Introducción a la programación y al análisis de datos

Programación para el análisis de datos

Departamento de Ciencias Sociales, UCU - Martín Opertti

Presentaciones

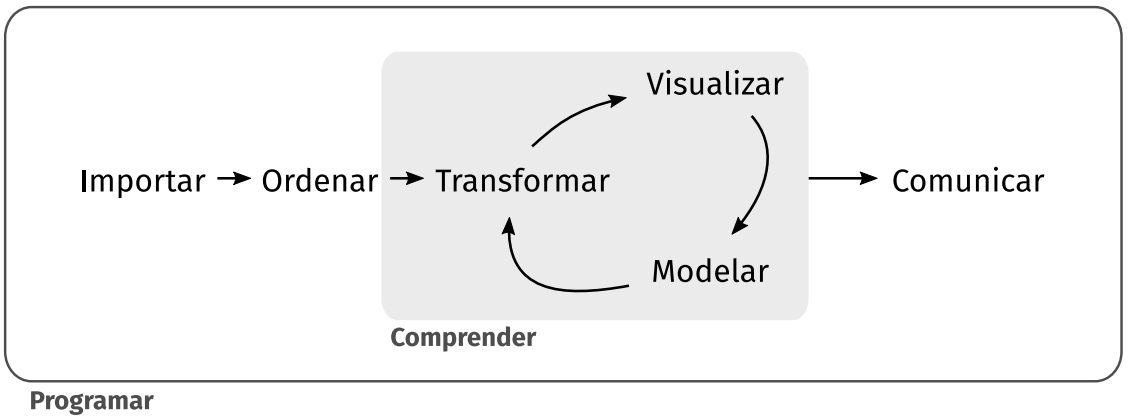
Presentaciones

- ¿Cómo se llaman?
- ¿Qué carrera están cursando?
- Sensaciones y experiencia con programación, análisis de datos y ciencia de datos o "data science".
- ¿En qué les gustaría trabajar?

Análisis de datos

Análisis de datos

"Data analysis is the process by which data becomes understanding, knowledge and insight". Hadley Wickham



Fuente: R4DS

¿Por qué analizar datos?

"In God we trust, others must bring data". W. E. Deming

- **Investigación**
- **Mercado laboral**
- **Revolución de los datos y las computadoras**

Ciencia de datos

drewconway.com

Softwares estadísticos y programación

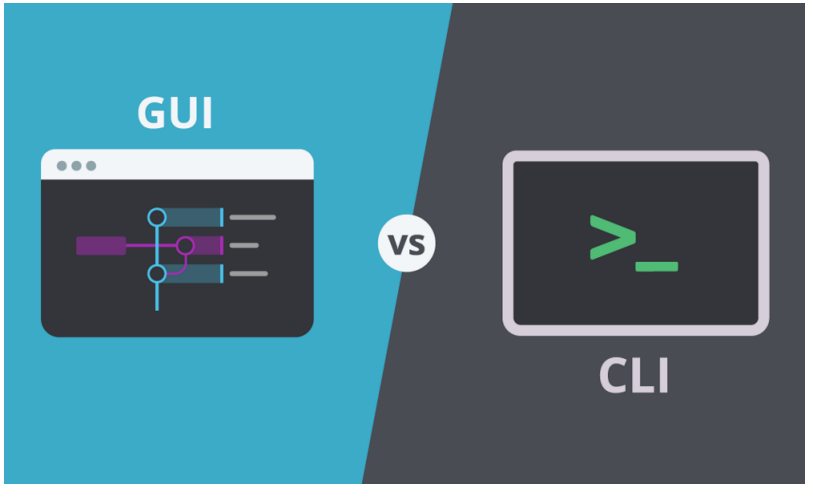
Softwares estadísticos

Los softwares o paquetes estadísticos son programas informáticos diseñado para llevar a cabo análisis estadísticos. Algunos de los más utilizados son SPSS, Stata, SAS o R.

