

```
In [3]: # 1. Load libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# 2. Load the CSV file
df = pd.read_csv("Week-5-Student-Scores 1.csv")
```

```
In [4]: # 3. Basic info
print("Shape:", df.shape)
print("\nColumn Data Types:")
print(df.dtypes)
```

Shape: (121, 10)

Column Data Types:

Student_ID	object
Gender	object
Subject	object
Study_Hours_per_Week	float64
Attendance_Rate	float64
Previous_Score	float64
Quiz1	float64
Quiz2	float64
Quiz3	float64
Final_Score	float64
dtype:	object

```
In [5]: # 4. Missing values per column
print("\nMissing Values per Column:")
print(df.isna().sum())
```

Missing Values per Column:

Student_ID	0
Gender	0
Subject	0
Study_Hours_per_Week	5
Attendance_Rate	5
Previous_Score	0
Quiz1	0
Quiz2	5
Quiz3	0
Final_Score	0
dtype:	int64

```
In [6]: # 5. Number of duplicates
print("\nNumber of duplicate rows:", df.duplicated().sum())
```

Number of duplicate rows: 1

```
In [7]: # --- Diagnostics Table ---
diagnostics = pd.DataFrame({
    "Column": df.columns,
    "Data_Type": df.dtypes.values,
    "Missing_Values": df.isna().sum().values,
})
diagnostics.loc[len(diagnostics.index)] = ['TOTAL', '', diagnostics['Missing_Values'].sum()]
diagnostics
```

Out [7]:

	Column	Data_Type	Missing_Values
0	Student_ID	object	0
1	Gender	object	0
2	Subject	object	0
3	Study_Hours_per_Week	float64	5
4	Attendance_Rate	float64	5
5	Previous_Score	float64	0
6	Quiz1	float64	0
7	Quiz2	float64	5
8	Quiz3	float64	0
9	Final_Score	float64	0
10	TOTAL		15

In [8]:

```
before = len(df)
df = df.dropna()
after = len(df)
print(f"Dropped {before - after} rows with missing values.")
```

Dropped 14 rows with missing values.

In [9]:

```
# --- Diagnostics Table ---
diagnostics = pd.DataFrame({
    "Column": df.columns,
    "Data_Type": df.dtypes.values,
    "Missing_Values": df.isna().sum().values,
})
diagnostics.loc[len(diagnostics.index)] = ['TOTAL', '', diagnostics['Missing_Values'].sum()]
diagnostics
```

Out [9]:

	Column	Data_Type	Missing_Values
0	Student_ID	object	0
1	Gender	object	0
2	Subject	object	0
3	Study_Hours_per_Week	float64	0
4	Attendance_Rate	float64	0
5	Previous_Score	float64	0
6	Quiz1	float64	0
7	Quiz2	float64	0
8	Quiz3	float64	0
9	Final_Score	float64	0
10	TOTAL		0

In [10]:

```
# Select only numeric columns
numeric_df = df.select_dtypes(include='number')

# Compute summary stats
desc = numeric_df.describe(percentiles=[0.25, 0.5, 0.75]).T

# Print summary stats for clarity
```

```

desc = desc.rename(columns={
    '25%': '25th_percentile',
    '50%': 'median',
    '75%': '75th_percentile'
})

# Keep only the requested columns
desc = desc[['mean', 'median', 'std', 'min', '25th_percentile', '75th_percentile', 'max']]
desc

```

```

Out[10]:

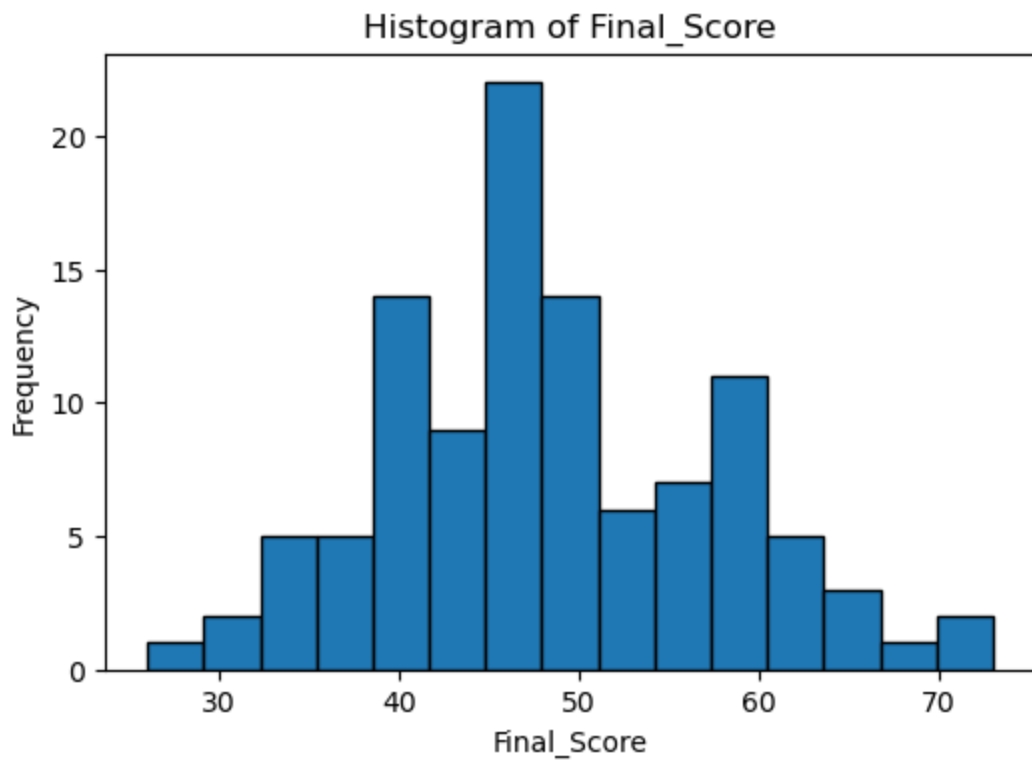
```

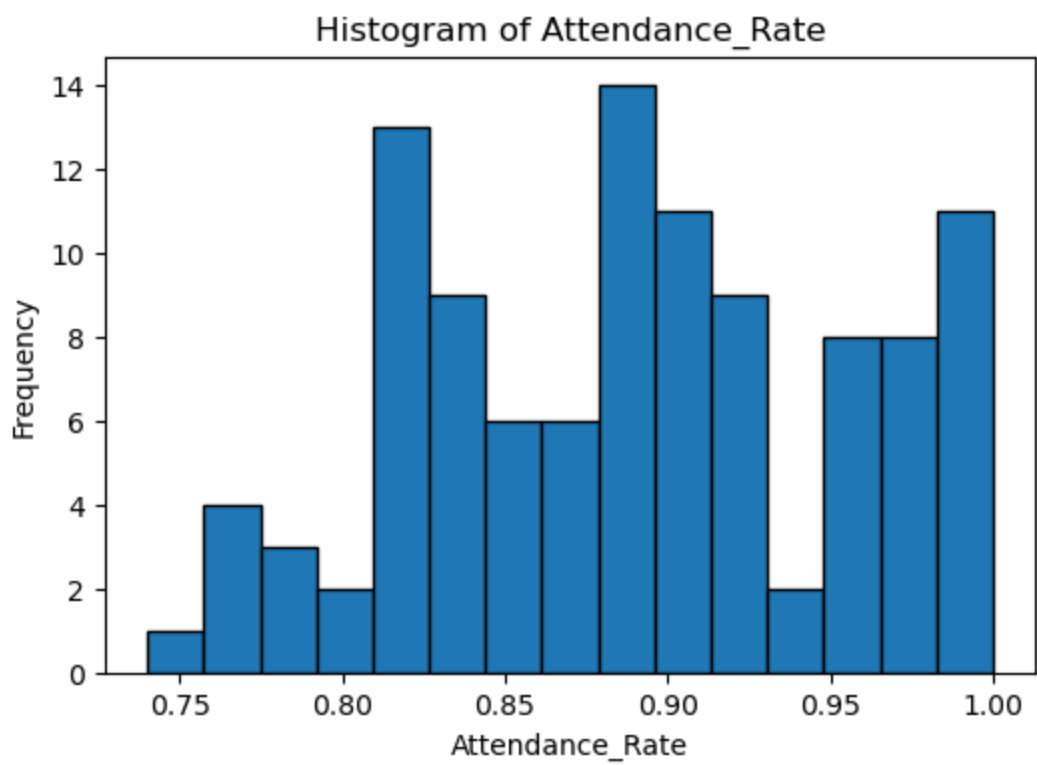
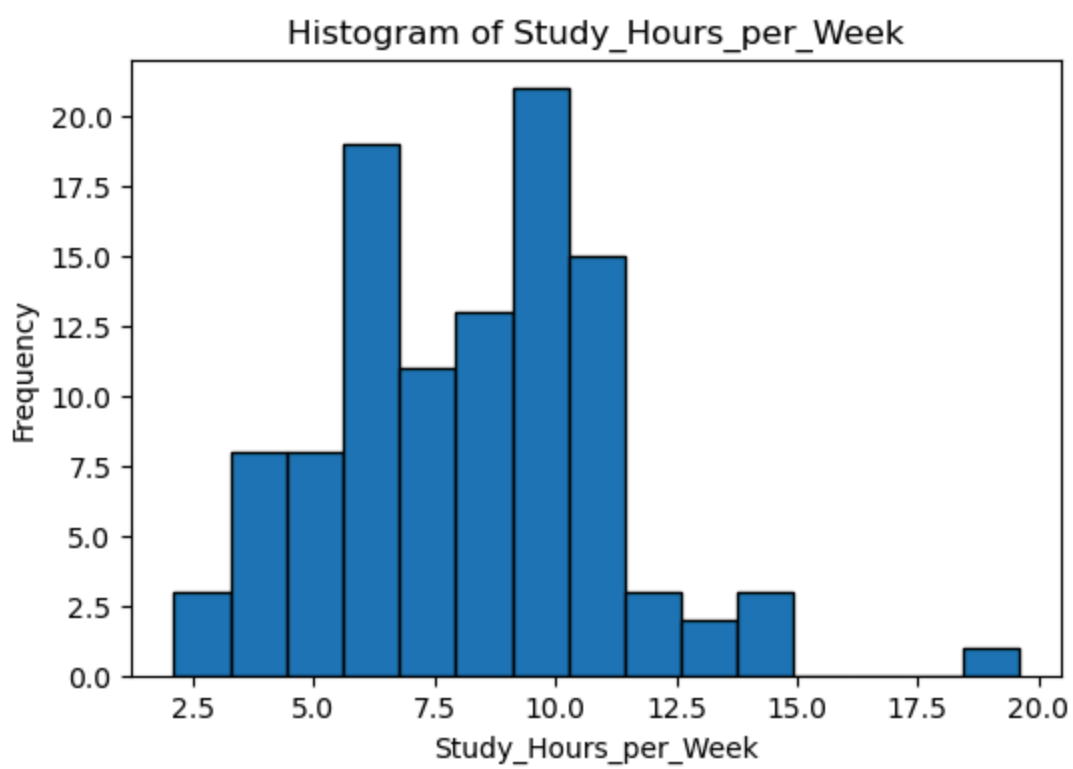
	mean	median	std	min	25th_percentile	75th_percentile	max
Study_Hours_per_Week	8.209346	8.20	2.895785	2.10	6.05	10.050	19.6
Attendance_Rate	0.889907	0.89	0.067844	0.74	0.83	0.945	1.0
Previous_Score	60.289720	60.00	11.858449	34.00	52.00	68.500	98.0
Quiz1	49.112150	49.00	12.109818	18.00	41.00	55.500	84.0
Quiz2	49.700935	50.00	13.304996	18.00	41.50	58.000	81.0
Quiz3	51.897196	51.00	10.893078	29.00	44.00	60.000	77.0
Final_Score	48.233645	47.00	9.392469	26.00	41.50	55.000	73.0

```

In [11]: for col in ['Final_Score', 'Study_Hours_per_Week', 'Attendance_Rate']:
plt.figure(figsize=(6,4))
plt.hist(df[col], bins=15, edgecolor='black')
plt.title(f'Histogram of {col}')
plt.xlabel(col)
plt.ylabel('Frequency')
plt.show()

```





In []: