Práctica 1: Programación en R

Probabilidad y Estadística - Manuel Martínez Ramón

Introducción

El objetivo de esta práctica es realizar un estudio muy superficial de modelos de regresión simple. Para ello, necesitaremos un conjunto de datos (*dataset*) sobre el que hacer el estudio. En la primera parte de la práctica, cargaremos dicho *dataset* y realizaremos una exploración estadística de ellos. En la segunda parte, realizaremos un estudio de regresión más específico.

Información del *dataset*

El dataset elegido es "*Boston Housing*". Consiste en un *dataset* público con información de viviendas en Boston. En concreto, el *dataset* contine las siguientes columnas:

- **CRIM**: Per capita crime rate by town
- **ZN**: Proportion of residential land zoned for lots over 25,000 sq. ft
- **INDUS**: Proportion of non-retail business acres per town
- **CHAS**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- **NOX**: Nitric oxide concentration (parts per 10 million)
- **RM**: Average number of rooms per dwelling
- **AGE**: Proportion of owner-occupied units built prior to 1940
- **DIS**: Weighted distances to five Boston employment centers
- **RAD**: Index of accessibility to radial highways
- **TAX**: Full-value property tax rate per $10,000
- **PTRATIO**: Pupil-teacher ratio by town
- **B**: 1000(Bk - 0.63)?, where Bk is the proportion of [people of African American descent] by town
- **LSTAT**: Percentage of lower status of the population
- **MEDV**: Median value of owner-occupied homes in $1000s"'

El *dataset* lo cargaremos en R mediante la instalación de un paquete y su posterior importación:

```
require("mlbench")
```

```
## Cargando paquete requerido: mlbench
```

```
library(mlbench)
```

Ya cargado en R, procedemos a guardar el *dataset* en la variable `housing` para que realicéis el resto de la práctica sobre ella:

```
data("BostonHousing")
housing <- BostonHousing
```

Información de la práctica.

- La práctica se calificará sobre 10 puntos.
- La práctica se resuelve sobre este mismo código.
- Antes de entregarlo, habrá que cambiar el nombre del fichero, sustituyendo *nombre1* y *apellido1* por los propios del alumno.
- Si el alumno considera relevante la instalación de alguna librería extra que pueda mostrar mejores resultados o gráficos, es libre para hacerlo. Puntuará más si se realiza un trabajo óptimo en esta parte.
- Sobre todo en la parte final (regresión), puntuará más aquellos comentarios del alumno que muestren haber investigado el significado de todos los análisis realizados.
- Si el alumno se encuentra con errores durante la ejecución del código, tiene que aprender a lidiar con ellos como futuro ingeniero informático.
- Es recomendable consultar los diferentes comandos, así como nuevos comendos y librerías, através de google, el cual os llevará a paginas web adecuadas de programación en R.

1) Análisis exploratorio inicial:

Se pide utilizar los comandos `str`, `head`, `dim`, `summary` sobre `housing` para explorar distribución inicial de los datos en el *dataset*.

```
# Estructura del dataset
str(housing)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : num  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ b      : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
# Primeras filas del dataset
head(housing)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio      b lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
```

```
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```
# Dimensiones del dataset
dim(housing)
```

```
## [1] 506  14
```

```
# Resumen estadístico
summary(housing)
```

```
##       crim                zn             indus       chas         nox
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   0:471   Min.   :0.3850
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1: 35   1st Qu.:0.4490
##  Median : 0.25651   Median :  0.00   Median : 9.69           Median :0.5380
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14           Mean   :0.5547
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10           3rd Qu.:0.6240
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74           Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000
##  1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000
##  Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000
##  Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549
##  3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000
##       tax           ptratio            b               lstat
##  Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73
##  1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95
##  Median :330.0   Median :19.05   Median :391.44   Median :11.36
##  Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95
##  Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97
##       medv
##  Min.   : 5.00
##  1st Qu.:17.02
##  Median :21.20
##  Mean   :22.53
##  3rd Qu.:25.00
##  Max.   :50.00
```

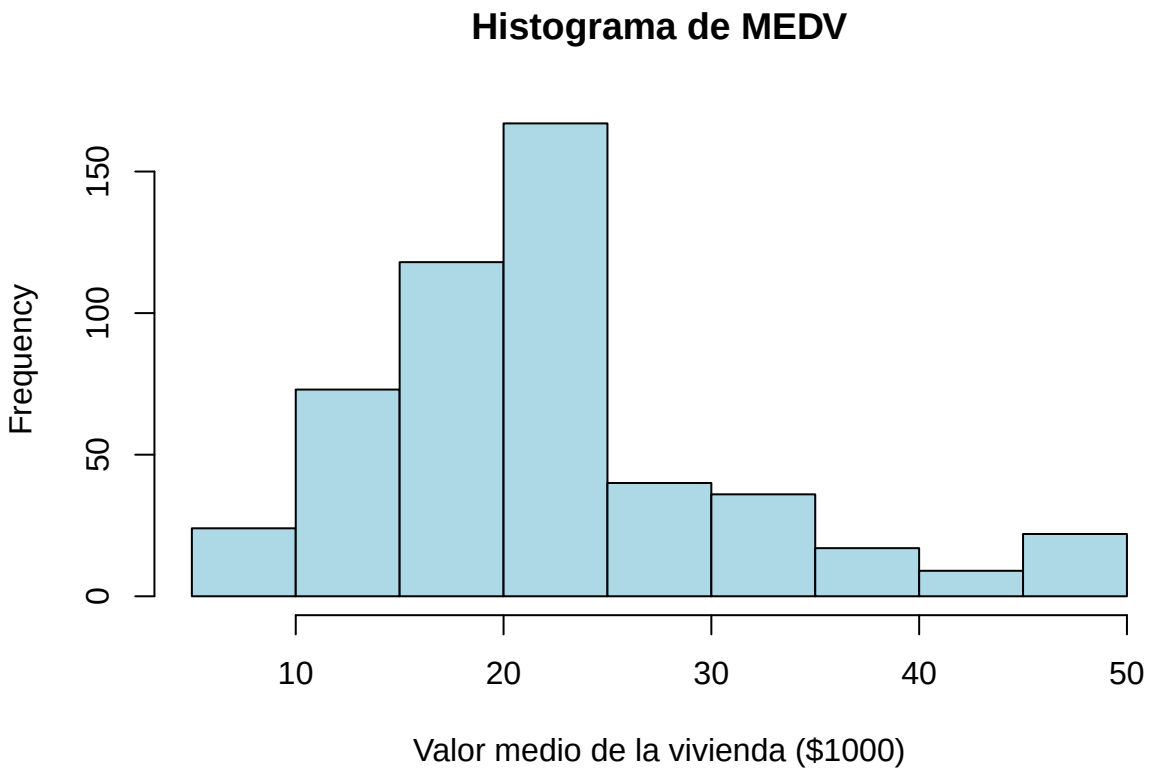
Comentarios del alumno (máximo 100 palabras):

```
# El comando 'str(housing)' muestra la estructura del dataset, indicando el tipo de cada
# variable y las primeras observaciones. 'head(housing)' permite visualizar las primeras seis
# filas para obtener una vista rápida de los datos. 'dim(housing)' proporciona las dimensiones
# del dataset, es decir, el número de filas y columnas. 'summary(housing)' ofrece un resumen
# estadístico de las variables numéricas, como el mínimo, máximo, media, mediana y cuartiles,
# lo que permite una primera aproximación a la distribución de los datos y a posibles valores
# atípicos o sesgos.
```

2) Análisis exploratorio de la variable objetivo:

En esta práctica, trataremos de predecir el valor del precio medio de la vivienda MEDV. Para ello, primero exploraremos la distribución de sus datos. Se pide dibujar un histograma, calcular asimetría y apuntamiento de MEDV. Se pide dibujar un boxplot, calcular los cuartiles y los percentiles 10-90 sobre MEDV. Se pide describir los elementos más importantes de ambas gráficas.

```
# Cargar el paquete necesario para los análisis
library(mlbench)

# Cargar el dataset
data("BostonHousing")
housing <- BostonHousing

# Histograma de la variable MEDV
hist(housing$medv, main="Histograma de MEDV", xlab="Valor medio de la vivienda ($1000)", col="lightblue
```




```
# Calcular asimetría y apuntamiento
if (!require("e1071")) install.packages("e1071", repos="http://cran.us.r-project.org")
```

```
## Cargando paquete requerido: e1071
```

```
## Warning: package 'e1071' was built under R version 4.4.2
```

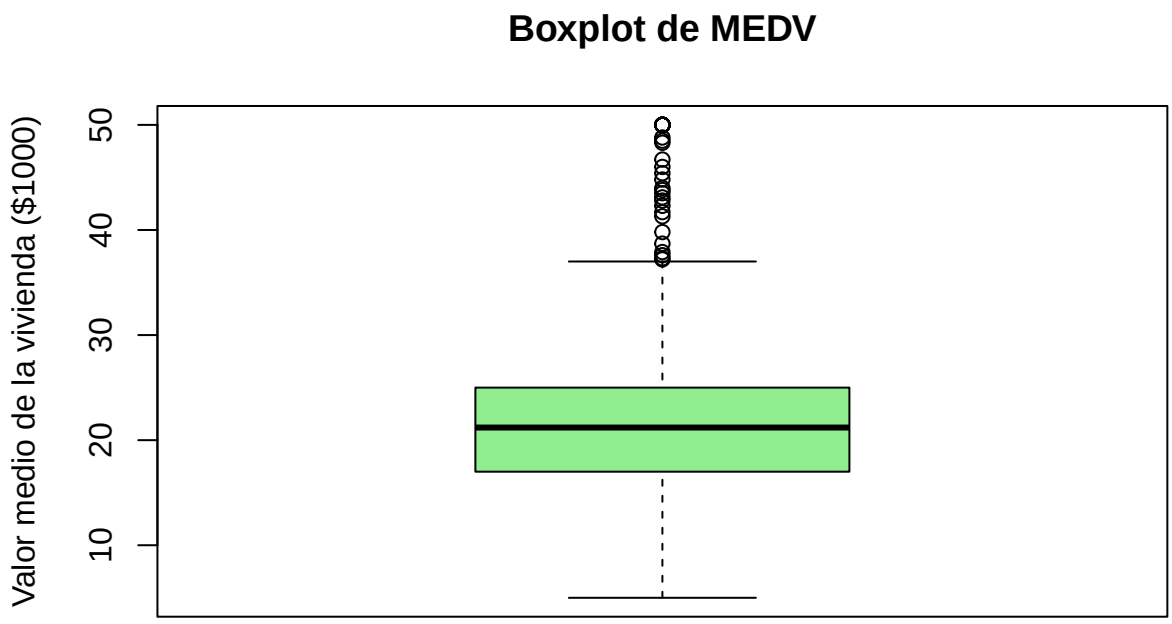
```r
library(e1071)
asimetria <- skewness(housing$medv)
apuntamiento <- kurtosis(housing$medv)
cat("Asimetría:", asimetria, "\n")
```

```
## Asimetría: 1.101537
```

```r
cat("Apuntamiento:", apuntamiento, "\n")
```

```
## Apuntamiento: 1.450984
```

```r
# Boxplot de MEDV
boxplot(housing$medv, main="Boxplot de MEDV", ylab="Valor medio de la vivienda ($1000)", col="lightgreen")
```




```r
# Calcular los cuartiles y percentiles 10-90
cuartiles <- quantile(housing$medv)
percentiles_10_90 <- quantile(housing$medv, probs=c(0.10, 0.90))

# Formatear y mostrar los cuartiles de forma clara
cat("Cuartiles de MEDV:\n")
```

```
## Cuartiles de MEDV:
```

```r
print(format(cuartiles, nsmall = 2, justify = "right"), quote = FALSE)
```

```
##     0%    25%    50%    75%   100%
##  5.000 17.025 21.200 25.000 50.000
```

```r
# Formatear y mostrar los percentiles 10-90 de forma clara
cat("Percentiles 10-90 de MEDV:\n")
```

```
## Percentiles 10-90 de MEDV:
```

```r
print(format(percentiles_10_90, nsmall = 2, justify = "right"), quote = FALSE)
```

```
##   10%   90%
## 12.75 34.80
```

Comentarios del alumno (máximo 100 palabras):

```r
# Para generar el PDF desde el archivo .Rmd fue necesario configurar un mirror de CRAN,
# ya que sin esa línea solo se ejecutaba correctamente en RStudio.

# El código realiza un análisis exploratorio de 'medv' (valor medio de viviendas),
# con un histograma para su distribución, asimetría y apuntamiento para evaluar forma,
# un boxplot para dispersión y cuartiles, y los percentiles 10-90 para ver los rangos.
```

3) Correlación entre variables:

Lo primerio es que hay que tener en cuenta solo las variables cuantitativas. Se recomienda calcular la matriz de correlaciones de las variables cuantitativas con el comando `cor`.Además se pide utilizar el comando `corrplot` de la libreria `corrplot`, y la librería `RColorBrewer` (que posiblemente tengais que instalar) para mostrar graficamente las correlaciones entre todas las variables cuantitativas. Se pide describir los elementos de la gráfica que aportan mayor información.

```r
# Instalar los paquetes si no están instalados
install.packages("corrplot")
```

```
## Installing package into 'C:/Users/manue/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'corrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\manue\AppData\Local\Temp\RtmpYV8U3c\downloaded_packages
```

```r
install.packages("RColorBrewer")
```

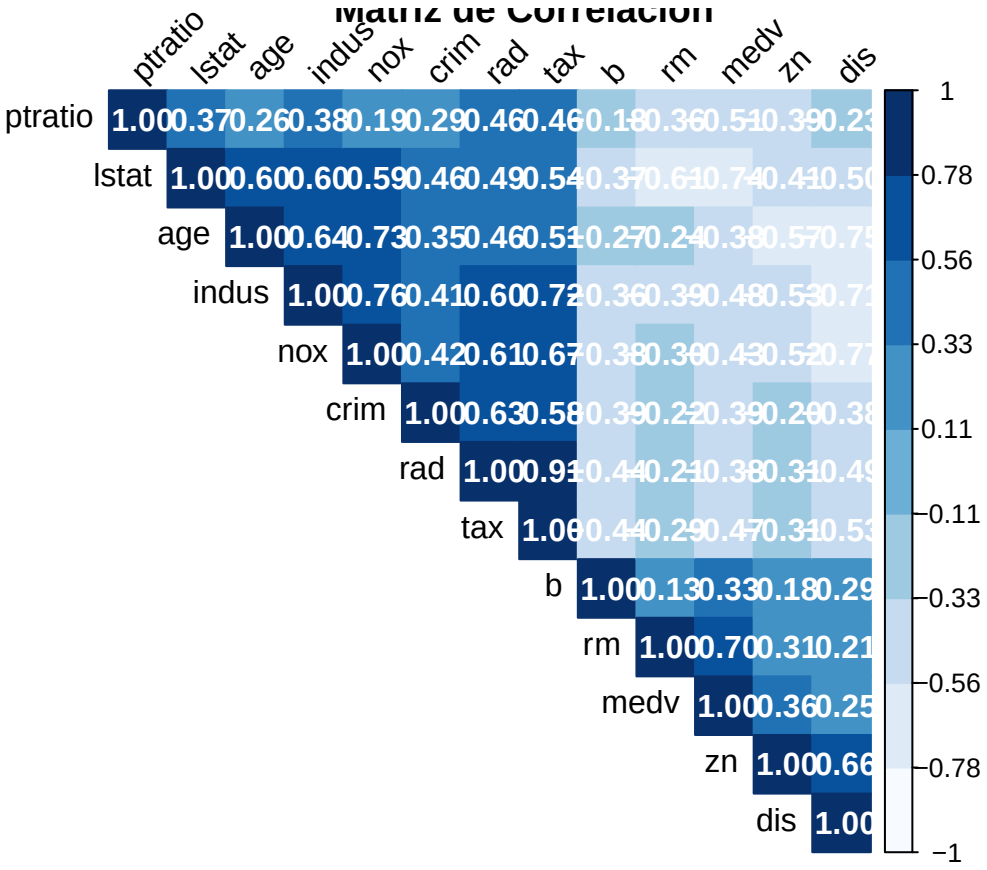
```
## Installing package into 'C:/Users/manue/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'RColorBrewer' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\manue\AppData\Local\Temp\RtmpYV8U3c\downloaded_packages
```

```r
# Cargar las librerías necesarias
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.2

## corrplot 0.95 loaded
```

```r
library(RColorBrewer)

# Filtrar solo las variables cuantitativas
cuantitativas <- housing[, sapply(housing, is.numeric)]

# Calcular la matriz de correlación
correlation_matrix <- cor(cuantitativas)

# Mostrar la matriz de correlación de forma visual
corrplot(correlation_matrix, method="color", col=brewer.pal(n=9, name="Blues"),
         type="upper", order="hclust", addCoef.col="white",
         tl.col="black", tl.srt=45, title="Matriz de Correlación")
```



Comentarios del alumno (máximo 100 palabras):

```
# Este código calcula y visualiza la matriz de correlación de las variables
# cuantitativas del dataset "BostonHousing". Se seleccionan las variables numéricas
# y se calcula su correlación. La visualización usa 'corrplot' con la paleta
# "Blues", donde los colores más oscuros indican correlaciones más fuertes.
# Los coeficientes numéricos dentro de las celdas muestran el grado exacto de
# la correlación entre las variables. El gráfico facilita la identificación de
# dependencias entre variables, lo cual puede ser relevante para la construcción
# de modelos predictivos.
```

4) Regresiones lineales simples:

Se pide Regresiones lineales simples:Se pide escoger cuatro variables independientes (a vuestro juicio, las mejores) y realizar cuatro regresiones simples con cada una de ellas sobre la variable dependiente MEDV. Se pide dibujar los *scatterplots* de cada variable independiente con MEDV y la recta de regresión resultante sobre cada *scatterplot*, para ello habrá que instalar la libreria `ggplot2`. Describir brevemente el resultado de este análisis.

```
# Instalar y cargar la librería ggplot2 si no está instalada
install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/manue/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

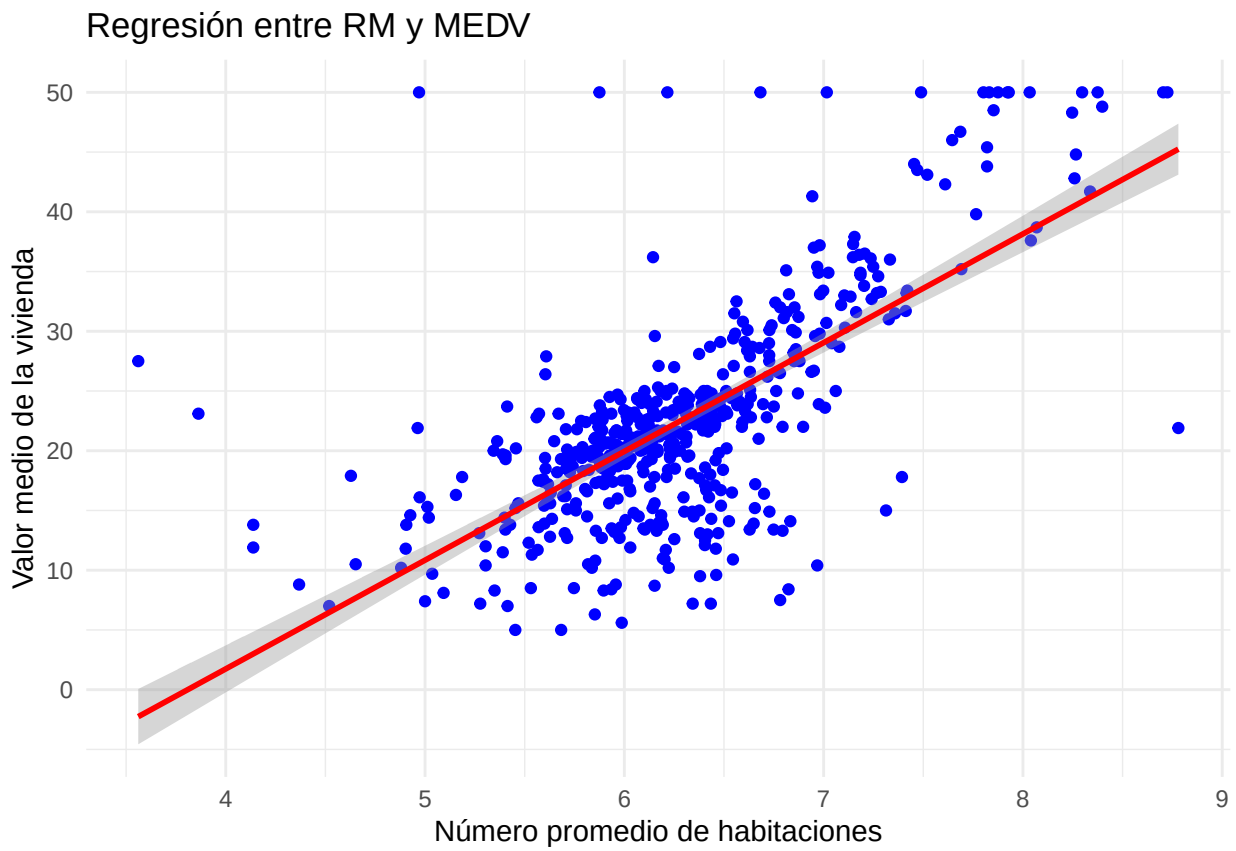
