act_5_A01742161

Rogelio Lizárraga

2024-08-14

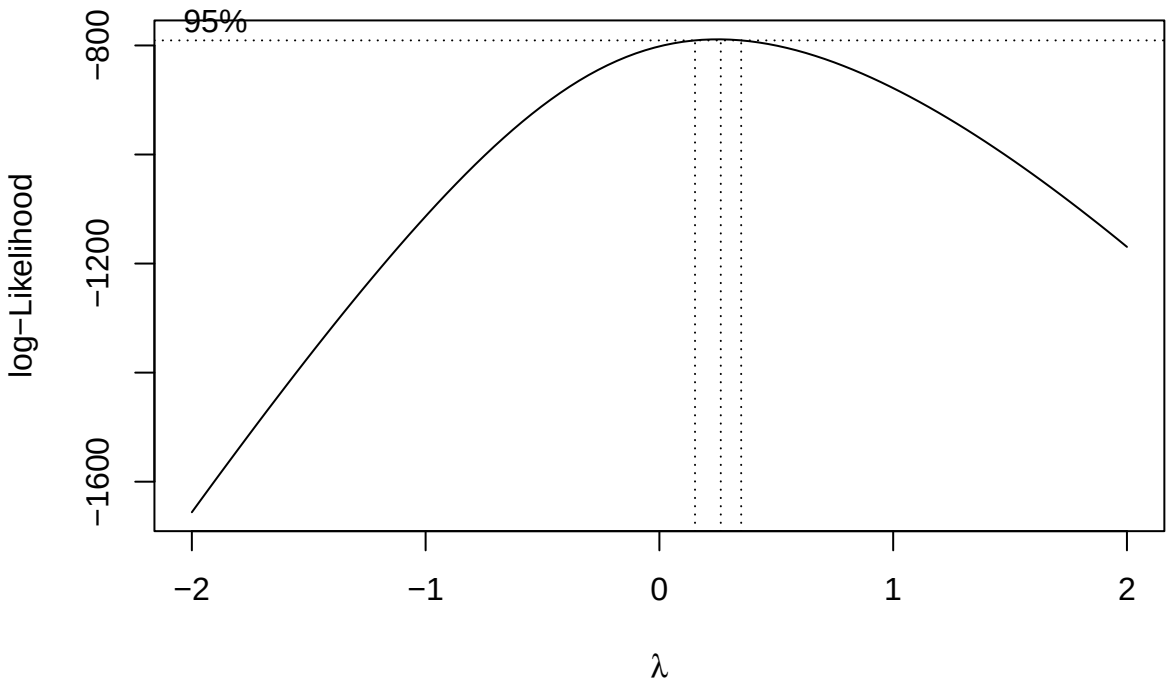
Selecciona una variable, que no sea Calorías, y encuentra la mejor transformación de datos posible para que la variable seleccionada se comporte como una distribución Normal.

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

```
##   [1]   3   3   2   2   2   3   3   4   3   4   2   3   2   3   3   3   3   4
##  [19]   3  15  16  15  15  15   7   8   7   3   3   3   4  17  17  17  18  14
##  [37]  14   2   0  32  32  18   9  10  12  10   9  10   6   7   7  14   7   7
##  [55]   7   6  11  10   8  11   9  11   9  16  14   7   5   6   6   5   7   6
##  [73]   8   6  12  10  14  12   0   0   0   0   1   5   4   5   4   6  12  10
##  [91]   8   7   3   2   3   2   0   0   0   0   2   3  23  13  15  13   6  48
## [109]  43  45  39  55  76  28   0   0   0   0  35  51  70  26   0   0   0   0
## [127]  37  54  74  27  12  22  19  30  39  58   0   0   0   0   0  36  45  54
## [145]  27   0   0   0  12  15  20  38  48  59  38  47  58  36  45  56  12  15
## [163]  20  13  16  21  39  48  59  38  48  59  37  46  56  13  16  21  42  53
## [181]  63  43  53  64  40  50  60  41  51  61  45  56  68  46  57  69  22  30
## [199]  45  21  28  42  20  28  41  19  26  39   1   2   2  34  43  62  35  43
## [217]  62  33  41  59  33  41  59  57  71  88  57  71  88  67  81  99  44  54
## [235]  70  44  54  70  46  56  72  63  81 101  79 100 123  77  97 120  93 115
## [253]  89 128  59  64  85  43 103  51
```

1 . 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)
exacto<-boxcox((Sugar + 1)~1)
```




```
l_exacto =exacto$x[which.max(exacto$y)]

aproximado = sqrt(Sugar + 1)
l_exacto
```

```
## [1] 0.2626263
```

```
sugar_1 = aproximado
sugar_2 = ((Sugar + 1)^l_exacto-1)/l_exacto
```

2. Escribe las ecuaciones de los modelos encontrados.

Modelo aproximado =

$$\sqrt{x+1}$$

Modelo exacto =

$$\frac{(x+1)^{0.26}-1}{0.26}$$

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

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

```
library(e1071)
cat('Datos originales \n')
```

```
## Datos originales
```

```
summary(Sugar)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.75   17.50   29.42   48.00  128.00
```

```
print("Curtosis")
```

```
## [1] "Curtosis"
```

```
kurtosis(Sugar)
```

```
## [1] 0.460967
```

```
print("Sesgo")
```

```
## [1] "Sesgo"
```

```
skewness(Sugar)
```

```
## [1] 1.020064
```

```
cat("Rango medio:", IQR(Sugar))
```

```
## Rango medio: 42.25
```

```
cat('\nModelo aproximado \n')
```

```
##
## Modelo aproximado
```

```
summary(sugar_1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.597   4.301   4.825   7.000  11.358
```

```r
print("Curtosis")
```

```
## [1] "Curtosis"
```

```r
kurtosis(sugar_1)
```

```
## [1] -1.01447
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(sugar_1)
```

```
## [1] 0.2794263
```

```r
cat("Rango medio:", IQR(sugar_1))
```

```
## Rango medio: 4.403314
```

```r
cat('\nModelo exacto \n')
```

```
##
## Modelo exacto
```

```r
summary(sugar_2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.477   4.385   4.519   6.774   9.837
```

```r
print("Curtosis")
```

```
## [1] "Curtosis"
```

```r
kurtosis(sugar_2)
```

```
## [1] -1.113324
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(sugar_2)
```

```
## [1] -0.1056929
```

```r
cat("Rango medio:", IQR(sugar_2))
```

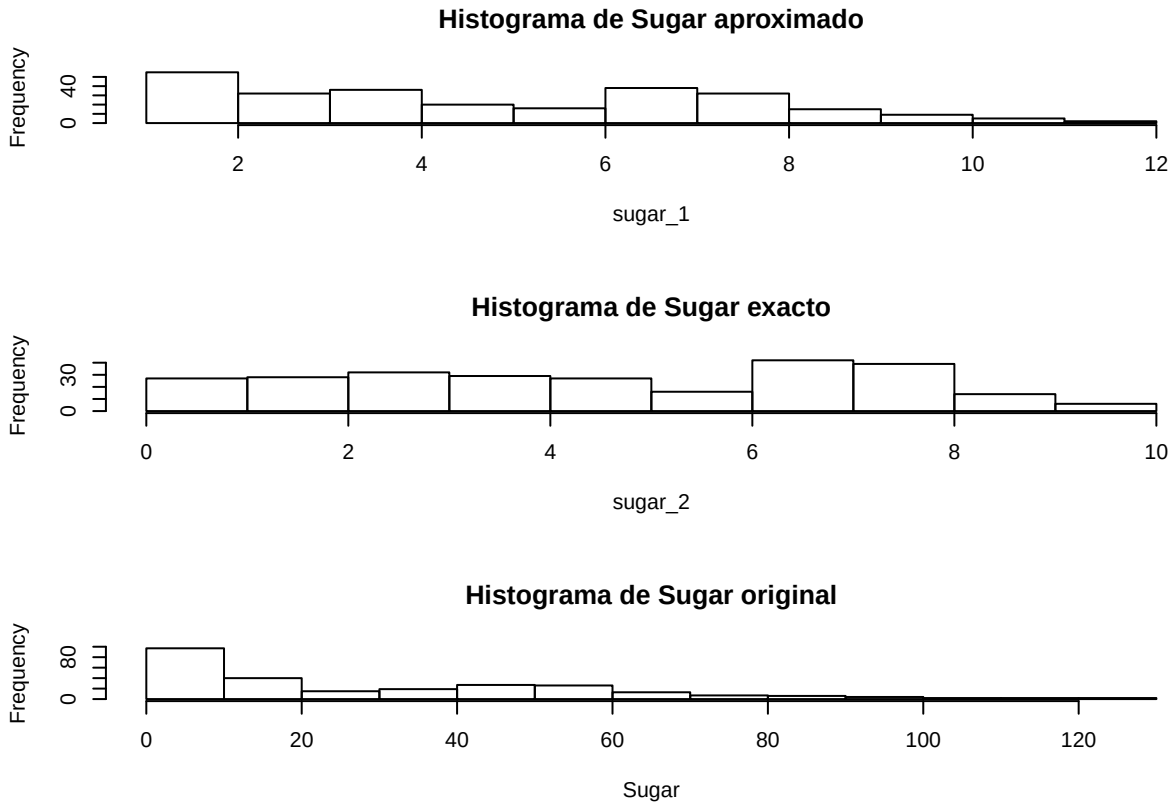
```
## Rango medio: 4.29701
```

Para los datos originales, la media, la mediana y el rango medio están muy alejados, por lo que pareciera que no es normal. Se observa una curtosis muy elevada y un sesgo alto. Para el modelo aproximado, observamos que la media, la mediana y el rango medio están mucho más cercanos entre sí y los cuartiles tienen distancias relativamente aceptables a la mediana. Observamos que la curtosis está muy elevada y el sesgo está ligeramente elevado, por lo que pareciera que no es normal.

Para el modelo exacto, observamos que la media, la mediana y el rango medio también están muy cercanos entre sí y los cuartiles tienen distancias relativamente aceptables a la mediana. Observamos que el sesgo es relativamente bajo, pero la curtosis está muy elevada, por lo que pareciera que no es normal.

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

```r
par(mfrow=c(3,1))
hist(sugar_1,col=0,main="Histograma de Sugar aproximado")
hist(sugar_2,col=0,main="Histograma de Sugar exacto")
hist(Sugar,col=0,main="Histograma de Sugar original")
```



En los histogramas, no se observa una distribución normal, pues se tiene un sesgo hacia la derecha en los ceros en el modelo aproximado y los datos originales. Para el modelo exacto, se observa una distribución muy plana, pues su curtosis es altamente negativa (platicúrtica).

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

H_0 = el conjunto de datos se distribuye como una normal. H_1 = el conjunto de datos no se distribuye como una normal.

```
library(nortest)
x_1 = ad.test(sugar_1)$p.value
x_2 = ad.test(sugar_2)$p.value
x_3 = ad.test(Sugar)$p.value
cat('Valor p modelo aproximado:', x_1, '\n')
```

```
## Valor p modelo aproximado: 3.531062e-10
```

```
cat('Valor p modelo exacto:', x_2, '\n')
```

```
## Valor p modelo exacto: 1.857266e-08
```

```
cat('Valor p datos originales:', x_3, '\n')
```

```
## Valor p datos originales: 3.7e-24
```

Como todos los valores $p < 0.05$, se rechaza H_0 para todos los conjuntos de datos, por lo que estos no se distribuyen de manera normal.

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

```
library(dplyr)
```

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

## The following object is masked from 'package:MASS':
##
##     select

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

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

```
M2 <- M %>%
filter(!(Sugars == 0 & grepl("(?i)diet coke|iced tea|coffee|diet dr pepper|dasani water", M$Item)))
head(M2)
```

```
##    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
```

Eliminaremos los ceros de datos que no tienen relevancia, como refrescos sin azúcar y botellas de agua.

```r
Sugar_fixed = M2$Sugars
summary(Sugar_fixed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.00   21.00   31.35   51.00  128.00
```

5. 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).

```r
library(VGAM)
```

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

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

```r
sugar_yeo<- yeo.johnson(Sugar_fixed, lambda = l_exacto)
aprox_yeo<- sqrt(sugar_yeo + 1)
```

6. Escribe la ecuación del modelo encontrado.

Modelo Yeo-Johnson exacto $=$

$$\frac{(x+1)^{0.26}-1}{0.26}$$

Modelo Yeo-Johnson aproximado $=$

$$\sqrt{x+1}$$

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

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

```r
library(e1071)
cat('Datos originales \n')
```

```
## Datos originales
```

```r
summary(Sugar)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.75   17.50   29.42   48.00  128.00
```

```r
print("Curtosis")
```

```
## [1] "Curtosis"
```

```r
kurtosis(Sugar)
```

```
## [1] 0.460967
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(Sugar)
```

```
## [1] 1.020064
```

```r
cat("Rango medio:", IQR(Sugar))
```

```
## Rango medio: 42.25
```

```r
cat('\nModelo exacto con limpieza \n')
```

```
##
## Modelo exacto con limpieza
```

```r
summary(sugar_yeo)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.766   4.767   4.815   6.940   9.837
```

```r
print("Curtosis")
```

```
## [1] "Curtosis"
```

```r
kurtosis(sugar_yeo)
```

```
## [1] -1.071926
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(sugar_yeo)
```

```
## [1] -0.1006817
```

```r
cat("Rango medio:", IQR(sugar_yeo))
```

```
## Rango medio: 4.173932
```

```r
cat('\nModelo aproximado con limpieza \n')
```

```
##
## Modelo aproximado con limpieza
```

```r
summary(aprox_yeo)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.941   2.401   2.347   2.818   3.292
```

```r
print("Curtosis")
```

```
## [1] "Curtosis"
```

```r
kurtosis(aprox_yeo)
```

```
## [1] -0.5814327
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(aprox_yeo)
```

```
## [1] -0.5077639
```

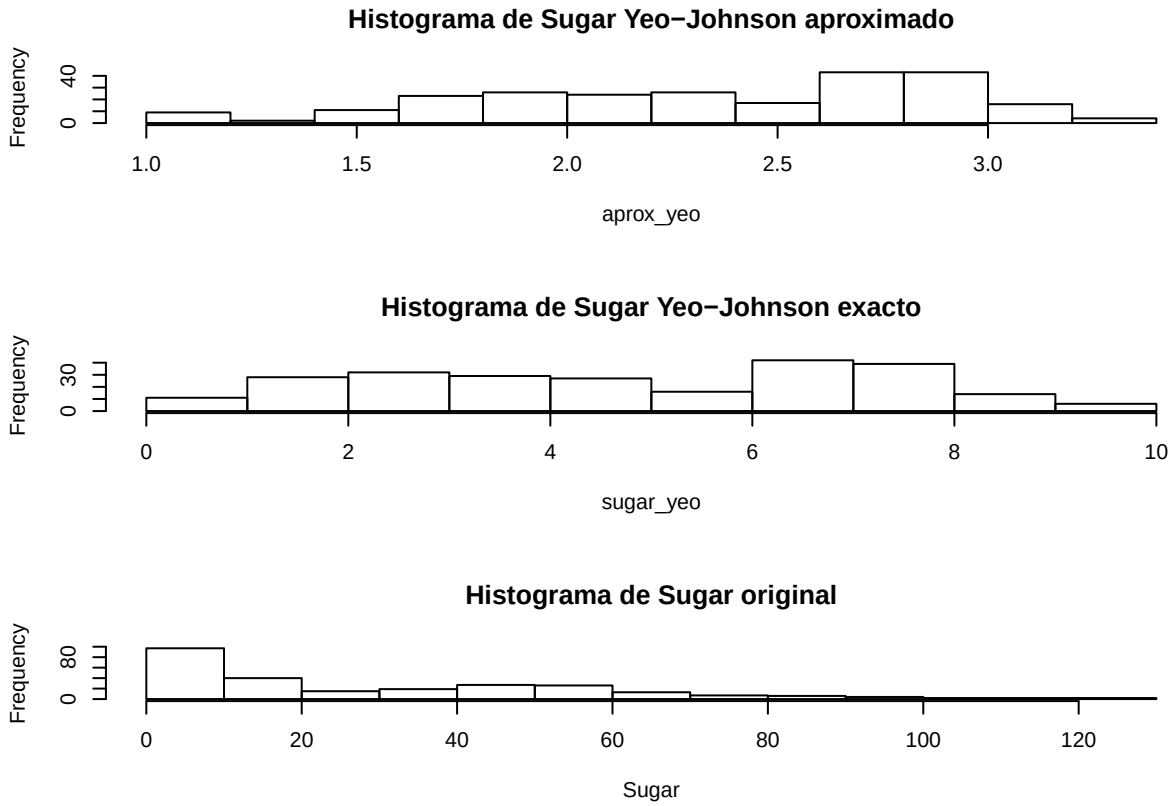
```r
cat("Rango medio:", IQR(aprox_yeo))
```

```
## Rango medio: 0.8771366
```

Como habíamos dicho anteriormente, en los datos originales, la media, la mediana y el rango medio están muy alejados, por lo que pareciera que no es normal. Se observa una curtosis muy elevada y un sesgo alto. Observamos que en la transformación Yeo-Johnson exacta, la media, la mediana están cercanos, pero el rango medio está un poco alejado. Además, vemos que contamos con un sesgo bajo, pero una curtosis muy elevada, por lo que pareciera que los datos transformados no se distribuyen como una normal. Ohservamos que en la transformación Yeo-Johnson aproximada, la media, la mediana están cercanos, pero el rango medio está muy alejado. Además, vemos que contamos con un sesgo bastante alto y una curtosis muy elevada, por lo que pareciera que los datos transformados no se distribuyen como una normal.

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

```r
par(mfrow=c(3,1))
hist(aprox_yeo ,col=0,main="Histograma de Sugar Yeo-Johnson aproximado")
hist(sugar_yeo ,col=0,main="Histograma de Sugar Yeo-Johnson exacto")
hist(Sugar,col=0,main="Histograma de Sugar original")
```



En los histogramas, no se observa una distribución normal, pues se tiene un sesgo hacia la derecha los datos originales. Para el modelo exacto y aproximado, se observa una distribución muy desequilibrada y anormal, debido a sus niveles de sesgo y curtosis.

c. Realiza la prueba de normalidad de Anderson-Darling para los datos transformados y los originales

H_0 = el conjunto de datos se distribuye de manera normal. H_1 = el conjunto de datos NO se distribuye de manera normal.

```r
library(nortest)
yeo_1 = ad.test(aprox_yeo)$p.value
yeo_2 = ad.test(sugar_yeo)$p.value
yeo_3 = ad.test(Sugar)$p.value
cat('Valor p modelo Yeo-Johnson aproximado:', yeo_1, '\n')
```

```
## Valor p modelo Yeo-Johnson aproximado: 2.098766e-09
```

```r
cat('Valor p modelo Yeo- Johnsons exacto:', yeo_2, '\n')
```

```
## Valor p modelo Yeo- Johnsons exacto: 1.169923e-07
```

```r
cat('Valor p datos originales:', yeo_3, '\n')
```

```
## Valor p datos originales: 3.7e-24
```

Como todos los valores $p < 0.05$, se rechaza H_0 para todos los conjuntos de datos, por lo que estos NO se distribuyen de manera normal.

8 . 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.

Como podemos observar en los valores p, ningún modelo es aceptable, pues estos valores son muy pequeños. Además, en todos la curtosis afecta de manera significativa y el sesgo afecta en la mayoría. Es decir, los datos NO se distribuyen de manera normal para ningún modelo, ni los datos originales. Sin embargo, si tuviera que escoger uno, sería el modelo de Yeo-Johnson exacto, pues su valor p es el menos peor de todos los encontrados y el modelo realizado no es tan complejo de elaborar. Es decir, es un equilibrio entre complejidad leve y la mejora del modelo.

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

Las ventajas de Box-Cox es que puede estabilizar la varianza en los datos, ayuda a hacer los datos más normalmente distribuidos (más cercanos a una distribución normal que los datos originales), tiene una amplia aplicación en estadística y puede llegar a mejorar el modelo.

Las desventajas de Box-Cox es que solo se puede usar para datos positivos (será necesario convertirlos a positivos), no siempre llega a exitosamente normalizar la distribución de los datos (como fue nuestro caso), es relativamente complejo de interpretar y no puede trabajar con datos discretos.

Las ventajas de Yeo-Johnson es que puede trabajar con valores positivos puede estabilizar la varianza en los datos, ayuda a hacer los datos más normalmente distribuidos (más cercanos a una distribución normal que los datos originales), tiene una amplia aplicación en estadística y puede llegar a mejorar el modelo.

Sus desventajas es que Yeo-Johnson, en algunos casos, no transforma exitosamente los datos a una distribución normal, los datos transformados pueden ser difíciles de interpretar, puede llegar a ser computacionalmente costoso por los procesos iterativos para encontrar lambda y no puede trabajar con datos discretos.

En nuestro caso, los datos no son negativos, pero sí contamos ceros, por lo que tenemos que recorrer uno los valores para poder utilizar Box-Cox. Sin embargo, ninguno de los dos modelos transformaron exitosamente los datos, pues no se logró normalidad.

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

La transformación y el escalamiento tienen distintos propósitos, pues la transformación busca cambiar la distribución de los datos, mientras que el escalamiento solo busca ajustar las magnitudes a escalas similares.

La transformación modifica la forma de la distribución de los datos, lo cual se usa cuando se busca linealidad, normalidad, entre otras, mientras que el escalamiento no modifica las distribuciones.

La transformación utiliza distintas técnicas que el escalamiento, pues la transformación involucra técnicas, tales como Box-Cox y Yeo-Johnson, mientras que el escalamiento solo involucra técnicas que ajusten los valores, como el min-max scaler, que escala los datos a valores entre 0 y 1.

De esta manera, la transformación se utiliza cuando los datos no siguen una distribución normal y el modelo a implementar lo requiere, mientras que el escalamiento es esencial cuando el modelo o los algoritmos a implementar requieren los valores en escalas similares (como puede ser el uso de "weights").