CACCS: Secuencia de taller de RStudio – Parte 3

Rashid C.J. Marcano Rivera

10 de oct. de 2025

Contents

Recapitulando **2**

Modelos estadísticos **4**

Correlaciones **6**

Modelo de regresión lineal simple **9**

Supuestos de modelos lineales 14

Predicciones con estimados 19

Regresión múltiple **23**

Análisis de suposiciones 28

Modelos lineales generalizados **69**

Análisis de la bondad del modelo 74

Modelos jerárquicos **75**

Modelo nulo 77

Modelo con predictor de primer nivel 79

Modelo con pendientes aleatorias 81

Modelos longitudinales **84**

Modelo inicial 93

Modelo con variable de sexo 96

Series de tiempo: ejemplo de finanzas **97**

Qué aprendimos en este taller **113**

Inferencia estadística

Este taller está basado en elementos y ejemplos del libro de Rafael Irizarry, aquí, así como pedazos del curso de la Universidad de la República, en Uruguay, disponible en esta página de RPubs de RStudio y este repaso sobre modelos longitudinales por Alessio Crippa también en RPubs de RStudio. Recomiendo complementar el análisis con el libro *Data Analysis Using Regression and Multilevel/Hierarchical Models* de Andrew Gelman y Jennifer Hill.

Si aún no has instalado R, está aquí. Acto seguido, baja RStudio. Puedes también ir a la nube en Posit Cloud.

Recapitulando

La vez anterior, tomamos un recorrido a través de distintos tipos de visualizaciones. En efecto, una buena visualización puede demostrar bastante sobre elementos en nuestros estudios. Por ejemplo

```
library(wooldridge)

data(wage1)
head(wage1)
```

```
##   wage educ exper tenure nonwhite female married numdep smsa northcen south
## 1 3.10   11     2      0        0      1       0      2    1        0     0
## 2 3.24   12    22      2        0      1       1      3    1        0     0
## 3 3.00   11     2      0        0      0       0      2    0        0     0
## 4 6.00    8    44     28        0      0       1      0    1        0     0
## 5 5.30   12     7      2        0      0       1      1    0        0     0
## 6 8.75   16     9      8        0      0       1      0    1        0     0
##   west construc ndurman trcommpu trade services profserv profocc clerocc
## 1    1        0       0        0     0        0        0       0       0
## 2    1        0       0        0     0        1        0       0       0
## 3    1        0       0        0     1        0        0       0       0
## 4    1        0       0        0     0        0        0       0       1
## 5    1        0       0        0     0        0        0       0       0
## 6    1        0       0        0     0        0        1       1       0
##   servocc    lwage expersq tenursq
## 1       0 1.131402       4       0
## 2       1 1.175573     484       4
## 3       0 1.098612       4       0
## 4       0 1.791759    1936     784
## 5       0 1.667707      49       4
## 6       0 2.169054      81      64
```

¿Qué aprendemos de ver estos datos así? ¿Podemos rápidamente determinar a si años de educación se traducen a mayores ingresos? ¿Podemos determinar si afecta, en algo, la relación marital? Para muchos humanos, es difícil extraer información con meramente mirar a números sin contexto adicional. Pero podríamos ver algo en este gráfico

Vivimos en una era de creciente disponibilidad de conjuntos de datos informativos y de herramientas de software, con lo cual el uso de visualizaciones ha aumentado en diversos espacios: académicos, gubernamentales, organizaciones sociales, prensa, e industrias varias. Sin embargo, R, un programa estadístico diseñado

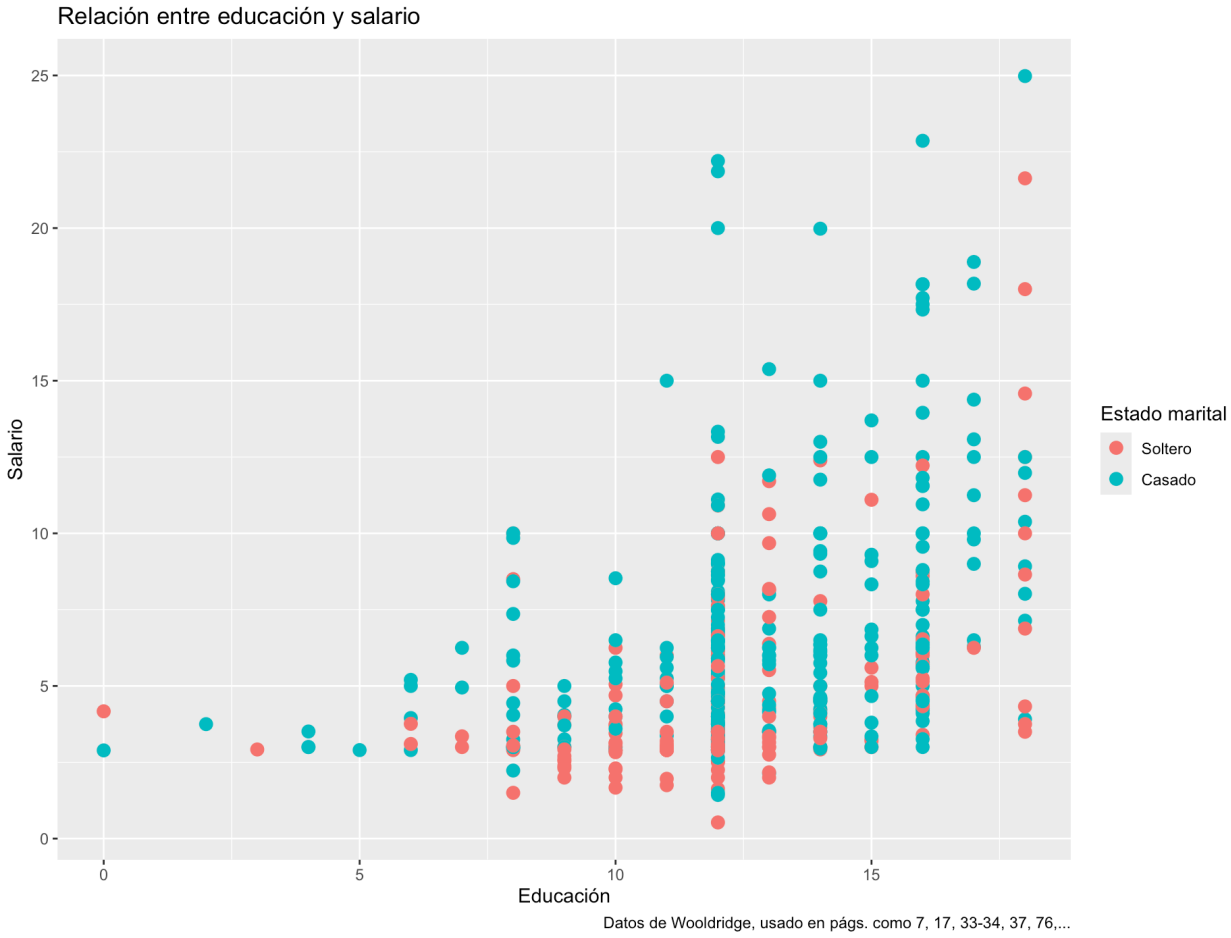


Figure 1: Meta 1: datos de salario, educación, y estado marital.

para el manejo y análisis de distintos formatos de datos, tenemos un sinnúmero de opciones para trabajar distintos tipos de complejidades en datos y análisis.

En esta secuencia de cuatro semanas, continuamos de lo aprendido en el pasado; aplicaremos funciones para cargar datos, manipularlos, y retomamos problemas que visualizamos la semana pasada como el señalado, para analizarlos con análisis estadísticos y visualizaciones relativas a la inferencia estadística. De una vez estaremos entrando en algunos de los diagnósticos que podremos desplegar para evaluar qué tan adecuado es un modelo para trabajar ciertos datos. Aprenderemos hoy:

1. Varias operaciones estadísticas,
2. Estadística inferencial, en sus varias versiones para modelos simples, lineales, jerárquicos y longitudinales.
3. Gráficos y pruebas que ayuden a entender y diagnosticar los modelos estadísticos que usaremos.

Modelos estadísticos

Queríamos entender la vez anterior la relación de ingreso con otras variables. Para cargar los datos escribiremos

```
library(wooldridge)
base <- wage1
#View(base)
names(base)
```

```
##  [1] "wage"     "educ"     "exper"    "tenure"   "nonwhite" "female"
##  [7] "married"  "numdep"   "smsa"     "northcen" "south"    "west"
## [13] "construc" "ndurman"  "trcommpu" "trade"    "services" "profserv"
## [19] "profocc"  "clerocc"  "servocc"  "lwage"    "expersq"  "tenursq"
```

```
#?wage1
```

Las variables que utilizaremos son las siguientes:

- wage: salario promedio por hora.
- educ: años de educación.
- exper: años de experiencia potencial.
- tenure: años con el empleador actual (antigüedad).
- nonwhite: es igual a 1 si la persona no es blanca, 0 si no.
- female: es igual a 1 si la persona es mujer, 0 si no
- married: es igual a 1 si la persona es casada, 0 si no.

En primer lugar queremos cambiar el nombre de la variable que está en la posición 4:

```
names(base)[4] <- "antigüedad"
names(base)[24] <- "antigüedadcuad"
```

Por ahora lo que nos interesaba es un subconjunto de variables, todas de la 1 a la 7 (la 4.ª ha quedado como antigüedad), la 22 (log. de salario), la 23 (experiencia al cuadrado) y la 24 (la antigüedad al cuadrado).

```
base1 <- base[,c(1: 7, 22:24)]
```

Veremos ahora las primeras filas

```
head(base1, n = 10)
```

```
##     wage educ exper antigüedad nonwhite female married    lwage expersq
## 1   3.10   11     2          0        0      1       0 1.131402       4
## 2   3.24   12    22          2        0      1       1 1.175573     484
## 3   3.00   11     2          0        0      0       0 1.098612       4
## 4   6.00    8    44         28        0      0       1 1.791759    1936
## 5   5.30   12     7          2        0      0       1 1.667707      49
## 6   8.75   16     9          8        0      0       1 2.169054      81
## 7  11.25   18    15          7        0      0       0 2.420368     225
## 8   5.00   12     5          3        0      1       0 1.609438      25
## 9   3.60   12    26          4        0      1       0 1.280934     676
## 10 18.18   17    22         21        0      0       1 2.900322     484
##    antigüedadcuad
## 1               0
## 2               4
## 3               0
## 4             784
## 5               4
## 6              64
## 7              49
## 8               9
## 9              16
## 10            441
```

También podríamos llamarlos con el nombre de variable

```
datos1 <- base[,c("wage","educ","exper", "antigüedad" )]

head(datos1, n = 5)
```

```
##   wage educ exper antigüedad
## 1 3.10   11     2          0
## 2 3.24   12    22          2
## 3 3.00   11     2          0
## 4 6.00    8    44         28
## 5 5.30   12     7          2
```

o como cubrimos al tocar `tidyverse`, la función `select()`

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
```

```
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
colnames(base)
```

```
##  [1] "wage"           "educ"           "exper"          "antigüedad"
##  [5] "nonwhite"       "female"         "married"        "numdep"
##  [9] "smsa"           "northcen"       "south"          "west"
## [13] "construc"       "ndurman"        "trcommpu"       "trade"
## [17] "services"       "profserv"       "profocc"        "clerocc"
## [21] "servocc"        "lwage"          "expersq"        "antigüedadcuad"
```

```
datos2 <- base |>
  dplyr::select(wage, educ, exper, antigüedad) #¿qué pasa si no tengo la especificación?
head(datos2, n=7)
```

```
##    wage educ exper antigüedad
## 1  3.10   11     2          0
## 2  3.24   12    22          2
## 3  3.00   11     2          0
## 4  6.00    8    44         28
## 5  5.30   12     7          2
## 6  8.75   16     9          8
## 7 11.25   18    15          7
```

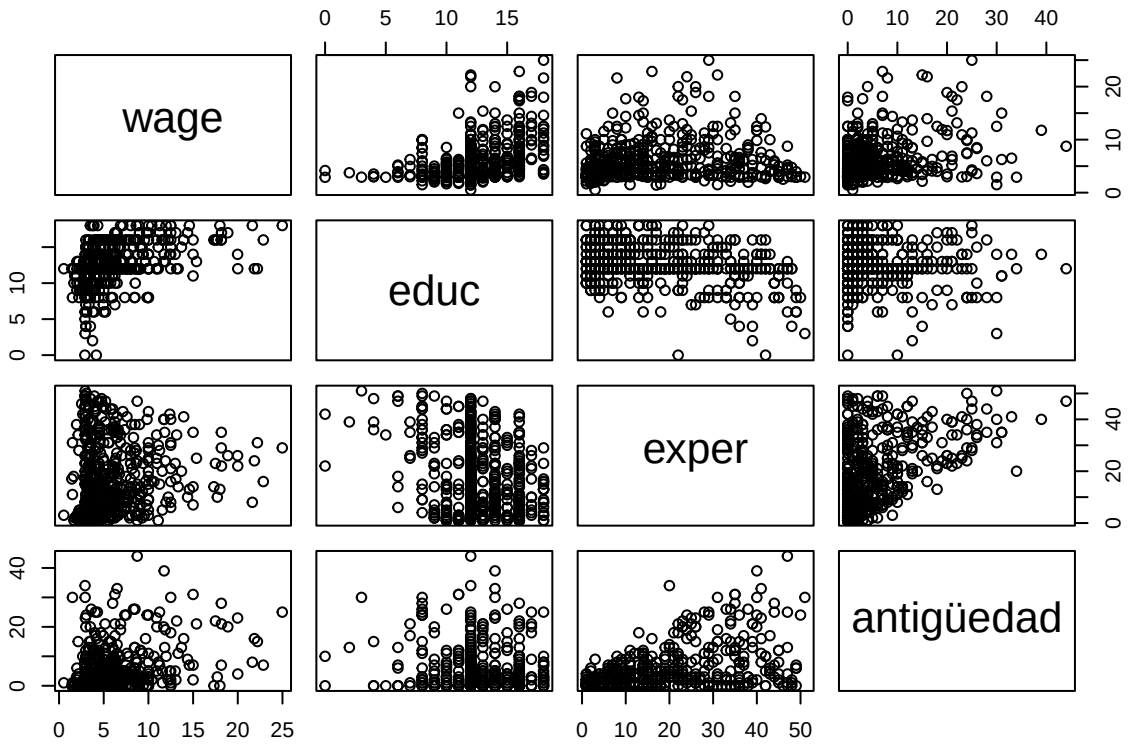
Si solamente quisiéramos los datos de los casados podríamos usar la función de filtrado

```
datos3 <- base |>
  dplyr::select("wage", "educ", "exper", "antigüedad", "married") |>
  filter(married == 1)
```

Correlaciones

Para investigar si hay correlación entre alguna de las variables se puede realizar un gráfico en el que se presenta la dispersión para cada par de variables.

```
plot(datos2)
```



Y también calcular la matriz de correlaciones de las variables que figuran en *datos2*.

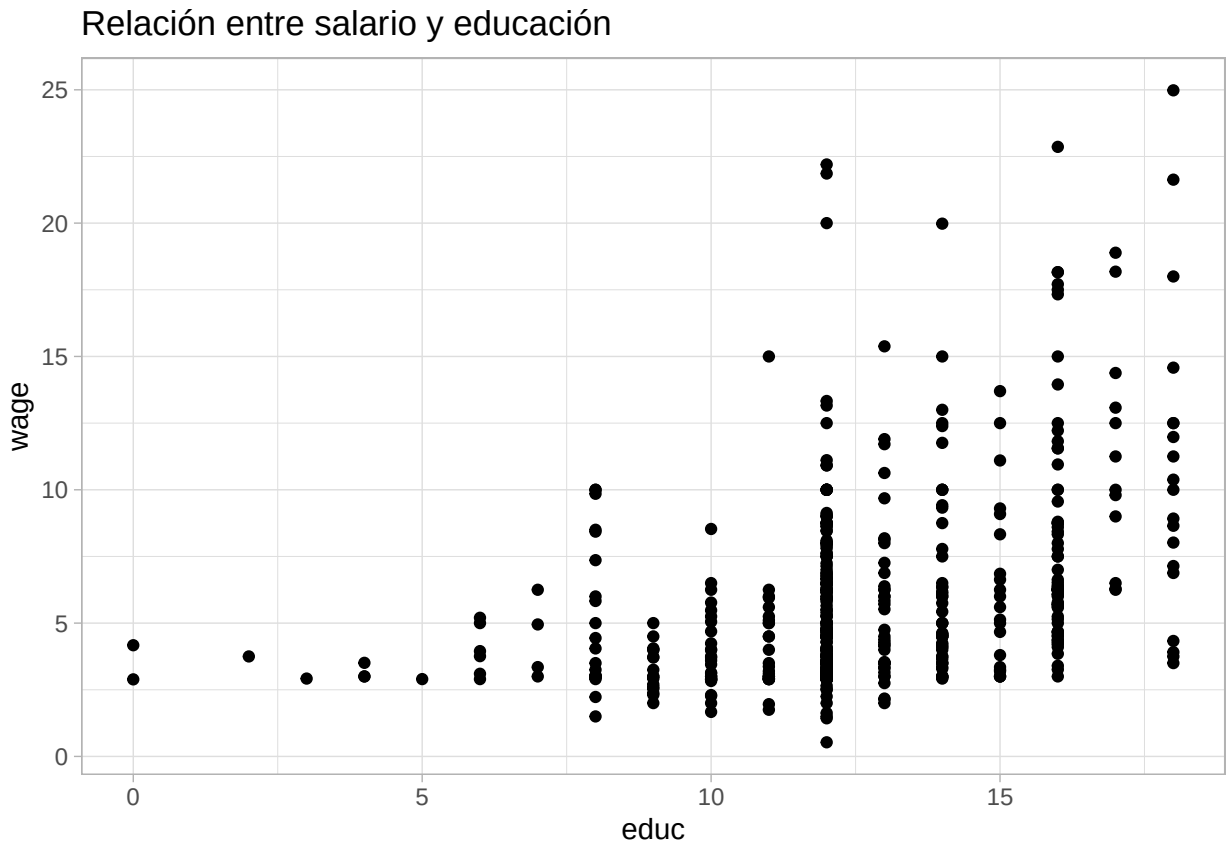
```
cor(datos2)
```

```
##                wage        educ       exper  antigüedad
## wage       1.0000000  0.40590333  0.1129034  0.34688957
## educ       0.4059033  1.00000000 -0.2995418 -0.05617257
## exper      0.1129034 -0.29954184  1.0000000  0.49929145
## antigüedad 0.3468896 -0.05617257  0.4992914  1.00000000
```

Notamos que existe una correlación positiva entre el salario y la educación (0.4059)

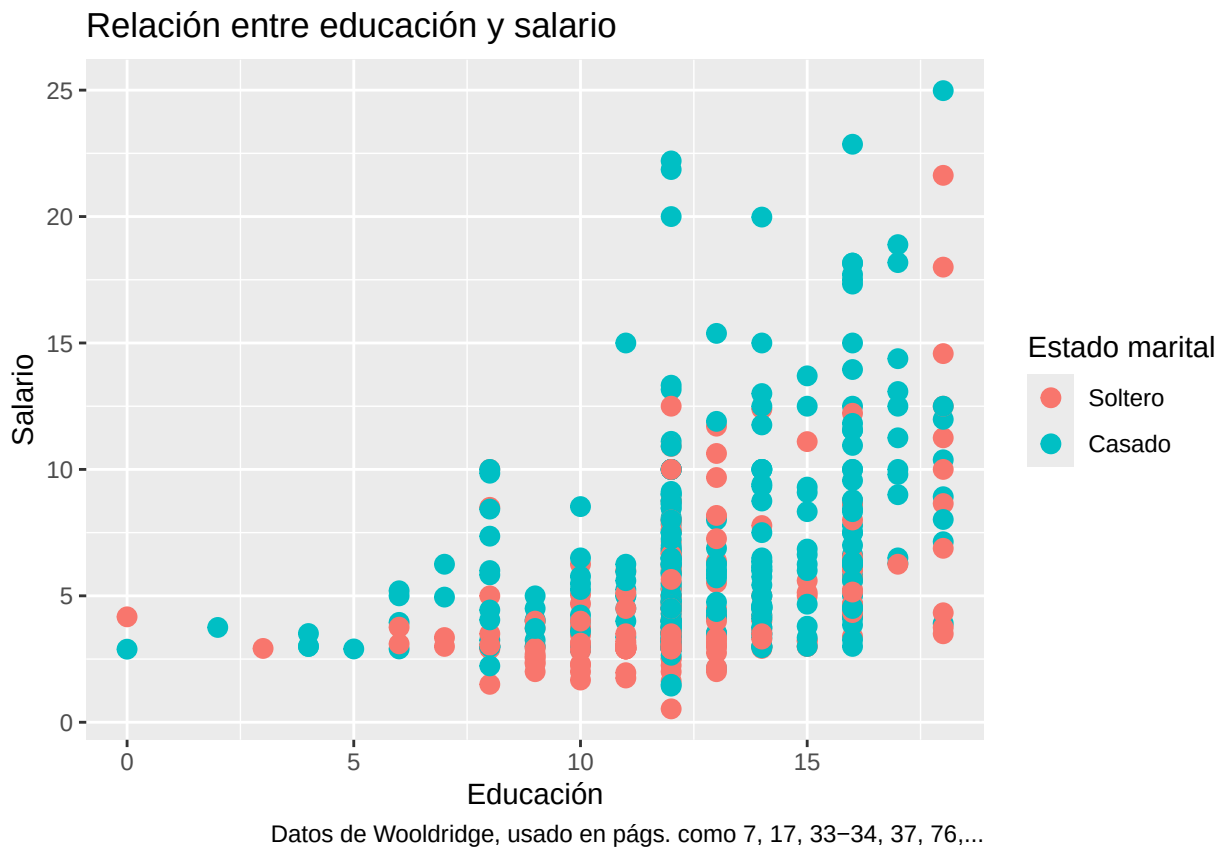
Crearemos el diagrama de dispersión entre salario y educación utilizando las funciones de la librería ggplot2.

```
ggplot(datos2, aes(x = educ, y = wage)) +
  geom_point() + theme_light() +
  ggtitle("Relación entre salario y educación")
```



Se acordarán que la vez anterior añadimos complicaciones como un nivel adicional en la capa de color:

```
wage1 |>
  mutate(marital = factor(married, levels = c(0, 1), labels = c("Soltero", "Casado"))) |>
  ggplot(aes(educ, wage)) +  geom_point(aes(colour = marital), size = 3)+
  labs(title="Relación entre educación y salario",
       x = "Educación",
       y = "Salario",
       color = "Estado marital",
       caption = "Datos de Wooldridge, usado en págs. como 7, 17, 33-34, 37, 76,...")
```



Esto lo exploraremos más al entrar en modelos de regresión lineal *múltiple*. Empezaremos por el modelo de regresión lineal *simple*.

Modelo de regresión lineal simple

Estimamos un modelo de regresión lineal simple, con el método mínimos cuadrados ordinarios (en adelante MCO, OLS en inglés) que explique los salarios en función de los años de educación de las personas. En R esto se hace con la función `lm()`:

```
mod1 <- lm(wage ~ educ, data=datos2)
summary(mod1) # para imprimir la salida
```

```
##
## Call:
## lm(formula = wage ~ educ, data = datos2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3396 -2.1501 -0.9674  1.1921 16.6085
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.90485    0.68497  -1.321    0.187
## educ         0.54136    0.05325  10.167   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.378 on 524 degrees of freedom
## Multiple R-squared:  0.1648, Adjusted R-squared:  0.1632
## F-statistic: 103.4 on 1 and 524 DF,  p-value: < 2.2e-16
```

¿Cómo se lee esta salida? La primera línea indica la fórmula que se utilizó. La segunda es sobre la distribución de residuos las diferencias entre los valores observados de `wage` y los valores predichos por el modelo. La mediana cercana a cero indica que los residuos están centrados alrededor de cero. El rango de los residuos sugiere que hay algunos valores atípicos, especialmente en el extremo máximo (16.6085), lo que podría indicar la presencia de salarios excepcionalmente altos no explicados completamente por el modelo.

Vamos a los coeficientes: la información se presenta a través de varias columnas. La columna de `Estimate` tiene el valor estimado de coeficientes ($\hat{\beta}$, mientras que la columna de `Std. Error` nos da en promedio lo que varía el estimado en relación a la variable dependiente. Esto es de utilidad para computar intervalos de confianza y establecer la métrica con la cual determinar la hipótesis la existencia de una relación entre una variable y otra. El puntaje t reporta la distancia estandarizada en distribución t de nuestro coeficiente, en relación a la posibilidad de que tuviera cero efecto. Mientras mayor sea el número de puntaje, y se mantuvieran relativamente mayores en relación al error estándar indica que una relación existe. La última columna provee el valor p, probabilidad asociada al puntaje t, que indica la probabilidad de estar viendo un valor tan extremo como el reportado por azar.

El intercepto de -0.9 representa el salario promedio cuando los años de educación son cero. Sin embargo, en la práctica, es poco común que una persona tenga cero años de educación (o que tenga ingreso negativo por un trabajo), por lo que este valor tiene una interpretación limitada. El valor p es algo elevado (es decir, supera los niveles utilizados normalmente como corte para evitar errores tipo 1), lo que indica que no podríamos afirmar con certeza su diferencia de cero. Por otro lado, el coeficiente en educación indica que cada año en educación (una unidad adicional) se traduciría en un aumento de 0.54136 unidades en salario (si son pesos, pues 54 chavos). El valor p es bajo (o el puntaje t es elevado, distante a 2, y mucho mayor que el error estándar), lo que indica con cierta certeza que el estimado no es cero. Vemos esto acompañado con asteriscos, simbolizando el nivel de significancia, atado al valor seleccionado α, el punto de corte anteriormente mencionado.

El modelo luego continúa reportando otros diagnósticos generales:

- `Residual standard error` (Error estándar de los residuos): más o menos 3.378
  - Indica la variabilidad promedio de los residuos; en otras palabras, mide la precisión del modelo. Los modelos lineales incluyen un término estocástico, que captura las desviaciones de nuestra relación predicha con las observaciones. En este caso, el valor de 3.378, sugiere que las predicciones individuales del salario pueden variar en promedio ± 3.378 unidades del valor real a través del espacio de nuestras observaciones.
- `Degrees of freedom` (Grados de libertad): 524 – Calculado como el número de observaciones menos el número de parámetros estimados (n - k). El objeto utilizado para análisis tenía 526 filas, y estimamos la pendiente y el intercepto.
- El coeficiente de determinación (R^2) nos da 0.1648, que indica que el 16.48% de la variabilidad se explica con el modelo: es decir una proporción de la varianza explicada. El *R cuadrado ajustado* es similar, pero penaliza la inclusión de términos adicionales, algo que el R^2 no hará: al seguir añadiendo variables, R^2 seguirá aumentando, sean buenas o no las adiciones; el R^2 *ajustado* es preferido para análisis en regresión múltiple.
- El estadístico F y el valor p global evalúa la hipótesis nula de que todos los coeficientes sean iguales a cero.

Al objeto creado para almacenar el modelo lineal le podemos hacer análisis varios. Por ejemplo, podríamos hacerle el análisis de varianzas. Este se calcula con la función `anova()`:

```
anova(mod1) # para imprimir el análisis de varianzas
```

```
## Analysis of Variance Table
##
## Response: wage
##            Df Sum Sq Mean Sq F value    Pr(>F)
## educ        1 1179.7 1179.73  103.36 < 2.2e-16 ***
## Residuals 524 5980.7   11.41
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Este análisis descompone la variabilidad total de la variable dependiente (`wage` en este caso) en componentes atribuibles al modelo y a los residuos. ¿Para qué haríamos esto? Quizás en este caso simple no parezca muy revelador (aunque da más detalles, por ejemplo, sobre el cómputo que lleva al puntaje F, viendo). Si tenemos variables categóricas con más de dos niveles, podremos aclarar más el impacto de la variable en general, en lugar de meramente entender la diferencia de distintos grupos en referencia a un valor base.

El objeto creado a través de la función `lm()`, de clase lm, tiene en sí doce componentes:

```
names(mod1)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

Se puede acceder a ellos como a las variables dentro de un objeto, utilizando el operador `$` entre el objeto y el elemento. Por ejemplo, para extraer los coeficientes estimados se escribe lo siguiente:

```
mod1$coefficients
```

```
## (Intercept)        educ
##  -0.9048516   0.5413593
```

```
coef(summary(mod1))
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -0.9048516 0.68496782 -1.321013 1.870735e-01
## educ         0.5413593 0.05324804 10.166746 2.782599e-22
```

Notemos que la función de `coef`, que extrae coeficientes de objetos creados por funciones de modelaje estadístico, nos retorna una matriz con cuatro columnas y la cantidad de filas adecuadas para los parámetros estimados, en este caso dos.

Una variación podría ser:

```
coefficients(mod1)
```

```
## (Intercept)        educ
##  -0.9048516   0.5413593
```

```
summary(mod1)$coefficients #noten, es igual en resultado a la función de coef(summary(modelo))
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -0.9048516 0.68496782 -1.321013 1.870735e-01
## educ         0.5413593 0.05324804 10.166746 2.782599e-22
```

Para consultar la estimación de un coeficiente de regresión se utilizan *corchetes [ ]* y se indica la ubicación del mismo dentro de la salida del `summary()`. Por ejemplo, al alfa gorro o sombrerito (es decir, $\hat{\alpha}$, alfa estimado) se podría sacar llamando a la *[primera fila, primera columna]*:

```
ahat <- coef(summary(mod1))[1,1]
ahat
```

```
## [1] -0.9048516
```

Y el beta gorro $\hat{\beta}$ o beta estimado se podría obtener llamando a la *[segunda fila, primera columna]*:

```
bhat <- coef(summary(mod1))[2,1]
bhat
```

```
## [1] 0.5413593
```

Del modelo podemos sacar también:

```
coeficientes <- mod1$coefficients #vector de coeficientes estimados
ygorro <- mod1$fitted.values #valores predichos
resid <- mod1$residuals #residuos
#los valores estimados o predichos también se pueden sacar con la función predict():
ygor1<- predict(mod1)
head(ygor1)
```

```
##        1        2        3        4        5        6
## 5.050100 5.591459 5.050100 3.426022 5.591459 7.756896
```

```
head(ygorro)
```

```
##        1        2        3        4        5        6
## 5.050100 5.591459 5.050100 3.426022 5.591459 7.756896
```

```
all.equal(ygor1, ygorro)
```

```
## [1] TRUE
```

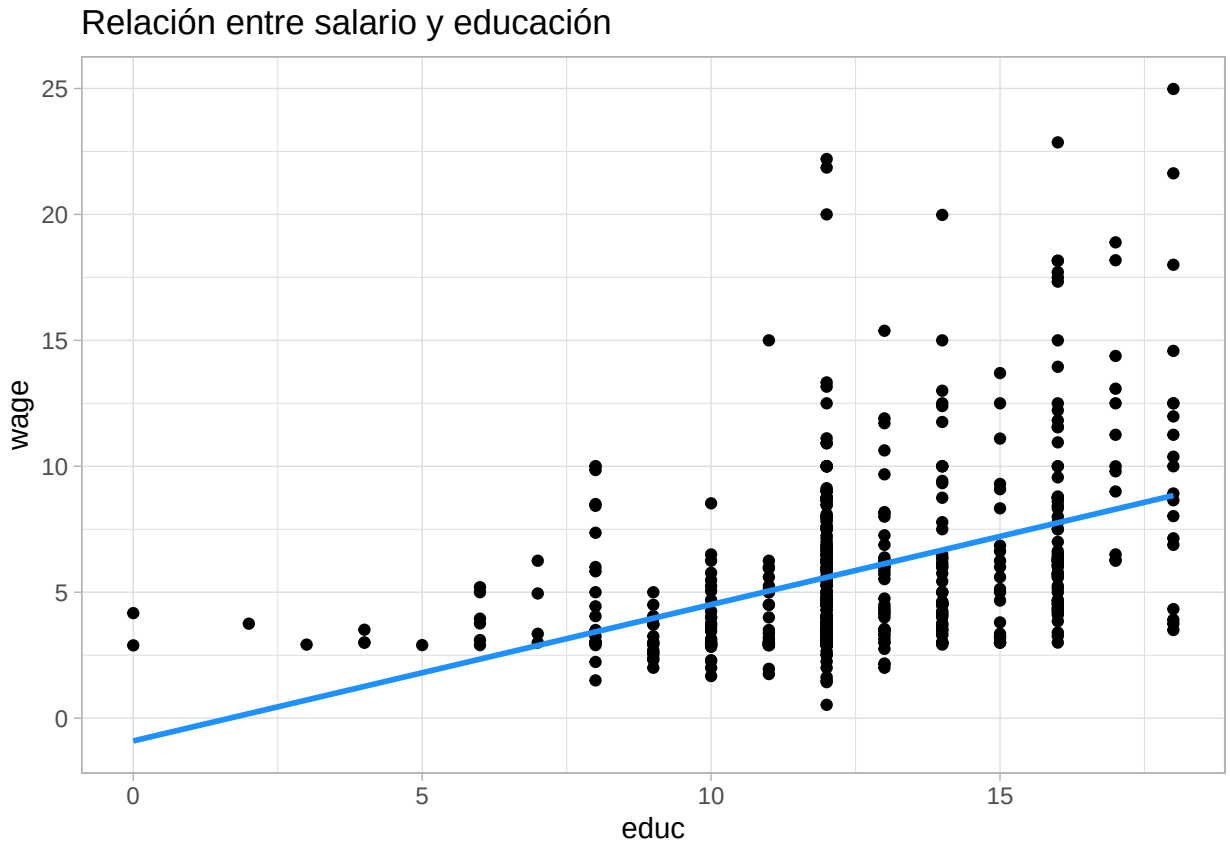
Agregaremos al conjunto de datos original, las predicciones, en este caso salario estimado, con la función `predict()`.

```r
datos2$predicciones <- predict(mod1)
head(datos2, 6)
```

```
##   wage educ exper antigüedad predicciones
## 1 3.10   11     2          0     5.050100
## 2 3.24   12    22          2     5.591459
## 3 3.00   11     2          0     5.050100
## 4 6.00    8    44         28     3.426022
## 5 5.30   12     7          2     5.591459
## 6 8.75   16     9          8     7.756896
```

El gráfico de dispersión puede establecerse

```r
ggplot(datos2, aes(x = educ, y = wage)) +
  geom_point() +
  geom_smooth(method = 'lm', formula = y ~ x, se = FALSE, col = 'dodgerblue1') +
  theme_light() +
  ggtitle("Relación entre salario y educación")
```



Si quisiéramos un gráfico de dispersión interactivo, podemos usar plotly. Así, posicionándose encima de cada observación, se ven los valores de (x, y) para cada uno de los individuos. Para construir dicho gráfico se necesita la función ggplotly() del paquete plotly.

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
ggplotly(data = datos2, x = ~ educ, y = ~ wage)
```

```
## Google Chrome was not found. Try setting the `CHROMOTE_CHROME` environment variable to the executable

## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please
```

Supuestos de modelos lineales

Ahora, a la hora de evaluar si nuestro modelo es bueno, además de los estimados que vimos sobre el ajuste del modelo o su capacidad explicativa de la varianza, tenemos que revisar las asunciones o presunciones que hace un modelo sobre los datos que evalúa. En la evaluación de un modelo lineal, tenemos cinco.

1. **Linealidad**: La relación entre las variables dependiente (y) e independiente(s) (x) debe ser lineal en los parámetros. Esto significa que el modelo es lineal en los coeficientes, aunque las variables en sí mismas puedan estar transformadas (por ejemplo, mediante logaritmos).
2. **Independencia de los errores**: Los residuos o errores (la diferencia entre los valores observados y los valores predichos por el modelo) deben ser independientes entre sí. Esto significa que no debe haber correlación entre los errores.
3. **Homoscedasticidad**: La varianza de los errores debe ser constante a lo largo de todos los valores de las variables independientes. Esto implica que la dispersión de los residuos debe ser más o menos la misma a lo largo del rango de valores de la variable independiente. En caso de heteroscedasticiddad, las estimaciones se tornan ineficientes y los errores estándar incorrectos.
4. **Normalidad de los errores**: Los errores deben seguir una distribución normal. Este supuesto es importante para la realización de pruebas de hipótesis y la construcción de intervalos de confianza. Cabe señalar que en este caso, los estimadores de los coeficientes siguen siendo insesgados incluso si los errores no fueran normales. La falta de normalidad afecta principalmente la inferencia estadística.
5. **No multicolinealidad**: Las variables independientes no deben estar altamente correlacionadas entre sí. La multicolinealidad puede dificultar la estimación precisa de los coeficientes de regresión.

De violarse estas presunciones, el modelo estará sesgado y sus resultados no serán del todo fiables.

Podremos revisar algunos de estos a través de varios métodos: Podemos por ejemplo calcular los residuos del modelo simple y los agregamos al conjunto de datos (*datos2*) de la siguiente forma:

```
datos2$residuos <- datos2$wage - datos2$predicciones

head(datos2, 5)
```

```
##   wage educ exper antigüedad predicciones   residuos
## 1 3.10   11     2          0     5.050100 -1.9501003
## 2 3.24   12    22          2     5.591459 -2.3514594
## 3 3.00   11     2          0     5.050100 -2.0501002
## 4 6.00    8    44         28     3.426022  2.5739776
## 5 5.30   12     7          2     5.591459 -0.2914593
```

Esto es lo mismo que R hace internamente, y se puede llamar con la función `residuals`:

```
datos2$residuosmod <- residuals(mod1)
head(datos2, 5)
```

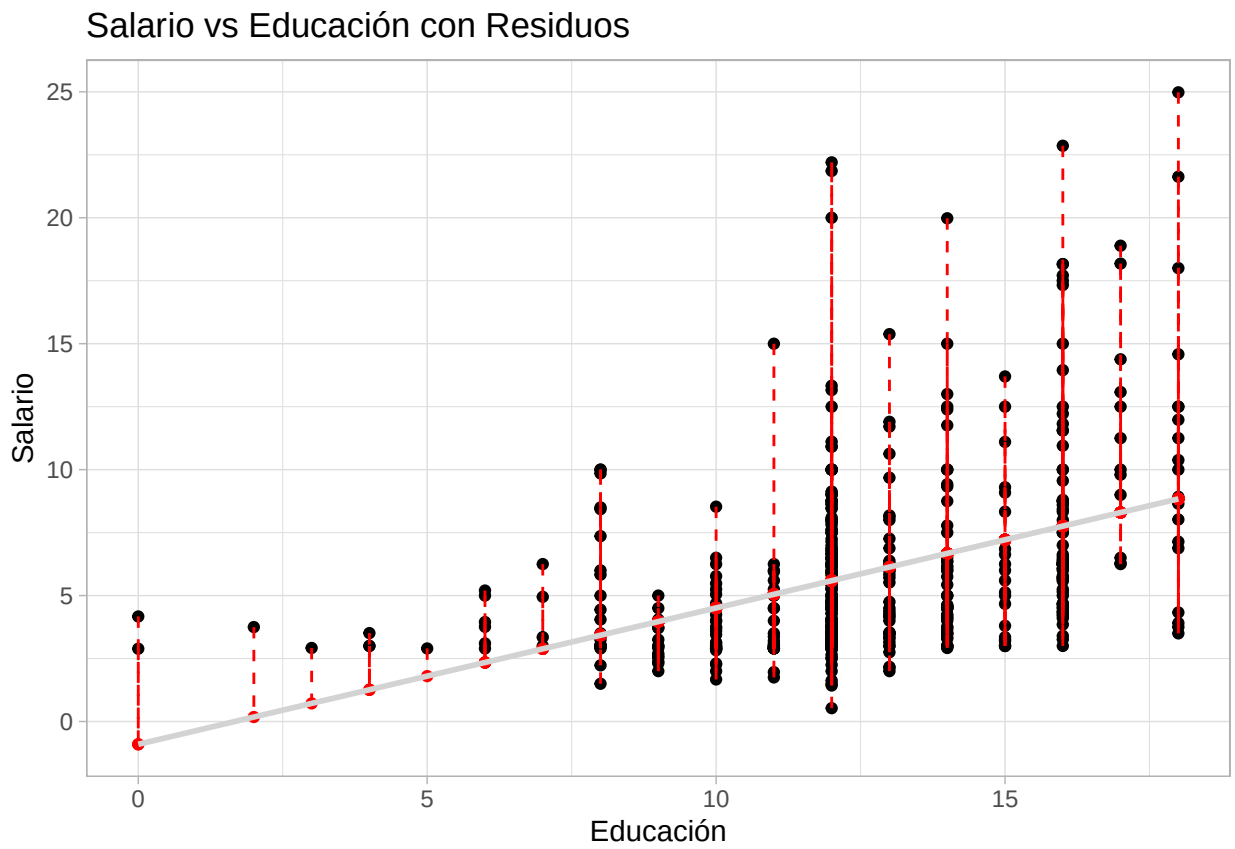
```
##   wage educ exper antigüedad predicciones   residuos residuosmod
## 1 3.10   11     2          0     5.050100 -1.9501003  -1.9501003
## 2 3.24   12    22          2     5.591459 -2.3514594  -2.3514594
## 3 3.00   11     2          0     5.050100 -2.0501002  -2.0501002
## 4 6.00    8    44         28     3.426022  2.5739776   2.5739776
## 5 5.30   12     7          2     5.591459 -0.2914593  -0.2914593
```

Comparamos los residuos calculados manualmente con los que nos dio el modelo. Otra forma de extraerlos es llamando con acceso `$` en el objeto a los residuos. `mod1$residuals` es equivalente a `residuals(mod1)`. Sin embargo, utilizar la función `residuals()` es generalmente preferible porque es más compatible con diferentes tipos de modelos y objetos en R.

Añadimos al gráfico el elemento de los valores estimados de y, $\hat{y}_i$, en rojo y muestro los residuos $\hat{\varepsilon}_i$:

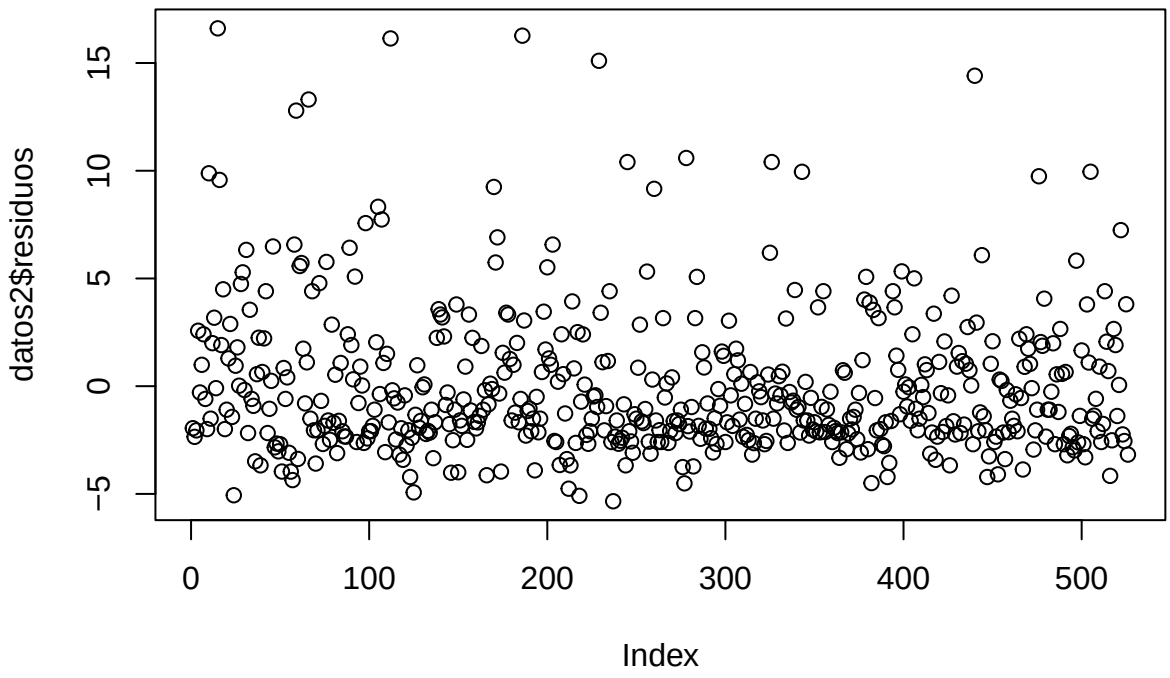
```
ggplot(datos2, aes(x = educ, y = wage)) +
  geom_point() +
  geom_segment(aes(xend = educ, yend = predicciones), color = 'red', linetype = 'dashed') +
  geom_point(aes(y = predicciones), color = 'red') +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey") +
  theme_light() +
  labs(title = "Salario vs Educación con Residuos", x = "Educación", y = "Salario")
```

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



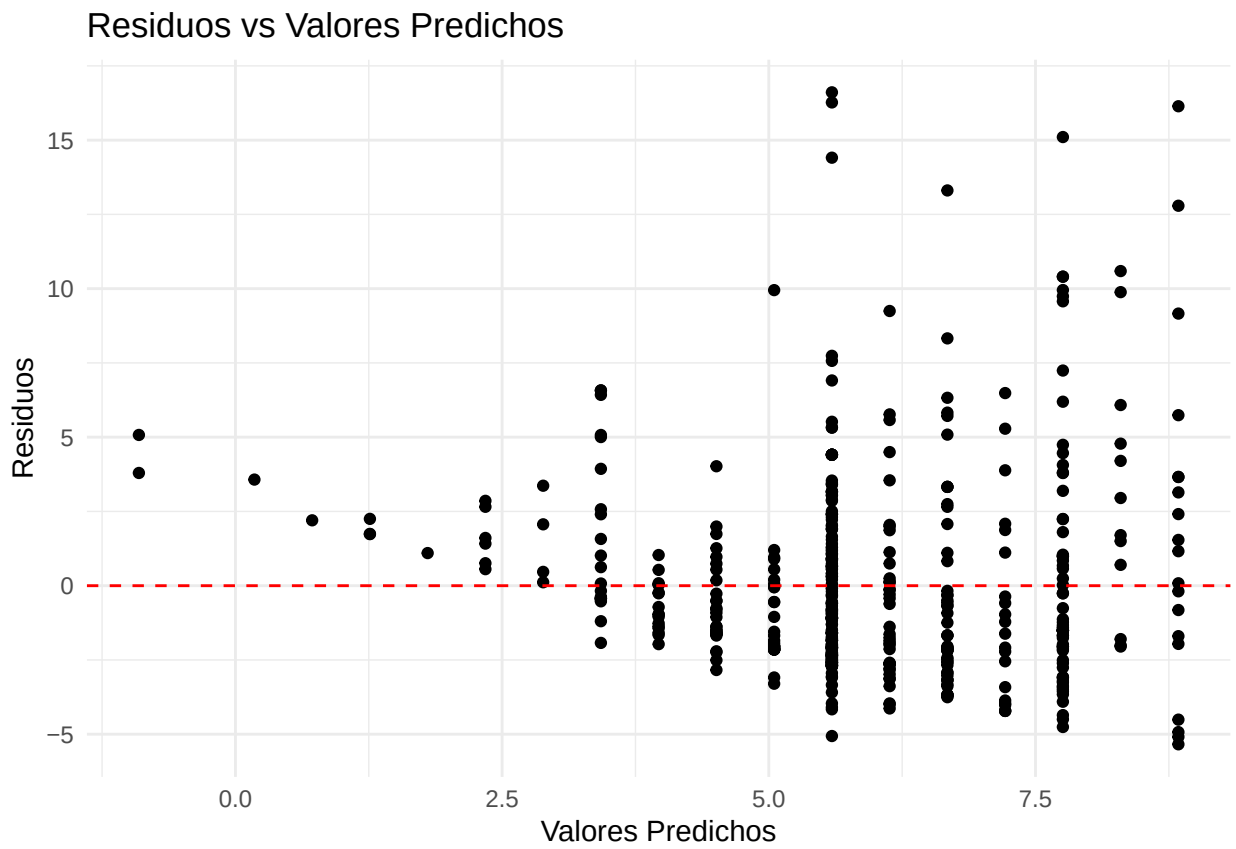
Podemos realizar un gráfico de dispersión para inspeccionar de forma gráfica los residuos.

```
plot(datos2$residuos)
```



El comando `plot(datos2$residuos)` simplemente grafica los residuos en función de su índice, lo cual puede no ser muy informativo (aunque ciertamente se aprecia algún patrón aglomerándose mucho por debajo de cero, y dispersándose hacia valores más altos). Sería más útil graficar los residuos contra los valores predichos o contra la variable independiente para detectar patrones que indiquen violaciones de los supuestos del modelo.

```
ggplot(datos2, aes(x = predicciones, y = residuos)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  theme_minimal() +
  labs(title = "Residuos vs Valores Predichos", x = "Valores Predichos", y = "Residuos")
```

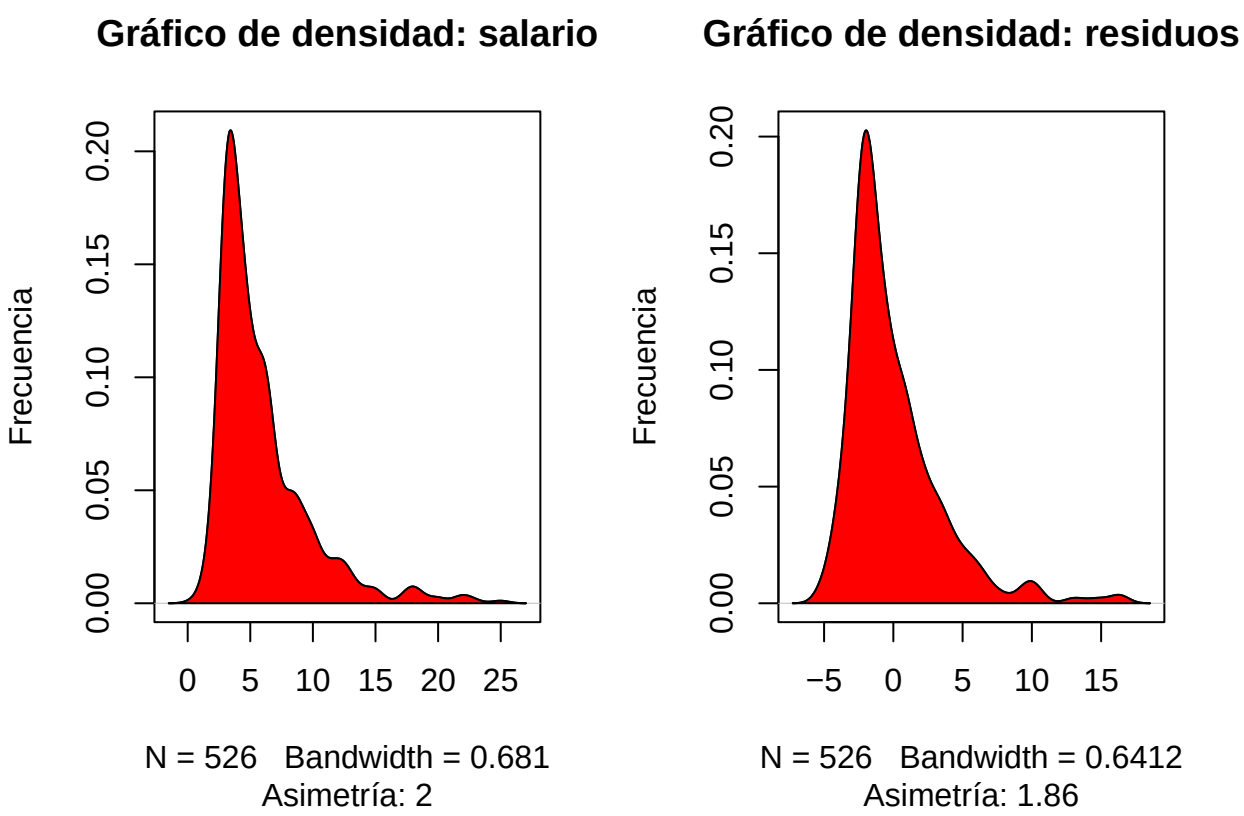


Hasta ahora vemos problemas en la dispersión de los residuos, pues no parecen estar dispersos como ruido alrededor de cero. Podríamos mejorar esto algo al ver la asimetría (skewness) de la distribución

```
library(e1071)  # para la función skewness
```

```
##
## Attaching package: 'e1071'

## The following object is masked from 'package:ggplot2':
##
##     element
```

```
par(mfrow = c(1, 2))  # divide el área de gráficos en 2 columnas

plot(density(datos2$wage), main = "Gráfico de densidad: salario", ylab = "Frecuencia", sub = paste("Asi

polygon(density(datos2$wage), col = "red")

plot(density(datos2$residuos), main = "Gráfico de densidad: residuos", ylab = "Frecuencia", sub = paste

polygon(density(datos2$residuos), col = "red")
```



Volveremos más tarde con la resolución de este problema, pero por ahora continuaremos con este modelo tal cual para mostrar otros elementos útiles.

Predicciones con estimados

Generaremos la predicción puntual y el intervalo de confianza para una educación promedio en años.

```
mean(datos2$educ)
```

```
## [1] 12.56274
```

Primero, calculamos el salario esperado para una persona con la educación promedio que es 12.56 años: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}$.

```
sal_pred <- ahat + bhat * 12.56274
sal_pred
```

```
## [1] 5.896104
```

La función predict() se puede igual usar para obtener la predicción puntual:

```
nuevo <- data.frame(educ = mean(datos2$educ))
sal_pred <- predict(mod1, newdata = nuevo)
sal_pred
```

```
##        1
## 5.896103
```

Ahora, calculamos la predicción y los intervalos para una persona con 15 años de educación para ilustrar cómo varían estos valores con diferentes niveles de educación. Entonces ($x = 15$). El argumento `interval = prediction` devuelve el valor para la predicción puntual junto a su intervalo de predicción, que estima el rango en el cual caerá una nueva observación individual con un cierto nivel de confianza.

```
nuevo <- data.frame(educ = 15)
future_y <- predict(object = mod1, newdata = nuevo, interval = "prediction", level = 0.95)
future_y
```

```
##        fit       lwr      upr
## 1 7.215537 0.5674896 13.86358
```

Si desean obtener solo la predicción puntual, pueden omitir el argumento `interval`.

Luego, generamos el intervalo de confianza para $E(y \mid x)$. en este caso, debemos cambiar el argumento a `interval = "confidence"`. El intervalo de confianza es para la media esperada de la variable dependiente dado un valor específico de la independiente, aquí 15 años de educación.

```
future_esp_y <- predict(object = mod1, newdata = nuevo, interval = "confidence", level = 0.95)
future_esp_y <- as.data.frame(future_esp_y)

IC_inf_esp_y <- future_esp_y$lwr
IC_sup_esp_y <- future_esp_y$upr
```

Agregando los pedazos

```
# Calcular intervalos de predicción para todas las observaciones
future_y_all <- predict(object = mod1, newdata = datos2, interval = "prediction", level = 0.95)
future_y_all <- as.data.frame(future_y_all)

# Calcular intervalos de confianza para todas las observaciones
future_esp_y_all <- predict(object = mod1, newdata = datos2, interval = "confidence", level = 0.95)
future_esp_y_all <- as.data.frame(future_esp_y_all)

# Combinar todos los datos en un solo data frame
nuevos_datos <- cbind(datos2, future_y_all,
                      IC_inf_esp_y = future_esp_y_all$lwr,
                      IC_sup_esp_y = future_esp_y_all$upr)
```

Finalmente, generamos los gráficos correspondientes con los intervalos de confianza para la predicción puntual (`IC_y`) y para el valor esperado (`IC_esp_y`) con el siguiente código:

```
IC_y <- ggplot(nuevos_datos, aes(x = educ, y = wage)) +
  geom_point() +
  geom_line(aes(y = lwr), color = "red", linetype = "dashed") +
  geom_line(aes(y = upr), color = "red", linetype = "dashed") +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, level = 0.95, col = 'blue', fill = 'pink2') +
  theme_light() +
  ggtitle("Predicción de y al 95%")
```

```
IC_esp_y <- ggplot(nuevos_datos, aes(x = educ, y = wage)) +
  geom_point() +
  geom_line(aes(y = IC_inf_esp_y), color = "blue", linetype = "dashed") +
  geom_line(aes(y = IC_sup_esp_y), color = "blue", linetype = "dashed") +
  geom_smooth(method = lm, formula = y ~ x, se = FALSE, col = 'blue') +
  theme_light() +
  ggtitle("Intervalo de Confianza de E(y|x) al 95%")
```

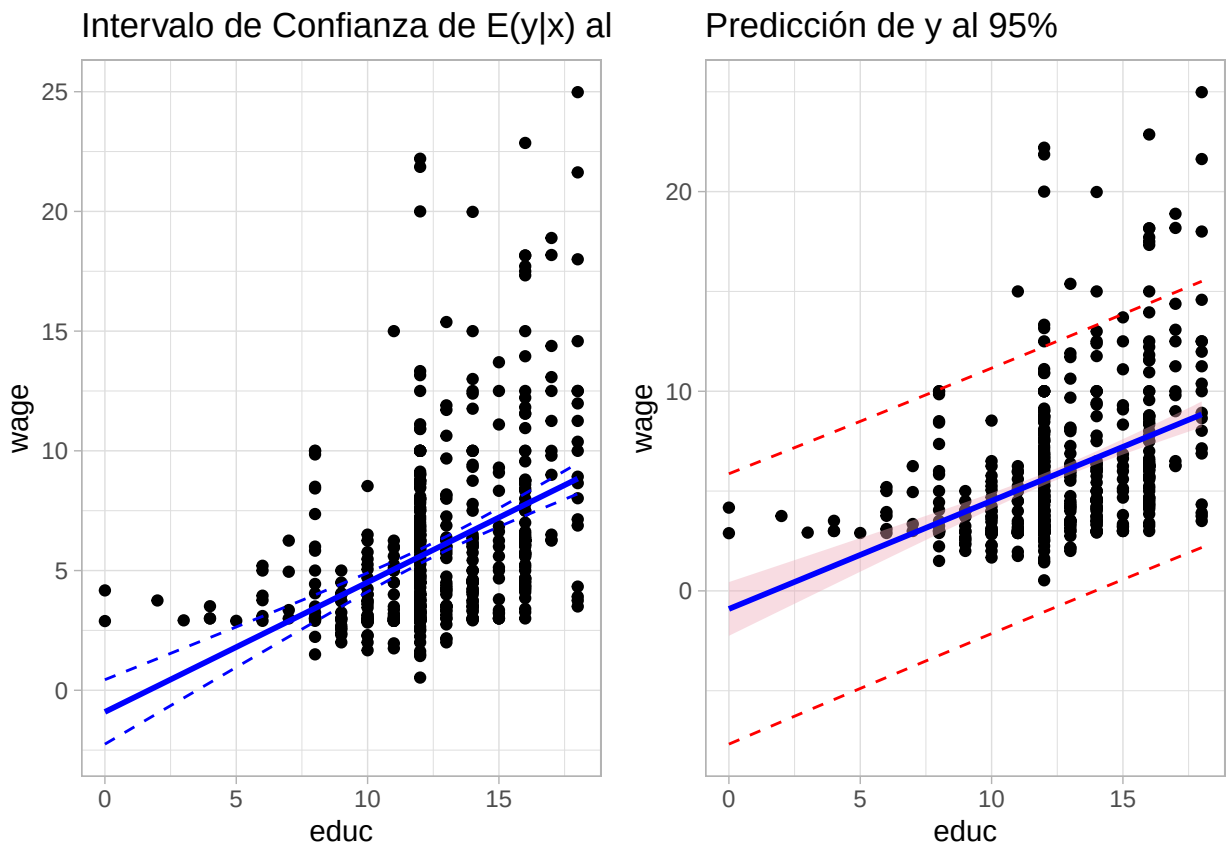
Imprimimos los gráficos uno al lado del otro, para poder compararlos mejor. ¿Cuál de los dos tiene mayor amplitud?

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(IC_esp_y, IC_y, ncol = 2, nrow = 1)
```

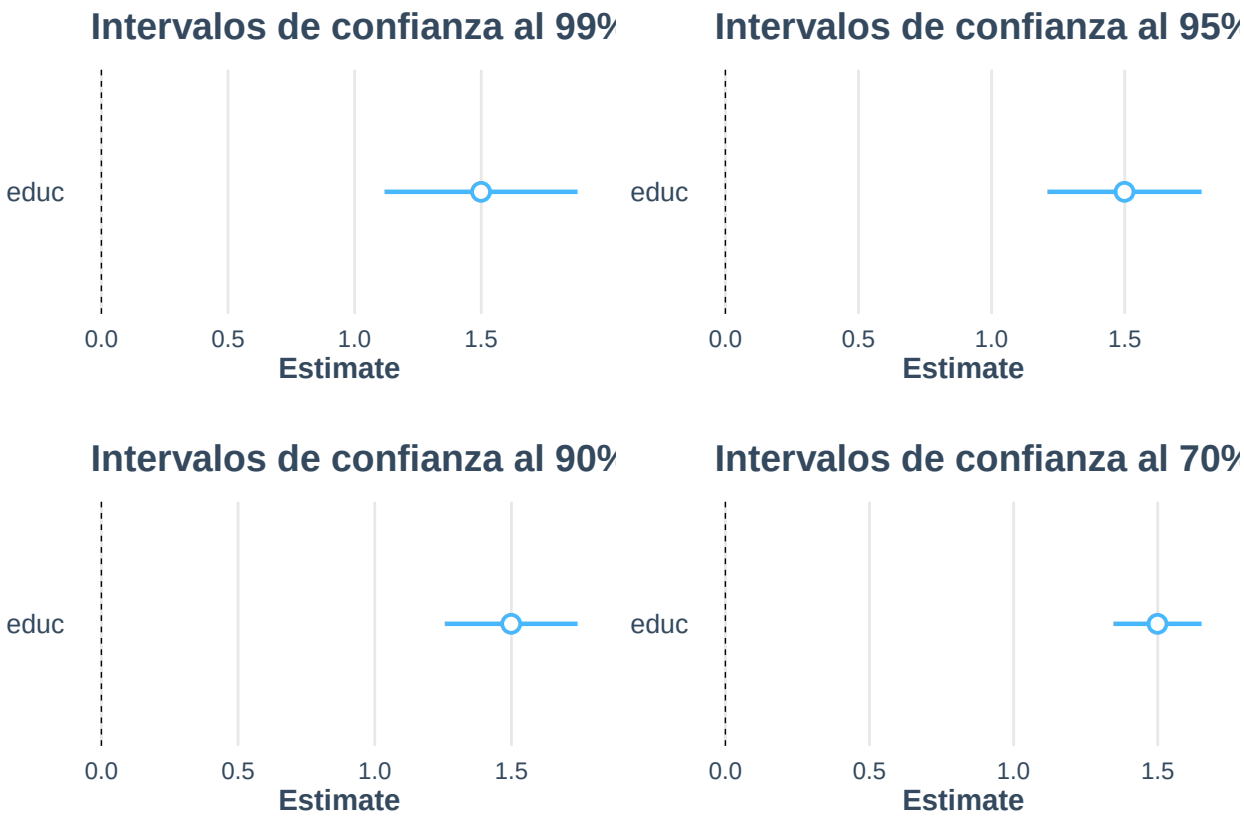


Al comparar ambos gráficos, observamos que el intervalo de predicción es más amplio que el intervalo de confianza. Esto es esperado, ya que el intervalo de predicción considera la variabilidad adicional de las

observaciones individuales, mientras que el intervalo de confianza se centra únicamente en la precisión de la estimación del valor medio esperado.

Veamos, a través de la comparación de dos gráficos el impacto que tiene el nivel de confianza en la amplitud de los intervalos. Para ello, tendremos que descargar e instalar algunas librerías. Se presenta primero el código y luego los gráficos obtenidos.

```
# Instalar y cargar las librerías necesarias si no las tiene
#install.packages("jtools")
#install.packages("gridExtra")
library(jtools)
library(gridExtra)

# Crear los gráficos con diferentes niveles de confianza
a <- plot_summs(mod1, scale = TRUE, plot.distributions = FALSE, ci_level = 0.99) +
  ggtitle("Intervalos de confianza al 99%")

b <- plot_summs(mod1, scale = TRUE, plot.distributions = FALSE, ci_level = 0.95) +
  ggtitle("Intervalos de confianza al 95%")

c <- plot_summs(mod1, scale = TRUE, plot.distributions = FALSE, ci_level = 0.90) +
  ggtitle("Intervalos de confianza al 90%")

d <- plot_summs(mod1, scale = TRUE, plot.distributions = FALSE, ci_level = 0.70) +
  ggtitle("Intervalos de confianza al 70%")

# Mostrar los gráficos uno debajo del otro
grid.arrange(a, b, c,d, ncol = 2, nrow = 2)
```



Al comparar los gráficos, observamos que un nivel de confianza más alto (99%) produce intervalos más amplios, ya que estamos buscando abarcar una mayor proporción de posibles valores verdaderos de los coeficientes. Por el contrario, ir bajando hacia un nivel de confianza menor (70%) resulta en intervalos más estrechos.

Regresión múltiple

En esta sección, queremos evaluar cómo varias variables independientes se relacionan con una variable dependiente. Específicamente, analizaremos cómo la educación y la antigüedad influyen en el salario. Luego añadiremos otras variables adicionales, como señaláramos más temprano:

- wage: salario promedio por hora.
- educ: años de educación.
- exper: años de experiencia potencial.
- tenure: años con el empleador actual (antigüedad).
- nonwhite: es igual a 1 si la persona no es blanca, 0 si no.
- female: es igual a 1 si la persona es mujer, 0 si no
- married: es igual a 1 si la persona es casada, 0 si no.

Digamos que queremos evaluar la relación de varias variables con la dependiente. Podemos calcular las correlaciones de los pares de variables, indicando que queremos trabajar con 3 decimales:

```r
round(cor(base1, method = "pearson"), 3)
```

```
##                  wage   educ  exper antigüedad nonwhite female married  lwage
## wage            1.000  0.406  0.113      0.347   -0.039 -0.340   0.229  0.937
## educ            0.406  1.000 -0.300     -0.056   -0.085 -0.085   0.069  0.431
## exper           0.113 -0.300  1.000      0.499    0.014 -0.042   0.317  0.111
## antigüedad      0.347 -0.056  0.499      1.000    0.012 -0.198   0.240  0.326
## nonwhite       -0.039 -0.085  0.014      0.012    1.000 -0.011  -0.062 -0.039
## female         -0.340 -0.085 -0.042     -0.198   -0.011  1.000  -0.166 -0.374
## married         0.229  0.069  0.317      0.240   -0.062 -0.166   1.000  0.271
## lwage           0.937  0.431  0.111      0.326   -0.039 -0.374   0.271  1.000
## expersq         0.030 -0.331  0.961      0.459    0.009 -0.028   0.217  0.023
## antigüedadcuad  0.267 -0.069  0.423      0.922   -0.007 -0.176   0.167  0.236
##                expersq antigüedadcuad
## wage             0.030          0.267
## educ            -0.331         -0.069
## exper            0.961          0.423
## antigüedad       0.459          0.922
## nonwhite         0.009         -0.007
## female          -0.028         -0.176
## married          0.217          0.167
## lwage            0.023          0.236
## expersq          1.000          0.414
## antigüedadcuad   0.414          1.000
```

Antes de estimar el modelo de regresión múltiple, es útil explorar las relaciones entre las variables mediante la matriz de correlaciones. Esto nos ayuda a detectar posibles problemas de multicolinealidad y a entender las relaciones bivariadas. En este caso, haremos una restricción a tres variables, ayudando a mejorar la legibilidad.

```r
# Seleccionamos las variables de interés
variables_interes <- base1[, c("wage", "educ", "antigüedad")]

# Calculamos la matriz de correlaciones y redondeamos a 3 decimales
matriz_correlaciones <- round(cor(variables_interes, method = "pearson"), 3)
matriz_correlaciones
```

```
##             wage   educ antigüedad
## wage       1.000  0.406      0.347
## educ       0.406  1.000     -0.056
## antigüedad 0.347 -0.056      1.000
```

Interpretación:

- Salario y Educación: Un coeficiente positivo indica que a mayor nivel educativo, el salario tiende a ser mayor. La fuerza de esta relación parece ser moderada, 0.406.
- Salario y Antigüedad: Un coeficiente positivo sugiere que con más años de antigüedad, el salario también aumenta. Parece ser de una fuerza similar a educación.
- Educación y Antigüedad: El coeficiente es negativo, y pequeño. Entre estas variables la relación no es clara en fuerza pero parecen ir en direcciones opuestas. Es menos probable que haya multicolinealidad entre estas variables independientes.

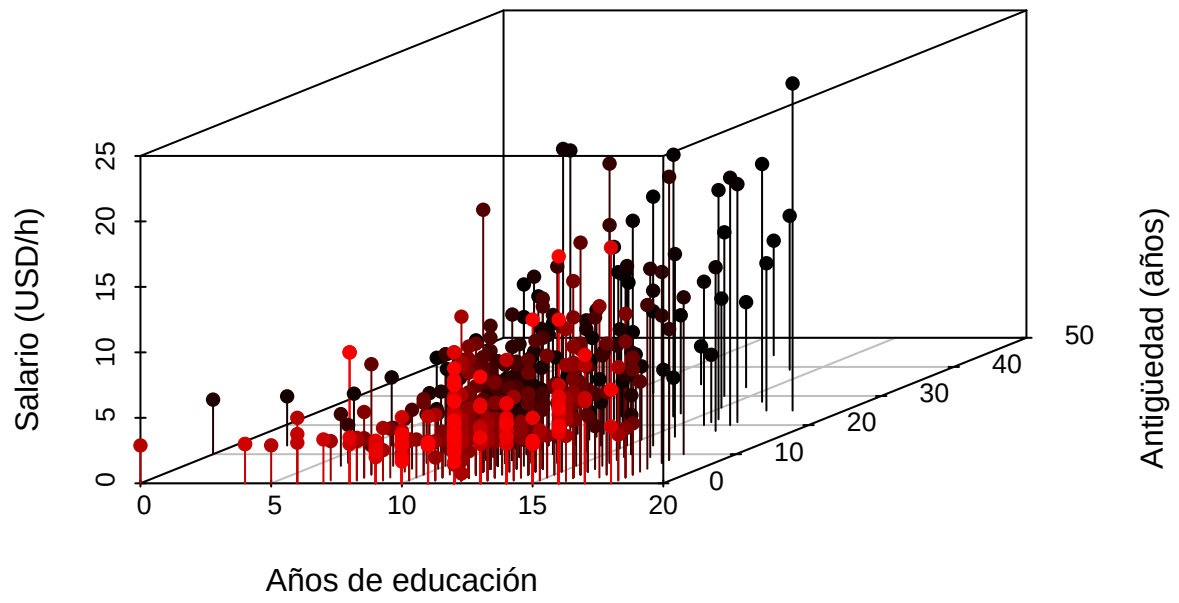
Para visualizar la relación conjunta entre las tres variables, utilizamos gráficos tridimensionales.

```
library(ggplot2)
library(plotly)
attach(base1)
plot_ly(x = educ, y = antigüedad, z = wage, type = "scatter3d", color = wage) |>
  layout(scene = list(xaxis = list(title = 'educación (en años)'),
                      yaxis = list(title = 'antigüedad (en años)'),
                      zaxis = list(title = 'Salario (en USD/h)')))
```

```
## No scatter3d mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```

Este gráfico interactivo permite rotar y explorar la relación entre las variables desde diferentes ángulos, facilitando la identificación de patrones. El próximo es menos interactivo, pero logra un efecto similar:

```
library(scatterplot3d)
graf <- scatterplot3d(x = educ, y = antigüedad, z = wage, pch = 16,
              cex.lab = 1, highlight.3d = TRUE, type = "h",
              xlab = 'Años de educación',
              ylab = 'Antigüedad (años)',
              zlab = 'Salario (USD/h)')
```



En este caso, las líneas verticales ayudan a visualizar la posición de cada punto en el espacio tridimensional.

Ahora, estimamos un modelo de regresión lineal múltiple para cuantificar el efecto de la educación y la antigüedad en el salario.

```
# Estimamos el modelo de regresión múltiple
mod2 <- lm(wage ~ educ + antigüedad, data = base1)

# Resumen del modelo
summary(mod2)
```

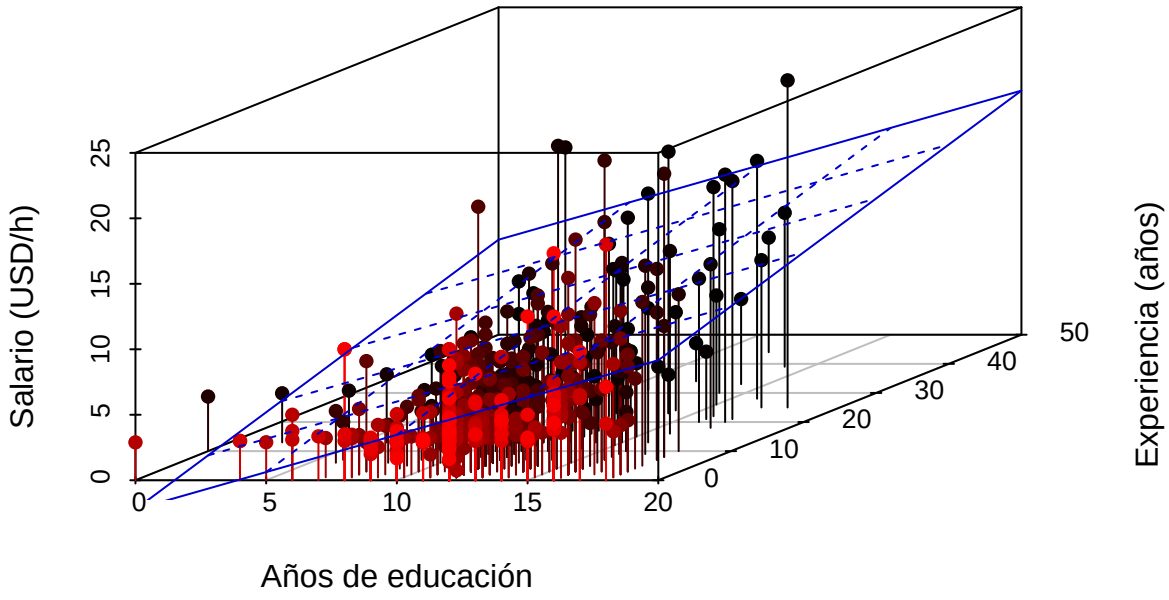
```
##
## Call:
## lm(formula = wage ~ educ + antigüedad, data = base1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1438 -1.7288 -0.6372  1.2575 14.7482
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.22162    0.64015   -3.47 0.000563 ***
## educ         0.56914    0.04881   11.66  < 2e-16 ***
## antigüedad   0.18958    0.01871   10.13  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.092 on 523 degrees of freedom
## Multiple R-squared:  0.3019, Adjusted R-squared:  0.2992
## F-statistic: 113.1 on 2 and 523 DF,  p-value: < 2.2e-16
```

La salida es similar al modelo anterior, con filas adicionales para cada coeficiente estimado adicional. El intercepto o constante representa el salario promedio cuando la educación y la antigüedad son cero (interpretación limitada en este contexto). Educ indica el cambio esperado en el salario por cada año adicional de educación, manteniendo constante la antigüedad en su valor esperado. Antigüedad muestra el cambio esperado en el salario por cada año adicional de antigüedad, manteniendo constante la educación. En este caso vemos que un año adicional de educación se traduce a unos 57 chavos adicionales en ingreso, mientras que antigüedad añade cerca de 19 chavos al ingreso.

Todos los coeficientes reportan valores t relativamente grandes en relación a sus errores estándar, y rechazamos en cada caso la hipótesis nula de no ser significativamente distinto a un efecto nulo. Notamos que tanto R^2 y R^2 *ajustado* han aumentado ambos, aunque la distancia entre ambos ahora es algo más notable: duplicamos la varianza explicada.

¿Cómo se ve esto? Añadimos el plano de regresión al gráfico 3D para visualizar cómo el modelo ajusta los datos.

```
graf <-  scatterplot3d(x = educ, y = antigüedad, z = wage, pch = 16,
                       cex.lab = 1, highlight.3d = TRUE, type = "h",
                       xlab = 'Años de educación',
                       ylab = 'Experiencia (años)',
                       zlab = 'Salario (USD/h)')

graf$plane3d(mod2, lty.box = "solid", col = 'mediumblue')
```



El plano representa las predicciones del modelo para diferentes combinaciones de educación y antigüedad. Podemos observar qué tan bien el plano ajusta a los datos reales, intentando cortar por un espacio vectorial estimado linealmente.

El ANOVA nos permite evaluar la significancia global del modelo y la contribución de cada variable independiente.

```
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: wage
##             Df Sum Sq Mean Sq F value    Pr(>F)
## educ         1 1179.7 1179.73  123.43 < 2.2e-16 ***
## antigüedad   1  981.7  981.72  102.71 < 2.2e-16 ***
## Residuals  523 4999.0    9.56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Sumas de Cuadrados (Sum Sq): Indican la variabilidad explicada por cada variable independiente y la variabilidad residual.

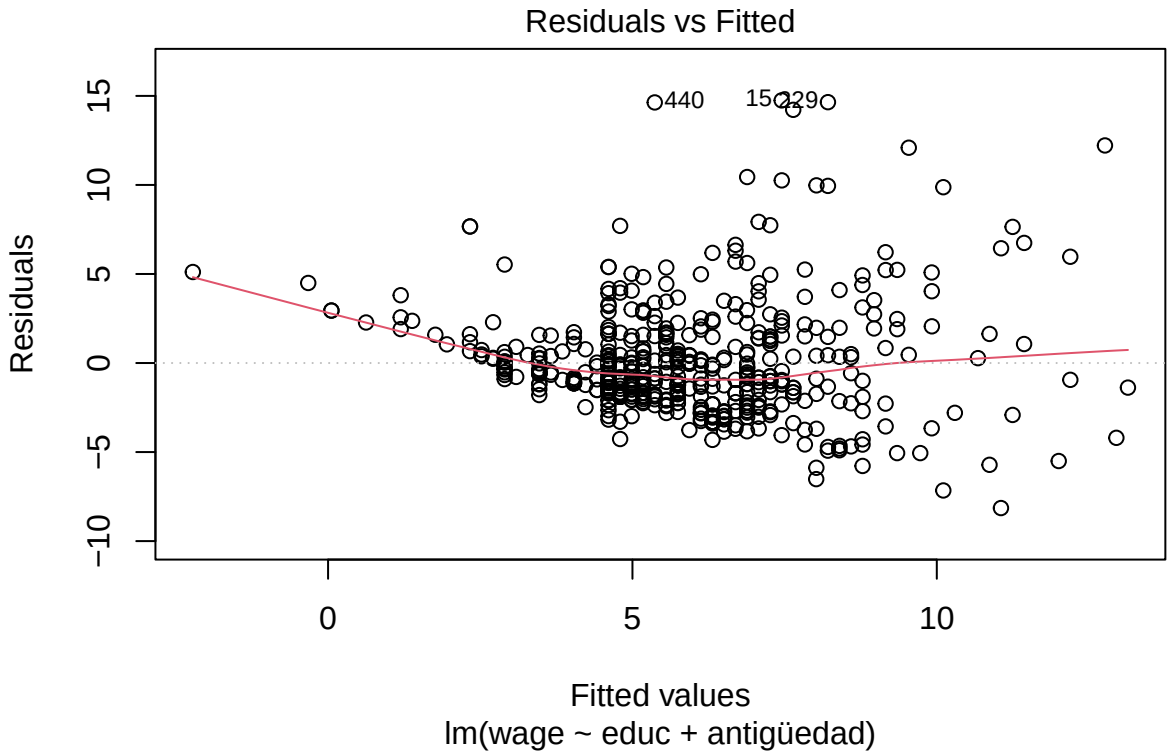
Para garantizar la validez de nuestro modelo de regresión múltiple, es fundamental verificar que se cumplen los supuestos básicos del modelo lineal. A continuación, revisaremos cada uno de estos supuestos en el orden mencionado anteriormente, aplicando pruebas diagnósticas y proporcionando interpretaciones detalladas.

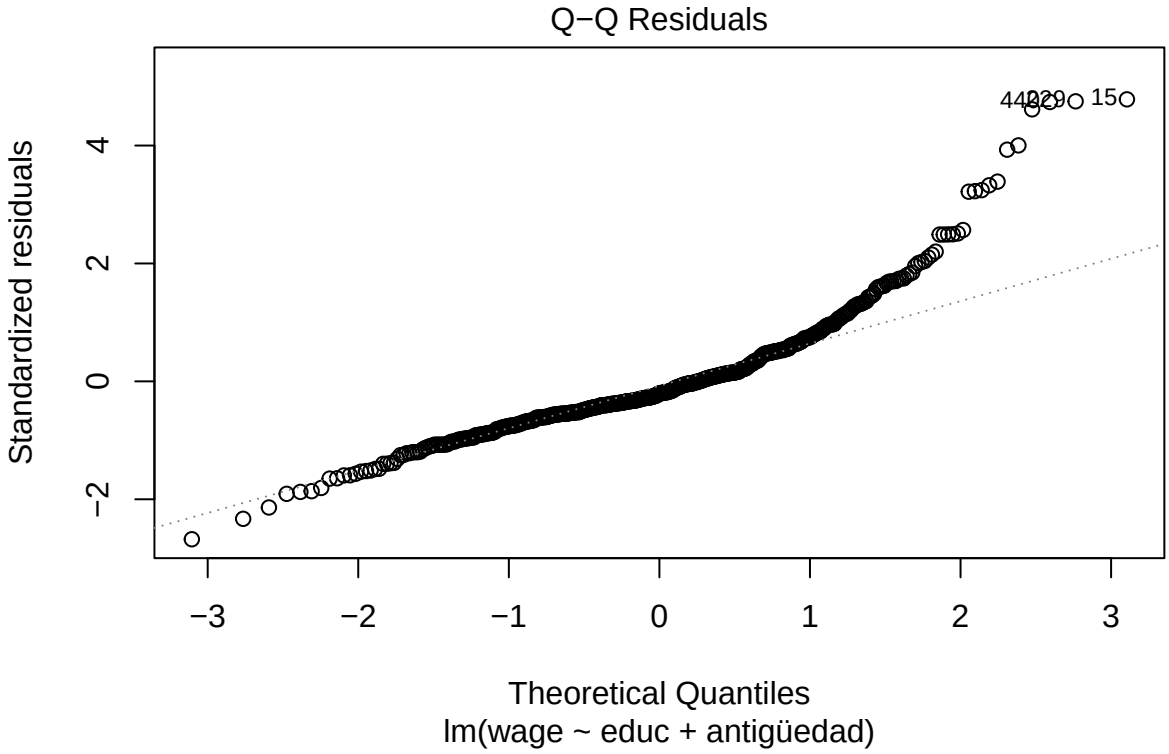
Análisis de suposiciones

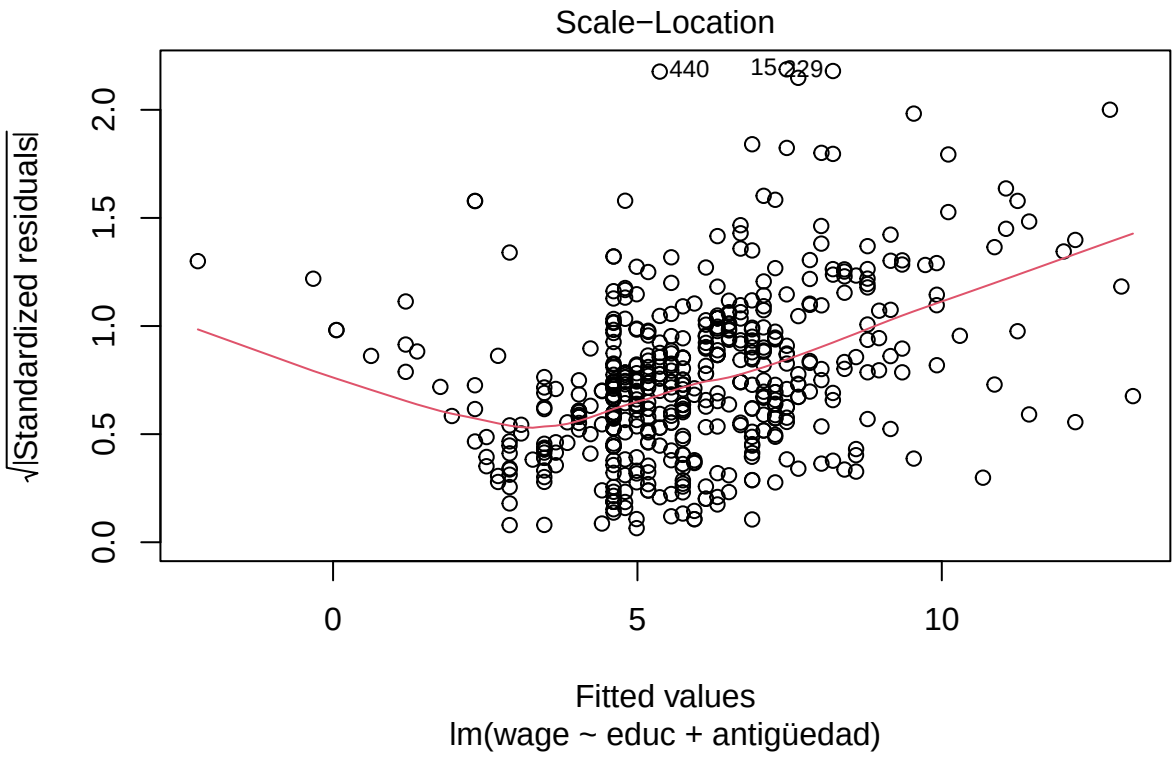
Antes evaluamos elementos de los residuos, pero lo hicimos por encima. R tiene en forma base unos gráficos que ayudan a informarnos sobre si los modelos tienen problemas en cómo están siendo aplicados.

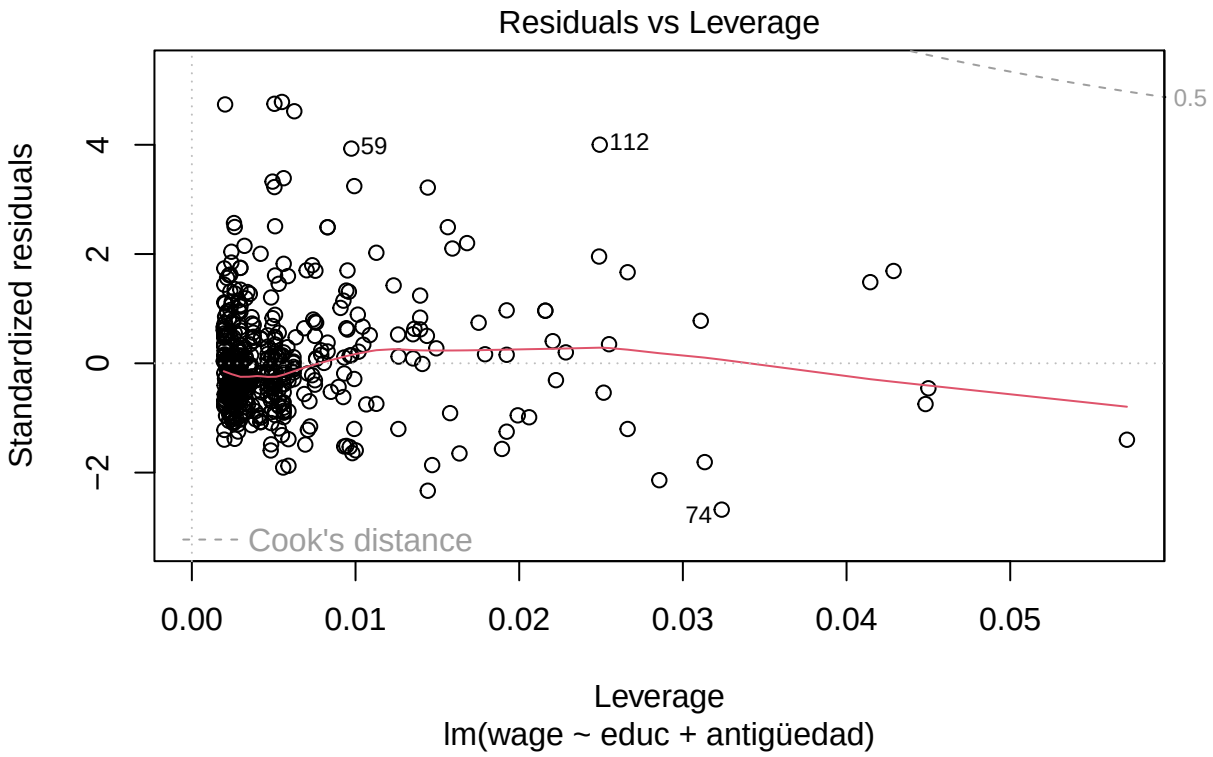
Primero, podemos evaluar el modelo gráficamente con la función `plot()`:

```
plot(mod2)
```




Q−Q Residuals
Standardized residuals
4
2
0
−2
−3
−2
−1
0
1
2
3
15
Theoretical Quantiles
lm(wage ~ educ + antigüedad)


Scale−Location
440
152
229
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5
10
Fitted values
lm(wage ~ educ + antigüedad)



1. Residuos vs Ajustados (Residuals vs Fitted):

- Qué muestra: Este gráfico representa los residuos estandarizados en función de los valores ajustados por el modelo.
- Interpretación: Sirve para detectar patrones no lineales y evaluar la homogeneidad de la varianza (homocedasticidad). Si los puntos se distribuyen aleatoriamente alrededor de la línea horizontal (residuo = 0) sin formar patrones, indica que el modelo es adecuado. Patrones sistemáticos o formas específicas (como una curva) sugieren que el modelo no captura adecuadamente la relación entre las variables, o que existe heterocedasticidad.

2. Gráfico Q-Q Normal (Normal Q-Q Plot):

- Qué muestra: Compara la distribución de los residuos estandarizados con una distribución normal teórica.
- Interpretación: Evalúa la normalidad de los residuos, un supuesto clave en modelos lineales. Si los puntos siguen aproximadamente una línea recta, los residuos se distribuyen normalmente. Desviaciones significativas de la línea recta indican que los residuos no son normales, lo que puede afectar la validez de los intervalos de confianza y pruebas de hipótesis.

3. Escala-Ubicación (Scale-Location Plot):

- Qué muestra: Grafica la raíz cuadrada de los residuos estandarizados `(sqrt(\|Residuos estandarizados\|))` frente a los valores ajustados.

- Interpretación: Ayuda a verificar la homocedasticidad. Una dispersión uniforme de puntos sugiere varianza constante de los residuos. Si los puntos muestran un patrón (por ejemplo, se ensanchan o estrechan a lo largo del eje de los ajustados), indica heterocedasticidad, lo que puede afectar la eficiencia de los estimadores.

4. Residuos Estandarizados vs Apalancamiento (Residuals vs Leverage):

- Qué muestra: Muestra los residuos estandarizados frente al apalancamiento de cada observación, con curvas de distancia de Cook superpuestas.
- Interpretación: Identifica observaciones influyentes que tienen un gran impacto en el ajuste del modelo. Puntos con alto apalancamiento y residuos grandes pueden distorsionar los resultados. Las líneas de distancia de Cook ayudan a detectar estos puntos. Observaciones más allá de estas líneas merecen una revisión adicional.

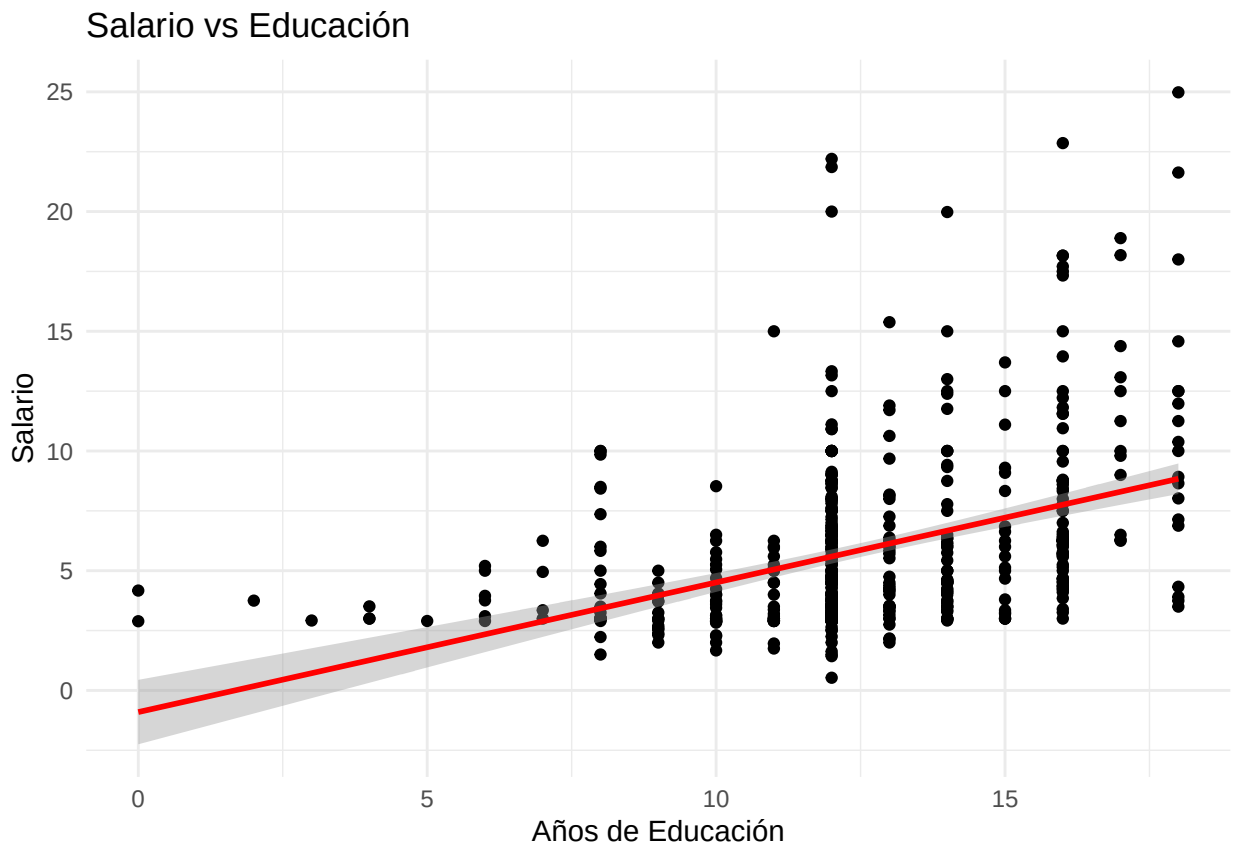
Notamos que el modelo tiene sus problemas, violentando varios de los principios señalados antes.

Linealidad

La relación entre las variables independientes y la variable dependiente es lineal.

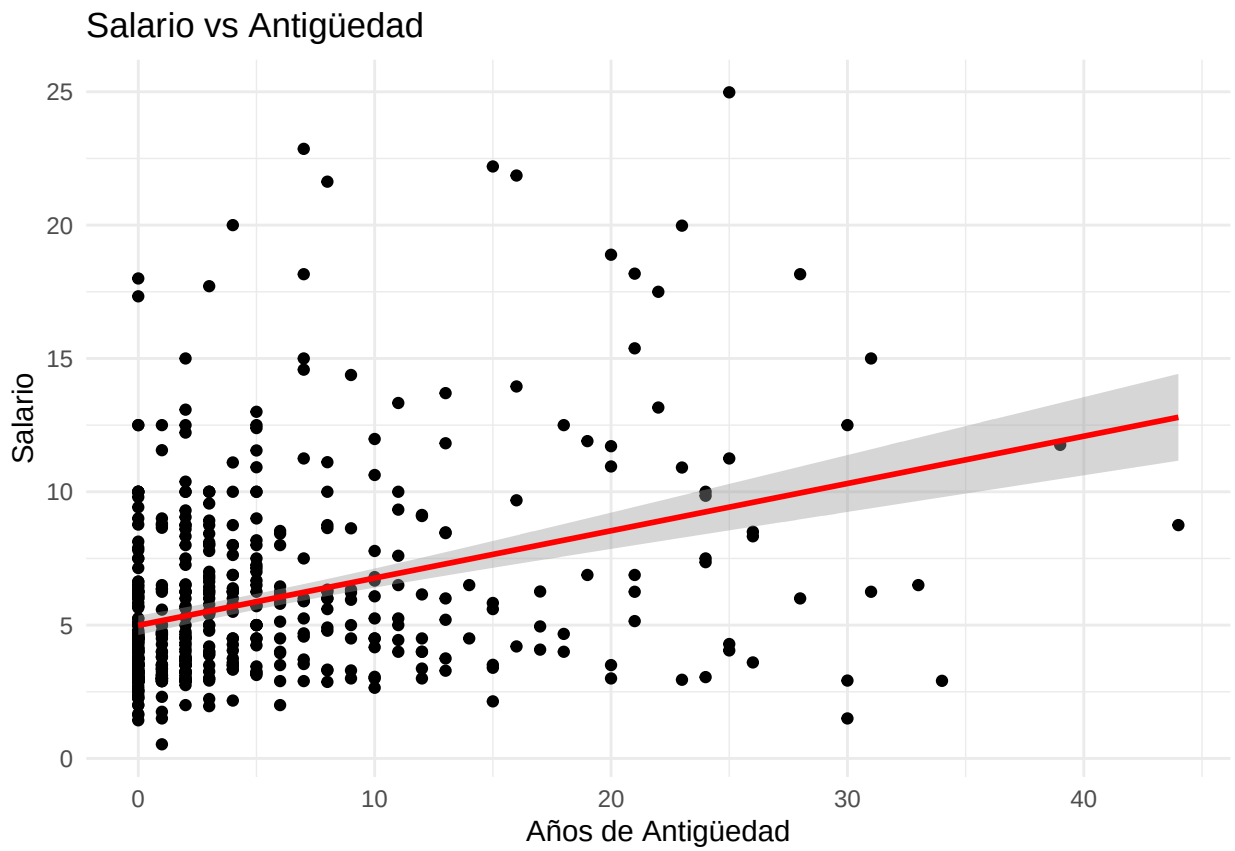
```
# Gráficos de dispersión con línea de regresión para cada variable independiente
library(ggplot2)

ggplot(base1, aes(x = educ, y = wage)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Salario vs Educación", x = "Años de Educación", y = "Salario") +
  theme_minimal()
```

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




```
ggplot(base1, aes(x = antigüedad, y = wage)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Salario vs Antigüedad", x = "Años de Antigüedad", y = "Salario") +
  theme_minimal()
```

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



Independencia de errores

Realizando la prueba de Durbin-Watson para verificar la autocorrelación estocástica. La hipótesis nula es que los residuos no tienen autocorrelación (es posible una alternativa, editando opciones), es decir que los residuos son normales.

```
# Prueba de Durbin-Watson
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
dwtest(mod2)
```

```
##
##  Durbin-Watson test
##
```

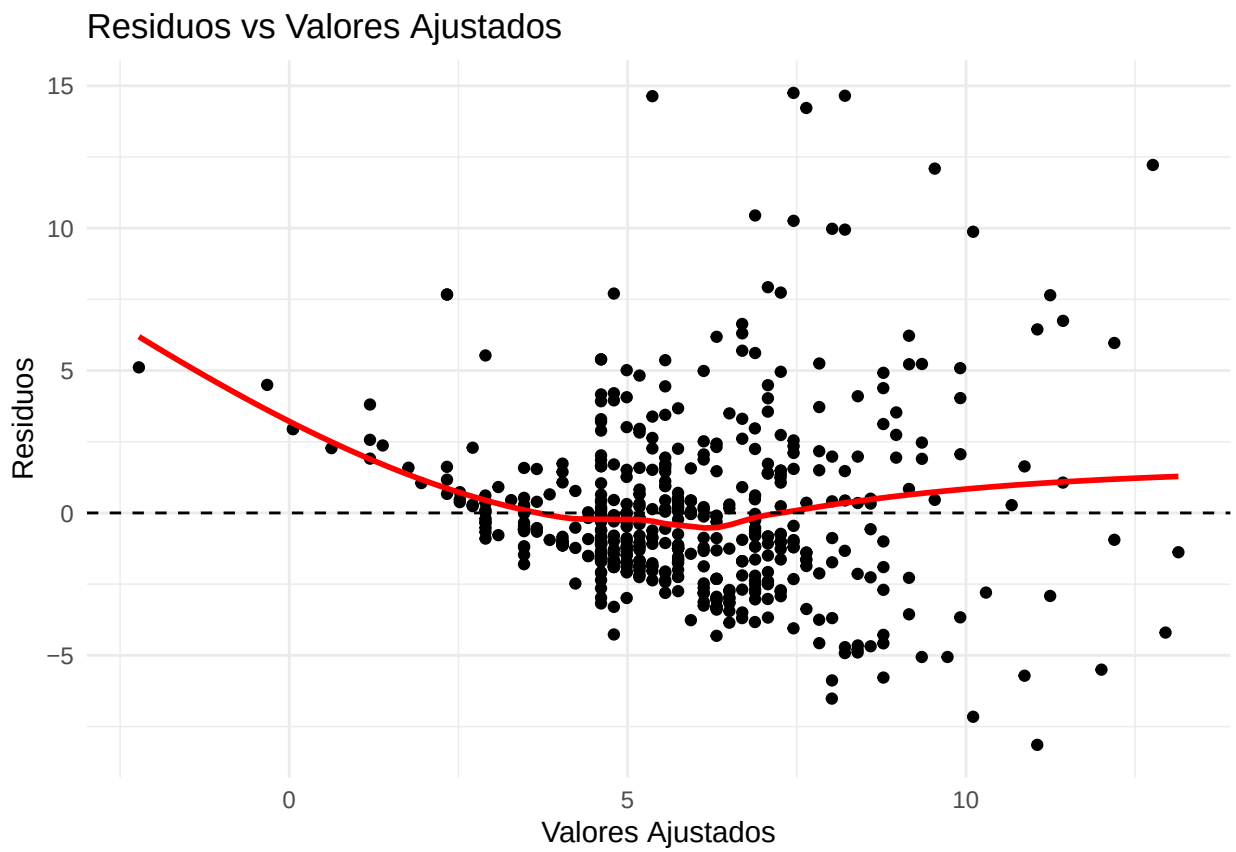
```
## data:  mod2
## DW = 1.7907, p-value = 0.008009
## alternative hypothesis: true autocorrelation is greater than 0
```

En este caso la prueba rechaza la nula. Tenemos problemas en nuestros residuos tal cual modelados al presente, pues se viola el supuesto de independencia.

Homoscedasticidad

```
# Gráfico de Residuos vs Valores Ajustados
ggplot(data = base1, aes(x = mod2$fitted.values, y = mod2$residuals)) +
  geom_point() +
  geom_smooth(method = "loess", col = "red", se = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuos vs Valores Ajustados", x = "Valores Ajustados", y = "Residuos") +
  theme_minimal()
```

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




```
# Prueba de Breusch-Pagan
library(lmtest)
bptest(mod2)
```

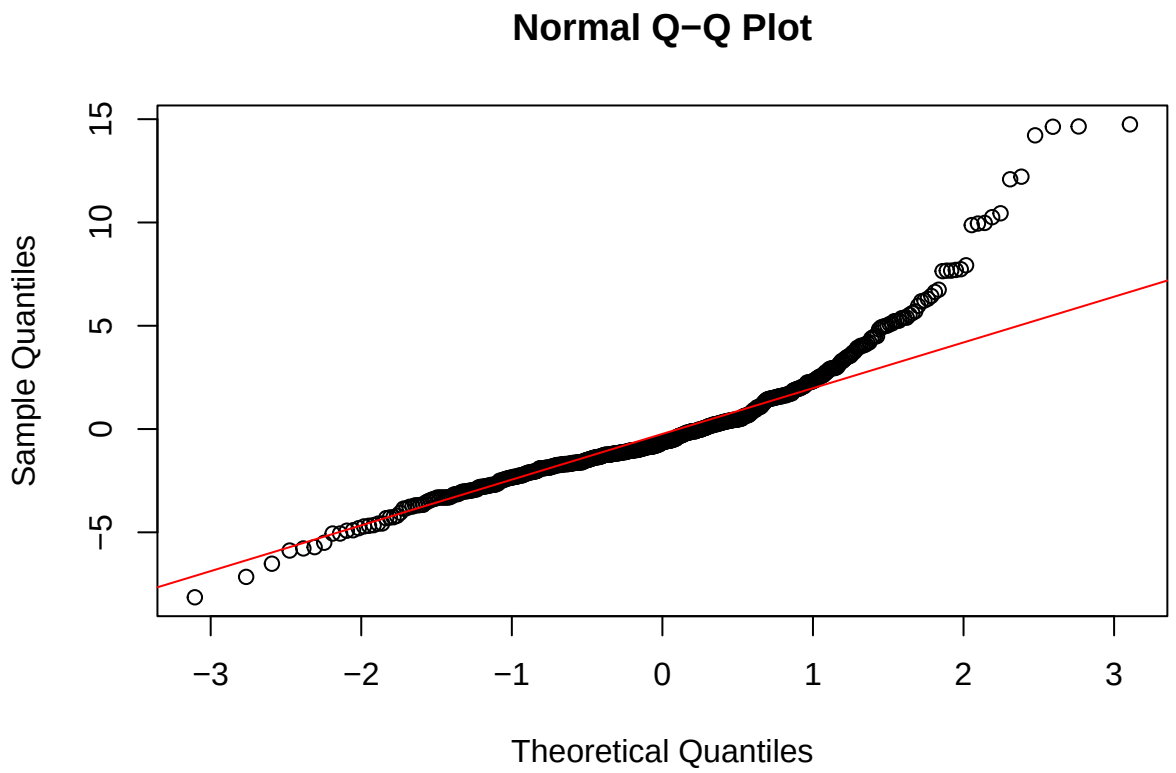
```
##
##  studentized Breusch-Pagan test
##
## data:  mod2
## BP = 40.798, df = 2, p-value = 1.383e-09
```

Vemos una representación visual (con `ggplot()`) de los residuos contrastados a los valores ajustados. La prueba Breusch Pagan tiene la hipótesis nula siguiente: varianza constante. Por consiguiente, rechazarla es encontrar heteroscedasticidad.

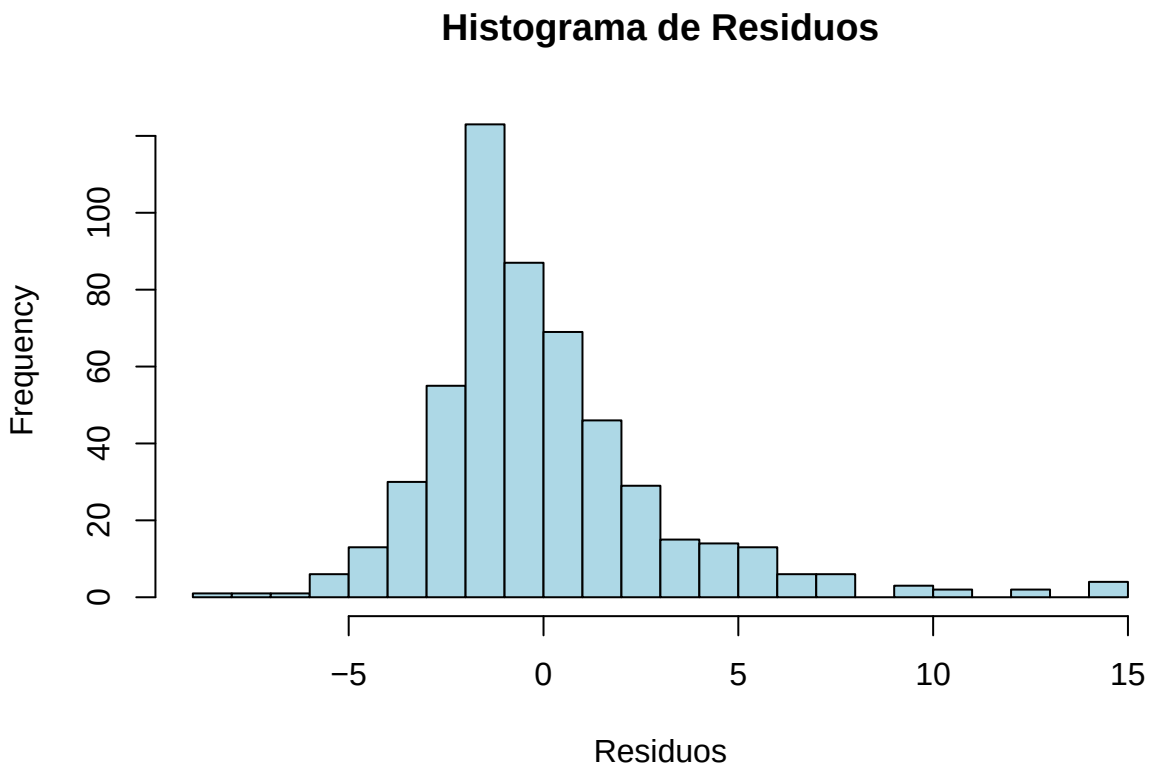
Normalidad de errores

El supuesto a verificar ahora (aunque no está yendo bien en general) es saber si los errores se distribuyen normalmente. Hay varias opciones gráficas y con pruebas estadísticas: el gráfico Q-Q Plot, el histograma de los residuos, el diagrama de densidad de residuos, y la prueba Shapiro-Wilk.

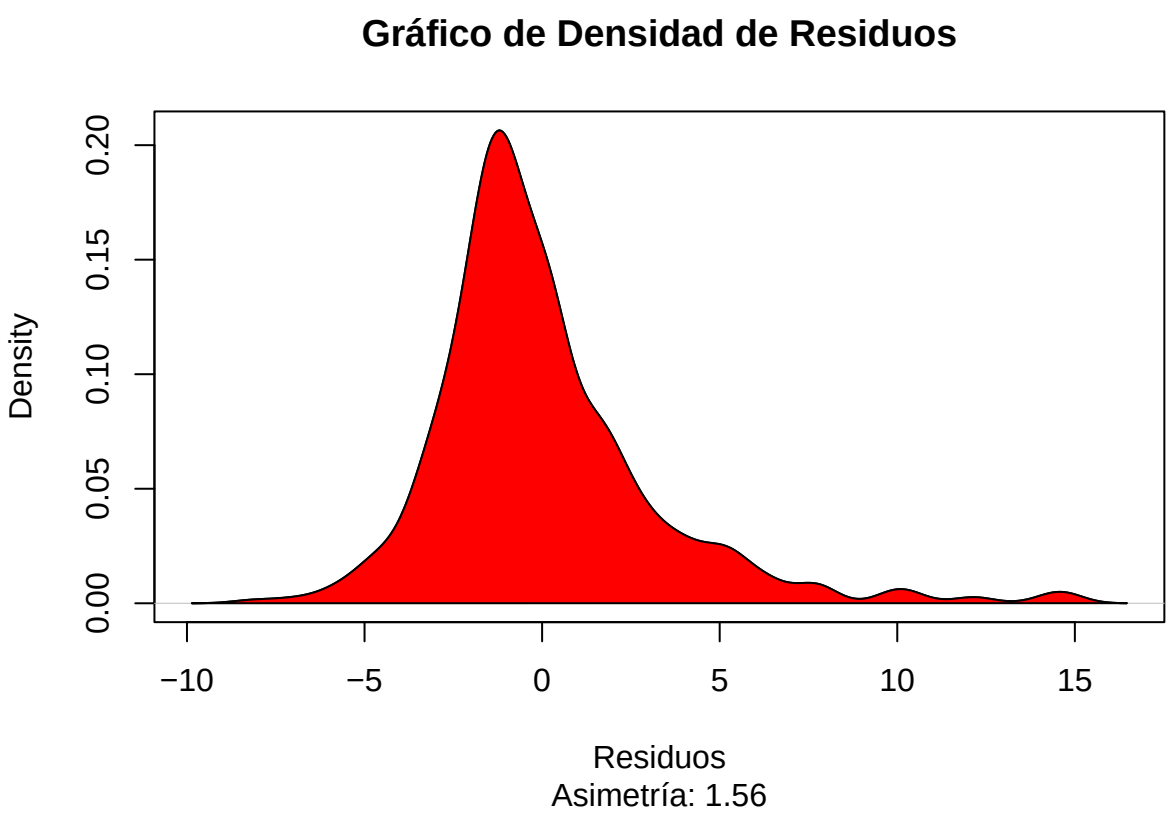
```r
# Gráfico Q-Q de los residuos
qqnorm(mod2$residuals)
qqline(mod2$residuals, col = "red")
```




```r
# Histograma de los residuos
hist(mod2$residuals, breaks = 20, main = "Histograma de Residuos", xlab = "Residuos", col = "lightblue"
```




```
# Gráfico de densidad de los residuos con asimetría
library(e1071)  # Para la función skewness()

plot(density(mod2$residuals),
     main = "Gráfico de Densidad de Residuos", xlab = "Residuos",
     sub = paste("Asimetría:", round(skewness(mod2$residuals), 2)))
polygon(density(mod2$residuals), col = "red")
```



Por ejemplo, estos residuos distan de ser normales. Las cuantilas de los residuos distan de la normalidad esperada, por *MUCHO*. Verificamos

```
shapiro.test(mod2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod2$residuals
## W = 0.88987, p-value < 2.2e-16
```

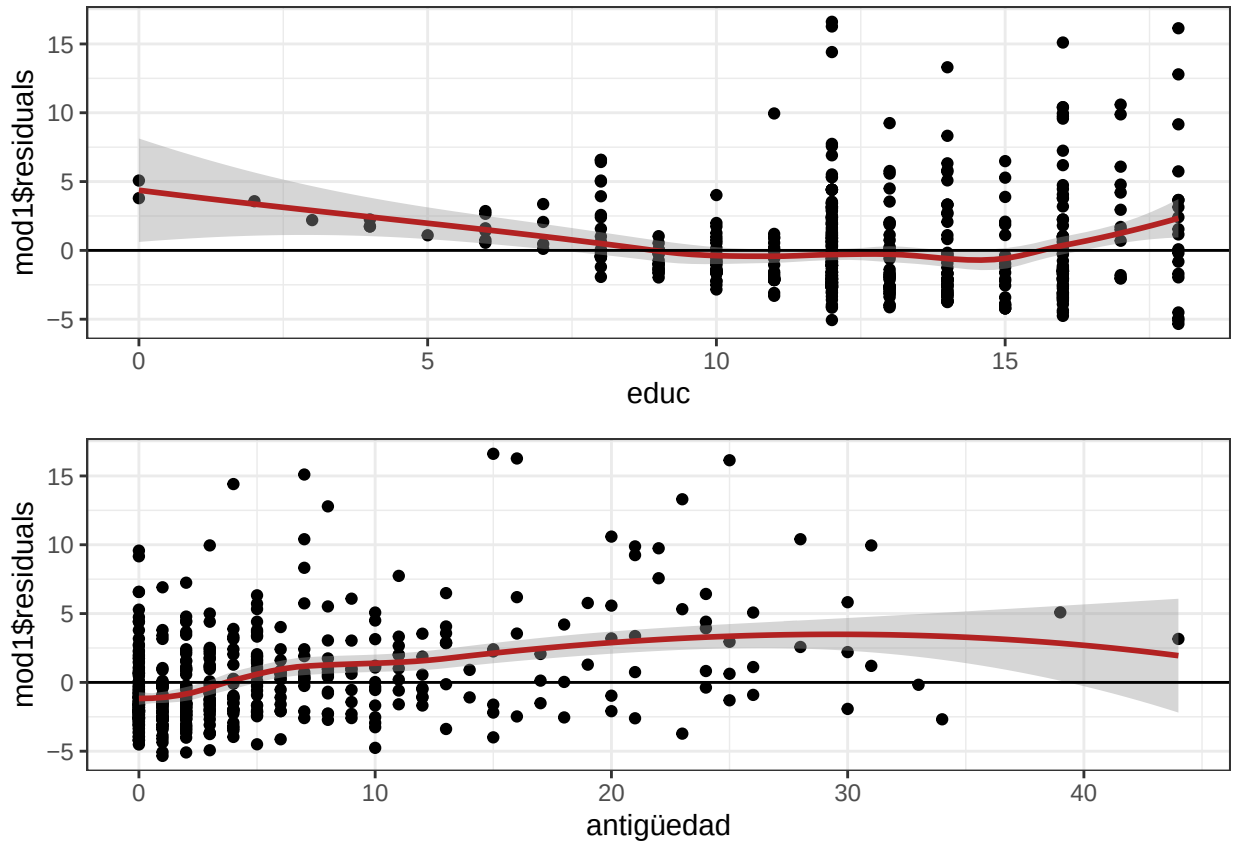
La hipótesis nula de la prueba de Shapiro-Wilk es que estamos ante normalidad en los errores. Esto confirma lo sugerido por el análisis gráfico y estadístico precedente.

En los siguientes gráficos se muestran los residuos contra cada uno de los regresores, los cuales se realizan con el siguiente código:

```
plot1 <- ggplot(data = base1, aes(educ, mod1$residuals)) +
  geom_point() +
  geom_smooth(color = "firebrick") +
  geom_hline(yintercept = 0) +
  theme_bw()

plot2 <- ggplot(data = base1, aes(antigüedad, mod1$residuals)) +
  geom_point() +
  geom_smooth(color = "firebrick") +
```

```
  geom_hline(yintercept = 0) +
  theme_bw()

grid.arrange(plot1, plot2)
```

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



El análisis gráfico indica problemas de heteroscedasticidad y de correlación entre los residuos y el nivel de los regresores.

Multicolinealidad

```
# Matriz de correlaciones entre variables independientes
vars_indep <- base1[, c("educ", "antigüedad")]
cor(vars_indep)
```

```
##                    educ  antigüedad
## educ         1.00000000 -0.05617257
## antigüedad -0.05617257  1.00000000
```

```r
# Cálculo del VIF
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```r
vif(mod2)
```

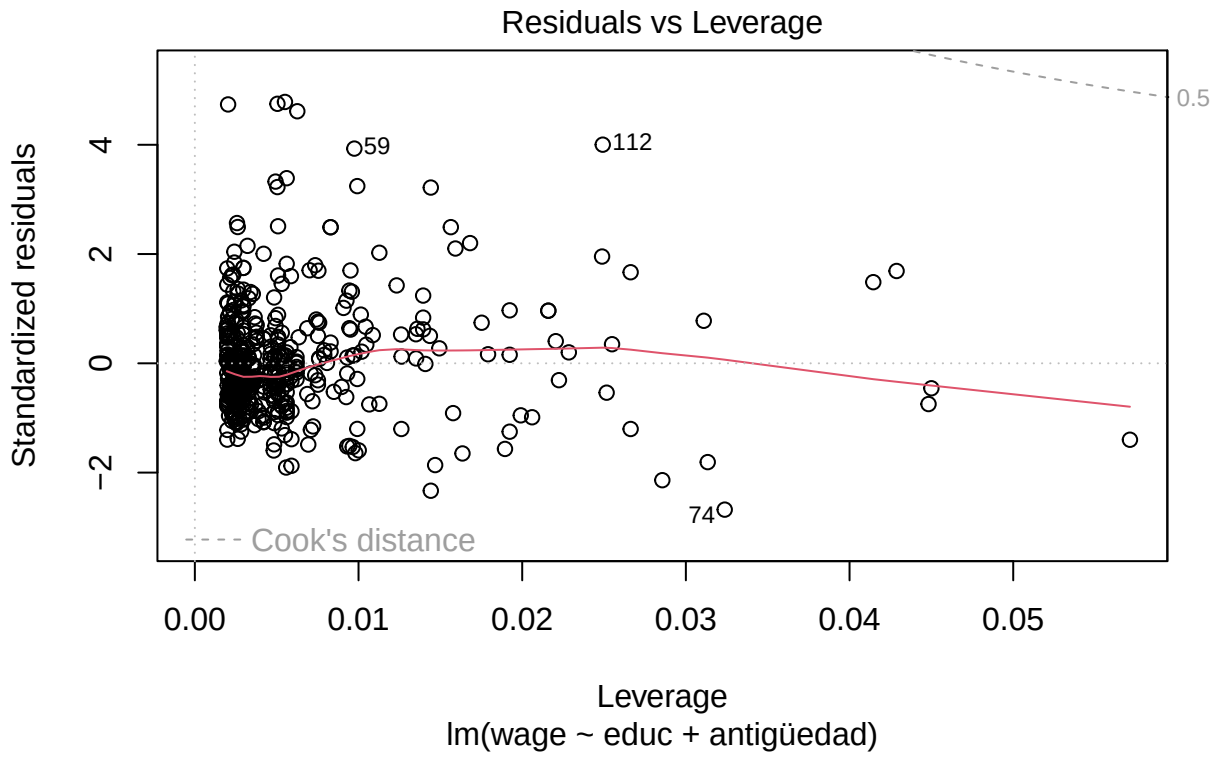
```
##       educ antigüedad
##   1.003165   1.003165
```

Los valores no son particularmente altos en correlación (alto es cerca de 1 ó -1, bajo es cerca de 0). En nuestro caso, la correlación entre educación y antigüedad es baja (-0.056), lo que sugiere baja multicolinealidad. La prueba del factor de inflación de varianza (VIF) se puede usar para verificar numéricamente si hay multicolinealidad.

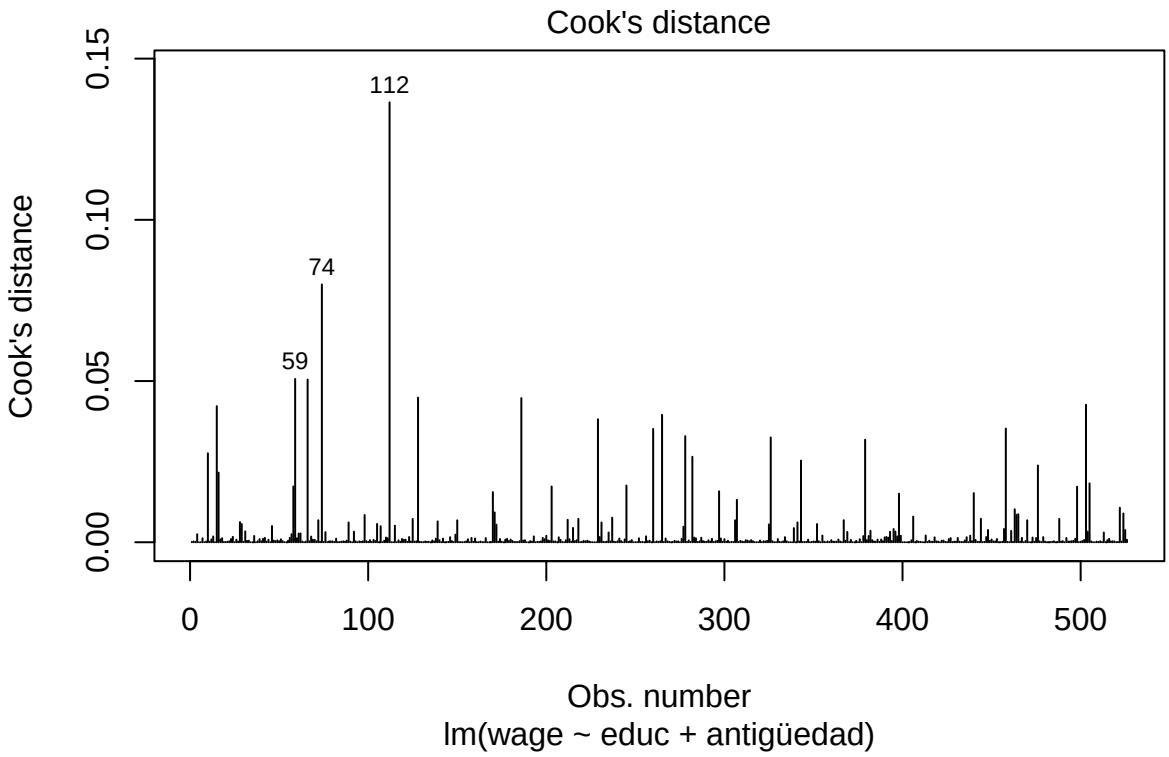
Identificar observaciones influyentes

Aunque no es un supuesto básico, es importante identificar puntos que puedan influir excesivamente en el modelo. Esto se puede verificar: - Analizando el gráfico de Residuos Estandarizados vs Apalancamiento. - Calculando la Distancia de Cook.

```r
# Gráfico de Residuos Estandarizados vs Apalancamiento
plot(mod2, which = 5)
```




```
# Gráfico de Distancia de Cook
plot(mod2, which = 4)
```




```
# Identificar observaciones con alta Distancia de Cook
cooksd <- cooks.distance(mod2)
influential <- as.numeric(names(cooksd)[(cooksd > (4 / nrow(base1)))])
influential
```

```
##  [1]  10  15  16  58  59  66  74  98 112 128 170 171 186 203 229 237 245 260 265
## [20] 278 282 297 307 326 343 379 398 406 440 458 463 464 465 476 498 503 505 522
## [39] 524
```

```
# Mostrar las observaciones influyentes
base1[influential, ]
```

```
##      wage educ exper antigüedad nonwhite female married     lwage expersq
## 10  18.18   17    22         21        0      0       1 2.9003222     484
## 15  22.20   12    31         15        0      0       1 3.1000924     961
## 16  17.33   16    14          0        0      0       1 2.8524392     196
## 58  10.00    8    13          0        1      0       0 2.3025851     169
## 59  21.63   18     8          8        0      1       0 3.0740812      64
## 66  19.98   14    26         23        0      0       1 2.9947317     676
## 74   2.91   12    20         34        0      1       1 1.0681531     400
## 98  13.16   12    34         22        0      0       1 2.5771818    1156
## 112 24.98   18    29         25        0      0       1 3.2180755     841
## 128  1.50    8    31         30        0      0       0 0.4054651     961
## 170 15.38   13    25         21        0      0       1 2.7330680     625
## 171 14.58   18    13          7        0      1       0 2.6796508     169
```

```
## 186 21.86   12    24         16        0      0       1 3.0846586     576
## 203 10.00    8     9          0        0      0       1 2.3025851      81
## 229 22.86   16    16          7        0      0       1 3.1293886     256
## 237  3.50   18     3          1        0      1       0 1.2527629       9
## 245 18.16   16    29          7        0      0       1 2.8992214     841
## 260 18.00   18    13          0        0      1       0 2.8903718     169
## 265  8.75   12    47         44        0      0       1 2.1690538    2209
## 278 18.89   17    26         20        0      0       1 2.9386327     676
## 282  2.95   14    41         23        1      0       1 1.0818052    1681
## 297  3.60   10    34         26        0      1       1 1.2809339    1156
## 307  6.25   11    35         31        0      0       1 1.8325815    1225
## 326 18.16   16    35         28        0      0       1 2.8992214    1225
## 343 15.00   11    35         31        0      0       1 2.7080503    1225
## 379  4.17    0    22         10        0      1       0 1.4279160     484
## 398  4.29   12    47         25        0      1       1 1.4562867    2209
## 406  8.43    8    27          3        0      0       1 2.1317968     729
## 440 20.00   12    22          4        0      0       1 2.9957323     484
## 458  6.50   14    41         33        0      0       1 1.8718022    1681
## 463  3.05    8    50         24        0      1       0 1.1151416    2500
## 464  3.50   12    26         20        0      1       1 1.2527629     676
## 465  2.92    3    51         30        1      0       0 1.0715836    2601
## 476 17.50   16    23         22        0      0       1 2.8622010     529
## 498  5.15   16    39         21        0      1       0 1.6389967    1521
## 503  2.89    0    42          0        0      1       1 1.0612565    1764
## 505 17.71   16    10          3        0      0       1 2.8741293     100
## 522 15.00   16    14          2        0      1       1 2.7080503     196
## 524  4.67   15    13         18        0      0       1 1.5411590     169
##     antigüedadcuad
## 10             441
## 15             225
## 16               0
## 58               0
## 59              64
## 66             529
## 74            1156
## 98             484
## 112            625
## 128            900
## 170            441
## 171             49
## 186            256
## 203              0
## 229             49
## 237              1
## 245             49
## 260              0
## 265           1936
## 278            400
## 282            529
## 297            676
## 307            961
## 326            784
## 343            961
## 379            100
```

```
## 398            625
## 406              9
## 440             16
## 458           1089
## 463            576
## 464            400
## 465            900
## 476            484
## 498            441
## 503              0
## 505              9
## 522              4
## 524            324
```

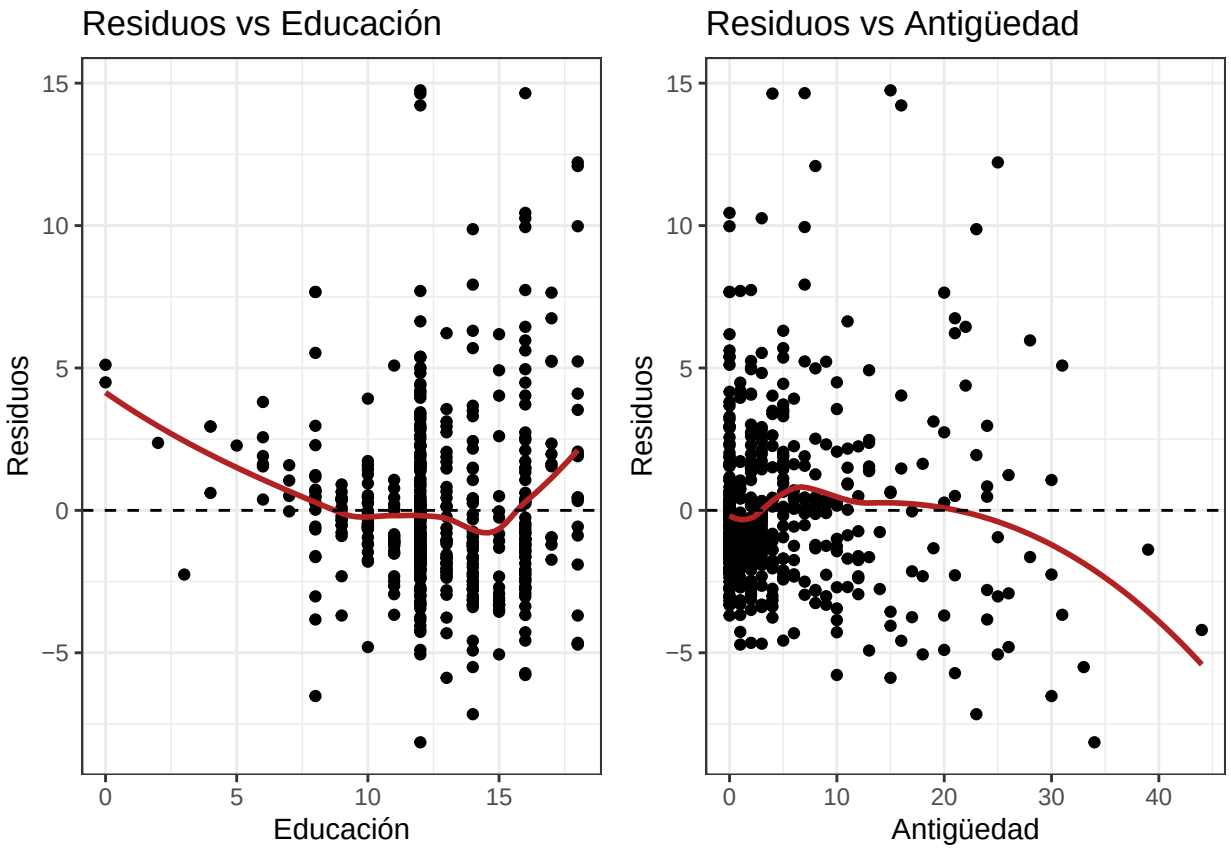
Puntos fuera de las líneas de referencia pueden ser influyentes. Los gráficos señalan los valores que parecen distantes al resto de lo observado. En estos casos, se recomienda revisar las observaciones para verificar si hay errores en los datos o si representan casos especiales.

Análisis de residuos contra variables independientes

Es útil graficar los residuos contra cada una de las variables independientes para detectar patrones específicos.

```
# Residuos vs Educación
plot1 <- ggplot(data = base1, aes(x = educ, y = mod2$residuals)) +
  geom_point() +
  geom_smooth(method = "loess", color = "firebrick", se = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuos vs Educación", x = "Educación", y = "Residuos") +
  theme_bw()

# Residuos vs Antigüedad
plot2 <- ggplot(data = base1, aes(x = antigüedad, y = mod2$residuals)) +
  geom_point() +
  geom_smooth(method = "loess", color = "firebrick", se = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuos vs Antigüedad", x = "Antigüedad", y = "Residuos") +
  theme_bw()

# Mostrar los gráficos lado a lado
library(gridExtra)
grid.arrange(plot1, plot2, ncol = 2)
```

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



- Residuos vs Educación: Si observamos un patrón sistemático, podría indicar que la relación no es completamente lineal o que hay variables omitidas.
- Residuos vs Antigüedad: Patrones similares pueden sugerir problemas de heterocedasticidad o relaciones no lineales.

¿Qué hacemos?

Como detectamos problemas de heterocedasticidad y normalidad de errores, podemos probar transformando la variable dependiente. Por ahora añadiré unas variables adicionales y también haremos unas variaciones con transformaciones (una de estas ya estaba en el conjunto de datos). Es posible que al incorporar variables relevantes que puedan explicar mejor la variabilidad en el salario, mejore el ajuste del modelo. Por otro lado, es posible que transformar la variable dependiente o algunas independientes para corregir violaciones a los supuestos del modelo lineal arreglen estos problemas.

A continuación, implementaremos estas estrategias paso a paso.

Añadiremos variables que podrían influir significativamente en el salario, como:

- married: Estado civil (1 si está casado, 0 si no).
- female: Género (1 si es mujer, 0 si es hombre).
- nonwhite: Etnicidad (1 si no es blanco, 0 si es blanco).

```
# Actualizamos el modelo añadiendo la variable 'married'
mod3 <- update(mod2, wage ~ educ + antigüedad + married)
```

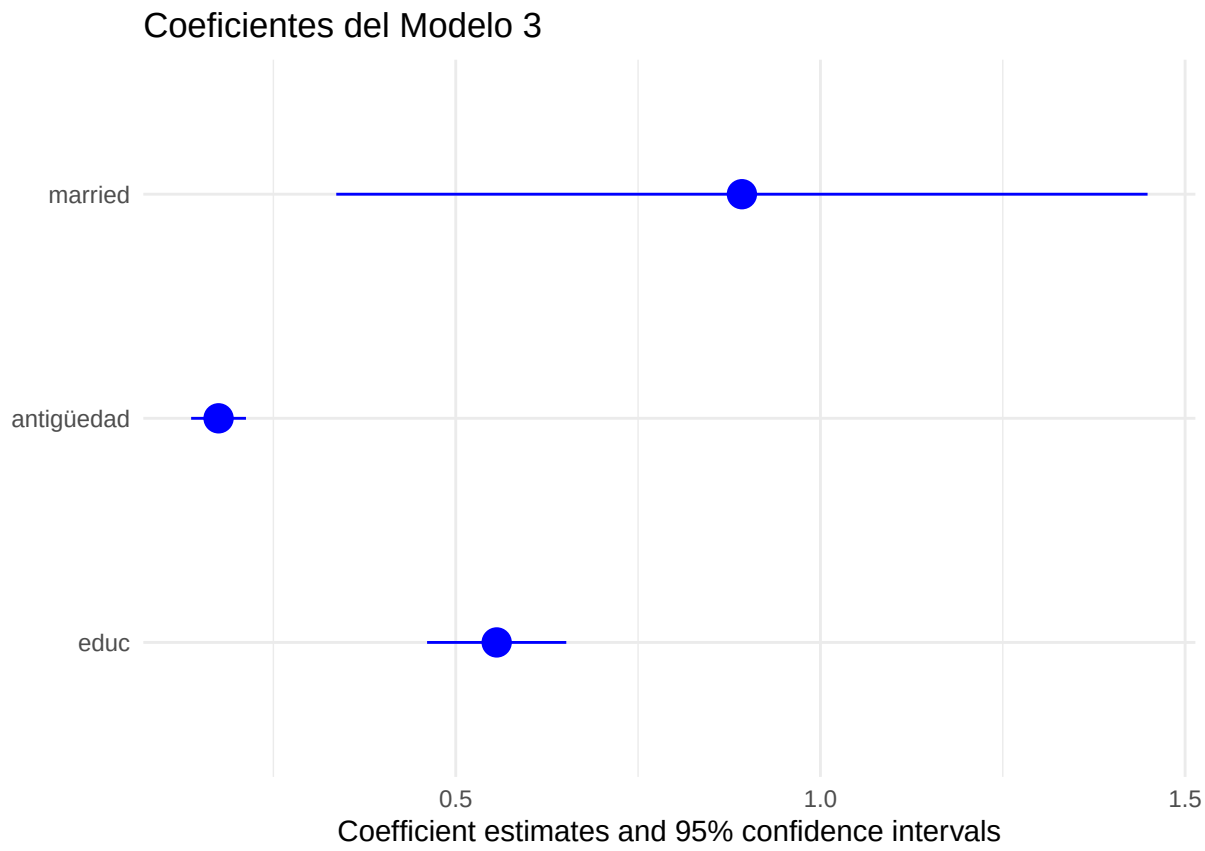
```r
# Resumen del modelo
summary(mod3)
```

```
##
## Call:
## lm(formula = wage ~ educ + antigüedad + married, data = base1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0742 -1.7278 -0.5579  1.1913 14.5374
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.52575    0.64205  -3.934 9.49e-05 ***
## educ         0.55614    0.04857  11.450  < 2e-16 ***
## antigüedad   0.17482    0.01913   9.139  < 2e-16 ***
## married      0.89235    0.28310   3.152  0.00171 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.066 on 522 degrees of freedom
## Multiple R-squared:  0.3149, Adjusted R-squared:  0.311
## F-statistic: 79.98 on 3 and 522 DF,  p-value: < 2.2e-16
```

- Intercepto (-2.52575): Representa el salario promedio cuando educ, antigüedad y married son cero. Aunque no es interpretable en este contexto, es necesario para el modelo.
- Educación (0.55614): Por cada año adicional de educación, el salario promedio aumenta en aproximadamente 56 chavos, manteniendo constantes la antigüedad y el estado civil.
- Antigüedad (0.17482): Por cada año adicional de antigüedad con el empleador actual, el salario promedio aumenta en aproximadamente $0.17, manteniendo constantes las otras variables.
- Casado (0.89235): Estar casado se asocia con un aumento promedio de 89 chavos en el salario, manteniendo constantes la educación y la antigüedad.
- Significancia estadística: Todos los coeficientes son estadísticamente significativos ($p < 0.05$), lo que indica que tienen un efecto significativo en el salario.
- Ajuste del modelo:
  - R-cuadrado (0.3149): El modelo explica aproximadamente el 31% de la variabilidad en el salario, lo que es una mejora respecto a los modelos anteriores.
  - Error estándar residual (3.066): Es menor que en los modelos previos, lo que indica un mejor ajuste.

Para visualizar los coeficientes del modelo, podemos utilizar la librería `modelsummary` y la función `modelplot()`:

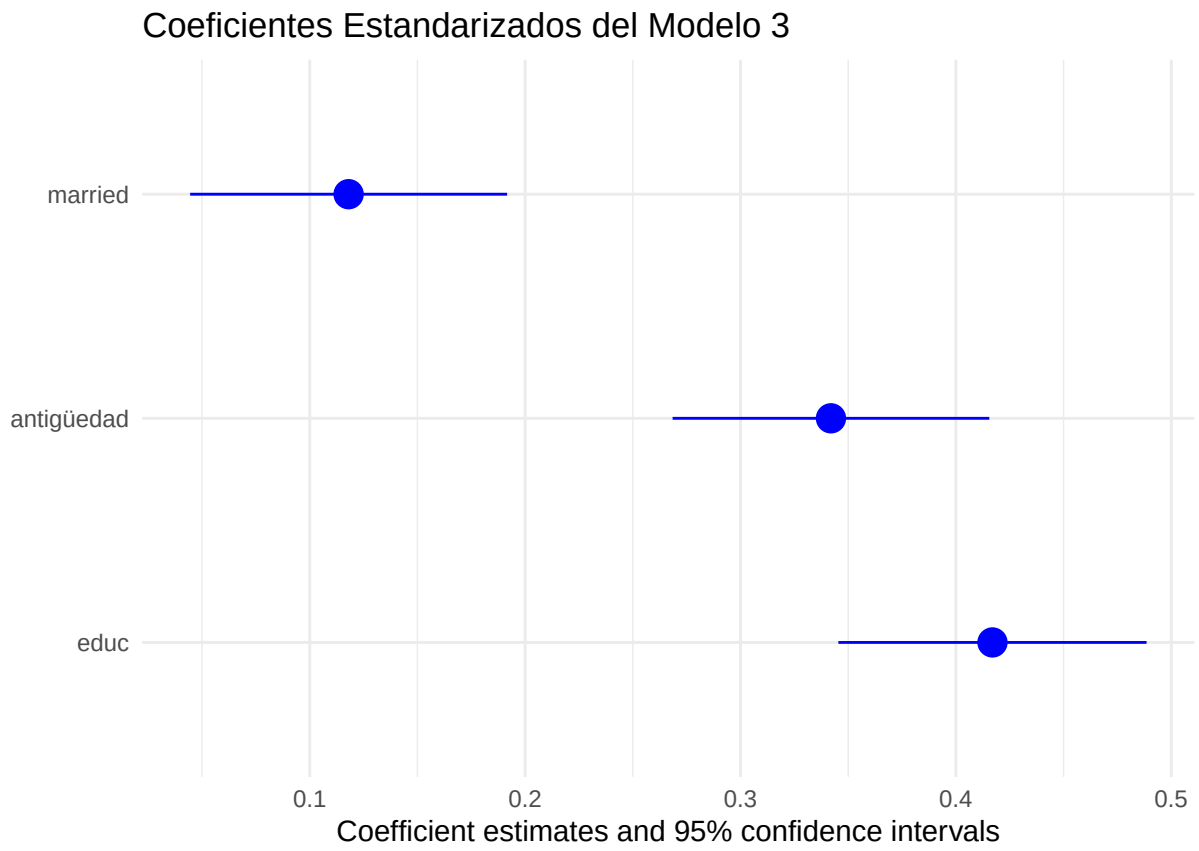
```r
library(modelsummary)
modelplot(mod3, coef_omit = "Intercept", color = "blue", size = 1) +
  labs(title = "Coeficientes del Modelo 3")
```



Nota: Al incluir variables categóricas como married, es importante considerar que sus coeficientes representan diferencias respecto a la categoría de referencia.

Estandarización de Variables Para comparar los efectos de las variables en la misma escala, podemos estandarizar las variables numéricas:

```
base1_estandarizado <- base1 |>
  mutate(across(where(is.numeric), scale))

# Ajustamos el modelo con los datos estandarizados
mod3_est <- lm(wage ~ educ + antigüedad + married, data = base1_estandarizado)

# Visualicemos los coeficientes estandarizados
modelplot(mod3_est, coef_omit = "Intercept", color = "blue", size = 1) +
  labs(title = "Coeficientes Estandarizados del Modelo 3")
```

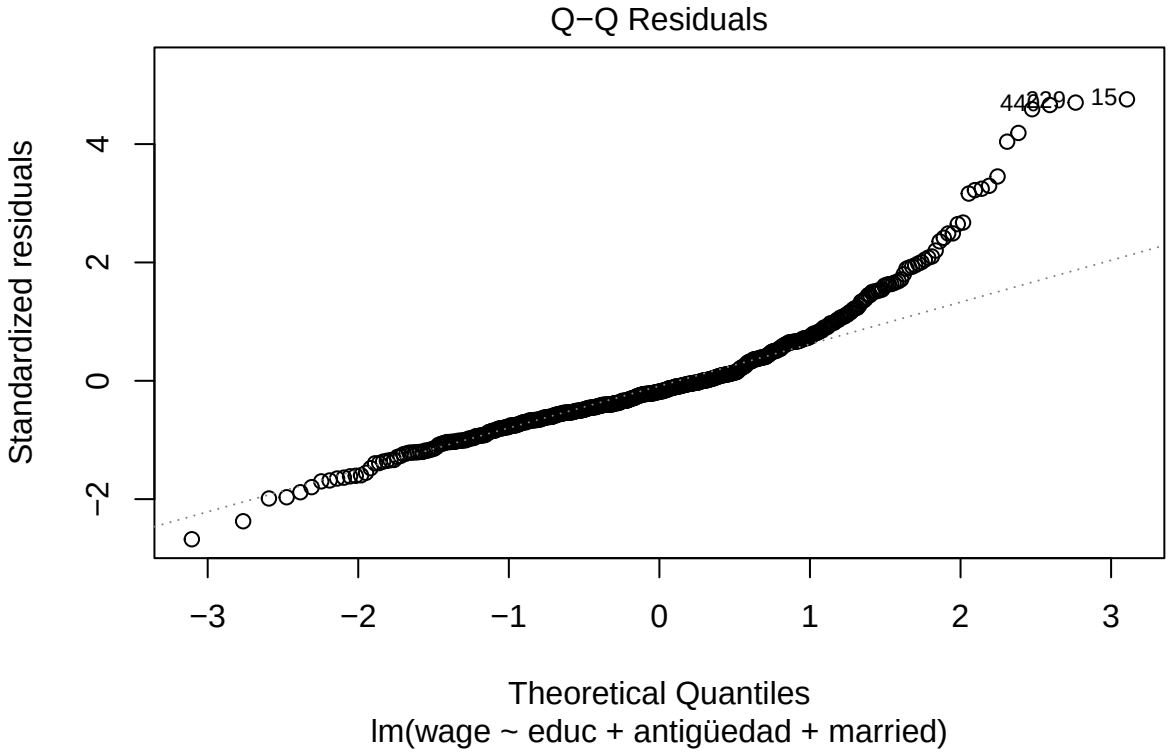


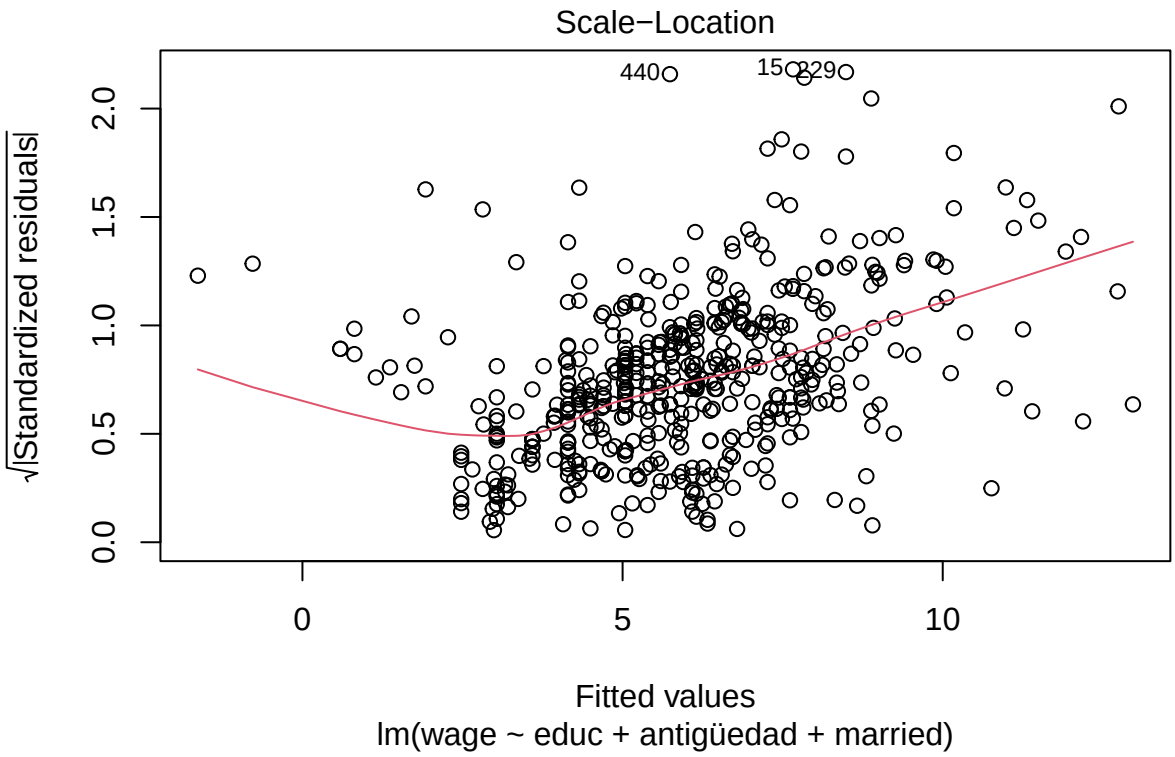
Los coeficientes estandarizados permiten comparar directamente el efecto relativo de cada variable en el salario. Un coeficiente más grande en valor absoluto indica un efecto mayor.

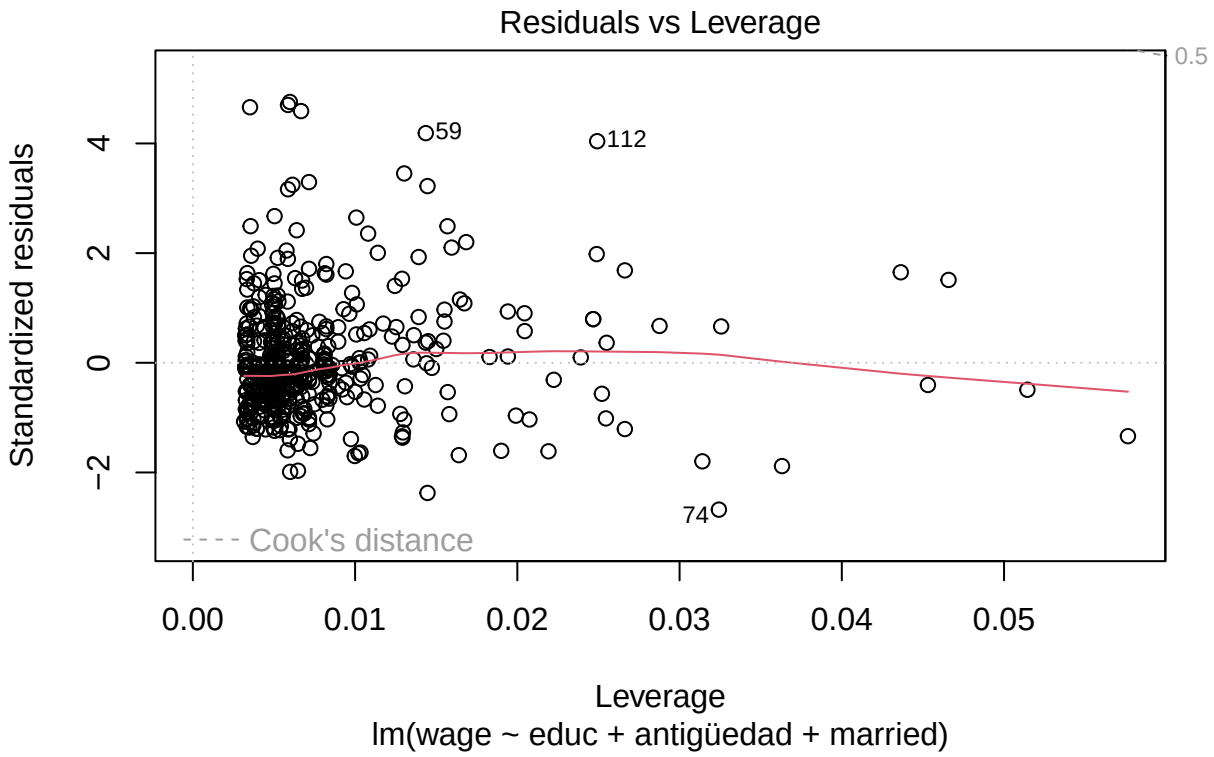
Al ejecutar `plot(mod3)`, obtenemos los gráficos diagnósticos para evaluar los supuestos del modelo.

```
plot(mod3)
```


Residuals vs Fitted
Residuals
Fitted values
lm(wage ~ educ + antigüedad + married)
440
15
229
15
10
5
0
−5
−10
0
5
10


Q−Q Residuals
Standardized residuals
4
2
0
−2
−3
−2
−1
0
1
2
3
15
Theoretical Quantiles
lm(wage ~ educ + antigüedad + married)


Scale−Location
440
15
229
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5
10
Fitted values
lm(wage ~ educ + antigüedad + married)




```
#library(lmtest)
bptest(mod3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mod3
## BP = 43.296, df = 3, p-value = 2.129e-09
```

```
shapiro.test(mod3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod3$residuals
## W = 0.89096, p-value < 2.2e-16
```

Los residuos todavía siguen distando de ser una distribución normal. Ampliaremos el modelo con otras variables que señaláramos antes como de interés.

```
# Creamos un nuevo modelo incluyendo 'female' y 'nonwhite'
mod4 <- lm(wage ~ educ + antigüedad + married + female + nonwhite, data = base1)

# Resumen del modelo
summary(mod4)
```

```
##
## Call:
## lm(formula = wage ~ educ + antigüedad + married + female + nonwhite,
##     data = base1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4948 -1.7690 -0.5398  1.0261 13.9814
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.12270    0.66394  -1.691   0.0914 .
## educ         0.52878    0.04718  11.208  < 2e-16 ***
## antigüedad   0.15423    0.01873   8.234 1.47e-15 ***
## married      0.68258    0.27546   2.478   0.0135 *
## female      -1.71149    0.26644  -6.424 3.01e-10 ***
## nonwhite    -0.06516    0.42715  -0.153   0.8788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.956 on 520 degrees of freedom
## Multiple R-squared:  0.3653, Adjusted R-squared:  0.3592
## F-statistic: 59.85 on 5 and 520 DF,  p-value: < 2.2e-16
```

- Educación y antigüedad: Siguen siendo significativas y positivas.
- Married (0.68258): Estar casado se asocia con un aumento promedio de $0.68 en el salario, manteniendo constantes las demás variables.
- Female (-1.71149): Ser mujer se asocia con una disminución promedio de $1.71 en el salario, manteniendo constantes las demás variables. Este efecto es estadísticamente significativo.
- Nonwhite (-0.06516): No es estadísticamente significativo (p = 0.8788), lo que sugiere que, en este modelo, la variable nonwhite no tiene un efecto significativo en el salario.
- Ajuste del modelo:
  - R-cuadrado (0.3653), ajustado (0.3592): El modelo explica aproximadamente el 36% de la variabilidad en el salario, mejorando levemente respecto al modelo anterior.

Análisis Es importante verificar la multicolinealidad al añadir nuevas variables.

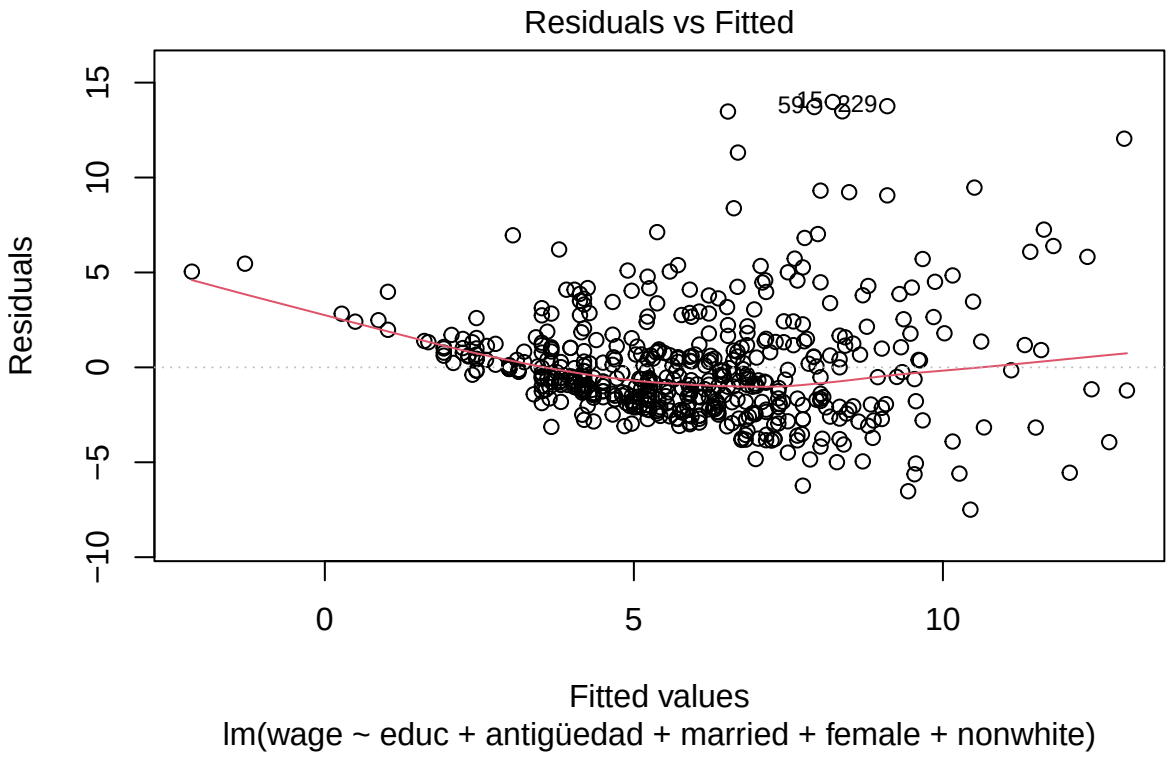
```
#library(car)
vif(mod4)
```

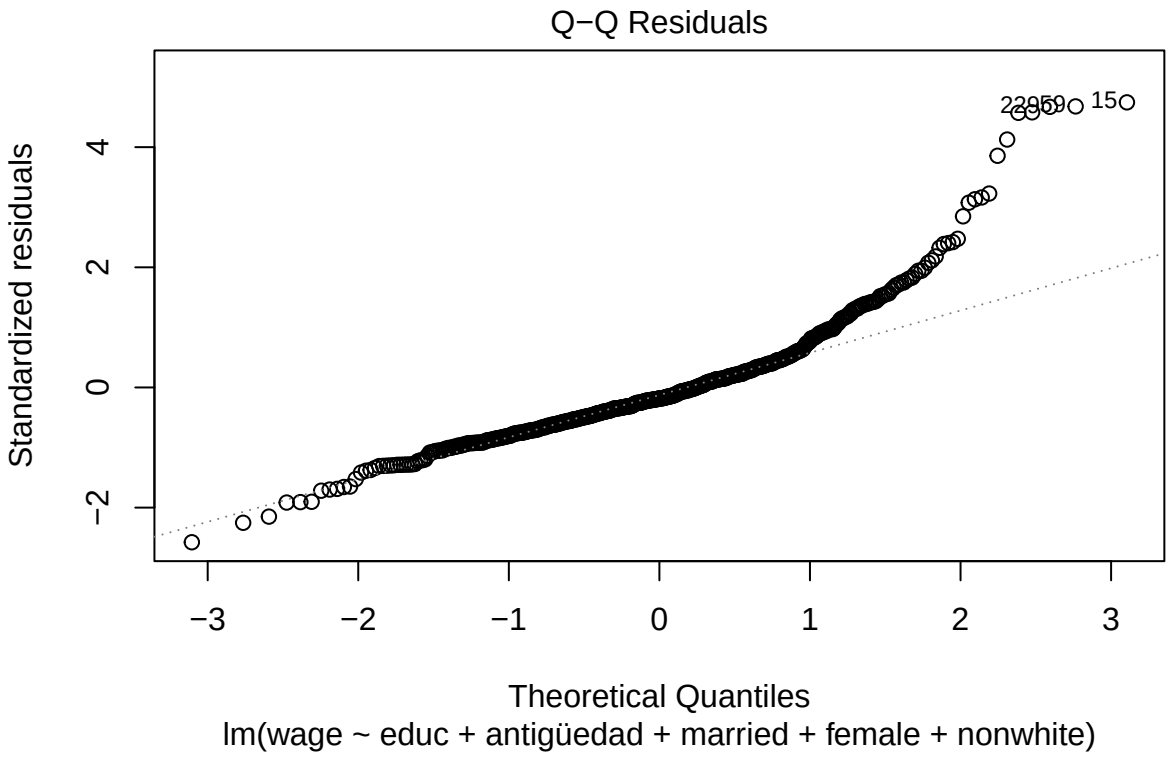
```
##       educ antigüedad    married     female   nonwhite
##   1.025155   1.099796   1.087978   1.066217   1.011546
```

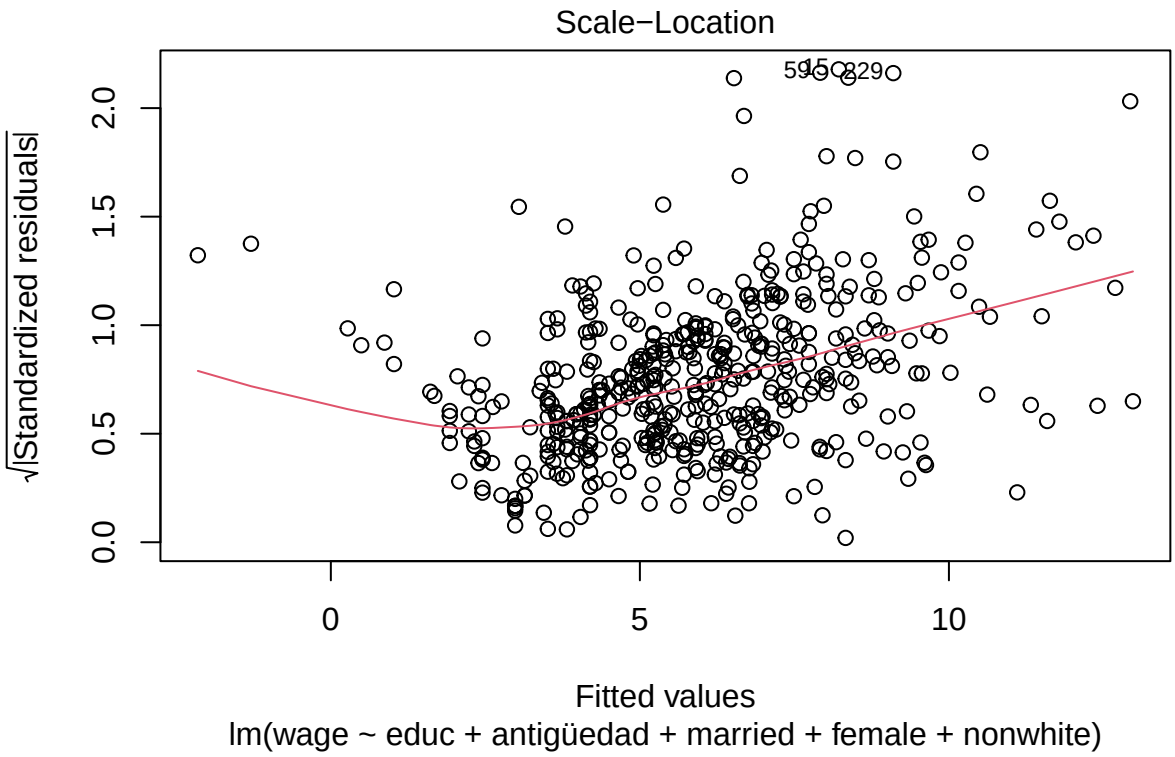
No parece haber problema de multicolinealidad, todas están cerca de 1.

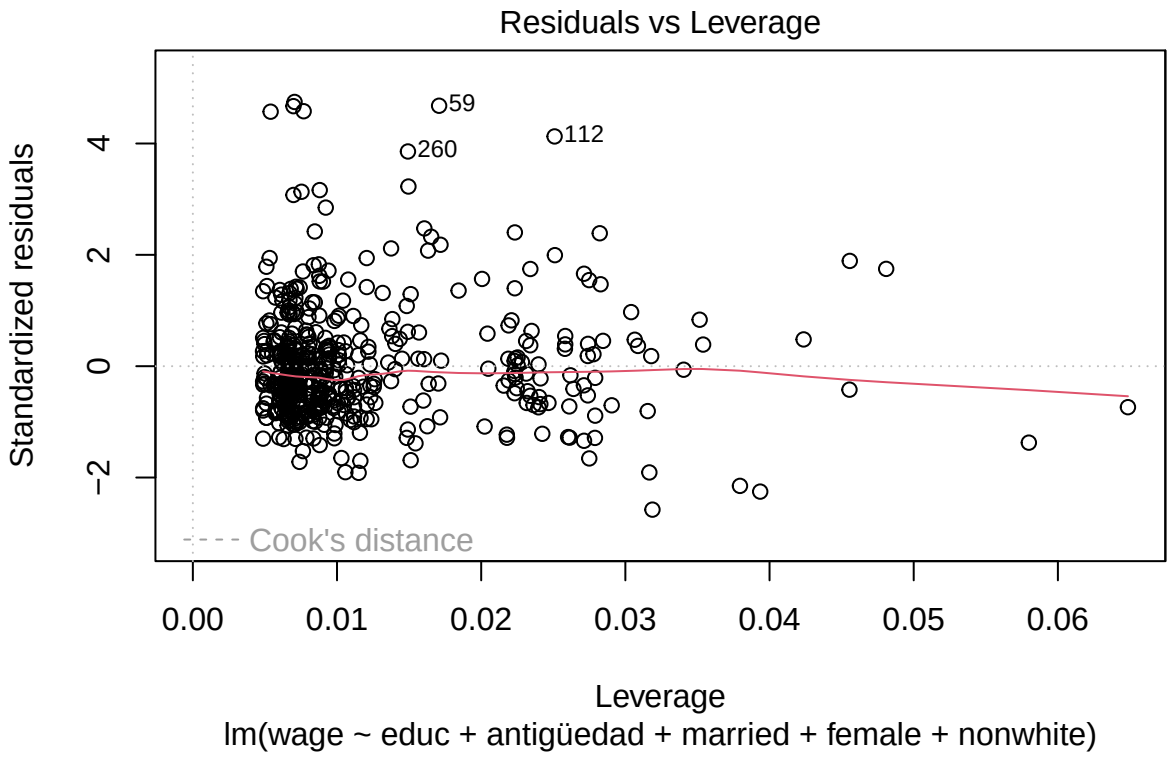
Verificaremos rápido igual los gráficos del modelo.

```
plot(mod4)
```


Residuals vs Fitted
Residuals
15
10
5
0
−5
−10
0
5
10
Fitted values
lm(wage ~ educ + antigüedad + married + female + nonwhite)


Q−Q Residuals
Standardized residuals
−2
0
2
4
−3
−2
−1
0
1
2
3
15
Theoretical Quantiles
lm(wage ~ educ + antigüedad + married + female + nonwhite)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5
10
Fitted values
lm(wage ~ educ + antigüedad + married + female + nonwhite)




```
#library(lmtest)
bptest(mod4)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mod4
## BP = 43.589, df = 5, p-value = 2.806e-08
```

```
shapiro.test(mod4$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod4$residuals
## W = 0.88709, p-value < 2.2e-16
```

Seguimos rechazando la hipótesis nula de Shapiro-Wilk aún en este caso. El problema no parece ser resuelto por añadidura de otros términos. Aplicamos una transformación logarítmica a la variable dependiente, salario.

```
# Estimamos el modelo con el logaritmo del salario
mod5 <- lm(log(wage) ~ educ + antigüedad + married + female + nonwhite, data = base1)
```

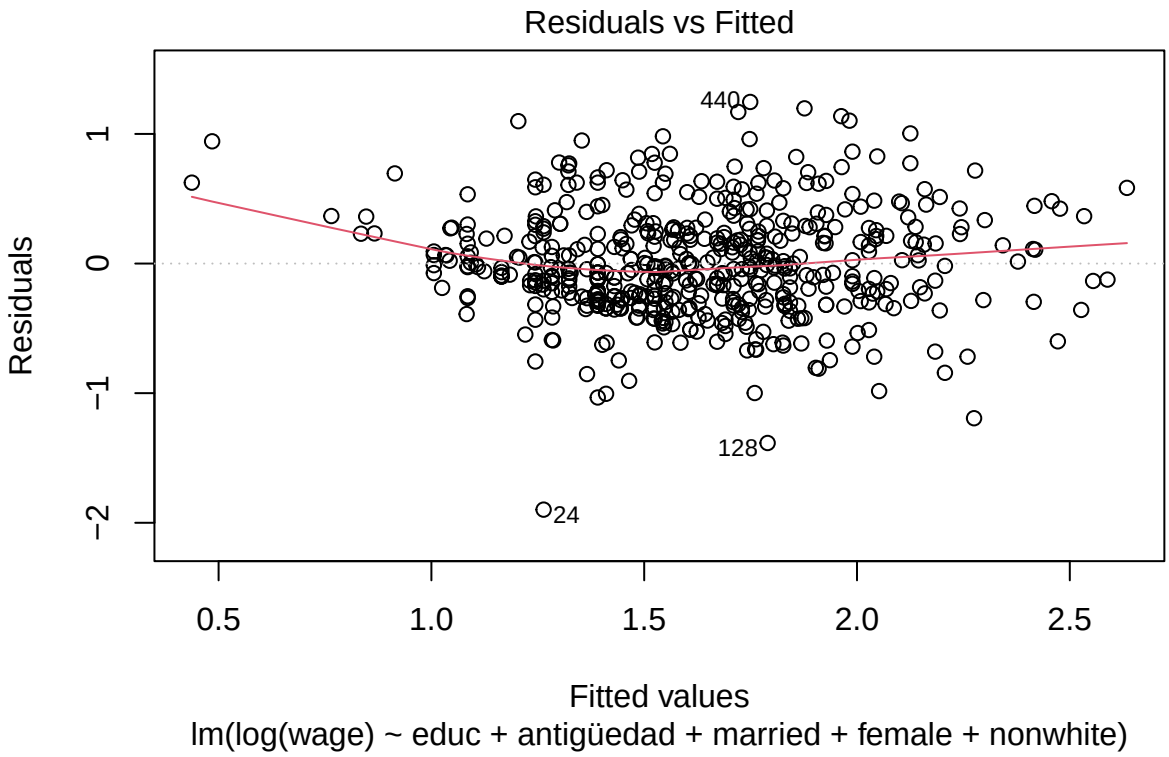
```r
# Resumen del modelo
summary(mod5)
```

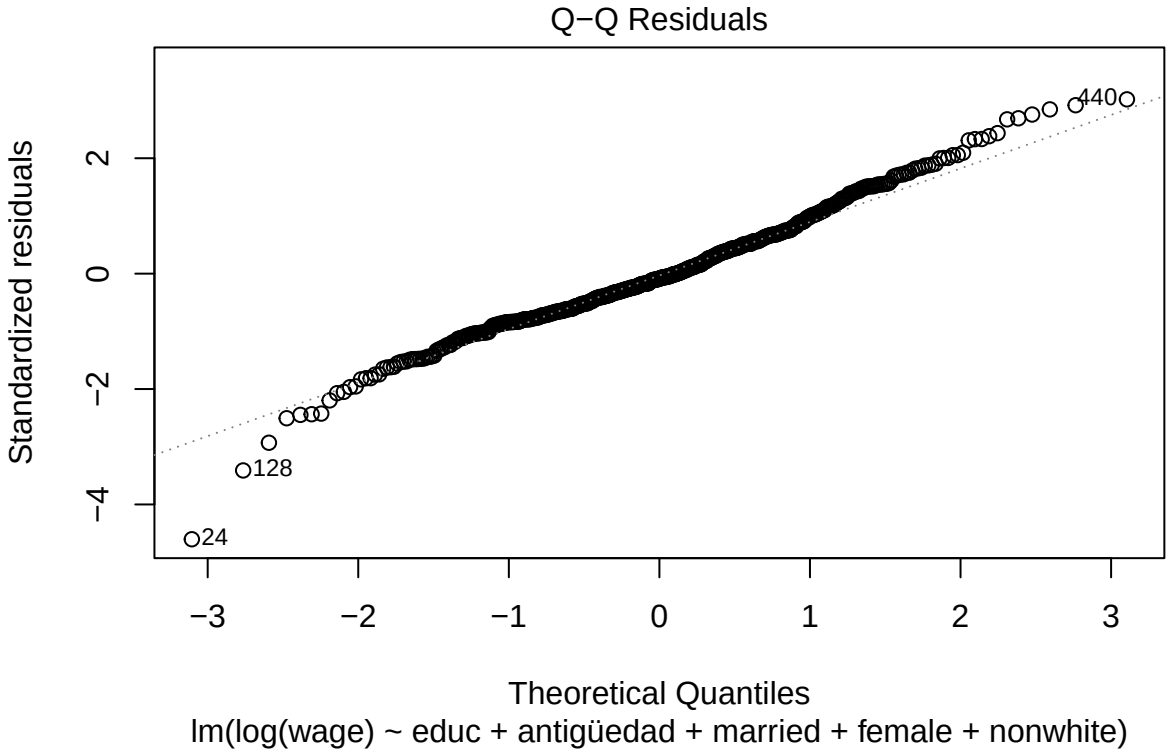
```
##
## Call:
## lm(formula = log(wage) ~ educ + antigüedad + married + female +
##     nonwhite, data = base1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89855 -0.27105 -0.03577  0.24487  1.24663
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.570835   0.092937   6.142 1.62e-09 ***
## educ         0.079483   0.006604  12.035  < 2e-16 ***
## antigüedad   0.019443   0.002622   7.416 4.94e-13 ***
## married      0.146703   0.038558   3.805 0.000159 ***
## female      -0.280401   0.037296  -7.518 2.45e-13 ***
## nonwhite    -0.002419   0.059791  -0.040 0.967738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4138 on 520 degrees of freedom
## Multiple R-squared:  0.3996, Adjusted R-squared:  0.3939
## F-statistic: 69.23 on 5 and 520 DF,  p-value: < 2.2e-16
```

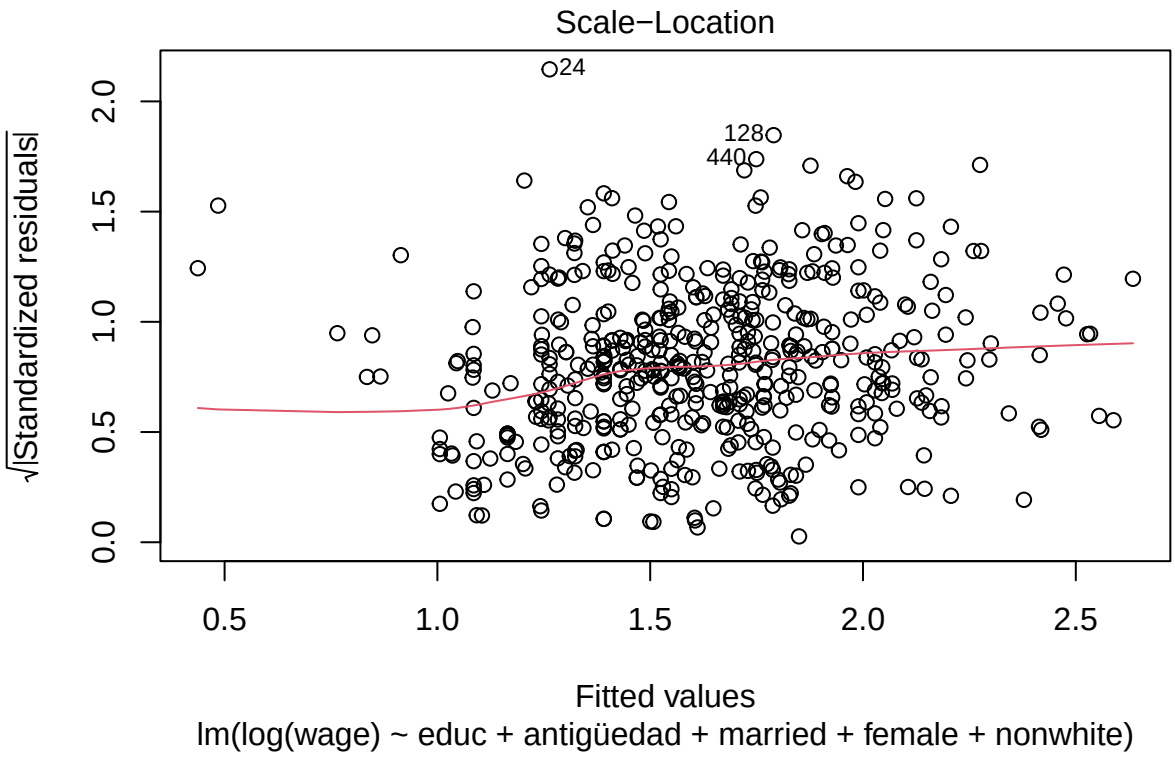
Interpretación:

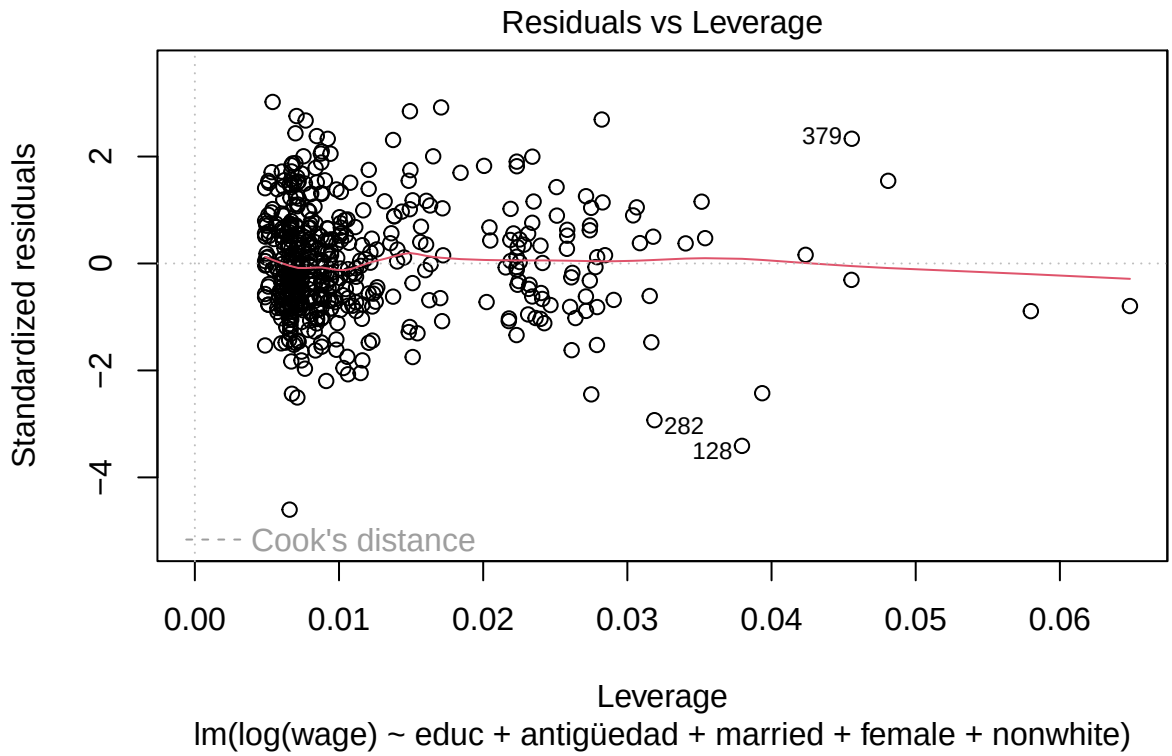
Coeficientes interpretados en términos logarítmicos: - Educación (0.079483): Un año adicional de educación se asocia con un aumento promedio del 7.95% en el salario, manteniendo constantes las demás variables. - Antigüedad (0.019443): Un año adicional de antigüedad se asocia con un aumento promedio del 1.94% en el salario. - Married (0.146703): Estar casado se asocia con un aumento promedio del 14.67% en el salario. - Female (-0.280401): Ser mujer se asocia con una disminución promedio del 28.04% en el salario. - Nonwhite (-0.002419): No es significativo. - Mejora en los supuestos del modelo: - La transformación logarítmica puede ayudar a corregir problemas de heterocedasticidad y normalidad de los residuos.

```r
plot(mod5)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log(wage) ~ educ + antigüedad + married + female + nonwhite)
440
128
24
1
0
−1
−2
0.5
1.0
1.5
2.0
2.5


Q−Q Residuals
Standardized residuals
2
0
−2
−4
440
128
24
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(log(wage) ~ educ + antigüedad + married + female + nonwhite)


Scale−Location
24
128
440
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.5
1.0
1.5
2.0
2.5
Fitted values
lm(log(wage) ~ educ + antigüedad + married + female + nonwhite)



Estos gráficos se ven **distintos** a los anteriores. Los residuos parecen dispersarse de manera más uniforme alrededor de cero. Hagamos varias pruebas para verificar:

```
#library(lmtest)
bptest(mod5)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mod5
## BP = 11.928, df = 5, p-value = 0.03579
```

```
shapiro.test(mod5$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod5$residuals
## W = 0.98592, p-value = 5.75e-05
```

Sigue rechazándose la nula, aunque por menos margen que antes en el caso de Breusch-Pagan (y menos mejorado en Shapiro-Wilk).

Haremos unos últimos intentos de ajuste, como el uso de errores estándar robustos (ajustan las estimaciones de la varianza para corregir la heterocedasticidad sin cambiar los coeficientes estimados).

```r
# Instalar y cargar los paquetes necesarios
#install.packages("lmtest")
#install.packages("sandwich")
library(lmtest)
library(sandwich)

# Recalcular los errores estándar usando la matriz de varianza-covarianza robusta
coef(summary(mod5))
```

```
##                 Estimate  Std. Error     t value     Pr(>|t|)
## (Intercept)  0.570834892 0.092936693  6.14219071 1.618659e-09
## educ         0.079482847 0.006604044 12.03548155 1.330233e-29
## antigüedad   0.019442804 0.002621755  7.41594863 4.940979e-13
## married      0.146703436 0.038558103  3.80473683 1.588325e-04
## female      -0.280400745 0.037295927 -7.51826714 2.448824e-13
## nonwhite    -0.002419444 0.059791466 -0.04046471 9.677382e-01
```

```r
coeftest(mod5, vcov = vcovHC(mod5, type = "HC1"))
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.5708349  0.1019869  5.5971 3.531e-08 ***
## educ         0.0794828  0.0072800 10.9180 < 2.2e-16 ***
## antigüedad   0.0194428  0.0031956  6.0842 2.272e-09 ***
## married      0.1467034  0.0400634  3.6618 0.0002761 ***
## female      -0.2804007  0.0378265 -7.4128 5.048e-13 ***
## nonwhite    -0.0024194  0.0614213 -0.0394 0.9685938
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Los coeficientes estimados permanecen iguales en este caso, pero los errores estándar y los valores p pueden cambiar. Para más información sobre este método recomiendo este artículo.

Si quisiéramos hacer otras transformaciones adicionales, no satisfechos con estos errores robustos, podríamos buscar una transformación Box-Cox, un tipo de transformación de potencia que redistribuye la variable con un logaritmo y luego eleva a un exponente óptimo para normalizar la variable. Para más información recomiendo este enlace.

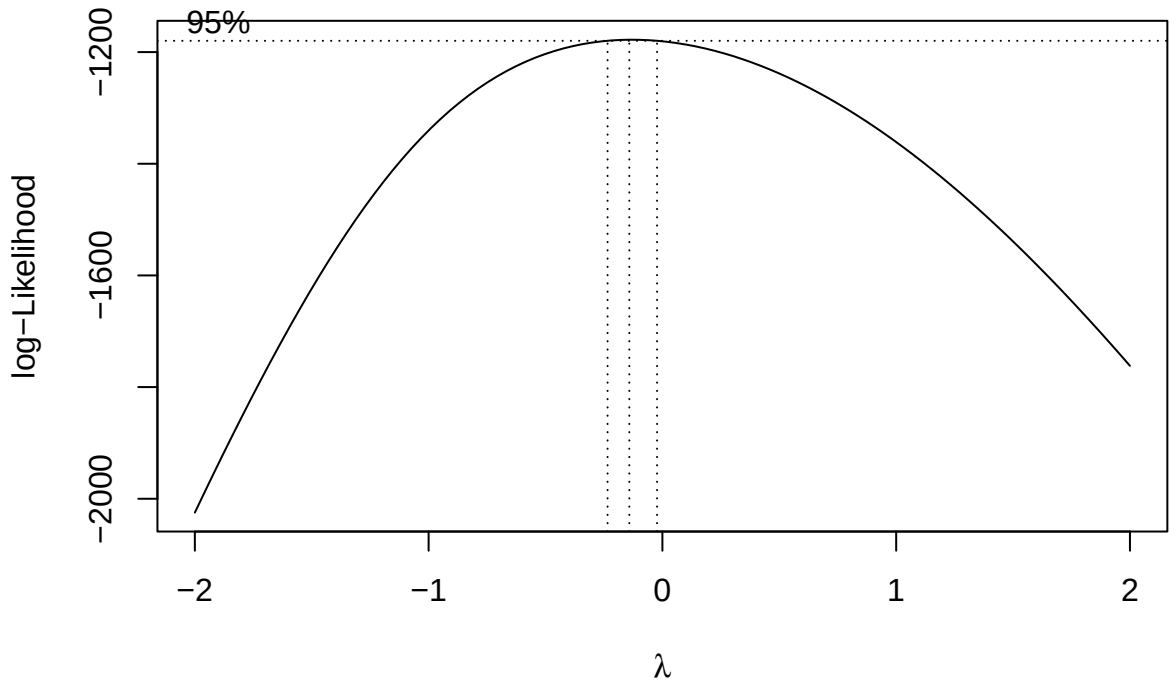
```r
# Instalar y cargar el paquete MASS
#install.packages("MASS")
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:plotly':
##
##     select
```

```
## The following object is masked from 'package:dplyr':
##
##     select

## The following object is masked from 'package:wooldridge':
##
##     cement
```

```
# Encontrar el lambda óptimo para la transformación Box-Cox
boxcox_mod <- boxcox(mod4, plotit = TRUE)
```




```
lambda_optimo <- boxcox_mod$x[which.max(boxcox_mod$y)]
lambda_optimo
```

```
## [1] -0.1414141
```

```
# Aplicar la transformación Box-Cox al salario
base1$wage_boxcox <- (base1$wage^lambda_optimo - 1) / lambda_optimo

# Reestimar el modelo con la variable transformada
mod_boxcox <- lm(wage_boxcox ~ educ + antigüedad + married + female + nonwhite, data = base1)
summary(mod_boxcox)
```

```
##
```

```
## Call:
## lm(formula = wage_boxcox ~ educ + antigüedad + married + female +
##     nonwhite, data = base1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81512 -0.21056 -0.02169  0.20059  0.90536
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.612834   0.073483   8.340 6.71e-16 ***
## educ         0.062016   0.005222  11.877  < 2e-16 ***
## antigüedad   0.014772   0.002073   7.126 3.47e-12 ***
## married      0.120575   0.030487   3.955 8.72e-05 ***
## female      -0.220928   0.029489  -7.492 2.94e-13 ***
## nonwhite    -0.001225   0.047276  -0.026    0.979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3272 on 520 degrees of freedom
## Multiple R-squared:  0.3941, Adjusted R-squared:  0.3883
## F-statistic: 67.64 on 5 and 520 DF,  p-value: < 2.2e-16
```

```
# Verificar los supuestos nuevamente
shapiro.test(mod_boxcox$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod_boxcox$residuals
## W = 0.98056, p-value = 1.786e-06
```

```
bptest(mod_boxcox)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mod_boxcox
## BP = 8.7255, df = 5, p-value = 0.1205
```

No rechazamos la hipótesis nula de homocedasticidad por Breusch-Pagan. La heterocedasticidad se ha mitigado en este modelo. Sin embargo, rechazamos la hipótesis nula de normalidad. Los residuos aún no siguen una distribución normal.

También podríamos incluir otros términos polinomiales o interacciones.

```
# Modelo con términos cuadráticos
mod_poly <- lm(log(wage) ~ educ + I(educ^2) + antigüedad + I(antigüedad^2) + married + female + nonwhite

# Resumen del modelo
summary(mod_poly)
```

```
##
## Call:
## lm(formula = log(wage) ~ educ + I(educ^2) + antigüedad + I(antigüedad^2) +
##     married + female + nonwhite, data = base1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86343 -0.25631 -0.02747  0.24719  1.26424
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.0890157  0.1836224   5.931 5.52e-09 ***
## educ            -0.0200558  0.0298881  -0.671 0.502501
## I(educ^2)        0.0042157  0.0012474   3.380 0.000780 ***
## antigüedad       0.0407787  0.0064974   6.276 7.34e-10 ***
## I(antigüedad^2) -0.0008442  0.0002312  -3.652 0.000287 ***
## married          0.1264775  0.0381141   3.318 0.000969 ***
## female          -0.2668000  0.0367257  -7.265 1.38e-12 ***
## nonwhite        -0.0262610  0.0586796  -0.448 0.654678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4048 on 518 degrees of freedom
## Multiple R-squared:  0.4277, Adjusted R-squared:   0.42
## F-statistic: 55.31 on 7 and 518 DF,  p-value: < 2.2e-16
```

```r
# Verificar los supuestos
shapiro.test(mod_poly$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod_poly$residuals
## W = 0.98657, p-value = 9.058e-05
```

```r
bptest(mod_poly)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mod_poly
## BP = 11.462, df = 7, p-value = 0.1197
```

Tampoco rechazamos en esta transformación la hipótesis nula de homoscedasticidad. La heteroscedasticidad se ha reducido. Pero sí volvemos a notar que los errores no son normales. Soluciones como bootstrapping podrían ser útiles, así como modelos generalizados. Por ahora, podemos enfocarnos en sacar las tablas de los modelos, para uso en LaTeXo para otros programas.

```r
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
stargazer(mod1,mod2,mod3,mod4,mod5,mod_boxcox) #de base, para LaTeX
```

```
##
## % Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac a
## % Date and time: Fri, Oct 10, 2025 - 12:47:54
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lccccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{6}{c}{\textit{Dependent variable:}} \\
## \cline{2-7}
## \\[-1.8ex] & \multicolumn{4}{c}{wage} & log(wage) & wage\_boxcox \\
## \\[-1.8ex] & (1) & (2) & (3) & (4) & (5) & (6)\\
## \hline \\[-1.8ex]
##  educ & 0.541$^{***}$ & 0.569$^{***}$ & 0.556$^{***}$ & 0.529$^{***}$ & 0.079$^{***}$ & 0.062$^{***}$
##   & (0.053) & (0.049) & (0.049) & (0.047) & (0.007) & (0.005) \\
##   & & & & & & \\
##  antigüedad &  & 0.190$^{***}$ & 0.175$^{***}$ & 0.154$^{***}$ & 0.019$^{***}$ & 0.015$^{***}$ \\
##   &  & (0.019) & (0.019) & (0.019) & (0.003) & (0.002) \\
##   & & & & & & \\
##  married &  &  & 0.892$^{***}$ & 0.683$^{**}$ & 0.147$^{***}$ & 0.121$^{***}$ \\
##   &  &  & (0.283) & (0.275) & (0.039) & (0.030) \\
##   & & & & & & \\
##  female &  &  &  & $-$1.711$^{***}$ & $-$0.280$^{***}$ & $-$0.221$^{***}$ \\
##   &  &  &  & (0.266) & (0.037) & (0.029) \\
##   & & & & & & \\
##  nonwhite &  &  &  & $-$0.065 & $-$0.002 & $-$0.001 \\
##   &  &  &  & (0.427) & (0.060) & (0.047) \\
##   & & & & & & \\
##  Constant & $-$0.905 & $-$2.222$^{***}$ & $-$2.526$^{***}$ & $-$1.123$^{*}$ & 0.571$^{***}$ & 0.613$
##   & (0.685) & (0.640) & (0.642) & (0.664) & (0.093) & (0.073) \\
##   & & & & & & \\
## \hline \\[-1.8ex]
## Observations & 526 & 526 & 526 & 526 & 526 & 526 \\
## R$^{2}$ & 0.165 & 0.302 & 0.315 & 0.365 & 0.400 & 0.394 \\
## Adjusted R$^{2}$ & 0.163 & 0.299 & 0.311 & 0.359 & 0.394 & 0.388 \\
## Residual Std. Error & 3.378 (df = 524) & 3.092 (df = 523) & 3.066 (df = 522) & 2.956 (df = 520) & 0.4
## F Statistic & 103.363$^{***}$ (df = 1; 524) & 113.067$^{***}$ (df = 2; 523) & 79.978$^{***}$ (df = 3
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{6}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

```
stargazer(mod1,mod2,mod3, type="text", title = "Resultados de regresión")
```

```
##
```

```
## Resultados de regresión
## ================================================================================================
##                                                Dependent variable:
##                     ---------------------------------------------------------------------------
##                                                       wage
##                                (1)                      (2)                      (3)
## ------------------------------------------------------------------------------------------------
## educ                         0.541***                 0.569***                 0.556***
##                              (0.053)                  (0.049)                  (0.049)
##
## antigüedad                                            0.190***                 0.175***
##                                                       (0.019)                  (0.019)
##
## married                                                                        0.892***
##                                                                                (0.283)
##
## Constant                     -0.905                   -2.222***                -2.526***
##                              (0.685)                  (0.640)                  (0.642)
##
## ------------------------------------------------------------------------------------------------
## Observations                   526                      526                      526
## R2                           0.165                    0.302                    0.315
## Adjusted R2                  0.163                    0.299                    0.311
## Residual Std. Error     3.378 (df = 524)         3.092 (df = 523)         3.066 (df = 522)
## F Statistic         103.363*** (df = 1; 524) 113.067*** (df = 2; 523) 79.978*** (df = 3; 522)
## ================================================================================================
## Note:                                                                *p<0.1; **p<0.05; ***p<0.01
```

```
stargazer(mod4,mod5,mod_boxcox, type="text", title = "Resultados de regresión 2")#de base, para LaTeX
```

```
##
## Resultados de regresión 2
## ===================================================================
##                                        Dependent variable:
##                                ---------------------------------
##                                  wage    log(wage) wage_boxcox
##                                   (1)       (2)        (3)
## -------------------------------------------------------------------
## educ                            0.529***  0.079***   0.062***
##                                  (0.047)   (0.007)    (0.005)
##
## antigüedad                      0.154***  0.019***   0.015***
##                                  (0.019)   (0.003)    (0.002)
##
## married                          0.683**  0.147***   0.121***
##                                  (0.275)   (0.039)    (0.030)
##
## female                          -1.711*** -0.280***  -0.221***
##                                  (0.266)   (0.037)    (0.029)
##
## nonwhite                         -0.065    -0.002     -0.001
##                                  (0.427)   (0.060)    (0.047)
##
## Constant                         -1.123*  0.571***   0.613***
```

```
##                                 (0.664)   (0.093)    (0.073)
##
## ---------------------------------------------------------------
## Observations                     526       526        526
## R2                              0.365     0.400      0.394
## Adjusted R2                     0.359     0.394      0.388
## Residual Std. Error (df = 520)  2.956     0.414      0.327
## F Statistic (df = 5; 520)      59.848*** 69.227***  67.641***
## ===============================================================
## Note:                               *p<0.1; **p<0.05; ***p<0.01
```

Modelos lineales generalizados

Al trabajar con datos como los de salarios en el conjunto de datos de Wooldridge, podemos encontrar problemas con las suposiciones fundamentales de los Mínimos Cuadrados Ordinarios (OLS). Aunque los modelos lineales clásicos nos proporcionan resultados, estos pueden ser problemáticos si los errores no están bien distribuidos, es decir, si violan los supuestos de normalidad y homocedasticidad. Esto significa que las inferencias obtenidas pueden no ser adecuadas o estar sesgadas.

Para abordar estas limitaciones, los Modelos Lineales Generalizados (GLM) extienden el marco de los modelos lineales al permitir una mayor flexibilidad en la relación entre las variables independientes y la variable dependiente. Un GLM se compone de tres elementos clave:

1. Predictor lineal (η): $\eta = X\beta$

Donde $\boldsymbol{X}$ es la matriz de variables independientes y es el vector de coeficientes.

2. Función de Enlace (g): Esta es monótona y diferenciable en todo su dominio, y que transforma el predictor lineal $g(\mu) = \eta$. Su inversa permite obtener las predicciones de la variable dependiente: $y = g^{-1}(X\beta)$, $\hat{y} = g^{-1}(\eta)$).
3. Distribución de Respuesta: La variable dependiente se asume que sigue una distribución de la familia exponencial, denotada por $f(y|\mu)$. Algunas distribuciones comunes incluyen la normal, binomial, Poisson y gamma.

Estos componentes proporcionan la flexibilidad necesaria para modelar diferentes tipos de variables dependientes, permitiendo que el modelo capture mejor las características de los datos y aborde problemas como la heterocedasticidad y la no normalidad de los errores, así como cuando el rango de la variable de respuesta está limitado, entre otros.

En R, la función glm() se utiliza para ajustar Modelos Lineales Generalizados. Esta función permite especificar tanto la distribución de la variable dependiente como la función de enlace adecuada.

Sintaxis básica de glm()

```
glm(formula, family = family_type(link = link_function), data = dataset)
```

- `formula`: Especifica la relación entre las variables dependiente e independientes (similar a `lm()`).
- `family`: Define la distribución de la variable dependiente. Puede ser `gaussian` (para regresión lineal), `binomial` (para regresión logística), `poisson` (para modelos de conteo), `gamma`, entre otros.
- `link`: Es la función de enlace que conecta la media de la variable dependiente con las variables independientes (por ejemplo, `log`, `identity` o `inverse`).

La función de enlace predeterminada para una familia puede cambiarse especificando un enlace a la función de familia. Si no se informa nada, el modelo correrá en la práctica un modelo lineal simple. Comparen los resultados:

```
glm(wage ~ educ + antigüedad + married, data = base1)
```

```
##
## Call:  glm(formula = wage ~ educ + antigüedad + married, data = base1)
##
## Coefficients:
## (Intercept)         educ   antigüedad      married
##     -2.5258       0.5561       0.1748       0.8924
##
## Degrees of Freedom: 525 Total (i.e. Null);  522 Residual
## Null Deviance:       7160
## Residual Deviance: 4906  AIC: 2677
```

```
lm(wage ~ educ + antigüedad + married, data = base1)
```

```
##
## Call:
## lm(formula = wage ~ educ + antigüedad + married, data = base1)
##
## Coefficients:
## (Intercept)         educ   antigüedad      married
##     -2.5258       0.5561       0.1748       0.8924
```

Por ejemplo, si la variable de respuesta es no negativa y la varianza es proporcional a la media, se usaría la función de enlace "identity" con la familia "quasipoisson". Esto se especificaría como:

```
family = quasipoisson(link = "identity")
```

La decisión sobre qué familia es apropiada no se discute a profundidad en esta secuencia, pero estos pueden ser:

```
binomial(link = "logit")
gaussian(link = "identity")
Gamma(link = "inverse")
inverse.gaussian(link = "1/mu^2")
poisson(link = "log")
quasi(link = "identity", variance = "constant")
quasibinomial(link = "logit")
quasipoisson(link = "log")
```

- gaussian: Para variables continuas que siguen una distribución normal (equivalente a la regresión lineal clásica).
- binomial: Para variables categóricas binarias (regresión logística).
- poisson: Para datos de conteo (números enteros no negativos).
- Gamma: Para variables continuas y positivas, especialmente cuando la varianza aumenta con la media.
- inverse.gaussian: Para variables continuas positivas con varianza que aumenta rápidamente con la media.

Algunas funciones de enlace comunes son: - identity: Sin transformación (usada en regresión lineal). - log: Transforma la media mediante el logaritmo natural (útil para variables positivas). - logit: Función logística, usada en regresión logística. - inverse: Utiliza la inversa de la media.

Cada familia tiene una función de enlace predeterminada, pero puede modificarse según las necesidades del análisis. Revisar la página de ayuda de `glm` y la documentacion de objetos familiares para modelos ayudará en gran medida.

Volviendo al ejemplo que traíamos antes, queremos modelar el salario (wage), que es una variable continua y positiva, y sospechamos que la varianza aumenta con la media. Podemos utilizar la familia Gamma con una función de enlace logarítmica, a través de `family = Gamma(link = "log")`:

```
# Ajustando un modelo GLM con distribución gamma y enlace logarítmico
mod_glm <- glm(wage ~ educ + antigüedad + married + female + nonwhite,
               family = Gamma(link = "log"), data = base1)

# Resumen del modelo
summary(mod_glm)
```

```
##
## Call:
## glm(formula = wage ~ educ + antigüedad + married + female +
##     nonwhite, family = Gamma(link = "log"), data = base1)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6336436  0.1007043   6.292 6.65e-10 ***
## educ         0.0817722  0.0071560  11.427  < 2e-16 ***
## antigüedad   0.0217897  0.0028409   7.670 8.53e-14 ***
## married      0.1220276  0.0417808   2.921  0.00364 **
## female      -0.2856600  0.0404131  -7.068 5.07e-12 ***
## nonwhite    -0.0003761  0.0647888  -0.006  0.99537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2010781)
##
##     Null deviance: 158.876  on 525  degrees of freedom
## Residual deviance:  90.711  on 520  degrees of freedom
## AIC: 2305
##
## Number of Fisher Scoring iterations: 6
```

Los coeficientes representan el efecto multiplicativo de las variables independientes sobre el salario. Por pasos:

El intercepto se interpreta como el valor cuando las variables independientes son cero. En este caso, el logaritmo del salario esperado es 0.6336. - Exponenciando el intercepto: $exp(0.6336) \approx 1.884$. - Esto significa que, para una persona con cero años de educación y antigüedad, no casada, hombre y blanco, el salario promedio esperado es aproximadamente $1.88 por hora (recordemos, esto es de 1976).

Cada año adicional de educación se asocia con un incremento en el logaritmo del salario de 0.08177. Los coeficientes se han de exponenciar (y son multiplicativos, entendiéndose como cambios porcentuales por unidad adicional).

```r
exp(coef(mod_glm))
```

```
## (Intercept)        educ  antigüedad     married      female    nonwhite
##   1.8844643   1.0852086   1.0220288   1.1297853   0.7515181   0.9996239
```

Por ejemplo, por cada año adicional de educación, el salario promedio aumenta en aproximadamente un 8.53%, manteniendo constantes las demás variables. En el caso de la variable 'dummy', como female, el exponenciado es menos que uno. Esto implica que las mujeres ganaban, en promedio, un 24.82% menos que los hombres, manteniendo constantes las demás variables en el modelo. Otros elementos en las columnas del estimado son similares a lo que vimos al usar `lm()`.

Sin embargo tenemos información adicional. Hay un parámetro de dispersión. Este es un estimado de la varianza de los residuos en el modelo Gamma. Un valor más pequeño indica menor variabilidad de los datos alrededor del modelo ajustado. La raíz cuadrada de ese parámetro es la desviación estándar estimada. Las desviaciones nulas y residuales son comparaciones entre un modelo vacío (con sólo un intercepto), y el modelo ajustado con todas las variables.

En este caso reporta el criterio de información de Akaike, que es una medida de la calidad del modelo que penaliza por la complejidad (número de parámetros). Al comparar modelos, un AIC más bajo indica un modelo preferido. Como sólo tenemos un modelo, el AIC nos sirve para comparar con futuros modelos alternativos. Finalmente, declara la cantidad de Iteraciones que fueron necesarias para converger y encontrar los estimadores de máxima verosimilitud.

Verifiquemos otra vez con el modelo transformado

```r
# Resumen del modelo
summary(mod5)
```

```
##
## Call:
## lm(formula = log(wage) ~ educ + antigüedad + married + female +
##     nonwhite, data = base1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89855 -0.27105 -0.03577  0.24487  1.24663
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.570835   0.092937   6.142 1.62e-09 ***
## educ         0.079483   0.006604  12.035  < 2e-16 ***
## antigüedad   0.019443   0.002622   7.416 4.94e-13 ***
## married      0.146703   0.038558   3.805 0.000159 ***
## female      -0.280401   0.037296  -7.518 2.45e-13 ***
## nonwhite    -0.002419   0.059791  -0.040 0.967738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4138 on 520 degrees of freedom
## Multiple R-squared:  0.3996, Adjusted R-squared:  0.3939
## F-statistic: 69.23 on 5 and 520 DF,  p-value: < 2.2e-16
```

Notamos que la transformación aproxima los coeficientes a los conseguidos con el `glm()`.

```r
library(arm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: lme4
```

```
##
## arm (Version 1.14-4, built: 2024-4-1)
```

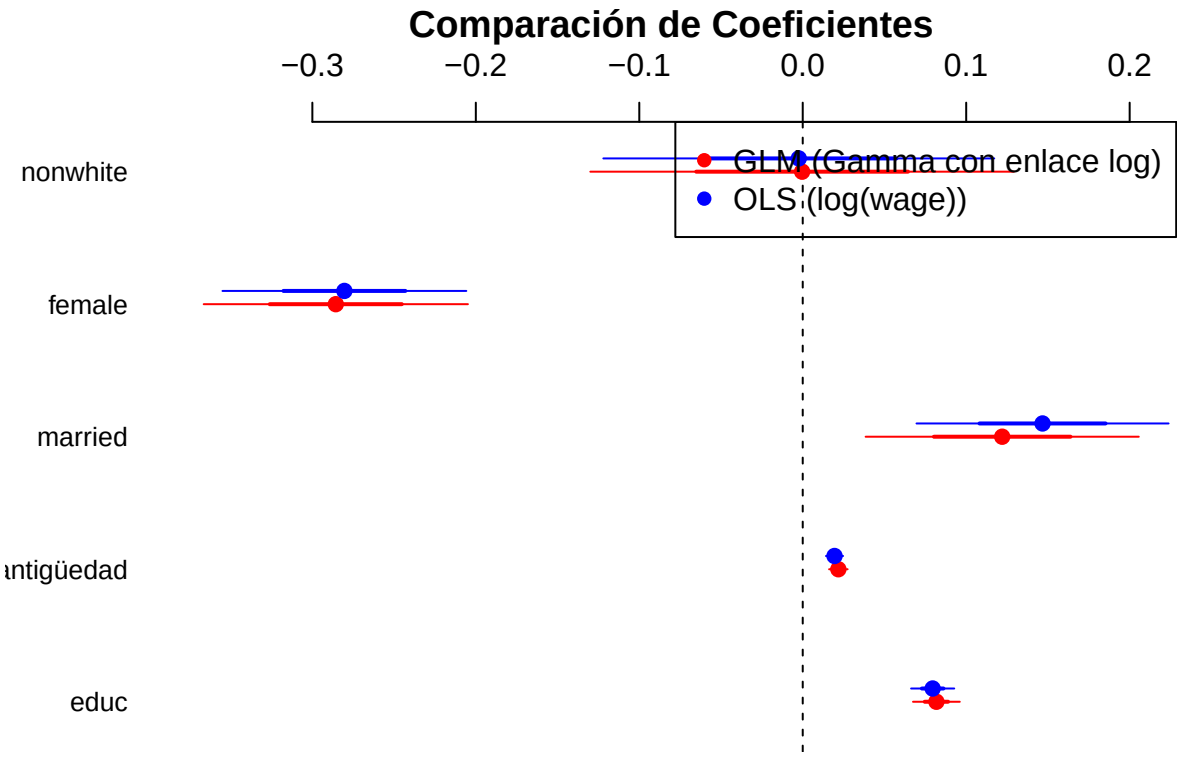
```
## Working directory is /Users/rashid/Library/CloudStorage/OneDrive-UniversityofPuertoRico/Universidad
```

```
##
## Attaching package: 'arm'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

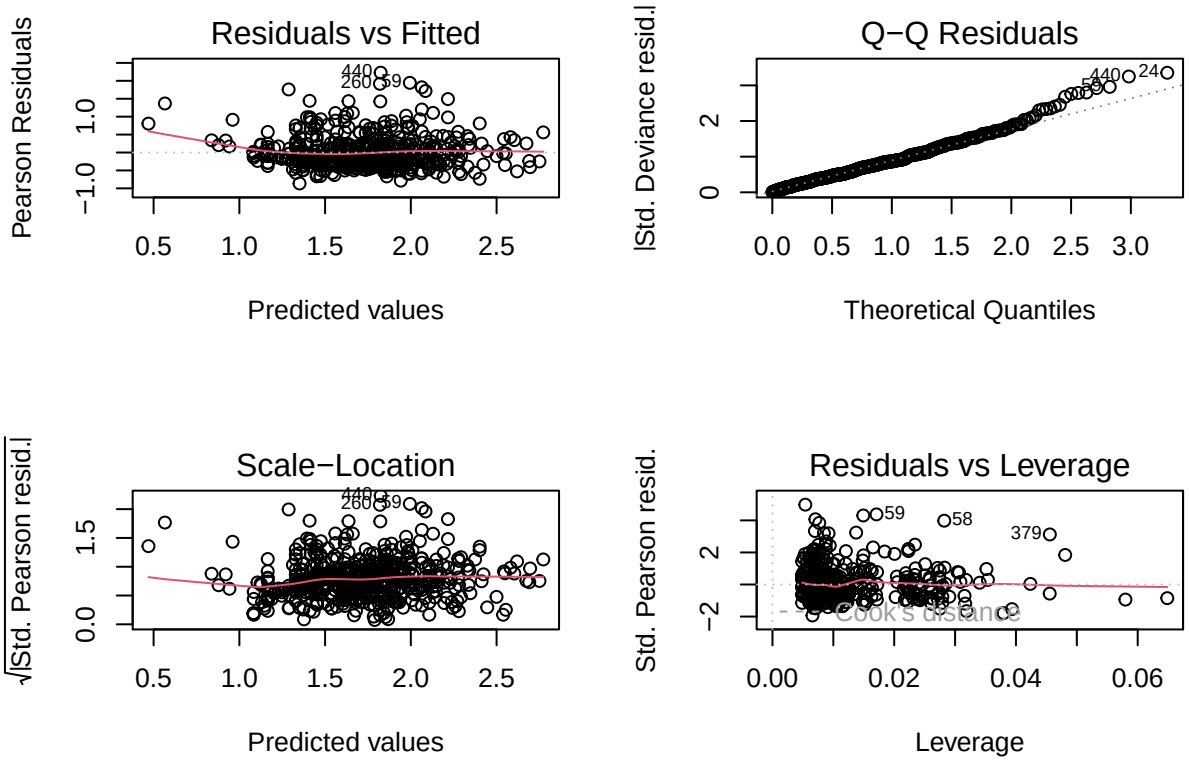
```
## The following object is masked from 'package:jtools':
##
##     standardize
```

```r
# Gráfico de coeficientes para el modelo GLM
coefplot(mod_glm, col.pts = "red", cex.pts = 1.5, main = "Comparación de Coeficientes")

# Añadimos los coeficientes del modelo OLS al mismo gráfico
coefplot(mod5, add = TRUE, col.pts = "blue", cex.pts = 1.5)

legend("topright", legend = c("GLM (Gamma con enlace log)", "OLS (log(wage))"),
       col = c("red", "blue"), pch = 16)
```



Análisis de la bondad del modelo

```
# Generamos los gráficos diagnósticos
par(mfrow = c(2, 2))
plot(mod_glm)
```



Esto se ve posiblemente mejor que la variante de OLS, aunque tampoco parece perfecto.

Modelos jerárquicos

Los modelos jerárquicos, también conocidos como modelos lineales mixtos o modelos multinivel, son una extensión de los modelos lineales que permiten analizar datos en los que las observaciones están agrupadas o anidadas en diferentes niveles. Estos modelos son especialmente útiles cuando se espera que exista correlación entre las observaciones dentro de los mismos grupos.

En este análisis, utilizaremos el paquete `lme4` en R, que es ampliamente utilizado para ajustar modelos lineales mixtos. A continuación, te guiaré a través de los pasos para ajustar y comprender estos modelos, explicando cada componente y resultado de manera clara. La sintaxis de `lme4` se basa en la sintaxis de los modelos lineales que ya conocemos de `lm()`.

La función `lmer()` de `lme4` añade la especificación de la variable de grupo/sujeto y de la estructura de efectos aleatorios que se van a estimar. En paréntesis adicionales (ver abajo), el término a la izquierda de `|` especifica los efectos aleatorios que se van a estimar. El término a la derecha de `|` representa la(s) variable(s) que definen la estructura de agrupación (o anidamiento) de los datos.

Un `1` en la parte izquierda del paréntesis significa que se debe estimar un componente de varianza de intercepto aleatorio. Si también se coloca la variable predictora a la izquierda de `|`, esto indica que se deben incluir pendientes aleatorias. La forma básica de estos será:

```
lmer(data = datos, VarDependiente ~ VarIndependiente + (1 | VarGrupal))
```

Corresponde a un modelo que puede describirse de la siguiente manera: “La variable dependiente es predicha

por la variable independiente. Al mismo tiempo, la varianza de los residuos de nivel 2 del intercepto es un parámetro del modelo".

Comencemos con modelos HLM que incluyen solo variables predictoras de nivel 1 (individuos anidados en grupos). Luego, en un segundo paso, añadiremos predictores de nivel 2. Finalmente, analizaremos los modelos HLM de medidas repetidas, donde el nivel más bajo (nivel 1) corresponde a observaciones repetidas dentro de los individuos, y esas observaciones están anidadas en los participantes individuales (nivel 2).

Nos limitaremos aquí a modelos de 2 niveles. Los principios de los modelos HLM pueden ilustrarse de manera bastante parsimoniosa de esta forma, y expandir los modelos a más de dos niveles de análisis es bastante sencillo.

```
# Cargar las librerías necesarias
#library(tidyverse)
library(lme4)
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step
```

```
# Cargar los datos desde una URL
df <- read_csv("https://raw.githubusercontent.com/methodenlehre/data/master/salary-data.csv")
```

```
## Rows: 600 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): firma, sector
## dbl (2): experience, salary
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Convertir las variables 'firma' y 'sector' en factores
df <- df |>
  mutate(firma = as.factor(firma),
         sector = as.factor(sector))

# Ver las últimas filas del conjunto de datos
tail(df)
```

```
## # A tibble: 6 x 4
##   firma    experience salary sector
##   <fct>         <dbl>  <dbl> <fct>
```

```
## 1 Firma 20       3.58  6838. Privat
## 2 Firma 20       3.18  7604. Privat
## 3 Firma 20       3.39  5714. Privat
## 4 Firma 20       7.12 10089. Privat
## 5 Firma 20       2.98  6940. Privat
## 6 Firma 20       6.45  9330. Privat
```

Modelo nulo

En este caso saco unos datos de salarios de compañías ficticias en Suiza. Con `mutate()` convertí las variables firma (empresa) y sector en factores, ya que representan categorías. Antes de añadir predictores, ajustamos un modelo nulo que solo incluye el intercepto y un término aleatorio para capturar la variabilidad entre las empresas (firma). Este modelo nos permite estimar la correlación intraclase (ICC) y entender qué proporción de la variabilidad total del salario se debe a diferencias entre empresas.

```
library(Matrix)
library(lme4)
# Ajustar el modelo nulo con intercepto aleatorio por empresa
modelo_nulo <- lmer(salary ~ 1 + (1 | firma), data = df, REML = TRUE)

# Obtener las predicciones del modelo nulo
df$predicciones_nulo <- predict(modelo_nulo)

# Resumen del modelo
summary(modelo_nulo)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: salary ~ 1 + (1 | firma)
##    Data: df
##
## REML criterion at convergence: 10433.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9816 -0.6506 -0.0494  0.5779  4.2131
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  firma    (Intercept)  851249   922.6
##  Residual             1954745  1398.1
## Number of obs: 600, groups:  firma, 20
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   8737.6      214.1   19.0   40.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Explicación:

- lmer(): Ajusta un modelo lineal mixto.

- `salary ~ 1`: Indica que solo se ajusta el intercepto fijo (media general del salario).
- `(1 | firma)`: Especifica un intercepto aleatorio para cada empresa (firma), capturando la variabilidad entre empresas.
- `REML = TRUE`: Utiliza el método de Máxima Verosimilitud Restringida para estimar los parámetros, lo cual es apropiado cuando se comparan modelos con diferentes efectos fijos.
- `predict()`: Genera predicciones del modelo para cada observación.

La varianza del intercepto aleatorio (firma) representa la variabilidad del salario promedio entre empresas. Por otro lado la varianza residual captura la variabilidad del salario dentro de las empresas. Calcularemos la correlación intraclase como la proporción de la varianza total que se debe a diferencias entre empresas. Se define como: $\rho = \frac{\sigma^2_{\text{Nivel-2}}}{\sigma^2_{\text{Nivel-2}} + \sigma^2_{\text{Nivel-1}}}$

```
# Extraer las varianzas del modelo
var_intercepto <- as.numeric(VarCorr(modelo_nulo)$firma[1])
var_residual <- attr(VarCorr(modelo_nulo), "sc")^2

# Calcular la ICC
ICC <- var_intercepto / (var_intercepto + var_residual)
ICC
```

```
## [1] 0.303368
```

Correlación intraclase: $\hat{\rho} = \frac{\hat{\sigma}^2_{v_0}}{\hat{\sigma}^2_{v_0} + \hat{\sigma}^2_{\epsilon}} = \frac{851249}{851249 + 1954745} = 0.3034$

Es decir, 30% de la variabilidad total del salario se debe a diferencias entre empresas. Existe una correlación notable entre los salarios de los empleados dentro de la misma empresa, lo que justifica el uso de un modelo multinivel.

Con la función ranef() podemos ver los efectos aleatorios (residuos de segundo nivel del intercepto):

```
ranef(modelo_nulo)
```

```
## $firma
##          (Intercept)
## Firma 01   789.65416
## Firma 02  1105.01807
## Firma 03  1923.02618
## Firma 04 -1136.50080
## Firma 05   953.54595
## Firma 06  -958.93185
## Firma 07   -10.14627
## Firma 08  -254.84048
## Firma 09  -651.31802
## Firma 10   768.48590
## Firma 11  -506.26620
## Firma 12   940.40709
## Firma 13  -742.84383
## Firma 14  -975.45482
## Firma 15 -1161.36931
## Firma 16   -97.68008
## Firma 17   661.96052
## Firma 18  -168.11195
## Firma 19   351.23926
```

```
## Firma 20  -829.87351
##
## with conditional variances for "firma"
```

Para determinar si la varianza entre empresas es estadísticamente significativa, comparamos el modelo nulo con un modelo sin efectos aleatorios (modelo lineal simple).

```
# Utilizar la función ranova() para comparar modelos
anova_aleatorio <- ranova(modelo_nulo)
anova_aleatorio
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## salary ~ (1 | firma)
##             npar  logLik   AIC    LRT Df Pr(>Chisq)
## <none>         3 -5216.7 10440
## (1 | firma)    2 -5295.5 10595 157.44  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Este comparó el modelo actual con un modelo reducido que elimina el efecto aleatorio. El valor p indica que incluir el intercepto aleatorio para firma mejora significativamente el modelo. Aproximadamente, el 30% de la varianza total del salario se puede atribuir a diferencias entre empresas.

Modelo con predictor de primer nivel

Ahora, añadimos un predictor de nivel 1 (experience), que es una variable individual, al modelo. Mantenemos el intercepto aleatorio para capturar la variabilidad entre empresas.

```
# Ajustar el modelo con 'experience' como predictor fijo y intercepto aleatorio por empresa
modelo_intercepto_aleatorio <- lmer(salary ~ experience + (1 | firma), data = df, REML = TRUE)

# Obtener las predicciones del modelo
df$predicciones_intercepto_aleatorio <- predict(modelo_intercepto_aleatorio)

# Resumen del modelo
summary(modelo_intercepto_aleatorio)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: salary ~ experience + (1 | firma)
##    Data: df
##
## REML criterion at convergence: 10127.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8109 -0.6884  0.0005  0.5980  3.8833
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev.
##  firma    (Intercept)  614367   783.8
##  Residual             1184502  1088.3
## Number of obs: 600, groups:  firma, 20
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  5964.18     229.41   48.00   26.00   <2e-16 ***
## experience    534.34      27.21  589.48   19.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## experience -0.615
```

Los resultados del modelo mixto muestran que el efecto fijo de la experiencia es significativamente positivo, con un coeficiente estimado de $\hat{\gamma}_{10} = 534.34$ y un valor p menor a 0.001. Esto indica que, en promedio, el salario aumenta en aproximadamente 534 unidades (por ejemplo, francos suizos) por cada año adicional de experiencia, controlando por las diferencias entre empresas. El modelo incluye un intercepto aleatorio a nivel de empresa (firma), lo que nos permite ajustar diferencias entre las empresas en los salarios base.

Se le puede añadir un nivel adicional al modelo para tanto acomodar por un intercepto así como pendiente aleatoria. Al aplicar este modelo la varianza (aleatoria) de la pendiente de la variable independiente de primer nivel se puede estimar, dándonos una idea de las diferencias en el efecto (pendiente) de experiencia en salario entre las unidades del segundo nivel (firmas). Si comparamos los efectos entre este y el modelo anterior, vemos la disminución marginal de la varianza del intercepto aleatorio en comparación con el modelo nulo, lo que sugiere que parte de la variabilidad entre empresas se explica por la experiencia de los empleados.

```
fixef(modelo_intercepto_aleatorio)
```

```
## (Intercept)  experience
##   5964.1757    534.3446
```

```
ranef(modelo_intercepto_aleatorio)
```

```
## $firma
##          (Intercept)
## Firma 01   204.30371
## Firma 02   646.14732
## Firma 03  1492.00151
## Firma 04  -910.78990
## Firma 05   389.16512
## Firma 06  -924.63977
## Firma 07   577.66959
## Firma 08  -516.51767
## Firma 09  -638.24646
## Firma 10   768.48113
## Firma 11  -619.55111
## Firma 12  1091.33530
## Firma 13  -773.67207
## Firma 14  -738.17926
## Firma 15  -652.94087
```

```
## Firma 16    57.33923
## Firma 17   458.05487
## Firma 18   -89.38416
## Firma 19   944.22822
## Firma 20  -764.80474
##
## with conditional variances for "firma"
```

Modelo con pendientes aleatorias

Para investigar si el efecto de la experiencia en el salario varía entre empresas, ajustamos un modelo que incluye tanto intercepto como pendiente aleatorios para experience.

```
# Ajustar el modelo con intercepto y pendiente aleatorios por empresa
modelo_coeficientes_aleatorios <- lmer(salary ~ experience + (experience | firma), data = df, REML = TRU

# Obtener las predicciones del modelo
df$predicciones_coeficientes_aleatorios <- predict(modelo_coeficientes_aleatorios)

# Resumen del modelo
summary(modelo_coeficientes_aleatorios)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: salary ~ experience + (experience | firma)
##    Data: df
##
## REML criterion at convergence: 10117.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8307 -0.6804  0.0037  0.5999  3.3608
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  firma    (Intercept)  722588   850.1
##           experience    18392   135.6   -0.51
##  Residual             1136296  1066.0
## Number of obs: 600, groups:  firma, 20
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  5933.72     240.38   18.89   24.68 7.77e-16 ***
## experience    530.85      40.59   18.95   13.08 6.20e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr)
## experience -0.690
```

En este caso `(experience | firma)` especifica que tanto el intercepto como la pendiente de experience varían aleatoriamente entre empresas.

Los resultados muestran que el efecto fijo de la experiencia sigue siendo significativamente positivo, con un coeficiente estimado de $\hat{\gamma}_{10} = 530.85$ y un valor p menor a 0.001. Esto indica que, en promedio, el salario aumenta en aproximadamente 531 unidades (por ejemplo, CHF para estos datos ficticios) por cada año adicional de experiencia. Una covarianza negativa sugiere que las empresas con salarios base más altos tienden a tener incrementos más pequeños por año de experiencia, y viceversa.

Para evaluar si el modelo con pendientes aleatorias proporciona un mejor ajuste que el modelo con solo intercepto aleatorio, realizamos una comparación de modelos.

```
# Ajustar el modelo con intercepto aleatorio (modelo reducido)
modelo_intercepto_aleatorio <- lmer(salary ~ experience + (1 | firma), data = df, REML = FALSE)

# Ajustar el modelo con intercepto y pendiente aleatorios (modelo completo)
modelo_coeficientes_aleatorios <- lmer(salary ~ experience + (experience | firma), data = df, REML = FA

# Comparar los modelos utilizando ANOVA
anova(modelo_intercepto_aleatorio, modelo_coeficientes_aleatorios)
```

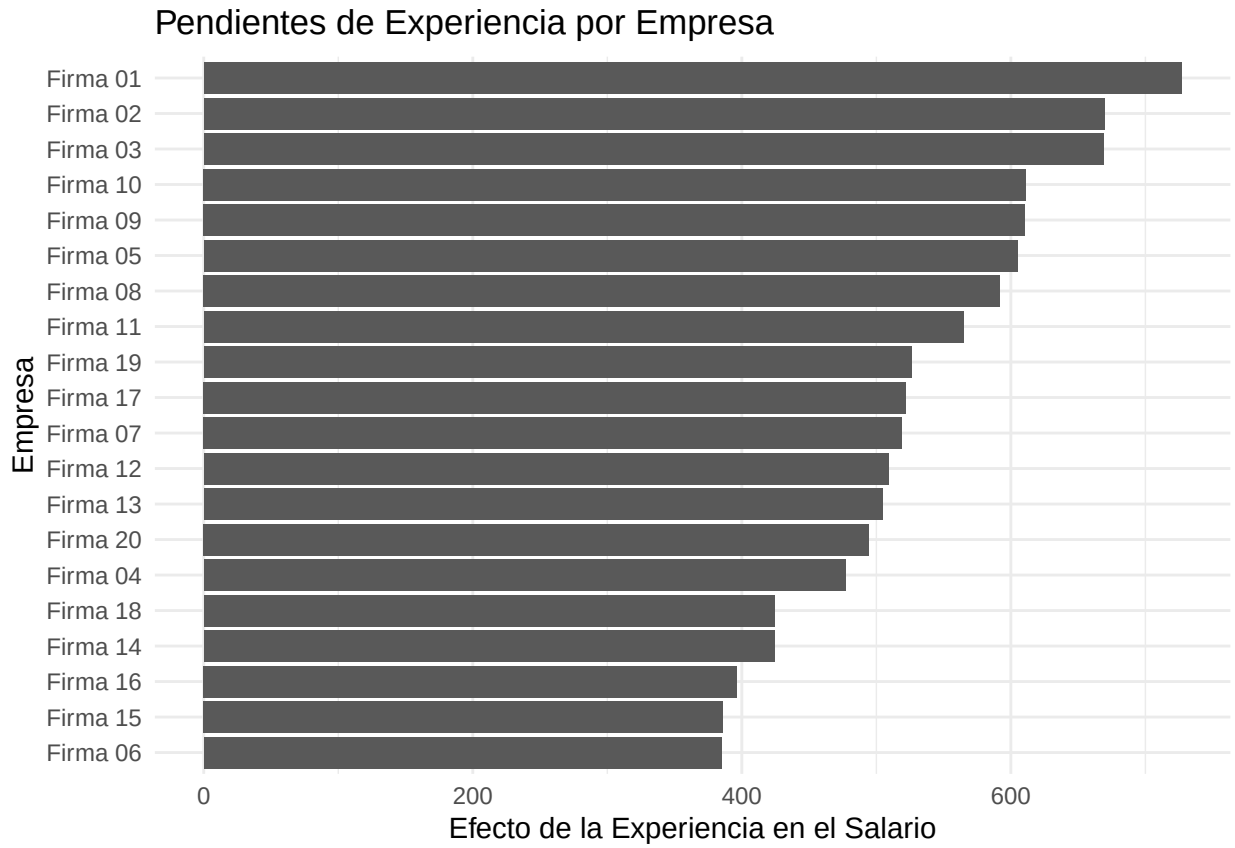
```
## Data: df
## Models:
## modelo_intercepto_aleatorio: salary ~ experience + (1 | firma)
## modelo_coeficientes_aleatorios: salary ~ experience + (experience | firma)
##                                npar   AIC   BIC  logLik -2*log(L)  Chisq Df
## modelo_intercepto_aleatorio       4 10156 10173 -5073.9     10148
## modelo_coeficientes_aleatorios    6 10150 10177 -5069.2     10138 9.3127  2
##                                Pr(>Chisq)
## modelo_intercepto_aleatorio
## modelo_coeficientes_aleatorios   0.009501 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El p-valor pequeño indica que el modelo con pendientes aleatorias es significativamente mejor que el modelo con solo intercepto aleatorio. Existe variación significativa en el efecto de la experiencia en el salario entre empresas.

Para ayudar a comprender mejor los resultados, es útil visualizar cómo varía el efecto de la experiencia en el salario entre empresas.

```
#library(ggplot2)
# Extraer los coeficientes aleatorios
coeficientes_aleatorios <- coef(modelo_coeficientes_aleatorios)$firma

# Renombrar las columnas
colnames(coeficientes_aleatorios) <- c("Intercepto", "Pendiente")

# Añadir el identificador de la empresa
coeficientes_aleatorios$firma <- rownames(coeficientes_aleatorios)

# Graficar las pendientes de experiencia por empresa
ggplot(coeficientes_aleatorios, aes(x = reorder(firma, Pendiente), y = Pendiente)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Pendientes de Experiencia por Empresa",
       x = "Empresa",
```

```
       y = "Efecto de la Experiencia en el Salario") +
  theme_minimal()
```




```
# Crear un conjunto de datos con las predicciones del modelo
df$predicciones <- predict(modelo_coeficientes_aleatorios)

# Graficar los datos y las líneas de regresión por empresa
ggplot(df, aes(x = experience, y = salary, color = firma, group = firma)) +
  geom_point(alpha = 0.5) +
  geom_line(aes(y = predicciones)) +
  labs(title = "Relación entre experiencia y salario por empresa",
       x = "Experiencia",
       y = "Salario") +
  theme_minimal()
```

Modelos longitudinales

Los modelos longitudinales son una herramienta estadística esencial para analizar datos en los que se realizan observaciones repetidas de los mismos individuos a lo largo del tiempo. Este tipo de análisis es fundamental en campos como la medicina, psicología, economía, política, sociología y otras disciplinas donde es importante entender cómo cambian las mediciones en los individuos o unidades a lo largo del tiempo y qué factores influyen en esos cambios.

Los datos longitudinales son un caso especial de datos jerárquicos o multinivel, donde las observaciones (mediciones) están anidadas dentro de individuos. En este contexto, las observaciones repetidas de un mismo individuo tienden a ser más similares entre sí que las observaciones de diferentes individuos. Esta correlación dentro de los sujetos es crucial y debe ser tenida en cuenta para obtener inferencias estadísticas válidas. Una vista general sobre paquetes útiles en R para analizar este tipo de datos correlacionados se puede encontrar en la CRAN Task View dedicada.

En este taller, exploraremos cómo ajustar y analizar modelos longitudinales utilizando R, centrándonos en un conjunto de datos que mide el crecimiento dental en niños y niñas a diferentes edades. Trabajaremos con un conjunto de datos que contiene medidas longitudinales del crecimiento dental en 27 niños (16 niños y 11 niñas). Las medidas corresponden a la distancia pituitaria-pterigomaxilar (una medida de crecimiento dental) y se tomaron a las edades de 8, 10, 12 y 14 años. Nuestro objetivo es describir cómo cambia esta distancia con la edad y comparar el patrón de crecimiento entre niños y niñas.

Primero, cargamos los datos y observamos su estructura:

```
load(url("http://alecri.github.io/downloads/data/dental.RData"))
head(dental)
```

```
## # A tibble: 6 x 6
##      id sex      y8   y10   y12   y14
##   <dbl> <fct> <dbl> <dbl> <dbl> <dbl>
## 1     1 Girl   21    20    21.5  23
## 2     2 Girl   21    21.5  24    25.5
## 3     3 Girl   20.5  24    24.5  26
## 4     4 Girl   23.5  24.5  25    26.5
## 5     5 Girl   21.5  23    22.5  23.5
## 6     6 Girl   20    21    21    22.5
```

Los datos se presentan en formato ancho, donde las mediciones repetidas se encuentran en columnas separadas para cada edad. Este formato no es ideal para el análisis longitudinal, por lo que convertiremos los datos a un formato largo usando la función pivot_longer():

```
library(labelled)   # etiquetado de datos
library(rstatix)    # estadísticas descriptivas
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ggpubr)     # estadísticas descriptivas y gráficos convenientes
library(GGally)     # gráficos avanzados
library(car)        # útil para ANOVA/pruebas de Wald
library(Epi)        # fácil obtención de intervalos de confianza para coeficientes/predicciones del mod
```

```
##
## Attaching package: 'Epi'
```

```
## The following object is masked from 'package:lme4':
##
##     factorize
```

```
#library(lme4)       # modelos lineales de efectos mixtos
#library(lmerTest)   # pruebas para modelos lineales de efectos mixtos
library(emmeans)    # medias marginales
```

```
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
```

```
##
## Attaching package: 'emmeans'

## The following object is masked from 'package:GGally':
##
##     pigs
```

```
library(multcomp)   # intervalos de confianza para combinaciones lineales de coeficientes del modelo
```

```
## Loading required package: mvtnorm

##
## Attaching package: 'mvtnorm'

## The following object is masked from 'package:arm':
##
##     standardize

## The following object is masked from 'package:jtools':
##
##     standardize

## Loading required package: survival

## Loading required package: TH.data

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(geepack)    # ecuaciones de estimación generalizadas
library(ggeffects)  # efectos marginales, predicciones ajustadas
```

```
##
## Attaching package: 'ggeffects'

## The following object is masked from 'package:wooldridge':
##
##     fish
```

```
library(gt)         # tablas bonitas
dental_long <- pivot_longer(dental, cols = starts_with("y"),
                            names_to = "measurement", values_to = "distance") |>
  mutate(
    age = parse_number(measurement),
    measurement = fct_inorder(paste("Medida a los", age))
  ) |>
```

```
  set_variable_labels(
    age = "Edad del niño/a al momento de la medición",
    measurement = "Etiqueta de medición temporal",
    distance = "Medición de distancia"
  )

head(dental_long)
```

```
## # A tibble: 6 x 5
##      id sex   measurement     distance   age
##   <dbl> <fct> <fct>              <dbl> <dbl>
## 1     1 Girl  Medida a los 8      21       8
## 2     1 Girl  Medida a los 10     20      10
## 3     1 Girl  Medida a los 12     21.5    12
## 4     1 Girl  Medida a los 14     23      14
## 5     2 Girl  Medida a los 8      21       8
## 6     2 Girl  Medida a los 10     21.5    10
```

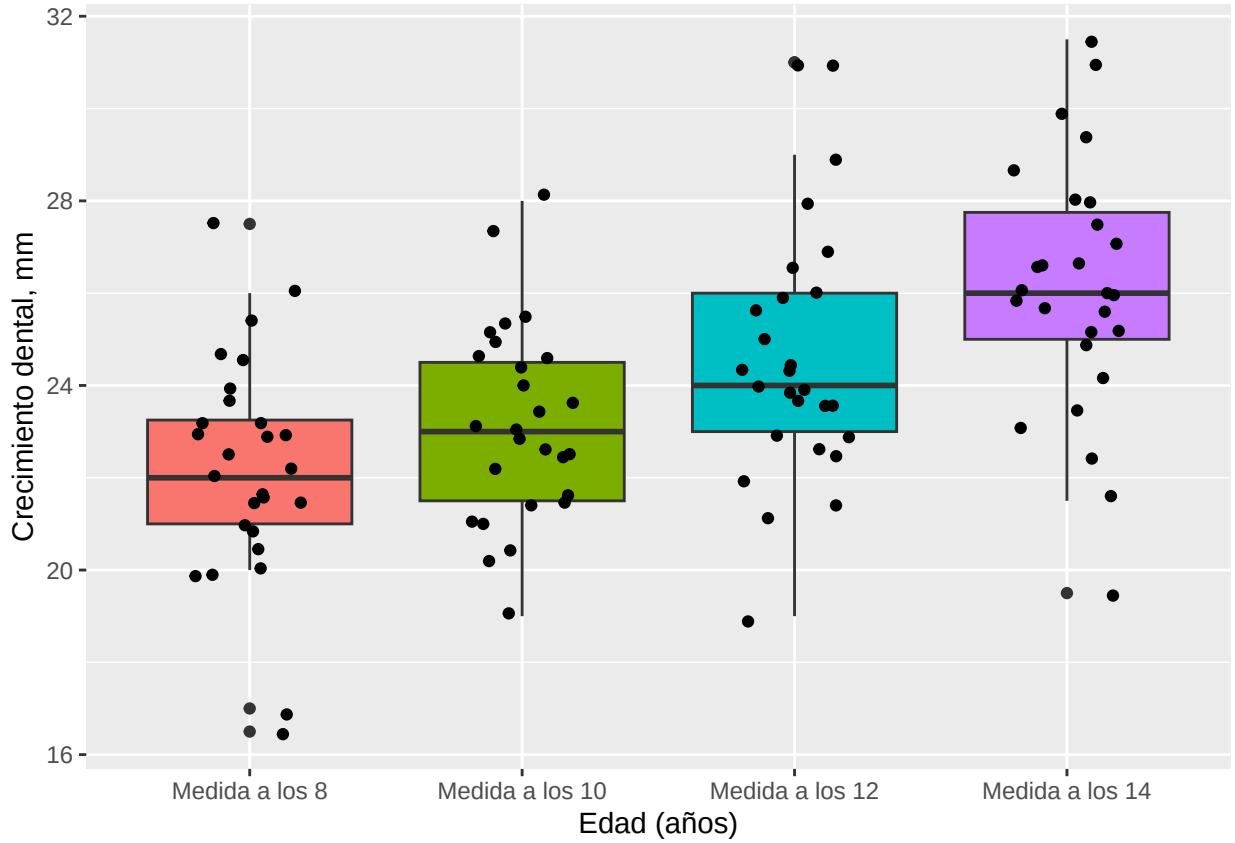
Ahora, cada fila representa una medición de un individuo en una edad específica, lo que facilita el análisis longitudinal. Antes de ajustar modelos, exploramos los datos de forma descriptiva para entender las tendencias generales.

```
group_by(dental_long, age) |>
  get_summary_stats(distance)
```

```
## # A tibble: 4 x 14
##     age variable     n   min   max median    q1    q3   iqr   mad  mean    sd
##   <dbl> <fct>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     8 distance    27  16.5  27.5     22  21    23.2  2.25  1.48  22.2  2.43
## 2    10 distance    27  19    28       23  21.5  24.5  3     2.22  23.2  2.16
## 3    12 distance    27  19    31       24  23    26    3     2.22  24.6  2.82
## 4    14 distance    27  19.5  31.5     26  25    27.8  2.75  2.22  26.1  2.77
## # i 2 more variables: se <dbl>, ci <dbl>
```

Observamos que la distancia media aumenta con la edad, lo que sugiere un crecimiento dental a medida que los niños y niñas crecen. Creamos un diagrama de caja para visualizar la distribución de la distancia a cada edad:

```
ggplot(dental_long, aes(measurement, distance, fill = measurement)) +
  geom_boxplot() +
  geom_jitter(width = 0.2) +
  guides(fill = "none") +
  labs(x = "Edad (años)", y = "Crecimiento dental, mm")
```



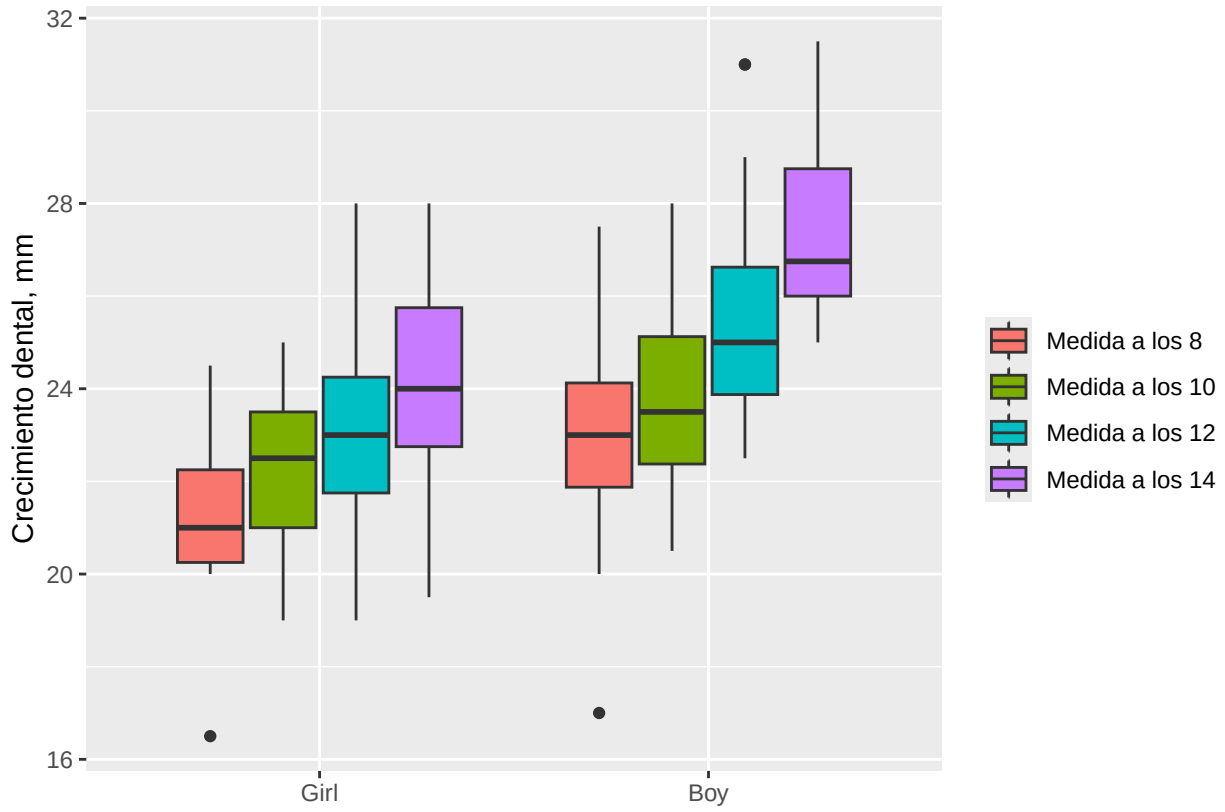
El gráfico muestra que la mediana y los valores de la distancia aumentan con la edad. La dispersión también parece aumentar ligeramente, lo que indica mayor variabilidad en edades superiores. Exploremos si hay diferencias en el crecimiento dental entre niños y niñas.

```
group_by(dental_long, sex, measurement) |>
  get_summary_stats(distance, show = c("mean", "sd"))
```

```
## # A tibble: 8 x 6
##   sex   measurement     variable     n  mean    sd
##   <fct> <fct>           <fct>    <dbl> <dbl> <dbl>
## 1 Girl  Medida a los 8  distance    11  21.2  2.12
## 2 Girl  Medida a los 10 distance    11  22.2  1.90
## 3 Girl  Medida a los 12 distance    11  23.1  2.37
## 4 Girl  Medida a los 14 distance    11  24.1  2.44
## 5 Boy   Medida a los 8  distance    16  22.9  2.45
## 6 Boy   Medida a los 10 distance    16  23.8  2.14
## 7 Boy   Medida a los 12 distance    16  25.7  2.65
## 8 Boy   Medida a los 14 distance    16  27.5  2.08
```

Y gráficamente

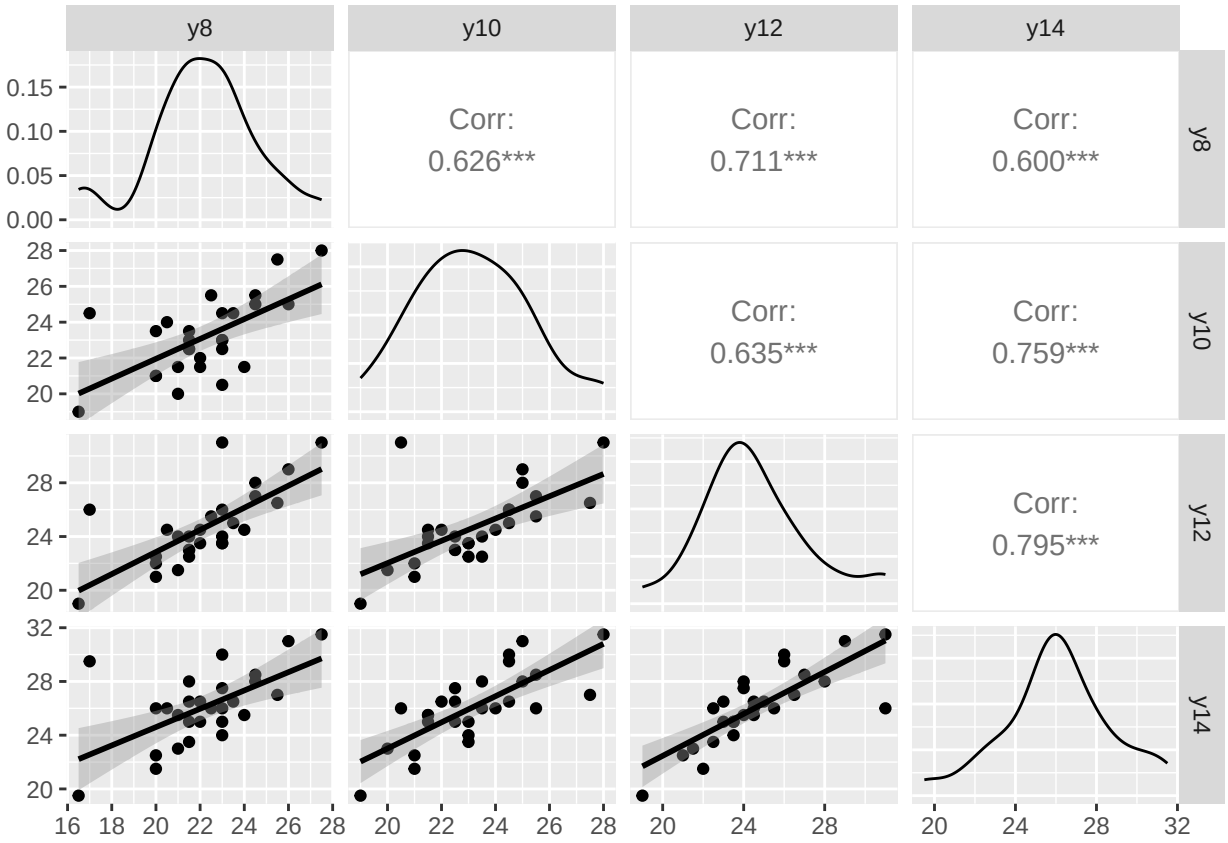
```
ggplot(dental_long, aes(sex, distance, fill = measurement)) +
  geom_boxplot() +
  labs(x = "", y = "Crecimiento dental, mm", fill = "")
```



Visualmente, parece que los niños tienen, en promedio, una distancia ligeramente mayor que las niñas en cada edad.

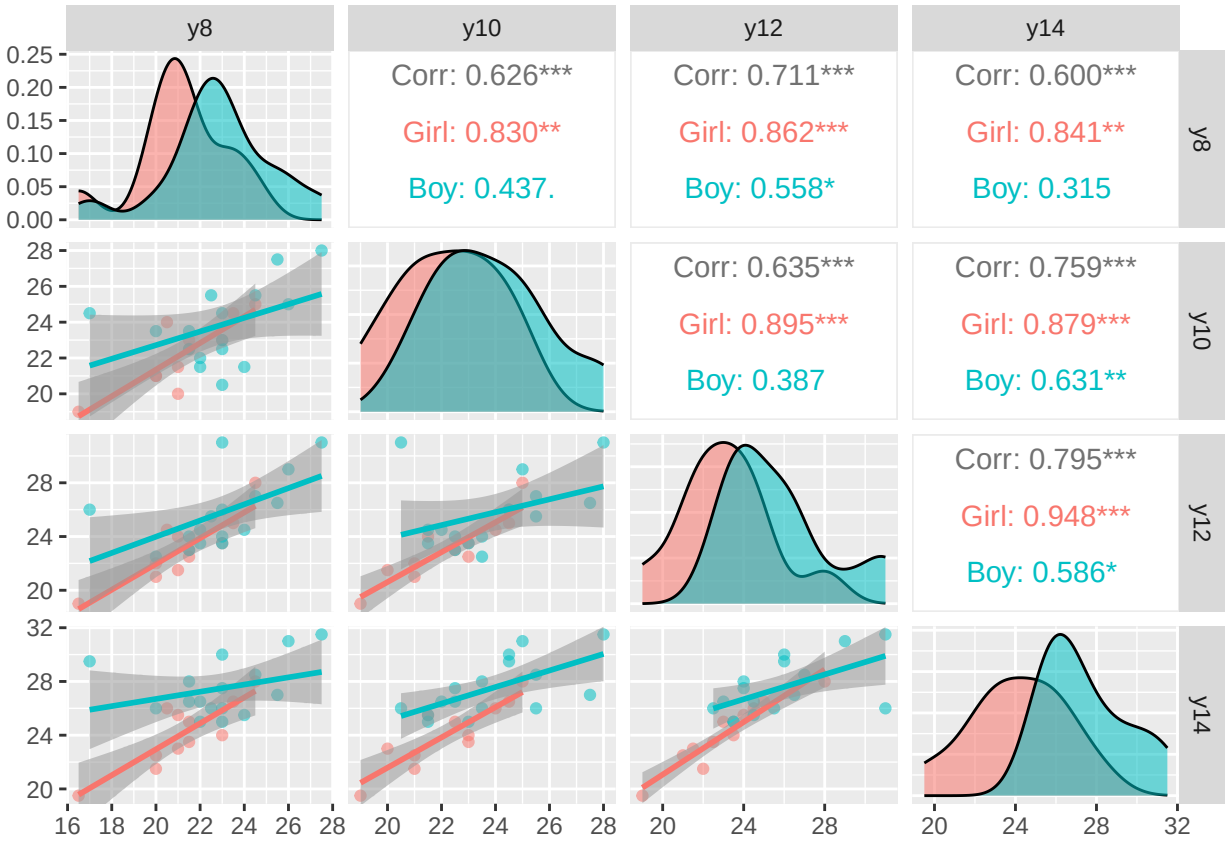
Evaluemos las correlaciones entre medidas de interés:

```
ggpairs(select(dental, starts_with("y")), lower = list(continuous = "smooth"))
```



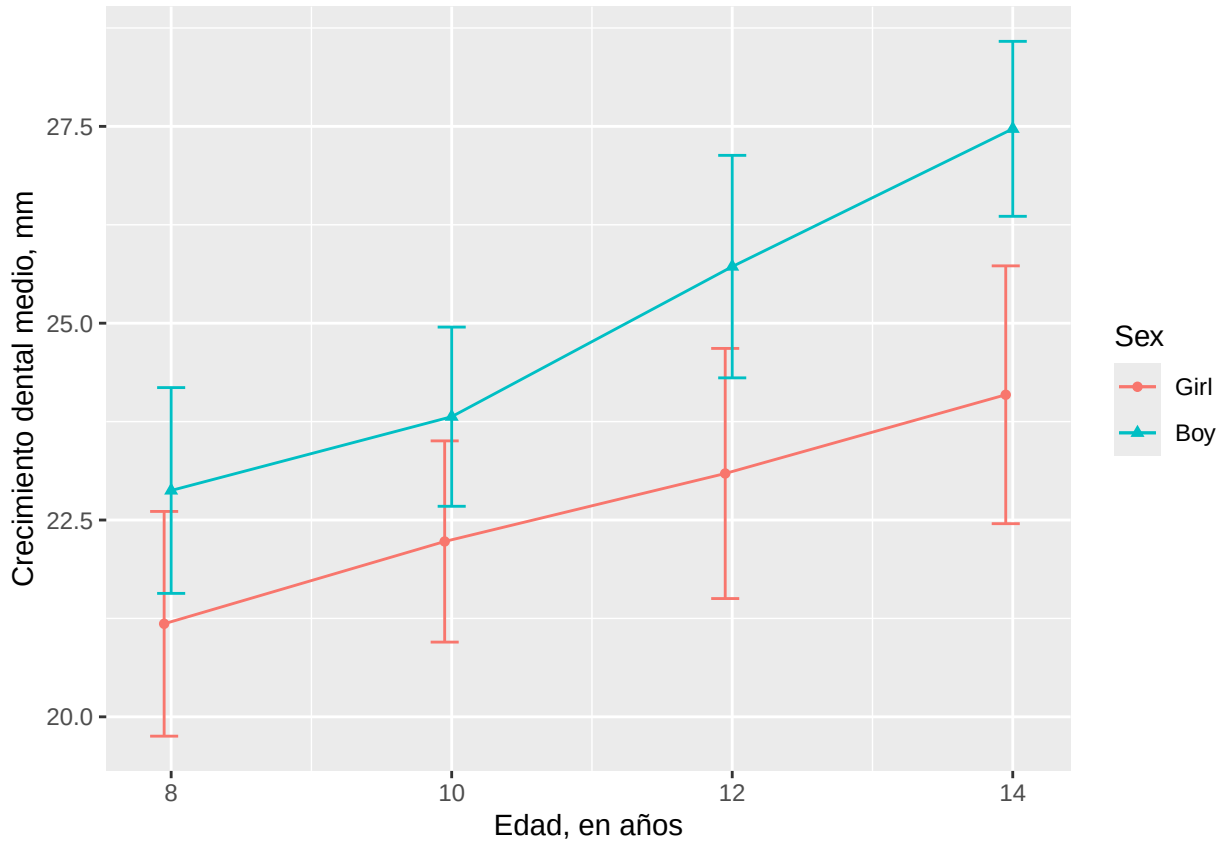
y separemos nuevamente por grupo (sexo)

```
ggpairs(dental, mapping = aes(colour = sex,alpha = 0.5), columns = 3:6,
        lower = list(continuous = "smooth"))
```

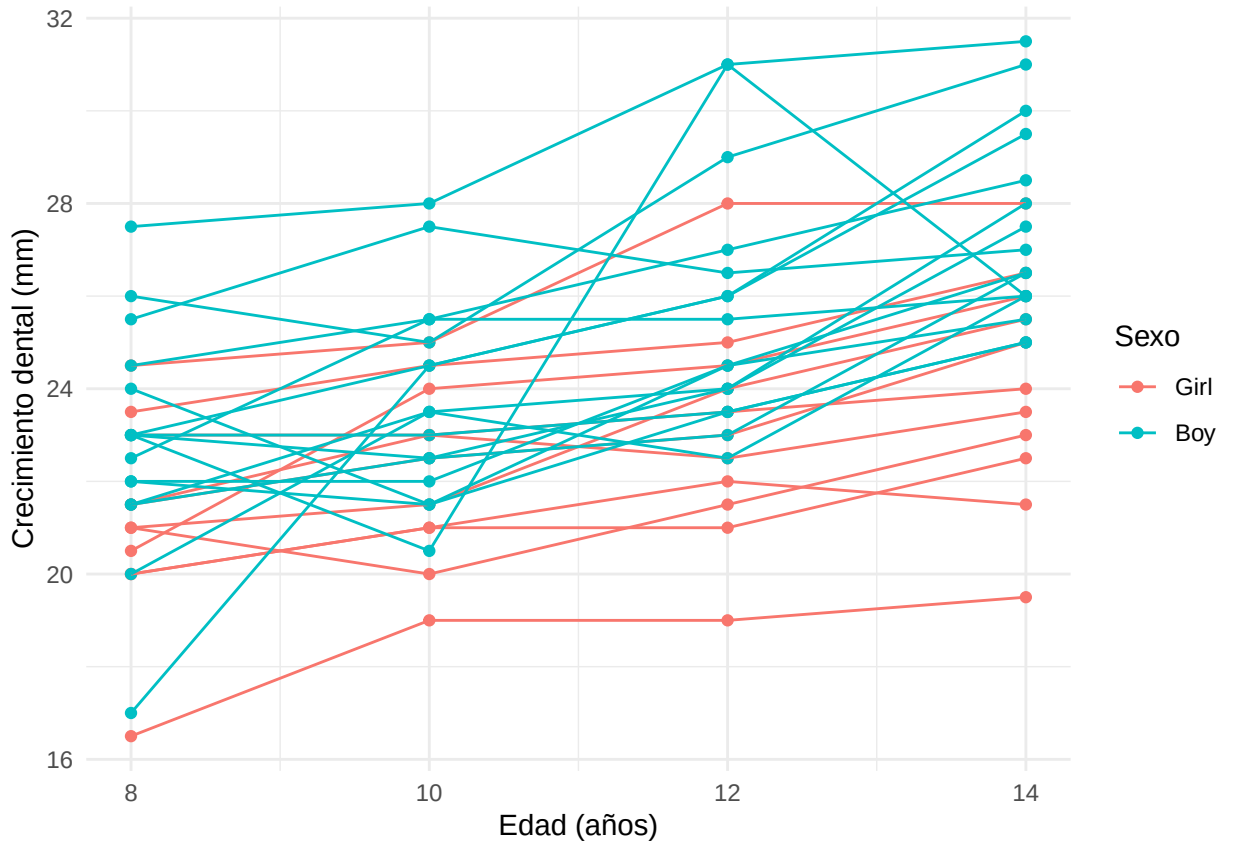


Digamos que queremos entender si el efecto varía con el tiempo. Podríamos evaluar gráficamente esto:

```
group_by(dental_long, sex, age) |>
  summarise(mean = list(mean_ci(distance)), .groups = "drop") |>
  unnest_wider(mean) |>
  mutate(agex = age - .05 + .05*(sex == "Boy")) |>
  ggplot(aes(agex, y, col = sex, shape = sex)) +
  geom_point() +
  geom_errorbar(aes(ymin = ymin, ymax = ymax), width = 0.2) +
  geom_line() +
  labs(x = "Edad, en años", y = "Crecimiento dental medio, mm", shape = "Sex", col = "Sex")
```




```r
ggplot(dental_long, aes(x = age, y = distance, color = sex)) +
  geom_point() +
  geom_line(aes(group = id)) +
  labs(x = "Edad (años)", y = "Crecimiento dental (mm)", color = "Sexo") +
  theme_minimal()
```



Para analizar los datos longitudinales, utilizamos modelos mixtos lineales, que permiten modelar tanto los efectos fijos (comunes a todos los individuos) como los efectos aleatorios (específicos de cada individuo).

Modelo inicial

Ajustamos un modelo que considera la edad como factor categórico (variables indicadoras) y un intercepto aleatorio para capturar la variabilidad entre individuos.

```
# Ajustar el modelo mixto
modelo_edad <- lmer(distance ~ factor(age) + (1 | id), data = dental_long)

# Resumen del modelo
summary(modelo_edad)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: distance ~ factor(age) + (1 | id)
##    Data: dental_long
##
## REML criterion at convergence: 443.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7376 -0.5248  0.0153  0.4027  3.7212
##
```

```
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 4.465    2.113
##  Residual             2.078    1.442
## Number of obs: 108, groups:  id, 27
##
## Fixed effects:
##               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    22.1852     0.4923 43.3911  45.066  < 2e-16 ***
## factor(age)10   0.9815     0.3924 78.0000   2.501   0.0145 *
## factor(age)12   2.4630     0.3924 78.0000   6.277 1.80e-08 ***
## factor(age)14   3.9074     0.3924 78.0000   9.958 1.52e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fc()10 fc()12
## factor(g)10 -0.399
## factor(g)12 -0.399  0.500
## factor(g)14 -0.399  0.500  0.500
```

En este caso el intercepto captura la distancia media dental a los 8 años (valor de referencia). Esta es 22.18 mm. La diferencia entre las respuestas medias de los niños de 10 y 8 años es de 0.98 mm. La diferencia entre las respuestas medias de los niños de 12 y 8 años es de 2.46 mm. La diferencia entre las respuestas medias de los niños de 14 y 8 años es de 3.91 mm.

Esto indica que la distancia pituitaria-pterigomaxilar aumenta significativamente con la edad, lo que es consistente con el crecimiento observado durante este período.

Usamos un análisis de varianza para evaluar si la edad tiene un efecto significativo en el crecimiento dental.

```
library(car)

Anova(modelo_edad)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: distance
##              Chisq Df Pr(>Chisq)
## factor(age) 114.12  3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El efecto de la edad es altamente significativo ($p < 0.001$), indicando que la edad influye en el crecimiento dental.

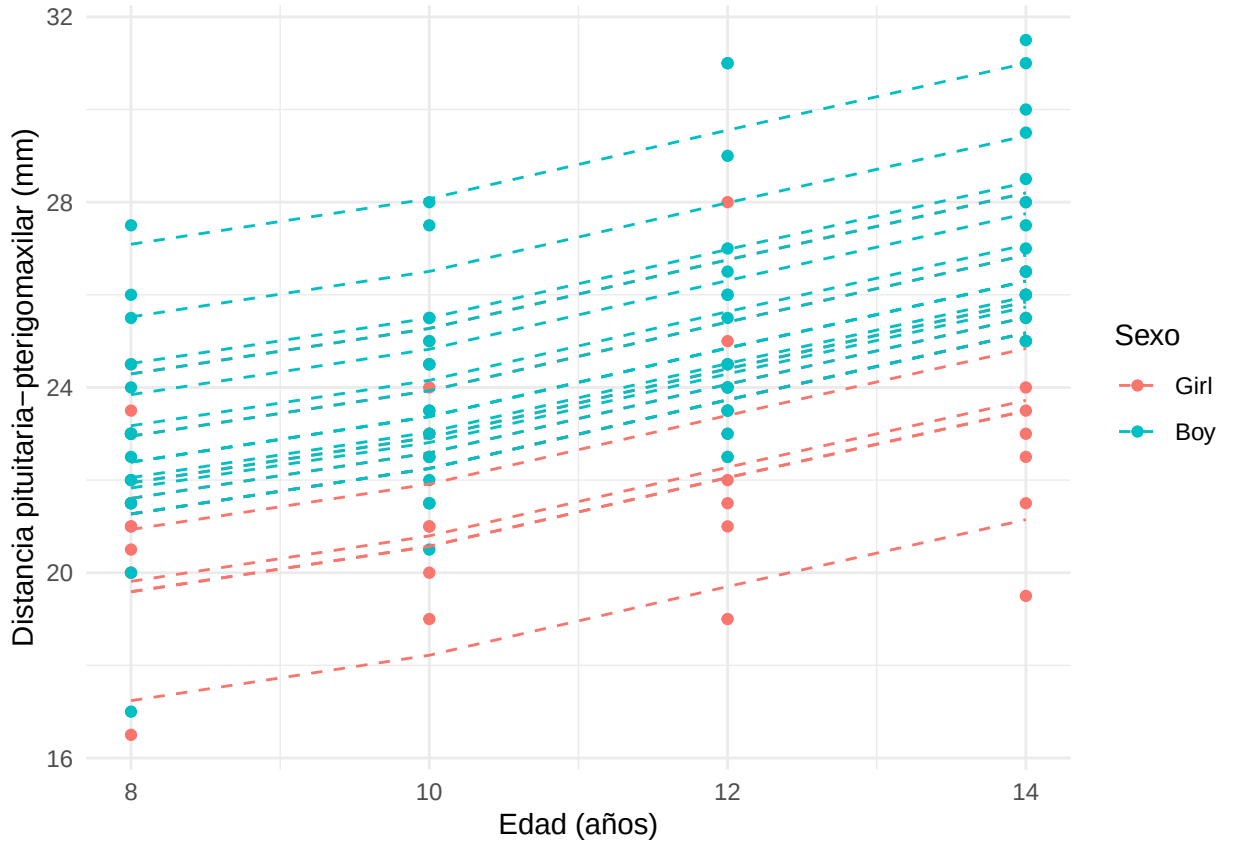
Calculamos las medias marginales estimadas para cada edad y sus intervalos de confianza.

```
library(emmeans)

# Calcular las medias marginales
medias_edad <- emmeans(modelo_edad, ~ age)

# Mostrar los resultados
summary(medias_edad)
```

```
##  age emmean    SE   df lower.CL upper.CL
##    8   22.2 0.492 43.4     21.2     23.2
##   10   23.2 0.492 43.4     22.2     24.2
##   12   24.6 0.492 43.4     23.7     25.6
##   14   26.1 0.492 43.4     25.1     27.1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

Las medias estimadas confirman que la distancia media aumenta con la edad, aumentando la tasa de crecimiento en cada momento de medición. Podemos presentar la trayectoria estimada del modelo ajustado utilizando la función predict:

```
# Generar predicciones del modelo ajustado
predicciones <- predict(modelo_edad)

# Visualizar la trayectoria estimada
ggplot(dental_long, aes(x = age, y = distance, color = sex)) +
  geom_point() +
  geom_line(aes(y = predicciones, group = id), linetype = "dashed") +
  labs(x = "Edad (años)", y = "Distancia pituitaria-pterigomaxilar (mm)", color = "Sexo") +
  theme_minimal()
```



Este gráfico muestra tanto los datos reales de distancia en función de la edad como las predicciones del modelo para cada niño, con líneas discontinuas que representan las trayectorias estimadas a lo largo del tiempo.

Modelo con variable de sexo

```r
# Ajustar el modelo con sexo y su interacción con la edad
modelo_sexo <- lmer(distance ~ factor(age) * sex + (1 | id), data = dental_long)

# Resumen del modelo
summary(modelo_sexo)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: distance ~ factor(age) * sex + (1 | id)
##    Data: dental_long
##
## REML criterion at convergence: 423.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7358 -0.4761  0.0469  0.4691  3.6613
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept) 3.285    1.813
##  Residual             1.975    1.405
## Number of obs: 108, groups:  id, 27
##
## Fixed effects:
##                      Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)           21.1818     0.6915 46.0791  30.630  < 2e-16 ***
## factor(age)10          1.0455     0.5992 75.0000   1.745   0.0851 .
## factor(age)12          1.9091     0.5992 75.0000   3.186   0.0021 **
## factor(age)14          2.9091     0.5992 75.0000   4.855 6.41e-06 ***
## sexBoy                 1.6932     0.8983 46.0791   1.885   0.0658 .
## factor(age)10:sexBoy  -0.1080     0.7784 75.0000  -0.139   0.8901
## factor(age)12:sexBoy   0.9347     0.7784 75.0000   1.201   0.2337
## factor(age)14:sexBoy   1.6847     0.7784 75.0000   2.164   0.0336 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) fc()10 fc()12 fc()14 sexBoy f()10: f()12:
## factor(g)10 -0.433
## factor(g)12 -0.433  0.500
## factor(g)14 -0.433  0.500  0.500
## sexBoy      -0.770  0.334  0.334  0.334
## fctr(g)10:B  0.334 -0.770 -0.385 -0.385 -0.433
## fctr(g)12:B  0.334 -0.385 -0.770 -0.385 -0.433  0.500
## fctr(g)14:B  0.334 -0.385 -0.385 -0.770 -0.433  0.500  0.500
```

Interpretando salidas:

- La varianza del Intercepto Aleatorio (id) es 3.285 con una desviación estándar de 1.813. Existe variabilidad significativa entre los individuos en sus medidas iniciales de distancia dental.

- La varianza residual es 1.975 con una desviación estándar de 1.405. Esta captura variabilidad no explicadas por el modelo.
- El intercepto captura el ser una niña de 8 años como el punto de referencia: 21.1818mm.
- Coeficientes de factor(g)AA: El efecto de edad está dándonos el cambio en la distancia dental en comparación con la edad de referencia (8 años) para las niñas. Por ejemplo, las niñas de 10 años tienen, en promedio, una distancia dental 1.05 mm mayor que las niñas de 8 años, mientras que las de 14 años tienen, en promedio, una distancia dental 2.91 mm mayor que las niñas de 8 años.
- Coeficiente `sexBoy`: Los niños de 8 años tienen, en promedio, una distancia dental 1.69 mm mayor que las niñas de 8 años. Esta es una variable dummy.
- Los coeficientes de interacción indican cuánto cambia la diferencia entre niños y niñas en cada edad en comparación con la edad de referencia (8 años). Por ejemplo, a los 14 años, los niños tienen una distancia dental adicional de 1.68 mm en comparación con las niñas, más allá de la diferencia observada a los 8 años. Esa edad es la única que reporta distancias significativas a la referencia; al entrar más en la adolescencia las diferencias se tornan notables.

```
Anova(modelo_sexo)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: distance
##                    Chisq Df Pr(>Chisq)
## factor(age)     120.0950  3  < 2.2e-16 ***
## sex               9.2921  1   0.002301 **
## factor(age):sex   7.0847  3   0.069247 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El efecto de la edad es altamente significativo. Esto indica que existen diferencias significativas en la distancia dental media entre las diferentes edades (8, 10, 12 y 14 años), independientemente del sexo. También el sexo del paciente parece ser significativo. La interacción entre edad y sexo es marginalmente significativa ($p < 0.10$, pero > 0.05).

Series de tiempo: ejemplo de finanzas

Las series de tiempo son un conjunto de observaciones registradas en momentos sucesivos en el tiempo, ordenadas cronológicamente. El análisis de series de tiempo es fundamental en diversos campos como la economía, finanzas, meteorología, y ciencias sociales, ya que permite comprender y predecir comportamientos futuros basados en datos históricos.

En esta ñapita del taller, me enfoco en el análisis de series de tiempo financieras, específicamente en el estudio de los precios de las acciones. El objetivo es proporcionar una guía clara y práctica sobre modelos de series de tiempo utilizando R.

Las series de tiempo financieras, como los precios de las acciones o los tipos de cambio, son intrínsecamente volátiles y están influenciadas por múltiples factores económicos y políticos. El análisis de estas series permite a los inversores y analistas comprender las tendencias del mercado, evaluar riesgos, anticiparse a estas y tomar decisiones informadas.

Para este análisis, utilizaremos datos históricos del precio de cierre ajustado de las acciones de Apple Inc. (AAPL). Usaremos el paquete quantmod para descargar los datos directamente desde Yahoo Finance.

```r
# Instalar paquetes si no están instalados
#install.packages("quantmod")
#install.packages("forecast")
#install.packages("tseries")
#install.packages("ggplot2")
#install.packages("quantmod", repos = "https://cloud.r-project.org/")

# Cargar los paquetes
library(quantmod)
```

```
## Loading required package: xts

##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

## Loading required package: TTR

##
## Attaching package: 'TTR'

## The following object is masked from 'package:Epi':
##
##     ROC

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(forecast)
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:ggpubr':
##
##     gghistogram
```

```
library(tseries)
library(ggplot2)
```

Descargamos los datos de los últimos cinco años.

```
# Establecer el rango de fechas
fecha_inicio <- as.Date("2019-01-01")
fecha_fin <- as.Date(Sys.Date())

# Descargar los datos de AAPL
getSymbols("AAPL", src = "yahoo", from = fecha_inicio, to = fecha_fin)
```

```
## [1] "AAPL"
```

```
# Ver las primeras filas
head(AAPL)
```

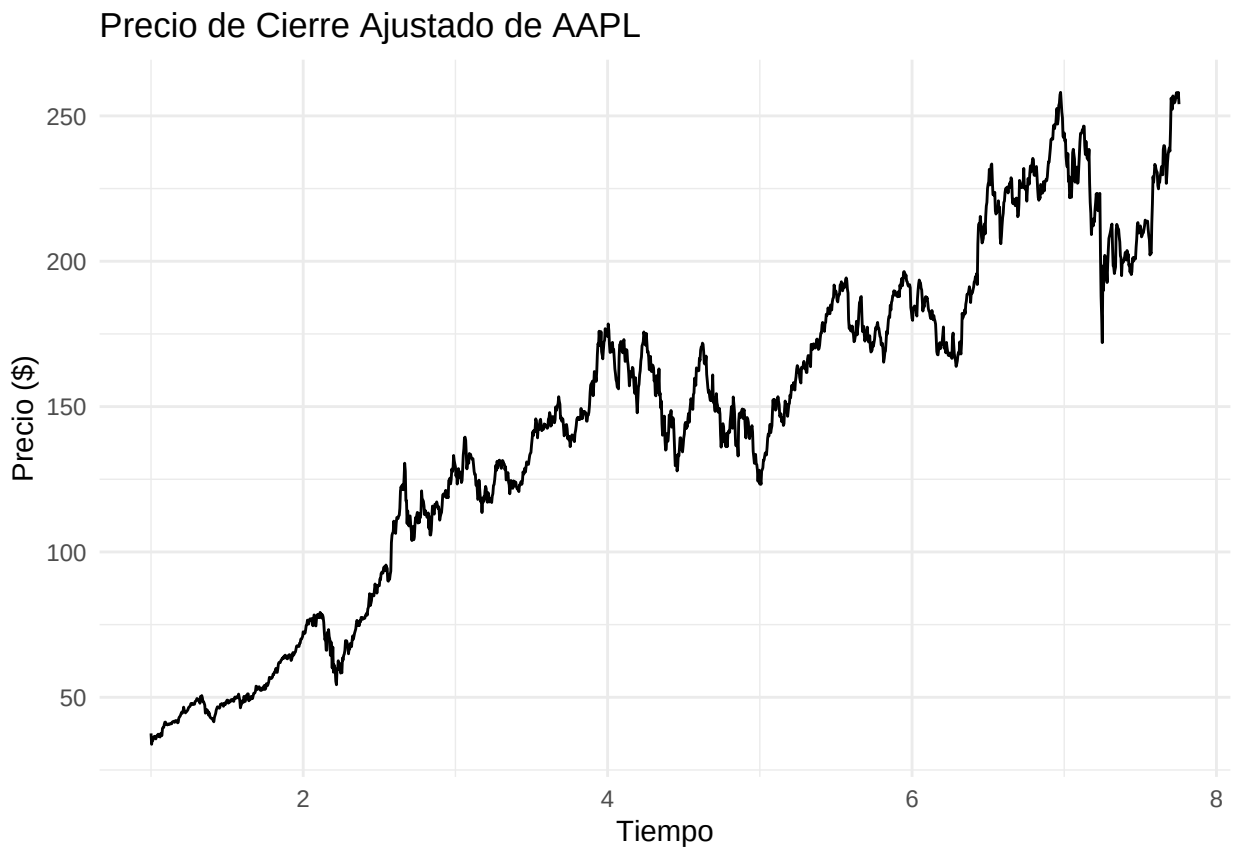
```
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2019-01-02   38.7225   39.7125  38.5575    39.4800   148158800      37.57521
## 2019-01-03   35.9950   36.4300  35.5000    35.5475   365248800      33.83244
## 2019-01-04   36.1325   37.1375  35.9500    37.0650   234428400      35.27673
## 2019-01-07   37.1750   37.2075  36.4750    36.9825   219111200      35.19819
## 2019-01-08   37.3900   37.9550  37.1300    37.6875   164101200      35.86919
## 2019-01-09   37.8225   38.6325  37.4075    38.3275   180396400      36.47831
```

Extraemos la columna de precios de cierre ajustados y creamos un objeto de serie de tiempo.

```
# Extraer el precio de cierre ajustado
precio_cierre <- AAPL[, "AAPL.Adjusted"]

# Convertir a serie de tiempo
serie_tiempo <- ts(precio_cierre, frequency = 252)  # 252 días hábiles en un año
```

Antes de ajustar cualquier modelo, es importante entender las características de la serie de tiempo.

```
# Graficar la serie de tiempo
autoplot(serie_tiempo) +
  labs(title = "Precio de Cierre Ajustado de AAPL",
       x = "Tiempo",
       y = "Precio ($)") +
  theme_minimal()
```



El gráfico muestra la evolución del precio de AAPL desde 2019 hasta la fecha actual. Observamos tendencias ascendentes y periodos de volatilidad, especialmente durante eventos económicos significativos como la pandemia de COVID-19 y la subsiguiente alza en valuaciones financieras.

Descomponemos la serie para analizar sus componentes: tendencia, estacionalidad y residuales.

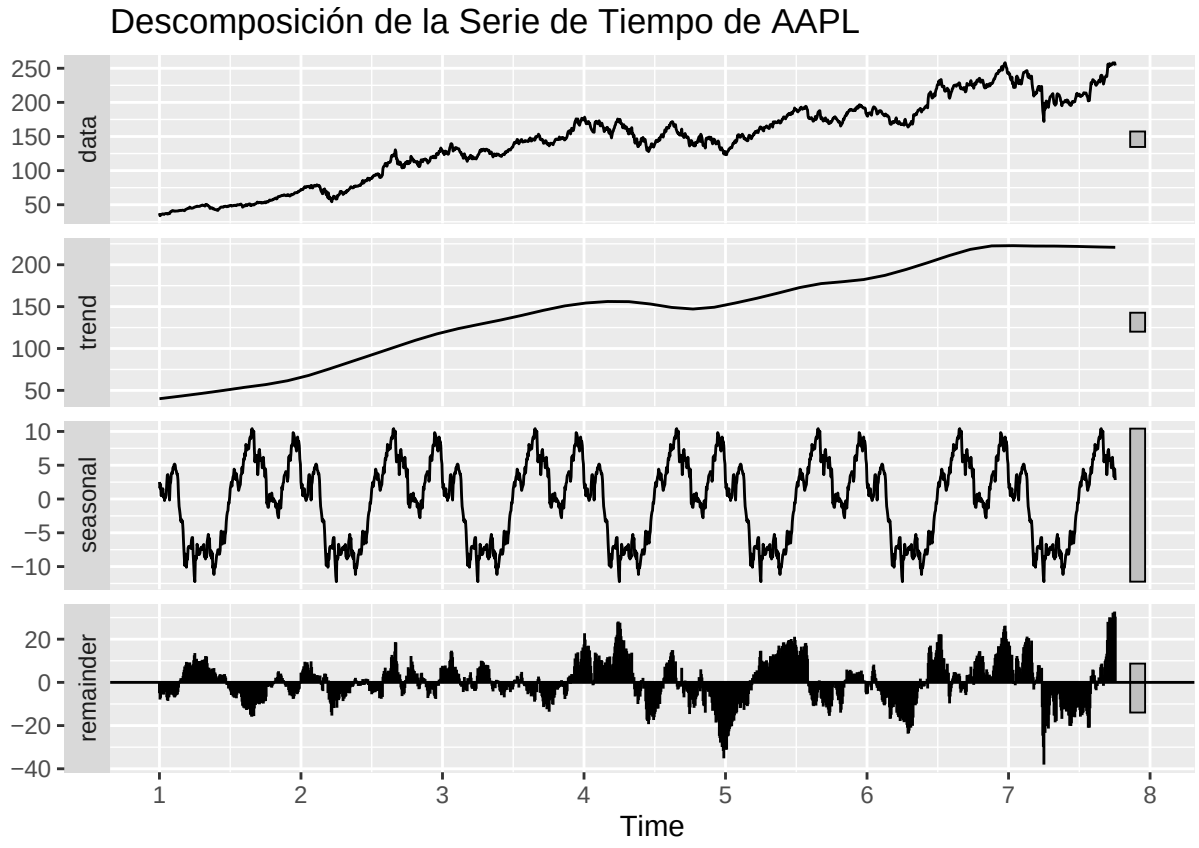
```
# Verificar la estructura de 'serie_tiempo'
str(serie_tiempo)
```

```
##  Time-Series [1:1703, 1] from 1 to 7.75: 37.6 33.8 35.3 35.2 35.9 ...
##  - attr(*, "index")= num [1:1703] 1.55e+09 1.55e+09 1.55e+09 1.55e+09 1.55e+09 ...
##   ..- attr(*, "tzone")= chr "UTC"
##   ..- attr(*, "tclass")= chr "Date"
##  - attr(*, "src")= chr "yahoo"
##  - attr(*, "updated")= POSIXct[1:1], format: "2025-10-10 12:48:04"
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "AAPL.Adjusted"
```

```
# Convertir a vector numérico
precio_cierre_vector <- as.numeric(precio_cierre)

# Crear la serie de tiempo
serie_tiempo <- ts(precio_cierre_vector, frequency = 252)

# Descomposición utilizando STL (Seasonal and Trend decomposition using Loess)
```

```r
descomposición <- stl(serie_tiempo, s.window = "periodic")

# Graficar la descomposición
autoplot(descomposición) +
  labs(title = "Descomposición de la Serie de Tiempo de AAPL")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the forecast package.
##   Please report the issue at <https://github.com/robjhyndman/forecast/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



- Tendencia: Muestra el movimiento a largo plazo del precio.
- Estacionalidad: En series financieras diarias, la estacionalidad puede no ser pronunciada, pero pueden existir patrones semanales o mensuales.
- Residuales: Parte de la serie no explicada por la tendencia ni la estacionalidad.

La estacionariedad es una propiedad clave en el análisis de series de tiempo. Una serie estacionaria tiene estadísticas (media, varianza) constantes en el tiempo. Realizamos la prueba Dickey-Fuller Aumentada (ADF) para verificar si la serie es estacionaria.

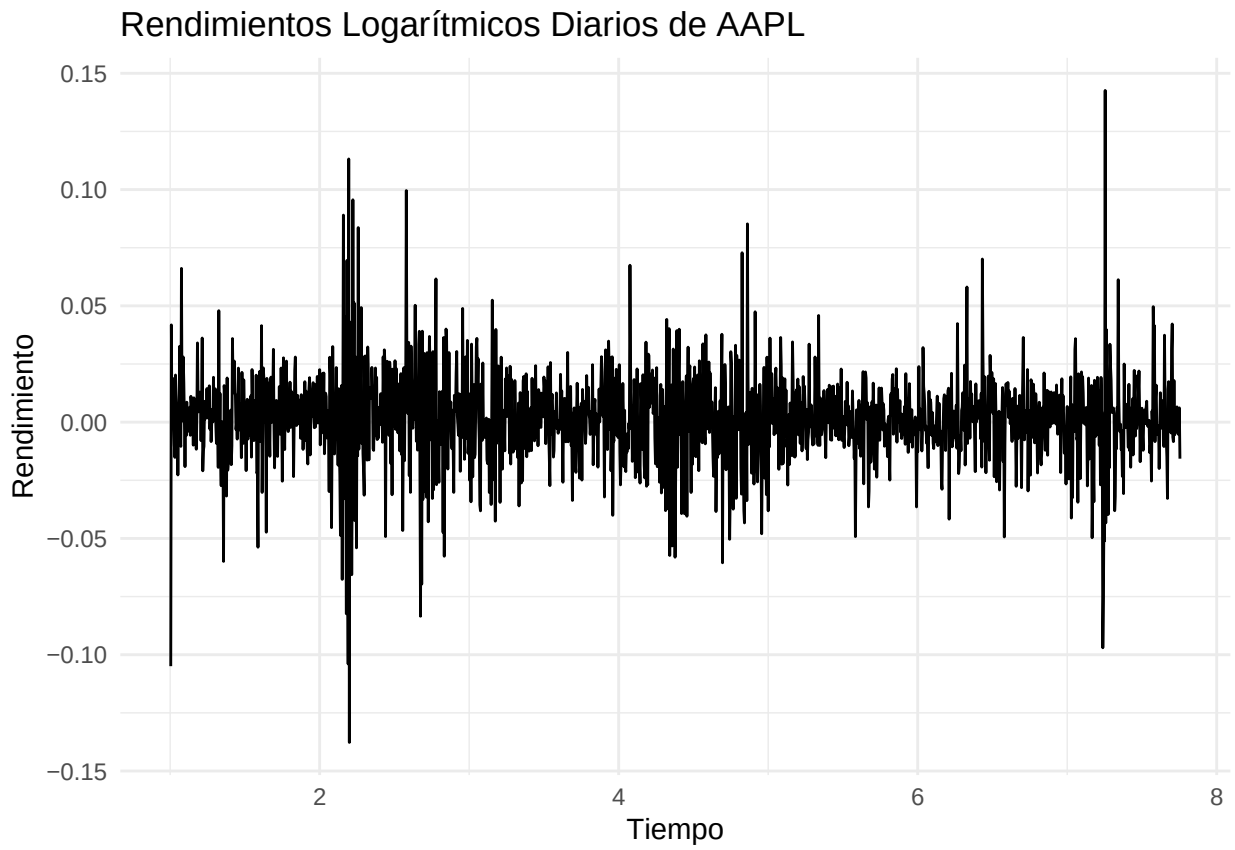
```r
# Prueba ADF
adf.test(serie_tiempo, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  serie_tiempo
## Dickey-Fuller = -2.9984, Lag order = 11, p-value = 0.1557
## alternative hypothesis: stationary
```

El p-valor es alto (mayor que 0.05), lo que indica que no podemos rechazar la hipótesis nula de que la serie tiene una raíz unitaria (no estacionaria). Concluimos que la serie no es estacionaria.

Para lograr estacionariedad, transformamos los precios en rendimientos logarítmicos.

```r
# Calcular los rendimientos logarítmicos
rendimientos <- diff(log(serie_tiempo))

# Graficar los rendimientos
autoplot(rendimientos) +
  labs(title = "Rendimientos Logarítmicos Diarios de AAPL",
       x = "Tiempo",
       y = "Rendimiento") +
  theme_minimal()
```



Verificamos nuevamente la serie transformada con ADF.

```r
# Prueba ADF en los rendimientos
adf.test(rendimientos, alternative = "stationary")
```

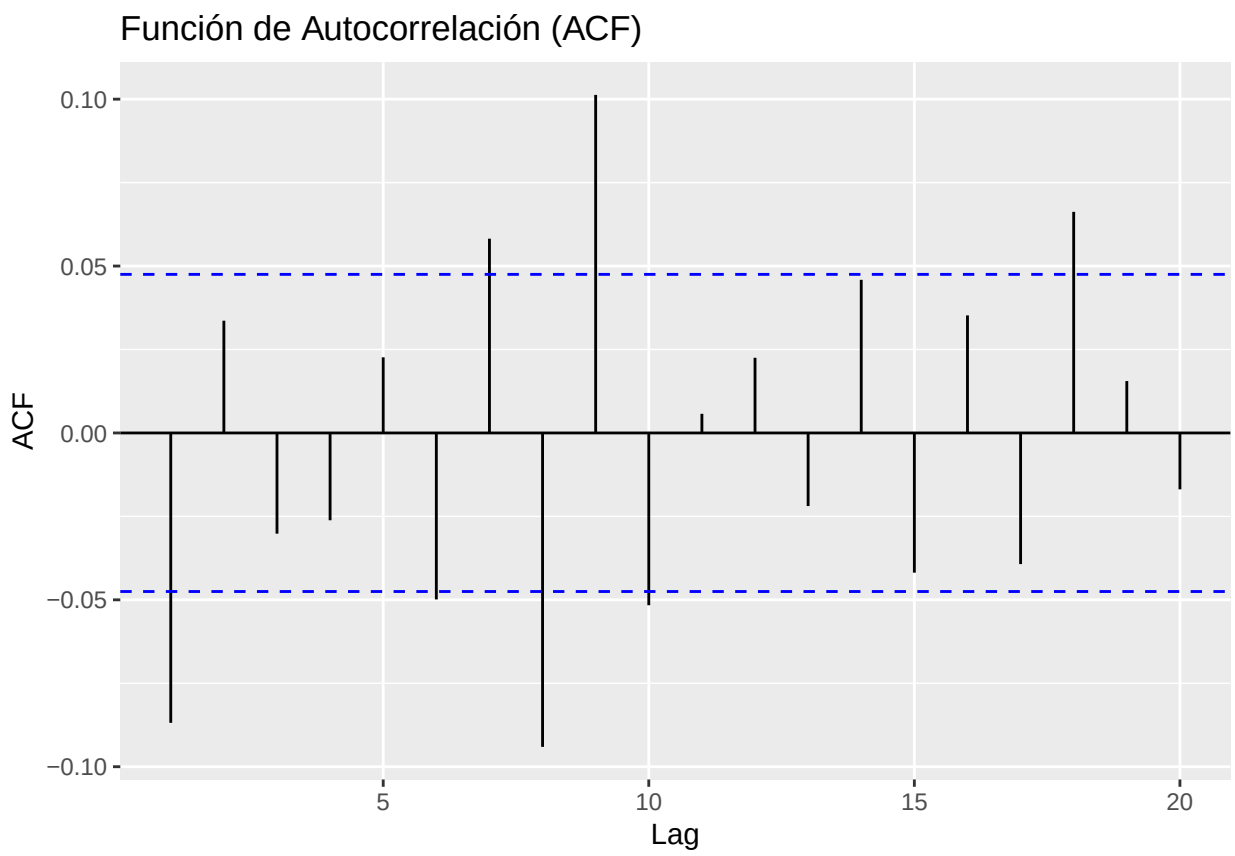
```
## Warning in adf.test(rendimientos, alternative = "stationary"): p-value smaller
## than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  rendimientos
## Dickey-Fuller = -11.99, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

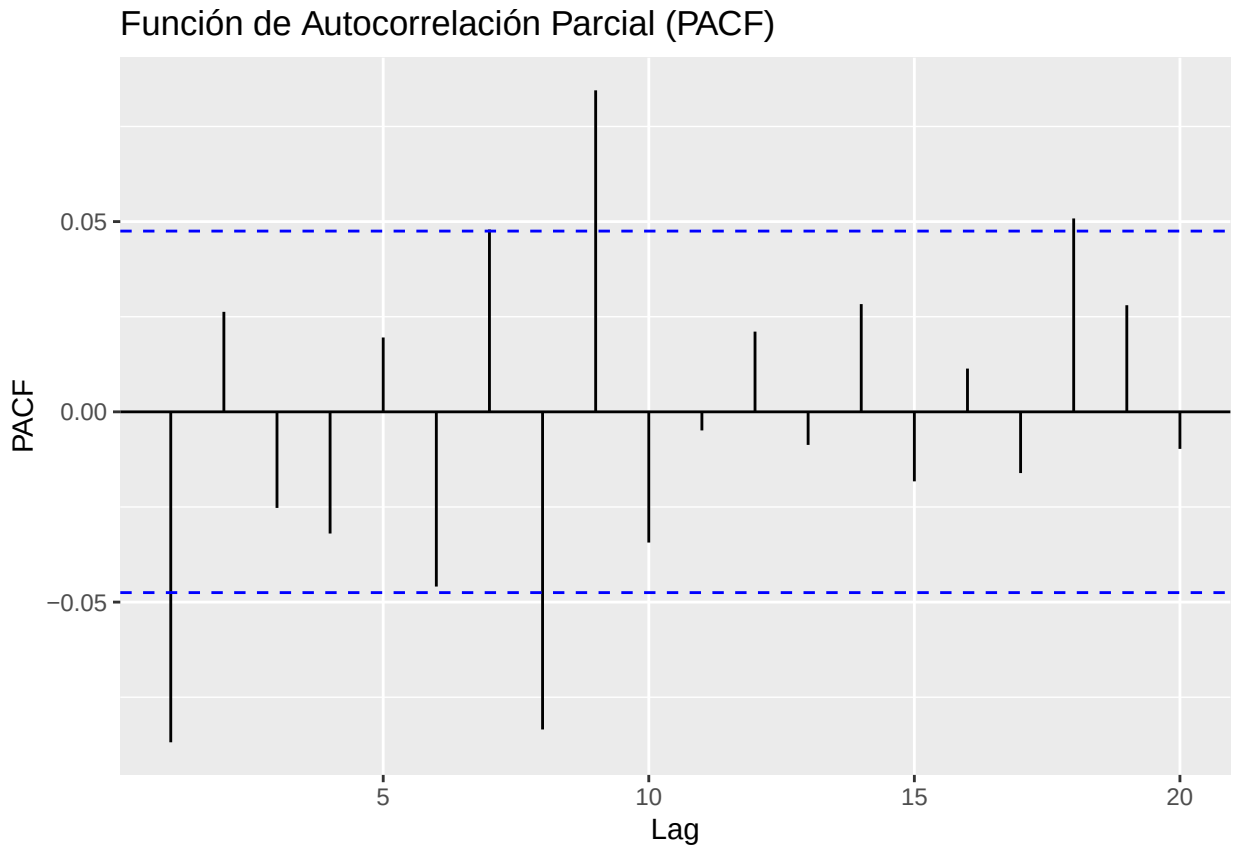
El p-valor es bajo (0.01), lo que indica que podemos rechazar la hipótesis nula.

El modelo ARIMA (Autoregressive Integrated Moving Average) es ampliamente utilizado para modelar series de tiempo estacionarias. Analizamos las funciones de autocorrelación (ACF) y autocorrelación parcial (PACF) para identificar los órdenes del modelo.

```r
# Graficar ACF y PACF
ggAcf(rendimientos, lag.max = 20) + ggtitle("Función de Autocorrelación (ACF)")
```




```r
ggPacf(rendimientos, lag.max = 20) + ggtitle("Función de Autocorrelación Parcial (PACF)")
```



Las gráficas nos ayudan a identificar posibles valores de p y q para el modelo ARIMA(p, d, q). En los rendimientos financieros, a menudo se observa poca autocorrelación significativa. También podríamos hacer que el algoritmo seleccione los valores para el modelo. Para esto utilizamos la función `auto.arima()`, y seleccionará estos usando criterios de información (AICc).

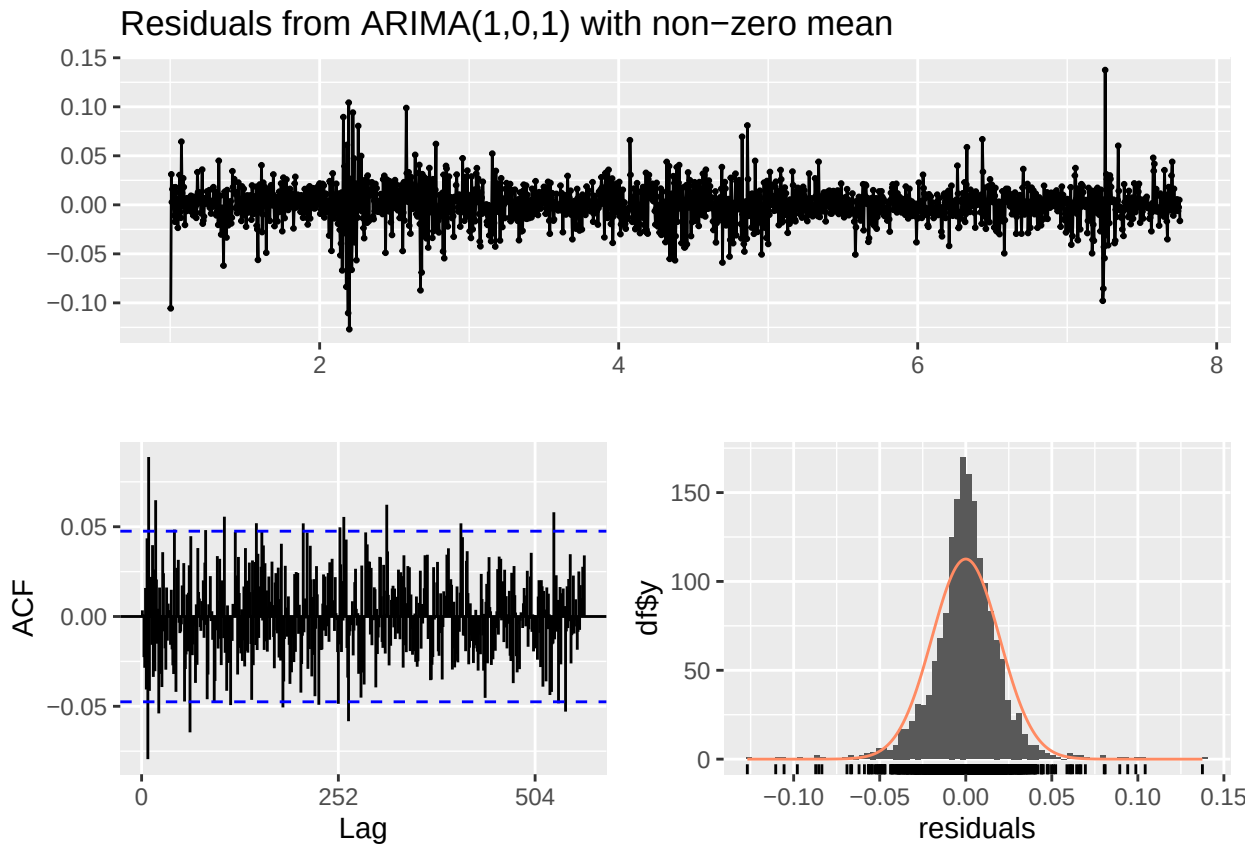
```
# Selección automática del modelo ARIMA
modelo_arima <- auto.arima(rendimientos, seasonal = FALSE)

# Resumen del modelo
summary(modelo_arima)
```

```
## Series: rendimientos
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1     ma1    mean
##       -0.3563  0.2687  0.0011
## s.e.   0.1880  0.1926  0.0004
##
## sigma^2 = 0.000388:  log likelihood = 4270.74
## AIC=-8533.49   AICc=-8533.46   BIC=-8511.73
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE        ACF1
## Training set 7.868675e-07 0.01967924 0.01368463 NaN  Inf 0.6399815 0.003433146
```

El modelo seleccionado es un ARIMA(1,0,0), es decir, un modelo ARMA con un término autorregresivo pero sin media móvil, realmente, un modelo AR.

Verificamos si el modelo ajustado cumple con los supuestos necesarios.

```
# Graficar los residuos
checkresiduals(modelo_arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 404.26, df = 338, p-value = 0.007673
##
## Model df: 2.   Total lags used: 340
```

- Residuos Estandarizados: Deben comportarse como ruido blanco ( ).
- ACF de Residuos: No debe mostrar autocorrelación significativa ( ).
- Prueba de Ljung-Box: Un p-valor alto indica que no hay autocorrelación en los residuos ( ).

Podríamos verificar la heteroscedasticidad. En series financieras, es común que exista esta estructura en los datos.

```
# Prueba de Arch
library(FinTS)
```

```
##
## Attaching package: 'FinTS'

## The following object is masked from 'package:forecast':
##
##     Acf

## The following object is masked from 'package:gt':
##
##     sp500
```

```
ArchTest(resid(modelo_arima))
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  resid(modelo_arima)
## Chi-squared = 252.27, df = 12, p-value < 2.2e-16
```

La prueba es significativa,

Para capturar la heterocedasticidad, utilizamos modelos GARCH (Generalized Autoregressive Conditional Heteroskedasticity). Utilizamos el paquete `rugarch` para ajustar un modelo GARCH.

```
# Instalar y cargar el paquete rugarch
#install.packages("rugarch")
library(rugarch)
```

```
## Loading required package: parallel

##
## Attaching package: 'rugarch'

## The following object is masked from 'package:purrr':
##
##     reduce
```

```
# Especificar el modelo ARIMA(1,0,0)-GARCH(1,1)
especificacion <- ugarchspec(mean.model = list(armaOrder = c(1,0)),
                             variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                             distribution.model = "norm")

# Ajustar el modelo
modelo_garch <- ugarchfit(spec = especificacion, data = rendimientos)

# Resumen del modelo
show(modelo_garch)
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
```

```
## *----------------------------------*
##
## Conditional Variance Dynamics
## ----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001757    0.000419  4.19360 0.000027
## ar1    -0.006134    0.026550 -0.23104 0.817286
## omega   0.000016    0.000008  2.05969 0.039428
## alpha1  0.103685    0.024239  4.27758 0.000019
## beta1   0.850075    0.023875 35.60535 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001757    0.000927  1.89472 0.058129
## ar1    -0.006134    0.027401 -0.22386 0.822864
## omega   0.000016    0.000042  0.38990 0.696611
## alpha1  0.103685    0.119621  0.86678 0.386064
## beta1   0.850075    0.100043  8.49709 0.000000
##
## LogLikelihood : 4449.132
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.2222
## Bayes        -5.2063
## Shibata      -5.2223
## Hannan-Quinn -5.2163
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                   0.004899  0.9442
## Lag[2*(p+q)+(p+q)-1][2]  0.078508  1.0000
## Lag[4*(p+q)+(p+q)-1][5]  0.916086  0.9586
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.2805  0.5964
## Lag[2*(p+q)+(p+q)-1][5]    1.7351  0.6822
## Lag[4*(p+q)+(p+q)-1][9]    3.3195  0.7053
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
```

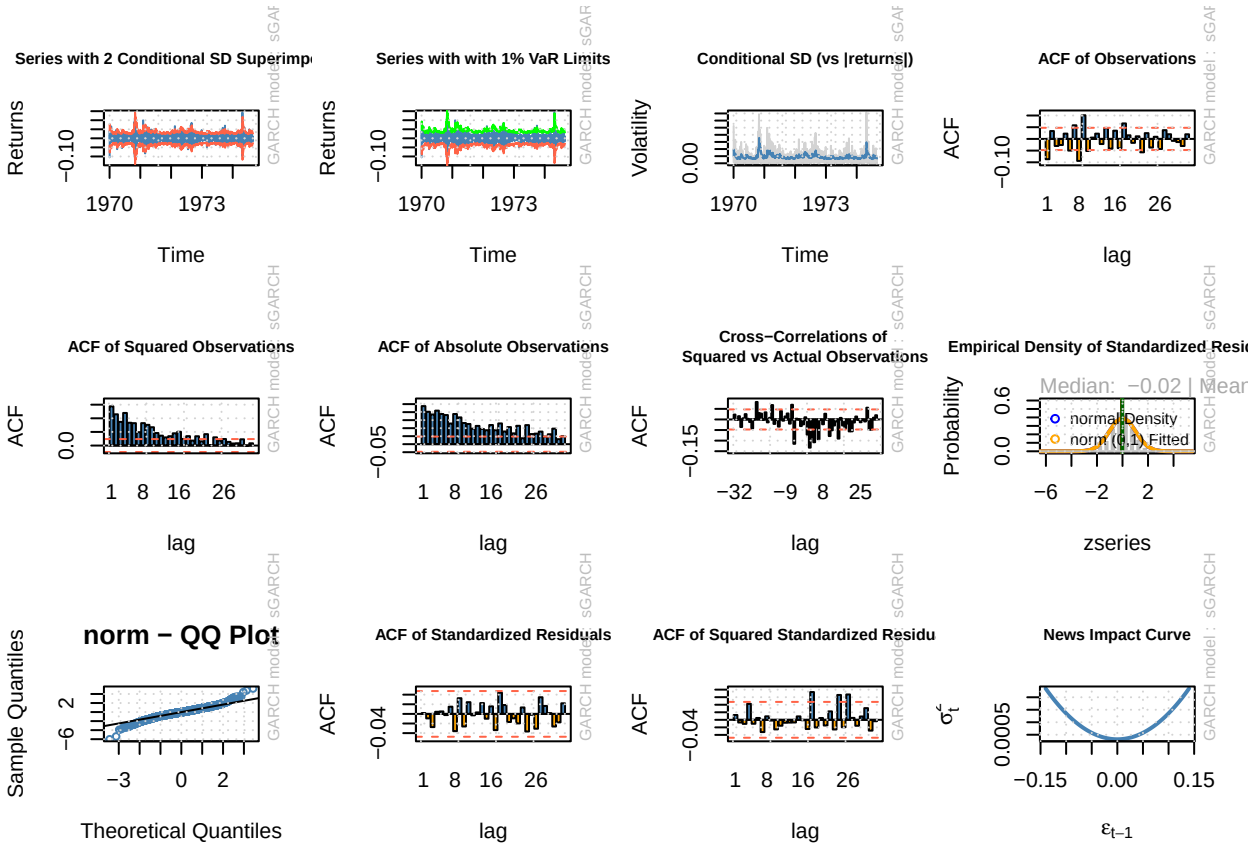
```
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1955 0.500 2.000  0.6583
## ARCH Lag[5]    2.7575 1.440 1.667  0.3269
## ARCH Lag[7]    3.7495 2.315 1.543  0.3841
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  6.5409
## Individual Statistics:
## mu     0.50048
## ar1    0.41970
## omega  0.77608
## alpha1 0.12120
## beta1  0.06688
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           0.2167 0.8285
## Negative Sign Bias  1.2057 0.2281
## Positive Sign Bias  0.2568 0.7974
## Joint Effect        3.1504 0.3690
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     63.95    9.048e-07
## 2    30     65.11    1.372e-04
## 3    40     98.52    4.701e-07
## 4    50     94.30    1.081e-04
##
##
## Elapsed time : 0.09418893
```

- Mu (μ): El término de la media es significativo ($p < 0.01$), lo que indica que hay un rendimiento promedio distinto de cero.
- AR(1): El coeficiente AR(1) no es significativo ($p \approx 0.42$), lo que sugiere que el término autorregresivo puede no ser necesario.
- Omega (ω): Parámetro significativo que representa la varianza incondicional.
- Alpha1 (α_1) y Beta1 (β_1): Ambos son altamente significativos, confirmando la presencia de efectos ARCH y GARCH.

Las pruebas diagnósticas indican que no aparentan haber ni autocorrelación significativa en los residuos ni en los residuos al cuadrado, así como tampoco aparentan haber sesgos significativos en los residuos (signos). El modelo GARCH parece haber capturado adecuadamente la heteroscedasticidad. Los parámetros podrían ser algo inestables (verificar Nyblom)

```r
# Graficar los residuos estandarizados
plot(modelo_garch, which = "all")
```

```
##
## please wait...calculating quantiles...
```



Podría considerar un modelo más sencillo sin AR(1), y con colas más amplias en distribución t.

```r
# Especificación de un modelo GARCH(1,1) sin término AR(1)
especificacion_simple <- ugarchspec(mean.model = list(armaOrder = c(0,0)),
                                    variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                                    distribution.model = "norm")

modelo_garch_simple <- ugarchfit(spec = especificacion_simple, data = rendimientos)

# Modelo GARCH con distribución t de Student
especificacion_t <- ugarchspec(mean.model = list(armaOrder = c(0,0)),
                               variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                               distribution.model = "std")

modelo_garch_t <- ugarchfit(spec = especificacion_t, data = rendimientos)
modelo_garch_t
```

```
##
```

```
## *----------------------------------*
## *          GARCH Model Fit         *
## *----------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001645    0.000356   4.6259    4e-06
## omega   0.000014    0.000003   5.3664    0e+00
## alpha1  0.098953    0.007006  14.1242    0e+00
## beta1   0.867278    0.013834  62.6927    0e+00
## shape   4.858992    0.465962  10.4279    0e+00
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001645    0.000368   4.4711 0.000008
## omega   0.000014    0.000005   2.9376 0.003307
## alpha1  0.098953    0.018211   5.4336 0.000000
## beta1   0.867278    0.015912  54.5048 0.000000
## shape   4.858992    0.702672   6.9150 0.000000
##
## LogLikelihood : 4520.934
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.3066
## Bayes        -5.2906
## Shibata      -5.3066
## Hannan-Quinn -5.3007
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.0144  0.9045
## Lag[2*(p+q)+(p+q)-1][2]    0.0755  0.9372
## Lag[4*(p+q)+(p+q)-1][5]    0.9305  0.8752
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.4434  0.5055
## Lag[2*(p+q)+(p+q)-1][5]    1.8899  0.6447
## Lag[4*(p+q)+(p+q)-1][9]    3.5150  0.6718
## d.o.f=2
##
```

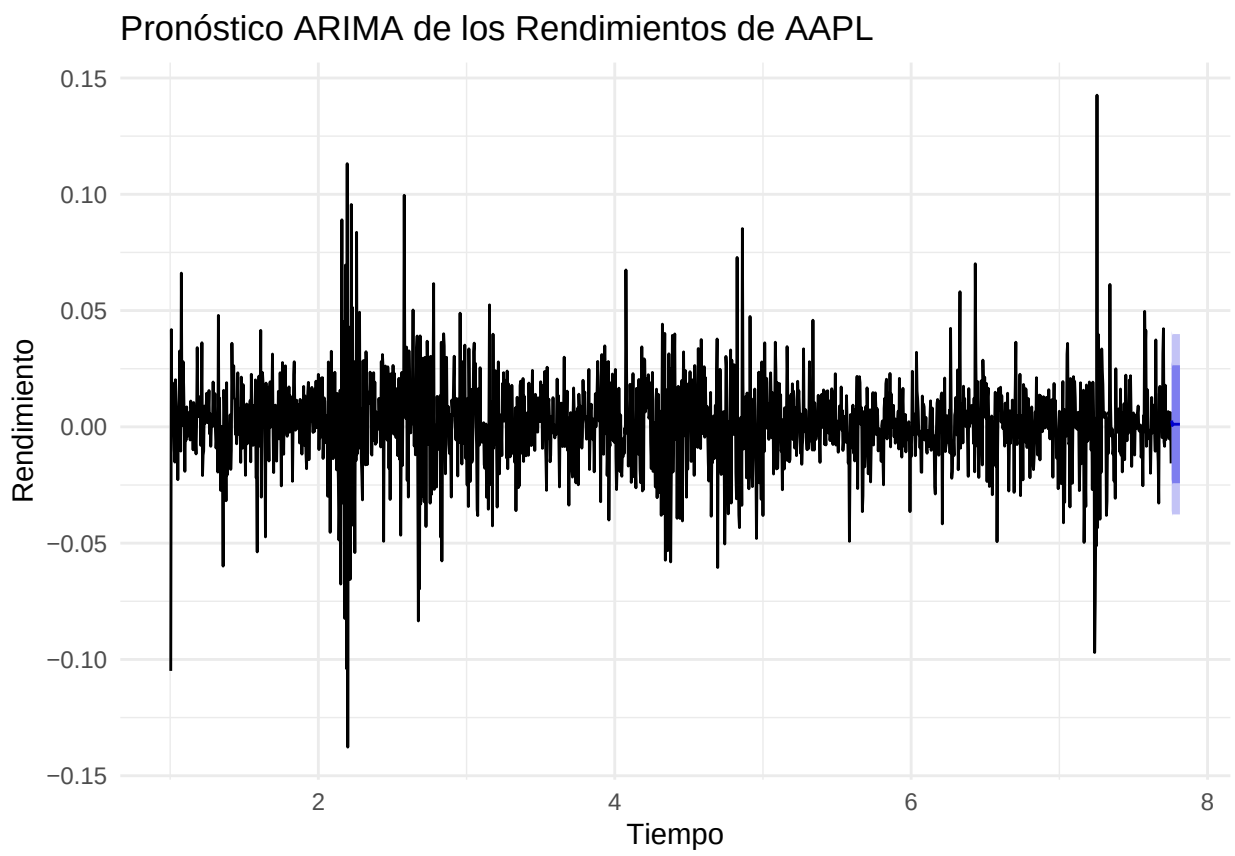
```
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1404 0.500 2.000  0.7079
## ARCH Lag[5]    2.8321 1.440 1.667  0.3151
## ARCH Lag[7]    3.8448 2.315 1.543  0.3695
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  15.2003
## Individual Statistics:
## mu     0.6110
## omega  3.5493
## alpha1 0.5532
## beta1  0.4535
## shape  0.8674
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           0.2684 0.7884
## Negative Sign Bias  1.2589 0.2082
## Positive Sign Bias  0.1646 0.8693
## Joint Effect        3.3637 0.3389
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     16.61       0.6160
## 2    30     28.45       0.4941
## 3    40     27.35       0.9194
## 4    50     32.61       0.9655
##
##
## Elapsed time : 0.05744004
```

Finalmente, quiero visualizar usando el ARIMA(1,0,0) el rendimiento futuro de AAPL por 15 días.

Uso para esto la función `forecast()`: Esta función del paquete forecast genera pronósticos basados en el modelo ARIMA ajustado (`modelo_arima`). Con `h=15` indico que quiero saber los próximos 15 periodos (días). Graficamos ese pronóstico con `autoplot`, y algo de personalización.

```
# Pronóstico de los próximos 15 días
pronostico_arima <- forecast(modelo_arima, h = 15)

# Graficar el pronóstico
autoplot(pronostico_arima) +
  labs(title = "Pronóstico ARIMA de los Rendimientos de AAPL",
       x = "Tiempo",
```

```
       y = "Rendimiento") +
  theme_minimal()
```




```
# Pronóstico de la volatilidad futura
pronostico_garch <- ugarchforecast(modelo_garch, n.ahead = 10)

# Extraer la volatilidad pronosticada
volatilidad_pronosticada <- sigma(pronostico_garch)

# Mostrar los resultados
volatilidad_pronosticada
```

```
##       1974-08-30
## T+1  0.01410122
## T+2  0.01435463
## T+3  0.01459223
## T+4  0.01481529
## T+5  0.01502495
## T+6  0.01522223
## T+7  0.01540803
## T+8  0.01558317
## T+9  0.01574841
## T+10 0.01590440
```

- Los pronósticos nos ayudan a entender las posibles tendencias futuras en los rendimientos y la volatilidad.

Conclusiones

- Análisis Exploratorio: Identificamos que los precios de las acciones no son estacionarios, pero los rendimientos logarítmicos sí lo son.
- Modelado ARIMA: Ajustamos un modelo ARIMA para capturar la dinámica en los rendimientos.
- Volatilidad y GARCH: Detectamos heterocedasticidad y utilizamos un modelo GARCH para modelar la volatilidad condicional.
- Pronósticos: Generamos pronósticos de rendimientos y volatilidad.

Es importante tener en cuenta que los mercados financieros son influenciados por muchos factores impredecibles, y ningún modelo puede capturar completamente esa complejidad. Por lo tanto, los pronósticos deben interpretarse con cautela y complementarse con análisis cualitativos y contextualizados.

Recomiendo aplicar estos métodos a diferentes activos financieros o índices para fortalecer la comprensión. Además de R, existen otras herramientas como Python y EViews que también son útiles para el análisis de series de tiempo. EViews está en declive según tengo entendido, pero sigue siendo utilizado en varios espacios.

Qué aprendimos en este taller

Aprendimos la existencia de varios modelos, maneras de representar gráficamente sus resultados, y evaluar su utilidad. He añadido elementos que aumentan la descripción de pedazos del taller, y que buscan corregir el orden en el que se trabajaron ejemplos.

Nos veremos próximamente (fecha por determinar en el taller) con un nuevo tema: análisis de redes.