1 . Calculate The Mean And Standard Deviation.

```
import pandas as pd
import numpy as np
from scipy import stats
data = {
   "Values": [12, 15, 14, 16, 12, 15, 14, 14, 13, 16, 17, 14, 14, 12, 15, 14,
13, 17, 12, 16]
}

df = pd.DataFrame(data)
mean_value = df["Values"].mean()
median_value = df["Values"].median()
mode_value = stats.mode(df["Values"])[0][0]
std_dev = df["Values"].std()

print(f"Mean: {mean_value}")
print(f"Median: {median_value}")
print(f"Mode: {mode_value}")
print(f"Standard Deviation: {std_dev}")
```

Output:

Mean: 14.25
Median: 14.0
Mode: 14
Standard Deviation: 1.6181535936466533

2 . Read The CSV File.

```
import pandas as pd
df = pd.read_csv("data.csv")
print("Original Data:\n", df.head())

# ---- Apply Data Mining Filters ----
# 1. Filter rows where 'Salary' is greater than 50,000
high_salary = df[df["Salary"] > 50000]
print("\nEmployees with Salary > 50,000:\n", high_salary)

# 2. Select employees from the 'IT' department
it_department = df[df["Department"] == "IT"]
print("\nEmployees in IT Department:\n", it_department)

# 3. Filter rows where 'Age' is between 25 and 40
age_filter = df[(df["Age"] >= 25) & (df["Age"] <= 40)]
print("\nEmployees aged between 25 and 40:\n", age_filter)

# 4. Filter only selected columns: 'Name', 'Salary', 'Department'
selected_columns = df[["Name", "Salary", "Department"]]
print("\nSelected Columns:\n", selected_columns)

 5. Remove duplicate rows
df_cleaned = df.drop_duplicates()
print("\nData after removing duplicates:\n", df_cleaned.head())

# 6. Handle missing values (fill with mean)
df_filled = df.fillna(df.mean(numeric_only=True))
print("\nData after filling missing values:\n", df_filled.head())
```

Output:

```
PS C:\Users\Smart\Desktop\Data Mining> python -u "c:\Users\Smart\Desktop\Data Mining\2.py"
Original Data:
      Name  Age  Salary Department  Experience
0   Ganesh   25   55000         IT           3
1   Yogesh   30   72000    Finance           5
2   Mahesh   28   48000         HR           2
3   Dinesh   35   85000         IT           8
4  Harshal   40   60000  Marketing          10

Employees with Salary > 50,000:
       Name  Age  Salary Department  Experience
0    Ganesh   25   55000         IT           3
1    Yogesh   30   72000    Finance           5
3    Dinesh   35   85000         IT           8
4   Harshal   40   60000  Marketing          10
5    Sachin   27   53000         HR           4
6    Ramesh   32   76000         IT           6
7    Suresh   29   51000    Finance           3
8    Rajesh   45   90000        CEO          20
9   Prakash   50   92000        CEO          22
11    Kiran   31   58000         HR           5

Employees in IT Department:
      Name  Age  Salary Department  Experience
0   Ganesh   25   55000         IT           3
3   Dinesh   35   85000         IT           8
6   Ramesh   32   76000         IT           6
10    Alok   23   42000         IT           1

Employees aged between 25 and 40:
       Name  Age  Salary Department  Experience
0    Ganesh   25   55000         IT           3
1    Yogesh   30   72000    Finance           5
2    Mahesh   28   48000         HR           2
3    Dinesh   35   85000         IT           8
4   Harshal   40   60000  Marketing          10
5    Sachin   27   53000         HR           4
6    Ramesh   32   76000         IT           6
7    Suresh   29   51000    Finance           3
11    Kiran   31   58000         HR           5
```

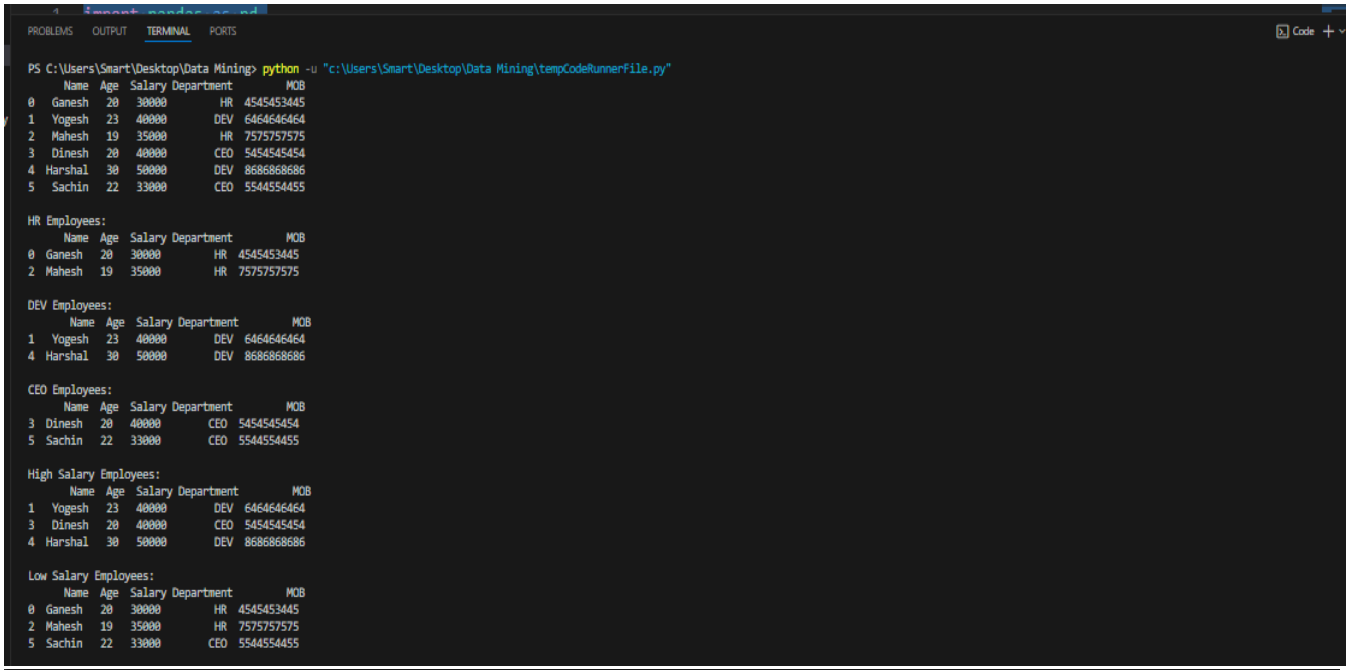
```
10    Alok   42000         IT
11   Kiran   58000         HR

Data after removing duplicates:
      Name  Age  Salary Department  Experience
0   Ganesh   25   55000         IT           3
1   Yogesh   30   72000    Finance           5
2   Mahesh   28   48000         HR           2
3   Dinesh   35   85000         IT           8
4  Harshal   40   60000  Marketing          10

Data after filling missing values:
      Name  Age  Salary Department  Experience
0   Ganesh   25   55000         IT           3
1   Yogesh   30   72000    Finance           5
2   Mahesh   28   48000         HR           2
3   Dinesh   35   85000         IT           8
4  Harshal   40   60000  Marketing          10
PS C:\Users\Smart\Desktop\Data Mining>
```

