Exp No: 4 **Date:**	**EDA-Data Inspection and Analysis**

Aim

To understand how to view, inspect, and summarize data stored in a DataFrame for initial exploration and analysis.

Problem Statement

Large datasets are hard to understand at first. To make them meaningful, we first view and inspect the data to know its structure, then filter and select only the required rows or columns, and finally calculate basic statistics like mean, median, and standard deviation to summarize the data.

Algorithm

Step 1: Import pandas and load/create the DataFrame.

Step 2: View data using head(), tail(), shape, dtypes, and info().

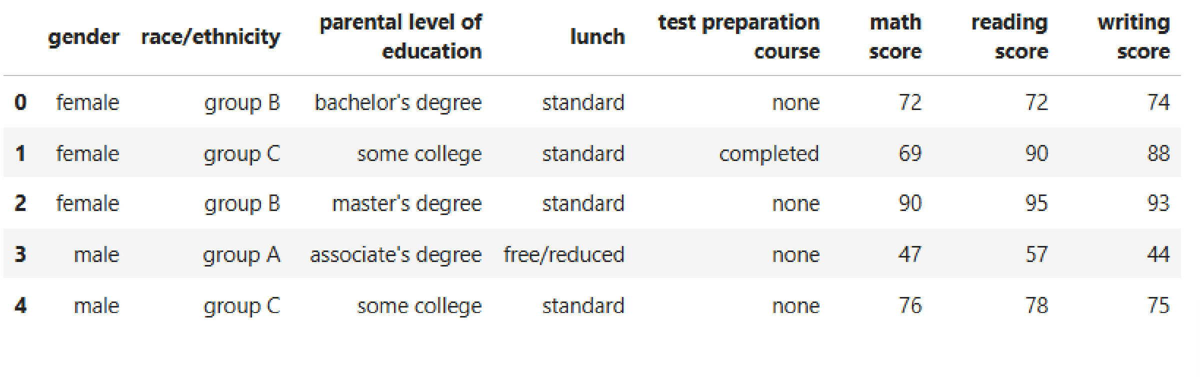
Step 3: Filter rows and select columns using conditions and logical operators.

Step 4: Calculate mean, median, mode, range, variance, and standard deviation.

Step 5: Interpret the results to find patterns and spread of data.

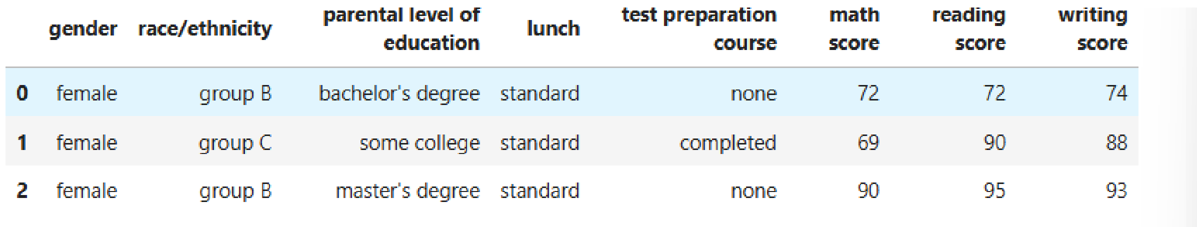
Sample Code

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import matplotlib.pyplot as plt

# Step 1: Load dataset
df = pd.read_csv('StudentsPerformance.csv')
df.head()
```



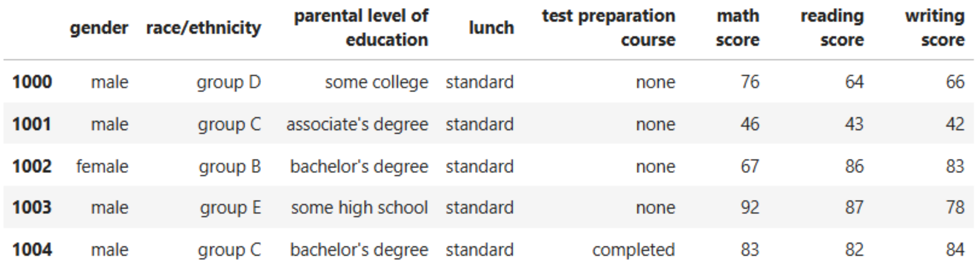
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

df.head(3)



	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93

df.tail()

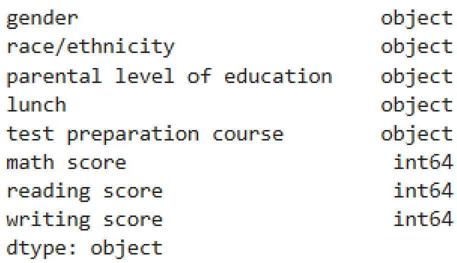


	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1000	male	group D	some college	standard	none	76	64	66
1001	male	group C	associate's degree	standard	none	46	43	42
1002	female	group B	bachelor's degree	standard	none	67	86	83
1003	male	group E	some high school	standard	none	92	87	78
1004	male	group C	bachelor's degree	standard	completed	83	82	84

```
df.shape
(1005, 8)
df.columns.tolist()
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course',
 'math score',
 'reading score',
```

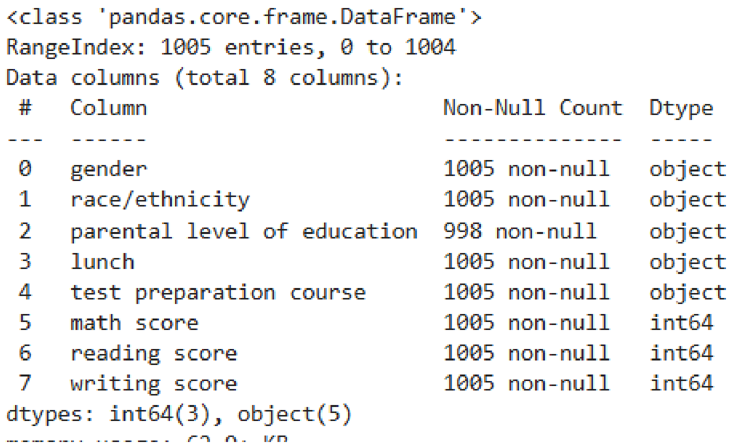
'writing score']

df.dtypes

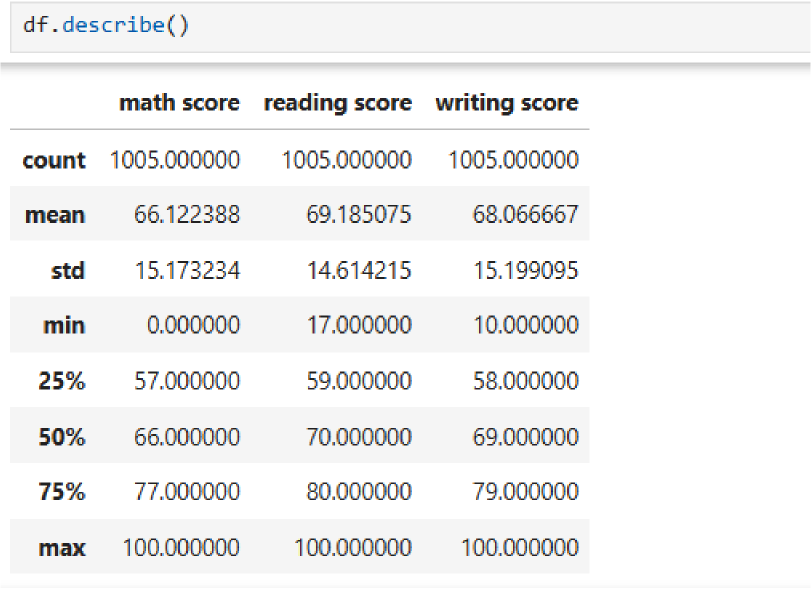


```
gender                         object
race/ethnicity                 object
parental level of education    object
lunch                          object
test preparation course        object
math score                      int64
reading score                   int64
writing score                   int64
dtype: object
```

df.info()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1005 entries, 0 to 1004
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   gender                       1005 non-null   object
 1   race/ethnicity               1005 non-null   object
 2   parental level of education  998 non-null    object
 3   lunch                        1005 non-null   object
 4   test preparation course      1005 non-null   object
 5   math score                   1005 non-null   int64 
 6   reading score                1005 non-null   int64 
 7   writing score                1005 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.9+ KB
```



```
df.describe()
```

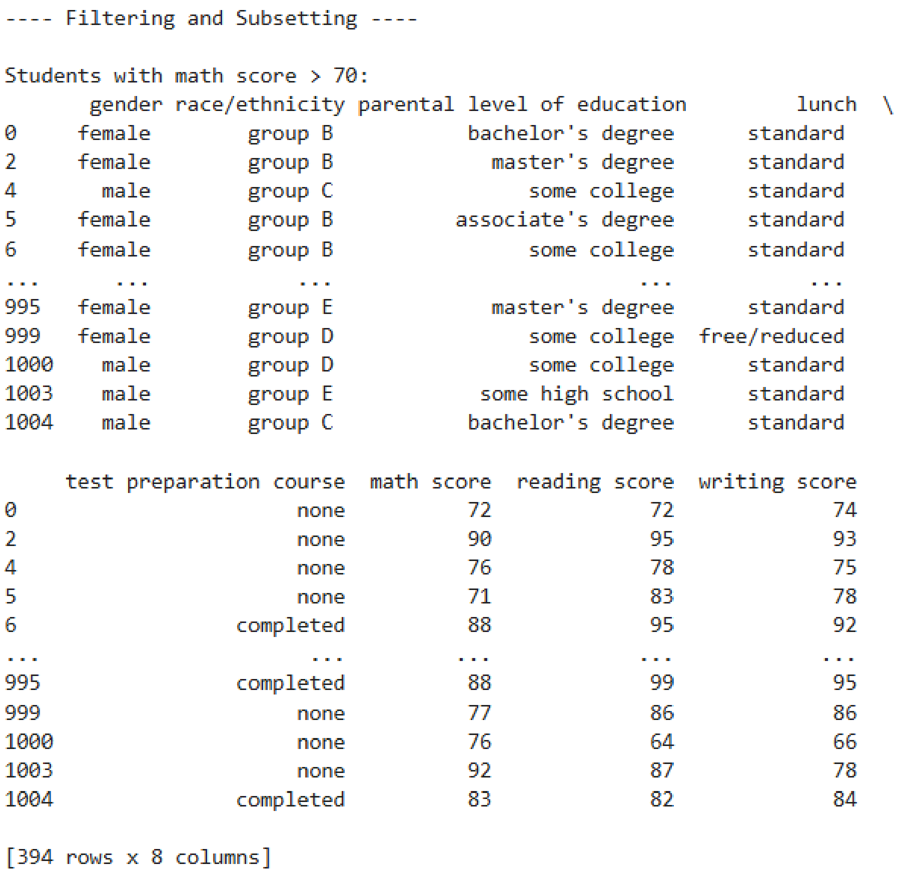
	math score	reading score	writing score
count	1005.000000	1005.000000	1005.000000
mean	66.122388	69.185075	68.066667
std	15.173234	14.614215	15.199095
min	0.000000	17.000000	10.000000
25%	57.000000	59.000000	58.000000
50%	66.000000	70.000000	69.000000
75%	77.000000	80.000000	79.000000
max	100.000000	100.000000	100.000000

## Step 3: Filtering and Subsetting Data

print("\n---- Filtering and Subsetting ----")

# Students with math score > 70

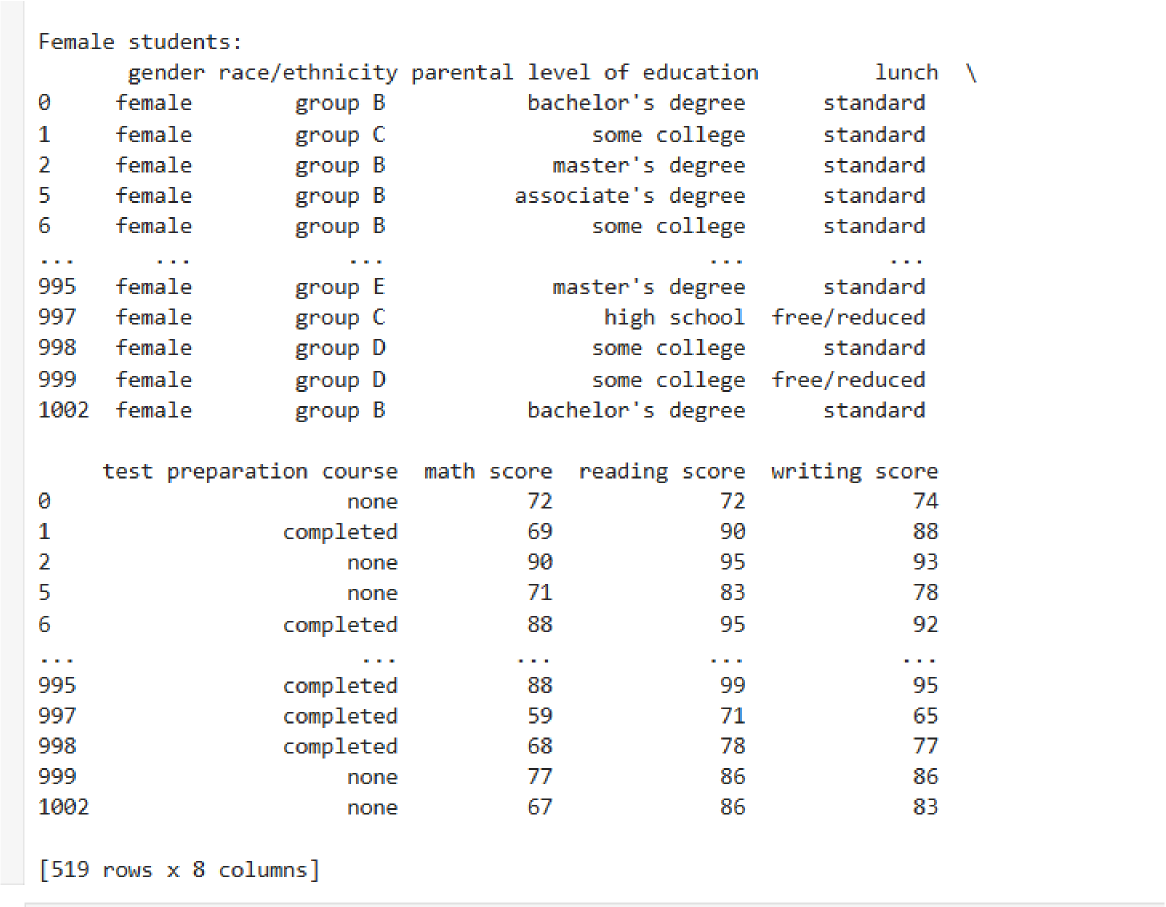
print("\nStudents with math score > 70:\n", df[df["math score"] > 70])



```
---- Filtering and Subsetting ----

Students with math score > 70:
      gender race/ethnicity parental level of education         lunch  \
0     female        group B           bachelor's degree      standard   
2     female        group B             master's degree      standard   
4       male        group C                some college      standard   
5     female        group B          associate's degree      standard   
6     female        group B                some college      standard   
...      ...            ...                         ...           ...   
995   female        group E             master's degree      standard   
999   female        group D                some college  free/reduced   
1000    male        group D                some college      standard   
1003    male        group E            some high school      standard   
1004    male        group C           bachelor's degree      standard   

     test preparation course  math score  reading score  writing score  
0                       none          72             72             74  
2                       none          90             95             93  
4                       none          76             78             75  
5                       none          71             83             78  
6                  completed          88             95             92  
...                      ...         ...            ...            ...  
995                completed          88             99             95  
999                     none          77             86             86  
1000                    none          76             64             66  
1003                    none          92             87             78  
1004               completed          83             82             84  

[394 rows x 8 columns]
```

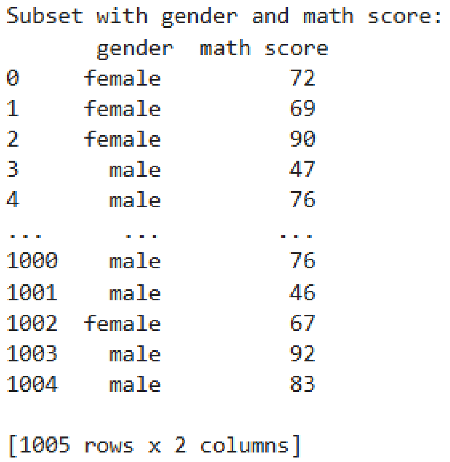
# Female students only

print("\nFemale students:\n", df[df["gender"] == "female"])



```
Female students:
       gender race/ethnicity parental level of education         lunch  \
0     female        group B           bachelor's degree      standard
1     female        group C                some college      standard
2     female        group B             master's degree      standard
5     female        group B          associate's degree      standard
6     female        group B                some college      standard
...      ...            ...                         ...           ...
995   female        group E             master's degree      standard
997   female        group C                 high school  free/reduced
998   female        group D                some college      standard
999   female        group D                some college  free/reduced
1002  female        group B           bachelor's degree      standard

     test preparation course  math score  reading score  writing score
0                       none          72             72             74
1                  completed          69             90             88
2                       none          90             95             93
5                       none          71             83             78
6                  completed          88             95             92
...                      ...         ...            ...            ...
995                completed          88             99             95
997                completed          59             71             65
998                completed          68             78             77
999                     none          77             86             86
1002                    none          67             86             83

[519 rows x 8 columns]
```

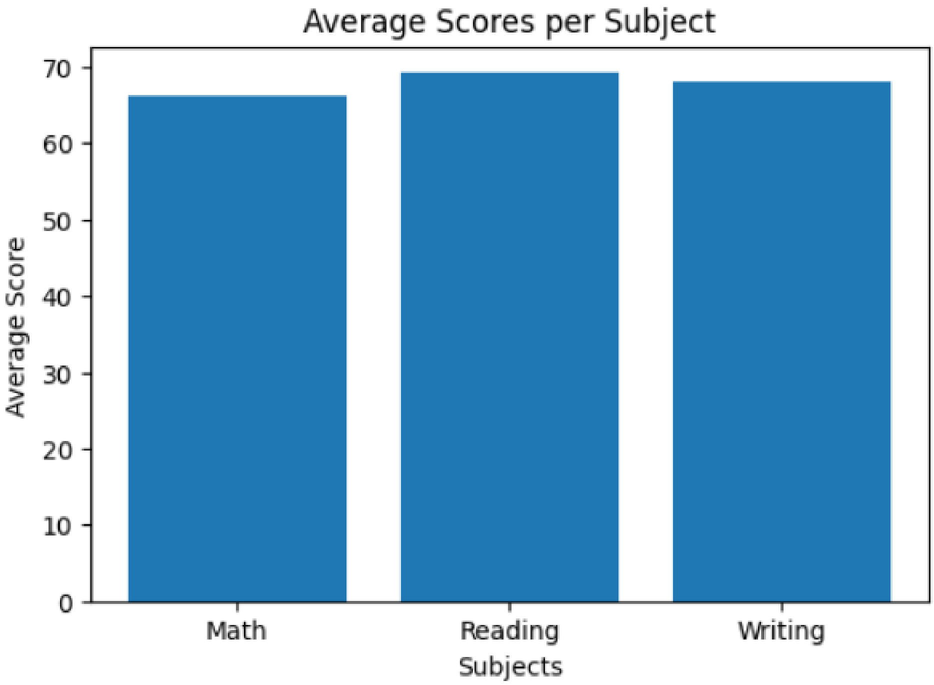
```
# Select only 'gender' and 'math score' columns

print("\nSubset with gender and math score:\n", df[["gender", "math score"]])
```



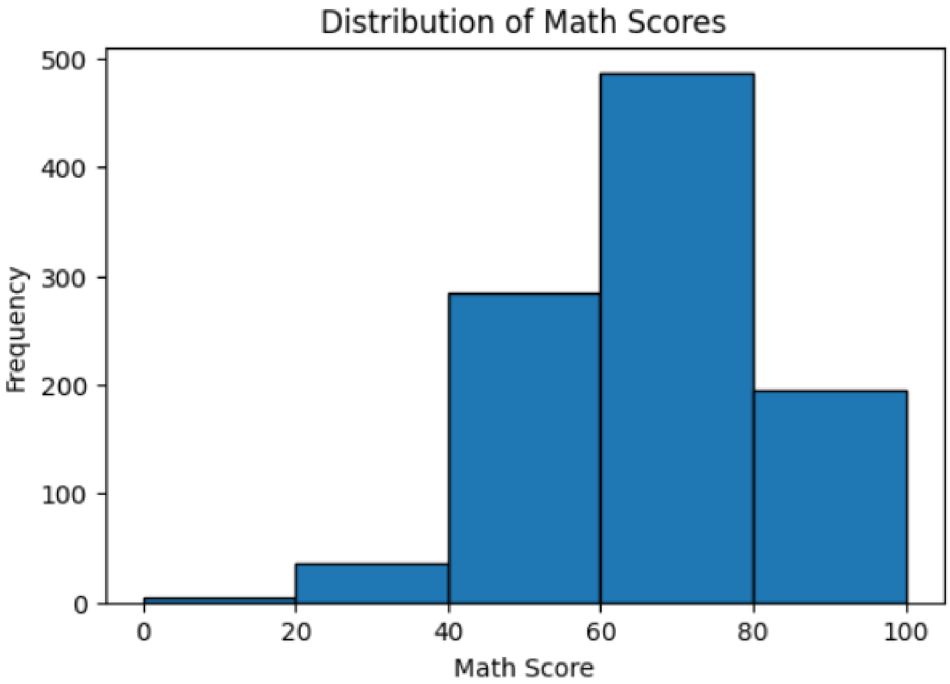
```
Subset with gender and math score:
       gender  math score
0     female          72
1     female          69
2     female          90
3       male          47
4       male          76
...      ...         ...
1000    male          76
1001    male          46
1002  female          67
1003    male          92
1004    male          83

[1005 rows x 2 columns]
```

```
print("\n---- Descriptive Statistics ----")

math_scores = df["math score"]
```

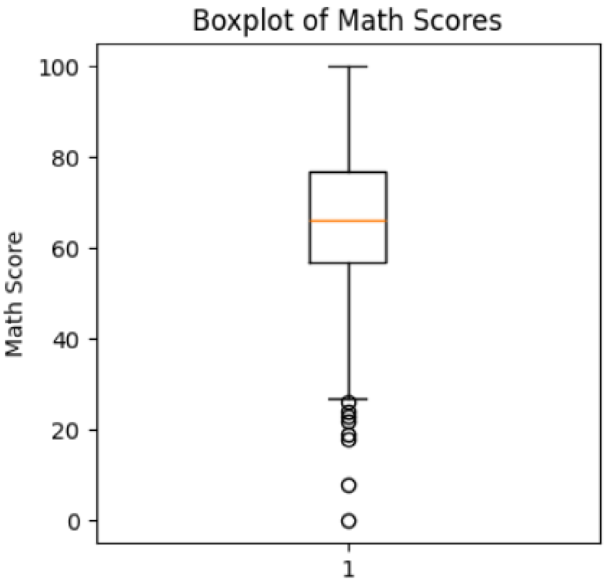
```
mean = math_scores.mean()
median = math_scores.median()
mode = math_scores.mode()[0]  # mode() returns a Series

_range = math_scores.max() - math_scores.min()
variance = math_scores.var()
std_dev = math_scores.std()

print(f"\nMean (Math Score): {mean}")
print(f"Median (Math Score): {median}")
print(f"Mode (Math Score): {mode}")
print(f"Range (Math Score): {_range}")
print(f"Variance (Math Score): {variance}")
print(f"Standard Deviation (Math Score): {std_dev}")
---- Descriptive Statistics ----

Mean (Math Score): 66.12238805970149
Median (Math Score): 66.0
Mode (Math Score): 65
Range (Math Score): 100
Variance (Math Score): 230.2270381161917
Standard Deviation (Math Score): 15.173234266832885

print("\n---- Visualization ----")

# 1. Bar chart: Average scores per subject
avg_scores = {
    "Math": df["math score"].mean(),
    "Reading": df["reading score"].mean(),
    "Writing": df["writing score"].mean()
}

plt.figure(figsize=(6, 4))
plt.bar(avg_scores.keys(), avg_scores.values())
plt.title("Average Scores per Subject")
plt.ylabel("Average Score")
plt.xlabel("Subjects")
plt.show()
```



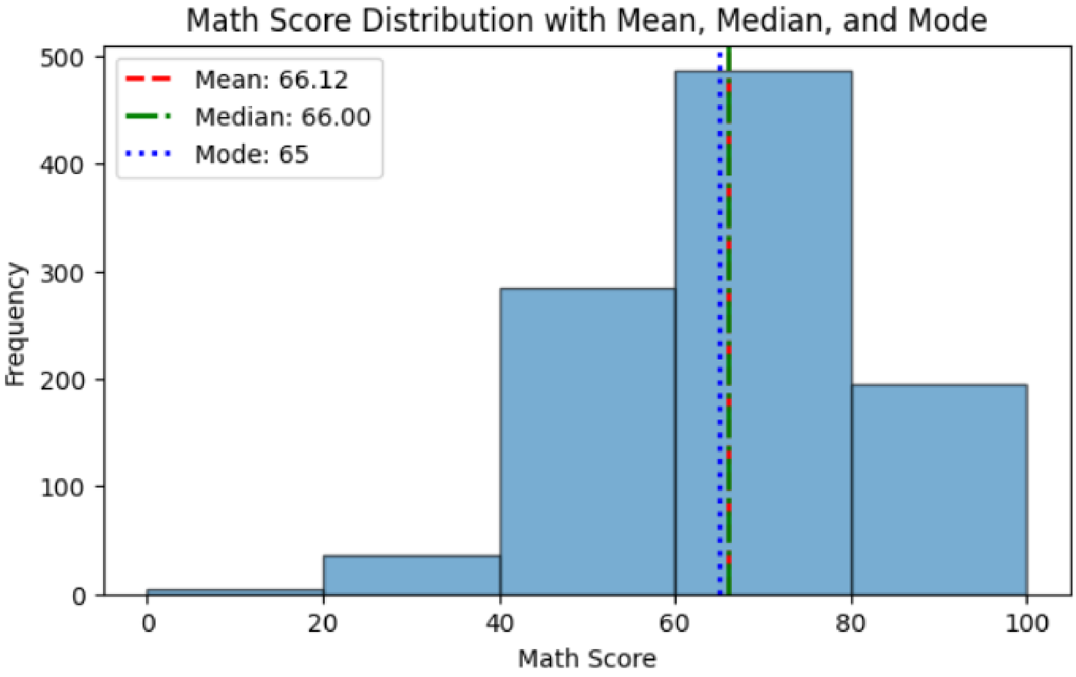
```
# 2. Histogram: Distribution of math scores
plt.figure(figsize=(6, 4))
plt.hist(df["math score"], bins=5, edgecolor="black")
plt.title("Distribution of Math Scores")
plt.xlabel("Math Score")
plt.ylabel("Frequency")
plt.show()
```



```
# 3. Boxplot: Spread of math scores
plt.figure(figsize=(4, 4))
plt.boxplot(df["math score"])
plt.title("Boxplot of Math Scores")
plt.ylabel("Math Score")
plt.show()
```



```
import matplotlib.pyplot as plt
# Plot Histogram with Mean, Median, and Mode Lines
plt.figure(figsize=(7, 4))
plt.hist(df["math score"], bins=5, edgecolor="black", alpha=0.6)
plt.axvline(mean, color='red', linestyle='--', linewidth=2, label=f"Mean: {mean:.2f}")
plt.axvline(median, color='green', linestyle='-.', linewidth=2, label=f"Median: {median:.2f}")
plt.axvline(mode, color='blue', linestyle=':', linewidth=2, label=f"Mode: {mode}")
plt.title("Math Score Distribution with Mean, Median, and Mode")
plt.xlabel("Math Score")
plt.ylabel("Frequency")
plt.legend()
plt.show()
```



Result

Thus, the Exploratory Data Analysis (EDA) was successfully performed by viewing, filtering, and summarizing the dataset. Data visualization was done using bar charts, histograms, and boxplots in Matplotlib to better understand the distribution and trends in the students' performance.