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

```
Choose Files crime_data.csv
• crime_data.csv(text/csv) - 1328 bytes, last modified: 2/24/2023 - 100% done
Saving crime_data.csv to crime_data.csv
```

```
import pandas as pd
df=pd.read_csv("crime_data.csv")
df.head()
#df.shape
```

| | Unnamed: 0 | Murder | Assault | UrbanPop | Rape |
|---|---|---|---|---|---|
| **0** | Alabama | 13.2 | 236 | 58 | 21.2 |
| **1** | Alaska | 10.0 | 263 | 48 | 44.5 |
| **2** | Arizona | 8.1 | 294 | 80 | 31.0 |
| **3** | Arkansas | 8.8 | 190 | 50 | 19.5 |
| **4** | California | 9.0 | 276 | 91 | 40.6 |

```
df.dtypes
df.head()
```

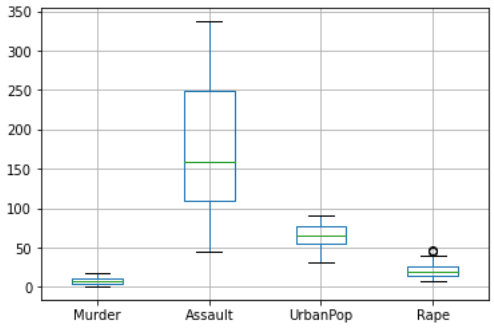
| | Murder | Assault | UrbanPop | Rape |
|---|---|---|---|---|
| **0** | 13.2 | 236 | 58 | 21.2 |
| **1** | 10.0 | 263 | 48 | 44.5 |
| **2** | 8.1 | 294 | 80 | 31.0 |
| **3** | 8.8 | 190 | 50 | 19.5 |
| **4** | 9.0 | 276 | 91 | 40.6 |

```
df.isnull().sum()
```

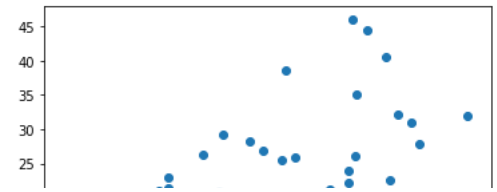
```
Murder      0
Assault     0
UrbanPop    0
Rape        0
dtype: int64
```

```
df.boxplot(None)
```

```
<AxesSubplot:>
```




```
import matplotlib.pyplot as plt
plt.scatter(df["Assault"],df["Rape"])
plt.show()
```




```
X=df.iloc[:,0:4]
X.head()
from sklearn.preprocessing import StandardScaler
stscaler = StandardScaler().fit(X)
X = stscaler.transform(X)
X
```

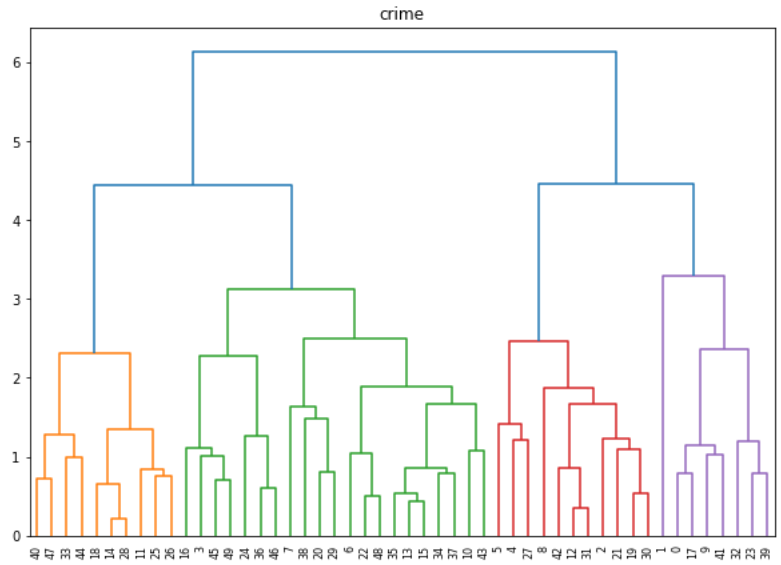
```
array([[ 1.25517927,  0.79078716, -0.52619514, -0.00345116],
       [ 0.51301858,  1.11805959, -1.22406668,  2.50942392],
       [ 0.07236067,  1.49381682,  1.00912225,  1.05346626],
       [ 0.23470832,  0.23321191, -1.08449238, -0.18679398],
       [ 0.28109336,  1.2756352 ,  1.77678094,  2.08881393],
       [ 0.02597562,  0.40290872,  0.86954794,  1.88390137],
       [-1.04088037, -0.73648418,  0.79976079, -1.09272319],
       [-0.43787481,  0.81502956,  0.45082502, -0.58583422],
       [ 1.76541475,  1.99078607,  1.00912225,  1.1505301 ],
       [ 2.22926518,  0.48775713, -0.38662083,  0.49265293],
       [-0.57702994, -1.51224105,  1.21848371, -0.11129987],
       [-1.20322802, -0.61527217, -0.80534376, -0.75839217],
       [ 0.60578867,  0.94836277,  1.21848371,  0.29852525],
       [-0.13637203, -0.70012057, -0.03768506, -0.0250209 ],
       [-1.29599811, -1.39102904, -0.5959823 , -1.07115345],
       [-0.41468229, -0.67587817,  0.03210209, -0.34856705],
       [ 0.44344101, -0.74860538, -0.94491807, -0.53190987],
       [ 1.76541475,  0.94836277,  0.03210209,  0.10439756],
       [-1.31919063, -1.06375661, -1.01470522, -1.44862395],
       [ 0.81452136,  1.56654403,  0.10188925,  0.70835037],
       [-0.78576263, -0.26375734,  1.35805802, -0.53190987],
       [ 1.00006153,  1.02108998,  0.59039932,  1.49564599],
       [-1.1800355 , -1.19708982,  0.03210209, -0.68289807],
       [ 1.9277624 ,  1.06957478, -1.5032153 , -0.44563089],
       [ 0.28109336,  0.0877575 ,  0.31125071,  0.75148985],
       [-0.41468229, -0.74860538, -0.87513091, -0.521125  ],
       [-0.80895515, -0.83345379, -0.24704653, -0.51034012],
       [ 1.02325405,  0.98472638,  1.0789094 ,  2.671197  ],
       [-1.31919063, -1.37890783, -0.66576945, -1.26528114],
       [-0.08998698, -0.14254532,  1.63720664, -0.26228808],
       [ 0.83771388,  1.38472601,  0.31125071,  1.17209984],
       [ 0.76813632,  1.00896878,  1.42784517,  0.52500755],
       [ 1.20879423,  2.01502847, -1.43342815, -0.55347961],
       [-1.62069341, -1.52436225, -1.5032153 , -1.50254831],
       [-0.11317951, -0.61527217,  0.66018648,  0.01811858],
       [-0.27552716, -0.23951493,  0.1716764 , -0.13286962],
       [-0.66980002, -0.14254532,  0.10188925,  0.87012344],
       [-0.34510472, -0.78496898,  0.45082502, -0.68289807],
       [-1.01768785,  0.03927269,  1.49763233, -1.39469959],
       [ 1.53348953,  1.3119988 , -1.22406668,  0.13675217],
       [-0.92491776, -1.027393  , -1.43342815, -0.90938037],
       [ 1.25517927,  0.20896951, -0.45640799,  0.61128652],
       [ 1.13921666,  0.36654512,  1.00912225,  0.46029832],
       [-1.06407289, -0.61527217,  1.00912225,  0.17989166],
       [-1.29599811, -1.48799864, -2.34066115, -1.08193832],
       [ 0.16513075, -0.17890893, -0.17725937, -0.05737552],
       [-0.87853272, -0.31224214,  0.52061217,  0.53579242],
       [-0.48425985, -1.08799901, -1.85215107, -1.28685088],
       [-1.20322802, -1.42739264,  0.03210209, -1.1250778 ],
       [-0.22914211, -0.11830292, -0.38662083, -0.60740397]])
```

```
from sklearn.cluster import AgglomerativeClustering
cluster=AgglomerativeClustering(n_clusters=2,affinity="euclidean",linkage="single")
Y=cluster.fit_predict(X)
Y_new=pd.DataFrame(Y)
Y_new.value_counts()
```

```
0    49
1     1
dtype: int64
```

```
import scipy.cluster.hierarchy as shc
import matplotlib.pyplot as plt
plt.figure(figsize=(10,7))
```

```
plt.title("crime")
dend=shc.dendrogram(shc.linkage(X,method="complete"))
```




```
from sklearn.cluster import AgglomerativeClustering
agm=AgglomerativeClustering(n_clusters=4,affinity="euclidean",linkage="complete")
y=agm.fit_predict(X)
Y=pd.DataFrame(y)
Y.value_counts()
```

```
1    21
2    11
3    10
0     8
dtype: int64
```

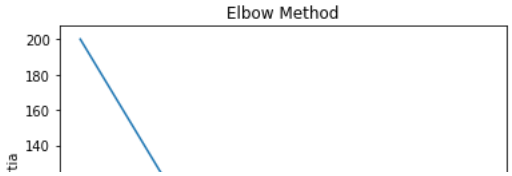
################ KMEANS #################

```
from sklearn.cluster import KMeans
Km=KMeans(n_clusters=3,n_init=20)
y=Km.fit_predict(X)
Y=pd.DataFrame(y)
Y.value_counts()
```

```
1    20
2    17
0    13
dtype: int64
```

```
inertia = []
for i in range(1,6):
    km = KMeans(n_clusters=i,random_state=0)
    km.fit(X)
    inertia.append(km.inertia_)

plt.plot(range(1, 6), inertia)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('inertia')
plt.show()
```




```
from sklearn.cluster import DBSCAN
DBSCAN()
dbscan = DBSCAN(eps=5, min_samples=7)
dbscan.fit(X)
dbscan.labels_#### used to kmow the how many clusters are forming
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0])
```

```
c=pd.DataFrame(dbscan.labels_,columns=["cluster"])
c
c["cluster"].value_counts()
```

```
0    50
Name: cluster, dtype: int64
```

```
clustered = pd.concat([df,c],axis=1)

noisedata = clustered[clustered['cluster']==-1]
finaldata = clustered[clustered['cluster']==0]

clustered.head()
```

| | Murder | Assault | UrbanPop | Rape | cluster |
|---|---|---|---|---|---|
| **0** | 13.2 | 236 | 58 | 21.2 | 0 |
| **1** | 10.0 | 263 | 48 | 44.5 | 0 |
| **2** | 8.1 | 294 | 80 | 31.0 | 0 |
| **3** | 8.8 | 190 | 50 | 19.5 | 0 |
| **4** | 9.0 | 276 | 91 | 40.6 | 0 |