Regresión Lineal

Packages base de datos

```
library(caret)

## Loading required package: ggplot2
## Loading required package: lattice

setwd("C:\\Users\\81799\\OneDrive\\Documentos\\ESFM_CLASES\\Servicio Social ARTF\\Machine Learni
auto <- read.csv("auto-mpg.csv")
head(auto,5)

##   No mpg cylinders displacement horsepower weight acceleration model_year
## 1  1  28         4          140         90   2264         15.5         71
## 2  2  19         3           70         97   2330         13.5         72
## 3  3  36         4          107         75   2205         14.5         82
## 4  4  28         4           97         92   2288         17.0         72
## 5  5  21         6          199         90   2648         15.0         70
##               car_name
## 1 chevrolet vega 2300
## 2     mazda rx2 coupe
## 3        honda accord
## 4     datsun 510 (sw)
## 5         amc gremlin
```

auto es un DataFrame en donde se tiene:

- mpg: Consumo del coche de millas por galeón.
- cylinders: Cilindros (Variable categórica).
- displacement: Desplazamiento.
- horsepower: Caballos
- weight:Peso
- acceleration:Aceleración
- model_year: Año del modelo.
- car_name: Nombre de la marca del carro

```
auto <- auto[,-1]
colnames(auto)=c("Consumo","Cilindros", "Desplazamiento", "Caballos","Peso", "Aceleracion","Mode
head(auto,5)

##   Consumo Cilindros Desplazamiento Caballos Peso Aceleracion Modelo
## 1      28         4            140       90 2264        15.5     71
## 2      19         3             70       97 2330        13.5     72
## 3      36         4            107       75 2205        14.5     82
## 4      28         4             97       92 2288        17.0     72
## 5      21         6            199       90 2648        15.0     70
##                 Marca
## 1 chevrolet vega 2300
## 2     mazda rx2 coupe
```

```
## 3        honda accord
## 4     datsun 510 (sw)
## 5          amc gremlin
```

Convirtiendo la variable categórica a factor, indicando los niveles en formato de etiqueta

```
auto$Cilindros <- factor(auto$Cilindros ,
                         levels = c(3,4,5,6,8),
                         labels = c("3c", "4c", "5c", "6c", "8c"))
```

Para el método de *lm* no hay problema de que tengamos variables categóricas, ya que la función los detecta (Aquí no hay que generar variables dummies).
Genero mi semilla y mi partición de los datos de entrenamiento.

```
set.seed(1)#Genera la semilla
t.id <- createDataPartition(auto$Consumo , p = 0.7, list = F)#Entrenamiento al 70%
```

Antes de hacer mi predición observo las variables que me interesarína para obtener una estimación del consumo del coche

```
names(auto)

## [1] "Consumo"        "Cilindros"      "Desplazamiento" "Caballos"
## [5] "Peso"           "Aceleracion"    "Modelo"         "Marca"
```

Observó que en este caso, la marca no me interea ni tampoco el modelo.
Predecir el consumo del coche en función del resto de categorias

```
mod <- lm(Consumo ~ ., data = auto[t.id,-c(7,8)])#Con los datos de entrenamiento
mod

##
## Call:
## lm(formula = Consumo ~ ., data = auto[t.id, -c(7, 8)])
##
## Coefficients:
##    (Intercept)     Cilindros4c     Cilindros5c     Cilindros6c     Cilindros8c
##      39.544596       10.878999       13.447272        7.445811       11.063253
## Desplazamiento        Caballos            Peso     Aceleracion
##      -0.005945       -0.082137       -0.004102       -0.282439
```

Con el anterior modelo obtendríamos lo siguiente:

$$\begin{aligned} Consumo &= 39.5 + 10.9(4c) + 13.4(5c) + 7.4(6c) + 11(8c) + 0.006(Desplazamiento) - 0.08(Caballos) \\ &= -0.004(Peso) + 0.02(Aceleración) \end{aligned}$$

Recordemos que:
Prueba de significancia de la regresión: La prueba individual de un coeficiente de regresión puede ser útil para determinar si:

- Se incluyen otra variable regresora.
- Se elimina uno o más variables regresoras presentes en el modelo.

La hipótesis para probar la significancia de cualquier coeficiente de regresión es:

$$\begin{aligned} H_0: &\quad \beta_i = 0 \\ H_a: &\quad \beta_i \neq 0 \end{aligned}$$

Obteniendo un análisis del modelo anterior:

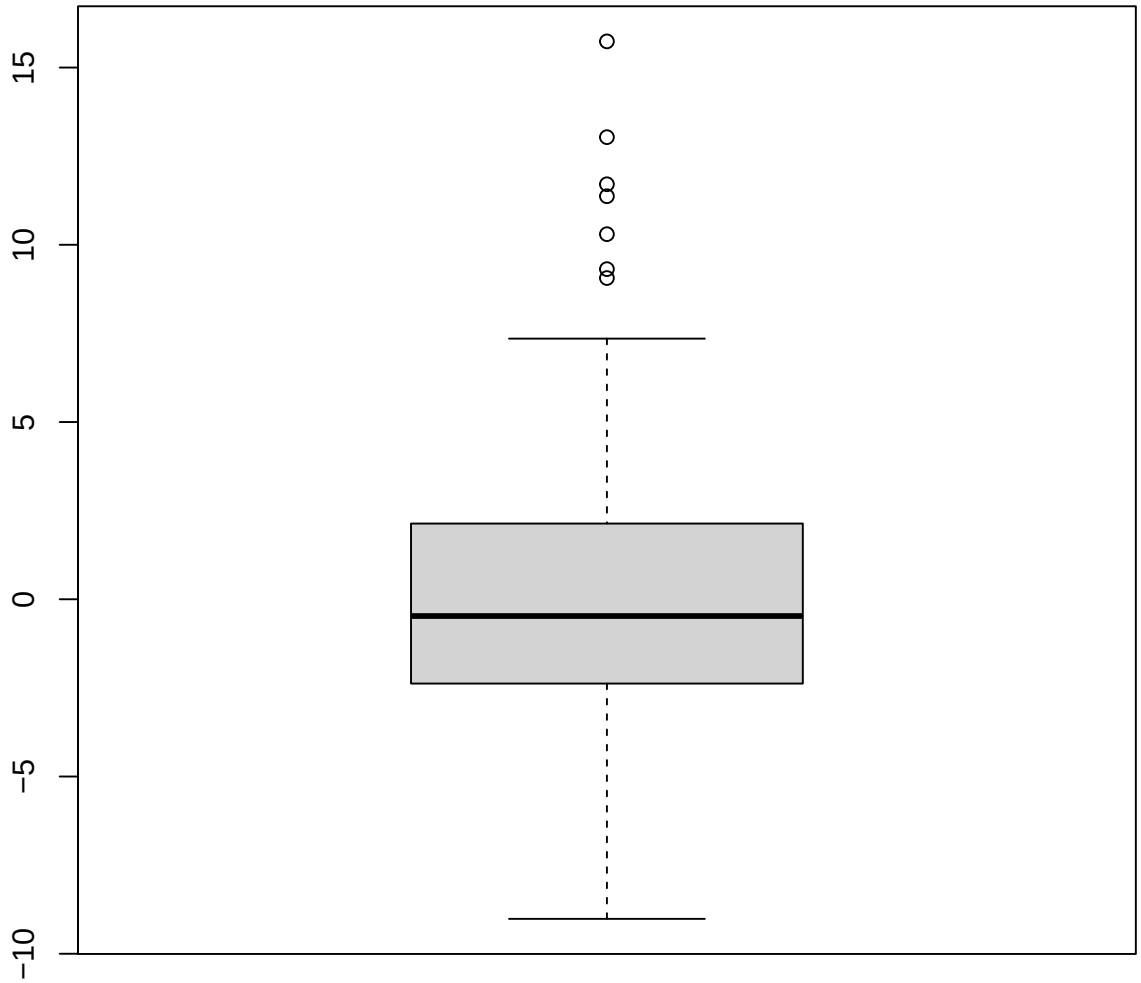
```
summary(mod)
```

```
##
## Call:
## lm(formula = Consumo ~ ., data = auto[t.id, -c(7, 8)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0153 -2.3610 -0.4723  2.1336 15.7380
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    39.5445959  3.6913345  10.713  < 2e-16 ***
## Cilindros4c    10.8789988  2.7301319   3.985 8.69e-05 ***
## Cilindros5c    13.4472717  3.8142431   3.526 0.000496 ***
## Cilindros6c     7.4458106  2.9589840   2.516 0.012436 *
## Cilindros8c    11.0632530  3.3283272   3.324 0.001010 **
## Desplazamiento -0.0059451  0.0097074  -0.612 0.540763
## Caballos       -0.0821373  0.0180071  -4.561 7.71e-06 ***
## Peso           -0.0041022  0.0008576  -4.783 2.83e-06 ***
## Aceleracion    -0.2824392  0.1373667  -2.056 0.040731 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.749 on 271 degrees of freedom
## Multiple R-squared:  0.7743,Adjusted R-squared:  0.7676
## F-statistic: 116.2 on 8 and 271 DF,  p-value: < 2.2e-16
```

Aplicando prueba de significancia dando un $\alpha = 0.05$, observamos que aceptaremos la hipótesis para los $p - valor > 0.05$.
También observamos que tenemos un p-value: $<$ 2.2e-16, esto quiere decir que el modelo que se ha generado es bueno (También se puede observar mediante Adjusted R-squared: 0.7676)

Podemos observar como se distribuyen los residuos mediante un diagrama de bigote:

```
boxplot(mod$residuals)
```



Me diante eldiagram podemos observar que el residuo más pequeño se encuentra en un valor -9.0153 y el error máximo está en 15.7380.
También podemos calcular la raíz del error cuadrático medio de forma manual.

```
sqrt(mean((mod$fitted.values - auto[t.id,]$Consumo)^2))

## [1] 3.68808
```

Haciendo la predicción y sacando la raíz de dicha predicción tenemos:

```
pred <- predict(mod, auto[-t.id, -c(7,8)])
sqrt(mean((pred - auto[-t.id,]$Consumo)^2))

## [1] 4.693771
```

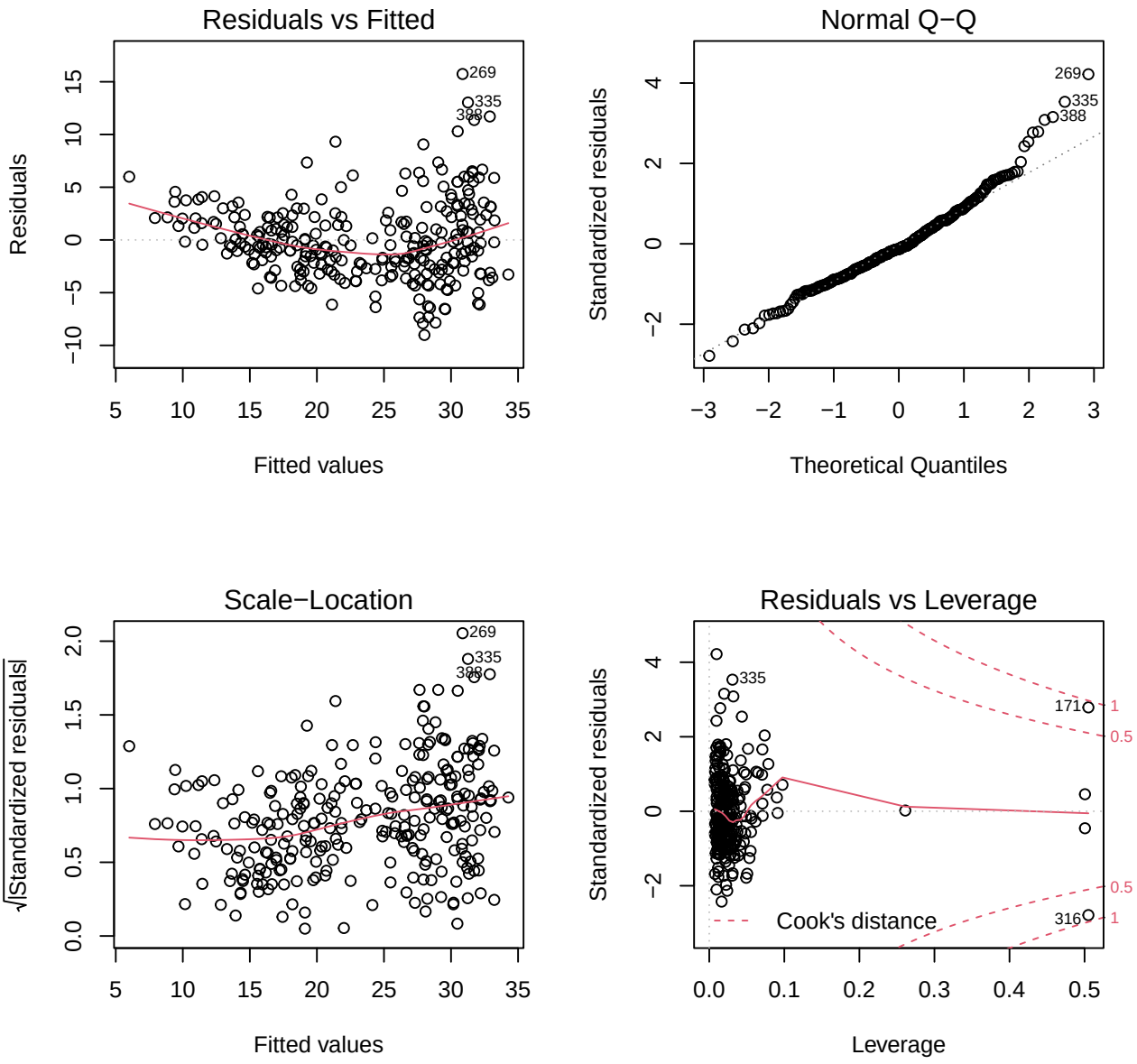
Observamos que el modelo es bueno, ya que tenemos un margen de error de ± 4.7

Gráfico de los residuos en un modelo lineal

$$
\begin{aligned}
X &= \{(x_{i1}, x_{i2}, \ldots, x_{ik}) : \forall\, 1 \leq i \leq n\} \\
y_i &\sim (x_{i1}, x_{i2}, \ldots, x_{ik}) \\
\hat{y}_i &= \beta_0 + \beta_1 y_{i1} + \cdots + \beta_k x_{ik} \\
y_i &= \beta_0 + \beta_1 y_{i1} + \cdots + \beta_k x_{ik} + \varepsilon_i
\end{aligned}
$$

Residuos: $\varepsilon_i = y_i - \hat{y}_i$

```
par(mfrow=c(2,2))#Matriz de 2 filas por 2 columnas
plot(mod) #Gráficos de los residuos
```



Interpretando los 4 gráficos dados:

- **Residuals vs Fitted**, es decir, Residuos vs Valores Ajustados: En este gráfico se muestra si los residuos tiene un patrón NO lineal.

En este caso podremos observar como se podría formar una parábola, es decir, la relación que existe entre las variables independientes y dependientes NO es lineal, sino que podría ser cuadrática, cúbica o logarítmica.

- **Normal Q-Q:** Este es un gráfico cuantil cuantil en el cual se representa si los errores se distribuyen según una normal estándar.

  En este caso podemos observar que los errores se distribuyen a una distribución normal, ya que se acercan mucho a la recta.

- **Scale-Location:** Este gráfico habla sobre la escala y localización de los residuos, es decir, nos intenta explicar acerca de los rangos que toman los errores de los predictores.

  En este caso es lo mismo que el primer gráfico, solamente que aquí los valores de los residuos están estandarizados.

- **Residuals vs Leverage:** En este gráfico nos muestra la representación de los residuos frente al apalancamiento, es decir, nos ayuda a encontrar posibles sujetos influyentes dentro de nuestro modelo (Si están fuera de la zona de Cooks distance).

  En nuestro caso tenemos que las filas 171 y 316 influyen en la regresión. así que tendríamos que localizarlo y excluirlo (Puede que tengamos un mejor modelo sin estos caso) ya que podría ser un dato erróneo.

En este caso, podemos utilzar que el modelo tome como referencia "4cza que en el modelo anterior se uso de referencia "3c

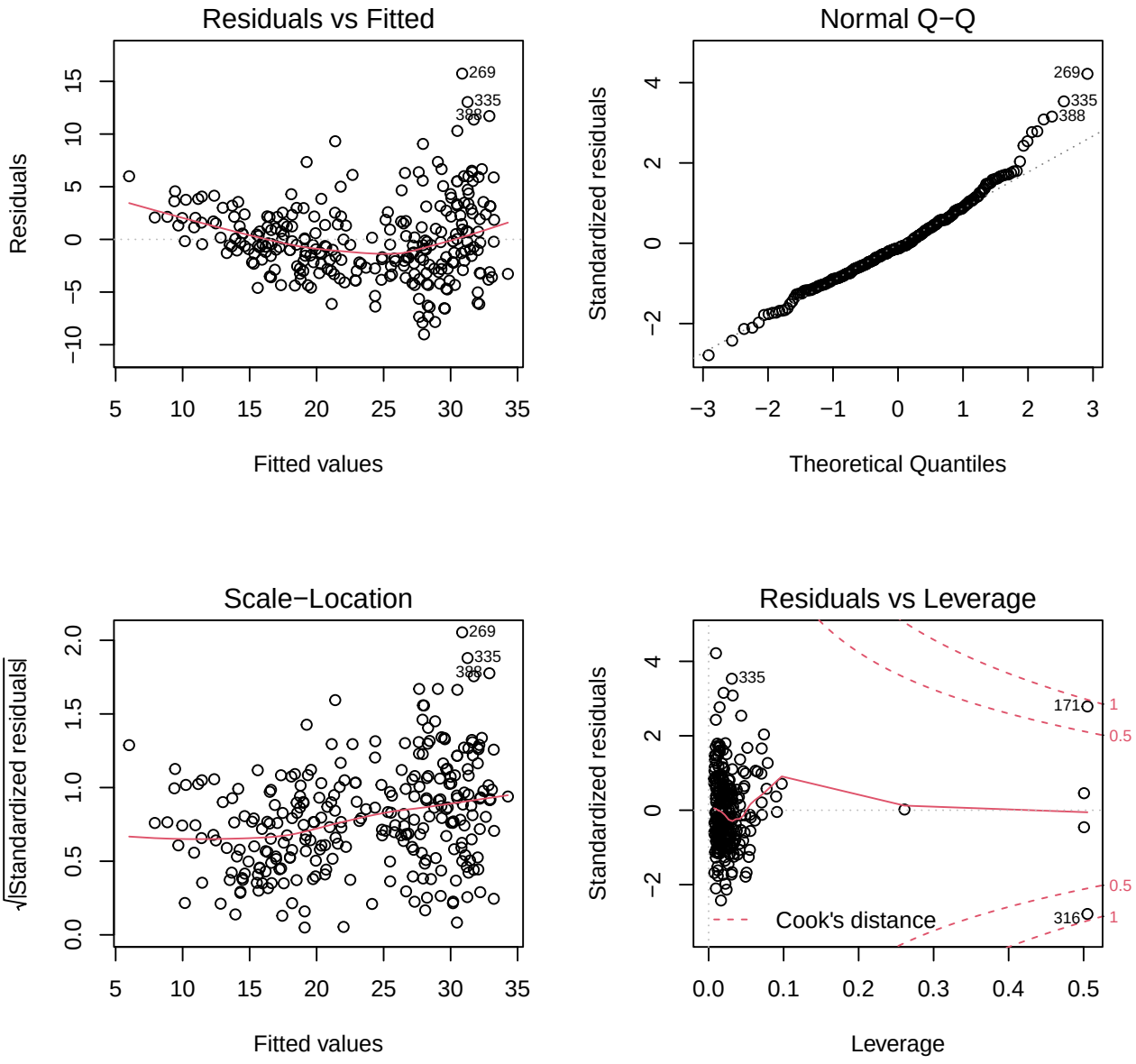
```
auto <- within(auto,
               Cilindros <- relevel(Cilindros, ref="4c"))
mod_4c <- lm(Consumo ~. , data = auto[t.id, -c(7,8)])
summary(mod_4c)

##
## Call:
## lm(formula = Consumo ~ ., data = auto[t.id, -c(7, 8)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0153 -2.3610 -0.4723  2.1336 15.7380
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.042e+01  2.714e+00  18.576  < 2e-16 ***
## Cilindros3c    -1.088e+01  2.730e+00  -3.985 8.69e-05 ***
## Cilindros5c     2.568e+00  2.696e+00   0.952 0.341712
## Cilindros6c    -3.433e+00  9.809e-01  -3.500 0.000543 ***
## Cilindros8c     1.843e-01  1.723e+00   0.107 0.914923
## Desplazamiento -5.945e-03  9.707e-03  -0.612 0.540763
## Caballos       -8.214e-02  1.801e-02  -4.561 7.71e-06 ***
## Peso           -4.102e-03  8.576e-04  -4.783 2.83e-06 ***
## Aceleracion    -2.824e-01  1.374e-01  -2.056 0.040731 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.749 on 271 degrees of freedom
## Multiple R-squared:  0.7743,Adjusted R-squared:  0.7676
## F-statistic: 116.2 on 8 and 271 DF,  p-value: < 2.2e-16
```

```
pred <- predict(mod_4c, auto[-t.id, -c(7,8)])
sqrt(mean((pred-auto[-t.id,]$Consumo )^2))

## [1] 4.693771

par(mfrow=c(2,2))#Matriz de 2 filas por 2 columnas
plot(mod_4c)
```



La función step para simplificar el modelo lineal.

```
library(MASS)
mod_4c

##
## Call:
## lm(formula = Consumo ~ ., data = auto[t.id, -c(7, 8)])
##
## Coefficients:
##    (Intercept)     Cilindros3c     Cilindros5c     Cilindros6c     Cilindros8c
```

```
##      50.423595      -10.878999        2.568273       -3.433188        0.184254
## Desplazamiento        Caballos            Peso     Aceleracion
##      -0.005945       -0.082137       -0.004102       -0.282439
```

```
summary(mod_4c)
```

```
##
## Call:
## lm(formula = Consumo ~ ., data = auto[t.id, -c(7, 8)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0153 -2.3610 -0.4723  2.1336 15.7380
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.042e+01  2.714e+00  18.576  < 2e-16 ***
## Cilindros3c    -1.088e+01  2.730e+00  -3.985 8.69e-05 ***
## Cilindros5c     2.568e+00  2.696e+00   0.952 0.341712
## Cilindros6c    -3.433e+00  9.809e-01  -3.500 0.000543 ***
## Cilindros8c     1.843e-01  1.723e+00   0.107 0.914923
## Desplazamiento -5.945e-03  9.707e-03  -0.612 0.540763
## Caballos       -8.214e-02  1.801e-02  -4.561 7.71e-06 ***
## Peso           -4.102e-03  8.576e-04  -4.783 2.83e-06 ***
## Aceleracion    -2.824e-01  1.374e-01  -2.056 0.040731 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.749 on 271 degrees of freedom
## Multiple R-squared:  0.7743,Adjusted R-squared:  0.7676
## F-statistic: 116.2 on 8 and 271 DF,  p-value: < 2.2e-16
```

```
#Criterio de Información AIC
step.model <- stepAIC(mod_4c, direction="backward")#Elimina predictores
```

```
## Start:  AIC=748.86
## Consumo ~ Cilindros + Desplazamiento + Caballos + Peso + Aceleracion
##
##                  Df Sum of Sq    RSS    AIC
## - Desplazamiento  1      5.27 3813.8 747.25
## <none>                        3808.5 748.86
## - Aceleracion     1     59.41 3868.0 751.19
## - Caballos        1    292.40 4100.9 767.57
## - Peso            1    321.56 4130.1 769.55
## - Cilindros       4    739.32 4547.9 790.53
##
## Step:  AIC=747.25
## Consumo ~ Cilindros + Caballos + Peso + Aceleracion
##
##               Df Sum of Sq    RSS    AIC
## <none>                      3813.8 747.25
## - Aceleracion  1     56.27 3870.1 749.35
## - Caballos     1    344.99 4158.8 769.49
## - Peso         1    447.46 4261.3 776.31
## - Cilindros    4    744.44 4558.3 789.17
```

Al final nos da un modelo significativo pero con la diferencia que tendremos menos variables.

```
summary(step.model)

##
## Call:
## lm(formula = Consumo ~ Cilindros + Caballos + Peso + Aceleracion,
##     data = auto[t.id, -c(7, 8)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0826 -2.4098 -0.3328  2.1053 15.7793
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.041e+01  2.711e+00  18.592  < 2e-16 ***
## Cilindros3c -1.058e+01  2.684e+00  -3.943 0.000102 ***
## Cilindros5c  2.618e+00  2.692e+00   0.972 0.331677
## Cilindros6c -3.772e+00  8.088e-01  -4.664 4.87e-06 ***
## Cilindros8c -4.965e-01  1.315e+00  -0.378 0.706039
## Caballos    -8.535e-02  1.721e-02  -4.960 1.24e-06 ***
## Peso        -4.335e-03  7.675e-04  -5.649 4.06e-08 ***
## Aceleracion -2.732e-01  1.364e-01  -2.003 0.046141 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.745 on 272 degrees of freedom
## Multiple R-squared:  0.774,Adjusted R-squared:  0.7681
## F-statistic:   133 on 7 and 272 DF,  p-value: < 2.2e-16
```