In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
df=pd.read_csv("Mall_Customers.csv")
```

In [4]:
```
df.head(10)
```

	CustomerID	Gender	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

In [5]:
```
df.describe()
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

In [6]:
```
df.shape
```

```
(200, 5)
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [8]:
```
df.isnull().sum()
```

```
CustomerID              0
Gender                  0
Age                     0
Annual Income (k$)      0
Spending Score (1-100)  0
dtype: int64
```

In [9]:
```
df.columns
```

```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

In [10]:
```
x=df[['Annual Income (k$)','Spending Score (1-100)']]
```

In [11]:

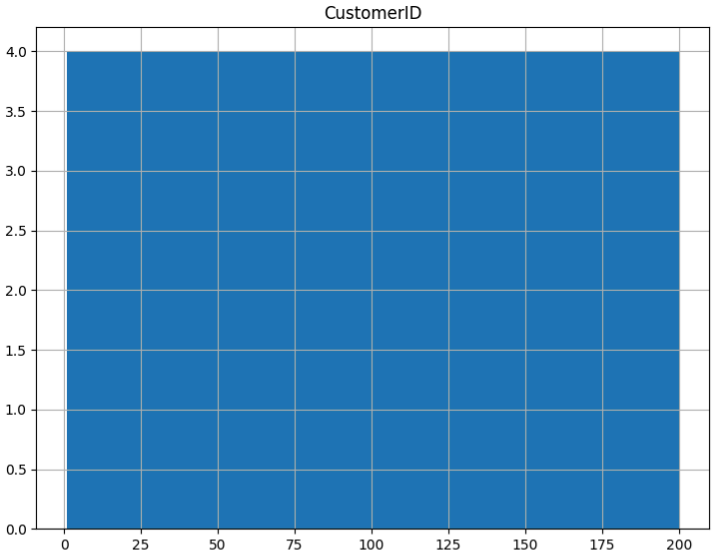
```
1 x
```

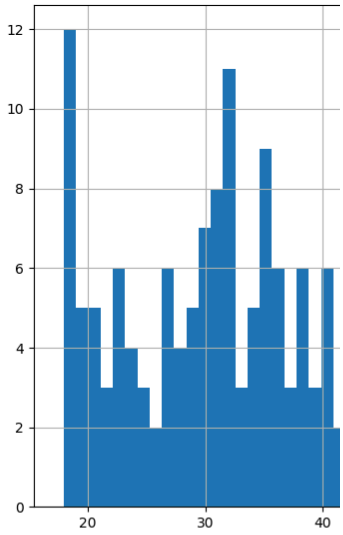
	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

