In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [77]:
```
data = pd.read_csv('Mall_Customers (1).csv')
```

In [78]:
```
data.head()
```

Out[78]:

| | CustomerID | Genre | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| 0 | 1 | Male | 19 | 15 | 39 |
| 1 | 2 | Male | 21 | 15 | 81 |
| 2 | 3 | Female | 20 | 16 | 6 |
| 3 | 4 | Female | 23 | 16 | 77 |
| 4 | 5 | Female | 31 | 17 | 40 |

In [79]:
```
x = data[['Annual Income (k$)','Spending Score (1-100)']]
```

In [80]:
```
from sklearn.cluster import KMeans
```

In [81]:
```
k_cluster = KMeans(n_clusters=5)
k_cluster.fit(x)
```

Out[81]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [82]:
```
y_lable = k_cluster.fit_predict(x)
y_lable
```

Out[82]:
```
array([1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 4,
       1, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 0, 2, 0, 2, 0, 2, 0, 2, 0, 4, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
       2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
       2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
       2, 0])
```

In [83]:
```
k_cluster.cluster_centers_
```

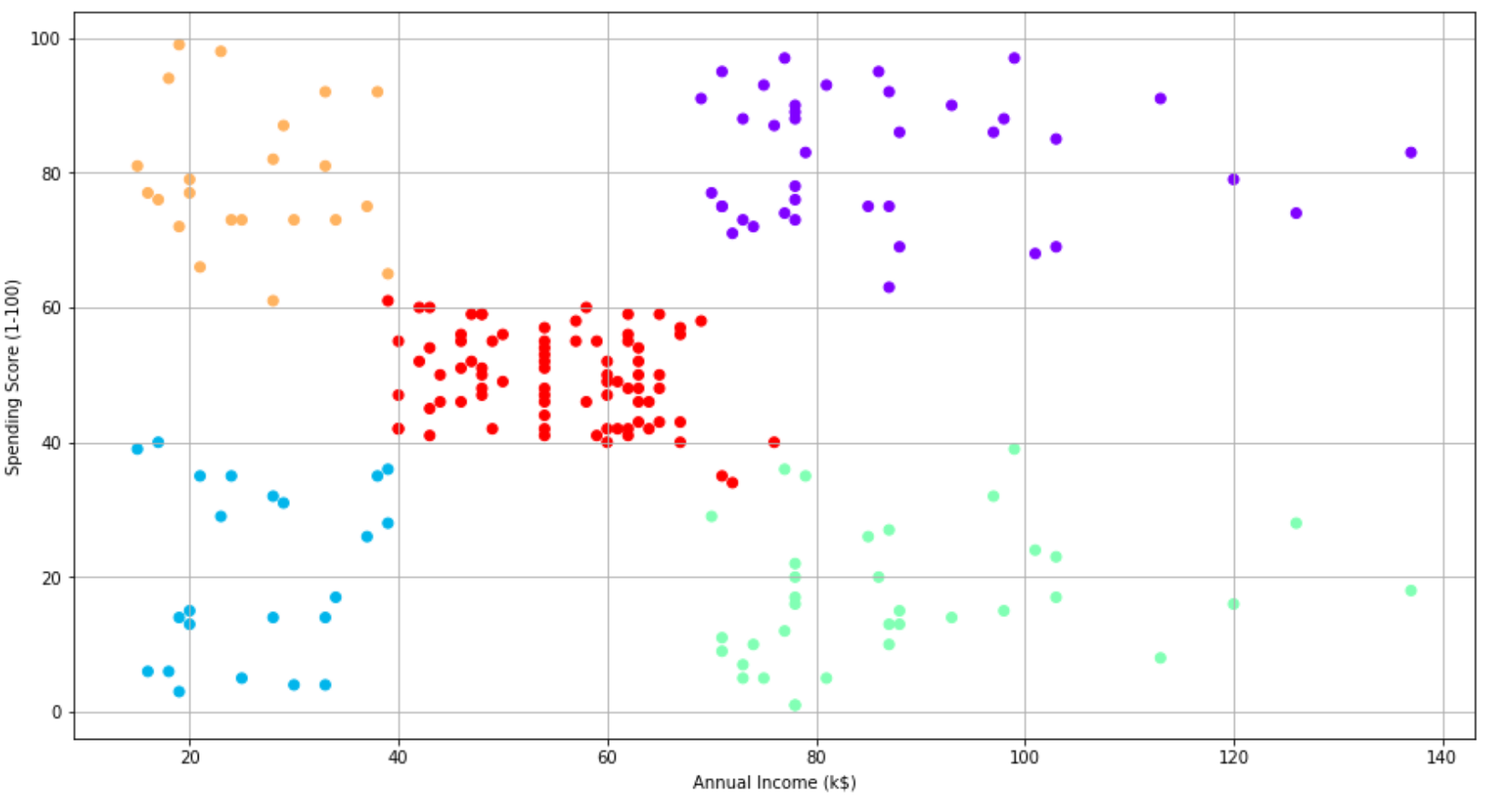
Out[83]:
```
array([[86.53846154, 82.12820513],
       [26.30434783, 20.91304348],
       [88.2       , 17.11428571],
       [25.72727273, 79.36363636],
       [55.2962963 , 49.51851852]])
```

In [84]:
```
from sklearn.metrics import silhouette_score
```

In [85]:
```
print('sihouette Score :-',silhouette_score(x,y_lable))
```

```
sihouette Score :- 0.553931997444648
```

In [102…
```
plt.figure(figsize=(15,8))
plt.scatter(data['Annual Income (k$)'],data['Spending Score (1-100)'],c=y_lable,cmap='rainbow')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.grid()
```




## Elbow Method

WCSS :- Within cluster sum square.

- This is used to detemine the optimal number of clusters.
- Calculate the within cluster sum of squred errors(WSS) FOR THE different values of k
- It plot the graph between the WSS vs the k.
- It will shows that the sum of square within the cluster are to diminished with increase in cluser.

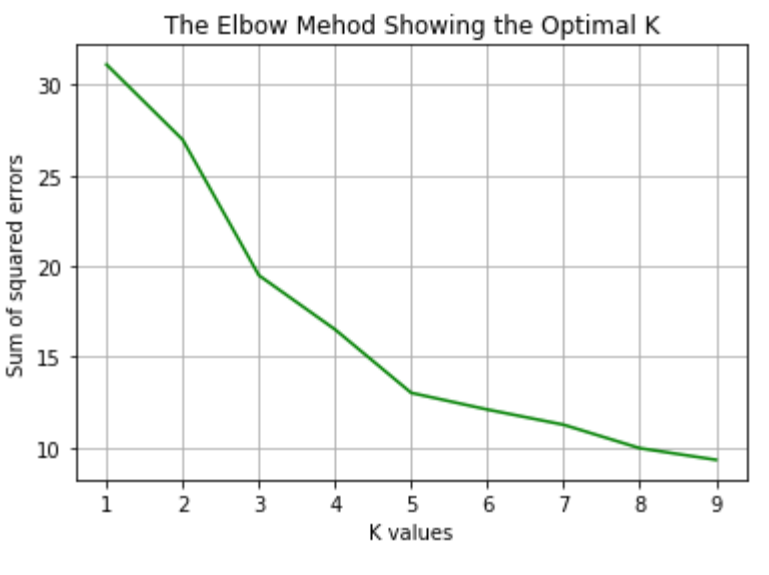
In [91]:
```
import sklearn.metrics
from scipy.spatial.distance import cdist
```

In [92]:
```
#k -means ditermine the k

distortions =[]
K = range(1,10)

for k in K:
    kmeansmodel = KMeans(n_clusters=k)
    kmeansmodel.fit(x)
    distortions.append(sum(np.min(cdist(x,kmeansmodel.cluster_centers_,'euclidean'),axis=1))/x.shape[0])
```

In [93]:
```
distortions
```

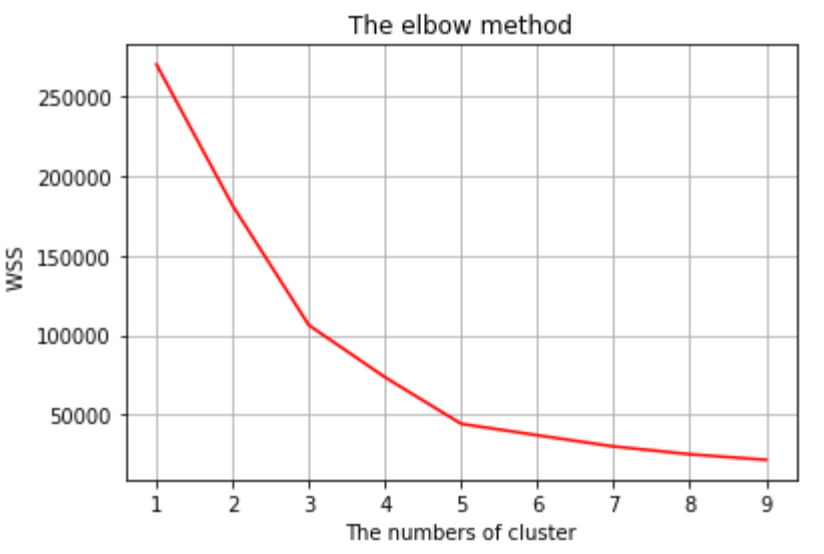
Out[93]:
```
[31.11956466178274,
 26.9728389711296,
 19.49852756350156,
 16.513850087397763,
 13.020126350791745,
 12.10048793253976,
 11.26320337095862,
 9.970333766315191,
 9.32923133542803]
```

In [94]:
```
plt.plot(K,distortions,color='g')
plt.title('The Elbow Mehod Showing the Optimal K')
plt.grid()
plt.xlabel('K values')
plt.ylabel('Sum of squared errors')
plt.show()
```




In [98]:
```
from sklearn.cluster import KMeans
```

In [104…
```
WSS =[]
for i in range(1,10):
    k_means = KMeans(n_clusters=i,init='k-means++',random_state=42)
    k_means.fit(x)
    WSS.append(k_means.inertia_)

plt.plot(range(1,10),WSS,color='r')
plt.title('The elbow method')
plt.xlabel('The numbers of cluster')
plt.grid()
plt.ylabel('WSS ')
plt.show()
```




In [ ]:

In [ ]: