



San Francisco Bay University

MATH208 - Probability and Statistics 2023 Fall Homework #1

Due day: 9/30/2023

Instruction:

1. Homework answer sheet should contain the original questions and corresponding answers.
2. Answer sheet must be in PDF file format with Github links for the programming questions, but MS Word file can't be accepted. As follows is the answer sheet name format.
<course_id>_week<week_number>_StudentID_FirstName_LastName.pdf
3. The program name in Github must follow the format like
<course_id>_week<week_number>_q<question_number>_StudentID_FirstName_LastName
4. If the calculation in Excel is needed, the original file must be provided.
5. Show screenshot of all running results, including the system date/time.
6. The calculation process must be **printed** if needed, handwriting can't be accepted.
7. Only accept homework submission uploaded via Canvas.
8. Overdue homework submission can't be accepted.
3. Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)

For the students in Engineering School

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.

Ans:

Work PAS.ipynb - Collaboratory sweetchha/Computer_Organizati x | +

https://colab.research.google.com/drive/150rcU2AQ-kRahikDpxfib0jBFyAeBv#scrollTo=4P14pqemyR4f

PAS.ipynb File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
#Question no 1.
import math
import random
import matplotlib.pyplot as plt

def random_num():
    #random.seed(2) I used this function to test for functionality
    a = []
    sum_val = 0
    for i in range(500):
        val= random.uniform(-20,20)
        a.append(val)
        sum_val= sum_val+ val
        mean = sum_val/500
        a.sort()
    print(sum_val)
    print(a)
    if len(a)%2 == 0 :
        even_start= a[len(a)//2 -1 ]
        even_end= a[len(a)//2]
        median = (even_start+even_end)/2
    else:
        median= a[len(a)//2 ]
    half_equation= sum((x-mean)**2 for x in a)
    std = math.sqrt(half_equation/500)
    plt.hist(a, bins=10, color='pink', edgecolor='black')
    plt.xlabel('Values')
```

15°C Clear

Work PAS.ipynb - Collaboratory sweetchha/Computer_Organizati x | +

https://colab.research.google.com/drive/150rcU2AQ-kRahikDpxfib0jBFyAeBv#scrollTo=4P14pqemyR4f

PAS.ipynb File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

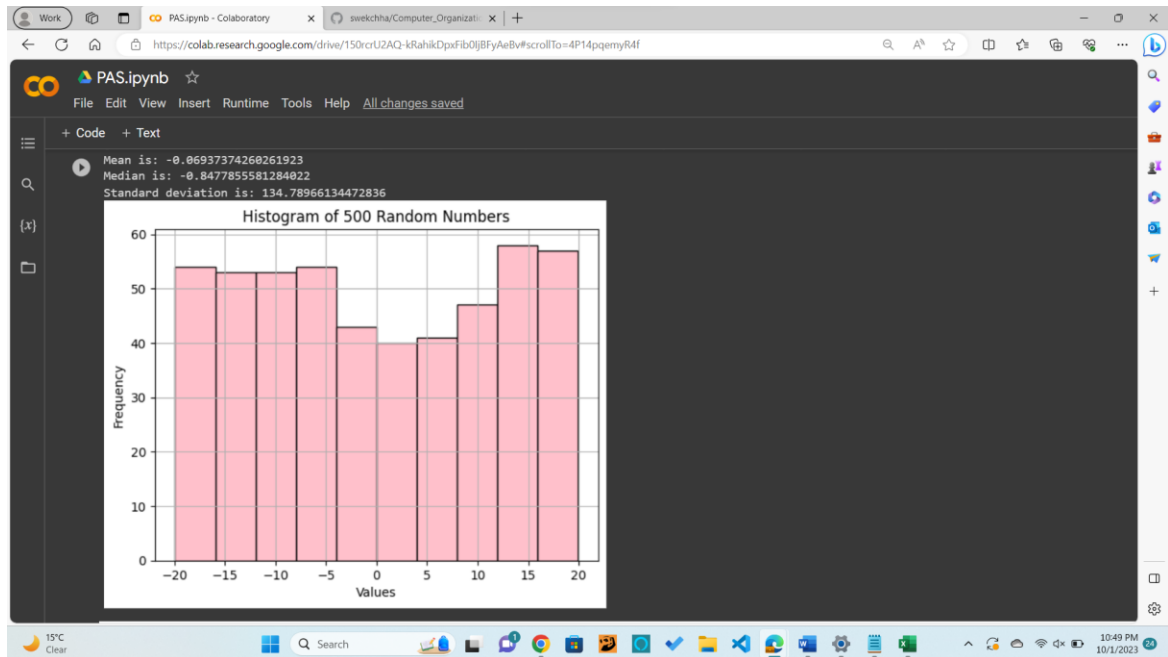
```
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Histogram of 500 Random Numbers')
plt.grid(True)

print('Mean is:',mean)
print('Median is:',median)
print('Standard deviation is:',std)

random_num()
```

-34.686871301309615
[-19.95112329963007, -19.911957323369055, -19.901845138595064, -19.85481608386209, -19.83226153868832, -19.789834275678157, -19.704676209688806, -19.6
Mean is: -0.06937374260261923
Median is: -0.8477855581284022
Standard deviation is: 134.78966134472836

15°C Clear



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.

Ans:

```
#Question no 2
import numpy as np
import math
import random
import matplotlib.pyplot as plt

random.seed(7)

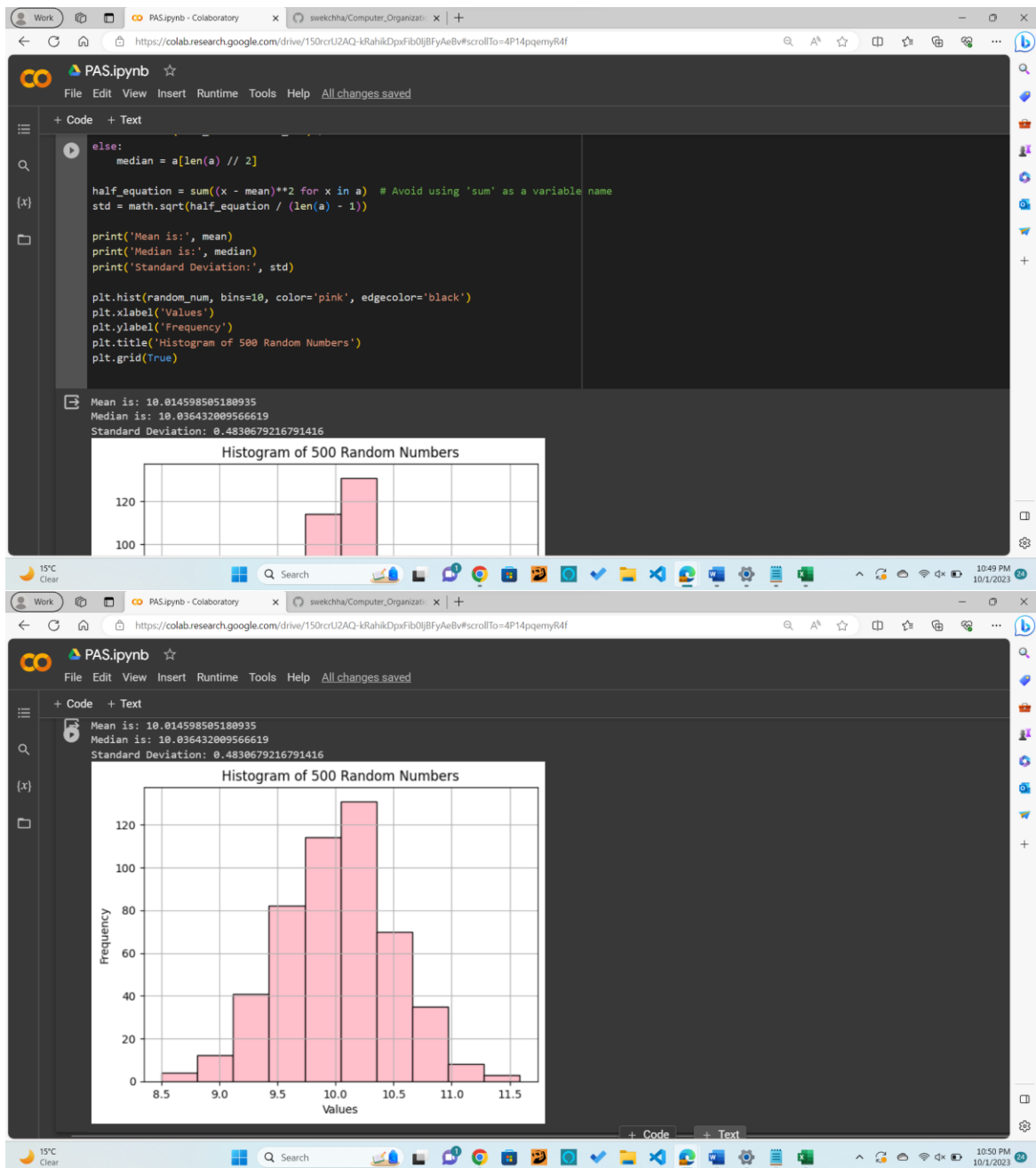
given_mean = 10
given_std = 0.5
sum_val = 0

random_num = np.random.normal(given_mean, given_std, 500)
a = list(random_num)

for i in random_num:
    sum_val = sum_val + i

mean = sum_val / len(a)
a.sort()

if len(a) % 2 == 0:
    even_start = a[len(a) // 2 - 1]
    even_end = a[len(a) // 2]
    median = (even_start + even_end) / 2
else:
    median = a[len(a) // 2]
```



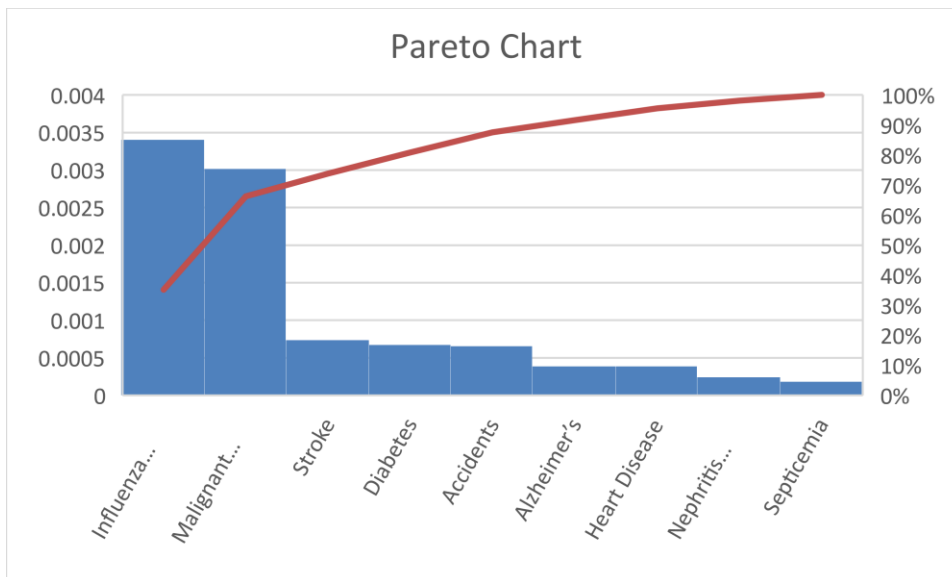
- The 10-leading causes of death in the United States during 2006 were listed on the Centers for Disease Control and Prevention website. There are a total of 1,855,610 deaths recorded. Plot the Pareto chart in Python or Excel and explain your results.

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:

Cause of death	Number(x10,000)	Cumulative percent
Alzheimer's	7.2	0.000388013
Diabetes	12.5	0.000673633
Heart Disease	7.2	0.000388013
Influenza/Pneumonia	63.2	0.003405888
Malignant Neoplasms	56	0.003017876
Accidents	12.2	0.000657466
Nephritis/Nephrosis	4.5	0.000242508
Septicemia	3.4	0.000183228
Stroke	13.7	0.000738302



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

Ans:

The image shows two screenshots of a Google Colab notebook. The first screenshot displays the code for calculating the median and mode of a dataset. The second screenshot shows the output of the code, including the median, mode, and various percentiles.

```

import numpy as np
from statistics import mode

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
]

median = np.median(data)

try:
    mode_value = mode(data)
except Error:
    mode_value = "There is no unique mode"
  
```

```

q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)

p10 = np.percentile(data, 10)
p95 = np.percentile(data, 95)

print("Median:", median)
print("Mode:", mode_value)
print("Q1 (25th percentile):", q1)
print("Q3 (75th percentile):", q3)
print("P10 (10th percentile):", p10)
print("P95 (95th percentile):", p95)
  
```

Median: 17.0
 Mode: 16
 Q1 (25th percentile): 15.0
 Q3 (75th percentile): 20.75
 P10 (10th percentile): 14.0
 P95 (95th percentile): 39.599999999999996