

ASSIGNMENT-1

1. SOURCE CODE → import pandas as pd

```
data = {'Roll': [100, 102, 103, 104],
        'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Total Marks': [85, 92, 78, 88],
        'Class': ['10th', '10th', '10th', '12th']}

```

```
df = pd.DataFrame(data)
```

```
df_sorted = df.sort_values(by=['Name', 'Total Marks'])
print(df_sorted)
```

■ OUTPUT →

	Roll	Name	Total Marks	Class
0	100	Alice	85	10th
1	102	Bob	92	10th

2. ■ SOURCE CODE → data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],  
 'Age': [24, 22, 23, 25],  
 'Rating': [4.5, 4.0, 4.7, 4.8]}

```
df = pd.DataFrame(data)
```

a) Sum\_cols = df.sum(numeric\_only=True)  
 print("Sum of Columns: \n", Sum\_cols)

b) Avg\_age = df['Age'].mean()

```
Avg_rating = df['Rating'].mean()
```

```
Print("Average Age: ", Avg_age)
```

```
Print("Average Rating: ", Avg_rating)
```

c) Std\_dev = df.describe()

```
Print("Standard Deviation: ", Std_dev)
```

Name: Debachuti Banerjee

Roll No.: 62

Section: A

Year: 3<sup>rd</sup> (2024)



d) desc = df.describe()

- Print ("Description: ", desc)

■ OUTPUT → Sum of Columns : Age 94.0  
Rating 18.0

b) Average age : 23.5

Average Rating : 4.5

c) Standard Deviation : Age 1.290994  
Rating 0.355903

d) Description :	Age	Rating
Count	4.00	4.00
mean	23.500	4.500
Std	1.290	0.355
min	22.000	4.000
max	25.000	4.800
25%	22.750	4.3750
50%	23.500	4.6000
75%	24.250	4.7250

3. ■ SOURCE CODE → import numpy as np  
df = pd.DataFrame(np.random.randn(5,3), columns=['A', 'B', 'C'])  
row\_sum = df.apply(np.sum, axis=1)  
Print (row\_sum)

■ OUTPUT → 0 0.306601  
1 1.628716  
2 -2.782953  
3 -0.407796

4. ■ SOURCE CODE → data = {'A': [1.0, 4.0, 7.0, np.nan],  
'B': [2, 5, 8, np.nan],  
'C': [3.0, 6.0, 9.0, np.nan]}

df = pd.DataFrame(data)

Name: Debahuti Banerjee

Roll No: 62

Section: A

Year: 3<sup>rd</sup> (2024)



- a) `Print (df[:10])`
- b) `row_agg = df.agg(['Sum', 'Min', 'Max'], axis=1)`  
`Print ("Row wise aggregate : ", row_agg)`
- c) `col_agg = df.agg(['Sum', 'Min', 'Max'], axis=0)`  
`Print ("Column-wise aggregate : ", col_agg)`

■ OUTPUT → a)

	A	B	C
0	1.0	2	3.0
1	4.0	5	6.0

b) Row-wise aggregate :

	Sum	Min	Max
0	6.0	1.0	3.0
1	15.0	4.0	6.0

c) Column-wise aggregate :

	Sum	Min	Max
0	15.0	1.0	18.0
1	1.0	2.0	3.0

5. ■ SOURCE CODE → `Ph_data = pd.read_csv("phone_data.csv")`

- a) `num_rows = ph_data.head(0)`
- b) `longest_entry = ph_data['duration'].max()`
- c) `total_ph_sec = ph_data[ph_data['item'] == 'call']`  
`['duration'].sum()`
- d) `unique_networks = ph_data['network'].unique()`
- e) `ph_data['month'] = pd.to_datetime(ph_data['date']).dt.month`
- f) `1st_entry_per_month = ph_data.groupby('month').first()`
- g) `sum_dur_per_mon = ph_data.groupby('month')['duration'].sum()`
- h) `num_entries_per_month = ph_data.groupby('month')['date'].unique()`
- i) `sum_dur_calls_per_net = ph_data[ph_data['item'] == 'call'].groupby('network')['duration'].sum()`
- j) `calls_sms_data_per_month = ph_data.groupby(['month', 'item']).size().unstack(fill_value=0)`

Name: Debaluti Banerjee

Roll No.: 62

Section: A

Year: 3<sup>rd</sup> (2024)



```

k) monthly_network_split = ph_data.groupby(['month', 'network',
        'item']).size().unstack(fill_value=0)

Print("Number of rows: ", num_rows)
Print("Longest entry duration: ", longest_entry)
Print("Total phone call seconds: ", total_ph_sec)
Print("Unique Networks: ", unique_networks)
Print("Entries per month: ", entries_per_month)
Print("First entry per month: ", 1st_entry_per_month)
Print("Sum of duration per month: ", sum_dur_per_mon)
Print("No of entries per month: ", num_entries_per_month)
Print("Sum of durations for calls per network: ", sum_dur_calls_per_net)
Print("Calls, Sms Data entries per month: ", calls_sms_data_per_month)
Print("Monthly network split: ", monthly_network_split)
    
```

■ OUTPUT → a) Number of rows: 830

b) Longest entry duration: 10528.0

c) Total phone call seconds: 92321.0

d) Unique Networks: 9

e) Entries per month: 

month	entries
12	150
11	132
10	125

f) First Entry per month:

month	index	date	duration	item	network	network_type
1	111	01/11/14	06:58	data	data	data
2	114	02/11/14	06:58	SMS	vodafone	mobile

g) Sum of duration per month: 

month	sum
1	13923.296
2	12775.009

h) Number of entries per month: 

month	num
1	99
2	81

Name: Debahuti Banerjee

Roll No.: 62

Section: A

Year: 2nd (2024)



- i) Sum of durations for calls per network: network  
 Meteor 7200.0  
 Tesco 13828.0
- j) Calls, sms, Data entries per month: month item call sms data  
 1 54 29 24  
 2 47 23 21
- k) Monthly network split: month network call data sms  
 1 Meteor 5 0 0  
 Tesco 17 0 0

6. ■ SOURCE CODE → game\_data = pd.read\_csv('asiangamestep10.csv')  
 print(game\_data)

a) total\_rec = game\_data.shape[0]

b) top\_n\_rec = game\_data.head(n=10)

c) game\_data.rename(columns = {'NOC': 'Country'}, inplace = True)

d) top15\_year\_medals = game\_data.groupby('Year')['Total'].sum().nlargest(15)  
 top15\_year\_medals.plot(kind='bar')

e) top15\_country\_medals = game\_data.groupby('country')['Total'].sum().nlargest(15)  
 top15\_country\_medals.plot(kind='bar')

f) max\_medals = game\_data['Total'].max()

min\_medals = game\_data['Total'].min()

mean\_medals = game\_data['Total'].mean()

Print("Total no. of records: ", total\_rec)

Print("Total 10 records: ", top\_n\_rec)

Print("Max medals: ", max\_medals)

Print("Min medals: ", min\_medals)

Print("Mean medals: ", mean\_medals)

■ OUTPUT → Total no of records: 444  
 Top 10 records: 0 Year 1951 NOC Japan(JPN) Gold 24 Silver 21 Bronze 15 Total 60  
 max medals: 416  
 Min medals: 1  
 Mean medals: 34.448198

Name: Debanuti Banerjee

Roll No.: 62

Section: A

Year: 3<sup>rd</sup> (2024)