skill-test-3 / Assignment_3 230901114.ipynb

Varun130114 Add files via upload d7860eb · 8 minutes ago

514 lines (514 loc) · 337 KB

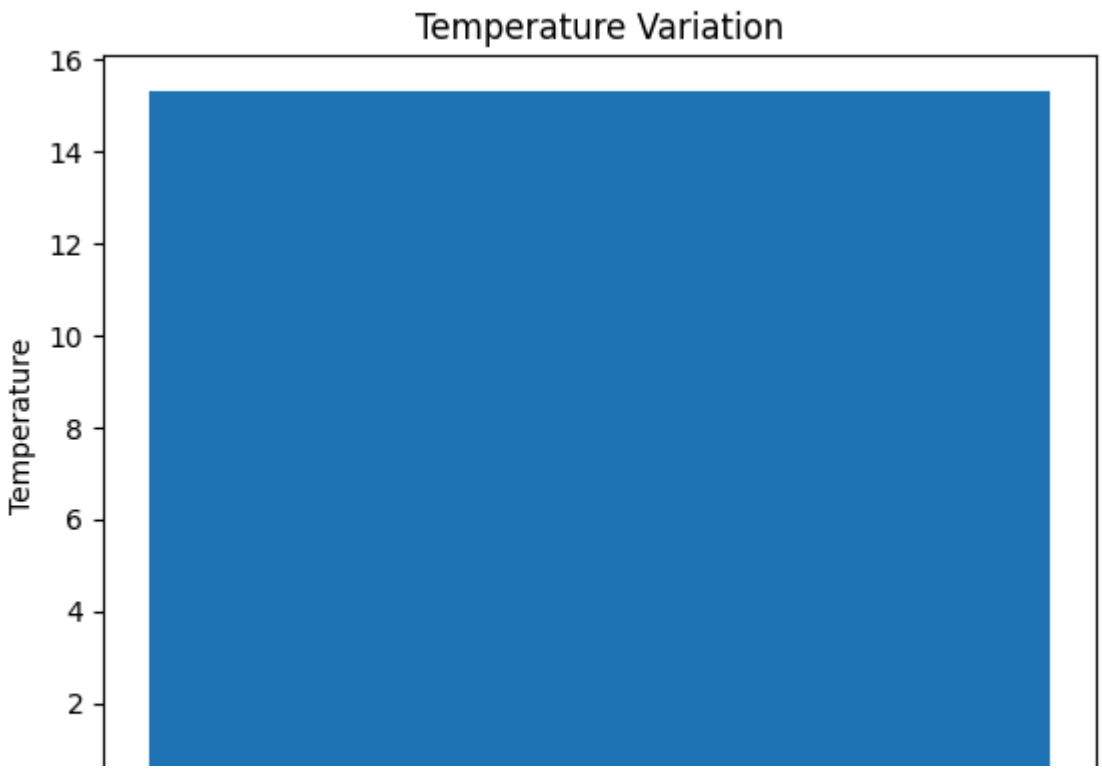
Preview Code Blame Raw

In [112…

```
import pandas as pd
import matplotlib.pyplot as plt
url='https://docs.google.com/spreadsheets/d/162KkAKBBCTB7ZJoGW1BR1IfOI
df=pd.read_csv(url)
print("Original Dataframe:\n",df)
print("\nAverage Temperature:\n",df['Temperature(°C)'].mean())
plt.xlabel('Months')
plt.ylabel('Temperature')
plt.title('Temperature Variation')
plt.bar('January',df['Temperature(°C)'].mean())
plt.show()
```

```
Original Dataframe:
         Date  Temperature(°C)  Humidity(%)  Wind Speed(km/h)  Rainfall
(mm)
0  01-01-2025             15.2           80              10.5
0.0
1  02-01-2025             16.8           75               8.0
0.0
2  03-01-2025             14.5           85              12.0
1.2
3  04-01-2025             13.0           90              15.2
5.4
4  05-01-2025             17.6           70               7.4
0.0
5  06-01-2025             18.1           65               6.0
0.0
6  07-01-2025             16.2           78               9.1
0.5
7  08-01-2025             12.4           88              14.3
2.1
8  09-01-2025             13.8           83              11.5
0.0
9  10-02-2025             15.5           79              10.0
0.0

Average Temperature:
 15.310000000000002
```



In [116…

```
import pandas as pd
url='https://docs.google.com/spreadsheets/d/15eKrZdLqEEt-x0wfykobbPQcC
df=pd.read_csv(url)
print(df[['Maths','Physics','Chemistry']].describe())
df['Avg']=df[['Maths','Physics','Chemistry']].mean(axis=1)
print('\n',df[['Student ID','Avg']])
b=df[['Maths','Physics','Chemistry']]<60
df['b']=b.sum(axis=1)>2
print("\nStudents below 60 marks in more than 2 subjects:")
c=0
for i in range(len(df)):
  if df['b'][i]==True:
    print(df['Student ID'][i])
    c+=1
if c==0:
  print("     No student scored below 60 in morethan 2 subject")
```

```
           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

    Student ID        Avg
0         101  86.666667
1         102  77.666667
2         103  92.000000
3         104  66.333333
4         105  76.000000
5         106  57.666667
6         107  86.333333
7         108  72.000000
8         109  95.000000
9         110  80.000000

Students below 60 marks in more than 2 subjects:
     No student scored below 60 in morethan 2 subject
```

In [117…

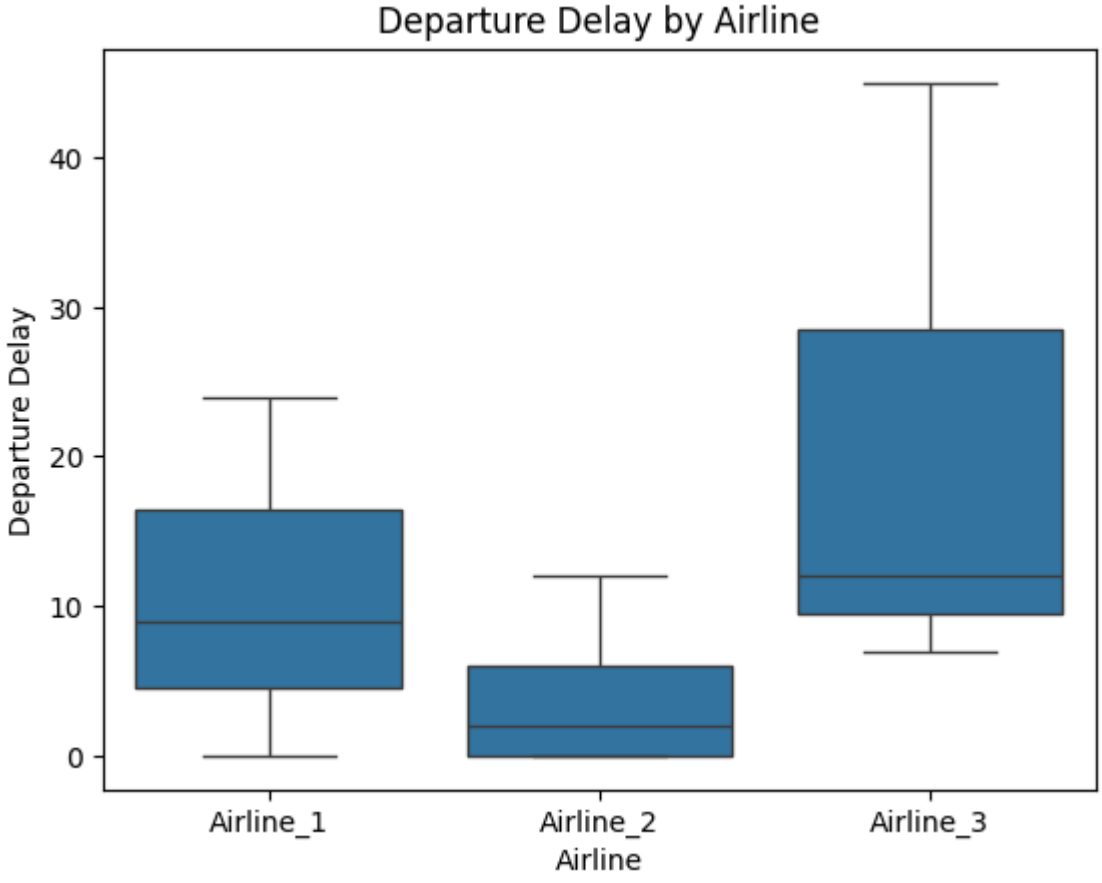
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
url='https://docs.google.com/spreadsheets/d/1fzYRnUl3WYoAFJ_T7iSbIZ8SD
df=pd.read_csv(url)
print("Original dataframe:",df)
print("\nSummary statistics for departure delays:\n",df['Departure Del
k=df.groupby('Airline').agg({'Departure Delay':'mean'})
print("\nAverage Delay for each airline:\n",k)
print("\nMost common origin:",df['Origin'].mode()[0])
print("Most common Destination:",df['Destination'].mode()[0])
print("\n")
sns.boxplot(x='Airline',y='Departure Delay',data=df)
plt.title('Departure Delay by Airline')
plt.show()
```

```
Original dataframe:      Airline  Origin Destination  Departure Time  De
parture Delay
0  Airline_1      UK         USA               6               9
1  Airline_2     USA          UK              12               4
2  Airline_3   India    Srilanka              23              12
3  Airline_2      UK      Canada               8               0
4  Airline_1   Dubai    Srilanka               6              24
5  Airline_3  Canada         USA               4               7
6  Airline_1     USA          UK               8               0
7  Airline_3   Dubai       India              11              45
8  Airline_2      UK    Srilanka              23              12
9  Airline_2   India       Dubai               7               0

Summary statistics for departure delays:
 count    10.000000
mean     11.300000
std      13.976568
min       0.000000
25%       1.000000
50%       8.000000
75%      12.000000
max      45.000000
Name: Departure Delay, dtype: float64

Average Delay for each airline:
           Departure Delay
Airline
Airline_1        11.000000
Airline_2         4.000000
Airline_3        21.333333

Most common origin: UK
Most common Destination: Srilanka
```



In [ ]:
```
import math
import numpy as np
p1=np.array([1,2])
p2=np.array([3,4])
print(f"Euclidean distance between {p1} and {p2} is",np.linalg.norm(p1
A=np.array([[1,2],[3,4]])
B=np.array([[5,6],[7,8]])
print(f"\nDot product of {A} and {B} is",np.dot(A,B))
```

```
Euclidean distance between [1 2] and [3 4] is 2.8284271247461903

Dot product of [[1 2]
 [3 4]] and [[5 6]
 [7 8]] is [[19 22]
 [43 50]]
```

In [122…
```
import datetime
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
dates=pd.date_range(datetime.datetime.today(),periods=365)
df=pd.DataFrame({'Dates':dates})
df['Temperature']=np.random.randint(15,36,size=365)
print("Original DataFrame:\n",df)
print("\nAnalysis:\n",df['Temperature'].describe())
df.set_index('Dates', inplace=True)
monthly_avg = df.resample('M').mean()
print("\nMonthly Average Temperatures:")
print(monthly_avg)
plt.plot(monthly_avg.index, monthly_avg['Temperature'], label='Monthly
plt.xlabel('Month')
plt.ylabel('Temperature')
plt.title('Monthly Average Temperatures')
plt.show()
```

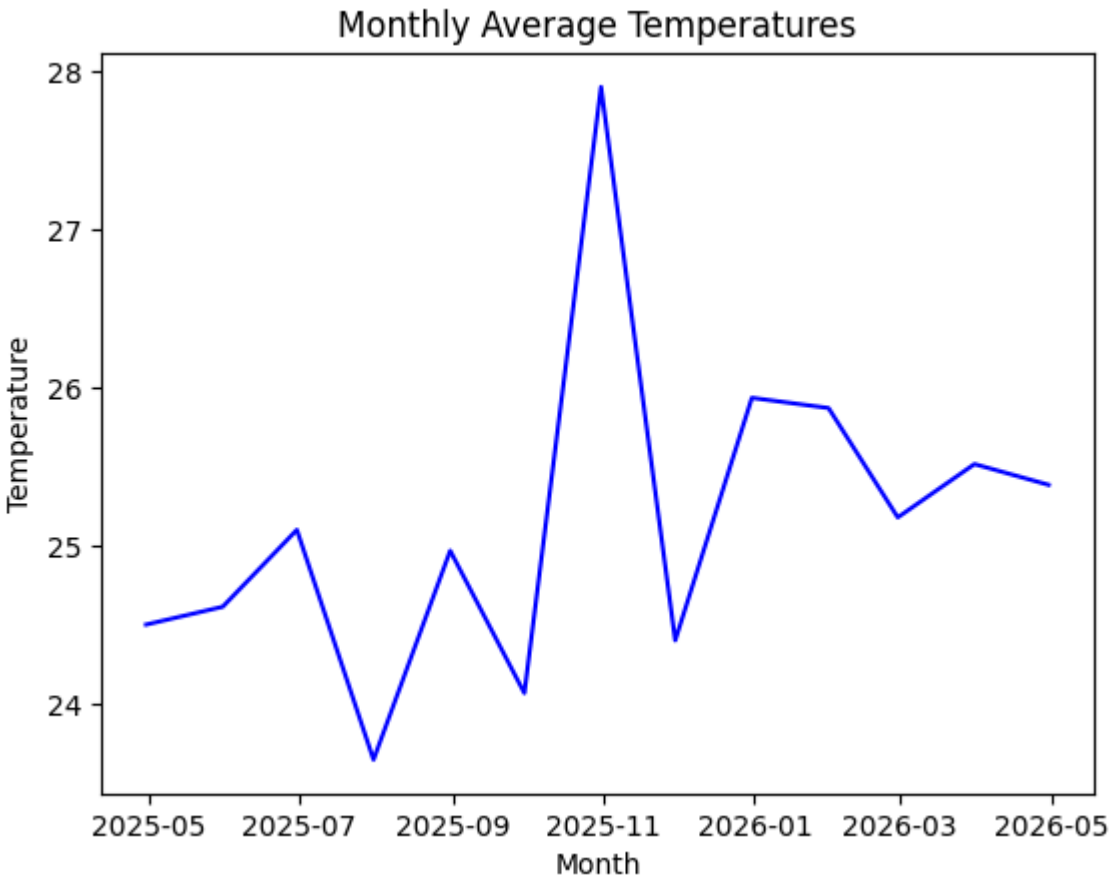
```
Original DataFrame:
                         Dates  Temperature
0   2025-04-27 12:02:49.930759           33
1   2025-04-28 12:02:49.930759           17
2   2025-04-29 12:02:49.930759           22
3   2025-04-30 12:02:49.930759           26
4   2025-05-01 12:02:49.930759           16
..                         ...          ...
360 2026-04-22 12:02:49.930759           33
361 2026-04-23 12:02:49.930759           17
362 2026-04-24 12:02:49.930759           32
363 2026-04-25 12:02:49.930759           19
364 2026-04-26 12:02:49.930759           28

[365 rows x 2 columns]

Analysis:
 count    365.000000
mean      25.210959
std        5.977696
min       15.000000
25%       20.000000
50%       26.000000
75%       31.000000
max       35.000000
Name: Temperature, dtype: float64
```

```
Monthly Average Temperatures:
            Temperature
Dates
2025-04-30    24.500000
2025-05-31    24.612903
2025-06-30    25.100000
2025-07-31    23.645161
2025-08-31    24.967742
2025-09-30    24.066667
2025-10-31    27.903226
2025-11-30    24.400000
2025-12-31    25.935484
2026-01-31    25.870968
2026-02-28    25.178571
2026-03-31    25.516129
2026-04-30    25.384615
```

```
<ipython-input-122-f52269b1aff7>:11: FutureWarning: 'M' is deprecated an
d will be removed in a future version, please use 'ME' instead.
  monthly_avg = df.resample('M').mean()
```



In [ ]:

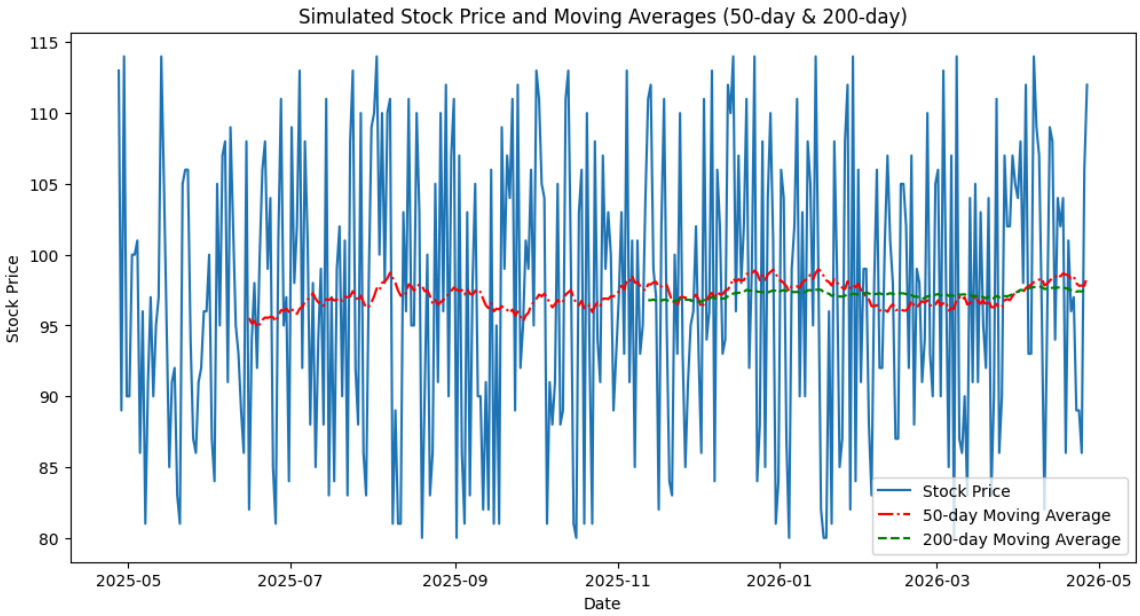
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
dates=pd.date_range(datetime.datetime.today(),periods=365)
df=pd.DataFrame({'Date':dates})
df.set_index('Date',inplace=True)
df['Stock Price']=np.random.randint(80,115,size=365)
print("Original DataFrame:\n",df)
print("\nBasic Metrics:",df['Stock Price'].describe())
df['50-day MA']=df['Stock Price'].rolling(window=50).mean()
df['200-day MA']=df['Stock Price'].rolling(window=200).mean()
fig=plt.figure(figsize=(12,6))
plt.plot(df.index,df['Stock Price'],label='Stock Price')
plt.plot(df.index,df['50-day MA'],label='50-day Moving Average',color=
```

```
plt.plot(df.index,df['200-day MA'],label='200-day Moving Average',colo
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.title('Simulated Stock Price and Moving Averages (50-day & 200-day
plt.legend()
plt.show()
```

```
Original DataFrame:
                            Stock Price
Date
2025-04-27 11:35:52.709460          113
2025-04-28 11:35:52.709460           89
2025-04-29 11:35:52.709460          114
2025-04-30 11:35:52.709460           90
2025-05-01 11:35:52.709460           90
...                                 ...
2026-04-22 11:35:52.709460           89
2026-04-23 11:35:52.709460           89
2026-04-24 11:35:52.709460           86
2026-04-25 11:35:52.709460          106
2026-04-26 11:35:52.709460          112

[365 rows x 1 columns]

Basic Metrics: count    365.000000
mean      97.164384
std        9.744840
min       80.000000
25%       90.000000
50%       97.000000
75%      105.000000
max      114.000000
Name: Stock Price, dtype: float64
```



In [ ]:

```
import pandas as pd
import numpy as np
fruits=['Apple','Banana','Mango','Orange','Dragon Fruit','Grapes','Kiw
price=[120,150,90,60,100,80,78,84,112,132]
Fruit_df=pd.DataFrame({'Fruits':fruits,'Price':price})
print("Original DataFrame:\n",df)
print("\nInfo:")
print(Fruit_df.info())
print('\n')
print(Fruit_df.to_string())
```

```
print("\n")
print(Fruit_df.describe())
```

```
Original DataFrame:
          Fruits  Price
0         Apple    120
1        Banana    150
2         Mango     90
3        Orange     60
4  Dragon Fruit    100
5        Grapes     80
6          Kiwi     78
7        Papaya     84
8    Watremelon    112
9      Rasberry    132

Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Fruits  10 non-null     object
 1   Price   10 non-null     int64
dtypes: int64(1), object(1)
memory usage: 292.0+ bytes
None


         Fruits  Price
0         Apple    120
1        Banana    150
2         Mango     90
3        Orange     60
4  Dragon Fruit    100
5        Grapes     80
6          Kiwi     78
7        Papaya     84
8    Watremelon    112
9      Rasberry    132


            Price
count   10.000000
mean   100.600000
std     27.697573
min     60.000000
25%     81.000000
50%     95.000000
75%    118.000000
max    150.000000
```