

ACÀMICA

Agenda

Repaso: Numpy, bitácora y challenge.

Explicación: Máscara, Pandas.

Break.

Hands-on training: Pandas

Cierre.



Estadística y Pandas

Hoy repasaremos algunos conceptos estadísticos, en particular Estadística Descriptiva. Luego, veremos cómo hacemos en Python para trabajar con conjuntos de datos usando Pandas.

REPASO

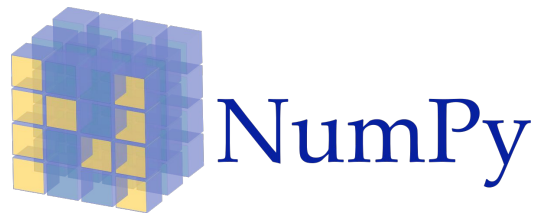


Repaso del encuentro pasado



Numpy: nuestra primer librería

Nuestra primera librería:



- Fundamental para hacer cálculo numérico con Python
- Muy buena [documentación](#)
- Como muchas librerías, trae una estructura de datos propia: los **arrays** o arreglos.

Numpy: arrays

array: a primer orden, es como una **lista**. De hecho, se pueden crear a partir de una lista.

Importamos la librería
(*numpy*) y le ponemos un
nombre (*np*)

```
[1]: import numpy as np
```

```
arreglo = np.array([1,2,3,4,5])  
arreglo
```

Es una lista

```
[1]: array([1, 2, 3, 4, 5])
```

```
[2]: print(arreglo)
```

```
[1 2 3 4 5]
```

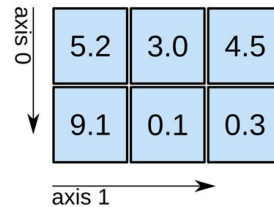
Numpy: arrays

1D array



shape: (4,)

2D array

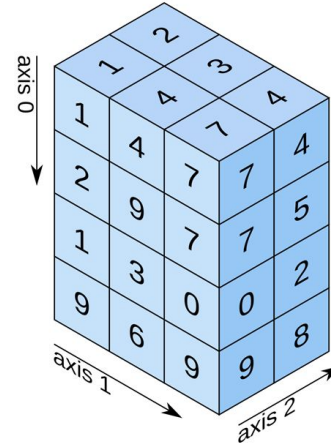


shape: (2, 3)

filas

columnas

3D array



shape: (4, 3, 2)

Numpy

¿Qué hacen cada una de estas instrucciones?

- `np.arange()`
- `np.linspace()`
- `np.zeros()`
- `np.ones()`

Numpy

¿Qué hacen cada una de estas instrucciones?

- **np.arange()**

```
[2]: arreglo = np.arange(3,21,2)  
arreglo
```

- np.linspace()

```
[2]: array([ 3,  5,  7,  9, 11, 13, 15, 17, 19])
```

- np.zeros()

- np.ones()

Numpy

¿Qué hacen cada una de estas instrucciones?

- `np.arange()`
- **`np.linspace()`**
- `np.zeros()`
- `np.ones()`

```
[3]: arreglo = np.linspace(3,21,15)  
arreglo
```

```
[3]: array([ 3.          ,  4.28571429,  5.57142857,  6.85714286,  8.14285714,  
           9.42857143, 10.71428571, 12.          , 13.28571429, 14.57142857,  
          15.85714286, 17.14285714, 18.42857143, 19.71428571, 21.          ])
```

Numpy

¿Qué hacen cada una de estas instrucciones?

- `np.arange()`
- `np.linspace()`
- **`np.zeros()`**
- `np.ones()`

```
[4]: arreglo = np.zeros(10)
      arreglo
```

```
[4]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
[5]: arreglo = np.zeros((3,5))
      arreglo
```

```
[5]: array([[0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0.]])
```

Numpy

¿Qué hacen cada una de estas instrucciones?

- `np.arange()`
- `np.linspace()`
- `np.zeros()`
- **`np.ones()`**

```
[4]: arreglo = np.zeros(10)
      arreglo
```

```
[4]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
[5]: arreglo = np.zeros((3,5))
      arreglo
```

```
[5]: array([[0., 0., 0., 0., 0.],
           [0., 0., 0., 0., 0.],
           [0., 0., 0., 0., 0.]])
```

¿Qué cambia?

Operaciones lógicas

Un tipo importante de operación en programación son las **operaciones lógicas**. Estas pueden realizarse sobre **variables booleanas**.

```
In [27]: variable_1 = True  
         variable_2 = False  
         print(variable_1 or variable_2)
```

True

```
In [28]: print(not(variable_1))
```

False

El resultado es también una **variable booleana**.

A	B	A & B
False	False	False
False	True	False
True	False	False
True	True	True

A	B	A or B
False	False	False
False	True	True
True	False	True
True	True	True

A	A!
False	True
True	False

CONDICIONALES - if / elif / else

Además del **if** y el **else**, uno puede agregar más condiciones a través de condicional **elif** (else if). De esta forma se puede agregar un número arbitrario de condiciones.

```
In [80]: edad = 20

if edad < 18:
    print('Esta persona tiene menos de 18 años')
elif edad > 18:
    print('Esta persona tiene mas de 18 años')
else:
    print('Esta persona tiene justo 18 años')
```

Esta persona tiene mas de 18 años

REPASO

EJERCICIO DEL ENCUENTRO PASADO

¡Muéstranos qué hiciste!

¿Qué cosas te costaron más del ejercicio? ¿Cómo las resolviste?

¿Cuál el principal aprendizaje que te llevas?

Si tuvieras que hacerle alguna recomendación a alguien que va a hacer el ejercicio por primera vez, ¿qué le dirías?



REPASO

EJERCICIO DEL ENCUENTRO PASADO

¿Alguien hizo algo diferente que quiera mostrar?



Repaso de la bitácora



REPASO

TEMAS BITÁCORA



Probabilidad y Estadística



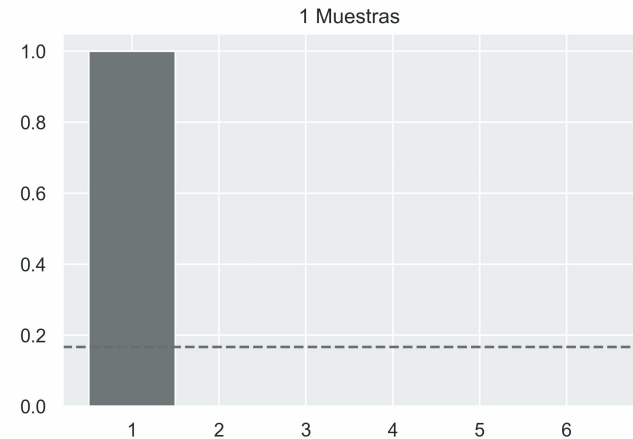
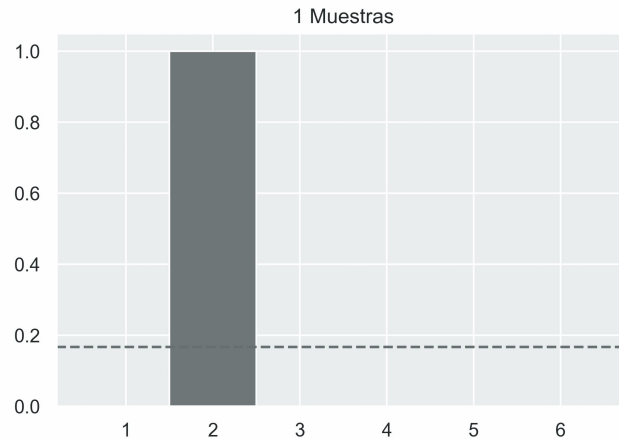
Probabilidad y Estadística

Tenemos dos dados. Suponemos que uno está cargado.
¿Cómo nos damos cuenta cuál?



Probabilidad y Estadística

Tenemos dos dados. Suponemos que uno está cargado.
¿Cómo nos damos cuenta cuál?



Tipos de valores estadísticos:

Media: es el valor promedio estándar (lo que siempre conocimos por promedio).

Mediana: es el valor medio exacto en un conjunto de datos ordenados. Es decir, el 50% de los valores son menores que la media y el 50% son mayores.

Moda: el valor con mayor frecuencia en un conjunto de datos.

Ejemplo

Muestra: {5, 6, 7, 6, 7, 8, 6, 5, 6}

- **Media = 6.22**
- **Mediana = 6**
5, 5, 6, 6, 6, 6, 7, 7, 8
- **Moda = 6**
5, 5, 6, 6, 6, 6, 7, 7, 8

Varianza

Mide la variabilidad o dispersión de un conjunto de números (*muestra*).

