ACTIVITY - Student Performance Analysis Using Regression

CODING:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from scipy import stats


def univariate_analysis(df, column):
    freq = df[column].value_counts()
    mean = df[column].mean()
    median = df[column].median()
    mode = df[column].mode()[0]
    std = df[column].std()
    return freq, mean, median, mode, std


# ------------------------------
# 1. UPLOAD & LOAD
# ------------------------------
from google.colab import files
uploaded = files.upload()
import pandas as pd
df = pd.read_csv("StudentsPerformance.csv")
print(df.shape)
df.head()
```

```
# ----------------------------

# 2. UNIVARIATE ANALYSIS

# ----------------------------
def univariate(df, column):
    freq = df[column].value_counts()
    mean = df[column].mean()
    median = df[column].median()
    mode = df[column].mode()[0]
    std = df[column].std()
    return freq, mean, median, mode, std

freq, mean, median, mode, std = univariate(df, "math score")

print("Frequency:\n", freq)
print("Mean:", mean)
print("Median:", median)
print("Mode:", mode)
print("Std Dev:", std)

# ----------------------------

# 3. LINEAR REGRESSION MODEL

# ----------------------------

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

X = df[["reading score"]]   # independent
y = df["math score"]        # dependent

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)
mse = mean_squared_error(y_test, y_pred)

print("Linear Regression MSE:", mse)
```

```
# ------------------------------
# 4. MULTIPLE REGRESSION MODEL
# ------------------------------

features = ["reading score", "writing score"]

X = df[features]
y = df["math score"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

multi = LinearRegression()
multi.fit(X_train, y_train)

y_pred = multi.predict(X_test)

mse_multi = mean_squared_error(y_test, y_pred)
print("Multiple Regression MSE:", mse_multi)



# 5. VISUALIZATIONS
# ------------------------------

plt.figure(figsize=(8,5))

plt.hist(df["math score"], bins=15, color="skyblue", edgecolor="black")

plt.title("Histogram of Math Scores")

plt.xlabel("Math Score")

plt.ylabel("Number of Students")

plt.show()
```

OUTPUT:

StudentsPerformance.csv(text/csv) - 72036 bytes, last modified: 12/11/2025 - 100% done
Saving StudentsPerformance.csv to StudentsPerformance (1).csv

In[1]:

```
(1000, 8)
```

	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

Next steps: Generate code with df | New interactive sheet

In[2]:

Frequency:

math score	
65	36
62	35
69	32
59	32
73	27
24	1
26	1
19	1
23	1
8	1

Name: count, Length: 81, dtype: int64

Mean: 66.089

Median: 66.0

Mode: 65

Std Dev: 15.163080096009468

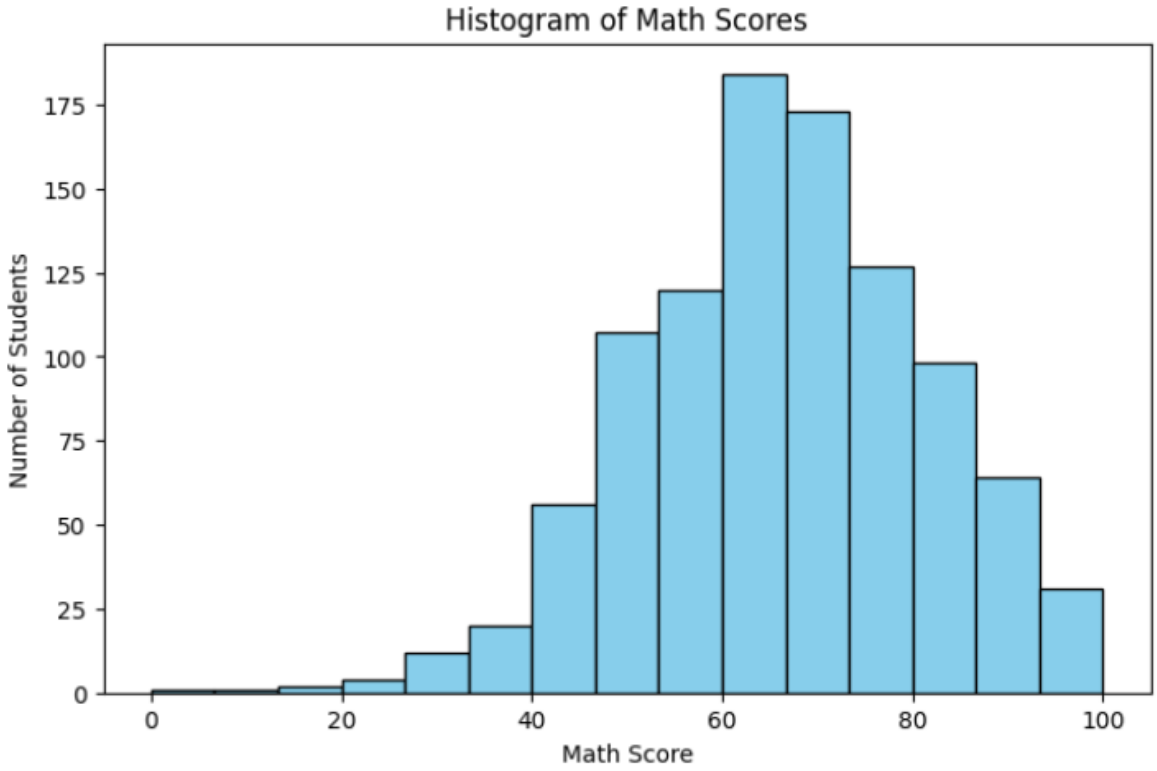
In[3]:

Linear Regression MSE: 77.75953982761706

In[4]:

Multiple Regression MSE: 77.24297821278955

PLOT:



SUMMARY:

This project focuses on analyzing student academic performance using statistical techniques and regression modeling. The dataset contains information such as math score, reading score, writing score, gender, parental education level, and other related factors. The project begins with data cleaning and formatting, followed by univariate analysis to understand how each feature is distributed. Statistical measures like mean, median, mode, and standard deviation were calculated to identify common patterns in student scores and variability across subjects.

Visual analysis, such as histograms, helped reveal that most students scored in the mid-range for mathematics, while extremely high and low scores were less common. Bivariate analysis showed strong relationships between reading, writing, and math scores, indicating that performance in one subject is closely related to performance in others.

A simple linear regression model was built to predict math score using reading score. The model demonstrated that reading ability is a strong indicator of mathematical performance. A multiple regression model was then constructed using reading and writing scores together, and this model provided better accuracy due to the combined influence of multiple academic skills.

Overall, this project demonstrates how statistical analysis and regression techniques can be applied to educational datasets to uncover learning patterns, evaluate student performance trends, and build predictive models. It shows that student performance is influenced by multiple academic factors and that machine learning can help understand these relationships more deeply.

PLOT EXPLANATION:

This plot represents the distribution of **math scores** of students from your dataset. The histogram shows how many students fall into each score range. Most students appear to score between **60 and 80 marks**, indicating that the dataset has a high concentration of average to above-average performers.

Lower scores (below 40) occur less frequently, meaning very few students struggle severely in mathematics. Likewise, very high scores (above 90) are also rare, showing that only a small number of students excel at the highest level.

The shape of the histogram helps us understand the performance pattern:

- A **cluster in the middle** suggests a normal distribution.
- Fewer students at the extremes indicate **lower variability**.

Overall, the plot provides a clear visual understanding of how math performance is spread across the student group, helping identify common score ranges and performance trends.