# ASSIGNMENT 3

**Kulsoom Khurshid**

**SP20-BCS-044**

In [1]:
```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd
```

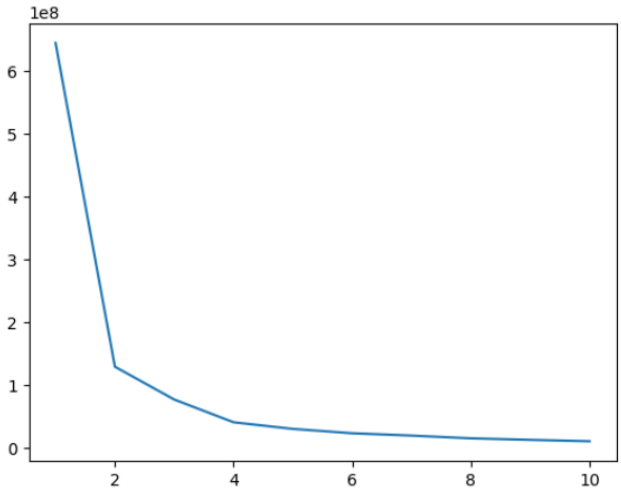
In [2]:
```
df = pd.read_csv("cars.csv")
wcss = []
for i in range(1,11):
         km = KMeans(n_clusters=i)
         km.fit_predict(df)
         wcss.append(km.inertia_)

wcss
```

```
C:\Users\sa\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_N
UM_THREADS=1.
  warnings.warn(
```

Out[2]:
```
[645268746.0425,
 129683516.17501615,
 77368714.72161247,
 41114884.26392993,
 30620446.798656296,
 23637512.45157393,
 19954185.3958014,
 15601779.295683932,
 13159105.904521989,
 10890026.132061496]
```

In [3]:
```
plt.plot(range(1,11),wcss)
```

Out[3]:
```
[<matplotlib.lines.Line2D at 0x27e4afa3340>]
```




In [4]:
```
from sklearn.model_selection import train_test_split
```

In [5]:
```
x=df.drop('origin', axis='columns')
#print(x)
```

In [6]:
```
y=df['origin']
#print(y)
```

In [7]:
```
"""X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.20, random_state=42)"""
kmeans = KMeans(n_clusters=5)
kmeans.fit(df)
```

Out[7]:
```
KMeans(n_clusters=5)
```

In [8]:
```
labels = kmeans.labels_
```

In [9]:
```
correct_labels = sum(y == labels)
```

In [10]:
```
print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))
```

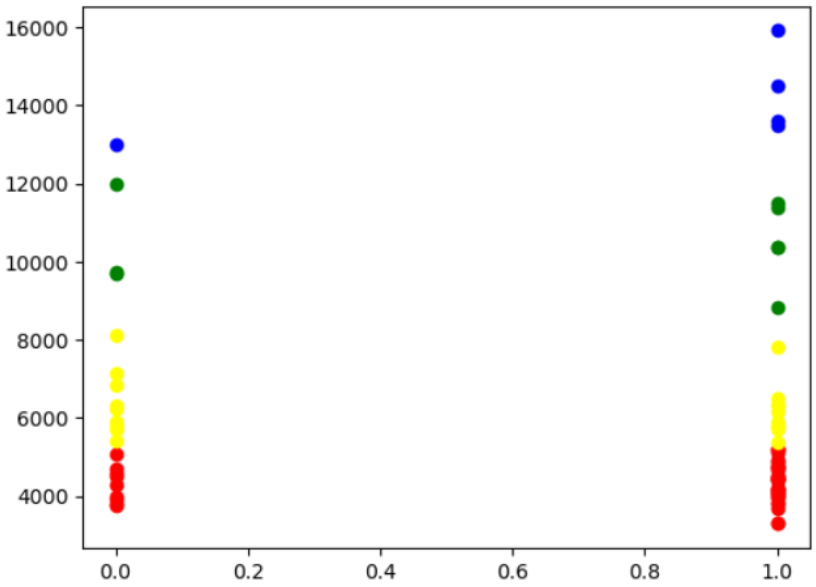
```
Result: 13 out of 74 samples were correctly labeled.
```

In [11]:
```
print(kmeans.inertia_)
print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))
```

```
30749322.7743309
Accuracy score: 0.18
```

In [33]:
```
plt.scatter(X[y_means==0,0],X[y_means==0,1],color='red')
plt.scatter(X[y_means==1,0],X[y_means==1,1],color='blue')
plt.scatter(X[y_means==2,0],X[y_means==2,1],color='yellow')
plt.scatter(X[y_means==3,0],X[y_means==3,1],color='green')
```

Out[33]: <matplotlib.collections.PathCollection at 0x23e3ffc0fd0>




## Cluster 1:

In [12]:
```
"""X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.20, random_state=42)"""
kmeans = KMeans(n_clusters=1)
kmeans.fit(df)
```

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

Out[12]: KMeans(n_clusters=1)

In [13]:
```
labels = kmeans.labels_
```

In [14]:
```
correct_labels = sum(y == labels)
```

In [15]:
```
print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))
```

```
Result: 22 out of 74 samples were correctly labeled.
```

In [16]:
```
print(kmeans.inertia_)
print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))
```

```
645268746.0425
Accuracy score: 0.30
```

## Cluster 2:

```
In [17]: """X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.20, random_state=42)"""
         kmeans = KMeans(n_clusters=2)
         kmeans.fit(df)

Out[17]: KMeans(n_clusters=2)

In [18]: labels = kmeans.labels_

In [19]: correct_labels = sum(y == labels)

In [20]: print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))

         Result: 27 out of 74 samples were correctly labeled.

In [21]: print(kmeans.inertia_)
         print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))

         129683516.17501615
         Accuracy score: 0.36
```

## Cluster 3:

```
In [22]: """X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.20, random_state=42)"""
         kmeans = KMeans(n_clusters=3)
         kmeans.fit(df)

Out[22]: KMeans(n_clusters=3)

In [23]: labels = kmeans.labels_

In [24]: correct_labels = sum(y == labels)

In [25]: print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))

         Result: 18 out of 74 samples were correctly labeled.

In [26]: print(kmeans.inertia_)
         print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))

         77317967.05184798
         Accuracy score: 0.24
```

## Cluster 4:

```
In [27]: """X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.20, random_state=42)"""
         kmeans = KMeans(n_clusters=4)
         kmeans.fit(df)

Out[27]: KMeans(n_clusters=4)

In [28]: labels = kmeans.labels_

In [29]: correct_labels = sum(y == labels)

In [30]: print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))

         Result: 42 out of 74 samples were correctly labeled.

In [31]: print(kmeans.inertia_)
         print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))

         41114884.26392993
         Accuracy score: 0.57
```

## Cluster 5:

In [32]:
```
"""X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.20, random_state=42)"""
kmeans = KMeans(n_clusters=5)
kmeans.fit(df)
```

Out[32]:
```
KMeans(n_clusters=5)
```

In [33]:
```
labels = kmeans.labels_
```

In [34]:
```
correct_labels = sum(y == labels)
```

In [35]:
```
print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))
```

```
Result: 12 out of 74 samples were correctly labeled.
```

In [36]:
```
print(kmeans.inertia_)
print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))
```

```
30638267.897469945
Accuracy score: 0.16
```