Transformaciones

Nallely Serna

2024-08-15

Trabaja con el set de datos Mc Donalds menu Download Mc Donalds menu, que contiene diversas características del menú de alimentos de Mc Donalds.

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:

Utiliza la transformación Box-Cox. Utiliza el modelo exacto y el aproximado de acuerdo con las sugerencias de Box y Cox para la transformación Escribe las ecuaciones de los modelos encontrados. Analiza la normalidad de las transformaciones obtenidas con los datos originales. Utiliza como argumento de normalidad: Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales. Realiza la prueba de normalidad de Anderson-Darling o de Jarque Bera para los datos transformados y los originales Detecta anomalías y corrige tu base de datos (datos atípicos, ceros anámalos, etc). 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). Escribe la ecuación del modelo encontrado. Analiza la normalidad de las transformaciones obtenidas con los datos originales. Utiliza como argumento de normalidad: Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales. Realiza la prueba de normalidad de Anderson-Darling para los datos transformados y los originales 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. Concluye sobre las ventajas y desventajes de los modelos de Box Cox y de Yeo Johnson. Analiza las diferencias entre la transformación y el escalamiento de los datos: Escribe al menos 3 diferencias entre lo que es la transformación y el escalamiento de los datos Indica cuándo es necesario utilizar cada uno

```
mcdonalds=read.csv("mc-donalds-menu.csv") #leer la base de datos
#M$variable   #para llamar una variable, aunque también la puedes leer
con corchetes cuadrados M[renglón, columna]
head(mcdonalds)
```

```
##    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
```

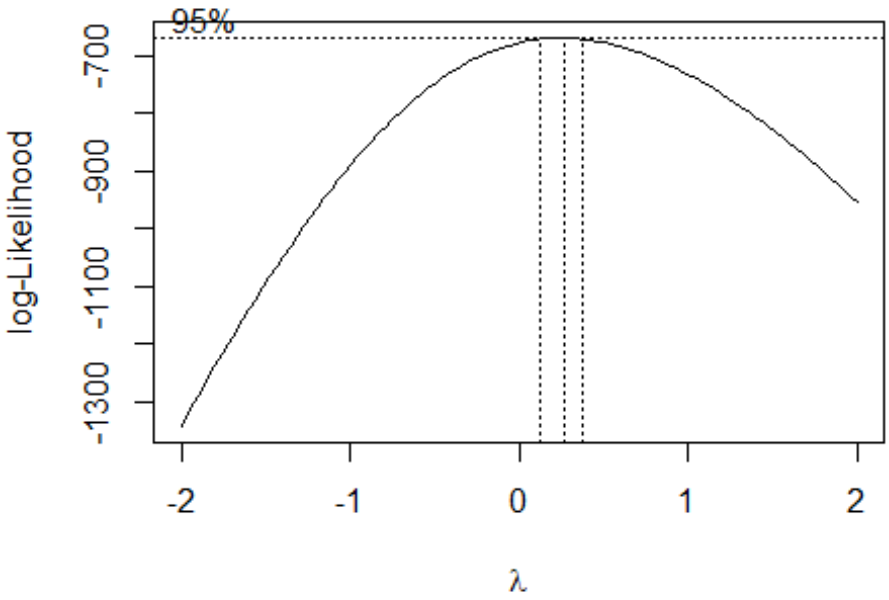
```
# Selección de la variable
sodio <- mcdonalds$Sugars

# Verificar si hay valores menores o iguales a 0
sum(sodio <= 0)
```

```
## [1] 25
```

```
# Eliminar valores no positivos
sodio_positivo <- sodio[sodio > 0]
```

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

```
# Aplicar la transformación Box-Cox a los datos ajustados
boxcox_result <- boxcox(sodio_positivo ~ 1, lambda = seq(-2, 2, 0.1))
```




```r
# Encontrar el valor óptimo de lambda
lambda_opt <- boxcox_result$x[which.max(boxcox_result$y)]

# Transformación exacta con lambda óptimo
sodio_bc <- (sodio_positivo^lambda_opt - 1) / lambda_opt

# Transformación aproximada con lambda redondeado
lambda_approx <- round(lambda_opt, digits = 1)
sodio_bc_approx <- (sodio_positivo^lambda_approx - 1) / lambda_approx
```

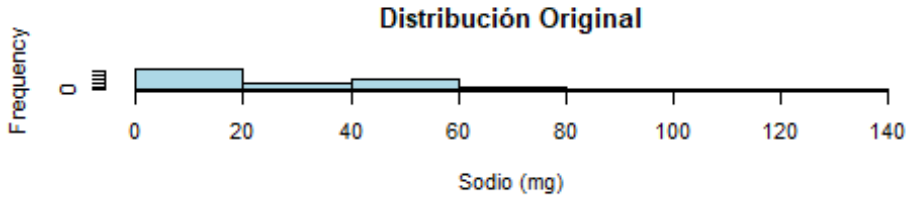
#Analiza la normalidad de las transformaciones obtenidas con los datos originales. Utiliza como argumento de normalidad:

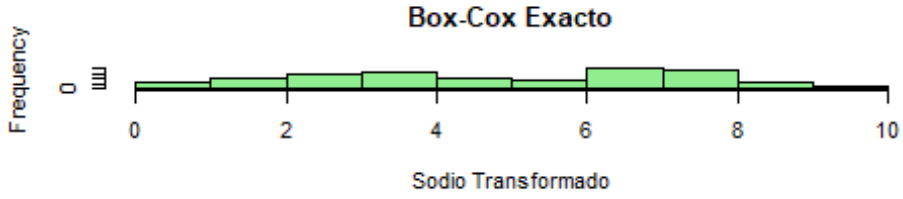
```r
#Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil
3, sesgo y curtosis.



# Prueba de normalidad Anderson-Darling
D0 <- ad.test(sodio_positivo)
D1 <- ad.test(sodio_bc)
D2 <- ad.test(sodio_bc_approx)

# Resumen de medidas para cada conjunto de datos
m0 <- round(c(as.numeric(summary(sodio_positivo)),
kurtosis(sodio_positivo), skewness(sodio_positivo), D0$p.value), 3)
```

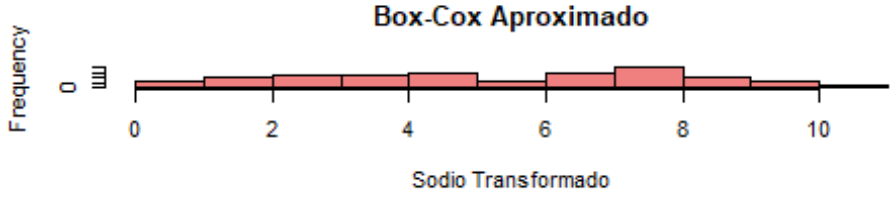
```r
m1 <- round(c(as.numeric(summary(sodio_bc)), kurtosis(sodio_bc),
skewness(sodio_bc), D1$p.value), 3)
m2 <- round(c(as.numeric(summary(sodio_bc_approx)),
kurtosis(sodio_bc_approx), skewness(sodio_bc_approx), D2$p.value), 3)

# Crear tabla con los resultados
m <- as.data.frame(rbind(m0, m1, m2))
row.names(m) <- c("Original", "Box-Cox Exacto", "Box-Cox Aproximado")
names(m) <- c("Minimo", "Q1", "Mediana", "Media", "Q3", "Máximo",
"Curtosis", "Sesgo", "Valor p")

# Mostrar tabla
print(m)
```

```
##                    Minimo    Q1 Mediana  Media     Q3  Máximo Curtosis
Sesgo
## Original                1 8.000  23.000 32.553 52.000 128.000    0.356
0.947
## Box-Cox Exacto          0 2.766   4.868  4.844  6.940   9.809   -1.147
-0.094
## Box-Cox Aproximado      0 2.887   5.205  5.236  7.573  10.957   -1.149
-0.041
##                    Valor p
## Original                 0
## Box-Cox Exacto           0
## Box-Cox Aproximado       0
```

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

# Histogramas
par(mfrow=c(3,1))
hist(sodio_positivo, main="Distribución Original", xlab="Sodio (mg)",
col="lightblue")
hist(sodio_bc, main="Box-Cox Exacto", xlab="Sodio Transformado",
col="lightgreen")
hist(sodio_bc_approx, main="Box-Cox Aproximado", xlab="Sodio
Transformado", col="lightcoral", breaks=10)
```








```r
# Detección de valores atípicos usando el método de Tukey (IQR)
Q1 <- quantile(sodio_positivo, 0.25)
Q3 <- quantile(sodio_positivo, 0.75)
IQR <- Q3 - Q1

# Definición de límites inferior y superior
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Identificación de outliers
outliers <- sodio_positivo[sodio_positivo < lower_bound | sodio_positivo
> upper_bound]

# Eliminar outliers de la base de datos
sodio_limpio <- sodio_positivo[!(sodio_positivo < lower_bound |
sodio_positivo > upper_bound)]


# Transformación Yeo-Johnson
yeo_johnson <- powerTransform(sodio_limpio, family="yjPower")

# Extraer el valor óptimo de lambda
lambda_yj <- yeo_johnson$lambda

# Aplicar la transformación Yeo-Johnson
sodio_yj <- (sodio_limpio^lambda_yj - 1) / lambda_yj
```

```r
# Prueba de normalidad Anderson-Darling para Yeo-Johnson
D3 <- ad.test(sodio_yj)

# Resumen de medidas para Yeo-Johnson
m3 <- round(c(as.numeric(summary(sodio_yj)), kurtosis(sodio_yj),
skewness(sodio_yj), D3$p.value), 3)

# Agregar Yeo-Johnson a la tabla
m <- rbind(m, m3)
row.names(m)[4] <- "Yeo-Johnson"

# Mostrar tabla actualizada
print(m)
```

```
##                    Minimo    Q1 Mediana  Media     Q3  Máximo Curtosis
Sesgo
## Original                1 8.000  23.000 32.553 52.000 128.000    0.356
0.947
## Box-Cox Exacto          0 2.766   4.868  4.844  6.940   9.809   -1.147
-0.094
## Box-Cox Aproximado      0 2.887   5.205  5.236  7.573  10.957   -1.149
-0.041
## Yeo-Johnson             0 2.610   4.494  4.466  6.374   8.570   -1.181
-0.175
##                    Valor p
## Original                 0
## Box-Cox Exacto           0
## Box-Cox Aproximado       0
## Yeo-Johnson              0
```

```r
# Abrir una nueva ventana gráfica grande
dev.new(width = 10, height = 7)

# Histogramas para Yeo-Johnson
par(mfrow=c(4,1))
hist(sodio_positivo, main="Distribución Original", xlab="Sodio (mg)",
col="lightblue")
hist(sodio_bc, main="Box-Cox Exacto", xlab="Sodio Transformado",
col="lightgreen")
hist(sodio_bc_approx, main="Box-Cox Aproximado", xlab="Sodio
Transformado", col="lightcoral", breaks=10)
hist(sodio_yj, main="Yeo-Johnson", xlab="Sodio Transformado",
col="lightpink", breaks=10)
```

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

Box-Cox: Ventajas: Solo es aplicable a datos positivos, transforma de manera efectiva distribuciones sesgadas. Desventajas: No es aplicable a datos con valores cero o negativos.

Yeo-Johnson: Ventajas: Puede manejar datos con valores cero o negativos, lo que lo hace más versátil. Desventajas: Puede ser más complejo de interpretar en algunos casos.

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

Transformación: Cambia la forma de la distribución de los datos. Se utiliza para mejorar la normalidad de los datos o la linealidad en un modelo. Ejemplos: Transformaciones logarítmicas, Box-Cox, Yeo-Johnson.

Escalamiento: Ajusta los datos a un rango específico, sin cambiar su distribución. Se utiliza para normalizar la escala de los datos, especialmente antes de aplicar algoritmos sensibles a la escala. Ejemplos: Min-Max Scaling, Z-score normalization.