# Actividad

Catherine Rojas

2024-08-20

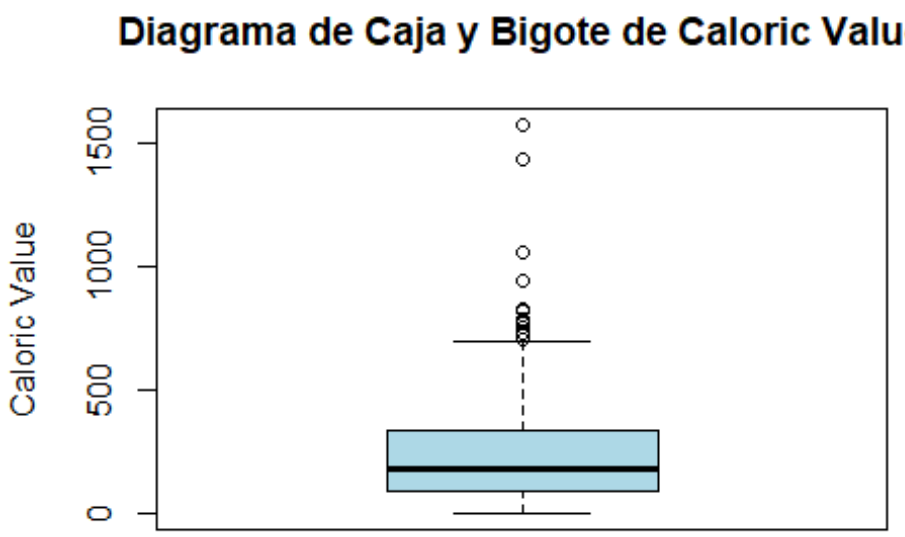
**Variable 1: Calorías**

```
library(readr)

## Warning: package 'readr' was built under R version 4.3.3

data<- read_csv("food_data_g.csv")

## New names:
## Rows: 551 Columns: 37
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter:
"," chr
## (1): food dbl (36): ...1, Unnamed: 0, Caloric Value, Fat, Saturated
Fats,
## Monounsatura...
## ℹ Use `spec()` to retrieve the full column specification for this
data. ℹ
## Specify the column types or set `show_col_types = FALSE` to quiet this
message.
## • `` -> `...1`

# Seleccionar la columna de interés
calorias <- data$`Caloric Value`
```

## 1. Para analizar datos atípicos se te sugiere:

### Graficar el diagrama de caja y bigote

```
# Graficar el diagrama de caja y bigote
boxplot(calorias, main = "Diagrama de Caja y Bigote de Caloric Value",
ylab = "Caloric Value", col = "lightblue")
```




## 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

```
# Calcular las principales medidas
summary_stats <- summary(calorias)
Q1 <- summary_stats["1st Qu."]
Q3 <- summary_stats["3rd Qu."]
media <- summary_stats["Mean"]
desviacion_estandar <- sd(calorias)
IQR <- IQR(calorias)

# Mostrar los resultados
cat("Cuartil 1 (Q1):", Q1, "\n")

## Cuartil 1 (Q1): 94.5

cat("Cuartil 3 (Q3):", Q3, "\n")

## Cuartil 3 (Q3): 337

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

## Media: 237.3593

cat("Rango Intercuartílico (IQR):", IQR, "\n")

## Rango Intercuartílico (IQR): 242.5
```

```r
cat("Desviación Estándar:", desviacion_estandar, "\n")
```

```
## Desviación Estándar: 199.2356
```

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

```r
# Calcular las cotas para identificar datos atípicos
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Identificar datos atípicos
outliers <- calorias[calorias < lower_bound | calorias > upper_bound]

# Mostrar los resultados
cat("Cota Inferior:", lower_bound, "\n")
```

```
## Cota Inferior: -269.25
```

```r
cat("Cota Superior:", upper_bound, "\n")
```

```
## Cota Superior: 700.75
```

```r
cat("Número de datos atípicos:", length(outliers), "\n")
```

```
## Número de datos atípicos: 19
```

Los valores atípicos fueron identificados fuera de las cotas de -269.25 (inferior) y 700.75 (superior).

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

```r
# Calcular la media y desviación estándar
media <- mean(calorias)
desviacion_estandar <- sd(calorias)

# Calcular las cotas para identificar datos atípicos
lower_bound <- media - 3 * desviacion_estandar
upper_bound <- media + 3 * desviacion_estandar

# Identificar datos atípicos
outliers <- calorias[calorias < lower_bound | calorias > upper_bound]

# Mostrar los resultados
cat("Cota Inferior (3 desviaciones estándar por debajo de la media):",
lower_bound, "\n")
```

```
## Cota Inferior (3 desviaciones estándar por debajo de la media): -
360.3474
```

```r
cat("Cota Superior (3 desviaciones estándar por encima de la media):",
upper_bound, "\n")
```

```
## Cota Superior (3 desviaciones estándar por encima de la media):
835.0661
```

```r
cat("Número de datos atípicos:", length(outliers), "\n")
```

```
## Número de datos atípicos: 6
```

Los valores atípicos según este criterio fueron identificados fuera de las cotas de -360.35 (inferior) y 835.07 (superior).

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

```r
# Calcular los cuartiles y el rango intercuartílico (IQR)
Q1 <- quantile(calorias, 0.25, na.rm = TRUE)
Q3 <- quantile(calorias, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1

# Calcular las cotas para identificar datos extremos
lower_bound_extreme <- Q1 - 3 * IQR
upper_bound_extreme <- Q3 + 3 * IQR

# Identificar datos extremos
extreme_values <- calorias[calorias < lower_bound_extreme | calorias >
upper_bound_extreme]

# Mostrar los resultados
cat("Cota Inferior (3 IQR por debajo de Q1):", lower_bound_extreme, "\n")
```

```
## Cota Inferior (3 IQR por debajo de Q1): -633
```

```r
cat("Cota Superior (3 IQR por encima de Q3):", upper_bound_extreme, "\n")
```

```
## Cota Superior (3 IQR por encima de Q3): 1064.5
```

```r
cat("Número de datos extremos:", length(extreme_values), "\n")
```

```
## Número de datos extremos: 2
```

Los valores extremos fueron identificados fuera de las cotas de -633 (inferior) y 1064.5 (superior).

### Interpreta los resultados obtenidos y argumenta sobre el comportamiento de los datos atípicos y extremos en la variable seleccionada

**Sesgo y Curtosis:** - El coeficiente de sesgo (1.9227) indica que la distribución de los valores calóricos está sesgada a la derecha, lo que significa que hay más valores extremos en la cola derecha de la distribución. - El coeficiente de curtosis (9.7608) es

significativamente mayor que 3, lo que sugiere que la distribución tiene colas más pesadas y un pico más alto que una distribución normal (leptocúrtica).

**Normalidad:** - Las pruebas de normalidad rechazaron la hipótesis nula de que los datos siguen una distribución normal, lo que es consistente con la alta curtosis y el sesgo positivo observados.

Los datos de "Caloric Value" presentan varios valores atípicos y algunos extremos que no se ajustan bien a una distribución normal. La presencia de un sesgo positivo y una alta curtosis indican una distribución asimétrica con colas pesadas.

## 2. 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.

**H0 (Hipótesis Nula): Los datos siguen una distribución normal.**

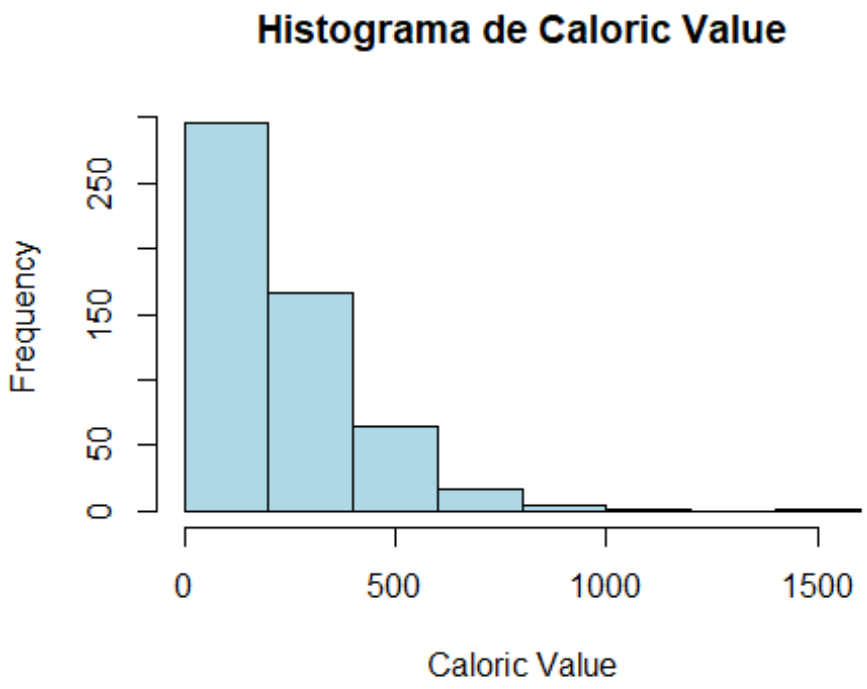
**H1 (Hipótesis Alternativa): Los datos no siguen una distribución normal.**

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

## Warning: package 'tseries' was built under R version 4.3.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

# Prueba de Anderson-Darling
ad_test <- ad.test(calorias)

# Prueba de Jarque-Bera
jb_test <- jarque.bera.test(calorias)

# Mostrar los resultados
cat("Resultados de la prueba de Anderson-Darling:\n")

## Resultados de la prueba de Anderson-Darling:

print(ad_test)

##
##  Anderson-Darling normality test
##
## data:  calorias
## A = 15.326, p-value < 2.2e-16

cat("\nResultados de la prueba de Jarque-Bera:\n")
```

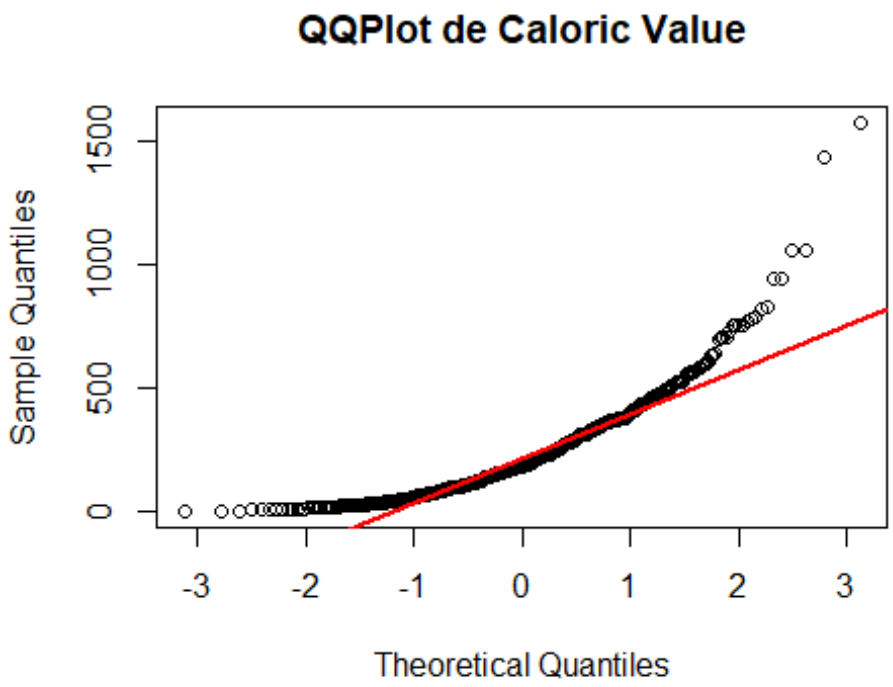
```
##
## Resultados de la prueba de Jarque-Bera:

print(jb_test)

##
##  Jarque Bera Test
##
## data:  calorias
## X-squared = 1388.9, df = 2, p-value < 2.2e-16
```

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

```
# Graficar los datos en un histograma
hist(calorias, main = "Histograma de Caloric Value", xlab = "Caloric
Value", col = "lightblue", border = "black")
```




```
# Graficar el QQPlot
qqnorm(calorias, main = "QQPlot de Caloric Value")
qqline(calorias, col = "red", lwd = 2)  # Agregar la línea de referencia
```




## Calcula el coeficiente de sesgo y el coeficiente de curtosis

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```r
# Calcular el coeficiente de sesgo
coef_sesgo <- skewness(calorias)

# Calcular el coeficiente de curtosis
coef_curtosis <- kurtosis(calorias)

# Mostrar los resultados
cat("Coeficiente de Sesgo:", coef_sesgo, "\n")
```

```
## Coeficiente de Sesgo: 1.917503
```

```r
cat("Coeficiente de Curtosis:", coef_curtosis, "\n")
```

```
## Coeficiente de Curtosis: 6.725447
```

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

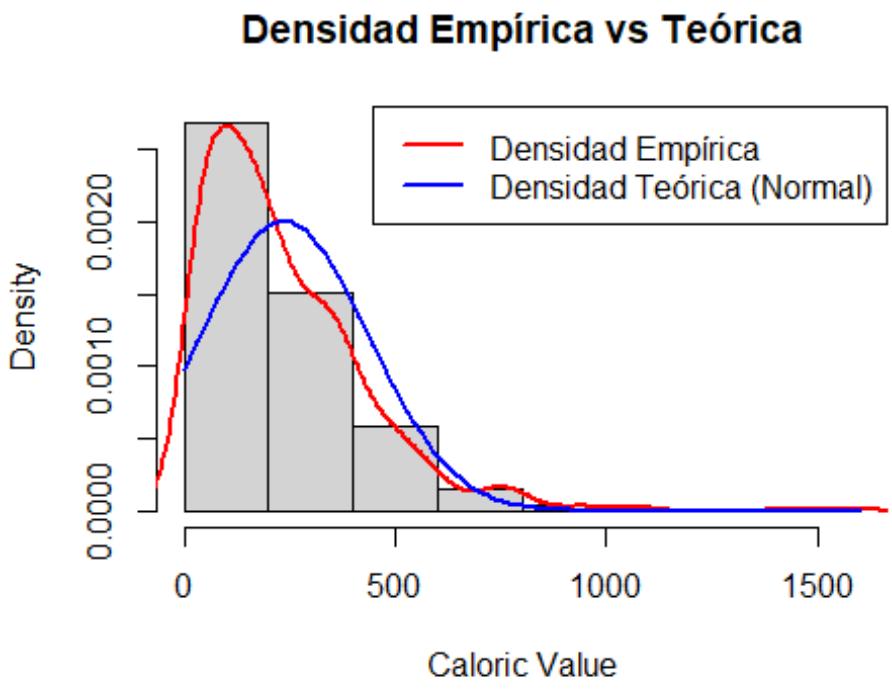
```r
# Calcular la media
media <- mean(calorias, na.rm = TRUE)

# Calcular la mediana
mediana <- median(calorias, na.rm = TRUE)
```

```r
# Calcular el rango medio
rango_medio <- (min(calorias, na.rm = TRUE) + max(calorias, na.rm =
TRUE)) / 2

# Mostrar los resultados
cat("Media de Caloric Value:", media, "\n")

## Media de Caloric Value: 237.3593

cat("Mediana de Caloric Value:", mediana, "\n")

## Mediana de Caloric Value: 186

cat("Rango Medio de Caloric Value:", rango_medio, "\n")

## Rango Medio de Caloric Value: 790.5
```

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

hist(datos,freq=FALSE) lines(density(datos),col="red")
curve(dnorm(x,mean=mean(datos,sd=sd(datos)), from=-6, to=6, add=TRUE,
col="blue",lwd=2)

```r
# Histograma con la densidad empírica
hist(calorias, freq = FALSE, main = "Densidad Empírica vs Teórica", xlab
= "Caloric Value", col = "lightgray", border = "black")

# Agregar la línea de densidad empírica
lines(density(calorias), col = "red", lwd = 2)

# Agregar la curva de densidad teórica (suponiendo normalidad)
curve(dnorm(x, mean = mean(calorias), sd = sd(calorias)),
      col = "blue", lwd = 2, add = TRUE)

# Agregar una leyenda para identificar las líneas
legend("topright", legend = c("Densidad Empírica", "Densidad Teórica
(Normal)"),
       col = c("red", "blue"), lwd = 2)
```




## Interpreta los gráficos y los resultados obtenidos en cada punto con vías a indicar si hay normalidad de los datos

**Histograma de Caloric Value** - El histograma muestra una distribución asimétrica hacia la derecha, lo que sugiere que la mayoría de los valores están concentrados en los rangos más bajos (cerca de 0 a 500 calorías), mientras que hay una cola larga hacia valores más altos.

**QQPlot de Caloric Value** - El QQPlot compara los cuantiles de los datos con los cuantiles de una distribución normal teórica. Se observa que los puntos se desvían significativamente de la línea de referencia (línea roja) en las colas, especialmente en la parte superior derecha, lo que indica que los datos tienen más valores extremos (outliers) de lo que se esperaría bajo una distribución normal.

**Densidad Empírica vs Densidad Teórica** - La línea roja representa la densidad empírica (la distribución real de los datos), mientras que la línea azul representa la densidad teórica de una distribución normal con la misma media y desviación estándar. - Se observa que la densidad empírica se desvía considerablemente de la densidad teórica, especialmente en la cola derecha, lo que indica nuevamente que los datos no siguen una distribución normal.

**Anderson-Darling Test y Jarque-Bera Test:** - Ambas pruebas arrojan p-valores extremadamente bajos (menores a 2.2e-16), lo que nos lleva a rechazar la hipótesis nula de que los datos siguen una distribución normal.

**Sesgo (1.9227):** - Un sesgo positivo de 1.9227 indica una asimetría hacia la derecha.

**Curtosis (9.7608):** - Una curtosis de 9.7608, que es mucho mayor que 3, indica una distribución con colas más pesadas y un pico más pronunciado que una distribución normal.

### Comenta las características encontradas:

#### Considera alejamientos de normalidad por simetría, curtosis

Los datos de "Caloric Value" se alejan considerablemente de una distribución normal debido a la fuerte asimetría positiva y la alta curtosis.

#### Comenta si hay aparente influencia de los datos atípicos en la normalidad de los datos

Los datos atípicos, especialmente aquellos en la cola derecha de la distribución, juegan un papel crucial en este alejamiento, contribuyendo a la forma leptocúrtica de la distribución. Estos outliers no solo afectan las medidas de tendencia central (como la media), sino que también distorsionan la percepción general de la distribución.

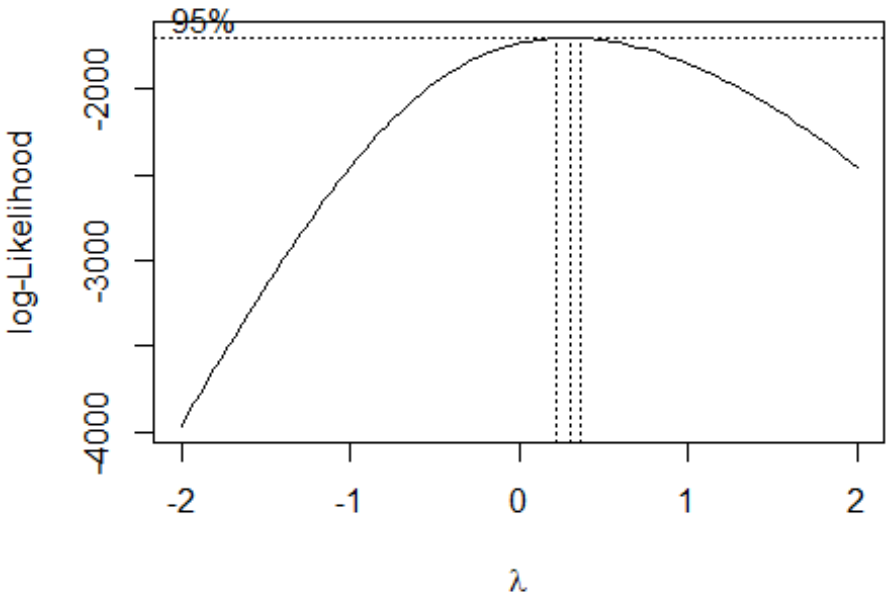
#### Emite una conclusión sobre la normalidad de los datos. Se debe argumentar en términos de los 3 puntos analizados: las pruebas de normalidad, los gráficos y las medidas.

Los gráficos, junto con las pruebas de normalidad y los coeficientes de sesgo y curtosis, indican claramente que los datos de "Caloric Value" no siguen una distribución normal. La distribución es asimétrica hacia la derecha, con valores extremos en la cola derecha que influyen significativamente en la forma de la distribución.

## 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.

```
# Cargar la librería 'MASS' para 'boxcox'
library(MASS)

x <- data$`Caloric Value`

# Box Cox
bc <- boxcox((x+1) ~ 1)
```




```r
# Encontrar el mejor lambda
lambda <- bc$x[which.max(bc$y)]
lambda

## [1] 0.3030303

# Transformación Box-Cox exacta
trans_exact <- (x^lambda - 1) / lambda

# Aplicar la transformación Box-Cox aproximada
trans_aprox <- sqrt(x)
```

## Escribe las ecuaciones de los modelos de transformación encontrados.

**a)Transformación Exacta de Box-Cox:**

$$x_{\text{transformado}} = \frac{x^{\lambda} - 1}{\lambda} \quad \text{si } \lambda \neq 0$$

$$x_{\text{transformado}} = \frac{x^{0.3030303} - 1}{0.3030303} \quad \text{si } \lambda \neq 0$$

Donde:

- $x_{\text{transformado}}$ es el valor transformado de la variable calorias
- $x$ es el valor original de la variable calorias

- $\lambda$ es el valor óptimo encontrado que maximiza la log-verosimilitud en el análisis de Box-Cox.

En este caso, $\lambda$ = {0.3030303}

**b)Transformación Aproximada de Box-Cox:**

$$x_{\text{transformado}} = \sqrt{x}$$

Donde:

- $x_{\text{transformado}}$ es el valor transformado de la variable calorias
- $x$ es el valor original de la variable

La elección entre la transformación exacta y la aproximada depende del valor óptimo de ($\lambda$ ) encontrado en el análisis de Box-Cox.

- Si ($\lambda$ ) es cercano a 0, la transformación aproximada se puede utilizar como una simplificación razonable. Sin embargo, si ($\lambda$ ) es significativamente diferente de 0, es preferible utilizar la transformación exacta para obtener una corrección más precisa.

## 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.**

```
# Resumen de los datos originales
cat("\nMedidas de los datos originales:\n")

##
## Medidas de los datos originales:

resumen_x <- summary(x)
resumen_x

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     3.0    94.5   186.0   237.4   337.0  1578.0

library(e1071)

# Sesgo y curtosis de los datos originales
sesgo_x <- skewness(x)
curtosis_x <- kurtosis(x)

cat("\nSesgo con los datos originales:\n", sesgo_x, "\n")

##
## Sesgo con los datos originales:
##  1.917503
```

```r
cat("\nCurtosis con los datos originales:\n", curtosis_x, "\n")
```

```
##
## Curtosis con los datos originales:
##  6.725447
```

```r
# Resumen de los datos transformados con Box-Cox exacto
cat("\nMedidas con Box-Cox exacto:\n")
```

```
##
## Medidas con Box-Cox exacto:
```

```r
resumen_trans_exact <- summary(trans_exact)
resumen_trans_exact
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.304   9.796  12.778  12.736  15.951  27.435
```

```r
# Sesgo y curtosis de los datos transformados con Box-Cox exacto
sesgo_trans_exact <- skewness(trans_exact)
curtosis_trans_exact <- kurtosis(trans_exact)

cat("\nSesgo con Box-Cox exacto:\n", sesgo_trans_exact, "\n")
```

```
##
## Sesgo con Box-Cox exacto:
##  -0.02223906
```

```r
cat("\nCurtosis con Box-Cox exacto:\n", curtosis_trans_exact, "\n")
```

```
##
## Curtosis con Box-Cox exacto:
##  -0.1868361
```

```r
# Resumen de los datos transformados con Box-Cox aproximado
cat("\nMedidas con Box-Cox aproximado:\n")
```

```
##
## Medidas con Box-Cox aproximado:
```

```r
resumen_trans_aprox <- summary(trans_aprox)
resumen_trans_aprox
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.732   9.721  13.638  14.133  18.358  39.724
```

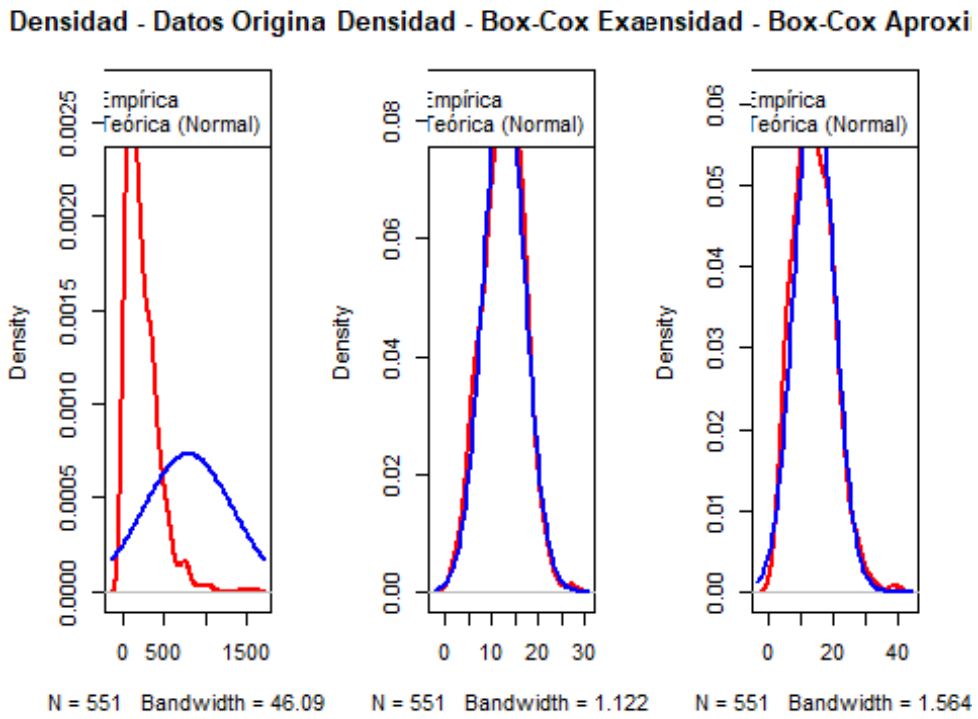
```r
# Sesgo y curtosis de los datos transformados con Box-Cox aproximado
sesgo_trans_aprox <- skewness(trans_aprox)
curtosis_trans_aprox <- kurtosis(trans_aprox)

cat("\nSesgo con Box-Cox aproximado:\n", sesgo_trans_aprox, "\n")
```

```
##
## Sesgo con Box-Cox aproximado:
##  0.476366

cat("\nCurtosis con Box-Cox aproximado:\n", curtosis_trans_aprox, "\n")

##
## Curtosis con Box-Cox aproximado:
##  0.34169

# Tabla con resultados
tabla_resultados <- data.frame(
  "Medida" = c("Min", "1st Qu.", "Median", "Mean", "3rd Qu.", "Max",
"Sesgo", "Curtosis"),
  "Datos_Originales" = c(resumen_x["Min."], resumen_x["1st Qu."],
resumen_x["Median"], resumen_x["Mean"], resumen_x["3rd Qu."],
resumen_x["Max."], sesgo_x, curtosis_x),
  "Box-Cox_Exacto" = c(resumen_trans_exact["Min."],
resumen_trans_exact["1st Qu."], resumen_trans_exact["Median"],
resumen_trans_exact["Mean"], resumen_trans_exact["3rd Qu."],
resumen_trans_exact["Max."], sesgo_trans_exact, curtosis_trans_exact),
  "Box-Cox_Aproximado" = c(resumen_trans_aprox["Min."],
resumen_trans_aprox["1st Qu."], resumen_trans_aprox["Median"],
resumen_trans_aprox["Mean"], resumen_trans_aprox["3rd Qu."],
resumen_trans_aprox["Max."], sesgo_trans_aprox, curtosis_trans_aprox)
)

tabla_resultados

##     Medida Datos_Originales Box.Cox_Exacto Box.Cox_Aproximado
## 1      Min         3.000000     1.30358470           1.732051
## 2  1st Qu.        94.500000     9.79568908           9.721077
## 3   Median       186.000000    12.77848538          13.638182
## 4     Mean       237.359347    12.73576346          14.132788
## 5  3rd Qu.       337.000000    15.95138551          18.357540
## 6      Max      1578.000000    27.43528700          39.724048
## 7    Sesgo         1.917503    -0.02223906           0.476366
## 8 Curtosis         6.725447    -0.18683613           0.341690
```

**Grafica las funciones de densidad empírica y teórica de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.**

```
# Cargar las librerías necesarias
library(MASS)

# Gráfico de densidad empírica y teórica para los datos originales
par(mfrow = c(1, 3))  # Configura la ventana de gráficos para 3 gráficos
en una fila

# Datos originales
plot(density(x), main = "Densidad - Datos Originales", col = "red", lwd =
```

```r
2)
curve(dnorm(x, mean = mean(x), sd = sd(x)), col = "blue", lwd = 2, add =
TRUE)
legend("topright", legend = c("Empírica", "Teórica (Normal)"), col =
c("red", "blue"), lwd = 2)

# Datos transformados con Box-Cox exacto
plot(density(trans_exact), main = "Densidad - Box-Cox Exacto", col =
"red", lwd = 2)
curve(dnorm(x, mean = mean(trans_exact), sd = sd(trans_exact)), col =
"blue", lwd = 2, add = TRUE)
legend("topright", legend = c("Empírica", "Teórica (Normal)"), col =
c("red", "blue"), lwd = 2)

# Datos transformados con Box-Cox aproximado
plot(density(trans_aprox), main = "Densidad - Box-Cox Aproximado", col =
"red", lwd = 2)
curve(dnorm(x, mean = mean(trans_aprox), sd = sd(trans_aprox)), col =
"blue", lwd = 2, add = TRUE)
legend("topright", legend = c("Empírica", "Teórica (Normal)"), col =
c("red", "blue"), lwd = 2)
```




```r
# Restablecer la ventana gráfica
par(mfrow = c(1, 1))
```

**Realiza la prueba de normalidad de Anderson-Darling y de Jarque Bera para los datos transformados y los originales** Para todas las pruebas de normalidad, la hipótesis nula $[H_0]$ es que los datos siguen una distribución normal.

```
library(nortest)

# Prueba de normalidad de Anderson-Darling para los datos originales
ad_test_x <- ad.test(x)
ad_test_x

##
##  Anderson-Darling normality test
##
## data:  x
## A = 15.326, p-value < 2.2e-16

# Prueba de normalidad de Anderson-Darling para los datos con Box-Cox
exacto
ad_test_exact <- ad.test(trans_exact)
ad_test_exact

##
##  Anderson-Darling normality test
##
## data:  trans_exact
## A = 0.57755, p-value = 0.1328

# Prueba de normalidad de Anderson-Darling para los datos con Box-Cox
aproximado
ad_test_aprox <- ad.test(trans_aprox)
ad_test_aprox

##
##  Anderson-Darling normality test
##
## data:  trans_aprox
## A = 1.2496, p-value = 0.002964

library(tseries)

# Prueba de normalidad de Jarque-Bera para los datos originales
jb_test_x <- jarque.bera.test(x)
jb_test_x

##
##  Jarque Bera Test
##
## data:  x
## X-squared = 1388.9, df = 2, p-value < 2.2e-16
```

```r
# Prueba de normalidad de Jarque-Bera para Box-Cox exacto
jb_test_exact <- jarque.bera.test(trans_exact)
jb_test_exact
```

```
##
##  Jarque Bera Test
##
## data:  trans_exact
## X-squared = 0.76166, df = 2, p-value = 0.6833
```

```r
# Prueba de normalidad de Jarque-Bera para Box-Cox aproximado
jb_test_aprox <- jarque.bera.test(trans_aprox)
jb_test_aprox
```

```
##
##  Jarque Bera Test
##
## data:  trans_aprox
## X-squared = 23.828, df = 2, p-value = 6.697e-06
```

```r
## Tabla con resultados
resultados_normalidad <- data.frame(
  Test = c("Anderson-Darling", "Anderson-Darling", "Anderson-Darling",
"Jarque-Bera", "Jarque-Bera", "Jarque-Bera"),
  Transformation = c("Original", "Box-Cox Exacto", "Box-Cox Aproximado",
"Original", "Box-Cox Exacto", "Box-Cox Aproximado"),
  Statistic = c(ad_test_x$statistic, ad_test_exact$statistic,
ad_test_aprox$statistic,
                jb_test_x$statistic, jb_test_exact$statistic,
jb_test_aprox$statistic),
  P_Value = c(ad_test_x$p.value, ad_test_exact$p.value,
ad_test_aprox$p.value,
              jb_test_x$p.value, jb_test_exact$p.value,
jb_test_aprox$p.value)
)

resultados_normalidad
```

```
##               Test     Transformation    Statistic      P_Value
## 1 Anderson-Darling           Original   15.3256259 3.700000e-24
## 2 Anderson-Darling     Box-Cox Exacto    0.5775492 1.328423e-01
## 3 Anderson-Darling Box-Cox Aproximado    1.2496291 2.964427e-03
## 4      Jarque-Bera           Original 1388.9024936 0.000000e+00
## 5      Jarque-Bera     Box-Cox Exacto    0.7616580 6.832947e-01
## 6      Jarque-Bera Box-Cox Aproximado   23.8277649 6.696789e-06
```

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

```r
# Calcular la frecuencia de ceros
num_zeros <- sum(x == 0)
total_values <- length(x)
```

```r
percent_zeros <- (num_zeros / total_values) * 100

# Mostrar la frecuencia de ceros
cat("Número de ceros:", num_zeros, "\n")

## Número de ceros: 0

cat("Porcentaje de ceros:", percent_zeros, "%\n")

## Porcentaje de ceros: 0 %
```

No se quitan datos atipicos, pues son datos que reflejan infoormación significativa de la variable calorías y no existen ceros anomalos.

```r
# Filtrar los datos para eliminar valores extremos
calorias_sin_extremos <- x[x >= lower_bound_extreme & x <=
upper_bound_extreme]

# Mostrar la cantidad de datos después de eliminar los extremos
cat("Número de datos originales:", length(x), "\n")

## Número de datos originales: 551

cat("Número de datos después de eliminar extremos:",
length(calorias_sin_extremos), "\n")

## Número de datos después de eliminar extremos: 549
```

## Comenta la normalidad de las transformaciones obtenidas. Utiliza como argumento de normalidad:

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

**Transformación Exacta de Box-Cox:**

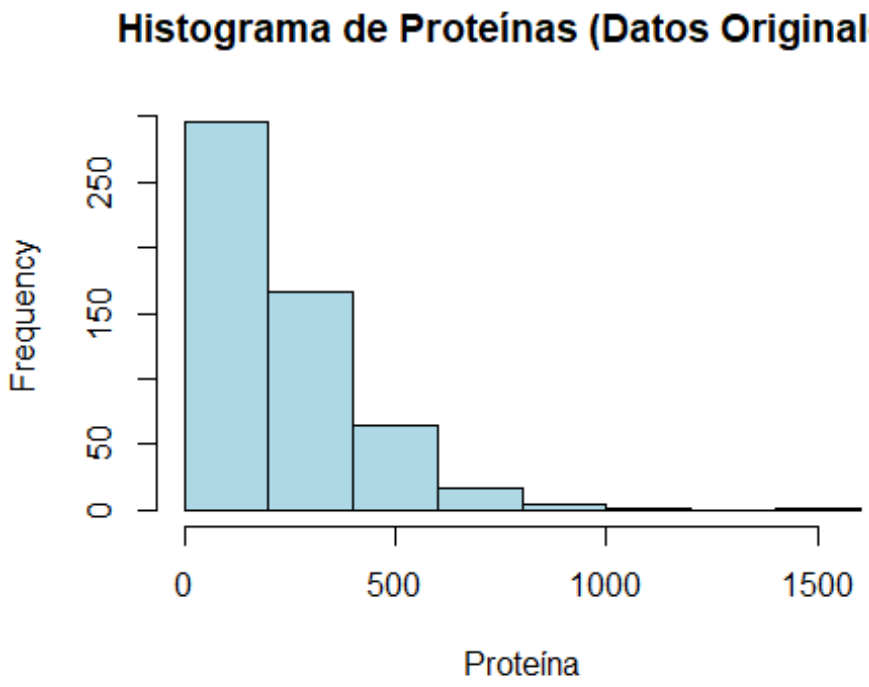
Se obtuvó una mejora significativa en la simetría y la forma de la distribución. El sesgo prácticamente desaparece y la curtosis se aproxima a la de una distribución normal, lo que sugiere que esta transformación es más efectiva para normalizar los datos.
Transformación Aproximada (Raíz Cuadrada):

Aunque mejora la simetría y reduce la curtosis, no es tan efectiva como la transformación exacta de Box-Cox. El sesgo sigue presente y la curtosis, aunque mejorada, aún indica una ligera leptocurtosis.

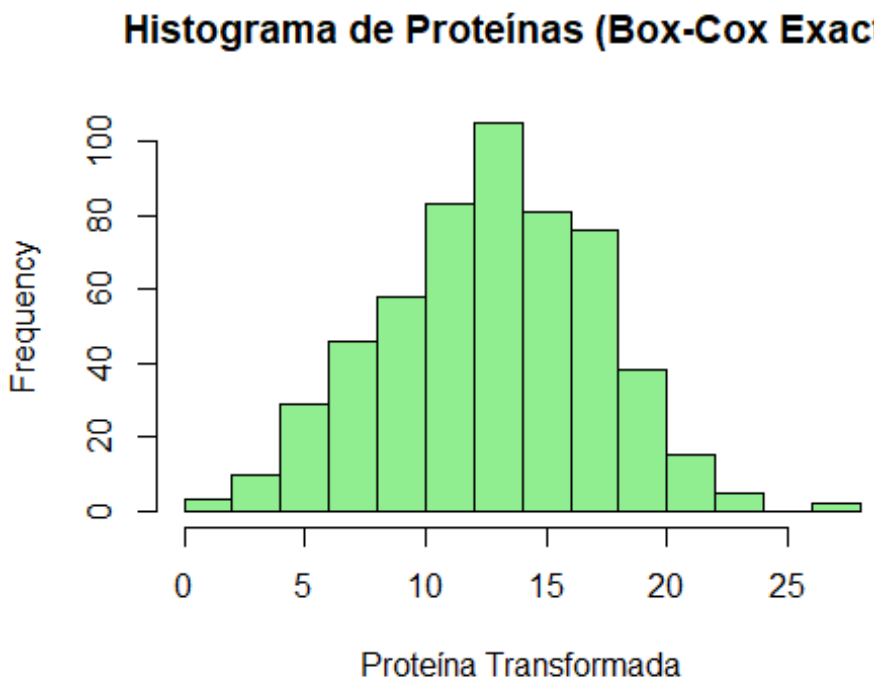
Basado en las métricas de sesgo y curtosis, la transformación exacta de Box-Cox parece ser la mejor opción para aproximar la distribución de “Caloric Value” a una distribución normal.

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

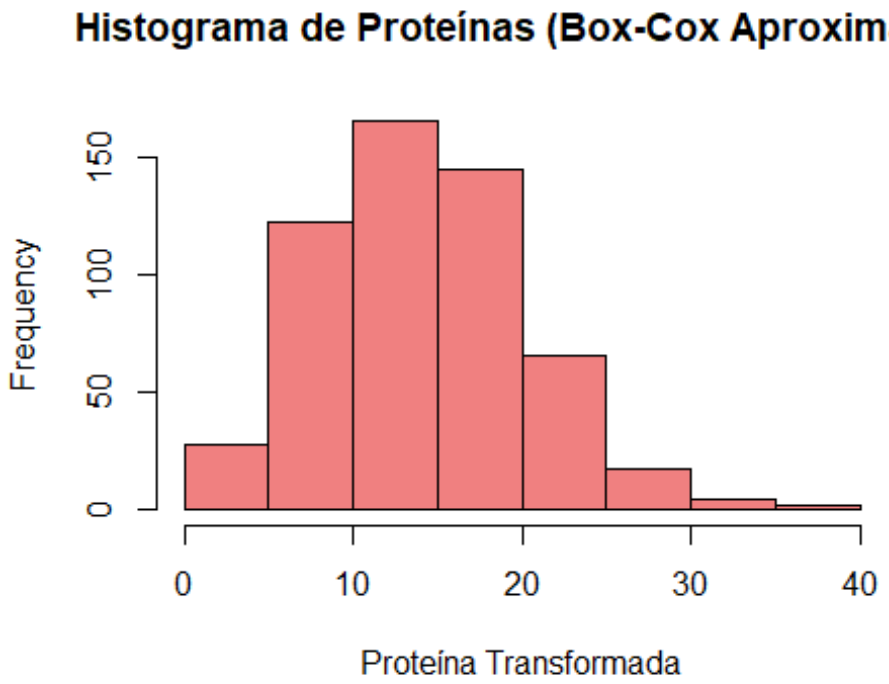
```r
# Histograma de los datos originales
hist(x, main = "Histograma de Proteínas (Datos Originales)", xlab =
"Proteína", col = "lightblue", border = "black")
```




```r
# Histograma de la transformación Box-Cox exacta
hist(trans_exact, main = "Histograma de Proteínas (Box-Cox Exacta)", xlab
= "Proteína Transformada", col = "lightgreen", border = "black")
```




```r
# Histograma de la transformación Box-Cox aproximada
hist(trans_aprox, main = "Histograma de Proteínas (Box-Cox Aproximada)",
xlab = "Proteína Transformada", col = "lightcoral", border = "black")
```



**Interpreta la prueba de normalidad de Anderson-Darling y Jarque Bera para los datos transformados y los originales**

Datos Originales: Las pruebas confirman una desviación significativa de la normalidad, lo que está alineado con las observaciones de alto sesgo y curtosis.

Transformación Exacta de Box-Cox: Ambas pruebas sugieren que esta transformación ha sido efectiva en normalizar los datos, ya que los p-valores no permiten rechazar la hipótesis nula de normalidad.

Transformación Aproximada (Raíz Cuadrada): Aunque la transformación ha mejorado la normalidad de los datos en comparación con los originales, no ha sido suficiente para hacer que los datos sean normales, según los resultados de las pruebas.

Por lo tanto, la transformación exacta de Box-Cox es la mejor opción si el objetivo es lograr una distribución lo más cercana a la normalidad posible.

**Indica posibilidades de motivos de alejamiento de normalidad (sesgo, curtosis, datos atípicos, etc)**

Un sesgo significativo lleva a que la media se desplace hacia la cola de la distribución, haciendo que la distribución no sea simétrica alrededor de la media

**La alta curtosis es indicativa de la presencia de valores extremos (outliers) que inflan las colas de la distribución y contribuyen a que se desvíe de la normalidad.**
La presencia de outliers no solo distorsiona la forma de la distribución, sino que también afecta las pruebas de normalidad, que son sensibles a estos valores extremos.

El alejamiento de la normalidad en los datos de “Caloric Value” se puede atribuir a varios factores, incluidos un sesgo positivo significativo, alta curtosis y la presencia de outliers. Cada uno de estos factores contribuye de manera diferente a distorsionar la forma de la distribución, alejándola de la simetría y forma de campana características de una distribución normal.

## 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.

Mejor Transformación: La transformación exacta de Box-Cox es la mejor opción para normalizar los datos de “Caloric Value”.

Redujo el sesgo a casi cero, indicando una distribución muy simétrica. Ajustó la curtosis a un valor cercano a 3, lo que sugiere una distribución de colas normales y sin picos anómalos. Pasó las pruebas de normalidad (Anderson-Darling y Jarque-Bera), lo

que confirma que la distribución transformada es consistente con una distribución normal.