Actividad 5. Transformaciones

José Carlos Sánchez Gómez

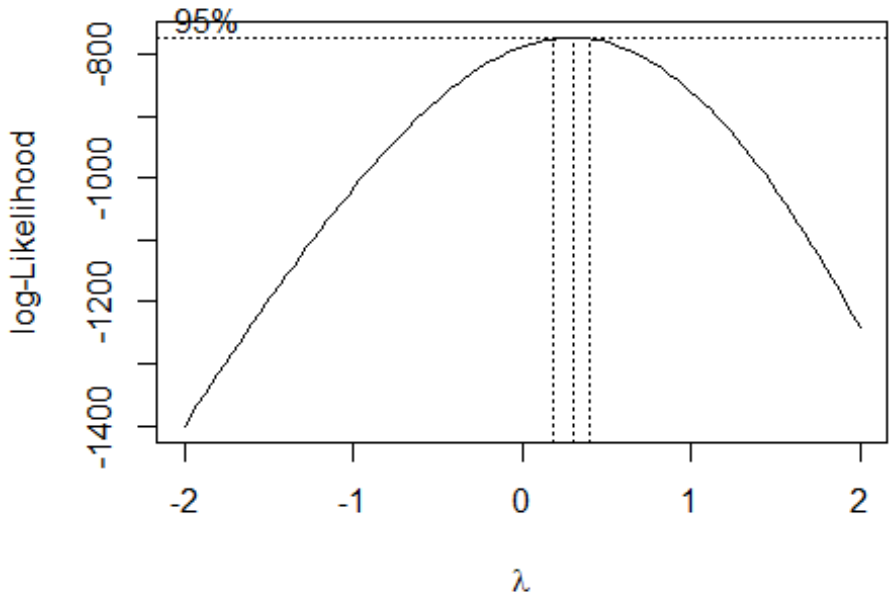
2024-08-15

Leyendo los datos del excel

```
data =
read.csv("C:\\Users\\jcsg6\\Documentos\\Uni\\SeptimoSemestre\\Estadistica
\\mc-donalds-menu.csv")
fat = data$Total.Fat
```

Obteniendo la lambda que maximiza la función de verosimilitud

```
library(MASS)
bc = boxcox((data$Total.Fat + 1) ~ 1)
```

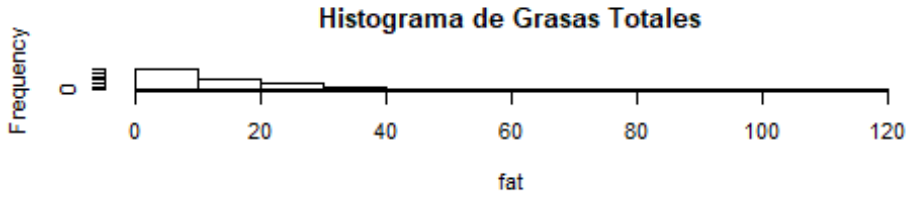


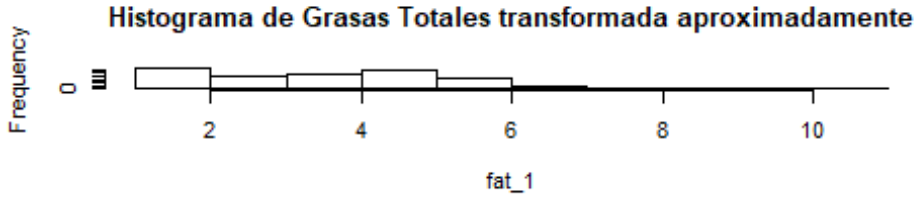

```
l = bc$x[which.max(bc$y)]
l

## [1] 0.3030303
```

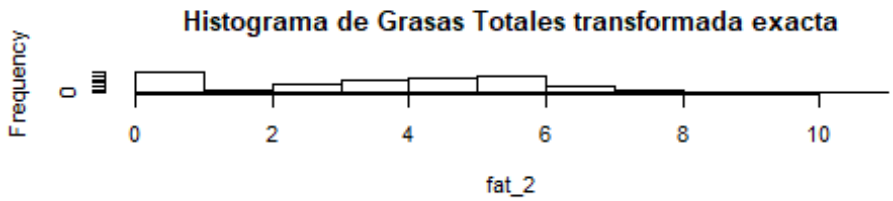
Uso de modelo exacto y aproximado de Box-Cox

Dado que nuestra lambda es de 0.3030, la función recomendada para la aproximación es $\sqrt{(x)}$, y para la exacta será $\frac{x^{\lambda}-1}{\lambda}$, que quedaria como $\frac{x^{(0.3030)}-1}{0.3030}$

```
# Histograma con la información original
par(mfrow=c(3,1))
hist(fat, col=0, main="Histograma de Grasas Totales")
# Histograma con boxcox aproximado
fat_1 = sqrt(fat + 1)
hist(fat_1, col= 0, main="Histograma de Grasas Totales transformada
aproximadamente")
# Histograma con boxcox exacto
fat_2 = ((fat + 1)^l - 1) / l
hist(fat_2, col= 0, main="Histograma de Grasas Totales transformada
exacta")
```








```
library(e1071)
library(nortest)

# resumen de los datos normales
fat_summary = summary(fat)
fat_kurtosis = kurtosis(fat)
fat_sesgo = skewness(fat)
fat_aproximado_summary = summary(fat_1)
fat_exacto_summary = summary(fat_2)
```

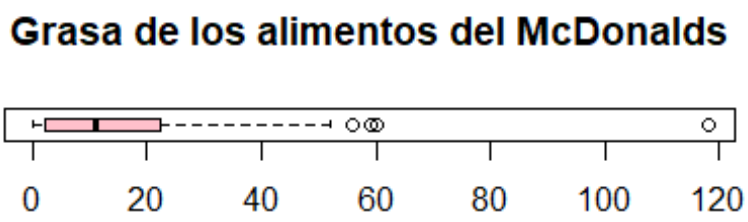
```r
p_value_normal = ad.test(fat)$p.value
p_value_aproximado = ad.test(fat_1)$p.value
p_value_exacto = ad.test(fat_2)$p.value

datos = data.frame(
  Estadistico = c(names(fat_summary), "Curtosis", "Sesgo", "P-Value"),
  Orginal = c(as.numeric(fat_summary), fat_kurtosis, fat_sesgo,
p_value_normal),
  "Modelo Aproximado" = c(as.numeric(fat_aproximado_summary),
kurtosis(fat_1), skewness(fat_1), p_value_aproximado),
  "Modelo Exacto" = c(as.numeric(fat_exacto_summary), kurtosis(fat_2),
skewness(fat_2), p_value_exacto)
)
datos
```

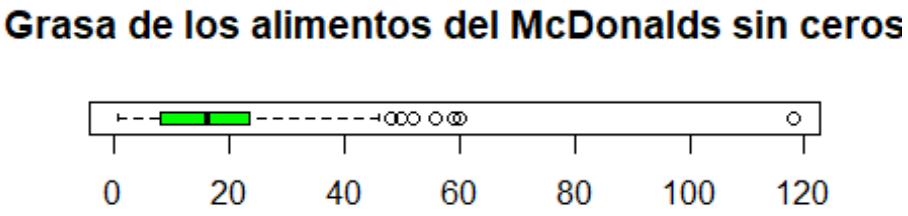
```
##   Estadistico      Orginal Modelo.Aproximado Modelo.Exacto
## 1        Min. 0.000000e+00      1.000000e+00  0.000000e+00
## 2     1st Qu. 2.375000e+00      1.836134e+00  1.468694e+00
## 3      Median 1.100000e+01      3.464102e+00  3.707104e+00
## 4        Mean 1.416538e+01      3.450438e+00  3.432516e+00
## 5     3rd Qu. 2.225000e+01      4.821619e+00  5.261814e+00
## 6        Max. 1.180000e+02      1.090871e+01  1.074325e+01
## 7    Curtosis 1.035171e+01     -8.053187e-02 -8.519420e-01
## 8       Sesgo 2.128023e+00      3.078819e-01 -1.151632e-01
## 9     P-Value 1.463660e-16      6.861263e-10  1.380766e-14
```

Quitando los ceros del modelo

Las opciones dentro del menú del McDonalds corresponden a bebidas y agua, las eliminaremos del modelo, ya que, estas no son muestras representativas de lo que es el menú del restaurante.

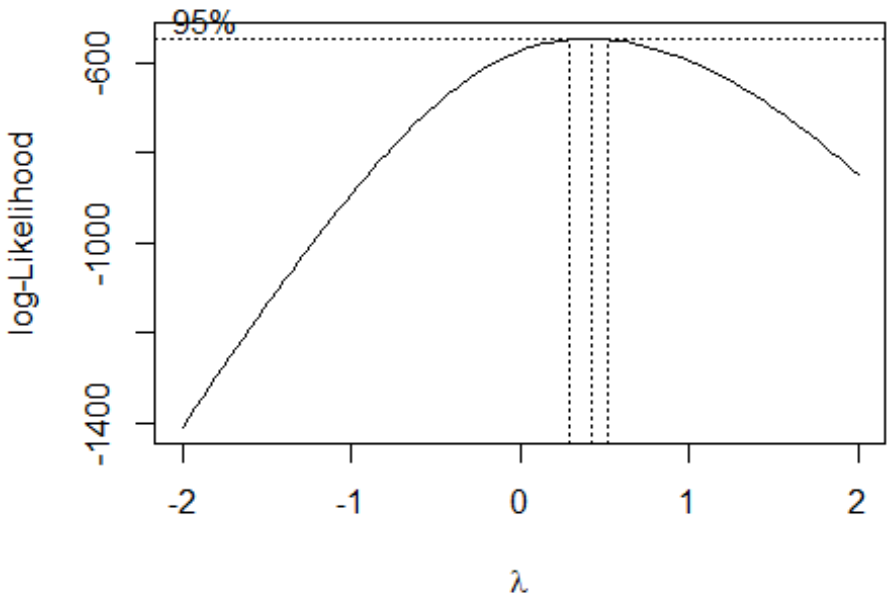
```r
filtered_fat = subset(fat, fat > 0)
par(mfrow = c(2, 1))
boxplot(fat, horizontal = TRUE, col = 'pink', main = "Grasa de los
alimentos del McDonalds")
boxplot(filtered_fat, horizontal = TRUE, col = 'green', main = "Grasa de
los alimentos del McDonalds sin ceros")
```



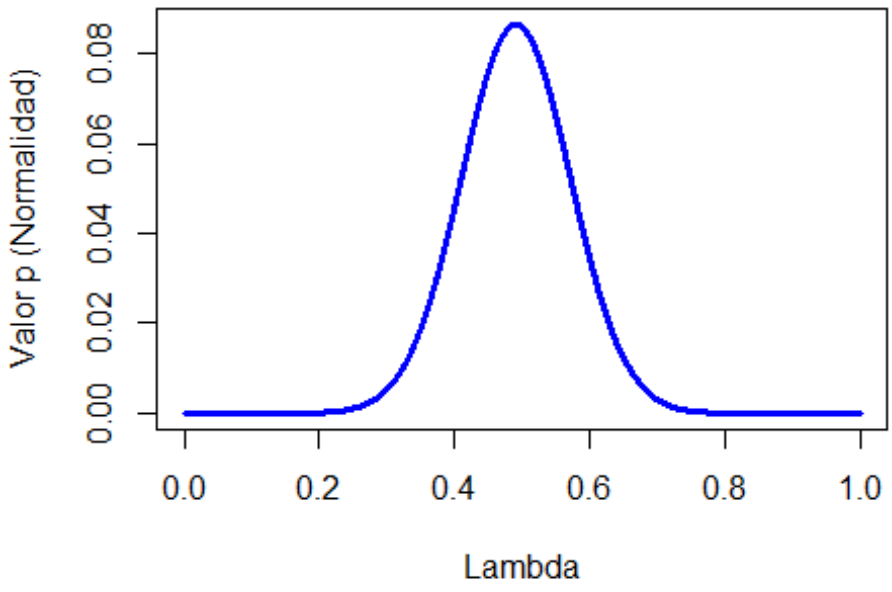


Obtencion del modelo mediante Yeo-Johnson

```
library(MASS)
bc_zeros = boxcox((filtered_fat) ~ 1)
```

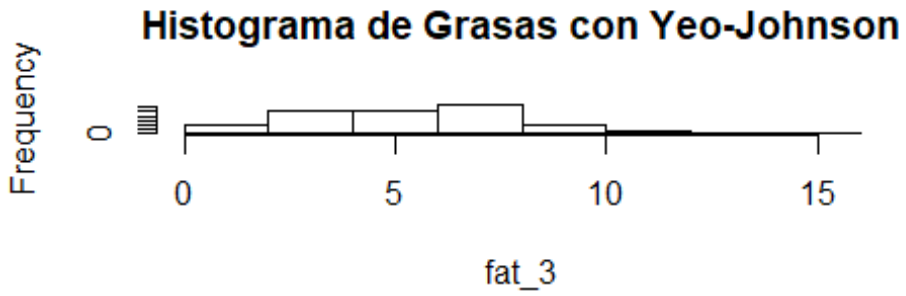


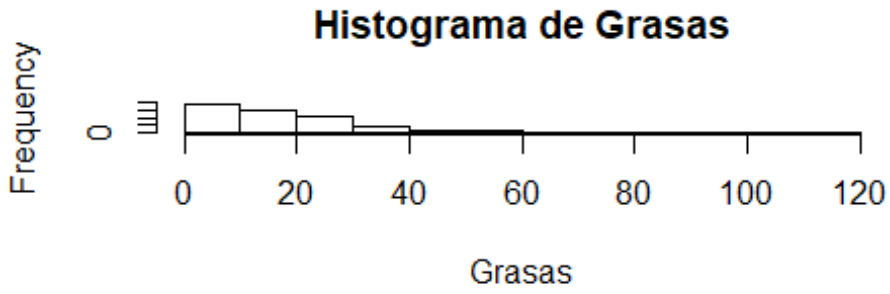
```
l_zeros = bc_zeros$x[which.max(bc_zeros$y)]

library(VGAM)

## Cargando paquete requerido: stats4

## Cargando paquete requerido: splines

fat_3 = yeo.johnson(filtered_fat, lambda = l_zeros)

lp <- seq(0,1,0.001) # Valores de Lambda propuestos
nlp <- length(lp)
n=length(filtered_fat)
D <- matrix(as.numeric(NA),ncol=2,nrow=nlp)
d <-NA
for (i in 1:nlp){
d= yeo.johnson(filtered_fat, lambda = lp[i])
p=ad.test(d)
D[i,]=c(lp[i],p$p.value)}

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




```
G = data.frame(subset(N, N$`P-Value` == max(N$`P-Value`)))
G
```

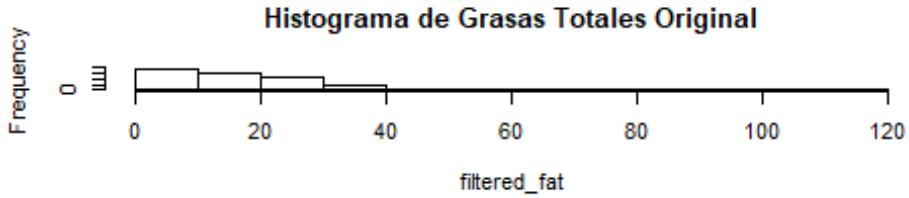
```
##     Lambda    P.Value
## 492  0.491 0.08644269

par(mfrow = c(2,1))
hist(fat_3, col = 0, main = "Histograma de Grasas con Yeo-Johnson")
hist(filtered_fat, col = 0, main = "Histograma de Grasas", xlab =
"Grasas")
```

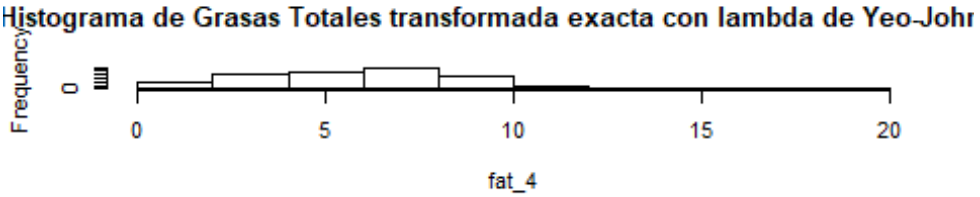


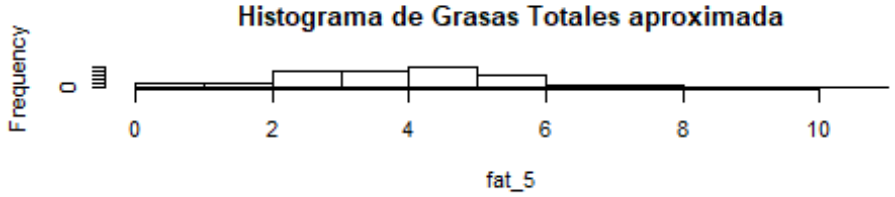


Tras hacer la transformacion de Yeo-Johnson, encontramos que el valor de lambda para maximizar p, es de 0.491, por lo que la ecuación para el modelo quedaría de esta forma $\frac{x^{(0.491)}-1}{0.491}$

```
library(e1071)
library(nortest)

par(mfrow = c(3, 1))
hist(filtered_fat, col= 0, main="Histograma de Grasas Totales Original")
fat_4 = ((filtered_fat + 1)^G$Lambda - 1) / G$Lambda
hist(fat_4, col= 0, main="Histograma de Grasas Totales transformada
exacta con lambda de Yeo-Johnson")
fat_5 = sqrt(filtered_fat)
hist(fat_5, col= 0, main="Histograma de Grasas Totales aproximada")
```








```
filtered_model_df = data.frame(
  Estadistico = c(names(summary(filtered_fat)), "Curtosis", "Sesgo", "P-
Value"),
  Original = c(as.numeric(summary(filtered_fat)), kurtosis(filtered_fat),
skewness(filtered_fat), ad.test(filtered_fat)$p.value),
  "Modelo Exacto" = c(as.numeric(summary(fat_4)), kurtosis(fat_4),
skewness(fat_4), ad.test(fat_4)$p.value),
  "Modelo Aproximado" = c(as.numeric(summary(fat_5)), kurtosis(fat_5),
skewness(fat_5), ad.test(fat_5)$p.value)
)

filtered_model_df
```

```
##   Estadistico     Original Modelo.Exacto Modelo.Aproximado
## 1        Min. 5.000000e-01    0.44864296        0.70710678
## 2     1st Qu. 8.000000e+00    3.95368160        2.82842712
## 3      Median 1.600000e+01    6.14928712        4.00000000
## 4        Mean 1.745498e+01    5.93692056        3.86067715
## 5     3rd Qu. 2.350000e+01    7.75770883        4.84740550
## 6        Max. 1.180000e+02   19.24532185       10.86278049
## 7    Curtosis 1.247634e+01    1.06025977        0.94580524
## 8       Sesgo 2.369224e+00    0.41676870        0.29776179
## 9     P-Value 1.530289e-10    0.08644269        0.07931096
```

Definir la mejor transformación para los datos

```
resultado_df = data.frame(
  Estadistico = datos$Estadistico,
```

```
  Original = as.numeric(filtered_model_df$Modelo.Exacto),
  Box_Cox_Exacto = as.numeric(datos$Modelo.Exacto),
  Yeo_Johnson = as.numeric(filtered_model_df$Original)
)
resultado_df
```

```
##   Estadistico    Original Box_Cox_Exacto  Yeo_Johnson
## 1        Min.  0.44864296   0.000000e+00 5.000000e-01
## 2     1st Qu.  3.95368160   1.468694e+00 8.000000e+00
## 3      Median  6.14928712   3.707104e+00 1.600000e+01
## 4        Mean  5.93692056   3.432516e+00 1.745498e+01
## 5     3rd Qu.  7.75770883   5.261814e+00 2.350000e+01
## 6        Max. 19.24532185   1.074325e+01 1.180000e+02
## 7    Curtosis  1.06025977  -8.519420e-01 1.247634e+01
## 8       Sesgo  0.41676870  -1.151632e-01 2.369224e+00
## 9     P-Value  0.08644269   1.380766e-14 1.530289e-10
```

Comparando los valores obtenidos de los modelos que generamos, podemos darnos una mejor idea de la normalidad que se genera en nuestros modelos. Entre los modelos de Box-Cox y el de Yeo-Johnson podemos ver que el segundo tiene un mayor valor de p, lo que significa que este modelo sigue más una distribuión normal que el primero. Viendo las medidas estadisticas, tambien podemos concluir que el modelo de Yeo-Johnson es mejor, ya que en este modelo la media se acerca más a la mediana que en el otro, además de contar con una mejor curtosis, puesto que se acerca más al valor deseado (3), sin embargo, el modelo de Box-Cox gana en que tiene un sesgo más cercano al 0 que el de Yeo-Johnson. Hablando sobre la economía del modelo, considero que el de Box-Cox es mejor generalmente, puesto que se necesita de menores calculos computaciones para poder obtener los resultados del modelo, sin embargo, el modelo de Yeo-Johnson es mejor manejando los datos que incluyen ceros o algún valor negativo. Debido a estas consideraciones, necesitariamos saber sobre los datos con los que vamos a trabajar para poder decidir que modelo es mejor. En este caso, considero que es mejor el modelo de Yeo-Johnson, además de que nos proporcionó con mejores resultados estadisticos.

Ventajas y desventajas de Box-Cox y Yeo-Johnson

Algunas ventajas que tiene el modelo de Box-Cox es que es muy eficaz para la transformación de datos para que se asemejen a una distribución normal. Es uno de los modelos más usados, por lo que existe bastante documentación sobre ella, y permite ajustar el lambda para poder encontrar la mejor transformación que normalice los datos. Sin embargo, este modelo se limite bastante, puesto que únicamente se puede implementar cuando existen datos positivos, además de que puede llegar a ser muy sensible con los valores que se encuentren a los extremos, lo que puede llegar a afectar la calidad de la transformación.

Yeo-Johnson igualmente cuenta con algunas ventajas. La principal de estas es la flexibildad que tiene con los datos. A diferencia de Box-Cox, Yeo-Johnson puede manejar datos incluyan valores negativos o ceros, este púede llegar a adaptarse a una gama más amplia de distribuciones, además de que es menos sensible a los datos

ruidosos. Sin embargo, tambien tiene sus limitaciones, y es que para lograr este modelo se necesitan de mayores calculos computacionales, por lo que, con datasets muy grandes, obtener este modelo puede llegar a ser tardado.

Diferencias entre transformación y escalamiento de datos

- Tienen diferentes objetivos. La transformación busca cambiar la forma en la que los datos se distribuyen (como nuestro caso que los transformamos para obtener una distribución normal), mientras que el escalamiento busca ajustar los valores de los datos a un rango en especifico.

- Hay un diferente impacto en los datos. Mientras que el escalamiento no cambia la distribución, sólo ajusta el rango, la transformación si que impacta directamente en ellos. Lo que puede alterar su forma, y su resumen estadistico (curtosis, sesgo, media)

- Diferentes aplicaciones. El escalamiento normalmente se usa en algoritmos de machine learning, los cuales pueden llegar a ser sensibles a la escala de los datos; mientas que la transformación se usa para corregir problema de asimetria o modelos como la regresion lineal.

Cuando utilizar cada uno

- Transformación: Se necesita usar cuando los datos no siguen la distribución necesaria para un análisis estadistico.

- Escalamiento: Se necesita usar cuando se van a utilizar métodos o algoritmos que son sensibles a la magnitud de las variables (escala).