



**Data for
Development Impact**

Programación 101

Rony Rodriguez-Ramírez

January 15, 2021

The World Bank | DIME

LAMBDA

Introducción al Curso

¿Quién soy?

- Nombre: Rony Rodrigo Maximiliano Rodríguez-Ramírez
- Trabajo: Banco Mundial, Washington, D.C.
 - Development Impact Evaluation Unit (DIME)
- Educación: Economista con máster en Development Policy (KDI School)
- Experiencia:
 - Innovations for Poverty Action (IPA) – Instituto Tecnológico Autónomo de México
 - Monash University (Australia)
 - KDI School of Public Policy and Management
- La mayoría de mi trabajo es evaluación de impacto de Randomized Controlled Trials en países en desarrollo (e.g., Rwanda, Liberia, etc.).

Organización del Curso

Enfoque del Curso

- ¿Por qué es importante aprender un software de análisis econométrico hoy en día (Stata, R, etc.)?
- Mejoramiento del conocimiento de programación aplicada a la econometría y la investigación.
- Buenas prácticas de programación de la Unidad de Evaluación de Impacto (DIME) del Banco Mundial.

Objetivos

Objetivos del Curso:

- Aprender los conceptos clave y las técnicas econométricas asociadas a la programación estadísticas usando Stata enfatizando la eficiencia de escribir códigos:
 - Desarrollar buenos hábitos de programación.
 - Aprende a implementar programación básica y avanzada.
 - Aprenda varias características y detalles específicos del lenguaje de programación popular en econometría, Stata.

Sesiones

Este curso estará dividido en 5 tópicos fundamentales:

1. Programación 101 (2 horas)
2. Manejo y Limpieza de Datos (2 horas)
3. Construcción de Datos (2 horas)
4. Análisis de Datos (4 horas)
5. Programando modelos de evaluación de Impacto
 - 5.1 RCTs y Datos Panel (2 horas)
 - 5.2 Diferencias en Diferencias (2 horas)
 - 5.3 Regresión Discontinua (2 horas)

Materiales del Curso

Repositorio en Github

Los materiales del curso se pueden encontrar en el siguiente [enlace](#). Este repositorio será actualizado semanalmente con las siguientes carpetas. A su vez, subiré la misma información al [Aula Virtual](#).

1. lectures (Slides)
 - 01-programming-intro
 - 02-manejo-limpieza-datos
2. data
3. codes (Stata)
 - 01-programming-intro
 - 02-manejo-limpieza-datos
4. syllabus

Encuestas

- Uso de Stata
- Uso de Git
- Uso de \LaTeX

Programación 101

¿Puedo ocupar Excel para análisis de regresiones?

La principal razón por la cual escribimos códigos

- En Excel, se realiza cambios directamente en los datos y guarda nuevas versiones del conjunto de datos.
- En Stata (o R), realiza cambios en las instrucciones sobre cómo pasar de los datos sin procesar al análisis final y guarda nuevas versiones de las instrucciones.
 - En Stata utilizamos do files (or ado files).
 - En R utilizamos RScripts.

Tu código es un resultado

¿Cómo deberíamos de tratar nuestro código?

Los investigadores a menudo tratan el código como un medio para un fin, pero el punto principal de esta presentación es que su código es tanto un fin en sí mismo como el documento o el informe que está escribiendo.

Objetivo de esta primera sesión

- Para convertirse en un gran coder, ambos deben codificar como un coder y pensar como un coder.
- Ambos requieren mucha práctica para dominar, pero el objetivo de esta sesión es brindarle un marco para pensar como un codificador respondiendo preguntas como:
 - ¿Por qué programar?
 - ¿Cómo programo para que sea lo más útil para otras personas de mi equipo?

Programar en academia vs workplace

- En academia:
 - Estar en el correcto importa.
- Workplace:
 - Estar en lo correcto es igual de importante que en la academia,
 - Los miembros del equipo pasados, actuales y futuros contribuirán al mismo código y, por lo tanto, debemos estandarizar cómo programamos y centrarnos en las habilidades para programar como equipo.

Pensamiento crítico sobre los datos.

- ¿Creo en este número?
- ¿Qué puede salir mal en mi código?
- ¿Cómo se tratarán los valores perdidos en este comando?
- ¿Qué pasaría si se agregaran más observaciones al conjunto de datos?
- ¿Qué sucedería si algunas observaciones se eliminaran del conjunto de datos?

Explore un conjunto de datos sin procesar

¿Qué es lo primero que buscamos cada vez que abre un nuevo conjunto de datos por primera vez?

1. Unidad de observación.
2. Identificación completa y única de una variable ID.

Explore a raw data set

- Household_data.csv

HHID	Village	District	HH number	HH head	HHH Age
022501	25	2	1	Andrew	52
022502	25	2	2	Patrick	48
023207	32	2	7	Charles	29
023205	32	2	5	Jeffrey	37
012501	25	1	1	Walter	48
011103	11	1	3	Anne	26
011205	12	1	5	Lawrence	61
024502	45	2	2	Dennis	45
024501	45	2	1	Nancy	41

Explore a raw data set

- Clinic_data.csv

Clinic ID	Clinic Number	District	Patient	Age
02452	542	2	Andrew	52
02543	543	2	Patrick	48
02156	156	2	Charles	29
01152	152	1	Jeffrey	37
01152	152	1	Walter	49
01238	238	1	Anne	26
01122	122	1	Lawrence	61
02122	122	2	Dennis	45
02122	122	2	Nancy	41

Variable ID

- Solo trabaje con un conjunto de datos que tenga una variable de ID. Si el conjunto de datos que ha recibido no tiene uno, entonces crearlo es su primera tarea.
- Probar que la variable ID es única y completamente identificable.
- Use solo una variable como variable de ID.

División de roles en el trabajo de datos

Asistentes de investigación

- Nadie mirará los datos tanto como el RA.
- Las irregularidades en los datos que la RA no identifica a menudo nunca se descubrirán

Economistas

- A cargo de decidir qué irregularidades se corregirán y cómo.
- Los economistas dependen completamente de los RA para identificar irregularidades y obtener la información para tomar la mejor decisión.

Estilo

Stata no distingue entre un espacio vacío y muchos espacios vacíos, o un salto de línea o muchos saltos de línea. Es una gran diferencia para el ojo humano y nunca compartiríamos un documento de Word, una hoja de Excel o una presentación de PowerPoint sin pensar en espacios en blanco, aunque en este caso lo llamamos formato.

¿Es esta diapositiva fácil de leer?

White space - Espacio blanco

- Stata no distingue entre un espacio vacío y muchos espacios vacíos, o un salto de línea o muchos saltos de línea.
- Es una gran diferencia para el ojo humano y nunca compartiríamos un documento de Word, una hoja de Excel o una presentación de PowerPoint sin pensar en espacios en blanco, aunque en este caso lo llamamos formato.

Líneas verticales

```
gen NoPlotDataB1 = 0
replace NoPlotDataB1 = 1 if c_plots_total_area >= .

gen NoHarvValueDataB1 = 0
replace NoHarvValueDataB1 = 1 if c_harv_value >= .

rename c_gross_yield c1_gross_yield
rename c_net_yield c1_net_yield
rename c_harv_value c1_harv_value
rename c_total_earnings c1_total_earnings
rename c_input_spend c2_inp_total_spending
rename c_plots_total_area c1_total_plotsize

tempfile BL_append
save `BL_append'
```

```
gen      NoPlotDataB1 = 0
replace  NoPlotDataB1 = 1      if c_plots_total_area >= .

gen      NoHarvValueDataB1 = 0
replace  NoHarvValueDataB1 = 1    if c_harv_value >= .

rename   c_gross_yield      c1_gross_yield
rename   c_net_yield        c1_net_yield
rename   c_harv_value       c1_harv_value
rename   c_total_earnings   c1_total_earnings
rename   c_input_spend      c2_inp_total_spending
rename   c_plots_total_area c1_total_plotsize

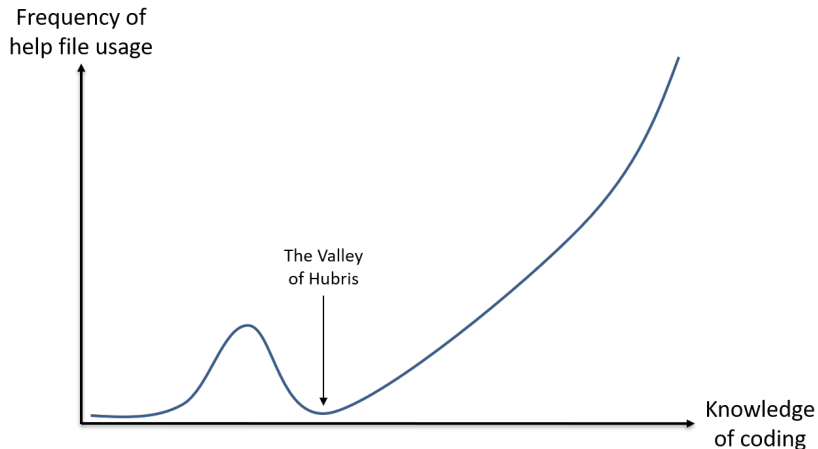
tempfile BL_append
save     `BL_append'
```

Líneas verticales

```
*-create dummy for employed
gen employed = 1
replace employed = 0 if _merge == 2
label var employed "Person exists in employment data"
label define yesno 1"yes" 0"no"
label values employed yesno

*-create dummy for employed
gen      employed = 1
replace  employed = 0 if _merge == 2
label var employed "Person exists in employment data"
label def yesno 1"yes" 0"no"
label val employed yesno
```

Uso del archivos de ayuda y el conocimiento de programación



Archivos de ayuda (help files)

- Escriba Stata: `help nombre_comando`
- ¡Acostúmbrate a usar el archivo de ayuda con la mayor frecuencia posible!
 - Incluso con comandos familiares, siempre más para aprender.
- Los archivos de ayuda son solo resúmenes del manual de referencia prácticas de codificación, errores comunes, enfoques alternativos.

¿Dónde están las regresiones

- Todavía no hemos realizado nada sobre regresiones.
- En programación, el análisis es la parte fácil siempre que el conjunto de datos esté configurado correctamente para el análisis.
- Cómo usar los comandos para el análisis es mucho más fácil buscarlo en Google o preguntarle a alguien cómo hacerlo comparado con la limpieza de datos, la gestión de datos y el aseguramiento de la calidad de los datos.

En palabras sencillas

“Cuando tu código funciona, solo estás a medias”.

Revisar el código de las demás

- Compare su código y discuta las diferencias con otras personas.
- Pregunte qué es más fácil de entender si piensa en su **do file** como una instrucción. ¿Que es difícil?
- Aplique un pensamiento crítico al trabajo de datos al código de cada uno, i.e., pruebe si habrá errores en su código, o si faltan datos.
- Si nadie le permite ver su código, pídale a las personas que lo vean.
 - ¿Alguna vez le has pedido a alguien que te ayude a corregir tu documento de Word?

Leer el código de otras personas

- Busque el código en GitHub
- <https://github.com/trending/stata> (Stata)
- <https://github.com/vikjam/mostly-harmless-replication> (Stata y otros idiomas)
- Googlear, pero antes, hágase preguntas críticas sobre el código que encontró.
- ¿Por qué esta persona programó de esta manera?
- ¿Esto se aplica a mi contexto?

Literatura de gestión de bases de datos

- La carpeta de su proyecto es una base de datos informal, y personas muy inteligentes que trabajan con bases de datos han estado pensando mucho en esto.
- No tengo un libro específico para recomendar, ya que no conozco un libro escrito para nuestro contexto, por lo que este método no es para los débiles

Conclusión

- Su código es una herramienta que debe desarrollar como si otras personas lo estuvieran usando.
- Solicite ayuda de sus pares para revisar su código.
- Al enviar el código, formatee con el mismo cuidado que formatearía su currículum o su carta de presentación.

Nos vemos mañana.