Actividad 5_Módulo1_transformaciones

Rodolfo Jesús Cruz Rebollar

2024-08-14

```r
# Cargar los datos del archivo csv para su análisis

datos = read.csv("mcdonalds.csv")

head(datos)
```

```
##    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
```

```
# Seleccionar la variable sodio (Sodium) para la transformación

sodio = datos$Sodium

head(sodio)
```

```
## [1] 750 770 780 860 880 960
```

1. Transformación de Box-Cox

Para utilizar la transformación de Box-Cox para normalizar los datos, primero es necesario asegurarnos de que todos los datos sean positivos o mayores que 0, por lo que en caso contrario, será necesario sumar a los datos ya sea 1 unidad (en caso de que el dato mínimo sea 0) o bien, la cantidad de unidades que haya desde el dato mínimo (valor negativo) hasta el 0 y esa diferencia más otra unidad (esto último en caso de que el mínimo de los datos sea un número negativo), esto con el propósito principal de que no quede ningún dato 0 o negativo para aplicar Box-Cox.

```
# Verificar si hay datos 0 o negativos antes de aplicar Box-Cox

sodio[sodio <= 0]
```

```
## [1] 0 0 0 0 0 0 0 0 0 0
```

Dado que se encontraron datos 0, es necesario sumar a los datos originales 1 unidad para que los valores que son 0 ahora sean 1 (valor positivo pequeño que no afectará en gran medida la realización del análisis).

```
# Sumar 1 a cada uno de los datos originales de sodio para eliminar los valores 0

sodio = sodio + 1

# Verificar que se hayan eliminado los valores 0

sodio[!(sodio > 0)]

## numeric(0)
```

Dado que ya no se tienen datos de sodio que sean 0, ahora se puede proceder a realizar la transformación de Box-Cox para normalización.

Box-Cox: modelo aproximado

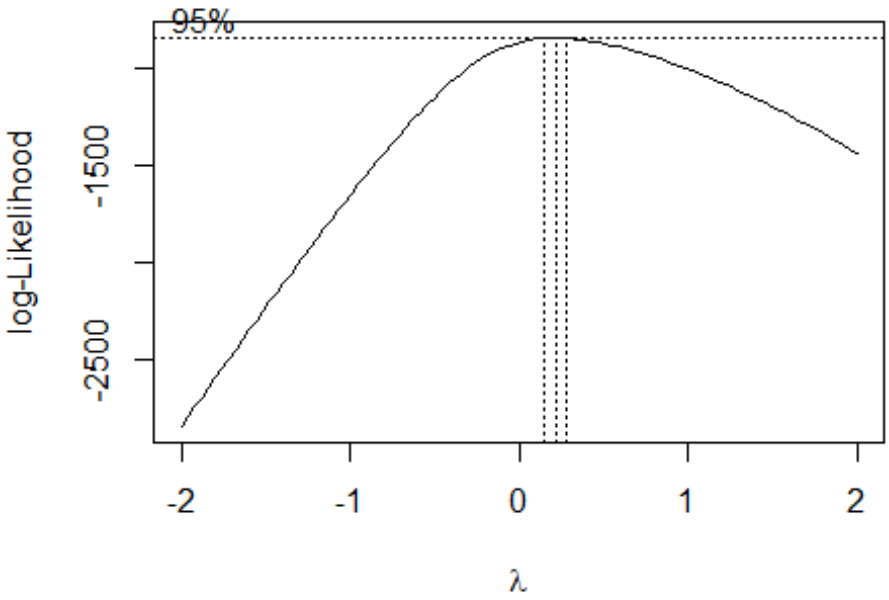
Para realizar la transformación de Box-Cox por el método aproximado, primero es necesario determinar el valor del parámetro λ que maximiza la función de verosimilitud para nuestro conjunto de datos.

```
# Biblioteca MASS para realizar transformación de Box-Cox

library(MASS)

# Realizar la transformación de Box-Cox para la variable sodio

box_cox_sodio = boxcox(sodio ~ 1)
```




```
# Calcular el valor del Lambda que maximiza la función de verosimilitud

lambda_optimo = box_cox_sodio$x[which.max(box_cox_sodio$y)]

box_cox_sodio
```

```
## $x
##   [1] -2.00000000 -1.95959596 -1.91919192 -1.87878788 -1.83838384 -
1.79797980
##   [7] -1.75757576 -1.71717172 -1.67676768 -1.63636364 -1.59595960 -
1.55555556
##  [13] -1.51515152 -1.47474747 -1.43434343 -1.39393939 -1.35353535 -
1.31313131
##  [19] -1.27272727 -1.23232323 -1.19191919 -1.15151515 -1.11111111 -
1.07070707
##  [25] -1.03030303 -0.98989899 -0.94949495 -0.90909091 -0.86868687 -
0.82828283
##  [31] -0.78787879 -0.74747475 -0.70707071 -0.66666667 -0.62626263 -
0.58585859
##  [37] -0.54545455 -0.50505051 -0.46464646 -0.42424242 -0.38383838 -
0.34343434
##  [43] -0.30303030 -0.26262626 -0.22222222 -0.18181818 -0.14141414 -
0.10101010
##  [49] -0.06060606 -0.02020202  0.02020202  0.06060606  0.10101010
0.14141414
##  [55]  0.18181818  0.22222222  0.26262626  0.30303030  0.34343434
0.38383838
```

```
##  [61]  0.42424242  0.46464646  0.50505051  0.54545455  0.58585859
0.62626263
##  [67]  0.66666667  0.70707071  0.74747475  0.78787879  0.82828283
0.86868687
##  [73]  0.90909091  0.94949495  0.98989899  1.03030303  1.07070707
1.11111111
##  [79]  1.15151515  1.19191919  1.23232323  1.27272727  1.31313131
1.35353535
##  [85]  1.39393939  1.43434343  1.47474747  1.51515152  1.55555556
1.59595960
##  [91]  1.63636364  1.67676768  1.71717172  1.75757576  1.79797980
1.83838384
##  [97]  1.87878788  1.91919192  1.95959596  2.00000000
##
## $y
##   [1] -2847.0818 -2796.8929 -2746.8131 -2696.8479 -2647.0026 -
2597.2824
##   [7] -2547.6927 -2498.2399 -2448.9306 -2399.7718 -2350.7714 -
2301.9374
##  [13] -2253.2790 -2204.8058 -2156.5283 -2108.4582 -2060.6077 -
2012.9909
##  [19] -1965.6225 -1918.5193 -1871.6995 -1825.1828 -1778.9921 -
1733.1515
##  [25] -1687.6889 -1642.6349 -1598.0233 -1553.8937 -1510.2876 -
1467.2549
##  [31] -1424.8498 -1383.1330 -1342.1760 -1302.0539 -1262.8571 -
1224.6825
##  [37] -1187.6377 -1151.8453 -1117.4321 -1084.5386 -1053.3104 -
1023.8924
##  [43]  -996.4279  -971.0541  -947.8776  -926.9939  -908.4592  -
892.2846
##  [49]  -878.4644  -866.9311  -857.5992  -850.3567  -845.0653  -
841.5838
##  [55]  -839.7658  -839.4642  -840.5426  -842.8739  -846.3375  -
850.8324
##  [61]  -856.2623  -862.5453  -869.6106  -877.3926  -885.8391  -
894.9008
##  [67]  -904.5368  -914.7120  -925.3940  -936.5571  -948.1769  -
960.2331
##  [73]  -972.7081  -985.5857  -998.8527 -1012.4966 -1026.5069 -
1040.8741
##  [79] -1055.5897 -1070.6463 -1086.0368 -1101.7553 -1117.7961 -
1134.1538
##  [85] -1150.8238 -1167.8014 -1185.0823 -1202.6623 -1220.5373 -
1238.7033
##  [91] -1257.1564 -1275.8926 -1294.9079 -1314.1983 -1333.7597 -
1353.5879
##  [97] -1373.6786 -1394.0273 -1414.6296 -1435.4811
```

```r
# Mostrar el valor del lambda que maximiza la función de verosimilitud

cat("Lambda óptimo: ", lambda_optimo)
```

```
## Lambda óptimo:  0.2222222
```

Se observa que el valor de lambda que maximiza la función de verosimilitud es 0.2222222, lo cual es ligeramente más cercano a 0 que a 0.5, motivo por el cual, utilizaremos la transformación aproximada correspondiente a un valor de λ igual a 0 para realizar la transformación aproximada de Box-Cox:

$$log(x)$$

```r
# Realizar la transformación aproximada de Box-Cox utilizando la
expresión anterior

# Calcular el logaritmo base 10 (log(x)) de cada uno de los datos

boxcox_aprox = log(sodio, 10)

# Mostrar datos transformados mediante el modelo aproximado

head(boxcox_aprox)
```

```
## [1] 2.875640 2.887054 2.892651 2.935003 2.944976 2.982723
```

Box-Cox: modelo exacto

Dado que el valor óptimo de lambda calculado anteriormente es igual a 0.22, lo cual es diferente de 0, se usará la siguiente expresión que se utiliza precisamente cuando dicho lambda es distinto de 0:

$$f(x,\lambda) = \frac{x^{\lambda} - 1}{\lambda}$$

```r
# Normalizar los datos mediante el modelo exacto de Box-Cox

boxcox_exacto = (sodio ^ lambda_optimo - 1) / lambda_optimo

# Mostrar datos normalizados con el modelo exacto

head(boxcox_exacto)
```

```
## [1] 15.09944 15.21425 15.27078 15.70391 15.80727 16.20332
```

2. Ecuaciones de los modelos encontrados

Modelo aproximado: $sodio_1 = log(sodio + 1)$

Modelo exacto: $sodio_2 = \frac{sodio^{0.22}-1}{0.22}$

Donde sodio representa los valores de sodio a los cuales se les sumó inicialmente 1 unidad para quitar los valores 0 y que todavía no están normalizados. sodio 1 y sodio 2 son los nuevos datos de sodio ya normalizados, utilizando el modelo aproximado y exacto, respectivamente.

3. Datos transformados vs originales

Comparación de medidas

```
library(moments)

# Medidas de los datos originales (sin normalizar)

summary(sodio)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   108.5   191.0   496.8   866.0  3601.0

# sesgo y curtosis

cat("Sesgo de datos originales: ", skewness(sodio), "\n")

## Sesgo de datos originales:  1.535166

cat("Curtosis de datos originales: ", kurtosis(sodio))

## Curtosis de datos originales:  5.796412

# Medidas de los datos transformados con el modelo aproximado

summary(boxcox_aprox)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.035   2.281   2.294   2.937   3.556

# sesgo y curtosis

cat("Sesgo de datos normalizados: ", skewness(boxcox_aprox), "\n")

## Sesgo de datos normalizados:  -1.039777

cat("Curtosis de datos normalizados: ", kurtosis(boxcox_aprox))

## Curtosis de datos normalizados:  4.332194

# Medidas de los datos transformados con el modelo exacto

summary(boxcox_exacto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   8.249   9.958  11.008  15.730  23.267
```

```
# sesgo y curtosis
```

```
cat("Sesgo de datos normalizados: ", skewness(boxcox_exacto), "\n")
```

```
## Sesgo de datos normalizados:  -0.02985208
```

```
cat("Curtosis de datos normalizados: ", kurtosis(boxcox_exacto))
```

```
## Curtosis de datos normalizados:  2.328831
```

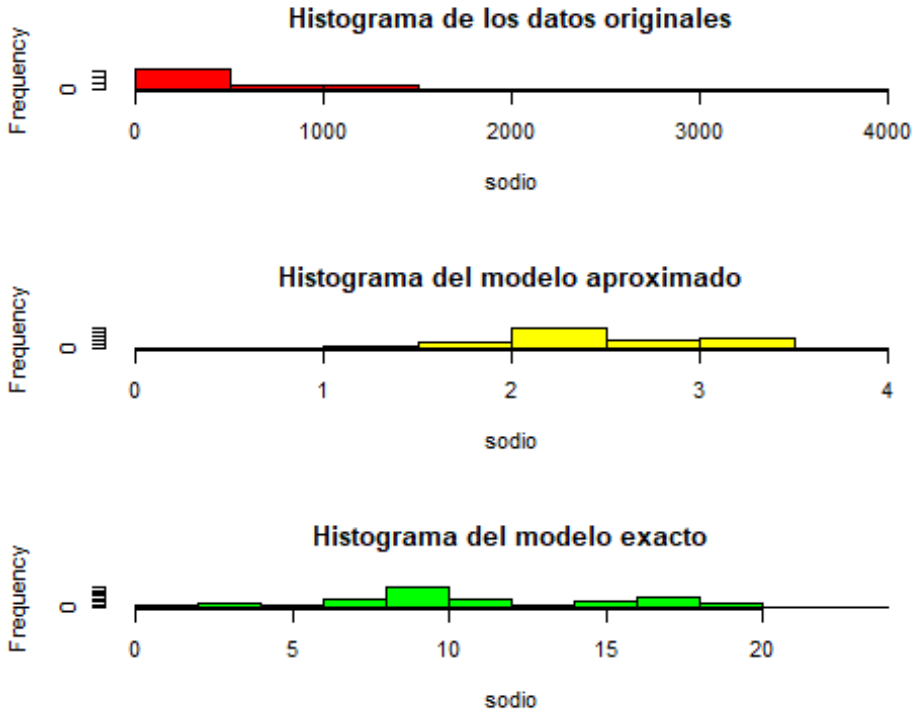
Al momento de calcular las medidas estadísticas tanto de los datos originales como de los normalizados tanto con el modelo aproximado como exacto, se puede observar que los valores de dichas medidas presentan ciertas diferencias o variaciones entre sí dependiendo del modelo utilizado para normalizar los datos, motivo por el cual, primeramente, en el caso de los datos originales, comenzamos teniendo una media y mediana significativamente alejadas entre ellas (media=496.8 y mediana=191), junto con un sesgo medianamente significativo (1.53) y una curtosis significativamente elevada (5.79), lo cual indica que inicialmente la distribución de los datos es prácticamente asimétrica y con sesgo negativo (a la izquierda), motivo por el cual, tanto la media como la mediana se encuentran significativamente alejadas entre ellas, por tanto, inicialmnte los datos siguen una distribución que no es normal. No obstante, al normalizar los datos con el modelo aproximado, se puede observar que la media (2.294) y la mediana (2.281) de la distribución de los datos se encuentran mucho más cerca entre ellas a comparación del caso anterior, además de que en esta ocasión, el sesgo paso de ser positivo a ser negativo (la aglomeración de datos se desplazó hacia la derecha), sin embargo el sesgo sigue siendo medianamente significativo (-1.039), mientras que al mismo tiempo, la curtosis de los datos disminuyó de 5.79 a 4.33, sin embargo, 4.33 sigue siendo un valor considerablemente elevado de curtosis, por lo que la distribución de los datos normalizados con el modelo aproximado aún es asimétrica y por tanto no es normal. Por otro lado, al normalizar los datos con el modelo exacto, se obtiene una media de 11.008 y una mediana de 9.958, por lo que dichas medidas están medianamente cerca entre ellas, además para este caso, el sesgo de la distribución de los datos es de -0.029 mientras que la curtosis es de 2.3288, por lo que al normalizar con el modelo exacto, el sesgo es negativo pero muy cercano a 0, lo cual es deseable, mientras que la curtosis también disminuye a 2.3288, motivo por el cual, comparando las medidas de la distribución de los datos originales con las de la distribución de los datos normalizados mediante el modelo exacto, podemos apreciar que tanto la media como la mediana de los nuevos datos están mucho más cerca entre ellas de lo que estaban en el caso de los datos originales, mientras que también los nuevos datos normalizados tienen un sesgo mucho menor que los originales junto con un valor de curtosis medianamente más bajo que los datos iniciales, lo cual arrojará como resultado una distribución mucho más cercana a la normal en comparación con la distribución de los datos originales, aunque no totalmente normal, debido a que la medida de curtosis aún permanece medianamente alta (2.3288) y el sesgo aunque sea muy cercano a 0, no es 0 como tal.

Histograma del modelo exacto, aproximado y de los datos originales

```
# Graficar histograma de los datos normalizados (con ambos modelos) y los originales

par(mfrow = c(3, 1))

hist(sodio, col = "red", main = "Histograma de los datos originales")
hist(boxcox_aprox, col = "yellow", main = "Histograma del modelo aproximado",
     xlab ="sodio")
hist(boxcox_exacto, col = "green", main = "Histograma del modelo exacto",
xlab = "sodio")
```



En los histogramas anteriores se observa que en comparación con los datos originales que tienen una distribución bastante asimétrica, la distribución de los datos normalizados con el modelo aproximado resulta tener una cantidad notablemente mayor de simetría, aunque dicha distribución también tiene ciertas zonas asimétricas, ya que no todas las barras del histograma tienen la misma altura, sobre todo en los extremos del histograma, motivo por el cual, es posible considerar que al usar el modelo aproximado, se obtienen como resultado datos cuya distribución es mejor que la de los datos originales, pero que al mismo tiempo, la distribución de los datos resultantes de la normalización aún no se puede considerar como una distribución normal. Por otra parte, en cuanto al tercer histograma referente a los datos normalizados mediante el modelo exacto, es posible notar que la distribución de los datos resultantes tampoco se termina de ajustar a una distribución normal, sin

embargo, los datos se encuentran aglomerados tanto en el centro como en la región izquierda del gráfico, aunque la aglomeración es mayor en el centro del mismo, no obstante, los datos también se encuentran mejor distribuidos que en el caso de los datos originales, aunque también con ciertas asimetrías en ciertas regiones del tercer histograma en color verde, por lo que una de las razones que puede estar causando asimetría específicamente en el extremo izquierdo de los histogramas son los valores 0, por lo que al eliminarlos, posiblemente mejore aún más la distribución de los datos.

Test de normalidad de Anderson Darling para datos normalizados y originales

Prueba de hipótesis:

H_0: la variable sodio sigue una distribución normal.

H_1: la variable sodio no sigue una distribución normal.

```
# Librería para tests de normalidad

library(nortest)

# Hacer el test de normalidad de Anderson Darling de los datos originales

ad.test(sodio)
```

```
##
##  Anderson-Darling normality test
##
## data:  sodio
## A = 21.406, p-value < 2.2e-16
```

```
# Hacer el test de normalidad de Anderson Darling de los datos
transformados con el modelo
# aproximado

ad.test(boxcox_aprox)
```

```
##
##  Anderson-Darling normality test
##
## data:  boxcox_aprox
## A = 5.5342, p-value = 1.12e-13
```

```
# Hacer el test de normalidad de Anderson Darling de los datos
transformados con el modelo
# exacto

ad.test(boxcox_exacto)
```

```
##
##  Anderson-Darling normality test
##
## data:  boxcox_exacto
## A = 4.0199, p-value = 4.98e-10
```

Como se puede observar en los test de normalidad previos, tanto en el caso de los datos originales como de los datos transformados (con ambos modelos), el p valor resultante del test de Anderson Darling es mucho menor que 0.05, lo cual conduce a que en los 3 casos, se rechaze H_0, motivo por el cual, en los 3 casos y en general, se concluye que la variable sodio no sigue una distribución normal, por lo que en resumen, la normalización mediante Box-Cox, no resulta ser suficientemente efectiva para lograr que los datos de la variable sodio sigan una distribución normal.

4. Detección de anomalías y corrección de la base de datos

Como paso adicional antes de proceder a implementar el método de normalización de Yeo Johnson, primero verificamos que entre los datos de la variable sodio, no existan valores que sean 0.

```
# Verificar que no existan valores que sean 0 dentro del conjunto de
datos de sodio

sodio[sodio == 0]

## numeric(0)
```

Dado que no existen valores que sean 0 dentro del conjunto de datos de sodio, prácticamente los datos de dicha variable no requieren de ninguna otra modificación o corrección, debido a que los datos de la variable en cuestión no presentan ninguna otra inconsistencia, o error en su estructura, además, dado que el método de normalización de Yeo Johnson no requiere que los datos sean estrictamente valores positivos, tampoco resulta necesario hacer una traslación de los datos u otro cambio por el estilo, ya que el método de Yeo Johnson no se ve afectado negativamente por los datos que sean 0 o números negativos, por lo que dicho método puede trabajar prácticamente con datos que tengan cualquier valor ya sea 0, positivo, o negativo.

5. Transformación de Yeo Johnson

```
# Librería para implementar la normalización de Yeo Johnson

library(VGAM)

## Loading required package: stats4

## Loading required package: splines
```

```r
# Realizar la normalización de Yeo Johnson con base en el valor óptimo de lambda calculado
# al implementar Box-Cox

YJ = yeo.johnson(sodio, lambda = lambda_optimo) # Modelo exacto de Yeo Johnson

head(YJ)
```

```
## [1] 15.10523 15.21992 15.27641 15.70912 15.81239 16.20810
```

```r
# Modelo aproximado de Yeo Johnson

# Dado que el valor de lambda utilizado es 0.2222 y además es más cercano a 0 que a 0.5, se procede a utilizar el logaritmo base 10 de los datos para realizar la normalización

YJ_aprox = log(sodio, 10) # logaritmo base 10 de los datos de sodio

head(YJ_aprox)
```

```
## [1] 2.875640 2.887054 2.892651 2.935003 2.944976 2.982723
```

```r
# Test de normalidad de Anderson Darling para el caso de lambda = 0.22

ad.test(YJ)
```

```
##
##  Anderson-Darling normality test
##
## data:  YJ
## A = 4.1924, p-value = 1.904e-10
```

Ensayos con otros valores de lambda

Ensayo con Lambda = -2

```r
# Ensayo con lambda = -2

YJ_2 = yeo.johnson(sodio, lambda = -2)

# Prueba de normalidad de Anderson Darling

ad.test(YJ_2)
```

```
##
##  Anderson-Darling normality test
##
## data:  YJ_2
## A = 91.084, p-value < 2.2e-16
```

Ensayo con Lambda = 0.5

```
# Ensayo con lambda = 0.5

YJ_3 = yeo.johnson(sodio, lambda = 0.5)

# Prueba de normalidad de Anderson Darling

ad.test(YJ_3)
```

```
##
##  Anderson-Darling normality test
##
## data:  YJ_3
## A = 9.791, p-value < 2.2e-16
```

Ensayo con Lambda = 1

```
# Ensayo con lambda = 1

YJ_4 = yeo.johnson(sodio, lambda = 1)

# Prueba de normalidad de Anderson Darling

ad.test(YJ_4)
```

```
##
##  Anderson-Darling normality test
##
## data:  YJ_4
## A = 21.406, p-value < 2.2e-16
```

En resumen, dado que se observa que en los otros 3 ensayos llevados a cabo con diferentes valores de λ el valor p no es mayor que aquel obtenido de utilizar el valor de lambda óptimo calculado al implementar la transformación de Box-Cox, el valor de λ que resulta conveniente conservar es el óptimo calculado en el método de Box-Cox, esto debido a que al utilizar dicho valor de λ, se obtiene prácticamente el mayor valor p posible, específicamente 1.904e-10, en comparación con los 3 ensayos restantes que arrojaron valores p menores a 2.2e-16, motivo por el cual, nos quedamos con el valor de $\lambda = 0.2222222$ obtenido en el método de Box-Cox.

6. Ecuaciones de los modelos encontrados

Se utiliza el siguiente modelo exacto, dado que todos los datos en cuestión son positivos y además, el valor de λ utilizado es distinto de 0:

$$sodio_{normalizado} = \frac{(sodio_{actual} + 1)^{0.2222222} - 1}{0.2222222}$$

En cambio, la ecuación que simboliza al modelo aproximado, dado que el valor de λ es cercano a 0, es la siguiente:

$$sodio_{normalizado} = log(sodio_{actual})$$

7. Análisis de normalidad de las transformaciones de Yeo Johnson

Comparación de medidas estadísticas

```
# Calcular mínimo, máximo, media, mediana, cuartil 1 y cuartil 3 de los
datos transformados
# con el modelo aproximado de Yeo Johnson

summary(YJ_aprox)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.035   2.281   2.294   2.937   3.556
```

```
# Calcular sesgo

cat("Sesgo: ", skewness(YJ_aprox), "\n")
```

```
## Sesgo:  -1.039777
```

```
# Calcular curtosis

cat("Curtosis: ", kurtosis(YJ_aprox))
```

```
## Curtosis:  4.332194
```

```
# Calcular mínimo, máximo, media, mediana, cuartil 1 y cuartil 3 de los
datos transformados
# con el modelo exacto de Yeo Johnson

summary(YJ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7494  8.2748  9.9748 11.0606 15.7349 23.2687
```

```
# Calcular sesgo

cat("Sesgo: ", skewness(YJ), "\n")
```

```
## Sesgo:  0.03033371
```

```
# Calcular curtosis

cat("Curtosis: ", kurtosis(YJ))
```

```
## Curtosis:  2.236273
```

```r
# Calcular mínimo, máximo, media, mediana, cuartil 1 y cuartil 3 de los datos originales

summary(sodio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   108.5   191.0   496.8   866.0  3601.0
```

```r
# Calcular sesgo

cat("Sesgo: ", skewness(sodio), "\n")
```

```
## Sesgo:  1.535166
```

```r
# Calcular curtosis

cat("Curtosis: ", kurtosis(sodio))
```

```
## Curtosis:  5.796412
```

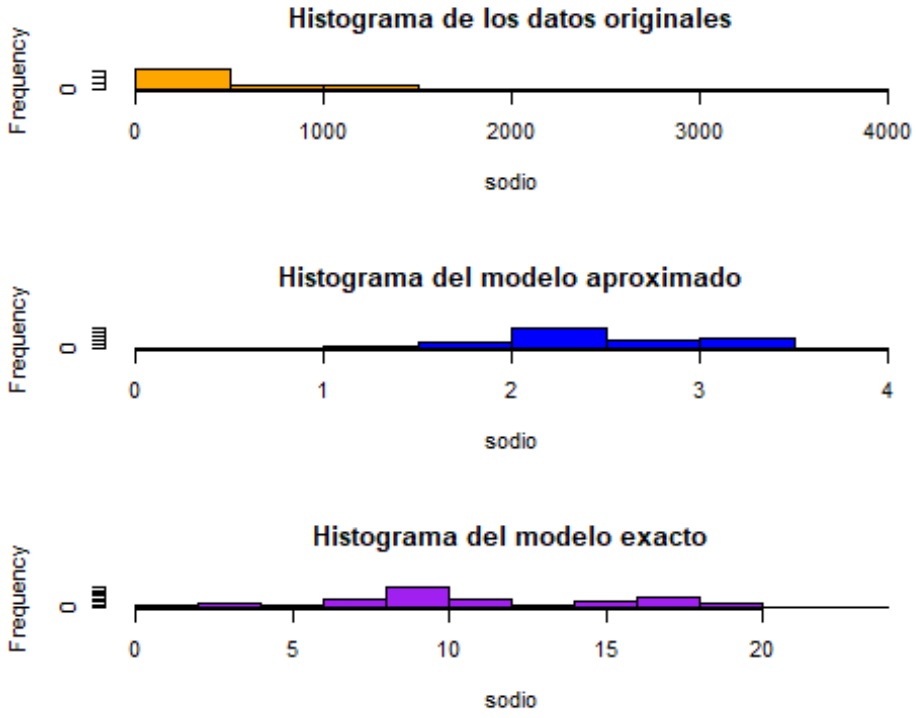
En términos generales, al momento de comparar las medidas estadísticas de los datos originales contra las de los datos transformados tanto con el modelo exacto como con el aproximado, es posible observar que en cuanto a los datos transformados con el modelo aproximado de Yeo Johnson, la media (2.294) y la mediana (2.281) están muy próximas entre sí, además el sesgo tiene un valor de -1.039, mientras que la curtosis posee un valor de 4.332, con lo cual, respecto a los valores de sesgo y curtosis de los datos originales, los de los datos transformados con el modelo aproximado presentan un sesgo medianamente significativo a la izquierda, además de una curtosis todavía elevada, por lo que la distribución de los datos resultantes del modelo aproximado de Yeo Johnson sigue siendo en su mayoría asimétrica, esto debido principalmente al sesgo que en este caso es negativo y a la curtosis que sigue siendo alta, por lo que prácticamente no se logra obtener una distribución normal (con sesgo y curtosis cercanos a 0) mediante el modelo aproximado de Yeo Johnson. Adicionalmente, en cuanto a las medidas estadísticas de los datos obtenidos con el modelo exacto de Yeo johnson, se observa que nuevamente la media (11.06) y la mediana (9.97) se encuentran próximas entre ellas en comparación con los datos originales, además de que también la distribución de los datos obtenidos a partir del modelo exacto, tiene una medida de sesgo de 0.03 y una curtosis de 2.23, por lo que el modelo exacto arroja datos con un sesgo significativamente menor que el de los datos originales y una curtosis también medianamente menor que la de los datos iniciales, por lo que hasta el momento, según el método de normalización de Yeo Johnson, las evidencias presentadas apuntan a que el modelo exacto es mejor que el aproximado para normalizar los datos de sodio, ya que se obtiene una distribución lo más parecida posible a la distribución normal, aunque en realidad no sea normal en sí misma.

Histograma de los modelos y de los datos originales

```r
# Graficar histograma de los datos normalizados (con ambos modelos) y los originales
```

```r
par(mfrow = c(3, 1))

hist(sodio, col = "orange", main = "Histograma de los datos originales")
hist(YJ_aprox, col = "blue", main = "Histograma del modelo aproximado",
     xlab ="sodio")
hist(YJ, col = "purple", main = "Histograma del modelo exacto", xlab =
"sodio")
```



En los histogramas anteriores se aprecia en general que la distribución correspondiente al modelo aproximado se asemeja ligeramente más a la forma que tendría una distribución normal, sin embargo, dicha distribución no es normal debido principalmente a la asimetría que resulta muy evidente a lo largo de toda la distribución de los datos, pero más en la región izquierda del histograma, por lo cual, los datos no se encuentran distribuidos de forma simétrica, con lo que ya no se cumplen las principales características de una distribución normal y por ello, el comportamiento de los datos resultantes del modelo aproximado no sigue una distribución normal. Además de lo anterior, en cuanto al histograma del modelo exacto, se aprecia que éste mismo tampoco es completamente simétrico, ya que el histograma presenta asimetría a lo largo de toda la distribución, sin embargo, dicho histograma también presenta una aparente simetría en 2 pequeñas regiones (donde las barras tienen mayor altura), pero dicha simetría no se generaliza a toda la distribución en general, motivo por el cual, se puede afirmar que el modelo exacto de Yoe Johnson no produce datos cuyo comportamiento siga una distribución normal.

Test de normalidad de Anderson Darling para los datos originales y transformados

Prueba de hipótesis:

H_0: la variable sodio sigue una distribución normal.

H_1: la variable sodio no sigue una distribución normal.

```
# Prueba de normalidad de Anderson Darling para datos originales

ad.test(sodio)

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

# Prueba de Anderson Darling para datos transformados con el modelo
exacto

ad.test(YJ)

##
##  Anderson-Darling normality test
##
## data:  YJ
## A = 4.1924, p-value = 1.904e-10

# Prueba de Anderson Darling para datos transformados con el modelo
aproximado

ad.test(YJ_aprox)

##
##  Anderson-Darling normality test
##
## data:  YJ_aprox
## A = 5.5342, p-value = 1.12e-13
```

De acuerdo con los tests de normalidad de Anderson Darling realizados previamente, se puede apreciar que el test que arroja el valor p más grande es la aplicada a los datos resultantes del modelo de Yoe Johnson exacto, siendo éste valor de 1.904e-10, mientras que el resto de los tests arrojaron valores p de 1.12e-13 y menores a 2.2e-16, motivo por el cual, en los 3 casos, el valor p es menor a 0.05, por lo tanto, se rechaza H_0 y como consecuencia, se concluye que de acuerdo al test de Anderson Darling, el comportamiento tanto de los datos originales como de los transformados por el método de Yoe Johnson, no sigue una distribución normal.

8. Mejor transformación de los datos

Considerando todo lo anterior, es posible afirmar que el "mejor" modelo obtenido por medio de la normalización de Box-Cox es el modelo exacto de Box-Cox, mismo que arroja un p valor de 4.98e-10 en el test de normalidad de Anderson Darling y que además tiene un sesgo de -0.02985208 y una curtosis de 2.328831, los cuales son los valores más pequeños posibles de sesgo y curtosis de acuerdo a con lo normalización de Box-Cox, no obstante, en cuanto al método de normalización de Yoe Johnson, la mejor transformación obtenida es la derivada del modelo exacto de Yoe Johnson y que arroja un p valor de 1.904e-10 en la prueba de normalidad de Anderson Darling y además su sesgo es de 0.03033371 y su curtosis es de 2.236273, mismos que son los valores mínimos de sesgo y de curtosis que fue posible obtener empleando la normalización de Yoe Johnson tanto exacta como aproximada, motivo por el cual, comparando las medidas estadísticas junto con los valores p de éstos 2 "mejores" modelos mencionados anteriormente, se concluye que el mejor modelo de normalización para los datos de sodio es el modelo de Box-Cox exacto, principalmente debido a que a pesar de que el sesgo de dicho modelo (-0.0298) es aproximadamente igual al del modelo exacto de Yoe Johnson (0.03) y además la curtosis tampoco varía mucho de un modelo al otro (2.32 del modelo exacto de Box-Cox y 2.23 del modelo exacto de Yoe Johnson), sin embargo, lo que si cambia de forma más significativa es el valor p de cada modelo en la prueba de normalidad de Anderson Darling, siendo el modelo exacto de Box-Cox aquel que tiene el p valor más grande de ambos modelos mencionados (4.98e-10 de Box-Cox exacto contra 1.904e-10 de Yoe Johnson exacto), por lo que principalmente de acuerdo al p valor, el modelo exacto de Box-Cox es el que se encuentra relativamente más cerca de pasar normalidad, aunque en realidad, su p valor no sobrepasa 0.05 para que los datos sean oficialmente normales, sin embargo, es la aproximación más "cercana" hasta el momento a lo que sería una distribución normal, por lo que en conclusión, el "mejor" modelo de transformación es el modelo exacto de Box-Cox.

9. Ventajas y desventajas de Box-Cox y Yeo Johnson

Ventajas y desventajas de Box-Cox

Después de todo el análisis realizado, en primer lugar, en cuanto al método de Box-Cox, entre sus principales ventajas se encuentra el hecho de que es computacionalmente más rápido de calcular que el de Yoe Johnson, debido a que los elementos presentes en la función del método de Box-Cox son sencillos de calcular, por lo que en caso de llevar a cabo varios ensayos o iteraciones que involucren al modelo de Box-Cox, éstas se ejecutarán en una menor cantidad de tiempo. Además, otra de las ventajas de utlizar el método de normalización de Box-Cox radica en que para implementarlo no es necesario implementar algún otro método antes a comparación de Yoe Johnson que requiere del valor óptimo de λ para poderse calcular, para lo cual, es necesario implementar primero el método de Box-Cox para calcular el valor óptimo de λ para luego en base a ese valor, realizar los cálculos

pertinentes con Yoe Johnson, es decir, que básicamente Yoe Johnson depende de implementar Box-Cox antes para poderlo implementar. No obstante, entre algunas de las desventajas de usar Box-Cox se encuentra el hecho de que éste método no puede operar con datos que sean negativos, o iguales a 0, por lo que de ser ese el caso, se requiere aplicar una ligera traslación a los datos para quitar los datos que sean 0 o negativos, antes de aplicar el método, a diferencia de Yoe Johnson que puede operar prácticamente con cualquier valor, ya sea positivo, negativo, o igual a 0 y sin la restricción de que los datos originales sean estrictamente mayores que 0 (valores positivos) y también sin tener que hacer ninguna modificación a los datos antes de aplicar el método.

Ventajas y desventajas de Yoe Johnson

Por otra parte, en cuanto al método de normalización de Yoe Johnson, una de sus ventajas consiste en que a diferencia del método de Box-Cox, Yoe Johnson no requiere que los datos sean estrictamente positivos, sino que puede operar con todo tipo de valores numéricos incluyendo negativos y el 0, y sin tener que aplicar alguna modificación a los datos antes de aplicar éste método, lo cual permite ahorrar tiempo que sería de otra manera invertido en adecuar los datos en caso de usar Box-Cox y que los datos originales fueran algunos negativos, o iguales a 0. En cambio, entre las desventajas del método de Yoe Johnson se encuentra el hecho de que éste método a pesar de poder trabajar con todo tipo de números, necesita del valor de λ calculado mediante Box-Cox para funcionar, lo cual implica gastar tiempo implementando antes el método de Box-Cox con el objetivo de calcular el valor óptimo de λ para después todavía implementar Yoe Johnson, cuando el método de Box-Cox nos permite ahorrar ese tiempo, al no requerir ningún parámetro que deba ser calculado a partir de otra metodología, sino que en base a los datos que se tienen, se verifica primero si se tienen datos negativos o iguales a 0, y de ser así, se realiza la traslación pertinente de los datos y en caso contrario, se aplica directamente el método sobre los datos originales, por lo que éstas 2 alternativas demandan menos tiempo que implementar primero una metodología completa y acto seguido implementar una segunda que requiera elementos de la primera.

10. Diferencias entre transformación y escalamiento de los datos

Diferencias entre transformación y escalamiento

1. Una transformación tiene como propósito principal modificar el tipo de distribución que siguen los datos, mientras que el escalamiento se enfoca en modificar la escala de los datos para poder comparar fácilmente datos que tienen originalmente escalas significativamente diferentes, pero sin cambiar la distribución de los datos en cuestión.

2. El escalamiento por lo general facilita la comprensión e interpretación de los datos, al dejarlos todos en la misma escala, mientras que por el contrario, una transformación deja nuevos datos que generalmente son difíciles de

interpretar en el contexto del problema que se esté resolviendo, dificultando de esa manera la interpretación de todo el análisis de datos.

3. El escalamiento siempre resulta efectivo para modificar la escala de las variables y así poder compararlas con mayor facilidad, mientras que al contrario, una transformación no siempre resulta efectiva, dado que en ciertos casos, debido a la propia naturaleza de los datos u otro tipo de cuestiones subyacentes de los mismos, a pesar de implementar correctamente los métodos de normalización, no se consigue que los nuevos datos resultantes sigan una distribución normal, mientras que en otros casos, sí se logra que los datos transformados sigan una distribución normal.

¿Cuándo es necesario utilizar escalamiento y transformación?

Escalamiento: se utiliza principalmente cuando se requiere comparar de forma sencilla datos de variables que originalmente se encuentran en escalas numéricas significativamente distintas entre sí, pero sin que eso afecte, o modifique el comportamiento o distribución original de los datos.

Transformación: se emplea cuando se requiere que los datos originales sigan una distribución en específico, por lo general una distribución normal (normalización), lo cual implica que se necesita modificar incluso completamente el comportamiento original de los datos de tal manera que la versión transformada de los mismos siga la distribución deseada.