5. Transformaciones

Facundo Colasurdo Caldironi

2024-08-14

R Markdown

Selecciona una variable, que no sea Calorías, y encuentra la mejor transformación de datos posible para que la variable seleccionada se comporte como una distribución Normal. Realiza:

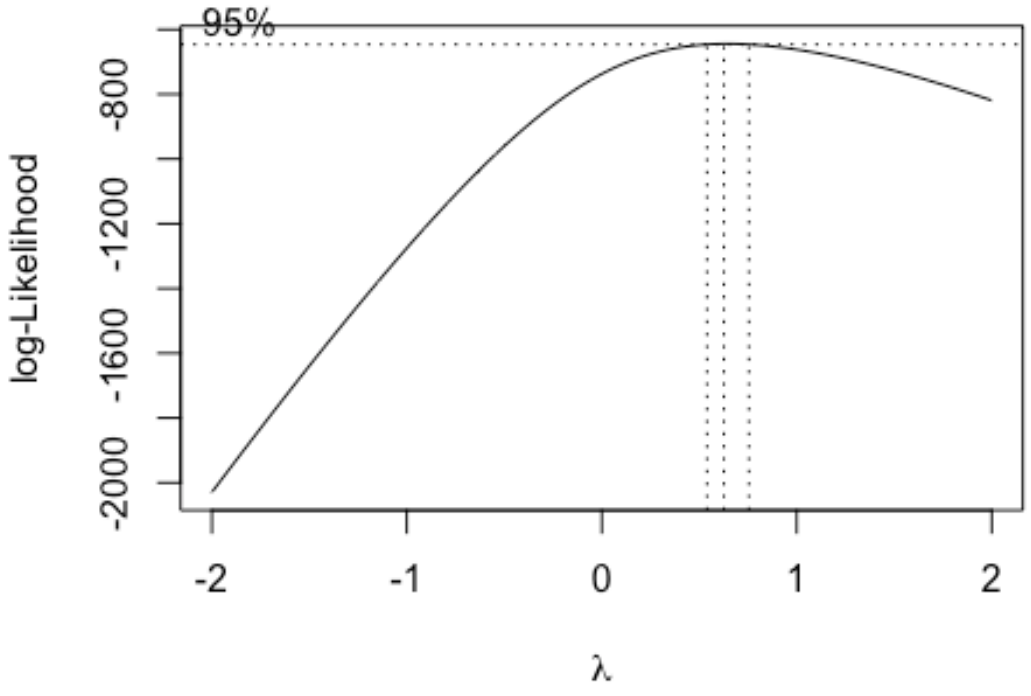
```
M=read.csv("file:///Users/facundocolasurdocaldironi/Downloads/mc-donalds-menu.csv") #leer la base de datos
M$Carbohydrates
```

```
##   [1]  31  30  29  30  30  31  38  43  36  42  34  39  36  42  34
40  41  46
##  [19]  38  48  47  44  48  46  57  55  56  51  56  50  55 111 116
110 115  60
##  [37]  61  26  15  66  58  49  47  41  48  46  45  42  32  33  35
51  34  35
##  [55]  34  35  44  55  42  58  44  57  43  65  51  43  40  43  41
40  56  42
##  [73]  56  42  68  55  61  47  12  18  30  59 118  39  10  22   8
20  42  28
##  [91]  37  30  34  27  32  25  30  44  67  15   4   4  30  32  21
22   7  53
## [109]  60  49  39  55  76  28   0   0   0   0  37  53  72  27   0
0   0   0
## [127]  37  54  74  27  12  23  21  34  44  65   0   0   0   0   0
36  45  54
## [145]  27   0   0   0  15  18  24  40  50  62  40  50  62  38  48
60  24  29
## [163]  37  15  19  25  41  51  63  40  51  63  39  49  60  24  30
38  49  60
## [181]  72  49  60  73  45  55  66  45  56  67  50  61  73  50  61
74  23  31
## [199]  47  22  29  43  21  29  43  20  27  41   9  12  18  41  50
70  41  50
## [217]  71  38  46  65  38  47  65  65  80  98  64  79  96  76  91
111  50  62
```

```
## [235]  79  47  58  74  50  61  78  86 109 135  90 114 140  91 114
141 109 135
## [253]  96 139  64  80 106  53 114  57
```

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

```
library(MASS)

carbo <- M$Carbohydrates
bc <- boxcox((carbo + 1) ~ 1)
```



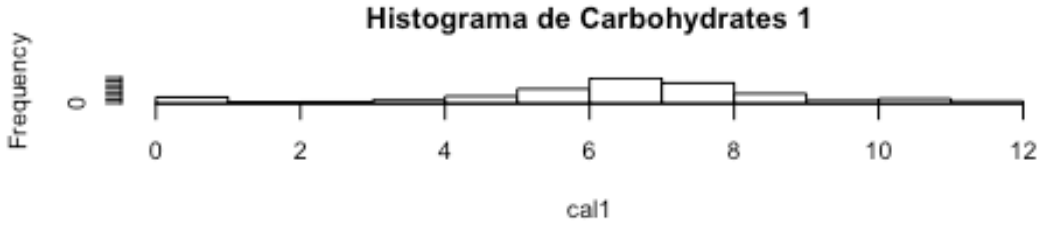

```
l <- bc$x[which.max(bc$y)]
print(l)

## [1] 0.6262626
```

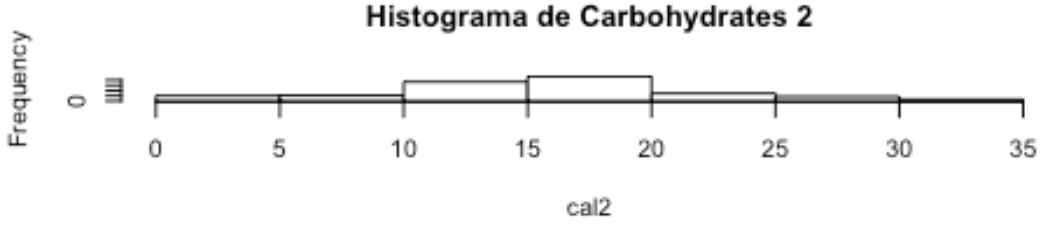
Escribe las ecuaciones de los modelos encontrados.

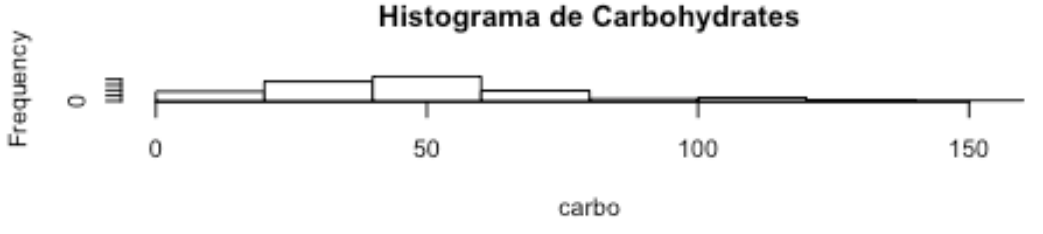
$\sqrt[x]$

$$\frac{(x+1)^{0.6262626}-1}{0.6262626}$$

```
cal1=sqrt(carbo)
cal2=((carbo+1)^l-1)/l
par(mfrow=c(3,1))
hist(cal1,col=0,main="Histograma de Carbohydrates 1")
hist(cal2,col=0,main="Histograma de Carbohydrates 2")
hist(carbo,col=0,main="Histograma de Carbohydrates")
```








```
library(nortest)
carbo <- M$Carbohydrates
```

```r
D=ad.test(carbo)
D$p.value
```

```
## [1] 2.546548e-10
```

```r
D=ad.test(cal1)
D$p.value
```

```
## [1] 2.795534e-15
```

```r
D=ad.test(cal2)
D$p.value
```

```
## [1] 8.1823e-08
```

```r
library(e1071)
summary(M$Carbohydrates)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   30.00   44.00   47.35   60.00  141.00
```

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

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

```r
kurtosis(M$Carbohydrates)
```

```
## [1] 1.324083
```

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

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

```r
skewness(M$Carbohydrates)
```

```
## [1] 0.9021952
```

```r
print("Curtosis, transformacion 1")
```

```
## [1] "Curtosis, transformacion 1"
```

```r
kurtosis(cal1)
```

```
## [1] 1.429156
```

```r
print("Sesgo, transformacion 1")
```

```
## [1] "Sesgo, transformacion 1"
```

```
skewness(cal1)
```

```
## [1] -0.7769342
```

```
print("Curtosis, transformacion 2")
```

```
## [1] "Curtosis, transformacion 2"
```

```
kurtosis(cal2)
```

```
## [1] 0.6381974
```

```
print("Sesgo, transformacion 2")
```

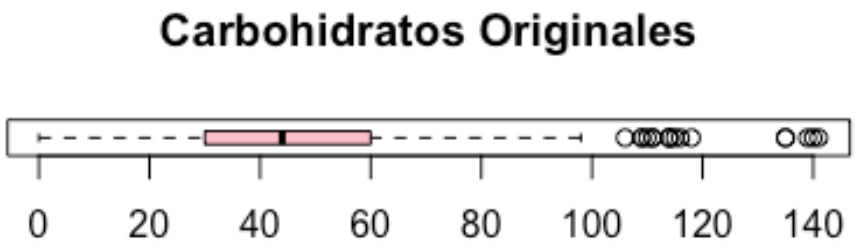
```
## [1] "Sesgo, transformacion 2"
```

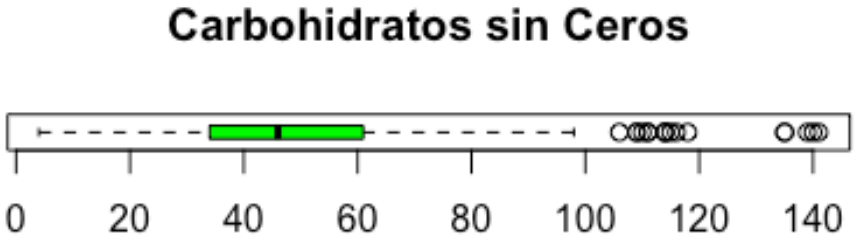
```
skewness(cal2)
```

```
## [1] -0.08250202
```

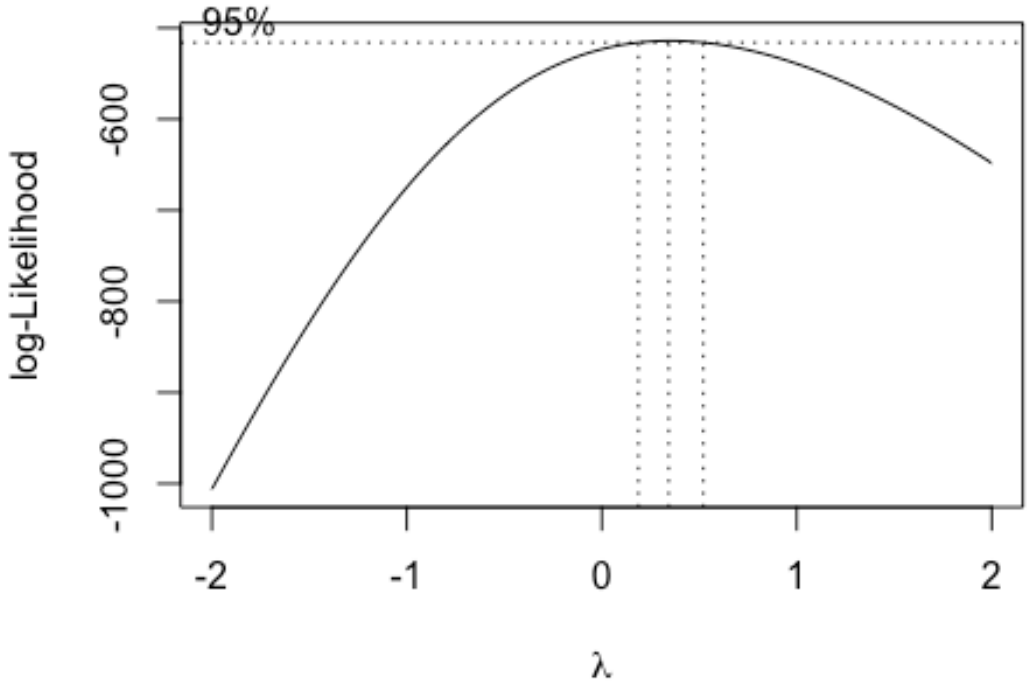
Detecta anomalías y corrige tu base de datos (datos atípicos, ceros anámalos, etc).

```
M2 <- subset(M, Carbohydrates > 0)
par(mfrow=c(2,1))
boxplot(M$Carbohydrates, horizontal = TRUE, col = "pink", main =
"Carbohidratos Originales")
boxplot(M2$Carbohydrates, horizontal = TRUE, col = "green", main =
"Carbohidratos sin Ceros")
```






```r
library(MASS)

M2 <- subset(M, Carbohydrates > 0)
M2_Carbohydrates <- M2$Carbohydrates
bc <- boxcox((M2_Carbohydrates + 1) ~ 1)
```



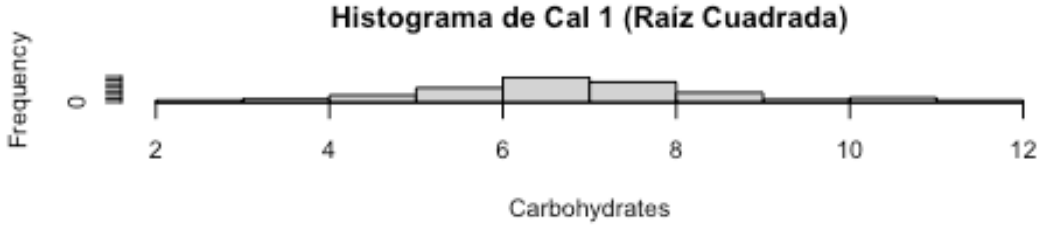

```
l <- bc$x[which.max(bc$y)]



print(l)

## [1] 0.3434343
```

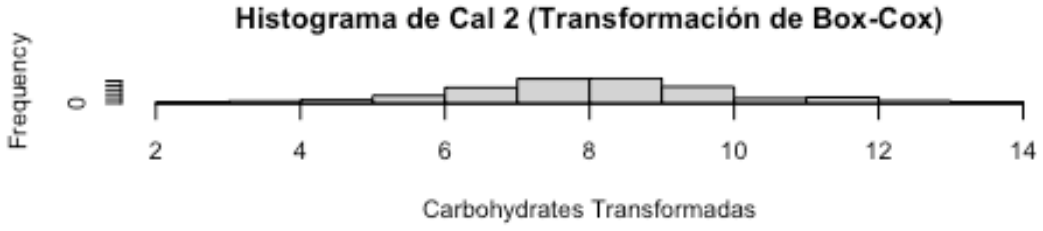
$\sqrt[x]$

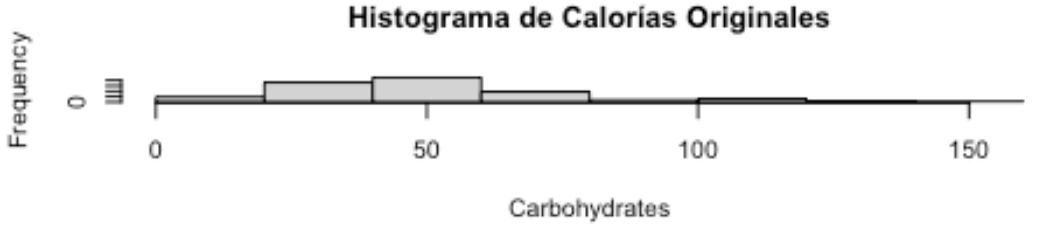
$$\frac{(x+1)^{0.3434343}-1}{0.3434343}$$

```
Carbohydrates <- M2$Carbohydrates
cal1 <- sqrt(Carbohydrates)
cal2 <- ((Carbohydrates + 1)^l - 1) / l
```

```r
par(mfrow=c(3,1))
hist(cal1, main="Histograma de Cal 1 (Raíz Cuadrada)",
xlab="Carbohydrates")
hist(cal2, main="Histograma de Cal 2 (Transformación de Box-Cox)",
xlab="Carbohydrates Transformadas")
hist(Carbohydrates, main="Histograma de Calorías Originales",
xlab="Carbohydrates")
```








```r
library(nortest)
# Pruebas de normalidad
D0 <- ad.test(Carbohydrates)
print(D0)
```

```
##
##  Anderson-Darling normality test
```

```
##
## data:  Carbohydrates
## A = 5.9462, p-value = 1.149e-14

D1 <- ad.test(cal1)
print(D1)

##
##  Anderson-Darling normality test
##
## data:  cal1
## A = 1.7716, p-value = 0.0001518

D2 <- ad.test(cal2)
print(D2)

##
##  Anderson-Darling normality test
##
## data:  cal2
## A = 1.4145, p-value = 0.001149

# Estadísticas descriptivas
m0 <- round(c(as.numeric(summary(Carbohydrates)),
kurtosis(Carbohydrates), skewness(Carbohydrates), D0$p.value), 3)
m1 <- round(c(as.numeric(summary(cal1)), kurtosis(cal1),
skewness(cal1), D1$p.value), 3)
m2 <- round(c(as.numeric(summary(cal2)), kurtosis(cal2),
skewness(cal2), D2$p.value), 3)

# Crear un data frame con las estadísticas descriptivas
m <- as.data.frame(rbind(m0, m1, m2))
row.names(m) <- c("Original", "Raíz Cuadrada", "Transformación de Box-
Cox")
names(m) <- c("Mínimo", "Q1", "Mediana", "Media", "Q3", "Máximo",
"Curtosis", "Sesgo", "Valor p")

# Imprimir el data frame
print(m)

##                            Mínimo     Q1 Mediana  Media     Q3
Máximo Curtosis
## Original                    4.000 34.000  46.000 50.451 61.000
```

```
141.000    1.763
## Raíz Cuadrada               2.000  5.831   6.782  6.871  7.810
11.874    0.602
## Transformación de Box-Cox  2.149  6.961   8.013  8.036  9.103
13.059    0.629
##                           Sesgo Valor p
## Original                  1.214   0.000
## Raíz Cuadrada             0.318   0.000
## Transformación de Box-Cox 0.026   0.001
```

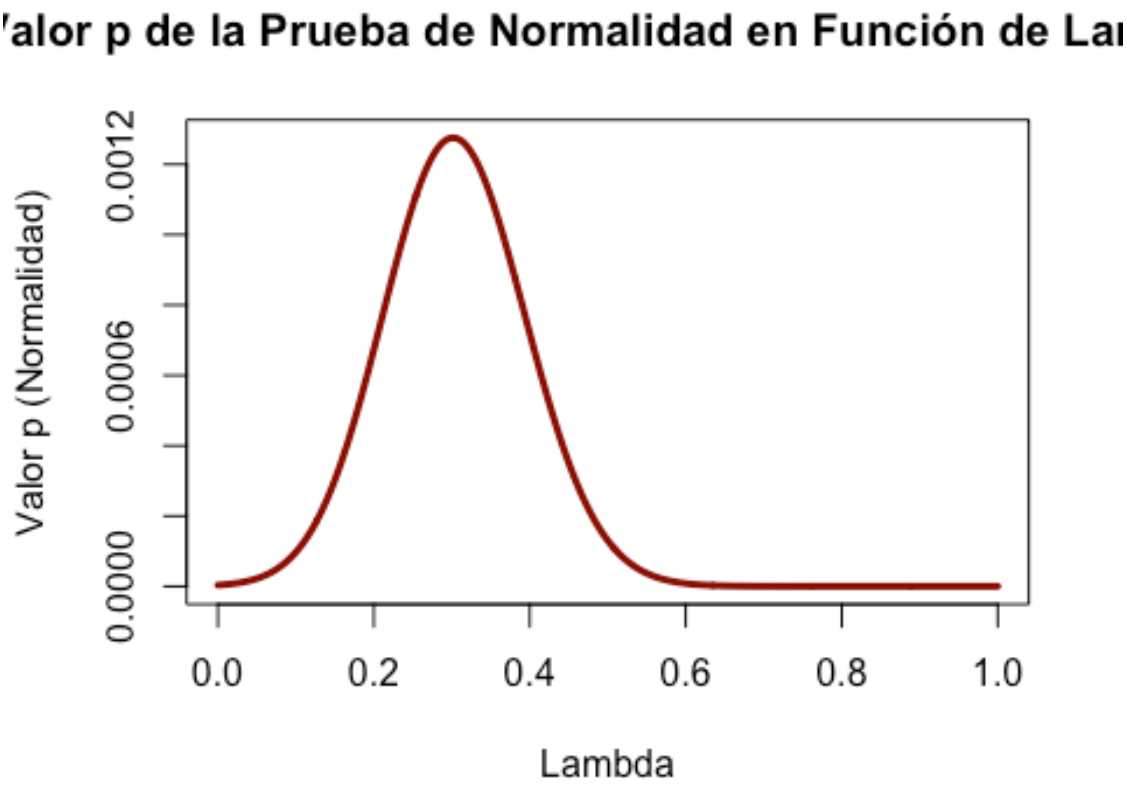
Transformación Yeo Johnson

```
library(VGAM)

## Loading required package: stats4

## Loading required package: splines

cal3<- yeo.johnson(M2$Carbohydrates, lambda = l)
print(cal3)

##   [1]  6.661862  6.558042  6.451999  6.558042  6.558042  6.661862
7.334900
##   [8]  7.768312  7.151308  7.684321  6.961079  7.424383  7.151308
7.684321
##  [15]  6.961079  7.512409  7.599037  8.013000  7.334900  8.170478
8.092278
##  [22]  7.851059  8.170478  8.013000  8.831207  8.690535  8.761276
8.398969
##  [29]  8.761276  8.323790  8.690535 11.808924 12.031391 11.763652
11.987404
##  [36]  9.036363  9.103273  6.119233  4.633812  9.427608  8.900351
8.247637
##  [43]  8.092278  7.599037  8.170478  8.013000  7.932608  7.684321
6.763573
##  [50]  6.863280  7.057061  8.398969  6.961079  7.057061  6.961079
7.057061
##  [57]  7.851059  8.690535  7.684321  8.900351  7.851059  8.831207
7.768312
##  [64]  9.364045  8.398969  7.768312  7.512409  7.768312  7.599037
7.512409
##  [71]  8.761276  7.684321  8.761276  7.684321  9.552889  8.690535
9.103273
```

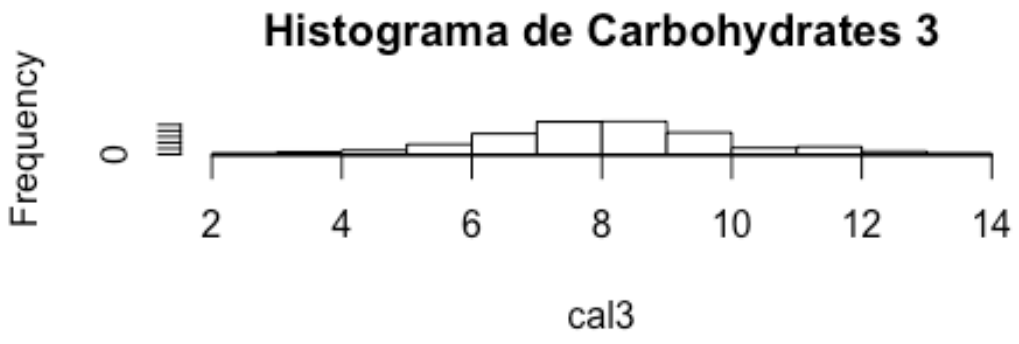
```
##  [78]  8.092278  4.114470  5.092550  6.558042  8.968729 12.118630
7.424383
##  [85]  3.722706  5.635367  3.280877  5.372459  7.684321  6.343609
7.243897
##  [92]  6.558042  6.961079  6.232737  6.763573  6.002935  6.558042
7.851059
##  [99]  9.490551  4.633812  2.148889  2.148889  6.558042  6.763573
5.505875
## [106]  5.635367  3.035378  8.546525  9.036363  8.247637  7.424383
8.690535
## [113] 10.031445  6.343609  7.243897  8.546525  9.796473  6.232737
7.243897
## [120]  8.618960  9.914988  6.232737  4.114470  5.761213  5.505875
6.961079
## [127]  7.851059  9.364045  7.151308  7.932608  8.618960  6.232737
4.633812
## [134]  5.092550  5.883661  7.512409  8.323790  9.169478  7.512409
8.323790
## [141]  9.169478  7.334900  8.170478  9.036363  5.883661  6.451999
7.243897
## [148]  4.633812  5.234803  6.002935  7.599037  8.398969  9.234997
7.512409
## [155]  8.398969  9.234997  7.424383  8.247637  9.036363  5.883661
6.558042
## [162]  7.334900  8.247637  9.036363  9.796473  8.247637  9.036363
9.855993
## [169]  7.932608  8.690535  9.427608  7.932608  8.761276  9.490551
8.323790
## [176]  9.103273  9.855993  8.323790  9.103273  9.914988  5.761213
6.661862
## [183]  8.092278  5.635367  6.451999  7.768312  5.505875  6.451999
7.768312
## [190]  5.372459  6.232737  7.599037  3.509057  4.114470  5.092550
7.599037
## [197]  8.323790  9.675807  7.599037  8.323790  9.736415  7.334900
8.013000
## [204]  9.364045  7.334900  8.092278  9.364045  9.364045 10.258533
11.198200
## [211]  9.299847 10.202464 11.099648 10.031445 10.847286 11.808924
8.323790
## [218]  9.169478 10.202464  8.092278  8.900351  9.914988  8.323790
9.103273
```

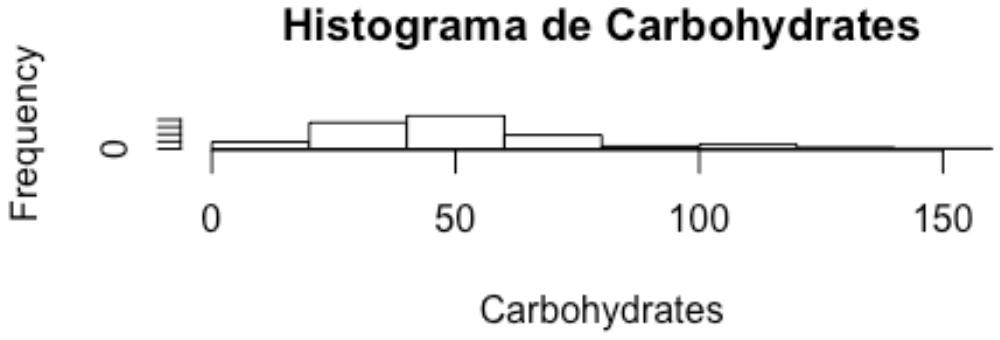
```
## [225] 10.145932 10.585750 11.718111 12.823962 10.795739 11.943168
13.020295
## [232] 10.847286 11.943168 13.059011 11.718111 12.823962 11.099648
12.981399
## [239]  9.299847 10.258533 11.579836  8.546525 11.943168  8.831207
```

```r
library(VGAM)
lp <- seq(0,1,0.001) # Valores de lambda propuestos
nlp <- length(lp)
n=length(M2[ ,1])
D <- matrix(as.numeric(NA),ncol=2,nrow=nlp)
d <-NA
for (i in 1:nlp){
d= yeo.johnson(M2$Carbohydrates, lambda = lp[i])
p=ad.test(d)
D[i,]=c(lp[i],p$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)",
     main="Valor p de la Prueba de Normalidad en Función de Lambda")
```




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

##     Lambda     Valor.p
## 303  0.302 0.001275547

par(mfrow=c(2,1))
hist(cal3,col=0,main="Histograma de Carbohydrates 3")
hist(Carbohydrates,col=0,main="Histograma de
Carbohydrates",xlab="Carbohydrates")
```






```
library(nortest)
summary(cal3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.149   6.961   8.013   8.036   9.103  13.059
```

```
print("Curtosis 3")
```

```
## [1] "Curtosis 3"
```

```
print(kurtosis(cal3))
```

```
## [1] 0.6287967
```

```
print("Sesgo 3")
```

```
## [1] "Sesgo 3"
```

```r
print(skewness(cal3))
```

```
## [1] 0.0257253
```

```r
variable <- M2$Carbohydrates  # Reemplaza con tu variable real
summary(variable)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.00   34.00   46.00   50.45   61.00  141.00
```

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

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

```r
print(kurtosis(variable))
```

```
## [1] 1.763188
```

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

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

```r
print(skewness(variable))
```

```
## [1] 1.214372
```

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.

La mejor de las opciones para este caso, es Yeo-Johnson, debido a que puede ser usada cuando existen ceros en los datos, en el caso de este modelo, se puede ver que en los carbohidratos existen valores 0 del mismo, razón suficiente para utilizar esta transformación aún cuando es más compleja que la de box Cox

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

Box Cox Ventajas: Es muy fácil realizar las transformaciones dentro de los datos. Los resultados son muy fáciles de interpretar Desventajas: Solo puede ser usada en números positivos, ni negativos ni ceros. Si existen valores atípicos, estos pueden afectar fuertemente este modelo.

Yeo Johnson. Ventajas: Puede ser usada en datos que contengan valores negativos y ceros Es bueno cuando se tiene una gran cantidad de datos disponibles dentro del documento.

Desventajas: Mucho más complejo en comparación de la de Box Cox, por lo que se requiere más cálculos del mismo. Puede ser más confuso a la hora de interpretar los resultados

Escribe al menos 3 diferencias entre lo que es la transformación y el escalamiento de los datos 1.-La transformación se utiliza cuando se quiere cambiar la distribución de algunos datos, mientras que el escalamiento se utiliza para ajustarlos a un rango específico

2.-La transformación utiliza funciones matemáticas para cambiar la forma de los datos, mientras que el escalamiento cambia su media o rango, más no cambia su forma.

3.-Cambia la forma de los datos, para que sea más o menos simétrica dependiendo de lo que se busque, por otra parte, el escalamiento mantiene la forma de la misma, pero cambia los valores dentro de los datos para que estén a una escala diferente.

Indica cuándo es necesario utilizar cada uno Se usa la transformación cuando los datos que se tienen no siguen ninguna distribución normal, a su vez que se busca estabilizar su varianza, un ejemplo de esto seria cuando se aplica una regresión lineal en los datos.

Se usa el escalamiento cuando los datos son importantes pero se requieren en una escala mayor o menor para ser usadas en el machine learning, un ejemplo de esto sería en distintos modelos de machine learning.