# Customer Segmentation using simple RFM Scoring

**2020-01-28**



## Introduction:

One of the most important tasks for any business is to know their customers. In today's world every business needs to offer personalized products and services to its customers or risk losing them.

Customers are both similar and different. It is impossible to have individualized products and services for each customer. Hence the need to segment customers with similar characteristics and have tailored offerings to each group.

There are many characteristics on which customers can be segmented. Common characteristics used are customer behaviour, demography and interests.

Data like customer purchase date and value are readily available with vendors. It therefore makes sense to use them for targeted marketing. Recency of purchase, Frequency of purchases and Monetary value of purchases - popularly referred to as RFM (Receny-Frequency-Monetary) are one of the most effective methods used for customer segmentation.

Here we will explore using a simple RFM scoring method and segment customers into Platinum, Gold, Silver and Bronze customers

### Data:

We will use Superstore Orders data for this analysis.

Load dependent libraries

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

Load data from CSV file

```
# Load store orders data from csv file
orders <-
  read_csv("https://raw.githubusercontent.com/madankundapur/DataAnalytics/master/Data/SuperstoreOrders.csv")
```

The dataset has 9994 observations with 21 variables and contains store orders data for the United States.

```
str(orders)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 9994 obs. of  21 variables:
##  $ Row ID       : num  1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ Order ID     : chr  "CA-2017-152156" "CA-2017-152156" "CA-2017-138688" "US-2016-108966" ...
##  $ Order Date   : chr  "11/8/2017" "11/8/2017" "6/12/2017" "10/11/2016" ...
##  $ Ship Date    : chr  "11/11/2017" "11/11/2017" "6/16/2017" "10/18/2016" ...
##  $ Ship Mode    : chr  "Second Class" "Second Class" "Second Class" "Standard Class" ...
##  $ Customer ID  : chr  "CG-12520" "CG-12520" "DV-13045" "SO-20335" ...
##  $ Customer Name: chr  "Claire Gute" "Claire Gute" "Darrin Van Huff" "Sean O'Donnell" ...
##  $ Segment      : chr  "Consumer" "Consumer" "Corporate" "Consumer" ...
##  $ Country      : chr  "United States" "United States" "United States" "United States" ...
##  $ City         : chr  "Henderson" "Henderson" "Los Angeles" "Fort Lauderdale" ...
##  $ State        : chr  "Kentucky" "Kentucky" "California" "Florida" ...
##  $ Postal Code  : num  42420 42420 90036 33311 33311 ...
##  $ Region       : chr  "South" "South" "West" "South" ...
##  $ Product ID   : chr  "FUR-BO-10001798" "FUR-CH-10000454" "OFF-LA-10000240" "FUR-TA-10000577" ...
##  $ Category     : chr  "Furniture" "Furniture" "Office Supplies" "Furniture" ...
##  $ Sub-Category : chr  "Bookcases" "Chairs" "Labels" "Tables" ...
##  $ Product Name : chr  "Bush Somerset Collection Bookcase" "Hon Deluxe Fabric Upholstered Stacking
Chairs, Rounded Back" "Self-Adhesive Address Labels for Typewriters by Universal" "Bretford CR4500
Series Slim Rectangular Table" ...
##  $ Sales        : num  262 731.9 14.6 957.6 22.4 ...
##  $ Quantity     : num  2 3 2 5 2 7 4 6 3 5 ...
##  $ Discount     : num  0 0 0 0.45 0.2 0 0 0.2 0.2 0 ...
##  $ Profit       : num  41.91 219.58 6.87 -383.03 2.52 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Row ID` = col_double(),
##   ..   `Order ID` = col_character(),
##   ..   `Order Date` = col_character(),
##   ..   `Ship Date` = col_character(),
##   ..   `Ship Mode` = col_character(),
##   ..   `Customer ID` = col_character(),
##   ..   `Customer Name` = col_character(),
##   ..   Segment = col_character(),
##   ..   Country = col_character(),
##   ..   City = col_character(),
##   ..   State = col_character(),
##   ..   `Postal Code` = col_double(),
##   ..   Region = col_character(),
##   ..   `Product ID` = col_character(),
##   ..   Category = col_character(),
##   ..   `Sub-Category` = col_character(),
##   ..   `Product Name` = col_character(),
##   ..   Sales = col_double(),
##   ..   Quantity = col_double(),
##   ..   Discount = col_double(),
##   ..   Profit = col_double()
##   .. )
```

