MR - Trabajo

Alicia Losada | alicia.losada.sanchez@udc.es María Cardoso | m.cardoso@udc.es
Nicolás Muñiz | nicolas.muniz@udc.es

11/12/2024

Regresión Lineal Múltiple

- Antes de empezar, cargamos los datos *OzonoLA.rda*

```
load("Datos/OzonoLA.rda")
attach(OzonoLA)
```

```
## The following objects are masked from OzonoLA (pos = 4):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 6):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 8):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 10):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 12):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 14):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 17):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad
```

```
## The following objects are masked from OzonoLA (pos = 21):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad

## The following objects are masked from OzonoLA (pos = 37):
##
##     DiaMes, DiaSemana, Grad_Pres, Humedad, Inv_Alt_b, Inv_T_b, Mes, Ozono, Pres_Alt, T_ElMonte,
##     T_Sandburg, Vel_Viento, Visibilidad
```

1. Análisis descriptivo

Para el análisis descriptivo de las variables podemos comenzar con una visión general de las variables mediante las funciones `str()` y `summary()`.

```
str(OzonoLA)
```

```
## 'data.frame':    203 obs. of  13 variables:
##  $ Mes        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DiaMes     : int  5 6 7 8 9 12 13 14 15 16 ...
##  $ DiaSemana  : int  1 2 3 4 5 1 2 3 4 5 ...
##  $ Ozono      : num  5.34 5.77 3.69 3.89 5.76 6.39 4.73 4.35 3.94 7 ...
##  $ Pres_Alt   : int  5760 5720 5790 5790 5700 5720 5760 5780 5830 5870 ...
##  $ Vel_Viento : int  3 4 6 3 3 3 6 6 3 2 ...
##  $ Humedad    : int  51 69 19 25 73 44 33 19 19 19 ...
##  $ T_Sandburg : int  54 35 45 55 41 51 51 54 58 61 ...
##  $ T_ElMonte  : num  45.3 49.6 46.4 52.7 48 ...
##  $ Inv_Alt_b  : int  1450 1568 2631 554 2083 111 492 5000 1249 5000 ...
##  $ Grad_Pres  : int  25 15 -33 -28 23 9 -44 -44 -53 -67 ...
##  $ Inv_T_b    : num  57 53.8 54.1 64.8 52.5 ...
##  $ Visibilidad: int  60 60 100 250 120 150 40 200 250 200 ...
```

La salida de `str()` nos dice que los datos constan de 203 observaciones de 13 variables:

- `Mes`: Número del mes en el que se hicieron las observaciones (Entero)
- `DiaMes`: Número del día del mes en el que se hicieron las observaciones (Entero)
- `DíaSemana`: Número del día de la semana en el que se hicieron las observaciones (Entero)
- `Ozono`: Nivel de Ozono medido (Numérica)
- `Pres_Alt`: Altura en metros a la que se alcanza una presion de 500 milibares (Entero)
- `Vel_Viento`: Velocidad del viento en millas por hora en el Aeropuerto Internacional de Los Angeles (Entero)
- `Humedad`: Humedad en porcentaje en LAX (Entero)
- `T_Sandburg`: Temperatura (F) en Sandburg, CA (Entero)
- `T_ElMonte`: Temperatura (F) en El Monte, CA (Numérica)
- `Inv_ALt_b`: Inversion de la altura base (en pies) en LAX (Entero)
- `Grand_Pres`: Gradiente de presion de LAX a Daggett, CA (Entero)
- `Inv_T_b`: Inversion de la temperatura base (F) en LAX (Numérica)
- `Visibilidad`: Visibilidad (millas) evaluada en LAX (Entero)

```
summary(OzonoLA)
```

```
##       Mes             DiaMes       DiaSemana         Ozono          Pres_Alt      Vel_Viento
##  Min.   : 1.000   Min.   : 1.0   Min.   :1.000   Min.   : 0.72   Min.   :5320   Min.   : 0.000
##  1st Qu.: 3.000   1st Qu.: 9.0   1st Qu.:2.000   1st Qu.: 4.77   1st Qu.:5690   1st Qu.: 3.000
##  Median : 6.000   Median :15.0   Median :3.000   Median : 8.90   Median :5760   Median : 5.000
##  Mean   : 6.522   Mean   :15.7   Mean   :3.005   Mean   :11.37   Mean   :5746   Mean   : 4.867
##  3rd Qu.:10.000   3rd Qu.:23.0   3rd Qu.:4.000   3rd Qu.:16.07   3rd Qu.:5830   3rd Qu.: 6.000
```

```
##  Max.   :12.000   Max.   :31.0   Max.   :5.000   Max.   :37.98   Max.   :5950   Max.   :11.000
##     Humedad        T_Sandburg      T_ElMonte       Inv_Alt_b      Grad_Pres         Inv_T_b
##  Min.   :19.00   Min.   :25.00   Min.   :27.68   Min.   : 111   Min.   :-69.00   Min.   :27.50
##  1st Qu.:46.00   1st Qu.:51.50   1st Qu.:49.64   1st Qu.: 869   1st Qu.:-14.00   1st Qu.:51.26
##  Median :64.00   Median :61.00   Median :56.48   Median :2083   Median : 18.00   Median :60.98
##  Mean   :57.61   Mean   :61.11   Mean   :56.54   Mean   :2602   Mean   : 14.43   Mean   :60.69
##  3rd Qu.:73.00   3rd Qu.:71.00   3rd Qu.:66.20   3rd Qu.:5000   3rd Qu.: 43.00   3rd Qu.:70.88
##  Max.   :93.00   Max.   :93.00   Max.   :82.58   Max.   :5000   Max.   :107.00   Max.   :90.68
##   Visibilidad
##  Min.   :  0.0
##  1st Qu.: 60.0
##  Median :100.0
##  Mean   :122.2
##  3rd Qu.:150.0
##  Max.   :350.0
```

Ahora realizaremos un análisis descriptivo de cada variable:

Análisis descriptivo de la variable `Mes` :

```
summary(Mes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   6.000   6.522  10.000  12.000
```

Desviación típica y rango intercuartílico:

```
sd(Mes)
```

```
## [1] 3.594998
```

```
IQR(Mes)
```

```
## [1] 7
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Mes, na.rm = FALSE)
```

```
## [1] 0.03220505
```

```
kurtosis(Mes, na.rm = FALSE)
```

```
## [1] 1.671129
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis menor que tres, las colas de la variable comparadas con una normal son más ligeras.

Vemos si hay registros atípicos

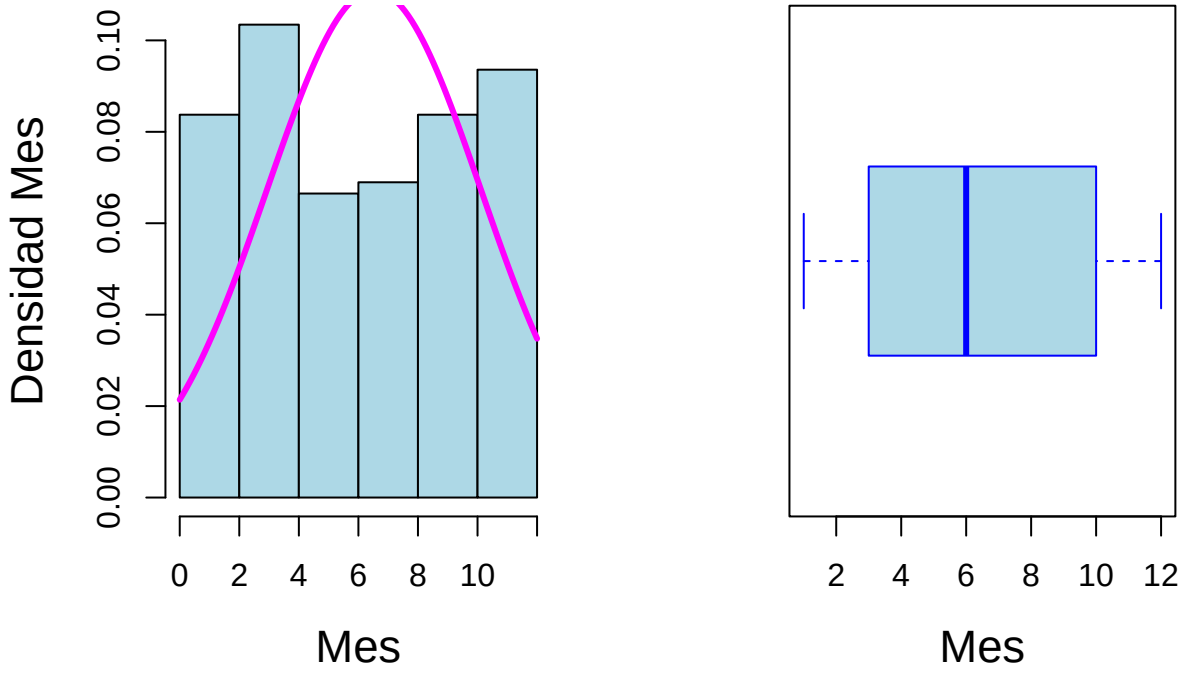
```
boxplot.stats(Mes)$out
```

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

Como podemos ver no existe ningún registro atípico

```
par(mfrow=c(1,2))
hist(Mes, breaks=5,freq=FALSE, main = "", xlab="Mes",
     cex.lab=1.4, ylab = "Densidad Mes", col = "lightblue")
curve( dnorm(x,mean=mean(Mes),sd=sd(Mes)),
```

```
       col="magenta", lwd=3, add=TRUE)
boxplot(Mes, main = "", xlab="Mes",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable DiaMes :

```
summary(Mes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   6.000   6.522  10.000  12.000
```

Desviación típica y rango intercuartílico:

```
sd(DiaMes)
```

```
## [1] 8.569537
```

```
IQR(DiaMes)
```

```
## [1] 14
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(DiaMes, na.rm = FALSE)
```

```
## [1] 0.0395616
```

```
kurtosis(DiaMes, na.rm = FALSE)
```

```
## [1] 1.868548
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis menor que tres, las colas de la variable comparadas con una normal son más ligeras.

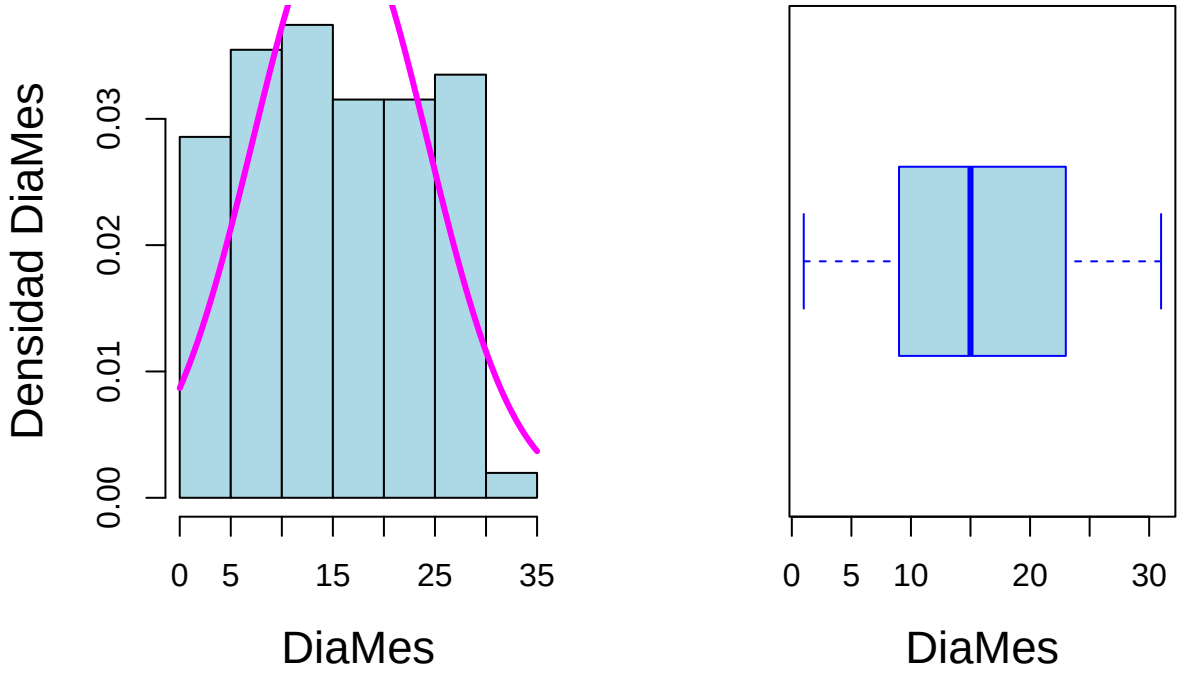
Vemos si hay registros atípicos

```
boxplot.stats(DiaMes)$out
```

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

Como podemos ver no existe ningún registro atípico

```
par(mfrow=c(1,2))
hist(DiaMes, breaks=5,freq=FALSE, main = "", xlab="DiaMes",
     cex.lab=1.4, ylab = "Densidad DiaMes", col = "lightblue")
curve( dnorm(x,mean=mean(DiaMes),sd=sd(DiaMes)),
       col="magenta", lwd=3, add=TRUE)
boxplot(DiaMes, main = "", xlab="DiaMes",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `DiaSemana` :

```
summary(DiaSemana)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.005   4.000   5.000
```

Desviación típica y rango intercuartílico:

```
sd(DiaSemana)
```

```
## [1] 1.401899
```

```
IQR(DiaSemana)
```

```
## [1] 2
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(DiaSemana, na.rm = FALSE)
```

```
## [1] 0.04527053
```

```
kurtosis(DiaSemana, na.rm = FALSE)
```

```
## [1] 1.731687
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis menor que tres, las colas de la variable comparadas con una normal son más ligeras.

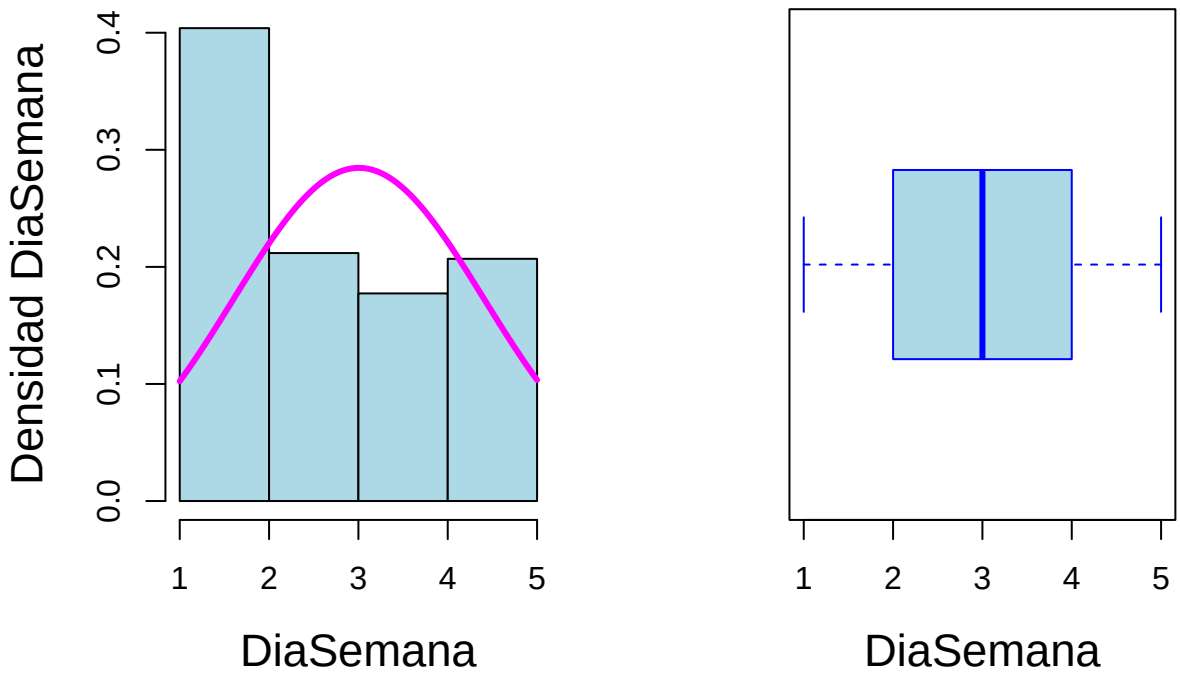
Vemos si hay registros atípicos

```
boxplot.stats(DiaSemana)$out
```

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

Como podemos ver no existe ningún registro atípico

```
par(mfrow=c(1,2))
hist(DiaSemana, breaks=5,freq=FALSE, main = "", xlab="DiaSemana",
     cex.lab=1.4, ylab = "Densidad DiaSemana", col = "lightblue")
curve( dnorm(x,mean=mean(DiaSemana),sd=sd(DiaSemana)),
       col="magenta", lwd=3, add=TRUE)
boxplot(DiaSemana, main = "", xlab="DiaSemana",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `Ozono` :

```
summary(Ozono)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.72    4.77    8.90   11.37   16.07   37.98
```

Desviación típica y rango intercuartílico:

```
sd(Ozono)
```

```
## [1] 8.192652
```

```
IQR(Ozono)
```

```
## [1] 11.305
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Ozono, na.rm = FALSE)
```

```
## [1] 0.9652702
```

```
kurtosis(Ozono, na.rm = FALSE)
```

```
## [1] 3.089498
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es próximo a tres, las colas de la variable son similares a las de una normal

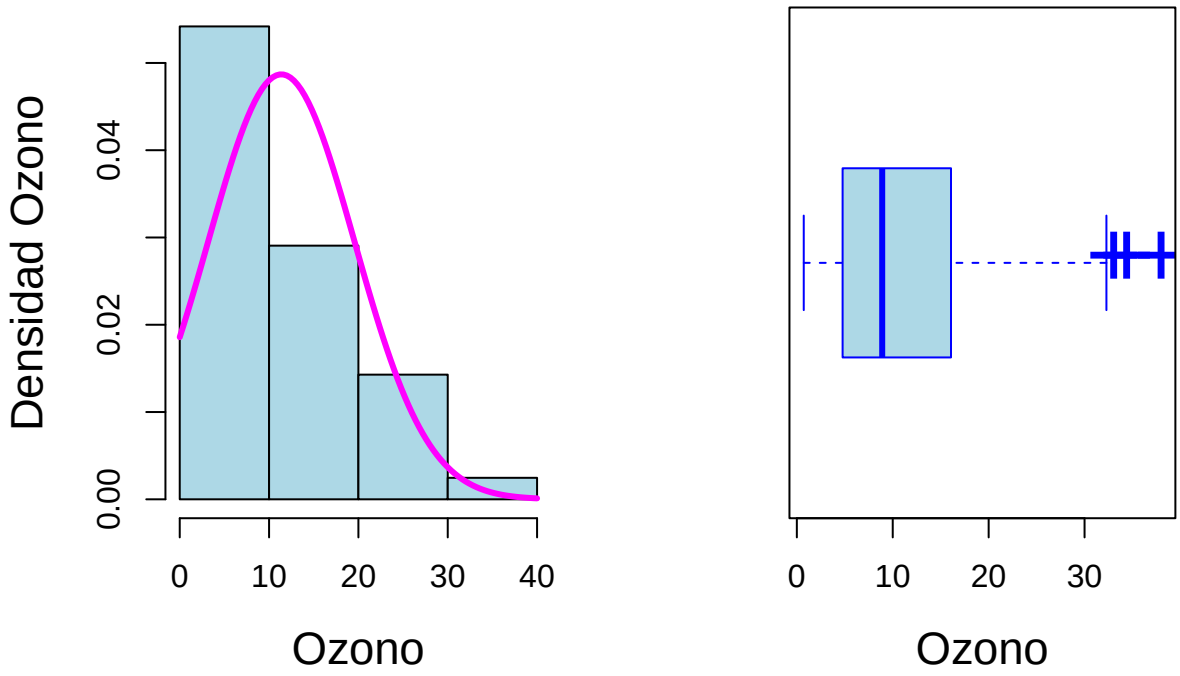
Vemos si hay registros atípicos

```
boxplot.stats(Ozono)$out
```

```
## [1] 33.04 34.39 37.98
```

Como podemos ver existen 4 registros atípicos

```
par(mfrow=c(1,2))
hist(Ozono, breaks=5,freq=FALSE, main = "", xlab="Ozono",
     cex.lab=1.4, ylab = "Densidad Ozono", col = "lightblue")
curve( dnorm(x,mean=mean(Ozono),sd=sd(Ozono)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Ozono, main = "", xlab="Ozono",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `Pres_Alt` :

```
summary(Pres_Alt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5320    5690    5760    5746    5830    5950
```

Desviación típica y rango intercuartílico:

```
sd(Pres_Alt)
```

```
## [1] 113.0277
```

```
IQR(Pres_Alt)
```

```
## [1] 140
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Pres_Alt, na.rm = FALSE)
```

```
## [1] -0.9499496
```

```
kurtosis(Pres_Alt, na.rm = FALSE)
```

```
## [1] 4.198772
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es mayor a tres, las colas de la variable son más grandes que las de una normal.

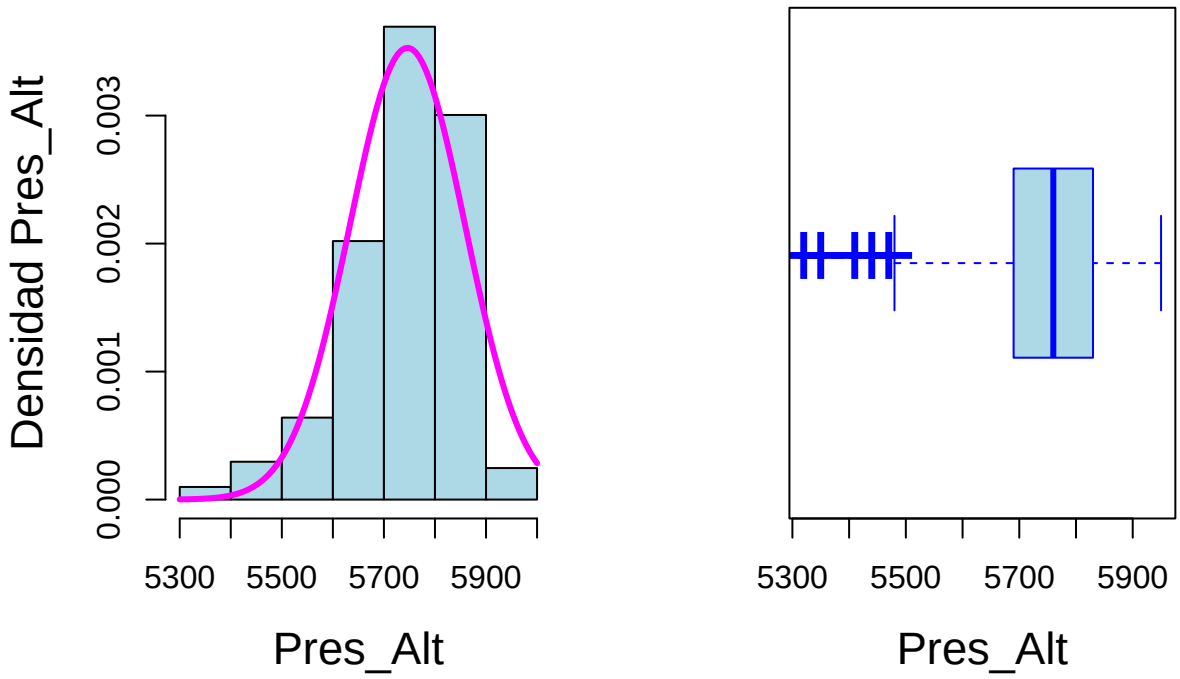
Vemos si hay registros atípicos

```
boxplot.stats(Pres_Alt)$out
```

```
## [1] 5410 5350 5470 5320 5440
```

Como podemos ver existen 5 registros atípicos

```
par(mfrow=c(1,2))
hist(Pres_Alt, breaks=5,freq=FALSE, main = "", xlab="Pres_Alt",
     cex.lab=1.4, ylab = "Densidad Pres_Alt", col = "lightblue")
curve( dnorm(x,mean=mean(Pres_Alt),sd=sd(Pres_Alt)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Pres_Alt, main = "", xlab="Pres_Alt",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `Vel_Viento` :

```
summary(Vel_Viento)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   5.000   4.867   6.000  11.000
```

Desviación típica y rango intercuartílico:

```
sd(Vel_Viento)
```

```
## [1] 2.105402
```

```
IQR(Vel_Viento)
```

```
## [1] 3
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Vel_Viento, na.rm = FALSE)
```

```
## [1] 0.09612047
```

```
kurtosis(Vel_Viento, na.rm = FALSE)
```

```
## [1] 3.378636
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es próximo a tres, las colas de la variable son similares a las de una normal.

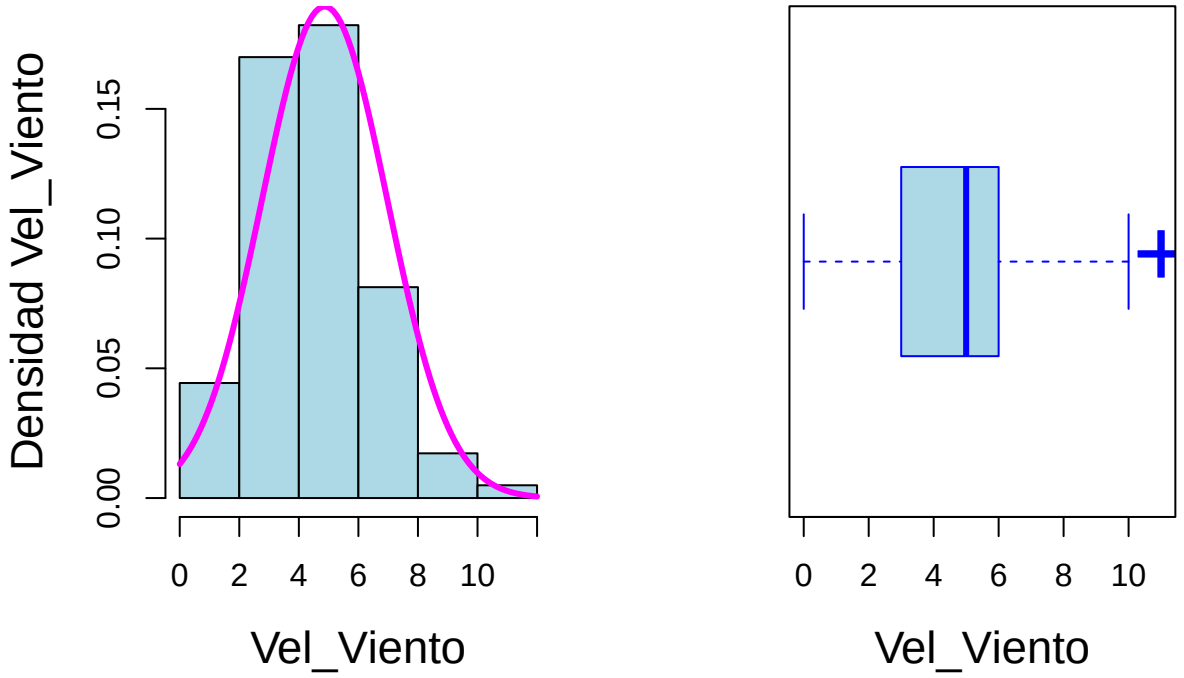
Vemos si hay registros atípicos

```
boxplot.stats(Vel_Viento)$out
```

```
## [1] 11 11
```

Como podemos ver existen 2 registros atípicos

```r
par(mfrow=c(1,2))
hist(Vel_Viento, breaks=5,freq=FALSE, main = "", xlab="Vel_Viento",
     cex.lab=1.4, ylab = "Densidad Vel_Viento", col = "lightblue")
curve( dnorm(x,mean=mean(Vel_Viento),sd=sd(Vel_Viento)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Vel_Viento, main = "", xlab="Vel_Viento",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `Humedad` :

```r
summary(Humedad)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   46.00   64.00   57.61   73.00   93.00
```

Desviación típica y rango intercuartílico:

```r
sd(Humedad)
```

```
## [1] 20.84766
```

```r
IQR(Humedad)
```

```
## [1] 27
```

Evaluamos la asimetría y kurtoisis

```r
library(moments)
skewness(Humedad, na.rm = FALSE)
```

```
## [1] -0.6935066
```

```r
kurtosis(Humedad, na.rm = FALSE)
```

```
## [1] 2.307891
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es próximo a tres, las colas de la variable son similares a las de una normal.

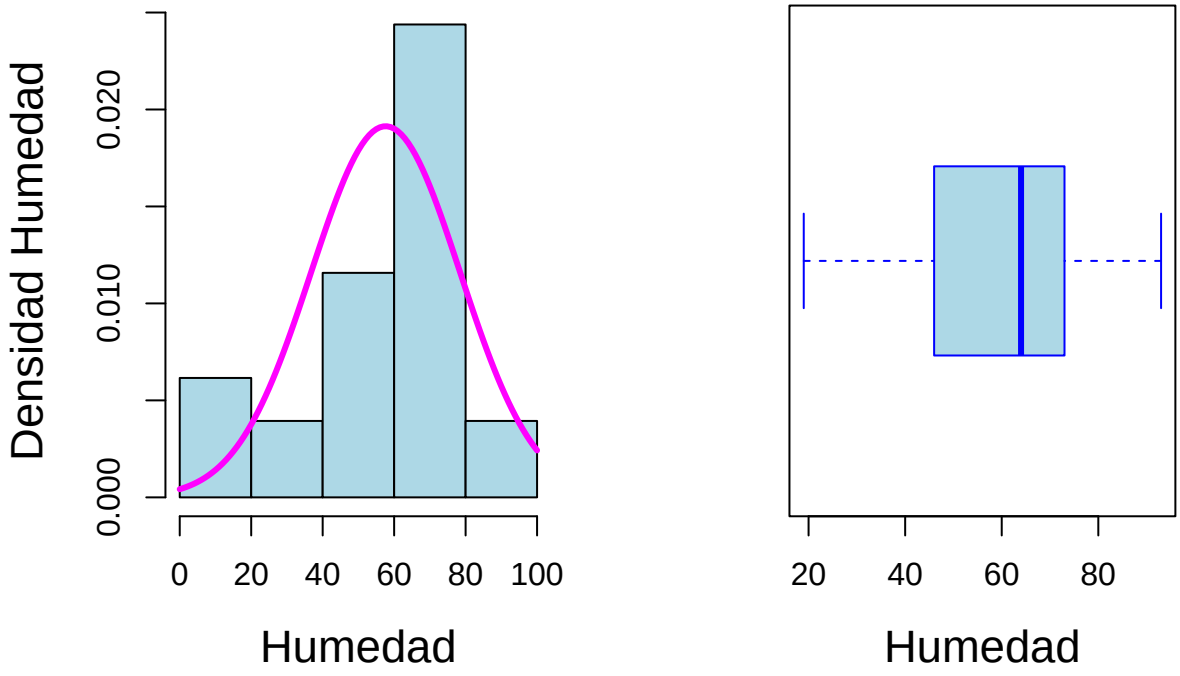
Vemos si hay registros atípicos

```
boxplot.stats(Humedad)$out
```

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

Como podemos ver no existen registros atípicos

```
par(mfrow=c(1,2))
hist(Humedad, breaks=5,freq=FALSE, main = "", xlab="Humedad",
     cex.lab=1.4, ylab = "Densidad Humedad", col = "lightblue")
curve( dnorm(x,mean=mean(Humedad),sd=sd(Humedad)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Humedad, main = "", xlab="Humedad",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `T_Sandburg` :

```
summary(T_Sandburg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   51.50   61.00   61.11   71.00   93.00
```

Desviación típica y rango intercuartílico:

```
sd(T_Sandburg)
```

```
## [1] 14.20647
```

```
IQR(T_Sandburg)
```

```
## [1] 19.5
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(T_Sandburg, na.rm = FALSE)
```

```
## [1] 0.006212875
```

```
kurtosis(T_Sandburg, na.rm = FALSE)
```

```
## [1] 2.510297
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es próximo a tres, las colas de la variable son similares a las de una normal.

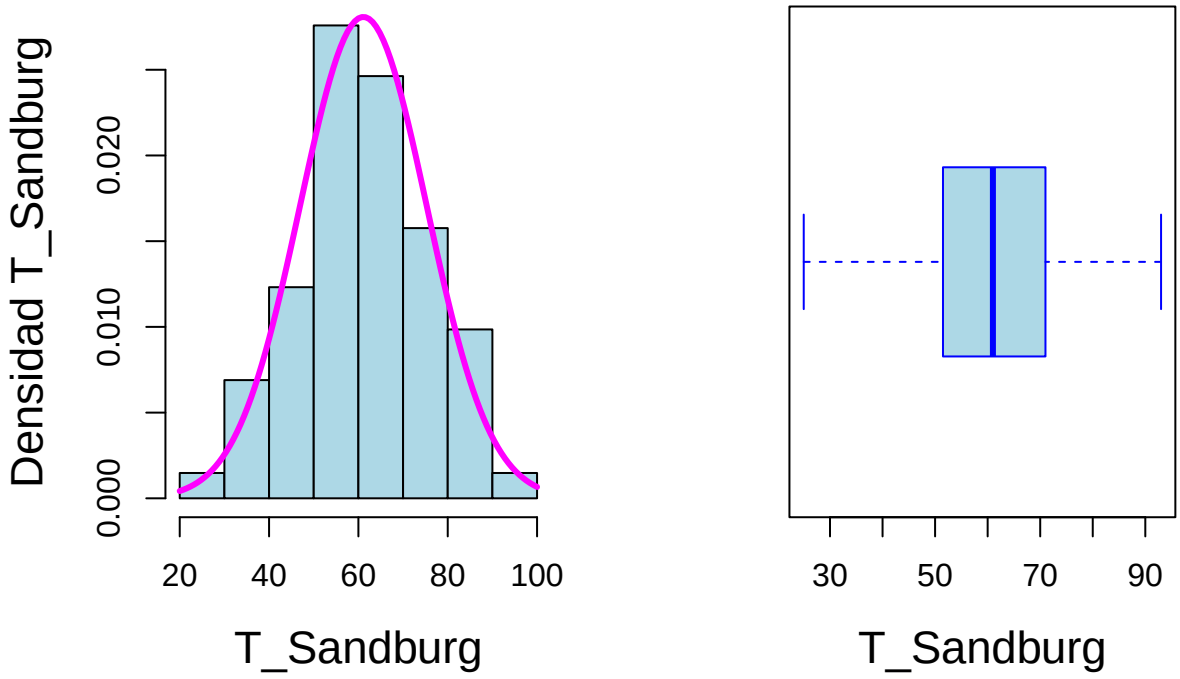
Vemos si hay registros atípicos

```
boxplot.stats(T_Sandburg)$out
```

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

Como podemos ver no existen registros atípicos

```
par(mfrow=c(1,2))
hist(T_Sandburg, breaks=5,freq=FALSE, main = "", xlab="T_Sandburg",
     cex.lab=1.4, ylab = "Densidad T_Sandburg", col = "lightblue")
curve( dnorm(x,mean=mean(T_Sandburg),sd=sd(T_Sandburg)),
       col="magenta", lwd=3, add=TRUE)
boxplot(T_Sandburg, main = "", xlab="T_Sandburg",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



- ANÁLISIS DESCRIPTIVO VARIABLE 'T_ElMonte'

```
summary(T_ElMonte)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.68   49.64   56.48   56.54   66.20   82.58
```

Desviación típica y rango intercuartílico:

```
sd(T_ElMonte)
```

```
## [1] 11.74267
```

```
IQR(T_ElMonte)
```

```
## [1] 16.56
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(T_ElMonte, na.rm = FALSE)
```

```
## [1] -0.1025587
```

```
kurtosis(T_ElMonte, na.rm = FALSE)
```

```
## [1] 2.486231
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es próximo a tres, las colas de la variable son similares a las de una normal.

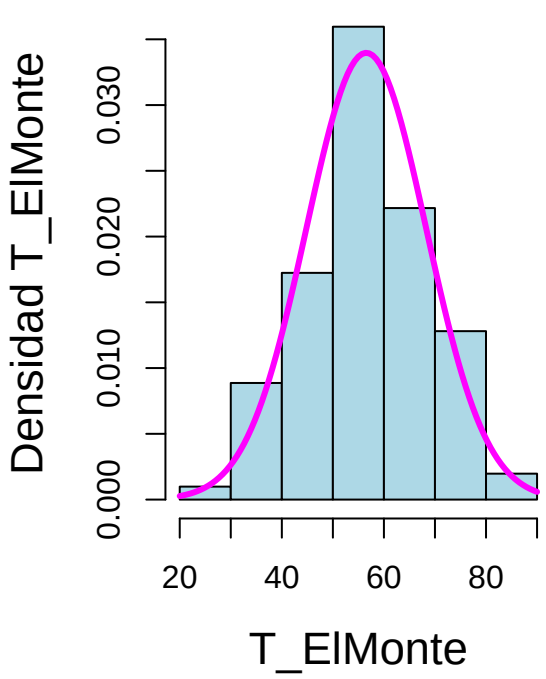
Vemos si hay registros atípicos

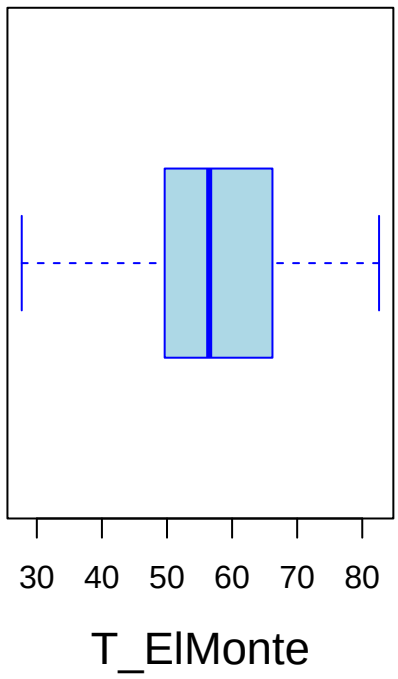
```
boxplot.stats(T_ElMonte)$out
```

```
## numeric(0)
```

Como podemos ver no existen registros atípicos

```
par(mfrow=c(1,2))
hist(T_ElMonte, breaks=5,freq=FALSE, main = "", xlab="T_ElMonte",
     cex.lab=1.4, ylab = "Densidad T_ElMonte", col = "lightblue")
curve( dnorm(x,mean=mean(T_ElMonte),sd=sd(T_ElMonte)),
       col="magenta", lwd=3, add=TRUE)
boxplot(T_ElMonte, main = "", xlab="T_ElMonte",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```





Análisis descriptivo de la variable `Inv_Alt_b` :

```
summary(Inv_Alt_b)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     111     869    2083    2602    5000    5000
```

Desviación típica y rango intercuartílico:

```
sd(Inv_Alt_b)
```

```
## [1] 1859.889
```

```
IQR(Inv_Alt_b)
```

```
## [1] 4131
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Inv_Alt_b, na.rm = FALSE)
```

```
## [1] 0.2355015
```

```
kurtosis(Inv_Alt_b, na.rm = FALSE)
```

```
## [1] 1.374057
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es menor a tres, las colas de la variable son más ligeras a las de una normal.

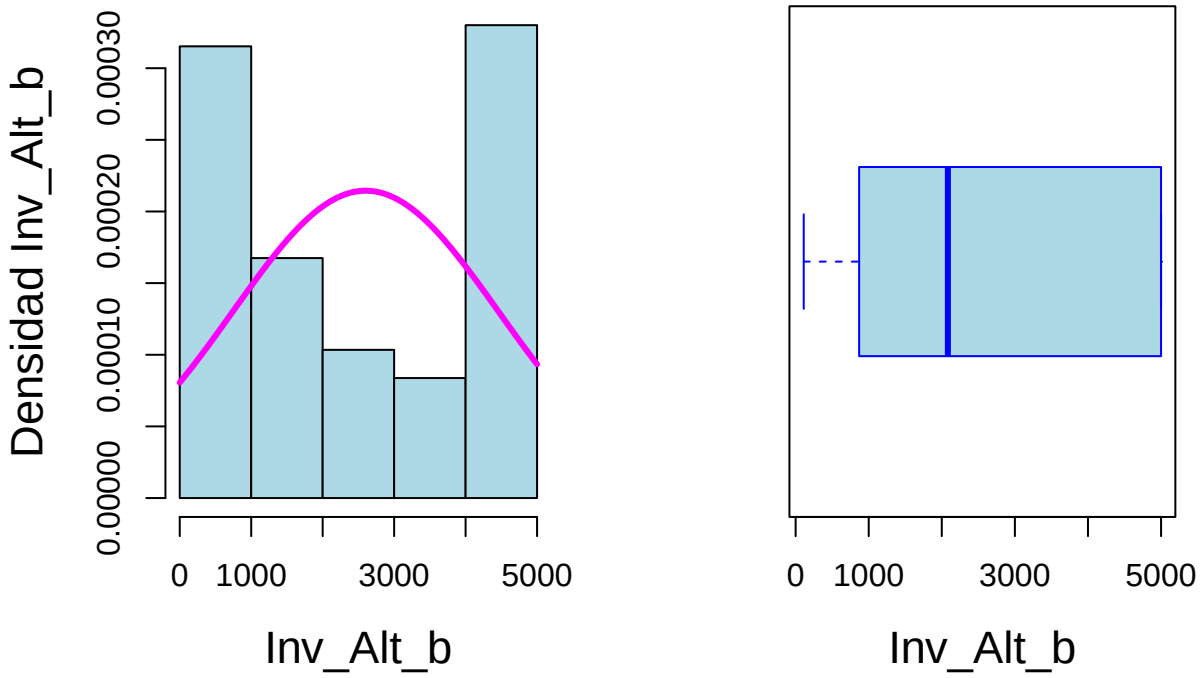
Vemos si hay registros atípicos

```
boxplot.stats(Inv_Alt_b)$out
```

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

Como podemos ver no existen registros atípicos

```r
par(mfrow=c(1,2))
hist(Inv_Alt_b, breaks=5,freq=FALSE, main = "", xlab="Inv_Alt_b",
     cex.lab=1.4, ylab = "Densidad Inv_Alt_b", col = "lightblue")
curve( dnorm(x,mean=mean(Inv_Alt_b),sd=sd(Inv_Alt_b)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Inv_Alt_b, main = "", xlab="Inv_Alt_b",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable Grad_Pres :

```r
summary(Grad_Pres)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -69.00  -14.00   18.00   14.43   43.00  107.00
```

Desviación típica y rango intercuartílico:

```r
sd(Grad_Pres)
```

```
## [1] 36.3172
```

```r
IQR(Grad_Pres)
```

```
## [1] 57
```

Evaluamos la asimetría y kurtoisis

```r
library(moments)
skewness(Grad_Pres, na.rm = FALSE)
```

```
## [1] -0.131977
```

```r
kurtosis(Grad_Pres, na.rm = FALSE)
```

```
## [1] 2.316879
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es menor a tres, las colas de la variable son más ligeras a las de una normal.

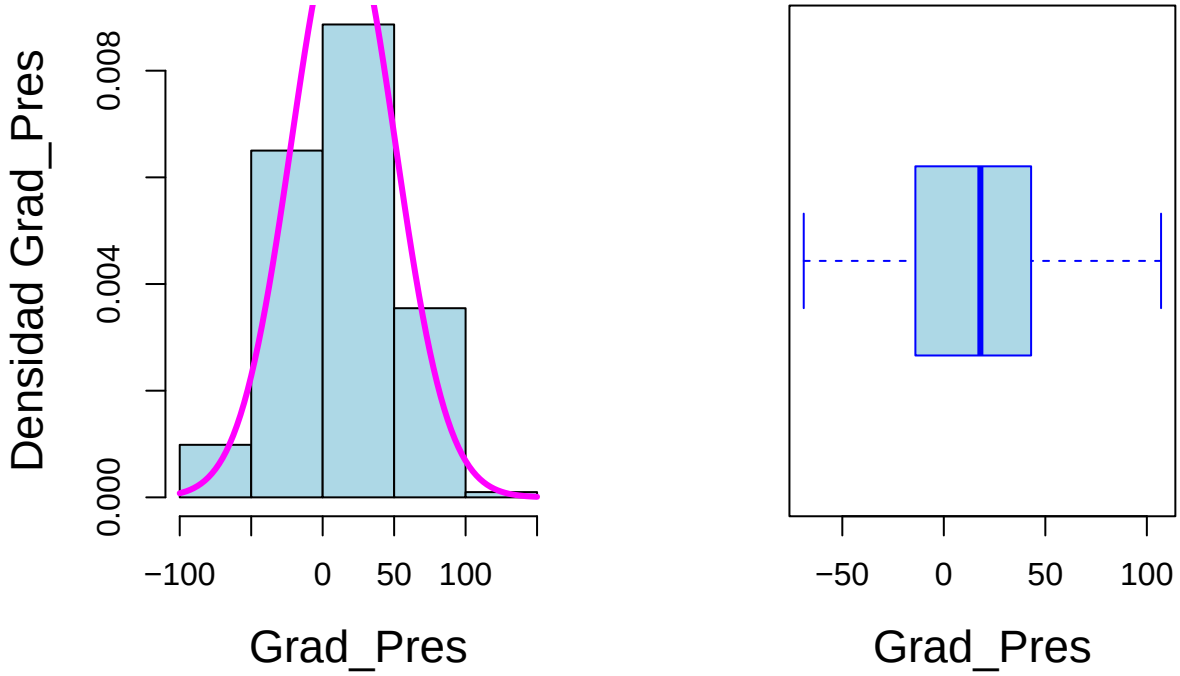
Vemos si hay registros atípicos

```
boxplot.stats(Grad_Pres)$out
```

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

Como podemos ver no existen registros atípicos

```
par(mfrow=c(1,2))
hist(Grad_Pres, breaks=5,freq=FALSE, main = "", xlab="Grad_Pres",
     cex.lab=1.4, ylab = "Densidad Grad_Pres", col = "lightblue")
curve( dnorm(x,mean=mean(Grad_Pres),sd=sd(Grad_Pres)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Grad_Pres, main = "", xlab="Grad_Pres",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `Inv_T_b` :

```
summary(Inv_T_b)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.50   51.26   60.98   60.69   70.88   90.68
```

Desviación típica y rango intercuartílico:

```
sd(Inv_T_b)
```

```
## [1] 14.12473
```

```
IQR(Inv_T_b)
```

```
## [1] 19.62
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Inv_T_b, na.rm = FALSE)
```

```
## [1] -0.1886259
```

```
kurtosis(Inv_T_b, na.rm = FALSE)
```

```
## [1] 2.354789
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis es menor a tres, las colas de la variable son más ligeras a las de una normal.

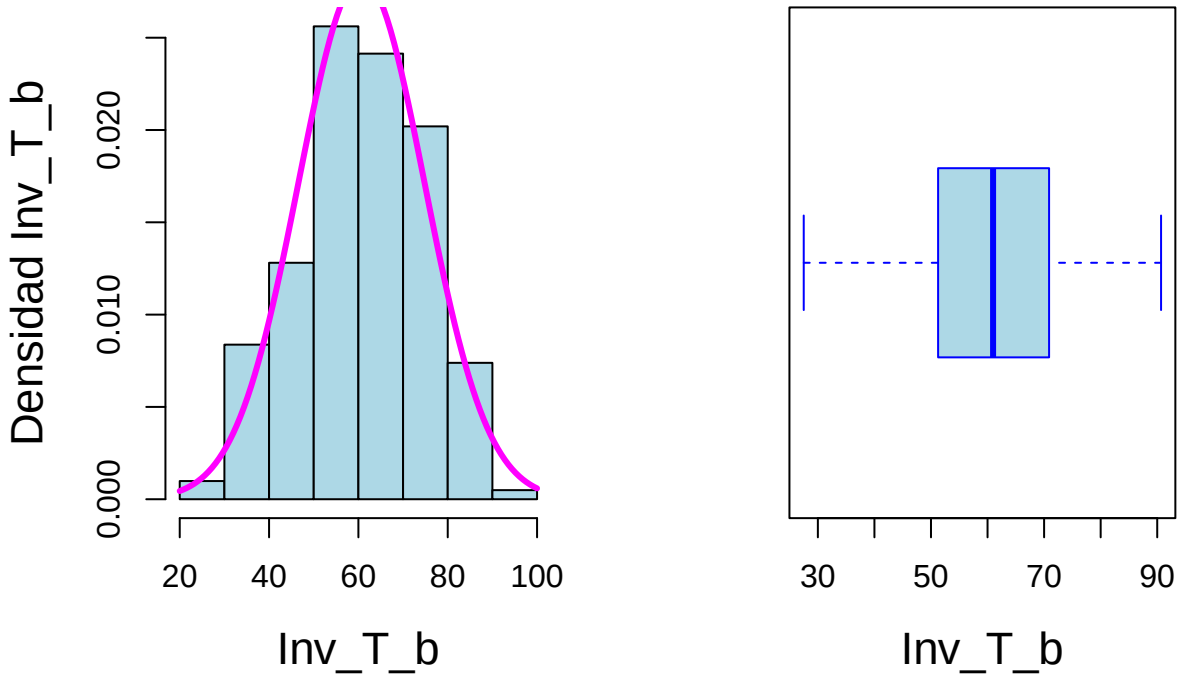
Vemos si hay registros atípicos

```
boxplot.stats(Inv_T_b)$out
```

```
## numeric(0)
```

Como podemos ver no existen registros atípicos

```
par(mfrow=c(1,2))
hist(Inv_T_b, breaks=5,freq=FALSE, main = "", xlab="Inv_T_b",
     cex.lab=1.4, ylab = "Densidad Inv_T_b", col = "lightblue")
curve( dnorm(x,mean=mean(Inv_T_b),sd=sd(Inv_T_b)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Inv_T_b, main = "", xlab="Inv_T_b",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



Análisis descriptivo de la variable `Visibilidad` :

```
summary(Visibilidad)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    60.0   100.0   122.2   150.0   350.0
```

Desviación típica y rango intercuartílico:

```
sd(Visibilidad)
```

```
## [1] 81.17132
```

```
IQR(Visibilidad)
```

```
## [1] 90
```

Evaluamos la asimetría y kurtoisis

```
library(moments)
skewness(Visibilidad, na.rm = FALSE)
```

```
## [1] 0.8067613
```

```
kurtosis(Visibilidad, na.rm = FALSE)
```

```
## [1] 2.903426
```

Podemos ver que al ser el coeficiente de asimetría cercano a 0 que puede ser una variable simética y al ser el coeficiente de Kurtosis próximo a tres, las colas de la variable son próximas a las de una normal.

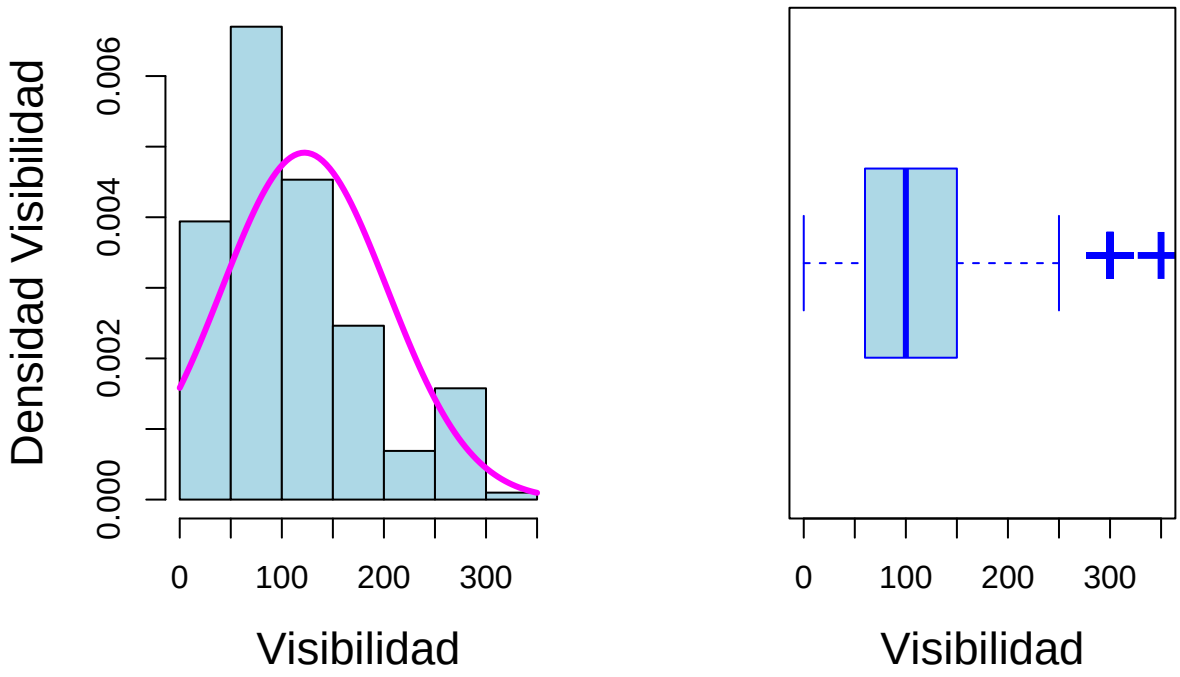
Vemos si hay registros atípicos

```
boxplot.stats(Visibilidad)$out
```

```
##  [1] 350 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300 300
```

Como podemos ver no existen registros atípicos

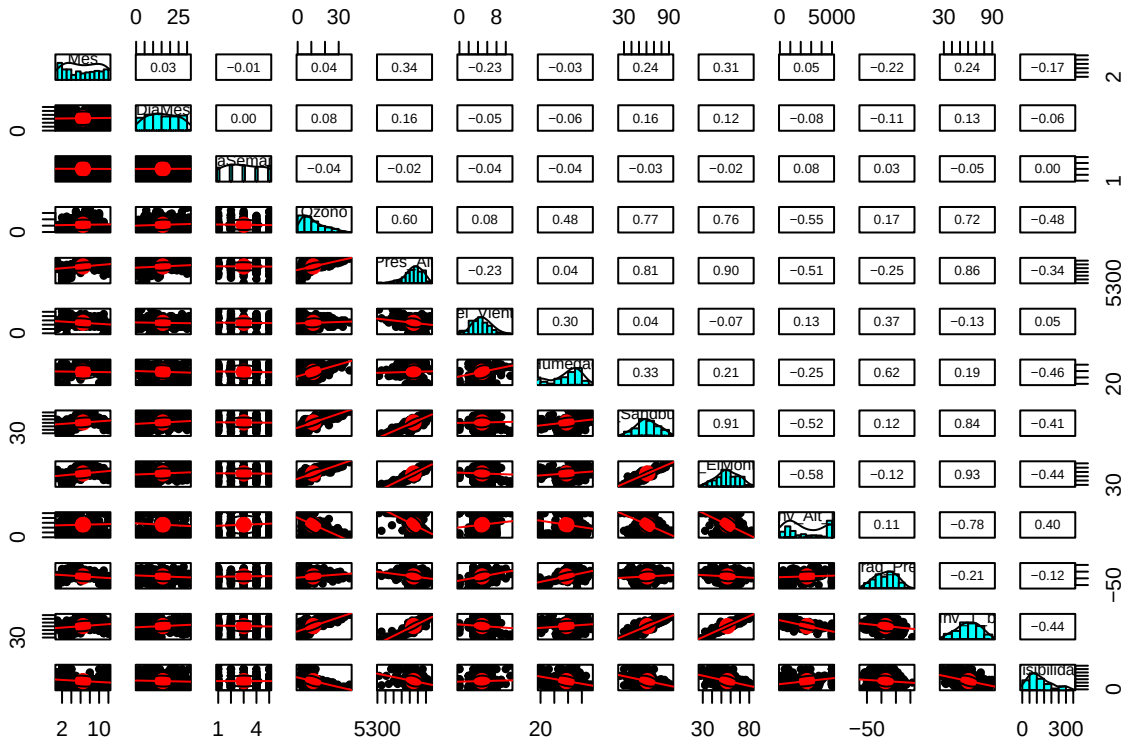
```
par(mfrow=c(1,2))
hist(Visibilidad, breaks=5,freq=FALSE, main = "", xlab="Visibilidad",
     cex.lab=1.4, ylab = "Densidad Visibilidad", col = "lightblue")
curve( dnorm(x,mean=mean(Visibilidad),sd=sd(Visibilidad)),
       col="magenta", lwd=3, add=TRUE)
boxplot(Visibilidad, main = "", xlab="Visibilidad",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)
```



2. Análisis de correlación

- Correlaciones simples bivariantes(análisis gráfico y numérico):

```
library(psych)
pairs.panels(OzonoLA, smooth = TRUE, density=TRUE, digits = 2,
             ellipses=TRUE, method="pearson", pch = 20,
             lm=TRUE, cor=TRUE)
```



```r
cor(OzonoLA)
```

```
##                    Mes       DiaMes     DiaSemana       Ozono    Pres_Alt  Vel_Viento     Humedad
## Mes          1.000000000  0.029780944 -6.406562e-03  0.04417525  0.33793183 -0.22689301 -0.03472729
## DiaMes       0.029780944  1.000000000  3.418381e-03  0.08364060  0.15808064 -0.04609084 -0.06473986
## DiaSemana   -0.006406562  0.003418381  1.000000e+00 -0.03750993 -0.02206218 -0.03667633 -0.03855381
## Ozono        0.044175248  0.083640605 -3.750993e-02  1.00000000  0.59612683  0.08179858  0.47947091
## Pres_Alt     0.337931827  0.158080640 -2.206218e-02  0.59612683  1.00000000 -0.23161673  0.03869121
## Vel_Viento  -0.226893006 -0.046090839 -3.667633e-02  0.08179858 -0.23161673  1.00000000  0.30356343
## Humedad     -0.034727288 -0.064739863 -3.855381e-02  0.47947091  0.03869121  0.30356343  1.00000000
## T_Sandburg   0.235445072  0.157156363 -3.035349e-02  0.77335204  0.80633038  0.04122208  0.33132296
## T_ElMonte    0.314323892  0.117127229 -2.481044e-02  0.76001956  0.89689385 -0.06983510  0.21158607
## Inv_Alt_b    0.045305170 -0.082352709  7.998485e-02 -0.55196217 -0.50891157  0.12834881 -0.24703914
## Grad_Pres   -0.218837079 -0.111239793  3.418479e-02  0.17391799 -0.24549047  0.37328762  0.62433536
## Inv_T_b      0.236540625  0.127530054 -5.365959e-02  0.71756186  0.85642134 -0.12959891  0.19101936
## Visibilidad -0.167796386 -0.057896954 -8.572216e-06 -0.47629112 -0.34272720  0.04534341 -0.45750232
##              T_Sandburg   T_ElMonte   Inv_Alt_b   Grad_Pres     Inv_T_b   Visibilidad
## Mes          0.23544507  0.31432389  0.04530517 -0.21883708  0.23654062 -1.677964e-01
## DiaMes       0.15715636  0.11712723 -0.08235271 -0.11123979  0.12753005 -5.789695e-02
## DiaSemana   -0.03035349 -0.02481044  0.07998485  0.03418479 -0.05365959 -8.572216e-06
## Ozono        0.77335204  0.76001956 -0.55196217  0.17391799  0.71756186 -4.762911e-01
## Pres_Alt     0.80633038  0.89689385 -0.50891157 -0.24549047  0.85642134 -3.427272e-01
## Vel_Viento   0.04122208 -0.06983510  0.12834881  0.37328762 -0.12959891  4.534341e-02
## Humedad      0.33132296  0.21158607 -0.24703914  0.62433536  0.19101936 -4.575023e-01
## T_Sandburg   1.00000000  0.91396229 -0.51539621  0.11765666  0.84310310 -4.103864e-01
## T_ElMonte    0.91396229  1.00000000 -0.57965832 -0.12091597  0.93080989 -4.389790e-01
## Inv_Alt_b   -0.51539621 -0.57965832  1.00000000  0.11350236 -0.78286145  3.966979e-01
## Grad_Pres    0.11765666 -0.12091597  0.11350236  1.00000000 -0.20663872 -1.200549e-01
## Inv_T_b      0.84310310  0.93080989 -0.78286145 -0.20663872  1.00000000 -4.377177e-01
## Visibilidad -0.41038641 -0.43897902  0.39669789 -0.12005488 -0.43771768  1.000000e+00
```

- Correlaciones parciales:

```r
partial.r(OzonoLA)
```

```
##                    Mes       DiaMes     DiaSemana       Ozono    Pres_Alt  Vel_Viento     Humedad
## Mes          1.000000000 -0.01473632 -0.029646884 -0.239632308 -0.008364478 -0.19289804  0.16086022
## DiaMes      -0.014736319  1.00000000  0.017131467  0.023224469  0.074079502  0.01519492 -0.03992322
## DiaSemana   -0.029646884  0.01713147  1.000000000 -0.015463849 -0.014083279 -0.05267203 -0.05035826
## Ozono       -0.239632308  0.02322447 -0.015463849  1.000000000 -0.134822542 -0.04003920  0.26277407
## Pres_Alt    -0.008364478  0.07407950 -0.014083279 -0.134822542  1.000000000 -0.29270094 -0.09532118
## Vel_Viento  -0.192898039  0.01519492 -0.052672027 -0.040039195 -0.292700944  1.00000000  0.15651029
## Humedad      0.160860221 -0.03992322 -0.050358261  0.262774072 -0.095321178  0.15651029  1.00000000
## T_Sandburg   0.008578204  0.20842819 -0.037515653  0.141155532  0.108888567  0.08938736 -0.04472740
## T_ElMonte    0.131026789 -0.12847809  0.050717722  0.312487718  0.344311253  0.11902520 -0.04353431
## Inv_Alt_b    0.230043843 -0.02868566  0.036820690 -0.111064127  0.120880379  0.11170466 -0.05762633
## Grad_Pres   -0.127208517 -0.13665426  0.068684046  0.001780773 -0.044096421  0.05542912  0.50554293
## Inv_T_b      0.048692150 -0.02999001 -0.008230412 -0.076866881  0.140848869  0.01217894  0.06712657
## Visibilidad -0.108506988 -0.06279200 -0.037003418 -0.074160846  0.014979648  0.11148387 -0.32142715
##              T_Sandburg   T_ElMonte   Inv_Alt_b    Grad_Pres      Inv_T_b Visibilidad
## Mes          0.008578204  0.13102679  0.23004384 -0.127208517  0.048692150 -0.10850699
## DiaMes       0.208428191 -0.12847809 -0.02868566 -0.136654263 -0.029990011 -0.06279200
## DiaSemana   -0.037515653  0.05071772  0.03682069  0.068684046 -0.008230412 -0.03700342
## Ozono        0.141155532  0.31248772 -0.11106413  0.001780773 -0.076866881 -0.07416085
## Pres_Alt     0.108888567  0.34431125  0.12088038 -0.044096421  0.140848869  0.01497965
```

```
## Vel_Viento   0.089387359  0.11902520  0.11170466  0.055429122  0.012178940  0.11148387
## Humedad     -0.044727403 -0.04353431 -0.05762633  0.505542925  0.067126570 -0.32142715
## T_Sandburg   1.000000000  0.35489823  0.18928541  0.498084949  0.229456614  0.08539386
## T_ElMonte    0.354898232  1.00000000  0.39942102 -0.051952353  0.579597071 -0.12200008
## Inv_Alt_b    0.189285412  0.39942102  1.00000000 -0.155715887 -0.818841765  0.09905698
## Grad_Pres    0.498084949 -0.05195235 -0.15571589  1.000000000 -0.326942874  0.01948577
## Inv_T_b      0.229456614  0.57959707 -0.81884177 -0.326942874  1.000000000  0.03558761
## Visibilidad  0.085393863 -0.12200008  0.09905698  0.019485768  0.035587611  1.00000000
```

3. Modelo matemático

$$\mathbb{E}(\vec{Y}|\boldsymbol{X}) = \beta_0 + \sum_{i=1}^{n} \beta_i X_{ij} \tag{1}$$

```
MOD_FULL <- lm(Ozono~., data=OzonoLA)
MOD_FULL
```

```
##
## Call:
## lm(formula = Ozono ~ ., data = OzonoLA)
##
## Coefficients:
## (Intercept)          Mes       DiaMes    DiaSemana     Pres_Alt   Vel_Viento      Humedad   T_Sandbu
##  55.4279486   -0.3431326    0.0120308   -0.0473689   -0.0133495   -0.0959961    0.0880372    0.13662
##   T_ElMonte    Inv_Alt_b    Grad_Pres      Inv_T_b  Visibilidad
##   0.5597690   -0.0006176    0.0003624   -0.1244500   -0.0049469
```

```
coef(MOD_FULL)
```

```
##   (Intercept)           Mes        DiaMes     DiaSemana      Pres_Alt    Vel_Viento       Humedad
## 55.4279486216 -0.3431325880  0.0120307523 -0.0473688814 -0.0133495197 -0.0959961221  0.0880371866
##    T_Sandburg     T_ElMonte     Inv_Alt_b     Grad_Pres       Inv_T_b   Visibilidad
##  0.1366230525  0.5597690142 -0.0006175971  0.0003623595 -0.1244500321 -0.0049468590
```

Ozono_i = 55.428 - 0.343Mes_i + 0.012$Diames_i$ - 0.047$DiaSemana_i$ - 0.0133$Pres_Alt_i$ - 0.096Vel_Viento_i + 0.088$Humedad_i$ + 0.1366$T_Sandburg_i$ + 0.5598$T_ElMonte_i$ - 0.0006$Inv_Alt_b_i$ + 0.0004$Grad_Pres_i$ - 0.124$Inv_T_b_i$ - 0.005$Visibilidad_i$

Suma de residuos al cuadrado media:

```
( MSSR <- summary(MOD_FULL)$sigma^2 )
```

```
## [1] 19.24102
```

Grados de libertad de los residuos:

```
( gl.R <- MOD_FULL$df )
```

```
## [1] 190
```

Número de parámetros:

```
( gl.E <- MOD_FULL$rank )
```

```
## [1] 13
```

4. Análisis de multicolinealidad

```r
summary(MOD_FULL)
```

```
##
## Call:
## lm(formula = Ozono ~ ., data = OzonoLA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0342  -2.8582  -0.4764   2.6584  12.7160
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 55.4279486 37.6060409   1.474 0.142161
## Mes         -0.3431326  0.1008551  -3.402 0.000815 ***
## DiaMes       0.0120308  0.0375710   0.320 0.749158
## DiaSemana   -0.0473689  0.2222014  -0.213 0.831415
## Pres_Alt    -0.0133495  0.0071178  -1.876 0.062255 .
## Vel_Viento  -0.0959961  0.1737974  -0.552 0.581361
## Humedad      0.0880372  0.0234515   3.754 0.000231 ***
## T_Sandburg   0.1366231  0.0695151   1.965 0.050828 .
## T_ElMonte    0.5597690  0.1234488   4.534 1.02e-05 ***
## Inv_Alt_b   -0.0006176  0.0004009  -1.540 0.125116
## Grad_Pres    0.0003624  0.0147623   0.025 0.980443
## Inv_T_b     -0.1244500  0.1171095  -1.063 0.289275
## Visibilidad -0.0049469  0.0048259  -1.025 0.306638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.386 on 190 degrees of freedom
## Multiple R-squared:  0.7304, Adjusted R-squared:  0.7133
## F-statistic: 42.89 on 12 and 190 DF,  p-value: < 2.2e-16
```

Obtenemos que muchos de los coeficientes son no significativos, por lo que debemos hacer una selección de las variables. No obstante, como esto se puede deber a la presencia de multicolinealidad, vamos a analizarla.

Para ello, utilizaremos la librería "mctest", que proporciona un análisis completo de multicolinealidad:

```r
library(mctest)
mctest(MOD_FULL, type="o")
```

```
##
## Call:
## omcdiag(mod = mod, Inter = TRUE, detr = detr, red = red, conf = conf,
##     theil = theil, cn = cn)
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.0001         1
## Farrar Chi-Square:      1900.8790         1
## Red Indicator:             0.3656         0
## Sum of Lambda Inverse:    85.6887         1
## Theil's Method:           -1.2174         0
## Condition Number:        586.6642         1
```

```
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

Este test proporciona 6 medidas, de las cuales 4 indican que estamos ante un caso en el que la multicolinealidad está presente.

Para solucionar esto y conseguir un ajuste correcto, sobre el que hacer inferencia debemos hacer una selección de variables.

5. Selección del modelo

Para hacer la selección del modelo, utilizaremos la selección sistemática por STEPWISE, utilizando como criterio el AIC del modelo. Elegimos este método de selección por ser el mejor, al permitir incluir y eliminar variables a lo largo del proceso.

Primero, definimos el modelo con solo el intercept.

```
Mod_NULL <- lm(Ozono ~ 1, data = OzonoLA)
```

Ahora, aplicaremos la siguiente función para obtener el modelo óptimo:

```
stepMod <- step(Mod_NULL, direction = "both", trace = 1,
                scope = list(lower = Mod_NULL,
                             upper = MOD_FULL) )
```

```
## Start:  AIC=854.91
## Ozono ~ 1
##
##                Df Sum of Sq     RSS    AIC
## + T_Sandburg    1    8108.8  5449.4 671.88
## + T_ElMonte     1    7831.6  5726.6 681.95
## + Inv_T_b       1    6981.0  6577.1 710.06
## + Pres_Alt      1    4818.1  8740.0 767.78
## + Inv_Alt_b     1    4130.7  9427.5 783.15
## + Humedad       1    3116.9 10441.2 803.88
## + Visibilidad   1    3075.7 10482.4 804.68
## + Grad_Pres     1     410.1 13148.0 850.68
## <none>                      13558.1 854.91
## + DiaMes        1      94.8 13463.3 855.49
## + Vel_Viento    1      90.7 13467.4 855.55
## + Mes           1      26.5 13531.7 856.52
## + DiaSemana     1      19.1 13539.1 856.63
##
## Step:  AIC=671.88
## Ozono ~ T_Sandburg
##
##                Df Sum of Sq     RSS    AIC
## + Humedad       1     759.0  4690.4 643.43
## + Inv_Alt_b     1     434.3  5015.0 657.02
## + Visibilidad   1     411.8  5037.6 657.93
## + Mes           1     273.0  5176.4 663.45
## + T_ElMonte     1     233.1  5216.3 665.01
## + Inv_T_b       1     201.4  5247.9 666.23
## + Grad_Pres     1      94.5  5354.8 670.33
## <none>                      5449.4 671.88
## + Vel_Viento    1      33.8  5415.5 672.62
```

```
## + Pres_Alt     1      29.2  5420.2 672.79
## + DiaMes       1      20.0  5429.4 673.14
## + DiaSemana    1       2.7  5446.7 673.78
## - T_Sandburg   1    8108.8 13558.1 854.91
##
## Step:  AIC=643.43
## Ozono ~ T_Sandburg + Humedad
##
##               Df Sum of Sq     RSS    AIC
## + T_ElMonte    1     505.3  4185.1 622.29
## + Inv_T_b      1     371.8  4318.5 628.67
## + Inv_Alt_b    1     335.7  4354.6 630.35
## + Mes          1     175.2  4515.2 637.70
## + Visibilidad  1     116.1  4574.2 640.34
## + Grad_Pres    1      92.0  4598.4 641.41
## <none>                      4690.4 643.43
## + Pres_Alt     1      41.5  4648.9 643.63
## + Vel_Viento   1       7.8  4682.6 645.09
## + DiaMes       1       1.0  4689.3 645.39
## + DiaSemana    1       0.6  4689.7 645.40
## - Humedad      1     759.0  5449.4 671.88
## - T_Sandburg   1    5750.9 10441.2 803.88
##
## Step:  AIC=622.29
## Ozono ~ T_Sandburg + Humedad + T_ElMonte
##
##               Df Sum of Sq    RSS    AIC
## + Mes          1    358.12 3827.0 606.13
## + Inv_Alt_b    1    126.22 4058.9 618.08
## + Pres_Alt     1    108.61 4076.5 618.96
## <none>                     4185.1 622.29
## + Visibilidad  1     19.69 4165.4 623.34
## + Inv_T_b      1     18.92 4166.2 623.37
## + Grad_Pres    1     11.28 4173.8 623.75
## + Vel_Viento   1      3.68 4181.4 624.11
## + DiaMes       1      1.50 4183.6 624.22
## + DiaSemana    1      0.65 4184.4 624.26
## - T_Sandburg   1    100.19 4285.3 625.10
## - T_ElMonte    1    505.29 4690.4 643.43
## - Humedad      1   1031.23 5216.3 665.01
##
## Step:  AIC=606.13
## Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes
##
##               Df Sum of Sq    RSS    AIC
## + Pres_Alt     1     70.84 3756.1 604.34
## <none>                     3827.0 606.13
## + Inv_Alt_b    1     34.70 3792.3 606.28
## + Visibilidad  1     34.59 3792.4 606.29
## - T_Sandburg   1     63.90 3890.9 607.50
## + Vel_Viento   1      2.21 3824.8 608.02
## + DiaMes       1      1.48 3825.5 608.06
## + Inv_T_b      1      1.30 3825.7 608.07
## + Grad_Pres    1      0.91 3826.1 608.09
```

```
## + DiaSemana    1      0.74 3826.2 608.09
## - Mes          1    358.12 4185.1 622.29
## - T_ElMonte    1    688.22 4515.2 637.70
## - Humedad      1    946.87 4773.8 649.01
##
## Step:  AIC=604.34
## Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes + Pres_Alt
##
##               Df Sum of Sq    RSS    AIC
## + Inv_Alt_b    1     41.91 3714.2 604.06
## <none>                     3756.1 604.34
## + Visibilidad  1     36.56 3719.6 604.36
## + Vel_Viento   1     18.08 3738.0 605.36
## + Inv_T_b      1      6.40 3749.7 606.00
## + DiaMes       1      3.86 3752.3 606.13
## - Pres_Alt     1     70.84 3827.0 606.13
## - T_Sandburg   1     72.62 3828.7 606.23
## + DiaSemana    1      0.92 3755.2 606.29
## + Grad_Pres    1      0.07 3756.1 606.34
## - Mes          1    320.34 4076.5 618.96
## - T_ElMonte    1    664.43 4420.6 635.41
## - Humedad      1    678.82 4434.9 636.07
##
## Step:  AIC=604.06
## Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes + Pres_Alt + Inv_Alt_b
##
##               Df Sum of Sq    RSS    AIC
## <none>                     3714.2 604.06
## - Inv_Alt_b    1     41.91 3756.1 604.34
## + Inv_T_b      1     26.12 3688.1 604.63
## + Visibilidad  1     25.74 3688.5 604.65
## + Vel_Viento   1      8.67 3705.5 605.59
## + DiaMes       1      2.73 3711.5 605.91
## + Grad_Pres    1      1.61 3712.6 605.98
## + DiaSemana    1      0.19 3714.0 606.05
## - Pres_Alt     1     78.05 3792.3 606.28
## - T_Sandburg   1     87.87 3802.1 606.81
## - Mes          1    228.30 3942.5 614.17
## - T_ElMonte    1    515.95 4230.2 628.47
## - Humedad      1    596.56 4310.8 632.30
```

```
summary((stepMod))
```

```
##
## Call:
## lm(formula = Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes +
##     Pres_Alt + Inv_Alt_b, data = OzonoLA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0749  -3.0474  -0.1831   2.7775  12.6395
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 51.3444845 35.1290934   1.462 0.145454
```

```
## T_Sandburg   0.1242673  0.0577088   2.153 0.032513 *
## Humedad      0.0975694  0.0173897   5.611 6.80e-08 ***
## T_ElMonte    0.4743962  0.0909164   5.218 4.59e-07 ***
## Mes         -0.3324536  0.0957810  -3.471 0.000638 ***
## Pres_Alt    -0.0134013  0.0066034  -2.029 0.043763 *
## Inv_Alt_b   -0.0003211  0.0002159  -1.487 0.138571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.353 on 196 degrees of freedom
## Multiple R-squared:  0.7261, Adjusted R-squared:  0.7177
## F-statistic: 86.58 on 6 and 196 DF,  p-value: < 2.2e-16
```

El modelo resultante de la selección secuencial es: Ozono_i = 51.3444845 - 0.3324536Mes_i - 0.0134013 $Pres\_Alt\_i$ + 0.0975694$Humedad_i$ + 0.1242673$T_Sandburg_i$ + 0.4743962$T_ElMonte_i$ - 0.0003211$Inv_Alt_b_i$

No obstante, con un 10% de significación, la variable Inv_Alt_b no es significativa, por lo que examinaremos si se debe excluir del modelo:

```
ajuste_sin_inv_alt_b <- update(stepMod, .~.-Inv_Alt_b)
```

Lo comprobaremos con un anova de modelos anidados:

```
anova(ajuste_sin_inv_alt_b, stepMod)
```

```
## Analysis of Variance Table
##
## Model 1: Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes + Pres_Alt
## Model 2: Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes + Pres_Alt + Inv_Alt_b
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    197 3756.1
## 2    196 3714.2  1    41.913 2.2117 0.1386
```

Prueba no significativa, por lo que nos quedamos con el modelo sin la variable.

```
ajuste <- ajuste_sin_inv_alt_b
```

Comprobaremos si es mejor que el modelo completo, utilizando un anova de modelos anidados:

```
anova(ajuste, MOD_FULL)
```

```
## Analysis of Variance Table
##
## Model 1: Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes + Pres_Alt
## Model 2: Ozono ~ Mes + DiaMes + DiaSemana + Pres_Alt + Vel_Viento + Humedad +
##     T_Sandburg + T_ElMonte + Inv_Alt_b + Grad_Pres + Inv_T_b +
##     Visibilidad
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    197 3756.1
## 2    190 3655.8  7    100.33 0.7449 0.6342
```

El resultado es no significativo, por lo que la selección ha merecido la pena.

6. Posible Interacción

Debido a la posible necesidad de interacción, decidimos probar si un modelo que incluya interacción es mejor que nuestro modelo completo.

Comenzamos definiendo este modelo, con todas las interacciones posibles:

```r
ajuste_completo <- glm(Proximidad~., data = Oro, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
ajuste.i <- update(ajuste_completo,.~.^3, family=binomial, data=Oro)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(ajuste.i)
```

```
##
## Call:
## glm(formula = Proximidad ~ As + Sb + Corredor + As:Sb + As:Corredor +
##     Sb:Corredor + As:Sb:Corredor, family = binomial, data = Oro)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9714   0.0000   0.0000   0.0000   1.9345
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -8.939  34483.934   0.000    1.000
## As                 -47.382 105299.858   0.000    1.000
## Sb                 -33.817 196896.288   0.000    1.000
## Corredor1            9.617  34483.934   0.000    1.000
## As:Sb               47.999  60183.576   0.001    0.999
## As:Corredor1        46.489 105299.858   0.000    1.000
## Sb:Corredor1        26.827 196896.289   0.000    1.000
## As:Sb:Corredor1    -44.627  60183.576  -0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 87.7202  on 63  degrees of freedom
## Residual deviance:  7.5068  on 56  degrees of freedom
## AIC: 23.507
##
## Number of Fisher Scoring iterations: 21
```

Ningún coeficiente es significativo, por lo que consideramos que esto se puede deber a la presencia de multicolinealidad debido a las interacciones.

Decidimos hacer una selección de variables, por si alguna interacción entre variables originales resultase significativa. La haremos igual que en el apartado anterior:

```r
step(M0, direction="forward", trace=1,
     scope = list(lower=M0,upper=ajuste.i))
```

```
## Start:  AIC=89.72
## Proximidad ~ 1
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##            Df Deviance    AIC
## + As        1   22.603 26.603
## + Sb        1   45.332 49.332
## + Corredor  1   45.848 49.848
## <none>          87.720 89.720
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=26.6
## Proximidad ~ As

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##            Df Deviance    AIC
## + Sb        1   18.306 24.306
## + Corredor  1   19.990 25.990
## <none>          22.603 26.603

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=24.31
## Proximidad ~ As + Sb

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##            Df Deviance    AIC
## + Corredor  1   14.194 22.194
## <none>          18.306 24.306
## + As:Sb     1   17.249 25.249

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=22.19
## Proximidad ~ As + Sb + Corredor

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance    AIC
## <none>             14.194 22.194
## + Sb:Corredor  1   12.253 22.253
## + As:Sb        1   12.688 22.688
## + As:Corredor  1   14.137 24.137

##
## Call:  glm(formula = Proximidad ~ As + Sb + Corredor, family = "binomial",
##     data = Oro)
##
## Coefficients:
## (Intercept)           As           Sb    Corredor1
##      -7.610        1.205        1.421        3.197
##
## Degrees of Freedom: 63 Total (i.e. Null);  60 Residual
## Null Deviance:       87.72
## Residual Deviance: 14.19     AIC: 22.19
```

Finalmente, vemos que en este caso, la interacción de las variables no aporta nada a nuestro ajuste.

7. Inferencia modelo

Ahora ya podemos comenzar la inferencia.

```
summary(ajuste)
```

```
##
## Call:
## lm(formula = Ozono ~ T_Sandburg + Humedad + T_ElMonte + Mes +
##     Pres_Alt, data = OzonoLA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7435  -2.9604   0.0761   2.9540  12.5572
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45.210973  34.993284   1.292   0.1979
## T_Sandburg   0.111767   0.057269   1.952   0.0524 .
## Humedad      0.102313   0.017147   5.967 1.11e-08 ***
## T_ElMonte    0.514331   0.087127   5.903 1.54e-08 ***
## Mes         -0.375442   0.091596  -4.099 6.06e-05 ***
## Pres_Alt    -0.012738   0.006609  -1.928   0.0554 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.367 on 197 degrees of freedom
## Multiple R-squared:  0.723,  Adjusted R-squared:  0.7159
## F-statistic: 102.8 on 5 and 197 DF,  p-value: < 2.2e-16
```

Las únicas variables que parecen ser significativas son Mes, Humedad y T_ElMonte. También podemos considerar que son bastante significativas, pero no tanto, las variables T_Sandburg y Pres_Alt. Por otra parte, según el coeficiente de bondad, con este ajuste podemos explicar el 73,04% de la variabilidad de los datos. Por último, gracias a la última linea del summary deducimos que es mejor este ajuste en comparación al modelo que contiene únicamente el intercept, debido al p-valor $<$ 2.2e-16.

8. Validación modelo seleccionado

Por abreviar la notación, tenemos:

```
MS <- ajuste  # Ajuste modelo elegido.
MC <- MOD_FULL # Ajuste modelo completo
```

Primero, calculamos el coeficiente de robusted del ajuste:

```
library(DAAG)
( B2 <- sum(residuals(MS)^2)/press(MS) )
```

```
## [1] 0.9442346
```

Elevado y superior al del modelo completo

```
sum(residuals(MC)^2)/press(MC)
```

```
## [1] 0.8823052
```

Haremos una validación del tipo LOOCV (Leave One Out Cross Validation):

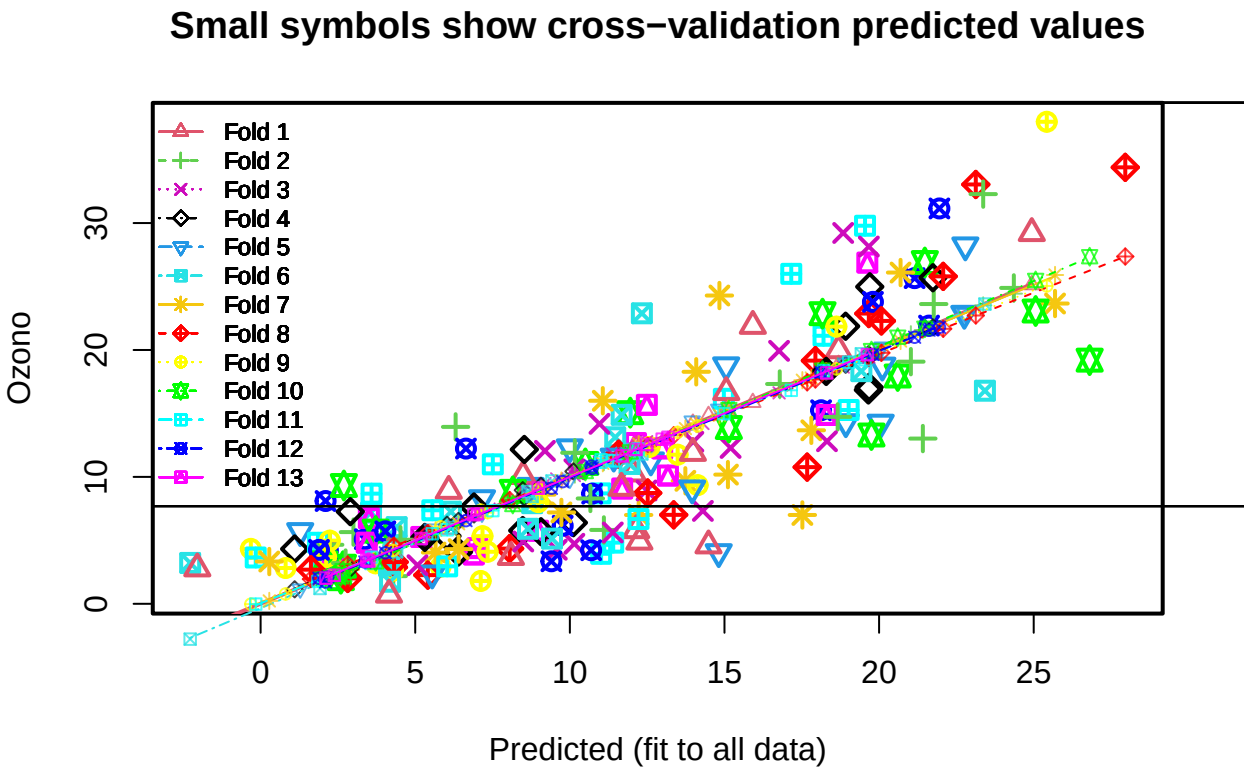
Primero, para MS:

```r
class(OzonoLA) # ya es un data frame
```

```
## [1] "data.frame"
```

```r
set.seed(5198)
cv_k3_MS <- cv.lm(data=OzonoLA,form.lm= formula(MS),m=length(OzonoLA))
```

```
## Warning in cv.lm(data = OzonoLA, form.lm = formula(MS), m = length(OzonoLA)):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 15
##                    13        14       26        32        37       52        62        68        80
## Predicted    8.094337 12.249041 12.14553 12.040563 -2.046856 6.083821  8.485786 15.916928 18.688605
## cvpred       7.928546 12.448259 12.50498 12.175213 -2.091523 5.861667  8.638667 15.858296 18.855042
## Ozono        3.690000  4.900000  5.80000 10.270000  2.790000 8.900000 10.070000 21.900000 19.980000
## CV residual -4.238546 -7.548259 -6.70498 -1.905213  4.881523 3.038333  1.431333  6.041704  1.124958
##                   146       149       160       177       203
## Predicted    14.48427 15.057249 11.668834 13.978194  4.152568
## cvpred       14.83124 15.318839 11.467194 14.219525  4.540370
## Ozono         4.60000 16.680000  9.140000 11.890000  0.720000
## CV residual -10.23124  1.361161 -2.327194 -2.329525 -3.820370
##
## Sum of squares = 345.01    Mean square = 23    n = 15
##
```

```
## fold 2
## Observations in test set: 16
##                    20       40        51         69         83        94       114       123       1
## Predicted    4.289757 3.001376  6.305743 16.7880673 24.3579150 10.161111 21.415164 11.100279 21.7667
## cvpred       4.047500 3.000899  6.044460 16.8250824 24.4386217 10.180638 21.572202 11.159753 21.8423
## Ozono        2.180000 5.650000 13.940000 17.3200000 24.8900000 11.900000 13.020000  5.820000 23.6200
## CV residual -1.867500 2.649101  7.895540  0.4949176  0.4513783  1.719362 -8.552202 -5.339753  1.7776
##                   129       133       134       176       186       199       202
## Predicted   23.36661 21.028438 18.660654 10.663269 2.234342 2.7478914 4.5226442
## cvpred      23.51191 21.238023 18.726176 10.753336 2.109278 2.7259569 4.5451227
## Ozono       32.28000 19.080000 14.730000  8.300000 4.650000 3.2100000 5.0500000
## CV residual  8.76809 -2.158023 -3.996176 -2.453336 2.540722 0.4840431 0.5048773
##
## Sum of squares = 291.53    Mean square = 18.22    n = 16
##
## fold 3
## Observations in test set: 16
##                     7        18         27       28        49        54       101       122       137
## Predicted   10.145667 15.198364 10.1461421 13.99231 12.481257  9.194827 16.765867  9.577206 18.83929
## cvpred      10.327833 15.224939 10.1021928 13.90347 12.774365  9.468645 16.619596  9.838587 18.65523
## Ozono        4.730000 12.280000 10.6000000 12.77000  8.930000 12.050000 19.930000  4.260000 29.22000
## CV residual -5.597833 -2.944939  0.4978072 -1.13347 -3.844365  2.581355  3.310404 -5.578587 10.56477
##                   143       155       166       167       168       184
## Predicted   14.302186 19.669771 11.388138  8.49343 10.957545  5.072194
## cvpred      14.209082 19.436405 11.787592  8.99322 11.169943  5.381118
## Ozono        7.320000 28.150000  5.620000  4.91000 14.180000  3.040000
## CV residual -6.889082  8.713595 -6.167592 -4.08322  3.010057 -2.341118
##
## Sum of squares = 438.43    Mean square = 27.4    n = 16
##
## fold 4
## Observations in test set: 16
##                     1         2         6        16        24       38        43       87       109
## Predicted    6.0272008  8.476699 10.114204  9.066450  6.395332 1.101268 6.9002835 19.65340 18.918020
## cvpred       6.2316117  8.955479 10.430674  9.301757  6.529657 1.142138 7.0397051 19.66444 18.833878
## Ozono        5.3400000  5.770000  6.390000  5.680000  4.080000 4.320000 7.6300000 16.85000 21.870000
## CV residual -0.8916117 -3.185479 -4.040674 -3.621757 -2.449657 3.177862 0.5902949 -2.81444  3.036122
##                   124       128       136         152       189        193
## Predicted    8.527456 21.740586 19.678253 18.28949833 2.901051 5.30822110
## cvpred       8.386945 21.779251 19.502586 18.23452174 2.769524 5.16059425
## Ozono       12.160000 25.690000 17.060000 18.31000000 7.260000 5.23000000
## CV residual  3.773055  3.910749 -2.442586  0.07547826 4.490476 0.06940575
##
## Sum of squares = 159.05    Mean square = 9.94    n = 16
##
## fold 5
## Observations in test set: 16
##                    11        19       25       29        41        55        75        77        78
## Predicted    14.81354 8.7324036 7.158984 1.280414 2.5217442  9.998969 15.053715 12.619515  5.548119
## cvpred       15.34371 8.7472667 7.149033 1.173599 2.2954502  9.819741 15.154112 12.521553  5.900876
## Ozono         4.07000 9.2900000 8.320000 5.730000 3.0100000 12.330000 18.790000 11.300000  2.390000
## CV residual -11.27371 0.5427333 1.170967 4.556401 0.7145498  2.510259  3.635888 -1.221553 -3.510876
##                    99       106       111        118       121       178
## Predicted   22.75617745 20.051111 22.789989 11.4187177 20.104689 13.955777
```

```
## cvpred      22.87034274 20.152481 22.936382 11.4109341 20.239842 14.409956
## Ozono       22.85000000 14.270000 28.240000 11.6000000 18.770000  9.090000
## CV residual -0.02034274 -5.882481  5.303618  0.1890659 -1.469842 -5.319956
##
## Sum of squares = 298.29    Mean square = 18.64    n = 16
##
## fold 6
## Observations in test set: 16
##                    3       31       34        36       39         56        60        67       70
## Predicted   1.920606 4.399085 12.32896 -2.278041 3.570676  8.7810354 11.467473 11.675348 6.153595 23
## cvpred      1.208680 4.100656 11.80690 -2.783057 3.180221  8.4552205 11.270727 11.467761 6.015144 23
## Ozono       3.690000 6.040000 22.89000  3.220000 7.190000  7.9300000 13.120000 14.890000 7.260000 16
## CV residual 2.481320 1.939344 11.08310  6.003057 4.009779 -0.5252205  1.849273  3.422239 1.244856 -6
##                   138       144       148       175       183       200
## Predicted   19.427262 11.931665  9.422161  8.646439  4.150858  4.185553
## cvpred      19.696974 12.159854  9.678004  8.846143  4.179757  4.416318
## Ozono       18.330000 11.020000  5.140000  5.910000  3.010000  1.740000
## CV residual -1.366974 -1.139854 -4.538004 -2.936143 -1.169757 -2.676318
##
## Sum of squares = 289.34    Mean square = 18.08    n = 16
##
## fold 7
## Observations in test set: 16
##                     10         22        30        46        50         53        71       105
## Predicted   12.225524  2.7442540  5.657275 14.836293 15.111495  9.0952917 13.735185 25.690079 20.694
## cvpred      12.592491  2.9789799  5.788564 14.859092 15.379864  9.0000181 13.811725 25.908061 20.707
## Ozono        7.000000  2.7400000  4.040000 24.290000 10.180000  8.6000000  9.690000 23.660000 26.100
## CV residual -5.592491 -0.2389799 -1.748564  9.430908 -5.199864 -0.4000181 -4.121725 -2.248061  5.392
##                   119       154       157       162       169       188       197
## Predicted   17.808745  17.51887 14.087396  9.727227 11.065375  6.402289 0.2796873
## cvpred      17.787173  17.61701 14.161069  9.710282 11.074217  6.305709 0.2271546
## Ozono       13.670000   7.00000 18.280000  7.200000 16.000000  4.310000 3.3300000
## CV residual -4.117173 -10.61701  4.118931 -2.510282  4.925783 -1.995709 3.1028454
##
## Sum of squares = 392.47    Mean square = 24.53    n = 16
##
## fold 8
## Observations in test set: 16
##                      8        35        66         79       82       88        90        98       104
## Predicted    8.053183  5.406914  4.296046 11.5765500 23.12714 17.94095 12.517893 17.674731 27.961562
## cvpred       8.184036  5.577890  4.533929 11.5300011 22.71633 17.67660 12.446652 17.450143 27.364771
## Ozono        4.350000  2.260000  2.880000 11.7900000 33.04000 19.16000  8.730000 10.770000 34.390000
## CV residual -3.834036 -3.317890 -1.653929  0.2599989 10.32367  1.48340 -3.716652 -6.680143  7.025229
##                   126       135       179       187       192       201
## Predicted   20.074647 22.073606 13.358434  4.316830  2.816824 1.6256739
## cvpred      19.774315 21.668364 13.279167  4.577333  3.102041 1.9485752
## Ozono       22.290000 25.800000  7.010000  3.290000  2.000000 2.6900000
## CV residual  2.515685  4.131636 -6.269167 -1.287333 -1.102041 0.7414248
##
## Sum of squares = 323.38    Mean square = 20.21    n = 16
##
## fold 9
## Observations in test set: 16
##                     23          58        93       117      130        153       156       161
```

```
## Predicted    5.821553 -0.31621478  7.117428 14.133350 25.42285 12.5794175 18.620086 13.485664  4.2848
## cvpred       6.075590 -0.02684803  7.383264 14.015243 25.04355 12.2398938 18.505847 13.511314  4.2053
## Ozono        2.920000  4.33000000  1.800000  9.350000 37.98000 12.3600000 21.840000 11.750000  2.6100
## CV residual -3.155590  4.35684803 -5.583264 -4.665243 12.93645  0.1201062  3.334153 -1.761314 -1.5953
##                  165       172       173      181        182      190      195
## Predicted    8.981806  7.175938  7.341584 0.808824  3.7122724 2.237552 2.119022
## cvpred       9.016919  7.248433  7.177546 0.778277  3.6065147 2.210584 2.068595
## Ozono        8.010000  5.330000  4.100000 2.820000  3.1900000 4.980000 3.680000
## CV residual -1.006919 -1.918433 -3.077546 2.041723 -0.4165147 2.769416 1.611405
##
## Sum of squares = 294.78    Mean square = 18.42    n = 16
##
## fold 10
## Observations in test set: 15
##                     17        33         42       59       73        96       103       107      131
## Predicted   10.5005827 11.953390  2.5946826 2.692560 3.721097 21.479187 20.602838 19.753000 25.06198
## cvpred      10.4251573 11.969931  2.1307498 2.307281 3.309010 21.708719 21.005845 19.951867 25.47833
## Ozono       11.0600000 15.060000  1.9800000 9.320000 5.730000 26.890000 17.950000 13.300000 23.07000
## CV residual  0.6348427  3.090069 -0.1507498 7.012719 2.420990  5.181281 -3.055845 -6.651867 -2.40833
##                  140       141       159      191       194
## Predicted   8.154963 18.170346 15.141476 2.064364 2.6622221
## cvpred      7.824355 18.409577 15.299591 1.956526 2.6030435
## Ozono       8.860000 22.860000 13.890000 3.230000 2.9600000
## CV residual 1.035645  4.450423 -1.409591 1.273474 0.3569565
##
## Sum of squares = 242.25    Mean square = 16.15    n = 15
##
## fold 11
## Observations in test set: 15
##                      9        45       63          64       74        76        86        89       91
## Predicted   11.011052 10.996926 1.853942 -0.15812249 3.587772 18.203266 11.402628 14.980089 12.21948
## cvpred      11.344612 11.149834 1.766860 -0.02451469 3.668608 18.356396 11.278811 14.979919 12.21746
## Ozono        3.940000  8.700000 4.810000  3.65000000 8.680000 21.120000  4.820000 16.150000  6.68000
## CV residual -7.404612 -2.449834 3.043140  3.67451469 5.011392  2.763604 -6.458811  1.170081 -5.53746
##                  150       151       164       174       185
## Predicted   17.159701 19.54534 5.552941  7.508760  6.033263
## cvpred      16.823109 19.22993 5.523302  7.343206  5.833228
## Ozono       26.000000 29.79000 7.370000 10.990000  2.950000
## CV residual  9.176891 10.56007 1.846698  3.646794 -2.883228
##
## Sum of squares = 425.04    Mean square = 28.34    n = 15
##
## fold 12
## Observations in test set: 15
##                      15       48       61        72        84        85        95       102        116
## Predicted    9.768927 2.100862 3.369735  6.640398 21.950700 10.721300 21.149123 18.124340 21.6153762
## cvpred       9.627881 1.721019 3.292343  6.556732 21.787628 10.667742 21.002762 18.222198 21.5626454
## Ozono        6.150000 8.100000 5.090000 12.230000 31.150000  8.680000 25.660000 15.250000 21.9200000
## CV residual -3.477881 6.378981 1.797657  5.673268  9.362372 -1.987742  4.657238 -2.972198  0.3573546
##                  139       147       180       196       198
## Predicted    9.403464 10.689221 2.174727 4.033710 1.887455
## cvpred       9.104369 10.846232 2.130822 4.015462 2.011241
## Ozono        3.350000  4.220000 4.200000 5.710000 4.250000
## CV residual -5.754369 -6.626232 2.069178 1.694538 2.238759
```

```
##
## Sum of squares = 316.8    Mean square = 21.12    n = 15
##
## fold 13
## Observations in test set: 15
##                     4         5        12        21        44         47        57       65       81
## Predicted    6.892439  8.978100  7.011831 2.1112054 12.489117 12.1398415 11.691632 3.499894 19.61933
## cvpred       7.028784  9.115879  7.121245 2.0296193 12.559004 12.1678026 11.664949 3.377166 19.63630
## Ozono        3.890000  5.760000  4.390000 2.9400000 15.680000 12.6700000  9.090000 6.760000 26.89000
## CV residual -3.138784 -3.355879 -2.731245 0.9103807  3.120996  0.5021974 -2.574949 3.382834  7.25370
##                   125        145       158       170       171
## Predicted   18.305004 12.9865216 13.171724 3.432383 2.3829847
## cvpred      18.200942 12.9734517 13.120489 3.454883 2.2531721
## Ozono       14.880000 12.2500000 10.110000 4.820000 2.9000000
## CV residual -3.320942 -0.7234517 -3.010489 1.365117 0.6468279
##
## Sum of squares = 133    Mean square = 8.87    n = 15
##
## Overall (Sum over all 15 folds)
##       ms
## 19.45499
```

Se calcula la raíz cuadrada de la media de los cuadrados de las diferencias entre predicciones y observaciones:

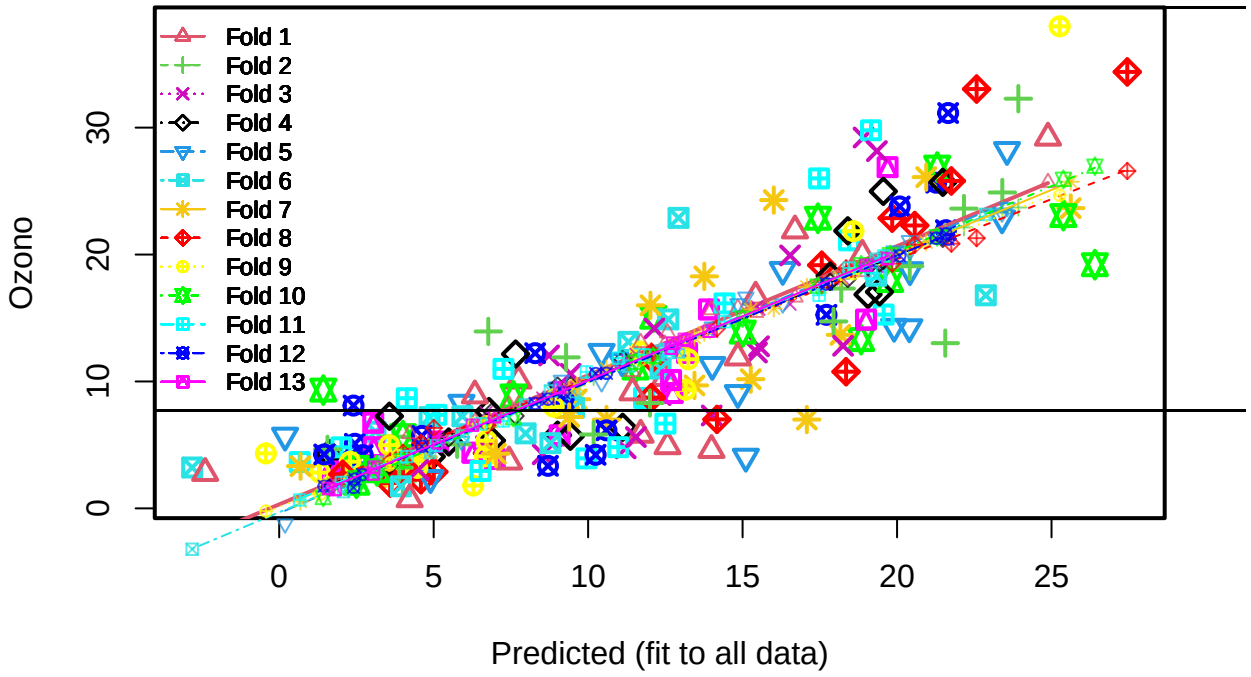
```
errores <- cv_k3_MS$cvpred - cv_k3_MS$Ozono # predicho por cv - predicción real
( error_cv_k3_MS <- sqrt(mean(errores^2)) ) # estimador RMSE (raiz media suma residuos al cuadrado)
```

```
## [1] 4.410781
```

Finalmente, para MC:

```
set.seed(5198)
cv_k3_MC <- cv.lm(data=OzonoLA,form.lm=formula(MC),m=length(OzonoLA))
```

```
## Warning in cv.lm(data = OzonoLA, form.lm = formula(MC), m = length(OzonoLA)):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 15
##                    13        14        26        32        37       52        62        68        80
## Predicted    7.440524 12.570232 11.700192 12.481891 -2.396811 6.332799  7.744519 16.695646 18.876144
## cvpred       8.153843 13.829313 13.062716 12.921002 -3.355326 6.788498  8.150007 16.643705 18.650796
## Ozono        3.690000  4.900000  5.800000 10.270000  2.790000 8.900000 10.070000 21.900000 19.980000
## CV residual -4.463843 -8.929313 -7.262716 -2.651002  6.145326 2.111502  1.919993  5.256295  1.329204
##                   146       149       160       177       203
## Predicted    13.99472 15.42288 11.436518 14.844856  4.222157
## cvpred       15.64258 15.43417 10.779545 15.770028  5.884059
## Ozono         4.60000 16.68000  9.140000 11.890000  0.720000
## CV residual -11.04258  1.24583 -1.639545 -3.880028 -5.164059
##
## Sum of squares = 415.17    Mean square = 27.68    n = 15
##
## fold 2
## Observations in test set: 16
##                    20       40        51         69        83        94       114       123       127
## Predicted    4.001573 3.698056  6.767830 18.1923441 23.406972  9.285851 21.56123 10.178986 22.16576
## cvpred       3.938462 3.759339  6.400434 18.0794809 23.527989  9.271308 21.82530 10.353186 22.14034
## Ozono        2.180000 5.650000 13.940000 17.3200000 24.890000 11.900000 13.02000  5.820000 23.62000
## CV residual -1.758462 1.890661  7.539566 -0.7594809  1.362011  2.628692 -8.80530 -4.533186  1.47966
##                   133       134       176       186        199       202
## Predicted   20.410373 17.945077 12.002891 1.562951  3.8866690  5.7592135
## cvpred      20.662457 18.076038 11.995393 1.740678  3.8349488  5.6766484
## Ozono       19.080000 14.730000  8.300000 4.650000  3.2100000  5.0500000
## CV residual -1.582457 -3.346038 -3.695393 2.909322 -0.6249488 -0.6266484
##
```

```
## Sum of squares = 283.22    Mean square = 17.7    n = 16
##
## fold 3
## Observations in test set: 16
##                     7        18        27        28        49        54       101       122      137
## Predicted   11.215916 15.465775  9.436245 15.531555 12.347251  8.734629 16.536223  8.529697 18.91607
## cvpred      11.777745 15.887994  9.404568 15.689903 12.176068  8.761862 16.088214  8.695824 18.72904
## Ozono        4.730000 12.280000 10.600000 12.770000  8.930000 12.050000 19.930000  4.260000 29.22000
## CV residual -7.047745 -3.607994  1.195432 -2.919903 -3.246068  3.288138  3.841786 -4.435824 10.49096
##                   143       155       166       167       168       184
## Predicted   14.000615 19.347816 11.546461  6.694141 12.142312  4.504243
## cvpred      13.947722 19.063656 12.145192  7.550493 12.330433  5.106427
## Ozono        7.320000 28.150000  5.620000  4.910000 14.180000  3.040000
## CV residual -6.627722  9.086344 -6.525192 -2.640493  1.849567 -2.066427
##
## Sum of squares = 449.9    Mean square = 28.12    n = 16
##
## fold 4
## Observations in test set: 16
##                     1         2         6        16         24       38        43        87       109
## Predicted    6.873243  9.009641 11.117544  9.423672  4.9232014 1.622697 6.7912501 19.060610 18.40529
## cvpred       7.604531  9.681745 11.756068  9.702572  4.9079724 1.636612 6.9089036 19.039561 18.07529
## Ozono        5.340000  5.770000  6.390000  5.680000  4.0800000 4.320000 7.6300000 16.850000 21.87000
## CV residual -2.264531 -3.911745 -5.366068 -4.022572 -0.8279724 2.683388 0.7210964 -2.189561  3.79470
##                   110       124       128       136        152       189        193
## Predicted   19.555138  7.652500 21.48177 19.431825 17.8343745 3.567057  5.4861801
## cvpred      19.495939  7.281611 21.23630 19.309081 17.8576265 3.347544  5.5191732
## Ozono       24.980000 12.160000 25.69000 17.060000 18.3100000 7.260000  5.2300000
## CV residual  5.484061  4.878389  4.45370 -2.249081  0.4523735 3.912456 -0.2891732
##
## Sum of squares = 187.37    Mean square = 11.71    n = 16
##
## fold 5
## Observations in test set: 16
##                    11         19       25         29        41        55        75        77
## Predicted    15.10421  9.1336905 5.899857  0.1895359 2.2981852 10.436443 16.299885 14.018014  4.89854
## cvpred       16.64953 10.1034438 5.288593 -1.2187132 2.3202081  9.903537 16.114479 14.296348  4.77980
## Ozono         4.07000  9.2900000 8.320000  5.7300000 3.0100000 12.330000 18.790000 11.300000  2.39000
## CV residual -12.57953 -0.8134438 3.031407  6.9487132 0.6897919  2.426463  2.675521 -2.996348 -2.38980
##                    97         99       106       111        118       121       178
## Predicted   19.904221 23.3981119 20.397884 23.55944 11.0902103 20.417815 14.845379
## cvpred      20.286428 23.2796716 21.108762 23.70526 11.1936349 20.276326 16.129878
## Ozono       14.310000 22.8500000 14.270000 28.24000 11.6000000 18.770000  9.090000
## CV residual -5.976428 -0.4296716 -6.838762  4.53474  0.4063651 -1.506326 -7.039878
##
## Sum of squares = 399.82    Mean square = 24.99    n = 16
##
## fold 6
## Observations in test set: 16
##                     3       31       34        36       39        56        60        67       70
## Predicted   2.081988 3.117250 12.92545 -2.817305 4.866487  9.541240 11.305685 12.596424 5.930993 22.8
## cvpred      1.321883 2.932481 12.32884 -3.220900 4.400694  9.199472 11.214526 12.280983 5.686535 23.1
## Ozono       3.690000 6.040000 22.89000  3.220000 7.190000  7.930000 13.120000 14.890000 7.260000 16.7
## CV residual 2.368117 3.107519 10.56116  6.440900 2.789306 -1.269472  1.905474  2.609017 1.573465 -6.3
```

```
##                   138       144       148       175        183       200
## Predicted   19.325532 12.344723  8.779764  7.975095  3.3944500  3.940621
## cvpred      19.632142 12.384675  9.249603  8.159317  3.6304783  4.016822
## Ozono       18.330000 11.020000  5.140000  5.910000  3.0100000  1.740000
## CV residual -1.302142 -1.364675 -4.109603 -2.249317 -0.6204783 -2.276822
##
## Sum of squares = 262.29    Mean square = 16.39    n = 16
##
## fold 7
## Observations in test set: 16
##                    10         22         30        46        50         53        71        105
## Predicted   10.594280  3.4312358  4.6632159 16.009568 15.264726  9.633851 13.442198 25.622650 20.9444
## cvpred      11.002061  3.6307681  5.0326898 15.751721 15.678303  9.694815 13.448517 25.739853 20.9569
## Ozono        7.000000  2.7400000  4.0400000 24.290000 10.180000  8.600000  9.690000 23.660000 26.1000
## CV residual -4.002061 -0.8907681 -0.9926898  8.538279 -5.498303 -1.094815 -3.758517 -2.079853  5.1430
##                   119       154       157       162       169       188       197
## Predicted   18.16524  17.07908 13.761385  9.386545 12.017325  7.013945 0.6901916
## cvpred      18.16042  17.45495 13.841444  9.377751 11.795743  7.041877 0.5420600
## Ozono       13.67000   7.00000 18.280000  7.200000 16.000000  4.310000 3.3300000
## CV residual -4.49042 -10.45495  4.438556 -2.177751  4.204257 -2.731877 2.7879400
##
## Sum of squares = 353.86    Mean square = 22.12    n = 16
##
## fold 8
## Observations in test set: 16
##                     8        35        66         79       82        88        90        98       104
## Predicted    6.739690  4.586791  5.004902 12.0791409 22.57508 17.562176 12.047731 18.349383 27.45264
## cvpred       6.545341  5.159333  6.333695 12.1911483 21.29180 17.567112 12.360251 18.910913 26.58481
## Ozono        4.350000  2.260000  2.880000 11.7900000 33.04000 19.160000  8.730000 10.770000 34.39000
## CV residual -2.195341 -2.899333 -3.453695 -0.4011483 11.74820  1.592888 -3.630251 -8.140913  7.80518
##                   113       126       135       179       187       192       201
## Predicted   19.842784 20.573563 21.757166 14.169995  4.008337  3.591144 2.0571540
## cvpred      19.288651 20.780328 20.855775 14.187255  4.382413  3.705436 2.4157289
## Ozono       22.870000 22.290000 25.800000  7.010000  3.290000  2.000000 2.6900000
## CV residual  3.581349  1.509672  4.944225 -7.177255 -1.092413 -1.705436 0.2742711
##
## Sum of squares = 401.49    Mean square = 25.09    n = 16
##
## fold 9
## Observations in test set: 16
##                    23         58        93       117       130        153       156       161
## Predicted    3.839587 -0.4308846  6.281735 13.155307 25.26400 11.6791979 18.588454 13.244652  3.6072
## cvpred       4.154127 -0.2118862  6.664557 13.108784 24.66399 11.6941871 18.494933 13.285754  3.5938
## Ozono        2.920000  4.3300000  1.800000  9.350000 37.98000 12.3600000 21.840000 11.750000  2.6100
## CV residual -1.234127  4.5418862 -4.864557 -3.758784 13.31601  0.6658129  3.345067 -1.535754 -0.9838
##                   165       172       173       181        182       190       195
## Predicted    8.8837739  6.710209  6.661469 1.318443  4.4697715 3.564775 2.291775
## cvpred       8.9481276  6.831756  6.527415 1.089418  4.1187978 3.420703 2.321299
## Ozono        8.0100000  5.330000  4.100000 2.820000  3.1900000 4.980000 3.680000
## CV residual -0.9381276 -1.501756 -2.427415 1.730582 -0.9287978 1.559297 1.358701
##
## Sum of squares = 269.38    Mean square = 16.84    n = 16
##
## fold 10
```

```
## Observations in test set: 15
##                      17       33         42        59       73        96       103       107       13
## Predicted   11.5364232 12.10590  2.5007446 1.4263141 4.000462 21.299458 19.776046 18.837340 25.38220
## cvpred      11.5187686 11.97613  2.4263812 0.8591298 3.754198 21.398705 19.989241 19.190002 25.92737
## Ozono       11.0600000 15.06000  1.9800000 9.3200000 5.730000 26.890000 17.950000 13.300000 23.07000
## CV residual -0.4587686  3.08387 -0.4463812 8.4608702 1.975802  5.491295 -2.039241 -5.890002 -2.85737
##                     132      140       141       159       191        194
## Predicted   26.402518 7.583810 17.439592 15.002898 2.4738283  3.1660519
## cvpred      26.961438 7.143562 17.509398 14.983142 2.2911007  3.1850573
## Ozono       19.200000 8.860000 22.860000 13.890000 3.2300000  2.9600000
## CV residual -7.761438 1.716438  5.350602 -1.093142 0.9388993 -0.2250573
##
## Sum of squares = 256.52    Mean square = 17.1    n = 15
##
## fold 11
## Observations in test set: 15
##                       9        45       63        64       74        76        86        89        91
## Predicted    9.943837 11.809295 2.011643 0.6636727 4.129266 18.445270 10.975906 14.411125 12.501165
## cvpred      10.751684 12.098156 2.038901 0.6654128 4.087515 18.933147 11.840547 14.713225 12.524334
## Ozono        3.940000  8.700000 4.810000 3.6500000 8.680000 21.120000  4.820000 16.150000  6.680000
## CV residual -6.811684 -3.398156 2.771099 2.9845872 4.592485  2.186853 -7.020547  1.436775 -5.844334
##                     150      151      164       174       185
## Predicted   17.477059 19.15705 5.084568  7.258785  6.518272
## cvpred      16.797102 18.53135 4.954099  6.889160  6.655734
## Ozono       26.000000 29.79000 7.370000 10.990000  2.950000
## CV residual  9.202898 11.25865 2.415901  4.100840 -3.705734
##
## Sum of squares = 455.1    Mean square = 30.34    n = 15
##
## fold 12
## Observations in test set: 15
##                      15       48       61        72        84         85       95      102        116
## Predicted   10.592034 2.405465 2.447951  8.284350 21.654136  9.2063365 21.26473 17.71477 21.5816040
## cvpred      10.643806 1.652116 2.309982  8.272994 21.208759  9.1158536 21.28625 17.67977 21.6362454
## Ozono        6.150000 8.100000 5.090000 12.230000 31.150000  8.6800000 25.66000 15.25000 21.9200000
## CV residual -4.493806 6.447884 2.780018  3.957006  9.941241 -0.4358536  4.37375 -2.42977  0.2837546
##                     139       147      180       196       198
## Predicted    8.700437 10.243076 2.703505 4.6184035 1.454674
## cvpred       8.603268 10.618833 2.661679 4.8832021 1.743392
## Ozono        3.350000  4.220000 4.200000 5.7100000 4.250000
## CV residual -5.253268 -6.398833 1.538321 0.8267979 2.506608
##
## Sum of squares = 302.28    Mean square = 20.15    n = 15
##
## fold 13
## Observations in test set: 15
##                       4         5        12       21        44         47        57       65        81
## Predicted    6.976510  9.076506  6.253474 1.604130 13.912924 13.1096748 12.736124 3.044066 19.693546
## cvpred       7.228818  9.521362  6.557988 1.572653 13.956235 13.3257148 13.006213 2.718416 19.683888
## Ozono        3.890000  5.760000  4.390000 2.940000 15.680000 12.6700000  9.090000 6.760000 26.890000
## CV residual -3.338818 -3.761362 -2.167988 1.367347  1.723765 -0.6557148 -3.916213 4.041584  7.206112
##                      92       125        145       158       170       171
## Predicted   5.14694398 18.994232 13.2040056 12.685143 3.060696 1.816891
## cvpred      5.18736734 19.161706 13.2412375 12.627759 3.251042 1.485275
```

```
## Ozono       5.27000000 14.880000 12.2500000 10.110000 4.820000 2.900000
## CV residual 0.08263266 -4.281706 -0.9912375 -2.517759 1.568958 1.414725
##
## Sum of squares = 148.99    Mean square = 9.93    n = 15
##
## Overall (Sum over all 15 folds)
##       ms
## 20.61771
```

```
errores <- cv_k3_MC$cvpred - cv_k3_MC$Ozono
( error_cv_k3_MC <- sqrt(mean(errores^2)) )
```

```
## [1] 4.540672
```

```
par(mfrow=c(1,1))
```

Obtenemos un comportamiento mejor con el MS que con MC, pues tenemos un menor error.

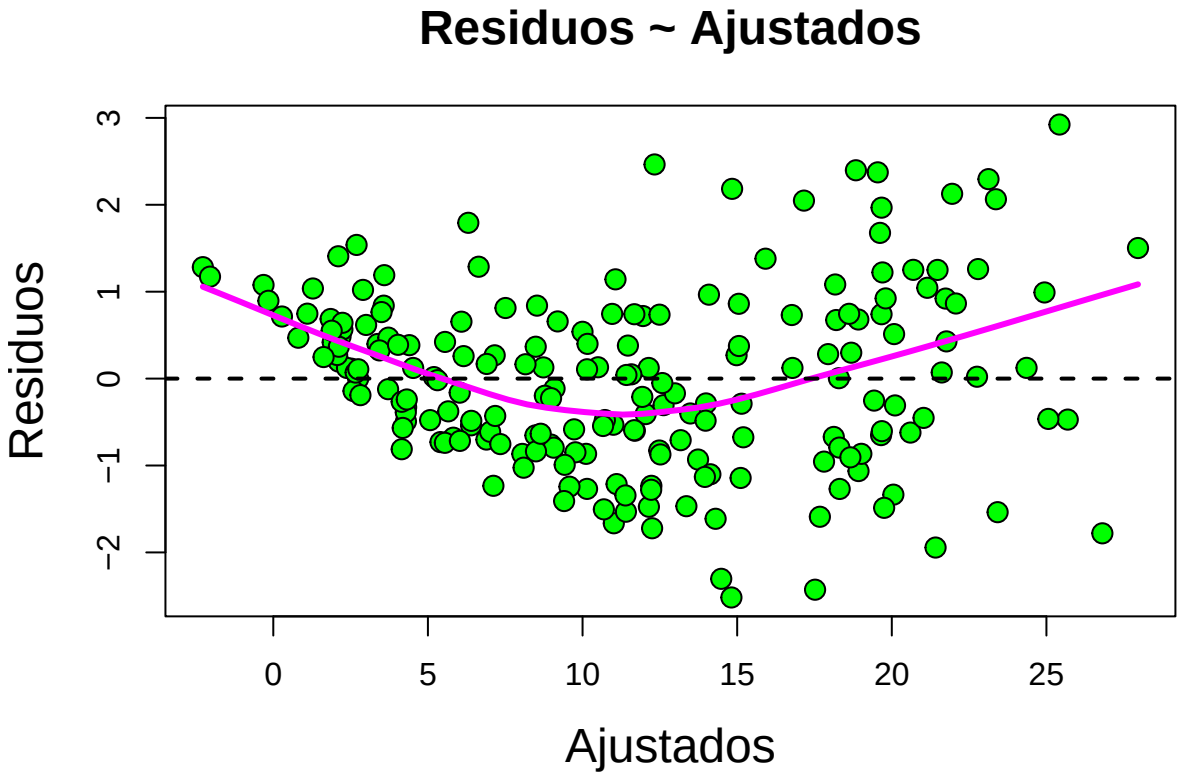
9. Análisis de residuos modelo seleccionado

Para realizar el análisis de los residuos usaremos los residuos estandarizados

```
library(MASS)
res.est  <- stdres(ajuste)
```

- Linealidad:

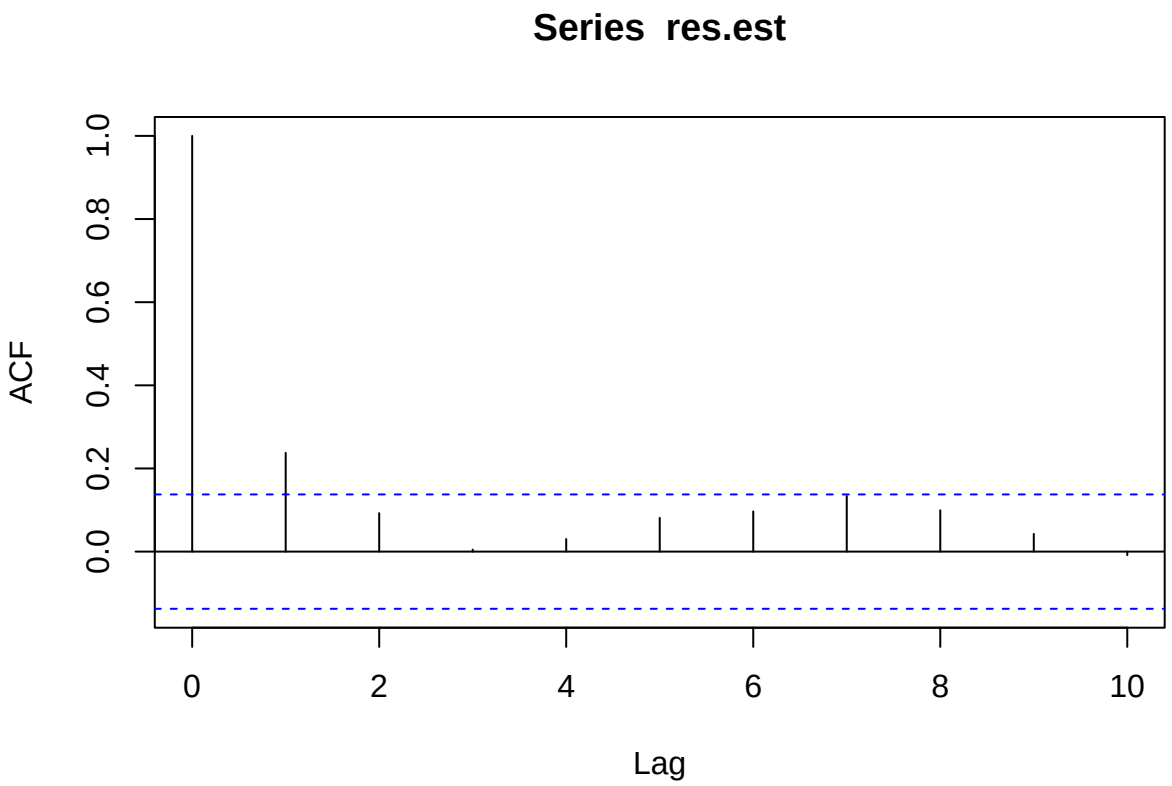
```
scatter.smooth(ajuste$fit, res.est, main="Residuos ~ Ajustados",
               xlab="Ajustados",ylab="Residuos", pch = 21,
               bg = "green", cex.lab=1.5, cex=1.4, cex.main=1.5,
               lpars = list(col = "magenta", lwd = 3) )
abline(h=0,lty=2,lwd=2)
```



Como podemos observar en el gráfico no sería correcto afirmar linealidad

-Aleatoriedad:

```
acf(res.est, lag.max = 10, type = "correlation")$acf
```




```
## , , 1
##
##               [,1]
##  [1,]  1.000000000
##  [2,]  0.237527376
##  [3,]  0.092555682
##  [4,]  0.004735799
##  [5,]  0.030258389
##  [6,]  0.081295983
##  [7,]  0.096738673
##  [8,]  0.133778400
##  [9,]  0.099382832
## [10,]  0.042686424
## [11,] -0.008619252
```

Como podemos ver en la matriz de correlaciones, las correlaciones entre un dato y el anterior son muy bajas, con lo cual, si sería correcto asimir aleatoriedad,

- Normalidad:

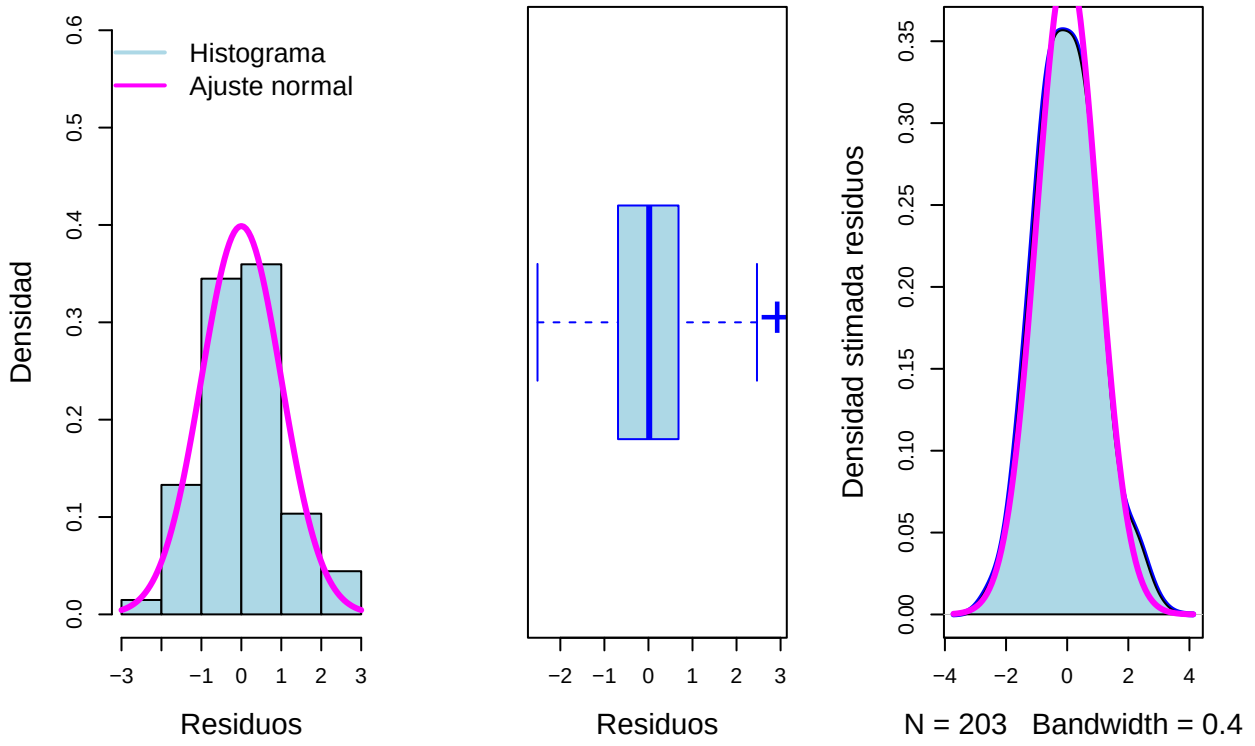
```
par(mfrow=c(1,3))

hist(res.est, breaks=6,freq=FALSE, main = "", xlab="Residuos", cex.lab=1.4,
     ylab = "Densidad", col = "lightblue", ylim=c(0,0.6))
curve( dnorm(x), col="magenta", lwd=3, add=TRUE)
etiquetas <- c("Histograma","Ajuste normal")
```

```
legend("topright",etiquetas, lwd=2, col=c("lightblue","magenta"),
       lty=c(1,1), cex=1.3, inset=0.02, box.lty=0)

boxplot(res.est, main = "", xlab="Residuos",
        cex.lab=1.4, border = "blue", col= "lightblue", pch="+",
        horizontal = TRUE, cex=3)

plot(density(res.est, bw=0.4),main="",lwd=3,col="blue",
     ylab="Densidad stimada residuos", cex.lab=1.4, cex.lab=1.4)
polygon(density(res.est,bw=0.4), col="lightblue")
curve( dnorm(x), col="magenta", lwd=3, add=TRUE)
```




```
par(mfrow=c(1,1))
```

Gráficamente podemos deducir que nuestros datos no siguen exactamente una distribución normal pero si muy semejante.

```
library(nortest)
lillie.test(res.est)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  res.est
## D = 0.043464, p-value = 0.4597
```

```
cvm.test(res.est)
```

```
##
##  Cramer-von Mises normality test
##
## data:  res.est
```

```
## W = 0.049805, p-value = 0.5113
```

```
ad.test(res.est)
```

```
##
##  Anderson-Darling normality test
##
## data:  res.est
## A = 0.371, p-value = 0.4201
```

```
shapiro.test(res.est)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.est
## W = 0.99263, p-value = 0.4023
```

Analíticamente se confirma la teoría anterior ya que los p-vlores, en todos los tests nos dan lo suficientemente grandes cómo para no rechazar la hipótesis nula, la cual se refiere a la normalidad.

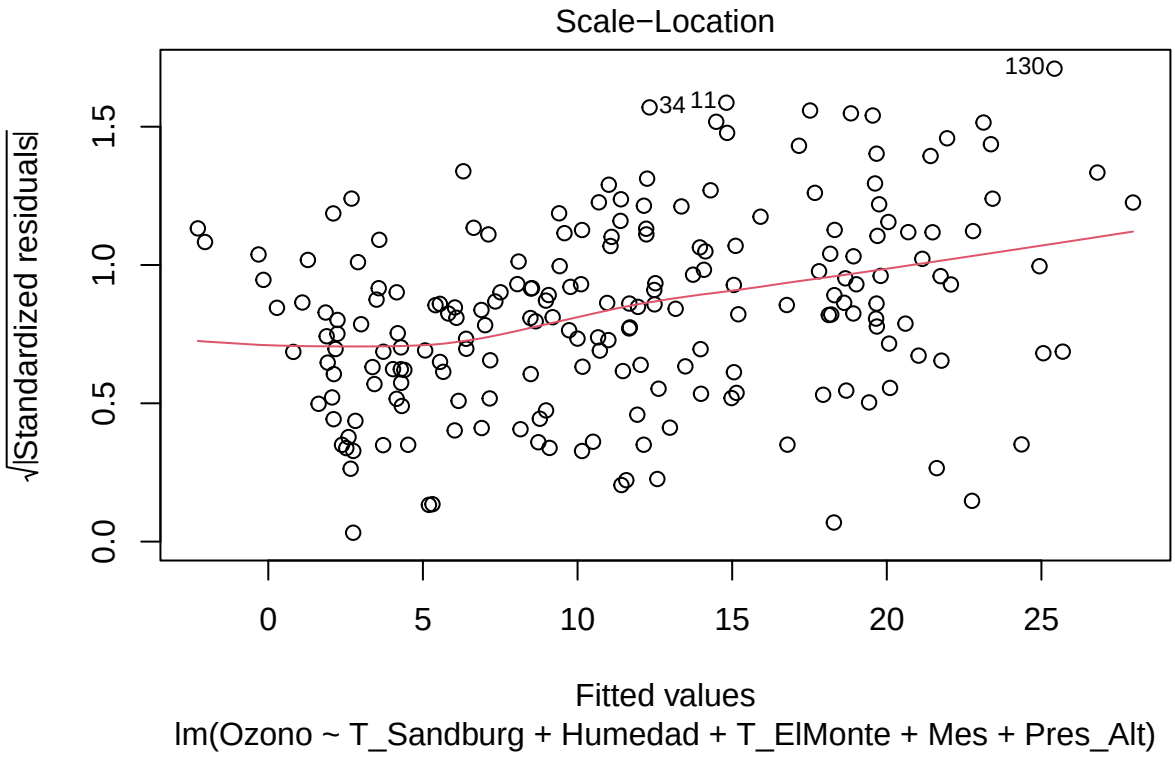
- Homoscedasticidad:

H0: sigmaˆ2=cte vs H1: sigmaˆ2!=cte Test de Breusch-Pagan

```
library(lmtest)
bptest(ajuste)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ajuste
## BP = 30.592, df = 5, p-value = 1.127e-05
```

```
plot(ajuste, which=3)
```



Tanto con el test de Breusch-Pagan cómo con el gráfico de los residuos podemos concluir que se rechaza la hipótesis nula de homoscedasticidad

10. Análisis de influencia modelo seleccionado

```
influencia <- influence(ajuste)
```

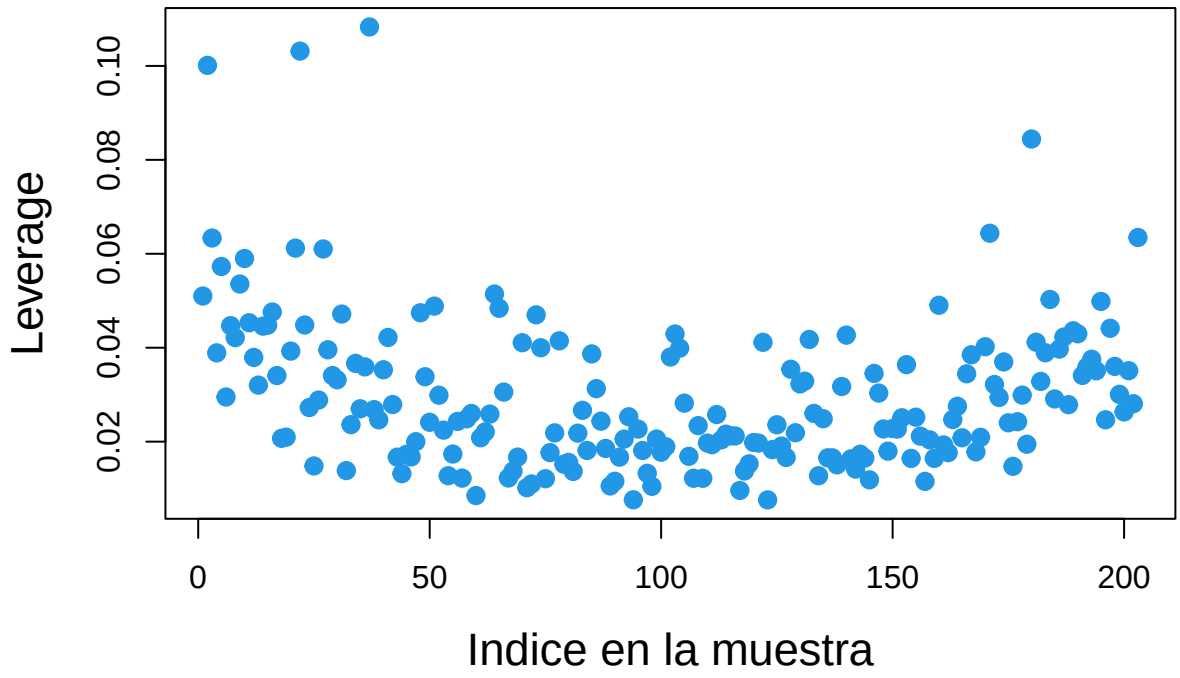
- Leverages:

```
( lev <- influencia$hat )
```

```
##           1           2           3           4           5           6           7           8
## 0.051004533 0.100118110 0.063353326 0.038901236 0.057292132 0.029504367 0.044699289 0.042142597 0.053
##          10          11          12          13          14          15          16          17
## 0.058999848 0.045316050 0.037927053 0.032030216 0.044591122 0.044794620 0.047601639 0.034072047 0.020
##          19          20          21          22          23          24          25          26
## 0.020958123 0.039285066 0.061202903 0.103157934 0.044837489 0.027256843 0.014820471 0.028854153 0.061
##          28          29          30          31          32          33          34          35
## 0.039561314 0.034070028 0.033172749 0.047170000 0.013831179 0.023618798 0.036665945 0.026994586 0.035
##          37          38          39          40          41          42          43          44
## 0.108294004 0.026819690 0.024642938 0.035307834 0.042179044 0.027902374 0.016700080 0.013177610 0.017
##          46          47          48          49          50          51          52          53
## 0.016751144 0.019989681 0.047443016 0.033818580 0.024123642 0.048854298 0.029884637 0.022446719 0.011
##          55          56          57          58          59          60          61          62
## 0.017358486 0.024301573 0.012231780 0.024858222 0.025995466 0.008517176 0.020824328 0.022069875 0.025
##          64          65          66          67          68          69          70          71
## 0.051427381 0.048398000 0.030547447 0.012262030 0.013718236 0.016704886 0.041066459 0.010184037 0.010
##          73          74          75          76          77          78          79          80
## 0.046995569 0.039992959 0.012106927 0.017653974 0.021866160 0.041464701 0.015184470 0.015573086 0.013
##          82          83          84          85          86          87          88          89
## 0.021803173 0.026667344 0.018124672 0.038679833 0.031289356 0.024355628 0.018552378 0.010559547 0.011
```

```
##          91          92          93          94          95          96          97          98
## 0.016692669 0.020536664 0.025292281 0.007618958 0.022668846 0.018112380 0.013260597 0.010488328 0.020
##         100         101         102         103         104         105         106         107
## 0.017752125 0.018907211 0.038021897 0.042915452 0.039894195 0.028194448 0.016875515 0.012195945 0.023
##         109         110         111         112         113         114         115         116
## 0.012198702 0.019699758 0.019350208 0.025767425 0.020380083 0.021589260 0.021220480 0.021209660 0.009
##         118         119         120         121         122         123         124         125
## 0.013752753 0.015282239 0.019815852 0.019696594 0.041152943 0.007599460 0.018294206 0.023591584 0.019
##         127         128         129         130         131         132         133         134
## 0.016633226 0.035407575 0.021889407 0.032290170 0.032876522 0.041757398 0.026020614 0.012744226 0.024
##         136         137         138         139         140         141         142         143
## 0.016547995 0.016550445 0.015056617 0.031743561 0.042676052 0.016325904 0.014151822 0.017323007 0.010
##         145         146         147         148         149         150         151         152
## 0.011866087 0.034525849 0.030343243 0.022697343 0.017971587 0.022791841 0.022679458 0.025080923 0.030
##         154         155         156         157         158         159         160         161
## 0.016425299 0.025206772 0.021162962 0.011536468 0.020383495 0.016435645 0.049035767 0.019303646 0.017
##         163         164         165         166         167         168         169         170
## 0.024706279 0.027531394 0.020820168 0.034444497 0.038500137 0.017818551 0.020918925 0.040207313 0.064
##         172         173         174         175         176         177         178         179
## 0.032161389 0.029407861 0.036972329 0.024015601 0.014739854 0.024243207 0.029908747 0.019439987 0.084
##         181         182         183         184         185         186         187         188
## 0.041168346 0.032818537 0.038919592 0.050277147 0.029092469 0.039706808 0.042279588 0.027870849 0.043
##         190         191         192         193         194         195         196         197
## 0.042959948 0.034105787 0.035912223 0.037546203 0.035075934 0.049857798 0.024654985 0.044136868 0.030
##         199         200         201         202         203
## 0.030067272 0.026318070 0.035107863 0.028076025 0.063466155
```

```
plot(lev, xlab = "Indice en la muestra", ylab = "Leverage",
     cex = 1.2, pch=19, col=4, cex.lab=1.4)
```

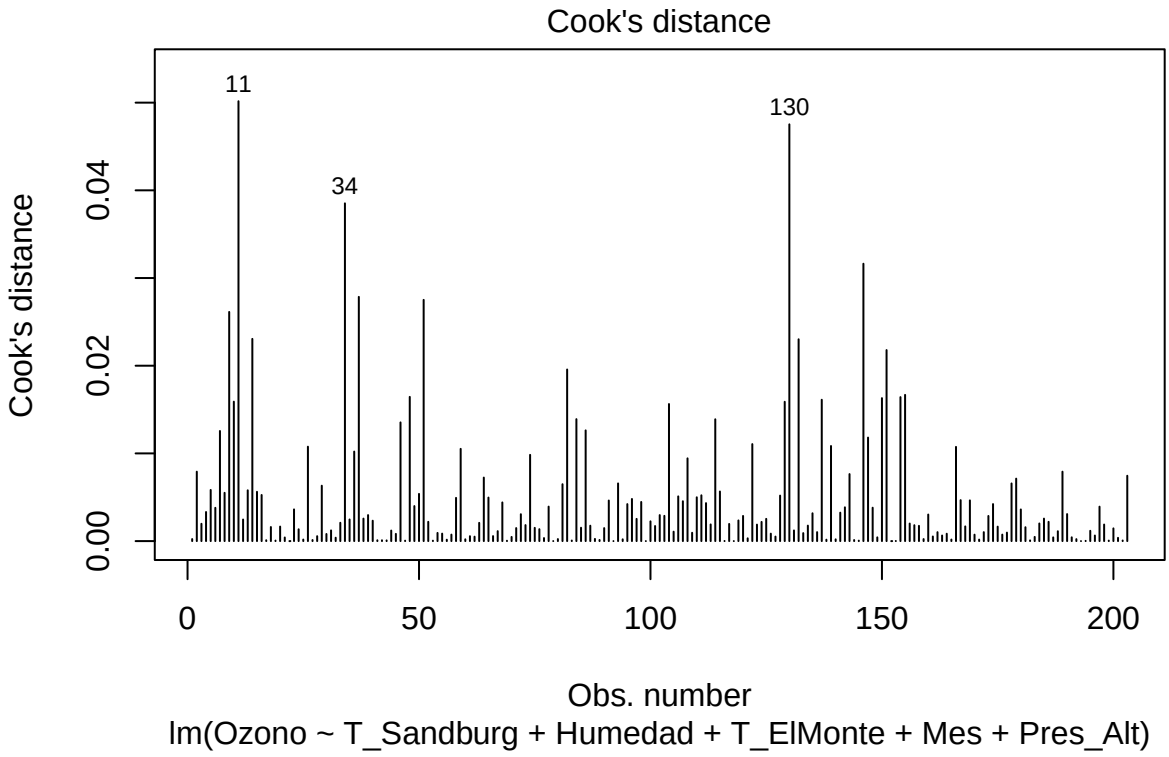


- Distancias de Cook:

```
cooks.distance(ajuste)
```

```
##            1            2            3            4            5            6            7
## 2.337883e-04 7.917652e-03 1.976251e-03 3.318567e-03 5.835991e-03 3.797878e-03 1.255738e-02 5.506109e-
##            9           10           11           12           13           14           15
## 2.613318e-02 1.590399e-02 5.016515e-02 2.462166e-03 5.796584e-03 2.306248e-02 5.620396e-03 5.260765e-
##           17           18           19           20           21           22           23
## 9.989770e-05 1.607351e-03 5.942438e-05 1.656068e-03 4.169627e-04 2.028807e-08 3.616785e-03 1.349835e-
##           25           26           27           28           29           30           31
## 1.799214e-04 1.076836e-02 1.246229e-04 5.601068e-04 6.319682e-03 8.113835e-04 1.222874e-03 3.897213e-
##           33           34           35           36           37           38           39
## 2.090101e-03 3.852095e-02 2.468260e-03 1.020990e-02 2.785237e-02 2.564552e-03 2.966163e-03 2.326532e-
##           41           42           43           44           45           46           47
## 9.580712e-05 9.752100e-05 8.039513e-05 1.204357e-03 8.263426e-04 1.353620e-02 5.113648e-05 1.644913e-
##           49           50           51           52           53           54           55
## 3.993715e-03 5.384996e-03 2.751168e-02 2.201388e-03 5.036966e-05 9.315808e-04 8.538716e-04 1.616119e-
##           57           58           59           60           61           62           63
## 7.417304e-04 4.932964e-03 1.052064e-02 2.068222e-04 5.618432e-04 5.062753e-04 2.081930e-03 7.245192e-
##           65           66           67           68           69           70           71
## 4.965408e-03 5.697068e-04 1.135329e-03 4.412855e-03 4.273307e-05 4.778742e-04 1.486835e-03 3.067627e-
##           73           74           75           76           77           78           79
## 1.825408e-03 9.836152e-03 1.513801e-03 1.360448e-03 3.478398e-04 3.934542e-03 6.235294e-06 2.342613e-
##           81           82           83           84           85           86           87
## 6.487672e-03 1.957221e-02 6.966171e-05 1.390727e-02 1.524533e-03 1.262938e-02 1.757770e-03 2.501968e-
##           89           90           91           92           93           94           95
## 1.290467e-04 1.484113e-03 4.630848e-03 1.084641e-06 6.579871e-03 2.044829e-04 4.221267e-03 4.807860e-
##           97           98           99          100          101          102          103
## 2.534480e-03 4.464089e-03 1.648750e-06 2.256339e-03 1.719062e-03 2.967252e-03 2.882102e-03 1.563354e-
##          105          106          107          108          109          110          111
## 1.075488e-03 5.100790e-03 4.549588e-03 9.438896e-03 9.523053e-04 4.998263e-03 5.224292e-03 4.330931e-
##          113          114          115          116          117          118          119
## 1.902742e-03 1.389404e-02 5.658202e-03 1.795796e-05 1.957608e-03 4.061660e-06 2.359802e-03 2.868832e-
##          121          122          123          124          125          126          127
## 3.191575e-04 1.106226e-02 1.880604e-03 2.189512e-03 2.537413e-03 8.497058e-04 5.164271e-04 5.188581e-
##          129          130          131          132          133          134          135
## 1.588980e-02 4.752664e-02 1.219183e-03 2.301796e-02 9.102609e-04 1.765870e-03 3.175230e-03 1.025267e-
##          137          138          139          140          141          142          143
## 1.611887e-02 1.633431e-04 1.084570e-02 2.023320e-04 3.243630e-03 3.856254e-03 7.644681e-03 1.239775e-
##          145          146          147          148          149          150          151
## 5.762644e-05 3.163203e-02 1.180606e-02 3.809058e-03 4.289603e-04 1.630474e-02 2.178355e-02 9.695261e-
##          153          154          155          156          157          158          159
## 1.649348e-05 1.642144e-02 1.667563e-02 2.001793e-03 1.814238e-03 1.740497e-03 2.325955e-04 3.031100e-
##          161          162          163          164          165          166          167
## 5.285385e-04 1.021180e-03 6.368696e-04 8.402137e-04 1.792640e-04 1.074512e-02 4.674516e-03 1.676631e-
##          169          170          171          172          173          174          175
## 4.644995e-03 7.346230e-04 1.718303e-04 1.022676e-03 2.867341e-03 4.223205e-03 1.650266e-03 7.412977e-
##          177          178          179          180          181          182          183
## 9.705622e-04 6.577370e-03 7.122881e-03 3.611486e-03 1.583265e-03 8.365122e-05 4.793880e-04 2.012250e-
##          185          186          187          188          189          190          191
## 2.564607e-03 2.196358e-03 4.248397e-04 1.128549e-03 7.917198e-03 3.083580e-03 4.341800e-04 2.253412e-
##          193          194          195          196          197          198          199
## 2.167849e-06 2.920000e-05 1.176317e-03 6.365920e-04 3.928924e-03 1.891097e-03 5.965869e-05 1.451272e-
##          201          202          203
## 3.733973e-04 7.225258e-05 7.452611e-03
```

```r
plot(ajuste,which=4)
```

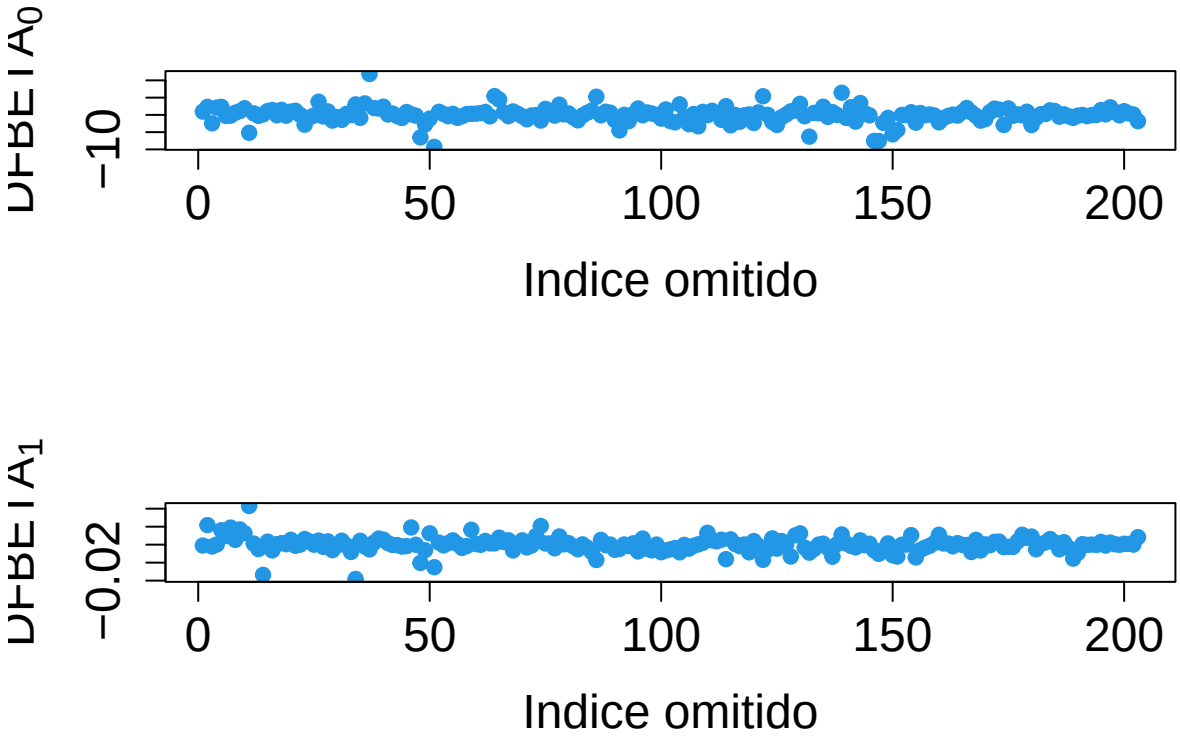


- DFFITs

```r
DFFITs <- dffits(ajuste)
```

- DFBETAs

```r
DFBETAS <- dfbeta(ajuste)
par(mfrow = c(2,1), pch=19, col=1, cex.lab = 1.5, cex.axis = 1.5)
plot(dfbeta(ajuste)[,1],xlab = "Indice omitido",
     ylab = expression(DFBETA[0]),col=4)
plot(dfbeta(ajuste)[,2], xlab = "Indice omitido", col=4,
     ylab = expression(DFBETA[1]))
```




```
par(mfrow = c(1,1))
```

11. Estimación media condicionada y predicción

Finalmente, obtengamos el intervalo de confianza y de predicción para el nivel de ozono medio al 95% de confianza con el modelo seleccionado para cada mes con todas las demás variables fijadas en su valor medio .

```
new.dat <- data.frame(T_Sandburg = mean(T_Sandburg), Humedad = mean(Humedad),
                      T_ElMonte = mean(T_ElMonte), Mes = c(1:12),
                      Pres_Alt = mean(Pres_Alt), Inv_Alt_b = mean(Inv_Alt_b))
predict(ajuste, newdata = new.dat, interval="confidence", level = 0.95)
```

```
##          fit       lwr      upr
## 1  13.447245 12.280941 14.61355
## 2  13.071803 12.055667 14.08794
## 3  12.696361 11.818831 13.57389
## 4  12.320918 11.564056 13.07778
## 5  11.945476 11.281488 12.60946
## 6  11.570034 10.958334 12.18173
## 7  11.194592 10.584075 11.80511
## 8  10.819149 10.158437 11.47986
## 9  10.443707  9.691638 11.19578
## 10 10.068265  9.196524 10.94001
## 11  9.692823  8.683113 10.70253
## 12  9.317380  8.157918 10.47684
```

```
predict(ajuste, newdata = new.dat, interval="prediction", level = 0.95)
```

```
##          fit       lwr      upr
## 1  13.447245 4.7574694 22.13702
## 2  13.071803 4.4009050 21.74270
## 3  12.696361 4.0406113 21.35211
```

```
## 4  12.320918 3.6765688 20.96527
## 5  11.945476 3.3087627 20.58219
## 6  11.570034 2.9371829 20.20288
## 7  11.194592 2.5618244 19.82736
## 8  10.819149 2.1826871 19.45561
## 9  10.443707 1.7997758 19.08764
## 10 10.068265 1.4131004 18.72343
## 11  9.692823 1.0226753 18.36297
## 12  9.317380 0.6285201 18.00624
```

```
rm(list = ls())
par(mfrow=c(1,1))
```

Regresión Logística

- Antes de empezar, cargamos los datos *Oro.rda*

```
load("Datos/Oro.rda")
Oro <- Oro
```

1. Análisis descriptivo

Para el análisis descriptivo de las variables podemos comenzar con una visión general de las variables mediante las funciones `str()` y `summary()`.

```
str(Oro)
```

```
## 'data.frame':    64 obs. of  4 variables:
##  $ As        : num  6.77 15.03 6.43 0.1 0.1 ...
##  $ Sb        : num  3.08 6.15 2.35 0.3 0.3 9.62 0.51 3.71 4.32 0.8 ...
##  $ Corredor  : int  1 1 1 0 0 1 0 1 0 0 ...
##  $ Proximidad: int  1 1 1 0 0 1 0 1 0 0 ...
```

La salida de `str()` nos dice que los datos constan de 64 observaciones de 4 variables:

- `As`: Nivel de concentración de arsénico en la muestra de agua. (numérica)
- `Sb`: Nivel de concentración de antimonio en la muestra de agua. (numérica)
- `Corredor`: Variable binaria indicando si la zona muestreada está (1) o no está (0) en alguno de los corredores delimitados por las lineas sobre el mapa. (categórica)
- `Proximidad` : Variable de respuesta que toma los valores 1 o 0 según que el depósito esté próximo o esté muy lejano al lugar.

```
attach(Oro)
```

```
## The following objects are masked from Oro (pos = 4):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 6):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 8):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 10):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 12):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 14):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 16):
##
##     As, Corredor, Proximidad, Sb

## The following objects are masked from Oro (pos = 17):
##
```

```
##     As, Corredor, Proximidad, Sb
## The following objects are masked from Oro (pos = 21):
##
##     As, Corredor, Proximidad, Sb
## The following objects are masked from Oro (pos = 23):
##
##     As, Corredor, Proximidad, Sb
## The following objects are masked from Oro (pos = 24):
##
##     As, Corredor, Proximidad, Sb
## The following objects are masked from Oro (pos = 29):
##
##     As, Corredor, Proximidad, Sb
```

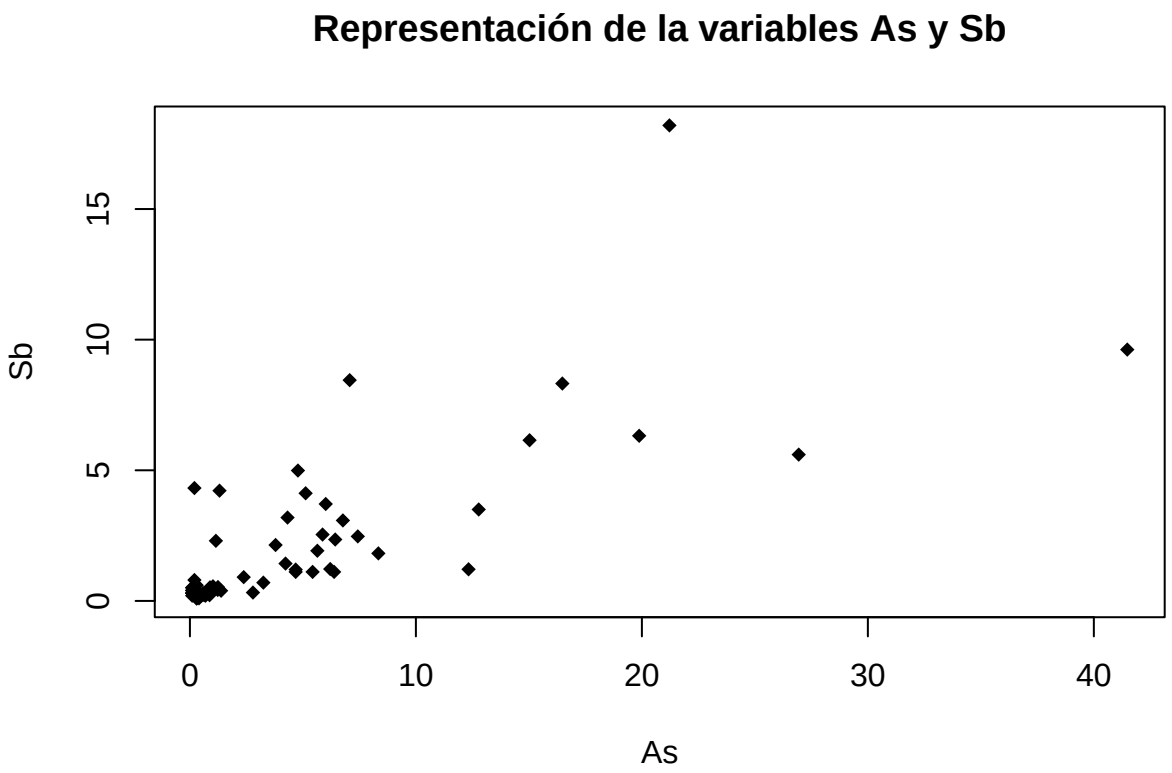
```
Oro$Corredor <- as.factor(Oro$Corredor) # Convertimos la variable Corredor a factor
numericas.oro <- Oro[1:2]               # Almacenamos las variables numéricas
respuesta.oro <- Proximidad             # Almacenamos la variable de respuesta
```

Con la salida de `summary()` y graficando `As` frente a `Sb` podemos ver que, basándonos en la diferencia entre las medias y las medianas, las variables numéricas se concentran en valores bajos, aunque deben de existir registros con valores relativamente altos:

```
summary(Oro)
```

```
##        As               Sb         Corredor   Proximidad
##  Min.   : 0.100   Min.   : 0.100   0:32     Min.   :0.0000
##  1st Qu.: 0.400   1st Qu.: 0.300   1:32     1st Qu.:0.0000
##  Median : 1.235   Median : 0.650            Median :0.0000
##  Mean   : 4.645   Mean   : 2.039            Mean   :0.4375
##  3rd Qu.: 5.905   3rd Qu.: 2.487            3rd Qu.:1.0000
##  Max.   :41.480   Max.   :18.200            Max.   :1.0000
```

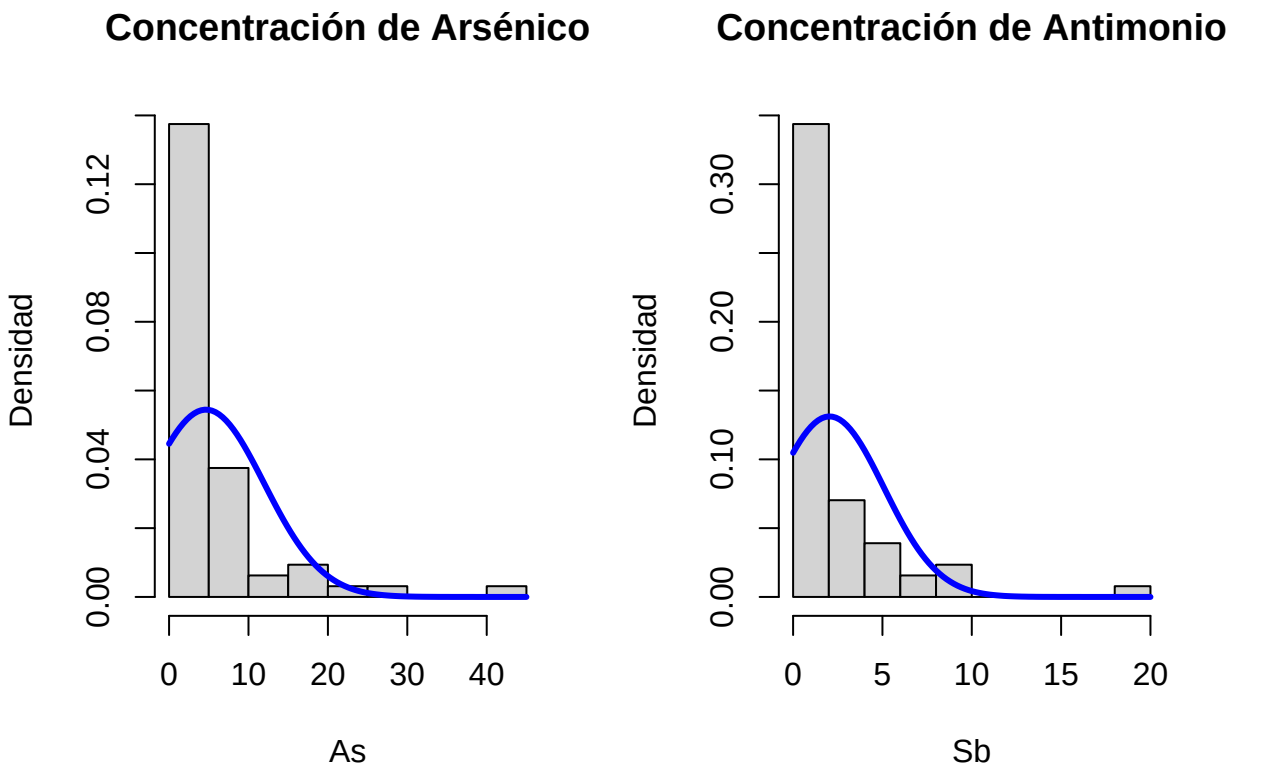
```r
plot(numericas.oro, pch=18,
     main="Representación de la variables As y Sb")
```



Este hecho se confirma también al mirar los histogramas y diagramas de cajas:

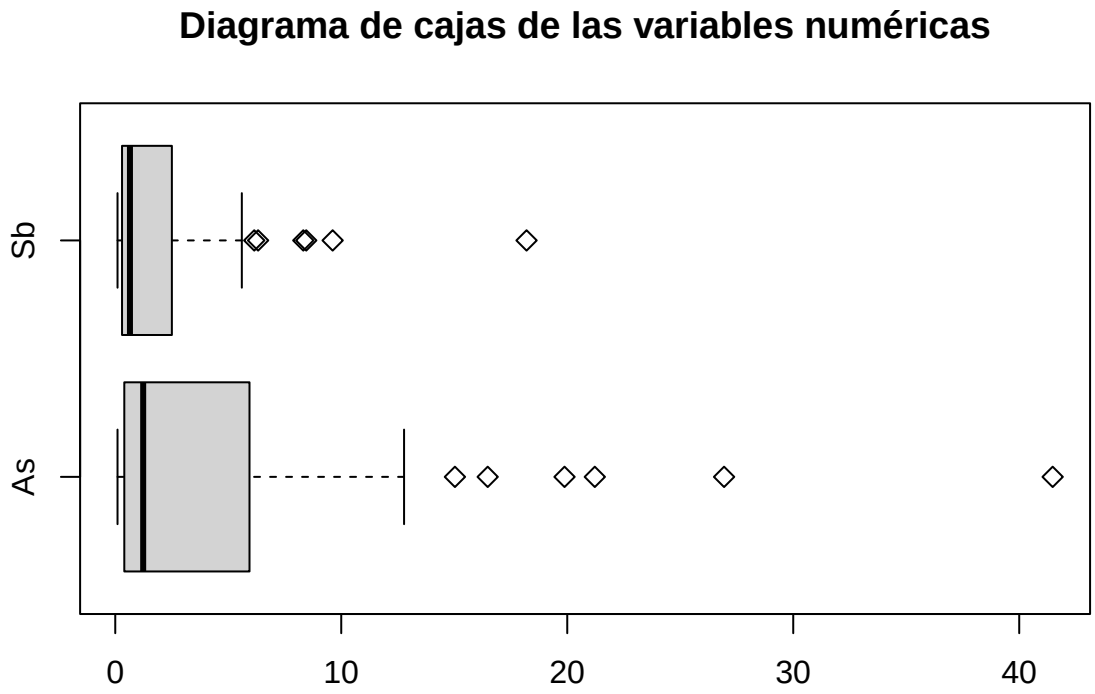
```r
old.par <- par(mfrow=c(1,2))
hist(As, freq=F, xlab="As", ylab = "Densidad",
     main="Concentración de Arsénico")
curve(dnorm(x,mean=mean(As), sd=sd(As)),
      col="blue", lwd=3, add=TRUE)

hist(Sb, freq=F, xlab="Sb", ylab = "Densidad",
     main="Concentración de Antimonio")
curve(dnorm(x,mean=mean(Sb), sd=sd(Sb)),
      col="blue", lwd=3, add=TRUE)
```




```
par(old.par)

boxplot(numericas.oro, horizontal=T, pch=5,
        main="Diagrama de cajas de las variables numéricas")
```



Distribución de la variable `Proximidad`:

```
table(Proximidad); table(Proximidad)/nrow(Oro)
```

```
## Proximidad
##  0  1
## 36 28
```

```
## Proximidad
##      0      1
## 0.5625 0.4375
```

Distribución de la variable `Corredor`:

```
table(Corredor)
```

```
## Corredor
##  0  1
## 32 32
```

Observamos que si los datos se encuentran en alguno de los corredores, suelen estar próximos a un depósito de oro y lejanos si no es así:

```
xtabs(~Proximidad + Corredor, data=Oro)
```

```
##           Corredor
## Proximidad  0  1
##          0 30  6
##          1  2 26
```

2. Modelo matemático

Dado que contamos con una muestra de n realizaciones $(\vec{X}^t, Y)$ con $\vec{X}^t = (X_1, \ldots, X_k)$ que asumimos independientes, y que la variable respuesta, `Proximidad`, es binaria (0 o 1), debemos de elegir un modelo que tenga esto en cuenta. En nuestro caso hemos elegido una transformación del modelo lineal, definida por la distribución logística de la ecuación 2.

$$F(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \tag{2}$$

Por tanto, nuestro modelo logístico quedaría de la forma

$$Y|(\vec{X} = \vec{X}_i) \sim Be(p_i), \quad p_i = \mathbb{P}(Y = 1|\vec{X}_i) = \frac{1}{1+e^{-\eta}} \tag{3}$$

Tal que

$$\eta = \beta_0 + \beta_1 As + \beta_2 Sb + \tau I(Corredor = 1) \tag{4}$$

siendo $I(Corredor = 1)$ la variable indicadora para cuando Corredor toma el valor 1. Además,

$$1 - p_i = \mathbb{P}(Y = 0|\vec{X}_i) = 1 - \frac{1}{1+e^{-\eta}} = \frac{e^{-\eta}}{1+e^{-\eta}} \tag{5}$$

3. Interpretación del modelo

Para una mejor interpretación del modelo, podemos definir el **odds**$_i$ de manera que

$$odds_i = odds(Y|\vec{X}_i) = \frac{p_i}{1-p_i} = e^{\eta} = e^{\vec{\beta}^t \vec{X}_i} = e^{\beta_0} e^{\beta_1 X_{i1}} \cdots e^{\beta_k X_{ik}} \quad , \quad 1 \leq i \leq n \tag{6}$$

Este es un modelo multiplicativo, en el cual e^{β_0} es la respuesta cuando $\vec{X}_i = \vec{0}$, mientras que e^{β_j}, para $1 \leq j \leq k$, es el incremento multiplicativo $(e^{\beta_j})^l$ en el odds para algún incremento l en X_j

Si resulta que existe una variable binaria podemos utilizar el **odds-ratio**, que indica en qué medida el suceso $Y = 1$ es más posible que $Y = 0$ si $X = 1$ que si $X = 0$:

$$OR = \frac{\mathbb{P}(Y=1|X=1)/\mathbb{P}(Y=0|X=1)}{\mathbb{P}(Y=1|X=0)/\mathbb{P}(Y=0|X=0)} = \frac{e^{\beta_0+\beta_1}}{e^{\beta_0}} \tag{7}$$

Si X es cualitativa podemos seguir aplicando el OR con $g-1$ variables *dummy*, siendo g el número de categorías.

También podemos expresar el modelo aplicando logaritmos a la ecuación 6, de manera que

$$\ln(\frac{p_i}{1-p_i}) = \eta = \vec{\beta}^t \vec{X}_i \tag{8}$$

Los cuales denominaremos como **logit**$_i$. Estos logits son interpretables mucho más fácilmente ya que son interpretables linealmente.

Finalmente, por lo comentado en el apartado del modelo matemático y en este, este modelo sigue las tres siguientes hipótesis estructurales:

1. Linealidad de los logits.
2. Respuesta binaria de la Y.
3. Independencia de las observaciones.

4. Análisis de multicolinealidad

Debemos analizar si estamos ante un caso de multicolinealidad. Si así fuera, las estimaciones de los parámetros no serían correctos, y nuestro modelo solo serviría para predecir, no para explicar el comportamiento de la respuesta.

Utilizaremos los factores de inflacción de la varianza generalizada, para ver si nos encontramos con variables correlacionadas:

```
ajuste_completo <- glm(Proximidad~., data = Oro, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
library(car)
vif(ajuste_completo)
```

```
##       As       Sb Corredor
## 1.577307 2.293729 1.872848
```

Los factores de inflacción de la varianza son todos menores que 10, por lo que no estamos ante un caso de multicolinealidad.

5. Selección del modelo

A pesar de no tener multicolinealidad en los datos, decidimos hacer una selección de variables, debido a la no significación de todas las variables.

Para ello, decidimos utilizar un método de selección exhaustiva con el BIC, ya que esta medida de selección de modelos 'castiga' a modelos con un número elevado de variables:

```
library(bestglm)
M1.exh.AIC <- bestglm(Oro, IC = "BIC", family = binomial,
                      method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
M1.exh.AIC$Subsets
```

```
##    Intercept    As    Sb Corredor logLikelihood      BIC
## 0       TRUE FALSE FALSE    FALSE    -43.860109 87.72022
## 1       TRUE  TRUE FALSE    FALSE    -11.301429 26.76174
## 2*      TRUE  TRUE  TRUE    FALSE     -9.152897 26.62356
## 3       TRUE  TRUE  TRUE     TRUE     -7.097155 26.67096
```

```
# La fila con el asterisco indica el modelo seleccionado.
# Aquí el modelo es el modelo sin corredor.
# Esto también nos lo indicaba el p-valor inicial.
```

Por lo tanto, definimos el ajuste sin corredor y vemos la significación del resto de las variables:

```
ajuste_sin_corredor <- update(ajuste_completo,.~.-Corredor)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(ajuste_sin_corredor)
```

```
##
## Call:
## glm(formula = Proximidad ~ As + Sb, family = "binomial", data = Oro)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.02141  -0.19496  -0.14513   0.06255   2.60217
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.9664     1.3675  -3.632 0.000281 ***
## As            1.2490     0.3777   3.307 0.000943 ***
## Sb            0.9235     0.4486   2.059 0.039518 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 87.720  on 63  degrees of freedom
## Residual deviance: 18.306  on 61  degrees of freedom
```

```
## AIC: 24.306
##
## Number of Fisher Scoring iterations: 8
```

6. Posible Interacción

Debido a la posible necesidad de interacción, decidimos probar si un modelo que incluya interacción es mejor que nuestro modelo completo.

Comenzamos definiendo este modelo, con todas las interacciones posibles:

```
ajuste.i <- update(ajuste_completo, .~.^3, family=binomial, data=Oro)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(ajuste.i)
```

```
##
## Call:
## glm(formula = Proximidad ~ As + Sb + Corredor + As:Sb + As:Corredor +
##     Sb:Corredor + As:Sb:Corredor, family = binomial, data = Oro)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9714   0.0000   0.0000   0.0000   1.9345
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -8.939  34483.934   0.000    1.000
## As                 -47.382 105299.858   0.000    1.000
## Sb                 -33.817 196896.288   0.000    1.000
## Corredor1            9.617  34483.934   0.000    1.000
## As:Sb               47.999  60183.576   0.001    0.999
## As:Corredor1        46.489 105299.858   0.000    1.000
## Sb:Corredor1        26.827 196896.289   0.000    1.000
## As:Sb:Corredor1    -44.627  60183.576  -0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 87.7202  on 63  degrees of freedom
## Residual deviance:  7.5068  on 56  degrees of freedom
## AIC: 23.507
##
## Number of Fisher Scoring iterations: 21
```

Ningún coeficiente es significativo, por lo que consideramos que esto se puede deber a la presencia de multicolinealidad debido a las interacciones.

Decidimos hacer una selección de variables, por si alguna interacción entre variables originales resultase significativa. La haremos igual que en el apartado anterior:

```
M0 <- update(ajuste_completo, Proximidad~1)
step(M0, direction="forward", trace=1,
     scope = list(lower=M0,upper=ajuste.i))
```

```
## Start:  AIC=89.72
## Proximidad ~ 1
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##            Df Deviance    AIC
## + As        1   22.603 26.603
## + Sb        1   45.332 49.332
## + Corredor  1   45.848 49.848
## <none>          87.720 89.720
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Step:  AIC=26.6
## Proximidad ~ As
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##            Df Deviance    AIC
## + Sb        1   18.306 24.306
## + Corredor  1   19.990 25.990
## <none>          22.603 26.603
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Step:  AIC=24.31
## Proximidad ~ As + Sb
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##            Df Deviance    AIC
## + Corredor  1   14.194 22.194
## <none>          18.306 24.306
## + As:Sb     1   17.249 25.249
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Step:  AIC=22.19
## Proximidad ~ As + Sb + Corredor
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##               Df Deviance    AIC
## <none>             14.194 22.194
## + Sb:Corredor  1   12.253 22.253
## + As:Sb        1   12.688 22.688
## + As:Corredor  1   14.137 24.137
```

```
##
## Call:  glm(formula = Proximidad ~ As + Sb + Corredor, family = "binomial",
##     data = Oro)
##
## Coefficients:
## (Intercept)           As           Sb    Corredor1
```

```
##      -7.610        1.205        1.421        3.197
##
## Degrees of Freedom: 63 Total (i.e. Null);  60 Residual
## Null Deviance:       87.72
## Residual Deviance: 14.19     AIC: 22.19
```

Finalmente, vemos que en este caso, la interacción de las variables no aporta nada a nuestro ajuste.

```
ajuste <- ajuste_sin_corredor
```

7. Inferencia

Empezamos la inferencia haciendo los intervalos de confianza para los parámetros. Haremos los intervalos basados en las sd de las pruebas de Wald y en los cuantiles de una normal:

```
confint.default(ajuste)
```

```
##                    2.5 %    97.5 %
## (Intercept) -7.64658528 -2.286183
## As           0.50875493  1.989343
## Sb           0.04431076  1.802633
```

Teniendo en cuenta la ecuación 8, los coeficientes ajustados y las variables significativas, el modelo quedaría como en la ecuación 9

$$ln(\frac{\hat{p}}{1-\hat{p}}) = \hat{\eta} = -4.9664 + 1.2490As + 0.9235Sb \tag{9}$$

Para la interpretación de los coeficientes del modelo ajustado utilizaremos los odds, que calcularemos a partir de los valores que devuelve el `summary()` del ajuste:

```
( estimates <- summary(ajuste)$coef )
```

```
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -4.9663844  1.3674746 -3.631793 0.0002814590
## As           1.2490491  0.3777080  3.306917 0.0009432888
## Sb           0.9234717  0.4485597  2.058749 0.0395183372
```

```
logits_ajuste <- estimates[1:3]
```

Aquí podemos observar que el odds de las respuestas cuando $\beta_j = 0$, para $j \in \{1, \ldots, k\}$, tiene un valor de $e^{-4.9663844} = 0.006968297$, que indica que el cociente $\frac{p}{1-p}$ tiene una probabilidad de 0.00697:1 de estar próximo a un yacimiento de oro cuando la concentración de Arsénico y antimonio es nula.

En cuanto a los coeficientes de las variables `As` y `Sb`, indican un incremento multiplicativo del odds de $e^{1.2490491} = 3.487025$ y $e^{0.9234717} = 2.518017$ respectivamente, cuando el resto de las variables se mantiene constante.

O lo que es lo mismo, el valor de los logits cuando $\beta_j = 0$, para $j \in \{1, \ldots, k\}$ es de -4.9663844, que un incremento de una unidad en `As` representa un cambio en los logits de 1.2490491 y que un incremento de una unidad en `Sb` representa un cambio en los logits de 0.9234717 (manteniendo) el resto de las variables constantes.

8. Estimación media y probabilidad condicionada

Haremos los intervalos de confianza y de probabilidad manteniendo las dos variables en su media:

```
new <- with(Oro, data.frame(As = mean(As), Sb = mean(Sb)))
```

Utilizamos predict para la predicción estimada:

```r
p_est_proximidad <- predict(ajuste, newdata = new,
                            type = "response")
cbind(new,p_est_proximidad)
```

```
##         As       Sb p_est_proximidad
## 1 4.644844 2.039062        0.9380961
```

Para obtener los intervalos de confianza para estas predicciones, utilizaremos la siguiente función proporcionada en el Script de R Logísitica:

```r
est.media.cond.CI <- function(ajuste, newdata, level = 0.95){
  # Predicciones de los logit
  pred <- predict(object = ajuste, newdata = newdata, se.fit = TRUE)
  # CI para los logits
  za <- qnorm(p = (1 - level) / 2)
  lwr <- pred$fit + za * pred$se.fit
  upr <- pred$fit - za * pred$se.fit
  # Back-transformada a probabilidades
  fit <- 1 / (1 + exp(-pred$fit))
  lwr <- 1 / (1 + exp(-lwr))
  upr <- 1 / (1 + exp(-upr))
  # Acomodamos en una matriz la salida
  result <- cbind(fit, lwr, upr)
  colnames(result) <- c("p", "LI", "LS")
  return(result)
}
```

La aplicamos del siguiente modo:

```r
est.media.cond.CI(ajuste, newdata = new)
```

```
##           p        LI        LS
## 1 0.9380961 0.6190406 0.9929738
```

9. Bondad del ajuste

10. Análisis de residuos

El modelo de regresión logísitica tiene 3 hipótesis estructurales: 1) La linealidad de los Logits. 2) La independencia de las n observaciones. 3) La respuesta Y debe ser binaria.

Tal y como sucede en regresión lineal, podemos utilizar los residuos para chequear las hipótesis estructurales. No obstante, debemos tener en cuenta que en regresión logística existen dos tipos de residuos, con fines distintos.

Obtención residuos de Pearson:

```r
res.p <- residuals(ajuste, type="pearson")
```

Obtención residuos de la Deviance:

```r
res.d <- residuals(ajuste, type="deviance")
```

Los estandarizamos: Residuos Pearson estandarizados:

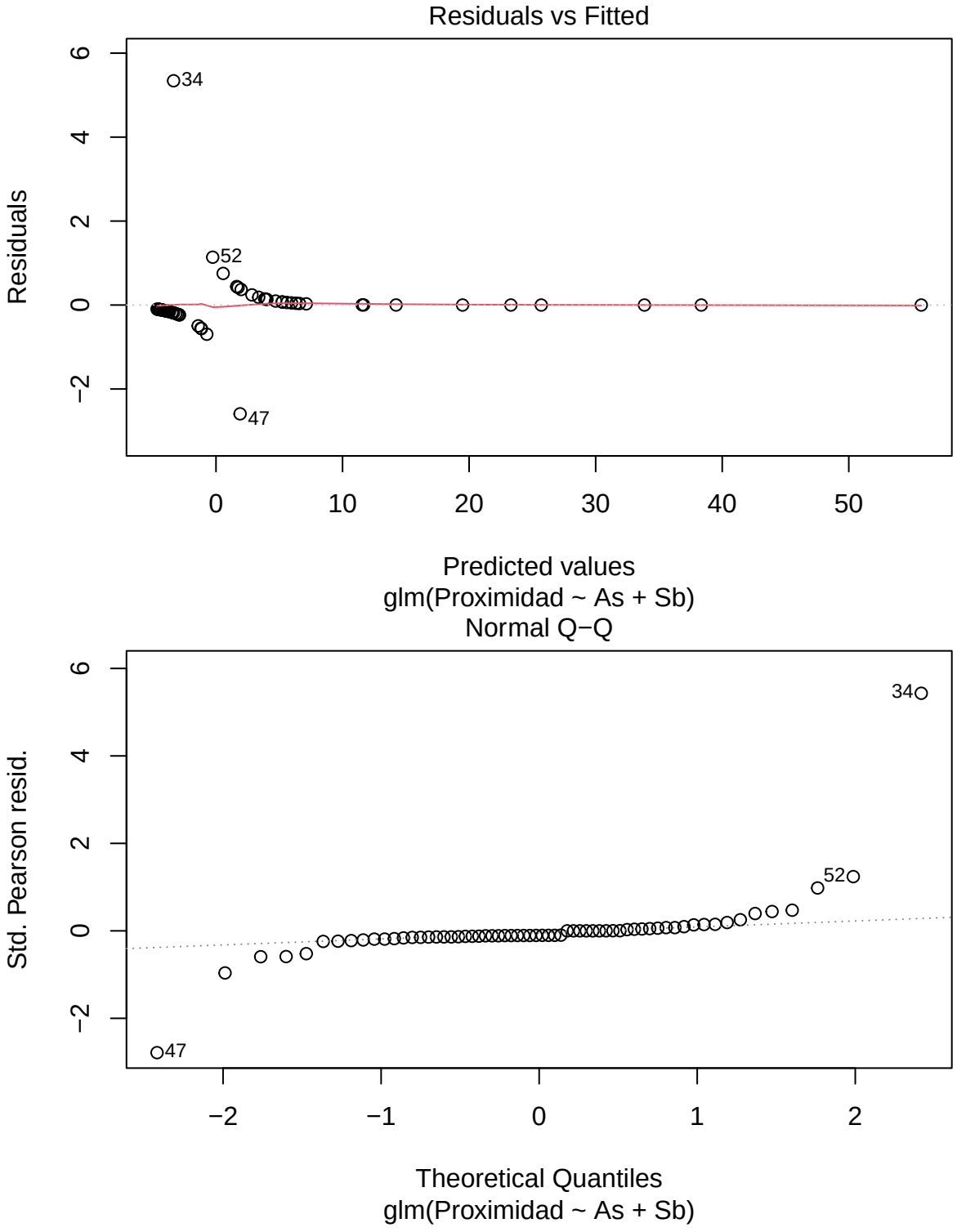
```r
res.p.e <- res.p/sqrt(1 - hatvalues(ajuste))
```

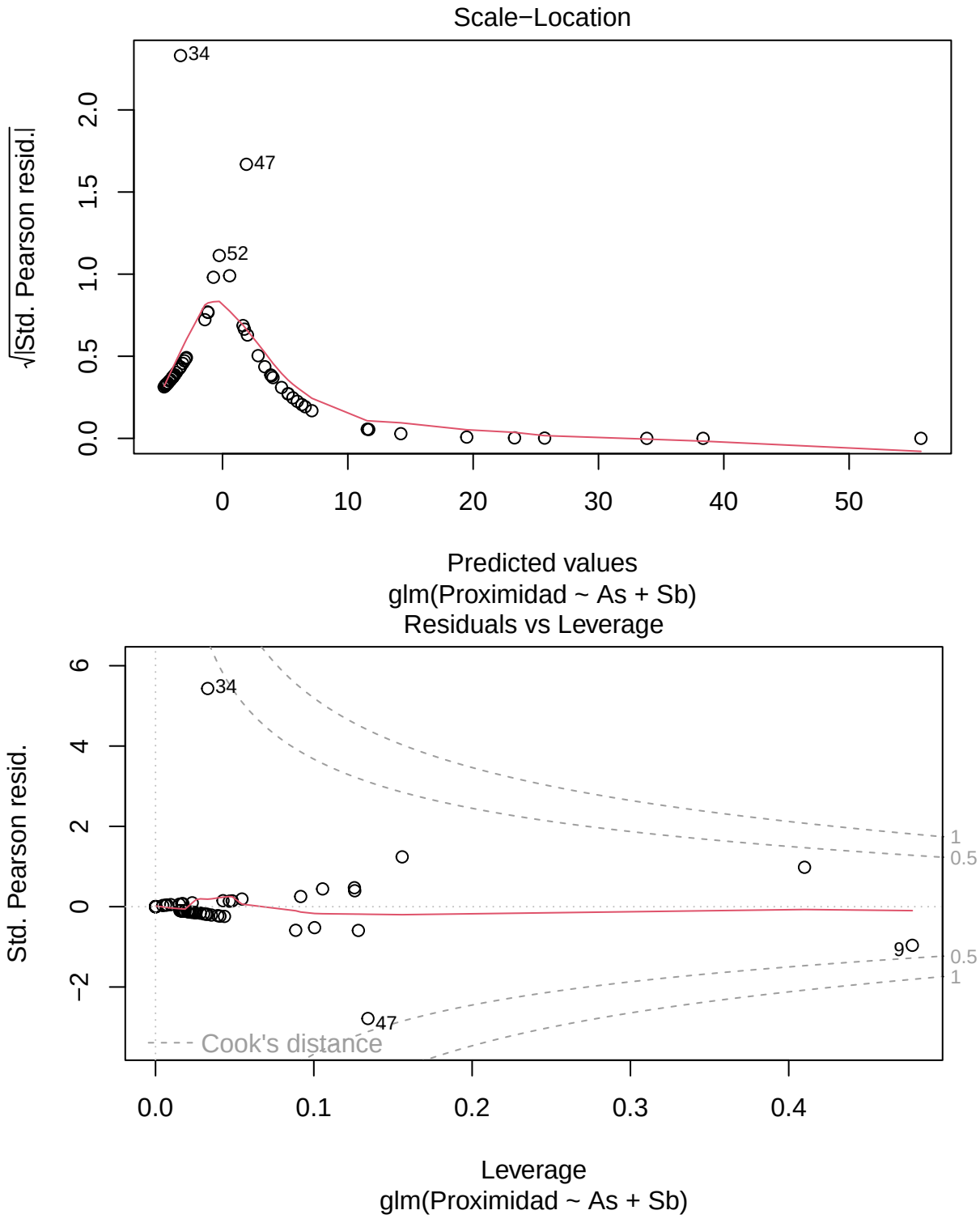
Residuos deviance estandarizados:

```
res.d.e <- res.d/sqrt(1 - hatvalues(ajuste))
```

Obtenemos los gráficos de residuos:

```
plot(ajuste)
```



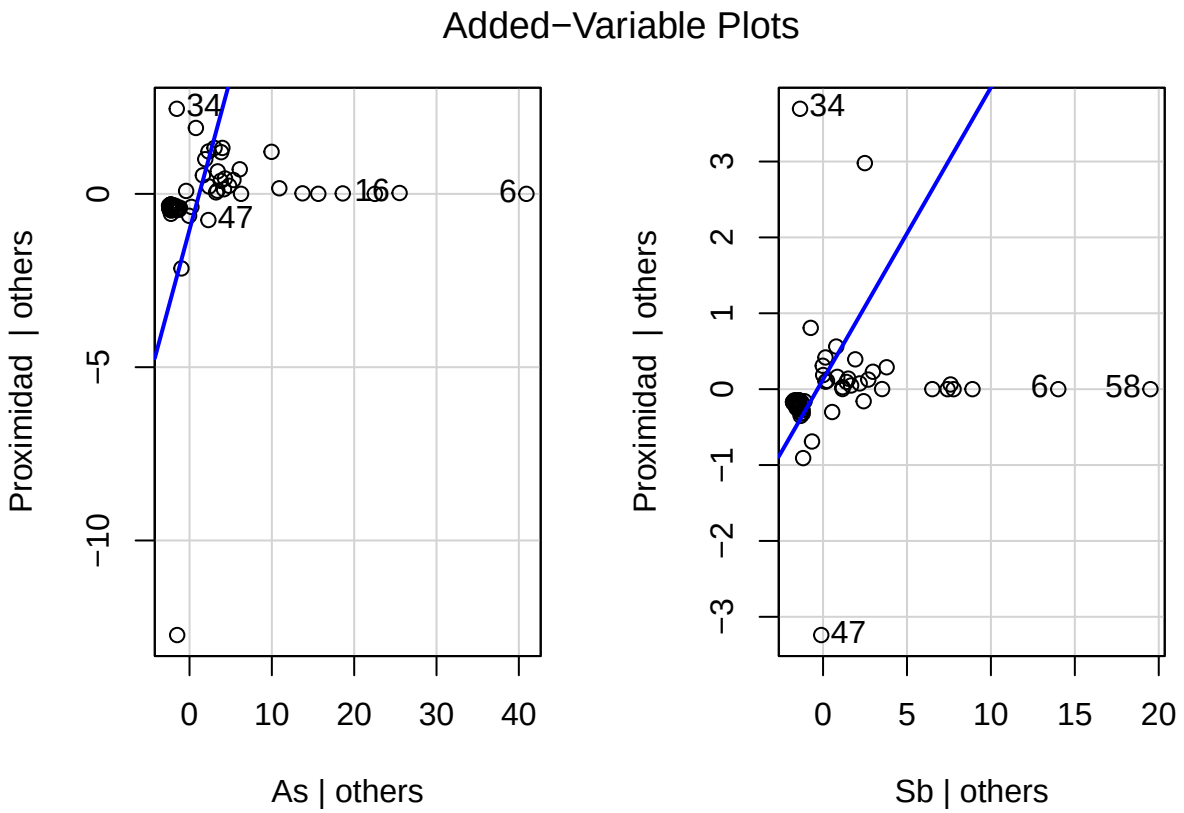


La función plot de R enfrenta los residuos estandarizados de Pearson con los logits del ajuste. Este tipo de residuo es útil simplemente para chequear la normalidad que, en este caso, evidentemente no está presente, como se aprecia en el segundo gráfico de la salida.

Para chequear la linealidad, se utilizan los residuos del segundo tipo, es decir, los de la deviance, del siguiente modo:

```
car::avPlots(ajuste,terms=~.)
```

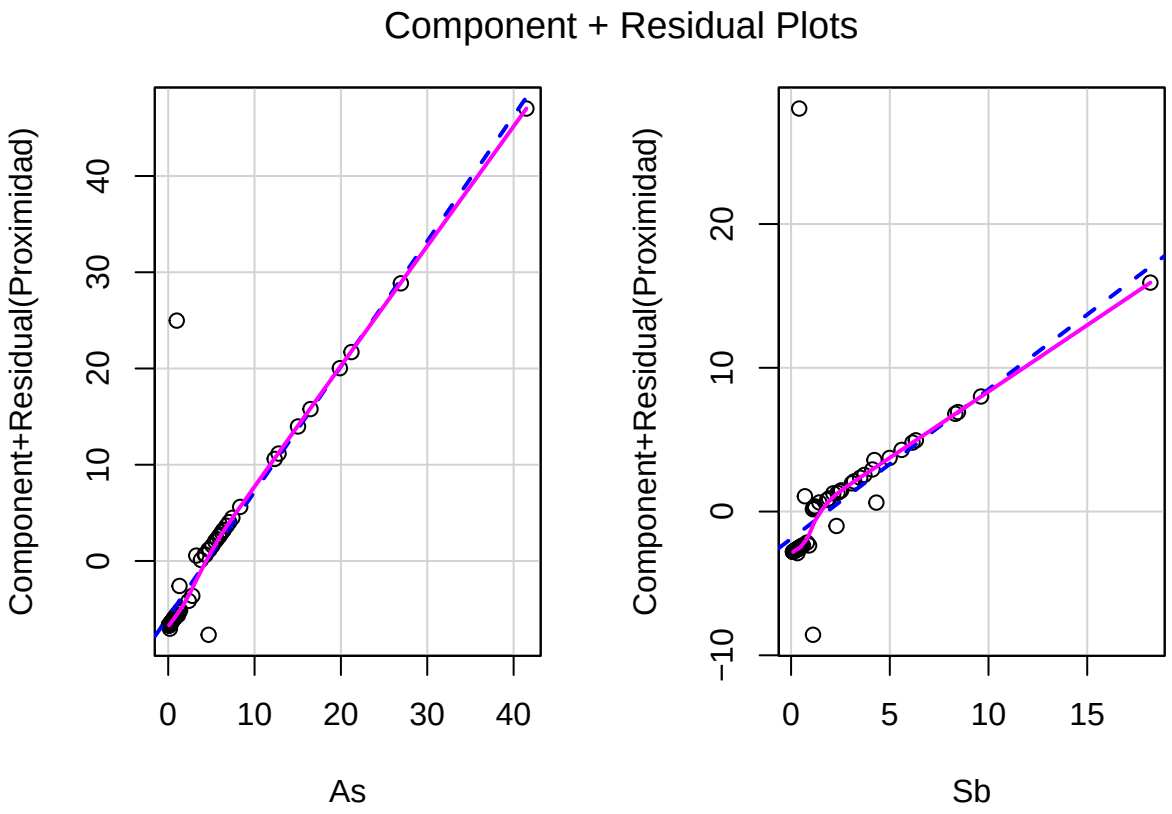
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



Podemos ver que prácticamente los datos están en torno a 0. El problema con las rectas es que, debido a la presencia de muchos datos entre 0 y 10, su pendiente varía mucho. ACABAR INTERPRETACIÓN

También podemos hacer gráficos de residuos parciales, para ver si la falta de linealidad es achacable a alguna variable concreta:

```
library(car)
crPlots(ajuste)
```

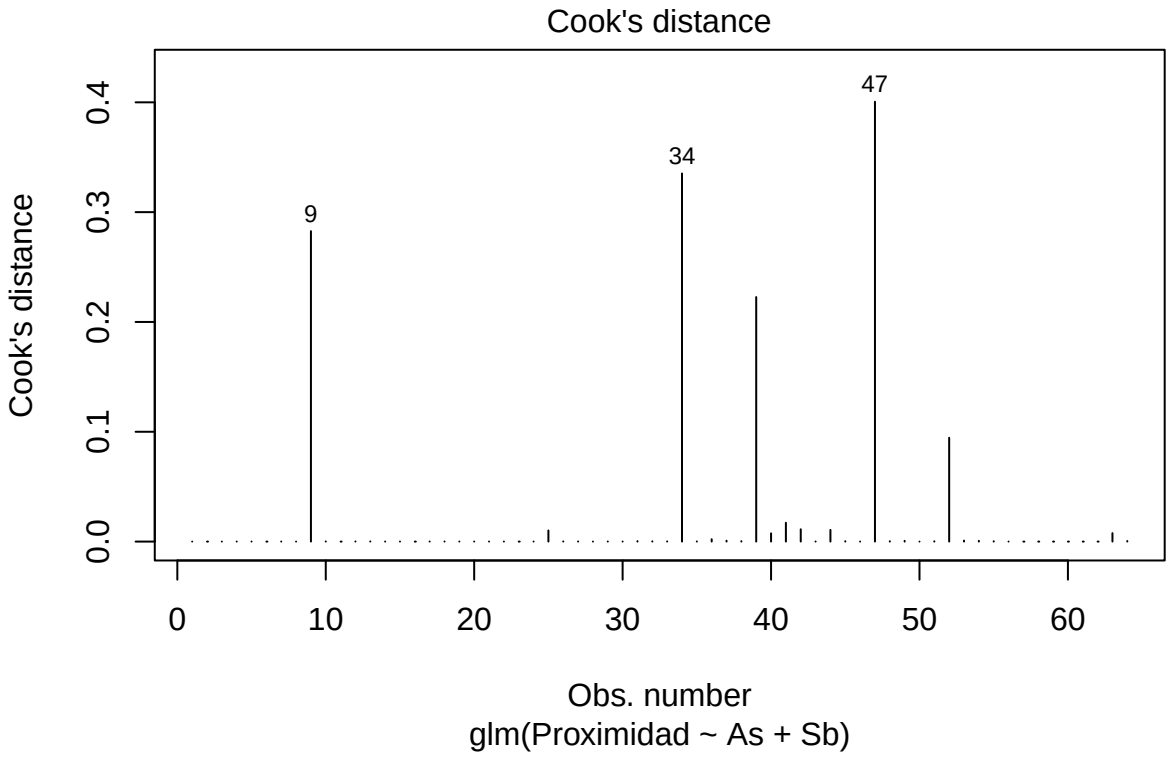


11. Análisis de influencia

Finalmente, los residuos de Pearson también se pueden utilizar para el análisis de influencia.

Para ver el gráfico de la distancia de Cook, se ejecuta el siguiente comando:

```
plot(ajuste, which = 4)
```



Vemos 3 observaciones con una distancia de Cook mayor que el resto de observaciones: {34, 39, 47}

Tal y como haciamos en regresión lineal múltiple, podemos utilizar la siguiente función de R para obtener las medidas del análisis de influencia automáticamente:

```
im <- influence.measures(ajuste)
summary(im)
```

```
## Potentially influential observations of
##   glm(formula = Proximidad ~ As + Sb, family = "binomial", data = Oro) :
##
##    dfb.1_  dfb.As  dfb.Sb  dffit   cov.r   cook.d hat
## 9  -0.13    0.85   -1.64_* -2.22_*  1.55_*  0.28   0.48_*
## 34  1.10_* -0.66   -0.69    1.13_*  0.26_*  0.34   0.03
## 39 -0.28   -0.23    1.63_*  1.95_*  1.37_*  0.22   0.41_*
## 47  0.38   -1.43_*  0.08   -1.80_*  0.50_*  0.40   0.13
## 52  0.40    0.42   -0.48    1.16_*  0.88    0.09   0.16_*
```

Nos centramos en las columnas “cook.d”, “hat” y “dffit”:

Con respecto a los leverages de Pregibon, vemos que las observaciones {9, 39, 52} parecen influyentes.

Con respecto a la distancia de Cook, vemos que las observaciones