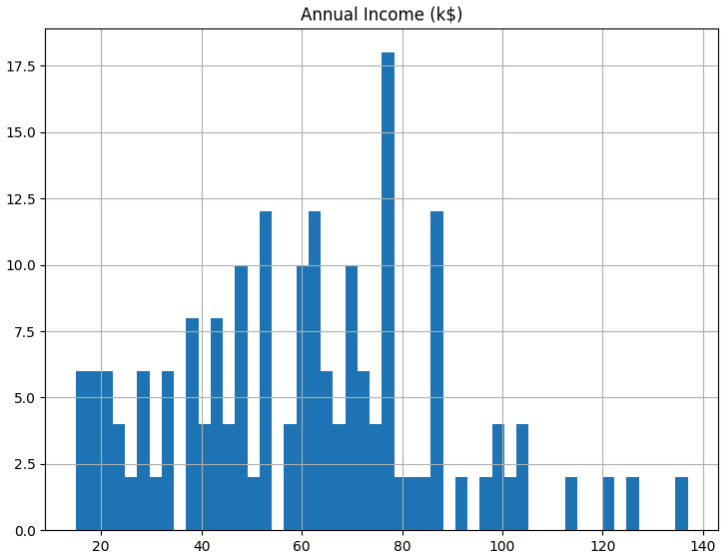
200 rows × 2 columns

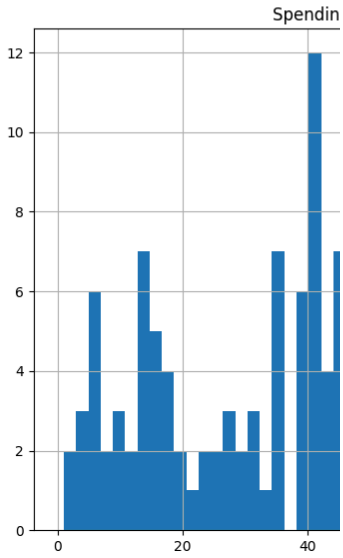
In [12]:

```
1 import matplotlib.pyplot as plt
2 df.hist(bins=50,figsize=(20,15))
3 plt.show()
```



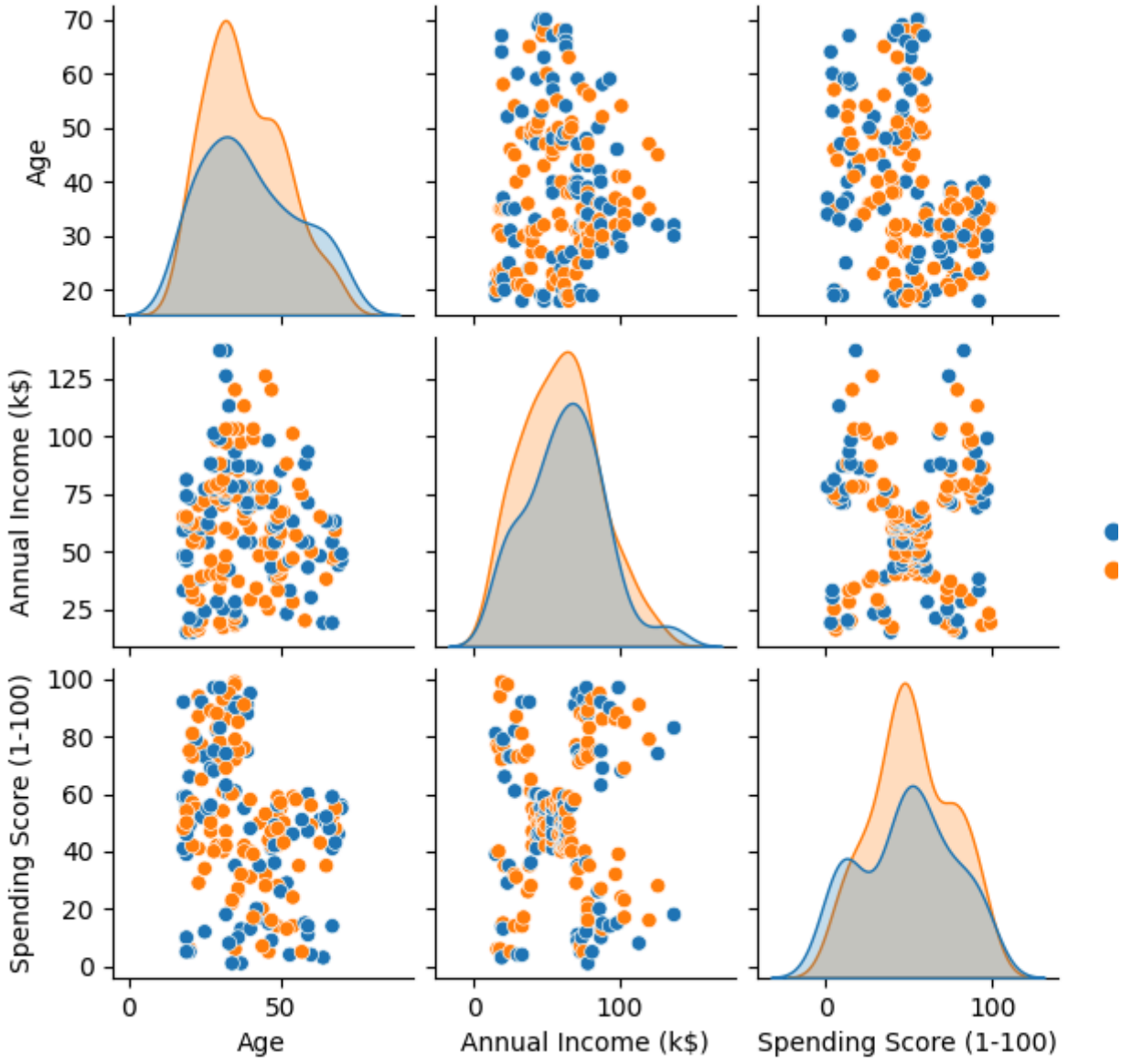






In [13]:

```
import seaborn as sns
sns.pairplot(df.drop(['CustomerID'],axis=1),hue='Gender',height=2)
plt.show()
```



In [14]:

```
from sklearn.cluster import KMeans
k_means=KMeans()
k_means.fit(x)
```

KMeans

KMeans()

In [15]:

```
k_means=KMeans(n_clusters=5)
k_means.fit_predict(x)
```

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

In [16]:

```
wcss=[]
for i in range(1,11):
    k_means=KMeans(n_clusters=i)
    k_means.fit(x)
    wcss.append(k_means.inertia_)
```

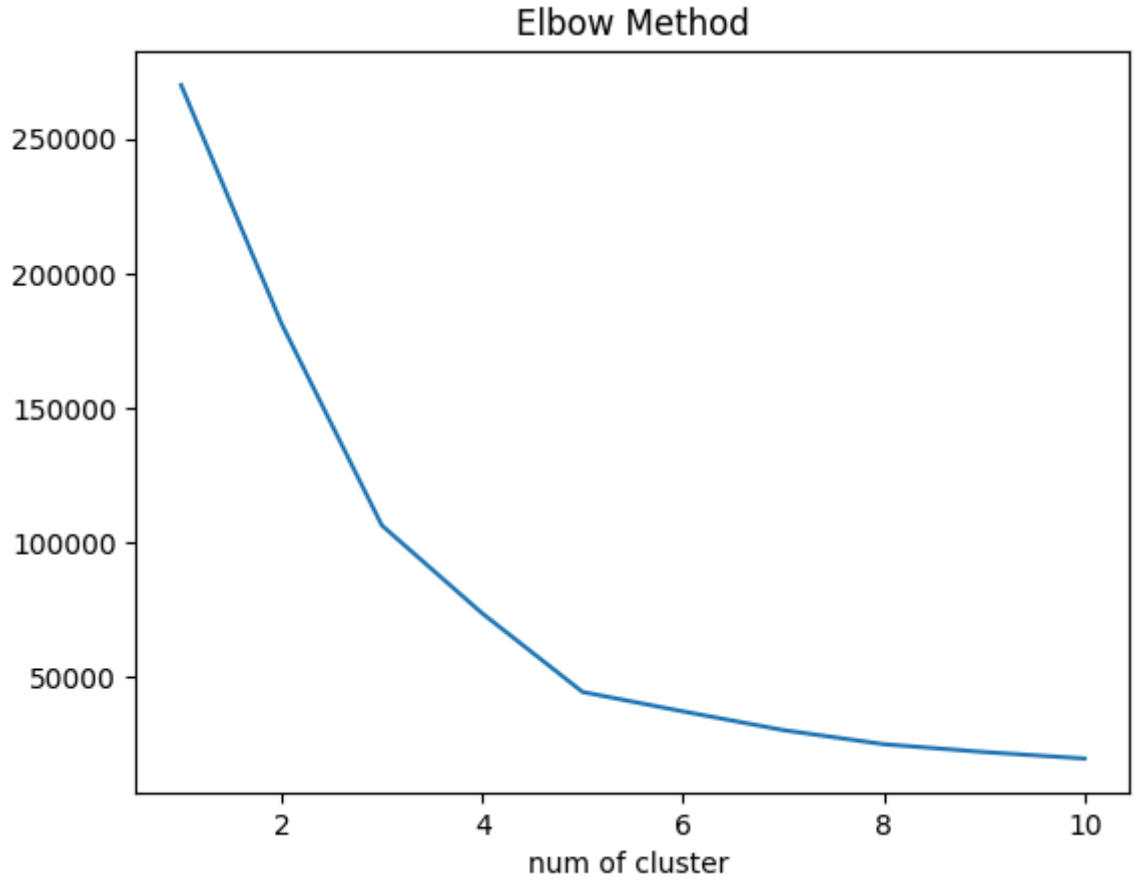
In [17]:

```
wcss
```

```
[269981.28,
 181363.595959596,
 106348.37306211118,
 73679.78903948836,
 44448.45544793371,
 37233.81451071001,
 30273.394312070042,
 25043.890043290045,
 22131.92051101073,
 19721.54752731275]
```

In [18]:

```
plt.plot(range(1,11),wcss)
plt.title("Elbow Method")
plt.xlabel("num of cluster ")
plt.show()
```



In [19]:

```
k_means=KMeans(n_clusters=5,random_state=42)
y_means=k_means.fit_predict(x)
```

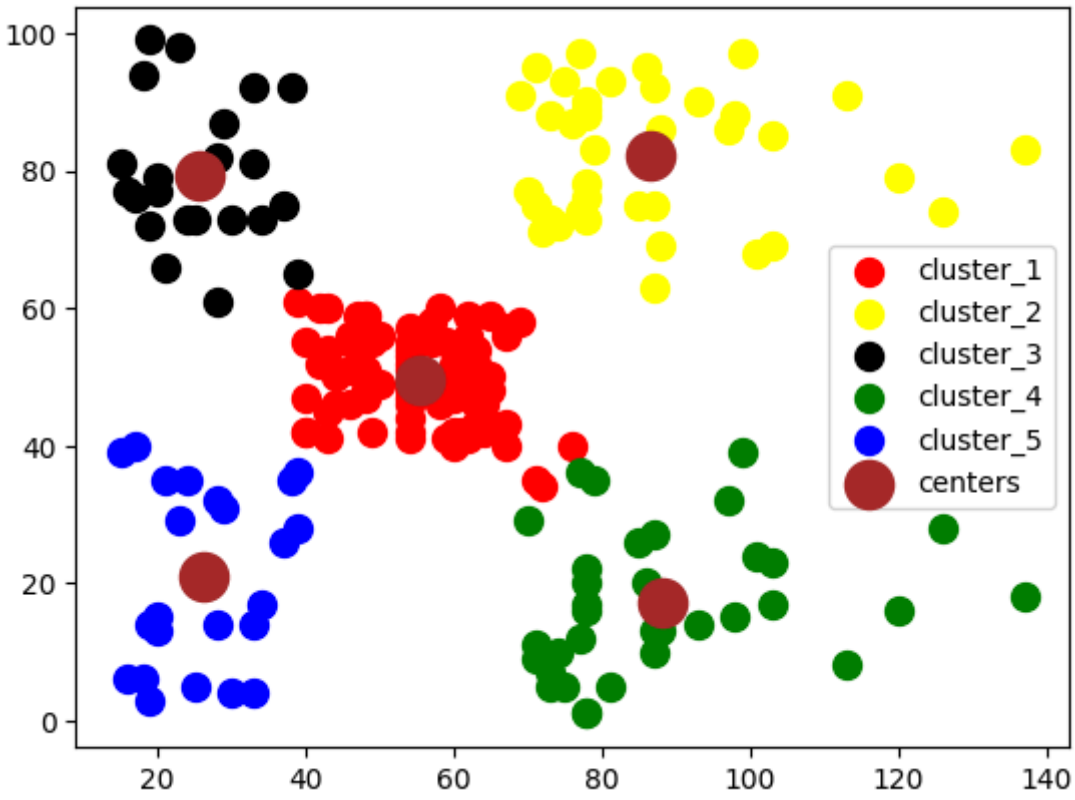
In [20]:

```
y_means
```

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

In [21]:

```
plt.scatter(x.iloc[y_means==0,0],x.iloc[y_means==0,1],s=100,c="red",label
plt.scatter(x.iloc[y_means==1,0],x.iloc[y_means==1,1],s=100,c="yellow",la
plt.scatter(x.iloc[y_means==2,0],x.iloc[y_means==2,1],s=100,c="black",lab
plt.scatter(x.iloc[y_means==3,0],x.iloc[y_means==3,1],s=100,c="green",lab
plt.scatter(x.iloc[y_means==4,0],x.iloc[y_means==4,1],s=100,c="blue",labe
plt.scatter(k_means.cluster_centers_[:,0],k_means.cluster_centers_[:,1],s
plt.legend()
plt.show()
```

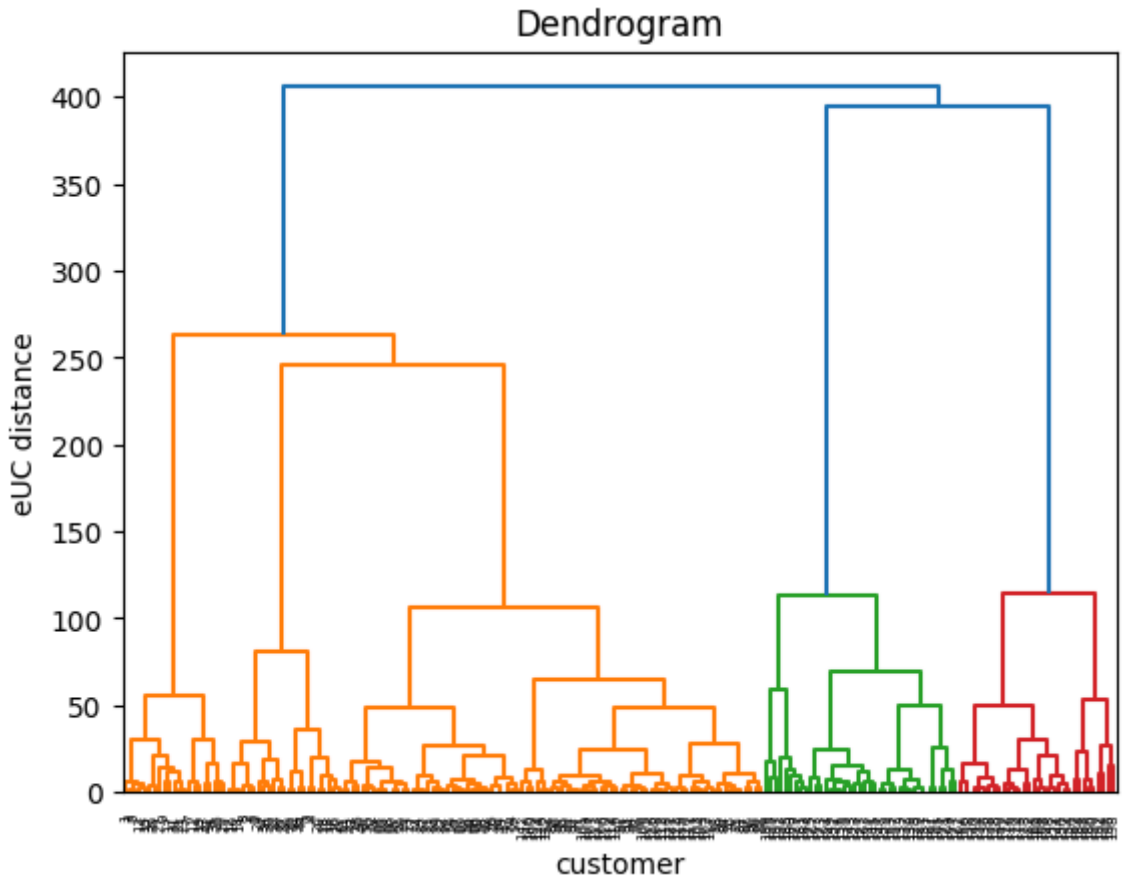


In [22]:

```
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
```

In [23]:

```
dendrogram=sch.dendrogram(sch.linkage(x,method="ward"))
plt.title("Dendrogram")
plt.xlabel("customer")
plt.ylabel("eUC distance")
plt.show()
```

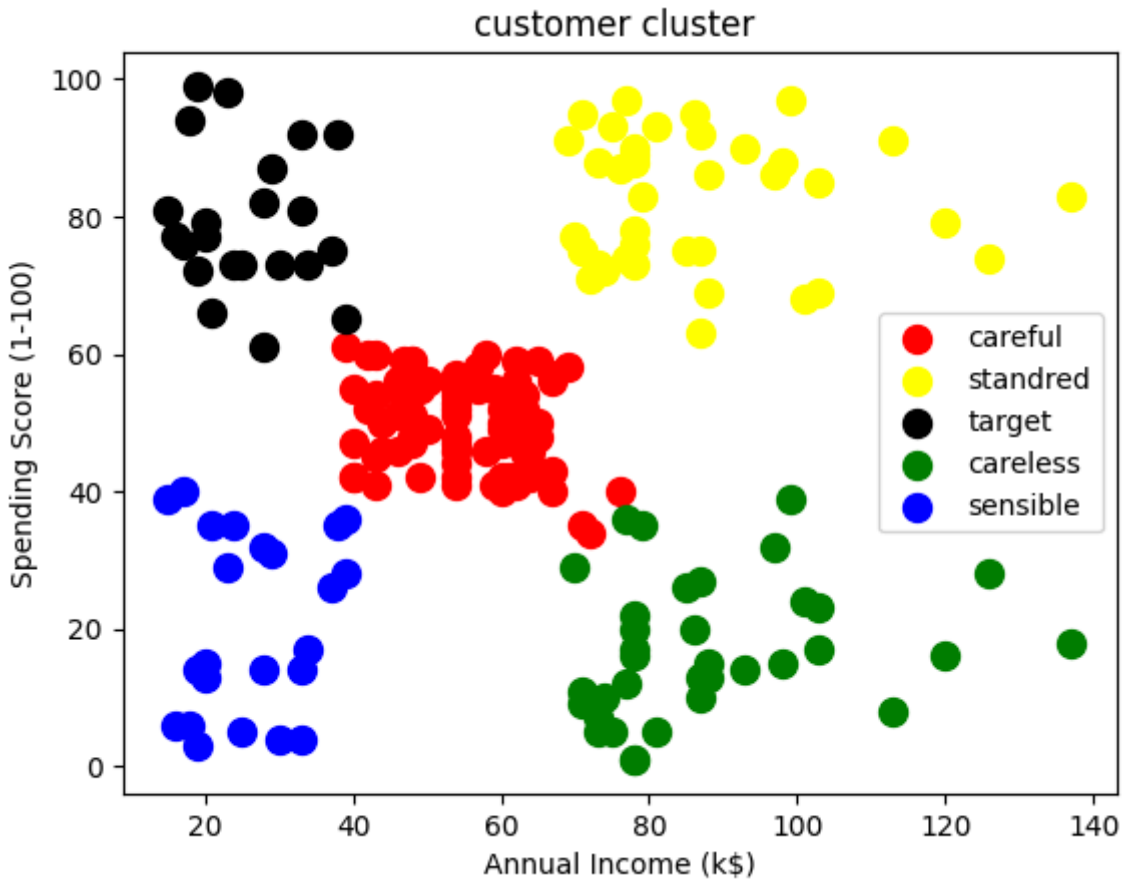


In [24]:

```
hc = AgglomerativeClustering(n_clusters=5, affinity="euclidean", linkage
y_hc = hc.fit_predict(x)
```

In [25]:

```
plt.scatter(x.iloc[y_means==0,0],x.iloc[y_means==0,1],s=100,c="red",label
plt.scatter(x.iloc[y_means==1,0],x.iloc[y_means==1,1],s=100,c="yellow",la
plt.scatter(x.iloc[y_means==2,0],x.iloc[y_means==2,1],s=100,c="black",lab
plt.scatter(x.iloc[y_means==3,0],x.iloc[y_means==3,1],s=100,c="green",lab
plt.scatter(x.iloc[y_means==4,0],x.iloc[y_means==4,1],s=100,c="blue",labe
plt.title("customer cluster ")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.legend()
plt.show()
```



In [ ]:

```
1
```