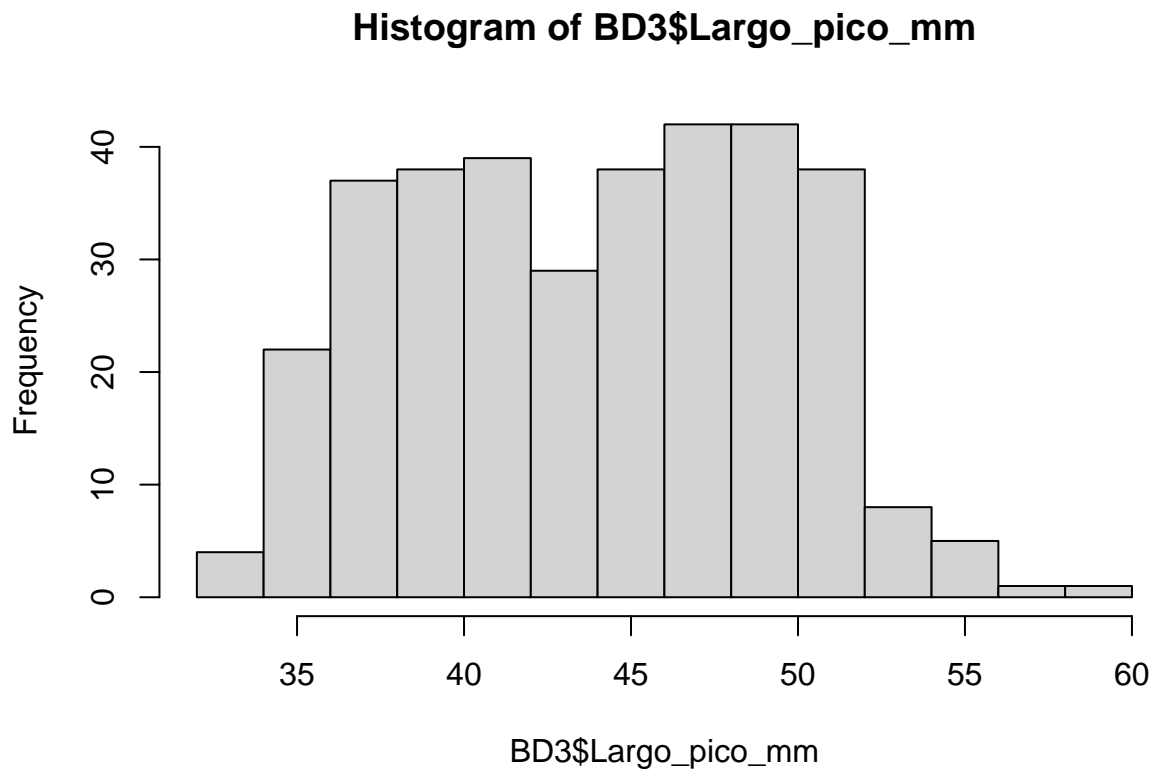
2.- Determinar el numero de observaciones

```
length(BD3$Largo_pico_mm)
```

```
## [1] 344
```

3.- Visualización de la distribución de la variable

```
hist(BD3$Largo_pico_mm)
```



## CREACIÓN DE LA MATRIZ <50

Variable: Masa\_corporal\_g

1.- visualizamos el nombre de las variables para identificar el número de la columna.

```
colnames(BD3)
```

```
## [1] "X"           "Especies"    "Isla"        "Largo_pico_mm"
## [5] "Ancho_pico_mm" "Largo_aleta_mm" "Masa_corporal_g" "Sexo"
## [9] "Año"
```

2.- Se seleccionan de las filas 4 ala 36 de la columna 7

```
peso<-BD3[4:36,7]
```

3.- Se visualiza la variable

```
peso
```

```
## [1] 3700 3450 3650 3625 4675 3475 4250 3300 3700 3200 3800 4400 3700 3450 4500
## [16] 3325 4200 3400 3600 3800 3950 3800 3800 3550 3200 3150 3950 3250 3900 3300
## [31] 3900 3325 4150
```

# KOLMOGOROV-SMIRNOV

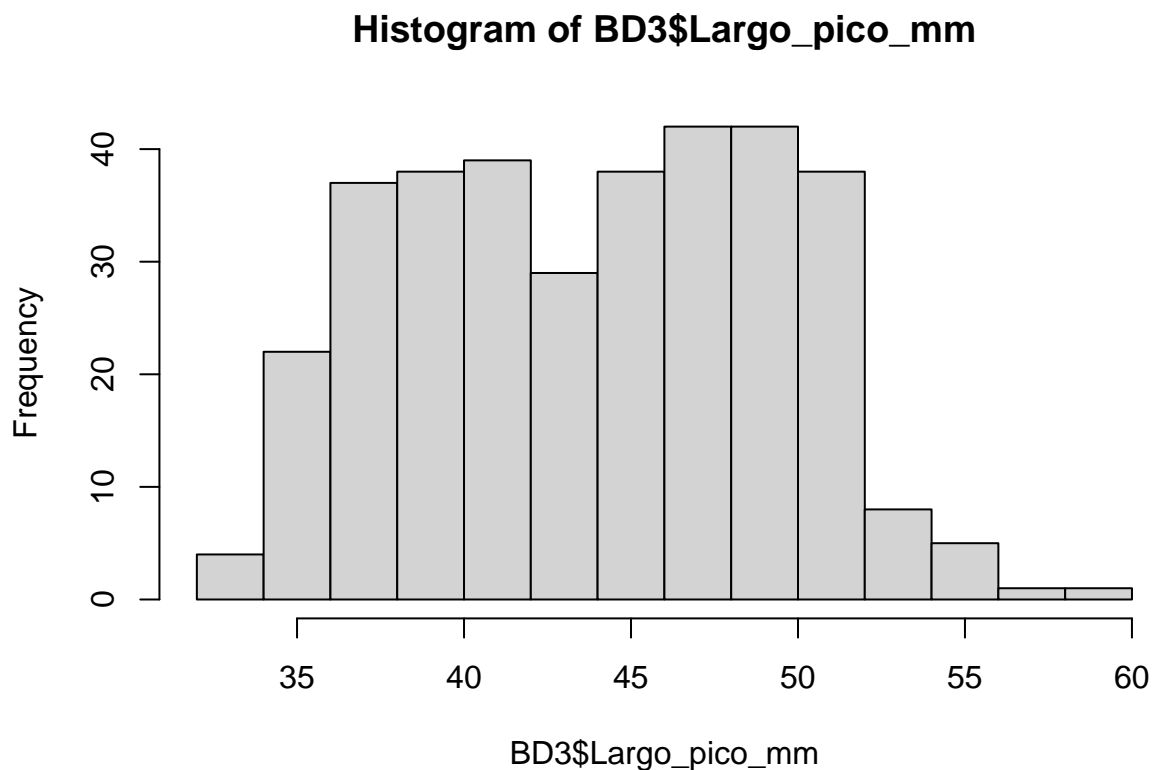
NOTA: Se aplica si tenemos mas de 50 observaciones

H0: la variable tiene una distribucion normal Ha: La variable tiene una distribucion diferente a lo normal

Interpretación: p-valor  $>0.05$  No rechazo H0  $<0.05$  rechazo H0

1.-Exploración de la variable

```
hist(BD3$Largo_pico_mm)
```



2.- Aplicación de la prueba de hipótesis

```
lillie.test(BD3$Largo_pico_mm)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  BD3$Largo_pico_mm  
## D = 0.070865, p-value = 0.0002714
```

## 3.- Interpretación

p.valor= 0.0002714, es menor a 0.05. por lo tanto, **RECHAZO H0**. Los datos siguen una distribución diferente a la normal.

# SHAPIRO WILKS

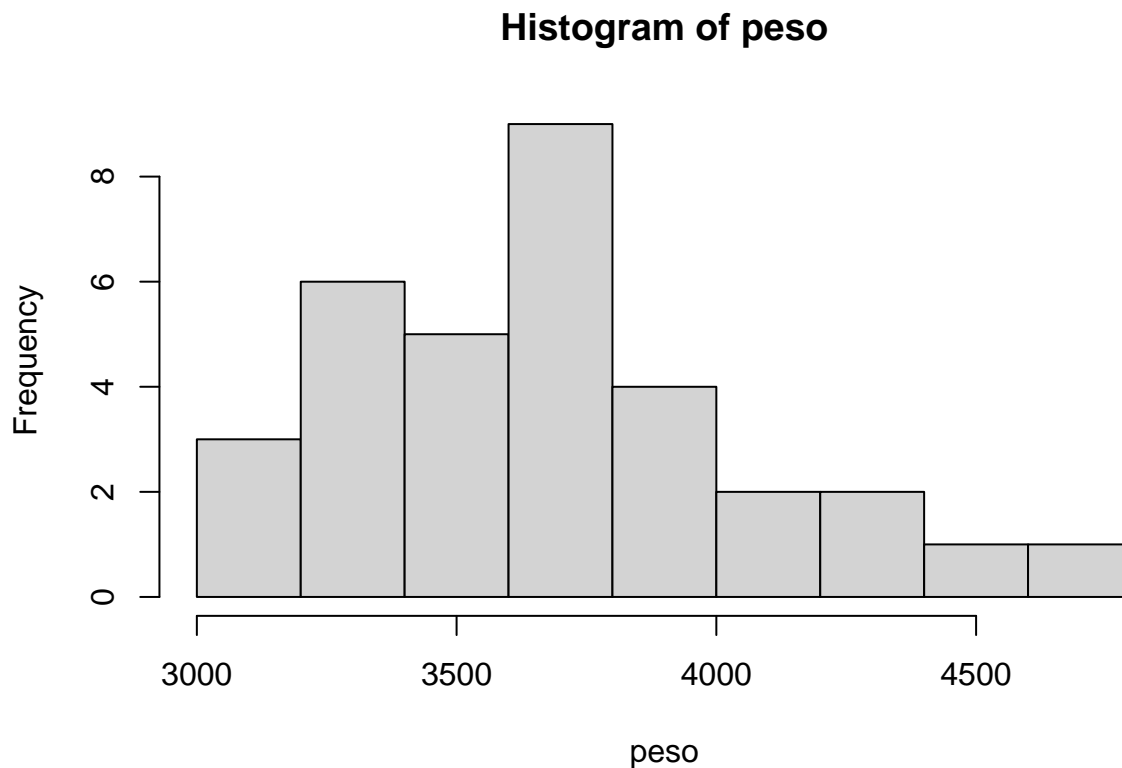
NOTA: Se aplica sí tenemos MENOS de 50 observaciones

Ho: La variable tiene distribución normal Ha: La variable tiene una distribución deferente a lo normal

Interpretación:  $p.\text{valor} > 0.05$  No rechazo Ho  $< 0.05$  rechazo Ho

1.- Exploración de la variable

```
hist(peso)
```



```
length(peso)
```

```
## [1] 33
```

2.- Prueba de hipótesis

```
shapiro.test(peso)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  peso  
## W = 0.94792, p-value = 0.1157
```

### 3.- Interpretación

p-valor: 0.1157, es mayor que 0.05. Por lo tanto. NO se rechaza  $H_0$ , eso quiere decir que los datos siguen una distribución normal.

### Importación de la Matriz

1.- Seleccionamos las filas de la especie Gentoo

```
BD3$Especies
```

```
## [1] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [7] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [13] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [19] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [25] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [31] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [37] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [43] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [49] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [55] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [61] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [67] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [73] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [79] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [85] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [91] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [97] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [103] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [109] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [115] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [121] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [127] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [133] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [139] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [145] "Adelie" "Adelie" "Adelie" "Adelie" "Adelie" "Adelie"
## [151] "Adelie" "Adelie" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [157] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [163] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [169] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [175] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [181] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [187] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [193] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [199] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [205] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [211] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [217] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [223] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [229] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [235] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
## [241] "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo" "Gentoo"
```

```
## [247] "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"
## [253] "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"
## [259] "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"
## [265] "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"
## [271] "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"      "Gentoo"
## [277] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [283] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [289] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [295] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [301] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [307] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [313] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [319] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [325] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [331] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [337] "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"   "Chinstrap"
## [343] "Chinstrap"   "Chinstrap"
```

```
gentoo<-BD3[153:277,]
```

## PRUEBA DE NORMALIDAD

2.- Realizamos la prueba de normalidad de Kolmogorov-Smirnov.

```
lillie.test(gentoo$Largo_pico_mm)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  gentoo$Largo_pico_mm
## D = 0.065044, p-value = 0.2183
```

3.- Interpretación:

**Ho:** La variable tiene distribución normal.

**Ha:** La variable tiene una distribución diferente a la normal.

## PEARSON

### PEARSON PARA 2 VARIABLES

Se implementa para datos cuantitativos con distribución normal.

*Se parte de la hipótesis:*

**Ho:** ( $p=0$ ) Las variables NO guardan una relación lineal entre ellas. **Ha:** ( $p\neq 0$ ) Las variables guardan una relación lineal entre ellas.

**NOTA:** Revisar el valor de la correlación.

# PEARSON PARA MUCHAS VARIABLES

Se instalará la paquetería corrplot y se activará

```
install.packages("corrplot")
```

```
library( corrplot )
```

```
## corrplot 0.92 loaded
```

1.- Preparación de la matriz.

Se seleccionan sólo las variables numéricas.

```
gentoo2 <- gentoo [, 4 : 7 ]
```

2.- Cálculo de la matriz de correlaciones.

```
cor_group <- round (cor ( gentoo2 ), 2 )
```

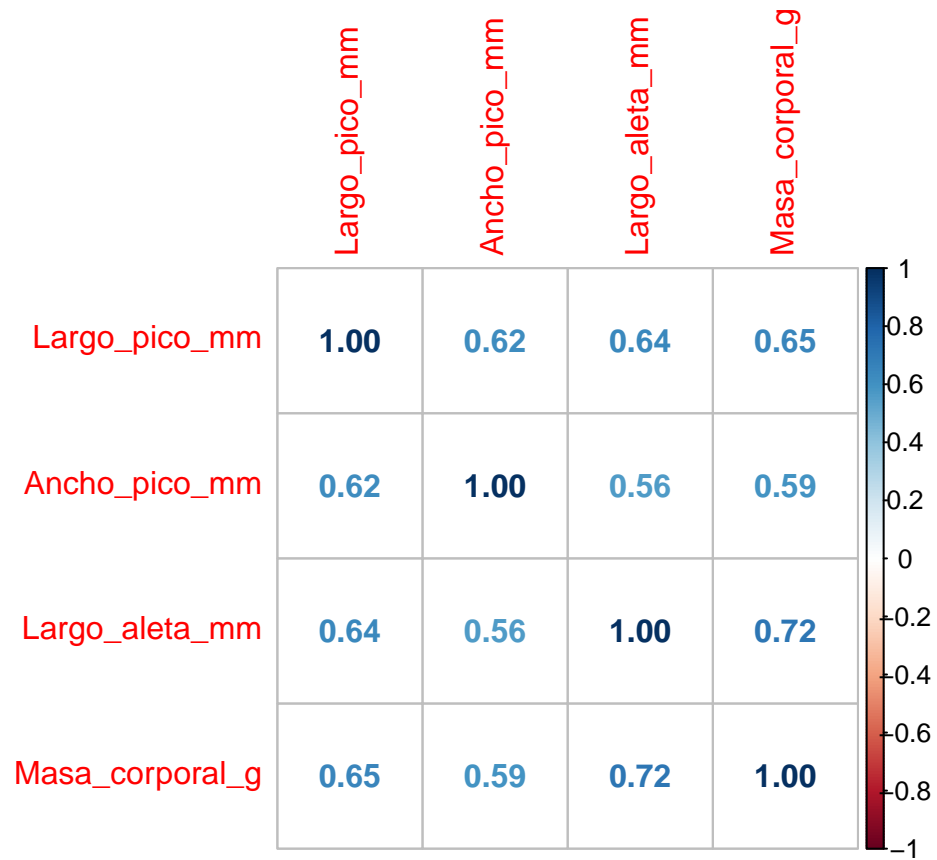
## 3.- Visualización de los resultados.

```
head(cor_group)
```

```
##                Largo_pico_mm Ancho_pico_mm Largo_aleta_mm Masa_corporal_g
## Largo_pico_mm                1.00          0.62          0.64          0.65
## Ancho_pico_mm                0.62          1.00          0.56          0.59
## Largo_aleta_mm                0.64          0.56          1.00          0.72
## Masa_corporal_g              0.65          0.59          0.72          1.00
```

4.- Generación del gráfico de forma completa.

```
corrplot(cor_group, method = c("number"), type="full")
```



#4.1.- Generación del gráfico del cuadrante inferior

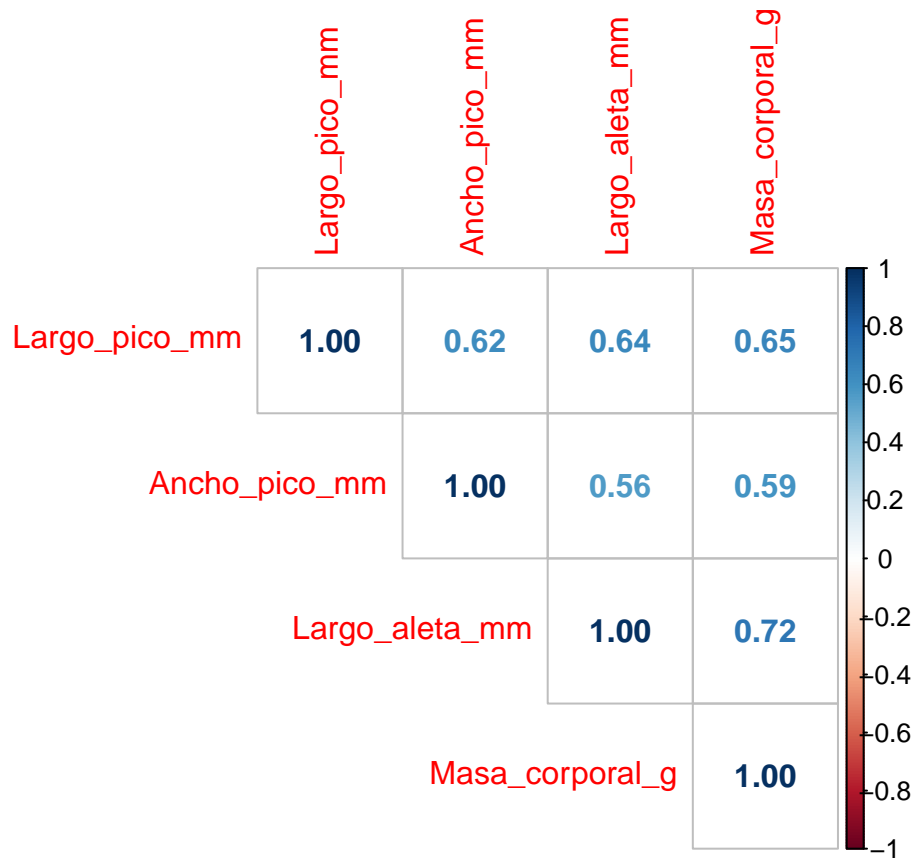
```
corrplot(cor_group, method = c("number"), type="lower")
```





#4.2.- Generación del gráfico del cuadrante superior

```
corrplot(cor_group, method = c("number"), type="upper")
```



5.- Cálculo del p-valor con nivel del confianza de 0.95, y alfa=0.05

```
cor.mtest(gentoo2, conf.level=0.95)
```

```
## $p
##           Largo_pico_mm Ancho_pico_mm Largo_aleta_mm Masa_corporal_g
## Largo_pico_mm  0.000000e+00  1.971893e-14  1.713497e-15  4.551314e-16
## Ancho_pico_mm  1.971893e-14  0.000000e+00  1.040731e-11  3.018587e-13
## Largo_aleta_mm  1.713497e-15  1.040731e-11  0.000000e+00  1.461511e-21
## Masa_corporal_g  4.551314e-16  3.018587e-13  1.461511e-21  0.000000e+00
##
## $lowCI
##           Largo_pico_mm Ancho_pico_mm Largo_aleta_mm Masa_corporal_g
## Largo_pico_mm  1.0000000  0.4943008  0.5176208  0.5296995
## Ancho_pico_mm  0.4943008  1.0000000  0.4271972  0.4664795
## Largo_aleta_mm  0.5176208  0.4271972  1.0000000  0.6281373
## Masa_corporal_g  0.5296995  0.4664795  0.6281373  1.0000000
##
## $uppCI
##           Largo_pico_mm Ancho_pico_mm Largo_aleta_mm Masa_corporal_g
## Largo_pico_mm  1.0000000  0.7146527  0.7296498  0.7373381
## Ancho_pico_mm  0.7146527  1.0000000  0.6703380  0.6964919
## Largo_aleta_mm  0.7296498  0.6703380  1.0000000  0.7980530
## Masa_corporal_g  0.7373381  0.6964919  0.7980530  1.0000000
```

6.- Generación del gráfico con diagrama de dispersión, coeficiente de correlación, nivel de significancia e

```
histograma.
```

```
install.packages("PerformanceAnalytics")
```

```
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 4.0.5
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.0.5
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.0.5
```

```
##
```

```
## Attaching package: 'zoo'
```

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

```
##
```

```
##      as.Date, as.Date.numeric
```

```
##
```

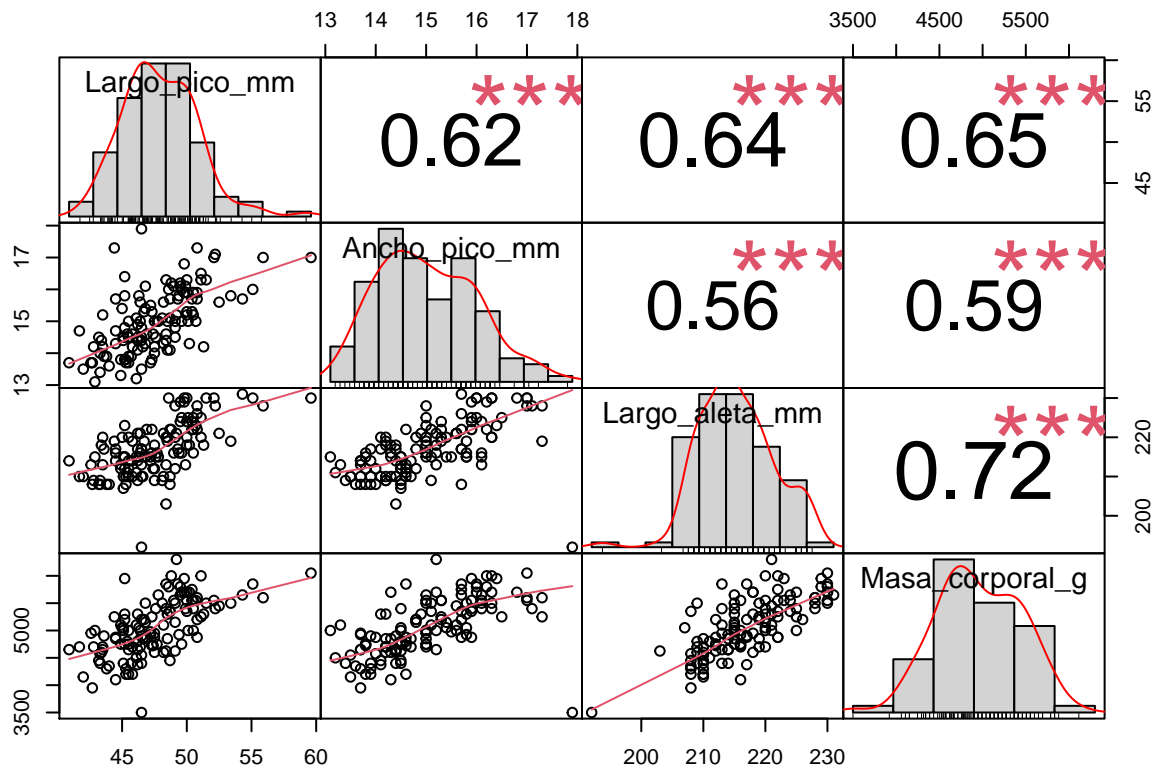
```
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
```

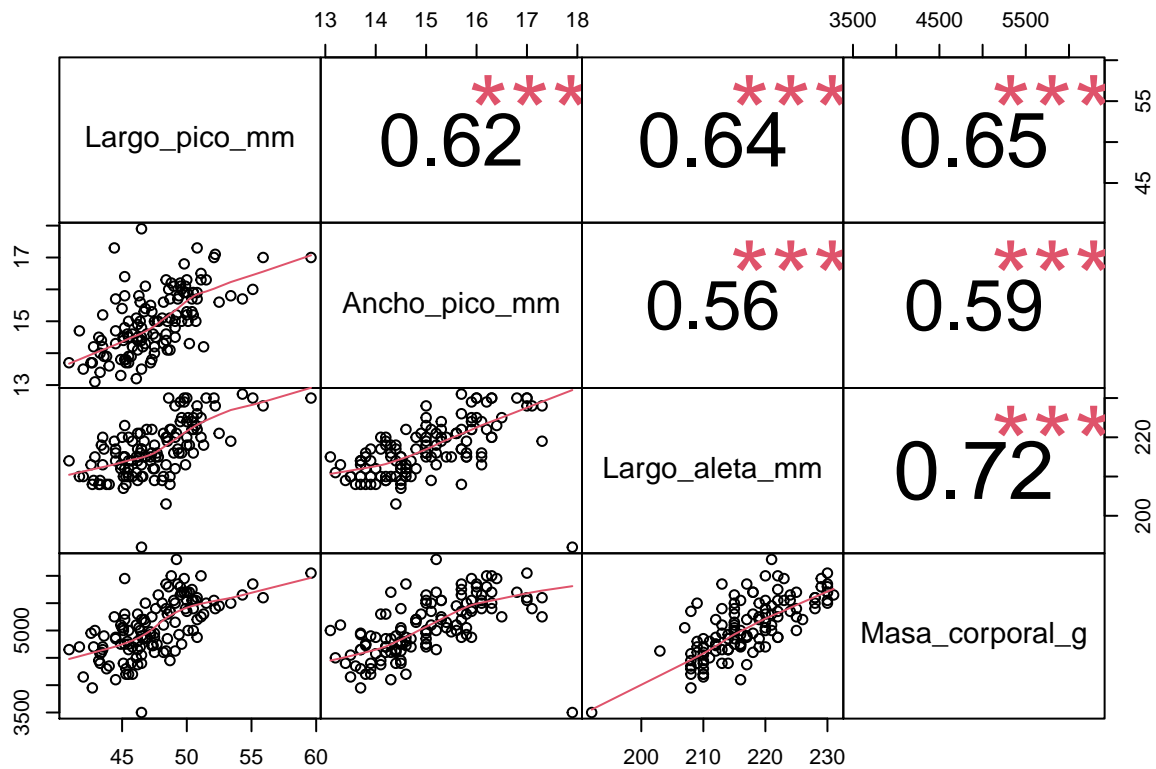
```
##
```

```
##      legend
```

```
chart.Correlation(gentoo2, histogram = T, method= "pearson", pch=18)
```



```
chart.Correlation(gentoo2, histogram = F, method= "pearson", pch=18)
```



## RHO SPEARMAN

Se implementa cuando los datos tiene una distribución *DIFERENTE* a la Normal.

*Se parte de la hipótesis:*

**Ho:** ( $p=0$ ) Las variables NO guardan una relación lineal entre ellas. **Ha:** ( $p \neq 0$ ) Las variables guardan una relación lineal entre ellas.

## Matriz de datos.

1.- Creación de la matriz.

1.1.- Generación de calificaciones ficticias para las asignaturas de matemáticas, español, historia y geografía.

```
set.seed(5)
```

```
mate<-sample(5:10, size= 45, replace=TRUE)
```

```
espa<-sample(5:10, size= 45, replace = TRUE)
```

```
hist<-sample(5:10, size= 45, replace= TRUE)
```

```
geogr<-sample(5:10, size= 45, replace= TRUE)
```

1.2.- Unir la matriz con las diferentes asignaturas

```
calif_2<-data.frame(mate, espa, hist, geogr)
```

1.3.- Explorar la matriz calif\_2

```
View(calif_2)
```

```
str(calif_2)
```

```
## 'data.frame': 45 obs. of 4 variables:
## $ mate : int 6 7 5 7 5 5 9 10 7 7 ...
## $ espa : int 9 7 10 9 7 7 7 6 8 5 ...
## $ hist : int 10 9 8 10 10 7 5 10 5 8 ...
## $ geogr: int 7 7 7 9 7 7 7 9 7 9 ...
```

1.4.- Sacar una copia de la matriz generada.

```
calif_3<-calif_2
```

1.5.- Codificar la matriz de integer a numeric

```
calif_3$mate<-as.numeric(calif_3$mate)
```

```
calif_3$espa<-as.numeric(calif_3$espa)
```

```
calif_3$hist<-as.numeric(calif_3$hist)
```

```
calif_3$geogr<-as.numeric(calif_3$geogr)
```

1.6.- Verificación de los cambios

```
View(calif_3)
```

```
str(calif_3)
```

```
## 'data.frame': 45 obs. of 4 variables:
## $ mate : num 6 7 5 7 5 5 9 10 7 7 ...
## $ espa : num 9 7 10 9 7 7 7 6 8 5 ...
## $ hist : num 10 9 8 10 10 7 5 10 5 8 ...
## $ geogr: num 7 7 7 9 7 7 7 9 7 9 ...
```

## PRUEBA DE NORMALIDAD

```
shapiro.test(calif_3$mate)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: calif_3$mate  
## W = 0.9189, p-value = 0.003882
```

```
shapiro.test(calif_3$espa)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: calif_3$espa  
## W = 0.91278, p-value = 0.002428
```

## RHO DE SPEARMAN PARA 2 VARIABLES

1.- Realizar el cálculo de Rho.

```
spearman<-cor.test(calif_3$mate,calif_3$espa, method = "spearman")
```

```
## Warning in cor.test.default(calif_3$mate, calif_3$espa, method = "spearman"):  
## Cannot compute exact p-value with ties
```

2.- Visualizar el resultado

```
spearman
```

```
##  
## Spearman's rank correlation rho  
##  
## data: calif_3$mate and calif_3$espa  
## S = 18040, p-value = 0.2151  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## -0.1884368
```

3.- Interpretación de los resultados (Anota la interpretación) Se obtuvo una Rho de **0.1884**, lo que significa que las variables no guardan correlación lineal.

## Rho de SPEARMAN PARA MUCHAS VARIABLES

2.- Generación del gráfico de dispersión, histograma, correlaciones y significancia.

```
chart.Correlation(calif_3, histogram = T, method= "spearman", pch=18)
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):  
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):  
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):  
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):  
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):  
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):  
## Cannot compute exact p-value with ties
```

