Fetching Data from Multiple Sources using Pandas

1. Introduction

In real-world data science, data comes from multiple sources: CSV, Excel, SQL, JSON, Web APIs, HTML tables, or multiple files. Pandas provides functions to read, merge, and process this data efficiently.

2. Reading Data from CSV Files

```python
import pandas as pd
import glob

# Single CSV
df = pd.read_csv('data.csv')

# Multiple CSVs in a folder
files = glob.glob('data_folder/*.csv')
df_list = [pd.read_csv(file) for file in files]
df_all = pd.concat(df_list, ignore_index=True)
```

3. Reading from Excel Files

```python
# Single sheet
df = pd.read_excel('data.xlsx', sheet_name='Sheet1')

# Multiple sheets
data = pd.read_excel('data.xlsx', sheet_name=None)  # dict of DataFrames
for name, sheet in data.items():
    print(name)
    print(sheet.head())

# Combine multiple Excel files
files = glob.glob('excels/*.xlsx')
df_all = pd.concat([pd.read_excel(f) for f in files])
```

4. Reading from SQL Databases

```
import sqlite3
conn = sqlite3.connect('students.db')
df = pd.read_sql_query('SELECT * FROM students', conn)

# Merge multiple tables
df1 = pd.read_sql_query('SELECT * FROM students', conn)
df2 = pd.read_sql_query('SELECT * FROM marks', conn)
merged = pd.merge(df1, df2, on='student_id')
```

With SQLAlchemy (MySQL/PostgreSQL)

from sqlalchemy import create_engine engine = create_engine('mysql+pymysql://user:password@localhost:3306/database') df = pd.read_sql('SELECT * FROM employees', engine)

~~~
---

# 5. Reading from JSON Files or APIs
```python
# JSON file
df = pd.read_json('data.json')

# From Web API
import requests
url = 'https://jsonplaceholder.typicode.com/users'
response = requests.get(url)
data = response.json()
df = pd.DataFrame(data)

# Flatten nested JSON
from pandas import json_normalize
df = json_normalize(data)
~~~

---

6. Reading Tables from HTML Pages

```
url = 'https://www.w3schools.com/html/html_tables.asp'
tables = pd.read_html(url)
df = tables[0]
```

7. Reading from Text Files

```
df = pd.read_csv('data.txt', delimiter='\t')
```

8. Combining Data from Multiple Sources

```
csv_df = pd.read_csv('sales_2023.csv')
excel_df = pd.read_excel('sales_2024.xlsx')
sql_df = pd.read_sql('SELECT * FROM sales_data', conn)

combined = pd.concat([csv_df, excel_df, sql_df], ignore_index=True)
```

9. Reading from Online URLs

```
csv_url = 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv'
df = pd.read_csv(csv_url)
```

10. Merging Data from Different Sources

```
df1 = pd.read_csv('customers.csv')
df2 = pd.read_excel('orders.xlsx')
df3 = pd.read_sql('SELECT * FROM payments', conn)

merged = pd.merge(df1, df2, on='customer_id')
final_df = pd.merge(merged, df3, on='customer_id')
```

11. Exporting Combined Data

```
final_df.to_csv('final_output.csv', index=False)
final_df.to_excel('final_output.xlsx', index=False)
final_df.to_sql('final_table', conn, if_exists='replace', index=False)
```

Summary of Pandas Data Sources

Source	Function	Example
CSV	read_csv	pd.read_csv('file.csv')
Excel	read_excel	pd.read_excel('data.xlsx')
SQL	read_sql	pd.read_sql('SELECT *', conn)
JSON	read_json	pd.read_json('data.json')
HTML	read_html	pd.read_html(url)[0]
API	requests.get + DataFrame	pd.DataFrame(requests.get(url).json())
Multiple files	glob + concat	pd.concat([pd.read_csv(f) for f in files])

End of Guide: Fetching Data from Multiple Sources