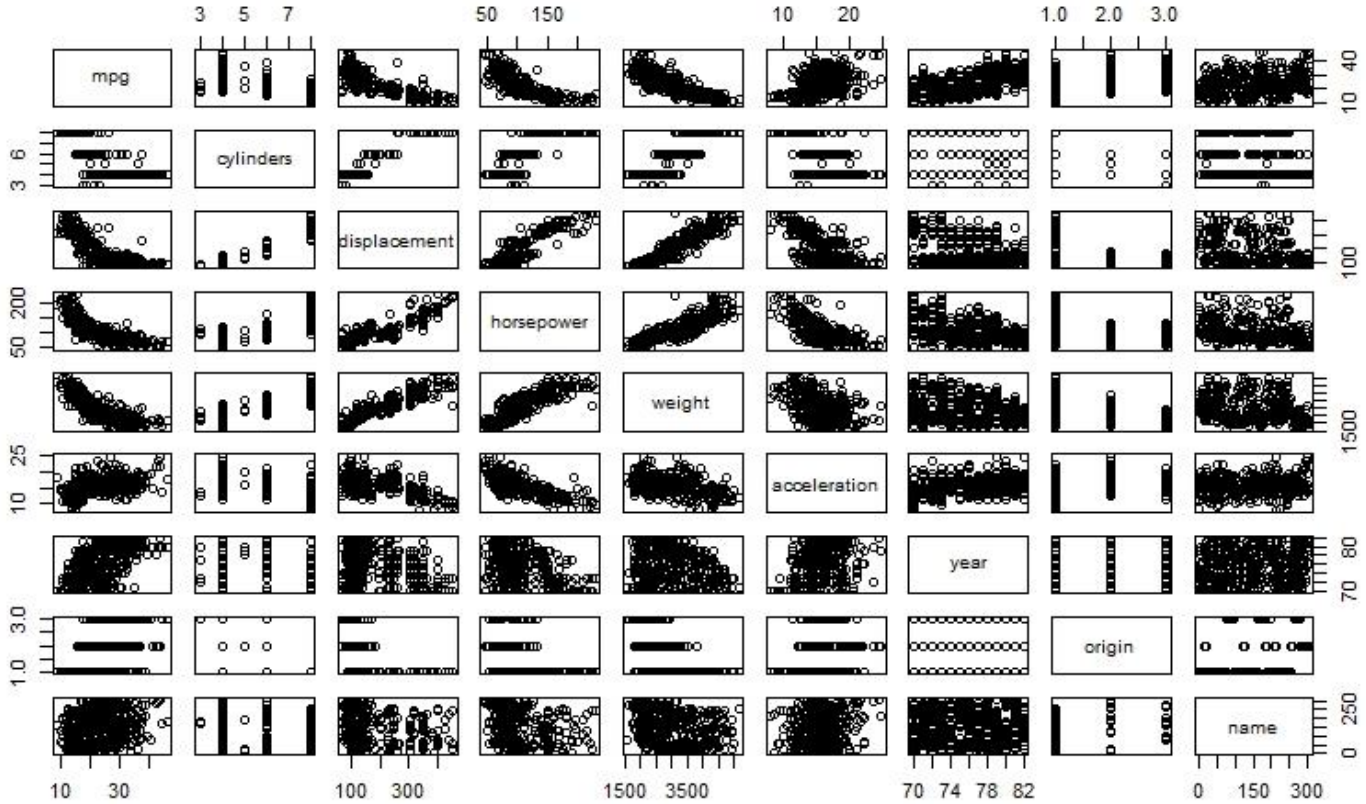
```
#=======================================
#  TALLER DE REGRESION LINEAL
#=======================================

library(MASS)
install.packages("ISLR")
library(ISLR)
install.packages("corrplot")
install.packages("GGally")

#1.- Selección de la data
data(Auto)
str(Auto)
summary(Auto)

#2.- Matriz con gráficos de dispersión de todas las variables de la base
pairs(Auto)
```

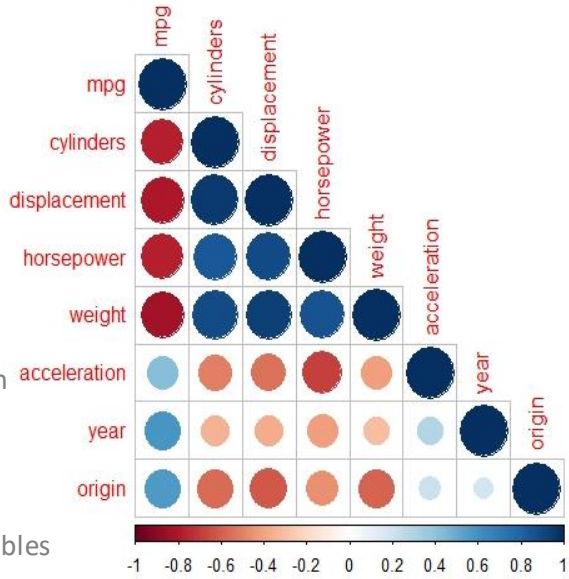



```
# A continuación, estudiamos la relación entre las variables para identificar
# cuales pueden ser los mejores predictores o si hay alguna con una relación
# tipo no lineal o detectar indicios de colinealidad (relación entre variables
# explicativas). Excluimos la variable cualitativa name

#3.- Matriz de correlación entre predictores
round(cor(subset(Auto, select = -name), method = "pearson"), digits = 3)
# Valores de correlación r próximos a 1 o -1 indican una alta correlación de
# variables. También podemos representarlos gráficamente
```

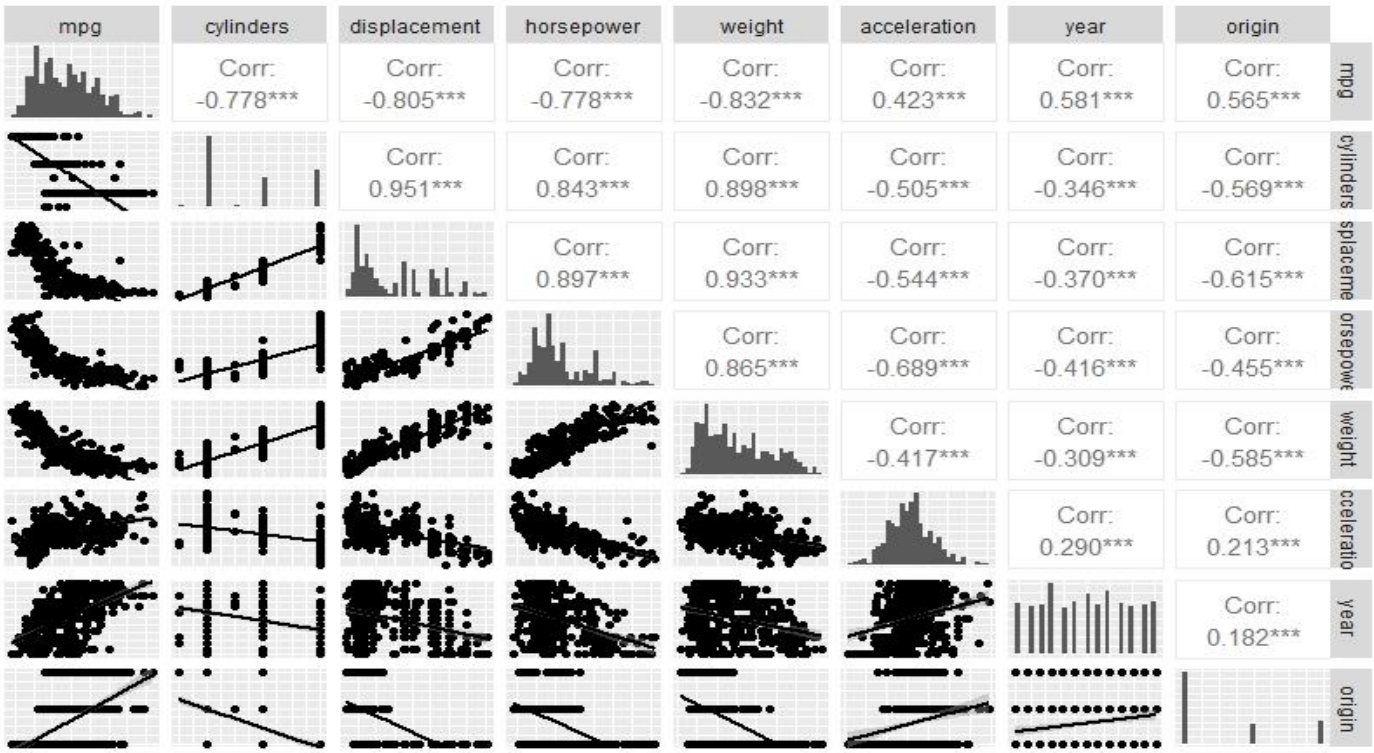
	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin
mpg	1.000	-0.778	-0.805	-0.778	-0.832	0.423	0.581	0.565
cylinders	-0.778	1.000	0.951	0.843	0.898	-0.505	-0.346	-0.569
displacement	-0.805	0.951	1.000	0.897	0.933	-0.544	-0.370	-0.615
horsepower	-0.778	0.843	0.897	1.000	0.865	-0.689	-0.416	-0.455
weight	-0.832	0.898	0.933	0.865	1.000	-0.417	-0.309	-0.585
acceleration	0.423	-0.505	-0.544	-0.689	-0.417	1.000	0.290	0.213
year	0.581	-0.346	-0.370	-0.416	-0.309	0.290	1.000	0.182
origin	0.565	-0.569	-0.615	-0.455	-0.585	0.213	0.182	1.000

```
require(corrplot)
corrplot(round(cor(subset(Auto, select = -name)), digits = 3), type = "lower")
```




```
#4.- Distribución de densidad de las variables cuantitativas del modelo
library(dplyr)
require(GGally)
ggpairs(select(Auto, -name), lower = list(continuous = "smooth"),
        diag = list(continuous = "bar"), axisLabels = "none")

#-----------------------------------------------------------------------
# De lo analizado hasta ahora podemos concluir que:
#    i) Las variables que mayor relación (no siendo del todo lineal) tienen
# con mpg son: displacement (r = -0.8), weight (r = -0.83),
# horsepower (r = -0.77) y cylinders (r = -0.77), siendo la relacion
# todas, negativas.
#   ii) Se observa una alta correlacion (colinealidad) entre pares de variables
#       como displacement y cylinders (r = 0.95) y displacement y
```



```
#       weight (r = 0.93). Con ello, posiblemente no seria util introducir
#       pares en el modelo
#  iii) La distribucion de las variables parece acercarse bastante a una
#       distribucion normal, dado el numero de observaciones con las que
#       disponemos.
#-------------------------------------------------------------------

# Vamos a generar el modelo con todos los predictores a excepcion de la variable
# name que proporciona el nombre del modelo del coche, y que en este caso es
# prescindible ya que no aporta informacion importante al modelo. R generara
# variables  dummy automaticamente para las variables cualitativas. Con la funcion
# contrasts() podriamos conocer que valor R ha asociado a cada nivel del

#5.- Creacion del modelo inicial
modelo.lineal <- lm(mpg ~ . - name, data = Auto)
summary(modelo.lineal)
#-----------------------------------------
