



Department of Computer Science and Engineering (Data Science)

B.Tech. Sem: III Subject: Statistics for Data Science

Experiment 2

Name: Rishabh Singhvi

SAP ID: 60009210206

Date:	Experiment Title: Measures of Central tendency and dispersion
Aim	To measure central tendency and dispersion of data using Python
Software	Google Colab
Implementation	<p>1. Find arithmetic mean of 20, 2, 7, 1, 34</p> <pre>In [2]: import numpy as np a=[20,2,7,1,34]</pre> <pre>In [3]: x=np.mean(a)</pre> <pre>In [4]: print(x)</pre> <p>12.8</p> <p>2. Create following matrix using multidimensional array and calculate arithmetic mean for each column, each row and considering entire data.</p> <pre>14 17 12 33 44 15 6 27 8 19 23 2 54 1 4</pre>



```
b = np.array([[14, 17,12, 33,44], [15, 6, 27,8,19],[15,6,27,8,19],[23,2,54,1,4]])
```

```
print(b)
```

```
[[14 17 12 33 44]
 [15  6 27  8 19]
 [15  6 27  8 19]
 [23  2 54  1  4]]
```

```
row_mean = np.mean(b, axis=1)
```

```
row1_mean = row_mean[0]
print("Mean of Row 1 is", row1_mean)
```

```
row2_mean = row_mean[1]
print("Mean of Row 2 is", row2_mean)
```

```
row3_mean = row_mean[2]
print("Mean of Row 3 is", row3_mean)
```

```
Mean of Row 1 is 24.0
```

```
Mean of Row 2 is 15.0
```

```
Mean of Row 3 is 15.0
```

```
column_mean = np.mean(b, axis=0)
```

```
column1_mean = column_mean[0]
print("Mean of column 1 is", column1_mean)
```

```
column2_mean = column_mean[1]
print("Mean of column 2 is", column2_mean)
```

```
column3_mean = column_mean[2]
print("Mean of column 3 is", column3_mean)
```

```
Mean of column 1 is 16.75
```

```
Mean of column 2 is 7.75
```

```
Mean of column 3 is 30.0
```

3. Find minimum value, maximum value and range for entire data, column wise and row wise.

3,7,5

8,4,3

2,4,9

```
c=np.array([[3,7,5],[8,4,3],[2,4,9]])  
print(c)
```

```
[[3 7 5]  
 [8 4 3]  
 [2 4 9]]
```

```
row_min=np.min(c,axis=1)
```

```
row1_min=row_min[0]  
print(row1_min)
```

```
row2_min=row_min[1]  
print(row2_min)
```

```
row3_min=row_min[2]  
print(row3_min)
```

```
3
```

```
3
```

```
2
```

```
column_max=np.max(c,axis=0)
```

```
column1_max=column_max[0]  
print(column1_max)
```

```
column2_max=column_max[1]  
print(column2_max)
```

```
column3_max=column_max[2]  
print(column3_max)
```

```
8
```

```
7
```

```
9
```

```
print('Range of the matrix c : ', np.ptp(c))
```

```
Range of the matrix c : 7
```

4. Find weighted average for the data given below.

Outcomes	Frequency
1	4
2	3
3	2
4	1

```
d=[1,2,3,4]  
e=[4,3,2,1]  
np.average(d,weights=e)
```

```
2.0
```



	<p>5. The speed of 13 vehicles is 99,86,87,88,111,86,103,87,94,78,77,85,86 Find mean, median and mode.</p> <pre>speed=[99,86,87,111,86,103,87,94,78,77,85,86] np.mean(speed)</pre> <p>89.91666666666667</p> <pre>np.median(speed)</pre> <p>86.5</p> <pre>from scipy import stats</pre> <pre>stats.mode(speed)</pre> <p>ModeResult(mode=array([86]), count=array([3]))</p> <p>6. Calculate geometric mean for each column, each row and considering entire data.</p> <p>1 3 27 3 4 6 7 6 3 3 6 8</p>
--	---

```
a=np.array([[1,3,27],[3,4,6],[7,6,3],[3,6,8]])
print(a)
```

```
[[ 1  3 27]
 [ 3  4  6]
 [ 7  6  3]
 [ 3  6  8]]
```

```
g_mean=gmean(a, axis = 1)
```

```
row1_g_mean=g_mean[0]
print("The geometric Mean of the first row is :",row1_g_mean)
```

The geometric Mean of the first row is : 4.326748710922226

```
row2_g_mean=g_mean[1]
print("The geometric Mean of the second row is :",row2_g_mean)
```

The geometric Mean of the second row is : 4.160167646103808

```
row3_g_mean=g_mean[2]
print("The geometric Mean of the third row is :",row3_g_mean)
```

The geometric Mean of the third row is : 5.0132979349645845

```
g_mean=gmean(a, axis = 0)
```

```
column1_g_mean=g_mean[0]
print("the geometric mean of the first column is: ",column1_g_mean)
```

```
column2_g_mean=g_mean[1]
print("the geometric mean of the second column is: ",column2_g_mean)
```

```
column3_g_mean=g_mean[2]
print("the geometric mean of the third column is: ",column3_g_mean)
```

the geometric mean of the first column is: 2.8173132472612576
 the geometric mean of the second column is: 4.559014113909556
 the geometric mean of the third column is: 7.896444077714953

7. Calculate harmonic mean for 1, 3, 5, 7, 9

```
b=[1,3,5,7,9]
hmean(b)
```

2.797513321492007

8. Calculate median for each column, each row and considering entire data.

30 65 70

80 95 10

50 90 60

```
a7=np.array([[30,65,70], [80,95,10], [50,90,60]])
print("columnwise")
print(np.median(a7,axis=0))
print("rowwise")
print(np.median(a7,axis=1))
print("columnwise")
```

```
columnwise
[50. 90. 60.]
rowwise
[65. 80. 60.]
columnwise
```

9. The number of solar heating systems available to the public is quite large, and their heat-storage capacities are quite varied. Here is a distribution of heat-storage capacity (in days) of 28 systems that were tested recently by University Laboratories, Inc.:

Days	Frequency
0-0.99	2
1-1.99	4
2-2.99	6
3-3.99	7
4-4.99	5
5-5.99	3
6-6.99	1

University Laboratories, Inc., knows that its report on the tests will be widely circulated and used as the basis for tax legislation on solar-heat allowances. It therefore wants the measures it uses to be as reflective of the data as possible.

(a) Compute the mean for these data.

```
a = [[0,0.99,2],[1,1.99,4],[2,2.99,6],[3,3.99,7],[4,4.99,5],[5,5.99,3],[6,6.99,1]]
n = 0
d = 0
a[6][1] = (a[6][1]+7)/2
a[1][0] = -0.0
for i in range (0,6):
    a[i][1] = (a[i][1] + a[i+1][0])/2
    a[i+1][0] = a[i][1]

print(a)

for i in range(0,7):
    mi = (a[i][0]+a[i][1])/2
    n = n + a[i][2]*mi
    d = d + a[i][2]

mean = n/d
print(mean)
```

```
[[0, 0.495, 2], [0.495, 1.995, 4], [1.995, 2.995, 6], [2.995, 3.995, 7], [3.995, 4.995, 5], [4.995, 5.995, 3], [5.995, 6.995, 1]]
3.2273214285714285
```

(b) Compute the mode for these data.

```
a = [[0,0.99,2],[1,1.99,4],[2,2.99,6],[3,3.99,7],[4,4.99,5],[5,5.99,3],[6,6.99,1]]
a[6][1] = (a[6][1]+7)/2
for i in range (0,6):
    a[i][1] = (a[i][1] + a[i+1][0])/2
    a[i+1][0] = a[i][1]

max = -1;
for i in range (0,7):
    if(a[i][2]>max):
        max = a[i][2]
        index = i

f0 = a[index-1][2]
f1 = a[index][2]
f2 = a[index+1][2]
L = a[index][0]
I = (a[index][1]-a[index][0])

mode = L + ((f1-f0)/(2*f1-f0-f2))*I
print(mode)
```

```
3.3283333333333336
```

(c) Compute the median for these data.

```

a = [[0,0.99,2],[1,1.99,4],[2,2.99,6],[3,3.99,7],[4,4.99,5],[5,5.99,3],[6,6.99,1]]

a[6][1] = (a[6][1]+7)/2
for i in range (0,6):
    a[i][1] = (a[i][1] + a[i+1][0])/2
    a[i+1][0] = a[i][1]

cf = 0
for i in range (0,7):
    cf = cf + a[i][2]
    a[i].append(cf)

N = a[6][3]

for i in range(0,7):
    if(a[i][3]>N/2):
        index = i
        break

L = a[index][0]
cf = a[index-1][3]
f = a[index][2]
I = (a[index][1]-a[index][0])

median = L + (((N/2)-cf)/f)*I
print(median)

```

3.2807142857142857

(d) Select the answer among parts (a), (b), and (c) that best reflects the central tendency of the test data and justify your choice.

10. Write a function for calculating percentile and determine 30, 50, 75 and 90 percentiles for the following data.

30,40,72,83,25,10,50,90,60,15,5,9,34,23,67,80,67,45

```

a=[30,40,72,82,25,10,50,90,60,15,5,9,34,23,67,80,67,45]
print("For 30")
print(np.percentile(a,30,axis=0))
print("For 50")
print(np.percentile(a,50,axis=0))
print("For 75")
print(np.percentile(a,75,axis=0))
print("For 90")
print(np.percentile(a,90,axis=0))

```

For 30
25.5
For 50
42.5
For 75
67.0
For 90
80.6

11. The numbers of apartments in 27 apartment complexes in Cary, North Carolina, are given below. 91 79 66 98 127 139 154 147 192 88 97 92 87 142 127 184 145 162 95 89 86 98 145 129 149 158 241

(a) Construct a frequency distribution using intervals 66–87, 88–109, 220–241.

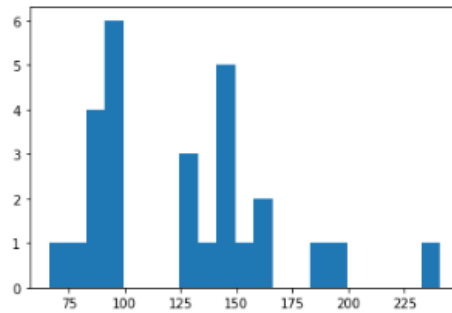
(b) Estimate the modal value

(c) Compute the mean of the raw data.

(d) Compare your, answers in parts (b) and (c) and comment on which of the two is the better measure of central tendency of these data and why.

```
z=[91,79,66,98,127,139,154,147,192,88,97,92,87,142,127,184,145,162,95,89,86,98,145,129,149,158,241]
plt.hist(z,21,range=(np.min(z),np.max(z)))
z1=stats.mode(z)
print(z1)
np.mean(z)
```

```
ModeResult(mode=array([98]), count=array([2]))
126.18518518518519
```



Mean is the better measure of central tendency.

12. There are 39 plants in the garden. A few plants were selected randomly and their heights in cm were recorded as follows: 51, 38, 79, 46, 57. Calculate the standard deviation of their heights.

```
a=[51, 38, 79, 46, 57]
print(np.std(a))
```

```
13.876599006961325
```

13. The Casual Life Insurance Company is considering purchasing a new fleet of company cars. The financial department's director, Tom Dawkins, sampled 40 employees to determine the number of miles each drove over a 1-year period. The results of the study follow. Calculate the range and interquartile range.

3,600	4,200	4,700	4,900	5,300	5,700	6,700	7,300
7,700	8,100	8,300	8,400	8,700	8,700	8,900	9,300
9,500	9,500	9,700	10,000	10,300	10,500	10,700	10,800
11,000	11,300	11,300	11,800	12,100	12,700	12,900	13,100
13,500	13,800	14,600	14,900	16,300	17,200	18,500	20,300

```
a=[3600,4200,4700,4900,5300,5700,6700,7300,7700,8100,8300,8400,8700,8900,9300,9500,9500,9700,10000,10300,10500,10700,10800,11000,11300,11300,11800,12100,12700,12900,13100,13500,13800,14600,14900,16300,17200,18500,20300]
print("Range:")
print((np.ptp(a,axis=0)))
print("Interquartile:")
b=np.percentile(a,25,axis=0)
c=np.percentile(a,75,axis=0)
print(c-b)
```

```
Range:
16700
Interquartile:
4600.0
```

14. The head chef of The Flying Taco has just received two dozen tomatoes from her supplier, but she isn't ready to accept them. She knows from the invoice that the average weight of a tomato is 7.5 ounces, but she insists that all be of uniform weight. She will accept them only if the average weight is

7.5 ounces and the standard deviation is less than 0.5 ounce. Here are the weights of the tomatoes

6.3 7.2 7.3 8.1 7.8 6.8 7.5 7.8 7.2 7.5 8.1 8.2

8.0 7.4 7.6 7.7 7.6 7.4 7.5 8.4 7.4 7.6 6.2 7.4

What is the chef's decision and why?

```
a=[6.3,7.2,7.3,8.1,7.8,6.8,7.5,7.8,7.2,7.5,8.1,8.2,8.0,7.4,7.6,7.7,7.6,7.4,7.5,8.4,
print("Average weight")
print(np.average(a))
print("Standard deviation")
print(np.std(a))
print("Since standard deviation is more than 0.5 she will not accept tomatoes")
```

Average weight

7.5

Standard deviation

0.5163977794943222

Since standard deviation is more than 0.5 she will not accept tomatoes

15. A company is considering employing one of two training programs. Two groups were trained for the same task. Group 1 was trained by program A; group 2, by program B. For the first group, the times required to train the employees had an average of 32.11 hours and a variance of 68.09. In the second group, the average was 19.75 hours and the variance was 71.14. Which training program has less relative variability in its performance?

Ans: Group 1 has less relative variability in its performance

16. Here is a frequency distribution of the weight of 150 people who used a ski lift a certain day. Construct a histogram for these data.

Class	Frequency	Class	Frequency
75–89	10	150–164	23
90–104	11	165–179	9
105–119	23	180–194	9
120–134	26	195–209	6
135–149	31	210–224	2

(a) What can you see from the histogram about the data that was not immediately apparent from the frequency distribution?

The histogram shows that the lower tail of the distribution is fatter than the upper tail.

(b) If each ski lift chair holds two people but is limited in total safe weight capacity to 400 pounds, what can the operator do to maximize the people capacity of the ski lift without exceeding the safe weight capacity of a chair? Do the data support your proposal?

There are very few people weighing 180 pounds or above.

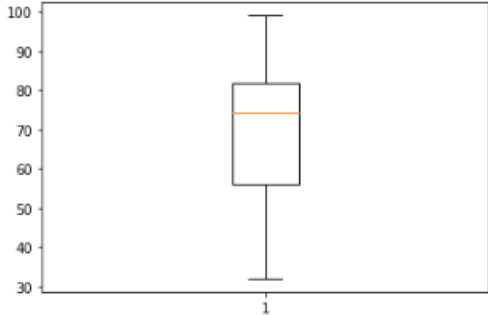
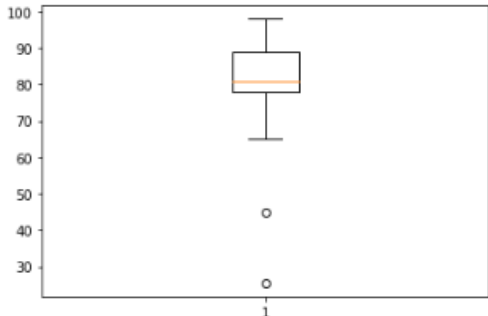
So that the operator affords to pair each person who appears to be heavy person With low weight person. This can be done without exceeding the safe weight capacity of a chair.



	<p>17. Test scores for a college statistics class held during the day are: 99 56 78 55.5 32 90 80 81 56 59 45 77 84.5 84 70 72 68 32 79 90</p> <p>Test scores for a college statistics class held during the evening are: 98 78 68 83 81 89 88 76 65 45 98 90 80 84.5 85 79 78 98 90 79 81 25.5</p> <ol style="list-style-type: none">Find the smallest and largest values, the median, and the first and third quartile for the day class.Find the smallest and largest values, the median, and the first and third quartile for the night class.For each data set, what percentage of the data is between the smallest value and the first quartile? the first quartile and the median? the median and the third quartile? the third quartile and the largest value? What percentage of the data is between the first quartile and the largest value?Create a box plot for each set of data. Use one number line for both box plots.Which box plot has the widest spread for the middle 50% of the data (the data between the first and third quartiles)? What does this mean for that set of data in comparison to the other set of data.
--	--



	<pre>a=[99,56,78,55.5,32,90,80,81,56,59,45,77,84.5,84,70,72,68,32,79,90] b=[98,78,68,83,81,89,88,76,65,45,98,90,80,84.5,85,79,78,98,90,79,81,25.5] print("Day") print("Large value") print(np.amax(a,axis=0)) print("Small value") print(np.amin(a,axis=0)) print("Median") print(np.percentile(a,50,axis=0)) print("First Quartile") print(np.percentile(a,25,axis=0)) print("Third quartile") print(np.percentile(a,25,axis=0)) print('-----') print("Night") print("Large value") print(np.amax(b,axis=0)) print("Small value") print(np.amin(b,axis=0)) print("Median") print(np.percentile(b,50,axis=0)) print("First Quartile") print(np.percentile(b,25,axis=0)) print("Third quartile") print(np.percentile(b,25,axis=0)) print('-----') print("for day") plt.boxplot(a)</pre>
	<pre>Day Large value 99.0 Small value 32.0 Median 74.5 First Quartile 56.0 Third quartile 56.0 ----- Night Large value 98.0 Small value 25.5 Median 81.0 First Quartile 78.0 Third quartile 78.0 -----</pre>

	<pre>for day</pre> <pre>Out[25]: {'whiskers': [<matplotlib.lines.Line2D at 0x11dafabeb50>, <matplotlib.lines.Line2D at 0x11dafabed30>], 'caps': [<matplotlib.lines.Line2D at 0x11daface040>, <matplotlib.lines.Line2D at 0x11daface310>], 'boxes': [<matplotlib.lines.Line2D at 0x11dafabe880>], 'medians': [<matplotlib.lines.Line2D at 0x11daface5e0>], 'fliers': [<matplotlib.lines.Line2D at 0x11daface8b0>], 'means': []}</pre>  <pre>In [26]: b=[98,78,68,83,81,89,88,76,65,45,98,90,80,84.5,85,79,78,98,90,79,81,25.5] print("for Night") plt.boxplot(b)</pre> <pre>for Night</pre> <pre>Out[26]: {'whiskers': [<matplotlib.lines.Line2D at 0x11dafb23910>, <matplotlib.lines.Line2D at 0x11dafb23be0>], 'caps': [<matplotlib.lines.Line2D at 0x11dafb23eb0>, <matplotlib.lines.Line2D at 0x11dafb321c0>], 'boxes': [<matplotlib.lines.Line2D at 0x11dafb23610>], 'medians': [<matplotlib.lines.Line2D at 0x11dafb32490>], 'fliers': [<matplotlib.lines.Line2D at 0x11dafb32760>], 'means': []}</pre> 
Conclusion	

Signature of Faculty