	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**

Aim: Practical based on Data Loading, Storage and File Formats

IDE:

load, manipulate, and store data using Python (over reading and writing CSV, JSON, and Excel files)

Library Installation

pip install pandas openpyxl

Sample Data:

Create a folder for this experiment and add the following sample data files:

sample_data.csv (Name,Age,City

Alice,30,New York

Bob,25,Los Angeles

Charlie,35,Chicago)

sample_data.json ([

{"Name": "David", "Age": 28, "City": "San Francisco"},

{"Name": "Eve", "Age": 22, "City": "Seattle"}

])

sample_data.xlsx (you can create this using Excel with similar data**)**

Loading Data from CSV

Read the CSV file and perform basic data manipulation.

```
import pandas as pd


# Load data from CSV
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**

```
csv_file_path = 'sample_data.csv'

df_csv = pd.read_csv(csv_file_path)


# Display the DataFrame

print("CSV Data:")

print(df_csv)


# Basic data manipulation: Filter by age

filtered_data = df_csv[df_csv['Age'] > 30]

print("\nFiltered Data (Age > 30):")

print(filtered_data)
```

Output:

```
CSV Data:
      Name  Age         City
0    Alice   30     New York
1      Bob   25  Los Angeles
2  Charlie   35      Chicago

Filtered Data (Age > 30):
      Name  Age     City
2  Charlie   35  Chicago
```

Loading Data from JSON

Read the JSON file and manipulate the data.

```
# Load data from JSON
```

```
json_file_path = 'sample_data.json'

df_json = pd.read_json(json_file_path)


# Display the DataFrame

print("\nJSON Data:")

print(df_json)


# Basic data manipulation: Find the average age

average_age = df_json['Age'].mean()

print("\nAverage Age:", average_age)
```

Output:

```
JSON Data:
    Name  Age           City
0  David   28  San Francisco
1    Eve   22        Seattle

Average Age: 25.0
```

Loading Data from Excel

Read the Excel file and display its contents.

```
# Load data from Excel

excel_file_path = 'sample_data.xlsx'

df_excel = pd.read_excel(excel_file_path)
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**

```
# Display the DataFrame

print("\nExcel Data:")

print(df_excel)


# Basic data manipulation: Count the number of entries

entry_count = df_excel.shape[0]

print("\nNumber of entries in Excel file:", entry_count)
```

Output:

```
Excel Data:
      Name  Age           City
0    Alice   30       New York
1      Bob   25    Los Angeles
2  Charlie   35        Chicago
3    David   28  San Francisco
4      Eve   22        Seattle
```

Writing Data to Different Formats

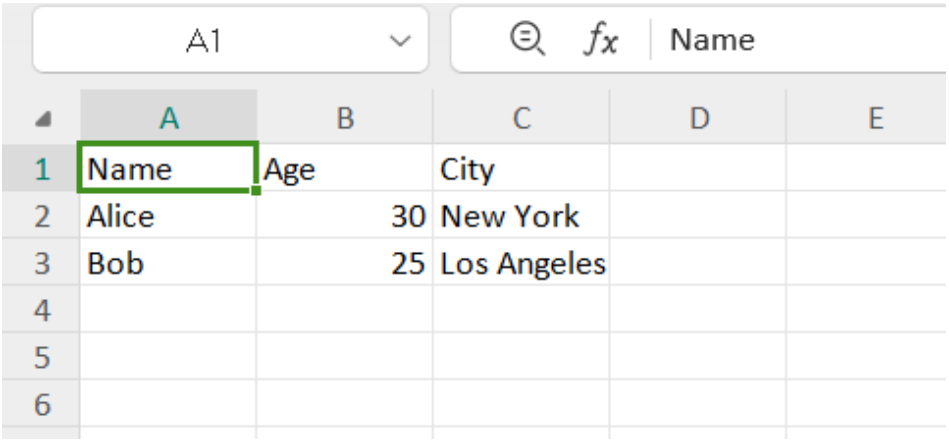
Save manipulated DataFrames to new files in different formats.

```
# Save filtered CSV data to a new file

filtered_data.to_csv('filtered_data.csv', index=False)

print("\nFiltered data saved to 'filtered_data.csv'.")
```

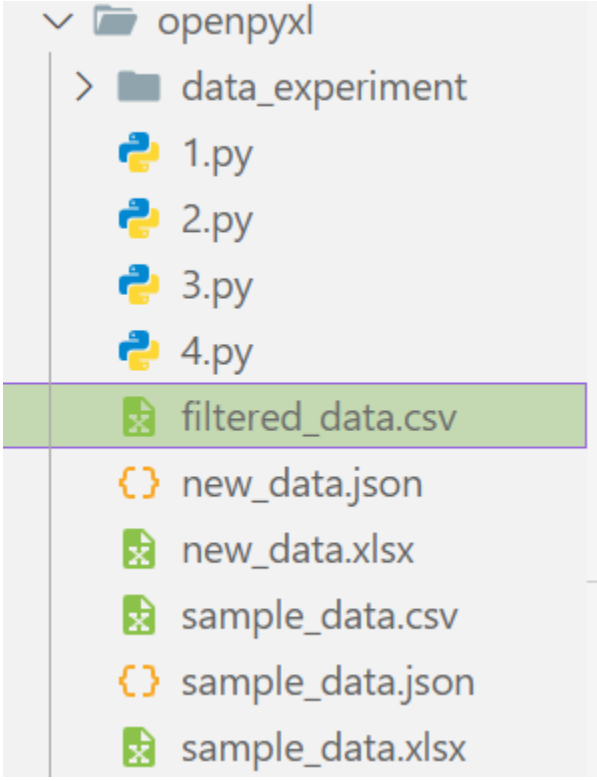
Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**

```
# Save DataFrame to a new JSON file

df_json.to_json('new_data.json', orient='records', lines=True)

print("JSON data saved to 'new_data.json'.")


# Save DataFrame to a new Excel file

df_excel.to_excel('new_data.xlsx', index=False)

print("Excel data saved to 'new_data.xlsx'.")
```

Output:



Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**



Post Lab:

https://github.com/keshvi1234/PWP_experiment

Write a code snippet to check the data types of each column in a DataFrame.

Code:

```
import pandas as pd

df_csv = pd.read_csv(r"D:\SEM 3 Subjects\Python\openpyxl\sample_data.csv")
df_json = pd.read_json(r"D:\SEM 3 Subjects\Python\openpyxl\sample_data.json")
df_excel = pd.read_excel(r"D:\SEM 3 Subjects\Python\openpyxl\sample_data.xlsx")
def check_data_types(df, name):
    print(f"\nData Types for {name}:")
    print(df.dtypes)

check_data_types(df_csv, "CSV DataFrame")
check_data_types(df_json, "JSON DataFrame")
check_data_types(df_excel, "Excel DataFrame")
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**

Output:

```
Data Types for CSV DataFrame:
Name    object
Age      int64
City    object
dtype: object

Data Types for JSON DataFrame:
Name    object
Age      int64
City    object
dtype: object

Data Types for Excel DataFrame:
Name    object
Age      int64
City    object
dtype: object
PS D:\SEM 3 Subjects\Python>
```

Write a code snippet that demonstrates how to fill missing values with the mean of a column.

Code:

```python
import pandas as pd
import numpy as np

df_csv = pd.read_csv(r"D:\SEM 3 Subjects\Python\openpyxl\sample_data.csv")
df_json = pd.read_json(r"D:\SEM 3 Subjects\Python\openpyxl\sample_data.json")
df_excel = pd.read_excel(r"D:\SEM 3 Subjects\Python\openpyxl\sample_data.xlsx")

df_csv.loc[1, 'Age'] = np.nan  # CSV DataFrame
df_json.loc[1, 'Age'] = np.nan
df_excel.loc[1, 'Age'] = np.nan
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Data Loading, Storage and File Formats	
Experiment No: 22	**Date:**	**Enrollment No: 92510133028**

```python
def fill_missing_with_mean(df, name):
    df['Age'] = df['Age'].fillna(df['Age'].mean())
    print(f"\n{name} after filling missing 'Age' values with mean:")
    print(df)

fill_missing_with_mean(df_csv, "CSV DataFrame")
fill_missing_with_mean(df_json, "JSON DataFrame")
fill_missing_with_mean(df_excel, "Excel DataFrame")
```

Output:

```
CSV DataFrame after filling missing 'Age' values with mean:
      Name   Age         City
0    Alice  30.0     New York
1      Bob  32.5  Los Angeles
2  Charlie  35.0      Chicago

JSON DataFrame after filling missing 'Age' values with mean:
    Name   Age           City
0  David  28.0  San Francisco
1    Eve  28.0        Seattle

Excel DataFrame after filling missing 'Age' values with mean:
      Name    Age           City
0    Alice  30.00       New York
1      Bob  28.75    Los Angeles
2  Charlie  35.00        Chicago
3    David  28.00  San Francisco
4      Eve  22.00        Seattle
```