Tarea: Modelos de regresión lineal simple y múltiple

Aprendizaje Máquina (I). Máster en Ciencia de Datos - UV

Adrián Carrasco Alcalá y Clara Montalvá Barcenilla

Curso 2025-2026

En la librería `MASS` puedes encontrar un famoso banco de datos llamado `Boston` que contiene información sobre 506 barrios de Boston, Massachusetts, en 1970. La base de datos contiene 14 variables relativas a 506 barrios. Para saber qué información está contenida en las variables puedes escribir `?Boston` (después de haber cargado la librería `MASS`).

En esta tarea trabajaremos con el conjunto de datos "boston.xlsx" que encontraréis en el aula virtual (con 199 datos y 11 variables).

Ejercicio 1: Considera la variable respuesta `crim` relacionándola con la variable `X` con la que tenga mayor relación lineal.

```
lm_crim_global <- lm(crim ~. , data=boston)
summary(lm_crim_global)
```

```
##
## Call:
## lm(formula = crim ~ ., data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2246 -1.3862 -0.4447  1.1041 14.9000
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.750915   4.667657  -4.446 1.50e-05 ***
## indus         0.154381   0.066057   2.337 0.020487 *
## chas         -2.213542   0.596883  -3.709 0.000274 ***
## nox           1.632409   3.631098   0.450 0.653543
## rm            2.147244   0.435016   4.936 1.75e-06 ***
## age          -0.018398   0.012519  -1.470 0.143355
## dis          -0.410022   0.195131  -2.101 0.036950 *
## ptratio       0.477426   0.116245   4.107 5.97e-05 ***
## black         0.001829   0.006245   0.293 0.769941
## lstat         0.338194   0.054297   6.229 3.01e-09 ***
## medv         -0.111799   0.047486  -2.354 0.019586 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.851 on 188 degrees of freedom
## Multiple R-squared:  0.6035, Adjusted R-squared:  0.5824
## F-statistic: 28.61 on 10 and 188 DF,  p-value: < 2.2e-16
```

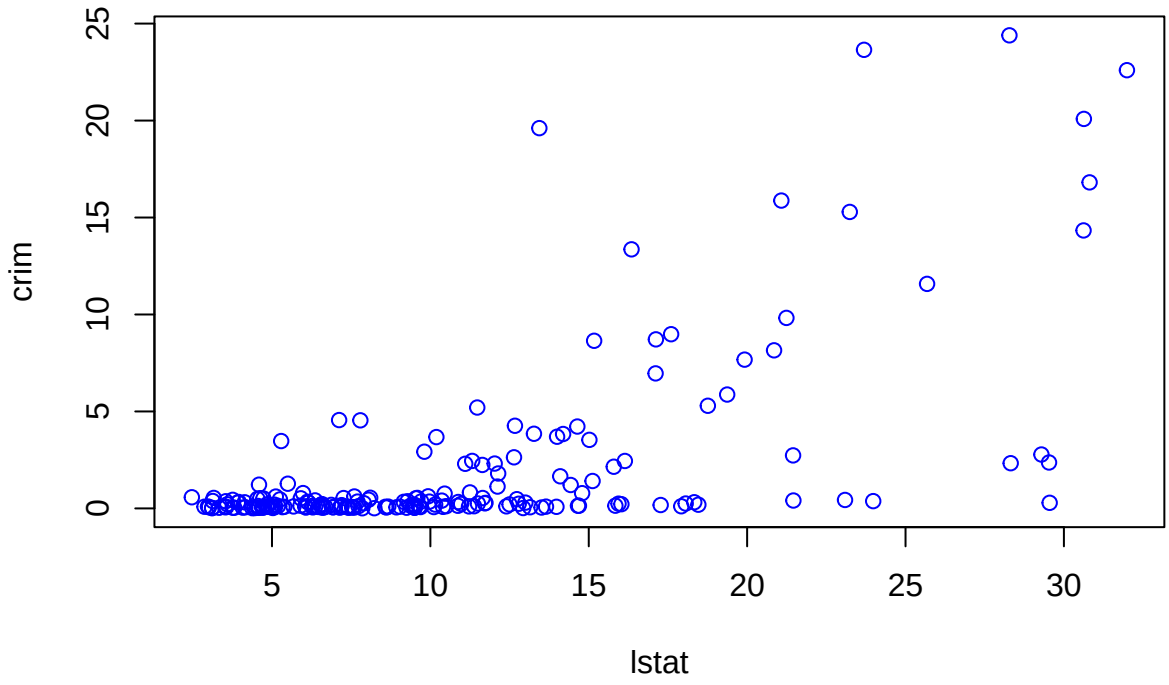
Observamos que la variable más significativa (la de p-valor más pequeño) es `lstat`, por lo que deducimos que es esta la variable con la que `crim` tiene mayor relación lineal.

1. Evalúa el efecto de `X` sobre `crim`, gráficamente y numéricamente. Es decir, indica como es la relación (fuerza y tipo).

```
lm_crim_lstat <- lm(crim ~ lstat, data=boston)
summary(lm_crim_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7449 -1.4171 -0.1196  1.0338 16.4752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.62305    0.45836  -5.723 3.86e-08 ***
## lstat        0.42834    0.03673  11.663  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.401 on 197 degrees of freedom
## Multiple R-squared:  0.4085, Adjusted R-squared:  0.4055
## F-statistic:   136 on 1 and 197 DF,  p-value: < 2.2e-16
```

```
plot(boston$lstat, boston$crim, col='BLUE', xlab = 'lstat', ylab = 'crim')
```



Cuando `lstat` aumenta en una unidad, `crim` aumenta en 0.4283435. Vemos que pese a que la variable sea muy significativa, en el gráfico de dispersión se observa una relación de fuerza baja o moderada, ya que los puntos no se agrupan alrededor de ninguna recta y se identifica un aumento en la dispersión de `crim` a medida que `lstat` incrementa. Este efecto es un posible indicador de heterocedasticidad, ya que muestra que la varianza no es constante.

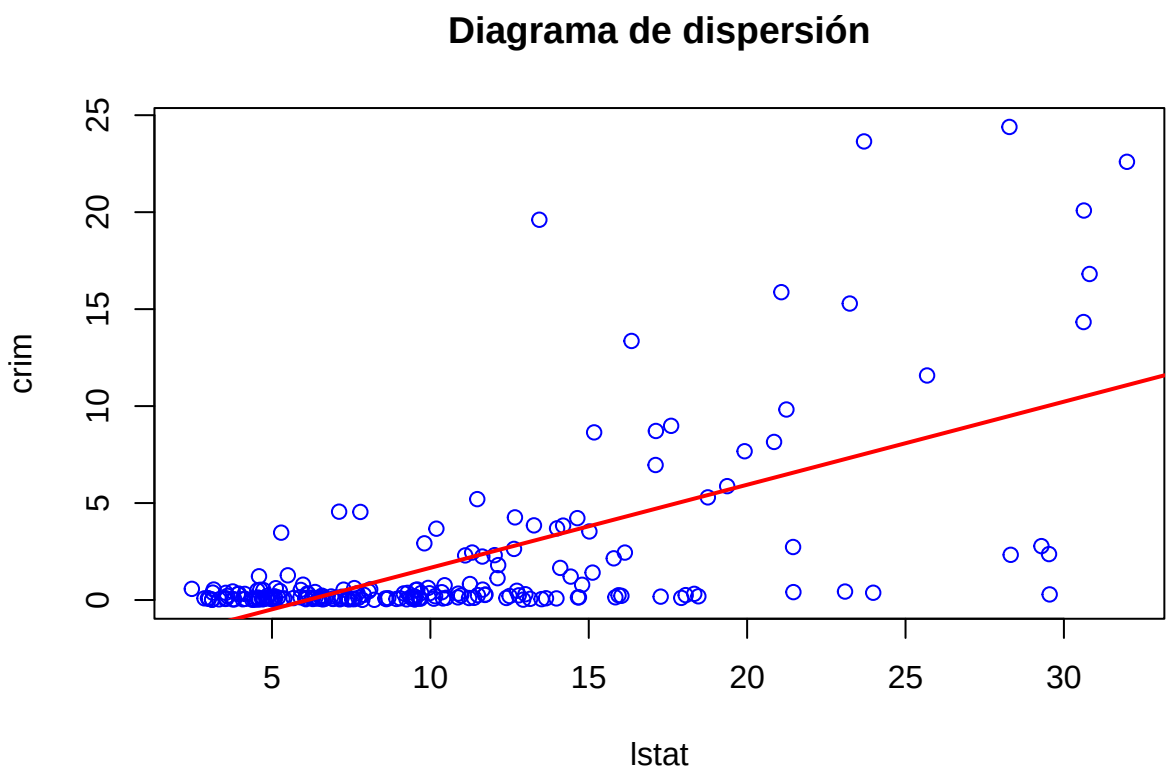
2. Obtén la recta de mínimos cuadrados. Interpreta los resultados obtenidos (coeficientes, significatividad, R^2, contraste del modelo, etc...).

```
summary(lm_crim_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7449 -1.4171 -0.1196  1.0338 16.4752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.62305    0.45836  -5.723 3.86e-08 ***
## lstat        0.42834    0.03673  11.663  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.401 on 197 degrees of freedom
## Multiple R-squared:  0.4085, Adjusted R-squared:  0.4055
## F-statistic:   136 on 1 and 197 DF,  p-value: < 2.2e-16
```

```r
plot(boston$lstat, boston$crim, col='BLUE', main = "Diagrama de dispersión", xlab = 'lstat', ylab = 'cr
abline(coef=coef(lm_crim_lstat), lwd = 2, col='RED')
```



Obtenemos la siguiente recta de mínimos cuadrados:

$$\widehat{\text{crim}} = -2.623 + 0.428 \text{ lstat},$$

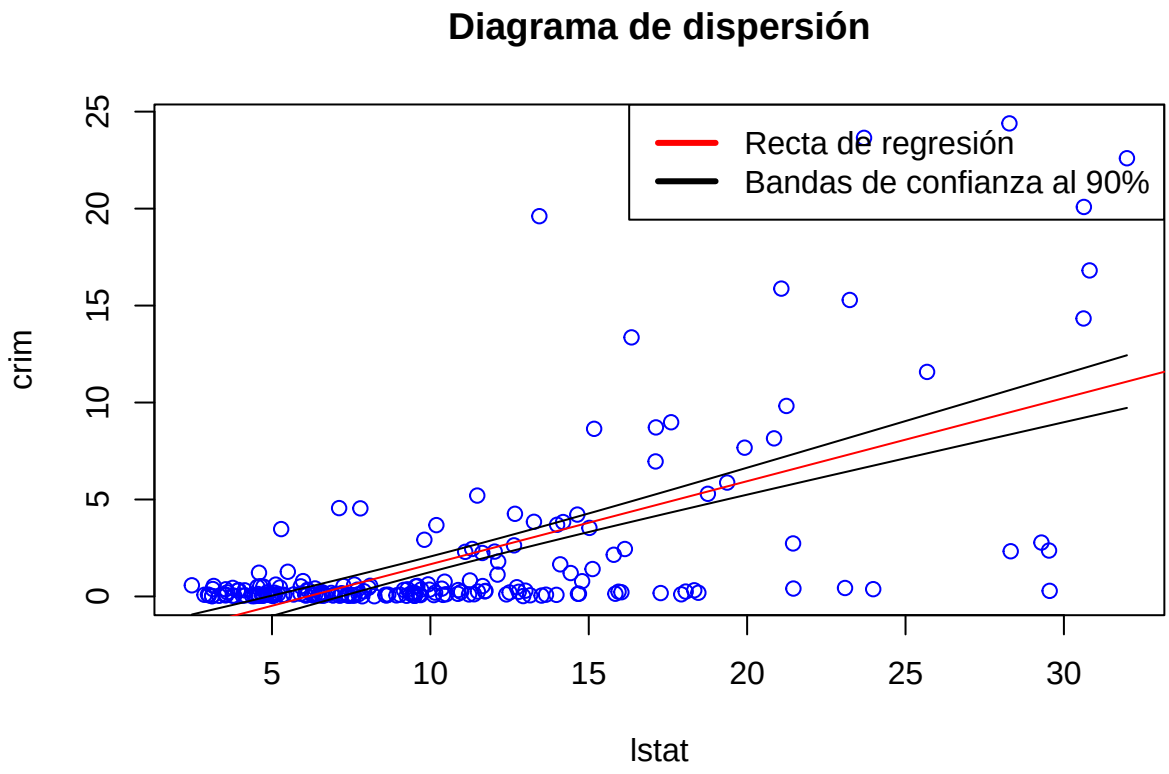
siendo β_0 = -2.6230522 el intercepto y β_1 = 0.4283435 la pendiente de la recta.

El p-valor del estadístico F es prácticamente 0, por lo que rechazamos la hipótesis nula de que ninguna variable es significativa para explicar el modelo. Ahora bien, pese a que la variable `lstat` es muy significativa (al 0%) y la recta pasa por el "centro" de los datos, los puntos están muy dispersos alrededor de ella. Esto visualmente respalda el R^2 ajustado de 0.4055 (menor que el del modelo global), confirmando que aún queda más de la mitad de la varianza del modelo por explicar.

Vemos también que la recta tiende a predecir valores negativos de `crim` para valores bajos de `lstat`, lo cual no tiene sentido si tenemos en cuenta que la variable `crim` mide la tasa de criminalidad en la ciudad. Tenemos una subestimación de `crim` para valores bajos y altos de `lstat` y una sobreestimación para valores medios. Además, la dispersión no uniforme de los datos y la concentración de puntos cerca de cero sugiere transformación logarítmica para la relación.

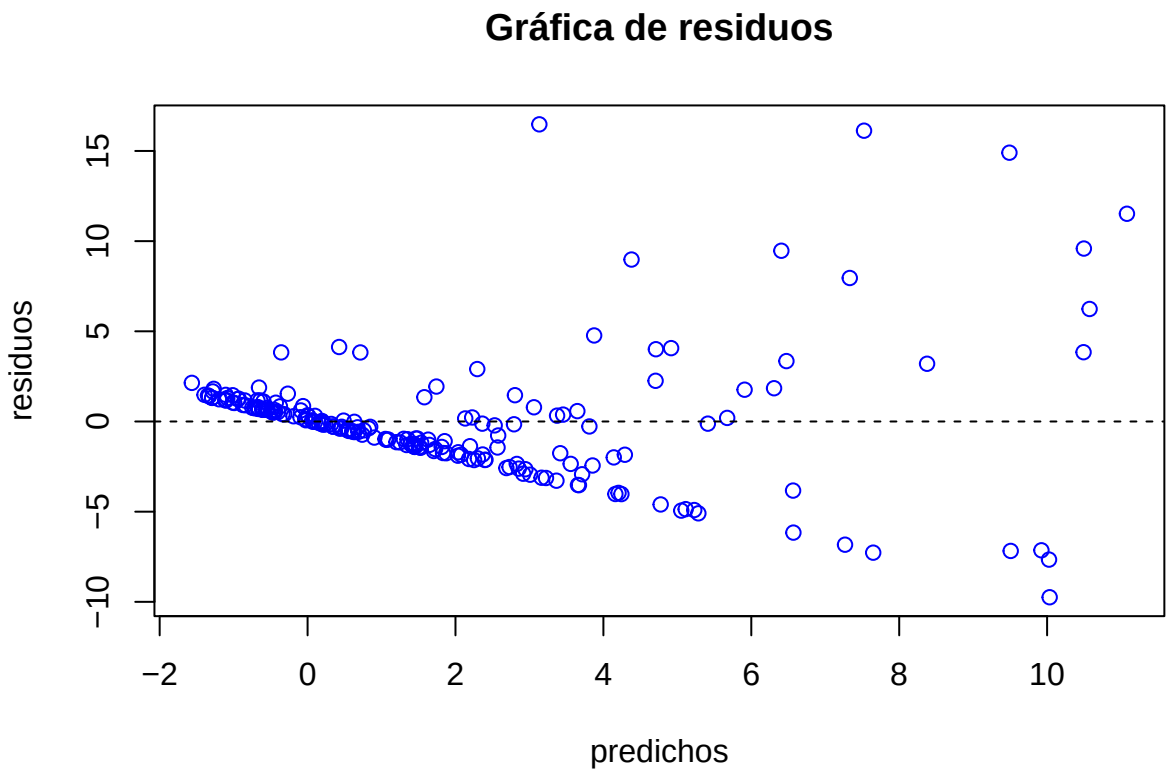
3. Dibuja el diagrama de dispersión, la recta de regresión y las bandas de confianza al 90%.

```r
# Obtención de las bandas de estimación
min <- range(boston$lstat)[1]; max <- range(boston$lstat)[2]
nuevos <- data.frame(list(lstat = seq(min,max,length=100)))
bandas_est <- predict(lm_crim_lstat, newdata = nuevos, interval = "confidence", level = 0.90)

# Representación gráfica
plot(boston$lstat, boston$crim, col='BLUE', main = 'Diagrama de dispersión', xlab = 'lstat', ylab = 'cr
abline(coef=coef(lm_crim_lstat), col='RED')
lines(nuevos$lstat, bandas_est[,2],col='BLACK')
lines(nuevos$lstat, bandas_est[,3],col='BLACK')
legend('topright', legend = c('Recta de regresión', 'Bandas de confianza al 90%'), lwd = 3, col = c('re
```

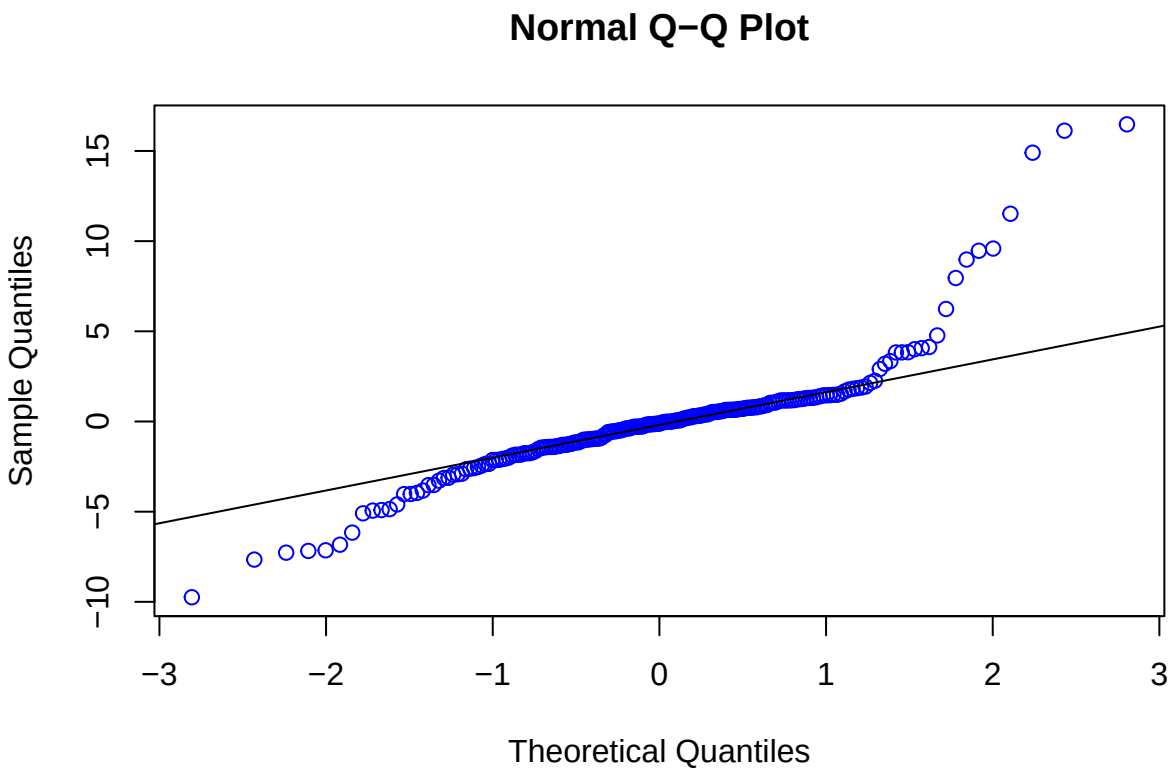


4. Realiza un diagnóstico de los residuos. Si falla alguna de las condiciones, busca una (o varias) posible solución.

```r
# Diagnóstico de linealidad y homocedasticidad
residuos <- residuals(lm_crim_lstat)
predichos <- fitted.values(lm_crim_lstat)
plot(predichos, residuos, col='BLUE', main = 'Gráfica de residuos')
abline(h=0, lty=2)
```




```r
# Diagnóstico de normalidad de los residuos
qqnorm(residuos, col='BLUE')
qqline(residuos)
```




```
shapiro.test(residuos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos
## W = 0.82896, p-value = 4.984e-14
```

Siendo el resultado del p-valor prácticamente 0 en el test de Shapiro_Wilk, rechazamos la hipótesis nula de normalidad de los residuos. Por tanto, hay evidencia estadística a favor de que los residuos no siguen una distribución normal.

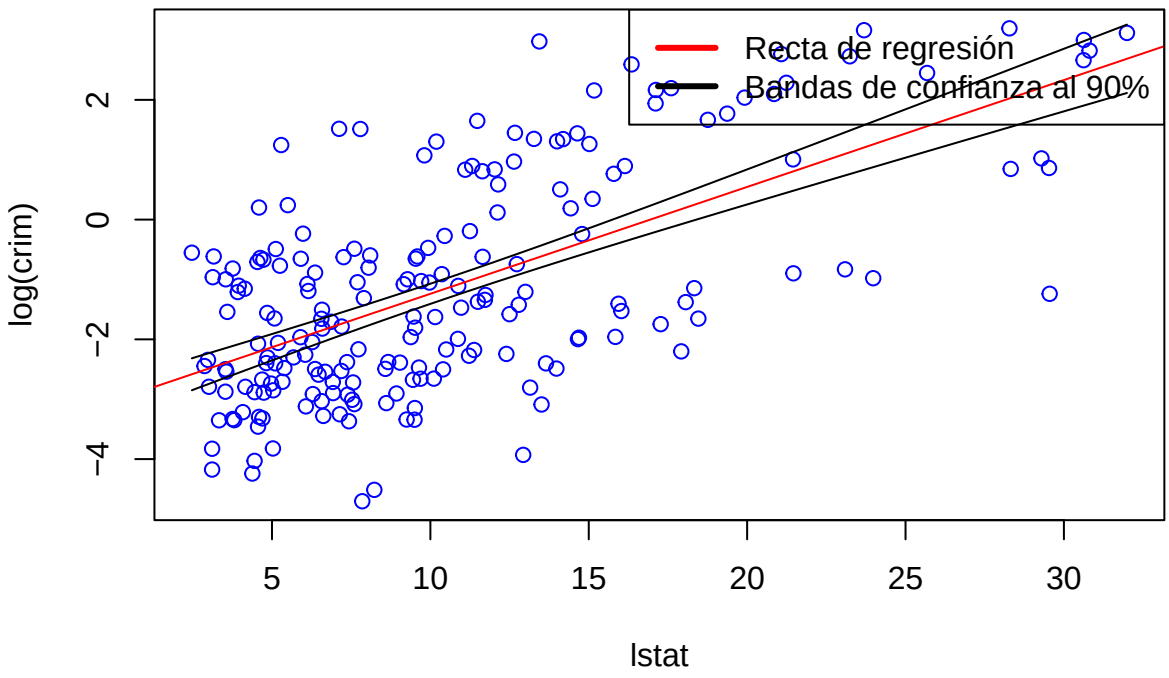
En cuanto a las gráficas, la primera de ellas muestra la no linealidad de los residuos, ya que no aparecen como una nube aleatoria de puntos alrededor de la recta $y = 0$. Además, observamos que la varianza de los mismos aumenta a medida que incrementan los valores predichos, indicando un claro fallo de heterocedasticidad. La segunda gráfica confirma lo que deducido previamente del test de Shapiro-Wilk, es decir, que los residuos no siguen una distribución normal, debido a desviaciones importantes de la línea diagonal, especialmente en los extremos o colas.

En conclusión, el modelo simple no es adecuado para estos datos debido a fallos en la linealidad, homocedasticidad y normalidad de los residuos. Una posible solución es aplicar una transformación logarítmica a la variable respuesta (`crim`) para estabilizar la varianza y mejorar el ajuste.

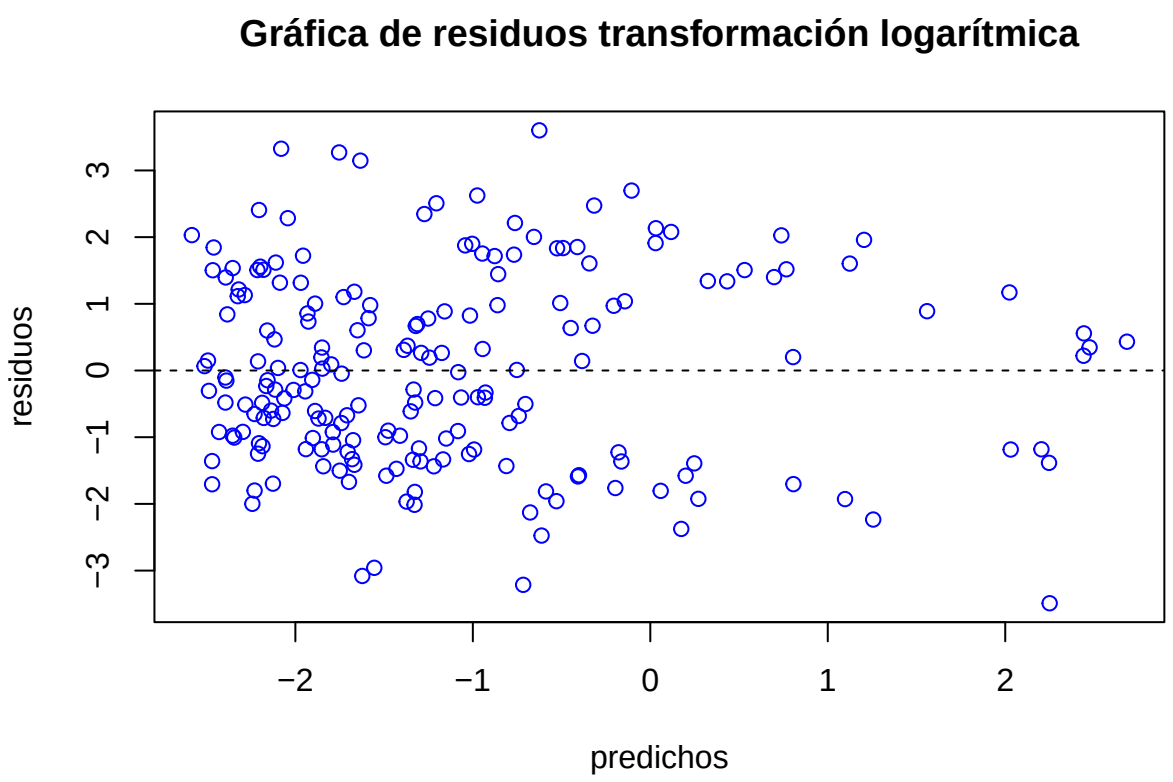
```
# Aplicamos la transformación logarítmica
lm_crim_log <- lm(log(crim)~lstat, data = boston, na.action = na.exclude)
summary(lm_crim_log)
```

```
##
## Call:
## lm(formula = log(crim) ~ lstat, data = boston, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4897 -1.1734 -0.1405  1.1228  3.6012
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.02416    0.19284  -15.68   <2e-16 ***
## lstat        0.17849    0.01545   11.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.431 on 197 degrees of freedom
## Multiple R-squared:  0.4038, Adjusted R-squared:  0.4008
## F-statistic: 133.4 on 1 and 197 DF,  p-value: < 2.2e-16
```

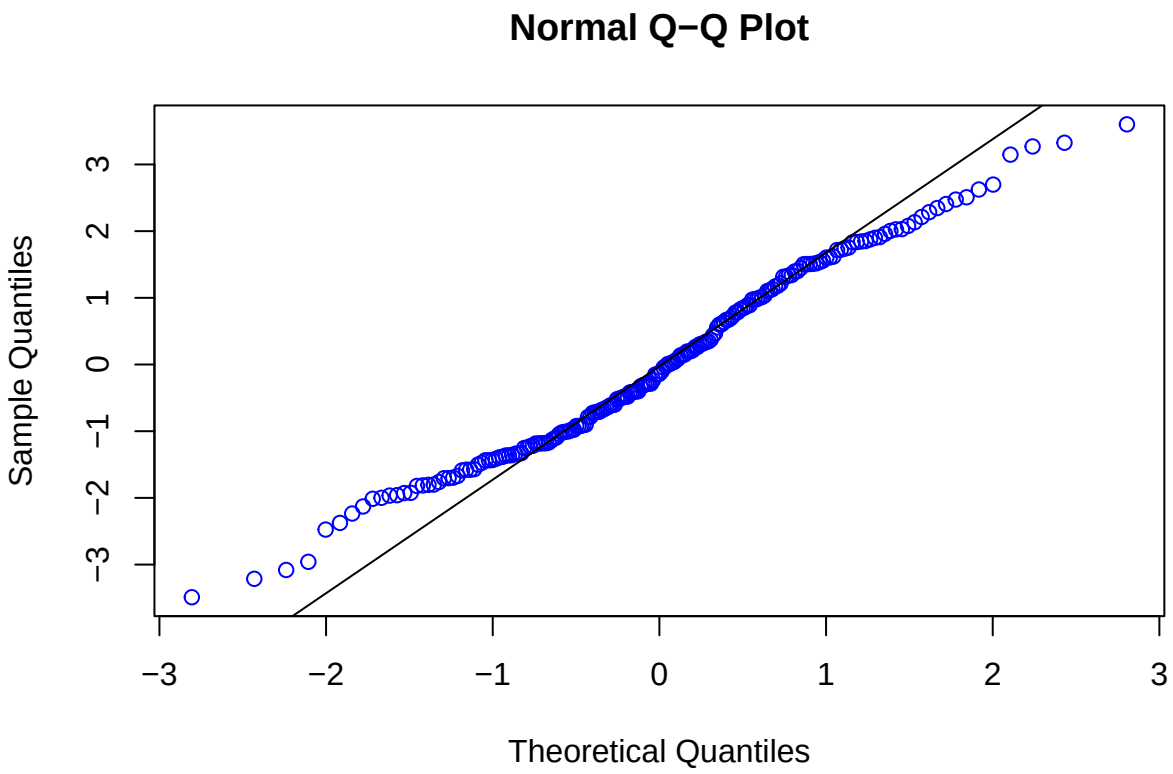
```r
# Representamos el diagrama de dispersión junto con la recta de regresión y las bandas de estimación al
min <- range(boston$lstat)[1]; max <- range(boston$lstat)[2]
nuevos <- data.frame(list(lstat = seq(min,max,length=100)))
bandas_est <- predict(lm_crim_log, newdata = nuevos, interval = "confidence", level = 0.90)

plot(boston$lstat, log(boston$crim), col='blue', main = 'Diagrama de dispersión transformación logarítm
abline(coef = coef(lm_crim_log), col='RED')
lines(nuevos$lstat, bandas_est[,2],col='BLACK')
lines(nuevos$lstat, bandas_est[,3],col='BLACK')
legend('topright', legend = c('Recta de regresión', 'Bandas de confianza al 90%'), lwd = 3, col = c('re
```




```r
# Diagnóstico de linealidad y homocedasticidad
residuos <- residuals(lm_crim_log)
predichos <- fitted.values(lm_crim_log)
plot(predichos, residuos, col='blue', main = 'Gráfica de residuos transformación logarítmica')
abline(h=0, lty=2)
```




```r
# Diagnóstico de normalidad de los residuos
qqnorm(residuos, col='blue')
qqline(residuos)
```




```
shapiro.test(residuos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos
## W = 0.98449, p-value = 0.02745
```

