

06. a) program to find mean, median and mode for a given set of numbers in a list with userdefined functions.

```
def my_mean(Sample):
    return sum(Sample) / len(Sample)
my_mean([1, 2, 3, 4, 5, 6])
```

```
def my_median(Sample):
    n = len(Sample)
    index = n // 2
    if n % 2:
        return sorted(Sample)[index]
    else:
        return sum(sorted(Sample)[index - 1 : index + 1]) / 2
my_median([3, 5, 1, 4, 2])
```

~~from collections import Counter~~

```
def my_mode(Sample):
    c = Counter(Sample)
    return [k for k, v in c.items() if v == c.most_common(1)[0][1]]
my_mode([4, 1, 2, 2, 3, 4, 5])
```

b) program to define a function that can find all the duplicate values in a list.

```
def duplicate_list(input_list):  
    new_dict, new_list = {}, []  
    for i in input_list:  
        if not i in new_dict:  
            new_dict[i] = 1  
        else:  
            new_dict[i] += 1  
    for key, value in new_dict.items():  
        if value > 1:  
            new_list.append(key)  
    return new_list  
if __name__ == '__main__':  
    input_list = [1, 2, 1, 2, 3, 5, 6, 5, 9, 8, 9]  
    print(duplicate_list(input_list))
```

10  
10  
✓  
27/11/23



6(a)  
o/p: 3.5

o/p: 2

o/p: ~~[4,2]~~

6(b)

O/p: [1, 2, 5, 9]

7. class write a program to demonstrate working of ATM using oops concept.

Class BankATM:

bal = 5000

def login(self, pin):

if (pin == 2345):

return True

else:

return print("wrong password")

def credit(self, amt):

self.bal += amt

def debit(self, amt):

self.bal -= amt

def show(self):

print("The current balance", +str(self.bal))

obj = BankATM()

flag = False

for i in range(1, 4):

if obj.login(int(input("enter the pin:"))):

flag = True

break

if flag:

while True:

o = input("c for credit d for debit b for bal  
e for exit\n")

if o == 'c' or o == 'C':

Teacher's Signature : \_\_\_\_\_



```
obj.credit(int(input("enter the amount")))
print("after credited")
obj.show()
elif o == 'd' or o == 'D':
    amt = int(input("enter the amount"))
    if amt < obj.bal:
        obj.debit(amt)
        print("the amount is:")
        obj.show()
    else:
        print("insufficient balance")
        obj.show()
elif o == 'b' or o == 'B':
    obj.show()
elif o == 'e' or o == 'E':
    break
else:
    print("pincode attempt is completed")
```

8  
10/10

O/p: 1. Enter the pin: 2345

c for credit

d for debit

b for balance

e for exit

Enter the amount: 1000

After credited ✓

the current balance: 6000

d → 500

Enter the amount 500

the amount is 5500

b → the current balance is 5500

e → exit ✓

Exit !! Visit again.

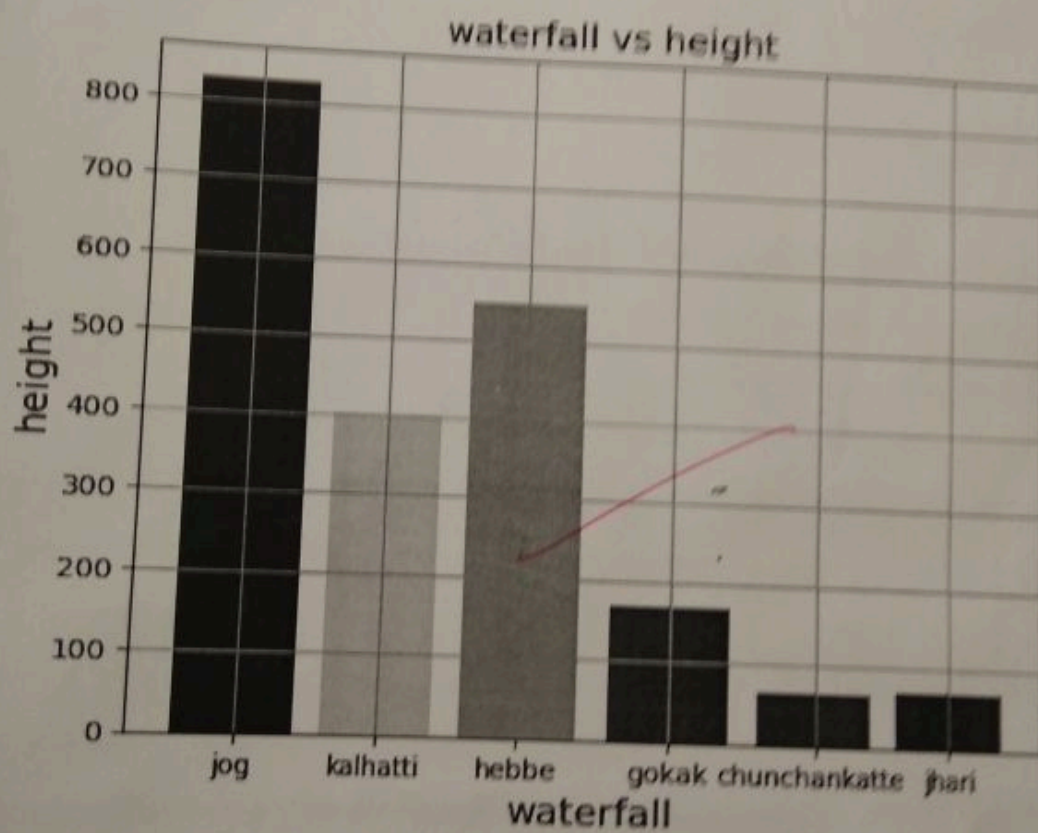
Transaction completed.



08. Demonstrate various graphs and plots that enable data visualization (Bar graph, pie chart, histogram, Box plot, Scatter plot).

```
-> import matplotlib.pyplot as plt
waterfall = ['Jog', 'Kalthatti', 'Hebbe', 'Gokak',
             'chunchankatte', 'Jhans']
height = [829, 403, 551, 170, 66, 70]
colors = ['Blue', 'yellow', 'Aqua', 'Green', 'Red']
plt.bar(waterfall, height, color = colors)
plt.title('waterfall', fontsize = 14)
plt.title('waterfall vs height', fontsize = 14)
plt.xlabel('waterfall', fontsize = 14)
plt.ylabel('height', fontsize = 14)
plt.grid(True)
plt.show()
```

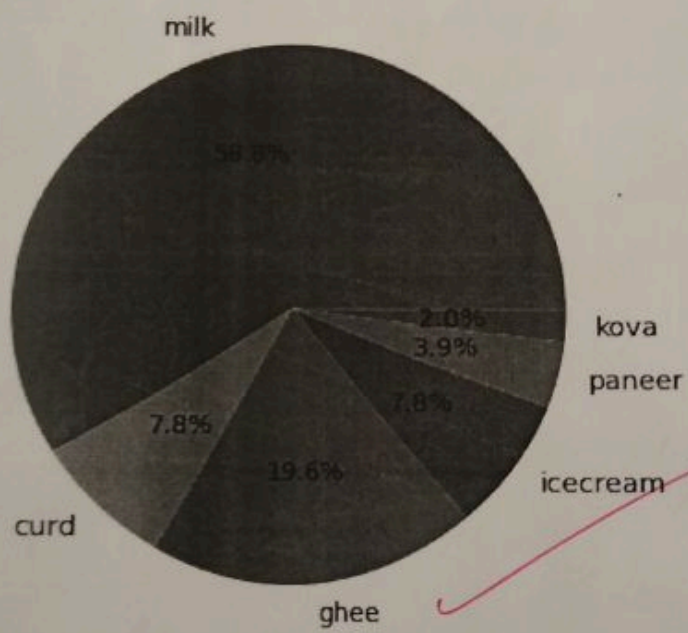




(8)

```
→ import matplotlib.pyplot as plt
Amount = [1500, 200, 500, 200, 100, 50]
products = ['milk', 'curd', 'Ghee', 'icecream',
            'paneer', 'kova']
plt.pie(Amount, labels = products, autopct =
        '%1.1f%%')
plt.axis('equal')
plt.show()
```

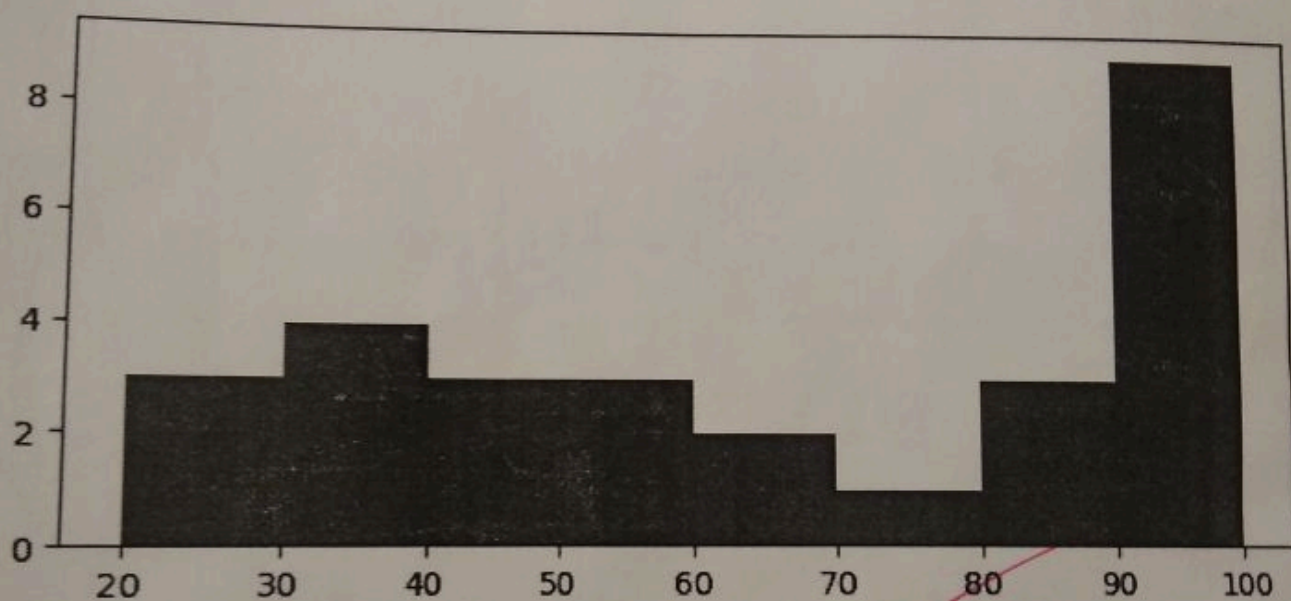




Q

```
-> import matplotlib.pyplot as plt
import numpy as np
array = np.array([15, 11, 21, 22, 33, 40, 41, 50, 55, 70, 90])
figure, axis = plt.subplots(figsize = (7, 3))
axis.hist(array, bins = [20, 30, 40, 50, 60, 70, 80, 90, 100])
plt.title('HISTOGRAM', fontsize = 14)
plt.show()
```

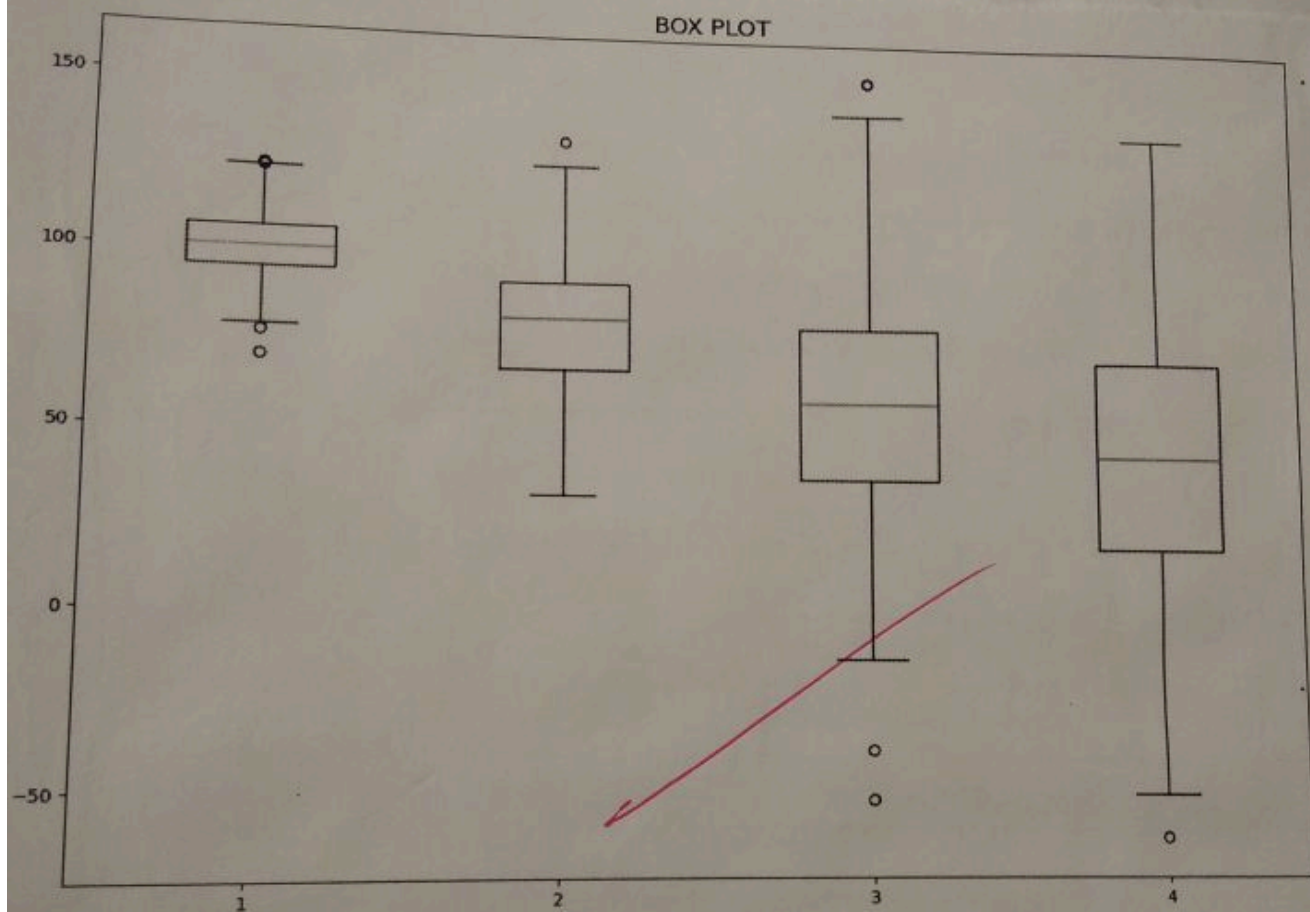




8

```
-> import matplotlib.pyplot as plt
import numpy as np
np.random.seed(10)
dataset1 = np.random.normal(100, 10, 200)
dataset2 = np.random.normal(80, 20, 200)
dataset3 = np.random.normal(60, 35, 200)
dataset4 = np.random.normal(50, 40, 200)
dataset = [dataset1, dataset2, dataset3, dataset4]
figure = plt.figure(figsize = (10, 7))
ax = figure.add_subplot(1, 1, 1)
bp = ax.boxplot(dataset)
plt.show()
```



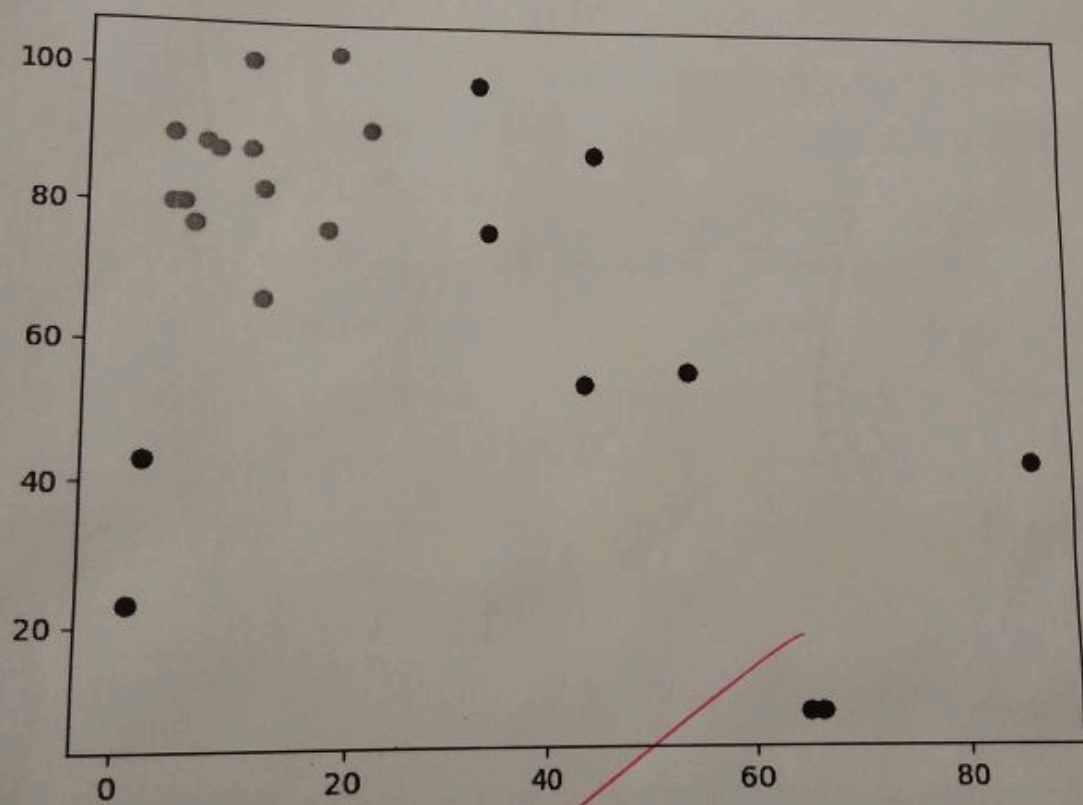


→ import matplotlib.pyplot as plt  
import numpy as np  
x = np.array ([4, 5, 7, 8, 1, 11, 6, 9, 10, 11, 20, 22, 25])  
y = np.array ([5, 6, 71, 66, 55, 51, 5, 88, 90, 91, 92, 1, 2])  
plt.scatter (x, y, color = 'orange')  
x1 = np.array ([1, 2, 44, 66, 77, 66, 11, 12, 18])  
y1 = np.array ([1, 2, 44, 77, 66, 11, 12, 18, 17])  
plt.scatter (x1, y1, color = 'Black')  
plt.show()

✓ 10/10  
11/5/23

Teacher's Signature : \_\_\_\_\_





98) Pgm to illustrate the working of map  
Print & filter & reduce.

```
* def multiplication(Number):  
    return Number * 2
```

```
Number = map(multiplication, [1, 2, 3, 4, 5, 6])
```

```
print("multiplication of number is:")
```

```
for ele in Number:  
    print(ele)
```

O/p: multiplication of number is:

2  
4  
6  
8  
10  
12

```
* def votingAge(age):
```

```
    if age >= 18:
```

```
        return age
```

```
a = [3, 45, 41, 18, 3, 4, 19, 25]
```

```
result = filter(votingAge, a)
```

```
print(result)
```

```
print("eligibility age for voting", list(result))
```

O/p: eligibility age for voting [45, 41, 18, 19, 25]

```
* from functools import reduce
```

```
def addnumber(x,y):
```

```
    return x+y
```

```
a = [12, 2, 4, 5, 6, 10, 10]
```

```
print ("The Sum of all list")
```

```
print (reduce (addnumber, a))
```

O/p: The Sum of all list.

49.



10) write a pgm for processing the image.

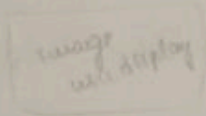
→ from PIL import Image

im = Image.open("orange.jpg")

im.rotate(45).show()

\* im

o/p:



\* im.size

o/p: (728, 410)

\* im.format

o/p: 'JPEG'

\* im.save("orange.png")

o/p: it save as .png format in pc

\* im.thumbnail((300, 200))

im.show()

o/p: image will display with size 300, 200

\* photoGray = Image.open("orange.jpg").convert('L')

photoGray.show()

o/p: Image will display as gray colour.