Introducción a Python Científico

Entornos, NumPy, Pandas y Matplotlib

Cesar Garcia

2025

Introducción

Objetivos de la sesión

- Crear y gestionar entornos de Python.
- Entender qué son NumPy, Pandas y Matplotlib.
- Realizar ejemplos básicos con cada librería.
- Visualizar datos de manera sencilla.

Sección 1

Creación de un entorno de Python

¿Por qué usar entornos?

- Mantener versiones específicas de librerías.
- Evitar conflictos entre proyectos.
- Reproducibilidad.
- Aislamiento completo del sistema global.

Crear un entorno con venv

1. Crear el entorno

```
python3 -m venv .venv
```

2. Activar el entorno

- **Linux / macOS**

```
source .venv/bin/activate
```

- **Windows**

```
.\.venv\Scripts^^Gctivate
```

3. Instalar paquetes

```
pip install numpy pandas matplotlib
```

Verificar el entorno activo

```
which python
which pip
pip list
```

Desactivar el entorno

```
deactivate
```

Sección 2

NumPy

¿Qué es NumPy?

- Librería fundamental para cálculo numérico.
- Provee estructuras de datos eficientes: **ndarray**.
- Operaciones vectorizadas (muy rápidas).
- Funciones matemáticas avanzadas.

Crear arreglos

```
import numpy as np

a = np.array([1, 2, 3])
b = np.zeros((2, 3))
c = np.random.rand(3, 3)

print(a)
print(b)
print(c)
```

Operaciones element-wise

```
import numpy as np

x = np.array([1, 2, 3])
y = np.array([4, 5, 6])

print(x + y)
print(x * y)
print(np.dot(x, y))
```

Estadísticas rápidas

```
mat = np.random.randn(4, 4)

mat.mean()
mat.std()
mat.sum()
mat.max()
```

Sección 3

Pandas

¿Qué es Pandas?

- Librería para **análisis de datos**.
- Estructuras principales:
  - **Series** (vector)
  - **DataFrame** (tabla)
- Ideal para datos CSV, Excel, JSON, SQL.

Crear un DataFrame

```
import pandas as pd

datos = {
    "Nombre": ["Ana", "Luis", "Cesar"],
    "Edad": [23, 34, 59],
    "Ciudad": ["CDMX", "Madrid", "Austin"]
}

df = pd.DataFrame(datos)
print(df)
```

Cargar datos desde CSV

```
df = pd.read_csv("datos.csv")
df.head()
df.info()
df.describe()
```

Filtros y selección

```
df[df["Edad"] > 30]
df[["Nombre", "Ciudad"]]
df.sort_values("Edad", ascending=False)
```

Agregaciones

```
df.groupby("Ciudad")["Edad"].mean()
```

Sección 4

Matplotlib

¿Qué es Matplotlib?

- Librería para visualización de datos.
- Permite crear:
  - Gráficas de líneas
  - Barras
  - Histogramas
  - Dispersión
  - Subplots

Gráfica básica

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.plot(x, y)
plt.title("Función seno")
plt.xlabel("x")
plt.ylabel("sin(x)")
plt.show()
```

Gráfica de dispersión

```
x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y)
plt.title("Scatter Plot")
plt.show()
```

Histogramas

```
data = np.random.randn(1000)
plt.hist(data, bins=30)
plt.title("Histograma")
plt.show()
```

Ejemplo final

Combinar Pandas + Matplotlib

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("ventas.csv")
df.groupby("mes")["monto"].sum().plot(kind="bar")

plt.title("Ventas por mes")
plt.ylabel("Monto")
plt.show()
```

Conclusiones

- Los entornos permiten aislar proyectos.
- NumPy $\rightarrow$ cálculo numérico eficiente.
- Pandas $\rightarrow$ manipulación de datos tabulares.
- Matplotlib $\rightarrow$ visualización clara y flexible.

¡Gracias!

Preguntas