Customer Segmentation

Riya Agarwal

12/14/2016

Objective

To divide customers into groups that share certain characteristics. This project aims to identify high value and low value customers. I will be using k-means clustering to segment customers based on their purchasing habits.

What is k-means clustering?

The k-means clustering algorithm works by finding like groups based on Euclidean distance, a measure of distance or similarity. We can select k groups to cluster, and the algorithm finds the best centroids for those groups. We can then use those groups to determine which factors group members relate.

Data

I will be using the dataset from the University of California at Irvine machine learning repository at: https://archive.ics.uci.edu/ml/datasets/Online+Retail (https://archive.ics.uci.edu/ml/datasets/Online+Retail). This data represents online transactions that relates to a UK based non-store online retail agency.

Attributes

- InvoiceNo: Invoice number. If the number starts with ???c???, it indicates a cancellation.
- StockCode: Product code. It is uniquely assigned to each distinct product.
- Description: Product name.
- Quantity: The quantities of a product per transaction.
- InvoiceDate: Invice Date and time.
- UnitPrice: Product price per unit.
- CustomerID: It is a number uniquely assigned to each customer.
- Country:It is the name of the country where each customer resides.

Let us get started!

Import the data

```
library(openxlsx)
raw.data <- read.xlsx("Online Retail.xlsx", sheet = 1)
head(raw.data)
```

```
##   InvoiceNo StockCode                          Description Quantity
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6
## 2    536365     71053                 WHITE METAL LANTERN        6
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2
##   InvoiceDate UnitPrice CustomerID        Country
## 1    40513.35      2.55      17850 United Kingdom
## 2    40513.35      3.39      17850 United Kingdom
## 3    40513.35      2.75      17850 United Kingdom
## 4    40513.35      3.39      17850 United Kingdom
## 5    40513.35      3.39      17850 United Kingdom
## 6    40513.35      7.65      17850 United Kingdom
```

To avoid loading the data into R again, I keep the data preserved in the variable raw.data

```
data <- raw.data
data$InvoiceDate<- as.Date(data$InvoiceDate, origin="1900-01-01")
```

Remove for Null values for Customer ID as they are not useful for the analysis.

```
data <- subset(data, !is.na(data$CustomerID))
```

To get an accurate customer behavior let us work region wise.

```
table(data$Country)
```

```
##
##            Australia              Austria              Bahrain
##                 1259                  401                   17
##              Belgium               Brazil               Canada
##                 2069                   32                  151
##      Channel Islands               Cyprus       Czech Republic
##                  758                  622                   30
##              Denmark                 EIRE   European Community
##                  389                 7485                   61
##              Finland               France              Germany
##                  695                 8491                 9495
##               Greece              Iceland               Israel
##                  146                  182                  250
##                Italy              Japan              Lebanon
##                  803                  358                   45
##            Lithuania               Malta          Netherlands
##                   35                  127                 2371
##               Norway              Poland             Portugal
##                 1086                  341                 1480
##                  RSA       Saudi Arabia            Singapore
##                   58                   10                  229
##                Spain              Sweden          Switzerland
##                 2533                  462                 1877
##                  USA United Arab Emirates       United Kingdom
##                  291                   68               361878
##          Unspecified
##                  244
```

United Kingdom seems to be having the most number of customers. Let us choose UK for our analysis!

```
data <- subset(data, Country == "United Kingdom")
```

Now let us look for any cancelled transactions.

```
data$item.cancelled <- grepl("C", data$InvoiceNo, fixed=TRUE)
data$purchase.invoice <- ifelse(data$item.cancelled=="TRUE", 0, 1)
```

Let us create a new customer level dataset

```
customer_data <- as.data.frame(unique(data$CustomerID))
names(customer_data) <- "CustomerID"
```

Let us check how recently the customers have made a purchase?

```
data$recency <- as.Date("2011-12-10") - as.Date(data$InvoiceDate)
```

Let us only consider customers who have not made any cancellations.

```
temp_data <- subset(data, purchase.invoice == 1)
```

Let us find the number of days since the most recent purchase based on customers who did not cancel any order.

```
recency <- aggregate(recency ~ CustomerID, data=temp_data, FUN=min, na.rm=TRUE)
remove(temp_data)
```

Now let us add the recency to our customer level data:

```
customer_data <- merge(customer_data, recency, by="CustomerID", all=TRUE, sort=TRUE)
remove(recency)
customer_data$recency <- as.numeric(customer_data$recency)
customer_data<-subset(customer_data, !is.na(customer_data$recency))
data<-subset(data, !is.na(data$recency))
customer_data <- subset(customer_data, recency > 0)
data<-subset(data, recency > 0)
range(customer_data$recency)
```

```
## [1]   0.2743056 371.5881944
```

Now let us check how frequently have the customers made a purchase?

```
customer.invoices <- subset(data, select = c("CustomerID","InvoiceNo", "purchase.invoic
e"))
customer.invoices <- customer.invoices[!duplicated(customer.invoices), ]
customer.invoices <- customer.invoices[order(customer.invoices$CustomerID),]
row.names(customer.invoices) <- NULL
```

Number of invoices for a customer who has made no cancellations:

```
annual.invoices <- aggregate(purchase.invoice ~ CustomerID, data=customer.invoices,
FUN=sum, na.rm=TRUE)
names(annual.invoices)[names(annual.invoices)=="purchase.invoice"] <- "frequency"
```

Add the frequency to the customers data:

```
customer_data <- merge(customer_data, annual.invoices, by="CustomerID", all=TRUE, sort=T
RUE)
remove(customer.invoices, annual.invoices)
```

Let us get a range for the number of purchases made by a customer

```
range(customer_data$frequency)
```

```
## [1]   0 206
```

Remove the customers who have made no purchases.

```
customer_data <- subset(customer_data, frequency > 0)
```

Revenue generated by a customer:

How much have the customers spent on each Invoice?

```
data$Amount <- data$Quantity * data$UnitPrice
```

Now let us aggregate this amount for each customer:

```
annual.revenue<-aggregate(Amount ~ CustomerID, data=data, FUN=sum, na.rm=TRUE)
names(annual.revenue)[names(annual.revenue)=="Amount"] <- "Revenue"
```

Now, let us add this revenue generated by each customer to our customers dataset:

```
customer_data <- merge(customer_data, annual.revenue, by="CustomerID", all.x=TRUE,
sort=TRUE)
remove(annual.revenue)
```

Let's get a range for the revenue generated by the customers:

```
range(customer_data$Revenue)
```

```
## [1]  -1165.3 244953.0
```

The negative revenue means the customers have been returning some items from previous purchases. Let us reset these values to 0, meaning they did not contribute to the revenue.

```
customer_data$Revenue <- ifelse(customer_data$Revenue < 0, 0, customer_data$Revenue)
range(customer_data$Revenue)
```

```
## [1]      0 244953
```

Standardize the data:

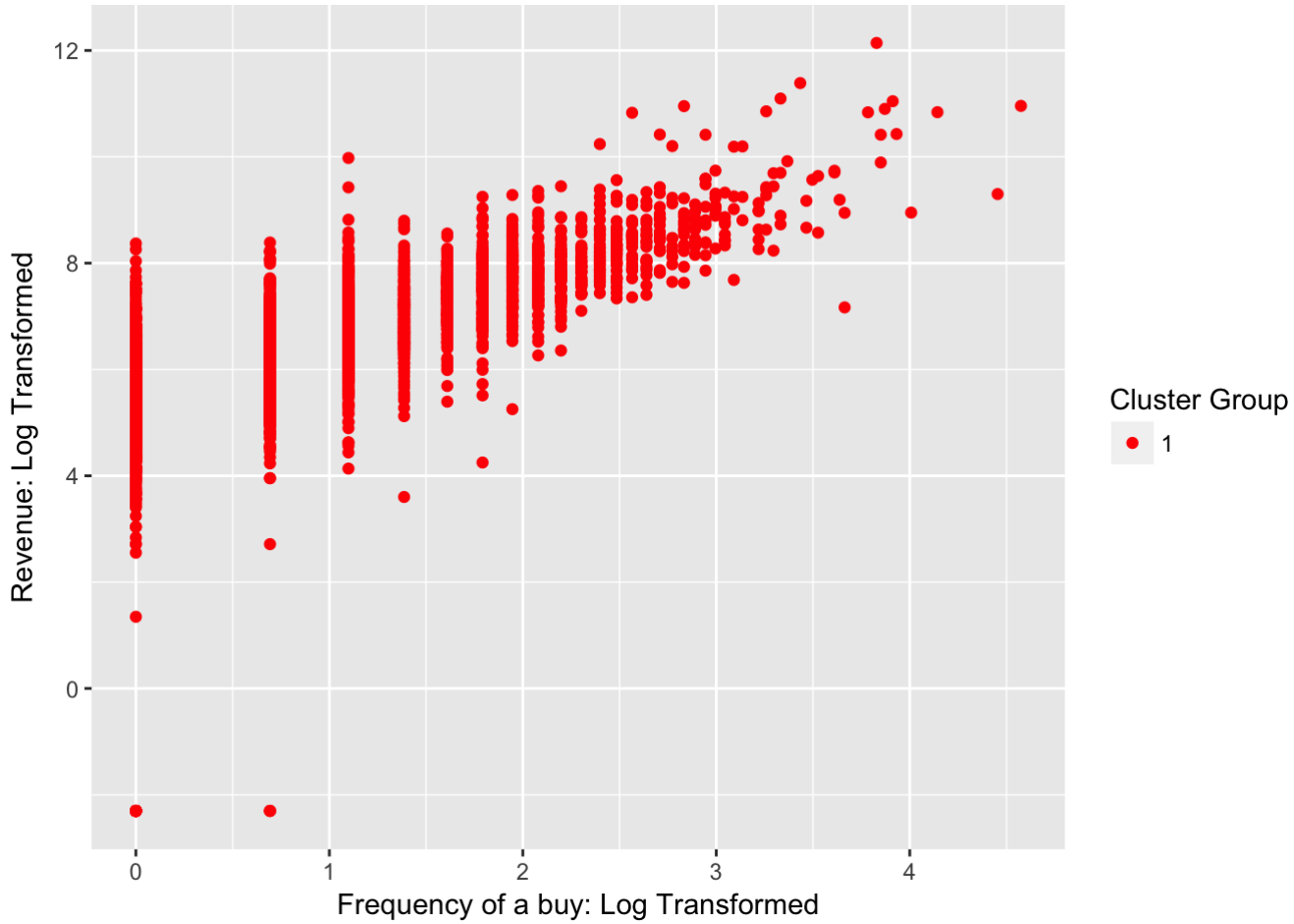
In order for k-means to work efficiently I need continuous variables that are normally-distributed. Without standardizing the variables with larger variances will make the results biased. Thus, I log transform the input variables to reduce positive skew and then standardize them as z-scores.

```
# Log-transform positively-skewed variables
customer_data$recency.log <- log(customer_data$recency)
customer_data$frequency.log <- log(customer_data$frequency)
customer_data$Revenue.log <- customer_data$Revenue + 0.1
customer_data$Revenue.log <- log(customer_data$Revenue.log)

# Z-scores
customer_data$recency.z <- scale(customer_data$recency.log, center=TRUE, scale=TRUE)
customer_data$frequency.z <- scale(customer_data$frequency.log, center=TRUE, scale=TRUE)
customer_data$Revenue.z <- scale(customer_data$Revenue.log, center=TRUE, scale=TRUE)

customer_data<-na.omit(customer_data)
```

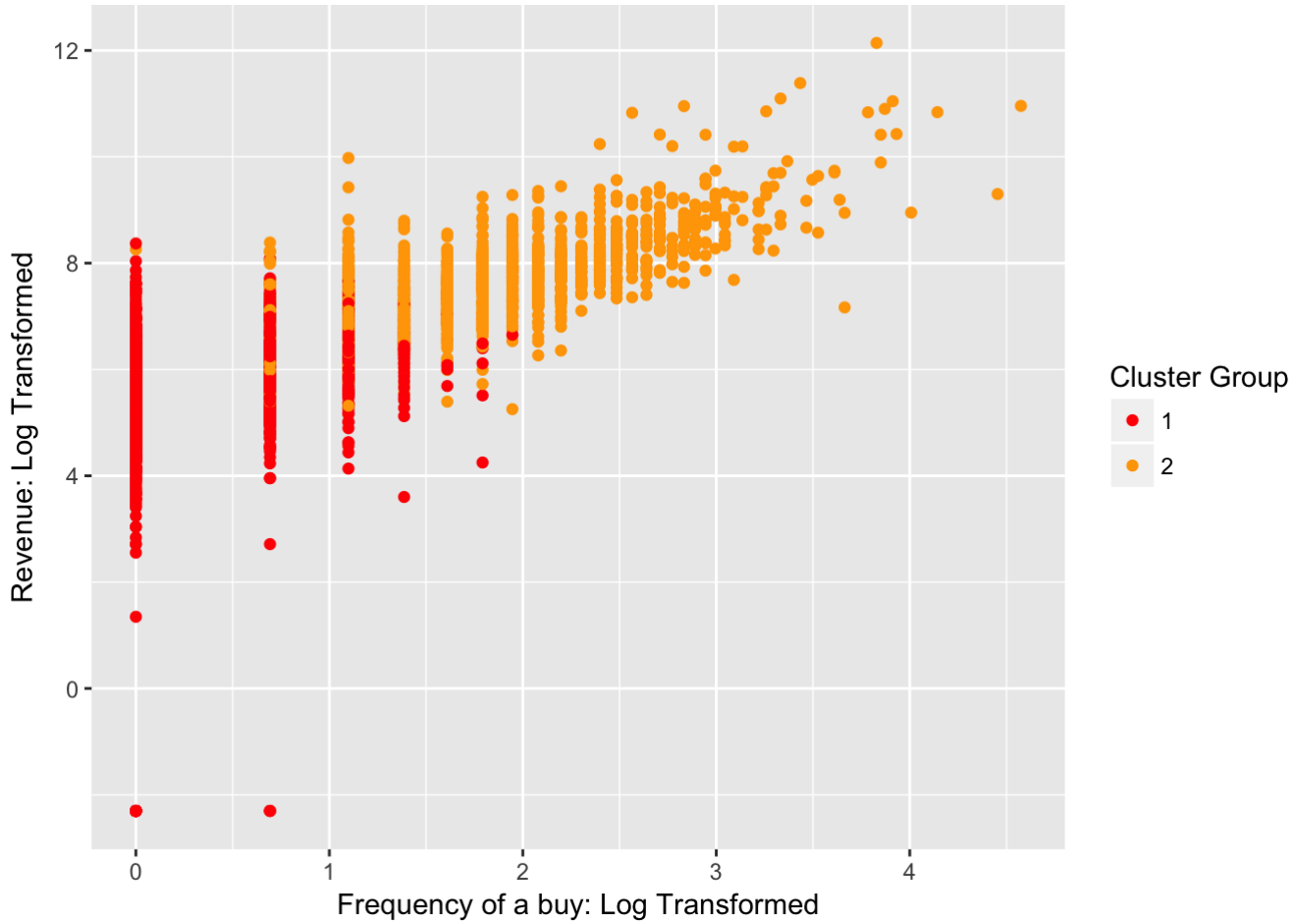
Perform k-means clustering:

I do not know the number of clusters to choose for this problem. I want to run a for loop to run the k-means with a range of clusters, giving me a variety of graphs of the clusters and their corresponding average and variances.Based on that I can look at the data and decide what is the most optimum number of clusters giving me results.

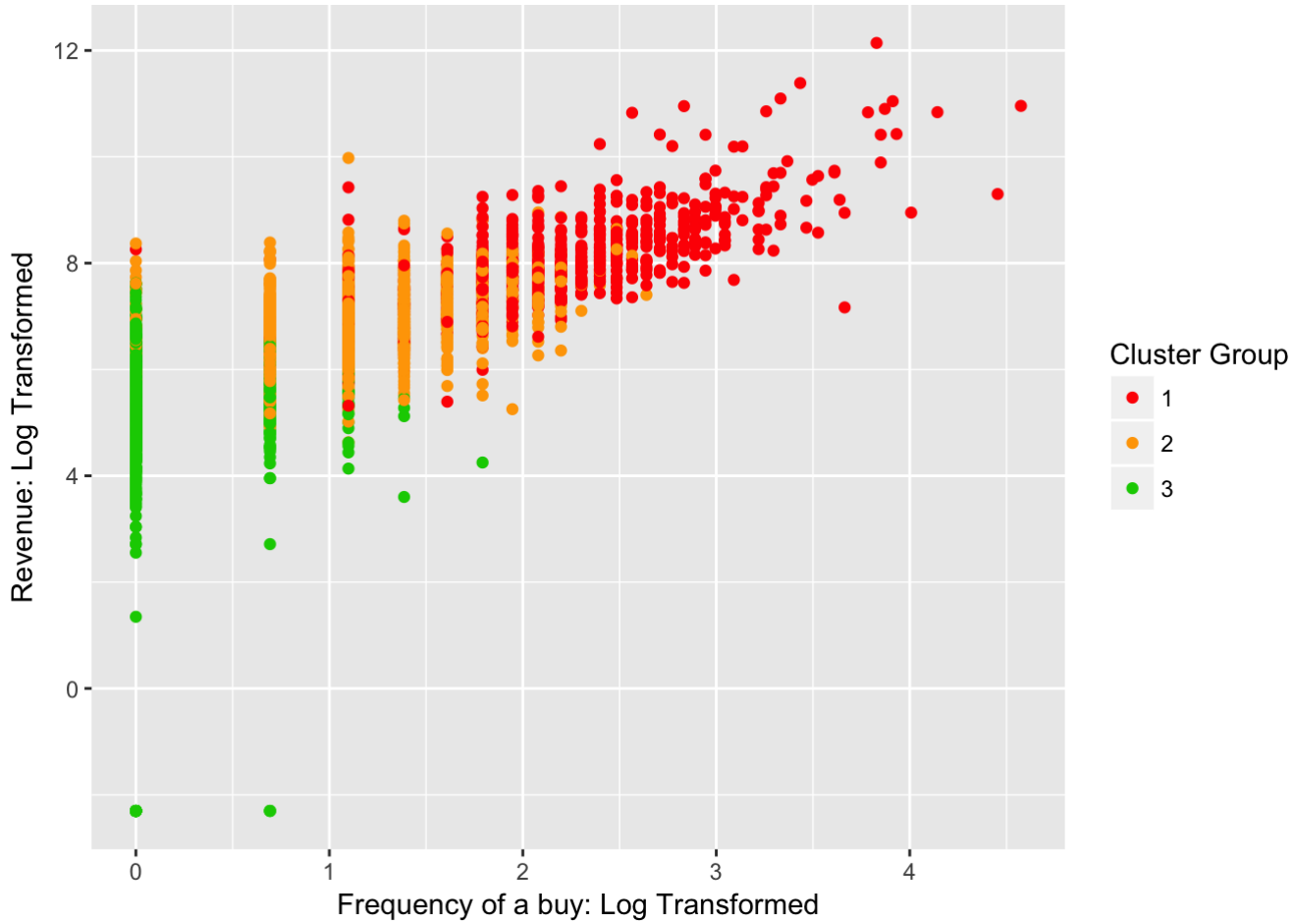
```
# Loading the required libraries
library(plyr)
library(ggplot2)

# Using my standardized data for k-means
d <- customer_data[,8:10]

# I am setting the maximum number of clusters I want to try with as 10

j <- 10
models <- data.frame(k=integer(),
                     tot.withinss=numeric(),
                     betweenss=numeric(),
                     totss=numeric(),
                     rsquared=numeric())

for (k in 1:j ) {

    output <- kmeans(d, centers = k, nstart = 20)


    # Add the cluster assignment to each of the customers
    var.name<-paste("cluster", k, sep="_")
    customer_data[,(var.name)] <- output$cluster
    customer_data[,(var.name)] <- factor(customer_data[,(var.name)], levels = c(1:k))


     # Create graphs for the clusters generated
    cluster_plot <- ggplot(customer_data, aes(x = frequency.log, y = Revenue.log))
    cluster_plot <- cluster_plot + geom_point(aes(colour = customer_data[,(var.name)]))
    colors <- c('red','orange','green3','deepskyblue','blue','grey','hotpink1','firebric
k4','gold2','black')
    cluster_plot <- cluster_plot + scale_colour_manual(name = "Cluster Group", values=co
lors)
    cluster_plot <- cluster_plot + xlab("Frequency of a buy: Log Transformed")
    cluster_plot <- cluster_plot + ylab("Revenue: Log Transformed")

    # Print the cluster graph

    print(cluster_plot)

    # Let's find out the mean values or centers for each of the clusters for the 3 metri
cs we wish to observe
    # We are using the median because the data is heavily skewed, we are using the non-s
tandardized data for this!
    cluster_centers <- ddply(customer_data, .(customer_data[,(var.name)]), summarize,
                             Revenue=round(median(Revenue),2),
                             frequency=round(median(frequency),1),
                             recency=round(median(recency), 0))
    names(cluster_centers)[names(cluster_centers)=="customer_data[, (var.name)]"] <- "Cl
uster"
    print(cluster_centers)
}
```



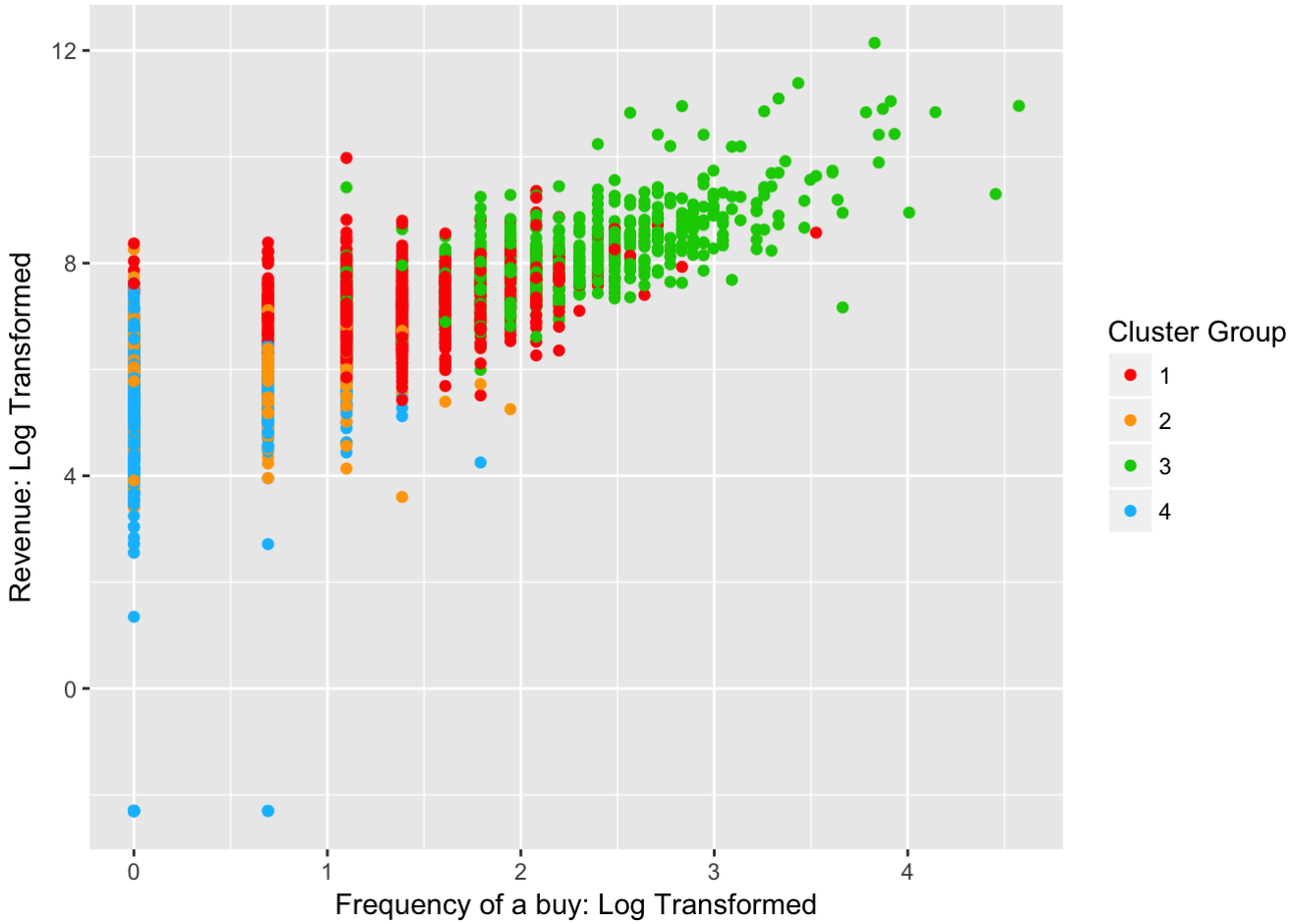

```
##   Cluster Revenue frequency recency
## 1       1  613.95         2      51
```



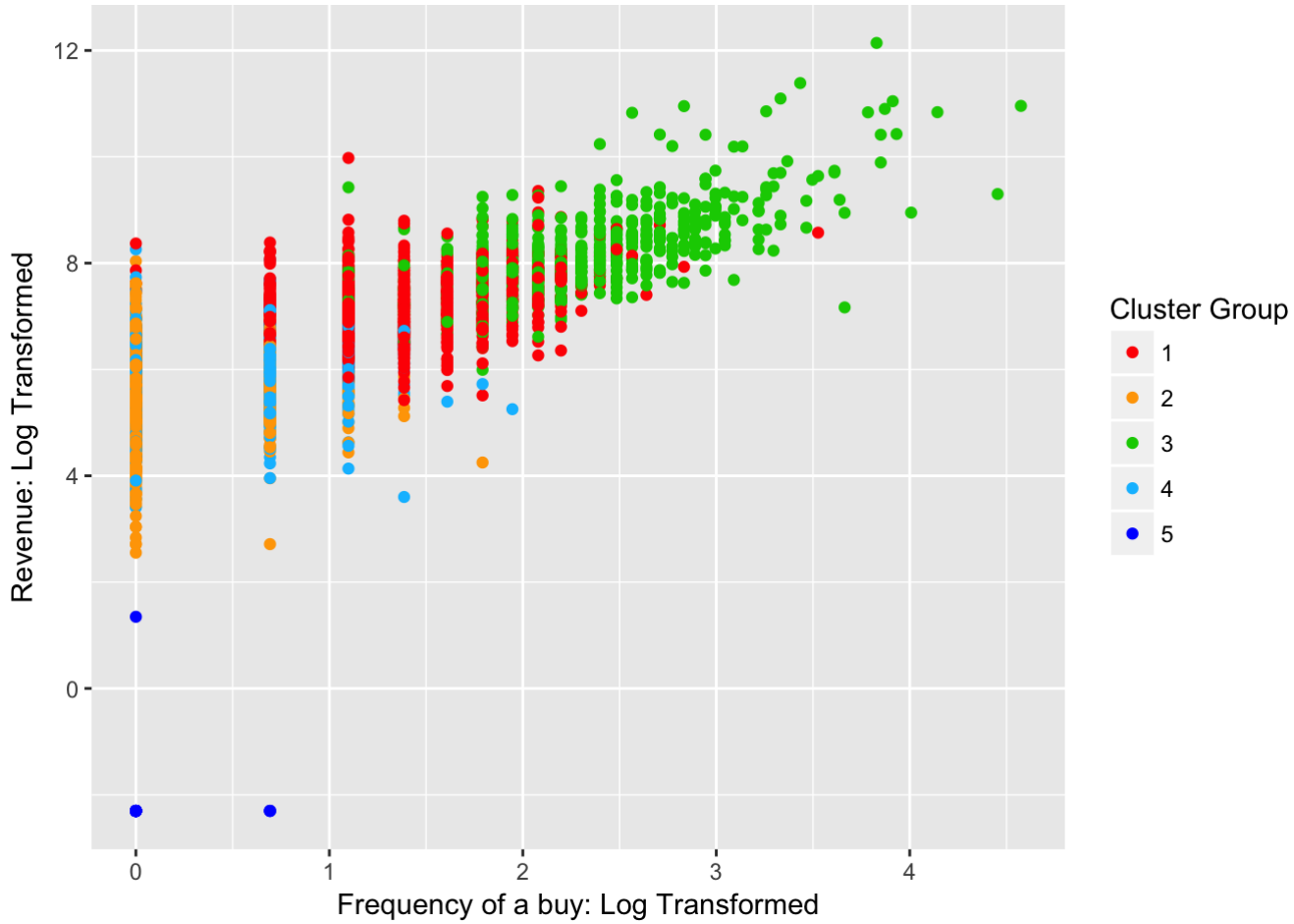

```
##   Cluster Revenue frequency recency
## 1       1  327.27         1     105
## 2       2 1776.81         5      16
```



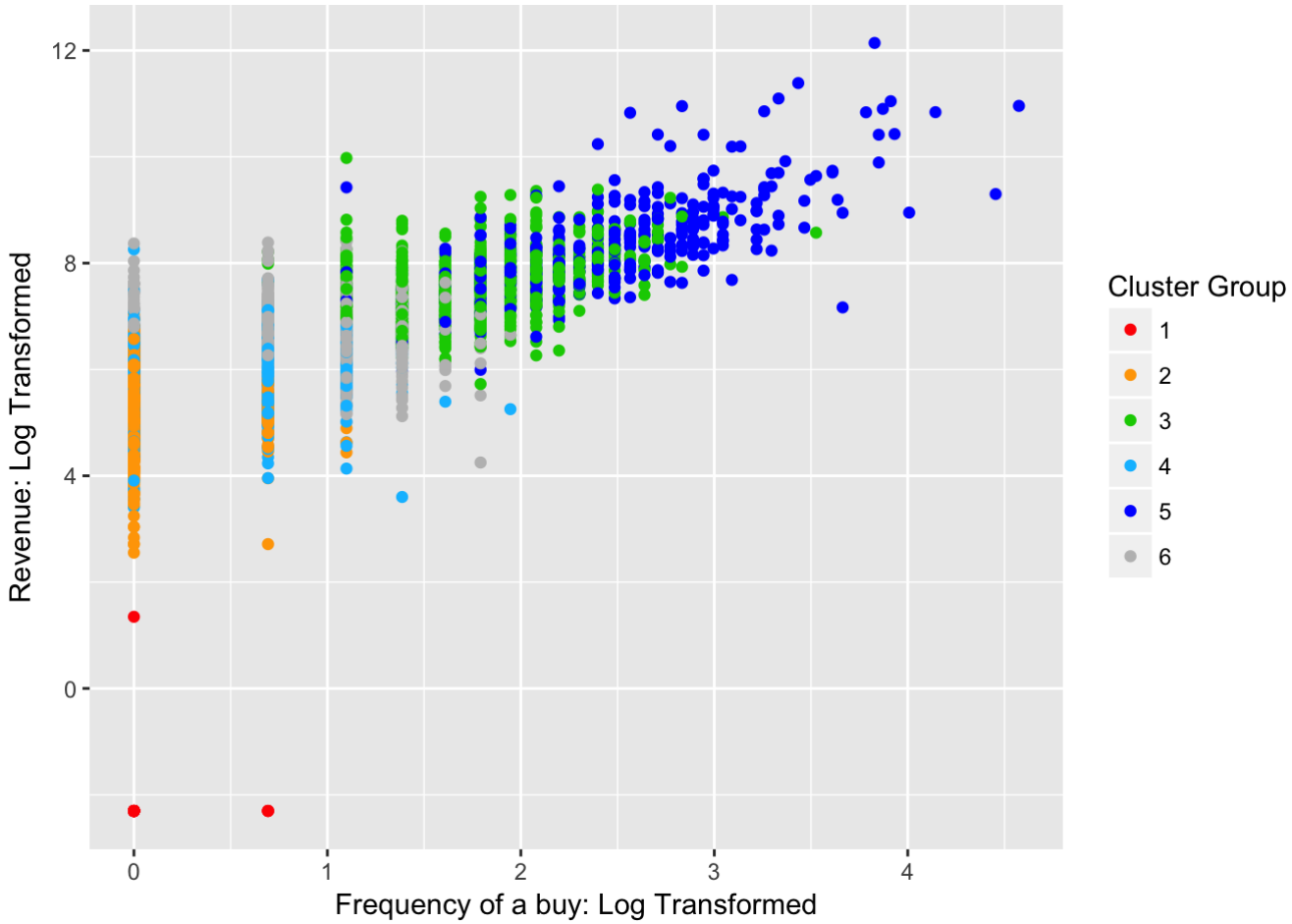

```
##   Cluster Revenue frequency recency
## 1       1 2861.55         8       7
## 2       2  910.93         3      38
## 3       3  253.05         1     152
```



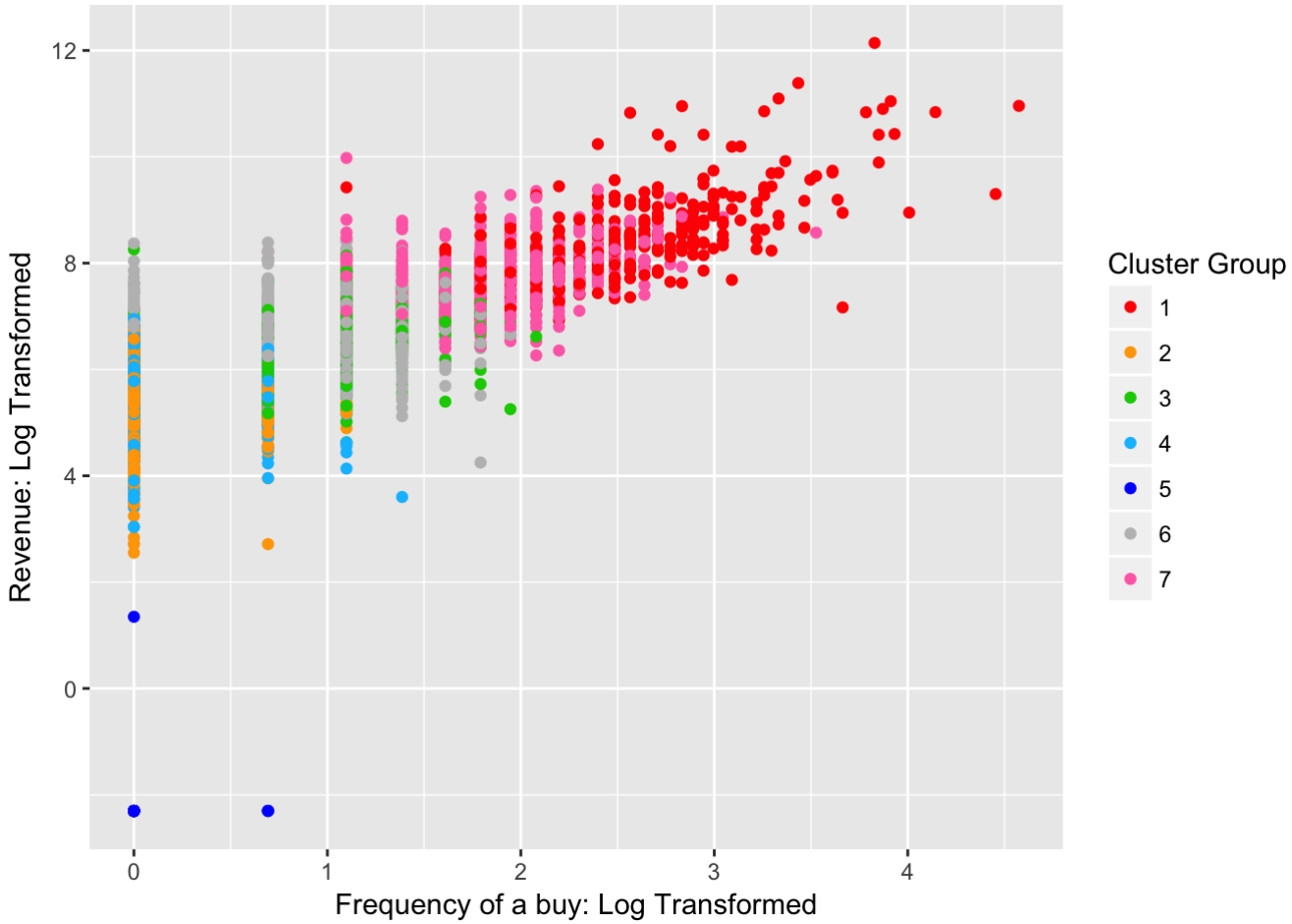

```
##   Cluster Revenue frequency recency
## 1       1 1142.30         4      57
## 2       2  382.80         2      17
## 3       3 2888.75         9       7
## 4       4  255.90         1     183
```



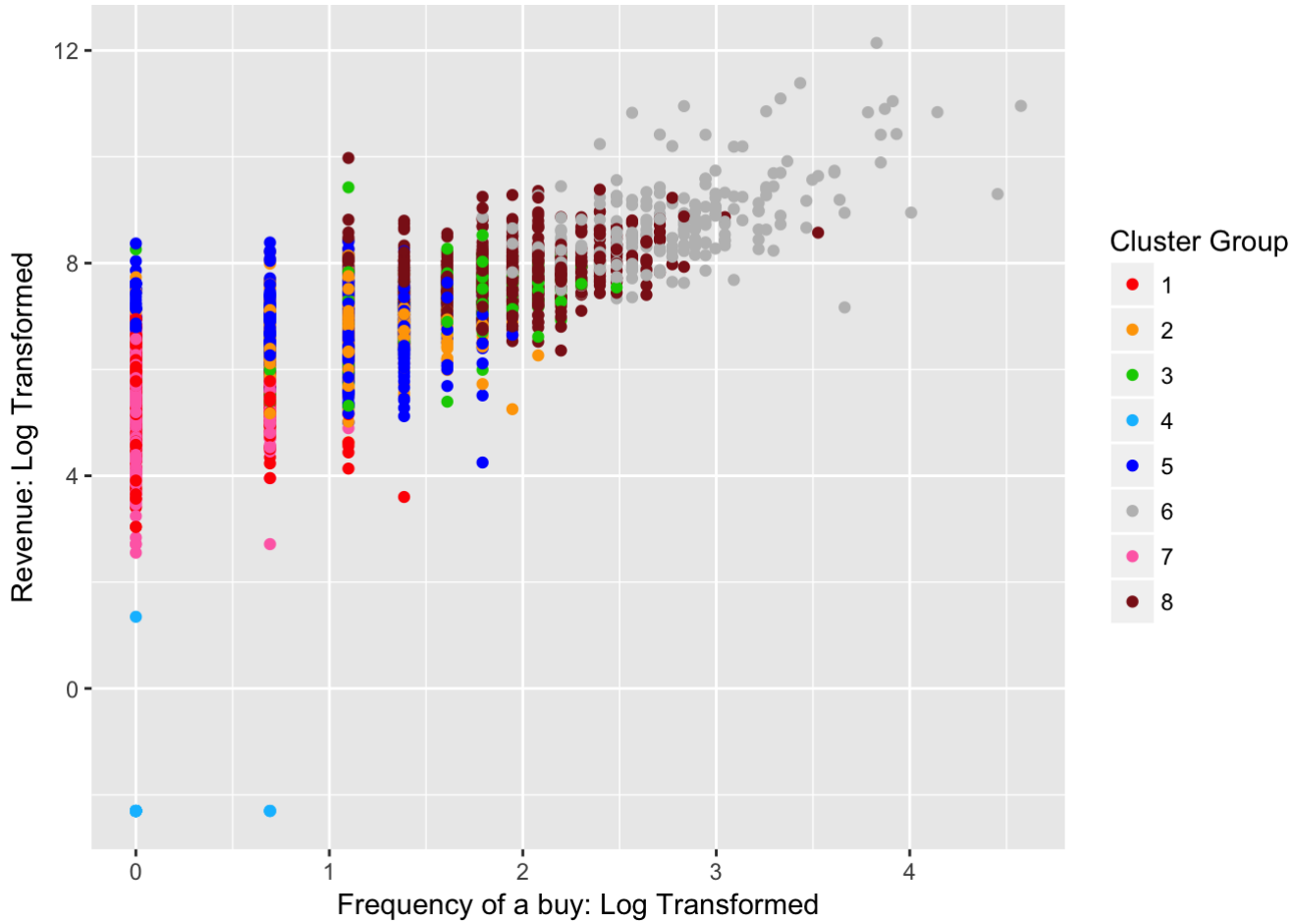

```
##   Cluster Revenue frequency recency
## 1       1 1186.97         4      51
## 2       2  277.80         1     175
## 3       3 2974.65         9       7
## 4       4  384.52         2      17
## 5       5    0.00         1     136
```



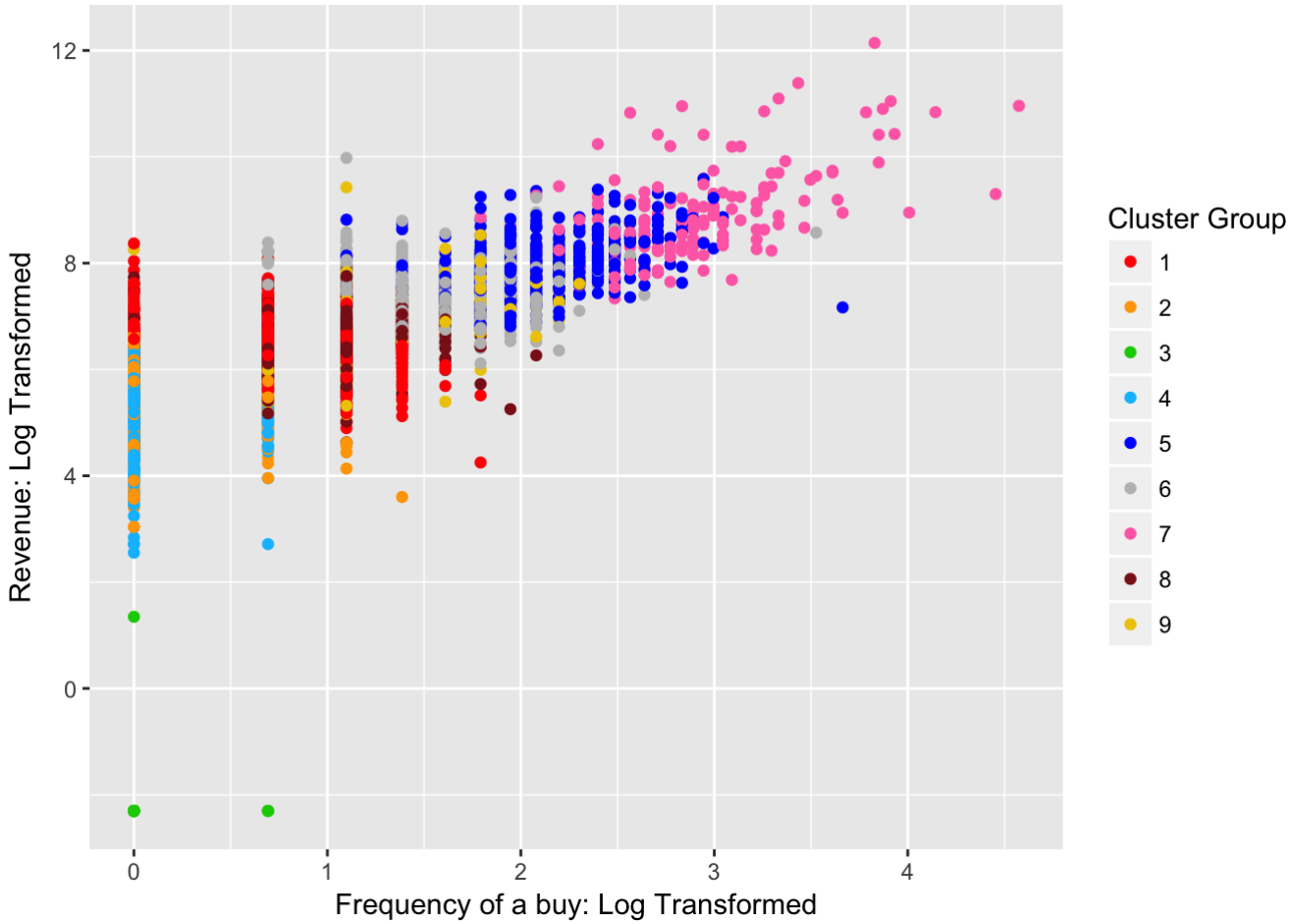

```
##   Cluster Revenue frequency recency
## 1       1    0.00         1     136
## 2       2  226.37         1     189
## 3       3 1789.55         5      24
## 4       4  379.65         2      17
## 5       5 3605.08        11       3
## 6       6  716.00         2      92
```



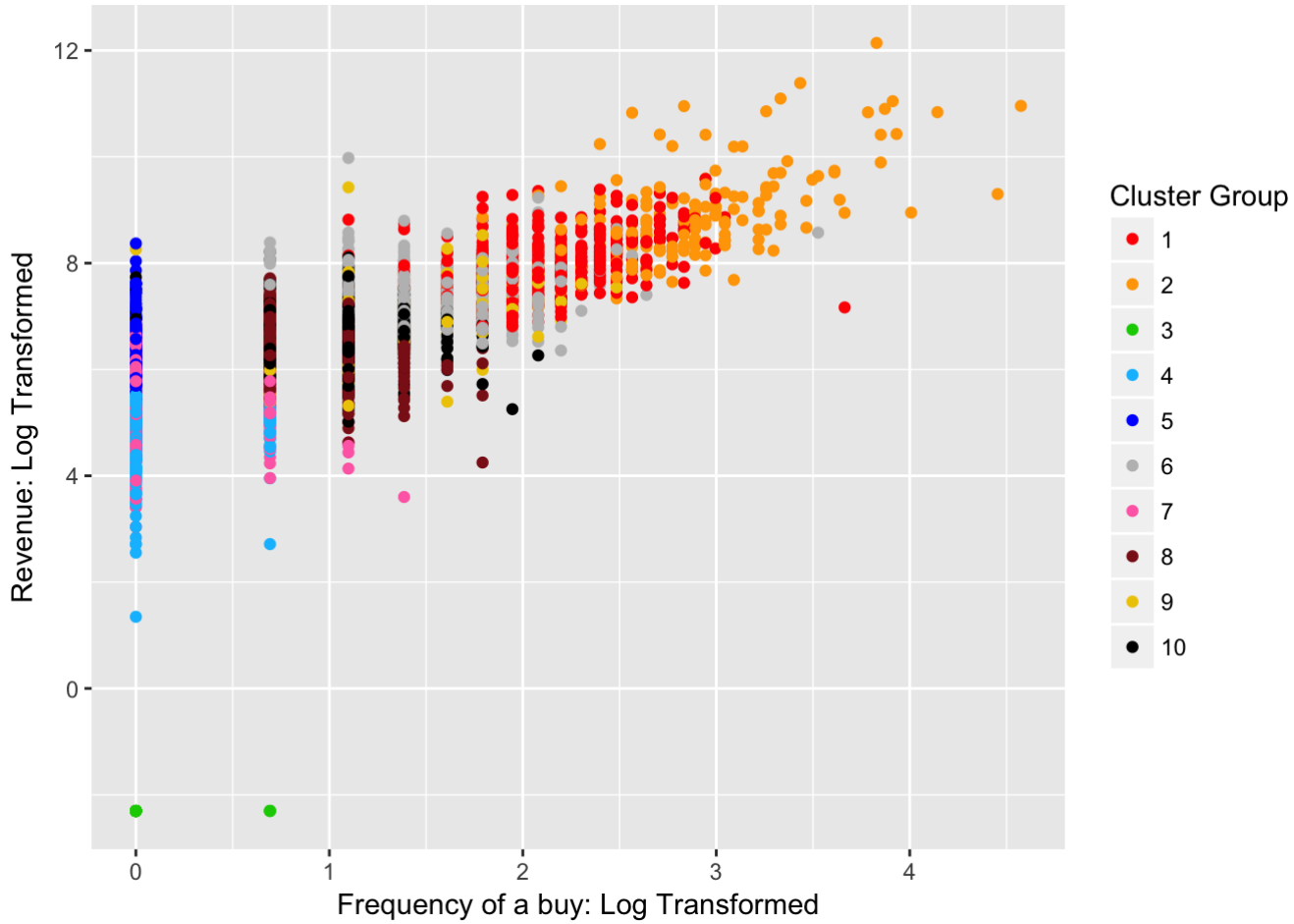

```
##   Cluster Revenue frequency recency
## 1       1 4109.97        12       4
## 2       2  224.43         1     239
## 3       3  746.88         3       8
## 4       4  272.14         1      39
## 5       5    0.00         1     136
## 6       6  752.60         3      84
## 7       7 2106.84         6      28
```




```
##   Cluster Revenue frequency recency
## 1       1  260.17         1      42
## 2       2  780.50         3      19
## 3       3 1161.27         4       2
## 4       4    0.00         1     136
## 5       5  721.23         3     106
## 6       6 5107.38        14       5
## 7       7  215.74         1     241
## 8       8 2286.89         6      29
```




```
##   Cluster Revenue frequency recency
## 1       1  628.38         2     126
## 2       2  246.81         1      39
## 3       3    0.00         1     136
## 4       4  203.48         1     244
## 5       5 2618.23         7      15
## 6       6 1659.75         5      64
## 7       7 6284.07        17       2
## 8       8  732.16         3      20
## 9       9 1044.38         4       2
```




```
##    Cluster Revenue frequency recency
## 1        1 2620.40         8      15
## 2        2 6515.32        17       2
## 3        3    0.00         1     132
## 4        4  142.40         1     241
## 5        5  391.87         1     175
## 6        6 1738.59         5      64
## 7        7  222.67         1      31
## 8        8  571.83         2     126
## 9        9 1098.48         4       2
## 10      10  757.05         3      20
```

Results:

As I saw the graphs and centers of the clusters, I observed that cluster 2 gives me a group of high value customers who generate a revenue of $1776.81 and made the most recent purchase around 17 days back. They have bought atleast 5 times in the past. The low value customers have only generated a revenue of $327.27 in the past.

I found the 5-cluster solution to be most optimum. It gives me a range of high, medium and low value customers. This gives me a broader insight into what the typical behaviour of the customers looks like. The most recent customers buy the products atleast twice and they tend to make the second purchase within the same month or two months.

This helps me find the most high value customers over a range of customers.