Al realizar la transformación logarítmica, el coeficiente se entiende ahora como que por cada aumento de una unidad (un punto porcentual) en la población de nivel socioeconómico bajo (`lstat`), se espera que la tasa de criminalidad (`crim`) aumente aproximadamente en un 17.85%.

Observamos en el gráfico de dispersión que esta vez los puntos están distribuidos de forma mucho más homogénea alrededor de la recta. Además, en la gráfica de dispersión de los residuos detectamos la linealidad y homocedasticidad que buscábamos, al aparecer los residuos como una nube aleatoria de puntos alrededor de la recta horizontal y la varianza ser constante a lo largo del eje de valores predichos.

Para el test de normalidad de Shapiro-Wilk, obtenemos un p-valor de 0.02745, por lo que, si consideráramos una significancia estadística del 1% ($\alpha = 0.01$), podríamos rechazar la hipótesis nula y no tener suficiente evidencia estadística a favor de la no normalidad de los residuos.

Por lo que, pese a que el R^2 es ligeramente menor (seguimos explicando algo más del 40% de la varianza), se puede considerar que los fallos del anterior modelo se han solucionado con esta transformación.

Ejercicio 2: Considera la variable respuesta `crim` relacionándola con el predictor `medv`.

1. Evalúa el efecto de `medv` sobre `crim`.

```r
lm_crim_medv <- lm(crim ~ medv, data = boston)
summary(lm_crim_medv)
```

```
##
## Call:
## lm(formula = crim ~ medv, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1341 -2.4009 -1.0186  0.7805 18.5528
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.5183     0.8108  10.506  < 2e-16 ***
## medv         -0.2550     0.0296  -8.615 2.26e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.769 on 197 degrees of freedom
## Multiple R-squared:  0.2736, Adjusted R-squared:  0.2699
## F-statistic: 74.21 on 1 and 197 DF,  p-value: 2.258e-15
```

La variable predictora `medv`, que es muy significativa (al 0%), tiene un efecto negativo sobre `crim`: cuando `medv` aumenta una unidad, `crim` se reduce en 0.2549801.

2. Obtén la recta de mínimos cuadrados. Interpreta los resultados obtenidos (coeficientes, significatividad, R^2, contraste del modelo, etc...).

Obtenemos la siguiente recta de mínimos cuadrados:

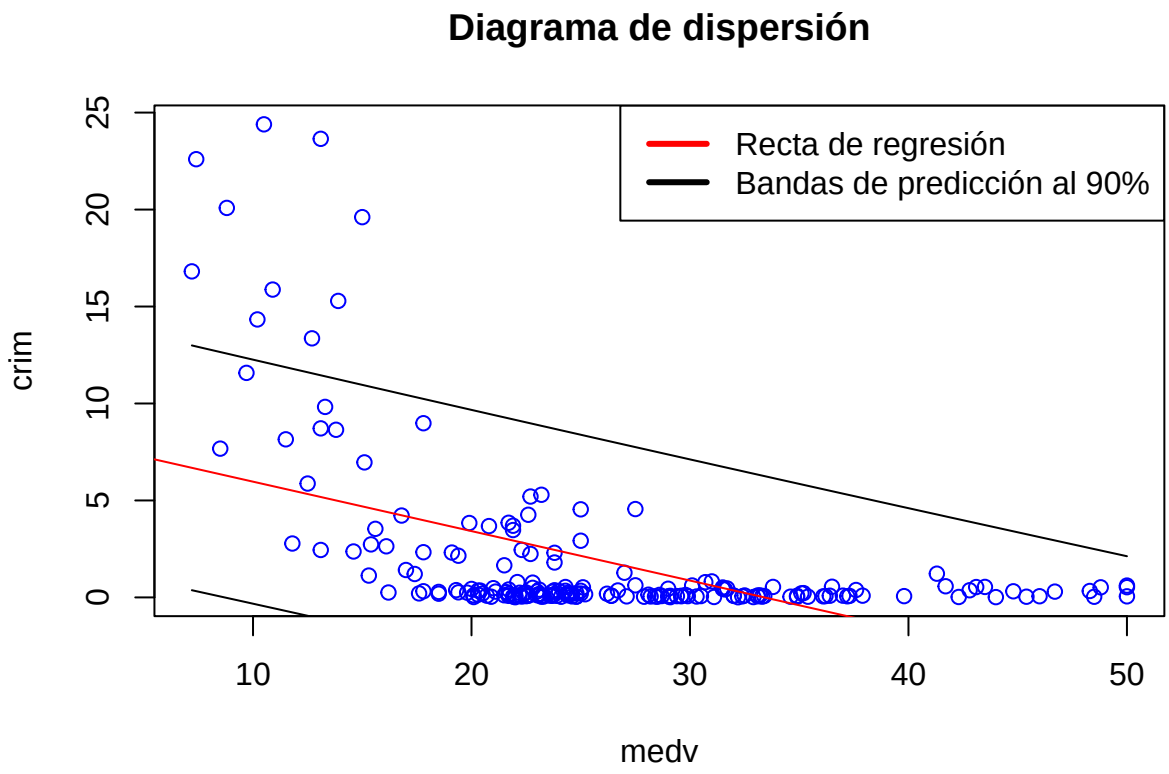
$$\hat{\text{crim}} = -8.5183 - 0.255 \text{ medv},$$

siendo $\beta_0 = 8.518305$ el intercepto y β_1 = -0.2549801 la pendiente de la recta.

Como se ha comentado en el apartado anterior, aparentemente `medv` es muy significativa y si aumenta una unidad, produce una disminución en `crim` de 0.2549801. El p-valor del estadístico F es prácticamente 0, por lo que se rechaza la hipótesis nula de que ninguna variable es significativa. Sin embargo, el R^2 es relativamente bajo (0.2736) por lo que la varianza del modelo no está siendo explicada (estamos dejando casi el 75% de la vaarianza del modelo sin explicar).

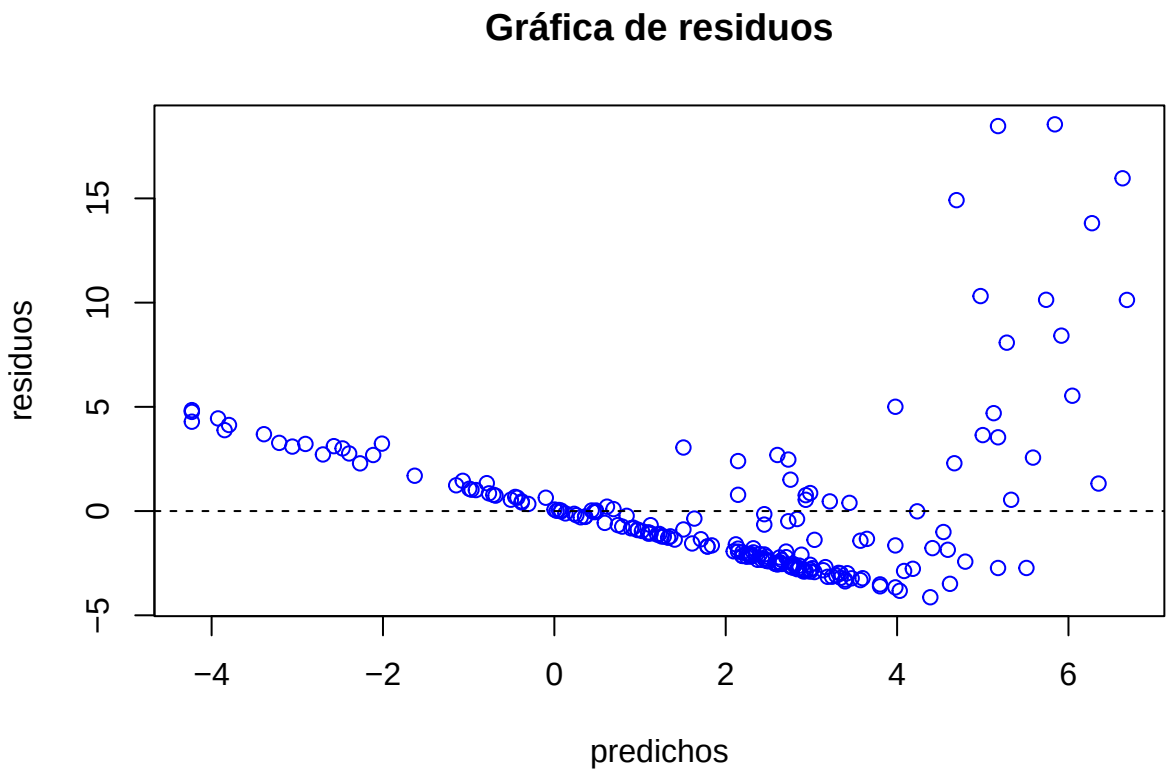
3. Dibuja el diagrama de dispersión, la recta de regresión y las bandas de predicción al 90%.

```r
# Obtención de las bandas de estimación
min2 <- range(boston$medv)[1]; max2 <- range(boston$medv)[2]
nuevos2 <- data.frame(list(medv = seq(min2,max2,length=100)))
bandas_est2 <- predict(lm_crim_medv, newdata = nuevos2, interval = "prediction", level = 0.90)

# Representación gráfica
plot(boston$medv, boston$crim, col='BLUE', main = 'Diagrama de dispersión', xlab = 'medv', ylab = 'crim
abline(coef=coef(lm_crim_medv), col='RED')
lines(nuevos2$medv, bandas_est2[,2],col='BLACK')
lines(nuevos2$medv, bandas_est2[,3],col='BLACK')
legend('topright', legend = c('Recta de regresión', 'Bandas de predicción al 90%'), lwd = 3, col = c('r
```

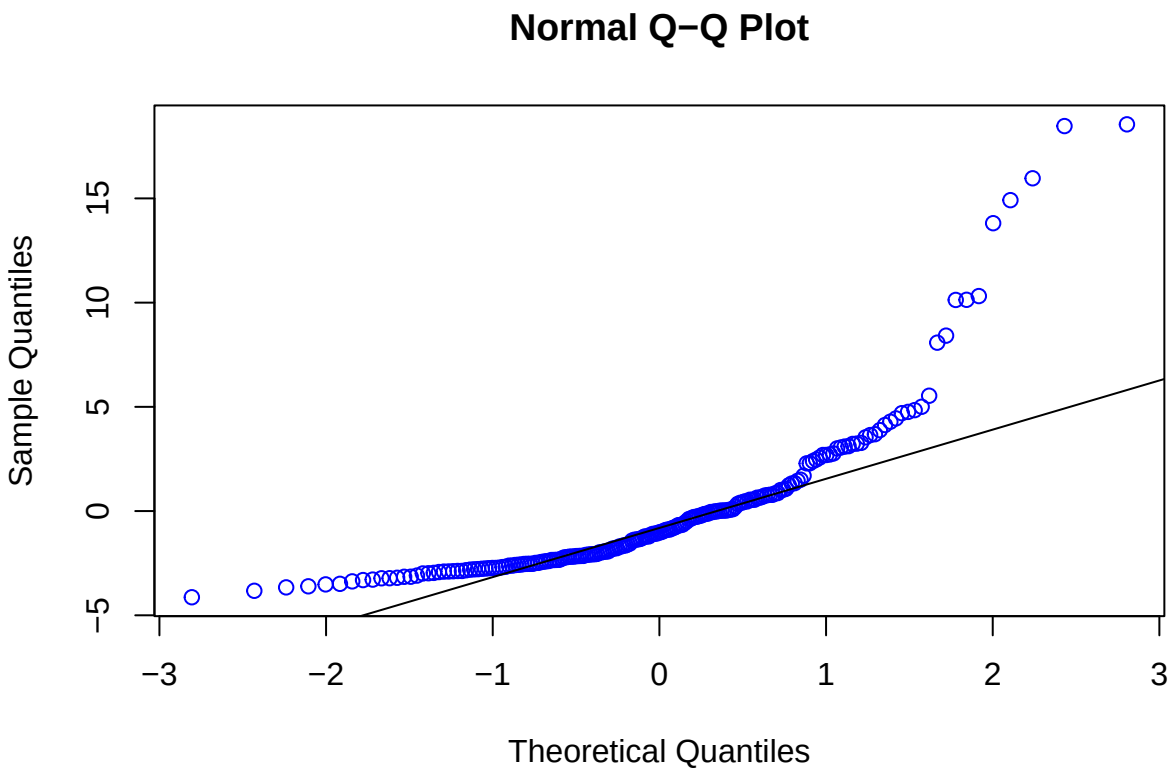


4. Realiza un análisis de los residuos.

```
# Diagnóstico de linealidad y homocedasticidad
residuos2 <- residuals(lm_crim_medv)
predichos2 <- fitted.values(lm_crim_medv)
plot(predichos2, residuos2, col='BLUE', main = 'Gráfica de residuos', xlab = 'predichos', ylab = 'residu
abline(h=0,lty=2)
```




```
# Diagnóstico de normalidad de los residuos
qqnorm(residuos2, col='BLUE')
qqline(residuos2)
```




```
shapiro.test(residuos2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos2
## W = 0.74068, p-value < 2.2e-16
```

Observando el resultado del p-valor cercano a 0 en el test de normalidad de Shapiro-Wilk, rechazamos la hipótesis nula de normalidad de los residuos, es decir, hay evidencia estadística para afirmar que los residuos no siguen una distribución normal.

En cuanto a las gráficas, la primera muestra que la relación no es puramente lineal ya que la dispersión de los residuos se amplía a medida que los valores predichos aumentan y forman una especie de U invertida, representando un fallo de heterocedasticidad. La segunda gráfica, confirma que los residuos no siguen una distribución normal, debido a desviaciones importantes de la línea diagonal, especialmente en los extremos.

En cuanto a las gráficas, la primera de ellas muestra la no linealidad de los residuos, ya que no aparecen como una nube aleatoria de puntos alrededor de la recta $y = 0$. Además, observamos que la dispersión de los mismos aumenta a medida que incrementan los valores predichos, formando una especie de U, indicando un claro fallo de heterocedasticidad. La segunda gráfica confirma lo que deducido previamente del test de Shapiro-Wilk, es decir, que los residuos no siguen una distribución normal, debido a desviaciones importantes de la línea diagonal, especialmente en el extremo o cola superior.

5. ¿Te parece adecuado haber realizado regresión lineal o es preferible otro tipo de regresión?. Ajusta el modelo que te parezca más adecuado.

El modelo simple no es adecuado para estos datos debido a fallos en la linealidad, homocedasticidad y normalidad de los residuos. Una posible solución, viendo la forma de la dispersión de los residuos, es realizar un ajuste parabólico sobre la variable predictora `medv` para estabilizar la varianza y mejorar el ajuste.

En primer lugar, realizamos una transformación logarítmica de los datos.

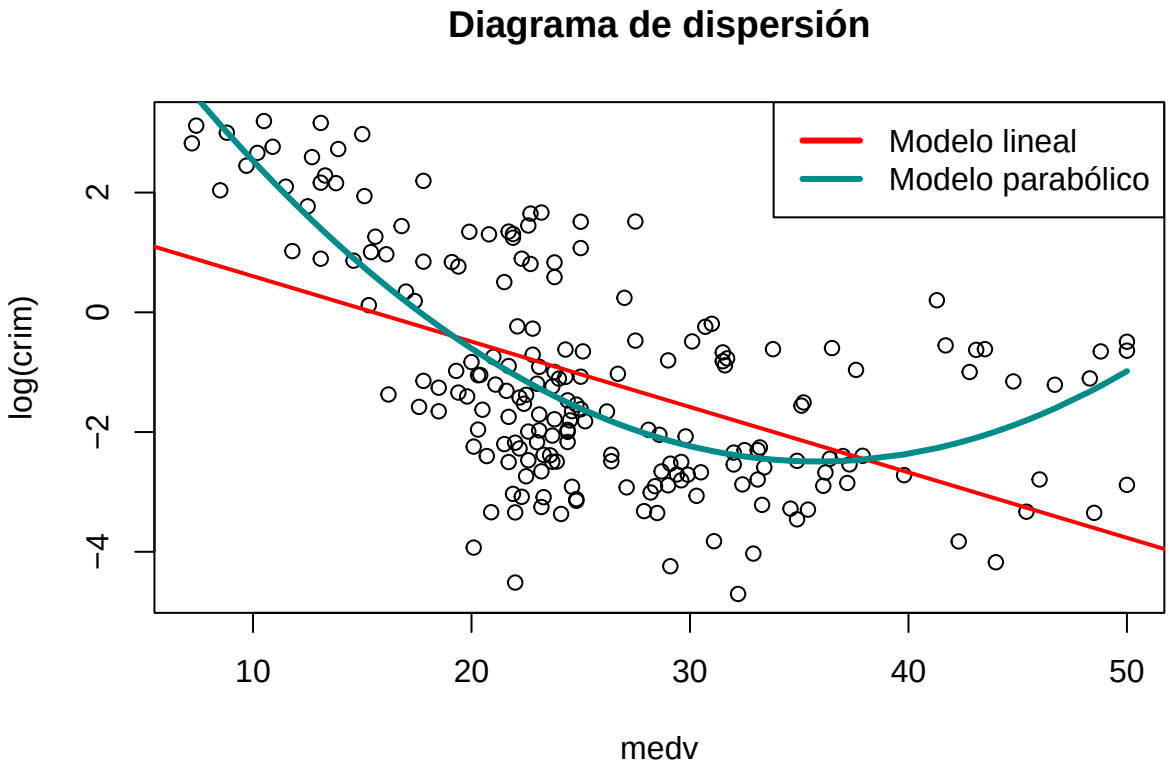
```
lm_medv_lineal <- lm(log(crim) ~ medv, data = boston)
summary(lm_medv_lineal)
```

```
##
## Call:
## lm(formula = log(crim) ~ medv, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8060 -1.1916 -0.3639  1.3985  3.2741
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6954     0.3369   5.032 1.09e-06 ***
## medv         -0.1092     0.0123  -8.881 4.13e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.566 on 197 degrees of freedom
## Multiple R-squared:  0.2859, Adjusted R-squared:  0.2823
## F-statistic: 78.87 on 1 and 197 DF,  p-value: 4.128e-16
```

```
# Planteamos una solución con un ajuste parabólico sobre la variable predictora medv
lm_medv_parab <- lm(log(crim) ~ medv + I(medv^2), data = boston)
summary(lm_medv_parab)
```

```
##
## Call:
## lm(formula = log(crim) ~ medv + I(medv^2), data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4634 -0.8703 -0.1432  0.8783  3.4844
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.1725950  0.6608985  10.853   <2e-16 ***
## medv        -0.5392536  0.0480278 -11.228   <2e-16 ***
## I(medv^2)    0.0075225  0.0008205   9.168   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.314 on 196 degrees of freedom
## Multiple R-squared:  0.5002, Adjusted R-squared:  0.4951
## F-statistic: 98.08 on 2 and 196 DF,  p-value: < 2.2e-16
```

```r
# Representación del ajuste
plot(boston$medv, log(boston$crim), col = 'black', main = "Diagrama de dispersión", xlab = "medv", ylab
abline(coef = coef(lm_medv_lineal), lwd = 2, col = 'red')
lines(sort(boston$medv), fitted(lm_medv_parab)[order(boston$medv)], col = 'darkcyan', lwd = 3)
legend('topright', legend = c('Modelo lineal', 'Modelo parabólico'), lwd = 3, col = c('red', 'darkcyan'
```



Para saber si el coeficiente cuadrático es significativo, utilizamos la función `anova`. Además, para comparar el nuevo modelo parabólico con el modelo lineal, calculamos el coeficiente de información de Akaike con la función `AIC`.

```r
anova(lm_medv_parab, lm_medv_lineal)
```

```
## Analysis of Variance Table
##
## Model 1: log(crim) ~ medv + I(medv^2)
## Model 2: log(crim) ~ medv
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    196 338.16
## 2    197 483.17 -1      -145 84.046 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
print(paste("AIC del modelo lineal:", AIC(lm_medv_lineal)))
```

```
## [1] "AIC del modelo lineal: 747.261213322804"
```

```r
print(paste("AIC del modelo parabólico:", AIC(lm_medv_parab)))
```

```
## [1] "AIC del modelo parabólico: 678.250378785413"
```

Obtenemos un p-valor significativo al 0%, por lo que rechazamos la hipótesis nula y concluimos que hay evidencia estadística a favor de la significatividad del coeficiente cuadrático. Comprobamos también que el valor del AIC se reduce, con lo cual hemos mejorado con nuestro nuevo modelo.

Si hacemos lo mismo para el caso cúbico, empleando la función `poly`, y comparamos con el caso parabólico (para comparar la introducción de polinomios ortogonales) tenemos

```r
lm_medv_3 <- lm(log(crim) ~ poly(medv, 3), data = boston)
anova(lm_medv_parab, lm_medv_3, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: log(crim) ~ medv + I(medv^2)
## Model 2: log(crim) ~ poly(medv, 3)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    196 338.16
## 2    195 336.06  1    2.1033 1.2205 0.2706
```

```r
print(paste("AIC del modelo parabólico:", AIC(lm_medv_3)))
```

```
## [1] "AIC del modelo parabólico: 679.008747098827"
```

El p-valor que obtenemos ya no es menor que el nivel de significación $\alpha = 0.05$, por lo que no rechazamos la hipótesis nula y hay evidencia estadística para afirmar que el coeficiente de orden 3 no es significativo. Además, obtenemos un AIC superior. Por tanto, nos quedamos con el modelo parabólico.

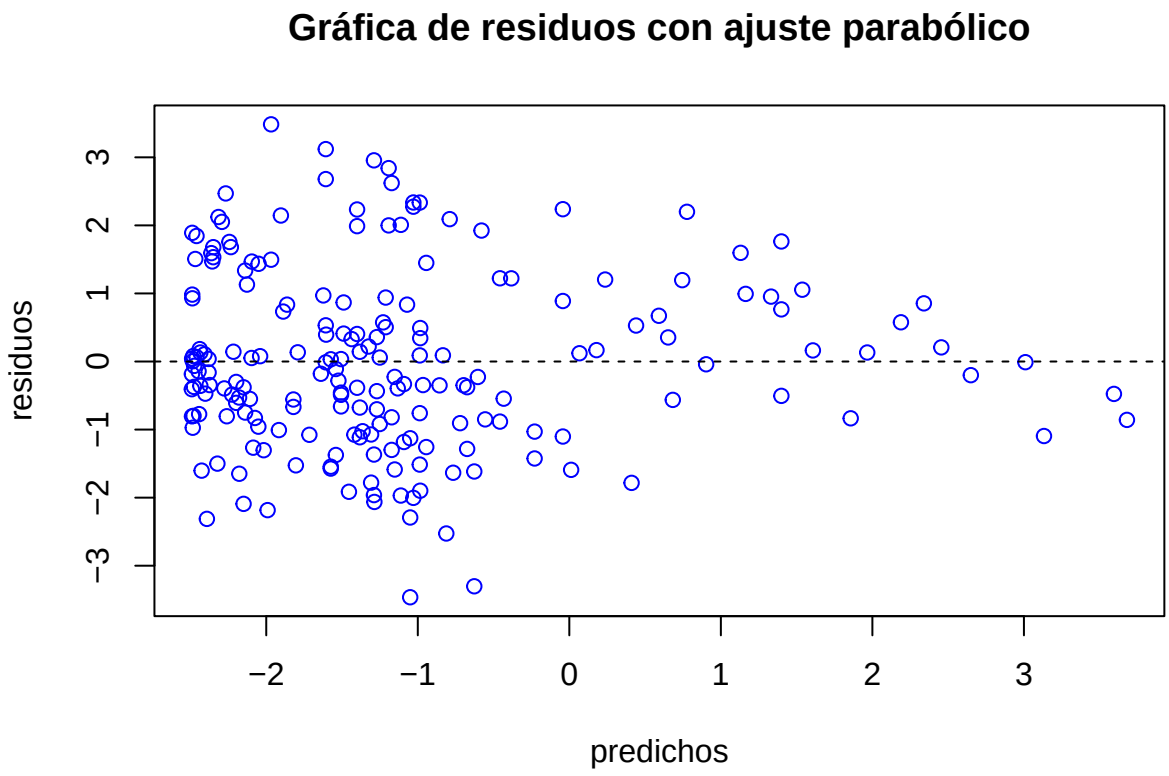
Observamos que con nuestro nuevo modelo obtenemos un valor R^2 ajustado de 0.4951, es decir, explicamos aproximadamente la mitad de la varianza de la variable `crim`, frente a menos del 30 que explicábamos con el modelo de regresión lineal original.

Realicemos un análisis de residuos.

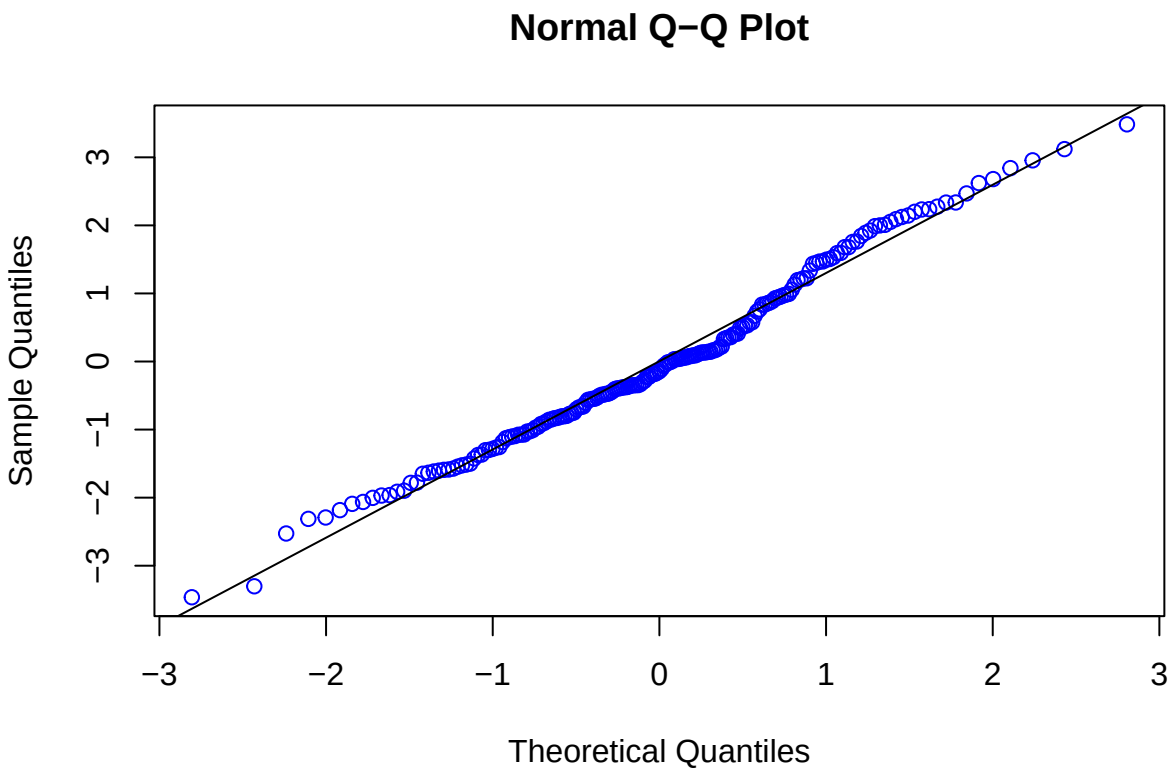
```r
# Diagnóstico de linealidad y homocedasticidad
  # Test para la homocedasticidad
bptest(lm_medv_parab)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_medv_parab
## BP = 2.1995, df = 2, p-value = 0.3329
```

```r
  # Representación gráfica
residuos_trans <- residuals(lm_medv_parab)
predichos_trans <- fitted.values(lm_medv_parab)

plot(predichos_trans, residuos_trans, col='blue', main = 'Gráfica de residuos con ajuste parabólico', xl
abline(h = 0, lty = 2)
```




```r
# Diagnóstico de normalidad de los residuos
qqnorm(residuos_trans, col='blue')
qqline(residuos_trans)
```




```
shapiro.test(residuos_trans)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_trans
## W = 0.98652, p-value = 0.05507
```

Observamos en la gráfica de dispersión de los residuos que la linealidad de los mismos ha mejorado. Además, obtenemos para el test de Breusch-Pagan un p-valor de 0.3329, por lo que no rechazamos la hipótesis nula y concluimos que no hay suficiente evidencia estadística para rechazar la homocedasticidad de los residuos.

Por lo que respecta al test de normalidad de Shapiro-Wilk, obtenemos un p-valor de 0.05507, es decir, tampoco rechazamos la hipótesis nula. En consecuencia, no existe evidencia estadística suficiente para afirmar que los residuos no siguen una distribución normal.

En resumen, nuestro nuevo modelo parabólico ha resuelto los fallos en la linealidad, homocedasticidad y normalidad que presentaban los residuos del modelo lineal. Además, hemos conseguido aumentar el porcentaje de varianza de la variable `crim` explicada por el modelo.

6. ¿Qué tasa de criminalidad se espera para aquellos barrios con un precio mediano de la vivienda de 30000 dólares? ¿Y 10000? ¿Y 100000? Calcula e interpreta los intervalos de confianza y de predicción.

Realizaremos las predicciones y los cálculos de los intervalos de confianza y predicción con el modelo parabólico del apartado anterior, el cual habíamos entrenado sobre los datos transformados con una transformación logarítmica. Deberemos, por tanto, realizar la transformación inversa (función exponecial) a los resultados antes de darles una interpretación.

```
medv_pred_log <- predict(lm_medv_parab, newdata = data.frame(medv = c(30, 10, 100)), interval = "predic
int_pred <- exp(medv_pred_log)

medv_conf_log <- predict(lm_medv_parab, newdata = data.frame(medv = c(30, 10, 100)), interval = "confid
int_conf <- exp(medv_conf_log)
```

Para aquellos barrios con un precio mediano de la vivienda de 30000 dólares la tasa de criminalidad esperada es de 0.107, con un intervalo de confianza de [0.0871, 0.1315] y un intervalo de predicción de [0.0121, 0.9472].

Por lo que respecta a los barrios con precio mediano de la vivienda de 10000 dólares, la tasa de criminalidad esperada es de 12.5825, con un intervalo de confianza de [7.8905, 20.0646] y un intervalo de predicción de [1.3661, 115.8955].

Por último, para los barrios con precio mediano de la vivienda de 100000 dólares, la tasa de criminalidad esperada es de 2.3184884×10^{12}, con un intervalo de confianza de $[2.3033144 \times 10^{9}, 2.3337624 \times 10^{15}]$ y un intervalo de predicción de $[1.6513448 \times 10^{9}, 3.2551582 \times 10^{15}]$.

Notamos que obtenemos valores desproporcionados para el caso de los barrios con precio mediano de la vivienda de 100000 dólares. Esto se debe a que nuestro modelo ha sido entrenado para valores de la variable `medv` entre 7.2 y 50.0, por lo que tratar de hacer predicciones para valores tan alejados de este intervalo puede llevar a resultados fuera del rango lógico de la variable `crim`.

Ejercicio 3:

1. Encuentra el número óptimo de variables a incluir en un modelo predictivo de `crim`, según los criterios R^2, BIC y CP, utilizando la metodología RegSubsets. Indica brevemente en que consiste esta metodología.

```
reg_crim <- regsubsets(crim ~ ., data = boston)
summary <- summary(reg_crim)
```

```
lm1 <- lm(crim ~ lstat, data = boston)
lm2 <- lm(crim ~ ptratio + lstat, data = boston)
lm3 <- lm(crim ~ ptratio + lstat + nox, data = boston)
lm4 <- lm(crim ~ ptratio + lstat + rm + indus, data = boston)
lm5 <- lm(crim ~ ptratio + lstat + rm + indus + chas, data = boston)
lm6 <- lm(crim ~ ptratio + lstat + rm + indus + chas + medv, data = boston)
lm7 <- lm(crim ~ ptratio + lstat + rm + indus + chas + medv + dis, data = boston)
lm8 <- lm(crim ~ ptratio + lstat + rm + indus + chas + medv + dis + age, data = boston)

summary(lm1)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7449 -1.4171 -0.1196  1.0338 16.4752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.62305    0.45836  -5.723 3.86e-08 ***
## lstat        0.42834    0.03673  11.663  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.401 on 197 degrees of freedom
## Multiple R-squared:  0.4085, Adjusted R-squared:  0.4055
## F-statistic:   136 on 1 and 197 DF,  p-value: < 2.2e-16
```

```
summary(lm2)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4744 -1.5043 -0.1688  1.3027 15.2234
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.4685     1.9081  -6.010 8.88e-09 ***
## ptratio       0.5430     0.1140   4.761 3.74e-06 ***
## lstat         0.3768     0.0365  10.321  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.228 on 196 degrees of freedom
## Multiple R-squared:  0.4698, Adjusted R-squared:  0.4644
## F-statistic: 86.83 on 2 and 196 DF,  p-value: < 2.2e-16
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + lstat + nox, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1257 -1.7058 -0.1846  1.1153 15.2917
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.85438    2.12165  -7.473 2.58e-12 ***
## ptratio       0.58564    0.11015   5.317 2.88e-07 ***
## lstat         0.27300    0.04321   6.318 1.76e-09 ***
## nox           8.81297    2.14029   4.118 5.65e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.104 on 195 degrees of freedom
## Multiple R-squared:  0.5122, Adjusted R-squared:  0.5047
## F-statistic: 68.25 on 3 and 195 DF,  p-value: < 2.2e-16
```

```r
summary(lm4)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + lstat + rm + indus, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2628 -1.4329 -0.3536  1.1487 15.0517
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.34034    3.33639  -6.996 4.17e-11 ***
## ptratio       0.53438    0.10668   5.009 1.23e-06 ***
## lstat         0.38209    0.04889   7.816 3.38e-13 ***
## rm            1.55316    0.35984   4.316 2.53e-05 ***
## indus         0.21310    0.04510   4.725 4.40e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.981 on 194 degrees of freedom
## Multiple R-squared:  0.5524, Adjusted R-squared:  0.5432
## F-statistic: 59.86 on 4 and 194 DF,  p-value: < 2.2e-16
```

```r
summary(lm5)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + lstat + rm + indus + chas, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6170 -1.5394 -0.4504  1.2780 14.5798
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.22754    3.23640  -7.486 2.46e-12 ***
## ptratio       0.54542    0.10325   5.282 3.42e-07 ***
## lstat         0.38740    0.04732   8.187 3.57e-14 ***
## rm            1.66549    0.34940   4.767 3.68e-06 ***
## indus         0.23869    0.04415   5.406 1.89e-07 ***
## chas         -2.26481    0.59967  -3.777 0.000212 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.884 on 193 degrees of freedom
## Multiple R-squared:  0.5832, Adjusted R-squared:  0.5724
## F-statistic: 54.02 on 5 and 193 DF,  p-value: < 2.2e-16
```

```r
summary(lm6)
```

```
##
```

```
## Call:
## lm(formula = crim ~ ptratio + lstat + rm + indus + chas + medv,
##     data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7970 -1.5485 -0.3691  1.1445 14.9299
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.65959    3.24276  -7.296 7.60e-12 ***
## ptratio       0.49686    0.10721   4.634 6.60e-06 ***
## lstat         0.36243    0.04964   7.301 7.37e-12 ***
## rm            2.06325    0.42745   4.827 2.82e-06 ***
## indus         0.22082    0.04537   4.867 2.36e-06 ***
## chas         -2.24622    0.59737  -3.760 0.000225 ***
## medv         -0.07215    0.04502  -1.602 0.110694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.873 on 192 degrees of freedom
## Multiple R-squared:  0.5887, Adjusted R-squared:  0.5759
## F-statistic: 45.81 on 6 and 192 DF,  p-value: < 2.2e-16
```

```
summary(lm7)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + lstat + rm + indus + chas + medv +
##     dis, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2109 -1.4888 -0.4556  1.0463 14.9060
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.44533    3.53419  -5.785 2.92e-08 ***
## ptratio       0.49074    0.10622   4.620 7.04e-06 ***
## lstat         0.32494    0.05208   6.239 2.78e-09 ***
## rm            2.04706    0.42340   4.835 2.73e-06 ***
## indus         0.15639    0.05379   2.907 0.004076 **
## chas         -2.30041    0.59214  -3.885 0.000141 ***
## medv         -0.10344    0.04684  -2.208 0.028427 *
## dis          -0.29747    0.13653  -2.179 0.030569 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.845 on 191 degrees of freedom
## Multiple R-squared:  0.5987, Adjusted R-squared:  0.584
## F-statistic: 40.71 on 7 and 191 DF,  p-value: < 2.2e-16
```

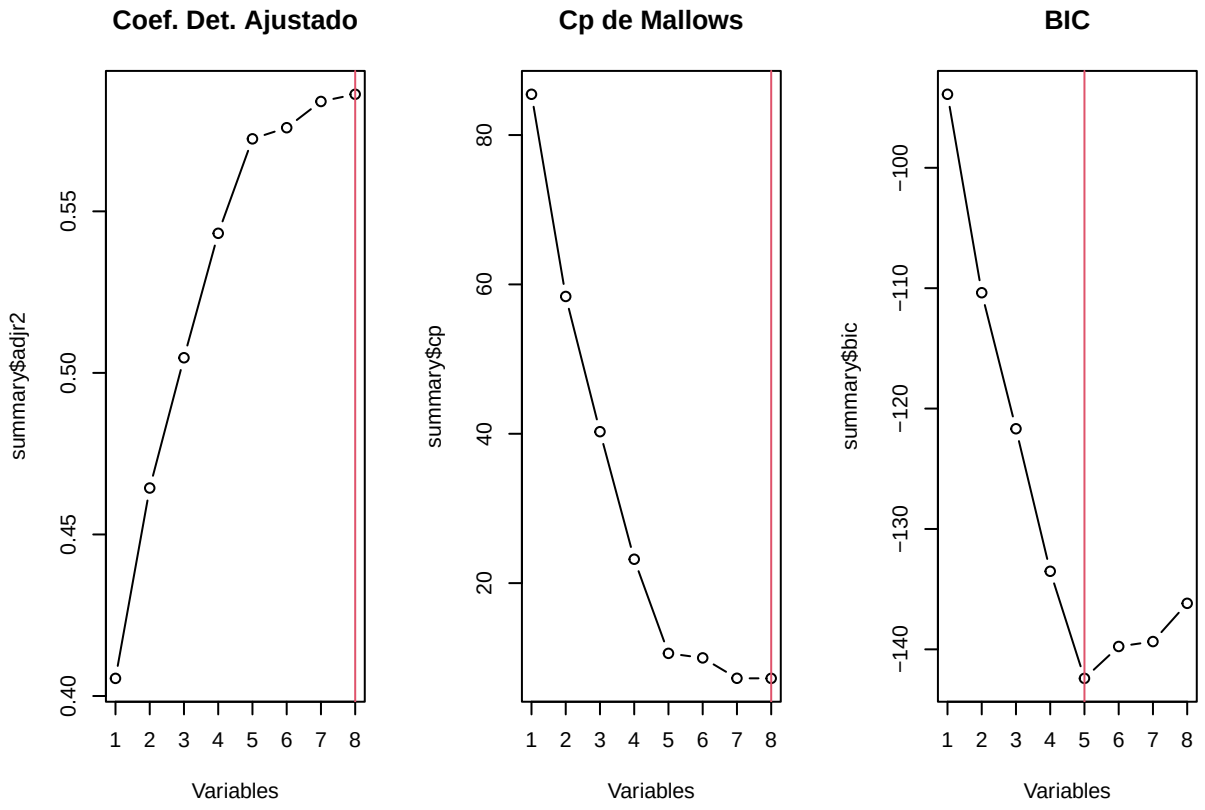
```r
summary(lm8)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + lstat + rm + indus + chas + medv +
##     dis + age, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4121 -1.4495 -0.4459  1.1125 14.9193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.29915    3.61527  -5.338 2.66e-07 ***
## ptratio       0.47713    0.10636   4.486 1.25e-05 ***
## lstat         0.34221    0.05334   6.416 1.09e-09 ***
## rm            2.16730    0.43061   5.033 1.12e-06 ***
## indus         0.16455    0.05395   3.050 0.002615 **
## chas         -2.23558    0.59230  -3.774 0.000214 ***
## medv         -0.11273    0.04717  -2.390 0.017836 *
## dis          -0.44988    0.17313  -2.598 0.010099 *
## age          -0.01759    0.01234  -1.425 0.155726
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.837 on 190 degrees of freedom
## Multiple R-squared:  0.603,  Adjusted R-squared:  0.5862
## F-statistic: 36.07 on 8 and 190 DF,  p-value: < 2.2e-16
```

```r
resultado <- cbind(summary$rsq, summary$adjr2, summary$cp, summary$bic)
colnames(resultado) <- c('Rsq', 'RsqAdj', 'Cp', 'BIC')

length(summary$adjr2)
```

```
## [1] 8
```

```r
par(mfrow = c(1,3))
plot(1:8, summary$adjr2, xlab = "Variables", main = "Coef. Det. Ajustado",
     type="b")
abline(v = which.max(summary$adjr2), col = 2)
plot(1:8, summary$cp, xlab = "Variables", main = "Cp de Mallows",
     type='b')
abline(v = which.min(summary$cp), col = 2)
plot(1:8, summary$bic, xlab = "Variables", main = "BIC",
     type = "b")
abline(v = which.min(summary$bic), col = 2)
```




```
par(mfrow = c(1,1))

AIC(lm1, lm2, lm3, lm4, lm5, lm6, lm7, lm8)
```

```
##     df       AIC
## lm1  3 1055.9282
## lm2  4 1036.1499
## lm3  5 1021.5584
## lm4  6 1006.4255
## lm5  7  994.2365
## lm6  8  993.5925
## lm7  9  990.7069
## lm8 10  990.5907
```

Observamos que el modelo 5 tiene el valor de BIC más bajo (-142.41) y el modelo 8 el AIC más bajo (990.59), por lo que significa que son los modelos que mejor equilibran el ajuste. El modelo 8 contiene el R^2 ajustado más elevado pese a tener `age` como variable no significativa, por lo que es el que mejor explica la varianza de la variable respuesta. Por último, el modelo 8 tiene el CP de Mallows más bajo por lo que tiene menor error de predicción.

Teniendo en cuenta estos resultados, podríamos elegir como mejor modelo el 5, por penalización de complejidad (menos variables) o el modelo 8 si preferimos una predicción más exacta ya que el BIC suele ser mejor en modelos más simples (como en el 5).

- ¿Qué variables incluye el modelo obtenido? (Seleccionar el criterio que más os guste). Interpreta los coeficientes obtenidos. ¿Tienen todas sentido?. ¿Son significativos?.

El modelo final incluye 8 variables: `ptratio`, `lstat`, `rm`, `indus`, `chas`, `medv`, `dis` y `age`.

La mayoria de los coeficientes tienen la dirección esperada. Las características de desventaja (`ptratio`, `lstat`, `indus`) aumentan `crim` (signo positivo), mientras que las características de ventaja (`chas`, `medv`, `dis`) lo disminuyen (signo negativo). El coeficiente de `rm` (número promedio de habitaciones) es positivo. Esto va en contra de la expectativa de que más habitaciones (casas mas grandes/caras) indicarian mayor riqueza y, por lo tanto, menor criminalidad. Esto sugiere que `rm` está correlacionada con alguna otra variable.

2. Selecciona el mejor modelo con el método stepwise. Indica brevemente en que consiste esta metodología y contesta a las siguientes preguntas:

- ¿Qué modelo piensas que es mejor? (Entre este y el/los obtenido/s mediante Regsubsets).

```
crim_global <- lm(crim ~., data = boston)
step_crim <- step(crim_global, direction = 'both', trace = 0)
summary(step_crim)
```

```
##
## Call:
## lm(formula = crim ~ indus + chas + rm + age + dis + ptratio +
##     lstat + medv, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4121 -1.4495 -0.4459  1.1125 14.9193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.29915    3.61527  -5.338 2.66e-07 ***
## indus         0.16455    0.05395   3.050 0.002615 **
## chas         -2.23558    0.59230  -3.774 0.000214 ***
## rm            2.16730    0.43061   5.033 1.12e-06 ***
## age          -0.01759    0.01234  -1.425 0.155726
## dis          -0.44988    0.17313  -2.598 0.010099 *
## ptratio       0.47713    0.10636   4.486 1.25e-05 ***
## lstat         0.34221    0.05334   6.416 1.09e-09 ***
## medv         -0.11273    0.04717  -2.390 0.017836 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.837 on 190 degrees of freedom
## Multiple R-squared:  0.603,  Adjusted R-squared:  0.5862
## F-statistic: 36.07 on 8 and 190 DF,  p-value: < 2.2e-16
```

Coincide con una de las opciones planteadas con la otra metodología (modelo 8), de acuerdo con los estadísticos contrastados, por lo que este modelo sería la mejor opción.

- ¿Qué % de la varianza de `crim` explica el modelo?

  El modelo con stepwise se compone de 8 variables y explica un 60% de la variación en la tasa de criminalidad (`crim`). La mayoria de los predictores son muy significativos, exceptuando la variable `age`.

- ¿Cuál es el efecto de la variable `chas` sobre `crim`?

El coeficiente de `chas` -2.23558 significa que, manteniendo todas las demás variables constantes, las areas que colindan con el río Charles (`chas` = 1) tienen una tasa de criminalidad 2,24 unidades menor que las areas que no coinciden con el río.

3. Con el modelo obtenido con stepwise, realiza el diagnóstico de tu modelo, sin emprender ninguna acción, e indica los problemas que presenta.

Repetimos que las variables explican un 60% de la varianza del modelo y que este en su conjunto es altamente significativo (estadístico F con un p-valor cercano a 0). Esto indica que al menos una de las variables predictoras es útil para el modelo.

Las variables `indus`, `chas`, `rm`, `ptratio`, y `lstat` son las que tienen el impacto más significativo en la predicción de la tasa de criminalidad. La variable age no es estadísticamente significativa, lo que hace que su inclusión no mejora significativamente la predicción del modelo.

Se podría considerar hacer un modelo final eliminando `age` y `rm` por su supuesta multicolinealidad mencionada anteriormente para ver si esto mejora el R^2 ajustado.

4. Emprende ahora las acciones que te parezcan oportunas e indica los problemas que has conseguido solucionar o mejorar un poco.

```
lm_crim_nuevo <- lm(crim ~ lstat + indus + chas + dis + ptratio + medv, data=boston)
summary(lm_crim_nuevo)
```

```
##
## Call:
## lm(formula = crim ~ lstat + indus + chas + dis + ptratio + medv,
##     data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1032 -1.6556 -0.4247  1.3006 14.3443
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.63114    3.05702  -3.478 0.000626 ***
## lstat         0.27218    0.05381   5.058 9.87e-07 ***
## indus         0.15835    0.05684   2.786 0.005870 **
## chas         -2.13663    0.62467  -3.420 0.000764 ***
## dis          -0.30906    0.14424  -2.143 0.033403 *
## ptratio       0.53369    0.11184   4.772 3.61e-06 ***
## medv          0.02051    0.04143   0.495 0.621118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.006 on 192 degrees of freedom
## Multiple R-squared:  0.5496, Adjusted R-squared:  0.5355
## F-statistic: 39.05 on 6 and 192 DF,  p-value: < 2.2e-16
```

Eliminando las dos variables mencionadas anteriormente, vemos que ahora medv deja de ser significativa. La eliminamos y volvemos a hacer la regresión.

```r
lm_crim_nuevo <- lm(crim ~ lstat + indus + chas + dis + ptratio, data=boston)
summary(lm_crim_nuevo)
```

```
##
## Call:
## lm(formula = crim ~ lstat + indus + chas + dis + ptratio, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8133 -1.6998 -0.3565  1.2702 14.3779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.44529    1.89569  -4.983 1.39e-06 ***
## lstat        0.25597    0.04262   6.006 9.28e-09 ***
## indus        0.14625    0.05121   2.856 0.004763 **
## chas        -2.12297    0.62284  -3.409 0.000795 ***
## dis         -0.33604    0.13329  -2.521 0.012507 *
## ptratio      0.51833    0.10724   4.833 2.73e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3 on 193 degrees of freedom
## Multiple R-squared:  0.549,  Adjusted R-squared:  0.5373
## F-statistic: 46.99 on 5 and 193 DF,  p-value: < 2.2e-16
```

Ahora vemos que todas las variables son significativas y sus coeficientes tienen sentido. Además el R^2 es mayor de 50% y el estadistico F confirma que al menos una de las variables es significativa.

5. Obtén la predicción de la tasa de criminalidad para un barrio en la mediana de los predictores en el modelo escogido. *Notar que las variables categóricas se tratan de diferente manera, no hay mediana.*

```r
med_ptratio <- median(boston$ptratio)
med_lstat <- median(boston$lstat)
med_indus <- median(boston$indus)
med_dis <- median(boston$dis)

datos_medianos <- data.frame(
  ptratio = med_ptratio,
  lstat = med_lstat,
  indus = med_indus,
  chas = 0, #asumimos chas = 0
  dis = med_dis
)

print(prediccion_crim <- predict(lm_crim_nuevo, newdata = datos_medianos))
```

```
##        1
## 1.628668
```

La tasa de incidentes criminales por $10,000$ habitantes en este barrio hipotético (con características medianas) es de 1.63