{desafío}
latam_

Estadística Univariada y Control de Flujo_

Sesión Presencial 1

Recapitulando

Materia vista esta unidad:

* Introducción a Python y Jupyter.
* Estadística Univariada

{desafío}
latam_

El ambiente de trabajo

{desafío}
latam_

¿Qué es Python?

- Lenguaje con forma sintáctica humana = Menos complejo!
- Amplia comunidad que está constantemente mejorando librerías.
- Multifuncional y utilizado en variados ámbitos.

¿Qué es Anaconda?

- Es una *distribución* de Python que provee de todos los elementos necesarios para un entorno de trabajo en Ciencia de Datos. Dentro de sus instalaciones destacan:
    - `iPython`: el kernel específico que utilizaremos.
    - Una serie de liberías estandares para el trabajo.
    - `conda`: Un administrador de paquetes.
    - Genera ambientes virutalizados
    - Instala Jupyter Notebook.

¿Qué es Jupyter?

- Jupyter (antes llamado `iPython` notebooks) es una aplicación web que permite crear documentos reproducibles que contienen el código, texto, visualizaciones y ecuaciones.

- Jupyter es un acrónimo de:
    - **Ju** lia.
    - **Pyt** hon.
    - **R**.

- Permite exportar el documento a múltiples formatos.

Algunas buenas costumbres

- Siempre generen una carpeta del proyecto donde van a guardar sus datos, scripts y notebook.
- Recuerden iniciarlo con `jupyter notebook` y cerrarlo con `Ctrl + c`.
- A lo largo del curso ocuparemos muchos atajos para facilitar nuestro trabajo.

Acción	Comando
Autocompletar	`Tab`
Leer documentación	`Shift + Tab`
Ejecutar celda	`Shift + Enter`
Paleta de Comandos	`Shift + Cmd + P`

Los principales elementos de Python

{desafío}
latam_

Sintáxis

- A diferencia de otros lenguajes, Python busca ser de fácil lectura:

- Utiliza menos ornamentos que otros.

- Prioriza el uso de palabras en inglés por sobre operadores lógicos.

- Delimita los flujos mediante bloques indentados.

Variables

- Una variable es un contenedor de valores y/o funciones definidas por el usuario.
- Nos permiten guardar temporalmente los valores de una operación para reutilizarla posteriormente.

In [26]:
```
variable = 5
print(variable)
```

```
5
```

Declaraciones

- Son las órdenes que se instruyen al intérprete de Python.

In [27]:
```
# esta es una declaración
24 + 5 / 3
```

Out[27]:
```
25.666666666666668
```

- Existen tres tipos de declaraciones:

Declaraciones de asignación

- Cuando instruímos **guardar** una expresión a una variable.

```
In [28]: variable = 5
```

Declaraciones de impresión

- Cuando instruímos **evaluar** una expresión.

```
In [29]: print(variable)
```

```
5
```

Declaraciones de importación

- Cuando instruímos **incorporar** una librería.

In [30]:
```
import pandas as pd
```

Librerías

- Parte importante del trabajo orientado a los datos con Python es la utilización de librería.
- Uno de los puntos más fuertes de Python es la amplia variedad de librerías y su mantención. Para el trabajo de análisis de datos

Datos

- Las declaraciones con las que hemos trabajado requieren datos que dependen de su naturaleza.
- En Python existen 4 grandes tipos de datos:

Strings

- Secuencias alfanuméricas de valores literales.
- Permiten almacenar elementos como oraciones o nombres.

In [31]:
```
esta_es_una_cadena = "Esta es una cadena"
```

In [32]:
```
print(esta_es_una_cadena)
```

```
Esta es una cadena
```

In [33]:
```
print(type(esta_es_una_cadena))
```

```
<class 'str'>
```

Integer

- Datos numéricos que representa una cantidad finita.

```
In [34]: entero = 5

In [35]: print(entero)
         5

In [36]: type(entero)
Out[36]: int
```

Float

- Datos numéricos que reprensentan números racionales (con decimales).

In [37]:
```
flotante = 2.5
```

In [38]:
```
print(flotante)
```

```
2.5
```

In [39]:
```
type(flotante)
```

Out[39]:
```
float
```

{desafío}
latam_

Booleano

- Representaciones de una expresión lógica, que toma los valores `True` o `False`.

```
In [40]: perro = True
```

```
In [41]: print(perro)
```

```
True
```

```
In [42]: type(perro)
```

```
Out[42]: bool
```

Medidas de Tendencia Central

{desafío}
latam_

Media

Resume el comportamiento de una variable en una cifra.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

In [43]:
```
altura_presidentes = [1.93, 1.92, 1.98, 1.88, 1.88,1.92, 1.98,
                      1.88, 1.88,1.92, 1.98, 1.88, 1.88]

sum(altura_presidentes) / len(altura_presidentes)
```

Out[43]:
```
1.916153846153846
```

In [44]:
```
# De forma alternativa lo podemos realizar con numpy
import numpy as np
np.mean(altura_presidentes)
```

Out[44]:
```
1.916153846153846
```

Moda

Es el valor que más se repite en una variable

- Su implementación en Python nativo es un compleja. Utilicemos `scipy.stats`

In [45]:
```
import scipy.stats as stats
stats.mode(altura_presidentes)
```

Out[45]:
```
ModeResult(mode=array([1.88]), count=array([6]))
```

Mediana

Punto equidistante en una variable.

- La forma más fácil de obtener la mediana es con `numpy` o pandas

```
In [46]: np.median(altura_presidentes)
```

```
Out[46]: 1.92
```

Medidas de Dispersión

- Una vez que ya localizamos el punto donde se concentra la mayoría de los casos, el segundo momento es ver qué tan dispersos están alrededor de la tendencia central. Para ello nos valemos de las medidas de dispersión.

Varianza

- Si tenemos un vector $n \in X : x_1, x_2, \ldots, x_n$, la varianza se puede obtener a partir de la siguiente fórmula:

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N - 1}$$

In [47]:
```
np.var(altura_presidentes)
```

Out[47]:
```
0.001562130177514796
```

Desviación Estándar

$$\text{DesvEst}(x) = \sqrt{\text{Var}(x)} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}$$

In [48]:
```
np.std(altura_presidentes)
```

Out[48]:
```
0.039523792549738895
```