3. Transformaciones

Diego Rodríguez A00829925

2023-08-21

Lectura de datos y limpieza

Primero hay que cargar algunas librerias, leer la base de datos y revisar las variables que contiene.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
M = read.csv("mc-donalds-menu-1.csv")
summary(M)
```

```
##    Category             Item           Serving.Size          Calories
##  Length:260         Length:260         Length:260         Min.   :   0.0
##  Class :character   Class :character   Class :character   1st Qu.: 210.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 340.0
##                                                           Mean   : 368.3
##                                                           3rd Qu.: 500.0
##                                                           Max.   :1880.0
##  Calories.from.Fat   Total.Fat       Total.Fat....Daily.Value. Saturated.Fat
##  Min.   :   0.0    Min.   :  0.000   Min.   :  0.00            Min.   : 0.000
```

```
##  1st Qu.:  20.0    1st Qu.:  2.375   1st Qu.:  3.75           1st Qu.: 1.000
##  Median : 100.0    Median : 11.000   Median : 17.00           Median : 5.000
##  Mean   : 127.1    Mean   : 14.165   Mean   : 21.82           Mean   : 6.008
##  3rd Qu.: 200.0    3rd Qu.: 22.250   3rd Qu.: 35.00           3rd Qu.:10.000
##  Max.   :1060.0    Max.   :118.000   Max.   :182.00           Max.   :20.000
##  Saturated.Fat....Daily.Value.   Trans.Fat       Cholesterol
##  Min.   :  0.00                Min.   :0.0000   Min.   :  0.00
##  1st Qu.:  4.75                1st Qu.:0.0000   1st Qu.:  5.00
##  Median : 24.00                Median :0.0000   Median : 35.00
##  Mean   : 29.97                Mean   :0.2038   Mean   : 54.94
##  3rd Qu.: 48.00                3rd Qu.:0.0000   3rd Qu.: 65.00
##  Max.   :102.00                Max.   :2.5000   Max.   :575.00
##  Cholesterol....Daily.Value.     Sodium       Sodium....Daily.Value.
##  Min.   :  0.00              Min.   :   0.0   Min.   :  0.00
##  1st Qu.:  2.00              1st Qu.: 107.5   1st Qu.:  4.75
##  Median : 11.00              Median : 190.0   Median :  8.00
##  Mean   : 18.39              Mean   : 495.8   Mean   : 20.68
##  3rd Qu.: 21.25              3rd Qu.: 865.0   3rd Qu.: 36.25
##  Max.   :192.00              Max.   :3600.0   Max.   :150.00
##  Carbohydrates    Carbohydrates....Daily.Value. Dietary.Fiber
##  Min.   :  0.00   Min.   : 0.00                 Min.   :0.000
##  1st Qu.: 30.00   1st Qu.:10.00                 1st Qu.:0.000
##  Median : 44.00   Median :15.00                 Median :1.000
##  Mean   : 47.35   Mean   :15.78                 Mean   :1.631
##  3rd Qu.: 60.00   3rd Qu.:20.00                 3rd Qu.:3.000
##  Max.   :141.00   Max.   :47.00                 Max.   :7.000
##  Dietary.Fiber....Daily.Value.     Sugars          Protein
##  Min.   : 0.000                Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 0.000                1st Qu.:  5.75   1st Qu.: 4.00
##  Median : 5.000                Median : 17.50   Median :12.00
##  Mean   : 6.531                Mean   : 29.42   Mean   :13.34
##  3rd Qu.:10.000                3rd Qu.: 48.00   3rd Qu.:19.00
##  Max.   :28.000                Max.   :128.00   Max.   :87.00
##  Vitamin.A....Daily.Value. Vitamin.C....Daily.Value. Calcium....Daily.Value.
##  Min.   :  0.00            Min.   :  0.000           Min.   : 0.00
##  1st Qu.:  2.00            1st Qu.:  0.000           1st Qu.: 6.00
##  Median :  8.00            Median :  0.000           Median :20.00
##  Mean   : 13.43            Mean   :  8.535           Mean   :20.97
##  3rd Qu.: 15.00            3rd Qu.:  4.000           3rd Qu.:30.00
##  Max.   :170.00            Max.   :240.000           Max.   :70.00
##  Iron....Daily.Value.
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 4.000
##  Mean   : 7.735
##  3rd Qu.:15.000
##  Max.   :40.000
```

Selección de una variable y transformación Box-Cox:

Selecciona una variable numérica y continua del conjunto de datos (que no sea "Calorías"). En este caso se ha seleccionado la variable "Carbohydrates". Aplica la transformación Box-Cox para encontrar el valor de lambda óptimo tanto para el modelo exacto como el aproximado.

Ecuaciones de los modelos encontrados:

Modelo exacto: data_transformada_exacta = (xˆlambda_exact - 1) / lambda_exact

Modelo aproximado: data_transformada_aproximada = log(x)

Para encontrar el mejor valor de lambda usaremos la función boxcox.lambda de la librería forecast:

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
variable <- M$Carbohydrates

lambda = BoxCox.lambda(variable)
```

```
## Warning in guerrero(x, lower, upper): Guerrero's method for selecting a Box-Cox
## parameter (lambda) is given for strictly positive data.
```

```
lambda
```

```
## [1] 0.5919362
```

Ahora aplicamos las fórmulas con el valor encontrado de lambda

```
carb1 = sqrt(variable+1)
carb2 = ((variable+1)^lambda-1)/lambda

data_transformada_aproximada <- carb1
data_transformada_exacta <- carb2

variable = variable[variable != 0]
data_transformada_aproximada = data_transformada_aproximada[data_transformada_aproximada != 0]
data_transformada_exacta = data_transformada_exacta[data_transformada_exacta != 0]
```

Análisis de normalidad:

Compara medidas como mínimo, máximo, media, mediana, cuartiles, sesgo y curtosis entre los datos originales y las transformaciones. También puedes graficar histogramas.

```
library(moments)

skewness_original <- skewness(variable)
kurtosis_original <- kurtosis(variable)
skewness_exacta <- skewness(data_transformada_exacta)
kurtosis_exacta <- kurtosis(data_transformada_exacta)
skewness_aproximada <- skewness(data_transformada_aproximada)
kurtosis_aproximada <- kurtosis(data_transformada_aproximada)

summary_a <- data.frame(
```

```
  Min = c(min(variable), min(data_transformada_exacta), min(data_transformada_aproximada)),
  Q1 = c(quantile(variable, 0.25), quantile(data_transformada_exacta, 0.25), quantile(data_transformada
  Median = c(median(variable), median(data_transformada_exacta), median(data_transformada_aproximada)),
  Mean = c(mean(variable), mean(data_transformada_exacta), mean(data_transformada_aproximada)),
  Q3 = c(quantile(variable, 0.75), quantile(data_transformada_exacta, 0.75), quantile(data_transformada
  Max = c(max(variable), max(data_transformada_exacta), max(data_transformada_aproximada)),
  Std_Dev = c(sd(variable), sd(data_transformada_exacta), sd(data_transformada_aproximada)),
  Skewness = c(skewness_original, skewness_exacta, skewness_aproximada),
  Kurtosis = c(kurtosis_original, kurtosis_exacta, kurtosis_aproximada)
)


rownames(summary_a) <- c("Original", "Box-Cox (Exacta)", "Box-Cox (Aproximada)")

print("Comparación de Medidas y Normalidad:")
```

```
## [1] "Comparación de Medidas y Normalidad:"
```

