Act 5. Transformaciones

Andrés Villarreal González

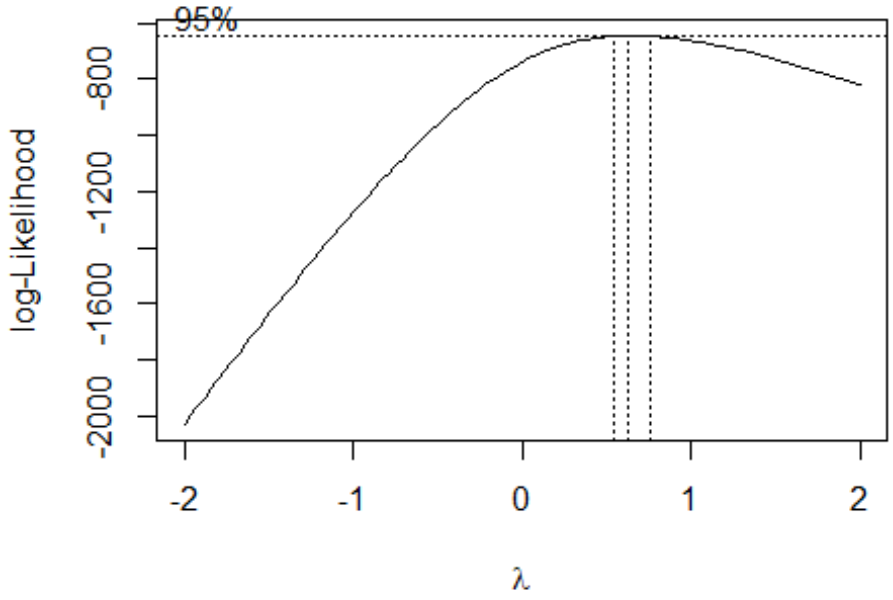
2024-08-15

Actividad 5

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

library(MASS)
library(ggplot2)
library(e1071)
library(nortest)
```

Transformaación Box-Cox

```
# Seleccionar la variable Sodium
sodium <- M$Carbohydrates

# Aplicar la transformación de Box-Cox
bc <- boxcox((sodium + 1) ~ 1)
```



```r
# Encontrar el mejor valor de lambda
l <- bc$x[which.max(bc$y)]
l

## [1] 0.6262626
```

Ecuaciones de los modelos:

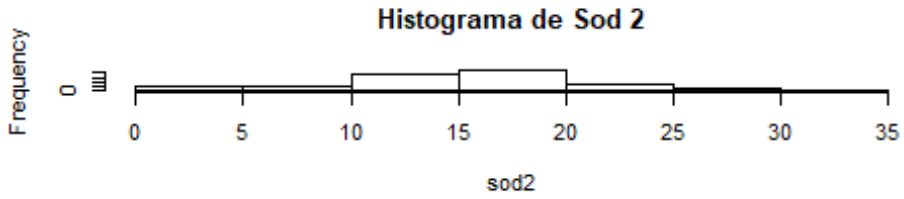
Aproximado:

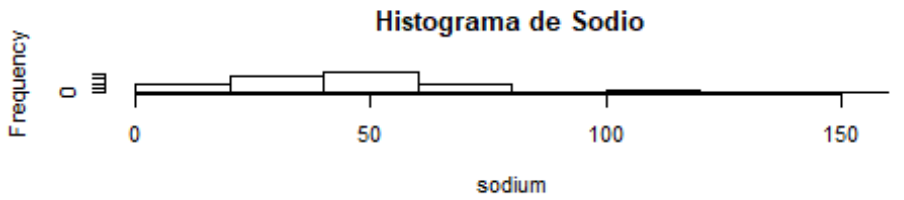
$$cal_1 = \sqrt{x+1}$$

Exacto:

$$cal_2 = \frac{(x+1)^{0.63} - 1}{0.63}$$

```r
sod1=sqrt(sodium+1)
sod2=((sodium+1)^l-1)/l
par(mfrow=c(3,1))
hist(sod1,col=0,main="Histograma de Sod 1")
hist(sod2,col=0,main="Histograma de Sod 2")
hist(sodium,col=0,main="Histograma de Sodio")
```






```r
# Análisis descriptivo para los datos originales y transformados
analisis_descriptivo <- function(datos, nombre) {
  resumen <- summary(datos)
  sesgo <- skewness(datos)
```

```r
  curtosis <- kurtosis(datos)

  cat(paste("\nMedidas descriptivas para", nombre, ":\n"))
  cat("Mínimo:", resumen["Min."], "\n")
  cat("Cuartil 1:", resumen["1st Qu."], "\n")
  cat("Mediana:", resumen["Median"], "\n")
  cat("Media:", resumen["Mean"], "\n")
  cat("Cuartil 3:", resumen["3rd Qu."], "\n")
  cat("Máximo:", resumen["Max."], "\n")
  cat("Sesgo:", sesgo, "\n")
  cat("Curtosis:", curtosis, "\n")
}

# Datos originales
analisis_descriptivo(M$Sodium, "Sodium original")
```

```
##
## Medidas descriptivas para Sodium original :
## Mínimo: 0
## Cuartil 1: 107.5
## Mediana: 190
## Media: 495.75
## Cuartil 3: 865
## Máximo: 3600
## Sesgo: 1.526317
## Curtosis: 2.75191
```

```r
# Modelo exacto (Box-Cox exacto)
analisis_descriptivo(sod1, "Sodium transformado (modelo aproximado)")
```

```
##
## Medidas descriptivas para Sodium transformado (modelo aproximado) :
## Mínimo: 1
## Cuartil 1: 5.567764
## Mediana: 6.708204
## Media: 6.583244
## Cuartil 3: 7.81025
## Máximo: 11.91638
## Sesgo: -0.4939626
## Curtosis: 0.90923
```

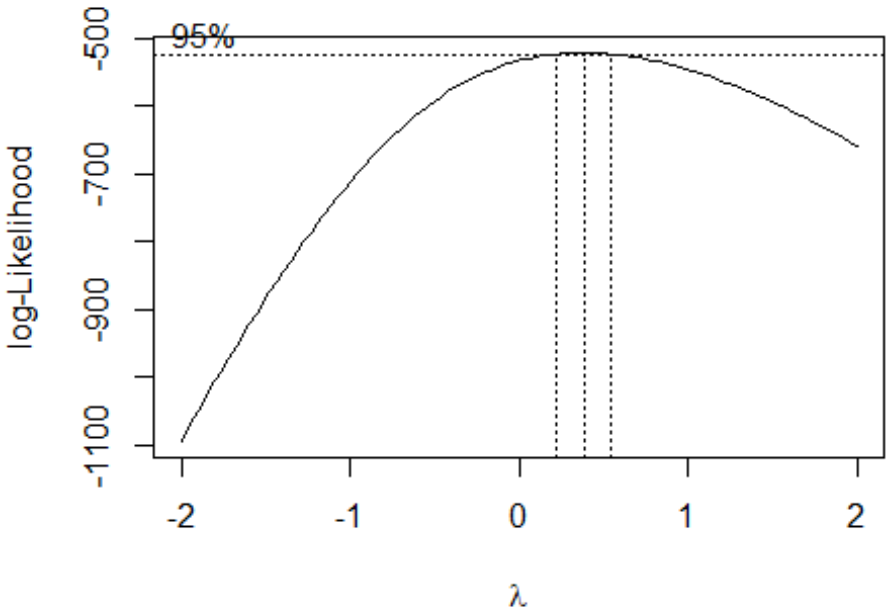
```r
# Modelo aproximado (logaritmo)
analisis_descriptivo(sod2, "Sodium transformado (modelo exacto)")
```

```
##
## Medidas descriptivas para Sodium transformado (modelo exacto) :
## Mínimo: 0
## Cuartil 1: 12.11923
## Mediana: 15.72485
## Media: 15.66877
## Cuartil 3: 19.36021
```

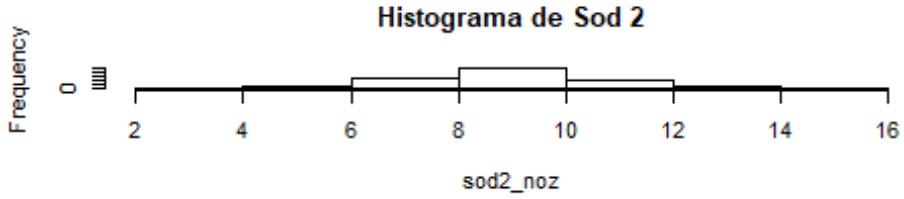
```
## Máximo: 33.97793
## Sesgo: -0.08250202
## Curtosis: 0.6381974

# Prueba de normalidad: Anderson-Darling o Jarque-Bera
cat("\nPrueba de normalidad Anderson-Darling (Original):\n")

##
## Prueba de normalidad Anderson-Darling (Original):

ad.test(M$Sodium)

##
##  Anderson-Darling normality test
##
## data:  M$Sodium
## A = 21.406, p-value < 2.2e-16

cat("\nPrueba de normalidad Anderson-Darling (Modelo exacto):\n")

##
## Prueba de normalidad Anderson-Darling (Modelo exacto):

ad.test(sod1)

##
##  Anderson-Darling normality test
##
## data:  sod1
## A = 4.4524, p-value = 4.482e-11

cat("\nPrueba de normalidad Anderson-Darling (Modelo aproximado):\n")

##
## Prueba de normalidad Anderson-Darling (Modelo aproximado):

ad.test(sod2)

##
##  Anderson-Darling normality test
##
## data:  sod2
## A = 3.1076, p-value = 8.182e-08
```

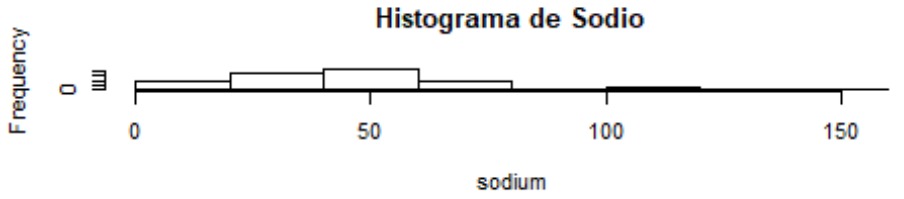
Eliminando Ceros

```
# Eliminar los ceros de la variable
sodium_no_zeros <- sodium[sodium > 0]
# Aplicar la transformación de Box-Cox
bc <- boxcox(sodium_no_zeros ~ 1)
```




```r
# Encontrar el mejor valor de lambda
l_noz <- bc$x[which.max(bc$y)]
cat("El valor óptimo de lambda es:", l_noz, "\n")

## El valor óptimo de lambda es: 0.3838384

sod1_noz=sqrt(sodium_no_zeros)
sod2_noz=((sodium_no_zeros+1)^l_noz-1)/l_noz
par(mfrow=c(3,1))
hist(sod1_noz,col=0,main="Histograma de Sod 1")
hist(sod2_noz,col=0,main="Histograma de Sod 2")
hist(sodium,col=0,main="Histograma de Sodio")
```






```
D0=ad.test(sodium_no_zeros)
D1=ad.test(sod1_noz)
D2=ad.test(sod2_noz)

m0=round(c(as.numeric(summary(sodium_no_zeros)),kurtosis(sodium_no_zeros)
,skewness(sodium_no_zeros),D0$p.value),3)
m1=round(c(as.numeric(summary(sod1_noz)),kurtosis(sod1_noz),skewness(sod1
_noz),D1$p.value),3)
m2=round(c(as.numeric(summary(sod2_noz)),kurtosis(sod2_noz),skewness(sod2
_noz),D2$p.value),3)

m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Primer modelo","Segundo Modelo")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesg
o","Valor p")

m

##                Minimo     Q1 Mediana  Media     Q3  Máximo Curtosis
Sesgo
## Original        4.000 34.000  46.000 50.451 61.000 141.000    1.763
1.214
## Primer modelo   2.000  5.831   6.782  6.871  7.810  11.874    0.602
0.318
## Segundo Modelo  2.227  7.593   8.815  8.866 10.096  14.852    0.592
0.112
##                Valor p
```

```
## Original         0.000
## Primer modelo    0.000
## Segundo Modelo   0.001
```

Transformacion Yeo - Johnson

```
library(VGAM)
```

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

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

```
library(car)       # Para yeo.johnson
```

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

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

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

```
library(nortest)  # Para ad.test

sod3<- yeo.johnson(sodium_no_zeros, lambda = l_noz)

library(VGAM)
lp <- seq(0, 1, 0.001)
nlp <- length(lp)
n <- length(sodium_no_zeros)
D <- matrix(as.numeric(NA), ncol = 2, nrow = nlp)
d <- NA

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

max_p <- max(D[, 2])
lambda_max_p <- D[D[, 2] == max_p, 1]


cat("Máximo valor p:", max_p, "\n")
```

```
## Máximo valor p: 0.001275547
```

```
cat("Correspondiente lambda:", lambda_max_p, "\n")
```

```
## Correspondiente lambda: 0.302
```

Despues de aplicar ambas transformaciones a los datos podemos ver que ni una de ellas logran seguir una distribucion normal desdpues de hacer las pruebas de normalidad, se acercan un poco mas a esta distribucion pero habra que probar con otras variables.

Diferencias entre Transformación y Escalamiento de Datos

Propósito: Transformación: Cambia la distribución de los datos para mejorar su ajuste a modelos o manejar valores atípicos (ej. transformación logarítmica). Escalamiento: Ajusta la escala de los datos para que sean comparables, sin cambiar la distribución (ej. normalización y estandarización).

Efecto en la Distribución: Transformación: Modifica la forma de la distribución de los datos. Escalamiento: No cambia la forma de la distribución, solo la escala.

Uso en Modelos: Transformación: Para ajustar distribuciones no normales o cumplir supuestos de modelos. Escalamiento: Para asegurar que las características tengan la misma escala en modelos sensibles a la escala, como K-means o PCA. Cuándo Utilizar Cada Uno Transformación: Cuando los datos no cumplen con los supuestos del modelo o tienen distribuciones sesgadas. Escalamiento: Cuando los algoritmos de aprendizaje automático requieren que las características estén en la misma escala.