Sporting Goods Store RFM and CLV Analysis

Code

data source: https://www.kaggle.com/datasets/cnezhmar/sporting-goods-store?resource=download (https://www.kaggle.com/datasets/cnezhmar/sporting-goods-store?resource=download)

Hide

```
# Load Packages
library(readxl)
library(dplyr)
library(lubridate)
library(car)
```

Hide

```
# Load datasets
customers <- read_excel("Customer.xlsx")
sales <- read_excel("Sales.xlsx")
product <- read_excel("Product.xlsx")
territories <- read_excel("Territories.xlsx")
```

Hide

```
# OrderDate column into Date format
sales <- sales %>%
  mutate(OrderDate = as.Date(OrderDate))

# join sales with customers and territories datasets
# filter for sales only in the United States
sales_data_us <- sales %>%
  left_join(customers, by = "CustomerKey") %>%
  left_join(territories, by = "SalesTerritoryKey") %>%
  filter(Country == "United States")

# make analysis date 1 day after latest transaction
analysis_date <- max(sales_data_us$OrderDate, na.rm = TRUE) + 1
```

Hide

```
# RFM
rfm <- sales_data_us %>%
  group_by(CustomerKey) %>%
  summarise(
    Recency = as.numeric(analysis_date - max(OrderDate, na.rm = TRUE)),
    Frequency = n_distinct(SalesOrderNumber),  # Count of unique orders
    Monetary = sum(SalesAmount, na.rm = TRUE)
  )

rfm <- rfm %>%
  mutate(
    R_Score = ntile(-Recency, 5),   # Q1 = Least recent, Q5 = Most recent
    F_Score = ntile(Frequency, 5),  # Q1 = Least frequent, Q5 = Most frequent
    M_Score = ntile(Monetary, 5),   # Q1 = Least total spent, Q5 = Most total spent
    RFM_Score = R_Score * 100 + F_Score * 10 + M_Score  # Optional combined score
  )

head(rfm)
```

CustomerKey <dbl>	Recency <dbl>	Frequency <int>	Monetary <dbl>	R_Score <int>	F_Score <int>	M_Score <int>	RFM_Score <dbl>
11012	75	2	81.26	4	4	3	443
11013	260	1	38.98	2	1	2	212
11014	244	2	138.45	2	4	3	243
11015	344	1	2500.97	1	1	5	115
11016	322	1	2332.28	1	1	4	114
11021	339	1	2371.96	1	1	4	114

6 rows

Hide

```
# CLV Calculation (Basic Total Revenue Approach)

clv <- sales_data_us %>%
  group_by(CustomerKey) %>%
  summarise(
    CLV_TotalRevenue = sum(SalesAmount, na.rm = TRUE),           # total lifetime spend
    FirstPurchase = min(OrderDate, na.rm = TRUE),
    LastPurchase = max(OrderDate, na.rm = TRUE),
    NumOrders = n_distinct(SalesOrderNumber),                     # frequency of orders
    AvgOrderValue = CLV_TotalRevenue / NumOrders
  ) %>%
  mutate(
    LifespanDays = as.numeric(LastPurchase - FirstPurchase),
    LifespanMonths = LifespanDays / 30.44,  # rough average month
    MonthlyValue = ifelse(LifespanMonths > 0, CLV_TotalRevenue / LifespanMonths, CLV_TotalRevenue)
  )
```

Hide

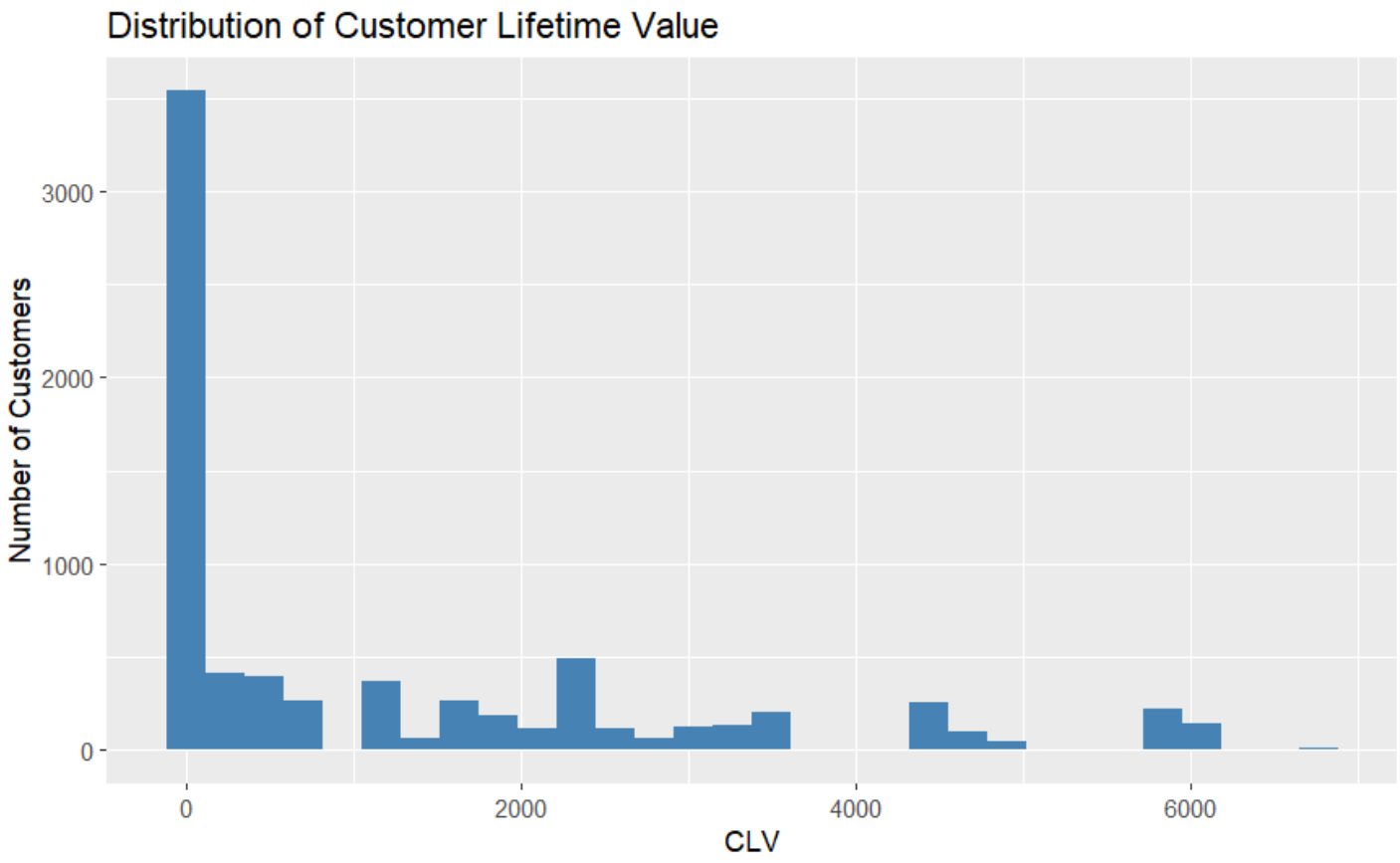
```
clv_segmented <- clv %>%
  mutate(
    CLV_Quintile = ntile(CLV_TotalRevenue, 5),  # Q1 = lowest CLV, Q5 = highest
    CLV_Segment = case_when(
      CLV_Quintile == 5 ~ "Top 20% (High CLV)",
      CLV_Quintile == 4 ~ "Upper-Mid",
      CLV_Quintile == 3 ~ "Middle",
      CLV_Quintile == 2 ~ "Lower-Mid",
      TRUE              ~ "Bottom 20% (Low CLV)"
    )
  )
```

Hide

```
# Visualize CLV

library(ggplot2)

ggplot(clv_segmented, aes(x = CLV_TotalRevenue)) +
  geom_histogram(bins = 30, fill = "steelblue") +
  labs(title = "Distribution of Customer Lifetime Value", x = "CLV", y = "Number of Customers")
```



Hide

```
rfm_clv <- rfm %>%
  left_join(clv, by = "CustomerKey")

rfm_clv %>%
  select(Recency, Frequency, Monetary, CLV_TotalRevenue) %>%
  cor(use = "complete.obs")
```

```
                   Recency  Frequency  Monetary CLV_TotalRevenue
Recency          1.0000000 -0.1092280 0.1773591        0.1773591
Frequency       -0.1092280  1.0000000 0.6455392        0.6455392
Monetary         0.1773591  0.6455392 1.0000000        1.0000000
CLV_TotalRevenue 0.1773591  0.6455392 1.0000000        1.0000000
```

Hide

```
model <- lm(CLV_TotalRevenue ~ Recency + Frequency + Monetary, data = rfm_clv)
summary(model)
```

```


Call:
lm(formula = CLV_TotalRevenue ~ Recency + Frequency + Monetary,
    data = rfm_clv)

Residuals:
     Min       1Q   Median       3Q      Max
-5.9e-10  1.0e-14  6.0e-14  1.3e-13  6.9e-12

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept) -1.436e-11  3.134e-13 -4.581e+01   <2e-16 ***
Recency      1.474e-14  5.085e-16  2.898e+01   <2e-16 ***
Frequency    1.530e-11  2.567e-13  5.960e+01   <2e-16 ***
Monetary     1.000e+00  6.400e-17  1.562e+16   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.806e-12 on 7523 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:      1
F-statistic: 1.562e+32 on 3 and 7523 DF,  p-value: < 2.2e-16
```

Hide

```
rfm_clv_segmented <- rfm %>%
  left_join(clv_segmented, by = "CustomerKey")

# Customers in Top CLV Quintile
top_clv_customers <- rfm_clv_segmented %>%
  filter(CLV_Quintile == 5) %>%
  select(CustomerKey)

# Customers with RFM Score at least 444
top_rfm_customers <- rfm_clv_segmented %>%
  filter(R_Score >= 4, F_Score >= 4, M_Score >= 4) %>%
  select(CustomerKey)

# Customers who are in both Top RFM and CLV
top_both_customers <- rfm_clv_segmented %>%
  filter(R_Score >= 4, F_Score >= 4, M_Score >= 4, CLV_Quintile == 5) %>%
  select(CustomerKey)
```

Hide

```
vif(model) # check for multicollinearity (not an issue here)
```

```
  Recency Frequency  Monetary
 1.132843  1.881103  1.919026
```

Hide

```
# Write a CSV of the segmented data
write.csv(rfm_clv_segmented, "rfm_clv_segmented.csv", row.names = FALSE)
```