Homework3

Group 10

11/2/2019

The Business Problem and Our Approach

Defining the business problem

Central Perk, a coffee shop in New York City, believes that their current customer base is fairly loyal. The coffee shop has never looked at their data and do not have a good sense of their demand. They wish to learn about the general volume distribution and types of items being ordered which could help them understand the current customers' buying patterns. There are certain periods and products where the store sales have been below par. We are here to help Central Perk find unique ways to improve revenue and customer engagement for the impaired periods by leveraging their historical sales data.

Key-Question:

What strategies can we adopt to improve the net sales for the cafe during its underperforming weeks?

Sub-Question

In order to answer the key question we broke it down into three parts:

1. Who are our customers and when do they visit?
2. What are the general purchasing patterns of customers?
3. When do we not perform as well as we expect to?

Approach

We get to our recommendations by solving three parts:

1. To understand more about our customer base we clustered our visitors based on the how often they visit the cafe (frequency), when was the last time they visited (recency) and the total amount they spent in the cafe (monetary amount spent). This RFM analysis would help us understand our customer base annd identify our most loyal customers. Examining the transactions of customers gets us this result.
2. The next step lies in exploration of the transaction data to identify general trends in item sales. We analysed the underlying distribution of the items being sold across different time periods (hour of the day, day of the week and months in a year), we broke this down for the customer segments we arrived from our RFM model. Furthermore, we used the apriori association rule algorithm to understand the lift in net sales for the items being sold at different time periods (hours and days) and the lift associated with different products being bought together.
3. The last part of the dataset lies in identifying when we dont perform as well as we expect to. We identified these anamolous weeks by decomposing the net sales across weeks into short-term and long-term trends. Subtracting the trends with the actual data would give us a seasonally adjusted time series which can be inspected to understand weeks we dont perform well.

Combining the results from the three parts described above enabled us to come up with promotional schemes and specialised recommendations.

Data Preparation

Load Libraries

```
library(rmarkdown)
library(magrittr)
library(plyr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(gridExtra)
library(data.table)
library(stringr)
library(lubridate)
library(arules)
library(rfm)
library(tidyverse)
library("xts")
library(fpp)
```

Read and Merge Data

```
path = "~/Desktop/MSBA1/Exploratory Data Analytics & Visualization/HW 3/Central Perk"
filenames <- list.files(path = path, pattern = "*", full.names=TRUE)
central <- plyr::ldply(filenames, read.csv)
```

Data Cleaning

To prepare the data for our following analysis, we need to deal with abnormal and missing values.

```
clean <- central %>%
# Removing the refund related transactions to eliminate ambiguity
  filter(!(Event.Type == "Refund")) %>%
# Removing items with no quantity value populated
filter(!is.na(Qty)) %>%
# Cleaning up the date field to retain expected values
  filter(!(Date == 'Unknown Error')) %>%
# Keeping only the relevant item types which can inform our analysis
mutate(Item = as.character(Item)) %>%
  filter(Item != 'Custom Amount' & Category != 'None') %>%
# Processing the item types to contain palatable characters
  mutate(Item = str_trim(str_replace(Item, "SM$", ""))) %>%
  mutate(Item = str_trim(str_replace(Item, "LG$", ""))) %>%
  mutate(Item = str_replace(Item, "<LemonadeðY\u008d<", "Lemonade")) %>%
# Obtaining one unified time field & transforming the constituent time fields
  mutate(Timestamp = as.POSIXct(paste(Date, Time), format="%m/%d/%y %H:%M:%S"),
         Hour = factor(hour(Timestamp)),
         Day = factor(day(Timestamp)),
         Month = factor(month(Timestamp)),
         Year = factor(year(Timestamp)),
         Weekday = factor(weekdays(Timestamp))) %>%
# Maintaining consistent field values for Price.Point.Name with the assumption that
```

```
#  the two values hold the same meaning
mutate(Price.Point.Name = str_replace(Price.Point.Name, "Regular Price", "Regular"))
# Cleaning up numeric fields to remove accounting style formatting
clean$Gross.Sales <- str_remove_all(clean$Gross.Sales, "[$]")
clean$Discounts <- str_remove_all(clean$Discounts, "[$]")
clean$Net.Sales <- str_remove_all(clean$Net.Sales, "[$]")
clean$Tax <- str_remove_all(clean$Tax, "[$]")

# Removing emoticons attached with lemonade item cells
clean[clean$Item %like% "Lemonade", ]$Item <- "Lemonade"

# Add Quarter Info
clean$quarter <- 0
clean[clean$Month %in% c(10,11,12),]$quarter <- 4
clean[clean$Month %in% c(1,2,3),]$quarter <- 1
clean[clean$Month %in% c(4,5,6),]$quarter <- 2
clean[clean$Month %in% c(7,8,9),]$quarter <- 3

clean$quarter <- as.factor(clean$quarter)
clean$Category <- as.factor(clean$Category)
clean$Item <- as.factor(clean$Item)
clean$Year <- as.factor(clean$Year)
write.csv(clean,'cleandata.csv')
```

1. Analysis on Current Customer Base

To understand what proportion of customers, we have performed RFM (recency, frequency, monetary) analysis on the customers with valid customer IDs. It is a behavior based technique which is used to segment customers by examining their transaction history such as how recently a customer has purchased, how often they purchase, how much the customer spends. This will help us to identify customers who are more likely to respond to promotions by segmenting them into different loyalty categories.

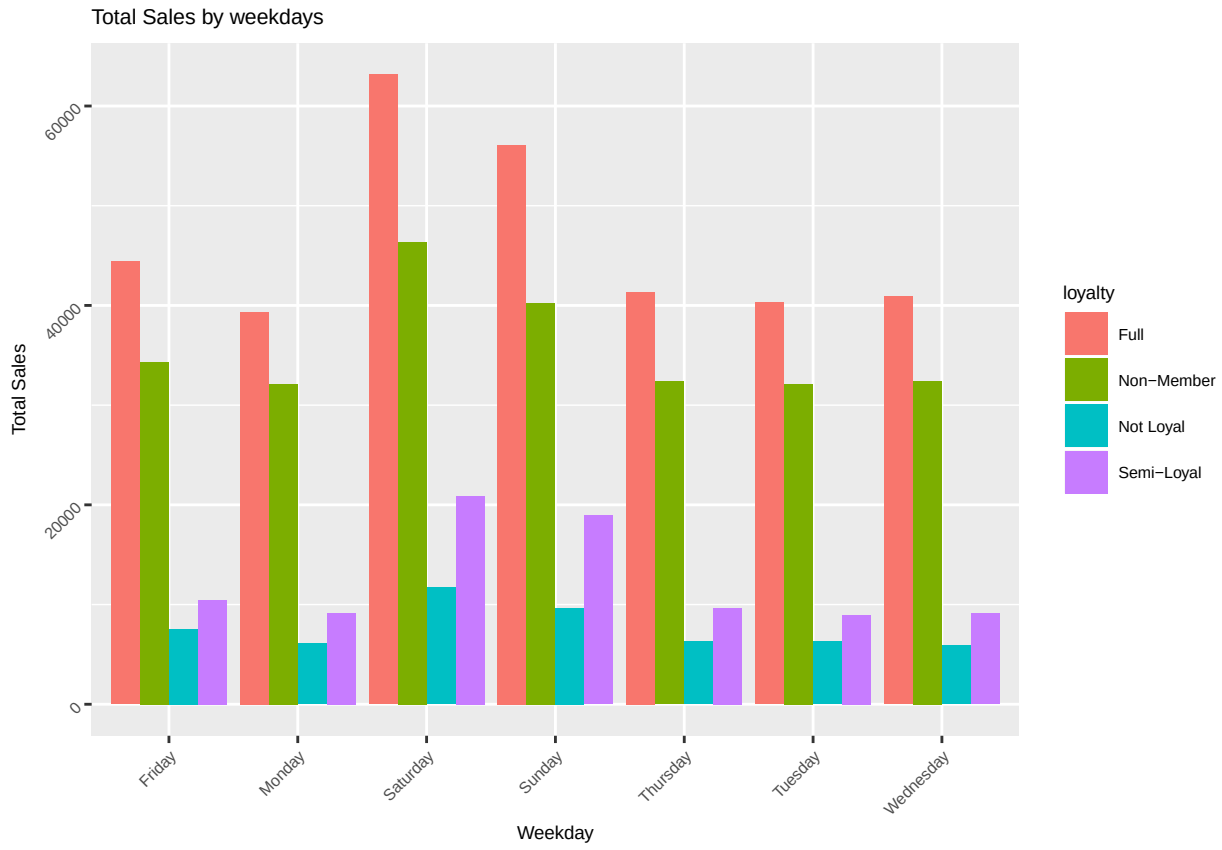
We obtain the following sub results to conlcude our loyalty analysis: a.) Recency score is assigned to each customer based on date of most recent purchase. b.) Customers with high purchase frequency are assigned a higher score compared to the low frequency customers. c.) Monetary score is assigned on the basis of the total revenue generated by the customer in the period under consideration for the analysis. d.) A RFM score is generated as an aggregated measure which is simply the three individual scores concatenated into a single value.

```
# Remove the customers with missing customer IDs
df2 <- df[!(is.na(df$Customer.ID)),]

# Keeping data only with valid timestamps & converting it to the appropriate form
df3 <- df2[df2$Year >0,]
df3$Timestamp1 <- as.character(df3$Timestamp)
df3$Timestamp2 <- lubridate::as_date(df3$Timestamp1, tz = "UTC")

# Setting an anchor date for recency analysis
analysis_date <- lubridate::as_date("2018-08-01", tz = "UTC")

# Running the final analysis
rfm_result <- rfm::rfm_table_order(df3, Customer.ID, Timestamp2, Net.Sales, analysis_date)
```

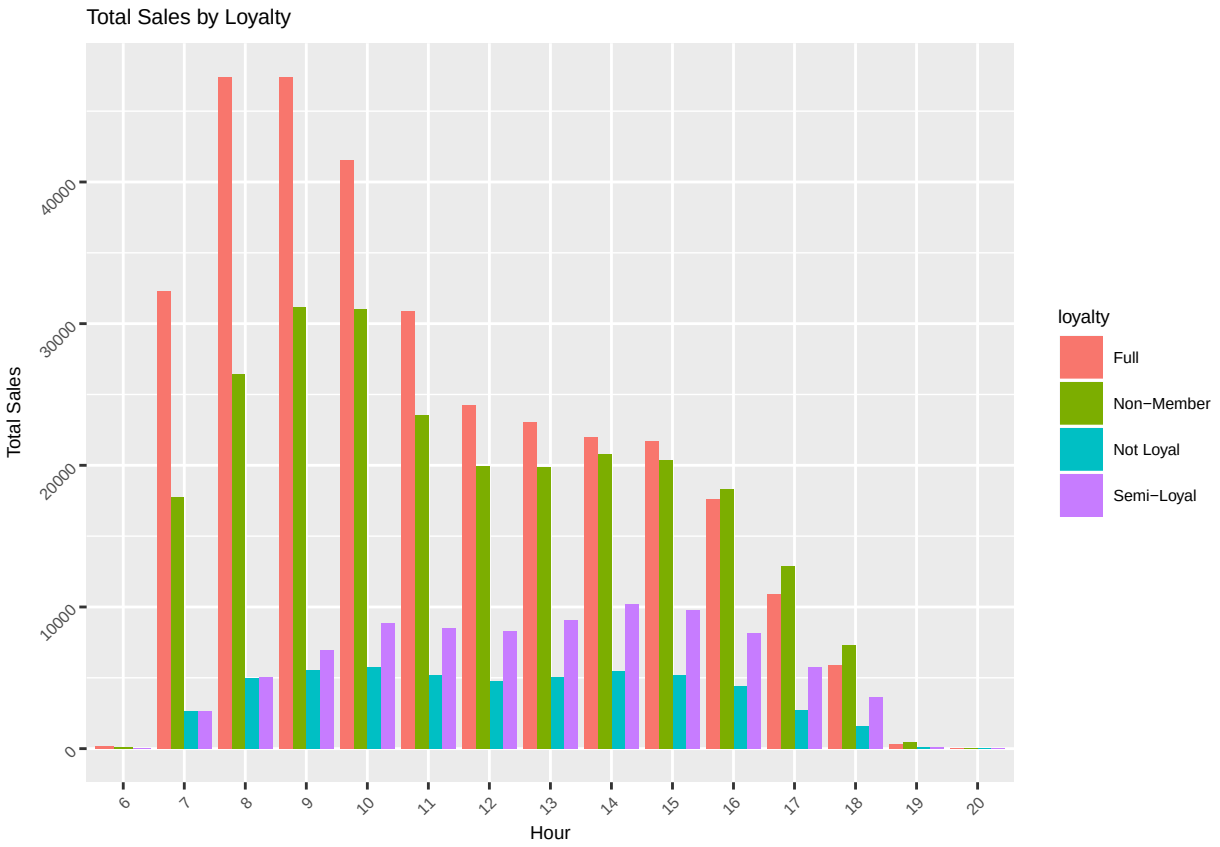
```r
write.table(rfm_result$rfm , "rfm.csv",sep = ",", row.names = FALSE)

x <- read.csv("rfm.csv")

# Categorizing the customer loyalties
x$loyalty <- ifelse (x$recency_score>=2 & x$frequency_score>=3 & x$monetary_score >=3,
                     x$loyalty <- "Full",
                     ifelse (x$recency_score>=2 & x$frequency_score>=1 &
                      x$monetary_score >=1,x$loyalty <- "Semi-Loyal",x$loyalty <- "Not Loyal"))

write.table(x , "rfm_cat.csv",sep = ",", row.names = FALSE)
```

