Profesor: Manuel Fernández

Estudiantes:
- Federico Dueñas - 201630985
- Germán C. Rodríguez - 201413436

Taller 1

Econometría Avanzada, 2022-1
Fecha de Entrega: 25 de febrero

Universidad de los Andes

Primer ejercicio

Por motivo de la grave situación económica en el país luego de la pandemia, el Gobierno lanzó un programa social destinado a mejorar los ingresos de hogares pequeños cultivadores de café. En particular, la ayuda consistió en una capacitación acerca de la elección y uso de fertilizantes, ofrecida al jefe de hogar, junto con la entrega de una cantidad fija de dinero. Sin embargo, los recursos para desarrollar esta política eran limitados, por lo que sólo se podía brindar el programa a algunos de los hogares registrados en las bases de datos del Gobierno (los hogares no se tenían que inscribir para ser candidatos a recibir el programa). Por lo tanto, buscando evitar favorecimientos indebidos, se eligió aleatoriamente un subconjunto de hogares que recibirían el programa. Todos las familias seleccionadas para recibir la ayuda participaron activamente.

En busca de evaluar la efectividad de la medida, el Gobierno recolectó información acerca de la cosecha posterior al programa para todos los n hogares que fueron elegibles para el programa (esto es, tanto para los que recibieron el programa como para aquellos que no). Entre la información recolectada, se cuenta con la cantidad de kilogramos de café producido por hectárea cultivada por cada hogar C_i, así como una dummy D_i que toma el valor de uno si el hogar participó en el programa y cero de lo contrario.

Confiando en las habilidades de los estudiantes de econometría avanzada, a ustedes los contrata el Departamento Nacional de Planeación para llevar a cabo la evaluación de impacto respectiva.

Su jefe les dice que una manera de modelar el problema es a través de un modelo de regresión lineal dado por

$$C_i = \alpha + \tau D_i + \epsilon_i \tag{1}$$

donde ϵ_i es un componente aleatorio idiosincrásico de media cero ($\mathbb{E}[\epsilon_i] = 0$) y con varianza σ^2 ($\mathbb{E}[\epsilon_i^2] = \sigma^2$).

a) Sean $C_i(1)$ y $C_i(0)$ los resultados potenciales de haber participado o no en el programa respectivamente. Similarmente, sean $\epsilon_i(1)$ y $\epsilon_i(0)$ los resultados potenciales análogos del error idiosincrático. Suponga además que $\mathbb{E}[\epsilon_i(1)] = \mathbb{E}[\epsilon_i(0)] = 0$.

   i) ¿Cuáles son las formas funcionales de $C_i(1)$ y $C_i(0)$ inducidas por el modelo (1)?

   - Solución:

$$C_i(1) = \alpha + \tau + \epsilon_i(1)$$
$$C_i(0) = \alpha + \epsilon_i(0)$$

   ii) ¿Cuál parámetro del modelo captura el ATT del programa? Justifique matemáticamente.

   - Solución:

     El ATT se define como: $\tau_{\hat{ATT}} = (\bar{C}/D = 1) - (\bar{C}/D = 0)$.

     Sabemos que $E[\epsilon_i/D_i] = 0$, por lo que en el modelo de regresión $\hat{\tau}$ es insesgado y consistente.

     Entonces, tomando expectativa condicional del ATT definido y teniendo en cuenta el supuesto de exogeneidad se tiene que:

$$E[C_i(1)/D_i = 1] = E(\alpha + \tau + \epsilon_i/D_i = 1)$$
$$E[C_i(1)/D_i = 1] = \alpha + \tau + \cancel{E[\epsilon_i/D_i = 1]}$$
$$E[C_i(1)/D_i = 1] = \alpha + \tau$$

$$E[C_i(0)/D_i = 0] = E[\alpha + \epsilon_i/D_i = 0]$$
$$E[C_i(0)/D_i = 0] = \alpha + \cancel{E[\epsilon_i/D_i = 0]}$$
$$E[C_i(0)/D_i = 0] = \alpha$$

Restando las dos expresiones de arriba para tener al parámetro del τ_{ATT}:

$$\tau_{ATT} = E[C_i(1)/D_i = 1] - E[C_i(0)/D_i = 0]$$

$$\tau_{ATT} = \alpha + \tau - \alpha$$
$$\tau_{ATT} = \tau$$

Por lo que τ es el parámetro que captura del ATT en el modelo.

iii) Imagine un escenario donde el programa, en lugar de ser asignado aleatoriamente, se entregaba a las familias que viven en lugares con climas menos favorables para el cultivo. ¿En este escenario el parámetro del inciso ii) sigue capturando el ATT?
Pista: Recuerde que

$$C_i = D_i\, C_i(1) + (1 - D_i)\, C_i(0); \qquad \epsilon_i = D_i\, \epsilon_i(1) + (1 - D_i)\, \epsilon_i(0)$$

- Solución:
Dado que el tratamiento ya no es aleatorio, ahora va a haber un sesgo de selección,por lo que el parámetro $\hat{\tau}$ ya no es estimador consistente de τ_{ATT}:

$$\tau_{ATT} = E[C_i(1)/D_i] - E[C_i(0)/D_i]$$

Ya que no hay autorización: $E[C_i(0)/D_i = 0] \neq E[C_i(0)/D_i = 1]$ entonces:

$$\tau_{ATT} = E[C_i(1) - C_i(0)/D_i = 1] + [E[C_i(0)/D_i = 1] - E[C_i(0)/D_i = 0]]$$

Con lo que se llega a que:

$$\tau_{ATT} = \hat{\tau} + E[C_i(0)/D_i = 1] - E[C_i(0)/D_i = 0]$$

Por lo que vemos que $\hat{\tau}$ no captura el τ_{ATT}.

b) Bajo el cumplimiento de los supuestos del modelo clásico lineal, demuestre que el estimador de MCO $\hat{\tau}$ es un estimador consistente del ATT y derive explícitamente su distribución asintótica (la varianza asintótica debe depender únicamente de n, σ^2 y $p = \mathbb{P}(D_i = 1)$).

Pista: Recuerde el siguiente teorema visto en clase:

Una secuencia de vectores aleatorios $\{x_N : N = 1, 2, \ldots\}$ de $K \times 1$ converge en distribución al vector aleatorio x, si para cualquier vector no aleatorio c de $K \times 1$,

$$c^T x_N \xrightarrow{d} c^T x$$

- Solución:
Tenemos que por definición:
$$ATT = E[C_i(1)/D_i] - E[C_i(0)/D_i]$$
$$ATT = E[C_i(1) - C_i(0)/D_i = 1] + E[C_i(0)/D_i = 1] - E[C_i(0)/D_i = 0]$$
Y bajo supuestos MCL se tiene que:
$$ATT = \tau$$
Por lo que se que $\tau = ATT$, entonces $\tau_{\hat{MCO}}$ es consistente de τ? :

Para probarlo, primero sabemos que por definición y bajo supuestos MCL, $\tau_{\hat{MCO}}$ es:

$$\tau_{\hat{MCO}} = \bar{C}_i(1)/D_i = 1 - \bar{C}_i(0)/D_i = 0$$

Aplicando $plim$ a la expresión anterior se tiene:

$$plim(\hat{\tau_{MCO}}) = plim(\bar{C}_i(1)/D_i = 1) - plim(\bar{C}_i(0)/D_i = 0)$$

Por WLLN tenemos que:

$$plim(\hat{\tau_{MCO}}) = E[C_i(1)/D_i = 1] - E[C_i(0)/D_i = 0]$$

Y como ya se mostró, esta expresión bajo supuestos de MCL resulta en:

$$plim(\hat{\tau_{MCO}}) = \tau$$

Por lo que se concluye que el $\hat{\tau_{MCO}}$ es estimador consistente del ATT.

Ahora para derivar la distribución asintótica: Sabemos que el modelo es $C_i = \alpha + \tau D_i + \epsilon_i$ y sabemos que:

$$\beta \xrightarrow{a} N(\beta, \frac{1}{n}\sigma^1 Q_{xx}^{-1})$$

Donde β es el vector de estimadores del modelo de regresión (α, τ). Entonces, sabiendo la distribución de β puedo saber la distribución de τ usando el teorema y usando un vector C tal que la primera entrada sea cero y la segunda 1, de manera que:

$$\hat{\tau_{MCO}} \xrightarrow{a} N(\tau, \frac{1}{n}\sigma^2 Q_{xx}^{-1})$$

Ahora bien, el término $\sigma^2 Q_{xx}^{-}1$ no es observable, pero sé que $\hat{\sigma}^2$ es estimador consistente de σ y $(\frac{D_i' D_i}{n})^{-1}$ es estimador consistente de Q_{xx}^{-1}, y tambíen sé que para la varianza de τ solo necesito la entrada (2,2) de la matriz $(\frac{D_i' D_i}{n})^{-1}$.
La entrada (2,2) de $(\frac{D_i' D_i}{n})^{-1}$ es :

$$\frac{\sum D_i' D_i}{n}$$

Dado que el vector de D_i es de 1s y 0s, por lo que $\sum D_i' D_i$, da como resultado simplemente la cantidad de 1s en la muestra, es decir la cantidad de tratados. Ahora eso dividido n, me da la probabilidad de ser tratado: $p = \mathbb{P}(D_i = 1)$.
Entonces, puedo reescribir la distribución de tau así:

$$\hat{\tau_{MCO}} \xrightarrow{a} N(\tau, \frac{1}{n}\hat{\sigma}^2 p)$$

Contentos por los reveladores hallazgos de los incisos $a)$ y $b)$, ustedes van por un tinto a la cafetería. En ella, encuentran dos colegas debatiendo acerca de cómo estimar correctamente el ATT. Uno de ellos argumenta que, debido a que hay aleatorización, el estimador debe ser una diferencia "ingenua" de medias de la variable dependiente entre los hogares participantes y aquellos que, aunque elegibles, no accedieron al programa. Su otro compañero lo contradice, pues afirma que la manera correcta de hacerlo es estimando τ del modelo (1) por MCO dadas las conclusiones del inciso $a)$ y $b)$. ¿Quién tiene la razón?

Queriendo resolver esta encrucijada, a ustedes se les ocurre una idea: si logran probar que ambos estimadores son numéricamente equivalentes, entonces no debería importar cuál procedimiento utilicen.

c) Prueben que la diferencia ingenua de medias es numéricamente equivalente al estimador de MCO del parámetro τ del modelo (1).

Ayuda: Considere el modelo vectorial

$$\boldsymbol{C} = \boldsymbol{X}\beta + \boldsymbol{\epsilon}$$

donde

$$\boldsymbol{X} = \begin{pmatrix} 1 & D_1 \\ 1 & D_2 \\ \vdots & \vdots \\ 1 & D_n \end{pmatrix}; \qquad \boldsymbol{C} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix}; \qquad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}; \qquad \beta = \begin{pmatrix} \alpha \\ \tau \end{pmatrix}$$

Puede usar sin probar que

$$(\boldsymbol{X}'\boldsymbol{X})^{-1} = \frac{1}{\left(\sum_{i=1}^{n} D_i\right)\left(\sum_{i=1}^{n}(1-D_i)\right)} \begin{pmatrix} \sum_{i=1}^{n} D_i & -\sum_{i=1}^{n} D_i \\ -\sum_{i=1}^{n} D_i & n \end{pmatrix}$$

y que

$$\boldsymbol{X}'\boldsymbol{C} = \begin{pmatrix} \sum_{i=1}^{n} C_i \\ \sum_{i=1}^{n} D_i C_i \end{pmatrix}$$

- Solución:
  Sé que:

$$\hat{\beta}_{MCO} = (X'X)^{-1}X'C$$

Por lo que operando las matrices y sacando la entrada (2,1) de la matríz resultante tengo que:

$$\hat{\tau}_{MCO} = \frac{n\sum D_i C_i - \sum D_i \sum C_i}{\sum D_i(\sum(1-D_i))}$$

Ahora bien, sé que cuando el modelo de regresión tiene una matriz X de tamaño (N x 2) y tiene una columna de 1s y una columna de Dummys, el estimador que acompañar la Dummy es la diferencia de medias simple:

$$C_i = \alpha + \tau D_i + \epsilon_i$$

$$\hat{\tau}_{MCO} = \bar{C}_1 - \bar{C}_0$$

Entonces, noto que:

$$\bar{C}_i/D_i = 1 = \frac{\sum C_i D_i}{\sum D_i} = \frac{\sum C_i}{\sum 1} = \frac{\sum C_1}{n}$$

$$\bar{C}_i/D_i = 0 = \frac{\sum C_i(1-D_i)}{\sum(1-D_i)} = \frac{\sum C_i}{\sum 1 - 0} = \frac{\sum C_0}{n}$$

Ahora, defino la diferencia ingenua de media como:

$$\hat{\tau}_{ing} = \bar{C}_i/D_i = 1 - \bar{C}_i/D_i = 0$$

Ahora reemplazando, tengo que:

$$\hat{\tau}_{ing} = \frac{\sum C_i D_i}{\sum D_i} - \frac{\sum C_i(1-D_i)}{\sum(1-D_i)}$$

$$= \frac{\sum(1-D_i)\sum D_i C_i - \sum D_i \sum C_i(1-D_i)}{\sum D_i(\sum(1-D_i))}$$

Operando solo el numerador, tengo que puedo cancelar las $\sum D_i$ cuando $D_i = 0$ y se que $\sum 1 = n$:

$$(n - \cancel{\sum D_i})\sum C_i D_i - \sum D_i(\sum C_i(1 - \cancel{D_i}))$$

Con lo que llego a la expresión:

$$\hat{\tau}_{ing} = \frac{n\sum D_i C_i - \sum D_i \sum C_i}{\sum D_i(\sum(1-D_i))} = \hat{\tau}_{MCO}$$

Se demuestra entonces que $\hat{\tau}_{ing} = \hat{\tau}_{MCO}$

Ahora una compañera le comenta que, según lo que ha leído, ella cree que el éxito del programa depende en gran medida de la calidad de la tierra donde se siembra. De manera que, pueden existir posibles *efectos heterogéneos* dependiendo de las dotaciones de este factor en cada hogar. Asimismo, le comenta que afortunadamente se cuenta con una variable Z_i en la base de datos que captura esta información. Por lo tanto, ella propone que un modelo más completo estaría dado por

$$C_i = \beta_0 + \beta_1 D_i + \beta_2 \tilde{Z}_i + \beta_3 \, D_i \tilde{Z}_i + \epsilon_i \tag{2}$$

donde $\tilde{Z}_i = (Z_i - \bar{Z})$ y $\bar{Z}$ es el promedio muestral. Finalmente, suponga que $\mathbb{E}[\epsilon_i(1)|Z_i] = 0$ y $\mathbb{E}[\epsilon_i(0)|Z_i] = 0$, esto es, que la calidad de la tierra es exógena.

d) Partiendo del modelo (2):

i) Calcule el efecto esperado en los tratados como función de Z:

$$ATT(Z) = \mathbb{E}[C_i(1) - C_i(0)|D_i = 1, Z]$$

¿Cuál es la interpretación de β_1 y β_3?

- Solución:

$$ATT(Z) = E[C_i(1) - C_i(0)/D_i = 1, Z]$$

Aplicando esperanza condicional:

$$E[C_i(1)/D_i = 1, Z] = \beta_0 + \beta_1 + \beta_2 E[\hat{Z}/D_i = 1] + \beta_3 E[\hat{Z}/D_i = 1] + \cancel{E[\epsilon_i/D_i = 1, Z]}$$

(Se cancela el valor esperado condicional del error por supuesto de exogeneidad), y por la aleatorización se que $E[C_i(0)/D_i = 1, Z] = E[C_i(0)/D_i = 0, Z]$

$$E[C_i(0)/D_i = 1, Z] = \beta_0 + \beta_2 E[\hat{Z}/D_i = 0] + \cancel{E[\epsilon_i/D_i = 0, Z]}$$

Uniendo los términos:

$$ATT(Z) = \cancel{\beta_0} + \beta_1 + \cancel{\beta_2 E[\hat{Z}/D_i = 1]} + \beta_3 E[\hat{Z}/D_i = 1] - \cancel{\beta_0} - \cancel{\beta_2 E[\hat{Z}/D_i = 0]}$$

Llego a que:

$$ATT(Z) = \beta_1 + \beta_3 E[\hat{Z}/D_i = 1]$$

Usando LEI:

$$ATT(Z) = \beta_1 + \beta_3 E[\hat{Z}]$$

En este escenario β_1 se interpreta como el cambio esperado en C_i cuando cambio D_i, dejando todo lo demás constante. En última el β_1 es el efecto del tratamiento. Por otro lado, β_3 se puede entender como las desviaciones del efecto del tratamiento.

ii) Halle el parámetro que ahora captura el ATT del programa.

- Solución: Sabemos que :

$$ATT = \beta_1 + \beta_3 \hat{Z}$$
$$ATT(Z) = \beta_1 + \beta_3 E[\hat{Z}]$$

Sé que por definición: $E[\hat{Z}] = E[Z_i - \bar{Z}] = 0$, por lo que llego a:

$$ATT = \beta_1$$

Entonces, el B_1 captura el ATT en este modelo.

iii) ¿Qué pasaría si Z_i no fuese exógena?

- Solución:
Si Z_i no es exógena entonces $E[\epsilon_i/Z_i] \neq 0$ y por ende el estimador MCO no sería consistente ni insesgado.

Finalmente, su jefe les tiene un último reto: probar si para al menos el 95 % de los hogares participantes el efecto de participar fue positivo.

e) Suponga que $Z_i \sim \mathbb{N}(\mu, 1)$. Usando la forma funcional del modelo (2):

i) Encuentre la distribución de $ATT(Z)$

- Solución:

$$ATT(Z) = \beta_1 + \beta_3 E[\hat{Z}]$$

Dado que $Z_i \sim \mathbb{N}(\mu, 1)$ y que $\hat{Z}_i = (Z_i - \bar{Z})$, sé que:

$$\hat{Z}_i \sim \mathbb{N}(0, 1)$$

Sabiendo la distribución de $\hat{Z}_i$, puedo saber la distribución de $ATT(Z)$:

$$ATT(Z) \sim \mathbb{N}(\hat{\beta}_1, \hat{\beta}_3^2)$$

ii) Encuentre el percentil 5 ($p_{0.05}$) de la distribución de $ATT(Z)$ en términos de los parámetros del modelo.

- Solución:

$$P(ATT(Z) \leq p_{0.05}) = 0.05$$

Estandarizando:

$$P(\frac{ATT(Z) - \beta_1}{\beta_3} \leq \frac{p_{0.05 - \beta_1}}{\beta_3}) = 0.05$$

Sé que:

$$\frac{ATT(Z) - \beta_1}{\beta_3} \sim N(0, 1)$$

Cómo es una normal estándar, el punto critico de la distribución en el percentil 5 es 1,64:

$$P(1,64 \leq \frac{p_{0.05 - \beta_1}}{\beta_3}) = 0.05$$

Por último, despejando tengo que:

$$\hat{p}_{0.05} = 1,64\hat{\beta}_3 + \hat{\beta}_1$$

iii) Suponga que usted estimó por MCO que

$$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 2 \\ 0.3 \\ -0.03 \end{pmatrix}; \qquad S = \begin{pmatrix} 0.6 & -0.26 & 0.33 & 0 \\ -0.26 & 0.7 & 0.5 & 0 \\ 0.33 & 0.5 & 0.35 & -0.25 \\ 0 & 0 & -0.25 & 0.08 \end{pmatrix}$$

donde S es la matriz de varianza-covarianza estimada de los parámetros.

Construya un estimador consistente de $p_{0.05}$ y diseñe un test que le permita probar:

$$\begin{cases} H_0 : p_{0.05} \leq 0 \\ H_a : p_{0.05} > 0 \end{cases}$$

Use los datos disponibles para ejecutar su test. Concluya para un nivel de signifancia $\alpha = 0.05$.

- Solución:

$$\hat{p}_{0.05} = 1,64\hat{\beta}_3 + \hat{\beta}_1$$

Puedo saber la distribución de $\hat{p}_{0.05}$:

$$\hat{p}_{0.05} \sim N(1,64\hat{\beta}_3 + \hat{\beta}_1, (1,64)^2 var(\hat{\beta}_3) + var(\hat{\beta}_1) + 2(1,64)cov(\hat{\beta}_1, \hat{\beta}_3))$$

De ahí puedo tener un estadístico teórico Z_c:

$$Z_c = \frac{\hat{p}_{0.05} - p_{0.05}}{\sqrt{var(\hat{p}_{0.05})}} = \frac{1,64\hat{\beta}_3 + \hat{\beta}_1 - (1,64\beta_3 + \beta_1)}{\sqrt{(1,64)^2 var(\hat{\beta}_3) + var(\hat{\beta}_1) + 2(1,64)cov(\hat{\beta}_1, \hat{\beta}_3))}}$$

De lo anterior, se tiene que $Z_c \sim N(0, 1)$ dado que está estandarizado. Con esto puedo construir el estadístico de prueba t_c:

$$t_c = \frac{\hat{p}_{0.05} - p_{0.05}}{\sqrt{var(\hat{p}_{0.05})}} = \frac{(1,64)(-0.03) + 2 - (1,64\beta_3 + \beta_1)}{\sqrt{(1,64)^2(0,08) + (0,7) + 2(1,64)0}}$$

Bajo H_0 se tiene que: $\beta_1 = \beta_3 = 0$, por lo que la prueba sería:

$$t_c = \frac{(1,64)(-0.03) + 2}{\sqrt{(1,64)^2(0,08) + (0,7)}} = 2,039$$

A un nivel de significancia del 0.05, se rechaza la hipótesis nula en favor de la alterna.

iv) Verdadero o falso: Si el p-valor del test es mayor a 0.05, entonces existe evidencia estadística de que el efecto de que el hogar en el percentil 5 de la distribución de efectos fue negativo o nulo. Justifique.

Segundo ejercicio

Una curva de aprendizaje se puede pensar como el cambio en la productividad o eficiencia con la que se hacen las cosas en el transcurso del tiempo. Este concepto es utilizado en Economía, por ejemplo, para modelar el cambio de los costos reales de las firmas según el tiempo que llevan operando. En particular, se supone que las firmas van aprendiendo a producir de manera más eficiente, por lo que el costo real de producción debería, *ceteris paribus*, decrecer.

Supongan que la curva generalizada de aprendizaje de una firma hipotética está dada por

$$C_i = C_0 N_i^{\frac{\alpha}{\gamma}} Y_i^{\frac{1-\alpha}{\gamma}} \exp(u_i), \tag{1}$$

donde C_i corresponde a los costos reales unitarios que enfrenta una firma i; Y_i es su nivel de producción; N_i es la producción acumulada a lo largo del tiempo; C_0 corresponde a una medida de costos iniciales; y u_i es un término estocástico de media cero desconocido para el econometrista. El parámetro α determina la dirección de la elasticidad del costo unitario con respecto a la producción acumulada. Este es el principal parámetro de interés. Finalmente, el parámetro $\gamma \in \mathbb{R}^+$ caracteriza los retornos a escala de la función de aprendizaje: si $\gamma = 1$ la curva tiene retornos constantes a escala, si $\gamma < 1$ la curva tiene retornos decrecientes a escala, y si $\gamma > 1$ la curva tiene retornos crecientes a escala.[1]

Usted, como investigador, está interesado en estimar la curva de aprendizaje que enfrentan las firmas de la industria manufacturera colombiana. Para ello, cuenta con información de C_i, N_i y Y_i para N firmas de la industria.

a) Propongan un modelo de regresión lineal que capture la forma funcional dada en (1). Muestre cómo los parámetros de su nuevo modelo dependen de los parámetros de la curva de aprendizaje. Mencionen y discutan los supuestos necesarios para que los estimadores por MCO de los parámetros del modelo sean insesgados y consistentes. ¿Son estos supuestos plausibles en el contexto del problema?

- Solución:

  La ecuación 1 representa la *curva de aprendizaje* de que enfrentan las firmas de la industria manufacturera colombiana. Se observa, que la relación entre los costos reales unitarios que enfrenta una firma i (C_i), respecto a su nivel de producción (Y_i) y la producción acumulada a lo largo del tiempo (N_i) es no lineal. No obstante, por la forma de la curva de aprendizaje, si se aplica el logaritmo natural como transformación matemática a la curva de aprendeziaje original, es posible representar la relación entre las variables de interés del modelo como un modelo de regresión lineal[2] que capture la forma funcional de 1.

  Al aplicar el logaritmo natural a la curva de aprendizaje original se obtiene el siguiente *modelo de regresión lineal*:

$$log(C_i) = \beta_0 + \beta_n log(N_i) + \beta_y log(Y_i) + u_i \tag{3}$$

  Donde, claramente se observa que los parámetros del nuevo modelo dependen de los parámetros de la curva de aprendizaje:

  - $\beta_0 = log(C_0)$

[1] Si la tecnología de producción muestra rendimientos constantes a escala, los costos unitarios reales no deben variar con el nivel de producción. Por el contrario, si los rendimientos son crecientes a escala, los costos unitarios deberían disminuir a medida que aumenta el nivel de producción, y si son decrecientes, se esperaría lo contrario.

[2] Básicamente que se pueda representar la curva de aprendizaje como una relación matemática lineal en los parámetros del modelo.

- $\beta_n = \frac{\alpha}{\gamma}$
- $\beta_y = \left(\frac{1-\alpha}{\gamma}\right)$

Ahora bien, el *modelo clásico lineal (MCL)* tiene cuatro supuestos básicos:

1. El modelo es lineal en los parámetros:
$$y_i = x_{i1}\beta_1 + x_{i2}\beta_2 + \ldots + x_{ik}\beta_k + \varepsilon_i$$
2. Exogeneidad de las variables independientes:
$$E[\varepsilon_i|x_{j1}, x_{j2}, \ldots, x_{jk}] = 0 \quad \text{para} \quad j = 1, \ldots, N$$
El cual entre otras cosas es un supuestos clave para la identificación de parámetros en el MCL.
3. **X** es una matriz estocástica y de rango completo:
$$rank(\mathbf{X}) = K$$
EL cual también es un supuesto clave para la identificación de parámetros en el MCL.
4. Homoecesdasticidad y no autocorrelación:
$$\begin{aligned} Var[\varepsilon_i|\mathbf{X}] = \sigma^2 \quad & \text{para} \quad i = 1, \ldots, N \\ Cov[\varepsilon_i, \varepsilon_j|\mathbf{X}] = 0 \quad & \text{para} \quad i \neq j \end{aligned}$$

De los 4 supuestos del MCL mencionados anteriormente, solo los 3 primeros son necesarios para que los estimadores por MCO de los parámetros del modelo sean *insesgados* y *consistentes*. El *supuesto de linealidad en los parámetros* es necesario para poder escribir el modelo en términos de operadores lineales representados por las matrices que hacen parte de la formulación del MCL y del estimador OLS. El *supuesto de exogeneidad* es necesario para garantizar que $E[\varepsilon|\mathbf{X}] = 0$, lo que garantiza que $E[\hat{\beta}_{MCO}] = \beta$. Finalmente, el *supuesto de rango completo* es necesario para que $\mathbf{X}'\mathbf{X}$ sea invertible.

Nota: Importante mencionar que en ningún momento se usa el supuestos de homocedasticidad ni correlación serial para garantizar insesgamiento o consistencia en el estimador MCO.

Los 3 supuestos necesarios para que el estimador MCO sea insesgado y consistente son supuestos plausibles en el contexto del problema:

- *El supuestos de linealidad en los parámetros*: es plausible en la medida de que una curva de aprendizaje tipo *Cobb-Douglass* como la planteada en la ecuación 1 captura los elementos más importantes en nivel de producción (Y_i) y producción acumulada a lo largo del tiempo (N_i) para explicar el costo real unitario (C_i). De igual forma, también tiene sentido introducir el error de forma $exp(u_i)$ dado que es un reescalamiento útil de un término estocástico u_i de media cero y desconocido para el investigador. Como ya se explico antes, mediante una transformación matemática logarítmica es posible escribir la curva de costo original 1 como un modelo de regresión lineal en los parámetros como el que muestra el modelo 3.
- *El supuestos de exogeneidad*: también es posible en la medida que las variables más relevante a la hora de explicar el costo real unitario (C_i) posiblemente son el nivel de producción (Y_i) y la producción acumulada a lo largo del tiempo (N_i). Si la información relevante para explicar C_i se encuentran principalmente en Y_i y en N_i, es muy factible que cualquier componente idiosincrático no observable que se encuentre almacenado en u_i sea exógeno o no tenga relaciones lineal o no lineal con Y_i y N_i, es decir se satisface que $E[\varepsilon_i|Y_i, N_i] = 0$.
- *El supuestos de rango completo de **X***: también es un supuesto muy plausible en la medida que no es factible realmente que haya un problema de multicolinealidad perfecta en el modelo de regresión lineal 3, dado que no es factible que haya una combinación lineal perfecta entre los regresores Y_i y N_i, lo que implica que las columnas de la matriz **X** son linealmente independientes.

No obstante, se resalta que el supuesto de *homocedasticidad y no autocorrelación* es muy poco plausible que se cumpla dado que se espera que hayan elementos idiosincráticos no observables que hagan parte del componente u_i de la curva de aprendizaje de cada firma que afecten la dispersión o varianza de dicho término u_i. Por tanto, es muy factible que haya heterocedasticidad en la muestra de corte transversal de las firmas de la industria manufacturera colombiana y el supuesto de homocedasticidad y no autocorrelación no se satisfaga. No obstante, como dicho supuestos no se necesita para que el estimador de MCO sea *insesgado* y *consistente*, se tiene que si se estima la ecuación 3 por MCO los parámetros $\beta_0 = log(C_0)$, $\beta_n = \frac{\alpha}{\gamma}$ y $\beta_y = \left(\frac{1-\alpha}{\gamma}\right)$ van a ser insesgados y consistentes.

b) Supongan que se cumplen los supuestos que ustedes discutieron en el inciso anterior. Propongan un estimador consistente de α y γ.

- Solución:

  Lo primero que hay que notar es que el modelo que se va a estimar es el *modelo de regresión lineal* planteado en la ecuación 3. Los parámetros de interés que van a ser estimados en este modelo son β_n y β_y. De la discusión anterior, se sabe que si estima la ecuación 3 por MCO los estimadores que se obtengan para β_n y β_y son insesgados y consistentes, dado que, como se menciono en el literal anterior, es muy plausible que los 3 supuestos de: 1) linealidad en los parámetros, 2) exogeneidad $E[u_i|Y_i, N_i] = 0$ y 3) rango completo de **X** se satisfagan para el modelo de regresión lineal planteado en 3 y las características del problema que se está investigando. Entonces, al estimar 3 por MCO, se tiene que por consistencia de dicho estimador:

$$\text{plim}(\hat{\beta}_n) = \beta_n$$
$$\text{plim}(\hat{\beta}_y) = \beta_y$$

  Ahora bien, se busca *estimadores consistentes* para los parámetros α y γ que hacen parte de la curva de aprendizaje original de la ecuación 1.

  Para ello, va a ser necesario emplear el *teorema de mapeo continuo o teorema de Slutsky* que dice que:

  Sea $\mathbf{g} : R^K \to R^J$ una función continua. Sea $\{\mathbf{x}_N : N = 1, 2, \ldots\}$ una secuencia de vectores aleatorios de $K \times 1$ tal que $\mathbf{x_N} \xrightarrow{p} \mathbf{c}$, entonces $\mathbf{g}(\mathbf{x}_N) \xrightarrow{p} \mathbf{g}(\mathbf{c})$

  Como el modelo que se va a estimar es el modelo de regresión lineal 3, se van a obtener los estimadores MCO de $\hat{\beta}_n$ y $\hat{\beta}_y$.

  Se sabe que:

$$\beta_n = \frac{\alpha}{\gamma}$$
$$\beta_y = \left(\frac{1-\alpha}{\gamma}\right)$$

  Entonces, para recuperar los parámetros α y γ, es necesario despejarlos de las dos ecuaciones anteriores.

  EL parámetro γ se puede escribir como $\gamma = \frac{\alpha}{\beta_n}$

  Lo que hace que,

$$\beta_y = \frac{1-\alpha}{\gamma}$$
$$\beta_y = \frac{1}{\gamma} - \frac{\alpha}{\gamma}$$
$$\beta_y = \frac{1}{\left(\frac{\alpha}{\beta_n}\right)} - \beta_n$$
$$\beta_y = \frac{\beta_n}{\alpha} - \beta_n$$
$$\frac{\beta_n}{\alpha} = \beta_n + \beta_y$$
$$\alpha = \frac{\beta_n}{\beta_n + \beta_y}$$

  y

$$\gamma = \frac{\alpha}{\beta_n}$$
$$\gamma = \left(\frac{1}{\beta_n}\right)\left(\frac{\beta_n}{\beta_n + \beta_y}\right)$$
$$\gamma = \frac{1}{\beta_n + \beta_y}$$

Se proponen como estimadores consistentes de α y γ:

$$\hat{\alpha} = \frac{\hat{\beta}_n}{\hat{\beta}_n + \hat{\beta}_y}$$
$$\hat{\gamma} = \frac{1}{\hat{\beta}_n + \hat{\beta}_y}$$

Si se define la función g como:

$$\begin{aligned} \mathbf{g}: \quad & \mathbb{R}^2 \quad \to \quad \mathbb{R}^2 \\ & \begin{pmatrix} \beta_n \\ \beta_y \end{pmatrix} \to \mathbf{g}\begin{pmatrix} \beta_n \\ \beta_y \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \frac{\beta_n}{\beta_n+\beta_y} \\ \frac{1}{\beta_n+\beta_y} \end{pmatrix} \end{aligned}$$

Ahora bien, si se aplica el *teorema de Slutsky* se tiene que:

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \mathbf{g}\begin{pmatrix} \hat{\beta}_n \\ \hat{\beta}_y \end{pmatrix} \xrightarrow{p} \mathbf{g}\begin{pmatrix} \beta_n \\ \beta_y \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{4}$$

Dado que, por consistencia del estimador de MCO al estimar la regresión lineal 3, se sabe que:

$$\begin{pmatrix} \hat{\beta}_n \\ \hat{\beta}_y \end{pmatrix} \xrightarrow{p} \begin{pmatrix} \beta_n \\ \beta_y \end{pmatrix}$$

Lo cual hace que $\hat{\alpha}$ y $\hat{\gamma}$ sean estimadores consistentes de α y γ.

c) Propongan estimadores de las desviaciones estándar de $\widehat{\alpha}$ y $\widehat{\gamma}$. Para esto, supongan que la matriz de varianzas y covarianzas asintótica de $\sqrt{N}(\widehat{\beta}-\beta)$ y su respectivo estimador están dados por

$$V = Avar(\sqrt{N}(\widehat{\beta}-\beta)) = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 \end{pmatrix} \quad \text{y} \quad \hat{V} = \widehat{Avar}(\sqrt{N}(\widehat{\beta}-\beta)) = \begin{pmatrix} \widehat{\sigma}_1^2 & \widehat{\sigma}_{12} & \widehat{\sigma}_{13} \\ \widehat{\sigma}_{12} & \widehat{\sigma}_2^2 & \widehat{\sigma}_{23} \\ \widehat{\sigma}_{13} & \widehat{\sigma}_{23} & \widehat{\sigma}_3^2 \end{pmatrix} \tag{5}$$

respectivamente. Asegúrense de que sus expresiones estén en términos de $\widehat{\beta}_1$, $\widehat{\beta}_2$ y los elementos que componen $\hat{V}$.

Pista: El método delta puede resultar útil.

- Solución:
Podemos usar el *método delta* y el *teorema de Slutsky* para obtener la *matriz var-cov* y de ahí tener estimadores de las desviaciones estándar.En particular sabemos que una expresión de la forma:

$$J[\hat{V}]J^T$$

Donde J es la matriz de derivadas parciales[3] de los estimadores de α y γ, da la matriz var-cov que necesitamos[4].

Definimos J:

$$J = \begin{pmatrix} \frac{\partial \alpha}{\partial \widehat{\beta_0}} & \frac{\partial \alpha}{\partial \widehat{\beta_1}} & \frac{\partial \alpha}{\partial \widehat{\beta_2}} \\ \frac{\partial \gamma}{\partial \widehat{\beta_0}} & \frac{\partial \gamma}{\partial \widehat{\beta_1}} & \frac{\partial \gamma}{\partial \widehat{\beta_2}} \end{pmatrix} = \begin{pmatrix} 0 & \frac{\widehat{\beta_2}}{(\widehat{\beta_1}+\widehat{\beta_2})^2} & \frac{-\widehat{\beta_1}}{(\widehat{\beta_1}+\widehat{\beta_2})^2} \\ 0 & \frac{-1}{(\widehat{\beta_1}+\widehat{\beta_2})^2} & \frac{-1}{(\widehat{\beta_1}+\widehat{\beta_2})^2} \end{pmatrix}$$

Y ya sabemos que $\hat{V}$ es:

$$\hat{V} = \begin{pmatrix} \widehat{\sigma}_1^2 & \widehat{\sigma}_{12} & \widehat{\sigma}_{13} \\ \widehat{\sigma}_{12} & \widehat{\sigma}_2^2 & \widehat{\sigma}_{23} \\ \widehat{\sigma}_{13} & \widehat{\sigma}_{23} & \widehat{\sigma}_3^2 \end{pmatrix}$$

[3]También conocida en la literatura como matriz Jacobiana
[4]Por simplicidad de notación, $\beta_n = \beta_1$ y $\beta_y =_2$

Entonces aplicando Sltusky tenemos:

$$\begin{pmatrix} 0 & \frac{\widehat{\beta_2}}{(\widehat{\beta_1}+\widehat{\beta_2})^2} & \frac{-\widehat{\beta_1}}{(\widehat{\beta_1}+\widehat{\beta_2})^2} \\ 0 & \frac{-1}{(\widehat{\beta_1}+\widehat{\beta_2})^2} & \frac{-1}{(\widehat{\beta_1}+\widehat{\beta_2})^2} \end{pmatrix} \begin{pmatrix} \widehat{\sigma}_1^2 & \widehat{\sigma}_{12} & \widehat{\sigma}_{13} \\ \widehat{\sigma}_{12} & \widehat{\sigma}_2^2 & \widehat{\sigma}_{23} \\ \widehat{\sigma}_{13} & \widehat{\sigma}_{23} & \widehat{\sigma}_3^2 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ \frac{\widehat{\beta_2}}{(\widehat{\beta_1}+\widehat{\beta_2})^2} & \frac{-1}{(\widehat{\beta_1}+\widehat{\beta_2})^2} \\ \frac{-\widehat{\beta_1}}{(\widehat{\beta_1}+\widehat{\beta_2})^2} & \frac{-1}{(\widehat{\beta_1}+\widehat{\beta_2})^2} \end{pmatrix}$$

Finalmente, operando las matrices se llega a:

$$\begin{pmatrix} \frac{\widehat{\beta_2}}{\widehat{\phi}}[\frac{\widehat{\sigma}_2^2\widehat{\beta_2}-\widehat{\sigma}_{23}\widehat{\beta_1}}{\widehat{\phi}}] - \frac{\widehat{\beta_1}}{\widehat{\phi}}[\frac{\widehat{\sigma}_{23}\widehat{\beta_2}-\widehat{\sigma}_3^2\widehat{\beta_1}}{\widehat{\phi}}] & -\frac{1}{\widehat{\phi}}[\frac{\widehat{\sigma}_2^2\widehat{\beta_2}-\widehat{\sigma}_{23}\widehat{\beta_1}}{\widehat{\phi}}] - \frac{1}{\widehat{\phi}}[\frac{\widehat{\sigma}_{23}\widehat{\beta_2}-\widehat{\sigma}_3^2\widehat{\beta_1}}{\widehat{\phi}}] \\ \frac{\widehat{\beta_2}}{\widehat{\phi}}[\frac{-\widehat{\sigma}_2^2-\widehat{\sigma}_{23}}{\widehat{\phi}}] + \frac{\widehat{\beta_1}}{\widehat{\phi}}[\frac{\widehat{\sigma}_3^2+\widehat{\sigma}_{23}}{\widehat{\phi}}] & \frac{1}{\widehat{\phi}}[\frac{\widehat{\sigma}_2^2+\widehat{\sigma}_{23}}{\widehat{\phi}}] + \frac{1}{\widehat{\phi}}[\frac{\widehat{\sigma}_3^2+\widehat{\sigma}_{23}}{\widehat{\phi}}] \end{pmatrix}$$

Donde $\widehat{\phi} = (\widehat{\beta_1} + \widehat{\beta_2})^2$

Entonces, los estimadores de las desviaciones estándar $\widehat{\alpha}$ y $\widehat{\gamma}$ serán:

$$\widehat{DE_{\widehat{\alpha}}} = \sqrt{\frac{\widehat{\beta_2}}{\widehat{\phi}}[\frac{\widehat{\sigma}_2^2\widehat{\beta_2} - \widehat{\sigma}_{23}\widehat{\beta_1}}{\widehat{\phi}}] - \frac{\widehat{\beta_1}}{\widehat{\phi}}[\frac{\widehat{\sigma}_{23}\widehat{\beta_2} - \widehat{\sigma}_3^2\widehat{\beta_1}}{\widehat{\phi}}]}$$

$$\widehat{DE_{\widehat{\gamma}}} = \sqrt{\frac{1}{\widehat{\phi}}[\frac{\widehat{\sigma}_2^2 + \widehat{\sigma}_{23}}{\widehat{\phi}}] + \frac{1}{\widehat{\phi}}[\frac{\widehat{\sigma}_3^2 + \widehat{\sigma}_{23}}{\widehat{\phi}}]}$$

Usted cuenta con la base de datos "manufacturaCol.dta" para implementar la estimación de los parámetros propuesta en los incisos anteriores. La base de datos cuenta con información de las siguientes variables para cada una de las 11,500 firmas de la muestra:

- *costos:* Costos unitarios reales de producción que enfrenta la firma en el momento que se levantaron los datos medido en miles de millones pesos.
- *producto:* Producción de la firma en miles de millones pesos en el momento que se levantaron los datos.
- *producto_acum:* Producción acumulada histórica de la firma en el momento en que se levantaron los datos medida en miles de millones de pesos .

Resuelvan los siguientes incisos a partir de los datos disponibles en la base de datos.

d) A manera de estadísticas descriptivas, representen mediante un gráfico de dispersión y su respectiva línea de ajuste las siguientes relaciones: 1. La relación de $\log(C_i)$ con $\log(N_i)$ una vez se ha "removido" el efecto de $\log(Y_i)$. 2. La relación de $\log(C_i)$ con $\log(Y_i)$ una vez se ha "removido" el efecto de $\log(N_i)$.

   Ayuda: para lograr esto, apliquen la intuición de *partialling out* o residualización de los modelos de regresión múltiples. Interprete los gráficos de dispersión.

- Solución:

  Para la realización de las gráficas de dispersión y su respectiva línea de ajuste, se empleo el siguiente código:

```
// Taller 1
use "/Users/federicoduenas/Desktop/Econometri a_ Avz_/taller 1/manufacturaCol.dta"

foreach var of varlist producto_acum producto costos {
    gen ln_`var'=ln(`var'+1)
}


// relacion de log_c con log_n controlando por log_y
// usando Partialling out:

reg ln_producto_acum ln_producto

gen aux_1 = 9.256792 + 0.1120002*ln_producto

// para tener la parte de N que no es explicada por Y
gen e_1 = ln_producto_acum - aux_1

// ahora para tener los C_i no explicada por Y_i

```

```
reg ln_costos ln_producto

gen aux_2_1 = -.019997 + .0052894*ln_producto

// los residuales: parte de C_i no explicada por Y_i
gen e_1_2 = ln_costos - aux_2_1

// sactter de la relaci n C_i con N_i una vez se removi  el efecto de Y_i
**#

scatter e_1_2 e_1 || lfit e_1_2 e_1

/*


// ahora si calculo la relacion entre N y C, controlando por Y

reg ln_costos e_1

// Que, por teorema de Waugh-Frisch-Lovell, es equivalente al par metro de log_N de la reg lineal:
reg ln_costos ln_producto_acum ln_producto

// gr fica de la relaci n de C y N:
scatter ln_costos e_1  || lfit ln_costos e_1
*/
// ahora la relacion de log_C y log_Y controlando por log_N

reg ln_producto ln_producto_acum

gen aux_2 = .3859203 + .4743226*ln_producto_acum

gen e_2 = ln_producto - aux_2


// ahora remuevo el efecto de N_i sobre C_i

reg ln_costos ln_producto_acum
gen aux_2_2 = -.013807 + .0020865*ln_producto_acum
gen e_2_2 = ln_costos - aux_2_2


// sactter de la relaci n de C y Y removiendo el efecto de N_i

scatter e_2_2 e_2 || lfit e_2_2 e_2

/*
reg ln_costos e_2

reg ln_costos ln_producto_acum ln_producto


scatter ln_costos e_2  || lfit ln_costos e_2
```

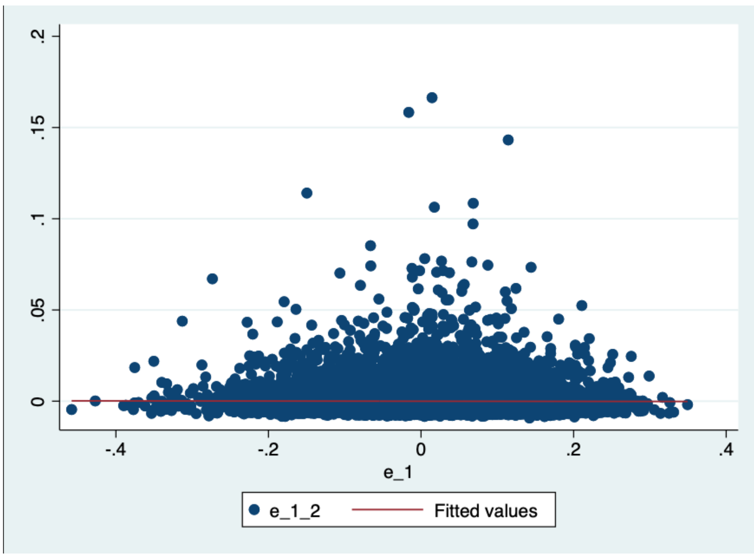


Figura 1: relación de $\log(C_i)$ con $\log(N_i)$ una vez se ha "removido" el efecto de $\log(Y_i)$

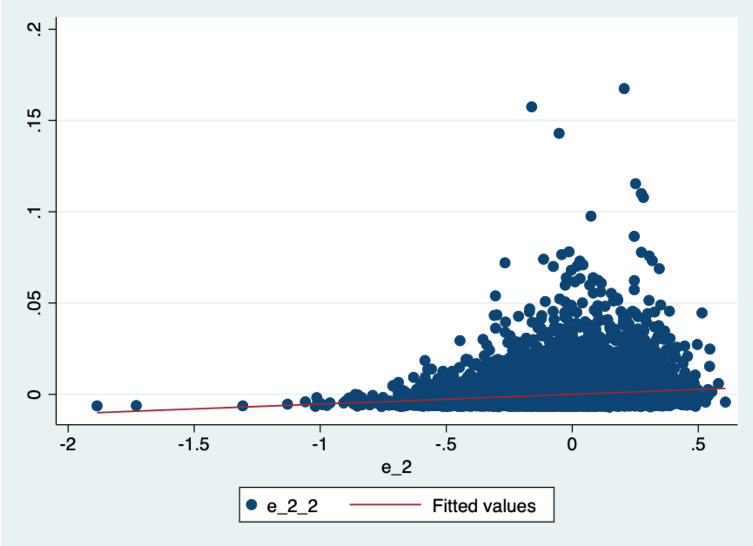


Figura 2: relación de $\log(C_i)$ con $\log(Y_i)$ una vez se ha "removido" el efecto de $\log(N_i)$

De los gráficos de dispersión, se concluye que hay una bajo asociación lineal o correlación entre $log(C_i)$ y $log(N_i)$ que se visualiza claramente por la alta dispersión de la gráfica de dispersión sin un patrón lineal claro y por lo relativamente planto que resulta ser la línea de ajuste asociada. Por otro lado, se observa que hay existe un poco más grad de asoción lineal o correlación entre $log(C_i)$ y $log(Y_i)$ (aunque también se podría decir que es relativamente bajo). No obstante, se observa una mayor patrón lineal en los datos que se puede corroborar con la pendiente positiva (aunque pequeña) de la línea de ajuste.

e) Finalmente, estimen por MCO la ecuación propuesta en el inciso a). A partir de estos resultados, estimen α, γ, y sus respectivos errores estándar siguiendo el planteamiento de los incisos b) y c). Presenten en una tabla los parámetros estimados por MCO. ¿Son estos parámetros consistentes con las gráficas que presentaron en el inciso anterior? Discutan brevemente. Adicionalmente, en una segunda tabla presenten los parámetros α y γ que ustedes estimaron y sus respectivos errores estándar. Interpreten estos últimos. ¿Hay rendimientos crecientes, decrecientes o constantes a escala? ¿Qué tipo de curva de aprendizaje hay en la industria?

- Solución:

Para estimar por MCO el modelo propuesto en 3, se utilizó el siguiente fragmento de código:

```

// estimar la ecuaci n del modelo:

reg ln_costos ln_producto_acum ln_producto
*/
reg ln_costos ln_producto_acum ln_producto


matlist e(b)
matlist e(V)
```

A partir de la estimación de MCO se encuentra que los parámetros estimados son:

$$\hat{\beta}_{MCO} = \begin{pmatrix} \widehat{\beta_0} \\ \widehat{\beta_1} \\ \widehat{\beta_2} \end{pmatrix} = \begin{pmatrix} -0.0158676 \\ -0.0004461 \\ 0.0053394 \end{pmatrix}$$

Se puede afirmar que los estimadores de MCO son consistentes con las gráficas presentadas en el inciso anterior. En primer lugar, el β_1 que es el efecto de $log(N_i)$ sobre $log(C_i)$ manteniendo todo lo demás constante. Se observa en la gráfica que no hay una relación aparente, lo cual es consistente con el valor de β_1 que es cercano a cero. Además, el β_2 que es el efecto de $log(Y_i)$ sobre $log(C_i)$ manteniendo todo lo demás constante, en la gráfica se muestra una relación positiva pero pequeña, lo cual es consistente con el valor del parámetro.

De la estimación se pueden obtener los valores de $\widehat{\alpha}$ y $\widehat{\gamma}$:

$$\begin{pmatrix} \widehat{\alpha} \\ \widehat{\gamma} \end{pmatrix} = \begin{pmatrix} \frac{\widehat{\beta_1}}{\widehat{\beta_1}+\widehat{\beta_2}} \\ \frac{1}{\widehat{\beta_1}+\widehat{\beta_2}} \end{pmatrix} = \begin{pmatrix} \frac{-0.0004461}{-0.0004461+0.0053394} \\ \frac{1}{-0.0004461+0.0053394} \end{pmatrix} = \begin{pmatrix} -0.09116547 \\ 204.36107 \end{pmatrix}$$

Finalmente, se observa que hay rendimiento crecientes a escala dado que se obtuvo un valor de $\gamma = 204.36107 > 1$, lo que por teoría económica implica que hay rendimiento crecientes a escala. Lo anterior, también significa que hay una *curva de aprendizaje* con *rendimientos crecientes* porque γ es mayor a 1[5].

Tercer ejercicio

Aunque las propiedades asintóticas de los estimadores son de gran utilidad para el ejercicio práctico de la estadística, lo cierto es que muchas veces encontrar resultados teóricos en este ámbito es extremadamente complicado. En estas ocasiones, es usual recurrir a otras herramientas, como lo son los procedimientos de Monte Carlo. En breve, los métodos de Monte Carlo aplicados a la estadística buscan explorar las propiedades de los estimadores (insesgamiento, consistencia, eficiencia, suficiencia, etc.) al observar el comportamiento de los mismos en varias muestras aleatorias simuladas. Este ejercicio los guiará a través de una serie de actividades que les permitirán entender algunos métodos clásicos de simulación de muestras aleatorias para, posteriormente, usarlos para evaluar el comportamiento de uno de los estimadores más importantes que se encontrarán en sus carreras: el estimador de MCO.

Suponga que el Ministerio de Educación ha decidido lanzar un programa de preparación y acompañamiento para la presentación de la prueba ICFES a estudiantes que se encuentren en su último año de bachillerato. En particular, suponga que usted conoce que el puntaje ICFES estandarizado potencial $Y_i(\cdot)$ obedece el siguiente modelo:

$$Y_i(0) \sim N(0,1)$$
$$Y_i(1) = Y_i(0) + 3$$

donde $Y_i(1)$ y $Y_i(0)$ son los resultados potenciales del puntaje ICFES en caso de participar y de no participar respectivamente, y donde $Y_i(0)$ tiene una distribución normal estándar.

a) Escriba el puntaje ICFES observado, Y_i, en términos de los resultados potenciales y la exposición al tratamiento, D_i. Suponga que el Ministerio escoge aleatoriamente a los participantes del programa y que todo individuo seleccionado obligatoriamente participa. Proponga un modelo de regresión lineal que le permita estimar el efecto del tratamiento sobre el puntaje, llámese δ.

- Solución:

  Para encontrar el puntaje del ICFES observado (Y_i), en términos de los resultados potenciales y la exposición al tratamiento, D_i, se parte del *modelo causal de Rubin*:

$$Y_i = D_i Y_i(1) + (1 - D_i) Y_i(0)$$
$$Y_i = Y_i(0) + (Y_i(1) - Y_i(0)) D_i$$

  El *modelo de regresión lineal* que permita estimar el efecto del tratamiento sobre el puntaje (δ) es:

$$Y_i = Y_i(0) + (Y_i(1) - Y_i(0)) D_i$$
$$Y_i = E[Y_i(0)] + (Y_i(1) - Y_i(0)) D_i + E[Y_i(0)] - Y_i(0)$$
$$Y_i = \alpha + \delta D_i + \epsilon_i$$

  Donde $\alpha = E[Y_i(0)]$ y $\epsilon_i = E[Y_i(0)] - Y_i(0)$

[5]Como se explica en el inciso, γ es el parámetro que caracteriza los retornos a escala de la función de aprendizaje y si la tecnologıa de producción muestra rendimientos crecientes a escala, los costos unitarios deberıan disminuir a medida que aumenta el nivel de producción lo que se refleja claramente con el $\gamma > 1$

El modelo de regresión lineal $Y_i = \alpha + \delta D_i + \epsilon_i$ permite estimar el efecto del tratamiento sobre el puntaje (δ), dado que al haber aleatorización se elimina el sesgo de selección (SB), lo que hace que la simple diferencia de medios (SDO), que por definición es equivalente al parámetro que acompaña a la variable de asignación a tratamiento D que se estima por MCO, sea igual al efecto del tratamiento sobre el puntaje (δ)[6].

b). Inicialmente, buscaremos validar a través de simulaciones algunas de las propiedades asintóticas del estimador por MCO. En particular, a usted le interesa saber si a medida que aumenta el tamaño de muestra sus estimativos del efecto del programa se tornan más precisos.

- Solución:

En primer lugar, fue necesario definir la semilla para generar reproducibilidad en el código[7]. Posteriormente, se definieron unas funciones auxiliares para calcular la simple diferencia de medias, la matriz de regresores $\mathbf{X}$, el estimador $\hat{\sigma}$, el estimador $\hat{Q}^{-1}_{XX}$ y el estimador para la matriz de *varianzas-covarianzas* para errores robustos *Huber_White* $a\hat{v}ar(\hat{\beta_{mco}}) = (\mathbf{X}^{'}\mathbf{X})^{-1}\left(\sum_{i=1}^{N}\hat{\varepsilon}^2\mathbf{x_i}\mathbf{x_i}^{'}\right)(\mathbf{X}^{'}\mathbf{X})^{-1}$

Posteriormente, se construyó la función *delta*, que es la función más importante del código, dado que es la base para todo lo que se realiza subsecuentemente[8]. La función *delta* está diseñada para calcular entre otras cosas: el *estimador MCO* de δ que seo obtiene al estimar por MCO la ecuación $Y_i = \alpha + \delta D_i + \epsilon_i$, el estimador $\hat{\sigma}$ y el estimador de la varianza $v\hat{a}r(\hat{\delta})$, dependiendo si se uso la matriz de varianzas-covarianzas tradicional o la matriz de varianzas-covarianzas para errores robustos *Huber-White*.

```r
library(tidyverse)

# Definir la semilla para reproducibilidad en los resultados
set.seed(12345)

# Punto 3 ----

# Ejercicio a. ----

# Switching regression:
## y_{i} = y_{i}^{1} * D_{i} + (1 - D_{i}) * y_{i}^{0}
## y_{i} = y_{i}^{0} + (y_{i}^{1} - y_{i}^{0}) * D_{i}
## y_{i} = delta * D_{i} + epsilon_{i}

# Donde el par metro causal:
## delta = E[y_{i}^{1} | D_{i} = 1] - E[y_{i}^{0} | D_{i} = 0]
## delta = 3 - 0
## delta = 3

# Ejercicio b. ----

# Nota: Dado las condiciones para simular y_0 = y_{i}^{0} y y_1 = y_{i}^{2},
#       uno esperar a que el SDO = 3 y por tanto el delta = 3
#       Ademas, dado que el SB = 0 (teniendo en cuenta que y_0 y y_1 son independientes de D)
#       se tiene que delta representa el impcto causal del programa del ministerio

# I. ----

# Definici n de par metros para la simulaci n

## N mero de muestras:
t = 100

## Tama o de muestras:

n10 = 10
n20 = 20
n50 = 50
n100 = 100
n500 = 500
n1000 = 1000

#__________________________
# Funciones auxiliares:
#__________________________

# C lculo del SDO: (SDO := Simple difference of outcomes) (Funci n auxiliar)
SDO_calculo = function(Y, D){
```

[6]Recordar, que como la simple diferencia de medias es $SDO = \delta + SB$, si el sesgo de selección (SB) se elimina con la aleatorizión, entonce la simple diferencia de medias es igual al efecto causal del tratamiento $SDO = \delta$

[7]La semilla es una manera en la que podemos replicar código que presente aleatoriedad o que utilice funciones de distribución simuladas, explotando el hecho de que los algoritmos de aleatorización de los computadores son pseudo-aleatorios. De esta forma, es posible generar reproducibilidad en los resultados y obtener los mismos resultados

[8]Esta función es lo suficientemente general que permite hacer las estimaciones de δ bajo las diferentes especificaciones del problema. Es decir, permite estimar delta bajo una muestra que distribuye normal, bajo una muestra que distribuye Cauchy y bajo una muestra que distribuye normal pero que presenta heterocedasticidad en los errores. Es la función base, para toda la simulación de Monte Carlo que se realizará subsecuentemente.

```r
  # Por construcci n
  ## Y = Y_0 Si D = 0
  ## Y = Y_1 Si D = 1
  # Nota: Y y D son vectores del mismo tama o
  # Inicializaci n de contadores
  count0 = 0
  count1 = 0
  # Inicializaci n de la suma para c lculo del SDO
  suma0 = 0
  suma1 = 0
  # Iteraci n a trav s de la muestra
  for (i in 1:length(Y)){
    if (D[i] == 0){
      suma0 = suma0 + Y[i]
      count0 = count0 + 1
    }else{
      suma1 = suma1 + Y[i]
      count1 = count1 + 1
    }
  }
  SDO = (suma1/count1) -(suma0/count0) # SDO: Simple diferencia de muestras
  return(SDO)
}

# Construcci n matriz X para un modelo de regresi n lineal con constante: (Funci n auxiliar)
ols_X = function(...){
  # ... contiene los regresores necesarios para construir la matriz X
  regresores = list(...)
  # Nota: Cada regresor es un vector con el mismo n mero de observaciones n
  n = length(regresores[[1]]) # No importa que se tome el primer regresor dado que todos los regresores tienen el mismo tama o
  const = rep(1, times = n)
  X = cbind(const, ...)
  return(X)
}

# Construcci n del estimador \Hat{sigma^{2}}: (Funci n auxiliar)
estimador_sigma = function(y, X, beta_est){
  # Variables:
  ## y: es el vector que contiene la variable dependiente
  ## X: es la matriz que contiene las variables regresoras
  ## beta_est: es el vector de los par metros estimados por OLS
  e = y - X %*% beta_est # e es el vector de residuales
  k = ncol(X)  # El n mero de par metros a estimar
  n = length(y)  # El n mero de observaciones en la mustra
  sigma_est = (t(e) %*% e) / (n - k)   # estimador de sigma (varianza del t rmino de error)
  # El resultado de la funci n es: sigma_est (que es un escalar)
  return(sigma_est)
}

# Construcci n del estimador para la matriz \Hat{Q}_{XX}^{-1}: (Funci n auxiliar)
estimador_Q_xx_inv = function(X){
  n = nrow(X) # donde n es el tama o de la muestra
  Q = solve((1/n) * (t(X) %*% X)) # Q es el estimador que se est  buscando
  # Q es una matriz de tama o k x k, donde k es en n mero de par metros a estimar
  # La funci n me retorna a una matriz Q de tama o k x k
  return(Q)
}

# Construcci n del estimador de la  matriz de varianzas y covarianzas
# Huber-White: (Funci n auxiliar)
Huber_White = function(y, X, beta_est){
  # Variables:
  ## y: es el vector que contiene la variable dependiente
  ## X: es la matriz que contiene las variables regresoras
  ## beta_est: es el vector de los par metros estimados por OLS
  e = y - X %*% beta_est # e es el vector de residuales
  k = ncol(X)  # El n mero de par metros a estimar
  n = length(y)  # El n mero de observaciones en la mustra
  # Construcci n de (X^{'}X)^{-1}
  mat_X = solve(t(X) %*% X)
  # Matriz intermedia de los errores robustos White (que se encuentra por medio de una suma)
  # Inicializo la matriz como una matriz de ceros
  mat_intermedia = matrix(0, nrow = k, ncol = k)
  # El for se dise a para llenar la matriz mat_intermedia
  for (i in 1:n){
    mat_intermedia = mat_intermedia + (e[i])^2 * (X[i, ] %*% t(X[i, ]))
  }
  # mat_final me dal estimador de la matriz de varianzas y covarianzas de Huber-White
  mat_final = mat_X %*% mat_intermedia %*% mat_X
  return(mat_final)
}

# y = rnorm(1000)
# D = rnorm(1000, mean = 2)
# beta = c(1, 2)
# X = ols_X(D)
#
# Huber_White(y, X, beta)
```

```r
# l estimador para la matriz \Hat{Q}_{XX}^{-1}: (Funci n auxiliar)

##_________________________________________________________________________
# delta: Es la funci n principal (m s importante) de todo el c digo.
#        Con base en la funci n delta es que se deduce todo lo dem s
#        en el c digo.
#        Es una funci n bastante general que contempla todas las situaciones
#        que puedan surgir en el c digo. En ese caso, contempla la simulaci n
#        de la Cauchy y de una muestra heteroced stico donde la volatilidad
#        de los tratados es diferente a la volatilidad de los no tratados
# Nota:  Revisar toda el enunciado del ejercicio primero para entender mejor
#        la l gica de la funci n
##_________________________________________________________________________

# Funci n para estimar el efecto tratamiento (delta) en una base de datos
delta = function(t, n, distro, hetero = F, varianza_y0 = 1, varianza_y1 = 1, robustos = F){
  # Defino la semilla para reproducibilidad del resultado
  set.seed(5678)
  # Definici n de variables:
  ## t: n mero de muestras a simular
  ## n: tama o de las muestras
  ## distro: tipo de distribuci n con el que se va a simular Y_{i}^{0}
  ## Nota: Si distro == "normal", entonces los par metros opcionales
  ##       empiezan a tomar relevancia
  ## hetero: Para saber si la volatilidad entre tratados y no tratados es diferente
  ## varianza_y0: El valor de la varianza de la distribuci n normal con la que se simula y_{i}^{0}
  ## varianza_y1: El valor de la varianza de la distribuci n normal con la que se simula y_{i}^{1}
  ## robustos: Solo se activa si se escoge la opci n hetero == T y adem s robustos == T
  ##           Permite estimar errores robustos usando la matriz de varianzas y covarianzas Huber-White
  # Consideraciones adicionales:
  # Condici n l gica para saber si y_0 sigue una distribuci n normal est ndar o una distribuci n Cauchy est ndar
  if (distro == "normal"){
    # Hay dos posibles casos en caso de que la distribuci n sea normal o no:
    # 1. Homoced sticidad entre tratados y no tratados
    ## En caso que la volatilidad de y_{i}^{0} y y_{i}^{1} sea la misma hetero == F
    # 2. Heterocedastidad entre tratados y no tratados (diferente volatilidad dependiendo si fueron tratados o no)
    ## En caso que la volatilidad de y_{i}^{0} y y_{i}^{1} sea diferente hetero == T
    if (hetero == F){
      # df: Dataframe que almacena el SDO y el delta_estimado por OLS
      # El par metro delta es el p a r a etro asociado a la asignaci n a tratamiento (D)
      df = data.frame(SDO = double(), delta_est = double(), sigma = double(), var_tlc = double(), var_ols = double())
      # Meter un for para generar las t diferentes muestas
      for (muestra in 1:t){
        # Simulaci n de los outcomes potenciales y de la variable tratamitno
        y_0 = rnorm(n, mean = 0, sd = 1) # Simulaci n del outcome potencial de ausencia de tratamiento
        y_1 = y_0 + 3  # Simulaci n del outcome potencial de presencia de tratamiento
        D = rbinom(n, 1, prob = 0.3) # Simulaci n de una variable Bernoulli con una probabilidad de  xito  de 0.3
        # Modelo causal de Rubin
        y = y_0 + (y_1 - y_0) * D    # y es un vector Nx1, donde n es el n mero de observaciones (tama o de muestra)
        # Corregir (Debo calcular es el SDO)
        SDO = SDO_calculo(y, D) # C lculo del SDO utlilizando la funci n SDO_calculo
        # Estimaci n del par metro delta mediante una regresi n lineal con constante
        # X es una matriz NxK, donde n es el n mero de observaciones y K el n mero de p a r a etros
        X = ols_X(D) # D es la variable tratamiento
        beta_ols = solve(t(X) %*% X) %*% t(X) %*% y # beta_ols es un vector de Kx1, donde K es el n mero de par metos
        # Ahora, se calcularon 3 par metros adicionales:
        # el estimador de sigma, el estimador de var(sqrt(n) (delta_{OLS} - delta) y el estimador de var(delta_{OLS}))
        ## estimador de sigma:
        sigma = estimador_sigma(y, X, beta_ols)
        ## estimador de \Hat{Q}_{XX}^{-1}:
        Q_mat = estimador_Q_xx_inv(X)
        ## estimador de var_tlc = var(sqrt(n) (delta_{OLS} - delta))
        var_tlc = sigma * Q_mat[[2,2]]
        ## estimador de var_ols = var(delta_{OLS})
        n = nrow(X)
        var_ols = 1/n * sigma * Q_mat[[2,2]]
        # Dataframe que contiene todos los par metros de inter s
        df[muestra, ] = c(SDO, beta_ols[[2,1]], sigma, var_tlc, var_ols)    # Extraigo el segundo par metro que es el par metro delta
      }
    }else{
      # df: Dataframe que almacena el SDO y el delta_estimado por OLS
      # El par metro delta es el p a r a etro asociado a la asignaci n a tratamiento (D)
      df = data.frame(SDO = double(), delta_est = double(), sigma = double(), var_ols = double())
      # Meter un for para generar las t diferentes muestas
      for (muestra in 1:t){
        # Simulaci n de los outcomes potenciales y de la variable tratamitno
        D = rbinom(n, 1, prob = 0.3) # Simulaci n de una variable Bernoulli con una probabilidad de  xito  de 0.3
```

```
        y_0 = rnorm(n, mean = 0, sd = sqrt(varianza_y0)) # Simulaci n del outcome potencial de ausencia de tratamiento
        y_1 = rnorm(n, mean = 3, sd = sqrt(varianza_y1))  # Simulaci n del outcome potencial de presencia de tratamiento
        # Modelo causal de Rubin
        y = y_0 + (y_1 - y_0) * D    # y es un vector Nx1, donde n es el n mero de observaciones (tama o de muestra)
        # Corregir (Debo calcular es el SDO)
        SDO = SDO_calculo(y, D) # C lculo del SDO utlilizando la funci n SDO_calculo
        # Estimaci n del par metro delta mediante una regresi n lineal con constante
        # X es una matriz NxK, donde n es el n mero de observaciones y K el n mero de p a r a etros
        X = ols_X(D) # D es la variable tratamiento
        beta_ols = solve(t(X) %*% X) %*% t(X) %*% y # beta_ols es un vector de Kx1, donde K es el n mero de par metos
        if (robustos == F){
          # Ahora, se calcularon 3 par metros adicionales:
          # el estimador de sigma, el estimador de var(sqrt(n) (delta_{OLS} - delta) y el estimador de var(delta_{OLS}))
          ## estimador de sigma:
          sigma = estimador_sigma(y, X, beta_ols)
          ## estimador de \Hat{Q}_{XX}^{-1}:
          Q_mat = estimador_Q_xx_inv(X)
          ## estimador de var_ols = var(delta_{OLS})
          n = nrow(X)
          var_ols = 1/n * sigma * Q_mat[[2,2]]
          # Dataframe que contiene todos los par metros de inter s
          df[muestra, ] = c(SDO, beta_ols[[2,1]], sigma, var_ols)    # Extraigo el segundo par metro que es el par metro delta
        }else{
          # Sigma, es provisional, luego se retira porque no tiene sentido para un estimador con matriz de varianzas y covarianzas
          # con errores robustos Huber-White (Dado la heterocedasticidad de los errores)
          sigma = 0
          # C lculo errores robustos Huber-White (me genera una matriz k x k, donde k es el n mero de par metros)
          var_ols = Huber_White(y, X, beta_ols)[[2,2]] # Extraigo la componente 2 de la matriz de varianzas y covarianzas de Huber-White
          df[muestra, ] = c(SDO, beta_ols[[2,1]], sigma, var_ols)    # Extraigo el segundo par metro que es el par metro delta
        }
      }
      # Condicional final para eliminar la columna sigma si robustos == T
      if (robustos == T){
        df = df %>%
          select(SDO, delta_est, var_ols)
      }
    }
  }else if (distro == "cauchy"){
    # df: Dataframe que almacena el SDO y el delta_estimado por OLS
    # El par metro delta es el p a r a etro asociado a la asignaci n a tratamiento (D)
    df = data.frame(SDO = double(), delta_est = double())
    for (muestra in 1:t){
      # Simulaci n de los outcomes potenciales y de la variable trataminto
      y_0 = rcauchy(n, location = 0, scale = 1) # Simulaci n del outcome potencial de ausencia de tratamiento
      y_1 = y_0 + 3  # Simulaci n del outcome potencial de presencia de tratamiento
      D = rbinom(n, 1, prob = 0.3) # Simulaci n de una variable Bernoulli con una probabilidad de  xito de 0.3
      # Modelo causal de Rubin
      y = y_0 + (y_1 - y_0) * D    # y es un vector Nx1, donde n es el n mero de observaciones (tama o de muestra)
      # Corregir (Debo calcular es el SDO)
      SDO = SDO_calculo(y, D) # C lculo del SDO utlilizando la funci n SDO_calculo
      # Estimaci n del par metro delta mediante una regresi n lineal con constante
      # X es una matriz NxK, donde n es el n mero de observaciones y K el n mero de p a r a etros
      X = ols_X(D) # D es la variable tratamiento
      beta_ols = solve(t(X) %*% X) %*% t(X) %*% y # beta_ols es un vector de Kx1, donde K es el n mero de par metos
      df[muestra, ] = c(SDO, beta_ols[[2,1]])    # Extraigo el segundo par metro que es el par metro delta
    }
  }
  # La funci n delta retorna un df con el c lculo de la SDO
  # y el par metro estimado delta, que proviene de una switching regression
  return(df)
}
```

Luego de tener listas las *funciones auxiliares* y la *función principal* **delta**, se procede a realizar las simulaciones.

I. Simule 100 muestras $\{(Y_i(0), Y_i(1), D_i\}_{i=1}^{n}$ de tamaño $n = 10, 20, ..., 1000$ donde $D_i \sim Bernoulli(0.3)$. Para cada muestra, estime el efecto de la política del Ministerio sobre el puntaje ICFES y almacene su estimado.

- Solución:
  El siguiente fragmento de código permite realizar las simulaciones. En cada simulación, se simulan

100 muestras, en donde lo que cambian son los tamaños empleados[9].

```

# Simulaci n
# delta(t, n10, distro = "normal") # Existe el riesgo de que haya
              # multicolinealidad perfecta cuando se usa n10 = 10
              # Puede generar un vector D = rep(0, times = 10)

delta20 = delta(t, n20, distro = "normal")
delta50 = delta(t, n50, distro = "normal")
delta100 = delta(t, n100, distro = "normal")
delta500 = delta(t, n500, distro = "normal")
delta1000 = delta(t, n1000, distro = "normal")
```

II. Obtenga la media y la varianza muestral de los estimadores almacenados para los distintos tamaños de muestra.

- Solución:
  Para encontrar la media y la varianza muestral de los estimadores se empleo el siguiente fragmento de código:

```

# n_vector es un vector que almacena los diferentes tama os de muestra
n_vector = seq(from = 20, to = 1000, by = 10)

# Funci n que genera un dataframe con el tama o de muestra,
# la media y la varianza del estimador de delta para cada
# tama o diferente de muestra

media_varianza_delta = function(t, n_vector, distro, hetero = F, varianza_y0 = 1,
    varianza_y1 = 1, robustos = F){
  # Variables:
  ## t: N mero de muestras que se van a generar por cada tama o de muestra
  ## n_vector: Variable que almacena los diferentes tama os de muestras
  ## distro, hetero, varianza_y0 y varianza_y1 son par metros definidos para la
      funci n delta
  # df: DataFrame que almacena el tama o de muestra,
  #     la media y la varianza del estimador de delta para cada tama o de muestra
  df = data.frame(tama o = double(), media = double(), varianza = double())
  # Llamo a la funci n delta para cada tama o diferente de muestra
  for (i in 1:length(n_vector)){
    n_muestra = n_vector[i]
    delta_n = delta(t, n_muestra, distro, hetero, varianza_y0, varianza_y1, robustos)$
        delta_est # delta_n es el vector de deltas por cada tama o de muestra
    df[i,] = c(n_muestra, mean(delta_n), var(delta_n))
  }
  return(df)
}

# Dataframe con el tama o de muestra y la media y varianza del
# estimador de delta para los diferentes tama os de muestra
media_varianza = media_varianza_delta(100, n_vector, distro = "normal"); glimpse(media_
    varianza)
```

  Lo que se puede observar es el estimador es insesgado[10] y consistente[11]. Igualmente, se observa que la varianza muestral de δ cae a medida que aumenta el tamaño de las muestras.

III. Haga las siguientes dos gráficas: 1) Grafique el promedio muestral de los estimadores contra el tamaño de muestra. 2) Grafique las varianzas muestrales contra el tamaño de muestra correspondientes. ¿Qué puede decir acerca del comportamiento de las estimaciones a medida que aumenta el tamaño de muestra? ¿Qué propiedades del estimador de MCO se ven reflejadas en el ejercicio?

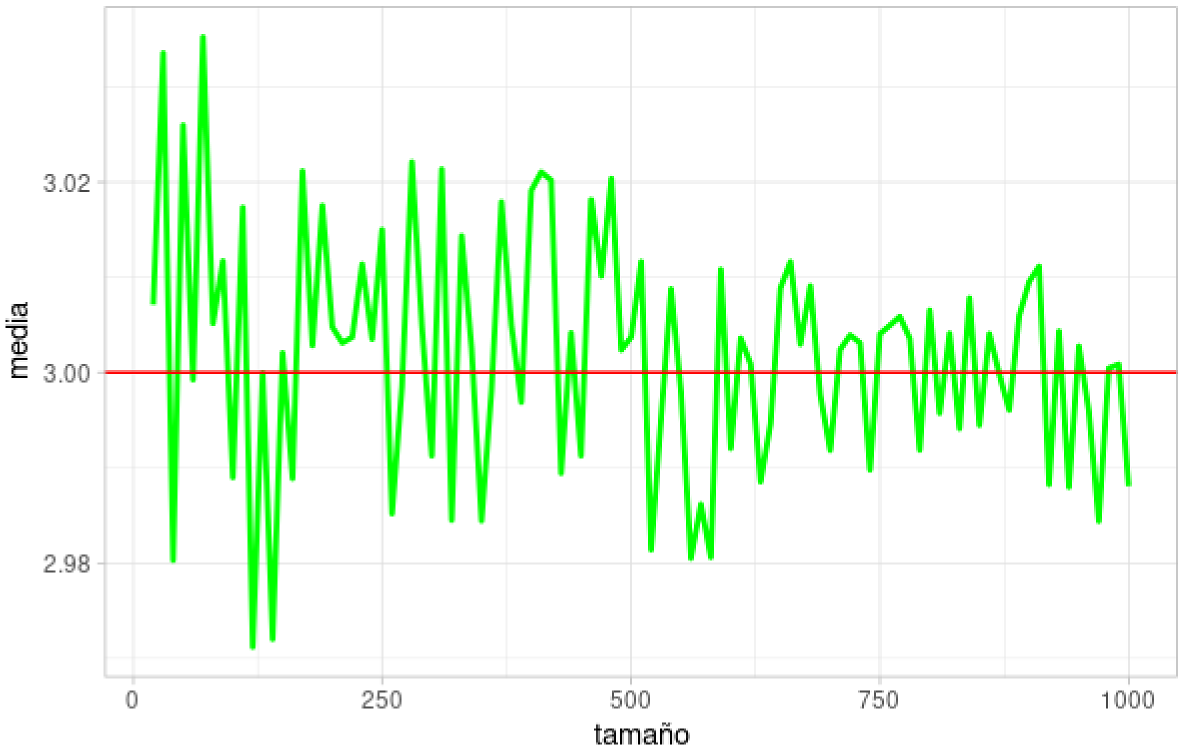
- Solución:
  La gráfica de *promedio muestral* de los estimadores contra el tamaño de muestra:

[9] Se utilizan tamaños de muestras de: 20, 50, 100, 500 y 1000 observaciones. No se utilizan tamaños de muestras de 10 observaciones porque al tener la asignación al tratamiento un parámetro de probabilidad de éxito de $\theta = 0.3$, hace factible generar un vector de la variable tratamiento donde ninguno de las 10 observaciones reciba tratamiento, lo que hace que no haya variabilidad en el regresor y genere multicolinealidad perfecta con la variable que modela la constante del modelo.

[10] Con muestra práticamente de tamaño 20 ya converge al verdadero valor $= 3$

[11] Con muestra muy grandes los estimadores se aproximan muy bien a $= 3$

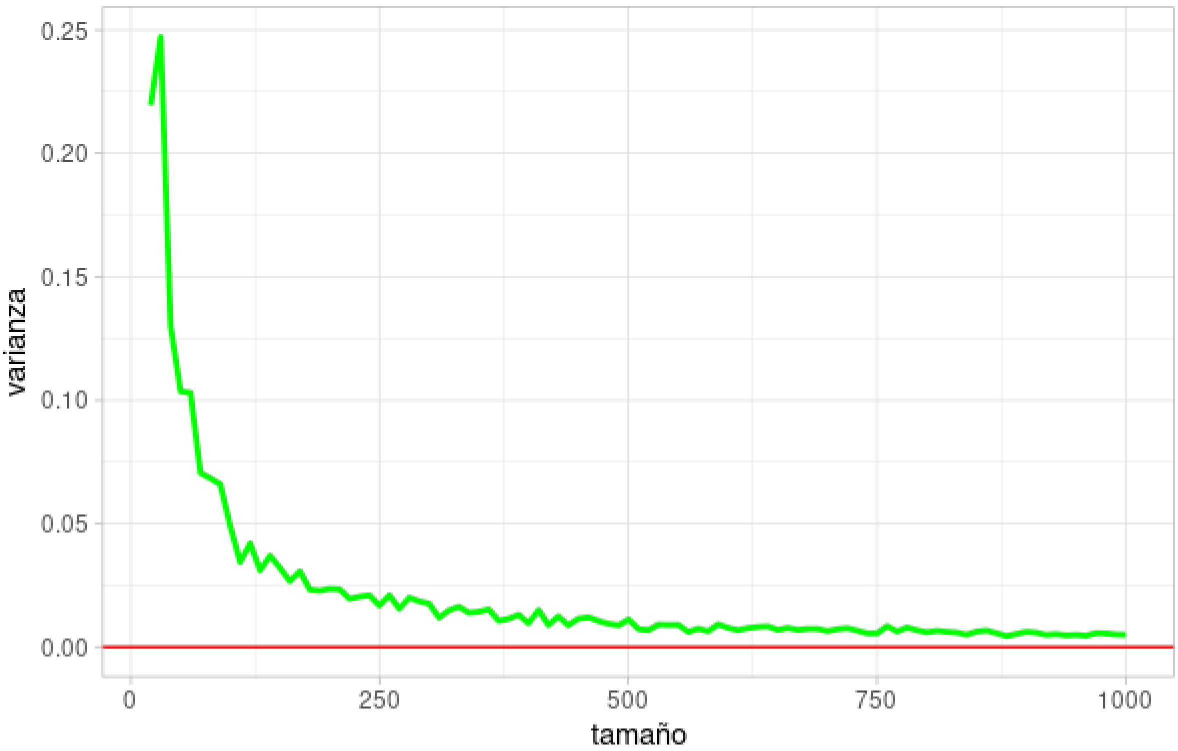
Figura 3: Gráfica promedio muestral de los estimadores contra el tamaño de muestra.



Variables incluidas: media: promedio muestral de los estimadores MCO de δ y tamaño: tamaño de muestra

La gráfica de la varianza muestral de los estimadores contra el tamaño de muestra:

Figura 4: Gráfica varianza muestral de los estimadores MCO de δ contra el tamaño de muestra.



Variables incluidas: varianza: varianza muestral de los estimadores y tamaño: tamaño de muestra

Para construir las dos gráficas: 1) el promedio muestral de los estimadores contra el tamaño de muestra y 2) las varianzas muestrales contra el tamaño de muestra correspondiente se empleo el siguiente fragmento de código:

```

grafica_propiedas = function(df, variable_y, y_intercepto = 0){
  variable_y = ensym(variable_y)
  graph = df %>%
    ggplot(aes(x = tama o, y = !!variable_y)) +
    geom_line(color = "green", size = 1) +
    geom_hline(yintercept = y_intercepto, color = "red") +
    theme_light()
  return(graph)
}

grafica_propiedas(media_varianza, media, y_intercepto = 3)
grafica_propiedas(media_varianza, varianza)
```

Ahora bien, para contexturalizar, se sabe que por las instrucciones dadas, para cada tamaño de muestra empleada, se simularon 100 muestras. Para la generación de las gráficas se construyó una especie de *grid* o rejilla que permitiera generara muestran que iban de tamaño 20 a tamaño 1000, aumentando el tamaño de la muestra de 10 en 10 observaciones, pero siempre generando 100 muestras por cada tamaño de muestra[12]. Con esta metodología, y como se está mantienindo siempre el número de muestras simuladas por tamaño de muestra de $t = 100$, es posible visualizar cuáles son los efectos de aumentar el tamaño de las muestras sobre los estimadores MCO del parámetro de interés δ. Como se puede observar de 3, a medida que n aumenta se puede observar por la serie gráficada que la media muestra de dichos estimadores oscila alrededor del parámetro poblacional conocido $\delta = 3$. Aún más interesante, es observar que desde $n = 20$ la media de dichos estimadores es muy cercana a 3 lo que muestra la propiedad de insesgadez del estimador MCO y que prácticamente a lo largo de la serie se observa que dichas medias muestrales del estimador rara vez exceden los valores de 3.03 o caen poer debajo de de 2.07, lo que muestran nuevamente la alta presicisión del estimador MCO para todos los tamaños de muestra. De igual forma, de la 3 se observa que a medida que el tamaño de la muestra aumenta la volatilidad de la media muestral del estimador MCO disminuye y cada vez dicha media se acerca más y más a $\delta = 3$, lo que permite ver

[12] Es importante notar que en una simulación de Monte Carlo hay dos parámetros clave, un parámetro t que denota el número de muestras que se está simulando y un parámetro n que denota el tamaño de cada muestra

la propiedad asintótica del estimador MCO de consistencia $\operatorname{plim} \delta_{MCO} = \delta = 3$. Finalmente, de la figura 4, claramente se observa que la varianza muestral del estimador MCO δ_{MCO} de δ disminuye a medida que aumenta el tamaño de muestra, donde claramente se ve una varianza muestral de hasta 0.25 en las primeras muestras, que rápidamente a medida que aumenta el tamaño de las muestras (mantiendo el número de muestras simuladas por tamaño de muestra constante en $t = 100$) cae a 0.05 en las muestras de tamaño $n = 100$ y ya por muestras con tamaño por encima a 250 observaciones se observa prácticamente que la varianza del estimador MCO de δ es casi indistinguible de cero, lo cuál claramente muestra que el estimador MCO es más eficiente a medida que aumenta el tamaño de muestra y que la distribución asintótica del estimador MCO de δ tiende a una distribución normal degenerada centrada en el parámetro poblacional $\delta = 3$.

Uno de los propósitos principales de los modelos de regresión lineal es poder estudiar las relaciones que existen entre la variable dependiente y las variables independientes. Más precisamente, nos interesa dilucidar el tamaño y el signo de dicha relación. No obstante, en la práctica, esto es difícil de establecer puesto que desconocemos el proceso generador de los datos y en general, contamos con una sola muestra para nuestra estimación. Así las cosas, si nuestro objetivo es hacer inferencia, debemos preguntarnos qué tan precisas son nuestras estimaciones, esto es, qué tan dependientes son de la muestra particular que tenemos. Para lograrlo, es útil entender a los estimadores como variables aleatorias, que dependen de una muestra, pero que tienen una distribución definida. En particular, es de nuestro interés dicha distribución cuando el tamaño de muestra es grande, pues conocer la distribución exacta para muestras reducidas puede ser complicado.

c) En este inciso vamos a aproximar la distribución asintótica del estimador de MCO. Para ello, realicen las siguientes instrucciones:

   I. Siguiendo los pasos expuestos en el punto b), simule 1000 muestras $\{(Y_i(0), Y_i(1), D_i\}_{i=1}^n$ de tamaño $n = 10, 20, 100, 1000$ donde $D_i \sim Bernoulli(0.3)$. Para cada muestra, estime el efecto del programa del Ministerio y calcule y almacene

$$a_{k,n} = \sqrt{n}(\hat{\delta}_{k,n} - \delta)$$

   donde k indexa las muestras de un determinado tamaño.

- Solución:

  Dado que se va a estudiar las *propiedades asintóticas del estimador MCO* es necesario recurrir a teoría y teoremas de teoría asintótica para realizar dicho análisis. Lo primero, es recurrir al *teorema del limite central* para encontrar la distribución asintótica del estimador δ_{MCO}. Se sabe que:

$$\sqrt{N}\left(\hat{\beta}_{MCO} - \beta\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2 Q_{XX}^{-1}) \tag{6}$$

  Dicha ecuación, implica que la distribución asintótica de $\hat{\beta}_{MCO}$ sea:

$$\hat{\beta}_{MCO} \xrightarrow{d} \mathcal{N}\left(\beta, \sigma^2 Q_{XX}^{-1}\right) \tag{7}$$

  Ahora bien, partiendo de la ecuación 7 se puede obtener las distribución asintótica del estimador $\hat{\delta}_{MCO}$.
  Como lo muestra la gráfica 5, claramente, el estimador $\hat{\delta}_{MCO}$ está centrado en 3 y su varianza tiende a disminuir a medida que aumente N lo cuál hace pensar que la distribución asintóticamente converge a una normal degenerada centrada en $\delta = 3$, que es el parámetro poblacional.
  Ahora bien, como lo que se está solicitando es:

$$a_{k,n} = \sqrt{n}(\hat{\delta}_{k,n} - \delta)$$

  el siguiente código calcula dicho $a_{k,n}$ y almacena sus valores:

```

# Ejercicio c. ----

# I. ----

# Definici n de par metros para la simulaci n

## N mero de muestras:
t2 = 1000

## Tama o de muestras:

# No se puede trabajar con n10 = 10 porque no satisface la condici n de rango
```

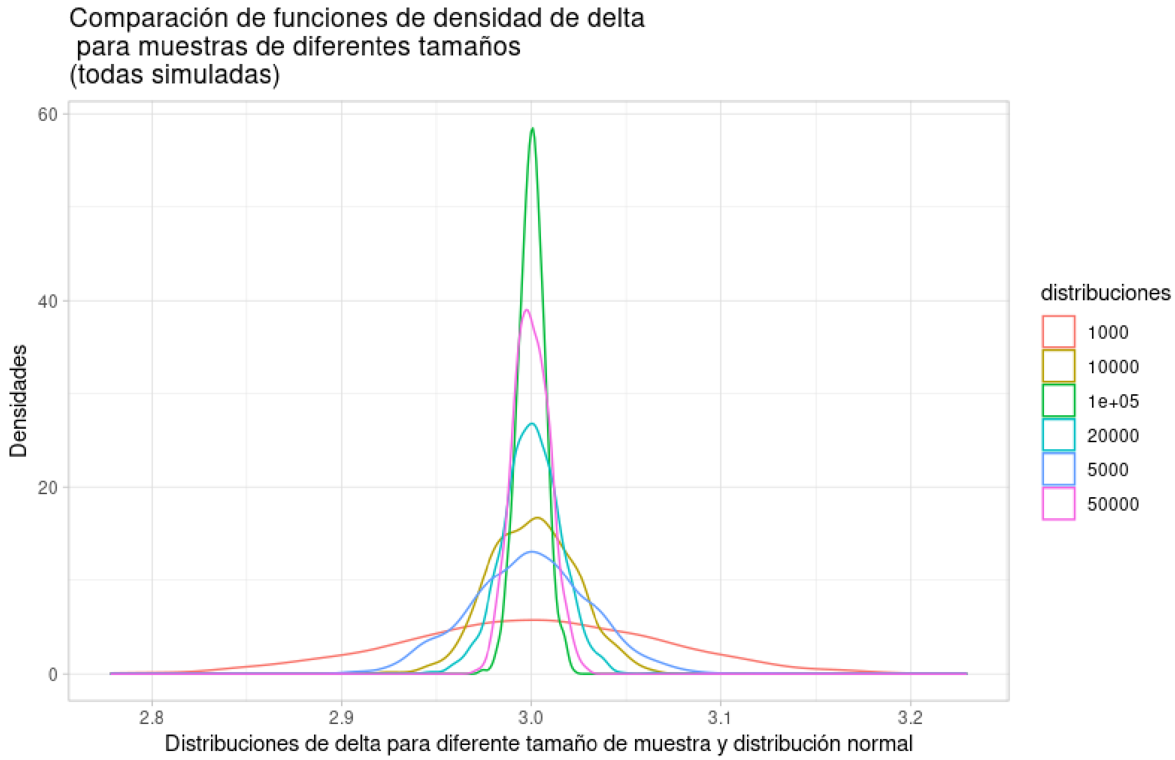


Figura 5: Comportamiento de la distribución asintótica del estimador δ_{MCO} para una muestra que distribuye normal a medida que el tamaño de la muestra aumenta

```
# n10 = 10 Genera probelmas de singularidad cuando se generan muchas muestras
#           Porque por chance, se genera una muestra donde D = rep(0, times = 10)
n20 = 20
n100 = 100
n1000 = 1000
n5000 = 5000
n10000 = 10000
n20000 = 20000
n50000 = 50000
n100000 = 100000
n_vector2 = c(n20, n100, n1000, n5000, n10000, n20000, n50000, n100000)

# Funci n que calcula a = sqrt(n) (delta_est - truth_delta) para cada tama o de
    muestra n
calculo_a = function(t, n, distro, truth_delta){
  # Variables:
  ## t: n mero de muestras a simular
  ## n: tama o de la muestra
  ## truth_delta: verdadero valor del par metro delta (valor poblcional del par metro)
  # df_estimados es el dataframe que contiene el delta estimado por SDO
  # y por medio de regresi n
  df_estimados = delta(t, n, distro)
  # df_a es el dataframe que contiene
  df_a = df_estimados %>%
    mutate(a = sqrt(n) * (delta_est - truth_delta), tama o = rep(n, times = t))
  return(df_a)
}

# Defino una funci n para crear un dataframe que se va a utilizar
# Para construir la gr fica multipanel.
base_para_graficas = function(t, n_vector, distro, truth_delta){
  for (i in 1:length(n_vector)){
    # Si i == 1 significa que estamos en la primera iteraci n
    if (i == 1){
      df_a_total = calculo_a(t, n_vector[i], distro, truth_delta = 3)
    }else{
      df_prov = calculo_a(t, n_vector[i], distro, truth_delta = 3)
      df_a_total = bind_rows(df_a_total, df_prov)
    }
  }
  # Retorna un dataframe con los diferentes valores de a para cada tama o de muestra
      listo
  # para construir la gr fica multipanel
  return(df_a_total)
}

# La base multipanel me tiene lista los diferentes a
# para cada tama o de muestra
base_multipanel = base_para_graficas(t2, n_vector2, distro = "normal", truth_delta = 3)
```

II. Grafique las densidades estimadas de los $a_{k,n}$ para cada n. ¿Qué puede apreciar a medida que aumenta n? ¿A qué se debe este resultado?

- Solución:

  Las gráficas de las densidades estimadas de los $a_{k,n}$ para cada tamaño de muestra n están dadas por[13] la figura 6.

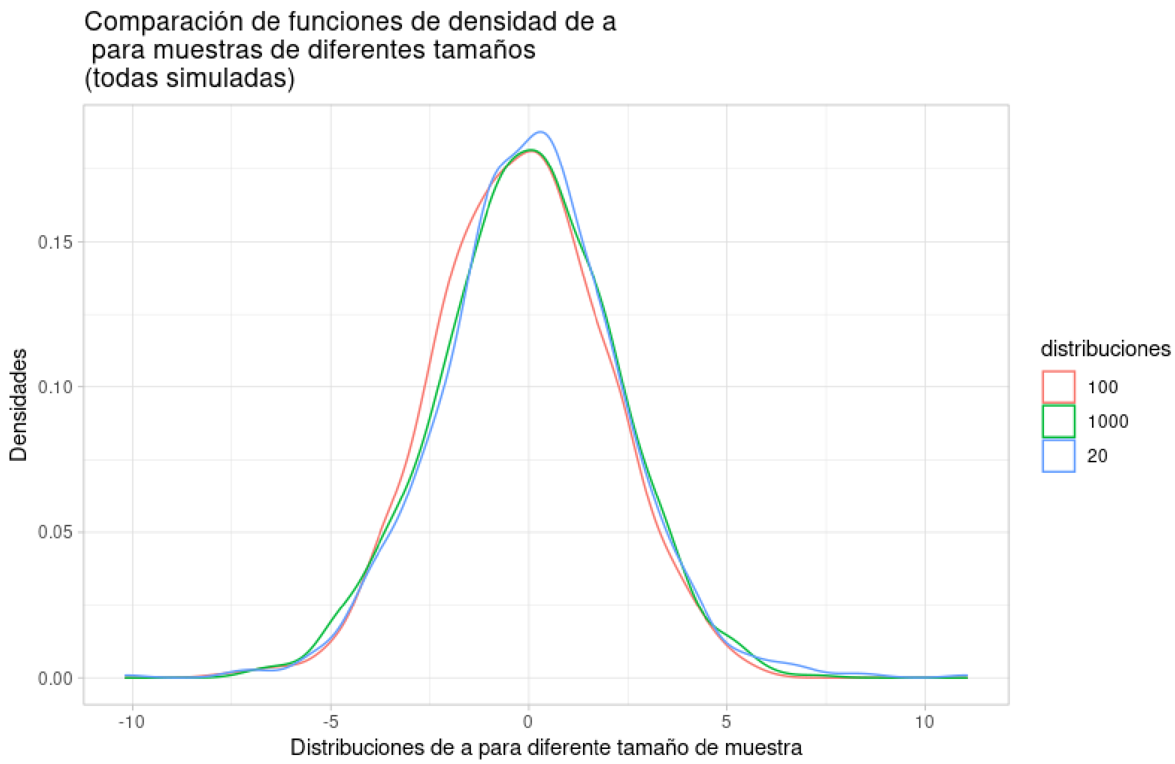


Figura 6: Distribuciones asintóticas de $a_{k,n}$ en donde claramente se satisface el teorema de limite central para una muestra que distribuye normal.

Lo que se puede apreciar de la figura 6 es que claramente cuando se tiene muestras normales, el *teorema del limite central* se satisface para la media muestral del estimador δ_{MCO}. Como se puede oberva para todo tamaño de muestra, se observa claramente una distribución normal centrada en cero y que a medida que n aumenta se observa una convergencia de la distribución. Nuevamente, dicha convergencia se debe principalemnte a que la media muestral del estimador δ_{MCO} satisface el *teorema del limite central*.

El código para generar la gráfica fue el siguiente:

```

# Defino una funci n para crear la gr fica multipanel con el histograma y la funci n
    de densidad
# para cada tama o de muestra n
histogram_grid = function(df, titulo, x_lab, y_lab, num_bins = 30, y_upper_limit){
  histog_grid = df %>%
    ggplot(aes(a)) +
    scale_y_continuous(limits = c(0, y_upper_limit)) +
    geom_histogram(aes(y = ..density..), color = "black", bins = num_bins) +
    geom_density(color = "green") +
    facet_grid(cols = vars(tama o)) +
    ggtitle(titulo) +
    xlab(x_lab) +
    ylab(y_lab) +
    theme_light()
}

grafica_multi = histogram_grid(df = base_multipanel, titulo = "Gr fico multipanel para
    a con diferentes tama os de muestra", x_lab = "Tama o de muestras (n)", y_lab = ""
    , num_bins = 30, y_upper_limit = 0.2); grafica_multi

# Gr fica que compara directamente las densidades
compracion_densidades_grafica = function(t, df, var){
  # Variables:
  ## t: n mero de muestras simuladas por tama o de muestra
```

[13] Por las instrucciones del taller, se emplearon muestras de tamaño 20, 100, 1000. Las muestras de tamaño n = 10 no se emplearon por los problemas de multicolinealidad comentados anteriormente

```
  ## df: df multipanel que tiene las simulaciones de las diferentes muestra por tama o
      de muestra
  ## var: variable simulada de la cual se va a generar las funciones de densidad
  # Nota:
  # Modifico el dataframe multipanel para seleccionar solo
  # las variables a y tama o que permiten hacer la gr fica de densidades
  # df_mod es un dataframe modifiado del dataframe multipanel
  df_mod = df %>%
    rename(valores_sim = {{ var }}, distribuciones = tama o) %>%
    mutate(distribuciones = as.character(distribuciones)) %>%
    select(valores_sim, distribuciones)
  # Simulo de una distribuci n normal est ndar porque quiero probar
  # si se cumple o no el teorema del l mite central
  df_normal = data.frame(valores_sim = rnorm(t), distribuciones = rep("Normal est ndar"
      , times = t))
  # base_grafica ya es la base de datos lista para realizar las gr ficas de
  # las densidades de las a y de una normal est ndar
  base_grafica = df_mod
  # base_grafica = bind_rows(df_mod, df_normal)
  # Gr fica de densidades
  density_comparacion = base_grafica %>%
    ggplot(aes(x = valores_sim, color = distribuciones)) +
    geom_density() +
    theme_light() +
    ggtitle("Comparaci n de funciones de densidad de a\n para muestras de diferentes
        tama os \n(todas simuladas)") +
    ylab("Densidades") +
    xlab("Distribuciones de a para diferente tama o de muestra")
  return(density_comparacion)
}

# Visualizaci n de la base de datos con las variables simuladas
# delta_est y a
glimpse(base_multipanel)

base_filtrada = base_multipanel %>%
  filter(100 < tama o)

base_filtrada_a = base_multipanel %>%
  filter(10 < tama o) %>%
  filter(tama o < 2000)

# Gr ficas de las funciones de densidad para la simulaci n de los delta estimados
# y de los a, para el caso de un Y_{i}^{0} = norm(0, 1)
grafica_a = compracion_densidades_grafica(t = t2, df = base_filtrada_a, var = a);
    grafica_a
grafica_delta_est = compracion_densidades_grafica(t = t2, df = base_filtrada, var =
    delta_est); grafica_delta_est
```

Suponga ahora que usted sabe que existen factores ajenos a las políticas del Ministerio que inciden sobre los resultados potenciales de los alumnos. Por ejemplo, sabe que existen alumnos que, independientemente de si son o no seleccionados, igual se inscribirían en cursos de preparación para presentar el ICFES. Similarmente, hay alumnos que por condiciones adversas (enfermedades, salones menos adecuados para la presentación del examen) exhiben un desempeño muy inferior a lo esperado. Estos factores producen una mayor volatilidad en los resultados potenciales de los alumnos.

d) En este sentido, usted sabe que es más plausible considerar que los resultados potenciales en realidad obedecen la ley $Y_i(0) \sim$ *Cauchy* (Distribución Cauchy estándar) en vez de una normal estándar. Esto es, en ocasiones usted observa datos atípicos que no parecen corresponder con el patrón general de la muestra. Repita los incisos I y II del c) bajo estas condiciones. ¿Por qué en este caso no se satisface el Teorema del Límite Central?¿Qué consecuencias tendría esto para la inferencia de los parámetros de MCO si usáramos los procedimientos estadísticos usuales en este caso?

- Solución:

  Ahora bien, para la solución de este ejercicio se volvieron a repetir los incisos I y II del punto c) pero asumiendo que el outcome potencial en ausencia de tratamiento distribuye como una $Y(0)$ $Cauchy(0,1)$[14]. La idea de utilizar una distribución Cauchy, es que permite observar datos atıpicos que no parecen corresponder con el patrón general de la muestra.

  Lo primero, fue simular el estimador de MCO$\hat{}_{MCO}$ bajo la nueva muestra que distribuye Cauchy.

  La gráfica 7 muestra la distribución asintótica del estimador$\hat{}_{MCO}$. Lo primero que es evidente que es que se observa que dicho estimador *distribuye como una Cauchy*. Esto es particularmente cierto, si se

[14]Una distribución $Cauchy(0,1)$, es una distribución Cauchy estándar con parámetro de localización 0 y parámetro de escala 1(que le da la forma a la distribución)

Figura 7: Comportamiento de la distribución asintótica del estimador δ_{MCO} para una muestra que distribuye normal a medida que el tamaño de la muestra aumenta

tiene en cuenta la cantidad muy alta de observaciones que se encuentran alrededor del parámetro de localización 0, pero también por que existen valores atípicos muy extremos[15].

Ahora bien, las gráficas de las densidades estimadas de los $a_{k,n}$ para cada tamaño de muestra n, donde la muestra distribuye Cauchy en lugar de normal están dadas por[16] por la figura 8.

El siguiente código se empleo para realizar las dos gráficas anteriores de donde simulo de una *distribución Cauchy*:

```

# Construyo la base multipanel para los diferentes a, asumiendo que Y_{i}^{0} sigue una
    distribuci n Cauchy

base_multipanel_cauchy = base_para_graficas(t2, n_vector2, distro = "cauchy", truth_delta =
    3)

# II. ----

base_filtrada_cauchy = base_multipanel_cauchy %>%
  filter(tama o > 20)
  # filter(tama o < 10000)

base_filtrada_cauchy2 = base_multipanel_cauchy %>%
  filter(tama o > 10) %>%
  filter(tama o < 2000)

# Gr ficas de las funciones de densidad para la simulaci n de los delta estimados
# y de los a, para el caso de un Y_{i}^{0} = cauchy(0, 1)
grafica2_a = compracion_densidades_grafica(t = t2, df = base_filtrada_cauchy2, var = a);
    grafica2_a
grafica2_delta_est = compracion_densidades_grafica(t = t2, df = base_filtrada_cauchy, var =
    delta_est); grafica2_delta_est
```

Como se puede observar de 8 claramente, no se satisface el *teorema del limite central* cuando se tiene una muestra que *distribuye Cauchy*.

El teorema del limite central no se satisface principalmente porque la *distribución Cauchy* es una *distribución patológica*, en el sentido que no es una distribución que tenga las propiedades usuales que uno esperaría en una distribución. En específico, la distribución Cauchy, tiene media infinita y varianza

[15] A medida que el tamaño de las muestras aumente, es más probable que surgan valores atípicos muy extremos por lo que es más probable encontrar valores atípicos bastante extremos en los estimadores MCO para muestras de tamaño 1000 en comparación con las muestras de tamaño 20

[16] Por las instrucciones del taller, se emplearon muestras de tamaño 20, 100, 1000. Las muestras de tamaño n = 10 no se emplearon por los problemas de multicolinealidad comentados anteriormente

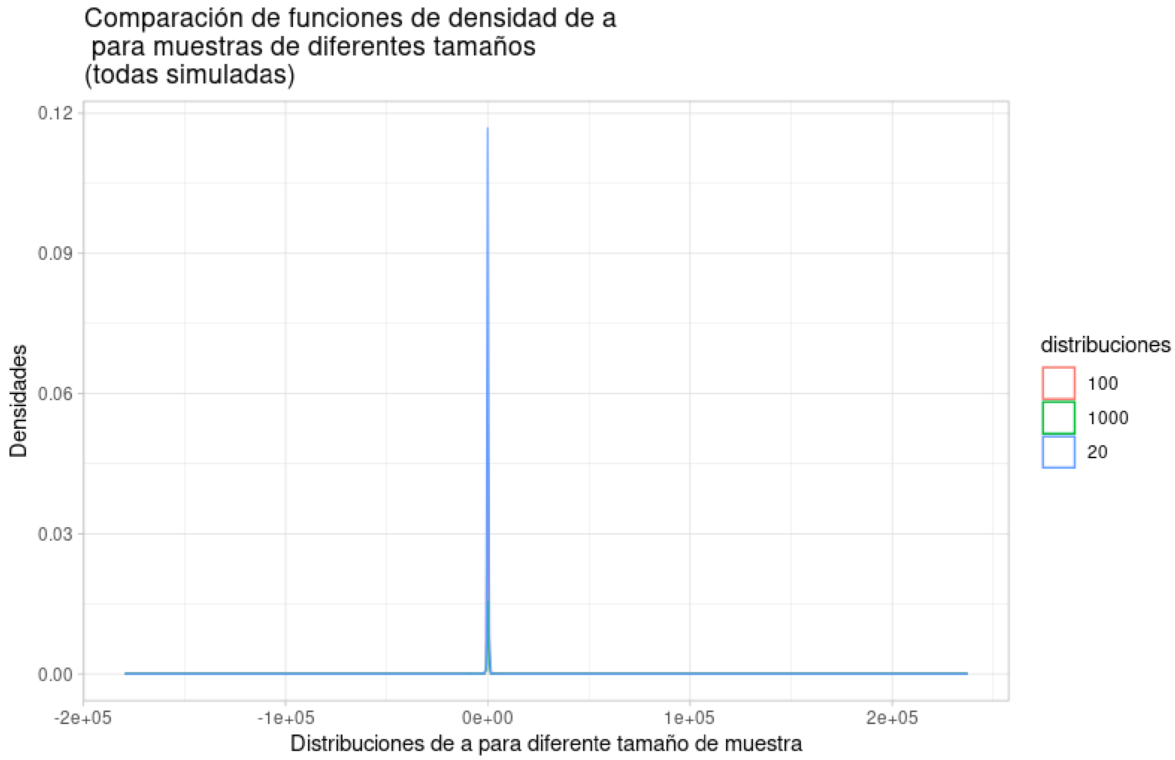


Figura 8: Distribuciones asintóticas de $a_{k,n}$ en donde claramente **no** se satisface el teorema de limite central para una muestra que distribuye Cauchy.

infinita, lo cuál hace que sea una distribución que tenga características muy particulares, entre ellas que pueda tener valores extremos muy atípicos y además que no satisfaga los teoremas clásicos de teoría asintótica como la ley débil de los grandes números y el teorema del límite central. Si se mira con detalle, el teorema del limite central está planteado para una distribución en la que se conozca la media poblacional y varianza poblacional de la variable aleatoria en la que se está aplicando el teorema. Como en una variable aleatoria, dicha media poblacional y varianza poblacional no existe, o más específicamente son infinitas, entonces no es posible que se satisfaga el teorema del limite central porque dos de las propiedades necesarias que debe tener una variable aleatoria para satisfacer el teorema de limite central, en particular tener valor esperado y varianza finita, no se satisface. En particular, cuando se tiene una variable aleatoria Cauchy, se sabe que en lugar de que su media muestral, debidamente normalizada, en lugar de converger a una distribución normal con varianza conocida, distribuye también como una variable aleatoria Cauchy[17]

Ahora bien, si empleáramos *inferencia convencional* para parámetros MCO a una muestra que distribuya Cauchy, se van a obtener conclusiones incorrectas a partir de dicha inferencia que se realice. Esto se debe a que la inferencia convencional que uno emplea cuándo hace estimaciones por MCO, solo es válida si se satisface la teoría asintótica del estimador MCO, y la distribución del estimador MCO es la esperada por lo que dictamina la teoría asintótica, en particular, dicha inferencia convencional que se usa cuando se emplea el estimador MCO depende crucialmente de que se satisfaga el teorema del limite central. Ahora bien, como en una muestra que distribuya Cauchu, en general no se cumple la teoría asintótica convencional, y en particular no se satisface el teorema del limite central, toda inferencia convencional del estimador MCO que se emplee bajo dicha muestra Cauchy será incorrecta dado que, supuestos importante como el cumplimiento del teorema del limite central no se da, y eso que la distribución asintótica del estimador de MCO deje de ser la esperada por teoría asintótica.

Finalmente, suponga que usted sabe que el programa implementado por el ministerio tiene el objetivo de nivelar los conocimientos de los estudiantes que en el participan. Por dicha razón, los resultados de las pruebas de los estudiantes que no participan en el programa son mas volátiles.

Una manera de modelar esta observación es a través del modelo:

$$Y_i = 3 * D_i + v_i$$

[17]Es decir, la media muestral de variables Cauchy i.i.d. también distribuye como una variable aleatoria Cauchy.

donde $v_i \sim N(0, 2 - D_i)$.

e) Para este nuevo modelo, realice el siguiente procedimiento:

I. Simule 1000 muestras $\{(Y_i(0), Y_i(1), D_i\}_{i=1}^{1000}$. Para cada muestra, recupere los intervalos de confianza clásicos al 90 %, 95 % y 99 % obtenidos para δ y codifique en una matriz si dicho intervalo contiene o no a δ.

- Solución:

Se tiene que el siguiente modelo representa las observaciones para el caso expuesto en este inciso:

$$Y_i = 3D_i + v_i \tag{8}$$

Ahora bien, como conocemos que $v_i \sim \mathcal{N}(0, 2-D_i)$, entonces se puede decir que una representación de la distribución de Y_i estaría dada por:

$$Y_i = 3D_i + \mathcal{N}(0, 2 - D_i)$$

Por tanto, si $D_i = 0$, entonces $Y_i = Y_i(0) = 0 + \mathcal{N}(0, 2)$, lo que hace que:

$$Y_i(0) \sim \mathcal{N}(0, 2)$$

y si si $D_i = 1$, entonces $Y_i = Y_i(1) = 3 + \mathcal{N}(3, 1)$, lo que hace que:

$$Y_i(1) \sim \mathcal{N}(3, 1)$$

Dicho lo anterior, se simulan 1000 muestras de tamaño $n = 1000$ cada una, y se recuperan los intervalos de confianza clásicos al 90 %, 95 % y 99 % obtenidos para δ y se codifican en una matriz que informa si dicho intervalo contiene o no δ. El código para realizar ello es el siguiente:

```

# I. ----

# Tama os de muestra
n20 = 20
n100 = 100
n200 = 200
n500 = 500
n1000 = 1000
n10000 = 10000

hetero20_no_robustos = delta(t = 1000, n = 20, distro = "normal", hetero = T, varianza_
    y0 = 2, varianza_y1 = 1, robustos = F)
hetero100_no_robustos = delta(t = 1000, n = 100, distro = "normal", hetero = T, varianza
    _y0 = 2, varianza_y1 = 1, robustos = F)
hetero200_no_robustos = delta(t = 1000, n = 200, distro = "normal", hetero = T, varianza
    _y0 = 2, varianza_y1 = 1, robustos = F)
hetero500_no_robustos = delta(t = 1000, n = 500, distro = "normal", hetero = T, varianza
    _y0 = 2, varianza_y1 = 1, robustos = F)
hetero1000_no_robustos = delta(t = 1000, n = 1000, distro = "normal", hetero = T,
    varianza_y0 = 2, varianza_y1 = 1, robustos = F)
hetero10000_no_robustos = delta(t = 1000, n = 10000, distro = "normal", hetero = T,
    varianza_y0 = 2, varianza_y1 = 1, robustos = F)

n_vect_hetero = c(n20, n100, n200, n500, n1000, n10000)

# Dataframe con el tama o de muestra y la media y varianza del
# estimador de delta para los diferentes tama os de muestra
media_varianza_hetero = media_varianza_delta(t = 1000, n_vect_hetero, distro = "normal",
     hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F); glimpse(media_varianza
    _hetero)

# Nota: Se observa que a pesar de la heterocedasticidad en Y_{i} hay insesgadez
#       En la estimaci n de delta
#       De igual forma, la varianza muestral tambi n disminuye a medida que aumenta
#       la muestra (independiente de que haya heteroecedasticidad en Y_{i})

# Construir una funci n que me compute los intervalos de confianza
# Intervalos de confianza: Ya sea para intervalos cl sicos o para intervalos robustos
#                          a la heterocedasticidad como los calculados por la matriz de
    varianzas y covarianzas Huber-White

inter_confianza = function(df_estimaciones, delta_true, int_conf){
  norm_inf = qnorm((1 - int_conf)/2)
  norm_sup = qnorm((1 - int_conf)/2 + int_conf)
  lim_inf = df_estimaciones$delta_est + norm_inf * sqrt(df_estimaciones$var_ols)
  lim_sup = df_estimaciones$delta_est + norm_sup * sqrt(df_estimaciones$var_ols)
  contiene_o_no = c()
  for (i in 1:length(lim_inf)){
    if ((lim_inf[i] < delta_true) && (delta_true < lim_sup[i])){
      contiene_o_no = append(contiene_o_no, 1)
```

```
    }else{
      contiene_o_no = append(contiene_o_no, 0)
    }
  }
  # Variable del dataframe
  df = data.frame(lim_inf, lim_sup, contiene_o_no)
  return(df)
}

# Intervalos de confianza del 90 %
int_conf1000_no_robustos_90 = inter_confianza(hetero1000_no_robustos, delta_true = 3,
    int_conf = 0.9)

# Intervalos de confianza del 95 %
int_conf1000_no_robustos_95 = inter_confianza(hetero1000_no_robustos, delta_true = 3,
    int_conf = 0.95)

# Intervalos de confianza del 99 %
int_conf1000_no_robustos_99 = inter_confianza(hetero1000_no_robustos, delta_true = 3,
    int_conf = 0.99)
```

II. ¿Qué porcentaje de los intervalos de cada nivel contiene al parámetro verdadero? Presente sus resultados en una tabla ¿A qué se deben estos resultados? ¿Qué le sugiere esto sobre su forma de computar los intervalos?

- Solución:

  Para encontrar el porcentaje de los intervalos de cada nivel de significancia que contiene al parámetro verdadero, se empleo el siguiente código:

```

# II. ----

# Porcentaje de los intervalos de confianza del 90 % que contienen el par metro
    verdadero
porcentaje_no_robustos_90 = sum(int_conf1000_no_robustos_90$contiene_o_no)/nrow(int_
    conf1000_no_robustos_90) * 100; porcentaje_no_robustos_90

# Porcentaje de los intervalos de confianza del 95 % que contienen el par metro
    verdadero
porcentaje_no_robustos_95 =  sum(int_conf1000_no_robustos_95$contiene_o_no)/nrow(int_
    conf1000_no_robustos_95) * 100; porcentaje_no_robustos_95

# Porcentaje de los intervalos de confianza del 99 % que contienen el par metro
    verdadero
porcentaje_no_robustos_99 = sum(int_conf1000_no_robustos_99$contiene_o_no)/nrow(int_
    conf1000_no_robustos_99) * 100; porcentaje_no_robustos_99
```

  Los resultados se pueden observar mejor en la tabla 1[18].

Cuadro 1: Tabla que muestra el porcentaje de los intervalos clásicos que contienen el parámetro poblacional $\delta = 3$ a diferentes niveles de significancia

Nivel de significancia	Porcentaje de los intervalos de cada nivel que contiene al parámetro verdadero
90 %	93.9 %
95 %	98 %
99 %	99.8 %

  Lo que se observa, es que en presencia de heterocedasticidad para los errores del modelo, los intervalos de confianza clásicos rechazan menos veces de las que deberían al nivel de significancia propuesta la hipótesis nula, es decir, son más amplios de lo que deberían ser, y por tanto, hay más muestras que contienen el parámetro verdadero $\delta = 3$ de las que deberían dado el nivel de significancia escogido, y ésto ocurre para todos los niveles de significancia. Pareciera que los intervalos de confianza son demasiado amplios de manera sistemática sin importar el nivel de significancia. Esto se debe principalmente, a que la matriz de varianzas y covarianzas, bajo el supuesto de homocedasticidad, no es la adecuada a la representar la matriz de varianzas y covarianzas asintótica del estimador MCO cuando hay heterocedasticidad en la muestra.

III. Repita los numerales I y II, esta vez empleando errores estándar de White (robustos) en la construcción de sus intervalos. Concluya.

- Solución:

[18] Es importante notar que se simular en total $t = 1000$ muestras, donde cada muestra tenía un tamaño de muestra de $n = 1000$. Lo que permite concluir que, los resultados obtenidos tanto en el inciso II. como en el inciso III.(que utilizó la misma metodología) son bastante robustos y confiables, y muestran claramente el comportamiento tanto de los intervalos clásicos como de los intervalos construidos con errores robustos ante la presencia de heterocedasticidad en la muestra.

Dado lo encontrado en los incisos I. y II. del punto e., se propone a repetir el mismo procedimiento que el anterior pero esta vez utilizando una *matriz de varianzas y covarianzas* correspondientes a *errores robustos a la heterocedasticidad tipo Huber-White*. Para ello, se emplea el siguiente fragmente de código:

```

# III. ----

hetero20_robustos = delta(t = 1000, n = 20, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero100_robustos = delta(t = 1000, n = 100, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero200_robustos = delta(t = 1000, n = 200, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero500_robustos = delta(t = 1000, n = 500, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero1000_robustos = delta(t = 1000, n = 1000, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero10000_robustos = delta(t = 1000, n = 10000, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)

n_vect_hetero = c(n20, n100, n200, n500, n1000, n10000)

# Dataframe con el tama o de muestra y la media y varianza del
# estimador de delta para los diferentes tama os de muestra
media_varianza_hetero = media_varianza_delta(t = 1000, n_vect_hetero, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T); glimpse(media_varianza_hetero)

# III. C lculo de intervalos de confianza cuando se tienen errores robustos ----

# Intervalos de confianza del 90 %
int_conf1000_robustos_90 = inter_confianza(hetero1000_robustos, delta_true = 3, int_conf = 0.9)

# Intervalos de confianza del 95 %
int_conf1000_robustos_95 = inter_confianza(hetero1000_robustos, delta_true = 3, int_conf = 0.95)

# Intervalos de confianza del 99 %
int_conf1000_robustos_99 = inter_confianza(hetero1000_robustos, delta_true = 3, int_conf = 0.99)

# III. Porcentaje de los intervalos de confianza cuando se tienen errores robustos ----

# Porcentaje de los intervalos de confianza del 90 % que contienen el par metro verdadero
porcentaje_robustos_90 = sum(int_conf1000_robustos_90$contiene_o_no)/nrow(int_conf1000_robustos_90) * 100; porcentaje_robustos_90

# Porcentaje de los intervalos de confianza del 95 % que contienen el par metro verdadero
porcentaje_robustos_95 = sum(int_conf1000_robustos_95$contiene_o_no)/nrow(int_conf1000_robustos_95) * 100; porcentaje_robustos_95

# Porcentaje de los intervalos de confianza del 99 % que contienen el par metro verdadero
porcentaje_robustos_99 = sum(int_conf1000_robustos_99$contiene_o_no)/nrow(int_conf1000_robustos_99) * 100; porcentaje_robustos_99

# Conclusi n: Los errores robustos c lculados con matriz de varianzas y covarianzas Huber-White,
#             s  sirven cu ndo hay heterocedasticidad en los errores de modelo
#             Por el contrario, si se utilizan intervalos de confianza cl sicos
#             sin corregir por heterocedasticidad, se observa que los intervalos
#             de confianza cl sicos de manera sistem tica para los niveles de
#             significancia sobreestiman, es decir calculan intervalos de
#             confianza m s amplios de los que en realidad deberian computarse
#             dada la presencia de la heterocedasticad
```

En dicho código, se generan los intervalos de confianza pero ahora construidos con una matriz de varianzas y covarianzas correspondiente a *errores robustos a la heterocedasticidad tipo Huber-White* y se calcula cuántos de éstos intervalos contienen el parámetro poblaciona $\delta = 3$

Los resultados se pueden observar mejor en la tabla 2.

Cuadro 2: Tabla que muestra el porcentaje de los intervalos construidos con errores robustos que contienen el parámetro poblacional $\delta = 3$ a diferentes niveles de significancia

Nivel de significancia	Porcentaje de los intervalos de cada nivel que contiene al parámetro verdadero
90 %	89.7 %
95 %	94.9 %
99 %	99.6 %

Lo que se puede concluir, es que al emplear intervalos construidos con errores robustos basados en una matriz de varianzas y covarianzas *Huber-White*, es posible observar que el porcentaje de intervalos que contienen el parámetro poblaciona $\delta = 3$ es muy cercano al nivel de significancia propuesto. Lo que quiere decir, es que al emplear una matriz de varianzas y covarianzas *Huber-White*, claramente los intervalos de confianza tienen una longitud que corresponde mucho mejor al nivel de significancia propuesto por lo que claramente éstS intervalos de confianza son mejores que los intervalos de confianza clásicos como se puede evindenciar fácilmente de las tablas 1 y 2. Lo anterior, se explica por el hecho de que la matriz de varianzas-covarianzas *Huber-White* está diseñada para capturar la posible heterocedasticidad que puede presentar la muestra por lo que se puede ver como una generalización de la matriz de vaianzas y covarianzas homocedásticas, y por tanto, en presencia de heterocedasticidad, arroja mejores intervalos de confianza.

Punto doctorado

Usted se encuentra cursando el curso de seminario de investigación doctoral. En las primeras semanas le piden que presente algunas relaciones de causalidad que tenga pensado abordar en su disertación. Su profesora de seminario lo invita a que avance en el planteamiento de sus ideas a través del uso de gráficos acíclicos dirigidos (Directed Acyclic Graph - DAG).[19] En breve, los DAGs se componen de dos elementos: flechas y variables. Las flechas entre variables indican causalidad en el sentido en el que esta apuntando la flecha. Un ***Camino*** es cualquier conexión entre dos variables realizada por flechas, sin importar su dirección o si existen otras variables intermedias. Un ***Camino por la puerta trasera*** de la variable X a la variable Y es un camino que empieza con una flecha dirigida hacia X. Un ***Colisionador*** es una variable a la que apuntan dos flechas en un camino.

a) Usando puntos negros para graficar las variables observadas y circunferencias para graficar las variables no observadas (X, Z y Y), realice los DAGs que representan:

   i) X tiene una relación de causalidad hacia Y. A su vez, Z tiene una relación de causalidad hacia X y otra hacia Y.

   ii) X tiene una relación de causalidad hacia Z. A su vez, Z tiene una relación de causalidad hacia Y.

   iii) La variable no observada: U tiene una relación de causalidad hacia Z y otra hacia Y. A su vez, X tiene una relación de causalidad hacia Y y otra hacia Z.

   iv) La variable X tiene una relación de causalidad hacia Z y otra hacia Y. A su vez, Y tiene una relación de causalidad hacia Z

   v) La variable no observada: U tiene una relación de causalidad hacia Z y otra hacia Y. A su vez, X tiene una relación de causalidad hacia Y, mientras que Z tiene una relación de causalidad hacia X.

b) Ahora usted quiere implementar un modelo de regresión lineal simple para evaluar los efectos causales de X sobre Y. Con este objetivo, usted está determinando la conveniencia de incluir la variable Z como variable de control en su modelo. Argumente, para cada caso representado por los DAGs, si la inclusión de Z, como control en el modelo, es apropiada. Para sus respuestas, tenga en cuenta que un variable es un mal control si: bloquea caminos causales entre X y Y o abre otros caminos que no son causales entre X y Y.

   (***Pista****: Bloquear un camino es equivalente a controlar por variables que no son Colisionadoras o no controlar por las Colisionadoras. Abrir un camino es equivalente a controlar por variables que son Colisionadoras o no controlar por no Colisionadoras)*

c) Finalmente, usted sabe que la variable X es independiente de los resultados potenciales de Y y de Z. Por otro lado, usted sospecha que Z podría ser un mal control y que este hecho puede generarse porque la variable Z es, a su vez, una variable de resultado en su modelo. Plantee una forma de aproximarse empíricamente para evaluar si este hecho puede estar ocurriendo

[19]Si no se encuentra familiarizado con los DAGs, una excelente referencia y explicación se encuentra en el tercer capítulo del libro de Scott Cunningham: Causal Inference: The mixtape, del año 2021. https://mixtape.scunning.com/

Anexo: Código segundo ejercicio

A continuación se anexa todo el código empleado para resolver el *segundo ejercicio* del taller:

```

// Taller 1
use "/Users/federicoduenas/Desktop/Econometri a_ Avz_/taller 1/manufacturaCol.dta"

foreach var of varlist producto_acum producto costos {
    gen ln_‘var’=ln(‘var’+1)
}


// relacion de log_c con log_n controlando por log_y
// usando Partialling out:

reg ln_producto_acum ln_producto

gen aux_1 = 9.256792 + 0.1120002*ln_producto

// para tener la parte de N que no es explicada por Y
gen e_1 = ln_producto_acum - aux_1

// ahora para tener los C_i no explicada por Y_i

reg ln_costos ln_producto

gen aux_2_1 = -.019997 + .0052894*ln_producto

// los residuales: parte de C_i no explicada por Y_i
gen e_1_2 = ln_costos - aux_2_1

// sactter de la relaci n C_i con N_i una vez se removi  el efecto de Y_i
**#

scatter e_1_2 e_1 || lfit e_1_2 e_1

/*


// ahora si calculo la relacion entre N y C, controlando por Y

reg ln_costos e_1

// Que, por teorema de Waugh-Frisch-Lovell, es equivalente al par metro de log_N de la reg lineal:
reg ln_costos ln_producto_acum ln_producto

// gr fica de la relaci n de C y N:
scatter ln_costos e_1  || lfit ln_costos e_1
*/
// ahora la relacion de log_C y log_Y controlando por log_N

reg ln_producto ln_producto_acum

gen aux_2 = .3859203 + .4743226*ln_producto_acum

gen e_2 = ln_producto - aux_2


// ahora remuevo el efecto de N_i sobre C_i

reg ln_costos ln_producto_acum
gen aux_2_2 = -.013807 + .0020865*ln_producto_acum
gen e_2_2 = ln_costos - aux_2_2


// sactter de la relaci n de C y Y removiendo el efecto de N_i

scatter e_2_2 e_2 || lfit e_2_2 e_2

/*
reg ln_costos e_2

reg ln_costos ln_producto_acum ln_producto


scatter ln_costos e_2  || lfit ln_costos e_2

// estimar la ecuaci n del modelo:

reg ln_costos ln_producto_acum ln_producto
*/
reg ln_costos ln_producto_acum ln_producto


matlist e(b)
matlist e(V)
```

Anexo: Código tercer ejercicio

A continuación se anexa todo el código empleado para resolver el *tercer ejercicio* del taller:

```
library(tidyverse)

# Definir la semilla para reproducibilidad en los resultados
set.seed(12345)

# Punto 3 ----

# Ejercicio a. ----

# Switching regression:
## y_{i} = y_{i}^{1} * D_{i} + (1 - D_{i}) * y_{i}^{0}
## y_{i} = y_{i}^{0} + (y_{i}^{1} - y_{i}^{0}) * D_{i}
## y_{i} = delta * D_{i} + epsilon_{i}

# Donde el par metro causal:
## delta = E[y_{i}^{1} | D_{i} = 1] - E[y_{i}^{0} | D_{i} = 0]
## delta = 3 - 0
## delta = 3

# Ejercicio b. ----

# Nota: Dado las condiciones para simular y_0 = y_{i}^{0} y y_1 = y_{i}^{2},
#       uno esperar a que el SDO = 3 y por tanto el delta = 3
#       Ademas, dado que el SB = 0 (teniendo en cuenta que y_0 y y_1 son independientes de D)
#       se tiene que delta representa el impcto causal del programa del ministerio

# I. ----

# Definici n de par metros para la simulaci n

## N mero de muestras:
t = 100

## Tama o de muestras:

n10 = 10
n20 = 20
n50 = 50
n100 = 100
n500 = 500
n1000 = 1000

#__________________________
# Funciones auxiliares:
#__________________________

# C lculo del SDO: (SDO := Simple difference of outcomes) (Funci n auxiliar)
SDO_calculo = function(Y, D){
  # Por construcci n
  ## Y = Y_0 Si D = 0
  ## Y = Y_1 Si D = 1
  # Nota: Y y D son vectores del mismo tama o
  # Inicializaci n de contadores
  count0 = 0
  count1 = 0
  # Inicializaci n de la suma para c lculo del SDO
  suma0 = 0
  suma1 = 0
  # Iteraci n a trav s de la muestra
  for (i in 1:length(Y)){
    if (D[i] == 0){
      suma0 = suma0 + Y[i]
      count0 = count0 + 1
    }else{
      suma1 = suma1 + Y[i]
      count1 = count1 + 1
    }
  }
  SDO = (suma1/count1) -(suma0/count0) # SDO: Simple diferencia de muestras
  return(SDO)
}

# Construcci n matriz X para un modelo de regresi n lineal con constante: (Funci n auxiliar)
ols_X = function(...){
  # ... contiene los regresores necesarios para construir la matriz X
  regresores = list(...)
  # Nota: Cada regresor es un vector con el mismo n mero de observaciones n
  n = length(regresores[[1]]) # No importa que se tome el primer regresor dado que todos los
      regresores tienen el mismo tama o
  const = rep(1, times = n)
  X = cbind(const, ...)
  return(X)
}

# Construcci n del estimador \Hat{sigma^{2}}: (Funci n auxiliar)
estimador_sigma = function(y, X, beta_est){
  # Variables:
```

```r
  ## y: es el vector que contiene la variable dependiente
  ## X: es la matriz que contiene las variables regresoras
  ## beta_est: es el vector de los par metros estimados por OLS
  e = y - X %*% beta_est # e es el vector de residuales
  k = ncol(X)  # El n mero de par metros a estimar
  n = length(y)  # El n mero de observaciones en la mustra
  sigma_est = (t(e) %*% e) / (n - k)   # estimador de sigma (varianza del t rmino de error)
  # El resultado de la funci n es: sigma_est (que es un escalar)
  return(sigma_est)
}

# Construcci n del estimador para la matriz \Hat{Q}_{XX}^{-1}: (Funci n auxiliar)
estimador_Q_xx_inv = function(X){
  n = nrow(X) # donde n es el tama o de la muestra
  Q = solve((1/n) * (t(X) %*% X)) # Q es el estimador que se est  buscando
  # Q es una matriz de tama o k x k, donde k es en n mero de par metros a estimar
  # La funci n me retorna a una matriz Q de tama o k x k
  return(Q)
}

# Construcci n del estimador de la  matriz de varianzas y covarianzas
# Huber-White: (Funci n auxiliar)
Huber_White = function(y, X, beta_est){
  # Variables:
  ## y: es el vector que contiene la variable dependiente
  ## X: es la matriz que contiene las variables regresoras
  ## beta_est: es el vector de los par metros estimados por OLS
  e = y - X %*% beta_est # e es el vector de residuales
  k = ncol(X)  # El n mero de par metros a estimar
  n = length(y)  # El n mero de observaciones en la mustra
  # Construcci n de (X^{'}X)^{-1}
  mat_X = solve(t(X) %*% X)
  # Matriz intermedia de los errores robustos White (que se encuentra por medio de una suma)
  # Inicializo la matriz como una matriz de ceros
  mat_intermedia = matrix(0, nrow = k, ncol = k)
  # El for se dise a para llenar la matriz mat_intermedia
  for (i in 1:n){
    mat_intermedia = mat_intermedia + (e[i])^2 * (X[i, ] %*% t(X[i, ]))
  }
  # mat_final me dal estimador de la matriz de varianzas y covarianzas de Huber-White
  mat_final = mat_X %*% mat_intermedia %*% mat_X
  return(mat_final)
}

# y = rnorm(1000)
# D = rnorm(1000, mean = 2)
# beta = c(1, 2)
# X = ols_X(D)
#
# Huber_White(y, X, beta)


# l estimador para la matriz \Hat{Q}_{XX}^{-1}: (Funci n auxiliar)

##_________________________________________________________________________
# delta: Es la funci n principal (m s importante) de todo el c digo.
#        Con base en la funci n delta es que se deduce todo lo dem s
#        en el c digo.
#        Es una funci n bastante general que contempla todas las situaciones
#        que puedan surgir en el c digo. En ese caso, contempla la simulaci n
#        de la Cauchy y de una muestra heteroced stico donde la volatilidad
#        de los tratados es diferente a la volatilidad de los no tratados
# Nota:  Revisar toda el enunciado del ejercicio primero para entender mejor
#        la l gica de la funci n
##_________________________________________________________________________

# Funci n para estimar el efecto tratamiento (delta) en una base de datos
delta = function(t, n, distro, hetero = F, varianza_y0 = 1, varianza_y1 = 1, robustos = F){
  # Defino la semilla para reproducibilidad del resultado
  set.seed(5678)
  # Definici n de variables:
  ## t: n mero de muestras a simular
  ## n: tama o de las muestras
  ## distro: tipo de distribuci n con el que se va a simular Y_{i}^{0}
  ## Nota: Si distro == "normal", entonces los par metros opcionales
  ##       empiezan a tomar relevancia
  ## hetero: Para saber si la volatilidad entre tratados y no tratados es diferente
  ## varianza_y0: El valor de la varianza de la distribuci n normal con la que se simula y_{i}^{0}
  ## varianza_y1: El valor de la varianza de la distribuci n normal con la que se simula y_{i}^{1}
  ## robustos: Solo se activa si se escoge la opci n hetero == T y adem s robustos == T
  ##          Permite estimar errores robustos usando la matriz de varianzas y covarianzas Huber-White
  # Consideraciones adicionales:
  # Condici n l gica para saber si y_0 sigue una distribuci n normal est ndar o una distribuci n Cauchy est ndar
  if (distro == "normal"){
    # Hay dos posibles casos en caso de que la distribuci n sea normal o no:
    # 1. Homoced sticidad entre tratados y no tratados
    ## En caso que la volatilidad de y_{i}^{0} y y_{i}^{1} sea la misma hetero == F
    # 2. Heterocedastidad entre tratados y no tratados (diferente volatilidad dependiendo si fueron tratados o no)
    ## En caso que la volatilidad de y_{i}^{0} y y_{i}^{1} sea diferente hetero == T
```

```
    if (hetero == F){
      # df: Dataframe que almacena el SDO y el delta_estimado por OLS
      # El par metro delta es el  p a r a etro  asociado a la asignaci n a tratamiento (D)
      df = data.frame(SDO = double(), delta_est = double(), sigma = double(), var_tlc = double(), var_ols = double())
      # Meter un for para generar las t diferentes muestas
      for (muestra in 1:t){
        # Simulaci n de los outcomes potenciales y de la variable trataminto
        y_0 = rnorm(n, mean = 0, sd = 1) # Simulaci n del outcome potencial de ausencia de tratamiento
        y_1 = y_0 + 3  # Simulaci n del outcome potencial de presencia de tratamiento
        D = rbinom(n, 1, prob = 0.3) # Simulaci n de una variable Bernoulli con una probabilidad de  xito  de 0.3
        # Modelo causal de Rubin
        y = y_0 + (y_1 - y_0) * D    # y es un vector Nx1, donde n es el n mero de observaciones (tama o de muestra)
        # Corregir (Debo calcular es el SDO)
        SDO = SDO_calculo(y, D) # C lculo del SDO utlilizando la funci n SDO_calculo
        # Estimaci n del par metro delta mediante una regresi n lineal con constante
        # X es una matriz NxK, donde n es el n mero de observaciones y K el n mero de p a r a etros
        X = ols_X(D) # D es la variable tratamiento
        beta_ols = solve(t(X) %*% X) %*% t(X) %*% y # beta_ols es un vector de Kx1, donde K es el n mero de par metos
        # Ahora, se calcularon 3 par metros adicionales:
        # el estimador de sigma, el estimador de var(sqrt(n) (delta_{OLS} - delta) y el estimador de var(delta_{OLS}))
        ## estimador de sigma:
        sigma = estimador_sigma(y, X, beta_ols)
        ## estimador de \Hat{Q}_{XX}^{-1}:
        Q_mat = estimador_Q_xx_inv(X)
        ## estimador de var_tlc = var(sqrt(n) (delta_{OLS} - delta))
        var_tlc = sigma * Q_mat[[2,2]]
        ## estimador de var_ols = var(delta_{OLS})
        n = nrow(X)
        var_ols = 1/n * sigma * Q_mat[[2,2]]
        # Dataframe que contiene todos los par metros de inter s
        df[muestra, ] = c(SDO, beta_ols[[2,1]], sigma, var_tlc, var_ols)    # Extraigo el segundo par metro que es el par metro delta
      }
    }else{
      # df: Dataframe que almacena el SDO y el delta_estimado por OLS
      # El par metro delta es el  p a r a etro  asociado a la asignaci n a tratamiento (D)
      df = data.frame(SDO = double(), delta_est = double(), sigma = double(), var_ols = double())
      # Meter un for para generar las t diferentes muestas
      for (muestra in 1:t){
        # Simulaci n de los outcomes potenciales y de la variable trataminto
        D = rbinom(n, 1, prob = 0.3) # Simulaci n de una variable Bernoulli con una probabilidad de  xito  de 0.3
        y_0 = rnorm(n, mean = 0, sd = sqrt(varianza_y0)) # Simulaci n del outcome potencial de ausencia de tratamiento
        y_1 = rnorm(n, mean = 3, sd = sqrt(varianza_y1))  # Simulaci n del outcome potencial de presencia de tratamiento
        # Modelo causal de Rubin
        y = y_0 + (y_1 - y_0) * D    # y es un vector Nx1, donde n es el n mero de observaciones (tama o de muestra)
        # Corregir (Debo calcular es el SDO)
        SDO = SDO_calculo(y, D) # C lculo del SDO utlilizando la funci n SDO_calculo
        # Estimaci n del par metro delta mediante una regresi n lineal con constante
        # X es una matriz NxK, donde n es el n mero de observaciones y K el n mero de p a r a etros
        X = ols_X(D) # D es la variable tratamiento
        beta_ols = solve(t(X) %*% X) %*% t(X) %*% y # beta_ols es un vector de Kx1, donde K es el n mero de par metos
        if (robustos == F){
          # Ahora, se calcularon 3 par metros adicionales:
          # el estimador de sigma, el estimador de var(sqrt(n) (delta_{OLS} - delta) y el estimador de var(delta_{OLS}))
          ## estimador de sigma:
          sigma = estimador_sigma(y, X, beta_ols)
          ## estimador de \Hat{Q}_{XX}^{-1}:
          Q_mat = estimador_Q_xx_inv(X)
          ## estimador de var_ols = var(delta_{OLS})
          n = nrow(X)
          var_ols = 1/n * sigma * Q_mat[[2,2]]
          # Dataframe que contiene todos los par metros de inter s
          df[muestra, ] = c(SDO, beta_ols[[2,1]], sigma, var_ols)    # Extraigo el segundo par metro que es el par metro delta
        }else{
          # Sigma, es provisional, luego se retira porque no tiene sentido para un estimador con matriz de varianzas y covarianzas
          # con errores robustos Huber-White (Dado la heterocedasticidad de los errores)
          sigma = 0
          # C lculo errores robustos Huber-White (me genera una matriz k x k, donde k es el n mero de par metros)
          var_ols = Huber_White(y, X, beta_ols)[[2,2]] # Extraigo la componente 2 de la matriz de varianzas y covarianzas de Huber-White
          df[muestra, ] = c(SDO, beta_ols[[2,1]], sigma, var_ols)    # Extraigo el segundo par metro que es el par metro delta
        }
      }
      # Condicional final para eliminar la columna sigma si robustos == T
```

```r
      if (robustos == T){
        df = df %>%
          select(SDO, delta_est, var_ols)
      }
    }
  }else if (distro == "cauchy"){
    # df: Dataframe que almacena el SDO y el delta_estimado por OLS
    # El par metro delta es el p a r a etro asociado a la asignaci n a tratamiento (D)
    df = data.frame(SDO = double(), delta_est = double())
    for (muestra in 1:t){
      # Simulaci n de los outcomes potenciales y de la variable tratamiento
      y_0 = rcauchy(n, location = 0, scale = 1) # Simulaci n del outcome potencial de ausencia de tratamiento
      y_1 = y_0 + 3  # Simulaci n del outcome potencial de presencia de tratamiento
      D = rbinom(n, 1, prob = 0.3) # Simulaci n de una variable Bernoulli con una probabilidad de  xito  de 0.3
      # Modelo causal de Rubin
      y = y_0 + (y_1 - y_0) * D    # y es un vector Nx1, donde n es el n mero de observaciones (tama o de muestra)
      # Corregir (Debo calcular es el SDO)
      SDO = SDO_calculo(y, D) # C lculo del SDO utlilizando la funci n SDO_calculo
      # Estimaci n del par metro delta mediante una regresi n lineal con constante
      # X es una matriz NxK, donde n es el n mero de observaciones y K el n mero de  p a r a etros
      X = ols_X(D) # D es la variable tratamiento
      beta_ols = solve(t(X) %*% X) %*% t(X) %*% y # beta_ols es un vector de Kx1, donde K es el n mero de par metos
      df[muestra, ] = c(SDO, beta_ols[[2,1]])    # Extraigo el segundo par metro que es el par metro delta
    }
  }
  # La funci n delta retorna un df con el c lculo de la SDO
  # y el par metro estimado delta, que proviene de una switching regression
  return(df)
}

# Simulaci n
# delta(t, n10, distro = "normal") # Existe el riesgo de que haya
              # multicolinealidad perfecta cuando se usa n10 = 10
              # Puede generar un vector D = rep(0, times = 10)

delta20 = delta(t, n20, distro = "normal")
delta50 = delta(t, n50, distro = "normal")
delta100 = delta(t, n100, distro = "normal")
delta500 = delta(t, n500, distro = "normal")
delta1000 = delta(t, n1000, distro = "normal")

# II. ----

# n_vector es un vector que almacena los diferentes tama os de muestra
n_vector = seq(from = 20, to = 1000, by = 10)

# Funci n que genera un dataframe con el tama o de muestra,
# la media y la varianza del estimador de delta para cada
# tama o diferente de muestra

media_varianza_delta = function(t, n_vector, distro, hetero = F, varianza_y0 = 1, varianza_y1 = 1, robustos = F){
  # Variables:
  ## t: N mero de muestras que se van a generar por cada tama o de muestra
  ## n_vector: Variable que almacena los diferentes tama os de muestras
  ## distro, hetero, varianza_y0 y varianza_y1 son par metros definidos para la funci n delta
  # df: DataFrame que almacena el tama o de muestra,
  #     la media y la varianza del estimador de delta para cada tama o de muestra
  df = data.frame(tama o = double(), media = double(), varianza = double())
  # Llamo a la funci n delta para cada tama o diferente de muestra
  for (i in 1:length(n_vector)){
    n_muestra = n_vector[i]
    delta_n = delta(t, n_muestra, distro, hetero, varianza_y0, varianza_y1, robustos)$delta_est # delta_n es el vector de deltas por cada tama o de muestra
    df[i,] = c(n_muestra, mean(delta_n), var(delta_n))
  }
  return(df)
}

# Dataframe con el tama o de muestra y la media y varianza del
# estimador de delta para los diferentes tama os de muestra
media_varianza = media_varianza_delta(100, n_vector, distro = "normal"); glimpse(media_varianza)

# III. ----

grafica_propiedas = function(df, variable_y, y_intercepto = 0){
  variable_y = ensym(variable_y)
  graph = df %>%
    ggplot(aes(x = tama o , y = !!variable_y)) +
    geom_line(color = "green", size = 1) +
    geom_hline(yintercept = y_intercepto, color = "red") +
    theme_light()
  return(graph)
}

grafica_propiedas(media_varianza, media, y_intercepto = 3)
grafica_propiedas(media_varianza, varianza)
```

```r

# Ejercicio c. ----

# I. ----

# Definici n de par metros para la simulaci n

## N mero de muestras:
t2 = 1000

## Tama o de muestras:

# No se puede trabajar con n10 = 10 porque no satisface la condici n de rango
# n10 = 10 Genera probelmas de singularidad cuando se generan muchas muestras
#           Porque por chance, se genera una muestra donde D = rep(0, times = 10)
n20 = 20
n100 = 100
n1000 = 1000
n5000 = 5000
n10000 = 10000
n20000 = 20000
n50000 = 50000
n100000 = 100000
n_vector2 = c(n20, n100, n1000, n5000, n10000, n20000, n50000, n100000)

# Funci n que calcula a = sqrt(n) (delta_est - truth_delta) para cada tama o de muestra n
calculo_a = function(t, n, distro, truth_delta){
  # Variables:
  ## t: n mero de muestras a simular
  ## n: tama o de la muestra
  ## truth_delta: verdadero valor del par metro delta (valor poblcional del par metro)
  # df_estimados es el dataframe que contiene el delta estimado por SDO
  # y por medio de regresi n
  df_estimados = delta(t, n, distro)
  # df_a es el dataframe que contiene
  df_a = df_estimados %>%
    mutate(a = sqrt(n) * (delta_est - truth_delta), tama o = rep(n, times = t))
  return(df_a)
}

# Defino una funci n para crear un dataframe que se va a utilizar
# Para construir la gr fica multipanel.
base_para_graficas = function(t, n_vector, distro, truth_delta){
  for (i in 1:length(n_vector)){
    # Si i == 1 significa que estamos en la primera iteraci n
    if (i == 1){
      df_a_total = calculo_a(t, n_vector[i], distro, truth_delta = 3)
    }else{
      df_prov = calculo_a(t, n_vector[i], distro, truth_delta = 3)
      df_a_total = bind_rows(df_a_total, df_prov)
    }
  }
  # Retorna un dataframe con los diferentes valores de a para cada tama o de muestra listo
  # para construir la gr fica multipanel
  return(df_a_total)
}

# La base multipanel me tiene lista los diferentes a
# para cada tama o de muestra
base_multipanel = base_para_graficas(t2, n_vector2, distro = "normal", truth_delta = 3)

# II. ----

# Defino una funci n para crear la gr fica multipanel con el histograma y la funci n de densidad
# para cada tama o de muestra n
histogram_grid = function(df, titulo, x_lab, y_lab, num_bins = 30, y_upper_limit){
  histog_grid = df %>%
    ggplot(aes(a)) +
    scale_y_continuous(limits = c(0, y_upper_limit)) +
    geom_histogram(aes(y = ..density..), color = "black", bins = num_bins) +
    geom_density(color = "green") +
    facet_grid(cols = vars(tama o)) +
    ggtitle(titulo) +
    xlab(x_lab) +
    ylab(y_lab) +
    theme_light()
}

grafica_multi = histogram_grid(df = base_multipanel, titulo = "Gr fico multipanel para a con diferentes tama os de muestra", x_lab = "Tama o de muestras (n)", y_lab = "", num_bins = 30, y_upper_limit = 0.2); grafica_multi

# Gr fica que compara directamente las densidades
compracion_densidades_grafica = function(t, df, var){
  # Variables:
  ## t: n mero de muestras simuladas por tama o de muestra
  ## df: df multipanel que tiene las simulaciones de las diferentes muestra por tama o de muestra
  ## var: variable simulada de la cual se va a generar las funciones de densidad
  # Nota:
  # Modifico el dataframe multipanel para seleccionar solo
  # las variables a y tama o que permiten hacer la gr fica de densidades
  # df_mod es un dataframe modifiado del dataframe multipanel
```

```r
  df_mod = df %>%
    rename(valores_sim = {{ var }}, distribuciones = tama o) %>%
    mutate(distribuciones = as.character(distribuciones)) %>%
    select(valores_sim, distribuciones)
  # Simulo de una distribuci n normal est ndar porque quiero probar
  # si se cumple o no el teorema del l mite central
  df_normal = data.frame(valores_sim = rnorm(t), distribuciones = rep("Normal est ndar", times = t))
  # base_grafica ya es la base de datos lista para realizar las gr ficas de
  # las densidades de las a y de una normal est ndar
  base_grafica = df_mod
  # base_grafica = bind_rows(df_mod, df_normal)
  # Gr fica de densidades
  density_comparacion = base_grafica %>%
    ggplot(aes(x = valores_sim, color = distribuciones)) +
    geom_density() +
    theme_light() +
    ggtitle("Comparaci n de funciones de densidad de a\n para muestras de diferentes tama os \n(todas simuladas)") +
    ylab("Densidades") +
    xlab("Distribuciones de a para diferente tama o de muestra")
  return(density_comparacion)
}

# Visualizaci n de la base de datos con las variables simuladas
# delta_est y a
glimpse(base_multipanel)

base_filtrada = base_multipanel %>%
  filter(100 < tama o)

base_filtrada_a = base_multipanel %>%
  filter(10 < tama o) %>%
  filter(tama o < 2000)

# Gr ficas de las funciones de densidad para la simulaci n de los delta estimados
# y de los a, para el caso de un Y_{i}^{0} = norm(0, 1)
grafica_a = compracion_densidades_grafica(t = t2, df = base_filtrada_a, var = a); grafica_a
grafica_delta_est = compracion_densidades_grafica(t = t2, df = base_filtrada, var = delta_est); grafica_delta_est

# Nota: Crear la otra gr fica mejor (script de simulaci n de df)

# Ejercicio d. ----

# Va a replicarse el ejercicio c. pero asumiendo que:
# Y_{i}^{0} ~ cauchy(0, 1) (Distribuci n Cauchy est ndar)

# I. ----

# Construyo la base multipanel para los diferentes a, asumiendo que Y_{i}^{0} sigue una distribuci n Cauchy

base_multipanel_cauchy = base_para_graficas(t2, n_vector2, distro = "cauchy", truth_delta = 3)

# II. ----

base_filtrada_cauchy = base_multipanel_cauchy %>%
  filter(tama o > 20)
  # filter(tama o < 10000)

base_filtrada_cauchy2 = base_multipanel_cauchy %>%
  filter(tama o > 10) %>%
  filter(tama o < 2000)

# Gr ficas de las funciones de densidad para la simulaci n de los delta estimados
# y de los a, para el caso de un Y_{i}^{0} = cauchy(0, 1)
grafica2_a = compracion_densidades_grafica(t = t2, df = base_filtrada_cauchy2, var = a); grafica2_a
grafica2_delta_est = compracion_densidades_grafica(t = t2, df = base_filtrada_cauchy, var = delta_est); grafica2_delta_est

# Nota: Super interesante ver como cambia la distribuci n Cauchy
#       bajo los mismos par metros de localizaci n y escala
#       La distribuci n Cauchy es una distribuci n patol gica
#       en el sentido que su media y varianza es infinita

# plot(density(rcauchy(1000, location = 0, scale = 1)))

# Ejercicio e. ----

# I. ----

# Tama os de muestra
n20 = 20
n100 = 100
n200 = 200
n500 = 500
n1000 = 1000
n10000 = 10000

hetero20_no_robustos = delta(t = 1000, n = 20, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F)
```

```r
hetero100_no_robustos = delta(t = 1000, n = 100, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F)
hetero200_no_robustos = delta(t = 1000, n = 200, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F)
hetero500_no_robustos = delta(t = 1000, n = 500, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F)
hetero1000_no_robustos = delta(t = 1000, n = 1000, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F)
hetero10000_no_robustos = delta(t = 1000, n = 10000, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F)

n_vect_hetero = c(n20, n100, n200, n500, n1000, n10000)

# Dataframe con el tama o de muestra y la media y varianza del
# estimador de delta para los diferentes tama os de muestra
media_varianza_hetero = media_varianza_delta(t = 1000, n_vect_hetero, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = F); glimpse(media_varianza_hetero)

# Nota: Se observa que a pesar de la heterocedasticidad en Y_{i} hay insesgadez
#       En la estimaci n de delta
#       De igual forma, la varianza muestral tambi n disminuye a medida que aumenta
#       la muestra (independiente de que haya heteroecedasticidad en Y_{i})

# Construir una funci n que me compute los intervalos de confianza
# Intervalos de confianza: Ya sea para intervalos cl sicos o para intervalos robustos
#                          a la heterocedasticidad como los calculados por la matriz de varianzas y covarianzas Huber-White

inter_confianza = function(df_estimaciones, delta_true, int_conf){
  norm_inf = qnorm((1 - int_conf)/2)
  norm_sup = qnorm((1 - int_conf)/2 + int_conf)
  lim_inf = df_estimaciones$delta_est + norm_inf * sqrt(df_estimaciones$var_ols)
  lim_sup = df_estimaciones$delta_est + norm_sup * sqrt(df_estimaciones$var_ols)
  contiene_o_no = c()
  for (i in 1:length(lim_inf)){
    if ((lim_inf[i] < delta_true) && (delta_true < lim_sup[i])){
      contiene_o_no = append(contiene_o_no, 1)
    }else{
      contiene_o_no = append(contiene_o_no, 0)
    }
  }
  # Variable del dataframe
  df = data.frame(lim_inf, lim_sup, contiene_o_no)
  return(df)
}

# Intervalos de confianza del 90 %
int_conf1000_no_robustos_90 = inter_confianza(hetero1000_no_robustos, delta_true = 3, int_conf = 0.9)

# Intervalos de confianza del 95 %
int_conf1000_no_robustos_95 = inter_confianza(hetero1000_no_robustos, delta_true = 3, int_conf = 0.95)

# Intervalos de confianza del 99 %
int_conf1000_no_robustos_99 = inter_confianza(hetero1000_no_robustos, delta_true = 3, int_conf = 0.99)

# II. ----

# Porcentaje de los intervalos de confianza del 90 % que contienen el par metro verdadero
porcentaje_no_robustos_90 = sum(int_conf1000_no_robustos_90$contiene_o_no)/nrow(int_conf1000_no_robustos_90) * 100; porcentaje_no_robustos_90

# Porcentaje de los intervalos de confianza del 95 % que contienen el par metro verdadero
porcentaje_no_robustos_95 =  sum(int_conf1000_no_robustos_95$contiene_o_no)/nrow(int_conf1000_no_robustos_95) * 100; porcentaje_no_robustos_95

# Porcentaje de los intervalos de confianza del 99 % que contienen el par metro verdadero
porcentaje_no_robustos_99 = sum(int_conf1000_no_robustos_99$contiene_o_no)/nrow(int_conf1000_no_robustos_99) * 100; porcentaje_no_robustos_99

# III. ----

hetero20_robustos = delta(t = 1000, n = 20, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero100_robustos = delta(t = 1000, n = 100, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero200_robustos = delta(t = 1000, n = 200, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero500_robustos = delta(t = 1000, n = 500, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero1000_robustos = delta(t = 1000, n = 1000, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)
hetero10000_robustos = delta(t = 1000, n = 10000, distro = "normal", hetero = T, varianza_y0 = 2, varianza_y1 = 1, robustos = T)

n_vect_hetero = c(n20, n100, n200, n500, n1000, n10000)

# Dataframe con el tama o de muestra y la media y varianza del
# estimador de delta para los diferentes tama os de muestra
```

```r
media_varianza_hetero = media_varianza_delta(t = 1000, n_vect_hetero, distro = "normal", hetero = T
    , varianza_y0 = 2, varianza_y1 = 1, robustos = T); glimpse(media_varianza_hetero)

# III. C lculo de intervalos de confianza cuando se tienen errores robustos ----

# Intervalos de confianza del 90 %
int_conf1000_robustos_90 = inter_confianza(hetero1000_robustos, delta_true = 3, int_conf = 0.9)

# Intervalos de confianza del 95 %
int_conf1000_robustos_95 = inter_confianza(hetero1000_robustos, delta_true = 3, int_conf = 0.95)

# Intervalos de confianza del 99 %
int_conf1000_robustos_99 = inter_confianza(hetero1000_robustos, delta_true = 3, int_conf = 0.99)

# III. Porcentaje de los intervalos de confianza cuando se tienen errores robustos ----

# Porcentaje de los intervalos de confianza del 90 % que contienen el par metro verdadero
porcentaje_robustos_90 = sum(int_conf1000_robustos_90$contiene_o_no)/nrow(int_conf1000_robustos_90)
    * 100; porcentaje_robustos_90

# Porcentaje de los intervalos de confianza del 95 % que contienen el par metro verdadero
porcentaje_robustos_95 = sum(int_conf1000_robustos_95$contiene_o_no)/nrow(int_conf1000_robustos_95)
    * 100; porcentaje_robustos_95

# Porcentaje de los intervalos de confianza del 99 % que contienen el par metro verdadero
porcentaje_robustos_99 = sum(int_conf1000_robustos_99$contiene_o_no)/nrow(int_conf1000_robustos_99)
    * 100; porcentaje_robustos_99

# Conclusi n: Los errores robustos c lculados con matriz de varianzas y covarianzas Huber-White,
#              s   sirven cu ndo hay heterocedasticidad en los errores de modelo
#             Por el contrario, si se utilizan intervalos de confianza cl sicos
#             sin corregir por heterocedasticidad, se observa que los intervalos
#             de confianza cl sicos de manera sistem tica para los niveles de
#             significancia sobreestiman, es decir calculan intervalos de
#             confianza m s amplios de los que en realidad deberian computarse
#             dada la presencia de la heterocedasticad
```