

1. Pandas for Data Processing:

Pandas is a Python library that provides high-performance, easy-to-use data structures, such as DataFrame, and data analysis tools. It is built on top of NumPy, another popular Python library for numerical computing. The key data structure in Pandas is the DataFrame, a two-dimensional table that can store heterogeneous data types and handle missing values. Here are some key concepts in Pandas:

DataFrame: A two-dimensional, tabular data structure with labeled axes (rows and columns). It is similar to a spreadsheet or SQL table.

Series: A one-dimensional labeled array capable of holding any data type.

Index: An immutable array that labels the rows or columns of a DataFrame.

Operations: Pandas provides a wide range of functions and methods for data manipulation, cleaning, and analysis. These include merging, reshaping, slicing, indexing, and statistical operations.

Example:

```
import pandas as pd
```

```
# Creating a DataFrame from a dictionary
```

```
data = {'Name': ['Alice', 'Bob', 'Charlie'],  
        'Age': [25, 30, 35],
```

```
'City': ['New York', 'San Francisco', 'Los Angeles']}]}
```

```
df = pd.DataFrame(data)
```

```
print("DataFrame:")
```

```
print(df)
```

2. Reading CSV Data using Pandas:

The `pd.read_csv()` function in Pandas is used to read data from a CSV (Comma Separated Values) file into a DataFrame. The function provides various parameters to customize the import process, such as specifying delimiters, handling missing values, skipping rows, and more.

```
# Reading CSV data into a DataFrame
```

```
file_path = 'data.csv'
```

```
df = pd.read_csv(file_path)
```

```
print("DataFrame after reading CSV:")
```

```
print(df)
```

This code reads the data from the CSV file specified in `file_path` and stores it in a Pandas DataFrame named `df`. You can adjust the parameters of `read_csv()` based on the specifics of your CSV file.

3. Read Data from CSV Files to Pandas Dataframes:

The process of reading data from CSV files involves creating a Pandas DataFrame to hold the data. Once the data is loaded, you can use various DataFrame methods and attributes to explore and manipulate the data.

4. Filter Data in Pandas DataFrame using query:

The query method in Pandas allows you to filter rows of a DataFrame using a query expression similar to SQL. This method provides a concise and readable way to filter data based on specified conditions.

```
# Filtering data where Age is greater than 30
```

```
filtered_df = df.query('Age > 30')
```

```
print("\nDataFrame after filtering:")
```

```
print(filtered_df)
```

This code creates a new DataFrame, `filtered_df`, containing only the rows where the 'Age' column is greater than 30.

```
In [1]: import pandas as pd

# Creating a DataFrame from a dictionary
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'San Francisco', 'Los Angeles']}
```

```
df = pd.DataFrame(data)
print("DataFrame:")
print(df)

DataFrame:
   Name  Age      City
0  Alice   25  New York
1   Bob   30  San Francisco
2  Charlie  35  Los Angeles
```

C:\Users\Abhishek\AppData\Local\Temp\ipykernel_48112\2978960849.py:1: DeprecationWarning: Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0), (to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries) but was not found to be installed on your system. If this would cause problems for you, please provide us feedback at <https://github.com/pandas-dev/pandas/issues/54466>

```
import pandas as pd
```

```
In [2]: df.to_csv('data.csv')
```

```
In [3]: # Reading CSV data into a DataFrame
file_path = 'data.csv'
df = pd.read_csv(file_path)

print("DataFrame after reading CSV:")
print(df)
```

```
DataFrame after reading CSV:
Unnamed: 0      Name  Age      City
0          0  Alice   25  New York
1          1   Bob   30  San Francisco
2          2  Charlie  35  Los Angeles
```

```
In [4]: # Filtering data where Age is greater than 30
filtered_df = df.query('Age > 30')

print("\nDataFrame after filtering:")
print(filtered_df)
```

```
DataFrame after filtering:
Unnamed: 0      Name  Age      City
2          2  Charlie  35  Los Angeles
```

```
In [5]: # Adding a new column
df['Salary'] = [50000, 60000, 75000]

# Calculating average age
average_age = df['Age'].mean()

print("\nDataFrame after operations:")
print(df)
print(f"\nAverage Age: {average_age}")
```

```
DataFrame after operations:
Unnamed: 0      Name  Age      City  Salary
0          0  Alice   25  New York   50000
1          1   Bob   30  San Francisco 60000
2          2  Charlie  35  Los Angeles 75000

Average Age: 30.0
```

```
In [6]: import pandas as pd

# Data provided
data1 = {'Product': ['Laptop', 'Headphones', 'Smartphone', 'Jacket', 'Jeans', 'Smartwatch'],
        'Category': ['Electronics', 'Electronics', 'Electronics', 'Apparel', 'Apparel', 'Electronics'],
        'Price': [1200, 100, 800, 150, 60, 200],
        'Quantity': [50, 100, 75, 30, 80, 40],
        'SalesDate': ['2023-01-15', '2023-01-20', '2023-01-25', '2023-02-02', '2023-02-05', '2023-02-10']}
```

```
In [7]: df1 = pd.DataFrame(data1)
print("DataFrame:")
print(df1)

DataFrame:
   Product  Category  Price  Quantity  SalesDate
0  Laptop  Electronics   1200        50  2023-01-15
1  Headphones  Electronics    100       100  2023-01-20
2  Smartphone  Electronics    800        75  2023-01-25
3   Jacket    Apparel    150        30  2023-02-02
4   Jeans    Apparel     60         80  2023-02-05
5  Smartwatch  Electronics    200        40  2023-02-10
```

```
In [8]: df1.to_csv('data2.csv')
```

```
In [9]: # Reading CSV data into a DataFrame
file_path = 'data2.csv'
df1 = pd.read_csv(file_path)

print("DataFrame after reading CSV:")
print(df1)
```

```
DataFrame after reading CSV:
Unnamed: 0      Product  Category  Price  Quantity  SalesDate
```

```
0      0      Laptop  Electronics 1200      50 2023-01-15
1      1  Headphones Electronics   100     100 2023-01-20
2      2  Smartphone Electronics   800      75 2023-01-25
3      3      Jacket      Apparel   150      30 2023-02-02
4      4      Jeans      Apparel    60      80 2023-02-05
5      5  Smartwatch Electronics   200      40 2023-02-10
```

In []:

In []:

In []: