Regresión lineal múltiple

Efrén Antonio Jiménez Delgado

1 de setiembre de 2016

Análisis del Problema

El desempeño de un automovil se puede medir de diferentes formas. Algunas comunes son la cantidad de caballos de fuerza y el rendimiento del mismo, que se puede resumir en cuántas millas puede recorrer el automóvil por cada galón de combustible que consume. Para los clientes, potenciales compradores de un automóvil, este rendimiento es importante pues puede ayudar a tomar una decisión con respecto a cuál automovil comprar (si, por ejemplo, el cliente quiere un auto que rinda por muchas millas y pueda economizar en la compra de combustible).

Desde este punto de vista, tanto a clientes como a fabricadores de automóviles, les conviene entender cuál es la relación entre diferentes características del automóvil y su rendimiento, pues el conocer estas relaciones les puede ayudar a inferir cuál va a ser la eficiencia del vehículo a partir de ver los valores de otras características. Para fabricantes, puede ser importante conocer estas relaciones para saber cómo hacer cada modelo más eficiente con respecto al anterior.

Entendimiento de los Datos

Con el fin de analizar y tratar de estimar las millas por galón de diferentes modelos de automóviles, se trabajó con un conjunto de datos que contiene 398 observaciones y 9 variables:

- mpg (millas por galón): numérica, con un rango de 9 a 46.60.
- cyl (cilindraje): categórica ordinal, con valores posibles de 3, 4, 5, 6 y 8.
- disp (desplazaiento): numérica, con un rango de 68 a 455.
- hp (caballos de fuerza): numérica, con un rango de 46 a 230 y 6 valores faltantes.
- weight (peso): numérica, con un rango de 1613 a 5140.
- acc (aceleración): numérica, con un rango de 8 a 24.80.
- model year (año): categórica, con 13 valores diferentes representando el año del automóvil.
- origin (origen): categórica, 3 valores posibles: 1, 2, ó 3.
- model name (nombre del modelo): categórica, con 305 posibles valores.

Exploración de los Datos

```
# librerías utilizadas
library(caTools)

# establezca el directorio de trabajo
setwd("D:\\Drive\\Universidad\\UTN\\2016\\III Cuatrimestre\\mineria_2016_III_cuatri\\Clase 11\\Regresio

# cargue el archivo a una variable que se llame autos usando la función
# read.table
autos <- read.csv("auto-mpg.txt", header = F, na.strings = "?")
autos <- data.frame(do.call("rbind", strsplit(as.character(autos$V1), "    ",
    fixed = TRUE)))
```

```r
colnames(autos) <- c("mpg", "cyl", "disp", "hp", "weight", "acc", "model.year",
    "origin", "model.name")

# cambiar las variables que corresponden a numéricas
autos$mpg <- as.numeric(as.character(autos$mpg))
autos$disp <- as.numeric(as.character(autos$disp))
autos$hp <- as.numeric(as.character(autos$hp))
```

```
## Warning: NAs introducidos por coerción
```

```r
autos$weight <- as.numeric(as.character(autos$weight))
autos$acc <- as.numeric(as.character(autos$acc))

# Utilice la función str() para ver la estructura del conjunto de datos:
str(autos)
```

```
## 'data.frame':    398 obs. of  9 variables:
##  $ mpg       : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cyl       : Factor w/ 5 levels "3","4","5","6",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ disp      : num  307 350 318 304 302 429 454 440 455 390 ...
##  $ hp        : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight    : num  3504 3693 3436 3433 3449 ...
##  $ acc       : num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year: Factor w/ 13 levels "70","71","72",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ origin    : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ model.name: Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 142 55 224 242 2
```

```r
# Dividir el conjunto de datos en uno de entrenamiento y otro de pruebas:
set.seed(1376)
splt <- sample.split(autos$mpg, SplitRatio = 0.7)
autos.entrenamiento <- autos[splt, ]
autos.prueba <- autos[!splt, ]
```

Es importante siempre validar los rangos de los conjuntos de datos creados, para evitar caer en extrapolación:

```r
summary(autos.entrenamiento)
```

```
##       mpg        cyl          disp             hp            weight
##  Min.   : 9.00   3:  3   Min.   : 70.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.50   4:155   1st Qu.: 98.0   1st Qu.: 75.0   1st Qu.:2220
##  Median :23.00   5:  2   Median :144.0   Median : 92.0   Median :2774
##  Mean   :23.88   6: 63   Mean   :188.7   Mean   :103.4   Mean   :2943
##  3rd Qu.:29.80   8: 70   3rd Qu.:250.0   3rd Qu.:120.5   3rd Qu.:3465
##  Max.   :46.60           Max.   :455.0   Max.   :230.0   Max.   :5140
##                                          NA's   :5
##       acc          model.year  origin              model.name
##  Min.   : 8.00   73     : 32   1:177   chevrolet impala  :  4
##  1st Qu.:13.90   78     : 27   2: 54   ford maverick     :  4
##  Median :15.50   80     : 27   3: 62   toyota corolla    :  4
##  Mean   :15.65   76     : 23           amc matador       :  3
##  3rd Qu.:17.30   79     : 23           chevrolet citation:  3
##  Max.   :24.60   81     : 23           chevrolet nova    :  3
##                  (Other):138           (Other)           :272
```

```
summary(autos.prueba)
```

```
##       mpg        cyl          disp             hp             weight
##  Min.   :10.0   3: 1   Min.   : 68.0   Min.   : 49.0   Min.   :1649
##  1st Qu.:16.5   4:49   1st Qu.:108.0   1st Qu.: 78.0   1st Qu.:2265
##  Median :21.5   5: 1   Median :199.0   Median : 96.5   Median :2945
##  Mean   :22.5   6:21   Mean   :206.7   Mean   :107.4   Mean   :3047
##  3rd Qu.:28.0   8:33   3rd Qu.:302.0   3rd Qu.:130.0   3rd Qu.:3725
##  Max.   :38.0          Max.   :455.0   Max.   :225.0   Max.   :4906
##                                        NA's   :1
##       acc          model.year origin              model.name
##  Min.   : 8.50   75     :11   1:72   amc gremlin       : 3
##  1st Qu.:13.50   76     :11   2:16   ford pinto        : 3
##  Median :15.10   77     :11   3:17   amc hornet        : 2
##  Mean   :15.34   70     : 9          amc matador       : 2
##  3rd Qu.:16.80   74     : 9          chevrolet chevette: 2
##  Max.   :24.80   78     : 9          chevrolet vega    : 2
##                  (Other):45          (Other)           :91
```

De acuerdo con los resúmenes anteriores, hay algunas observaciones en el conjunto de datos de prueba cuyo rango de las variables disp y weight se extiende más allá del rango en el conjunto de datos de entrenamiento, así que vamos a eliminar esas observaciones del conjunto de datos de prueba.

```
autos.prueba <- autos.prueba[autos.prueba$weight >= 1649 & autos.prueba$disp >=
    70, ]

summary(autos.entrenamiento)
```

```
##       mpg        cyl          disp             hp             weight
##  Min.   : 9.00   3:  3   Min.   : 70.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.50   4:155   1st Qu.: 98.0   1st Qu.: 75.0   1st Qu.:2220
##  Median :23.00   5:  2   Median :144.0   Median : 92.0   Median :2774
##  Mean   :23.88   6: 63   Mean   :188.7   Mean   :103.4   Mean   :2943
##  3rd Qu.:29.80   8: 70   3rd Qu.:250.0   3rd Qu.:120.5   3rd Qu.:3465
##  Max.   :46.60           Max.   :455.0   Max.   :230.0   Max.   :5140
##                                          NA's   :5
##       acc          model.year  origin              model.name
##  Min.   : 8.00   73     : 32   1:177   chevrolet impala  :  4
##  1st Qu.:13.90   78     : 27   2: 54   ford maverick     :  4
##  Median :15.50   80     : 27   3: 62   toyota corolla    :  4
##  Mean   :15.65   76     : 23           amc matador       :  3
##  3rd Qu.:17.30   79     : 23           chevrolet citation:  3
##  Max.   :24.60   81     : 23           chevrolet nova    :  3
##                  (Other):138           (Other)           :272
```

```
summary(autos.prueba)
```

```
##       mpg        cyl          disp             hp          weight
##  Min.   :10.00   3: 1   Min.   : 71.0   Min.   : 52   Min.   :1649
##  1st Qu.:16.38   4:48   1st Qu.:111.0   1st Qu.: 79   1st Qu.:2275
##  Median :21.25   5: 1   Median :199.5   Median : 97   Median :2945
```

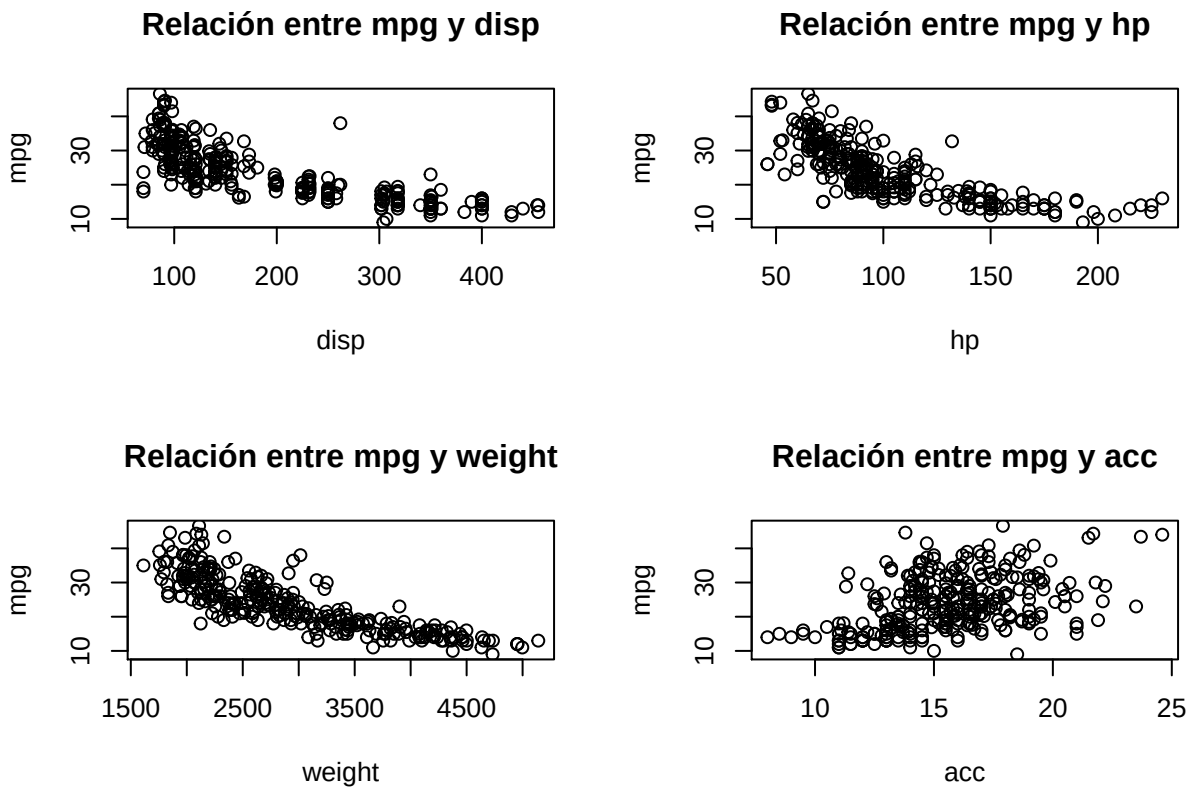
```
##  Mean   :22.43   6:21   Mean   :208.0   Mean   :108   Mean   :3059
##  3rd Qu.:27.40   8:33   3rd Qu.:302.5   3rd Qu.:130   3rd Qu.:3726
##  Max.   :38.00          Max.   :455.0   Max.   :225   Max.   :4906
##                                         NA's   :1
##       acc          model.year origin             model.name
##  Min.   : 8.50   75     :11   1:72   amc gremlin       : 3
##  1st Qu.:13.50   76     :11   2:15   ford pinto        : 3
##  Median :15.10   77     :11   3:17   amc hornet        : 2
##  Mean   :15.30   70     : 9          amc matador       : 2
##  3rd Qu.:16.65   74     : 9          chevrolet chevette: 2
##  Max.   :24.80   78     : 9          chevrolet vega    : 2
##                  (Other):44          (Other)           :90
```

En total, se eliminaron 2 observaciones.

Para trabajar con regresiones lineales, es importante trabajar sólo con variables cuantitativas y estudiar las relaciones que hay entre ellas. Con esto en mente, podemos comenzar nuestra exploración creando gráficos de dispersión para ver cuál es la relación entre nuestra variable de interés (mpg) y el resto de las variables cuantitativas:

```
par(mfrow = c(2, 2))  #crear una cuadrícula de 2 columnas y 2 hileras para ver cuatro gráficos.

plot(x = autos.entrenamiento$disp, y = autos.entrenamiento$mpg, main = "Relación entre mpg y disp",
    ylab = "mpg", xlab = "disp")
plot(x = autos.entrenamiento$hp, y = autos.entrenamiento$mpg, main = "Relación entre mpg y hp",
    ylab = "mpg", xlab = "hp")
plot(x = autos.entrenamiento$weight, y = autos.entrenamiento$mpg, main = "Relación entre mpg y weight",
    ylab = "mpg", xlab = "weight")
plot(x = autos.entrenamiento$acc, y = autos.entrenamiento$mpg, main = "Relación entre mpg y acc",
    ylab = "mpg", xlab = "acc")
```

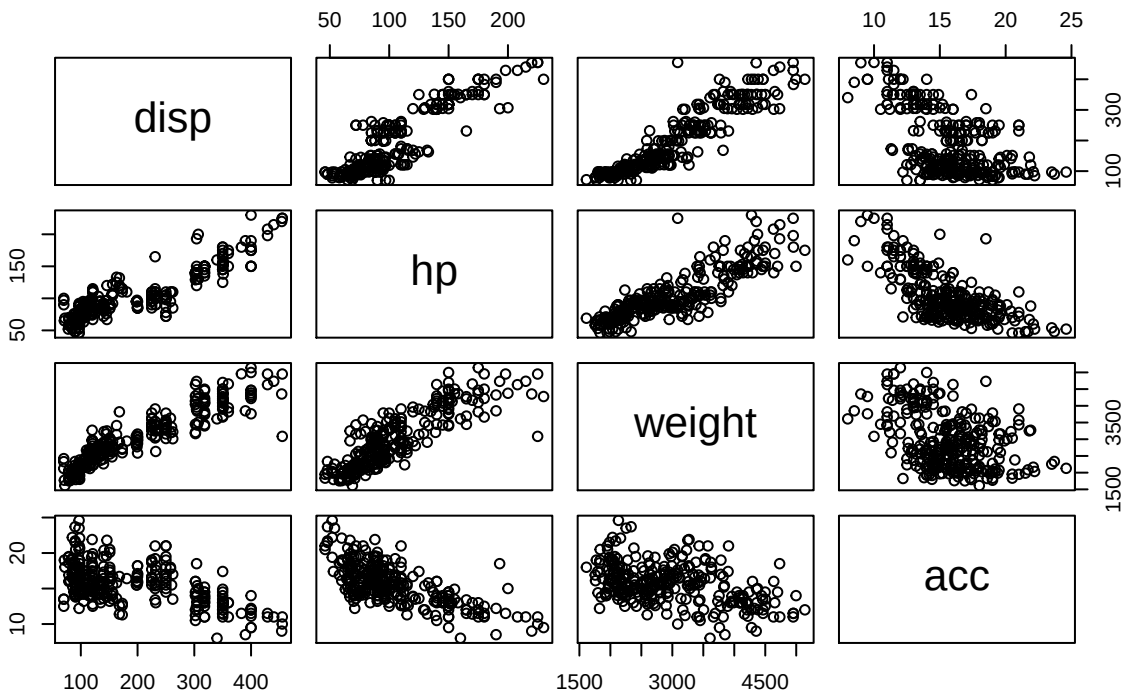


En los gráficos creados anteriormente, podemos ver como sí existe algún tipo de relación, aunque no sea exáctamente lineal, entre mpg y las otras cuatro variables cuantitativas. De estas cuatro variables, la que parece tener menor relación es la variable acc con la variable mpg.

También es importante visualizar la relación entre las diferentes variables predictoras, para lo cual podemos crear una matriz de gráficos de dispersión:

```
par(mfrow = c(1, 1))  #volver a solo un gráfico por visualización.

pairs(autos.entrenamiento[!is.na(autos$hp), c(3:6)], main = "Relación entre predictores")
```



La información del gráfico anterior podemos complementarla con una matriz de correlación:

```
cor(autos.entrenamiento[!is.na(autos.entrenamiento$hp), c(3:6)])
```

```
##              disp         hp     weight        acc
## disp    1.0000000  0.8970014  0.9294049 -0.5490654
## hp      0.8970014  1.0000000  0.8613336 -0.6944240
## weight  0.9294049  0.8613336  1.0000000 -0.4197444
## acc    -0.5490654 -0.6944240 -0.4197444  1.0000000
```

Como pudimos apreciar en la matriz de gráficos de dispersión, y confirmar con la matriz de correlación, hay una correlación significativa entre las variables hp y disp, disp y weight y hp y weight. La única variable que no tiene una correlación absoluta mayor a 0.75 es acc. Dado esto, debemos escoger una variable entre hp, disp y weight, para lo cual podemos crear una matriz de correlación entre esas tres variables y mpg para ver cuál tiene una correlación mayor con mpg:

```
cor(autos.entrenamiento[!is.na(autos.entrenamiento$hp), c(1, 3:5)])
```

```
##                mpg       disp         hp     weight
## mpg     1.0000000 -0.7933915 -0.7728878 -0.8228879
## disp   -0.7933915  1.0000000  0.8970014  0.9294049
## hp     -0.7728878  0.8970014  1.0000000  0.8613336
## weight -0.8228879  0.9294049  0.8613336  1.0000000
```

Basándonos en la correlación absoluta, se va a escoger la variable weight para ser incluida en el modelo.

Modelo de Minería de Datos

Una vez seleccionadas las variables para incluir en el modelo de regresión, se procede a crearlo:

```
reg.mpg <- lm(mpg ~ weight + acc, data = autos.entrenamiento)

summary(reg.mpg)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acc, data = autos.entrenamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5840  -2.8454  -0.4122   2.7258  15.8613
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.491419   2.289822   18.56   <2e-16 ***
## weight      -0.007592   0.000352  -21.57   <2e-16 ***
## acc          0.238284   0.104511    2.28   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.539 on 290 degrees of freedom
## Multiple R-squared:  0.6818, Adjusted R-squared:  0.6797
## F-statistic: 310.8 on 2 and 290 DF,  p-value: < 2.2e-16
```

En el resumen del modelo, podemos ver que ambas variables son significativas y que el modelo creado explica alrededor de un 69% de la variación en la variable de respuesta (mpg). Asimismo, podemos ver que el modelo es mejor que un modelo sin variables. Con este modelo, procedemos a hacer las predicciones sobre el conjunto de datos de prueba.

```
autos.prueba$Prediccion <- predict(reg.mpg, newdata = autos.prueba)
```

Evaluación

Para determinar qué tan bueno es el modelo, vamos a calcular dos métricas: primero la raíz cuadrada del promedio de los errores cuadrados (RMSE):

```
sqrt(mean((autos.prueba$mpg - autos.prueba$Prediccion)^2))
```

```
## [1] 3.605973
```

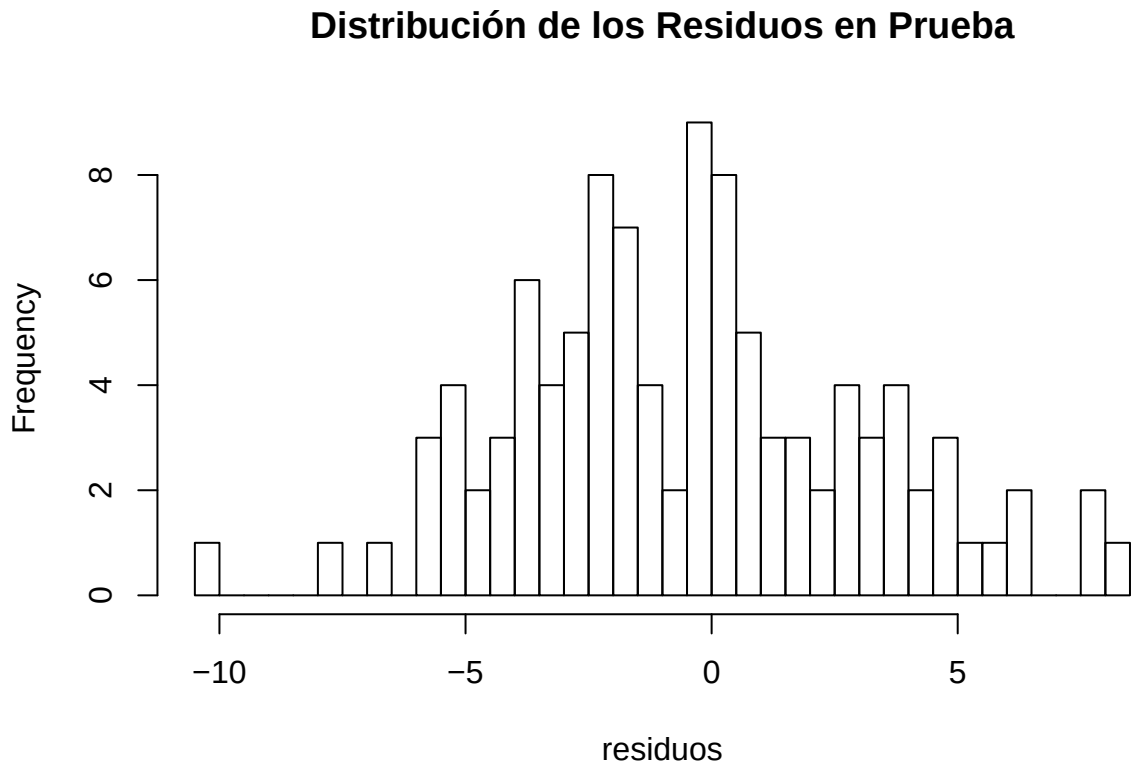
También es necesario calcular el r cuadrado:

```
Suma.Total.Cuadrados <- sum((mean(autos.entrenamiento$mpg) - autos.prueba$mpg)^2)  #error total si usam
Suma.Errores.Cuadrados <- sum((autos.prueba$Prediccion - autos.prueba$mpg)^2)  #error total de nuestro
1 - (Suma.Errores.Cuadrados/Suma.Total.Cuadrados)
```

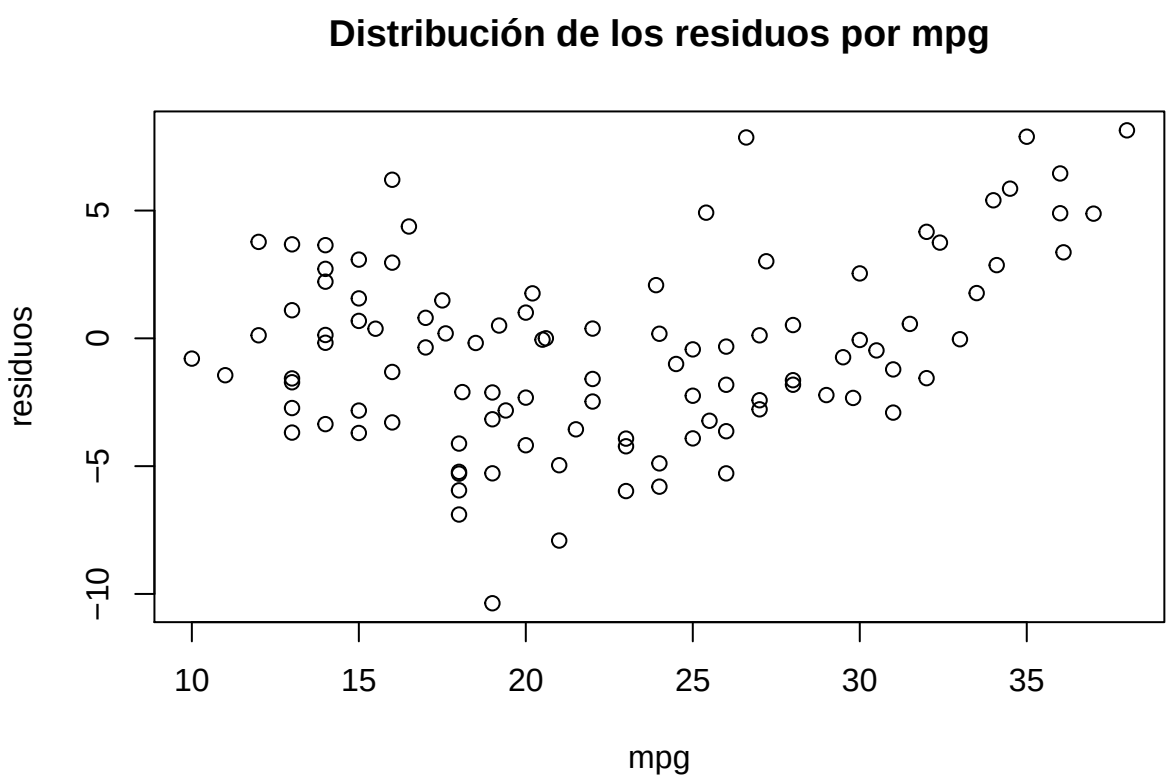
```
## [1] 0.7542167
```

Finalmente, procedemos a analizar la distribución de los residuos:

```
hist(autos.prueba$mpg - autos.prueba$Prediccion, breaks = 50, main = "Distribución de los Residuos en P
    xlab = "residuos")
```




```
plot(y = autos.prueba$mpg - autos.prueba$Prediccion, x = autos.prueba$mpg, main = "Distribución de los
    xlab = "mpg", ylab = "residuos")
```



Resultados

De acuerdo con la evaluación hecha, el modelo inicia con muy buenos números: puede explicar cerca de un 71% de la variación de la variable mpg en el conjunto de datos de prueba, y el error promedio es de alrededor de 3 mpg para arriba o para abajo. Sin embargo, el análisis de los residuos nos deja ver que hay un patrón no aleatorio para valores altos de mpg, específicamente para valores mayores a 30. A partir de este número, los residuos tienden al alza, lo cual indica que el modelo no es igual para diferentes valores de mpg.

En resumen: el modelo no debería ser utilizado, y se deben analizar más a fondo los datos para ver si se puede aplicar alguna transformación a los datos de entrada para volver a intentar crear un modelo