

ConstruccionModeloEstadisticoBase

Ariadna Jocelyn Guzmán Jiménez - A01749373

2022-09-14

Contaminación por mercurio

Descripción del problema

La contaminación por mercurio de peces en el agua dulce comestibles es una amenaza directa contra nuestra salud. Se lo llevó a cabo un estudio reciente en 53 lagos de Florida con el fin de examinar los factores que influían en el nivel de contaminación por mercurio.

Exploración de la base de datos

Lectura de datos

En la parte de la lectura de datos, importamos nuestra base y por otra parte, hacemos una nueva variable que solo cuente con los datos numéricos, ya que son los que nos servirán para nuestros modelos. Se omiten solamente la columna 1 y 2 ya que la primera sólo es un id para cada fila de datos y la segunda es el nombre del lago.

```
mercurio = read.csv("mercurio.csv")  
numericMercurio = mercurio[3:12]
```

Entendimiento de los datos

En nuestra base de datos, encontramos los siguiente atributos:

- **X1** = número de indentificación
- **X2** = nombre del lago
- **X3** = alcalinidad (mg/l de carbonato de calcio)
- **X4** = PH
- **X5** = calcio (mg/l)
- **X6** = clorofila (mg/l)
- **X7** = concentración media de mercurio (parte por millón) en el tejido muscular del grupo de peces estudiados en cada lago
- **X8** = número de peces estudiados en el lago
- **X9** = mínimo de la concentración de mercurio en cada grupo de peces
- **X10** = máximo de la concentración de mercurio en cada grupo de peces
- **X11** = estimación (mediante regresión) de la concentración de mercurio en el pez de 3 años (o promedio de mercurio cuando la edad no está disponible)
- **X12** = indicador de la edad de los peces (0: jóvenes; 1: maduros)

Dado lo anterior visualizamos medidas generales de nuestro dataset:

Dimensión del dataset

```
dimension = dim(mercurio)
cat("Número de filas:", dimension[1])
```

```
## Número de filas: 53
```

```
cat("\nNúmero de columnas:", dimension[2])
```

```
##
```

```
## Número de columnas: 12
```

```
cat("\nCantidad de datos:", dimension[1] * dimension[2])
```

```
##
```

```
## Cantidad de datos: 636
```

Cálculo de medidas estadísticas

Análisis estadístico

```
summary(mercurio)
```

```
##           X1           X2           X3           X4
## Min.      : 1   Length:53   Min.      : 1.20   Min.      :3.600
## 1st Qu.:14   Class :character 1st Qu.: 6.60   1st Qu.:5.800
## Median :27   Mode  :character Median :19.60   Median :6.800
## Mean      :27                                Mean      :37.53   Mean      :6.591
## 3rd Qu.:40                                3rd Qu.: 66.50   3rd Qu.:7.400
## Max.      :53                                Max.      :128.00   Max.      :9.100
##           X5           X6           X7           X8
## Min.      : 1.1   Min.      : 0.70   Min.      :0.0400   Min.      : 4.00
## 1st Qu.: 3.3   1st Qu.: 4.60   1st Qu.:0.2700   1st Qu.:10.00
## Median :12.6   Median :12.80   Median :0.4800   Median :12.00
## Mean      :22.2   Mean      :23.12   Mean      :0.5272   Mean      :13.06
## 3rd Qu.:35.6   3rd Qu.:24.70   3rd Qu.:0.7700   3rd Qu.:12.00
## Max.      :90.7   Max.      :152.40   Max.      :1.3300   Max.      :44.00
##           X9           X10          X11          X12
## Min.      :0.0400   Min.      :0.0600   Min.      :0.0400   Min.      :0.0000
## 1st Qu.:0.0900   1st Qu.:0.4800   1st Qu.:0.2500   1st Qu.:1.0000
## Median :0.2500   Median :0.8400   Median :0.4500   Median :1.0000
## Mean      :0.2798   Mean      :0.8745   Mean      :0.5132   Mean      :0.8113
## 3rd Qu.:0.3300   3rd Qu.:1.3300   3rd Qu.:0.7000   3rd Qu.:1.0000
## Max.      :0.9200   Max.      :2.0400   Max.      :1.5300   Max.      :1.0000
```

```
library(modeest)
n = ncol(numericMercurio)
d = matrix(NA, ncol = 8, nrow = n)
```

```

for (i in 1:n){
  d[i,] = c(as.numeric(summary(numericMercurio[,i])), sd(numericMercurio[,i]), mlv(numericMercurio[,i]),
}
m = as.data.frame(d)
row.names(m) = c("X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10", "X11", "X12")
names(m) = c("Minimo", "Q1", "Mediana", "Media", "Q3", "Maximo", "Desviacion Estandar", "Moda")
m

```

##	Minimo	Q1	Mediana	Media	Q3	Maximo	Desviacion Estandar	Moda
## X3	1.20	6.60	19.60	37.5301887	66.50	128.00	38.2035267	17.30
## X4	3.60	5.80	6.80	6.5905660	7.40	9.10	1.2884493	5.80
## X5	1.10	3.30	12.60	22.2018868	35.60	90.70	24.9325744	3.00
## X6	0.70	4.60	12.80	23.1169811	24.70	152.40	30.8163214	1.60
## X7	0.04	0.27	0.48	0.5271698	0.77	1.33	0.3410356	0.34
## X8	4.00	10.00	12.00	13.0566038	12.00	44.00	8.5606773	12.00
## X9	0.04	0.09	0.25	0.2798113	0.33	0.92	0.2264058	0.04
## X10	0.06	0.48	0.84	0.8745283	1.33	2.04	0.5220469	0.06
## X11	0.04	0.25	0.45	0.5132075	0.70	1.53	0.3387294	0.16
## X12	0.00	1.00	1.00	0.8113208	1.00	1.00	0.3949977	1.00

La función `summary` aplicada para la base de datos, nos permitió visualizar un resumen general en medidas estadísticas de cada uno de los datos almacenados en la base “mercurio”, sin embargo, se realizó posteriormente un dataframe que almacene 2 medidas más que no están consideradas en `summary`, desplegando de una forma más visual e interactiva el cálculo de cada uno de los datos.

Visualización de datos

Variables cuantitativas*

- X3

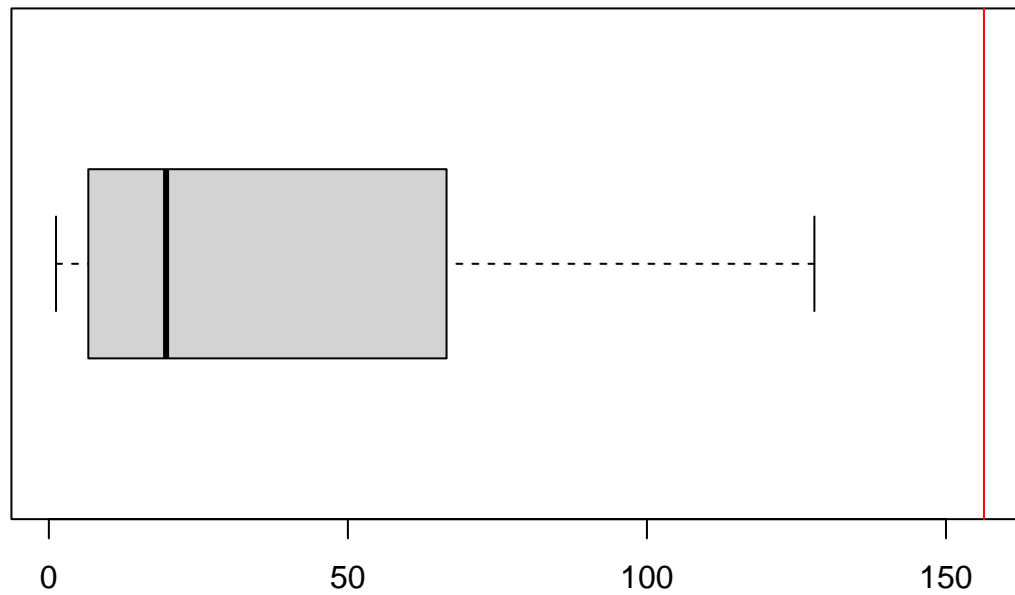
```

# Boxplot y valores atípicos
alcalino = mercurio$X3
q1_a = quantile(alcalino, 0.25)
q3_a = quantile(alcalino, 0.75)
Rc_a = q3_a - q1_a
y2_a = q3_a + 1.5 * Rc_a

boxplot(alcalino, horizontal=TRUE, ylim=c(0,y2_a) , main= "Boxplot alcalinidad")
abline(v=y2_a, col="red") #línea vertical en el límite de los datos atípicos

```

Boxplot alcalinidad

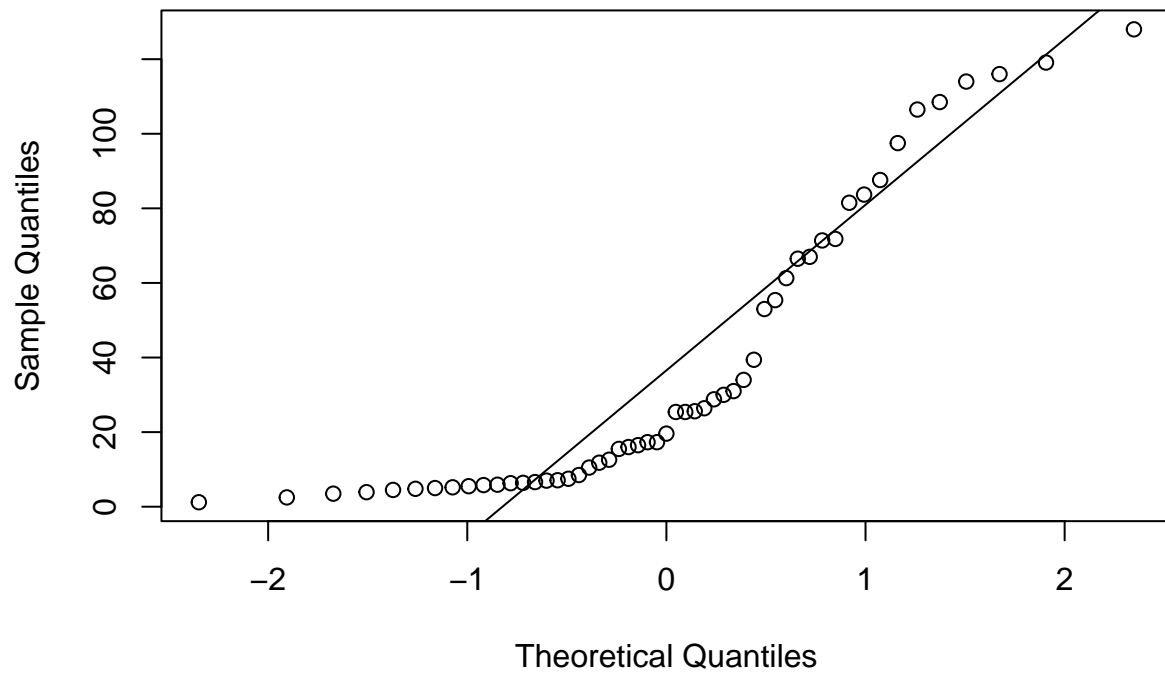


```
X= numericMercurio[alcalino<y2_a,c("X3")] #Quitar datos atípicos de la matriz db en la variable X
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.20   6.60   19.60   37.53   66.50   128.00
```

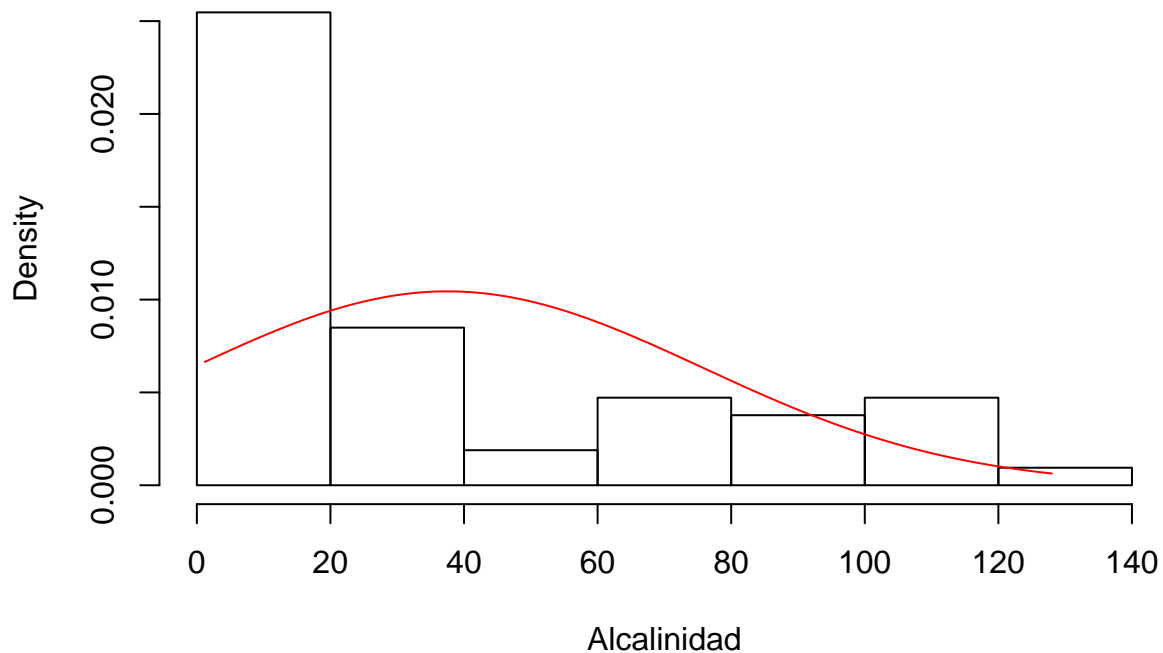
```
## QQplot
qqnorm(alcalino, main = "QQplot alcalinidad")
qqline(alcalino)
```

QQplot alcalinidad



```
# Histograma
hist(alcalino, prob = TRUE, col = 0, main = "Histograma de alcalinidad", xlab = "Alcalinidad")
x = seq(min(alcalino), max(alcalino), 0.1)
y = dnorm(x, mean(alcalino), sd(alcalino))
lines(x, y, col = "red")
```

Histograma de alcalinidad



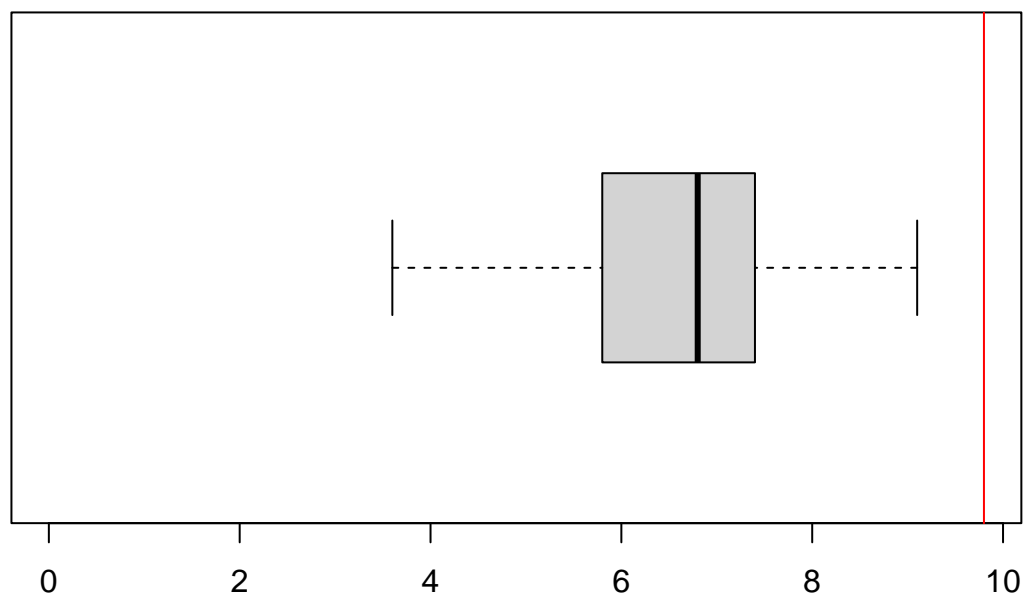
Tanto el histograma como el boxplot de alcalinidad, nos muestran una concentración de los datos hacia la izquierda, logrando visualizar un sesgo a la derecha indicando asimetría de la distribución con respecto a la media y mediana. Finalmente, de acuerdo al QQplot, se muestra una curtosis alta con una distribución leptocúrtica.

- X4

```
# Boxplot y valores atípicos
ph = mercurio$X4
q1_ph = quantile(ph, 0.25)
q3_ph = quantile(ph, 0.75)
Rc_ph = q3_ph - q1_ph
y2_ph = q3_ph + 1.5 * Rc_ph

boxplot(ph, horizontal=TRUE, ylim=c(0, y2_ph), main= "Boxplot PH")
abline(v=y2_ph, col="red") #línea vertical en el límite de los datos atípicos
```

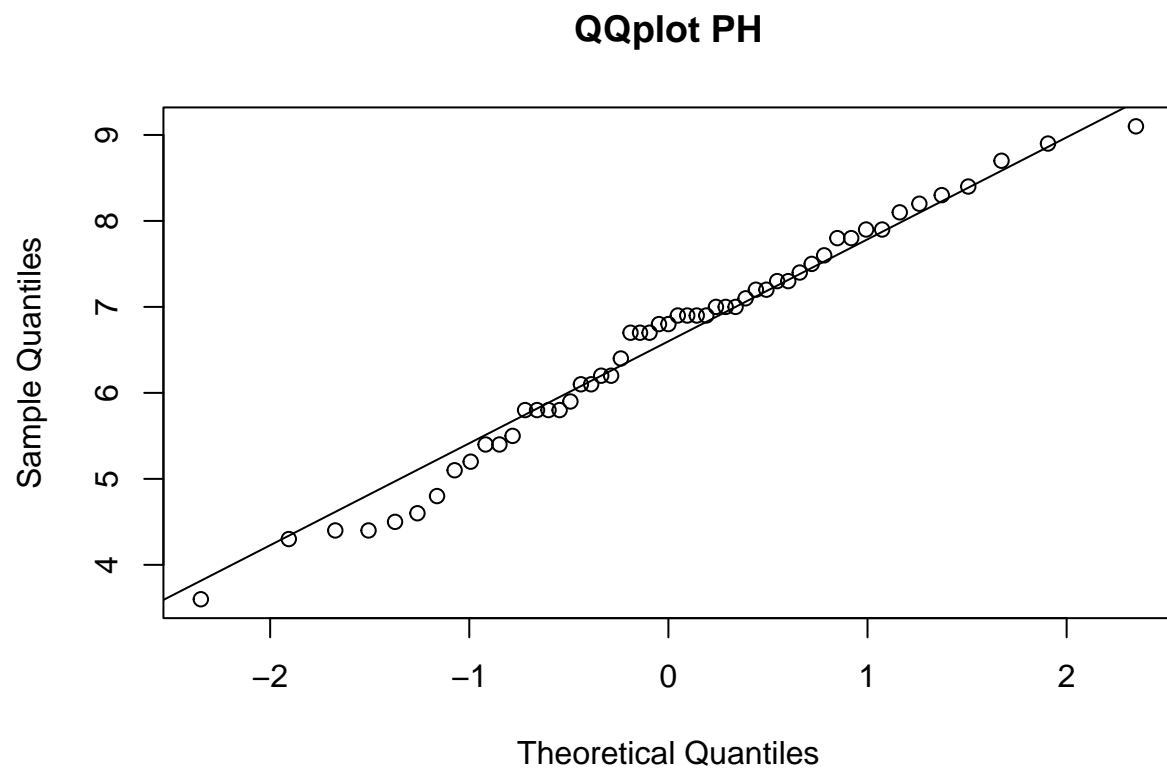
Boxplot PH



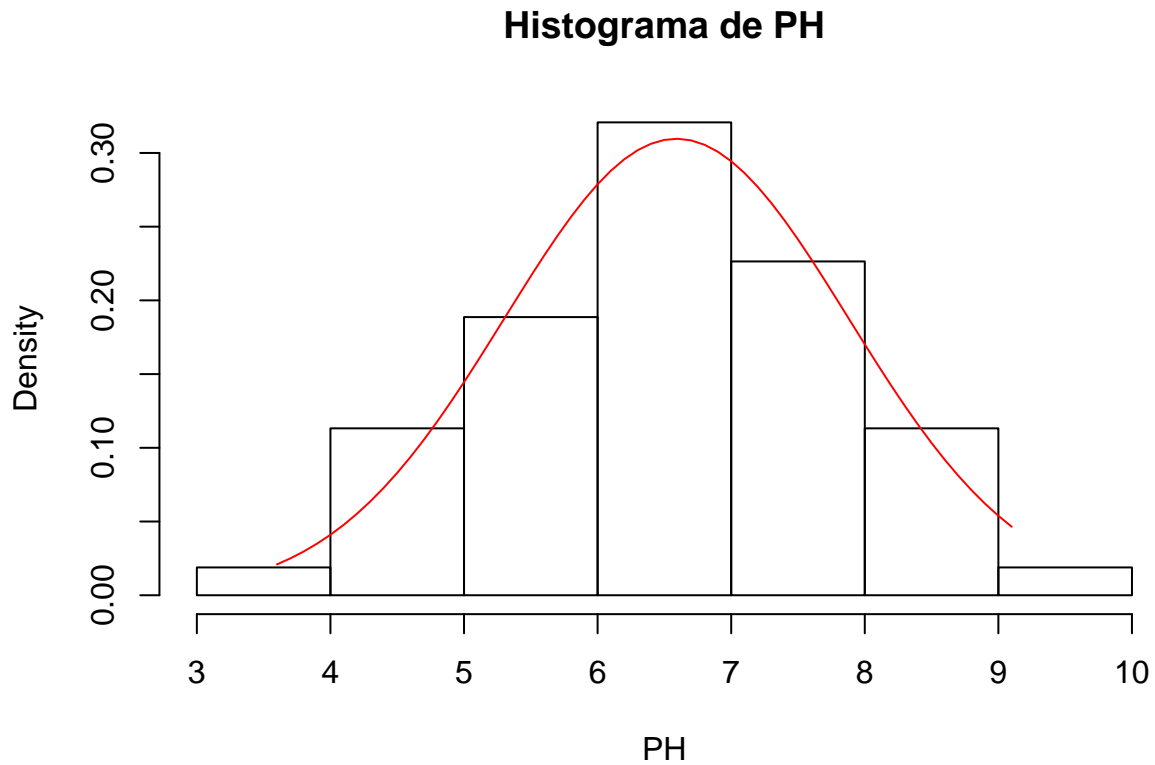
```
X= numericMercurio[ph<y2_ph,c("X4")] #Quitar datos atípicos de la matriz db en la variable X  
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##   3.600   5.800   6.800   6.591   7.400   9.100
```

```
## QQplot  
qqnorm(ph, main = "QQplot PH")  
qqline(ph)
```



```
# Histograma  
hist(ph, prob = TRUE, col = 0, main = "Histograma de PH", xlab = "PH")  
x = seq(min(ph), max(ph), 0.1)  
y = dnorm(x, mean(ph), sd(ph))  
lines(x, y, col = "red")
```

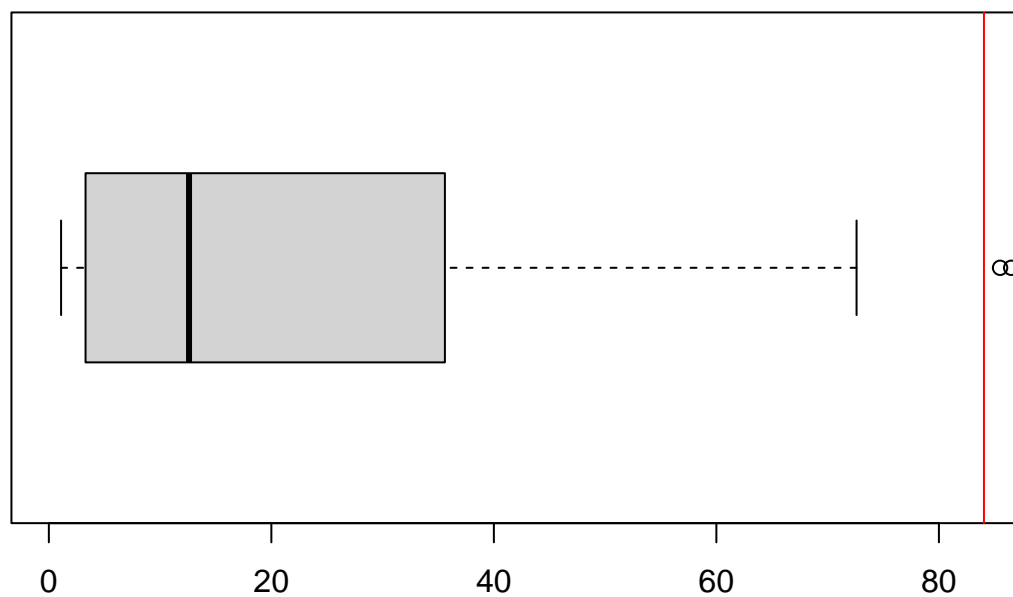
El histograma y el boxplot nos muestran una simetría en la distribución respecto a la media y mediana, logrando ver en el QQplot que la probabilidad normal es la ideal para la variable PH.

- X5

```
# Boxplot y valores atípicos
calcio = mercurio$X5
q1_cal = quantile(calcio, 0.25)
q3_cal = quantile(calcio, 0.75)
Rc_cal = q3_cal - q1_cal
y2_cal = q3_cal + 1.5 * Rc_cal

boxplot(calcio, horizontal=TRUE, ylim=c(0, y2_cal), main= "Boxplot calcio")
abline(v=y2_cal, col="red") #línea vertical en el límite de los datos atípicos
```

Boxplot calcio

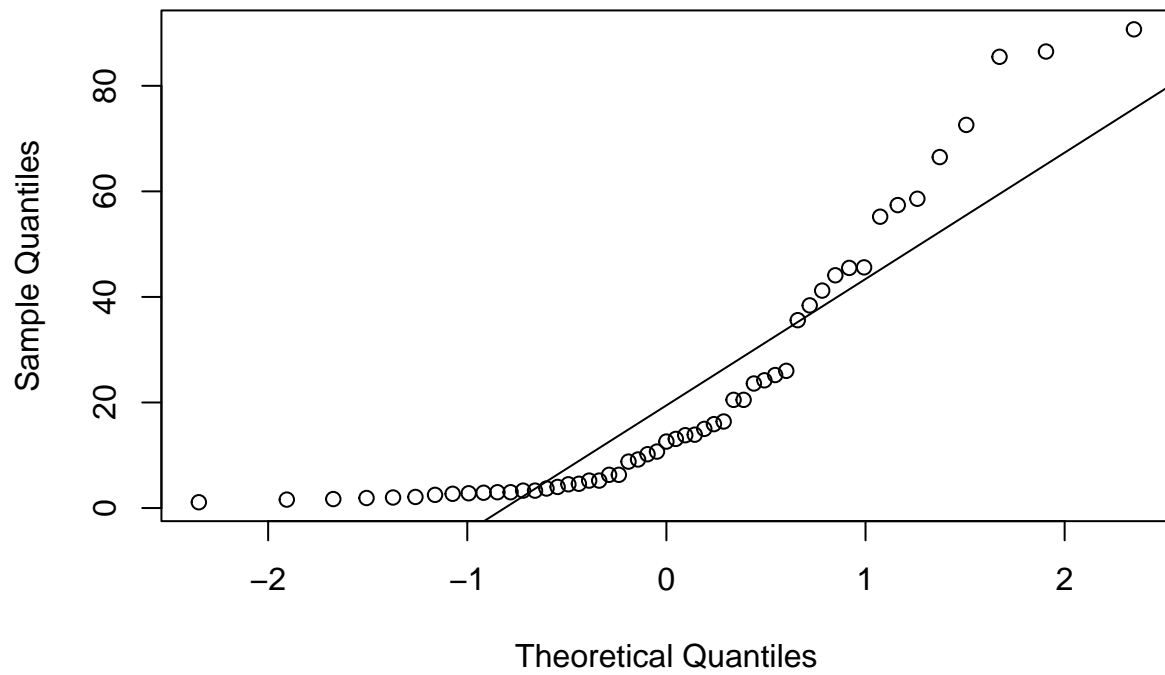


```
X= numericMercurio[calcio<y2_a,c("X5")] #Quitar datos atípicos de la matriz db en la variable X
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.1     3.3    12.6    22.2    35.6    90.7
```

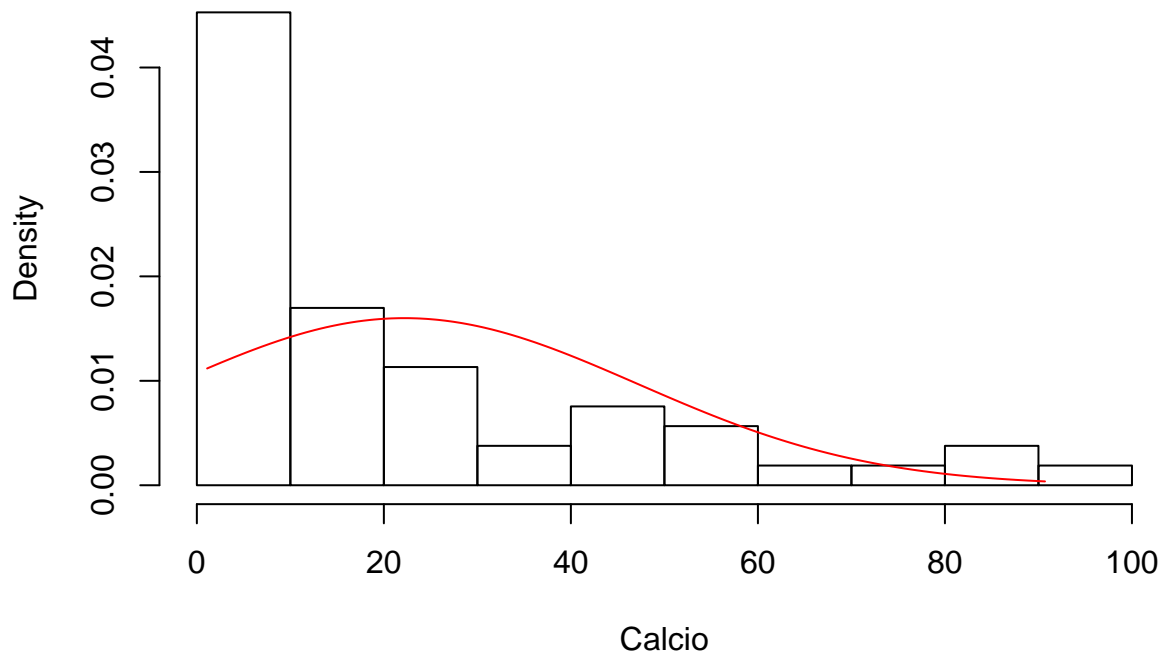
```
## QQplot
qqnorm(calcio, main = "QQplot calcio")
qqline(calcio)
```

QQplot calcio



```
# Histograma  
hist(calcio, prob = TRUE, col = 0, main = "Histograma de calcio", xlab = "Calcio")  
x = seq(min(calcio), max(calcio), 0.1)  
y = dnorm(x, mean(calcio), sd(calcio))  
lines(x, y, col = "red")
```

Histograma de calcio



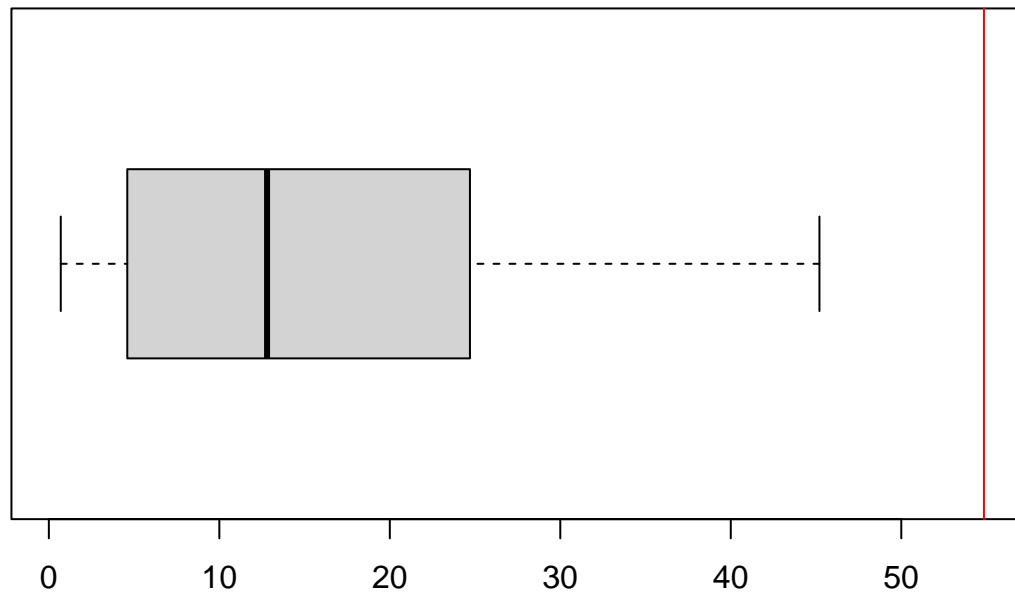
El histograma y el boxplot para el calcio, nos muestran una distribución cargada hacia la izquierda con respecto a la media y a la moda, viendo que se cuenta con un sesgo hacia la derecha y la presencia de datos atípicos y extremos, los cuales son diferentes a las demás observaciones del grupo de calcio. Por otra parte, en QQplot, nos lo confirma mostrando una asimetría positiva.

- X6

```
# Boxplot y valores atípicos
clorofila = mercurio$X6
q1_cl = quantile(clorofila, 0.25)
q3_cl = quantile(clorofila, 0.75)
Rc_cl = q3_cl - q1_cl
y2_cl = q3_cl + 1.5 * Rc_cl

boxplot(clorofila, horizontal=TRUE, ylim=c(0, y2_cl), main= "Boxplot clorofila")
abline(v=y2_cl, col="red") #línea vertical en el límite de los datos atípicos
```

Boxplot clorofila

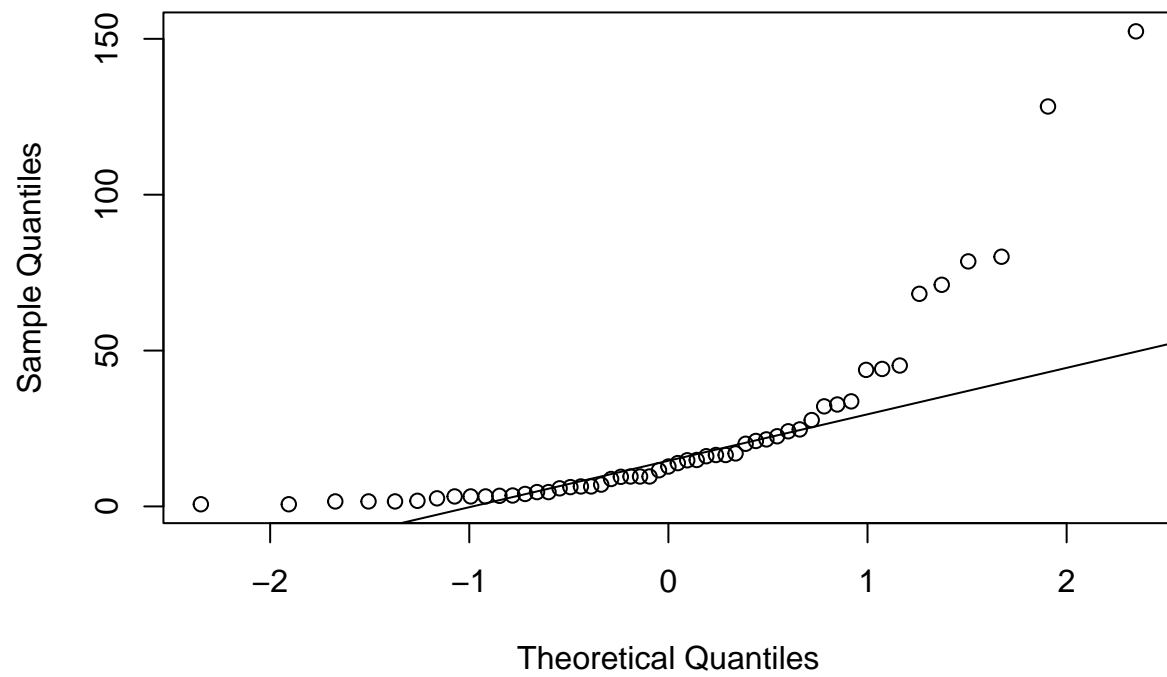


```
X= numericMercurio[clorofila<y2_cl,c("X6")] #Quitar datos atípicos de la matriz db en la variable X  
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##      0.70   3.75   9.60   13.76   20.55   45.20
```

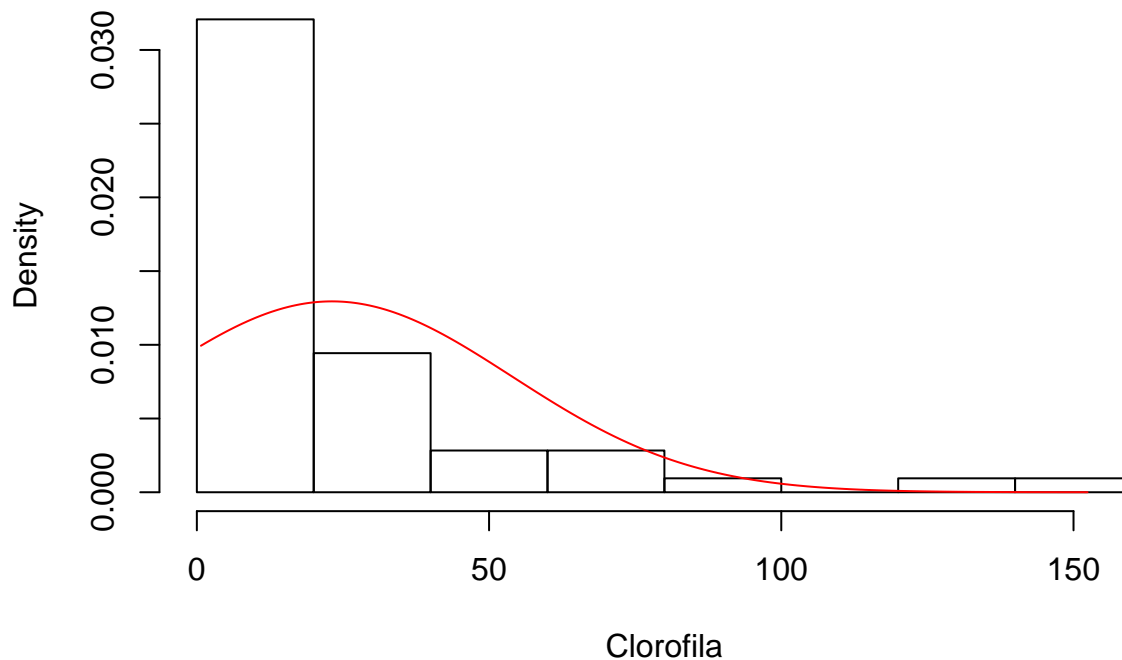
```
## QQplot  
qqnorm(clorofila, main = "QQplot clorofila")  
qqline(clorofila)
```

QQplot clorofila



```
# Histograma  
hist(clorofila, prob = TRUE, col = "blue", main = "Histograma de clorofila", xlab = "Clorofila")  
x = seq(min(clorofila), max(clorofila), 0.1)  
y = dnorm(x, mean(clorofila), sd(clorofila))  
lines(x, y, col = "red")
```

Histograma de clorofila



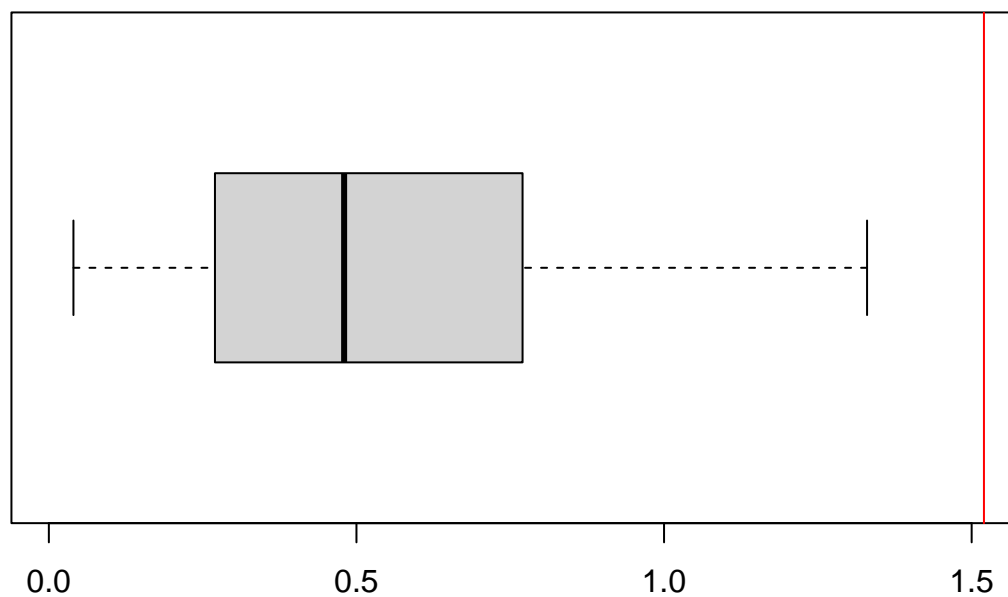
Para la clorofila, el caso es similar. Se nos muestra una distribución cargada hacia la izquierda con respecto a la media y a la moda, viendo que se cuenta con un sesgo hacia la derecha pero no cuenta con la presencia de datos atípicos. Por otra parte, en QQplot, nos lo confirma mostrando una asimetría positiva.

- X7

```
# Boxplot y valores atípicos
concentracion = mercurio$X7
q1_con = quantile(concentracion, 0.25)
q3_con = quantile(concentracion, 0.75)
Rc_con = q3_con - q1_con
y2_con = q3_con + 1.5 * Rc_con

boxplot(concentracion, horizontal=TRUE, ylim=c(0, y2_con), main= "Boxplot concentración")
abline(v=y2_con, col="red") #línea vertical en el límite de los datos atípicos
```

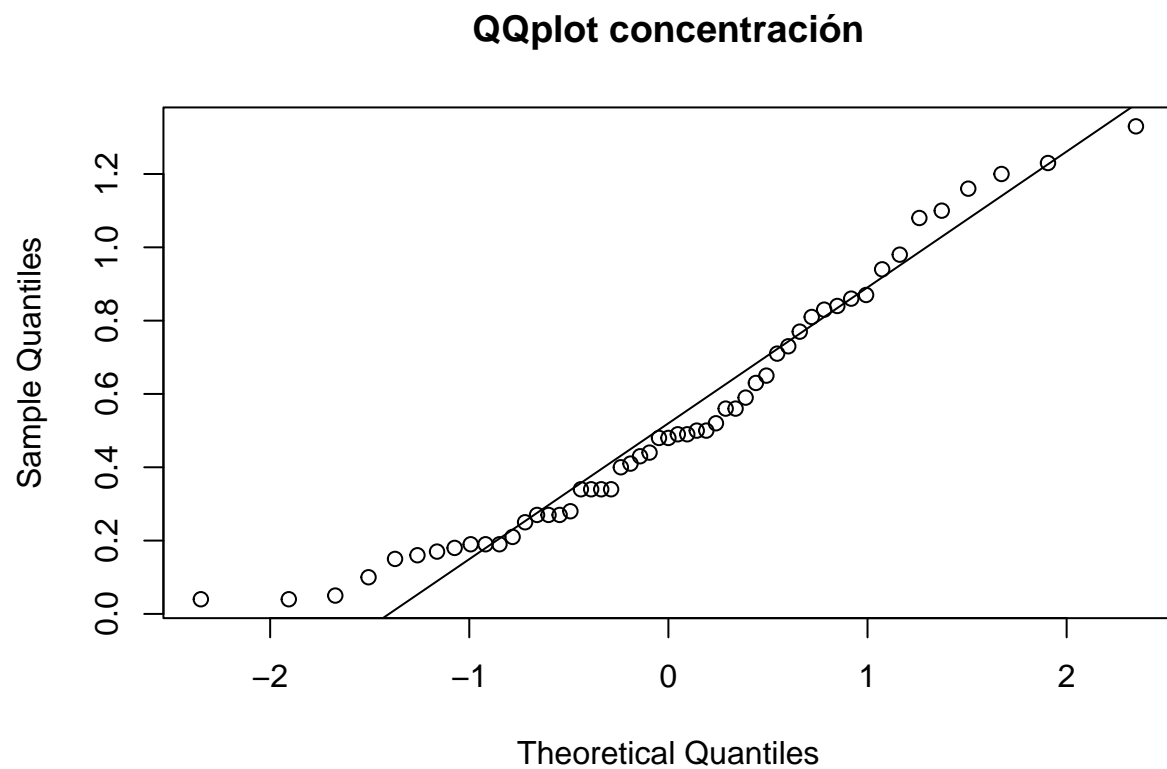
Boxplot concentración



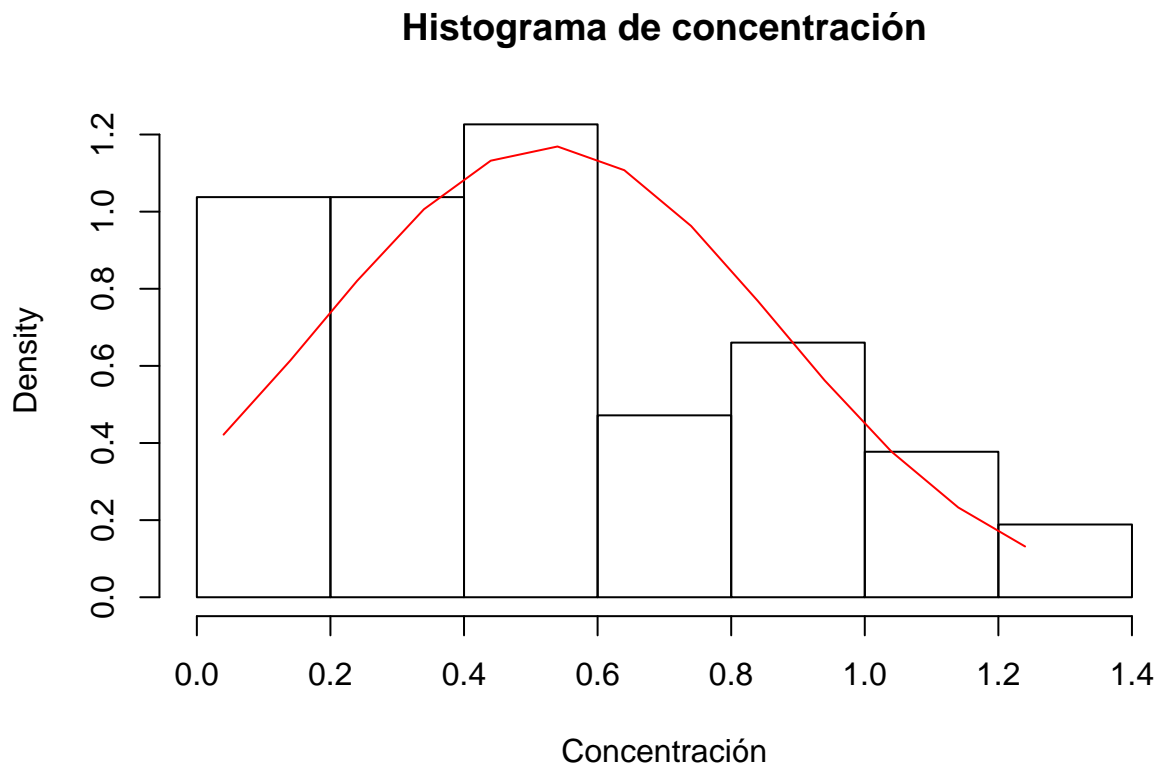
```
X= numericMercurio[concentracion<y2_con,c("X7")] #Quitar datos atípicos de la matriz db en la variable .
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0400  0.2700  0.4800  0.5272  0.7700  1.3300
```

```
## QQplot
qqnorm(concentracion, main = "QQplot concentración")
qqline(concentracion)
```

```
# Histograma
hist(concentracion, prob = TRUE, col = 0, main = "Histograma de concentración", xlab = "Concentración")
x = seq(min(concentracion), max(concentracion), 0.1)
y = dnorm(x, mean(concentracion), sd(concentracion))
lines(x, y, col = "red")
```



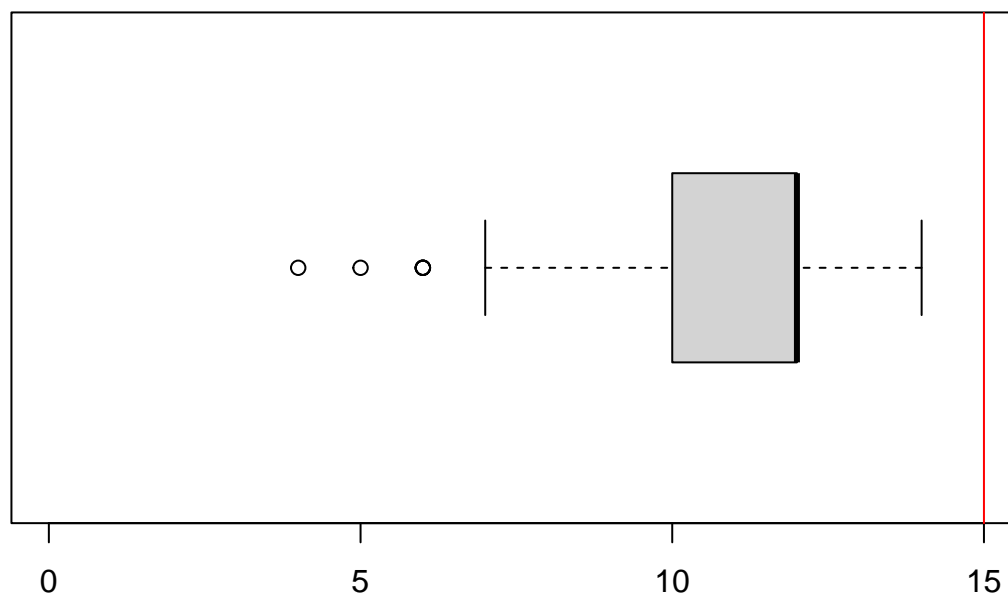
Para la concentración de la media, se muestra igualmente una distribución cargada hacia la izquierda con respecto a la media y a la moda, sesgo hacia la derecha y sin la presencia de datos atípicos. Por otra parte, en QQplot, nos lo confirma mostrando una asimetría positiva.

- X8

```
# Boxplot y valores atípicos
peces = mercurio$X8
q1_p = quantile(peces, 0.25)
q3_p = quantile(peces, 0.75)
Rc_p = q3_p - q1_p
y2_p = q3_p + 1.5 * Rc_p

boxplot(peces, horizontal=TRUE, ylim=c(0, y2_p), main= "Boxplot peces")
abline(v=y2_p, col="red") #línea vertical en el límite de los datos atípicos
```

Boxplot peces

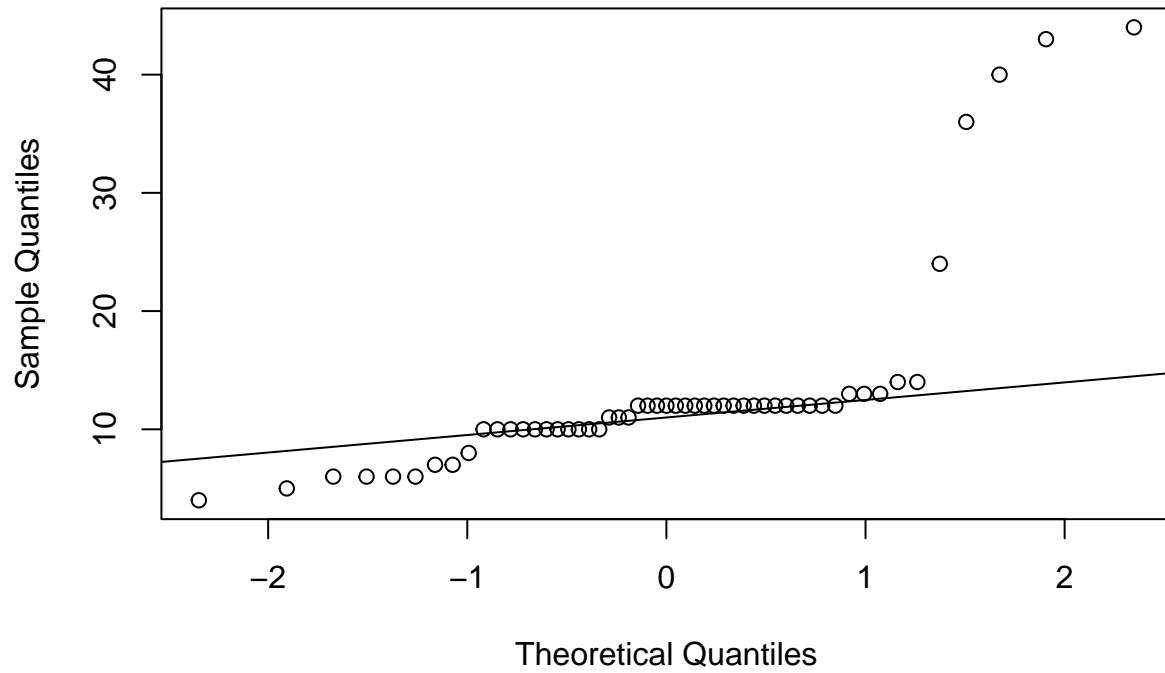


```
X= numericMercurio[peces<y2_p,c("X8")] #Quitar datos atípicos de la matriz db en la variable X
summary(X)
```

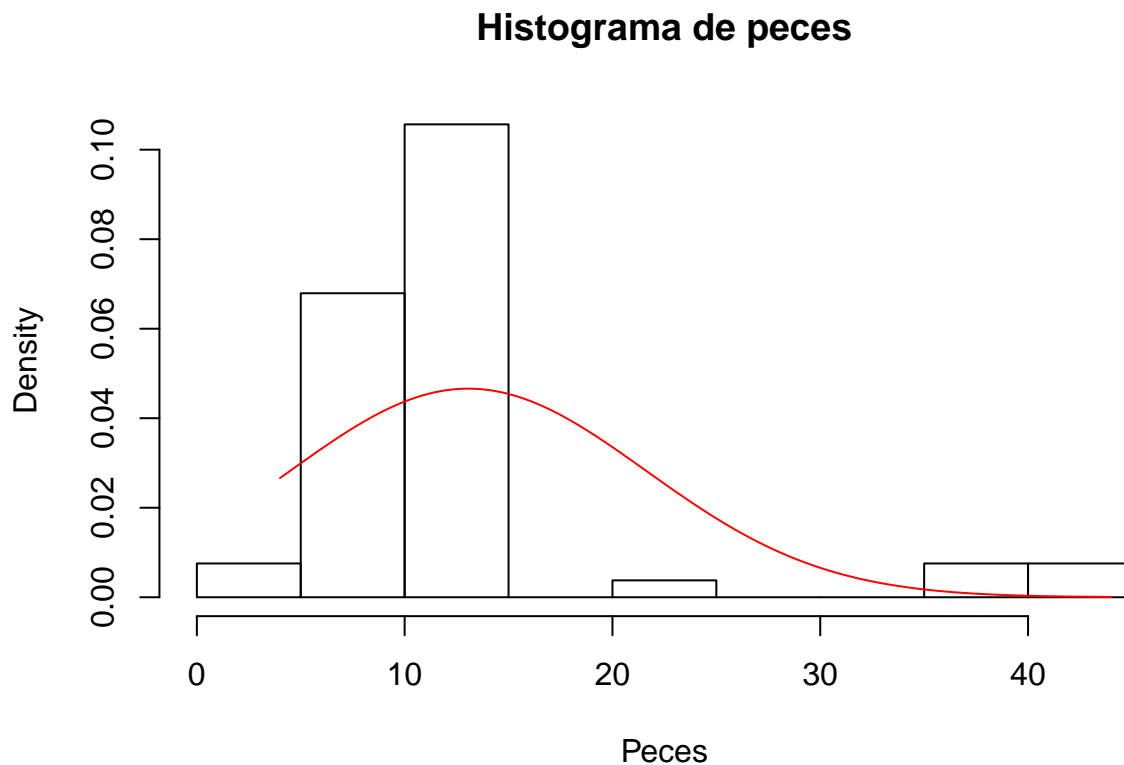
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      4.00  10.00   12.00   10.52  12.00   14.00
```

```
## QQplot
qqnorm(peces, main = "QQplot peces")
qqline(peces)
```

QQplot peces



```
# Histograma
hist(peces, prob = TRUE, col = 0, main = "Histograma de peces", xlab = "Peces")
x = seq(min(peces), max(peces), 0.1)
y = dnorm(x, mean(peces), sd(peces))
lines(x, y, col = "red")
```



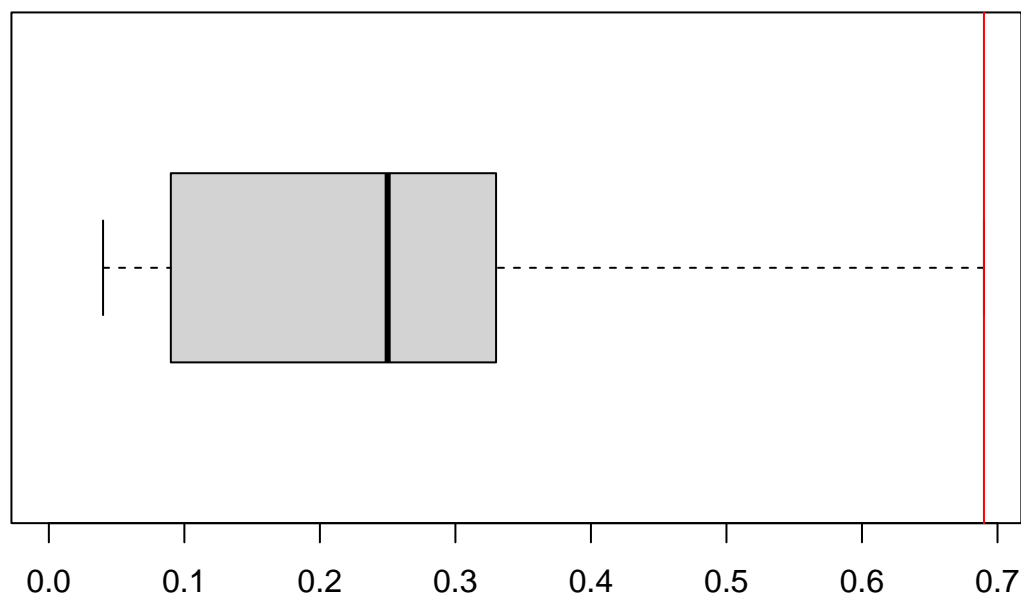
En la cantidad de peces del lago, se ve una casi simetría con respecto a la moda y la media en la distribución, pero, aquí si contamos con la presencia de 3 valores atípicos/extremos. El QQplot nos muestra una distribución con colas gruesas, es decir, baja curtosis y distribución platycúrtica.

- X9

```
# Boxplot y valores atípicos
minimo = mercurio$X9
q1_min = quantile(minimo, 0.25)
q3_min = quantile(minimo, 0.75)
Rc_min = q3_min - q1_min
y2_min = q3_min + 1.5 * Rc_min

boxplot(minimo, horizontal=TRUE, ylim=c(0, y2_min) , main= "Boxplot mínimo de concentración")
abline(v=y2_min, col="red") #línea vertical en el límite de los datos atípicos
```

Boxplot mínimo de concentración

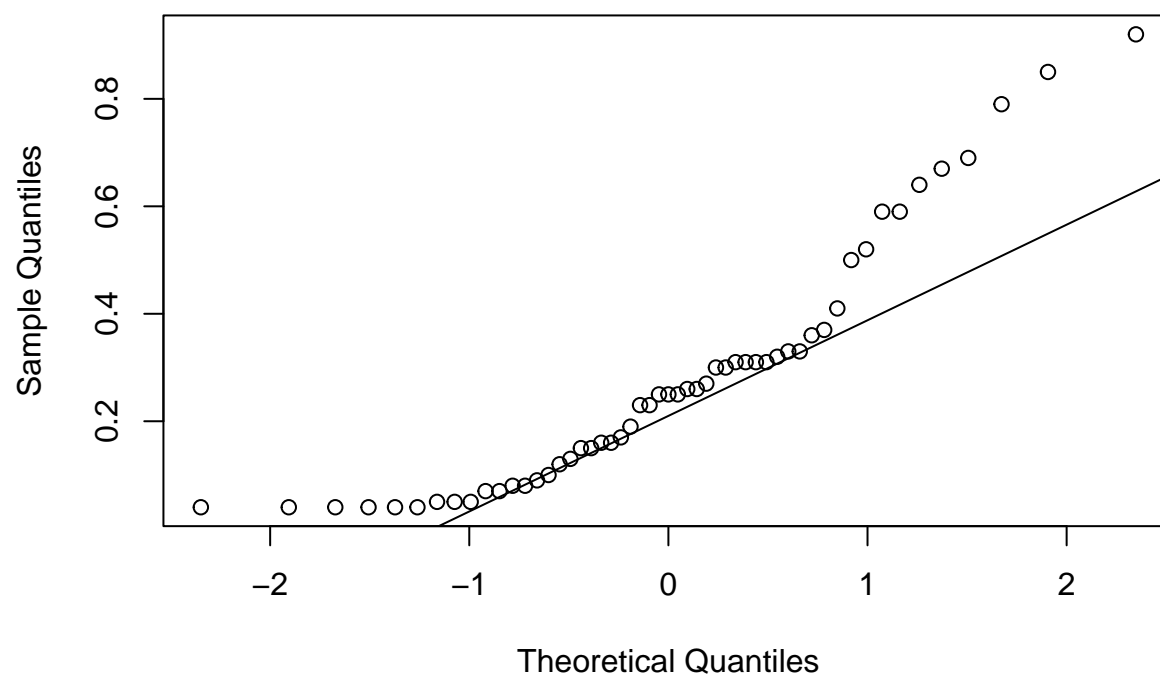


```
X= numericMercurio[minimo<y2_min,c("X9")] #Quitar datos atípicos de la matriz db en la variable X
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0400  0.0825  0.2400  0.2454  0.3175  0.6900
```

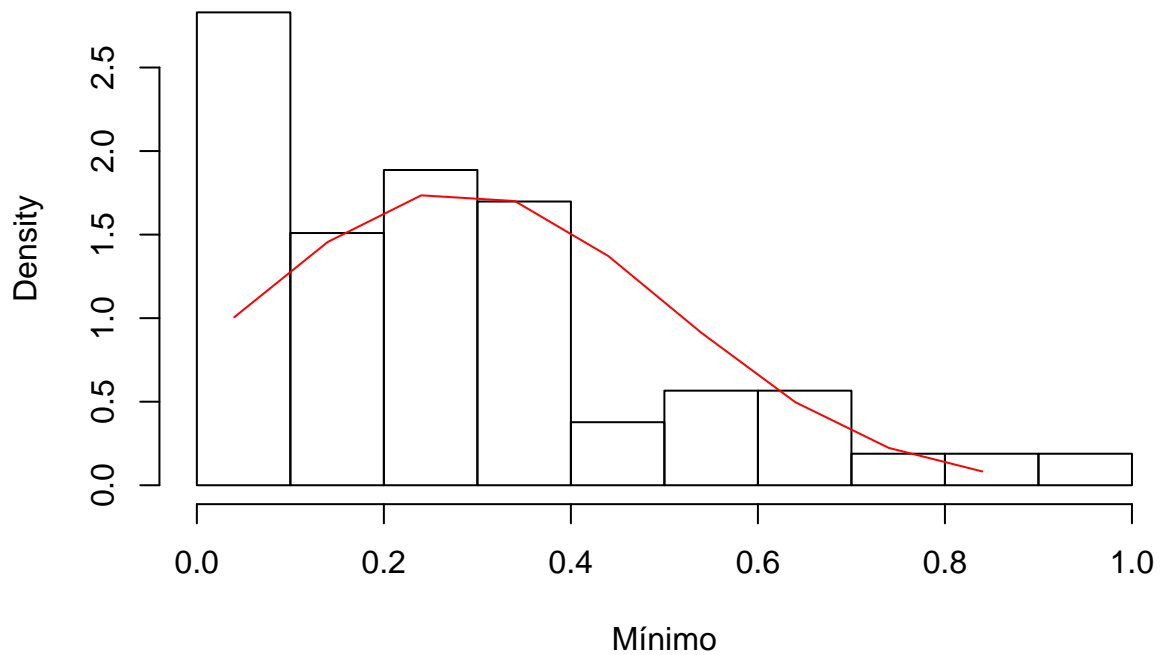
```
## QQplot
qqnorm(minimo, main = "QQplot mínimo de concentración")
qqline(minimo)
```

QQplot mínimo de concentración



```
# Histograma  
hist(minimo, prob = TRUE, col = 0, main = "Histograma de mínimoconcentración", xlab = "Mínimo")  
x = seq(min(minimo), max(minimo), 0.1)  
y = dnorm(x, mean(minimo), sd(minimo))  
lines(x, y, col = "red")
```

Histograma de mínimoconcentración



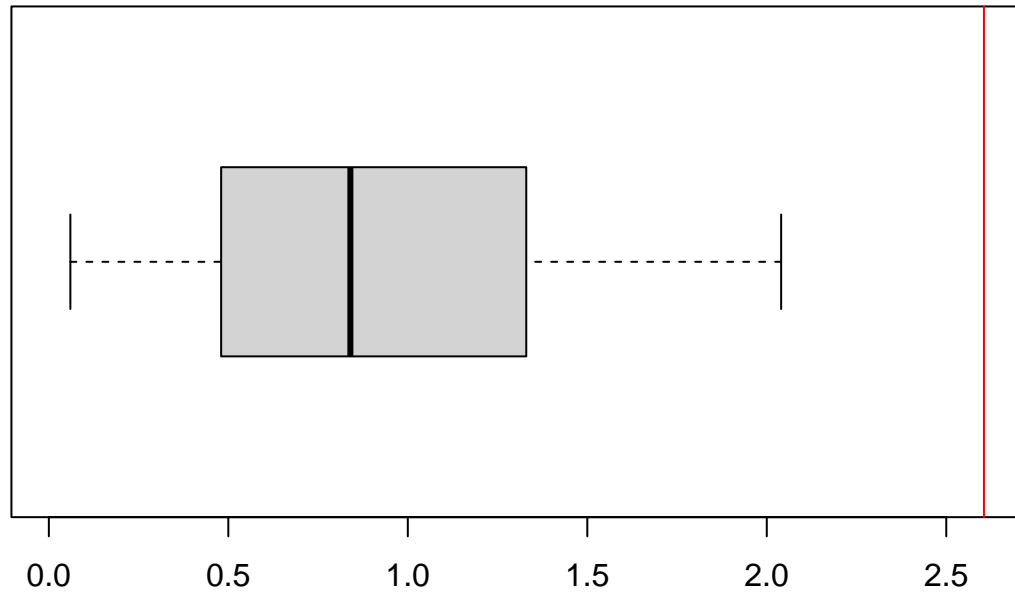
Se muestra un sesgo a la derecha con la distribución recargada hacia la izquierda de acuerdo a la media y mediana. Esta variable, no presenta datos atípicos y el QQplot, nos indica una asimetría positiva.

- X10

```
# Boxplot y valores atípicos
maximo = mercurio$X10
q1max = quantile(maximo, 0.25)
q3max = quantile(maximo, 0.75)
Rcmax = q3max - q1max
y2max = q3max + 1.5 * Rcmax

boxplot(maximo, horizontal=TRUE, ylim=c(0, y2max), main="Boxplot máximo de concentración")
abline(v=y2max, col="red") #línea vertical en el límite de los datos atípicos
```


Boxplot máximo de concentración

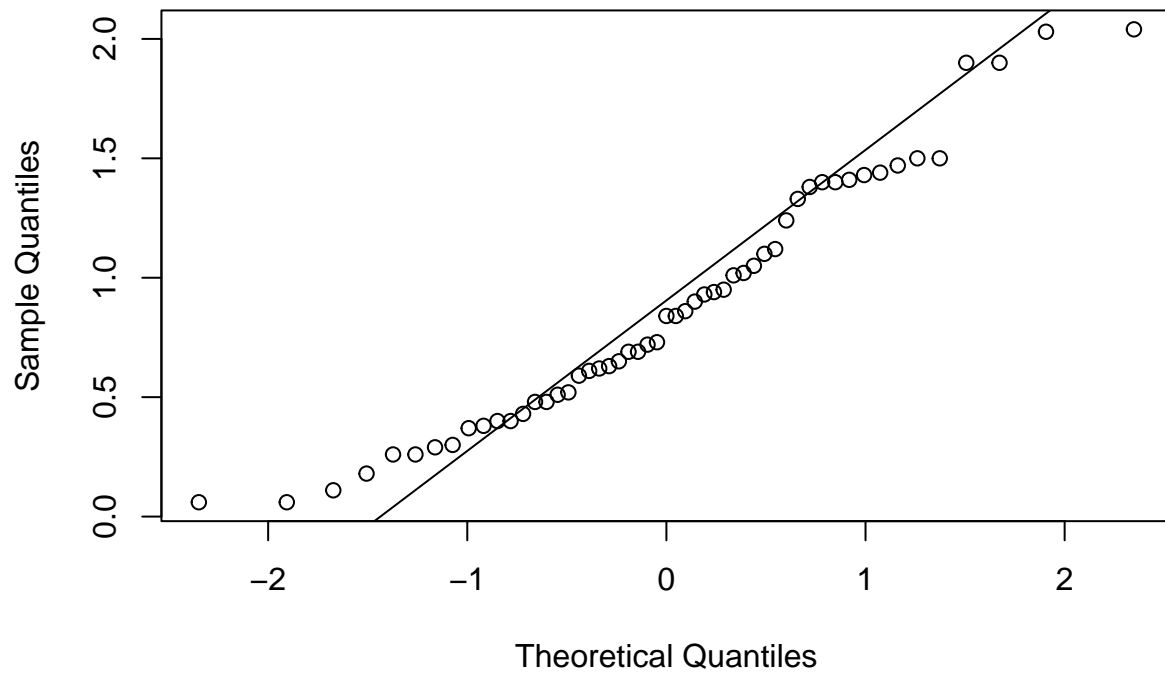


```
X= numericMercurio[maximo<y2max,c("X10")] #Quitar datos atípicos de la matriz db en la variable X
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0600  0.4800  0.8400  0.8745  1.3300  2.0400
```

```
## QQplot
qqnorm(maximo, main = "QQplot máximo de concentración")
qqline(maximo)
```

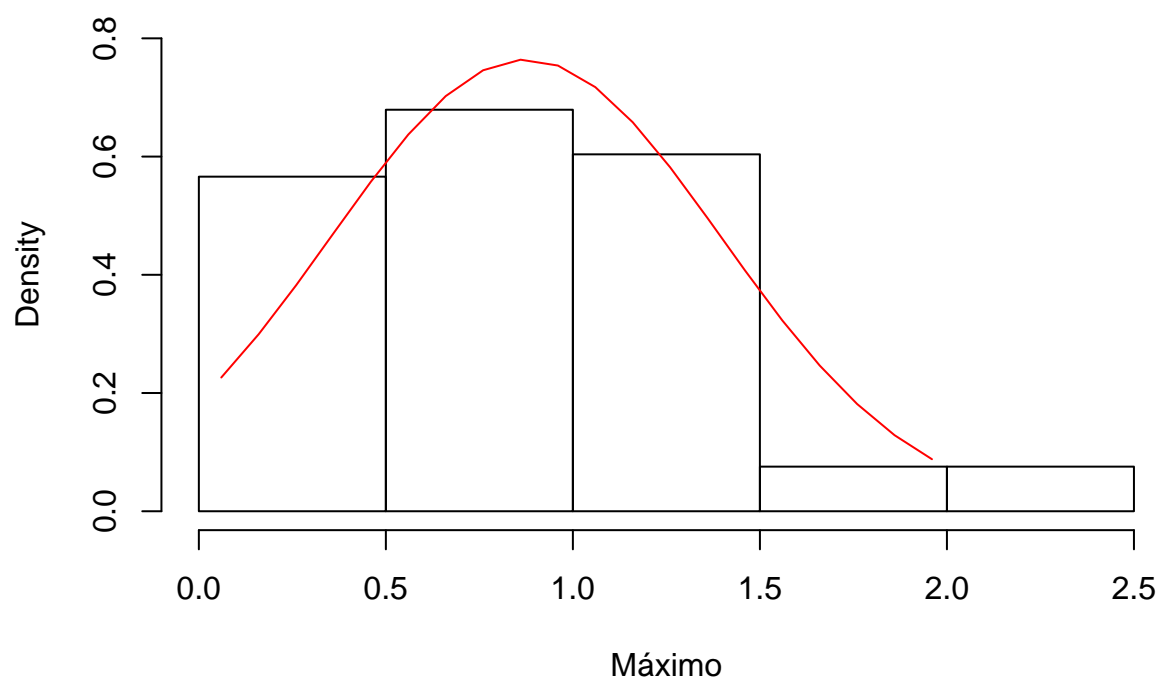
QQplot máximo de concentración



```
# Histograma
```

```
hist(maximo, prob = TRUE, col = 0, main = "Histograma de máximo de concentración", xlab = "Máximo", ylim = 0.05, las = 1)  
x = seq(min(maximo), max(maximo), 0.1)  
y = dnorm(x, mean(maximo), sd(maximo))  
lines(x, y, col = "red")
```

Histograma de máximo de concentración



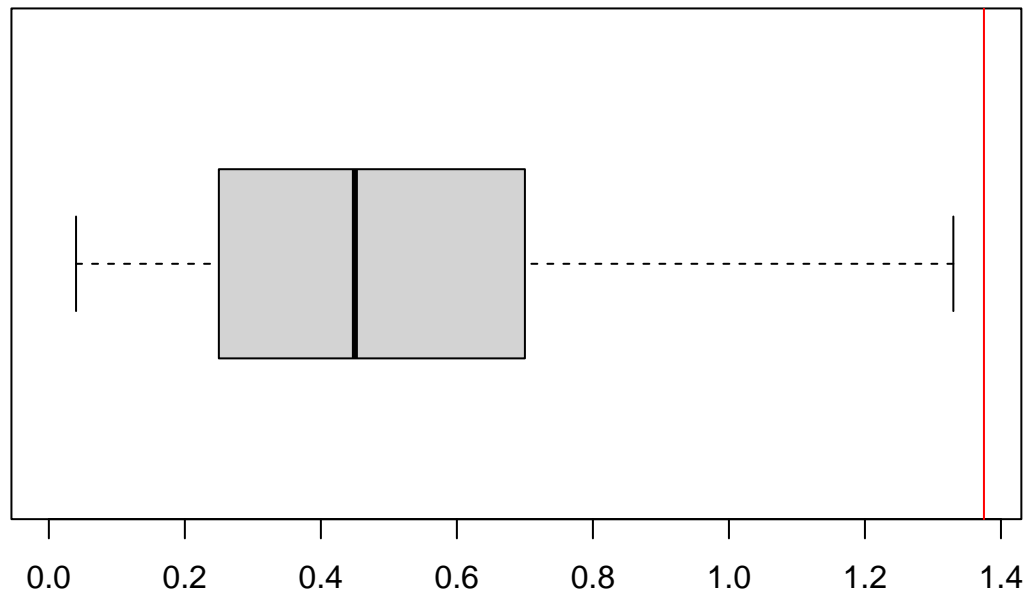
El máximo de la concentración nos muestra una simetría en los datos, ya que están de esta forma distribuidos de acuerdo a la mediana y la moda. Finalmente, el QQplot nos indica una probabilidad normal casi ideal.

- X11

```
# Boxplot y valores atípicos
estimacion = mercurio$X11
q1_est = quantile(estimacion, 0.25)
q3_est = quantile(estimacion, 0.75)
Rc_est = q3_est - q1_est
y2_est = q3_est + 1.5 * Rc_est

boxplot(estimacion, horizontal=TRUE, ylim=c(0, y2_est) , main= "Boxplot estimación")
abline(v=y2_est, col="red") #línea vertical en el límite de los datos atípicos
```

Boxplot estimación

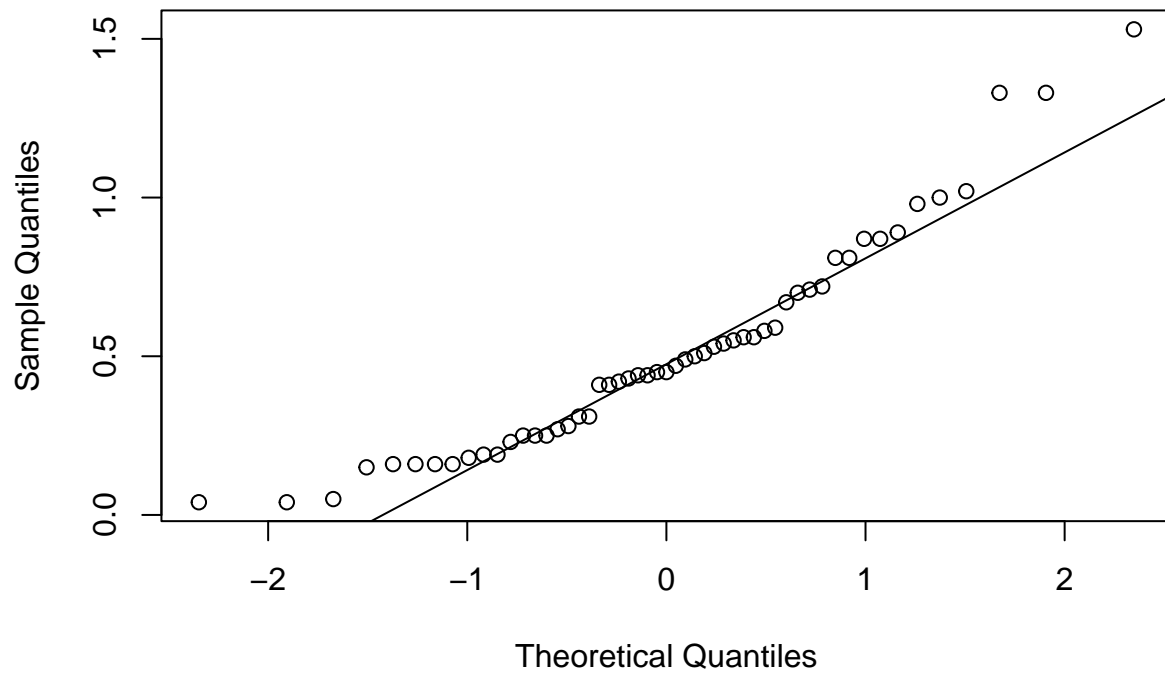


```
X= numericMercurio[estimacion<y2_est,c("X11")] #Quitar datos atípicos de la matriz db en la variable X
summary(X)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0400  0.2500  0.4500  0.4937  0.6775  1.3300
```

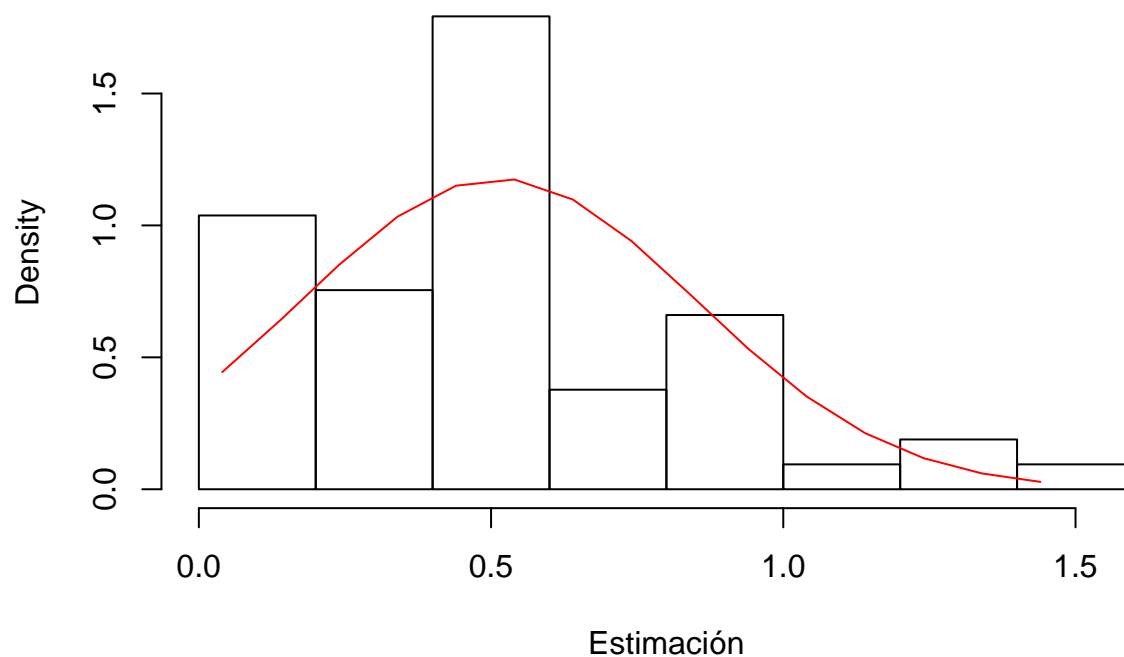
```
## qqplot
qqnorm(estimacion, main = "QQplot estimación")
qqline(estimacion)
```

QQplot estimación



```
# Histograma
hist(estimacion, prob = TRUE, col = 0, main = "Histograma de estimación", xlab = "Estimación")
x = seq(min(estimacion), max(estimacion), 0.1)
y = dnorm(x, mean(estimacion), sd(estimacion))
lines(x, y, col = "red")
```

Histograma de estimación

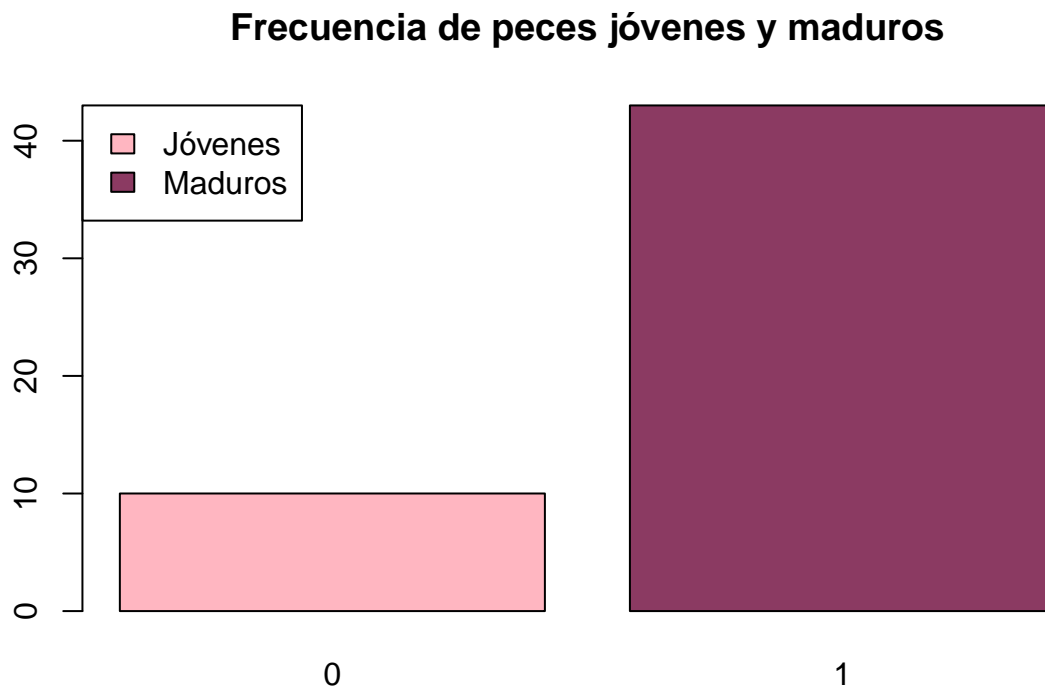


Estimación con sesgo hacia la derecha, distribución recargada hacia la izquierda con respecto a la media y mediana y sin datos atípicos. QQplot con asimetría positiva.

Medidas cualitativas

- X12

```
barplot(table(mercurio$X12),col=c("lightpink","hotpink4"), main = "Frecuencia de peces jóvenes y maduros")
```



Aquí notamos que son más la cantidad de peces maduros que jóvenes, lo que puede indicar un problema en que el mercurio, esta afectando más a los jóvenes.

- Medición de lagos que superaron los 0.5 mg de hg/kg

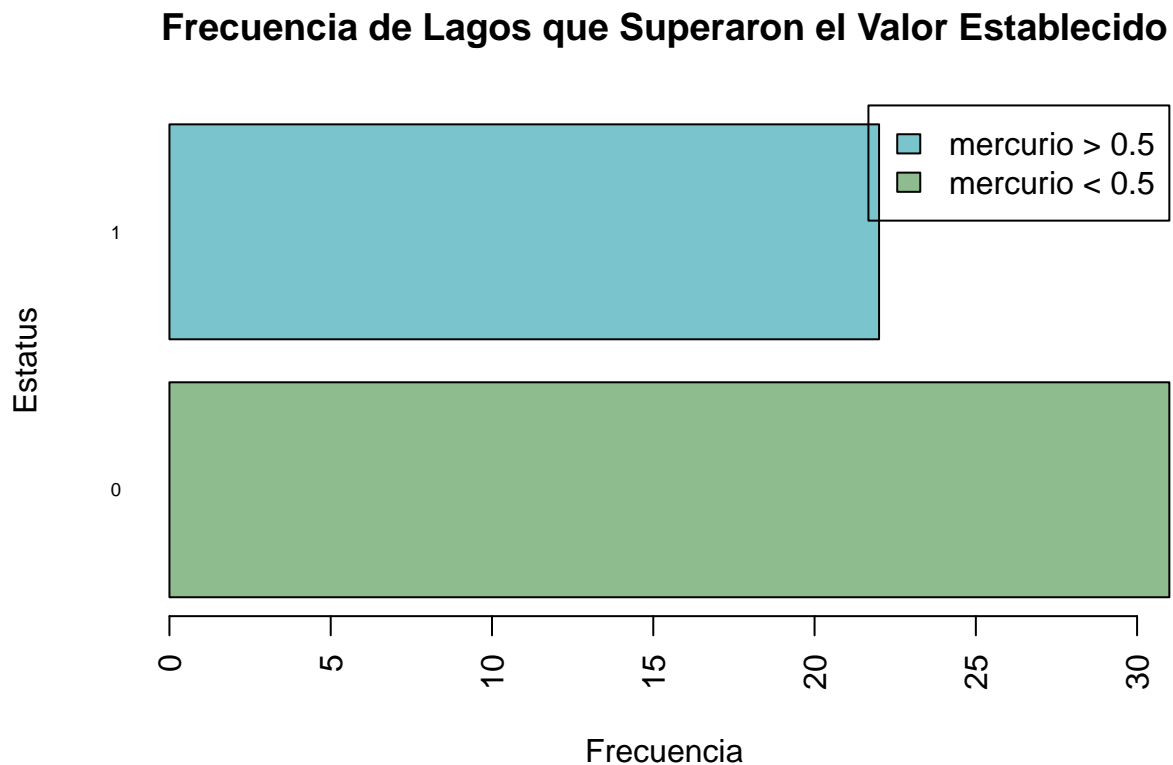
```
mediaCnt = numericMercurio
mediaCnt$X13 <- with(mediaCnt, ifelse(concentracion > 0.5, 1, 0))
tableMediaCnt = table(mediaCnt$X13)
print("Tabla de Distribución de Lagos que Superaron los 0.5 mg de Hg/Kg: ")
```

```
## [1] "Tabla de Distribución de Lagos que Superaron los 0.5 mg de Hg/Kg: "
```

```
tableMediaCnt
```

```
##
##  0  1
## 31 22
```

```
# Gráfica de Frecuencia
barplot(tableMediaCnt, width = 1, cex.names = 0.6, col = c("darkseagreen", "cadetblue3"), main = "Frecu
```



Con esta gráfica, visualizamos la cantidad de datos que superan la normativa de referencia de niveles máximos de mercurio, siendo más de la mitad los que no lo logran.

- División y frecuencia de ph

```
divPh = numericMercurio
divPh$X4 <- with(divPh, ifelse(ph < 7.0, "Ácido", ifelse(ph == 7.0, "Neutro", "Alcalino")))
tableDivPh = table(divPh$X4)
print("Tabla de Distribución del PH: ")
```

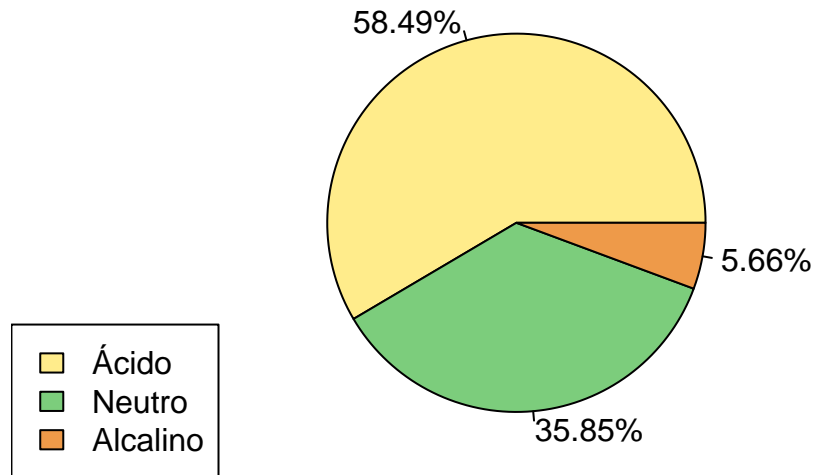
```
## [1] "Tabla de Distribución del PH: "
```

```
tableDivPh
```

```
##
##      Ácido Alcalino  Neutro
##       31       19       3
```

```
#Gráfica de pie
colors <- c("lightgoldenrod1", "palegreen3", "tan2")
pie(tableDivPh, col = colors, main = "Gráfica del PH", labels = paste0(round(100 * tableDivPh / sum(tableDivPh), 1), "%"),
legend("bottomleft", c("Ácido", "Neutro", "Alcalino"), fill=colors)
```


Gráfica del PH



En esta gráfica de pastel, vemos que dominan más las variables con ph ácido, seguido del neutro y finalmente alcalino, lo cual muestra el gran afectamiento hacia los peces ya que les puede provocar el no funcionamiento de algunas células gracias a la producción prolongada de ácido.

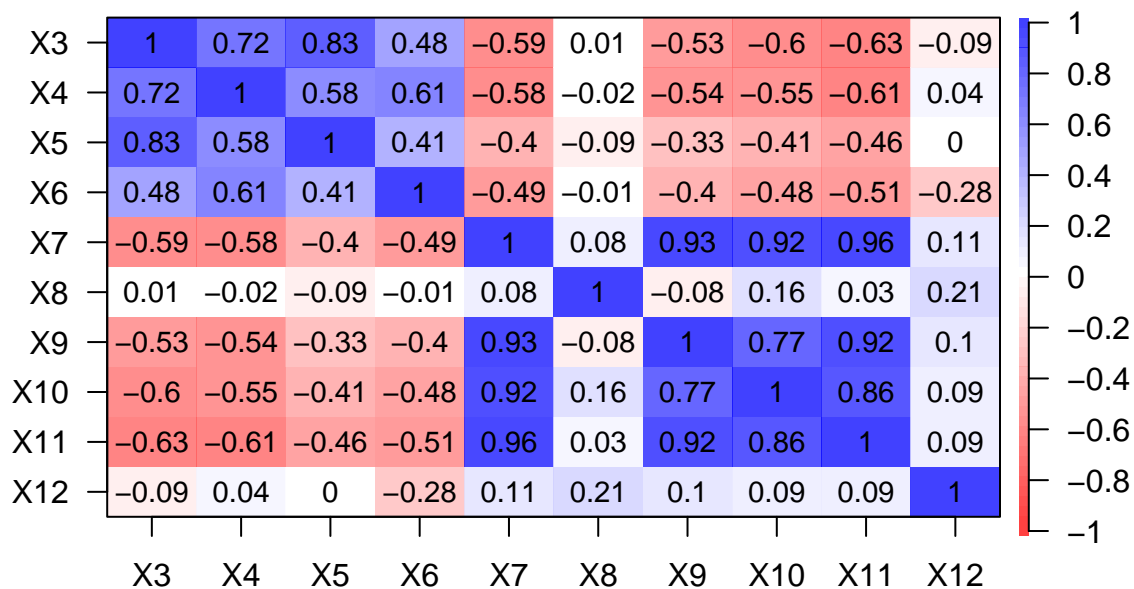
Correlaciones

Matriz de correlacion

```
library(psych)
```

```
## Registered S3 method overwritten by 'psych':  
##   method      from  
##   plot.residuals rmutl
```

```
corPlot(cor(numericMercurio))
```



Debido a la matriz de correlación anteriormente desplegada, logramos ver que hay una mayor interacción y dependencia entre las variables:

- $X7$ y $X9 \rightarrow 0.93$
- $X7$ y $X10 \rightarrow 0.92$
- $X7$ y $X11 \rightarrow 0.96$ Lo que da a indicar que la concentración media de mercurio en el tejido muscular del grupo de peces estudiados, tienen más relación con la estimación, el mínimo y el máximo de peces estudiados.

Sin embargo, escoger esas variables puede ser perjudicial para la implementación de nuestros modelos, ya que se puede dar un problema de multicolinealidad y dificultar los resultados de interpretación, es por ello que escogeremos a las que indican una correlación debilmente negativa, es decir **X3, X4, X5, X6**

Implementación de modelos estadísticos

ANOVA

Mediante el método ANOVA, buscamos resolver la pregunta: *¿Habrá diferencia significativa entre la concentración de mercurio por la edad de los peces?*

Para ello, utilizaremos las variables:

- **X7** -> Concentración media de mercurio
- **X12** -> Indicador de la edad de los peces (jóvenes o maduros)

Es necesario para poder empezar, hacer un análisis sobre cada valor de concentración para saber si pertenece a un pez joven o maduro y de esta forma conocer cuantos datos corresponden a cada uno. Para este problema, utilizaremos un nivel de significancia de $\alpha = 0.05$

```
mediaJ = numericMercurio[numericMercurio$X12 == 0,]$X7
mediaM = numericMercurio[numericMercurio$X12 == 1,]$X7
print("Jovenes")
```

```
## [1] "Jovenes"
```

```
mediaJ
```

```
## [1] 1.33 0.04 0.44 0.05 0.41 0.50 0.87 0.56 0.04 0.27
```

```
print("Maduros")
```

```
## [1] "Maduros"
```

```
mediaM
```

```
## [1] 1.23 1.20 0.27 0.48 0.19 0.83 0.81 0.71 0.50 0.49 1.16 0.15 0.19 0.77 1.08
## [16] 0.98 0.63 0.56 0.73 0.34 0.59 0.34 0.84 0.34 0.28 0.34 0.17 0.18 0.19 0.49
## [31] 1.10 0.16 0.10 0.48 0.21 0.86 0.52 0.65 0.94 0.40 0.43 0.25 0.27
```

```
print("Media por edad")
```

```
## [1] "Media por edad"
```

```
mediaEdad = c(mediaJ, mediaM)
mediaEdad
```

```
## [1] 1.33 0.04 0.44 0.05 0.41 0.50 0.87 0.56 0.04 0.27 1.23 1.20 0.27 0.48 0.19
## [16] 0.83 0.81 0.71 0.50 0.49 1.16 0.15 0.19 0.77 1.08 0.98 0.63 0.56 0.73 0.34
## [31] 0.59 0.34 0.84 0.34 0.28 0.34 0.17 0.18 0.19 0.49 1.10 0.16 0.10 0.48 0.21
## [46] 0.86 0.52 0.65 0.94 0.40 0.43 0.25 0.27
```

Por consiguiente, creamos ahora una variable llamada *edad* para poder tener nuestras clasificaciones correspondientes a ella y pasamos a la implementación del método ANOVA a través de la función *aov*

```
edad = c(rep("J", length(mediaJ)), rep("M", length(mediaM)))
edad = factor(edad)
```

```
A<-aov(mediaEdad~edad)
summary(A)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## edad       1  0.072  0.07151    0.61  0.438
## Residuals 51  5.976  0.11718
```

Mediante los resultados de nuestro ANOVA, verificamos que la diferencia entre medias **no es estadísticamente significativa** es decir, **no hay diferencia** entre estos valores, lo que nos da por conclusión que la relación entre la edad y la concentración no es suficiente evidencia para conocer como se está comportando y afectando el problema del mercurio en los lagos.

Debido a lo anterior, nos concentramos en la siguiente pregunta: **¿Hay evidencia para suponer que la concentración promedio de mercurio en los lagos es dañino para la salud humana?**

Para esto consideraremos nuestra variable cualitativa que examinaba cuáles eran los mayores a 0.5 y los menores, esto de acuerdo a la normatividad de referencia para evaluar los niveles máximos de Hg, por lo que como en el ejemplo anterior, dividiremos las variables y examinaremos cuantas corresponden a cada una.

```
menor = numericMercurio[mediaCnt$X13 == 0,]$X7
mayor = numericMercurio[mediaCnt$X13 == 1,]$X7

print("Menor")
```

```
## [1] "Menor"
```

```
menor
```

```
## [1] 0.04 0.44 0.27 0.48 0.19 0.50 0.49 0.05 0.15 0.19 0.41 0.34 0.34 0.50 0.34
## [16] 0.28 0.34 0.17 0.18 0.19 0.04 0.49 0.16 0.10 0.48 0.21 0.27 0.40 0.43 0.25
## [31] 0.27
```

```
print("Mayor")
```

```
## [1] "Mayor"
```

```
mayor
```

```
## [1] 1.23 1.33 1.20 0.83 0.81 0.71 1.16 0.77 1.08 0.98 0.63 0.56 0.73 0.59 0.84
## [16] 0.87 0.56 1.10 0.86 0.52 0.65 0.94
```

```
print("Reglamento")
```

```
## [1] "Reglamento"
```

```
mediaR = c(menor, mayor)
mediaR
```

```
## [1] 0.04 0.44 0.27 0.48 0.19 0.50 0.49 0.05 0.15 0.19 0.41 0.34 0.34 0.50 0.34
## [16] 0.28 0.34 0.17 0.18 0.19 0.04 0.49 0.16 0.10 0.48 0.21 0.27 0.40 0.43 0.25
## [31] 0.27 1.23 1.33 1.20 0.83 0.81 0.71 1.16 0.77 1.08 0.98 0.63 0.56 0.73 0.59
## [46] 0.84 0.87 0.56 1.10 0.86 0.52 0.65 0.94
```

Posteriormente pasamos a la implementación del ANOVA, realizando una variable nueva “reglamento”. De igual manera, se utilizará un nivel de significancia $\alpha = 0.05$

```
reglamento = c(rep("Menor", length(menor)), rep("Mayor", length(mayor)))
reglamento = factor(reglamento)
```

```
A<-aov(mediaR~reglamento)
summary(A)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## reglamento   1  4.201    4.201    116 9.68e-15 ***
## Residuals   51  1.847    0.036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A diferencia del anterior, logramos ver que en esta ocasión $p < \alpha$, mostrándonos que las diferencias entre algunas de las medias son estadísticamente significativas, es decir, no todas las medias de la población son iguales.

A través de este resultado, podemos visualizar que se pierde un grado de libertad, al contar con 52 en total siendo $n = 53$ y la variación entre tratamientos nos da un valor de 116, llegando a visualizar que como se mencionó anteriormente, ninguna de las concentraciones tiene un efecto igual a 0.

Análisis de cada reglamento por concentración

```
m = tapply(mediaR, reglamento, mean)
s = tapply(mediaR, reglamento, sd)
n = tapply(mediaR, reglamento, length)
sprintf("Medias del reglamento: ")
```

```
## [1] "Medias del reglamento: "
```

```
m
```

```
##      Mayor      Menor
## 0.8613636 0.2900000
```

```
sprintf("Desviación estándar del reglamento: ")
```

```
## [1] "Desviación estándar del reglamento: "
```

```
s
```

```
##      Mayor      Menor
## 0.2397478 0.1460593
```

```
sprintf("Tamaño de la muestra del reglamento: ")
```

```
## [1] "Tamaño de la muestra del reglamento: "
```

```
n
```

```
## Mayor Menor
##    22    31
```

Invertalos de confianza

```
sm = s/sqrt(n)
E=abs(qt(0.025,n-1))*sm
In=m-E
In
```

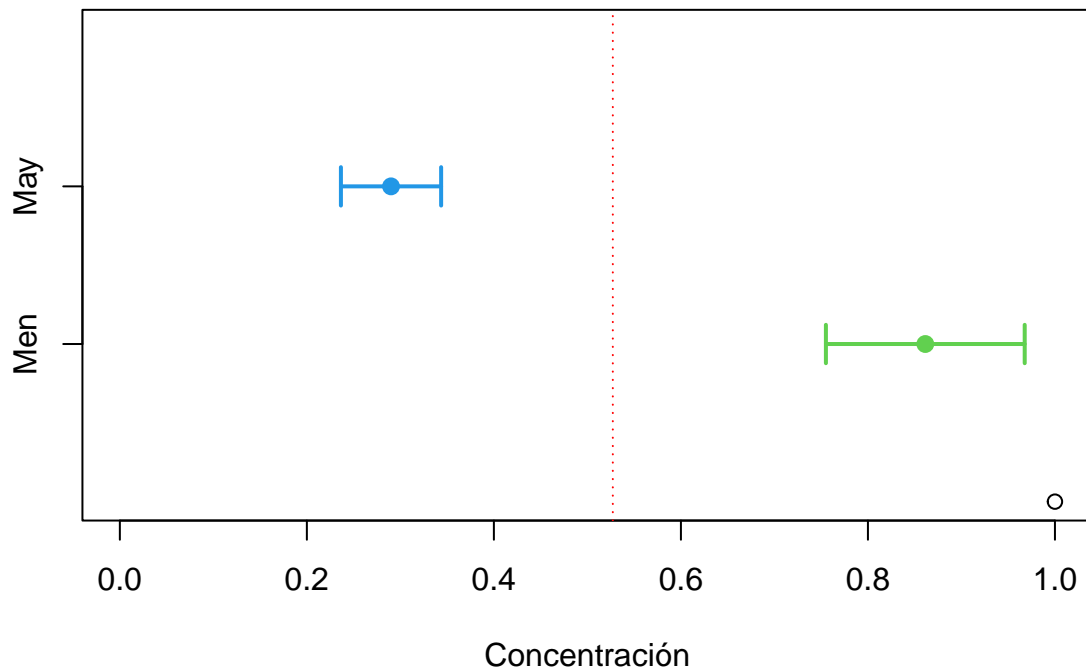
```
##      Mayor      Menor
## 0.7550654 0.2364250
```

```
Sup=m+E
Sup
```

```
##      Mayor      Menor
## 0.9676619 0.3435750
```

```
plot(0,ylim=c(0,3),xlim=c(0,1), yaxt="n", ylab="",xlab="Concentración",main="Normativas de referencia")
axis(2,at=c(1:2),labels=c("Men","May"))
for(i in 1:2){
arrows(In[i],i,Sup[i],i, angle=90, code=3, length = 0.1, lwd = 2,col=i+2)
points(m[i], i, pch=19, cex=1.1,col=i+2)}
abline(v=mean(mediaR),lty=3,col="red")
```

Normativas de referencia

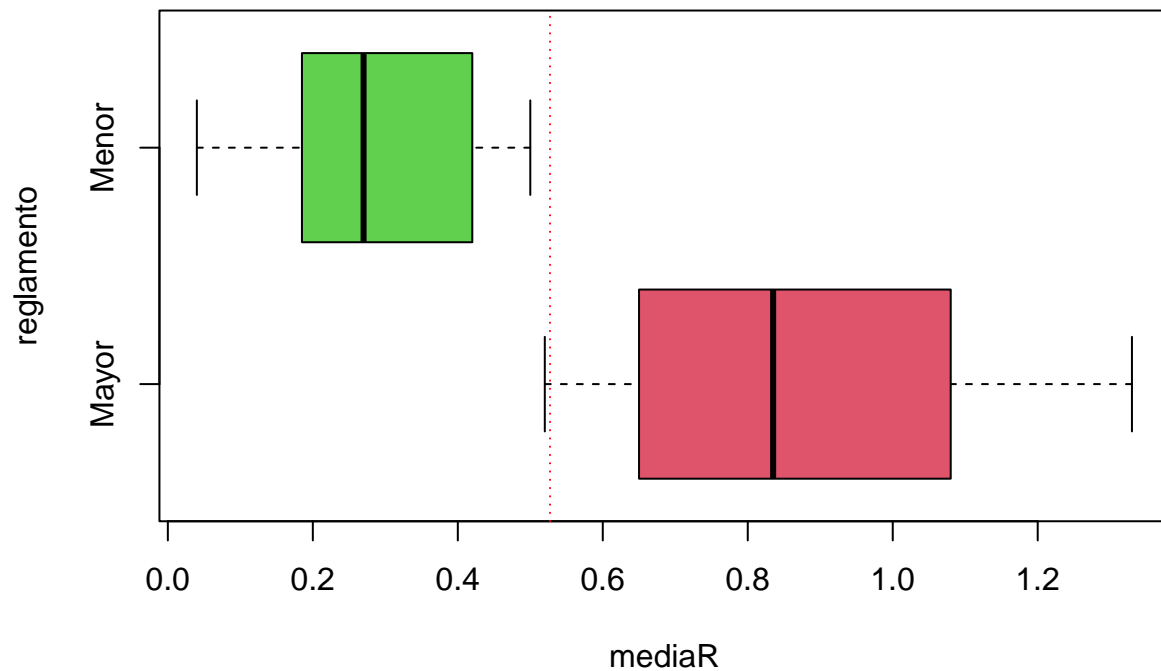


Esta gráfica nos muestra visualmente los intervalos de confianza siendo:

- **Menor** de 0.23 a 0.34

- Mayor de 0.75 a 0.96

```
boxplot(mediaR ~ reglamento, col = 2:5, horizontal = TRUE)
abline(v = mean(mediaR), lty = 3, col = "red")
```



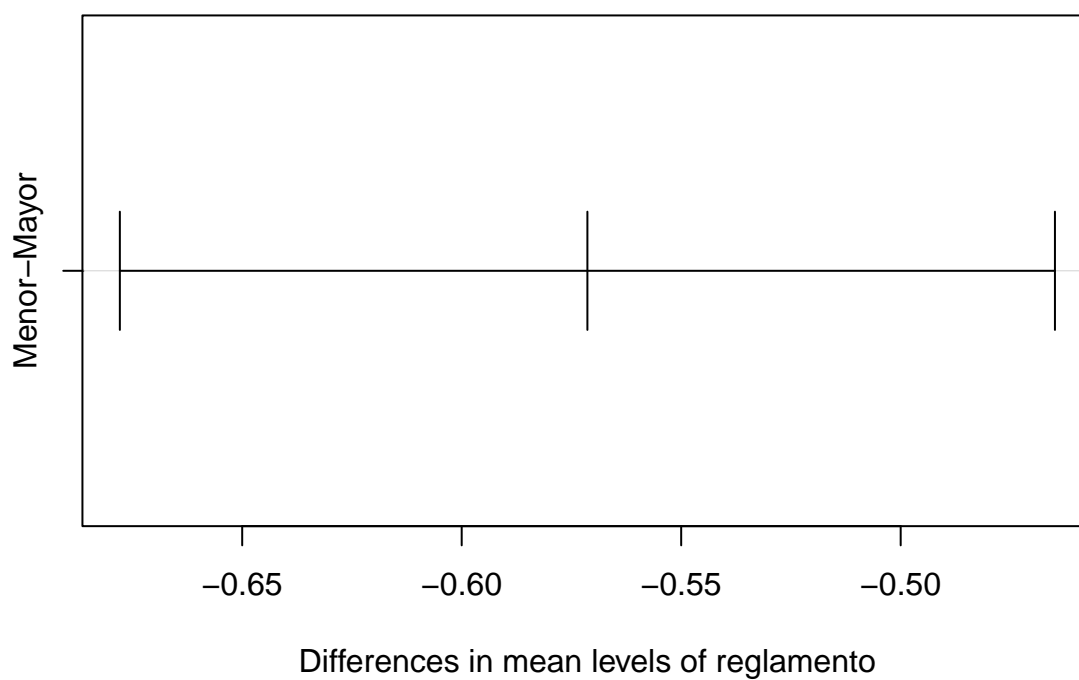
Los datos menores tienen una distribución concentrada hacia la derecha con respecto a su media, mientras que la mayor la tiene hacia la izquierda.

```
Tu=TukeyHSD(A)
Tu
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = mediaR ~ reglamento)
##
## $reglamento
##          diff          lwr          upr p adj
## Menor-Mayor -0.5713636 -0.6778698 -0.4648575 0
```

```
plot(TukeyHSD(A))
```

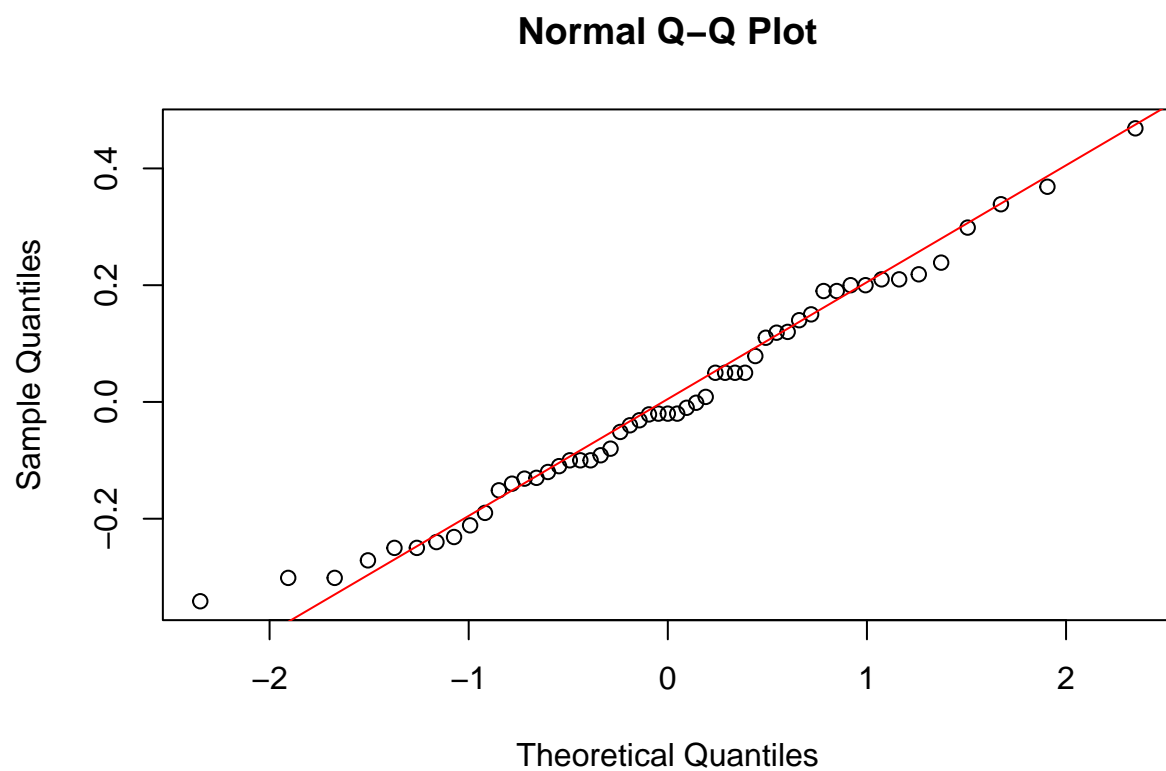
95% family-wise confidence level



Como en esta ocasion solo nos basamos un reglamento de “Mayor” y “Menor”, no hay comparación significativa con otros pares de variables.

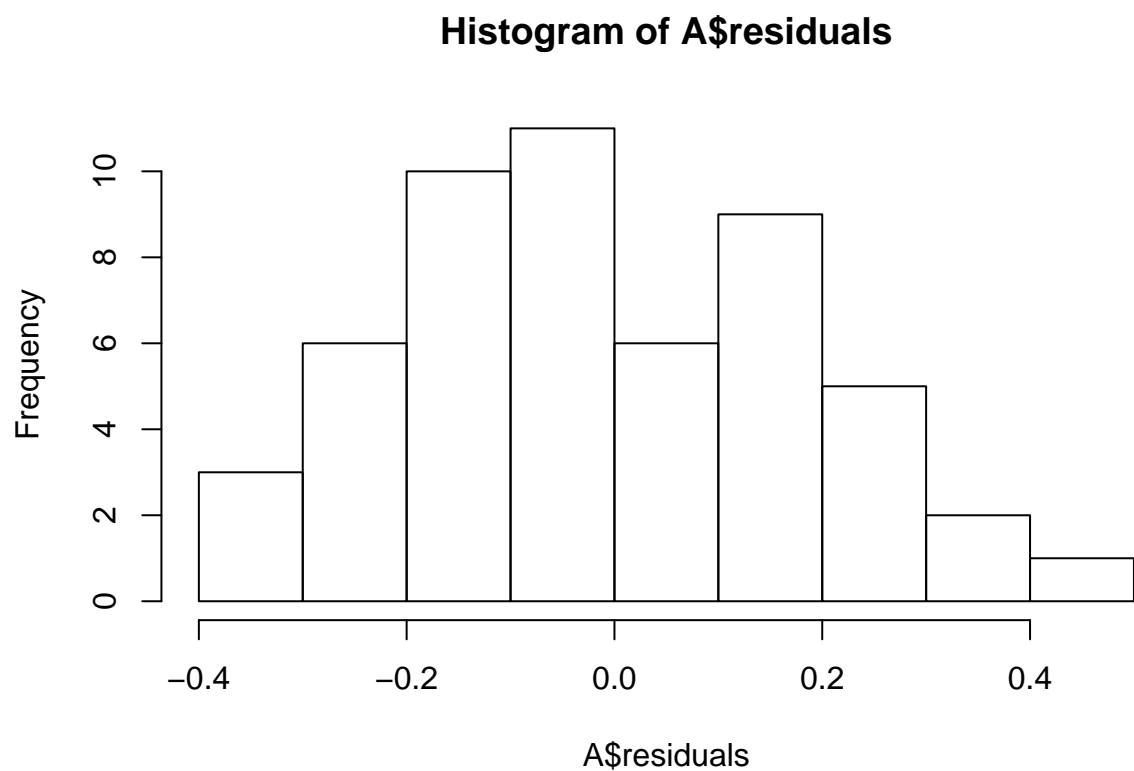
Verificación de supuestos

```
qqnorm(A$residuals)
qqline(A$residuals, col="red")
```

Normalidad

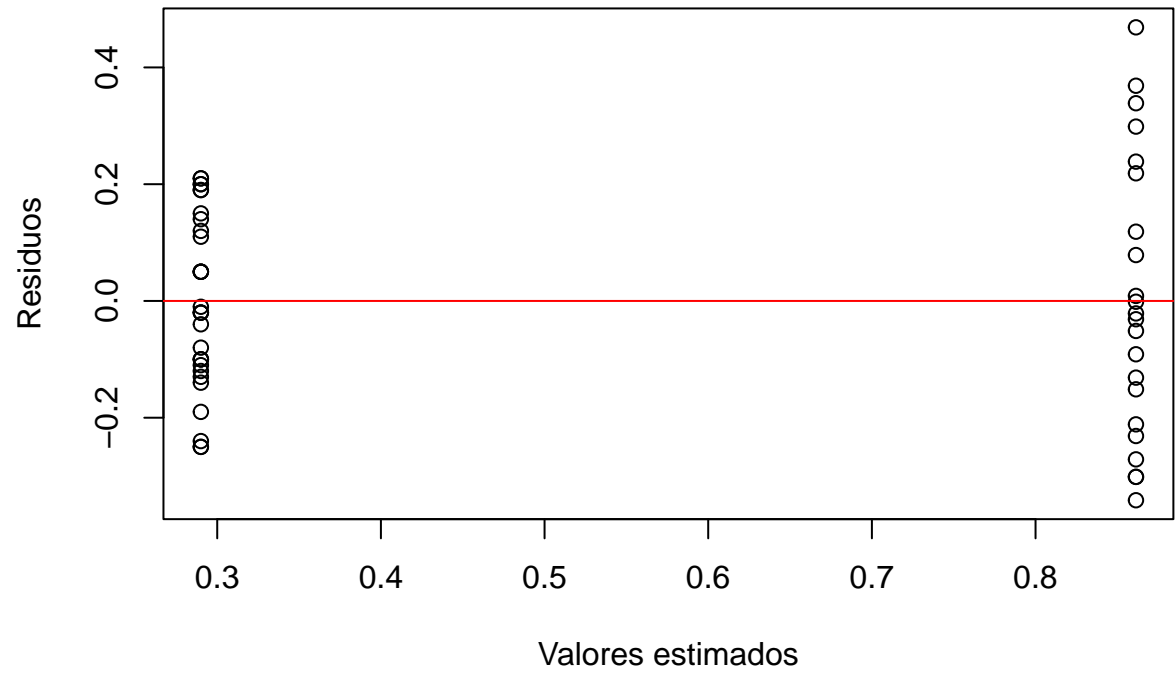
```
hist(A$residuals, col=0)
```



Mediante la gráfica podemos visualizar que la probabilidad Normal de nuestro análisis es ideal en su simetría.

```
plot(A$fitted.values,A$residuals,ylab="Residuos",xlab = "Valores estimados", main="Valores estimados vs  
abline(h=0, col="red")
```

Valores estimados vs residuos



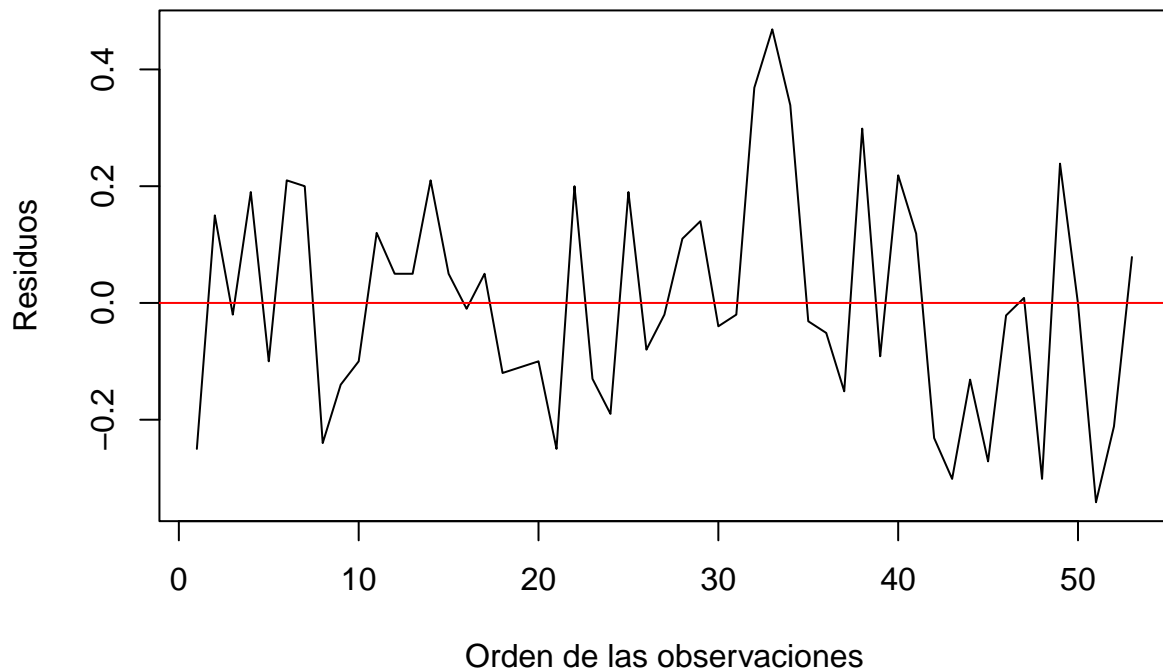
Homocedasticidad

Gracias a esta gráfica, vemos que los residuos son constantes y cumplen con los supuestos.

####Independencia

```
plot(c(1:53),A$residuals, main="Errores vs orden de observación", type="l", xlab="Orden de las observaciones",  
abline(h=0, col="red"))
```

Errores vs orden de observación



En la independencia. vemos que nuestra autocorrelación en los errores son negativos, es decir, se observa una alternancia muy marcada de residuos positivos y negativos.

Regresión lineal múltiple

```
medidas = numericMercurio[1:5]
medidas
```

```
##      X3  X4  X5   X6  X7
## 1    5.9 6.1  3.0  0.7 1.23
## 2    3.5 5.1  1.9  3.2 1.33
## 3  116.0 9.1 44.1 128.3 0.04
## 4   39.4 6.9 16.4   3.5 0.44
## 5    2.5 4.6  2.9   1.8 1.20
## 6   19.6 7.3  4.5  44.1 0.27
## 7    5.2 5.4  2.8   3.4 0.48
## 8   71.4 8.1 55.2  33.7 0.19
## 9   26.4 5.8  9.2   1.6 0.83
## 10   4.8 6.4  4.6  22.5 0.81
## 11   6.6 5.4  2.7  14.9 0.71
## 12  16.5 7.2 13.8   4.0 0.50
## 13  25.4 7.2 25.2  11.6 0.49
## 14   7.1 5.8  5.2   5.8 1.16
## 15 128.0 7.6 86.5  71.1 0.05
## 16  83.7 8.2 66.5  78.6 0.15
```

```
## 17 108.5 8.7 35.6 80.1 0.19
## 18 61.3 7.8 57.4 13.9 0.77
## 19 6.4 5.8 4.0 4.6 1.08
## 20 31.0 6.7 15.0 17.0 0.98
## 21 7.5 4.4 2.0 9.6 0.63
## 22 17.3 6.7 10.7 9.5 0.56
## 23 12.6 6.1 3.7 21.0 0.41
## 24 7.0 6.9 6.3 32.1 0.73
## 25 10.5 5.5 6.3 1.6 0.34
## 26 30.0 6.9 13.9 21.5 0.59
## 27 55.4 7.3 15.9 24.7 0.34
## 28 3.9 4.5 3.3 7.0 0.84
## 29 5.5 4.8 1.7 14.8 0.50
## 30 6.3 5.8 3.3 0.7 0.34
## 31 67.0 7.8 58.6 43.8 0.28
## 32 28.8 7.4 10.2 32.7 0.34
## 33 5.8 3.6 1.6 3.2 0.87
## 34 4.5 4.4 1.1 3.2 0.56
## 35 119.1 7.9 38.4 16.1 0.17
## 36 25.4 7.1 8.8 45.2 0.18
## 37 106.5 6.8 90.7 16.5 0.19
## 38 53.0 8.4 45.6 152.4 0.04
## 39 8.5 7.0 2.5 12.8 0.49
## 40 87.6 7.5 85.5 20.1 1.10
## 41 114.0 7.0 72.6 6.4 0.16
## 42 97.5 6.8 45.5 6.2 0.10
## 43 11.8 5.9 24.2 1.6 0.48
## 44 66.5 8.3 26.0 68.2 0.21
## 45 16.0 6.7 41.2 24.1 0.86
## 46 5.0 6.2 23.6 9.6 0.52
## 47 25.6 6.2 12.6 27.7 0.65
## 48 81.5 8.9 20.5 9.6 0.27
## 49 1.2 4.3 2.1 6.4 0.94
## 50 34.0 7.0 13.1 4.6 0.40
## 51 15.5 6.9 5.2 16.5 0.43
## 52 17.3 5.2 3.0 2.6 0.25
## 53 71.8 7.9 20.5 8.8 0.27
```

```
library(Hmisc)
```

Correlacion

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##      %+%, alpha

##
## Attaching package: 'Hmisc'

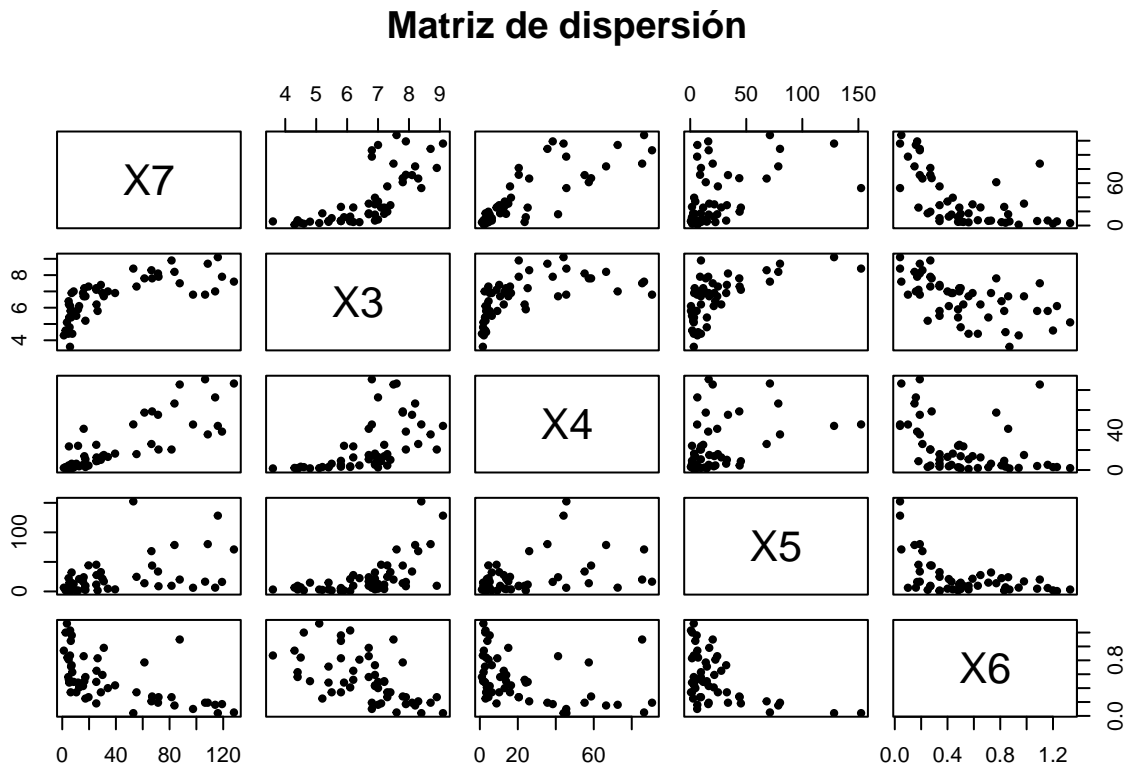
## The following object is masked from 'package:psych':
##
##      describe

## The following objects are masked from 'package:base':
##
##      format.pval, units
```

```
Rc = rcorr(as.matrix(medidas))
Rc
```

```
##      X3      X4      X5      X6      X7
## X3  1.00  0.72  0.83  0.48 -0.59
## X4  0.72  1.00  0.58  0.61 -0.58
## X5  0.83  0.58  1.00  0.41 -0.40
## X6  0.48  0.61  0.41  1.00 -0.49
## X7 -0.59 -0.58 -0.40 -0.49  1.00
##
## n= 53
##
##
## P
##      X3      X4      X5      X6      X7
## X3          0.0000 0.0000 0.0003 0.0000
## X4 0.0000          0.0000 0.0000 0.0000
## X5 0.0000 0.0000          0.0023 0.0029
## X6 0.0003 0.0000 0.0023          0.0002
## X7 0.0000 0.0000 0.0029 0.0002
```

```
pairs(medidas, labels=c("X7", "X3", "X4", "X5", "X6"), main="Matriz de dispersión", pch=20)
```



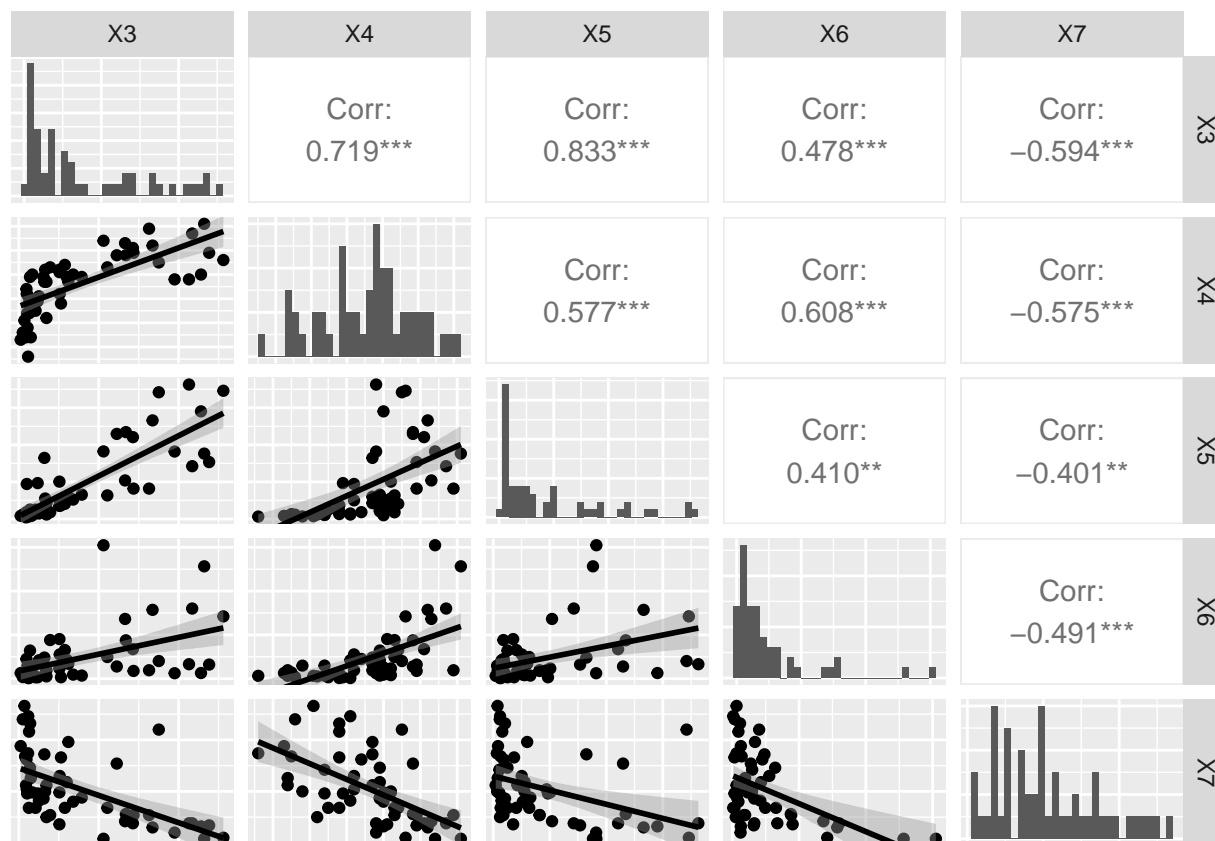
```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2
```

```
ggpairs(medidas, lower = list(continuous = "smooth"),
        diag = list(continuous = "barDiag"), axisLabels = "none")
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

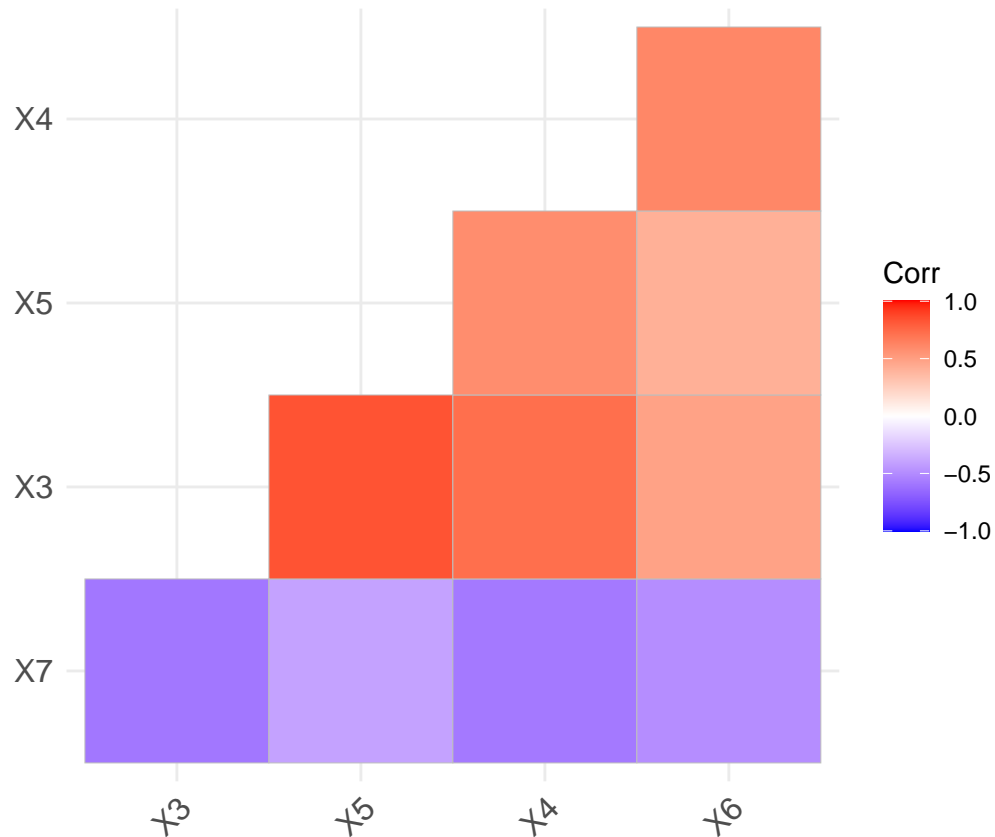


```
library(ggcorrplot)
library(polycor)
```

```
##
## Attaching package: 'polycor'
```

```
## The following object is masked from 'package:psych':
##
## polyserial
```

```
mat_cor <- hetcor(medidas)$correlations #matriz de correlación policorica
ggcorrplot(mat_cor, type="lower", hc.order = T)
```

El modelo

```
R=lm(X7~X3+X4+X5+X6, data = medidas)
summary(R)
```

```
##
## Call:
## lm(formula = X7 ~ X3 + X4 + X5 + X6, data = medidas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42260 -0.19155 -0.08438  0.14334  0.62234
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.004440   0.257561   3.900 0.000299 ***
## X3           -0.005503   0.002028  -2.713 0.009224 **
## X4           -0.046709   0.045329  -1.030 0.307968
## X5            0.004129   0.002648   1.559 0.125484
## X6           -0.002361   0.001497  -1.577 0.121257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2629 on 48 degrees of freedom
```

```
## Multiple R-squared:  0.4515, Adjusted R-squared:  0.4058
## F-statistic: 9.879 on 4 and 48 DF,  p-value: 6.499e-06
```

Selección del mejor modelo

```
step(R,direction="both",trace=1)
```

```
## Start:  AIC=-136.87
## X7 ~ X3 + X4 + X5 + X6
##
##           Df Sum of Sq    RSS    AIC
## - X4      1  0.07338 3.3904 -137.72
## <none>                    3.3171 -136.87
## - X5      1  0.16803 3.4851 -136.25
## - X6      1  0.17196 3.4890 -136.19
## - X3      1  0.50874 3.8258 -131.31
##
## Step:  AIC=-137.71
## X7 ~ X3 + X5 + X6
##
##           Df Sum of Sq    RSS    AIC
## <none>                    3.3904 -137.72
## - X5      1  0.18606 3.5765 -136.88
## + X4      1  0.07338 3.3171 -136.87
## - X6      1  0.35080 3.7412 -134.50
## - X3      1  0.90855 4.2990 -127.13

##
## Call:
## lm(formula = X7 ~ X3 + X5 + X6, data = medidas)
##
## Coefficients:
## (Intercept)          X3          X5          X6
##   0.744583   -0.006487   0.004333  -0.003035
```

El mejor modelo

```
R1=lm(X7~X3+X4+X5+X6, data = medidas)
S=summary(R1)
S
```

```
##
## Call:
## lm(formula = X7 ~ X3 + X4 + X5 + X6, data = medidas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42260 -0.19155 -0.08438  0.14334  0.62234
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.004440   0.257561   3.900 0.000299 ***
## X3          -0.005503   0.002028  -2.713 0.009224 **
## X4          -0.046709   0.045329  -1.030 0.307968
## X5           0.004129   0.002648   1.559 0.125484
## X6          -0.002361   0.001497  -1.577 0.121257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2629 on 48 degrees of freedom
## Multiple R-squared:  0.4515, Adjusted R-squared:  0.4058
## F-statistic: 9.879 on 4 and 48 DF,  p-value: 6.499e-06
```

Intervalos de confianza

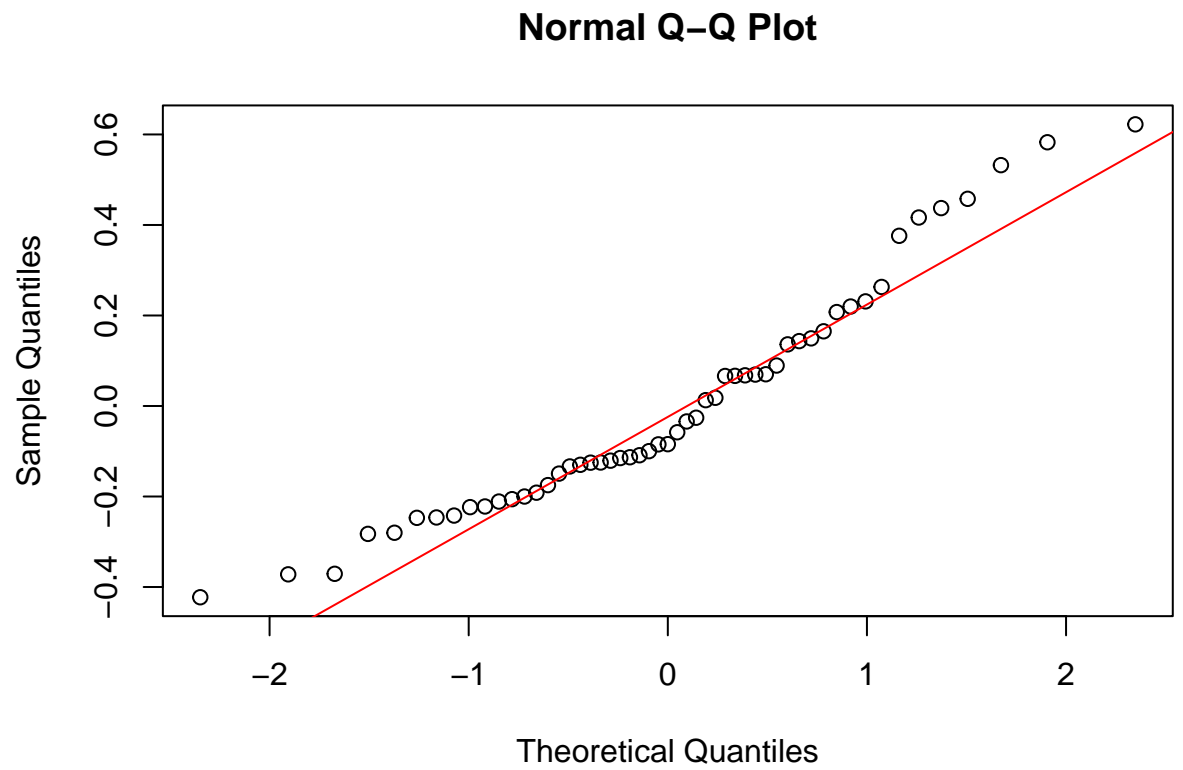
```
confint(R1)
```

```
##           2.5 %          97.5 %
## (Intercept)  0.486579664  1.5223004429
## X3          -0.009580788 -0.0014250351
## X4          -0.137848351  0.0444310564
## X5          -0.001195061  0.0094536648
## X6          -0.005371081  0.0006484214
```

Verificación de supuestos

```
E=R1$residuals
Y=R1$fitted.values

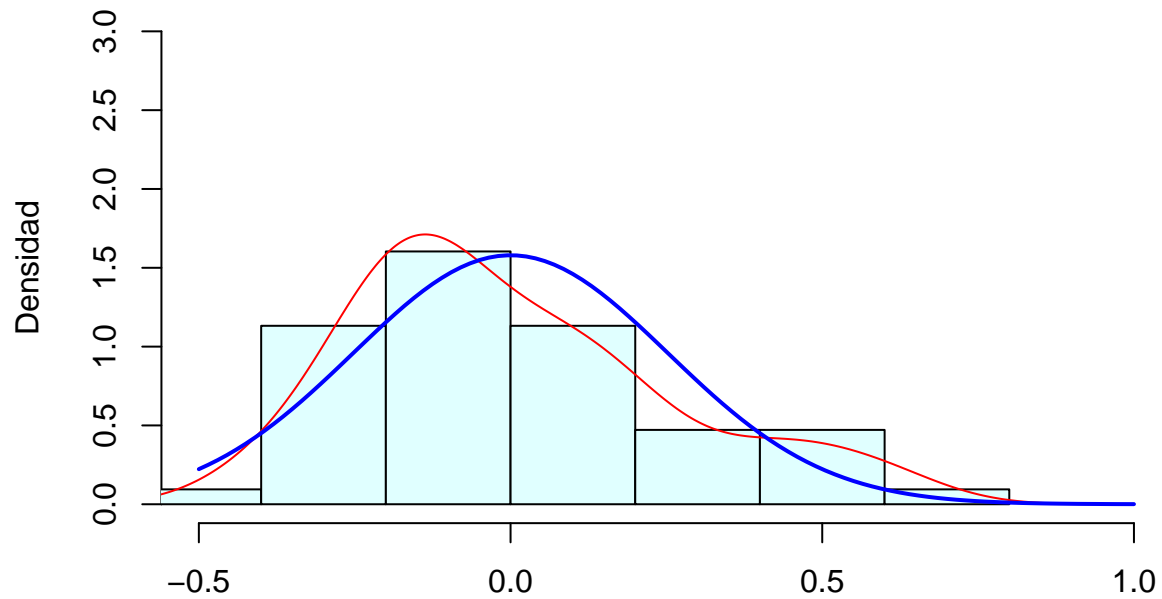
qqnorm(E)
qqline(E,col="red")
```



Normalidad

```
hist(E,col="lightcyan",freq=FALSE,main="Histograma de Residuos",ylim=c(0,3), xlim = c(-0.5, 1),xlab="",  
lines(density(E),col="red")  
curve(dnorm(x,mean=mean(E),sd=sd(E)), add=TRUE, col="blue",lwd=2)
```

Histograma de Residuos

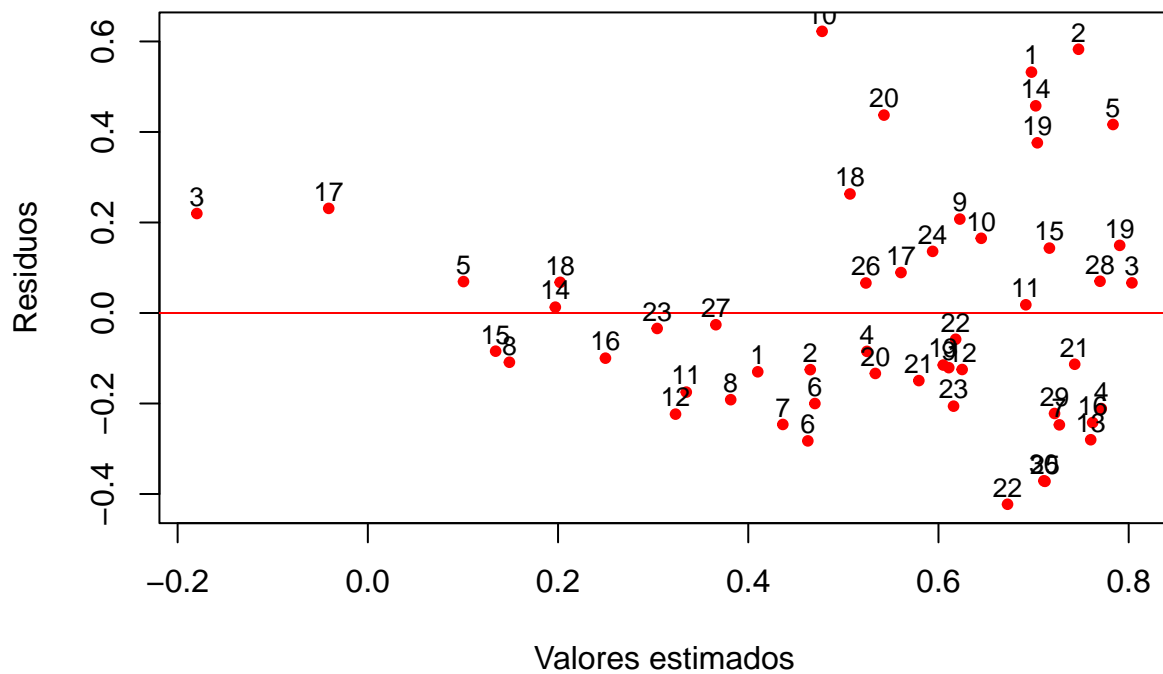


```
shapiro.test(E)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: E  
## W = 0.94148, p-value = 0.01176
```

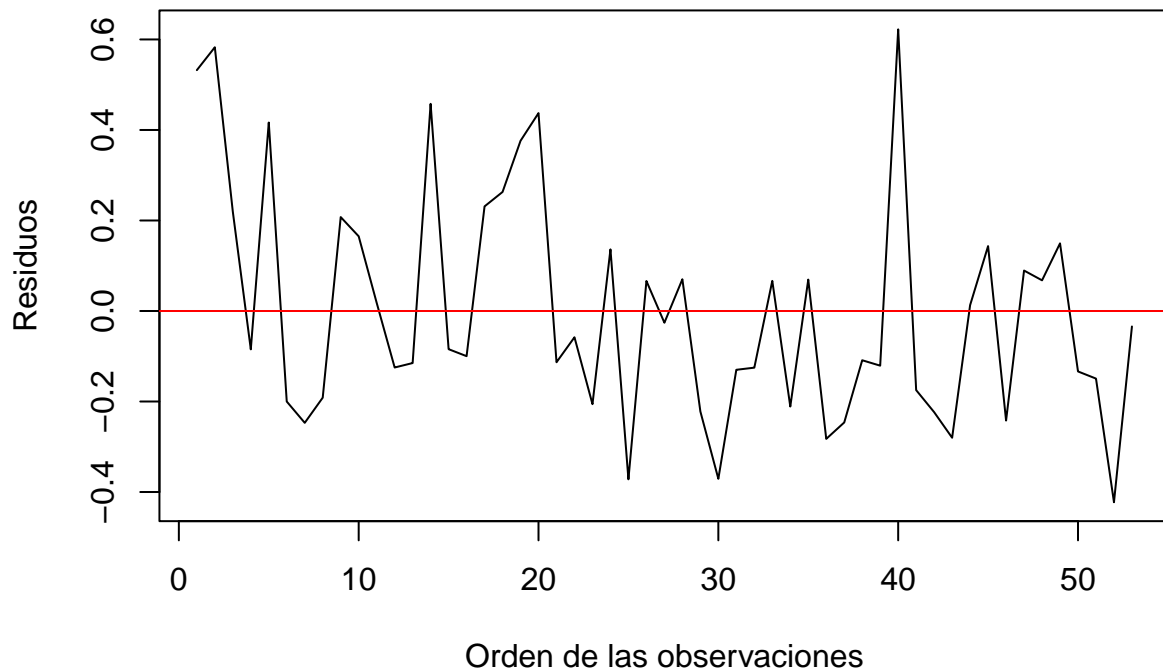
Homocedasticidad y modelo apropiado Gráfica Valores estimados vs Residuos

```
plot(Y,E,ylab="Residuos",xlab="Valores estimados",pch=20,col="red")  
abline(h=0,col="red")  
text(Y[,E[]],1:30,cex=0.8,pos=3,offset=0.2)
```



Independencia Errores vs Orden de observación

```
n=length(medidas$X7)
plot(c(1:n),R1$residuals,type="l",xlab="Orden de las observaciones",ylab="Residuos")
abline(h=0,col="red")
```



#Prueba de autocorrelación para verificar independencia: $H_0: \rho=0$

```
library(car)
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
```

```
##
```

```
##      logit
```

```
dwt(R1,alternative="two.sided")
```

```
## lag Autocorrelation D-W Statistic p-value
```

```
## 1 0.188473 1.537309 0.092
```

```
## Alternative hypothesis: rho != 0
```

Datos atípicos o influyentes

Datos atípicos Se estandarizan los residuos y se observa si hay distancias mayores a 3.

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
```

```
##
```

```
##      recode
```

```
## The following objects are masked from 'package:Hmisc':
```

```
##
```

```
##      src, summarize
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

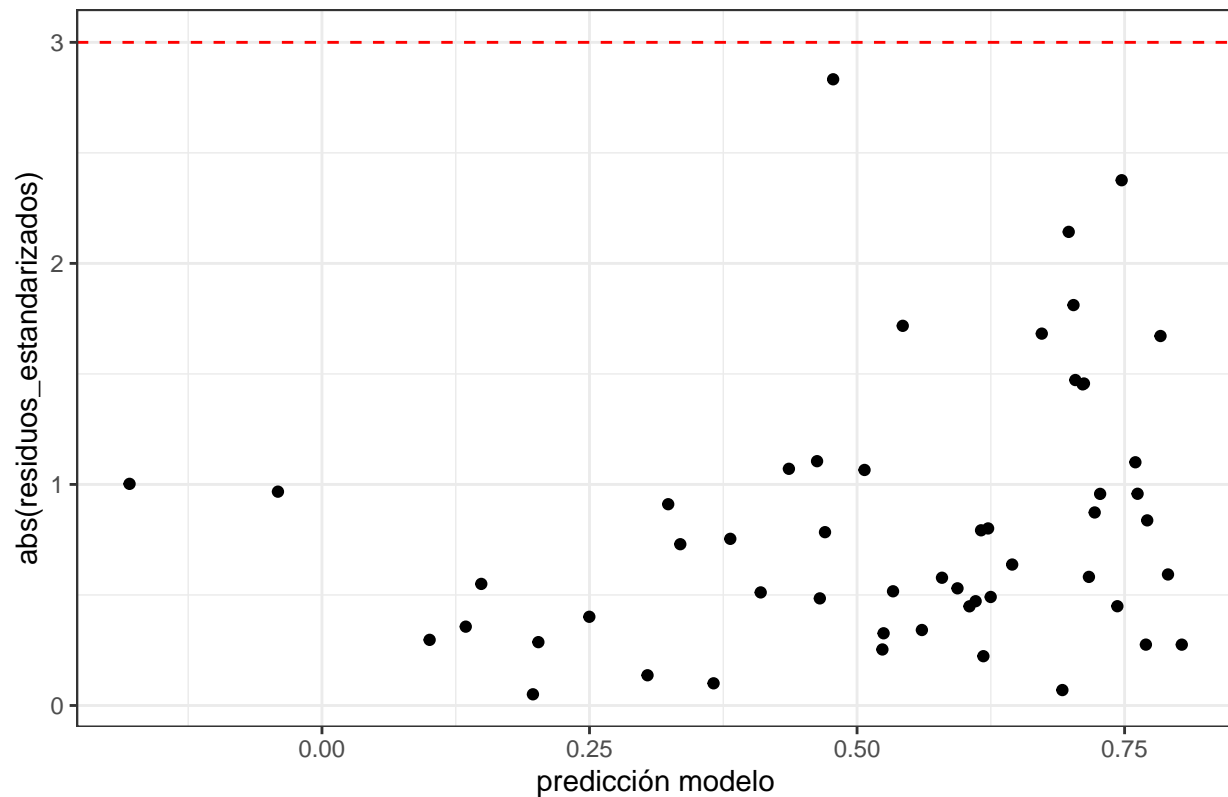
```
##
```

```
##      intersect, setdiff, setequal, union
```

```
medidas$residuos_estandarizados <- rstudent(R1) #Introduce una columna en D con los residuos del model
```

```
ggplot(data = medidas, aes(x = predict(R1), y = abs(residuos_estandarizados))) +  
  geom_hline(yintercept = 3, color = "red", linetype = "dashed") +  
  # se identifican en rojo observaciones con residuos estandarizados absolutos > 3  
  geom_point(aes(color = ifelse(abs(residuos_estandarizados) > 3, 'red', 'black'))) +  
  scale_color_identity() +  
  labs(title = "Distribución de los residuos estandarizados", x = "predicción modelo") +  
  theme_bw() + theme(plot.title = element_text(hjust = 0.5))
```


Distribución de los residuos estandarizados



```
which(abs(medidas$residuos_estandarizados)>3)
```

```
## integer(0)
```

```
summary(influence.measures(R1))
```

Datos influyentes

```
## Potentially influential observations of
## lm(formula = X7 ~ X3 + X4 + X5 + X6, data = medidas) :
##
```

	dfb.1_	dfb.X3	dfb.X4	dfb.X5	dfb.X6	dffit	cov.r	cook.d	hat
## 2	0.33	0.02	-0.24	-0.06	0.03	0.52	0.66_*	0.05	0.05
## 3	0.07	0.28	-0.12	-0.27	0.47	0.66	1.44_*	0.09	0.30_*
## 15	-0.08	-0.06	0.09	-0.05	-0.06	-0.18	1.38_*	0.01	0.20
## 33	0.12	0.05	-0.11	-0.02	0.04	0.12	1.33_*	0.00	0.17
## 35	0.01	0.14	-0.01	-0.09	-0.05	0.16	1.42_*	0.01	0.22
## 38	-0.02	0.15	0.04	-0.10	-0.43	-0.49	1.93_*	0.05	0.44_*
## 40	-0.12	-0.49	0.09	1.16_*	-0.39	1.41_*	0.63_*	0.35	0.20
## 48	-0.08	0.05	0.09	-0.07	-0.08	0.15	1.39_*	0.00	0.21

```
influence.measures(R1)
```

```
## Influence measures of
## lm(formula = X7 ~ X3 + X4 + X5 + X6, data = medidas) :
##
##      dfb.1_      dfb.X3      dfb.X4      dfb.X5      dfb.X6      dffit cov.r      cook.d      hat
## 1 -0.07040 -0.133599  0.15662 -0.02955 -0.170510  0.4389 0.726 3.59e-02 0.0403
## 2  0.32927  0.023407 -0.24113 -0.05998  0.029903  0.5186 0.660 4.90e-02 0.0455
## 3  0.07379  0.281977 -0.11703 -0.27015  0.472155  0.6633 1.437 8.80e-02 0.3044
## 4  0.02165 -0.011063 -0.03035  0.01965  0.044005 -0.0687 1.147 9.61e-04 0.0424
## 5  0.36806  0.060926 -0.30733 -0.02583  0.071785  0.4520 0.893 3.94e-02 0.0681
## 6  0.07800  0.053554 -0.09896  0.06019 -0.062552 -0.2074 1.114 8.67e-03 0.0654
## 7 -0.08766  0.008088  0.05210  0.02090  0.009517 -0.1881 1.048 7.09e-03 0.0372
## 8  0.10397  0.082928 -0.10634 -0.12273  0.057920 -0.2133 1.130 9.19e-03 0.0741
## 9  0.06617  0.061603 -0.04256 -0.05817 -0.044903  0.1538 1.077 4.77e-03 0.0356
## 10 -0.02594 -0.069595  0.04845  0.00380  0.014838  0.1340 1.111 3.63e-03 0.0423
## 11  0.00852  0.000192 -0.00620 -0.00242  0.003962  0.0143 1.157 4.15e-05 0.0402
## 12  0.09521  0.076036 -0.11389 -0.02473  0.076027 -0.1433 1.175 4.17e-03 0.0785
## 13  0.07000  0.076345 -0.08346 -0.05117  0.052065 -0.1172 1.162 2.79e-03 0.0639
## 14  0.05813 -0.080975  0.00829 -0.00666 -0.050908  0.3290 0.819 2.07e-02 0.0319
## 15 -0.07550 -0.055239  0.09020 -0.05323 -0.062196 -0.1809 1.378 6.67e-03 0.2047
## 16  0.00904  0.044819 -0.00222 -0.08036 -0.071460 -0.1479 1.241 4.45e-03 0.1196
## 17  0.00657  0.292153 -0.02735 -0.27454  0.163576  0.4452 1.220 3.97e-02 0.1748
## 18 -0.18173 -0.197977  0.19439  0.27012 -0.186581  0.3855 1.115 2.96e-02 0.1158
## 19  0.04156 -0.060559  0.01366 -0.01701 -0.050558  0.2719 0.917 1.44e-02 0.0330
## 20 -0.04741 -0.009920  0.09106 -0.05601 -0.067150  0.2683 0.839 1.38e-02 0.0238
## 21 -0.13240 -0.045445  0.11809  0.02706 -0.047717 -0.1454 1.202 4.30e-03 0.0950
## 22  0.01916  0.018024 -0.02674 -0.00101  0.017752 -0.0449 1.150 4.12e-04 0.0390
## 23 -0.02704  0.012523  0.00214  0.04091 -0.029630 -0.1420 1.073 4.06e-03 0.0311
## 24 -0.05329 -0.078417  0.07105  0.00760  0.021161  0.1331 1.146 3.60e-03 0.0594
## 25 -0.12298  0.001958  0.07161  0.02125  0.044210 -0.2740 0.922 1.47e-02 0.0342
## 26 -0.01370 -0.005586  0.02016 -0.00937 -0.007032  0.0427 1.135 3.72e-04 0.0276
## 27  0.00344 -0.013056 -0.00502  0.01607  0.004328 -0.0234 1.170 1.12e-04 0.0518
## 28  0.07030  0.014600 -0.06094 -0.00624  0.022454  0.0807 1.197 1.33e-03 0.0792
## 29 -0.19748 -0.045547  0.16931  0.04119 -0.091303 -0.2374 1.101 1.13e-02 0.0688
## 30 -0.02881  0.053031 -0.02759  0.02458  0.082205 -0.2776 0.925 1.51e-02 0.0352
## 31  0.04114  0.075224 -0.04129 -0.11133 -0.002429 -0.1509 1.175 4.62e-03 0.0800
## 32  0.05644  0.022252 -0.06879  0.03182 -0.000454 -0.1100 1.140 2.46e-03 0.0490
## 33  0.11962  0.045697 -0.11058 -0.01812  0.037785  0.1241 1.326 3.14e-03 0.1690
## 34 -0.22554 -0.070156  0.19625  0.04151 -0.057236 -0.2565 1.128 1.32e-02 0.0858
## 35  0.01077  0.135982 -0.01381 -0.09430 -0.051513  0.1595 1.418 5.19e-03 0.2240
## 36  0.05116  0.041828 -0.07590  0.07558 -0.108439 -0.2515 1.028 1.26e-02 0.0492
## 37 -0.15894 -0.010747  0.17899 -0.37146  0.117402 -0.5880 1.282 6.89e-02 0.2317
## 38 -0.02017  0.152266  0.03525 -0.09615 -0.431233 -0.4889 1.926 4.85e-02 0.4414
## 39  0.07256  0.054200 -0.09126  0.01245  0.037442 -0.1259 1.162 3.22e-03 0.0665
## 40 -0.11574 -0.492568  0.09493  1.16084 -0.393081  1.4137 0.631 3.49e-01 0.1994
## 41 -0.09238 -0.148614  0.10280 -0.06840  0.127949 -0.3385 1.276 2.31e-02 0.1772
## 42 -0.11670 -0.267784  0.11939  0.09474  0.125993 -0.3535 1.171 2.51e-02 0.1310
## 43 -0.00378  0.177061 -0.03333 -0.18567  0.079348 -0.2736 1.039 1.49e-02 0.0582
## 44 -0.00421  0.002871  0.00424 -0.00623  0.006546  0.0151 1.210 4.66e-05 0.0820
## 45 -0.05252 -0.197182  0.06808  0.19648 -0.007381  0.2283 1.237 1.06e-02 0.1335
## 46  0.05099  0.228231 -0.08524 -0.19697  0.046084 -0.2765 1.093 1.53e-02 0.0769
## 47  0.02263  0.003622 -0.01474 -0.01232  0.021314  0.0556 1.126 6.30e-04 0.0258
```

```

## 48 -0.08251  0.053038  0.08708 -0.07468 -0.081267  0.1460  1.387  4.35e-03  0.2061
## 49  0.16790  0.034352 -0.14716 -0.01302  0.056466  0.1891  1.179  7.25e-03  0.0924
## 50  0.04963 -0.004088 -0.06486  0.03205  0.069281 -0.1143  1.133  2.65e-03  0.0467
## 51  0.06119  0.040856 -0.08113  0.02392  0.029625 -0.1279  1.125  3.32e-03  0.0467
## 52 -0.27785 -0.172521  0.22410  0.15595  0.000283 -0.3934  0.875  2.98e-02  0.0519
## 53  0.01755 -0.024869 -0.01968  0.02756  0.026325 -0.0488  1.250  4.85e-04  0.1128
##      inf
## 1
## 2      *
## 3      *
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15      *
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33      *
## 34
## 35      *
## 36
## 37
## 38      *
## 39
## 40      *
## 41
## 42
## 43
## 44
## 45
## 46
## 47

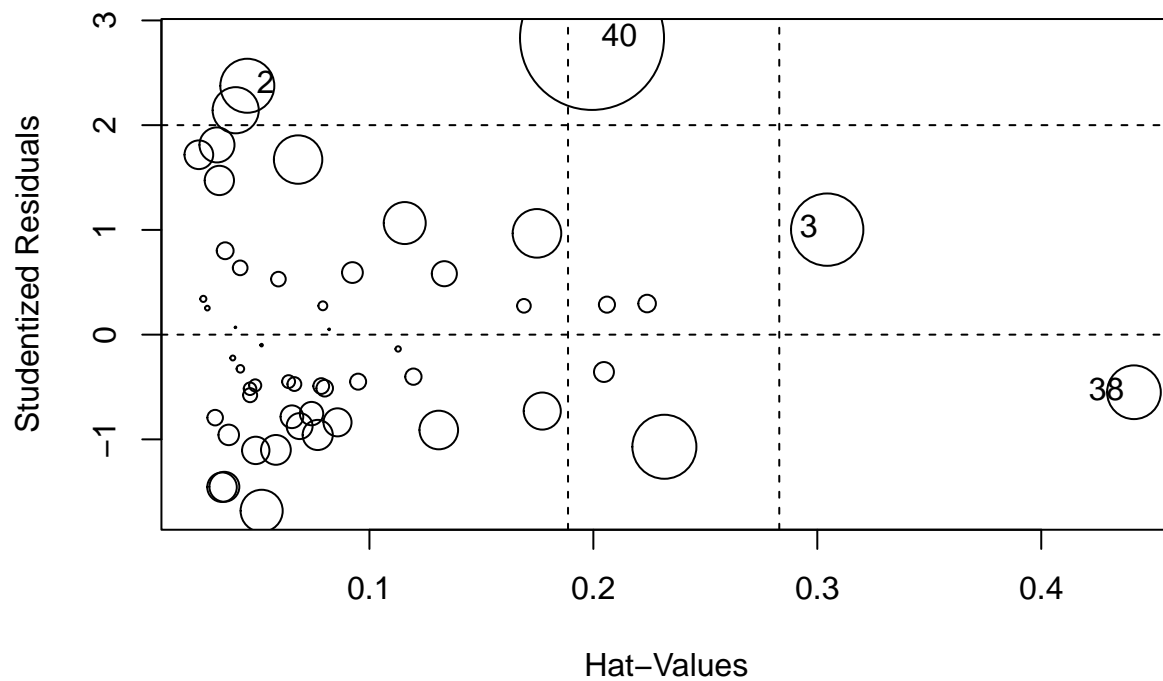
```

```
## 48 *
## 49
## 50
## 51
## 52
## 53
```

Se consideran influyentes aquellas observaciones:

- que tengan leverages mayores a $2.5(p+1)/n = 2.5*3/30 = 2.5/10 = 0.25$
- que tengan distancia de Cook superiores a $4/n$.

```
influencePlot(R1)
```



##	StudRes	Hat	CookD
## 2	2.3762467	0.04547087	0.04904881
## 3	1.0025970	0.30441997	0.08797551
## 38	-0.5500381	0.44139814	0.04851772
## 40	2.8328215	0.19939600	0.34869783