Data variable names have spaces in them and as a practise it is good to avoid spaces. Also note that the variable names are in proper casing - we will retain and follow that convention for naming data variables.

```
names(orders)<-str_replace_all(names(orders), c(" " = ""))
```

*OrderDate* variable is of type ‘character’. Changing it to ‘date’

```
orders$OrderDate <- as.Date(orders$OrderDate, "%m/%d/%Y")
class(orders$OrderDate)
```

```
## [1] "Date"
```

To keep data tidy check for duplicates and filter them out.

```
duplicates <- which(duplicated(orders))
duplicates
```

```
## integer(0)
```

```
# No duplicates exist in data
rm(duplicates)
```

Data that we need for RFM analysis is *OrderDate* and *Sales* amount by customer. The dataset has order details at the product level which we don't need. Let us aggregate *Sales* amount and select only necessary variables for further analysis

```
orders <- orders %>%
  group_by(CustomerID, OrderID , OrderDate) %>%
  summarize(Sales = sum(Sales)) %>%
  select(CustomerID, OrderID , OrderDate, Sales)

# Checking if the orders are equal to the observations in the dataset
length(unique(orders$OrderID ))
```

```
## [1] 5009
```

```
nrow(orders)
```

```
## [1] 5009
```

Order dates range from Jan 2015 through Dec 2018. Compute maximum date from the dataset. This will help compute *DaysSincePurchase* and *Recency*. Note that a day is added to the maximum date to ensure that there are no zeroes calculated (applies to purchases made on the last day).

```
range(orders$OrderDate)
```

```
## [1] "2015-01-03" "2018-12-30"
```

```
max_date <- max(orders$OrderDate)+1
```

Compute *PurchaseYear* - which is year part of order date and *DaysSincePurchase* - which is the difference between order date and the maximum date in the dataset

```
orders <- orders %>%
  mutate(PurchaseYear = as.numeric(format(OrderDate, "%Y")),
         DaysSincePurchase = as.numeric(difftime(max_date, OrderDate,"days")))

rm(max_date)
```

## Compute Recency, Frequency and Monetary Value:

For each customer compute RFM values:

- *Recency* is the duration in days since the last purchase made by the customer
- *Frequency* is the number of distinct orders by customer
- *Monetary* value is total sales amount for the customer

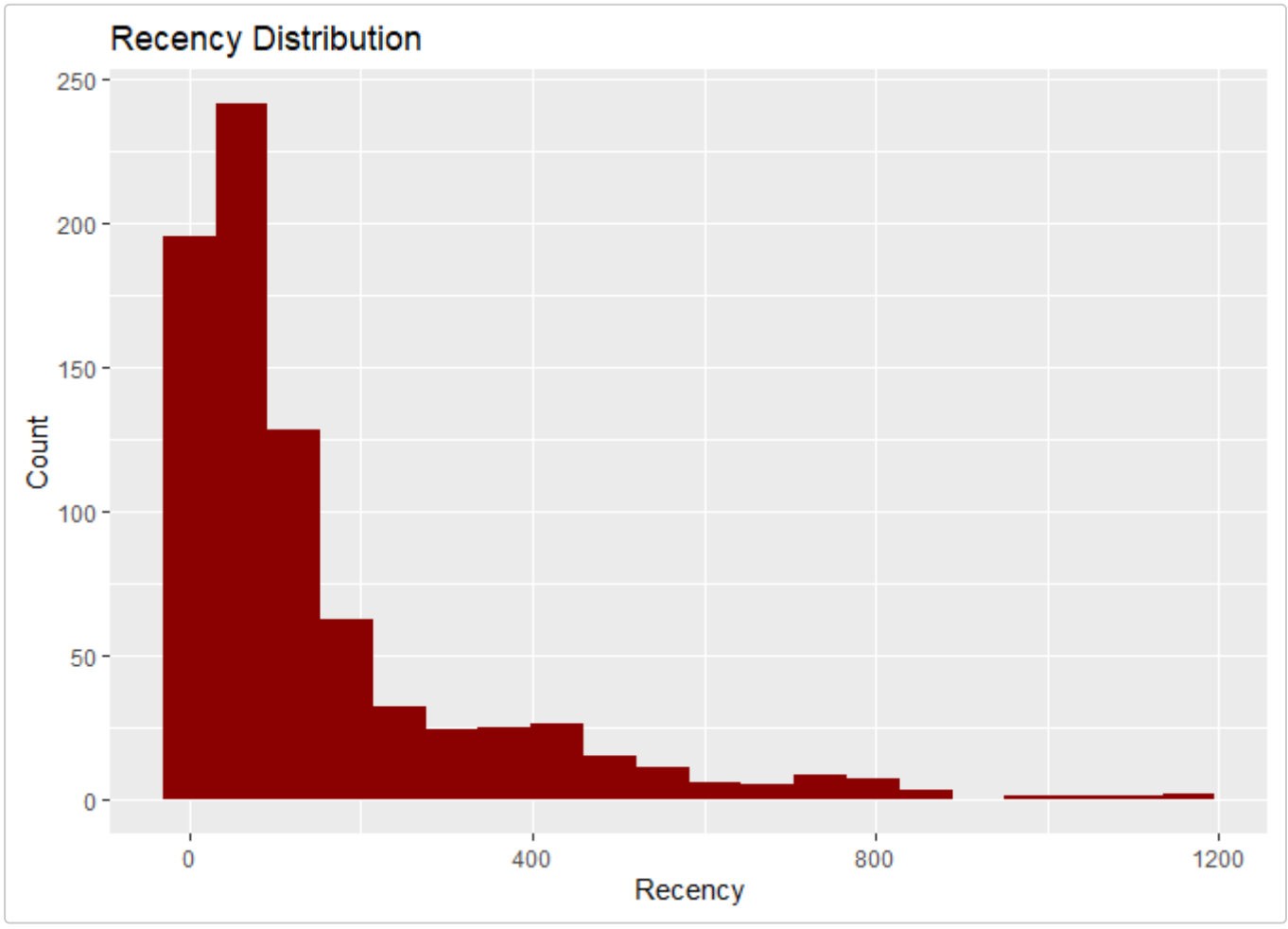
```
customers <- orders %>%
  group_by(CustomerID) %>%
  summarise(Recency = min(DaysSincePurchase),
            Frequency = n_distinct(OrderID),
            Monetary = sum(Sales))

knitr::kable(summary(customers))
```

| CustomerID | Recency | Frequency | Monetary |
|---|---|---|---|
| Length:793 | Min. : 1.0 | Min. : 1.000 | Min. : 4.833 |
| Class :character | 1st Qu.: 31.0 | 1st Qu.: 5.000 | 1st Qu.: 1146.050 |
| Mode :character | Median : 76.0 | Median : 6.000 | Median : 2256.394 |
| NA | Mean : 147.8 | Mean : 6.317 | Mean : 2896.848 |
| NA | 3rd Qu.: 184.0 | 3rd Qu.: 8.000 | 3rd Qu.: 3785.276 |
| NA | Max. :1166.0 | Max. :17.000 | Max. :25043.050 |

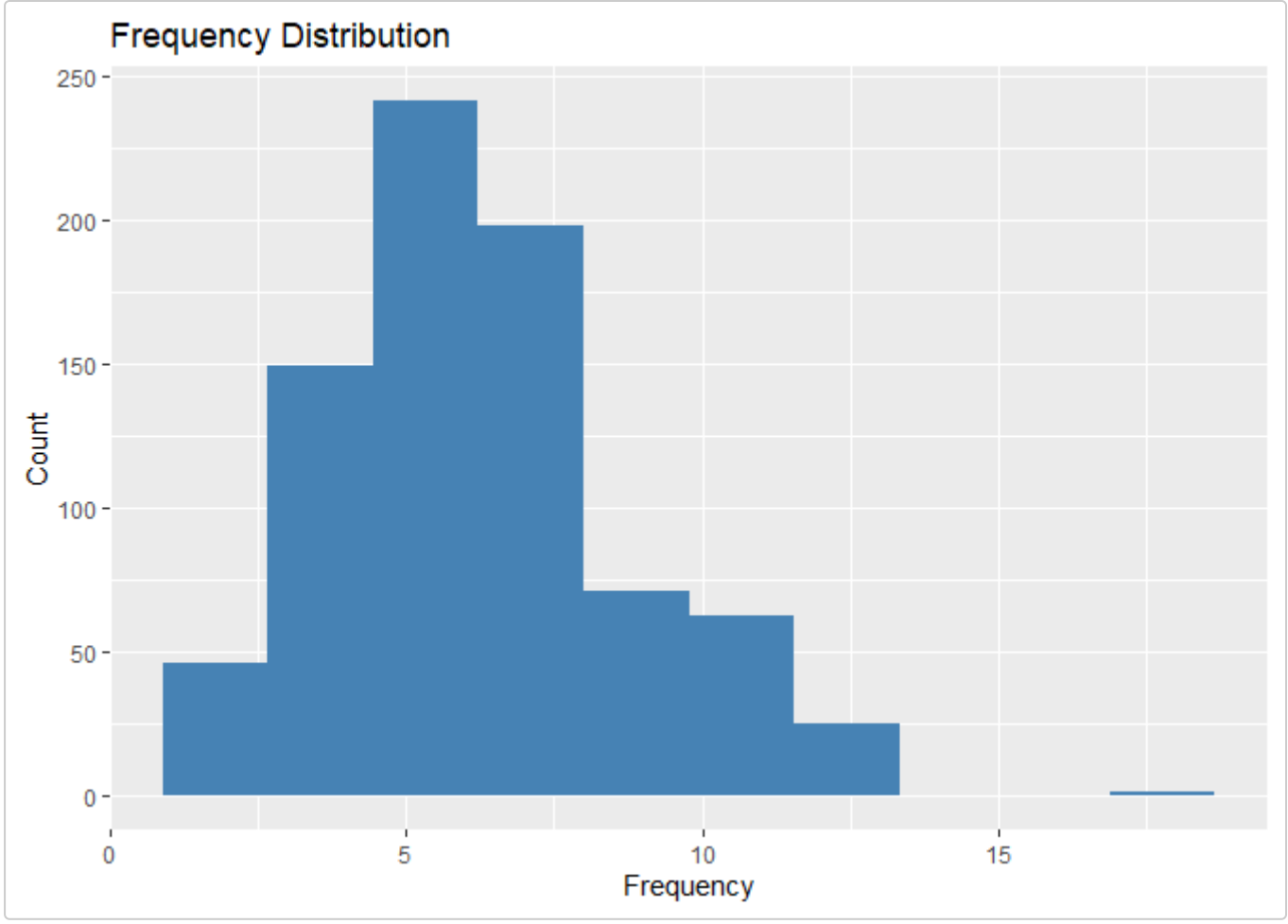
Plot distribution for Recency, Frequency and Monetary Value to explore RFM data

```
customers %>% ggplot(aes(Recency))  +
  geom_histogram(bins=20,fill = "darkred") +
  labs(x = "Recency", y = "Count", title = "Recency Distribution")
```



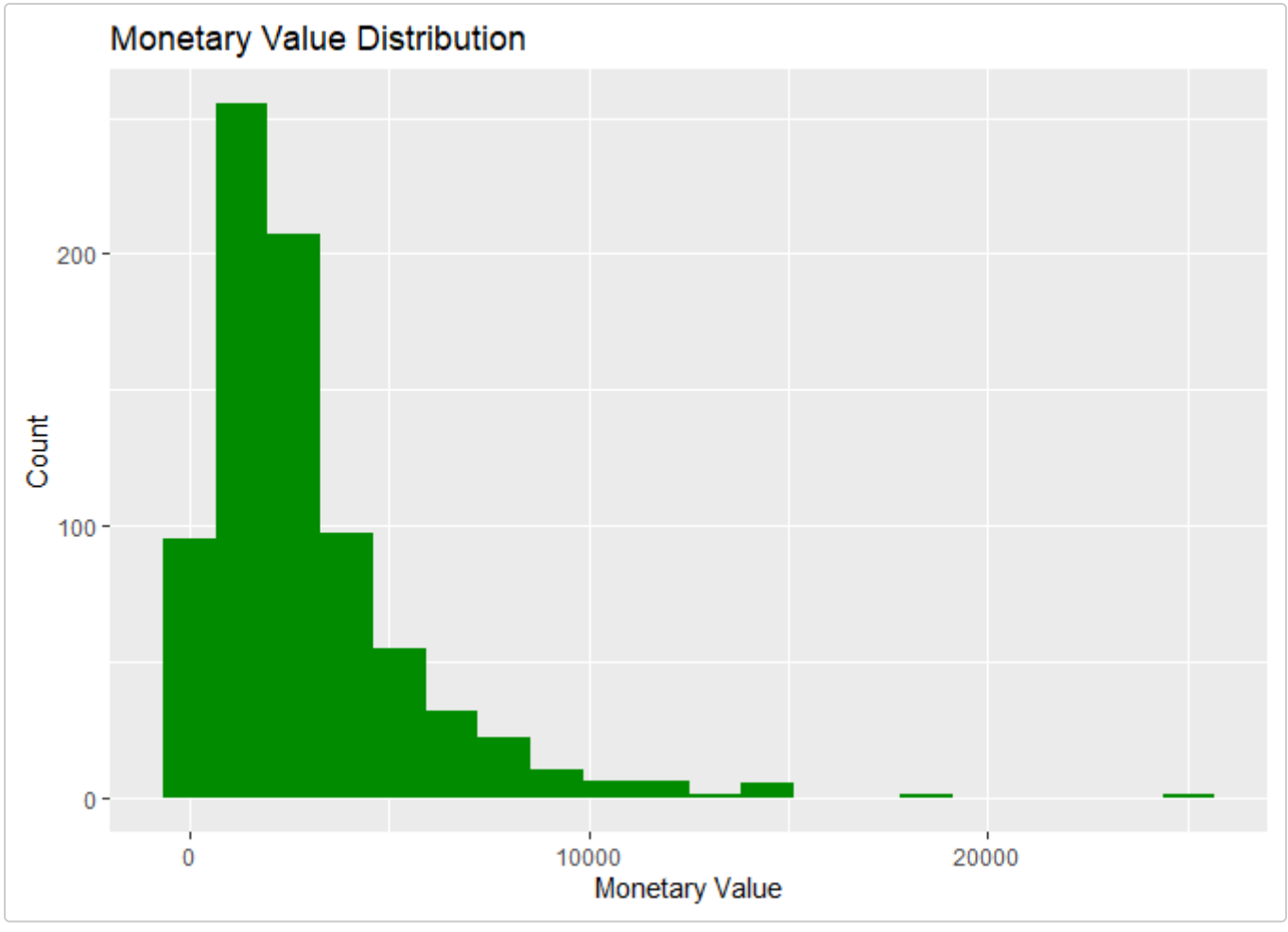

From the Recency plot, more than 80% of customers have been active in the last one year, which is a good sign.

```
customers %>% ggplot(aes(Frequency))  +
  geom_histogram(bins=10,fill = "steelblue")+
  labs(x = "Frequency", y = "Count", title = "Frequency Distribution")
```




From the Frequency plot, the values are more-or-less distributed and the range is between 1 and 13 with an outlier of 17.

```
customers %>% ggplot(aes(Monetary))  +
  geom_histogram(bins=20,fill = "green4") +
  labs(x = "Monetary Value", y = "Count", title = "Monetary Value Distribution")
```



From the Monetary value plot, more than 97% of customers have spent less than $10000 across years.

Since the scale of values are very different for Recency, Frequency and Monetary. Let us remove the skew and standardise the RFM values.

```
customers$RecencyZ <- scale(log(customers$Recency), center=TRUE, scale=TRUE)
customers$FrequencyZ <- scale(log(customers$Frequency), center=TRUE, scale=TRUE)
customers$MonetaryZ <- scale(log(customers$Monetary), center=TRUE, scale=TRUE)
```

We now have a tidy dataset with 793 observations of 8 variables to work with.

---

# Analysis:

## Simple RFM Scoring:

Using `ntile` analytic function divide Recency, Frequency and Monetary value into 4 buckets.

A value of 1 means Bronze, 2 means Silver, 3 means Gold and 4 means Platinum customer

Recency score `RScore` is based on the recency of purchase. Lower the Recency, the better the customer - note the code uses `desc` for the `ntile` function.

Frequency score `FScore` is based on the number of orders made by the customer. Higher the frequency, the better the customer.

Monetary value score `MScore` is based on the total value of sales by customer. Higher the Recency, the better the customer.

`RFMScore` is the mean of Recency, Frequency & Monetary Scores

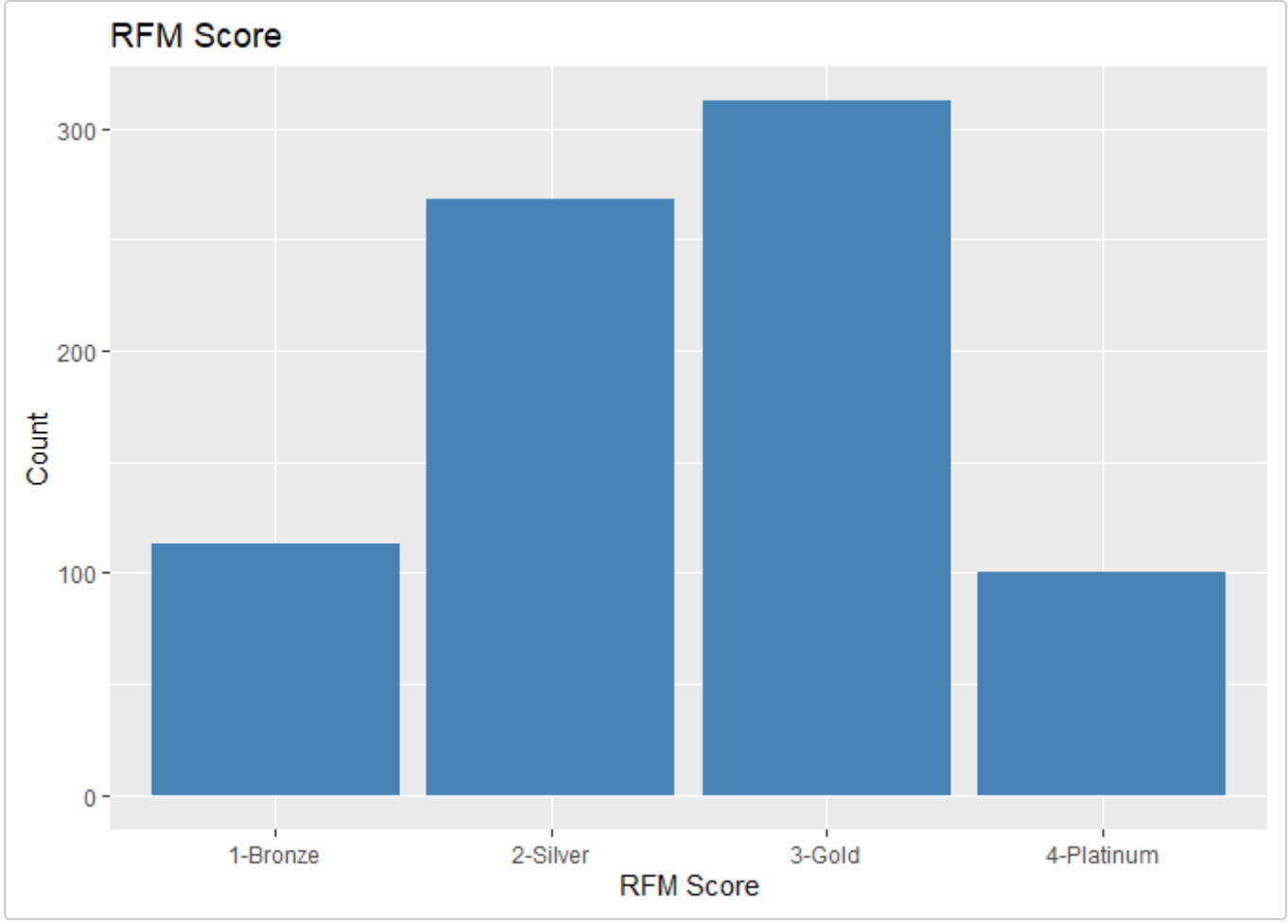
`RFMScoreLabel` is a label assigned based on `RFMScore`

```
customers <- customers %>%
  mutate(RScore = ntile(desc(Recency),4),
          FScore = ntile(Frequency,4),
          MScore = ntile(Monetary,4),
          RFMScore = round((RScore+FScore+MScore)/3,0),
          RFMScoreLabel = case_when(RFMScore == 1 ~ "1-Bronze",
                                    RFMScore == 2 ~ "2-Silver",
                                    RFMScore == 3 ~ "3-Gold",
                                    RFMScore == 4 ~ "4-Platinum"))

table(customers$RFMScoreLabel)
```

```
##
##   1-Bronze   2-Silver     3-Gold 4-Platinum
##        113        268        312        100
```

```
customers %>% ggplot(aes(RFMScoreLabel)) +
  geom_bar(fill = "steelblue") +
  labs(x = "RFM Score", y = "Count", title = "RFM Score")
```




## Conclusion:

```
customers %>%
  group_by(RFMScoreLabel) %>%
  summarize(Rmean = mean(Recency),
            Fmean = mean(Frequency),
            Mmean=mean(Monetary),
            Msum=sum(Monetary))
```

```
## # A tibble: 4 x 5
##   RFMScoreLabel Rmean Fmean Mmean     Msum
##   <chr>         <dbl> <dbl> <dbl>    <dbl>
## 1 1-Bronze      411.   3.21  804.   90904.
## 2 2-Silver      163.   5.11 1771.  474676.
## 3 3-Gold         78.9  7.46 3872. 1208031.
## 4 4-Platinum     24.7  9.51 5236.  523590.
```

### Bronze customers:

- They are about 14% of customers. Have on an average bought products more than a year back, average frequency is about 3 times with an average purchase value of about $800.
- While the revenue from this segment is about 4% they still are about 14% of the customers and action needs to be taken to attract them with discounts and offers.

### Silver customers:

- This segment has about 34% customers and brings about 20% of the revenue. On an average these customers bought products 6 months back, average frequency is about 5 times with an average purchase value of about $1800.

- This is a significant chunk and action needs to be taken to retain them and move towards Gold customers

## Gold customers:

- This segment has about 40% customers and brings more than half the business. On an average these customers bought products 3 months back, average frequency is about 7 times with an average purchase value of about $3900.
- This is defintely the most important customer segment and personalized attention needs to be given to each one of them. Further segmentation of these customers would yield more insights

## Platinum customers:

- This segment has about 13% customers and brings about 23% of the revenue. On an average these customers bought things as recent as 1 month back, average frequency is about 10 times with an average purchase value of about $5200.
- These customers are willing to spend and buy more frequently. Marketing campaigns to increase revenue from these customers are a must.

R-Markdown