In [15]:
```
# check if the plotly is installed or not
!pip install plotly
```

```
Requirement already satisfied: plotly in c:\programdata\anaconda3\lib\site-pa
ckages (5.12.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\programdata\anaconda3\li
b\site-packages (from plotly) (8.1.0)
```

In [16]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
import plotly.graph_objs as go
from plotly import tools
from plotly.subplots import make_subplots
import plotly.offline as py
```

In [17]:
```
data = pd.read_csv('C:/Users\DITU/Desktop/iriss.csv')
```

In [18]:
```
data.head()
```

Out[18]:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [19]:
```
data.describe()
```

Out[19]:

	5.1	3.5	1.4	0.2
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [20]:
```
data.describe().T
```

Out[20]:

	count	mean	std	min	25%	50%	75%	max
5.1	149.0	5.848322	0.828594	4.3	5.1	5.8	6.4	7.9
3.5	149.0	3.051007	0.433499	2.0	2.8	3.0	3.3	4.4
1.4	149.0	3.774497	1.759651	1.0	1.6	4.4	5.1	6.9
0.2	149.0	1.205369	0.761292	0.1	0.3	1.3	1.8	2.5

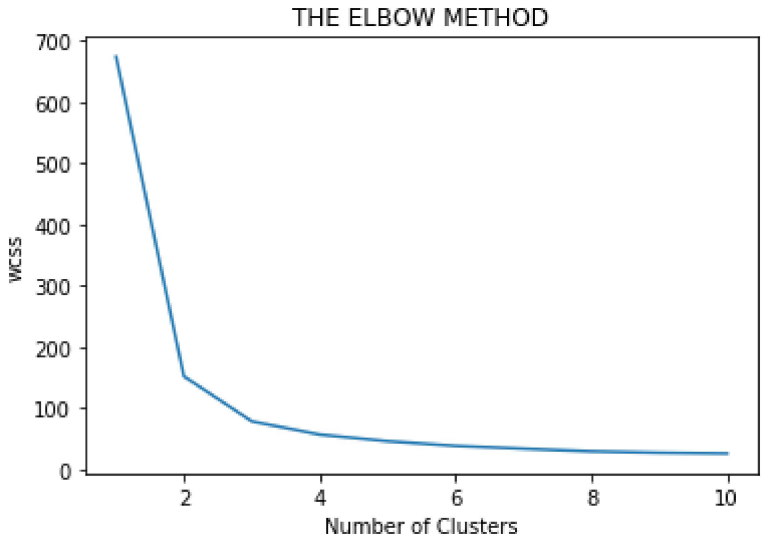
In [21]:
```
data['Iris-setosa'].unique()
```

Out[21]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [22]:
```
X=data.iloc[:,[0,1,2,3]].values
```

In [10]:
```
wcss = []
for i in range(1,11):
    kmeans =KMeans(n_clusters = i, init = 'k-means++', max_iter=200, n_init=10
,random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

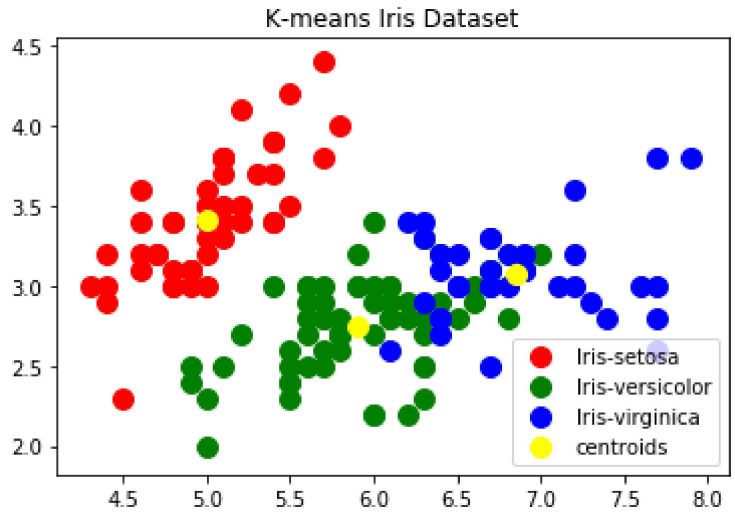
In [11]:
```
plt.plot(range(1,11),wcss)
plt.title('THE ELBOW METHOD')
plt.xlabel('Number of Clusters')
plt.ylabel('wcss')
plt.show()
```



In [12]:
```
kmeans =KMeans(n_clusters = 3, init = 'k-means++', max_iter=200, n_init = 10,r
andom_state = 0)
y_kmeans = kmeans.fit_predict(X)
```

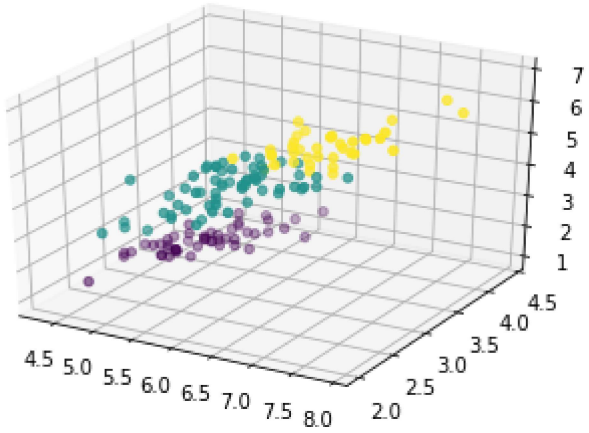
In [13]:
```
plt.scatter(X[y_kmeans == 0,0],X[y_kmeans == 0,1], s = 100, c = 'red', label = 'Iris-setosa')
plt.scatter(X[y_kmeans == 1,0],X[y_kmeans == 1,1], s = 100, c = 'green', label = 'Iris-versicolor')
plt.scatter(X[y_kmeans == 2,0],X[y_kmeans == 2,1], s = 100, c = 'blue', label = 'Iris-virginica')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1], s = 100, c= 'yellow', label = 'centroids')
plt.title('K-means Iris Dataset')

plt.legend()
```

Out[13]: <matplotlib.legend.Legend at 0x1a4cf556630>



In [23]:
```
# Plot the data in 3D space, color-coded by cluster
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=kmeans.labels_, cmap='viridis')
```

Out[23]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1a4cf63dd30>



In [26]:
```
# Add labels for the 3 species of iris
species = ['Setosa', 'Versicolor', 'Virginica']
for i in range(3):
    ax.text3D(np.mean(X[kmeans.labels_ == i, 0]), np.mean(X[kmeans.labels_ ==
i, 1]), np.mean(X[kmeans.labels_ == i, 2]), species[i], color='red', fontsize=
16)
    plt.show()
```

In [25]:

In [ ]: