IN4143 – Análisis de Datos e Inferencia Causal

Primavera 2024

Profesores: Daniel Schwartz, Omar Pérez.

Auxiliares: Antonia Aceituno, Camila Galarce.

Ayudantes: Guillermo Escobar, Bastián Medina.

LABORATORIO 1

El objetivo de este laboratorio es comprender el rol de la significancia estadística y la interpretación del p-valor. Esta evaluación es individual. Para responder, puede revisar el material del curso disponible en u-cursos y las cápsulas de clase.

En este caso, suponga que una empresa de servicios de viajes está realizando un "A/B testing" en su sitio web con clientes registrados. Los clientes (N = 1000) son asignados aleatoriamente a ver la página con o sin la intervención. Se han identificado varias variables dependientes, como el tiempo de visita, los clics en servicios y la cantidad de viajes desplegados, entre otras.

Para su informe, conteste:

1. Utilice el código borrando la sección en rojo y responda: ¿Qué representa el resultado que arroja el código? Explique brevemente. [20 puntos]
2. Utilice el código completo y responda: ¿Qué representa el resultado que arroja el código? Explique brevemente. [20 puntos]
3. Modifique el código y realice un gráfico con el porcentaje de "falsos positivos" en el eje Y. En el eje X, presente el caso cuando se tiene una sola variable dependiente, luego dos variables dependientes reportando la que dio significativa, luego tres variables, y así sucesivamente hasta llegar a diez. ¿Qué puede concluir de sus resultados sobre los efectos de la intervención? [50 puntos]
4. ¿Cómo se relacionan las preguntas anteriores con el p-valor? [10 puntos]

Bonus: ¿Qué sucede si cambia el número de participantes? ¿Y el número de simulaciones? [5 puntos]

Código R:

```
# Definir número de observaciones
num_obs <- 1000
# Muchos experimentos…
total_sim <- 1000
bingo <- 0
for (simul in 1:total_sim) {
  # Llegan participantes:
  participant <- 1:num_obs
  # Les damos un tratamiento en forma aleatoria:
  u <- runif(num_obs)
  tratamiento <- ifelse(rank(u) <= num_obs / 2, 1, 0)

  # Tomamos sus reacciones (cree las variables dependientes que necesite):
  vardep1 <- qnorm(runif(num_obs))
  vardep2 <- qnorm(runif(num_obs))
  vardep3 <- qnorm(runif(num_obs))
  vardep4 <- qnorm(runif(num_obs))
  vardep5 <- qnorm(runif(num_obs))

  # Veamos si hay diferencias (agregue “else if” para considerar otras variables dependientes.)
  # Sólo basta que una sea “significativa”.
  reg1 <- lm(vardep1 ~ tratamiento)
  if (summary(reg1)$coefficients[2, 4] < 0.05) {
    bingo <- bingo + 1
  } else {
    reg2 <- lm(vardep2 ~ tratamiento)
    if (summary(reg2)$coefficients[2, 4] < 0.05) {
      bingo <- bingo + 1
    } else {
      reg3 <- lm(vardep3 ~ tratamiento)
      if (summary(reg3)$coefficients[2, 4] < 0.05) {
        bingo <- bingo + 1
      } else {
        reg4 <- lm(vardep4 ~ tratamiento)
        if (summary(reg4)$coefficients[2, 4] < 0.05) {
          bingo <- bingo + 1
        } else {
          reg5 <- lm(vardep5 ~ tratamiento)
          if (summary(reg5)$coefficients[2, 4] < 0.05) {
            bingo <- bingo + 1
          }
        }
      }
    }
  }
}

print(bingo / total_sim)
```

Código Python:

```
import numpy as np
from scipy.stats import norm
import statsmodels.api as sm

# Definir número de observaciones
num_obs = 1000

# Muchos experimentos…
total_sim = 1000
bingo = 0

for simul in range(total_sim):
    # Llegan participantes:
    participant = np.arange(1, num_obs + 1)

    # Les damos un tratamiento en forma aleatoria:
    u = np.random.uniform(0, 1, num_obs)
    tratamiento = np.where(np.argsort(u) < num_obs / 2, 1, 0)

    # Tomamos sus reacciones (crea las variables dependientes que necesites):
    vardep1 = norm.ppf(np.random.uniform(0, 1, num_obs))
    vardep2 = norm.ppf(np.random.uniform(0, 1, num_obs))
    vardep3 = norm.ppf(np.random.uniform(0, 1, num_obs))
    vardep4 = norm.ppf(np.random.uniform(0, 1, num_obs))
    vardep5 = norm.ppf(np.random.uniform(0, 1, num_obs))

    # Veamos si hay diferencias (agrega “elif” para considerar otras variables dependientes.)
    # Sólo basta que una sea “significativa”.
    X = sm.add_constant(tratamiento)  # Agrega la constante para la regresión
    model1 = sm.OLS(vardep1, X).fit()

    if model1.pvalues[1] < 0.05:
        bingo += 1
    else:
        model2 = sm.OLS(vardep2, X).fit()
        if model2.pvalues[1] < 0.05:
            bingo += 1
        else:
            model3 = sm.OLS(vardep3, X).fit()
            if model3.pvalues[1] < 0.05:
                bingo += 1
            else:
                model4 = sm.OLS(vardep4, X).fit()
                if model4.pvalues[1] < 0.05:
                    bingo += 1
                else:
                    model5 = sm.OLS(vardep5, X).fit()
                    if model5.pvalues[1] < 0.05:
```

```
                    bingo += 1

print(bingo / total_sim)
```