Instituto Tecnológico y de Estudios Superiores de Monterrey

Actividad M1. Transformaciones

TC3006C.101 Inteligencia artificial avanzada para la ciencia de datos I

Profesores:

Ivan Mauricio Amaya Contreras

Blanca Rosa Ruiz Hernandez

Antonio Carlos Bento

Frumencio Olivas Alvarez

Hugo Terashima Marín

Alumno:

Alberto H Orozco Ramos - A00831719

21 de Agosto de 2023

Instrucciones

1. Baja el archivo de trabajo: datos de McDonald

In [ ]:
```
# Montar Google Drive
from google.colab import drive
drive.mount('/content/drive')
file_path = "/content/drive/MyDrive/Colab Notebooks/mc-donalds-menu-1.csv"
```

```
Mounted at /content/drive
```

In [ ]:
```
%load_ext rpy2.ipython
```

In [ ]:
```
# Pasamos la variable de la ruta de los datos a R
%Rpush file_path
```

In [ ]:
```
%%R
# Cargamos las librerías necesarias
install.packages("moments")
install.packages("e1071")
install.packages("nortest")
install.packages("knitr")
install.packages("VGAM")
install.packages("ggplot2")
library(nortest)
library(moments)
library(e1071)
library(MASS)
library(readr)
library(knitr)
library(VGAM)
library(ggplot2)
```

In [ ]:
```
# Cargamos los datos CSV desde Google Drive
%%R

M <- read.csv(file_path)
```

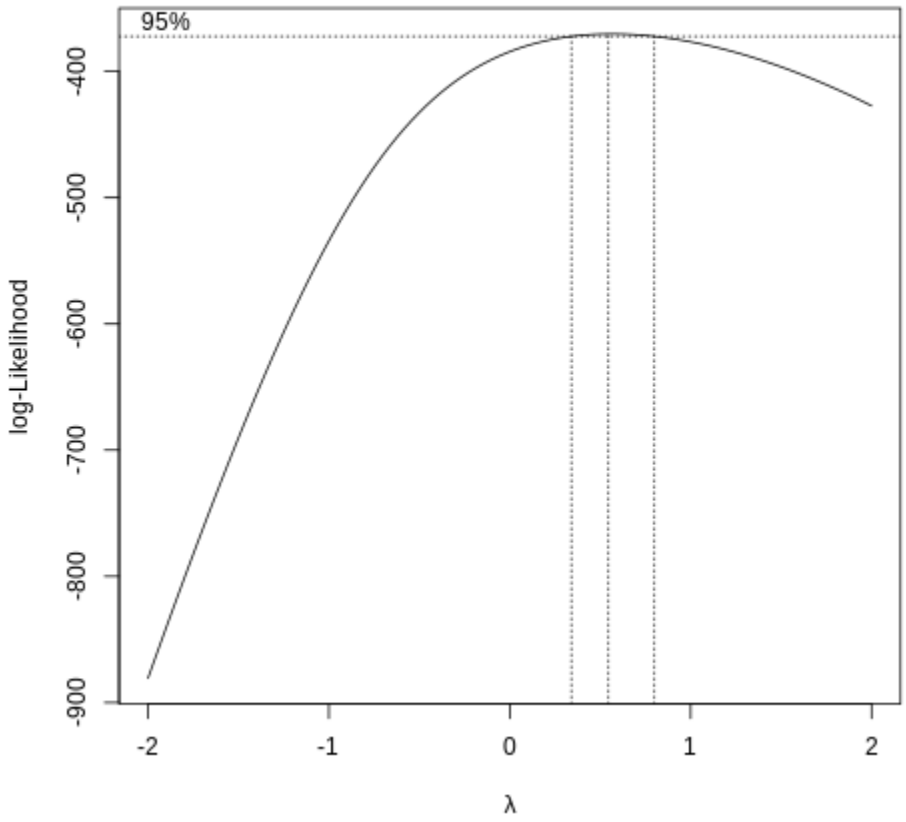
Proteínas

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

In [ ]:
```
%%R

# Extraemos la variable de proteínas de la data y
M1 <- M$Protein[M$Protein]

# Generamos la gráfica de Box-Cox
```

```
bc <- boxcox((M1 + 1)~1)

# Obtenemos el valor de lambda (Máximo valor de la función de verosimilitud generad
l <- bc$x[which.max(bc$y)]

cat("Lambda: ", l)
```

```
Lambda:  0.5454545
```



2. Escribe las ecuaciones de los modelos encontrados.

Transformación de Box-Cox

$$cal_1 = \sqrt{x+1}$$

$$cal_2 = \frac{(x+1)^{0.54}-1}{0.54}$$

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.
2. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.
3. Realiza la prueba de normalidad de Anderson-Darling o de Jarque Bera para los datos transformados y los originales

1. Comparar las medidas y 3. Prueba de Normalidad de Anderson-Darling

In [ ]:
```
%%R

# Transformación 1 (Fórmula)
cal1 <- sqrt(M1 + 1)

#Transformación 2 (Fórmula)
cal2 <- ((M1 + 1)^l - 1) / l

# Prueba de Normalidad
D0 = ad.test(M1)
D1 = ad.test(cal1)
D2 = ad.test(cal2)

print(D0)
print(D1)
print(D2)

# Resumen de Medidas
m0=round(c(as.numeric(summary(M1)),kurtosis(M1),skewness(M1),D0$p.value),3)
m1=round(c(as.numeric(summary(cal1)),kurtosis(cal1),skewness(cal1),D1$p.value),3)
```

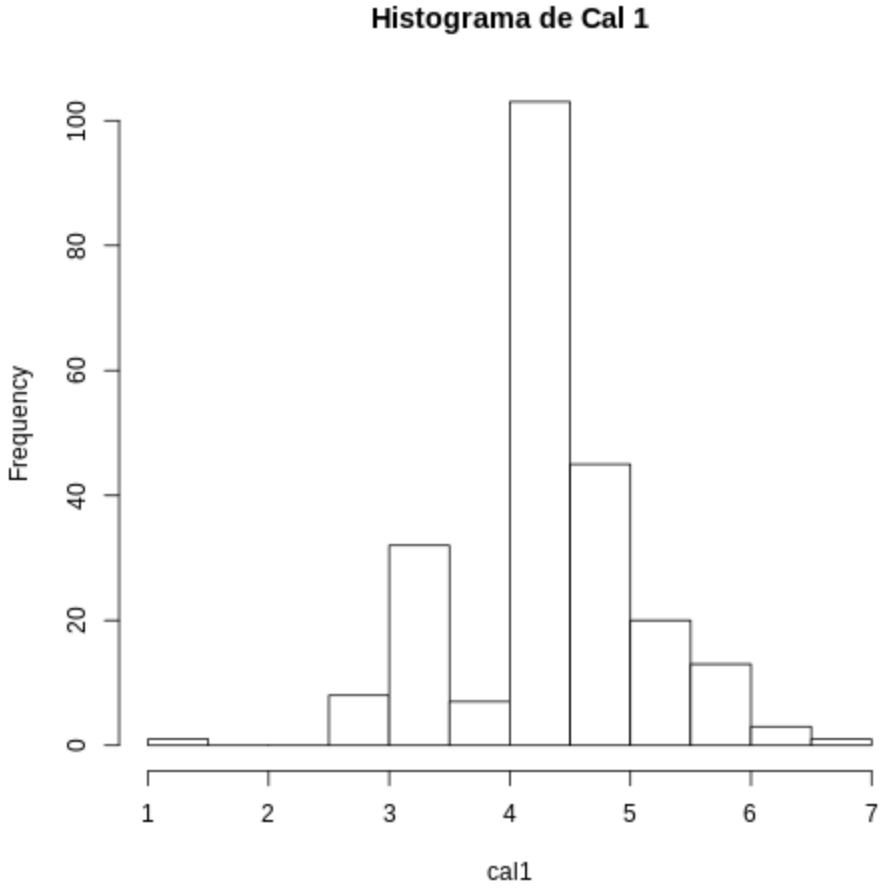
```
m2=round(c(as.numeric(summary(cal2)),kurtosis(cal2),skewness(cal2),D2$p.value),3)

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

# Mostramos la tabla
kable(m, format = "markdown", digits = 3)
```

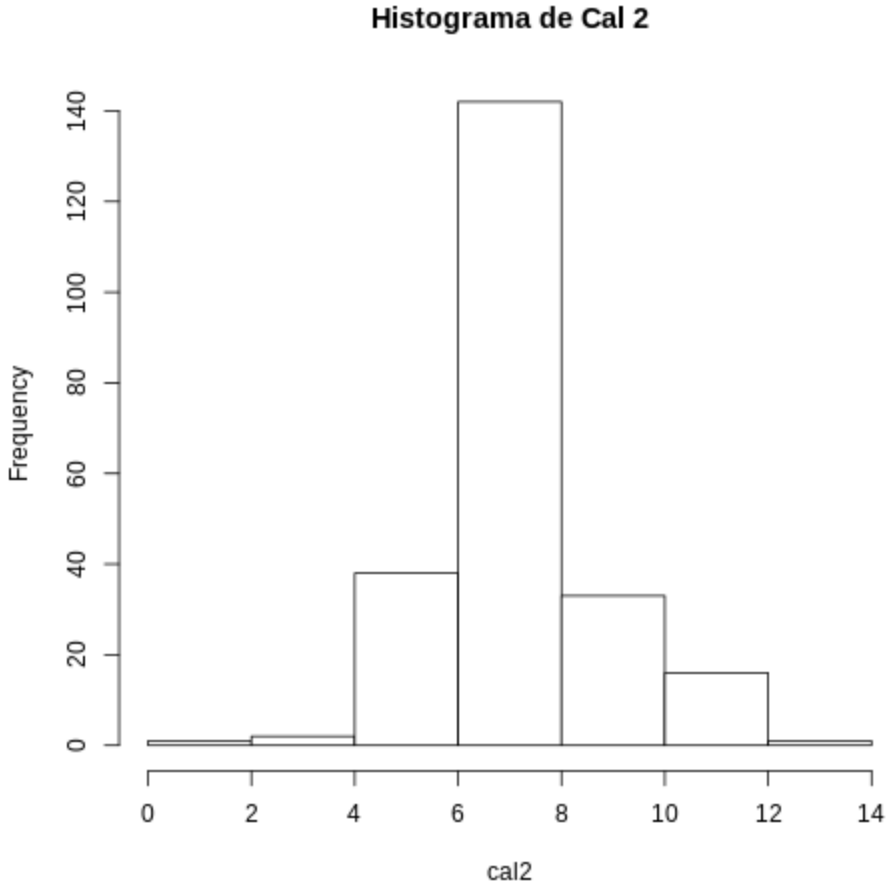
```
        Anderson-Darling normality test

data:  M1
A = 9.5674, p-value < 2.2e-16


        Anderson-Darling normality test

data:  cal1
A = 8.6687, p-value < 2.2e-16


        Anderson-Darling normality test

data:  cal2
A = 8.6654, p-value < 2.2e-16
```

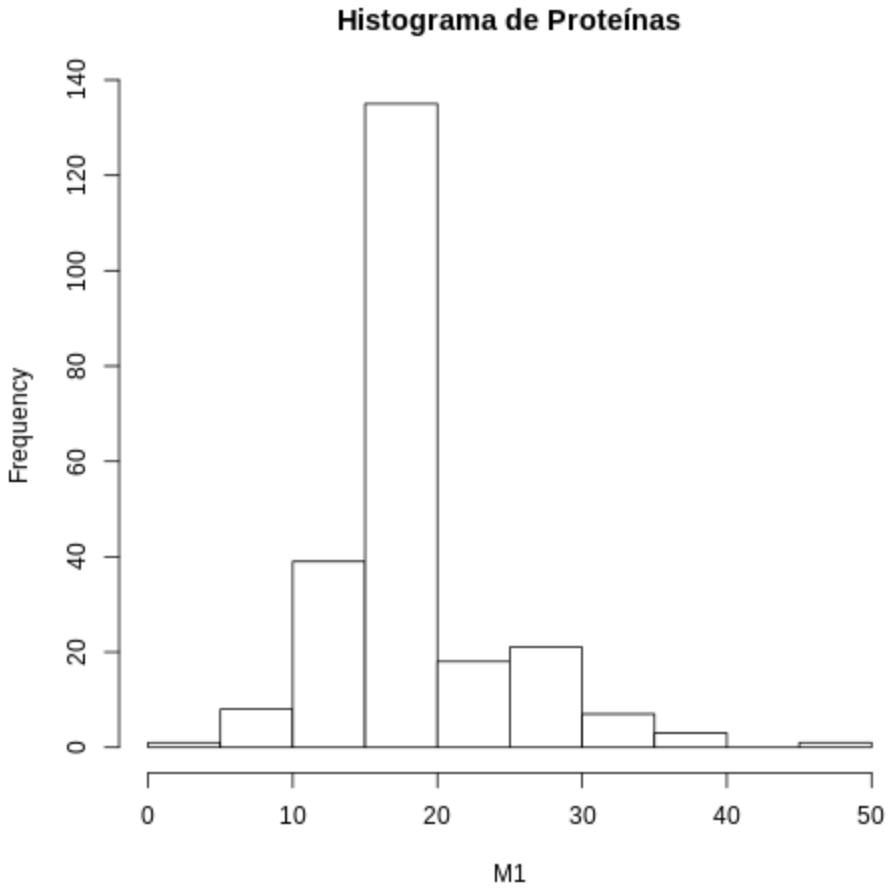
	Minimo	Q1	Mediana	Media	Q3	Máximo	Curtosis	Sesgo	Valor p
Original	1.000	17.000	18.000	18.734	20.000	48.000	2.473	0.894	0
Primer modelo	1.414	4.243	4.359	4.388	4.583	7.000	1.960	0.002	0
Segundo Modelo	0.842	7.037	7.302	7.380	7.815	13.483	1.870	0.094	0

2. Histogramas

In [ ]:
```
%%R

hist(cal1, col=0, main="Histograma de Cal 1")
hist(cal2, col=0, main="Histograma de Cal 2")
hist(M1, col=0, main="Histograma de Proteínas")
```


Histograma de Cal 1
Frequency
0
20
40
60
80
100
1
2
3
4
5
6
7
cal1


Histograma de Cal 2
Frequency
0
20
40
60
80
100
120
140
0
2
4
6
8
10
12
14
cal2

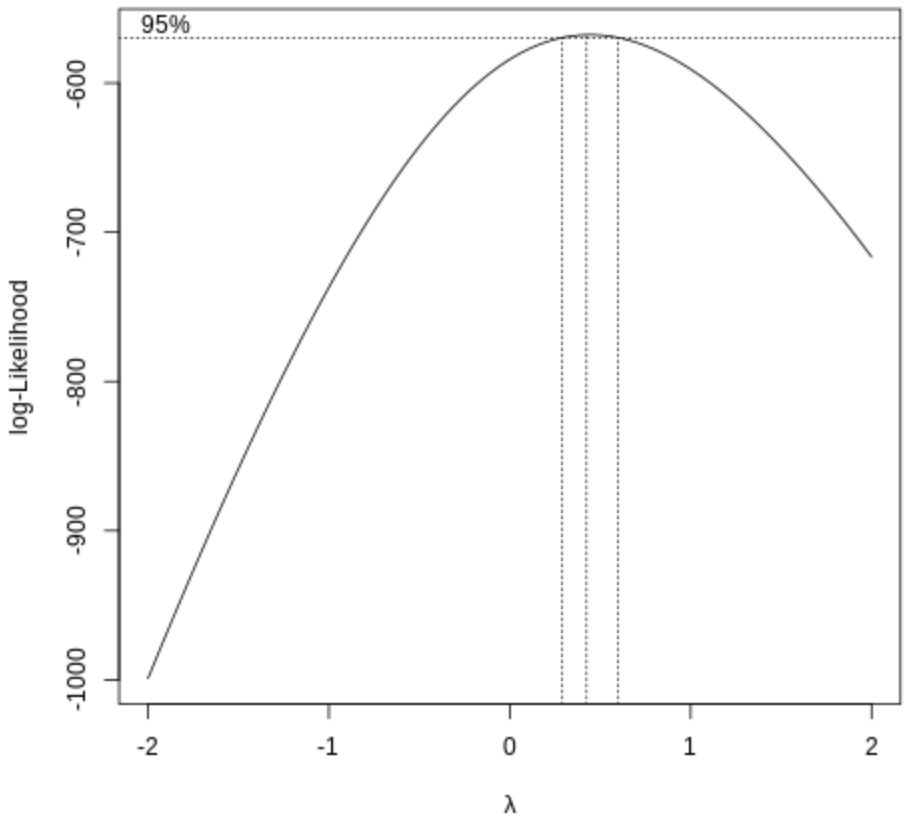


4. Detecta anomalías y corrige tu base de datos (datos atípicos, ceros anámalos, etc).

Considero que es importante excluir los datos que representan ceros dentro del conjunto de datos, debido a que estos no aportan valor para el análisis de la normalización, solo deberíamos tomar en cuenta datos que si influyan para el análisis de los mismos. Además, se puede excluir el valor atípico "87" debido a que afecta el proceso de normalización de los datos, esto porque se encuentra muy alejado del resto de conjunto de datos. Si repetimos el proceso hecho anteriormente pero sin considerar este dato atípico obtenemos una normalización algo más estable con los siguientes resultados:

In [ ]:
```
%%R

# Datos filtrados:
M1 <- M$Protein[M$Protein > 0 & M$Protein <= 80]

# Generamos la gráfica de Box-Cox
bc <- boxcox((M1 + 1)~1)

# Obtenemos el valor de lambda (Máximo valor de la función de verosimilitud generad
l <- bc$x[which.max(bc$y)]

cat("Lambda: ", l)
```

```
Lambda:  0.4242424
```



Transformación de Box-Cox

$$cal_1 = \sqrt{x+1}$$

$$cal_2 = \frac{(x+1)^{0.42} - 1}{0.42}$$

Comparar las medidas y Prueba de Normalidad de Anderson-Darling

In [ ]:
```
%%R
```

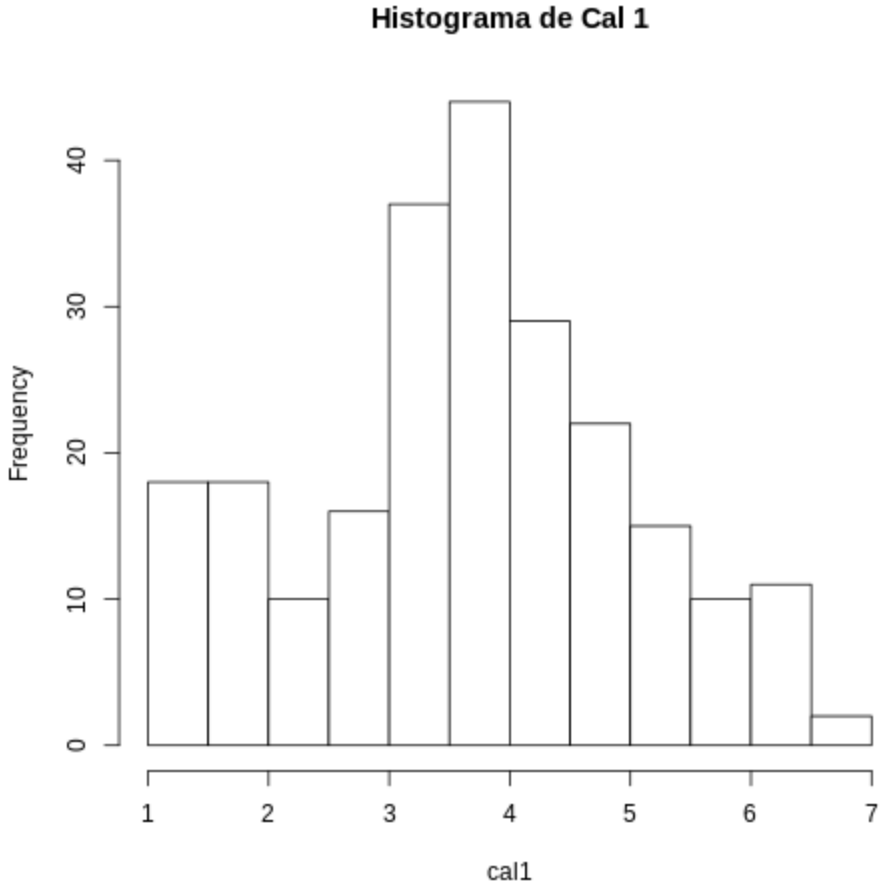
```
# Transformación 1 (Fórmula)
cal1 <- sqrt(M1 + 1)

#Transformación 2 (Fórmula)
cal2 <- ((M1 + 1)^l - 1) / l

# Prueba de Normalidad
D0 = ad.test(M1)
D1 = ad.test(cal1)
D2 = ad.test(cal2)

print(D0)
print(D1)
print(D2)

# Resumen de Medidas
m0=round(c(as.numeric(summary(M1)),kurtosis(M1),skewness(M1),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)

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

# Mostramos la tabla
kable(m, format = "markdown", digits = 3)
```

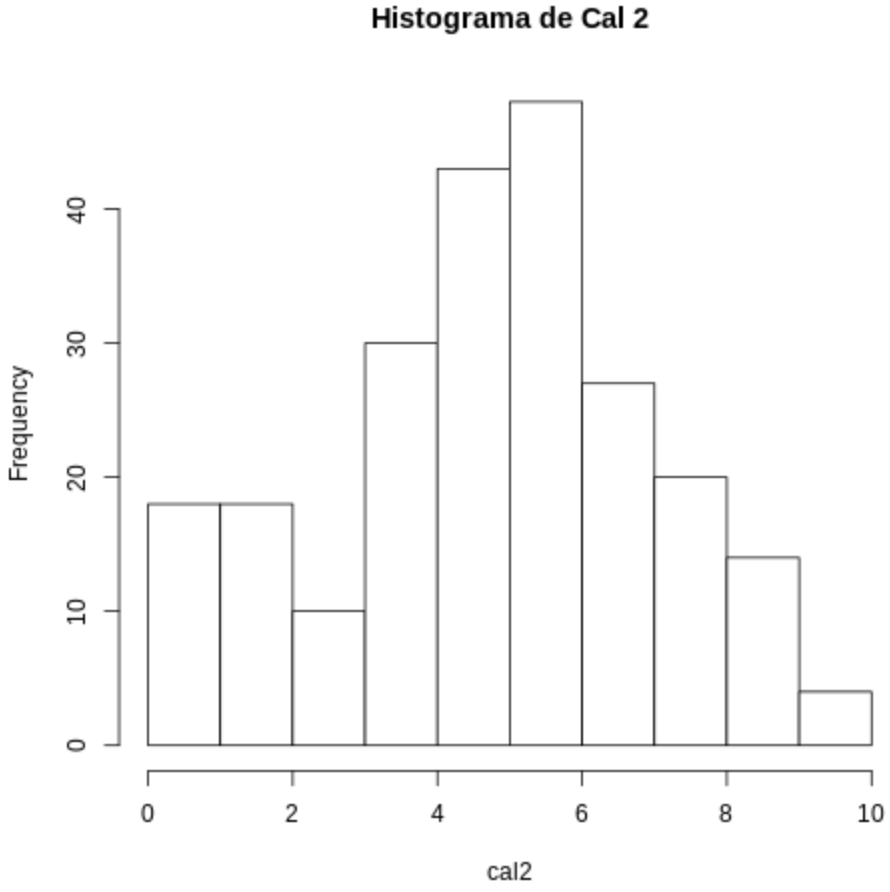
```
        Anderson-Darling normality test

data:  M1
A = 3.4288, p-value = 1.344e-08


        Anderson-Darling normality test

data:  cal1
A = 1.3614, p-value = 0.00155


        Anderson-Darling normality test

data:  cal2
A = 1.6192, p-value = 0.0003591
```

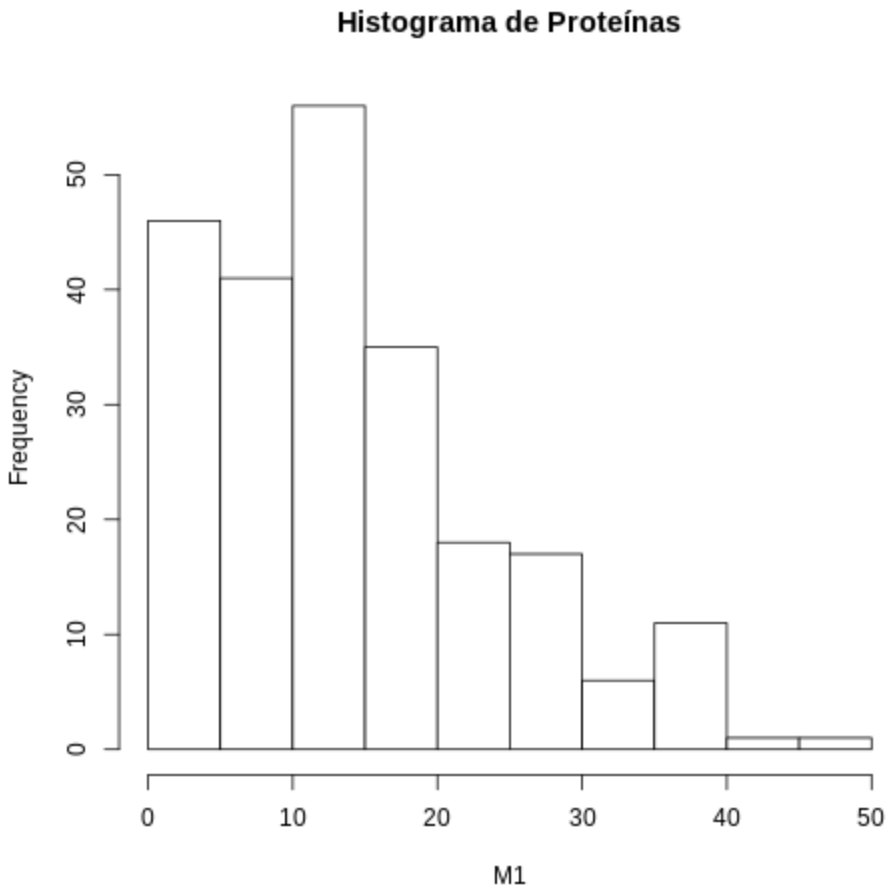
	Minimo	Q1	Mediana	Media	Q3	Máximo	Curtosis	Sesgo	Valor p
Original	1.000	8.00	13.000	14.573	20.000	48.00	0.223	0.785	0.000
Primer modelo	1.414	3.00	3.742	3.722	4.583	7.00	-0.539	0.011	0.002
Segundo Modelo	0.806	3.63	4.864	4.770	6.220	9.93	-0.553	-0.112	0.000

Histogramas

In [ ]:
```
%%R

hist(cal1, col=0, main="Histograma de Cal 1")
hist(cal2, col=0, main="Histograma de Cal 2")
hist(M1, col=0, main="Histograma de Proteínas")
```


Histograma de Cal 1
Frequency
0
10
20
30
40
1
2
3
4
5
6
7
cal1

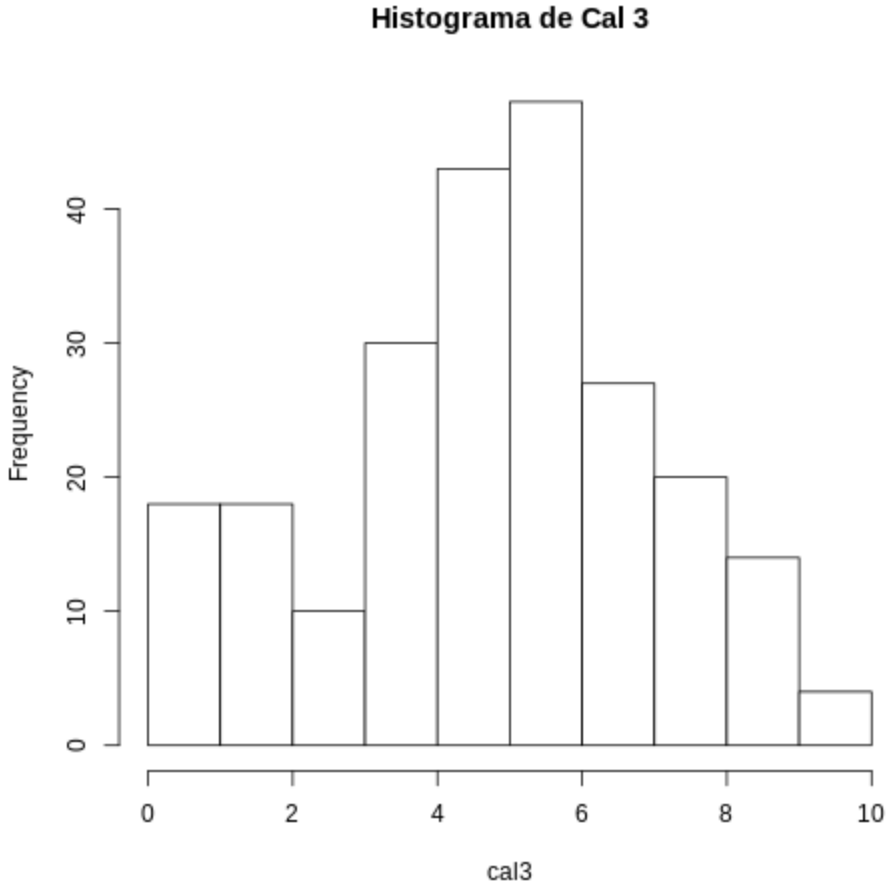

Histograma de Cal 2
Frequency
0
10
20
30
40
0
2
4
6
8
10
cal2

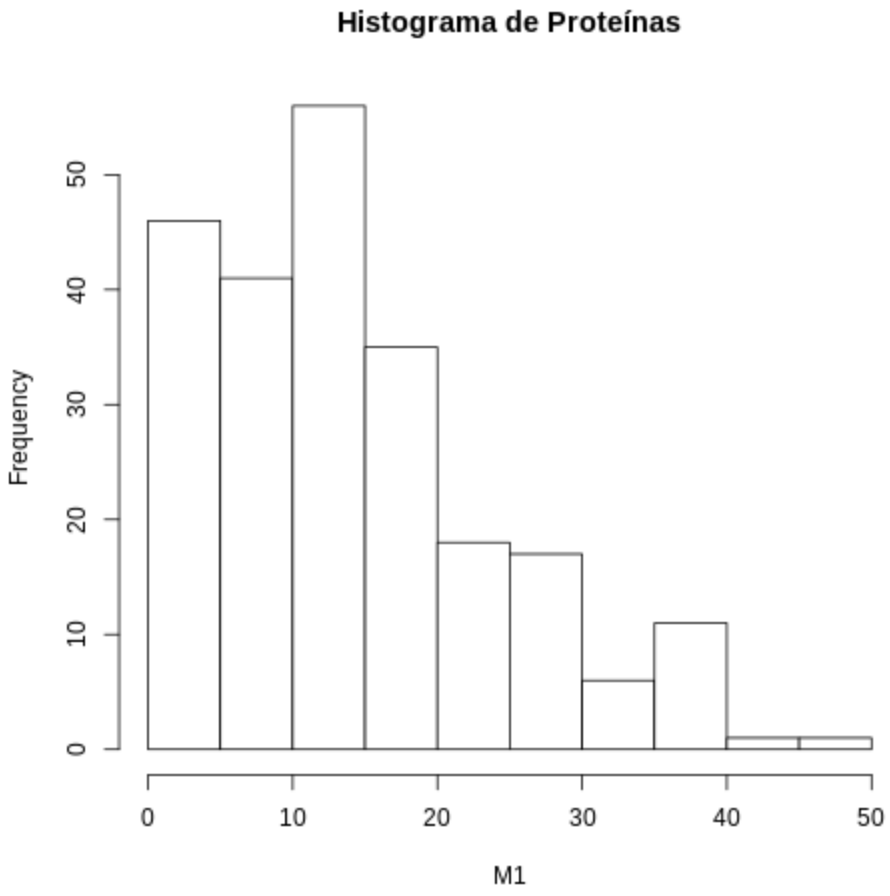


5. Utiliza la transformación de Yeo Johnson y encuentra el valor de lambda que maximiza el valor p de la prueba de normalidad que hayas utilizado (Anderson-Darling o Jarque Bera).

