In [55]:

```
from matplotlib import pyplot as plt
import pandas as pd
import pandas.util.testing as tm
import seaborn as sns
import numpy as np
%matplotlib inline
```

In [63]:

```
print("Seaborn Version "+sns.__version__)
print("Pandas Version "+pd.__version__)
print("Numpy Version "+np.__version__)
import matplotlib
print("Matplotlib Version "+matplotlib.__version__)
```

```
Seaborn Version 0.9.0
Pandas Version 1.1.0
Numpy Version 1.16.5
Matplotlib Version 3.1.1
```

In [35]:

```
df = pd.read_csv("Mall_Customers.csv")
df.head(10)
```

Out[35]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

In [36]:

```
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 [37]:

```
#Using the .corr method from pandas to see the correlation, default is Pearson Corellation
df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].corr()
```

Out[37]:

	Age	Annual Income (k$)	Spending Score (1-100)
Age	1.000000	-0.012398	-0.327227
Annual Income (k$)	-0.012398	1.000000	0.009903
Spending Score (1-100)	-0.327227	0.009903	1.000000

Customer Gender Visualization

In [38]:

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

Out[38]:

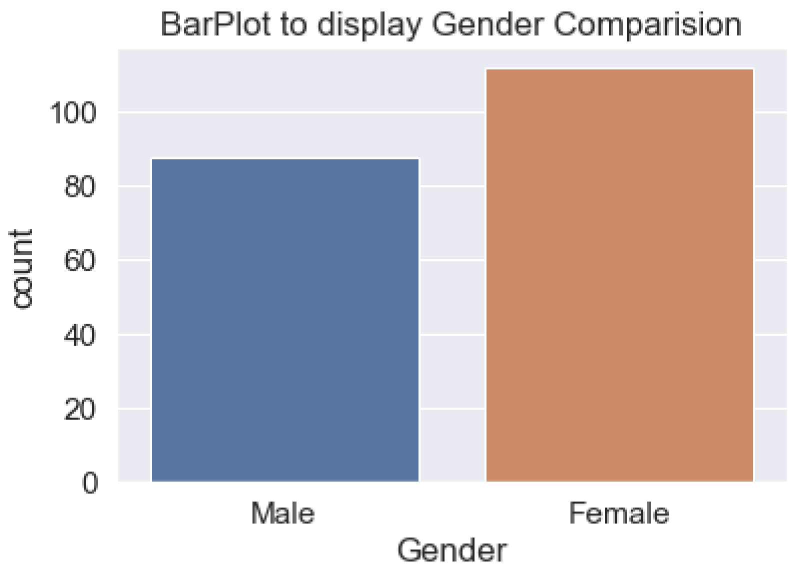
```
count        200
unique         2
top       Female
freq         112
Name: Gender, dtype: object
```

In [39]:

```
sns.countplot(x='Gender',data=df, ).set(title='BarPlot to display Gender Comparision')
```

Out[39]:

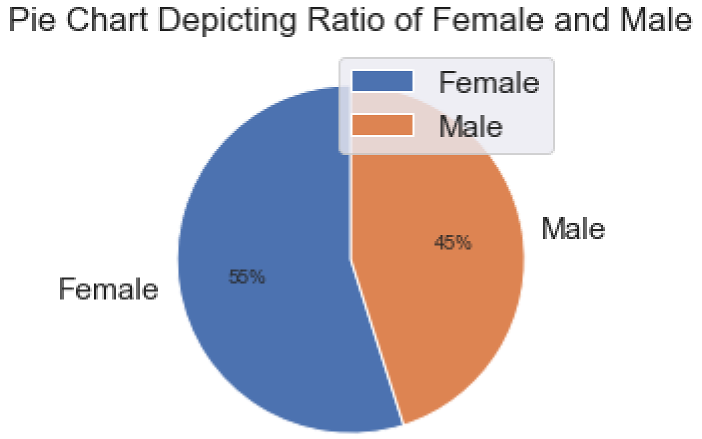
```
[Text(0.5, 1.0, 'BarPlot to display Gender Comparision')]
```



From the above barplot, we can see that the number of females is higher than the males.

In [40]:

```
sums = df.groupby(df["Gender"])["Age"].sum()
plt.pie(sums, labels=sums.index,autopct='%1.0f%%', startangle=90)
plt.legend()
plt.title("Pie Chart Depicting Ratio of Female and Male")
plt.show()
```



From the above graph, we conclude that the percentage of females is 55%, whereas the percentage of male in the customer dataset is 45%.

Visualization of Age Distribution

In [22]:

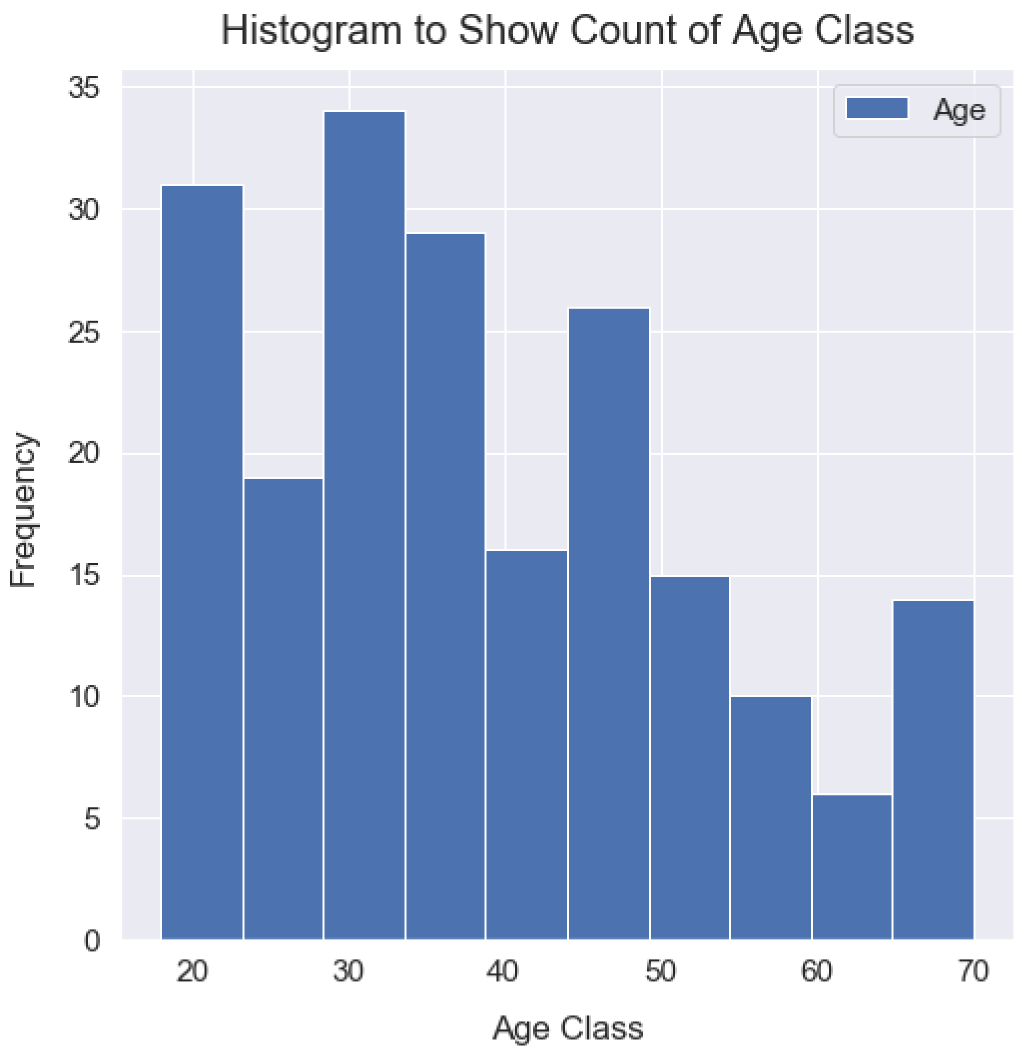
```
df['Age'].describe()
```

Out[22]:

```
count    200.000000
mean      38.850000
std       13.969007
min       18.000000
25%       28.750000
50%       36.000000
75%       49.000000
max       70.000000
Name: Age, dtype: float64
```

In [16]:

```
sns.set(font_scale=1.4)
df['Age'].plot(kind='hist', figsize=(8, 8));
plt.xlabel("Age Class", labelpad=14)
plt.ylabel("Frequency", labelpad=14)
plt.legend()
plt.title("Histogram to Show Count of Age Class", y=1.015, fontsize=20);
```

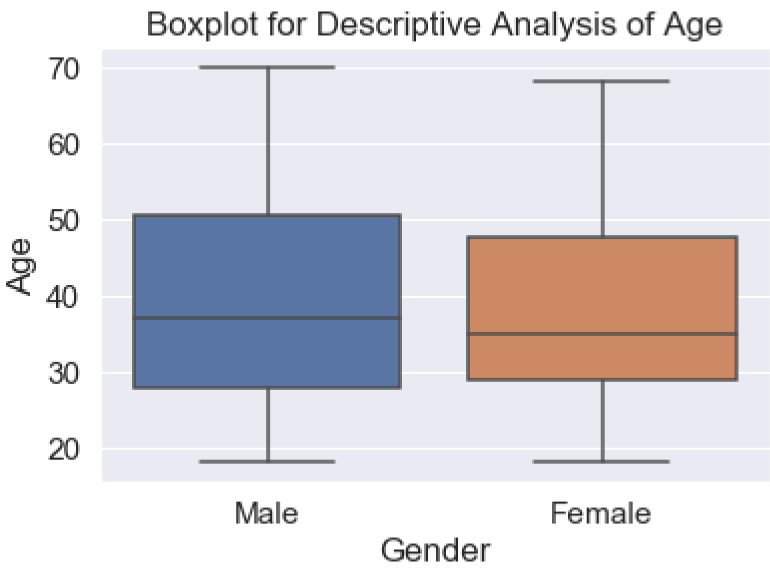


In [17]:

```
sns.boxplot(data=df, x='Gender', y='Age').set(title='Boxplot for Descriptive Analysis of Ag
```

Out[17]:

```
[Text(0.5, 1.0, 'Boxplot for Descriptive Analysis of Age')]
```



From the above two visualizations, we conclude that the maximum customer ages are between 30 and 35. The minimum age of customers is 18, whereas, the maximum age is 70.

Analysis of the Annual Income of the Customers

In [23]:

```
df['Annual Income (k$)'].describe()
```

Out[23]:

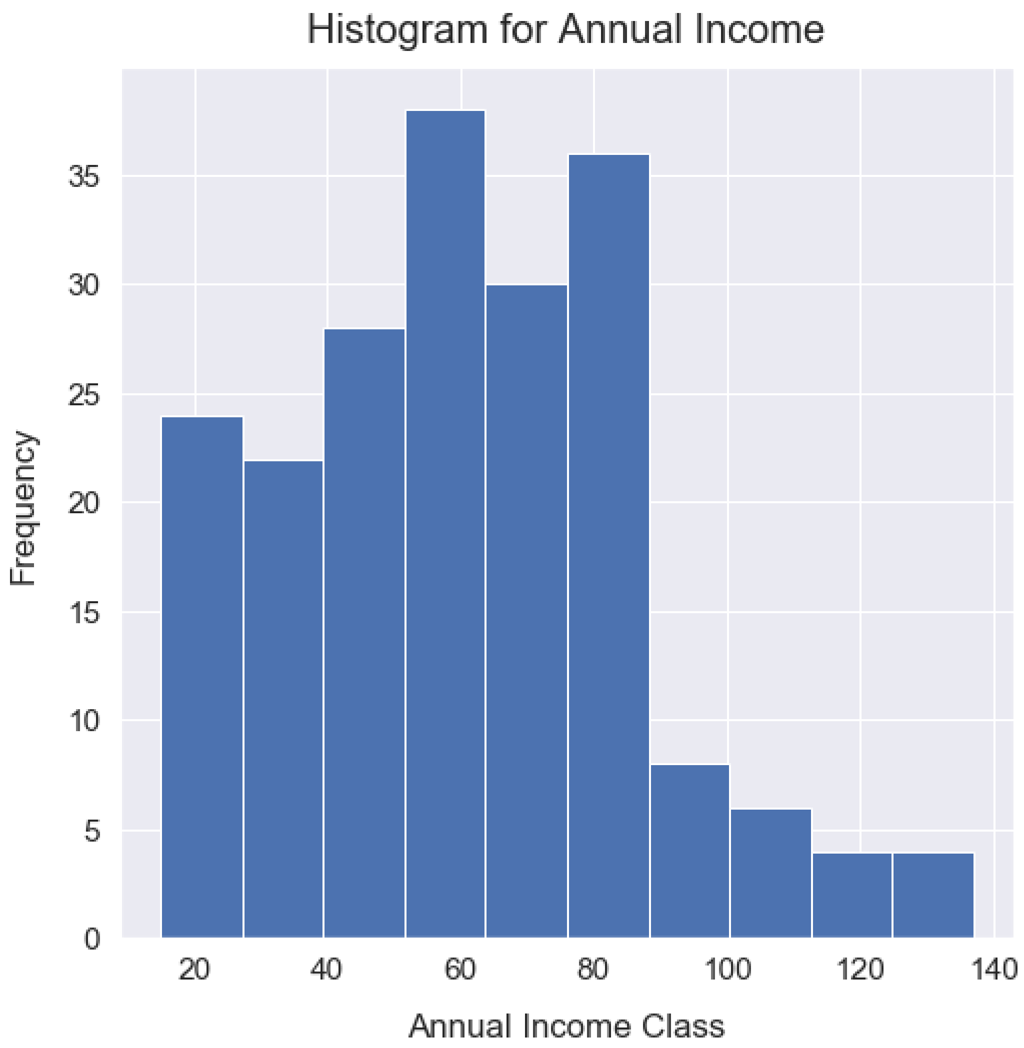
```
count    200.000000
mean      60.560000
std       26.264721
min       15.000000
25%       41.500000
50%       61.500000
75%       78.000000
max      137.000000
Name: Annual Income (k$), dtype: float64
```

In [19]:

```
sns.set(font_scale=1.4)
df['Annual Income (k$)'].plot(kind='hist', figsize=(8, 8));
plt.xlabel("Annual Income Class", labelpad=14)
plt.ylabel("Frequency", labelpad=14)
plt.title("Histogram for Annual Income", y=1.015, fontsize=20)
```

Out[19]:

```
Text(0.5, 1.015, 'Histogram for Annual Income')
```

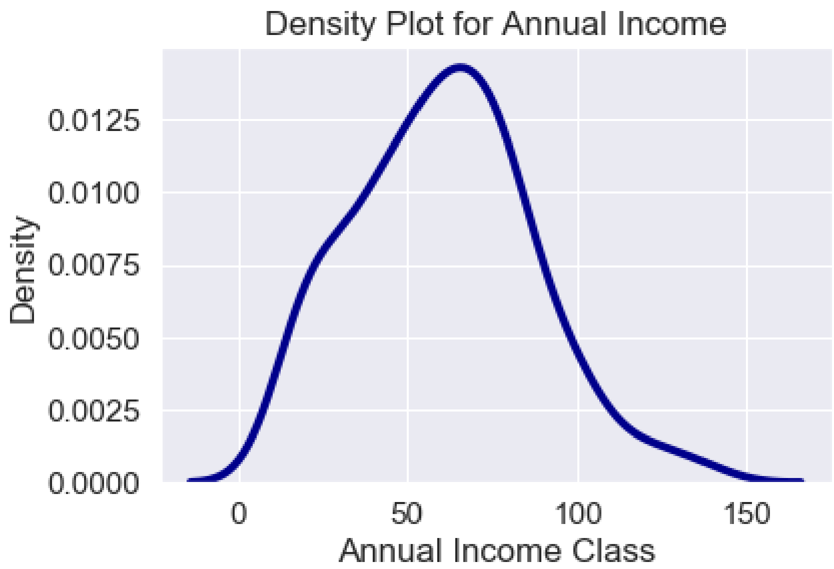


In [20]:

```
sns.distplot(df['Annual Income (k$)'], hist=False, kde=True,
             bins=10, color = 'darkblue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 4})
plt.title('Density Plot for Annual Income')
plt.xlabel('Annual Income Class')
plt.ylabel('Density')
```

Out[20]:

```
Text(0, 0.5, 'Density')
```



From the above descriptive analysis, we can conclude that the minimum annual income of the customers is 15 and the maximum income is 137. People earning an average income of 70 have the highest frequency count in our histogram distribution. The average salary of all the customers is 60.56.

Analyzing Spending Score of the Customers

In [24]:

```
df['Spending Score (1-100)'].describe()
```

Out[24]:

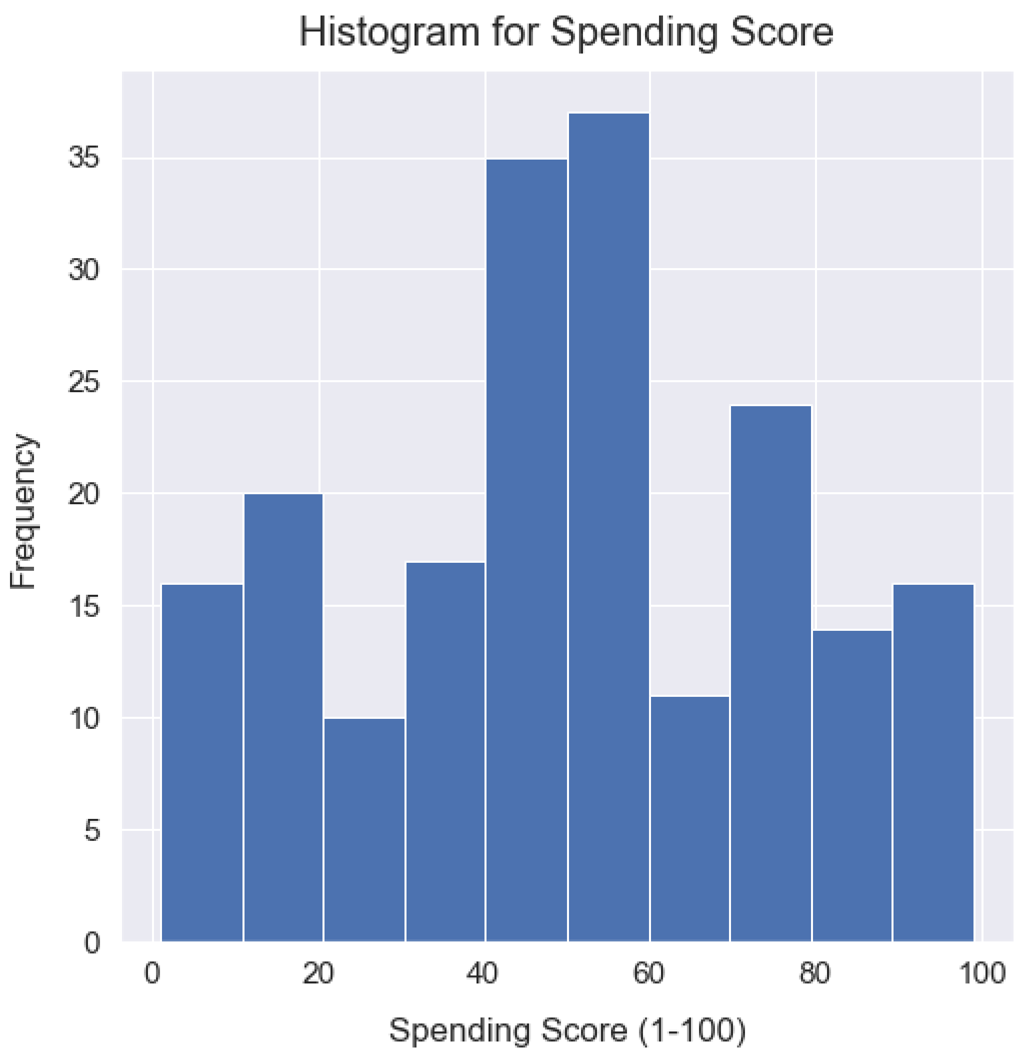
```
count    200.000000
mean      50.200000
std       25.823522
min        1.000000
25%       34.750000
50%       50.000000
75%       73.000000
max       99.000000
Name: Spending Score (1-100), dtype: float64
```

In [25]:

```
sns.set(font_scale=1.4)
df['Spending Score (1-100)'].plot(kind='hist', figsize=(8, 8));
plt.xlabel("Spending Score (1-100)", labelpad=14)
plt.ylabel("Frequency", labelpad=14)
plt.title("Histogram for Spending Score", y=1.015, fontsize=20)
```

Out[25]:

```
Text(0.5, 1.015, 'Histogram for Spending Score')
```



In [26]:

```
sns.boxplot(data=df, x='Spending Score (1-100)').set(title='BoxPlot for Descriptive Analysi
```

Out[26]:

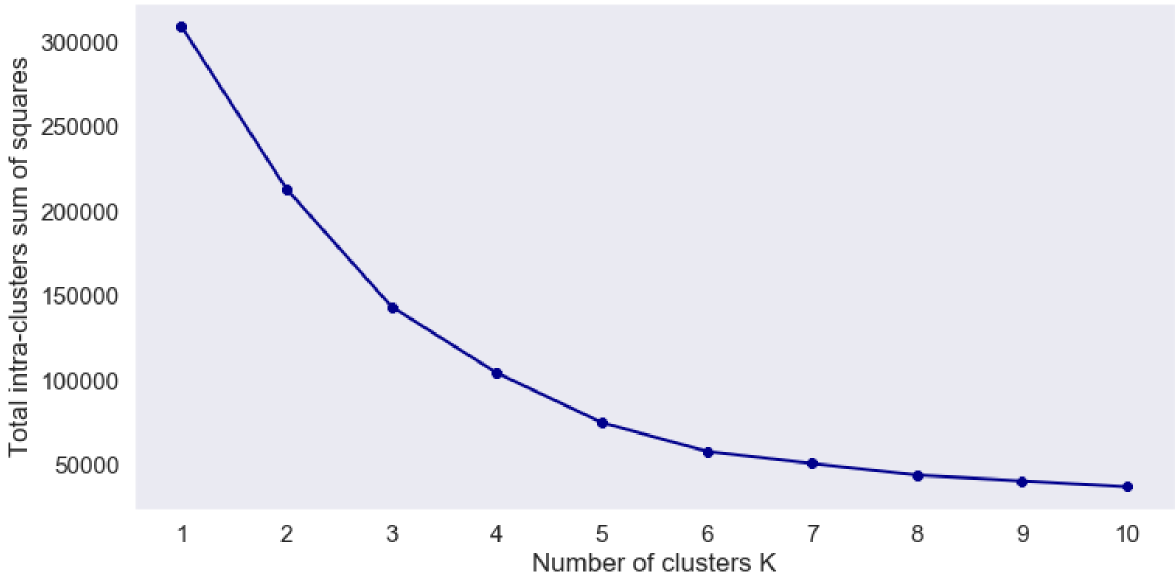
```
[Text(0.5, 1.0, 'BoxPlot for Descriptive Analysis of Spending Score')]
```

From the above descriptive analysis, we can conclude that the minimum spending score is 1, maximum is 99 and the average is 50.20. From the histogram, we can conclude that customers between class 50 and 60 have the highest spending score among all the classes.

Elbow Method

In [64]:

```
from sklearn.cluster import KMeans
wcss = []
for k in range(1,11):
    kmeans = KMeans(n_clusters=k, init="k-means++")
    kmeans.fit(df.iloc[:,2:])
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2, color="darkblue", marker ="8")
plt.xlabel("Number of clusters K")
plt.xticks(np.arange(1,11,1))
plt.ylabel("Total intra-clusters sum of squares")
plt.show()
```



The optimal K value is found to be 5 using the elbow method. Therefore we can conclude that 5 is the appropriate number of clusters since it seems to be appearing at the bend in the elbow plot.

In [42]:

```
km = KMeans(n_clusters=5)
y_predicted = km.fit_predict(df.iloc[:,2:])
y_predicted
```

Out[42]:

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

In [45]:

```
df.iloc[:,2:]
```

Out[45]:

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

200 rows × 3 columns

In [46]:

```
%matplotlib inline

from mpl_toolkits.mplot3d import Axes3D
```

In [48]:

```
km.cluster_centers_
```

Out[48]:

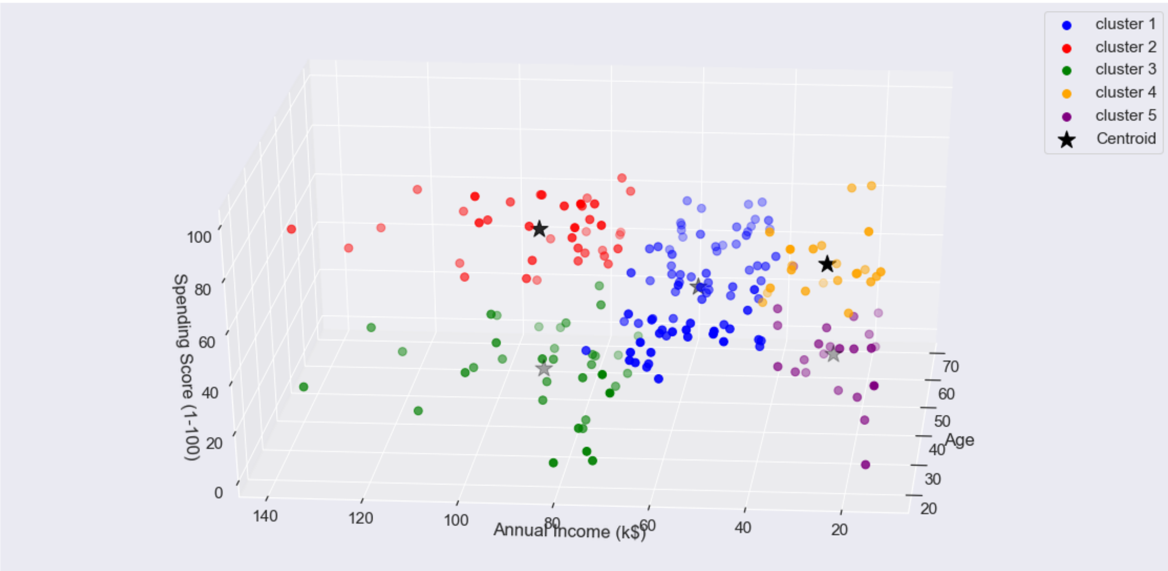
```
array([[45.2173913 , 26.30434783, 20.91304348],
       [43.08860759, 55.29113924, 49.56962025],
       [40.66666667, 87.75      , 17.58333333],
       [25.52173913, 26.30434783, 78.56521739],
       [32.69230769, 86.53846154, 82.12820513]])
```

In [50]:

```
km = KMeans(n_clusters=5)
clusters = km.fit_predict(df.iloc[:,2:])
df["label"] = clusters

fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Sco
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Sco
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Sco
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Sco
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Sco
ax.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:,1],
           km.cluster_centers_[:,2],
           s = 300, c = 'black', marker='*', label = 'Centroid')
plt.autoscale(enable=True, axis='x', tight=True)
ax.view_init(30, 185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
ax.legend()
plt.show()
```



In [ ]:

```

```