NAME :

Harshith Chowdary Kodali

Task :

Data Manipulation with Pandas : This task involves using the pandas library to manipulate the data Responsibility : Load a CSV File into pandas DataFrame Perform operations like filtering data based on conditions , handling missing values , and calculating summary statistics

CSV File :

https://www.kaggle.com/datasets/muhammaddawood42/nvidia-stock-data

Code :

Python

```python
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('/content/NVIDIA_STOCK.csv')

# Display the first few rows of the DataFrame
print("Loaded DataFrame:")
print(df.head())

# Ensure the 'Close' column is numeric
df['Close'] = pd.to_numeric(df['Close'], errors='coerce')

# Drop rows where 'Close' is NaN (caused by non-numeric values)
df = df.dropna(subset=['Close'])

# Now filter rows where Close price is greater than 500
filtered_df = df[df['Close'] > 500]

# Display the filtered DataFrame
print("\nFiltered DataFrame (Close > 500):")
print(filtered_df.head())

print("\nMissing values in each column:")
print(df.isnull().sum())
```

```python
# Drop rows with missing values
cleaned_df = df.dropna()

# Display the cleaned DataFrame
print("\nDataFrame after dropping missing values:")
print(cleaned_df.head())

# Identify numeric columns
numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns

# Fill missing values with the mean for numeric columns only
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# Display the DataFrame after filling missing values
print("\nDataFrame after filling missing values with mean (numeric columns
only):")
print(df.head())

# Display summary statistics for numeric columns
print("\nSummary Statistics:")
print(df.describe())

# Save the filtered DataFrame to a new CSV file
filtered_df.to_csv("filtered_data.csv", index=False)
print("\nFiltered data saved to 'filtered_data.csv'.")

# Save the cleaned DataFrame to a new CSV file
cleaned_df.to_csv("cleaned_data.csv", index=False)
print("\nCleaned data saved to 'cleaned_data.csv'.")
```

OUTPUT :

```
Loaded DataFrame:
        Price           Adj Close               Close               High  \
0      Ticker                NVDA                NVDA               NVDA
1        Date                 NaN                 NaN                NaN
2  2018-01-02   4.929879665374756   4.983749866485596  4.987500190734863
3  2018-01-03   5.254334926605225  5.3117499351501465   5.34250020980835
4  2018-01-04  5.2820329666137695   5.339749813079834  5.451250076293945

                 Low               Open     Volume
0               NVDA               NVDA       NVDA
1                NaN                NaN        NaN
2  4.862500190734863  4.894499778747559  355616000
3            5.09375  5.102499961853027  914704000
4  5.317249774932861  5.394000053405762  583268000

Filtered DataFrame (Close > 500):
Empty DataFrame
Columns: [Price, Adj Close, Close, High, Low, Open, Volume]
Index: []

Missing values in each column:
Price        0
Adj Close    0
Close        0
High         0
Low          0
Open         0
Volume       0
dtype: int64

DataFrame after dropping missing values:
        Price           Adj Close     Close               High  \
2  2018-01-02   4.929879665374756   4.98375  4.987500190734863
3  2018-01-03   5.254334926605225   5.31175   5.34250020980835
4  2018-01-04  5.2820329666137695   5.33975  5.451250076293945
5  2018-01-05   5.326793670654297   5.38500  5.422749996185303
6  2018-01-08   5.490012168884277   5.55000              5.625
```

```
                 Low               Open     Volume
2   4.862500190734863  4.894499778747559  355616000
3             5.09375  5.102499961853027  914704000
4   5.317249774932861  5.394000053405762  583268000
5  5.2769999504089355  5.354750156402588  580124000
6  5.4644999504089355  5.510000228881836  881216000

DataFrame after filling missing values with mean (numeric columns only):
        Price           Adj Close     Close               High  \
2  2018-01-02   4.929879665374756   4.98375  4.987500190734863
3  2018-01-03   5.254334926605225   5.31175   5.34250020980835
4  2018-01-04  5.2820329666137695   5.33975  5.451250076293945
5  2018-01-05   5.326793670654297   5.38500  5.422749996185303
6  2018-01-08   5.490012168884277   5.55000              5.625

                 Low               Open     Volume
2   4.862500190734863  4.894499778747559  355616000
3             5.09375  5.102499961853027  914704000
4   5.317249774932861  5.394000053405762  583268000
5  5.2769999504089355  5.354750156402588  580124000
6  5.4644999504089355  5.510000228881836  881216000

Summary Statistics:
             Close
count  1697.000000
mean     24.828411
std      29.216014
min       3.177000
25%       6.161750
50%      14.015750
75%      27.104000
max     135.580002

Filtered data saved to 'filtered_data.csv'.

Cleaned data saved to 'cleaned_data.csv'.
```

CSV Files After the manipulation :

Cleaned_data.csv
filtered_data.csv