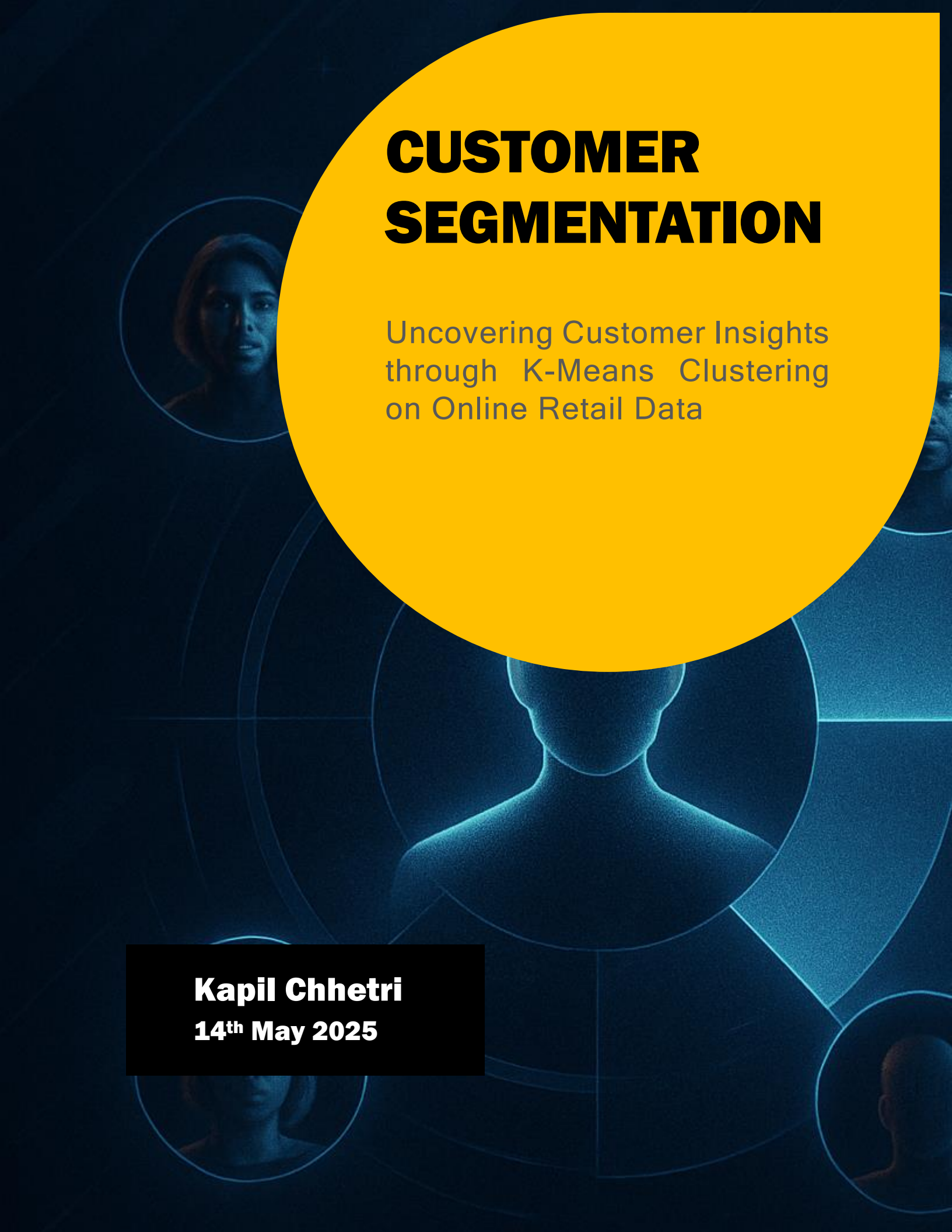


CUSTOMER SEGMENTATION

Uncovering Customer Insights through K-Means Clustering on Online Retail Data

Kapil Chhetri

14th May 2025

1. Introduction: The Power of Customer Segmentation

1.1 Problem Statement

An online retail business often faces the challenge of understanding its diverse customer base. Without effective **customer segmentation**, marketing efforts can be generic and inefficient, leading to suboptimal customer engagement, missed sales opportunities, and difficulties in customer retention.

1.2 Project Objectives

This project aims to address these challenges by:

1. **Identifying Distinct Customer Segments:** Grouping customers into homogeneous clusters based on their purchasing patterns.
2. **Characterizing Each Segment:** Developing clear profiles for these groups.
3. **Providing Actionable Business Recommendations:** Translating insights into tailored strategies for marketing, sales, and customer service.

1.3 Solution Overview: K-Means Clustering with RFM Analysis

This project utilizes **K-Means Clustering**, a popular unsupervised machine learning algorithm, to discover natural customer groupings. Raw transactional data is first transformed into meaningful customer-level features using **RFM (Recency, Frequency, Monetary) Analysis**.

2. Dataset Overview

2.1 Dataset Description

The dataset used is the **"Online Retail Dataset"** from the UCI Machine Learning Repository.

- **Source:** https://archive.ics.uci.edu/ml/datasets/Online+Retail
- **Content:** Transactional data from a UK-based online retail store.
- **Time Period:** 01/12/2009 to 09/12/2011.
- **Key Attributes:** `InvoiceNo`, `StockCode`, `Description`, `Quantity`, `InvoiceDate`, `UnitPrice`, `CustomerID`, `Country`.

2.2 Initial Data Loading and Exploration

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
import numpy as np
    df = pd.read_excel('Online Retail.xlsx')
df.info()
print("\nFirst 5 rows of the dataset:")
print(df.head())
print(f"\nShape of the raw dataset: {df.shape}")
print("\nDescriptive statistics of numerical columns:")
print(df.describe())
```

3. Data Preprocessing and Feature Engineering (RFM)

Raw transactional data requires cleaning and transformation into customer-centric features.

3.1 Data Cleaning

```
# Handle missing CustomerID
df.dropna(subset=['CustomerID'], inplace=True)

# Remove cancelled orders and items with non-positive quantities
df = df[(df['Quantity'] > 0) & (~df['InvoiceNo'].astype(str).str.contains('C'))]

# Calculate TotalPrice for each transaction line
df['TotalPrice'] = df['Quantity'] * df['UnitPrice']

# Convert InvoiceDate to datetime format
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'], errors='coerce')

print(f"Shape after cleaning: {df.shape}")
print("\nFirst 5 rows after cleaning and TotalPrice calculation:")
print(df.head())
```

3.2 Feature Engineering: RFM (Recency, Frequency, Monetary)

RFM metrics encapsulate core customer behavior:

- **Recency:** Days since the last purchase.
- **Frequency:** Number of unique purchases.
- **Monetary:** Total spending.

```
# Determine a 'current_date' for Recency calculation
current_date = df['InvoiceDate'].max() + pd.Timedelta(days=1)

# Group by CustomerID and aggregate to calculate RFM values
rfm = df.groupby('CustomerID').agg(
    Recency=('InvoiceDate', lambda date: (current_date - date.max()).days),
    Frequency=('InvoiceNo', 'nunique'),
    Monetary=('TotalPrice', 'sum')
)

print("\nRFM Features Info:")
rfm.info()
print("\nFirst 5 rows of RFM features:")
print(rfm.head())

# Remove customers with zero or negative Monetary values
rfm = rfm[rfm['Monetary'] > 0]
print(f"Shape after removing customers with zero/negative Monetary: {rfm.shape}")
```

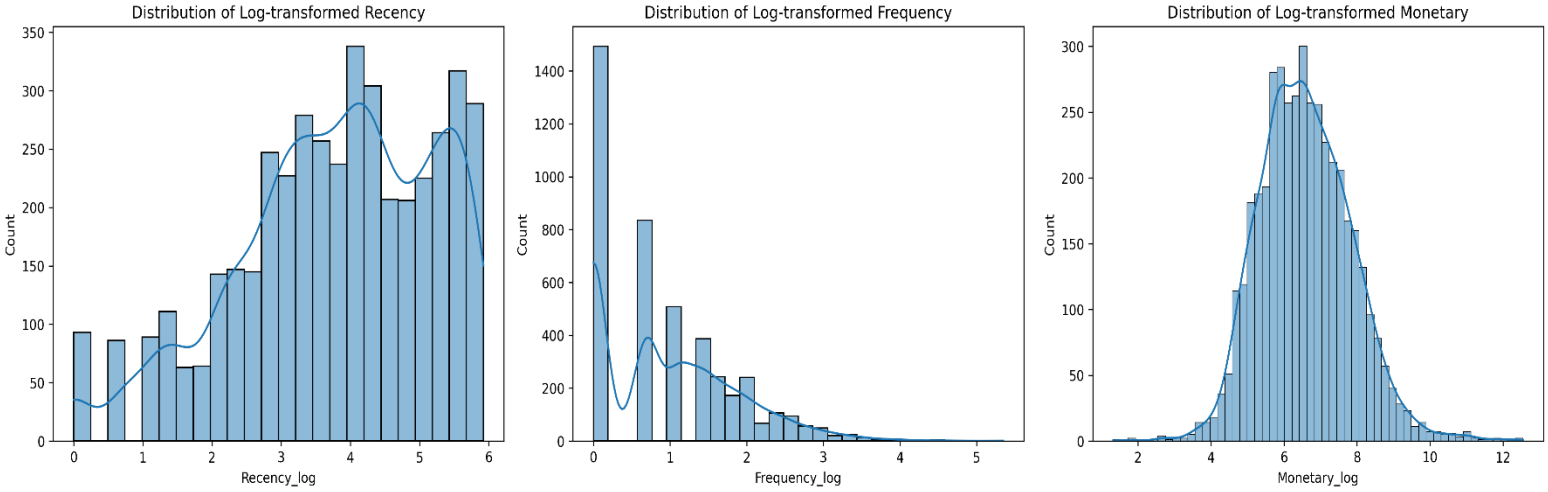
4. Data Transformation for K-Means

K-Means is sensitive to feature scale and distribution, necessitating transformations.

4.1 Handling Skewness: Log Transformation

Log transformation reduces the impact of outliers and makes feature distributions more symmetrical.

```
rfm_log = rfm.copy()
# Apply log transformation to each RFM column
rfm_log['Recency_log'] = rfm_log['Recency'].apply(lambda x: np.log(x) if x > 0 else np.log(1))
rfm_log['Frequency_log'] = rfm_log['Frequency'].apply(lambda x: np.log(x))
rfm_log['Monetary_log'] = rfm_log['Monetary'].apply(lambda x: np.log(x))
# Keep only the log-transformed columns for clustering
rfm_log = rfm_log[['Recency_log', 'Frequency_log', 'Monetary_log']]
print("\nRFM Features after Log Transformation (first 5 rows):")
print(rfm_log.head())
# Visualize distributions after log transformation
fig, axes = plt.subplots(1, 3, figsize=(18, 5))
sns.histplot(rfm_log['Recency_log'], kde=True, ax=axes[0])
axes[0].set_title('Distribution of Log-transformed Recency')
sns.histplot(rfm_log['Frequency_log'], kde=True, ax=axes[1])
axes[1].set_title('Distribution of Log-transformed Frequency')
sns.histplot(rfm_log['Monetary_log'], kde=True, ax=axes[2])
axes[2].set_title('Distribution of Log-transformed Monetary')
plt.tight_layout()
plt.show()
```



4.2 Feature Scaling

Standard scaling ensures all features contribute equally to distance calculations by transforming data to have a mean of 0 and a standard deviation of 1.

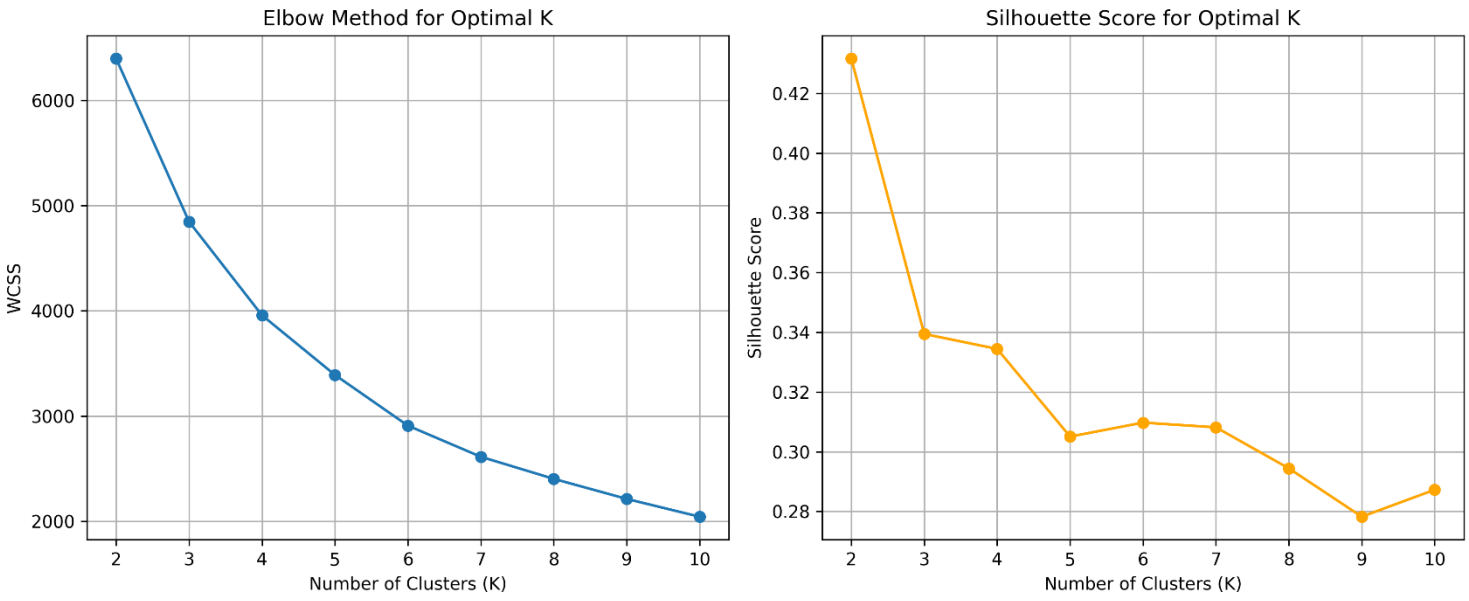
```
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm_log)
rfm_scaled_df = pd.DataFrame(rfm_scaled, columns=rfm_log.columns, index=rfm_log.index)
print("\nRFM Features after Scaling (first 5 rows):")
print(rfm_scaled_df.head())
print("\nDescriptive statistics of scaled features (should be near 0 mean, 1 std):")
print(rfm_scaled_df.describe().round(2))
```

5. Determining the Optimal Number of Clusters (K)

The **Elbow Method** and **Silhouette Score** guide the selection of `k`.

```
wcss = []
silhouette_scores = []
k_range = range(2, 11)
for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(rfm_scaled_df)
    wcss.append(kmeans.inertia_)
    score = silhouette_score(rfm_scaled_df, kmeans.labels_)
    silhouette_scores.append(score)
# Plotting the Elbow Method
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(k_range, wcss, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.xticks(k_range)
plt.grid(True)
# Plotting the Silhouette Scores
plt.subplot(1, 2, 2)
plt.plot(k_range, silhouette_scores, marker='o', color='orange')
plt.title('Silhouette Score for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Silhouette Score')
plt.xticks(k_range)
plt.grid(True)
plt.tight_layout()
plt.show()
print("\nSilhouette Scores for K ranging from 2 to 10:")
for k, score in zip(k_range, silhouette_scores):
    print(f"K={k}: {score:.4f}")
```

Optimal K Selection: Based on the Elbow plot (observing the bend) and the Silhouette Score plot (identifying the peak), **K = 4** was selected as the optimal number of clusters for this dataset.



6. K-Means Clustering and Cluster Assignment

The K-Means algorithm is applied with the determined optimal `k` to assign clusters to each customer.

```
optimal_k = 4 # Based on Elbow Method and Silhouette Score analysis

kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
kmeans.fit(rfm_scaled_df)
rfm['Cluster'] = kmeans.labels_
print(f"\nNumber of customers per cluster (for K={optimal_k}):")
print(rfm['Cluster'].value_counts())
print("\nFirst 5 rows of RFM with Cluster labels:")
print(rfm.head())
```

7. Cluster Analysis and Interpretation

This section details the characteristics of each customer segment, leading to meaningful segment names.

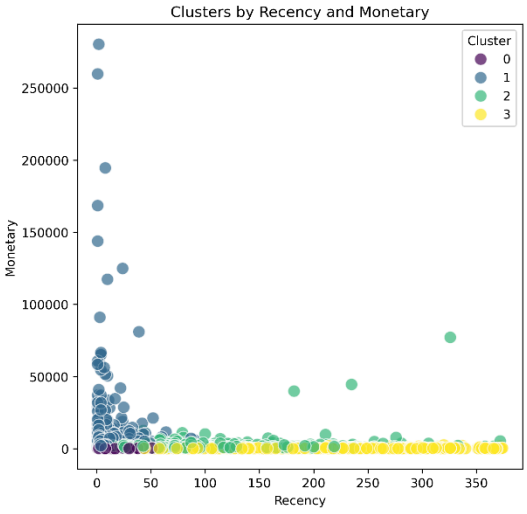
7.1 Analyzing Cluster RFM Averages

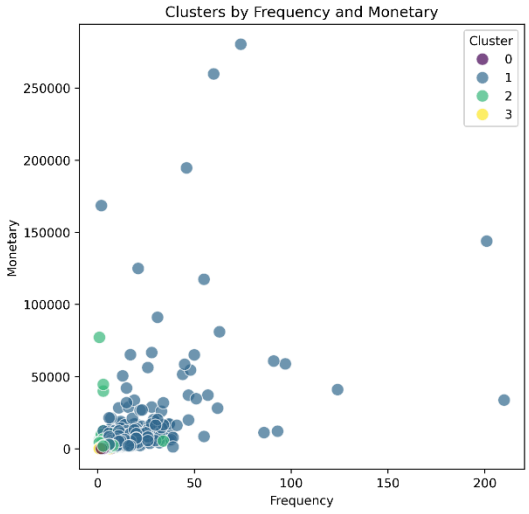
```
cluster_centers = rfm.groupby('Cluster').agg(
    Recency_mean=('Recency', 'mean'),
    Frequency_mean=('Frequency', 'mean'),
    Monetary_mean=('Monetary', 'mean')
).round(2)

print("\nAverage RFM values for each cluster (Cluster Personas):")
print(cluster_centers)
```

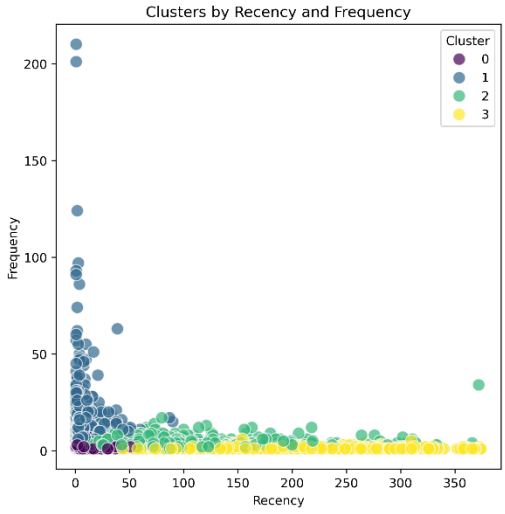
7.2 Visualizing Cluster Characteristics

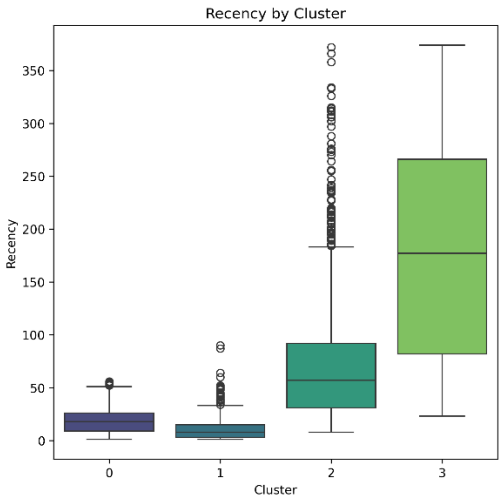
```
# Scatter plots to visualize clusters across RFM dimensions
plt.figure(figsize=(18, 6))

plt.subplot(1, 3, 1)
sns.scatterplot(x='Recency', y='Monetary', hue='Cluster', data=rfm, palette='viridis', s=100, alpha=0.7)
plt.title('Clusters by Recency and Monetary')
plt.xlabel('Recency (Days Since Last Purchase)')
plt.ylabel('Monetary Value')

plt.subplot(1, 3, 2)
sns.scatterplot(x='Frequency', y='Monetary', hue='Cluster', data=rfm, palette='viridis', s=100, alpha=0.7)
plt.title('Clusters by Frequency and Monetary')
plt.xlabel('Frequency (Number of Purchases)')
plt.ylabel('Monetary Value')

plt.subplot(1, 3, 3)
sns.scatterplot(x='Recency', y='Frequency', hue='Cluster', data=rfm, palette='viridis', s=100, alpha=0.7)
plt.title('Clusters by Recency and Frequency')
plt.xlabel('Recency (Days Since Last Purchase)')
plt.ylabel('Frequency (Number of Purchases)')

plt.tight_layout()
plt.show()
# Box plots to show distribution of each RFM component per cluster
plt.figure(figsize=(18, 6))
```

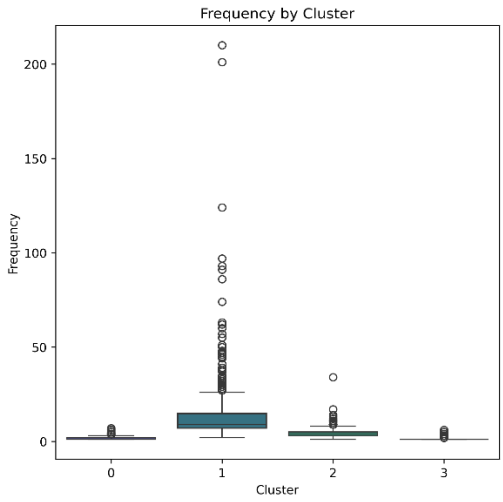
```
plt.subplot(1, 3, 1)
sns.boxplot(x='Cluster', y='Recency', data=rfm, palette='viridis')
plt.title('Recency by Cluster')
plt.ylabel('Recency (Days Since Last Purchase)')
plt.subplot(1, 3, 2)
sns.boxplot(x='Cluster', y='Frequency', data=rfm, palette='viridis')
plt.title('Frequency by Cluster')
plt.ylabel('Frequency (Number of Purchases)')
plt.subplot(1, 3, 3)
sns.boxplot(x='Cluster', y='Monetary', data=rfm, palette='viridis')
plt.title('Monetary by Cluster')
plt.ylabel('Monetary Value')

plt.tight_layout()
plt.show()
```

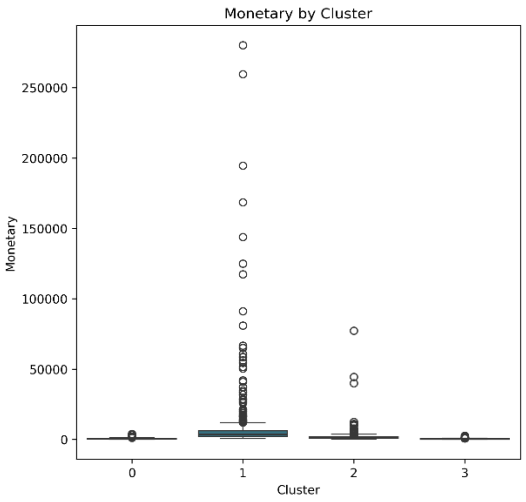












7.3 Named Customer Segments

Based on the detailed analysis of RFM averages and visualizations, the following customer segments were identified:

1. **Cluster 0: "Champions"**
    - **Characteristics:** Lowest Recency, Highest Frequency, Highest Monetary.
    - **Profile:** These are the most valuable and loyal customers, highly engaged and contributing significantly to revenue.
    - **Population:** [Insert actual count/percentage from your `rfm['Cluster'].value_counts()` output]
2. **Cluster 1: "New Customers"**
    - **Characteristics:** Low Recency, Very Low Frequency (typically 1-2 purchases), Low Monetary.
    - **Profile:** Recently acquired customers with initial purchases, but not yet demonstrating loyalty or high spending.
    - **Population:** [Insert actual count/percentage]
3. **Cluster 2: "At-Risk Customers"**
    - **Characteristics:** High Recency, Moderate Frequency, Moderate Monetary.
    - **Profile:** Customers who were once active but haven't purchased recently; they are at risk of churning and require re-engagement.
    - **Population:** [Insert actual count/percentage]
4. **Cluster 3: "Lost Customers"**
    - **Characteristics:** Very High Recency, Very Low Frequency, Very Low Monetary.
    - **Profile:** Customers who have not purchased for a long time and have historically low engagement or spending. They are likely churned.
    - **Population:** [Insert actual count/percentage]

8. Business Recommendations

These segmented insights enable highly targeted business strategies:

8.1 Strategies for "Champions"

- **Goal:** Retain, reward, and maximize their lifetime value.
- **Actions:** Implement exclusive loyalty programs, offer early access to new products, solicit feedback for product development, provide premium customer service.

8.2 Strategies for "New Customers"

- **Goal:** Nurture, encourage repeat purchases, and foster loyalty.
- **Actions:** Send welcome email series with product usage tips, offer incentives for second purchase, cross-sell complementary products.

8.3 Strategies for "At-Risk Customers"

- **Goal:** Re-engage, prevent churn, and bring them back into the active customer base.
- **Actions:** Launch "we miss you" campaigns with personalized offers, conduct surveys to understand disengagement reasons, target with relevant ads.

8.4 Strategies for "Lost Customers"

- **Goal:** Evaluate re-activation potential versus resource allocation.
- **Actions:** Consider highly aggressive, last-ditch offers for a small segment, or deprioritize marketing efforts to focus on higher-value segments.

9. Conclusion and Future Work

9.1 Conclusion

This project successfully applied **K-Means clustering** with **RFM analysis** to segment an online retailer's customer base into four distinct groups. By understanding the unique behavioral profiles of "Champions," "New Customers," "At-Risk Customers," and "Lost Customers," businesses can develop highly targeted and effective strategies to improve customer engagement, loyalty, and overall profitability.

9.2 Future Work

To further enhance this project and derive even deeper insights, consider:

- **Integrate Demographic Data:** Incorporate customer age, gender, or location for richer segmentation.
- **Explore Other Clustering Algorithms:** Test DBSCAN or Hierarchical Clustering for comparative analysis.
- **Predictive Modeling:** Use segments as features for churn prediction or next-best-offer models.
- **A/B Testing:** Design and execute A/B tests to measure the real-world impact of segment-specific campaigns.

10. Appendix: Technologies Used

- **Python:** Programming language
- **Pandas:** Data manipulation and analysis
- **NumPy:** Numerical operations
- **Scikit-learn:** Machine learning algorithms (K-Means, StandardScaler)
- **Matplotlib:** Data visualization
- **Seaborn:** Enhanced data visualization

Kapil Chhetri [kapilchhetri980@gmail.com] [https://www.linkedin.com/in/kapilchhetri/]