transformaciones_entrega

AUTHOR
Alfredo García

Leyendo los datos

```
M <- read.csv('mc-donalds-menu.csv')
head(M)
```

```
   Category                           Item   Serving.Size Calories
1 Breakfast                    Egg McMuffin 4.8 oz (136 g)      300
2 Breakfast                Egg White Delight 4.8 oz (135 g)      250
3 Breakfast                 Sausage McMuffin 3.9 oz (111 g)      370
4 Breakfast        Sausage McMuffin with Egg 5.7 oz (161 g)      450
5 Breakfast Sausage McMuffin with Egg Whites 5.7 oz (161 g)      400
6 Breakfast             Steak & Egg McMuffin 6.5 oz (185 g)      430
  Calories.from.Fat Total.Fat Total.Fat....Daily.Value. Saturated.Fat
1               120        13                        20             5
2                70         8                        12             3
3               200        23                        35             8
4               250        28                        43            10
5               210        23                        35             8
6               210        23                        36             9
  Saturated.Fat....Daily.Value. Trans.Fat Cholesterol
1                            25         0         260
2                            15         0          25
3                            42         0          45
4                            52         0         285
5                            42         0          50
6                            46         1         300
  Cholesterol....Daily.Value. Sodium Sodium....Daily.Value. Carbohydrates
1                          87    750                     31            31
2                           8    770                     32            30
3                          15    780                     33            29
4                          95    860                     36            30
5                          16    880                     37            30
6                         100    960                     40            31
  Carbohydrates....Daily.Value. Dietary.Fiber Dietary.Fiber....Daily.Value.
1                            10             4                             17
2                            10             4                             17
3                            10             4                             17
4                            10             4                             17
5                            10             4                             17
6                            10             4                             18
  Sugars Protein Vitamin.A....Daily.Value. Vitamin.C....Daily.Value.
1      3      17                        10                         0
```

```
2      3      18                        6                        0
3      2      14                        8                        0
4      2      21                       15                        0
5      2      21                        6                        0
6      3      26                       15                        2
  Calcium....Daily.Value. Iron....Daily.Value.
1                      25                    15
2                      25                     8
3                      25                    10
4                      30                    15
5                      25                    10
6                      30                    20
```

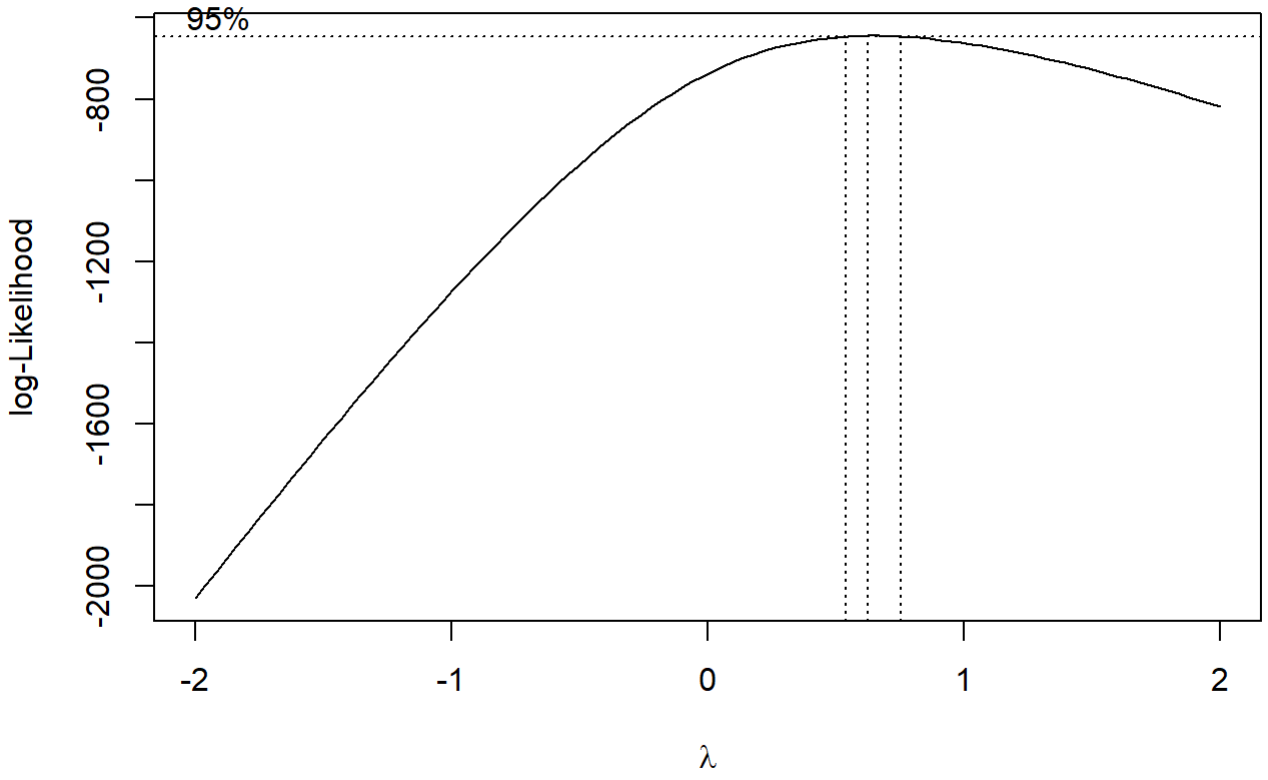
La variable seleccionada será Carbohydrates

