Prácticas y Ejercicios de Obtención de Datos

Guía de Estudio: Sesión 3 - Ciencia de Datos

Generado para el curso de Ciencia de Datos
Fecha: 17 de Junio de 2025

Contents

1 Ejercicios Integrados

1.1 Ejercicio 1: Consolidar datos de CSV y Excel

Descripción: Lee un archivo CSV y un archivo Excel, combina ambos DataFrames y guárdalos en un nuevo archivo Excel.

```
import pandas as pd
df_csv = pd.read_csv("ventas.csv")
df_excel = pd.read_excel("clientes.xlsx")
df_combined = pd.concat([df_csv, df_excel], ignore_index=True)
df_combined.to_excel("consolidado.xlsx", index=False)
```

Solución: `pd.concat()` combina los DataFrames, y `to_excel()` guarda el resultado.

Aplicación: Integrar datos de múltiples fuentes para un análisis unificado.

1.2 Ejercicio 2: Extraer tabla web y combinar con CSV

Descripción: Extrae una tabla de población de Wikipedia y combínala con datos de un archivo CSV. Guarda el resultado en un archivo CSV.

```
import pandas as pd
url = "https://es.wikipedia.org/wiki/Anexo:Pa%C3%
   ADses_y_territorios_dependientes_por_poblaci%C3%B3n"
tablas = pd.read_html(url)
df_web = tablas[0]
df_csv = pd.read_csv("pib.csv")
df_merged = pd.merge(df_web, df_csv, on="Pa s")
df_merged.to_csv("poblacion_pib.csv", index=False)
```

Solución: `pd.merge()` combina los DataFrames usando una columna común, y `to_csv()` guarda el resultado.

Aplicación: Análisis económico combinando datos demográficos y financieros.

1.3 Ejercicio 3: Limpieza de datos de múltiples fuentes

Descripción: Lee un archivo CSV y un archivo Excel, elimina valores nulos y guárdalos en un nuevo archivo Excel.

```
import pandas as pd
df_csv = pd.read_csv("ventas.csv", na_values=["N/A"])
df_excel = pd.read_excel("clientes.xlsx")
df_combined = pd.concat([df_csv, df_excel]).dropna()
df_combined.to_excel("datos_limpios.xlsx", index=False)
```

Solución: Se combinan los datos con `pd.concat()` y se limpian con `dropna()`.

Aplicación: Preparar datasets limpios para modelado de machine learning.

1.4 Ejercicio 4: Extraer y exportar datos web a múltiples formatos

Descripción: Extrae una tabla de Wikipedia y guárdala tanto en un archivo CSV como en un archivo Excel.

```
import pandas as pd
url = "https://es.wikipedia.org/wiki/Anexo:Pa%C3%
   ADses_y_territorios_dependientes_por_poblaci%C3%B3n"
tablas = pd.read_html(url)
df = tablas[0]
df.to_csv("poblacion.csv", index=False)
df.to_excel("poblacion.xlsx", index=False)
```

Solución: La tabla se extrae con `read_html()` y se guarda en ambos formatos.

Aplicación: Distribuir datos en formatos compatibles con diferentes herramientas.

1.5 Ejercicio 5: Filtrar y guardar datos de múltiples fuentes

Descripción: Lee un archivo CSV y un archivo Excel, filtra los datos para valores mayores a 100 en una columna numérica y guárdalos en un archivo CSV.

```
import pandas as pd
df_csv = pd.read_csv("ventas.csv")
df_excel = pd.read_excel("clientes.xlsx")
df_combined = pd.concat([df_csv, df_excel])
df_filtered = df_combined[df_combined["Monto"] > 100]
df_filtered.to_csv("ventas_filtradas.csv", index=False)
```

Solución: Se filtran los datos usando una condición y se guardan con `to_csv()`.

Aplicación: Generar reportes específicos basados en criterios numéricos.

1.6 Ejercicio 6: Análisis combinado con datos web

Descripción: Extrae una tabla de Wikipedia, combínala con un archivo Excel, calcula el promedio de una columna numérica y guárdalo en un archivo CSV.

```
import pandas as pd
url = "https://es.wikipedia.org/wiki/Anexo:Pa%C3%
   ADses_y_territorios_dependientes_por_poblaci%C3%B3n"
tablas = pd.read_html(url)
df_web = tablas[0]
df_excel = pd.read_excel("indicadores.xlsx")
df_combined = pd.merge(df_web, df_excel, on="Pa s")
promedio = df_combined["Poblaci n"].mean()
print(f"Promedio de poblaci n: {promedio}")
df_combined.to_csv("analisis.csv", index=False)
```

Solución: Se combinan los datos, se calcula el promedio con `mean()`, y se guarda el resultado.

Aplicación: Análisis estadístico de datos combinados de fuentes públicas y privadas.