In [ ]:

```
#Praveenkumar
```

In [31]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [32]:

```
df= pd.read_csv('Mall_Customers.csv')
df
```

Out[32]:

	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
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [33]:

```
print(df.head())
```

```
   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 [34]:

```
print(df.head(2))
print(df.shape)
```

```
   CustomerID Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1   Male   19                  15                      39
1           2   Male   21                  15                      81
(200, 5)
```

In [35]:

```
print(df.isnull().sum())
```

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

In [36]:

```
df.describe()
```

Out[36]:

	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 [37]:

```
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 [38]:

```
df.nlargest(10,'Spending Score (1-100)')
```

Out[38]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
11	12	Female	35	19	99
19	20	Female	35	23	98
145	146	Male	28	77	97
185	186	Male	30	99	97
127	128	Male	40	71	95
167	168	Female	33	86	95
7	8	Female	23	18	94
141	142	Male	32	75	93
163	164	Female	31	81	93
33	34	Male	18	33	92

In [39]:

```
df.nsmallest(10,'Spending Score (1-100)')
```

Out[39]:

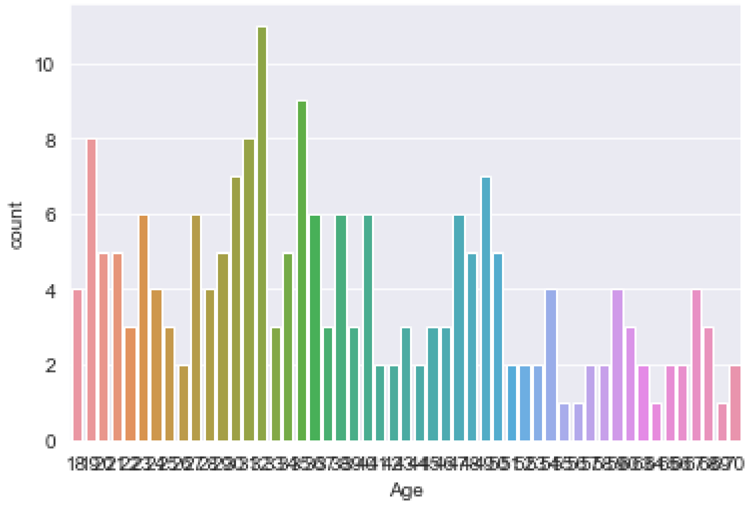
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
156	157	Male	37	78	1
158	159	Male	34	78	1
8	9	Male	64	19	3
30	31	Male	60	30	4
32	33	Male	53	33	4
22	23	Female	46	25	5
134	135	Male	20	73	5
140	141	Female	57	75	5
162	163	Male	19	81	5
2	3	Female	20	16	6

In [40]:

```
sb.countplot(x=df['Age'])
```

Out[40]:

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

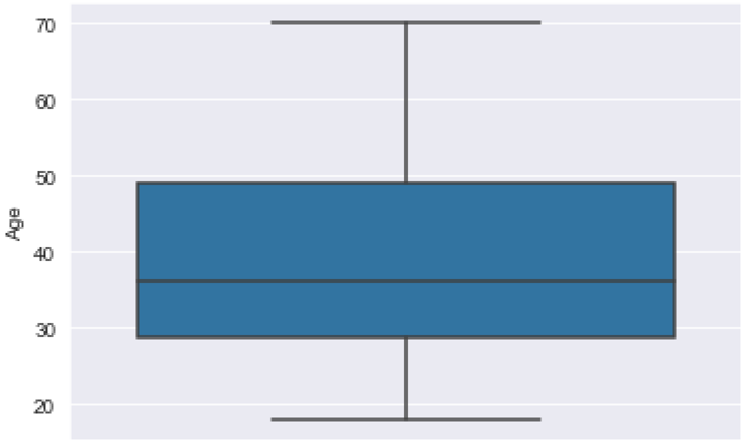


In [41]:

```
sb.boxplot(y=df['Age'])
```

Out[41]:

```
<AxesSubplot:ylabel='Age'>
```



In [42]:

```
sb.set_style('darkgrid')
```

In [43]:

```
x=df['Spending Score (1-100)']
```

In [44]:

```
y=df['Annual Income (k$)']
```

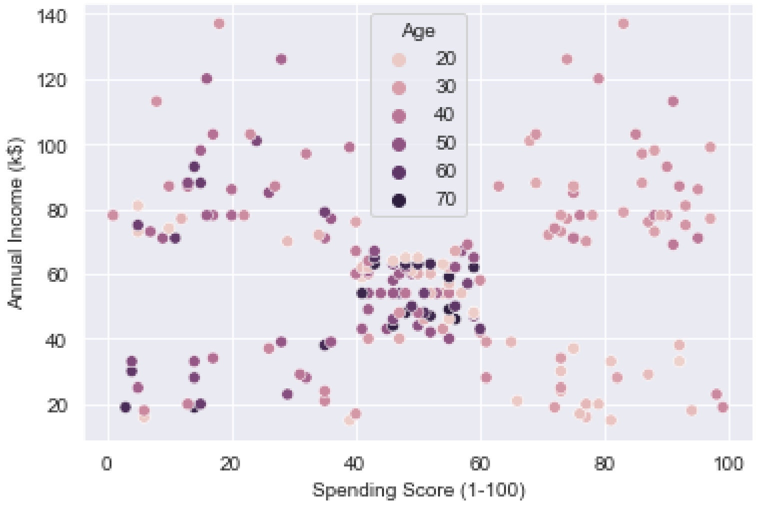
In [45]:

```
sb.scatterplot(x,y,hue=df['Age'])
```

```
E:\analytics\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pas
s the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments withou
t an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[45]:

```
<AxesSubplot:xlabel='Spending Score (1-100)', ylabel='Annual Income (k$)'>
```



In [49]:

```
Age_18_20=df[(df['Age']>=18)&(df['Age']<=20)]
Age_20_25=df[(df['Age']>20)&(df['Age']<=25)]
Age_25_30=df[(df['Age']>25)&(df['Age']<=30)]
Age_30_35=df[(df['Age']>30)&(df['Age']<=35)]
Age_35_40=df[(df['Age']>35)&(df['Age']<=40)]
Age_40_45=df[(df['Age']>40)&(df['Age']<=45)]
Age_45_50=df[(df['Age']>45)&(df['Age']<=50)]
Age_50_55=df[(df['Age']>50)&(df['Age']<=55)]
Age_55_60=df[(df['Age']>55)&(df['Age']<=60)]
Age_60_65=df[(df['Age']>60)&(df['Age']<=65)]
Age_65_70=df[(df['Age']>65)&(df['Age']<=70)]
xaxis=['18_20','20_25','25_30','30_35','35_40','40_45','45_50','50_55','55_60','60_65','65_
yaxis=[len(Age_18_20),len(Age_20_25),len(Age_25_30),len(Age_30_35),len(Age_35_40),len(Age_4
```

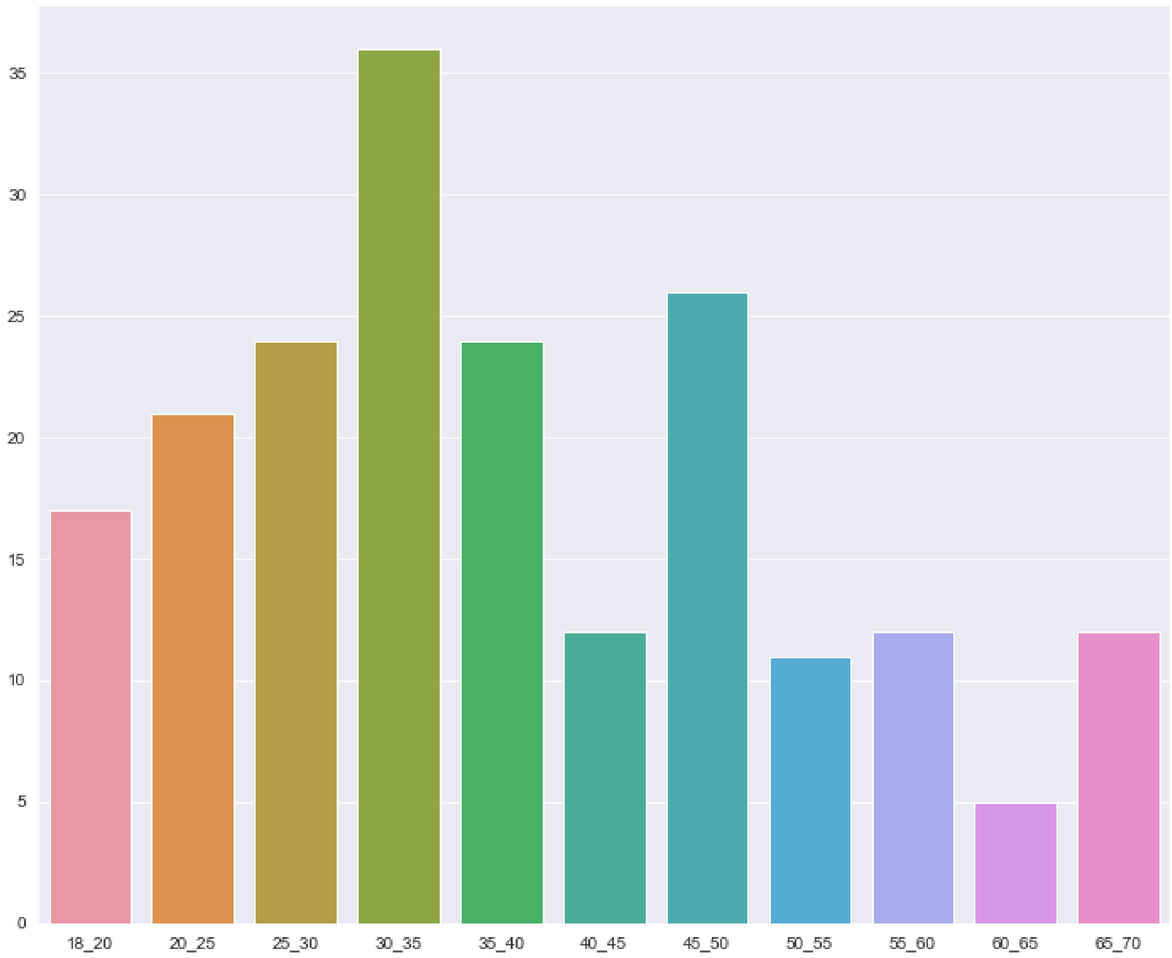
In [50]:

```
plt.figure(figsize=(12,10))
sb.barplot(xaxis,yaxis)
```

```
E:\analytics\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pas
s the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments withou
t an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[50]:

```
<AxesSubplot:>
```



In [ ]:

In [ ]: