Customer Buying Behavior Analysis using K-Means Clustering Algorithm

-Sanket Argade

Installing the Libraries

In [1]:

```
# for basic mathematics operation
import numpy as np

# for dataframe manipulations
import pandas as pd

# for Data Visualizations
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
import plotly
import plotly.express as px

# For Statistics
import statistics
```

Examining Data

In [2]:

```
# importing the dataset
data = pd.read_csv('Mall_Customers.csv')
```

In [3]:

```
# lets check the shape of the dataset
print("Shape of the dataset :", data.shape)
```

```
Shape of the dataset : (200, 5)
```

In [4]:

```
# lets check the head of the data
data.head()
```

Out[4]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	34	Male	18.0	33	92
1	66	Male	18.0	48	59
2	92	Male	18.0	59	41
3	115	Female	18.0	65	48
4	1	Male	19.0	15	39

In [5]:

```
# lets check the tail of the data
data.tail()
```

Out[5]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
195	91	Female	68.0	59	55
196	109	Male	68.0	63	43
197	58	Male	69.0	44	46
198	61	Male	70.0	46	56
199	71	Male	70.0	49	55

Descriptive Statistics

In [6]:

```
# describing the data
data.describe()
```

Out[6]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	196.000000	200.000000	200.000000
mean	100.500000	38.734694	60.560000	50.200000
std	57.879185	13.964094	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.000000	41.500000	34.750000
50%	100.500000	35.500000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

In [7]:

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

Out[7]:

```
CustomerID                0
Gender                    3
Age                       4
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

> Two types of variable missing
>
> > Gender(Categorical varibles) used Mode to replace null values.

> > Age() used Mean to replace null values.

In [8]:

```
# fill mode value of age at the place of missing values(Female)
data["Gender"].fillna(statistics.mode(data["Gender"]), inplace = True)
```

In [9]:

```
# fill Mean value of age at the place of missing values(38)
data["Age"].fillna(np.mean(data["Age"]), inplace = True)
```

In [10]:

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

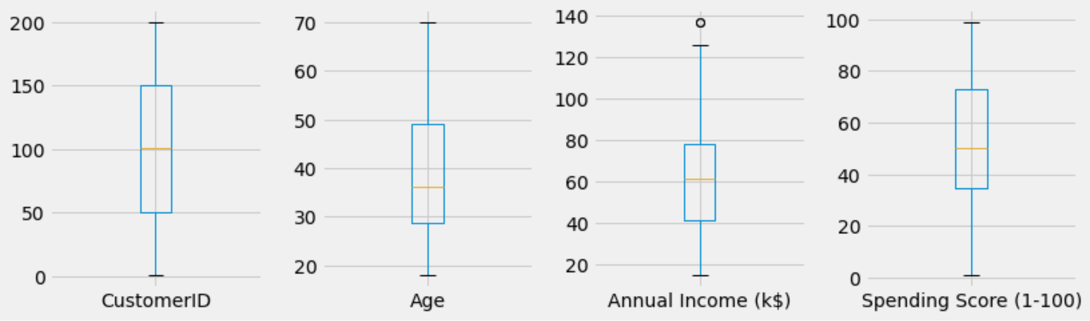
Out[10]:

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

> **Outliers**

In [11]:

```
# Finding outliers using Box Plot
plt.figure(figsize=(20,10))
for i, col in enumerate(['CustomerID', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)'], start=1):
    plt.subplot(3,7,i)
    data.boxplot(col)
    plt.tight_layout()
```



In [12]:

```
# Outliers
data[(data["Annual Income (k$)"]>130)]
```

Out[12]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
61	200	Male	30.0	137	83
80	199	Male	32.0	137	18

In [13]:

```
#delete Outliers
data.drop(range(198,200), axis=0, inplace=True)
```

In [14]:

```
data.shape
```

Out[14]:

```
(198, 5)
```

Correlation

In [15]:

```
data.corr().style.background_gradient(axis=None)
```

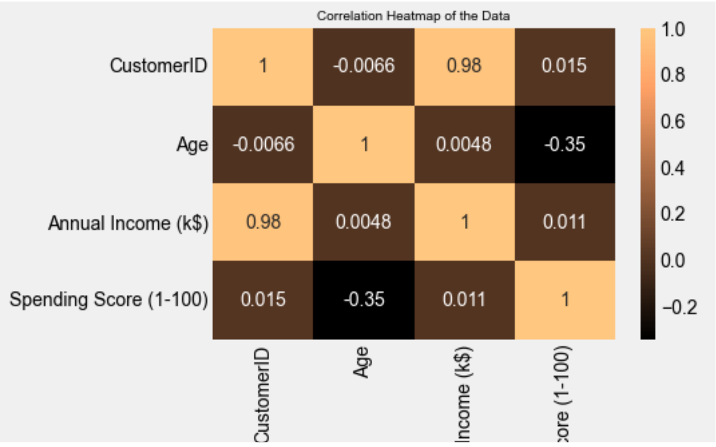
Out[15]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1.000000	-0.006636	0.977533	0.015125
Age	-0.006636	1.000000	0.004843	-0.346249
Annual Income (k$)	0.977533	0.004843	1.000000	0.010966
Spending Score (1-100)	0.015125	-0.346249	0.010966	1.000000

In [16]:

```
# lets check the Correlation Heat Map of the Data

sns.heatmap(data.corr(),annot = True,  cmap = 'copper')
plt.title('Correlation Heatmap of the Data', fontsize = 10)
sns.set(rc={'figure.figsize': (10, 9)})
plt.show()
```



The Above Graph for Showing the correlation between the different attributes of the Mall Customer Segementation Dataset, This Heat map reflects the most correlated features with dark Orange Color and least correlated features with yellow color.

> We can clearly see that these attributes do not have good correlation among them, that's why we will proceed with all of the features.

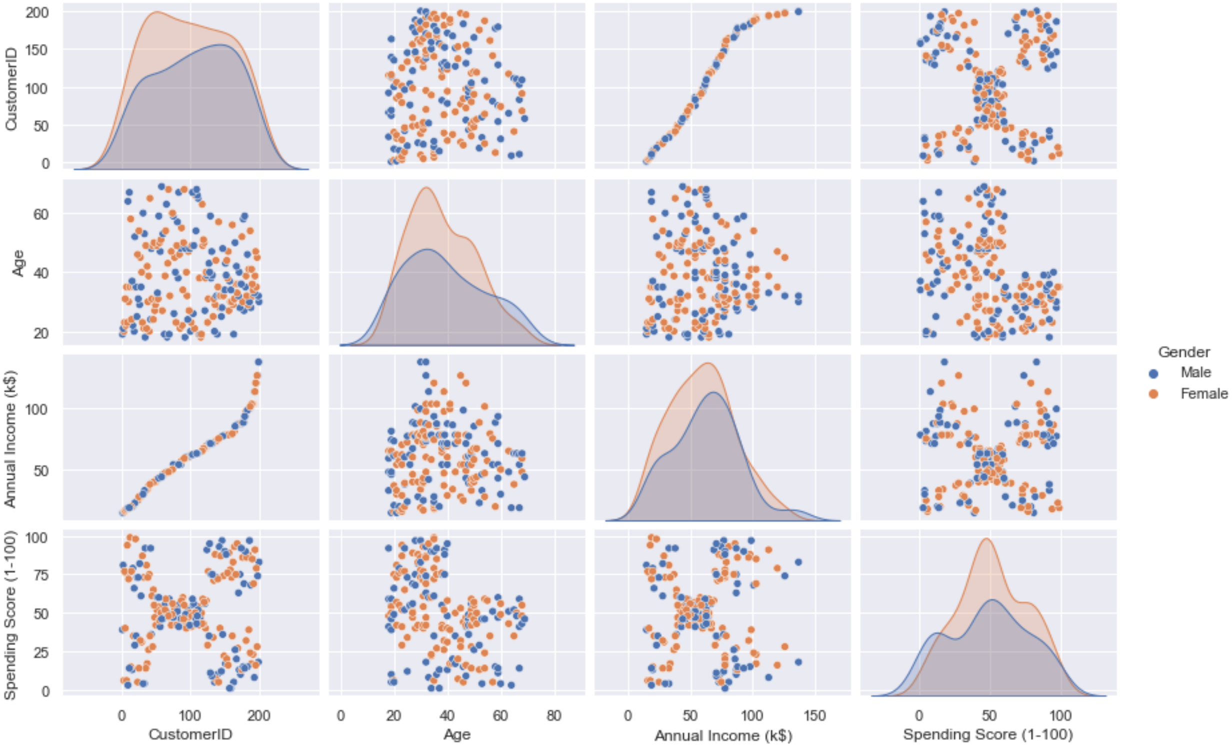
Data visualization

In [17]:

```
sns.pairplot(data, hue = "Gender", height=2, aspect=1.5)
```

Out[17]:

```
<seaborn.axisgrid.PairGrid at 0x1f2ba2643a0>
```



In [18]:

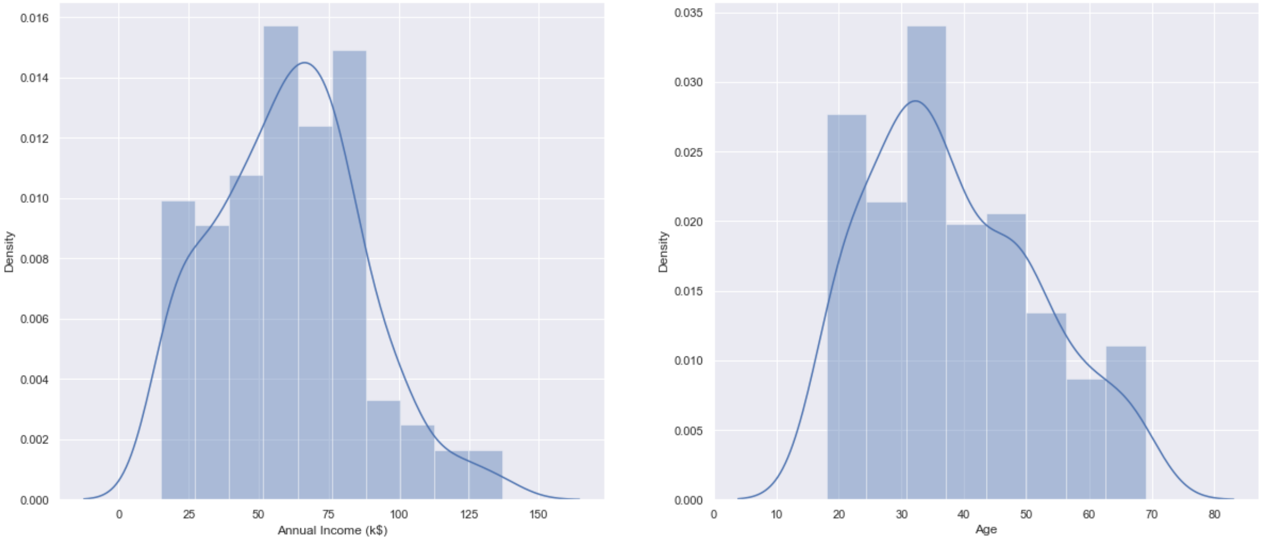
```
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (18, 8)

plt.subplot(1, 2, 1)
sns.distplot(data["Annual Income (k$)"], kde = True)

plt.subplot(1, 2, 2)
sns.distplot(data["Age"], kde = True)
```

Out[18]:

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



Here, In the above Plots we can see the Distribution pattern of Annual Income and Age, By looking at the plots,

> we can infer one thing that There are few people who earn more than 100 US Dollars. Most of the people have an earning of around 50-75 US Dollars. Also, we can say that the least Income is around 20 US Dollars.

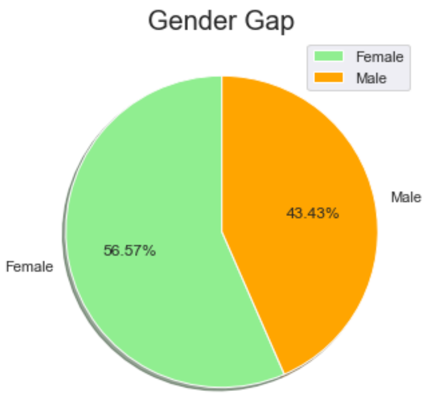
Taking inferences about the Customers.

> The most regular customers for the Mall has age around 30-35 years of age. Whereas the the senior citizens age group is the least frequent visitor in the Mall. Youngsters are lesser in umber as compared to the Middle aged people.

In [19]:

```python
labels = ['Female', 'Male']
size = data['Gender'].value_counts()
colors = ['lightgreen', 'orange']
explode = [0, 0.001]

plt.rcParams['figure.figsize'] = (5, 5)
plt.pie(size, colors = colors, explode = explode, labels = labels, shadow = True, startangle = 90, autopct = '%.2f%%')
plt.title('Gender Gap', fontsize = 20)
plt.axis('off')
plt.legend()
plt.show()
```



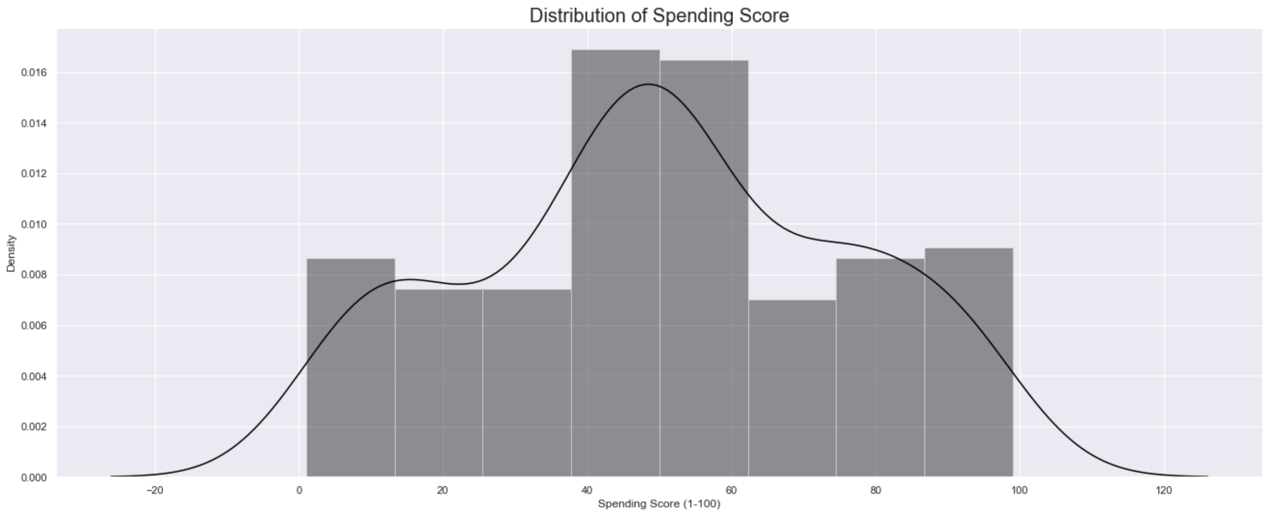
By looking at the above pie chart which explains about the distribution of Gender in the Mall

> Interestingly, The Females are in the lead with a share of 56% whereas the Males have a share of 44%, that's a huge gap specially when the population of Males is comparatively higher than Females.

In [20]:

```python
# lets check the distribution of Spending Score

plt.rcParams['figure.figsize'] = (20, 8)
sns.distplot(data['Spending Score (1-100)'], color = 'black')
plt.title('Distribution of Spending Score', fontsize = 20)
plt.show()
```



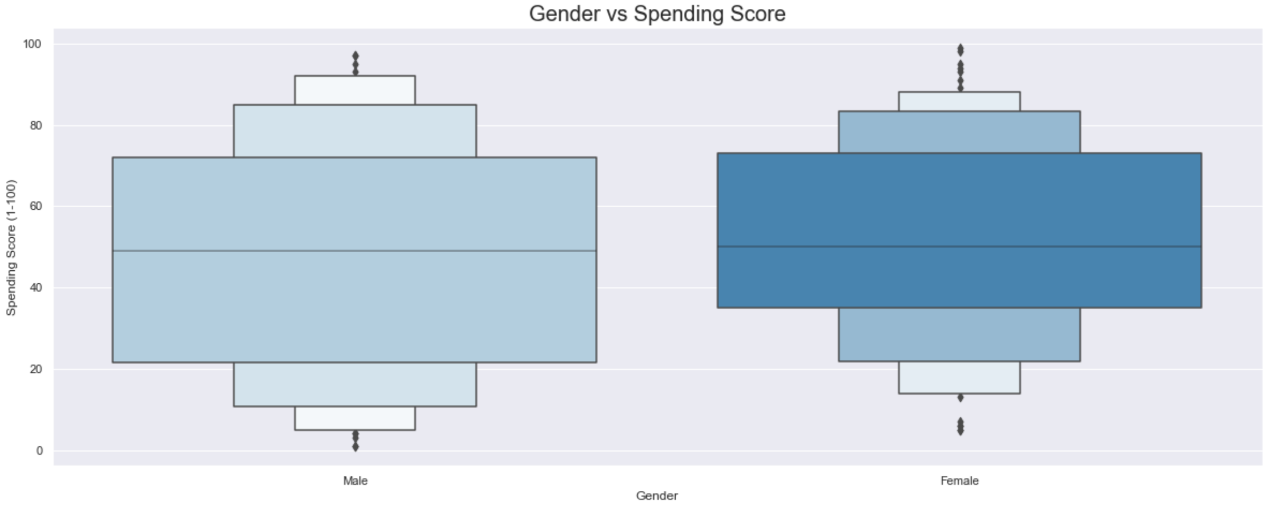
This is the Most Important Chart in the perspective of Mall, as It is very Important to have some intuition and idea about the Spending Score of the Customers Visiting the Mall.

> On a general level, we may conclude that most of the Customers have their Spending Score in the range of 40-60. Interesting there are customers having I spending score also, and 99 Spending score also, Which shows that the mall caters to the variety of Customers with Varying needs and requirements available in the Mall.

In [21]:

```
#  Gender vs Spendscore

plt.rcParams['figure.figsize'] = (18, 7)
sns.boxenplot(data['Gender'], data['Spending Score (1-100)'], palette = 'Blues')
plt.title('Gender vs Spending Score', fontsize = 20)
plt.show()
```



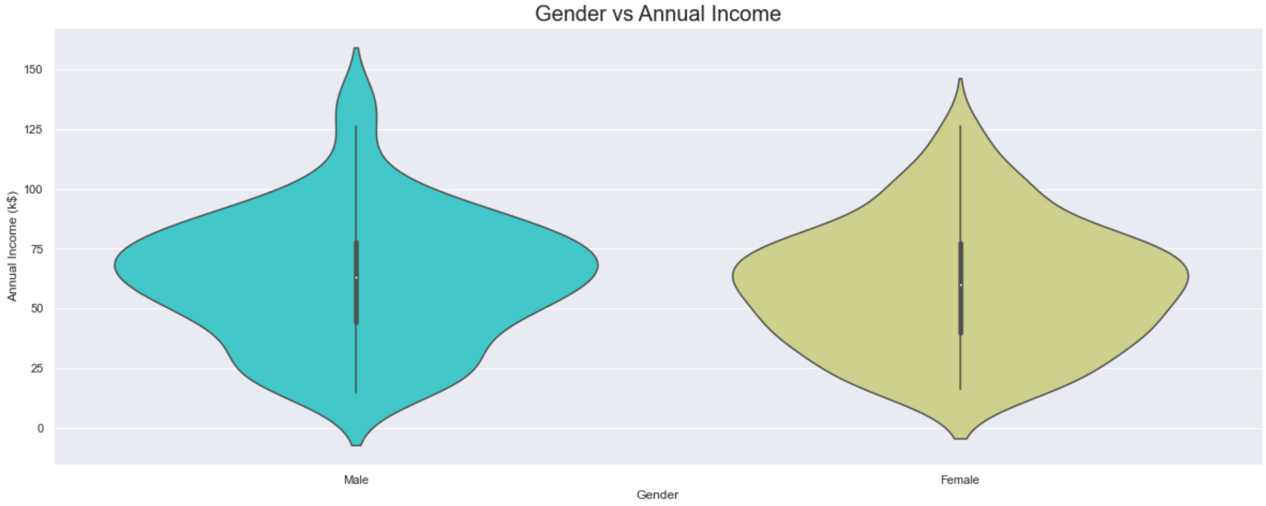
Bi-variate Analysis between Gender and Spending Score,

> It is clearly visible that the most of the males have a Spending Score of around 25k US Dollars to 70k US Dollars whereas the Females have a spending score of around 35k US Dollars to 75k US Dollars. which again points to the fact that women are Shopping Leaders.

In [22]:

```
## Gender vs Annual Income

plt.rcParams['figure.figsize'] = (18, 7)
sns.violinplot(data['Gender'], data['Annual Income (k$)'], palette = 'rainbow')
plt.title('Gender vs Annual Income', fontsize = 20)
plt.show()
```



Again a Bivariate Analysis between the Gender and the Annual Income, to better visualize the Income of the different Genders.

> There are more number of males who get paid more than females. But, The number of males and females are equal in number when it comes to low annual income.

Clustering Analysis

In [23]:

```
# we want to perform clusters of Customers who share similar behaviour for that lets select the columns
# Spending score, Annual Income, and Age
import warnings
warnings.filterwarnings('ignore')

# Lets select the Spending score, Annual Income and Age Columns from the Data
x = data.loc[:, ["Age",'Spending Score (1-100)', 'Annual Income (k$)']].values

print(x.shape)
```

```
(198, 3)
```

In [24]:

```
# lets also check the data, which we are going to use for the clustering analysis
x_data  = pd.DataFrame(x)
x_data.head()
# where o->Age, 1->Spending Score(1-100), 2->Annual Income
```

Out[24]:

	0	1	2
0	18.0	92.0	33.0
1	18.0	59.0	48.0
2	18.0	41.0	59.0
3	18.0	48.0	65.0
4	19.0	39.0	15.0

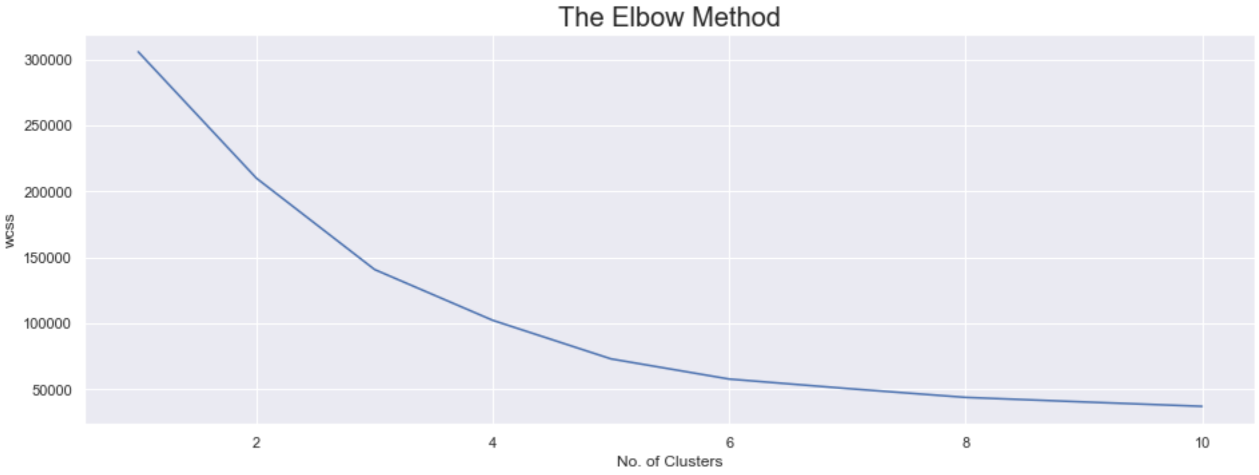
Kmeans Algorithm

The Elbow Method to find the No. of Optimal Clusters

In [25]:

```
from sklearn.cluster import KMeans
plt.figure(figsize=(14,5))
x= data[["Age", "Annual Income (k$)", "Spending Score (1-100)"]]
wcss = []
for i in range(1, 11):
    km = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    km.fit(x)
    wcss.append(km.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method', fontsize = 20)
plt.xlabel('No. of Clusters')
plt.ylabel('wcss')
plt.show()
```



In [26]:

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
print("Imported")
scorelist = []
for i in range (2,10):
    model = KMeans(n_clusters = i)
    resultlist = model.fit_predict(x)
    score = silhouette_score(x,resultlist)
    scorelist.append(score)
print(scorelist)
K_number = scorelist.index(max(scorelist))+2
print("Final K: ",K_number)
```

```
Imported
[0.292807793521909, 0.3855737839929377, 0.40597561992379555, 0.4478152315388148, 0.44500397175950607, 0.4339292433751572,
0.4218740521816975, 0.4038616214100085]
Final K:  5
```

In [27]:

```
finalmodel = KMeans(n_clusters = K_number)
resultlist = finalmodel.fit_predict(x)
x = x.assign(cluster = resultlist)
print(x)
```

```
      Age  Annual Income (k$)  Spending Score (1-100)  cluster
0    18.0                  33                      92        1
1    18.0                  48                      59        4
2    18.0                  59                      41        4
3    18.0                  65                      48        4
4    19.0                  15                      39        2
..    ...                 ...                     ...      ...
193  67.0                  62                      59        4
194  68.0                  48                      48        4
195  68.0                  59                      55        4
196  68.0                  63                      43        4
197  69.0                  44                      46        4

[198 rows x 4 columns]
```

In [28]:

```
df= x.groupby(x["cluster"]).mean()
df
```

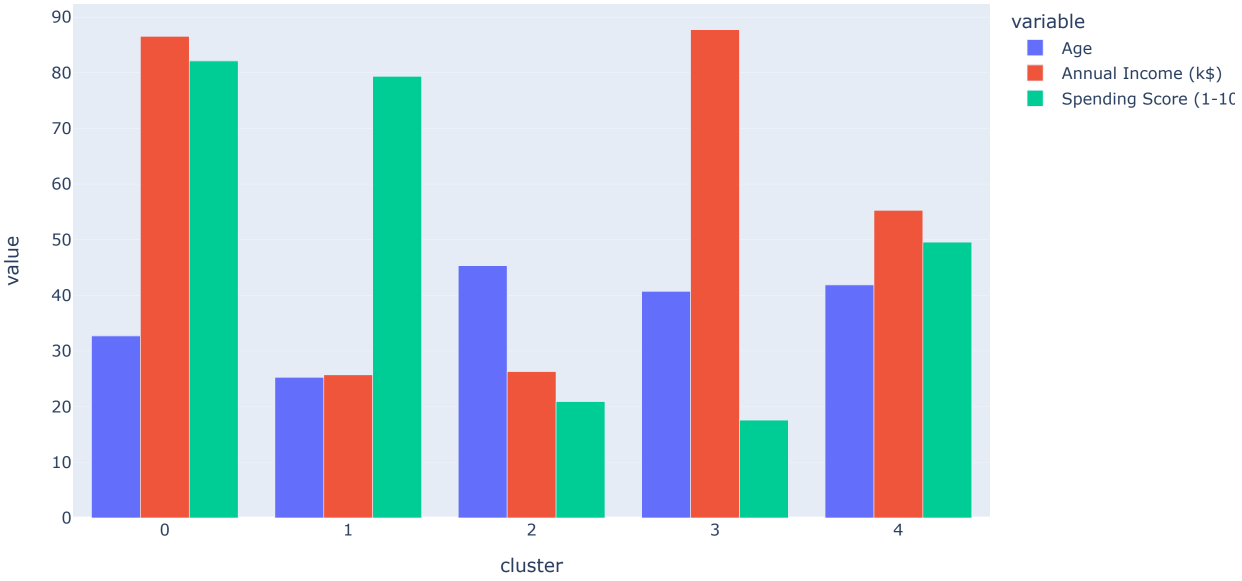
Out[28]:

	Age	Annual Income (k$)	Spending Score (1-100)
cluster			
0	32.711146	86.538462	82.128205
1	25.272727	25.727273	79.363636
2	45.336291	26.304348	20.913043
3	40.714853	87.750000	17.583333
4	41.881214	55.282051	49.564103

Visualizaing the Clusters

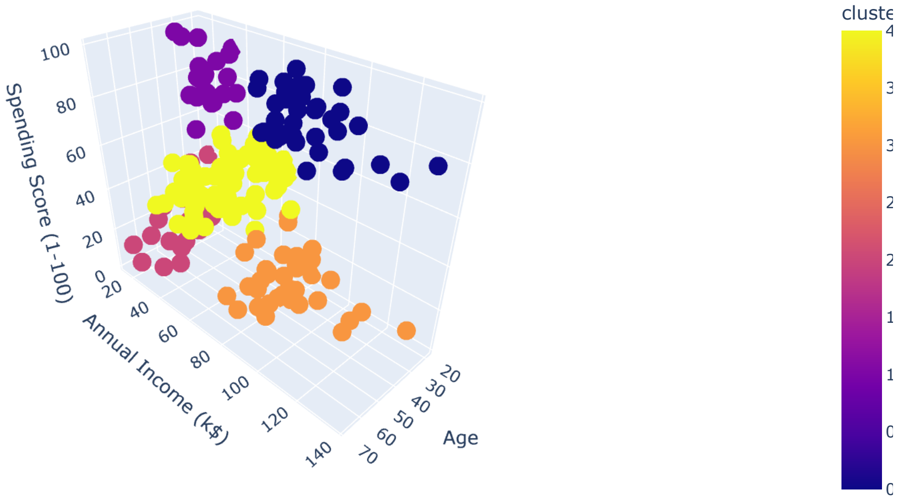
In [31]:

```
fig= px.bar(
    df,barmode="group")
fig.show("notebook")
```



In [32]:

```
import plotly.express as px
fig = px.scatter_3d(x, x='Age', y='Annual Income (k$)', z='Spending Score (1-100)',color='cluster')
fig.show()
```



This Clustering Analysis gives us a very clear insight about the different segments of the customers in the Mall. There are clearly Five segments of Customers namely as there color code.

> pink, yellow, blue, orange, and violet Based on their Annual Income, Spending Score and Age which are reportedly the best factors/attributes to determine the segments of a customer in a Mall.

Outcome

- It will help to Prioritized advertising boardings and digital signages.

- It will help to developing in-mall infrastructures.

- To Prioritized the customer segment from cluster number 4 and 2.

- And to discuss different marketing strategies and policies to attract cutomers from cluster number 0, 1, and 3.