Customer Segmentation

In [5]:

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

In [6]:

```
data = pd.read_csv('Mall_Customers.csv')
data.head()
```

Out[6]:

	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 [7]:

```
data.shape
```

Out[7]:

```
(200, 5)
```

In [8]:

```
data.describe()
```

Out[8]:

	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 [9]:

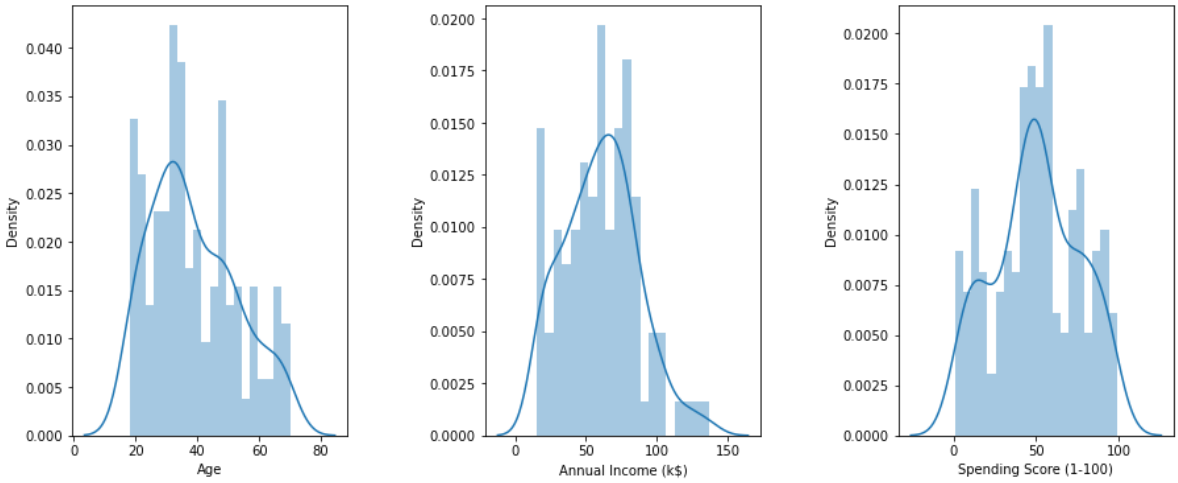
```
data.isnull().sum()
```

Out[9]:

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

In [10]:

```
plt.figure(figsize=(15,6))
n = 0
for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n+=1
    plt.subplot(1,3,n)
    plt.subplots_adjust(hspace = 0.5,wspace = 0.5)
    sns.distplot(data[x],bins = 20)
```



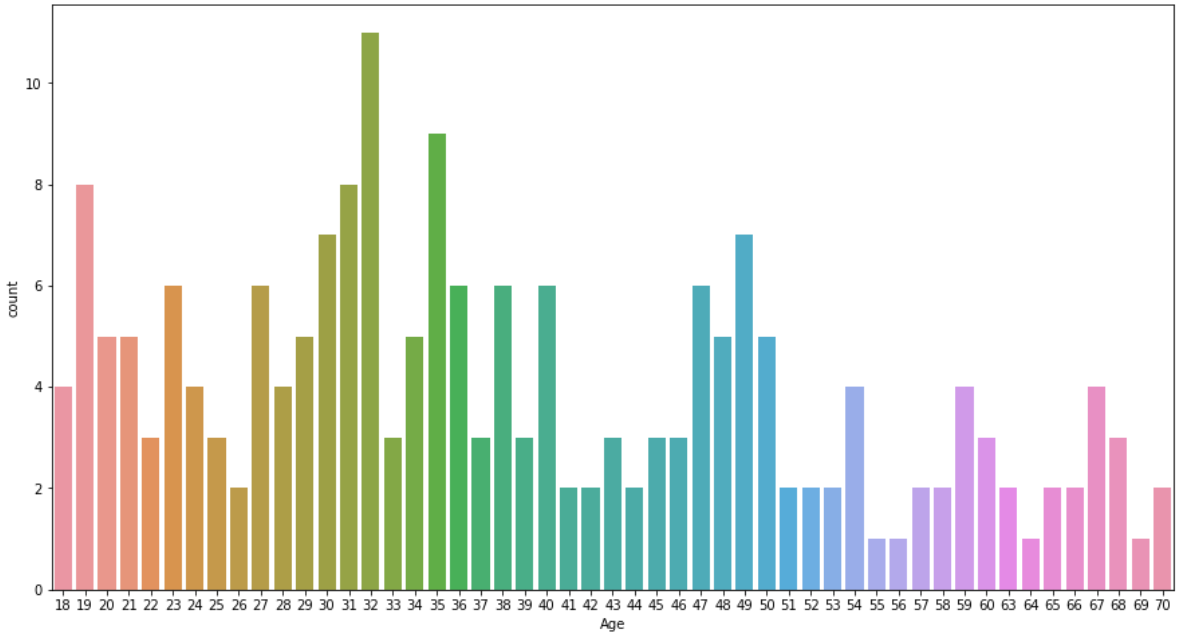
We see here that most of the people age's are from 35 to 40 and make around 60.000 a year

In [11]:

```
plt.figure(figsize=(15,8))
sns.countplot(data=data,x='Age')
```

Out[11]:

```
<AxesSubplot:xlabel='Age', ylabel='count'>
```

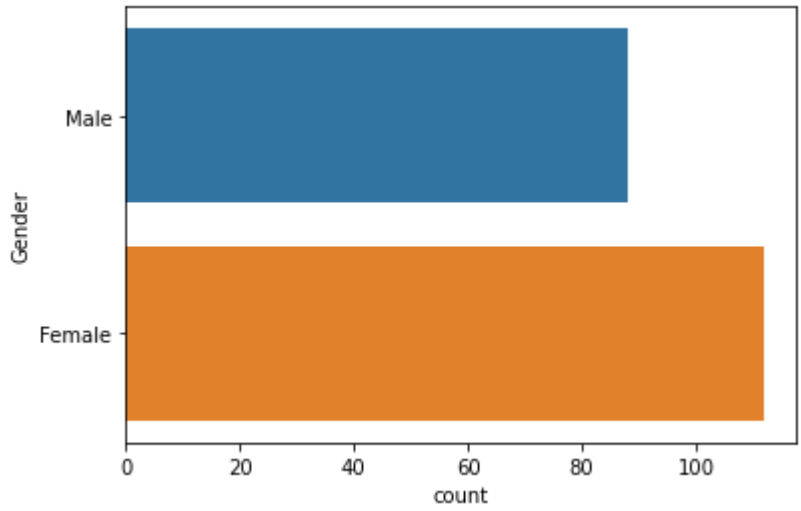


In [12]:

```
sns.countplot(data = data,y ='Gender')
```

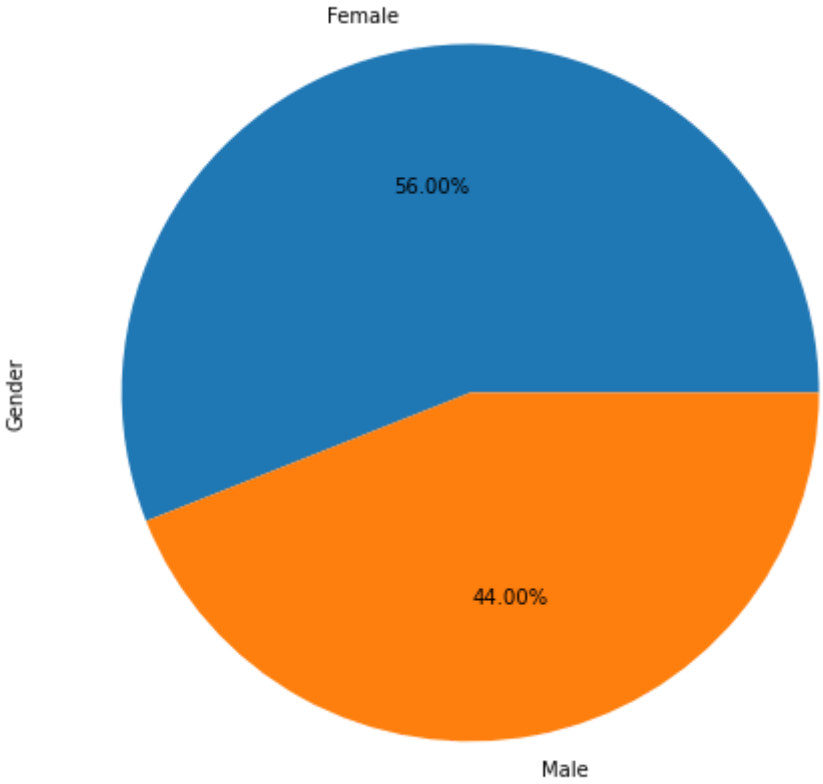
Out[12]:

```
<AxesSubplot:xlabel='count', ylabel='Gender'>
```



In [13]:

```
plt.figure(figsize=(8,8))
data['Gender'].value_counts().plot(kind='pie',autopct='%.2f%%')
plt.show()
```

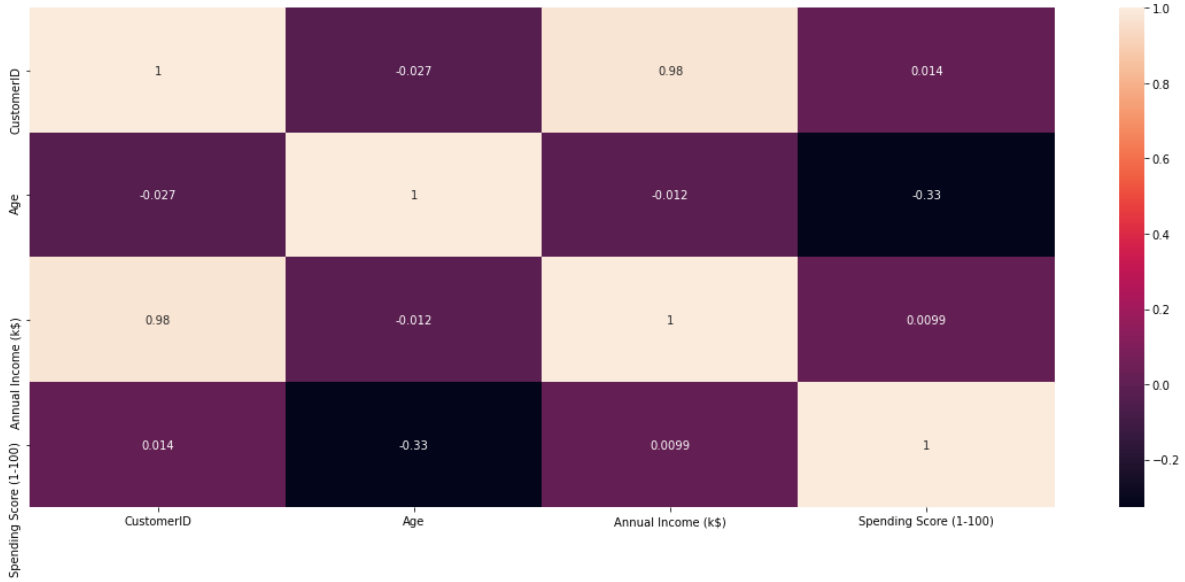


Now we know that most of the customers are women

Now we want to know what range of age is buying the most

In [14]:

```
plt.figure(figsize = (20,8))
sns.heatmap(data.corr(), annot = True)
plt.show()
```



In [15]:

```
Age_18_25 = data.Age[(data.Age >= 18)& (data.Age<=25)]
Age_26_35 = data.Age[(data.Age >= 26)& (data.Age<=35)]
Age_36_45 = data.Age[(data.Age >= 36)& (data.Age<=45)]
Age_46_55 = data.Age[(data.Age >= 46)& (data.Age<=55)]
Age_above_55 = data.Age[data.Age >= 56]
```

In [16]:

```
len(Age_18_25.values)
```

Out[16]:

```
38
```

In [17]:

```
agex = ['Age_18_25','Age_26_35','Age_36_45','Age_46_55','Age_above_55']
agey = [len(Age_18_25.values),len(Age_26_35.values),len(Age_36_45.values),len(Age_46_55.val
```

In [18]:

```
agex
```

Out[18]:

```
['Age_18_25', 'Age_26_35', 'Age_36_45', 'Age_46_55', 'Age_above_55']
```

In [19]:

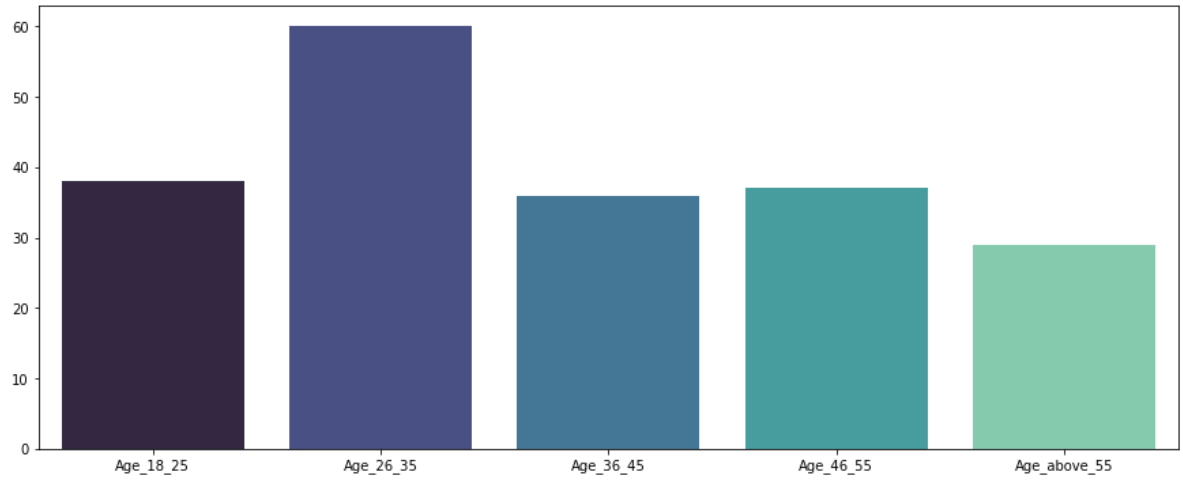
```
agey
```

Out[19]:

```
[38, 60, 36, 37, 29]
```

In [20]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = agex,y = agey , palette='mako')
plt.title = (" Range of age ")
plt.xlable = (" Range of age ")
plt.ylabel = ('No of Customers')
plt.show()
```



Classify the peoble based on their annual income

In [21]:

```
Annual_income_30 = data['Annual Income (k$)'][(data['Annual Income (k$)']>=0)&(data['Annual
Annual_income_60 = data['Annual Income (k$)'][(data['Annual Income (k$)']>=31)&(data['Annua
Annual_income_90 = data['Annual Income (k$)'][(data['Annual Income (k$)']>=61)&(data['Annua
Annual_income_120 = data['Annual Income (k$)'][(data['Annual Income (k$)']>=91)&(data['Annu
Annual_income_150 = data['Annual Income (k$)'][(data['Annual Income (k$)']>=121)&(data['Ann
```

In [22]:

```
AiX = ['0-30','30-60','60-90','90-120','120-150']
AiY =[len(Annual_income_30.values),len(Annual_income_60.values),len(Annual_income_90.values
```

In [23]:

```
AiX
```

Out[23]:

```
['0-30', '30-60', '60-90', '90-120', '120-150']
```

In [24]:

```
AiY
```

Out[24]:

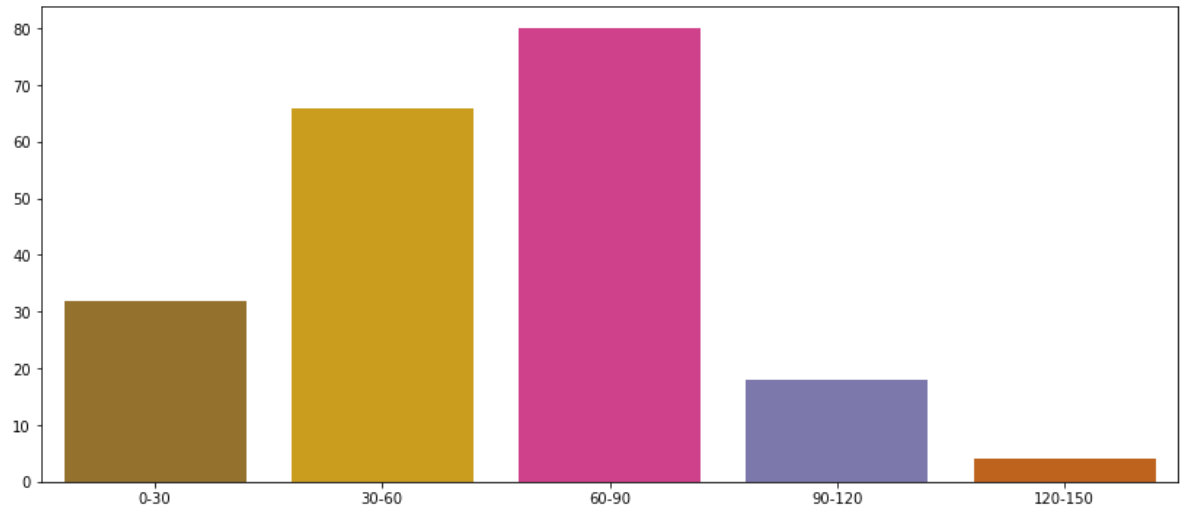
```
[32, 66, 80, 18, 4]
```

In [25]:

```
plt.figure(figsize = (14,6))
sns.barplot(x = AiX,y = AiY,palette = 'Dark2_r')
```

Out[25]:

```
<AxesSubplot:>
```



Most of the people's salary are from 60k to 90k

Cluster the data based on their Age and Annual Income

In [26]:

```
X1 = data.loc[:,['Age','Annual Income (k$)']].values
```

In [27]:

```
from sklearn.cluster import KMeans
```

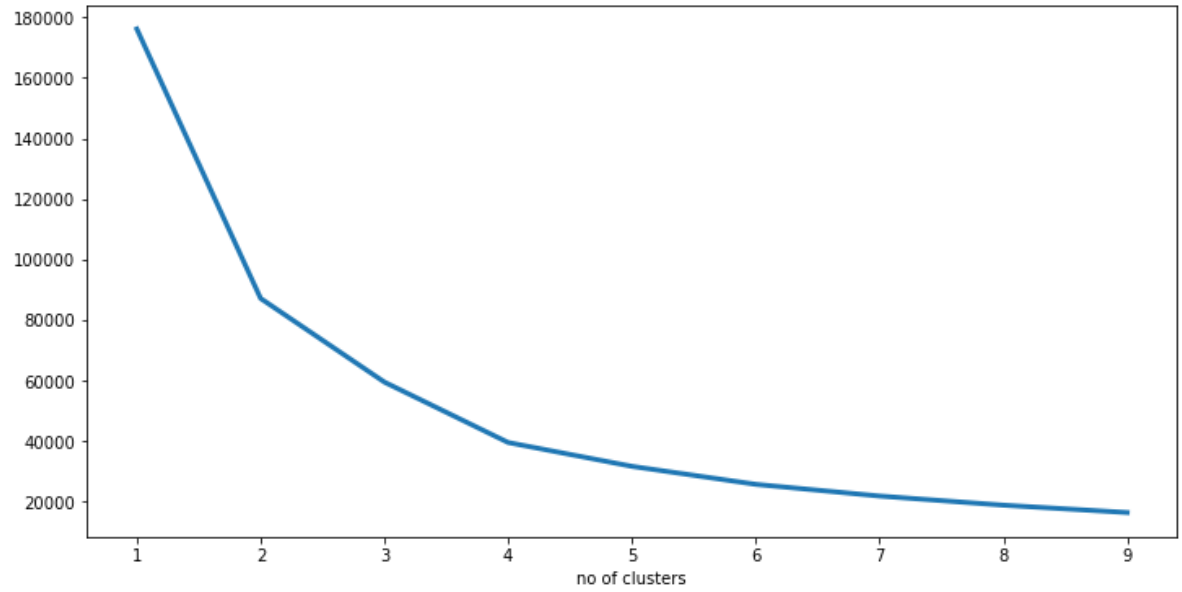
We want to know how many clusters to choose there is a common method called the elbow method

In [28]:

```
acc = []    #append how well the data is clustered when k is from 1 to 9
for k in range(1,10):
  kmeans = KMeans(n_clusters = k)
  kmeans.fit(X1)
  acc.append(kmeans.inertia_)  #Inertia measures how well a dataset was clustered by K-Mean
                               #It is calculated by measuring the distance between each dat

plt.figure(figsize = (12,6))
plt.plot(range(1,10),acc,linewidth = 3)
plt.xlabel("no of clusters")

plt.show()
```



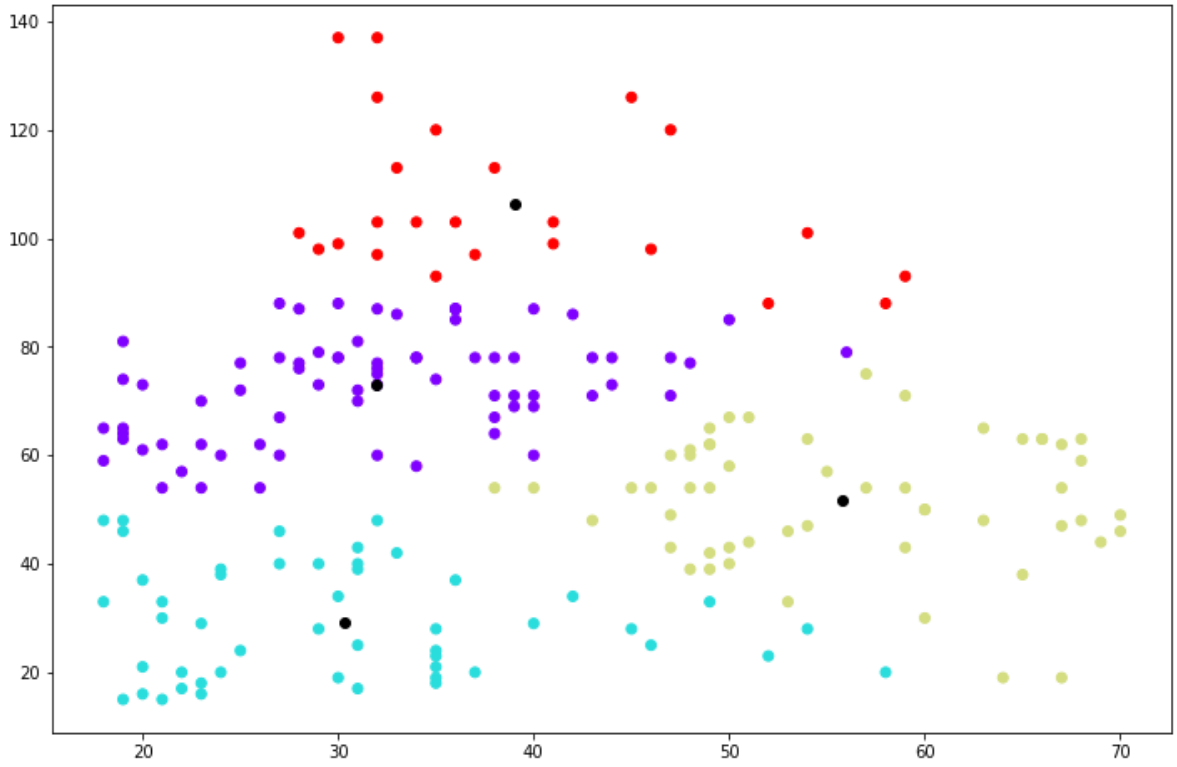
In [29]:

```
kmean = KMeans(n_clusters = 4)
labels = kmean.fit_predict(X1)
print(labels)
```

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

In [30]:

```
plt.figure(figsize = (12,8))
plt.scatter(X1[:,0],X1[:,1],c = kmean.labels_,cmap = 'rainbow')
plt.scatter(kmean.cluster_centers_[:,0],kmean.cluster_centers_[:,1],color = 'black')
plt.show()
```



Cluster the data based on their Annual Income and Spending Score

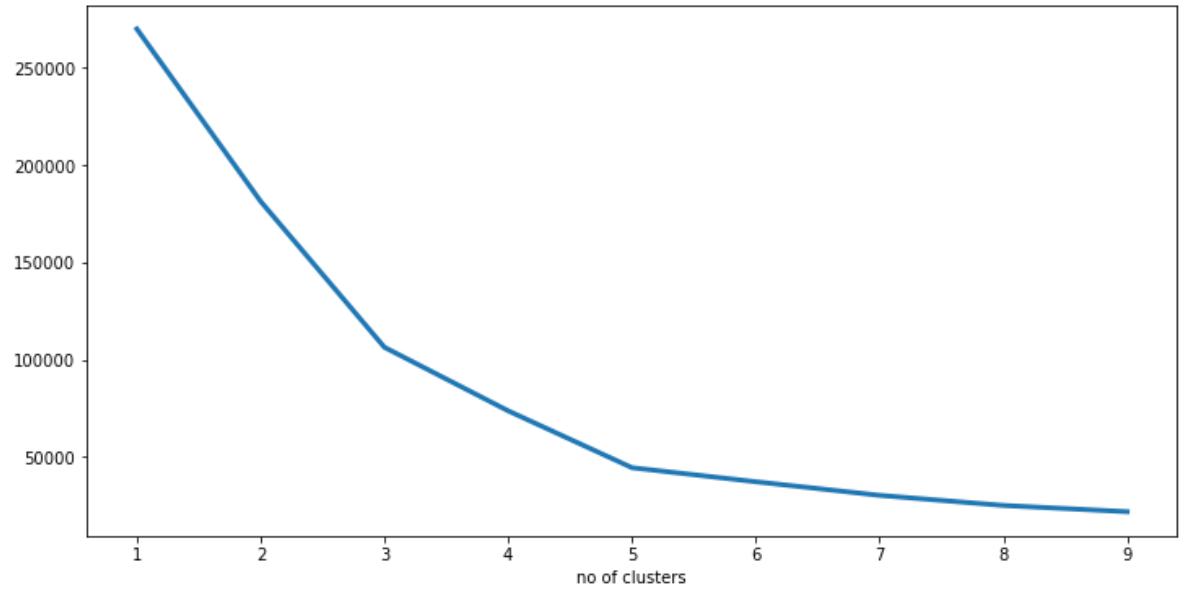
In [31]:

```
X2 = data.loc[:,['Annual Income (k$)','Spending Score (1-100)']].values
```

In [32]:

```
acc = []
for k in range(1,10):
  kmeans = KMeans(n_clusters = k)
  kmeans.fit(X2)
  acc.append(kmeans.inertia_)
plt.figure(figsize = (12,6))
plt.plot(range(1,10),acc,linewidth = 3)
plt.xlabel("no of clusters")

plt.show()
```



In [33]:

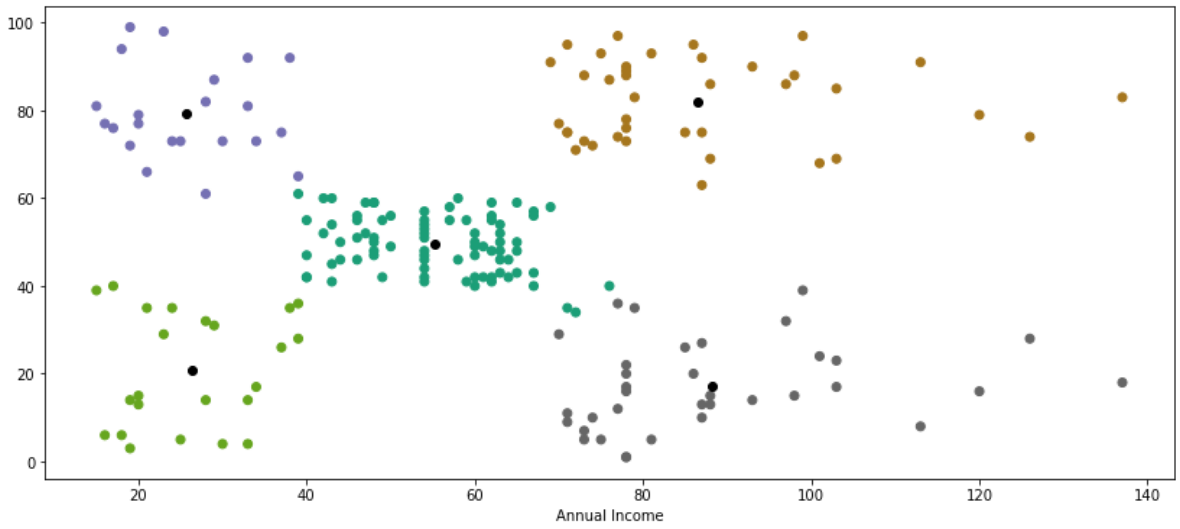
```
kmean2 = KMeans(n_clusters = 5)
labels = kmean2.fit_predict(X2)
print(labels)
```

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

In [34]:

```
plt.figure(figsize = (14,6))
plt.scatter(X2[:,0],X2[:,1],c = kmean2.labels_,cmap = 'Dark2')
plt.scatter(kmean2.cluster_centers_[:,0],kmean2.cluster_centers_[:,1],color = 'black')
plt.xlabel("Annual Income")


plt.show()
```



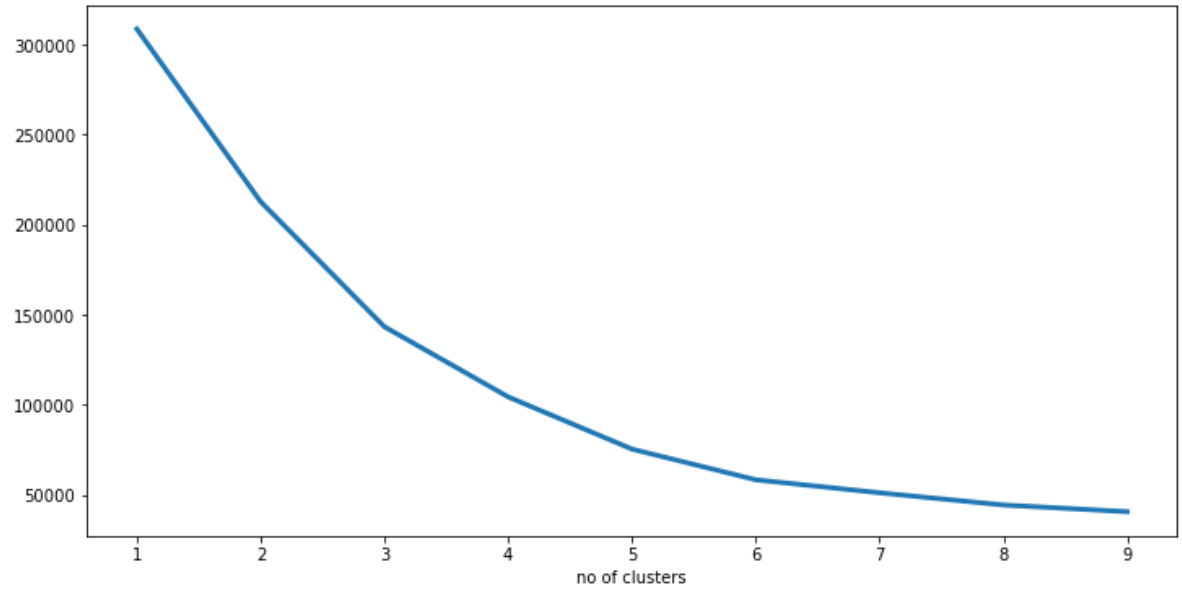
Cluster the data based on their Age and Annual Income and Spending Score

In [35]:

```
X3 = data.loc[:,['Age','Annual Income (k$)','Spending Score (1-100)']].values
```

In [36]:

```
acc = []
for k in range(1,10):
  kmeans = KMeans(n_clusters = k)
  kmeans.fit(X3)
  acc.append(kmeans.inertia_)
plt.figure(figsize = (12,6))
plt.plot(range(1,10),acc,linewidth = 3)
plt.xlabel("no of clusters")
plt.show()
```



In [37]:

```
kmean3 = KMeans(n_clusters = 5)
clusters = kmean3.fit_predict(X2)
data['labels'] = clusters
print(labels)
```

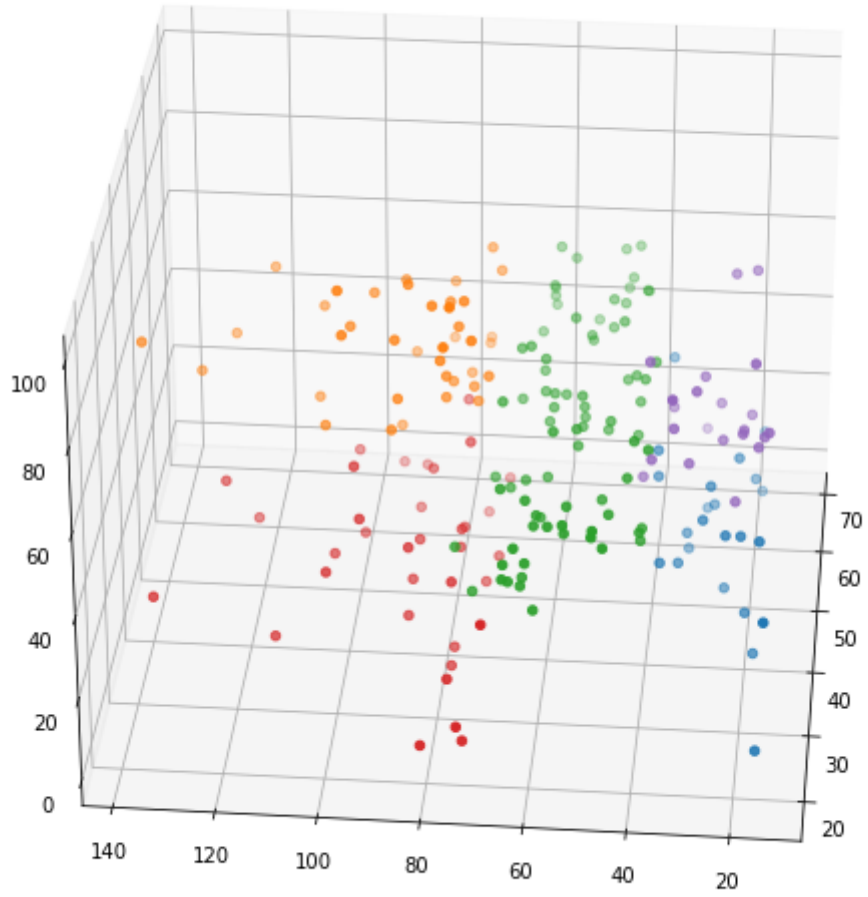
```
[2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
 1 2 1 2 1 2 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 3 4 3 0 3 4 3 4 3 0 3 4 3 4 3 4 3 4 3 0 3 4 3 4 3
 4 3 4 3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

Visualize The Clusters in 3D

In [38]:

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(111,projection = '3d')
ax.scatter(data.Age[data.labels==0],data['Annual Income (k$)'][data.labels == 0],data['Spen
ax.scatter(data.Age[data.labels==1],data['Annual Income (k$)'][data.labels == 1],data['Spen
ax.scatter(data.Age[data.labels==2],data['Annual Income (k$)'][data.labels == 2],data['Spen
ax.scatter(data.Age[data.labels==3],data['Annual Income (k$)'][data.labels== 3],data['Spend
ax.scatter(data.Age[data.labels==4],data['Annual Income (k$)'][data.labels == 4],data['Spen
ax.view_init(30,185)

plt.show()
```



In [ ]:

```
```