In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [5]:

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

Out[5]:

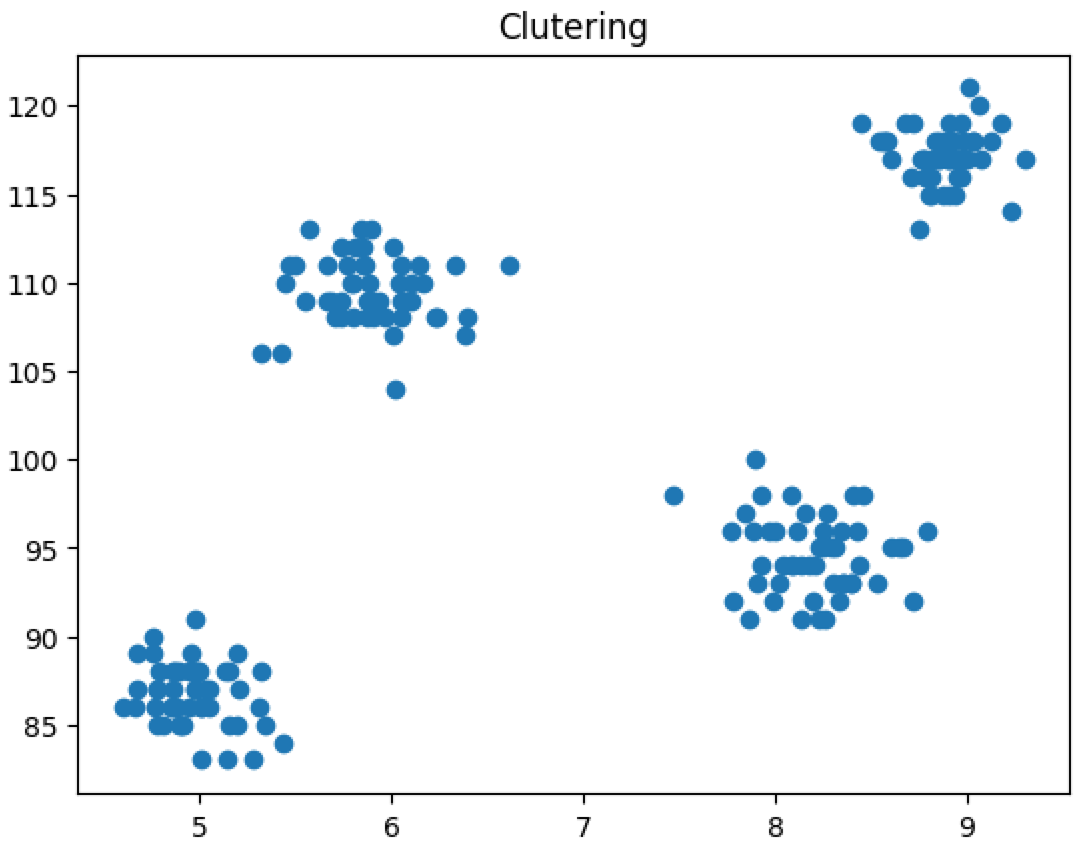
	cgpa	iq
0	5.13	88
1	5.90	113
2	8.36	93
3	8.27	97
4	5.45	110

In [7]:

```
plt.scatter(df['cgpa'],df['iq'])
plt.title('Clutering')
```

Out[7]:

```
Text(0.5, 1.0, 'Clutering')
```



In [11]:
```
from sklearn.cluster import KMeans
wcss = []
```

In [12]:

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

```
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
```

```
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [13]:

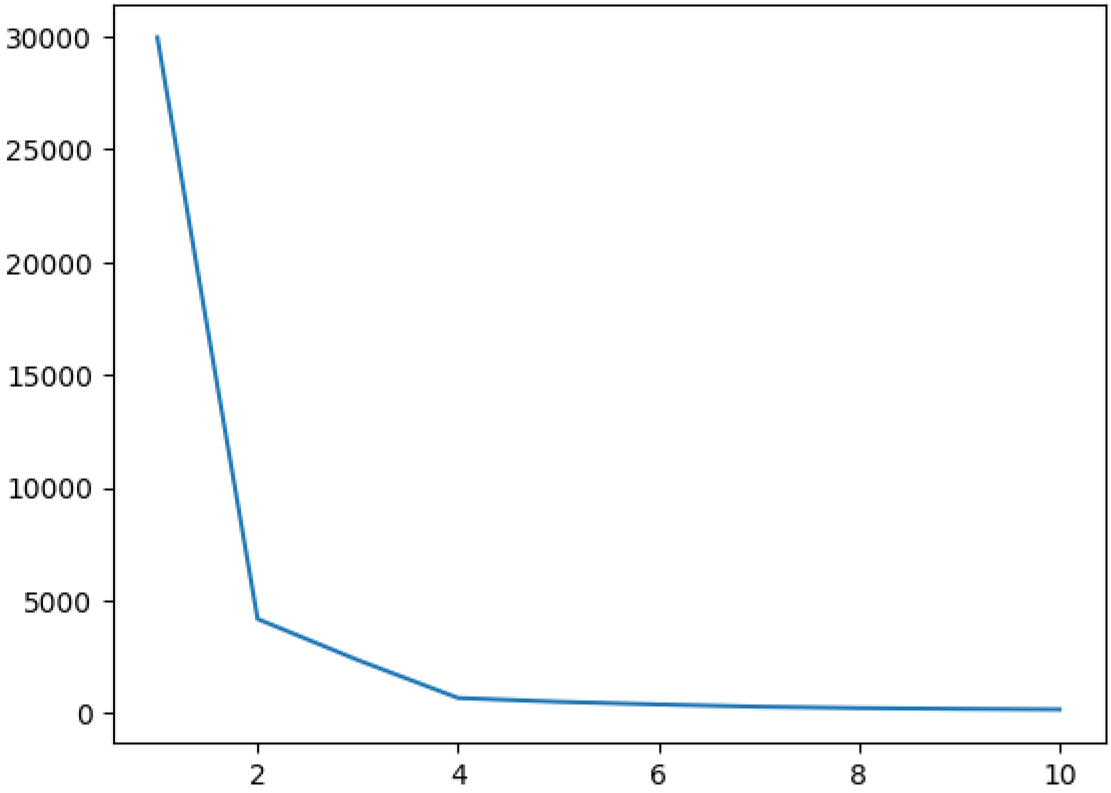
```
wcss
```

Out[13]:

```
[29957.898287999997,
 4184.14127,
 2362.7133490000006,
 681.96966,
 514.1616803171115,
 395.96058776918323,
 295.43918959431915,
 235.30768874397904,
 199.99120032567836,
 174.01644967366929]
```

In [14]:
```
plt.plot(range(1,11),wcss)
```

Out[14]:
```
[<matplotlib.lines.Line2D at 0x15163547b20>]
```



In [15]:
```
x=df.iloc[:,:].values
```

In [16]:
```
km=KMeans(n_clusters=4)
y_means=km.fit_predict(x)
```

```
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: 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
  warnings.warn(
C:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [17]:

```
y_means
```

Out[17]:

```
array([2, 0, 1, 1, 0, 0, 1, 3, 0, 1, 2, 0, 1, 2, 0, 1, 0, 1, 0, 0, 1, 2,
       1, 2, 2, 1, 2, 3, 1, 0, 3, 0, 3, 0, 1, 1, 3, 0, 2, 0, 2, 1, 1, 2,
       3, 3, 1, 0, 3, 0, 2, 2, 3, 1, 3, 0, 0, 3, 0, 3, 0, 1, 1, 3, 2, 3,
       1, 2, 0, 1, 0, 3, 1, 2, 0, 3, 0, 3, 2, 1, 1, 3, 0, 2, 3, 2, 3, 0,
       3, 0, 3, 3, 1, 2, 1, 1, 3, 1, 2, 3, 0, 2, 2, 3, 2, 2, 1, 2, 3, 3,
       1, 3, 0, 0, 1, 3, 1, 0, 3, 2, 2, 0, 1, 3, 1, 2, 1, 0, 2, 1, 1, 0,
       2, 2, 0, 3, 0, 2, 1, 1, 1, 2, 0, 2, 2, 3, 2, 3, 0, 2, 3, 2, 3, 3,
       2, 1, 0, 3, 0, 1, 2, 3, 0, 1, 3, 2, 0, 2, 2, 3, 3, 0, 3, 2, 2, 1,
       3, 0, 2, 3, 3, 0, 0, 0, 1, 2, 1, 1, 3, 0, 1, 1, 2, 2, 1, 2, 3, 0,
       0, 3])
```

In [18]:
```
x[y_means==0]
```

Out[18]:
```
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 [19]:
```
x[y_means==1]
```

Out[19]:
```
array([[  8.36,  93.  ],
       [  8.27,  97.  ],
       [  8.41,  98.  ],
       [  8.09,  94.  ],
       [  8.16,  97.  ],
       [  8.31,  95.  ],
       [  7.87,  91.  ],
       [  7.47,  98.  ],
       [  7.78,  92.  ],
       [  7.93,  98.  ],
       [  8.04,  94.  ],
       [  7.77,  96.  ],
       [  8.  ,  96.  ],
       [  8.43,  96.  ],
       [  8.02,  93.  ],
       [  8.14,  94.  ],
       [  8.12,  96.  ],
       [  8.34,  96.  ],
       [  8.65,  95.  ],
       [  8.53,  93.  ],
       [  8.29,  95.  ],
       [  7.93,  94.  ],
       [  8.72,  92.  ],
       [  8.14,  91.  ],
       [  8.2 ,  92.  ],
       [  8.67,  95.  ],
       [  8.18,  94.  ],
       [  8.61,  95.  ],
       [  7.99,  92.  ],
       [  8.08,  94.  ],
       [  8.26,  91.  ],
       [  8.25,  95.  ],
       [  8.4 ,  93.  ],
       [  7.84,  97.  ],
       [  8.08,  98.  ],
       [  8.25,  96.  ],
       [  8.3 ,  93.  ],
       [  7.9 , 100.  ],
       [  7.97,  96.  ],
       [  8.21,  94.  ],
       [  8.23,  95.  ],
       [  8.35,  93.  ],
       [  8.33,  92.  ],
       [  8.46,  98.  ],
       [  7.89,  96.  ],
       [  7.91,  93.  ],
       [  8.23,  91.  ],
       [  8.4 ,  93.  ],
       [  8.44,  94.  ],
       [  8.79,  96.  ]])
```

In [20]:
```
x[y_means==2]
```

Out[20]:
```
array([[ 5.13, 88.  ],
       [ 4.6 , 86.  ],
       [ 5.  , 88.  ],
       [ 4.86, 86.  ],
       [ 4.78, 87.  ],
       [ 4.96, 88.  ],
       [ 4.86, 87.  ],
       [ 5.44, 84.  ],
       [ 5.34, 85.  ],
       [ 5.31, 86.  ],
       [ 5.14, 83.  ],
       [ 4.95, 86.  ],
       [ 5.21, 87.  ],
       [ 4.91, 85.  ],
       [ 5.28, 83.  ],
       [ 5.15, 88.  ],
       [ 4.9 , 85.  ],
       [ 4.89, 88.  ],
       [ 5.05, 86.  ],
       [ 4.98, 91.  ],
       [ 5.01, 86.  ],
       [ 4.95, 88.  ],
       [ 4.96, 89.  ],
       [ 4.85, 86.  ],
       [ 4.76, 90.  ],
       [ 4.98, 87.  ],
       [ 4.78, 87.  ],
       [ 5.2 , 85.  ],
       [ 5.05, 87.  ],
       [ 5.01, 83.  ],
       [ 4.77, 86.  ],
       [ 4.68, 87.  ],
       [ 4.81, 85.  ],
       [ 5.03, 87.  ],
       [ 4.98, 87.  ],
       [ 5.32, 88.  ],
       [ 4.86, 88.  ],
       [ 4.89, 85.  ],
       [ 4.88, 86.  ],
       [ 5.01, 86.  ],
       [ 4.67, 86.  ],
       [ 5.15, 85.  ],
       [ 4.97, 88.  ],
       [ 4.87, 88.  ],
       [ 5.2 , 89.  ],
       [ 4.99, 88.  ],
       [ 4.79, 88.  ],
       [ 4.76, 89.  ],
       [ 4.78, 85.  ],
       [ 4.68, 89.  ]])
```

In [21]:

```
x[y_means==3]
```

Out[21]:

```
array([[  8.8 , 115.  ],
       [  9.18, 119.  ],
       [  8.86, 117.  ],
       [  8.83, 118.  ],
       [  8.56, 118.  ],
       [  8.96, 116.  ],
       [  8.78, 116.  ],
       [  8.45, 119.  ],
       [  8.79, 116.  ],
       [  8.81, 115.  ],
       [  8.88, 115.  ],
       [  9.07, 117.  ],
       [  8.92, 118.  ],
       [  8.75, 113.  ],
       [  8.71, 116.  ],
       [  8.86, 118.  ],
       [  9.3 , 117.  ],
       [  9.01, 121.  ],
       [  8.97, 116.  ],
       [  9.  , 117.  ],
       [  8.76, 117.  ],
       [  8.78, 117.  ],
       [  9.23, 114.  ],
       [  9.03, 118.  ],
       [  9.13, 118.  ],
       [  8.91, 119.  ],
       [  8.98, 118.  ],
       [  9.03, 118.  ],
       [  8.86, 117.  ],
       [  8.89, 118.  ],
       [  8.97, 117.  ],
       [  8.72, 119.  ],
       [  8.93, 118.  ],
       [  8.58, 118.  ],
       [  8.94, 117.  ],
       [  8.6 , 117.  ],
       [  8.77, 117.  ],
       [  8.81, 116.  ],
       [  8.54, 118.  ],
       [  8.97, 119.  ],
       [  8.91, 117.  ],
       [  8.68, 119.  ],
       [  9.06, 120.  ],
       [  8.9 , 117.  ],
       [  8.94, 115.  ],
       [  8.91, 115.  ],
       [  8.91, 117.  ],
       [  8.95, 116.  ],
       [  8.57, 118.  ],
       [  8.82, 117.  ]])
```

In [22]:
```
x[y_means==3,0]
```

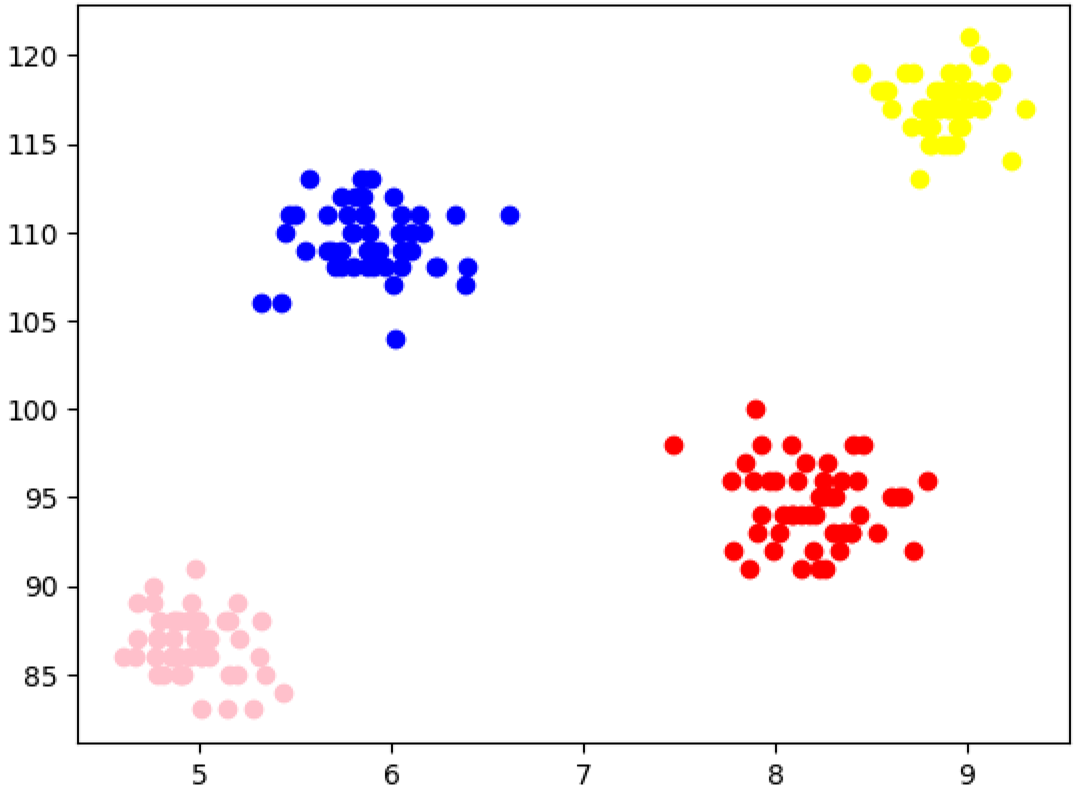
Out[22]:
```
array([8.8 , 9.18, 8.86, 8.83, 8.56, 8.96, 8.78, 8.45, 8.79, 8.81, 8.88,
       9.07, 8.92, 8.75, 8.71, 8.86, 9.3 , 9.01, 8.97, 9.  , 8.76, 8.78,
       9.23, 9.03, 9.13, 8.91, 8.98, 9.03, 8.86, 8.89, 8.97, 8.72, 8.93,
       8.58, 8.94, 8.6 , 8.77, 8.81, 8.54, 8.97, 8.91, 8.68, 9.06, 8.9 ,
       8.94, 8.91, 8.91, 8.95, 8.57, 8.82])
```

In [23]:
```
x[y_means==3,1]
```

Out[23]:
```
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 [24]:
```
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='pink')
plt.scatter(x[y_means==3,0],x[y_means==3,1],color='yellow')
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x1516358aec0>
```



In [ ]: