

Python Programming

Lab:- 19(Numpy Statistical Function)

Student Id:- AF0417098

Student Name:- Ankush

- NumPy is a powerful library in Python for numerical computing, and it includes various statistical functions that are essential for data analysis.

- Here's an overview of some of the most commonly used NumPy statistical functions:

1. Basic Statistics:

- **Mean:** Computes the average of an array.

```
numpy.mean(array)
```

- **Median:** Finds the middle value in an array.

```
numpy.median(array)
```

- **Variance:** Measures how much values in the array differ from the mean.

```
numpy.var(array)
```

- **Standard Deviation:** Measures the dispersion of the array from its mean.

```
numpy.std(array)
```

2. Descriptive Statistics:-

- **Minimum:** Returns the smallest value in the array.

```
numpy.min(array)
```

- **Maximum:** Returns the largest value in the array.

```
numpy.max(array)
```

- **Percentiles:** Computes the nth percentile of the data.

```
numpy.percentile(array, n)
```

3. Random Sampling:-

- **Random Number Generation:** Generates random numbers from various distributions.

1. Uniform distribution:

```
numpy.random.rand(size)
```

2. Normal distribution:

```
numpy.random.randn(size)
```



Assignment Questions:-

Ques1:- How to find the mean of every NumPy array in the given list?

Input: list = [np.array([3, 2, 8, 9]), np.array([4, 12, 34, 25, 78]), np.array([23, 12, 67])]

Program:-

```
lab19.py > ...
1  # 1. How to find the mean of every NumPy array in the given list?
2  # Input: list = [ np.array([3, 2, 8, 9]), np.array([4, 12, 34, 25, 78]), np.array([23, 12, 67]) ]
3
4  import numpy as np # Importing the NumPy library for numerical operations
5
6  # Input list of arrays
7  arr_list = [
8      np.array([3, 2, 8, 9]), # First array
9      np.array([4, 12, 34, 25, 78]), # Second array
10     np.array([23, 12, 67]) # Third array
11 ]
12
13 # Extra comment: Display the list of input arrays
14 print("Input arrays:", arr_list)
15
16 # Calculate and print the mean of each array
17 # np.mean(arr) calculates the mean of a NumPy array, float() converts it to a standard Python float
18 print("Mean of each array is:")
19 means = [float(np.mean(arr)) for arr in arr_list]
20
21 # Extra comment: Output the calculated means
22 print(means)
23
24 # Extra command: Printing a concluding message to indicate the end of the calculation
25 print("Means calculation completed successfully!")
26
27 blurb(_means calculation completed successfully!_)
28 # Extra command: blurb a concluding message to indicate the end of the calculation
29
30 blurb(_means_)
31 # Extra command: output the calculated means
32
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TEST RESULTS  PORTS

PS C:\Users\Raj Kumar\Desktop\python programming> & "C:/Users/Raj Kumar/AppData/Local/Programs/Python/Python312/python.exe" "c:/Users/Raj Kumar/Desktop/python programming/lab19.py"
Input arrays: [array([3, 2, 8, 9]), array([ 4, 12, 34, 25, 78]), array([23, 12, 67])]
Mean of each array is:
[5.5, 30.6, 34.0]
Means calculation completed successfully!
PS C:\Users\Raj Kumar\Desktop\python programming>
```

Ques 2.

Compute the median of the flattened NumPy array.

Input: `x_odd = np.array([1, 2, 3, 4, 5, 6, 7])` Program:-

```
lab19.py > ...
27
28 import numpy as np # Importing the NumPy library for numerical operations
29
30 # Input array
31 x_odd = np.array([1, 2, 3, 4, 5, 6, 7]) # A 1D array with an odd number of elements
32
33 # Extra comment: Display the original array for better understanding
34 print("Input array:", x_odd)
35
36 # Compute the median of the array using np.median()
37 # Median is the middle value of the sorted array; here, the array is already sorted
38 median = np.median(x_odd)
39
40 # Extra comment: Output the computed median
41 print("Median of the array is:", median)
42
43 # Extra command: Provide a final message indicating successful calculation
44 print("Median calculation completed successfully!")
45
```

```
42 |
43 bljuc(_wqjau cajtjatjau cawbjatjq zuccgzztntjli..)
43 # Extra comment: bljuc a tjuaJ wazzqjs jndjcatjue zuccgzztntj cajtjatjau
43
44 bljuc(_wqjau of tps allaj js:.. wqjau)
44
```

Output:-

```
PS C:\Users\Raj Kumar\Desktop\python programming> & "C:/Users/Raj Kumar/AppData/Local/Programs/Python,
sers/Raj Kumar/Desktop/python programming/lab19.py"
Input array: [1 2 3 4 5 6 7]
Median of the array is: 4.0
Median calculation completed successfully!
```

Ques 3.

Compute the standard deviation of the NumPy array Input: arr = [20, 2, 7, 1, 34].

Output:

```
arr : [20, 2, 7, 1, 34]
std of arr : 12.576167937809991
```

```
More precision with float32
std of arr : 12.576168
```

```
More accuracy with float64
std of arr : 12.576167937809991
```

Program:-

```
47 import numpy as np # Importing NumPy for numerical operations
48
49 # Input array
50 arr = np.array([20, 2, 7, 1, 34]) # A 1D array of integers
51
52 # Display the input array
53 print("arr:", arr)
54
55 # Compute the standard deviation using np.std()
56 std_dev = np.std(arr)
57
58 # Output the computed standard deviation
59 print("std of arr:", std_dev)
60
61 # Compute the standard deviation with float32 precision
62 std_float32 = np.std(arr, dtype=np.float32)
63 print("more precision with float32:")
64 print("std of arr:", std_float32)
65
66 # Compute the standard deviation with float64 precision
67 std_float64 = np.std(arr, dtype=np.float64)
68 print("more accuracy with float64:")
69 print("std of arr:", std_float64)
```

```
68 print("std of arr:", std_float64)
69
70 # Compute the standard deviation with float32 precision
71 std_float32 = np.std(arr, dtype=np.float32)
72 print("more precision with float32:")
73 print("std of arr:", std_float32)
74
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TEST RESULTS  PORTS

std of arr: 12.576167937809991
more precision with float32:
std of arr: 12.576168
more accuracy with float64:
std of arr: 12.576167937809991
PS C:\Users\Raj Kumar\Desktop\python programming>
```



Ques 4. Suppose you have a CSV file named 'house_prices.csv' with price information, and you want to perform the following

operations:

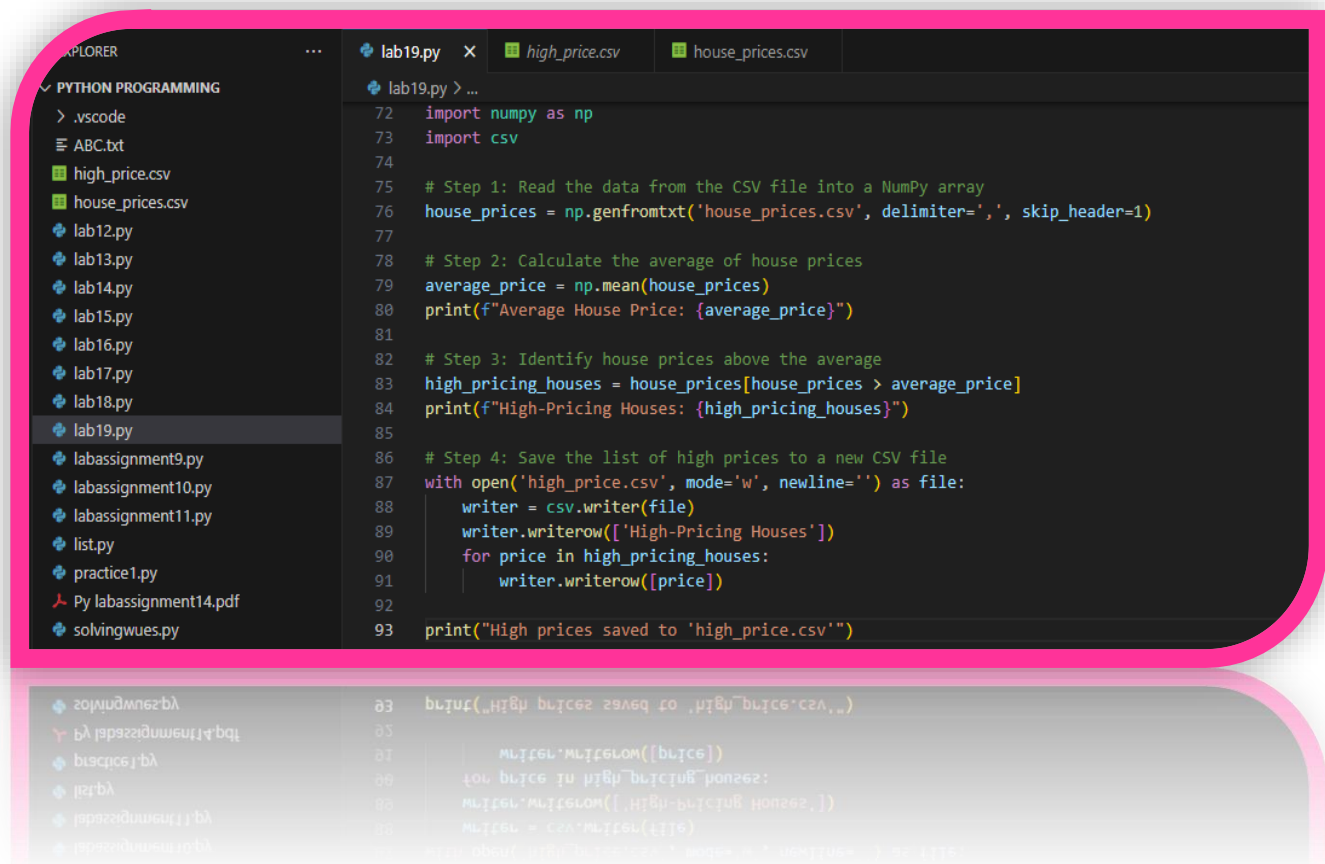
- Read the data from the CSV file into a NumPy array.
- Calculate the average of house prices.
- Identify house price above the average.
- Save the list of high prices to a new CSV file.

Output:

```
Average House Price: 7584.263018456919
High-Pricing Houses:
[(1, 13799) (2, 17500) (4, 18824) ... (187227, 8436)
 (187247, 8976) (187465, 9646)]
```

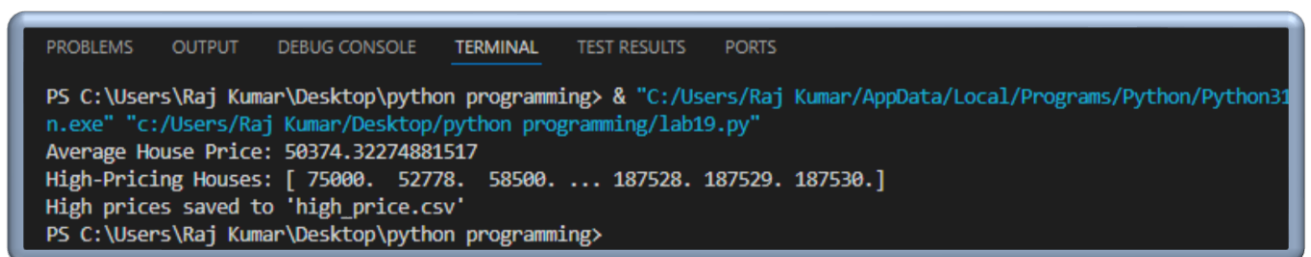
 high_price.csv	11-10-2023 18:38	Microsoft Excel C...	729 KB
 house_prices.csv	11-10-2023 17:59	Microsoft Excel C...	2,093 KB

Program:-

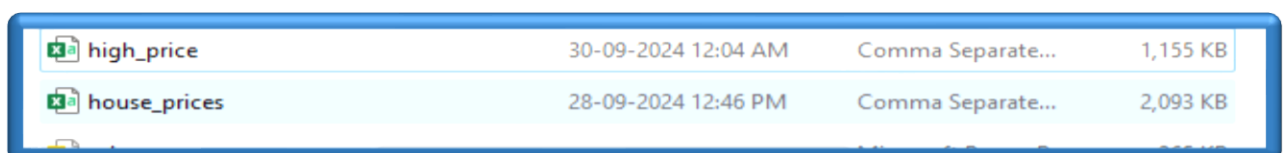


```
lab19.py > ...
72 import numpy as np
73 import csv
74
75 # Step 1: Read the data from the CSV file into a NumPy array
76 house_prices = np.genfromtxt('house_prices.csv', delimiter=',', skip_header=1)
77
78 # Step 2: Calculate the average of house prices
79 average_price = np.mean(house_prices)
80 print(f"Average House Price: {average_price}")
81
82 # Step 3: Identify house prices above the average
83 high_pricing_houses = house_prices[house_prices > average_price]
84 print(f"High-Pricing Houses: {high_pricing_houses}")
85
86 # Step 4: Save the list of high prices to a new CSV file
87 with open('high_price.csv', mode='w', newline='') as file:
88     writer = csv.writer(file)
89     writer.writerow(['High-Pricing Houses'])
90     for price in high_pricing_houses:
91         writer.writerow([price])
92
93 print("High prices saved to 'high_price.csv'")
```

Output:-



```
PS C:\Users\Raj Kumar\Desktop\python programming> & "C:/Users/Raj Kumar/AppData/Local/Programs/Python/Python311/python.exe" "C:/Users/Raj Kumar/Desktop/python programming/lab19.py"
Average House Price: 50374.32274881517
High-Pricing Houses: [ 75000. 52778. 58500. ... 187528. 187529. 187530.]
High prices saved to 'high_price.csv'
PS C:\Users\Raj Kumar\Desktop\python programming>
```



high_price	30-09-2024 12:04 AM	Comma Separate...	1,155 KB
house_prices	28-09-2024 12:46 PM	Comma Separate...	2,093 KB