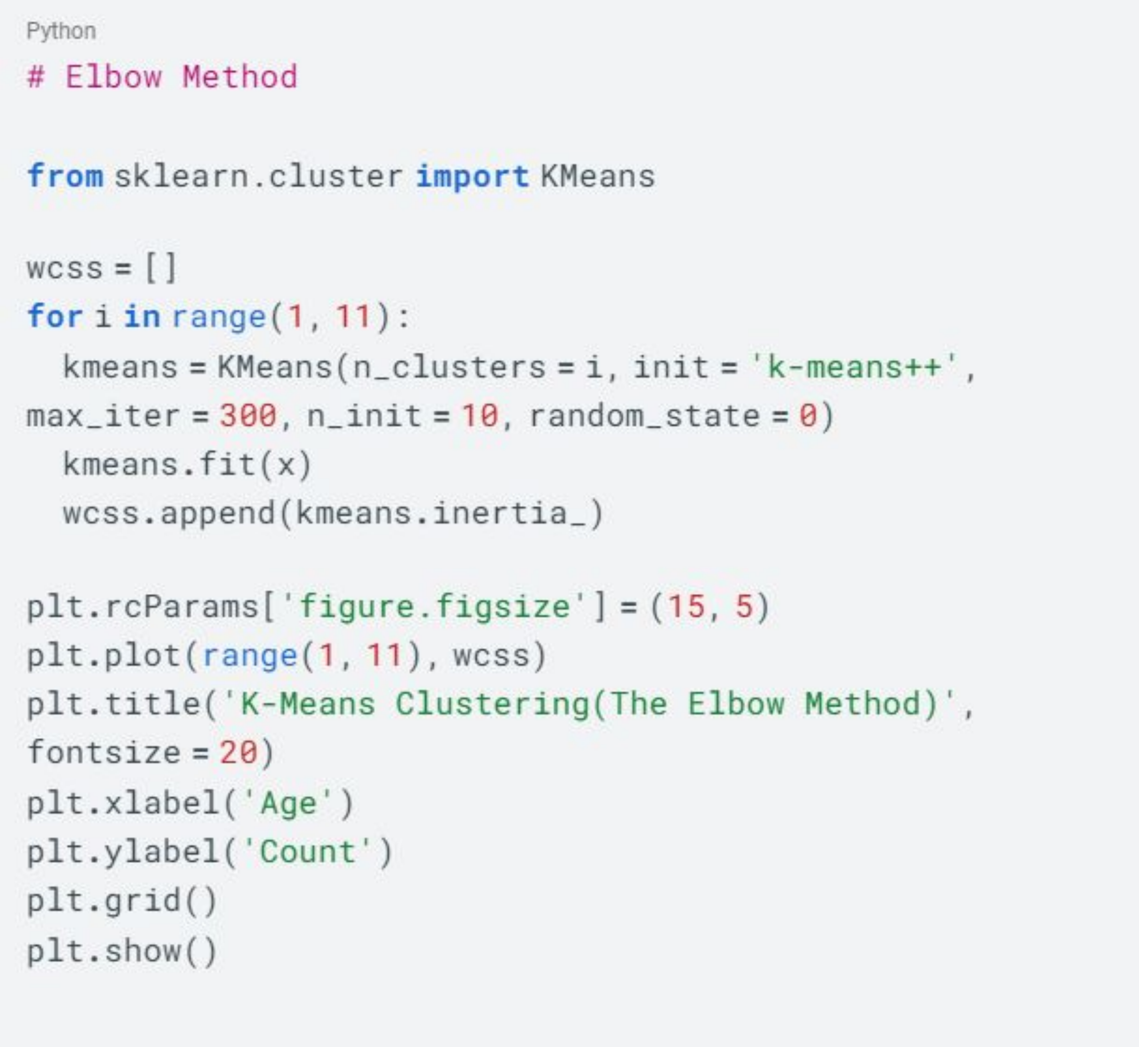
Welcome back!

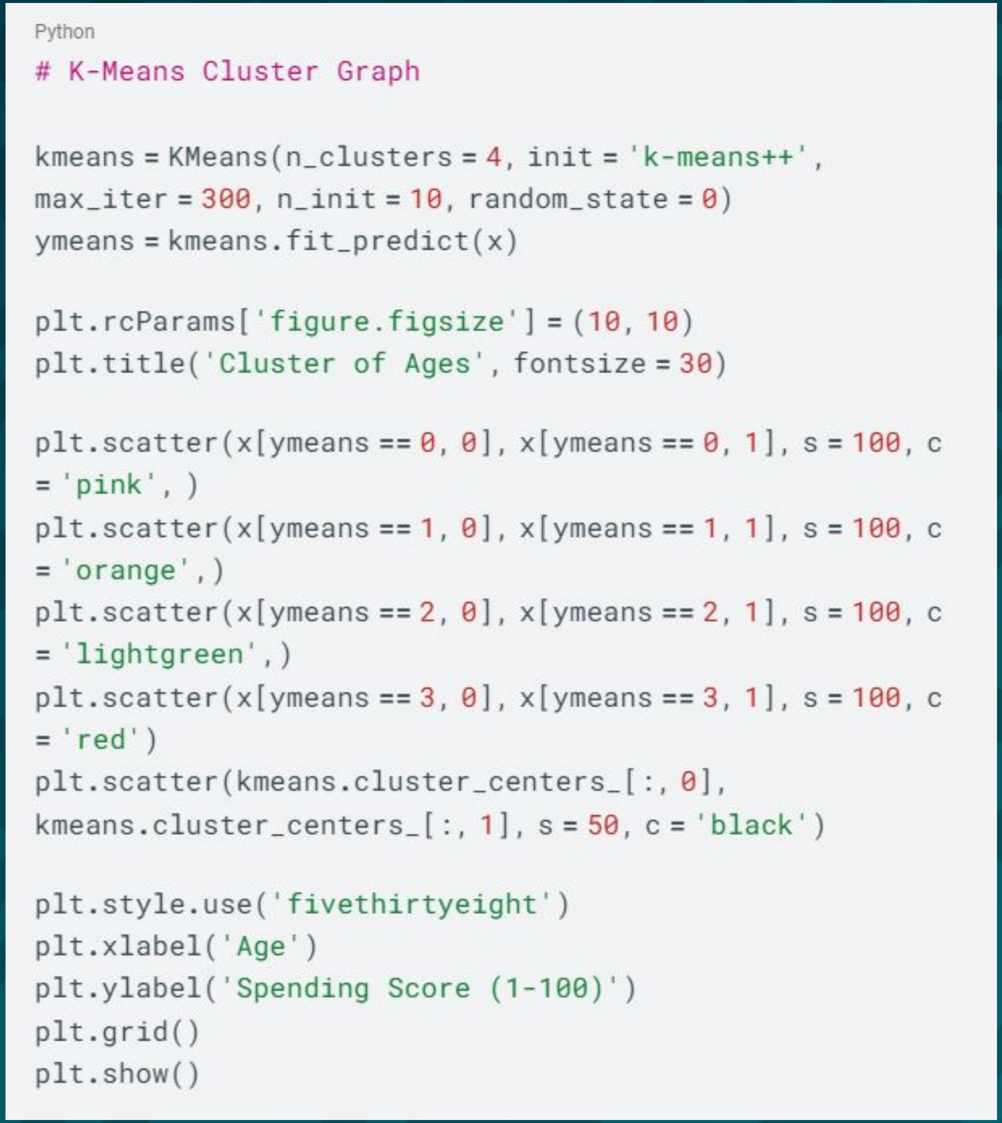
- Last week, we covered the K-Means clustering and the Elbow Method. For this process, we first use the Elbow method to determine the optimal number of centroids and then create a K-means graph using this number.
- The following is the key for the homework:



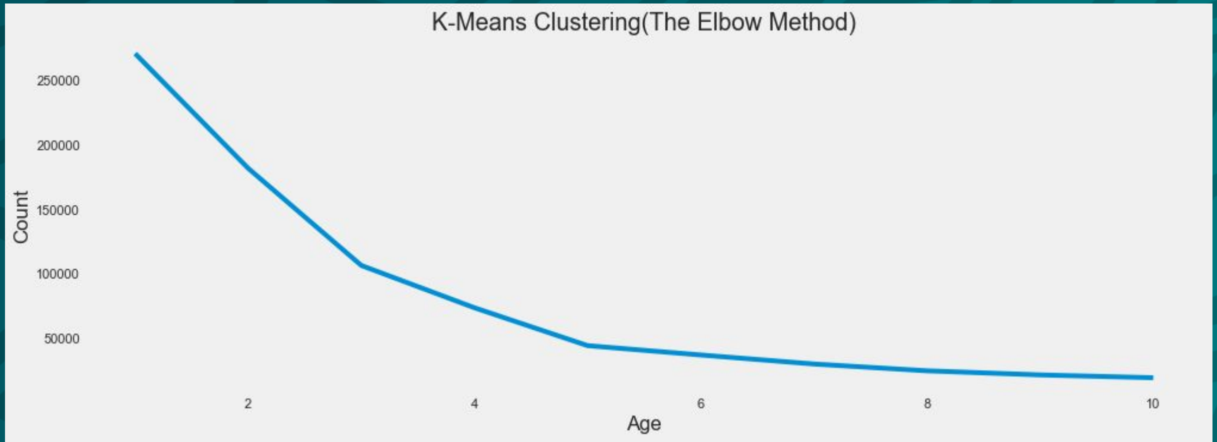
Python

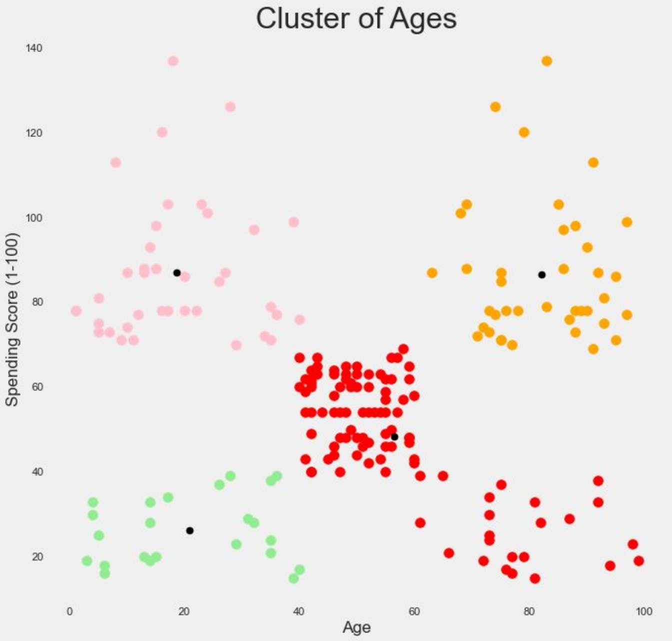
```python
# Elbow Method

from sklearn.cluster import KMeans

wcss = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters = i, init = 'k-means++',
max_iter = 300, n_init = 10, random_state = 0)
  kmeans.fit(x)
  wcss.append(kmeans.inertia_)

plt.rcParams['figure.figsize'] = (15, 5)
plt.plot(range(1, 11), wcss)
plt.title('K-Means Clustering(The Elbow Method)',
fontsize = 20)
plt.xlabel('Age')
plt.ylabel('Count')
plt.grid()
plt.show()
```

Part 2



```
Python
# K-Means Cluster Graph

kmeans = KMeans(n_clusters = 4, init = 'k-means++',
max_iter = 300, n_init = 10, random_state = 0)
ymeans = kmeans.fit_predict(x)

plt.rcParams['figure.figsize'] = (10, 10)
plt.title('Cluster of Ages', fontsize = 30)

plt.scatter(x[ymeans == 0, 0], x[ymeans == 0, 1], s = 100, c
= 'pink', )
plt.scatter(x[ymeans == 1, 0], x[ymeans == 1, 1], s = 100, c
= 'orange',)
plt.scatter(x[ymeans == 2, 0], x[ymeans == 2, 1], s = 100, c
= 'lightgreen',)
plt.scatter(x[ymeans == 3, 0], x[ymeans == 3, 1], s = 100, c
= 'red')
plt.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1], s = 50, c = 'black')

plt.style.use('fivethirtyeight')
plt.xlabel('Age')
plt.ylabel('Spending Score (1-100)')
plt.grid()
plt.show()
```



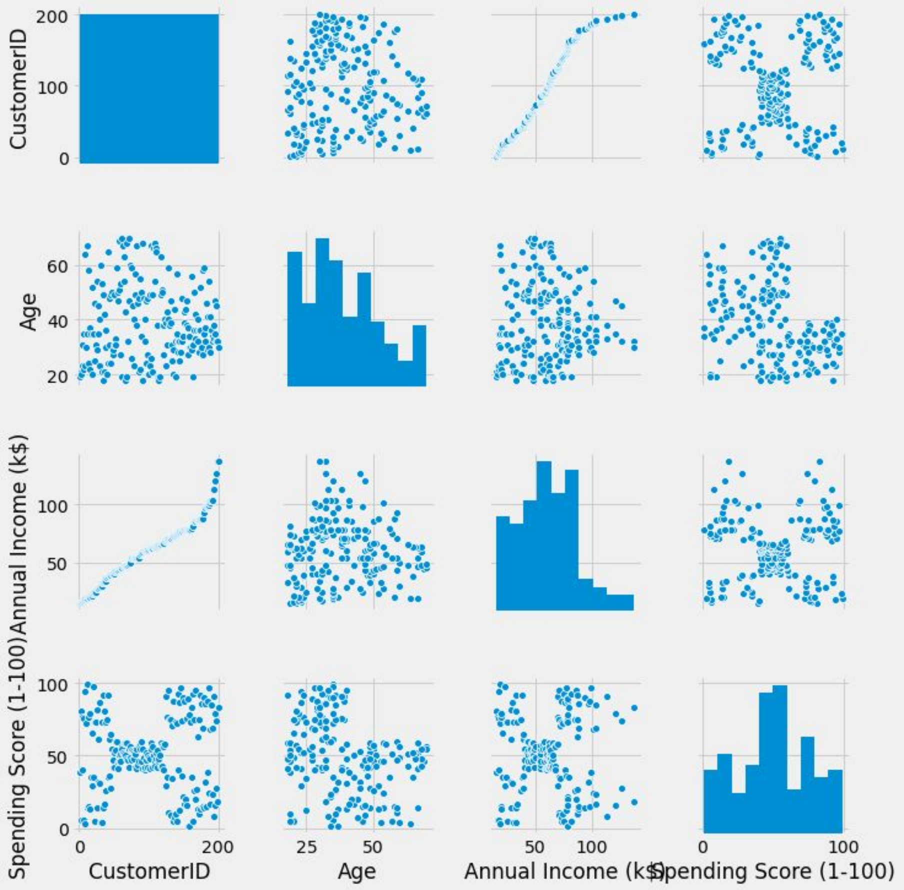


Visualization Techniques

Looking back at previous weeks, we have used various graphs and clustering techniques. Before we introduce the final project report, let's do a brief review of the different visualization techniques we used.

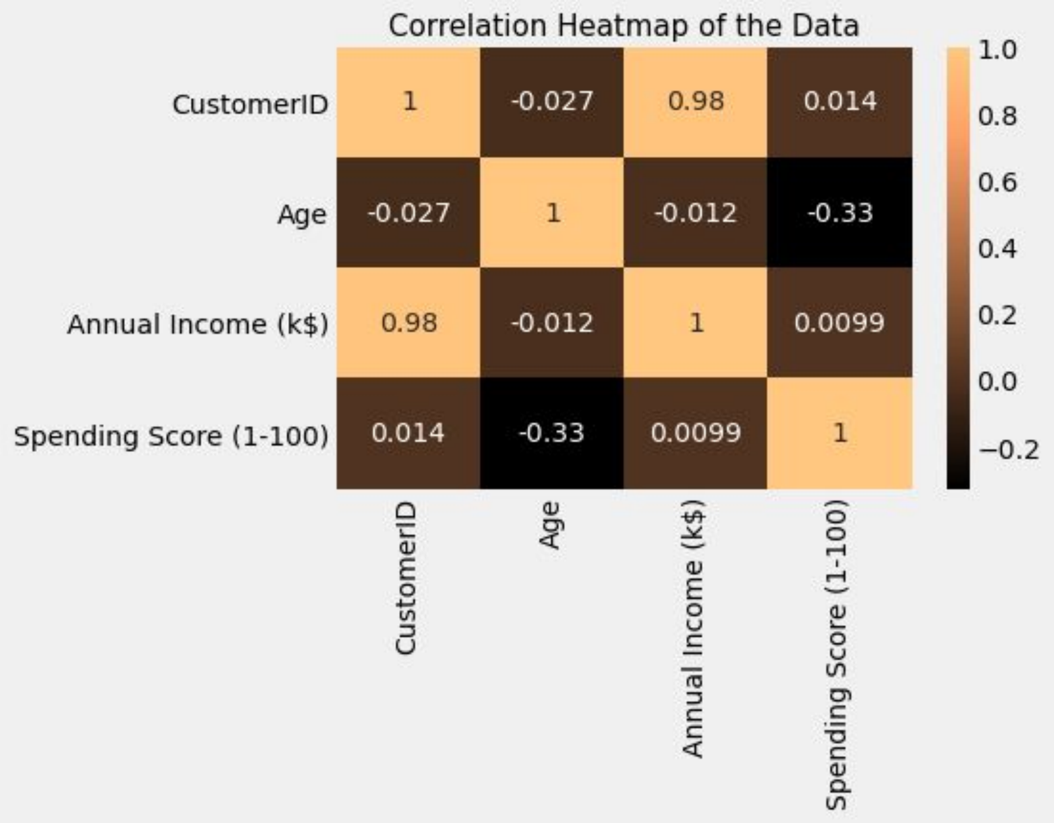
Pairplot

The Pairplot is one of the many Seaborn graphs we have explored, which is a graph that contains a grid of scatterplots comparing every numerical variable in the dataset against each other. This is great for looking at the specific correlation between each variable. Specifically, looking at if the correlation between certain variables is negative or positive, or if it is linear. Across the diagonal, we can see the KDE (Kernel Density Estimate) plots which show the distribution of each variable.



Heatmap

The Heatmap is another graph from the Seaborn package, which has a grid format similar to the Pairplot graph. However, this graph displays the correlation coefficient when comparing each set of variables. These numbers indicate the strength and direction of the linear relationship between variables. A negative value indicates a negative correlation, a positive value indicates a positive correlation, and a value close to zero indicates little to no correlation. The number also indicates the magnitude of the correlation on a scale of -1 to 1. This is a graph that is useful to quickly get a quantifiable correlation between variables.

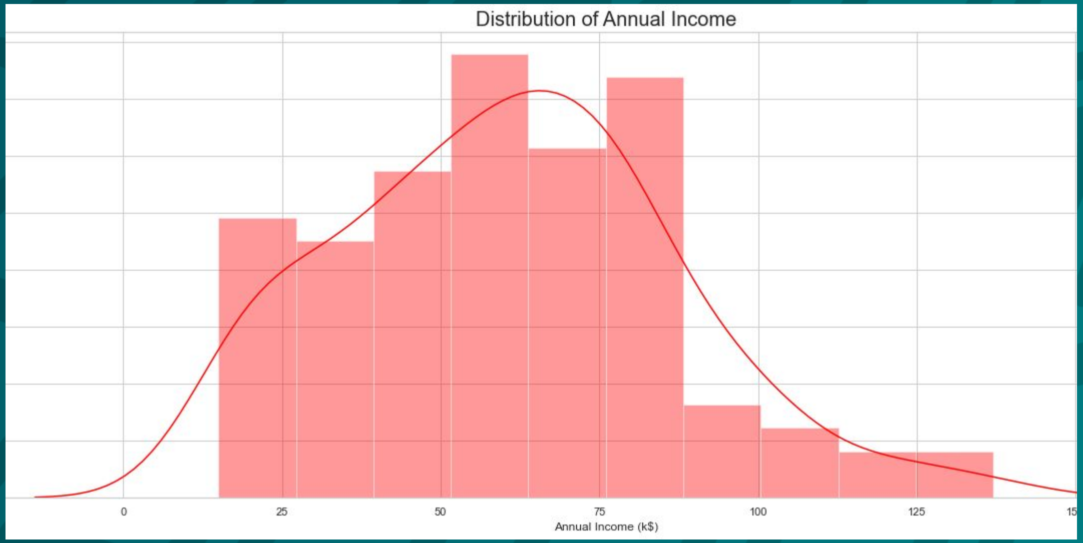


Correlation Heatmap of the Data

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1	-0.027	0.98	0.014
Age	-0.027	1	-0.012	-0.33
Annual Income (k$)	0.98	-0.012	1	0.0099
Spending Score (1-100)	0.014	-0.33	0.0099	1

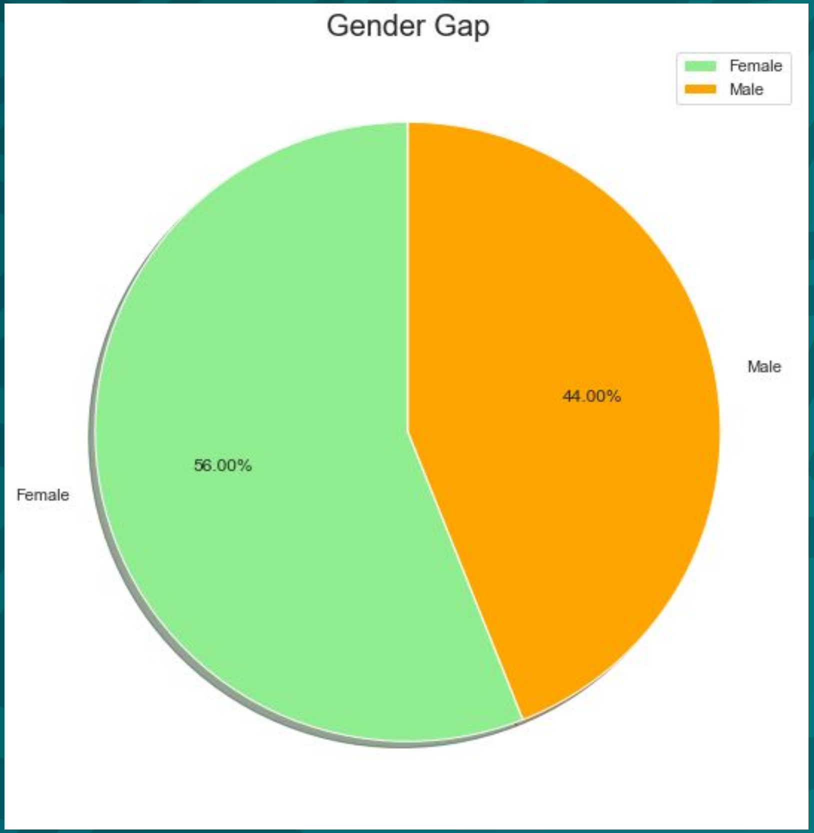
Distribution Plot

The Distribution plot is a graph that has both a Histogram component and a smoothened KDE curve. The Histogram portrays the data distribution by dividing the data range into numerous bins, with each bar indicating the count of data points falling into a specific bin’s range. The KDE line also does show the distribution of the data, however in a continuous manner which is helpful for datasets with few bars.



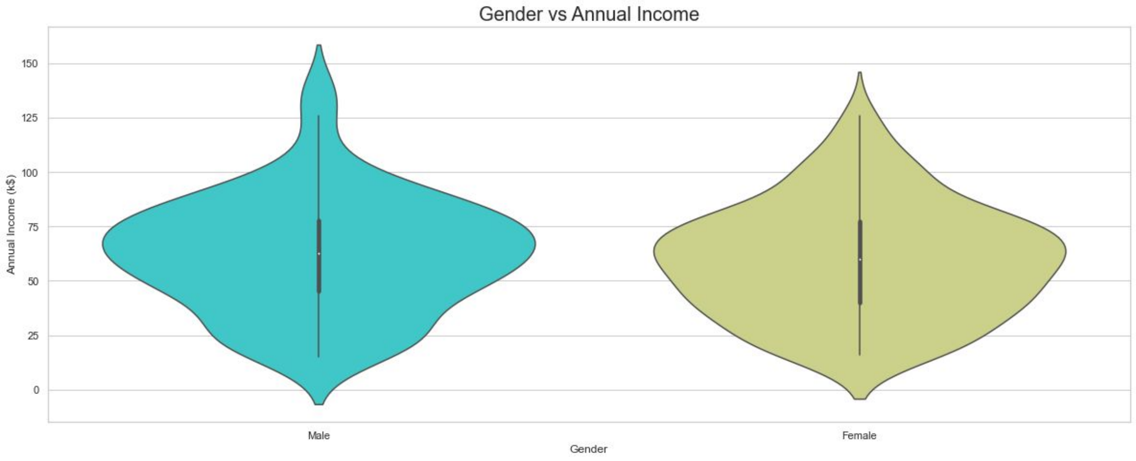
Pie Chart

The Pie Chart is another fairly common graph that illustrates the numerical proportion within data, presenting information in an easily digestible manner. It divides the whole into different slices, each slice's size directly corresponding to the proportion it signifies.



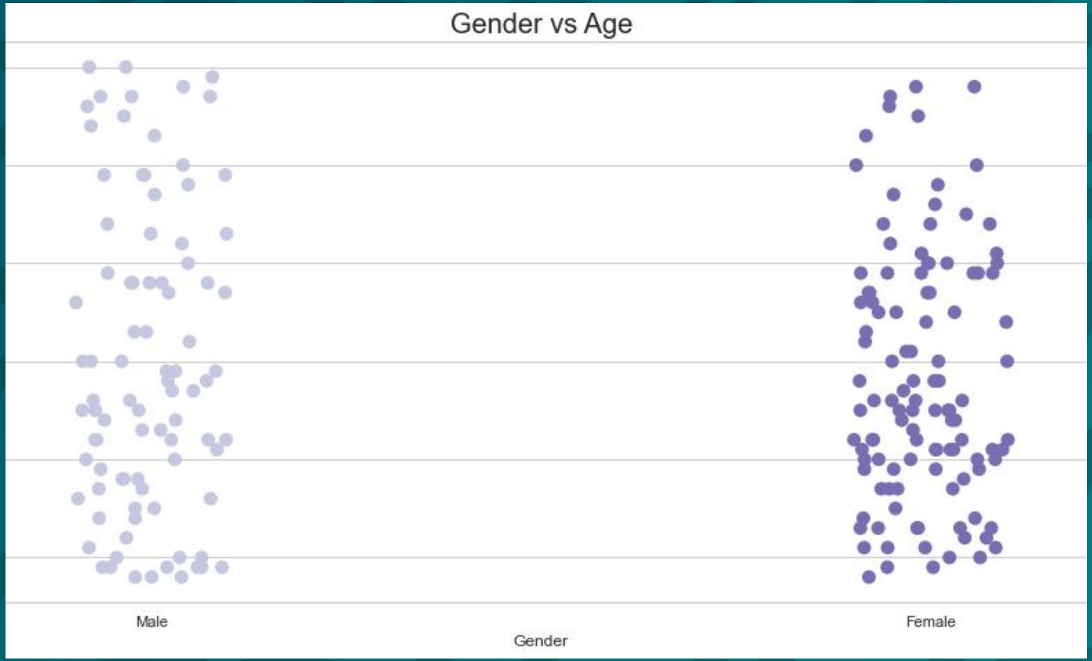
Violin Plot

The Violin plot combines elements of a box plot (highlights quartiles and potential outliers) and KDA plot to depict the distribution of numerical data within categories, providing insight into data density and central tendencies.



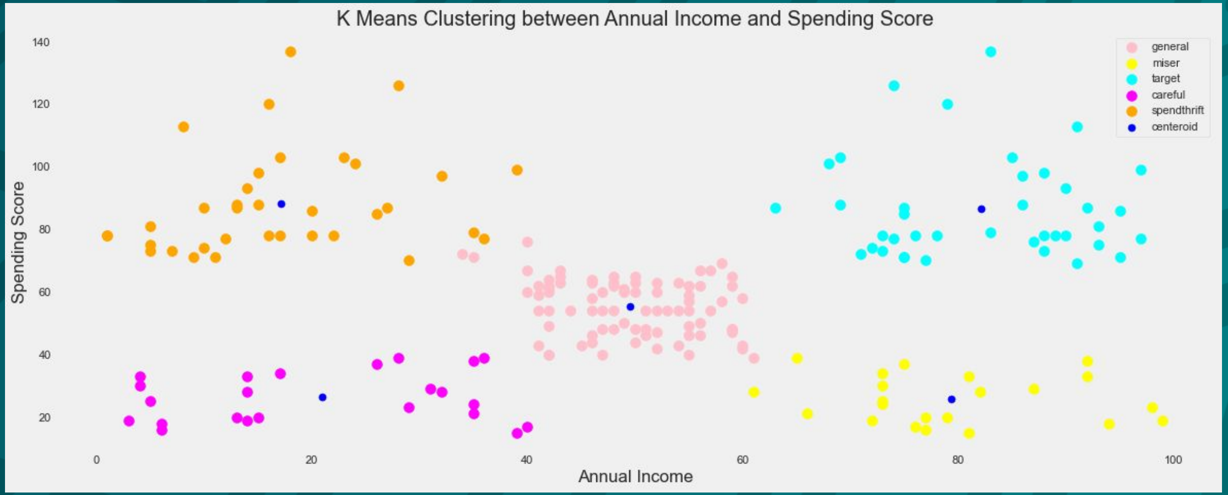
Violin Plot

The Strip Plot is used to display individual data points within a categorical variable. Each data point is represented as a dot along a single axis, making it easy to visualize the distribution and density of data in distinct categories. Strip plots help identify the spread and clustering of data points within these categories, making it easy to explore patterns.



KMeans Clustering (Elbow Method)

The KMeans Clustering graph uses the Elbow Method to first determine the optimal number of centroids. By iterating through a certain number of centroids and storing the inertia value of each iteration, we can then graph this and use this graph as an indicator for the optimal number of centroids. After this, data points are assigned to the nearest cluster centroid. Then, new centroids are computed as the average of data points within each cluster. Finally, the new centroid position is set to the center of each cluster. This process iteratively continues until no further changes are made to datapoints or centroids.



Thank You!