Akádemo
¡Estudia para crecer!

Bootcamp de Python

Septiembre, 2023

Antes de comenzar, es importante destacar que estas presentaciones tienen como objetivo abordar aspectos que el código por sí solo no puede mostrar por completo. Este curso no se trata de ejecutar grandes cantidades de código, sino de comprender los fundamentos esenciales. El objetivo es que los estudiantes puedan dar sus primeros pasos en el desarrollo de sus propios proyectos o adquirir una visión global de la programación aplicada.

El hecho de dedicar muchas horas a programar no garantiza automáticamente que alguien sea considerado un experto. Por ejemplo, un camionero puede conducir más kilómetros que un corredor de Fórmula 1, pero eso no implica que tenga una técnica superior. En programación, además de la cantidad de código escrito, es fundamental comprender las piezas clave de Python para tener una comprensión sólida de cualquier proyecto.

Durante este curso, destacaremos “lo mas usado en Python”. Esto les permitirá tener una comprensión acertada de cualquier código que encuentren.

Sobre el instructor:
Dereck Amesquita
Bachiller en Economía

Analista de proyectos en ESAN.
Curso de Finanzas Avanzadas del BCRP

Creador de una quasilibreria, BCRP-Scrapper.

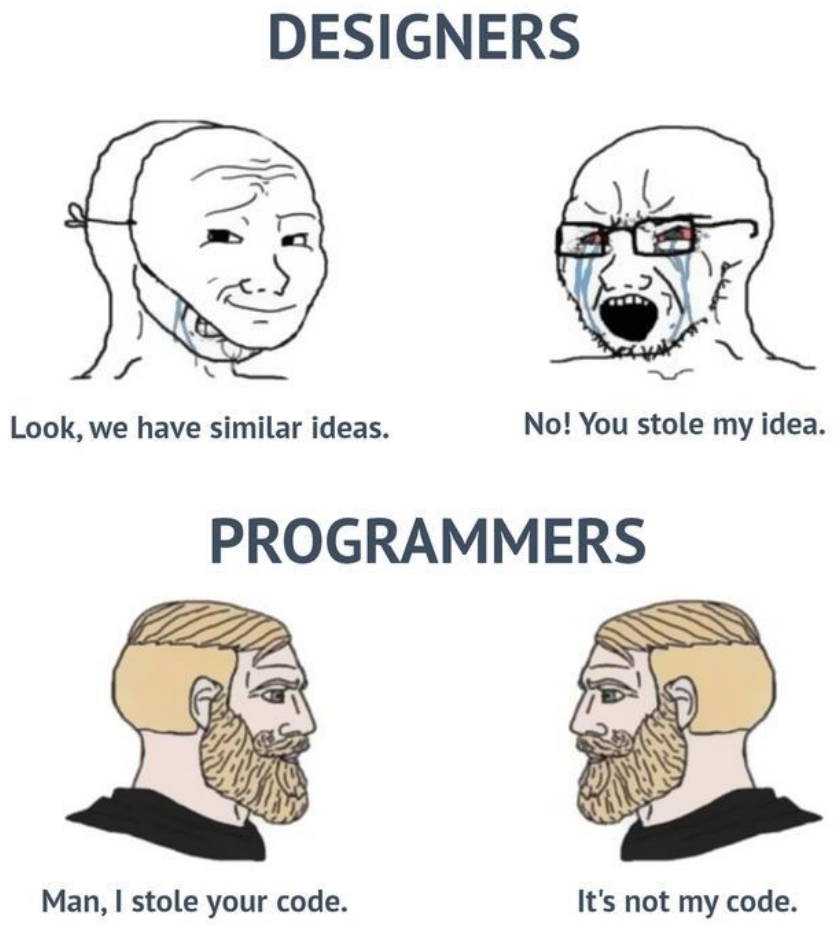
Orientado a data, comencé usando R, posteriormente migre a Python por su capacidad de conectarse en múltiples entornos y la asimilación de distintos lenguajes de programación y tecnologías.

Soy un fiel creyente que la ciencia de datos si busca ser rentable, su fin recae en la producción y no el "fit" de modelos.

Introducción

Historia y Aplicaciones de Python

- Fundado por Guido van Rossum en los Países Bajos.
- Lanzamiento de Python 1.0 en 1994.
- Nombre inspirado en los Monty Python.
- Ampliamente utilizado en empresas como Google, NASA, Dropbox, BCP, AC.
- Su versatilidad lo hace esencial en la programación moderna.



Introducción

¿Qué es Python?

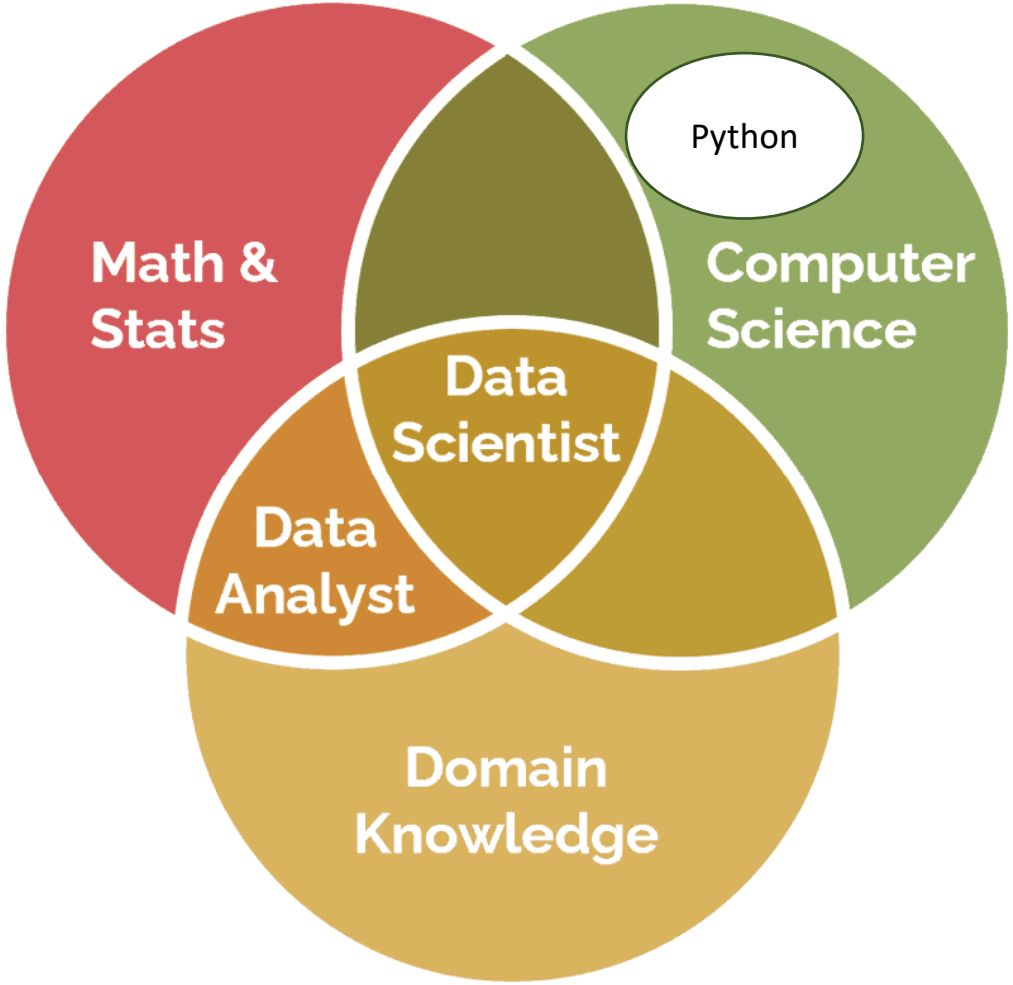
- Python es un lenguaje de programación de alto nivel.
- Es conocido por su sintaxis sencilla y legible. Imprimir un texto, se logra en una linea, a diferencia de cualquier otro lenguaje
- Tiene una de las comunidades mas grandes.
- Ha desarrollado aplicaciones para absolutamente todo.
- Curva de aprendizaje rentable.

¿Por qué es importante?

- Ampliamente utilizado en diversas industrias:
  - Ciencia de datos (pandas, NumPy).
  - Inteligencia artificial y aprendizaje automático (TensorFlow, PyTorch).
  - Automatización, scripting y desarrollo de juegos.

Introducción

¿Qué debo saber antes de aprender Python?



Introducción

¿Qué debo saber antes de aprender Python?

- *Python no es más que una herramienta para construir algo.*
- Es decir, por aprender a programar en python no necesariamente me volveré un Data Science, un Portfolio Manager, etc.
- El errór mas comun, es creer que si aprendemos Python, podremos obviar tópicos de estadística o de matemática.

Introducción

¿Cómo puedo usar Python?

- *Python en un lenguaje,* por lo cual, para poder usarlo, necesitaremos instalar el lenguaje como tal, y además un interprete que nos permitirá escribir y ejecutar nuestro código.
- El curso no pretende cubrir estos aspectos. Por lo cuál, usaremos Google Colab, un entorno que contiene el lenguaje y el interprete de Python.

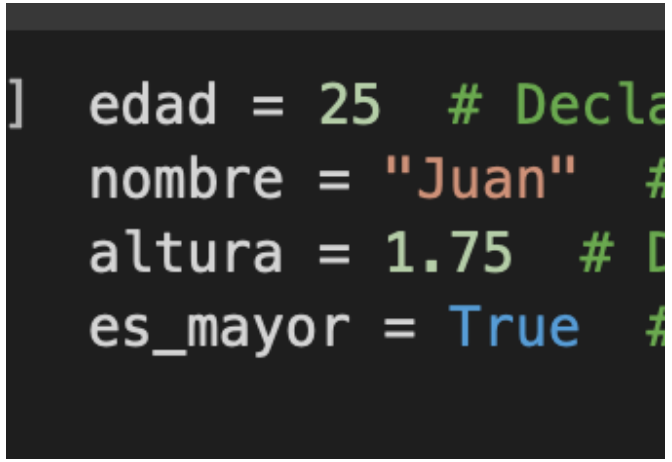
Tipos de Datos en Python

¿Qué son las variables?

- En Python, las variables son contenedores que almacenan información o datos. Piensa en ellas como etiquetas o nombres que se utilizan para representar valores en un programa.

Uso de Variables:

- Las variables son esenciales para la programación porque nos permiten almacenar y manipular datos. Son como cajas de almacenamiento que pueden contener números, texto, valores verdaderos o falsos, entre otros.
- Puedes imaginar una variable como un espacio de memoria donde guardamos información para usarla más tarde en nuestro programa.



Tipos de Datos en Python

En Python, existen diferentes tipos de datos que podemos almacenar en variables:

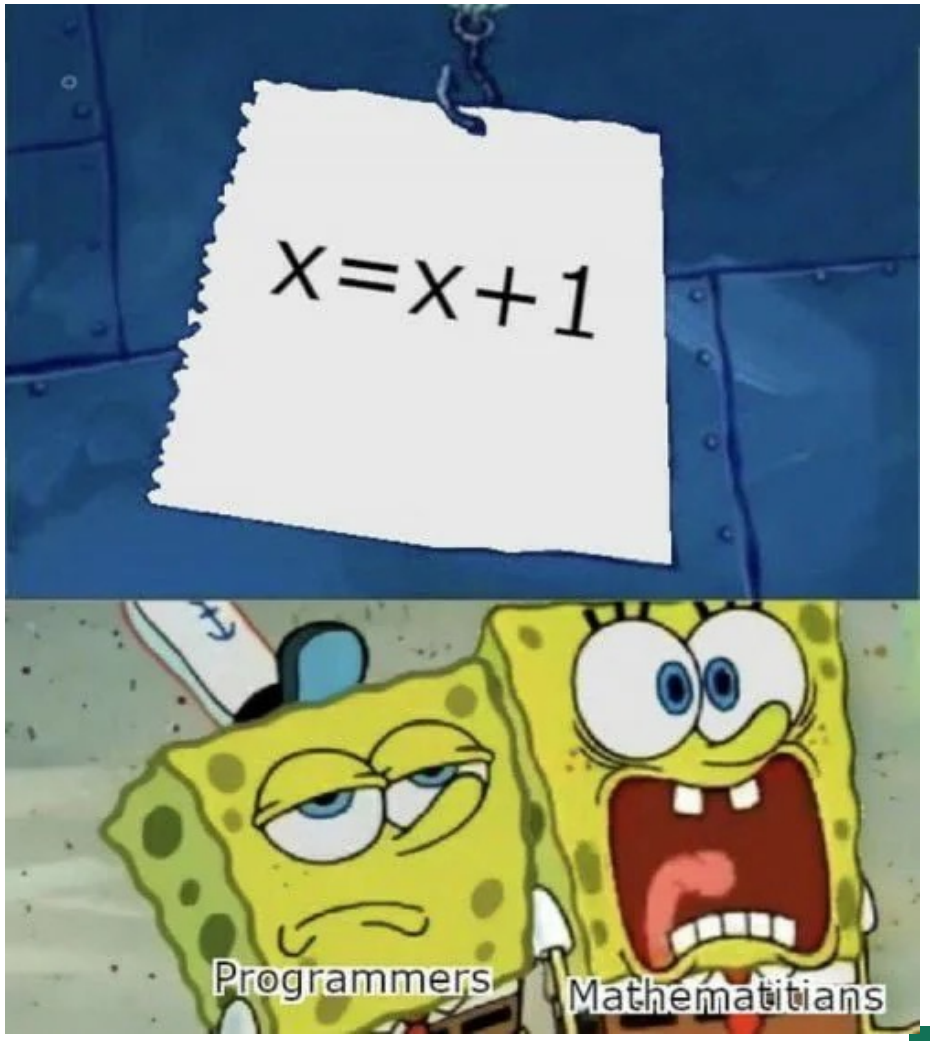
- **Números Enteros (int):** Utilizados para representar números enteros, como la edad de una persona o la cantidad de productos en stock.
- **Números Flotantes (float):** Sirven para números con decimales, como la altura de una persona o el precio de un producto.
- **Cadenas (strings - str):** Se utilizan para representar texto, como nombres, mensajes o descripciones.
- **Booleanos (bool):** Representan valores de verdad, es decir, verdadero (True) o falso (False). Se usan para tomar decisiones basadas en condiciones.

Tipos de Datos en Python

Un beneficio de renombrar variables es la reutilización de la variable. En este caso si inicialmente asignamos a X el valor de cero y posteriormente le añadimos una cantidad, el valor de X se actualizara sin problemas.

$$x = x + 1$$

$$x_t = x_{t-1} + 1$$



Sentencias Condicionales en Python: Ejemplos

Existen principalmente dos tipos de sentencias condicionales:

- **if (Si):** La sentencia if se utiliza para ejecutar un bloque de código si una condición es verdadera. Es el camino principal que se sigue si la condición se cumple.
- **else (Si no):** La sentencia else se utiliza junto con if y se ejecuta cuando la condición en el if es falsa. Proporciona un camino alternativo a seguir cuando la condición no se cumple.

Sentencias Condicionales en Python: Ejemplo

Además de if y else, también tenemos:

- elif (Si en otro caso): La sentencia elif se utiliza para evaluar múltiples condiciones en secuencia. Si la condición en un if es falsa, Python verifica la siguiente condición en un elif y así sucesivamente. Esto nos permite manejar varias condiciones de manera ordenada.
- Puedes utilizar los operadores de comparación >, >=, <, <=, así como los operadores lógicos and (y) y or (o) para evaluar condiciones complejas.

```
preciocompra = 50
precioactual = 60

if preciocompra < precioactual:
  print('Vende')
```

Bucles: For

- Imagina el bucle for *como un vehículo que te lleva a través del Nasdaq*, explorando las cotizaciones de empresas tecnológicas como Apple, Tesla, Microsoft y NVIDIA una por una. Cada vuelta del bucle te permite descubrir información valiosa sobre estas empresas y tomar decisiones emocionantes en el mundo financiero.
- En muy simple, este bucle te permite tomar cada elemento de un conjunto de forma ordenada.

```
tecnologicas = ["Apple", "Tesla", "Microsoft", "NVIDIA"]

for x in tecnologicas:
    print("Nasdaq: ", x)
```

Bucles: While

- Este mes menos usado que for. En este bucle se haran iteraciones hasta cumplir determinado requerimiento.
- En este ejemplo, el bucle while ira realizando iteraciones siempre que tengamos un numero diferente a 5. Se podria leer como: “*Mientras numero sea diferente de cinco, sumale uno*”.

```
numero = 0

while numero != 5:
    numero = numero + 1
    print("Número nuevo:", numero)

print("¡Encontraste el número 5!")
```

Funciones: Def

- Las funciones en Python son como herramientas personalizadas que puedes crear para realizar tareas específicas en tu código. Imagina que tienes una caja de herramientas con herramientas para diferentes trabajos.
- Las funciones, muchas veces ahorran grandes cantidades de código que no deben repetirse. Junto a For son la clave de la automatización.
- Una posible relación podria ser que funcionan como una tuberia, input y output.

```
def suma(a, b):
    resultado = a + b
    return resultado
```

Recapitulación

Hasta este momento hemos aprendido los siguientes temas puntuales:

- Variables
- Condicionales
- Bucle For
- Bucle While
- Funciones

Con estos insumos ya estariamos listos para aprender a usar librerias que acomodan nuestro trabajo. Hay 2 librerias clave: pandas y numpy.

Diccionarios

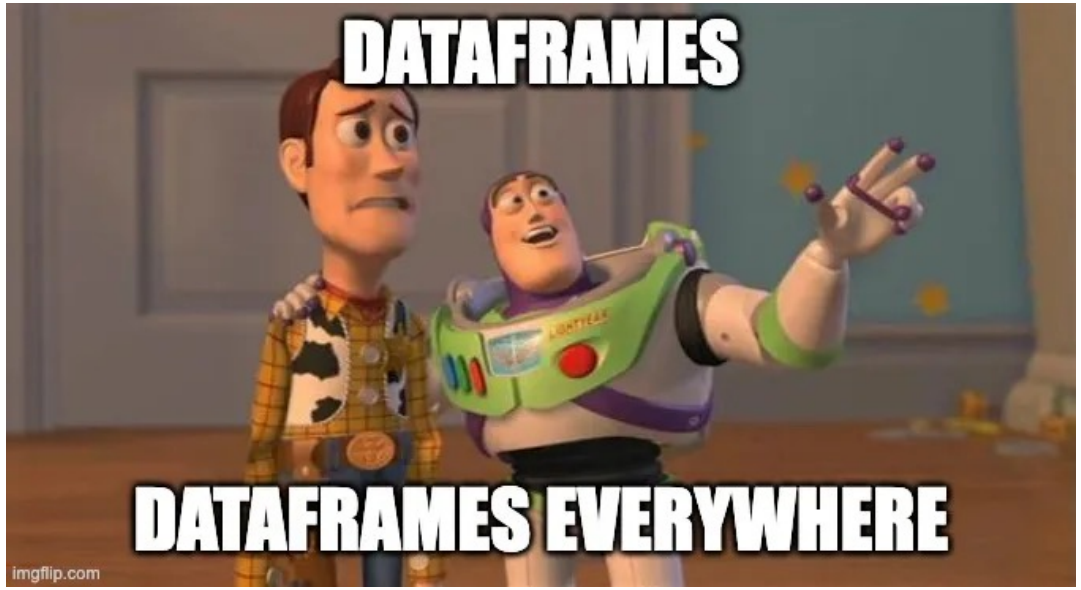
- Los diccionarios en Python son estructuras de datos que te permiten almacenar y organizar información de manera flexible. A diferencia de las listas que utilizan índices numéricos para acceder a sus elementos, los diccionarios utilizan claves (keys) para acceder a sus valores (values).
- Cada elemento en un diccionario consiste en una pareja de clave-valor.

```python
persona = {
    "nombre": "Juan",
    "edad": 30,
    "ciudad": "Ejemploville"
}

# Acceder a valores utilizando las claves
print("Nombre:", persona["nombre"])
print("Edad:", persona["edad"])
print("Ciudad:", persona["ciudad"])
```

Pandas Dataframe

- En un modo muy simple. Pandas es el excel de python. Si bien es cierto no es tan sencillo de usarlo como si fuera una hoja de calculo. Pandas nos permite hacer todo tipo imaginable de operaciones.
- Ningun analista vive sin excel, del mismo modo. Si algun día deciden dejar de usar excel (imposible), la alternativa mas corta será pandas.



Pandas Dataframe ¿Héroe o villano?

- Consumo de Memoria: Pandas puede ser intensivo en cuanto a memoria, especialmente para conjuntos de datos grandes. Esto puede ser una limitación en sistemas con recursos limitados.
- Rendimiento Limitado: Aunque es adecuado para la mayoría de las tareas de análisis de datos, pandas puede no ser la opción más rápida para ciertas operaciones muy intensivas en cómputo, como bucles sobre grandes DataFrames.
- Documentación Compleja: Aunque pandas tiene una documentación completa, algunas partes pueden ser abrumadoras debido a la cantidad de funciones y opciones disponibles.
- No es la Mejor Opción para Datos en Tiempo Real: Para aplicaciones que requieren procesamiento en tiempo real o baja latencia, pandas no es la elección ideal debido a su sobrecarga.

Sesión 2

Econometría:

- ¿Qué sucede con una regresión si el error no es normal?
- Teorema Central

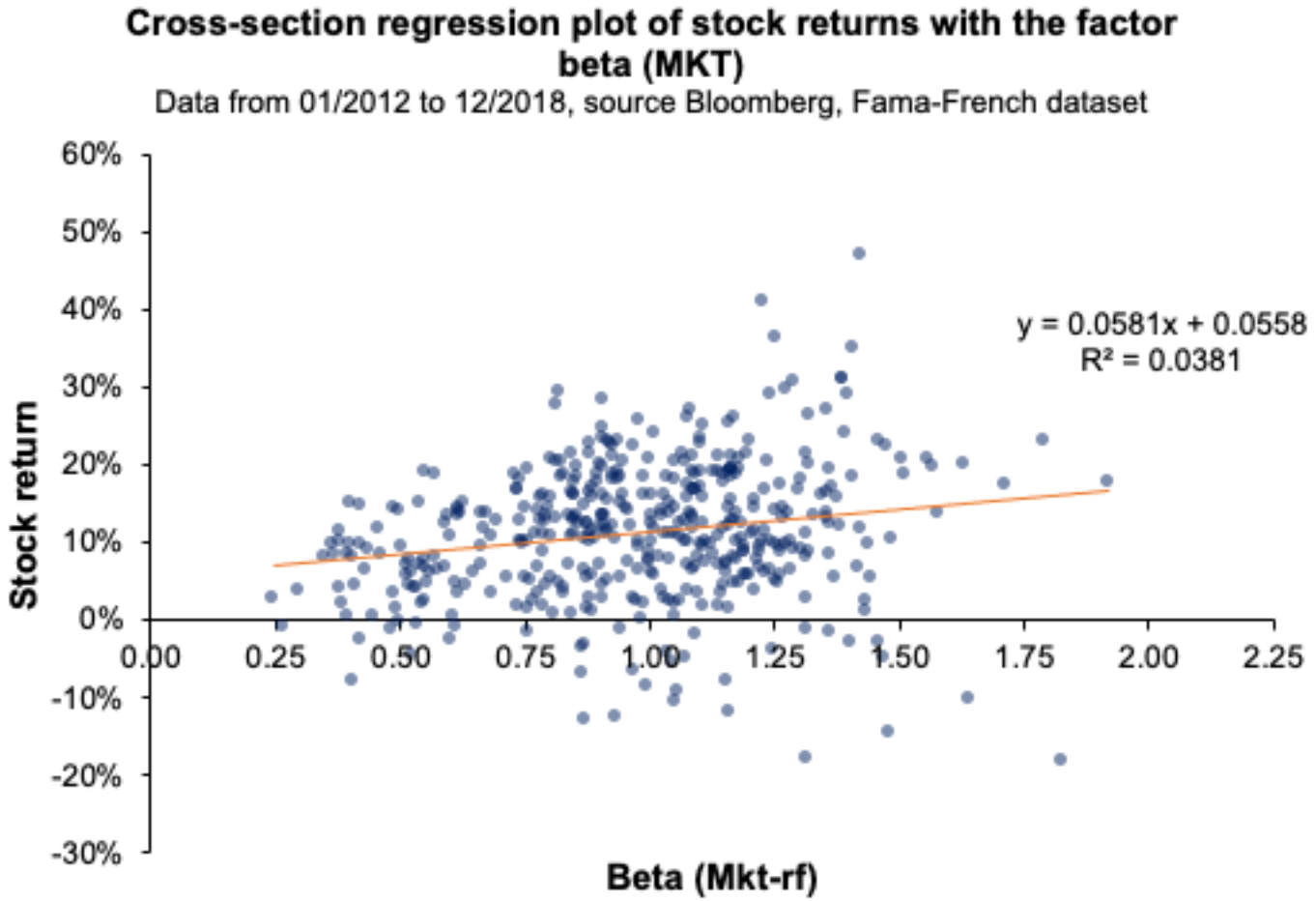
Finanzas:

- ¿Debemos seguir usando el CAPM?

Cálculo:

- ¿Los métodos numéricos deben remplazar las soluciones analíticas?

FINANZAS: ¿Beta explica el retorno?



FINANZAS: ¿Beta explica el retorno?

$$Ri=rf+\beta(Rm-rf)$$

$$(Ri-rf)=\beta(Rm-rf)$$

Exceso de Retorno: $(Ri-rf)$

Exceso de Rendimiento de mercado :$(Rm-rf)$

Modelo econométrico:

$$Y=\beta 0+\beta 1X$$

Econometría: No Normal Distribución

Supuestos de la regresión lineal.

- Linealidad: Se asume que la relación entre las variables independientes (predictoras) y la variable dependiente (respuesta) es lineal
- Independencia de errores: Los errores o residuos, que son las diferencias entre los valores observados y los valores predichos por el modelo, deben ser independientes entre sí. Esto implica que un error no debe influir en el error de otro punto de datos.
- Homocedasticidad: También conocida como homogeneidad de varianza, este supuesto establece que la variabilidad de los errores debe ser constante en todos los niveles de las variables independientes. En otras palabras, la dispersión de los errores debe ser uniforme en toda la gama de valores de las variables predictoras. Los gráficos de residuos suelen utilizarse para evaluar esta suposición.
- Normalidad de errores: Se asume que los errores se distribuyen normalmente con una media de cero. Esto significa que los valores de los residuos deben seguir una distribución de campana simétrica alrededor de cero. La normalidad de errores es importante para realizar pruebas de hipótesis y construir intervalos de confianza adecuados.

Econometría: No Normal Distribución

¿Es tan peligroso que no haya distribución normal ?

- **Inferencias erróneas:** Los errores no normales pueden llevar a conclusiones incorrectas en pruebas de hipótesis y estimaciones de coeficientes.
- **Inestabilidad:** Los coeficientes de regresión pueden volverse inestables con errores no normales.
- **Predicciones sesgadas:** Las predicciones del modelo pueden estar sesgadas si los errores no siguen una distribución normal.
- **Dificultad en la interpretación:** La relación entre variables puede ser difícil de interpretar con errores no normales.
- **Identificación de valores atípicos:** Detectar valores atípicos puede ser problemático si los errores no son normales.
- **Problemas en intervalos de confianza:** Los intervalos de confianza pueden ser inexactos si no se cumple la normalidad de errores.
- **Normalidad de errores:** Se asume que los errores se distribuyen normalmente con una media de cero. Esto significa que los valores de los residuos deben seguir una distribución de campana simétrica alrededor de cero. La normalidad de errores es importante para realizar pruebas de hipótesis y construir intervalos de confianza adecuados.

Econometría: No Normal Distribución

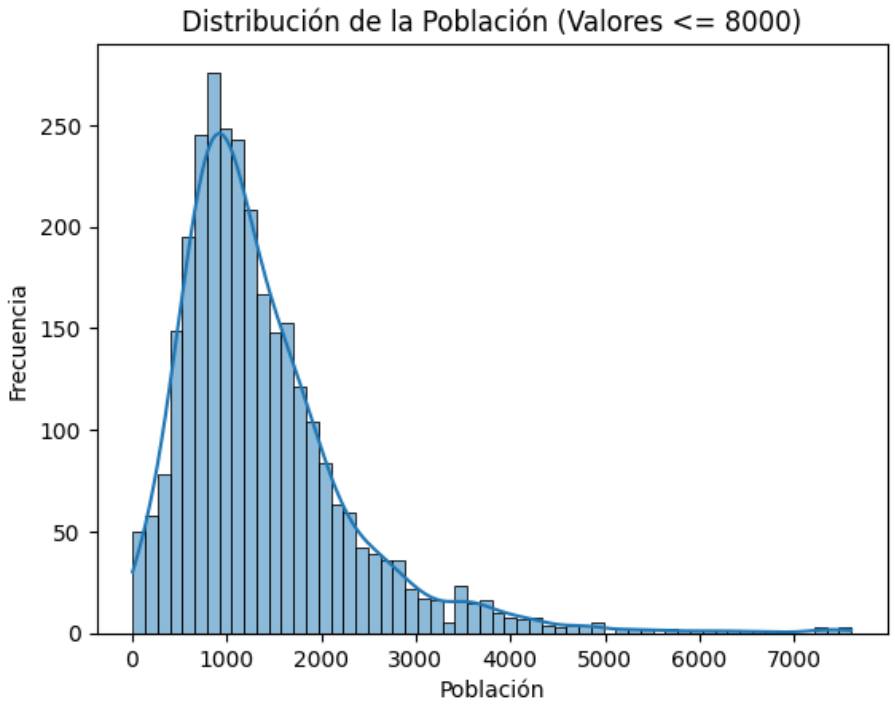
El principal problema sería que el estimador de la regresión no se acerca al “verdadero” parámetro, es decir, aparece el sesgo. Esto se puede demostrar matemáticamente, este no es el fin del curso.

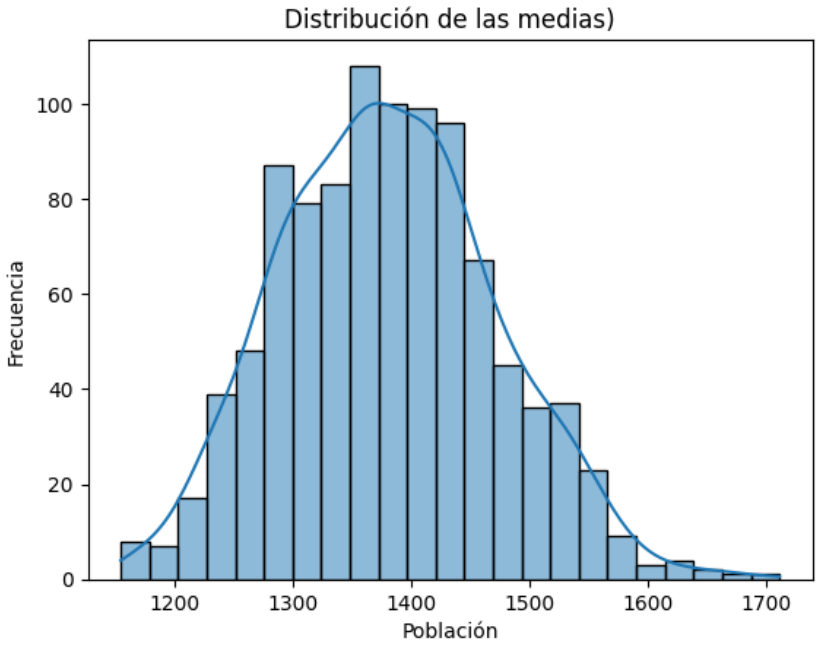
$$E\left[\beta^{estimado}\right] \neq \beta^{poblacional}$$

Si esto no se cumple, podríamos estar sobreestimado o subestimando el efecto que le atribuimos al beta.

Econometría: Teorema Central del Limite

Teorema Central del Límite (TCL) establece que, bajo ciertas condiciones, la distribución de las medias muestrales de una población tiende a seguir una distribución normal, incluso si la población original no es normal. El TCL es uno de los conceptos fundamentales en estadísticas y tiene importantes implicaciones en la inferencia estadística.






Akádemo
¡Estudia para crecer!