Transformación de Yeo-Johnson

In [ ]:
```
%%R
#Transformación 3 (Fórmula)
cal3<- yeo.johnson(M1, lambda = l)

hist(cal3, col=0, main="Histograma de Cal 3")
hist(M1, col=0, main="Histograma de Proteínas")
```


Histograma de Cal 3
Frequency
0
10
20
30
40
0
2
4
6
8
10
cal3

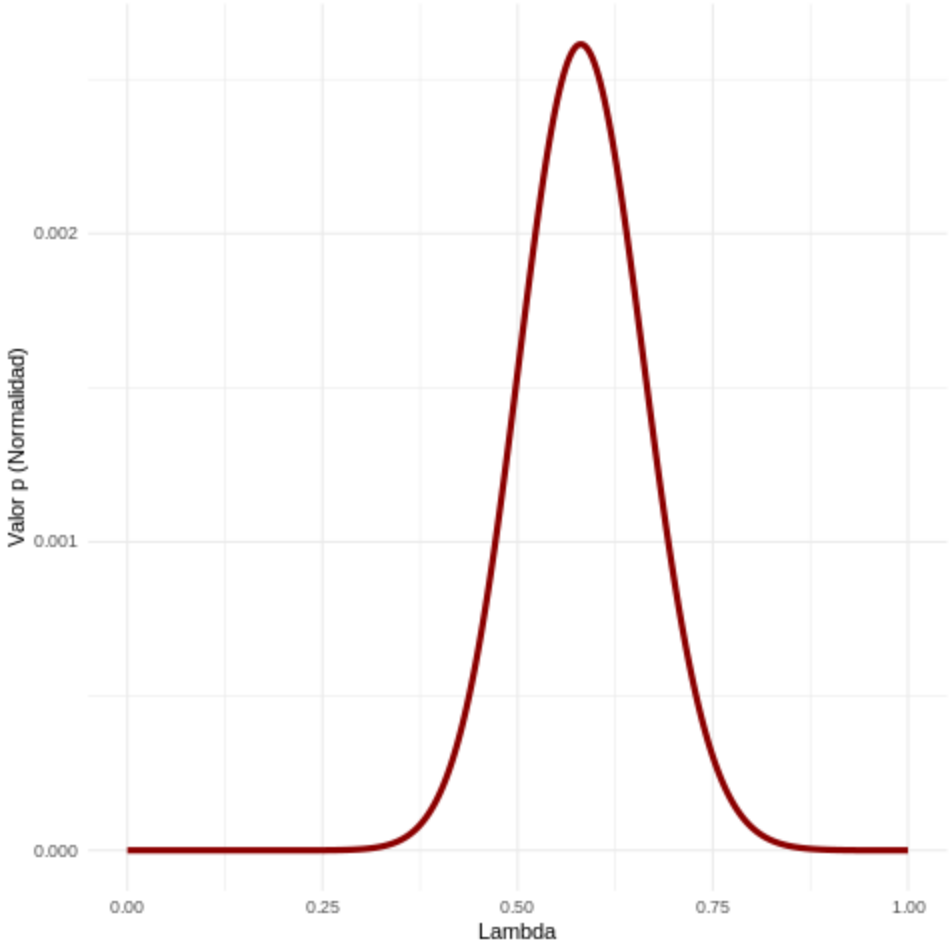


Valor de lambda (l) que maximiza el valor p

In [ ]:

```
%%R

lp <- seq(0, 1, 0.001) # Valores lambda propuestos
nlp <- length(lp)
n <- length(M1)
D <- matrix(as.numeric(NA), ncol = 2, nrow = nlp)
d <- NA

for (i in 1:nlp) {
  d = yeo.johnson(M1, 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 = "Valor p (Normalidad)") +
  theme_minimal()
```



In [ ]:
```
%%R

# Extraemos la fila con el máximo valor-p
G <- data.frame(subset(N, `P-Value` == max(N$`P-Value`)))
kable(G, format = "markdown")
```

	Lambda	P.Value
582	0.581	0.0026137

6. Escribe la ecuación del modelo encontrado.

$$cal_3 = \frac{(x+1)^{0.58}-1}{0.58}$$

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.
2. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.
3. Realiza la prueba de normalidad de Anderson-Darling para los datos transformados y los originales

Tabla Comparativa de Transformaciones

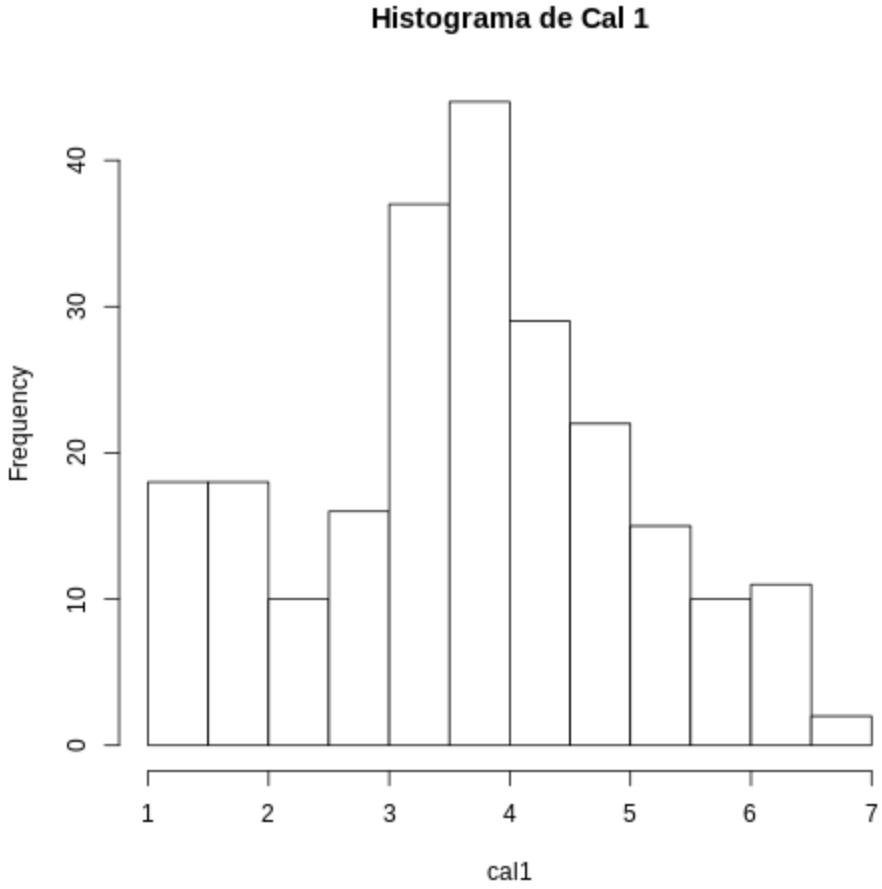
In [ ]:
```
%%R

# Prueba de Normalidad
D0 = ad.test(M1)
D1 = ad.test(cal1)
D2 = ad.test(cal2)
D3 = ad.test(cal3)

print(D0)
print(D1)
print(D2)
print(D3)

# Resumen de Medidas
m0=round(c(as.numeric(summary(M1)),kurtosis(M1),skewness(M1),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$p.value),3)
m3=round(c(as.numeric(summary(cal3)),kurtosis(cal3),skewness(cal3),D3$p.value),3)

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

# Mostramos la tabla
kable(m, format = "markdown", digits = 4)
```

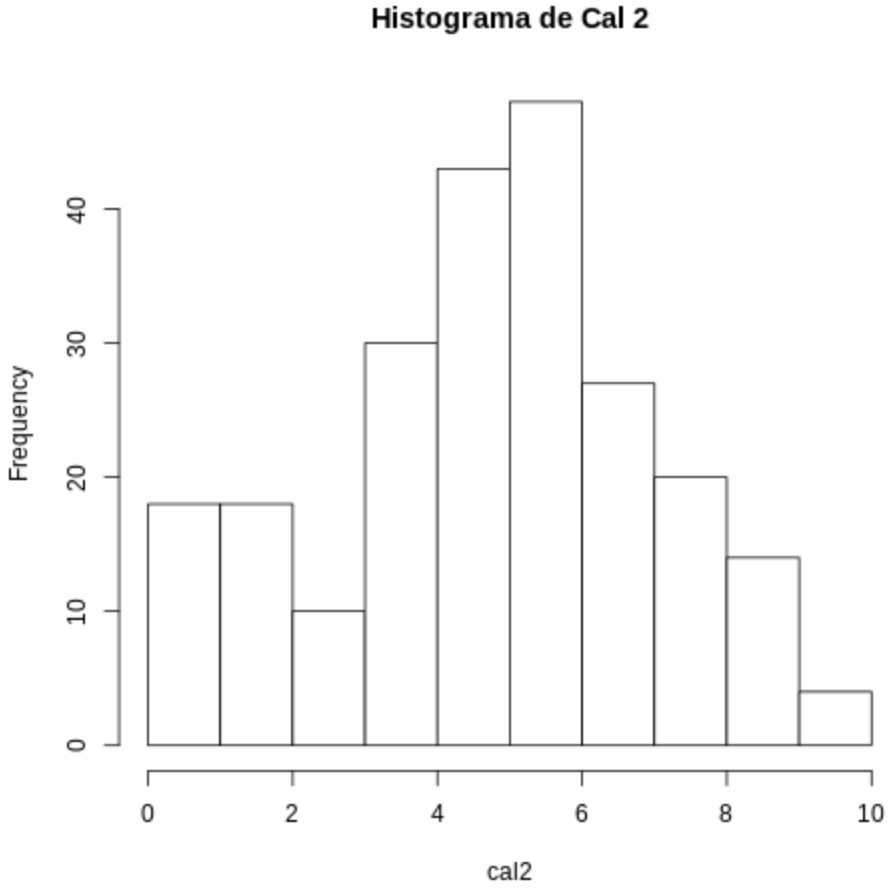
```
        Anderson-Darling normality test

data:  M1
A = 3.4288, p-value = 1.344e-08


        Anderson-Darling normality test

data:  cal1
A = 1.3614, p-value = 0.00155


        Anderson-Darling normality test

data:  cal2
A = 1.6192, p-value = 0.0003591


        Anderson-Darling normality test

data:  cal3
A = 1.6192, p-value = 0.0003591
```

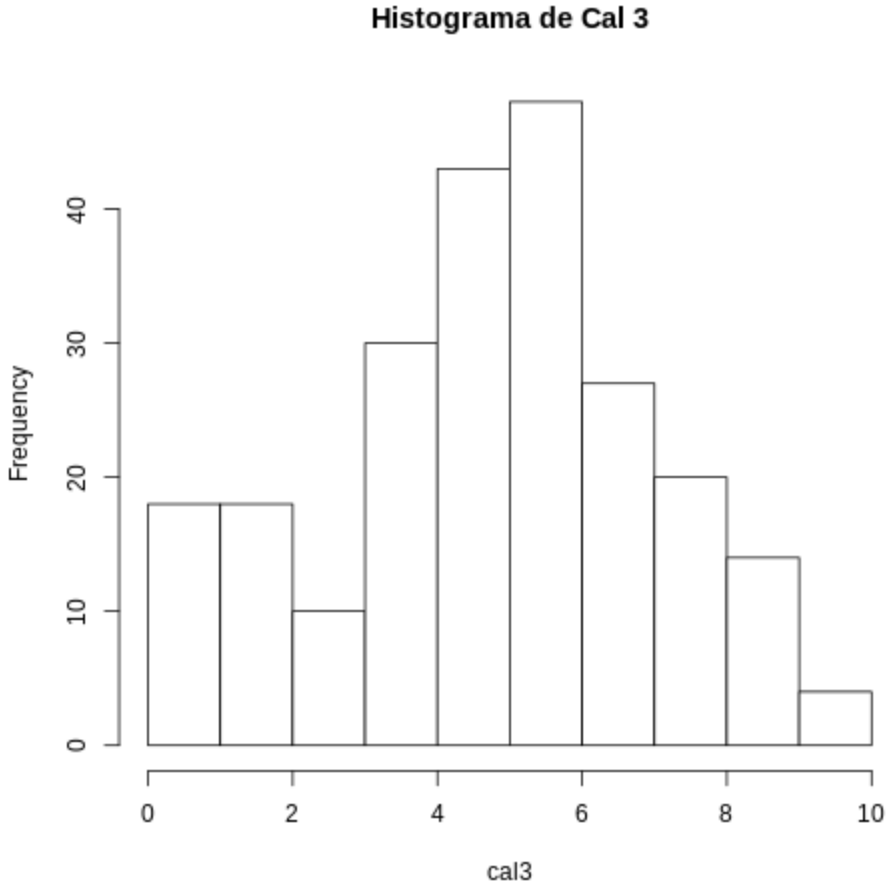
	Minimo	Q1	Mediana	Media	Q3	Máximo	Curtosis	Sesgo	Valor p
Original	1.000	8.00	13.000	14.573	20.000	48.00	0.223	0.785	0.000
Primer modelo	1.414	3.00	3.742	3.722	4.583	7.00	-0.539	0.011	0.002
Segundo Modelo	0.806	3.63	4.864	4.770	6.220	9.93	-0.553	-0.112	0.000
Tercer Modelo	0.806	3.63	4.864	4.770	6.220	9.93	-0.553	-0.112	0.000

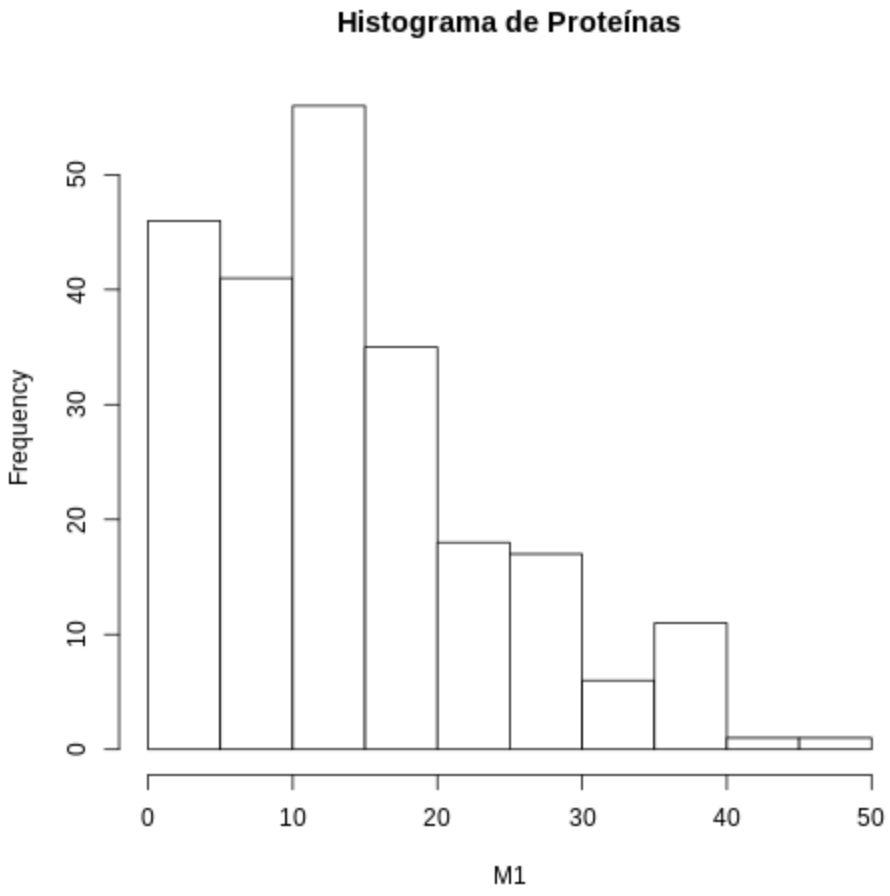
Histogramas comparativos de Transformaciones

In [ ]:
```
%%R

hist(cal1, col=0, main="Histograma de Cal 1")
hist(cal2, col=0, main="Histograma de Cal 2")
hist(cal3, col=0, main="Histograma de Cal 3")
hist(M1, col=0, main="Histograma de Proteínas")
```


Histograma de Cal 1
Frequency
0
10
20
30
40
1
2
3
4
5
6
7
cal1


Histograma de Cal 2
Frequency
0
10
20
30
40
0
2
4
6
8
10
cal2


Histograma de Cal 3
Frequency
0
10
20
30
40
0
2
4
6
8
10
cal3



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

En base a lo visto en clase y durante la elaboración de esta actividad, la mejor forma de determinar la trnasformación de datos más óptima es la que mejor se ajusta a la normalidad, la que minimiza el sesgo y logra obtener un valor-p cercano o mayor que 0.05 (indicando normalidad). Teniendo en cuenta la última tabla e histogramas comparativos, parece ser que tanto el segundo modelo (modelo exacto) y el tecerro (Yeo-Johnson) poseen valores muy similares. Respectivamente son los que tienen el valor más pequeño de senso, el p-value más alto, y una curtosis relativamente baja, lo cual cumple con las características mencionadas anteriormente bacia una distribución cercana a la normal.

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

Ventajas de Box-Cox:

1. Posee mayor simplicidad en cuanto a estimación de parámetros para un valor de lambda fijo.
2. Solo maneja números positivos.
3. Adecuada cuando se cumple el supuesto de varianza constante.

Desventajas de Box-Cox:

1. Limitado a datos positivos.
2. Se supone que los datos son continuos y positivos, además de cercanos a la normal.

Ventajas de Yeo-Johnson:

1. Es más flexible que Box-Cox, en el sentido de que puede manejar tanto datos positivos como negativos.
2. Proporciona mejores resuktados para datos que no puedan adherirse estrictamente a las suposiciones de normalidad.

Desventajas de Yeo-Johnson:

1. Su fórmula es un poco más compleja.
2. No siempre proporciona mejores resultados que Box-Cox, esto depende enteramente de los datos a analizar.

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.
    A. Propósito: La transformación de los datos ajusta su distribución para encontrar cietas suposiciones, en este caso se trata de la normalidad. Por otro lado, el escalamiento intenta reescalar los datos para obtener una escala consistente, usualmente entre 0 y 1, o bien con una media de 0 y una desviación estándar de 1.
    B. Efecto sobre los datos: La transformación cambia la distribución y forma de los datos, en cambio el escalamiento no cambia la distribución, solo escala los datos ya existentes sin alterar su forma.
    C. Suposiciones: La tranformación asume que los datos transformados coincidirán con suposiciones muy específicas (normalidad). Por otro lado, el escalamiento asume que la distribución original de los datos no se encuentra significativamente afectada, siendo así que se enfoca en las distancias relativas.
2. Indica cuándo es necesario utilizar cada uno.
    A. Se recomienda utilizar la transformación de los datos cuando su distribución no sea significativamente normal, y las suposiciones de pruebas estadísticas son incumplidos. Es muy útil para convertir los datosa algo más consistente para métodos de análisis que suponen normalidad.
    B. Se recomienda utilizar el escalamiento cuando diferentes características se encuentran en distintas escalas y se requiera trasladarlos a una escala común.