P10. Análisis de Datos en Python

Alumna: Daniela Mendez Ramirez Número
de Cuenta: 258331-9

Aplicaciones de Redes

Profesor: Omar Vázquez González

Fecha de Entrega: 26 de febrero de 2025

Abstract

Este reporte documenta una práctica orientada al análisis y visualización de datos utilizando Python, haciendo especial énfasis en las librerías Pandas y Matplotlib. Se exploraron diversas técnicas para la manipulación, transformación y representación gráfica de datos, destacando la eficiencia y flexibilidad de estas herramientas en la resolución de problemas reales. A lo largo del informe, se presentan ejemplos prácticos que demuestran cómo el uso de Pandas facilita la limpieza y el análisis de grandes conjuntos de datos, mientras que Matplotlib permite generar visualizaciones intuitivas que apoyan la toma de decisiones. El objetivo principal es evidenciar la aplicabilidad de estas librerías en entornos académicos y profesionales, promoviendo metodologías de análisis de datos accesibles y reproducibles.

Introducción

El análisis de datos se ha convertido en una disciplina esencial en numerosos campos, desde la investigación científica hasta la toma de decisiones empresariales. En este contexto, Python ha emergido como uno de los lenguajes de programación más populares, gracias a su sintaxis clara y la extensa oferta de librerías especializadas. Entre ellas, Pandas y Matplotlib destacan por su capacidad para facilitar tanto la manipulación como la visualización de datos.

Pandas ofrece estructuras de datos potentes y versátiles, permitiendo realizar operaciones complejas de transformación, filtrado y agregación de datos de forma eficiente. Esta librería se ha consolidado como una herramienta fundamental para el preprocesamiento y análisis exploratorio de datos. Por otro lado, Matplotlib proporciona un conjunto robusto de herramientas para la creación de gráficos y visualizaciones personalizadas, lo que resulta crucial para interpretar y comunicar de manera efectiva los resultados del análisis.

El presente reporte describe una serie de prácticas orientadas a la aplicación de estas herramientas en proyectos reales. Se busca no solo demostrar el potencial de Pandas y Matplotlib en la solución de problemas de análisis de datos, sino también fomentar su utilización en entornos educativos y profesionales, donde la capacidad de transformar datos en información relevante es cada vez más demandada.

Ejercicios

1. Cargar archivos csv

```
import pandas as pd

# Cargar el dataset
df = pd.read_csv('ruta_del_archivo.csv')

# Explorar el dataset
print(df.head())
print(df.info())
print(df.describe())
```

2. Limpieza

```
# Detectar valores nulos
print(df.isnull().sum())

# Eliminar filas con valores nulos
df = df.dropna()

# Eliminar duplicados
df = df.drop_duplicates()

print(df.info())
```

3. Ejercicio 3: Análisis y agrupación de datos con Pandas

```
# Agrupar por una columna categórica (por ejemplo, "Categoria")
agrupado = df.groupby('Categoria')['Valor'].mean()
print(agrupado)
```

4. Ejercicio 4: Visualización de datos básicos con Matplotlib

```
import matplotlib.pyplot as plt

# Gráfico de líneas
plt.figure(figsize=(8, 4))
plt.plot(df['Fecha'], df['Valor'], marker='o')
plt.xlabel('Fecha')
plt.ylabel('Valor')
plt.title('Evolución de Valor en el Tiempo')
plt.show()

# Gráfico de barras
plt.figure(figsize=(8, 4))
plt.bar(df['Categoria'], df['Valor'])
plt.xlabel('Categoría')
plt.ylabel('Valor')
plt.title('Comparación de Valores por Categoría')
plt.show()
```

```
# Gráfico de dispersión
plt.figure(figsize=(8, 4))
plt.scatter(df['Variable1'], df['Variable2'])
plt.xlabel('Variable 1')
plt.ylabel('Variable 2')
plt.title('Relación entre Variable 1 y Variable 2')
plt.show()
```

5. Ejercicio 5: Integración de Pandas y Matplotlib

```
# Agrupar datos y contar registros por categoría
conteo = df.groupby('Categoria').size()

# Convertir a DataFrame para facilitar la visualización
conteo_df = conteo.reset_index(name='Cantidad')

# Crear gráfico de barras
plt.figure(figsize=(8, 4))
plt.bar(conteo_df['Categoria'], conteo_df['Cantidad'])
plt.xlabel('Categoría')
plt.ylabel('Cantidad')
plt.title('Cantidad de Registros por Categoría')
plt.show()
```

6. Ejemplo 1: Carga y Exploración del Dataset

```
import pandas as pd

# Cargar el dataset
df = pd.read_csv('olympics.csv')

# Mostrar las primeras filas para ver la estructura del dataset
print("Primeras filas:")
print(df.head())

# Información general y tipos de datos
print("\nInformación del dataset:")
print(df.info())

# Estadísticas descriptivas de las columnas numéricas
print("\nEstadísticas descriptivas:")
print(df.describe())
```

7. Ejemplo 2: Ranking de Países por Número de Medallas

```
import matplotlib.pyplot as plt

# Ordenar el DataFrame de mayor a menor según el número de medallas
df_sorted = df.sort_values(by='medallas', ascending=False)

# Mostrar los 10 países con mayor cantidad de medallas
top10_paises = df_sorted[['pais', 'medallas']].head(10)
print("Top 10 países por medallas:")
print(top10_paises)

# Crear gráfico de barras para visualizar los 10 países con más medallas
plt.figure(figsize=(10, 6))
plt.bar(top10_paises['pais'], top10_paises['medallas'], color='skyblue')
plt.xlabel("País")
plt.ylabel("Número de Medallas")
plt.title("Top 10 países por número de medallas")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

8. Ejemplo 3: Participación de Personas por Deporte

```
# Agrupar los datos por deporte y sumar el número de personas participantes
personas_por_deporte = df.groupby('deporte')['personas'].sum().reset_index()

# Ordenar de mayor a menor según el número de personas
personas_por_deporte = personas_por_deporte.sort_values(by='personas',
ascending=False)
print("Participación por deporte:")
print(personas_por_deporte)

# Crear gráfico de barras para visualizar la participación de personas por
deporte
plt.figure(figsize=(10, 6))
plt.bar(personas_por_deporte['deporte'], personas_por_deporte['personas'],
color='salmon')
plt.xlabel("Deporte")
plt.ylabel("Número de Personas")
plt.title("Participación de personas por deporte")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

9. Ejemplo 4: Evolución de Medallas a lo Largo de los Años

```
# Agrupar los datos por la columna 'anios' y sumar el número de medallas por año
medallas_por_anio = df.groupby('anios')['medallas'].sum().reset_index()

# Mostrar la evolución de medallas a lo largo de los años
print("Medallas por año:")
print(medallas_por_anio)

# Crear gráfico de líneas para visualizar la tendencia de medallas
plt.figure(figsize=(10, 6))
plt.plot(medallas_por_anio['anios'], medallas_por_anio['medallas'], marker='o',
linestyle='-', color='green')
plt.xlabel("Años")
plt.ylabel("Número de Medallas")
plt.title("Evolución de medallas a lo largo de los años")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

10. Filtrado Mexico

```
# Agrupar los datos por la columna 'anios' y sumar el número de medallas por año
medallas_por_anio = df.groupby('anios')['medallas'].sum().reset_index()

# Mostrar la evolución de medallas a lo largo de los años
print("Medallas por año:")
print(medallas_por_anio)

# Crear gráfico de líneas para visualizar la tendencia de medallas
plt.figure(figsize=(10, 6))
plt.plot(medallas_por_anio['anios'], medallas_por_anio['medallas'], marker='o',
linestyle='-', color='green')
plt.xlabel("Años")
plt.ylabel("Número de Medallas")
plt.title("Evolución de medallas a lo largo de los años")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Conclusión

La práctica ha demostrado la gran capacidad de Python para analizar y visualizar datos utilizando las librerías Pandas y Matplotlib. Al trabajar con el archivo olympics.csv, se evidenció cómo Pandas facilita la carga, limpieza, transformación y exploración de conjuntos de datos, permitiendo filtrar información específica, agrupar datos y calcular estadísticas de manera intuitiva y eficiente. Por otro lado, Matplotlib complementa este proceso al ofrecer herramientas robustas para generar visualizaciones claras y efectivas, lo que facilita la interpretación de tendencias y relaciones entre variables. La integración de ambas librerías resulta esencial para transformar datos brutos en insights significativos que apoyen la toma de decisiones en contextos académicos y profesionales. En definitiva, esta práctica reafirma la importancia de dominar estas herramientas, destacando su aplicabilidad y valor en un mundo cada vez más impulsado por la información.