

## 2. Explorando bases

### Datos del alumno

Luis Ángel Guzmán Iribe - A01741757

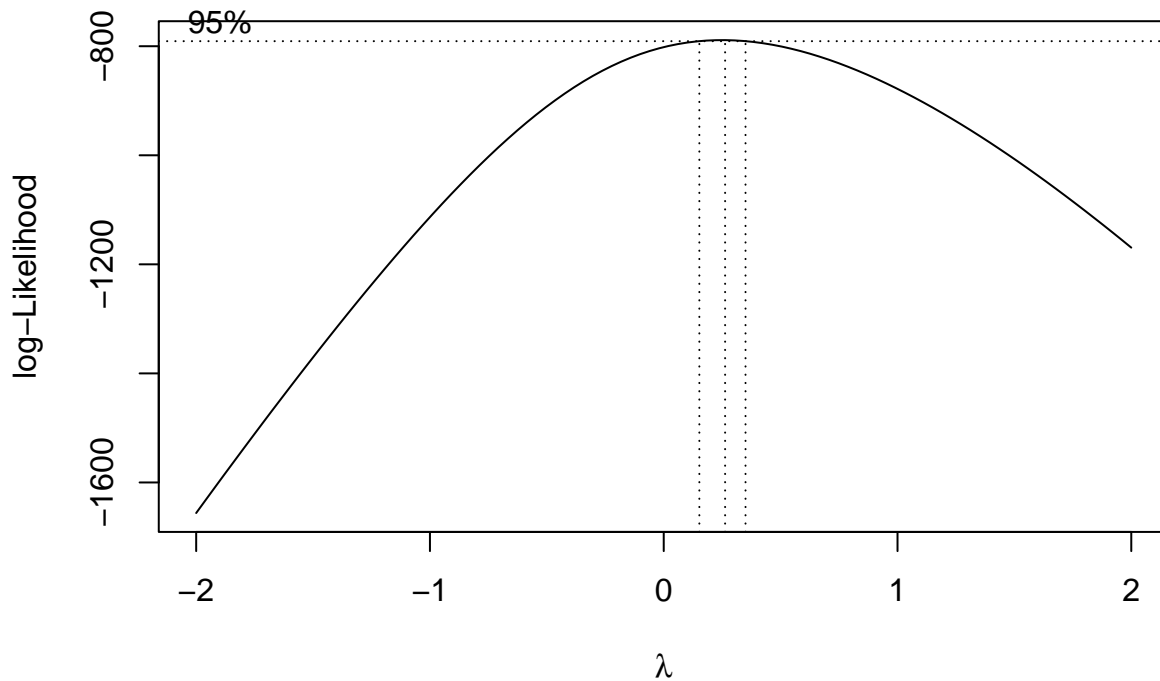
### Análisis de Azúcares

```
library(readr)
library(moments)
data <- read.csv("mc-donalds-menu-1.csv")

sugars <- 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

```
library(MASS)
bc <- boxcox((sugars[,1]+1)~1)
```



```
l = bc$x[which.max(bc$y)]
cat("Lambda: ", l)

## Lambda: 0.2626263
```

2. Escribe las ecuaciones de los modelos encontrados.

$$cal_1 = x + 1$$

$$cal_2 = \frac{(x + 1)^{0.26} - 1}{0.26}$$

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

```
library(e1071)
```

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

```
##
## Attaching package: 'e1071'

## The following objects are masked from 'package:moments':
##
##      kurtosis, moment, skewness

library(nortest)

cal1=sqrt(sugars[,1]+1)
cal2=((sugars[,1]+1)^1-1)/1

D0=ad.test(sugars[,1])
D1=ad.test(cal1)
D2=ad.test(cal2)

m0 = round(c(as.numeric(summary(sugars[,1])),kurtosis(sugars[,1]),skewness(sugars[,1]),D0$p.value),3)
m1 = round(c(as.numeric(summary(cal1)),kurtosis(cal1),skewness(cal1),D1$p.value),3)
m2 = round(c(as.numeric(summary(cal2)),kurtosis(cal2),skewness(cal2),D2$p.value),3)

m <- as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Primer modelo","Segundo Modelo")
names(m)=c("Mínimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo","Valor p")

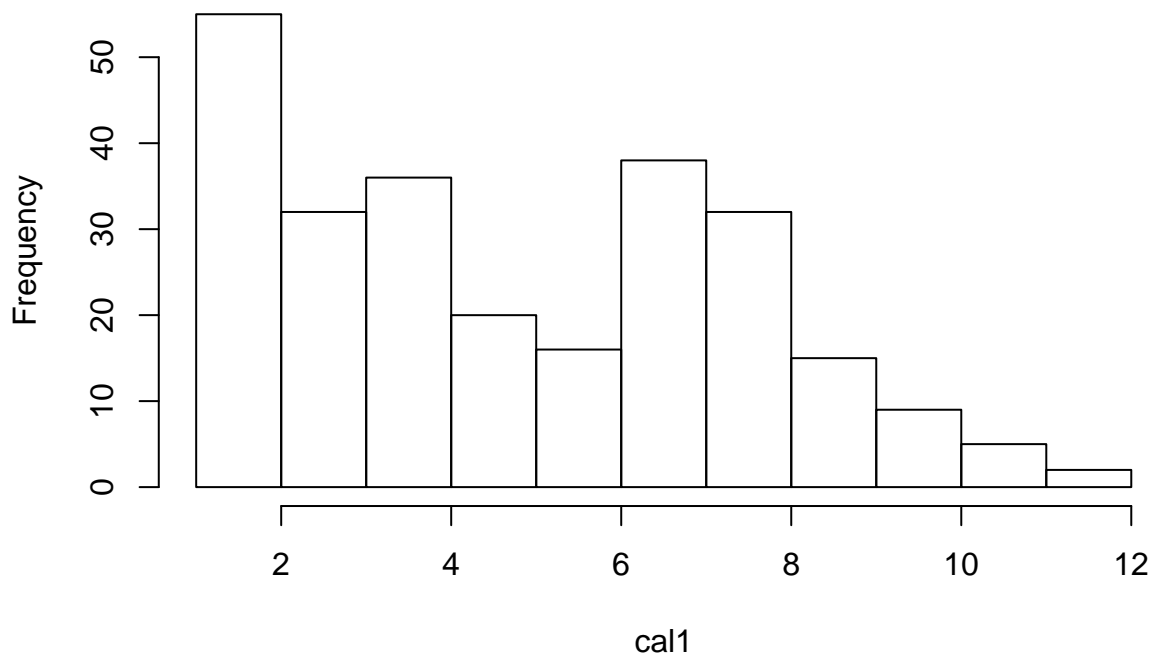
m

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

hist(cal1,col=0,main="Histograma de Azucar Cal_1")
```

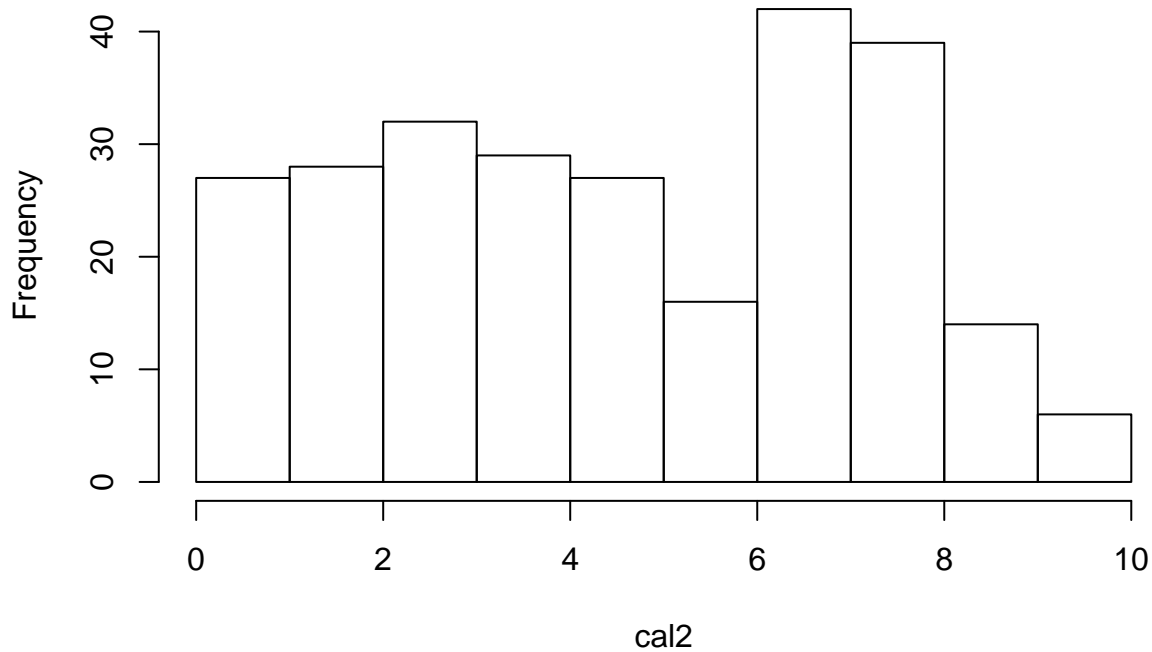
3.2 Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.

### Histograma de Azucar Cal\_1



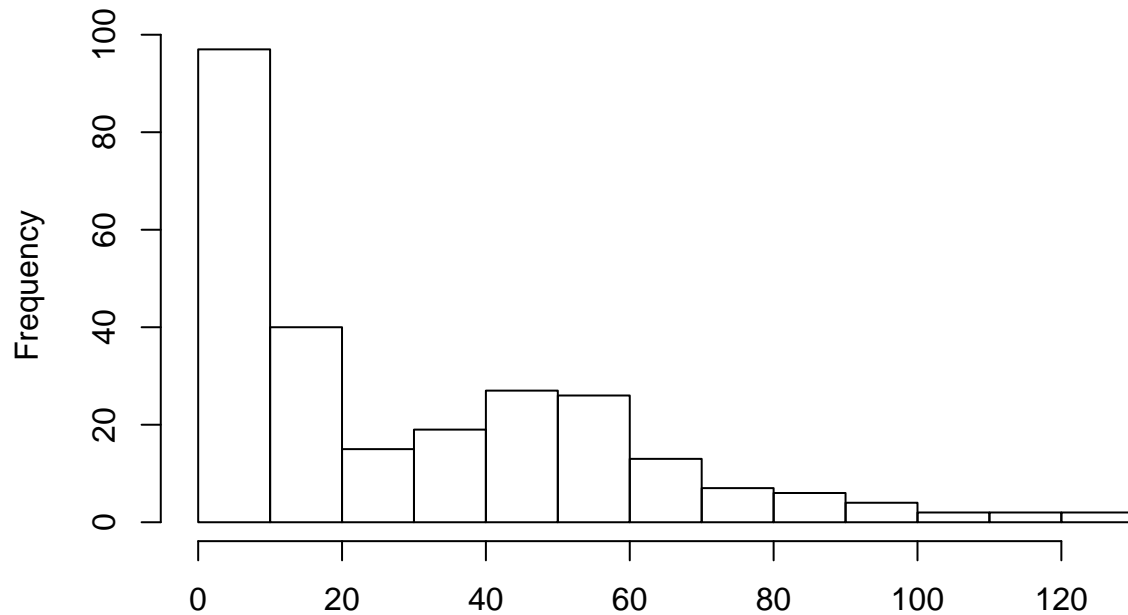
```
hist(cal2,col=0,main="Histograma de Azucar Cal_2")
```

### Histograma de Azucar Cal\_2



```
hist(sugars[ ,1],col=0,main="Histograma de Azucar Original")
```

## Histograma de Azucar Original



sugars[, 1]

####

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

```
library(nortest)
D0=ad.test(sugars[,1])
D1=ad.test(cal1)
D2=ad.test(cal2)
```

D0

```
##
## Anderson-Darling normality test
##
## data:  sugars[, 1]
## A = 9.9899, p-value < 2.2e-16
```

D1

```
##
## Anderson-Darling normality test
##
## data:  cal1
## A = 4.0816, p-value = 3.531e-10
```

D2

```
##
## Anderson-Darling normality test
##
## data:  cal2
## A = 3.3722, p-value = 1.857e-08
```

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).

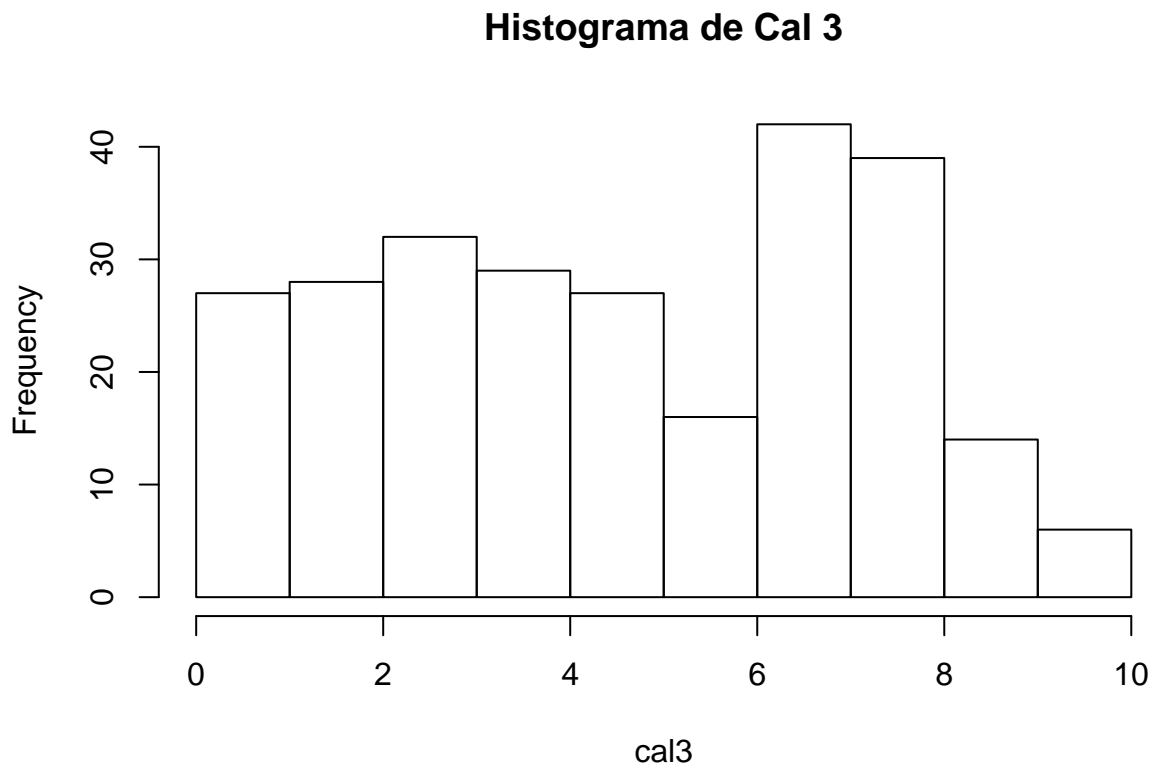
```
library(VGAM)
```

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

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

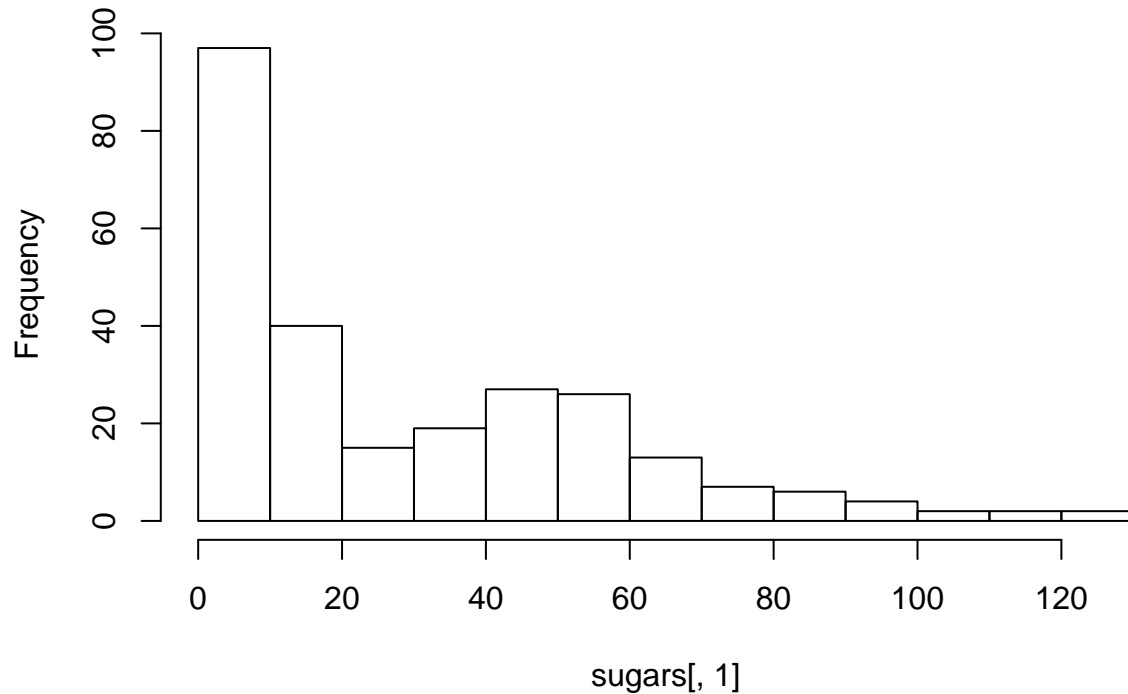
```
cal3 <- yeo.johnson(sugars[,1],lambda = 1)
```

```
hist(cal3, col=0, main="Histograma de Cal 3")
```



```
hist(sugars[,1], col=0, main="Histograma de Proteínas")
```

## Histograma de Proteínas



```
library(VGAM)
library(ggplot2)
lp <- seq(0, 1, 0.001) # Valores lambda propuestos
nlp <- length(lp)
n <- length(sugars)
D <- matrix(as.numeric(NA), ncol = 2, nrow = nlp)
d <- NA

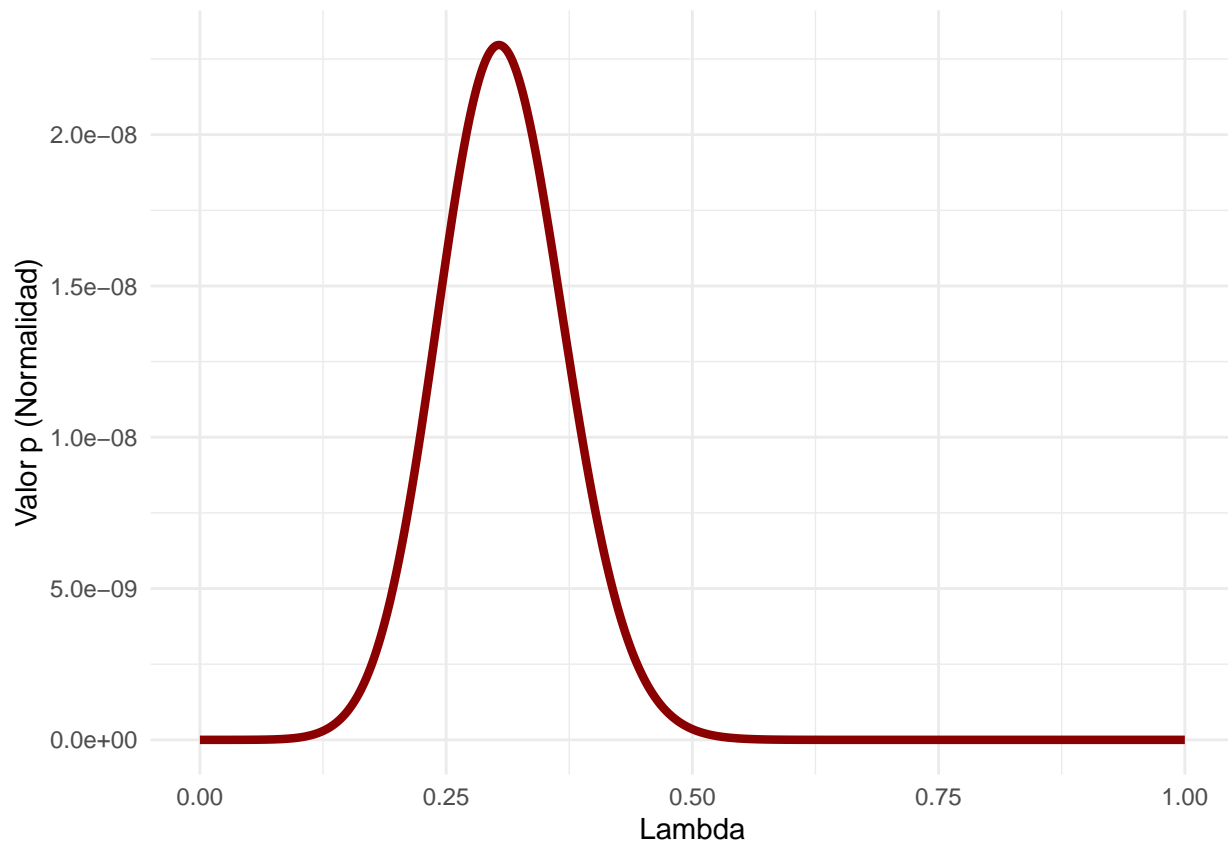
for (i in 1:nlp) {
  d = yeo.johnson(sugars[,1], lambda = lp[i])
  p = ad.test(d)
  D[i,] = c(lp[i], p$p.value)
}

N <- as.data.frame(D)

# Renombramos las columnas
colnames(N) <- c("Lambda", "P-Value")

# Creamos un gráfico utilizando ggplot2
ggplot(N, aes(x = `Lambda`, y = `P-Value`)) + geom_line(color = "darkred", size = 1.5) + labs(x = "Lambda", y = "P-Value")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
G <- data.frame(subset(N, `P-Value` == max(N$`P-Value`)))
G$Lambda
```

```
## [1] 0.304
```

6. Escribe la ecuación del modelo encontrado.

$$cal_3 = \frac{(x+1)^{0.305} - 1}{0.305}$$

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

```
library(e1071)
D0 = ad.test(sugars[,1])
D1 = ad.test(cal1)
D2 = ad.test(cal2)
D3 = ad.test(cal3)

m0 = round(c(as.numeric(summary(sugars)),kurtosis(sugars),skewness(sugars),D0$p.value),3)
```

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

```
## Warning: NAs introduced by coercion
## Warning in mean.default(x): argument is not numeric or logical: returning NA
## Warning in mean.default(x): argument is not numeric or logical: returning NA
```

```

m1 = round(c(as.numeric(summary(cal1)),kurtosis(cal1),skewness(cal1),D1$p.value),3)
m2 = round(c(as.numeric(summary(cal2)),kurtosis(cal2),skewness(cal2),D2$p.value),3)
m3 = round(c(as.numeric(summary(cal3)),kurtosis(cal3),skewness(cal3),D3$p.value),3)

m <- as.data.frame(rbind(m0,m1,m2,m3))
row.names(m)=c("Original","Primer modelo","Segundo Modelo", "Tercer Modelo")
names(m)=c("Mínimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo","Valor p")

m

```

##	Minimo	Q1	Mediana	Media	Q3	Máximo	Curtosis	Sesgo	Valor p
## Original	NA	NA	NA	NA	NA	NA	NA	NA	0
## Primer modelo	1	2.597	4.301	4.825	7.000	11.358	-1.014	0.279	0
## Segundo Modelo	0	2.477	4.385	4.519	6.774	9.837	-1.113	-0.106	0
## Tercer Modelo	0	2.477	4.385	4.519	6.774	9.837	-1.113	-0.106	0

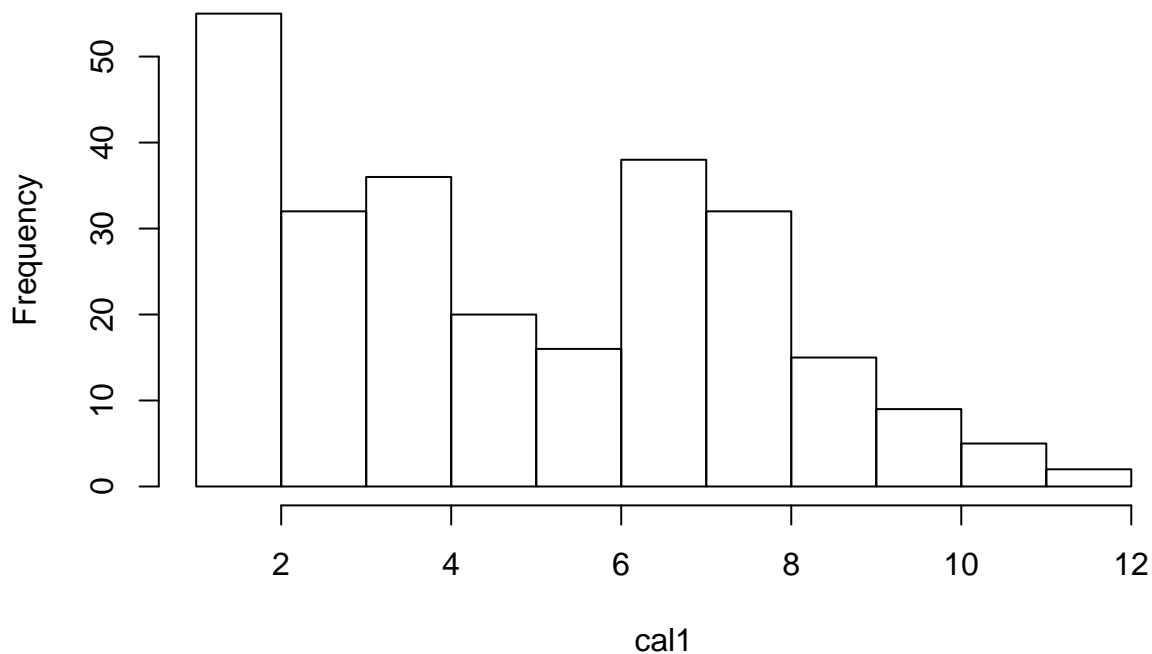
```

cal1=sqrt(sugars[,1] + 1)
hist(cal1,col=0,main="Histograma de Azucar Cal_1")

```

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

### Histograma de Azucar Cal\_1



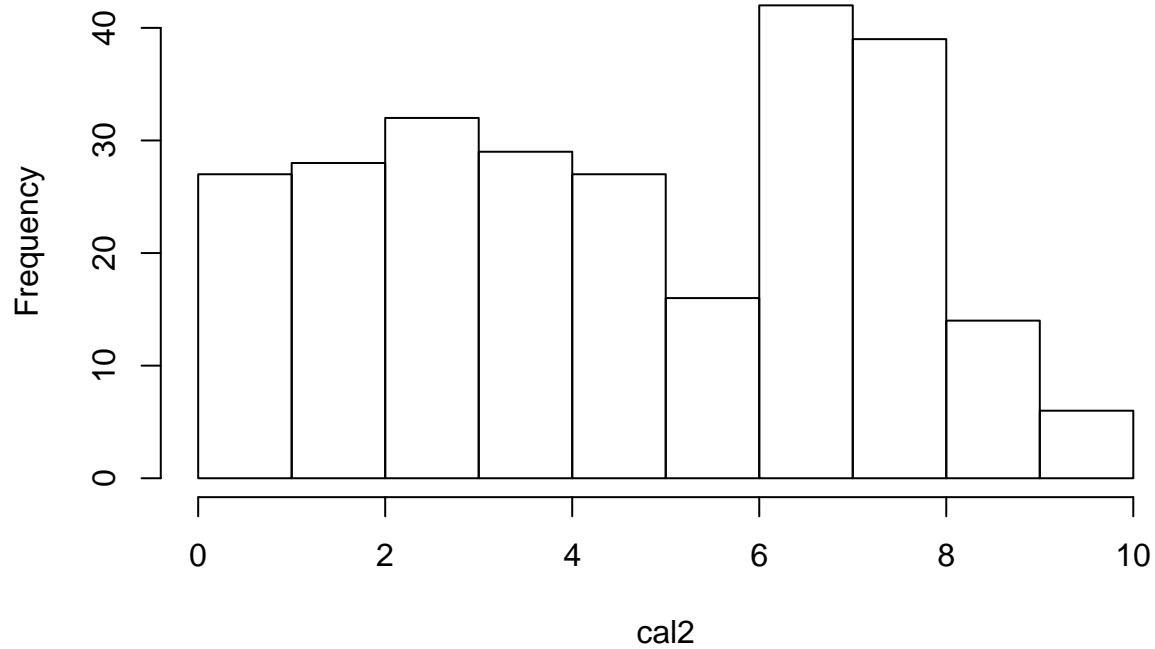
```

cal2=((sugars[,1]+1)^1-1)/1
hist(cal2,col=0,main="Histograma de Azucar Cal_2")

```

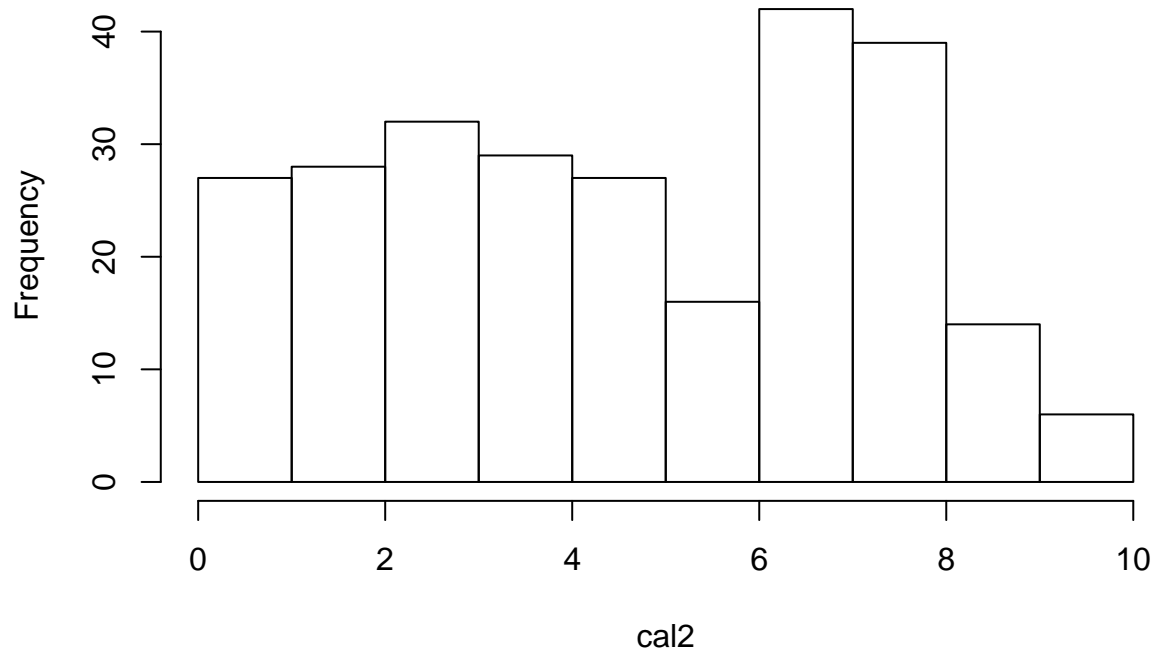


## Histograma de Azucar Cal\_2



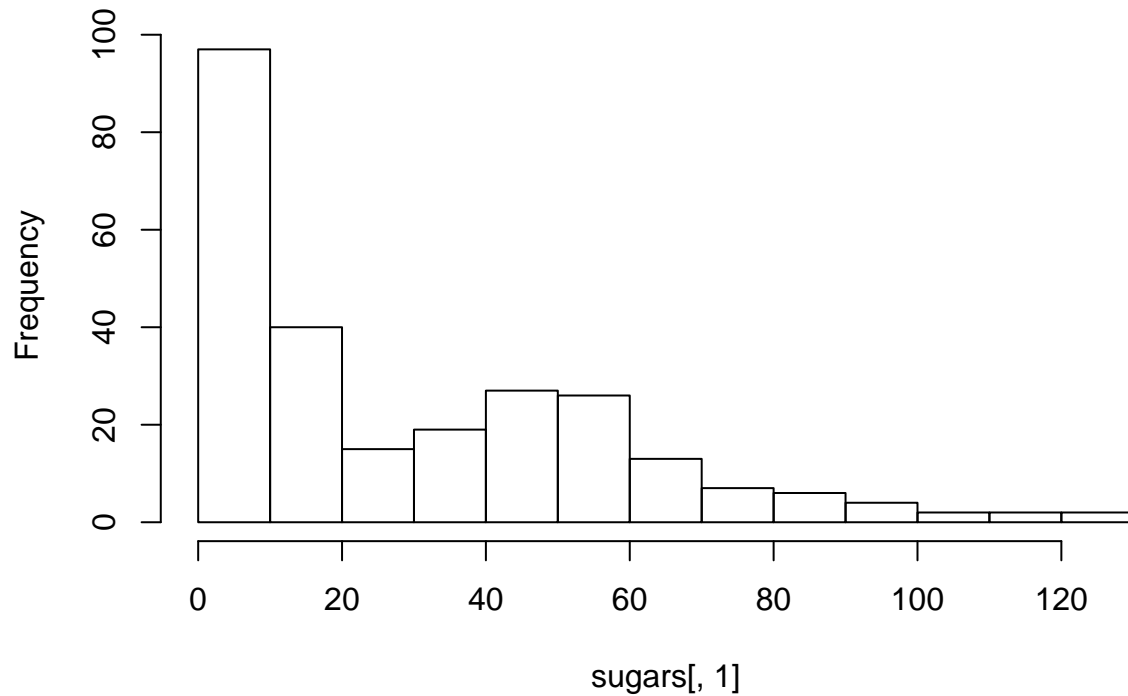
```
cal3=((sugars[,1]+1)^(G$Lambda)-1)/(G$Lambda)
hist(cal2,col=0,main="Histograma de Azucar Cal_3")
```

## Histograma de Azucar Cal\_3



```
hist(sugars[,1],col=0,main="Histograma de Azucar Original")
```

## Histograma de Azucar Original



```
library(nortest)
D0=ad.test(sugars[,1])
D1=ad.test(cal1)
D2=ad.test(cal2)
D3=ad.test(cal3)
```

D0

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

```
##
## Anderson-Darling normality test
##
## data: sugars[, 1]
## A = 9.9899, p-value < 2.2e-16
```

D1

```
##
## Anderson-Darling normality test
##
## data: cal1
## A = 4.0816, p-value = 3.531e-10
```

D2

```
##
## Anderson-Darling normality test
##
## data: cal2
```

```
## A = 3.3722, p-value = 1.857e-08
```

```
D3
```

```
##
```

```
## Anderson-Darling normality test
```

```
##
```

```
## data: cal3
```

```
## A = 3.3343, p-value = 2.296e-08
```