Laboratorio 2 – Regresión Lineal Simple

Table of contents

El propósito de este laboratorio es practicar el uso de **R** para realizar **pruebas de hipótesis** y ejecutar una **regresión MCO (OLS)** básica. El laboratorio puede realizarse en grupo. Para obtener el crédito, sube tu script **.R** al lugar correspondiente en **Canvas**. Si el trabajo se realiza en grupo, incluye los nombres de todos los integrantes en un comentario al inicio del archivo.

Primeros pasos

Abre un nuevo script de R (llámalo `ICL2_XYZ.R`, donde `XYZ` son tus iniciales) y agrega lo siguiente al inicio:

```
# Instalar paquetes solo si faltan (igual que el estilo de tus labs)
pkgs <- c("tidyverse", "modelsummary", "wooldridge", "broom", "kableExtra")
to_install <- pkgs[!pkgs %in% rownames(installed.packages())]
if (length(to_install) > 0) {
  install.packages(to_install, repos = "http://cran.us.r-project.org")
}

library(tidyverse)
```

```r
library(modelsummary)
library(broom)       # se usa para tidy() y glance()
library(wooldridge)
library(kableExtra)  # <- CLAVE: evita 'tblr' en PDF usando tablas LaTeX clásicas
```

Carga el conjunto de datos `audit` del paquete `wooldridge`:

```r
df <- as_tibble(audit)
```

Prueba unilateral

El conjunto de datos `audit` contiene tres variables: `w`, `b` y `y`.

- `b` indica si el currículum de la persona **negra** de un par fue seleccionado para una oferta de trabajo.
- `w` indica lo mismo, pero para la persona **blanca** del par.
- `y` corresponde a la diferencia entre ambas variables, es decir: `y = b - w`.

Queremos probar la siguiente hipótesis:

$$H_0 : \mu = 0 \quad \text{vs} \quad H_a : \mu < 0$$

donde $\mu = \theta_B - \theta_W$, es decir, la diferencia en las tasas de oferta laboral entre personas negras y blancas.

La función `t.test()` en R

Para realizar una prueba t en R, basta con entregar la información apropiada a la función `t.test()`.

¿Cómo sabemos cuál es la "información apropiada"?

- En la consola de RStudio, escribe `?t.test` y presiona Enter.
- Se abrirá una página de ayuda en la parte inferior derecha con el título *"Student's t-Test"*.
- En la sección `Usage` aparece `t.test(x, ...)`.
  - Esto significa que, como mínimo, debemos entregar un objeto `x`.
  - Los puntos suspensivos `...` indican que podemos entregar argumentos adicionales.
- En `Arguments` se explica que `x` debe ser "a (non-empty) numeric vector of data values".
  - Es decir, R espera que entreguemos un **vector numérico**, usualmente una **columna** de un data frame.

- La ayuda también indica configuraciones por defecto:
  - `alternative = "two.sided"`
  - `mu = 0`
  - y otras opciones que no utilizaremos por ahora.

Ahora realizamos la prueba de hipótesis descrita anteriormente:

```
t.test(df$y, alternative = "less")
```

R calcula automáticamente el estadístico t usando la fórmula:

$$\frac{\overline{y} - \mu}{SE_{\bar{y}}}$$

Lo único que tuvimos que entregar a R fue la muestra de datos (`y`) y el valor nulo (0, que es el valor por defecto de `t.test()`).

Interpretación del resultado de `t.test()`

Ahora que hemos realizado la prueba t, ¿cómo interpretamos el resultado? Al ejecutar el script, deberías ver algo como:

```
> t.test(df$y, alternative="less")

    One Sample t-test

data:  df$y
t = -4.2768, df = 240, p-value = 1.369e-05
alternative hypothesis: true mean is less than 0
95 percent confidence interval:
        -Inf -0.08151529
sample estimates:
 mean of x
-0.1327801
```

R reporta el valor del estadístico t, los grados de libertad y el p-value asociado. En este caso, el p-value es aproximadamente 0.00001369, mucho menor que 0.05 (nuestro nivel de significancia). Por lo tanto, rechazamos H_0.

Prueba bilateral

Ahora supongamos que queremos probar si las tasas de oferta laboral para personas negras son **distintas** de las de personas blancas.

Queremos probar:

$$H_0 : \theta_b = \theta_w \quad \text{vs} \quad H_a : \theta_b \neq \theta_w$$

Esta prueba considera el caso en que podría existir *discriminación inversa* (por ejemplo, por políticas de acción afirmativa).

El código para realizar esta prueba es similar al anterior:

```
t.test(df$b, df$w, alternative = "two.sided", paired = TRUE)
```

Notarás que el estadístico t es el mismo que en la prueba unilateral, pero el p-value de la prueba bilateral es aproximadamente el doble, porque ahora se consideran desviaciones en ambas direcciones.

Tu primera regresión (de este curso)

Ahora cargaremos un nuevo conjunto de datos y ejecutaremos una regresión MCO. Este set contiene estadísticas anuales por condado en Estados Unidos, incluyendo conteos de distintos delitos y características demográficas.

```
df <- as_tibble(countymurders)
```

Un comando útil para una vista rápida de un dataset es `glimpse()`:

```
glimpse(df)
```

`glimpse()` muestra el número de observaciones, el número de variables y el nombre y tipo de cada variable (por ejemplo, integer, double).[1]

[1] "double" significa *double precision floating point*, y es una forma (computacional) de expresar un número real (en contraste con un entero o un racional).

Sintaxis de regresión

Para estimar una regresión de y sobre x en R, usamos:

```
est <- lm(y ~ x, data = data.name)
```

Aquí, `est` es un objeto donde se guardan los coeficientes y otra información del modelo. `lm()` significa "linear model" y es la función que calcula los coeficientes MCO.

Regresar tasa de asesinatos sobre tasa de ejecuciones

Usando el `df` anterior, estimamos una regresión donde `murders` es la variable dependiente y `execs` la independiente:

```
est <- lm(murders ~ execs, data = df)
```

Para ver los resultados en un formato legible:

```
tidy(est)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)     6.84     0.242      28.3 3.97e-174
2 execs          65.5      2.15       30.5 7.44e-202
```

En la columna `estimate` se observan los coeficientes estimados para β_0 (el intercepto) y β_1 (`execs`). El objeto `est` contiene más información que usaremos después en el curso.

También puedes ver el R^2 escribiendo:

```
glance(est)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic   p.value    df   logLik    AIC    BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>    <dbl>  <dbl>  <dbl>
1    0.0243        0.0243  46.6      930. 7.44e-202     1 -196508. 3.93e5 3.93e5
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Por ahora, concéntrate solo en el término R^2.

Tabla final del modelo (sin error `tblr` en PDF)

Esta es la receta que te permite avanzar: en vez de dejar `modelsummary()` en modo automático (que en tu PDF está generando `\\begin{tblr}`), forzamos una salida compatible con LaTeX clásico usando `kableExtra`.

```
modelsummary(est, output = "kableExtra") |>
  kable_styling(
    full_width = FALSE,
    position = "center",
    latex_options = c("hold_position")
  )
```

	(1)
(Intercept)	6.838
	(0.242)
execs	65.465
	(2.146)
Num.Obs.	37 349
R2	0.024
R2 Adj.	0.024
AIC	393 021.2
BIC	393 046.8
Log.Lik.	−196 507.599
F	930.365
RMSE	46.64