

PANDAS 101





Mauricio Langleib

Clase 01: Estructuras de datos en Python. Datos numéricos, cadenas de caracteres, listas, conjuntos y diccionarios.



Joaquín Pereira

Clase 02: Control de flujo. Operaciones lógicas, uso de condicionales y bucles. Introducción a *list comprehensions*.



Juan Trinidad

Clase 03: Definición de funciones y el alcance de variables. Presentación de los módulos 'sys', 'os', 're', 'glob' y 'subprocess'. Creación de scripts y el uso de 'argparse' para manejar argumentos desde la línea de comandos.



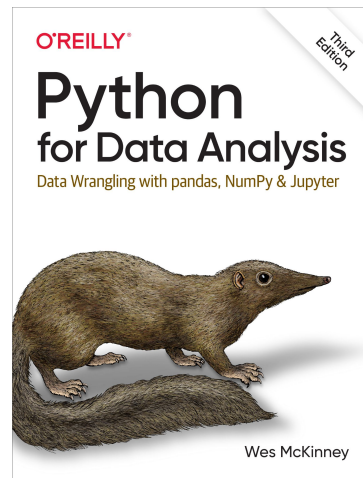
Héctor Romero

Clase 04: Librería Pandas y los DataFrames. Importación y exploración de datos desde archivos CSV/TSV.

Clase 05: Introducción a la visualización de datos con 'seaborn'.



Ningún docente es ingeniero en computación... tenemos una aproximación pragmática al tema !



Aspectos más importantes

- Pandas es una herramienta de análisis, con funcionalidades similares a SQL.
- Incluye soporte estadístico, analítico y capacidades de visualización de datos.
- Se basa en Cython y NumPy (PyArrow), lo que le otorga menor uso de memoria y mayor velocidad que Python puro.
- Se utiliza comúnmente para:
 - Reemplazar Excel.
 - Procesos ETL (extraer, transformar y cargar datos).
 - Procesar datos tabulares.
 - Cargar archivos CSV o JSON.
 - Preparación de datos para machine learning.
- Al usar operaciones vectorizadas, se alcanzan velocidades cercanas al nivel C, manteniendo la facilidad de escribir en Python.



Pandas: DataFrame

Column Label/ Header		0	1	2	3	4	
Index Label		Name	Age	Marks	Grade	Hobby	Column Index
0	S1	Joe	20	85.10	A	Swimming	
1	S2	Nat	21	77.80	B	Reading	
2	S3	Harry	19	91.54	A	Music	
3	S4	Sam	20	88.78	A	Painting	Row
4	S5	Monica	22	60.55	B	Dancing	
				Column		Element/ Value/ Entry	

Importar Datos

```
import pandas as pd
```

#Desde un archivo csv.

```
df = pd.read_csv('data.csv')
```

#Desde un archivo excel.

```
df = pd.read_excel('data.xlsx', sheet_name='Sheet1')
```

#Desde un json

```
df = pd.read_json('data.json')
```

```
import pandas as pd
```

```
df = pd.DataFrame({'growth': [.5, .7, 1.2], 'Name': ['Paul', 'George',  
'Ringo']})
```

#visualmente más claro.

```
df = pd.DataFrame({  
    'growth': [.5, .7, 1.2],  
    'Name': ['Paul', 'George', 'Ringo']  
})
```

Atributos de un DataFrame

- index
- columns
- axes
- dtypes
- size
- shape
- ndim
- empty
- T
- values

Algunas funciones de un DataFrame

- **df.head()** – Muestra las primeras filas del DataFrame (vista rápida del dataset).
- **df.info()** – Muestra información general: tipos de datos, cantidad de filas/columnas, presencia de nulos.
- **df.describe()** – Estadísticas descriptivas de columnas numéricas.
- **df.isnull()** / **df.isna()** – Identifica valores faltantes.
- **df.fillna()** – Rellena valores faltantes.
- **df.drop()** – Elimina columnas o filas.
- **df.sort_values()** – Ordena el DataFrame por una o más columnas.
- **df.groupby()** – Agrupa datos para cálculos agregados (sumas, medias, conteos, etc.).
- **df.loc[]** / **df.iloc[]** – Selección de datos por etiqueta (loc) o por posición (iloc).
- **df.merge()** / **df.join()** – Combina DataFrames (similar a SQL JOIN).

ARGENTINO
DICE



Reglas:

- Cada pregunta tiene un puntaje, si se contesta correctamente se suma, si es incorrecto se resta.
- No hay caminos correctos hay respuestas correctas.
- El tiempo se termina cuando un jugador dice "TUTÍ".
- Hay que tener al menos 5 puntos para cortar.

ROUND 0

1. Cuantas columnas y filas tiene el DataFrame de Pokemon? (1 punto)
2. Qué tipo de dato tienen las columnas? float, int, object, str... (1 punto)
3. Cuantas columnas de cada tipo de dato tenemos? (3 punto)
4. Cuantas columnas comienzan con la letra "a" ? (3 punto)
5. Cuantas columnas tienen valores nulos? Cual es la que tiene más valores no nulos?
(3 puntos)
6. Hay una columna cuyo tipo de dato no se condice con lo que uno esperaría el ver los valores manualmente, cual es? y cual es la razón por la que no tiene el tipo de dato esperado? (10 puntos)

ROUND 1

1. Convierta los nombres de los Pokémon en el index del DataFrame? (5 punto)
2. Cuales son los 10 Pokemon más altos? (5 punto)
3. Cuantos Pokemon hay en cada generación? (5 punto)
4. Sume las columnas "sp_attack", "sp_defense", "speed". Cual es el Pokemon con mayor valor en la suma? (10 punto)
5. Qué Pokemon tienen el mismo valor en las columnas "type1" y "type2"? (10 punto)
6. Calcule el promedio de altura para cada generación de Pokémon? (20 punto)

ROUND 2

1. Cuantos pokemon legendarios y no legendarios hay? (5 punto)
2. Cual es el pokemon con más masa y el de menos (Kg)? (5 punto)
3. Si modelamos los pokemon de forma cuadrada, cuál es el pokémon más denso? (30 punto)
4. Realice una normalización de tipo Z-score en la columna de "weight_kg". Cual es el desvío estándar? (30 punto)
5. Cual es el valor de la columna "base_total" del pokemon "Bronzor"? (10 punto)
6. El densidad del pokemon se puede calcular utilizando la función **apply()** o realizando operaciones matriciales entre columnas. Cual es más rápido y por cuanto? Pista, usar **%%time** magic line. (40 punto)
7. Utilizando el archivo "colores_pokemon.csv" agregue a cada pokemon su color correspondiente en el DataFrame original. (30 punto)