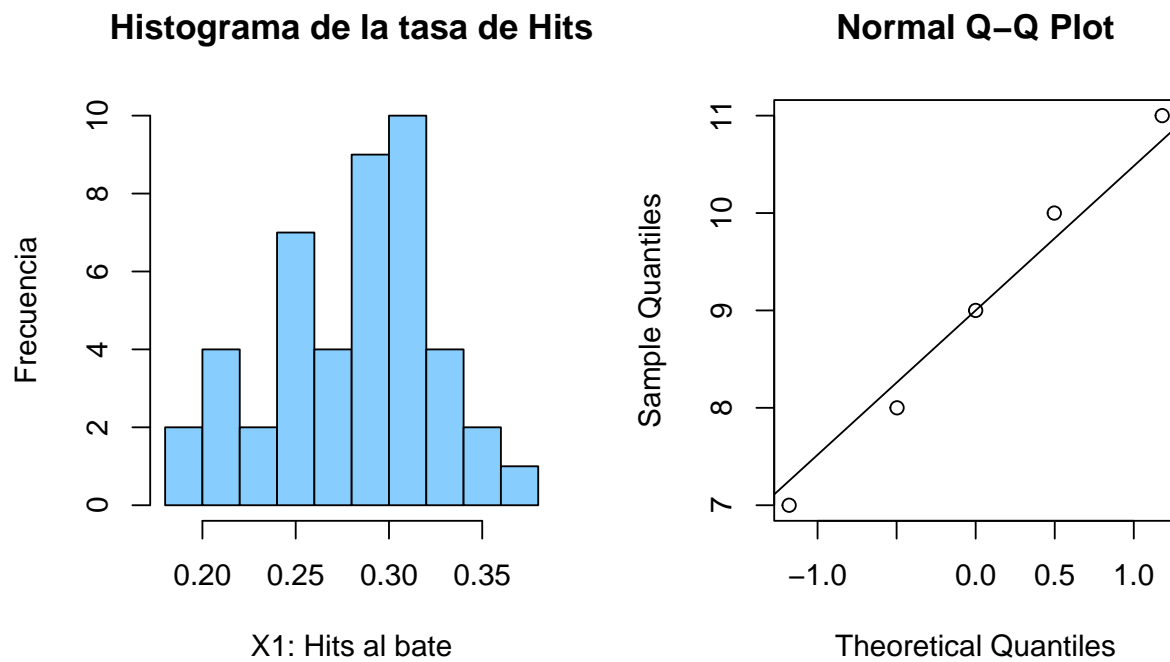


prueba de bondad de ajuste X1

Luis Riera

20 de abril de 2022

Usando el comando `hist` podemos obtener el histograma de las variables



Obtenemos entonces que si subdividimos en intervalos de longitud 0.02 , las frecuencias son:

<i>Intervalo</i>	0,18	0,20	0,22	0,24	0,26	0,28	0,30	0,32	0,34	0,36	0,38
<i>Frecuencia</i>	2	4	2	4	7	9	10	4	2	1	

<i>Intervalo</i>	0,18	0,20	0,22	0,24	0,26	0,28	0,30	0,32	0,34	0,36	0,38
<i>Frecuencia</i>	2	4	2	4	7	9	10	4	2	1	

Ahora agruparemos los datos en categorías de frecuencia de por lo menos 5 (para poder aplicar el metodo de bondad de ajuste):

- Primera categoría (o clase): $[0.18, 0.24)$ con frecuencia observada $f_1 = 8$
- Segunda categoría (o clase): $[0.24, 0.28)$ con frecuencia observada $f_2 = 11$
- Tercera categoría (o clase): $[0.28, 0.30)$ con frecuencia observada $f_3 = 9$
- Cuarta categoría (o clase): $[0.30, 0.32)$ con frecuencia observada $f_4 = 10$
- Quinta categoría (o clase): $[0.32, 0.38]$ con frecuencia observada $f_5 = 7$

Los datos agrupados parecieran ajustarse bastante bien a una distribución normal

```
# Tamaño de los fi
(k<- length(fi))

## [1] 5

# Numero total de datos
n<- sum(fi)

# Puntos medios de los intervalos
(mi<-c(0.18+(0.24-0.18)/2,0.24+(0.28-0.24)/2,0.28+(0.30-0.28)/2,0.30+(0.32-0.30)/2, 0.32+(0.38-0.32)/2))

## [1] 0.21 0.26 0.29 0.31 0.35

# Media de los datos
(xbarra<-sum(fi*mi)/n)

## [1] 0.2822222

# Vector con las medias
x_barra<-rep(xbarra,k)

# Varianza
(S_cuadrado<-sum(fi*(mi-x_barra)^{2})/(n-1))

## [1] 0.001990404

# Desviación estandar
(S<-sqrt(S_cuadrado))

## [1] 0.04461394

calculemos los pi

# P(Z<0.24)
(p1<-pnorm(0.24,mean= xbarra,sd=S))

## [1] 0.1719747

# P(0.24 < Z < 0.28)
(p2<-pnorm(0.28,mean= xbarra,sd=S)-pnorm(0.24,mean= xbarra,sd=S))

## [1] 0.3081622

(p3<-pnorm(0.30,mean= xbarra,sd=S)-pnorm(0.28,mean= xbarra,sd=S))

## [1] 0.174725

(p4<-pnorm(0.32,mean= xbarra,sd=S)-pnorm(0.30,mean= xbarra,sd=S))

## [1] 0.1465766

(p5<-pnorm(0.32,mean= xbarra,sd=S, lower.tail = F))

## [1] 0.1985615

# Vector con las probabilidades
(pi<-c(p1,p2,p3,p4,p5))

## [1] 0.1719747 0.3081622 0.1747250 0.1465766 0.1985615

# Suma de las probabilidades
(sum(pi))
```

```
## [1] 1
# Estadístico
(t<-sum(((fi-n*pi)^(2))/(n*pi)))

## [1] 2.942129
# p-valor
(p_Valor<- 1-pchisq(t,k-1-2))

## [1] 0.2296808
```