# De lo analizado hasta ahora
# podemos concluir que:
#   i) El modelo con todas las variables
#       introducidas como
#       predictores es     de explicar el
#       82.15 % de la varianza observada
#       en el consumo de  combustible
#       (R2 ajustado = 0.818)
#   ii) El p-value del modelo es
#       significativo (2.2e-16), por lo que
#       podemos decir que el modelo es
#       util y que existe una relacion
#       entre los predictores y la variable
#       respuesta (al menos uno de los
#      coeficientes es distinto de 0)
#  iii) Los predictores que parecen tener una relacion estadisticamente
#       significativa con la variable de respuesta son: displacement, weight,
#       origin, a diferencia de cylinders, horsepower, y aceleration
#   iv) Ejemplo de interpretacion de coeficiente: por cada año que pasa, se
#       recorre mas distancia por volumen de combustible (??year = 0.75)
#       manteniendose el resto de predictores constante, es decir, aumenta la
#       eficiencia
#-------------------------------------------------------------------

#6.- Determinar la calidad del modelo
step(modelo.lineal, direction = "both", trace = 1)
```

```
Call:
lm(formula = mpg ~ . - name, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5903 -2.1565 -0.1169  1.8690 13.0604

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780
displacement   0.019896   0.007515   2.647  0.00844 **
horsepower    -0.016951   0.013787  -1.230  0.21963
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.328 on 384 degrees of freedom
Multiple R-squared:  0.8215,    Adjusted R-squared:  0.8182
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
Start:  AIC=950.5
mpg ~ (cylinders + displacement + horsepower + weight + acceleration +
    year + origin + name) - name

               Df Sum of Sq    RSS     AIC
- acceleration  1      7.36 4259.6  949.18
- horsepower    1     16.74 4269.0  950.04
<none>                     4252.2  950.50
- cylinders     1     25.79 4278.0  950.87
- displacement  1     77.61 4329.8  955.59
- origin        1    291.13 4543.3  974.46
- weight        1   1091.63 5343.8 1038.08
- year          1   2402.25 6654.5 1124.06

Step:  AIC=949.18
mpg ~ cylinders + displacement + horsepower + weight + year +
    origin

               Df Sum of Sq    RSS     AIC
<none>                     4259.6  949.18
- cylinders     1     27.27 4286.8  949.68
+ acceleration  1      7.36 4252.2  950.50
- horsepower    1     53.80 4313.4  952.10
- displacement  1     73.57 4333.1  953.89
- origin        1    292.02 4551.6  973.17
- weight        1   1310.43 5570.0 1052.32
- year          1   2396.17 6655.7 1122.13
```

```
Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    year + origin, data = Auto)

Coefficients:
 (Intercept)     cylinders  displacement    horsepower        weight          year        origin
  -15.563492     -0.506685      0.019269     -0.023895     -0.006218      0.747516      1.428242
```

acceleration (la variable con mayor p-value) ha sido la unica variable explicativa
en el proceso de seleccion. Reajustamos el modelo excluyendo dicha variable

#7.- Actualizando el modelo

```
modelo.lineal2<- update(modelo.lineal, formula =~ . -acceleration)
summary(modelo.lineal2)
```

```
Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    year + origin, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7604 -2.1791 -0.1535  1.8524 13.1209

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.556e+01  4.175e+00  -3.728 0.000222 ***
cylinders    -5.067e-01  3.227e-01  -1.570 0.117236
displacement  1.927e-02  7.472e-03   2.579 0.010287 *
horsepower   -2.389e-02  1.084e-02  -2.205 0.028031 *
weight       -6.218e-03  5.714e-04 -10.883  < 2e-16 ***
year          7.475e-01  5.079e-02  14.717  < 2e-16 ***
origin        1.428e+00  2.780e-01   5.138 4.43e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.326 on 385 degrees of freedom
Multiple R-squared:  0.8212,    Adjusted R-squared:  0.8184
F-statistic: 294.6 on 6 and 385 DF,  p-value: < 2.2e-16
```

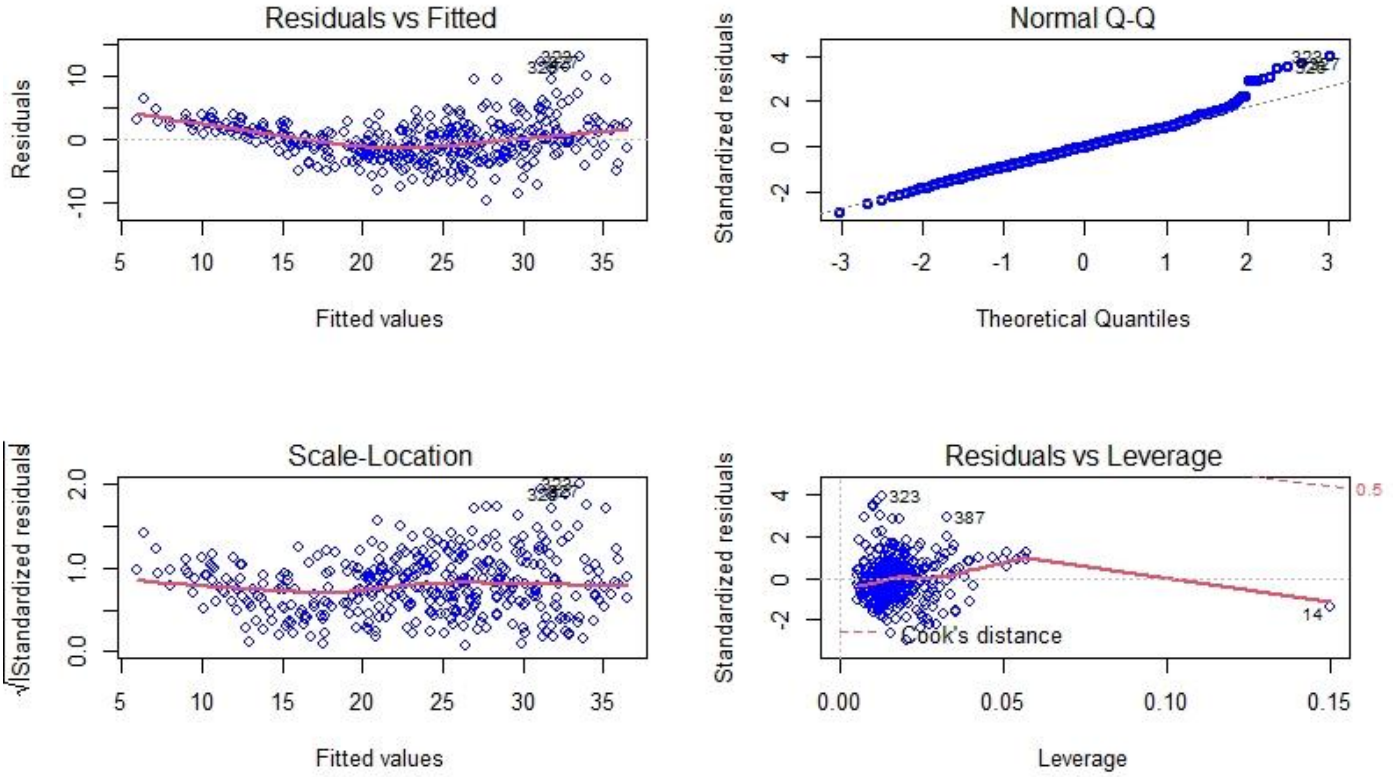
#8.- Los intervalos de confianza para cada uno de los coeficientes serian :

confint(modelo.lineal2)

	2.5 %	97.5 %
(Intercept)	-23.772628686	-7.354355925
cylinders	-1.141217264	0.127846990
displacement	0.004577392	0.033961179
horsepower	-0.045199801	-0.002590258
weight	-0.007341708	-0.005094914
year	0.647647254	0.847384650
origin	0.881647846	1.974835924

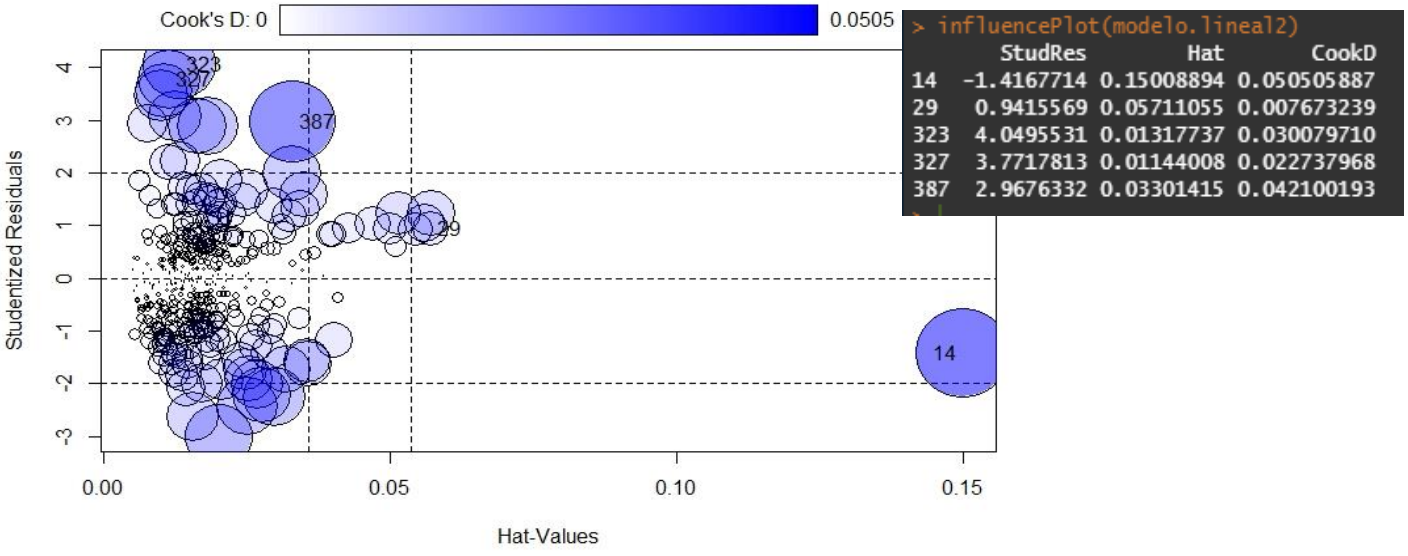
#9.- Visualizar los residuos

```
par(mfrow=c(2,2))
plot(modelo.lineal2, lwd=2, col="blue")
```



#10.- Deteccion y visualizacion de observaciones influyentes

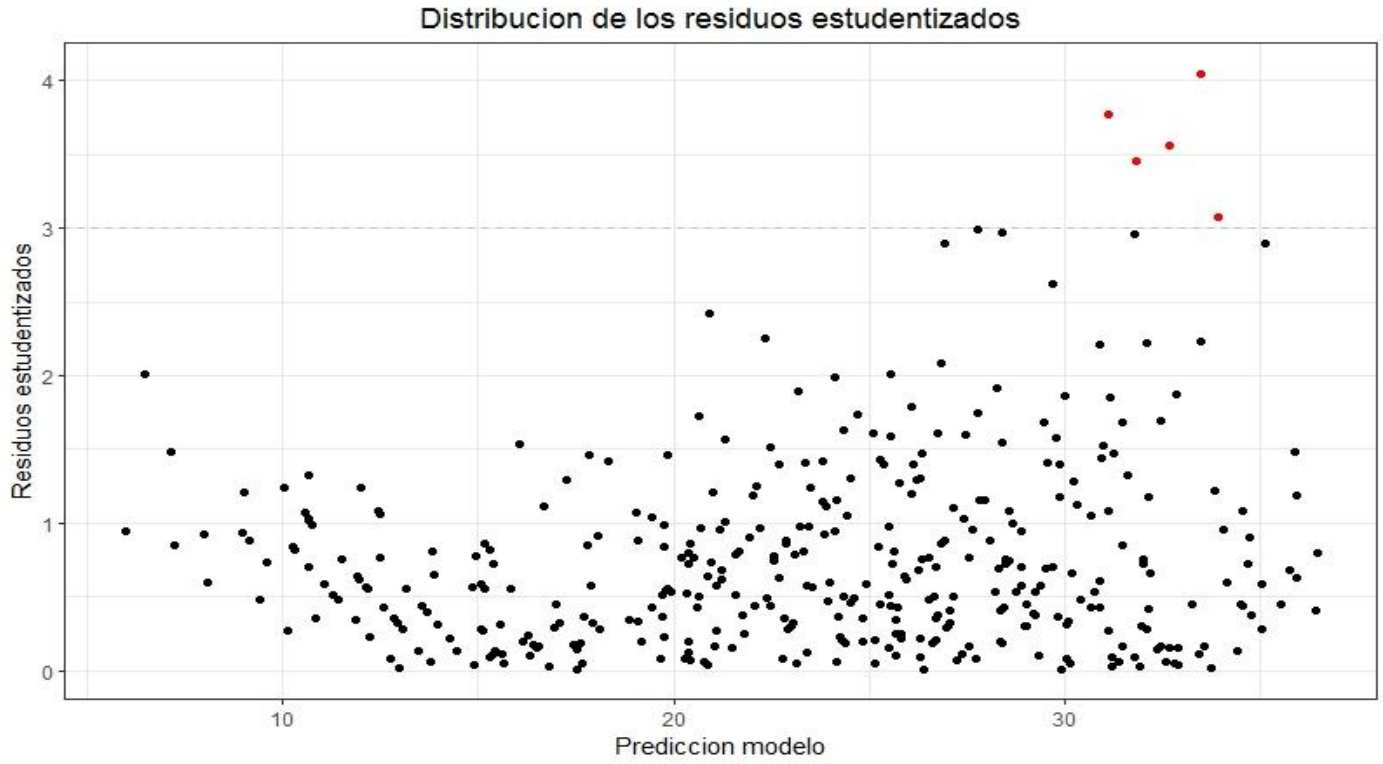
```
install.packages("car", dependencies = TRUE)
require(car)
par(mfrow=c(1,1))
influencePlot(modelo.lineal2)
```




```
> influencePlot(modelo.lineal2)
     StudRes        Hat       CookD
14 -1.4167714 0.15008894 0.050505887
29  0.9415569 0.05711055 0.007673239
323 4.0495531 0.01317737 0.030079710
327 3.7717813 0.01144008 0.022737968
387 2.9676332 0.03301415 0.042100193
```

#11.- Grafico de residuos estudentizados frente a valores ajustados por el modelo

```
library(ggplot2)
ggplot(data = Auto, aes(x = predict(modelo.lineal2),
    y = abs(rstudent(modelo.lineal2)))) + geom_hline(yintercept = 3,
    color = "grey", linetype = "dashed") +
    geom_point(aes(color = ifelse(abs(rstudent(modelo.lineal2)) > 3, "red",
    "black"))) + scale_colour_identity() +
    labs(title = "Distribucion de los residuos estudentizados",
    x = "Prediccion modelo", y = "Residuos estudentizados") +
    theme_bw() + theme(plot.title = element_text(hjust = 0.5))
```



#12.- Deteccion de los residuos estudentizados > 3 considerados como outlier

```
which(rstudent(modelo.lineal2) > 3)
outlierTest(modelo.lineal2)
```

#13.- Test de hipotesis para el analisis de normalidad de los residuos

```
shapiro.test(modelo.lineal2$residuals)
ks.test(modelo.lineal2$residuals, "pnorm")
```

```
> which(rstudent(modelo.lineal2) > 3)
245 323 326 327 394
243 321 324 325 389
> outlierTest(modelo.lineal2)
    rstudent unadjusted p-value Bonferroni p
323 4.049553         6.2098e-05      0.024343
```

```
> shapiro.test(modelo.lineal2$residuals)

        Shapiro-Wilk normality test

data:  modelo.lineal2$residuals
W = 0.97461, p-value = 2.327e-06

> ks.test(modelo.lineal2$residuals, "pnorm")

        One-sample Kolmogorov-Smirnov test

data:  modelo.lineal2$residuals
D = 0.26132, p-value < 2.2e-16
alternative hypothesis: two-sided
```

#14.- Test de contraste de homocedasticidad

```
# Breusch-Pagan
library(lmtest)
bptest(modelo.lineal2)
```

```
> bptest(modelo.lineal2)

        studentized Breusch-Pagan test

data:  modelo.lineal2
BP = 23.063, df = 6, p-value = 0.0007755
```

```
# Como hemos visto en el segundo paso del analisis, hay evidencias de alta
# colinealidad entre algunas variables. Podriamos utilizar la funcion vif()
# para calcular el factor de inflacion de la varianza y detectar variables
# con mayor colinealidad

corrplot(cor(select(Auto, cylinders, displacement, horsepower, weight, year,
                    origin)), method = "number", type = "lower")
#15.- Factores de inflacion de la varianza
vif(modelo.lineal2)
```

```
#15.- Factores de inflacion de la varianza
vif(modelo.lineal2)
 cylinders displacement   horsepower       weight         year       origin
 10.710150    21.608513     6.147752     8.324047     1.237304     1.772234
```


```
#-------------------------------------------------------------------
# Hasta el momento podemos concluir que:
#     i) El ajuste lineal parece no ser del todo preciso, ya que se
#        observa un patron curvo en los residuos frente a los valores
#        ajustados por el modelo, ademas de que no acaban de
#        distribuirse de forma homogenea en torno a 0. El test de
#        Breusch-Pagan tambien proporciona evidencias de falta de
#        homocedasticidad (p-value = 0.0007)
#    ii) El Q-Q plot refleja que hay indicios de falta de normalidad en los
#        residuos (aquellos de mayor valor), corroborado tambien por
#        el test de hipotesis de shapiro wilk (p-value = 2.32e-06)
#   iii) La observacion 14 parece tener un nivel alto de influencia, aunque
#        se considere como residuo de alta magnitud. La observacion 323 tambien
#        se considera influyente. Un analisis mas exhaustivo consistiria en
#        excluir las observaciones y ver el impacto sobre el modelo.
#    iv) Los predictores cylinders y displacement muestran una alta inflacion
#        la varianza
#     v) Cuatro de las seis variables que incluye el modelo estan muy
#        correlacionadas.
#-------------------------------------------------------------------

# Ya que algunas de las condiciones para el ajuste lineal no acaban de
# satisfacerse, y observando la matriz de correlacion podemos ver como la
# distribucion de las variables horsepower, displacement, y weight tiene un
# patron no lineal parecido frente a mpg, podriamos aproximar el ajuste
# utilizando un polinomio de grado 2. En el siguiente intento podemos incluir
# terminos polinomicos a estas variables y estudiar si el modelo mejora. Es
# importante no excederse en el grado de polinomio para evitar el "overfiting"
# ya que cuanto mayor es el polinomio, mas flexible es el modelo

#16.- Reajuste del modelo
modelo.lineal.poli <- update(modelo.lineal2, formula = ~ . +
                             poly(displacement, 2) + poly(horsepower, 2) +
                             poly(weight, 2))
summary(modelo.lineal.poli)
```

```
> summary(modelo.lineal.poli)

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    year + origin + poly(displacement, 2) + poly(horsepower,
    2) + poly(weight, 2), data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0799 -1.5267 -0.0789  1.4437 11.8994

Coefficients: (3 not defined because of singularities)
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.957e+01  3.671e+00  -5.332 1.67e-07 ***
cylinders              3.633e-01  3.271e-01   1.110 0.267519
displacement          -2.480e-03  7.493e-03  -0.331 0.740795
horsepower            -4.358e-02  1.043e-02  -4.177 3.66e-05 ***
weight                -4.710e-03  5.739e-04  -8.206 3.54e-15 ***
year                   7.790e-01  4.487e-02  17.362  < 2e-16 ***
origin                 5.704e-01  2.674e-01   2.133 0.033561 *
poly(displacement, 2)1         NA         NA      NA       NA
poly(displacement, 2)2 1.223e+01  6.397e+00   1.912 0.056593 .
poly(horsepower, 2)1           NA         NA      NA       NA
poly(horsepower, 2)2   1.466e+01  4.326e+00   3.388 0.000777 ***
poly(weight, 2)1               NA         NA      NA       NA
poly(weight, 2)2       1.599e+01  4.670e+00   3.424 0.000683 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.909 on 382 degrees of freedom
Multiple R-squared:  0.8643,    Adjusted R-squared:  0.8611
F-statistic: 270.3 on 9 and 382 DF,  p-value: < 2.2e-16
```

#17.- Test de hipotesis para evaluar si un modelo se ajusta mejor que el original

anova(modelo.lineal2, modelo.lineal.poli)

```
> #17.- Test de hipotesis para evaluar si un modelo se ajusta mejor que el original
> anova(modelo.lineal2, modelo.lineal.poli)
Analysis of Variance Table

Model 1: mpg ~ cylinders + displacement + horsepower + weight + year +
    origin
Model 2: mpg ~ cylinders + displacement + horsepower + weight + year +
    origin + poly(displacement, 2) + poly(horsepower, 2) + poly(weight,
    2)
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1    385 4259.6
2    382 3232.8  3    1026.7 40.44 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

#----------

Incluyendo terminos polinomicos (siendo en displacement menos significativo)
hemos conseguido mejorar el modelo y que explique casi un 5% mas de la
variabilidad (R2ajustado = 0,8611 y p-value de ANOVA = 2.2e-16). Las
observaciones 323 y 14 podrian estar influyendo en el modelo

#----------

```
x <- log(Auto$acceleration)
par(mfrow=c(1,1))
hist(x, prob=T)
```