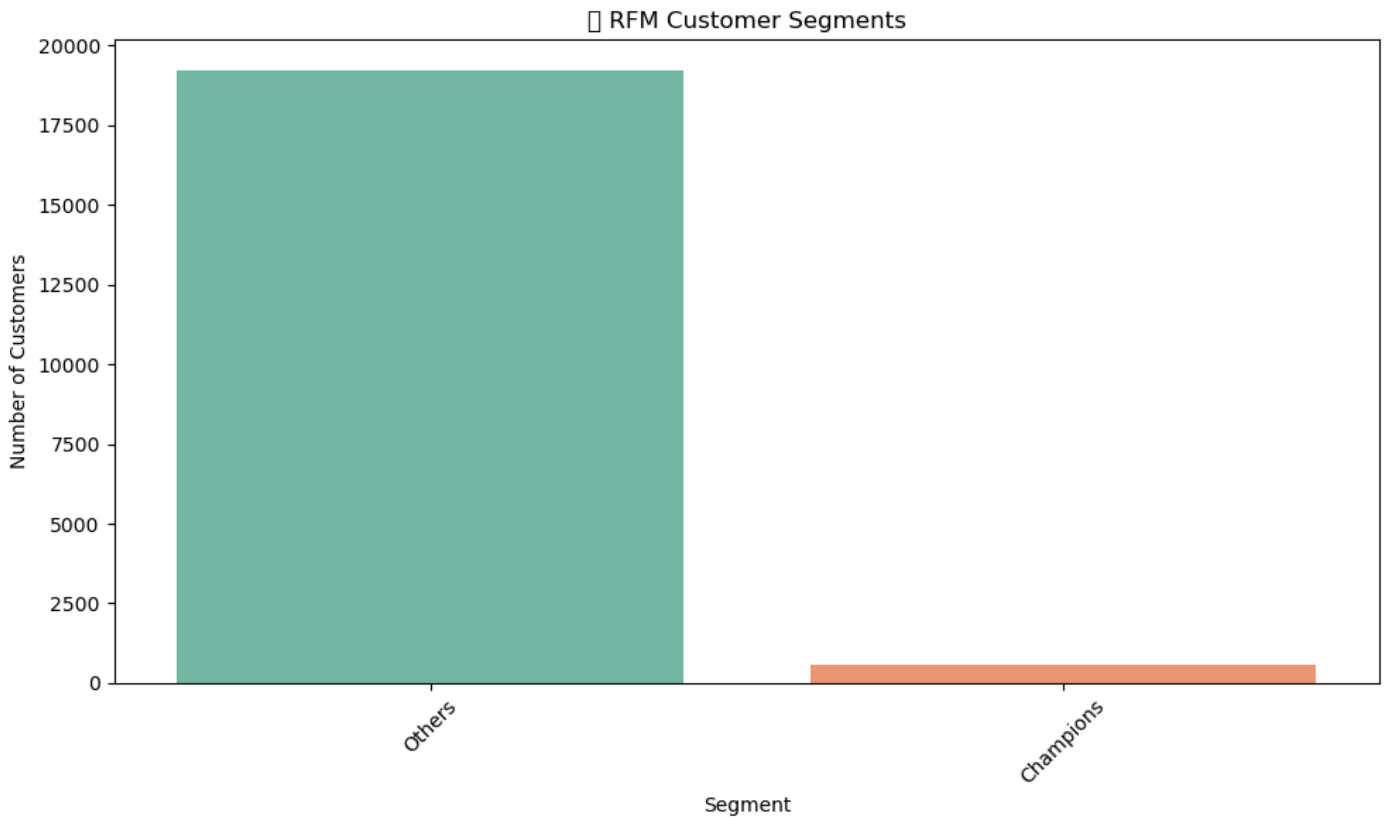
Report 3

Subtask 1: Calculate RFM Metrics

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

# Load your dataset
df = pd.read_csv("ecommerce_data_final_cleaned.csv")

# Rename columns for consistency
df = df.rename(columns={'value [USD]': 'total_revenue', 'date': 'order_date'})

# Convert date column to datetime
df['order_date'] = pd.to_datetime(df['order_date'], errors='coerce')

# Drop rows with missing customer_id or revenue or order_date
df = df.dropna(subset=['customer_id', 'order_date', 'total_revenue'])

# --------------------------------
# 1️⃣ Define a Reference Date
# --------------------------------
# Reference date = one day after the most recent purchase
ref_date = df['order_date'].max() + pd.Timedelta(days=1)

# --------------------------------
# 2️⃣ Group Data by Customer ID
# --------------------------------
rfm = df.groupby('customer_id').agg({
    'order_date': lambda x: (ref_date - x.max()).days,      # Recency
    'customer_id': 'count',                                 # Frequency
    'total_revenue': 'sum'                                  # Monetary
}).rename(columns={
    'order_date': 'Recency',
    'customer_id': 'Frequency',
```

```
    'total_revenue': 'Monetary'
}).reset_index()

# --------------------------------
# 3️⃣ Score Each RFM Metric
# --------------------------------
rfm['R_Score'] = pd.qcut(rfm['Recency'], 5, labels=[5, 4, 3, 2, 1])
rfm['F_Score'] = pd.qcut(rfm['Frequency'].rank(method='first'), 5, labels=[1, 2, 3, 4, 5])
rfm['M_Score'] = pd.qcut(rfm['Monetary'], 5, labels=[1, 2, 3, 4, 5])

# Combine RFM Score
rfm['RFM_Score'] = rfm['R_Score'].astype(str) + rfm['F_Score'].astype(str) + rfm['M_Score'].astype(str)

# --------------------------------
# 4️⃣ Segment Customers
# --------------------------------
def rfm_segment(row):
    if row['RFM_Score'] == '555':
        return 'Champions'
    elif row['R_Score'] == '5':
        return 'Loyal Customers'
    elif row['R_Score'] == '1':
        return 'Lost Customers'
    elif row['F_Score'] == '5':
        return 'Frequent Buyers'
    elif row['M_Score'] == '5':
        return 'Big Spenders'
    else:
        return 'Others'

rfm['Segment'] = rfm.apply(rfm_segment, axis=1)

# --------------------------------
# 5️⃣ Visualize Segments
```

```
# ------------------------------

plt.figure(figsize=(10, 6))

sns.countplot(data=rfm, x='Segment', order=rfm['Segment'].value_counts().index, palette='Set2')

plt.title("🧠 RFM Customer Segments")

plt.ylabel("Number of Customers")

plt.xlabel("Segment")

plt.xticks(rotation=45)

plt.tight_layout()

plt.show()


# ------------------------------

# 6️⃣ Summary of Key Insights

# ------------------------------

print("📊 RFM Segment Counts:\n")

print(rfm['Segment'].value_counts())


print("\n✅ Recommendations:")

print("• 🎯 Focus campaigns on 'Champions' and 'Big Spenders'.")

print("• 🚀 Reward 'Frequent Buyers' to build loyalty.")

print("• 🧲 Try to re-engage 'Lost Customers' with offers.")

print("• 💡 Use personalized marketing for each segment to improve retention.")
```



📊 RFM Segment Counts:

Segment

Others 19223

Champions 551

Name: count, dtype: int64

✅ Recommendations:

- 🎯 Focus campaigns on 'Champions' and 'Big Spenders'.
- 🚀 Reward 'Frequent Buyers' to build loyalty.
- 🧲 Try to re-engage 'Lost Customers' with offers.
- 💡 Use personalized marketing for each segment to improve retention.

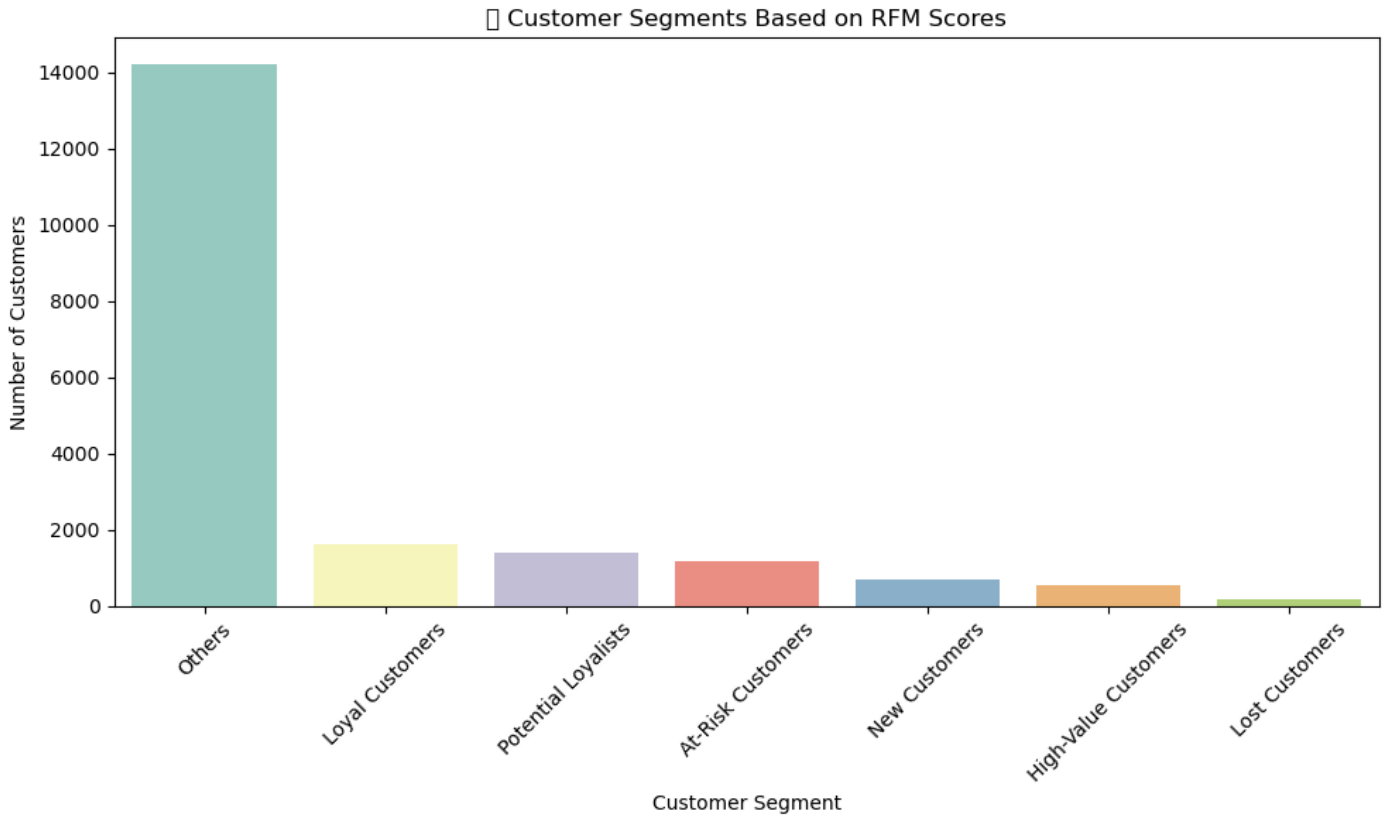
Subtask 2: Segment Customers Based on RFM Scores

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

# Load your dataset
df = pd.read_csv("ecommerce_data_final_cleaned.csv")

# ---------------------------------------------
# Step 1: Format Date Column
# ---------------------------------------------
# Replace with your actual date column
if 'order_date' in df.columns:
    df['order_date'] = pd.to_datetime(df['order_date'], errors='coerce')
elif 'date' in df.columns:
    df['order_date'] = pd.to_datetime(df['date'], errors='coerce')
else:
    raise Exception("❌ No valid date column found!")

df = df[df['order_date'].notnull()]

# ---------------------------------------------
# Step 2: Prepare RFM Input
# ---------------------------------------------
# Rename columns for consistency
df.rename(columns={
    'customer_id': 'CustomerID',
    'value [USD]': 'TotalRevenue'
}, inplace=True)

# Remove rows with missing values
df = df.dropna(subset=['CustomerID', 'order_date', 'TotalRevenue'])

# Reference date (latest date in dataset)
```

```
ref_date = df['order_date'].max()

# Group data by CustomerID
rfm = df.groupby('CustomerID').agg({
    'order_date': lambda x: (ref_date - x.max()).days,  # Recency
    'CustomerID': 'count',                          # Frequency
    'TotalRevenue': 'sum'                           # Monetary
}).rename(columns={
    'order_date': 'Recency',
    'CustomerID': 'Frequency',
    'TotalRevenue': 'Monetary'
}).reset_index()

# ---------------------------------------------
# Step 3: Create RFM Score (1–5)
# ---------------------------------------------
rfm['R_Score'] = pd.qcut(rfm['Recency'], 5, labels=[5, 4, 3, 2, 1])
rfm['F_Score'] = pd.qcut(rfm['Frequency'].rank(method='first'), 5, labels=[1, 2, 3, 4, 5])
rfm['M_Score'] = pd.qcut(rfm['Monetary'], 5, labels=[1, 2, 3, 4, 5])

# ---------------------------------------------
# Step 4: Segment Customers Based on RFM Score
# ---------------------------------------------
def assign_segment(row):
    r = int(row['R_Score'])
    f = int(row['F_Score'])
    m = int(row['M_Score'])

    if r == 5 and f == 5 and m == 5:
        return 'High-Value Customers'
    elif r >= 4 and f >= 4 and m >= 4:
        return 'Loyal Customers'
    elif r <= 2 and f >= 3 and m >= 3:
        return 'At-Risk Customers'
```

```
    elif r == 5 and f <= 2 and m <= 2:
        return 'New Customers'
    elif r == 1 and f == 1 and m == 1:
        return 'Lost Customers'
    elif r >= 4 and f >= 3 and m >= 3:
        return 'Potential Loyalists'
    else:
        return 'Others'

rfm['Customer_Segment'] = rfm.apply(assign_segment, axis=1)

# ---------------------------------------------
# Step 5: Visualize Customer Segments
# ---------------------------------------------
plt.figure(figsize=(10, 6))
sns.countplot(data=rfm, x='Customer_Segment', order=rfm['Customer_Segment'].value_counts().index,
palette="Set3")
plt.title("📊 Customer Segments Based on RFM Scores")
plt.ylabel("Number of Customers")
plt.xlabel("Customer Segment")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# ---------------------------------------------
# Step 6: Insights & Marketing Strategy
# ---------------------------------------------
print("📋 Customer Segment Counts:\n")
print(rfm['Customer_Segment'].value_counts())

print("\n🎯 Marketing Recommendations:")
print("• High-Value Customers → Reward with loyalty perks.")
print("• Loyal Customers → Send exclusive product previews.")
print("• At-Risk Customers → Use reminder and discount emails.")
```

```
print("• New Customers → Nurture with onboarding offers.")
print("• Lost Customers → Win-back with aggressive offers.")
print("• Potential Loyalists → Encourage with reward points.")
```



📋 Customer Segment Counts:

Customer_Segment

Others 14193

Loyal Customers 1625

Potential Loyalists 1385

At-Risk Customers 1165

New Customers 696

High-Value Customers 551

Lost Customers 159

Name: count, dtype: int64

🎯 Marketing Recommendations:

• High-Value Customers → Reward with loyalty perks.

• Loyal Customers → Send exclusive product previews.

• At-Risk Customers → Use reminder and discount emails.

• New Customers → Nurture with onboarding offers.

• Lost Customers → Win-back with aggressive offers.

• Potential Loyalists → Encourage with reward points.

Subtask 3: Visualize Customer Segments

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

# Step 1: Load and clean data
df = pd.read_csv("ecommerce_data_final_cleaned.csv")

# Convert date column to datetime
df['date'] = pd.to_datetime(df['date'], format='%d/%m/%Y')

# Drop rows with missing customer IDs or values
df.dropna(subset=['customer_id', 'value [USD]'], inplace=True)

# Step 2: Calculate RFM Metrics
snapshot_date = df['date'].max() + pd.Timedelta(days=1)

rfm = df.groupby('customer_id').agg({
    'date': lambda x: (snapshot_date - x.max()).days,
    'customer_id': 'count',
    'value [USD]': 'sum'
})

rfm.columns = ['Recency', 'Frequency', 'Monetary']

# Step 3: Create RFM scores (1–5 scale)
rfm['Recency_Score'] = pd.qcut(rfm['Recency'], 5, labels=[5,4,3,2,1]).astype(int)
rfm['Frequency_Score'] = pd.qcut(rfm['Frequency'].rank(method='first'), 5, labels=[1,2,3,4,5]).astype(int)
rfm['Monetary_Score'] = pd.qcut(rfm['Monetary'], 5, labels=[1,2,3,4,5]).astype(int)

rfm['RFM_Score'] = rfm['Recency_Score'].astype(str) + rfm['Frequency_Score'].astype(str) +
rfm['Monetary_Score'].astype(str)
```

```
# Step 4: Segment customers based on RFM score
def assign_segment(row):
    r, f, m = row['Recency_Score'], row['Frequency_Score'], row['Monetary_Score']
    if r == 5 and f == 5 and m == 5:
        return 'High-Value Customers'
    elif r >= 4 and f >= 4:
        return 'Loyal Customers'
    elif r <= 2 and f >= 3 and m >= 3:
        return 'At-Risk Customers'
    elif r == 5 and f <= 2:
        return 'New Customers'
    elif r <= 2 and f <= 2 and m <= 2:
        return 'Lost Customers'
    elif r >= 4 and f >= 3:
        return 'Potential Loyalists'
    else:
        return 'Others'

rfm['Customer_Segment'] = rfm.apply(assign_segment, axis=1)

# 🎯 Summary Counts
print("\n📋 Customer Segment Counts:\n")
print(rfm['Customer_Segment'].value_counts())

# ----------------------------------------
# 📊 Visualizations
# ----------------------------------------

# 1️⃣ Bar Chart - Segment Distribution
plt.figure(figsize=(10, 6))
segment_counts = rfm['Customer_Segment'].value_counts()
sns.barplot(x=segment_counts.index, y=segment_counts.values, palette='viridis')
plt.title("Customer Segment Distribution")
plt.ylabel("Number of Customers")
```

```
plt.xlabel("Segment")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# 2️⃣ Scatter Plot - Recency vs Frequency
plt.figure(figsize=(8, 6))
sns.scatterplot(data=rfm, x='Recency', y='Frequency', hue='Customer_Segment', palette='tab10')
plt.title("Recency vs Frequency by Segment")
plt.tight_layout()
plt.show()

# 3️⃣ Heatmap - RFM Score Density
heatmap_data = rfm.groupby(['Recency_Score', 'Frequency_Score']).size().unstack()
plt.figure(figsize=(8, 6))
sns.heatmap(heatmap_data, cmap="YlGnBu", annot=True, fmt="g")
plt.title("Heatmap of Recency vs Frequency Scores")
plt.tight_layout()
plt.show()

# 4️⃣ Pie Chart - Segment Proportions
plt.figure(figsize=(8, 8))
rfm['Customer_Segment'].value_counts().plot.pie(autopct='%1.1f%%', startangle=140,
colors=sns.color_palette("Set2"))
plt.title("Customer Segment Proportions")
plt.ylabel("")
plt.tight_layout()
plt.show()

# ✅ Optional: Save the final RFM DataFrame
rfm.to_csv("rfm_segmented_customers.csv")

# 🧠 Marketing Suggestions
print("\n🎯 Marketing Recommendations:")
```

```
print("• High-Value Customers → Reward with loyalty perks.")
print("• Loyal Customers → Send exclusive product previews.")
print("• At-Risk Customers → Use reminder and discount emails.")
print("• New Customers → Nurture with onboarding offers.")
print("• Lost Customers → Win-back with aggressive offers.")
print("• Potential Loyalists → Encourage with reward points.")
```

📋 Customer Segment Counts:

Customer_Segment

Others 10975

Loyal Customers 3252

Potential Loyalists 1406

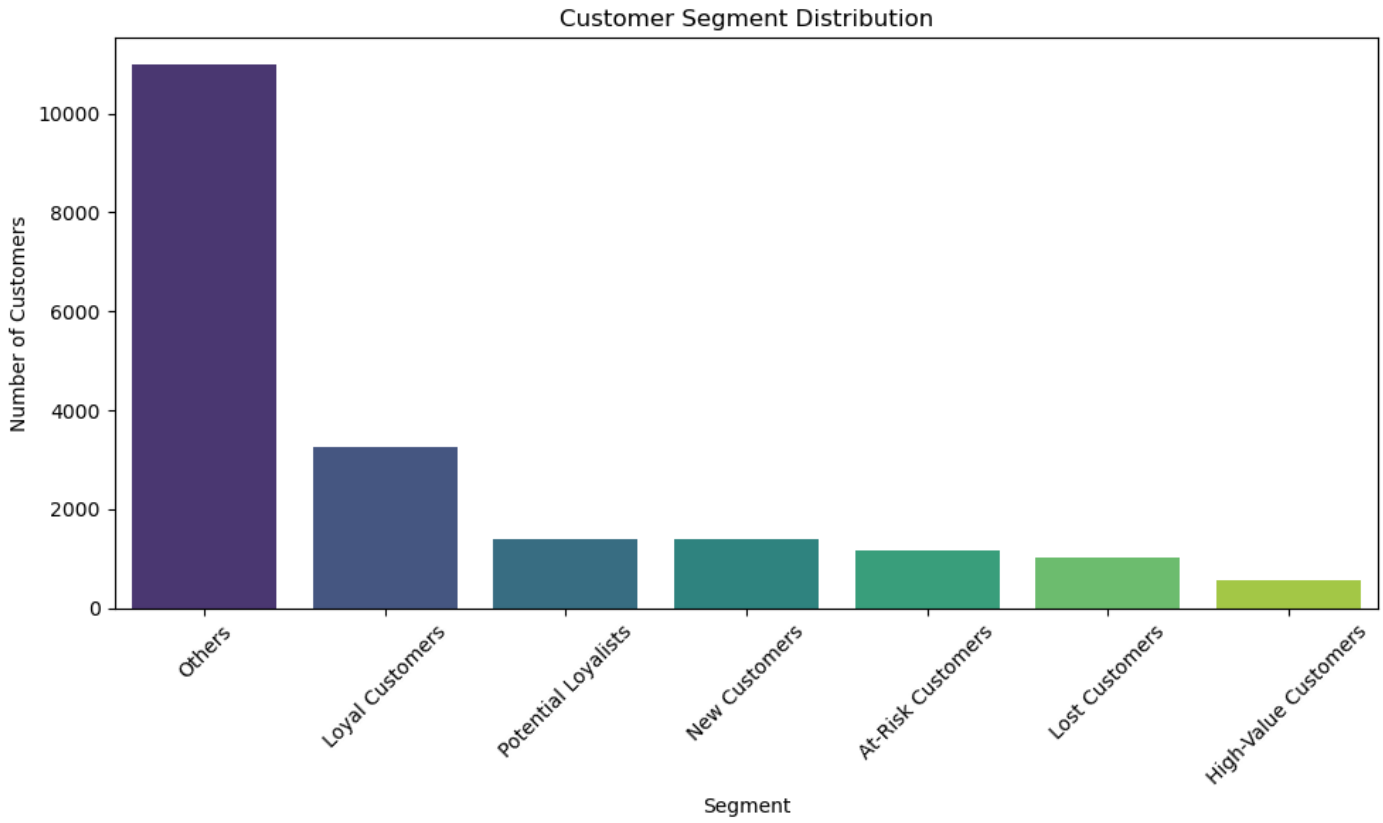
New Customers 1401

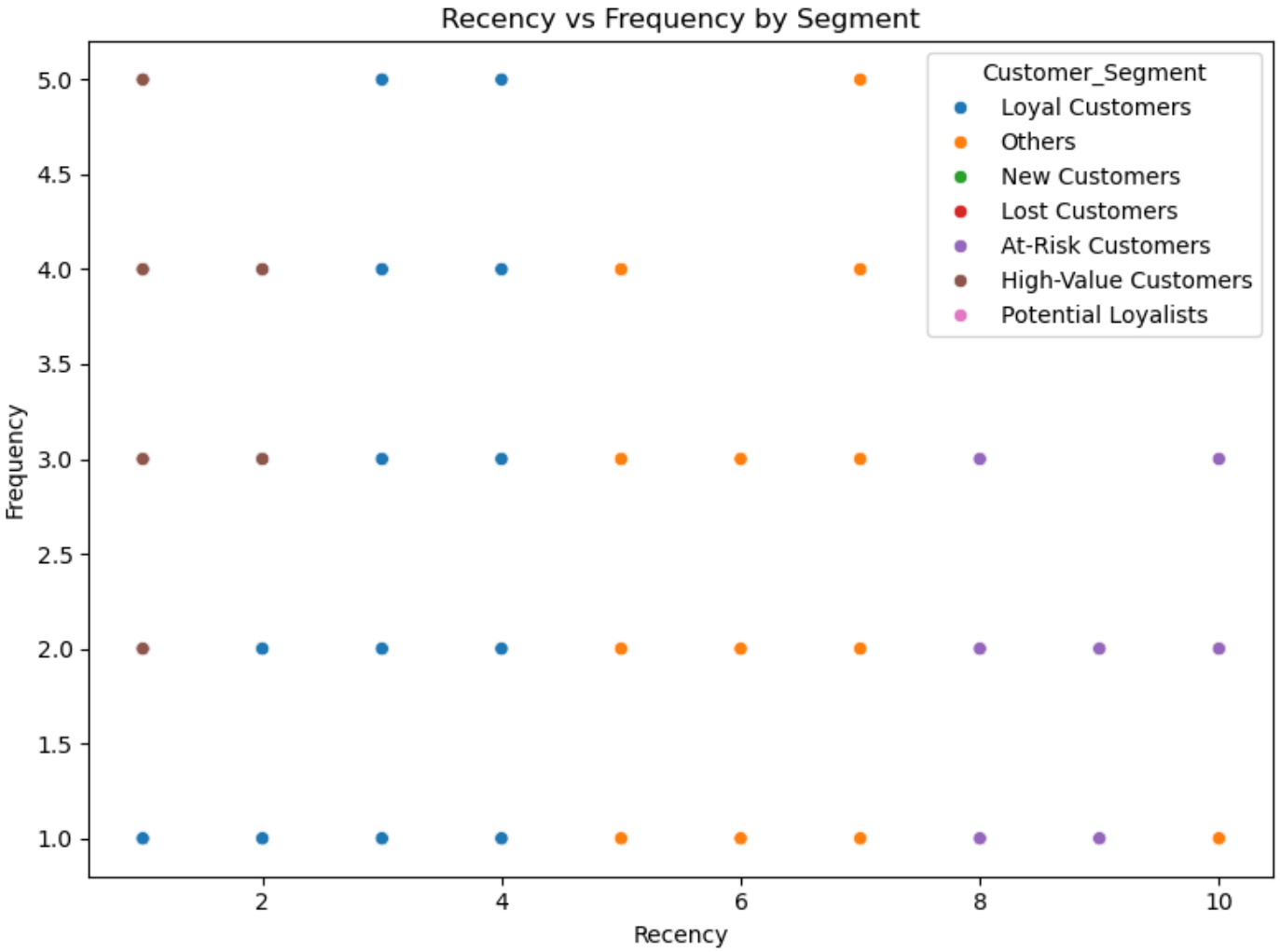
At-Risk Customers 1165

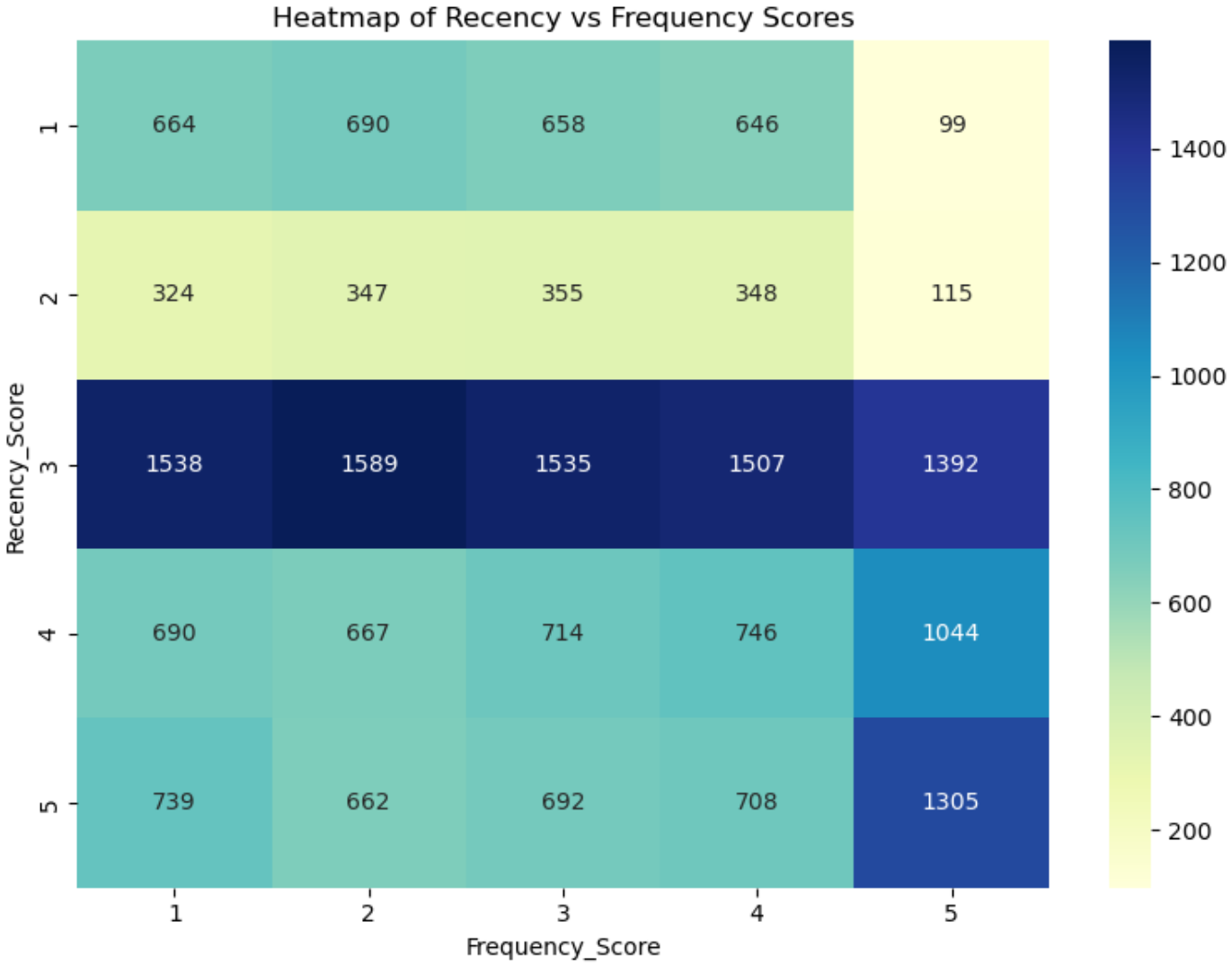
Lost Customers 1024

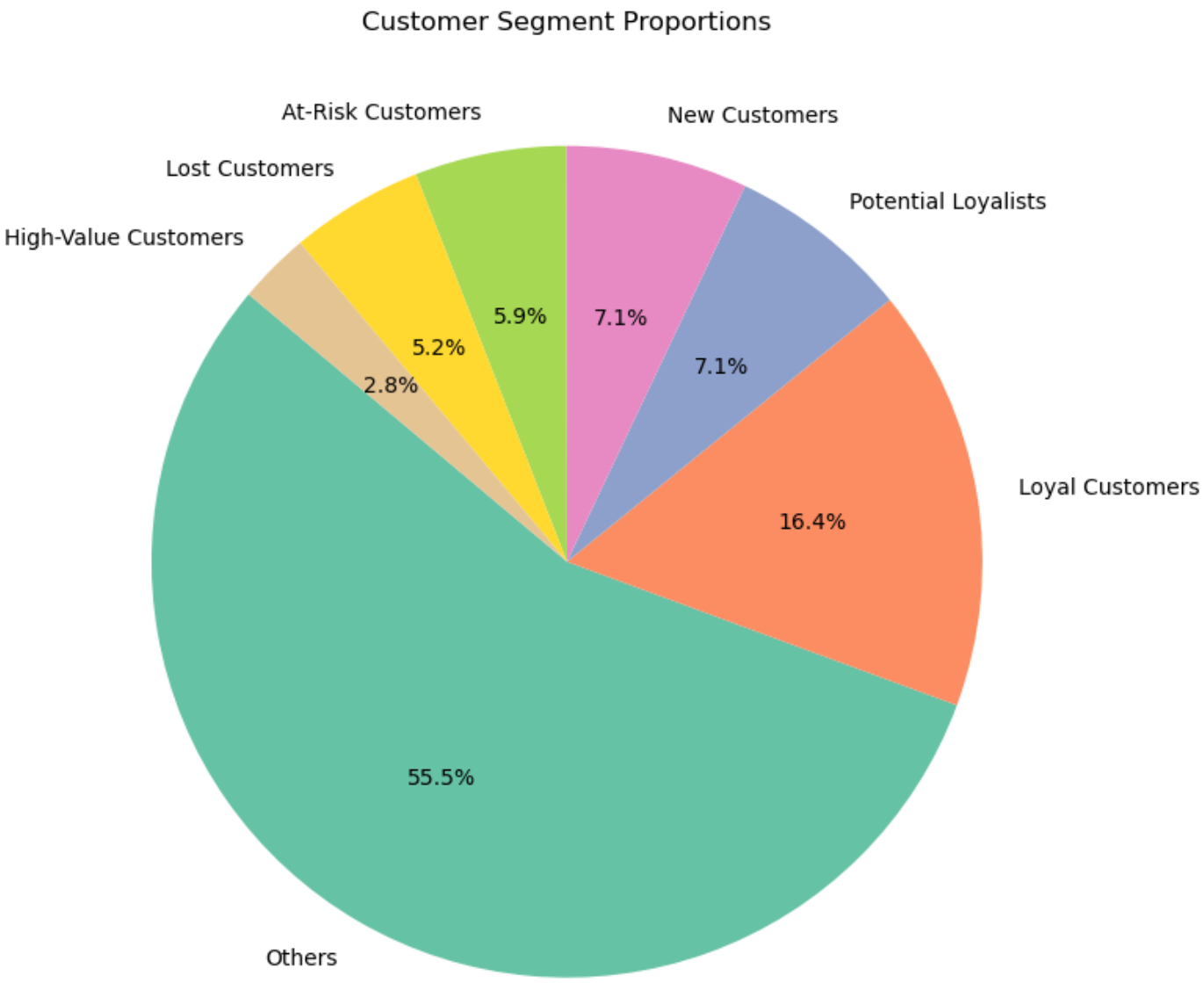
High-Value Customers 551

Name: count, dtype: int64




Recency vs Frequency by Segment
Customer_Segment
Loyal Customers
Others
New Customers
Lost Customers
At-Risk Customers
High-Value Customers
Potential Loyalists
Frequency
Recency
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
2
4
6
8
10


Heatmap of Recency vs Frequency Scores
Recency_Score
Frequency_Score
1
2
3
4
5
664
690
658
646
99
324
347
355
348
115
1538
1589
1535
1507
1392
690
667
714
746
1044
739
662
692
708
1305
1400
1200
1000
800
600
400
200



🎯 Marketing Recommendations:

- High-Value Customers → Reward with loyalty perks.
- Loyal Customers → Send exclusive product previews.
- At-Risk Customers → Use reminder and discount emails.
- New Customers → Nurture with onboarding offers.
- Lost Customers → Win-back with aggressive offers.
- Potential Loyalists → Encourage with reward points.