231501506

EX:2	EDA-Data Import and Export	AD23632

Aim:

- Importing data from CSV, Excel, SQL databases, and web scraping
- Handling different data formats
- Export a DataFrame to an Excel file

CODE:

```
import pandas as pd
import sqlite3
from google.colab import files

uploaded = files.upload()
df_csv = pd.read_csv('final_dataset.csv')
print("✅ CSV Data:")
print(df_csv.head())
```

```
Choose Files final_dataset.csv
• final_dataset.csv(text/csv) - 78368 bytes, last modified: 8/6/2025 - 100% done
Saving final_dataset.csv to final_dataset (1).csv
✅ CSV Data:
   Date  Month  Year  Holidays_Count  Days   PM2.5    PM10     NO2    SO2  \
0     1      1  2021               0     5  408.80  442.42  160.61  12.95
1     2      1  2021               0     6  404.04  561.95   52.85   5.18
2     3      1  2021               1     7  225.07  239.04  170.95  10.93
3     4      1  2021               0     1   89.55  132.08  153.98  10.42
4     5      1  2021               0     2   54.06   55.54  122.66   9.70

     CO  Ozone  AQI
0  2.77  43.19  462
1  2.60  16.43  482
2  1.40  44.29  263
3  1.01  49.19  207
4  0.64  48.88  149
```

```
uploaded = files.upload()
df_excel = pd.read_excel('converted_file.xlsx')
print("✅ Excel Data:")
print(df_excel.head())
```

231501506

```
Choose Files  converted_file.xlsx
• converted_file.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 1190399 bytes, last modified: 8/6/2025 - 100% done
Saving converted_file.xlsx to converted_file (2).xlsx
✅ Excel Data:
  letter  xbox  ybox  width  height  onpix  xbar  ybar  x2bar  y2bar  \
0      T     2     8      3       5      1     8    13      0      6
1      I     5    12      3       7      2    10     5      5      4
2      D     4    11      6       8      6    10     6      2      6
3      N     7    11      6       6      3     5     9      4      6
4      G     2     1      3       1      1     8     6      6      6

   xybar  x2ybar  xy2bar  xedge  xedgey  yedge  yedgex
0      6      10       8      0       8      0       8
1     13       3       9      2       8      4      10
2     10       3       7      3       7      3       9
3      4       4      10      6      10      2       8
4      6       5       9      1       7      5      10
```

```
conn = sqlite3.connect(':memory:')
cursor = conn.cursor()

cursor.execute("CREATE TABLE products (id INTEGER, name TEXT, price INTEGER)")
cursor.executemany("INSERT INTO products VALUES (?, ?, ?)", [
    (1, 'Laptop', 1000),
    (2, 'Tablet', 600),
    (3, 'Phone', 400)
])

conn.commit()

df_sql = pd.read_sql_query("SELECT * FROM products", conn)
print("✅ SQL Data:")
print(df_sql)
```

```
✅ SQL Data:
   id    name  price
0   1  Laptop   1000
1   2  Tablet    600
2   3   Phone    400
```

```
df_web = pd.read_html("https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)")[2]
print("✅ Web-Scraped Table:")
print(df_web.head())
```

231501506

```
✅ Web-Scraped Table:
  Country/Territory IMF[1][12]            World Bank[13]             \
  Country/Territory   Forecast       Year       Estimate       Year
0             World  113795678       2025      111326370       2024
1     United States   30507217       2025       29184890       2024
2             China   19231705  [n 1]2025       18743803  [n 3]2024
3           Germany    4744804       2025        4659929       2024
4             India    4187017       2025        3912686       2024

  United Nations[14]
            Estimate       Year
0          100834796       2022
1           27720700       2023
2           17794782  [n 1]2023
3            4525704       2023
4            3575778       2023
```

```
json_data = '{"name": ["Alice", "Bob"], "age": [25, 30]}'
df_json = pd.read_json(json_data)
print("✅ JSON Data:")
print(df_json)
```

```
✅ JSON Data:
    name  age
0  Alice   25
1    Bob   30
/tmp/ipython-input-2620177547.py:2: FutureWarning: Passing literal json to 'read_json'
  df_json = pd.read_json(json_data)
```

```
data_dict = {
    'Fruit': ['Apple', 'Banana', 'Orange'],
    'Quantity': [10, 15, 20]
}
df_dict = pd.DataFrame(data_dict)
print("✅ Dictionary Data:")
print(df_dict)
```

```
✅ Dictionary Data:
    Fruit  Quantity
0   Apple        10
1  Banana        15
2  Orange        20
```

```
df_dict.to_excel("exported_file.xlsx", index=False)
print("✅ DataFrame exported to Excel: exported_file.xlsx")
```

```
✅ DataFrame exported to Excel: exported_file.xlsx
```

231501506

Result:

Thus the EDA-Data Import and Export is done successfully.