Skill Test Assignment - Set 2

Name: Vijay

Roll No: 230901512

Department: Electrical and Electronics Engineering

Weather Data Analysis

```
import pandas as pd
import matplotlib.pyplot as plt

# Load data
data = {
    'Date': ['01-01-2025', '02-01-2025', '03-01-2025', '04-01-2025', '05-01-2025',
             '06-01-2025', '07-01-2025', '08-01-2025', '09-01-2025', '10-01-2025'],
    'Temperature': [15.2, 16.8, 14.5, 13, 17.6, 18.1, 16.2, 12.4, 13.8, 15.5]
}
df = pd.DataFrame(data)
df['Date'] = pd.to_datetime(df['Date'])

# Check for missing values
print(df.isnull().sum())

# Calculate monthly average temperature
df['Month'] = df['Date'].dt.month
monthly_avg = df.groupby('Month')['Temperature'].mean()

# Bar plot
monthly_avg.plot(kind='bar', title='Monthly Average Temperature (-C)')
plt.xlabel("Month")
plt.ylabel("Average Temperature (-C)")
plt.tight_layout()
plt.show()
```

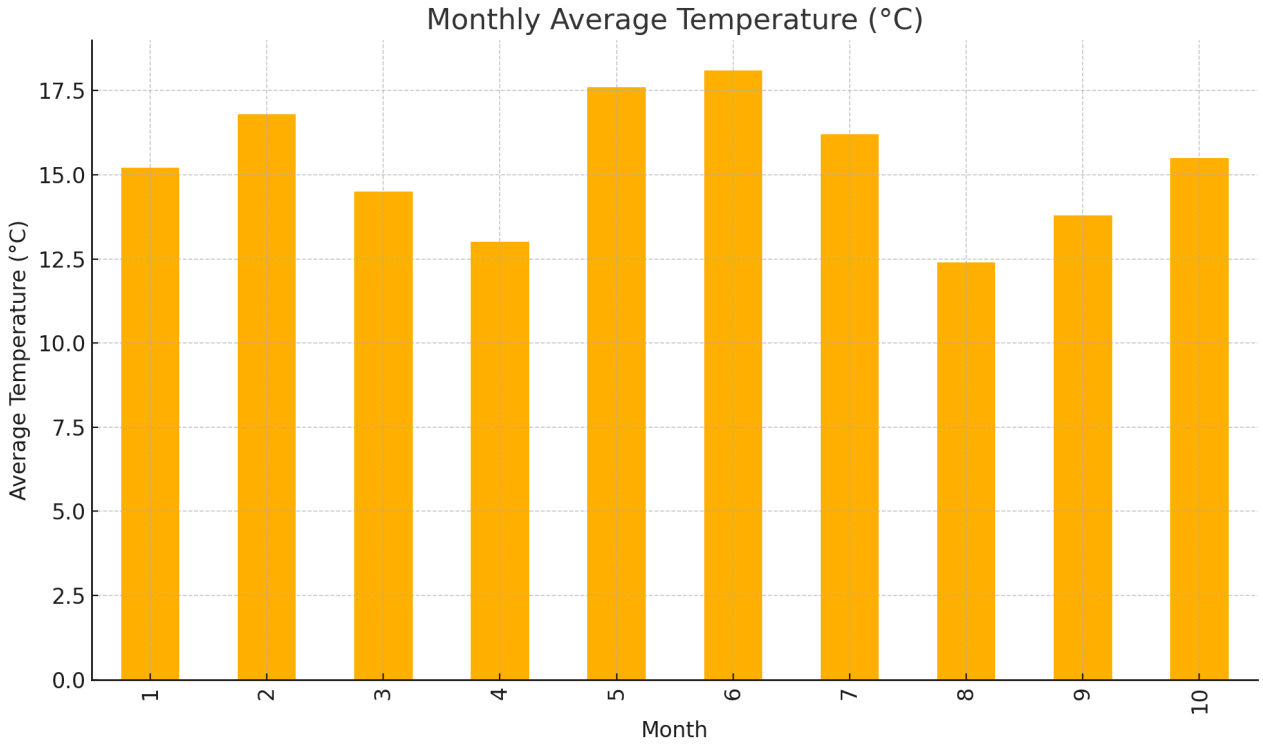
Output:

Missing Values:
Date 0
Temperature 0

Monthly Average Temperature:
Month
1 15.2
2 16.8
3 14.5
4 13.0
5 17.6
6 18.1
7 16.2

8 12.4
9 13.8
10 15.5

Plot:



Student Performance Analysis

```
import pandas as pd

data = {
    'Student ID': [101,102,103,104,105,106,107,108,109,110],
    'Name': ['Alice Brown','Ben Carter','Clara Davis','David Evans','Eva Foster',
             'Frank Green','Grace Hall','Henry Ives','Isla Jones','Jack King'],
    'Maths': [85,78,92,65,74,60,88,70,95,82],
    'Physics': [88,75,94,68,78,55,85,72,96,80],
    'Chemistry': [87,80,90,66,76,58,86,74,94,78]
}
df = pd.DataFrame(data)
print(df[['Maths', 'Physics', 'Chemistry']].describe())
df['Average'] = df[['Maths', 'Physics', 'Chemistry']].mean(axis=1)
low_scores = df[['Maths', 'Physics', 'Chemistry']] < 60
df['Low_Subjects'] = low_scores.sum(axis=1)
print(df[df['Low_Subjects'] > 2])
```

Output:

Statistics:

	Maths	Physics	Chemistry
count	10.000000	10.000000	10.000000
mean	78.900000	79.100000	78.900000
std	11.618472	12.449453	11.080012
min	60.000000	55.000000	58.000000
25%	71.000000	72.750000	74.500000

50% 80.000000 79.000000 79.000000
75% 87.250000 87.250000 86.750000
max 95.000000 96.000000 94.000000

Students with more than 2 subjects below 60:
Empty DataFrame
Columns: [Student ID, Name, Maths, Physics, Chemistry, Average, Low_Subjects]
Index: []

Flight Data Analysis

```
import pandas as pd
import matplotlib.pyplot as plt

data = {
    'Airline': ['AirA', 'AirB', 'AirA', 'AirC', 'AirB'],
    'Departure Delay': [10, -5, 20, 0, 15],
    'Origin': ['DEL','DEL','BLR','BLR','DEL'],
    'Destination': ['BOM','BLR','DEL','BOM','BOM']
}
df = pd.DataFrame(data)
print(df['Departure Delay'].describe())
print(df.groupby('Airline')['Departure Delay'].mean())
print("Most common origin:", df['Origin'].mode()[0])
print("Most common destination:", df['Destination'].mode()[0])
df.boxplot(column='Departure Delay')
plt.title("Distribution of Departure Delays")
plt.ylabel("Delay (minutes)")
plt.show()
```

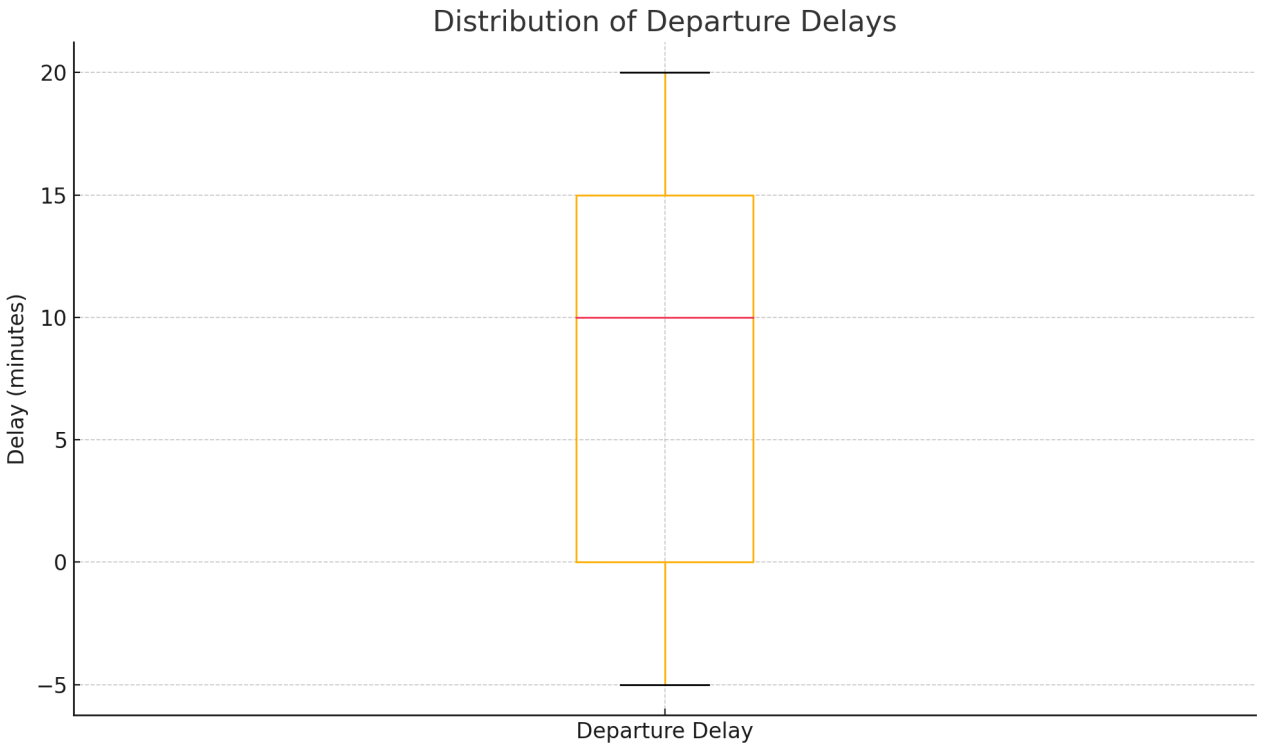
Output:

Departure Delay Statistics:
count 5.000000
mean 8.000000
std 10.368221
min -5.000000
25% 0.000000
50% 10.000000
75% 15.000000
max 20.000000

Average Delay by Airline:
Airline
AirA 15.0
AirB 5.0
AirC 0.0

Most Common Origin: DEL
Most Common Destination: BOM

Plot:



Euclidean Distance & Dot Product

```
import numpy as np

point1 = np.array([1, 2])
point2 = np.array([4, 6])
distance = np.linalg.norm(point1 - point2)
print("Euclidean Distance:", distance)

A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])
dot_product = np.dot(A, B)
print("Dot Product:\n", dot_product)
```

Output:

Euclidean Distance: 5.0

Dot Product:
[[19 22]
 [43 50]]

Time Series Analysis with NumPy & pandas

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dates = pd.date_range(start="2025-01-01", periods=30)
temps = np.random.normal(25, 5, size=30)
df = pd.DataFrame({'Date': dates, 'Temperature': temps})
df.set_index('Date', inplace=True)
print(df.describe())
df.plot(title='Daily Temperatures')
```

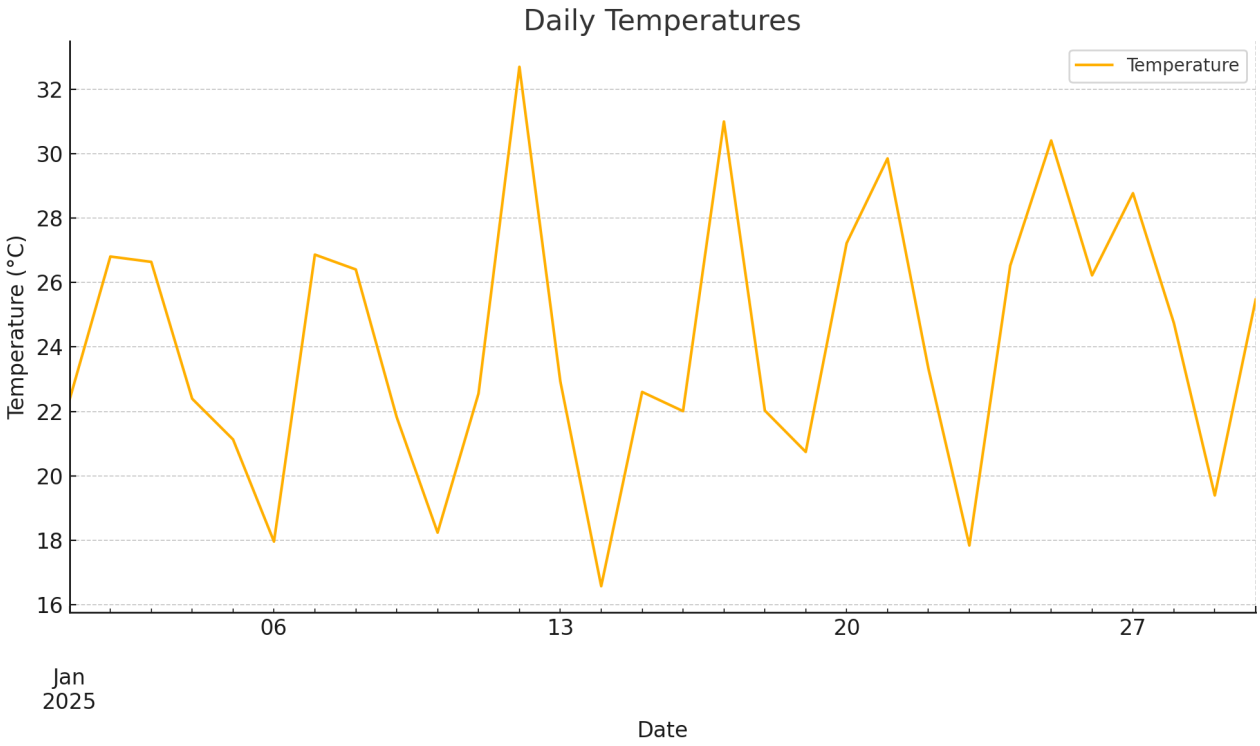
```
plt.ylabel("Temperature (-C)")
plt.show()
```

Output:

Temperature Statistics:

	Temperature
count	30.000000
mean	24.116239
std	4.161110
min	16.573056
25%	21.866546
50%	23.127500
75%	26.764981
max	32.695337

Plot:



Daily Stock Price Simulation

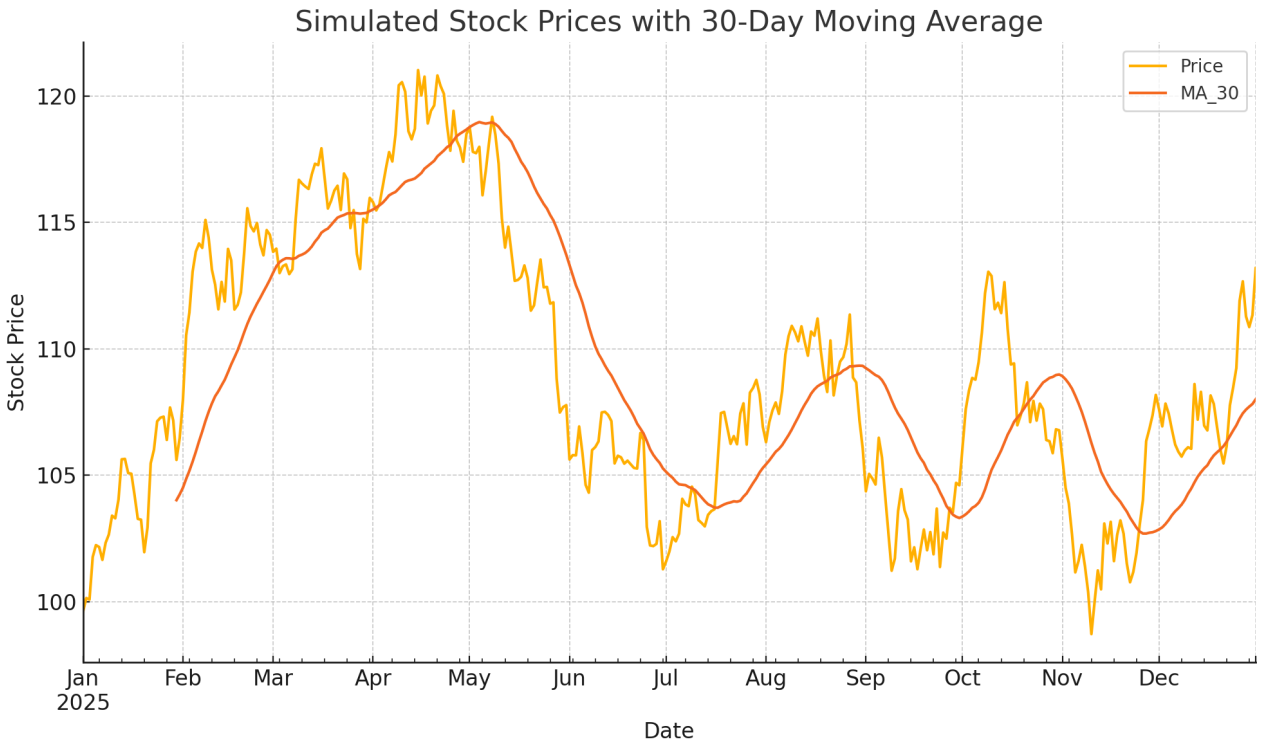
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dates = pd.date_range("2025-01-01", periods=365)
prices = np.cumsum(np.random.normal(0, 1, size=365)) + 100
df = pd.DataFrame({'Date': dates, 'Price': prices})
df.set_index('Date', inplace=True)
df['MA_30'] = df['Price'].rolling(window=30).mean()
df.plot(title="Simulated Stock Prices with 30-Day Moving Average")
plt.ylabel("Stock Price")
plt.show()
```

Output:

Plot generated for simulated stock prices with 30-day moving average.

Plot:



Fruits DataFrame Basic Operations

```
import pandas as pd

data = ['Apple', 'Banana', 'Cherry']
prices = [100, 40, 150]
fruits_df = pd.DataFrame({'Fruit': data, 'Price': prices})
fruits_df.info()
print(fruits_df.to_string(index=False))
print(fruits_df['Price'].describe())
```

Output:

DataFrame Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 2 columns):
 #   Column Non-Null Count Dtype
--- ------ -------------- -----
 0   Fruit   3 non-null   object
 1   Price   3 non-null   int64
dtypes: int64(1), object(1)
memory usage: 180.0+ bytes

Data:
 Fruit Price
 Apple   100
Banana   40
Cherry   150

Price Statistics:

count	3.000000
mean	96.666667
std	55.075705
min	40.000000
25%	70.000000
50%	100.000000
75%	125.000000
max	150.000000