

```
In [1]: #Experiment no 9 to perform and find the accuracy of K-Means algorithm.
```

```
In [2]: #Name: Shravani M Karne  
#Roll no.: 39  
#Sec: A  
#Subject:Big Data Analysis (ET 2 Lab)
```

```
In [12]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [13]: df = pd.read_csv('student_clustering.csv')
```

```
In [14]: df
```

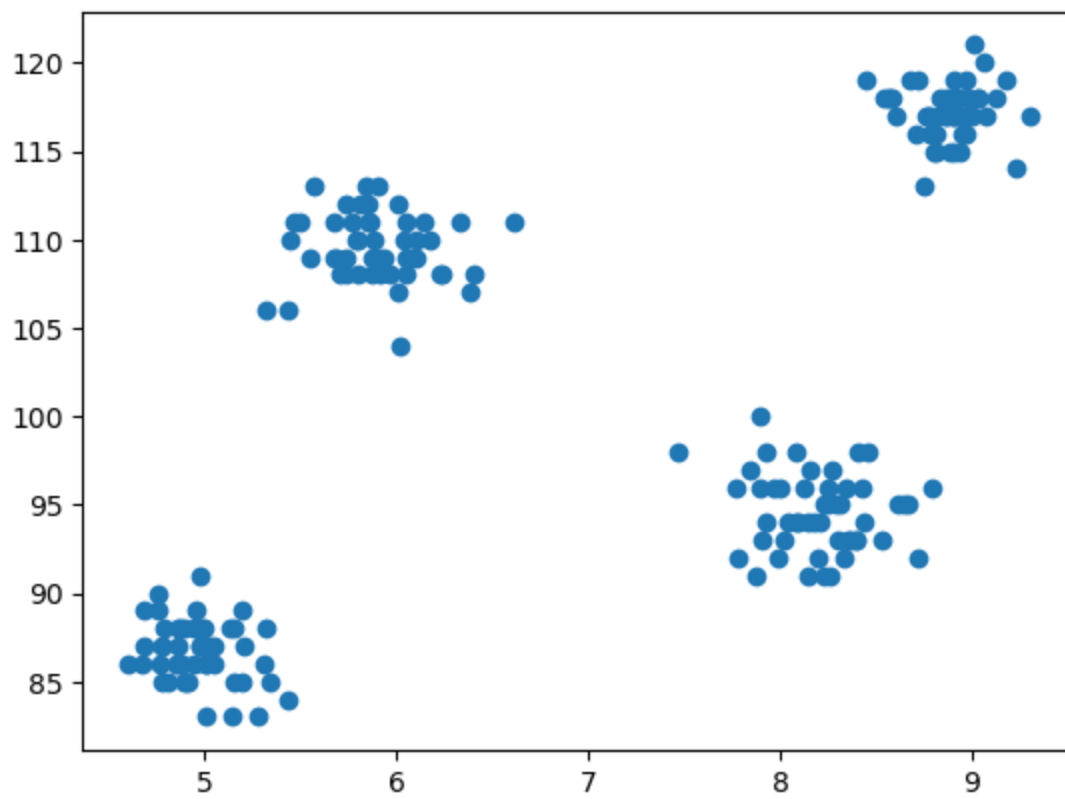
```
Out[14]:
```

	cgpa	iq
0	5.13	88
1	5.90	113
2	8.36	93
3	8.27	97
4	5.45	110
...
195	4.68	89
196	8.57	118
197	5.85	112
198	6.23	108
199	8.82	117

200 rows × 2 columns

```
In [15]: plt.scatter(df['cgpa'],df['iq'])
```

```
Out[15]: <matplotlib.collections.PathCollection at 0x1e9d1bd9190>
```



```
In [16]: from sklearn.cluster import KMeans
```

```
In [17]: wcss = []  
for i in range(1,11):  
    km = KMeans(n_clusters=i)  
    km.fit_predict(df)  
    wcss.append(km.inertia_)
```



```

g: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less chunks th
an available threads. You can avoid it by setting the environment variable OMP_NUM_THREA
DS=1.
    warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarnin
g: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less chunks th
an available threads. You can avoid it by setting the environment variable OMP_NUM_THREA
DS=1.
    warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarnin
g: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less chunks th
an available threads. You can avoid it by setting the environment variable OMP_NUM_THREA
DS=1.
    warnings.warn(

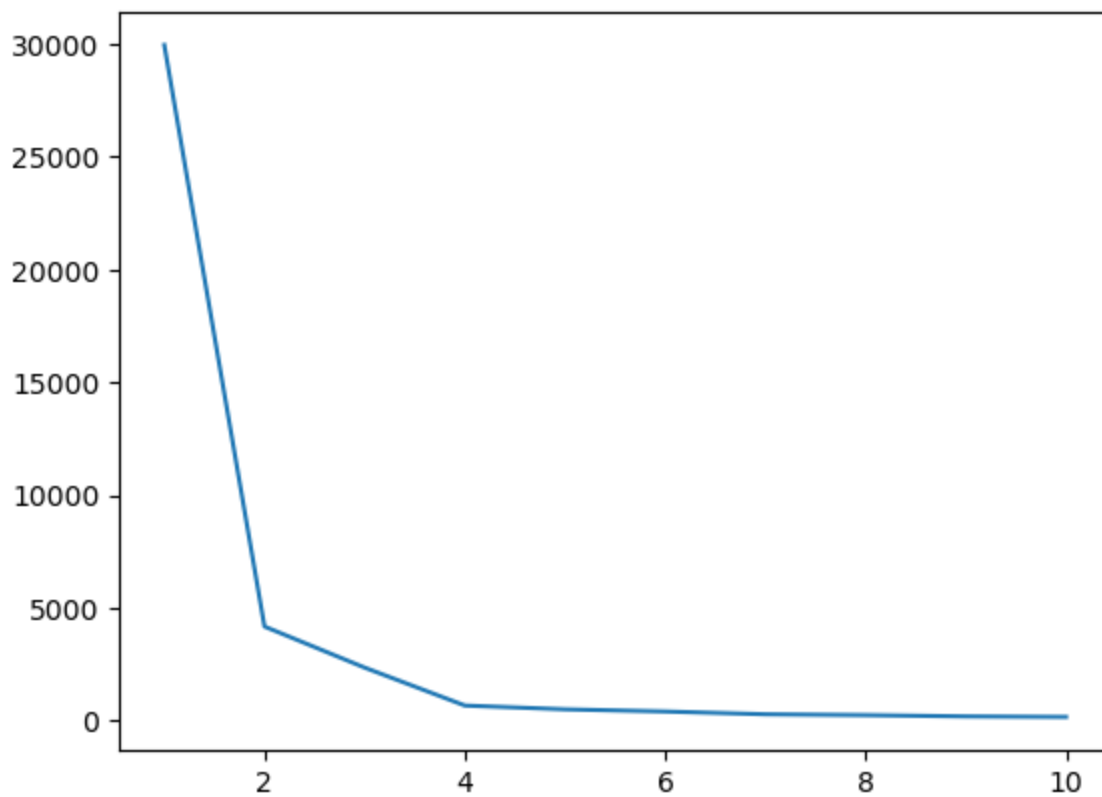
```

In [18]: wcss

Out[18]: [29957.898288,
4184.14127,
2362.7133489999997,
681.96966,
514.1616803171115,
417.8541311203876,
295.4391895943192,
257.15247762264204,
201.05722088123395,
176.73118902921556]

In [19]: plt.plot(range(1,11),wcss)

Out[19]: [<matplotlib.lines.Line2D at 0x1e9d1b3b0d0>]



```
In [20]: X = df.iloc[:, :].values
km = KMeans(n_clusters = 4)
y_means = km.fit_predict(X)
```

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
 super()._check_params_vs_input(X, default_n_init=10)
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
 warnings.warn(

```
In [21]: y_means
```

```
Out[21]: array([1, 0, 3, 3, 0, 0, 3, 2, 0, 3, 1, 0, 3, 1, 0, 3, 0, 3, 0, 0, 3, 1,
        3, 1, 1, 3, 1, 2, 3, 0, 2, 0, 2, 0, 3, 3, 2, 0, 1, 0, 1, 3, 3, 1,
        2, 2, 3, 0, 2, 0, 1, 1, 2, 3, 2, 0, 0, 2, 0, 2, 0, 3, 3, 2, 1, 2,
        3, 1, 0, 3, 0, 2, 3, 1, 0, 2, 0, 2, 1, 3, 3, 2, 0, 1, 2, 1, 2, 0,
        2, 0, 2, 2, 3, 1, 3, 3, 2, 3, 1, 2, 0, 1, 1, 2, 1, 1, 3, 1, 2, 2,
        3, 2, 0, 0, 3, 2, 3, 0, 2, 1, 1, 0, 3, 2, 3, 1, 3, 0, 1, 3, 3, 0,
        1, 1, 0, 2, 0, 1, 3, 3, 3, 1, 0, 1, 1, 2, 1, 2, 0, 1, 2, 1, 2, 2,
        1, 3, 0, 2, 0, 3, 1, 2, 0, 3, 2, 1, 0, 1, 1, 2, 2, 0, 2, 1, 1, 3,
        2, 0, 1, 2, 2, 0, 0, 0, 3, 1, 3, 3, 2, 0, 3, 3, 1, 1, 3, 1, 2, 0,
        0, 2])
```

```
In [22]: X[y_means==0]
```

```

Out[22]: array([[ 5.9 , 113. ],
 [ 5.45, 110. ],
 [ 5.88, 109. ],
 [ 5.79, 110. ],
 [ 6.1 , 110. ],
 [ 5.71, 108. ],
 [ 5.5 , 111. ],
 [ 6.05, 111. ],
 [ 5.84, 113. ],
 [ 5.43, 106. ],
 [ 6.01, 112. ],
 [ 5.32, 106. ],
 [ 5.91, 108. ],
 [ 5.57, 113. ],
 [ 6.4 , 108. ],
 [ 5.67, 109. ],
 [ 6.05, 108. ],
 [ 5.85, 111. ],
 [ 5.87, 109. ],
 [ 6.02, 104. ],
 [ 5.77, 111. ],
 [ 6.06, 109. ],
 [ 5.55, 109. ],
 [ 5.81, 112. ],
 [ 5.47, 111. ],
 [ 5.74, 109. ],
 [ 5.8 , 108. ],
 [ 5.88, 110. ],
 [ 5.91, 109. ],
 [ 5.67, 111. ],
 [ 5.74, 108. ],
 [ 5.69, 109. ],
 [ 6.05, 109. ],
 [ 6.14, 111. ],
 [ 5.74, 112. ],
 [ 5.94, 109. ],
 [ 5.86, 111. ],
 [ 6.38, 107. ],
 [ 6.61, 111. ],
 [ 6.04, 110. ],
 [ 6.24, 108. ],
 [ 6.1 , 109. ],
 [ 5.8 , 110. ],
 [ 5.87, 108. ],
 [ 5.97, 108. ],
 [ 6.17, 110. ],
 [ 6.01, 107. ],
 [ 6.33, 111. ],
 [ 5.85, 112. ],
 [ 6.23, 108. ]])

```

```

In [23]: X[y_means==0,0],X[y_means==0,1]

```

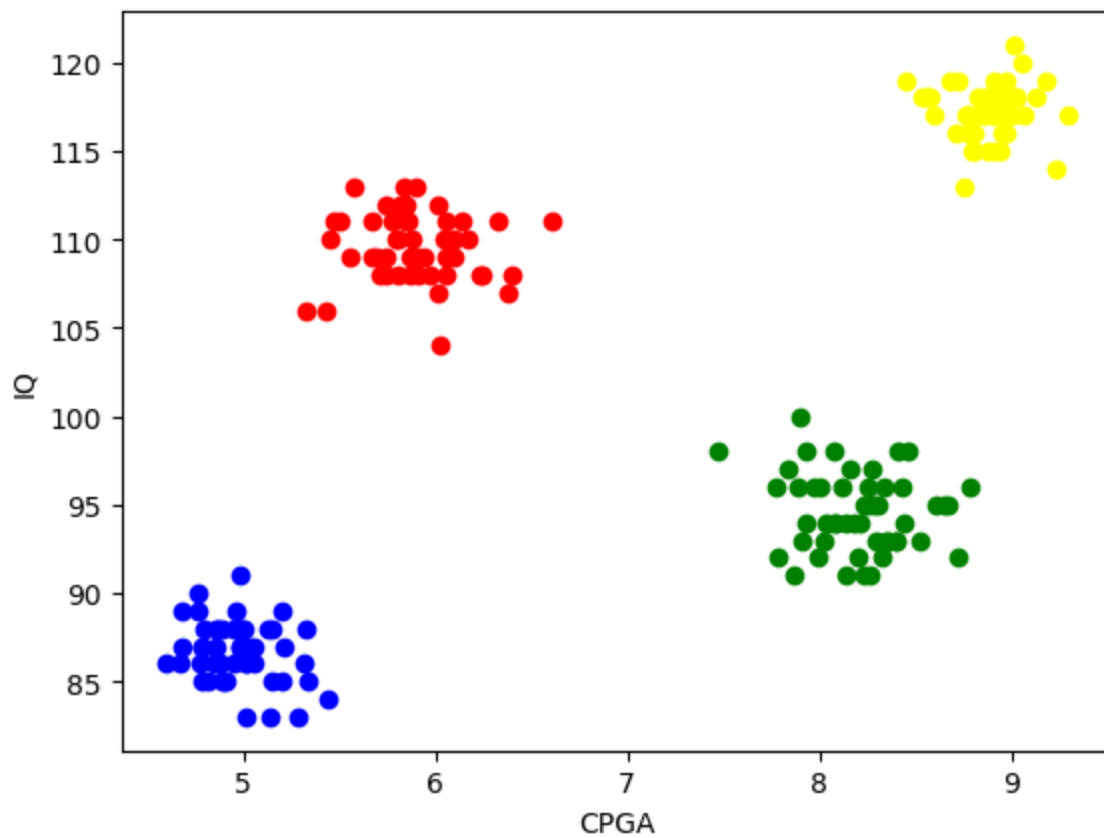
```

Out[23]: (array([5.9 , 5.45, 5.88, 5.79, 6.1 , 5.71, 5.5 , 6.05, 5.84, 5.43, 6.01,
 5.32, 5.91, 5.57, 6.4 , 5.67, 6.05, 5.85, 5.87, 6.02, 5.77, 6.06,
 5.55, 5.81, 5.47, 5.74, 5.8 , 5.88, 5.91, 5.67, 5.74, 5.69, 6.05,
 6.14, 5.74, 5.94, 5.86, 6.38, 6.61, 6.04, 6.24, 6.1 , 5.8 , 5.87,
 5.97, 6.17, 6.01, 6.33, 5.85, 6.23]),
 array([113., 110., 109., 110., 110., 108., 111., 111., 113., 106., 112.,
 106., 108., 113., 108., 109., 108., 111., 109., 104., 111., 109.,
 109., 112., 111., 109., 108., 110., 109., 111., 108., 109., 109.,
 111., 112., 109., 111., 107., 111., 110., 108., 109., 110., 108.,
 108., 110., 107., 111., 112., 108.]))

```

```
In [24]: plt.scatter(X[y_means==0,0],X[y_means==0,1],color='red')
plt.scatter(X[y_means==1,0],X[y_means==1,1],color='blue')
plt.scatter(X[y_means==2,0],X[y_means==2,1],color='yellow')
plt.scatter(X[y_means==3,0],X[y_means==3,1],color='green')
plt.xlabel('CPGA')
plt.ylabel('IQ')
```

Out[24]: Text(0, 0.5, 'IQ')



In []: