RFM Analysis

Md Shamsul Hoque Khan

2023-08-23

Introduction to RFM Analysis

RFM (Recency, Frequency, Monetary) analysis is a customer segmentation technique that helps marketers understand the behavior of their customers. Each customer is scored on three dimensions:

Recency: How recently the customer made a purchase Frequency: How often the customer makes a purchase Monetary: How much the customer spends Through clustering these metrics, businesses can identify customer groups and design targeted marketing strategies.

Preprocessing

Load Required Libraries

```r
# Load tidyverse for data manipulation and ggplot2 for plotting
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
# Load readxl for reading Excel files
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.3
```

Load the Data

```r
# Load the sample_superstore data set downloaded from Tableau Public
suppressWarnings({
  df <- read_excel("sample_superstore.xls")
})
```

Data Exploration

Check for Missing Values

```
# Check for any missing values in the data
sum(is.na(df))
```

```
## [1] 0
```

No missing values are found in the dataset.

Check Data Structure

```
# Examine the structure of the dataset
str(df)
```

```
## tibble [9,994 x 21] (S3: tbl_df/tbl/data.frame)
##  $ Row ID       : num [1:9994] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Order ID     : chr [1:9994] "CA-2016-152156" "CA-2016-152156" "CA-2016-138688" "US-2015-108966" .
##  $ Order Date   : POSIXct[1:9994], format: "2016-11-08" "2016-11-08" ...
##  $ Ship Date    : POSIXct[1:9994], format: "2016-11-11" "2016-11-11" ...
##  $ Ship Mode    : chr [1:9994] "Second Class" "Second Class" "Second Class" "Standard Class" ...
##  $ Customer ID  : chr [1:9994] "CG-12520" "CG-12520" "DV-13045" "SO-20335" ...
##  $ Customer Name: chr [1:9994] "Claire Gute" "Claire Gute" "Darrin Van Huff" "Sean O'Donnell" ...
##  $ Segment      : chr [1:9994] "Consumer" "Consumer" "Corporate" "Consumer" ...
##  $ Country      : chr [1:9994] "United States" "United States" "United States" "United States" ...
##  $ City         : chr [1:9994] "Henderson" "Henderson" "Los Angeles" "Fort Lauderdale" ...
##  $ State        : chr [1:9994] "Kentucky" "Kentucky" "California" "Florida" ...
##  $ Postal Code  : num [1:9994] 42420 42420 90036 33311 33311 ...
##  $ Region       : chr [1:9994] "South" "South" "West" "South" ...
##  $ Product ID   : chr [1:9994] "FUR-BO-10001798" "FUR-CH-10000454" "OFF-LA-10000240" "FUR-TA-1000057
##  $ Category     : chr [1:9994] "Furniture" "Furniture" "Office Supplies" "Furniture" ...
##  $ Sub-Category : chr [1:9994] "Bookcases" "Chairs" "Labels" "Tables" ...
##  $ Product Name : chr [1:9994] "Bush Somerset Collection Bookcase" "Hon Deluxe Fabric Upholstered St
##  $ Sales        : num [1:9994] 262 731.9 14.6 957.6 22.4 ...
##  $ Quantity     : num [1:9994] 2 3 2 5 2 7 4 6 3 5 ...
##  $ Discount     : num [1:9994] 0 0 0 0.45 0.2 0 0 0.2 0.2 0 ...
##  $ Profit       : num [1:9994] 41.91 219.58 6.87 -383.03 2.52 ...
```

The data columns are in the correct format.

RFM Calculation

Recency Calculation

```
# Identify the latest date in the dataset
max(df$`Order Date`)
```

```
## [1] "2017-12-30 UTC"
```

The dataset contains data up to December 2017.

```
# Calculate recency for each customer
reference_date <- as.Date(max(df$"Order Date")) + 1

recency_df <- df %>%
  group_by(`Customer ID`) %>%
  summarise(Recency = as.numeric(difftime(reference_date, max(`Order Date`), units="days"))) %>%
  arrange(Recency)

tail(recency_df)
```

```
## # A tibble: 6 x 2
##   `Customer ID` Recency
##   <chr>           <dbl>
## 1 PC-19000          883
## 2 VT-21700         1001
## 3 CM-12715         1036
## 4 RE-19405         1098
## 5 GR-14560         1136
## 6 NB-18580         1166
```

Frequency Calculation

```
# Calculate the frequency of purchases for each customer
frequency_df <- df %>%
  group_by(`Customer ID`) %>%
  summarise(Frequency = n()) %>%
  arrange(desc(Frequency))
```

Monetary Value Calculation

```
# Calculate the monetary value (based on Sales) for each customer
monetary_df <- df %>%
  group_by(`Customer ID`) %>%
  summarise(Monetary = sum(Sales))
```

Combine RFM Metrics

```r
# Combine Recency, Frequency, and Monetary metrics into one RFM DataFrame
rfm_df <- recency_df %>%
  inner_join(frequency_df, by = "Customer ID") %>%
  inner_join(monetary_df, by = "Customer ID")

# Standardize the metrics for further clustering
scaled_rfm <- as.data.frame(scale(rfm_df[, c('Recency', 'Frequency', 'Monetary')]))
scaled_rfm_with_ID <- data.frame("Customer ID" = rfm_df$`Customer ID`, scaled_rfm)

head(scaled_rfm_with_ID)
```

```
##   Customer.ID    Recency   Frequency   Monetary
## 1    CC-12430 -0.7883636  1.02477624 -0.0101947
## 2    EB-13975 -0.7883636 -1.05770321 -0.4768927
## 3    JM-15580 -0.7883636 -0.89751249 -0.9863925
## 4    PO-18865 -0.7883636  0.06363188 -0.1535508
## 5    BP-11185 -0.7829934  0.86458551  0.2963523
## 6    BS-11755 -0.7829934  0.22382260 -0.1272379
```

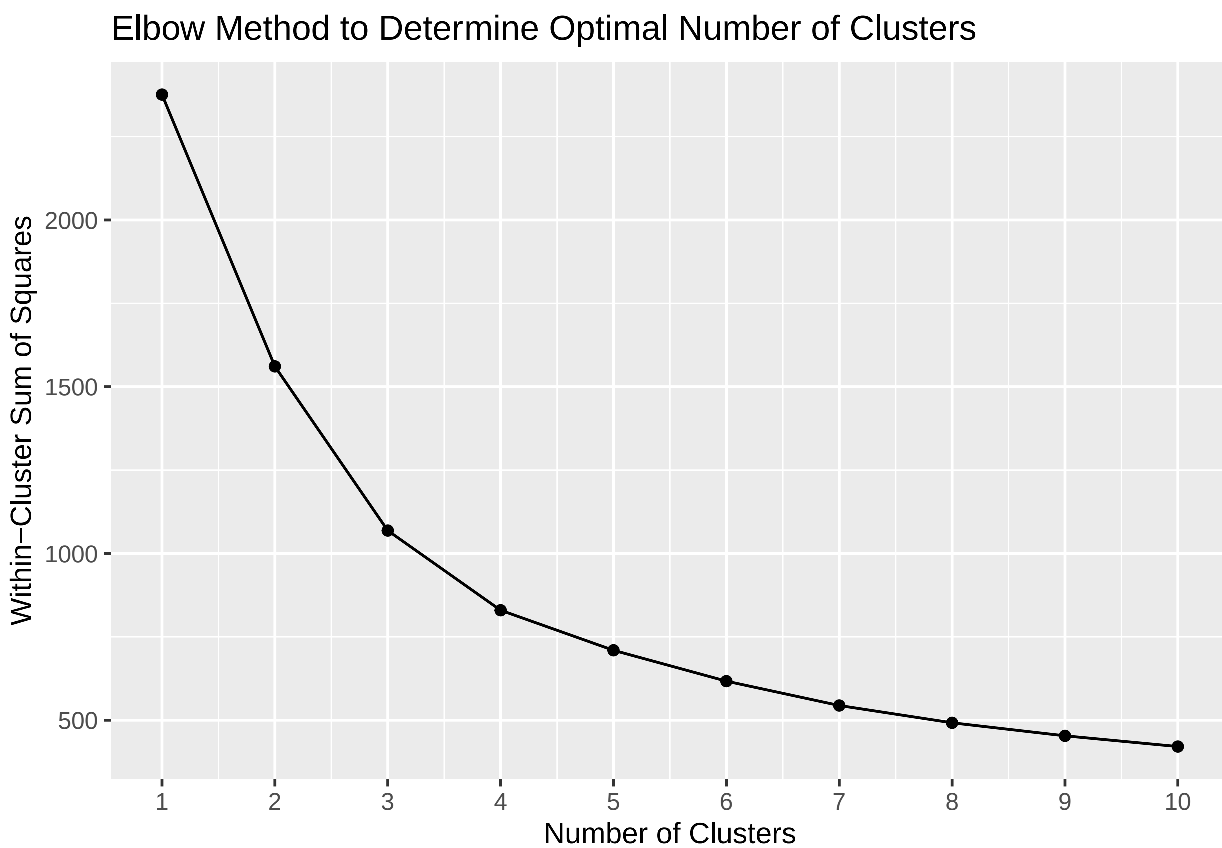
Clustering

Determine Number of Clusters

```r
# Use the elbow method to identify the optimal number of clusters
set.seed(12)
wss <- numeric()

for (i in 1:10){
  kmeans_model <- kmeans(scaled_rfm_with_ID[,-1], centers = i, nstart = 20)
  wss[i] <- kmeans_model$tot.withinss
}
```

Elbow Plot

```r
# Plot the elbow graph to determine the optimal number of clusters
ggplot(data.frame(Clusters = 1:10, WSS = wss), aes(x = Clusters, y = WSS)) +
  geom_point() +
  geom_line() +
  ggtitle("Elbow Method to Determine Optimal Number of Clusters") +
  xlab("Number of Clusters") +
  ylab("Within-Cluster Sum of Squares") +
  scale_x_continuous(breaks = seq(1, 10, by = 1))
```



Based on the elbow plot, 3 clusters appear to be optimal.

K-Means Clustering

```
# Run K-means clustering algorithm with the optimal number of clusters (3)
set.seed(12)
kmeans_model <- kmeans(scaled_rfm_with_ID[,-1], centers = 3, nstart = 20)

# Add the cluster labels to the original RFM DataFrame
rfm_df[, "Cluster"] <- kmeans_model$cluster
```

Cluster Analysis

Cluster Descriptions

```
# Calculate the average Recency, Frequency, and Monetary value for each cluster
group_rfm_df <- rfm_df %>%
  group_by(Cluster) %>%
  summarise(
    Recency = mean(Recency),
    Frequency = mean(Frequency),
```

```r
    Monetary = mean(Monetary)
  )

group_rfm_df
```

```
## # A tibble: 3 x 4
##   Cluster Recency Frequency Monetary
##     <int>   <dbl>     <dbl>    <dbl>
## 1       1    79.8     19.8     5614.
## 2       2   521.       7.68    1567.
## 3       3    84.9     10.4     1911.
```

Label Customer Types

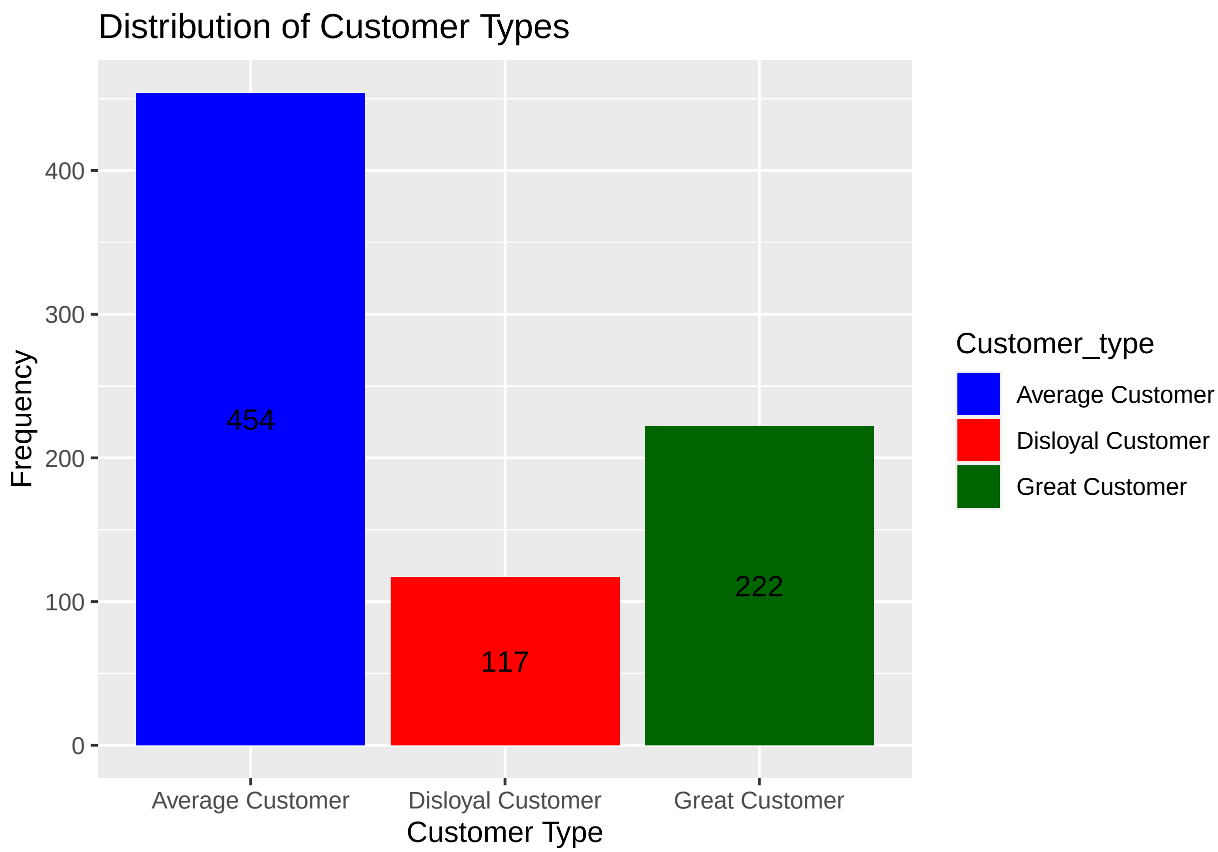
```r
# Label customer types based on their cluster
rfm_df_labeled <- rfm_df %>%
  mutate(
    Customer_type = case_when(Cluster == 1 ~ "Great Customer",
                              Cluster == 2 ~ "Disloyal Customer",
                              TRUE ~ "Average Customer")
  )

head(filter(rfm_df_labeled, Cluster == 2))
```

```
## # A tibble: 6 x 6
##   `Customer ID` Recency Frequency Monetary Cluster Customer_type
##   <chr>           <dbl>     <int>    <dbl>   <int> <chr>
## 1 MG-18205          265         2     16.7       2 Disloyal Customer
## 2 BD-11560          279         4    321.        2 Disloyal Customer
## 3 AC-10660          283         6    657.        2 Disloyal Customer
## 4 VG-21805          287         6    427.        2 Disloyal Customer
## 5 JK-15325          288         4    384.        2 Disloyal Customer
## 6 KS-16300          296         4     88.5       2 Disloyal Customer
```

Plotting Customer Types

```r
# Bar plot to show the distribution of customer types
ggplot(rfm_df_labeled, aes(x = Customer_type, fill = Customer_type)) +
  geom_bar() +
  geom_text(
    aes(label = ..count..),
    stat = 'count',
    position = position_stack(vjust = 0.5)
  ) +
  ggtitle("Distribution of Customer Types") +
  xlab("Customer Type") +
  ylab("Frequency") +
  scale_fill_manual(values = c("Great Customer" = "dark green", "Average Customer" = "blue", "Disloyal C
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
```



Conclusion

Based on the RFM analysis, we observe a large number of Average Customers and a reasonable number of Great Customers. Only about 14.75% of the customer base is classified as disloyal. The store should focus its marketing efforts on retaining Great Customers and converting Average Customers to Great Customers.