

EXP NO: 2	EDA – DATA IMPORT AND EXPORT
----------------------------	-------------------------------------

AIM

To import data from CSV, Excel, and SQL databases and export DataFrames.

Problem Statement

LOAD datasets in multiple formats and export a DataFrame to Excel.

ALGORITHM

Step 1: Import Required Libraries

- Import pandas for data manipulation.
- Import sqlite3 for database handling.
- Import requests and BeautifulSoup for web scraping.

Step 2: Import Data from CSV File

- Use `pd.read_csv(filename)` to load data from a CSV file into a DataFrame.
- Display the first few rows using `.head()`.

Step 3: Import Data from Excel File

- Use `pd.read_excel(filename)` to load data from an Excel file.
- Display the first few rows using `.head()`.

Step 4: Import Data from SQL Database

- Connect to or create an SQLite database using `sqlite3.connect()`.
- Create a table (if not already exists).
- Insert sample records (if needed).
- Use `pd.read_sql_query(query, connection)` to load table data into a DataFrame.

Step 5: Import Data from the Web (Web Scraping)

- Use `requests.get(url)` to fetch HTML content.
- Parse HTML with BeautifulSoup.
- Locate the desired table using `soup.find()` or `soup.find_all()`.
- Convert the HTML table to a DataFrame using `pd.read_html()`.

Step 6: Handle Different Data Formats

- Check for data type issues or format mismatches.
- Convert date columns using `pd.to_datetime()`.
- Convert categorical or boolean fields using `.astype()`.

Step 7: Export Data to Excel File

- Use `DataFrame.to_excel(filename, index=False)` to save a DataFrame to an Excel file.
- Confirm export success with a print statement.

SAMPLE CODE

Import necessary libraries

import pandas as pd

import sqlite3

import requests

from bs4 import BeautifulSoup

1. Importing data from CSV

csv_df = pd.read_csv('Iris.csv')

print("CSV Data:")

print(csv_df.head())

```
CSV Data:
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
0   1             5.1           3.5           1.4           0.2  Iris-setosa
1   2             4.9           3.0           1.4           0.2  Iris-setosa
2   3             4.7           3.2           1.3           0.2  Iris-setosa
3   4             4.6           3.1           1.5           0.2  Iris-setosa
4   5             5.0           3.6           1.4           0.2  Iris-setosa
```

2. Importing data from Excel

excel_df = pd.read_excel('heart stalog dataset.xlsx')

print("\nExcel Data:")

excel_df.head(5)

Excel Data:

	age	sex	chest	resting_blood_pressure	serum_cholesterol	fasting_blood_sugar	resting_electroc
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	

```
#import from SQL Database
```

```
import sqlite3
```

```
# Connect to (or create) the database
```

```
conn = sqlite3.connect('my_database.db')
```

```
cursor = conn.cursor()
```

```
# Create the 'employees' table
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS employees (
```

```
    id INTEGER PRIMARY KEY,
```

```
    name TEXT,
```

```
    department TEXT,
```

```
    salary REAL,
```

```
    hire_date TEXT
```

```
)
```

```
""")
```

```
# Insert example records
```

```
cursor.executemany("""
```

```
INSERT INTO employees (id, name, department, salary, hire_date) VALUES (?, ?, ?, ?, ?)
```

```
""", [
```

```
    (1, 'Alice Smith', 'HR', 55000, '2018-05-01'),
```

```

(2, 'Bob Johnson', 'IT', 72000, '2019-07-15'),
(3, 'Carol White', 'Finance', 68000, '2017-09-30'),
(4, 'David Brown', 'Marketing', 60000, '2020-02-10'),
(5, 'Eva Green', 'IT', 75000, '2021-04-25'),
])

# Commit and close
conn.commit()

print("Database and 'employees' table created with sample data.")

Database and 'employees' table created with sample data

sql_df = pd.read_sql_query("SELECT * FROM employees", conn)
print(sql_df)

```

	id	name	department	salary	hire_date
0	1	Alice Smith	HR	55000.0	2018-05-01
1	2	Bob Johnson	IT	72000.0	2019-07-15
2	3	Carol White	Finance	68000.0	2017-09-30
3	4	David Brown	Marketing	60000.0	2020-02-10
4	5	Eva Green	IT	75000.0	2021-04-25

```

import pandas as pd
import requests
from bs4 import BeautifulSoup

# URL of the Wikipedia page
url = "https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population"

# Fetch the page
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")

```

```
# Find the first table with class 'wikitable' (Wikipedia uses this)
```

```
html_table = soup.find("table", {"class": "wikitable"})
```

```
# Use pandas to read the HTML table into a DataFrame
```

```
web_df = pd.read_html(str(html_table))[0]
```

```
# Show the first few rows
```

```
print("\nWeb Scraped Data:")
```

```
print(web_df.head())
```

```
Web Scraped Data:
   Location  Population % of world  Date \
0      World  8232000000      100%  13 Jun 2025
1      India  1413324000      17.3%   1 Mar 2025
2      China  1408280000      17.2%  31 Dec 2024
3  United States  340110988       4.2%   1 Jul 2024
4    Indonesia  282477584       3.5%  30 Jun 2024

   Source (official or from the United Nations)  Notes
0                                UN projection[1][3]  NaN
1                        Official projection[4]      [b]
2                        Official estimate[5]      [c]
3                        Official estimate[6]      [d]
4      National annual projection[7]      NaN
```

```
# 5. Handling different data formats
```

```
# For example, converting a date column to datetime
```

```
if 'date' in csv_df.columns:
```

```
    csv_df['date'] = pd.to_datetime(csv_df['date'])
```

```
datetime64[ns]
```

```
# 6. Export a DataFrame to Excel
```

```
# Here we export the CSV data as an example
```

```
csv_df.to_excel('exported_data.xlsx', index=False)
```

```
print("\nData exported to 'exported_data.xlsx' successfully.")
```

```
Data exported to 'exported_data.xlsx' successfully.
```

RESULT:

Thus, the program successfully created a Jupyter Notebook showcasing Python code to import data from CSV, Excel, and SQL databases, as well as export DataFrames.