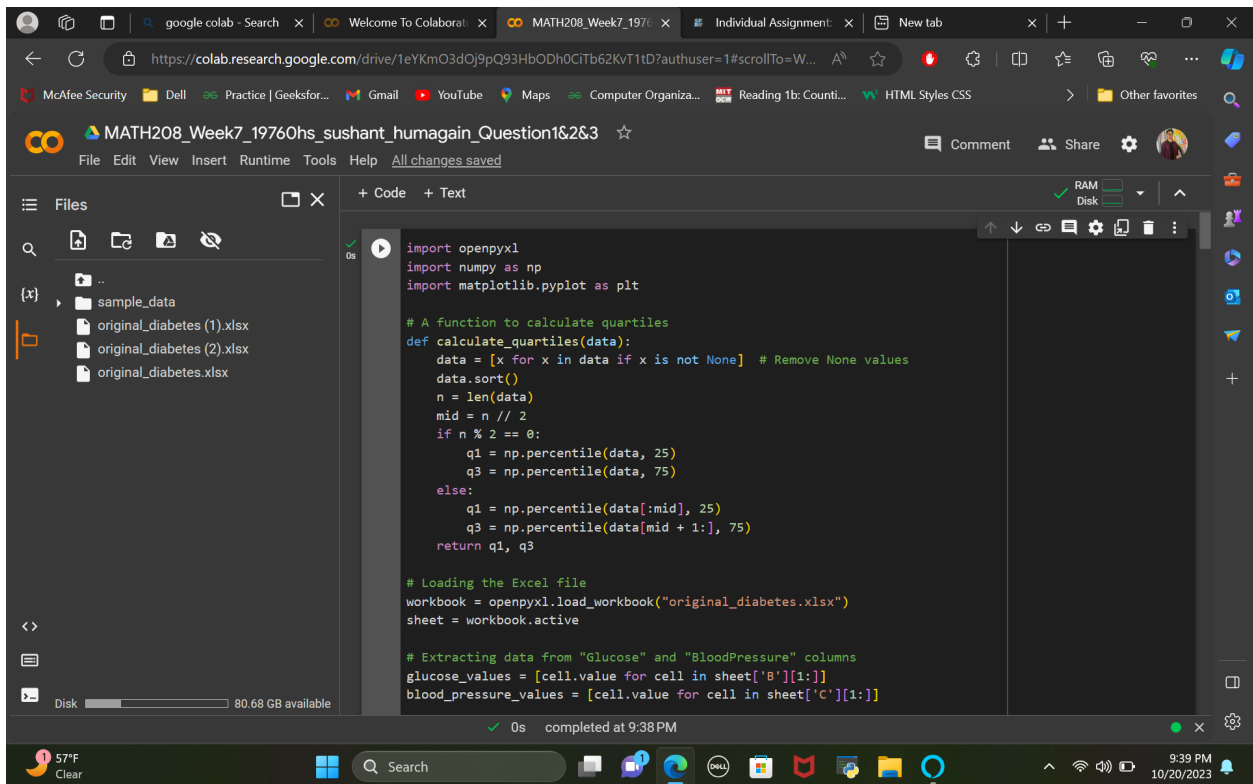


Question no.1



The screenshot shows a Google Colab environment with the following code in the main editor:

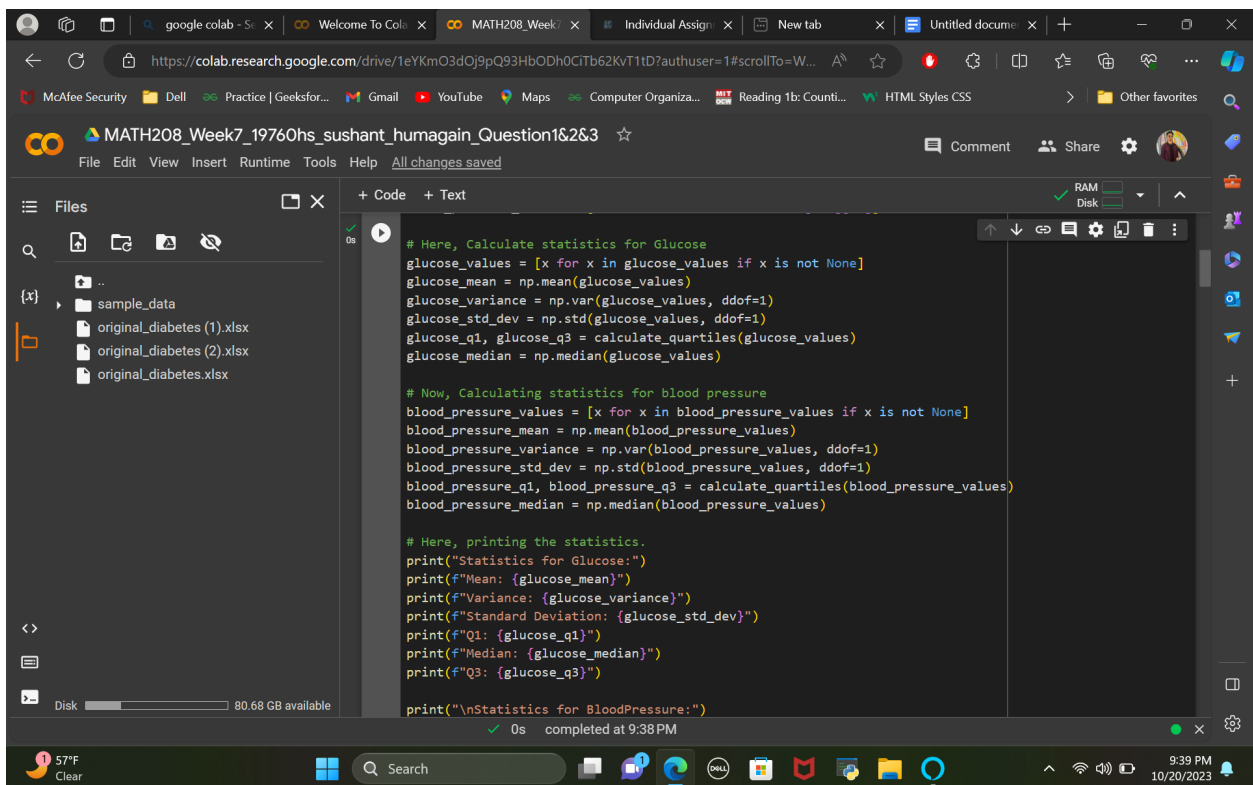
```
import openpyxl
import numpy as np
import matplotlib.pyplot as plt

# A function to calculate quartiles
def calculate_quartiles(data):
    data = [x for x in data if x is not None] # Remove None values
    data.sort()
    n = len(data)
    mid = n // 2
    if n % 2 == 0:
        q1 = np.percentile(data, 25)
        q3 = np.percentile(data, 75)
    else:
        q1 = np.percentile(data[:mid], 25)
        q3 = np.percentile(data[mid + 1:], 75)
    return q1, q3

# Loading the Excel file
workbook = openpyxl.load_workbook("original_diabetes.xlsx")
sheet = workbook.active

# Extracting data from "Glucose" and "BloodPressure" columns
glucose_values = [cell.value for cell in sheet['B'][1:]]
blood_pressure_values = [cell.value for cell in sheet['C'][1:]]
```

The file explorer on the left shows a folder named 'sample_data' containing three files: 'original_diabetes (1).xlsx', 'original_diabetes (2).xlsx', and 'original_diabetes.xlsx'. The status bar at the bottom indicates the code is completed at 9:38 PM.



The screenshot shows the continuation of the Python script in Google Colab:

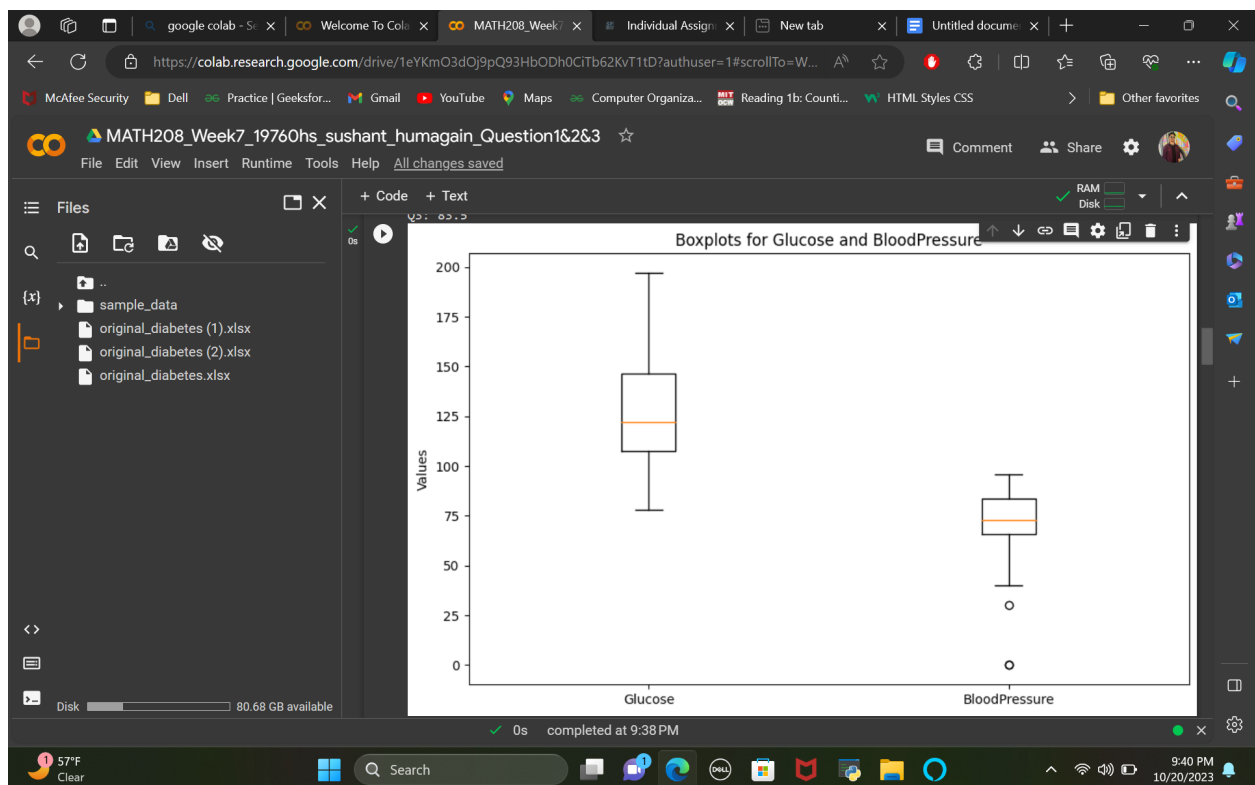
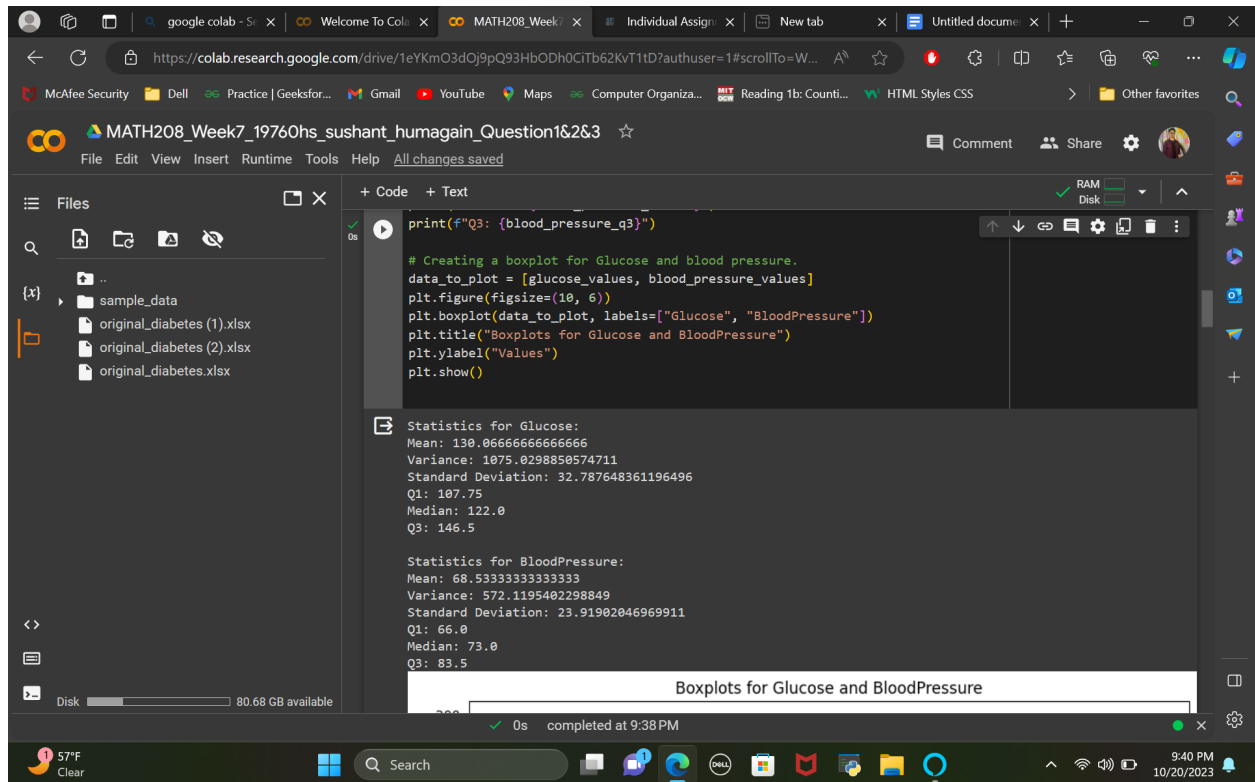
```
# Here, Calculate statistics for Glucose
glucose_values = [x for x in glucose_values if x is not None]
glucose_mean = np.mean(glucose_values)
glucose_variance = np.var(glucose_values, ddof=1)
glucose_std_dev = np.std(glucose_values, ddof=1)
glucose_q1, glucose_q3 = calculate_quartiles(glucose_values)
glucose_median = np.median(glucose_values)

# Now, Calculating statistics for blood pressure
blood_pressure_values = [x for x in blood_pressure_values if x is not None]
blood_pressure_mean = np.mean(blood_pressure_values)
blood_pressure_variance = np.var(blood_pressure_values, ddof=1)
blood_pressure_std_dev = np.std(blood_pressure_values, ddof=1)
blood_pressure_q1, blood_pressure_q3 = calculate_quartiles(blood_pressure_values)
blood_pressure_median = np.median(blood_pressure_values)

# Here, printing the statistics.
print("Statistics for Glucose:")
print(f"Mean: {glucose_mean}")
print(f"Variance: {glucose_variance}")
print(f"Standard Deviation: {glucose_std_dev}")
print(f"Q1: {glucose_q1}")
print(f"Median: {glucose_median}")
print(f"Q3: {glucose_q3}")

print("\nStatistics for BloodPressure:")
```

The file explorer on the left shows the same 'sample_data' folder with the three Excel files. The status bar at the bottom indicates the code is completed at 9:38 PM.



Question no. 2

```
#question no.2
import numpy as np

def verify_Chebyshev_ineq(lst, k):
    u = np.mean(lst)
    sd = np.std(lst)
    cnt = 0
    for x in lst:
        if abs(x - u) < k * sd:
            cnt += 1
    return cnt

lst = np.random.normal(10, 0.5, 50)

for k in [1, 2**0.5, 1.5, 2, 3]:
    cnt = verify_Chebyshev_ineq(lst, k)
    p = cnt / len(lst)
    print(f"When k = {k}, P(|X-u| < k*sd) >= {1-1/k**2} is {p >= 1-1/k**2}")
```

When k = 1, P(|X-u| < k*sd) >= 0.0 is True
When k = 1.4142135623730951, P(|X-u| < k*sd) >= 0.5000000000000001 is True
When k = 1.5, P(|X-u| < k*sd) >= 0.5555555555555556 is True
When k = 2, P(|X-u| < k*sd) >= 0.75 is True
When k = 3, P(|X-u| < k*sd) >= 0.8888888888888888 is True

Question no.3

```
#question no.3
import numpy as np
import matplotlib.pyplot as plt

X = np.array([2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
Y = np.array([30, 25, 95, 115, 265, 325, 570, 700, 1085, 1300])

n = len(X)
sum_x = np.sum(X)
sum_y = np.sum(Y)
sum_xy = np.sum(X * Y)
sum_x2 = np.sum(X ** 2)
sum_y2 = np.sum(Y ** 2)

b1 = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x ** 2)
b0 = (sum_y - b1 * sum_x) / n
r = (n * sum_xy - sum_x * sum_y) / np.sqrt((n * sum_x2 - sum_x ** 2) * (n * sum_y2 - sum_y ** 2))

print(f"b1: {b1:.4f}")
print(f"b0: {b0:.4f}")
print(f"r: {r:.4f}")

plt.scatter(X, Y)
plt.plot(X, b0 + b1 * X)
plt.show()
```

