Week 10 MATLAB and Python Statistics Activities

Contents

Time required: 120 minutes

Descriptive Statistical Functions in MATLAB and Python

This tutorial will guide you through using various descriptive statistical functions in MATLAB and Python. It covers basic functions like finding the mean, standard deviation, and median, as well as advanced functions like calculating percentiles and performing moving statistics.

Basic Statistics Functions

Python provides numerous functions for computing descriptive statistics. Here are some commonly used ones:

- **sum(x)**: Sums all elements in vector
- **min(x)**: Finds the minimum element in the vector `x`.
- **max(x)**: Finds the maximum element in the vector `x`.

- **mean(x)**: Calculates the average of all elements in `x`.
- **median(x)**: Finds the middle value of `x` when sorted in ascending order.
- **mode(x)**: Find the value that occurs the most.
- **std(x)**: Computes the standard deviation of `x`.
- **var(x)**: Calculates the variance of `x`, which is the square of the standard deviation.

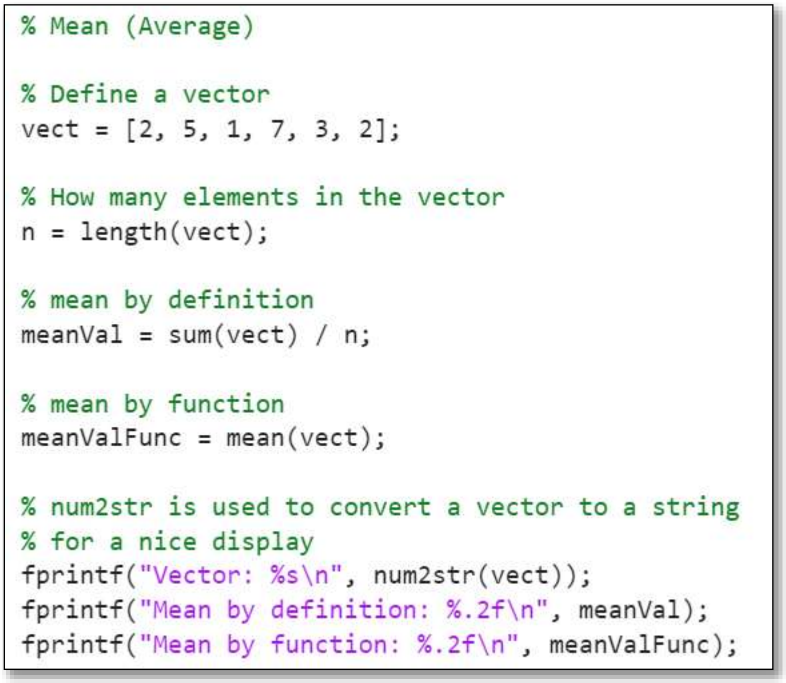
Tutorial 1: Mean

Mean, also known as the average, is a measure of central tendency that represents the typical value of a set of numbers.

- It is calculated by summing up all the values in the dataset and dividing the sum by the total number of values.
- The mean is sensitive to extreme values, also known as outliers, as they can significantly influence its value.
- It is widely used in statistics, mathematics, and various fields of science and engineering to describe the central tendency of data.

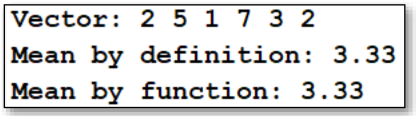
MATLAB

fprint works fine with single variables. To print a vector nicely with fprint, we use the num2str function to convert the vector into a string for display.



```
% Mean (Average)

% Define a vector
vect = [2, 5, 1, 7, 3, 2];

% How many elements in the vector
n = length(vect);

% mean by definition
meanVal = sum(vect) / n;

% mean by function
meanValFunc = mean(vect);

% num2str is used to convert a vector to a string
% for a nice display
fprintf("Vector: %s\n", num2str(vect));
fprintf("Mean by definition: %.2f\n", meanVal);
fprintf("Mean by function: %.2f\n", meanValFunc);
```

Example run:



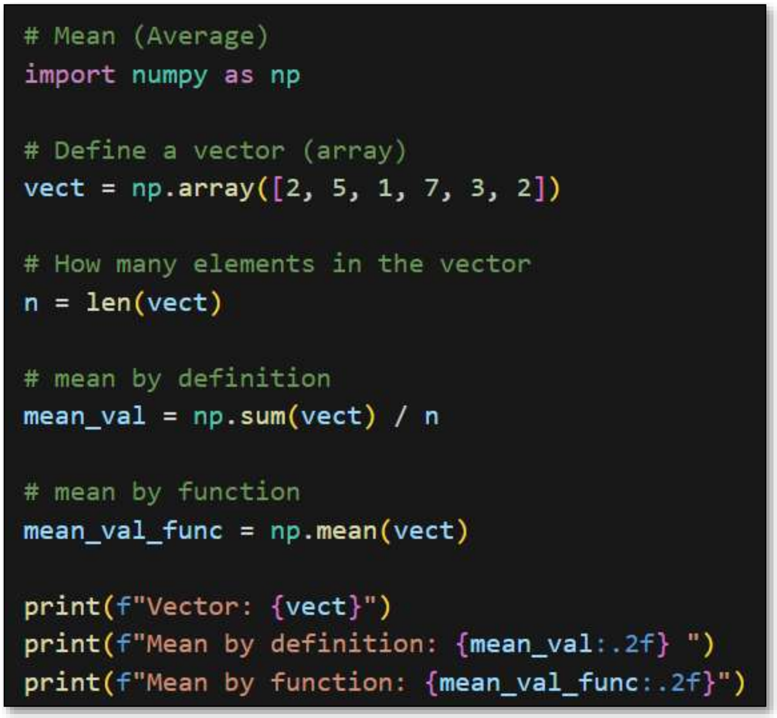
```
Vector: 2 5 1 7 3 2
Mean by definition: 3.33
Mean by function: 3.33
```

Python

NumPy is a Python library for numerical computing that provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. It is widely used in scientific computing, data analysis, and machine learning due to its speed, versatility, and ease of use. NumPy's ndarray (N-dimensional array) data structure enables efficient computation and manipulation of large datasets, making it an essential tool for numerical tasks in Python.

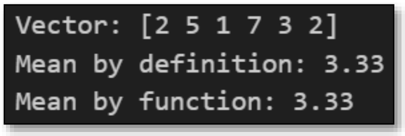
A NumPy array is the same as a MATLAB vector.

Add the following code to your Google Colab Notebook.



```python
# Mean (Average)
import numpy as np

# Define a vector (array)
vect = np.array([2, 5, 1, 7, 3, 2])

# How many elements in the vector
n = len(vect)

# mean by definition
mean_val = np.sum(vect) / n

# mean by function
mean_val_func = np.mean(vect)

print(f"Vector: {vect}")
print(f"Mean by definition: {mean_val:.2f} ")
print(f"Mean by function: {mean_val_func:.2f}")
```

Example run:

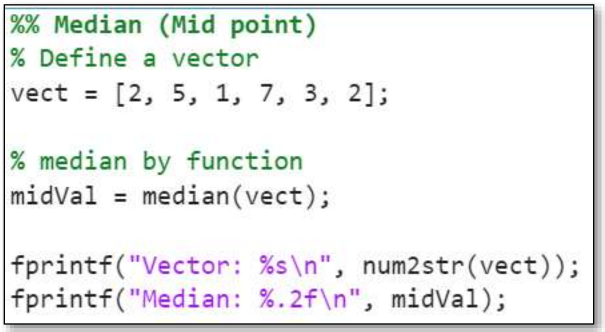


```
Vector: [2 5 1 7 3 2]
Mean by definition: 3.33
Mean by function: 3.33
```

Tutorial 2: Median

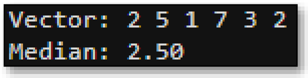
- The **median** is a statistical measure that represents the middle value of a dataset when it is ordered from smallest to largest.
- It is robust to outliers and extreme values, making it a useful measure of central tendency, particularly in skewed distributions.
- To compute the median, the dataset is arranged in ascending order, and the middle value is selected. If the dataset has an odd number of observations, the median is the middle value. If the dataset has an even number of observations, the median is the average of the two middle values.
- Unlike the mean, which can be influenced by extreme values, the median is less affected by outliers, making it a better representation of the central tendency in skewed distributions or datasets with extreme values.

MATLAB



```
%% Median (Mid point)
% Define a vector
vect = [2, 5, 1, 7, 3, 2];

% median by function
midVal = median(vect);

fprintf("Vector: %s\n", num2str(vect));
fprintf("Median: %.2f\n", midVal);
```

Example run:

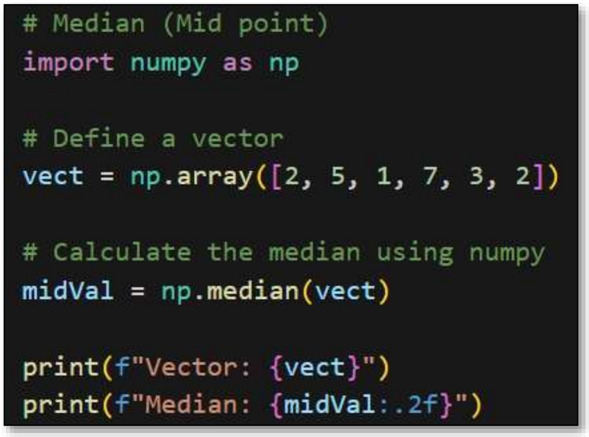


```
Vector: 2 5 1 7 3 2
Median: 2.50
```

Python

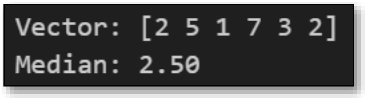
To calculate the median with Python, let’s again turn to NumPy.

Add the following code to your Google Colab Notebook.



```
# Median (Mid point)
import numpy as np

# Define a vector
vect = np.array([2, 5, 1, 7, 3, 2])

# Calculate the median using numpy
midVal = np.median(vect)

print(f"Vector: {vect}")
print(f"Median: {midVal:.2f}")
```

Example run:



```
Vector: [2 5 1 7 3 2]
Median: 2.50
```

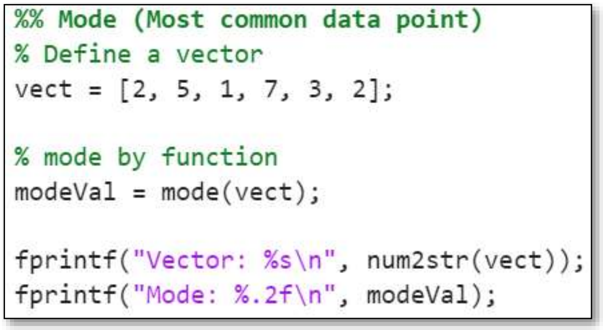
Tutorial 3: Mode

Mode: In statistics, the mode represents the value or values that occur most frequently in a dataset.

- It is one of the measures of central tendency, along with mean and median, providing insight into the central or typical value of a dataset.
- The mode can be particularly useful for categorical or discrete data, where it represents the most common category or value.
- Unlike mean and median, which require numerical data, the mode can be applied to any type of data, including categorical and ordinal data.
- A dataset may have one mode (unimodal), multiple modes (multimodal), or no mode if all values occur with equal frequency.
- Computing the mode involves identifying the value with the highest frequency or frequencies in the dataset.

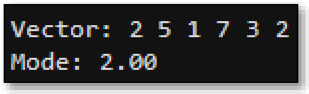
The mode of a set of data is the most common data point.

MATLAB



```
%% Mode (Most common data point)
% Define a vector
vect = [2, 5, 1, 7, 3, 2];

% mode by function
modeVal = mode(vect);

fprintf("Vector: %s\n", num2str(vect));
fprintf("Mode: %.2f\n", modeVal);
```

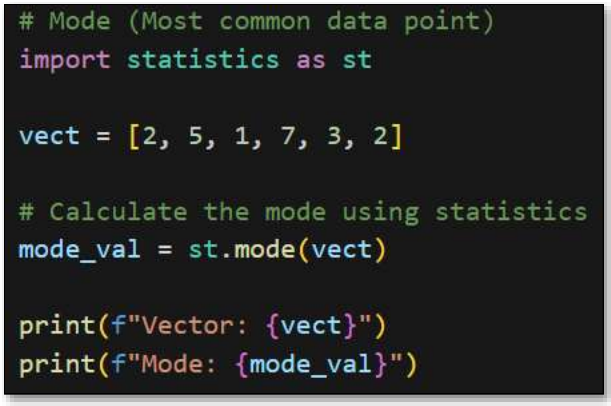
Example run:



```
Vector: 2 5 1 7 3 2
Mode: 2.00
```

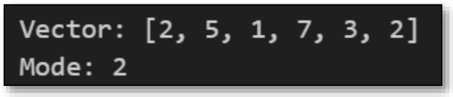
Python

NumPy does not have a mode function. We will use Python’s built in statistics library, statistics.



```
# Mode (Most common data point)
import statistics as st

vect = [2, 5, 1, 7, 3, 2]

# Calculate the mode using statistics
mode_val = st.mode(vect)

print(f"Vector: {vect}")
print(f"Mode: {mode_val}")
```

Example run:

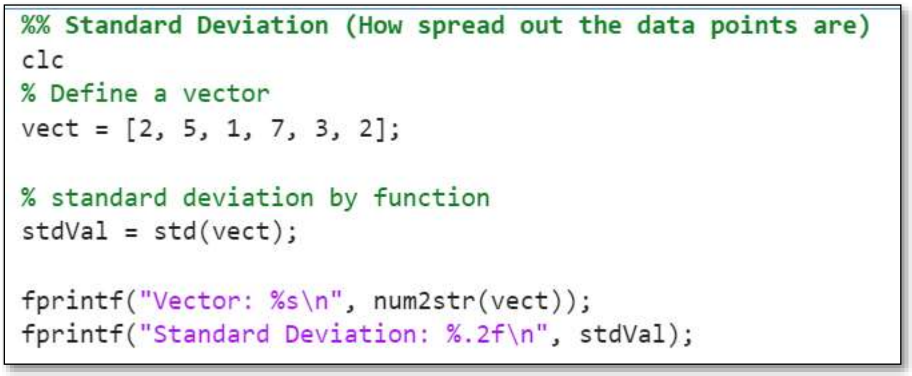


```
Vector: [2, 5, 1, 7, 3, 2]
Mode: 2
```

Tutorial 4: Standard Deviation

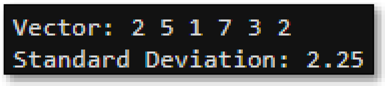
- **Standard Deviation**: Standard deviation is a measure of the dispersion or spread of a set of values from their mean (average) value.
- It quantifies the amount of variation or dispersion in a dataset, indicating how much individual data points deviate from the mean.
- A low standard deviation suggests that the data points tend to be close to the mean, while a high standard deviation indicates that the data points are spread out over a wider range of values.

MATLAB



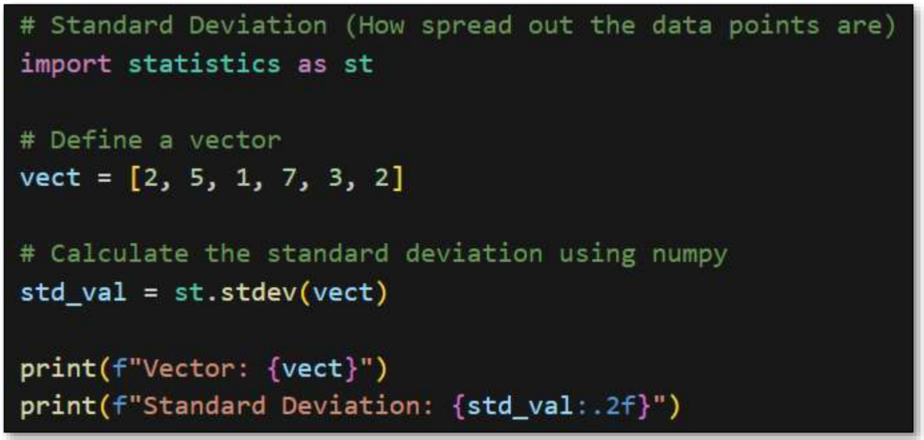
```
%% Standard Deviation (How spread out the data points are)
clc
% Define a vector
vect = [2, 5, 1, 7, 3, 2];

% standard deviation by function
stdVal = std(vect);

fprintf("Vector: %s\n", num2str(vect));
fprintf("Standard Deviation: %.2f\n", stdVal);
```

Example run:



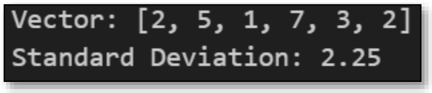
```
Vector: 2 5 1 7 3 2
Standard Deviation: 2.25
```

Python



```python
# Standard Deviation (How spread out the data points are)
import statistics as st

# Define a vector
vect = [2, 5, 1, 7, 3, 2]

# Calculate the standard deviation using numpy
std_val = st.stdev(vect)

print(f"Vector: {vect}")
print(f"Standard Deviation: {std_val:.2f}")
```

Example run:



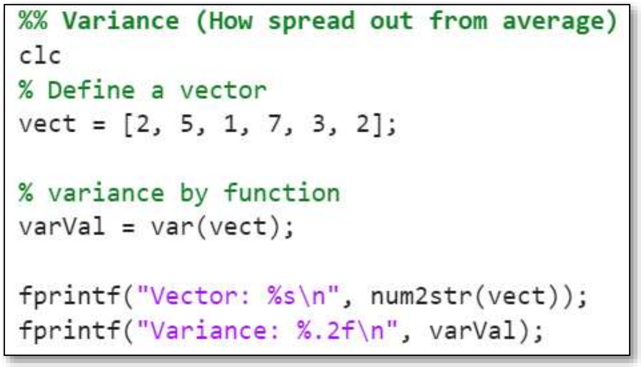
```
Vector: [2, 5, 1, 7, 3, 2]
Standard Deviation: 2.25
```

Tutorial 5: Variance

Variance is a statistical measure that represents the degree of spread or dispersion in a set of data points. It tells us how much the values in a dataset differ from the mean (average) of the dataset.

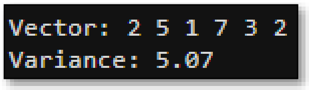
Variance helps us understand the variability within a dataset.

MATLAB



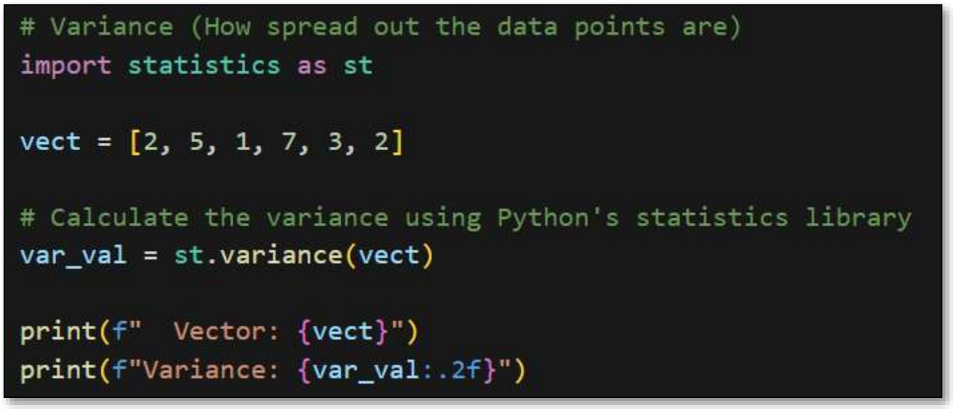
```matlab
%% Variance (How spread out from average)
clc
% Define a vector
vect = [2, 5, 1, 7, 3, 2];

% variance by function
varVal = var(vect);

fprintf("Vector: %s\n", num2str(vect));
fprintf("Variance: %.2f\n", varVal);
```

