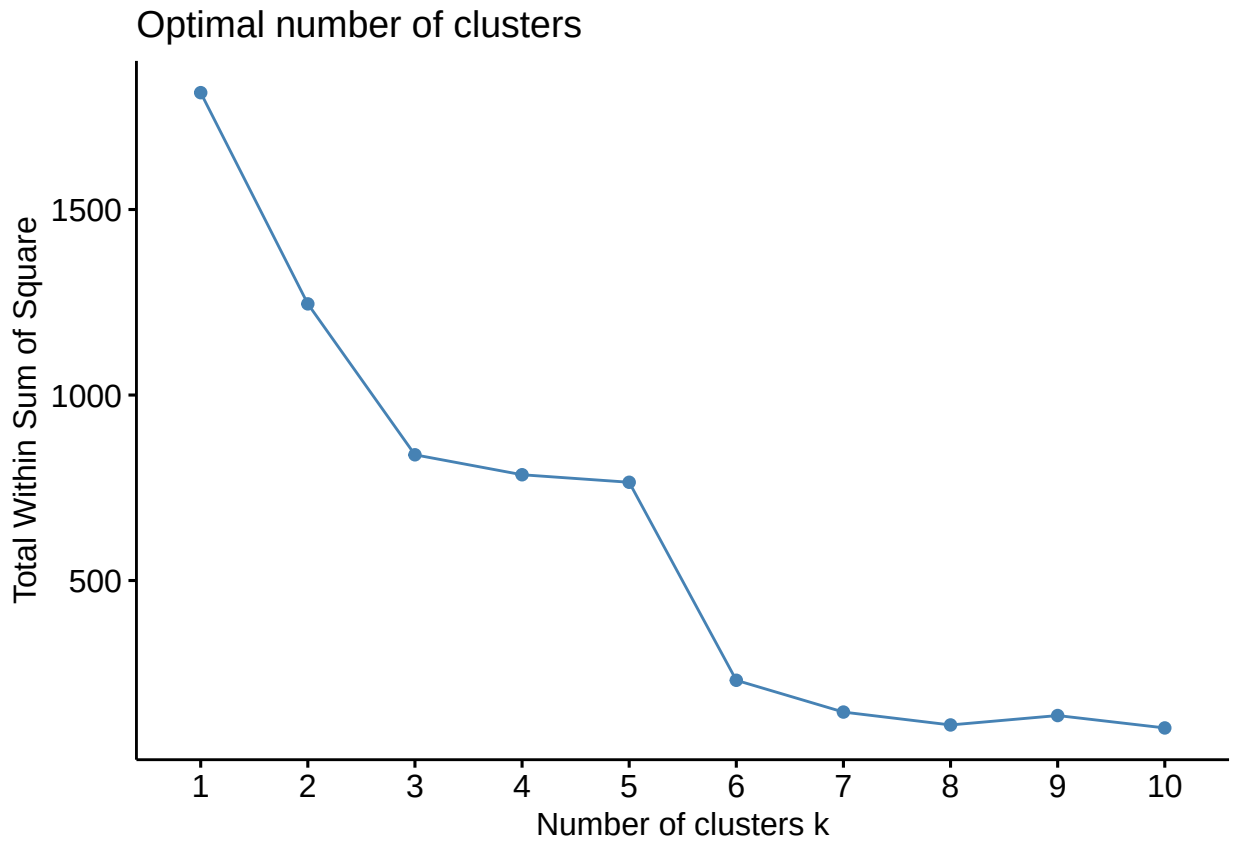
Retail Data Analysis Report

Anastasia Volokhova

2025-05-06

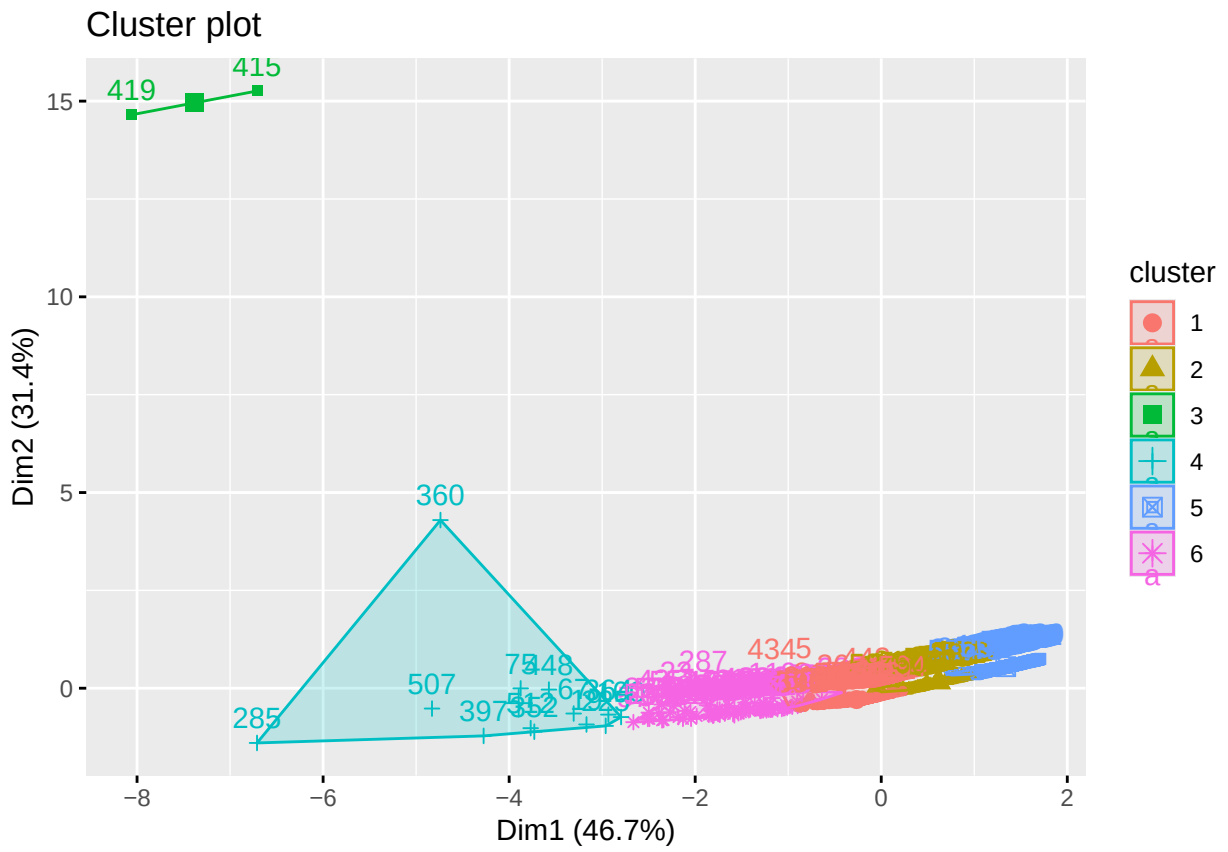
```r
library(factoextra)
library(lubridate)
library(dplyr)
library(ggplot2)
library(cluster)
library(forecast)
library(arulesViz)
library(arules)
data <- read.csv('retail data.csv')
data <- data %>% select(-1) %>% rename(Purchase_Date = Date)
data$Purchase_Date <- mdy(data$Purchase_Date)

# RFM Analysis and K-menas Clustering
ref_date <- max(data$Purchase_Date) + days(1)
rfm <- data %>%
  group_by(Customer) %>%
  summarise(
    Recency = as.numeric(ref_date - max(Purchase_Date)),
    Frequency = n_distinct(DocumentID),
    Monetary = sum(Price * Quantity)
  )
rfm.sc <- scale(rfm[,-1])
# Found optimal number of clusters
fviz_nbclust(rfm.sc, kmeans, method = "wss")
```



