TASK-2

calculate summary statistics

- **calculate mean, median, mode ,std for a dataset**

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
path=r"C:\Users\Sruth\Documents\Naresh it\EDA\Datafiles\Loan_prediction_data.csv
df=pd.read_csv(path)
df
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantInco
0	LP001002	Male	No	0	Graduate	No	5
1	LP001003	Male	Yes	1	Graduate	No	4
2	LP001005	Male	Yes	0	Graduate	Yes	3
3	LP001006	Male	Yes	0	Not Graduate	No	2
4	LP001008	Male	No	0	Graduate	No	6
...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2
610	LP002979	Male	Yes	3+	Graduate	No	4
611	LP002983	Male	Yes	1	Graduate	No	8
612	LP002984	Male	Yes	2	Graduate	No	7
613	LP002990	Female	No	0	Graduate	Yes	4

614 rows × 13 columns

In [5]:
```
df.shape
```

Out[5]:
```
(614, 13)
```

In [6]:
```
df.size
```

Out[6]:
```
7982
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [9]:
```
cat_cols=df.select_dtypes(include='object').columns
cat_cols
```

Out[9]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [11]:
```
num_cols=df.select_dtypes(exclude='object').columns
num_cols
```

Out[11]:
```
Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History'],
      dtype='object')
```

calculating mean

For numerical columns

In [14]:
```
df[['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credi
```

Out[14]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_His
0	5849	0.0	NaN	360.0	
1	4583	1508.0	128.0	360.0	
2	3000	0.0	66.0	360.0	
3	2583	2358.0	120.0	360.0	
4	6000	0.0	141.0	360.0	
...	...	...	...	...	
609	2900	0.0	71.0	360.0	
610	4106	0.0	40.0	180.0	
611	8072	240.0	253.0	360.0	
612	7583	0.0	187.0	360.0	
613	4583	0.0	133.0	360.0	

614 rows × 5 columns

In [16]:
```
count=len(df['ApplicantIncome'])
min_income=min(df['ApplicantIncome'])
max_income=max(df['ApplicantIncome'])
print("The number of wage samples are :",count)
print("The minimum wage is :" ,min_income)
print("The maximum wage is :" ,max_income)
```

```
The number of wage samples are : 614
The minimum wage is : 150
The maximum wage is : 81000
```

In [20]:
```
mean_income=round(df['ApplicantIncome'].mean(),2)
mean_income
```

Out[20]: 5403.46

In [21]:
```
median_income=round(df['ApplicantIncome'].median(),2)
median_income
```

Out[21]: 3812.5

In [22]:
```
std_income=round(df['ApplicantIncome'].std(),2)
std_income
```

Out[22]: 6109.04

In [26]:
```
values=[mean_income,median_income,std_income]
index=['Mean','median','standard deviation']
cols=['ApplicantIncome']
pd.DataFrame(values,index=index,columns=cols)
```

Out[26]:

	ApplicantIncome
Mean	5403.46
median	3812.50
standard deviation	6109.04

In [29]:
```
mean_coapp=round(df['CoapplicantIncome'].mean(),2)
median_coapp=round(df['CoapplicantIncome'].median(),2)
std_coapp=round(df['CoapplicantIncome'].std(),2)
values=[mean_coapp,median_coapp,std_coapp]
index=['Mean','median','standard deviation']
cols=['CoapplicantIncome']
pd.DataFrame(values,index=index,columns=cols)
```

Out[29]:

	CoapplicantIncome
Mean	1621.25
median	1188.50
standard deviation	2926.25

By using predefined method

describe function

In [28]:
```
round(df.describe(),2)
```

Out[28]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_H
count	614.00	614.00	592.00	600.00	
mean	5403.46	1621.25	146.41	342.00	
std	6109.04	2926.25	85.59	65.12	
min	150.00	0.00	9.00	12.00	
25%	2877.50	0.00	100.00	360.00	
50%	3812.50	1188.50	128.00	360.00	
75%	5795.00	2297.25	168.00	360.00	
max	81000.00	41667.00	700.00	480.00	

In [40]:
```
df1=pd.DataFrame()
for i in num_cols:
    mean_coapp=round(df[i].mean(),2)
    median_coapp=round(df[i].median(),2)
    std_coapp=round(df[i].std(),2)
    values=[mean_coapp,median_coapp,std_coapp]
    index=['Mean','median','standard deviation']
    cols=[i]
    df2=pd.DataFrame(values,index=index,columns=cols)
```

```
df1=pd.concat([df1,df2],axis=1)
```

In [41]:
```
df1
```

Out[41]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Cred
Mean	5403.46	1621.25	146.41	342.00	
median	3812.50	1188.50	128.00	360.00	
standard deviation	6109.04	2926.25	85.59	65.12	

In [ ]:
```

```