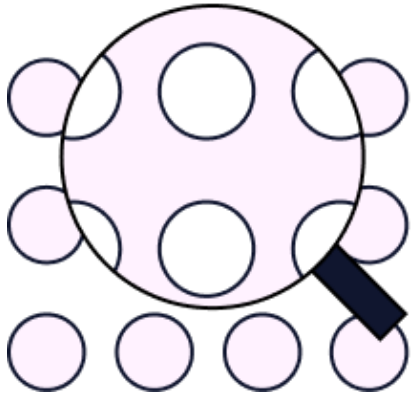
Jul. 2022

Unidad 8: Análisis de datos



Objetivos de aprendizaje

- Utilizar Python para leer y transformar datos en diferentes formatos
- Convertir datos de diferentes fuentes en formatos de almacenamiento o consulta
- Preparar los datos para el análisis estadístico, la visualización, el aprendizaje automático, etc.
- Generar estadísticas y métricas básicas
- Presentar los datos en forma de visualizaciones eficaces

Los cuadernos Jupyter (jupyter.org)

- Es la piedra angular de nuestro proceso de análisis
- Es una excelente plataforma para desarrollar código y comunicar resultados
- Se basa en la extensión del modelo de shell interactivo
  - creando documentos que pueden ejecutar código
  - mostrar documentación
  - presentar resultados como gráficos e imágenes
- Jupyter es una aplicación web
  - Se usa a través de Internet
  - Se ejecuta en el navegador web
  - Los cuadernos pueden compartirse

Los cuadernos Jupyter *(cont.)*

- Es compatible con el lenguaje de marcado Markdown
- La unidad básica de un cuaderno se llama celda.
- Una celda es un contenedor de código o de texto.
  - Una **celda de código** acepta código para ser ejecutado en el núcleo y mostrar la salida justo debajo.
  - Una **celda de texto** acepta Markdown y analizará y formateará cuando se ejecute la celda.

E1: Introducción a los cuadernos de Jupiter

1. Hola mundo
2. Operaciones aritméticas
3. Recuperar el valor del último objeto devuelto ("_" guión bajo)
4. Asignación de variables
5. Creación de funciones
6. Celdas markdown
7. Comandos del sistema operativo (! antes del comando)
8. Listar los comandos mágicos (%lsmagic)
9. Crear un archivo de texto (%%writefile)
10. Leer el contenido del fichero creado (open)
11. Mostrar la ayuda (%comando?)
12. Sistema de visualización sofisticados (from IPython.display import HTML, SVG, YouTubeVideo)

Componentes de la pila de ciencia de datos

- **NumPy**: Un paquete de manipulación numérica
- **pandas**: Una biblioteca de manipulación y análisis de datos
- **SciPy**: Una colección de algoritmos matemáticos construidos sobre NumPy
- **Matplotlib**: Una biblioteca de trazado y gráficos

1. NumPy (numpy.org)

- Excelente para manipular matrices multidimensionales
- Aplica funciones de álgebra lineal o estadísticas a esas matrices
- Motor numérico de un gran número de paquetes de Python
  - incluyendo pandas y scikit-learn
- Importar el paquete

```
import numpy as np
```

2. SciPy (scipy.org)

- Parte del ecosistema de bibliotecas para muchas áreas científicas
  - Matemáticas, la ciencia y la ingeniería.
  - NumPy, SciPy, scikit-learn

3. Matplotlib (matplotlib.org)

- Genera figuras en una variedad de formatos
- Está inspirada en la interfaz de trazado de MATLAB
- Como fuentes de datos puede utilizar
  - Datos nativos de Python
  - Arrays de NumPy
  - DataFrames de pandas
- Se considera de bajo nivel
  - se necesitan varias líneas de código para generar un gráfico
- Una de las extensiones (la biblioteca Seaborn)
- Se puede acceder a ella a través del módulo *matplotlib.pyplot.*
  - import matplotlib.pyplot as plt

Ejercicio

- matplotlib.ipynb

4. Pandas (pandas.pydata.org)

- Es una biblioteca de manipulación y análisis de datos
- Diseñada para trabajar con datos tabulares o etiquetados
  - similares a tablas SQL y archivos de Excel
- Dos estructuras de datos básicas:
  - Series (estructura de datos unidimensional)
  - DataFrame (estructura de datos bidimensional que soporta índices)
- Los datos en DataFrames y series pueden
  - estar ordenados o desordenados
  - ser homogéneos o heterogéneos
- Importar
  - import pandas as pd

- https://pandas.pydata.org/pandas-docs/dev/user_guide/visualization.html

Resumen

Hemos:

- aprendido sobre las bibliotecas de Python más comunes utilizadas en el análisis y la ciencia de datos que conforman la pila de la ciencia de datos de Python.
- aprendido cómo ingerir datos, seleccionarlos, filtrarlos y agregarlos.
- visto cómo exportar los resultados de nuestro análisis y generar algunas gráficas.