Fuente: R4DS

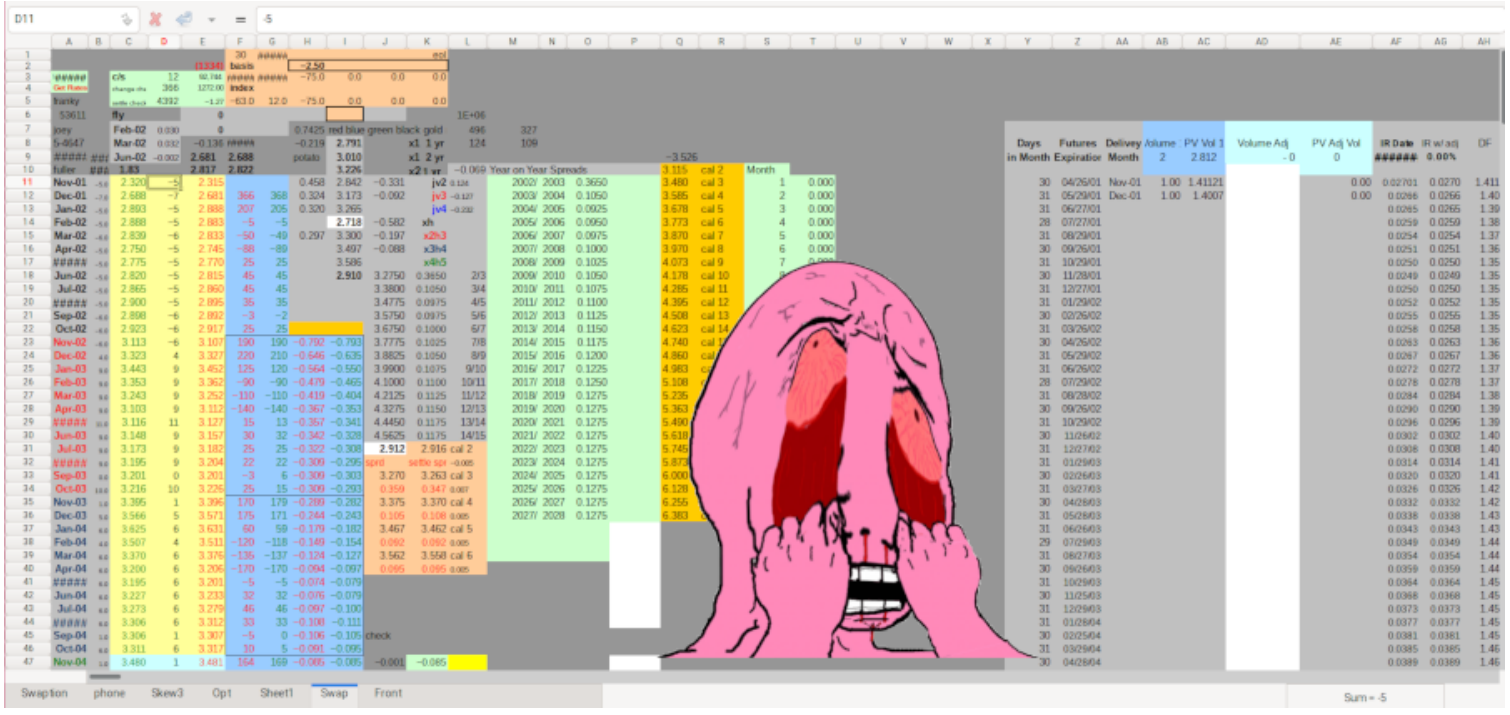
Modalidades de uso de softwares estadísticos

- Point and click o GUI (Interfaz gráfica del usuario)
- Programación



¿Por qué programar?

¿Por qué programar?



Fuente: https://www.brodrigues.co/

¿Por qué programar?

- El error de Reinhart y Rogoff
- El trabajo de Reinhart y Rogoff mostró que el crecimiento económico real promedio se desacelera (una disminución del 0,1 %) cuando la deuda de un país aumenta a más del 90 % del producto interno bruto (PIB), y esta cifra del 90 % se empleó repetidamente en argumentos políticos sobre austeridad
- Olvidaron seleccionar una fila que contenía 5 de los 20 países analizados, y el declive de 0.1% se transformaba en un crecimiento de 2.2%
- ¿Qué podría haber sido distinto si Reinhart y Rogoff hubieran utilizado un lenguaje de programación?

¿Por qué programar?

- **Eficiencia:** Si bien programar llevará más tiempo al principio, rápidamente permite ahorrar mucho tiempo en comparación a tareas realizadas a través de una interfaz gráfica.
- **Más posibilidades:** En los softwares estadísticos tradicionales muchas operaciones (las más complejas o especificas) suelen no estar disponibles en la interfaz gráfica por lo que solo se pueden realizar mediante el uso de código.
- **Recolección de datos:** Saber programar abre la posibilidad para la recolección de datos que no es posible o es muy costosa manualmente.
- **Reproducibildiad:** El análisis de datos mediante programación mejora la transparencia y reproducibilidad en el proceso de generación de conocimiento
- **Colaboración:** La programación abre la puerta para una colaboración mucho más sencilla y eficiente.

Sobre del curso

Objetivos generales

- Aprender a estructurar, manipular, analizar y visualizar datos
- Familiarizarse con técnicas de programación en general
- Conocer los distintos usos y posibles aplicaciones de Stata, R y Python
- **Aprender a aprender: este curso no es solo aprender las operaciones que vamos a ver en clase, es para aprender a resolver problemas nuevos de forma autónoma.**

Objetivos específicos

- Adquirir nociones básicas de Stata para el análisis de datos
- Adquirir herramientas para comprender y escribir código en R (reutilizable y interpretable) para el análisis de datos
- Adquirir conocimientos para aprender de forma independiente funciones en R
- Breve introducción a Python (usos y posibilidades)

Estructura

Semana	Tema
Semana 1	Introducción al curso y al análisis de datos
Semana 2	Introducción a Stata
Semana 3	Introducción a Stata II
Semana 4	Primera evaluación parcial
Semana 5	Introducción a R
Semana 6	Introducción a R II
Semana 7	Importar y limpiar datos en R
Semana 8	Estadística descriptiva en R
Semana 9	Manipulación de datos en R
Semana 10	Manipulación de datos en R II
Semana 11	Visualización de datos en R
Semana 12	Estadística inferencial en R
Semana 13	Reportes con R Markdown y técnicas de programación avanzadas
Semana 14	Segunda evaluación parcial
Semana 15	Introducción a Python
Semana 16	Presentaciones y cierre del curso

Materiales

Cada semana se colgarán en la webasignatura:

- Una presentación
- El código de la clase
- Ejercicios
- Solución de los ejercicios

Importante:

- **Calificación:** Una calificación mínima de B es requerida para exonerar el curso.
- **Asistencia:** Los alumnos deberían asistir al 75% de las clases
- **Horario:** El curso se dictar a los´días martes y jueves de 8:00 a 9:20 en el laboratorio Auriga
- **Evaluación:**
  - Primera evaluación parcial 20 %
  - Segunda evaluación parcial 40 %
  - Trabajo final 40 %

Regla de los 5 minutos

- Cuando estamos trabajando en ejercicios prácticos y nos enfrentamos a una situación que no podemos resolver en nuestro primer intento, debemos probar durante al menos 5 minutos resolverlo sin preguntar, aplicando los siguientes pasos.

Pasos a seguir si no funciona una línea de código: Paso 1

- En caso de haber un mensaje de error, leerlo. Traducirlo si es necesario. La mayoría de las veces la solución se puede deducir del mensaje de error.

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> my_data <- read.csv(mydata.csv)
Error in read.table(file = file, header = header, sep = sep, quote = quote,  :
  object 'mydata.csv' not found
> Can I have that in plain English, please?
Error: unexpected symbol in "Can I"
> :(
```

Pasos a seguir si no funciona una línea de código: Paso 2

- Chequear que no haya errores de tipeo. Si hay una letra fuera de lugar, los programas no podrán correr el código. Tener especial cuidados a mayusculas/minúsculas, acentos y puntuación.

Pasos a seguir si no funciona una línea de código: Paso 3

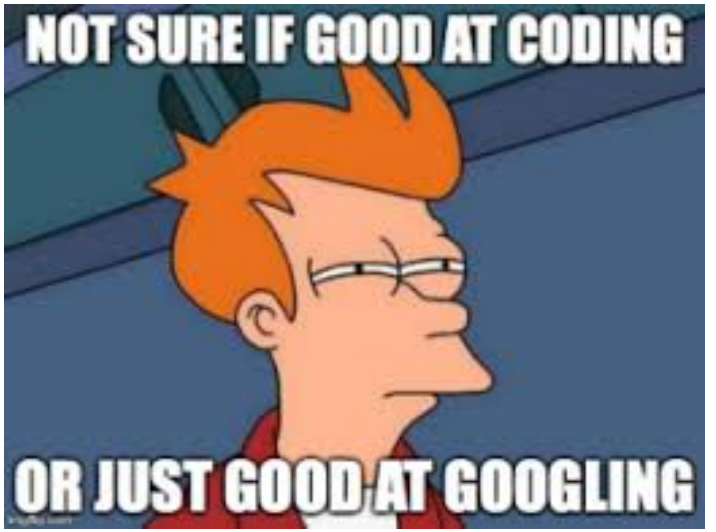
- Si los primeros 2 puntos no funcionan, no entrar en pánico. Probablemente sea tu culpa, la computadora no hace otra cosa que llevar a cabo las instrucciones que vos le das. No escudarse en "esto no anda". Es muy difícil y poco probable toparse con un error que no sea nuestro en las funciones que vamos a utilizar en este curso.

Pasos a seguir si no funciona una línea de código: Paso 4

- Redefinir el problema. Definir concretamente qué esperamos que ese código haga, a qué objetos y cuál es el resultado esperado. Chequear que le estemos dando las instrucciones correctas a la computadora.

Pasos a seguir si no funciona una línea de código: Paso 5

- Googlear. En programación, una enorme cantidad del tiempo es destinada a entender por qué las cosas no funcionan de la forma en que esperabamos. Abrir el navegador, escribir R al comienzo (o el lenguaje/software que estemos utilizando) y luego describir el problema. En lo posible, recomiendo hacerlo en inglés, incluso si es nescesario utilizar el traductor. Ser claro y especifico en la búsqueda, y leer con atención las publicaciones que aparecen (tanto la pregunta como la respuesta).



Pasos a seguir si no funciona una línea de código: Paso 6

- Pedir ayuda. En caso de que los anteriores pasos no funcionen, pidan ayuda. A compañeros o al profesor. Pero es importante agotar los recursos antes, porque los códigos que no funcionan no desaparecen.

Softwares estadísticos

Softwares estadísticos y lenguajes de programación

Software	Acceso	Uso
SPSS	Pago	GUI y código
Stata	Pago	GUI y código
R	Libre	Código
Python	Libre	Código

SPSS

- **Ventajas:**
  - Buen GUI
  - Funciona muy bien para datos de encuestas
- **Desventajas:**
  - Pago
  - Pocas funcionalidades

Stata

- **Ventajas:**
  - Relativamente sencillo de usar
  - Bastantes funcionalidades
- **Desventajas:**
  - Pago
  - Incompleto (machine learning, geolocalización, etc.)

R

- **Ventajas:**
  - Gratuito y libre
  - Muy completo y flexible
  - Comunidad de usuarios
  - Foco en estadítica
- **Desventajas:**
  - Puede ser lento
  - Curva de aprendizaje

Python

- **Ventajas:**
  - Muy potente
  - Muchas funcionalidades
  - Sintáxis clara
  - Buena comunidad
- **Desventajas:**
  - No está particularmente pensado para estadística
  - Curva de aprendizaje

¿Cómo pensar los datos?

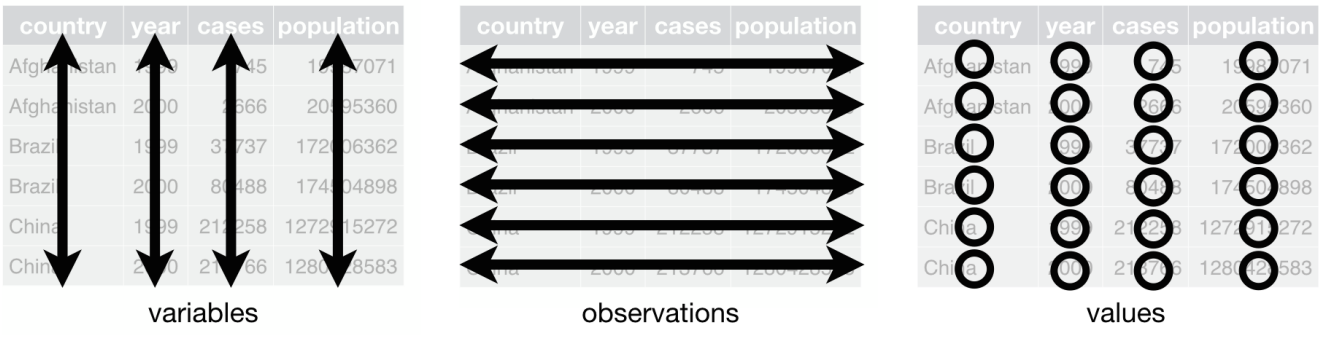
Ejercicio

En una planilla de excel creen una tabla con 5 países y datos sobre ellos. Debe tener al menos dos datos por país.

Ahora, en esa misma planilla transformen esa tabla en una base de datos.

Estructura de datos

- Un dataframe o marco de datos (es lo que nos solemos referir como "base de datos")
- Es una forma de estructurar datos con filas y columnas. Las filas suelen ser las observaciones y las columnas las variables



Wichkham & Grolemund (2018)

Base de datos

pais	capital	continente
Uruguay	Montevideo	América
Suiza	Berna	Europa
Egipto	El Cairo	África
Irán	Teherán	Asia
Austrlia	Canberra	Oceanía

Ejercicio

Agregar información sobre el lenguaje de estos países

Base de datos

pais	capital	continente	idiomas
Uruguay	Montevideo	América	Español
Suiza	Berna	Europa	Alemán, Italiano, Francés y Romance
Egipto	El Cairo	África	Árabe
Irán	Teherán	Asia	Persa
Austrlia	Canberra	Oceanía	Inglés

Base de datos

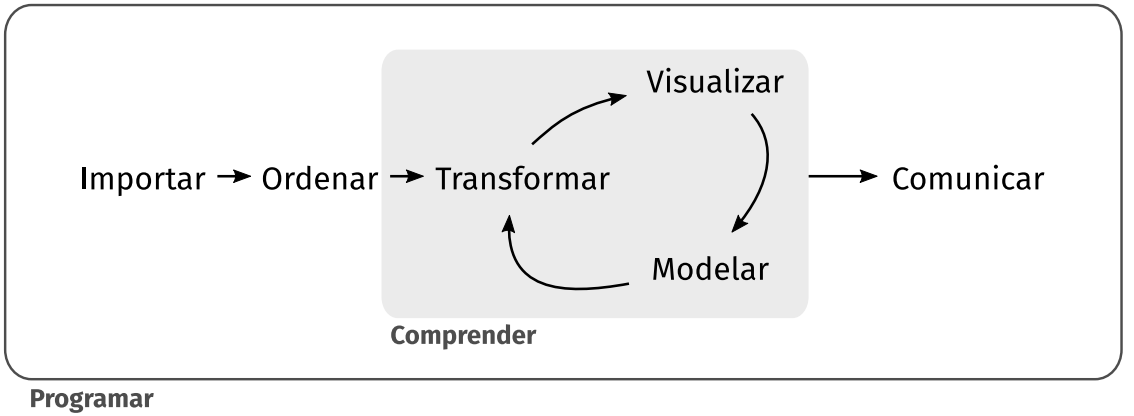
pais	capital	continente	idiomas
Uruguay	Montevideo	América	Español
Suiza	Berna	Europa	Alemán
Egipto	El Cairo	África	Árabe
Irán	Teherán	Asia	Persa
Austrlia	Canberra	Oceanía	Inglés

Base de datos

pais	capital	continente	idioma_1	idioma_2	idioma_3	idioma_4
Uruguay	Montevideo	América	Español			
Suiza	Berna	Europa	Alemán	Francés	Italiano	Romance
Egipto	El Cairo	África	Árabe			
Irán	Teherán	Asia	Persa			
Austrlia	Canberra	Oceanía	Inglés			

Flujo de trabajo

Flujo de trabajo



Fuente: R4DS

Ejercicio

¿Cuáles son los 5 países de América del Sur con mayor densidad de población?

Carpeta del curso

Crear una carpeta

- Crear una carpeta para guardar los archivos del curso, con el nombre que prefieran
- Descargar de la webasignatura la carpeta "data" y pegarla (con ese nombre), dentro de la carpeta creada en el punto anterior
- Crea una carpeta para guardar los do files y scripts (archivos de código) del curso.
- Crea además dos carpetas más dentro de tu carpeta: "graficos" y "resultados"