

Momento de Retroalimentación (Portafolio Implementación)

Construcción de un Modelo Estadístico Base

Jorge Chávez Badillo A01749448

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 llevó a cabo un estudio reciente en 53 lagos de Florida con el fin de examinar los factores que inflúan en el nivel de contaminación por mercurio. Las variables que se midieron se encuentran en mercurio.csv. Descargar mercurio.csv y su descripción es la siguiente:

- $X1$ = 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)

Alrededor de la principal pregunta de investigación que surge en este estudio: ¿Cuáles son los principales factores que influyen en el nivel de contaminación por mercurio en los peces de los lagos de Florida? pueden surgir preguntas paralelas que desglosan esta pregunta general:

1. ¿Hay evidencia para suponer que la concentración promedio de mercurio en los lagos es dañino para la salud humana? Considera que las normativas de referencia para evaluar los niveles máximos de Hg (Reglamento 34687-MAG y los reglamentos internacionales CE 1881/2006 y Codex Standard 193-1995) establecen que la concentración promedio de mercurio en productos de la pesca no debe superar los 0.5 mg de Hg/kg.
2. ¿Habrá diferencia significativa entre la concentración de mercurio por la edad de los peces?
3. Si el muestreo se realizó lanzando una red y analizando los peces que la red encontraba ¿Habrá influencia del número de peces encontrados en la concentración de mercurio en los peces?
4. ¿Las concentraciones de alcalinidad, clorofila, calcio en el agua del lago influyen en la concentración de mercurio de los peces?

Exploración de la Base de Datos

Lectura de Datos

```
db_mercurio = read.csv("mercurio.csv")
n_variables = length(db_mercurio)
n_rows = nrow(db_mercurio)

sprintf("Número de Variables: %s", n_variables)
```

```
## [1] "Número de Variables: 12"
```

```
sprintf("Número de Registros: %s", n_rows)
```

```
## [1] "Número de Registros: 53"
```

Cálculo de Medidas Estadísticas y Visualización de los Datos

Variables Cuantitativas

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

db_mercurio_num = db_mercurio[3:12]
n = length(db_mercurio_num) # número de variables
d = matrix(NA, ncol = 9, nrow = n)
for(i in 1:n) {
  d[i, ] <- c(as.numeric(summary(db_mercurio_num[, i])), sd(db_mercurio_num[, i]), var(db_mercurio_num[, i]))
}
m = as.data.frame(d)
row.names(m) = c("X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10", "X11", "X12")
names(m) = c("Mínimo", "Q1", "Mediana", "Media", "Q3", "Máximo", "Desv Est", "Varianza", "Moda")
m
```

Medidas de Tendencia Central y de Dispersión

	Mínimo	Q1	Mediana	Media	Q3	Máximo	Desv Est	Varianza	Moda
## X3	1.20	6.60	19.60	37.5301887	66.50	128.00	38.2035267	1.459509e+03	25.40
## X4	3.60	5.80	6.80	6.5905660	7.40	9.10	1.2884493	1.660102e+00	6.90
## X5	1.10	3.30	12.60	22.2018868	35.60	90.70	24.9325744	6.216333e+02	3.00
## X6	0.70	4.60	12.80	23.1169811	24.70	152.40	30.8163214	9.496457e+02	3.20
## X7	0.04	0.27	0.48	0.5271698	0.77	1.33	0.3410356	1.163053e-01	0.34
## X8	4.00	10.00	12.00	13.0566038	12.00	44.00	8.5606773	7.328520e+01	12.00
## X9	0.04	0.09	0.25	0.2798113	0.33	0.92	0.2264058	5.125958e-02	0.04
## X10	0.06	0.48	0.84	0.8745283	1.33	2.04	0.5220469	2.725329e-01	1.90
## X11	0.04	0.25	0.45	0.5132075	0.70	1.53	0.3387294	1.147376e-01	0.16
## X12	0.00	1.00	1.00	0.8113208	1.00	1.00	0.3949977	1.560232e-01	1.00

Al hacer el cálculo de los datos estadísticos, es posible tener un mayor entendimiento de la información para facilitar la elección de las variables que serán utilizadas, además de elegir cuáles herramientas estadísticas serán implementadas para la solución del problema.

Medidas de Posición Boxplot y Valores Atípicos

```
x3 = db_mercurio_num$X3
x4 = db_mercurio_num$X4
x5 = db_mercurio_num$X5
x6 = db_mercurio_num$X6
x7 = db_mercurio_num$X7
x8 = db_mercurio_num$X8
x9 = db_mercurio_num$X9
x10 = db_mercurio_num$X10
x11 = db_mercurio_num$X11

# Cuartiles Para x3
q1_3 = quantile(x3, 0.25)
q3_3 = quantile(x3, 0.75)
rc_3 = q3_3 - q1_3 # Rango intercuartílico
y2_3 = q3_3 + 1.5 * rc_3
# IQR(x3)

# Cuartiles Para x4
q1_4 = quantile(x4, 0.25)
q3_4 = quantile(x4, 0.75)
rc_4 = q3_4 - q1_4 # Rango intercuartílico
y2_4 = q3_4 + 1.5 * rc_4
# IQR(x3)

# Cuartiles Para x5
q1_5 = quantile(x5, 0.25)
q3_5 = quantile(x5, 0.75)
rc_5 = q3_5 - q1_5 # Rango intercuartílico
y2_5 = q3_5 + 1.5 * rc_5
# IQR(x5)

# Cuartiles Para x6
q1_6 = quantile(x6, 0.25)
q3_6 = quantile(x6, 0.75)
rc_6 = q3_6 - q1_6 # Rango intercuartílico
y2_6 = q3_6 + 1.5 * rc_6
# IQR(x6)

# Cuartiles Para x7
q1_7 = quantile(x7, 0.25)
q3_7 = quantile(x7, 0.75)
rc_7 = q3_7 - q1_7 # Rango intercuartílico
y2_7 = q3_7 + 1.5 * rc_7
# IQR(x7)

# Cuartiles Para x8
q1_8 = quantile(x8, 0.25)
```

```

q3_8 = quantile(x8, 0.75)
rc_8 = q3_8 - q1_8 # Rango intercuartílico
y2_8 = q3_8 + 1.5 * rc_8
# IQR(x8)

# Cuartiles Para x9
q1_9 = quantile(x9, 0.25)
q3_9 = quantile(x9, 0.75)
rc_9 = q3_9 - q1_9 # Rango intercuartílico
y2_9 = q3_9 + 1.5 * rc_9
# IQR(x9)

# Cuartiles Para x10
q1_10 = quantile(x10, 0.25)
q3_10 = quantile(x10, 0.75)
rc_10 = q3_10 - q1_10 # Rango intercuartílico
y2_10 = q3_10 + 1.5 * rc_10
# IQR(x10)

# Cuartiles Para x11
q1_11 = quantile(x11, 0.25)
q3_11 = quantile(x11, 0.75)
rc_11 = q3_11 - q1_11 # Rango intercuartílico
y2_11 = q3_11 + 1.5 * rc_11
# IQR(x11)

par(mfrow = c(3, 3))

boxplot(x3, main = "Boxplot Alcalinidad", horizontal = TRUE, ylim = c(0, y2_3))
abline(v = y2_3, col="red")
X_3 = db_mercurio_num[x3 < y2_3, c("X3")]
print("Summary x3")

```

```
## [1] "Summary x3"
```

```
summary(X_3)
```

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

```

boxplot(x4, main = "Boxplot PH", horizontal = TRUE, ylim = c(0, y2_4))
abline(v = y2_4, col="red")
X_4 = db_mercurio_num[x4 < y2_4, c("X4")]
print("Summary x4")

```

```
## [1] "Summary x4"
```

```
summary(X_4)
```

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

```

boxplot(x5, main = "Boxplot Calcio", horizontal = TRUE, ylim = c(0, y2_5))
abline(v = y2_5, col="red")
X_5 = db_mercurio_num[x5 < y2_5, c("X5")]
print("Summary x5")

```

```
## [1] "Summary x5"
```

```
summary(X_4)
```

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

```

boxplot(x6, main = "Boxplot Clorofila", horizontal = TRUE, ylim = c(0, y2_6))
abline(v = y2_6, col="red")
X_6 = db_mercurio_num[x6 < y2_6, c("X6")]
print("Summary x6")

```

```
## [1] "Summary x6"
```

```
summary(X_6)
```

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

```

boxplot(x7, main = "Boxplot Media Mercurio", horizontal = TRUE, ylim = c(0, y2_7))
abline(v = y2_7, col="red")
X_7 = db_mercurio_num[x7 < y2_7, c("X7")]
print("Summary x7")

```

```
## [1] "Summary x7"
```

```
summary(X_7)
```

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

```

boxplot(x8, main = "Boxplot # de Peces", horizontal = TRUE, ylim = c(0, y2_8))
abline(v = y2_8, col="red")
X_8 = db_mercurio_num[x8 < y2_8, c("X8")]
print("Summary x8")

```

```
## [1] "Summary x8"
```

```
summary(X_8)
```

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

```

boxplot(x9, main = "Boxplot Mínimo Mercurio", horizontal = TRUE, ylim = c(0, y2_9))
abline(v = y2_9, col="red")
X_9 = db_mercurio_num[x9 < y2_9, c("X9")]
print("Summary x9")

```

```
## [1] "Summary x9"
```

```
summary(X_9)
```

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

```

boxplot(x10, main = "Boxplot Máximo Mercurio", horizontal = TRUE, ylim = c(0, y2_10))
abline(v = y2_10, col="red")
X_10 = db_mercurio_num[x10 < y2_10, c("X10")]
print("Summary x10")

```

```
## [1] "Summary x10"
```

```
summary(X_10)
```

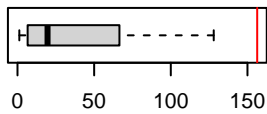
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0600  0.4800  0.8400  0.8745  1.3300  2.0400
```

```

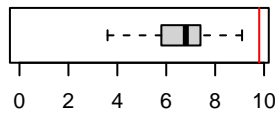
boxplot(x11, main = "Boxplot Estimación", horizontal = TRUE, ylim = c(0, y2_11))
abline(v = y2_11, col="red")

```

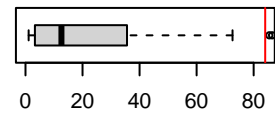
Boxplot Alcalinidad



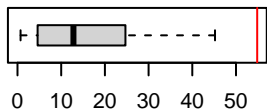
Boxplot PH



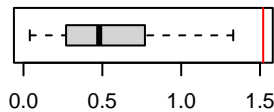
Boxplot Calcio



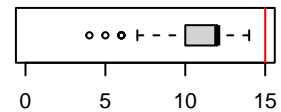
Boxplot Clorofila



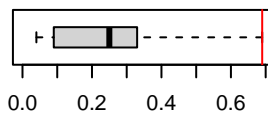
Boxplot Media Mercurio



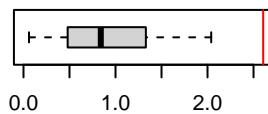
Boxplot # de Peces



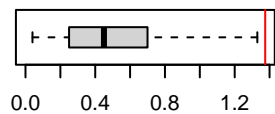
Boxplot Mínimo Mercurio



Boxplot Máximo Mercurio



Boxplot Estimación



```
X_11 = db_mercurio_num[x11 < y2_11, c("X11")]
print("Summary x11")
```

```
## [1] "Summary x11"
```

```
summary(X_11)
```

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

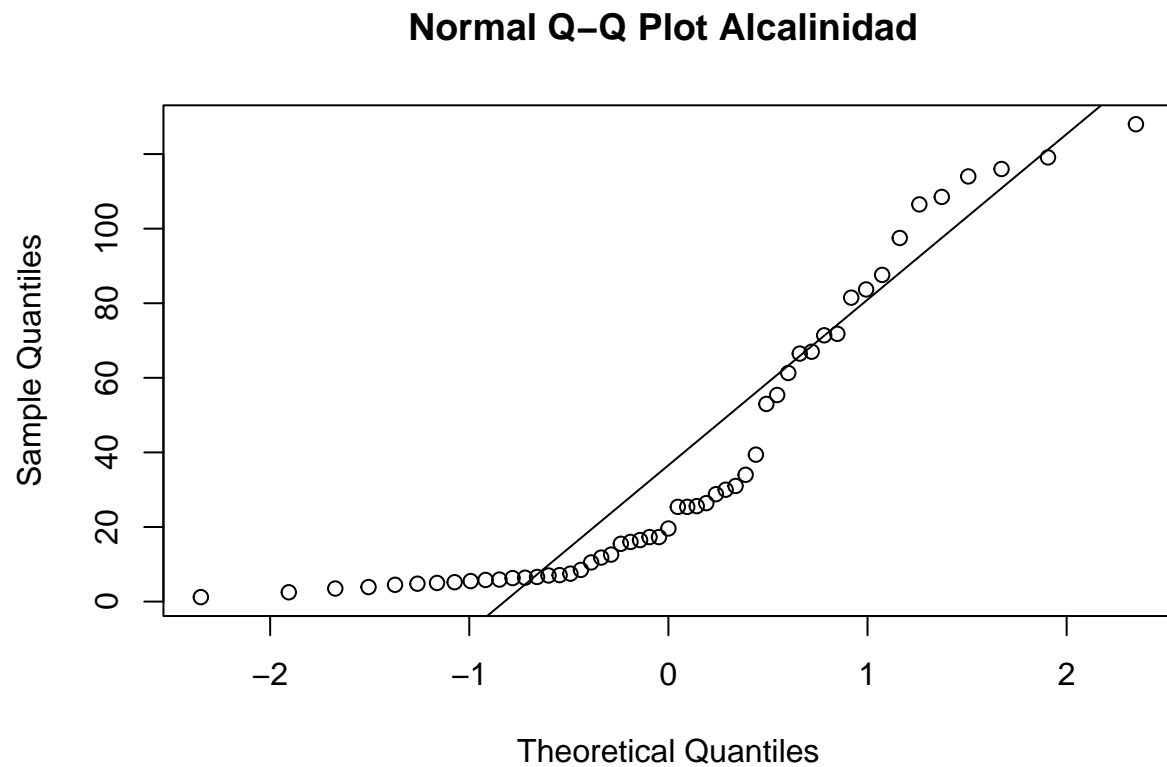
Como se puede observar en los boxblots anteriores, tenemos cada una de las variables numéricas representadas de esta forma para poder tener un mayor entendimiento de su comportamiento.

Con estos gráficos es posible obtener información sobre la forma general de la curva, es decir características como la simetría, la curtosis, la mediana, distribución de los datos hacia ambos lados de los valores centrales así como también la presencia de datos atípicos.

Es importante mencionar que los boxplots de las variables del número de peces y el calcio presentan algunos datos atípicos ya que los cálculos no exceden los valores del rango, al observar la gráfica se puede concluir que estos no tienen un gran efecto los cálculos posteriores pues no son una cantidad que represente un riesgo.

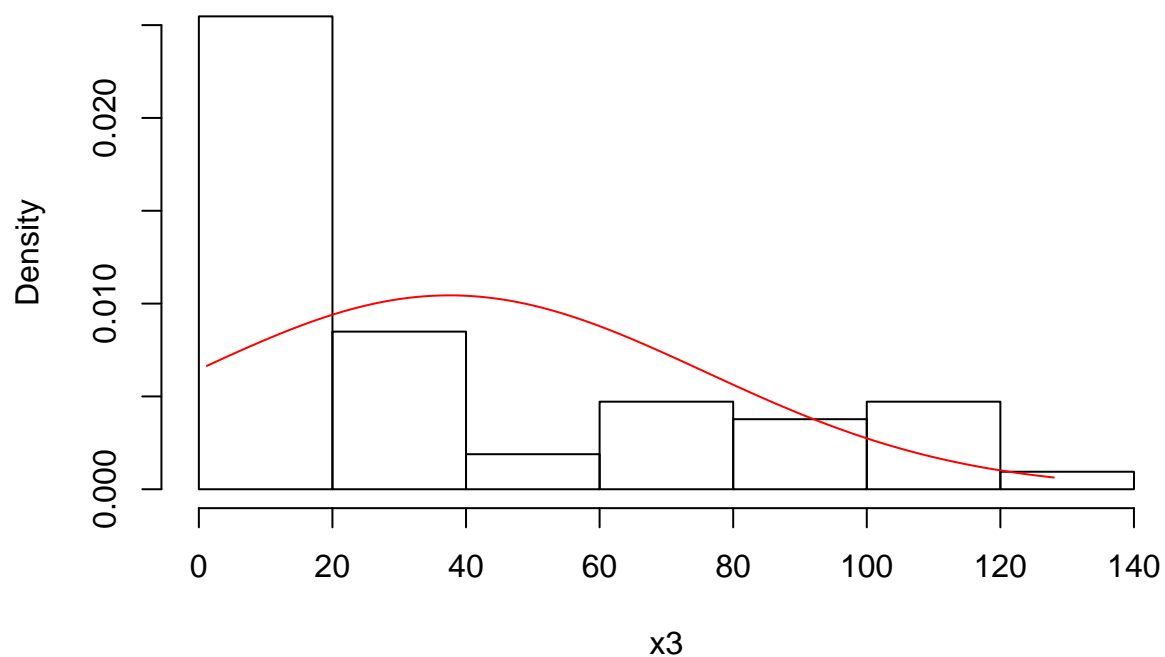
Distribución de los Datos Q-Q Plot e Histogramas

```
# x3
# QQplot
qqnorm(x3, main = "Normal Q-Q Plot Alcalinidad ")
qqline(x3)
```



```
# Histograma
hist(x3, main = "Histograma de Alcalinidad", prob = TRUE, col = 0)
x = seq(min(x3), max(x3), 0.1)
y = dnorm(x, mean(x3), sd(x3))
lines(x, y, col = "red")
```

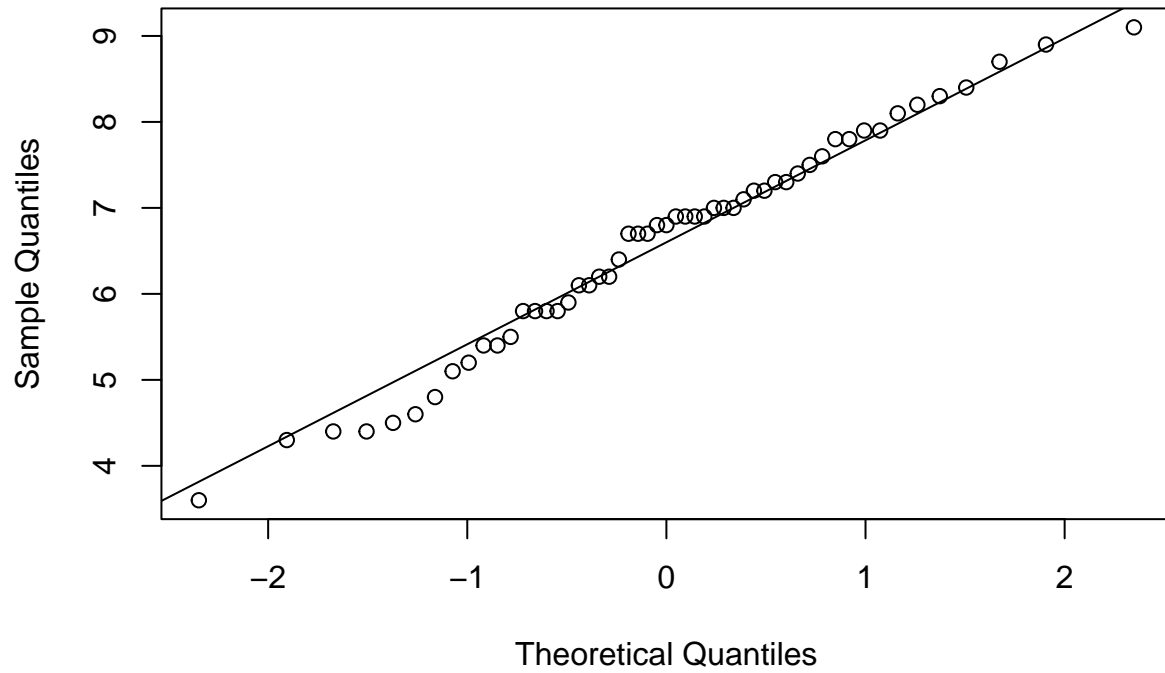

Histograma de Alcalinidad



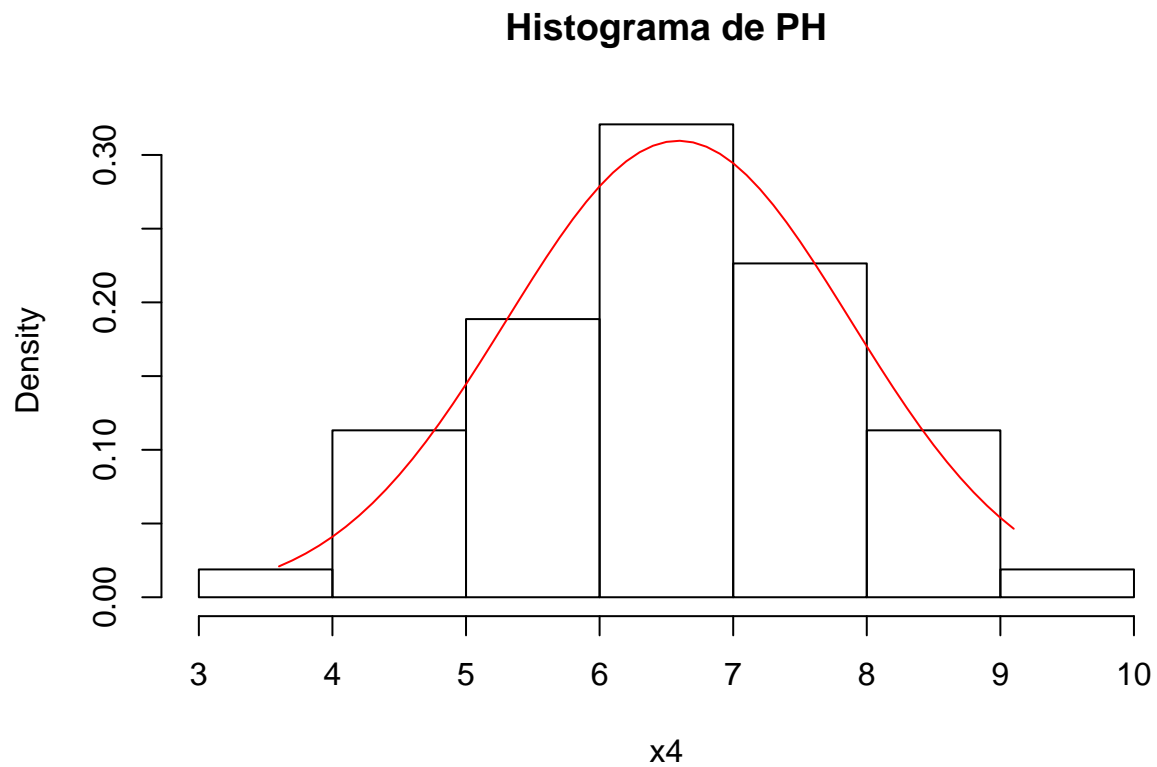
Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una distribución con colas delgadas (alta, curtosis y distribución leptocúrtica), lo cual se comprueba con el histograma y podemos verificar que, en efecto, la distribución cuenta con una gran concentración de valores.

```
# x4  
# QQplot  
qqnorm(x4, main = "Normal Q-Q Plot PH ")  
qqline(x4)
```

Normal Q-Q Plot PH



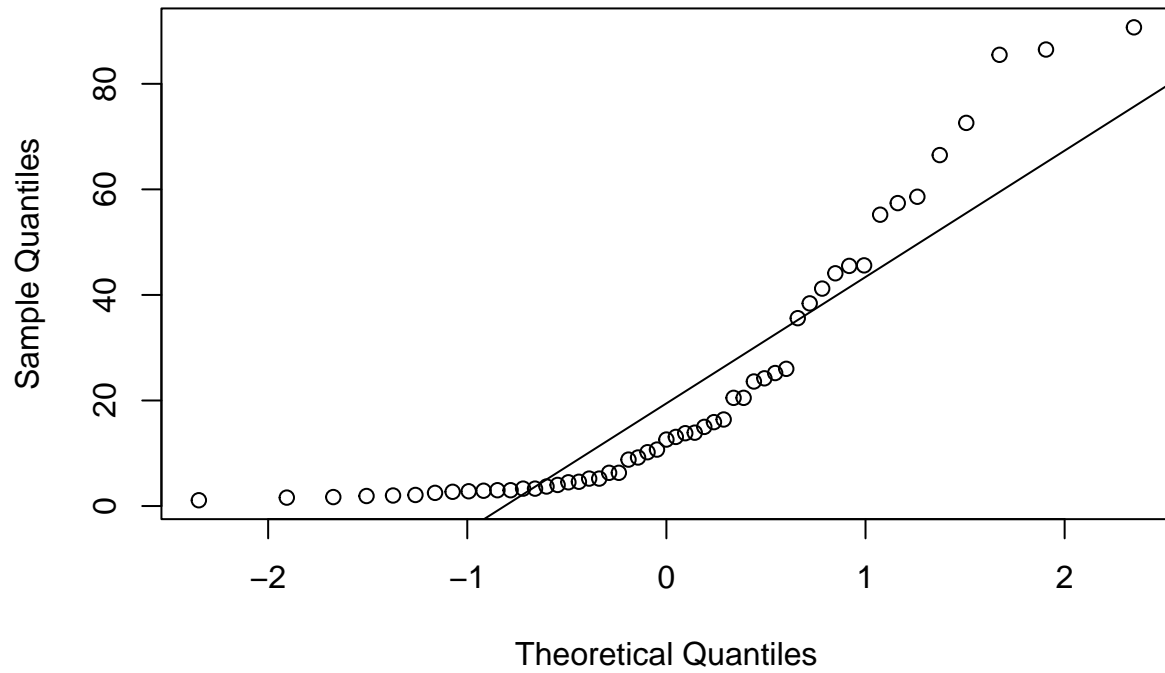
```
# Histograma  
hist(x4, main = "Histograma de PH", prob = TRUE, col = 0)  
x = seq(min(x4), max(x4), 0.1)  
y = dnorm(x, mean(x4), sd(x4))  
lines(x, y, col = "red")
```



Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal es aproximadamente ideal, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran con una distribución simétrica.

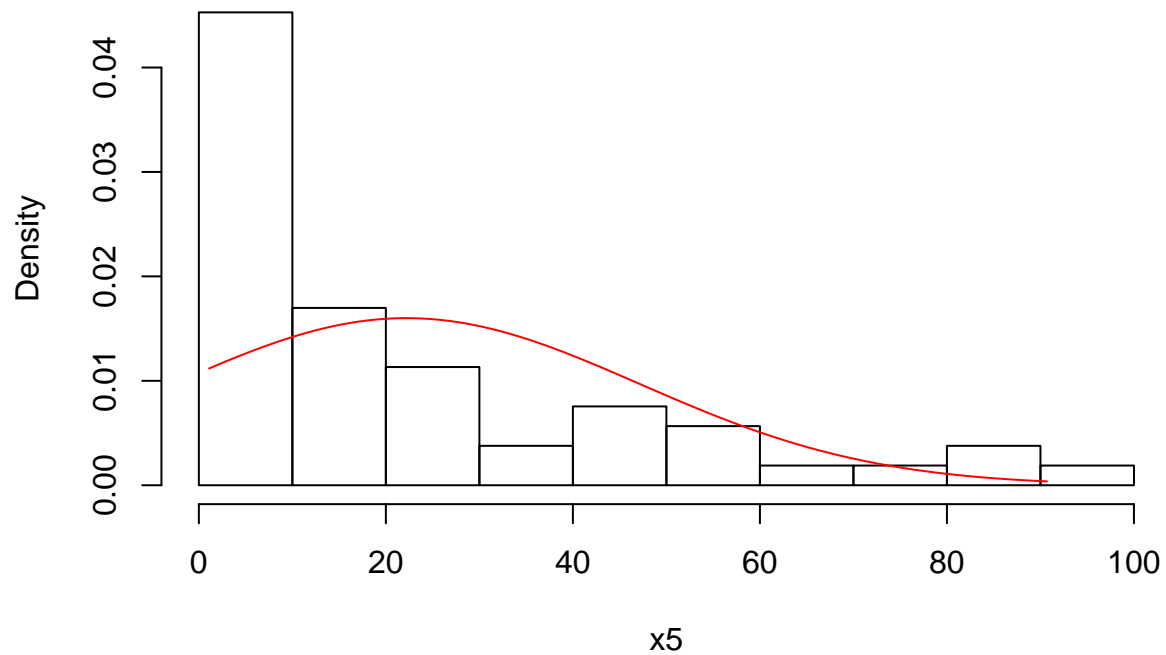
```
# x5
# QQplot
qqnorm(x5, main = "Normal Q-Q Plot Calcio ")
qqline(x5)
```

Normal Q-Q Plot Calcio



```
# Histograma
hist(x5, main = "Histograma de PH", prob = TRUE, col = 0)
x = seq(min(x5), max(x5), 0.1)
y = dnorm(x, mean(x5), sd(x5))
lines(x, y, col = "red")
```

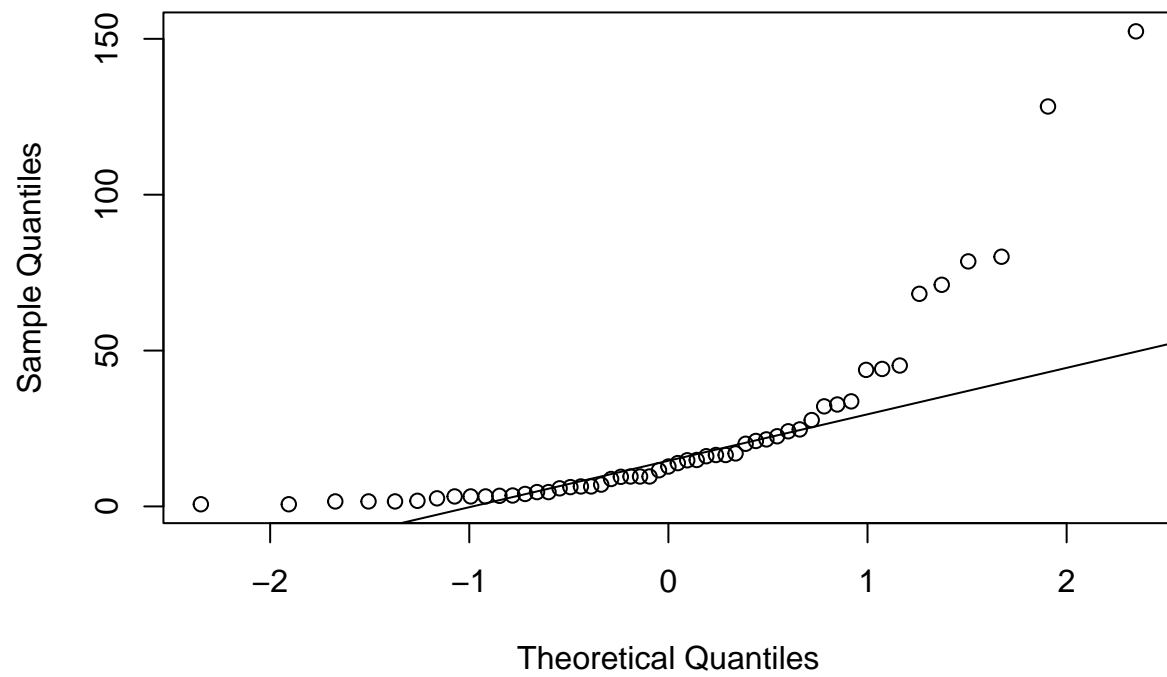
Histograma de PH



Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una asimetría positiva con un sesgo a la derecha, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran recargados hacia el lado izquierdo, lo que significa que la distribución se encuentra sesgada a la derecha.

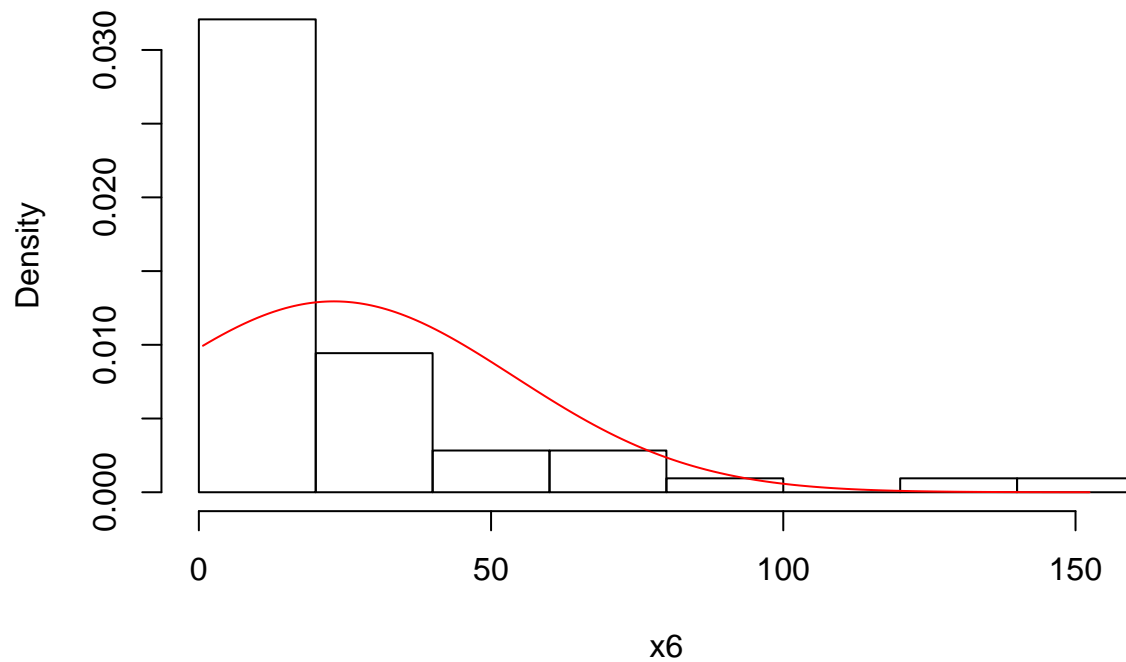
```
# x6
# QQplot
qqnorm(x6, main = "Normal Q-Q Plot Clorofila ")
qqline(x6)
```

Normal Q-Q Plot Clorofila



```
# Histograma  
hist(x6, main = "Histograma de Clorofila", prob = TRUE, col = 0)  
x = seq(min(x6), max(x6), 0.1)  
y = dnorm(x, mean(x6), sd(x6))  
lines(x, y, col = "red")
```

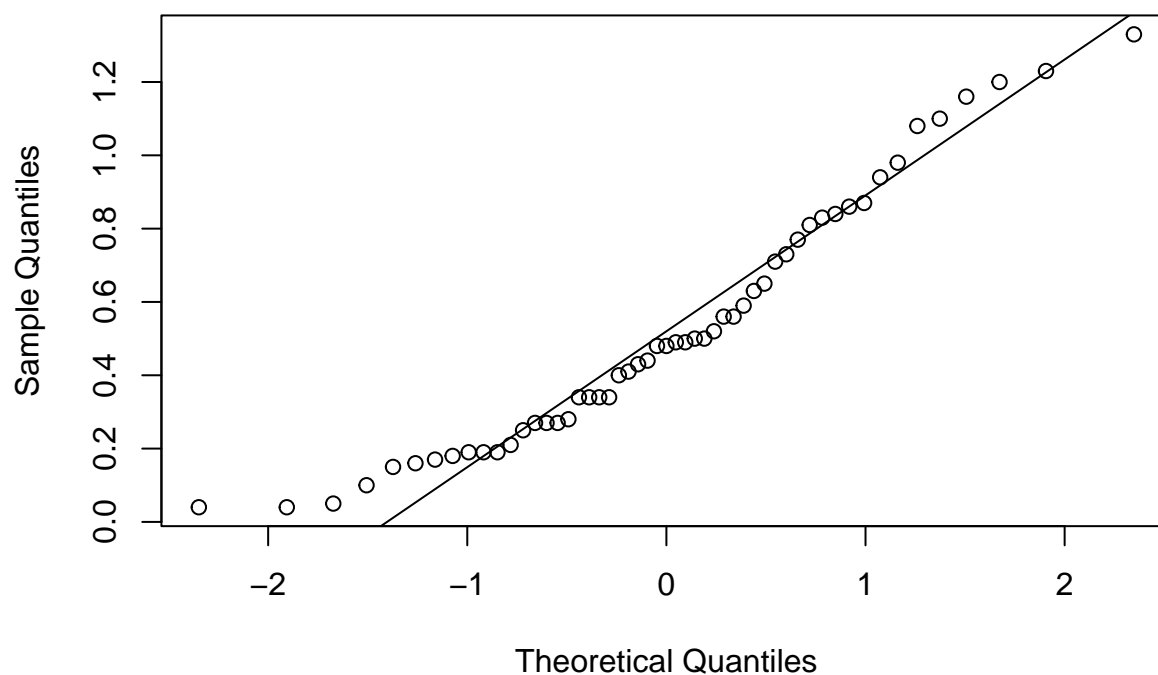
Histograma de Clorofila



Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una asimetría positiva con un sesgo a la derecha, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran recargados hacia el lado izquierdo, lo que significa que la distribución se encuentra sesgada a la derecha.

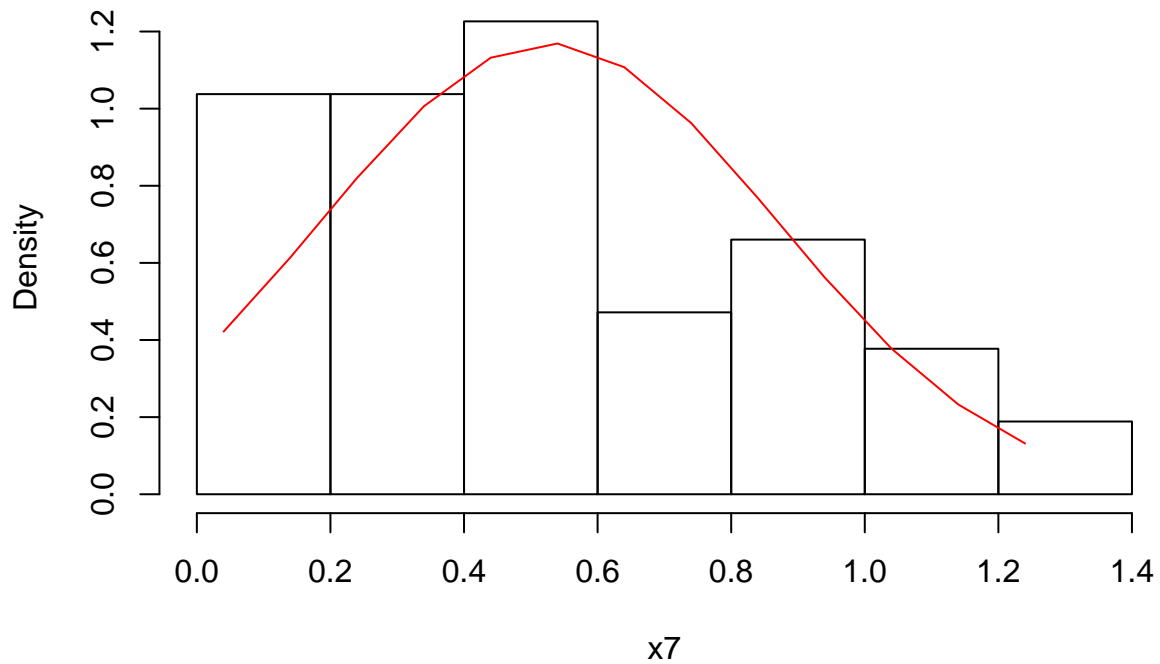
```
# x7
# QQplot
qqnorm(x7, main = "Normal Q-Q Plot Concentración Media de Mercurio")
qqline(x7)
```

Normal Q-Q Plot Concentración Media de Mercurio



```
# Histograma  
hist(x7, main = "Histograma de Concentración Media de Mercurio", prob = TRUE, col = 0)  
x = seq(min(x7), max(x7), 0.1)  
y = dnorm(x, mean(x7), sd(x7))  
lines(x, y, col = "red")
```

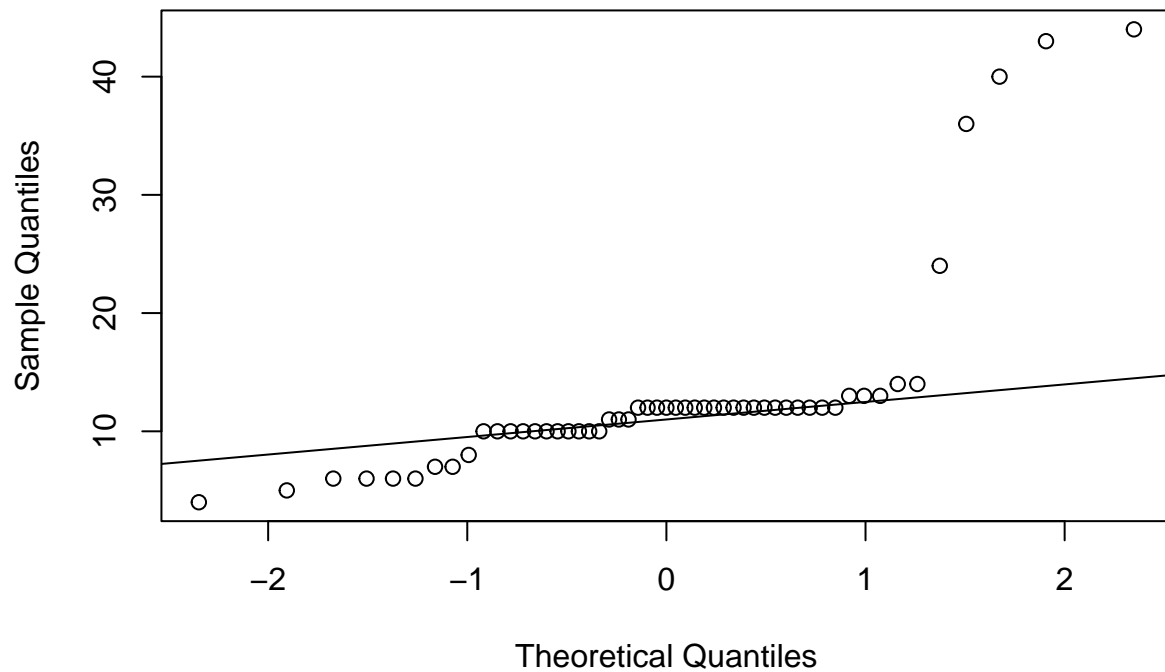

Histograma de Concentración Media de Mercurio



Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una asimetría positiva con un sesgo ligeramente a la derecha, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran mínimamente recargados hacia el lado izquierdo, lo que significa que la distribución se encuentra sesgada a la derecha.

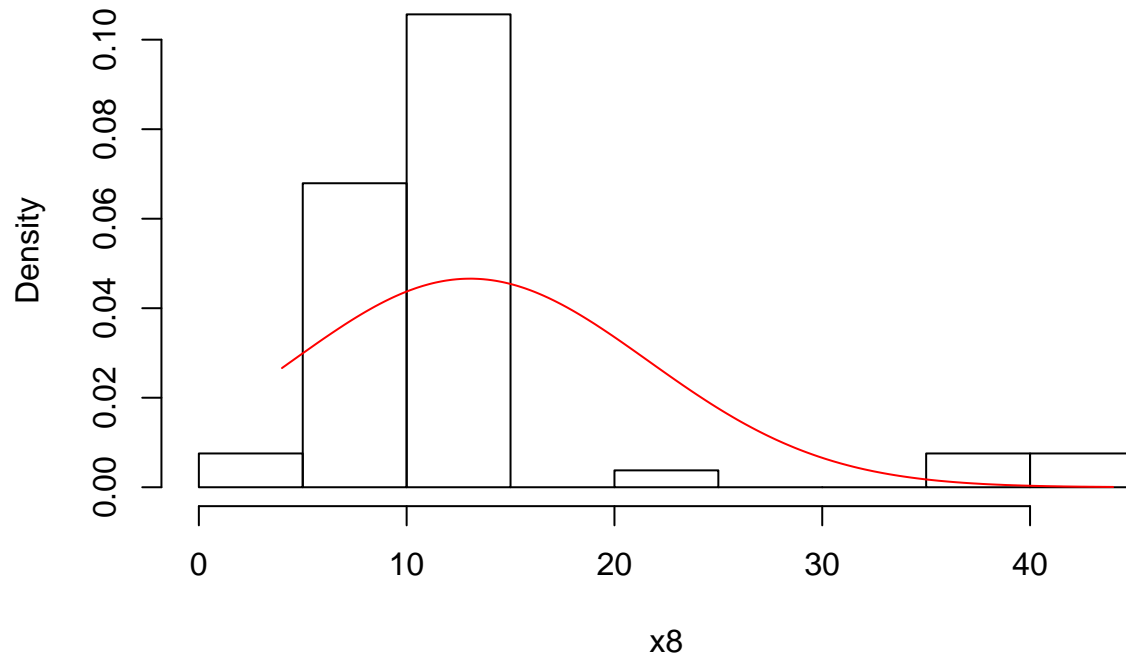
```
# x8
# QQplot
qqnorm(x8, main = "Normal Q-Q Plot Número de Peces Estudiados")
qqline(x8)
```

Normal Q-Q Plot Número de Peces Estudiados



```
# Histograma
hist(x8, main = "Histograma de Número de Peces Estudiados", prob = TRUE, col = 0)
x = seq(min(x8), max(x8), 0.1)
y = dnorm(x, mean(x8), sd(x8))
lines(x, y, col = "red")
```

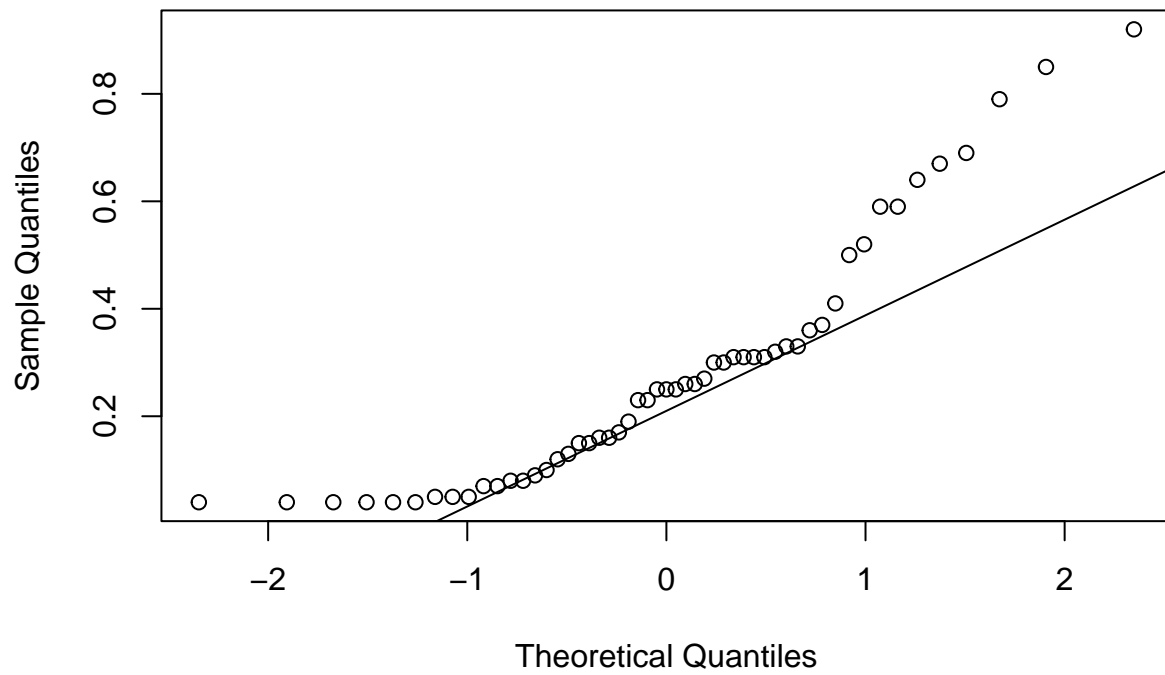
Histograma de Número de Peces Estudiados



Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una distribución con colas gruesas (baja curtosis, distribución platicúrtica), lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos son asimétricos con un ligero sesgo a la derecha.

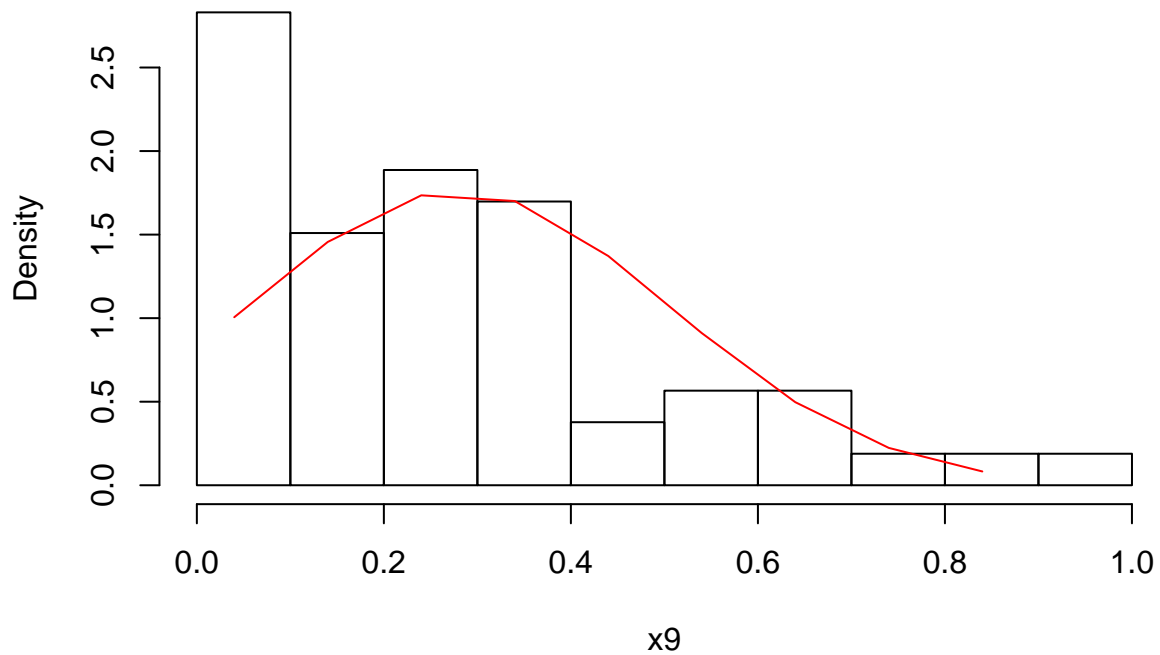
```
# x9
# qqplot
qqnorm(x9, main = "Normal Q-Q Plot Mínimo de la Concentración de Mercurio")
qqline(x9)
```

Normal Q-Q Plot Mínimo de la Concentración de Mercurio



```
# Histograma  
hist(x9, main = "Histograma de Mínimo de la Concentración de Mercurio", prob = TRUE, col = 0)  
x = seq(min(x9), max(x9), 0.1)  
y = dnorm(x, mean(x9), sd(x9))  
lines(x, y, col = "red")
```

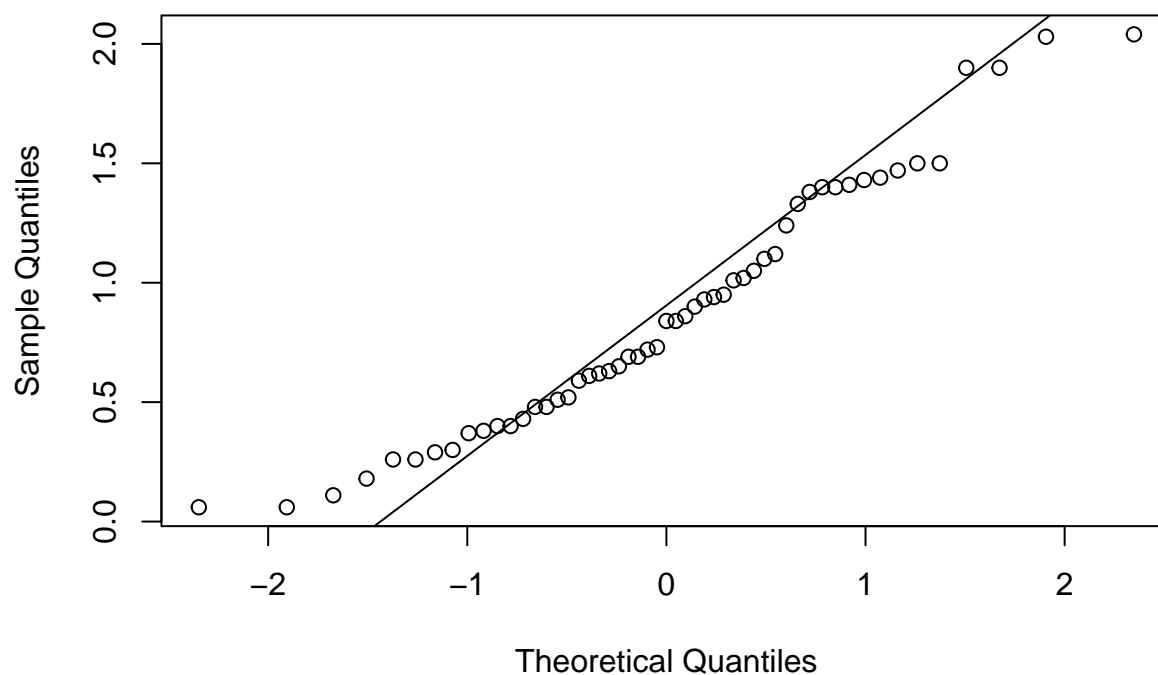
Histograma de Mínimo de la Concentración de Mercurio



Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una asimetría positiva con un sesgo a la derecha, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran recargados hacia el lado izquierdo, lo que significa que la distribución se encuentra sesgada a la derecha.

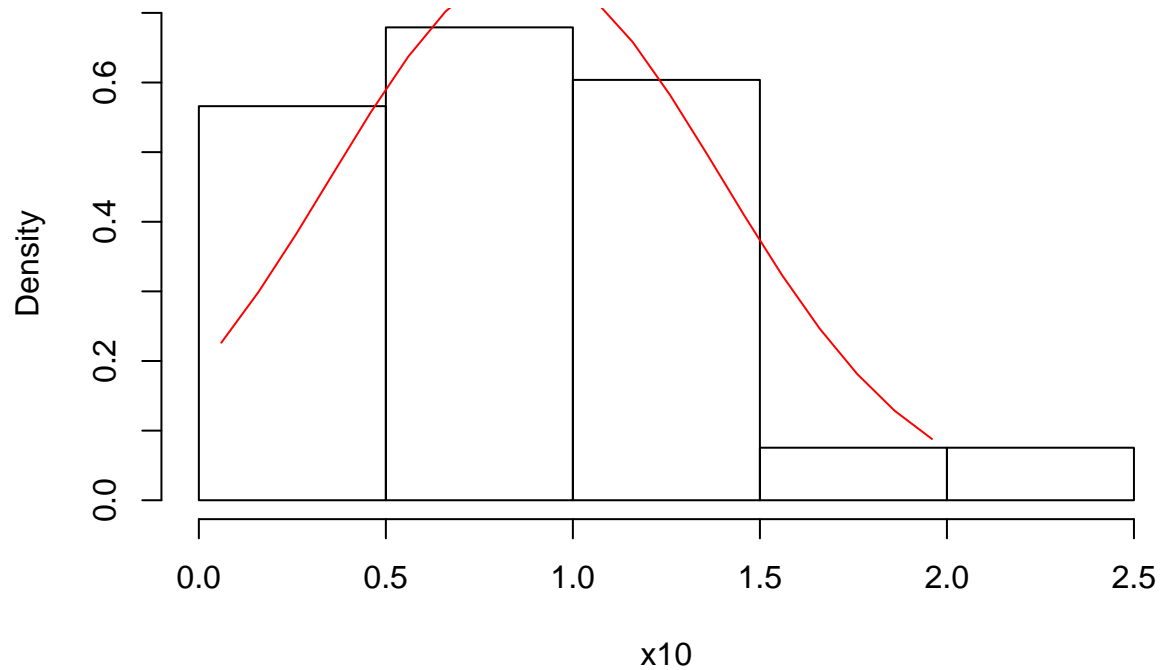
```
# x10
# QQplot
qqnorm(x10, main = "Normal Q-Q Plot Máximo de la Concentración de Mercurio")
qqline(x10)
```

Normal Q-Q Plot Máximo de la Concentración de Mercurio



```
# Histograma  
hist(x10, main = "Histograma de Máximo de la Concentración de Mercurio", prob = TRUE, col = 0)  
x = seq(min(x10), max(x10), 0.1)  
y = dnorm(x, mean(x10), sd(x10))  
lines(x, y, col = "red")
```

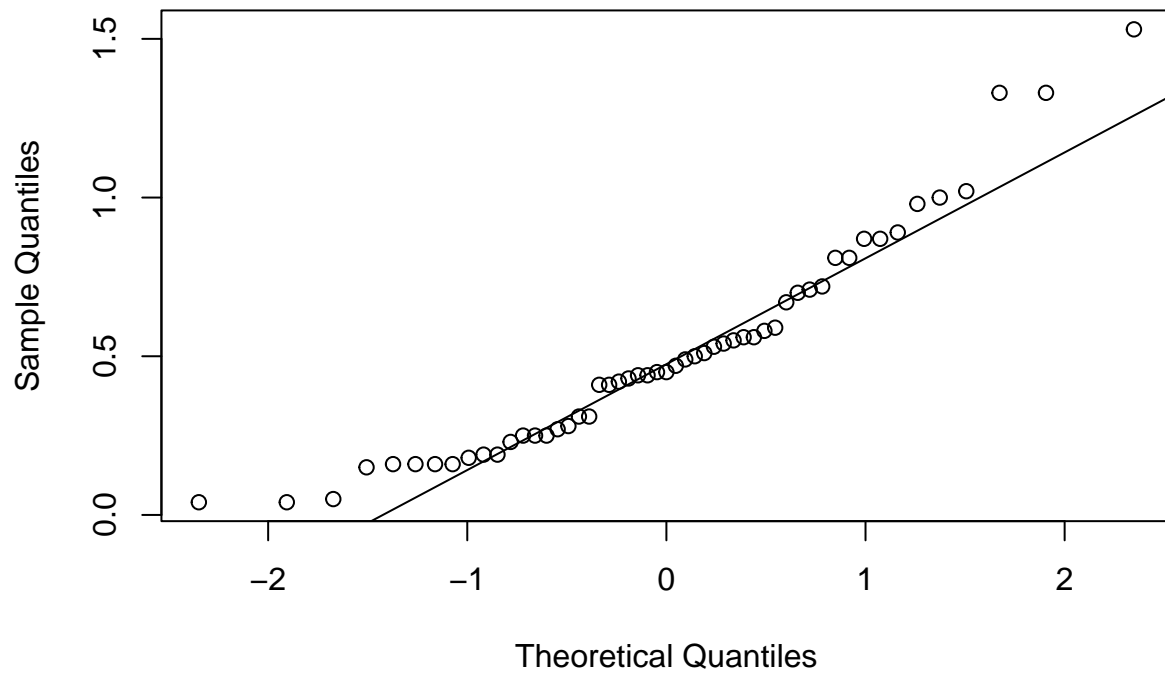
Histograma de Máximo de la Concentración de Mercurio



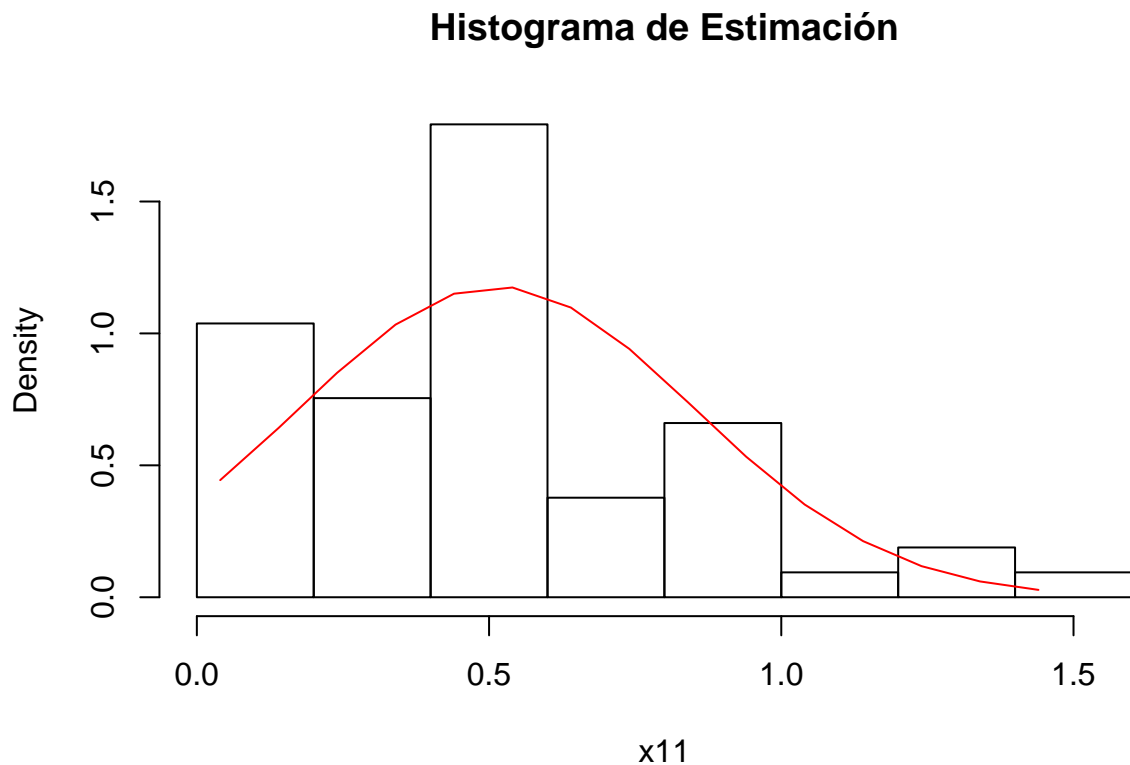
Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal es casi ideal, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran recargados casi simétricos.

```
# x11
# QQplot
qqnorm(x11, main = "Normal Q-Q Plot Estimación")
qqline(x11)
```

Normal Q-Q Plot Estimación



```
# Histograma
hist(x11, main = "Histograma de Estimación", prob = TRUE, col = 0)
x = seq(min(x11), max(x11), 0.1)
y = dnorm(x, mean(x11), sd(x11))
lines(x, y, col = "red")
```

Como podemos observar en la gráfica de qqplot, tenemos que la probabilidad normal tiene una asimetría positiva con un sesgo a la derecha, lo cual se comprueba con el histograma y podemos verificar que, en efecto, los datos se encuentran recargados hacia el lado izquierdo, lo que significa que la distribución se encuentra sesgada a la derecha.

Variables Cualitativas

Tabla de Distribución de Frecuencia Frecuencia de la Edad de Los Peces

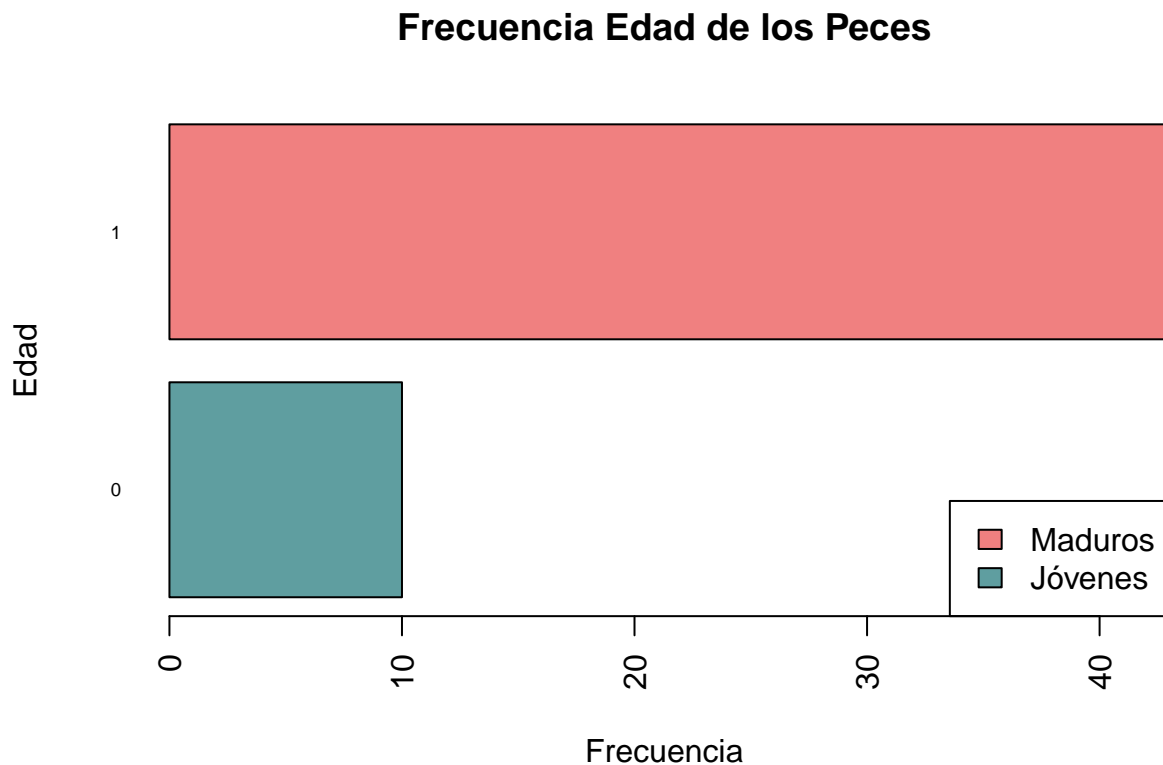
```
edad_peces_table = table(db_mercurio_num$X12)
print("Tabla de Distribución de Frecuencia Edad de los Peces: ")
```

```
## [1] "Tabla de Distribución de Frecuencia Edad de los Peces: "
```

```
edad_peces_table
```

```
##
##  0  1
## 10 43
```

```
# Gráfica de Frecuencia
barplot(edad_peces_table, width = 1, cex.names = 0.6, col = c("cadetblue", "lightcoral"), main = "Frecu
```



De acuerdo con la gráfica anterior podemos observar que en el dataset utilizado tenemos una mayor cantidad de peces de una edad madura que peces de una edad joven, lo cual es importante a tomar en cuenta si existe la posibilidad de que la concentración de mercurio varíe con respecto a la edad de los peces.

Frecuencia de Lagos que Superaron los 0.5mgdeHg/Kg Establecidos por los Reglamentos

```
db_mercurio_cnt = db_mercurio_num
db_mercurio_cnt$X13 <- with(db_mercurio_cnt, ifelse(X7 > 0.5, 1, 0))
db_mercurio_cnt_table = table(db_mercurio_cnt$X13)
print("Tabla de Distribución de Lagos que Superaron los 0.5 mg de Hg/Kg: ")
```

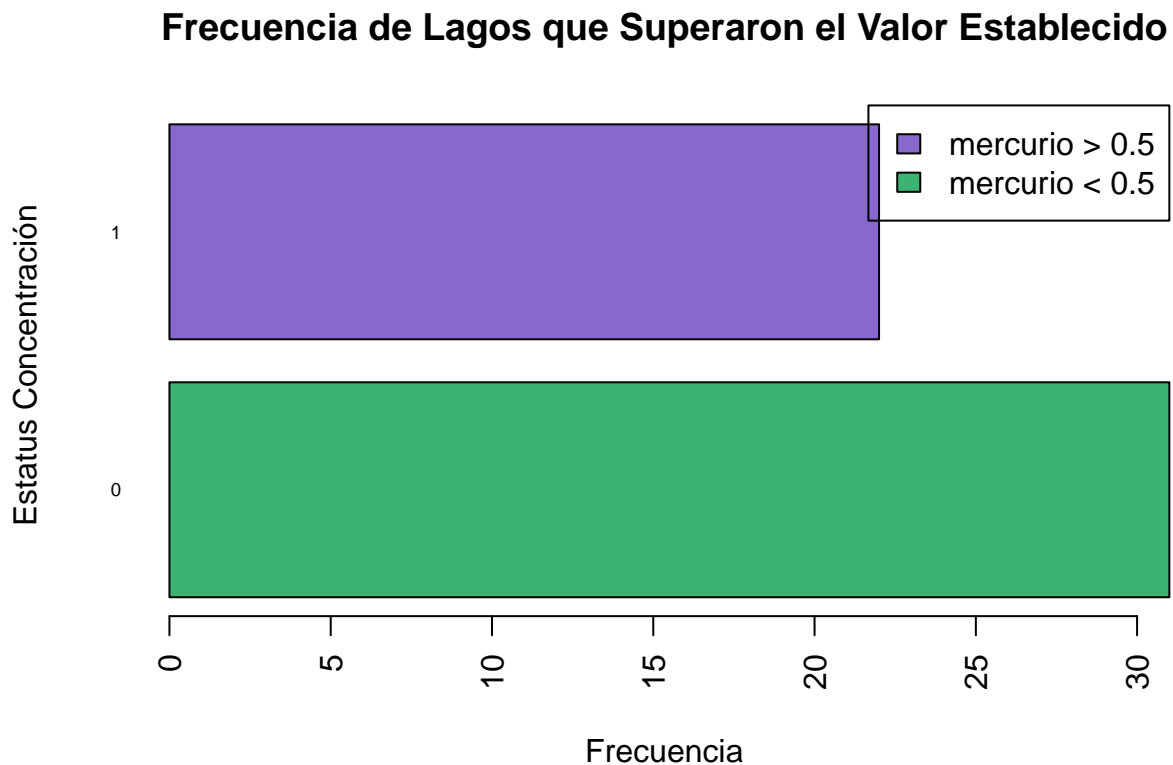
```
## [1] "Tabla de Distribución de Lagos que Superaron los 0.5 mg de Hg/Kg: "
```

```
db_mercurio_cnt_table
```

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

```
# Gráfica de Frecuencia
```

```
barplot(db_mercurio_cnt_table, width = 1, cex.names = 0.6, col = c("mediumseagreen", "mediumpurple3"),
```



El gráfico anterior se realizó creando una nueva columna tomando el valor permitido de concentración y mostrando así si cada río lo sobrepasaba o se mantenía debajo del mismo, obteniendo así que aproximadamente el 58% de los ríos cuenta con una concentración media de mercurio menor al 0.5 permitido por los reglamentos, lo que indica que una parte significativa se encuentra contaminada y es probable que sea dañina para el ser humano.

Porcentajes y Distribución de PH

```
db_mercurio_ph = db_mercurio_num
db_mercurio_ph$X4 <- with(db_mercurio_ph, ifelse(X4 < 7.0, "Ácido", ifelse(X4 == 7.0, "Neutro", "Alcalino")))
db_ph_cnt_table = table(db_mercurio_ph$X4)
print("Tabla de Distribución del PH: ")
```

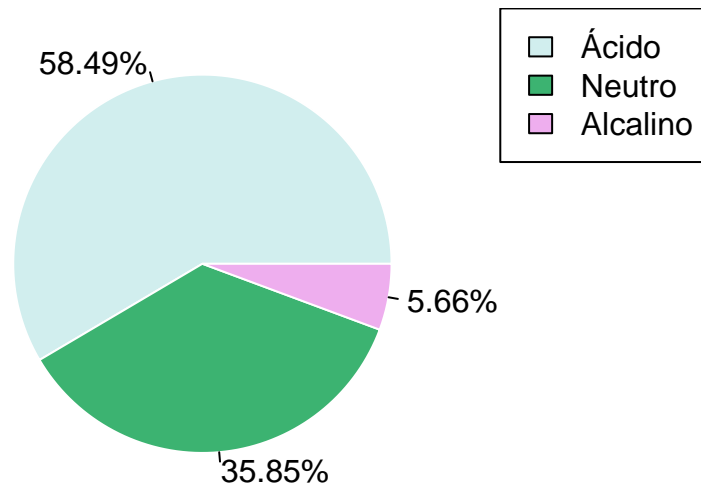
```
## [1] "Tabla de Distribución del PH: "
```

```
db_ph_cnt_table
```

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

```
#Gráfica de pie
colors <- c("lightcyan2", "mediumseagreen", "plum2")
pie(db_ph_cnt_table, border="white", col = colors, main = "Gráfica del PH", labels = paste0(round(100 *
legend("topright", c("Ácido", "Neutro", "Alcalino"), fill=colors)
```

Gráfica del PH



En este gráfico de pastel podemos observar que existe un mayor porcentaje de ácidos de acuerdo a los registros de ph, lo cual indica que, en efecto, existe contaminación en los lagos, sin embargo también se tiene que un 35.85% el ph es neutro, un 5.66% es alcalino y lo restante es ácido.

Búsqueda de Correlaciones

```
library(psych)
Rc = corr.test(db_mercurio_num)
Rc
```

```
## Call:corr.test(x = db_mercurio_num)
## Correlation matrix
##      X3      X4      X5      X6      X7      X8      X9      X10     X11     X12
## X3  1.00  0.72  0.83  0.48 -0.59  0.01 -0.53 -0.60 -0.63 -0.09
## X4  0.72  1.00  0.58  0.61 -0.58 -0.02 -0.54 -0.55 -0.61  0.04
## X5  0.83  0.58  1.00  0.41 -0.40 -0.09 -0.33 -0.41 -0.46  0.00
## X6  0.48  0.61  0.41  1.00 -0.49 -0.01 -0.40 -0.48 -0.51 -0.28
## X7 -0.59 -0.58 -0.40 -0.49  1.00  0.08  0.93  0.92  0.96  0.11
## X8  0.01 -0.02 -0.09 -0.01  0.08  1.00 -0.08  0.16  0.03  0.21
## X9 -0.53 -0.54 -0.33 -0.40  0.93 -0.08  1.00  0.77  0.92  0.10
## X10 -0.60 -0.55 -0.41 -0.48  0.92  0.16  0.77  1.00  0.86  0.09
## X11 -0.63 -0.61 -0.46 -0.51  0.96  0.03  0.92  0.86  1.00  0.09
## X12 -0.09  0.04  0.00 -0.28  0.11  0.21  0.10  0.09  0.09  1.00
## Sample Size
```

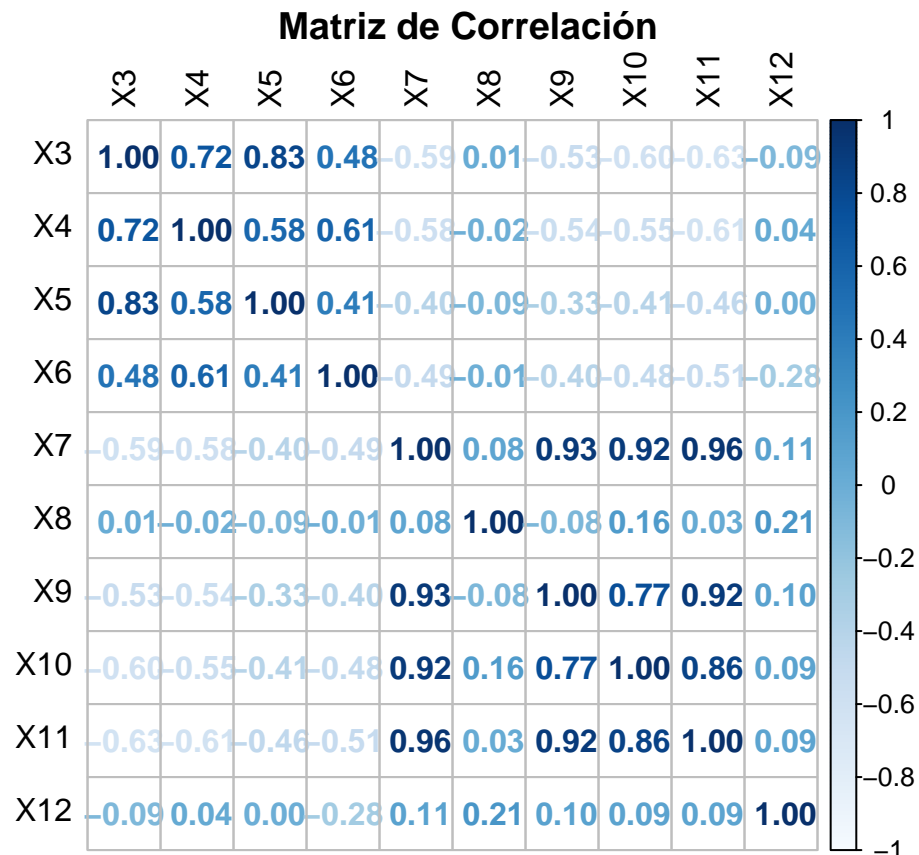
```
## [1] 53
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##      X3  X4  X5  X6  X7  X8  X9  X10  X11  X12
## X3  0.00 0.00 0.00 0.01 0.00 1.00 0.00 0.00 0.00 1.00
## X4  0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 1.00
## X5  0.00 0.00 0.00 0.05 0.06 1.00 0.27 0.05 0.01 1.00
## X6  0.00 0.00 0.00 0.00 0.00 1.00 0.06 0.01 0.00 0.68
## X7  0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 1.00
## X8  0.94 0.89 0.52 0.93 0.57 0.00 1.00 1.00 1.00 1.00
## X9  0.00 0.00 0.01 0.00 0.00 0.56 0.00 0.00 0.00 1.00
## X10 0.00 0.00 0.00 0.00 0.00 0.25 0.00 0.00 0.00 1.00
## X11 0.00 0.00 0.00 0.00 0.00 0.85 0.00 0.00 0.00 1.00
## X12 0.50 0.79 0.99 0.04 0.44 0.14 0.47 0.50 0.52 0.00
##
## To see confidence intervals of the correlations, print with the short=FALSE option
```

Visualización de la Matriz de Correlación

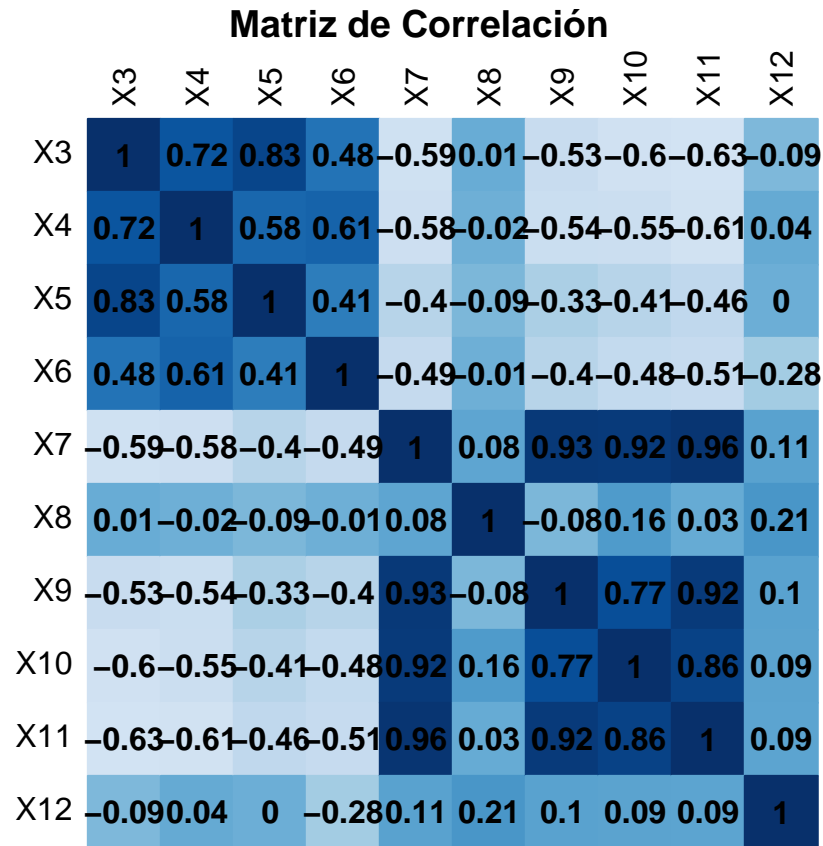
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(db_mercurio_num), method = "number", col = COL1("Blues"), tl.col = "black", main = "Matriz
```



```
corrplot(cor(db_mercurio_num), method = "color", col = COL1("Blues"), addCoef.col = "black", cl.pos = "t")
```



Las figuras anteriores muestran la matriz de correlaciones de los valores numéricos de nuestro dataset, en estas podemos observar qué variables se encuentran mayormente relacionadas entre sí para al momento de implementar el modelo estadístico, se elijan las variables correctas para tener mucho mejores resultados.

De acuerdo con la matriz, podemos observar que la variable x7 (concentración media de mercurio) está fuertemente correlacionada positivamente con las variables x9 (mínima concentración de mercurio), x10 (máxima concentración de mercurio), y x11 (estimación), sin embargo, al tener un valor tan fuerte de correlación, es posible que se presente el problema de multicolinealidad; este problema puede dificultar la interpretación de los resultados del modelo de regresión, de hecho, una de las formas más sencillas de detectar un posible problema de este tipo es observar la matriz de correlación y buscar que variables se encuentran altamente relacionadas entre sí.

Debido a este problema de multicolinealidad se decidió utilizar otras variables para el modelo de regresión, tomando en cuenta que estas tienen una correlación que va de débil a moderada con respecto a la variable x7 (concentración media de mercurio), dichas variables a utilizar en el modelo de regresión son:

- $X3$ = 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

Implementación de Herramientas Estadísticas

Para la solución de este problema, se decidió utilizar como herramientas estadísticas el ANOVA (Análisis de Varianza) para determinar si las discrepancias entre las medias de los tratamientos son mayores de lo que podría esperarse de las variaciones dentro de los tratamientos; y como segunda herramienta, se decidió implementar la regresión múltiple donde se genera un modelo en el que el valor de la variable dependiente es determinado a partir de un conjunto de variables independientes.

ANOVA

Para el análisis de varianza (ANOVA), se busca encontrar si existe alguna diferencia significativa entre el nivel de concentración media del mercurio dependiendo si se está analizando un pez joven o maduro, pero también existe la opción de comparar en vez de la edad del pez, la variable nueva que se creó sobre si la concentración de mercurio sobrepasa o no el valor permitido de $0.5mg \frac{Hg}{kg}$.

Es importante mencionar que se trabajará con un nivel de significancia de 0.05.

Usando la Edad de los Peces

```
# Edad
media_mercurio_j = db_mercurio_num[db_mercurio_num$X12 == 0, ]$X7
media_mercurio_m = db_mercurio_num[db_mercurio_num$X12 == 1, ]$X7

print("Jóvenes")

## [1] "Jóvenes"

media_mercurio_j

## [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"

media_mercurio_m

## [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

media_mercurio = c(media_mercurio_j, media_mercurio_m)
media_mercurio

## [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
```

```

len_j = length(media_mercurio_j)
len_m = length(media_mercurio_m)

edad = c(rep("J", len_j), rep("M", len_m))
edad = factor(edad)
edad

## [1] J J J J J J J J J M M M M M M M M M M M M M M M M M M M M M M M M M M M M
## [39] M M M M M M M M M M M M M M M M M
## Levels: J M

A <- aov(media_mercurio ~ 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

```

Al hacer el análisis de la varianza, podemos observar que el valor $pvalue = 0.438$ no es significativo ya que este es mayor a 0.05 (nivel de significancia), lo que significa que no se rechaza la hipótesis nula y se concluye que no se tiene la suficiente evidencia para decir que existe una diferencia estadística significativa entre las medias de concentración de mercurio y las edades de los peces, por lo tanto, no se puede seguir con el análisis pues no es posible determinar qué grupos se diferencian de cada uno.

A pesar de ello, aún podemos hacer el análisis utilizando nuestra variable que indica si la media se sobrepasa o no de los valores permitidos.

Utilizando la Variable del Nivel de Concentración (Mayor = 1, Menor = 0)

```

# Edad
media_mercurio_menor = db_mercurio_cnt[db_mercurio_cnt$X13 == 0, ]$X7
media_mercurio_mayor = db_mercurio_cnt[db_mercurio_cnt$X13 == 1, ]$X7

print("Menores")

## [1] "Menores"

media_mercurio_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("Mayores")

## [1] "Mayores"

```



```
media_mercurio_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
```

```
media_mercurio = c(media_mercurio_menor, media_mercurio_mayor)
media_mercurio
```

```
## [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
```

```
len_menor = length(media_mercurio_menor)
len_mayor = length(media_mercurio_mayor)
```

```
nivel_concentracion = c(rep("Menor", len_menor), rep("Mayor", len_mayor))
nivel_concentracion = factor(nivel_concentracion)
nivel_concentracion
```

```
## [1] Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor
## [13] Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor Menor
## [25] Menor Menor Menor Menor Menor Menor Menor Menor Mayor Mayor Mayor Mayor Mayor
## [37] Mayor Mayor Mayor Mayor Mayor Mayor Mayor Mayor Mayor Mayor Mayor Mayor Mayor
## [49] Mayor Mayor Mayor Mayor Mayor
## Levels: Mayor Menor
```

```
A <- aov(media_mercurio ~ nivel_concentracion)
summary(A)
```

```
##               Df Sum Sq Mean Sq F value    Pr(>F)
## nivel_concentracion  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
```

Como podemos observar, con esta variable sí se rechaza la hipótesis nula, donde se dice que cada las medias de los grupos son iguales y por ende, es posible generar los análisis posteriores para determinar y entender el comportamiento de esta variación.

```
m = tapply(media_mercurio, nivel_concentracion, mean)
s = tapply(media_mercurio, nivel_concentracion, sd)
n = tapply(media_mercurio, nivel_concentracion, length)

print("Medias de los tratamientos:")
```

```
## [1] "Medias de los tratamientos:"
```

```
m
```

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

```
print("Desviación estándar de los tratamientos:")
```

```
## [1] "Desviación estándar de los tratamientos:"
```

```
s
```

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

```
print("Tamaño de la muestra de los tratamientos")
```

```
## [1] "Tamaño de la muestra de los tratamientos"
```

```
n
```

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

Intervalos de Confianza

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

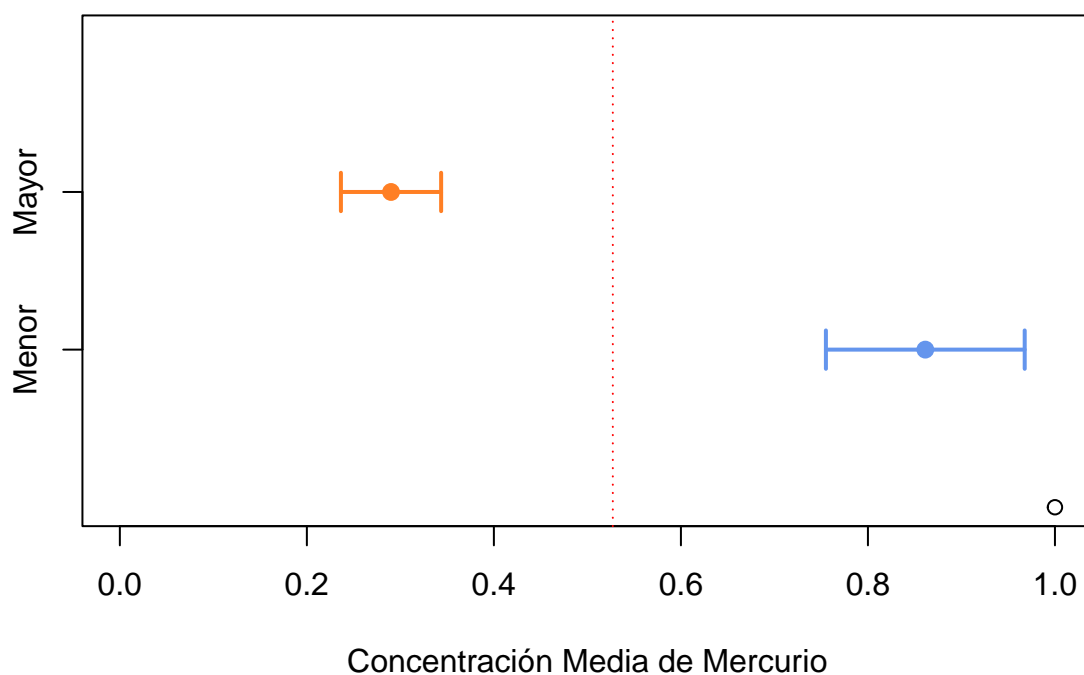
```
##      Mayor      Menor
## 0.7550654 0.2364250
```

```
Sup
```

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

```
plot(0, ylim = c(0,3), xlim = c(0, 1), yaxt = "n", ylab = "", xlab = "Concentración Media de Mercurio",
axis(2, at = c(1:2), labels = c("Menor", "Mayor"))
colores = c("cornflowerblue", "chocolate1")
for(i in 1:2) {
  arrows(In[i], i, Sup[i], i, angle = 90, code = 3, length = 0.1, lwd = 2, col = colores[i])
  points(m[i], i, pch = 19, cex = 1.1, col = colores[i])
}
abline(v = mean(media_mercurio), lty = 3, col = "red")
```

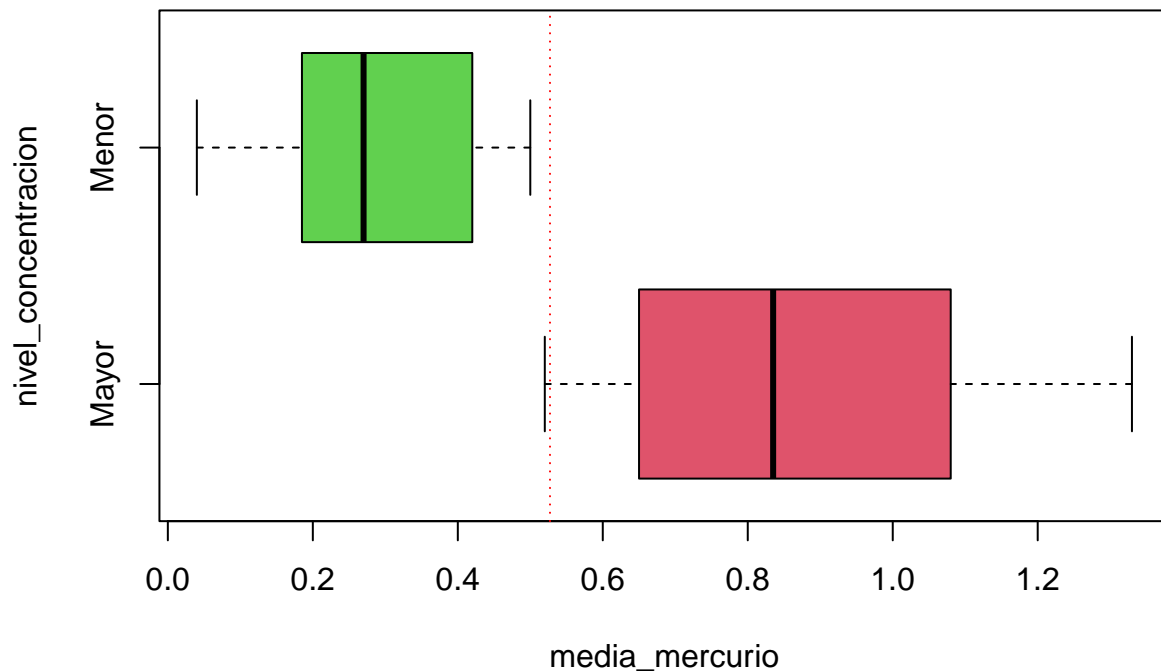
Concentración de Concentración Mayor o Menor



Al calcular los intervalos de confianza, obtenemos que los valores los valores son:

- Para Menor el intervalo va de 0.2364250 a 0.3435750
- Para Mayor el intervalo va de 0.7550654 a 0.9676619

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



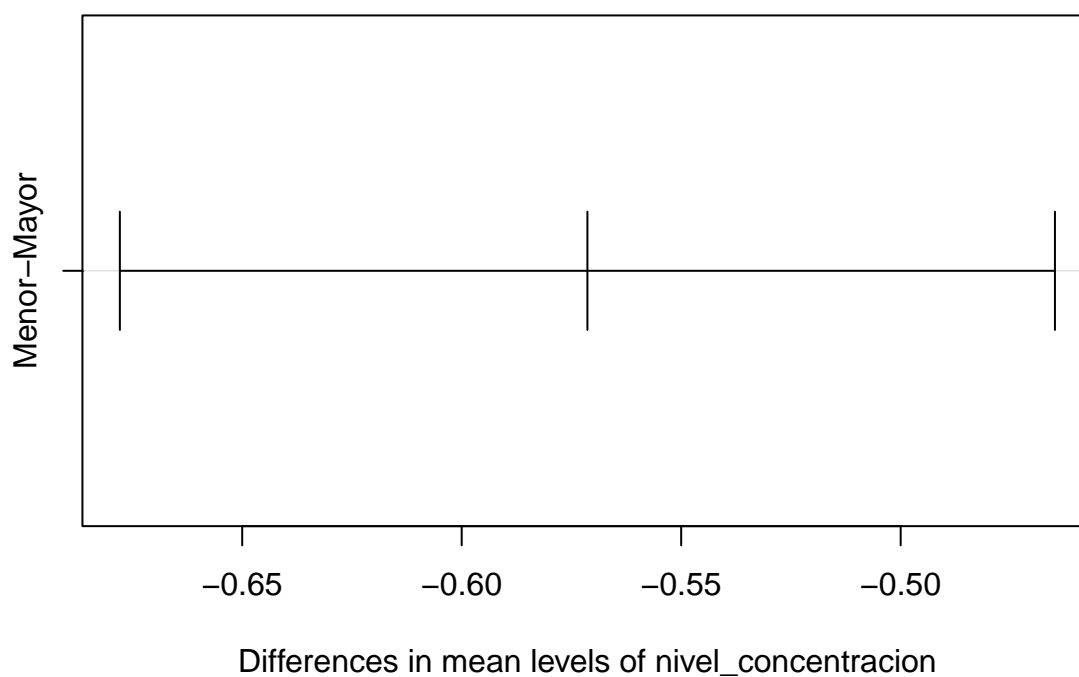
Cómo podemos observar en el boxplot, ambas muestras no coinciden y son diferentes de la concentración media de mercurio, lo que confirma que, en efecto, las medias de los tratamientos no son iguales y por lo tanto, se tiene que si existe una diferencia entre los lagos que sobrepasan el valor reglamentado de 0.5 y los que se mantienen por debajo.

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

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

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

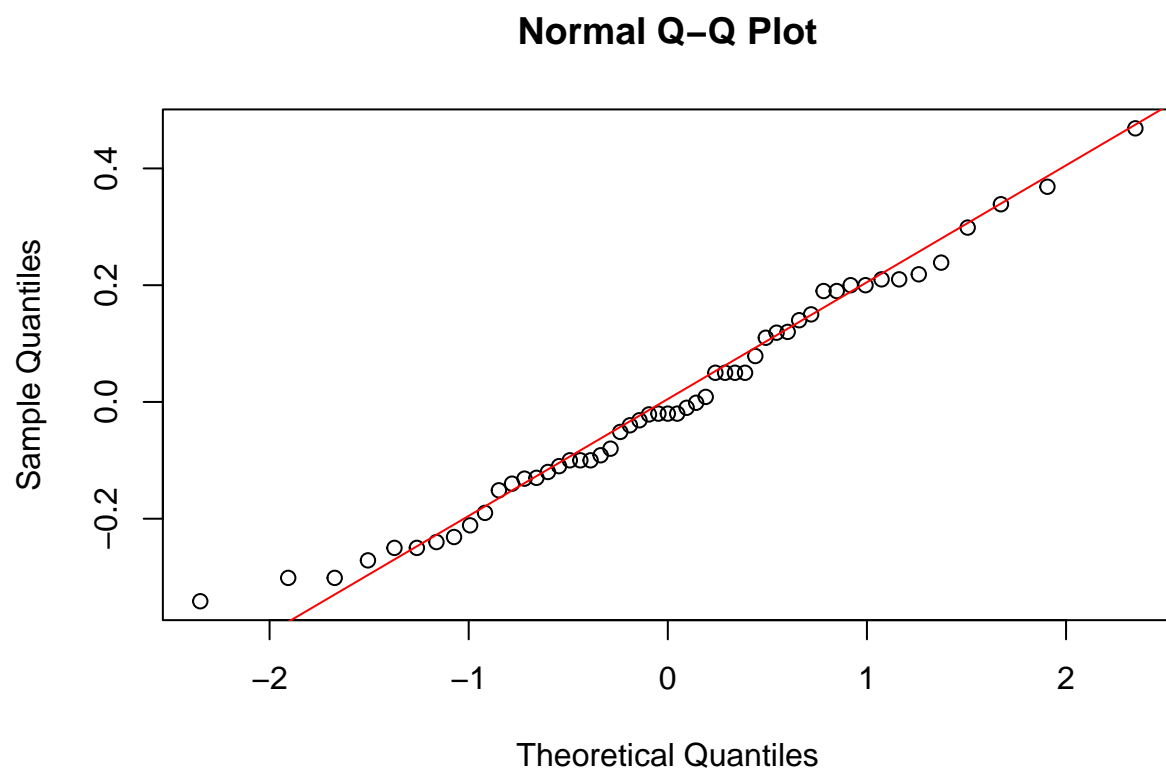
95% family-wise confidence level



La prueba de Tukey nos muestra que los intervalos de confianza de las diferencias por pares no incluyen al 0, entonces se confirma que la hipótesis nula se rechaza.

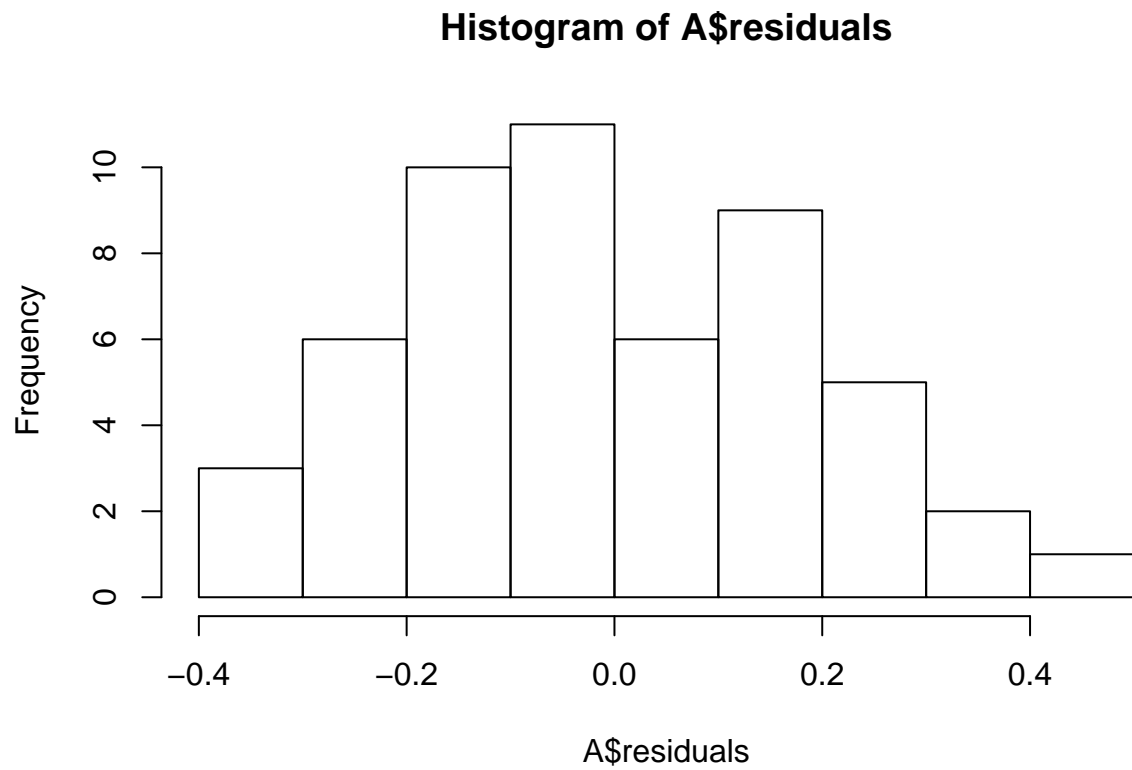
Verificación de supuestos:

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



Normalidad

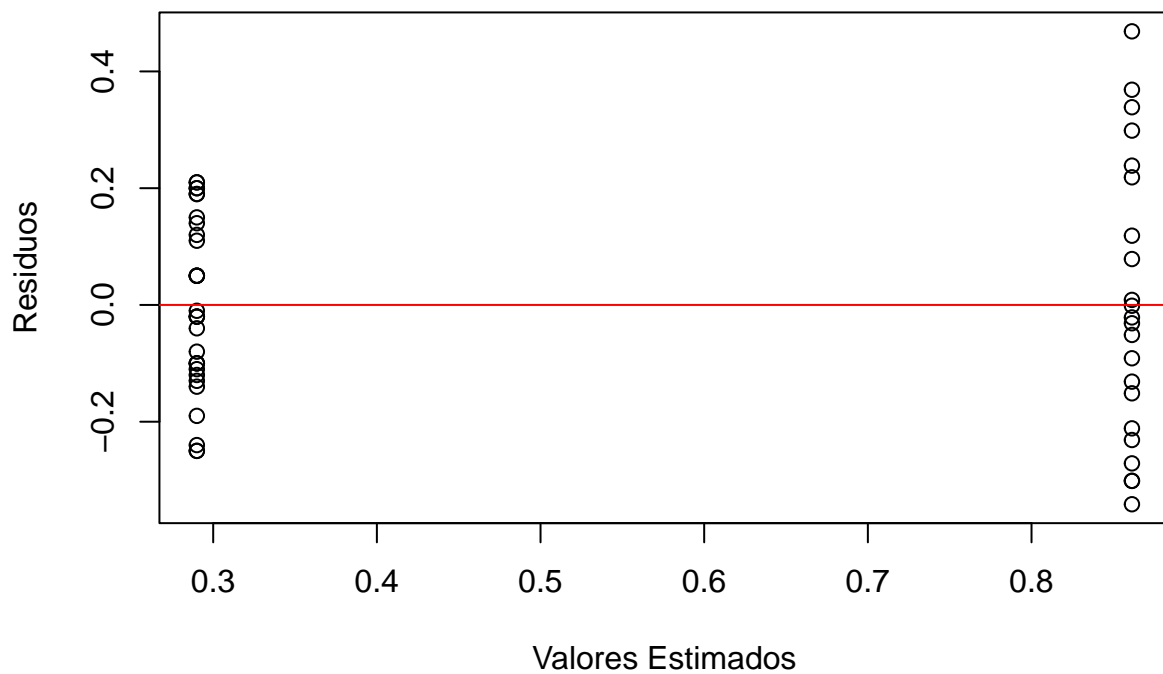
```
hist(A$residuals, col = 0)
```



De acuerdo con la gráfica de normalidad podemos observar que esta tiene un comportamiento ideal en su simetría.

Homocedasticidad Valores estimados vs residuos.

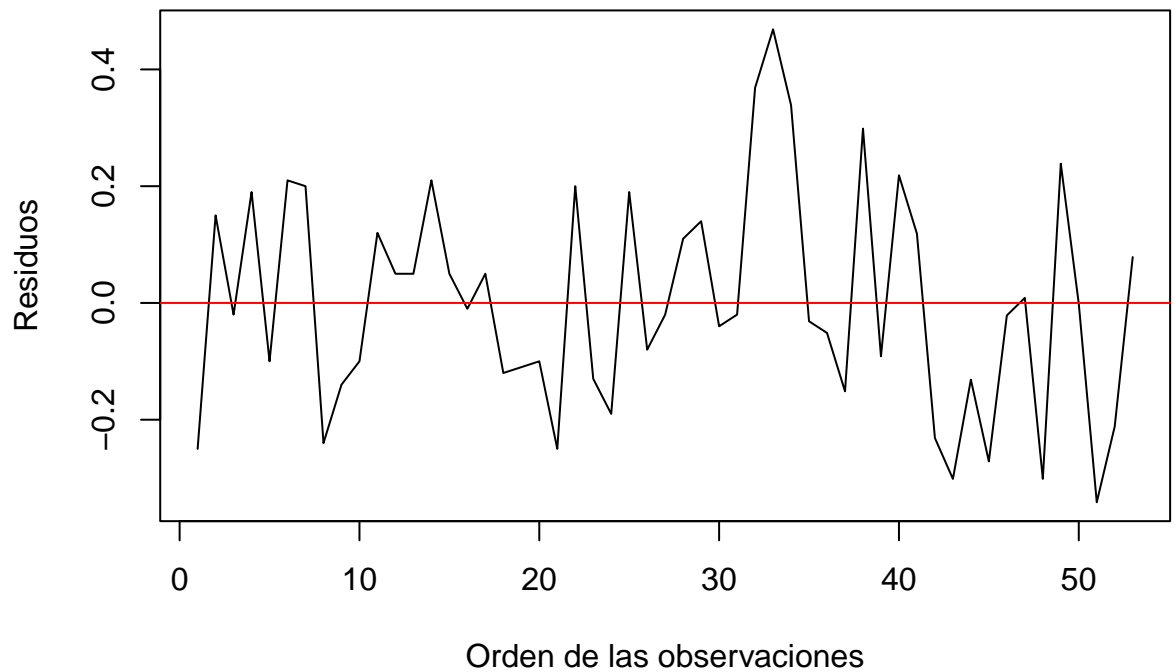
```
plot(A$fitted.values, A$residuals, ylab = "Residuos", xlab = "Valores Estimados")  
abline(h = 0, col = "red")
```



La gráfica de estimados y residuos muestra que efectivamente cumple con los supuestos.

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


Errores vs Orden de Observación



Independencia

La gráfica de independencia muestra una autocorrelación negativa y se observa una alternancia muy marcada de residuos positivos y negativos, lo cual puede ser causado por la existencia de ciclos en los errores o relaciones no lineales.

Regresión Múltiple

```
db_mercurio_mult = db_mercurio_num[1:5]
```

Correlación

```
library(Hmisc)
```

```
## 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(db_mercurio_mult))
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
```

```
##
```

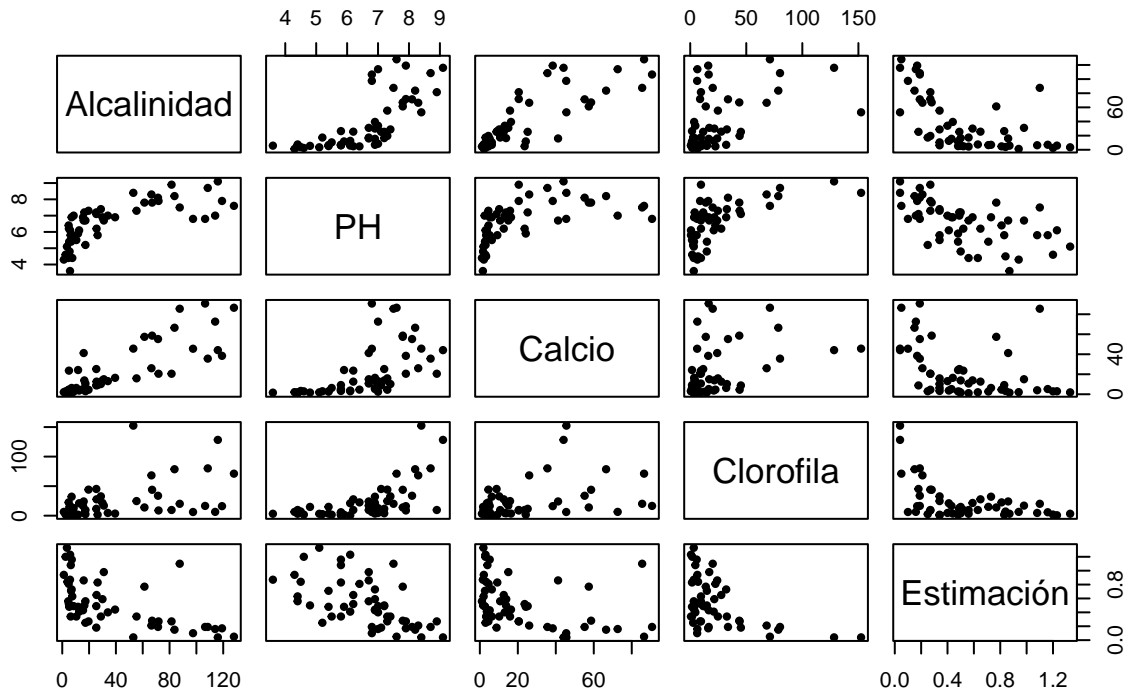
```
##
```

```
## 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(db_mercurio_mult, labels=c("Alcalinidad", "PH", "Calcio", "Clorofila", "Estimación"), main = "Mat.
```

Matriz de Dispersión



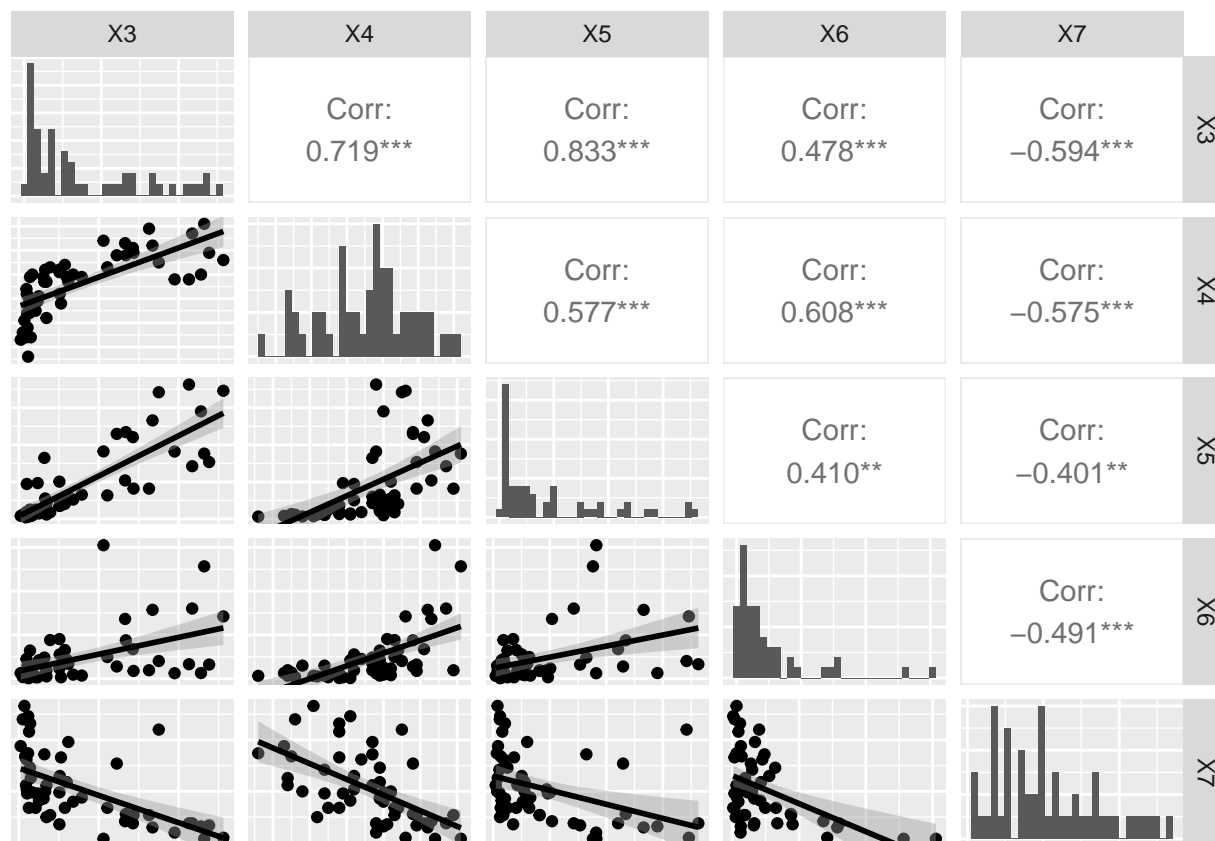
```
library(GGally)
```

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

```
ggpairs(db_mercurio_mult, 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'.
```

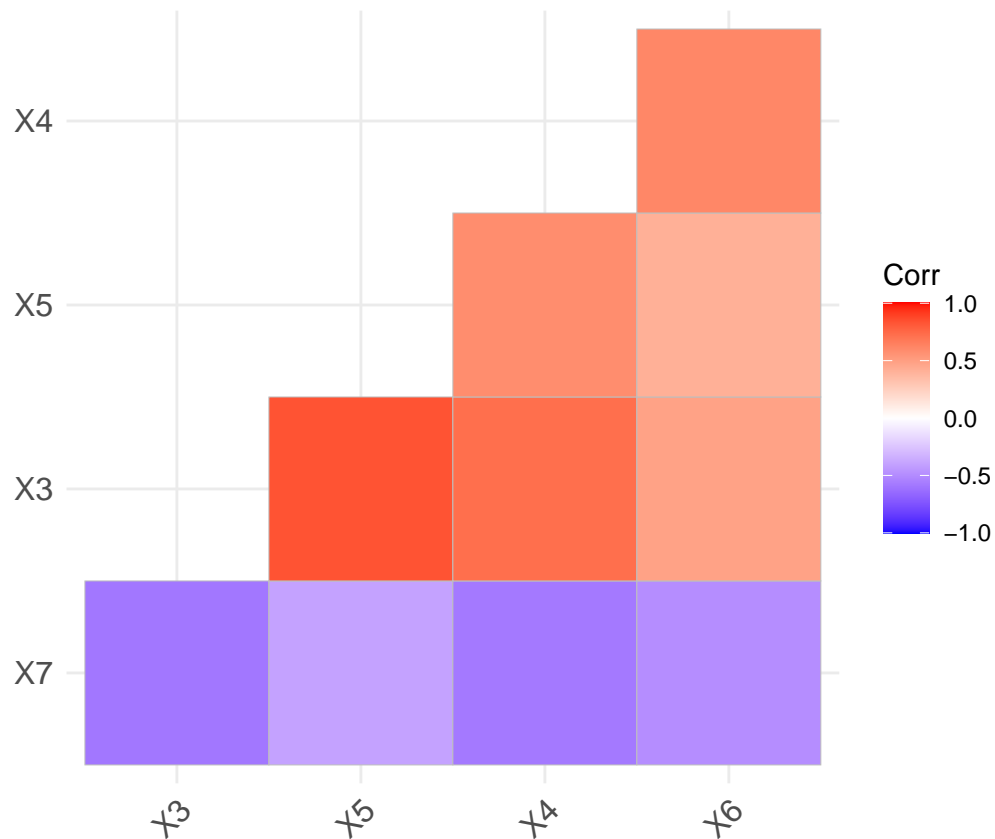


```
library(ggcorrplot)
library(polycor)
```

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

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

```
mat_cor <- hetcor(db_mercurio_mult)$correlations
ggcorrplot(mat_cor, type="lower", hc.order = T)
```



El Modelo

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

```
##
## Call:
## lm(formula = X7 ~ X3 + X4 + X5 + X6, data = db_mercurio_mult)
##
## 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 = db_mercurio_mult)
##
## Coefficients:
## (Intercept)          X3          X5          X6
##   0.744583   -0.006487   0.004333  -0.003035
```

El mejor modelo

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

```
##
## Call:
## lm(formula = X7 ~ X3 + X5 + X6, data = db_mercurio_mult)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38746 -0.18520 -0.07092  0.14490  0.61422
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.744583   0.052401  14.209 < 2e-16 ***
## X3          -0.006487   0.001790  -3.624 0.000689 ***
## X5           0.004333   0.002642   1.640 0.107445
## X6          -0.003035   0.001348  -2.252 0.028862 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.263 on 49 degrees of freedom
## Multiple R-squared:  0.4394, Adjusted R-squared:  0.4051
## F-statistic: 12.8 on 3 and 49 DF,  p-value: 2.676e-06
```

Intervalos de confianza

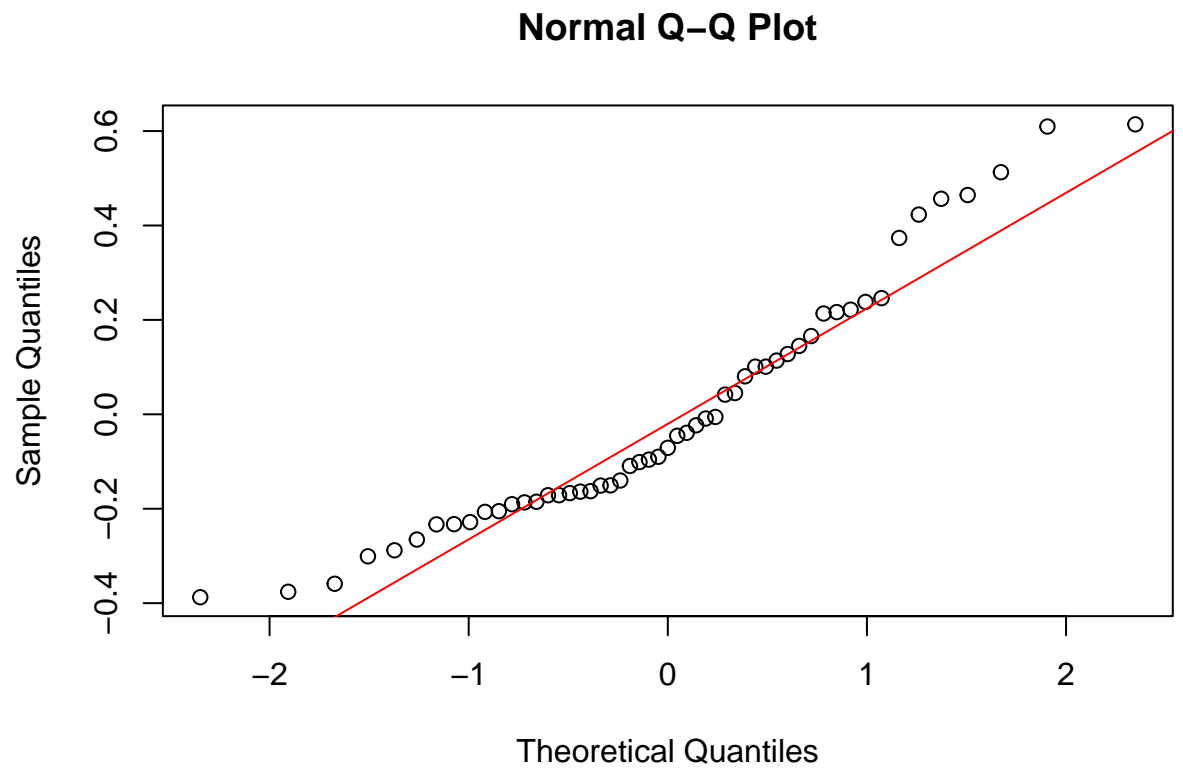
```
confint(R1)
```

```
##           2.5 %      97.5 %
## (Intercept)  0.6392783659  0.849887688
## X3          -0.0100848532 -0.002889577
## X5          -0.0009770002  0.009643095
## X6          -0.0057427822 -0.000326232
```

Verificación de supuestos

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

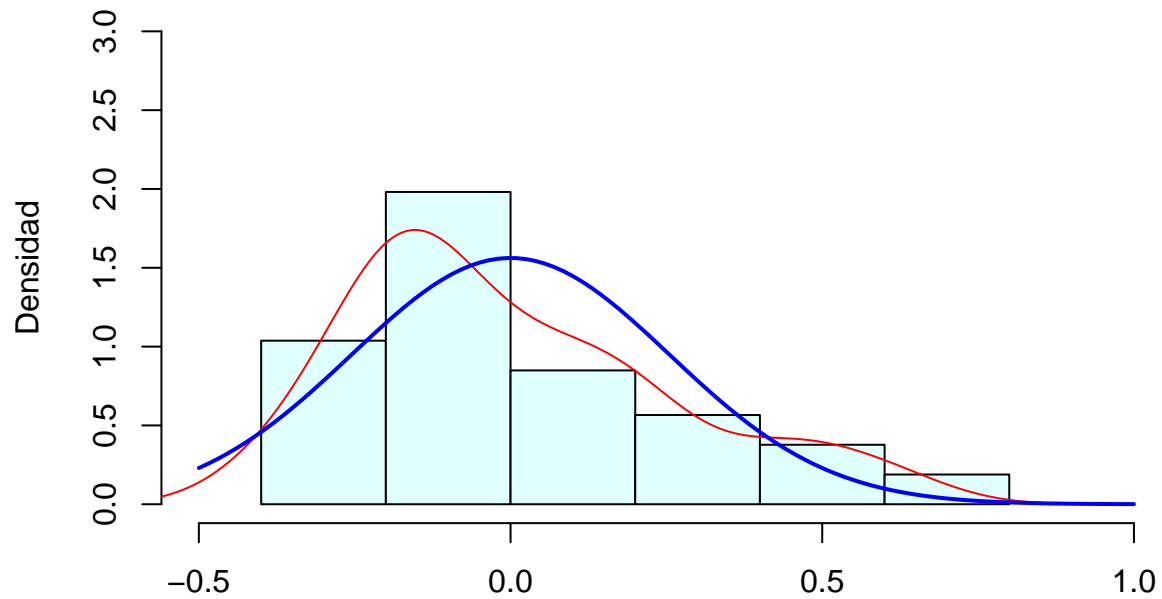
qqnorm(E)
qqline(E, col = "red")
```



Normalidad

```
hist(E, col = "lightcyan", freq = FALSE, main = "Histograma de Residuos", xlim = c(-0.5, 1), ylim = c(0, 0.1))  
lines(density(E), col = "red")  
curve(dnorm(x, mean = mean(E), sd = sd(E)), add = TRUE, col = "blue", lwd = 2)
```


Histograma de Residuos

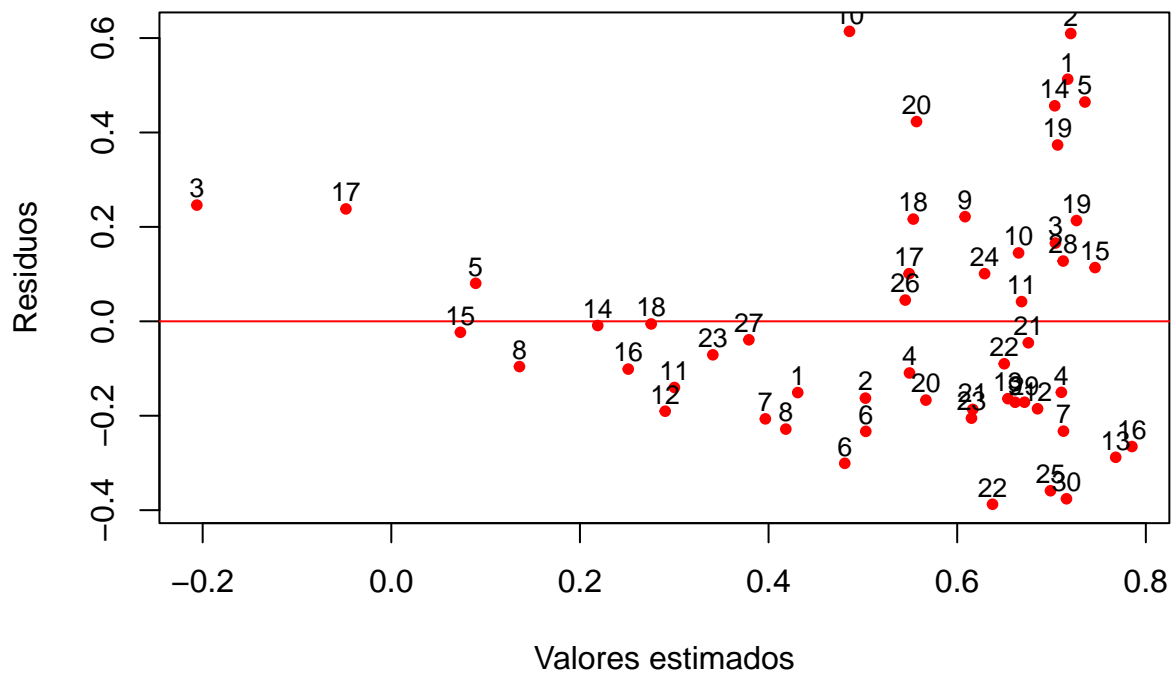


```
shapiro.test(E)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: E  
## W = 0.93258, p-value = 0.005116
```

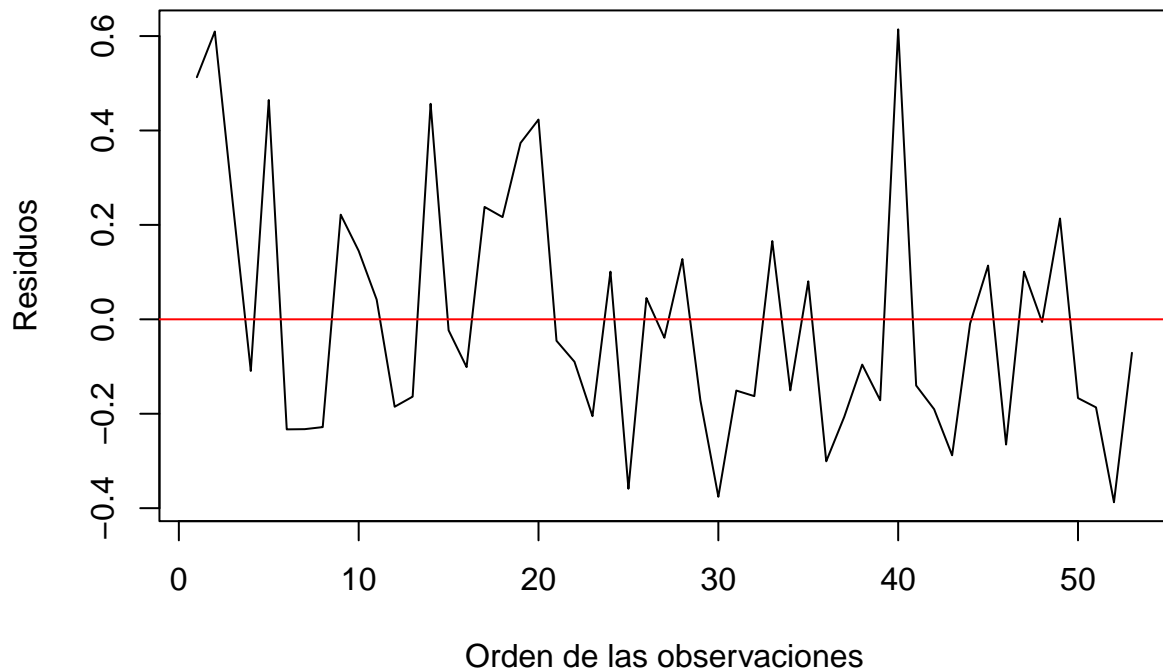
Homocedasticidad y modelo apropiado Gráfica Valores estimados vs Residuos

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



```
#Prueba de autocorrelación para verificar independencia: H0: 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 0.1660837 1.588784 0.126
```

```
## 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':
```

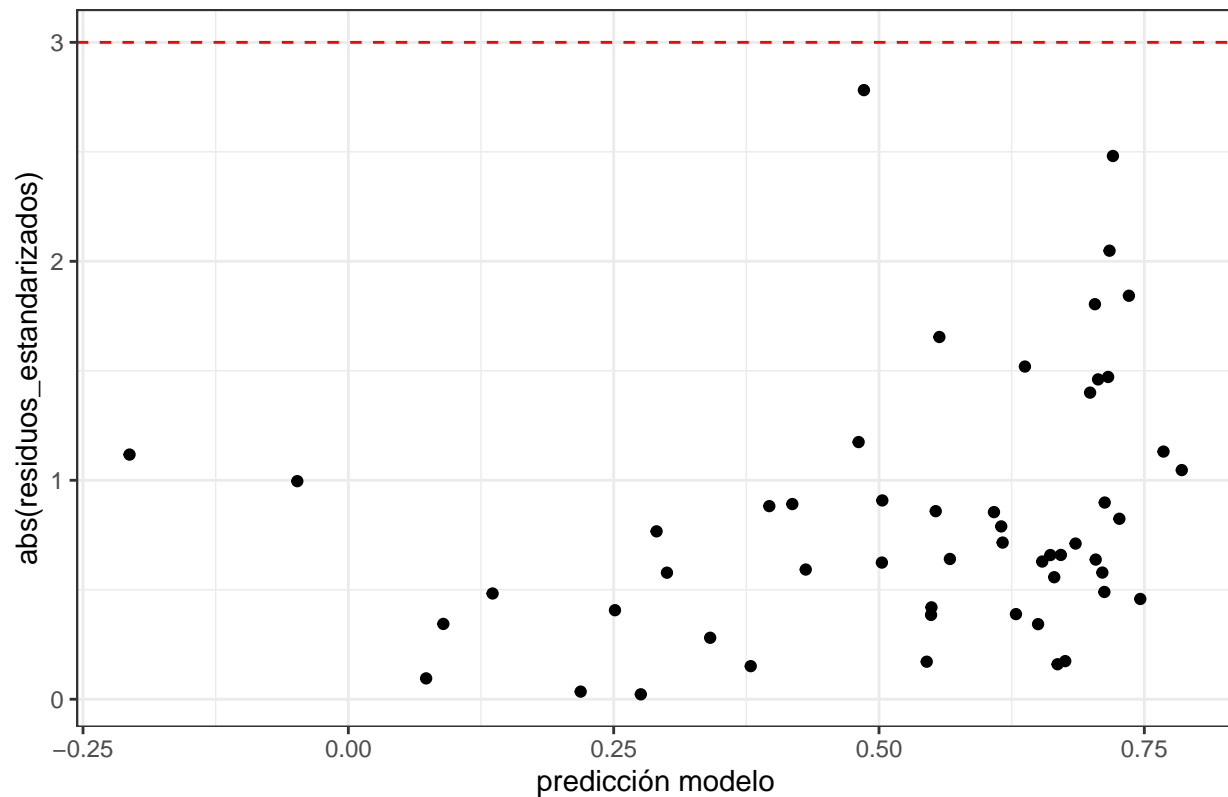
```
##
```

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

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

```
ggplot(data = db_mercurio_mult, 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(db_mercurio_mult$residuos_estandarizados)>3)
```

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

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

Datos influyentes

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

	dfb.1_	dfb.X3	dfb.X5	dfb.X6	dffit	cov.r	cook.d	hat
## 2	0.48	-0.11	-0.04	-0.09	0.48	0.70_*	0.05	0.04
## 3	-0.22	0.28	-0.29	0.52	0.72	1.39_*	0.13	0.29_*
## 15	0.02	0.00	-0.02	-0.01	-0.04	1.28_*	0.00	0.15
## 35	-0.02	0.17	-0.11	-0.07	0.18	1.38_*	0.01	0.22
## 37	0.07	0.07	-0.31	0.18	-0.46	1.29_*	0.05	0.21
## 38	0.06	0.17	-0.09	-0.40	-0.43	1.90_*	0.05	0.44_*
## 40	-0.11	-0.50	1.14_*	-0.38	1.38_*	0.75_*	0.42	0.20
## 41	0.03	-0.09	-0.06	0.15	-0.25	1.26_*	0.02	0.16
## 48	0.00	-0.01	0.01	0.00	-0.01	1.25_*	0.00	0.13

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

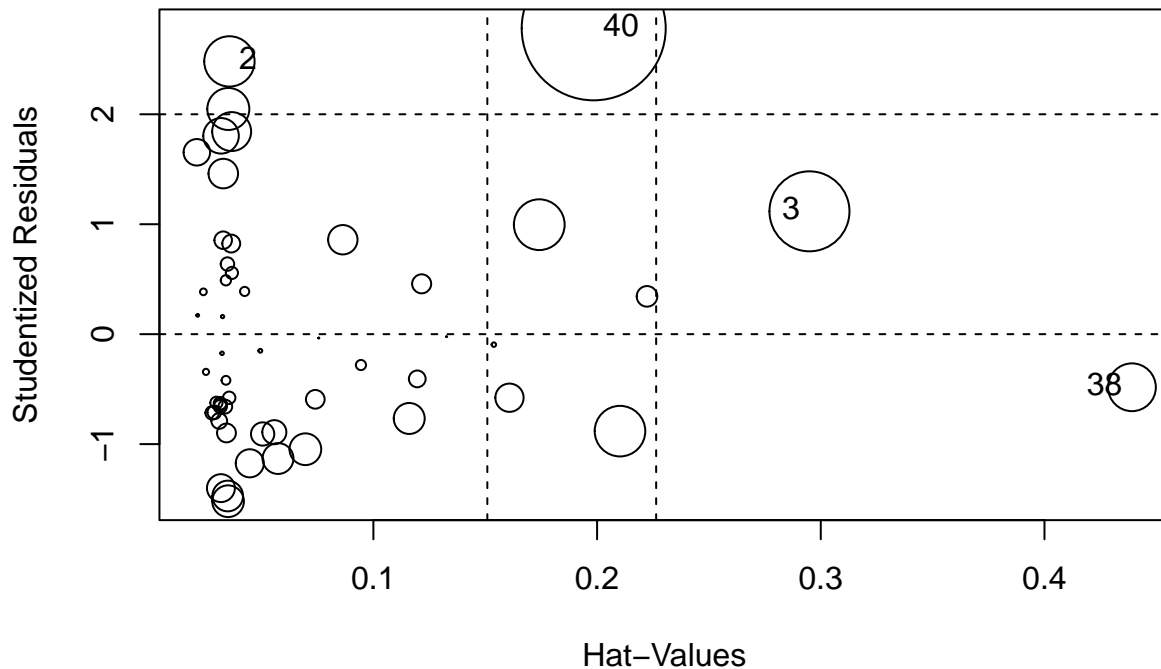
```
## Influence measures of
## lm(formula = X7 ~ X3 + X5 + X6, data = db_mercurio_mult) :
##
##      dfb.1_    dfb.X3    dfb.X5    dfb.X6    dffit cov.r    cook.d    hat inf
## 1  0.388970 -0.06468 -0.039438 -0.10827  0.39101 0.805 3.59e-02 0.0352
## 2  0.476052 -0.10616 -0.043719 -0.08697  0.47696 0.695 5.15e-02 0.0356 *
## 3 -0.222093  0.28467 -0.290188  0.51806  0.72269 1.390 1.30e-01 0.2949 *
## 4 -0.050777 -0.03679  0.028122  0.04376 -0.07886 1.108 1.58e-03 0.0341
## 5  0.358090 -0.10303 -0.003135 -0.07513  0.35939 0.858 3.08e-02 0.0366
## 6 -0.106736  0.00905  0.077856 -0.13501 -0.20935 1.068 1.10e-02 0.0505
## 7 -0.168885  0.03466  0.015983  0.03359 -0.16938 1.052 7.20e-03 0.0343
## 8 -0.000803  0.04359 -0.134767  0.01498 -0.21654 1.077 1.18e-02 0.0557
## 9  0.128410  0.05020 -0.058758 -0.07517  0.15752 1.057 6.24e-03 0.0329
## 10 0.092138 -0.04622  0.000161  0.03486  0.10886 1.099 3.01e-03 0.0368
## 11 0.027551 -0.00706 -0.004475  0.00319  0.02932 1.120 2.19e-04 0.0326
## 12 -0.113077  0.03581 -0.022954  0.04127 -0.12261 1.072 3.80e-03 0.0289
## 13 -0.079432  0.05788 -0.062119  0.02397 -0.11338 1.085 3.25e-03 0.0315
## 14 0.324494 -0.08702 -0.007267 -0.05235  0.32752 0.863 2.56e-02 0.0319
## 15 0.016324 -0.00374 -0.015564 -0.00657 -0.04060 1.282 4.21e-04 0.1538 *
## 16 0.034233  0.05030 -0.081520 -0.08160 -0.14990 1.217 5.71e-03 0.1196
## 17 -0.102253  0.32584 -0.281250  0.17347  0.45733 1.212 5.23e-02 0.1742
## 18 0.033523 -0.09567  0.203278 -0.08968  0.26404 1.118 1.75e-02 0.0863
## 19 0.268051 -0.06087 -0.017937 -0.04918  0.26942 0.944 1.77e-02 0.0329
## 20 0.197518  0.03595 -0.060583 -0.02930  0.24278 0.889 1.42e-02 0.0211
## 21 -0.030911  0.00432  0.006855  0.00159 -0.03181 1.119 2.58e-04 0.0324
## 22 -0.052783  0.00940  0.001515  0.01032 -0.05511 1.103 7.73e-04 0.0252
## 23 -0.122204  0.01528  0.040693 -0.03176 -0.14141 1.064 5.04e-03 0.0311
## 24 0.058190 -0.03705  0.001675  0.04215  0.08185 1.120 1.70e-03 0.0425
## 25 -0.249601  0.03884  0.015308  0.08053 -0.25395 0.956 1.58e-02 0.0318
## 26 0.020025  0.00299 -0.007356  0.00132  0.02538 1.107 1.64e-04 0.0215
## 27 -0.010863 -0.02630  0.024816  0.00358 -0.03446 1.140 3.03e-04 0.0495
## 28 0.090883 -0.02778 -0.002940 -0.00800  0.09201 1.102 2.15e-03 0.0341
## 29 -0.115528  0.02872  0.021206 -0.01434 -0.12326 1.084 3.84e-03 0.0338
## 30 -0.278097  0.04597  0.027058  0.07899 -0.27978 0.943 1.91e-02 0.0349
## 31 0.004002  0.07299 -0.125312 -0.02624 -0.16750 1.139 7.11e-03 0.0740
## 32 -0.068435 -0.01467  0.047262 -0.04320 -0.10944 1.084 3.03e-03 0.0298
## 33 0.120124 -0.01556 -0.021264 -0.02504  0.12102 1.088 3.71e-03 0.0348
## 34 -0.110458  0.01698  0.018111  0.02124 -0.11104 1.095 3.13e-03 0.0355
## 35 -0.015672  0.16987 -0.108245 -0.07402  0.18390 1.383 8.61e-03 0.2223 *
## 36 -0.120720  0.00731  0.086380 -0.16680 -0.25421 1.015 1.60e-02 0.0448
## 37 0.065182  0.06774 -0.313523  0.17655 -0.45503 1.289 5.20e-02 0.2102 *
## 38 0.061805  0.16772 -0.086795 -0.40496 -0.42728 1.899 4.64e-02 0.4391 *
## 39 -0.113188  0.01743  0.026472 -0.00363 -0.11885 1.082 3.57e-03 0.0316
## 40 -0.110028 -0.49826  1.135482 -0.38357  1.38433 0.745 4.21e-01 0.1985 *
## 41 0.031899 -0.08908 -0.059832  0.15064 -0.25298 1.259 1.62e-02 0.1608 *
## 42 0.000794 -0.20023  0.071853  0.16526 -0.27777 1.170 1.95e-02 0.1160
## 43 -0.184014  0.18795 -0.188752  0.07401 -0.27905 1.037 1.94e-02 0.0574
## 44 0.000190 -0.00378  0.004500 -0.00640 -0.00995 1.175 2.53e-05 0.0756
## 45 0.054380 -0.14636  0.150082  0.01941  0.17037 1.215 7.38e-03 0.1216
## 46 -0.173793  0.23206 -0.208028  0.01075 -0.28623 1.066 2.04e-02 0.0696
## 47 0.045465 -0.00424 -0.012677  0.01864  0.06040 1.099 9.28e-04 0.0240
```

```
## 48 -0.001014 -0.00801 0.006119 0.00361 -0.00881 1.252 1.98e-05 0.1328 *
## 49 0.158099 -0.05348 -0.002747 -0.01164 0.16034 1.065 6.47e-03 0.0365
## 50 -0.083993 -0.04831 0.045527 0.05603 -0.11580 1.084 3.39e-03 0.0316
## 51 -0.110028 0.00368 0.036853 -0.00788 -0.12129 1.071 3.71e-03 0.0279
## 52 -0.257310 -0.06797 0.124997 0.09761 -0.28953 0.933 2.04e-02 0.0350
## 53 -0.017163 -0.07863 0.059160 0.04006 -0.09064 1.191 2.09e-03 0.0944
```

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  2.4809441 0.03564225 0.05145857
## 3  1.1173717 0.29494242 0.12991250
## 38 -0.4829154 0.43910418 0.04636791
## 40  2.7817319 0.19849694 0.42117600
```

Conclusión

- Escribir el Modelo
- Variabilidad explicada por el modelo (coeficiente de determinación)

- Significancia del modelo: Valor p del modelo (F)
- Si se satisfacen todos los supuestos del modelo (indicar ligeros alejamientos)
- Si hay datos atípicos o datos que influyan en el modelo.
- En caso de que se haya hecho el análisis sin datos influyentes, se reportan ambos modelos comparando sus mejoras.