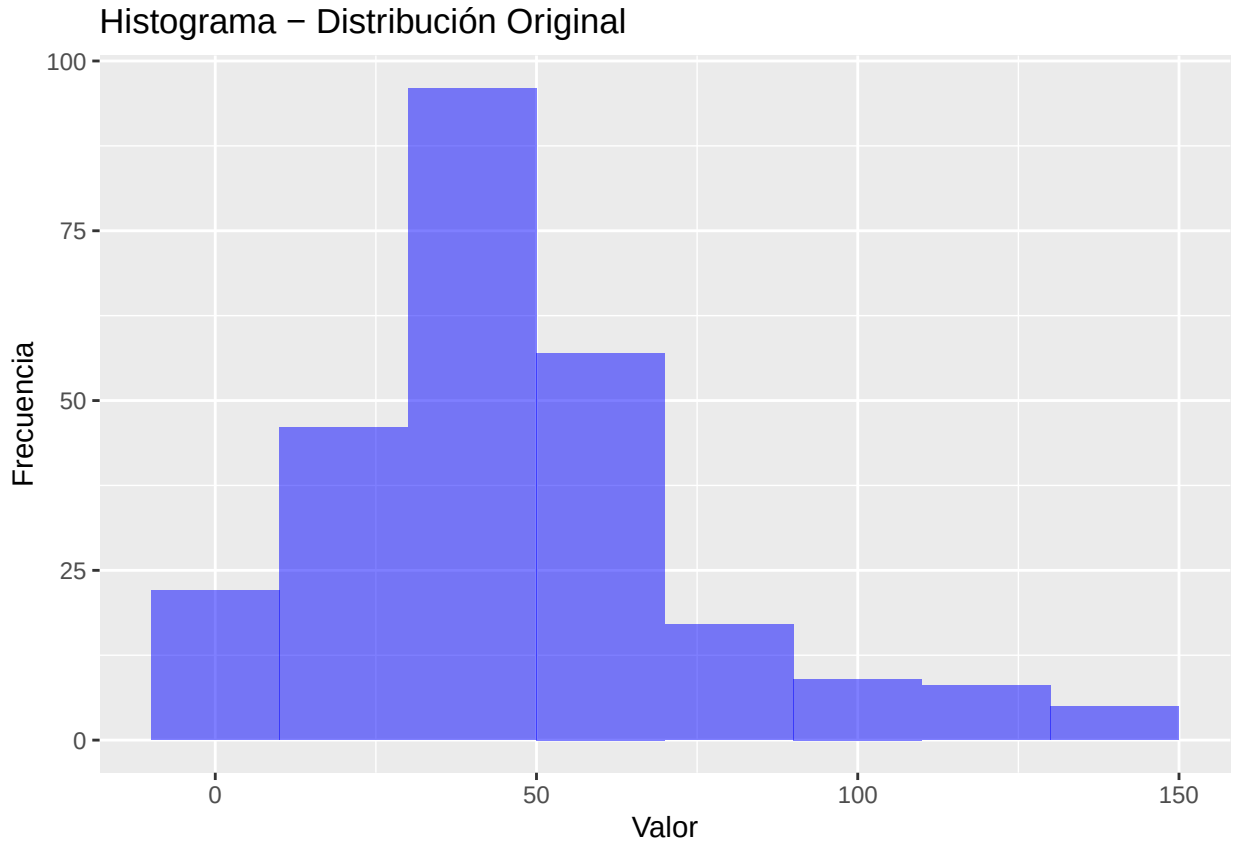
```
print(summary_a)
```

```
##                          Min        Q1    Median      Mean       Q3       Max
## Original             4.000000 34.000000 46.000000 50.450820 61.00000 141.00000
## Box-Cox (Exacta)     2.690595 12.169085 14.811252 15.198553 17.75109  30.06054
## Box-Cox (Aproximada) 1.000000  5.567764  6.708204  6.583244  7.81025  11.91638
##                        Std_Dev   Skewness Kurtosis
## Original             26.333756  1.2218756 4.802472
## Box-Cox (Exacta)      5.101460  0.5273609 3.711691
## Box-Cox (Aproximada)  2.241961 -0.4968262 3.939475
```

