Department of Engineering Sciences and Technology,

Second Year Btech in Computer Science

Project Based Learning-Python

Assignment - 16

Name - Paritosh kolwadkar

SRN – 31231313

Roll no – 39

Batch – D2

Problem statement : **Write a program to filter rows from a DataFrame based on a condition using boolean indexing or the query() method. Sort the filtered DataFrame by a specific column using sort_values().**

Pre-requisites: Install the Pandas library:

```
pip install pandas
```

Basic knowledge of Pandas DataFrames and filtering.

Code:

```
# Import Pandas

import pandas as pd



# Sample data

data = {

   "Name": ["Alice", "Bob", "Charlie", "David", "Eve"],

   "Age": [25, 30, 35, 40, 28],

   "Salary": [50000, 60000, 55000, 70000, 65000]

}
```

```python
# Create a DataFrame
df = pd.DataFrame(data)

# Filter rows where Age > 30 using Boolean indexing
filtered_df_boolean = df[df["Age"] > 30]

# Filter rows where Age > 30 using the query() method
filtered_df_query = df.query("Age > 30")

# Sort the filtered DataFrame by the 'Salary' column in descending order
sorted_df = filtered_df_boolean.sort_values(by="Salary", ascending=False)

# Display the original DataFrame
print("Original DataFrame:")
print(df)

# Display the filtered DataFrame (Boolean indexing)
print("\nFiltered DataFrame (Age > 30, Boolean indexing):")
print(filtered_df_boolean)

# Display the filtered DataFrame (query() method)
print("\nFiltered DataFrame (Age > 30, query() method):")
print(filtered_df_query)

# Display the sorted DataFrame
print("\nSorted DataFrame (Filtered rows sorted by Salary in descending order):")
print(sorted_df)
```

Explanation :

Create a DataFrame:

- **The `data` dictionary contains columns for `Name`, `Age`, and `Salary`.**
- **The dictionary is converted into a Pandas DataFrame using `pd.DataFrame()`.**

Filter Rows Using Boolean Indexing:

- **A condition (`df["Age"] > 30`) is applied to filter rows where the `Age` column's value is greater than 30.**
- **Boolean indexing selects rows that satisfy the condition.**

Filter Rows Using `query()` Method:

- **The `query()` method is used to filter rows with a condition provided as a string.**

Sort the Filtered DataFrame:

- **`sort_values(by="Salary", ascending=False)` sorts the filtered rows by the `Salary` column in descending order.**

Display Results:

- **The original DataFrame, filtered DataFrames (using both methods), and the sorted DataFrame are printed.**

Output:

Original DataFrame:

	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	55000
3	David	40	70000
4	Eve	28	65000

Filtered DataFrame (Age > 30, Boolean indexing):

	Name	Age	Salary
2	Charlie	35	55000

3 David 40 70000

Filtered DataFrame (Age > 30, query() method):

Name Age Salary

2 Charlie 35 55000

3 David 40 70000

Sorted DataFrame (Filtered rows sorted by Salary in descending order):

Name Age Salary

3 David 40 70000

2 Charlie 35 55000

Output Explained:

- **Boolean Indexing**:
  - Directly filters rows by applying conditions to the DataFrame.
- **`query()` Method**:
  - Provides an alternative to Boolean indexing with a string-based query syntax.
- **`sort_values()`**:
  - Sorts rows by a specified column, either in ascending or descending order.

This program demonstrates efficient row filtering and sorting techniques, crucial for data preprocessing and analysis tasks.