```python
# Programed By PREM RAHANGDALE
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns

print("Numpy version :", np.__version__)
print("Pandas version :", pd.__version__)
print("Matplotlib version :", plt.__version__)
print("Seaborn version :", sns.__version__)
```

```
Numpy version : 1.23.1
Pandas version : 1.4.3
Matplotlib version : 3.5.2
Seaborn version : 0.11.2
```

```python
# Programed By PREM RAHANGDALE
import pandas as pd
df=pd.read_csv("Marks.csv")
print(df.head())
print(df.columns)
print(df.describe())
df.to_csv("update_marks.csv", index=False)
```

```
       Name Gender         DOB   AAP    OS  DSPD    PC    ST   E&P
0      Prem      M   22/4/2005  85.0  87.0  89.0  88.0  89.0  95.0
1    Prem R      M    1/1/2000  75.0  64.0  90.0  61.0  58.0   2.0
2  Anshuman      M  25-05-2006  25.0  76.0  95.0  87.0  56.0  74.0
3     Mansi      F   12/8/2004  78.0  63.0  54.0  89.0  75.0  45.0
4     Kanak      F    2/9/2005  58.0  46.0  96.0  77.0  83.0  53.0
Index(['Name', 'Gender', 'DOB', 'AAP', 'OS', 'DSPD', 'PC', 'ST', 'E&P'], dtype='object')
             AAP         OS       DSPD         PC         ST        E&P
count   8.000000   8.000000   8.000000   8.000000   8.000000   8.000000
mean   59.375000  67.375000  72.375000  75.625000  73.500000  58.625000
std    19.588171  15.268431  25.489143  14.889474  12.861904  28.878749
min    25.000000  46.000000  25.000000  56.000000  56.000000   2.000000
25%    51.750000  57.500000  54.750000  60.250000  62.500000  50.250000
50%    56.500000  63.500000  82.000000  82.000000  75.500000  57.000000
75%    75.750000  78.750000  91.250000  88.250000  84.000000  77.250000
max    85.000000  89.000000  96.000000  89.000000  89.000000  95.000000
```

```python
# Programed By PREM RAHANGDALE
import pandas as pd
import json
with open("code2.json") as f:
    data=json.load(f)
df=pd.DataFrame([data])
print(df.head())
df.to_json("code3.json",orient="records",index=4)
```

```
              Name    City Roll No
0  Prem Rahangdale  Gondia     269
```

# Programed by PREM RAHANGDALE

## Practical No. 04 - Output

[1]:
```python
import pandas as pd
import numpy as np

data = {
    'Name': ['Prem', 'Ankit', 'Tohid', 'Ram'],
    'Age': [21, np.nan, 30, np.nan],
    'Salary': [60000, 50000, np.nan, 45000],
    'Department': ['GD', 'HR', 'IT', 'HR']
}
df = pd.DataFrame(data)

# Detecting missing values
print("Missing Values in DataFrame:\n", df.isnull().sum())

df['Age'] = df['Age'].fillna(df['Age'].mean())
df['Salary'] = df['Salary'].fillna(df['Salary'].median())
print("\nDataFrame after handling missing values:\n", df)

filtered_df = df[df['Salary'] > 50000]
print("\nFiltered DataFrame (Salary > 50000):\n", filtered_df)

sorted_df = df.sort_values(by='Age', ascending=False)
print("\nSorted DataFrame (by Age Descending):\n", sorted_df)

grouped_df = df.groupby('Department')['Salary'].mean()
print("\nGrouped DataFrame (Average Salary by Department):\n", grouped_df)
```

```
Missing Values in DataFrame:
 Name          0
Age            2
Salary         1
Department     0
dtype: int64

DataFrame after handling missing values:
     Name   Age   Salary Department
0    Prem  21.0  60000.0         GD
1   Ankit  25.5  50000.0         HR
2   Tohid  30.0  50000.0         IT
3     Ram  25.5  45000.0         HR

Filtered DataFrame (Salary > 50000):
    Name   Age   Salary Department
0   Prem  21.0  60000.0         GD

Sorted DataFrame (by Age Descending):
     Name   Age   Salary Department
2   Tohid  30.0  50000.0         IT
1   Ankit  25.5  50000.0         HR
3     Ram  25.5  45000.0         HR
0    Prem  21.0  60000.0         GD

Grouped DataFrame (Average Salary by Department):
 Department
GD    60000.0
HR    47500.0
IT    50000.0
Name: Salary, dtype: float64
```