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

Varianza

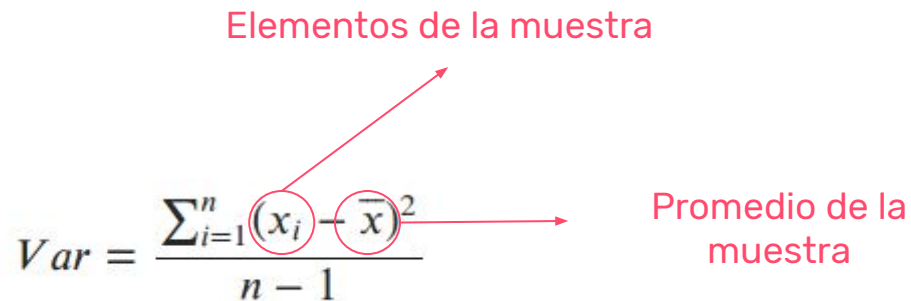
Mide la variabilidad o dispersión de un conjunto de números (*muestra*).

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

Promedio de la
muestra

Varianza

Mide la variabilidad o dispersión de un conjunto de números (*muestra*).



The diagram shows the formula for sample variance:
$$Var = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$
 Two red annotations are present: 1. The text "Elementos de la muestra" (Sample elements) with a red arrow pointing to the x_i term in the numerator. 2. The text "Promedio de la muestra" (Sample mean) with a red arrow pointing to the \bar{x} term in the numerator.

Elementos de la muestra

Promedio de la muestra

Varianza

Mide la variabilidad o dispersión de un conjunto de números (*muestra*).

El símbolo de sumatoria nos indica que debemos sumar sobre todos los valores del conjunto

Elementos de la muestra

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

Promedio de la muestra

Varianza

Mide la variabilidad o dispersión de un conjunto de números (*muestra*).

El símbolo de sumatoria nos indica que debemos sumar sobre todos los valores del conjunto

The diagram shows the formula for sample variance:
$$Var = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$
 Four red arrows point from text labels to parts of the formula: 1. From 'Elementos de la muestra' to the x_i term. 2. From 'Promedio de la muestra' to the \bar{x} term. 3. From 'Cantidad de elementos en la muestra' to the n in the denominator. 4. From 'El símbolo de sumatoria nos indica que debemos sumar sobre todos los valores del conjunto' to the summation symbol \sum .


Elementos de la muestra

Promedio de la muestra


Cantidad de elementos en la muestra

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

Veamos un ejemplo:

- Muestra: {5, 10, 8, 20} 
- N es 4
- El promedio, \bar{X} es 10,75

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


$$Var = \frac{(5-10,75)^2 + (10-10,75)^2 + (8-10,75)^2 + (20-10,75)^2}{4-1}$$

CHALLENGE BITÁCORA



1. ¿Qué es cuartil?
¿Y un percentil?
¿Por qué hay variabilidad en los datos?
2. ¿Resolviste el challenge del notebook?



CHALLENGE BITÁCORA



¿Alguien hizo algo diferente que quiera mostrar?



Máscaras



Máscaras - Filtros Booleanos

```
[66]: arreglo2d = np.arange(30).reshape(6,5)  
arreglo2d
```

```
[66]: array([[ 0,  1,  2,  3,  4],  
            [ 5,  6,  7,  8,  9],  
            [10, 11, 12, 13, 14],  
            [15, 16, 17, 18, 19],  
            [20, 21, 22, 23, 24],  
            [25, 26, 27, 28, 29]])
```

Máscaras - Filtros Booleanos

```
[66]: arreglo2d = np.arange(30).reshape(6,5)  
arreglo2d
```

```
[66]: array([[ 0,  1,  2,  3,  4],  
          [ 5,  6,  7,  8,  9],  
          [10, 11, 12, 13, 14],  
          [15, 16, 17, 18, 19],  
          [20, 21, 22, 23, 24],  
          [25, 26, 27, 28, 29]])
```

→
Creamos la
máscara

```
[67]: mask = arreglo2d < 20  
mask
```

```
[67]: array([[ True,  True,  True,  True,  True],  
          [ True,  True,  True,  True,  True],  
          [ True,  True,  True,  True,  True],  
          [ True,  True,  True,  True,  True],  
          [False, False, False, False, False],  
          [False, False, False, False, False]])
```

Máscaras - Filtros Booleanos

```
[66]: arreglo2d = np.arange(30).reshape(6,5)  
arreglo2d
```

```
[66]: array([[ 0,  1,  2,  3,  4],  
          [ 5,  6,  7,  8,  9],  
          [10, 11, 12, 13, 14],  
          [15, 16, 17, 18, 19],  
          [20, 21, 22, 23, 24],  
          [25, 26, 27, 28, 29]])
```

→
Creamos la
máscara

```
[67]: mask = arreglo2d < 20  
mask
```

```
[67]: array([[ True,  True,  True,  True,  True],  
          [ True,  True,  True,  True,  True],  
          [ True,  True,  True,  True,  True],  
          [ True,  True,  True,  True,  True],  
          [False, False, False, False, False],  
          [False, False, False, False, False]])
```

```
[68]: arreglo2d[mask]
```

```
[68]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,  
          17, 18, 19])
```

←
Y seleccionamos aquellos
elementos que cumplen la
condición que representa
la máscara

Pandas



DATASET

Es el conjunto de datos que utilizaremos en el workflow de data science. Los podemos generar, obtener de terceros o simular.

datasets
estructurados

similar a planilla de cálculo. Información pre-procesada. Suelen venir en .txt, .csv, .xlsx, .json, etc.

datasets
no estructurados

audio, imágenes, texto en crudo
humanos / redes neuronales



DATASET

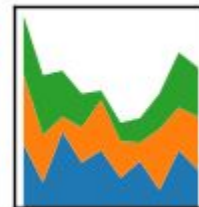
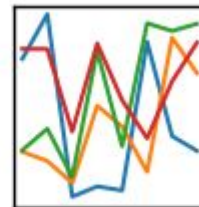
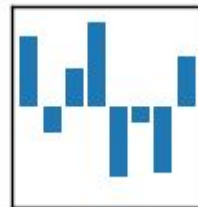
datasets
estructurados

similar a planilla de cálculo. Información pre-procesada. Suelen venir en .txt, .csv, .xlsx, .json, etc.

Para trabajar con datasets estructurados (y bueno, más), la librería estándar de Python es:

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



ARGENTINA DATASET

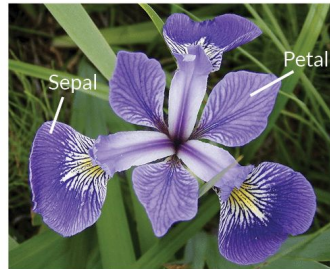


División Política,
Superficie y
Población



IRIS DATASET

Famoso dataset introducido por Ronald Fisher (padre de la estadística) en 1936.



Iris Versicolor



Iris Setosa



Iris Virginica



A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup is placed on a matching white saucer. In the background, a white napkin and a silver spoon are visible, though they are out of focus. The overall lighting is soft and even, highlighting the textures of the coffee and the smooth surface of the cup.

¡BREAK!

Pandas: Instalación

1. Activar el ambiente: *"conda activate datascience"*
2. Instalar Pandas: *"conda install pandas"*

Hands-on training





Trabajamos en el Notebook que descargaste en la bitácora 04, Sección 2: Pandas

Buenas prácticas de un data scientist



Buenas prácticas de un ~~data scientist~~ programador



PEP-20: The Zen of Python

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one -and preferably only one- obvious way to do it.
Although that may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking idea --let's do more of those!



Recursos



Probabilidad y Estadística

- <https://seeing-theory.brown.edu/basic-probability/index.html> - Este recurso no aparece en la bitácora, pero pueden mirarlo si tienen tiempo.

Pandas

- [Python Data Science Handbook](#) - Capítulo 3, “Data Manipulation With Pandas”.

encuesta “INICIO”

¡Queremos escucharte!



encuesta “INICIO”

Para la próxima

- Termina el notebook de hoy
- Lee la bitácora 05 y carga las dudas que tengas al Trello
- Resuelve el Challenge.

En el encuentro que viene uno/a de ustedes será seleccionado/a para mostrar cómo resolvió el challenge de la bitácora. De esta manera, ¡aprendemos todos/as de (y con) todas/as, así que vengan preparados/as.

ACÀMICA