trans

2024-08-16

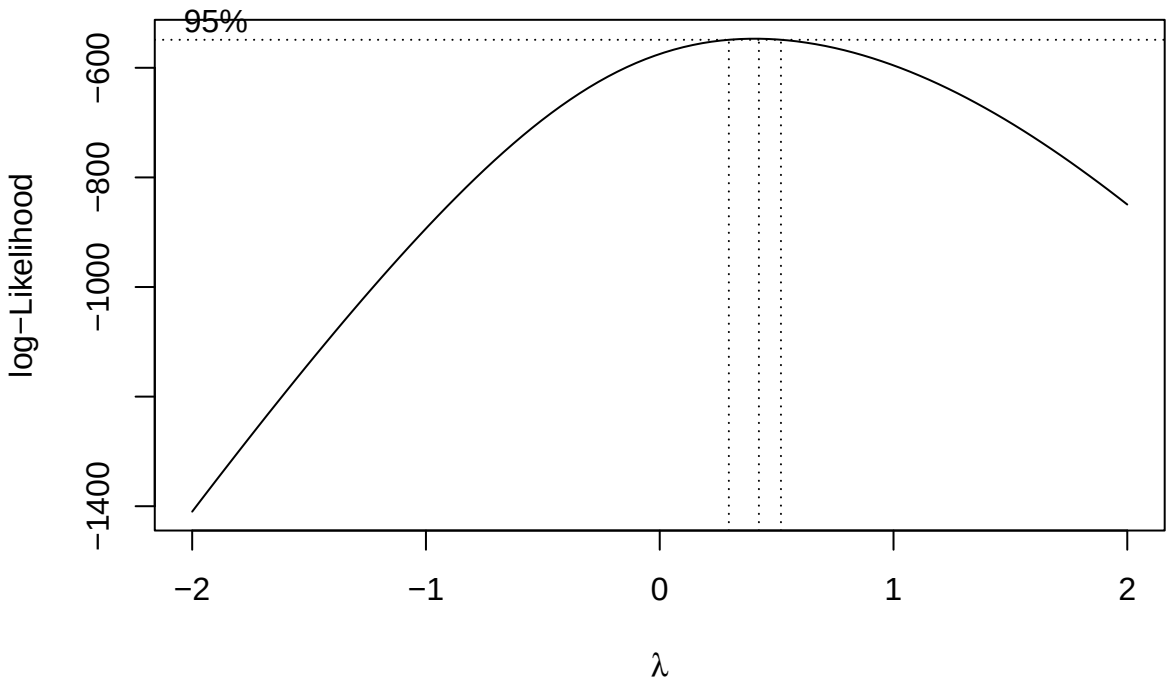
Cargar Datos

```
M1 = read.csv("downloads/mc-donalds-menu.csv")

Total_Fat = M1$Total.Fat
```

Utiliza la transformación Box-Cox. Utiliza el modelo exacto y el aproximado de acuerdo con las sugerencias de Box y Cox para la transformación

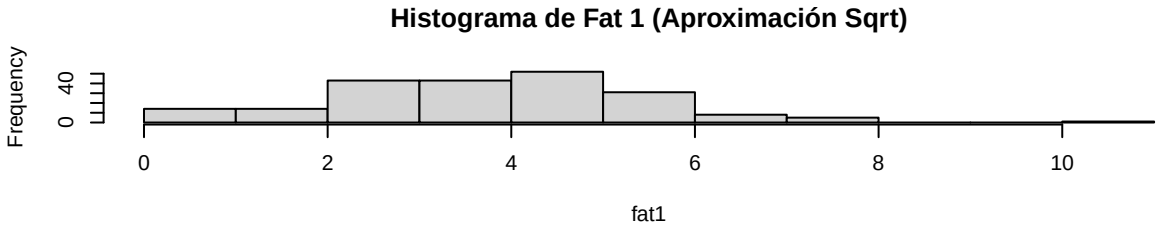
sqrt(X + 1) (Xˆlambda - 1) / lambda

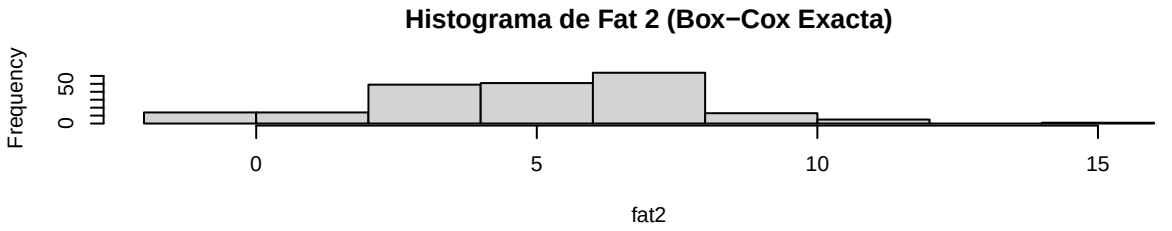
```
library(MASS)

M2 = subset(M1, Total.Fat > 0)
Total_FatP = M2$Total.Fat

bc = boxcox(Total_FatP ~ 1)
```

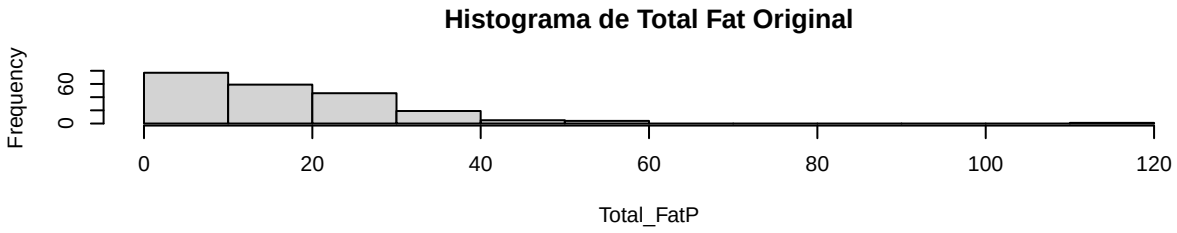



```
lambda_opt = bc$x[which.max(bc$y)]
```

```r
fat1 = sqrt(Total_FatP)
fat2 = (Total_FatP^lambda_opt - 1) / lambda_opt

par(mfrow=c(3,1))
hist(fat1, main="Histograma de Fat 1 (Aproximación Sqrt)")
hist(fat2, main="Histograma de Fat 2 (Box-Cox Exacta)")
hist(Total_FatP, main="Histograma de Total Fat Original")
```







Analiza la normalidad de las transformaciones obtenidas con los datos originales. Utiliza como argumento de normalidad:

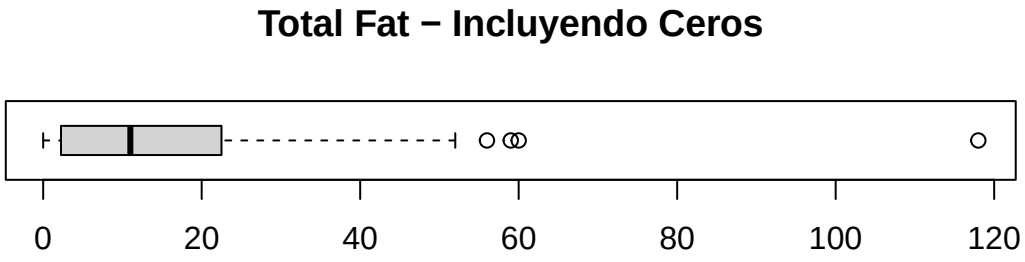
```r
library(e1071)
library(nortest)

kurtosis_original = kurtosis(Total_FatP)
skewness_original = skewness(Total_FatP)

kurtosis_fat1 = kurtosis(fat1)
skewness_fat1 = skewness(fat1)

kurtosis_fat2 = kurtosis(fat2)
skewness_fat2 = skewness(fat2)

D0 = ad.test(Total_FatP)
D1 = ad.test(fat1)
D2 = ad.test(fat2)
```

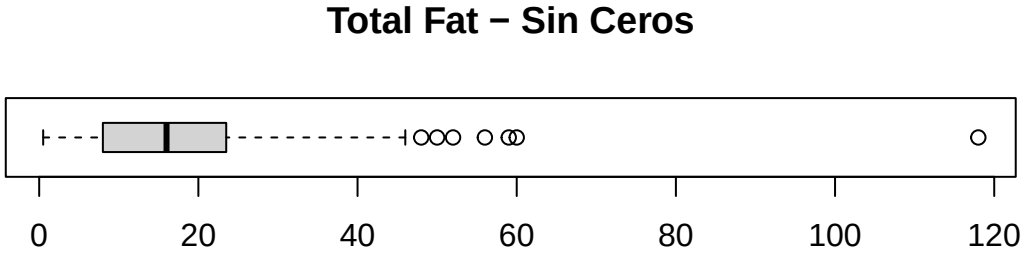
```
resumen_original = round(c(as.numeric(summary(Total_FatP)), kurtosis_original, skewness_original, D0$p.v
resumen_fat1 = round(c(as.numeric(summary(fat1)), kurtosis_fat1, skewness_fat1, D1$p.value), 3)
resumen_fat2 = round(c(as.numeric(summary(fat2)), kurtosis_fat2, skewness_fat2, D2$p.value), 3)

resumen = as.data.frame(rbind(resumen_original, resumen_fat1, resumen_fat2))
row.names(resumen) = c("Original", "Fat 1 (sqrt)", "Fat 2 (Box-Cox)")
colnames(resumen) = c("Mínimo", "Q1", "Mediana", "Media", "Q3", "Máximo", "Curtosis", "Sesgo", "Valor p

print(resumen)
```

```
##                 Mínimo    Q1 Mediana  Media     Q3  Máximo Curtosis Sesgo
## Original         0.500 8.000  16.000 17.455 23.500 118.000   12.476 2.369
## Fat 1 (sqrt)     0.707 2.828   4.000  3.861  4.847  10.863    0.946 0.298
## Fat 2 (Box-Cox) -0.601 3.338   5.285  4.970  6.638  15.482    0.523 0.052
##                 Valor p
## Original          0.000
## Fat 1 (sqrt)      0.079
## Fat 2 (Box-Cox)   0.022
```

Detecta anomalías y corrige tu base de datos (datos atípicos, ceros anámalos, etc).

```
par(mfrow=c(2,1))
boxplot(M1$Total.Fat, horizontal = TRUE, main="Total Fat - Incluyendo Ceros")
boxplot(Total_FatP, horizontal = TRUE, main="Total Fat - Sin Ceros")
```





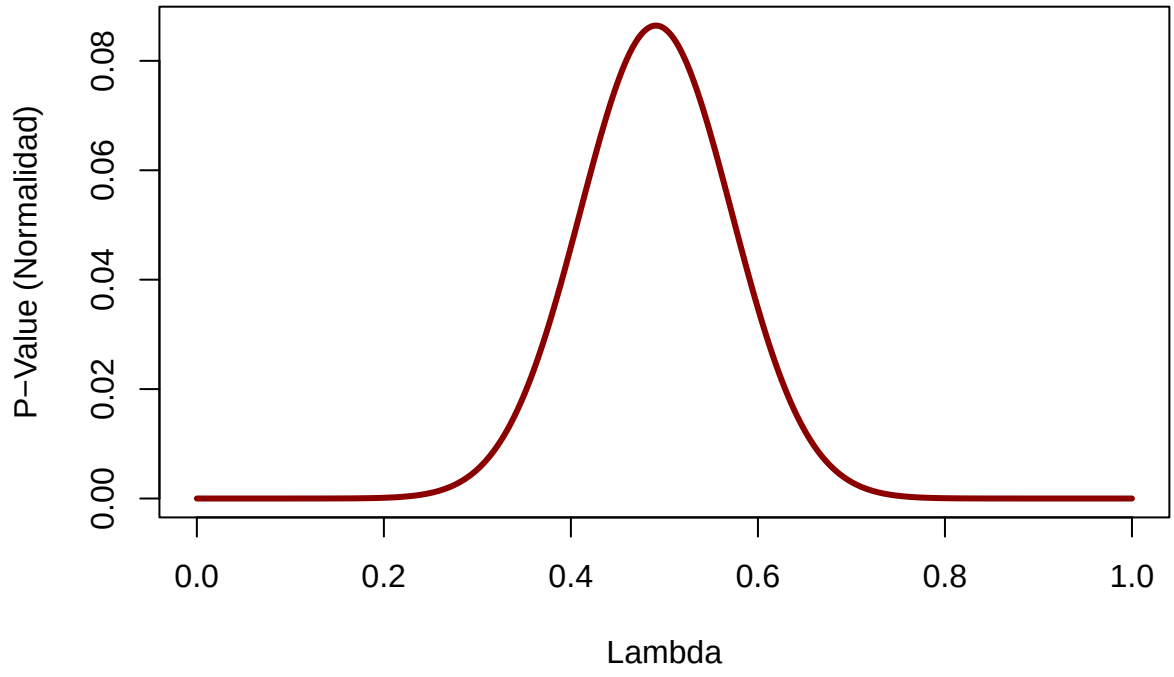

```
library(VGAM)
```

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

```r
library(nortest)

lp = seq(0, 1, 0.001)
nlp = length(lp)
n = length(M2$Total.Fat)

D = matrix(as.numeric(NA), ncol = 2, nrow = nlp)
d = NA

for (i in 1:nlp) {
  d = yeo.johnson(M2$Total.Fat, lambda = lp[i])
  p = ad.test(d)
  D[i,] = c(lp[i], p$p.value)
}

N = as.data.frame(D)
colnames(N) = c("Lambda", "P_Value")

G = data.frame(subset(N, N$P_Value == max(N$P_Value, na.rm = TRUE)))
G
```

```
##     Lambda    P_Value
## 492  0.491 0.08644269
```

```r
plot(N$Lambda, N$P_Value, type = "l",
     col = "darkred", lwd = 3,
     xlab = "Lambda",
     ylab = "P-Value (Normalidad)")
```



Utiliza la transformación de Yeo Johnson y encuentra el valor de lambda que maximiza el valor p de la prueba de normalidad que hayas utilizado (Anderson-Darling o Jarque Bera).

(X + 1)^lambda - 1) / lambda log(X + 1) -((-X + 1)^(2 - lambda) - 1) -log(-X + 1)

```
library(VGAM)
library(car)
```

```
## Loading required package: carData
```

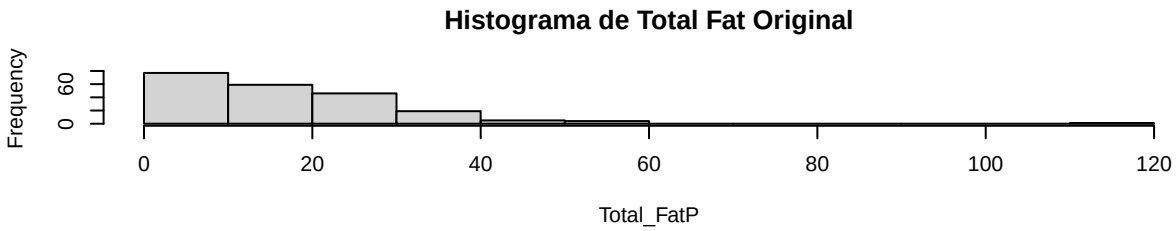
```
##
## Attaching package: 'car'
```

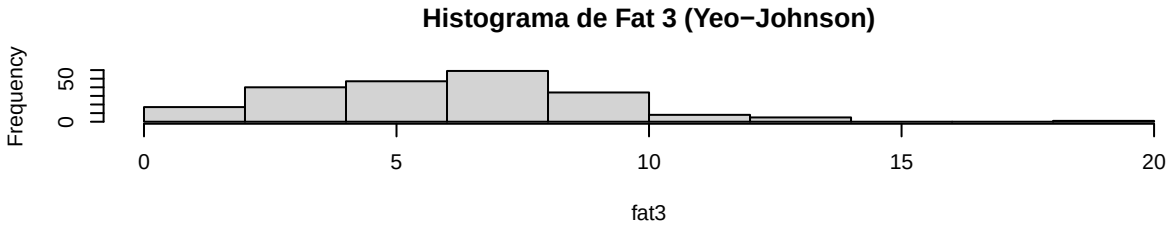
```
## The following object is masked from 'package:VGAM':
##
##     logit
```

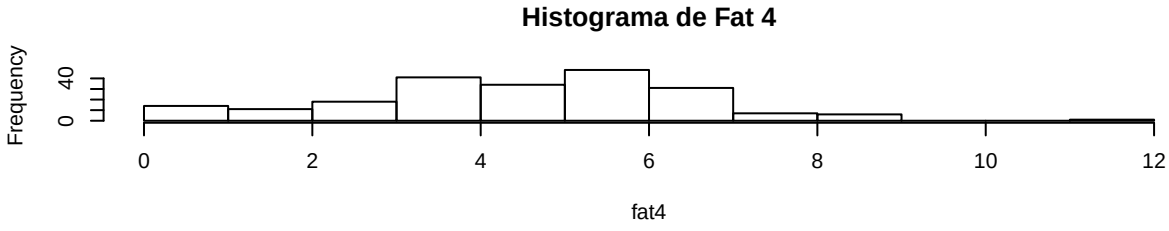
```
p_values = sapply(lp, function(lam) {
  transformed_data <- yeo.johnson(Total_FatP, lambda = lam)
  ad.test(transformed_data)$p.value
})

lambda_opt = lp[which.max(p_values)]

yj = powerTransform(M2$Total.Fat, family = "yjPower")
yj$lambda
```

```
## M2$Total.Fat
##    0.3355056
```

```
fat3 = yeo.johnson(Total_FatP, lambda = lambda_opt)

l4 = yj$lambda
fat4 = ((M2$Total.Fat+1)^l4-1)/l4

par(mfrow=c(3,1))
hist(Total_FatP, main="Histograma de Total Fat Original")
hist(fat3, main="Histograma de Fat 3 (Yeo-Johnson)")
hist(fat4,col=0,main="Histograma de Fat 4")
```








```
summary(fat4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4343  3.2489  4.7306  4.4745  5.7361 11.8329
```

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

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

```
kurtosis(fat4)
```

```
## [1] 0.2411101
```

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

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

```
skewness(fat4)
```

```
## [1] -0.005188732
```

Define la mejor transformación de los datos de acuerdo a las características de los modelos que encontraste. Toma en cuenta los criterios del inciso anterior para analizar normalidad y la economía del modelo.

En la normalidad seleccionamos el que tenga el valor p mas alto porque un valor mas alto sugiera que la distribucion transformada esta mas cerca de la normal.

Tambien se tiene que considerar la curtosis ya que la mejor transformacion deberia de acercar al sesgo de 0 y

curtosis de 3

Concluye sobre las ventajas y desventajes de los modelos de Box Cox y de Yeo Johnson.

BoxCox es adecuada para datos positivos como en este caso, tambien puede normalizar datos que no siguen una distribucion normal, solo no puede manejar datos con valores negativos.

Con Yeo-Jhonson si puedes manejar datos negativos y tambien es mas flexible, en casos como estos que todos los valores son positivos puede ser menos efectiva.

Analiza las diferencias entre la transformación y el escalamiento de los datos:

La transformacion cambia la distribucion de los datos, tambien este se usa para hacer que los datos cumplan con la normalidad para ciertos modelos. Algunos ejemplos es el boxcox y yeo jhonson como lo usamos aqui.

El escalamiento cambia la escala de los datos sin alterar su dsitribucion, este es necesario en casos donde sea sensible la escala de los datos, algunos ejemplos son normalizacion y estandarizacion.

La transformacion se utiliza cuando necesitas normalizar los datos o ajustar su distribucion. El escalamiento se usa cuando utilizas modelos de aprendizaje automatico que son sensibles a la magnitud de las variables.