Example run:



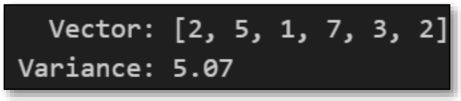
```
Vector: 2 5 1 7 3 2
Variance: 5.07
```

Python



```python
# Variance (How spread out the data points are)
import statistics as st

vect = [2, 5, 1, 7, 3, 2]

# Calculate the variance using Python's statistics library
var_val = st.variance(vect)

print(f"  Vector: {vect}")
print(f"Variance: {var_val:.2f}")
```

Example run:

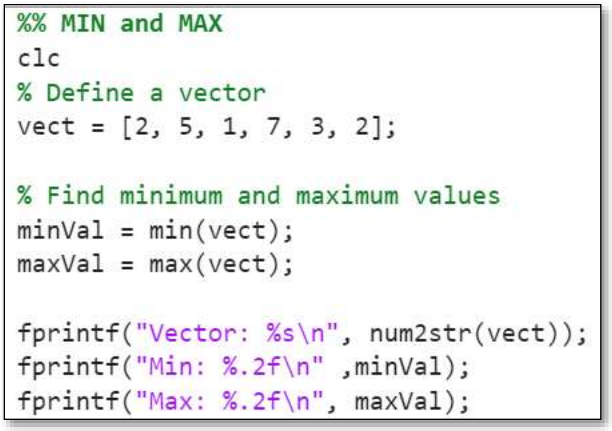


```
  Vector: [2, 5, 1, 7, 3, 2]
Variance: 5.07
```

Tutorial 6: MIN and MAX

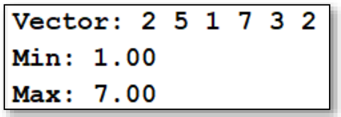
Pretty simple, the largest and smallest numbers of a dataset.

MATLAB



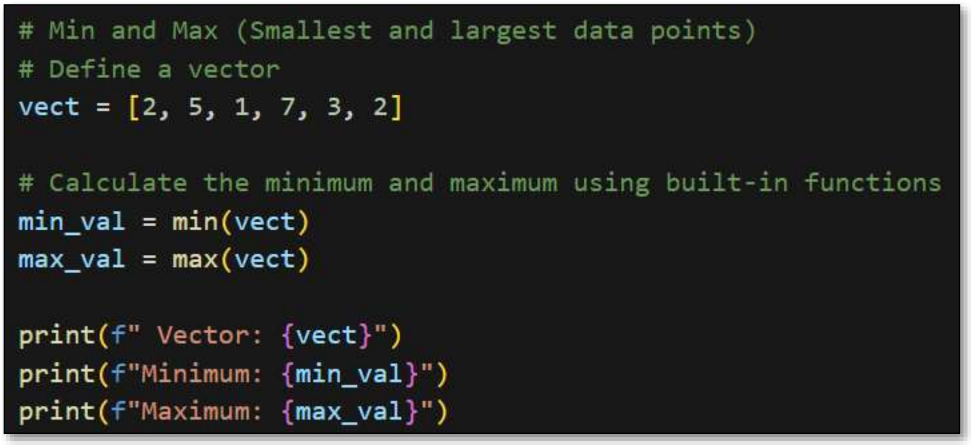
```matlab
%% MIN and MAX
clc
% Define a vector
vect = [2, 5, 1, 7, 3, 2];

% Find minimum and maximum values
minVal = min(vect);
maxVal = max(vect);

fprintf("Vector: %s\n", num2str(vect));
fprintf("Min: %.2f\n" ,minVal);
fprintf("Max: %.2f\n", maxVal);
```

Example run:



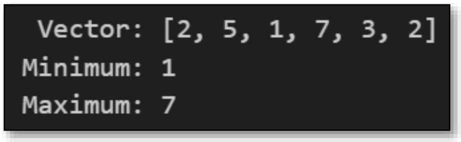
```
Vector: 2 5 1 7 3 2
Min: 1.00
Max: 7.00
```

Python



```python
# Min and Max (Smallest and largest data points)
# Define a vector
vect = [2, 5, 1, 7, 3, 2]

# Calculate the minimum and maximum using built-in functions
min_val = min(vect)
max_val = max(vect)

print(f" Vector: {vect}")
print(f"Minimum: {min_val}")
print(f"Maximum: {max_val}")
```

Example run:



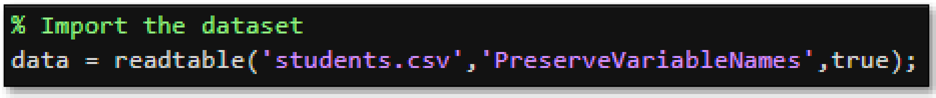
```
 Vector: [2, 5, 1, 7, 3, 2]
Minimum: 1
Maximum: 7
```

Assignment 1: Statistical Analysis of a Dataset

Objective: Apply statistical techniques using MATLAB and Python to analyze and interpret data.

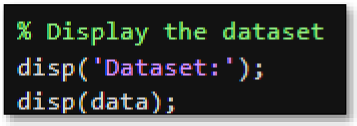
MATLAB

1. Upload students.csv into MATLAB online.
2. Import **students.csv** into MATLAB.



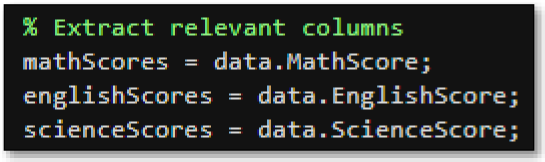
```matlab
% Import the dataset
data = readtable('students.csv','PreserveVariableNames',true);
```

3. Display the dataset to see the columns and rows.



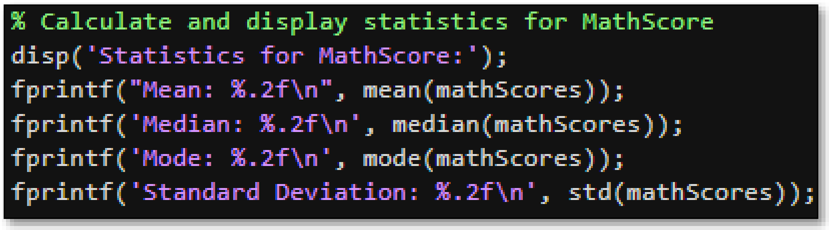
```
% Display the dataset
disp('Dataset:');
disp(data);
```

4. Extract the relevant columns.



```
% Extract relevant columns
mathScores = data.MathScore;
englishScores = data.EnglishScore;
scienceScores = data.ScienceScore;
```

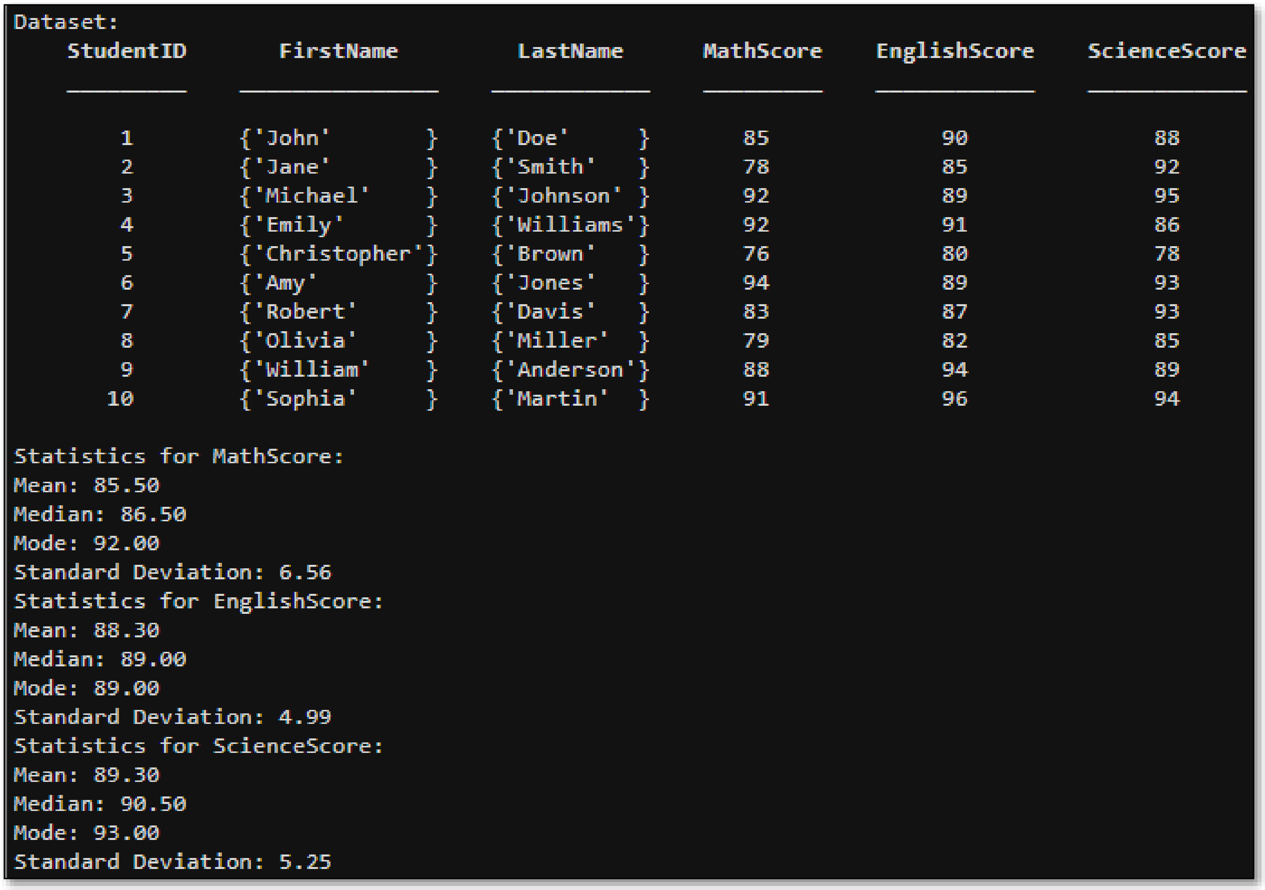
5. Calculate and display basic descriptive statistics (mean, median, mode, and standard deviation) for MathScore, EnglishScore, and ScienceScore. MathScore is shown here as an example of how to complete the others.



```
% Calculate and display statistics for MathScore
disp('Statistics for MathScore:');
fprintf("Mean: %.2f\n", mean(mathScores));
fprintf('Median: %.2f\n', median(mathScores));
fprintf('Mode: %.2f\n', mode(mathScores));
fprintf('Standard Deviation: %.2f\n', std(mathScores));
```

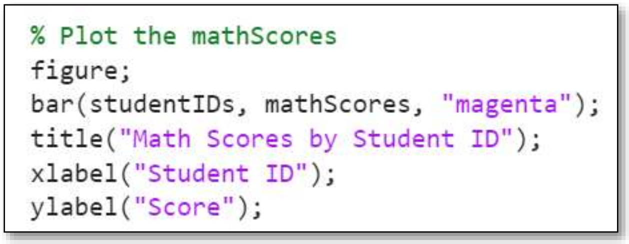
6. Calculate and display English Scores, and Science Scores.

Example run:

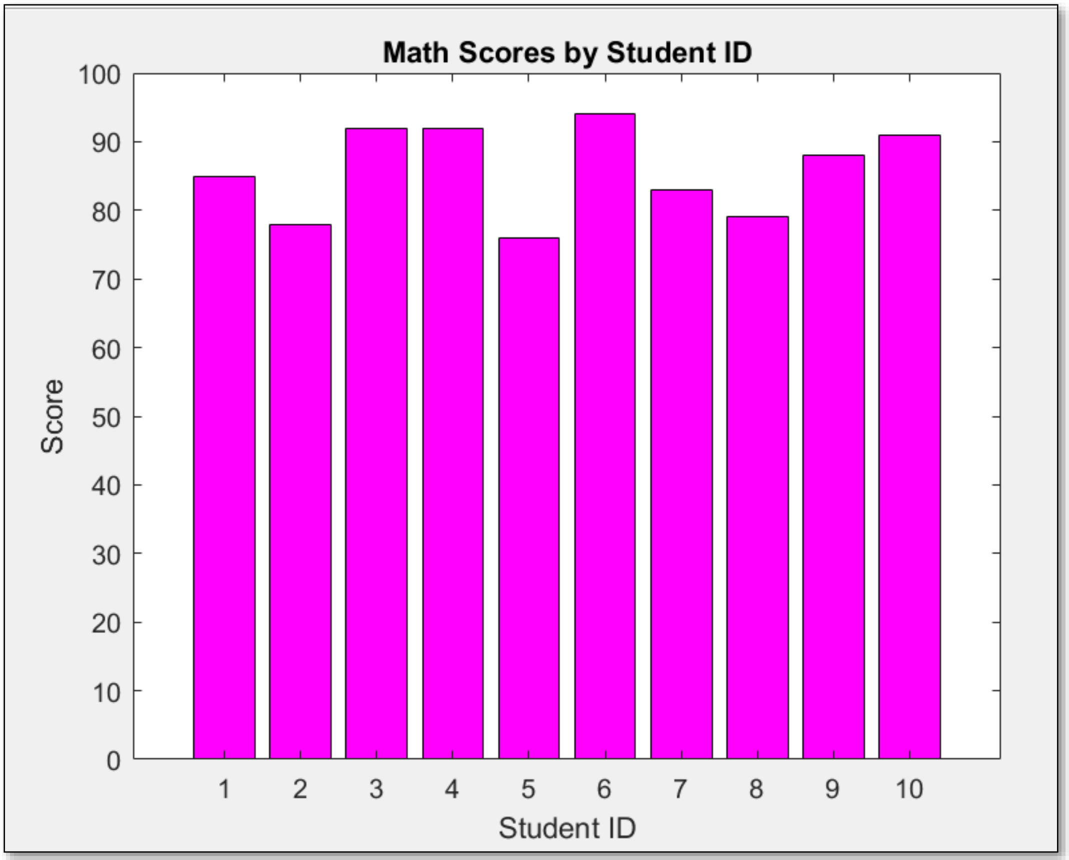


```
Dataset:
    StudentID        FirstName          LastName      MathScore    EnglishScore    ScienceScore
    _________    _________________    ____________    _________    ____________    ____________

        1        {'John'         }    {'Doe'     }       85             90              88
        2        {'Jane'         }    {'Smith'   }       78             85              92
        3        {'Michael'      }    {'Johnson' }       92             89              95
        4        {'Emily'        }    {'Williams'}       92             91              86
        5        {'Christopher'}      {'Brown'   }       76             80              78
        6        {'Amy'          }    {'Jones'   }       94             89              93
        7        {'Robert'       }    {'Davis'   }       83             87              93
        8        {'Olivia'       }    {'Miller'  }       79             82              85
        9        {'William'      }    {'Anderson'}       88             94              89
       10        {'Sophia'       }    {'Martin'  }       91             96              94

Statistics for MathScore:
Mean: 85.50
Median: 86.50
Mode: 92.00
Standard Deviation: 6.56
Statistics for EnglishScore:
Mean: 88.30
Median: 89.00
Mode: 89.00
Standard Deviation: 4.99
Statistics for ScienceScore:
Mean: 89.30
Median: 90.50
Mode: 93.00
Standard Deviation: 5.25
```

7. Plot the MathScores per student.



```
% Plot the mathScores
figure;
bar(studentIDs, mathScores, "magenta");
title("Math Scores by Student ID");
xlabel("Student ID");
ylabel("Score");
```

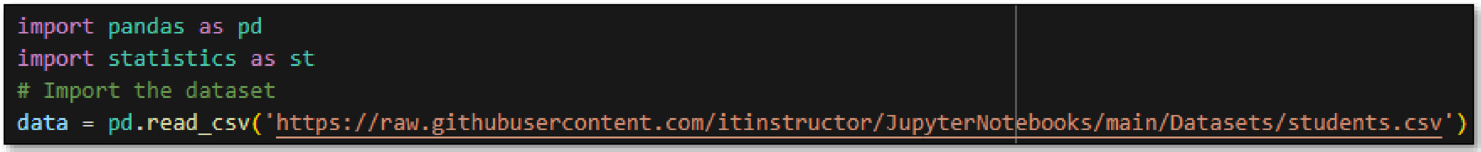


8. Plot the English and Science scores

Python

1. Import **students.csv** into Python from GitHub using the following link and code.
2. Right click the link below → Copy Link → Paste the link into the code as shown.

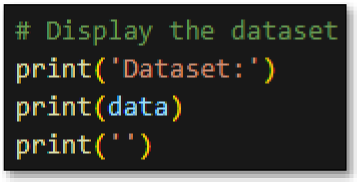
https://raw.githubusercontent.com/itinstructor/JupyterNotebooks/main/Datasets/students.csv



```
import pandas as pd
import statistics as st
# Import the dataset
data = pd.read_csv('https://raw.githubusercontent.com/itinstructor/JupyterNotebooks/main/Datasets/students.csv')
```

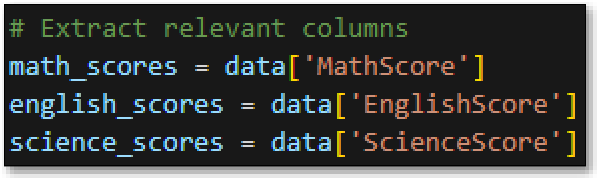
NOTE: The pandas library allows us to work with data tables like MATLAB does.

3. Display the dataset to see the columns and rows.



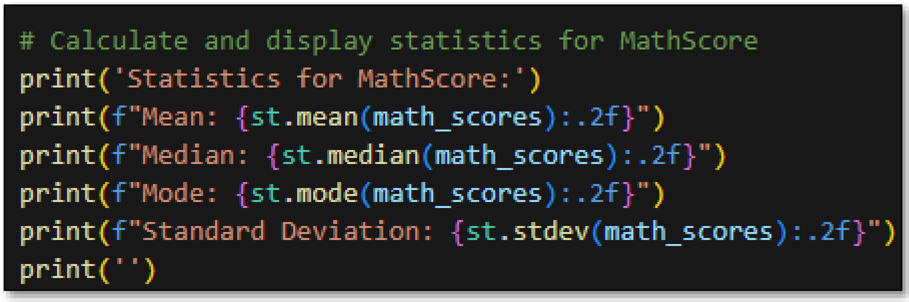
```python
# Display the dataset
print('Dataset:')
print(data)
print('')
```

4. Extract the relevant columns.



```python
# Extract relevant columns
math_scores = data['MathScore']
english_scores = data['EnglishScore']
science_scores = data['ScienceScore']
```

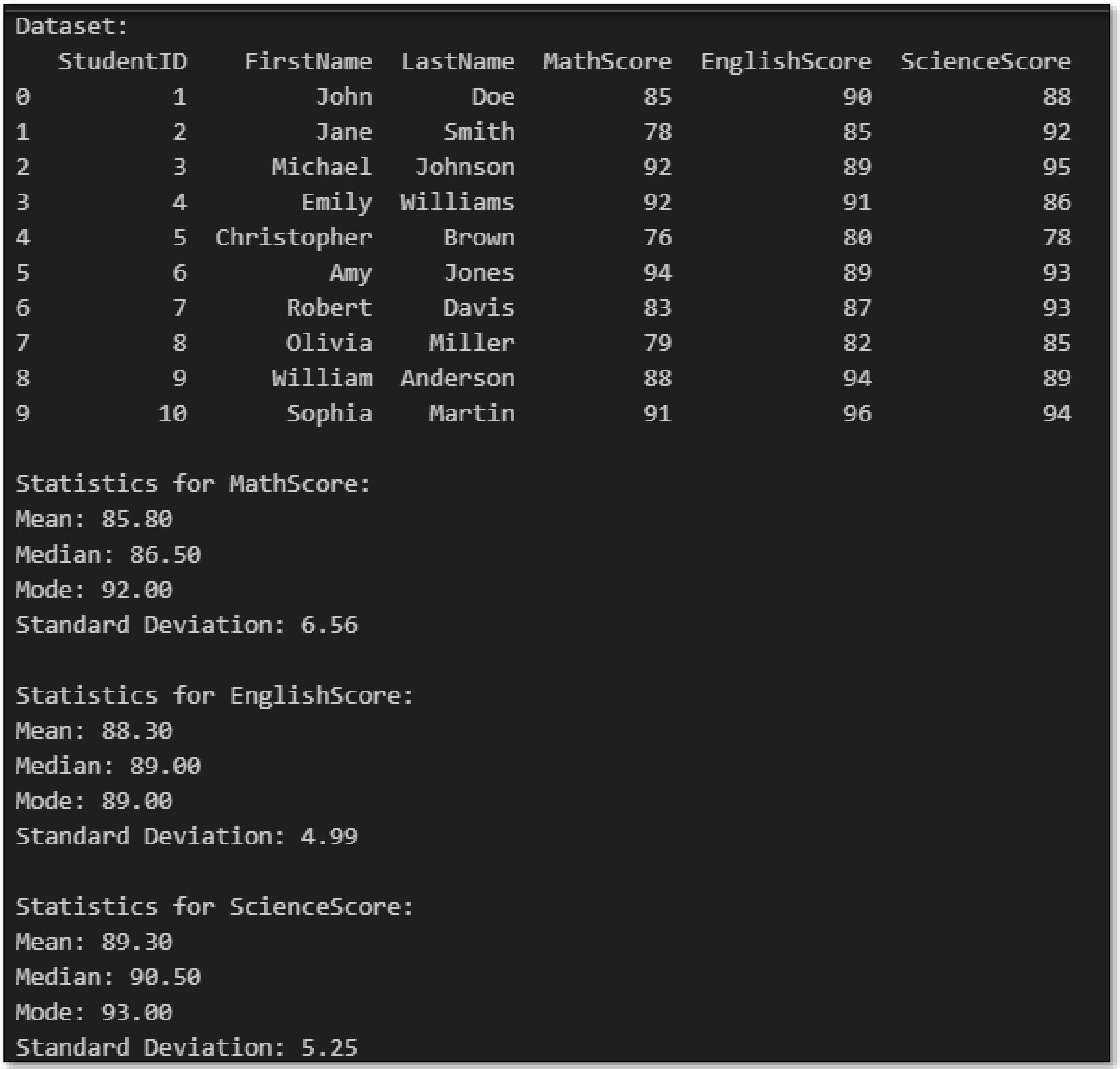
5. Calculate and display basic descriptive statistics (mean, median, mode, and standard deviation) for MathScore, EnglishScore, and ScienceScore. MathScore is shown here as an example of how to complete the others.



```python
# Calculate and display statistics for MathScore
print('Statistics for MathScore:')
print(f"Mean: {st.mean(math_scores):.2f}")
print(f"Median: {st.median(math_scores):.2f}")
print(f"Mode: {st.mode(math_scores):.2f}")
print(f"Standard Deviation: {st.stdev(math_scores):.2f}")
print('')
```

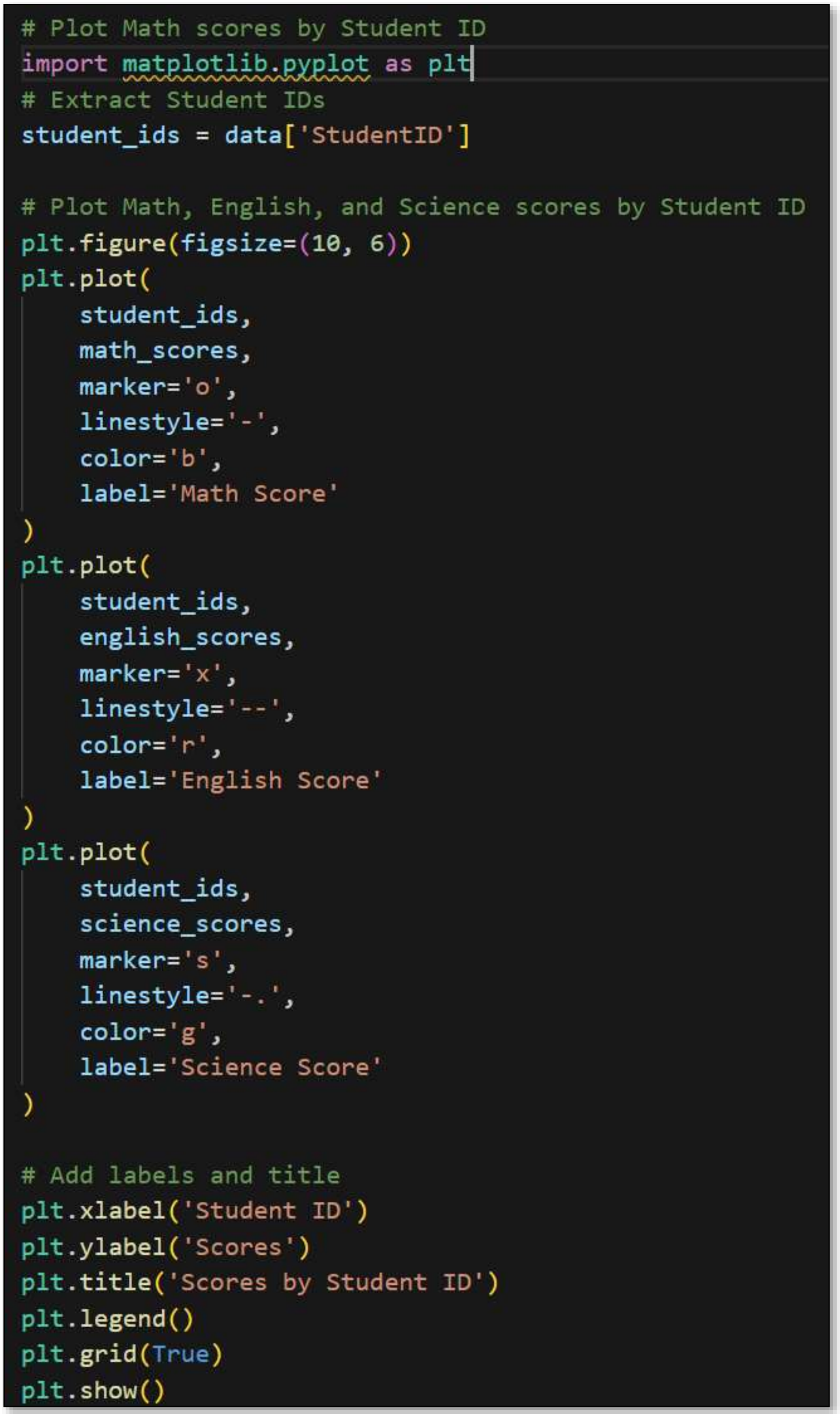
6. To finish the program on your own, calculate and display English Scores, and Science Scores.

Example run:

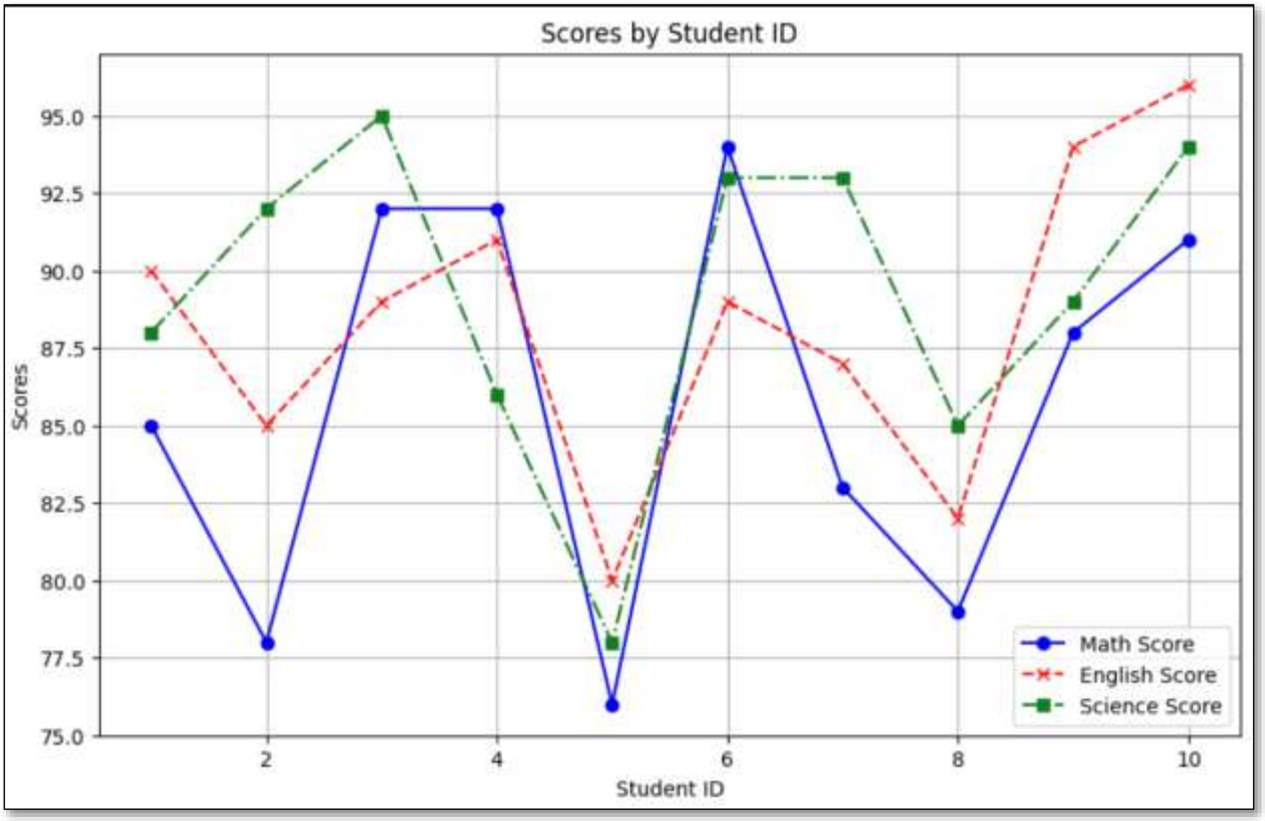


```
Dataset:
   StudentID    FirstName  LastName  MathScore  EnglishScore  ScienceScore
0          1         John       Doe         85            90            88
1          2         Jane     Smith         78            85            92
2          3      Michael   Johnson         92            89            95
3          4        Emily  Williams         92            91            86
4          5  Christopher     Brown         76            80            78
5          6          Amy     Jones         94            89            93
6          7       Robert     Davis         83            87            93
7          8       Olivia    Miller         79            82            85
8          9      William  Anderson         88            94            89
9         10       Sophia    Martin         91            96            94

Statistics for MathScore:
Mean: 85.80
Median: 86.50
Mode: 92.00
Standard Deviation: 6.56

Statistics for EnglishScore:
Mean: 88.30
Median: 89.00
Mode: 89.00
Standard Deviation: 4.99

Statistics for ScienceScore:
Mean: 89.30
Median: 90.50
Mode: 93.00
Standard Deviation: 5.25
```

9. Plot the Math Scores per student.



```python
# Plot Math scores by Student ID
import matplotlib.pyplot as plt
# Extract Student IDs
student_ids = data['StudentID']

# Plot Math, English, and Science scores by Student ID
plt.figure(figsize=(10, 6))
plt.plot(
    student_ids,
    math_scores,
    marker='o',
    linestyle='-',
    color='b',
    label='Math Score'
)
plt.plot(
    student_ids,
    english_scores,
    marker='x',
    linestyle='--',
    color='r',
    label='English Score'
)
plt.plot(
    student_ids,
    science_scores,
    marker='s',
    linestyle='-.',
    color='g',
    label='Science Score'
)

# Add labels and title
plt.xlabel('Student ID')
plt.ylabel('Scores')
plt.title('Scores by Student ID')
plt.legend()
plt.grid(True)
plt.show()
```



Assignment Submission

1. In Google Colab → Click the Share button in the upper right hand side.
    a. Change General Access → Anyone with the link → Click Copy link.
2. Attach your MATLAB file and a screenshot of the Command Window showing the successful execution of each script.
3. Attach all to the assignment in Blackboard.