In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```

df = pd.read_csv('student_clustering.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	cgpa	iq
0	5.13	88
1	5.90	113
2	8.36	93
3	8.27	97
4	5.45	110

In [6]:

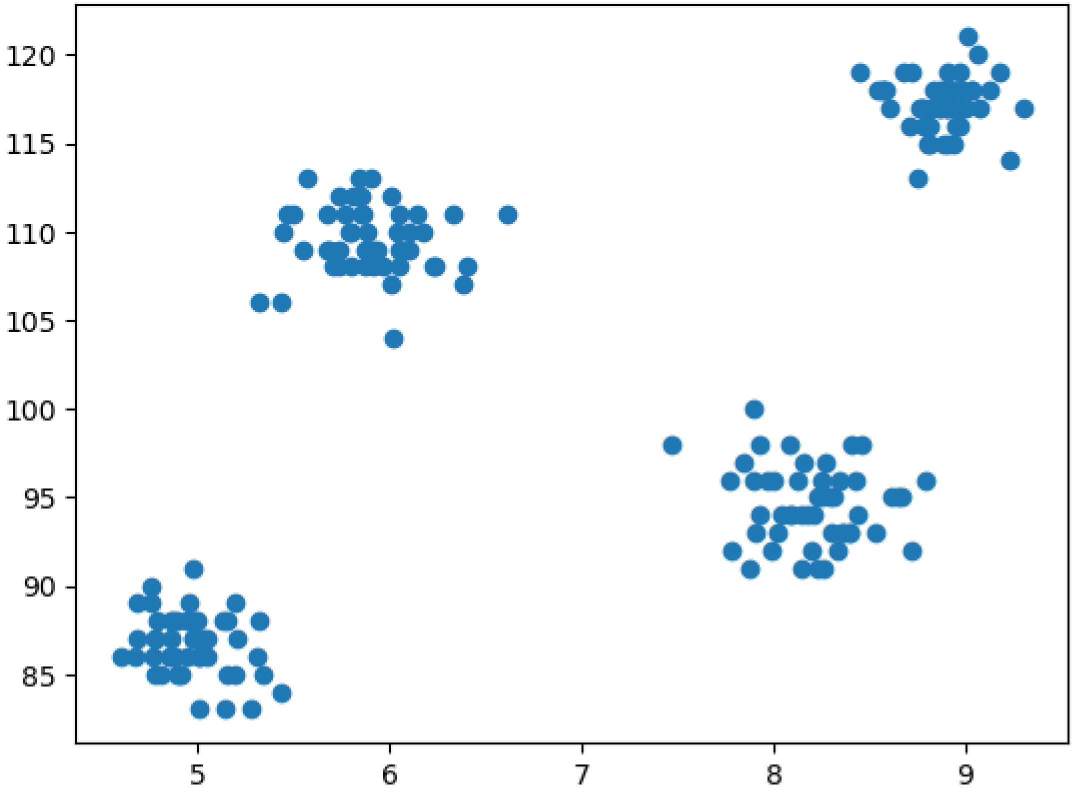
```
df.shape
```

Out[6]:

```
(200, 2)
```

In [7]:
```
import matplotlib.pyplot as plt
plt.scatter(df['cgpa'],df['iq'])
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x24e52b3fc10>
```



In [8]:
```
from sklearn.cluster import KMeans
wcss = []

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

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: UserWa
rning: KMeans is known to have a memory leak on Windows with MKL, when there ar
e less chunks than available threads. You can avoid it by setting the environme
nt variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [9]:

```
wcss
```

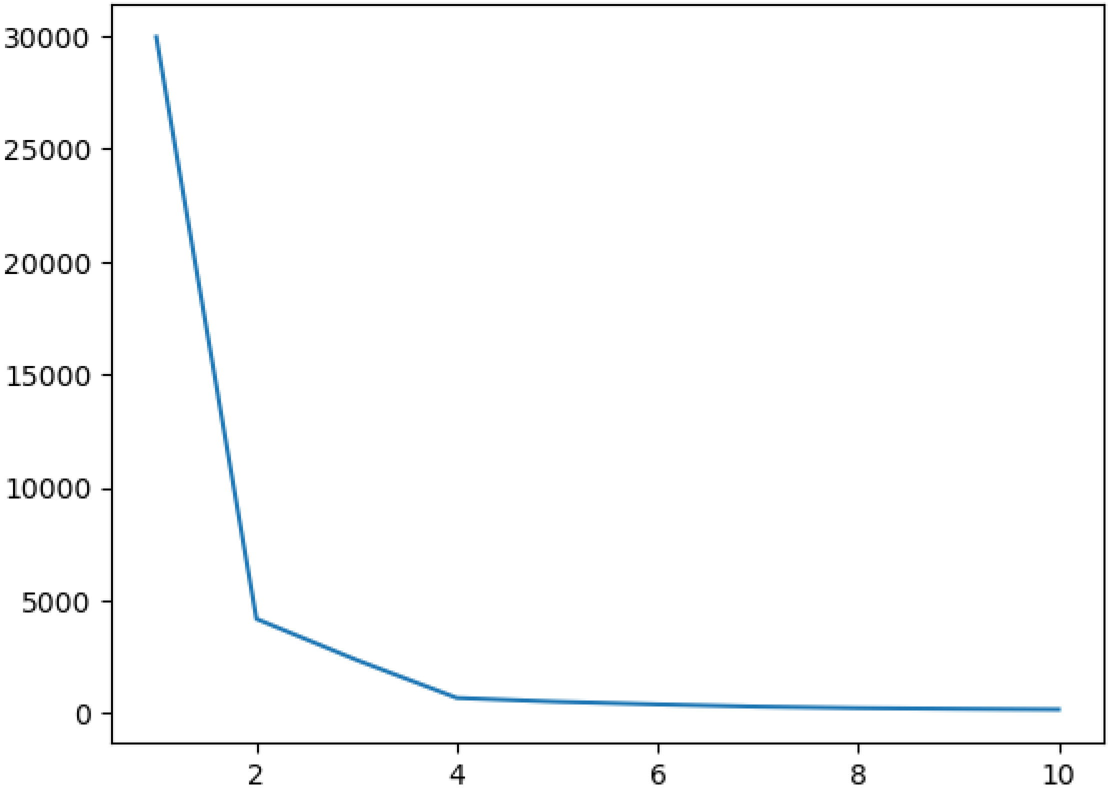
Out[9]:

```
[29957.898288,
 4184.14127,
 2362.7133489999997,
 681.96966,
 514.1616803171115,
 395.9605877691832,
 295.4391895943192,
 233.54082485509016,
 198.22433643678949,
 171.40590752168498]
```

In [10]:
```
plt.plot(range(1,11),wcss)
[]

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

Out[10]:
```
array([0, 1, 2, 2, 1, 1, 2, 3, 1, 2, 0, 1, 2, 0, 1, 2, 1, 2, 1, 1, 2, 0,
       2, 0, 0, 2, 0, 3, 2, 1, 3, 1, 3, 1, 2, 2, 3, 1, 0, 1, 0, 2, 2, 0,
       3, 3, 2, 1, 3, 1, 0, 0, 3, 2, 3, 1, 1, 3, 1, 3, 1, 2, 2, 3, 0, 3,
       2, 0, 1, 2, 1, 3, 2, 0, 1, 3, 1, 3, 0, 2, 2, 3, 1, 0, 3, 0, 3, 1,
       3, 1, 3, 3, 2, 0, 2, 2, 3, 2, 0, 3, 1, 0, 0, 3, 0, 0, 2, 0, 3, 3,
       2, 3, 1, 1, 2, 3, 2, 1, 3, 0, 0, 1, 2, 3, 2, 0, 2, 1, 0, 2, 2, 1,
       0, 0, 1, 3, 1, 0, 2, 2, 2, 0, 1, 0, 0, 3, 0, 3, 1, 0, 3, 0, 3, 3,
       0, 2, 1, 3, 1, 2, 0, 3, 1, 2, 3, 0, 1, 0, 0, 3, 3, 1, 3, 0, 0, 2,
       3, 1, 0, 3, 3, 1, 1, 1, 2, 0, 2, 2, 3, 1, 2, 2, 0, 0, 2, 0, 3, 1,
       1, 3])
```



In [11]:
```
X[y_means == 3,1]
```

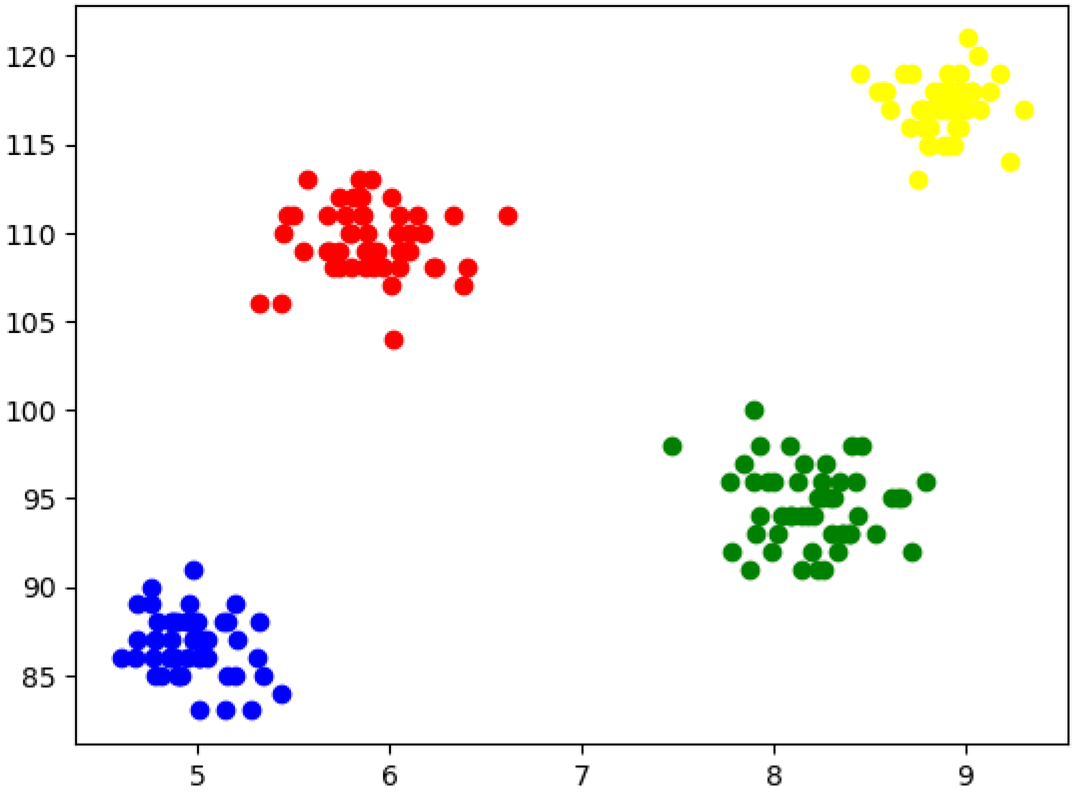
Out[11]:
```
array([115., 119., 117., 118., 118., 116., 116., 119., 116., 115., 115.,
       117., 118., 113., 116., 118., 117., 121., 116., 117., 117., 117.,
       114., 118., 118., 119., 118., 118., 117., 118., 117., 119., 118.,
       118., 117., 117., 117., 116., 118., 119., 117., 119., 120., 117.,
       115., 115., 117., 116., 118., 117.])
```

In [12]:

```
plt.scatter(X[y_means == 0,0],X[y_means == 0,1],color='blue')
plt.scatter(X[y_means == 1,0],X[y_means == 1,1],color='red')
plt.scatter(X[y_means == 2,0],X[y_means == 2,1],color='green')
plt.scatter(X[y_means == 3,0],X[y_means == 3,1],color='yellow')
```

Out[12]:

```
<matplotlib.collections.PathCollection at 0x24e56518bb0>
```



In [ ]: