DATA SCIENCE COURSE TUTORIAL # 08

2.4 File Formats (CSV, Excel, JSON, SQL, XML, Parquet, etc.)

Different types of data are stored in different file formats. Some of the most commonly used formats in Data Science are:

CSV (Comma-Separated Values)

A plain text file where values are separated by commas. It is widely used due to its simplicity and compatibility with most tools.

Example:

```
Name, Age, City
Alice, 30, New York
Bob, 25, Los Angeles
Charlie, 35, Chicago
```

Excel (XLS, XLSX)

Spreadsheet format used by Microsoft Excel. It can hold multiple sheets and allows formulas, formatting, and charts.

Example:

```
| Name    | Age | City        |
|---------|-----|-------------|
| Alice   | 30  | New York    |
| Bob     | 25  | Los Angeles |
| Charlie | 35  | Chicago     |
```

JSON (JavaScript Object Notation)

A lightweight format for storing and transporting data, especially used in APIs. It is easy to read and write for both humans and machines.

Example:

```
[
  {"Name": "John", "Age": 25, "Country": "USA"},
  {"Name": "Alice", "Age": 30, "Country": "UK"}
]
```

SQL Databases

Data stored in structured tables using SQL language. It is ideal for large-scale structured data and allows powerful querying.

Example:

```sql
SELECT Name, Age, City
FROM Users
WHERE Age > 25
ORDER BY Age DESC;
```

XML (eXtensible Markup Language)

Used for storing and transporting data. It uses custom tags and is similar to HTML but more focused on data structure.

Example:

```xml
<Users>
  <User>
    <Name>John</Name>
    <Age>25</Age>
    <City>New York</City>
  </User>
  <User>
    <Name>Alice</Name>
    <Age>30</Age>
    <City>London</City>
  </User>
</Users>
```

Parquet

A columnar storage format used in big data tools like Apache Spark. It is optimized for fast reading and writing and is very space-efficient. Parquet is binary, so it is not human-readable.

Example: Cannot be viewed directly as text, but when read in Python with Pandas:

```python
import pandas as pd

df = pd.read_parquet('data.parquet')
print(df)
```

Output:

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

Each format has its own strengths and is used depending on the size, type, and purpose of the data.