Actividad Integradora 1

Héctor Hibran Tapia Fernández - A01661114

2024-08-20

Por hacer…

Trabaja con el set de datos Nutrición Mundial, que contiene diversas características del alimentos que se consumen en el mundo. Pueden encontrar más información sobre ella en: Utsav Dey. (2024). Food Nutrition Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/8820139Links (https://doi.org/10.34740/KAGGLE/DSV/8820139Links) to an external site.. El resumen de su contenido es:

“The Comprehensive Nutritional Food Database provides detailed nutritional information for a wide range of food items commonly consumed around the world. This dataset aims to support dietary planning, nutritional analysis, and educational purposes by providing extensive data on the macro and micronutrient content of foods.”

Punto 1. Análisis Descriptivo de la Variable

Analiza una de las siguientes variables en cuanto a sus datos atípicos y normalidad. La variable que te corresponde analizar te será asignada por tu profesora al inicio de la actividad:

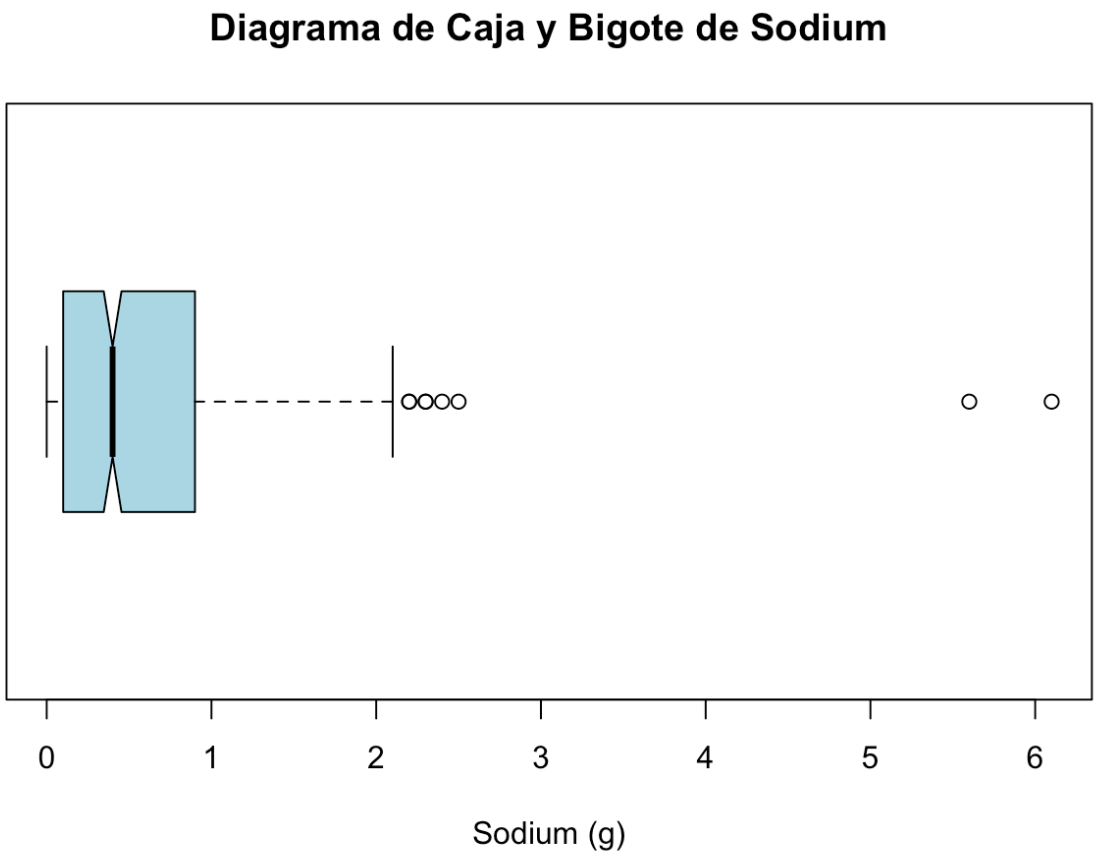
- Calorías
- Grasas saturadas
- Grasas monosaturadas
- Sodio
- Agua
- *Sodio*
- Densidad Nutricional

```
data <- read.csv("./food_data_g.csv", header = TRUE, sep = ",")
```

Para analizar datos atípicos se te sugiere:

Graficar el diagrama de caja y bigote

```
boxplot(data$Sodium, main = "Diagrama de Caja y Bigote de Sodium", xlab = "Sodium (g)", co
l = "lightblue", border = "black", notch = TRUE, horizontal = TRUE)
```



Calcula las principales medidas que te ayuden a identificar datos atípicos (utilizar summary te puede abreviar el cálculo): Cuartil 1, Cuartil 3, Media, Cuartil 3, Rango intercuartílico y Desviación estándar

```
summary(data$Sodium)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1000  0.4000  0.5732  0.9000  6.1000
```

```
cat("-----------------------------------------------\n")
```

```
## -----------------------------------------------
```

```
q1 <- quantile(data$Sodium, 0.25)
q2 <- quantile(data$Sodium, 0.50)
q3 <- quantile(data$Sodium, 0.75)

iqr <- IQR(data$Sodium)

# Límites para los datos atípicos
lower_bound <- q1 - 1.5 * iqr
upper_bound <- q3 + 1.5 * iqr

outliers_IQR <- data$Sodium[data$Sodium < lower_bound | data$Sodium > upper_bound] # Datos atípicos o outliers

cat("Cuartiles:\n")
```

```
## Cuartiles:
```

```
cat("Al 25%: ", q1, "\n")
```

```
## Al 25%:  0.1
```

```
cat("Al 50%: ", q2, "\n")
```

```
## Al 50%:  0.4
```

```
cat("Al 75%: ", q3, "\n")
```

```
## Al 75%:  0.9
```

```
cat("Rango Intercuartílico: ", iqr, "\n")
```

```
## Rango Intercuartílico:  0.8
```

```
media <- mean(data$Sodium, na.rm = TRUE)
mediana <- median(data$Sodium, na.rm = TRUE)
rango_medio <- (max(data$Sodium, na.rm = TRUE) + min(data$Sodium, na.rm = TRUE)) / 2

cat("-----------------------------------------------\n")
```

```
## -----------------------------------------------
```

```
cat("Media: ", media, "\n")
```

```
## Media:  0.5732051
```

```
cat("Mediana: ", mediana, "\n") # Es igual al q2
```

```
## Mediana:  0.4
```

```
cat("Rango Medio: ", rango_medio, "\n")
```

```
## Rango Medio:  3.05
```

```
desviacion_estandar <- sd(data$Sodium)
cat("Desviación Estándar: ", desviacion_estandar, "\n")
```

```
## Desviación Estándar:  0.6361261
```

```
sd_Sodium <- sd(data$Sodium, na.rm = TRUE)

lower_bound_sd <- media - 3 * sd_Sodium
upper_bound_sd <- media + 3 * sd_Sodium

outliers_sd <- data$Sodium[data$Sodium < lower_bound_sd | data$Sodium > upper_bound_sd]
```

Identifica la cota de 1.5 rangos intercuartílicos para datos atípicos, ¿hay datos atípicos de acuerdo con este criterio? ¿cuántos son?

```
cat("Outliers con la cota de 1.5 rangos intercuartílicos: ", outliers_IQR, "\n")
```

```
## Outliers con la cota de 1.5 rangos intercuartílicos:  2.4 2.3 2.5 2.2 2.3 2.2 5.6 6.1
```

Identifica la cota de 3 desviaciones estándar alrededor de la media, ¿hay datos atípicos de acuerdo con este criterio? ¿cuántos son?

```
cat("Outliers con la cota de 3 desviaciones estándar al rededor de la media: ", outliers_s
d, "\n")
```

```
## Outliers con la cota de 3 desviaciones estándar al rededor de la media:  2.5 5.6 6.1
```

Identifica la cota de 3 rangos intercuartílicos para datos extremos, ¿hay datos extremos de acuerdo con este criterio?

¿cuántos son?

```
lower_bound_3iqr <- q1 - 3 * iqr
upper_bound_3iqr <- q3 + 3 * iqr

extreme_outliers <- data$Sodium[data$Sodium < lower_bound_3iqr | data$Sodium > upper_bound
_3iqr]

cat("Límite inferior (3 IQR): ", lower_bound_3iqr, "\n")
```

```
## Límite inferior (3 IQR):  -2.3
```

```
cat("Límite superior (3 IQR): ", upper_bound_3iqr, "\n")
```

```
## Límite superior (3 IQR):  3.3
```

```
cat("Datos extremos: ", extreme_outliers, "\n")
```

```
## Datos extremos:  5.6 6.1
```

Para analizar normalidad se te sugiere:

Realiza pruebas de normalidad univariada para la variable (utiliza las pruebas de Anderson-Darling y de Jarque Bera). No olvides incluir H0 y H1 para la prueba de normalidad.

```
library(nortest)
library(moments)

ad_test <- ad.test(data$Sodium)
print(ad_test)
```

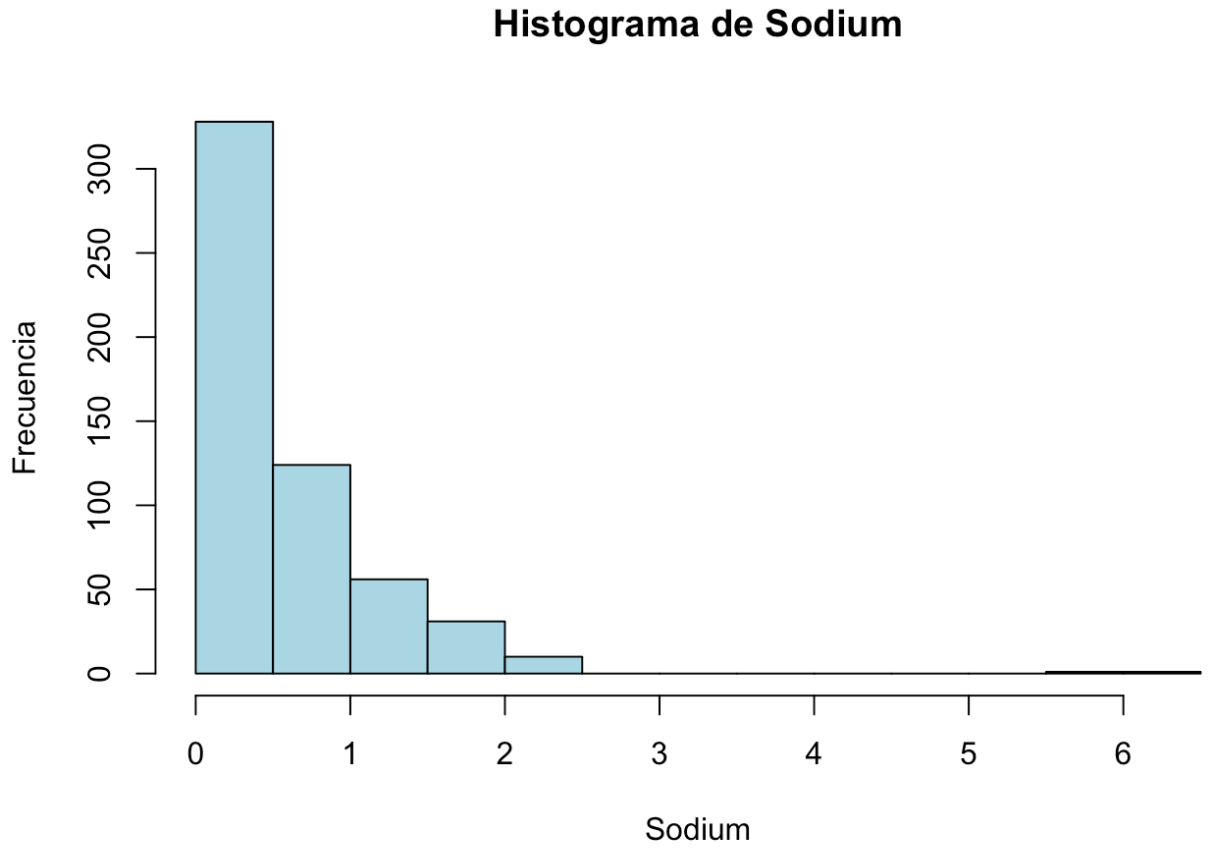
```
##
##  Anderson-Darling normality test
##
## data:  data$Sodium
## A = 24.827, p-value < 2.2e-16
```

```
jb_test <- jarque.test(data$Sodium)
print(jb_test)
```

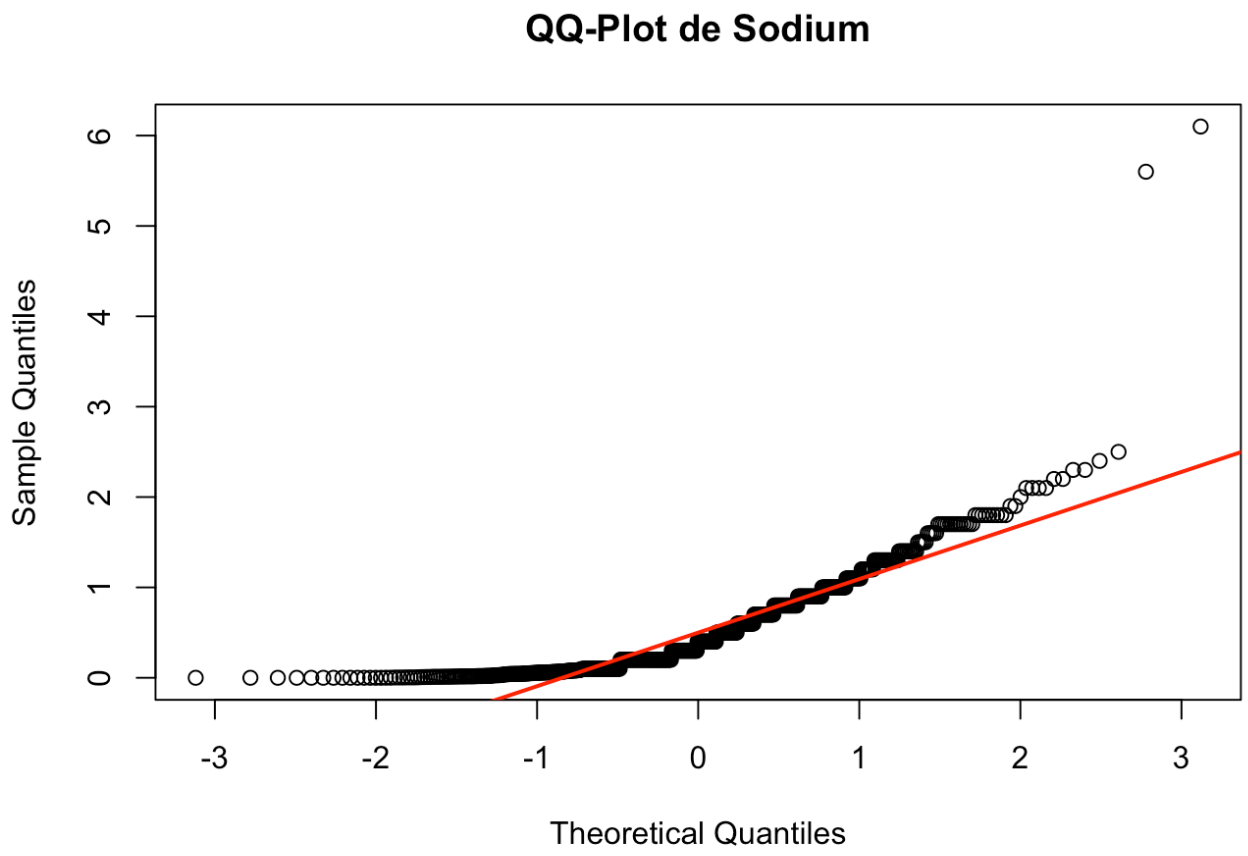
```
##
##  Jarque-Bera Normality Test
##
## data:  data$Sodium
## JB = 6834.2, p-value < 2.2e-16
## alternative hypothesis: greater
```

Grafica los datos y su respectivo QQPlot: qqnorm(datos) y qqline(datos).

```
hist(data$Sodium, main = "Histograma de Sodium", xlab = "Sodium", ylab = "Frecuencia", col = "lightblue", border = "black")
```




```
qqnorm(data$Sodium, main = "QQ-Plot de Sodium")
qqline(data$Sodium, col = "red", lwd = 2)
```



Calcula el coeficiente de sesgo y el coeficiente de curtosis

```
sesgo = skewness(data$Sodium)
curtosis = kurtosis(data$Sodium)

cat("Sesgo: ", sesgo, "\n")
```

```
## Sesgo:  2.735999
```

```
cat("Curtosis: ", curtosis, "\n")
```

```
## Curtosis:  19.3626
```

Compara las medidas de media, mediana y rango medio de cada variable.

```
media <- mean(data$Sodium, na.rm = TRUE)
mediana <- median(data$Sodium, na.rm = TRUE)
rango_medio <- (max(data$Sodium, na.rm = TRUE) + min(data$Sodium, na.rm = TRUE)) / 2

cat("Media: ", media, "\n")
```

```
## Media:  0.5732051
```

```
cat("Mediana: ", mediana, "\n") # Es igual al q2
```

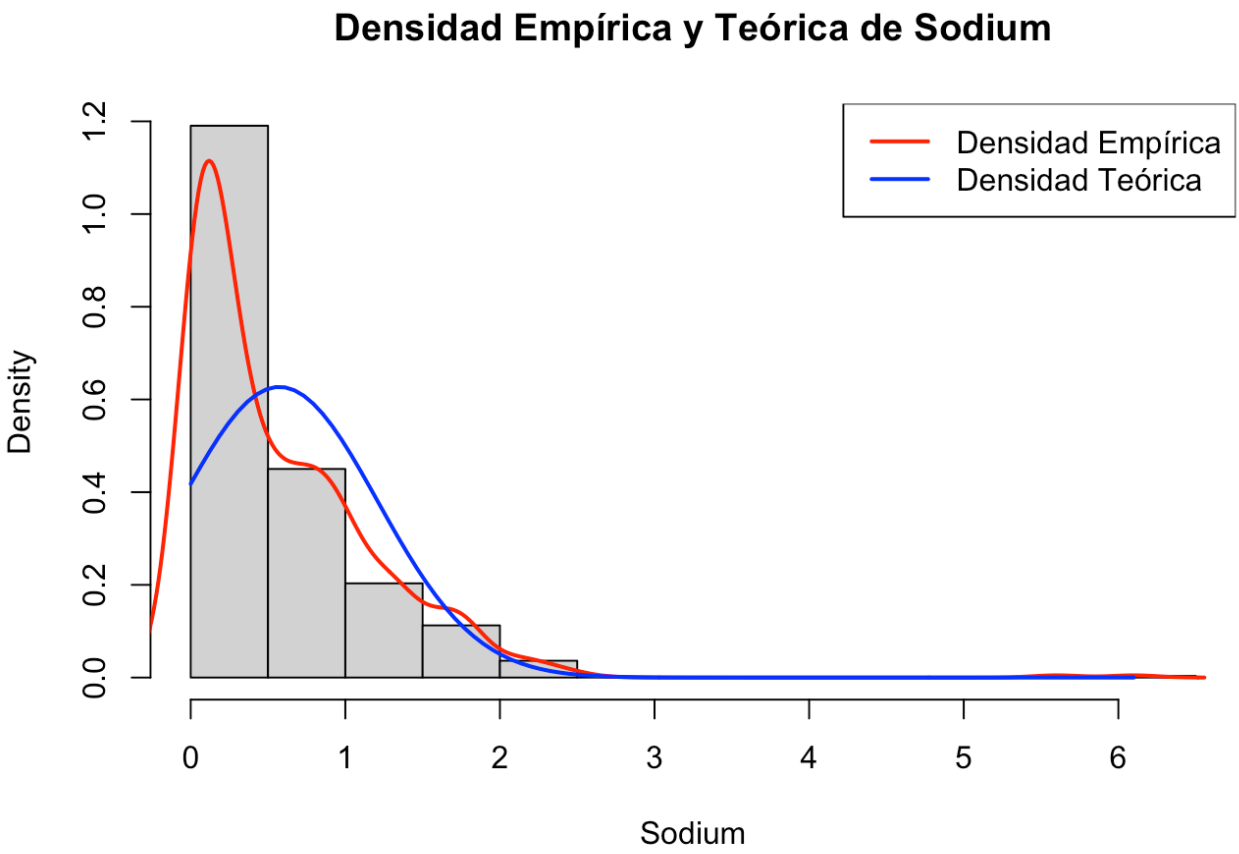
```
## Mediana:  0.4
```

```
cat("Rango Medio: ", rango_medio, "\n")
```

```
## Rango Medio:  3.05
```

Realiza el gráfico de densidad empírica y teórica suponiendo normalidad en la variable. Adapta el código:

```
hist(data$Sodium, freq = FALSE, main = "Densidad Empírica y Teórica de Sodium", xlab = "So
dium", col = "lightgray", border = "black")

lines(density(data$Sodium), col = "red", lwd = 2)

curve(dnorm(x, mean = mean(data$Sodium), sd = sd(data$Sodium)), from = min(data$Sodium), t
o = max(data$Sodium), add = TRUE, col = "blue", lwd = 2)

legend("topright", legend = c("Densidad Empírica", "Densidad Teórica"), col = c("red", "bl
ue"), lwd = 2)
```



Punto 2. Transformación a Normalidad

Encuentra la mejor transformación de los datos para lograr normalidad. Puedes hacer uso de la transformación Box-Cox o de Yeo Johnson o el comando powerTransform para encontrar la mejor lambda para la transformación. Utiliza el modelo exacto y el aproximado de acuerdo con las sugerencias de Box y Cox para la transformación.

```
library(car)
```

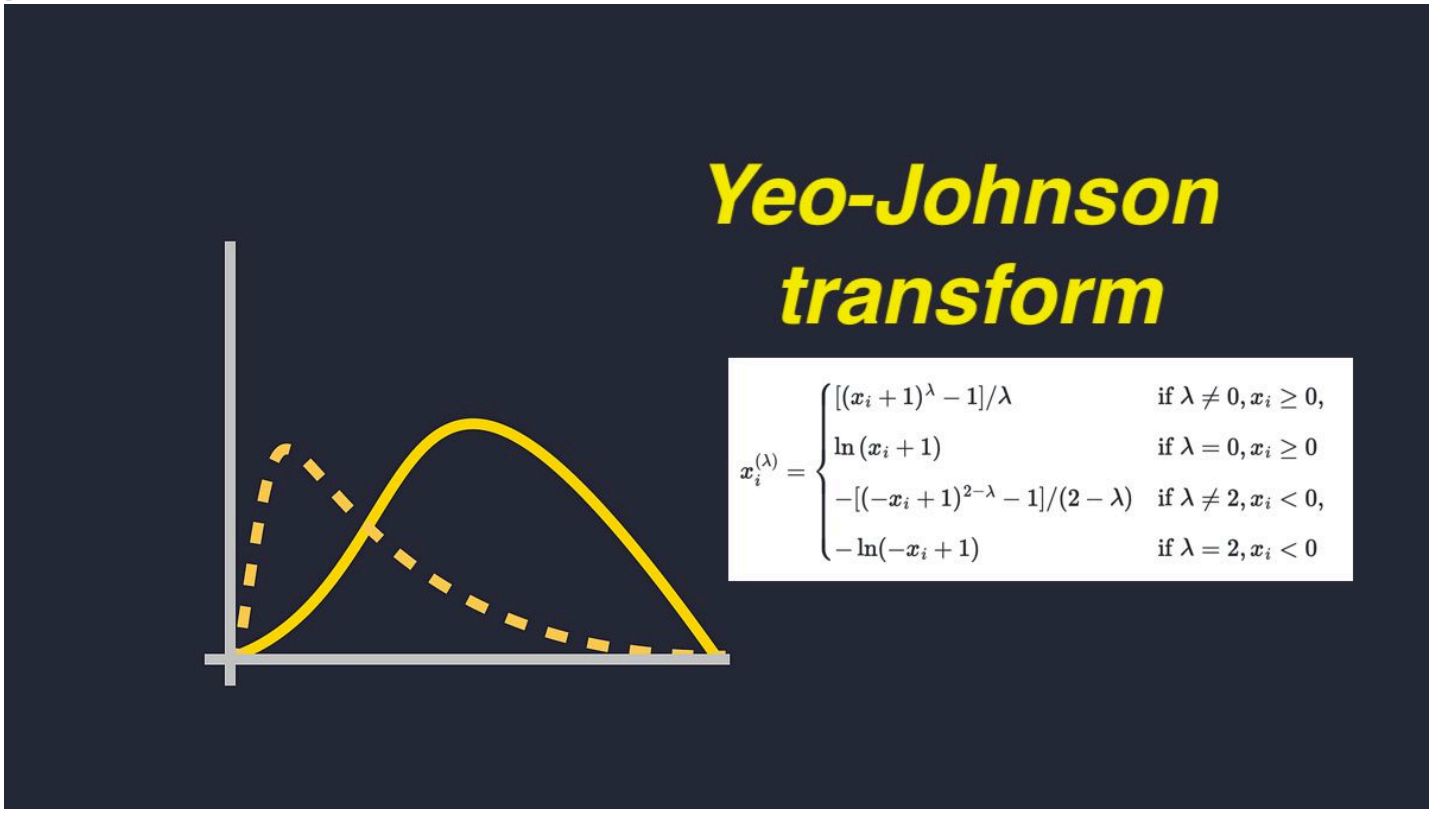
```
## Loading required package: carData
```

```
yj_transform <- powerTransform(data$Sodium, family = "yjPower")
summary(yj_transform)
```

```
## yjPower Transformation to Normality
##             Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## data$Sodium   -1.2375          -1      -1.5265      -0.9485
##
##  Likelihood ratio test that transformation parameter is equal to 0
##                            LRT df       pval
## LR test, lambda = (0) 85.66163  1 < 2.22e-16
```

Escribe las ecuaciones de los modelos de transformación encontrados.

[



]https://es.linkedin.com/posts/naren-castellon-1541b8101_la-transformaci%C3%B3n-yeo-johnson-es-una-t%C3%A9cnica-activity-7132784033536577536-Hor7# (https://es.linkedin.com/posts/naren-castellon-1541b8101_la-transformaci%C3%B3n-yeo-johnson-es-una-t%C3%A9cnica-activity-7132784033536577536-Hor7#):~:text=La%20f%C3%B3rmula%20general%20de%20la,representa%20los%20valores%20originales%20dem)

Basándonos en el siguiente link: https://feaz-book.com/numeric-yeojohnson (https://feaz-book.com/numeric-yeojohnson)

Usaremos el lambda estimado = -1.2375

$$Y_{\text{transformada}} = \frac{[(Y+1)^{-1.2375} - 1]}{-1.2375}, \quad \text{para } Y \geq 0$$

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

argumento de normalidad:

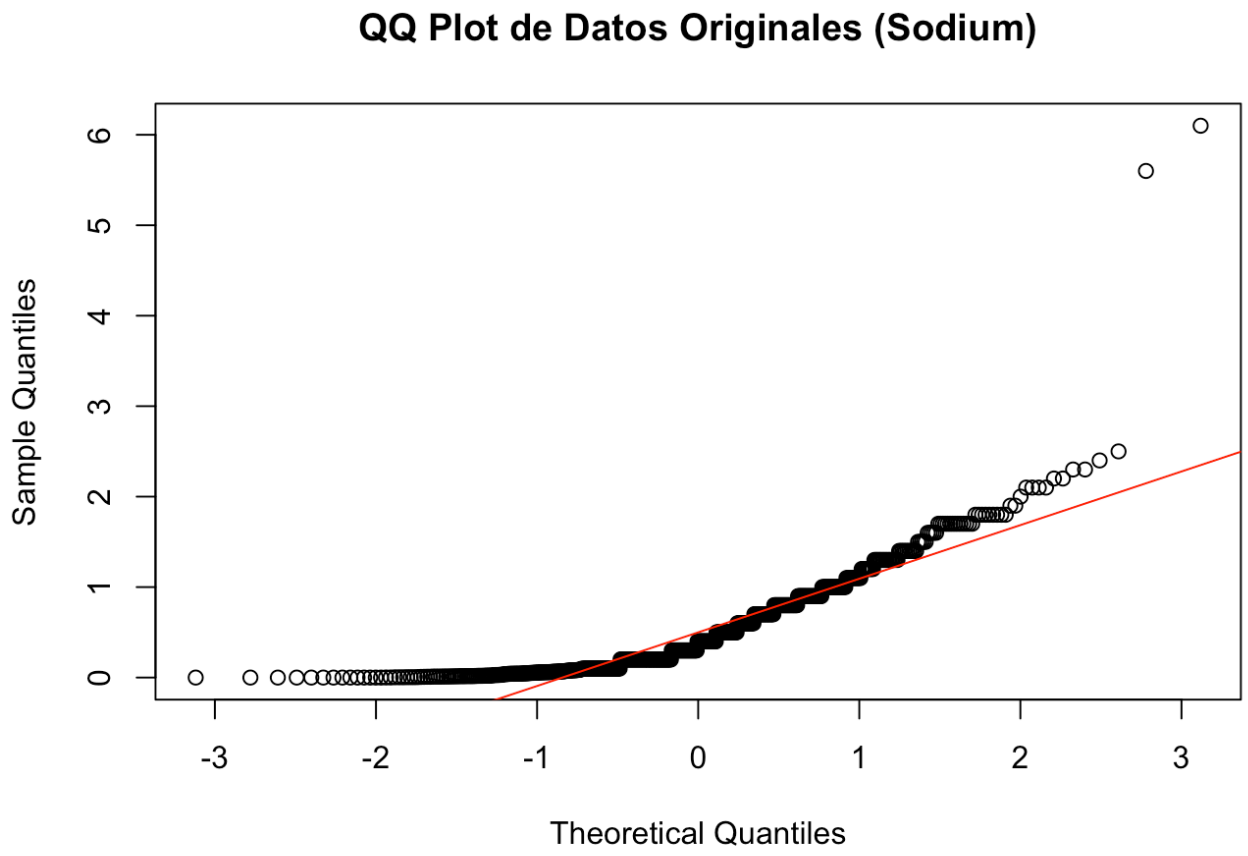
Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis.

```
original <- data$Sodium
transformado <- yjPower(original, lambda = yj_transform$lambda)

calcular_medidas <- function(datos) {
  medidas <- list(
    Minimo = min(datos),
    Maximo = max(datos),
    Media = mean(datos),
    Mediana = median(datos),
    Cuartil_1 = quantile(datos, 0.25),
    Cuartil_3 = quantile(datos, 0.75),
    Sesgo = skewness(datos),
    Curtosis = kurtosis(datos))
  return(medidas)
}

medidas_originales <- calcular_medidas(original)
medidas_transformados <- calcular_medidas(transformado)

comparacion_medidas <- data.frame(
  Medida = names(medidas_originales),
  Original = unlist(medidas_originales),
  Transformado = unlist(medidas_transformados)
)

print(comparacion_medidas)
```

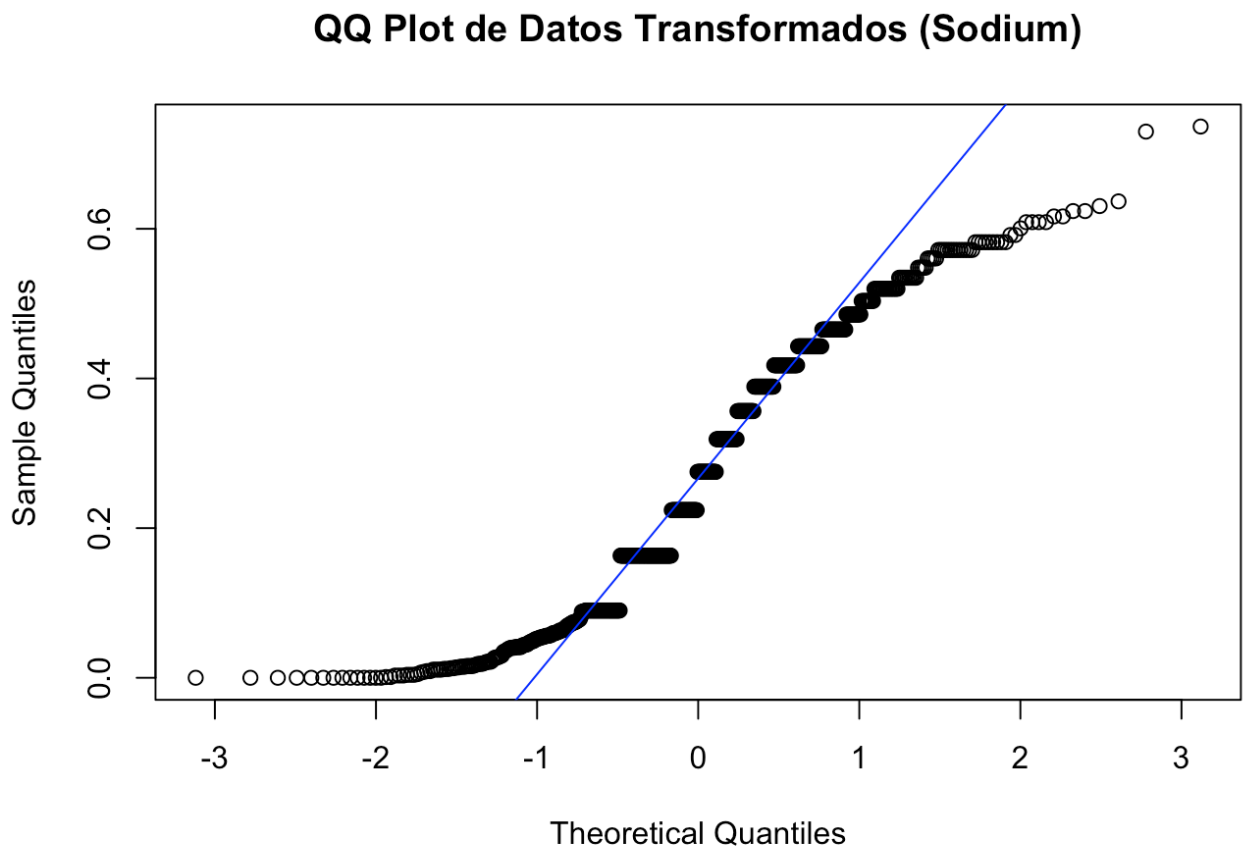
```
##                  Medida   Original Transformado
## Minimo           Minimo  0.0000000   0.00000000
## Maximo           Maximo  6.1000000   0.73663395
## Media             Media  0.5732051   0.27017274
## Mediana         Mediana  0.4000000   0.27521133
## Cuartil_1.25% Cuartil_1  0.1000000   0.08990411
## Cuartil_3.75% Cuartil_3  0.9000000   0.44291079
## Sesgo             Sesgo  2.7359990   0.18097044
## Curtosis       Curtosis 19.3626049   1.71195645
```

Extra, gráficando el QQ-Plot

```
qqnorm(original, main = "QQ Plot de Datos Originales (Sodium)")
qqline(original, col = "red")
```



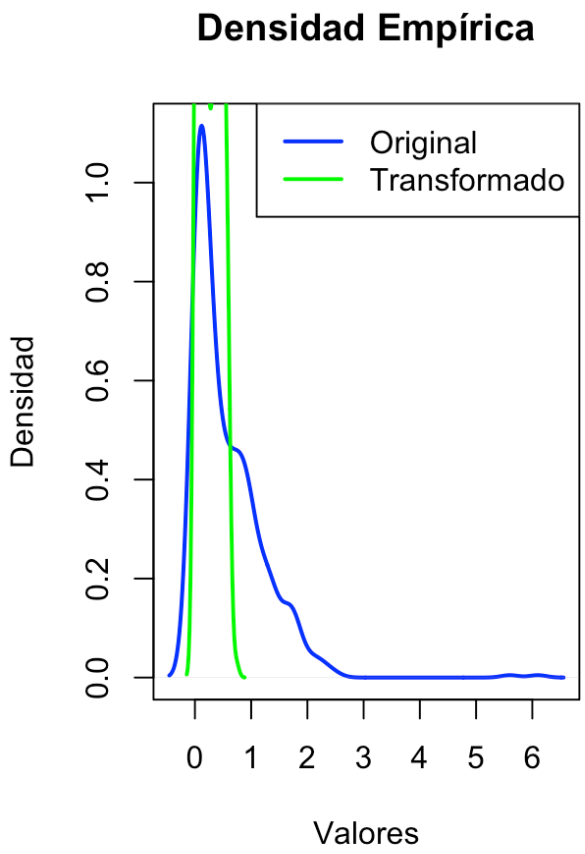

```
qqnorm(transformado, main = "QQ Plot de Datos Transformados (Sodium)")
qqline(transformado, col = "blue")
```

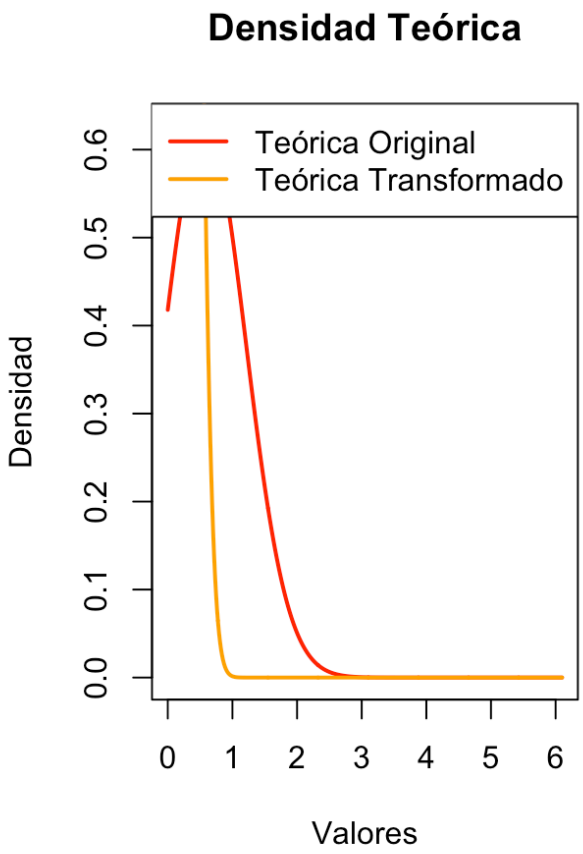


Grafica las funciones de densidad empírica y teórica de los 2

modelos obtenidos (exacto y aproximado) y los datos originales.

```
densidad_original <- density(original)
densidad_transformada <- density(transformado)

media_original <- mean(original)
sd_original <- sd(original)
media_transformada <- mean(transformado)
sd_transformada <- sd(transformado)

x_vals <- seq(min(c(original, transformado)), max(c(original, transformado)), length.out =
1000)

densidad_teorica_original <- dnorm(x_vals, mean = media_original, sd = sd_original)
densidad_teorica_transformada <- dnorm(x_vals, mean = media_transformada, sd = sd_transfor
mada)

par(mfrow = c(1, 2))
plot(densidad_original, main = "Densidad Empírica", col = "blue", lwd = 2, xlab = "Valore
s", ylab = "Densidad")
lines(densidad_transformada, col = "green", lwd = 2)
legend("topright", legend = c("Original", "Transformado"), col = c("blue", "green"), lwd =
2)

plot(x_vals, densidad_teorica_original, type = "l", main = "Densidad Teórica", col = "re
d", lwd = 2, xlab = "Valores", ylab = "Densidad")
lines(x_vals, densidad_teorica_transformada, col = "orange", lwd = 2)
legend("topright", legend = c("Teórica Original", "Teórica Transformado"), col = c("red",
"orange"), lwd = 2)
```





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

```
ad_original <- ad.test(original)
ad_transformado <- ad.test(transformado)

jb_original <- jarque.test(original)
jb_transformado <- jarque.test(transformado)

cat("Prueba de Anderson-Darling para los datos originales:\n")
```

```
## Prueba de Anderson-Darling para los datos originales:
```

```
print(ad_original)
```

```
##
##  Anderson-Darling normality test
##
## data:  original
## A = 24.827, p-value < 2.2e-16
```

```
cat("\nPrueba de Anderson-Darling para los datos transformados:\n")
```

```
##
## Prueba de Anderson-Darling para los datos transformados:
```

```
print(ad_transformado)
```

```
##
##  Anderson-Darling normality test
##
## data:  transformado
## A = 12.804, p-value < 2.2e-16
```

```
cat("\nPrueba de Jarque-Bera para los datos originales:\n")
```

```
##
## Prueba de Jarque-Bera para los datos originales:
```

```
print(jb_original)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  original
## JB = 6834.2, p-value < 2.2e-16
## alternative hypothesis: greater
```

```
cat("\nPrueba de Jarque-Bera para los datos transformados:\n")
```

```
##
## Prueba de Jarque-Bera para los datos transformados:
```

```
print(jb_transformado)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  transformado
## JB = 41.097, p-value = 1.191e-09
## alternative hypothesis: greater
```