In [1]:
```
#Import all the required libraries including KMeans
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

In [3]:
```
#reading the dataset
df = pd.read_csv('Iris.csv')
df.head(10)
```

Out[3]:

	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
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [7]:
```
x = df.iloc[:, [0,1,2,3]].values
x
```

```
[ 19. ,   5.7,   3.8,   1.7],
[ 20. ,   5.1,   3.8,   1.5],
[ 21. ,   5.4,   3.4,   1.7],
[ 22. ,   5.1,   3.7,   1.5],
[ 23. ,   4.6,   3.6,   1. ],
[ 24. ,   5.1,   3.3,   1.7],
[ 25. ,   4.8,   3.4,   1.9],
[ 26. ,   5. ,   3. ,   1.6],
[ 27. ,   5. ,   3.4,   1.6],
[ 28. ,   5.2,   3.5,   1.5],
[ 29. ,   5.2,   3.4,   1.4],
[ 30. ,   4.7,   3.2,   1.6],
[ 31. ,   4.8,   3.1,   1.6],
[ 32. ,   5.4,   3.4,   1.5],
[ 33. ,   5.2,   4.1,   1.5],
[ 34. ,   5.5,   4.2,   1.4],
[ 35. ,   4.9,   3.1,   1.5],
[ 36. ,   5. ,   3.2,   1.2],
[ 37. ,   5.5,   3.5,   1.3],
[ 38. ,   4.9,   3.1,   1.5],
```

In [ ]:
```

```

In [17]:
```
# Now, we are choose number of cluster random=5 and then fit the val
kmeans = KMeans(n_clusters=5)
y_kmeans = kmeans.fit_predict(x)
print(y_kmeans)

kmeans.cluster_centers_
```

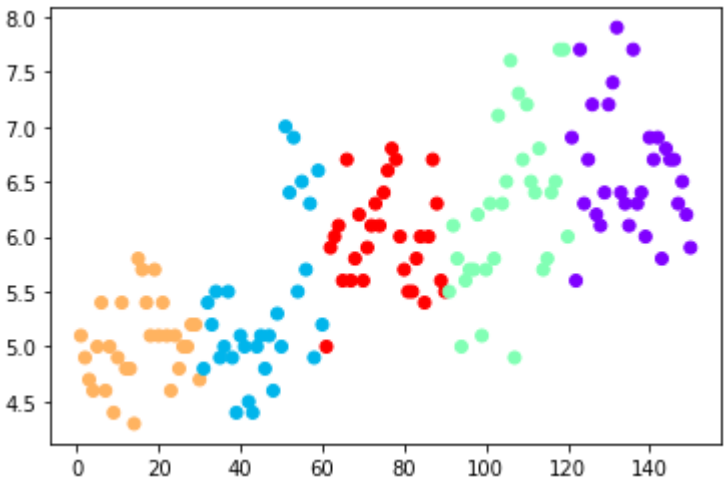
```
[3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1
1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4
4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2
 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0
 0 0]
```

Out[17]:
```
array([[135.5       ,   6.60666667,   3.01      ,   5.48333333],
       [ 45.5       ,   5.35      ,   3.20333333,   2.42333333],
       [105.5       ,   6.25333333,   2.85666667,   5.11333333],
       [ 15.5       ,   5.02666667,   3.45      ,   1.47333333],
       [ 75.5       ,   5.98      ,   2.75      ,   4.3       ]])
```

In [ ]:
```

```

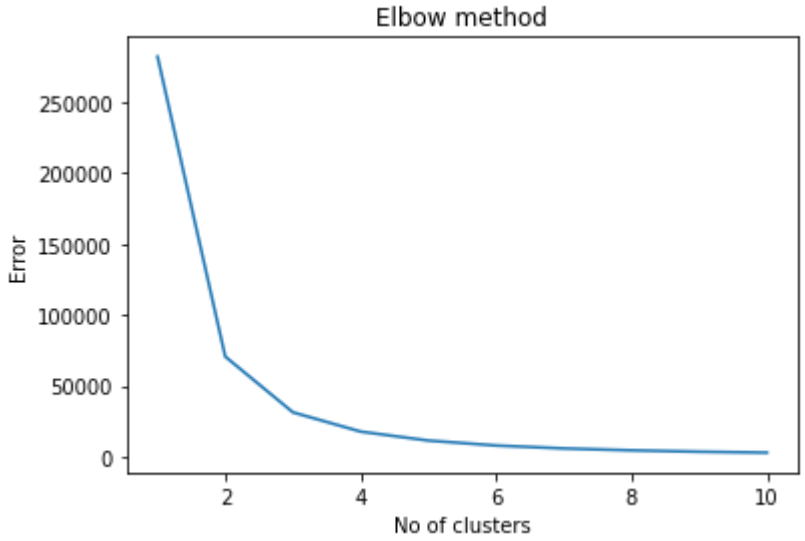
In [19]:
```
plt.scatter(x[:,0], x[:,1],c=y_kmeans,cmap='rainbow')

```

Out[19]:
```
<matplotlib.collections.PathCollection at 0x7f205dffafa0>
```



In [20]:

```
Error =[]
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i).fit(x)
    kmeans.fit(x)
    Error.append(kmeans.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(1, 11), Error)
plt.title('Elbow method')
plt.xlabel('No of clusters')
plt.ylabel('Error')
plt.show()
```



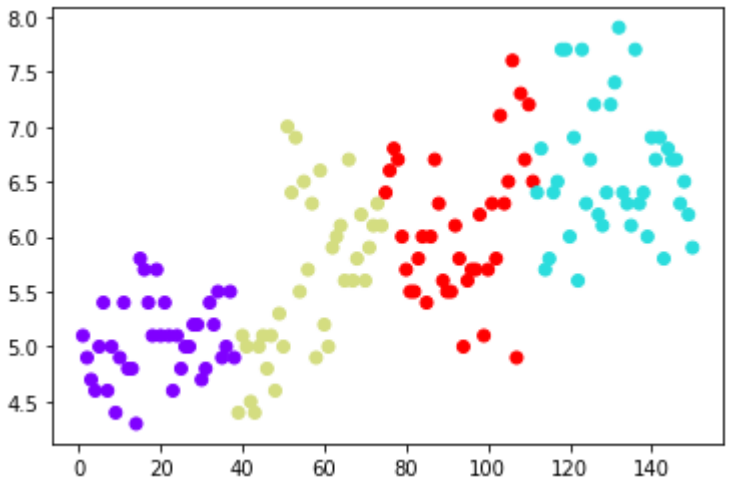
In [23]:
```
# Now, we are choose number of cluster random=4 and then fit the val
kmeans = KMeans(n_clusters=4)
y_kmeans = kmeans.fit_predict(x)
print(y_kmeans)

kmeans.cluster_centers_
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0
 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1
 1 1]
```

Out[23]:
```
array([[ 19.5       ,   5.05263158,   3.45263158,   1.46578947],
       [131.        ,   6.59487179,   2.97692308,   5.50769231],
       [ 56.5       ,   5.61666667,   2.95      ,   3.36111111],
       [ 93.        ,   6.08378378,   2.82702703,   4.65675676]])
```

In [24]:
```
plt.scatter(x[:,0], x[:,1],c=y_kmeans,cmap='rainbow')

```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x7f205e1421f0>
```



In [ ]:
```

```