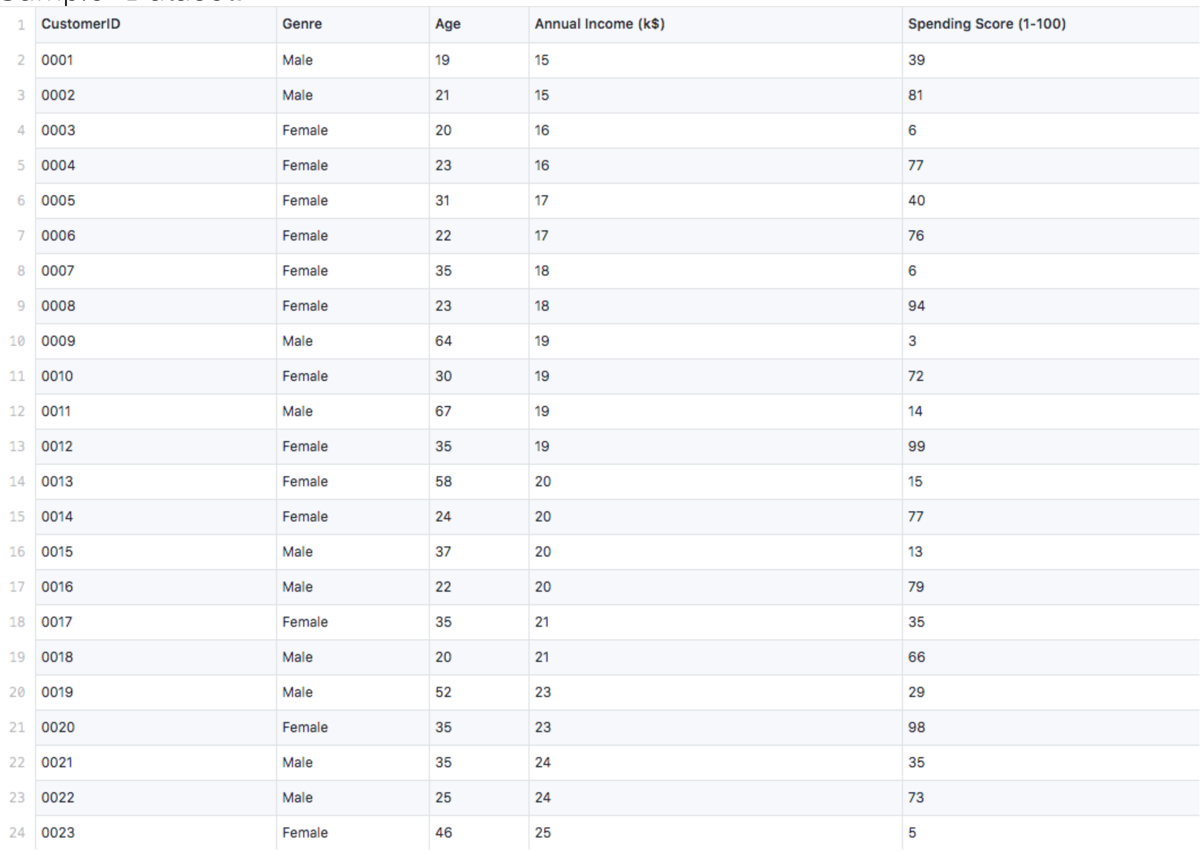
K-MEANS CLUSTERING

DEVELOPED BY: VISHRUTH KHARE
ROLL NUMBER: 2K18/CO/393

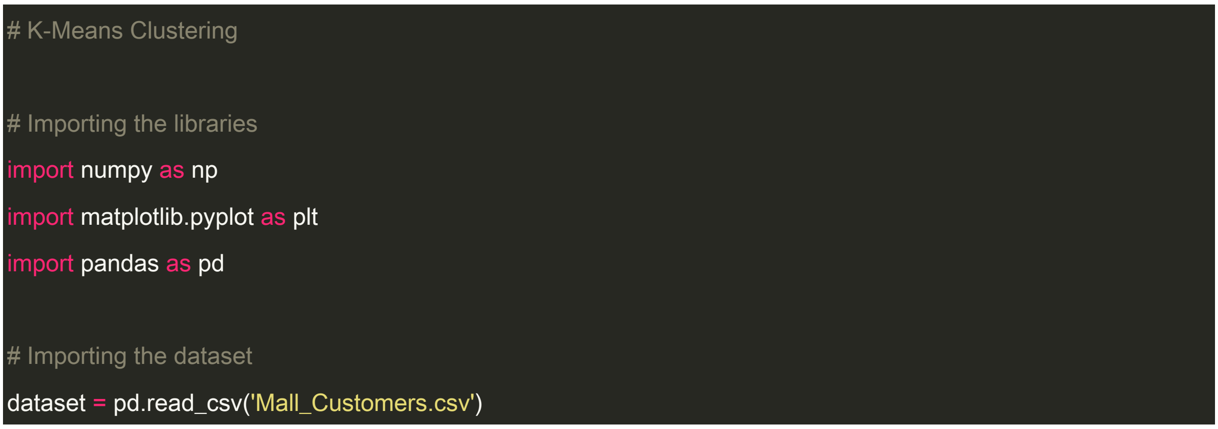
IDE: PyCharm
Compiler: Terminal (Python 3.6)
DataSet: Mall Customers

Sample Dataset:



CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0001	Male	19	15	39
0002	Male	21	15	81
0003	Female	20	16	6
0004	Female	23	16	77
0005	Female	31	17	40
0006	Female	22	17	76
0007	Female	35	18	6
0008	Female	23	18	94
0009	Male	64	19	3
0010	Female	30	19	72
0011	Male	67	19	14
0012	Female	35	19	99
0013	Female	58	20	15
0014	Female	24	20	77
0015	Male	37	20	13
0016	Male	22	20	79
0017	Female	35	21	35
0018	Male	20	21	66
0019	Male	52	23	29
0020	Female	35	23	98
0021	Male	35	24	35
0022	Male	25	24	73
0023	Female	46	25	5

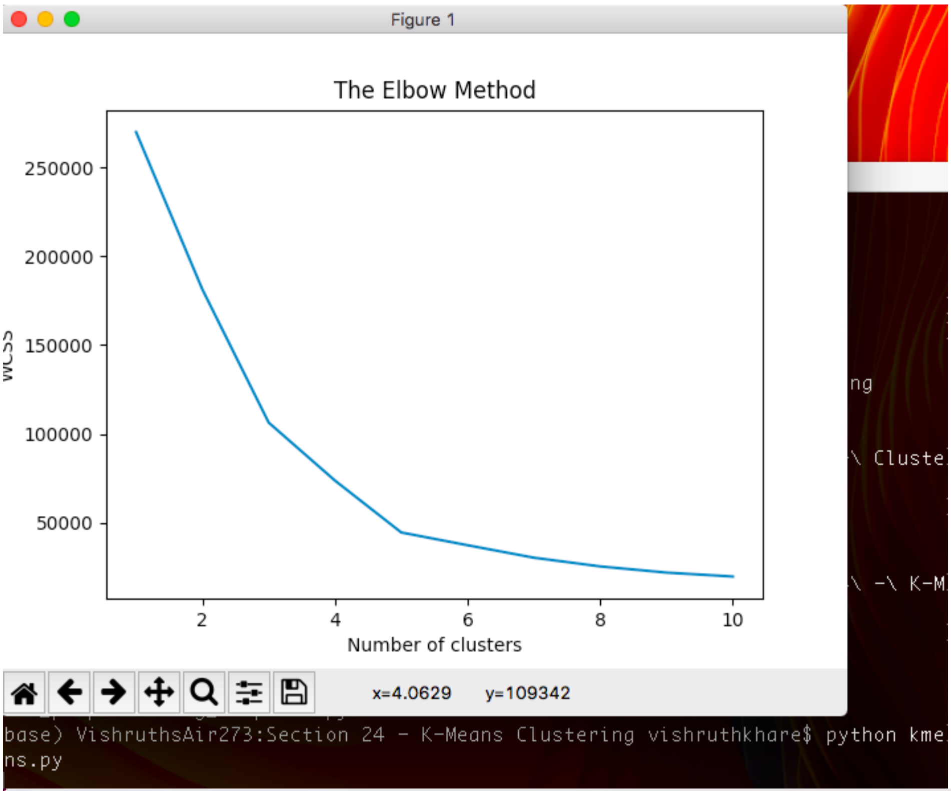
Source Code:



```
# K-Means Clustering

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Mall_Customers.csv')
```

```
X = dataset.iloc[:, [3, 4]].values
# y = dataset.iloc[:, 3].values

# Splitting the dataset into the Training set and Test set
"""from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)"""

# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)"""

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

OUTPUT：



```
base) VishruthsAir273:Section 24 - K-Means Clustering vishruthkhare$ python kme
ns.py
```

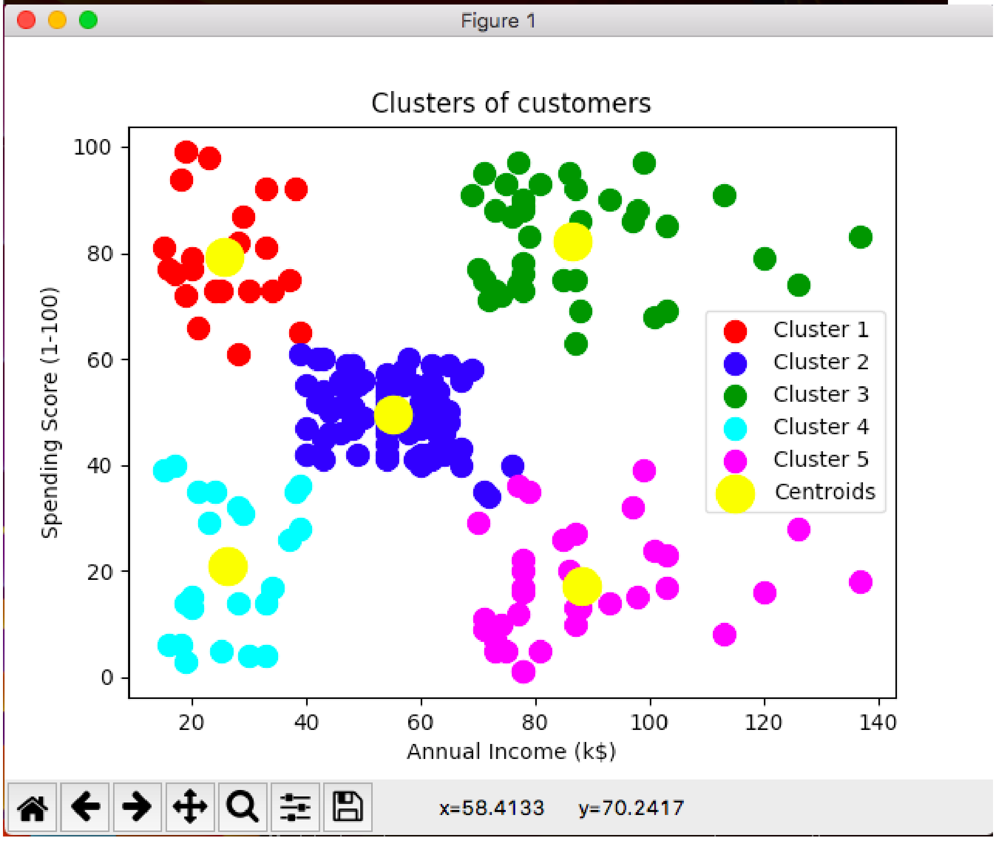


Figure 1
Clusters of customers
100
80
60
40
20
0
Spending Score (1-100)
20
40
60
80
100
120
140
Annual Income (k$)
● Cluster 1
● Cluster 2
● Cluster 3
● Cluster 4
● Cluster 5
● Centroids
x=58.4133 y=70.2417