Comprobación de supuestos para estadística multivariada en R (R Studio)

Felipe Ruiz.
Carolina García
Ayudantía Estadística IV 2015.

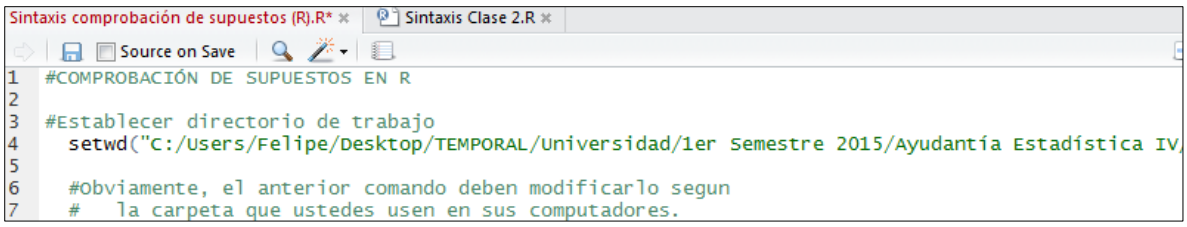
Aspectos previos:

Trabajaremos con el archivo "Ejemplo Supuestos" en formato CSV, disponible para descargar desde U-Cursos. Es una base de datos que mediante el comando "Guardar como" en SPSS, hemos guardado en formato CSV.

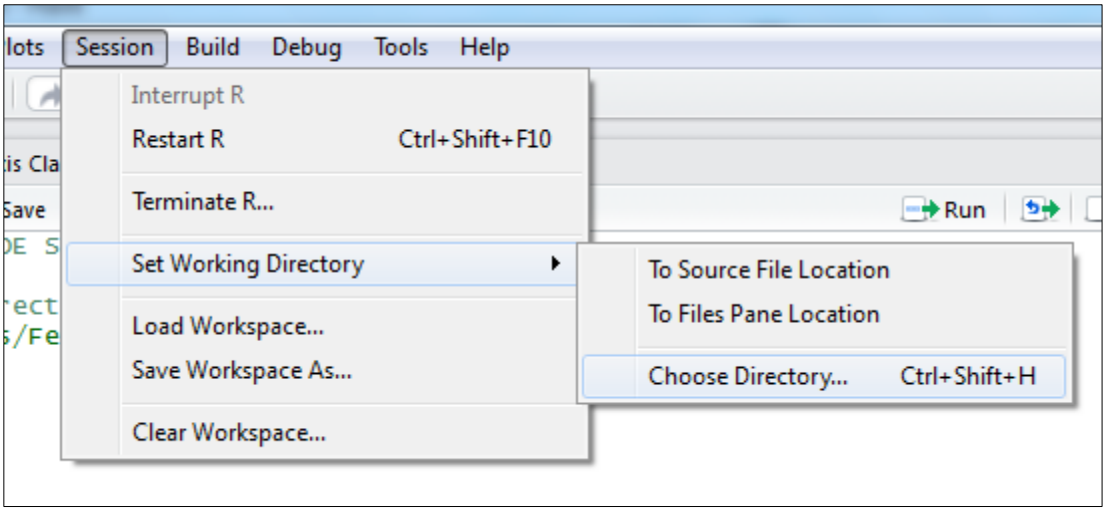
A	B	C
SAT	CLAB	CONFLIC
3	5	5
3	5	5
3,4	5	2,2
1	4,95	5
1	4,9	5
1,1	4,9	4,8
1,1	4,9	4,9
1,2	4,9	4,9
1,3	4,85	4,8
1,4	4,75	4,75
1,5	4,69	4,75
1,5	4,69	4,69
1,9	4,63	4,6
1,9	4,63	4,59
1,9	4,57	4,57
2	4,57	4,54

No olvidar establecer el directorio de trabajo en R (carpeta donde estarán los archivos que invocaremos en la sesión):

Esto se puede realizar mediante el comando *setwd* y especificando la ruta de acceso a la carpeta.



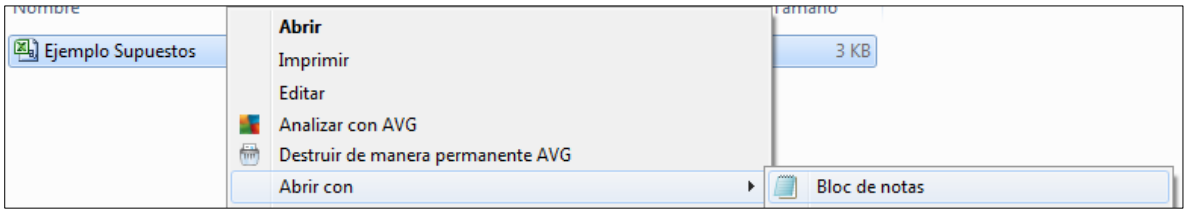
También se puede realizar mediante la interfaz de botones de R Studio; una vez abierto el cuadro de diálogo seleccionan la carpeta donde trabajarán:

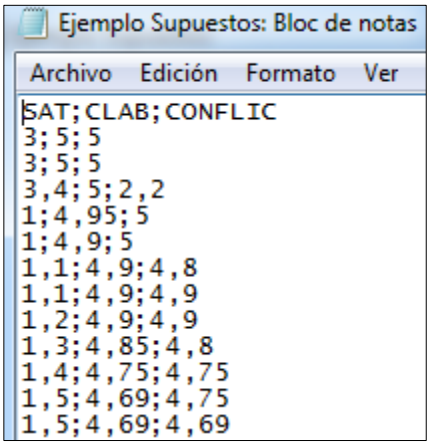


Mediante el siguiente comando se le pide al software que cree un objeto ("base") empleando el comando "read.csv" que permite que el programa lea una base de datos en formato .csv. Además se le indica que el separador de casos es un ";" y que el signo que indica la presencia de decimales es ",". Ojo que deben fijarse siempre que el nombre que ingresen sea exacto al nombre del archivo, algunos exploradores de internet le ponen guiones bajos a los espacios cuando bajan algún archivo desde internet y eso hace que el nombre (la ruta del archivo) cambie.

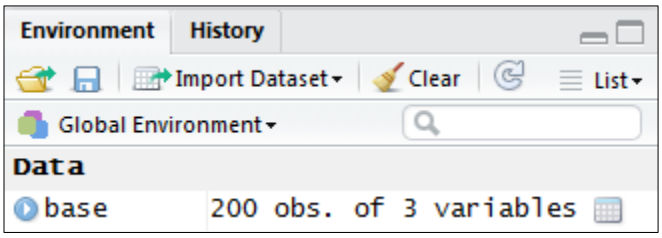
```
5
6  #Crear el objeto "base" a partir de una archivo existente
7  base<-read.csv(file="Ejemplo Supuestos.csv",sep=";",dec=",")
```

Esto puede corroborarse visualmente abriendo el archivo con el bloc de notas. En algunas bases, los casos vienen separados por comas, y los decimales indicados por puntos: si no establecen bien tales especificaciones, nunca lograrán abrir las bases.





Una vez que tenemos creado nuestro objeto en R, este queda disponible en la ventana de entorno ("Environment") de R Studio.



Debido a que tuvimos un inconveniente con esta base de datos para aplicar un comando, crearemos también otra base de datos a partir del archivo *EjemploAberrantes*. Seguimos el mismo procedimiento, y creamos un objeto con el nombre *baseA*, que se sumará a nuestro entorno.

```
#Emplearemos otra base de datos para los datos aberrantes multivariantes.

  baseA<-read.csv(file="EjemploAberrantes.csv",sep=";",dec=",")
```

Por el momento seguiremos trabajando con el objeto *base*. Ahora podemos observar sus características básicas mediante los siguientes comandos.

```
 9  #Observar el objeto. Inspección básica de las variables
10
11  summary(base)       #Estadísticos descriptivos de las variables
12
13  str(base)           #Resumen de la estructura de las variables
14
15  dim(base)           #Dimensión de la base:número de casos y variables.
```

El comando *summary* nos proporciona estadísticos descriptivos de las variables presentes en la base de datos:

```
      SAT              CLAB            CONFLIC
 Min.   :1.00   Min.   :1.000   Min.   :1.000
 1st Qu.:3.00   1st Qu.:2.200   1st Qu.:2.200
 Median :3.10   Median :2.900   Median :2.900
 Mean   :3.12   Mean   :2.922   Mean   :2.948
 3rd Qu.:3.50   3rd Qu.:3.650   3rd Qu.:3.800
 Max.   :5.00   Max.   :5.000   Max.   :5.000
```

El comando *str* nos indica la estructura y características de la base de datos y las variables:

```
> str(base)
'data.frame': 200 obs. of  3 variables:
 $ SAT    : num  3 3 3.4 1 1 1.1 1.1 1.2 1.3 1.4 ...
 $ CLAB   : num  5 5 5 4.95 4.9 4.9 4.9 4.9 4.85 4.75 ...
 $ CONFLIC: num  5 5 2.2 5 5 4.8 4.9 4.9 4.8 4.75 ...
```

Tamaño muestral:

El comando *dim* nos indica la dimensión de la base de datos (casos, variables):

```
> dim(base)
[1] 200   3
```

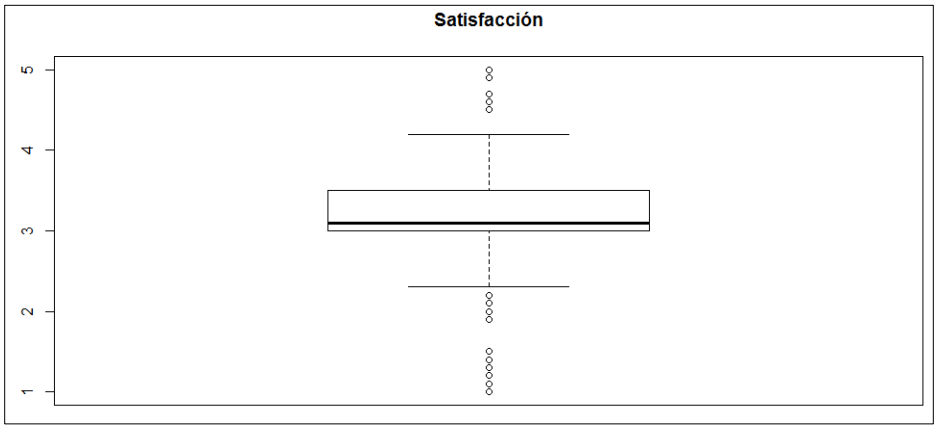
El resultado de estos comandos nos indica que tenemos 200 observaciones (n = 200), tres variables, (Satisfacción, Carga Laboral y Conflictividad) todas numéricas. A la vez, se ve que dado que son parecidas la mediana y la media en todas las variables pareciera ser que son simétricas.

Observación gráfica de las variables (tipo de distribución, casos atípicos)

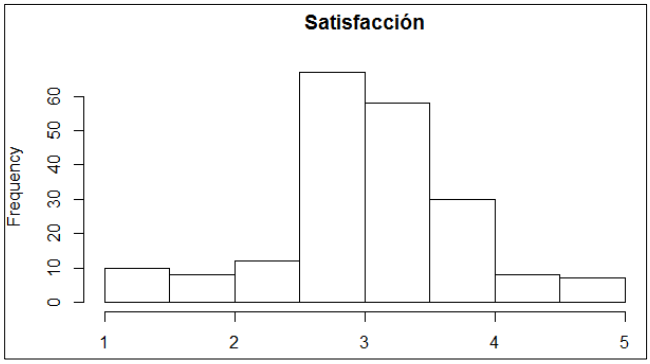
Ambos comandos tienen la misma estructura (X $ Y, main = "Z"). X es el nombre del objeto base de datos, Y es la variable y z es el título para el gráfico.

```
17  #Inspección gráfica de las variables
18
19  boxplot(base$SAT, main="Satisfacción")      #diagrama de caja
20
21  hist(base$SAT,main="Satisfacción")          #histograma
22
```

El diagrama de cajas nos permite detectar la presencia de casos atípicos en **cada variable**:



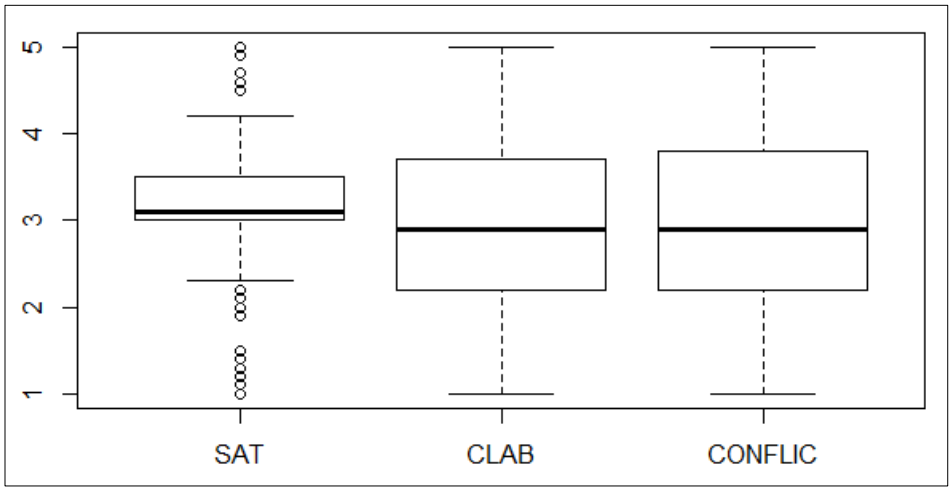
Los histogramas nos permiten conocer gráficamente la forma en que se distribuye **cada variable**, para tener una idea respecto a su comportamiento:




```
23  #Realizamos lo mismo con las otras dos variables
24
25  boxplot(base$CLAB, main="Carga Laboral")
26  hist(base$CLAB, main="Carga Laboral")
27
28  boxplot(base$CONFLIC, main="Conflictividad")
29  hist(base$CONFLIC, main="Conflictividad")
```

El siguiente comando puede ayudar a ahorrar tiempo pues muestra simultáneamente los diagrama de caja de cada una de las variables de la base:

```
33  boxplot(base)       #diagrama de cajas de cada una de las variables.
34
```



Normalidad Univariante:

Para comprobar normalidad univariante podemos mencionar la existencia de dos tests, cuyo uso es limitado según el tamaño muestral:

- Shapiro-Wilks: para tamaños muestrales que van entre 0 y 50 casos.
- Kolmogorov-Smirnov: para tamaños muestrales que oscilan entre 50 y 1000 casos.

Como tenemos 200 casos empleamos el test K-S

```
ks.test(base$SAT, "pnorm")
#"base de datos"$"variable" y tipo distribución (normal en este caso).

#Lo aplicamos a las otras variables
ks.test(base$CLAB, "pnorm")
ks.test(base$CONFLIC, "pnorm")
```

```
        One-sample Kolmogorov-Smirnov test

data:  base$SAT
D = 0.9213, p-value < 2.2e-16
alternative hypothesis: two-sided
```

En este caso el valor *p* (*p-value*) es menor a 0,05, por lo que la variable no se distribuye normalmente.

Otro modo de estimar la normalidad univariante, recomendado para cuando tengamos muestras grandes (más de 1.000 casos) y no podemos utilizar no Shapiro-Wilks ni Kolmogorov-Smirnov, es empleando el S de simetría para calcular el Z de simetría. Primero calcularemos el s de simetría, para posteriormente calcular el Z de simetría; la idea es que si nos da entre el intervalo de confianza especificado (+- 1,96), la variable distribuye normalmente. Emplearemos la siguiente ecuación:

$$Zsimetría = \frac{Simetría}{\sqrt{6/n}}$$

Para esto necesitamos descargar e instalar un paquete, y luego invocarlo:

```
install.packages("moments")      #Instalación paquete "moments"
library(moments)                 #Invocación del paquete al entorno R
```

Luego pedimos el estadístico de simetría; para ello creamos un objeto (sSAT → "simetría de la variable satisfacción"), al cual le asignamos el resultado del estadístico "skewness"; a este comando debemos especificar la base de datos y la respectiva variable a la cual nos estamos refiriendo:

```
sSAT<-skewness(base$SAT)
#se crea un objeto ("sSaT") que tendrá nuestro s de simetría
```

Empleando la fórmula ya especificada, creamos un objeto que contendrá el resultado del cálculo del Z de simetría, que posteriormente podremos observar para conocer su valor[1]:

```
ZsSAT<-sSAT/sqrt(6/200) #Creación de objeto que contendrá el cálculo
                        #del Z de simetria.
```

Ejecutamos el objeto para visualizar el valor del Z de simetría de la variable SAT; dado que escapa al intervalo de confianza establecido previamente ($\pm 1{,}96$), podemos concluir que la variable SAT no se distribuye de manera normal.

```
> ZsSAT
[1] -2.90052
>
```

Al calcular los Z de simetría de las otras dos variables, podemos concluir que – considerándolas por separado, en su dimensión univariante – tanto la variable CONFLIC como CLAB, se distribuyen de acuerdo a una curva normal:

```
> ZsCONFLIC
[1] 1.57147
>   ZsCLAB
[1] 1.577669
```

[1] "sqrt" quiere decir raíz cuadrada.

Normalidad multivariante

Para estimar la normalidad multivariante de nuestro conjunto de datos ocuparemos el test de Mardia. Para emplear este test necesitamos instalar un paquete adicional a R para mediciones de normalidad multivariante. Para ello emplearemos el siguiente comando:

```
#NORMALIDAD MULTIVARIANTE

  install.packages("MVN")   #Instalación paquete normalidad multivariante
  library("MVN")            # Invocación del paquete
```

Mediante el siguiente comando aplicamos el test de Mardia a nuestra base de datos; en este caso no especificamos variables, pues la base de datos está compuesta sólo por las tres variables que nos interesan:

```
mardiaTest(base) #Aplicación del test de mardia.
```

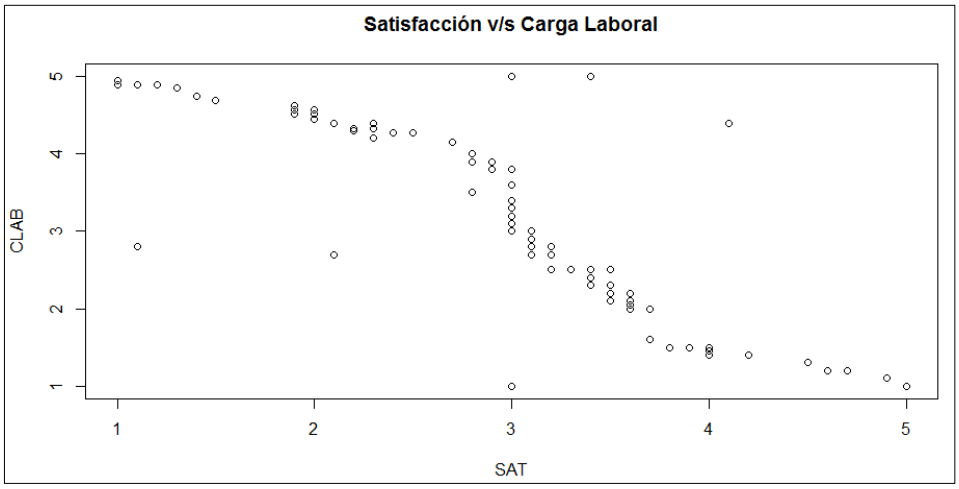
El resultado nos indica que el conjunto de variables no se distribuye normalmente en su dimensión multivariante:

```
   Mardia's Multivariate Normality Test
---------------------------------------
   data : base

   g1p            : 6.447252
   chi.skew       : 214.9084
   p.value.skew   : 1.24231e-40

   g2p            : 61.77281
   z.kurtosis     : 60.38343
   p.value.kurt   : 0

   chi.small.skew : 219.7721
   p.value.small  : 1.192889e-41

   Result          : Data are not multivariate normal.
---------------------------------------
```

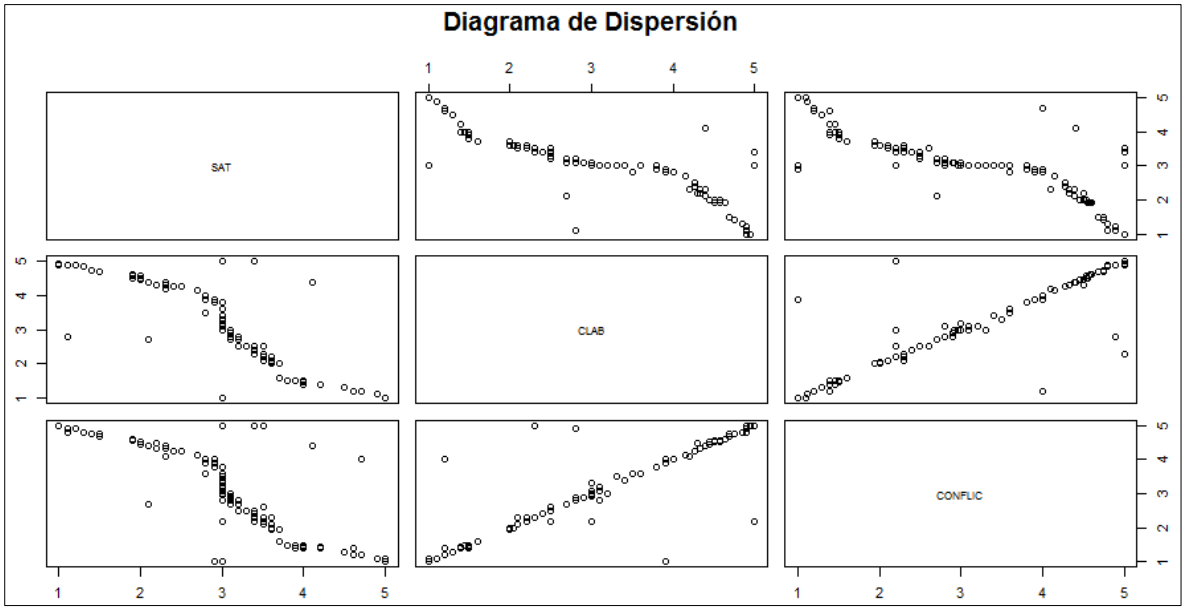
Colinealidad

Para evaluar la colinealidad en su dimensión bivariada, podemos hacer en primer lugar una inspección gráfica de la relación entre dos variables. Para ello pedimos un gráfica que nos permita visualizar la relación entre dos variables, que puede indicarnos de modo preliminar si hay algo de colinealidad, esto es de relación entre las variables. Algunas técnicas requieren que la variables estén relacionadas, otras requieren independencia entre las variables:

```
#Inspección visual: Gráfico de dos variables, titulado y con etiqueta de cada variable.

  plot(base$SAT,base$CLAB, main="Satisfacción v/s Carga Laboral", xlab="SAT", ylab="CLAB")
```



También podemos solicitar un digrama de dispersión que nos presente todas las relaciones entre las variables de la base de datos:



Podemos pedir la correlación entre dos variables, mediante el siguiente comando:

```
cor(base$SAT, base$CLAB)
```

El resultado indica que hay una fuerte correlación entre ambas variables:

```
> cor(base$SAT, base$CLAB)
[1] -0.838748
```

También podemos pedir la matriz de correlaciones, mediante el siguiente comando:

```
cor(base)
```

El resultado muestra un alto nivel de correlación entre todas las variables:

```
             SAT       CLAB     CONFLIC
SAT    1.0000000 -0.8387480 -0.8052449
CLAB  -0.8387480  1.0000000  0.8929269
CONFLIC -0.8052449  0.8929269  1.0000000
```

Multicolinealidad

Para evaluar la multicolinealidad, es decir, para evaluar la relación lineal entre las variables en su dimensión multivariante, ocuparemos una regresión lineal. Para ello creamos un objeto que contendrá los resultados de tal técnica:

```
regresion<-lm(formula=base$SAT~base$CLAB+base$CONFLIC,data=base)
```

Si bien parece ser compleja, la estructura del comando es sencilla y se explica como sigue:

- lm = modelo lineal.
- Luego viene la variable dependiente, esta se separa de las dependientes mediante el signo colita de chancho (~) → (altgr+"+").
- El separador entre cada variable dependiente es un signo más (+)

En vez de simplemente ejecutar el objeto, pedimos su resumen porque da más información:

```
summary(regresion)
```

En los resultados, observamos el Adjusted R Squared, que nos indica la normalidad multivariante (multicolinealidad). Para el caso del modelamiento de ecuaciones estructurales, por ejemplo interesa que sea bajo:

```
Residual standard error: 0.3827 on 197 degrees of freedom
Multiple R-squared:  0.7191, Adjusted R-squared:  0.7163
F-statistic: 252.2 on 2 and 197 DF,  p-value: < 2.2e-16
```

Si graficamos el objeto *regresion*, obtendremos una interpretación gráfica de los diferentes resultados de la regresión. Para ello debemos ir haciendo *enter* en la sintaxis, y los diferentes gráficos se irán desplegando en el visor:

```
plot(regresion)
```

Detección multivariante de casos atípicos

Haremos un método de detección de casos atípicos utilizando la D2 de Mahalanobis; en esta ocasión, emplearemos el objeto *baseA.* El comando *mvOutlier* calcula las distancias de Mahalanobis y nos indica cuales son candidatos a caso atípicos.

Primer creamos un objeto que contendrá el resultado del comando *mvOutlier* (atípicos multivariantes), al que se le precisa el conjunto de datos, si queremos un resultado gráfico y el método particular para calcular las distancias:

```
#D2 de Mahalanobis
  result<-mvOutlier(baseA, qqplot=FALSE, method = "adj.quan")
```

Luego pedimos visualizar el objeto *result*. El MD es la distancia de mahalanobis; los verdaderos (TRUE) son candidatos para ser atípicos (P<0,001).

```
20   43.395    TRUE
21   31.993    TRUE
22   22.910    TRUE
23   20.997    TRUE
24   19.039    TRUE
25   17.101    TRUE
26   12.375   FALSE
27   11.555   FALSE
28   11.479   FALSE
29   10.698   FALSE
```

Dentro de los resultados de este objeto (*result*) se encuentra otro objeto (*$newData*) que es la misma base de datos pero con los candidatos a casos atípicos excluidos.

```
$newData
    NUMEROEST PSULEN PSUMAT NEM
100        23    547    554 558
101        70    416    999 509
102        43    430    410 486
103        19    504    454 569
104        72    486    496 559
105        36    505    536 576
106        53    528    552 573
107       120    448    457 490
108       115    438    437 576
109        18    362    378 458
110       108    999    611 541
```

Podemos pedir que nos muestre los los cinco primeros casos del objeto *$newData* que están en el objeto *result:*

```
head(result$newData)
```

Luego ordenamos extraer el objeto *$newdata* del objeto *result*, y lo asignamos al objeto *base2*

```
base2<-result$newData
```

Finalmente podemos visualizar el objeto *base2* que corresponde a nuestra base de datos original pero habiéndola depurado de los casos atípicos. Esta queda guardada en el entorno de R y disponible para trabajarla como una base de datos depurada:

```
base2
```