4. Explorando bases

Jacobo Hirsch Rodriguez

2024-08-14

1. Baja el archivo de trabajo: datos de McDonald Download McDonald

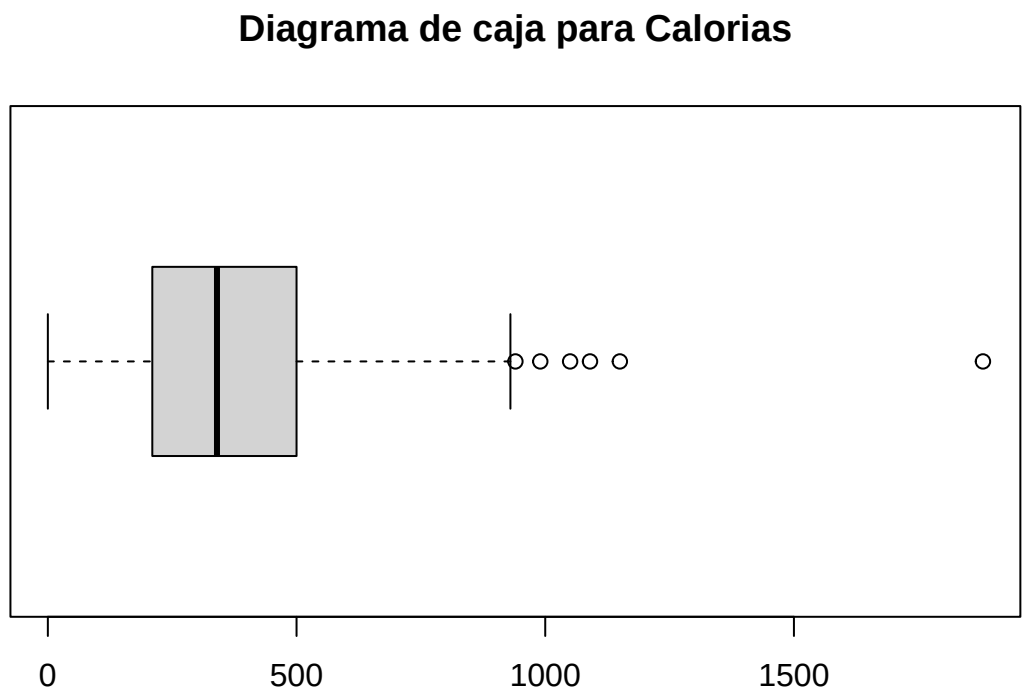
```
M=read.csv("./mc-donalds-menu.csv") #leer la base de datos
```

vamos a analizar la variable calorías y sodio

Primero vamos a analizar Calorías:

1_a) Diagrama de caja y bigote:

```
boxplot(M$Calories, main="Diagrama de caja para Calorias", horizontal=TRUE)
```



2_a) Cálculo del rango intercuartílico y cuartiles:

```r
q1 <- quantile(M$Calories, 0.25)

q3 <- quantile(M$Calories, 0.75)

ri <- q3 - q1
```

3_a)Identificación de datos atípicos con 1.5 rangos intercuartílicos:

```r
lim_inf <- q1 - 1.5 * ri
lim_sup <- q3 + 1.5 * ri

outliers_ri <- M$Calories[M$Calories < lim_inf | M$Calories > lim_sup]
outliers_ri
```

```
## [1] 1090 1150  990 1050  940 1880
```

4_a) Identificación de datos atípicos con 3 desviaciones estándar:

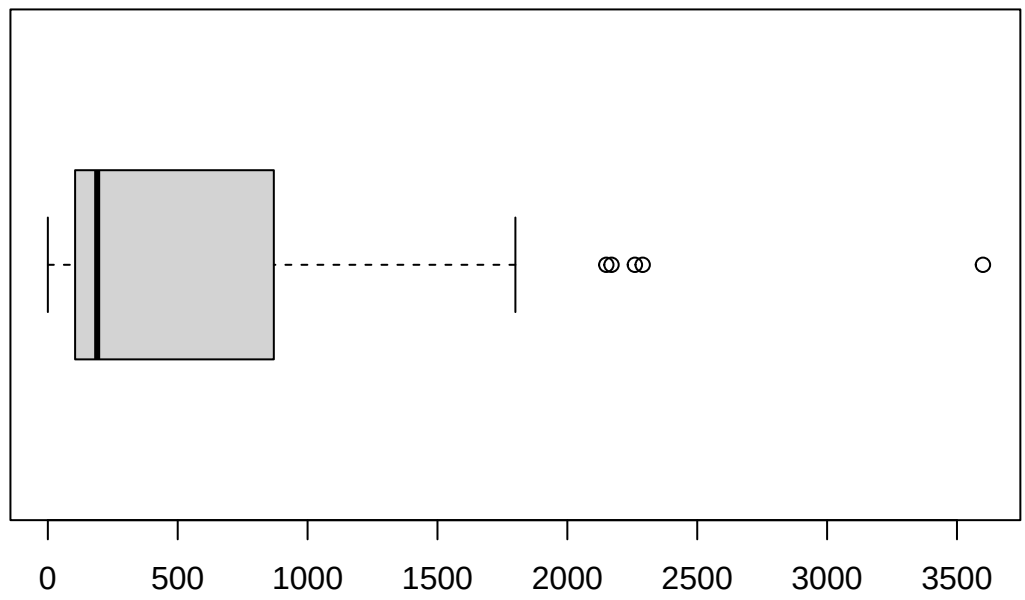
```r
mean_cal <- mean(M$Calories)
sd_cal <- sd(M$Calories)

lim_inf_sd <- mean_cal - 3 * sd_cal
lim_sup_sd <- mean_cal + 3 * sd_cal

outliers_sd <- M$Calories[M$Calories < lim_inf_sd | M$Calories > lim_sup_sd]
outliers_sd
```

```
## [1] 1090 1150 1880
```

ahora lo haremos para sodio:

1_b) Diagrama de caja y bigote:

```r
boxplot(M$Sodium, main="Diagrama de caja para Sodio", horizontal=TRUE)
```



2_b) Cálculo del rango intercuartílico y cuartiles:

```
q1_sodium <- quantile(M$Sodium, 0.25)
q3_sodium <- quantile(M$Sodium, 0.75)
ri_sodium <- q3_sodium - q1_sodium
```

3_b)Identificación de datos atípicos con 1.5 rangos intercuartílicos:

```
lim_inf_sodium <- q1_sodium - 1.5 * ri_sodium
lim_sup_sodium <- q3_sodium + 1.5 * ri_sodium

outliers_ri_sodium <- M$Sodium[M$Sodium < lim_inf_sodium | M$Sodium > lim_sup_sodium]
outliers_ri_sodium
```

```
## [1] 2150 2260 2170 2290 3600
```

4_b) Identificación de datos atípicos con 3 desviaciones estándar:

```
# Cálculo de la media y desviación estándar para Sodium
mean_sodium <- mean(M$Sodium)
sd_sodium <- sd(M$Sodium)

# Identificación de datos atípicos con 3 desviaciones estándar
lim_inf_sd_sodium <- mean_sodium - 3 * sd_sodium
lim_sup_sd_sodium <- mean_sodium + 3 * sd_sodium
```

```r
outliers_sd_sodium <- M$Sodium[M$Sodium < lim_inf_sd_sodium | M$Sodium > lim_sup_sd_sodium]
outliers_sd_sodium
```

```
## [1] 2260 2290 3600
```

para la variable calorías los valores atipicos parecen ser casos donde la comida tiene un alto nivel calorico, por lo que si los removemos podríamos afectar a el analisis de la oferta de comida en mac donalds.

4. Para analizar normalidad se te sugiere:

para Calorías:

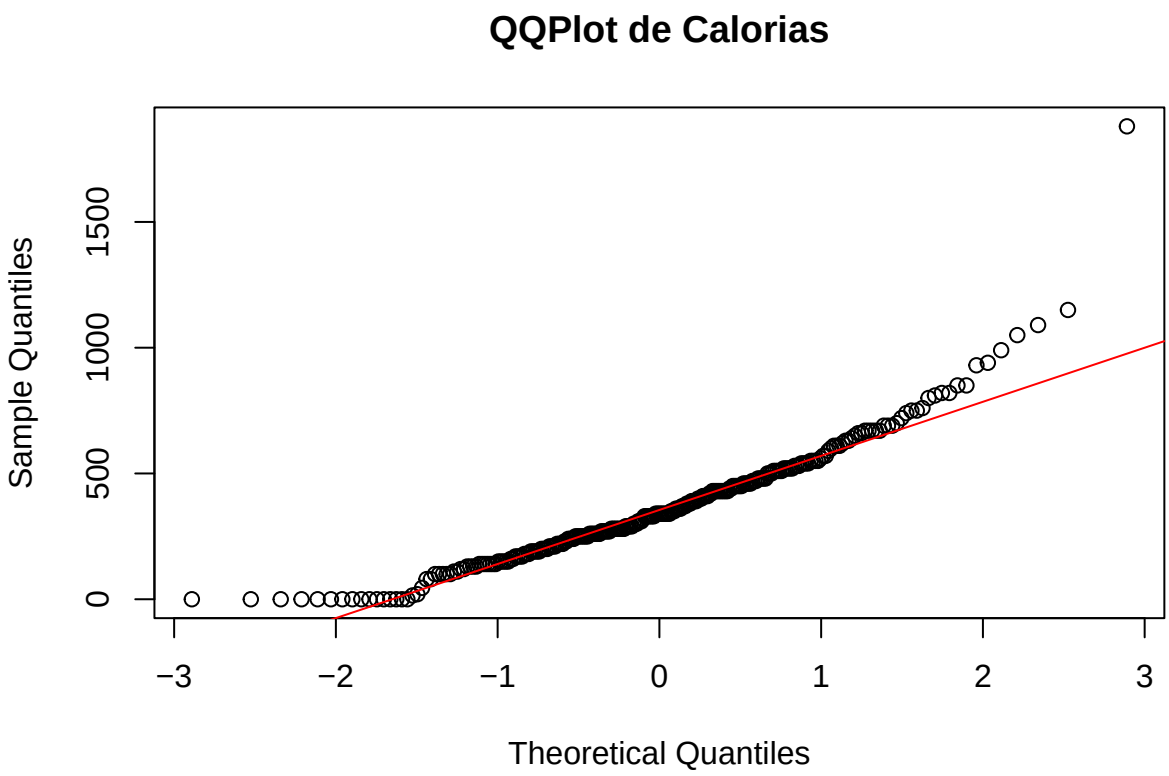
Realiza pruebas de normalidad univariada de las variables (selecciona entre los métodos vistos en clase)

```r
library(nortest)

# Prueba de Anderson-Darling para la variable Calorias
resultado_ad <- ad.test(M$Calories)

# Mostrar los resultados
print(resultado_ad)
```

```
##
##  Anderson-Darling normality test
##
## data:  M$Calories
## A = 2.5088, p-value = 2.369e-06
```

Grafica los datos y su respectivo QQPlot: qqnorm(datos) y qqline(datos) para cada variable

```r
qqnorm(M$Calories, main="QQPlot de Calorias")
qqline(M$Calories, col="red")
```



Calcula el coeficiente de sesgo y el coeficiente de curtosis de cada variable.

```
library(moments)

# Calcular sesgo y curtosis
sesgo_cal <- skewness(M$Calories)
curtosis_cal <- kurtosis(M$Calories)

# Mostrar resultados
print(paste("Sesgo:", sesgo_cal))
```

```
## [1] "Sesgo: 1.44410491051015"
```

```
print(paste("Curtosis:", curtosis_cal))
```

```
## [1] "Curtosis: 8.64527387047867"
```

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

```
# Calcular medidas
media_cal <- mean(M$Calories)
mediana_cal <- median(M$Calories)
rango_medio_cal <- (min(M$Caloriae) + max(M$Calories)) / 2
```

```
## Warning in min(M$Caloriae): no non-missing arguments to min; returning Inf
```

```r
# Mostrar resultados
print(paste("Media:", media_cal))
```

```
## [1] "Media: 368.269230769231"
```

```r
print(paste("Mediana:", mediana_cal))
```

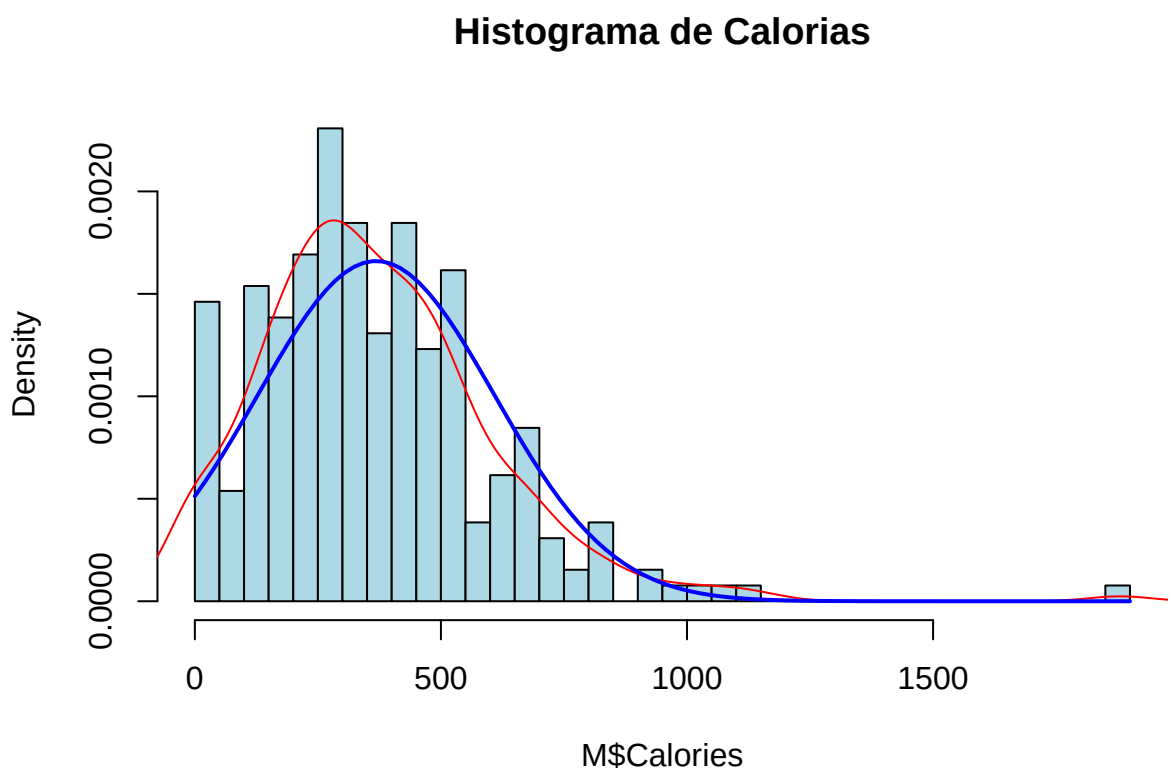
```
## [1] "Mediana: 340"
```

```r
print(paste("Rango Medio:", rango_medio_cal))
```

```
## [1] "Rango Medio: Inf"
```

Realiza el histograma y su distribución teórica de probabilidad:

```r
# Histograma de Calorias
hist(M$Calories, freq=FALSE, main="Histograma de Calorias", col="lightblue", breaks=30)

# Curva de densidad real
lines(density(M$Calories), col="red")

# Curva de distribución normal teórica
curve(dnorm(x, mean=mean(M$Calories), sd=sd(M$Calories)), col="blue", lwd=2, add=TRUE)
```



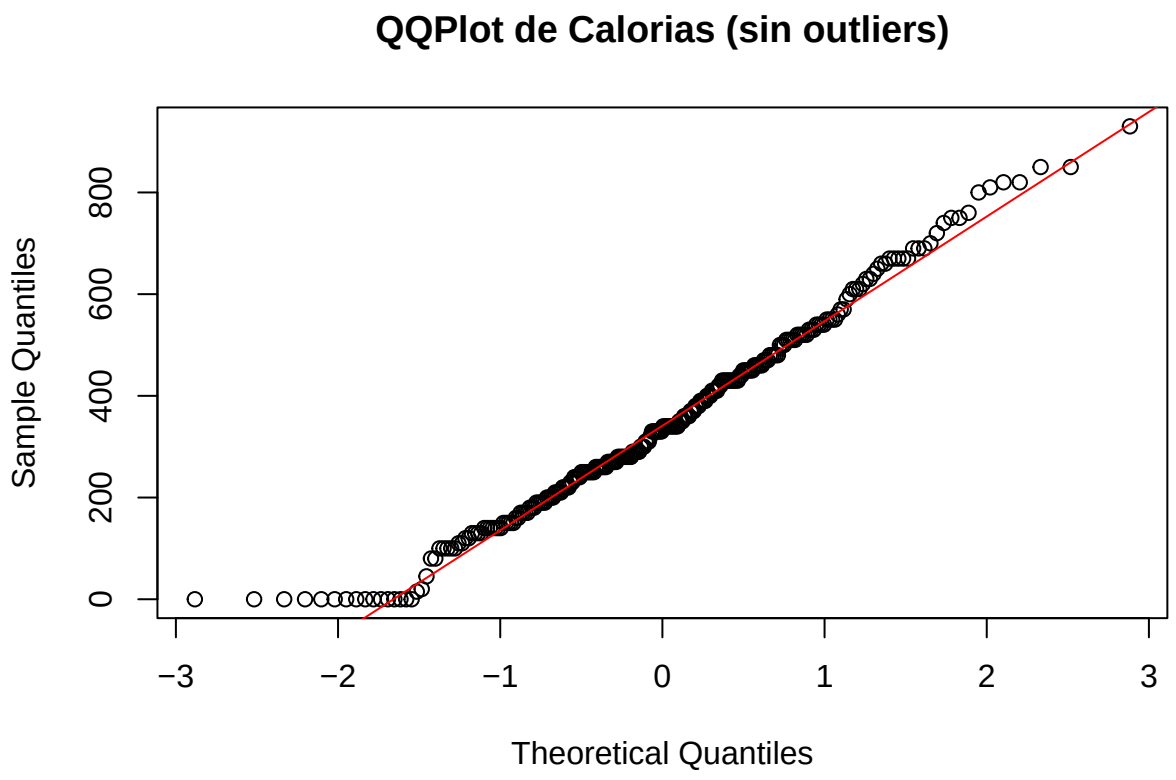
Identifica cómo influyen los datos atípicos en la normalidad de los datos:

para esto vamos a comparar la distribución de los datos con y sin los datos atipicos.

```
# Calcular IQR y eliminar outliers
q1 <- quantile(M$Calories, 0.25)
q3 <- quantile(M$Calories, 0.75)
iqr <- q3 - q1

lim_inf <- q1 - 1.5 * iqr
lim_sup <- q3 + 1.5 * iqr

cal_sin_outliers <- M$Calories[M$Calories >= lim_inf & M$Calories <= lim_sup]

# Repetir prueba de Anderson-Darling sin outliers
resultado_ad_sin_outliers <- ad.test(cal_sin_outliers)
print(resultado_ad_sin_outliers)
```

```
##
##  Anderson-Darling normality test
##
## data:  cal_sin_outliers
## A = 0.89786, p-value = 0.02166
```

```
# Repetir QQPlot sin outliers
qqnorm(cal_sin_outliers, main="QQPlot de Calorias (sin outliers)")
qqline(cal_sin_outliers, col="red")
```



```r
# Calcular sesgo y curtosis sin outliers
sesgo_cal_sin_outliers <- skewness(cal_sin_outliers)
curtosis_cal_sin_outliers <- kurtosis(cal_sin_outliers)

print(paste("Sesgo sin outliers:", sesgo_cal_sin_outliers))
```

```
## [1] "Sesgo sin outliers: 0.34905486802847"
```

```r
print(paste("Curtosis sin outliers:", curtosis_cal_sin_outliers))
```

```
## [1] "Curtosis sin outliers: 2.71682761548256"
```

Comenta los gráficos y los resultados obtenidos con vías a interpretar normalidad de los datos:

La eliminación de los valores atipicos ha resultado en una distribución que se acerca más a la normalidad, tanto en términos de sesgo como de curtosis. Esto indica que los valores atipicos estaban afectando la distribución original, y sin ellos, los datos de calorías son más simétricos y con una forma más cercana a una distribución normal.

Paria Sodio:

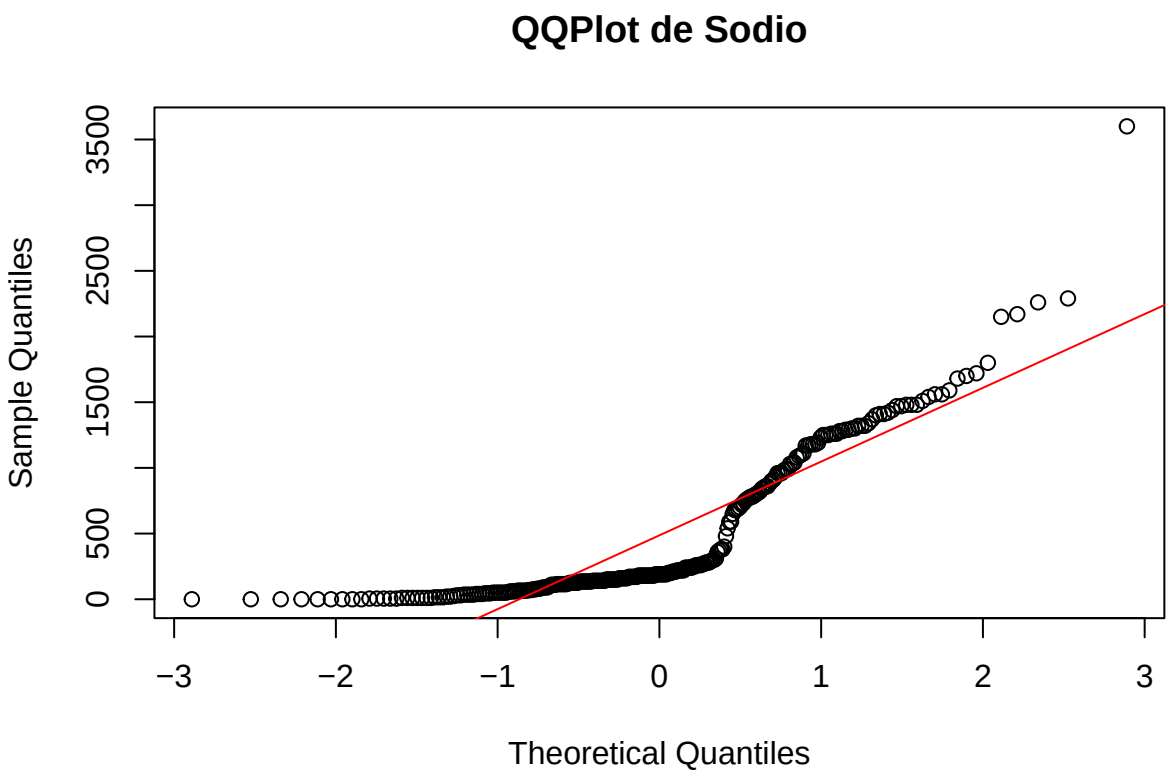
Realiza pruebas de normalidad univariada de las variables (selecciona entre los métodos vistos en clase)

```r
library(nortest)

# Prueba de Anderson-Darling para la variable Sodium
resultado_ad_sodium <- ad.test(M$Sodium)

# Mostrar los resultados
print(resultado_ad_sodium)
```

```
##
##  Anderson-Darling normality test
##
## data:  M$Sodium
## A = 21.406, p-value < 2.2e-16
```

Grafica los datos y su respectivo QQPlot: qqnorm(datos) y qqline(datos) para cada variable

```r
qqnorm(M$Sodium, main="QQPlot de Sodio")
qqline(M$Sodium, col="red")
```



Calcula el coeficiente de sesgo y el coeficiente de curtosis de cada variable.

```
library(moments)

# Calcular sesgo y curtosis para Sodium
sesgo_sodium <- skewness(M$Sodium)
curtosis_sodium <- kurtosis(M$Sodium)

# Mostrar resultados
print(paste("Sesgo:", sesgo_sodium))
```

```
## [1] "Sesgo: 1.53516568057938"
```

```
print(paste("Curtosis:", curtosis_sodium))
```

```
## [1] "Curtosis: 5.79641225557164"
```

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

```
# Calcular medidas
media_sodium <- mean(M$Sodium)
mediana_sodium <- median(M$Sodium)
rango_medio_sodium <- (min(M$Sodium) + max(M$Sodium)) / 2

# Mostrar resultados
print(paste("Media:", media_sodium))
```

```
## [1] "Media: 495.75"
```

```
print(paste("Mediana:", mediana_sodium))
```

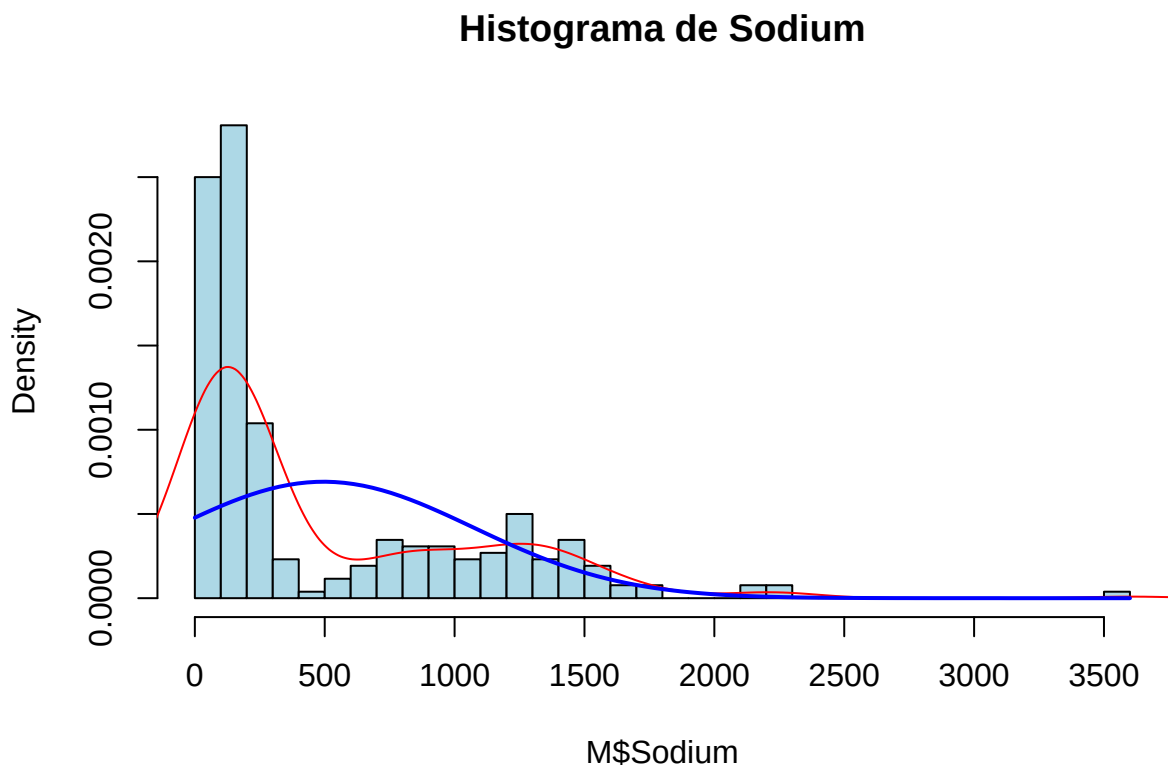
```
## [1] "Mediana: 190"
```

```
print(paste("Rango Medio:", rango_medio_sodium))
```

```
## [1] "Rango Medio: 1800"
```

Realiza el histograma y su distribución teórica de probabilidad:

```
# Histograma de Sodium
hist(M$Sodium, freq=FALSE, main="Histograma de Sodium", col="lightblue", breaks=30)

# Curva de densidad real
lines(density(M$Sodium), col="red")

# Curva de distribución normal teórica
curve(dnorm(x, mean=mean(M$Sodium), sd=sd(M$Sodium)), col="blue", lwd=2, add=TRUE)
```

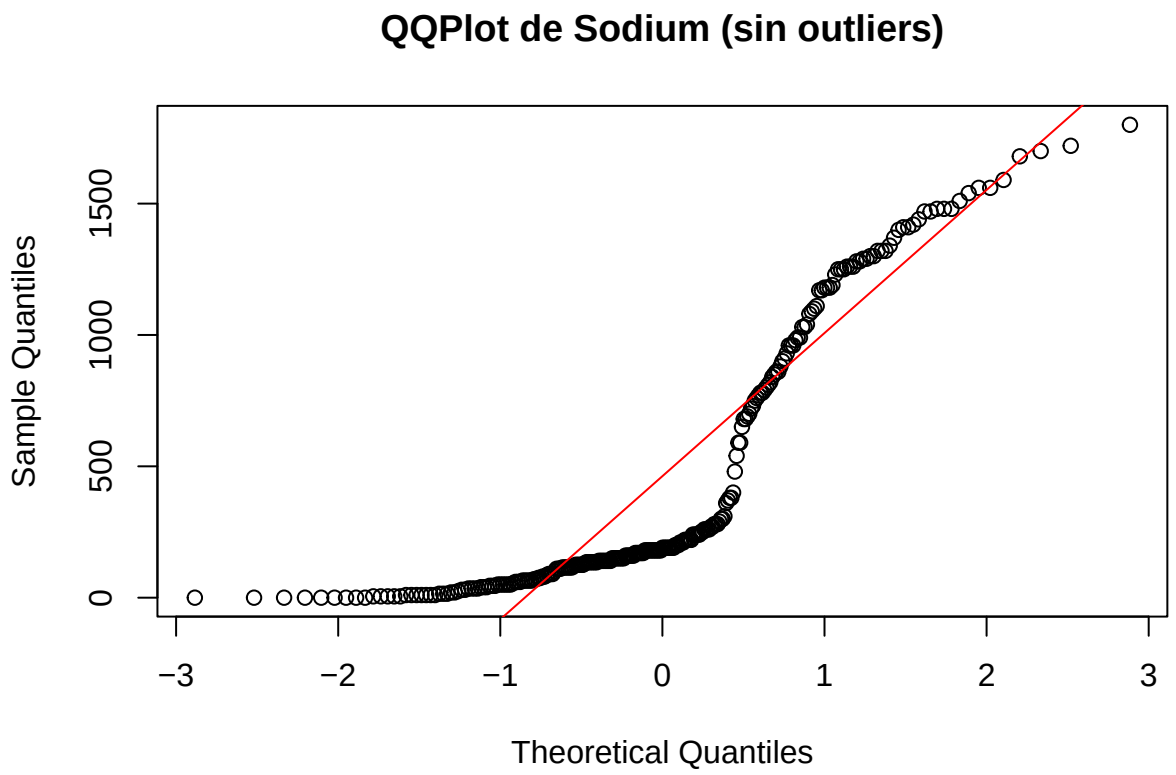


Identifica cómo influyen los datos atípicos en la normalidad de los datos:

```r
# Calcular IQR y eliminar outliers
q1_sodium <- quantile(M$Sodium, 0.25)
q3_sodium <- quantile(M$Sodium, 0.75)
iqr_sodium <- q3_sodium - q1_sodium

lim_inf_sodium <- q1_sodium - 1.5 * iqr_sodium
lim_sup_sodium <- q3_sodium + 1.5 * iqr_sodium

sodium_sin_outliers <- M$Sodium[M$Sodium >= lim_inf_sodium & M$Sodium <= lim_sup_sodium]

# Repetir prueba de Anderson-Darling sin outliers
resultado_ad_sodium_sin_outliers <- ad.test(sodium_sin_outliers)
print(resultado_ad_sodium_sin_outliers)
```

```
##
##  Anderson-Darling normality test
##
## data:  sodium_sin_outliers
## A = 22.737, p-value < 2.2e-16
```

```r
# Repetir QQPlot sin outliers
qqnorm(sodium_sin_outliers, main="QQPlot de Sodium (sin outliers)")
qqline(sodium_sin_outliers, col="red")
```



```r
# Calcular sesgo y curtosis sin outliers
sesgo_sodium_sin_outliers <- skewness(sodium_sin_outliers)
curtosis_sodium_sin_outliers <- kurtosis(sodium_sin_outliers)

print(paste("Sesgo sin outliers:", sesgo_sodium_sin_outliers))
```

```
## [1] "Sesgo sin outliers: 1.03416162141111"
```

```r
print(paste("Curtosis sin outliers:", curtosis_sodium_sin_outliers))
```

```
## [1] "Curtosis sin outliers: 2.59852836981709"
```

Comenta los gráficos y los resultados obtenidos con vías a interpretar normalidad de los datos:

aunque eliminar los valores atipicos ha reducido la curtosis y ha afectado ligeramente el sesgo, la distribución de Sodium sigue sin ser perfectamente normal. Por lo tanto, si la normalidad es un requisito, podrías considerar transformaciones adicionales o utilizar modelos que no dependan de la normalidad.