

Inferencia estadística robusta a *clusters* en Stata

Eduardo García Echeverri

Seminario web de Stata, Julio 2024

- 1 ¿Cómo afectan los *clusters* a la inferencia estadística?
- 2 El cluster-robust variance estimator (CRVE)
- 3 Alternativas cuando los supuestos del CRVE no se cumplen
 - Ajustar los grados de libertad
 - Wild cluster bootstrap
- 4 Conclusión

Outline

- 1 ¿Cómo afectan los *clusters* a la inferencia estadística?
- 2 El cluster-robust variance estimator (CRVE)
- 3 Alternativas cuando los supuestos del CRVE no se cumplen
 - Ajustar los grados de libertad
 - Wild cluster bootstrap
- 4 Conclusión

¿Qué son los errores *clusterizados*?

Definición:

Observaciones se pueden dividir en grupos (**clusters**) y el **outcome** (o el tratamiento) están **correlacionados dentro de cada grupo**.

Errores *clusterizados* en tus datos

1. Datos de **firmas** operando en diferentes sectores:
 - Errores **correlacionados dentro de industrias**.
2. Datos de **estudiantes de secundaria** en los EU:
 - Errores **correlacionados dentro de colegios**.
3. **Datos panel** sobre individuos:
 - Errores de cada individuo **correlacionados en el tiempo**.
4. Experimentos con **tratamiento a nivel agregado**.
 - Errores **correlacionados dentro de cada nivel**.

Ejemplo: Estados que cambian el salario mínimo.

Modelo lineal con errores *clusterizados*

Considere el modelo:

$$y_{ig} = X_{ig}\beta + \varepsilon_{ig}$$

donde,

y_{ig} : **outcome** de la observación i en el cluster g ;

X_{ig} : vector de **regresores** de la observación i en el cluster g ;

ε_{ig} : **término de error** de la observación i en el cluster g ;

β : **coeficientes** de interés;

$g = 1, 2, \dots, G$.

Errores *clusterizados*

Errores **en diferentes clusters** no están **correlacionados**:

$$\text{Cor}(\varepsilon_{ig}, \varepsilon_{j\tilde{g}}) = 0$$

Errores **dentro del mismo cluster** pueden estar **correlacionados**:

$$\text{Cor}(\varepsilon_{ig}, \varepsilon_{jg}) \neq 0$$

Entonces, estamos relajando el supuesto de **errores i.i.d.**

Los errores *clusterizados* complican la inferencia

Problema: IC's que suponen i.i.d. no tienen cobertura correcta

- **Cobertura** es (típicamente) **menor al 95%**;
- **Errores estándar** son (típicamente) **muy pequeños**;
- Puede llevar a **rechazar más de la cuenta la hipótesis nula** (falsos positivos).

Veamos esto en unas **simulaciones de Monte Carlo**.

Diseño experimental lineal

Proceso generador de datos:

$$y_{ig} = 1 + x_{ig} + z_{ig} + T_g + \mu_{ig} + \nu_g$$

donde,

y_{ig} : **outcome** de la observación i en el cluster g ;

x_{ig}, z_{ig} : variables de **control**, $N(0, 3)$ y $\chi^2(7)$ respectivamente;

Obs = 1000;

g : **observaciones asignadas al azar** entre **100 clusters**;

T_g : **33 clusters asignados al azar a tratamiento** ($T_g = 1$).

Diseño experimental lineal – errores *clusterizados*

Proceso generador de datos:

$$y_{ig} = 1 + x_{ig} + z_{ig} + T_g + \mu_{ig} + \nu_g$$

donde,

ν_g : **componente clusterizado** del **término de error**, $N(0, 0.5)$;

μ_{ig} : **componente individual** del **término de error**, $N(0, 0.5)$.

⇒ Errores están **correlacionados dentro de los clusters**.

Simulaciones de Monte Carlo

Procedimiento:

1. **Simular** el PGD
2. `regress y x z treat`
3. **Guardar el coeficiente** de `treat`: `[beta]`
4. **Revisar si IC** de `treat` **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1 estaba contenido en el IC**.

Resultados – Simulaciones de Monte Carlo

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	1.008647	.2367399	.1830714	1.775514
contained	1,000	.564	.4961352	0	1

Notas:

1. **Cobertura** de solo **56.4%** (vs. 95% significancia nominal)
2. Estimador **sigue siendo consistente**.
3. **IC's** son **demasiado angostos**.

Controlar por los clusters no es la solución

Procedimiento:

1. **Simular** el PGD
2. regress y x z treat *i.cvar*
3. **Guardar el coeficiente** de treat: [beta]
4. **Revisar si IC** de treat **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1 estaba contenido en el IC**.

Resultados – Monte Carlo con variables dummy por cluster

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	1.01037	1.482467	-3.426341	5.811284
contained	1,000	.471	.4994081	0	1

Notas:

1. **Cobertura** cae al 47.1% (vs. 95% significancia nominal)
2. **Ineficiencia:** coeficientes de **variables irrelevantes**.

Diseño experimental no lineal

Proceso generador de datos (modelo Probit):

$$y_{ig}^* = 1 + x_{ig} + z_{ig} + T_g - \mu_{ig} - \nu_g$$

$$y_{ig} = 1\{y_{ig}^* \geq 0\}$$

donde,

y_{ig} : **outcome** de la observación i en el cluster g ;

x_{ig}, z_{ig} : variables de **control**, $N(0, 3)$ y $\chi^2(7)$ respectivamente;

Obs = 10000;

g : **observaciones asignadas al azar** entre **100 clusters**;

T_g : **33 clusters asignados al azar a tratamiento** ($T_g = 1$).

Diseño experimental no lineal – errores *clusterizados*

Proceso generador de datos:

$$y_{ig}^* = 1 + x_{ig} + z_{ig} + T_g - \mu_{ig} - \nu_g$$

$$y_{ig} = 1\{y_{ig}^* \geq 0\}$$

donde,

ν_g : **componente clusterizado del término de error**, $N(0, 0.5)$;

μ_{ig} : **componente individual del término de error**, $N(0, 0.5)$.

Simulaciones de Monte Carlo

Procedimiento:

1. **Simular** el PGD
2. `probit y x z treat`
3. **Guardar el coeficiente** de `treat`: `[beta]`
4. **Revisar si IC** de `treat` **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1 estaba contenido en el IC**.

Resultados – Monte Carlo experimento no lineal

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	1.022974	.1905835	.3237958	1.550756
contained	1,000	.558	.4968731	0	1

Notas:

1. **Cobertura** de solo **55.8%** (vs. 95% significancia nominal)
2. Estimador **sigue siendo consistente**.
3. **IC's** son **demasiado angostos**.

Soluciones

1. Cluster-robust variance estimator (CRVE):

- Opción `vce(cluster cvarlist)`
Liang and Zeger (1986)
- Ajustar grados de libertad: `vce(hc2 cvar, dfadjust)` **[Stata 18]**
Bell and McCaffrey (2002)

2. Wild cluster bootstrap **[Stata 18]**

Cameron, Gelbach, and Miller (2008)

Outline

- 1 ¿Cómo afectan los *clusters* a la inferencia estadística?
- 2 El cluster-robust variance estimator (CRVE)
- 3 Alternativas cuando los supuestos del CRVE no se cumplen
 - Ajustar los grados de libertad
 - Wild cluster bootstrap
- 4 Conclusión

Cluster-robust variance estimator (CRVE)

IC's pueden **corregirse** usando el **CRVE**:

$$\hat{V} = \frac{G(N-1)}{(G-1)(N-k)} (XX')^{-1} \left(\sum_{g=1}^G X'_g \hat{\epsilon}_g \hat{\epsilon}'_g X_g \right) (XX')^{-1}$$

IC corregido de 95% para β_k : $\left[\hat{\beta}_k - 1.96\sqrt{\hat{V}_{k,k}}, \hat{\beta}_k + 1.96\sqrt{\hat{V}_{k,k}} \right]$

Implementación – opción `vce(cluster)`

Ejemplo:

```
estimation_command ..., vce(cluster cvarlist)
```

Revisar disponibilidad:

```
help estimation_command
```

Simulaciones de Monte Carlo – Diseño experimental lineal

$$y_{ig} = 1 + x_{ig} + z_{ig} + T_g + \mu_{ig} + \nu_g$$

Procedimiento:

1. **Simular** el PGD
2. regress y x z treat, `vce(cluster cvar)`
3. **Guardar el coeficiente** de treat: [beta]
4. **Revisar si IC** de treat **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1 estaba contenido en el IC**.

Resultados – Monte Carlo con CRVE

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	.9974977	.2236554	.1211772	1.817533
contained	1,000	.959	.1983894	0	1

Notas:

1. **Cobertura** del 95.9% (vs. 95% significancia nominal)

Simulaciones de Monte Carlo – Diseño experimental Probit

$$y_{ig}^* = 1 + x_{ig} + z_{ig} + T_g - \mu_{ig} - \nu_g$$
$$y_{ig} = 1\{y_{ig}^* \geq 0\}$$

Procedimiento:

1. **Simular** el PGD
2. `probit y x z treat, vce(cluster cvar)`
3. **Guardar el coeficiente** de `treat`: `[beta]`
4. **Revisar si IC** de `treat` **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1 estaba contenido en el IC**.

Resultados – Monte Carlo Probit CRVE

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	1.022447	.1861081	.3533935	1.591309
contained	1,000	.929	.2569534	0	1

Notas:

1. **Cobertura** del 92.9% (vs. 95% significancia nominal)

CRVE típicamente aumenta los EE, mejorando cobertura

Ejemplo: Regresión lineal **con y sin CRVE** (wagework.dta)

	wage_CRVE	wage_iid
Job tenure	0.609 (0.020)	0.609 (0.016)
Labor-market condition	-0.049 (0.029)	-0.049 (0.030)
Age in years	0.198 (0.007)	0.198 (0.005)
Intercept	13.506 (0.284)	13.506 (0.212)
Number of observations	1928	1928

Limitaciones del CRVE

El **CRVE** puede funcionar bien, pero esto depende de G .

El **CRVE** usualmente no funciona bien cuando:

1. El número de clusters **G** es pequeño.
2. Los cluster tienen **tamaños muy diferentes**.

De nuevo, veámoslo en una simulación de **Monte Carlo**.

Diseño experimental lineal con pocos clusters

Proceso genrados de datos:

$$y_{ig} = 1 + x_{ig} + z_{ig} + T_g + \mu_{ig} + \nu_g$$

donde,

y_{ig} : **outcome** de la obervación i en el cluster g ;

x_{ig}, z_{ig} : variables de **control**, $N(0, 3)$ y $\chi^2(7)$ respectivamente;

Obs = 1000;

g : **observaciones asignadas al azar** entre **21 clusters**;

T_g : **7 clusters** asignados al azar a **tratamiento** ($T_g = 1$).

Resultados – Monte Carlo con pocos clusters

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	1.021723	.8693036	-1.44612	3.697521
contained	1,000	.879	.3262905	0	1

Notas:

1. **Cobertura** del 87.9% (vs. 95% significancia nominal)

Outline

- 1 ¿Cómo afectan los *clusters* a la inferencia estadística?
- 2 El cluster-robust variance estimator (CRVE)
- 3 Alternativas cuando los supuestos del CRVE no se cumplen
 - Ajustar los grados de libertad
 - Wild cluster bootstrap
- 4 Conclusión

Solución 1: Ajustar los grados de libertad (DoF)

Bell and McCaffrey (2002): **ajustar DoF** basados en *cvar*

- **Mejora cobertura del IC** cuando hay pocos clusters.

Implementación:

```
estimation_command ..., vce(hc2 cvar, dfadjust)
```

hc2: implementa un estimador de los residuos **más conservador**.

Revisar disponibilidad:

```
help estimation_command
```

Veámoslo en acción en una **Monte Carlo simulation**.

De vuelta al experimento lineal (7 clusters)

$$y_{ig} = 1 + x_{ig} + z_{ig} + T_g + \mu_{ig} + \nu_g$$

Procedimiento:

1. **Simular** el PGD
2. regress y x z treat, `vce(hc2 cvar, dfadjust)`
3. **Guardar el coeficiente** de treat: [beta]
4. **Revisar si IC** de treat **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1** estaba contenido en el IC.

Resultados – Monte Carlo pocos clusters, DoF ajustados

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	1.040855	.835802	-1.290347	4.12301
contained	1,000	.978	.1467567	0	1

Notes:

1. **Cobertura** del 97.8% (vs. 95% de significancia nominal)

Solution 2: El comando wildbootstrap

Syntax:

wildbootstrap *estimator depvar [indepvars] [if] [in] [weight] [, options]*

estimator:

- regress
- areg
- xtreg (solo modelo **fixed-effects**; opción fe puede omitirse)

Quick Start

Estimar por WCB el p-valor y el IC para el coeficiente de x_1 en la regresión lineal de y sobre x_1 con clusters definidos por $cvar$

`wildbootstrap regress y x1, cluster(cvar)`

El algoritmo Wild cluster bootstrap

Considere el modelo:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_G \end{bmatrix} = X\beta + \epsilon = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_G \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_G \end{bmatrix}$$

where,

y_g : vector de **outcomes** del cluster g ,

X_g : vector de **regresores** del cluster g ,

ϵ_g : vector de **errores** del cluster g ,

β : **coeficientes** de interés.

Wild cluster restricted bootstrap (WCRB)

El **algoritmo WCRB** consiste en 4 pasos:

Suponga, por ejemplo, que queremos testear $\mathbb{H}_0 : \beta_k = 0$.

1. **Re-estimar** el modelo linal **con la restricción** $\beta_k = 0$.

$\tilde{\beta}$: coeficientes estimados

$\tilde{\epsilon}$: residuos estimados

Algoritmo WCRB

2. Crear una **replicación de bootstrap b** (repetir B veces):

2.1 Generar una **variable aleatoria** ν_g^b para cada cluster g :

E.g. rademacher: -1 ó $+1$ con igual probabilidad

2.2 Generar una **nueva variable dependiente** y_{ig}^b :

$$y_{ig}^b = X_{ig}\tilde{\beta} + \tilde{\epsilon}_{ig}\nu_g^b.$$

2.3 **Re-estimar el modelo** usando la variable y_{ig}^b .

2.4 Calcular el estadístico \mathbf{t} , $t_k^b = \frac{\hat{\beta}_k^b}{\sqrt{\hat{V}_{k,k}^b}}$

$\hat{\beta}_k^b$: **coeficiente estimado** de la replicación bootstrap.

$\hat{V}_{k,k}^b$: **CRVE** del k -ésimo coeficiente de la replicación bootstrap.

Algoritmo WCRB

Suponga que $\mathbb{H}_A : \beta_k \neq 0$.

3. Calcular los p-valores según la distribución del estadístico t :

- **Simétrica** en 0: $p_S = \frac{1}{B} \sum_{b=1}^B I(|t_k^b| > |t_k|)$

t_k : estadístico t original

- **Asimétrica**: $p_e = 2 \min(p_1, p_2)$

p_1, p_2 : p-valores de bootstrap usando **hipótesis alternativas de un lado**.

4. Calcular IC's **invirtiendo el test**. Encontrar el estadístico t tal que el p -valor es 0.05.

De vuelta al experimento lineal (7 clusters)

$$y_{ig} = 1 + x_{ig} + z_{ig} + T_g + \mu_{ig} + \nu_g$$

Procedimiento:

1. **Simular** el PGD
2. `wildbootstrap` `reg y x z treat, cluster(cvar)`
3. **Guardar el coeficiente** de `treat`: `[beta]`
4. **Revisar si IC** de `treat` **contiene 1**
5. **Repetir 1000 veces** los pasos 1-4.
6. **Contar** cuántas veces **1 estaba contenido en el IC**.

Resultados – Wild cluster bootstrap

Variable	Obs	Mean	Std. dev.	Min	Max
beta	1,000	.9961465	.4699228	-1.036362	2.653632
contained	1,000	.957	.2029586	0	1

Notas:

1. **Cobertura** del 95.7% (vs. 95% significancia nominal)

Ejemplo 1: Regresión lineal simple

```
. use https://www.stata-prepress.com/data/r18/wagework, replace
(Wages for 20 to 77 year olds, 2013-2016)
```

```
. wildbootstrap regress wage tenure, cluster(personid) rseed(12345)
```

Performing 1,000 replications for p-value for tenure = 0 ...

Computing confidence interval for tenure

Lower bound:10.....20. done (21)

Upper bound:10..... done (17)

Wild cluster bootstrap

Linear regression

Number of obs = 1,928

Number of clusters = 589

Cluster size:

Cluster variable: personid

min = 1

Error weight: Rademacher

avg = 3.3

max = 4

	wage	Estimate	t	p-value	[95% conf. interval]	
constraint						
	tenure = 0	.7807403	27.19	0.000	.7209754	.8368386

Ejemplo 1: Usando en vez el CRVE

```
. regress wage tenure, vce(cluster personid)
```

Linear regression

```
Number of obs   =    1,928
F(1, 588)       =    739.36
Prob > F        =    0.0000
R-squared       =    0.4212
Root MSE      =    3.5097
```

(Std. err. adjusted for 589 clusters in personid)

wage	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
tenure	.7807403	.028713	27.19	0.000	.7243477	.8371328
_cons	20.89884	.2135686	97.86	0.000	20.47939	21.31829

Notas: G es alto y clusters son homogéneos, $CI_{CRVE} \approx CI_{WCRB}$

Ejemplo 2: Pocos clusters heterogéneos

```
. use https://www.stata-press.com/data/r18/nlsw88, replace
(NLSW, 1988 extract)
```

```
. wildbootstrap regress wage tenure, cluster(industry) rseed(12345)
```

Performing 1,000 replications for p-value for tenure = 0 ...

Computing confidence interval for tenure

Lower bound:10.....20..... done (26)

Upper bound:10.....20.... done (24)

Wild cluster bootstrap

Linear regression

Number of obs = 2,217

Number of clusters = 12

Cluster size:

Cluster variable: industry

min = 4

Error weight: Rademacher

avg = 184.8

max = 817

	wage	Estimate	t	p-value	[95% conf. interval]	
constraint						
	tenure = 0	.1830716	6.95	0.000	.1274023	.3258156

Ejemplo 2: Usando en vez el CRVE

```
. regress wage tenure, vce(cluster industry)
```

Linear regression

```
Number of obs   =      2,217
F(1, 11)        =      48.30
Prob > F        =      0.0000
R-squared       =      0.0305
Root MSE      =      5.6853
```

(Std. err. adjusted for 12 clusters in industry)

wage	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
tenure	.1830716	.026341	6.95	0.000	.1250953	.2410478
_cons	6.710915	.4147967	16.18	0.000	5.797954	7.623877

Remark: Small G and clusters of dissimilar size, $CI_{CRVE} \neq CI_{WCRB}$

Example 3: Dos regresores

```
. wildbootstrap regress wage tenure age, cluster(industry) rseed(12345)

Performing 1,000 replications for p-value for tenure = 0 ...
Computing confidence interval for tenure
  Lower bound: .....10.....20.. done (22)
  Upper bound: .....10.....20.... done (24)

Performing 1,000 replications for p-value for age = 0 ...
Computing confidence interval for age
  Lower bound: .....10.....20.... done (24)
  Upper bound: .....10.....20..... done (27)

Wild cluster bootstrap
Linear regression
Cluster variable: industry
Error weight: Rademacher

Number of obs      = 2,217
Number of clusters = 12
Cluster size:
min = 4
avg = 184.8
max = 817
```

	wage	Estimate	t	p-value	[95% conf. interval]	
constraints						
	tenure = 0	.1869715	7.18	0.000	.13478	.3280472
	age = 0	-.0946592	-2.55	0.006	-.2232396	-.0279091

Outline

- 1 ¿Cómo afectan los *clusters* a la inferencia estadística?
- 2 El cluster-robust variance estimator (CRVE)
- 3 Alternativas cuando los supuestos del CRVE no se cumplen
 - Ajustar los grados de libertad
 - Wild cluster bootstrap
- 4 Conclusión

Conclusión

1. Es **crucial** ajustar los errores estándar cuando se trabaja con **datos clusterizados**.
 - **IC's** pueden ser **engañosos** de lo contrario
 - Especialmente en **experimentos** con **tratamiento por clusters**.
2. Cuando los clusters son **muchos y homogéneos**:
 - **CRVE**: `vce(cluster cvar)`
3. Cuando los clusters son **pocos y heterogéneos**:
 - **Ajustar DoF**: `vce(hc2 cvar, dfadjust)`
 - **Wild cluster bootstrap**: `wildbootstrap`

Aprende más...

1. Comando `help`
 - Acceso a toda nuestra **documentación**.
2. www.stata.com
 - Acceso a toda nuestra **documentación**;
 - **Preguntas frecuentes (FAQ)**.
3. www.youtube.com/@statacorp/featured
4. tech-support@stata.com
 - **Preguntas específicas** sobre el software.

Referencias

1. **When should you adjust standard errors for clustering?**
 - Abadie, Athey, Imbens, and Wooldridge (NBER, Working paper)
2. **How much should we trust differences-in-differences estimates?**
 - Esther Duflo (QJE, 2002)
3. **Bootstrap-based improvements for inference with clustered errors**
 - Cameron, Gelbach, and Miller (ReStat, 2007)
4. **Bias reduction in standard errors for linear regression with multi-stage samples**
 - Bell and McCaffrey (Survey Methodology, 2002)

Gracias!