CUSTOMER SEGMENTATION USING DATASCIENCE

Phase 3 submission Documents

Project Titile: Customer segmentation using datascience

Name: Yalisha.R
Reg No:712221104027
College:park college of engineering and technology

Introduction:

- Customer segmentation using data science is a powerful technique that involves dividing a company's customer base into distinct groups based on specific characteristics, behaviors, or demographics.

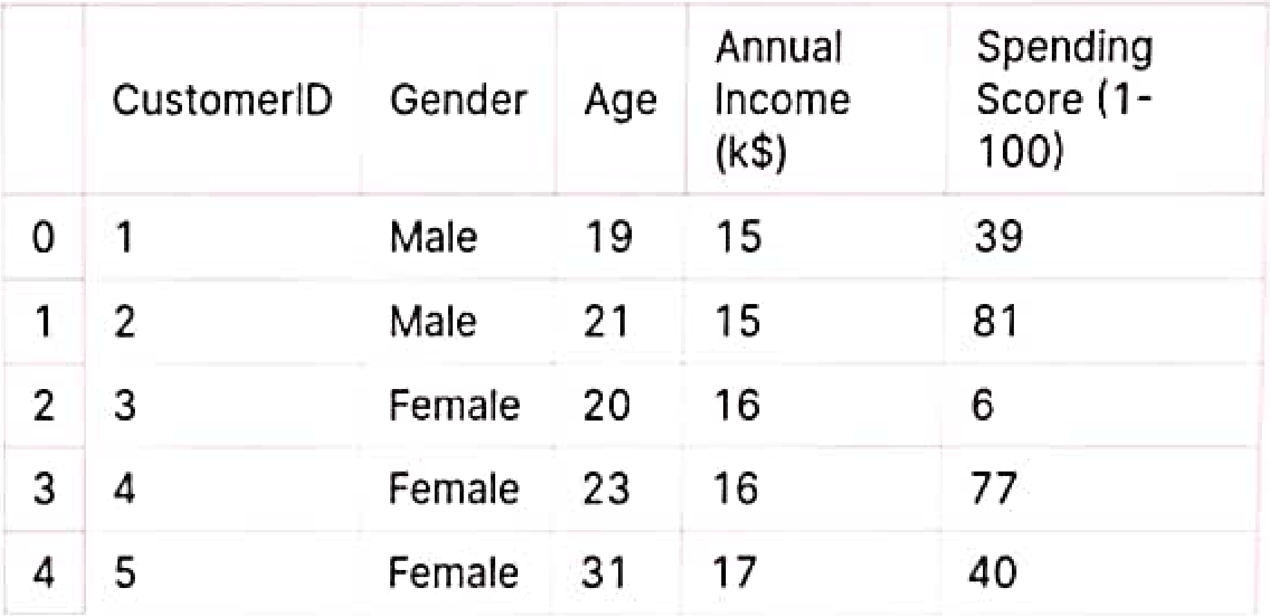
- By doing so, businesses can gain valuable insights into their customers, allowing them to tailor their marketing strategies, products, and services to meet the unique needs of each segment.

- Data science techniques, such as clustering algorithms, machine learning models, and data mining, are employed to analyze large sets of customer data.

- These methods identify patterns, trends, and relationships within the data, enabling businesses to create meaningful segments.

- The benefits of customer segmentation using data science include improved customer targeting, increased customer satisfaction, personalized marketing campaigns, and ultimately, higher sales and profitability.

GIVEN DATASET:



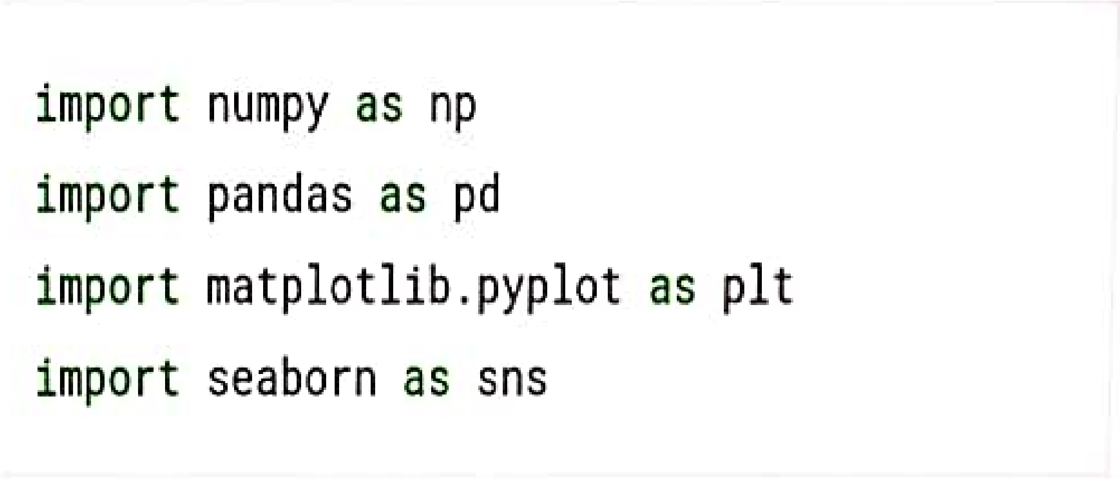
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Necessary step to follow:

1.Import libraries:

Start by importing the necessary libraries:

Program:

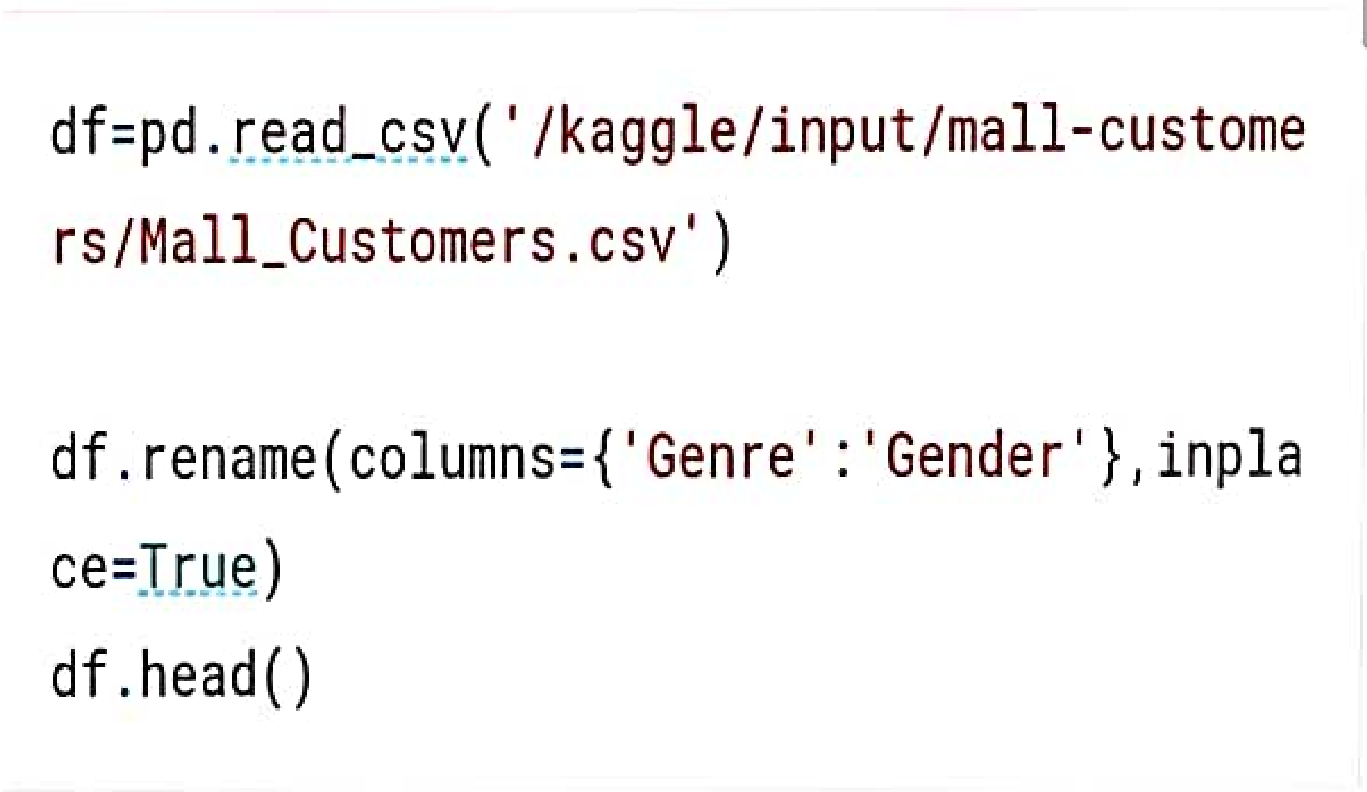


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Load the Dataset:

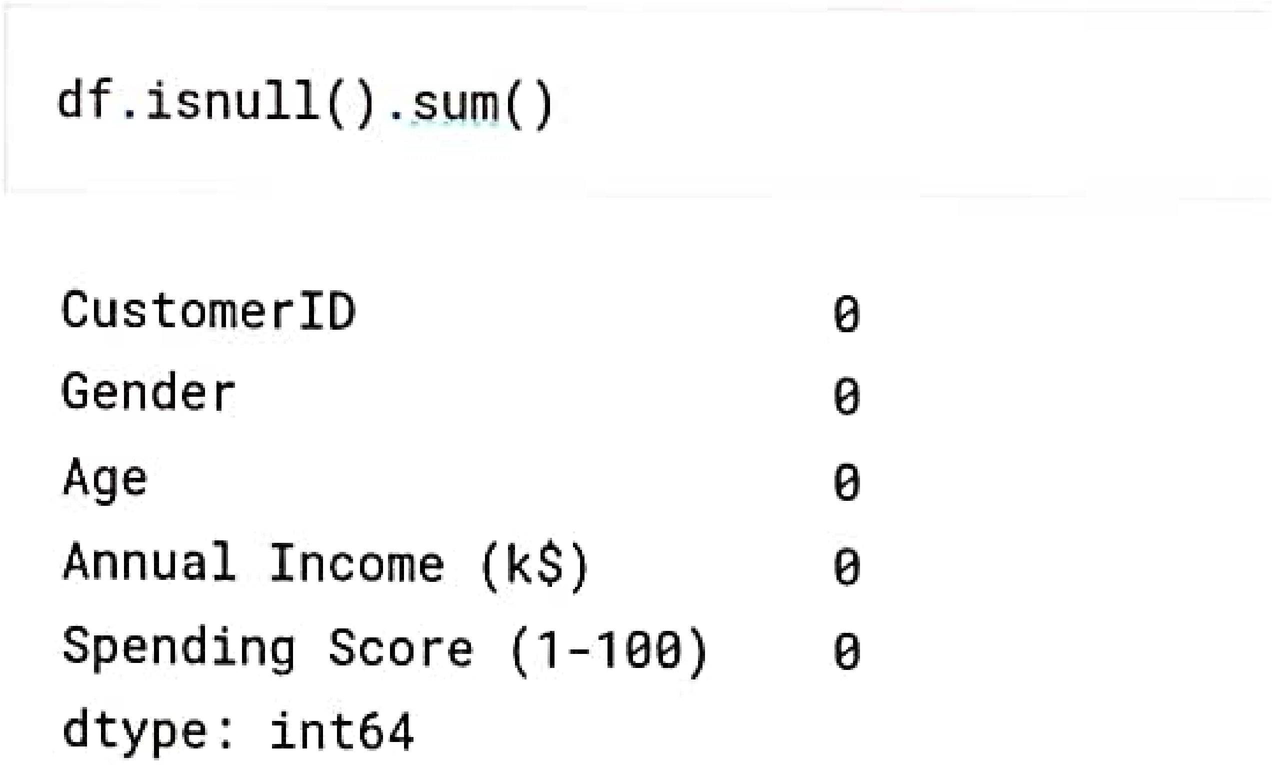
Load your dataset into a pandas dataframe. You Can typically find customer segmentation using datascience dataset in CSV format,but you can adapt code to other formats as needed.

Program:



```
df=pd.read_csv('/kaggle/input/mall-custome
rs/Mall_Customers.csv')

df.rename(columns={'Genre':'Gender'},inpla
ce=True)
df.head()
```

Df.describe()



```
df.isnull().sum()
```

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

challenge involved in loading and preprocessing a customer segmentation using datascience dataset:

Data Quality:

❖ Incomplete or missing data: You may encounter missing values in your dataset, and deciding how to handle them (imputation or removal) is critical.

❖ Outliers: Identifying and dealing with outliers that could skew your segmentation analysis is important.

Data Cleaning:

- Data may contain inconsistencies, errors, and duplicates that need to be addressed.
- Standardizing and normalizing data, especially for categorical variables, is necessary.

How to overcome the challenge involved in loading and preprocessing customer segmentation using datascience dataset:

Data Integration:

- Create a comprehensive data integration plan to merge data from different sources. Ensure that all data is consistent in terms of format and units.

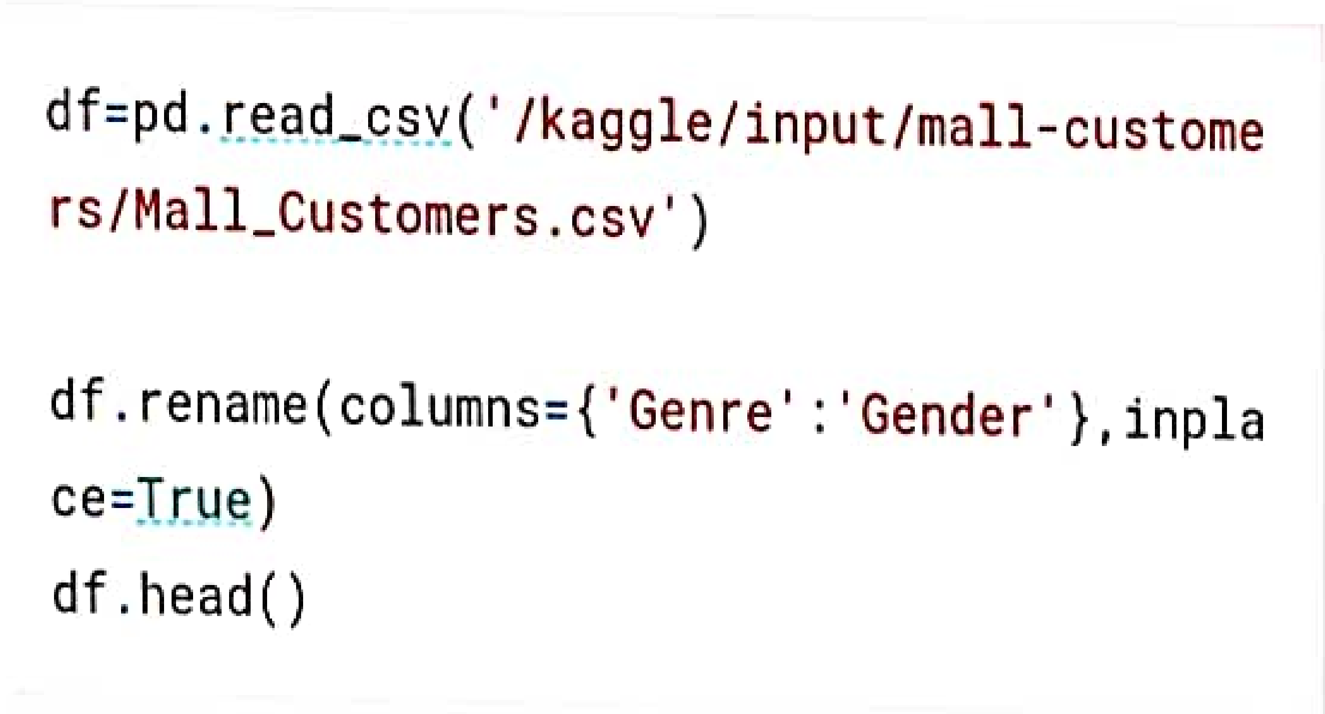
Data Scaling and Transformation:

- Scale numerical features to ensure that they have equal weight in the segmentation process.
- Apply necessary transformations, such as logarithmic transformations, to make data more suitable for clustering.

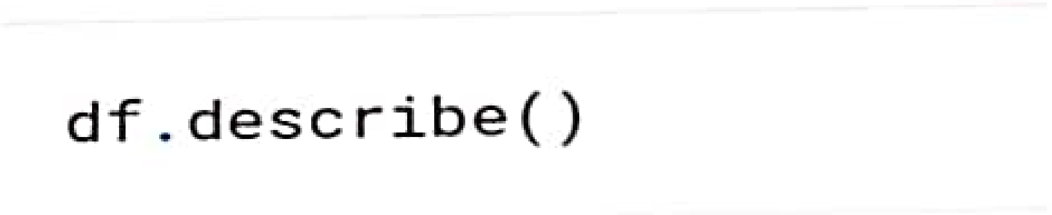
Loading the Dataset:

- **Data Exploration:** After loading the dataset, it's a good practice to explore the data to understand its structure and the information it contains. You can use functions like **`head()`**, **`info()`**, and **`describe()`** to get an initial overview of the data.
- **Load the Dataset:** You can load your dataset from various sources like CSV files, Excel files, or databases.

Program:



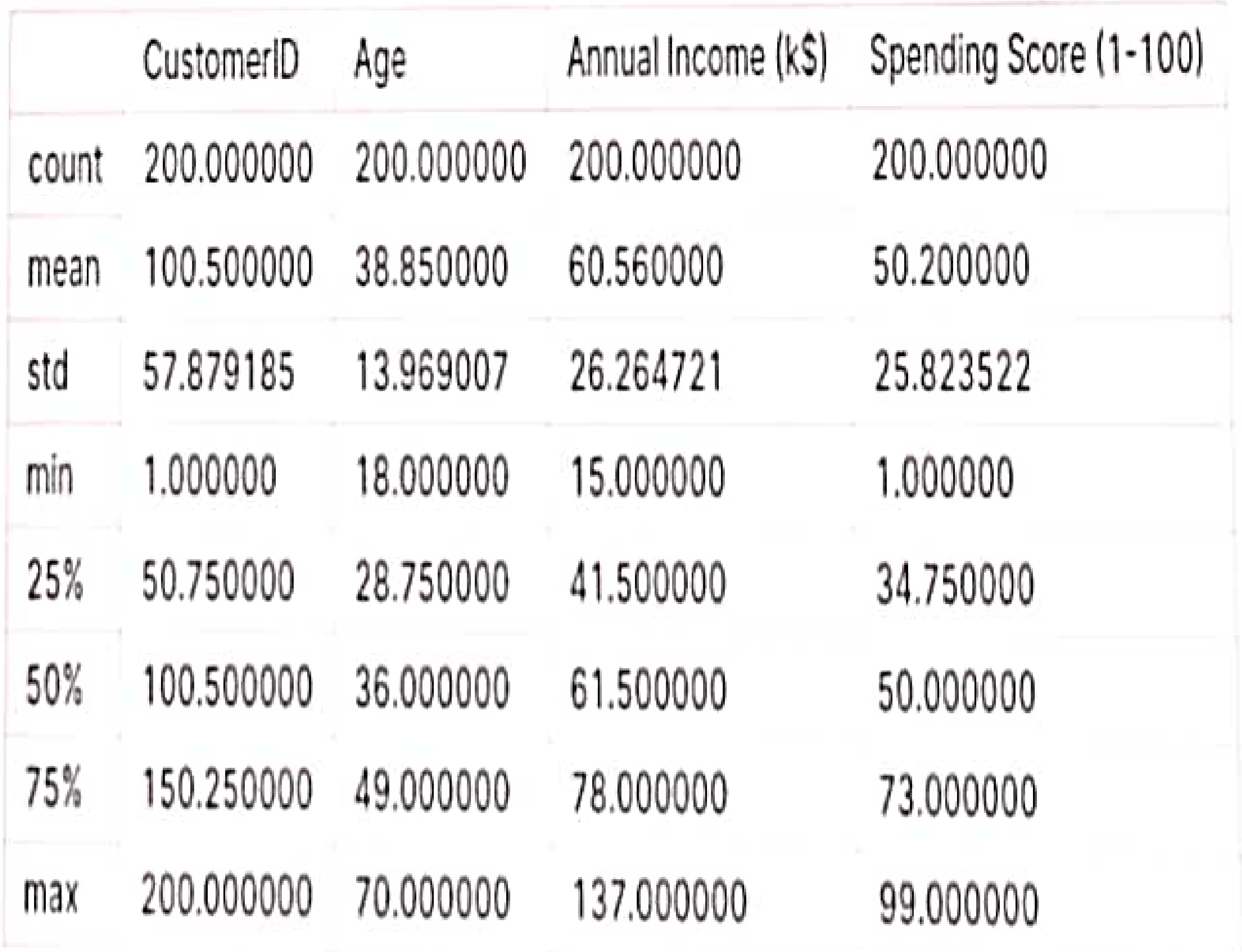
```
df=pd.read_csv('/kaggle/input/mall-customers/Mall_Customers.csv')

df.rename(columns={'Genre':'Gender'},inplace=True)
df.head()
```



```
df.describe()
```

Loading the Dataset:

Output:



	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
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

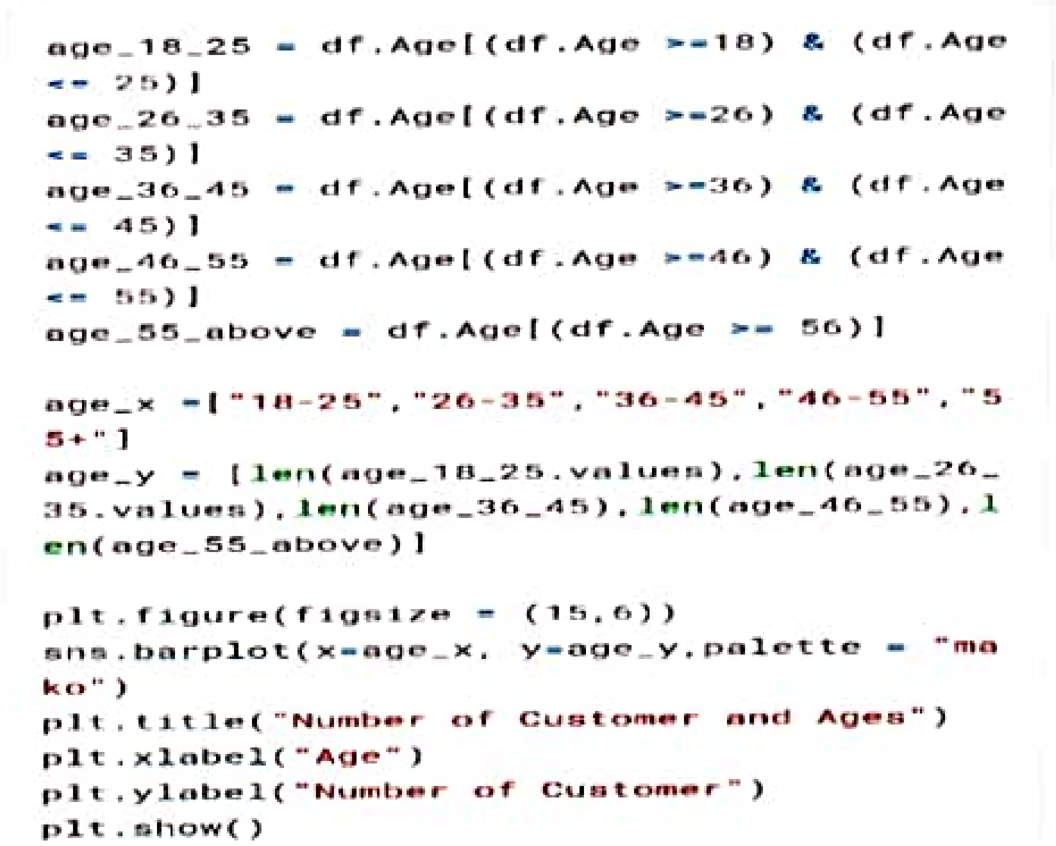
Preprocessing the Dataset:

- **Handling Missing Values:** Check for missing data in your dataset and decide on an appropriate strategy for handling them. You can either fill in missing values with a specific value (e.g., mean, median, or mode) or remove rows or columns with too many missing values.

- **Feature Scaling:** Depending on the algorithms you plan to use for segmentation, it might be necessary to scale or normalize your numerical features to have a consistent scale.

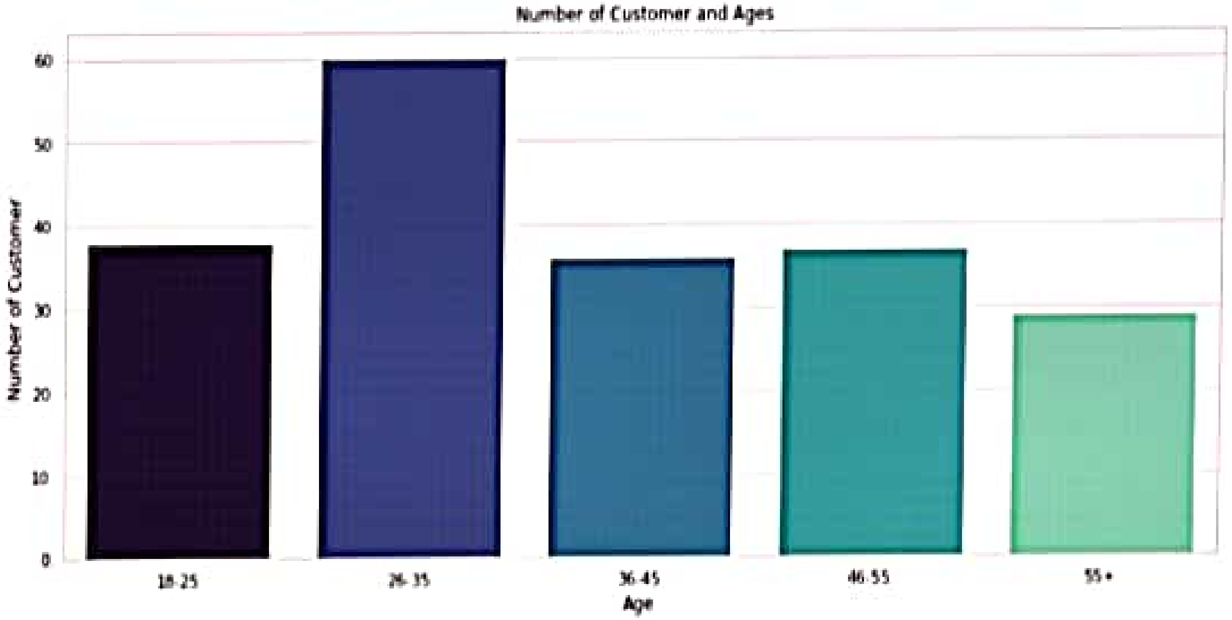
Virtualization and preprocessing of data:

In[1]:

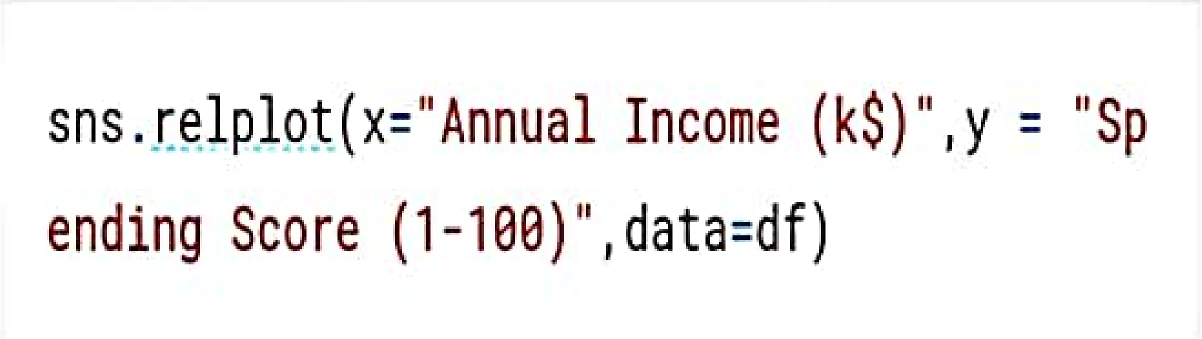


```
age_18_25 = df.Age[(df.Age >=18) & (df.Age
<= 25)]
age_26_35 = df.Age[(df.Age >=26) & (df.Age
<= 35)]
age_36_45 = df.Age[(df.Age >=36) & (df.Age
<= 45)]
age_46_55 = df.Age[(df.Age >=46) & (df.Age
<= 55)]
age_55_above = df.Age[(df.Age >= 56)]

age_x =["18-25","26-35","36-45","46-55","5
5+"]
age_y = [len(age_18_25.values),len(age_26_
35.values),len(age_36_45),len(age_46_55),l
en(age_55_above)]

plt.figure(figsize = (15,6))
sns.barplot(x=age_x, y=age_y,palette = "ma
ko")
plt.title("Number of Customer and Ages")
plt.xlabel("Age")
plt.ylabel("Number of Customer")
plt.show()
```

Out[1]:



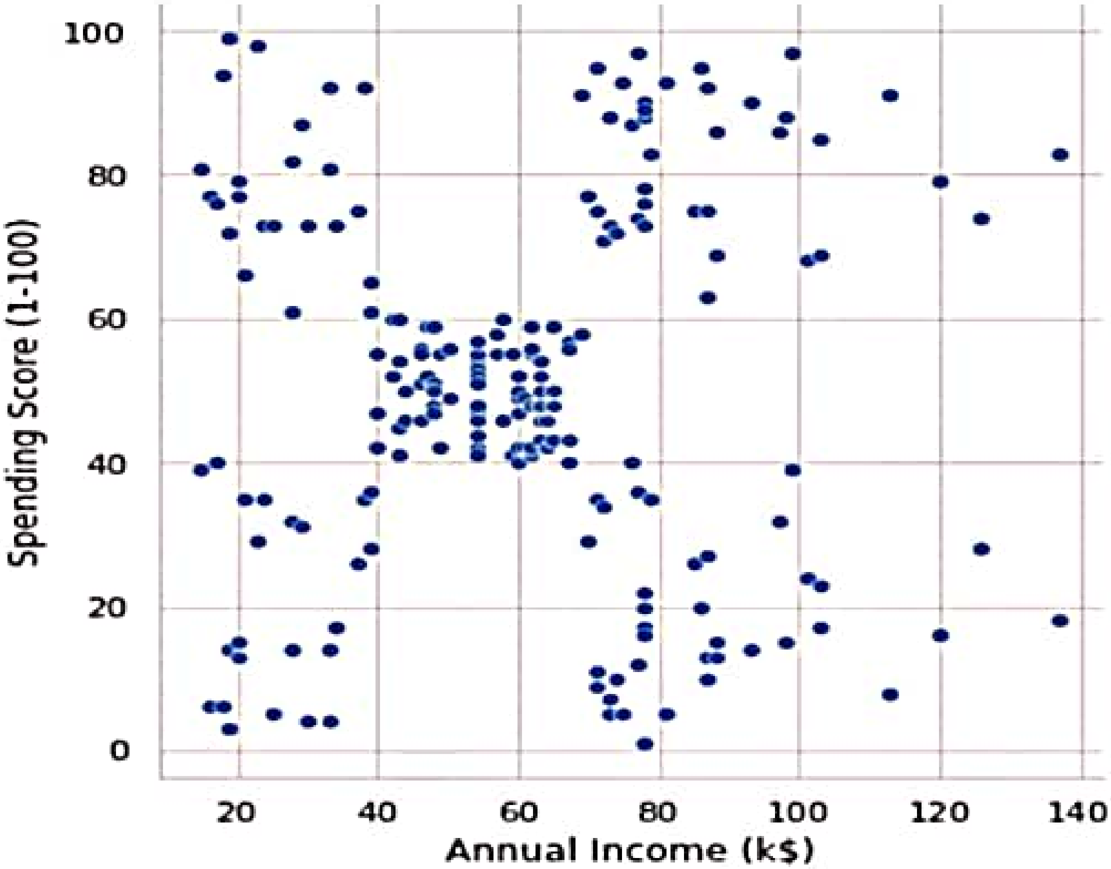
In[2]:



```
sns.relplot(x="Annual Income (k$)",y = "Sp
ending Score (1-100)",data=df)
```

Out[2]:

```
<seaborn.axisgrid.FacetGrid at 0x7fcef053
6fd0>
```



Spending Score (1-100)
Annual Income (k$)