Assignment – 10

Pandas for Data Processing:

Pandas is a powerful library in Python for data manipulation and analysis. It provides data structures like Series and DataFrame that are efficient for working with structured data. Pandas is widely used for tasks such as data cleaning, exploration, and transformation.

Reading CSV Data using Pandas:

Pandas provides the read_csv() function to read data from CSV (Comma-Separated Values) files into a DataFrame.

Filter Data in Pandas Dataframe using query:

The query() method in Pandas allows you to filter a DataFrame based on a query expression.

```
In [10]: # Example: Filter Data
         filtered_df = df.query('Age > 25')
```

Get Count by Status using Pandas Dataframe APIs:

You can use the value_counts() method to get the count of unique values in a column, which is often useful for status counts.

```
In [11]: # Example: Get Count by Status
         status_counts = df['Status'].value_counts()
```

Get count by Month and Status using Pandas Dataframe APIs:

You can use the groupby() method along with value_counts() to get counts based on multiple columns, like month and status.

```
In [12]: # Example: Get count by Month and Status
         df['Month'] = df['Date'].dt.month  # Extract month from Date
         counts_by_month_status = df.groupby(['Month', 'Status']).size().unstack(fill_value=0)
```

Create Dataframes using dynamic column list on CSV Data:

You can create DataFrames dynamically by selecting columns based on a list of column names.

Performing Inner Join between Pandas Dataframes:

The merge() function is used for performing joins between two DataFrames. An inner join keeps only the common rows between the two DataFrames.

In [13]:
```
# Example: Performing Inner Join
df2 = pd.DataFrame({'ID': [1, 2, 3, 4, 6], 'Score': [85, 90, 78, 92, 88]})
merged_df = pd.merge(df, df2, on='ID', how='inner')
```

Perform Aggregations on Join results:

After joining DataFrames, you can perform aggregations using the groupby() function and applying aggregation functions like sum(), mean(), etc.

In [14]:
```
# Example: Perform Aggregations on Join results
aggregated_result = merged_df.groupby('Status')['Score'].mean()
```

Sort Data in Pandas Dataframes:

You can use the sort_values() method to sort a DataFrame based on one or more columns.

In [15]:
```
# Example: Sort Data
sorted_df = df.sort_values(by='Age', ascending=True)
```

Writing Pandas Dataframes to Files:

Use the to_csv() method to write a DataFrame to a CSV file.

In [16]:
```
# Example: Writing Pandas Dataframes to Files
df.to_csv('sample_data_processed.csv', index=False)
```