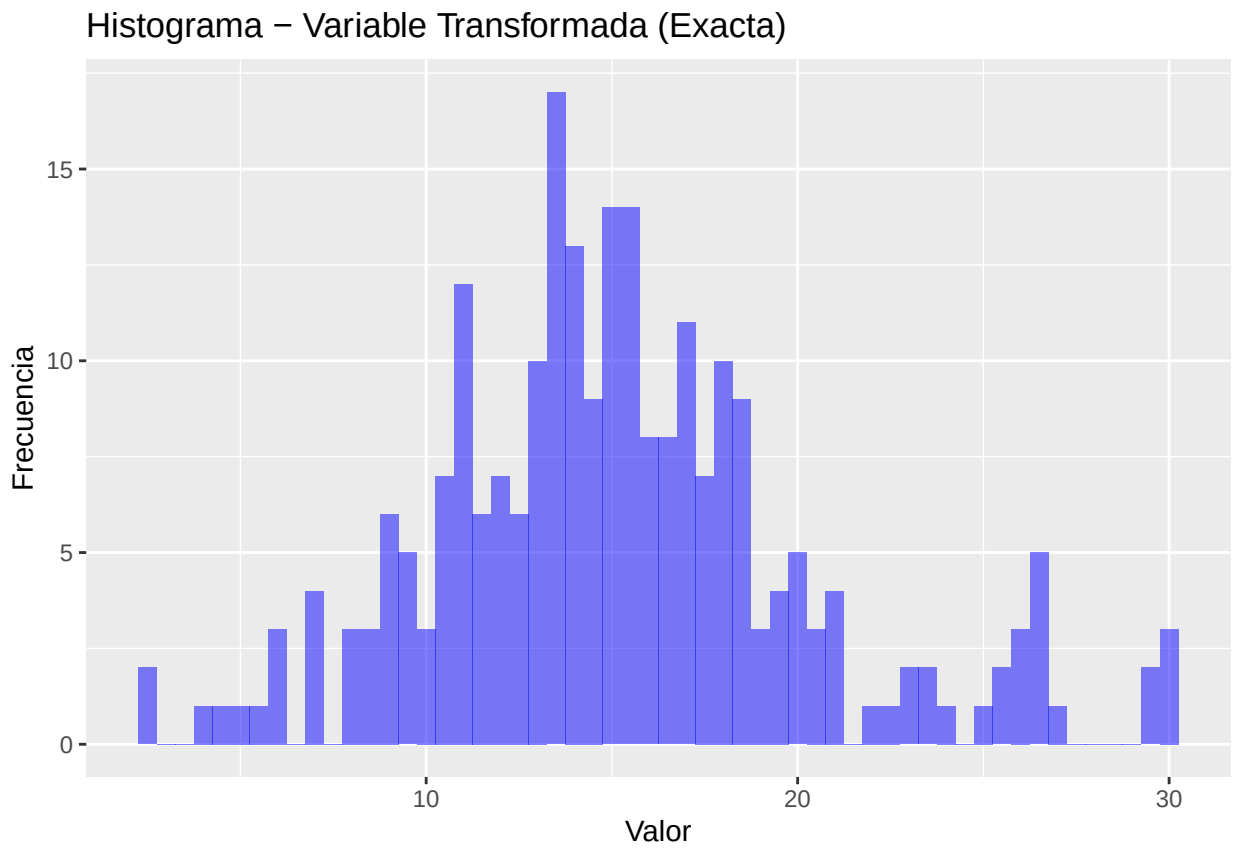
En términos de normalidad, las distribuciones transformadas parecen más simétricas y menos sesgadas en comparación con la distribución original. La transformación Box-Cox, ya sea exacta o aproximada, tiende a acercar los valores de sesgo y curtosis hacia valores más cercanos a 0, lo que sugiere una distribución más similar a la normal. Sin embargo, es importante recordar que estas medidas no proporcionan una prueba definitiva de normalidad, por lo que también debes considerar realizar pruebas estadísticas formales, como la prueba de Anderson-Darling, para evaluar la normalidad de las distribuciones transformadas.

```
hist_original <- ggplot(M, aes(x = Carbohydrates)) +
  geom_histogram(binwidth = 20, fill = "blue", alpha = 0.5) +
  labs(title = "Histograma - Distribución Original",
       x = "Valor",
       y = "Frecuencia")

# Crear un dataframe con la variable transformada exacta
data_transformada_exacta <- data.frame(variable = data_transformada_exacta)

# Crear el histograma
hist_exacta <- ggplot(data_transformada_exacta, aes(x = variable)) +
  geom_histogram(binwidth = 0.5, fill = "blue", alpha = 0.5) +
  labs(title = "Histograma - Variable Transformada (Exacta)",
       x = "Valor",
       y = "Frecuencia")
```

