In [73]:

```
import pandas as pd
import numpy as np
```

In [74]:

```
df=pd.read_csv("C://Users/ABHISEK GARAI/Desktop/New folder (2)/customer-segmentation-dataset/Mall_Customers.csv")
```

In [75]:

```
df.head()
```

Out[75]:

	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

In [76]:

```
df.info()
```

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

In [77]:

```
df.describe()
```

Out[77]:

	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 [78]:

```
df.columns
```

Out[78]:

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

In [79]:

```
df['Gender'].value_counts()
```

Out[79]:

```
Female    112
Male       88
Name: Gender, dtype: int64
```

In [80]:

```
from sklearn.preprocessing import LabelEncoder
```

In [81]:

```
le=LabelEncoder()
```

In [82]:

```
le.fit(df['Gender'])
```

Out[82]:

```
LabelEncoder()
```

In [83]:

```
df['Gender']=le.transform(df['Gender'])
```

In [84]:

```
df.head()
```

Out[84]:

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

In [85]:

```
from sklearn.preprocessing import StandardScaler
```

In [86]:

```
st=StandardScaler()
df.columns
```

Out[86]:

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

```
      dtype='object')
```

In [87]:

```
X=df[['Age','Annual Income (k$)','Spending Score (1-100)']]
```

In [88]:

```
X=st.fit_transform(X)
```

In [89]:

```
X[0:5]
```

Out[89]:

```
array([[-1.42456879, -1.73899919, -0.43480148],
       [-1.28103541, -1.73899919,  1.19570407],
       [-1.3528021 , -1.70082976, -1.71591298],
       [-1.13750203, -1.70082976,  1.04041783],
       [-0.56336851, -1.66266033, -0.39597992]])
```

In [90]:

```
X.shape
```

Out[90]:

```
(200, 3)
```

In [91]:

```
X.shape[0]
```

Out[91]:

```
200
```

In [92]:

```
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
inner,dist=[],[]
```

In [93]:

```
c=range(1,11)
```

In [94]:

```
for i in c:
    k=KMeans(n_clusters=i)
    k.fit(X)
    inner.append(k.inertia_)
    dist.append(sum(np.min(cdist(X,k.cluster_centers_,'euclidean'),axis=1))/X.shape[0])
```

In [95]:

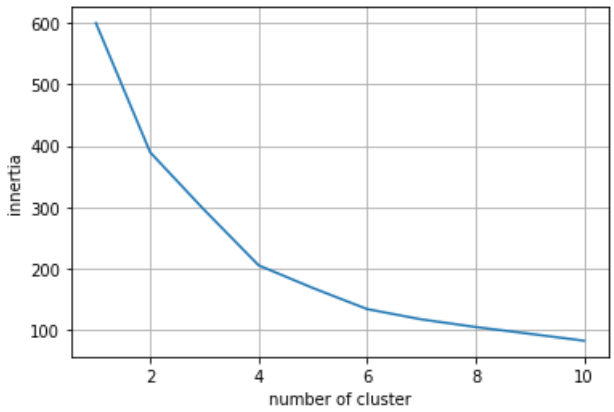
```
import matplotlib.pyplot as plt
len(inner)
```

Out[95]:

```
10
```

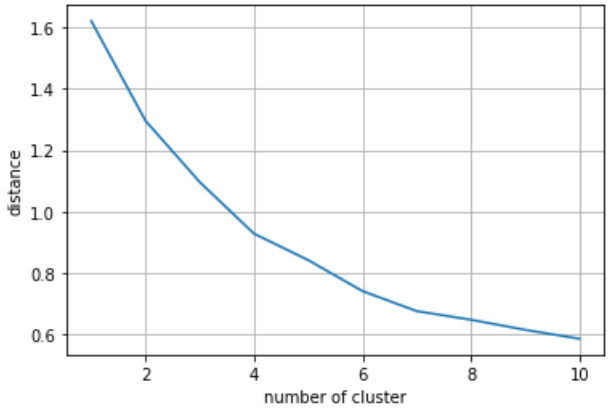
In [96]:

```
plt.plot(c,inner)
plt.xlabel('number of cluster')
plt.ylabel('innertia')
plt.grid()
plt.show()
```



In [97]:

```
plt.plot(c,dist)
plt.xlabel('number of cluster')
plt.ylabel('distance')
plt.grid()
plt.show()
```



In [98]:

```
K=KMeans(n_clusters=5)
K.fit(X)
```

Out[98]:

```
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 [99]:

```
labels=K.predict(X)
```

In [100]:

```
labels[0:5]
```

Out[100]:

```
array([1, 1, 3, 1, 1])
```

In [101]:

```
labels
```

Out[101]:

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

In [102]:

```
df['label']=labels
```

In [103]:

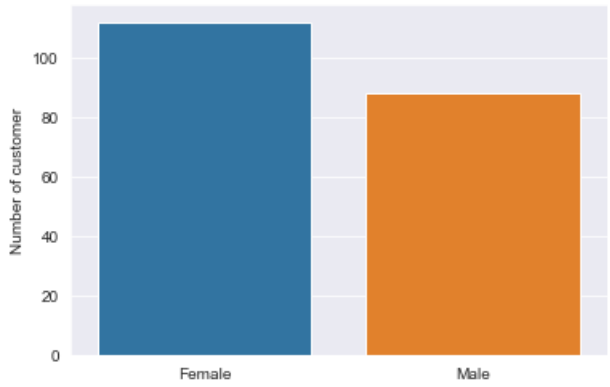
```
df['label'].value_counts()
```

Out[103]:

```
1    54
2    47
4    40
0    39
3    20
Name: label, dtype: int64
```

In [104]:

```
import seaborn as sns
sns.set_style('darkgrid')
sns.barplot(['Female','Male'],df['Gender'].value_counts().values)
plt.ylabel('Number of customer')
plt.show()
```



In [105]:

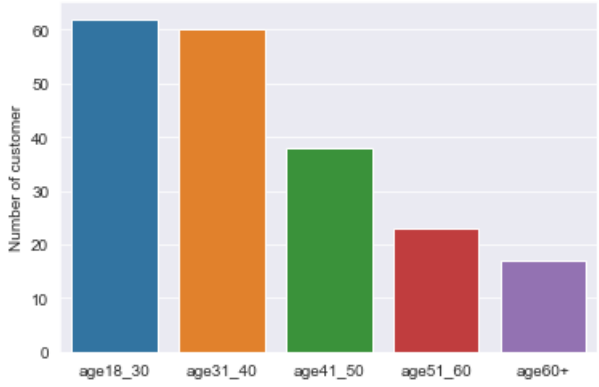
```
age18_30=df.Age[(df.Age>=18) & (df.Age<=30)].value_counts().sum()
age31_40=df.Age[(df.Age>=31) & (df.Age<=40)].value_counts().sum()
age41_50=df.Age[(df.Age>=41) & (df.Age<=50)].value_counts().sum()
age51_60=df.Age[(df.Age>=51) & (df.Age<=60)].value_counts().sum()
age60=df.Age[(df.Age>=61)].value_counts().sum()
```

In [106]:

```
xaxis=['age18_30','age31_40','age41_50','age51_60','age60+']
yaxis=[age18_30,age31_40,age41_50,age51_60,age60]
```

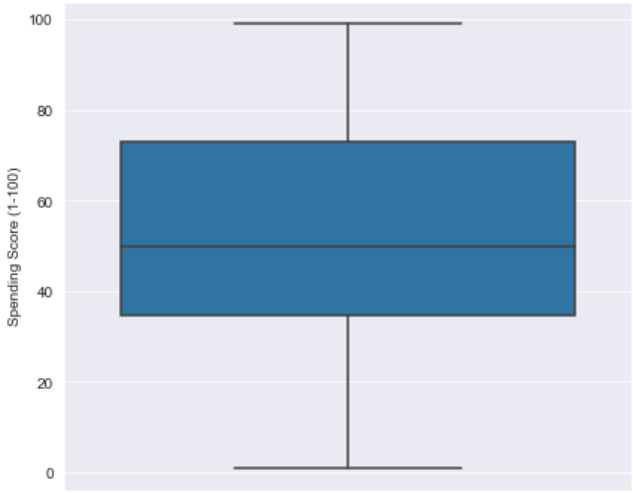
In [107]:

```
sns.barplot(x=xaxis,y=yaxis)
plt.ylabel('Number of customer')
plt.show()
```



In [108]:

```
plt.figure(figsize=(15,6))
plt.subplot(1,2,1)
sns.boxplot(df['Annual Income (k$)'],orient='v',color='red')
plt.subplot(1,2,2)
sns.boxplot(df['Spending Score (1-100)'],orient='v')
plt.show()
```



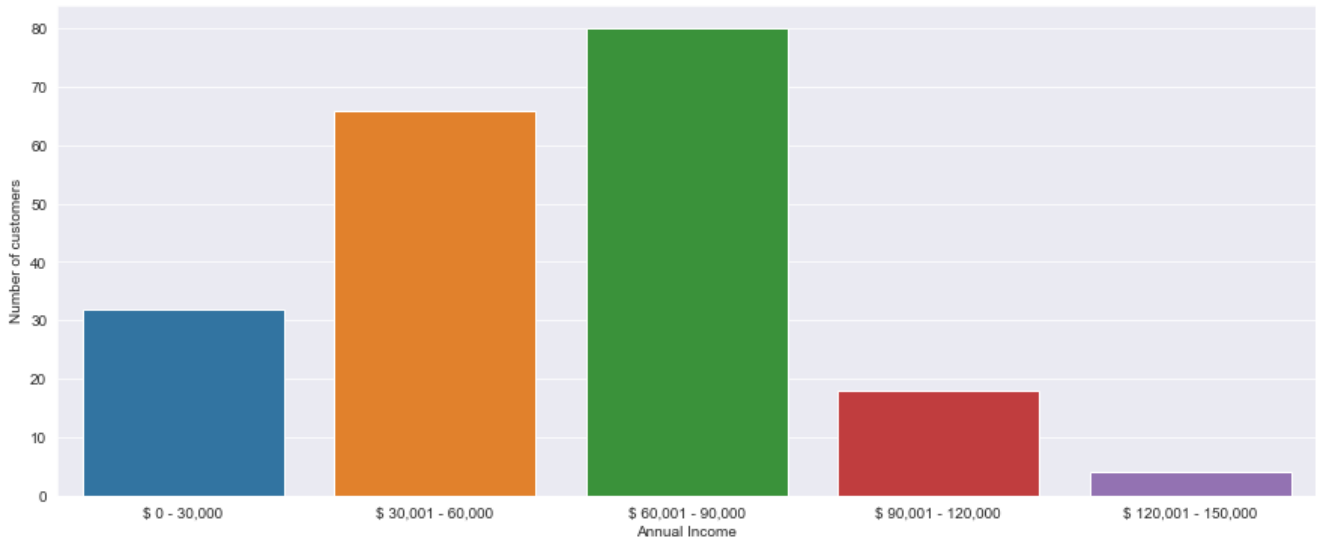
In [109]:

```
ai1_30=df['Annual Income (k$)'][(df['Annual Income (k$)']>=1) & (df['Annual Income
(k$)']<=30)].value_counts().sum()
ai31_60=df['Annual Income (k$)'][(df['Annual Income (k$)']>=31) & (df['Annual Income (k$)']<=60)].
value_counts().sum()
ai61_90=df['Annual Income (k$)'][(df['Annual Income (k$)']>=61) & (df['Annual Income (k$)']<=90)].
value_counts().sum()
ai91_120=df['Annual Income (k$)'][(df['Annual Income (k$)']>=91) & (df['Annual Income (k$)']<=120)
].value_counts().sum()
ai121_150=df['Annual Income (k$)'][(df['Annual Income (k$)']>=121) & (df['Annual Income (k$)']<=15
0)].value_counts().sum()
```

In [110]:

```
xaxis=["$ 0 - 30,000", "$ 30,001 - 60,000", "$ 60,001 - 90,000", "$ 90,001 - 120,000", "$ 120,001
- 150,000"]
yaxis=[ai1_30,ai31_60,ai61_90,ai91_120,ai121_150]
plt.figure(figsize=(15,6))
sns.barplot(x=xaxis,y=yaxis)
plt.ylabel('Number of customers')
plt.xlabel('Annual Income')
```

```
plt.show()
```

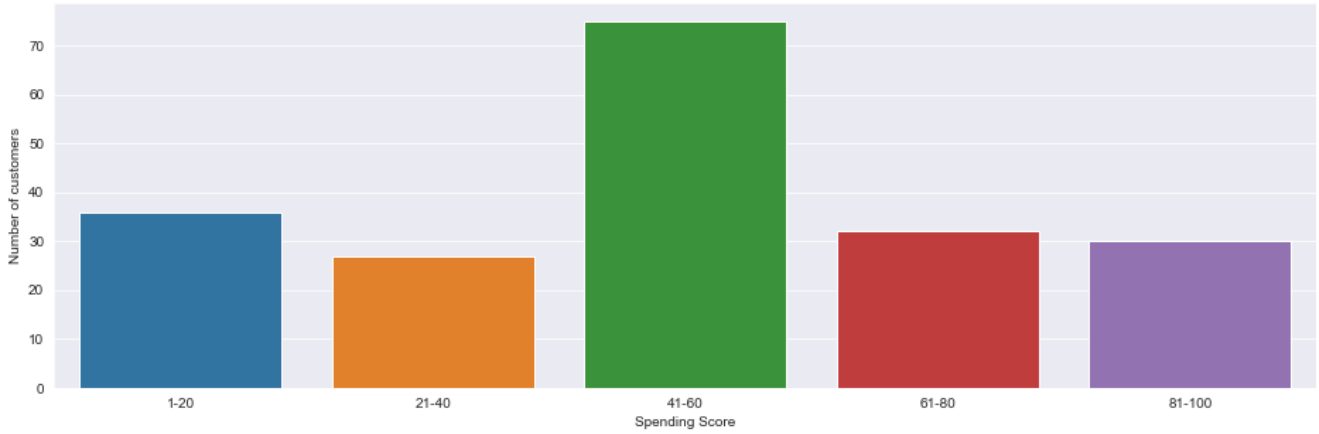


In [111]:

```
ss1_20=df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=1) & (df['Spending Score (1-100
)']<=20)].value_counts().sum()
ss21_40=df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=21) & (df['Spending Score (1-1
00)']<=40)].value_counts().sum()
ss41_60=df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=41) & (df['Spending Score (1-1
00)']<=60)].value_counts().sum()
ss61_80=df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=61) & (df['Spending Score (1-1
00)']<=80)].value_counts().sum()
ss81_100=df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=81) & (df['Spending Score (1-
100)']<=100)].value_counts().sum()
```

In [112]:

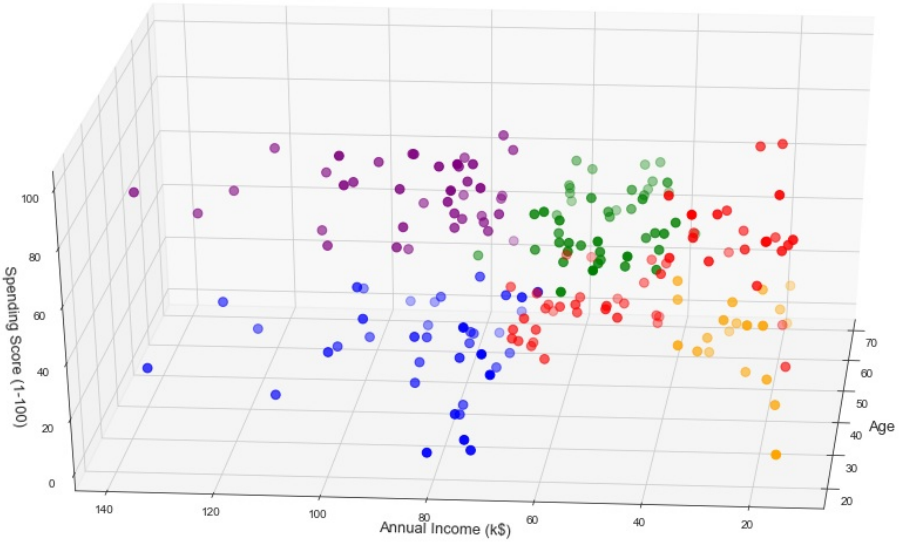
```
xaxis=["1-20", "21-40", "41-60", "61-80", "81-100"]
yaxis=[ss1_20,ss21_40,ss41_60,ss61_80,ss81_100]
plt.figure(figsize=(16,5))
sns.barplot(xaxis,yaxis)
plt.xlabel('Spending Score')
plt.ylabel('Number of customers')
plt.show()
```



In [113]:

```
from mpl_toolkits.mplot3d import Axes3D
sns.set_style('white')
fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(111,projection='3d')
ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Score (1-
100)"][df.label == 0], c='blue', s=60)
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Score (1-
```

```
100)"][df.label == 1], c='red', s=60)
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Score (1-
100)"][df.label == 2], c='green', s=60)
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Score (1-
100)"][df.label == 3], c='orange', s=60)
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Score (1-
100)"][df.label == 4], c='purple', s=60)
ax.view_init(30, 185)
plt.xlabel("Age",fontsize=13)
plt.ylabel("Annual Income (k$)",fontsize=13)
ax.set_zlabel('Spending Score (1-100)',fontsize=13)
plt.show()
```



In [ ]:

```
```