Sales across different customer loyalty segments

```r
rfm<-read.csv("rfm_cat.csv")

data_1<-dplyr::full_join(data,rfm,by=c("Customer.ID"="customer_id"))
data_1$loyalty<-as.character(data_1$loyalty)
data_1<-data_1%>%mutate(loyalty=if_else(is.na(data_1$loyalty),'Non-Member',loyalty))
data$Time<-lubridate::hms(data$Time)
data<-data%>%mutate(membership=ifelse(is.na(data$Customer.ID),'Non-Member','Member'))

#######Total sales across loyalties
x<-data_1%>%select("loyalty","Net.Sales","Weekday")%>%
  dplyr::group_by(Weekday,loyalty)%>%
  dplyr::summarise(net_sales=sum(as.numeric(Net.Sales)))%>%
  ungroup()

ggplot(x, aes(x=as.factor(Weekday),y = net_sales,fill=loyalty)) +
  geom_bar(stat="identity",position = position_dodge()) +
  xlab("Weekday")+ylab("Total Sales")+
  theme(text = element_text(size = 7),
    axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Total Sales by weekdays")
```



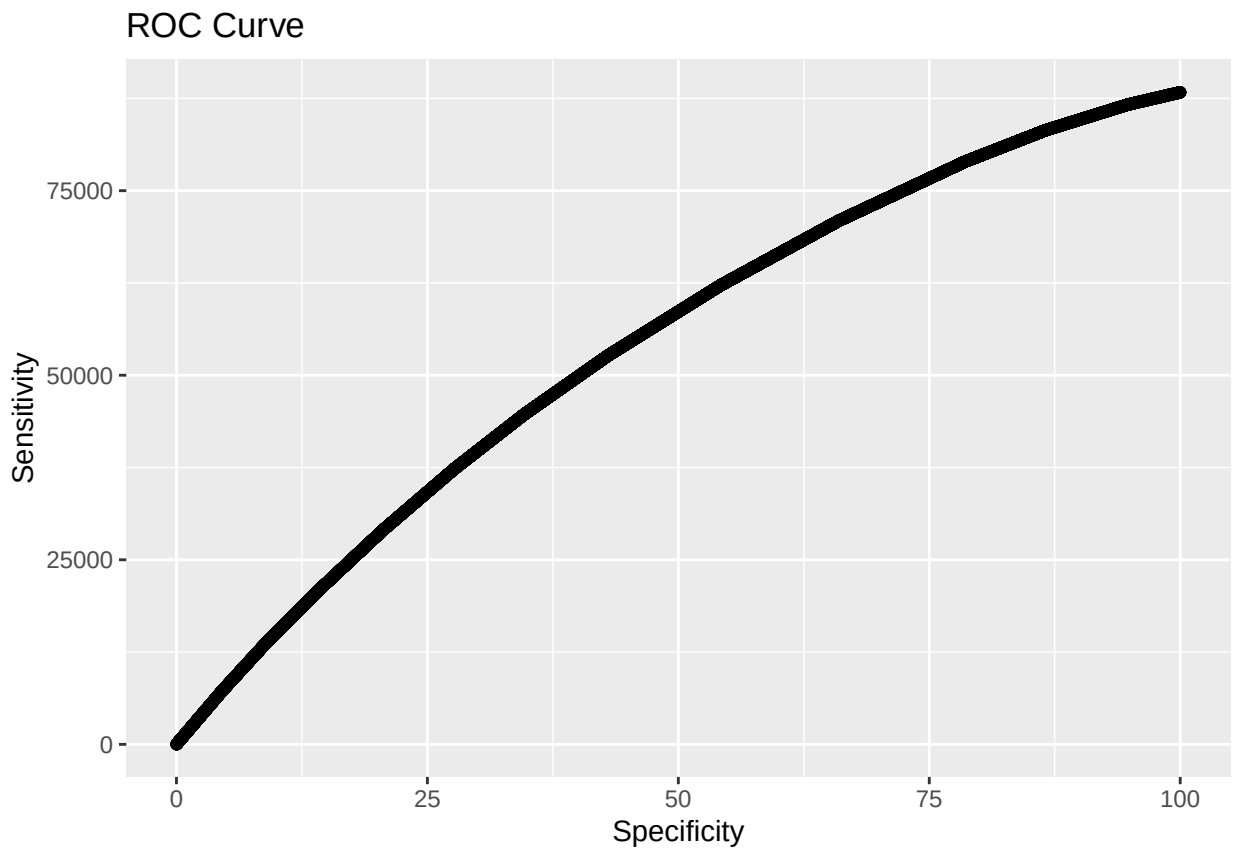

```
#######Total sales across loyalties by hour
x<-data_1%>%select("loyalty","Net.Sales","Hour")%>%
  group_by(Hour,loyalty)%>%
  dplyr::summarise(net_sales=sum(as.numeric(Net.Sales)))%>%
  ungroup()

ggplot(x, aes(x=as.factor(Hour),y = net_sales,fill=loyalty)) +
  geom_bar(stat="identity",position = position_dodge()) +
  xlab("Hour")+ylab("Total Sales")+
  theme(text = element_text(size = 7),
  axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Total Sales by Loyalty")
```



Cumulative Response curve for the RFM analysis

Plotting the cumulative response curve for the dataset, would give us an idea as to what percentage of the people are loyal. The cumulative repons curve shows the distribution of the customers on the x-axis as measured by specificity and the cumulative RFM score on the y-axis as measured by the sensitivity. The point with the slope changes the most would give us the percentage of customers who are loyal.

```
rfm_data$score <- (rfm_data$recency_score + rfm_data$frequency_score +
                     rfm_data$monetary_score )/3

subsetted_table <- rfm_data %>% select(customer_id, score) %>% arrange(desc(score))
subsetted_table$cumulative_score <- cumsum(subsetted_table$score)
row_count <- 1:nrow(subsetted_table)
row_count<- row_count/nrow(subsetted_table)*100

to_roc<-  as.data.frame(cbind(row_count,subsetted_table$cumulative_score))

ggplot(data=to_roc, aes(x= row_count, y=V2)) +
  geom_line()+
  geom_point() + labs(x = "Specificity", y = "Sensitivity" , title = "ROC Curve")
```



Assumptions Made For Further Analysis

By far, we have a better understanding for the customer behavior and buying patterns. We noticed that, customer buying trends over hours, days of weeks, months are comparatively stable. Most of the transcations and revenues are contributed by loyal cusomters. Further, we categorized customers with no customer IDs as non-membership customers, and non loyalty members. As we lack further information on the customer personal information, we could not dive further to for labeling. We have 51,477 such transactions where there are no customer id information (approximately 38%) during the 24 months.

2. Visual Exploration of trends in data

Analysis on Volume Distribution

To understand the demand of Central Perk's customers, we look into the volume distribution of all categories across quarter, month, week, and hour.

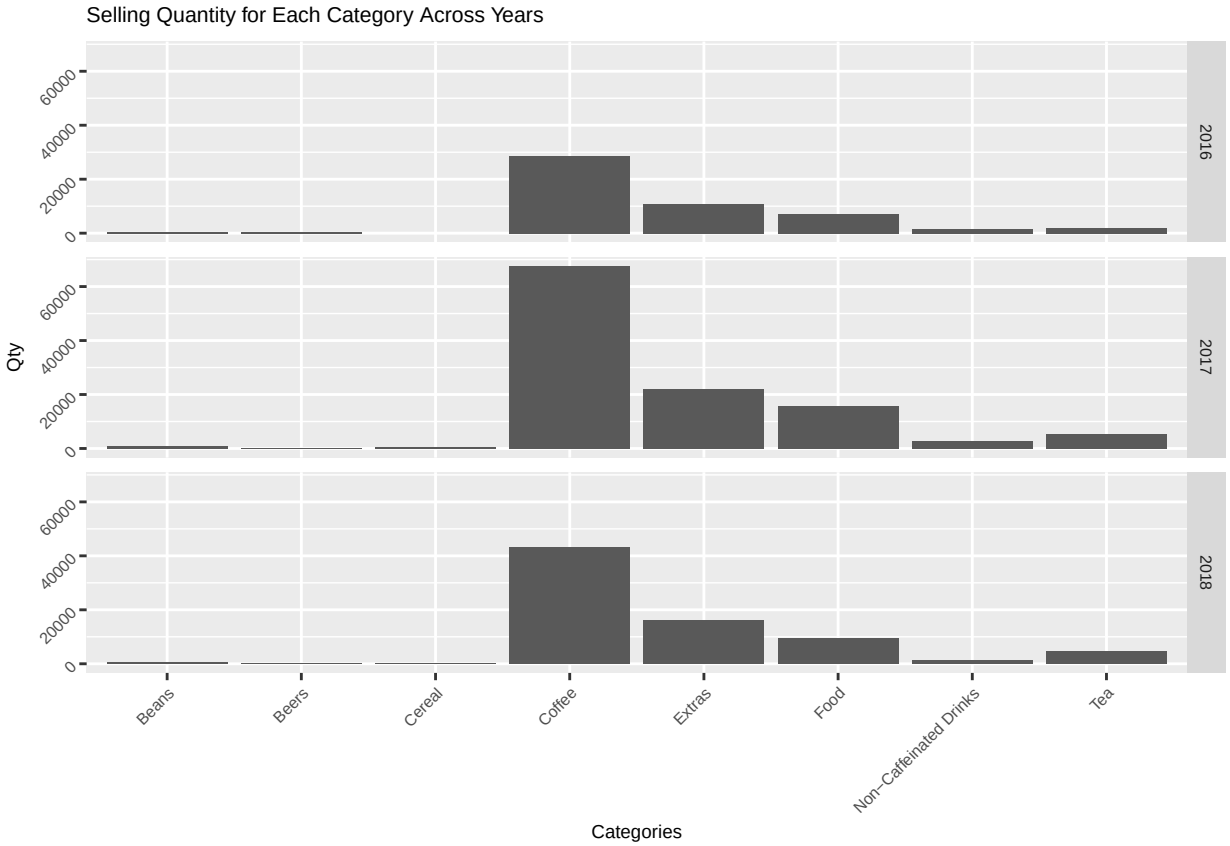
Distribution of amount of products sold and net sales across years

```
## Warning: Column `Customer.ID`/`customer_id` joining factors with different
## levels, coercing to character vector
```

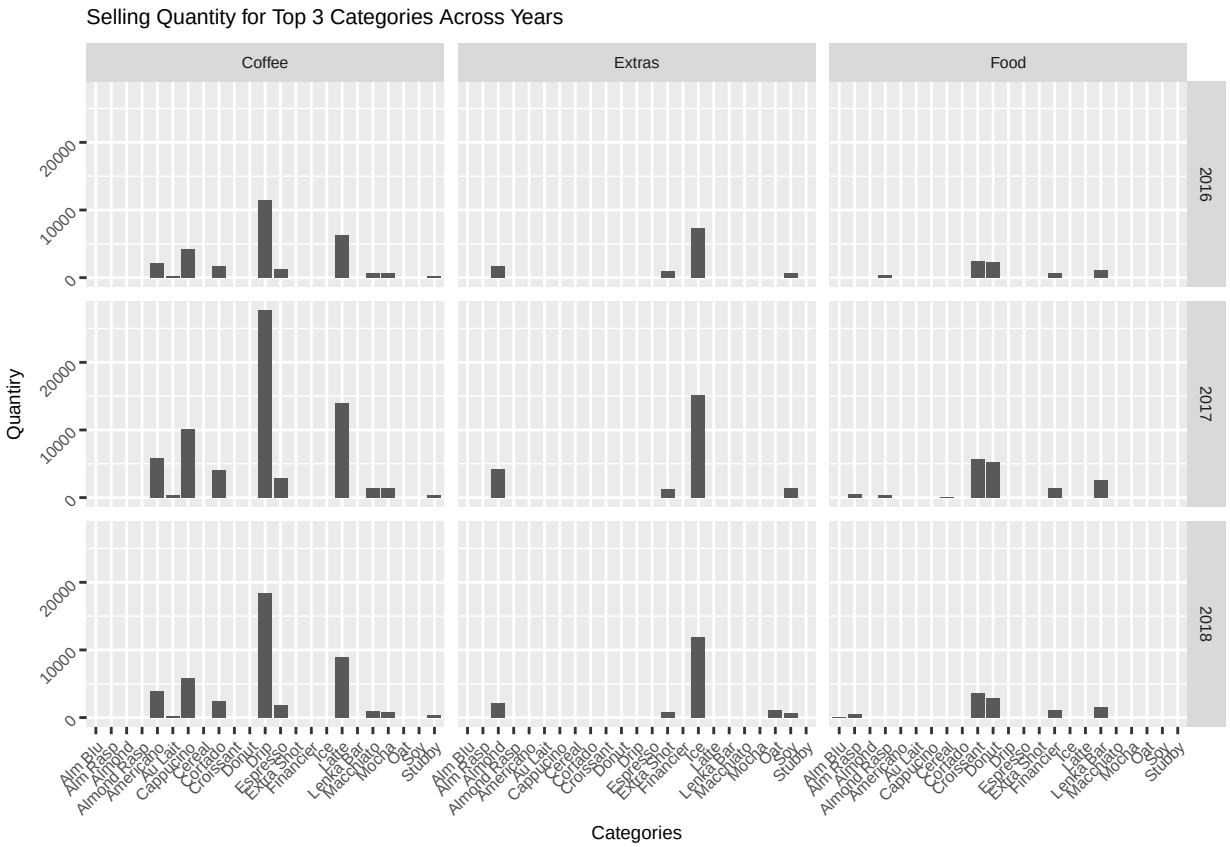
As shown in the below charts, customers buying patterns are quite stable across years. Coffee is the best selling category and main revenue source for Central Park, and with inside, drip, latte, cappuccino are the

top 3 sellers under coffee category. Next best selling category is the food category, in which croissant, donut, financier, lenka bar are the snacks people prefer to buy.

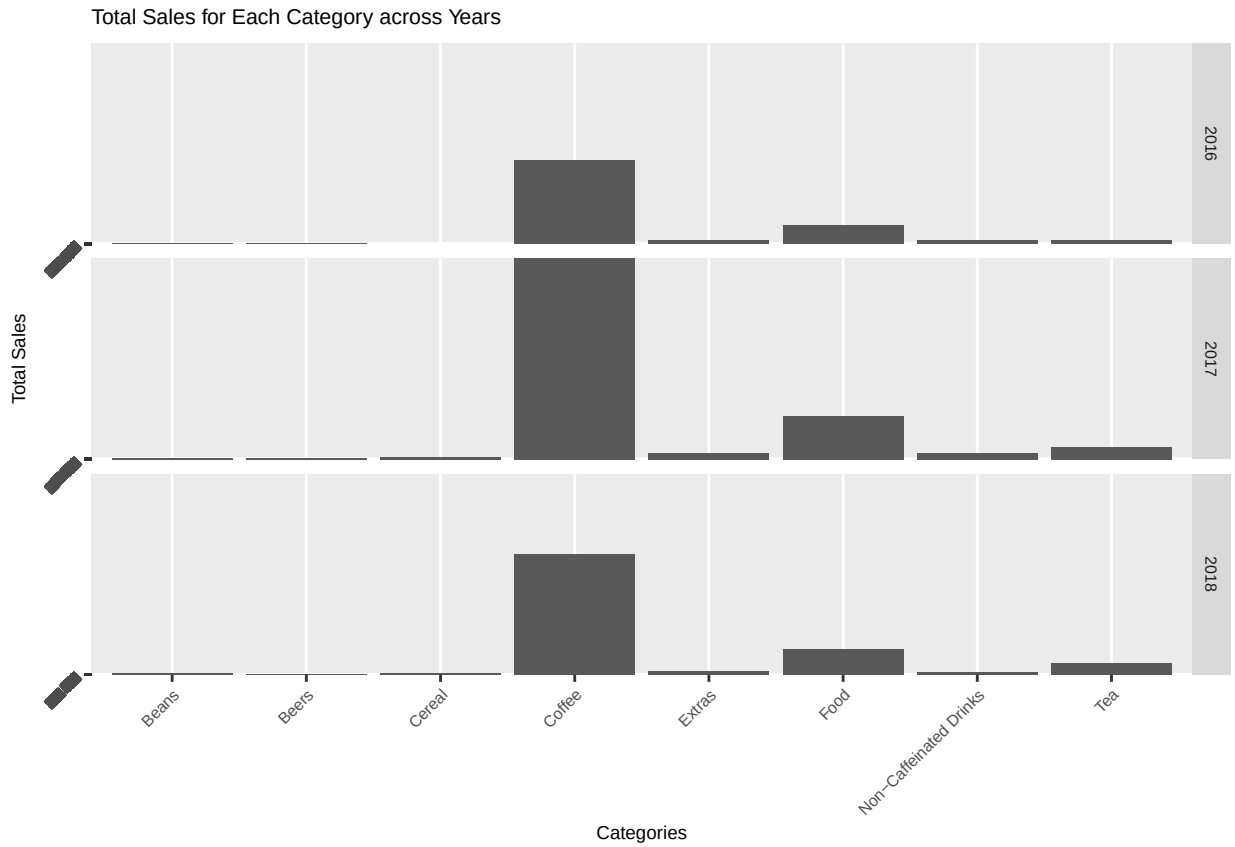
```
data_1$loyalty<-as.character(data_1$loyalty)
data_1<-data_1%>%mutate(loyalty=if_else(is.na(data_1$loyalty),'Non-Member',loyalty))
data$Time<-hms(data$Time)
data<-data%>%mutate(membership=ifelse(is.na(data$Customer.ID),'Non-Member','Member'))


 ggplot(data,aes(Category, Qty)) +
  geom_bar(stat="identity") +
  facet_grid(Year~.)+
  theme(text = element_text(size = 7),
                  axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Selling Quantity for Each Category Across Years") +
   xlab("Categories")+ylab("Qty")
```



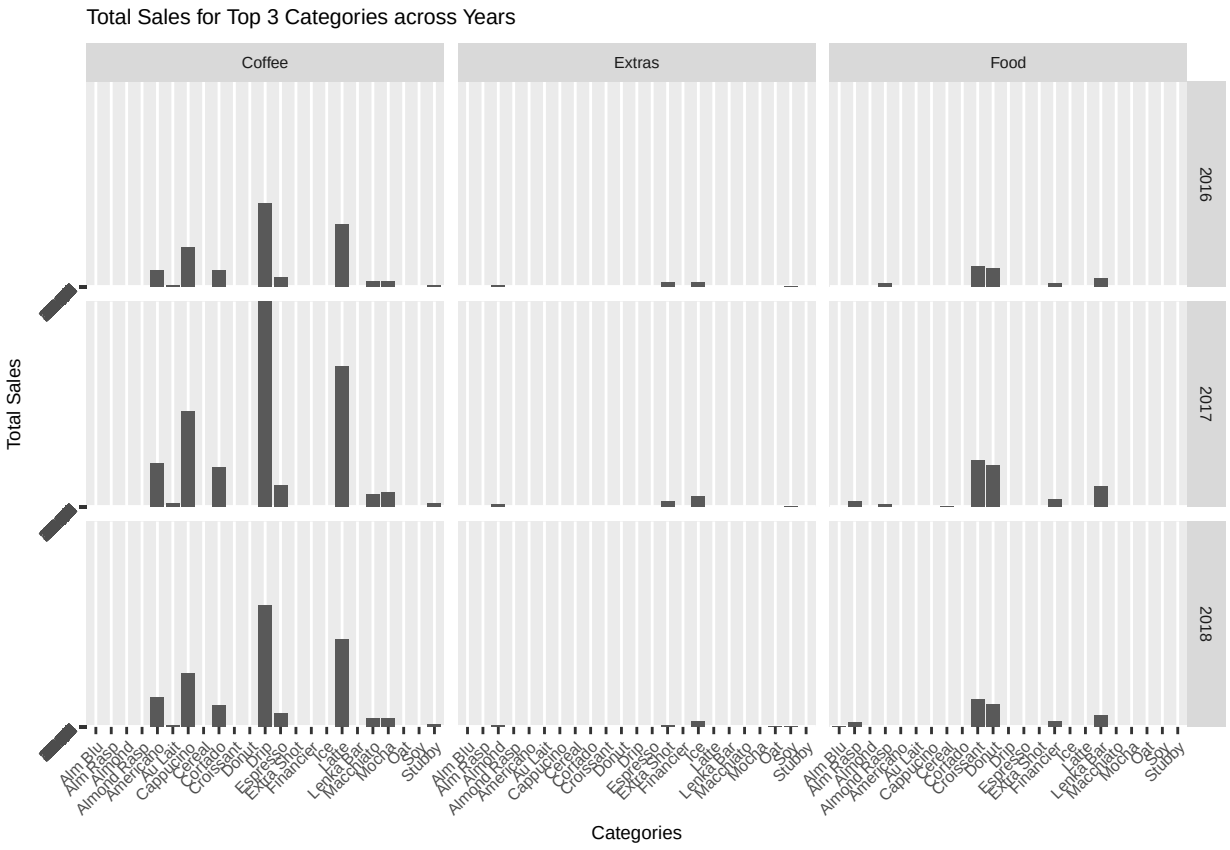

```
Top3Category <- data[data$Category %in% c('Coffee', 'Extras', 'Food'),]
ggplot(Top3Category,aes(as.factor(Item), Qty)) +
  geom_bar(stat="identity") +
  facet_grid(Year~Category)+
  theme(text = element_text(size = 7),
        axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Selling Quantity for Top 3 Categories Across Years")+
  xlab("Categories")+ylab("Quantiry")
```




```
ggplot(data,aes(Category, Net.Sales)) +
 geom_bar(stat="identity") +
 facet_grid(Year~.)+
 theme(text = element_text(size = 7),
                 axis.text = element_text(angle = 45, hjust = 1))+
 ggtitle("Total Sales for Each Category across Years") +
  xlab("Categories")+ylab("Total Sales")
```



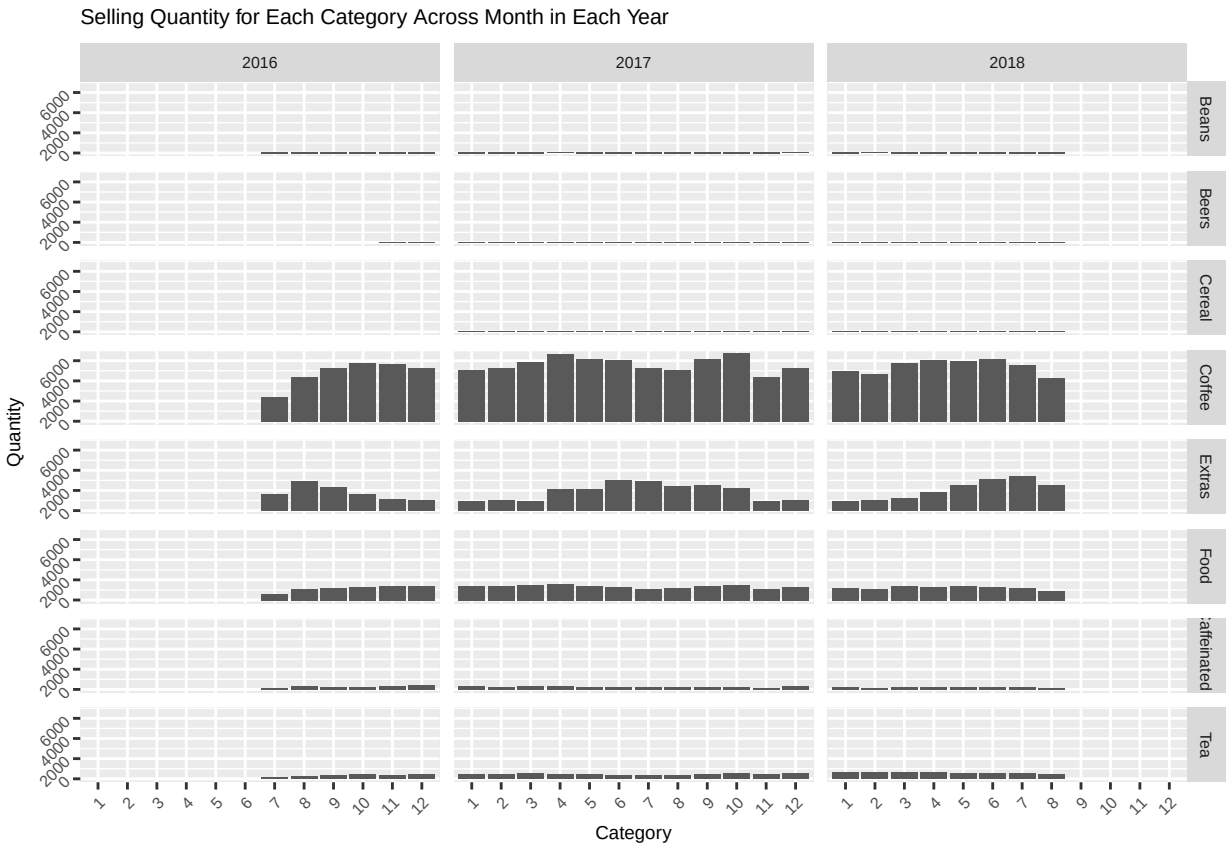

```
ggplot(Top3Category,aes(as.factor(Item), Net.Sales)) +
  geom_bar(stat="identity") +
  facet_grid(Year~Category)+
  theme(text = element_text(size = 7),
        axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Total Sales for Top 3 Categories across Years") +
  xlab("Categories")+ylab("Total Sales")
```



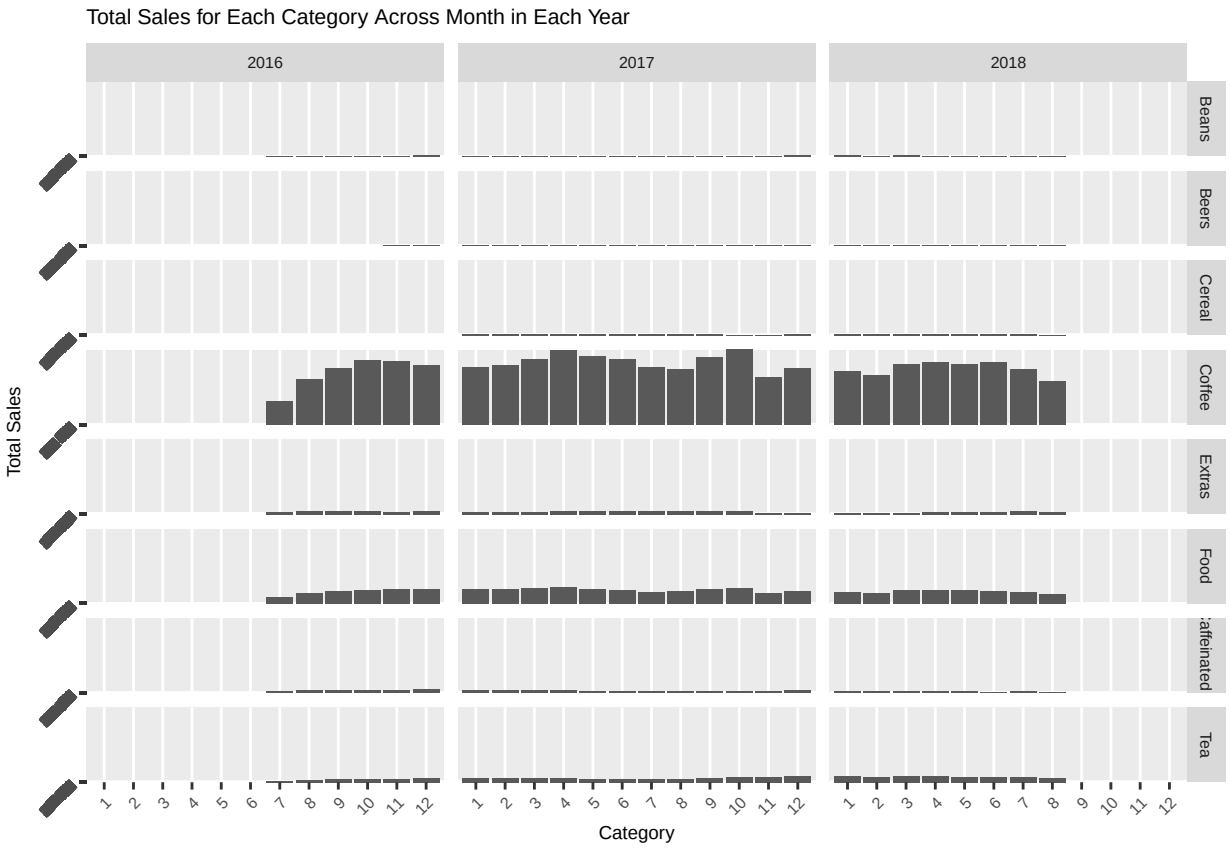
Deep diving into the months would help us understand seasonality of products

As shown in the below charts, customers buying patterns are quite stable across months. Coffee, Extras, and Food are also the top 3 selling categories in month level. But, sales drop during July and August.

```
data$Year_Month <- paste(data$Year, data$Month, sep = '_')

 ggplot(data, aes(Month, Qty))+
  geom_bar(stat = 'identity')+
  facet_grid(Category~Year)+
  theme(text = element_text(size = 7),
        axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Selling Quantity for Each Category Across Month in Each Year") +
  xlab("Category") + ylab("Quantity")
```



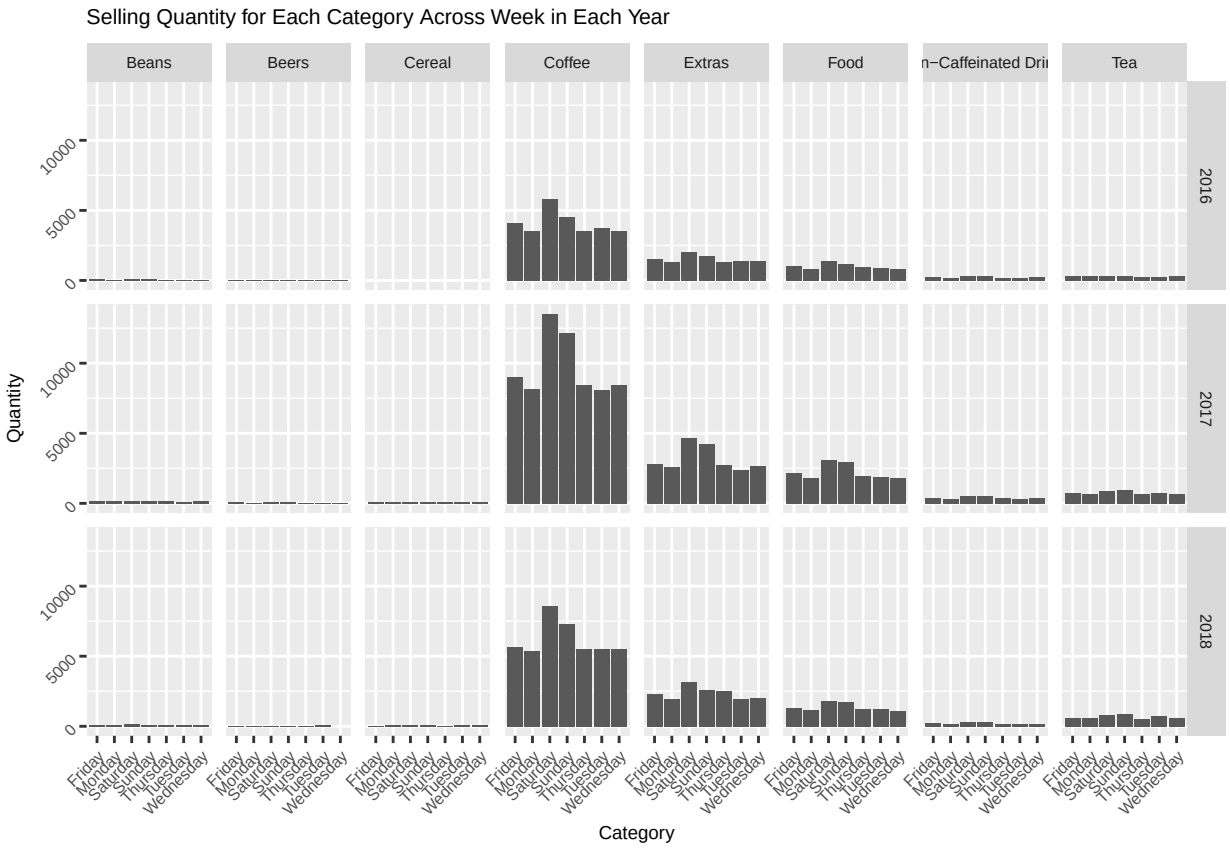

```
ggplot(data, aes(Month, Net.Sales))+
 geom_bar(stat = 'identity')+
 facet_grid(Category~Year)+
 theme(text = element_text(size = 7),
       axis.text = element_text(angle = 45, hjust = 1))+
 ggtitle("Total Sales for Each Category Across Month in Each Year") +
 xlab("Category") + ylab("Total Sales")
```



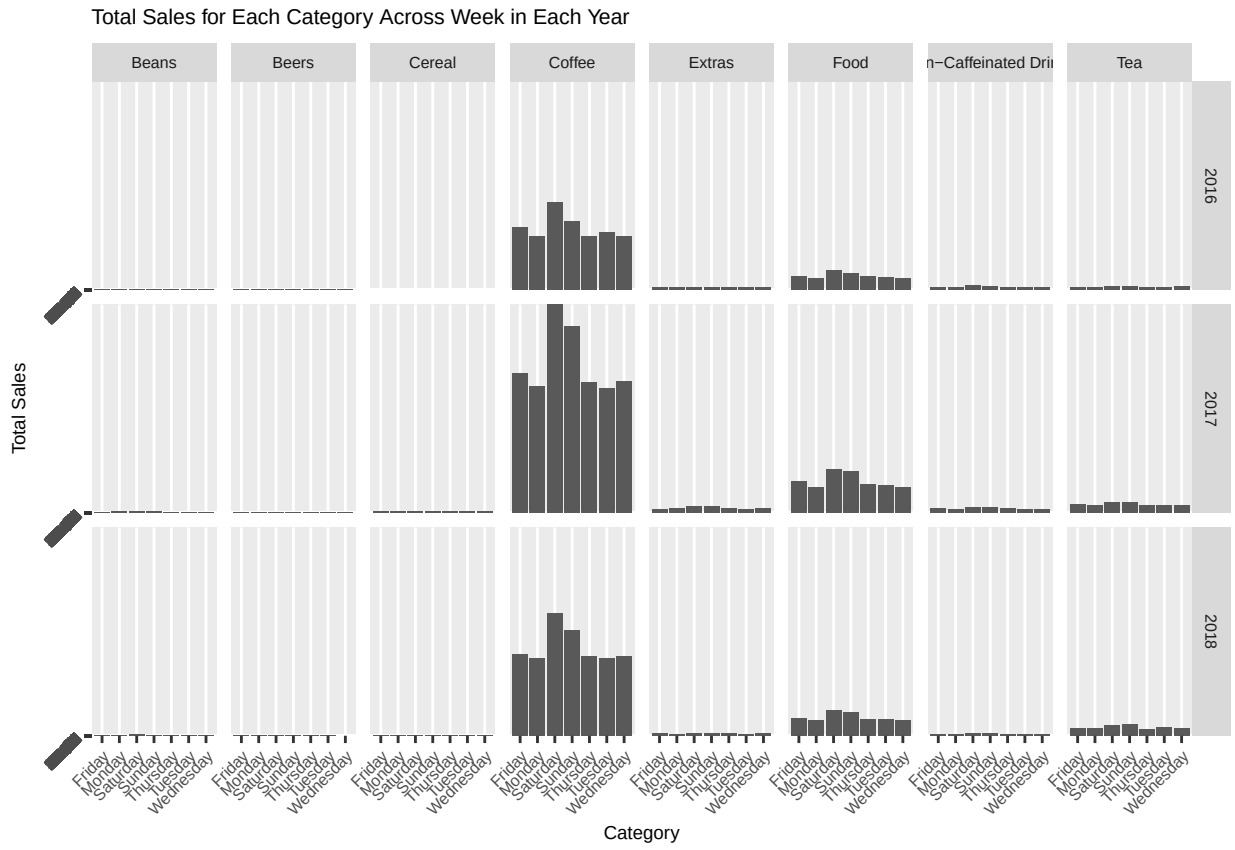
Distribution of Net-sales/Quantity across different days of the week

Across weekdays, sales increase during weekend, sales peak at Saturday, a slight drop in Sunday and stable sales across weekdays.

```
data$Year_Week <- paste(data$Year, data$Weekday, sep = '_')


ggplot(data, aes(Weekday, Qty))+
  geom_bar(stat = 'identity')+
  facet_grid(Year~Category)+
  theme(text = element_text(size = 7),
        axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Selling Quantity for Each Category Across Week in Each Year") +
  xlab("Category") + ylab("Quantity")
```



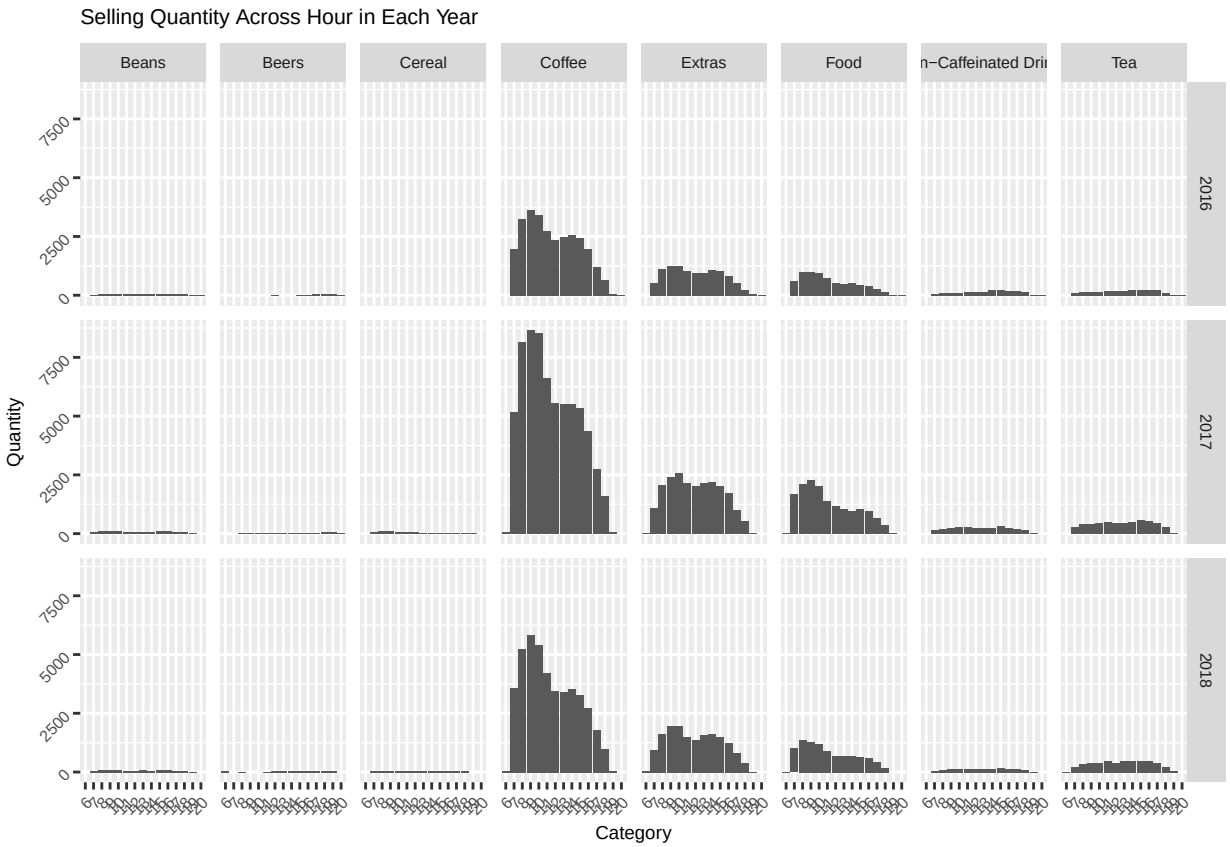

```
ggplot(data, aes(Weekday, Net.Sales))+
  geom_bar(stat = 'identity')+
  facet_grid(Year~Category)+
  theme(text = element_text(size = 7),
        axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Total Sales for Each Category Across Week in Each Year") +
  xlab("Category") + ylab("Total Sales")
```



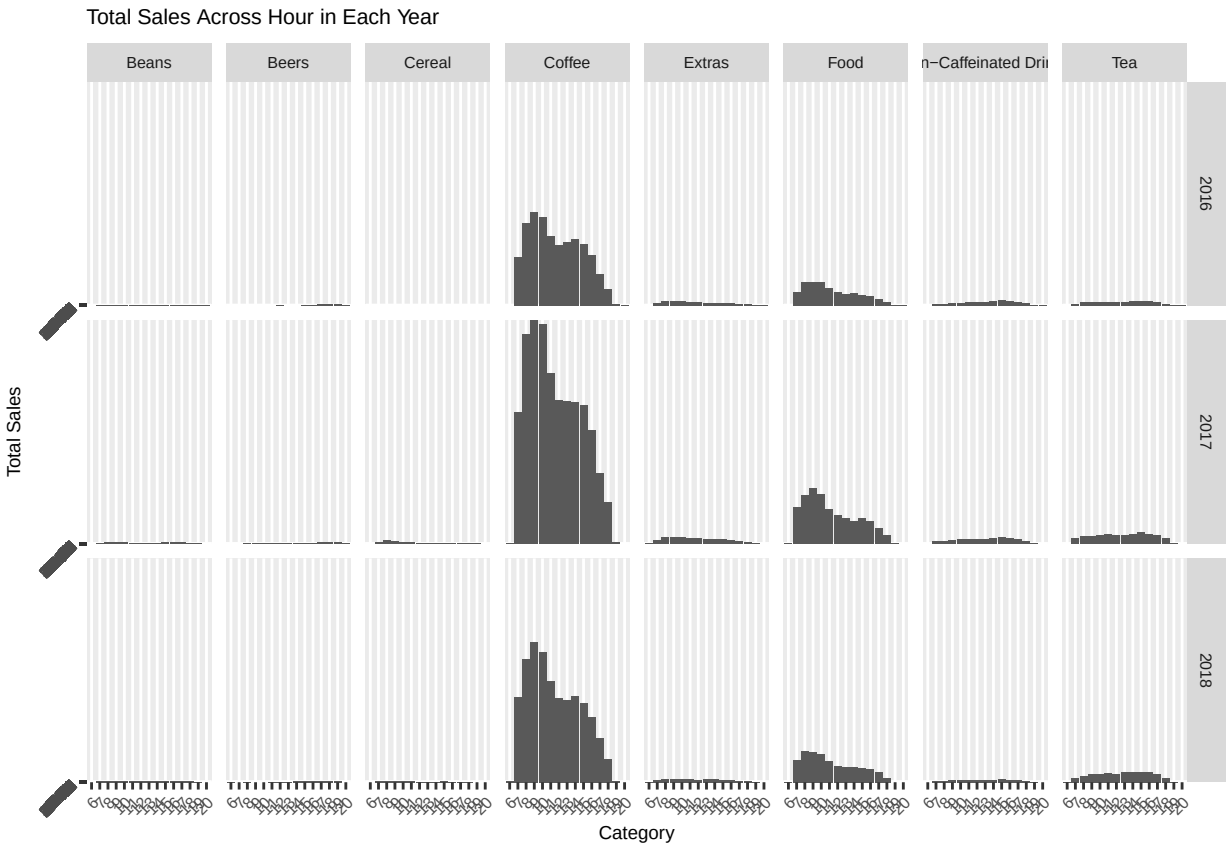
Distribution of Net-sales/Quantity across different hours of a day

Across hours, 8 to 10 o' clock is the peak hours in a day. Sales start to drop after 16 o' clock.

```
ggplot(data, aes(Hour, Qty))+
  geom_bar(stat = 'identity')+
  facet_grid(Year~Category)+
  ggtitle('Selling Quantity Across Hour in Each Year')+
  theme(text = element_text(size = 7),axis.text = element_text(angle = 45, hjust = 1))+
  xlab("Category") + ylab("Quantity")
```




```
ggplot(data, aes(Hour, Net.Sales))+
  geom_bar(stat = 'identity')+
  facet_grid(Year~Category)+
  ggtitle('Total Sales Across Hour in Each Year')+
  theme(text = element_text(size = 7),axis.text = element_text(angle = 45, hjust = 1))+
  xlab("Category") + ylab("Total Sales")
```



Association Rules

To understand the customer buying behaviors, we ran association rule for all transactions fall. This would also help us to show the probability of relationships between data items

Association Rules on all transactions with months on LHS

```
rule_data <- read.csv("all_transactions_month.csv")
orders = read.transactions(
  file = 'all_transactions_month.csv', header=TRUE,
  format = "basket",
  sep = ",",
  #cols = c("Timestamp","Combine"),
  rm.duplicates = T)

rules <- apriori(orders, parameter = list(supp = 0.01, conf = 0.01,minlen=1),
                 appearance = list(lhs =unique(rule_data$Month)))
```

```
sorted_rules<- head(sort(rules, decreasing = TRUE, by = c("lift")),10)
inspect(sorted_rules)
```

```
##      lhs            rhs         support    confidence lift     count
## [1]  {July}      => {Ice}       0.04329284 0.4657335  2.039108 5756
## [2]  {August}    => {Ice}       0.04296943 0.4370744  1.913631 5713
```

```
## [3]  {June}      => {Ice}       0.03249972 0.3832712  1.678066 4321
## [4]  {September} => {Ice}       0.02454214 0.3103186  1.358659 3263
## [5]  {December}  => {Cappucino} 0.01277124 0.1679359  1.221774 1698
## [6]  {July}      => {Drip}      0.04293182 0.4618497  1.192172 5708
## [7]  {August}    => {Drip}      0.04433079 0.4509219  1.163965 5894
## [8]  {February}  => {Cappucino} 0.01162047 0.1596404  1.161422 1545
## [9]  {May}       => {Ice}       0.02265428 0.2650942  1.160655 3012
## [10] {March}     => {Cappucino} 0.01320748 0.1553707  1.130359 1756
```

Association Rules on all transactions with Hours and Months in LHS

```
rule_data <- read.csv('all_transactions_hour.csv')
orders = read.transactions(
  file = 'all_transactions_hour.csv', header=TRUE,
  format = "basket",
  sep = ",",
  #cols = c("Timestamp","Combine"),
  rm.duplicates = T)
rules <- apriori(orders, parameter = list(supp = 0.01, conf = 0.01,minlen=1), appearance = list(lhs =c(
```

```
sorted_rules<- head(sort(rules, decreasing = TRUE, by = c("lift")),10)
inspect(sorted_rules)
```

```
##      lhs     rhs          support    confidence lift     count
## [1]  {7}  => {Croissant} 0.03382014 0.4153742  1.758426 1232
## [2]  {8}  => {Croissant} 0.04131437 0.3512252  1.486860 1505
## [3]  {16} => {Tea}       0.01004722 0.1512397  1.455962  366
## [4]  {9}  => {Croissant} 0.04362029 0.3413534  1.445069 1589
## [5]  {10} => {Croissant} 0.04016141 0.3261984  1.380913 1463
## [6]  {16} => {Donut}     0.01701987 0.2561983  1.245369  620
## [7]  {14} => {Americano} 0.01076095 0.1352657  1.232481  392
## [8]  {17} => {Donut}     0.01117272 0.2524814  1.227301  407
## [9]  {8}  => {Lenka Bar} 0.01685517 0.1432905  1.224727  614
## [10] {13} => {Donut}     0.01935324 0.2508897  1.219564  705
```

Interpretation for Association Rules

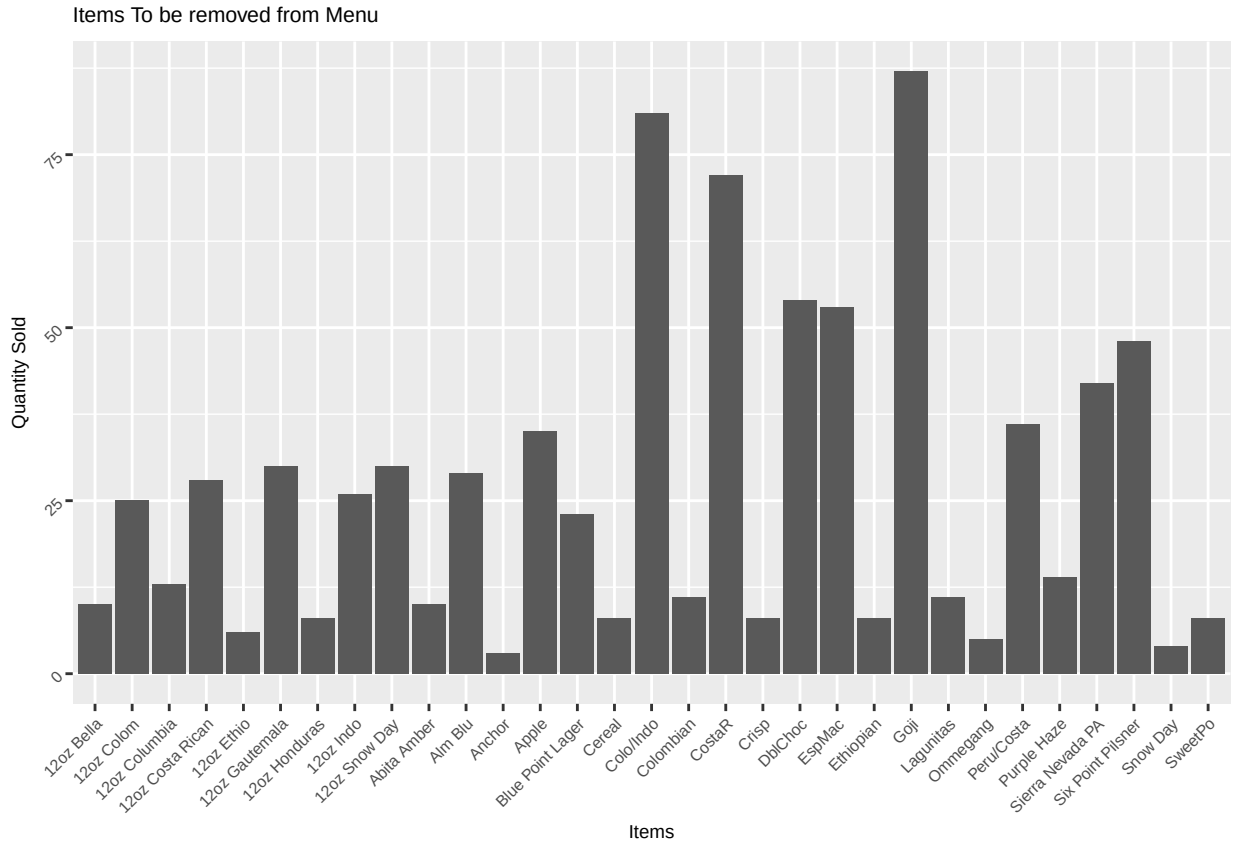
1.5, meaning that people buying croissant is more likely to occur when its at 7 to 10 o'clock. Same interpretation applies to 12,13,14,15 o'clock's lift in donut, and other hour's lift in Cappucino, Latte.

Association rules on hour help us further understand hinted consumption patterns on different hour, thus providing us insights on mainitaining inventory and promotions on non-peak hours.

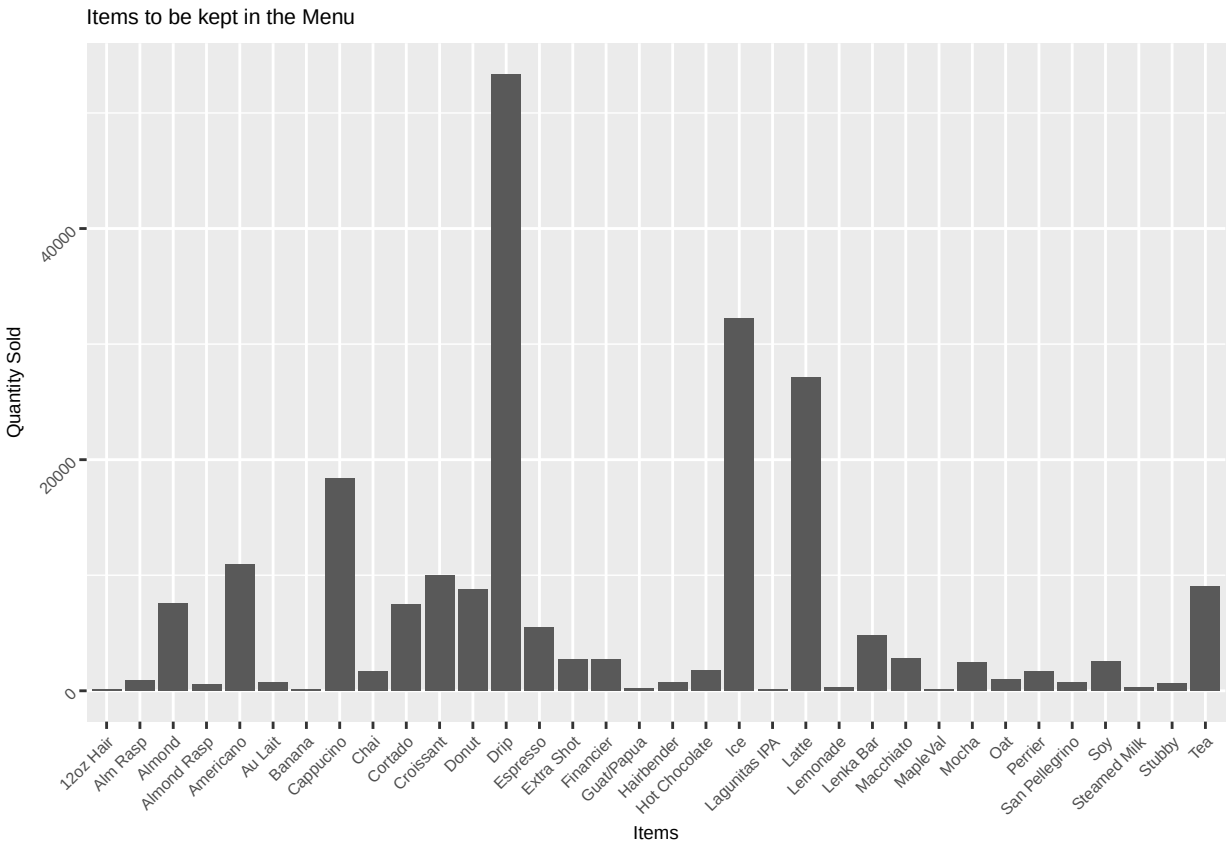
Effect of Purchasing Patterns

```
x<-data%>%select("Item","Month")%>%
  group_by(Item)%>%
  dplyr::summarise(count=n())%>%
  ungroup()%>%
  filter(count<=100)
ggplot(x,aes(Item,count),fill = count)+
```

```
  geom_bar(stat="identity")+
  theme(text = element_text(size = 7),
  axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Items To be removed from Menu") +
  xlab("Items") + ylab("Quantity Sold")
```




```
x<-data%>%select("Item","Month")%>%
  group_by(Item)%>%
  dplyr::summarise(count=n())%>%
  ungroup()%>%
  filter(count>=100)
ggplot(x,aes(Item,count), fill = count)+
  geom_bar(stat="identity")+
  theme(text = element_text(size = 7),
  axis.text = element_text(angle = 45, hjust = 1))+
  ggtitle("Items to be kept in the Menu") +
  xlab("Items") + ylab("Quantity Sold")
```



Inventory Insights

Above are the plots depicting the quantity sold for these items. We identified slow moving items. These are the items which have had very little consumption/sales across the two year period. We noticed that, in thes 2 years data, some category even don't have sales. For inventory management purpose, Central Perk could drop those items.

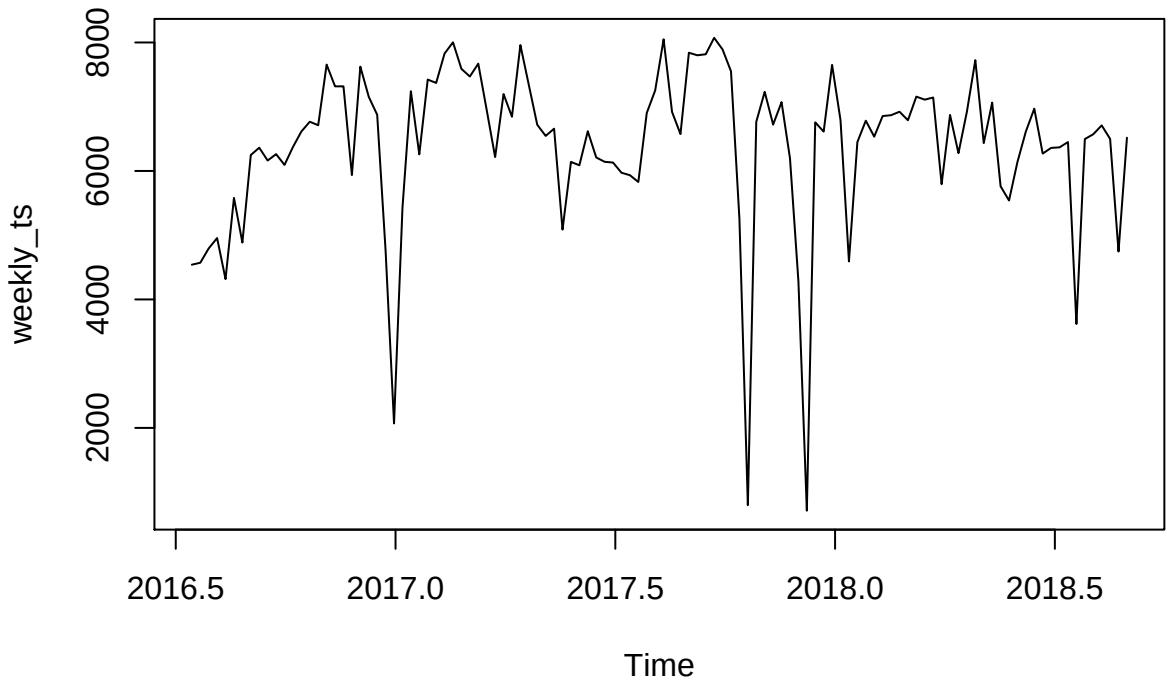
3. Analysing underperforming periods

We identify weeks where we perform much lower than expected by first decomposing the seasonality of data in the transactions. To do we started by understanding the trends at weekly level and splitting them into short-term seasonality and long-term trends. We are left with the remainder which measures the noise present in the data. Analysing this gives an idea of the anamolies present in the data.

```
master_data <- data

master_data <- master_data[,c(2:21)]

# Changing data to to get weekly trends
data1 <- master_data[,c(1,8,14:20)] # subsetting required columns

# sorting by date ascending
#data1$Date <- lubridate::mdy(data1$Date)

data1$Timestamp <- lubridate::ymd_hms(data1$Timestamp)
```

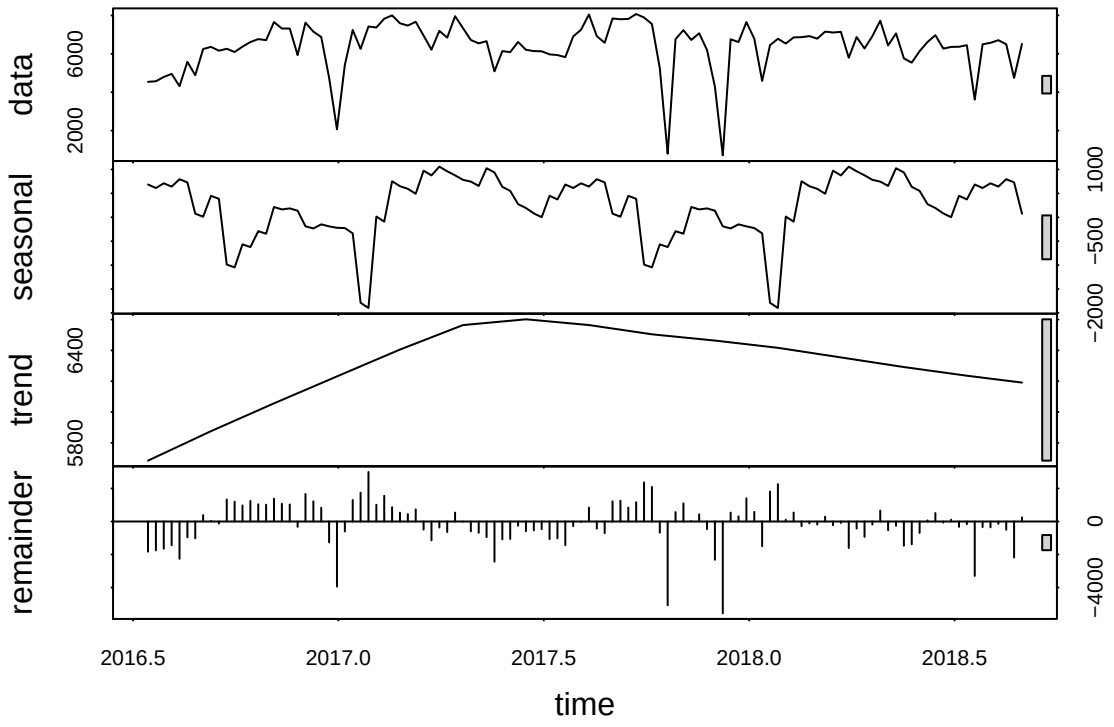
```r
data1$week <- lubridate::week(data1$Timestamp)

data1 <- data1 %>% arrange(Year,week)
data1$year_week = paste(data1$Year,data1$week)

data2 <- data1 %>% select(year_week,Net.Sales)

data3 <- aggregate(as.numeric(data2$Net.Sales), by=list(Category=data2$year_week), FUN=sum)

names(data3) <- c("year_week","Net_Sales")

data3 <- as.data.frame(data3)
write.csv(data3)
data_vector <- as.vector(data3$Net_Sales)

weekly_ts <- ts(data_vector,freq=52.17746,start = c(2016,29))
```

```r
plot(weekly_ts) #plotting the weekly trend
```



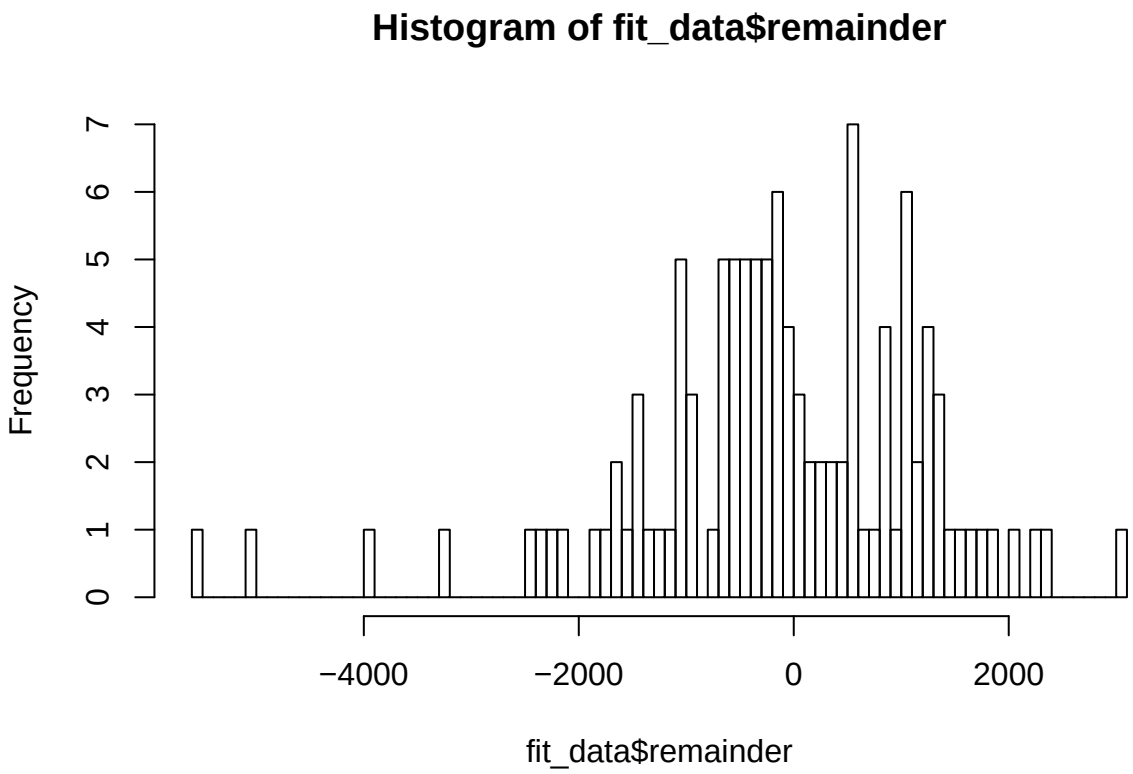
After preparation of the data. We first start by plotting the trends at weekly level. We see couple of steep dips.

```r
fit <- stl(weekly_ts, t.window=NULL, s.window="periodic", robust=TRUE)
plot(fit)
```



The data is seasonally decomposed into 3 parts - the acutal data, sesonality which showcases the short term effects and the trends which show long term pattern. The remainders showcases the noise in the data.

```
fit_data<- as.data.frame(fit$time.series)
hist(fit_data$remainder,100)
```



The data is not normally distributed as shown in the graph. We decomposed the data to get the top weeks where we dont perform as well as expected. These weeks are showcased below-

The weeks where we perform worse than the expected trend are :

```
remainder_data <- cbind(data3$year_week, fit_data$remainder)
remainder_data<- as.data.frame(remainder_data)
names(remainder_data) <- c("year_week" , "remainder" )
remainder_data <- remainder_data %>%  arrange(remainder)
remainder_data$remainder <- as.numeric(remainder_data$remainder)
remainder_data$z_score <- scale(remainder_data$remainder)

remainder_data$year_week[remainder_data$z_score < -0.5]
```

```
##  [1] 2016 35 2017 34 2017 29 2017 33 2017 28 2018 15 2017 2  2016 38
##  [9] 2016 52 2018 27 2018 6  2017 35 2016 32 2018 26 2018 1  2018 33
## [17] 2018 2  2016 31 2016 30 2018 16 2016 29 2018 22 2018 8  2018 18
## [25] 2016 33 2017 52 2017 27 2017 3  2017 37 2018 25 2018 14 2017 36
## [33] 2016 48 2018 32 2018 34 2018 4  2018 5  2017 20 2016 53 2018 20
## 112 Levels: 2016 29 2016 30 2016 31 2016 32 2016 33 2016 34 ... 2018 9
```

The weeks where we perform better than the expected trend are

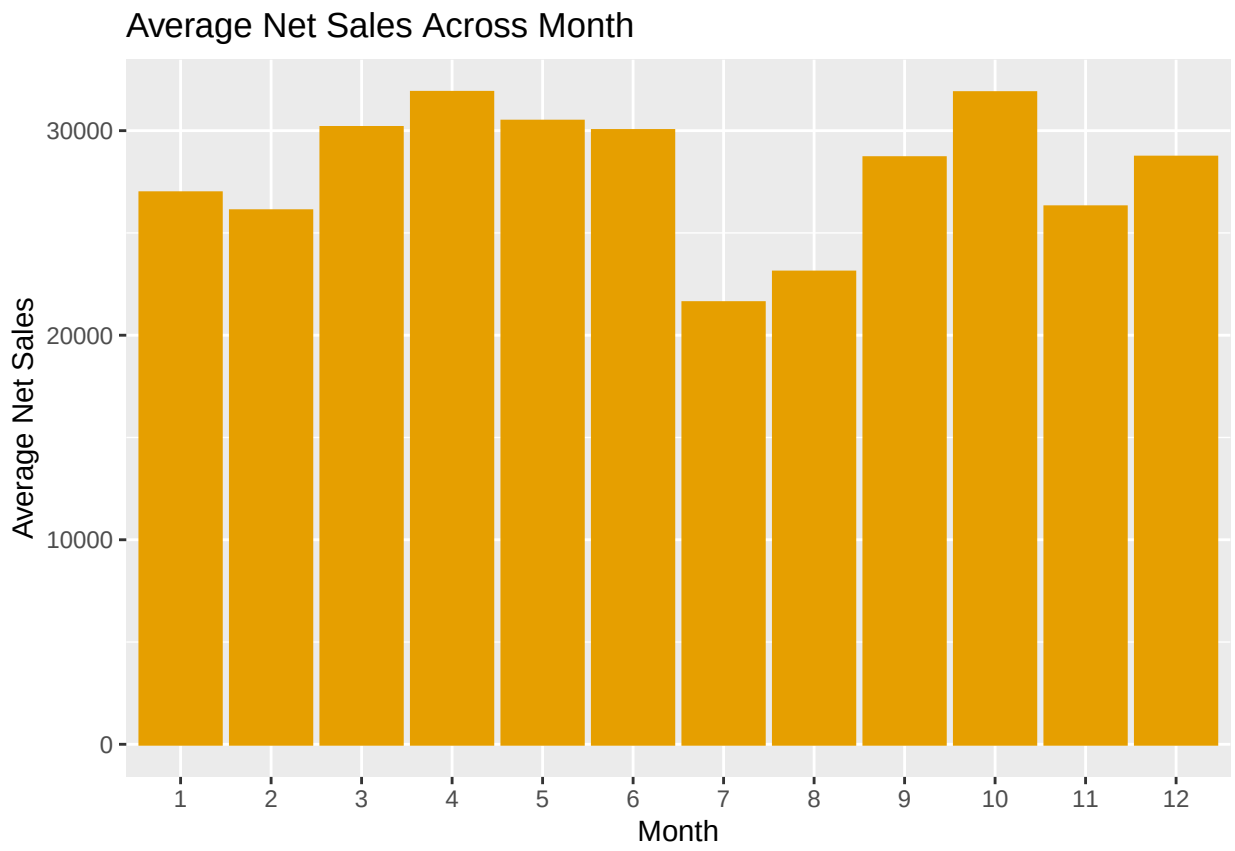
```
remainder_data$year_week[remainder_data$z_score > 0.5]
```

```
##  [1] 2016 40 2016 50 2017 40 2016 42 2017 41 2017 10 2016 39 2016 45
```

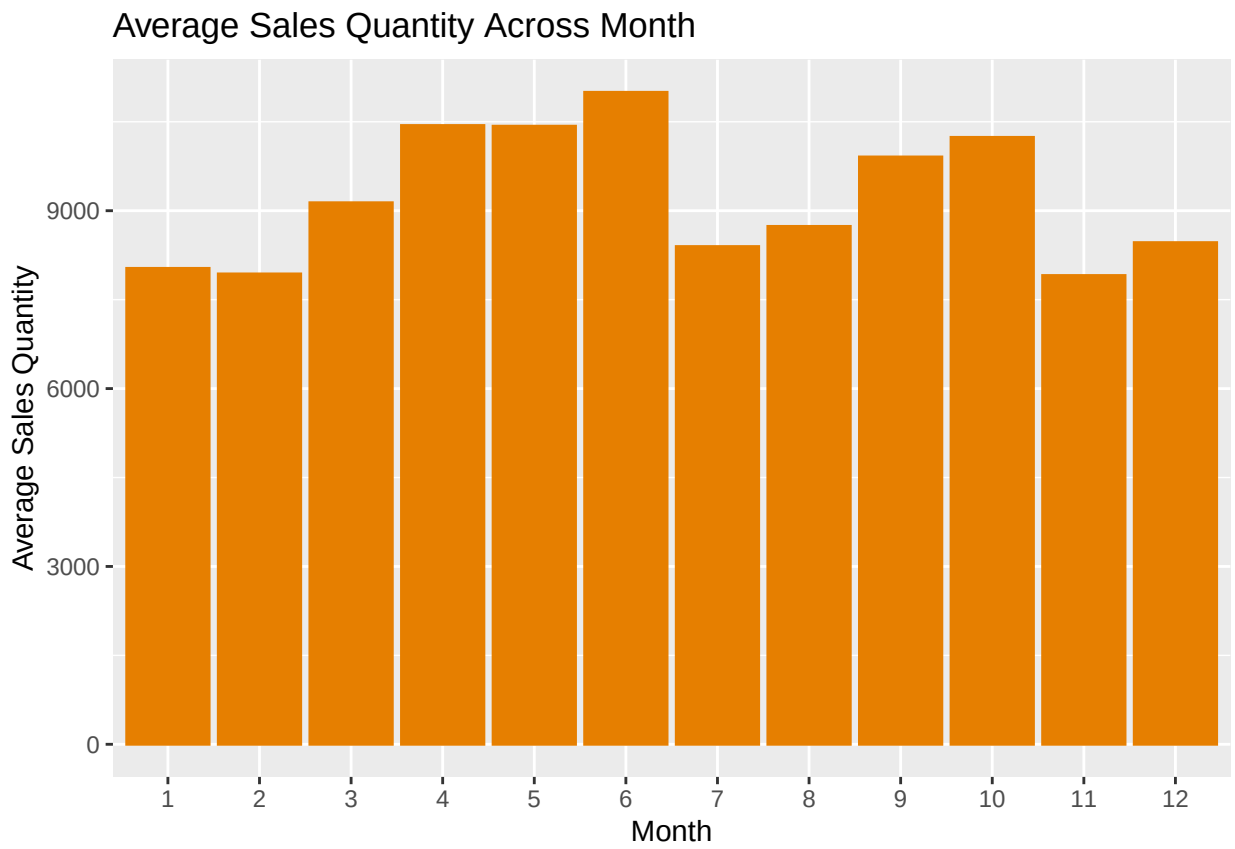
```
##  [9] 2017 8  2017 14 2016 49 2017 11 2018 10 2017 45 2018 11 2017 44
## [17] 2018 9  2017 5  2018 17 2017 12 2017 7  2016 37 2016 36 2017 50
## [25] 2017 17 2018 3  2017 16 2017 6  2017 22 2018 13 2017 48 2017 9
## [33] 2018 23 2018 29 2017 18 2016 51 2017 42 2017 38 2017 15 2016 41
## 112 Levels: 2016 29 2016 30 2016 31 2016 32 2016 33 2016 34 ... 2018 9
```

Distribution of Average Net-Sales and Quantities Across Months

```
#Open the dataset we had cleaned and transformed
clean <- data
#Calculate the average net sale across month
x<- clean %>%select("Year","Net.Sales","Month")%>%
  group_by(Year,Month)%>%
  dplyr::summarise(net_sales=sum(as.numeric(Net.Sales)))%>%
  ungroup()%>%
  group_by(Month)%>%
  dplyr::summarise(avg=mean(net_sales))%>%
  ungroup()

x$Month = as.factor(x$Month)

#Draw average net sales distribution across month
ggplot(x, aes(x =Month, y = avg))+ geom_bar(stat='identity',fill ="#E69F00",
                                            color = "#E69F00")+
  ylab("Average Net Sales")+xlab('Month')+
  ggtitle("Average Net Sales Across Month")
```




```
#Calculate Average sales quantity across month
y<- clean %>%select("Year","Qty","Month")%>%
  group_by(Year,Month)%>%
  dplyr::summarise(Q=sum(Qty))%>%
  ungroup()%>%
  group_by(Month)%>%
  dplyr::summarise(avg=mean(Q))%>%
  ungroup()

#Draw average sales quantity distribution across month
y$Month = as.factor(y$Month)
ggplot(y, aes(x =Month, y = avg))+ geom_bar(stat='identity',fill ="#E67F00",
                                            color = "#E67F00")+
  ylab("Average Sales Quantity")+
  xlab('Month')+
  ggtitle("Average Sales Quantity Across Month")
```



After we calculate the average net sales and quantities across 12 month, we found out that July and August have lowest average net sales and sales quantity in general.

Recommendation

1. Loyalty program

For loyal customers, our recommendation is to implement a loyalty point program which incentivizes customers to return to the store. Approximately 80% of our customer base is loyal/semi-loyal. Therefore, a rewards-based point system which offers customers 15% of the total amount spent per order would stimulate sales further and force people to enroll into our point program. Providing offers like 'Register & your first coffee is on us' would furthermore decrease the customer churn rate and encourage one-time customers (those without customer IDs) to enroll. Customers who refer their friends into the program can also be offered more rewards into the card to keep them active simultaneously increasing our customer base.

2. Happy hour - Double reward rate

Central Perk customers seldom buy multiple items in a one-time transaction; only 79,166 (35%) transactions involve multiple items. Given Central Perk's steady customer flow and buying behaviors, Happy hours during evenings on "Beers and Food Items" from 7 to 10 pm will help reduce the slump during these hours on these low moving items. We can also offer all-day happy hours during periods where the sales is much lower than expected volumes (which are weeks 27-38 and weeks 52,53 of a year - they majorly fall in the months July, August and December) This offer would allow customers to acquire twice the reward points they would get from a regular purchase.

12oz Bella	12ozSnowDay	Colombian	Ommegang	12ozIndo
12ozColom	AbitaAmber	CostaR	Peru/Costa	Colo/Indo
12ozColumbia	AlmBlu	Crisp	PurpleHaze	Lagunitas
12ozCostaRican	Anchor	DblChoc	SierraNevadaPA	
12ozEthio	Apple	EspMac	SixPointPilsner	
12ozGautemala	BluePointLager	Ethiopian	SnowDay	
12ozHonduras	Cereal	Goji	SweetPo	

Figure 1: Items with little consumption/sales

3. Reorganise inventory

The items showcased below have very little consumption/sales across the two year period. These items in total only account for 0.34% of the total transactions but make up nearly 50% of our inventory. The demand of these items is negligible irrespective of the season. Removing these items from their menu decrease our inventory, storage and shipping costs. It will also allow the Cafe to standardize its menu and focus on improving our existing products. We are assuming that all the items are part of their standard menu and none of the items are seasonal specials.

4. Bundling products

People have a tendency to buy donuts in the afternoon and Croissants in the mornings, they also tend to buy more Lenka Bars and Cappuccino's in August. Central Perk can leverage this and bundle the products customers are inclined to buy with slow moving items. Bundles can be created in such a way that people are offered a food item and a drink item. Drip, Latte, Cappuccino, Donuts, Lenka bars and Croissants can be one products in the bundle. Offering a 10% discount on bundles during slow moving weeks of July, August and December can help us increase our inventory movement.

5. Free add-ons in lean periods

For transactions in July and August, we found out that when people buy a latte, they are 3 times more likely to add extra oat, almond or soy than random chance. Therefore, to normalize the dipping sales in the months of July and August, we recommend Central Perk to offer complimentary 'add on' items for people who buy a latte. This way we multiply our sales by cashing on the popularity of the most popular product of the season.

6. Free ice in summer

There is an increase in drip sales on purchase of ice in July and August. If we stop charging for ice in summer it would likely increase the sales of drinks sold in the same period. The price for ice in these months is the same price as all other extras, such as soy, almond, etc. The margin on ice is only 1% of total revenue earned, the loss in revenue could be compensated for by the increase in sales from other products. We could test this recommendation by A/B testing it on selected weeks and observing the change in revenue.