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)
```

```
Warning: package 'MASS' was built under R version 4.2.2
```

```
# Add 1 to all values in M$Carbohydrates
variable <- M$Carbohydrates + 1

bc <- boxcox(variable ~ 1)
```




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

```
[1] 0.6262626
```

Como el valor de lambda es mas cercano a 0.5, tomaremos ese valor de la tabla y al escribir las ecuaciones haremos el ajuste de haber sumado 1 para evitar el problema con los 0s.

Escribe las ecuaciones de los modelos encontrados.

Modelo aproximado

$$\sqrt{x+1}$$

Modelo exacto

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

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

1. Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis.

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

```
Warning: package 'e1071' was built under R version 4.2.2
```

```
variable <- M$Carbohydrates + 1

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

car1=sqrt(variable)
car2=((variable)^l-1)/l


#summary(variable)
#print("Curtosis")
#kurtosis(variable)
#print("Sesgo")
#skewness(variable)

print("Original")
```

```
[1] "Original"
```

```
summary(M$Carbohydrates)
```

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

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

```
[1] 1.324083
```

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

```
[1] 0.9021952
```

```
D=ad.test(M$Carbohydrates)
D$p.value
```

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

```
print("Aproximada")
```

```
[1] "Aproximada"
```

```
summary(car1)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   5.568   6.708   6.583   7.810  11.916 
```

```
kurtosis(car1)
```

```
[1] 0.90923
```

```
skewness(car1)
```

```
[1] -0.4939626
```

```
D2=ad.test(car1)
D2$p.value
```

```
[1] 4.481723e-11
```

```
print("Exacta")
```

```
[1] "Exacta"
```

```
summary(car2)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00   12.12   15.72   15.67   19.36   33.98 
```

```
kurtosis(car2)
```

```
[1] 0.6381974
```

```
skewness(car2)
```

```
[1] -0.08250202
```

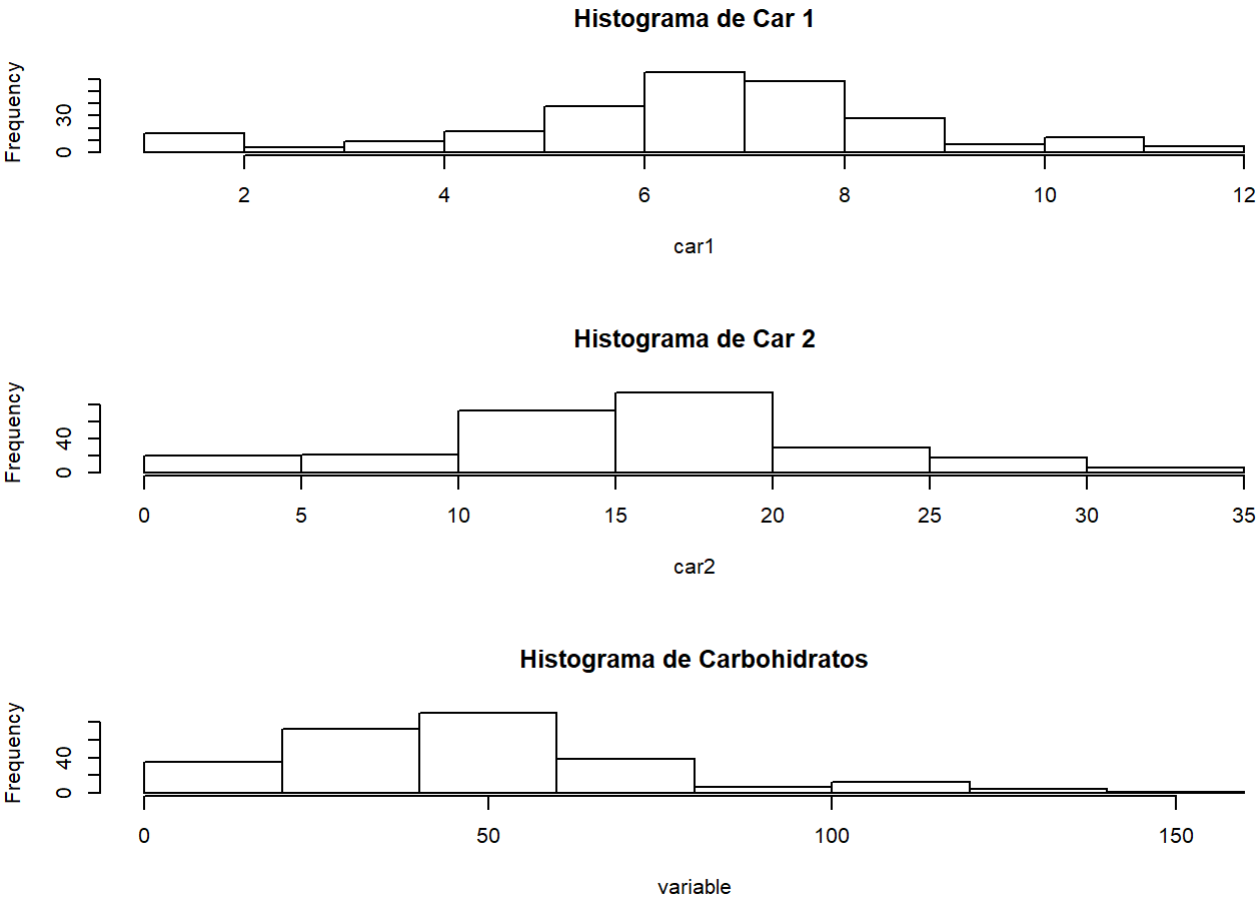
```
D3=ad.test(car2)
D3$p.value
```

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

2. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.

```
variable <- M$Carbohydrates

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

```
car1=sqrt(variable+1)
car2=((variable+1)^l-1)/l
par(mfrow=c(3,1))
hist(car1,col=0,main="Histograma de Car 1")
hist(car2,col=0,main="Histograma de Car 2")
hist(variable,col=0,main="Histograma de Carbohidratos")
```



3. Realiza la prueba de normalidad de Anderson-Darling o de Jarque Bera para los datos transformados y los originales

Aquí está el resumen

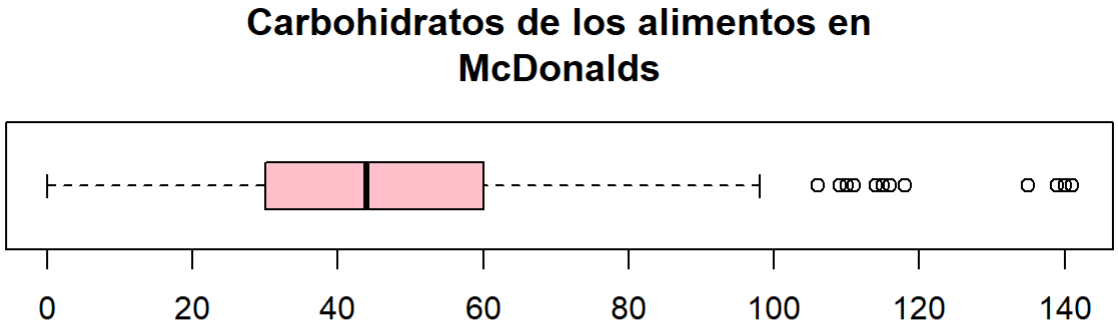
```
m0=round(c(as.numeric(summary(M$Carbohydrates)),kurtosis(M$Carbohydrates),skewness(M$Carbohydrate
m1=round(c(as.numeric(summary(car1)),kurtosis(car1),skewness(car1),D2$p.value),4)
m2=round(c(as.numeric(summary(car2)),kurtosis(car2),skewness(car2),D3$p.value),4)

m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Modelo Aprox","Modelo Exacto")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo","Valor p")
print(m)
```

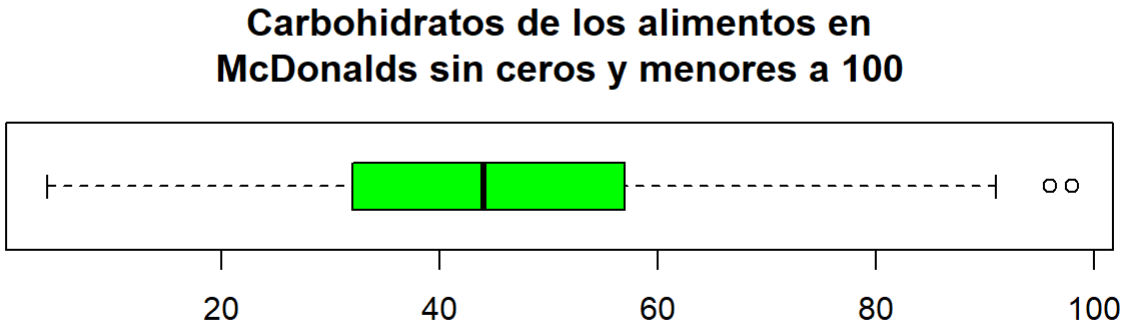
```
              Minimo      Q1 Mediana   Media      Q3   Máximo Curtosis   Sesgo
Original           0 30.0000 44.0000 47.3462 60.0000 141.0000   1.3241  0.9022
```

```
Modelo Aprox       1  5.5678  6.7082  6.5832  7.8102  11.9164   0.9092 -0.4940
Modelo Exacto      0 12.1192 15.7249 15.6688 19.3602  33.9779   0.6382 -0.0825
             Valor p
Original           0
Modelo Aprox       0
Modelo Exacto      0
```