3. Perform Data Filtering, And Calculate Aggregate Statistics.

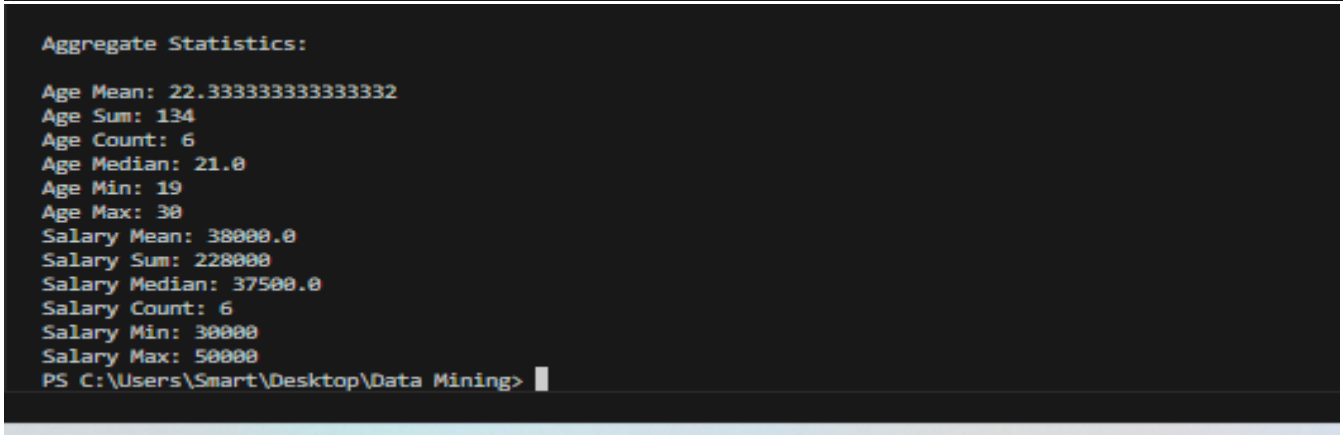
```
import pandas as pd
data = {
"Name": ["Ganesh", "Yogesh", "Mahesh", "Dinesh", "Harshal",
"Sachin"],
"Age": [20, 23, 19, 20, 30, 22],
"Salary": [30000, 40000, 35000, 40000, 50000, 33000],
"Department": ['HR', 'DEV', 'HR', 'CEO', 'DEV', 'CEO'],
"MOB": [4545453445, 6464646464, 7575757575, 5454545454,
8686868686, 5544554455]
}
# Create DataFrame
df = pd.DataFrame(data)
# Print the DataFrame
print(df)
print()
# Filter Employees by Department
hr_em = df[df['Department'] == 'HR']
dev_em = df[df['Department'] == 'DEV']
ceo_em = df[df['Department'] == 'CEO']

# Filter Employees by Salary
high_sal = df[df['Salary'] > 35000]
low_sal = df[df['Salary'] <= 35000]

# Print Filtered Data
print("HR Employees:\n", hr_em, "\n")
print("DEV Employees:\n", dev_em, "\n")
print("CEO Employees:\n", ceo_em, "\n")
print("High Salary Employees:\n", high_sal, "\n")
print("Low Salary Employees:\n", low_sal, "\n")
```

```
# Aggregate Statistics
agg_stats = {
 'Age Mean': df['Age'].mean(),
'Age Sum': df['Age'].sum(),
 'Age Count': df['Age'].count(),
'Age Median': df['Age'].median(),
 'Age Min': df['Age'].min(),
 'Age Max': df['Age'].max(),
 'Salary Mean': df['Salary'].mean(),
 'Salary Sum': df['Salary'].sum(),
 'Salary Median': df['Salary'].median(),
 'Salary Count': df['Salary'].count(),
 'Salary Min': df['Salary'].min(),
 'Salary Max': df['Salary'].max()}
# Print Aggregate Statistics
print("\nAggregate Statistics:\n")
for key, value in agg_stats.items():
print(f"{key}: {value}")
```

Output:



```
PROBLEMS  OUTPUT  TERMINAL  PORTS                                                    Code

PS C:\Users\Smart\Desktop\Data Mining> python -u "c:\Users\Smart\Desktop\Data Mining\tempCodeRunnerFile.py"
      Name  Age  Salary Department        MOB
0   Ganesh   20   30000         HR  4545453445
1   Yogesh   23   40000        DEV  6464646464
2   Mahesh   19   35000         HR  7575757575
3   Dinesh   20   40000        CEO  5454545454
4  Harshal   30   50000        DEV  8686868686
5   Sachin   22   33000        CEO  5544554455

HR Employees:
      Name  Age  Salary Department        MOB
0  Ganesh   20   30000         HR  4545453445
2  Mahesh   19   35000         HR  7575757575

DEV Employees:
      Name  Age  Salary Department        MOB
1   Yogesh   23   40000        DEV  6464646464
4  Harshal   30   50000        DEV  8686868686

CEO Employees:
      Name  Age  Salary Department        MOB
3  Dinesh   20   40000        CEO  5454545454
5  Sachin   22   33000        CEO  5544554455

High Salary Employees:
      Name  Age  Salary Department        MOB
1   Yogesh   23   40000        DEV  6464646464
3   Dinesh   20   40000        CEO  5454545454
4  Harshal   30   50000        DEV  8686868686

Low Salary Employees:
      Name  Age  Salary Department        MOB
0  Ganesh   20   30000         HR  4545453445
2  Mahesh   19   35000         HR  7575757575
5  Sachin   22   33000        CEO  5544554455
```



```
Aggregate Statistics:

Age Mean: 22.333333333333332
Age Sum: 134
Age Count: 6
Age Median: 21.0
Age Min: 19
Age Max: 30
Salary Mean: 38000.0
Salary Sum: 228000
Salary Median: 37500.0
Salary Count: 6
Salary Min: 30000
Salary Max: 50000
PS C:\Users\Smart\Desktop\Data Mining>
```

4. Calculate Total Sales By Month.

```
import pandas as pd
import matplotlib.pyplot as plt
data = {
 "Month": ["January", "February", "March", "April", "May",
"June", "July", "August", "September", "October", "November",
"December"],
"Sales": [5000, 7000, 6500, 8000, 9000, 7500, 6800, 7200,
7800, 8200, 9100, 9500]
}
df = pd.DataFrame(data)
total_sales = df["Sales"].sum()
max_row = df.loc[df["Sales"].idxmax()]
min_row = df.loc[df["Sales"].idxmin()]
mean_sales = df["Sales"].mean()

print("Monthly Sales for 2024:")
print(df)
print("\nSummary:")
print(f"Total Sales in 2024: {total_sales}")
print(f"Highest Sale: {max_row['Month']} with
{max_row['Sales']}")
print(f"Lowest Sale: {min_row['Month']} with
{min_row['Sales']}")
print(f"Mean Sales: {mean_sales:.2f}")
# Filter months with sales >= 8000
filtered_df = df[df["Sales"] >= 8000]
print("\nMonths with Sales >= 8000:")
print(filtered_df)
plt.figure(figsize=(10, 5))
plt.bar(df["Month"], df["Sales"], color="skyblue",
label="Monthly Sales")
```

```
plt.axhline(mean_sales, color="red", linestyle="--",
label=f"Mean Sales ({mean_sales:.2f})")
plt.xlabel("Months")
plt.ylabel("Sales")
plt.title("Monthly Sales for 2024")
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

Output:

```
PS C:\Users\Smart\Desktop\Data Mining> python -u "c:\Users\Smart\Desktop\Data Mining\4.py"
Monthly Sales for 2024:
        Month  Sales
0     January   5000
1    February   7000
2       March   6500
3       April   8000
4         May   9000
5        June   7500
6        July   6800
7      August   7200
8   September   7800
9     October   8200
10   November   9100
11   December   9500

Summary:
Total Sales in 2024: 91600
Highest Sale: December with 9500
Lowest Sale: January with 5000
Mean Sales: 7633.33

Months with Sales >= 8000:
       Month  Sales
3      April   8000
4        May   9000
9    October   8200
10  November   9100
11  December   9500
```