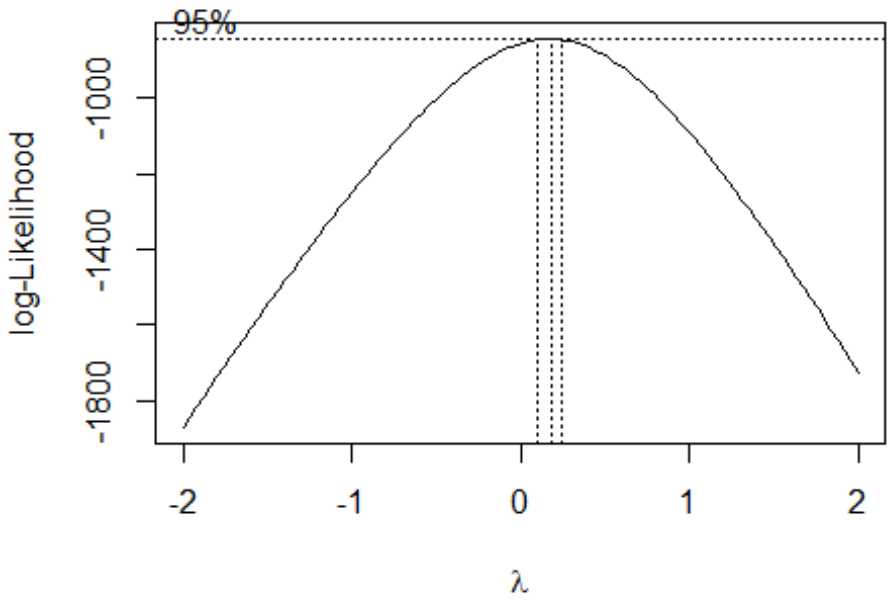
Act 3

Ana Lucía Cárdenas Pérez A01284090

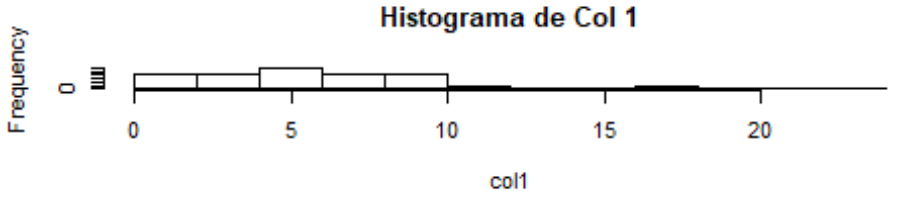
2023-08-22

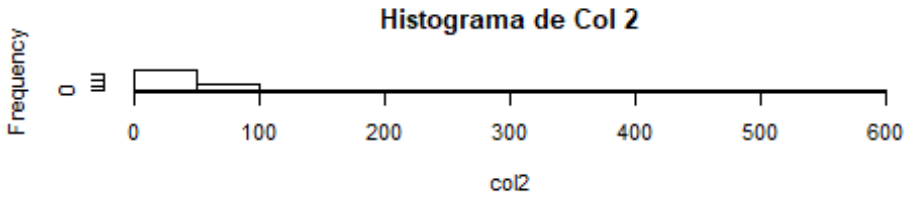
```
# Cargamos archivo
data <- read.csv("mc-donalds-menu-1.csv")

#Guardamos la columna de Cholesterol en una variable Colesterol
colesterol <- data$Cholesterol

library(MASS)
# Aplicar la transformación Box-Cox
bc <- boxcox((data[, 11] + 1) ~ 1)
```

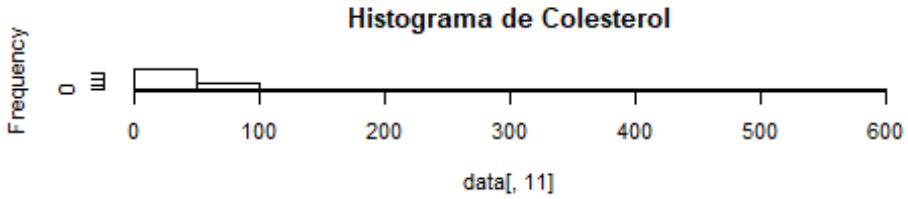



```
bc$x[which.max(bc$y)]

## [1] 0.1818182

#Creamos los histogramas del modelo exacto, aproximado y el de los datos originales
col1 = sqrt(data[,11]+1)
col2 = ((data[,11]+1)^1-1/1)
par(mfrow = c(3,1))
```

```r
hist(col1,col = 0, main = "Histograma de Col 1")
hist(col2,col = 0, main = "Histograma de Col 2")
hist(data[,11], col = 0, main = "Histograma de Colesterol")
```








```r
options(repos = c(CRAN = "https://cran.rstudio.com/"))

#Obtenemos un resumen de información incluyendo Q1, Q3, Min, Median, Mean,
Max, Sesgo, y Curtosis.
install.packages("e1071")
```

```
## Installing package into 'C:/Users/anaca/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'e1071' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\anaca\AppData\Local\Temp\Rtmpszw99c\downloaded_packages
```

```r
library(e1071)
  summary(colesterol)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.00   35.00   54.94   65.00  575.00
```

```r
print("Curtosis de Colesterol")
```

```
## [1] "Curtosis de Colesterol"
```

```r
  kurtosis(colesterol)
```

```
## [1] 16.87947
```

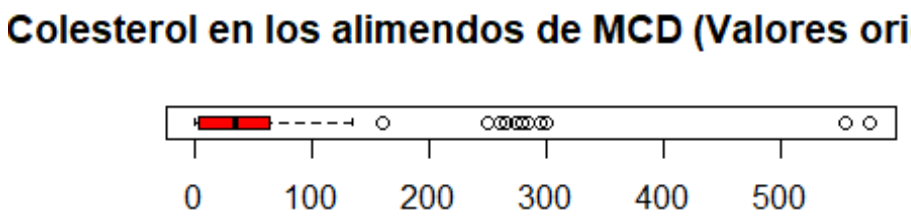
```r
print("Sesgo de Colesterol")
```

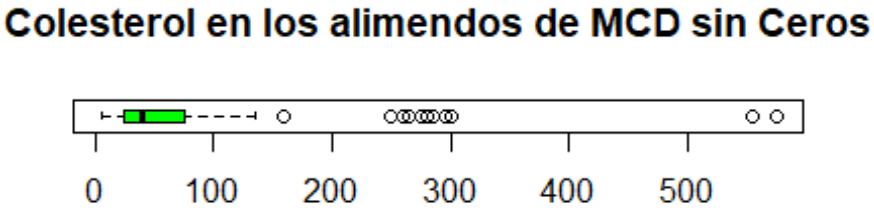
```
## [1] "Sesgo de Colesterol"
```

```r
  skewness(colesterol)
```

```
## [1] 3.755186
```

```r
# Crear dos conjuntos de datos basados en si los valores de 'Cholesterol' son mayores que cero
dataSinCeros <- subset(data, Cholesterol > 0)

par(mfrow = c(2,1))
# Crear boxplots para los dos conjuntos de datos
boxplot(data$Cholesterol, horizontal = TRUE, col = "red", main = "Colesterol en los alimendos de MCD (Valores originales)")

boxplot(dataSinCeros$Cholesterol, horizontal = TRUE, col = "green", main = "Colesterol en los alimendos de MCD sin Ceros")
```






```r
# Empezamos con la prueba de normalidad, en este caso la de Anderson-Darling para los datos transdormados y los datos normales.
install.packages("nortest")
```

```
## Installing package into 'C:/Users/anaca/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'nortest' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\anaca\AppData\Local\Temp\Rtmpszw99c\downloaded_packages

library(nortest)
D0 = ad.test(data[,11])
D1 = ad.test(col1)
D2 = ad.test(col2)

library(e1071)
m0 = round(c(as.numeric(summary(data[,11])), kurtosis(data[,11]),
skewness(data[,11]),D0$p.value),3)

m1 = round(c(as.numeric(summary(col1)), kurtosis(col1),
skewness(col1),D1$p.value),3)

m2 = round(c(as.numeric(summary(col2)), kurtosis(col2),
skewness(col2),D2$p.value),3)

m <- as.data.frame(rbind(m0, m1, m2))

row.names(m) = c("Original", "Primer modelo", "Segundo Modelo")

names(m) = c("Minimo", "Q1", "Mediana", "Media", "Q3", "Maximo", "Curtosis",
"Sesgo", "Valor p")

knitr::kable(m, format = "html", caption = "Resumen de estadísticas")
```

Resumen de estadísticas

Minimo

Q1

Mediana

Media

Q3

Maximo

Curtosis

Sesgo

Valor p

Original

0

5.000

35

54.942

65.000

575

16.879

3.755

0

Primer modelo

1

2.449

6

6.107

8.124

24

3.384

1.473

0

Segundo Modelo

0

5.000

35

54.942

65.000

575

16.879

3.755

0

```
# Instala y carga el paquete nortest (si no lo has hecho)
install.packages("nortest")
```

```
## Warning: package 'nortest' is in use and will not be installed
```

```
library(nortest)

# Realiza la prueba de normalidad de Anderson-Darling para los datos
originales
adOriginal <- ad.test(colesterol)
print("Prueba de Anderson-Darling para datos originales:")
```

```
## [1] "Prueba de Anderson-Darling para datos originales:"
```

```
print(adOriginal)
```

```
##
##  Anderson-Darling normality test
##
## data:  colesterol
## A = 31.884, p-value < 2.2e-16
```

```
# Realiza la prueba de normalidad de Anderson-Darling para los datos
transformados
adTransformado <- ad.test(col1)
print("Prueba de Anderson-Darling para datos transformados:")
```

```
## [1] "Prueba de Anderson-Darling para datos transformados:"
```

```
print(adTransformado)
```

```
##
##  Anderson-Darling normality test
##
## data:  col1
## A = 6.5954, p-value = 3.274e-16
```

```
#Transformacion Yeo-Johnson
install.packages("VGAM")
```

```
## Installing package into 'C:/Users/anaca/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'VGAM' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\anaca\AppData\Local\Temp\Rtmpszw99c\downloaded_packages
```

```r
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

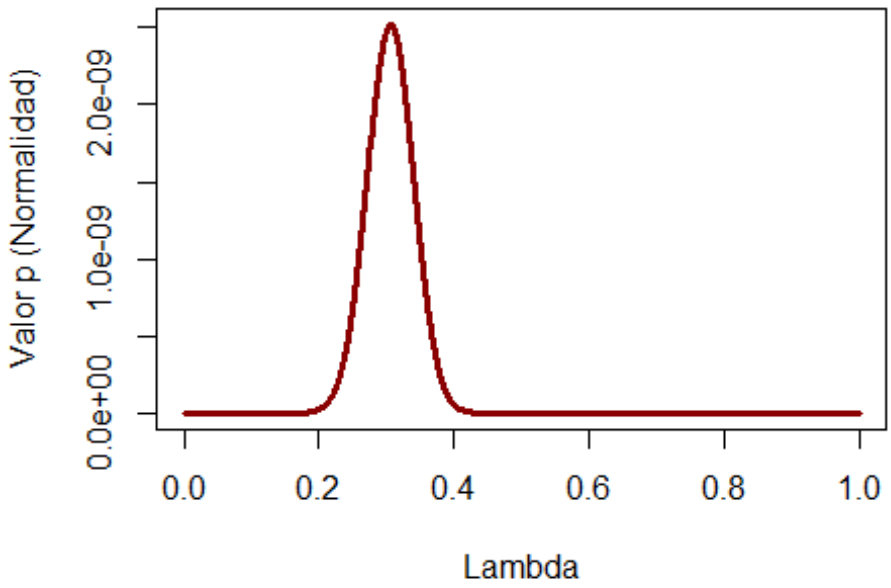
```r
col3 <- yeo.johnson(data[,11],lambda = 1)
```

```r
print(col3)
```

```
##   [1] 260  25  45 285  50 300 250 250  35  35  30  30 250 250  35  35  30
30
##  [19] 280 250  35  35 265  50 275  60 295 555 555  35  35 575 575  55  55
20
##  [37]  50 115   0  15   5   5  85  95 105 105  85 160  30  45  90 115  75
90
##  [55]  80  80  70  45  65  85 105  70  90  90 110  45  35  65  50  35  80
95
##  [73]  65  80  60  80  50  65  25  40  65 135 265  40  25  70  85  10  50
70
##  [91]  30  40  35  45  40  45   0   0   0   0   0   0   5   0  10  10   5
25
## [109]  30  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
## [127]   0   0   0   0  10   5   0   0   0   0   0   0   0   0   0   0   0
0
## [145]   0   0   0   0  25  30  40  25  30  40  25  30  40  25  30  40  25
30
## [163]  40   5   5  10   5   5  10   5   5  10   5   5  10   5   5  10  35
40
## [181]  50  15  15  20  35  40  50  15  15  20  40  50  60  15  15  20  15
25
## [199]  35  15  25  35  15  25  35  15  25  35  15  25  35  35  40  50  20
20
## [217]  25  35  40  50  20  20  25  65  75  90  65  80  95  65  80  95   5
5
## [235]   5   5   5   5   5   5   5  60  75  90  60  75  90  60  75  85  75
90
## [253]  50  75  35  45  55  30  60  30
```

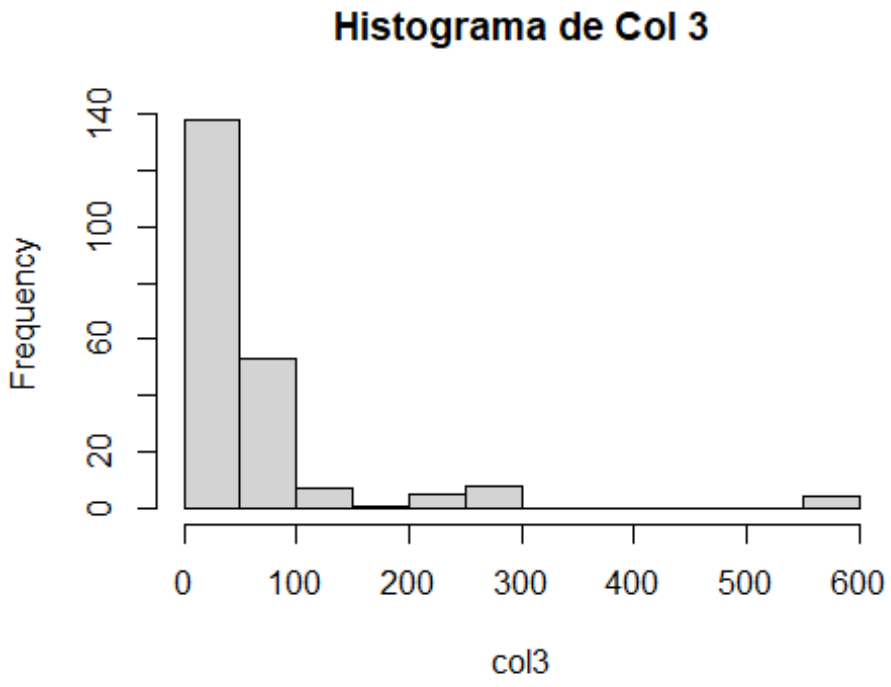
```r
library(nortest)
library(VGAM)
lp <- seq(0,1,0.001)
nlp <- length(lp)
n = length(data[,11])
#D <- matrix(as.numeric(NA, ncol = 2, nrow = nlp))
D <- matrix(nrow = nlp, ncol = 2)
d <- NA

for (i in 1:nlp){
  d = yeo.johnson(data[ ,11], lambda = lp[i])
```

```
  p = ad.test(d)
  D[i,] = c(lp[i],p$p.value)
}

N = as.data.frame(D)
colnames(N) <- c("Lambda", "Valor-p")
plot(N$Lambda,N$`Valor-p`,
type = "l",col = "darkred",lwd = 3,
xlab = "Lambda",
ylab = "Valor p (Normalidad)")
```

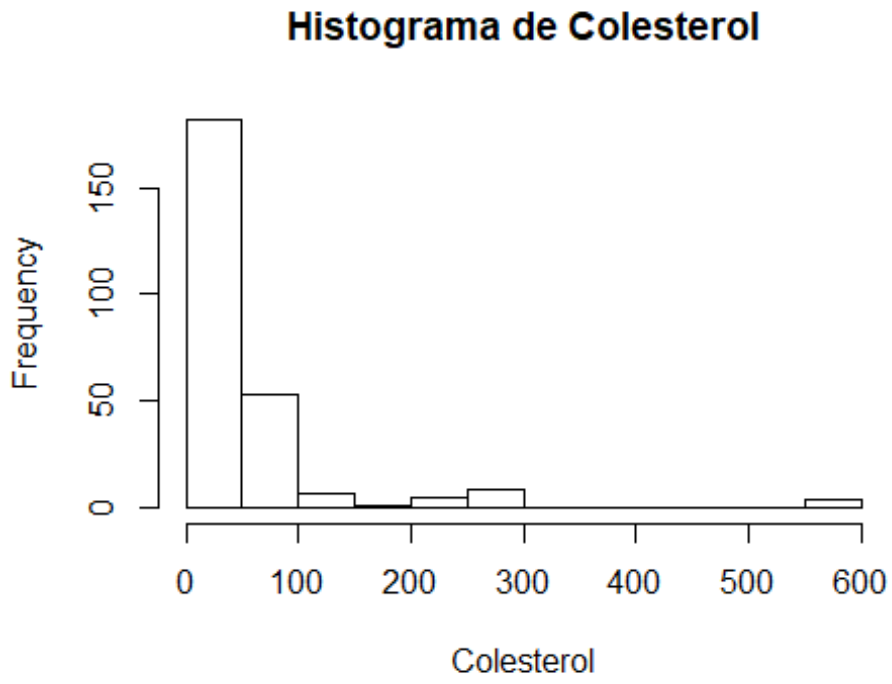



```
G = data.frame(subset(N,N$`Valor-p` == max(N$`Valor-p`)))
print(G)

##     Lambda      Valor.p
## 307  0.306 2.517614e-09

col3 <- dataSinCeros$Cholesterol

print(col3)

##   [1] 260  25  45 285  50 300 250 250  35  35  30  30 250 250  35  35  30
30
##  [19] 280 250  35  35 265  50 275  60 295 555 555  35  35 575 575  55  55
20
##  [37]  50 115  15   5   5  85  95 105 105  85 160  30  45  90 115  75  90
80
##  [55]  80  70  45  65  85 105  70  90  90 110  45  35  65  50  35  80  95
```

```
65
##  [73]  80  60  80  50  65  25  40  65 135 265  40  25  70  85  10  50  70
30
##  [91]  40  35  45  40  45   5  10  10   5  25  30  25  10   5  25  30  40
25
## [109]  30  40  25  30  40  25  30  40  25  30  40   5   5  10   5   5  10
5
## [127]   5  10   5   5  10   5   5  10  35  40  50  15  15  20  35  40  50
15
## [145]  15  20  40  50  60  15  15  20  15  25  35  15  25  35  15  25  35
15
## [163]  25  35  15  25  35  35  40  50  20  20  25  35  40  50  20  20  25
65
## [181]  75  90  65  80  95  65  80  95   5   5   5   5   5   5   5   5   5
60
## [199]  75  90  60  75  90  60  75  85  75  90  50  75  35  45  55  30  60
30

par(mfrow = c(1,1))
hist(col3 , main = "Histograma de Col 3")
```




```
hist(data[,11], col = 0, main = "Histograma de Colesterol", xlab =
"Colesterol")
```




```
library(e1071)
  summary(col3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   25.00   40.00   66.13   75.00  575.00
```

```
print("Curtosis Col3")
```

```
## [1] "Curtosis Col3"
```

```
  kurtosis(col3)
```

```
## [1] 14.67617
```

```
print("Sesgo Col3")
```

```
## [1] "Sesgo Col3"
```

```
  skewness(col3)
```

```
## [1] 3.558667
```

En la primera transformación nos dio un valor de curtosis de 16 mientras que la transformación de yeo johnson nos dió una curtosis de 16. Los valores de Sesgo en ambas pruebas fueron de 3.7 (primera prueba), 3.5 (segunda prueba).

1. Ventajas y desventajas de Box Cox y Yeo Johnson - Box Cox * Ventajas Bueno cuando hay valores con distribución exponencial. * Desventajas Malo cuando hay valores negativos y ceros. Puede producir valores no válidos.

- Yeo Johnson
  - Ventajas Bueno cuando hay valores negativos y ceros. Puede manejar más valores de las distribuciones de datos.
  - Desventajas Díficil de interpretear.

2. Diferencias entre transformación y escalamiento de datos: 2a. Escribe al menos 3 diferencias entre lo que es la transformación y el escalamiento de los datos - Transformación * Cambia la manera en la que vemos los datos a una forma más parecida a números normales. * Bueno cuando se necesitan ajustar los valores para pruebas de estadística. * La interpretación de los datos puede ser más complicadas.

- Escalamiento
  - Los datos se mantienen iguales, solo cambian los rangos.
  - Bueno para comparar los datos que estén en escalas distintas.
  - Bueno cuando se tienen algoritmos sensibles a las escalas.

2b. Indica cuándo es necesario utilizar cada uno - Transformación * Se usa cunaod los datos se necesitan ajustar para poder llevar a cabo las pruebas estadísticas.

- Escalamiento
  - Se usa cuando se quieren comparar variables con diferentes unidades o rangos, o cuando los algoritmos a usar, son más sensibles.