RFM Customer Segmentation and Monte-Carlo Simulation

by Louis Bailey

Summary

This project uses transaction data to understand customer behavior and classify customers into three segments: High-Value Customers, Growth Opportunities, and At-Risk or Churned Customers. The aim is to provide actionable insights for retention, re-engagement, and optimizing marketing efforts.

In [1]:

```
%matplotlib inline
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
plt.rcParams['figure.figsize'] = [12, 6]
```

Data

In [2]:

```
df = pd.read_excel(r'C:\Users\baile\Downloads\Online Retail.xlsx')
```

In [3]:

```
df.head(3)
```

Out[3]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom

Clean & Prepare

In [4]:

```
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
```

In [5]:

```
df = df[df.Quantity>0]
df = df[df.UnitPrice>0]
```

In [6]:

```
df['Total'] = df.Quantity*df.UnitPrice
df['Date'] = df.InvoiceDate.dt.date

latest_purchase = df.groupby('CustomerID')['Date'].max()
df['LatestPurchaseDate'] = df['CustomerID'].map(latest_purchase)
```

Basic Info on Customers

In [7]:

```
print(f'Number of Unique Customers: {len(df.CustomerID.unique())}')
```

```
Number of Unique Customers: 4338
```

In [8]:

```
#
customers_by_country = df.groupby("Country")["CustomerID"].nunique()
customers_by_country = customers_by_country.reset_index()
customers_by_country.columns = ["Country", "UniqueCustomers"]
```

```
customers_by_country = customers_by_country[customers_by_country.UniqueCustomers>10].sort_values(by='UniqueCustomers
                                                                                                  ascending=False)
customers_by_country       # Number of Unique Customers by Country (>10)
```

Out[8]:

	Country	UniqueCustomers
35	United Kingdom	3920
14	Germany	94
13	France	87
30	Spain	30
3	Belgium	25
32	Switzerland	21
26	Portugal	19
18	Italy	14
12	Finland	12
1	Austria	11

- The vast majority of customers are in the UK

EDA

product sales over time

In [9]:

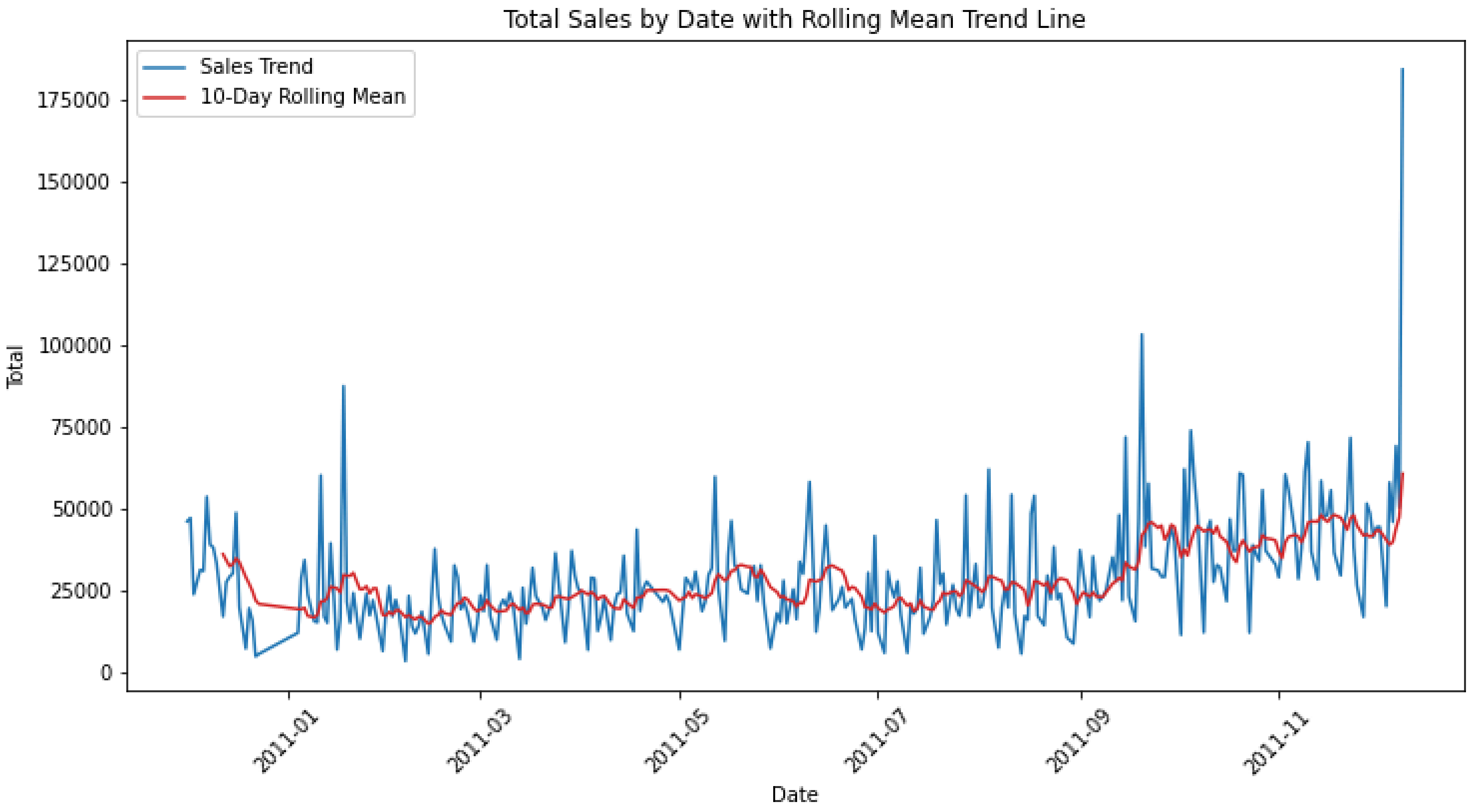
```
grouped = df.groupby('Date')['Total'].sum().reset_index()
total_by_date = grouped.sort_values(by='Date', ascending=True)
```

In [10]:

```
#
grouped['RollingMean'] = grouped['Total'].rolling(window=10, center=False).mean()

sns.lineplot(data=grouped, x='Date', y='Total', label="Sales Trend")
sns.lineplot(data=grouped, x='Date', y='RollingMean', color='tab:red',label="10-Day Rolling Mean")
plt.title('Total Sales by Date with Rolling Mean Trend Line')
plt.legend()
plt.xticks(rotation=45)


plt.show()
```



- There are three significant spikes in the sales trend, with a particularly prominent one occurring at the end.

---

best selling products by total revenue

In [11]:

```
product_stats = df.groupby('Description').agg(
    TotalQuantity=('Quantity', 'sum'),
    UniquePurchasers=('CustomerID', 'nunique'),
    TotalRevenue=('Total', 'sum')
).reset_index()

top_revenue = product_stats.sort_values(by='TotalRevenue', ascending=False).head(5)
```

In [12]:

```
top_revenue
```

Out[12]:

	Description	TotalQuantity	UniquePurchasers	TotalRevenue
2319	PAPER CRAFT , LITTLE BIRDIE	80995	1	168469.60
2767	REGENCY CAKESTAND 3 TIER	12374	881	142264.75
3698	WHITE HANGING HEART T-LIGHT HOLDER	36706	856	100392.10
1762	JUMBO BAG RED RETROSPOT	46078	635	85040.54
1992	MEDIUM CERAMIC TOP STORAGE JAR	77916	138	81416.73

best selling products by total quantity

In [13]:

```
product_stats = df.groupby('Description').agg(
    TotalQuantity=('Quantity', 'sum'),
    UniquePurchasers=('CustomerID', 'nunique'),
    TotalRevenue=('Total', 'sum')
).reset_index()

top_quantity = product_stats.sort_values(by='TotalQuantity', ascending=False).head(5)
```

In [14]:

```
top_quantity
```

Out[14]:

	Description	TotalQuantity	UniquePurchasers	TotalRevenue
2319	PAPER CRAFT , LITTLE BIRDIE	80995	1	168469.60
1992	MEDIUM CERAMIC TOP STORAGE JAR	77916	138	81416.73
3786	WORLD WAR 2 GLIDERS ASSTD DESIGNS	54319	307	13558.41
1762	JUMBO BAG RED RETROSPOT	46078	635	85040.54
3698	WHITE HANGING HEART T-LIGHT HOLDER	36706	856	100392.10

- The huge spike in the sales trend plot was from a single customer buying lots of 'PAPER CRAFT , LITTLE BIRDIE'.

In [15]:
```
df[df.Quantity==80995]
```

Out[15]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Total	Date	LatestPu
540421	581483	23843	PAPER CRAFT , LITTLE BIRDIE	80995	2011-12-09 09:15:00	2.08	16446.0	United Kingdom	168469.6	2011-12-09	

- And it was a single purchase

notable customer

In [16]:
```
df[df.CustomerID==16446.0]
```

Out[16]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Total	Date	LatestPu
194354	553573	22980	PANTRY SCRUBBING BRUSH	1	2011-05-18 09:52:00	1.65	16446.0	United Kingdom	1.65	2011-05-18	
194355	553573	22982	PANTRY PASTRY BRUSH	1	2011-05-18 09:52:00	1.25	16446.0	United Kingdom	1.25	2011-05-18	
540421	581483	23843	PAPER CRAFT , LITTLE BIRDIE	80995	2011-12-09 09:15:00	2.08	16446.0	United Kingdom	168469.60	2011-12-09	

- The customer who made the large purchase has only made three purchases this year. The other two were in May and were much smaller.

- Despite making only three purchases, the customer has spent a lot and made their purchases recently. They currently seem to be a high-value customer.

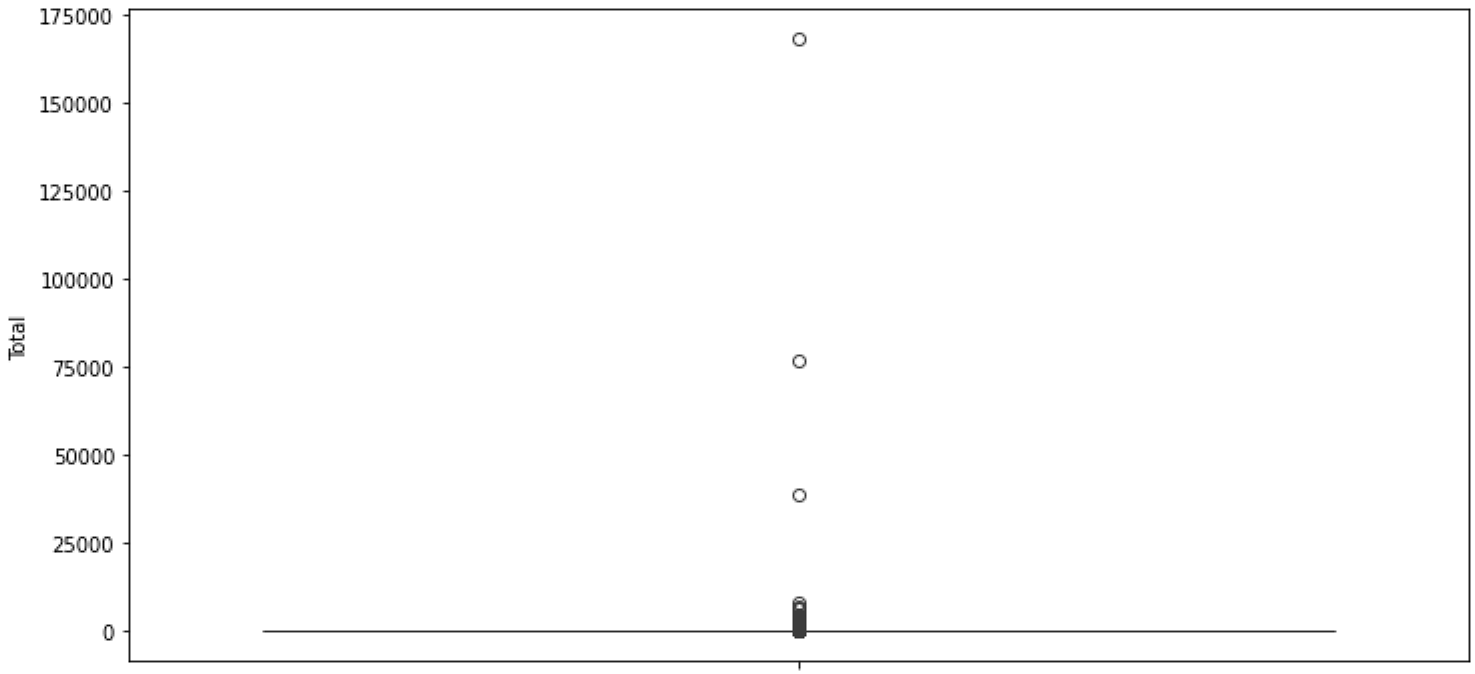
- Grouping customers by value could provide useful insights. I will use RFM Analysis to achieve this.

RFM Segmentation

capping outliers for single purchases

In [17]:

```
sns.boxplot(df.Total);
```



In [18]:
```
threshold = df.Total.quantile(0.999)
df = df[df.Total<=threshold]
```

- By capping single purchases at the 99.9th percentile the largest single purchases (more than about 600 dollars) are being removed. This is being done to prevent outliers from skewing everything from here on. I'm asuming that customers who make big purchases will be treated on a case-by-case basis rather than algorithmically.

creating RFM scores

- RFM analysis is used to divide customers into different segments like high-value customer, medium-value customer and so on. The RFM score is calculated based on recency, frequency, and monetary value.

- The idea of RFM is to segment customers into actionable insights:

1. Customer Retention: Focus on customers likely to churn.
2. Loyalty Programs: Reward frequent and high-spending customers.
3. Win-Back Campaigns: Re-engage customers with low recency.
4. Resource Allocation: Invest more in high-value segments and less in one-time buyers.

In [19]:

```
#calculating recency
df_recency = df.groupby(by='CustomerID', as_index=False)['Date'].max()
df_recency.columns = ['CustomerID', 'LastPurchaseDate']
recent_date = df_recency['LastPurchaseDate'].max()
df_recency['Recency'] = df_recency['LastPurchaseDate'].apply(lambda x: (recent_date - x).days)

#calculating frequency
frequency_df = df.drop_duplicates().groupby(
    by=['CustomerID'], as_index=False)['Date'].count()
frequency_df.columns = ['CustomerID', 'Frequency']

#calculating monetary value
monetary_df = df.groupby(by='CustomerID', as_index=False)['Total'].sum()
monetary_df.columns = ['CustomerID', 'Monetary']

#merging all 3 columns into one dataframe
rf_df = df_recency.merge(frequency_df, on='CustomerID')
rfm_df = rf_df.merge(monetary_df, on='CustomerID').drop(columns='LastPurchaseDate')

#ranking based on r,f and m
rfm_df['R_rank'] = rfm_df['Recency'].rank(ascending=False)
rfm_df['F_rank'] = rfm_df['Frequency'].rank(ascending=True)
rfm_df['M_rank'] = rfm_df['Monetary'].rank(ascending=True)

# normalizing the rank of the customers
rfm_df['R_rank_norm'] = (rfm_df['R_rank']/rfm_df['R_rank'].max())*100
rfm_df['F_rank_norm'] = (rfm_df['F_rank']/rfm_df['F_rank'].max())*100
rfm_df['M_rank_norm'] = (rfm_df['F_rank']/rfm_df['M_rank'].max())*100

#calculating rfm score
#recency is given the least weight(0.15), frequency the second highest(0.28), and monetary
```

```python
#is given the highest weight(0.57)
rfm_df['RFM_Score'] = 0.15*rfm_df['R_rank_norm']+0.28*rfm_df['F_rank_norm']+0.57*rfm_df['M_rank_norm']
rfm_df['RFM_Score'] *= 0.05
rfm_df = rfm_df.round(2)

# categorizing customers based on RFM_Score
rfm_df["Customer_segment"] = np.where(
rfm_df['RFM_Score'] > 4.0, "High-Value Customers",
np.where(
    rfm_df['RFM_Score'] > 2.5, "Growth Opportunities",
    "At-Risk or Churned Customers"
))

rfm_df.drop(columns=['Recency','Frequency','Monetary','R_rank','F_rank','M_rank'], inplace=True)
```

In [20]:

```python
rfm_df.head()
```

Out[20]:

	CustomerID	R_rank_norm	F_rank_norm	M_rank_norm	RFM_Score	Customer_segment
0	12347.0	96.20	88.22	88.22	4.47	High-Value Customers
1	12348.0	38.12	42.29	42.29	2.08	At-Risk or Churned Customers
2	12349.0	74.35	67.05	67.05	3.41	Growth Opportunities
3	12350.0	5.25	24.86	24.86	1.10	At-Risk or Churned Customers
4	12352.0	57.97	71.03	71.03	3.45	Growth Opportunities

- The normalized rank scores are from 0-100, and the RFM scores are from 0-5.

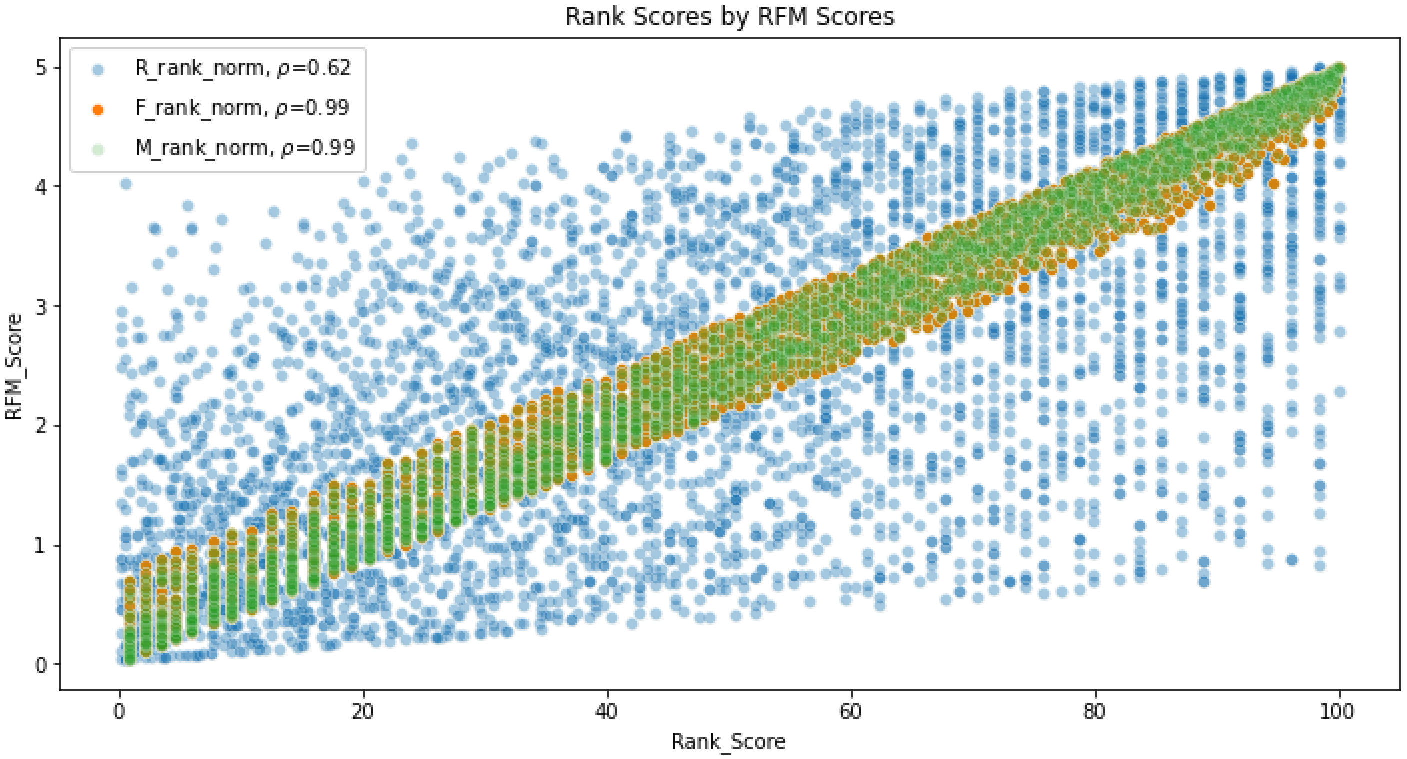
- The 3 groups are High-Value Customers, Growth Opportunities, and At-Risk or Churned Customers. Only three groups were made because the company has around 4,300 customers and the data only goes back about a year. I think having a separate group for At-Risk and Churned for example, would spread the customers too thin and be too strong of an assumption given the short history of purchases.

Making Sense of the RFM Scores

In [21]:

```
#
r_coeff = round(np.corrcoef(rfm_df.R_rank_norm,rfm_df.RFM_Score)[0,1],2)
f_coeff = round(np.corrcoef(rfm_df.F_rank_norm,rfm_df.RFM_Score)[0,1],2)
m_coeff = round(np.corrcoef(rfm_df.M_rank_norm,rfm_df.RFM_Score)[0,1],2)

sns.scatterplot(data=rfm_df,x='R_rank_norm',y='RFM_Score', label=fr'R_rank_norm, $\rho$={r_coeff}', alpha=0.4)
sns.scatterplot(data=rfm_df,x='F_rank_norm',y='RFM_Score', label=fr'F_rank_norm, $\rho$={f_coeff}')
sns.scatterplot(data=rfm_df,x='M_rank_norm',y='RFM_Score', label=fr'M_rank_norm, $\rho$={m_coeff}', alpha=0.2)
plt.title('Rank Scores by RFM Scores')
plt.xlabel('Rank_Score')
plt.legend();
```



- Frequency and monetary values are highly correlated with each other. This makes sense because the total amount spent by most customers tends to increase proportionally with the number of purchases they make. A exception would be our notable customer mentioned above.

- Recency is less correlated because the proportional relationship doesn't apply and because it was given less weight in the RFM calculation.

looking at a few examples

In [22]:

```
def customer_summary(n):
    customer_data = df[df['CustomerID'] == n]

    return customer_data.groupby('CustomerID').agg({'Total': 'sum',
                                                     'LatestPurchaseDate': 'max',
                                                     'CustomerID': 'size'
                                                   }).rename(columns={'CustomerID': 'UniquePurchases'})
```

In [23]:

```
#high value customer (RFM_Score>4)
customer_summary(12347.0)
```

Out[23]:

	Total	LatestPurchaseDate	UniquePurchases
CustomerID			
12347.0	4310.0	2011-12-07	182

- This high value customer has made a total of 182 puchases totalling to 4,310 dollars, and their last purchase was very recent.

In [24]:

```
#growth opportunity customer (2.5<RFM_Score<=4)
customer_summary(12356.0)
```

Out[24]:

```
               Total   LatestPurchaseDate   UniquePurchases
CustomerID
   12356.0   2811.43           2011-11-17                59
```

- This growth opportunity customer has made a total of 59 purchases totalling to 2,811 dollars, and their last purchase was almost a month ago.

In [25]:

```
#at-risk or churned customer (RFM_Score<=2.5)
customer_summary(12350.0)
```

Out[25]:

```
               Total   LatestPurchaseDate   UniquePurchases
CustomerID
   12350.0     334.4           2011-02-02                17
```

- This at-risk or churned customer has made a total of 17 purchases totalling to 334 dollars, and their last purchase was 10 months ago.

Actionable Insights

High-Value Customers: These are loyal and profitable customers. The focus here is on retention and rewarding loyalty.

1. Offer exclusive perks such as discounts, early access to sales, or loyalty programs to maintain engagement.
2. Personalize communication and marketing based on their preferences (e.g., recommend products related to past purchases).
3. Ask for feedback to understand their needs better and make them feel valued.

Growth Opportunity Customers: These customers have potential but are not yet consistent or high-value buyers. The goal is to increase their frequency and spending.

1. Provide incentives like discounts, free shipping, or bundle offers to encourage repeat purchases.
2. Send reminders or recommendations for products they've shown interest in or bought in the past.
3. Use email or SMS campaigns to re-engage them with personalized offers and promotions.

At-Risk or Churned Customers: These are customers who haven't engaged recently or show signs of disengagement. The focus is on reactivation or understanding their reasons for churn.

1. Reach out with win-back campaigns, such as limited-time discounts or special offers to re-engage them.
2. Conduct surveys to understand why they stopped engaging and address any issues identified.
3. Segment and target them based on the severity of disengagement (e.g., recent inactivity vs. long-term churn) and tailor your approach accordingly

Estimating Spending of each Group - Monte Carlo Simulation

In [26]:

```
#total spending for each customer in each category
total_spending_per_customer = df.groupby(["CustomerID"])["Total"].sum().reset_index()
total_spending_per_customer.columns = ["CustomerID", "Total_Spending"]

#merging total spending data with the RFM dataset
rfm_with_spending = pd.merge(rfm_df, total_spending_per_customer, on="CustomerID", how="inner")
```

In [27]:

```
rfm_with_spending.head(3)
```

Out[27]:

	CustomerID	R_rank_norm	F_rank_norm	M_rank_norm	RFM_Score	Customer_segment	Total_Spending
0	12347.0	96.20	88.22	88.22	4.47	High-Value Customers	4310.00
1	12348.0	38.12	42.29	42.29	2.08	At-Risk or Churned Customers	1797.24
2	12349.0	74.35	67.05	67.05	3.41	Growth Opportunities	1757.55

In [28]:

```
#running monte carlo simulation for each customer segment
num_simulations = 10000

segments = rfm_with_spending["Customer_segment"].unique()
simulation_results = {}

for segment in segments:
    segment_data = rfm_with_spending[rfm_with_spending["Customer_segment"] == segment]
    spending_data = segment_data["Total_Spending"]

    if spending_data.empty:
        continue

    #simulating spending
    simulated_spending = np.random.choice(spending_data, size=num_simulations, replace=True)
    simulation_results[segment] = {
        "mean": np.mean(simulated_spending),
        "std_dev": np.std(simulated_spending),
        "min": np.min(simulated_spending),
        "max": np.max(simulated_spending),
    }

#data frame to summarize simulation results
simulation_summary = pd.DataFrame.from_dict(simulation_results, orient="index")
simulation_summary.reset_index(inplace=True)
simulation_summary.columns = ["Customer Segment", "Mean Spending", "Std Dev", "Min Spending", "Max Spending"]
```

In [31]:

```
#reordering indexes
order = ["High-Value Customers", "Growth Opportunities", "At-Risk or Churned Customers"]
simulation_summary = simulation_summary.set_index("Customer Segment").reindex(order).reset_index()
```

In [32]:

```
simulation_summary
```

Out[32]:

	Customer Segment	Mean Spending	Std Dev	Min Spending	Max Spending
0	High-Value Customers	6370.083734	15595.268505	271.19	265989.86
1	Growth Opportunities	1552.467389	1839.516004	120.03	27329.72
2	At-Risk or Churned Customers	469.356734	586.053908	2.90	7829.89

- The Monte Carlo simulation takes the spending data for each customer segment, makes 10,000 random draws (with replacement) from that segment's spending data, and then calculates the summary statistics for the simulated distribution of 10,000 draws.

- The purpose of this simulation is to inform the budget for the 'Actionable Insights' above.

Conclusion

This project segmented customers using RFM analysis and estimated spending patterns through a Monte Carlo simulation. These insights support targeted strategies: retaining high-value customers, re-engaging at-risk customers, and nurturing growth opportunities. The goal of these insights is to help maximize customer value and improve performance.

```
rfm_df.to_csv('rfm_df.csv')
```

```
simulation_summary.to_csv('simulation_summary.csv')
```