

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import scipy.stats as stats
```

```
df=pd.read_excel('/content/domestic dataset.xlsx')
df.head(10)
```

	RESPONDENTS	Age(yrs)	Working Hours(daily)	Income	Milk	Tea	Bread	chappati / roti/ phulka	Fruits	Poha
0	1	54	6	6000	0	1	1	1	1	1
1	2	25	3	10000	0	0	0	0	0	0
2	3	35	5	12000	0	1	0	0	0	0
3	4	29	4	6000	0	1	1	0	1	1
4	5	34	1	4000	0	1	0	0	0	0
5	6	18	5	4000	0	0	0	0	0	0
6	7	19	3	9000	0	1	0	0	0	0
7	8	21	3	3500	0	1	0	0	0	0
8	9	21	2	3500	0	1	0	0	0	0
9	10	15	3	4000	0	1	0	0	0	0

10 rows × 48 columns



## evaluation

```
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 48 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   RESPONDENTS                          10 non-null     int64
1   Age(yrs)                             10 non-null     int64
2   Working Hours(daily)                 10 non-null     int64
3   Income                               10 non-null     int64
4   Milk                                 10 non-null     int64
```

```

5   Tea                                10 non-null    int64
6   Bread                             10 non-null    int64
7   chappati / roti/ phulka           10 non-null    int64
8   Fruits                             10 non-null    int64
9   Poha                              10 non-null    int64
10  Upma                              10 non-null    int64
11  Egg/egg omlet                      10 non-null    int64
12  Salad                             10 non-null    int64
13  Dal                               10 non-null    int64
14  Rice                              10 non-null    int64
15  Curd                              10 non-null    int64
16  Sukhi sabhji                      10 non-null    int64
17  Gravy sabji                       10 non-null    int64
18  Papad                             10 non-null    int64
19  meat                              10 non-null    int64
20  Fish                              10 non-null    int64
21  Mutton                            10 non-null    int64
22  chicken                           10 non-null    int64
23  Apple                             10 non-null    int64
24  Banana                            10 non-null    int64
25  Pineapple                         10 non-null    int64
26  Papaya                            10 non-null    int64
27  Chickoo                           10 non-null    int64
28  Grapes                            10 non-null    int64
29  Orange                            10 non-null    int64
30  Coconut                           10 non-null    int64
31  Sweetlime                         10 non-null    int64
32  Pomegranate                       10 non-null    int64
33  Watermelon                        10 non-null    int64
34  Muskmelon                         10 non-null    int64
35  Mung                             10 non-null    int64
36  Matki                             10 non-null    int64
37  Chawli                            10 non-null    int64
38  Masoor                            10 non-null    int64
39  Toor                              10 non-null    int64
40  Rajma                             10 non-null    int64
41  Vatana                            10 non-null    int64
42  Chana daal                        10 non-null    int64
43  Urad                              10 non-null    int64
44  6-7 hours                         10 non-null    int64
45  Less than 6 hours                 10 non-null    int64
46  More than 6 hours                 10 non-null    int64
47  Do you get regular periods        10 non-null    int64

```

dtypes: int64(48)

memory usage: 3.9 KB

None

	RESPONDENTS	Age(yrs)	Working Hours(daily)	Income	Milk \
count	10.00000	10.000000	10.000000	10.000000	10.0

mean

```

mean1=df['Income']
mean=np.mean(mean1)

```

```
x=df['Income']  
x=np.sort(x)  
median=np.median(x)  
print("median:",median)
```

```
median: 5000.0
```

```
print("mode:",stats.mode(x))
```

```
mode: ModeResult(mode=array([4000]), count=array([3]))
```

## Visualization

```
df['Income'].hist()
```

Double-click (or enter) to edit

```
#convert pandas DataFrame object to numpy array and sort  
h = np.asarray(df['Income'])  
h = sorted(h)
```

```
#use the scipy stats module to fit a normal distribution with same mean and standard deviation  
fit = stats.norm.pdf(h, np.mean(h), np.std(h))
```

```
#plot both series on the histogram  
plt.plot(h,fit,'-',linewidth = 2,label="Normal distribution with same mean and var")  
plt.hist(h,density=True,bins = 100,label="Actual distribution")  
plt.legend()  
plt.show()
```

```
#plot mean
plt.plot(h,fit,'-',linewidth = 2,label="Normal distribution with same mean and var")
plt.hist(h,density=False,bins = 100,label="Actual distribution")
plt.axvline(mean)
plt.legend()
plt.show()
```

```
#plot median
plt.plot(h,fit,'-',linewidth = 2,label="Normal distribution with same mean and var")
plt.hist(h,density=False,bins = 100,label="Actual distribution")
plt.axvline(median)
plt.legend()
plt.show()
```

```
#plot mode
plt.plot(h,fit,'-',linewidth = 2,label="Normal distribution with same mean and var")
plt.hist(h,density=False,bins = 100,label="Actual distribution")
mode=stats.mode(x)
plt.axvline(mode[0])
plt.legend()
plt.show()
```

Double-click (or enter) to edit

```
#range
print(np.ptp(x))
```

8500

```
#variance
from scipy.stats import variation
k=np.var(x) # coefficient of variation
k
```

8510000.0

```
#standard deviation
import statistics
print(statistics.pstdev(x)) #POPULATION SD
print(statistics.stdev(x)) #sample sd
```

2917.1904291629644  
3074.988617865113

```
#skewness
from scipy.stats import skew
skew(x)
```

```
0.8080872188720842
```

```
#kurtosis
from scipy.stats import kurtosis
kurtosis(x)
```

```
-0.8207748953674465
```

## quartiles

```
q1=np.percentile(x,25)
print("q1:",q1)
q2=np.percentile(x,50)
print("q2:",q2)
q3=np.percentile(x,75)
print("q3:",q3)
iqr=q3-q1
print("iqr:",iqr)
```

```
q1: 4000.0
q2: 5000.0
q3: 8250.0
iqr: 4250.0
```

## box plot visualisation

```
plt.boxplot(x)
plt.show()
```

