The University of Tennessee Knoxville
Haslam College of Business

BZAN 542 - FALL 2021

DATA MINING FINAL PROJECT REPORT

Customer Segmentation
[Clustering]
&
Association Rule Mining
on Retail Data

INSTRUCTOR: CHARLES LIU

TA: FAHIMEH IRENE RAHMANNIYAY

GROUP:

YANYU CHEN

THOA PHAM [KATE]

LAUREN OLIVIA BEAVERS

VENKAT VARUN GUNDAPUNEEDI

TABLE OF CONTENTS

BACKGROUND

The online retailer under consideration in this article is a UK-based and registered non-store business with some 80 members of staff. The company was established in 1981 mainly selling unique all-occasion gifts. For years in the past, the merchant relied heavily on direct mailing catalogues, and orders were taken over phone calls. It was only in 2009, that the company launched its own web site and shifted completely to the Web. Since then, the company has maintained a steady and healthy number of customers from all parts of the United Kingdom and Europe and has accumulated a huge amount of data about many customers. The company also uses Amazon.co.uk to market and sell its products.

DATA

Source : UCI Machine Learning Repository
URL : https://archive.ics.uci.edu/ml/datasets/Online+Retail+II
Contact:
Dr. Daqing Chen,
Course Director: MSc Data Science,
School of Engineering, London South Bank University,
London SE1 0AA, UK.
chend@lsbu.ac.uk

Data Set Information:

This Online Retail II data set contains all the transactions occurring for a UK-based and registered, non-store online retail in 2011.

- The company mainly sells unique all-occasion giftware
- Many customers of the company are wholesalers

ATTRIBUTE	DESCRIPTION
Invoice	Invoice number
StockCode	Product (item) code
Description	Product (item) name
Quantity	The quantities of each product (item) per transaction
InvoiceDate	Invoice date and time
Price	Unit price
Customer.ID	Customer number
Country	Country name

GOALS

1. Customer Segmentation [Clustering]:

- RFM, Recency, Frequency, and Monetary Value, is an often-used way to measure the value of a customer.
- We introduce a Customer Segmentation based on the RFM data computed by the K-means clustering algorithm and compare it to a traditional fixed Customer Segmentation defined by PUTLER also based on RFM features.

2. Association Rule Mining:

- We use this to uncover the product purchase behaviour of customers by extracting the association rules of the products and by customer segments.
- These will be helpful to understand purchase decisions made by the customers and the company can use the association rules in their future marketing strategies.

RESULTS

1. Customer Segmentation [Clustering]:

We believe that going with k-means cluster will be practical for a marketing team at an average scale.

The insights and clarity we get from PUTLER segmentation is great. But it would be great if the customer segments are actually feasible enough for a marketing plan.

We use Putler segments to establish a perspective on a whole and an intention to try the most traditional RFM segmentation of what's happening

The below plots are expected to give a high-level perspective on what the results mean…

Cluster 1: Champion

Low recency & high frequency, monetary

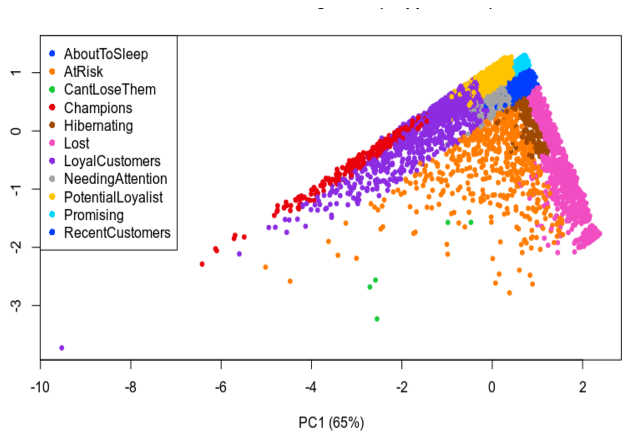
Cluster 2: Customer needing attention

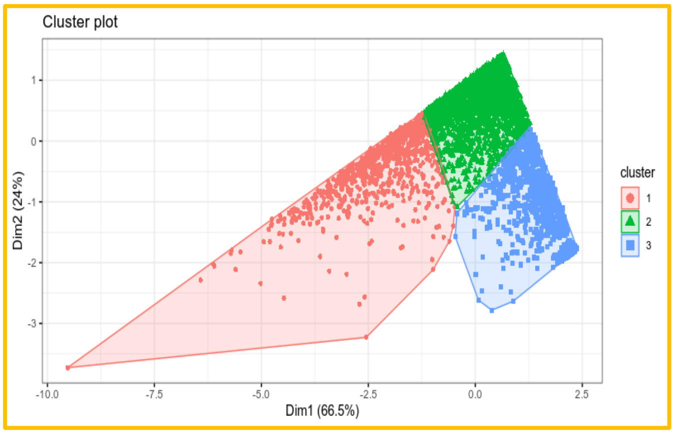
Medium recency, frequency, monetary

Cluster 3: At risk

High recency & low frequency, monetary

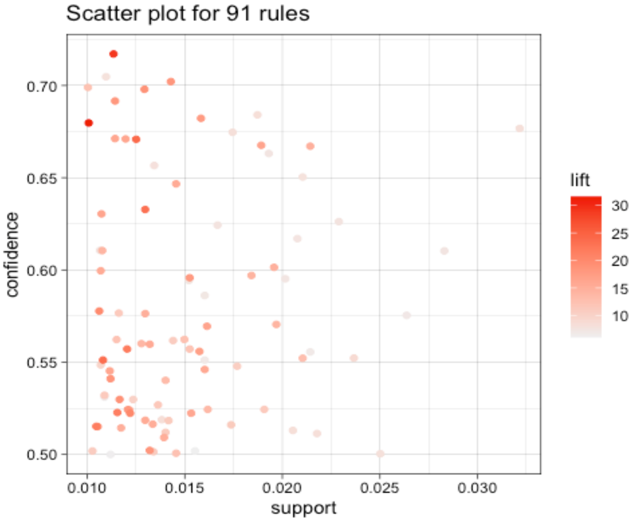
Highlighted one is the k-means Cluster result post PCA…





2. Association Rule Mining:

Creating association rules on the full dataset provided us with a broad understanding of types of products that were purchased together. In the results, the item sets were not unexpected; when creating rules using the product descriptions, we often found that most items in the consequents and antecedents were the same type of product that only differed in colour or pattern. These similarities occurred in both the association rules for all customers (retail and wholesale) and the association rules for only retail customers.



We further analysed association rules on each customer cluster (segment) to see if we can find any different patterns when narrowing the population of customers. As expected, there are some interesting findings in each customer segment.

Meanwhile, Needing Attention Customer usually buy kitchenware's, in door playsets for toddlers and Christmas decorations together. The gift receiver of this customers segment might be young parents with small children. Finally, At-risk/1-time customers are more interested in buying kitchen utensils, cutleries, specialized kitchen equipment's, house decorations, and indoor playsets for young children together.

Conclusion

1. Customer Segmentation [Clustering]:

Our generated customer segments end up with three [k-means and hierarchal] compared to traditional eleven for the PUTLER approach. The analysis, though, shows that the two sets of customer segments split the customers in very different ways, and the segments of the two approaches do not look like each other. It is impossible to say which set of the segment sets that are best because it depends on the intended use.

2. Association Rule Mining:

The Clustering and Association Rules results can be useful for the company to improve their profit by having marketing strategy customized for each segment based on AR results. For 1-time purchase, we can have customized coupons based on associated products or most frequently bought products, which can help transform them into repeat customers. Regarding at-risk customers, the company can send winning back newsletters based on associated products or most frequently bought products of this segment.

Future Work

It would be interesting to see based on the results we have in this analysis to see things such as cancellation patterns and build a dynamic model that can predict the customer segment for new ones using KNN/Random Forest. This would allow a company to build pipeline that does all this automatically.

Note: This is a gift store. Sometimes you might not see proper clustering patterns. So, these analysis gets better when you have more data. The thresholds you set for dividing RFM measures are the key and must be discussed thoroughly with business.

These thresholds sometime effect the customers right at the cluster borders. Instead of sending confusing promotions, approach can be product-based recommendation system rather than segment-based recommendations.

EXPLORATORY DATA ANALYSIS

Data engineering is part of the big data ecosystem and is closely linked to data science and machine learning. This work always tends to go in the background and do not get the same level of attention, but they are critical to the process. They vary depending on an organization's level of data maturity and staffing levels; however, there are some tasks, such as the extracting, loading, and transforming of data, that are foundational.

At the lowest level, data engineering involves the movement of data from one system or format to another system or format. Using more common terms, data engineers query data from a source (extract), they perform some modifications to the data (transform), and then they put that data in a location where users can access it and know that it is production quality (load).

The terms extract, transform, and load will be used a lot throughout this book and will often be abbreviated to ETL. This definition for data engineering is broad and simplistic.

```
data = read.csv("online_retail2.csv")
data$InvoiceDate = as.Date(data$InvoiceDate,'%m/%d/%Y')
min(data$InvoiceDate); max(data$InvoiceDate)
```

```
## [1] "2010-12-01"
## [1] "2011-12-09"
```

```
head(data)
```

```
##   Invoice StockCode                         Description Quantity InvoiceDate
## 1  536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6  2010-12-01
## 2  536365     71053                 WHITE METAL LANTERN        6  2010-12-01
## 3  536365    84406B      CREAM CUPID HEARTS COAT HANGER        8  2010-12-01
## 4  536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6  2010-12-01
## 5  536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6  2010-12-01
## 6  536365     22752        SET 7 BABUSHKA NESTING BOXES        2  2010-12-01
##   Price Customer.ID        Country
## 1  2.55       17850 United Kingdom
## 2  3.39       17850 United Kingdom
## 3  2.75       17850 United Kingdom
## 4  3.39       17850 United Kingdom
## 5  3.39       17850 United Kingdom
## 6  7.65       17850 United Kingdom
```

```
glimpse(data)
```

```
## Rows: 541,910

## Columns: 8

## $ Invoice     <chr> "536365", "536365", "536365", "536365", "536365",
"536365", "536365", "536366",…
```

```
## $ StockCode   <chr> "85123A", "71053", "84406B", "84029G", "84029E", "22752",
"21730", "22633", "22…

## $ Description <chr> "WHITE HANGING HEART T-LIGHT HOLDER", "WHITE METAL
LANTERN", "CREAM CUPID HEART…

## $ Quantity    <int> 6, 6, 8, 6, 6, 2, 6, 6, 6, 6, 3, 3, 3, 32, 6, 6, 8, 6, 6,
3, 2, 3, 3, 4, 4, 3, …

## $ InvoiceDate <date> 2010-12-01, 2010-12-01, 2010-12-01, 2010-12-01, 2010-12-
01, 2010-12-01, 2010-1…

## $ Price       <dbl> 2.55, 3.39, 2.75, 3.39, 3.39, 7.65, 4.25, 1.85, 1.85,
4.25, 4.95, 4.95, 4.95, 1…

## $ Customer.ID <int> 17850, 17850, 17850, 17850, 17850, 17850, 17850, 17850,
17850, 13047, 13047, 13…

## $ Country     <chr> "United Kingdom", "United Kingdom", "United Kingdom",
"United Kingdom", "United…
```

DATA PRE-PROCESSING

Data pre-processing is a broad area and consists of several different strategies and techniques that are interrelated in complex ways. Roughly speaking, pre-processing methods fall into two categories:

- Selecting data objects and attributes for the analysis
- Creating/changing the attributes.

In both cases, the goal is to improve the data mining analysis concerning time, cost, and quality. As mentioned in the previous section, this dataset is detailed and not suitable for high-level potential information mining, pre-processing takes a lot of time and attention in this project.

```
# NA/Null Value check
apply(is.na(data),2,sum)
```

```
##     Invoice   StockCode Description    Quantity InvoiceDate       Price
##           0           0           0           0           0           0
## Customer.ID     Country
##      135080           0
```

This data is not aggregated by invoices yet. All over – There are more than 100K rows among 500K that don’t have Customer.ID populated…

Based on intuition and reasonable assumptions lets see what those records summarize as…

```
length(unique(data$Invoice))
```

```
## [1] 25900
```

All those transaction records summed up to 25900 Invoices

Grouping Invoices

This below data represents the centralized grouping of invoice data to represent a customer visit.

We also summed up all the transformation to this part to avoid repetition…

```
invoice =
  data %>%
  group_by(Invoice, Customer.ID, Country) %>%
  summarise(InvoiceDate = max(InvoiceDate, na.rm = T),
            Items = n_distinct(StockCode),
            Quantity = sum(Quantity, na.rm = T),
            InvoiceAmount = sum(Quantity * Price, na.rm = T)) %>%
  mutate(Inv_Status = ifelse(substr(Invoice,1,1) %in% letters |
                               substr(Invoice,1,1) %in% LETTERS,
                             "Cancelled",
                             "Approved"))
```

```
table(invoice$Inv_Status)
```

```
##  Approved Cancelled
##     22061      3839
```

High level overview of these invoices will allow us to know what we will be working…

Coming back to what these records with no Customer.ID can be categorized as?

Because there is no way to use these for customer segmentation.

We found that there are around 3710 records among 25.9K don’t have customer id and have noticed that some of them have bigger baskets than the other ones...

Being experienced on looking at this kind of data, automated system doesn’t miss mandatory information such as customer.id. These records have entered manually and have a lot of potential for errors. As the emphasis is to work around the retail customers.

Let’s categorize the records with missing customer.ID as Wholesale Customers.

Let’s add a variable Group which segregates the retail and customer groups.

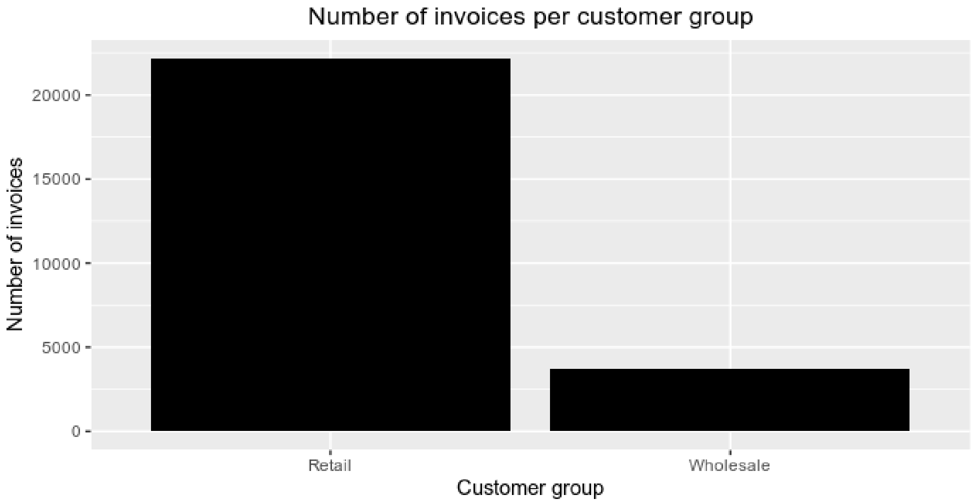
This is an assumption**

In real-time, we discuss this with the business and suggest them to have a process in place to capture data for wholesale customers. We see that this data is being pushed manually. Hence there is no customer.id associated. Doing this in next years will help us substantiate the analysis for wholesale group as well.

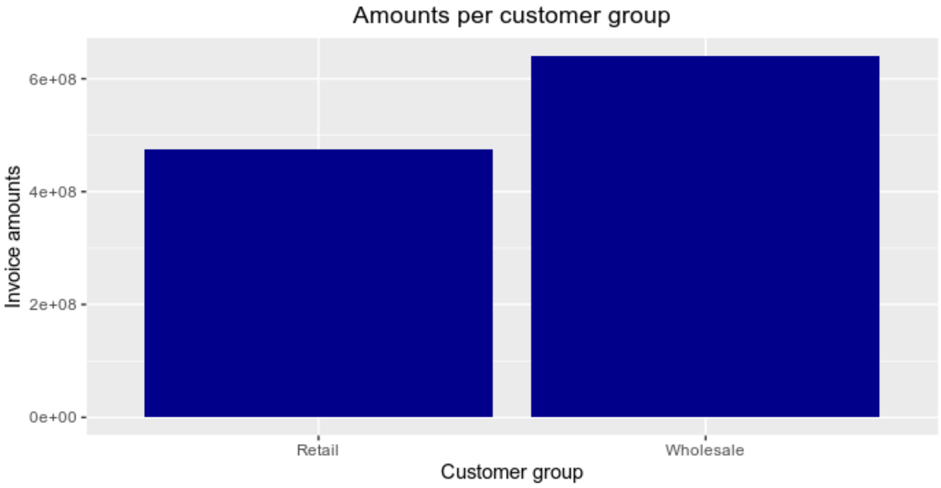
```
invoice =
  invoice %>%
  mutate(CustGroup = ifelse(is.na(Customer.ID), "Wholesale", "Retail"))
```

Let’s look at some stats that will push the company from doing any manual entries and why Wholesale data is very important to the analysis.

```
stats =
  invoice %>%
  group_by(CustGroup) %>%
  summarise(n_invoices = n(),
            invoiceAmount = sum(InvoiceAmount, na.rm = TRUE))

ggplot(stats) +
  geom_bar(aes(x = CustGroup, y = n_invoices), stat = "identity", fill = "black")
+
  labs(x = "Customer group",
       y = "Number of invoices",
       title = "Number of invoices per customer group") +
  theme(plot.title = element_text(hjust = 0.5))
```



Though the number of invoices is less in comparison to what they have with retail customers.



Look at this plot that signifies the importance of wholesale customers to the company. They bring in more revenue and it would be interesting to see the patterns if the information exists.

GROUPING ITEMS

This is done to understand what kind of products the store is selling. Just for all over idea to use them

```
items =
  data %>%
  group_by(StockCode,Description) %>%
  summarise(Sold = sum(Quantity, na.rm = T),
            Sales = sum(Quantity * Price, na.rm = T)) %>%
  arrange(desc(Sales))
```

WORKING WITH COUNTRIES

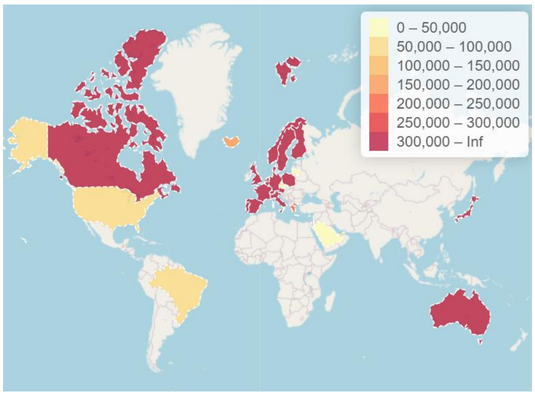
Let's see If attribute country is important in this dataset. On a global level, you would say no as 98% of the records come from the United Kingdom.

But it would still be interesting to see if there is any country that shows promise for the marketing team to look at. Obviously, there is an uptick of sales in November and December this being a gift store and those months being holiday seasons. Marketing teams will look at the weight of ROI on the investments they make.

```
# Top 5 countries in Sales

invoice %>%
  group_by(Country) %>%
  summarise(Amount = sum(InvoiceAmount, na.rm = TRUE)) %>%
  filter(rank(desc(Amount)) <= 5) %>%
  arrange(desc(Amount))
```

```
## # A tibble: 5 × 2
##   Country            Amount
##   <chr>               <dbl>
## 1 United Kingdom 992035911.
## 2 Netherlands     30597968.
## 3 EIRE            17365776.
## 4 Australia       15609438.
## 5 France          12747871.
```

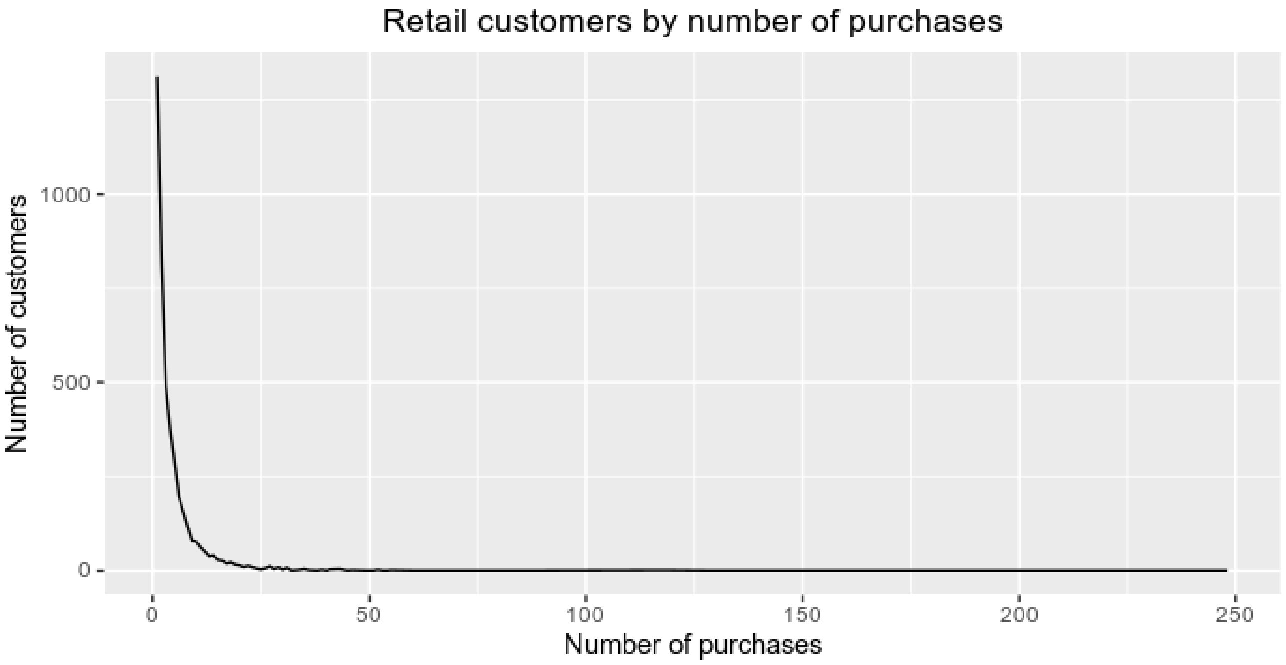
We tried to use leaflet to generate some interactive visualizations by country…



We are almost at that point where we step into modelling. Before that lets answer one more question…

How many times in a year do retail customers make purchases? How many of them are frequent repeat customers?

```
invoice %>%
  filter(CustGroup == "Retail") %>%
  group_by(Customer.ID) %>%
  summarise(NumInvoices = n()) %>%
  ungroup() %>%
  group_by(NumInvoices) %>%
  summarise(NumCustomers = n()) %>%
  arrange(desc(NumInvoices)) %>%
  collect() %>%
  ggplot() +
  geom_freqpoly(aes(x = NumInvoices, y = NumCustomers), stat = "identity", colour
= "black") +
  labs(x = "Number of purchases",
       y = "Number of customers",
       title = "Retail customers by number of purchases") +
  theme(plot.title = element_text(hjust = 0.5))
```



This is done to see how our model plots look like… Due to the lack of huge variances in the buying pattern. There is a good chance that you will not see clear clusters when we transform the data into variables that signify how valuable a particular customer is to the company.

Generally, pre-processing of this project consists of the following steps:

- Dealing with missing values
- Attribute reduction
- Attribute transformation
- Aggregation and dealing with outliers.

However, pre-processing is always purposeful. In this project, the purpose of pre-processing is to build an RFM model suitable for customer clustering.

Therefore, before diving into pre-processing steps, we try to provide a clear idea for the RFM model first.

RFM Model:

The RFM model was proposed by Hughes (2000). This model is popular in customer value analysis and has been widely used in measuring customer lifetime value and in customer segmentation and behaviour analysis. The RFM model also has been used in several cases, especially in choosing clustering indexes. The RFM segmentation model is a model that differentiates important customers according to three variables:

R represents "Recency", defined as the interval between the most recent transaction and the present.

F represents "Frequency", defined as how many times has a customer ever conducted commercial behaviours.

M represents "Monetary", defined as the total amount of consumption of a certain customer.

The RFM method is very effective at customer segmentation. Each customer is positioned in a three-dimensional space, corresponding to a coordinate of R, F, and M. With these RFMs sorted in descending order, the groups of customers are classified proportionately.

CLUSTERING PREPARATION

We work on records that have customer.ID which translates to retail customers in this project.

```
retail = data[!is.na(data_2010$Customer.ID), ]
```

Removing the invoices with negative quantity and price which are return and cancelled transactions…

```
retail = retail[retail$Quantity >= 0,]
retail = retail[retail$Price > 0,]

# The data has large number of free items in transactions... Ignoring them for ri
gid monetary measure
```

This is to see the quality of sales as most of the transactions are small ones.

We also observed that there is 1 cancellation every 6 transactions as a whole. One of the future works can be to look into the driving factor that's causing this by looking at product combination, region, etc...

At this point, we didn't have enough time to provide insights on the causes of negative factors in the dataset.

All the transformations that happen from here are aggregated to customer level for segmentation.

BUILDING RFM DATASET

```
# RFM Measures

RDate = max(retail$InvoiceDate)
CustomerRFM = retail %>%
  group_by(Customer.ID) %>%
  summarize(Recency = as.numeric(RDate - max(InvoiceDate)),
            Frequency = n_distinct(Invoice),
            Monetary = sum(Quantity * Price))

head(CustomerRFM)
```

```
## # A tibble: 6 × 4
##   Customer.ID Recency Frequency Monetary
##         <int>   <dbl>     <int>    <dbl>
## 1       12346     325         1   77184.
## 2       12347       2         7    4310
## 3       12348      75         4    1797.
## 4       12349      18         1    1758.
## 5       12350     310         1     334.
## 6       12352      36         8    2506.
```

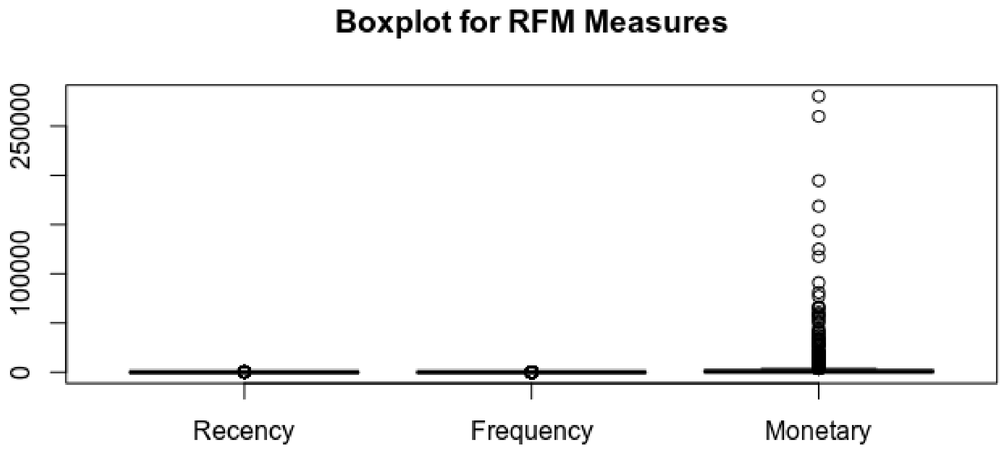
```
# structure of the RFM data
str(CustomerRFM)
```

```
## tibble [4,338 × 4] (S3: tbl_df/tbl/data.frame)
##  $ Customer.ID: int [1:4338] 12346 12347 12348 12349 12350 12352 12353 12354 1
2355 12356 ...
##  $ Recency    : num [1:4338] 325 2 75 18 310 36 204 232 214 22 ...
##  $ Frequency  : int [1:4338] 1 7 4 1 1 8 1 1 1 3 ...
##  $ Monetary   : num [1:4338] 77184 4310 1797 1758 334 ...
```

```
# Summary of the RFM data
summary(CustomerRFM
```

```
##   Customer.ID       Recency         Frequency          Monetary
##  Min.   :12346   Min.   :  0.00   Min.   :  1.000   Min.   :      3.75
##  1st Qu.:13813   1st Qu.: 17.00   1st Qu.:  1.000   1st Qu.:    307.42
##  Median :15300   Median : 50.00   Median :  2.000   Median :    674.48
##  Mean   :15300   Mean   : 92.06   Mean   :  4.272   Mean   :   2054.27
##  3rd Qu.:16779   3rd Qu.:141.75   3rd Qu.:  5.000   3rd Qu.:   1661.74
##  Max.   :18287   Max.   :373.00   Max.   :209.000   Max.   :280206.02
```

```
# Checking for outliers to prevent skewness as we will scale these variables befo
re clustering

boxplot(CustomerRFM[-1], main="Boxplot for RFM Measures")
```



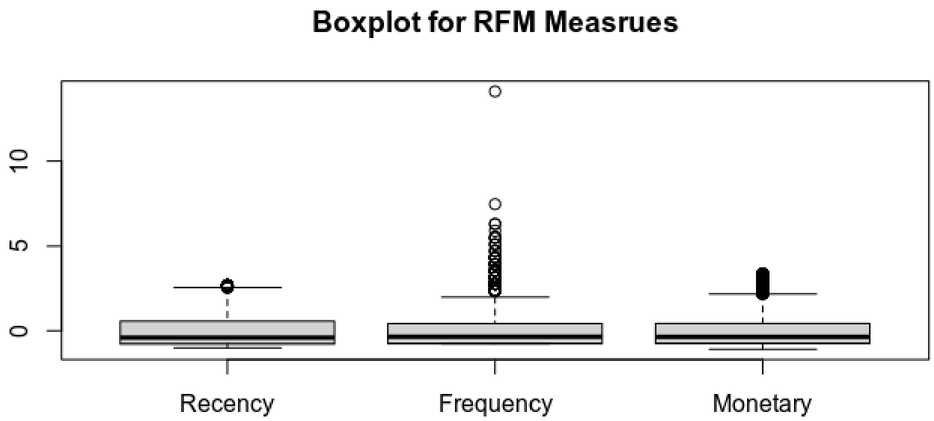
```
# Lets go ahead and remove these outliers for seamless clustering potentially

R_outliers = boxplot(CustomerRFM$Recency, plot=FALSE)$out
RFM = CustomerRFM[-which(CustomerRFM$Recency %in% R_outliers),]

F_outliers = boxplot(CustomerRFM$Frequency, plot=FALSE)$out
RFM = CustomerRFM[-which(CustomerRFM$Frequency %in% F_outliers),]

M_outliers = boxplot(CustomerRFM$Monetary, plot=FALSE)$out
RFM = CustomerRFM[-which(CustomerRFM$Monetary %in% M_outliers),]

head(RFM)
```

```
## # A tibble: 6 × 4
##   Customer.ID Recency Frequency Monetary
##         <int>   <dbl>     <int>    <dbl>
## 1       12348      75         4    1797.
## 2       12349      18         1    1758.
## 3       12350     310         1     334.
## 4       12352      36         8    2506.
## 5       12353     204         1      89
## 6       12354     232         1    1079.
```

```
# Scale the values
RFM$Recency = scale(RFM$Recency)
RFM$Frequency = scale(RFM$Frequency)
RFM$Monetary = scale(RFM$Monetary)
```

```
head(RFM)

boxplot(RFM[-1], main="Boxplot for RFM Measures")
```

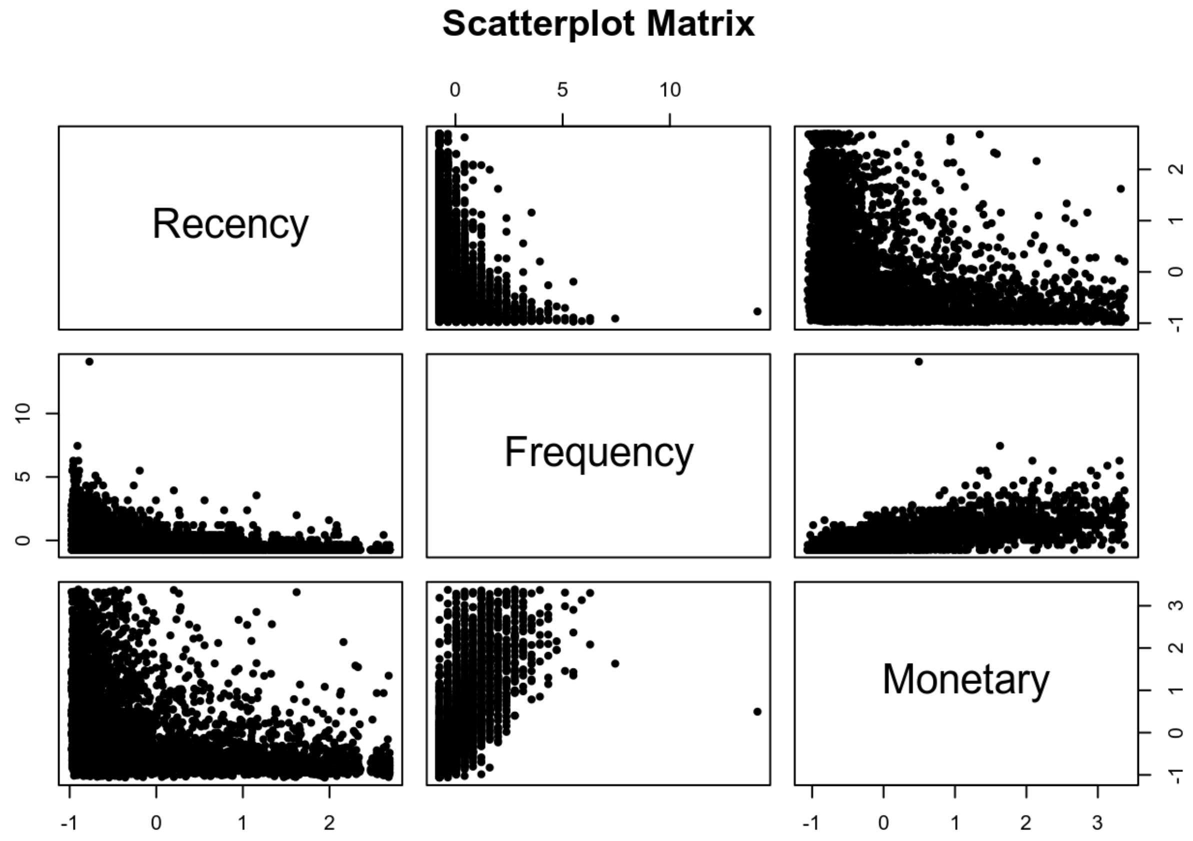
```
## # A tibble: 6 × 4
##   Customer.ID Recency[,1] Frequency[,1] Monetary[,1]
##         <int>       <dbl>         <dbl>        <dbl>
## 1       12348      -0.240         0.426         1.10
## 2       12349      -0.802        -0.745         1.05
## 3       12350       2.07         -0.745        -0.667
## 4       12352      -0.624         1.99          1.95
## 5       12353       1.03         -0.745        -0.963
## 6       12354       1.31         -0.745         0.233
```

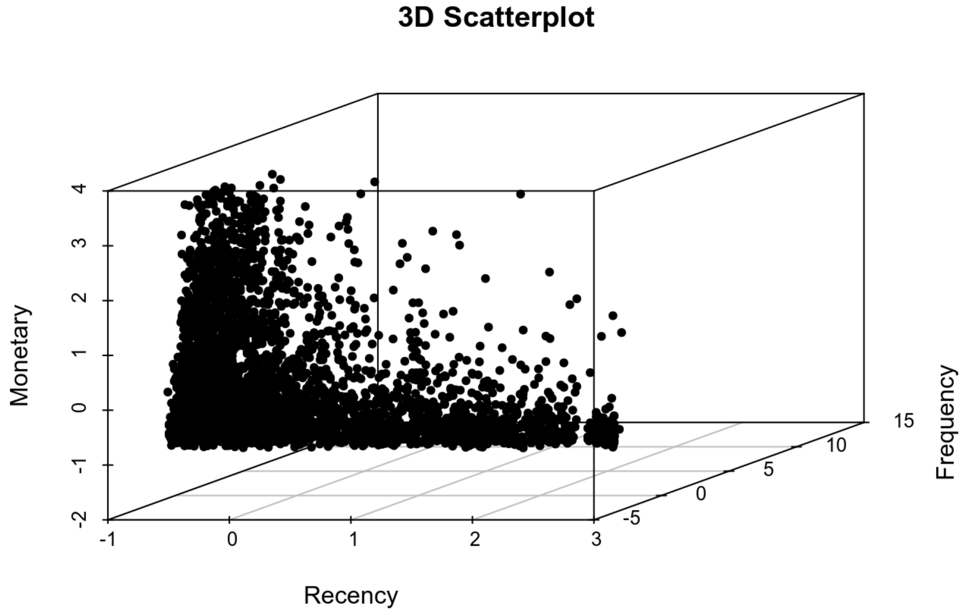


Pre-Clustering Visuals

```
pairs(~ Recency + Frequency + Monetary,
      data = RFM,
      main = "Scatterplot Matrix", pch = 20)
```

```
# 3D Plot of all 3 measures


scatterplot3d(RFM$Recency,
              RFM$Frequency,
              RFM$Monetary,
              pch = 20,
              main="3D Scatterplot",
              xlab = "Recency",
              ylab = "Frequency",
              zlab = "Monetary")
```





Though, these graphs look fancy – They are very hard to read

Principal Component Analysis [PCA]

PCA computes a new 3d coordinate system where each coordinate, in order, covers as much variability as possible. So, dropping the least important coordinate after PCA will usually give a much better result, and never be worse, than just dropping one of the coordinates without PCA.

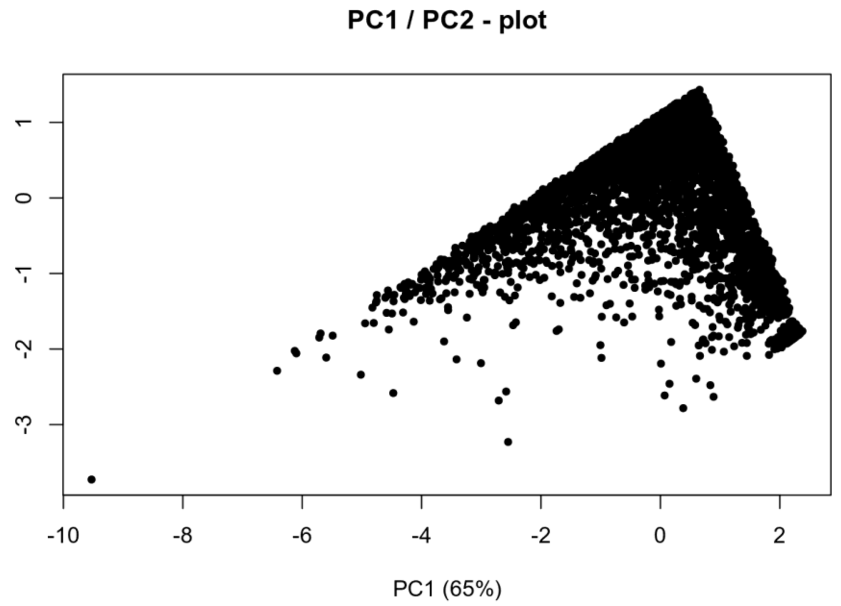
Let's use PCA to map all points from 3d to 2d

```
RFM_PCA = prcomp(RFM[c(2,3,4)])

summary(RFM_PCA)
```

```
## Importance of components:
##                           PC1    PC2     PC3
## Standard deviation     1.4128 0.8489 0.53220
## Proportion of Variance 0.6654 0.2402 0.09441
## Cumulative Proportion  0.6654 0.9056 1.00000
```

2 coordinates together explain close to 90% of the variance of the data, which means that using the first two coordinates will give a reasonable illustration of the customers RFM values

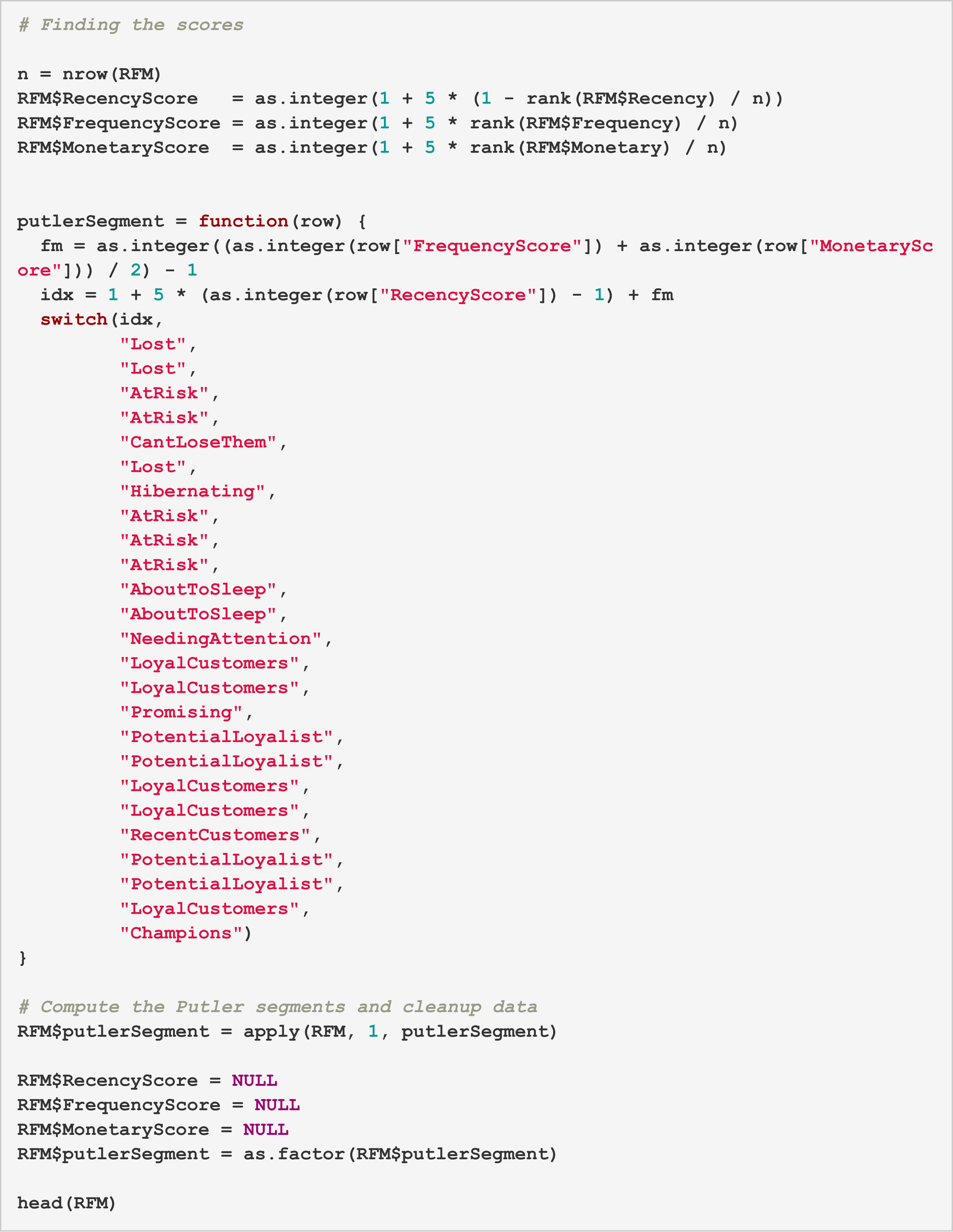
```
plot(RFM_PCA$x[,1],
     RFM_PCA$x[,2],
     xlab="PC1 (65%)",
     ylab = "PC2 (25%)",
     main = "PC1 / PC2 - plot",
     pch = 20, col = "black"
```



Customer Segmentation using the Putler method

Putler method recommends using 11 segments for RFM modelling…

Customer Segment	*Activity*	*Actionable Tip*
Champions	Bought recently, buy often and spend the most!	Reward them. Can be early adopters for new products. Will promote your brand.
Loyal Customers	Spend good money with us often. Responsive to promotions.	Upsell higher value products. Ask for reviews. Engage them.
Potential Loyalist	Recent customers, but spent a good amount and bought more than once.	Offer membership / loyalty program, recommend other products.
Recent Customers	Bought most recently, but not often.	Provide on-boarding support, give them early success, start building relationship.
Promising	Recent shoppers, but haven't spent much.	Create brand awareness, offer free trials
Customers Needing Attention	Above average recency, frequency and monetary values. May not have bought very recently though.	Make limited time offers, recommend based on past purchases. Reactivate them.
About To Sleep	Below average recency, frequency and monetary values. Will lose them if not reactivated.	Share valuable resources, recommend popular products / renewals at discount, reconnect with them.
At Risk	Spent big money and purchased often. But long time ago. Need to bring them back!	Send personalized emails to reconnect, offer renewals, provide helpful resources.
Can't Lose Them	Made biggest purchases, and often. But haven't returned for a long time.	Win them back via renewals or newer products, don't lose them to competition, talk to them.
Hibernating	Last purchase was long back, low spenders and low number of orders.	Offer other relevant products and special discounts. Recreate brand value.
Lost	Lowest recency, frequency and monetary scores.	Revive interest with reach out campaign, ignore otherwise



```
# Finding the scores

n = nrow(RFM)
RFM$RecencyScore   = as.integer(1 + 5 * (1 - rank(RFM$Recency) / n))
RFM$FrequencyScore = as.integer(1 + 5 * rank(RFM$Frequency) / n)
RFM$MonetaryScore  = as.integer(1 + 5 * rank(RFM$Monetary) / n)


putlerSegment = function(row) {
  fm = as.integer((as.integer(row["FrequencyScore"]) + as.integer(row["MonetarySc
ore"])) / 2) - 1
  idx = 1 + 5 * (as.integer(row["RecencyScore"]) - 1) + fm
  switch(idx,
         "Lost",
         "Lost",
         "AtRisk",
         "AtRisk",
         "CantLoseThem",
         "Lost",
         "Hibernating",
         "AtRisk",
         "AtRisk",
         "AtRisk",
         "AboutToSleep",
         "AboutToSleep",
         "NeedingAttention",
         "LoyalCustomers",
         "LoyalCustomers",
         "Promising",
         "PotentialLoyalist",
         "PotentialLoyalist",
         "LoyalCustomers",
         "LoyalCustomers",
         "RecentCustomers",
         "PotentialLoyalist",
         "PotentialLoyalist",
         "LoyalCustomers",
         "Champions")
}

# Compute the Putler segments and cleanup data
RFM$putlerSegment = apply(RFM, 1, putlerSegment)

RFM$RecencyScore = NULL
RFM$FrequencyScore = NULL
RFM$MonetaryScore = NULL
RFM$putlerSegment = as.factor(RFM$putlerSegment)

head(RFM)
```

```
## # A tibble: 6 × 5
##   Customer.ID Recency[,1] Frequency[,1] Monetary[,1] putlerSegment
##         <int>       <dbl>         <dbl>        <dbl> <fct>
## 1       12348      -0.240         0.426        1.10  LoyalCustomers
## 2       12349      -0.802        -0.745        1.05  PotentialLoyalist
## 3       12350       2.07         -0.745       -0.667 Lost
## 4       12352      -0.624         1.99         1.95  LoyalCustomers
## 5       12353       1.03         -0.745       -0.963 Lost
## 6       12354       1.31         -0.745        0.233 Lost
```

Pareto's rule says **80% of the results come from 20% of the causes.**

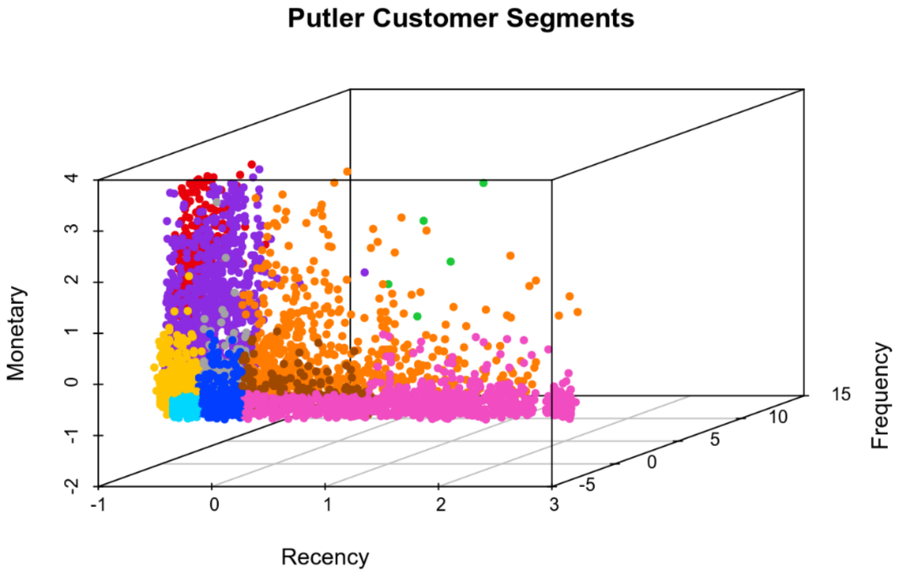
Similarly, **20% customers contribute to 80% of your total revenue.**

People who spent once are more likely to spend again. People who make big ticket purchases are more likely to repeat them.

Pareto Principle is at the core of RFM model. Focusing your efforts on critical segments of customers is likely to give you much higher return on investment!

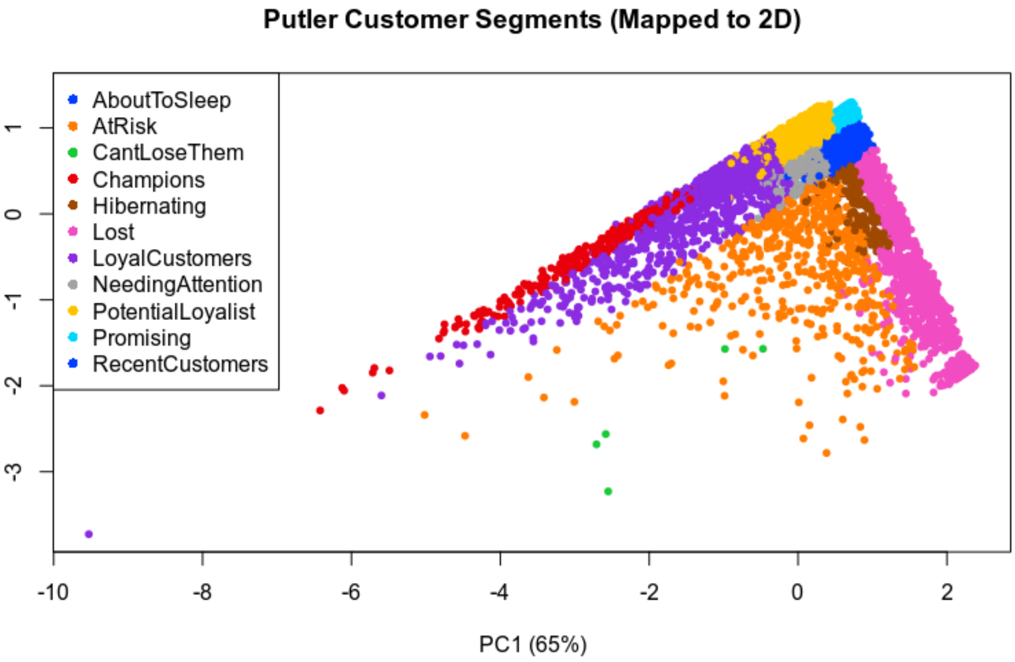
VISUALIZING PUTLER SEGMENTS

```
colorpalette = c("#023eff",
                 "#ff7c00",
                 "#1ac938",
                 "#e8000b",
                 "#9f4800",
                 "#f14cc1",
                 "#8b2be2",
                 "#a3a3a3",
                 "#ffc400",
                 "#00d7ff")

colors = colorpalette[RFM$putlerSegment]
plot3d = scatterplot3d(RFM$Recency,
                       RFM$Frequency,
                       RFM$Monetary,
                       xlab = "Recency",
                       ylab = "Frequency",
                       zlab = "Monetary",
                       main = "Putler Customer Segments",
                       color = colors,
                       pch = 20)
legend(plot3d$xyz.convert(2.9, 10, 40),
       legend = levels(RFM$putlerSegment),
       col = colorpalette,
       pch = 16)
```



Yay! Colour palette works…

```
# Check in 2d from PCA
colors = colorpalette[RFM$putlerSegment]

plot(RFM_PCA$x[,1], RFM_PCA$x[,2],
     xlab="PC1 (65%)", ylab = "PC2 (25%)",
     main = "Putler Customer Segments (Mapped to 2D)",
     pch = 20, col = colors)
legend("topleft", legend = levels(RFM$putlerSegment),
        col = colorpalette, pch = 16)
```



```
# number of customers in each Putler segment is
RFM %>%
  group_by(putlerSegment) %>%
  summarize(n_customers = n()) %>%
  arrange(desc(n_customers))
```

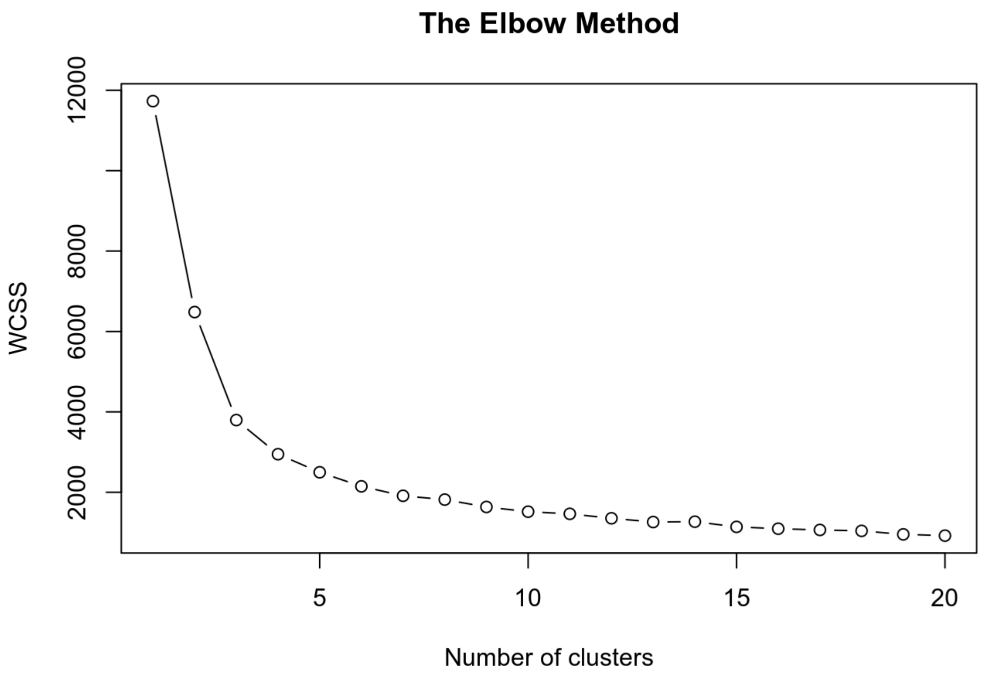
```
## # A tibble: 11 × 2
##    putlerSegment     n_customers
##    <fct>                   <int>
##  1 Lost                      875
##  2 LoyalCustomers            852
##  3 AtRisk                    531
##  4 PotentialLoyalist         510
##  5 AboutToSleep              356
##  6 Champions                 236
##  7 NeedingAttention          175
##  8 Promising                 152
##  9 Hibernating               148
## 10 RecentCustomers            71
## 11 CantLoseThem                5
```

CUSTOMER SEGMENTATION USING THE K-MEANS CLUSTERING METHOD

K means clustering is a data mining technique used to perform cluster analysis. It aims to partition observations in a data set into individual clusters. Each observation belongs to a one single cluster with the nearest mean. All observations in a cluster exhibit almost similar characteristic.

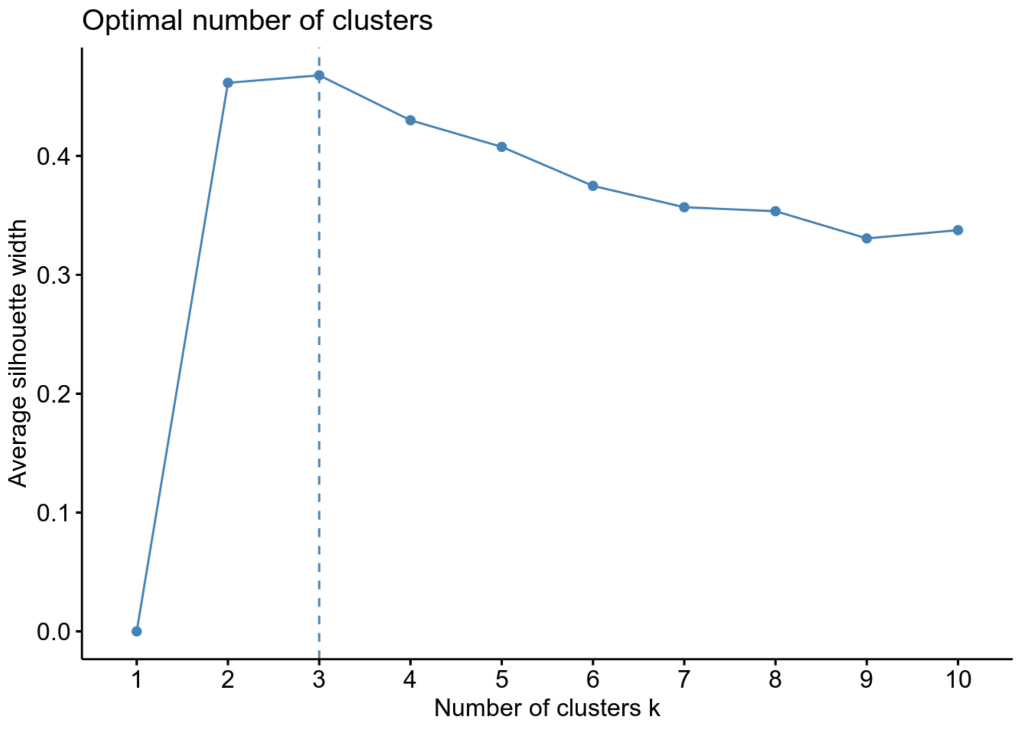
To perform K means clustering scaling is very important as it ensures scales of each variable is same. One important question while performing K-means clustering is to decide on how many clusters would be appropriate for to analyse for the business. In general, Elbow method and Silhouette analysis on data set allows to understand the optimal number of clusters that are present in the data set.

```
set.seed(542)
library(factoextra)

df = select(RFM, Recency, Frequency, Monetary)

# Finding Optimal K
# Method - 1
wcss = vector()

for (i in 1:20) wcss[i] = sum(kmeans(df, i)$withinss)
plot(1:20,
     wcss,
     type = 'b', # for lines and points
     main = paste('The Elbow Method'),
     xlab = 'Number of clusters',
     ylab = 'WCSS')
```



Optimal K = 3

```r
# Method - 2
# Silhouette Score
fviz_nbclust(df, kmeans, method = "silhouette")
```



Seems like 3 is the number…

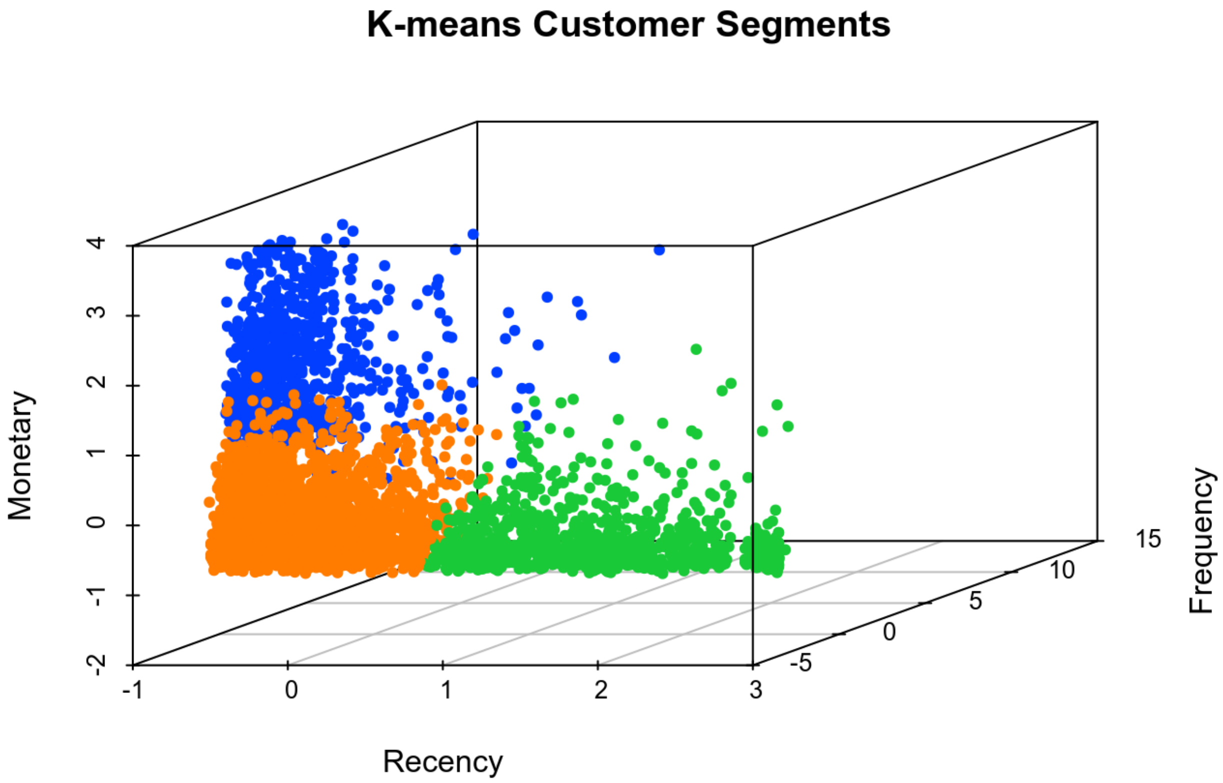
```
bestKmeans_k = 3

# Using the best clusters on the RFM data
k_result = kmeans(df, centers = 3, iter.max = 15, nstart = 25)

# Add the found clusters to the RFM
RFM$kmeansSegment = as.factor(k_result$cluster)

head(RFM)
```

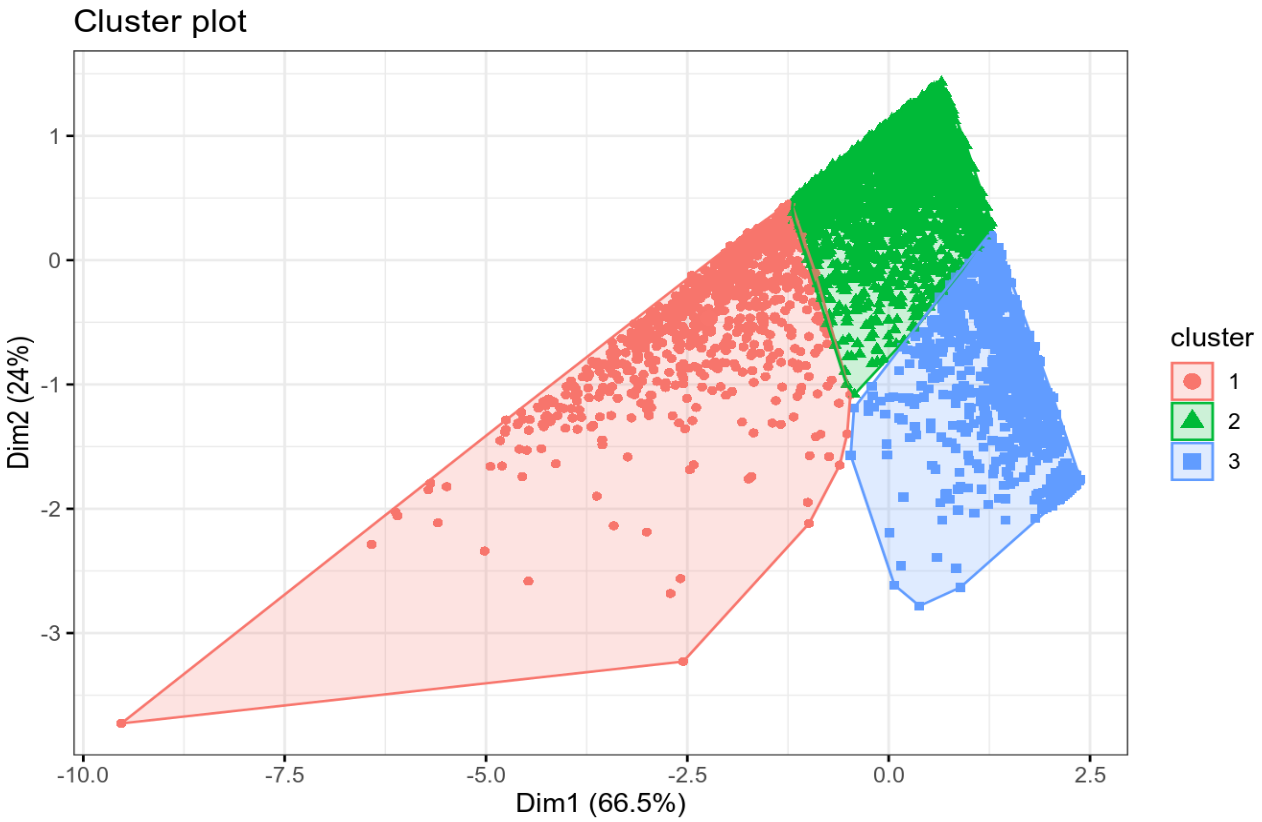
```
## # A tibble: 6 × 6
##   Customer.ID Recency[,1] Frequency[,1] Monetary[,1] putlerSegment kmeansSegme
nt
##         <int>       <dbl>         <dbl>        <dbl> <fct>         <fct>
## 1       12348      -0.240         0.426        1.10  LoyalCustome… 1
## 2       12349      -0.802        -0.745        1.05  PotentialLoy… 2
## 3       12350       2.07         -0.745       -0.667 Lost          3
## 4       12352      -0.624         1.99         1.95  LoyalCustome… 1
## 5       12353       1.03         -0.745       -0.963 Lost          3
## 6       12354       1.31         -0.745        0.233 Lost          3
```

Visualizing K-MEANS Segments

```
# Plotting 3D
colors = colorpalette[RFM$kmeansSegment]

plot3d = scatterplot3d(RFM$Recency,
                       RFM$Frequency,
                       RFM$Monetary,
                       xlab = "Recency",
                       ylab = "Frequency",
                       zlab = "Monetary",
                       main="K-means Customer Segments",
                       color = colors,
                       pch = 20)
legend(plot3d$xyz.convert(2.9, 10, 40),
       legend = levels(RFM$kmeansSegment),
       col = colorpalette, pch = 20)
```

```
fviz_cluster(k_result,
             data = df,
             geom="point",
             ellipse.type = "convex",
             ggtheme = theme_bw())
```


K-means Customer Segments
Monetary
4
3
2
1
0
-1
-2
Recency
-1
0
1
2
3
Frequency
15
10
5
0
-5

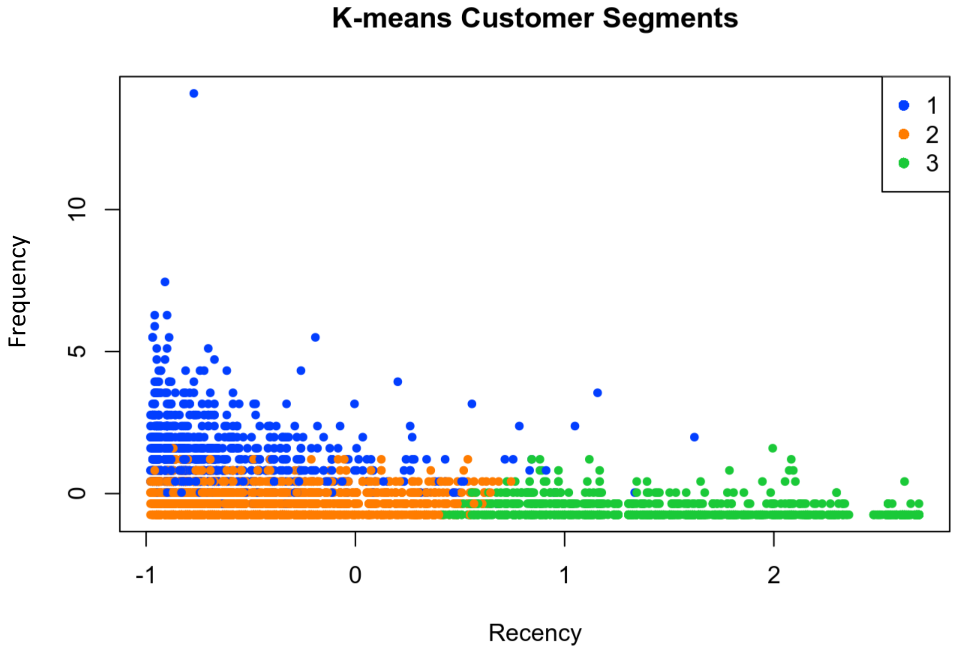

Cluster plot
Dim2 (24%)
1
0
-1
-2
-3
Dim1 (66.5%)
-10.0
-7.5
-5.0
-2.5
0.0
2.5
cluster
1
2
3

```
# number of customers in each K-means segment is:
RFM %>%
  group_by(kmeansSegment) %>%
  summarize(NumberOfCustomers = n()) %>%
  arrange(desc(NumberOfCustomers))
```

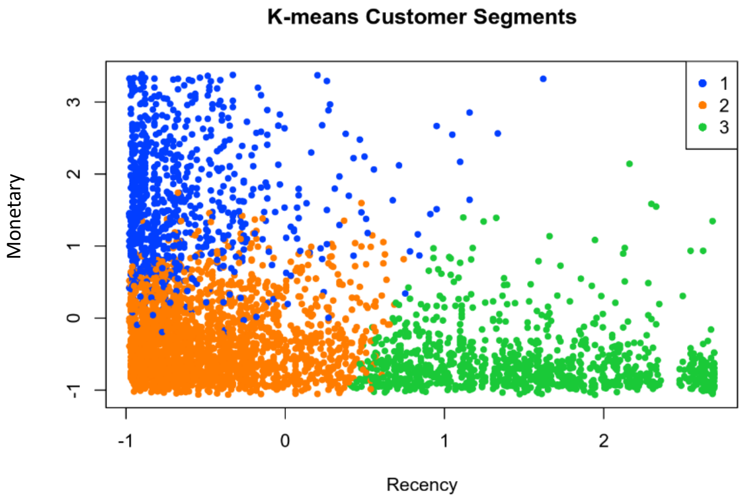
```
## # A tibble: 3 × 2
##   kmeansSegment NumberOfCustomers
##   <fct>                     <int>
## 1 2                          2131
## 2 3                           992
## 3 1                           788
```

Let's see what business interpretation we must give to 3 clusters

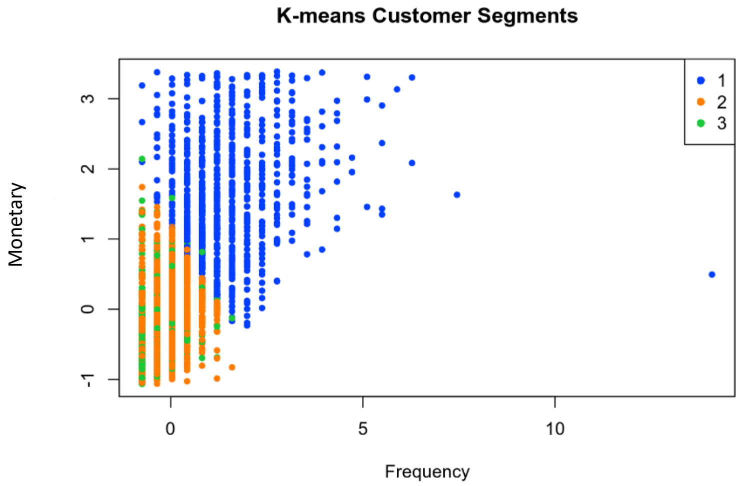
```
# Exhibit 1
colors = colorpalette[RFM$kmeansSegment]
plot(RFM$Recency,
     RFM$Frequency,
     xlab = "Recency", ylab = "Frequency",
     main="K-means Customer Segments",
     col = colors,
     pch = 20)
legend("topright",
       legend = levels(RFM$kmeansSegment),
       col = colorpalette,
       pch = 16)
```



```r
# Exhibit 2
plot(RFM$Recency,
     RFM$Monetary,
     xlab = "Recency", ylab = "Monetary",
     main="K-means Customer Segments",
     col = colors,
     pch = 20)
legend("topright",
        legend = levels(RFM$kmeansSegment),
        col = colorpalette,
        pch = 16)
```



```
# Exhibit 3

Frequency vs Monetary
```

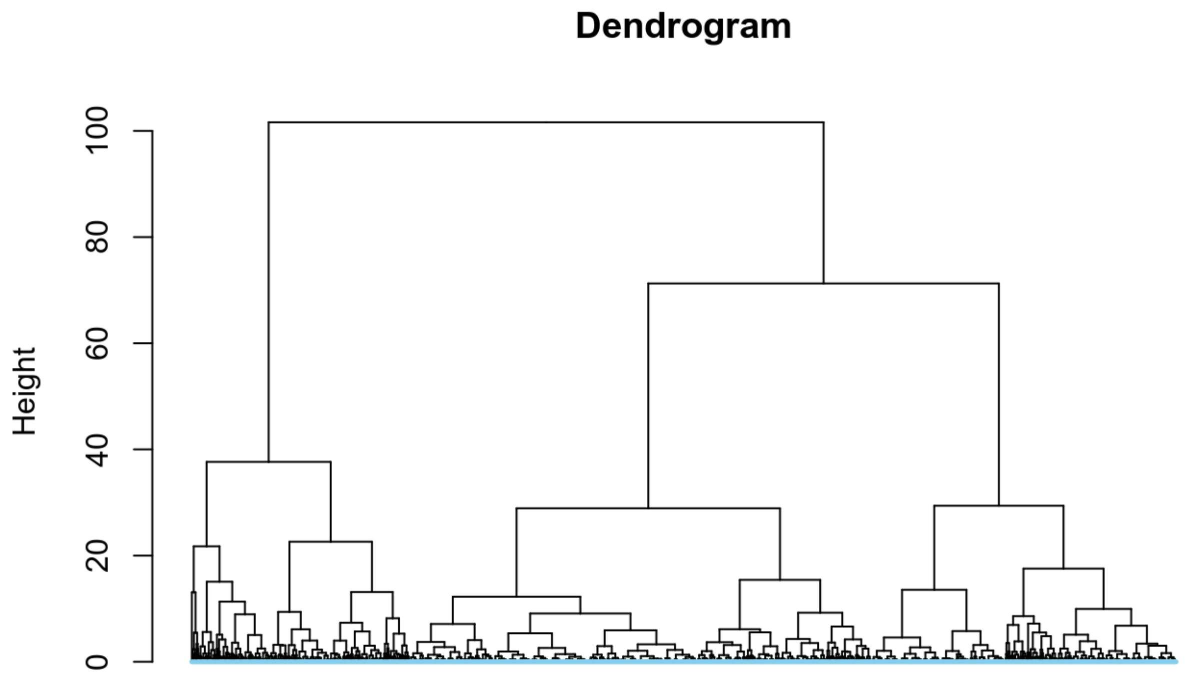


Customer Segmentation using the Hierarchical Clustering method

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly like each other.

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This iterative process continues until all the clusters are merged.

```
# calculate distance between vectors
D = dist(df, method='euclidean')

# H-Cluster
HC_result = hclust(D, method='ward.D2')

# Dendrogram, customizing the plot to remove labels
HC_model = as.dendrogram(HC_result)
nodePar = list(lab.cex = 0.6,
                pch = c(NA, 19),
                cex = 0.2,
                col = "skyblue")

plot(HC_model,
     ylab = "Height",
     nodePar = nodePar,
     leaflab = "none",
     main = "Dendrogram")
```



```
h_cluster = cutree(HC_result, k=3)
# Integrate RFM with H_clusters
RFM_HC = data.frame(RFM, h_cluster)
# size of groups

table(RFM_HC$h_cluster)
```

```
##
##    1    2    3
##  866 1834 1211
```

Snapshot of Customer Segments

```
head(RFM_HC)
```

```
##   Customer.ID    Recency  Frequency   Monetary     putlerSegment kmeansSegment
## 1       12348 -0.2403706  0.4255952  1.0990217    LoyalCustomers             1
## 2       12349 -0.8015356 -0.7454903  1.0511180 PotentialLoyalist             2
## 3       12350  2.0732046 -0.7454903 -0.6665468              Lost             3
## 4       12352 -0.6243256  1.9870425  1.9545048    LoyalCustomers             1
## 5       12353  1.0296345 -0.7454903 -0.9627313              Lost             3
## 6       12354  1.3052945 -0.7454903  0.2326278              Lost             3
##   h_cluster
## 1         1
## 2         2
## 3         3
## 4         1
## 5         3
## 6         3
```

Awesome! the hierarchical clusters match the ones we see in k-means. Let's see how DBSCAN takes this…

Customer Segmentation using the DBSCAN method

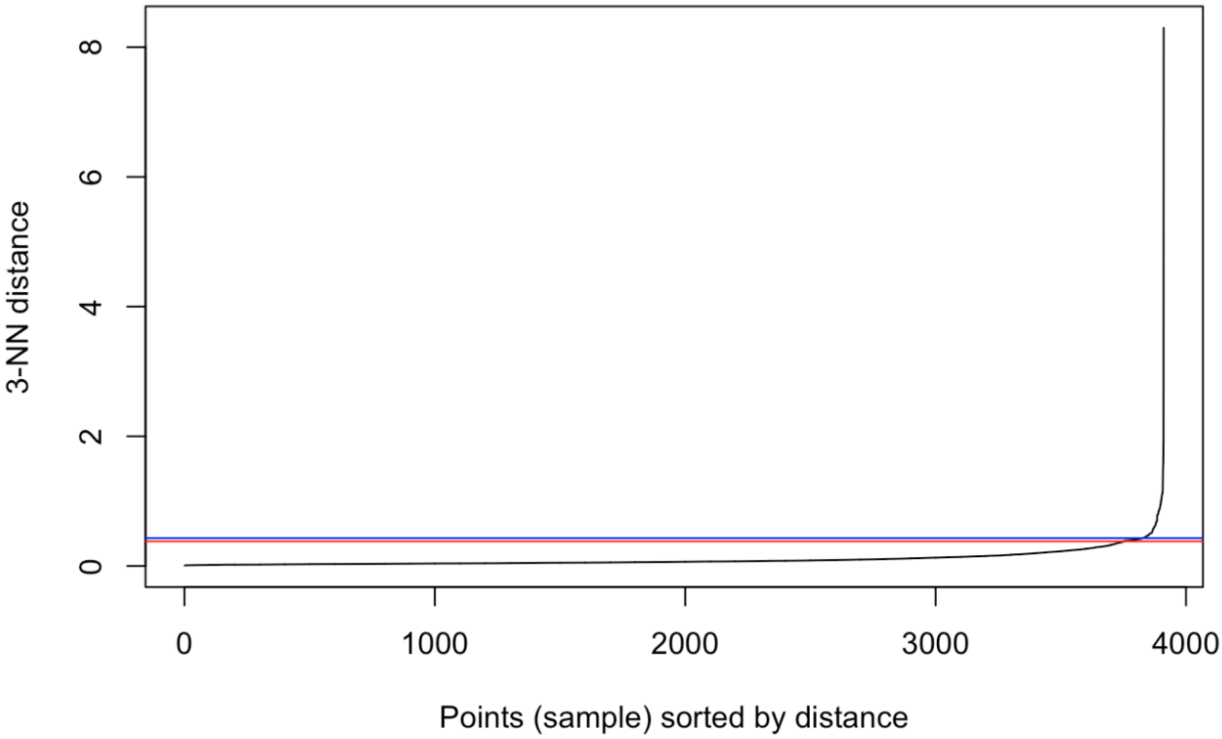
DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an unsupervised machine learning technique used to identify clusters of varying shape in a data set.

There are two parameters in the algorithms we need to estimate to get customer segmentations:

1. Minimum samples ("MinPts"): the fewest number of points required to form a cluster
2. ε (epsilon or "eps"): the maximum distance two points can be from one another while still belonging to the same cluster

For our dataset, we first used rule of thumb to determine minPts and eps, however, we only get one cluster. Then we adjust 2 parameters to match 3 clusters we got from 2 other methods.

```r
library(dbscan)
kNNdistplot(df, k = 3)
# abline(h = 0.1, col = "red")
# abline(h = 0.3, col = "red")
# abline(h = 0.4, col = "red")
abline(h = 0.38, col = "red")
abline(h = 0.43, col = "blue")
```



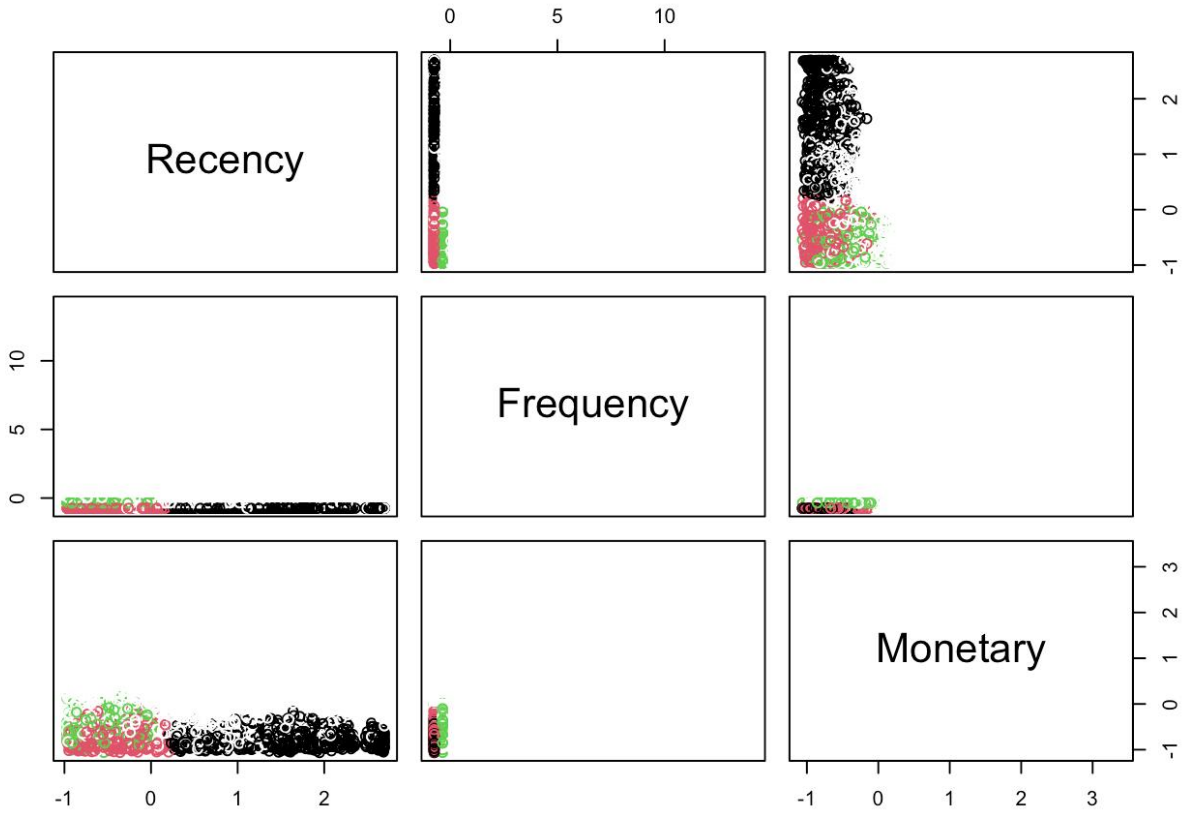
```r
#General roles for DBCAN: if the dataset dimension is larger than 2,
#the minPts is 2*dim
db0 <- dbscan(df, eps =0.43, minPts =6)
db0
```

```
## DBSCAN clustering for 3911 objects.
## Parameters: eps = 0.43, minPts = 6
## The clustering contains 1 cluster(s) and 82 noise points.
##    0    1
##   82 3829
##
## Available fields: cluster, eps, minPts
```

```
#Using our parameters from KNNdistplot and minPts,
#DBCAN can only capture one cluster.
db <- dbscan(df, eps =0.38, minPts =158)
db
## DBSCAN clustering for 3911 objects.
## Parameters: eps = 0.38, minPts = 158
## The clustering contains 3 cluster(s) and 2067 noise points.
##
##    0    1    2    3
## 2067  712  671  461
##
## Available fields: cluster, eps, minPts
```

Let’s check the plot...

```
fit.db <- db$cluster
plot(df, col = fit.db,color= colorpalette)
```



Association Rules

We approached the association rule to answer different questions…

1. Generic Association Rule Mining | Market Basket Analysis
2. Association Rule Mining by Customer Segments generated

In association rule mining, machine learning models (e.g., the apriori algorithm) are applied to transactional data to create rules with an "if-then" structure. The antecedent of each rule is the "if" statement, and the consequent is the "then" statement. If a customer has a certain product in his or her basket, then we can find the probability that he or she also has other products (the items in the consequent) in the basket. Confidence measures the probability that items in the antecedent and the consequent appear together. Support measures how often the rule is applicable to the dataset of transactions. Lift measures how much the probability of purchasing items in the consequent increases once it is known that the customer also has the items in the antecedent in his or her cart.

Association Rules: Retail and Wholesale Customers

To create association rules for retail and customers, we first read in the data. We used the full dataset to include the invoices from both retail and wholesale customers. The data must then be transformed to transaction data, so it is necessary to group by Invoice.

The descriptions of each product were used as the items in the basket – this makes the results much easier to interpret instead of using the stock code for each product. The apriori algorithm is then applied with a minimum confidence of 0.5 and a minimum support of 0.01. 340 rules are created, and then insignificant, redundant, and subset rules are removed before examining the top rules by confidence and lift.

```
INVOICE_LIST = data %>%
    group_by(Invoice) %>%
    summarize(StockList = list(unique(StockCode)),
        DescList  = list(unique(Description)))
head(INVOICE_LIST)

INVOICE_DESC=as(INVOICE_LIST$DescList, 'transactions')

DESCrules=apriori(INVOICE_DESC, parameter=list(support=0.01, confidence=0.5))
```

```
## set of 340 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 141 187  12
```

Association Rules: Retail Customers Only

Taking a similar approach to our previous work creating association rules, to create association rules for retail customers only, we first read in the data and look at only the relevant invoices for retail customers. Once again, the data must then be transformed to transaction data, so it is necessary to group by Invoice. We still used the descriptions of each product as the products in each basket since this is easier to interpret instead of using the stock code for each product. The apriori algorithm is then applied with a minimum confidence of 0.5 and a minimum support of 0.01. 256 rules are created, and then insignificant, redundant, and subset rules are removed before examining the top rules by confidence and lift.

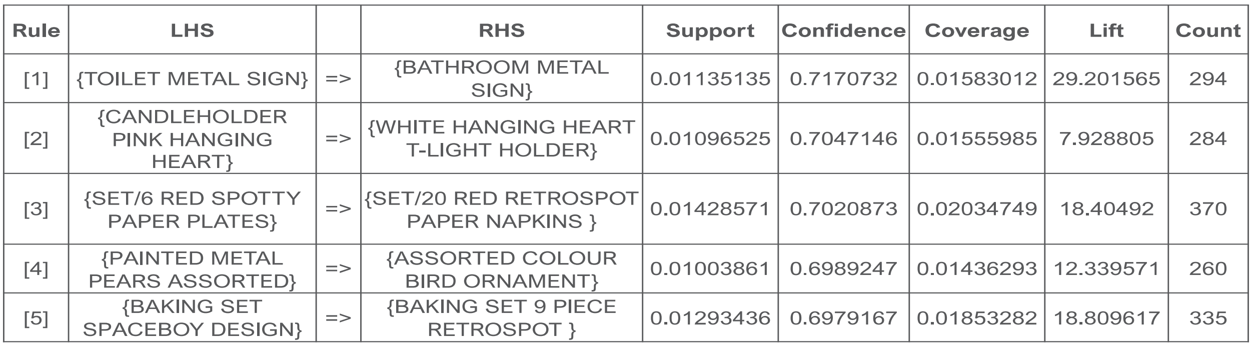
```
# Creating transaction data for each invoice
INVOICE_LIST = retail %>%
    group_by(Invoice) %>%
    summarize(StockList=list(unique(StockCode)), DescList  = list(unique(Descript
ion)))

INVOICE_DESC=as(INVOICE_LIST$DescList, 'transactions')

# Invoice-level rules using product descriptions
DESCrules=apriori(INVOICE_DESC, parameter=list(support=0.01, confidence=0.5))
```

```
## set of 265 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 139 118   8
```

```
# Keeping significant rules
DESCrules <- DESCrules[is.significant(DESCrules, INVOICE_DESC)]
# Removing redundant rules
DESCrules <- DESCrules[!is.redundant(DESCrules)]

# Removing subset rules
subsetRules <- which(colSums(is.subset(DESCrules, DESCrules)) > 1) # puts subset
rules in vector
DESCrules <- DESCrules[-subsetRules] # remove subset rules.

DESCrules_conf <- sort(DESCrules, by="confidence", decreasing=TRUE)
inspect(head(DESCrules_conf, 20)) # top 20 rules (confidence)

DESCrules_lift <- sort(DESCrules, by="lift", decreasing=TRUE)
inspect(head(DESCrules_lift, 20)) # top 20 rules (lift)
```

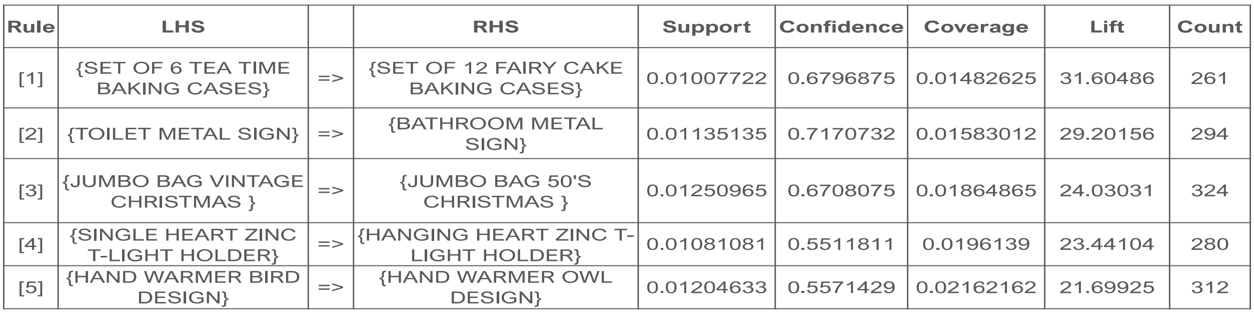
ASSOCIATION RULES OBSERVATIONS [MARKET BASKET ANALYSIS]

a) Top 5 Rules by Confidence: Retail & Wholesale Transaction-Level Data



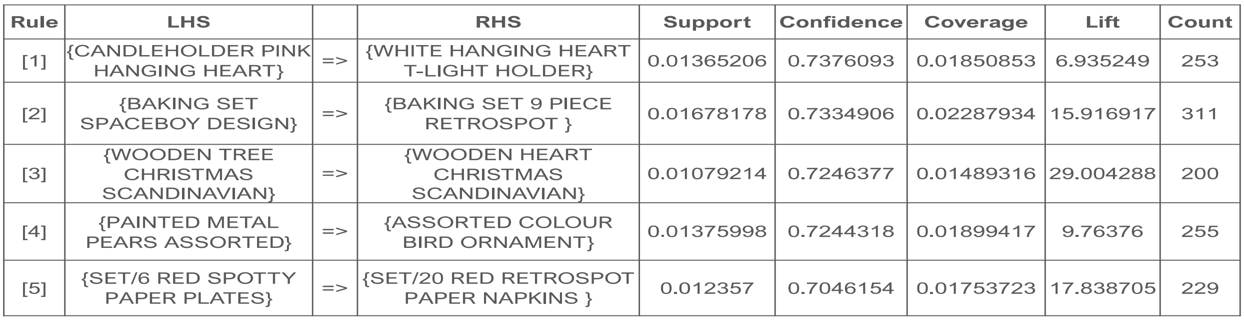
Rule	LHS		RHS	Support	Confidence	Coverage	Lift	Count
[1]	{TOILET METAL SIGN}	=>	{BATHROOM METAL SIGN}	0.01135135	0.7170732	0.01583012	29.201565	294
[2]	{CANDLEHOLDER PINK HANGING HEART}	=>	{WHITE HANGING HEART T-LIGHT HOLDER}	0.01096525	0.7047146	0.01555985	7.928805	284
[3]	{SET/6 RED SPOTTY PAPER PLATES}	=>	{SET/20 RED RETROSPOT PAPER NAPKINS }	0.01428571	0.7020873	0.02034749	18.40492	370
[4]	{PAINTED METAL PEARS ASSORTED}	=>	{ASSORTED COLOUR BIRD ORNAMENT}	0.01003861	0.6989247	0.01436293	12.339571	260
[5]	{BAKING SET SPACEBOY DESIGN}	=>	{BAKING SET 9 PIECE RETROSPOT }	0.01293436	0.6979167	0.01853282	18.809617	335

b) Top 5 Rules by Lift: Retail & Wholesale Transaction-Level Data



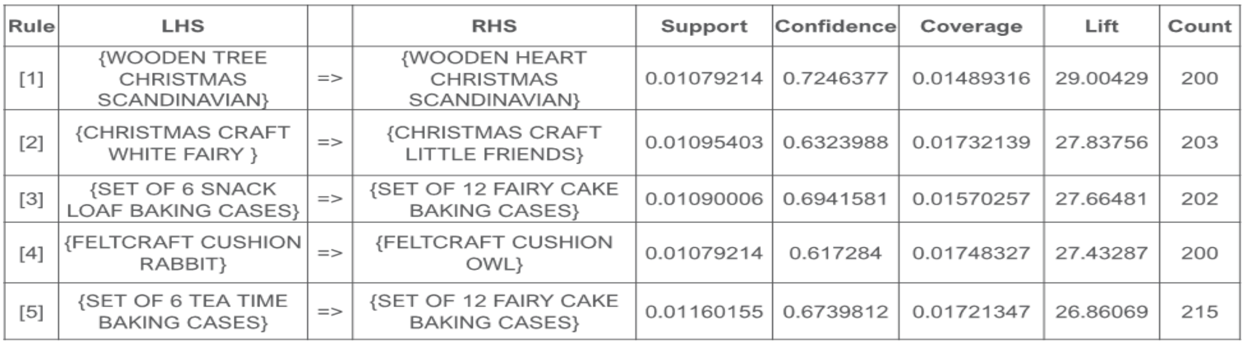
Rule	LHS		RHS	Support	Confidence	Coverage	Lift	Count
[1]	{SET OF 6 TEA TIME BAKING CASES}	=>	{SET OF 12 FAIRY CAKE BAKING CASES}	0.01007722	0.6796875	0.01482625	31.60486	261
[2]	{TOILET METAL SIGN}	=>	{BATHROOM METAL SIGN}	0.01135135	0.7170732	0.01583012	29.20156	294
[3]	{JUMBO BAG VINTAGE CHRISTMAS }	=>	{JUMBO BAG 50'S CHRISTMAS }	0.01250965	0.6708075	0.01864865	24.03031	324
[4]	{SINGLE HEART ZINC T-LIGHT HOLDER}	=>	{HANGING HEART ZINC T-LIGHT HOLDER}	0.01081081	0.5511811	0.0196139	23.44104	280
[5]	{HAND WARMER BIRD DESIGN}	=>	{HAND WARMER OWL DESIGN}	0.01204633	0.5571429	0.02162162	21.69925	312

c) Top 5 Rules by Confidence: Retail Transaction-Level Data



Rule	LHS		RHS	Support	Confidence	Coverage	Lift	Count
[1]	{CANDLEHOLDER PINK HANGING HEART}	=>	{WHITE HANGING HEART T-LIGHT HOLDER}	0.01365206	0.7376093	0.01850853	6.935249	253
[2]	{BAKING SET SPACEBOY DESIGN}	=>	{BAKING SET 9 PIECE RETROSPOT }	0.01678178	0.7334906	0.02287934	15.916917	311
[3]	{WOODEN TREE CHRISTMAS SCANDINAVIAN}	=>	{WOODEN HEART CHRISTMAS SCANDINAVIAN}	0.01079214	0.7246377	0.01489316	29.004288	200
[4]	{PAINTED METAL PEARS ASSORTED}	=>	{ASSORTED COLOUR BIRD ORNAMENT}	0.01375998	0.7244318	0.01899417	9.76376	255
[5]	{SET/6 RED SPOTTY PAPER PLATES}	=>	{SET/20 RED RETROSPOT PAPER NAPKINS }	0.012357	0.7046154	0.01753723	17.838705	229

d) Top 5 Rules by Lift: Retail Transaction-Level Data



Rule	LHS		RHS	Support	Confidence	Coverage	Lift	Count
[1]	{WOODEN TREE CHRISTMAS SCANDINAVIAN}	=>	{WOODEN HEART CHRISTMAS SCANDINAVIAN}	0.01079214	0.7246377	0.01489316	29.00429	200
[2]	{CHRISTMAS CRAFT WHITE FAIRY }	=>	{CHRISTMAS CRAFT LITTLE FRIENDS}	0.01095403	0.6323988	0.01732139	27.83756	203
[3]	{SET OF 6 SNACK LOAF BAKING CASES}	=>	{SET OF 12 FAIRY CAKE BAKING CASES}	0.01090006	0.6941581	0.01570257	27.66481	202
[4]	{FELTCRAFT CUSHION RABBIT}	=>	{FELTCRAFT CUSHION OWL}	0.01079214	0.617284	0.01748327	27.43287	200
[5]	{SET OF 6 TEA TIME BAKING CASES}	=>	{SET OF 12 FAIRY CAKE BAKING CASES}	0.01160155	0.6739812	0.01721347	26.86069	215

We see in the results for the full customer base (i.e., retail and wholesale customers) that most rules in the rules with the highest confidence and lift have products in the antecedent and consequent that only differ by colour or pattern. For example, Rule 5 in the highest confidence set of rules consists of two different baking sets: one with the “space boy design” and one with the “retro spot” pattern.

After creating and examining the association rules for only the retail customers in the data, we still see very similar products in the antecedents and consequents. However, the differences in products become more noticeable – there are fewer rules where it is obvious that the products are identical except in colour or design. For example, the rule with the highest lift contains what we assume are two different Christmas decorations (“Wooden tree Christmas Scandinavian” and “Wooden heart Christmas Scandinavian”). It is not unusual or unexpected for a retail customer to buy these items together. These items are decorations, and when customers buy decorations, they might often buy similar products to decorate in one cohesive theme or style. We still see the similar baking sets in Rule 2 of the highest confidence rules, but instead of this reflecting wholesale customers, we concluded that this is simply a reflection of the type of retailer we are studying: a gift shop. If a retail customer is buying gifts, it is not unusual to buy the same type of gift in different styles to give to others. When creating the association rules for only retail customers, the maximum confidence we found was about 73.8%, and the maximum lift was about 29.

While it is helpful to create these rules to understand the customer base, there is much more power and potential applications if these rules are created for each customer segment. This was our next area of exploration.

Association Rule Mining with Customer Segment – 1 : Champion Customers

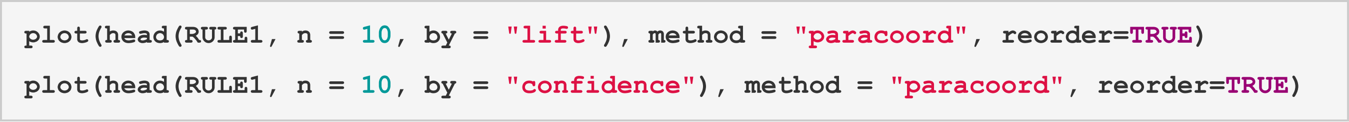
The idea here is to apply association rule mining by the customer segments from the results of the clustering techniques like k-means, hierarchal.

Libraries: arules; arulesviz

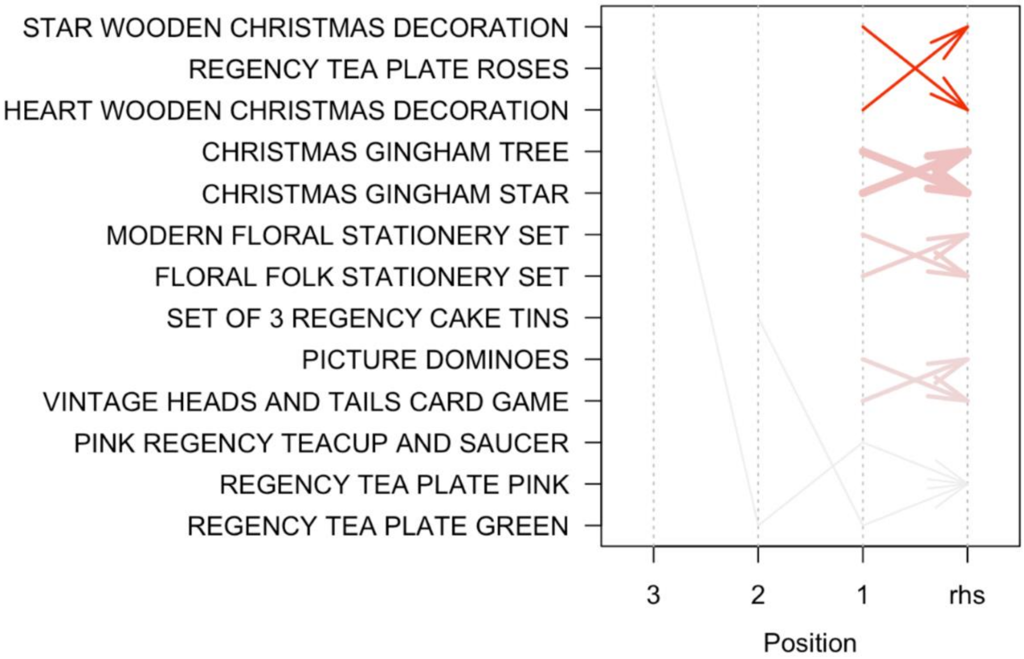
```
INVOICE_LIST1=DATA1 %>%
    group_by(Invoice) %>%
    summarize(StockList=list(unique(StockCode)), DescList  = list(unique(Descript
ion)))

INVOICE_DESC1 = as(INVOICE_LIST1$DescList, 'transactions')

DESCrules1 = apriori(INVOICE_DESC1, parameter=list(support=0.01, confidence=0.5))
RULE1 <- DESCrules1[!is.redundant(DESCrules1) & is.significant(DESCrules1, INVOIC
E_DESC1)]
```

You can look at the rules generated

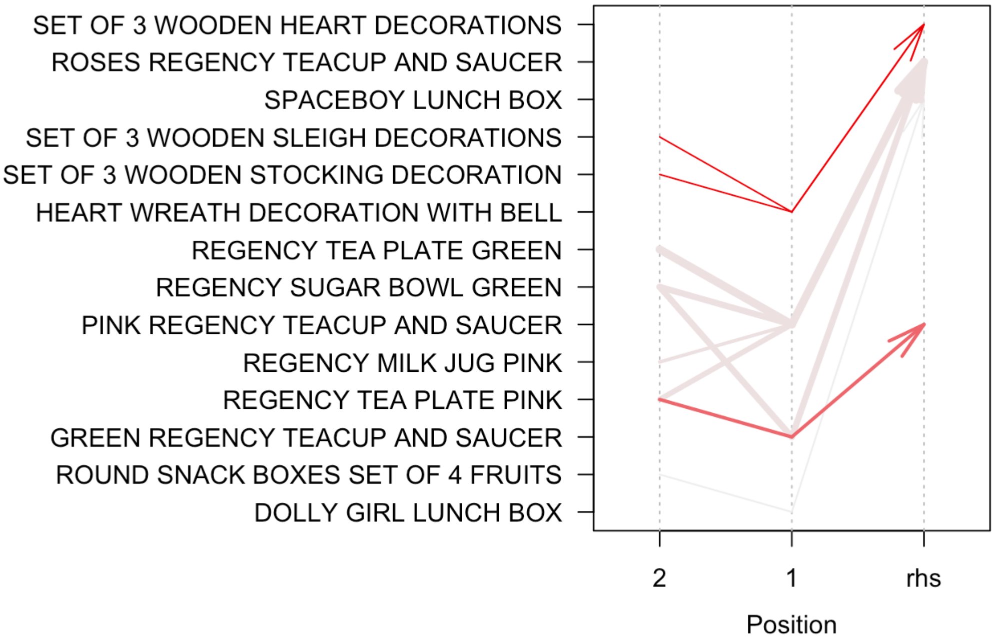
```
inspect(head(RULE1, n = 10, by = "lift"))

inspect(head(RULE1, n = 10, by = "confidence"))
```



```
plot(head(RULE1, n = 10, by = "lift"), method = "paracoord", reorder=TRUE)
plot(head(RULE1, n = 10, by = "confidence"), method = "paracoord", reorder=TRUE)
```



Ten rules with highest lift regarding Champion Customers
(Lift = 41-57, Confidence = 0.6 - 0.82, Count = 18 – 23, Support = 1 – 1.3%)



Ten rules with highest confidence regarding Champion Customers
(Lift = 18-33, Confidence = 1, Count = 18 – 25, Support = 1 – 1.3%)

From Figure-1, the “Heart wooden Christmas decoration” and “Star wooden Christmas decoration” have the strongest correlation amongst the other gifts. This rule can be read as, people who buy “Heart wooden Christmas decoration” tend to buy “Star wooden Christmas decoration” with a support, confidence and lift of 0.011, 0.75 and 57.07 respectively. A support of 1.1% means that out of all the Champion Customers, 1.1% of them have “Heart wooden Christmas decoration” and “Star wooden Christmas decoration” purchased together. A confidence of 76%, indicates that given a Champion customer bought “Heart wooden Christmas decoration”, has a 76% chance of purchasing “Star wooden Christmas decoration”. Finally, since the lift is 57.07 which is much greater than 1, it designates that “Heart wooden Christmas decoration” and “Star wooden Christmas decoration” are highly dependent on each other.

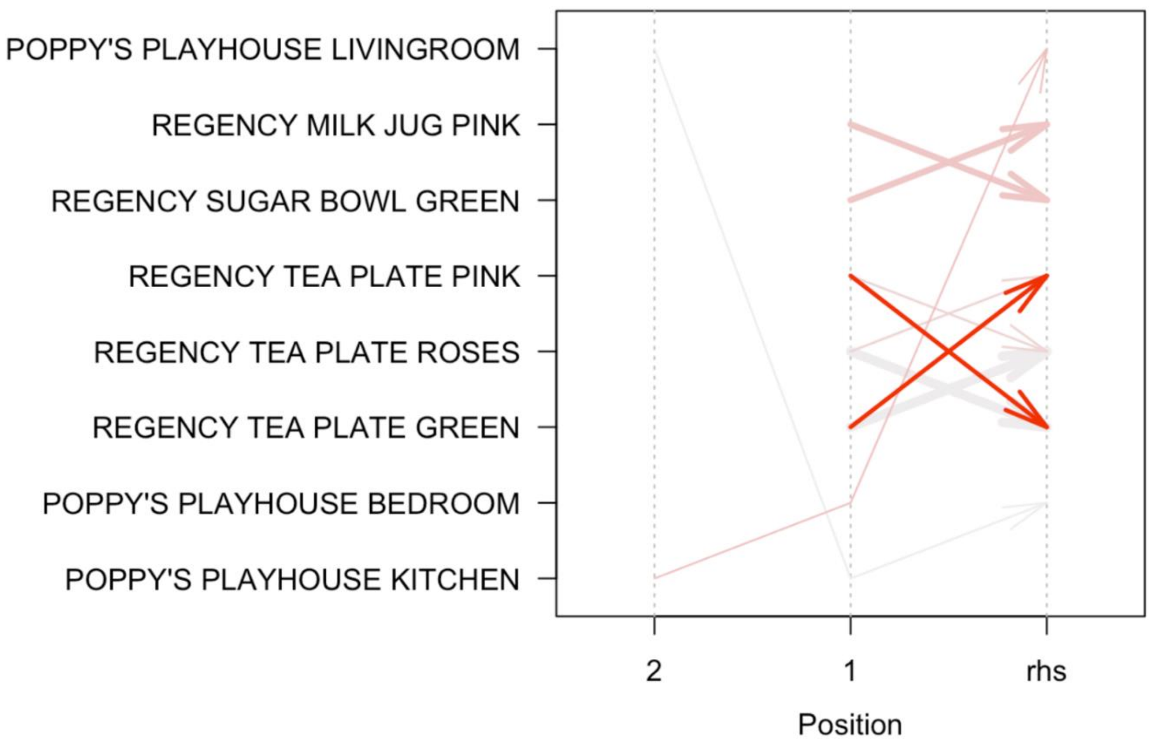
Similarly, one notable rule with 100% confidence from second figure is that customers who buy “Pink regency teacup and saucer” and “Regency milk jug pink” tend to buy “Roses regency teacup and saucer”. This rule has support of 0.012, lift of 18.77, and can be explained as out of all the Champion Customers, 1.15% of them have “Pink regency teacup and saucer”, “Regency milk jug pink” and “Roses regency teacup and saucer” purchased together. Furthermore, 100% of Champion Customers who buy “Pink regency teacup and saucer” and “Regency milk jug pink”, also decided to buy “Roses regency teacup and saucer”.

Those mentioned rules are not notable when we consider the whole population of customers, because the support is too low. Performing market basket analysis on each customer segment would help us to discover more interesting rules that can help to increase revenue of that segment.

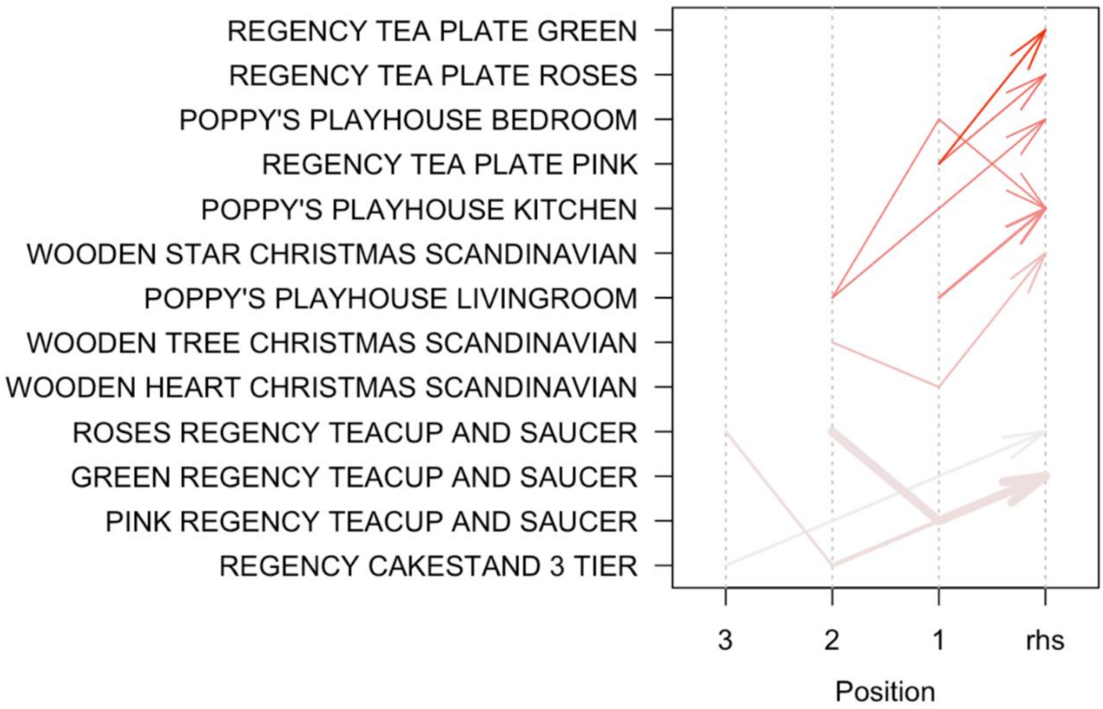
From the twenty rules with highest lift & confidence, we can see Champion Customers usually buy multiple Christmas gifts, parlor games, stationery sets, school lunch boxes together. Based on the gifts that were usually purchased together, it seems reasonable to think about the gift receivers of Champion Customers as middle-aged people who have school kids and are into indoor group games & activities.

ASSOCIATION RULE MINING WITH CUSTOMER SEGMENT – 2 : CUSTOMERS NEEDING ATTENTION

```
INVOICE_LIST2=DATA2 %>%
    group_by(Invoice) %>%
    summarize(StockList=list(unique(StockCode)), DescList  = list(unique(Descript
ion)))

INVOICE_DESC2=as(INVOICE_LIST2$DescList, 'transactions')

DESCrules2=apriori(INVOICE_DESC2, parameter=list(support=0.01, confidence=0.5))
RULE2<- DESCrules2[!is.redundant(DESCrules2) & is.significant(DESCrules2, INVOICE
_DESC2)]


plot(head(RULE2, n = 10, by = "lift"), method = "paracoord", reorder=TRUE)

plot(head(RULE2, n = 10, by = "confidence"), method = "paracoord", reorder=TRUE)
```



Ten rules with highest lift regarding Needing Attention Customers
(Lift = 51-67, Confidence = 0.73 - 0.9, Count = 153 – 171, Support = 1 – 1.1%)



Ten rules with highest confidence regarding Customers Needing Attention
(Lift = 25-67, Confidence = 0.87-0.93, Count = 153 – 298, Support = 1 – 1.9%)

From the twenty rules with highest lift and confidence above, it seems like Customers Needing Attention usually buy kitchenware's, indoor playsets for toddlers and Christmas decorations together. The gift receiver of those customers might be young parents with small children.

"Regency tea plate green" and "Regency tea plate pink" are most frequently bought together (Figure) with the support, confidence and lift of 0.011, 0.78 and 67.1 respectively. Out of all the customers who needs attention, 1.1% of them have those two plates purchased together. A confidence of 76% indicates that given a customer in this segment who bought "Regency tea plate green", they have a 78% chance of purchasing "Regency tea plate pink". We can conclude there is strong link between the two products since the lift is exceptionally high (67.1).

Based on Figure 2 and the result above, we can see that given a customer who buys "Poppy's playhouse living room", there is a 93.2% confidence that they will also buy "Poppy's playhouse kitchen". The support of 0.01 and lift of 49.9 indicate that there is a strong relationship between the two items. This can be applied to the marketing strategy for Customers Needing Attention segment.

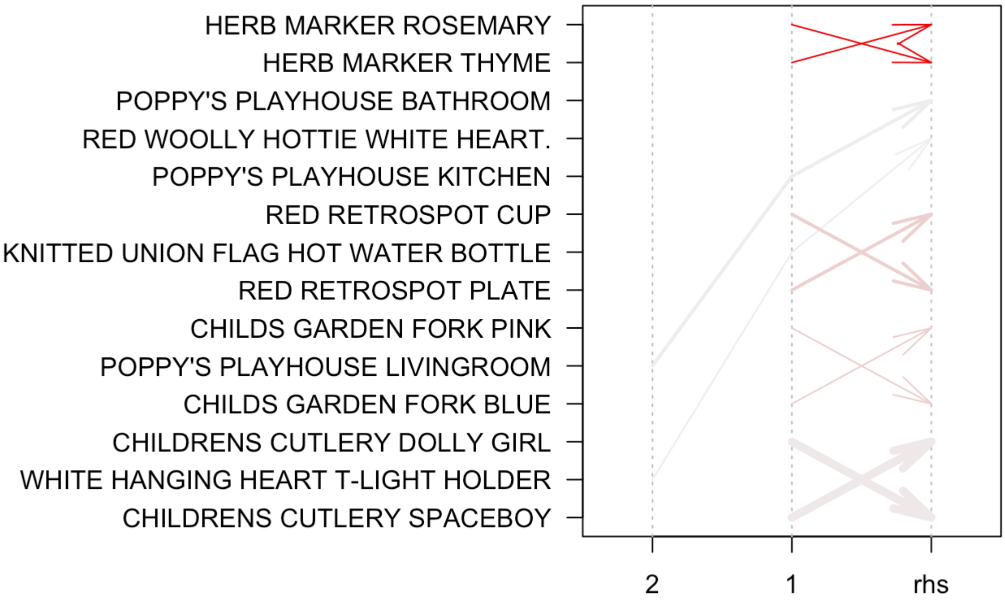
ASSOCIATION RULE MINING WITH CUSTOMER SEGMENT – 3 : AT RISK/ 1-TIME CUSTOMERS

```
INVOICE_LIST3=DATA3 %>%
    group_by(Invoice) %>%
    summarize(StockList=list(unique(StockCode)), DescList  = list(unique(Descript
ion)))

INVOICE_DESC3=as(INVOICE_LIST3$DescList, 'transactions')

DESCrules3=apriori(INVOICE_DESC3, parameter=list(support=0.01, confidence=0.5))
RULE3<- DESCrules3[!is.redundant(DESCrules3) & is.significant(DESCrules3, INVOICE
_DESC3)]
plot(head(RULE3, n = 10, by = "lift"), method = "paracoord", reorder=TRUE)

plot(head(RULE3, n = 10, by = "confidence"), method = "paracoord", reorder=TRUE)
```

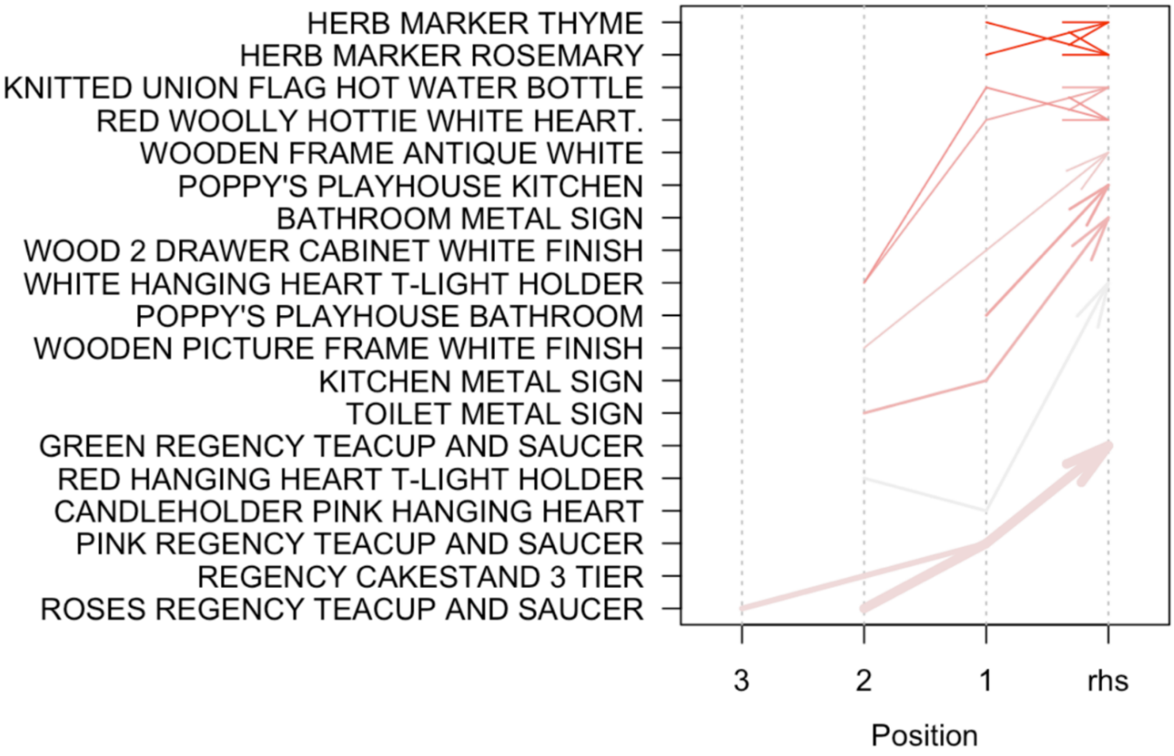
Overall, At-risk/1-time customers have preference for buying kitchen utensils, cutleries, specialized kitchen equipment's, house decorations, and indoor playsets for young children together. This is different from the analysis result of the other customer segments.

"Herb Maker Rosemary" and "Herb marker thyme" have the strongest correlation amongst all gifts, with the support, confidence and lift of 0.01, 1 and 95.5 respectively. This rule can be explained as, of the 1% of all At-risk/1-time customer who decide to buy "Herb Maker Rosemary", also purchase "Herb marker thyme". This rule is the most significant one in terms of both confidence and lift.

The second notable rule in this segment is that customer who buy "Kitchen metal sign" and "Toilet metal sign" have 100% chance to buy "Bathroom metal sign". With the solid support of 0.012 and high lift of 37.4, this rule can be very useful for marketers to use in campaigns for At-risk/1-time customers.



Ten rules with highest lift regarding At risk/1-time customers
(Lift = 54-95, Confidence = 0.70-1, Count = 18-22, Support = 1-1.1%)



Ten rules with highest confidence regarding At risk/1-time customers
(Lift =6.8 - 95 , Confidence = 0.91-1, Count = 18-42, Support = 1 – 1.9%)

REFERENCES

[1] Davenport, T.H. (2009) “Realizing the Potential of Retail Analytics: Plenty of Food for Those with the Appetite.” Working Knowledge Report, Babson Executive Education.

[2] Fuloria, S. (2011) “How Advanced Analytics Will Inform and Transform U.S. Retail. Cognizant Reports”, July, http://www.cognizant.com/InsightsWhitepapers/How-Advanced-Analytics-Will-Inform-and-Transform-US-Retail.pdf, accessed January 2012.

[3] Zhen, You. (2015) “A decision-making framework for precision marketing”, Expert Systems with Applications 42 (2015) 3357-3367.

[4] Wei, J. T., Lee, M. C., Chen, H. K., Wu, H. H. (2013). “Customer relationship management in The hairdressing industry: An application of data mining techniques.” Expert Systems with Applications, 40(18), 7513-7518.

[5] https://www.putler.com/rfm-analysis