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

```
#Quitando los alimentos que tienen cero Carbohydrates:
#M1=subset(M,M$Carbohydrates>0)
par(mfrow=c(2,1))
boxplot(M$Carbohydrates, horizontal = TRUE,col="pink", main="Carbohidratos de los alimentos en
McDonalds")
boxplot(M$Carbohydrates[M$Carbohydrates > 0 & M$Carbohydrates <= 100], horizontal = TRUE,col="gre
McDonalds sin ceros y menores a 100")
```

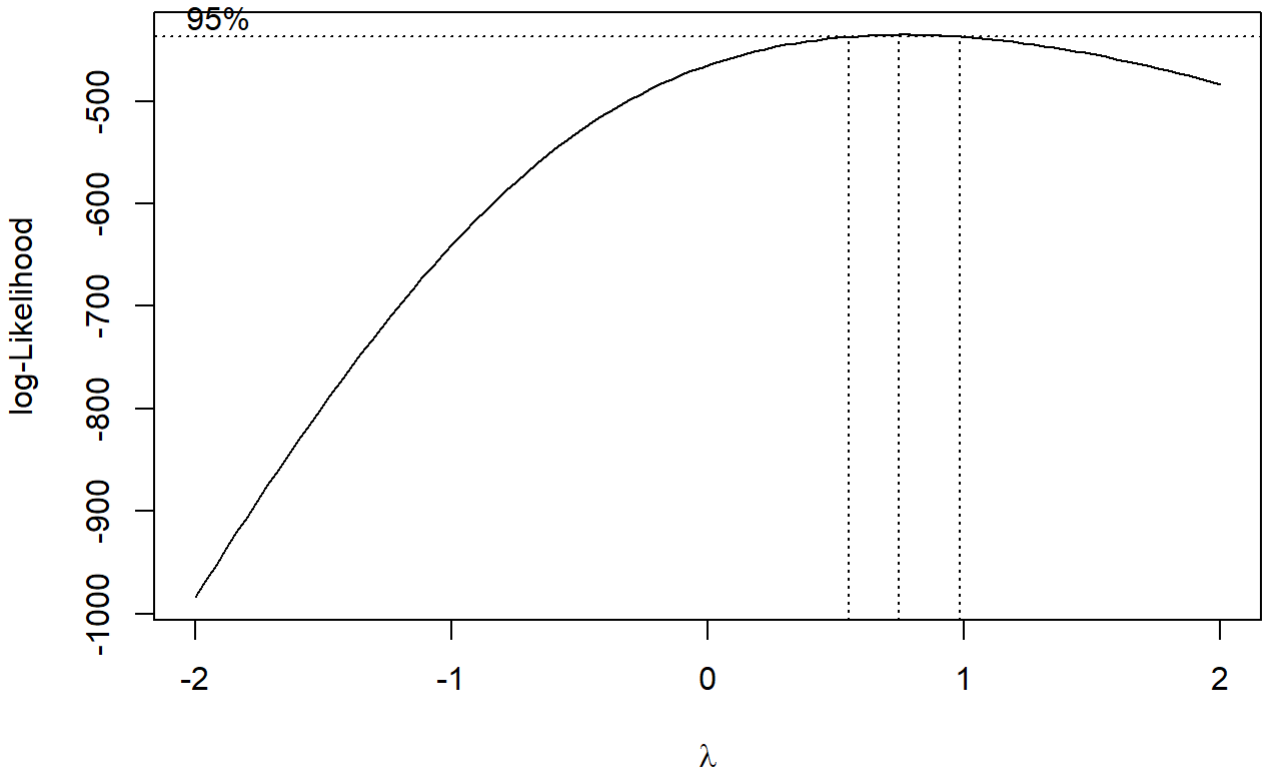




Entonces como podemos ver vamos a eliminar los 0 y a quitar los datos de carbohidratos mayores a 100.

```
library(MASS)
variable <- M$Carbohydrates[M$Carbohydrates > 0 & M$Carbohydrates <= 100]
```

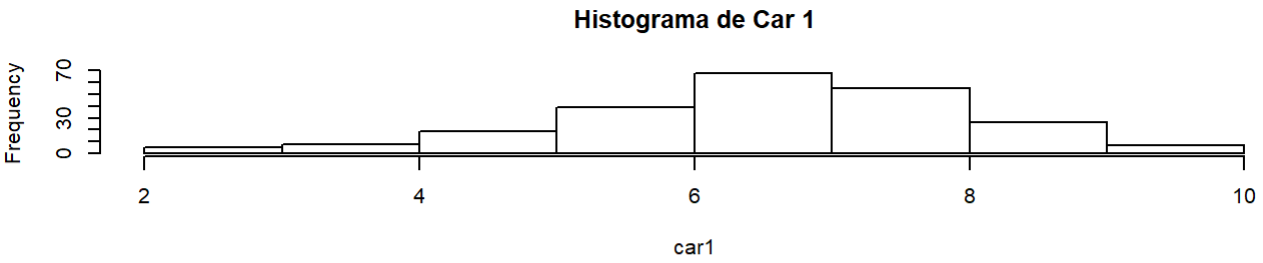
```
bc<-boxcox(variable~1)
```

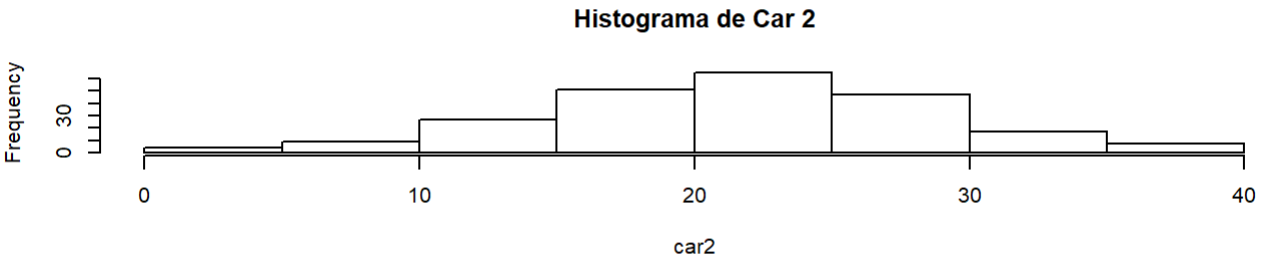


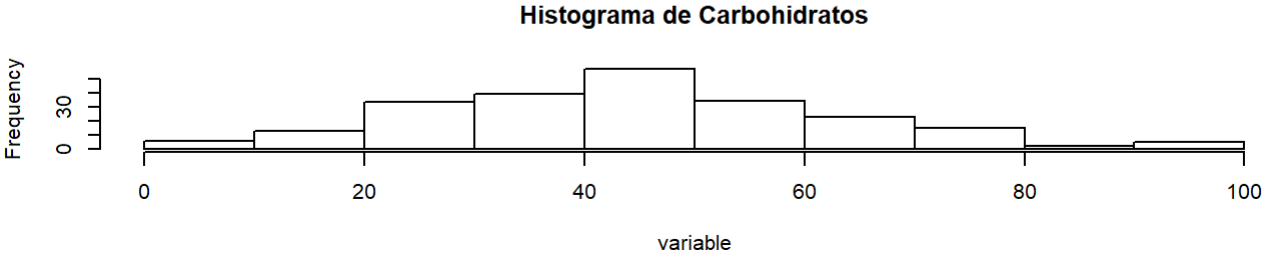

```
bc$x[which.max(bc$y)]
```

```
[1] 0.7474747
```

```
l <- 0.7474747

car1=sqrt(variable)
car2=(variable^l-1)/l
par(mfrow=c(3,1))
hist(car1,col=0,main="Histograma de Car 1")
hist(car2,col=0,main="Histograma de Car 2")
hist(variable,col=0,main="Histograma de Carbohidratos")
```








```r
variable <- M$Carbohydrates[M$Carbohydrates > 0 & M$Carbohydrates <= 100]
l <- 0.7474747
car1=sqrt(variable)
car2=(variable^l-1)/l

D=ad.test(variable)
D2=ad.test(car1)
D3=ad.test(car2)

m0=round(c(as.numeric(summary(M$Carbohydrates)),kurtosis(M$Carbohydrates),skewness(M$Carbohydrate
m1=round(c(as.numeric(summary(car1)),kurtosis(car1),skewness(car1),D2$p.value),4)
m2=round(c(as.numeric(summary(car2)),kurtosis(car2),skewness(car2),D3$p.value),4)

m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Modelo Aprox","Modelo Exacto")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo","Valor p")
print(m)
```

```
              Minimo      Q1 Mediana   Media      Q3   Máximo Curtosis   Sesgo
Original      0.0000 30.0000 44.0000 47.3462 60.0000 141.0000   1.3241  0.9022
Modelo Aprox  2.0000  5.6569  6.6332  6.5669  7.5498   9.8995   0.3649 -0.4208
Modelo Exacto 2.4329 16.5050 21.3004 21.3824 26.1332  39.8524  -0.0057 -0.0378
              Valor p
```

```
Original       0.3670
Modelo Aprox   0.0499
Modelo Exacto  0.8160
```

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).

```
library(VGAM)
```

```
Warning: package 'VGAM' was built under R version 4.2.3

Loading required package: stats4

Loading required package: splines
```

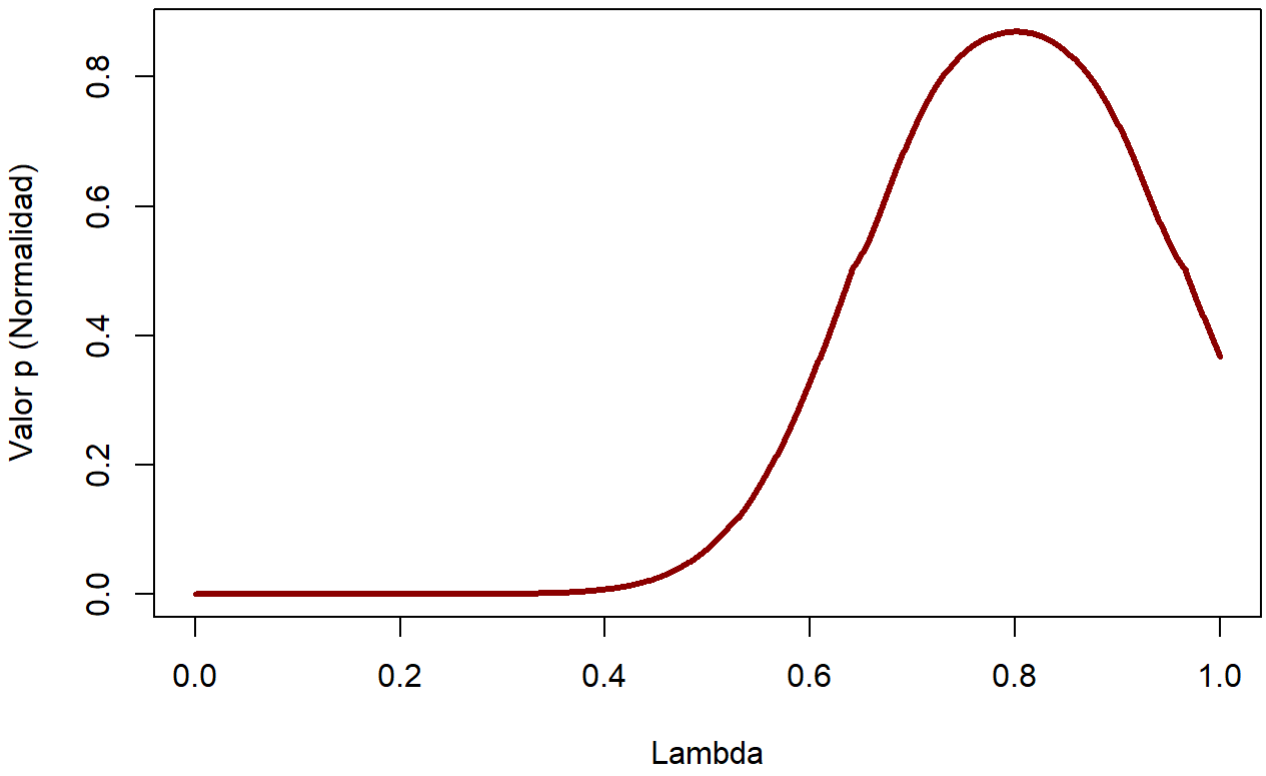
```
l <- 0.7474747
car3 <- yeo.johnson(M$Carbohydrates, lambda = l)
```

Para encontrar el valor de lambda que maximiza el valor p en la prueba de normalidad

```
variable <- M$Carbohydrates[M$Carbohydrates > 0 & M$Carbohydrates <= 100]
lp <- seq(0,1,0.001) # Valores de lambda propuestos
nlp <- length(lp)
n=length(variable)
D <- matrix(as.numeric(NA),ncol=2,nrow=nlp)
d <-NA
for (i in 1:nlp){
d= yeo.johnson(variable, lambda = lp[i])
p=ad.test(d)
D[i,]=c(lp[i],p$p.value)}
```

Grafica de lambda contra el valor p

```
N <- as.data.frame(D)
plot(N$V1,N$V2,
type="l",col="darkred",lwd=3,
xlab="Lambda",
ylab="Valor p (Normalidad)")
```



Valor de lambda que maximiza el valor p

```
G=data.frame(subset(N,N$V2==max(N$V2)))
G
```

```
       V1        V2
802 0.801 0.8695641
```

Escribe la ecuación del modelo encontrado.

Modelo exacto

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

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

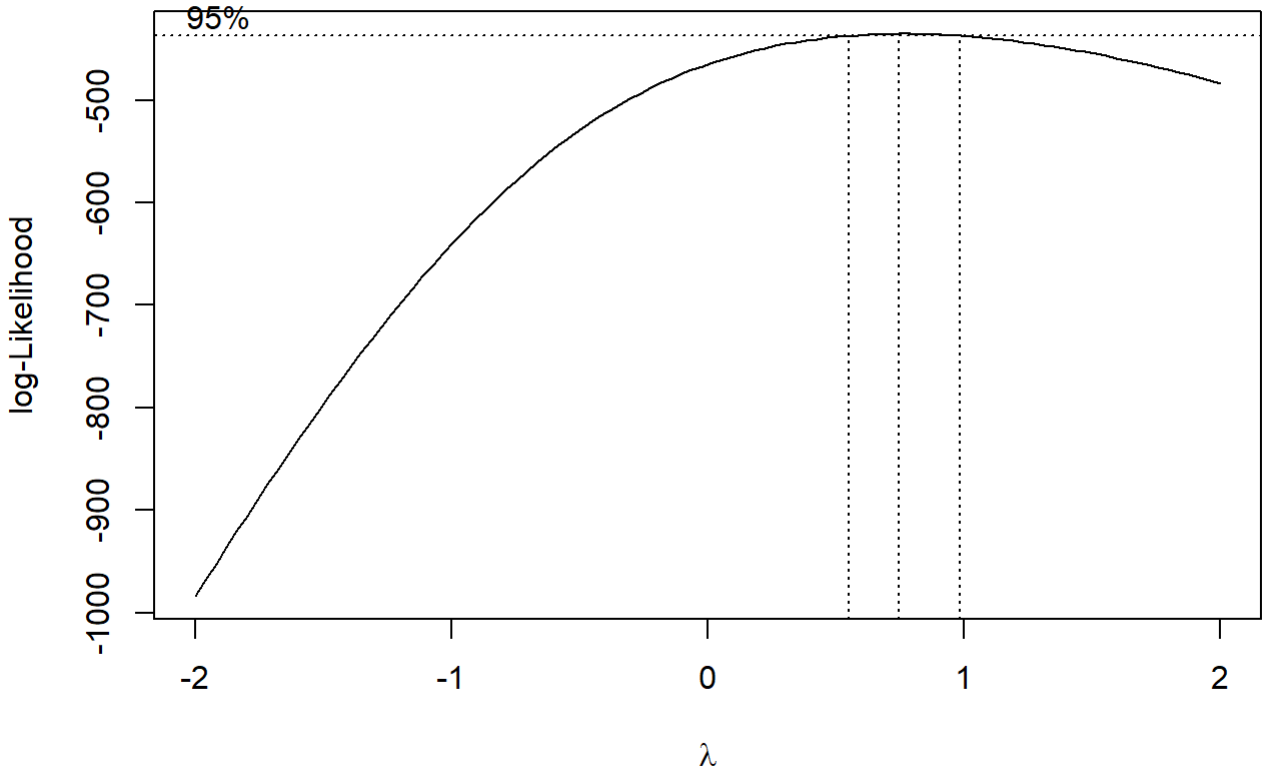
1. Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis.
2. Realiza la prueba de normalidad de Anderson-Darling para los datos transformados y los originales

```r
variable <- M$Carbohydrates[M$Carbohydrates > 0 & M$Carbohydrates <= 100]
l <- 0.801
car1=sqrt(variable)
car2=(variable^l-1)/l

D=ad.test(variable)
D2=ad.test(car1)
D3=ad.test(car2)

m0=round(c(as.numeric(summary(M$Carbohydrates)),kurtosis(M$Carbohydrates),skewness(M$Carbohydrate
m1=round(c(as.numeric(summary(car1)),kurtosis(car1),skewness(car1),D2$p.value),4)
m2=round(c(as.numeric(summary(car2)),kurtosis(car2),skewness(car2),D3$p.value),4)

m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Modelo Aprox","Modelo Exacto")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo","Valor p")
print(m)
```

```
              Minimo      Q1 Mediana   Media      Q3   Máximo Curtosis   Sesgo
Original      0.0000 30.0000 44.0000 47.3462 60.0000 141.0000   1.3241  0.9022
Modelo Aprox  2.0000  5.6569  6.6332  6.5669  7.5498   9.8995   0.3649 -0.4208
Modelo Exacto 2.5414 18.7959 24.6202 24.8270 30.5806  47.8807  -0.0277  0.0377
              Valor p
Original       0.3670
Modelo Aprox   0.0499
Modelo Exacto  0.8664
```

2. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.

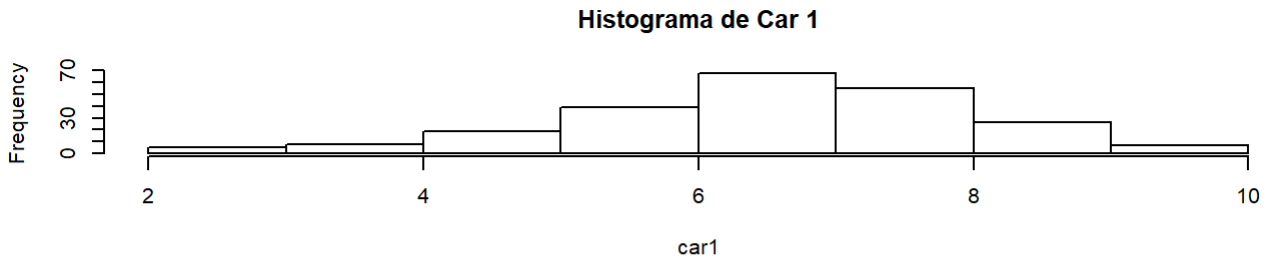
```r
library(MASS)
variable <- M$Carbohydrates[M$Carbohydrates > 0 & M$Carbohydrates <= 100]
bc<-boxcox(variable~1)
```

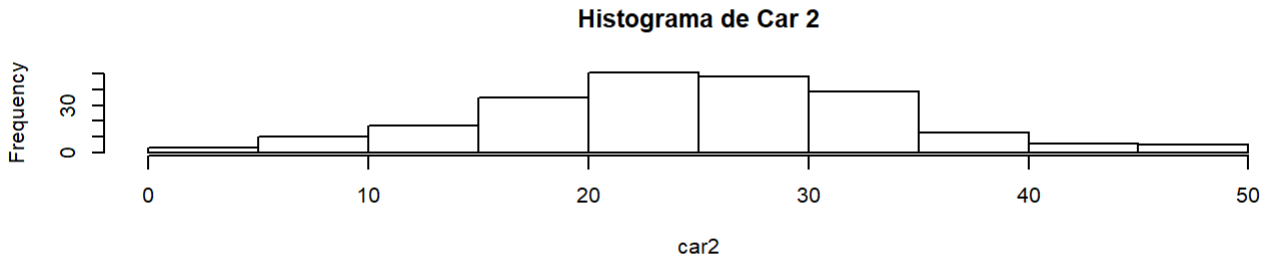


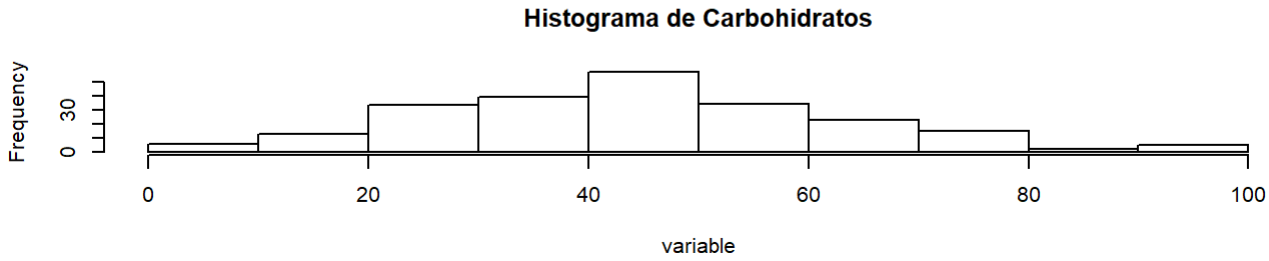

```
bc$x[which.max(bc$y)]
```

```
[1] 0.7474747
```

```
l <- 0.801

car1=sqrt(variable)
car2=(variable^l-1)/l
par(mfrow=c(3,1))
hist(car1,col=0,main="Histograma de Car 1")
hist(car2,col=0,main="Histograma de Car 2")
hist(variable,col=0,main="Histograma de Carbohidratos")
```







Define la mejor transformación de los datos de acuerdo a las características de los modelos que encontraste.

La mejor transformación de datos es la Trnasformación de Yeo Johnson que está dada por la formula: $$ ((x+1)^{0.801} - 1)

$$

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

Ventajas del modelo de Box-Cox: * Simplicidad * Interpretación * Aplicabilidad a datos positivos

Desventajas del modelo de Box-Cox: * Restricción a datos positivos * Sensibilidad a valores atípicos

Ventajas del modelo de Yeo-Johnson: * Flexibilidad * Robustez ante valores atípicos * Potencial para obtener distribuciones simétricas

Desventajas del modelo de Yeo-Johnson: * Complejidad * Interpretación

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

La transformación de datos implica aplicar una función matemática a los valores originales de una variable para cambiar su distribución o su relación con otras variables. Las transformaciones pueden ser útiles para corregir problemas en los datos, como falta de normalidad, asimetría o heterocedasticidad.

El escalado de datos implica cambiar la escala de los valores de una variable sin alterar su distribución o su relación con otras variables. El objetivo del escalado es asegurarse de que las diferentes variables estén en la misma escala, lo que puede ser importante para ciertos algoritmos de aprendizaje automático y análisis estadístico.

Escribe al menos 3 diferencias entre lo que es la transformación y el escalamiento de los datos. Indica cuándo es necesario utilizar cada uno

1. Naturaleza del Cambio:

- Transformación: Modifica la distribución y propiedades estadísticas de los valores de la variable.
- Escalamiento: Ajusta la magnitud de los valores para igualar la escala numérica de las variables.

2. Propósito:

- Transformación: Mejora la normalidad, simetría y heterocedasticidad de los datos para métodos estadísticos.
- Escalamiento: Garantiza que las variables estén en la misma escala para algoritmos de ML.

3. Diferentes Métodos:

- Transformación: Transformación logarítmica, raíz cuadrada, Box-Cox, diferencia.
- Escalamiento: Escalado estándar (z-score), min-max, escalado robusto.

Cuándo Utilizar Cada Uno:

Transformación de Datos: Utiliza transformaciones cuando las variables tengan distribuciones no normales, alta asimetría o varianza heterogénea para preparar los datos para métodos estadísticos.

Escalamiento de Datos: Utiliza el escalado cuando trabajes con algoritmos de aprendizaje automático que requieran variables en la misma escala, para evitar que la magnitud afecte el rendimiento.