Task 4:

Mall Customers Using Python

Crafted By:

Vidhi Bhutia

```python
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

```python
# Load the dataset
file_url = 'https://drive.google.com/file/d/11ObgdgfvTUwDUOC38XdVMtxEeMRt2ibK/view?usp=drive_link'
path = 'https://drive.google.com/uc?export=download&id='+file_url.split('/')[-2]
df = pd.read_csv(path)
```

```python
# Display the first few rows to get an overview of the data
print("First few rows of the dataset:")
print(df.head())
```

```
First few rows of the dataset:
   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
```

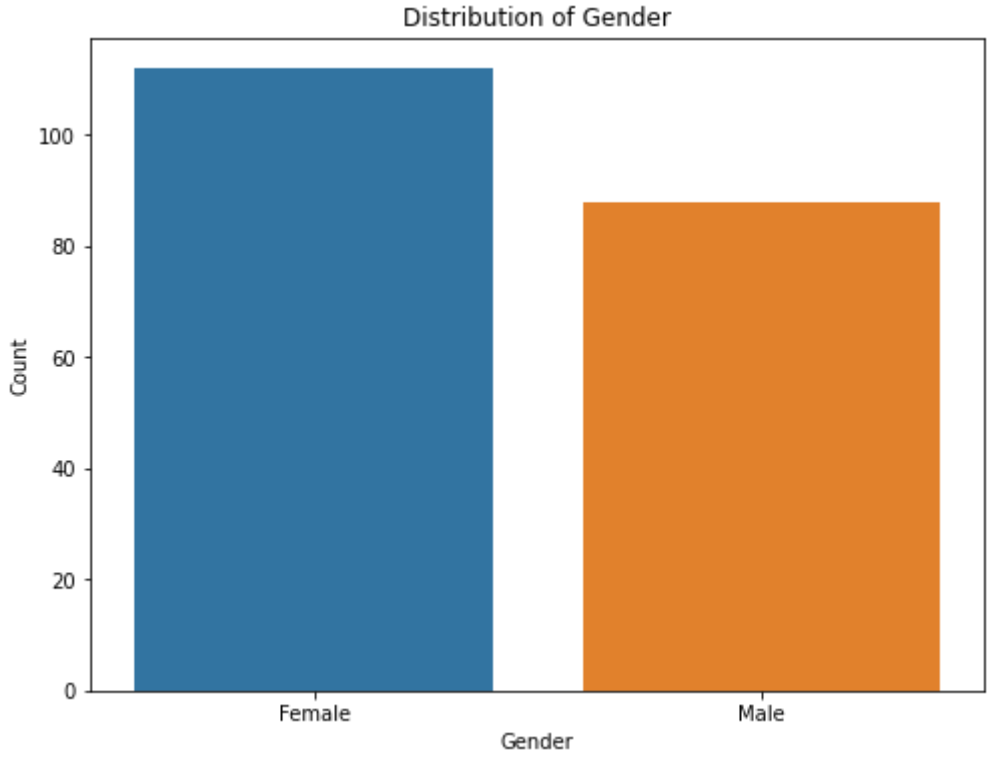
```python
# How many rows and columns are there in the dataset?
print("\nNumber of rows and columns:")
print(df.shape)  # Shape of the dataframe (rows, columns)
```

```
Number of rows and columns:
(200, 5)
```

```python
# What are the data types of the columns? Are there any missing values?
print("\nData types and missing values:")
print(df.info())

# Check for missing values
missing_values = df.isnull().sum()  # Summarize missing values per column
print("Missing values per column:")
print(missing_values)

# Check if there are any missing values in the entire DataFrame
any_missing = df.isnull().any().any()
print("\nAre there any missing values in the DataFrame?", any_missing)
```

```
Data types and missing values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 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$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
 5   Cluster                 200 non-null    int32
dtypes: int32(1), int64(4), object(1)
memory usage: 8.7+ KB
None
Missing values per column:
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
Cluster                   0
dtype: int64

Are there any missing values in the DataFrame? False
```

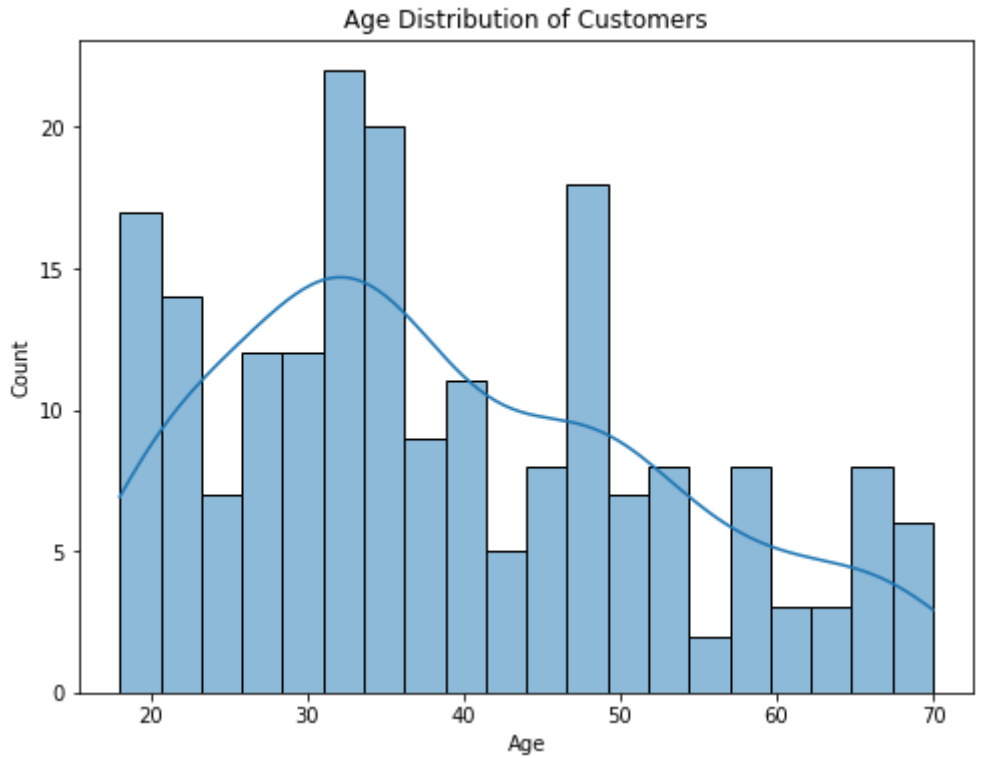
```python
# Explore the distribution of gender among the customers (Create a bar plot)
gender_counts = df['Gender'].value_counts()
plt.figure(figsize=(8, 6))
sns.barplot(x=gender_counts.index, y=gender_counts.values)
plt.title('Distribution of Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



```python
# Summary statistics for 'Annual Income' and 'Spending Score'
print("\nSummary statistics for 'Annual Income' and 'Spending Score':")
print(df[['Annual Income (k$)', 'Spending Score (1-100)']].describe())
```

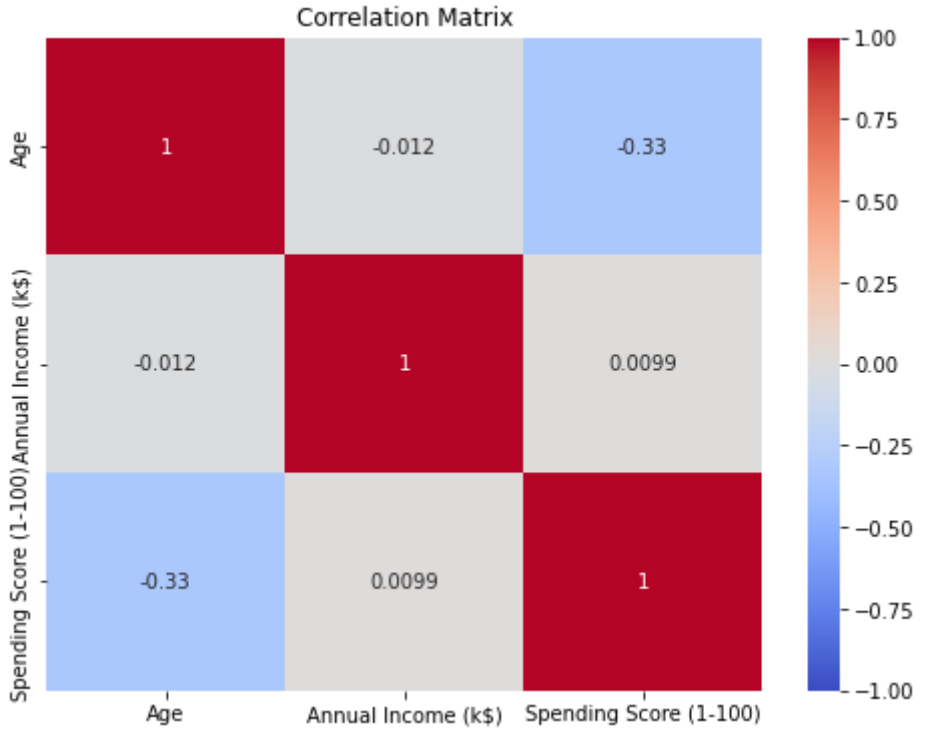
```
Summary statistics for 'Annual Income' and 'Spending Score':
       Annual Income (k$)  Spending Score (1-100)
count          200.000000              200.000000
mean            60.560000               50.200000
std             26.264721               25.823522
min             15.000000                1.000000
25%             41.500000               34.750000
50%             61.500000               50.000000
75%             78.000000               73.000000
max            137.000000               99.000000
```

```python
# Histogram for the 'Age' column to visualize the age distribution of customers
plt.figure(figsize=(8, 6))
sns.histplot(df['Age'], bins=20, kde=True)
plt.title('Age Distribution of Customers')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

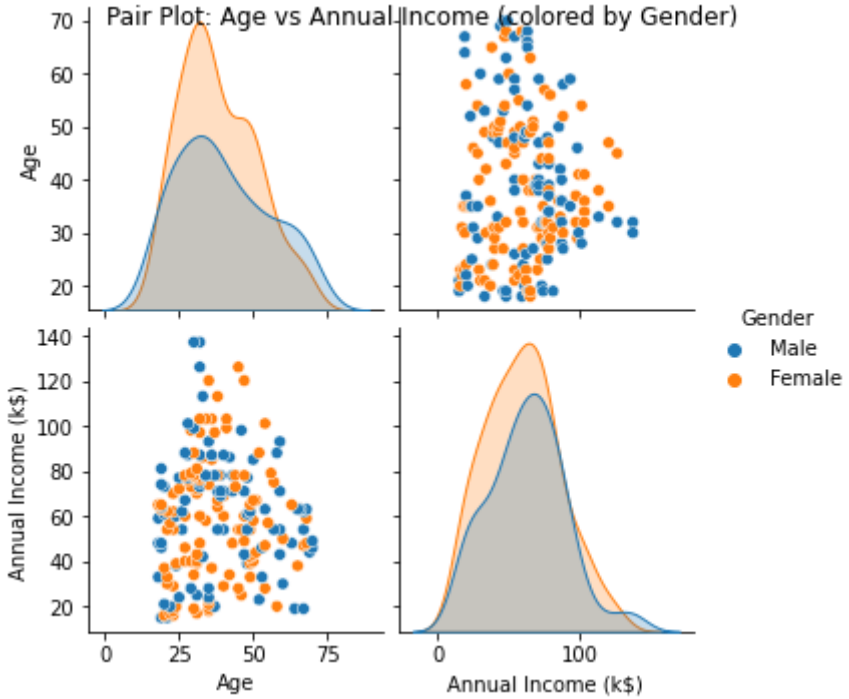


```python
# Scatter plot: 'Annual Income' vs 'Spending Score', colored by 'Gender'
plt.figure(figsize=(10, 8))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', hue='Gender', data=df)
plt.title('Annual Income vs Spending Score (colored by Gender)')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.show()
```

```python
# Correlation matrix for numerical columns
corr_matrix = df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix')
plt.show()
```



```
# Pairwise analysis of 'Age' and 'Annual Income' using a scatter plot matrix (pair plot)
sns.pairplot(df, vars=['Age', 'Annual Income (k$)'], hue='Gender')
plt.suptitle('Pair Plot: Age vs Annual Income (colored by Gender)')
plt.show()
```



```python
# KMeans clustering to segment customers based on 'Spending Score' and 'Annual Income'
X = df[['Annual Income (k$)', 'Spending Score (1-100)']]
kmeans = KMeans(n_clusters=5, random_state=0)  # Choosing 5 clusters arbitrarily
df['Cluster'] = kmeans.fit_predict(X)

# Visualizing the clusters
plt.figure(figsize=(10, 8))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', hue='Cluster', data=df, palette='viridis')
plt.title('Clusters of Customers based on Annual Income vs Spending Score')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.show()
```

The segments in the scatter plot are quite distinct, each representing a unique group of customers categorized by their annual income and spending score. Here are the key observations:

- High Income, Low Spending: One cluster shows customers with high income but low spending scores.
- Moderate Income & Spending: Another cluster represents customers with moderate values for both income and spending.
- Low Income, Low Spending: A separate cluster indicates customers with low income and low spending scores.
- High Income, High Spending: There is also a cluster for customers with high income and high spending scores.
- Low Income, High Spending: Lastly, a cluster exists for customers with low income but high spending scores.