Act5 Transformaciones Luis Maximiliano Lopez Ramirez

Luis Maximiliano López Ramírez

2024-08-14

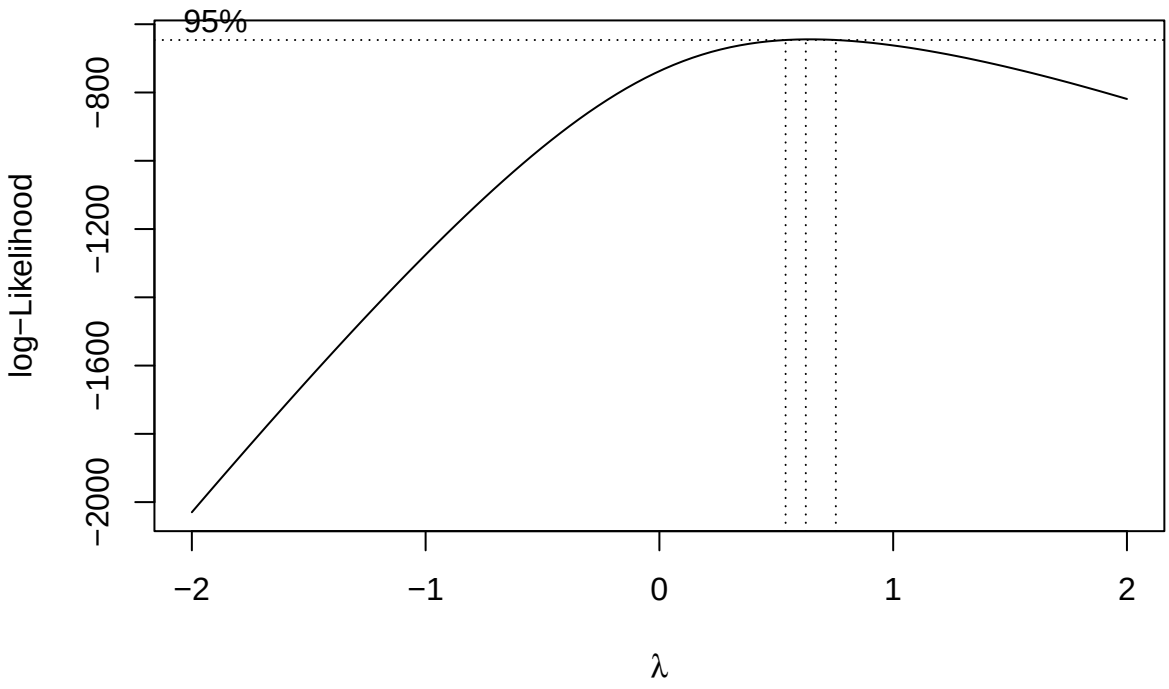
```
# Especificar el nuevo directorio
nuevo_directorio <- "C:/Users/luism/Escritorio/Documetos_2/Actividades Concentración"

# Cambiar al nuevo directorio
setwd(nuevo_directorio)
```

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:

```
# Carga los datos desde el archivo datosRes.csv
datos <- read.csv("mc-donalds-menu.csv")
X <- datos$Carbohydrates
```

 # 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)
bc<-boxcox((X+1)~1)
```




```
l=bc$x[which.max(bc$y)]
```

```
l
```

```
## [1] 0.6262626
```

 # Escribe las ecuaciones de los modelos encontrados.

Modelo aproximado $X_1 = \sqrt{X+1}$

Modelo exacto $X_2 = \frac{(X+1)^{0.6262626}-1}{0.6262626}$

 # 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.

```
library(e1071)  # Para sesgo y curtosis

X1=sqrt(X+1)
X2=((X+1)^l-1)/l

# Función para calcular las medidas
```

```r
calculate_stats <- function(x) {
  c(
    Min = min(x, na.rm = TRUE),
    Max = max(x, na.rm = TRUE),
    Mean = mean(x, na.rm = TRUE),
    Median = median(x, na.rm = TRUE),
    Q1 = quantile(x, 0.25, na.rm = TRUE),
    Q3 = quantile(x, 0.75, na.rm = TRUE),
    Skewness = skewness(x, na.rm = TRUE),
    Kurtosis = kurtosis(x, na.rm = TRUE)
  )
}

# Aplicar la función a cada variable
stats_X <- calculate_stats(X)
stats_X1 <- calculate_stats(X1)
stats_X2 <- calculate_stats(X2)

# Mostrar los resultados
print("Estadísticas con datos originales:")
```

```
## [1] "Estadísticas con datos originales:"
```

```r
print(stats_X)
```

```
##         Min         Max        Mean      Median      Q1.25%      Q3.75%
##   0.0000000 141.0000000  47.3461538  44.0000000  30.0000000  60.0000000
##    Skewness    Kurtosis
##   0.9021952   1.3240829
```

```r
print("Estadísticas con trasnformacion aproximada:")
```

```
## [1] "Estadísticas con trasnformacion aproximada:"
```

```r
print(stats_X1)
```

```
##        Min        Max       Mean     Median     Q1.25%     Q3.75%   Skewness
##  1.0000000 11.9163753  6.5832436  6.7082039  5.5677644  7.8102497 -0.4939626
##   Kurtosis
##  0.9092300
```

```r
print("Estadísticas con transformacion exacta")
```

```
## [1] "Estadísticas con transformacion exacta"
```

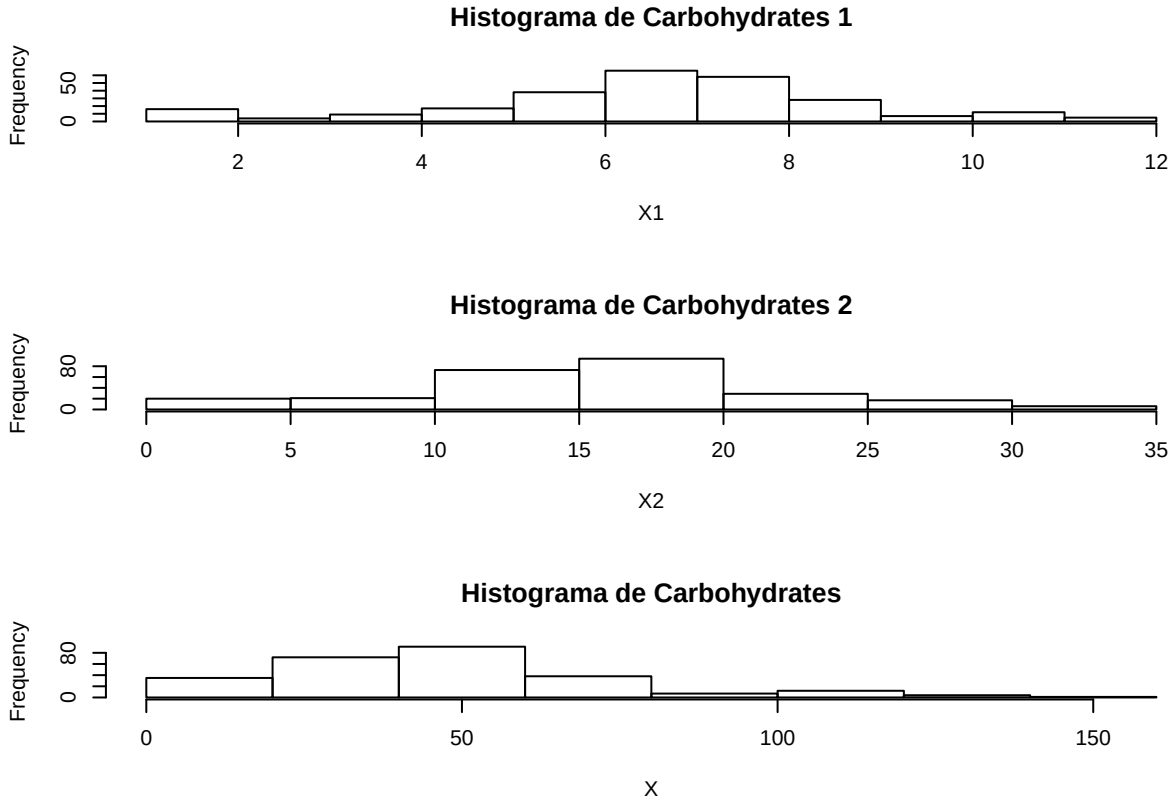
```r
print(stats_X2)
```

```
##         Min         Max        Mean      Median      Q1.25%      Q3.75%
##  0.00000000 33.97792660 15.66876584 15.72485414 12.11922959 19.36021340
##    Skewness    Kurtosis
## -0.08250202  0.63819744
```

Viendo los valores de sesgo y curtosis, parece ser que el modelo exacto de transformación es el mejor para conseguir la normalidad ya que tiene menores valores en dichas características.

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

```
par(mfrow=c(3,1))
hist(X1,col=0,main="Histograma de Carbohydrates 1")
hist(X2,col=0,main="Histograma de Carbohydrates 2")
hist(X,col=0,main="Histograma de Carbohydrates")
```



El primer histograma que corresponde a la transformacón aproximada si corrige bastante los datos originales a partir del tercer histograma pero sigue teniendo las colas altas sin embargo, el segundo histograma con la transformación exacta ya corrige las colas del primer histograma.

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

```
library(nortest)

# Prueba de normalidad de Anderson-Darling para cada variable
ad_test_X <- ad.test(X)
```

```r
ad_test_X1 <- ad.test(X1)
ad_test_X2 <- ad.test(X2)

# Mostrar los resultados
print("Prueba de Anderson-Darling con datos originales:")
```

```
## [1] "Prueba de Anderson-Darling con datos originales:"
```

```r
print(ad_test_X)
```

```
##
##  Anderson-Darling normality test
##
## data:  X
## A = 4.1402, p-value = 2.547e-10
```

```r
print("Prueba de Anderson-Darling con transformacion aproximada:")
```

```
## [1] "Prueba de Anderson-Darling con transformacion aproximada:"
```

```r
print(ad_test_X1)
```

```
##
##  Anderson-Darling normality test
##
## data:  X1
## A = 4.4524, p-value = 4.482e-11
```

```r
print("Prueba de Anderson-Darling con transformacion exacta:")
```

```
## [1] "Prueba de Anderson-Darling con transformacion exacta:"
```

```r
print(ad_test_X2)
```

```
##
##  Anderson-Darling normality test
##
## data:  X2
## A = 3.1076, p-value = 8.182e-08
```

La función ad.test devuelve el valor del estadístico de prueba y el valor p. Un valor p bajo (generalmente menor a 0.05) indica que los datos no siguen una distribución normal aún con las transformaciones.

 # Detecta anomalías y corrige tu base de datos (datos atípicos, ceros anámalos, etc).

```r
remove_outliers <- function(x) {
  mean_x <- mean(x, na.rm = TRUE)
  sd_x <- sd(x, na.rm = TRUE)
  lower_bound <- mean_x - 3 * sd_x
```

```
  upper_bound <- mean_x + 3 * sd_x

  # Identificar los valores atípicos
  outliers <- x[x < lower_bound | x > upper_bound]
  if (length(outliers) > 0) {
    cat("Valores atípicos en el vector:", outliers, "\n")
  } else {
    cat("No se encontraron valores atípicos.\n")
  }

  # Remover los valores atípicos
  x_clean <- x[x >= lower_bound & x <= upper_bound]
  return(x_clean)
}

# Identificar y remover valores atípicos
X_clean <- remove_outliers(X)
```

```
## Valores atípicos en el vector: 135 140 141 135 139
```

```
X1_clean <- remove_outliers(X1)
```

```
## No se encontraron valores atípicos.
```

```
X2_clean <- remove_outliers(X2)
```

```
## No se encontraron valores atípicos.
```

Se puede observar que no había valores atípicos para las trasnformaciones tatno aproximada y exacta pero si para los datos originales.

 # 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).

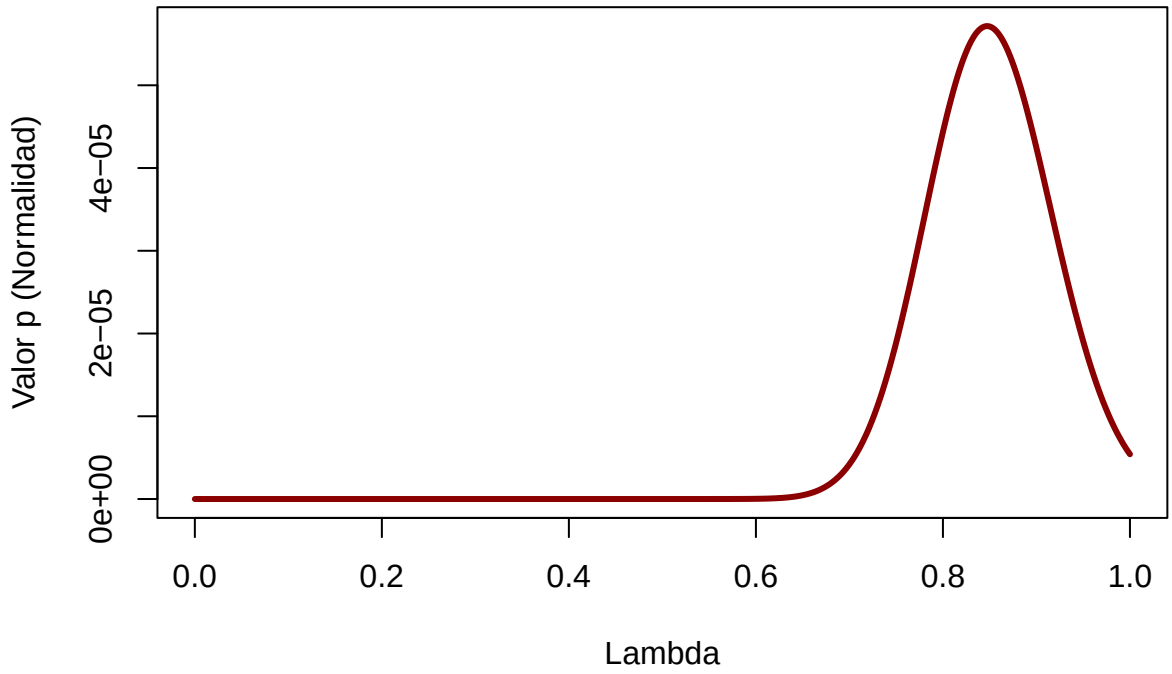
```
library(VGAM)
```

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

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

```
library(nortest)  # Asegúrate de tener cargada la librería para ad.test

lp <- seq(0, 1, 0.001)  # Valores de lambda propuestos
nlp <- length(lp)
n <- length(X_clean)
D <- matrix(as.numeric(NA), ncol=2, nrow=nlp)
d <- NA

for (i in 1:nlp) {
  d <- yeo.johnson(X_clean, lambda = lp[i])
```

```
  p <- ad.test(d)
  D[i, ] <- c(lp[i], p$p.value)
}

# Convertir la matriz en un data frame y asignar nombres a las columnas
N <- as.data.frame(D)
colnames(N) <- c("Lambda", "P_value")

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




```
# Encontrar el índice del máximo valor p
max_index <- which.max(N$P_value)

# Obtener el valor de lambda correspondiente
best_lambda <- N$Lambda[max_index]
max_p_value <- N$P_value[max_index]

# Imprimir los resultados
cat("El valor de lambda que maximiza el valor p es:", best_lambda, "\n")
```

```
## El valor de lambda que maximiza el valor p es: 0.847
```

```r
cat("El valor p máximo es:", max_p_value, "\n")
```

```
## El valor p máximo es: 5.715017e-05
```

Como se puede observar aún sin los datos atípicos y con la trasnformación Yeo Johnson maximizando el valor p no se pudo obtener la normalidad en los datos.

 # Escribe la ecuación del modelo encontrado.

Yeo Johnson $X_2 = \frac{(x+1)^{0.847}-1}{0.847}$

 # 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.

```r
X_aprox=sqrt(X_clean+1)
X1_yeo=yeo.johnson(X_clean, best_lambda)

# Función para calcular las medidas
calculate_stats <- function(x) {
  c(
    Min = min(x, na.rm = TRUE),
    Max = max(x, na.rm = TRUE),
    Mean = mean(x, na.rm = TRUE),
    Median = median(x, na.rm = TRUE),
    Q1 = quantile(x, 0.25, na.rm = TRUE),
    Q3 = quantile(x, 0.75, na.rm = TRUE),
    Skewness = skewness(x, na.rm = TRUE),
    Kurtosis = kurtosis(x, na.rm = TRUE)
  )
}

# Aplicar la función a cada variable
stats_X <- calculate_stats(X_clean)
stats_X1 <- calculate_stats(X_aprox)
stats_X2 <- calculate_stats(X1_yeo)

# Mostrar los resultados
print("Estadísticas con datos originales sin datos atípicos:")
```

```
## [1] "Estadísticas con datos originales sin datos atípicos:"
```

```r
print(stats_X)
```

```
##         Min         Max        Mean      Median      Q1.25%      Q3.75%
##   0.0000000 118.0000000  45.5686275  44.0000000  30.0000000  58.5000000
##    Skewness    Kurtosis
##   0.5780808   0.6442496
```

```r
print("Estadísticas con trasnformacion aproximada:")
```

```
## [1] "Estadísticas con trasnformacion aproximada:"
```

```r
print(stats_X1)
```

```
##        Min        Max       Mean     Median     Q1.25%     Q3.75%   Skewness
##  1.0000000 10.9087121  6.4811634  6.7082039  5.5677644  7.7135562 -0.7187367
##   Kurtosis
##  0.9595820
```

```r
print("Estadísticas con trasnformacion Yeo Johnson:")
```

```
## [1] "Estadísticas con trasnformacion Yeo Johnson:"
```

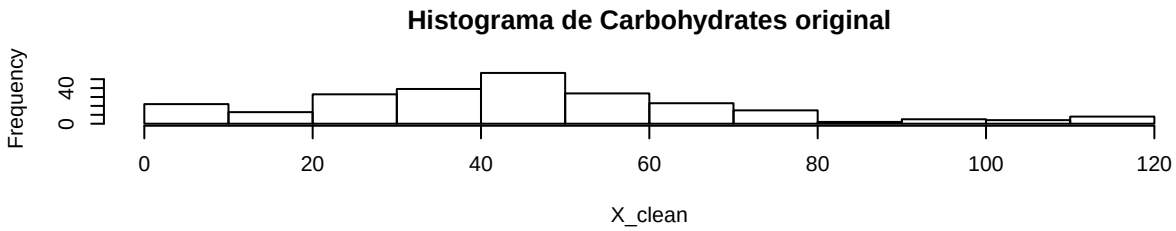
```r
print(stats_X2)
```

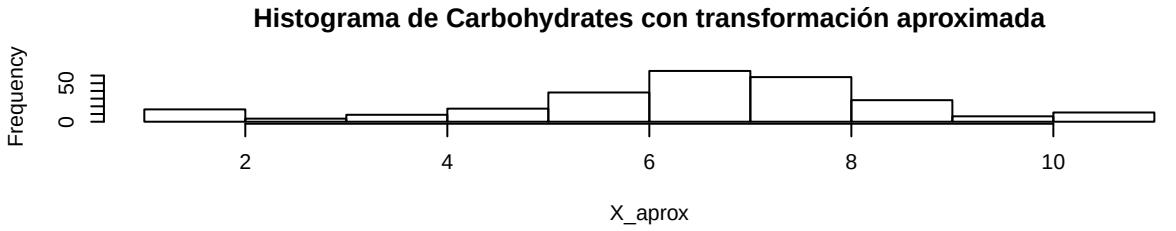
```
##        Min        Max       Mean     Median     Q1.25%     Q3.75%   Skewness
##  0.0000000 66.4439869 28.6834134 28.4941480 20.4614896 36.4143956  0.2375263
##   Kurtosis
##  0.4064258
```

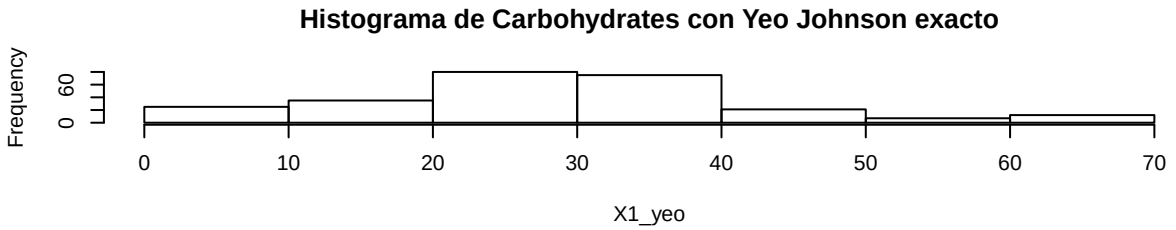
Se puede observar que la transformación aproximada incluso empeora a los datos originales sin datos atípicos debido a la curtosis y sesgo. Sin embargo, el modelo exacto Yeo Johnson y tiene menor curtosis y sesgo además de tener casi los mismo valores en media y mediana.

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

```r
par(mfrow=c(3,1))
hist(X_clean,col=0,main="Histograma de Carbohydrates original")
hist(X_aprox,col=0,main="Histograma de Carbohydrates con transformación aproximada")
hist(X1_yeo,col=0,main="Histograma de Carbohydrates con Yeo Johnson exacto")
```







El histograma de Yeo Johnson se ve más centralizado y sin colas altas al inicio y al final.

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

```
# Prueba de normalidad de Anderson-Darling para cada variable
ad_test_X <- ad.test(X_clean)
ad_test_X1 <- ad.test(X_aprox)
ad_test_X2 <- ad.test(X1_yeo)

# Mostrar los resultados
print("Prueba de Anderson-Darling con datos originales sin atípicos:")
```

```
## [1] "Prueba de Anderson-Darling con datos originales sin atípicos:"
```

```
print(ad_test_X)
```

```
##
##  Anderson-Darling normality test
##
## data:  X_clean
## A = 2.3621, p-value = 5.409e-06
```

```r
print("Prueba de Anderson-Darling con transformacion aproximada:")
```

```
## [1] "Prueba de Anderson-Darling con transformacion aproximada:"
```

```r
print(ad_test_X1)
```

```
##
##  Anderson-Darling normality test
##
## data:  X_aprox
## A = 4.8603, p-value = 4.649e-12
```

```r
print("Prueba de Anderson-Darling con transformacion exacta Yeo Johnson:")
```

```
## [1] "Prueba de Anderson-Darling con transformacion exacta Yeo Johnson:"
```

```r
print(ad_test_X2)
```

```
##
##  Anderson-Darling normality test
##
## data:  X1_yeo
## A = 1.9445, p-value = 5.715e-05
```

Ninguno de las tres transformaciones sobrepasa el 0.05 en el valor p por lo que no se consigue la normalidad. El modelo que más se acercó fue el de Yeo Johnson.

 # 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.

El mejor modelo fue el de Yeo Johnson ya que tiene el mayor valor p con diferencia pero aún así no obtuvo el valor p necesario para conseguir normalidad.

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

La transformación de Box-Cox es eficaz para normalizar datos positivos y estabilizar la varianza, siendo sencilla de aplicar, pero limitada a valores positivos y sensible a outliers. Por otro lado, la transformación de Yeo-Johnson extiende la aplicabilidad a datos que incluyen valores negativos, ofreciendo mayor flexibilidad y versatilidad. Sin embargo, Yeo-Johnson es más complejo de interpretar y menos conocido en comparación con Box-Cox. La elección entre ambas depende del tipo de datos: Box-Cox es preferible para datos estrictamente positivos, mientras que Yeo-Johnson es ideal para conjuntos de datos más diversos.

 # 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

Propósito:

Transformación: Cambia la distribución de los datos para cumplir con ciertos supuestos estadísticos, como la normalidad. Ejemplos incluyen las transformaciones logarítmica, Box-Cox y Yeo-Johnson, que se utilizan para estabilizar la varianza y normalizar los datos.

Escalamiento: Cambia la escala de los datos, ajustando los valores a un rango específico (como [0, 1] en la normalización o una distribución con media 0 y desviación estándar 1 en la estandarización), sin alterar la forma de la distribución.

Efecto en la Distribución:

Transformación: Modifica la distribución de los datos, lo que puede afectar la simetría, la curtosis y la presencia de colas largas. Por ejemplo, una transformación logarítmica puede convertir una distribución sesgada positivamente en una más simétrica.

Escalamiento: No altera la distribución subyacente de los datos. Un conjunto de datos sesgado seguirá siendo sesgado después del escalamiento; simplemente se ajustará a un nuevo rango.

Aplicaciones:

Transformación: Es necesaria cuando se requiere que los datos cumplan con ciertos supuestos estadísticos, como en la regresión lineal, ANOVA, o cuando se desea mejorar la normalidad de los datos para utilizar métodos paramétricos.

Escalamiento: Es esencial en métodos de machine learning que son sensibles a la magnitud de las características, como en algoritmos basados en la distancia (e.g., KNN, SVM) o en redes neuronales, donde la uniformidad en la escala de los datos puede mejorar la convergencia y la precisión del modelo.

Indica cuándo es necesario utilizar cada uno

Transformación es necesaria cuando los datos no cumplen con los supuestos de normalidad o homocedasticidad en modelos estadísticos, o cuando se desea estabilizar la varianza. También se utiliza en la preparación de datos antes de aplicar ciertos métodos estadísticos o en situaciones donde se busca linearizar relaciones no lineales.

Escalamiento se utiliza en el preprocesamiento de datos para algoritmos de machine learning, especialmente aquellos que dependen de la distancia o que son sensibles a las magnitudes de las características. Es crucial para asegurar que todas las características contribuyan de manera uniforme en el proceso de modelado.