Customer Segmentation for Marketing Analysis

DSC 680 - Applied Data Science

Andrews_Bill

Importing the Libraries

In [38]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore")
```

Loading the Dataset

In [39]:
```
df = pd.read_csv('C:\\Users\\Billa\\OneDrive\\Desktop\\DSC 680\\Mall_Customers.csv')
df.head()
```

Out[39]:

	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

Understanding the Dataset

In [40]:
```
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 [41]:
```
#Remove CustomerID and Gender as variables from the dataset.  They are not nee
ded.
df.drop(["CustomerID"], axis = 1, inplace=True)
df.drop(["Gender"], axis = 1, inplace=True)
```

In [42]:
```
df.shape
```

Out[42]:
```
(200, 3)
```

In [43]:
```
df.sample(20)
```

Out[43]:

	Age	Annual Income (k$)	Spending Score (1-100)
170	40	87	13
76	45	54	53
73	60	50	56
40	65	38	35
7	23	18	94
22	46	25	5
98	48	61	42
18	52	23	29
94	32	60	42
163	31	81	93
39	20	37	75
97	27	60	50
107	54	63	46
193	38	113	91
136	44	73	7
103	26	62	55
146	48	77	36
128	59	71	11
25	29	28	82
2	20	16	6

In [44]:
```
df.describe()
```

Out[44]:

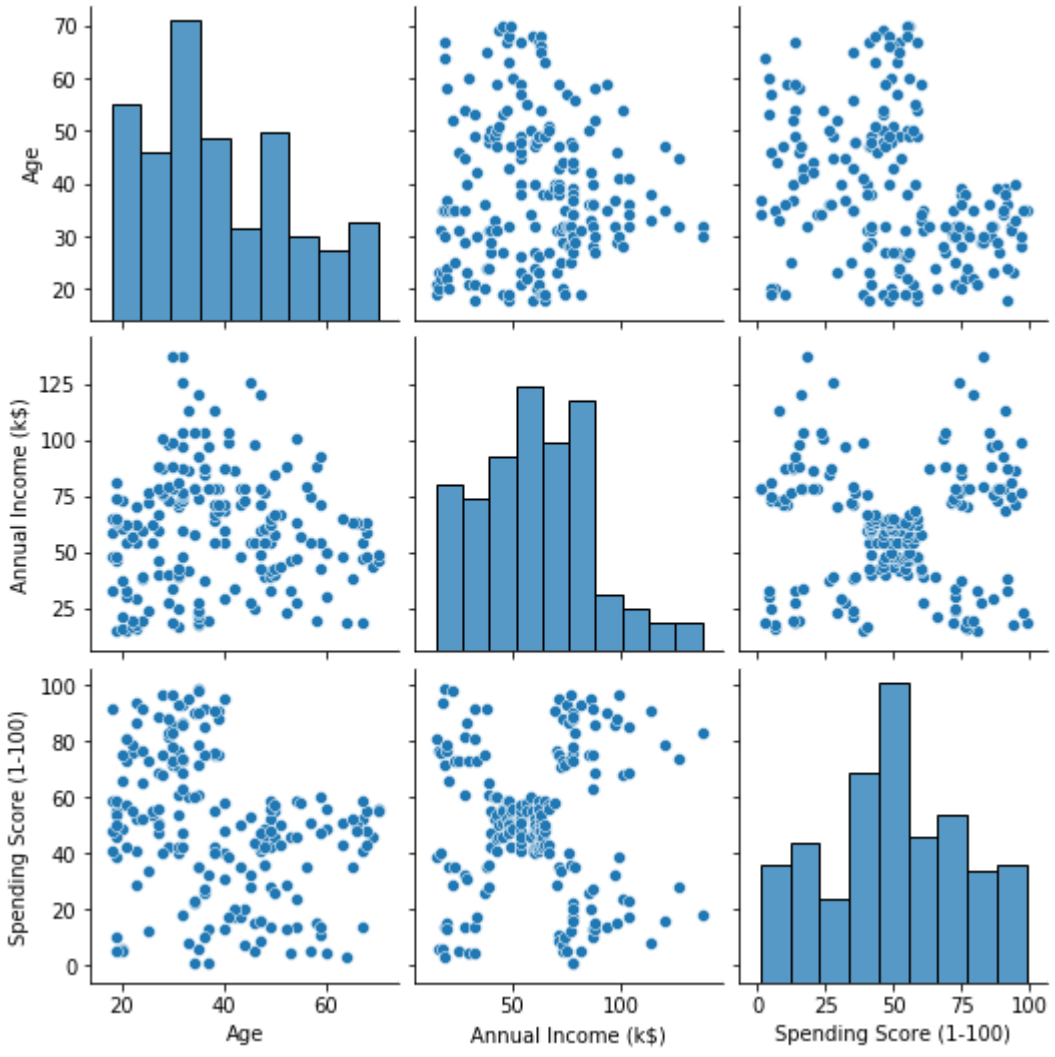
	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000
mean	38.850000	60.560000	50.200000
std	13.969007	26.264721	25.823522
min	18.000000	15.000000	1.000000
25%	28.750000	41.500000	34.750000
50%	36.000000	61.500000	50.000000
75%	49.000000	78.000000	73.000000
max	70.000000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

In [45]:
```
df.isnull().sum()
```

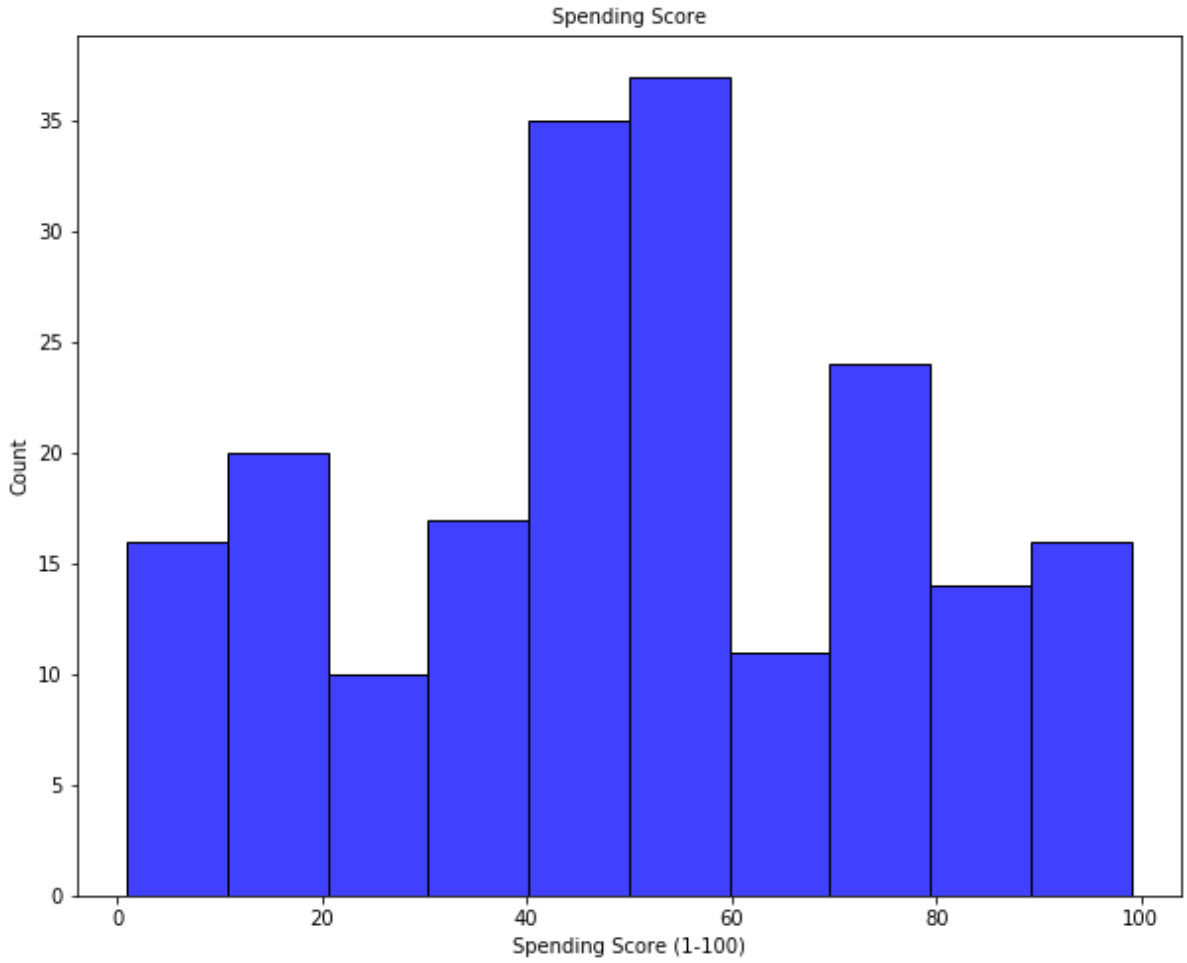
Out[45]:
```
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

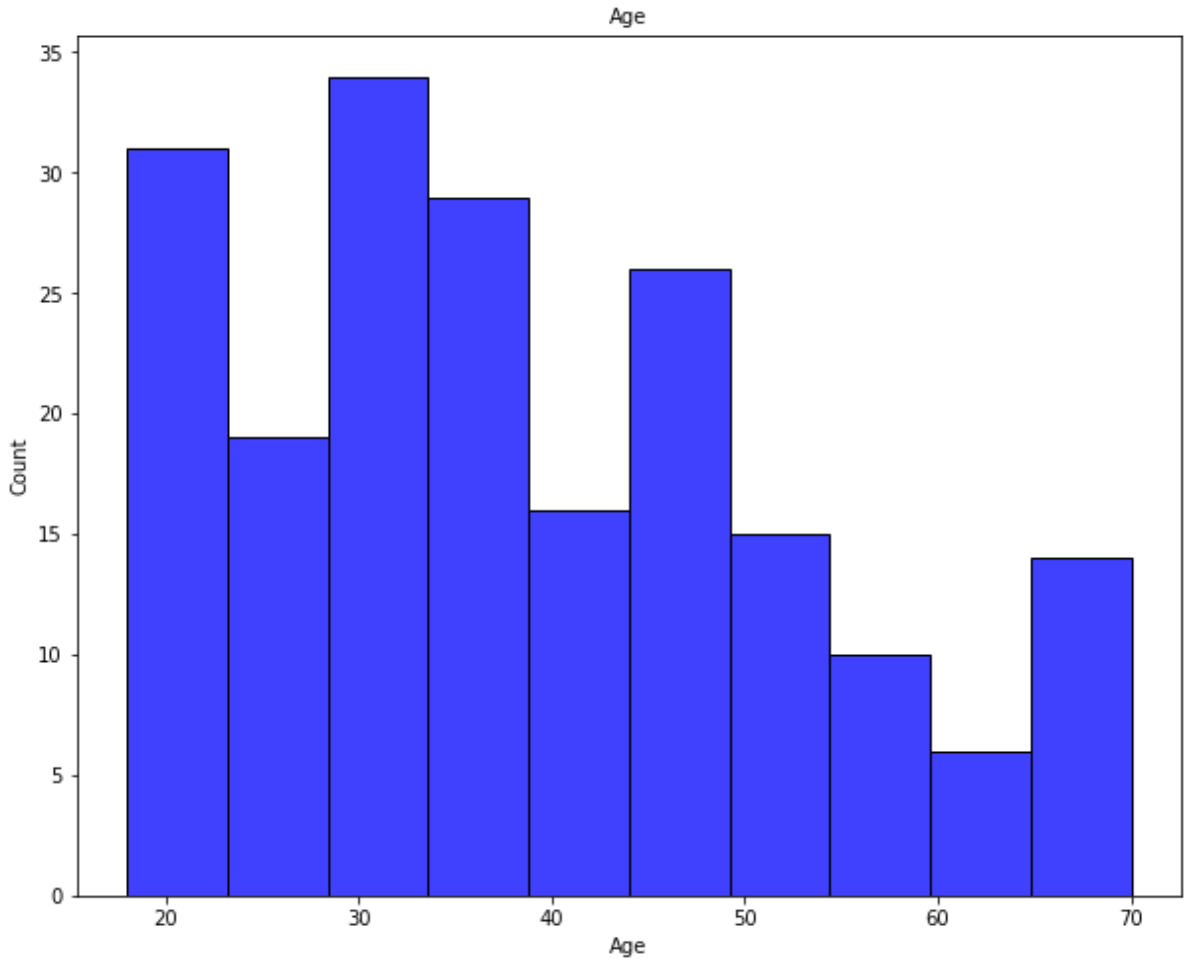
Exploratory Data Analysis (EDA)

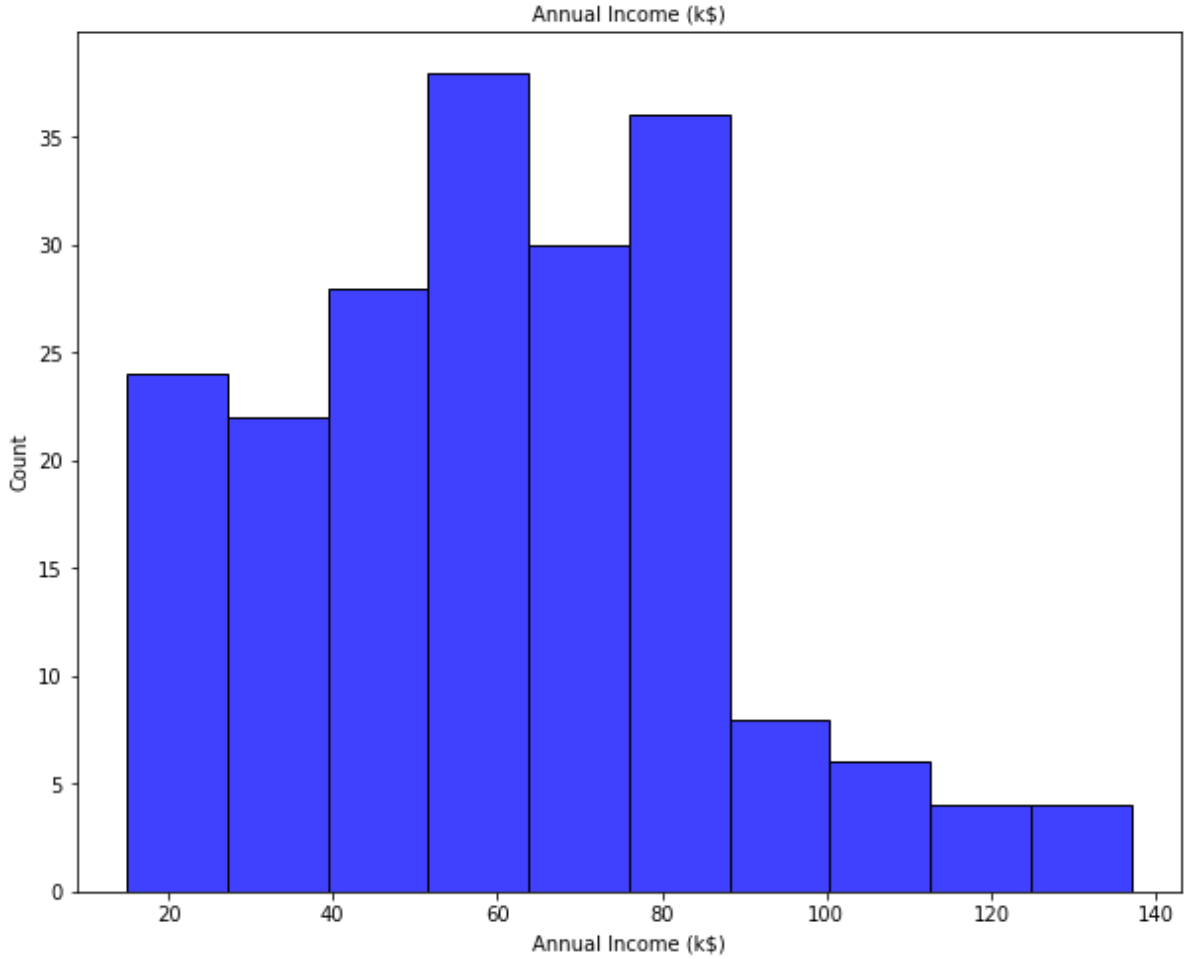
In [46]:
```
#Pairwise comparsion between Variables of the Database

sns.pairplot(df)
plt.show()
```



In [78]:
```
#Histogram distrubtion of Spending Score, Age, and Annual Income
plt.figure(figsize=(10,8))
sns.histplot(df['Spending Score (1-100)'], bins= 10, color='b')
plt.title("Spending Score", fontsize = 10)
plt.show()

#Between 40 and 60 has the highest population of spending scores

plt.figure(figsize=(10,8))
sns.histplot(df['Age'], bins= 10, color='b')
plt.title("Age", fontsize = 10)
plt.show()

#Between the ages of 30 and 40 has the largest population

plt.figure(figsize=(10,8))
sns.histplot(df['Annual Income (k$)'], bins= 10, color='b')
plt.title("Annual Income (k$)", fontsize = 10)
plt.show()

#Between $60K and $80k has the highest population of annual income
```

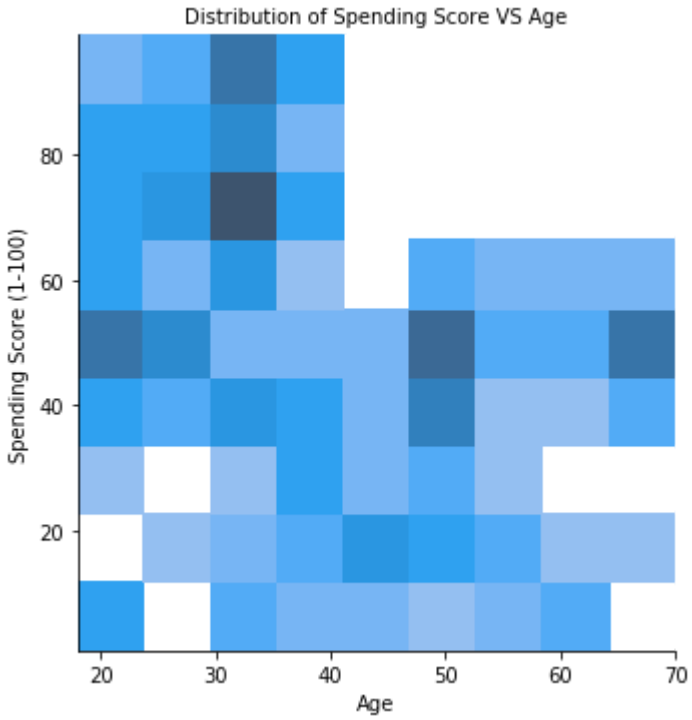

Spending Score
Count
Spending Score (1-100)
0
5
10
15
20
25
30
35
0
20
40
60
80
100


Age
Count
Age
0
5
10
15
20
25
30
35
20
30
40
50
60
70

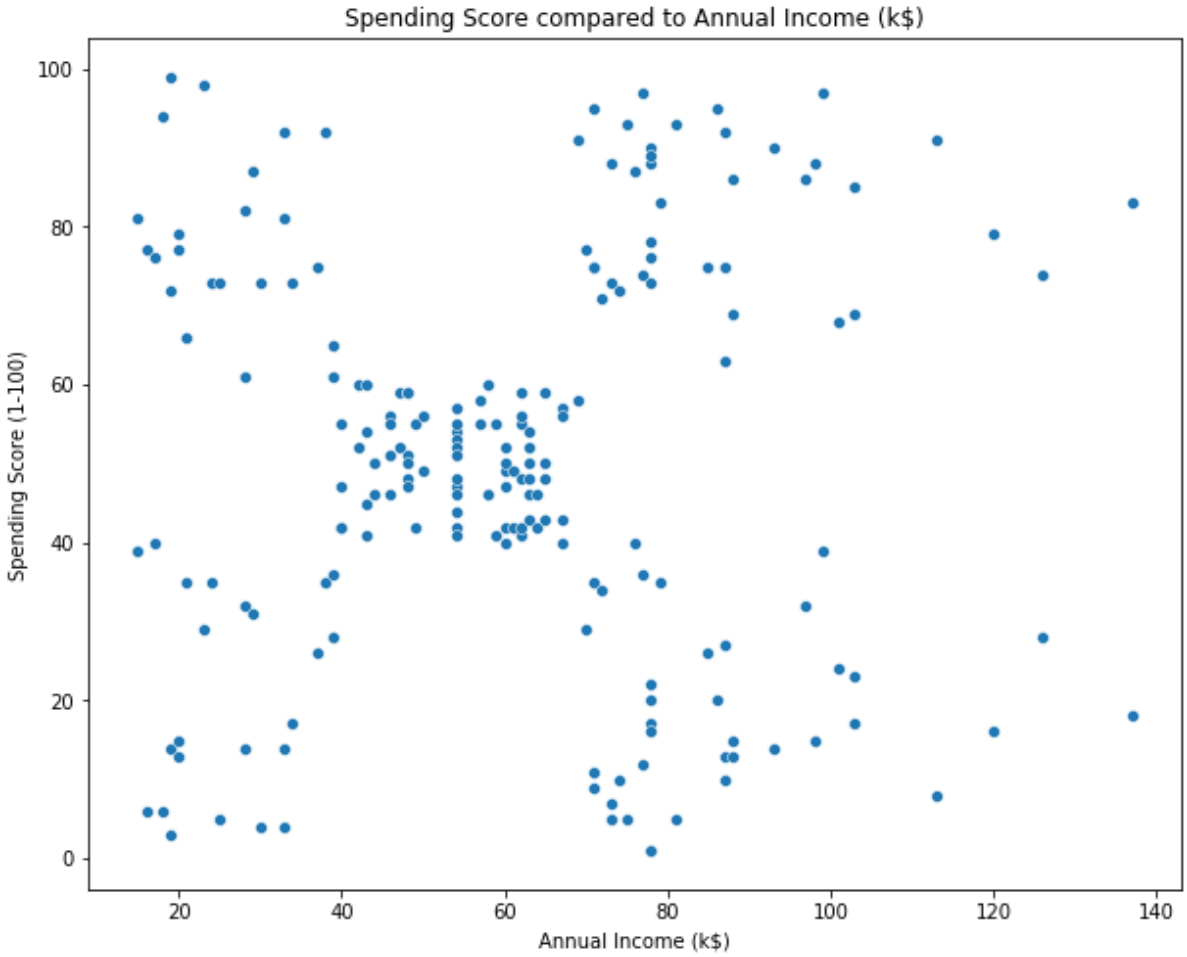

Annual Income (k$)
Count
0
5
10
15
20
25
30
35
20
40
60
80
100
120
140
Annual Income (k$)

In [79]:
```
plt.figure(figsize=(12,10))
sns.displot(data= df,y='Spending Score (1-100)',x='Age')
plt.title("Distribution of Spending Score VS Age", fontsize = 10)
plt.show()

#Between the ages of 20 and 40 have the highest spending scores
```

```
<Figure size 864x720 with 0 Axes>
```

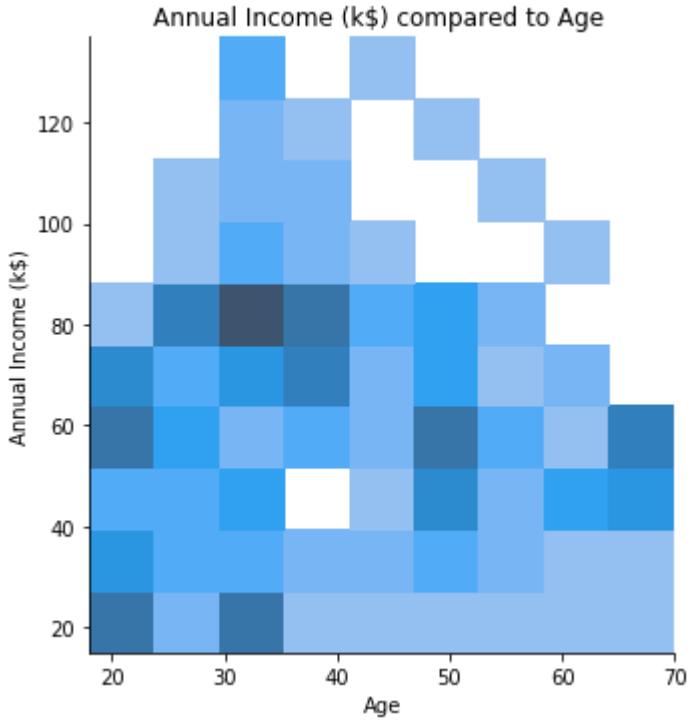


In [81]:
```
plt.figure(figsize=(10,8))
sns.scatterplot(data= df,y='Spending Score (1-100)',x='Annual Income (k$)')
plt.title("Spending Score compared to Annual Income (k$)", fontsize = 12)
plt.show()
```



In [82]:
```
plt.figure(figsize=(12,10))
sns.displot(data= df,y='Annual Income (k$)',x='Age')
plt.title("Annual Income (k$) compared to Age ", fontsize = 12)
plt.show()

#Between the ages of 30 and 50 have the highest annual income
```

```
<Figure size 864x720 with 0 Axes>
```

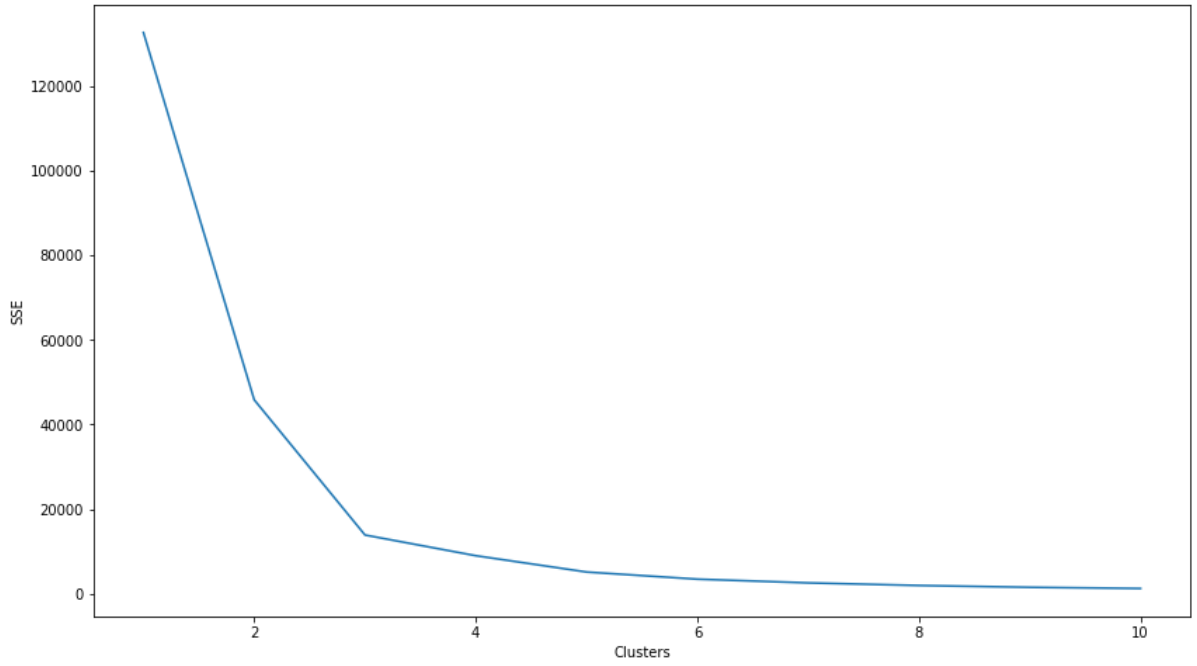


In [51]:
```
X = df.iloc[:,2:]
X.shape
```

Out[51]: (200, 1)

KMeans Clustering

In [52]:
```
from sklearn.cluster import KMeans
Inter = []
for i in range(1,11):
  model = KMeans(n_clusters = i)
  model.fit(X)
  Inter.append(model.inertia_)

# Using Elbow to determine number of clusters
plt.figure(figsize = (14, 8))
plt.plot(np.arange(1,11), Inter)
plt.xlabel('Clusters')
plt.ylabel('SSE')
plt.show()
```



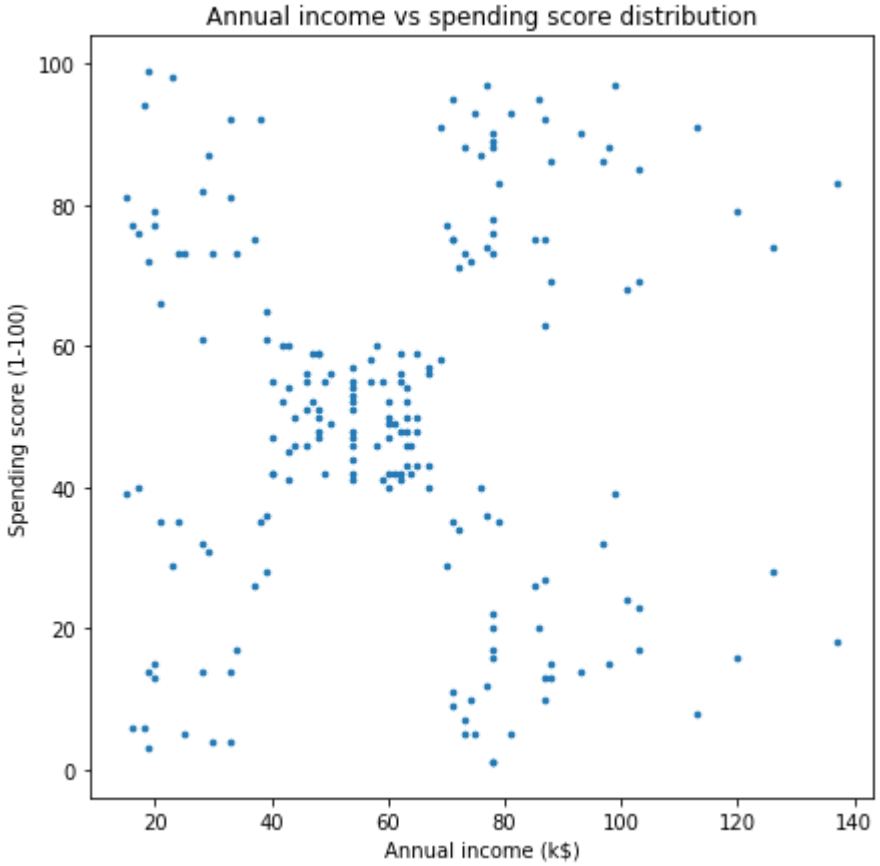
In [53]:
```
K_model = KMeans(n_clusters = 5)
K_model.fit(X)
```

Out[53]:
```
KMeans(n_clusters=5)
```

In [70]:
```
X = df[['Annual Income (k$)', 'Spending Score (1-100)']].values
X[:5]
```

Out[70]:
```
array([[15, 39],
       [15, 81],
       [16,  6],
       [16, 77],
       [17, 40]], dtype=int64)
```

In [71]:
```
plt.figure(figsize=(7,7))
plt.scatter(X[:,0], X[:,1], s=7)
plt.title('Annual income vs spending score distribution')
plt.xlabel('Annual income (k$)')
plt.ylabel('Spending score (1-100)')
plt.show()
```



In [73]:
```
kmeans = KMeans(n_clusters=5, random_state=44)
```

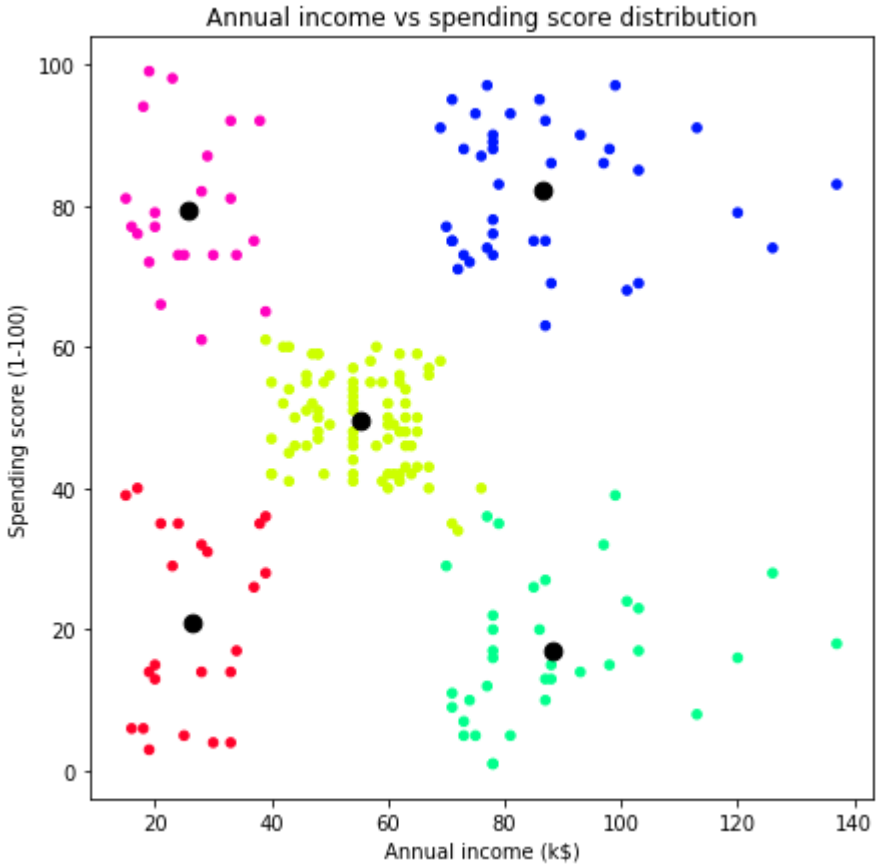
In [74]:
```
kmeans.fit(X)
```

Out[74]:
```
KMeans(n_clusters=5, random_state=44)
```

In [75]:
```
y_kmeans = kmeans.predict(X)
y_kmeans
```

Out[75]:
```
array([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, 1,
       0, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 2, 3, 1, 3, 2, 3, 2, 3,
       1, 3, 2, 3, 2, 3, 2, 3, 2, 3, 1, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3])
```

In [76]:
```
centroids = kmeans.cluster_centers_
plt.figure(figsize=(7,7))
plt.scatter(X[:,0], X[:,1], s=20, c=y_kmeans, cmap='gist_rainbow')
plt.scatter(centroids[:,0], centroids[:,1], s=75, c='black')
plt.title('Annual income vs spending score distribution')
plt.xlabel('Annual income (k$)')
plt.ylabel('Spending score (1-100)')
plt.show()
```



In [ ]: