In [1]:

```
import pandas as pd

df = pd.read_csv("ecommerce.csv")
df.head()
```

Out[1]:

	Customer ID	Gender	Age	City	Membership Type	Total Spend	Items Purchased	Average Rating	Disco App
0	101	Female	29	New York	Gold	1120.20	14	4.6	
1	102	Male	34	Los Angeles	Silver	780.50	11	4.1	F
2	103	Female	43	Chicago	Bronze	510.75	9	3.4	
3	104	Male	30	San Francisco	Gold	1480.30	19	4.7	F
4	105	Male	27	Miami	Silver	720.40	13	4.0	

In [2]:

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

df.fillna(df.median(numeric_only=True), inplace=True)
df.fillna(df.select_dtypes(include='object').mode().iloc[0], inplace=True)
```

```
Customer ID                 0
Gender                      0
Age                         0
City                        0
Membership Type             0
Total Spend                 0
Items Purchased             0
Average Rating              0
Discount Applied            0
Days Since Last Purchase    0
Satisfaction Level          2
dtype: int64
```

In [3]:

```
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
categorical_cols = df.select_dtypes(include=['object']).columns

print("Numerical columns:", numerical_cols.tolist())
print("Categorical columns:", categorical_cols.tolist())
```

```
Numerical columns: ['Customer ID', 'Age', 'Total Spend', 'Items Purchased', 'Aver
age Rating', 'Days Since Last Purchase']
Categorical columns: ['Gender', 'City', 'Membership Type', 'Satisfaction Level']
```

In [4]:

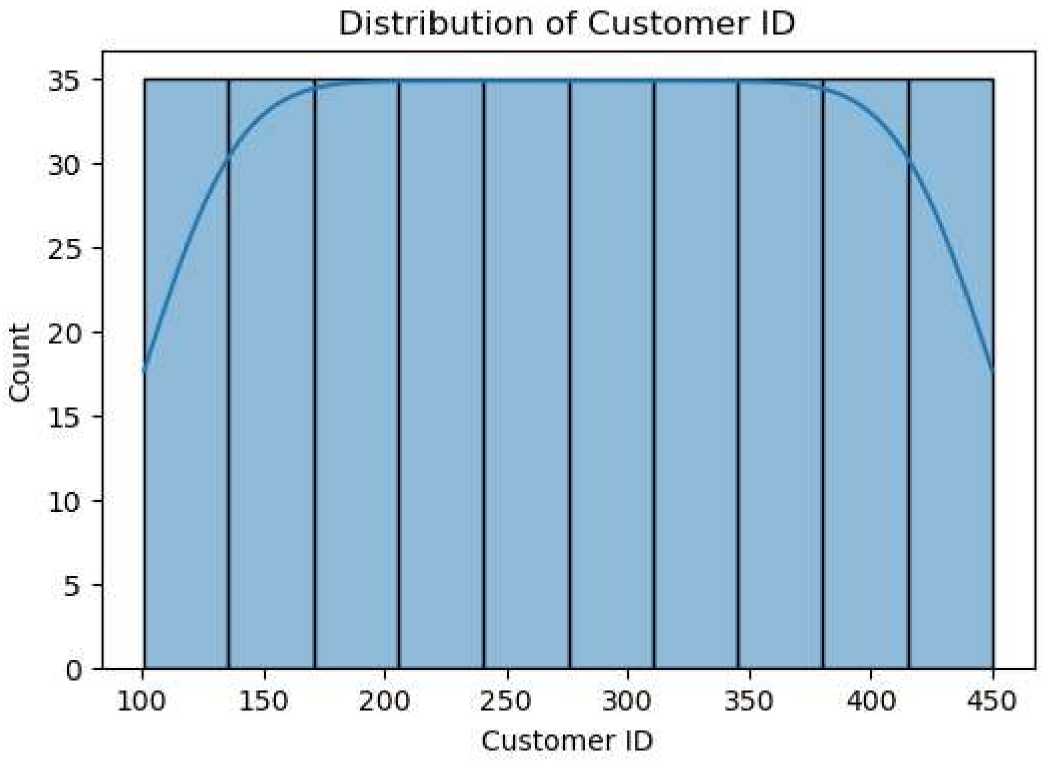
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

df_scaled = df.copy()
df_scaled[numerical_cols] = scaler.fit_transform(df[numerical_cols])
df_scaled.head()
```

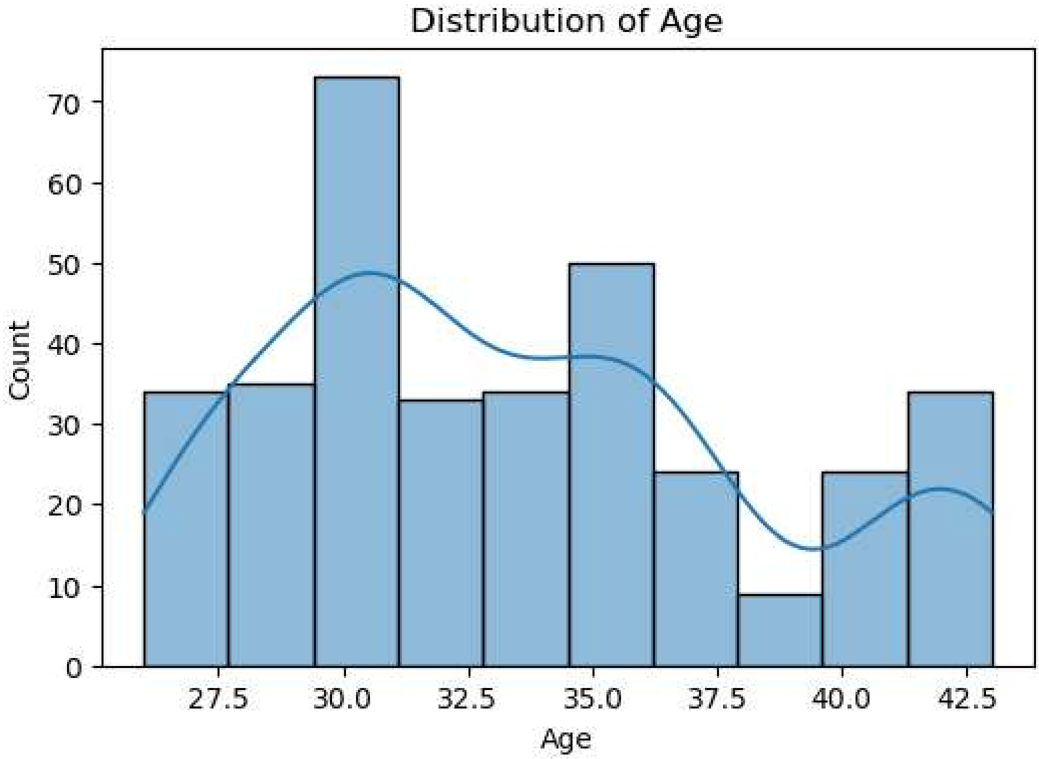
Out[4]:

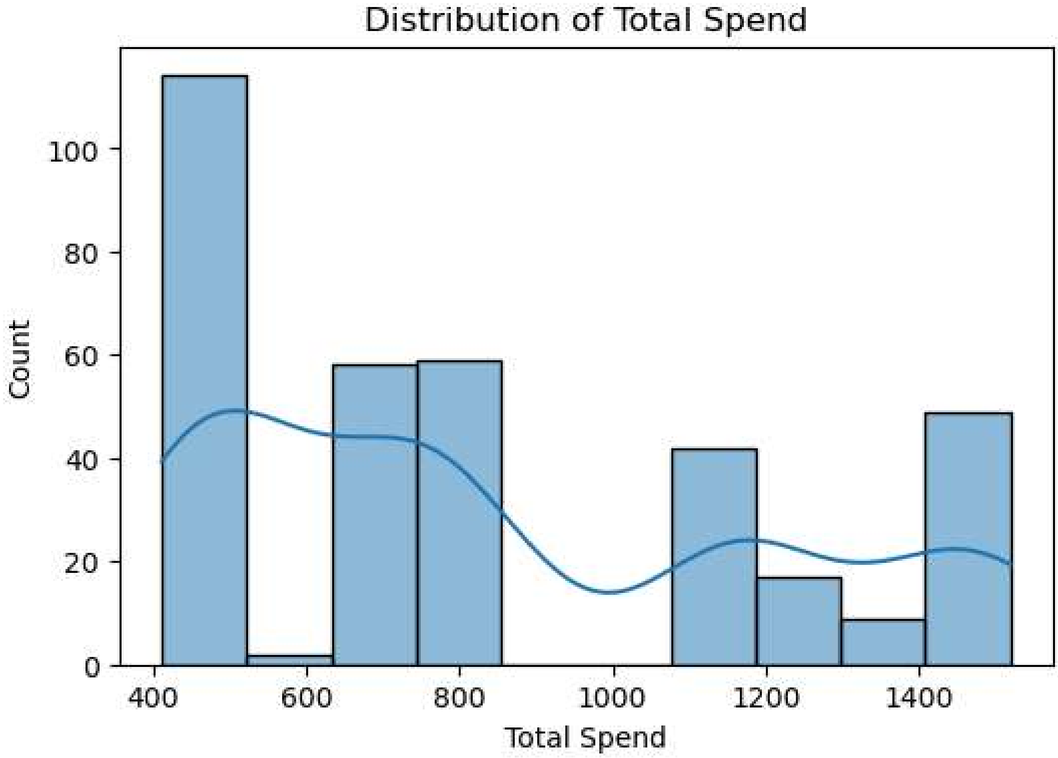
	Customer ID	Gender	Age	City	Membership Type	Total Spend	Items Purchased	Averag Ratin
0	-1.727109	Female	-0.945152	New York	Gold	0.760130	0.337346	1.00198
1	-1.717212	Male	0.082826	Los Angeles	Silver	-0.179459	-0.385538	0.13947
2	-1.707314	Female	1.933185	Chicago	Bronze	-0.925570	-0.867461	-1.06802
3	-1.697417	Male	-0.739557	San Francisco	Gold	1.756144	1.542153	1.17448
4	-1.687519	Male	-1.356343	Miami	Silver	-0.345692	0.096385	-0.03302

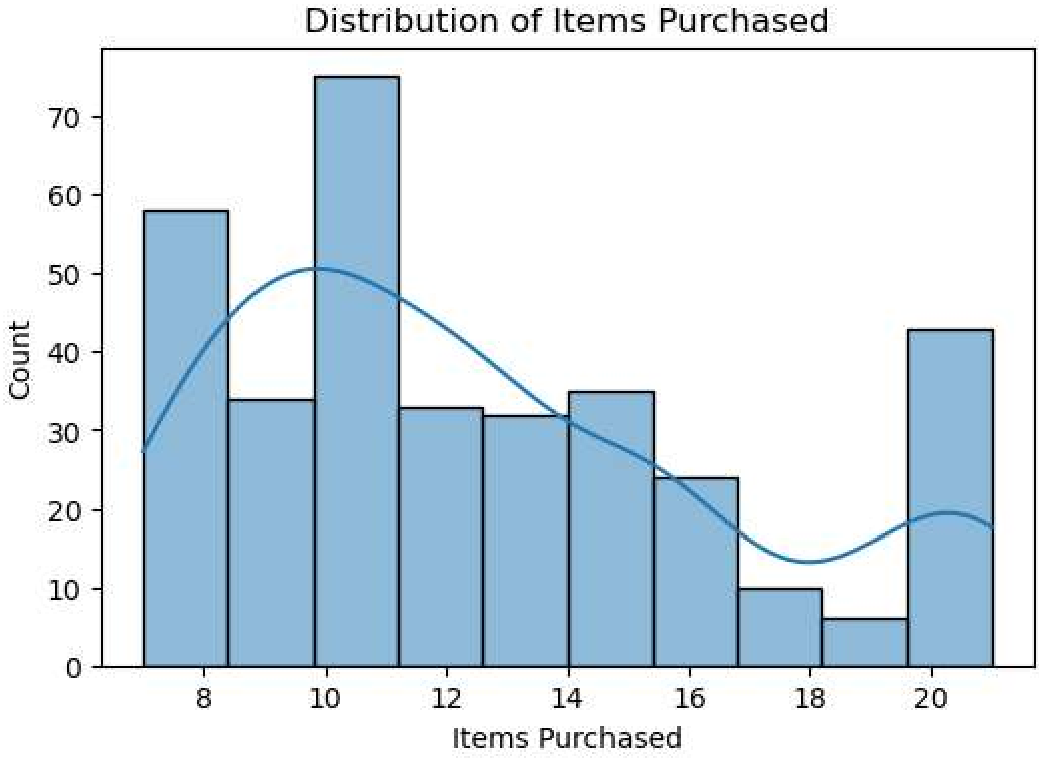
In [5]:

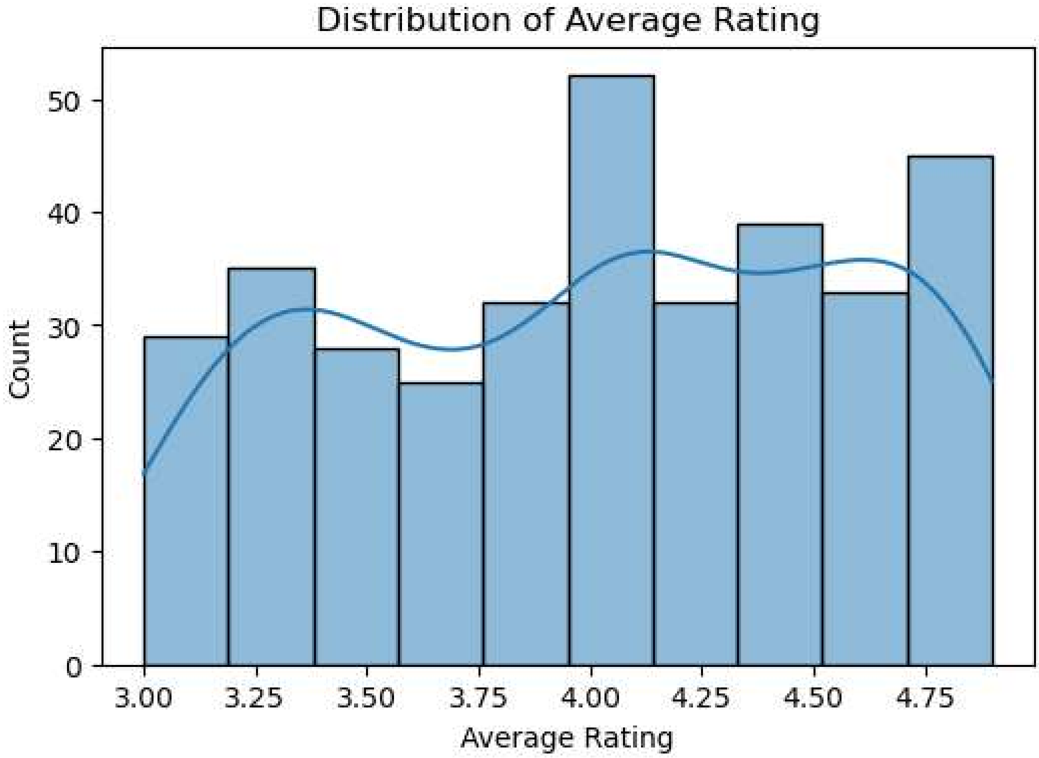
```
import matplotlib.pyplot as plt
import seaborn as sns

for col in numerical_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(df[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.show()
```

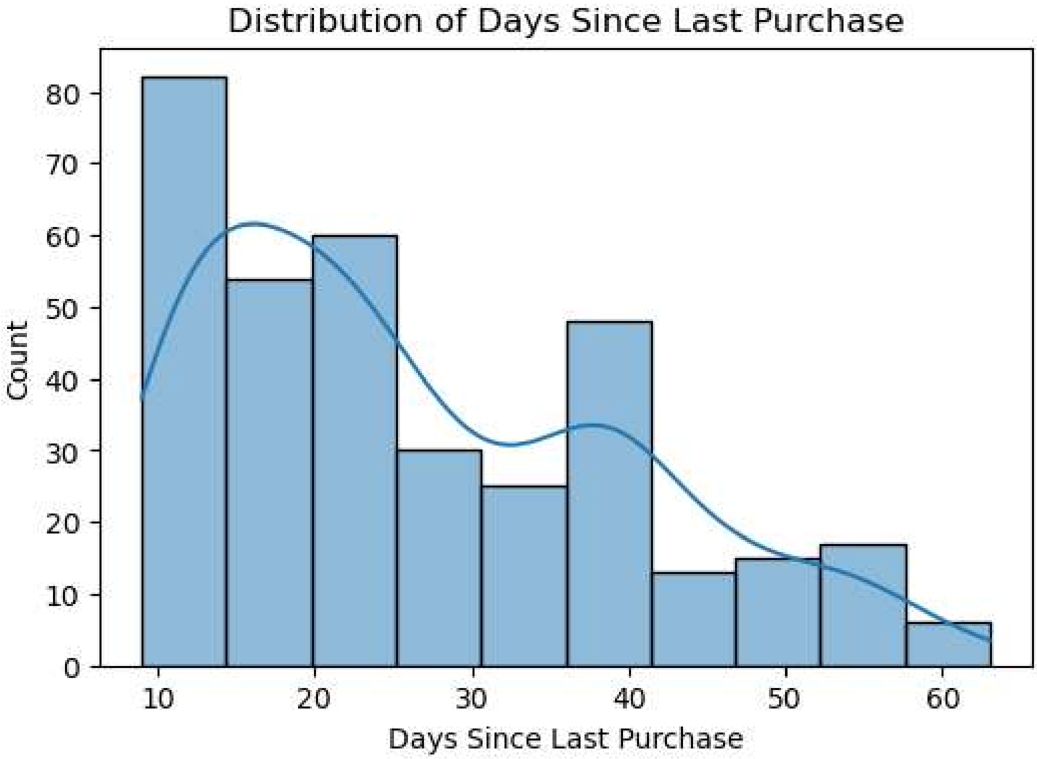



Distribution of Age
Count
0
10
20
30
40
50
60
70
27.5
30.0
32.5
35.0
37.5
40.0
42.5
Age

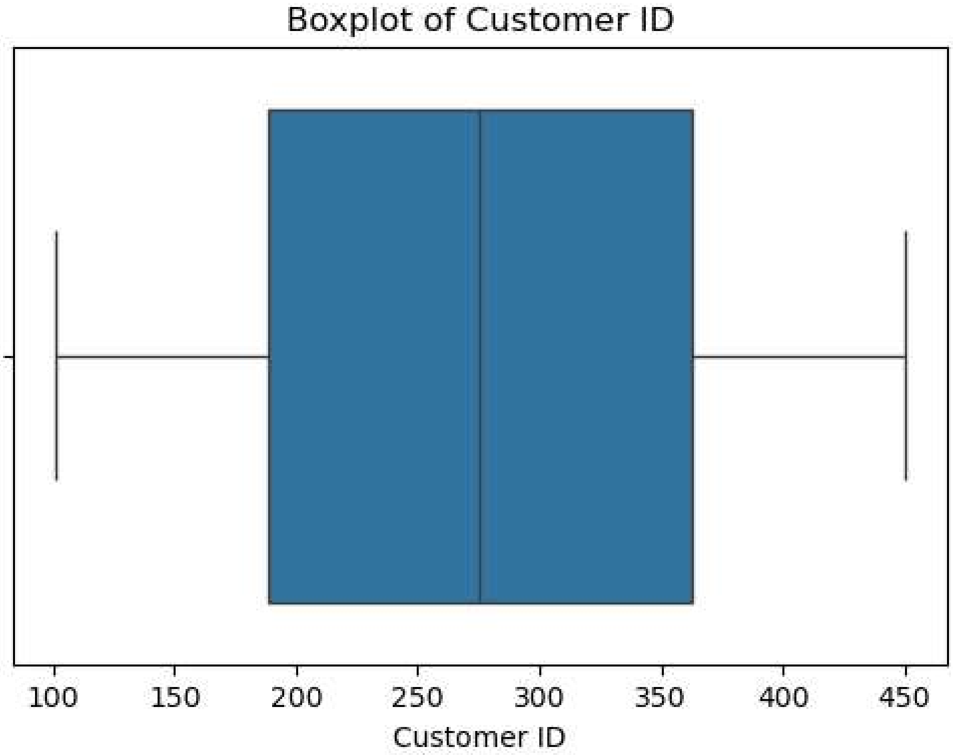

Distribution of Total Spend
Count
0
20
40
60
80
100
400
600
800
1000
1200
1400
Total Spend

