```
import numpy as np
from scipy import stats
from scipy.stats import skew
from scipy.stats import kurtosis
from scipy.stats import variation
import pandas as pd


from google.colab import files
files.upload()
```

```
Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving house_prices.csv to house_prices.csv
{'house_prices.csv': b'LotFrontage,LotArea,OverallQual,OverallCond,YearBuilt,YearRemodAd
```

```
df=pd.read_csv('house_prices.csv')
df.head()
```

	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea
0	65.0	8450	7	5	2003	2003	196.0
1	80.0	9600	6	8	1976	1976	0.0
2	68.0	11250	7	5	2001	2002	162.0
3	60.0	9550	7	5	1915	1970	0.0
4	84.0	14260	8	5	2000	2000	350.0

```
#Q. Evaluate the methods : shape, info, describe.
df.shape
```

```
(1379, 35)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1379 entries, 0 to 1378
Data columns (total 35 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   LotFrontage    1379 non-null   float64
 1   LotArea        1379 non-null   int64  
 2   OverallQual    1379 non-null   int64  
 3   OverallCond    1379 non-null   int64  
 4   YearBuilt      1379 non-null   int64
```

```
 5   YearRemodAdd   1379 non-null   int64
 6   MasVnrArea     1379 non-null   float64
 7   BsmtFinSF1     1379 non-null   int64
 8   BsmtFinSF2     1379 non-null   int64
 9   BsmtUnfSF      1379 non-null   int64
 10  TotalBsmtSF    1379 non-null   int64
 11  1stFlrSF       1379 non-null   int64
 12  2ndFlrSF       1379 non-null   int64
 13  LowQualFinSF   1379 non-null   int64
 14  GrLivArea      1379 non-null   int64
 15  BsmtFullBath   1379 non-null   int64
 16  BsmtHalfBath   1379 non-null   int64
 17  FullBath       1379 non-null   int64
 18  HalfBath       1379 non-null   int64
 19  BedroomAbvGr   1379 non-null   int64
 20  KitchenAbvGr   1379 non-null   int64
 21  TotRmsAbvGrd   1379 non-null   int64
 22  Fireplaces     1379 non-null   int64
 23  GarageYrBlt    1379 non-null   float64
 24  GarageCars     1379 non-null   int64
 25  GarageArea     1379 non-null   int64
 26  WoodDeckSF     1379 non-null   int64
 27  OpenPorchSF    1379 non-null   int64
 28  EnclosedPorch  1379 non-null   int64
 29  3SsnPorch      1379 non-null   int64
 30  ScreenPorch    1379 non-null   int64
 31  PoolArea       1379 non-null   int64
 32  MiscVal        1379 non-null   int64
 33  YrSold         1379 non-null   int64
 34  SalePrice      1379 non-null   int64
dtypes: float64(3), int64(32)
memory usage: 377.2 KB
```

```
df.describe()
```

	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemo
count	1379.000000	1379.000000	1379.000000	1379.000000	1379.000000	1379.00
mean	57.766497	10695.812183	6.187092	5.577955	1972.958666	1985.43
std	35.038221	10214.702133	1.345780	1.081031	29.379883	20.44
min	0.000000	1300.000000	2.000000	2.000000	1880.000000	1950.00
25%	41.500000	7741.000000	5.000000	5.000000	1955.000000	1968.00
50%	64.000000	9591.000000	6.000000	5.000000	1976.000000	1994.00
75%	79.000000	11708.500000	7.000000	6.000000	2001.000000	2004.00
max	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.00

```
"""Q. For the Saleprice attribute. Evaluate Mean,Median,Mode
Visualize histogram for saleprice."""
x=df['SalePrice']
x.head()
```

```
0    208500
1    181500
2    223500
3    140000
4    250000
Name: SalePrice, dtype: int64
```

```
mean=np.mean(x)
w=np.sort(x)
median=np.median(w)
print("Mean is ",mean)
print("Median is ",median)
mode=x.mode()
print("Mode is ",mode[0])
```

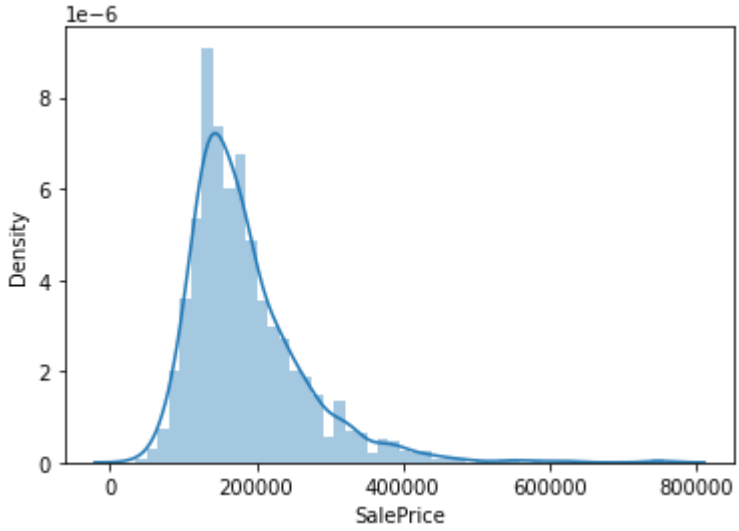
```
Mean is  185479.511240029
Median is  167500.0
Mode is  140000
```

```
#Visualize histogram for saleprice. USE THE PLOT OF MEAN,MEDIAN, MODE.
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
```

```
x=df['SalePrice']
plt.hist(x)
plt.axvline(mean,label="Mean",color="r")
plt.axvline(median,label="Median",color="b")
plt.axvline(mode[0],label="Mode",color="g")
plt.legend()
plt.show()
```


```
sns.distplot(x)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
```




```
#Q. Evaluate measures of spread :range,variance,standard deviation,skewness and kurtosis


#range
a=np.ptp(x)
print(a)
#Another Type
p=x.max()-x.min()
print(p)
```

```
719689
719689
```

```
#variance,standard
t=np.var(x)
u=np.std(x)
print(t)
print(u)
```

```
6240246804.531235
78995.2327962342
```

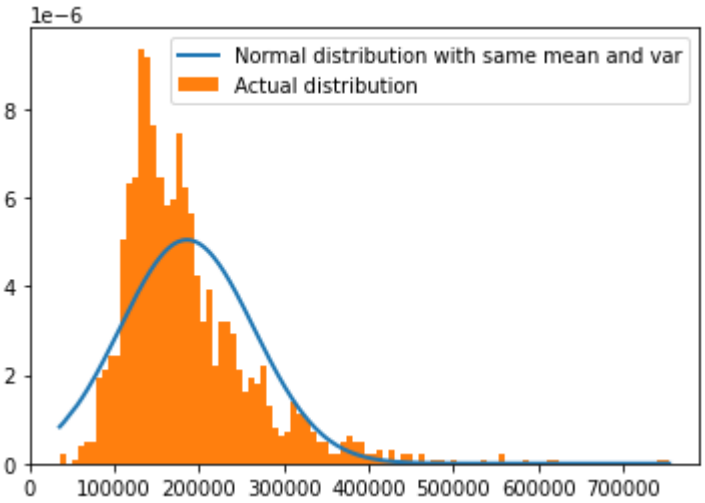
```
#Skewness
skew(x)
```

```
1.9332562820097063
```

```
#kurtosis
kurtosis(x)
```

```
6.706904068638849
```

```
#Q. Evaluate Visualize the histogram of the normal distribution.
h = np.asarray(df['SalePrice'])
h = sorted(h)
fit = stats.norm.pdf(h, np.mean(h), np.std(h))
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()
```




```
#Q. Evaluate the Quartiles q1, q3 and iqr USE IQR rule to detect outliers
q1=np.percentile(x,25)
q2=np.percentile(x,50)
q3=np.percentile(x,75)
print(q1)
print(q2)
print(q3)
iqr=q3-q1
print("iqr is ",iqr)
```

```
134000.0
167500.0
217750.0
iqr is  83750.0
```

```
#IQR rule to detect outliers
lowerBound=(q1-1.5*iqr)
```

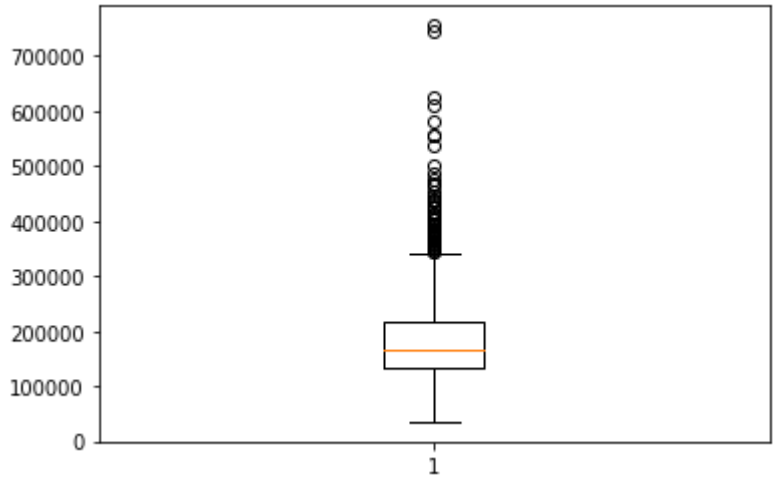
```
print("LowerBound is ",lowerBound)

upperBound=(q3+1.5*iqr)
print("UpperBound is ", upperBound)

outlier=(q1-1.5*iqr)
print("outlier is", outlier )
```

```
     LowerBound is  8375.0
     UpperBound is  343375.0
     outlier is 8375.0
```

```
#Q. Evaluate visualize boxplot
plt.boxplot(x)
plt.show()
```




```
"""Q. Evaluate the Correlation and covariance for the attributes in the
dataset
lotarea, grlivarea,garagearea,saleprice"""
import seaborn as sns
corelation=df[['LotArea','GrLivArea','GarageArea','SalePrice']].corr()
print (corelation)
sns.heatmap(corelation)
```

```
             LotArea  GrLivArea  GarageArea  SalePrice
LotArea     1.000000   0.257243    0.167622   0.252921
GrLivArea   0.257243   1.000000    0.478811   0.708172
GarageArea  0.167622   0.478811    1.000000   0.608405
SalePrice   0.252921   0.708172    0.608405   1.000000
<matplotlib.axes._subplots.AxesSubplot at 0x7f869722dc50>
```



```
#covariance
df[['LotArea','GrLivArea','GarageArea','SalePrice']].cov()
```

	LotArea	GrLivArea	GarageArea	SalePrice
LotArea	1.043401e+08	1.364127e+06	3.179236e+05	2.041596e+08
GrLivArea	1.364127e+06	2.695069e+05	4.615466e+04	2.905241e+07
GarageArea	3.179236e+05	4.615466e+04	3.447726e+04	8.927251e+06
SalePrice	2.041596e+08	2.905241e+07	8.927251e+06	6.244775e+09