San Francisco Bay University

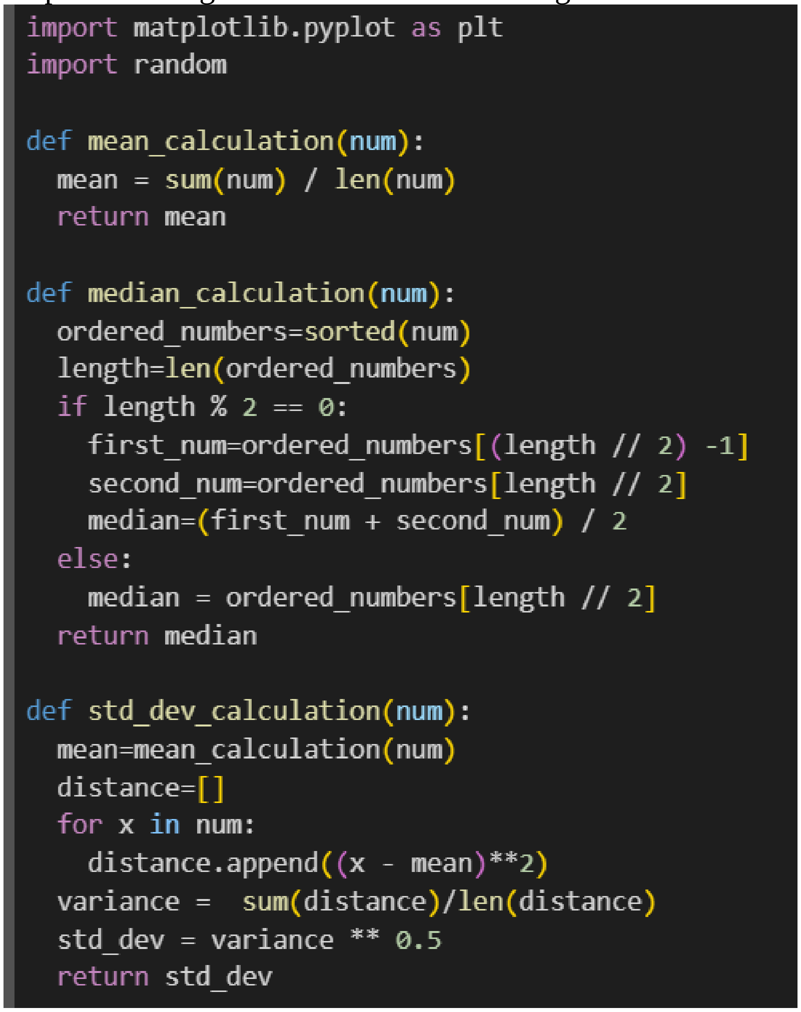
MATH208 - Probability and Statistics
2023 Fall Homework #1 (Engineering)

ID:19702
Kritika Regmi

1. Write the program in any computer language, Python preferred to create 500 random numbers from -20 to +20 in uniform distribution and find the mean, median and standard deviation. After that, plot the histogram with 10 bins. Notice that the only user defined function can be used to calculate the mean, median and standard deviation, don’t directly call existing function from Python library.

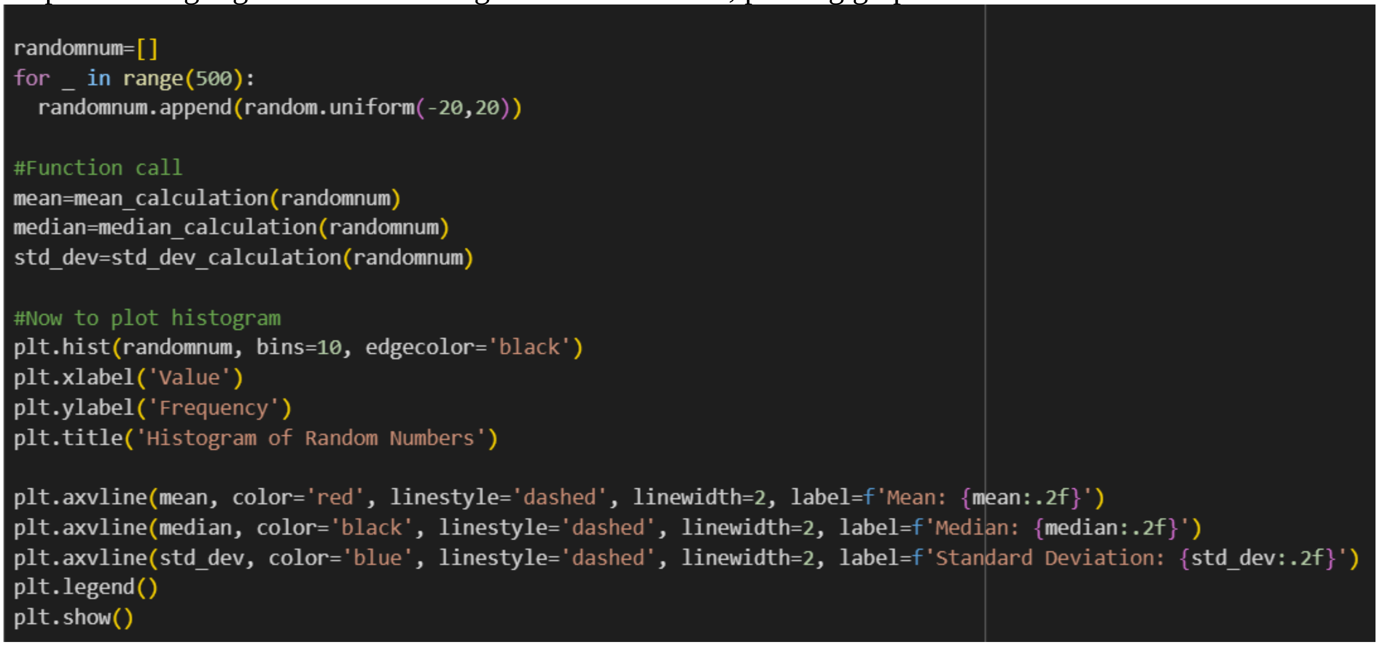
CODE:

Step 1: Defining functions first and defining calculations.



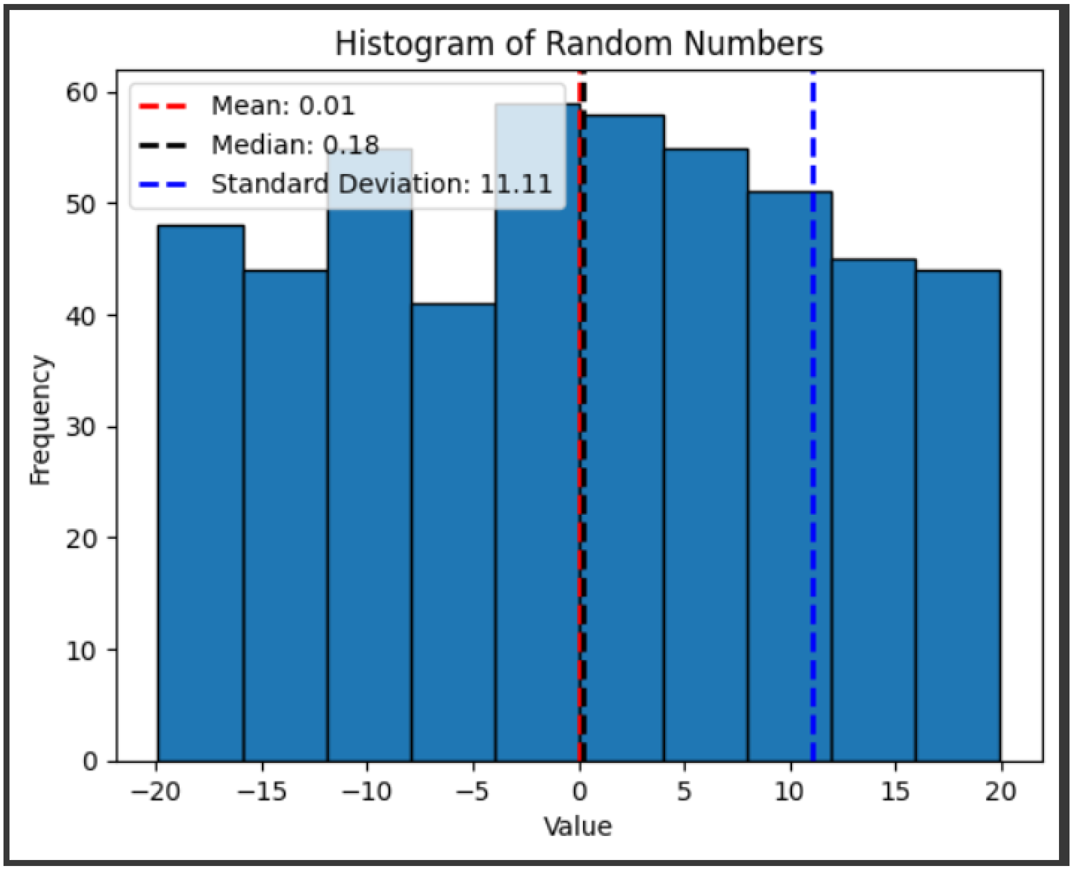
```python
import matplotlib.pyplot as plt
import random

def mean_calculation(num):
  mean = sum(num) / len(num)
  return mean

def median_calculation(num):
  ordered_numbers=sorted(num)
  length=len(ordered_numbers)
  if length % 2 == 0:
    first_num=ordered_numbers[(length // 2) -1]
    second_num=ordered_numbers[length // 2]
    median=(first_num + second_num) / 2
  else:
    median = ordered_numbers[length // 2]
  return median

def std_dev_calculation(num):
  mean=mean_calculation(num)
  distance=[]
  for x in num:
    distance.append((x - mean)**2)
  variance =  sum(distance)/len(distance)
  std_dev = variance ** 0.5
  return std_dev
```

Step 2: Passing arguments and calling the function. Also, plotting graph.



```
randomnum=[]
for _ in range(500):
  randomnum.append(random.uniform(-20,20))

#Function call
mean=mean_calculation(randomnum)
median=median_calculation(randomnum)
std_dev=std_dev_calculation(randomnum)

#Now to plot histogram
plt.hist(randomnum, bins=10, edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Random Numbers')

plt.axvline(mean, color='red', linestyle='dashed', linewidth=2, label=f'Mean: {mean:.2f}')
plt.axvline(median, color='black', linestyle='dashed', linewidth=2, label=f'Median: {median:.2f}')
plt.axvline(std_dev, color='blue', linestyle='dashed', linewidth=2, label=f'Standard Deviation: {std_dev:.2f}')
plt.legend()
plt.show()
```

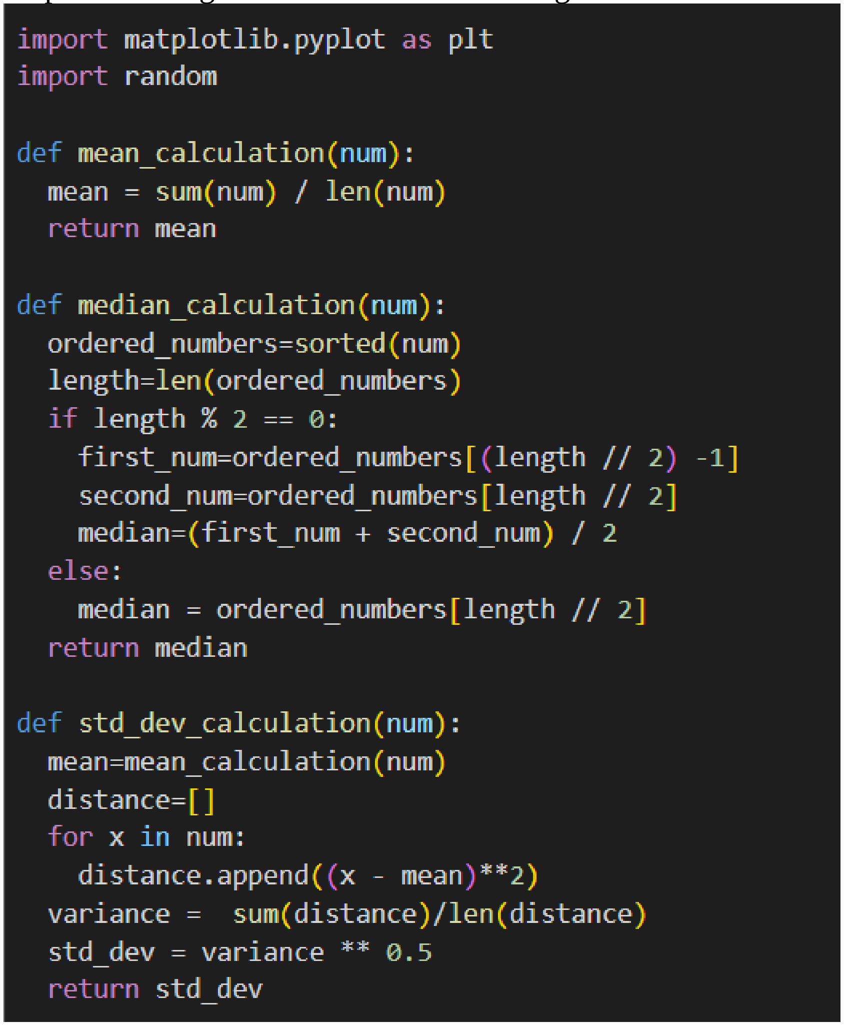
OUTPUT:



2. Similar to the above, write the program to create 500 random numbers with mean = 10 and standard deviation = 0.5 in Gaussian distribution and find the mean, median and standard deviation. After that, plot the histogram with 10 bins. Notice that the only user defined function can be used to calculate the mean, median and standard deviation, don’t directly call existing function from Python library.

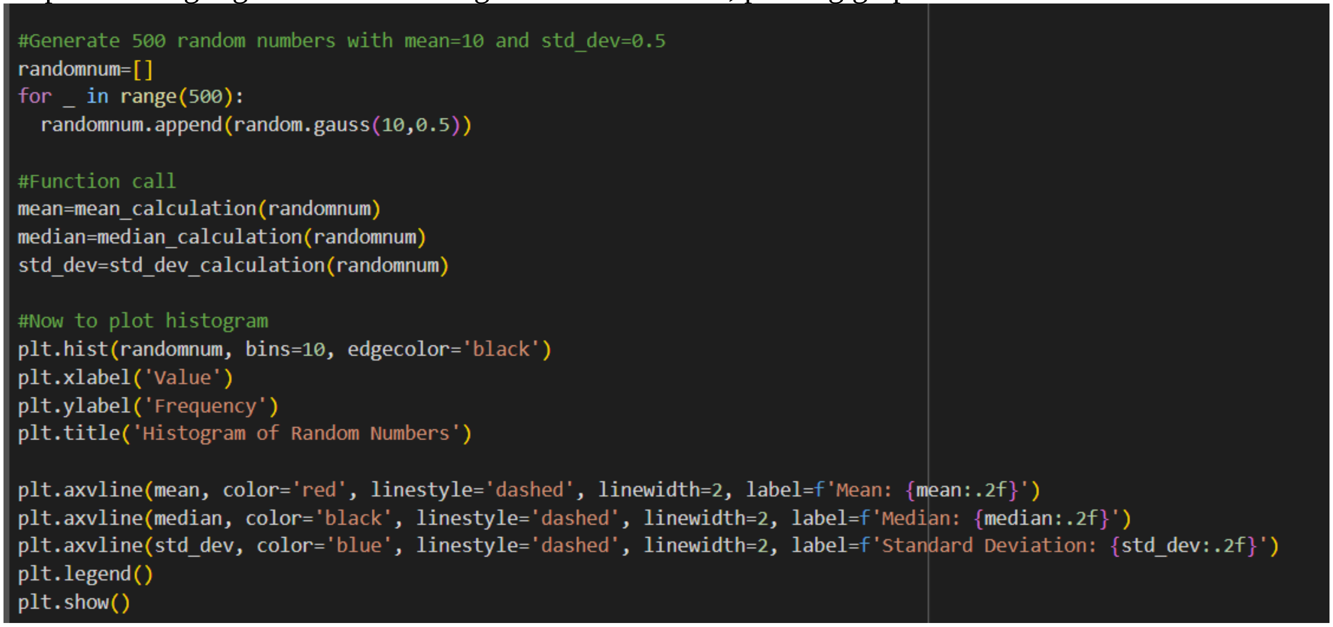
CODE:

Step 1: Defining functions first and defining calculations.



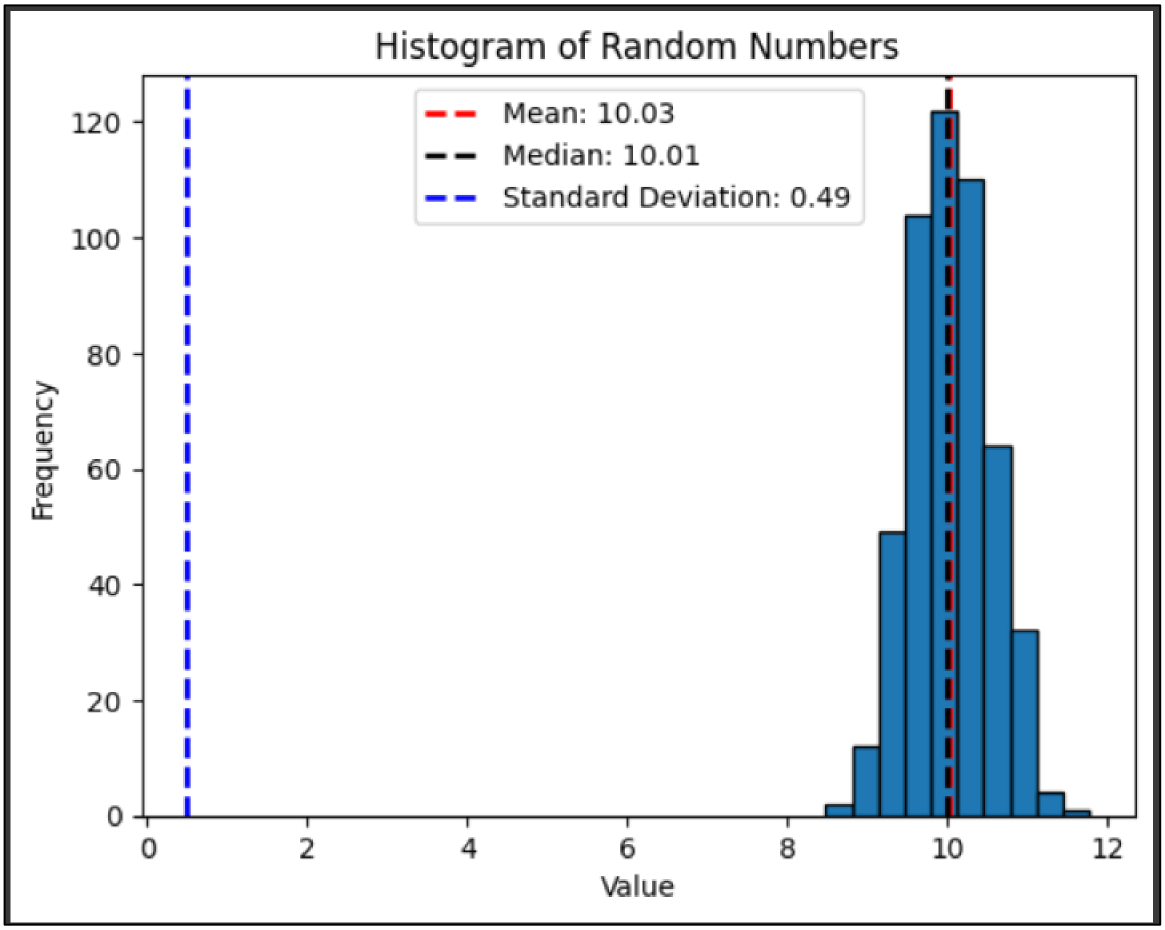
```python
import matplotlib.pyplot as plt
import random

def mean_calculation(num):
  mean = sum(num) / len(num)
  return mean

def median_calculation(num):
  ordered_numbers=sorted(num)
  length=len(ordered_numbers)
  if length % 2 == 0:
    first_num=ordered_numbers[(length // 2) -1]
    second_num=ordered_numbers[length // 2]
    median=(first_num + second_num) / 2
  else:
    median = ordered_numbers[length // 2]
  return median

def std_dev_calculation(num):
  mean=mean_calculation(num)
  distance=[]
  for x in num:
    distance.append((x - mean)**2)
  variance =  sum(distance)/len(distance)
  std_dev = variance ** 0.5
  return std_dev
```

Step 2: Passing arguments and calling the function. Also, plotting graph.



```
#Generate 500 random numbers with mean=10 and std_dev=0.5
randomnum=[]
for _ in range(500):
  randomnum.append(random.gauss(10,0.5))

#Function call
mean=mean_calculation(randomnum)
median=median_calculation(randomnum)
std_dev=std_dev_calculation(randomnum)

#Now to plot histogram
plt.hist(randomnum, bins=10, edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Random Numbers')

plt.axvline(mean, color='red', linestyle='dashed', linewidth=2, label=f'Mean: {mean:.2f}')
plt.axvline(median, color='black', linestyle='dashed', linewidth=2, label=f'Median: {median:.2f}')
plt.axvline(std_dev, color='blue', linestyle='dashed', linewidth=2, label=f'Standard Deviation: {std_dev:.2f}')
plt.legend()
plt.show()
```

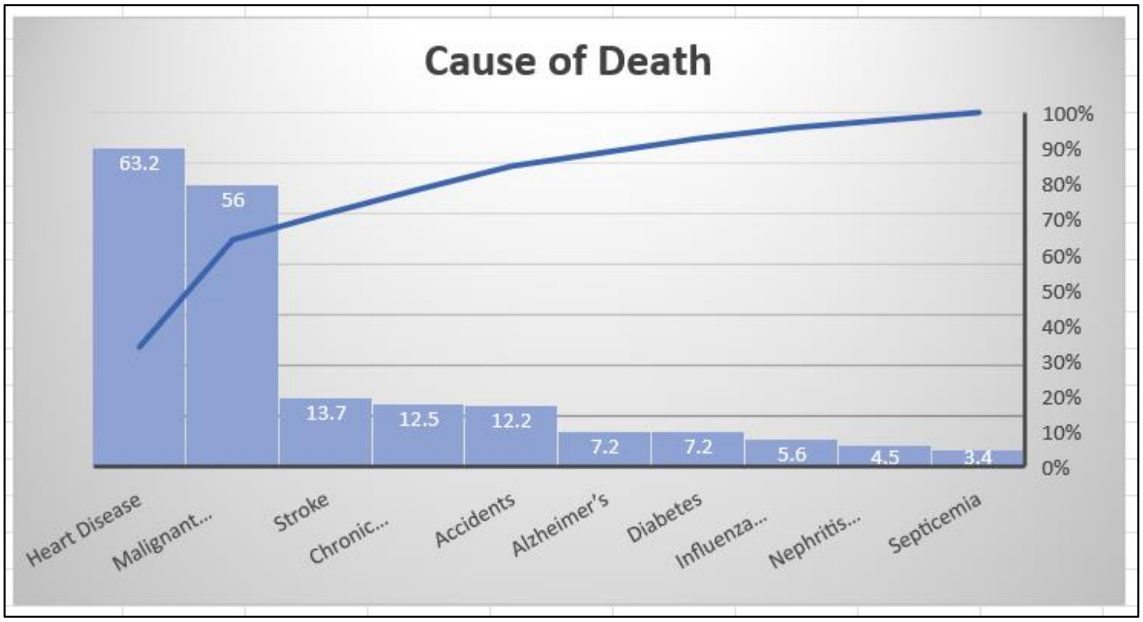
OUTPUT:



3. The 10-leading causes of death in the United States during 2006 were listed on the Centers for Disease Control and Prevention website. There are a total of 1,855,610 deaths recorded. Plot the Pareto chart in Python or Excel and explain your results.

Cause of Death	**Number (x 10,000)**
Alzheimer’s	7.2
Chronic Respiratory Disease	12.5
Diabetes	7.2
Heart Disease	63.2
Influenza/Pneumonia	5.6
Malignant Neoplasms	56.0
Accidents	12.2
Nephritis/Nephrosis	4.5
Septicemia	3.4
Stroke	13.7

ANS:

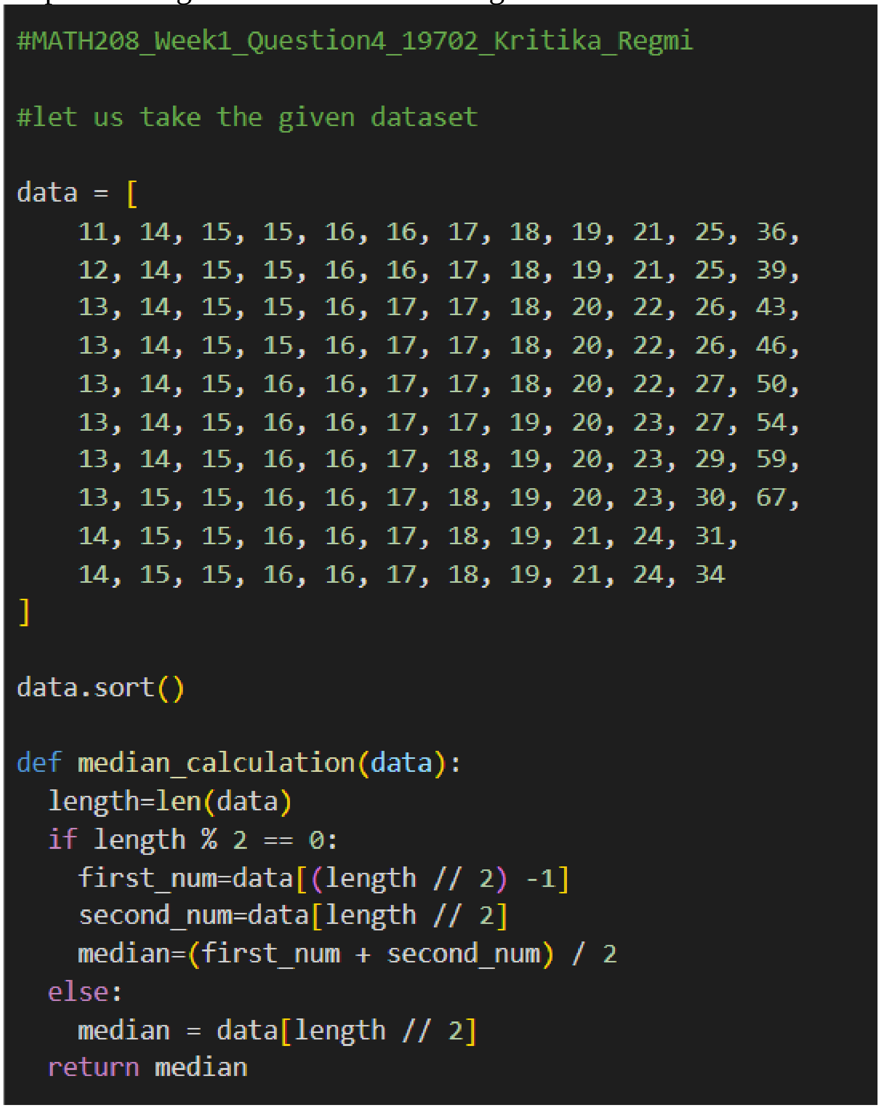


4. The following data are the ages of 118 known offenders who committed an auto theft last year in Garden City, Michigan. Write the program to find the median, the mode, Q1 and Q3, P10 and P95.

11	14	15	15	16	16	17	18	19	21	25	36
12	14	15	15	16	16	17	18	19	21	25	39
13	14	15	15	16	17	17	18	20	22	26	43
13	14	15	15	16	17	17	18	20	22	26	46
13	14	15	16	16	17	17	18	20	22	27	50
13	14	15	16	16	17	17	19	20	23	27	54
13	14	15	16	16	17	18	19	20	23	29	59
13	15	15	16	16	17	18	19	20	23	30	67
14	15	15	16	16	17	18	19	21	24	31	
14	15	15	16	16	17	18	19	21	24	34	

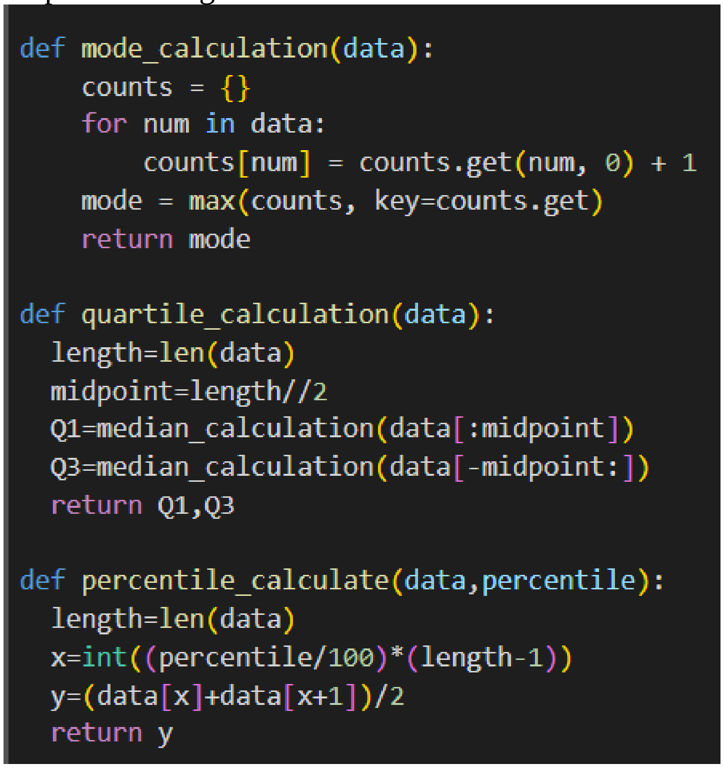
CODE:

Step 1: Taking the dataset and defining the functions.



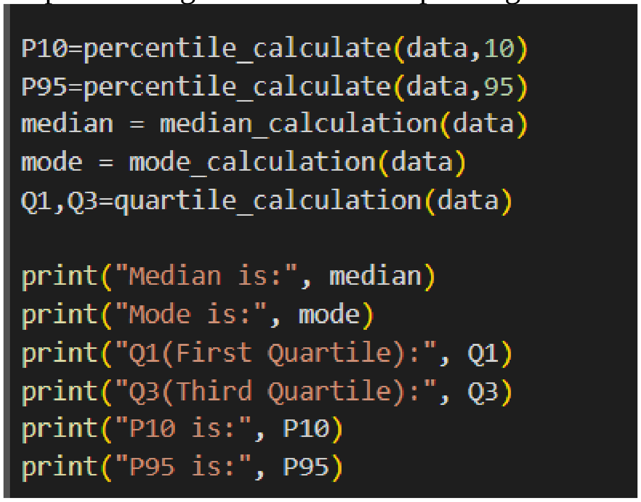
```
#MATH208_Week1_Question4_19702_Kritika_Regmi

#let us take the given dataset

data = [
    11, 14, 15, 15, 16, 16, 17, 18, 19, 21, 25, 36,
    12, 14, 15, 15, 16, 16, 17, 18, 19, 21, 25, 39,
    13, 14, 15, 15, 16, 17, 17, 18, 20, 22, 26, 43,
    13, 14, 15, 15, 16, 17, 17, 18, 20, 22, 26, 46,
    13, 14, 15, 16, 16, 17, 17, 18, 20, 22, 27, 50,
    13, 14, 15, 16, 16, 17, 17, 19, 20, 23, 27, 54,
    13, 14, 15, 16, 16, 17, 18, 19, 20, 23, 29, 59,
    13, 15, 15, 16, 16, 17, 18, 19, 20, 23, 30, 67,
    14, 15, 15, 16, 16, 17, 18, 19, 21, 24, 31,
    14, 15, 15, 16, 16, 17, 18, 19, 21, 24, 34
]

data.sort()

def median_calculation(data):
  length=len(data)
  if length % 2 == 0:
    first_num=data[(length // 2) -1]
    second_num=data[length // 2]
    median=(first_num + second_num) / 2
  else:
    median = data[length // 2]
  return median
```

Step 2: Defining the functions.



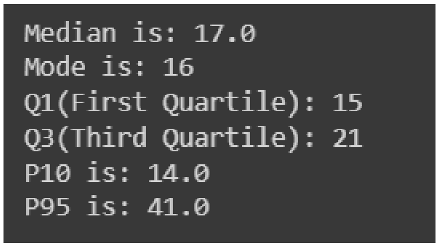
```
def mode_calculation(data):
    counts = {}
    for num in data:
        counts[num] = counts.get(num, 0) + 1
    mode = max(counts, key=counts.get)
    return mode

def quartile_calculation(data):
  length=len(data)
  midpoint=length//2
  Q1=median_calculation(data[:midpoint])
  Q3=median_calculation(data[-midpoint:])
  return Q1,Q3

def percentile_calculate(data,percentile):
  length=len(data)
  x=int((percentile/100)*(length-1))
  y=(data[x]+data[x+1])/2
  return y
```

Step 3: Calling the function and printing the values.



```
P10=percentile_calculate(data,10)
P95=percentile_calculate(data,95)
median = median_calculation(data)
mode = mode_calculation(data)
Q1,Q3=quartile_calculation(data)

print("Median is:", median)
print("Mode is:", mode)
print("Q1(First Quartile):", Q1)
print("Q3(Third Quartile):", Q3)
print("P10 is:", P10)
print("P95 is:", P95)
```

OUTPUT:



```
Median is: 17.0
Mode is: 16
Q1(First Quartile): 15
Q3(Third Quartile): 21
P10 is: 14.0
P95 is: 41.0
```