MATLAB and Python Statistics Activities

Contents

Time required: 90 minutes

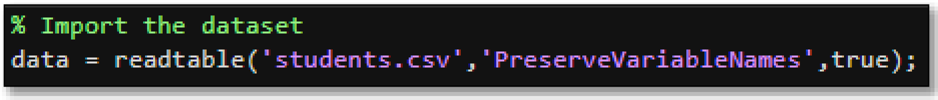
1. Save Python code in a Google Colab Notebook.
2. Save MATLAB code in a MATLAB file.

Assignment 1: Dataset Plots of Student Data

Objective: Create plots using MATLAB Python to analyze and interpret data.

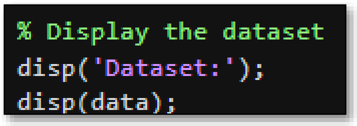
MATLAB

1. Import **students.csv** into MATLAB.



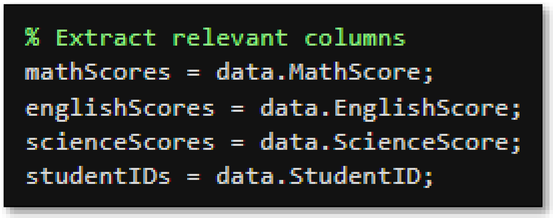
```
% Import the dataset
data = readtable('students.csv','PreserveVariableNames',true);
```

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



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

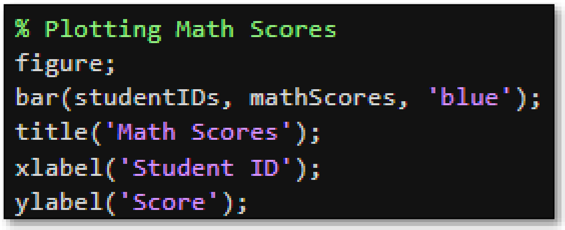
3. Extract the relevant columns.



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

s

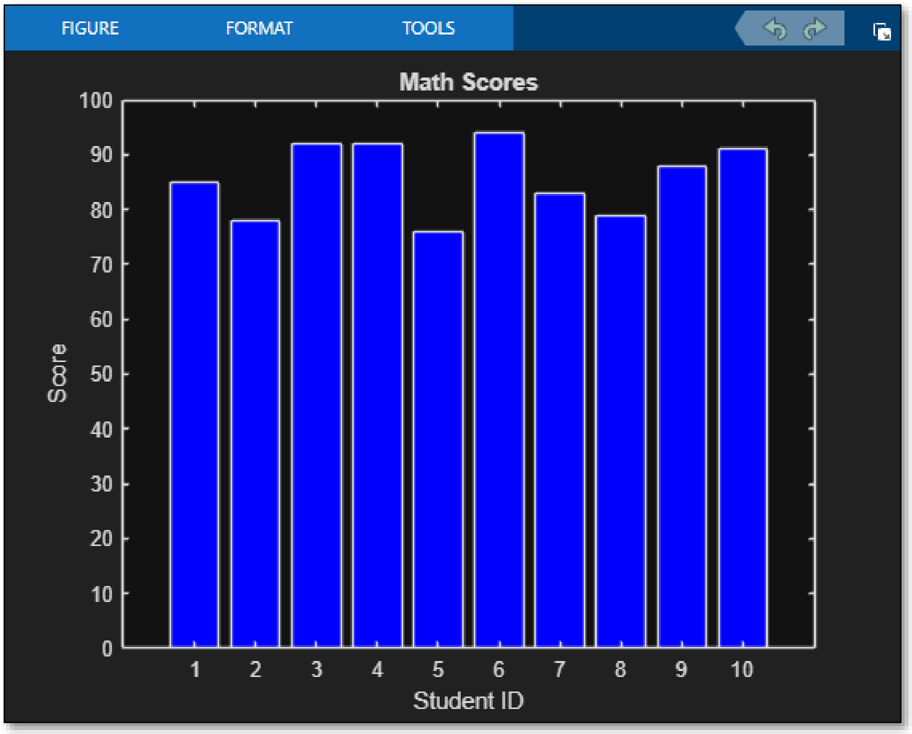
4. Plot MathScores. A MathScores plot is shown here as an example of how to complete the others.



```
% Plotting Math Scores
figure;
bar(studentIDs, mathScores, 'blue');
title('Math Scores');
xlabel('Student ID');
ylabel('Score');
```

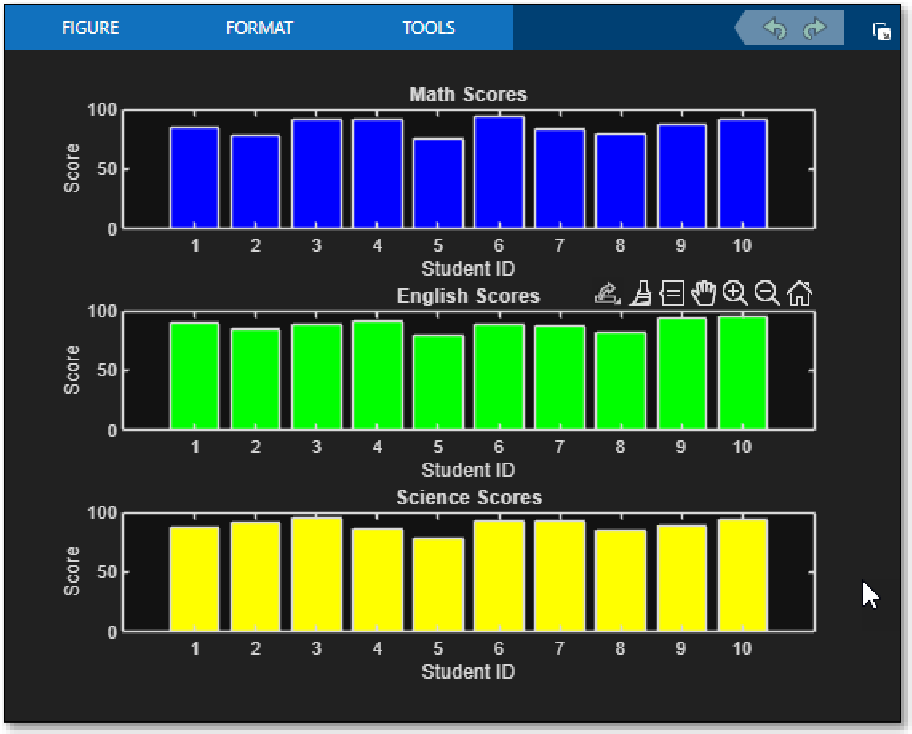
5. Plot English Scores and Science Scores. You will have 3 separate plots.

Example run:



6. Use subplots to place all three plots into a single plot.

Example run:



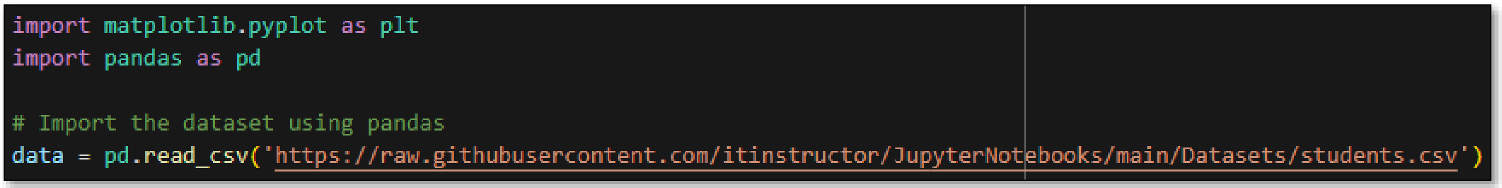
Assignment 2: Plots with Python Matplotlib

Use Google Colab for the Python tutorials.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

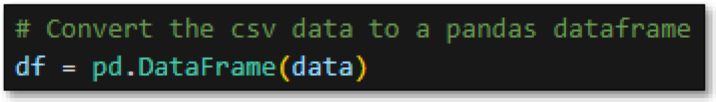
1. Import **students.csv** from GitHub. Read the data using the pandas read_csv() function.
2. Right click the link below → Copy Link → Paste the link into the code as shown.

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



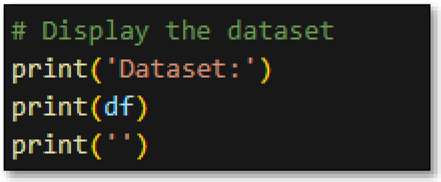
```
import matplotlib.pyplot as plt
import pandas as pd

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

3. Convert the csv data to a pandas dataframe.



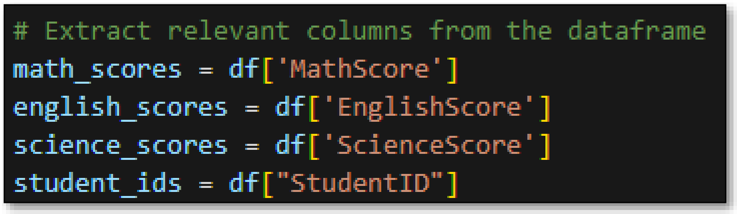
```
# Convert the csv data to a pandas dataframe
df = pd.DataFrame(data)
```

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



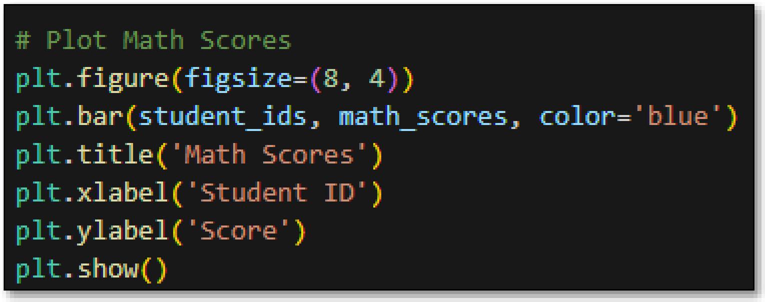
```
# Display the dataset
print('Dataset:')
print(df)
print('')
```

5. Extract the relevant columns.



```
# Extract relevant columns from the dataframe
math_scores = df['MathScore']
english_scores = df['EnglishScore']
science_scores = df['ScienceScore']
student_ids = df["StudentID"]
```

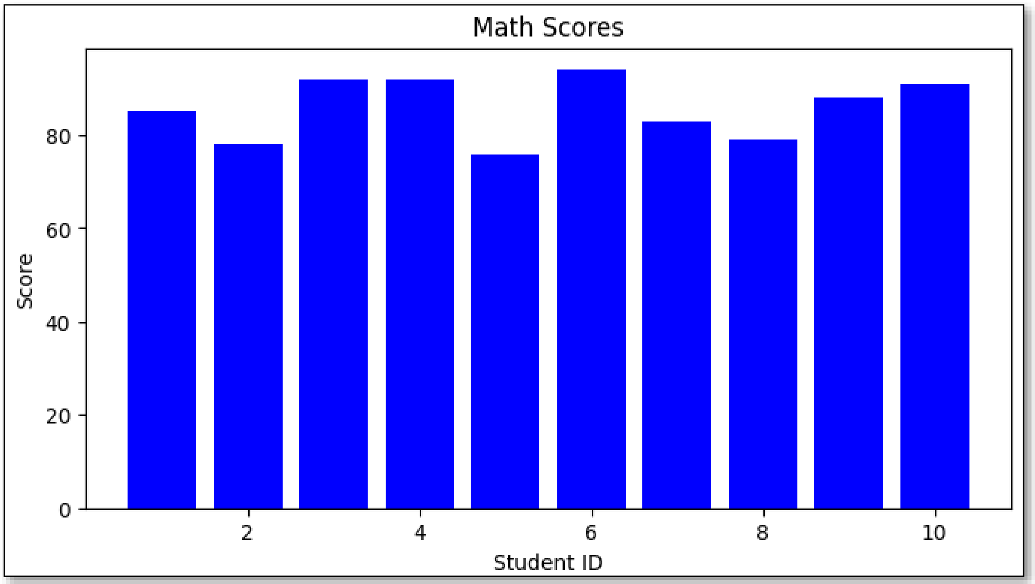
6. Plot MathScore, EnglishScore, and ScienceScore. MathScore is shown here as an example of how to complete the others.



```
# Plot Math Scores
plt.figure(figsize=(8, 4))
plt.bar(student_ids, math_scores, color='blue')
plt.title('Math Scores')
plt.xlabel('Student ID')
plt.ylabel('Score')
plt.show()
```

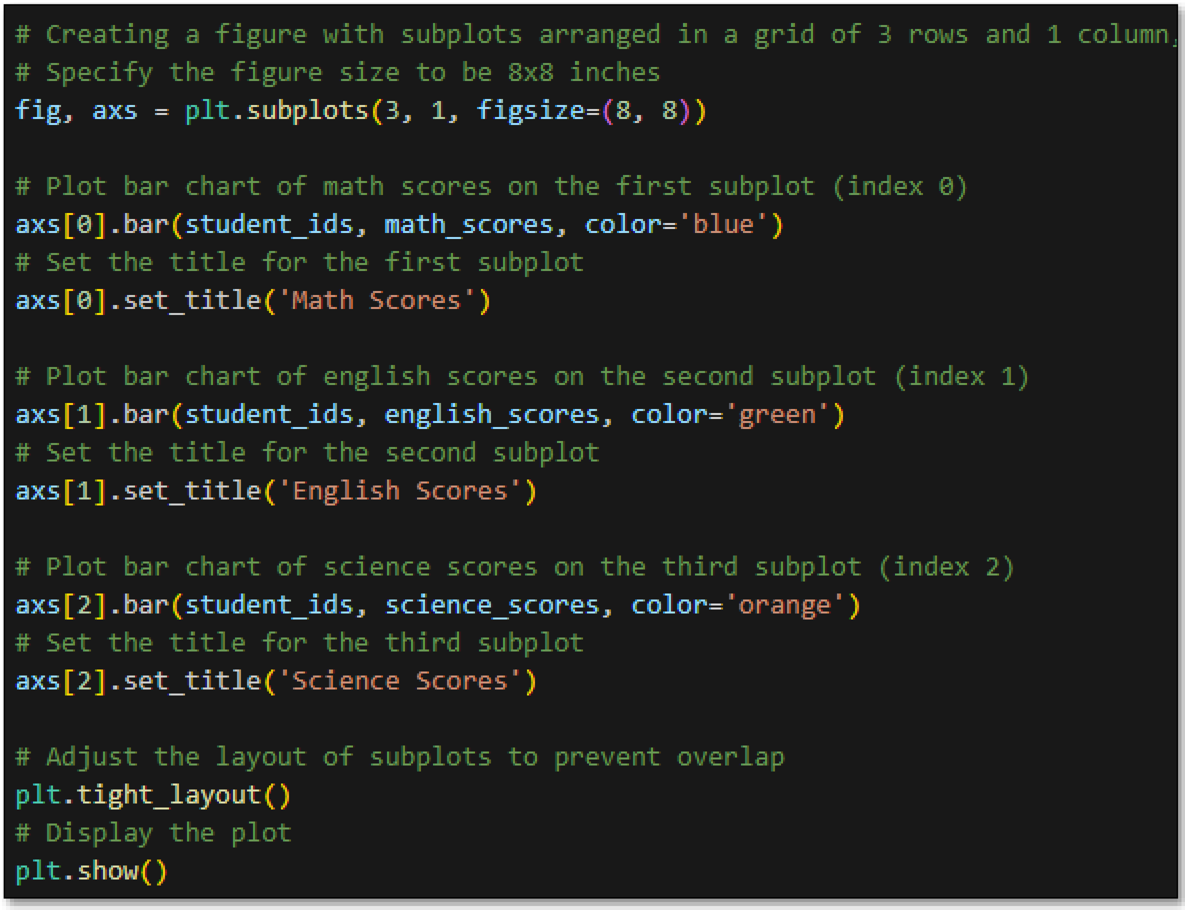
7. To finish the program on your own, plot English Scores, and Science Scores.

Example run:

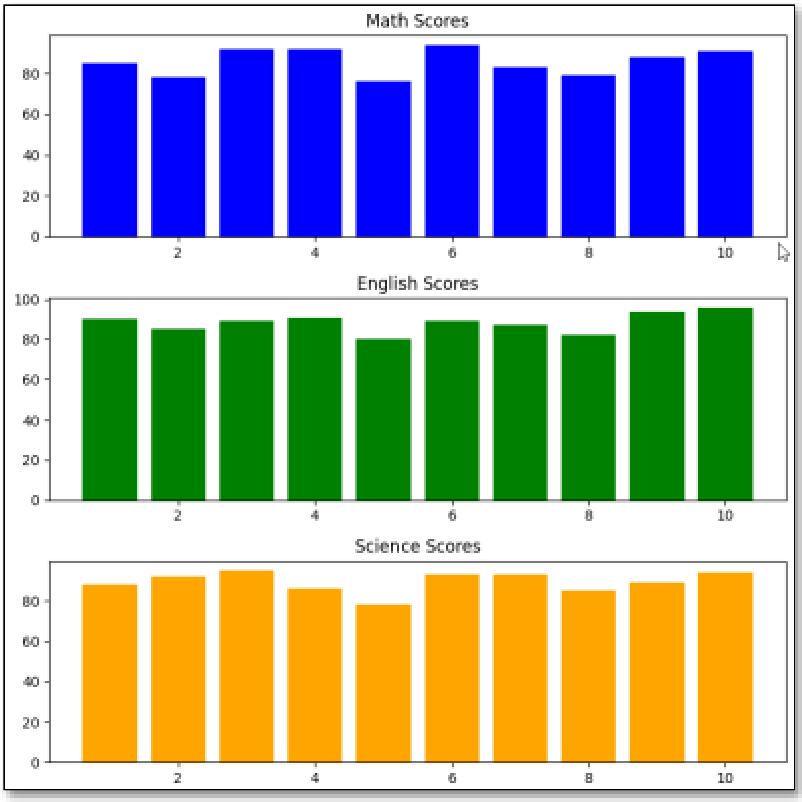


Tutorial 1: Subplots with Python

Same dataset, same plot as MATLAB. You want this in the same Colab notebook that has the student data.

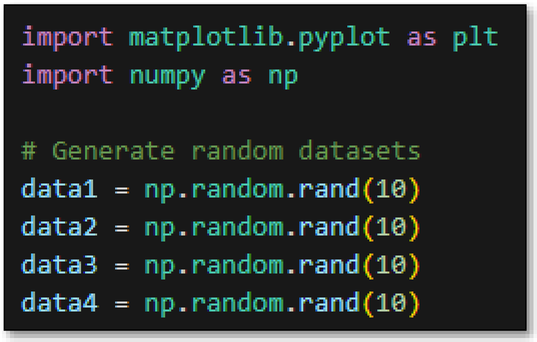


```python
# Creating a figure with subplots arranged in a grid of 3 rows and 1 column,
# Specify the figure size to be 8x8 inches
fig, axs = plt.subplots(3, 1, figsize=(8, 8))

# Plot bar chart of math scores on the first subplot (index 0)
axs[0].bar(student_ids, math_scores, color='blue')
# Set the title for the first subplot
axs[0].set_title('Math Scores')

# Plot bar chart of english scores on the second subplot (index 1)
axs[1].bar(student_ids, english_scores, color='green')
# Set the title for the second subplot
axs[1].set_title('English Scores')

# Plot bar chart of science scores on the third subplot (index 2)
axs[2].bar(student_ids, science_scores, color='orange')
# Set the title for the third subplot
axs[2].set_title('Science Scores')

# Adjust the layout of subplots to prevent overlap
plt.tight_layout()
# Display the plot
plt.show()
```



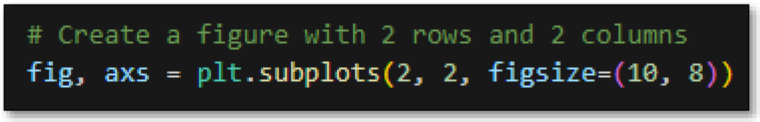
Assignment 3: Subplots with Matplotlib

1. Generate 4 random datasets with numpy.



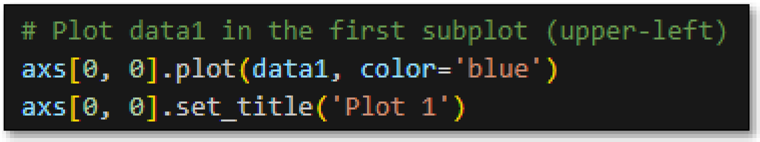
```
import matplotlib.pyplot as plt
import numpy as np

# Generate random datasets
data1 = np.random.rand(10)
data2 = np.random.rand(10)
data3 = np.random.rand(10)
data4 = np.random.rand(10)
```

2. Create a plot figure with 2 rows and 2 columns.



```
# Create a figure with 2 rows and 2 columns
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
```

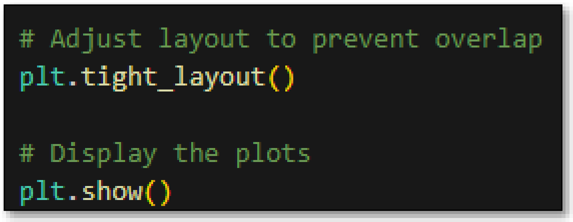
3. Plot the first dataset. Notice the second set of numbers to determine the locations of the subplot.



```python
# Plot data1 in the first subplot (upper-left)
axs[0, 0].plot(data1, color='blue')
axs[0, 0].set_title('Plot 1')
```

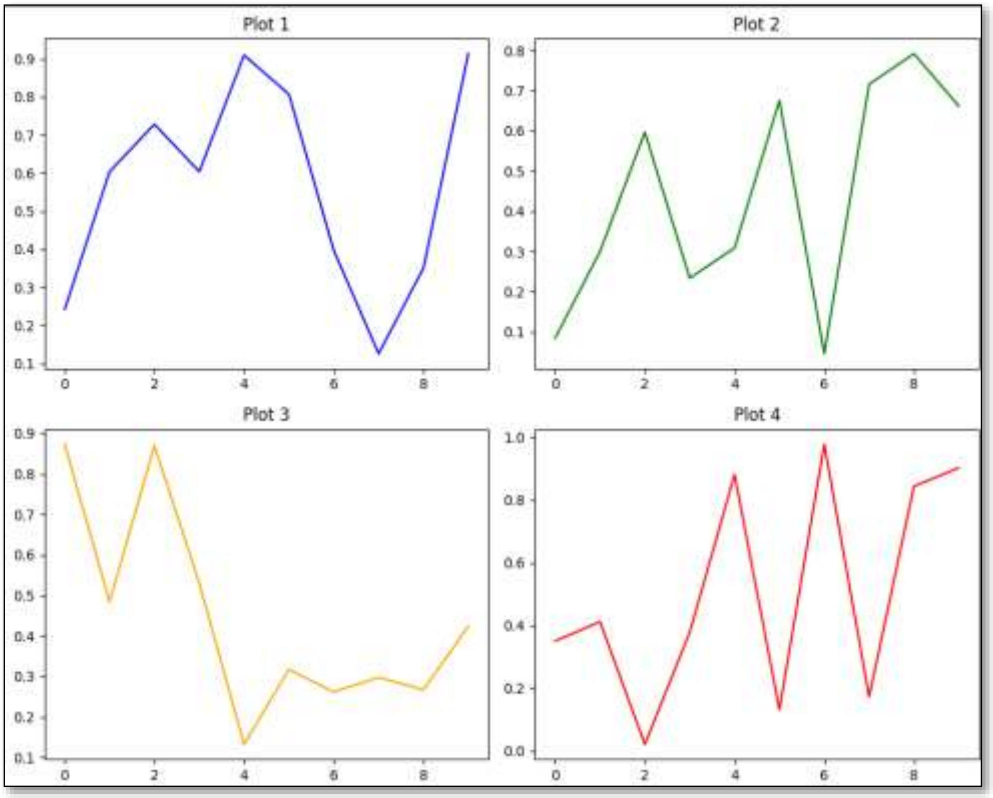
4. Finish the other dataset plots.

5. Set the layout and display the plots.



```python
# Adjust layout to prevent overlap
plt.tight_layout()

# Display the plots
plt.show()
```

Example run:



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 a screenshot of the Command Window showing the successful execution of each script.
3. Attach all to the assignment in Blackboard.