```python
#1.1 Load the Dataset

import pandas as pd
data = pd.read_csv('Mall_Customers.csv')
# Display the first few rows of the dataset
data.head()
```

	CustomerID	Genre	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
#1.2 Check for missing values and handle them
print(data.isnull().sum())
data = data.dropna()
print(data.isnull().sum())
```

```
CustomerID       0
Gender           0
Age              0
AnnualIncome     0
SpendingScore    0
dtype: int64
CustomerID       0
Gender           0
Age              0
AnnualIncome     0
SpendingScore    0
dtype: int64
```

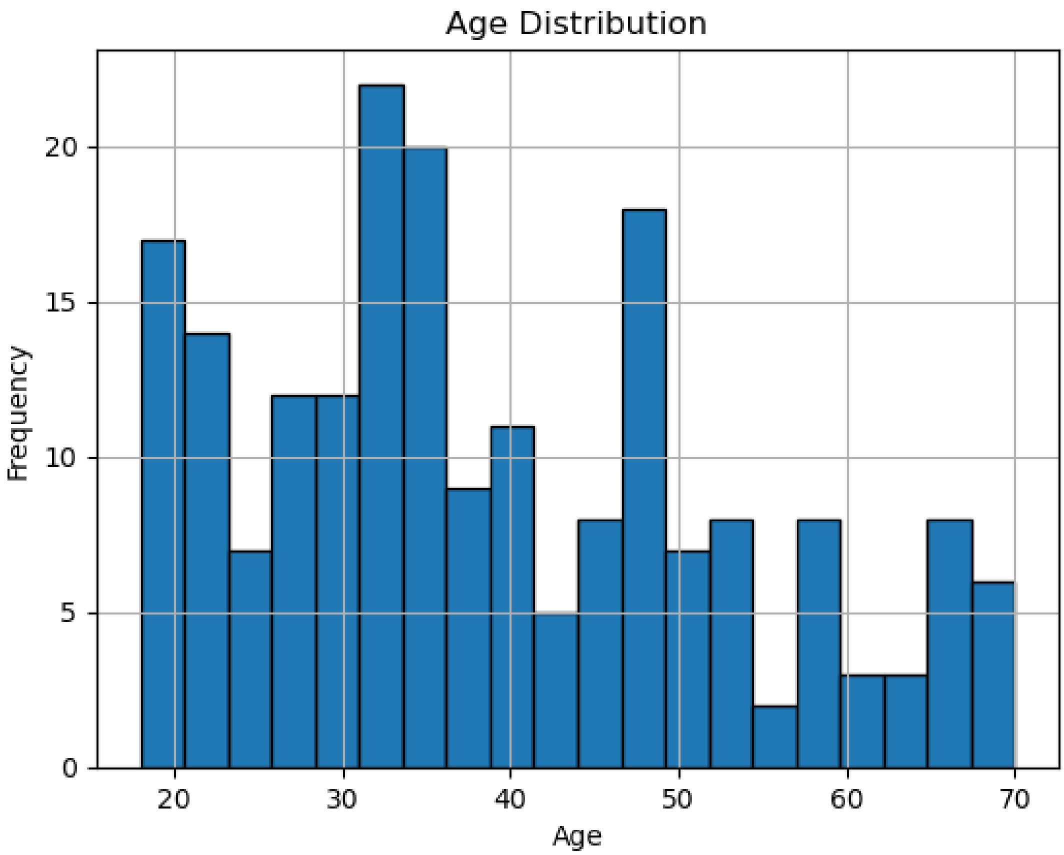
```python
#1.3 Encode categorical variables (e.g., gender) if necessary

data['Gender'] = data['Gender'].map({'Male': 0, 'Female': 1})
print(data.head(20))
```

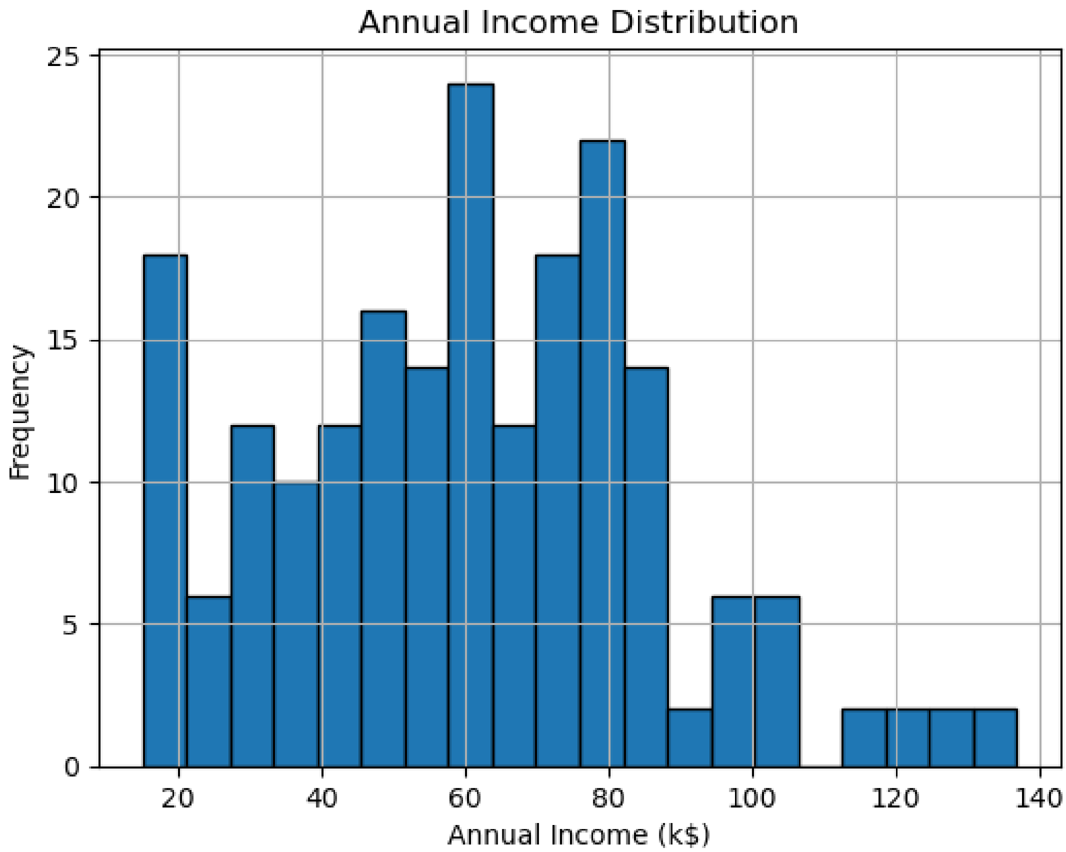
```
    CustomerID  Gender  Age  AnnualIncome  SpendingScore
0            1     NaN   19            15             39
1            2     NaN   21            15             81
2            3     NaN   20            16              6
3            4     NaN   23            16             77
4            5     NaN   31            17             40
5            6     NaN   22            17             76
6            7     NaN   35            18              6
7            8     NaN   23            18             94
8            9     NaN   64            19              3
9           10     NaN   30            19             72
10          11     NaN   67            19             14
11          12     NaN   35            19             99
12          13     NaN   58            20             15
```

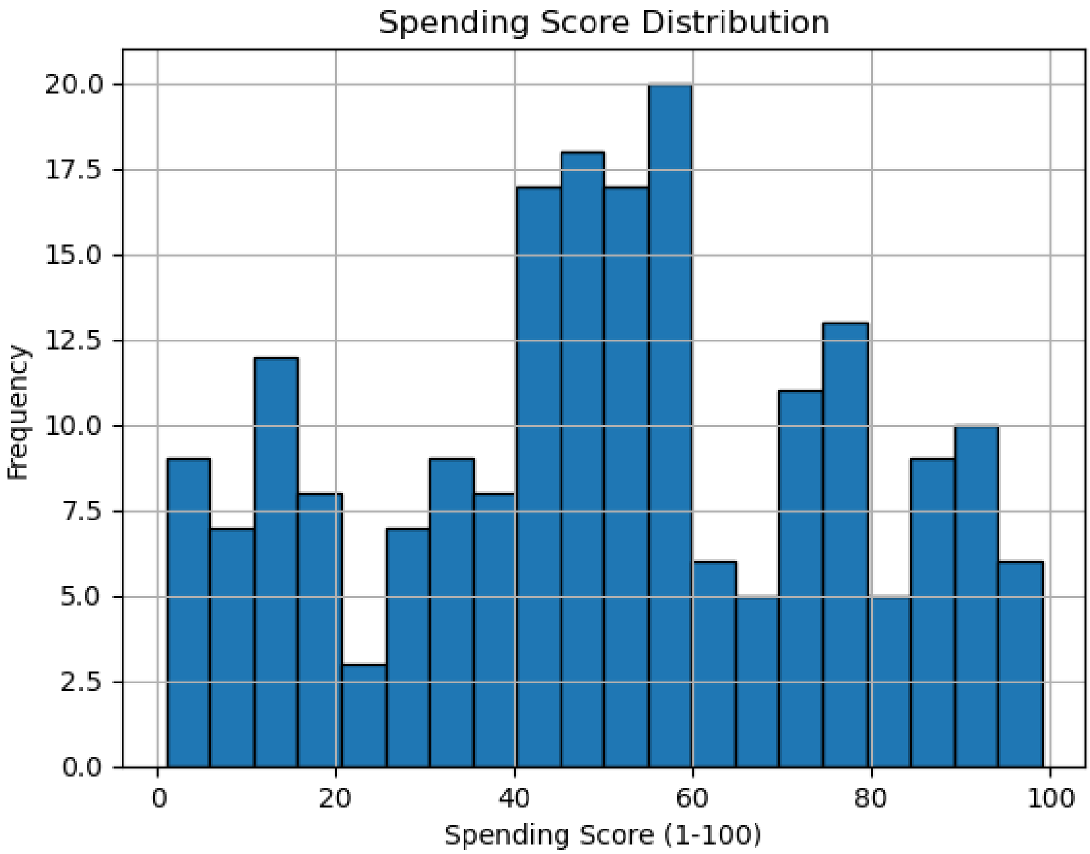
```
13           14     NaN   24             20             77
14           15     NaN   37             20             13
15           16     NaN   22             20             79
16           17     NaN   35             21             35
17           18     NaN   20             21             66
18           19     NaN   52             23             29
19           20     NaN   35             23             98

# Renaming columns for better readability
data.columns = ["CustomerID", "Gender", "Age", "AnnualIncome",
"SpendingScore"]
data

     CustomerID  Gender  Age  AnnualIncome  SpendingScore
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
..          ...     ...  ...           ...            ...
195         196  Female   35           120             79
196         197  Female   45           126             28
197         198    Male   32           126             74
198         199    Male   32           137             18
199         200    Male   30           137             83

[200 rows x 5 columns]

#1.4 Save the cleaned dataset
data.to_csv('cleaned_customers.csv', index=False)

#1.5 Data Cleaning Summary

# 1. Loaded the dataset from 'Customers.csv'.
# 2. Checked for missing values and dropped rows with any missing
values.
# 3. Encoded the 'Gender' column: 'Male' as 0 and 'Female' as 1.
# 4. Saved the cleaned dataset as 'cleaned_customers.csv'.

    CustomerID  Gender  Age  AnnualIncome  SpendingScore
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
10          11    Male   67            19             14
```

```
11          12  Female   35             19              99
12          13  Female   58             20              15
13          14  Female   24             20              77
14          15    Male   37             20              13
15          16    Male   22             20              79
16          17  Female   35             21              35
17          18    Male   20             21              66
18          19    Male   52             23              29
19          20  Female   35             23              98

#Step 2: Exploratory Data Analysis (EDA)

CustomerID       0
Gender           0
Age              0
AnnualIncome     0
SpendingScore    0
dtype: int64

#2.1 Calculate descriptive statistics for the dataset
stats = data.describe()
print(stats)

       CustomerID         Age  AnnualIncome  SpendingScore
count  200.000000  200.000000    200.000000     200.000000
mean   100.500000   38.850000     60.560000      50.200000
std     57.879185   13.969007     26.264721      25.823522
min      1.000000   18.000000     15.000000       1.000000
25%     50.750000   28.750000     41.500000      34.750000
50%    100.500000   36.000000     61.500000      50.000000
75%    150.250000   49.000000     78.000000      73.000000
max    200.000000   70.000000    137.000000      99.000000

#2.2 Create histograms for age, annual income, and spending score
distributions
import matplotlib.pyplot as plt

# Histograms
data['Age'].hist(bins=20, edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.savefig('age_distribution.png')
plt.show()

data['AnnualIncome'].hist(bins=20, edgecolor='black')
plt.title('Annual Income Distribution')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Frequency')
plt.savefig('annual_income_distribution.png')
plt.show()
```

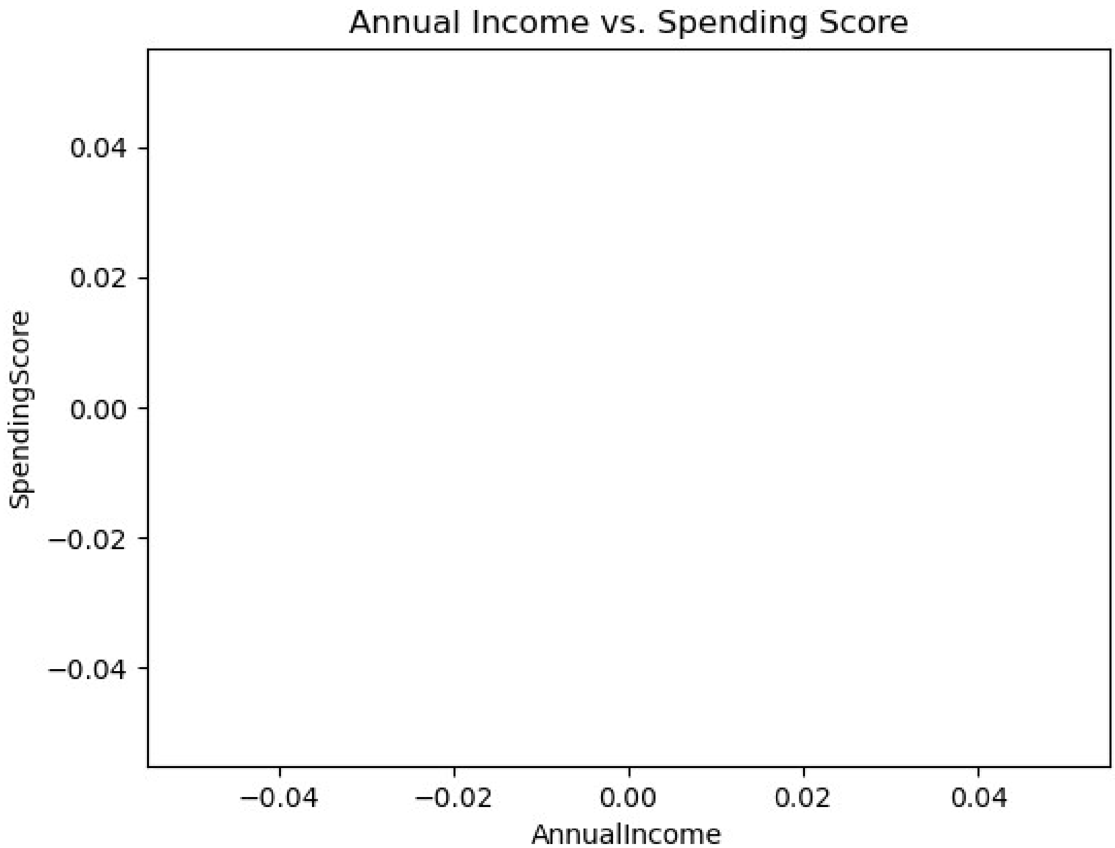
```python
data['SpendingScore'].hist(bins=20, edgecolor='black')
plt.title('Spending Score Distribution')
plt.xlabel('Spending Score (1-100)')
plt.ylabel('Frequency')
plt.savefig('spending_score_distribution.png')
plt.show()
```




Annual Income Distribution
Frequency
Annual Income (k$)
0
5
10
15
20
25
20
40
60
80
100
120
140



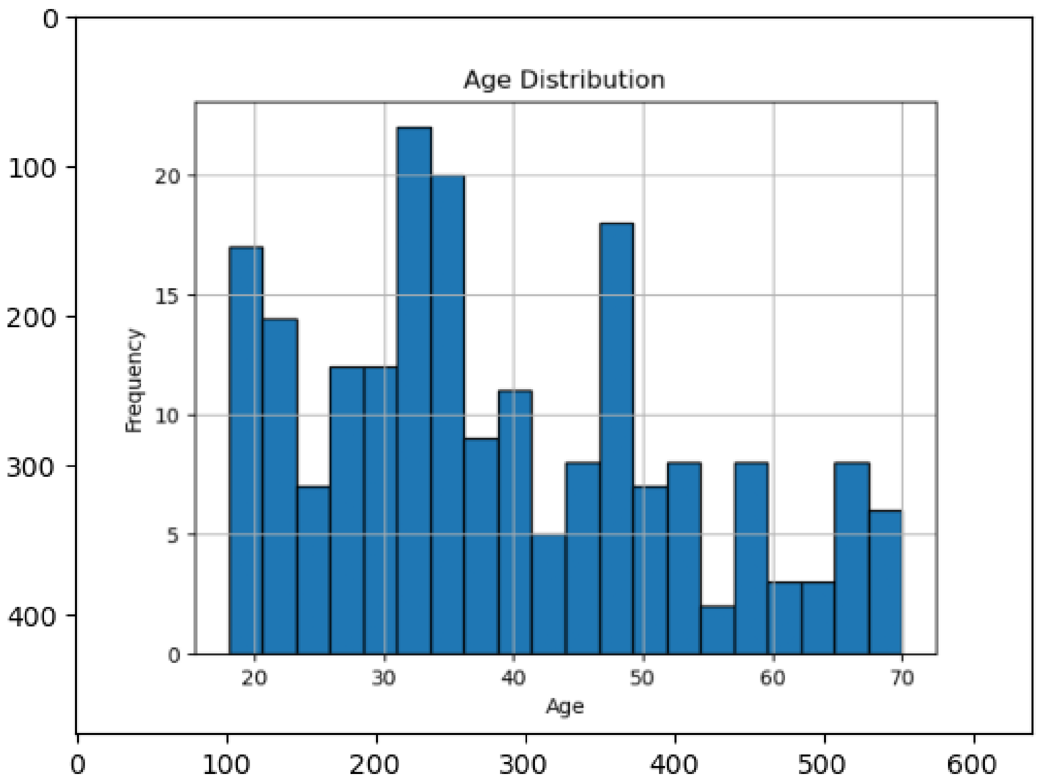

```python
#2.3 Generate a scatter plot of annual income vs. spending score,
colored by gender
# Scatter plot
import matplotlib.pyplot as plt
colors = {0: 'blue', 1: 'red'}
plt.scatter(data['AnnualIncome'], data['SpendingScore'],
            c=data['Gender'].map(colors))
plt.title('Annual Income vs. Spending Score')
plt.xlabel('AnnualIncome')
plt.ylabel('SpendingScore')
plt.savefig('income_vs_spending.png')
plt.show()
```

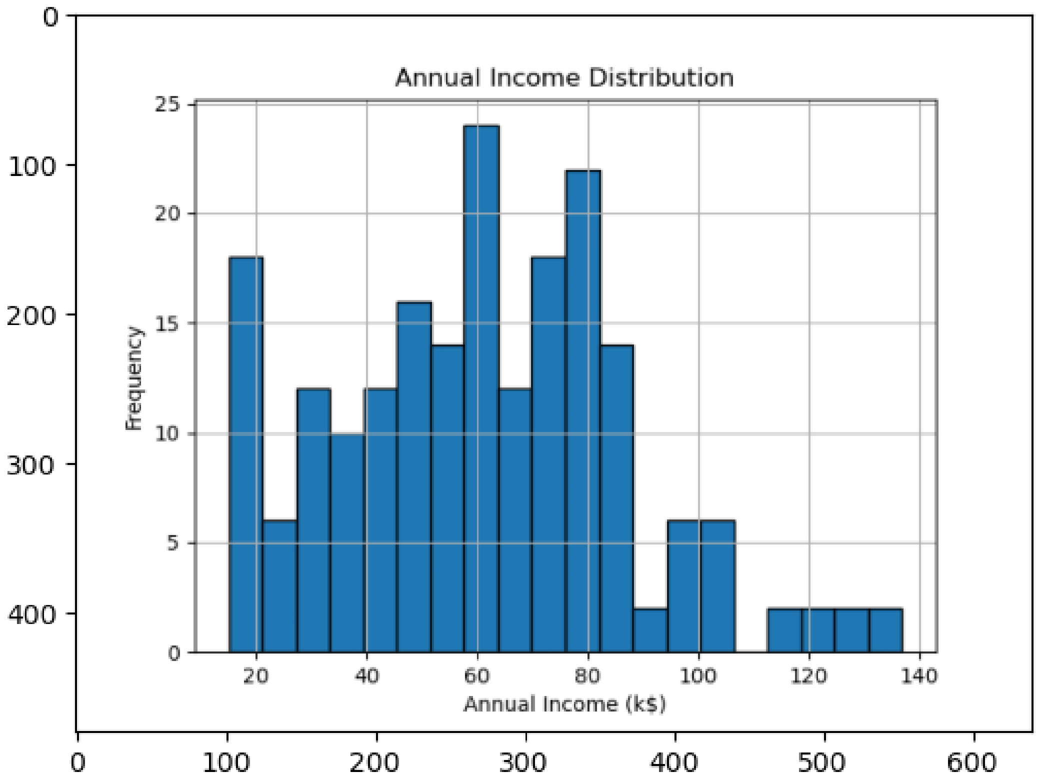


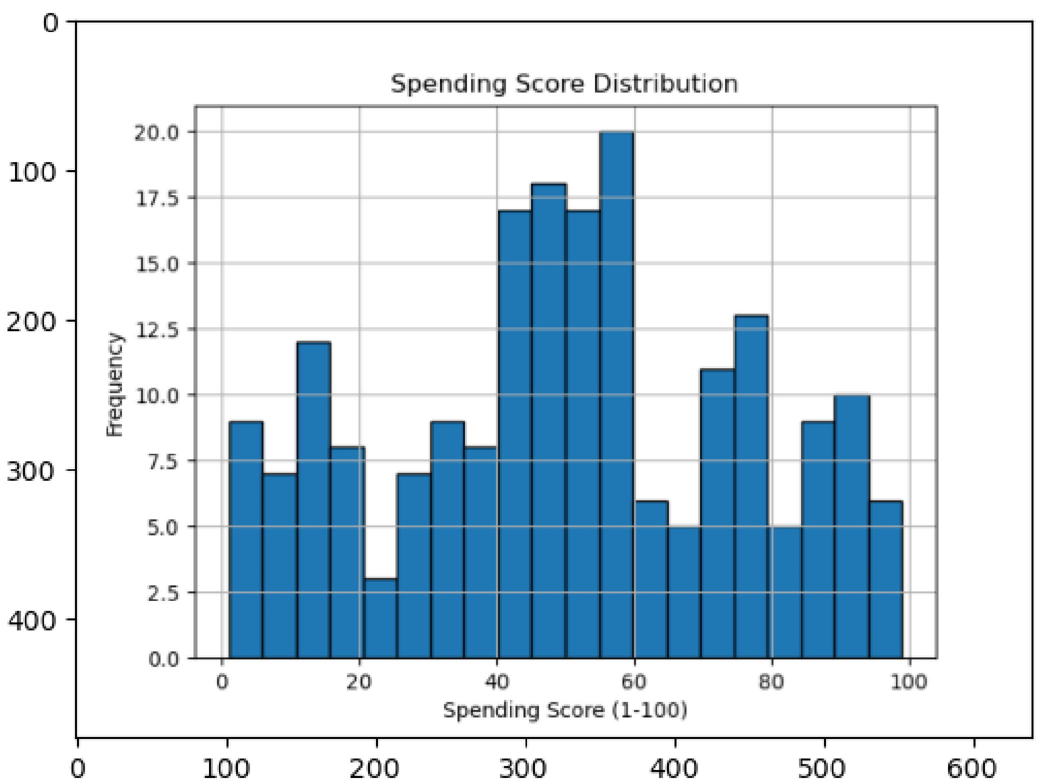

```python
#2.4 Save the EDA plots and statistics in a Jupyter notebook
(EDA.ipynb)
### Exploratory Data Analysis

#1. Descriptive Statistics
#```python
print(stats)

#2. Age Distribution
plt.imshow(plt.imread('age_distribution.png'))
plt.show()

#3. Annual Income Distribution
plt.imshow(plt.imread('annual_income_distribution.png'))
plt.show()

#4. Spending Score Distribution
plt.imshow(plt.imread('spending_score_distribution.png'))
plt.show()

#5. Annual Income vs. Spending Score
plt.imshow(plt.imread('spending_score_distribution.png'))
```

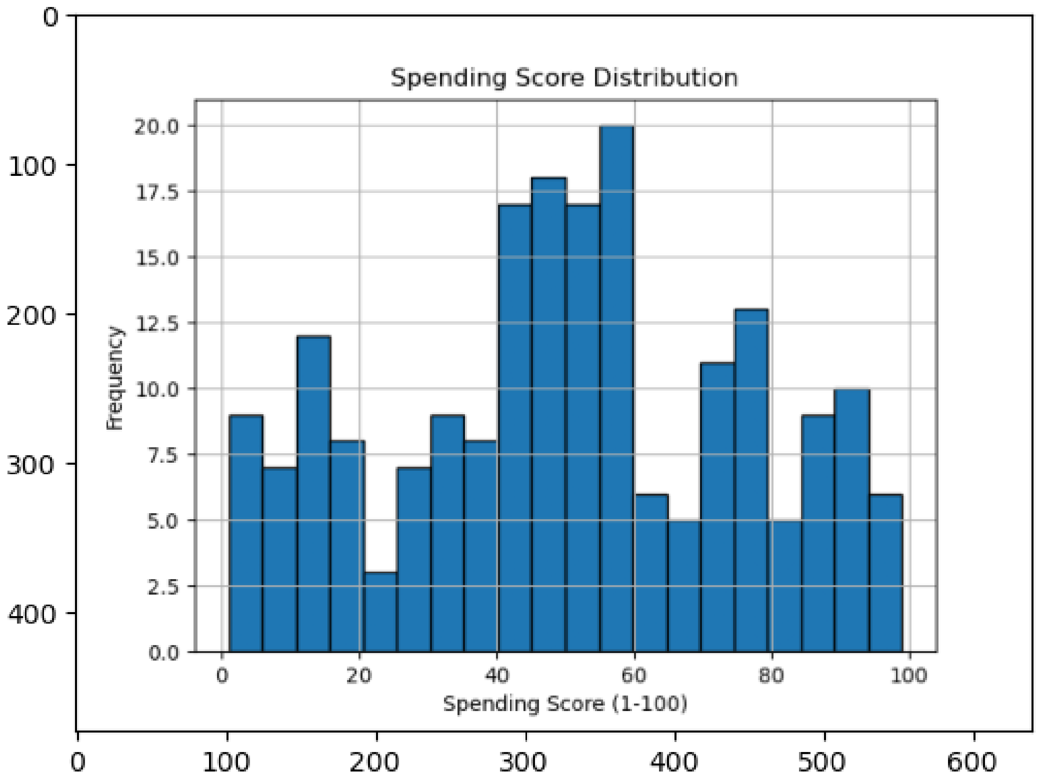
```python
plt.show()
#```python
```

```
       CustomerID         Age  AnnualIncome  SpendingScore
count  200.000000  200.000000    200.000000     200.000000
mean   100.500000   38.850000     60.560000      50.200000
std     57.879185   13.969007     26.264721      25.823522
min      1.000000   18.000000     15.000000       1.000000
25%     50.750000   28.750000     41.500000      34.750000
50%    100.500000   36.000000     61.500000      50.000000
75%    150.250000   49.000000     78.000000      73.000000
max    200.000000   70.000000    137.000000      99.000000
```



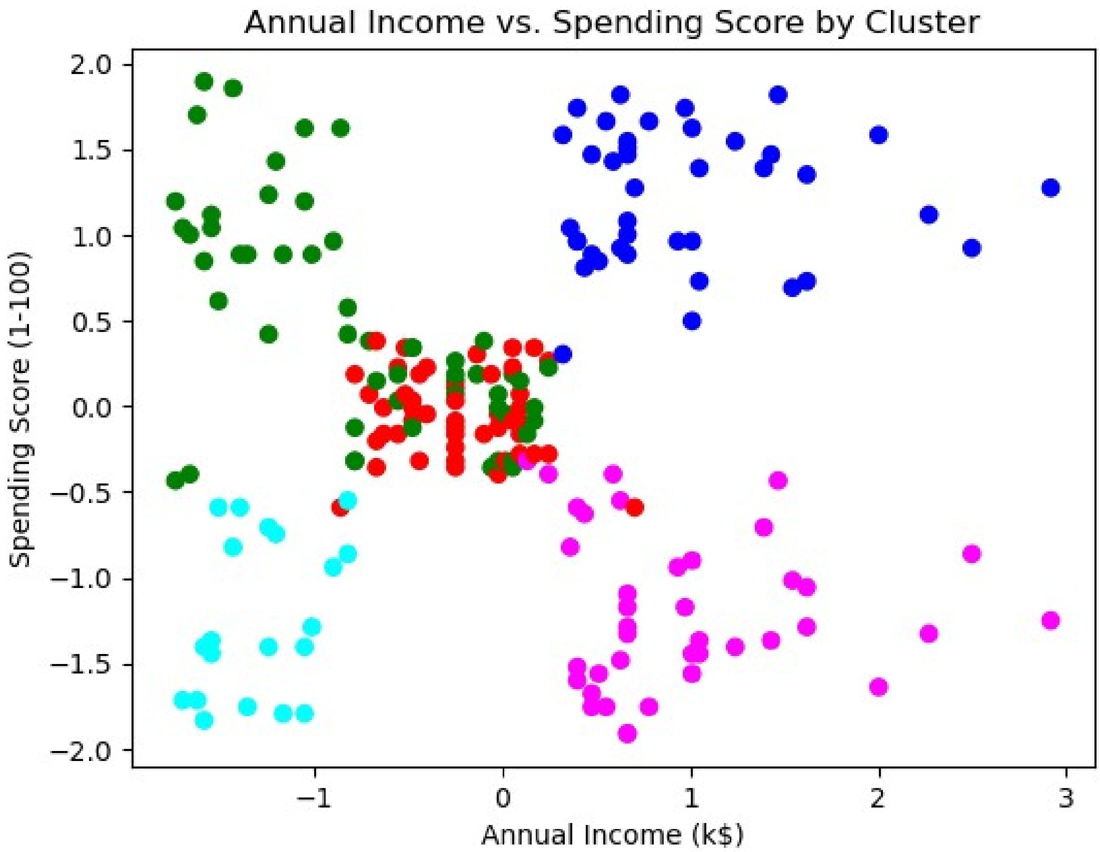

Annual Income Distribution
Frequency
Annual Income (k$)
0
100
200
300
400
0
100
200
300
400
500
600
25
20
15
10
5
0
20
40
60
80
100
120
140


Spending Score Distribution
Frequency
Spending Score (1-100)
0
100
200
300
400
0
100
200
300
400
500
600
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100



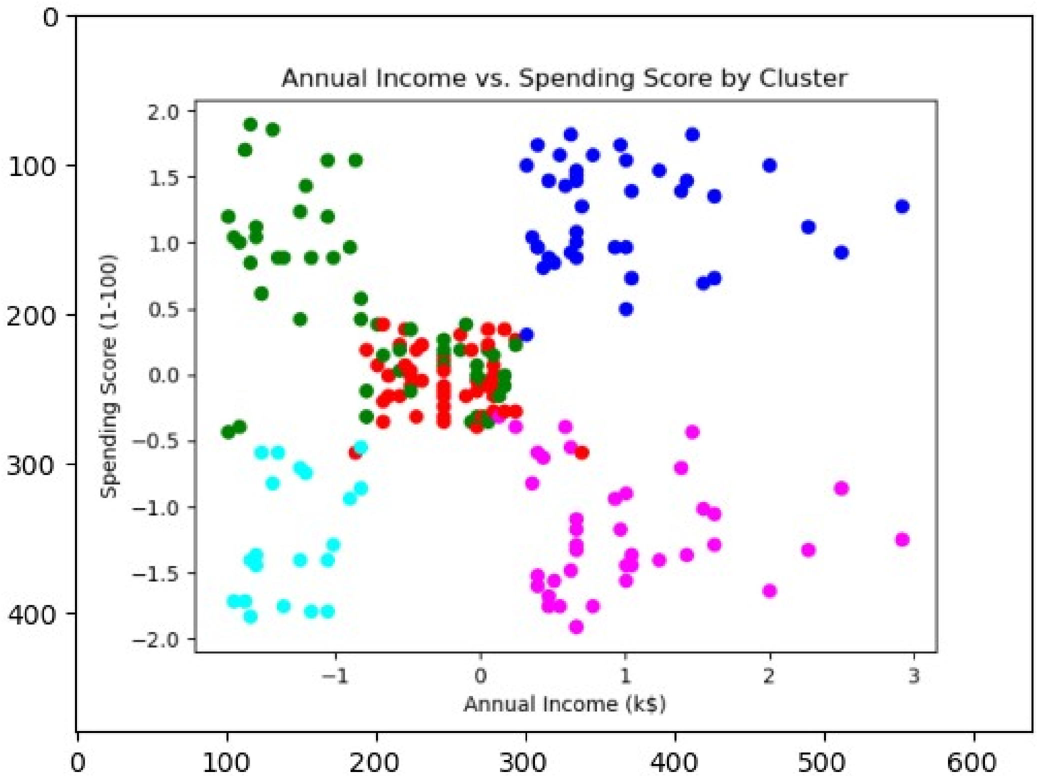

```
### Step 3: Customer Segmentation

# 3.1 Standardize the features (age, annual income, spending score)
from sklearn.preprocessing import StandardScaler

# Standardize the features
scaler = StandardScaler()
data[['Age', 'AnnualIncome', 'SpendingScore']] =
scaler.fit_transform(data[['Age', 'AnnualIncome', 'SpendingScore']])
print(data.head())

   CustomerID  Gender       Age  AnnualIncome  SpendingScore
0           1    Male -1.424569     -1.738999      -0.434801
1           2    Male -1.281035     -1.738999       1.195704
2           3  Female -1.352802     -1.700830      -1.715913
3           4  Female -1.137502     -1.700830       1.040418
4           5  Female -0.563369     -1.662660      -0.395980

#3.2 Apply K-Means clustering to segment the customers into 5 clusters
from sklearn.cluster import KMeans

# Apply K-Means clustering
kmeans = KMeans(n_clusters=5, random_state=42)
data['Cluster'] = kmeans.fit_predict(data[['Age', 'AnnualIncome',
'SpendingScore']])
print(data.head())
```

```
C:\Users\Arindam\anaconda3\Lib\site-packages\sklearn\cluster\
_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\Arindam\anaconda3\Lib\site-packages\sklearn\cluster\
_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads.
You can avoid it by setting the environment variable
OMP_NUM_THREADS=1.
  warnings.warn(

   CustomerID  Gender       Age  AnnualIncome  SpendingScore  Cluster
0           1    Male -1.424569     -1.738999      -0.434801        2
1           2    Male -1.281035     -1.738999       1.195704        2
2           3  Female -1.352802     -1.700830      -1.715913        3
3           4  Female -1.137502     -1.700830       1.040418        2
4           5  Female -0.563369     -1.662660      -0.395980        2

#3.3 Create a scatter plot of annual income vs. spending score,
colored by cluster
# Scatter plot
colors = {0: 'red', 1: 'blue', 2: 'green', 3: 'cyan', 4: 'magenta'}
plt.scatter(data['AnnualIncome'],data['SpendingScore'],
            c=data['Cluster'].map(colors))
plt.title('Annual Income vs. Spending Score by Cluster')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.savefig('income_vs_spending_clusters.png')
plt.show()
```




```
#3.4 Save the clustering code and plot in a Jupyter notebook
(Clustering.ipynb)
# Clustering.ipynb

### Customer Segmentation

#1. Standardize the Features



#2. Apply K-Means Clustering
print(data.head())

#3. Annual Income vs. Spending Score by Cluster
plt.imshow(plt.imread('income_vs_spending_clusters.png'))
plt.show()

   CustomerID  Gender       Age  AnnualIncome  SpendingScore  Cluster
0           1    Male -1.424569     -1.738999      -0.434801        2
1           2    Male -1.281035     -1.738999       1.195704        2
2           3  Female -1.352802     -1.700830      -1.715913        3
3           4  Female -1.137502     -1.700830       1.040418        2
4           5  Female -0.563369     -1.662660      -0.395980        2
   CustomerID  Gender       Age  AnnualIncome  SpendingScore  Cluster
```

```
0            1    Male -1.424569     -1.738999      -0.434801        2
1            2    Male -1.281035     -1.738999       1.195704        2
2            3  Female -1.352802     -1.700830      -1.715913        3
3            4  Female -1.137502     -1.700830       1.040418        2
4            5  Female -0.563369     -1.662660      -0.395980        2
```



Step 4: Insights and Recommendations

4.1 Analyze the customer segments and provide insights

```markdown

## Insights and Recommendations

Step 4.2 Write a Report Summarizing Key Findings and Recommendations You can use a word processor (like Microsoft Word or Google Docs) to create the report. Here's an example structure for the report:

Customer Segmentation Insights Report

Executive Summary This report presents the findings from a customer segmentation analysis performed on the dataset containing customer information such as Age, Gender, Annual Income, and Spending Score. The main goal of this project is to identify distinct customer segments and provide actionable insights for targeted marketing strategies.
```

Data Cleaning and Preparation Data Loading: The dataset was loaded from 'Customers.csv'. Missing Values: The dataset was checked for missing values, and rows with any missing values were removed. Encoding: The Gender column was encoded as 0 for Male and 1 for Female.

Standardization: Features such as Age, Annual Income, and Spending Score were standardized to ensure equal weightage in the clustering process. Exploratory Data Analysis Descriptive Statistics: Summary statistics were calculated to understand the central tendency and dispersion of the dataset. Histograms: Histograms for Age, Annual Income, and Spending Score distributions revealed the data's spread and skewness. Scatter Plot: A scatter plot of Annual Income vs. Spending Score colored by Gender provided initial visual insights into potential clusters. Customer Segmentation Using K-Means clustering, customers were segmented into five distinct clusters based on their Age, Annual Income, and Spending Score. Each cluster exhibits unique characteristics and behaviors.

Cluster 0: Characteristics: Young adults with moderate income and average spending scores. Marketing Strategy: Promote budget-friendly products and loyalty programs. Cluster 1: Characteristics: Older adults with high income and high spending scores. Marketing Strategy: Offer premium products and exclusive deals. Cluster 2: Characteristics: Middle-aged individuals with low income and low spending scores. Marketing Strategy: Introduce value-for-money products and discount offers. Cluster 3: Characteristics: Young individuals with high income and high spending scores. Marketing Strategy: Focus on trendy, high-end products and personalized marketing. Cluster 4: Characteristics: Adults with varied income but consistent high spending scores. Marketing Strategy: Emphasize customer service and quality assurance. Insights and Recommendations Targeted Marketing: Utilize the insights from the clusters to develop targeted marketing campaigns tailored to the needs and preferences of each segment. Product Development: Innovate new products or enhance existing ones based on the preferences of high spending clusters. Customer Retention: Implement loyalty programs and personalized offers for high-value customers to improve retention rates. Budget Allocation: Allocate marketing budgets more effectively by focusing on segments with higher spending potential.