THE SPARKS FOUNDATION

DATA SCIENCE AND BUSINESS ANALYTICS INTERN ### ### GRIPDECEMBER22 ### ### DECEMBER 2022

MRIDUL KAPOOR ### ### mridul.kapoor2002@gmail.com

TASK-2 ### ### PREDICTION USING UNSUPERVISED MACHINE LEARNING

In [ ]:
```
# IMPORTING MODULES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
import seaborn as sns
```

In [ ]:
```
data_file = pd.read_csv("C:/Users/Mridul_Work/Desktop/TSF_GripDecember22_Mridul Kapoor/Task_2/Iris.csv")
print("\nData imported successfully")

print("\nFirst five rows\n",data_file.head())

print("\nLast five rows\n",data_file.tail())
```

```
Data imported successfully

First five rows
    Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa

Last five rows
       Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica
```

In [ ]:
```
data_file.isnull().sum()
```

Out[ ]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

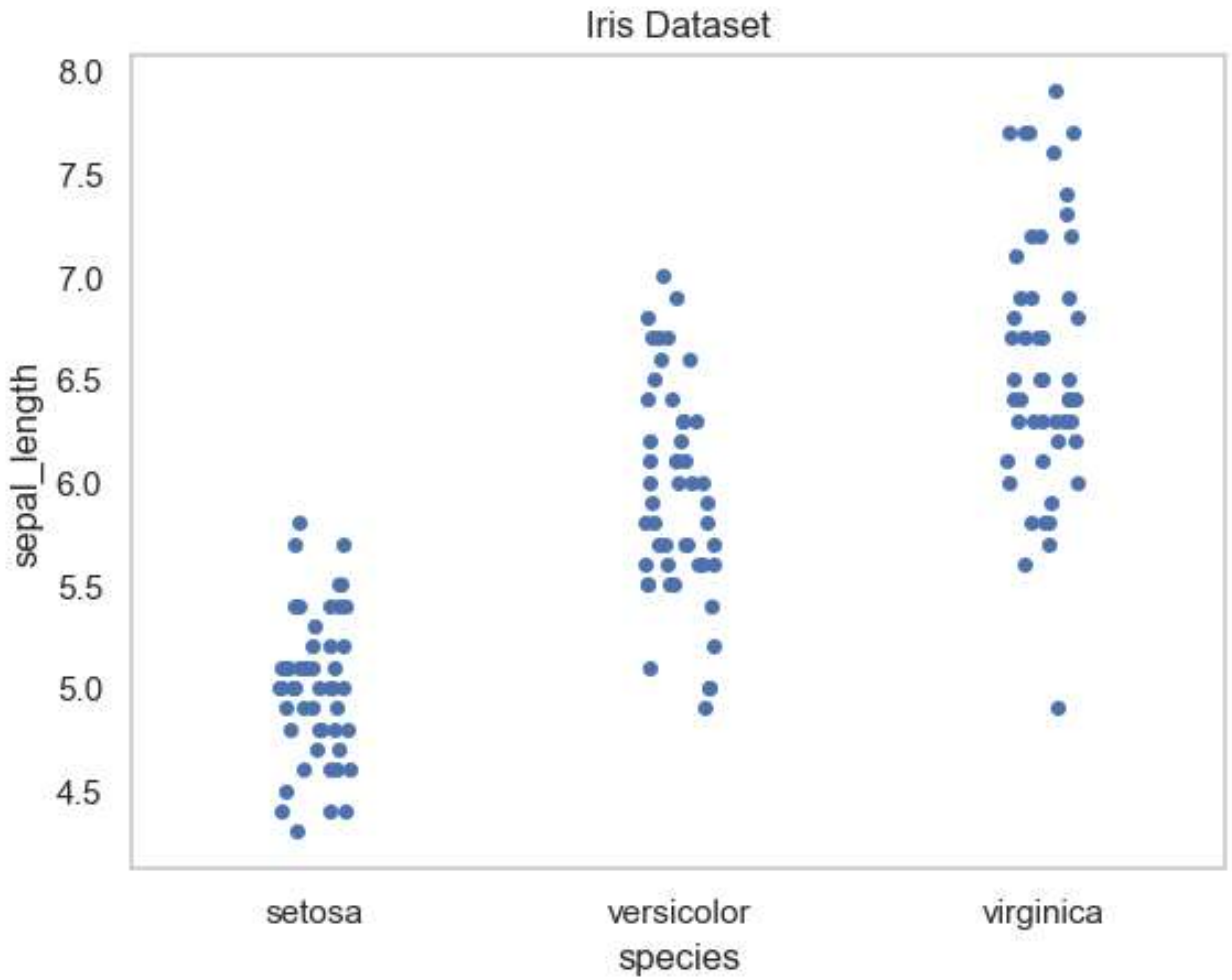
In [ ]:
```
print(data_file.Species.nunique())
print(data_file.Species.value_counts())
```

```
3
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

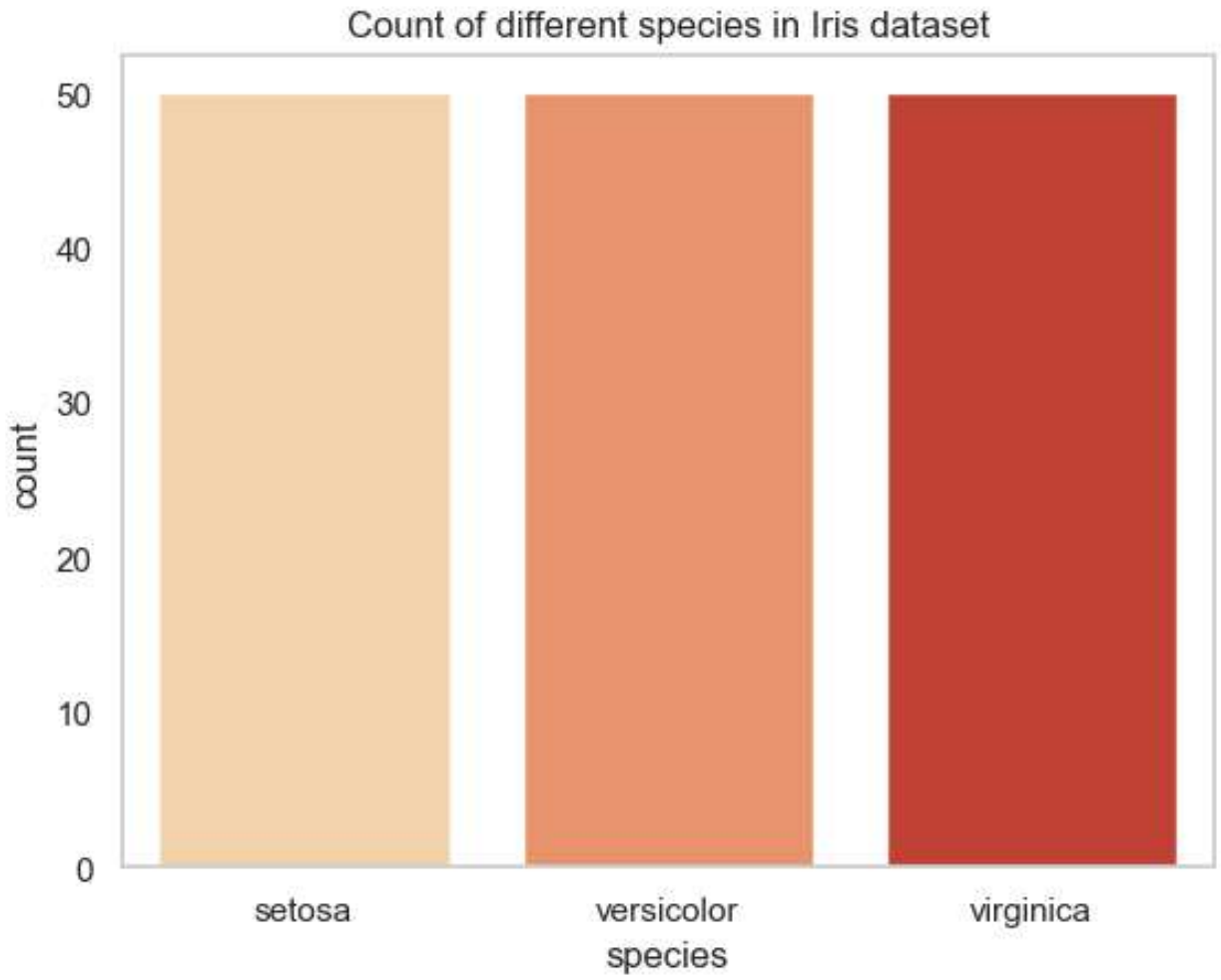
In [ ]:

```python
# DATA VISUALIZATION

print("\nDot Plot")

sns.set(style = 'whitegrid')
dataset_iris = sns.load_dataset('iris')
axis = sns.stripplot(x ='species',y = 'sepal_length',data = dataset_iris)

plt.title('Iris Dataset')
plt.grid(False)
plt.show()


print("\nCount Plot")

sns.countplot(x='species', data=dataset_iris, palette="OrRd")

plt.title("Count of different species in Iris dataset")
plt.grid(False)
plt.show()
```

```
Dot Plot
```


Iris Dataset
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
sepal_length
setosa
versicolor
virginica
species

Count Plot

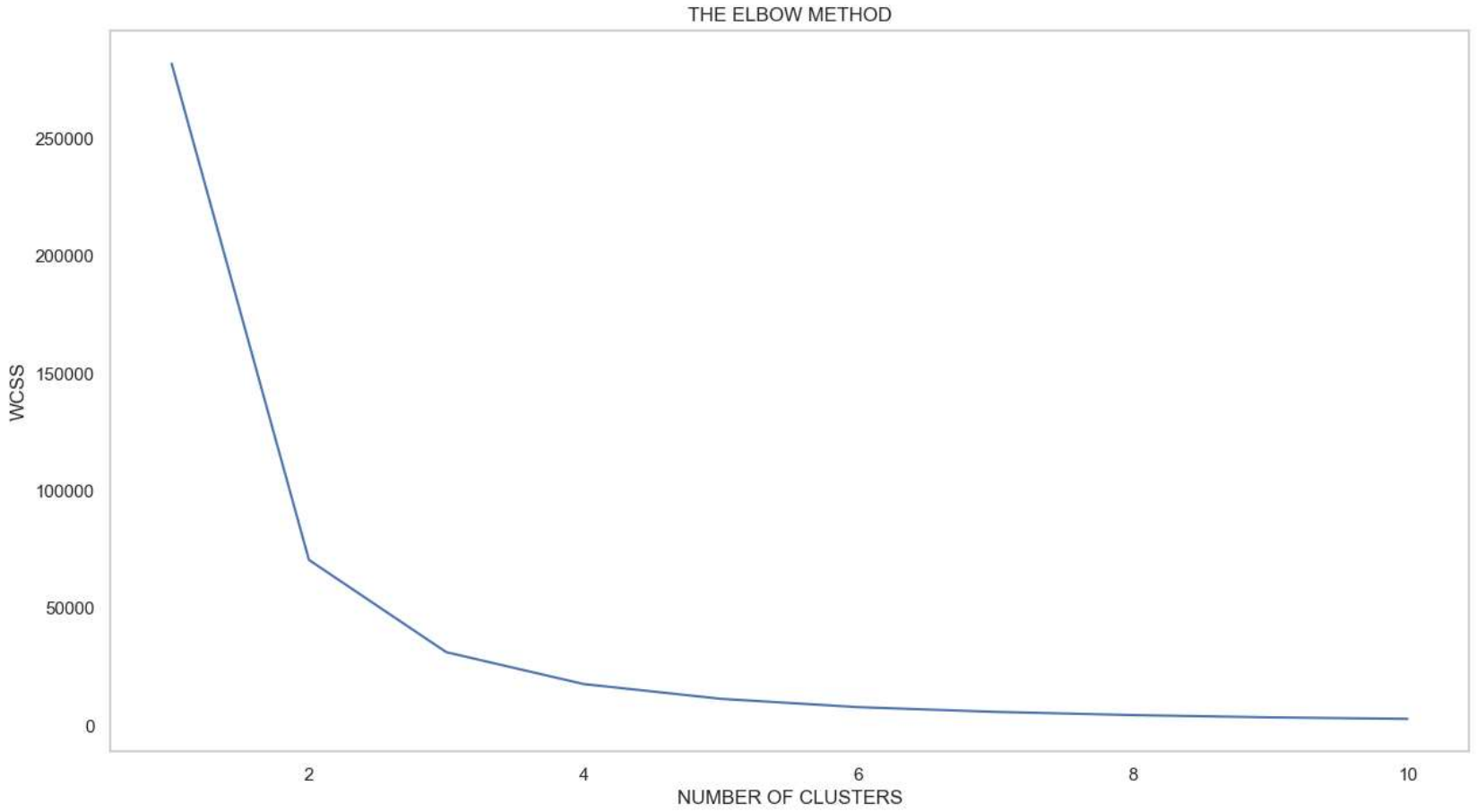


In [ ]:
```
# FINDING OPTIMUM NUMBER OF CLUSTERS FOR K-MEANS
x = data_file.iloc[:,[0,1,2,3]].values

from sklearn.cluster import KMeans
wcss = []

for i in range(1,11):

    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(x)
```

```
    wcss.append(kmeans.inertia_)

    print("k: {} ; wcss: {}".format(i,kmeans.inertia_))
```

```
k: 1 ; wcss: 281831.54466666665
k: 2 ; wcss: 70581.3808
k: 3 ; wcss: 31320.711199999998
k: 4 ; wcss: 17758.792503556186
k: 5 ; wcss: 11468.968747023808
k: 6 ; wcss: 7921.863473076924
k: 7 ; wcss: 5911.632365518541
k: 8 ; wcss: 4541.979023391813
k: 9 ; wcss: 3571.911095588235
k: 10 ; wcss: 2943.933100840336
```

In [ ]:
```
# Plotting the results onto a line graph, allowing us to observe 'The elbow'

plt.figure(figsize=(15,8))
plt.plot(range(1,11),wcss)
plt.title('THE ELBOW METHOD')
plt.xlabel('NUMBER OF CLUSTERS')
plt.ylabel('WCSS')
plt.grid(False)
plt.show()
```

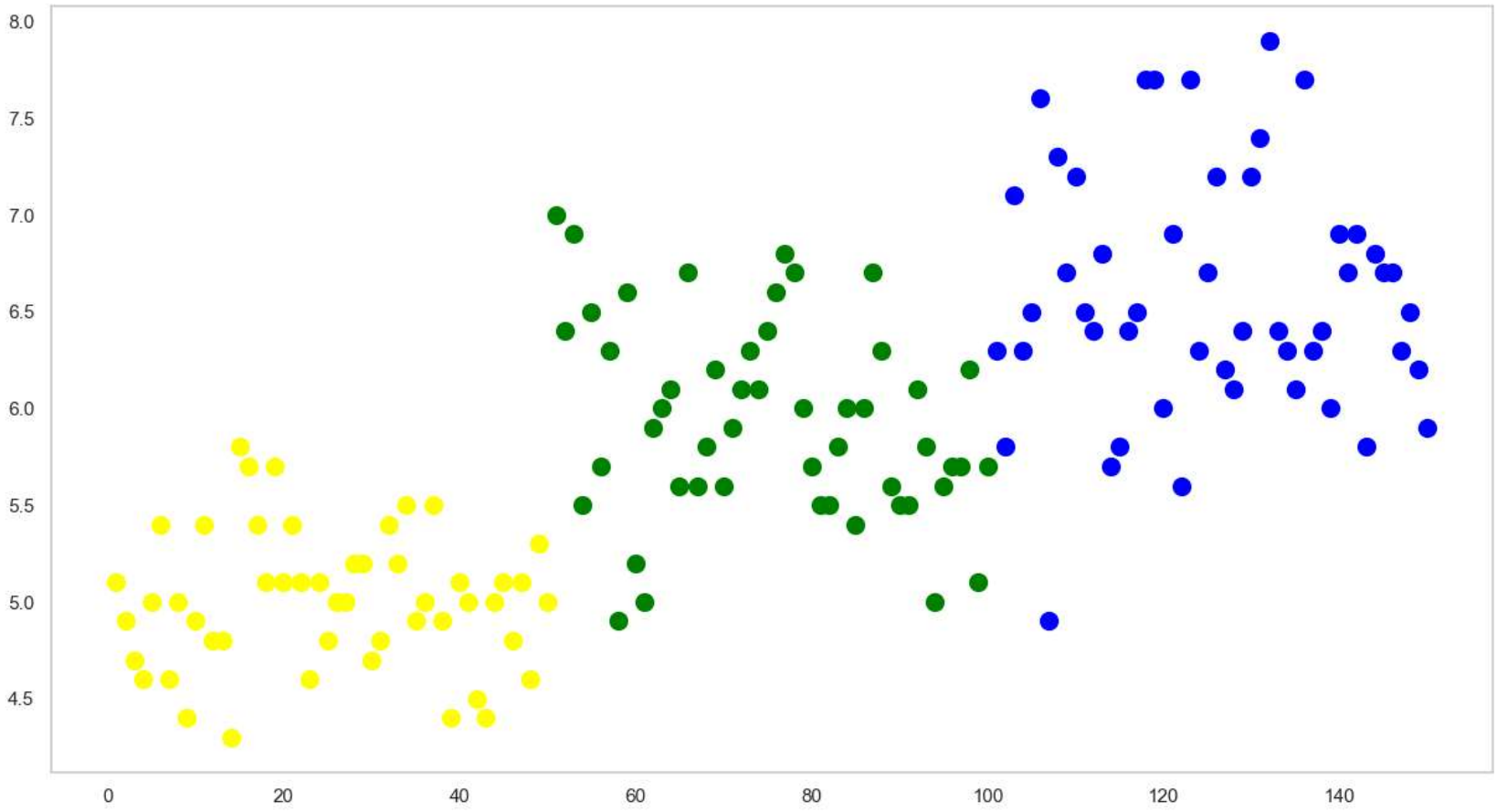


In [ ]:
```
#APPLYING K-MEANS -- K-MEANS CLASSFIER

kmeans = KMeans(n_clusters = 3, init = 'k-means++',max_iter = 300, n_init = 10, random_state = 0)

y_kmeans = kmeans.fit_predict(x)
```

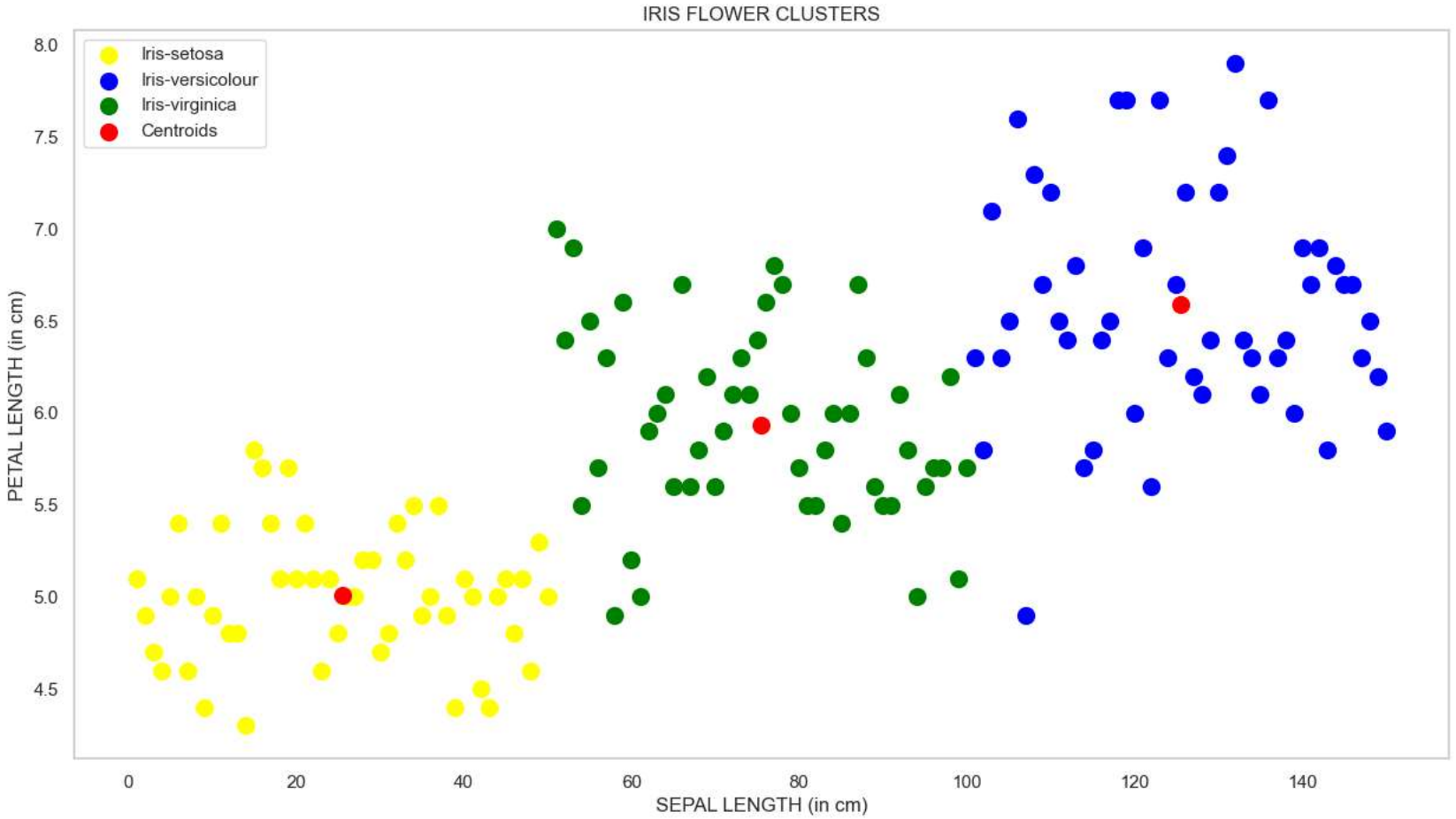
In [ ]:
```
# VISUALIZING THE CLUSTERS

print("\nWithout Plotting the Centroids of each species")

plt.figure(figsize=(15,8))
```

```
plt.scatter(x[y_kmeans==0,0],x[y_kmeans==0,1],s=100,c='yellow',label='Iris-setosa')
plt.scatter(x[y_kmeans==1,0],x[y_kmeans==1,1],s=100,c='blue',label='Iris-versicolour')
plt.scatter(x[y_kmeans==2,0],x[y_kmeans==2,1],s=100,c='green',label='Iris-virginica')

plt.grid(False)
plt.show()

##Plotting the centroids of the clusters
print("\n After Plotting the Centroids of all species (highlighted in red)")
plt.figure(figsize=(15,8))

plt.scatter(x[y_kmeans==0,0],x[y_kmeans==0,1],s=100,c='yellow',label='Iris-setosa')
plt.scatter(x[y_kmeans==1,0],x[y_kmeans==1,1],s=100,c='blue',label='Iris-versicolour')
plt.scatter(x[y_kmeans==2,0],x[y_kmeans==2,1],s=100,c='green',label='Iris-virginica')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=100,c='red',label='Centroids')

plt.title('IRIS FLOWER CLUSTERS')
plt.xlabel('SEPAL LENGTH (in cm)')
plt.ylabel('PETAL LENGTH (in cm)')
plt.legend()

plt.grid(False)
plt.show()
```

```
Without Plotting the Centroids of each species
```



After Plotting the Centroids of all species (highlighted in red)


IRIS FLOWER CLUSTERS
Iris-setosa
Iris-versicolour
Iris-virginica
Centroids
PETAL LENGTH (in cm)
SEPAL LENGTH (in cm)
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
0
20
40
60
80
100
120
140

END