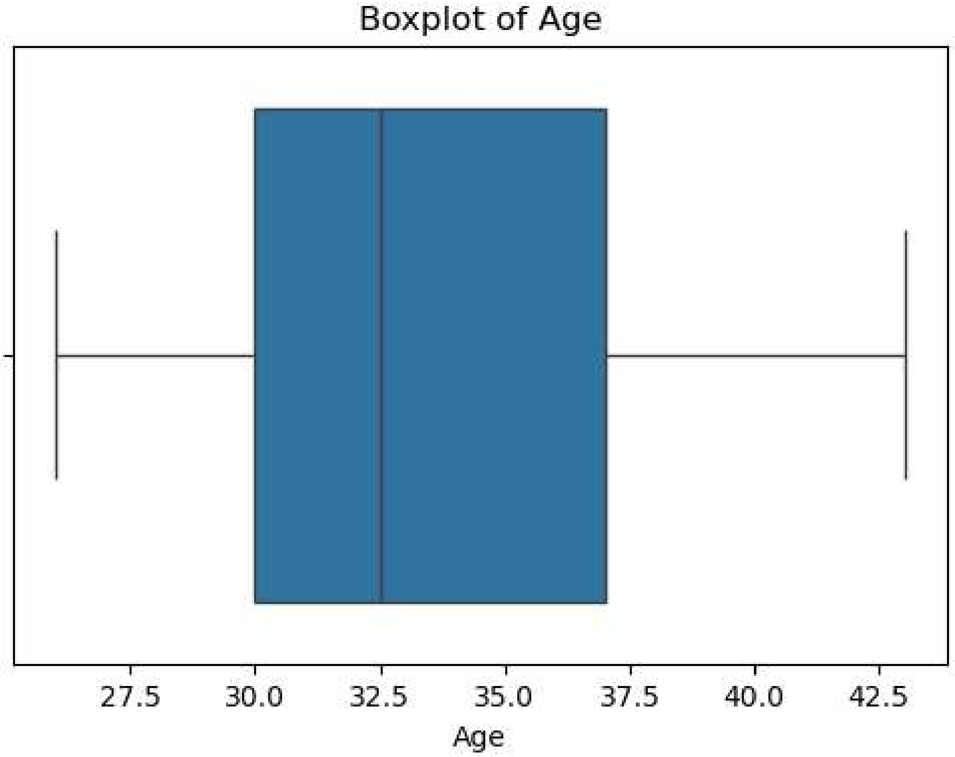

Distribution of Items Purchased
Count
0
10
20
30
40
50
60
70
8
10
12
14
16
18
20
Items Purchased

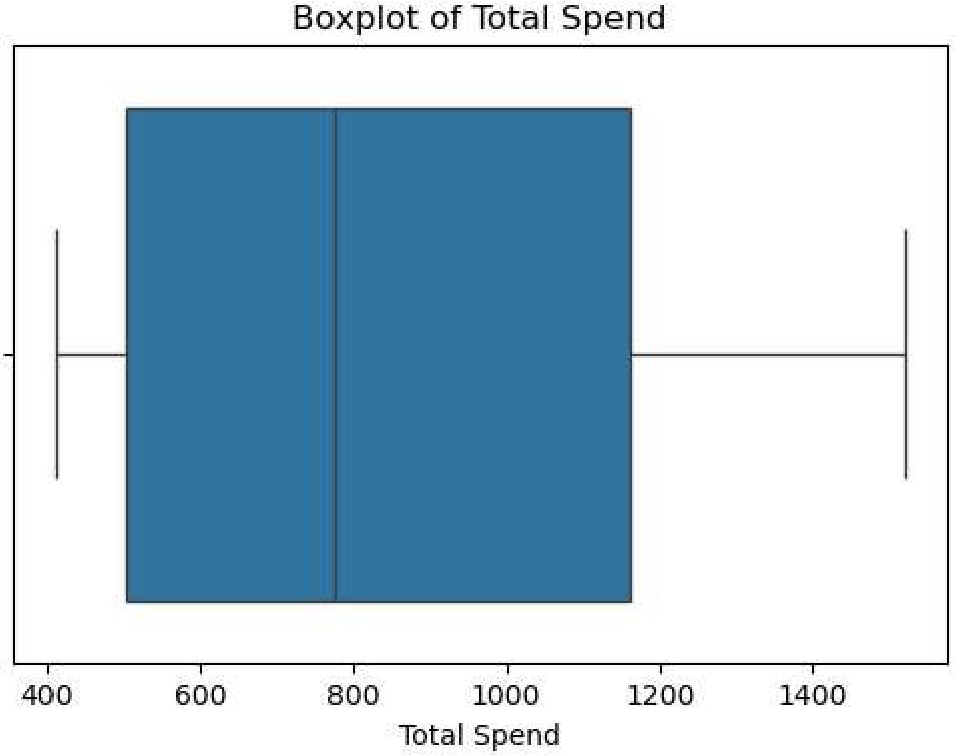

Distribution of Average Rating
Count
0
10
20
30
40
50
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
Average Rating

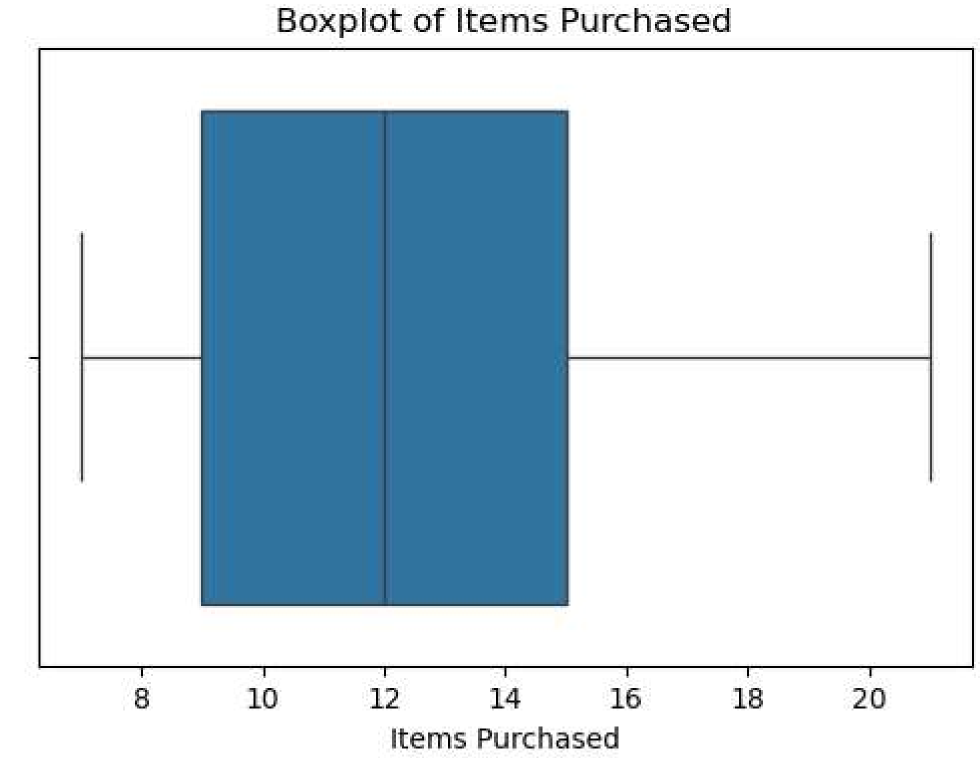


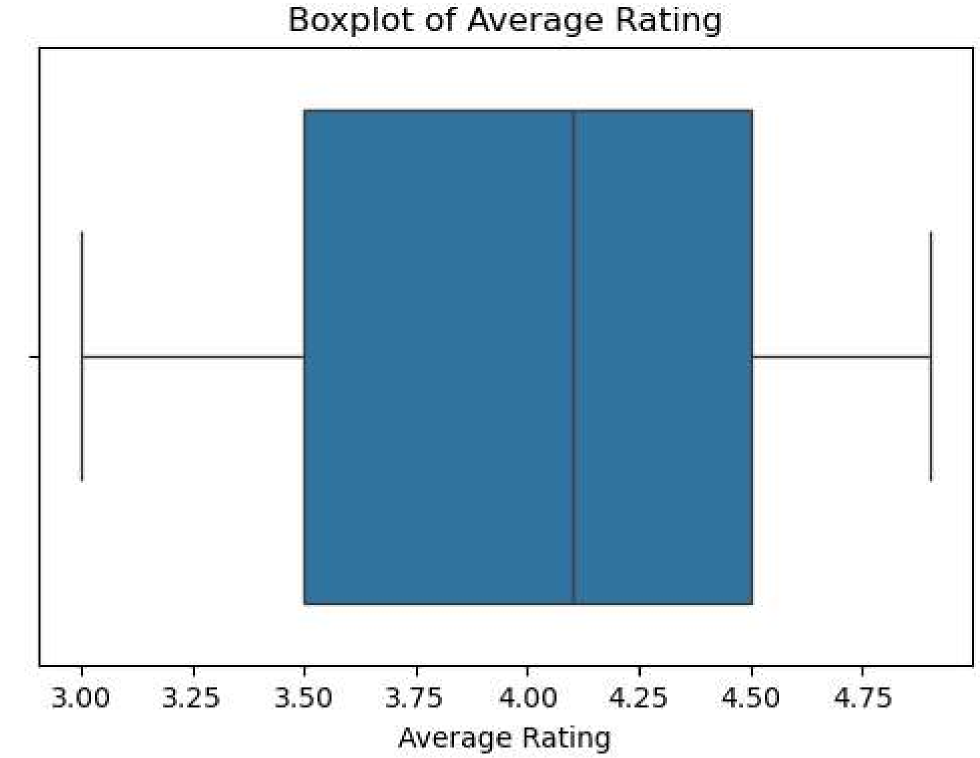
In [7]:
```
for col in numerical_cols:
    plt.figure(figsize=(6, 4))
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot of {col}')
    plt.show()
```

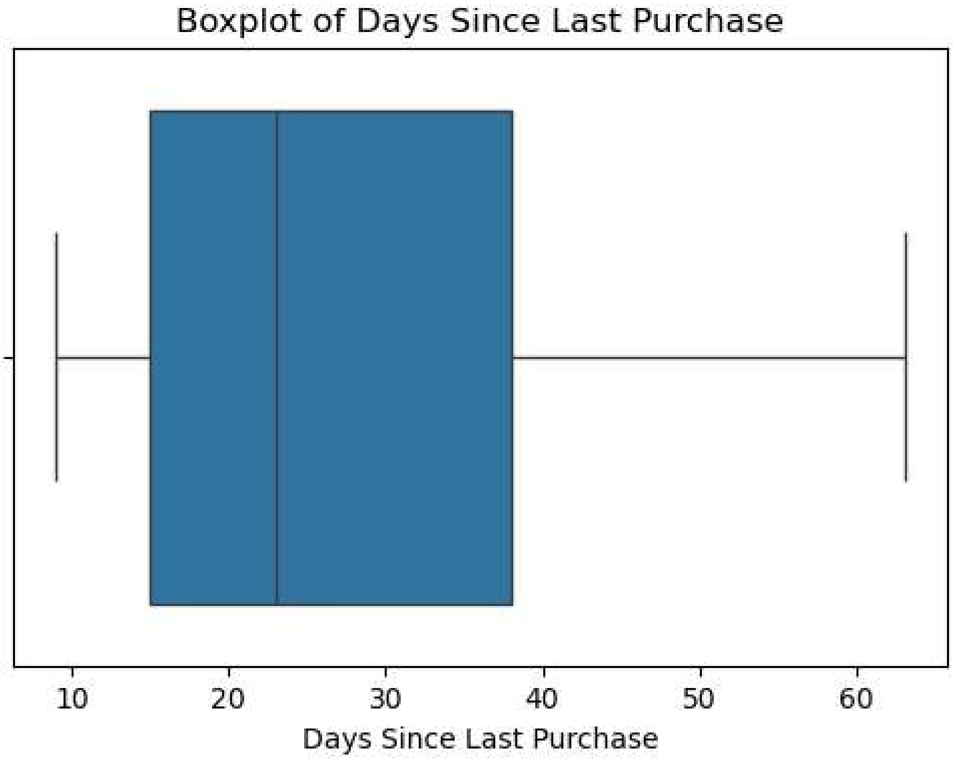



Boxplot of Age
27.5
30.0
32.5
35.0
37.5
40.0
42.5
Age


Boxplot of Total Spend
400
600
800
1000
1200
1400
Total Spend

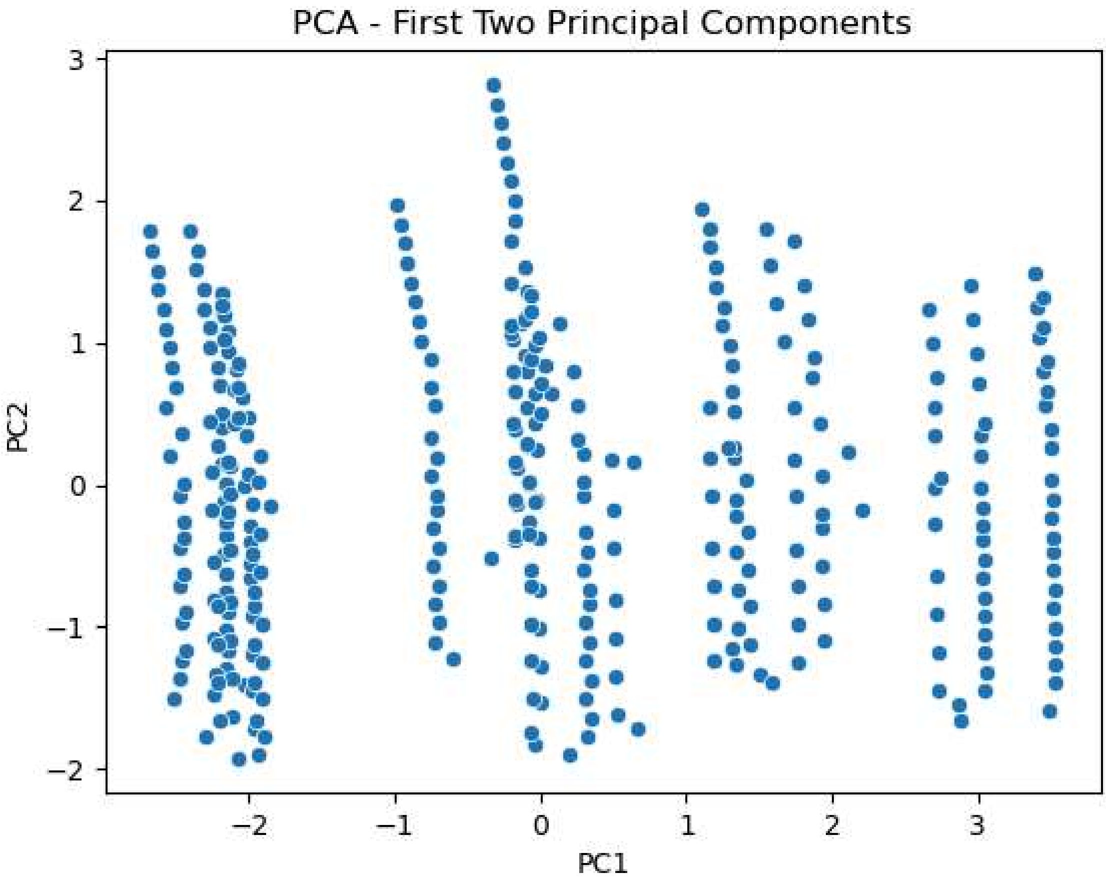

Boxplot of Items Purchased
8
10
12
14
16
18
20
Items Purchased


Boxplot of Average Rating
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
Average Rating



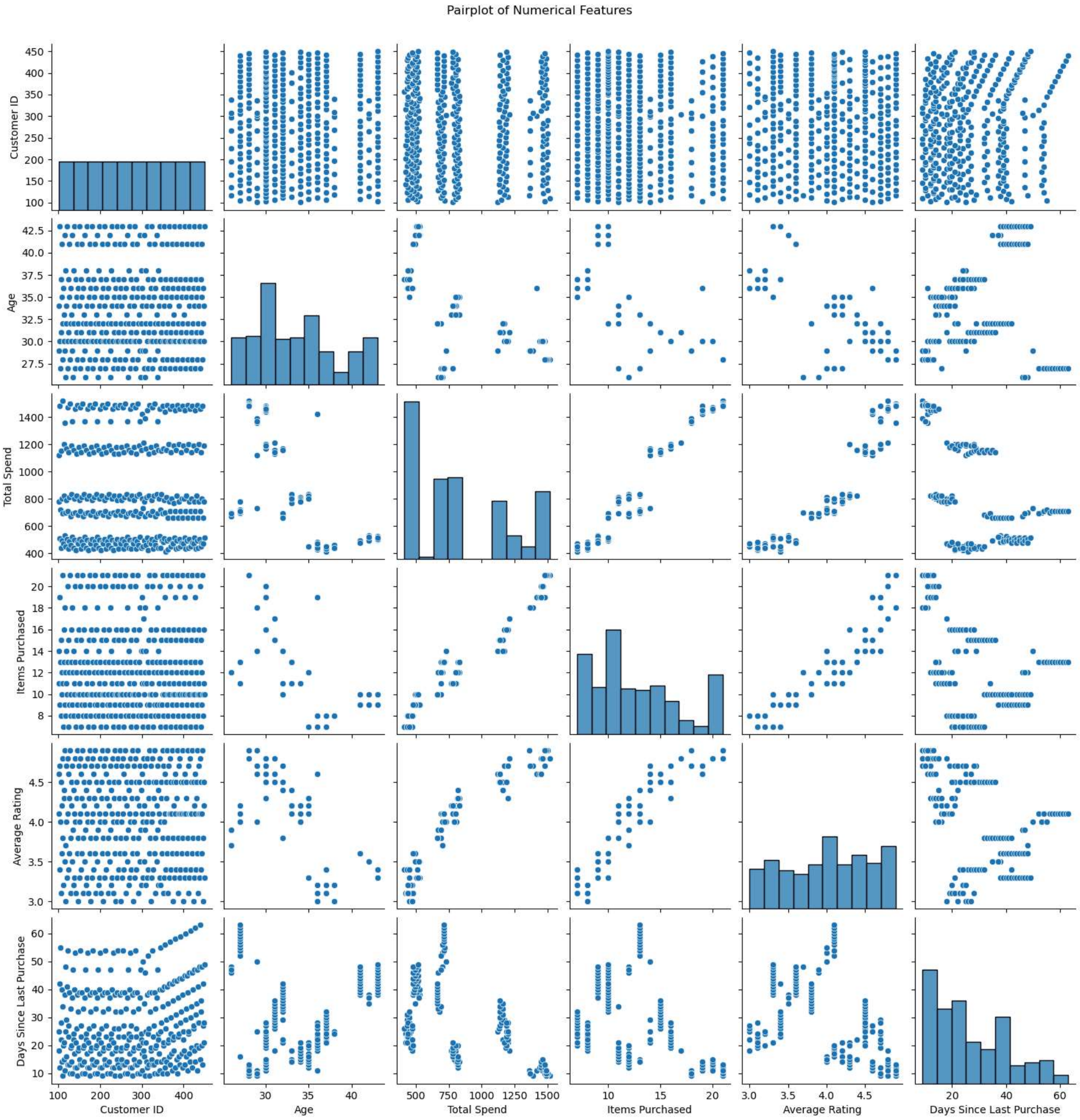
In [8]:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
pca_result = pca.fit_transform(df_scaled[numerical_cols])

df_pca = pd.DataFrame(pca_result, columns=['PC1', 'PC2'])

sns.scatterplot(x='PC1', y='PC2', data=df_pca)
plt.title('PCA - First Two Principal Components')
plt.show()
```

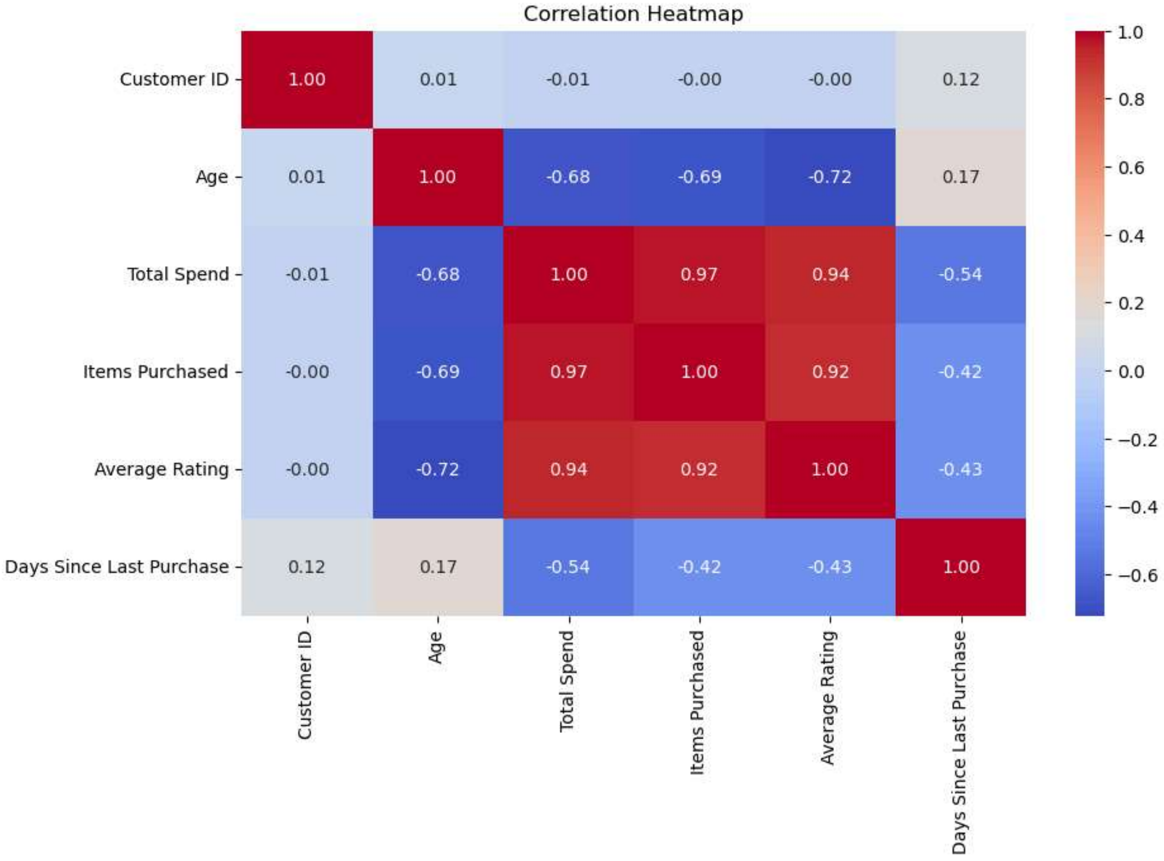


In [10]:

```
sns.pairplot(df[numerical_cols])
plt.suptitle("Pairplot of Numerical Features", y=1.02)
plt.show()
```

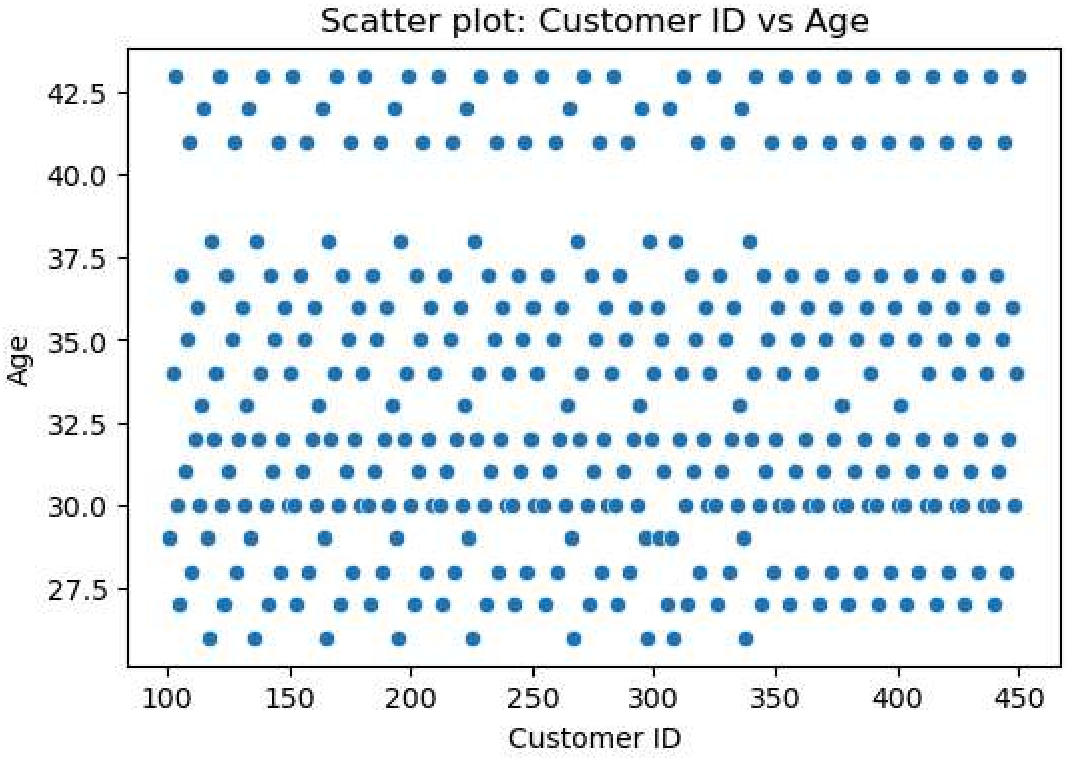


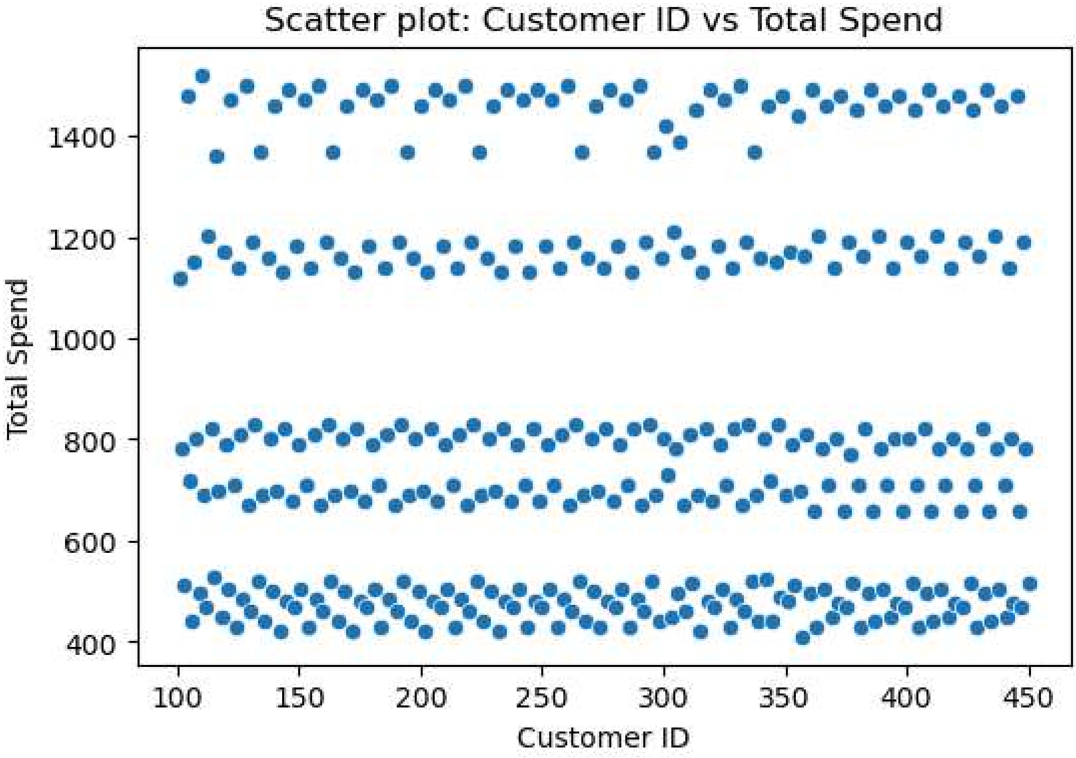
In [11]:
```
plt.figure(figsize=(10, 6))
sns.heatmap(df[numerical_cols].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

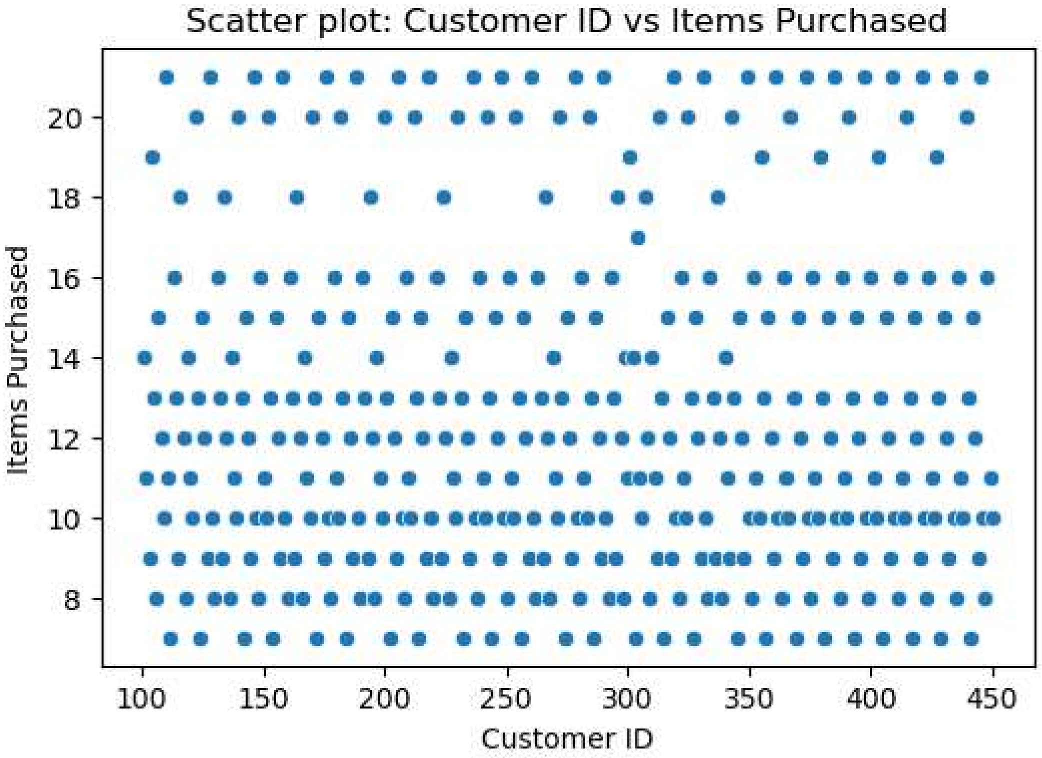


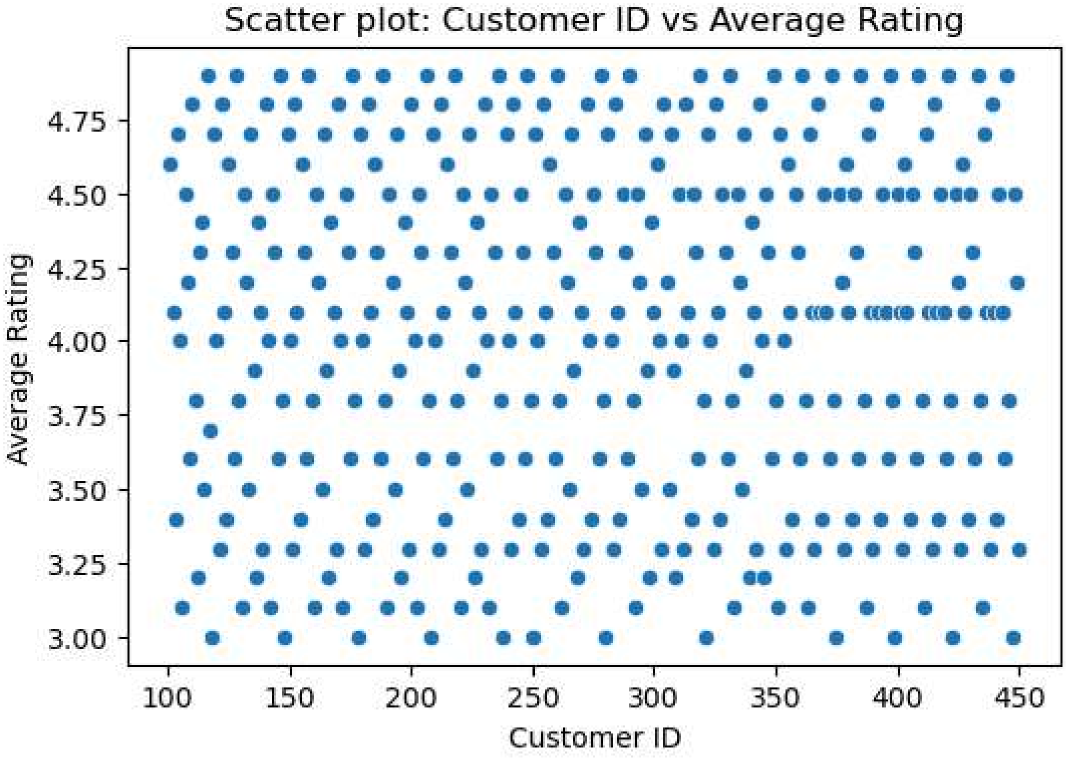
In [13]:

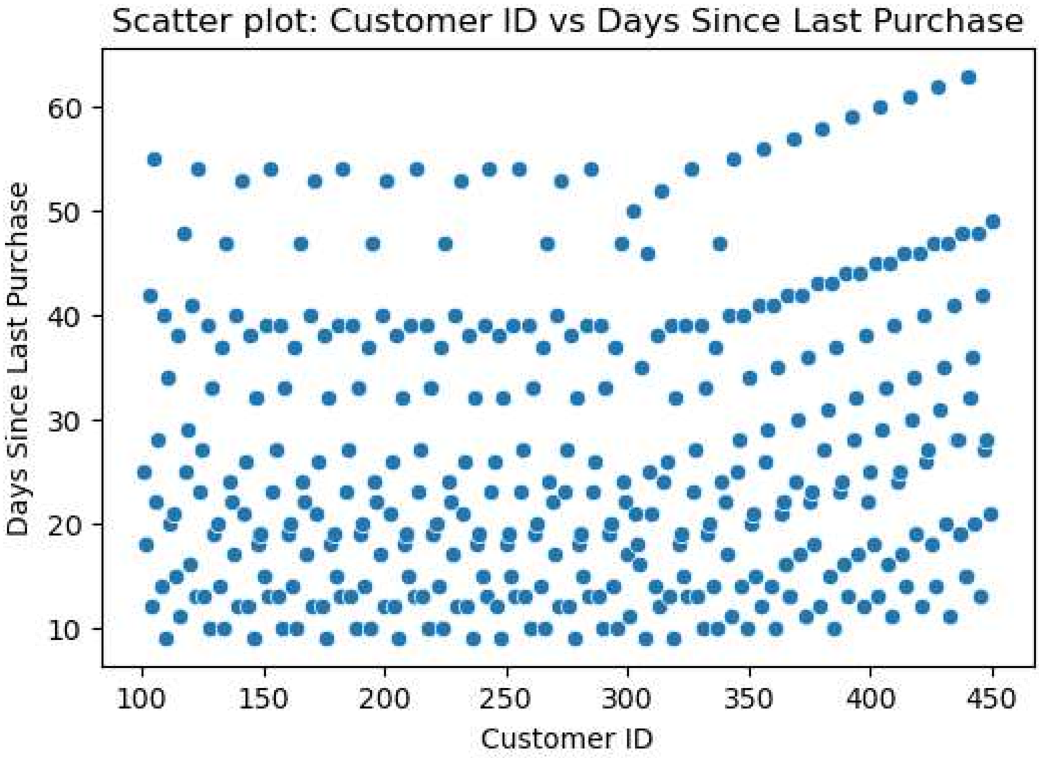
```
import itertools
import matplotlib.pyplot as plt
import seaborn as sns
numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns
col_pairs = list(itertools.combinations(numeric_cols, 2))

for x_col, y_col in col_pairs:
    plt.figure(figsize=(6, 4))
    sns.scatterplot(data=df, x=x_col, y=y_col)
    plt.title(f'Scatter plot: {x_col} vs {y_col}')
    plt.show()
```

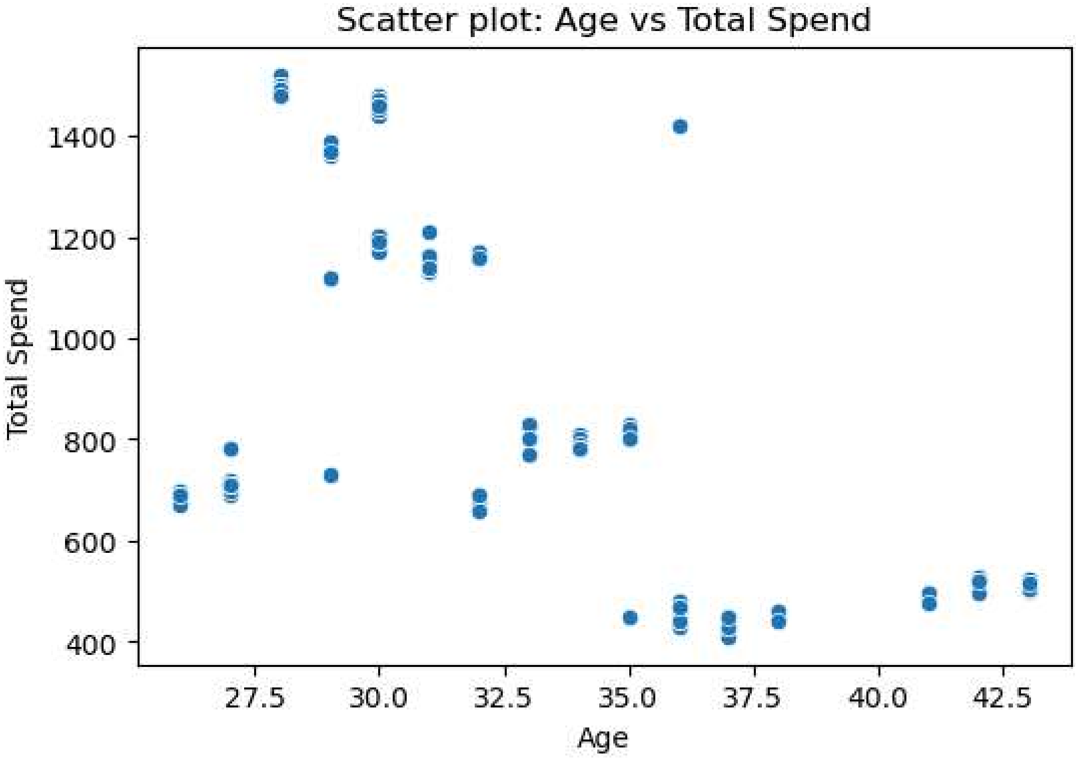

Scatter plot: Customer ID vs Age
Age
42.5
40.0
37.5
35.0
32.5
30.0
27.5
100
150
200
250
300
350
400
450
Customer ID

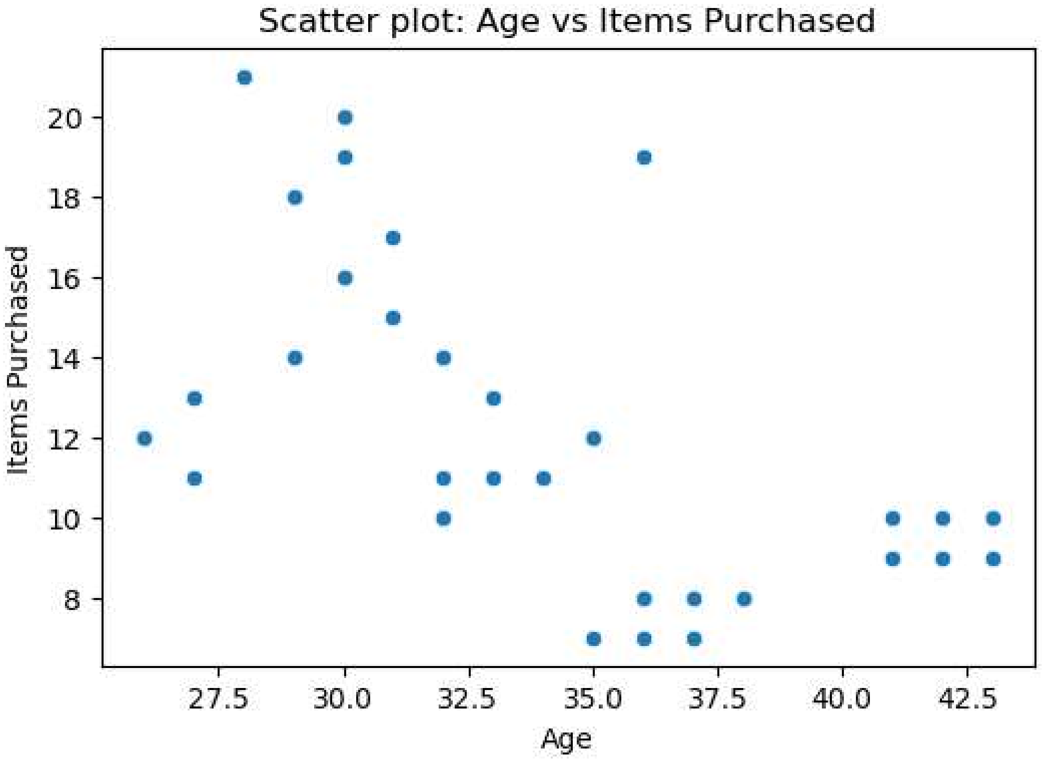

Scatter plot: Customer ID vs Total Spend
Total Spend
1400
1200
1000
800
600
400
100
150
200
250
300
350
400
450
Customer ID

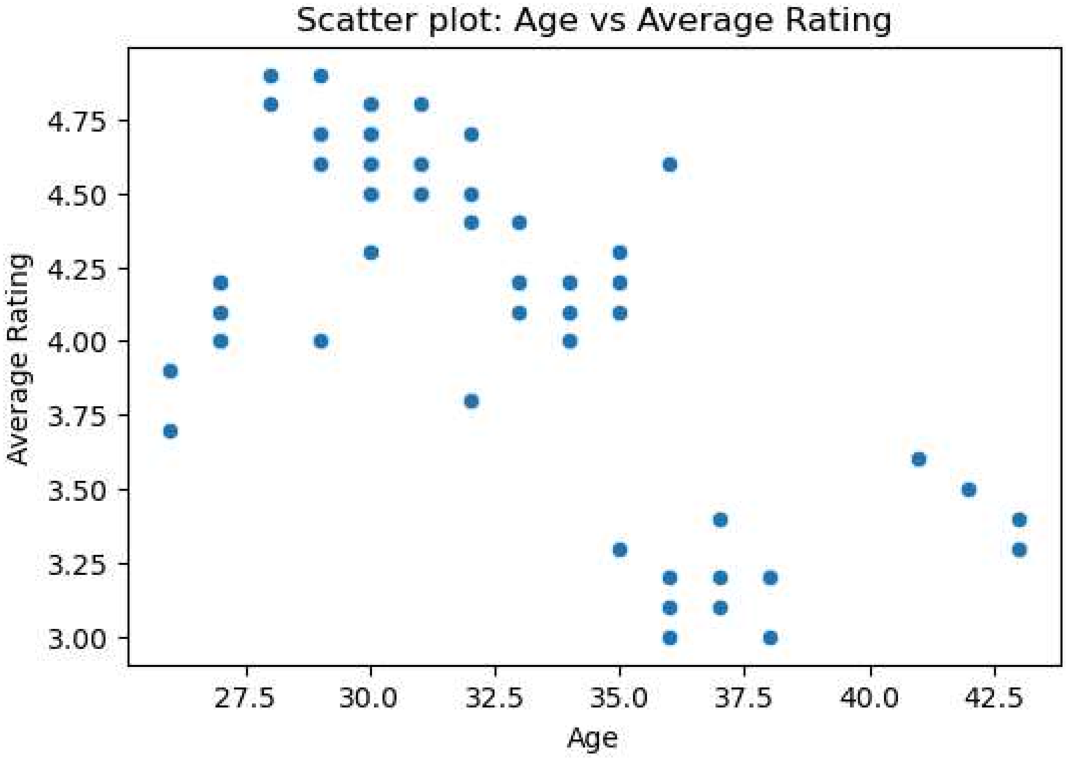

Scatter plot: Customer ID vs Items Purchased
Items Purchased
8
10
12
14
16
18
20
100
150
200
250
300
350
400
450
Customer ID

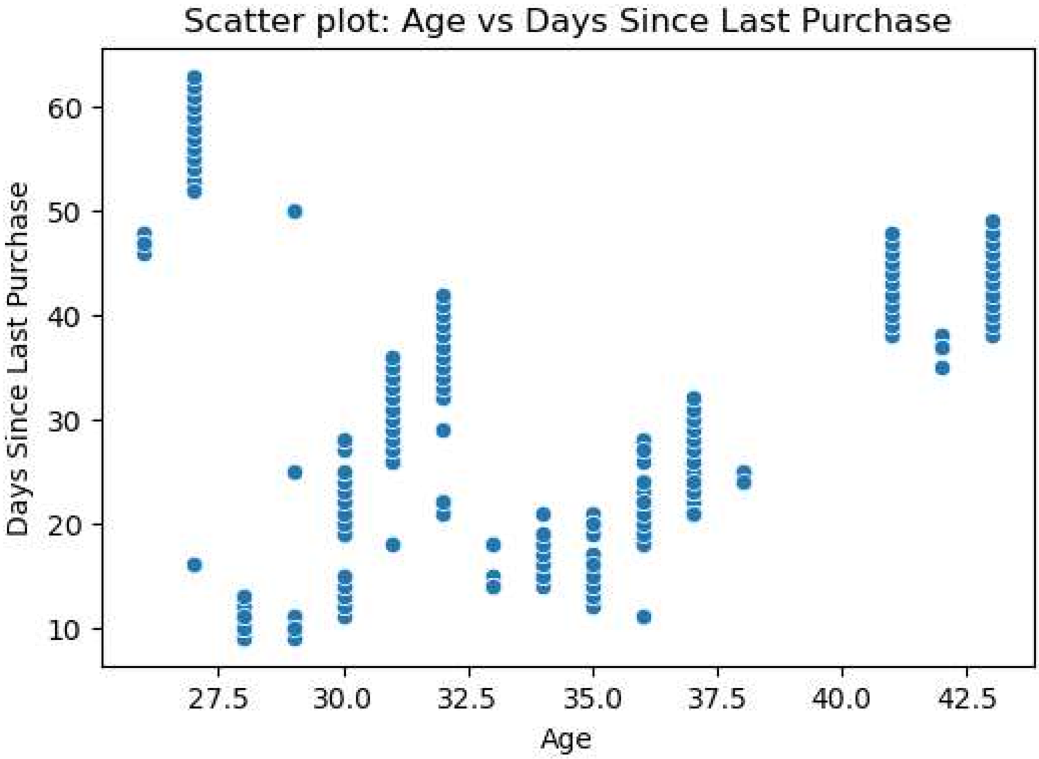

Scatter plot: Customer ID vs Average Rating
Average Rating
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
100
150
200
250
300
350
400
450
Customer ID

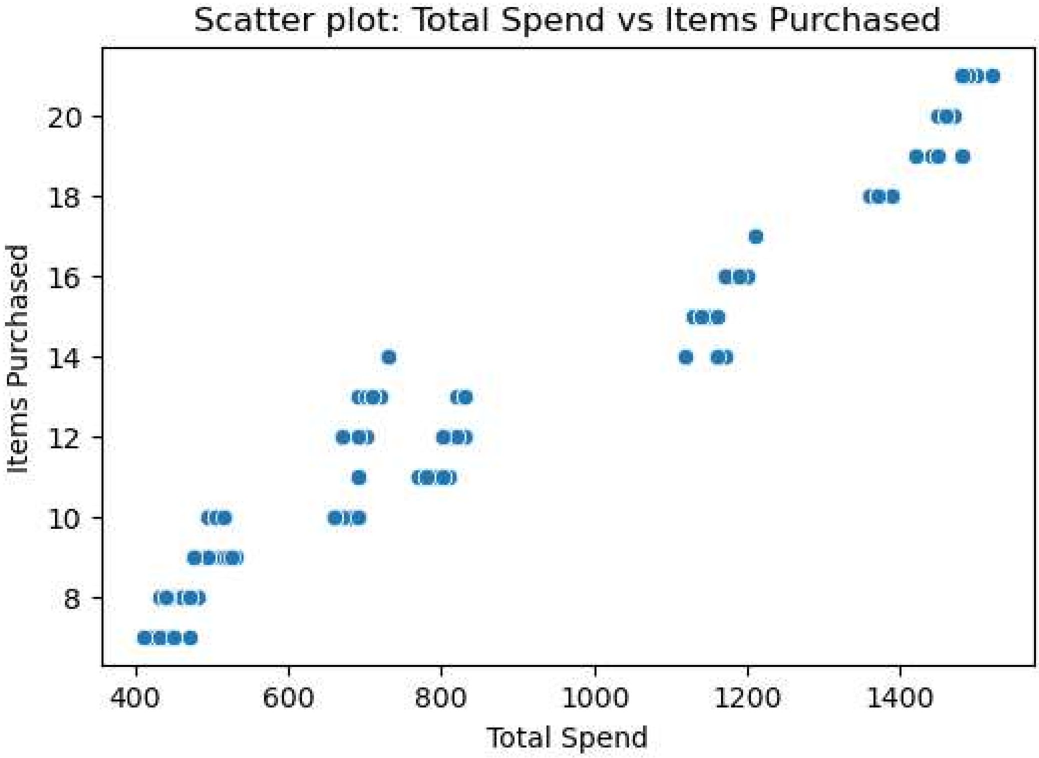

Scatter plot: Customer ID vs Days Since Last Purchase
Days Since Last Purchase
10
20
30
40
50
60
100
150
200
250
300
350
400
450
Customer ID

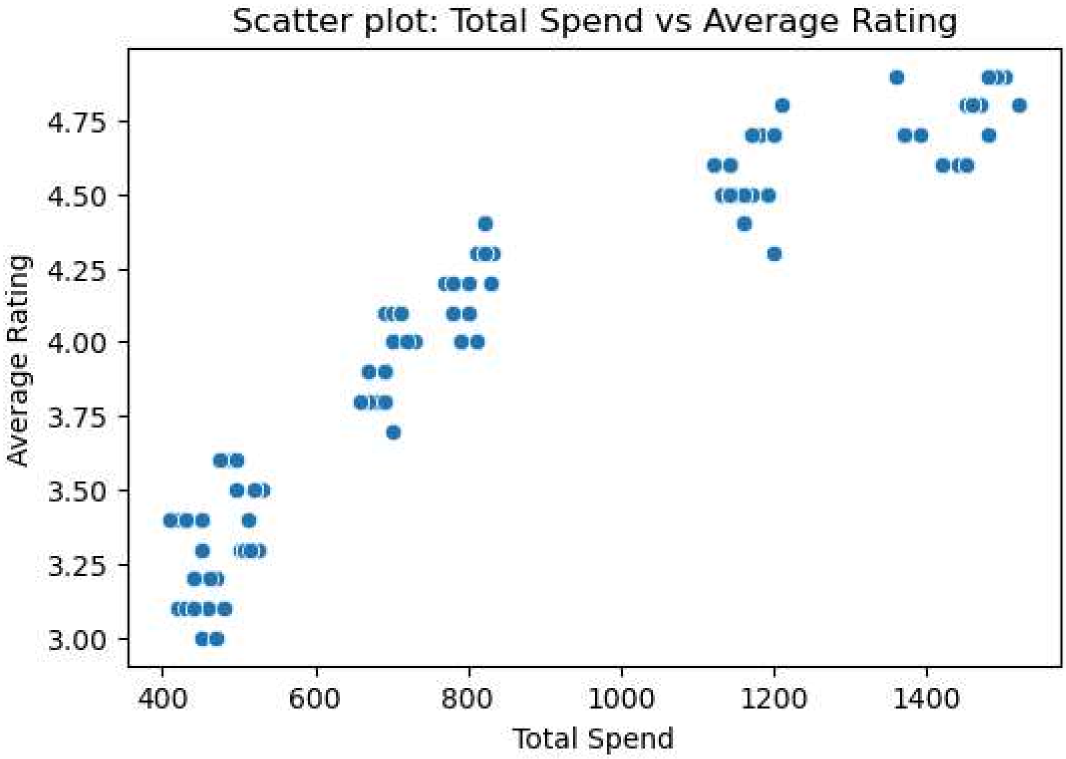

Scatter plot: Age vs Total Spend
Total Spend
400
600
800
1000
1200
1400
27.5
30.0
32.5
35.0
37.5
40.0
42.5
Age

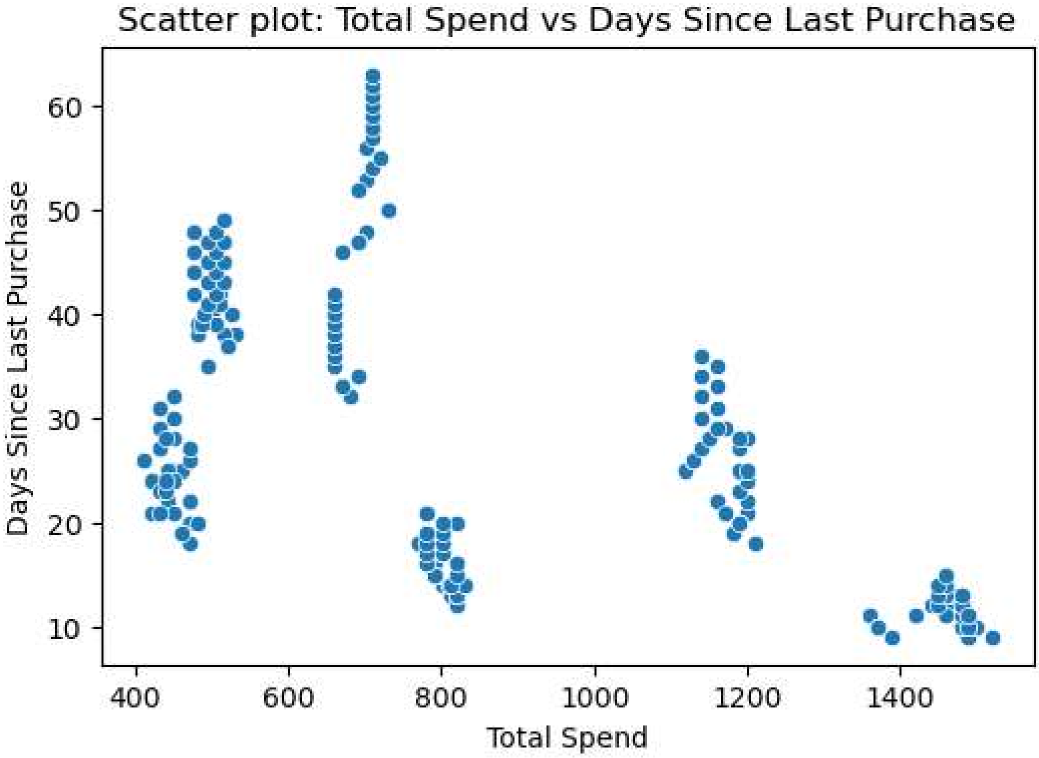

Scatter plot: Age vs Items Purchased
Items Purchased
Age
8
10
12
14
16
18
20
27.5
30.0
32.5
35.0
37.5
40.0
42.5

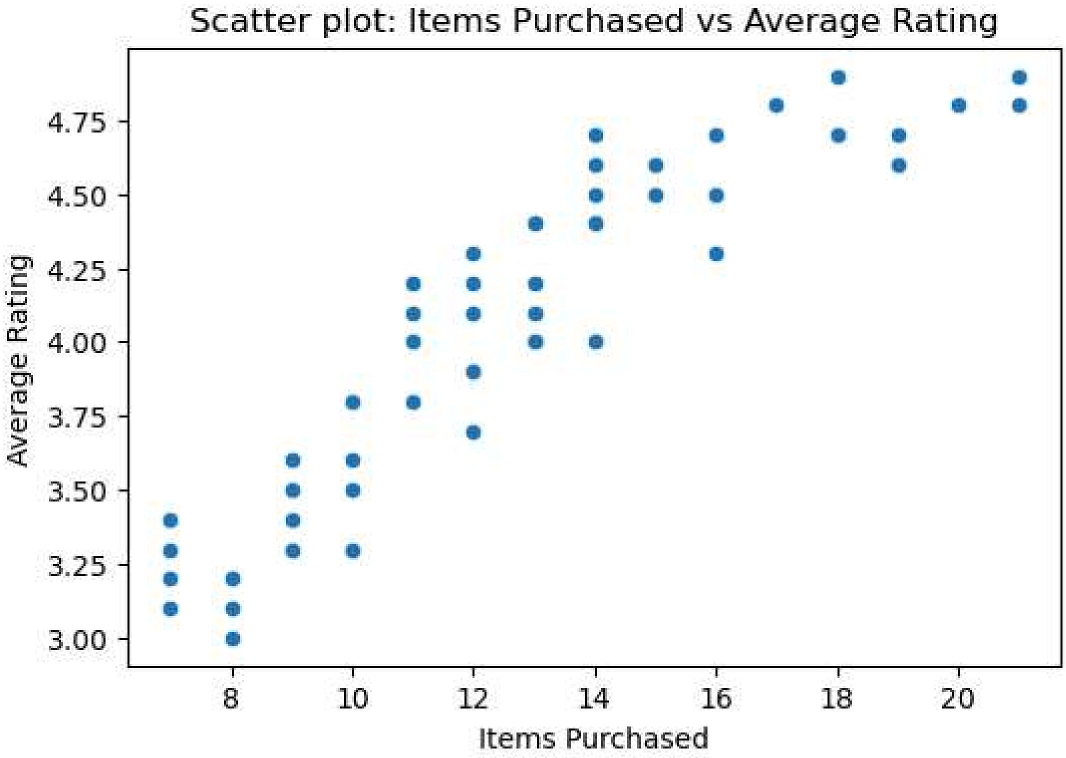

Scatter plot: Age vs Average Rating
Average Rating
Age
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
27.5
30.0
32.5
35.0
37.5
40.0
42.5

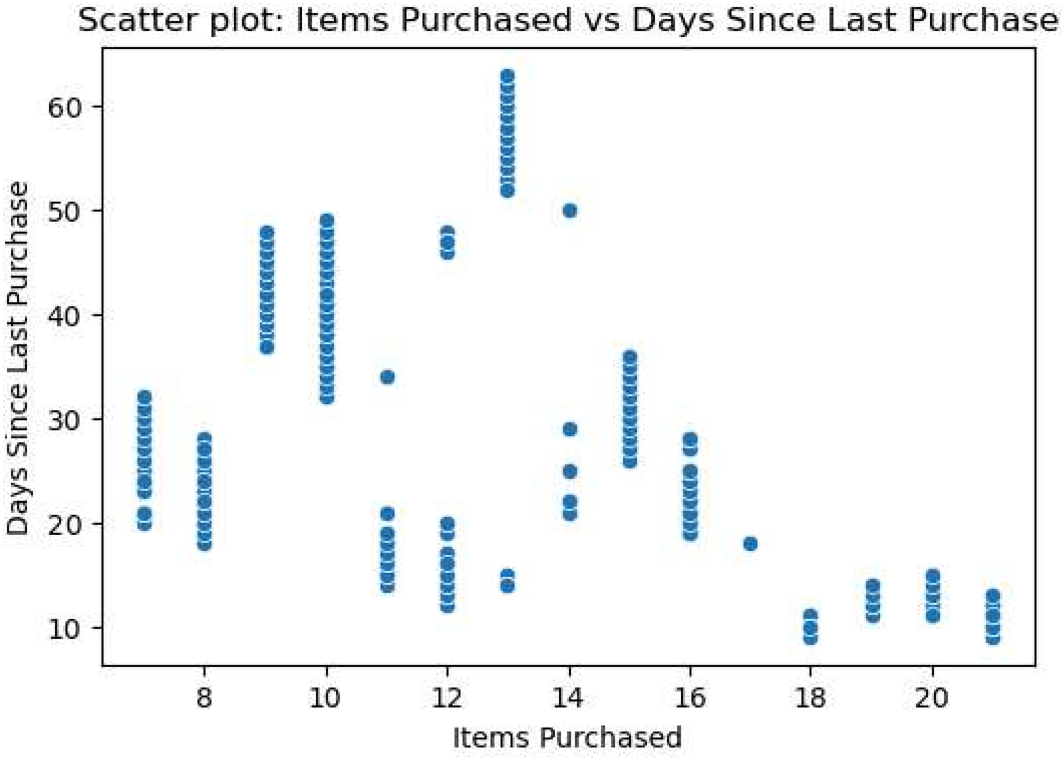

Scatter plot: Age vs Days Since Last Purchase
Days Since Last Purchase
10
20
30
40
50
60
27.5
30.0
32.5
35.0
37.5
40.0
42.5
Age

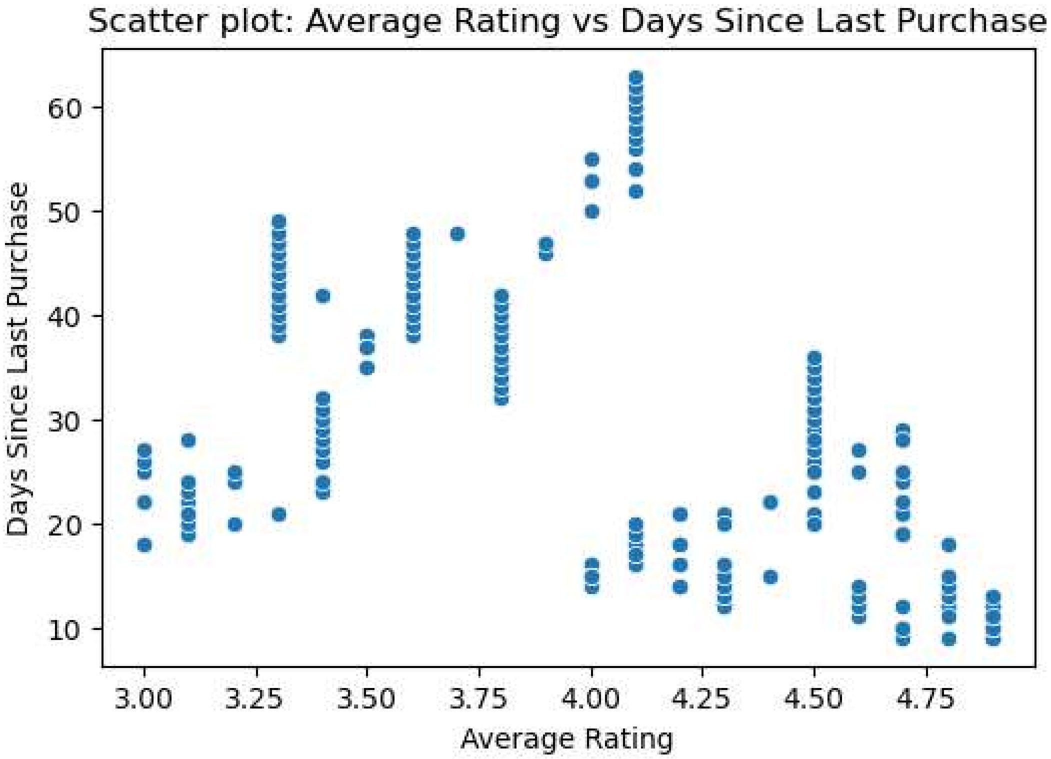

Scatter plot: Total Spend vs Items Purchased
Items Purchased
8
10
12
14
16
18
20
400
600
800
1000
1200
1400
Total Spend


Scatter plot: Total Spend vs Average Rating
Average Rating
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
400
600
800
1000
1200
1400
Total Spend


Scatter plot: Total Spend vs Days Since Last Purchase
Days Since Last Purchase
60
50
40
30
20
10
400
600
800
1000
1200
1400
Total Spend


Scatter plot: Items Purchased vs Average Rating
Average Rating
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
8
10
12
14
16
18
20
Items Purchased


Scatter plot: Items Purchased vs Days Since Last Purchase
Days Since Last Purchase
60
50
40
30
20
10
8
10
12
14
16
18
20
Items Purchased


Scatter plot: Average Rating vs Days Since Last Purchase
Days Since Last Purchase
10
20
30
40
50
60
3.00
3.25
3.50
3.75
4.00
4.25
4.50
4.75
Average Rating

In [ ]: