Duke University

EGRMGMT 590

Marketing Analytics and Research

Analyzing Customer Purchasing Behavior

TEAM 1

Yash Bansal

Aditya Akole

Bhavya Gursahani

Kunwar Sahib Singh Vohra

1 ABSTRACT

This report defines a full end to end marketing analytics project performed by Team 1 of EGRMGMT 590 – Marketing Analytics class of Duke University. This project was performed in R language and uses various marketing analytics and decision-making tools aimed to offer solutions to various business needs of brick and mortar stores.

This project was particularly performed for Walmart brick and mortar retail store. It consists of analyzing the market basket of customers. Thus, the main goal is to find which items are purchased together in their stores.

This project report will take users through the entire analysis step by step, starting by defining the problem to giving business focused recommendations towards the end. Moreover, the reader will see various visualizations throughout the report that will make it easier to understand the analysis.

At the end, reader will have a general idea of how marketing analytics can be used to drive results and solve real life problems in today's brick and mortar stores.

Contents

2 INTRODUCTION

2.1 INTRODUCTION & MOTIVATION

Brick & Mortar stores have been around since forever and are an integral part of our everyday lives. They have been around the corner since 1900's. We all have our favorite retail stores that we love to go to whether it be Walmart, Target, Trader Joes or any other. Ever imagined a day when they cease to exist? E-commerce is a major threat to Brick & Mortar stores and has shaken the entire industry since its birth. There are new possibilities that have come to light for both consumers and also for commercial businesses.

The only way Brick & Mortar stores have been able to survive for over a century now is by evolving every time they have been hit by a difficulty and now once again there is a need to evolve in order to survive. Major problems being faced are [1] drop-in sales, [2] drop-in customer satisfaction rate and [3] availability of alternate options to buy from.

To better understand the situation, we need to understand the customer mindset in today's world. With the new era of information and globalization, the list of options has increased exponentially. Now consumers can choose between a huge variety of products, special thanks to big e-commerce businesses such as amazon. The convenience of not getting out of the house and still getting products delivered to your doorstep is a huge advantage for online retail store. But, on the same time Brick & Mortar provide the ability of look and feel of products before purchasing, adding on to their advantage.

> *"People are always going to go shopping. A lot of our effort is just 'how do we make the retail experience a great one?"- Phillip Green, Chairman, Arcadia Group*

With the global COVID-19 pandemic scenario, consumer shopping trends have been greatly affected. One of the most impacted domains is the fast-moving consumer goods category. Mega stores such as Walmart, Target, Food Lion, etc. cater heavily to the everyday needs of consumers. However, the pandemic has greatly decreased the number of visits paid by customers to these stores. Customers prefer purchasing in bulk quantities per visit and thus it is a lucrative opportunity to perform market basket analysis to predict consumer purchase trends. The predictions will greatly help in understanding the [1] inventory requirements of

these megastores, [2] forecast product demands and [3] provide combo offers to consumers for increasing product consumption. Overall leading to a greater customer experience of visitors.

2.2 MARKET BASKET ANALYSIS

Image 1: MBA

Market Basket Analysis is a technique that looks for combinations of products that occur in purchases. Its premise is that customers who buy a particular group of products are more or less likely to buy another group of products. Market basket analysis encompasses a broad set of analytics techniques aimed at uncovering the associations and connections between specific objects, discovering customers behaviors and relations between items. This way marketing and sales teams can develop more effective pricing, product placement, cross-sell and up-sell strategies.

We will use MBA technique in our project to make recommendation on [1] placement of the product, [2] determine bundle purchase offers, [3] marketing promotions, [4] channel optimization, [5] increase up-sell and cross-sell.

To dive deeper MBA will also be used to make customer-based and item-based offers and points after analyzing the data of a retail store over a period of time. This is aimed towards increasing the customer satisfaction rate and net promoter score.

2.3 Dataset

There were a lot of retail datasets focusing on customer purchase records available online. Most of these datasets had similar variables. For this project we decided to choose the data set from Kaggle (https://www.kaggle.com/heeraldedhia/groceries-dataset). The dataset contains 38765 records of purchases made by customers. These records are split into 3 variables or columns:

1) Member_number: Unique identifier associated with each individual customer.
2) Date: Date of purchase of each individual item
3) itemDescription: Description of the purchased product.

Each record has a unique value of itemDescription and Member_number. This allows us to easily group and classify items in groups by members as shown below.

	Member_number <int>	Date <chr>	itemDescription <chr>
1	1808	21-07-2015	tropical fruit
2	2552	5/1/2015	whole milk
3	2300	19-09-2015	pip fruit
4	1187	12/12/2015	other vegetables
5	3037	1/2/2015	whole milk
6	4941	14-02-2015	rolls/buns

Image 2: Original Dataset

3 DATA EXPLORATION

3.1 Cleaning and Basic Analysis

The first and foremost thing we had to do with this dataset was to check missing values. Fortunately, our dataset did not have any missing values and hence there was no need to remove rows with missing data.

```
# Checking NA values
#is.na(grocery)
head(is.na(grocery))
sum(is.na(grocery))
```

```
     Member_number  Date itemDescription
[1,]         FALSE FALSE            FALSE
[2,]         FALSE FALSE            FALSE
[3,]         FALSE FALSE            FALSE
[4,]         FALSE FALSE            FALSE
[5,]         FALSE FALSE            FALSE
[6,]         FALSE FALSE            FALSE
[1] 0
```

Image 3: Checking for NA values in dataset using R

The first challenge with the dataset was the standardization and formatting of the dates. The dates had different formats: dd-mm-yyyy, dd/mm/yyyy, d/m/yyyy, etc. We fixed thev date column by setting it to a standard dd/mm/yyyy format.

```
### fixing date column
```{r}
#Steps:
#  replace - with /
#  change to datetime type
grocery$Date <- gsub('-', '/', grocery$Date)
#head(grocery)


grocery<- grocery %>%
  mutate(Date_corrected=gsub('-', '/', Date),
         Date_corrected=format(as.Date(Date_corrected, format = "%d/%m/%Y"), "%d/%m/%Y"))%>%
  arrange(Date_corrected,decreasing=FALSE)
head(grocery)
```
```

	Member_number <int>	Date <chr>	itemDescription <chr>	Date_corrected <chr>
1	2351	1/1/2014	cleaner	1/1/2014
2	2226	1/1/2014	sausage	1/1/2014
3	1922	1/1/2014	tropical fruit	1/1/2014
4	2943	1/1/2014	whole milk	1/1/2014
5	1249	1/1/2014	citrus fruit	1/1/2014
6	3681	1/1/2014	onions	1/1/2014

Image 4: Formatting and Cleaning dataset

The next step in the exploration was to identify the number of unique values in each column. This is a major step in this project since we need to understand the total number of members and the total number of individual items sold in the store.

```{r}
length(unique(grocery$Date_corrected))
length(unique(grocery$Member_number))
length(unique(grocery$itemDescription))
```

```
[1] 0
[1] 3898
[1] 167
```

Image 5: Dataset Exploration

This dataset contains the purchase history of 3898 individual members over a period of 728 days. From the data set it is clear that the store sold 167 edible items such as whole milk, tropical fruits, vegetables, eggs, etc. From our initial observations, it was clear that we would be able to derive valuable insight into consumer buying trends and how the store can use it to their advantage to promote certain products. These insights will also help the store identify key promotions which it should apply to those products.

Getting a summary of buying trends also shows us that the transactions are pretty evenly distributed across all days of the week:

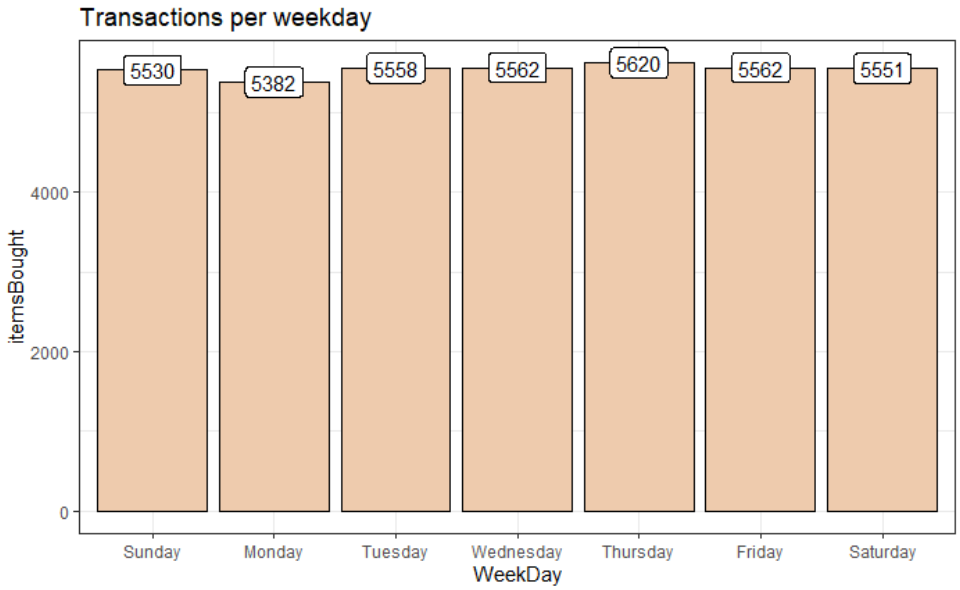

Image 6: Transactions per day over the period of 728 days

Exploring further, we also identified the months with the highest sales across the 728 days:

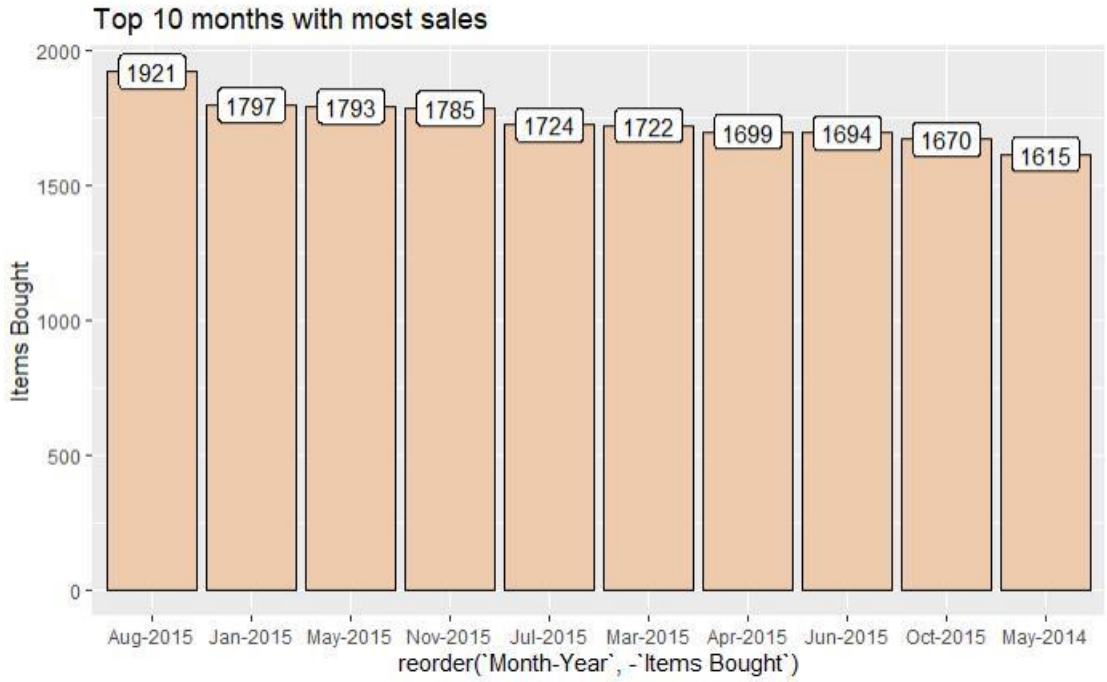

Image 7: Top 10 months with most Sales

One of the first analysis we conducted was to figure out the items that were sold the most in the 728-day period. Accordingly, the top 10 products sold in the store along with their frequency are as follows:

	itemDescription	**Frequency**
165	whole milk	2502
102	other vegetables	1898
122	Rolls/buns	1716
138	Soda	1514
166	Yogurt	1334
123	Root vegetables	1071
156	Tropical fruit	1032
11	Bottled water	933
130	Sausage	924
29	Citrus fruit	812

Image 8: Table of Top 10 products sold

```r
itemFrequencyPlot(basket, topN = 10, type = 'absolute', ylim = c(0,3000), col =
"peachpuff2")
items <- as.data.frame(itemFrequency(basket))
###make bottom 10 also
```

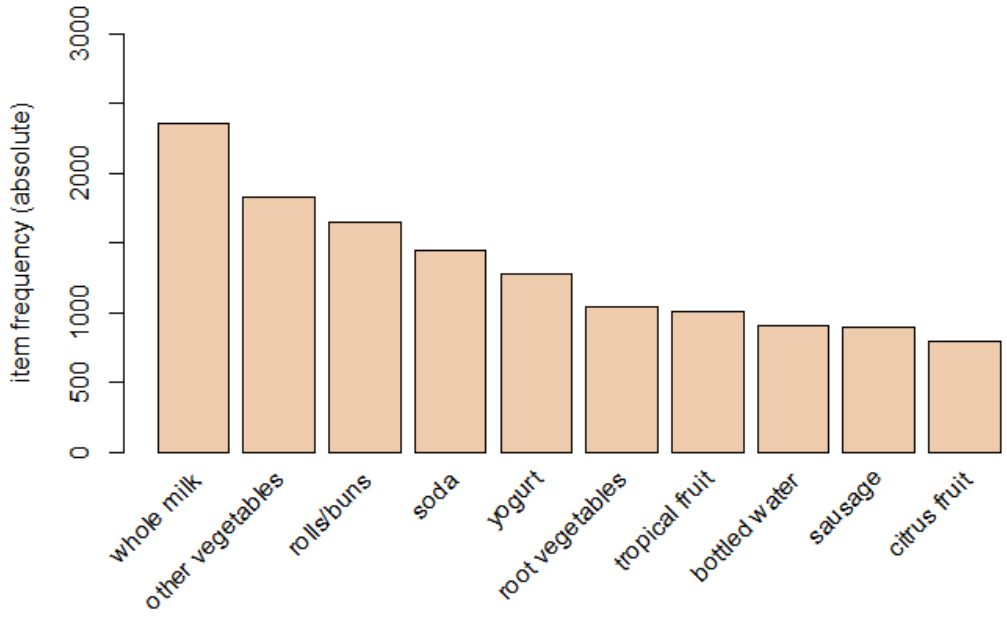

Image 9: Graphical representation of top 10 products sold

Our next step in the process was to prepare the input data for our analysis. This included sorting the dataset according to Member_number to get the transactions conducted by each and every member over the period of 728 days. This was done by creating a list column that included the items that were bought together by a particular member on a certain date.

```r
### Preparing input data for Market Basket analysis
```r
# Sorted data set according to Member_number
sorted_grocery <- grocery[order(grocery$Member_number),]
## Converting member number to numeric
sorted_grocery$Member_number <- as.numeric(sorted_grocery$Member_number)
#head(sorted_grocery)
library(plyr)
itemList <- ddply(sorted_grocery, c("Member_number","Date"),
function(df1)paste(df1$itemDescription,collapse = ","))
itemList<-itemList %>%
  arrange(Date, decreasing = FALSE)
colnames(itemList) <- c("Member_number", "Date", "item_List")
head(itemList)
```
```

Image 10: Preparing data for Market Basket Analysis

	Member_number <dbl>	Date <chr>	item_List <chr>
1	1249	1/1/2014	citrus fruit,coffee
2	1381	1/1/2014	curd,soda
3	1440	1/1/2014	other vegetables,yogurt
4	1659	1/1/2014	specialty chocolate,frozen vegetables
5	1789	1/1/2014	hamburger meat,candles
6	1922	1/1/2014	tropical fruit,other vegetables

Image 11: Itemized list for Market Basket Analysis

3.2 Hypotheses Testing

The objective was to understand the trend of visits made by the customers to the store to find out the most suitable time of the week to launch new products and special offers. We evaluated this by the hypothesis testing method. We defined a preliminary null hypothesis (Ho) and then used statistical testing methods to decide.

Null Hypothesis (Ho): The number of visits made to the Store on Weekdays is lesser than the visits made on Weekends.

```r
```{r}
grocery<- grocery %>%
  mutate(weekday_end=ifelse(WeekDay== "Saturday" | WeekDay== "Sunday", "weekend","weekday"))
members<-days_visit_summary %>%
  filter(days_visited>=10) %>%
  select(Member_number)

members<-as.list(members$Member_number)


  subset <- grocery %>%
    filter(Member_number %in% members) %>%
    group_by(itemDescription) %>%
    dplyr::summarise(count=n())

    subset2 <- grocery %>%
    filter(Member_number %in% members) %>%
    group_by(Member_number,itemDescription) %>%
    dplyr::summarise(count=n())

  subset3 <- grocery %>%
    group_by(Date_corrected, Member_number) %>%
    dplyr::summarise(count=n())

  subset4 <- subset3 %>%
    group_by(Date_corrected) %>%
    dplyr::summarise(count=n())

  subset5<-subset4%>%
  mutate(WeekDay=factor(weekdays(as.Date(Date_corrected, format =
"%d/%m/%Y")),levels=c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")))

  subset6<-subset5%>%
  mutate(weekday_end=ifelse(WeekDay== "Saturday" | WeekDay== "Sunday", "weekend","weekday"))
```
```

Image 12: Hypotheses Testing

```r
## To determine if there is any difference between the number of visits by all members on Weekdays Vs. Weekends
```{r}
library(tidyverse)
library(ggpubr)
library(rstatix)
week_end <- subset6 %>%
  filter(weekday_end == "weekend") %>%
  pull(count)


Week_Day <- subset6 %>%
  filter(weekday_end == "weekday") %>%
  pull(count)


TST_Test1 <- t.test(Week_Day,Week_end, conf.level = 0.95)
TST_Test1

OST_Test2 <- t.test(Week_Day,Week_end, conf.level = 0.95, alternative = "greater")
OST_Test2
```
```

Image 13: Hypotheses Testing

```
        Welch Two Sample t-test

data:  Week_Day and Week_end
t = -0.42379, df = 404, p-value = 0.6719
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.9054565  0.5843027
sample estimates:
mean of x mean of y
 20.50769  20.66827
```

Image 14: Two-Sided T Test Conclusion

<u>Two-sided T Test Conclusion:</u>

From the Welch Test, we fail to reject the null hypothesis since the p-value is 0.6719 against the alpha value of 0.05.

To revalidate the result, we performed a 1-sided T-Test to check which visits were greater. The results of the same are as below:

```
        Welch Two Sample t-test

data:  Week_Day and Week_end
t = -0.42379, df = 404, p-value = 0.664
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -0.7852593        Inf
sample estimates:
mean of x mean of y
 20.50769  20.66827
```

Image 15: One-Sided T Test Conclusion

One-Sided T_Test Conclusion:

Here, our Ho is that the visits on Weekdays is lesser than the visits on weekends. Based on the p-value, which is higher than 0.05, we fail to reject the null hypothesis. Thus, we can say that the number of visits on Weekdays is lesser than the number of visits on Weekends.

3.3 Market Basket Analysis

3.3.1 Initial Analysis

To conduct Market Basket Analysis (MBA), the list of items (itemList) created before was converted to a basket format using the library "arules".

```
# Convert to Basket Format
library(arules)
basket = read.transactions(file="ItemList.csv", rm.duplicates= TRUE,
format="basket",sep=",",cols=1);
print(basket)
```

```
distribution of transactions with duplicates:
items
  1   2   3   4
662  39   5   1
transactions in sparse format with
 14964 transactions (rows) and
 168 items (columns)
```

Image 16: MBA Initial Analysis

```
itemFrequencyPlot(basket, topN = 160, type = 'absolute', ylim = c(0,3000), col = rainbow(5))
```

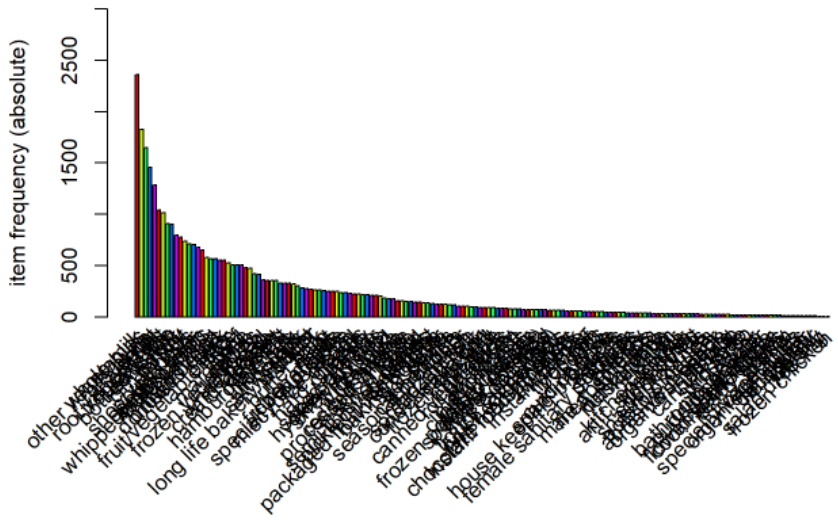

Image 17: Visualization of MBA Initial Analysis

As seen before, the items vary in the frequency they are bought. To visualize it better, we have included a a graph of what the buying patterns of individual items looks like and the same can be seen in Image 17.

```r
```{r}
library(ggplot2)
occurences <- data.frame(table(sorted_grocery$itemDescription))
colnames(occurences) <- c("itemDescription", "Frequency")
a <- occurences[rev(order(occurences$Freq, decreasing = TRUE)),]
ggplot(occurences[tail(order(occurences$Frequency), 7), ], ) +
     aes(itemDescription, Frequency) + geom_point() +
     scale_shape_manual(values=c(5,5)) +
     scale_color_manual(values=c("blue", "red"))
b <- occurences[rev(order(occurences$Freq, decreasing = FALSE)),]
a
b
summary(sorted_grocery)
```
```

Image 18: Code for MBA Initial Analysis

From the primary analysis we found that the top 10 most sold items were: whole milk, other vegetables, rolls/buns, soda, yogurt, root vegetables, tropical fruit, bottled water, sausage, and citrus fruit. While the least 10 sold products were: preservation products, kitchen utensils, baby cosmetics, bags, toilet cleaners, rubbing alcohol, make up remover, frozen chicken, salad dressing and whisky.

```r
```{r}
products_bought_summary<-grocery %>%
  group_by(Member_number,Date_corrected) %>%
  dplyr::summarise(prods_bought=n_distinct(itemDescription)) %>%
  arrange(desc(prods_bought)) %>%
  ungroup() %>%
  group_by(Member_number) %>%
  dplyr::summarise(avg_prods_bought=mean(prods_bought))

days_visit_summary<-grocery %>%
  group_by(Member_number) %>%
  dplyr::summarise(days_visited=n_distinct(Date_corrected)) %>%
  arrange(desc(days_visited)) %>%
left_join(products_bought_summary,by="Member_number")

summary(days_visit_summary)
# delete unwanted tables
#rm(list=c("test","products_bought_summary"))
```
```

```
`summarise()` regrouping output by 'Member_number' (override with `.groups` argument)
`summarise()` ungrouping output (override with `.groups` argument)
`summarise()` ungrouping output (override with `.groups` argument)
 Member_number   days_visited    avg_prods_bought
 Min.   :1000   Min.   : 1.000   Min.   :1.00
 1st Qu.:1999   1st Qu.: 2.000   1st Qu.:2.00
 Median :3004   Median : 4.000   Median :2.40
 Mean   :3003   Mean   : 3.839   Mean   :2.54
 3rd Qu.:4003   3rd Qu.: 5.000   3rd Qu.:2.80
 Max.   :5000   Max.   :11.000   Max.   :9.00
```

Image 19: Code for MBA

One of the main steps in the market basket analysis was to see what the average number of days a certain customer visited the store was and how many items he or she purchased. From the data it was clear that majority of the customers visited the store multiple times over the period of 728 days. The interesting insight into this was that the customers who visited the store multiple times ended up buying less products per visit on average, but the customers who visited the store only a few (1 or 2 times) throughout the period ended up purchasing products which were more than the normal average. This can be clearly seen in the data screenshot presented below. One of the major insights gained from this is the fact that the average number of transactions per customer is 2.54.

	Member_number	days_visited	avg_prods_bought
1	1379	11	2.545455
2	2193	11	2.454545
3	2271	11	2.545455
4	3737	11	3.000000
5	4338	11	2.545455
6	1052	10	2.600000
7	1275	10	2.400000
8	1410	10	2.600000
9	1574	10	2.300000
10	1793	10	2.400000

3886	4918	1	5
3887	4926	1	2
3888	4928	1	3
3889	4930	1	3
3890	4945	1	2
3891	4949	1	2
3892	4961	1	2
3893	4973	1	3
3894	4978	1	2
3895	4980	1	2
3896	4982	1	3
3897	4994	1	2
3898	4998	1	2

Image 20: Resulting Dataset

The main motivation behind conducting market basket analysis was the fact that it will enable us to understand which combinations of product to apply price promotions on to boost sales.

For this, it was clear that we had to either 2 product combos or 3 product combos since the average number of products bought by customers during each visit were 2.54. To support this decision, we decided to run “APRIORI” algorithm on both 2 product baskets and 3 product baskets and compare their maximum confidence levels.

```r
top10 <- c("whole milk","other vegetables","rolls/buns","soda","yogurt",
           "root vegetables","tropical fruit","bottled water","sausage",
           "citrus fruit")

basketrules1 <- apriori(basket, parameter = list(minlen=2, maxlen=2, sup = 0.001, conf =
0.01, target="rules"))

results_1=NULL

for (i in top10){
  rules.sub <- subset(basketrules1, subset = lhs %ain% i)
  test<-as.data.frame(inspect(rules.sub))
  results_1 = rbind(results_1,test)
}

results_1 <- results_1 %>%
  select(-"") %>%
  arrange(lhs,desc(confidence))

results_1 %>%
  arrange(desc(confidence))

summary(basketrules1)
```

Image 21: Code for MBA

We decided to consider "support" of the products sold from the itemList. The top 10 products with the highest support, which are the top 10 most sold products, were then sorted through the basket "itemList" (created before) to figure out the best combinations.

The APRIORI algorithm runs through the basket and gives out combinations that contain one of the items from the top10 list with a support of greater than 0.001 and a confidence level greater than 0.01.

We chose to have a minimum support of 0.001 because the total number of lists created in the basket are 14963. A support level of 0.001 means that these combinations were present om over 14.96 or 15 of those lists, which is a very conservative number. That combined with a confidence level of 0.01 creates a perfect filter for choosing our combinations.

A minimum length (minlen) and a maximum length (maxlen) of 2 means that we are looking for combinations of only 2 products.

```
summary of quality measures:
    support           confidence          coverage             lift
 Min.   :0.001002   Min.   :0.01032   Min.   :0.006816   Min.   :0.3752
 1st Qu.:0.001270   1st Qu.:0.02570   1st Qu.:0.031910   1st Qu.:0.7388
 Median :0.001604   Median :0.04005   Median :0.046913   Median :0.8230
 Mean   :0.002210   Mean   :0.04986   Mean   :0.056882   Mean   :0.8561
 3rd Qu.:0.002473   3rd Qu.:0.06737   3rd Qu.:0.069567   3rd Qu.:0.9411
 Max.   :0.014836   Max.   :0.17606   Max.   :0.157912   Max.   :1.6539
     count
 Min.   : 15.00
 1st Qu.: 19.00
 Median : 24.00
 Mean   : 33.07
 3rd Qu.: 37.00
 Max.   :222.00
```

Image 22: Summary for MBA

As seen from the output above, there are 1157 total 1-1 product combinations that contain one of the top 10 best selling products. The maximum confidence level here is 0.17606. A sample output of the top 10 rules by confidence is shown below:

```r
```{r}
library(arulesViz)
subRules1 = basketrules1[quality(basketrules1)$confidence > 0.07147]
top10RulesByConfidence = head(subRules1, n = 10, by = "confidence")
inspect(top10RulesByConfidence)
top10RulesByConfidence
plot(top10RulesByConfidence, method = "graph", engine = "htmlwidget")
plot(basketrules1[1:10], method = "paracoord")
plot(basketrules1, jitter = 0)
```
```

Image 23: Code for MBA Visualization

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>
[1]	{semi-finished bread}	=>	{whole milk}	0.001670676	0.1760563
[2]	{detergent}	=>	{whole milk}	0.001403368	0.1627907
[3]	{ham}	=>	{whole milk}	0.002739909	0.1601562
[4]	{bottled beer}	=>	{whole milk}	0.007150495	0.1578171
[5]	{frozen fish}	=>	{whole milk}	0.001069233	0.1568627
[6]	{candy}	=>	{whole milk}	0.002138466	0.1488372
[7]	{sausage}	=>	{whole milk}	0.008954825	0.1483942
[8]	{onions}	=>	{whole milk}	0.002940390	0.1452145
[9]	{processed cheese}	=>	{rolls/buns}	0.001470195	0.1447368
[10]	{processed cheese}	=>	{whole milk}	0.001470195	0.1447368

Image 24: MBA Visualization

The following few diagrams show scatterplot of the 1159 rules along with a parallel coordinates plot for the top 10 rules. Also pictured is an interactive html widget which displays the relationships of the 10 combinations which have the highest confidence amongst them.

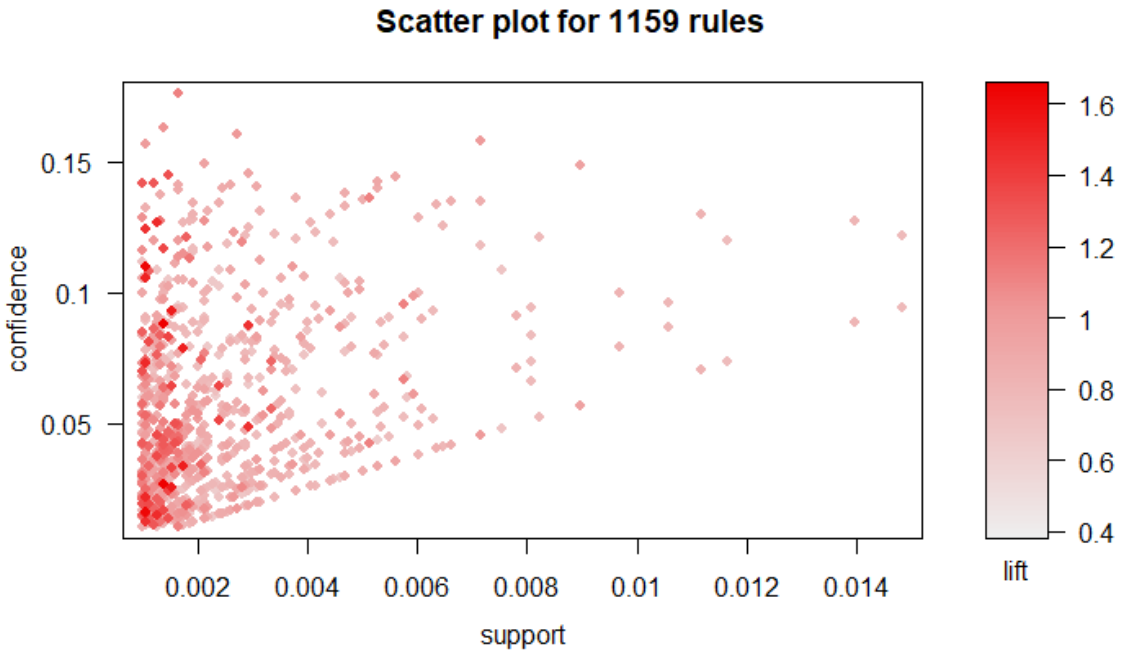

Image 25: MBA Scatter Plot

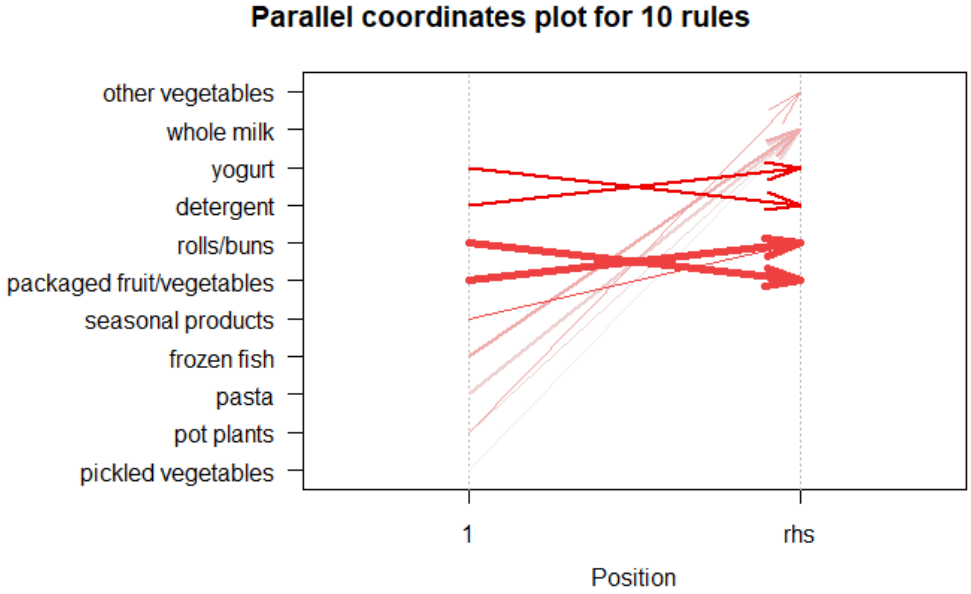

Image 22: Parallel Coordinates Plot

The following interactive widget allows us to individually inspect all the relationships not only by rules, but also by the individual products.

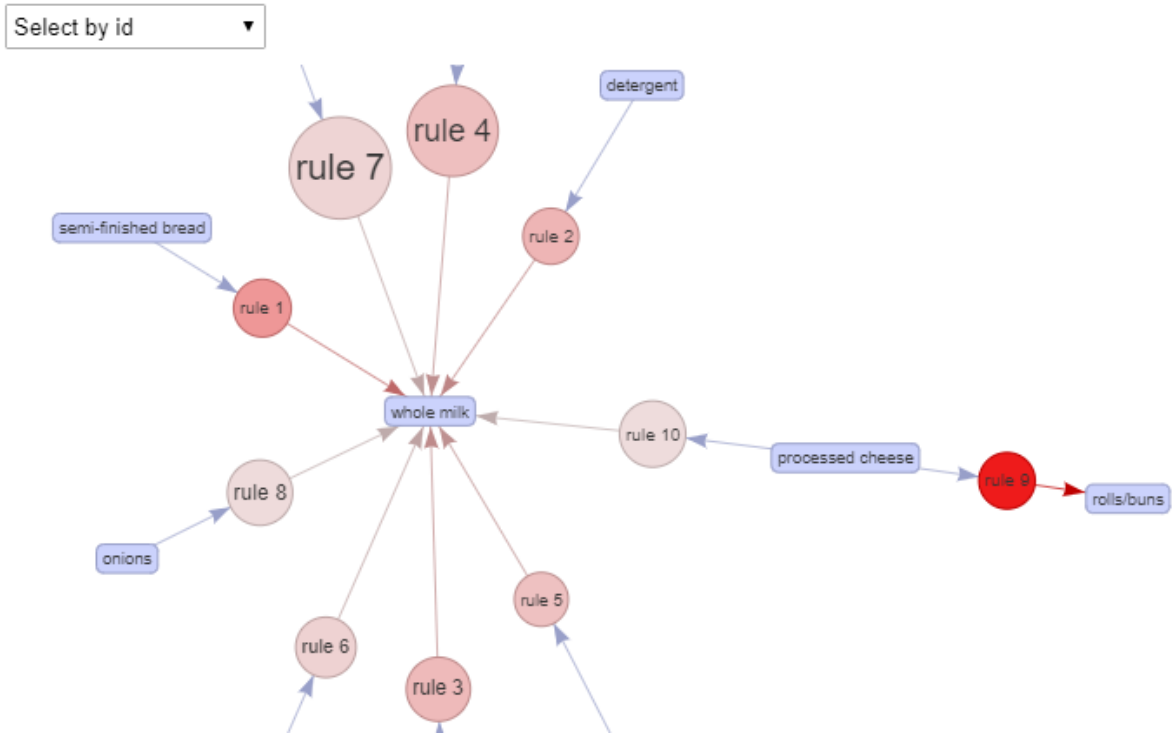

Image 23: Interactive Widget

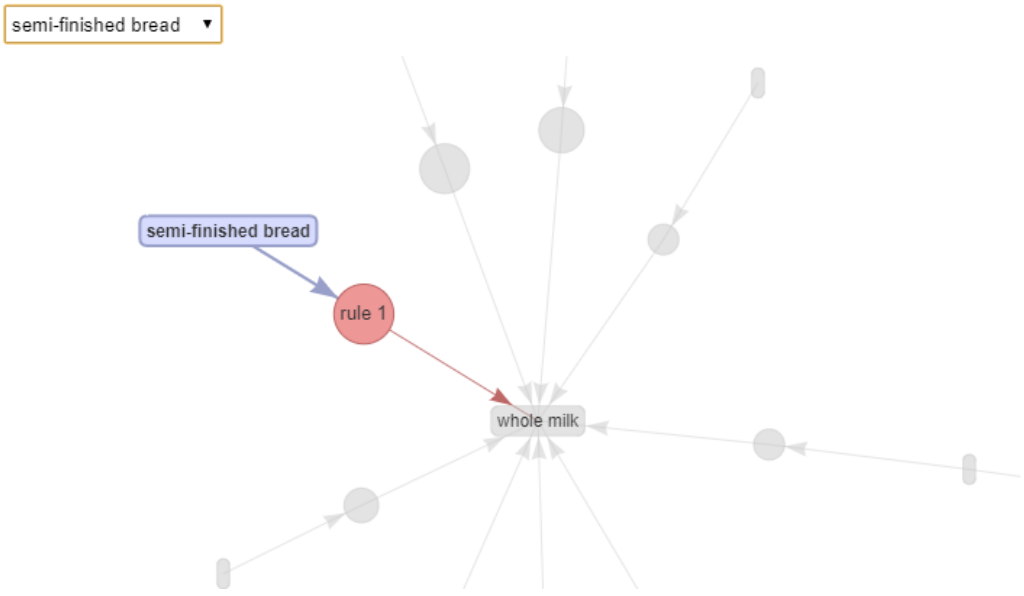

Image 24: Interactive Widget

Similar to the analysis with 1-1 i.e. 2 product combinations, we decided to analyze the confidence of 2-1 i.e. 3 product combinations for the top 10 most sold products. Same code along with the same confidence and support levels was used for this. The only thing different was the minimum length and maximum length which were both set at 3.

Here only 27 rules were discovered but the maximum confidence level saw a dramatic increase to 0.25581 from the previous 0.1760563. A summary of these rules along with the top 10 combinations is shown below:

```
summary of quality measures:
    support           confidence          coverage             lift
 Min.   :0.001002   Min.   :0.07177   Min.   :0.005346   Min.   :0.7054
 1st Qu.:0.001136   1st Qu.:0.08908   1st Qu.:0.008520   1st Qu.:0.7868
 Median :0.001136   Median :0.11724   Median :0.010559   Median :1.0825
 Mean   :0.001181   Mean   :0.12181   Mean   :0.010564   Mean   :1.0919
 3rd Qu.:0.001203   3rd Qu.:0.13612   3rd Qu.:0.012797   3rd Qu.:1.1915
 Max.   :0.001470   Max.   :0.25581   Max.   :0.014836   Max.   :2.1831
     count
 Min.   :15.00
 1st Qu.:17.00
 Median :17.00
 Mean   :17.67
 3rd Qu.:18.00
 Max.   :22.00
```

Image 25: Summary of Quality Measures

	lhs <chr>	<chr>	rhs <chr>	support <dbl>
[1]	{sausage,yogurt}	=>	{whole milk}	0.001470195
[2]	{rolls/buns,sausage}	=>	{whole milk}	0.001136060
[3]	{sausage,soda}	=>	{whole milk}	0.001069233
[4]	{rolls/buns,yogurt}	=>	{whole milk}	0.001336541
[5]	{sausage,whole milk}	=>	{yogurt}	0.001470195
[6]	{other vegetables,yogurt}	=>	{whole milk}	0.001136060
[7]	{rolls/buns,soda}	=>	{other vegetables}	0.001136060

Image 26: Resulting Visualization

The lift observed in these is also higher at a maximum of 2.1831. This clearly goes to show that it is better to give combinations that have 3 products instead of 2 products. Both these analyses prove our initial decision of giving pricing promotions of 3 products (which is higher than the average products bought by an individual customer).

Going forward, our first task is to identify which product will make the best combination with whole milk, other vegetables, and rolls/buns since they are the ones with the highest support. Once we have that we will find the product that is the least associated with whole mile, other vegetables, and rolls/buns.

We decided to use 'Lift' parameter because it tells us the strength of association between the two products and does not depend on the directionality of picking the products. On the other side, we decided not to use the 'confidence' parameter because it tells us about the conditional association between the products. Therefore, using 'Lift' parameter makes it easier for us to determine the product combinations.

3.3.2 Product Combinations

According to our market research we know that people are most confident with combinations of 3 items in a pack, including more than 3 items make buyers doubt the combination offer and affects the overall sales. Hence, we have proposed the following three combinations that the store should offer pricing promotions on.

- **COMBO 1: Whole Milk + Semi-Finished Bread + Buttermilk**

We have already identified the top selling product in our dataset i.e., Whole milk and decided to pair it up with two other products. We do this by finding the lift value by conducting APRIORI analysis as shown in the code chunk further below. We decided to pair Whole milk with the product with highest value of lift and with the product with lowest value of lift with respect to whole milk to boost up the sales of the retail store in total.

To start off, we find a product that has the highest lift with respect to whole milk. This is shown in the code chunk below. We keep the parameters of support and confidence level the same as before (supp = 0.001, conf = 0.01). Keeping minlen and maxlen equal to 2, we limit the combinations to only two products and sort them according to decreasing lift.

```r
top1 <- c("whole milk")

basketrules3 <- apriori(basket, parameter = list(minlen=2, maxlen=2, sup = 0.001, conf = 0.01, target="rules"))
summary(basketrules3)

results_3=NULL

for (i in top1){
  rules.sub3 <- subset(basketrules3, subset = lhs %ain% i)
  test3<-as.data.frame(inspect(rules.sub3))
  results_3 = rbind(results_3,test3)
}

results_3 <- results_3 %>%
  select(-"") %>%
  arrange(lhs,desc(lift))


results_3 %>%
  arrange(desc(lift))

results_3
```

lhs <chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
{whole milk}	{semi-finished bread}	0.001670676	0.01057977	0.1579123	1.1148993
{whole milk}	{butter milk}	0.001670676	0.01057977	0.1579123	0.6019608

Image 27: Combo 1 – Code + Result

Referring to the analysis, product with highest value of lift in respect to whole milk is Semi-finished bread [lift of 1.1148993] and the product with lowest value of lift in respect to whole milk is buttermilk [lift of 0.6019608] .

- **<u>COMBO 2: Other Vegetables + Frankfurter + Pastry</u>**

We have already identified the second most selling product in our dataset i.e., Other Vegetables and decided to pair it up with two other products. We do this by finding the lift value by conducting APRIORI analysis like we did for COMBO 1. All the parameters in the code remain the same, except we change the product in our loop from "Whole Milk" to "Other Vegetables". A summary of the analysis is shown below.

lhs <chr>	<chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
{other vegetables}	=>	{frankfurter}	0.005145683	0.04214559	0.122093	1.1162242
{other vegetables}	=>	{pastry}	0.003675488	0.03010400	0.122093	0.5820106

Image 28: Combo 2 – Result

Referring to the analysis, product with highest value of lift in respect to other vegetables is frankfurter [lift of 1.1162242] and the product with lowest value of lift in respect to other vegetables is pastry [lift of 0.5820106].

- **COMBO 3: Rolls/buns + Processed Cheese + Beef**

Our third most selling product is rolls/buns. We have to pair it up with two other products. Just like the previous combinations, we do this by conducting APRIORI analysis. Again, all the parameters in the code remain the same, except we change the product in our loop from "Whole Milk" to "rolls/buns". A summary of the analysis is shown below.

lhs <chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
{rolls/buns}	{processed cheese}	0.001470195	0.01336574	0.1099973	1.3158214
{rolls/buns}	{beef}	0.001603849	0.01458080	0.1099973	0.4295022

Image 29: Combo 3 – Result

Referring to the analysis, product with highest value of lift in respect to rolls/buns is processed cheese [lift of 1.3158214] and the product with lowest value of lift in respect to rolls/buns is beef [lift of 0.4295022].

3.4 Product Placements in the store

In this section, we have used confidence parameter to identify the best product placements to increase the sales in store because confidence parameter determines the directionality of the products in addition to the strong association between the products.

- WHOLE MILK should be placed in one corner of the store. This will help to ensure maximum travel of the customers across the store.
- To ensure that the store has maximum utilization of the available shelf-space to drive increase in customer purchase, we recommend the store to place SODA & SAUSAGE around WHOLE MILK. Our preference would be to place these 3 products in the same aisle or as close to each other as possible.

- To increase the sales of CANNED FRUITS & SALAD DRESSING, we recommend keeping them in the aisles between WHOLE MILK and OTHER VEGETABLES. As these are the top selling products and have the chance of maximum footfall, there is high chances that it will boost up the sales of the products placed between them.
- Customers should be able to buy maximum items in one walk around the store. For this, we have figured the top combinations of the products based on their confidence. SAUSAGES and YOGURT should be placed around WHOLE MILK. ROLLS / BUNS could be kept in the same region close to OTHER VEGETABLES. SODA should be kept in a shelf between WHOLE MILK, SAUSAGES & ROLLS/BUNS.

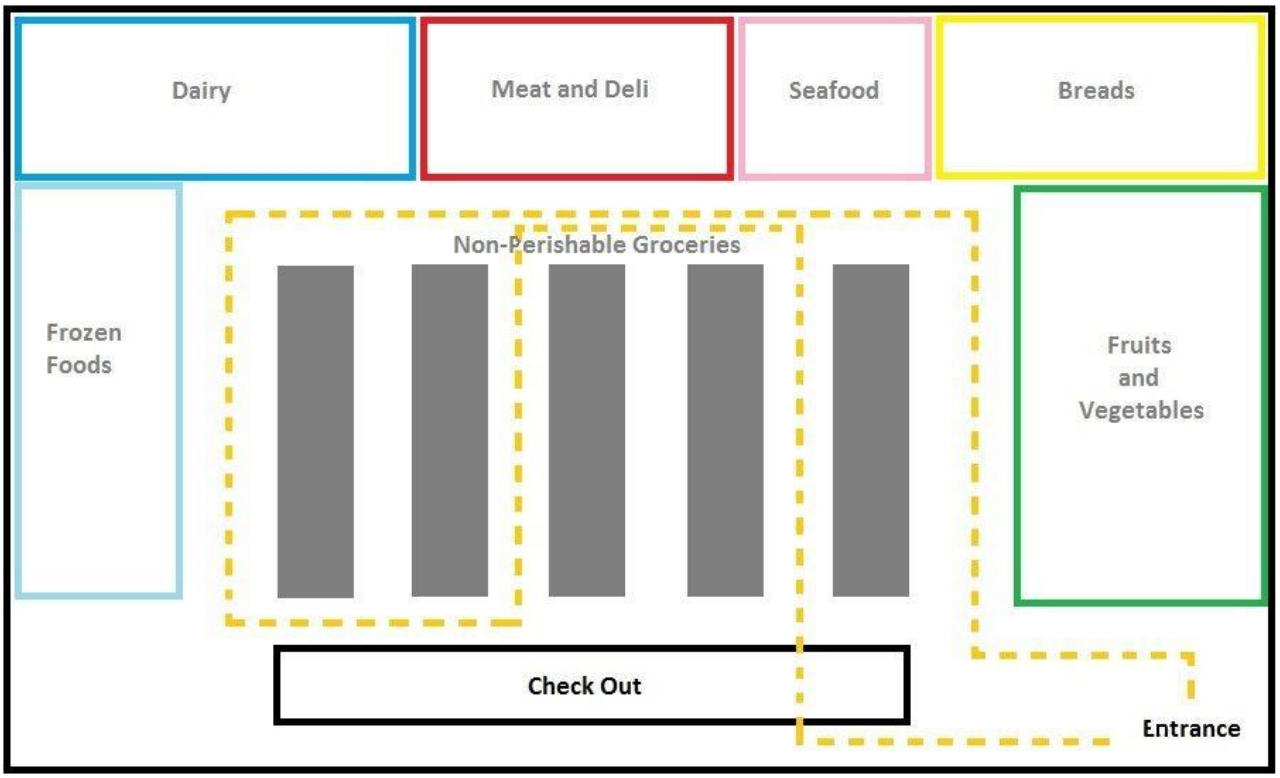

Image 30: Suggested Product Placement Layout

3.5 Membership Benefits

As seen before, we analyzed customer buying trends over the entire period of 728 days by weekday/ weekend and by month. There wasn't a significant difference between the frequencies of visits by weekday and visits per month. Hence, we decided to go a step further and analyze which items were bought by our most frequent customers.

```
 Member_number   days_visited    avg_prods_bought
 Min.   :1000   Min.   : 1.000   Min.   :1.00
 1st Qu.:1999   1st Qu.: 2.000   1st Qu.:2.00
 Median :3004   Median : 4.000   Median :2.40
 Mean   :3003   Mean   : 3.839   Mean   :2.54
 3rd Qu.:4003   3rd Qu.: 5.000   3rd Qu.:2.80
 Max.   :5000   Max.   :11.000   Max.   :9.00
```

Image 31: Membership Analysis

This summary shows (refer to page#) that the highest number of times a customer visited the store was 11 with an average of 3.839 visits. It also shows that the average number of products bought were 2.54 and the maximum number of products purchased by a customer during a single visit was 9.

In the example below, we have considered customer number 1379 who visited the store 11 times over the period of 728 days. This customer purchased a total of 26 items over those 11 visits. From the picture below it is clear that there is no significant difference between the most and least bought products. This remains same for almost all other customers.

```r
df1 <- grocery[with(grocery, Member_number == 1379),]
df1
#agg1 <- aggregate(df1$Member_number, by=list(df1$itemDescription), FUN=sum)
#agg1
factor(df1$itemDescription)
items1 <- table(df1$itemDescription)
i1379 <- as.data.frame(items1)
i1379[order(-i1379$Var1, i1379$Freq),]
```

	Var1 <fctr>	Freq <int>
22	pork	1
23	rolls/buns	1
24	root vegetables	1
25	white bread	1
17	other vegetables	2
26	whole milk	2

Image 29: Membership Analysis – Code + Result

Hence, in order to offer membership for customers, we decided to offer point-based membership benefits to customers who were the 75th percentile of the total number of visits per customers, which according to the days_visit_summary is every customer who visited the store 5 or more than 5 times in 728 days. This comes out to a total of 730 customers out of 3898 members. This seems reasonable as we are recommending that the store reward

its most frequent customers. This point-based membership will help the customers earn points based on their future purchases. They will be able to spend these points by applying them towards a price discount in the future and thus ensure their return to the store.

4 RECOMMENDATIONS

- We recommend the store to build a product combo of Whole Milk, Semi-Finished Bread and Buttermilk so that the sales of Whole Milk can be leveraged to enhance the sales of Semi-Finished Bread & Buttermilk.
- We recommend the store to build a product combo of Other vegetables, frankfurter, and pastry so that the sales of other vegetables can be leveraged to enhance the sales of frankfurter & pastry.
- We recommend the store to build a product combo of rolls/buns, processed cheese, and beef so that the sales of Rolls/buns can be leveraged to enhance the sales of Processed cheese & Beef.
- WHOLE MILK should be placed at one corner of the store.
- We recommend the store to place SODA & SAUSAGE around WHOLE MILK.
- To increase the sales of CANNED FRUITS & SALAD DRESSING, we recommend keeping them in the aisles between WHOLE MILK and OTHER VEGETABLES.
- SAUSAGES and YOGURT should be placed around WHOLE MILK. ROLLS / BUNS could be kept in the same region close to OTHER VEGETABLES. SODA should be kept in a shelf between WHOLE MILK, SAUSAGES & ROLLS/BUNS.
- WHOLE MILK, OTHER VEGETABLES & ROLLS / BUNS, are products whose cost can be increased since they are the maximum selling products at the store.
- The store should negotiate better deals for the top 10 selling products with its product vendors.
- NEW PRODUCT LAUNCHES should be done on Weekends since footfall over weekends is higher than the footfall over weekdays.
- We recommend the store to start Premium membership club for the most frequent/valuable customers, where their purchases help them earn points for discounts on future shopping.

5 REFERENCES

- Kaggle dataset : https://www.kaggle.com/heeraldedhia/groceries-dataset
- Supporting websites: https://techbusinessguide.com/what-is-market-basket-analysis/
- Classroom Exercises by Professor Dan Yang