Aayush Shah
19BCE245
24 March 2021

Practical 4

Gaussian probability density function

• **Definition :** Write a program which reads *m* (scan *m* from the user) real values in 2 variables (*m/2* in each) and computes and displays their (i) mean (ii) variance (iii) standard deviation. The program should also output z-score for each value. The program should furthermore display (i) scatter plot (ii) histograms for these variables for *n* bins where *n* is read from the user and (iii) density plot for these variables assuming Gaussian as the probability density function.

• **Code :**

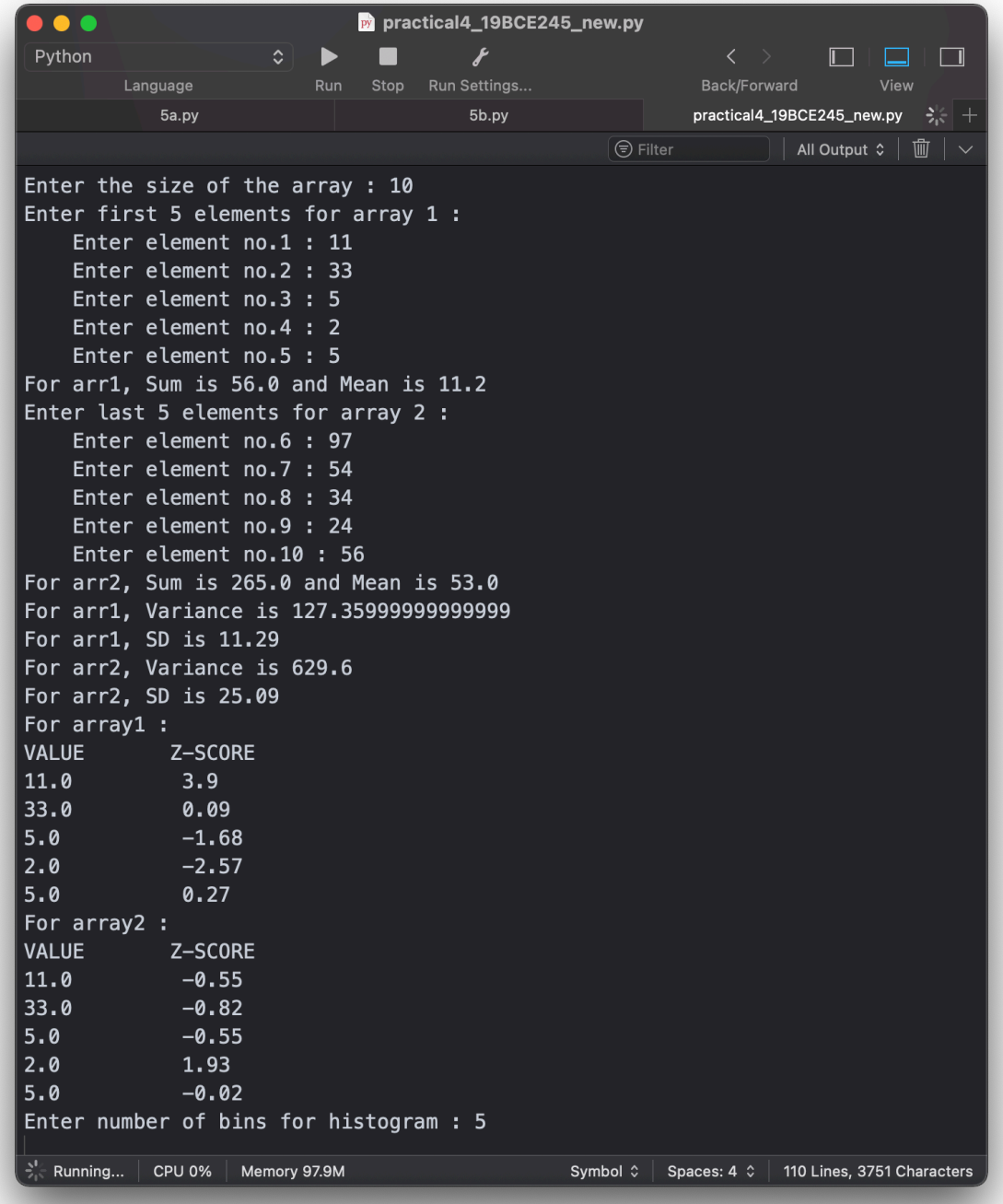
```
1. import math
2. import matplotlib.pyplot as plt
3.
4. def gaussFunction(s, x, m):
5.     return (1 / (s * math.sqrt(2 * math.pi))) * math.exp((-1
   / 2) * math.pow((x - m) / s, 2))
6.
7. if __name__ == "__main__":
8.     arr1 = []          #for storing 1st list
9.     arr2 = []          #for storing 2nd list
10.
11.     total_size = int(input("Enter the size of the array :
   "))
12.     size = int(total_size/2)
13.
14.     temp_number = float()
15.
16.     print("Enter first",size,"elements for array 1 : ")
17.     for i in range(size):
```

```
18.         temp_number = float(input("\tEnter element no.{} :
    ".format(i+1)))
19.         arr1.append(temp_number)
20.
21.     #calculating sum of all elements and mean
22.     sum_of_arr1 = sum(arr1)
23.     mean_of_arr1 = float(sum_of_arr1/size)
24.     print("For arr1, Sum is",sum_of_arr1,"and Mean
    is",mean_of_arr1)
25.
26.     print("Enter last",size,"elements for array 2 : ")
27.     for i in range(size):
28.         temp_number = float(input("\tEnter element no.{} :
    ".format(size + i + 1)))
29.         arr2.append(temp_number)
30.
31.     sum_of_arr2 = sum(arr2)
32.     mean_of_arr2 = float(sum_of_arr2/size)
33.     print("For arr2, Sum is",sum_of_arr2,"and Mean
    is",mean_of_arr2)
34.
35.     #calculating variance and SD
36.     arr1_diff = []
37.     arr2_diff = []
38.     sq_sum_of_arr1 = 0
39.     sq_sum_of_arr2 = 0
40.
41.     for i in range(size):
42.         arr1_diff.append(float(arr1[i]-mean_of_arr1))
43.         sq_sum_of_arr1 += float(arr1_diff[i]**2)
44.
45.     variance1 = float(sq_sum_of_arr1/size)
46.     sd1 = math.sqrt(variance1)
47.     print("For arr1, Variance is",variance1)
48.     print("For arr1, SD is",round(sd1,2))
49.
50.     for i in range(size):
51.         arr2_diff.append(float(arr2[i]-mean_of_arr2))
52.         sq_sum_of_arr2 += float(arr2_diff[i]**2)
53.
54.     variance2 = float(sq_sum_of_arr2/size)
55.     sd2 = math.sqrt(variance2)
56.     print("For arr2, Variance is",variance2)
```

```
57.     print("For arr2, SD is",round(sd2,2))
58.
59.     z = []      #for storing z scores
60.
61.     #calculating z-score
62.     for i in range(size):
63.         z.insert(i, (arr1[i] - mean_of_arr1) / sd1)
64.         z.insert(i, (arr2[i] - mean_of_arr2) / sd1)
65.
66.     print("For array1 : ")
67.     print("VALUE\t\tZ-SCORE")
68.     for index in range(size):
69.         print(arr1[index],"\t\t",round(z[index],2))
70.
71.     print("For array2 : ")
72.     print("VALUE\t\tZ-SCORE")
73.     for index in range(size):
74.         print(arr1[index],"\t\t",round(z[size + index],2))
75.
76.     arr1.sort()
77.     arr2.sort()
78.
79.     g1, g2 = [], []
80.     for i in range(size):
81.         g1.append(gaussFunction(sd1, arr1[i],
  mean_of_arr1))
82.         g2.append(gaussFunction(sd2, arr2[i],
  mean_of_arr2))
83.
84.     #Displaying histogram
85.     b = int(input("Enter number of bins for histogram : "))
86.     plt.subplot(2, 2, 1)
87.     plt.title("HISTOGRAM")
88.     arr = arr1 + arr2
89.     plt.hist(arr,bins=b, color='r')
90.
91.     #Displaying scatterplot
92.     plt.subplot(2, 2, 2)
93.     plt.title("SCATTERPLOT")
94.     plt.scatter(arr[:total_size // 2], arr[total_size //
  2:] )
95.
96.     #displaying gaussian fx. for first half of the array
```

```
97.     plt.subplot(2, 2, 3)
98.     plt.title("GAUSSSIAN FUNCTION 1")
99.     plt.plot(arr1, g1, color='y')
100.
101.     #displaying gaussian fx. for second half of the array
102.     plt.subplot(2, 2, 4)
103.     plt.title("GAUSSSIAN FUNCTION 2")
104.     plt.plot(arr2, g2, color='g')
105.
106.     plt.tight_layout()          #improves subplot spacings
107.     plt.show()
```

- **Sample I/O :**



- **Graph generated :**