In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
df=pd.read_csv("Mall_Customers.csv")
df.head()
```

Out[3]:

	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 [4]:

```
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   Genre                   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 [6]: x=df.iloc[:,[3,4]].values
        x
```

```
Out[6]: array([[ 15,  39],
               [ 15,  81],
               [ 16,   6],
               [ 16,  77],
               [ 17,  40],
               [ 17,  76],
               [ 18,   6],
               [ 18,  94],
               [ 19,   3],
               [ 19,  72],
               [ 19,  14],
               [ 19,  99],
               [ 20,  15],
               [ 20,  77],
               [ 20,  13],
               [ 20,  79],
               [ 21,  35],
               [ 21,  66],
               [ 23,  29],
               [ 23,  98],
               [ 24,  35],
               [ 24,  73],
               [ 25,   5],
               [ 25,  73],
               [ 28,  14],
               [ 28,  82],
               [ 28,  32],
               [ 28,  61],
               [ 29,  31],
               [ 29,  87],
               [ 30,   4],
               [ 30,  73],
               [ 33,   4],
               [ 33,  92],
               [ 33,  14],
               [ 33,  81],
               [ 34,  17],
               [ 34,  73],
               [ 37,  26],
               [ 37,  75],
               [ 38,  35],
               [ 38,  92],
               [ 39,  36],
               [ 39,  61],
               [ 39,  28],
               [ 39,  65],
               [ 40,  55],
               [ 40,  47],
               [ 40,  42],
               [ 40,  42],
               [ 42,  52],
               [ 42,  60],
               [ 43,  54],
               [ 43,  60],
               [ 43,  45],
               [ 43,  41],
               [ 44,  50],
               [ 44,  46],
               [ 46,  51],
               [ 46,  46],
               [ 46,  56],
               [ 46,  55],
               [ 47,  52],
               [ 47,  59],
               [ 48,  51],
               [ 48,  59],
               [ 48,  50],
               [ 48,  48],
               [ 48,  59],
               [ 48,  47],
               [ 49,  55],
               [ 49,  42],
               [ 50,  49],
               [ 50,  56],
               [ 54,  47],
               [ 54,  54],
               [ 54,  53],
               [ 54,  48],
               [ 54,  52],
               [ 54,  42],
               [ 54,  51],
               [ 54,  55],
               [ 54,  41],
               [ 54,  44],
               [ 54,  57],
               [ 54,  46],
               [ 57,  58],
               [ 57,  55],
               [ 58,  60],
               [ 58,  46],
               [ 59,  55],
               [ 59,  41],
               [ 60,  49],
               [ 60,  40],
               [ 60,  42],
               [ 60,  52],
               [ 60,  47],
               [ 60,  50],
               [ 61,  42],
               [ 61,  49],
               [ 62,  41],
               [ 62,  48],
               [ 62,  59],
               [ 62,  55],
               [ 62,  56],
               [ 62,  42],
               [ 63,  50],
               [ 63,  46],
               [ 63,  43],
               [ 63,  48],
               [ 63,  52],
               [ 63,  54],
               [ 64,  42],
               [ 64,  46],
               [ 65,  48],
               [ 65,  50],
               [ 65,  43],
               [ 65,  59],
               [ 67,  43],
               [ 67,  57],
               [ 67,  56],
               [ 67,  40],
               [ 69,  58],
               [ 69,  91],
               [ 70,  29],
               [ 70,  77],
               [ 71,  35],
               [ 71,  95],
               [ 71,  11],
               [ 71,  75],
               [ 71,   9],
               [ 71,  75],
               [ 72,  34],
               [ 72,  71],
               [ 73,   5],
               [ 73,  88],
               [ 73,   7],
               [ 73,  73],
               [ 74,  10],
               [ 74,  72],
               [ 75,   5],
               [ 75,  93],
               [ 76,  40],
               [ 76,  87],
               [ 77,  12],
               [ 77,  97],
               [ 77,  36],
               [ 77,  74],
               [ 78,  22],
               [ 78,  90],
               [ 78,  17],
               [ 78,  88],
               [ 78,  20],
               [ 78,  76],
               [ 78,  16],
               [ 78,  89],
               [ 78,   1],
               [ 78,  78],
               [ 78,   1],
               [ 78,  73],
               [ 79,  35],
               [ 79,  83],
               [ 81,   5],
               [ 81,  93],
               [ 85,  26],
               [ 85,  75],
               [ 86,  20],
               [ 86,  95],
               [ 87,  27],
               [ 87,  63],
               [ 87,  13],
               [ 87,  75],
               [ 87,  10],
               [ 87,  92],
               [ 88,  13],
               [ 88,  86],
               [ 88,  15],
               [ 88,  69],
               [ 93,  14],
               [ 93,  90],
               [ 97,  32],
               [ 97,  86],
               [ 98,  15],
               [ 98,  88],
               [ 99,  39],
               [ 99,  97],
               [101,  24],
               [101,  68],
               [103,  17],
               [103,  85],
               [103,  23],
               [103,  69],
               [113,   8],
               [113,  91],
               [120,  16],
               [120,  79],
               [126,  28],
               [126,  74],
               [137,  18],
               [137,  83]], dtype=int64)
```

In [7]:

```
#to build Kmeans clustering algo.
from sklearn.cluster import KMeans
```

In [8]:

```
# HEre we don't have domain knowledge hence we can not decide the value of K first.
#so we are using elbow method to decide the value of K.
```

Elbow method

In [11]:

```
wcss=[]
for k in range(1,30):
    km=KMeans(n_clusters=k,init="k-means++",n_init=10,max_iter=300,random_state=1)
    km.fit(x)
    wcss.append(km.inertia_)
```

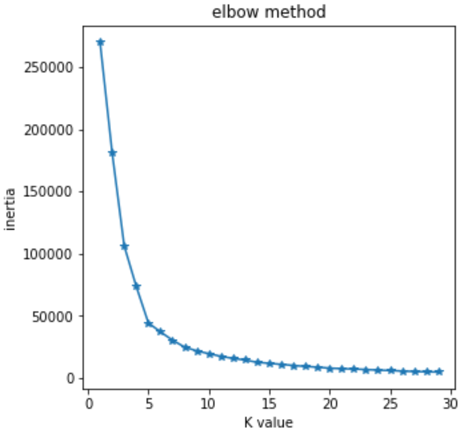
In [12]:

```
wcss
```

Out[12]:

```
[269981.28,
 181363.59595959596,
 106348.37306211119,
 73679.78903948834,
 44448.45544793371,
 37233.81451071001,
 30566.45113025186,
 25005.55037243283,
 21996.523372372307,
 19746.911957660894,
 17602.19046838677,
 15926.627039985106,
 14631.802353268315,
 12793.951692914929,
 12002.023047743332,
 11151.50775058275,
 10264.837447764541,
 9490.19002831011,
 8880.551059466847,
 8121.5007317801665,
 7667.180982236156,
 7385.859950013755,
 7018.027058579537,
 6517.28038699806,
 6232.733574758575,
 5756.705148119854,
 5413.182221401787,
 5208.137817737817,
 4989.701612276613]
```

In [15]:
```
plt.figure(figsize=(5,5))
plt.title("elbow method")
plt.plot(range(1,30),wcss,marker="*")
plt.xlabel("K value")
plt.ylabel("inertia")
plt.show()
```



In [16]:
```
#final K value from elbow method is 5 , so we will build the final with k value as 5
```

In [20]:
```
km1=KMeans(n_clusters=5,init="k-means++",n_init=10,max_iter=300,random_state=1)
labels=km1.fit_predict(x)
```

In [24]:
```
#printing value of inertia when k=5
km1.inertia_
```

Out[24]:
```
44448.45544793371
```

In [25]:
```
#printing values of centroid
km1.cluster_centers_
```

Out[25]:
```
array([[25.72727273, 79.36363636],
       [88.2       , 17.11428571],
       [55.2962963 , 49.51851852],
       [86.53846154, 82.12820513],
       [26.30434783, 20.91304348]])
```

In [21]:
```
labels
```

Out[21]:
```
array([4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 2,
       4, 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, 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, 2, 2, 2, 3, 1, 3, 2, 3, 1, 3, 1, 3,
       2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 2, 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, 3])
```

In [22]:
```
df["c_label"]=labels
```

In [23]:

```
df
```

Out[23]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)	c_label
0	1	Male	19	15	39	4
1	2	Male	21	15	81	0
2	3	Female	20	16	6	4
3	4	Female	23	16	77	0
4	5	Female	31	17	40	4
...	...	...	...	...	...	...
195	196	Female	35	120	79	3
196	197	Female	45	126	28	1
197	198	Male	32	126	74	3
198	199	Male	32	137	18	1
199	200	Male	30	137	83	3

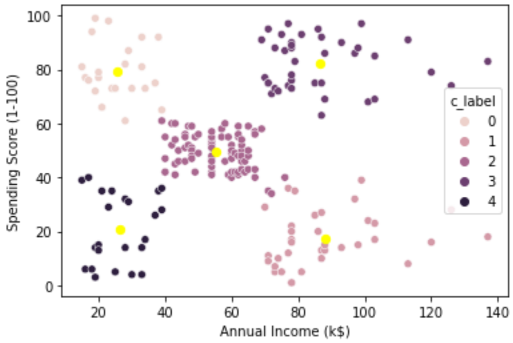
200 rows × 6 columns

In [31]:

```
centroid_df=pd.DataFrame(km1.cluster_centers_,columns=["X","Y"])
```

In [35]:

```
sns.scatterplot(data=df,x=df["Annual Income (k$)"],y="Spending Score (1-100)",hue=df["c_label"])
plt.scatter(centroid_df["X"],centroid_df["Y"],s=40,color='yellow')
plt.show()
```



In [ ]:

```

```