

Theory Activity No. 1

Name : Sowmya Gourkhede

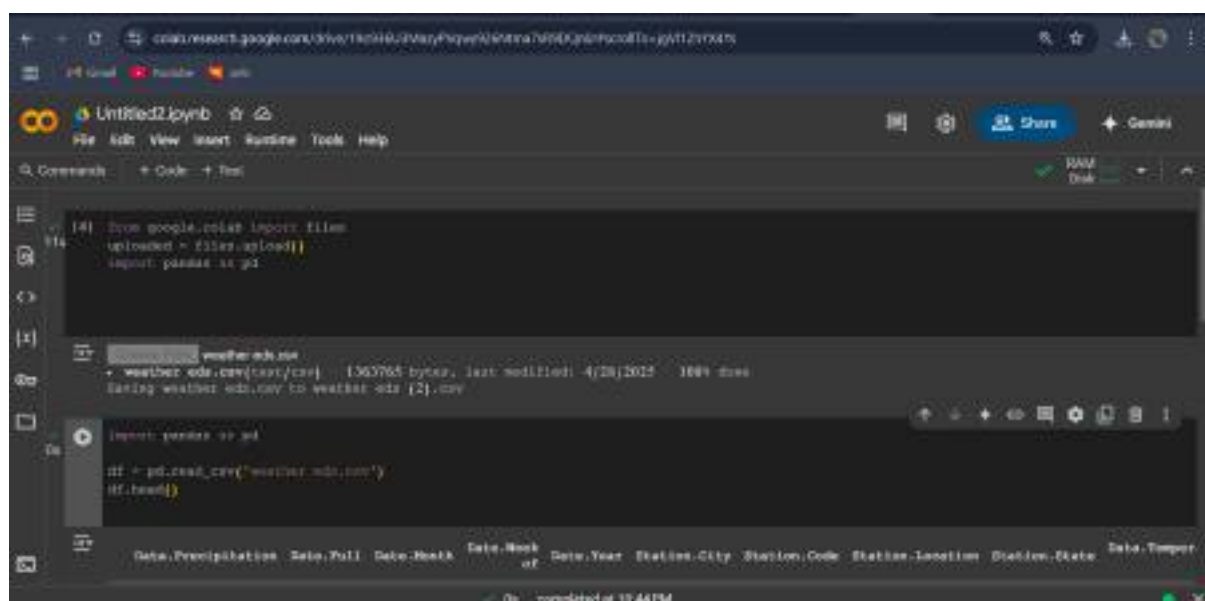
PRN: 202401100108

Roll No: 86

Class/Batch: CS5-4

Formulate 20 problem statements for a given dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements.

The dataset name is **weather dataset**.



```
from google.colab import Files
uploaded = Files.upload()
import pandas as pd

weather.csv
+ weather.csv [text/csv] · 150765 bytes, last modified: 4/28/2015 · 100% done
Saving weather.csv to weather.csv [2].csv

import pandas as pd

df = pd.read_csv('weather.csv')
df.head()
```

Date	Precipitation	Date_Fall	Date_Winter	Date_Spring	Date_Summer	Station_City	Station_Code	Station_Longitude	Station_State	Date_Temp
2015-01-01	0.0	2015-01-01	2015-01-01	2015-01-01	2015-01-01	San Francisco	1413	-122.42	CA	50.0
2015-01-02	0.0	2015-01-02	2015-01-02	2015-01-02	2015-01-02	San Francisco	1413	-122.42	CA	50.0
2015-01-03	0.0	2015-01-03	2015-01-03	2015-01-03	2015-01-03	San Francisco	1413	-122.42	CA	50.0
2015-01-04	0.0	2015-01-04	2015-01-04	2015-01-04	2015-01-04	San Francisco	1413	-122.42	CA	50.0
2015-01-05	0.0	2015-01-05	2015-01-05	2015-01-05	2015-01-05	San Francisco	1413	-122.42	CA	50.0

The screenshot shows a Jupyter Notebook with a pandas DataFrame containing weather data for Alabama in 2016. The DataFrame has the following columns: Data.Precipitation, Data.Fall, Data.Month, Data.Week, Data.Year, Station.City, Station.Code, Station.Location, Station.State, and Data.Temper. The data is as follows:

	Data.Precipitation	Data.Fall	Data.Month	Data.Week	Data.Year	Station.City	Station.Code	Station.Location	Station.State	Data.Temper
0	0.00	01-01-2016	1	1	2016	Birmingham	BHM	Birmingham, AL	Alabama	
1	0.00	01-01-2016	1	1	2016	Huntsville	HUN	Huntsville, AL	Alabama	
2	0.16	01-01-2016	1	1	2016	Mobile	MOB	Mobile, AL	Alabama	
3	0.00	01-01-2016	1	1	2016	Montgomery	MTM	Montgomery, AL	Alabama	
4	0.01	01-01-2016	1	1	2016	Anchorage	ANC	Anchorage, AK	Alaska	

1) Find the maximum (avg) temperature recorded.

```
[11] max_avgtemperature = df['Data.Temperature.Avg Temp'].max()
      print('Maximum Temperature Recorded:', max_avgtemperature, '*C')
```

Maximum Temperature Recorded: 100 °C

2) Find the minimum wind direction recorded.

```
min_data_wind_direction = df['Data.Wind.Direction'].min()
print(min_data_wind_direction)
```

0

3) Find the maximum wind direction recorded

```
max_wind_direction = df['Data.Wind.Direction'].max()
print('Maximum wind direction Recorded:', max_wind_direction, '°C')
Maximum wind direction Recorded: 36 °C
```

4) Calculate the average wind speed.

```
avg_wind_speed = df['Data.Wind.Speed'].mean()
print(avg_wind_speed)
6.3298202233769345
```

5) Find the minimum (avg) temperature recorded.

```
import pandas as pd
import numpy as np
min_avg_temp = df['Data.Temperature.Avg Temp'].min()
print(min_avg_temp)
-27
```

6) Calculate the maximum wind speed.

```
max_wind_speed = df['Data.Wind.Speed'].max()
print('Maximum wind speed Recorded:', max_wind_speed)
Maximum wind speed Recorded: 61.1
```

7) Calculate the minimum wind speed.

```
min_wind_speed = df['Data.Wind.Speed'].min()
print(min_wind_speed)
```

0.0

8)Find the Average wind direction.

```
avg_wind_direction = df['Data.Wind.Direction'].mean()
print(avg_wind_direction)
```

18.791315773756196

9)Filter the dataset for a State Location.

```
station_location = df[df['Station.Location'] == 'Huntsville, AL']  
print(station_location)
```

	Data.Precipitation	Date.Full	Date.Month	Date.Week of	Date.Year	\
1	0.00	03-01-2016	1	3	2016	
316	0.43	10-01-2016	1	10	2016	
631	0.16	17-01-2016	1	17	2016	
946	2.38	24-01-2016	1	24	2016	
1261	0.60	31-01-2016	1	31	2016	
1576	2.27	07-02-2016	2	7	2016	
1891	0.01	14-02-2016	2	14	2016	
2206	2.56	21-02-2016	2	21	2016	
2521	1.54	28-02-2016	2	28	2016	
2836	0.70	06-03-2016	3	6	2016	
3153	1.20	13-03-2016	3	13	2016	
3468	0.60	20-03-2016	3	20	2016	
3783	0.14	27-03-2016	3	27	2016	
4098	0.00	03-04-2016	4	3	2016	
4415	0.41	10-04-2016	4	10	2016	
4731	1.37	17-04-2016	4	17	2016	
5047	0.04	24-04-2016	4	24	2016	
5363	0.08	01-05-2016	5	1	2016	
5680	0.41	08-05-2016	5	8	2016	
5997	1.04	15-05-2016	5	15	2016	
6314	0.09	22-05-2016	5	22	2016	
6631	0.00	29-05-2016	5	29	2016	
6948	2.56	05-06-2016	6	5	2016	
7265	0.03	12-06-2016	6	12	2016	



```
station_location = df[df['Station.Location'] == 'Huntsville, AL']  
print(station_location)
```

[4]

	Station.City	Station.Code	Station.Location	Station.State	\
1	Huntsville	HSV	Huntsville, AL	Alabama	
316	Huntsville	HSV	Huntsville, AL	Alabama	
631	Huntsville	HSV	Huntsville, AL	Alabama	
946	Huntsville	HSV	Huntsville, AL	Alabama	
1261	Huntsville	HSV	Huntsville, AL	Alabama	
1576	Huntsville	HSV	Huntsville, AL	Alabama	
1891	Huntsville	HSV	Huntsville, AL	Alabama	
2206	Huntsville	HSV	Huntsville, AL	Alabama	
2521	Huntsville	HSV	Huntsville, AL	Alabama	
2836	Huntsville	HSV	Huntsville, AL	Alabama	
3153	Huntsville	HSV	Huntsville, AL	Alabama	
3468	Huntsville	HSV	Huntsville, AL	Alabama	
3783	Huntsville	HSV	Huntsville, AL	Alabama	
4098	Huntsville	HSV	Huntsville, AL	Alabama	
4415	Huntsville	HSV	Huntsville, AL	Alabama	
4731	Huntsville	HSV	Huntsville, AL	Alabama	
5047	Huntsville	HSV	Huntsville, AL	Alabama	
5363	Huntsville	HSV	Huntsville, AL	Alabama	
5680	Huntsville	HSV	Huntsville, AL	Alabama	
5997	Huntsville	HSV	Huntsville, AL	Alabama	
6314	Huntsville	HSV	Huntsville, AL	Alabama	
6631	Huntsville	HSV	Huntsville, AL	Alabama	
6948	Huntsville	HSV	Huntsville, AL	Alabama	
7265	Huntsville	HSV	Huntsville, AL	Alabama	

```
station_location = df[df['Station.Location'] == 'Huntsville, AL']  
print(station_location)
```

	Data.Temperature.Avg Temp	Data.Temperature.Max Temp	\
1	39	47	
316	43	50	
631	39	49	
946	33	40	
1261	46	58	
1576	48	60	
1891	36	45	
2206	51	58	
2521	50	59	
2836	47	58	
3153	66	75	
3468	59	71	
3783	56	67	
4098	57	66	
4415	56	70	
4731	62	71	
5047	67	79	
5363	71	78	
5680	64	77	
5997	70	80	
6314	69	80	
6631	75	88	
6948	79	88	
7265	77	91	

	Data.Temperature.Min	Temp	Data.Wind.Direction	Data.Wind.Speed
1		31	32	3.86
316		34	13	8.34
631		29	22	4.57
946		25	19	6.44
1261		34	20	6.40
1576		37	23	5.98
1891		26	20	8.61
2206		43	21	6.45
2521		39	19	8.44
2836		36	23	3.85
3153		57	14	8.61
3468		46	25	6.01
3783		45	21	7.57
4098		47	20	5.56
4415		43	17	7.08
4731		51	10	7.97
5047		53	14	4.21
5363		63	19	4.50
5680		50	29	5.64
5997		60	20	6.00
6314		58	20	5.45
6631		62	21	4.14
6948		69	15	2.32
7265		62	22	2.95
7580		70	22	4.54
7895		72	19	3.51
8210		68	28	3.00
8526		73	23	5.35

10) Calculate the variance of temperature.

```

var_datatemp = df['Data.Temperature.Avg Temp'].var()
print(var_datatemp)

```

353.37588371123974

11) Calculate the var of Wind direction


```
var_data_wind_dir = df['Data.Wind.Direction'].var()  
print(var_data_wind_dir)
```

```
41.751335402789486
```

12) Calculate the var of Wind speed

```
var_data_wind_speed = df['Data.Wind.Speed'].var()  
print(var_data_wind_speed)
```

```
12.213523961227596
```

13) Count the no. of times Alabama occurs under station state.

```
stationstate = (df['Station.State'] == 'Alabama').sum()  
print(stationstate)
```

```
212
```

14) Count the no. of times Nome AK occurs under station location.

```
stationloc = (df['Station.Location'] == 'Nome, AK').sum()  
print(stationloc)
```

```
53
```

15) Find the number of unique wind directions.

```
unique_direction = df['Data.Wind.Direction'].nunique()  
print(unique_direction)
```

37

16) Find the number of unique wind speed.

```
unique_speed = df['Data.Wind.Speed'].nunique()  
print(unique_speed)
```

1461

17) Find the number of unique station city.

```
unique_city = df['Station.City'].nunique()  
print(unique_city)
```

307

18) Filter the dataset for days with temperature below 0°C.

```
freezing_days = df[df['Data.Temperature.Min Temp'] < 0]  
print(freezing_days)
```

```

Data.Temperature.Min Temp  Data.Wind.Direction  Data.Wind.Speed
23      -9      20      0.36
62      -5      32      1.40
64     -23      12      0.40
97      -4       2      2.66
166     -12     15      1.80
...
16450     -9     18      2.04
16461     -6     12     14.01
16490    -11     20      0.80
16594     -1     21      3.60
16657     -3     19      2.05

[214 rows x 14 columns]

```

19) Filter the dataset for days with speed below 10.

```

wind_speed= df[df['Data.Wind.Speed'] < 10]
print(wind_speed)

```

```

Data.Temperature.Min Temp  Data.Wind.Direction  Data.Wind.Speed
0      32      33      4.33
1      31      32      3.86
2      41      35      9.73
3      38      32      6.86
4      29      19      7.80
...
16733     35     22      7.68
16735     20     25      9.51
16736     19     25      6.33
16740      4     26      1.65
16742      8     23      7.51

[14810 rows x 14 columns]

```

20) Check if there are any missing values in the dataset.

```
missing_values = df.isnull().sum()  
print(missing_values)
```

```
⇒ Data.Precipitation      0  
   Date.Full              0  
   Date.Month             0  
   Date.Week of          0  
   Date.Year              0  
   Station.City           0  
   Station.Code           0  
   Station.Location       0  
   Station.State          0  
   Data.Temperature.Avg Temp 0  
   Data.Temperature.Max Temp 0  
   Data.Temperature.Min Temp 0  
   Data.Wind.Direction     0  
   Data.Wind.Speed         0  
dtype: int64
```

21) Find how many days had wind speed above 30 km/h.

```
high_wind_days = (df['Data.Wind.Speed'] > 30).sum()  
print(high_wind_days)
```

```
⇒ 30
```