5. Transformaciones

Juan Bernal

2024-08-14

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

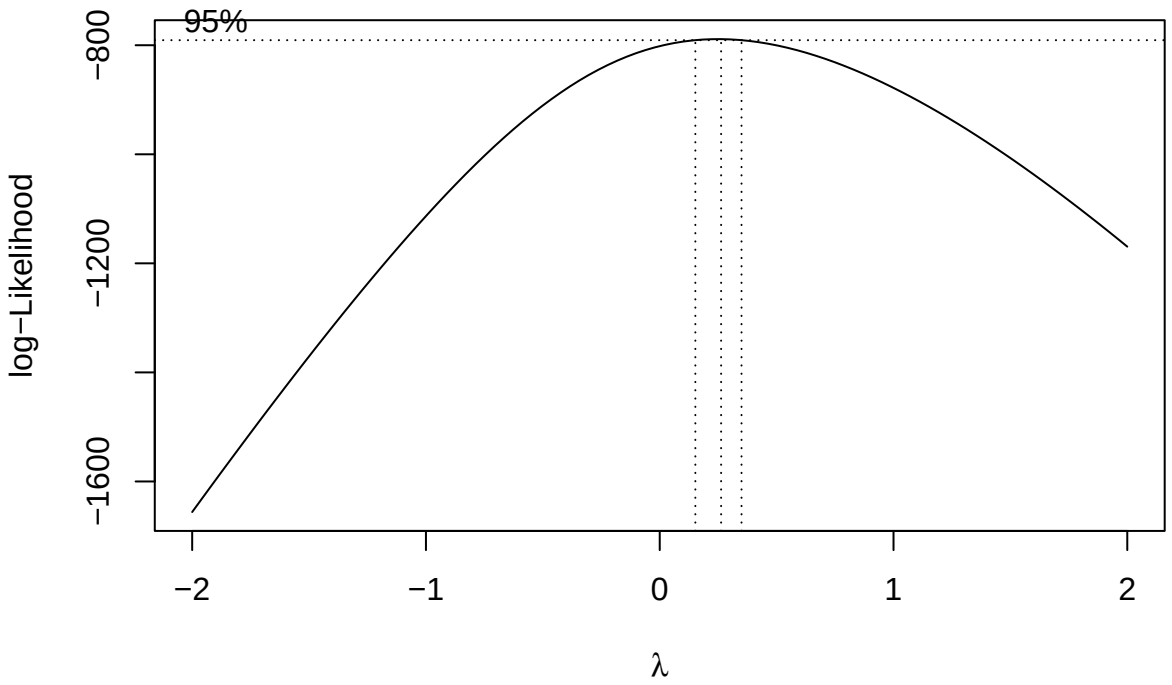
```
library(nortest)
library(MASS)

data=read.csv("mc-donalds-menu.csv") #leer la base de datos
```

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:

```
x = data$Sugars
```

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

```
bc = boxcox(lm(x+1~1))
```




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

```
l
```

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

El valor de lambda que maximiza la función de verosimilitud es 0.2626.

2. Escribe las ecuaciones de los modelos encontrados.

El modelo aproximado queda como $x_1 = \sqrt{x+1}$, y el modelo exacto queda como $x_2 = \frac{(x+1)^{0.2626}-1}{0.2626}$.

```
x1 = sqrt(x+1) # Modelo 1
x2 = ((x+1)^l - 1)/l # Modelo 2
```

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

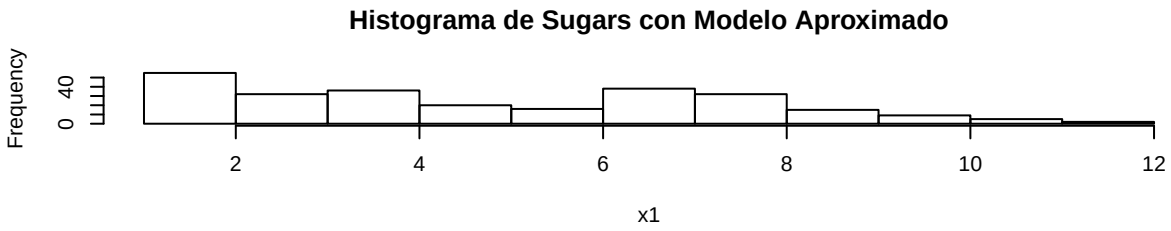
    1. Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis.

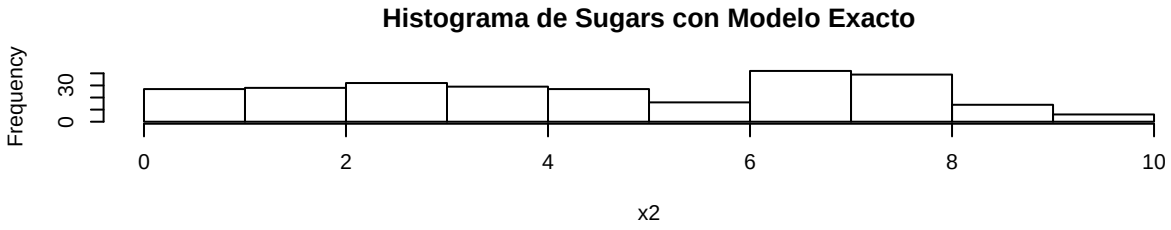
```
library(e1071)
m0=round(c(as.numeric(summary(x)),kurtosis(x),skewness(x)),3)
m1=round(c(as.numeric(summary(x1)),kurtosis(x1),skewness(x1)),3)
m2=round(c(as.numeric(summary(x2)),kurtosis(x2),skewness(x2)),3)

m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Modelo aproximado","Modelo exacto")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo")
m
```

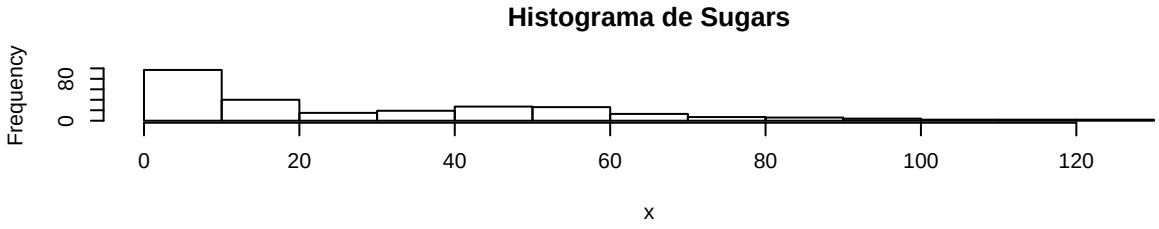
```
##                   Minimo    Q1 Mediana  Media     Q3  Máximo Curtosis  Sesgo
## Original               0 5.750  17.500 29.423 48.000 128.000    0.461  1.020
## Modelo aproximado      1 2.597   4.301  4.825  7.000  11.358   -1.014  0.279
## Modelo exacto          0 2.477   4.385  4.519  6.774   9.837   -1.113 -0.106
```

2. 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 Sugars con Modelo Aproximado")
hist(x2,col=0,main="Histograma de Sugars con Modelo Exacto")
hist(x,col=0,main="Histograma de Sugars")
```







Notemos que los datos originales muestran un sesgo a la derecha, mientras que el modelo aproximado y el modelo exacto parecen centrarse más, mas no son normales.

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

```
D=ad.test(x2)
print("P-value de Sugars con Modelo Exacto")
```

```
## [1] "P-value de Sugars con Modelo Exacto"
```

```
D$p.value
```

```
## [1] 1.857266e-08
```

Dado que el p-value es menor a un alpha estándar de 0.05, se rechaza la hipótesis inicial y la distribución del modelo exacto no es normal.

```
D=ad.test(x1)
print("P-value de Sugars con Modelo Aproximado")
```

```
## [1] "P-value de Sugars con Modelo Aproximado"
```

```
D$p.value
```

```
## [1] 3.531062e-10
```

Dado que el p-value es menor a un alpha estándar de 0.05, se rechaza la hipótesis inicial y la distribución del modelo aproximado no es normal.

```
D=ad.test(x)
print("P-value de Sugars")
```

```
## [1] "P-value de Sugars"
```

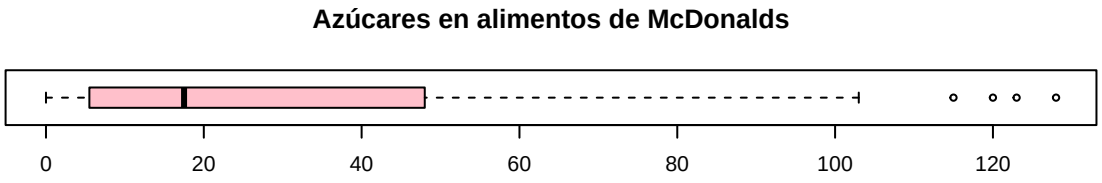
```
D$p.value
```

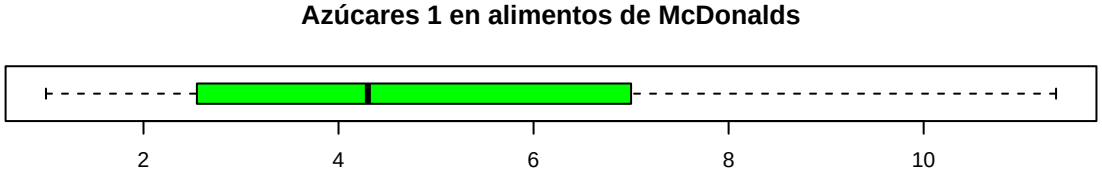
```
## [1] 3.7e-24
```

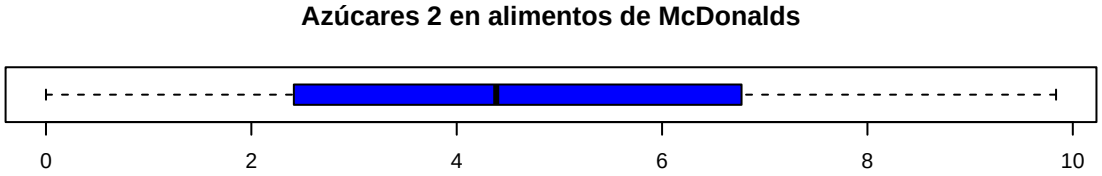
Dado que el p-value es menor a un alpha estándar de 0.05, se rechaza la hipótesis inicial y la distribución de los datos originales no es normal.

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

```
par(mfrow=c(3,1))
boxplot(x, horizontal = TRUE,col="pink", main="Azúcares en alimentos de McDonalds")
boxplot(x1, horizontal = TRUE,col="green", main="Azúcares 1 en alimentos de McDonalds")
boxplot(x2, horizontal = TRUE,col="blue", main="Azúcares 2 en alimentos de McDonalds")
```



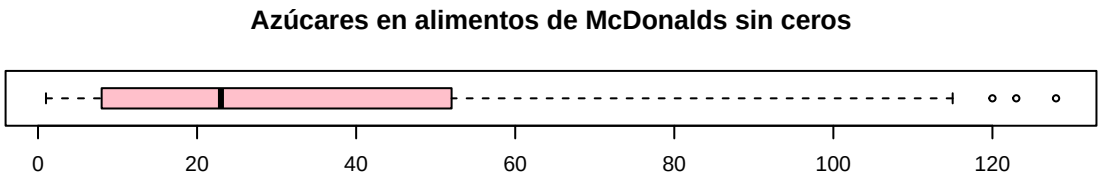


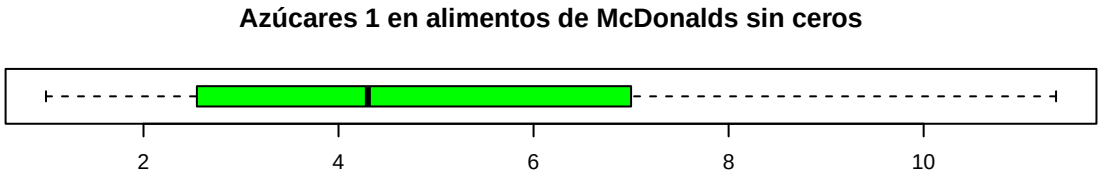


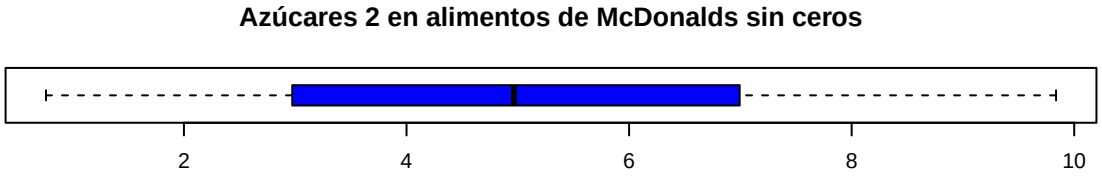
Notemos que solo los datos originales muestran datos atípicos y extremos, mientras que el modelo aproximado solo muestra un sesgo a la derecha y el modelo exacto es casi normal.

Quitaremos los ceros, debido al ruido que pudimos notar en los histogramas, ya que tampoco hace sentido que comida rápida tenga 0 azúcares.

```
xx = subset(x,x>0)
x11 = subset(x1,x1>0)
x22 = subset(x2,x2>0)

par(mfrow=c(3,1))
boxplot(xx, horizontal = TRUE,col="pink", main="Azúcares en alimentos de McDonalds sin ceros")
boxplot(x11, horizontal = TRUE,col="green", main="Azúcares 1 en alimentos de McDonalds sin ceros")
boxplot(x22, horizontal = TRUE,col="blue", main="Azúcares 2 en alimentos de McDonalds sin ceros")
```







El único cambio notable es la desaparición de un dato atípico en la boxplot de los datos originales

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).

```
library(VGAM)
```

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

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

```
library(car)        # Para la transformación Yeo-Johnson
```

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

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

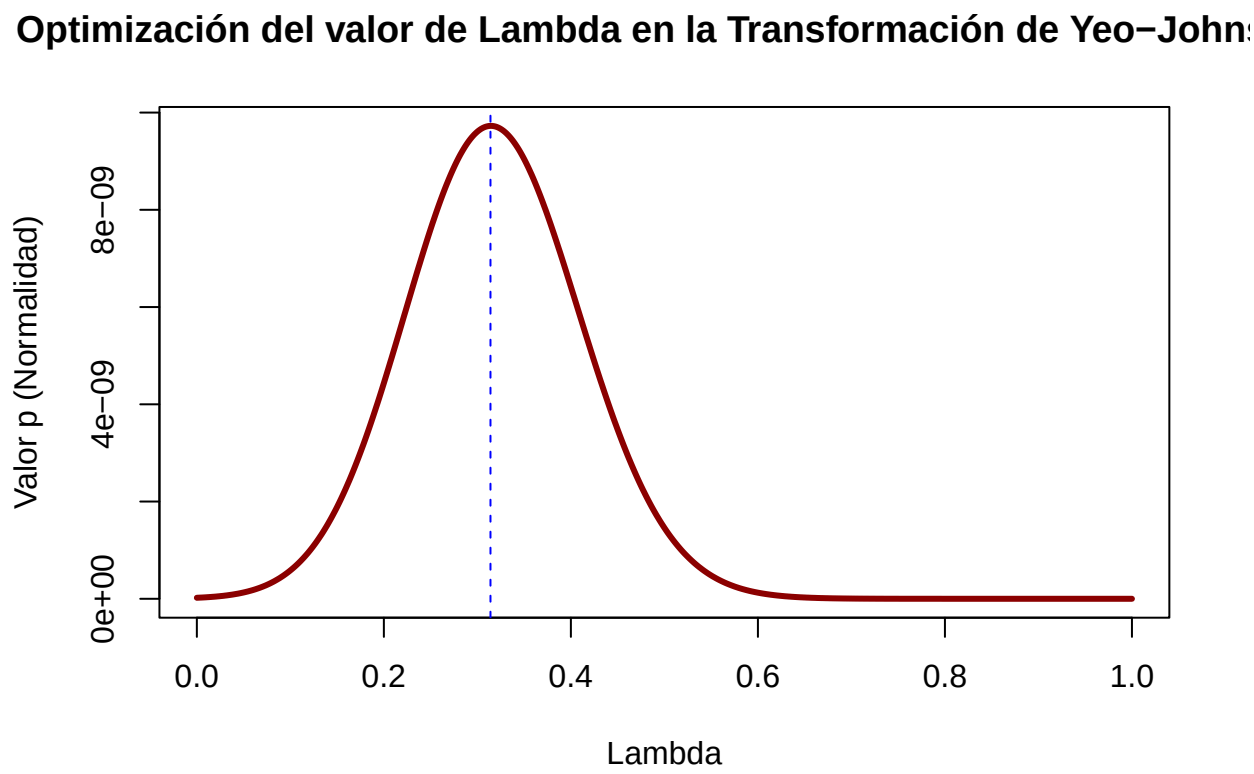
```
## The following object is masked from 'package:VGAM':
##
##     logit
```

```
library(nortest)    # Para la prueba de Anderson-Darling

# Supongamos que los datos están en un vector 'xx'
# xx <- your_data_here

# Paso 1: Definir la secuencia de valores de lambda
lp <- seq(0, 1, 0.001)  # Ampliamos el rango de lambda para cubrir más posibilidades
nlp <- length(lp)

# Paso 2: Crear una matriz para almacenar lambda y su valor p correspondiente
D <- matrix(NA, ncol=2, nrow=nlp)
```

```r
# Paso 3: Iterar sobre los valores de lambda y aplicar la transformación Yeo-Johnson
for (i in 1:nlp) {
  d <- yeo.johnson(xx, lambda = lp[i])  # Aplicar la transformación
  p <- ad.test(d)$p.value                 # Realizar la prueba de normalidad
  D[i,] <- c(lp[i], p)                    # Guardar el lambda y el valor p
}

# Paso 4: Convertir la matriz en un dataframe y asignar nombres a las columnas
N <- as.data.frame(D)
colnames(N) <- c("Lambda", "Valor_p")

# Paso 5: Encontrar el lambda que maximiza el valor p
best_lambda <- N$Lambda[which.max(N$Valor_p)]
max_p_value <- max(N$Valor_p)

# Imprimir los resultados
cat("El mejor valor de lambda es:", best_lambda, "con un valor p de:", max_p_value, "\n")
```

```
## El mejor valor de lambda es: 0.314 con un valor p de: 9.726043e-09
```

```r
# Paso 6: Graficar los resultados
plot(N$Lambda, N$Valor_p, type="l",
     col="darkred", lwd=3,
     xlab="Lambda",
     ylab="Valor p (Normalidad)",
     main="Optimización del valor de Lambda en la Transformación de Yeo-Johnson")
abline(v = best_lambda, col="blue", lty=2)  # Línea vertical en el mejor lambda
```



El mejor valor de lambda es: 0.314 con un P_value de 9.72e-09

6. Escribe la ecuación del modelo encontrado.

Dado los datos originales no contienen valores negativos y que la lambda obtenida es diferente de cero, entonces la ecuacion del modelo con transformada de Yeo Johnson queda: $x_3 = \frac{(x+1)^{0.314}-1}{0.314}$

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

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

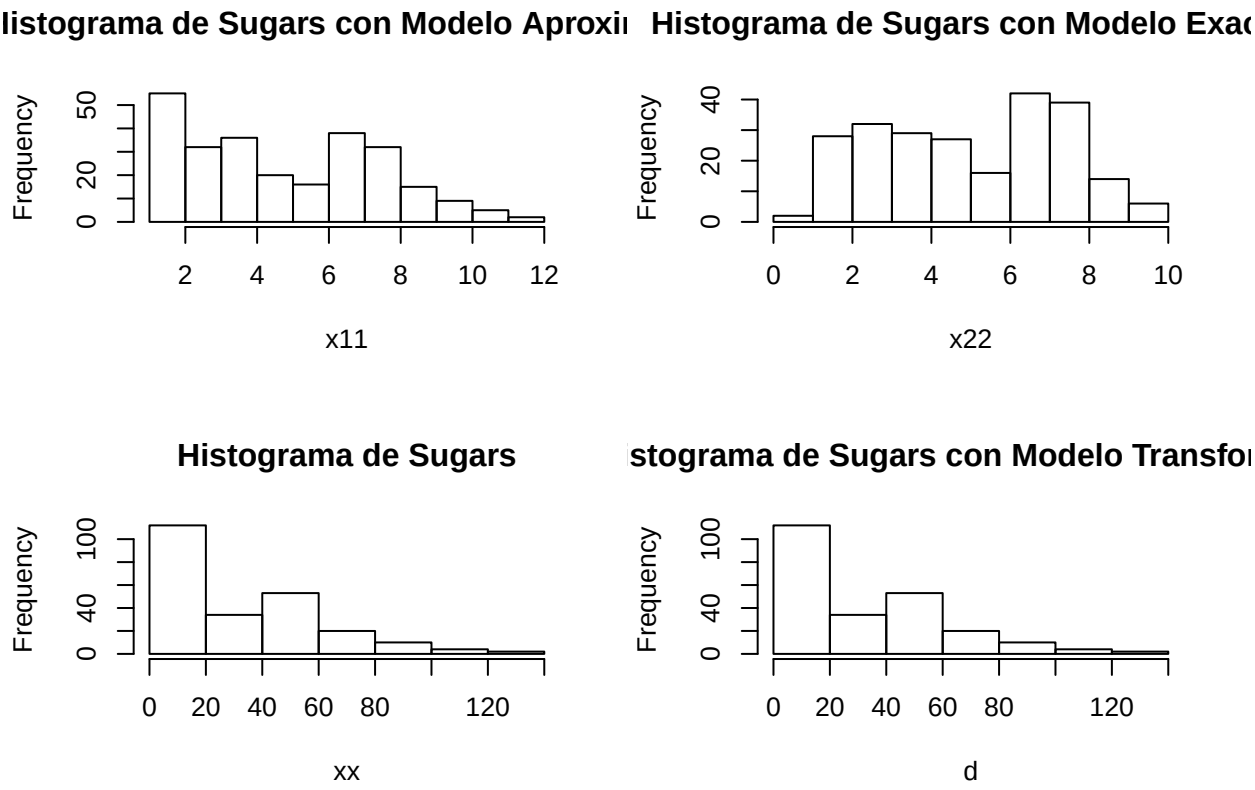
```
m3=round(c(as.numeric(summary(d)),kurtosis(d),skewness(d)),3)

m<-as.data.frame(rbind(m0,m1,m2,m3))
row.names(m)=c("Original","Modelo aproximado","Modelo exacto","Modelo Transformado")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo")
m
```

```
##                     Minimo    Q1 Mediana  Media     Q3  Máximo Curtosis  Sesgo
## Original                 0 5.750  17.500 29.423 48.000 128.000    0.461  1.020
## Modelo aproximado        1 2.597   4.301  4.825  7.000  11.358   -1.014  0.279
## Modelo exacto            0 2.477   4.385  4.519  6.774   9.837   -1.113 -0.106
## Modelo Transformado      1 8.000  23.000 32.553 52.000 128.000    0.356  0.947
```

El análisis de la tabla se realizará al final.

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

```
par(mfrow=c(2,2))
hist(x11,col=0,main="Histograma de Sugars con Modelo Aproximado")
hist(x22,col=0,main="Histograma de Sugars con Modelo Exacto")
hist(xx,col=0,main="Histograma de Sugars")
hist(d,col=0,main="Histograma de Sugars con Modelo Transformado")
```



En los histogramas notamos que el modelo exacto se centraliza más que los datos originales, el modelo aproximado y el modelo transformado. Pero aún no es normal, pues las frecuencias estan casi uniformemente

distribuidas, en lugar de centrarse y empezar presentar una clase de acumulación alrededor de un intervalo.

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

```
D=ad.test(x)
print("P-value de Sugars")
```

```
## [1] "P-value de Sugars"
```

```
D$p.value
```

```
## [1] 3.7e-24
```

Dado que el p-value es menor a un alpha estándar de 0.05, se rechaza la hipótesis inicial y la distribución de los datos originales no es normal.

```
D=ad.test(d)
print("P-value de Sugars con Modelo Transformado")
```

```
## [1] "P-value de Sugars con Modelo Transformado"
```

```
D$p.value
```

```
## [1] 3.162656e-18
```

Dado que el p-value es menor a un alpha estándar de 0.05, se rechaza la hipótesis inicial y la distribución del modelo transformado no es normal.

8. Define la mejor transformación de los datos de acuerdo a las características de los modelos que encontraste.

Es difícil definir al mejor, dado que todos presentan características diferentes. En los datos originales se obtuvo una curtosis más baja, pero un sesgo de más 1. En el modelo aproximado presenta el 2er mayor sesgo de los 3 modelos y la 2da menor curtosis de los mismos. Luego está el modelo exacto, que tiene el menor sesgo de los modelos, pero su curtosis es aún más alta que la del modelo aproximado. Y por último, el modelo transformado presenta la menor curtosis, pero el mayor sesgo de todos los modelos, muy cercano a uno, se observa que los datos están sesgados. Dadas estas características, la mejor transformación sería la del modelo exacto, ya que aunado a estas características, también obtuvo el p_value más alto de todas las pruebas de normalidad.

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

Box-Cox es ideal cuando se trabaja con datos estrictamente positivos y se necesita una transformación sencilla y eficaz para normalizar los datos. Mientras que Yeo-Johnson ofrece una mayor flexibilidad al poder manejar datos con valores negativos y cero, lo que lo hace más aplicable en situaciones donde los datos no son estrictamente positivos. La elección entre Box-Cox y Yeo-Johnson depende de la naturaleza de los datos y las necesidades específicas del análisis. En general, Yeo-Johnson es más versátil, mientras que Box-Cox es más sencillo y eficaz cuando se cumplen sus supuestos.

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

    1. Escribe al menos 3 diferencias entre lo que es la transformación y el escalamiento de los datos

Propósito: La transformación cambia la distribución de los datos, mientras que el escalamiento ajusta la escala sin alterar la distribución.

Impacto en la Distribución: La transformación puede alterar la forma de la distribución; el escalamiento no cambia la forma, solo la magnitud.

Métodos: La transformación usa métodos no lineales (como logaritmos), mientras que el escalamiento utiliza métodos lineales (como normalización).

`2. Indica cuándo es necesario utilizar cada uno`

La transformación tiene como objetivo cambiar la distribución de los datos para que se ajusten mejor a los supuestos de los modelos estadísticos o para mejorar la interpretación. Por ejemplo, se puede usar una transformación logarítmica para hacer que una distribución sesgada se acerque más a una distribución normal.

El escalamiento tiene como objetivo modificar la escala de los datos, es decir, cambiar los valores a un rango estándar (por ejemplo, entre 0 y 1 o con media 0 y desviación estándar 1) para que todas las características tengan la misma importancia relativa.