```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\manue\AppData\Local\Temp\RtmpYV8U3c\downloaded_packages
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

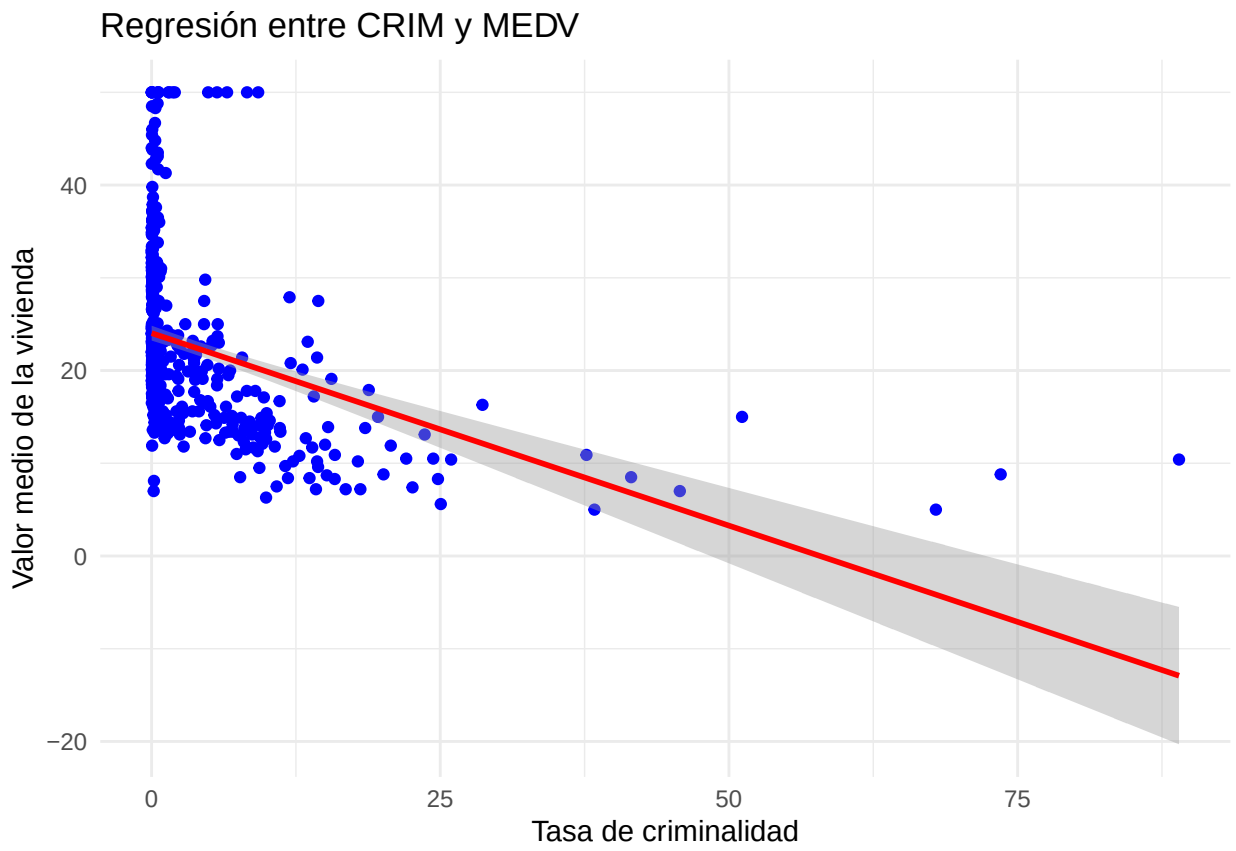
```
# Seleccionar las variables independientes
variables_independientes <- c("rm", "crim", "nox", "age")

# Crear un gráfico de cada variable independiente contra MEDV con la recta de regresión

# 1. Regresión entre 'rm' y 'medv'
ggplot(housing, aes(x=rm, y=medv)) +
  geom_point(color="blue") +  # Scatterplot
  geom_smooth(method="lm", color="red") +  # Recta de regresión
  labs(title="Regresión entre RM y MEDV", x="Número promedio de habitaciones", y="Valor medio de la vivi
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



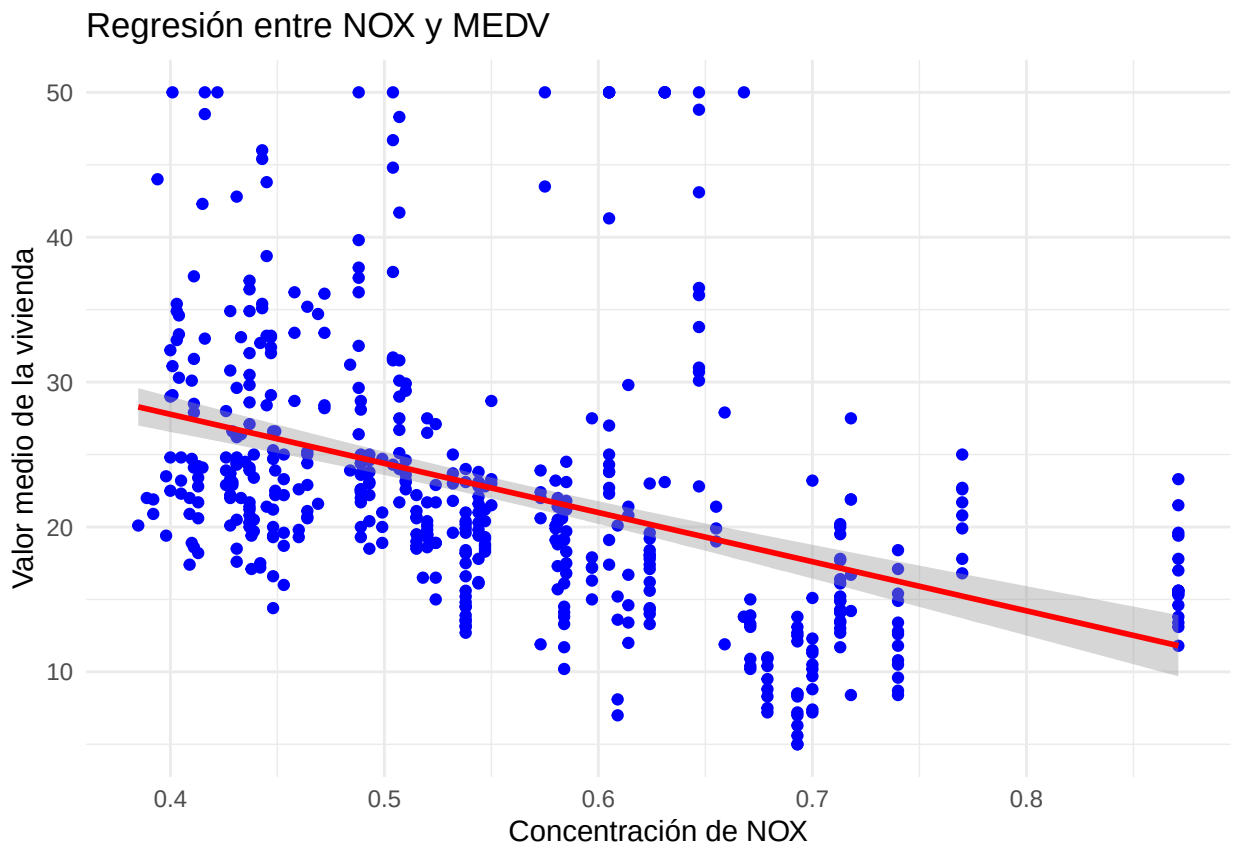

```
# 2. Regresión entre 'crim' y 'medv'
ggplot(housing, aes(x=crim, y=medv)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  labs(title="Regresión entre CRIM y MEDV", x="Tasa de criminalidad", y="Valor medio de la vivienda") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



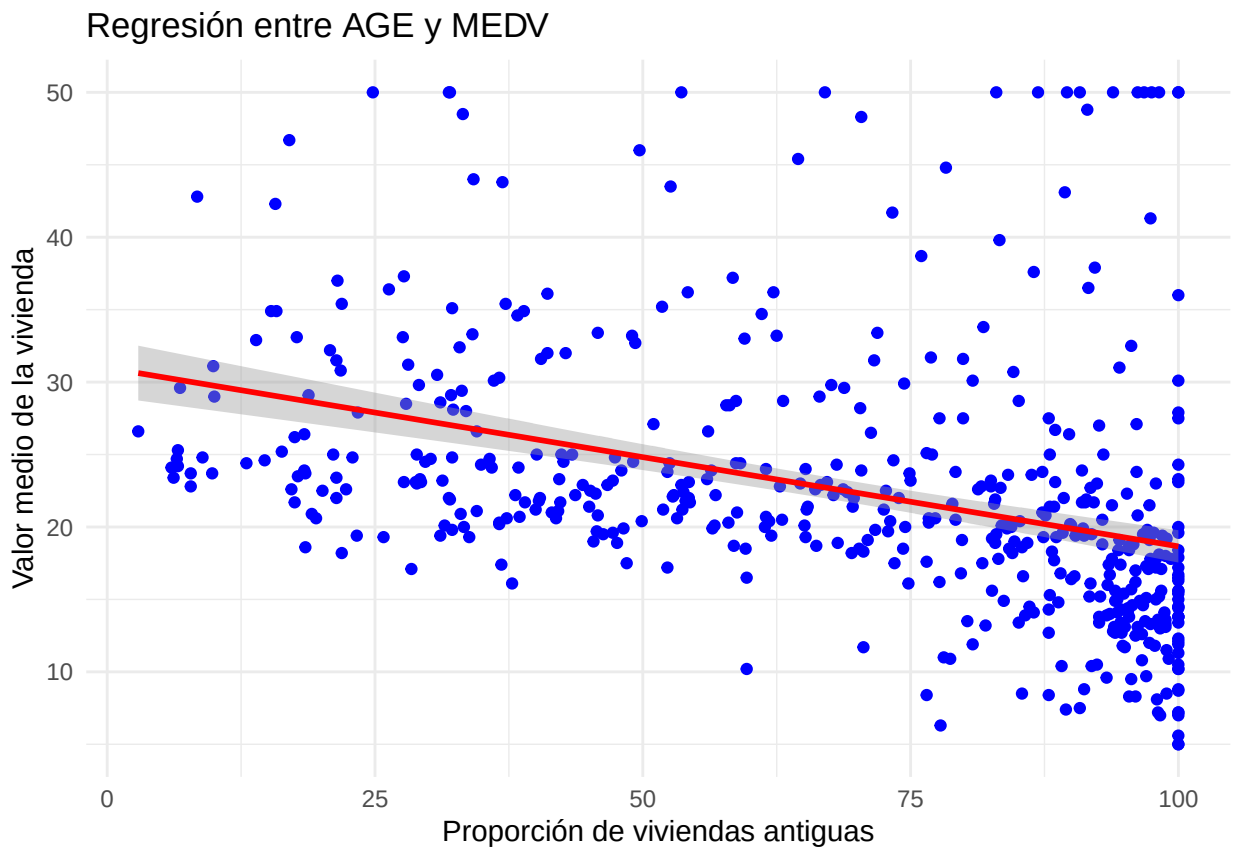

```
# 3. Regresión entre 'nox' y 'medv'
ggplot(housing, aes(x=nox, y=medv)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  labs(title="Regresión entre NOX y MEDV", x="Concentración de NOX", y="Valor medio de la vivienda") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# 4. Regresión entre 'age' y 'medv'
ggplot(housing, aes(x=age, y=medv)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  labs(title="Regresión entre AGE y MEDV", x="Proporción de viviendas antiguas", y="Valor medio de la v
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Comentarios del alumno (máximo 100 palabras):

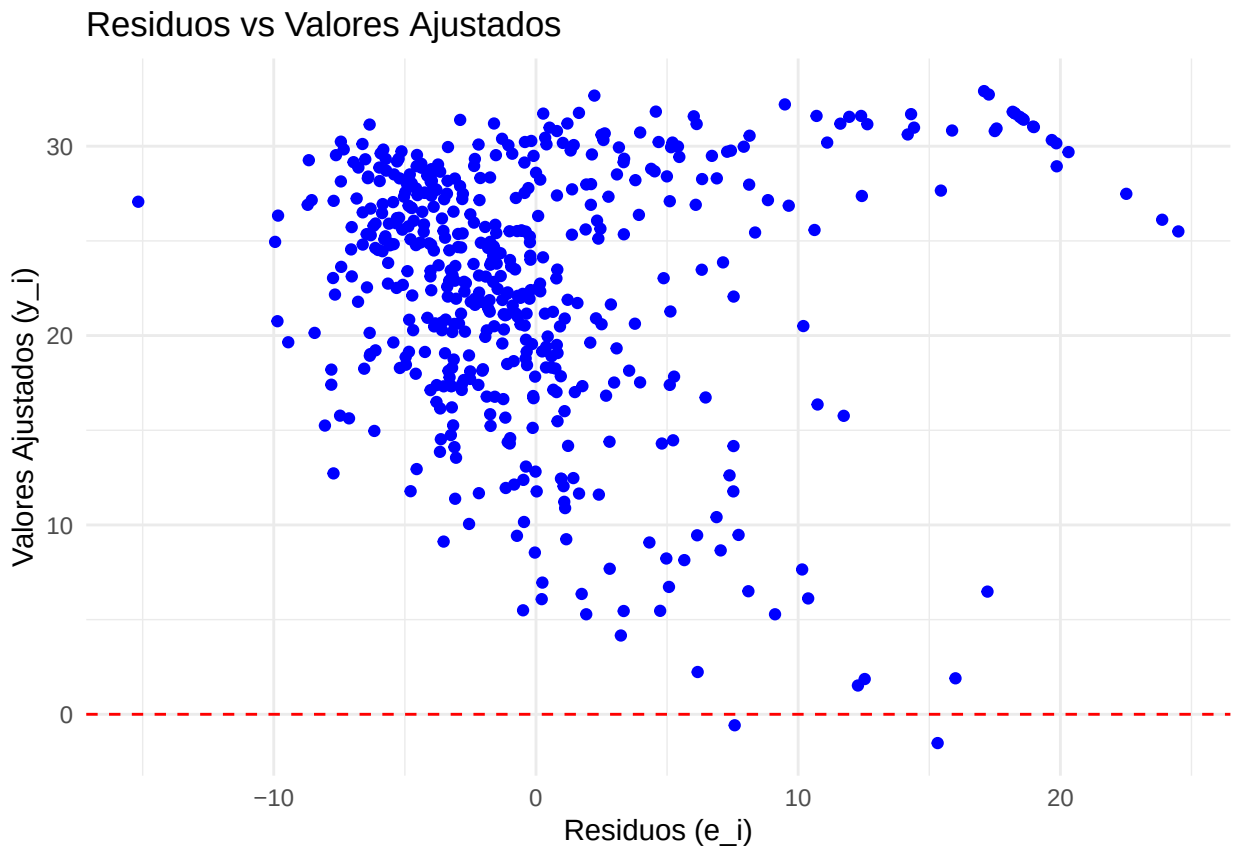
```
# Este código genera gráficos de regresión lineales simples entre la variable dependiente 'medv'
# (valor medio de la vivienda) y cuatro variables independientes: 'rm' (promedio de habitaciones),
# 'crim' (tasa de criminalidad), 'nox' (concentración de NOX) y 'age' (proporción de viviendas
# antiguas). Se espera que 'rm' tenga una correlación positiva con 'medv', mientras que 'crim',
# 'nox' y 'age' suelen estar negativamente correlacionados con el valor de las viviendas.
# Cada gráfico incluye un scatterplot y una recta de regresión, lo que permite visualizar
# la relación entre las variables y ayudar en el análisis de las tendencias.
```

5) Análisis de los residuos:

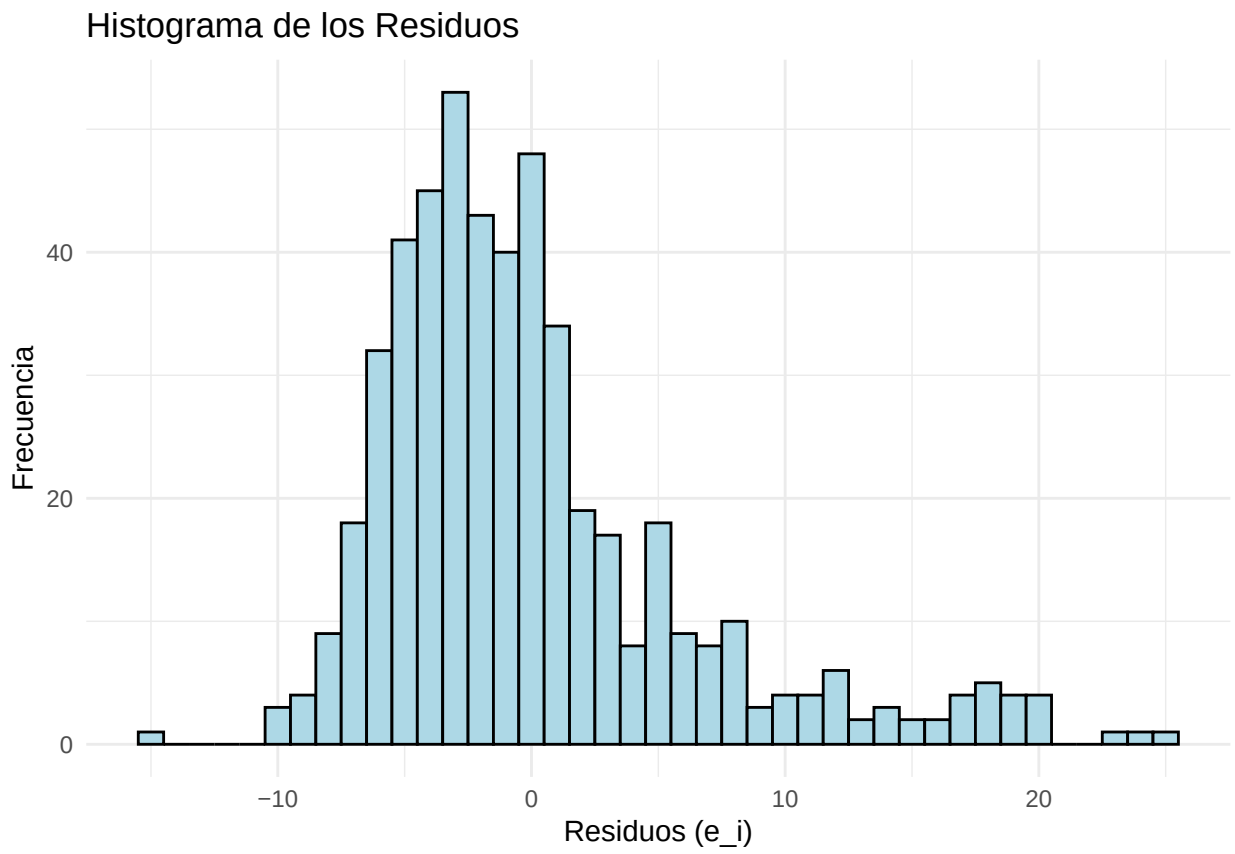
Se pide mostrar en un *scatterplot* los residuos e_i (eje-x) y la predicción que hace la recta de regresión de la variable independiente LSTAT respecto a la variable MEDV con cada punto $\hat{y}_i$ (eje-y), también llamada variable ajustada. Se pide realizar un histograma de los residuos exclusivamente. Se pide investigar el significado y la importancia de este gráfico y comentarlo brevemente.

```
# Instalar y cargar ggplot2 si no está disponible
if (!require("ggplot2")) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
library(ggplot2)

# Realizar la regresión lineal entre LSTAT y MEDV
modelo <- lm(medv ~ lstat, data = housing)

# Calcular los residuos e_i y los valores ajustados (predicciones)
```

```r
residuos <- residuals(modelo)
valores_ajustados <- fitted(modelo)

# Crear un data frame para ggplot con los residuos y los valores ajustados
data_residuos <- data.frame(residuos = residuos, valores_ajustados = valores_ajustados)

# 1. Scatterplot de los residuos vs. valores ajustados
ggplot(data = data_residuos, aes(x = residuos, y = valores_ajustados)) +
  geom_point(color = "blue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuos vs Valores Ajustados", x = "Residuos (e_i)", y = "Valores Ajustados (ŷ_i)") +
  theme_minimal()
```




```r
# 2. Histograma de los residuos
ggplot(data = data_residuos, aes(x = residuos)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black") +
  labs(title = "Histograma de los Residuos", x = "Residuos (e_i)", y = "Frecuencia") +
  theme_minimal()
```



Comentarios del alumno (máximo 100 palabras):

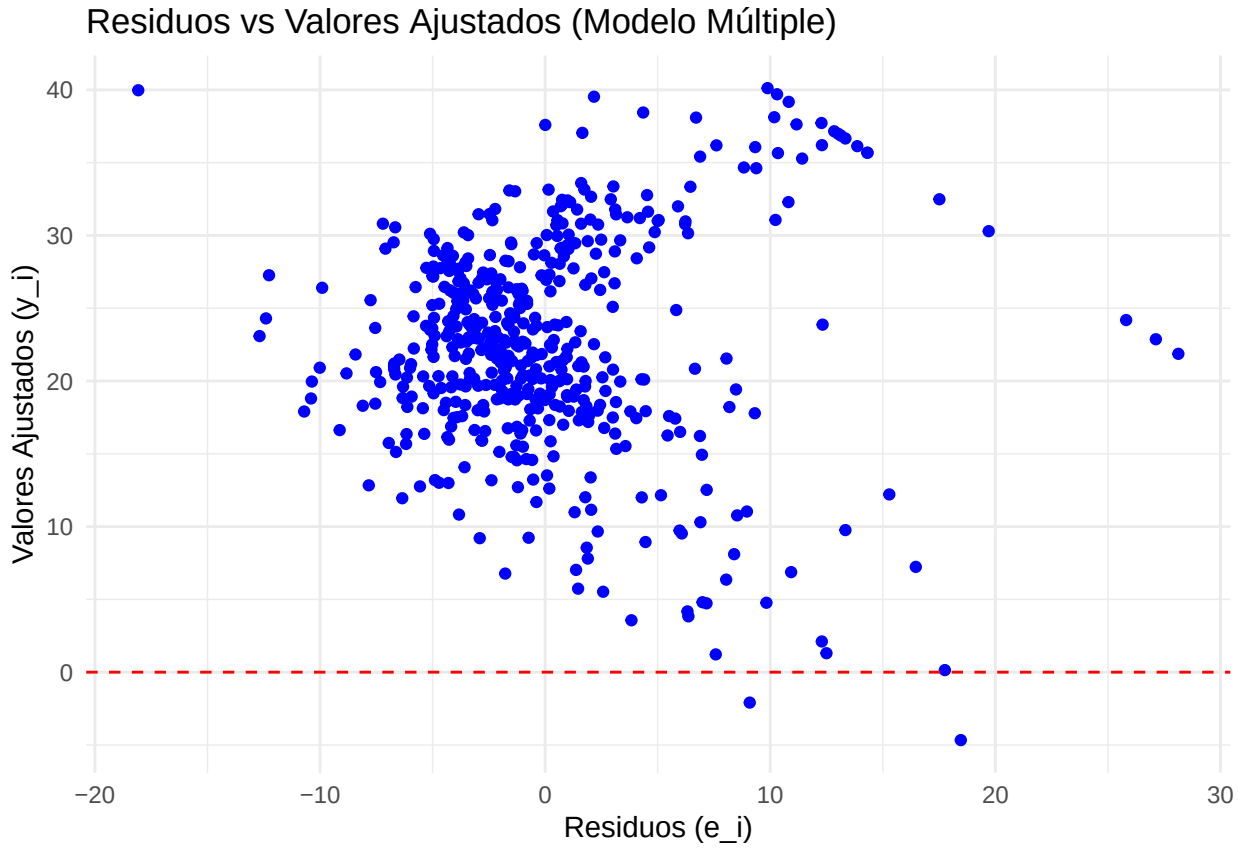
```
# Este código realiza un análisis de los residuos del modelo de regresión lineal entre
# 'LSTAT' (porcentaje de población de bajo estatus) y 'MEDV' (valor medio de la vivienda).
# Se calculan los residuos (diferencia entre valores observados y predichos) y los valores
# ajustados del modelo. Luego, se crean dos gráficos:
# - Un scatterplot de los residuos frente a los valores ajustados para identificar patrones
#   en los residuos que podrían indicar problemas de ajuste.
# - Un histograma de los residuos para verificar su distribución y confirmar si se asemeja
#   a una distribución normal, como se requiere en la regresión lineal.
```

6) Regresión lineal múltiple:

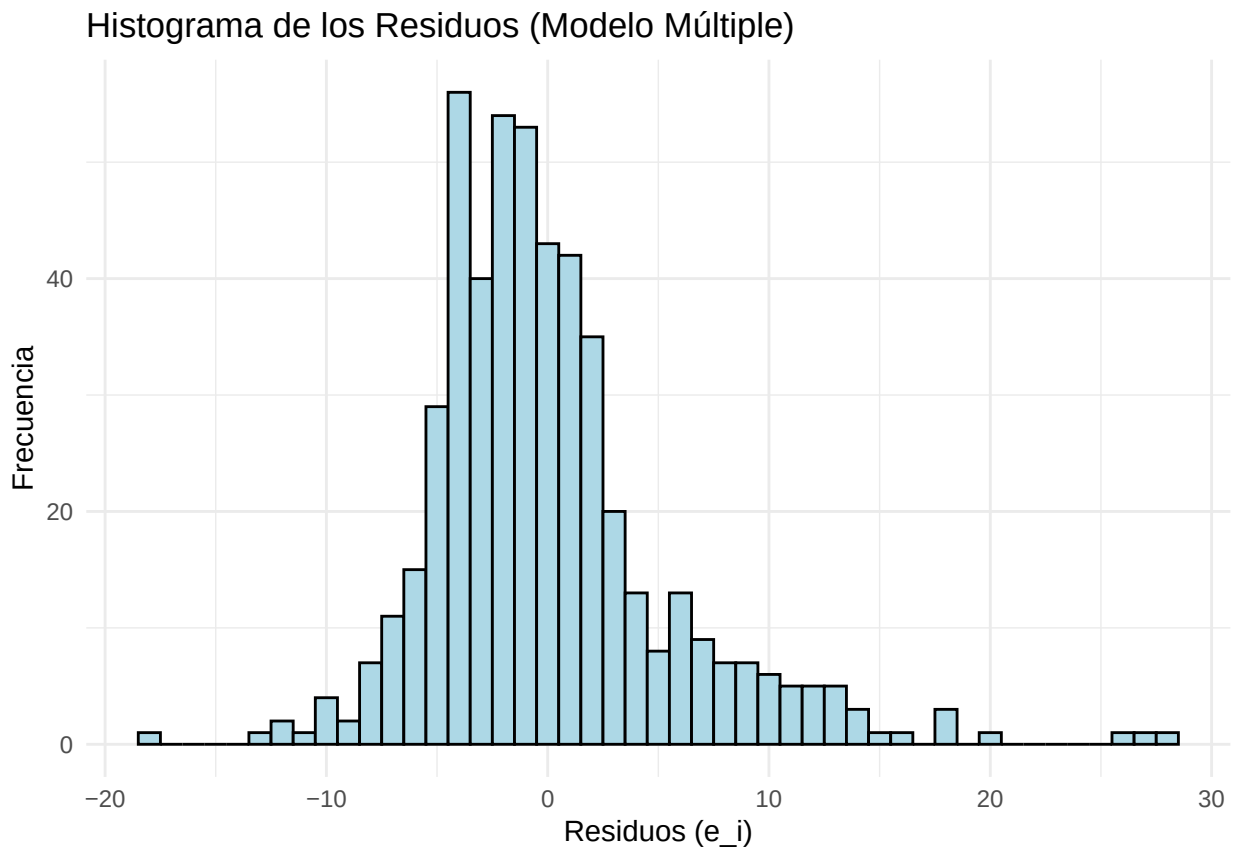
Se pide realizar una regresión lineal múltiple con las dos variables independientes a la vez sobre la variable MEDV. Se pide, además, realizar un análisis de los residuos, similar al realizado en el apartado anterior.

```
# Realizar la regresión lineal múltiple entre MEDV y las variables independientes LSTAT y RM
modelo_multiple <- lm(medv ~ lstat + rm, data = housing)

# Calcular los residuos y los valores ajustados
residuos_multiple <- residuals(modelo_multiple)
valores_ajustados_multiple <- fitted(modelo_multiple)

# Crear un data frame para ggplot con los residuos y los valores ajustados
data_residuos_multiple <- data.frame(residuos = residuos_multiple, valores_ajustados = valores_ajustados
```

```r
# Scatterplot de los residuos vs valores ajustados
library(ggplot2)
ggplot(data = data_residuos_multiple, aes(x = residuos, y = valores_ajustados)) +
  geom_point(color = "blue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuos vs Valores Ajustados (Modelo Múltiple)", x = "Residuos (e_i)", y = "Valores Aj
  theme_minimal()
```




```r
# Histograma de los residuos
ggplot(data = data_residuos_multiple, aes(x = residuos)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black") +
  labs(title = "Histograma de los Residuos (Modelo Múltiple)", x = "Residuos (e_i)", y = "Frecuencia") +
  theme_minimal()
```



Comentarios del alumno (máximo 150 palabras):

```
# Realizo una regresión lineal múltiple entre 'MEDV' (valor medio de la vivienda) y dos
# variables independientes: 'LSTAT' (porcentaje de población de bajo estatus) y 'RM'
# (promedio de habitaciones por vivienda). Se espera que un mayor número de habitaciones
# (RM) esté relacionado con precios más altos, mientras que un mayor porcentaje de población
# de bajo estatus (LSTAT) se asocie con precios más bajos. Luego, se calculan los residuos
# del modelo (diferencia entre valores observados y predichos) y se crean dos gráficos para
# evaluar el ajuste del modelo: un scatterplot de los residuos frente a los valores ajustados
# para verificar la aleatoriedad de los residuos. Un histograma para comprobar si los residuos
# siguen una distribución normal, un supuesto clave en la regresión lineal.
```