SUYASH GOUR A4-66 ECS

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

```
df = pd.read_csv('student_clustering.csv')
print("The shape of the dataset is: ", df.shape)
df.head()
```

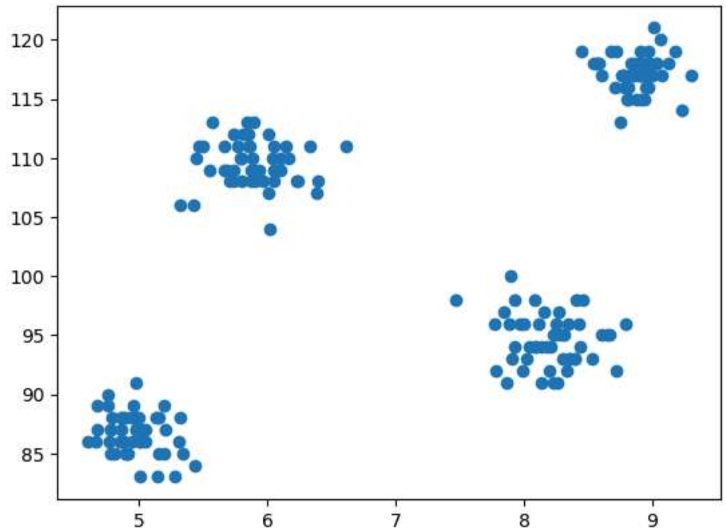
```
The shape of the dataset is:  (200, 2)
```

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

Next steps: Generate code with df | View recommended plots | New interactive sheet

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

```
<matplotlib.collections.PathCollection at 0x7bcd9b748100>
```



```
from sklearn.cluster import KMeans
```

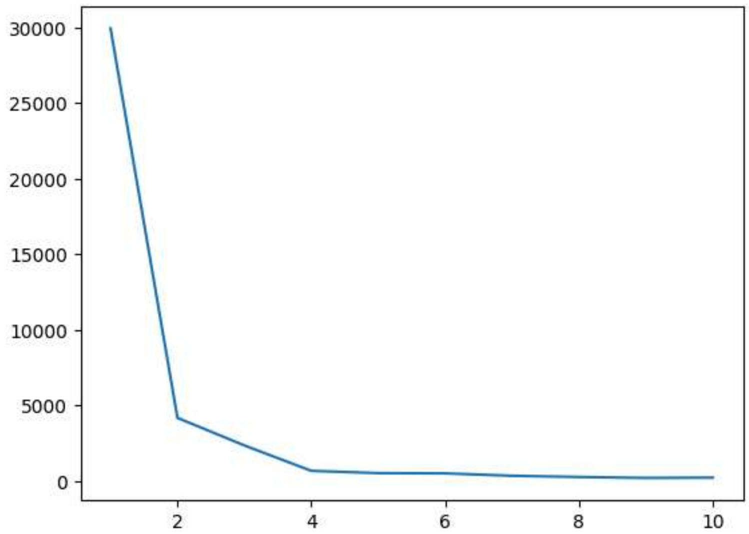
```
wcss = []
for i in range(1,11):
  km = KMeans(n_clusters=i)
  km.fit_predict(df)
  wcss.append(km.inertia_)
```

```
wcss
```

```
[29957.898287999997,
 4184.141269999999,
 2362.7133490000015,
 681.9696600000001,
 530.4298616986187,
 515.3871254832748,
 354.0352530907326,
 274.2294625999665,
 213.17281656519154,
 233.96773571921526]
```

```
plt.plot(range(1,11),wcss)
```

```
[<matplotlib.lines.Line2D at 0x7bcd921af430>]
```



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

```
y_means
```

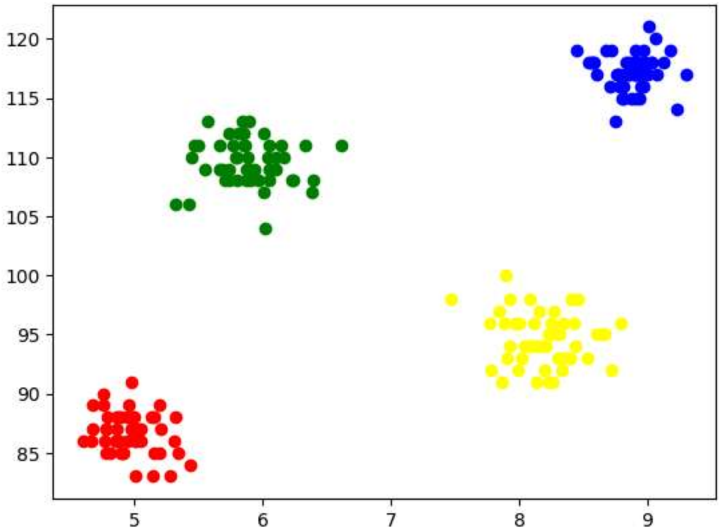
```
array([1, 2, 3, 3, 2, 2, 3, 0, 2, 3, 1, 2, 3, 1, 2, 3, 2, 3, 2, 2, 3, 1,
       3, 1, 1, 3, 1, 0, 3, 2, 0, 2, 0, 2, 3, 3, 0, 2, 1, 2, 1, 3, 3, 1,
       0, 0, 3, 2, 0, 2, 1, 1, 0, 3, 0, 2, 2, 0, 2, 0, 2, 3, 3, 0, 1, 0,
       3, 1, 2, 3, 2, 0, 3, 1, 2, 0, 2, 0, 1, 3, 3, 0, 2, 1, 0, 1, 0, 2,
       0, 2, 0, 0, 3, 1, 3, 3, 0, 3, 1, 0, 2, 1, 1, 0, 1, 1, 3, 1, 0, 0,
       3, 0, 2, 2, 3, 0, 3, 2, 0, 1, 1, 2, 3, 0, 3, 1, 3, 2, 1, 3, 3, 2,
       1, 1, 2, 0, 2, 1, 3, 3, 3, 1, 2, 1, 1, 0, 1, 0, 2, 1, 0, 1, 0, 0,
       1, 3, 2, 0, 2, 3, 1, 0, 2, 3, 0, 1, 2, 1, 1, 0, 0, 2, 0, 1, 1, 3,
       0, 2, 1, 0, 0, 2, 2, 2, 3, 1, 3, 3, 0, 2, 3, 3, 1, 1, 3, 1, 0, 2,
       2, 0], dtype=int32)
```

```
X[y_means == 3,1]
```

```
array([ 93.,  97.,  98.,  94.,  97.,  95.,  91.,  98.,  92.,  98.,  94.,
        96.,  96.,  96.,  93.,  94.,  96.,  96.,  95.,  93.,  95.,  94.,
        92.,  91.,  92.,  95.,  94.,  95.,  92.,  94.,  91.,  95.,  93.,
        97.,  98.,  96.,  93., 100.,  96.,  94.,  95.,  93.,  92.,  98.,
        96.,  93.,  91.,  93.,  94.,  96.])
```

```
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')
```

```
<matplotlib.collections.PathCollection at 0x7bcd915e0f70>
```

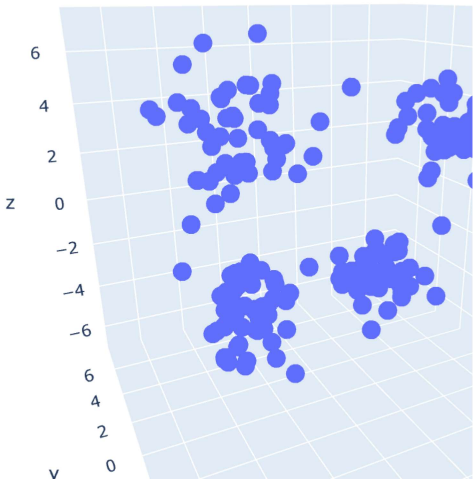


```
from sklearn.datasets import make_blobs

centroids = [(-5,-5,5),(5,5,-5),(3.5,-2.5,4),(-2.5,2.5,-4)]
cluster_std = [1,1,1,1]

X,y = make_blobs(n_samples=200,cluster_std=cluster_std,centers=centroids,n_features=3,random_state=1)
```

```
X
```

```
            [-3.37565464, -5.61175641,  4.47182825],
            [-2.37162301,  4.26041518, -3.03346075],
            [ 1.81594001, -3.6601701 ,  5.35010682],
            [ 5.04366899,  4.77368576, -3.66854289],
            [-4.19813897, -4.9534327 ,  4.81343023],
            [ 5.1340482 ,  6.20205486, -4.71525189],
            [ 3.39320601, -1.04857074,  3.38196315],
            [ 4.34086156, -2.60288722,  5.14690038],
            [-0.80619089,  2.69686978, -3.83013074],
            [-5.62353073, -4.47942366,  3.85565861],
            [ 5.56578332, -3.97115693,  3.1698281 ],
```

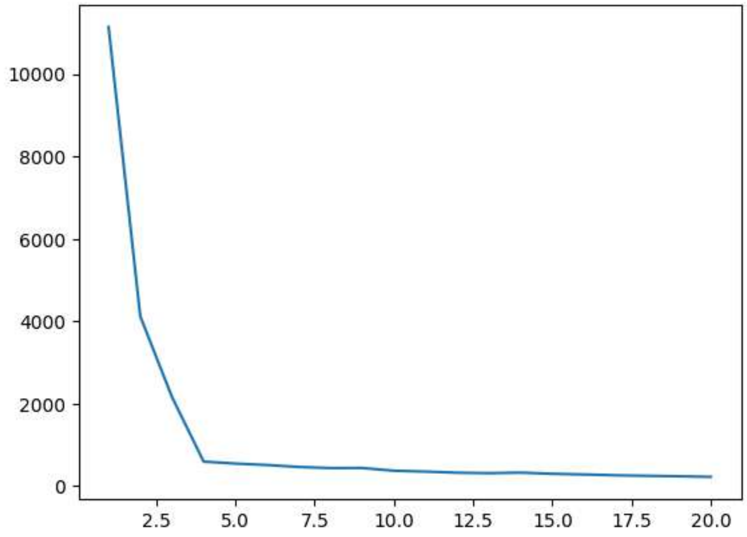
```
import plotly.express as px
fig = px.scatter_3d(x=X[:,0], y=X[:,1], z=X[:,2])
fig.show()
```



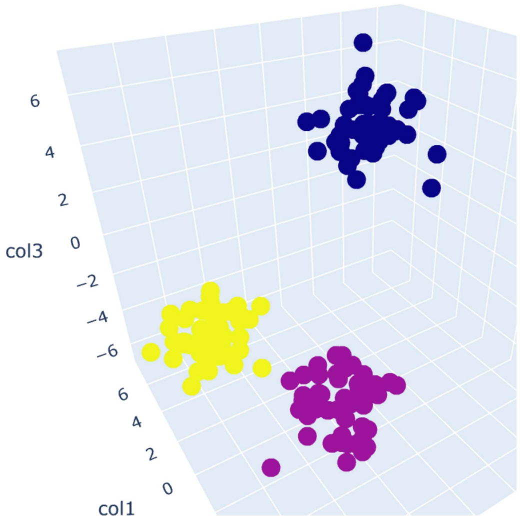
```
wcss = []
for i in range(1,21):
    km = KMeans(n_clusters=i)
    km.fit_predict(X)
    wcss.append(km.inertia_)
```

```
plt.plot(range(1,21),wcss)
```

```
[<matplotlib.lines.Line2D at 0x7bcd8cdca5f0>]
```



```
km = KMeans(n_clusters=4)
y_pred = km.fit_predict(X)


df = pd.DataFrame()

df['col1'] = X[:,0]
df['col2'] = X[:,1]
df['col3'] = X[:,2]
df['label'] = y_pred



fig = px.scatter_3d(df,x='col1', y='col2', z='col3',color='label')
fig.show()
```



Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.