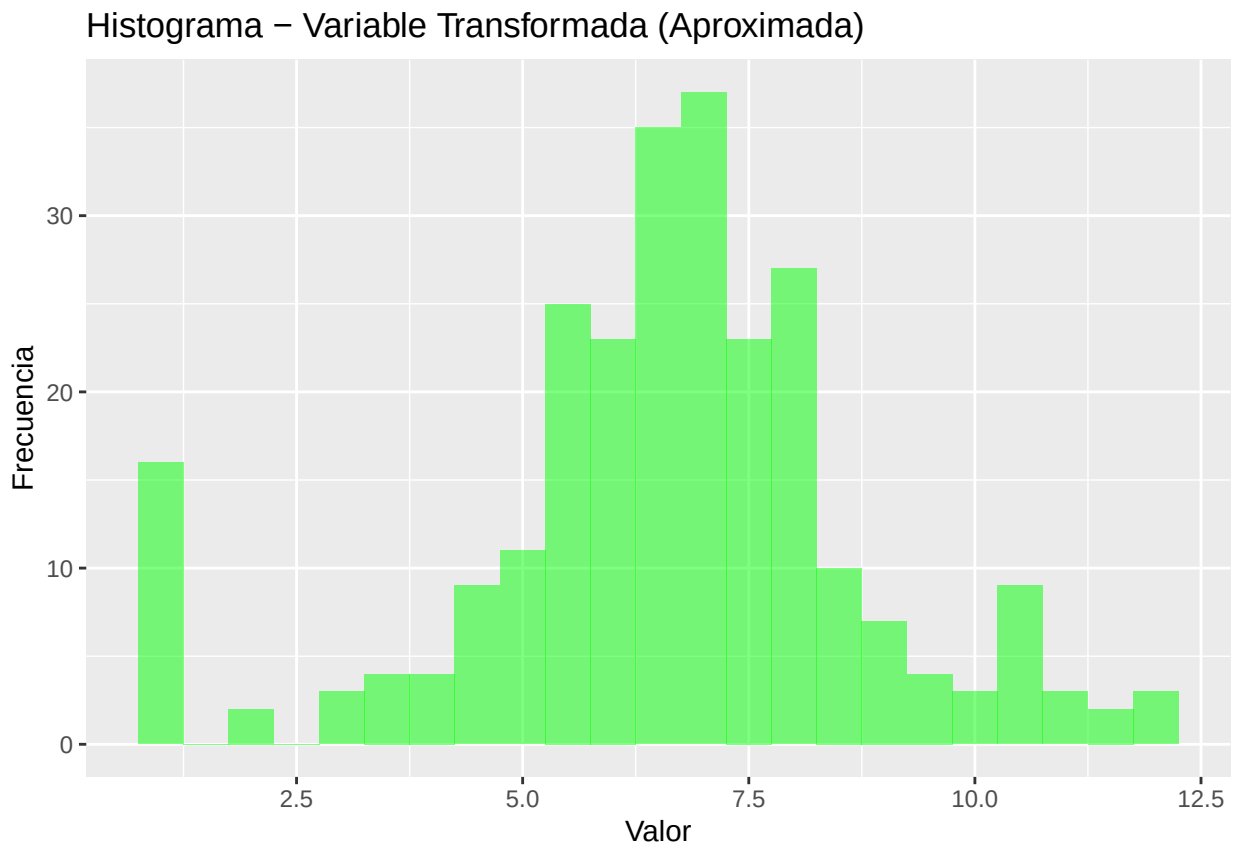
```r
data_transformada_aproximada <- data.frame(variable = data_transformada_aproximada)

hist_aproximada <- ggplot(data_transformada_aproximada, aes(x = variable)) +
  geom_histogram(binwidth = 0.5, fill = "green", alpha = 0.5) +
  labs(title = "Histograma - Variable Transformada (Aproximada)",
       x = "Valor",
       y = "Frecuencia")

print(hist_original)
```




```r
print(hist_exacta)
```




```
print(hist_aproximada)
```



Los histogramas de las variables transformadas tienen una forma más parecida a la de una distribución normal, sin embargo necesitamos una prueba anderson darling para asegurar normalidad.

```
library(nortest)

data_transformada_aproximada <- carb1
data_transformada_exacta <- carb2
data_transformada_aproximada = data_transformada_aproximada[data_transformada_aproximada != 0]
data_transformada_exacta = data_transformada_exacta[data_transformada_exacta != 0]

ad_test_original <- ad.test(variable)

ad_test_exacta <- ad.test(data_transformada_exacta)

ad_test_aproximada <- ad.test(data_transformada_aproximada)

print("Prueba de Normalidad Anderson-Darling:")
```

```
## [1] "Prueba de Normalidad Anderson-Darling:"
```

```
print(ad_test_original)
```

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

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

```
print(ad_test_exacta)
```

```
##
##  Anderson-Darling normality test
##
## data:  data_transformada_exacta
## A = 2.2439, p-value = 1.052e-05
```

```
print(ad_test_aproximada)
```

```
##
##  Anderson-Darling normality test
##
## data:  data_transformada_aproximada
## A = 4.4524, p-value = 4.482e-11
```

variable original:El estadístico A es significativamente mayor que el valor crítico, y el p-valor es muy cercano a cero. Esto indica que los datos originales no siguen una distribución normal. La hipótesis nula de normalidad es rechazada.

transformación exacta: El estadístico A es significativamente mayor que el valor crítico, y el p-valor es muy cercano a cero. Esto indica que los datos transformados con la transformación Box-Cox (Exacta) no siguen una distribución normal. La hipótesis nula de normalidad es rechazada.

variable aproximada: El estadístico A es significativamente mayor que el valor crítico, y el p-valor es muy cercano a cero. Esto indica que los datos transformados con la transformación Box-Cox (Aproximada) tampoco siguen una distribución normal. La hipótesis nula de normalidad es rechazada.

En resumen, en todos los casos, los p-valores son muy pequeños, lo que sugiere que los datos no siguen una distribución normal. Esto significa que los tres conjuntos de datos originales y transformados no se ajustan bien a una distribución normal y es posible que tengan una distribución diferente.

Transformacion Yeo Johnson

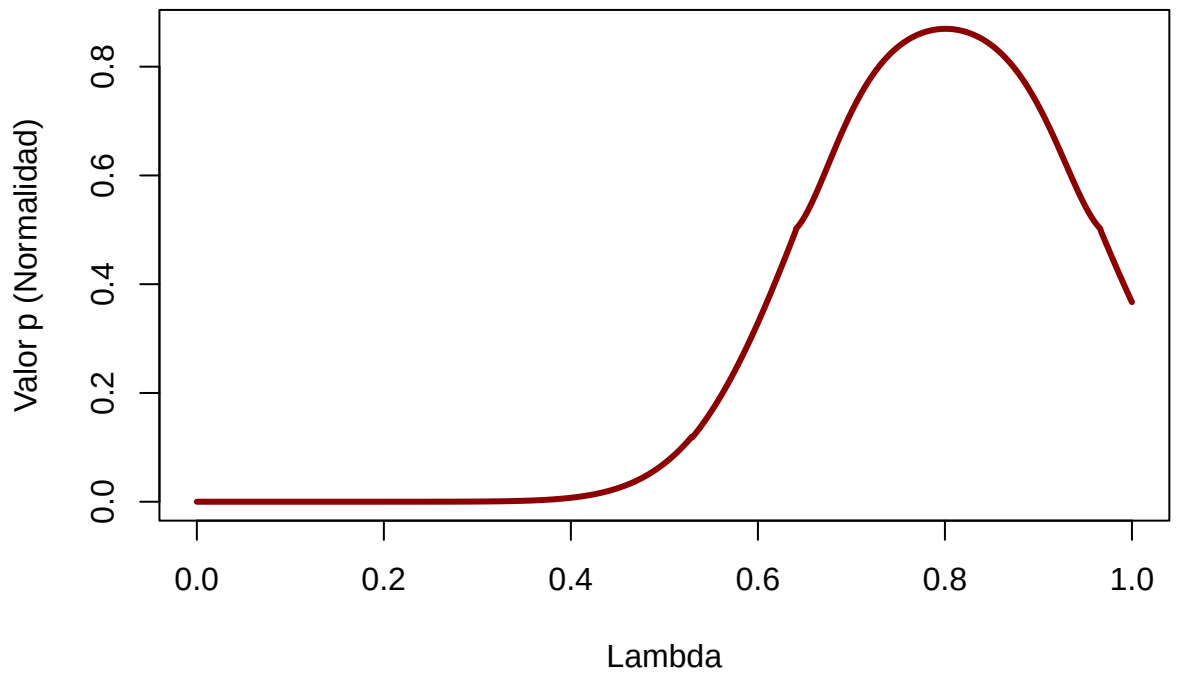
```
library(VGAM)
```

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

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

```
#Elimina 0s
val_sin0 = M$Carbohydrates[M$Carbohydrates !=0]

q1c = quantile(val_sin0, probs = 0.25)
q3c = quantile(val_sin0, probs = 0.75)
ric = IQR(val_sin0)
```

```r
val= val_sin0[val_sin0 < q3c + 1.5*ric]


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

N=as.data.frame(D)

plot(N$V1,N$V2,
type="l",col="darkred",lwd=3,
xlab="Lambda",
ylab="Valor p (Normalidad)")
```



Eliminando los valores 0 y los valores atípicos podemos ver en la gráfica que con un valor lambda de 0.8 se consigue un valor p aproximado de 0.8.

```r
val_yeo = yeo.johnson(val, lambda = 0.8)

ad.test(val_yeo)
```

```
##
##  Anderson-Darling normality test
##
## data:  val_yeo
## A = 0.20579, p-value = 0.8696
```

Dado que el valor p es bastante alto (0.8696) y el estadístico de Anderson-Darling (A) es muy bajo (0.20579), no hay evidencia sólida para rechazar la hipótesis nula de que los datos en val_yeo se distribuyen normalmente. Esto indica que los datos podrían considerarse aproximadamente normales según los resultados de la prueba de Anderson-Darling.

Ventajas y Desventajas de los Modelos de Transformación de Box-Cox y Yeo-Johnson:

El modelo de Box-Cox es una técnica de transformación utilizada para normalizar los datos y ajustarlos a una distribución más cercana a la normal. Sus ventajas incluyen la simplicidad y la interpretabilidad, además de ser efectivo en la normalización de datos estrictamente positivos. Sin embargo, presenta limitaciones al no poder manejar datos que contengan ceros o valores negativos, y puede ser sensible a valores atípicos.

Por otro lado, el modelo de Yeo-Johnson es una extensión del modelo de Box-Cox que permite manejar datos con ceros y valores negativos. Su flexibilidad lo hace adecuado para una amplia gama de distribuciones, incluyendo aquellas que son asimétricas o bimodales. Aunque es más complejo que el modelo de Box-Cox y puede ser más difícil de interpretar, su capacidad para lidiar con una variedad de situaciones lo hace valioso en análisis estadísticos y modelado.

Diferencias entre Transformación y Escalamiento de Datos:

La transformación de datos implica aplicar una función matemática a los valores individuales de los datos para cambiar su distribución. Se utiliza para lograr suposiciones estadísticas como la normalidad en análisis posteriores. Por otro lado, el escalamiento de datos se refiere a cambiar la escala de los valores de los datos sin alterar su distribución. Esta técnica es útil para estandarizar los datos y asegurarse de que estén en una misma escala, lo que es beneficioso para algoritmos sensibles a la magnitud de las características.

Cuándo utilizar cada uno:

La transformación de datos se usa cuando se necesita ajustar los datos a una distribución más normal, lo que es importante en análisis estadísticos y modelado donde se asumen suposiciones sobre la distribución. El escalamiento de datos, por otro lado, se aplica cuando se desean datos en una misma escala para compararlos de manera equitativa o cuando se usan algoritmos que son sensibles a la magnitud de las características, como regresión lineal o redes neuronales.