

Practical No. 04

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

X = np.random.uniform(0,1,50)
Y = np.random.uniform(0,1,50)
df_xy = pd.DataFrame(columns=["X","Y"])
df_xy.X = X
df_xy.Y = Y

df_xy.plot(x="X" , y="Y", kind = "scatter")
model1 = KMeans(n_clusters = 3).fit(df_xy)
df_xy.plot(x = "X", y = "Y", c = model1.labels_, kind="scatter", s = 10, cmap = plt.cm.coolwarm
)

univ1 = pd.read_excel("C:\\Users\\CSE-09\\Downloads\\University_Clustering.xlsx")

univ1.describe()

univ = univ1.drop(["State"], axis=1)

#normalization function
def norm_func(i):
    x = (i-i.min()) / (i.max()-i.min())
    return (x)

#normalization data frame (considering the numerical part of data)

df_norm = norm_func(univ.iloc[:, 1:])

#elbow curve
TWSS = []
k = list(range(2, 9))
```

```

for i in k:
    kmeans = KMeans(n_clusters = i)
    kmeans.fit(df_norm)
    TWSS.append(kmeans.inertia_)

TWSS
#scree plot
plt.plot(k, TWSS, 'ro-');plt.xlabel("No_of_Clusters");plt.ylabel("total_within_SS")

model1.labels_
mb = pd.Series(model1.labels_)
univ['Clust'] = mb

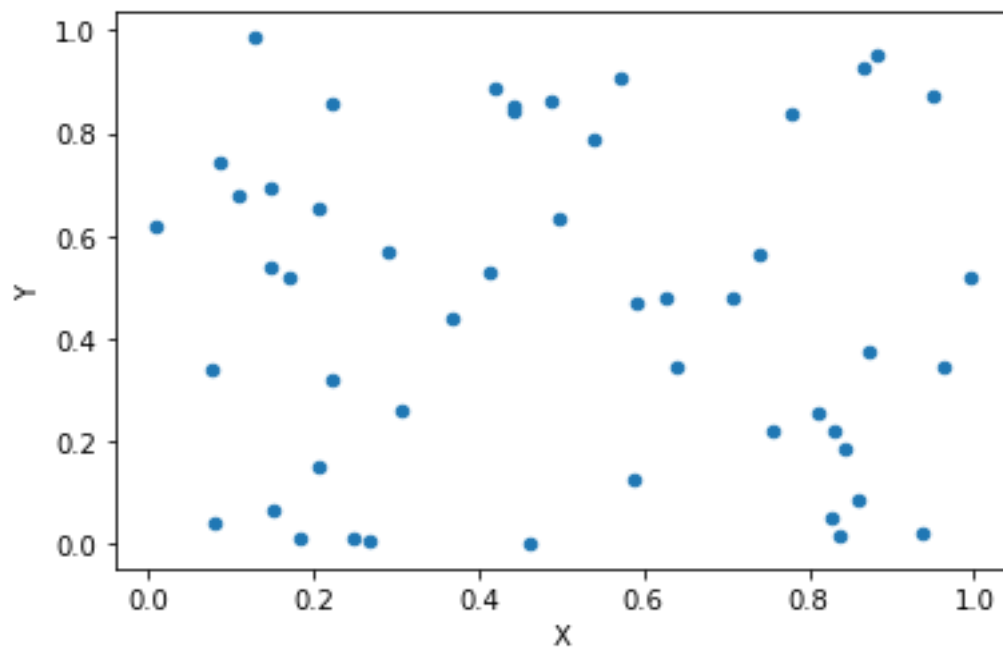
univ.head()
df_norm.head()
univ = univ.iloc[:,[7,0,1,2,3,4,5,6]]

univ.head()

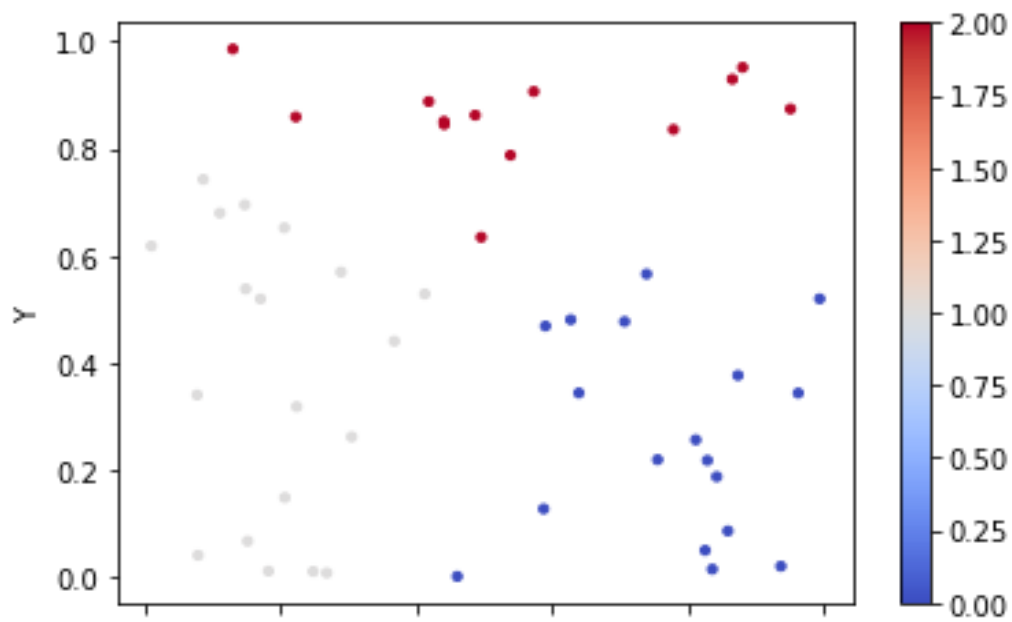
univ.iloc[:, 2:8].groupby(univ.clust).mean()

```

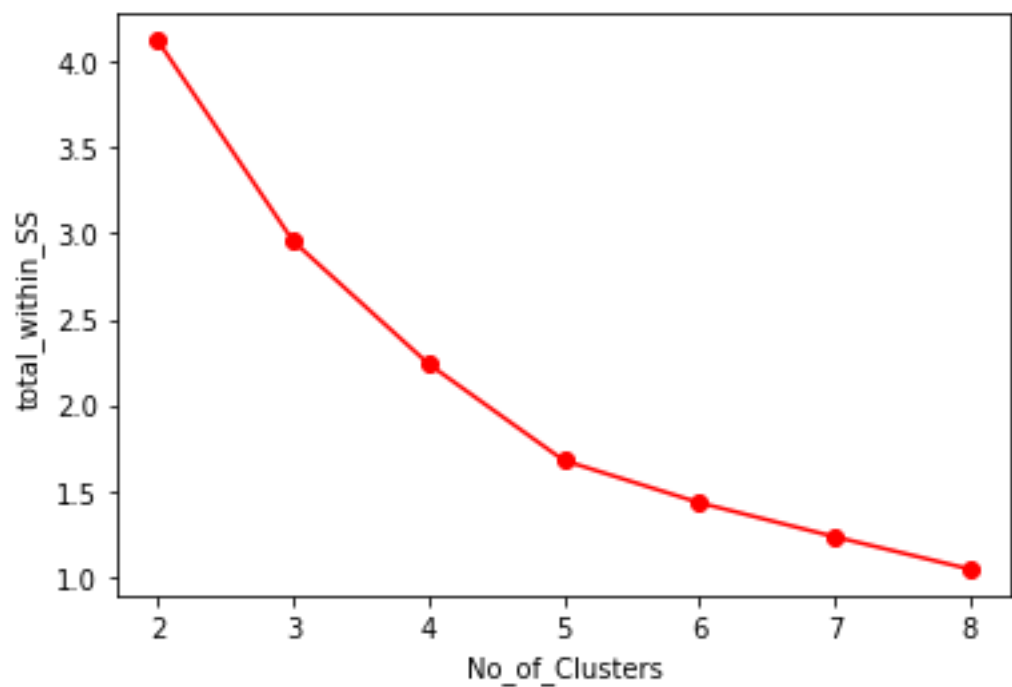
Outputs:



Scatter plot show the slits of random data



Scatter plot show the slits of random data using Kmeans



Relation between TWSS (Total Within Sum of Square) and No. of Cluster