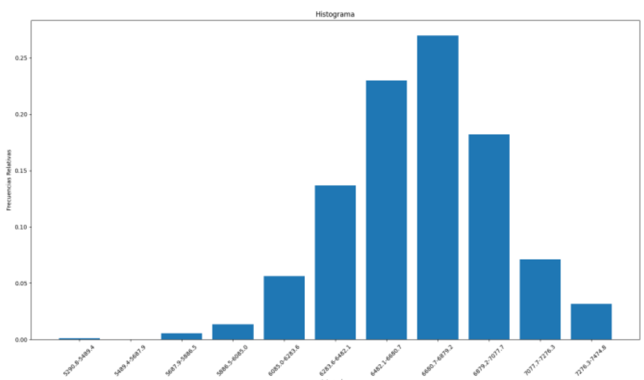
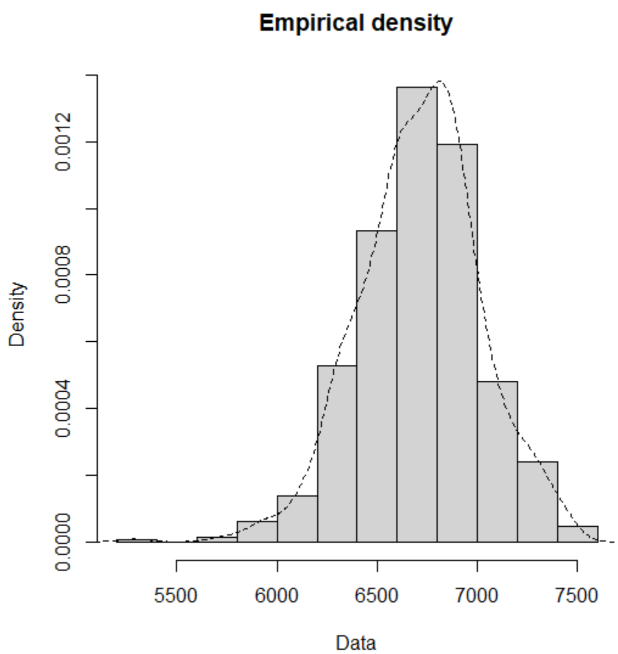
Jaider Castañeda Villa

CC 1001015442

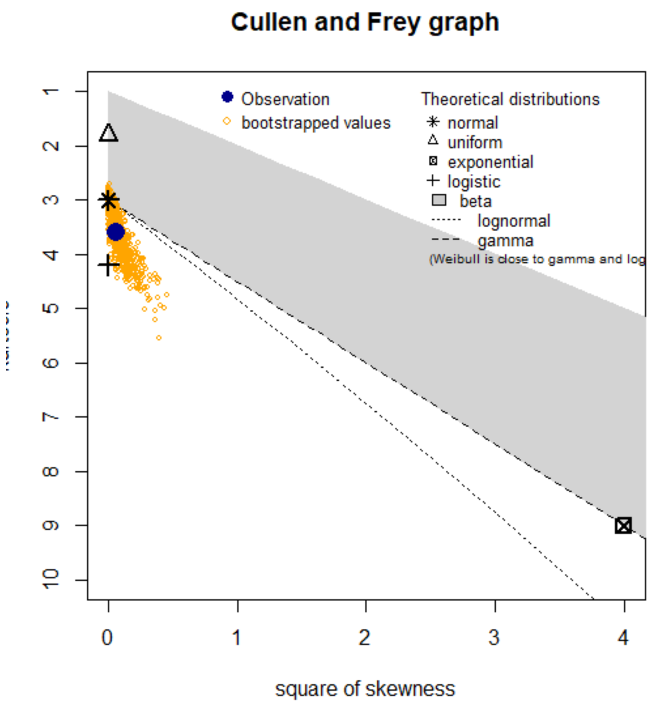
1. Comparar el histograma de frecuencias obtenido con las funciones de R con uno obtenido con el método descrito en el libro de ejercicios.

   R/ Con el método descrito los histogramas quedan de manera diferente teniendo una densidad mayor en la parte izquierda del pico.





2. Siguiendo el procedimiento descrito en la clase 7 y el libro de R "¿Cómo ajustar datos de entrada? Proponer y ajustar modelos para representar la muestra de datos suministrada.



Viendo la estimación de asimetría y el valor de la Kurtosis podría decir que se acerca a las distribuciones, normal, logistic, log normal, gamma y Weibull.
Solo viendo el grafico de densidad uno podría suponer una distribución normal.

A continuación, se hará una evaluación de ajuste a las distribuciones para ver cuál es la más adecuada para representar los datos.

Para el valor critico D para la prueba de ajuste Kolmogorov Smirnov va a tomar el siguiente valore $D_{0.05,730}$=0.05

H0: La muestra se distribuye como la distribución
H1: La muestra no se distribuye como la distribución

Si $D_{estimado}$ > $D_{0.05,730}$ Se rechaza la nula por lo que podemos decir que los datos no se distribuyen como esa distribución en la siguiente tabla se marcaran las que se pueda decir que los datos se distribuyen como tal.

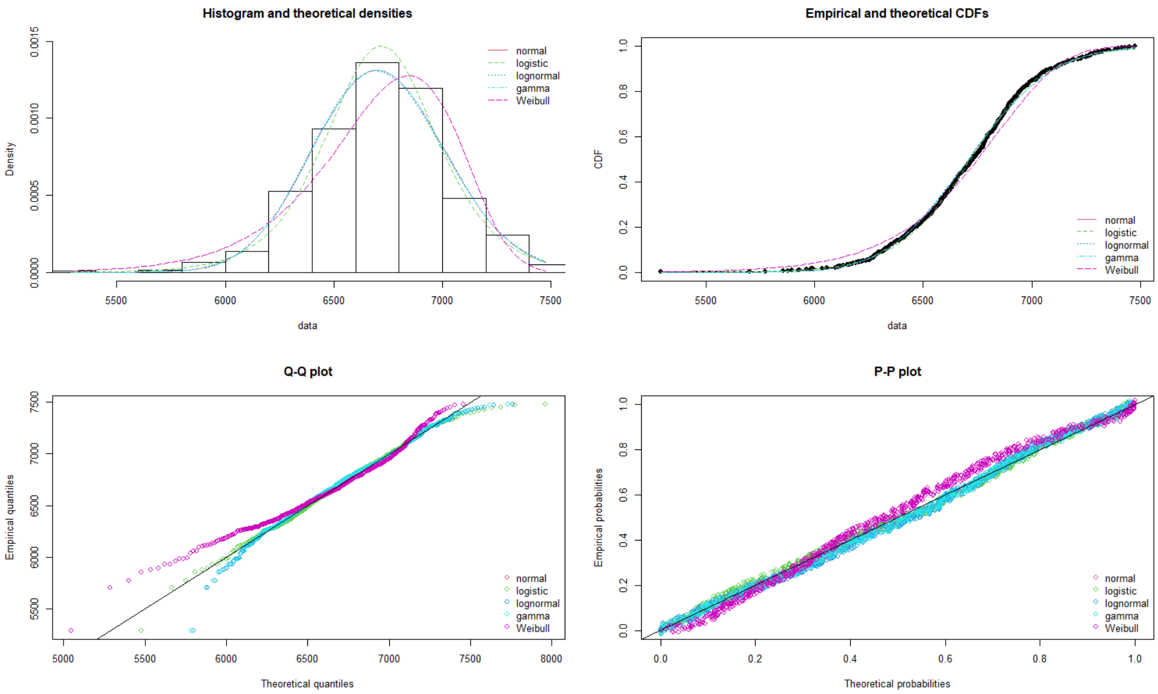
Goodness-of-fit statistics

	1-mle-gamma	2-mle-logis	3-mle-lnorm	4-mle-gamma	5-mle-weibull
Kolmogorov-Smirnov statistic	0.03300634	0.02960387	0.03555651	0.03300634	0.06236909
Cramer-von Mises statistic	0.18812564	0.08400074	0.22840051	0.18812564	0.82041727
Anderson-Darling statistic	1.04786278	0.57360571	1.26536570	1.04786278	5.89267472

Como se puede ver todos excepto Weibull, los valores D son menores que el crítico por lo que teóricamente los datos se pueden distribuir como todas excepto Weibull, ahora pasaremos al siguiente criterio que es el valor mínimo de Akaike’s o Bayesian

```
Goodness-of-fit criteria
                               1-mle-gamma 2-mle-logis 3-mle-lnorm 4-mle-gamma 5-mle-weibull
Akaike's Information Criterion    10423.77    10415.58    10428.19    10423.77      10467.41
Bayesian Information Criterion    10432.96    10424.77    10437.37    10432.96      10476.60
```

Aquí la distribución logistic es la que tiene el menor valor en ambos criterios de información por lo que podemos tomarla como la distribución más adecuada para nuestros datos.
A continuación, se harán las gráficas para determinar definitivamente si la logistic es la más adecuada.



Se puede apreciar que en los diagramas la distribución que sigue más de cerca la muestra de datos es la logistic, principalmente en el Q-Qplot que al inicio y casi hasta el final sigue la muestra.

3. Concluir cuál es el modelo que mejor se ajusta a los datos.

El modelo que mas se ajusta a los datos por las razones previamente mencionadas es el Logistic.

//Codigo R

```
library("MASS")
library("car")
library("fitdistrplus")
library("moments")

tmin<-read.table("demcom1.txt")
x<-tmin$V1
```

```r
#Medidas de posicion
mean(x)
median(x)
mode(x)

#medidas de forma
skewness(x,na.rm = FALSE)
kurtosis(x,na.rm = FALSE)

##Graficar los datos
plotdist(x,histo=TRUE,demp = TRUE)
summary(x)

#Valores
descdist(x,boot = 1000) #Saca las medidas de arriba xd
#Ademas del grafico de Cullen
# Ajusta las distribuciones
fit_n <- fitdist(x, "norm")
fit_lg <- fitdist(x, "logis")
fit_ln <- fitdist(x, "lnorm")
fit_g <- fitdist(x, "gamma")
fit_w <- fitdist(x, "weibull")

gofstat(list(fit_g,fit_lg,fit_ln,fit_g,fit_w))

par(mfrow=c(2,2))

plot.legend<-c("normal","logistic","lognormal", "gamma","Weibull")
denscomp(list(fit_g,fit_lg,fit_ln,fit_g,fit_w),legendtext = plot.legend)
cdfcomp(list(fit_g,fit_lg,fit_ln,fit_g,fit_w),legendtext = plot.legend)
qqcomp(list(fit_g,fit_lg,fit_ln,fit_g,fit_w),legendtext = plot.legend)
ppcomp(list(fit_g,fit_lg,fit_ln,fit_g,fit_w),legendtext = plot.legend)
```

//Codigo Python

```python
import math
import matplotlib.pyplot as plt

with open('demcom1.txt', 'r') as archivo:
    datos = [float(linea.strip()) for linea in archivo.readlines()]

n = len(datos)
k = math.ceil(math.log(n,2) + 1)

min_valor = min(datos)
```

```
max_valor = max(datos)
intervalos = [min_valor + i * (max_valor - min_valor) / k for i in
range(k)]+[max_valor]
frecuencias = [0] * (k)

for dato in datos:
    for i in range(k):
            if dato >= intervalos[i] and dato < intervalos[i + 1]:
                frecuencias[i] += 1
print(sum(frecuencias))
frecuencias_relativas = [frecuencia / n for frecuencia in frecuencias]

fig = plt.figure()
ax = fig.add_subplot(111)
x = range(1, k + 1)
ax.bar(x, frecuencias_relativas)

# Etiquetas para el eje X
rangos_etiquetas = [f"{intervalos[i]:.1f}-{intervalos[i + 1]:.1f}" for i in
range(k)]
ax.set_xticks(x)
ax.set_xticklabels(rangos_etiquetas, rotation=45)

plt.ylabel("Frecuencias Relativas")
plt.xlabel("Intervalos")
plt.title("Histograma")
plt.show()
```