Name: Sadaf Iqbal
Department: BS Computer Science (BSCS)
University Roll No.: 23-ARID-740
Course: Python Programming
Semester: 5th – Evening
Submitted To: [M.Azhar]
Date: [12/08/2025]

---------- Statistic Analysis ----------

create file/array of Data for different Domain like computing, medical , social sciences etc . Generate sample Data using AI OR DOWNLOAD ANY SAMPLE DATASET.

and perform Statictic Analysis .

Steps to perform:

code in python

1. load Data Set of different Domain.

2. Statistic Analysis techniques/ formulas

3. peform calculation Function (using build-in/custom function)

4. Show result in Graph & Table format

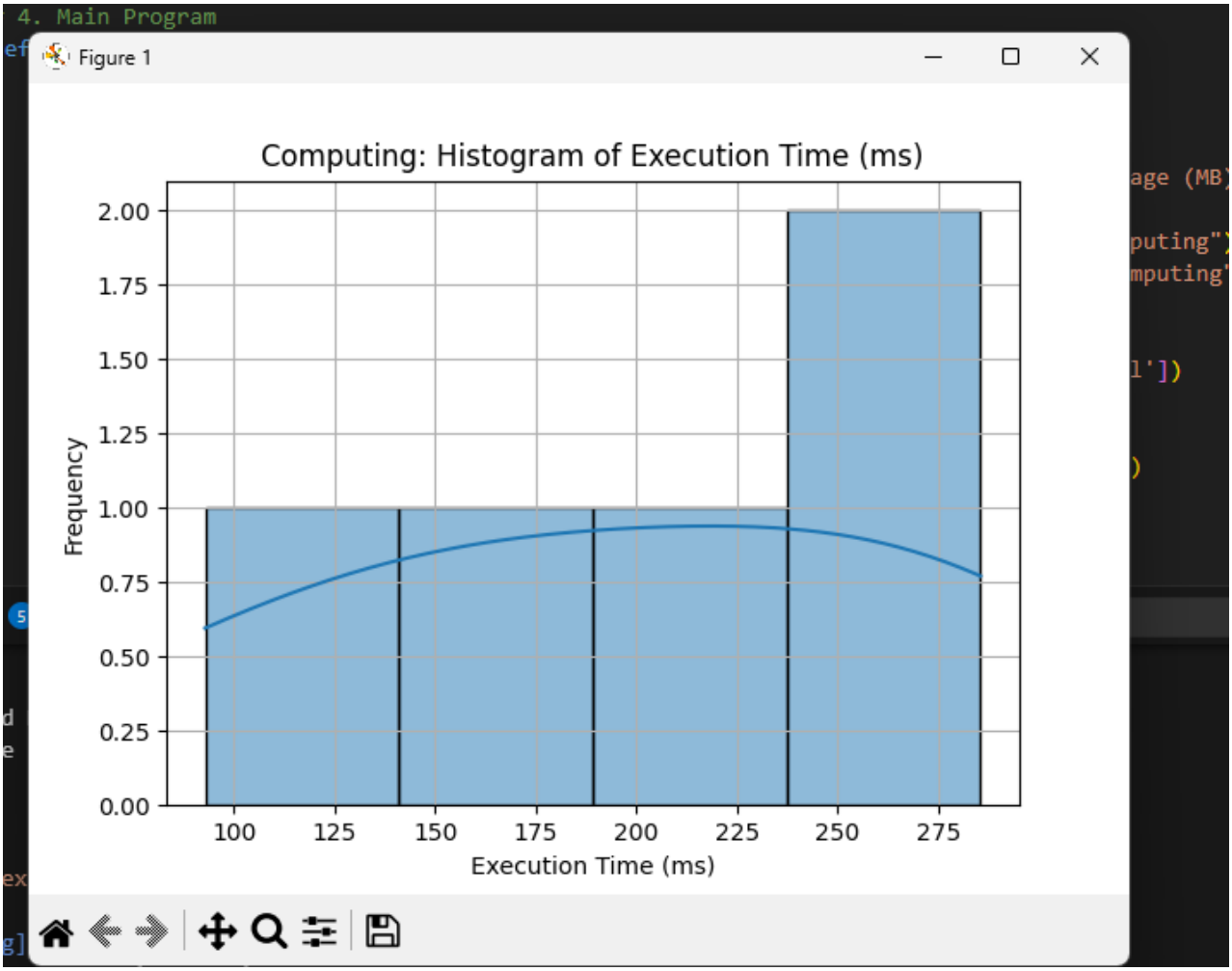
Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

# 1. Generate Sample Datasets from Different Domains
def generate_sample_data():
    # Computing Domain (e.g., algorithm performance)
    computing_data = pd.DataFrame({
        'Algorithm': ['A', 'B', 'C', 'D', 'E'],
        'Execution Time (ms)': np.random.normal(200, 50, 5),
        'Memory Usage (MB)': np.random.normal(120, 15, 5)
    })

    # Medical Domain (e.g., patient data)
    medical_data = pd.DataFrame({
        'Patient ID': range(1, 11),
        'Age': np.random.randint(20, 80, 10),
```

```
        'Blood Pressure': np.random.randint(110, 180, 10),
        'Cholesterol': np.random.randint(150, 250, 10)
    })

    # Social Science Domain (e.g., survey)
    social_data = pd.DataFrame({
        'Respondent ID': range(1, 11),
        'Happiness Score': np.random.uniform(1, 10, 10),
        'Income (k)': np.random.randint(20, 100, 10),
        'Education Level': np.random.choice(['High School', 'Bachelor', 'Master',
'PhD'], 10)
    })

    return computing_data, medical_data, social_data

# 2. Define Statistical Analysis Function
def perform_statistics(df, numeric_columns):
    stats_dict = {}
    for col in numeric_columns:
        mode_val = stats.mode(df[col], keepdims=True).mode[0]  # Fixed for SciPy
>= 1.9
        stats_dict[col] = {
            'Mean': np.mean(df[col]),
            'Median': np.median(df[col]),
            'Mode': mode_val,
            'Standard Deviation': np.std(df[col]),
            'Variance': np.var(df[col]),
            'Min': np.min(df[col]),
            'Max': np.max(df[col]),
        }
    return pd.DataFrame(stats_dict)

# 3. Plotting Functions
def plot_histograms(df, numeric_columns, title):
    for col in numeric_columns:
        plt.figure()
        sns.histplot(df[col], kde=True)
        plt.title(f"{title}: Histogram of {col}")
        plt.xlabel(col)
        plt.ylabel('Frequency')
        plt.grid(True)
        plt.show()

def plot_correlation(df, numeric_columns, title):
    plt.figure(figsize=(6, 4))
```

```
    sns.heatmap(df[numeric_columns].corr(), annot=True, cmap='coolwarm')
    plt.title(f"{title}: Correlation Matrix")
    plt.show()

# 4. Main Program
def main():
    computing_data, medical_data, social_data = generate_sample_data()

    print("---------- Computing Domain ----------")
    comp_stats = perform_statistics(computing_data, ['Execution Time (ms)',
'Memory Usage (MB)'])
    print(comp_stats)
    plot_histograms(computing_data, ['Execution Time (ms)', 'Memory Usage (MB)'],
"Computing")
    plot_correlation(computing_data, ['Execution Time (ms)', 'Memory Usage
(MB)'], "Computing")

    print("\n---------- Medical Domain ----------")
    med_stats = perform_statistics(medical_data, ['Age', 'Blood Pressure',
'Cholesterol'])
    print(med_stats)
    plot_histograms(medical_data, ['Age', 'Blood Pressure', 'Cholesterol'],
"Medical")
    plot_correlation(medical_data, ['Age', 'Blood Pressure', 'Cholesterol'],
"Medical")

    print("\n---------- Social Sciences Domain ----------")
    social_numeric = ['Happiness Score', 'Income (k)']
    social_stats = perform_statistics(social_data, social_numeric)
    print(social_stats)
    plot_histograms(social_data, social_numeric, "Social Sciences")
    plot_correlation(social_data, social_numeric, "Social Sciences")

# ✅ Correct way to run the script
if __name__ == "__main__":
    main()
```

output:


4. Main Program
Figure 1
Computing: Histogram of Execution Time (ms)
Frequency
Execution Time (ms)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
100
125
150
175
200
225
250
275

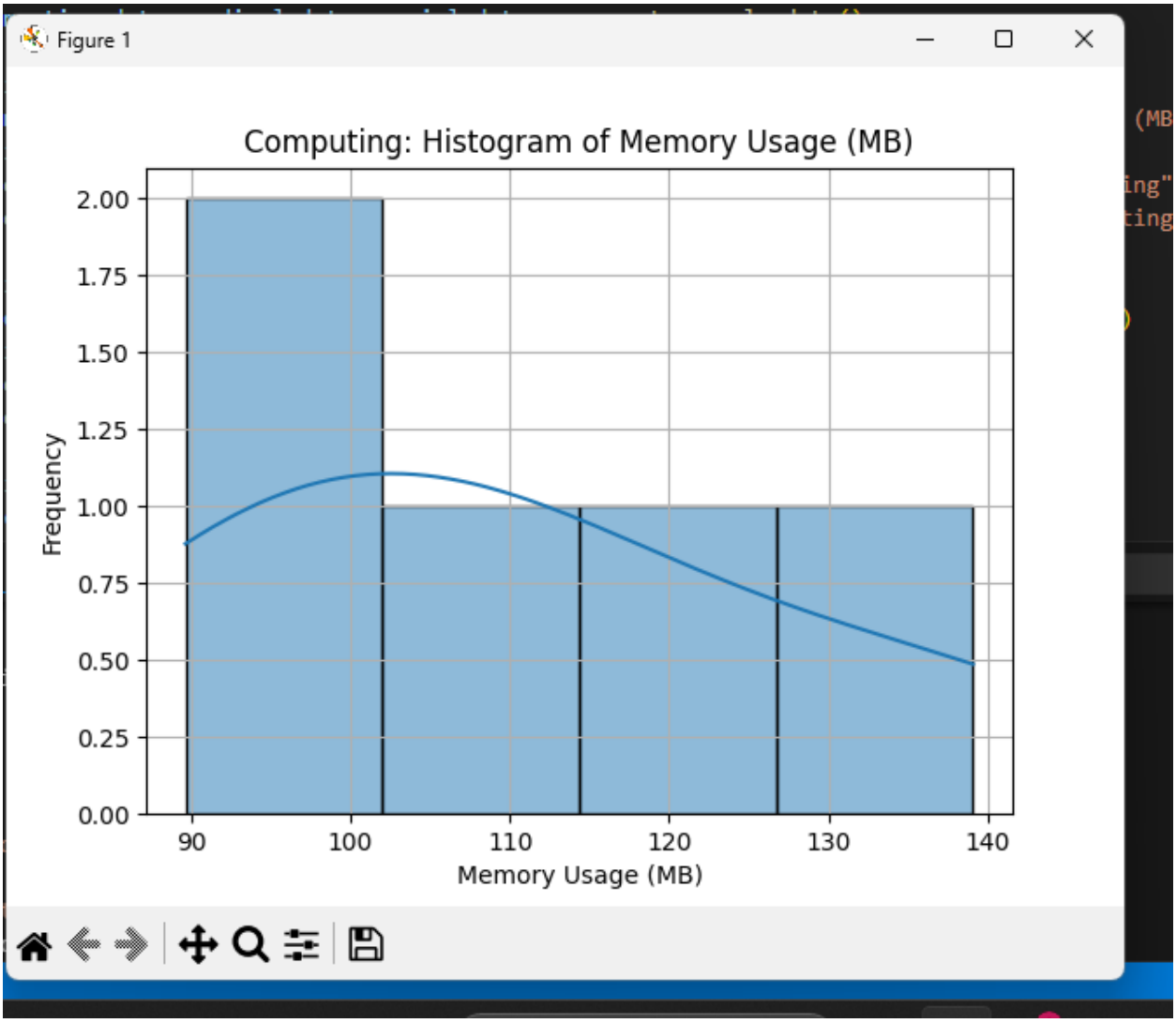


Figure 1
Computing: Histogram of Memory Usage (MB)
Frequency
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
90
100
110
120
130
140
Memory Usage (MB)

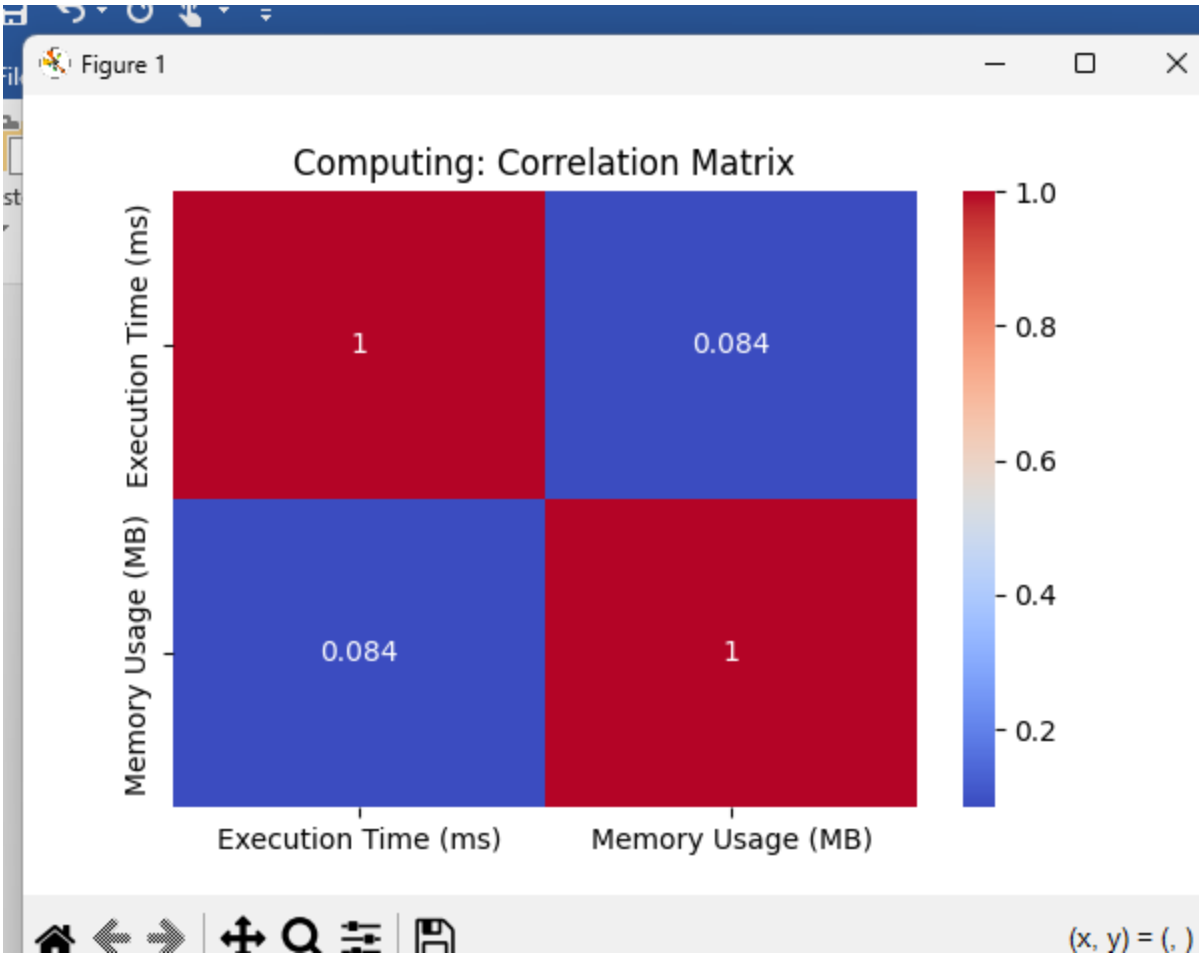


Figure 1
Computing: Correlation Matrix
Execution Time (ms)
Memory Usage (MB)
1
0.084
0.084
1
Execution Time (ms)
Memory Usage (MB)
1.0
0.8
0.6
0.4
0.2
(x, y) = (,)

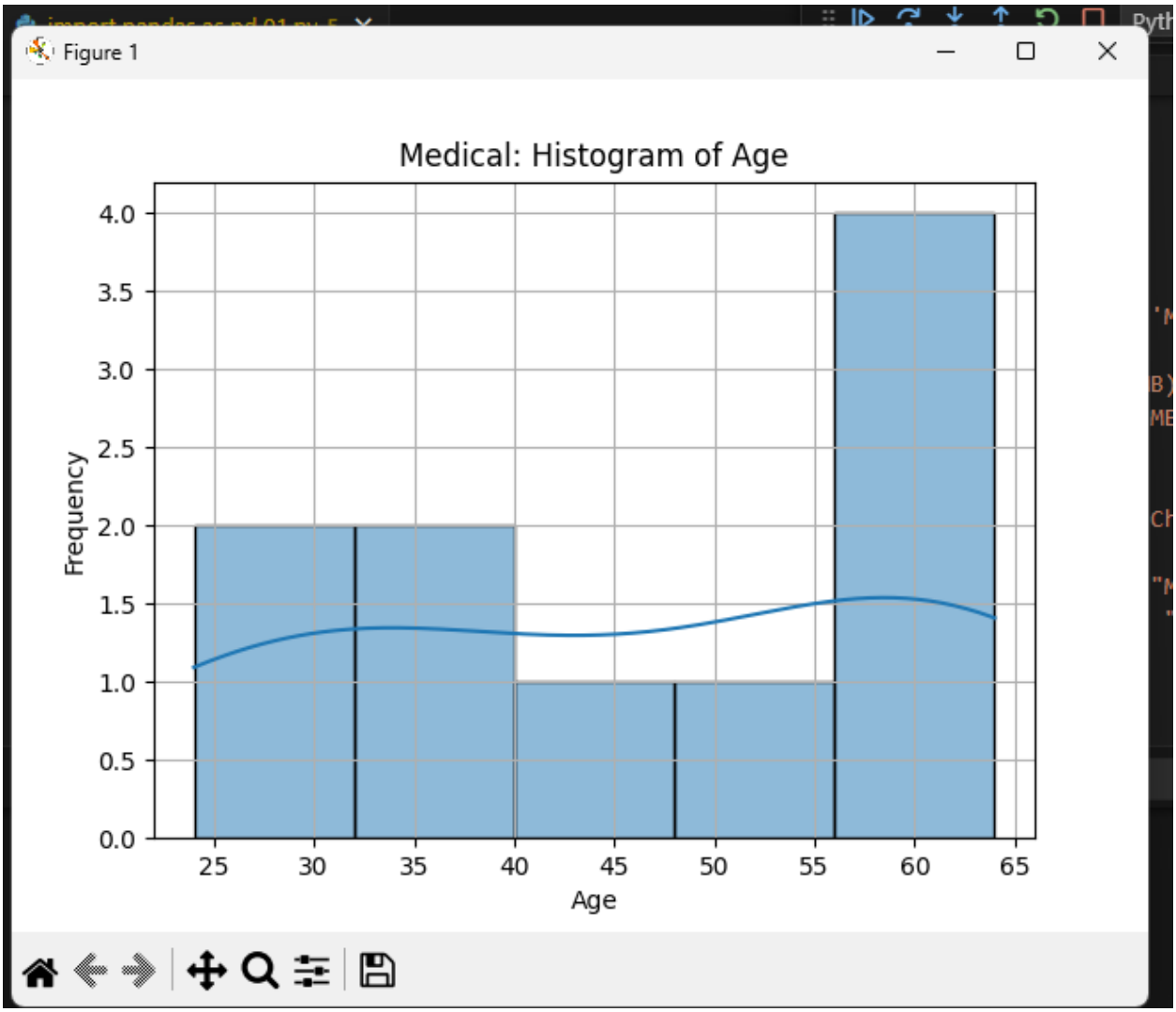


Figure 1
Medical: Histogram of Age
Frequency
Age
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
25
30
35
40
45
50
55
60
65

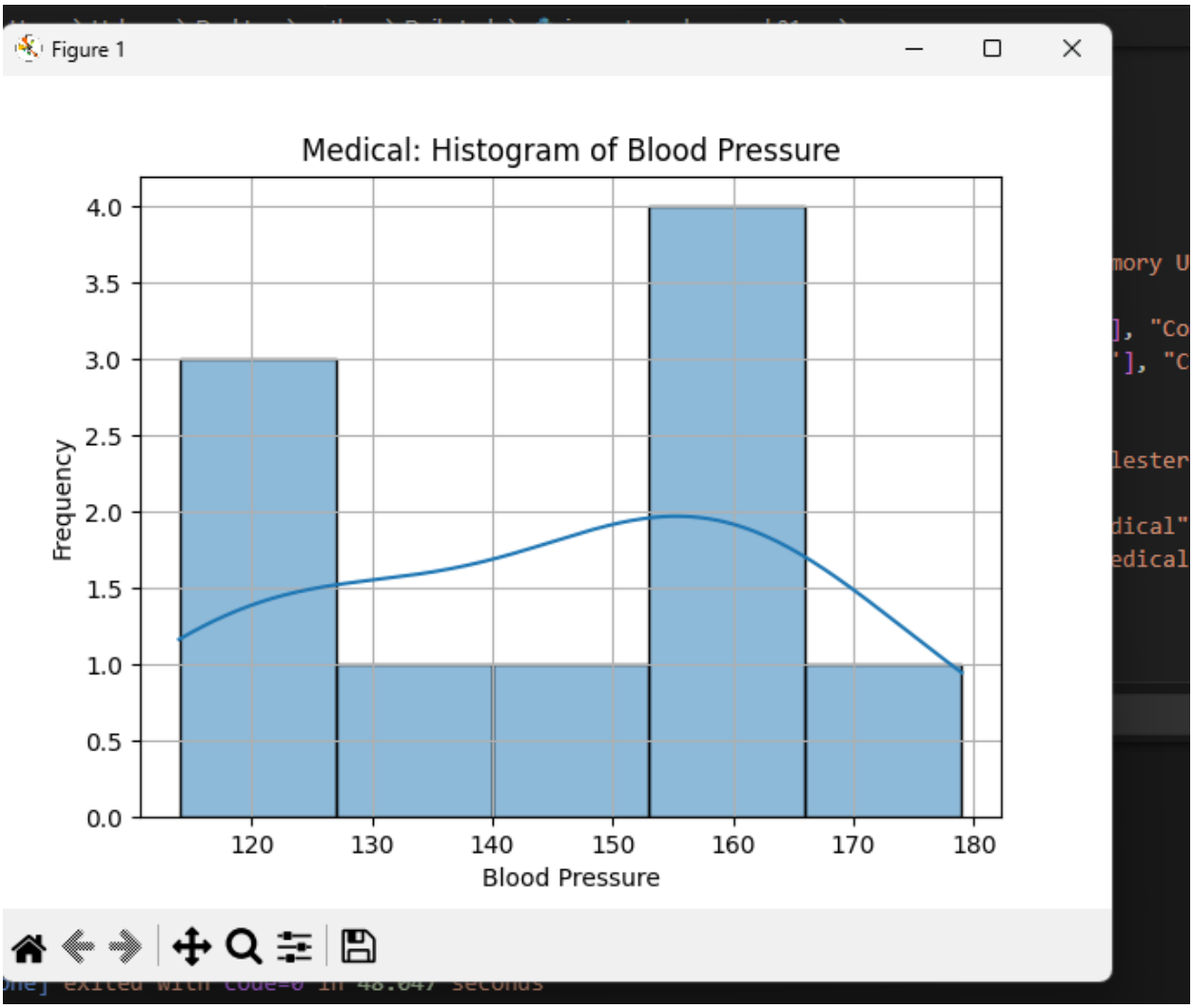


Figure 1
Medical: Histogram of Blood Pressure
Frequency
Blood Pressure
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
120
130
140
150
160
170
180

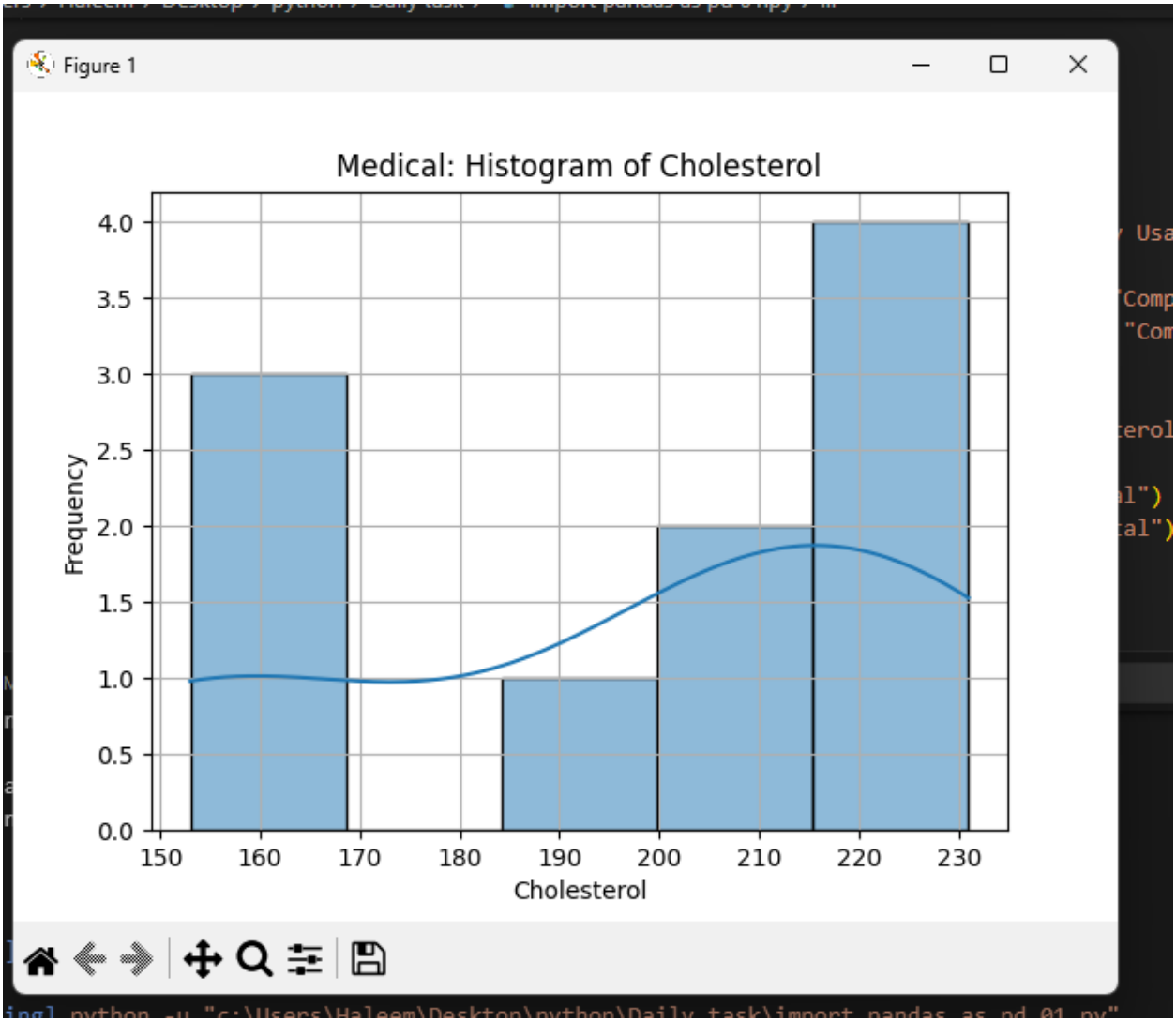


Figure 1
Medical: Histogram of Cholesterol
Frequency
Cholesterol
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
150
160
170
180
190
200
210
220
230

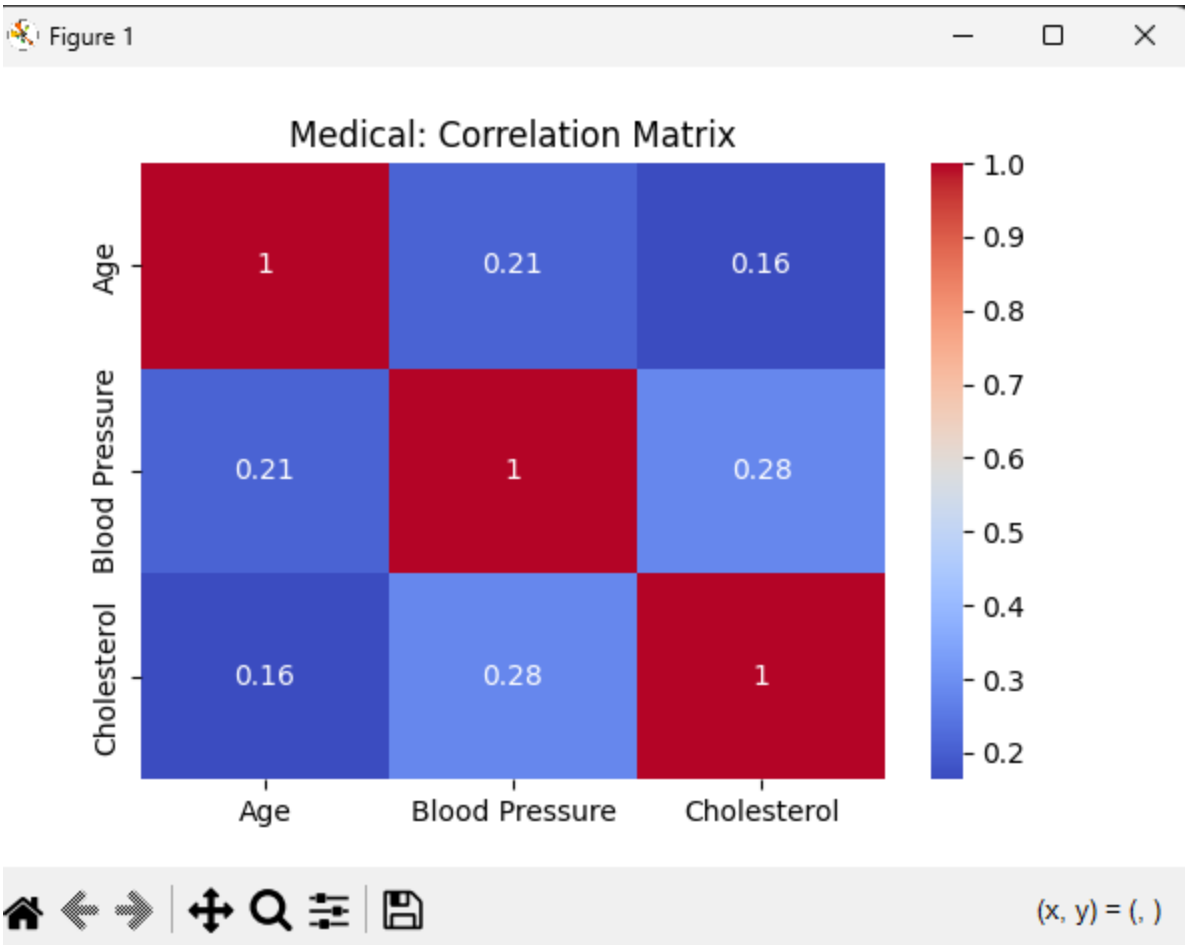


Figure 1
Medical: Correlation Matrix
Age
Blood Pressure
Cholesterol
1
0.21
0.16
0.21
1
0.28
0.16
0.28
1
Age
Blood Pressure
Cholesterol
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
(x, y) = (,)

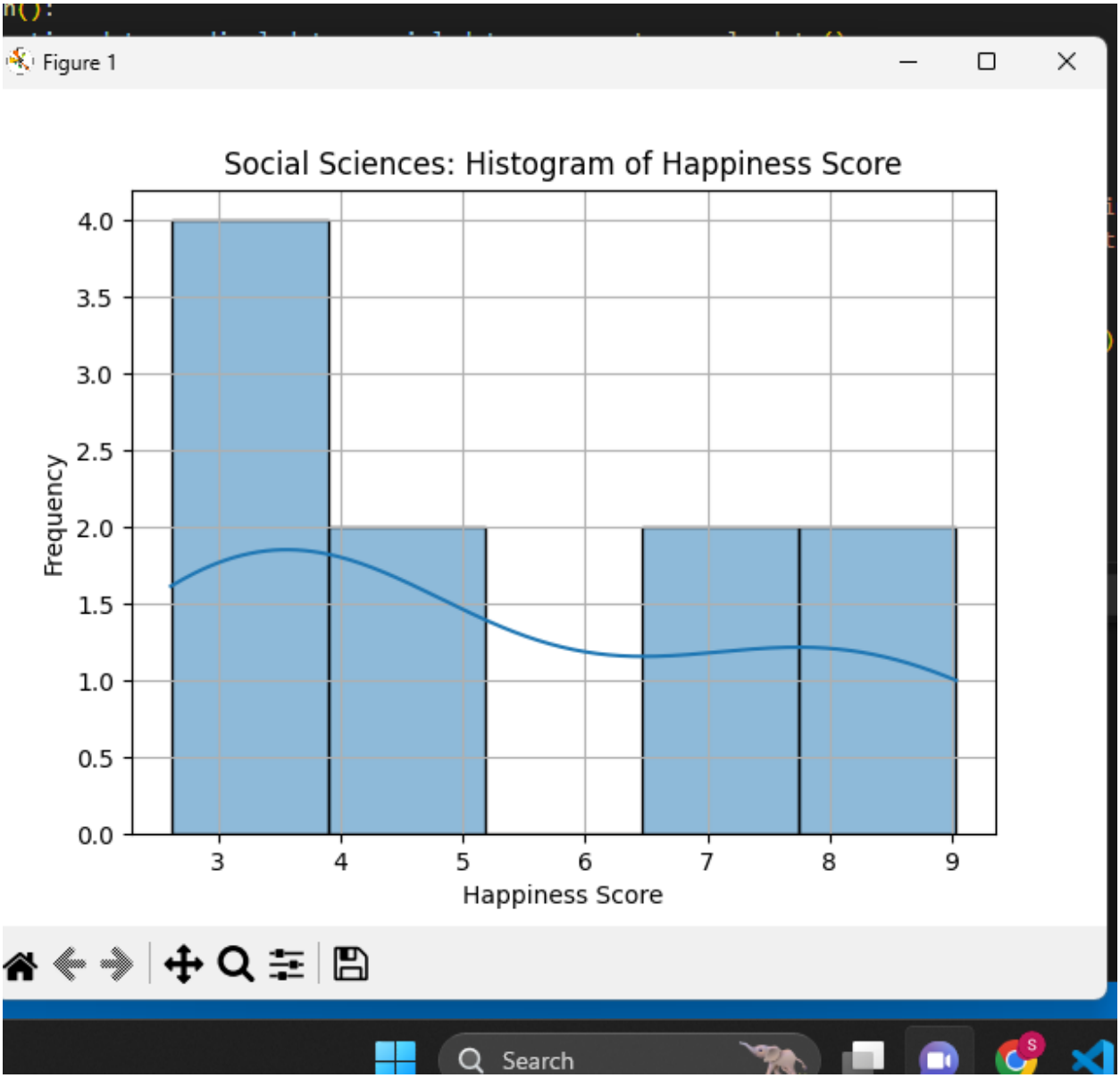
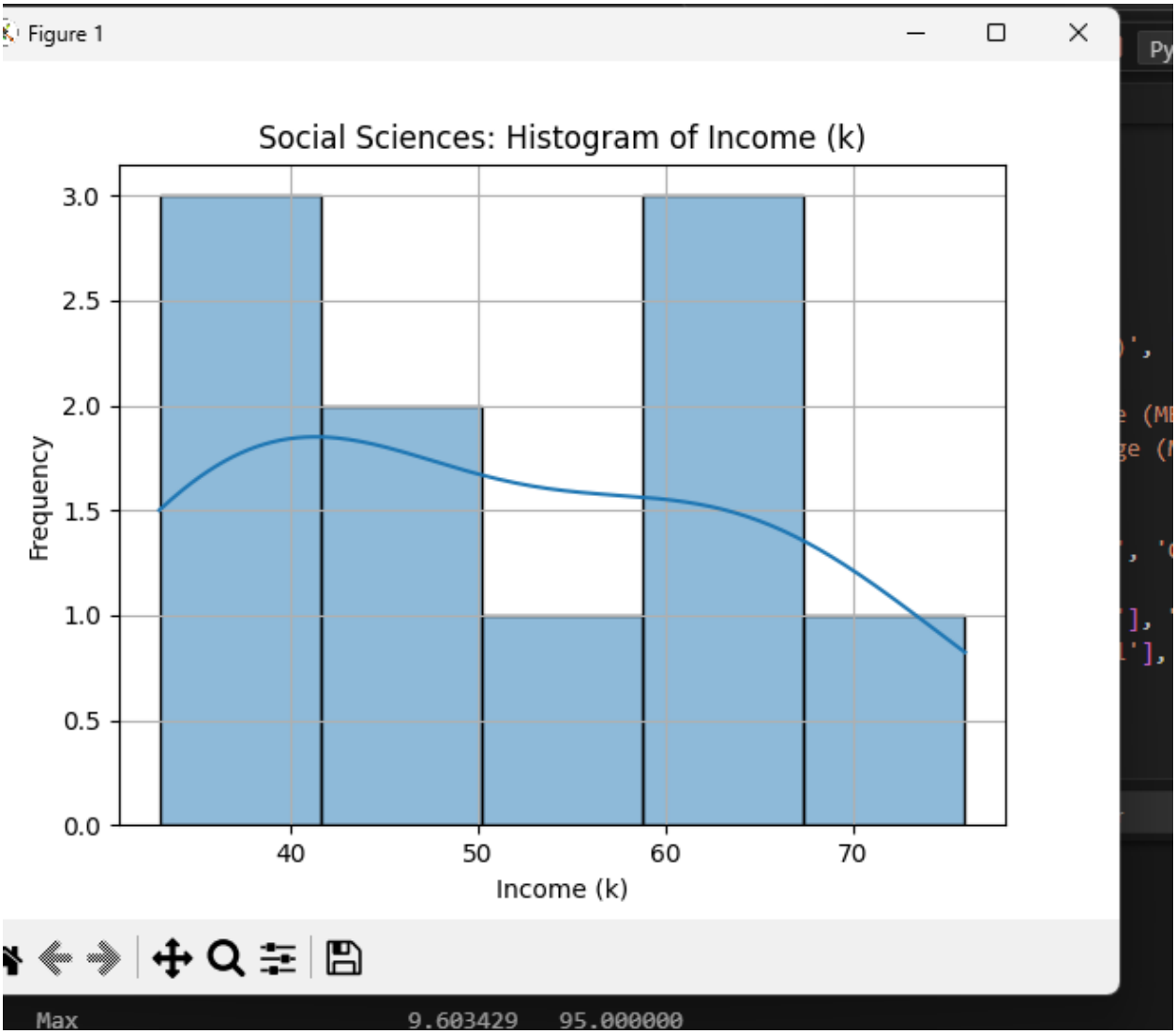
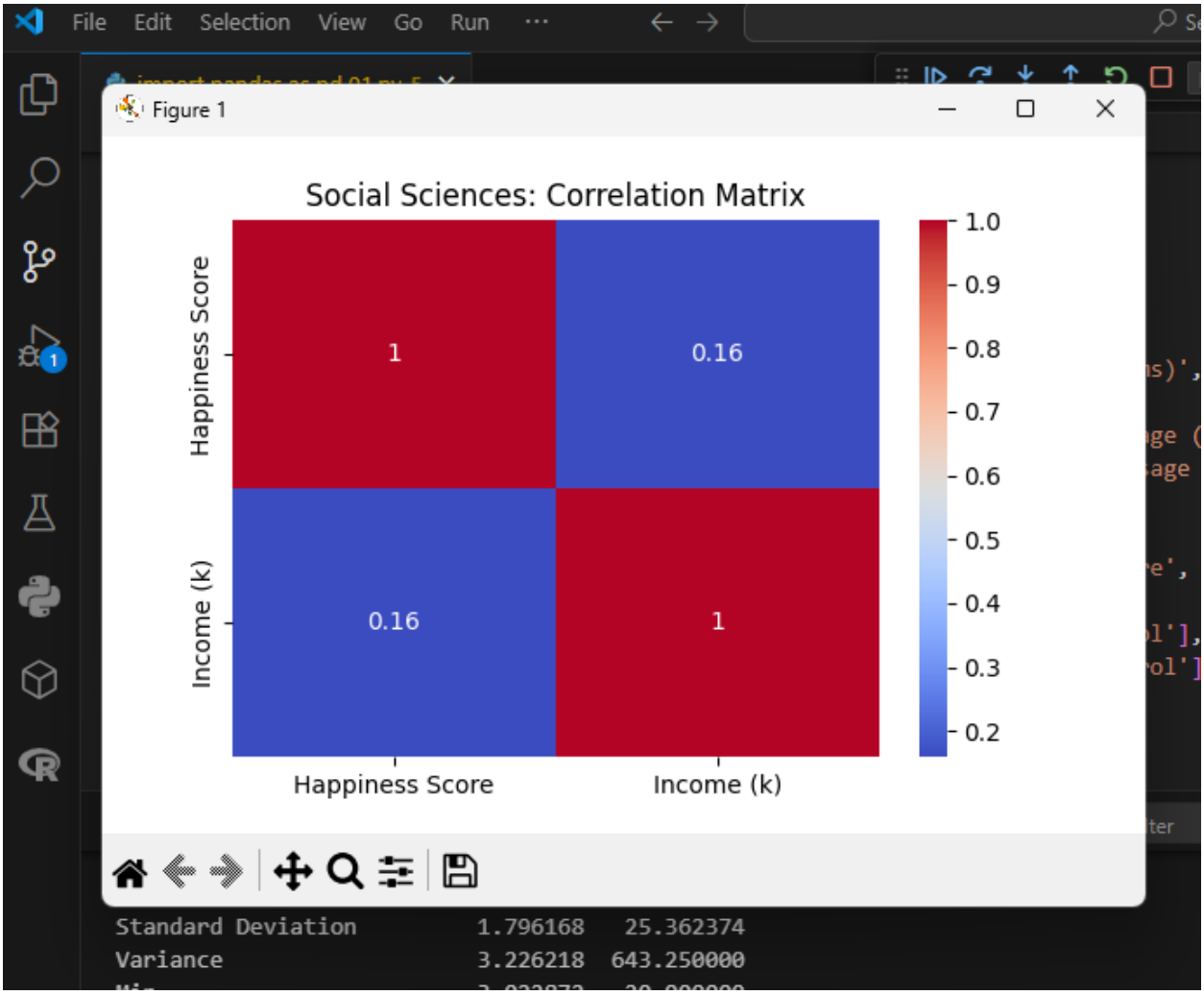


Figure 1
Social Sciences: Histogram of Happiness Score
Frequency
Happiness Score
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
3
4
5
6
7
8
9
Search



Max 9.603429 95.000000

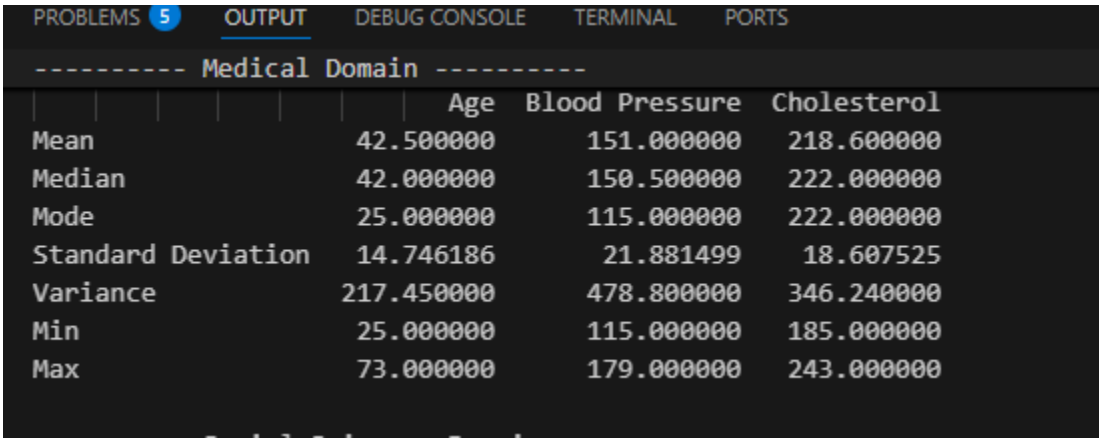


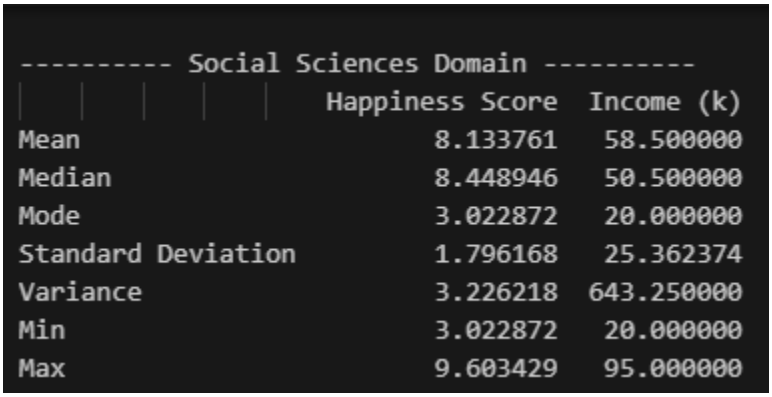
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

---------- Computing Domain ----------

	Execution Time (ms)	Memory Usage (MB)
Mean	187.485199	124.003527
Median	198.670953	120.789707
Mode	126.993003	118.700724
Standard Deviation	50.361013	5.003301
Variance	2536.231583	25.033019
Min	126.993003	118.700724
Max	258.917473	130.085378

Medical Domain


PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
---------- Medical Domain ----------
Age Blood Pressure Cholesterol
Mean 42.500000 151.000000 218.600000
Median 42.000000 150.500000 222.000000
Mode 25.000000 115.000000 222.000000
Standard Deviation 14.746186 21.881499 18.607525
Variance 217.450000 478.800000 346.240000
Min 25.000000 115.000000 185.000000
Max 73.000000 179.000000 243.000000


---------- Social Sciences Domain ----------
Happiness Score Income (k)
Mean 8.133761 58.500000
Median 8.448946 50.500000
Mode 3.022872 20.000000
Standard Deviation 1.796168 25.362374
Variance 3.226218 643.250000
Min 3.022872 20.000000
Max 9.603429 95.000000