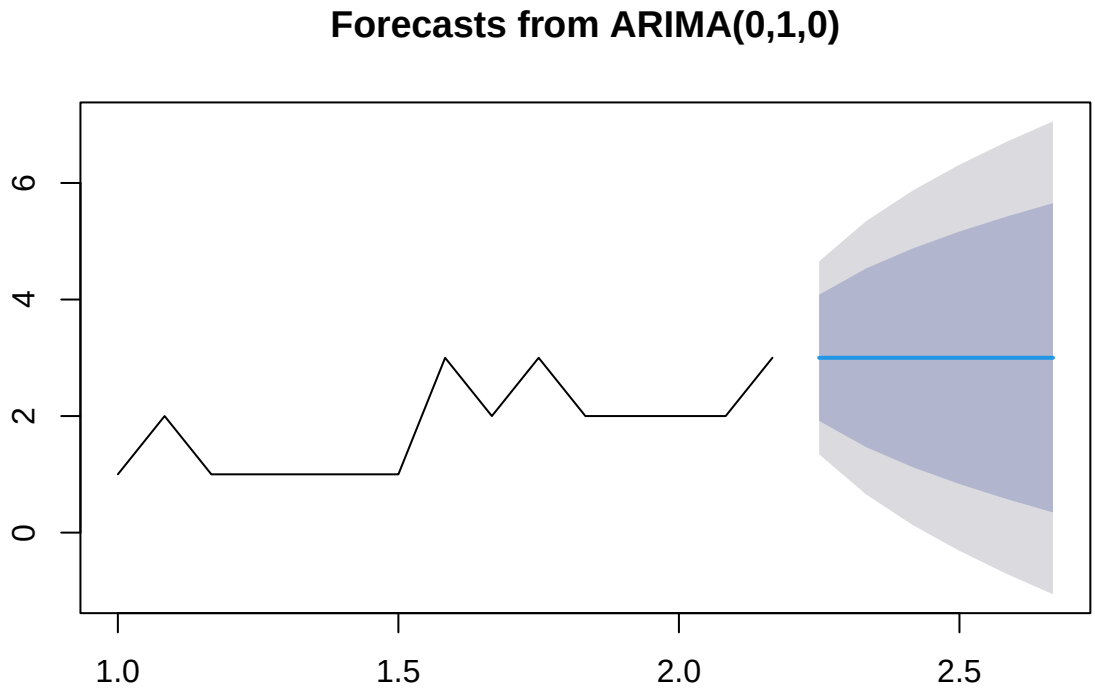

```
#K-menas clustering
set.seed(100)
rfm_cl <- kmeans(rfm.sc, 6, nstart = 20)
rfm$Cluster <- rfm_cl$cluster
# Cluster Summary
rfm %>%
  group_by(Cluster) %>%
  summarise(
    Avg_Recency = mean(Recency),
    Avg_Frequency = mean(Frequency),
    Avg_Monetary = mean(Monetary),
    Count = n()
  )
```

```
## # A tibble: 6 x 5
##   Cluster Avg_Recency Avg_Frequency Avg_Monetary Count
##     <int>       <dbl>         <dbl>        <dbl> <int>
## 1       1        83.6         15.5       911859.   266
## 2       2       656.           6.47       77004.   123
## 3       3         6.5        101      892662684.     2
## 4       4        29.5        385       44783324.    13
## 5       5      1158.           3.91       30494.   130
## 6       6        53.8        156.       5766360.    72
```

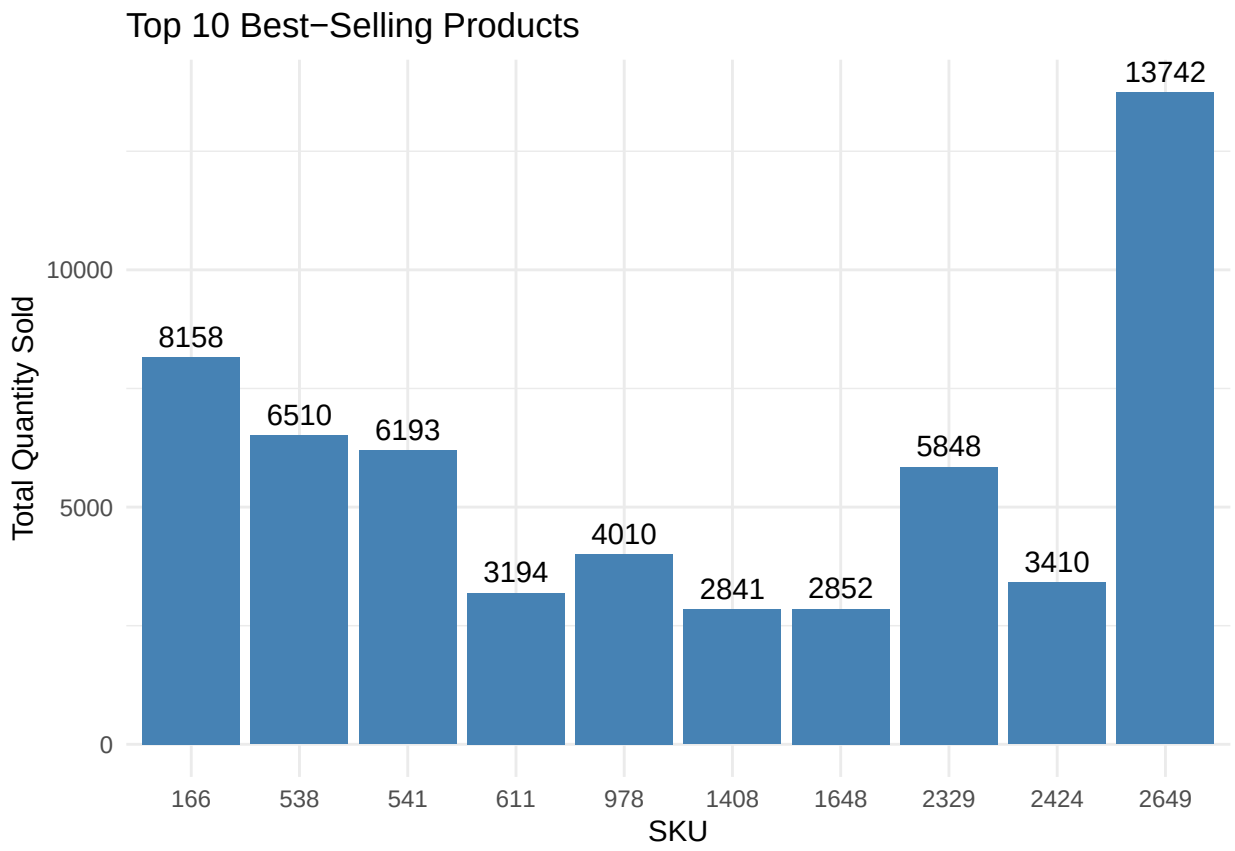
```r
#Cluster Plot
fviz_cluster(rfm_cl, data = rfm.sc)
```



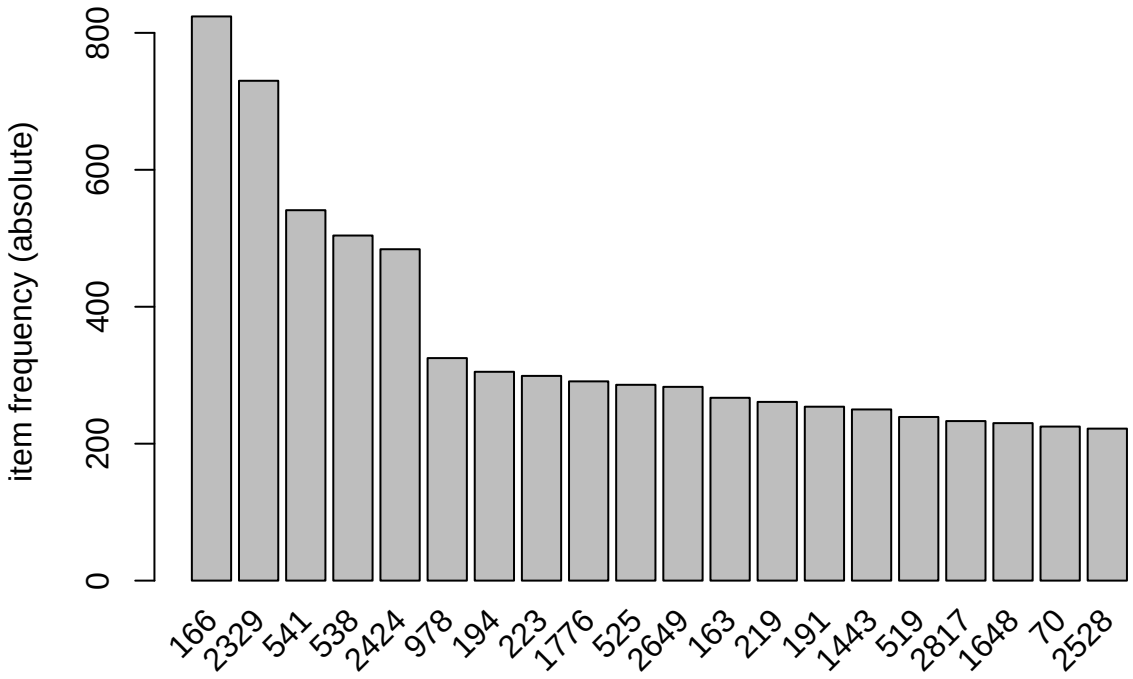

```r
# Product Demand Forecasting
sku_sales <- data %>%
  filter(SKU == 1039) %>%
  group_by(month = floor_date(Purchase_Date, "month")) %>%
  summarise(Demand = sum(Quantity))

ts_data <- ts(sku_sales$Demand, frequency = 12)
fit <- auto.arima(ts_data)
forecast_demand <- forecast(fit, h = 6)

plot(forecast_demand)
```




```
#Top-Selling Products
top_sku <- data %>%
  group_by(SKU) %>%
  summarise(Total_SKU_Sold = sum(Quantity)) %>%
  arrange(desc(Total_SKU_Sold))

top10 <- head(top_sku, 10)

ggplot(top10, aes(x = factor(SKU), y = Total_SKU_Sold)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = Total_SKU_Sold), vjust = -0.5, size = 4) +
  labs(title = "Top 10 Best-Selling Products", x = "SKU", y = "Total Quantity Sold") +
  theme_minimal()
```




```
#Market Basket Analysis
transaction_list <- data %>%
  group_by(DocumentID) %>%
  summarise(items = list(as.character(SKU)))

trans <- as(transaction_list$items, 'transactions')

# Plot item frequency
itemFrequencyPlot(trans, topN = 20, type = "absolute")
```




```
# Generate association rules
rules <- apriori(trans, parameter = list(support = 0.001, confidence = 0.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[2905 item(s), 15752 transaction(s)] done [0.00s].
## sorting and recoding items ... [371 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [30 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
rules_sorted <- sort(rules, by = "lift", decreasing = TRUE)

# Show top 20 rules
inspect(rules_sorted[1:20])
```

```
##      lhs             rhs    support     confidence coverage    lift      count
## [1]  {1556}       => {1039} 0.001079228 0.6296296  0.001714068 431.21417 17
## [2]  {1039}       => {1556} 0.001079228 0.7391304  0.001460132 431.21417 17
## [3]  {868}        => {1310} 0.002285424 0.7826087  0.002920264 224.13913 36
## [4]  {1310}       => {868}  0.002285424 0.6545455  0.003491620 224.13913 36
## [5]  {475}        => {1765} 0.001841036 0.6304348  0.002920264 206.88768 29
## [6]  {1765}       => {475}  0.001841036 0.6041667  0.003047232 206.88768 29
## [7]  {2850}       => {2851} 0.001460132 0.6571429  0.002221940 188.20571 23
## [8]  {2734}       => {2736} 0.001079228 0.5312500  0.002031488 149.43304 17
## [9]  {2886}       => {2862} 0.001333164 0.5833333  0.002285424 148.20430 21
## [10] {2886}       => {2889} 0.001142712 0.5000000  0.002285424 125.01587 18
## [11] {1456, 978}  => {1547} 0.001142712 0.6428571  0.001777552 105.48214 18
## [12] {1408, 1456} => {1547} 0.001079228 0.6296296  0.001714068 103.31173 17
## [13] {1408, 1547} => {1456} 0.001079228 0.5666667  0.001904520  53.77189 17
## [14] {1547, 978}  => {1456} 0.001142712 0.5454545  0.002094972  51.75904 18
## [15] {525, 530}   => {529}  0.001079228 0.5000000  0.002158456  46.60355 17
## [16] {1547, 978}  => {1408} 0.001269680 0.6060606  0.002094972  45.24487 20
## [17] {1456, 1547} => {1408} 0.001079228 0.5666667  0.001904520  42.30395 17
## [18] {519, 525}   => {523}  0.001015744 0.5000000  0.002031488  36.12844 16
## [19] {1408, 1547} => {978}  0.001269680 0.6666667  0.001904520  32.31179 20
## [20] {1456, 1547} => {978}  0.001142712 0.6000000  0.001904520  29.08062 18
```

Strategic Recommendations:

Cluster 3 & 4 (VIPs): Enroll in loyalty programs, offer exclusive deals.

Cluster 1 & 6 (Mid-tier Active): Target with upselling and cross-selling.

Cluster 2 (Low spenders): Send discounts and reminders.

Cluster 5 (Low-value): Reactivation campaigns or exclude from active marketing.