

```
In [4]: import pandas as pd
import numpy as np
import matplotlib as plt
import seaborn as sns
sns.set()
```

```
In [5]: df=sns.load_dataset('iris')
```

```
In [6]: df
```

Out[6]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
In [7]: df['species'].unique()
```

Out[7]: array(['setosa', 'versicolor', 'virginica'], dtype=object)

```
In [8]: df.groupby('species').describe(include='all')
```

Out[8]:

	sepal_length								sepal_width		...	petal_length		petal_width									
	count	mean	std		min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std		min	25%	50%	75%	max
species																							
setosa	50.0	5.006	0.352490		4.3	4.800	5.0	5.2	5.8	50.0	3.428	...	1.575	1.9	50.0	0.246	0.105386		0.1	0.2	0.2	0.3	0.6
versicolor	50.0	5.936	0.516171		4.9	5.600	5.9	6.3	7.0	50.0	2.770	...	4.600	5.1	50.0	1.326	0.197753		1.0	1.2	1.3	1.5	1.8
virginica	50.0	6.588	0.635880		4.9	6.225	6.5	6.9	7.9	50.0	2.974	...	5.875	6.9	50.0	2.026	0.274650		1.4	1.8	2.0	2.3	2.5

3 rows × 32 columns

```
In [9]: df.groupby('species').describe(include='all').T
```

Out[9]:

	species	setosa	versicolor	virginica
sepal_length	count	50.000000	50.000000	50.000000
	mean	5.006000	5.936000	6.588000
	std	0.352490	0.516171	0.635880
	min	4.300000	4.900000	4.900000
	25%	4.800000	5.600000	6.225000
	50%	5.000000	5.900000	6.500000
	75%	5.200000	6.300000	6.900000
	max	5.800000	7.000000	7.900000
sepal_width	count	50.000000	50.000000	50.000000
	mean	3.428000	2.770000	2.974000
	std	0.379064	0.313798	0.322497
	min	2.300000	2.000000	2.200000
	25%	3.200000	2.525000	2.800000
	50%	3.400000	2.800000	3.000000
	75%	3.675000	3.000000	3.175000
	max	4.400000	3.400000	3.800000
petal_length	count	50.000000	50.000000	50.000000
	mean	1.462000	4.260000	5.552000
	std	0.173664	0.469911	0.551895
	min	1.000000	3.000000	4.500000
	25%	1.400000	4.000000	5.100000
	50%	1.500000	4.350000	5.550000
	75%	1.575000	4.600000	5.875000
	max	1.900000	5.100000	6.900000
petal_width	count	50.000000	50.000000	50.000000

species	setosa	versicolor	virginica
mean	0.246000	1.326000	2.026000
std	0.105386	0.197753	0.274650
min	0.100000	1.000000	1.400000
25%	0.200000	1.200000	1.800000
50%	0.200000	1.300000	2.000000
75%	0.300000	1.500000	2.300000
max	0.600000	1.800000	2.500000

```
In [10]: arr=[]
s=int(input("Enter no of values"))
for i in range(0,s):
    a=int(input())
    arr.append(a)

g_sum=sum(arr)
mean=g_sum/s

print("The mean is ", mean)
```

```
Enter no of values6
12
45
56
23
89
78
The mean is  50.5
```

```
In [2]: arr=[]
s=int(input("Enter no of values "))
for i in range(0,s):
    a=int(input())
    arr.append(a)

arr.sort()

if (s%2==0):
    m1=arr[s//2]
    m2=arr[s//2-1]
    median = (m1+m2)/2

else:
    median = arr[s//2]

print("The median is ", median)
```

Enter no of values 6

65

45

95

32

15

75

The median is 55.0

```
In [3]: from collections import Counter
arr = []
s = int(input('Enter num of input: '))
for i in range(0,s):
    app = int(input())
    arr.append(app)

data = Counter(arr)
get_mode = dict(data)
mode = [k for k, v in get_mode.items() if v == max(list(data.values()))]

if len(mode) == s:
    get_mode = "No mode found"
else:
    get_mode = "Mode is / are: " + ', '.join(map(str, mode))
print(get_mode)
```

```
Enter num of input: 5
2
2
3
1
4
Mode is / are: 2
```

```
In [11]: df.groupby('species').mean()
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

```
In [14]: df.groupby('species').median()
```

```
Out[14]:
```

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	5.0	3.4	1.50	0.2
versicolor	5.9	2.8	4.35	1.3
virginica	6.5	3.0	5.55	2.0

```
In [ ]:
```