Central Perk

Team 5

November 2, 2019

Contents

Our Task 1

Our Approach 1

Data Preparation 2

Data Cleaning 2

Data Transformation 3

Sales Pattern Overview 3

Transaction Traffic Trend 3

What Kind of Items are Popular? 6

Who Contributes to the most Transactions/Sales? Members or Non-members? 8

Central Perk's Customer Base 9

Why Focusing on Loyal Customers? 11

Bundling 18

Staff Shifting and Optimization 19

Final Takeaways 19

Our Task

The client for this project is Central Perk, a New York City coffee shop. The shop believes it has a loyal customer base and is looking to profit from existing customers rather than acquiring new ones. Our team will analyze the data to assess the validity of the assumption of a loyal customer base. If indeed their customers are loyal there are several business benefits. Retaining customers generally require less marketing cost than acquiring new ones. Customers enrolled in a loyalty program are easier to contact assuming an email or phone number was required to enroll in the first place, and thereby gathering data from such customers allows you to identify buying patterns and personalize marketing campaigns. Another goal of the client is to smoothen the demand. Normalizing demand could potentially lead to more efficient staffing or improved stocking and supply chain.

Our Approach

To help Central Perk gain maximum returns on dollar expenses, it is essential that we identify the right target cohort and tailor the marketing tactics based on their purchase behavior. We will define the core loyal customers, based on their transaction history and not just merely based on CustomerID. Adopting this methodology would ensure any small change in their pattern would result in sustained gains to the cafe.

After the target group is identified, we will leverage marketing tactics to focus on normalizing the demand so that Central Perk ensures that revenue generation is more uniformly distributed leading to better staffing and enhanced customer experience.

Data Preparation

Data Cleaning

Loading Packages and Data

```r
library(naniar)
library(dplyr)
library(stringr)
library(tidyverse)
library(arules)
library(Hmisc)
library(arulesViz)
library(grid)
library(lubridate)
library(tidyr)
library(ggplot2)


df1 <- read.csv("Central Perk Item Sales Summary 2017.csv")
df2 <- read.csv("Central Perk Item Sales Summary 2016.csv")
df3 <- read.csv("Central Perk Item Sales Summary 2018.csv")
df <- rbind(df3, df2, df1)
```

Inconsistent data

We first observed inconsistent data in the “Item” column. Noisy strings, such as “12oz”, resulted in different names for the same item, so we removed or replaced them to increase consistency in our data. For example, “Almond Rasp” and “Alm Rasp” represent the same item, so we changed the item “Alm Rasp” to “Almond Rasp” to maintain data consistency.

```r
df$Item <- as.character(df$Item)
df$Item <- ifelse(df$Item == 'Alm Rasp', 'Almond Rasp', df$Item)
df$Item <- gsub("^12oz ", "", df$Item)
df$Item <- ifelse(str_detect(df$Item, "Lemonade") == TRUE, "Lemonade", df$Item)
```

Noisy data

We then dealt with the noisy data in the “Category”, “Qty”, “Discounts”, “Net.Sales”, “Gross.Sales” and “Tax” Columns.

```r
# Item "Oat" has two categories "Extra" and "None", unifying its category to "Extra"
df$Category[df$Item == 'Oat'] <- 'Extras'
# Filter out "None" category and negative quantity, which means "Refund"
df <- df %>% filter(Category != 'None')%>% filter(Qty > 0)



df$Discounts <- str_replace(df$Discounts, "[$]", "")
df$Discounts <- str_replace(df$Discounts, "[(]", "-")
df$Discounts <- str_replace(df$Discounts, "[)]", "")
df$Discounts <- as.numeric(df$Discounts)
```

```r
df$Net.Sales <- str_replace(df$Net.Sales, "[$]", "")
df$Net.Sales <- str_replace(df$Net.Sales, "[(]", "-")
df$Net.Sales <- str_replace(df$Net.Sales, "[)]", "")
df$Net.Sales <- as.numeric(df$Net.Sales)

df$Gross.Sales <- str_replace(df$Gross.Sales, "[$]", "")
df$Gross.Sales <- str_replace(df$Gross.Sales, "[(]", "-")
df$Gross.Sales <- str_replace(df$Gross.Sales, "[)]", "")
df$Gross.Sales <- as.numeric(df$Gross.Sales)

df$Tax <- str_replace(df$Tax, "[$]", "")
df$Tax <- str_replace(df$Tax, "[(]", "-")
df$Tax <- str_replace(df$Tax, "[)]", "")
df$Tax <- as.numeric(df$Tax)
```

Data Transformation

In the last step for data preparation, we transform the data type of "Date" and "Hour" columns from factor to date and generate three new columns month, weekday and hour, which can help us develop deep investigation in sale pattern in terms of season, week and day. The new dataset after cleaning has 221,389 rows while the original dataset has 221,561 rows.

```r
df <- df[df$Date != 'Unknown Error',]
#make a copy
df4 <- df
df$Date <- mdy(df$Date)
df$Time <- hms(df$Time)
df$Month <- month(df$Date)
df$Hour <- hour(df$Time)
df$Weekday <- wday(df$Date, week_start = 1)

dim(df)
```

```
## [1] 221389     16
```

Sales Pattern Overview

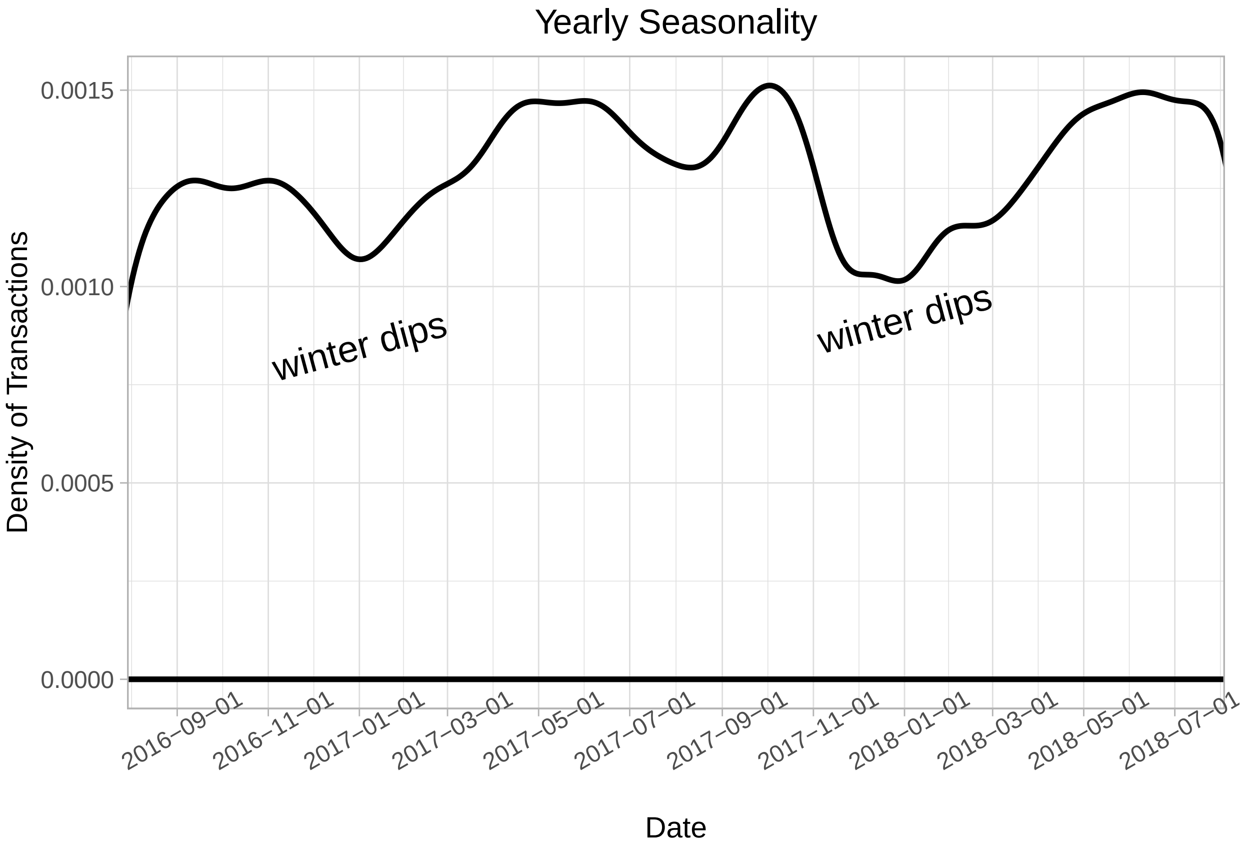
Our analysis of the sales of the coffee shop starts with a general exploration of the sales pattern, including how does the number of transactions change seasonally, daily and hourly, as well as who contributes the most to overall sales and which kinds of items are popular as well as which are not.

Transaction Traffic Trend

The number of transactions is indicative of overall traffic changes in the coffee shop, which can help develop strategies in staffing and insights on customers' consumption times. For example, are most of the customers coming at a certain period or are they spread out over different periods?

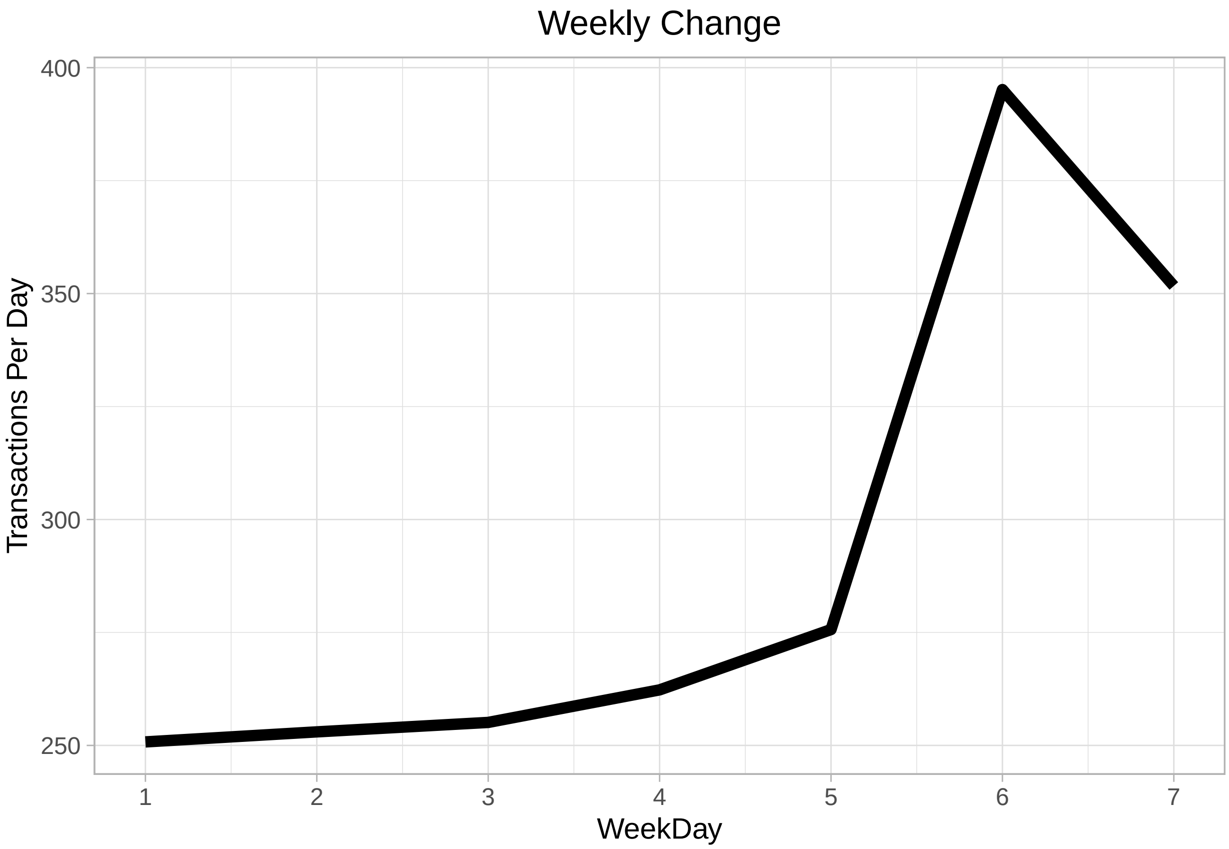
Seasonaility

```
ggplot(df, aes(x = Date)) + geom_density(size = 1) +
  coord_cartesian(xlim=c(as.Date("2016-09-01"), as.Date("2018-07-01"))) +
  scale_x_date(date_breaks = "2 months") +
  theme_light() +
  theme(axis.text.x = element_text(angle = 30)) +
  labs(title= "Yearly Seasonality", y="Density of Transactions", x = "Date") +
  theme(plot.title = element_text(hjust = 0.5)) +
  annotate("text", x = as.Date("2017-01-01"),
           y = 0.00085, label = "winter dips", angle=15, size=5,
           color='black') +
  annotate("text", x = as.Date("2018-01-01"),
           y = 0.00092, label = "winter dips", angle=15, size=5,
           color='black')
```



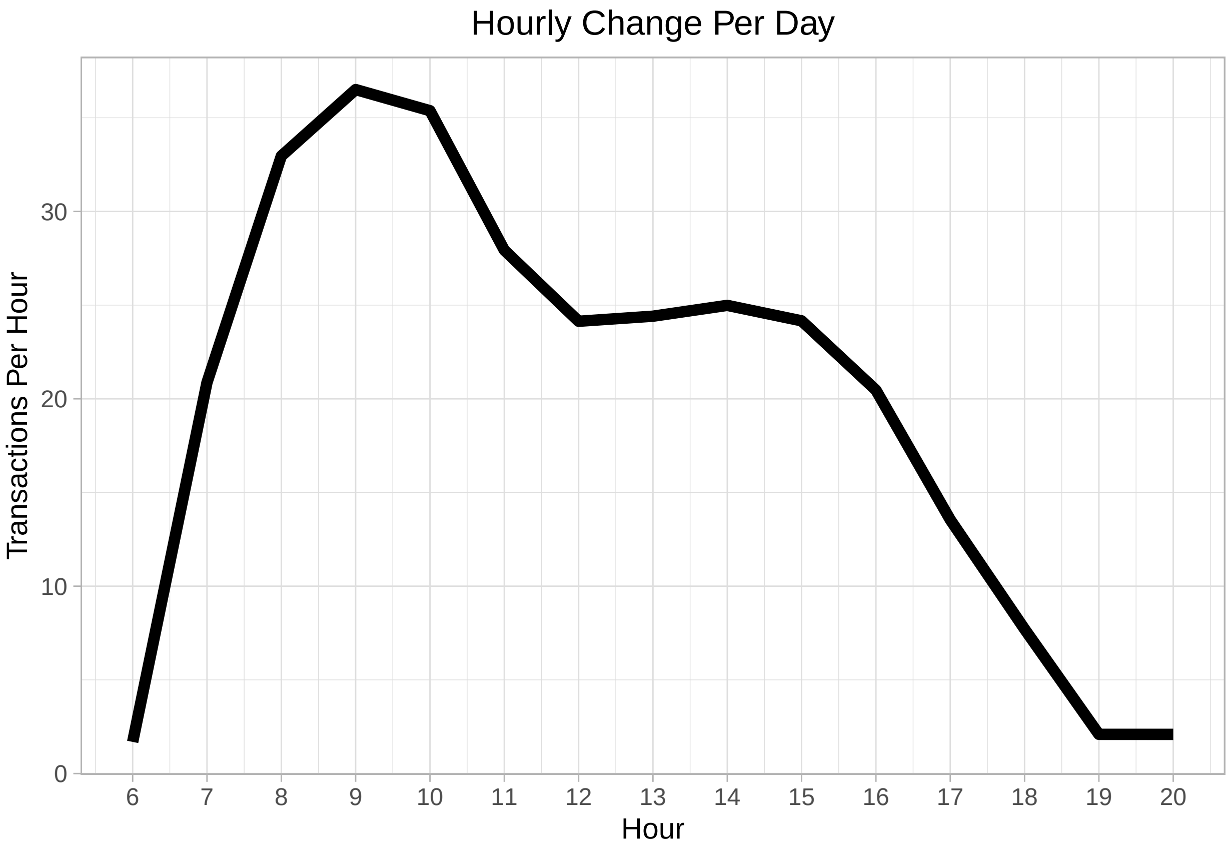
Daily

```
w <- df %>% group_by(Weekday)%>%summarise(tran_per_day = n()/length(unique(Date)))
ggplot(w, aes(x=Weekday, y=tran_per_day))+geom_line(size = 2)+
  scale_x_continuous(breaks = 1:7) +
  labs(title= "Weekly Change", y="Transactions Per Day", x = "WeekDay") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5))
```



Hourly

```r
d <- df %>% group_by(Hour)%>%summarise(tran_per_hour = n()/length(unique(Date)))
ggplot(d, aes(x=Hour, y=tran_per_hour))+geom_line(size = 2)+
  scale_x_continuous(breaks = 6:20) +
  labs(title= "Hourly Change Per Day", y="Transactions Per Hour", x = "Hour") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5))
```

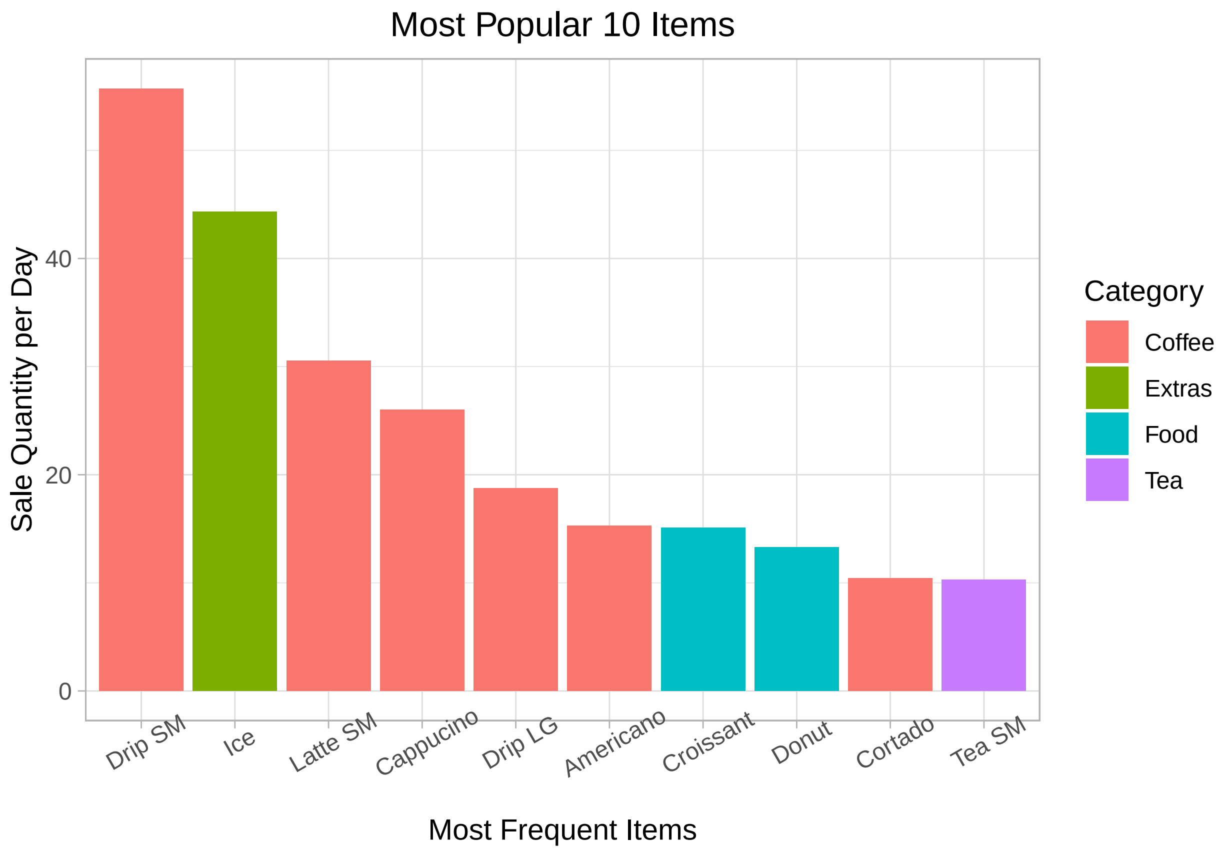


While the data spans 770 days it only captures one full calendar year, however, the plots above indicate that transactions dip during the winter each year, while transactions during other seasons remain higher. Transaction numbers on weekends are significantly higher than on weekdays. As for hourly change, it is obvious that there is a peak period for transactions in the morning from 8 am to 11 am, followed by the afternoon from 11 am to 4 pm, while early morning and late evening have the fewest transactions.

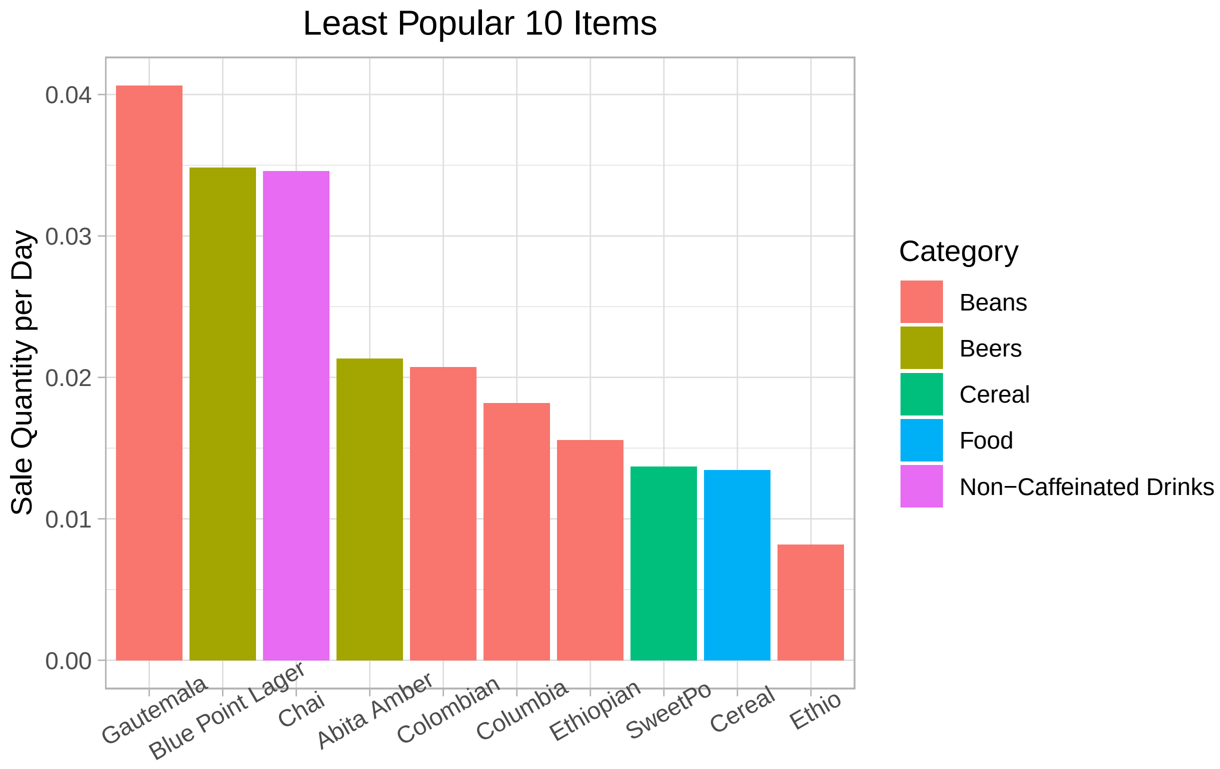
What Kind of Items are Popular?

Lastly, we would like to get an overview of consumers' preference from the quantity sold for each item during the given period.

```r
item <- df %>% group_by(Category, Item) %>%
  summarise(qty_per_day = sum(Qty)/as.numeric((as.Date('2018-8-24') - min(Date)))) %>%
  arrange(desc(qty_per_day))


head(item, 10)%>% ggplot(., aes(x = reorder(Item, -qty_per_day), y = qty_per_day,
                                fill = Category)) +
  geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 30)) +
  xlab('Most Frequent Items') +
  ylab('Sale Quantity per Day') +
  ggtitle("Most Popular 10 Items") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 30))
```



```
tail(item, 10) %>% ggplot(., aes(x = reorder(Item, -qty_per_day), y = qty_per_day,
                                 fill = Category)) +
  geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 30)) +
  xlab('Least Frequent Items') +
  ylab('Sale Quantity per Day') +
  ggtitle("Least Popular 10 Items") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 30))
```



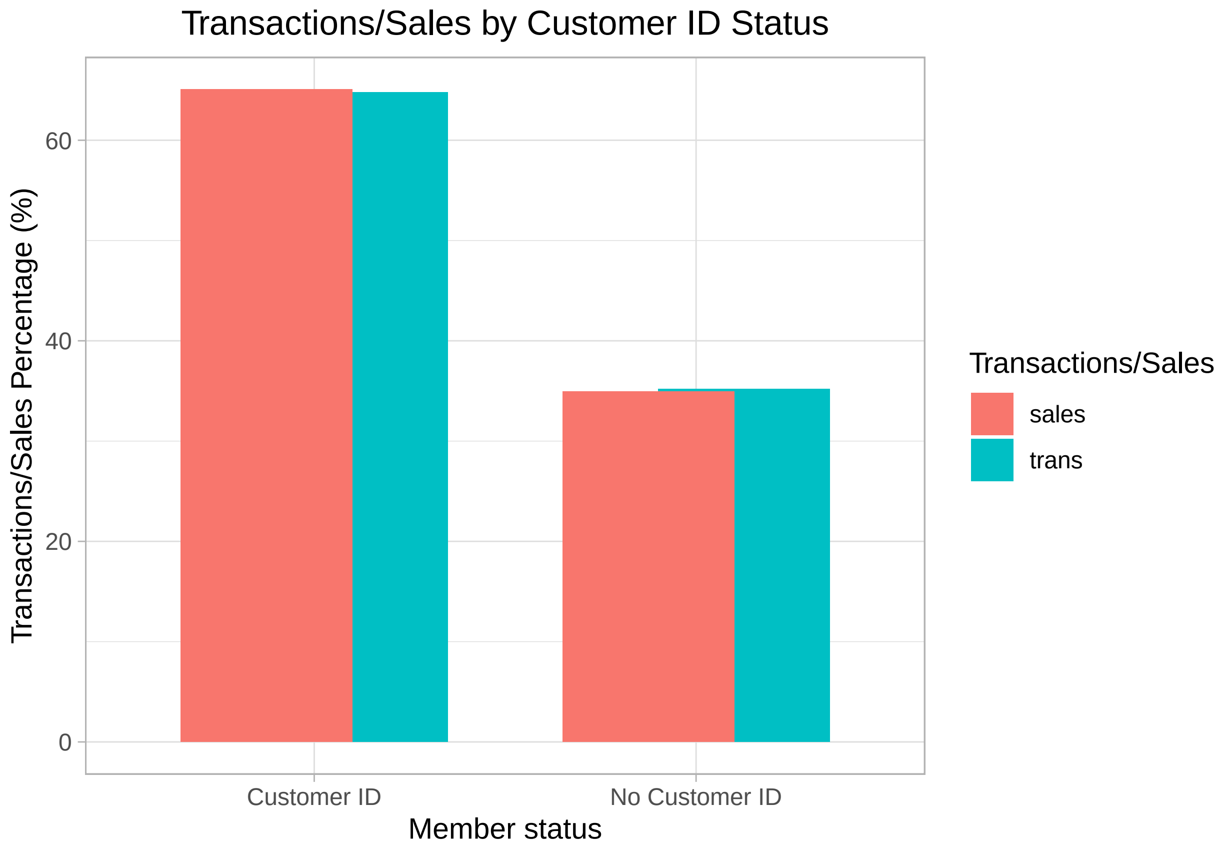
Least Frequent Items

The two graphs above show the most popular items and the least popular items. Coffee products, such as drip, latte, and some bread items in the food category, as well as ice, are the most popular items, while beans and some beers are the least purchased. This can help us identify which items should we focus on when developing recommendations and which items might be better to be removed from the menu.

Who Contributes to the most Transactions/Sales? Members or Non-members?

Since we noticed that the coffee shop has its own membership program, which is indicated by the "Customer.ID" column, if the "Customer.ID" is NA, then the transaction is from a non-member, otherwise from a member. We wanted to identify how important these consumers who are a member are relative to non-members. It can further support our assumption of focusing on member-consumers, as they are much easier to approach since the customer ID allows us to track transactions over time.

```
trans_member <- nrow(df[!is.na(df$Customer.ID),])/nrow(df)*100
sales_member <- sum(df[!is.na(df$Customer.ID),]$Gross.Sales)/sum(df$Gross.Sales)*100
m <- data.frame(c('Customer ID', 'No Customer ID'), c(trans_member, 100- trans_member),
                c(sales_member, 100- sales_member ))
colnames(m) <- c('status', 'trans', 'sales')
m <- m %>% gather(type, value, trans:sales)
ggplot(m, aes(status)) +
  geom_bar(aes(weight = value, fill = type), position = position_dodge(width = 0.5)) +
  labs(title= "Transactions/Sales by Customer ID Status",
       y="Transactions/Sales Percentage (%)",
       x = "Member status", fill="Transactions/Sales") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5))
```



It is obvious that consumers who are in the membership program are significantly more important than those who are not in, with over 60% of transactions and sales come from members.

Central Perk's Customer Base

Central Perk, a boutique coffee shop in New York, strongly believes that their business is mainly driven by "Loyal" customers and their business overall has been consistent across years. To understand our scope of the analysis and the validating the assumption, we dissected the customer purchase pattern using exploratory analysis.

As mentioned previously, of the customers who had visited Central Perk over the period of ~2-year timeframe, 65% of the overall transactions are derived by the customers who are identified with the "CustomerID". Without digging deeper, this looks promising and adds credence to the cafe's assumption of a loyal customer base.

But interestingly, upon bucketing the customers based on their transaction behavior instead of CustomerID, we saw that almost 95% of this cohort had visited on less than 6 days over the entire 2-year timeframe. This was a major red flag given our initial assumption that customers with a valid Customer ID are treated as loyal customers, but in reality, they were highly infrequent and their purchase pattern is almost similar to the customers without a customer ID. Within the customers with a Customer ID, a very small cohort representing ~5%(1.7K users) generated almost 46% of the total sales.

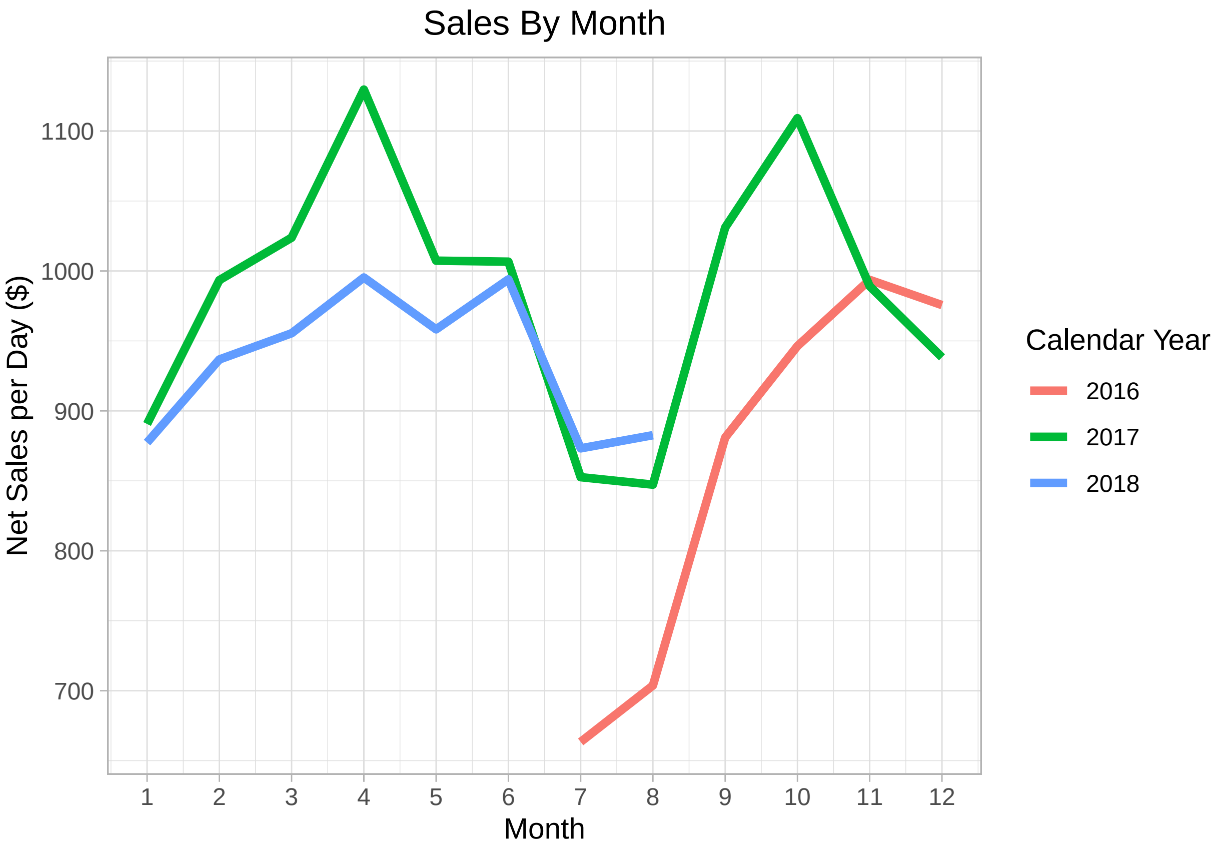
Based on our findings, this small cohort represents the truly loyal customers and they are just a minuscule portion of the overall customers and the rest of the segments should be treated on par with non-loyal members who do not have a customer ID because of their poor purchase behavior.

```
data <- df
data$paste <-  paste(data$Date, data$Time)
```

```
data <- rename(data, customer_id = Customer.ID)
data1 <- data %>% filter(!is.na(customer_id) ) %>% group_by(customer_id) %>%
  summarise(count = n_distinct(paste), qty = sum(Qty), net_sales = sum(Net.Sales)) %>%
  arrange(desc(count))
data1$Flag <- ifelse(as.integer(data1$count) >= 1
                     & as.integer(data1$count) <= 3,'1-3 Transactions',
                     ifelse(as.integer(data1$count) >=4
                            & as.integer(data1$count) < 7,'4-6 Transactions',
                            'More than 6 Transactions'
                     ))

data2 <- data1 %>% group_by(Flag) %>% summarise(no_of_users =  n_distinct(customer_id),
                                                tot_sales = sum(net_sales),tot_qty = sum(qty)) %>%
  mutate(percentage_users =  no_of_users/ sum(no_of_users)*100,
         percentage_qty = tot_qty/sum(tot_qty)*100,
         percentage_sales = tot_sales / sum(tot_sales)*100)
```

After testing the loyal customer base assumption, we focused on the general growth of the café business as the Central Perk believes that they are fairly consistent across years and coupling the truly loyal customers with the growth of the business helps in understanding whether the marketing tactics should be focused on just retaining that group or focusing on cross-selling to that group to boost demand. Café business as in any other business is highly influenced by seasonality and because of the fragmented datasets across years, we compared 2016 months vs. 2017 (Aug- Dec) and 2017 vs. 2018(Jan - Aug) to achieve better comparison results.

```
data$year <- year(as.Date(df$Date, format = "%d-%m-%Y"))
data$month <- month(as.Date(df$Date, format = "%d-%m-%Y"))

data3 <- data %>% group_by(year, month) %>% summarise(days = n_distinct(paste(Date))
                                                      ,tot_sales = sum(Net.Sales),
                                                      tot_qty = sum(Qty)) %>%
  mutate(sales_per_day = tot_sales / days, qty_per_day = tot_qty / days)

data3$year <- as.factor(data3$year)

ggplot(data = data3, aes(x = month, y = sales_per_day, color = year)) +
  geom_line(size = 1.5) +  scale_x_continuous(breaks = 1:12) +
  labs(title="Sales By Month", y="Net Sales per Day ($)", x="Month", col="Calendar Year") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5))
```



Overall, we observe that the Central Perk had a strong performance in 2017 in comparison to both 2016 and 2018. This is particularly interesting because the demand/revenue generation is not consistent across years as what was believed initially. Further, we trifurcated the dataset based on the number of visits by customers (customers who had visited on 1-3 days, 4-6, and 7+), to understand whether any particular segment had a strong/weak performance.

Net Sales Growth Rate	2016 - 2017	2017 - 2018
1-3 Days of Transactions	-0.3%	11.1%
4-6 Days of Transactions	14.5%	11.9%
7 or More Days of Transactions	11.4%	-5.8%

During the 2016-2017 time frame, both the core loyal cohort (>6 visits) and 4-6 visits cohorts had increased revenue generation of ~13% year over year for the same months and were primarily responsible for the strong performance in 2017. However, the overall net sales growth was not sustained in 2018, sales YoY remained fairly flat or witnessed a slight decline. The core loyal cohort (>6 visits) was the only declining cohort with their YoY growth declining at ~6%. Though the infrequent customers did grow in 2018, the decline of the core loyal group hurt the overall sales for Central Perk. Given the importance of this cohort, in terms of revenue generation for Central Perk, it is imperative we stop this decline and boost sales by leveraging various marketing tactics detailed below.

Why Focusing on Loyal Customers?

After understanding the whole customers' buying pattern, we decided to focus on our most loyal group of customers, i.e. who purchased on more than six days in the past two years. The justification of focusing them are listed below:

```r
# count of unique days customers visited the shop
customer_counts <- df %>% select(Date, Customer.ID) %>%
                          unique() %>%
                          filter(!is.na(Customer.ID)) %>%
                          group_by(Customer.ID) %>%
                          summarise(trans = n())

table(customer_counts$trans)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 24728  2935  1147   620   383   284   204   167   151   123    92    85
##    13    14    15    16    17    18    19    20    21    22    23    24
##    72    60    59    54    50    30    29    32    36    29    36    20
##    25    26    27    28    29    30    31    32    33    34    35    36
##    14    16    17    23    13    12    12    10    11    11    16     5
##    37    38    39    40    41    42    43    44    45    46    47    48
##    10    15    11    10     4     6     5     9     2     2     3     2
##    49    50    52    53    54    55    56    57    58    59    60    61
##     5     5     3     6     6     4     3     4     2     2     1     4
##    62    63    64    65    66    68    69    70    71    72    73    75
##     1     2     2     2     3     1     2     3     2     2     3     1
##    77    80    81    82    83    84    85    86    87    88    89    90
##     1     2     2     1     2     5     2     1     1     1     4     1
##    91    93    95    96    97    98    99   100   101   103   106   107
##     2     3     2     2     1     1     1     1     2     1     1     2
##   108   110   112   113   115   116   118   120   121   132   134   142
##     1     1     1     1     1     1     1     1     1     1     1     2
##   145   146   148   151   155   157   171   181   186   192   197   198
##     1     1     1     1     1     1     1     2     1     1     1     1
##   203   207   216   250   257   280   338   351
##     1     1     1     2     1     1     1     1
```

```r
# evaluate customers who visited a certain number of days and find what percentage
# of total customers they make up compared to what percentage of sales they account for

totals <- df %>% filter(!is.na(Customer.ID))
total_net <- sum(totals$Net.Sales)
total_qty <- sum(totals$Qty)
total_custs <- nrow(customer_counts)

# customers who visited more than 6 days
ids <- customer_counts[customer_counts$trans > 6, 1]
df_more_than3 <- df[df$Customer.ID %in% ids$Customer.ID,]
nrow(customer_counts[customer_counts$trans > 6,]) / total_custs
```

```
## [1] 0.05388073
```

```r
sum(df_more_than3$Net.Sales)/total_net
```

```
## [1] 0.4648011
```

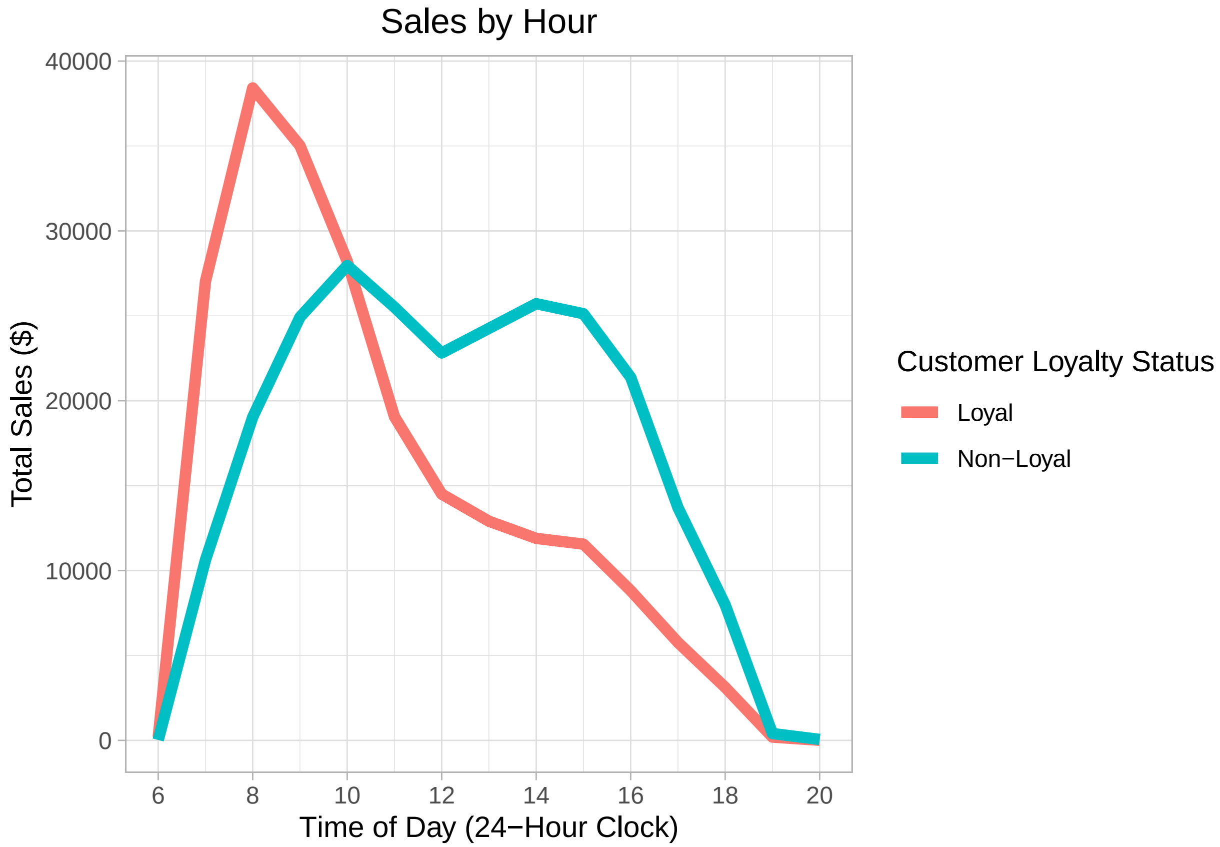
```r
sum(df_more_than3$Qty)/total_qty
```

```
## [1] 0.4559676
```

1. From the analysis above, we can see they are only 5.4% of the total customers with a customer ID but

they contributed 46.5% of the total sales, which means that they are the most valuable customers to our business.

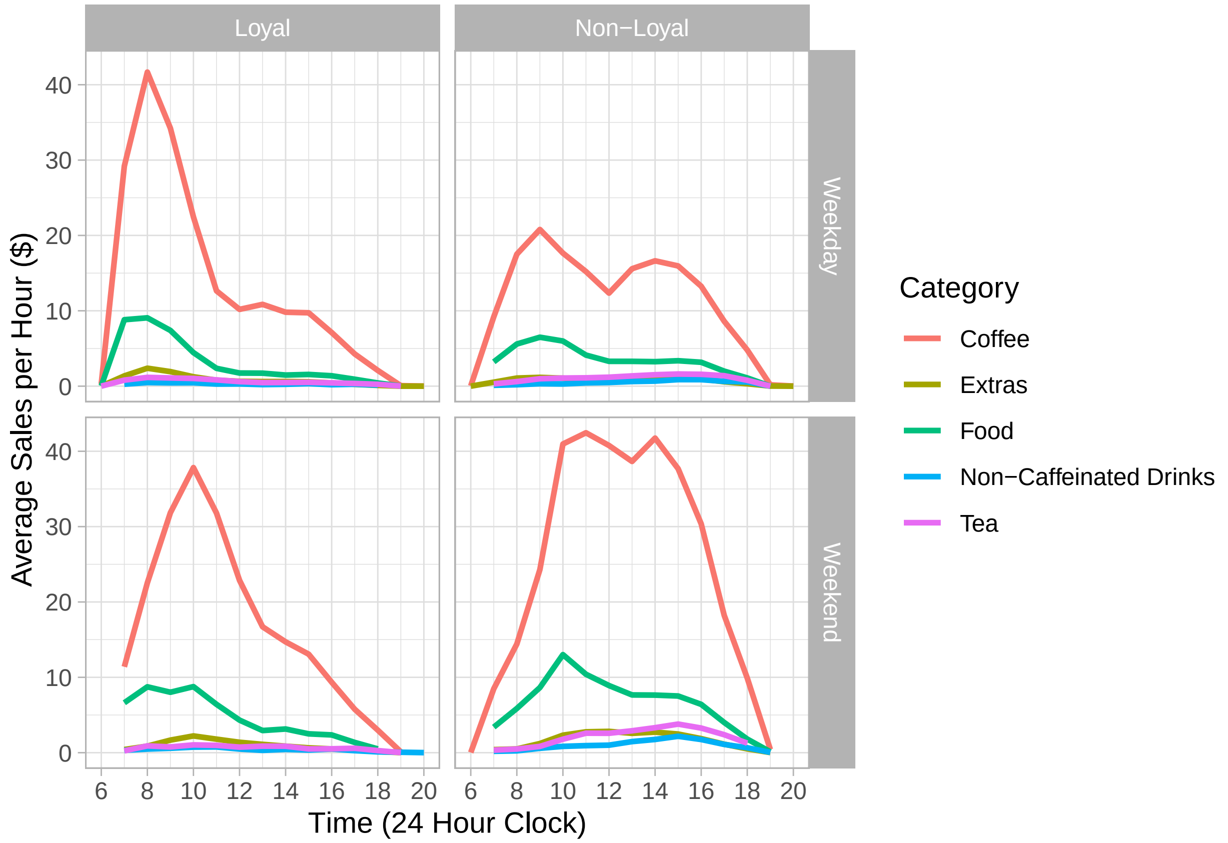
```
df<-merge(x=df,y=customer_counts,by="Customer.ID")
df$loyalty <- ifelse(df$trans > 6, "Loyal", "Non-Loyal")


df %>% filter(!is.na(Customer.ID)) %>%
  group_by(hour(hms(Time)), loyalty) %>%
  summarise(tot_sales = sum(Net.Sales)) %>%
  ggplot(aes(x = `hour(hms(Time))`, y = tot_sales, col = loyalty)) +
  geom_line(size = 2) + scale_x_continuous(breaks = seq(6, 20, by=2)) +
  labs(title= "Sales by Hour", y="Total Sales ($)", x = "Time of Day (24-Hour Clock)",
       col="Customer Loyalty Status") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
df$weekday_weekend = ifelse(weekdays(ymd(df$Date)) %in% c('Saturday', 'Sunday'),
                            'Weekend', 'Weekday')

categs <- df %>% filter(!is.na(Customer.ID)) %>%
  group_by(Category, loyalty, hour(hms(Time)), weekday_weekend) %>%
  summarise(tot_sales = sum(Net.Sales))

# there are more weekdays than weekend days so need to average

df %>% select(Date, weekday_weekend) %>%
  unique() %>%
  group_by(weekday_weekend) %>%
  summarise(count = n())
```

```
## # A tibble: 2 x 2
##   weekday_weekend count
##   <chr>           <int>
## 1 Weekday           541
## 2 Weekend           217
```

```
# 541 weekdays, 217 weekend days

categs[categs$weekday_weekend == 'Weekday', 5] <- categs[categs$weekday_weekend
                                                          == 'Weekday', 5]/541
categs[categs$weekday_weekend == 'Weekend', 5] <- categs[categs$weekday_weekend
                                                          == 'Weekend', 5]/217


categs %>%
  filter(Category %in% c('Coffee', 'Food', 'Tea', 'Non-Caffeinated Drinks', 'Extras')) %>%
  ggplot(aes(x=`hour(hms(Time))`, y = tot_sales, col = Category)) +
  geom_line(size = 1) +
  scale_x_continuous(breaks = seq(6, 20, by=2)) +
  theme_light() +
  facet_grid(vars(weekday_weekend), vars(loyalty)) +
  labs(y="Average Sales per Hour ($)", x = "Time (24 Hour Clock)")
```



2. From the analysis above, we can see the buying patterns for the loyal customers are extremely right-skewed compared to the customers who visited on 6 or fewer days, which means most of their transactions were done in the morning periods. This is constant across weekdays and weekends and is consistent

for both food and coffee. Since our goal is to smooth the demand across the day, we could start at boosting their demand in the afternoon and evening periods.

3. Since they came more than six times in the past few years, we **assume** they had a better understanding of the products and already developed loyalty to the coffee shop. Therefore, it is much easier to convert them to purchase more in the afternoon and evening.
4. Since most of the people in New York take the subway or walk to commute, people don't drive that much to get what they want, we **assume** that these people live or work near the coffee shop. Hence, they could visit the shop easily even in the afternoon or evening.
5. As they are already a part of our loyalty program network since they have a customer ID, we **assume** the shop has more personal information about them so that we could access them much easier compared to other customers. Apart from that, they have a higher chance to open marketing messages such as email or direct mails.

Association Rules to Boost Sales from Loyal Customers

With all these justifications, we choose two time periods to be our first target to boost sales, one from 11:00 to 14:00 and another from 16:00 to 19:00. We chose these time periods based on our previous assumptions that they live or work nearby our coffee shop into consideration. These two three-hour time periods are when they will have a lunch break and after work, so that they have time to stop by our shop. We ran association rules on the whole dataset to gain insights about what are the most popular product bundles, and we further try to bundle these products together with sales promotions to attract these loyal customers to come back within our targeted time period.

```r
df4$Time <- as.character(df4$Time)
associate_a <- df4 %>%
  filter(Category != "Extras") %>%
  filter(Time >= "11:00:00" & Time <="14:00:00") %>%
  group_by(Customer.ID, Date, Time) %>%
  select(Customer.ID, Date, Time, Item) %>%
  summarise(Purchase=paste(Item, collapse=","))
s_a <- strsplit(associate_a$Purchase, split = ",")
trans_a <- as(s_a, "transactions")
rule_a <- apriori(trans_a, parameter = list(supp = 0.005, conf = 0.001, minlen = 2))
```

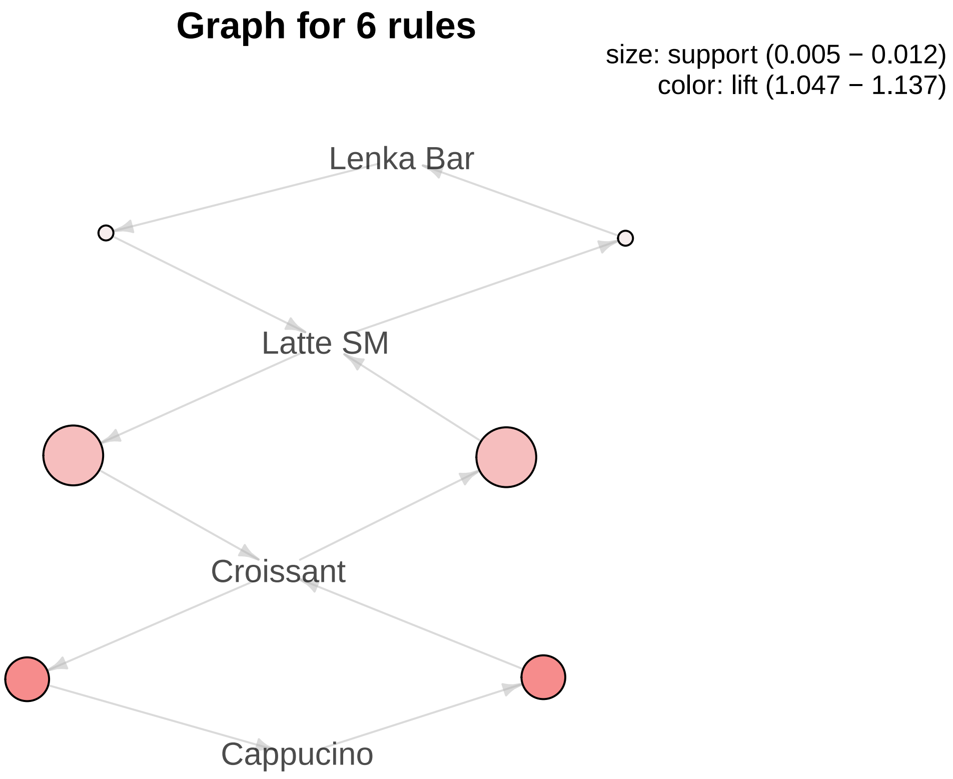
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##       0.001    0.1    1 none FALSE            TRUE       5   0.005      2
##  maxlen target  ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 170
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[58 item(s), 34106 transaction(s)] done [0.00s].
## sorting and recoding items ... [22 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [30 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rule_a <- sort(rule_a, by = "lift", decreasing = TRUE)
inspect(rule_a[0:10])
```

```
##      lhs            rhs         support     confidence lift      count
## [1]  {Cappucino} => {Croissant} 0.009353193 0.07001756 1.1371518 319
## [2]  {Croissant} => {Cappucino} 0.009353193 0.15190476 1.1371518 319
## [3]  {Croissant} => {Latte SM}  0.011522899 0.18714286 1.1014140 393
## [4]  {Latte SM}  => {Croissant} 0.011522899 0.06781708 1.1014140 393
## [5]  {Lenka Bar} => {Latte SM}  0.005453586 0.17782027 1.0465467 186
## [6]  {Latte SM}  => {Lenka Bar} 0.005453586 0.03209664 1.0465467 186
## [7]  {Donut}     => {Latte SM}  0.012197267 0.16720257 0.9840571 416
## [8]  {Latte SM}  => {Donut}     0.012197267 0.07178602 0.9840571 416
## [9]  {Donut}     => {Cappucino} 0.009587756 0.13143087 0.9838852 327
## [10] {Cappucino} => {Donut}     0.009587756 0.07177349 0.9838852 327
```

We only need bundles with a lift higher than 1, so we only choose the top six rules. Within them, we could derive three most popular product bundles, namely Cappuccino and Croissant, Croissant and Latte, Lenka Bar and Latte.

```
plot(rule_a[0:6],method="graph",cex = 1)
```



```
associate_e <- df4 %>%
  filter(Category != "Extras") %>%
  filter(Time >= "16:00:00" & Time <="19:00:00") %>%
  group_by(Customer.ID, Date, Time) %>%
  select(Customer.ID, Date, Time, Item) %>%
  summarise(Purchase=paste(Item, collapse=","))
```

```
s_e <- strsplit(associate_e$Purchase, split = ",")
trans_e <- as(s_e, "transactions")
rule_e <- apriori(trans_e, parameter = list(supp = 0.005, conf = 0.001, minlen = 2))
```

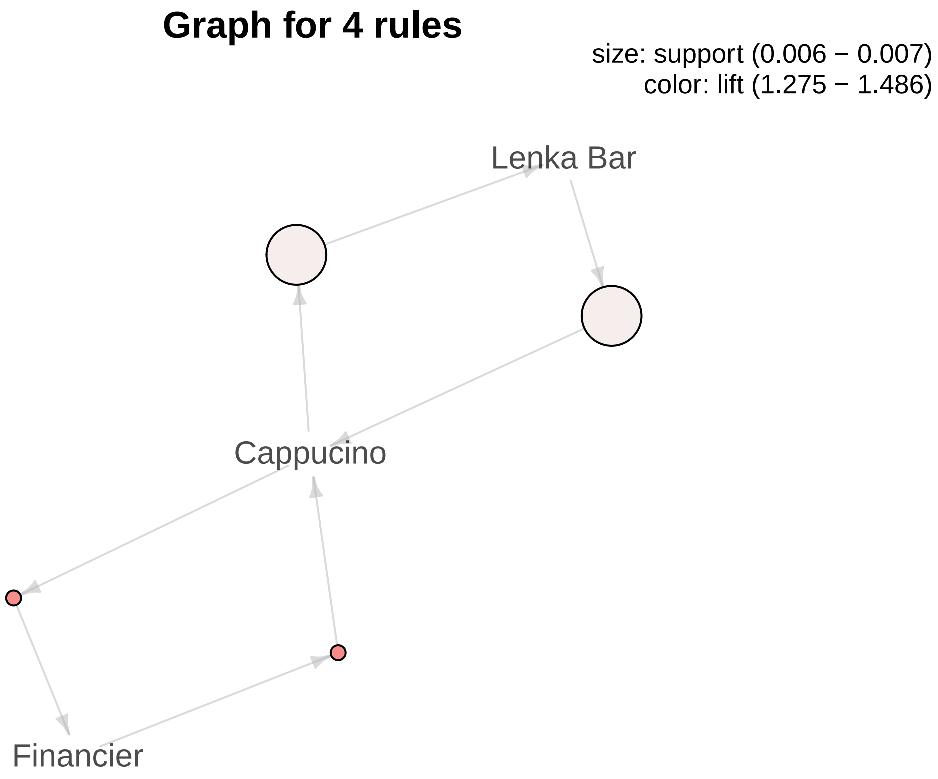
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##       0.001    0.1    1 none FALSE            TRUE       5   0.005      2
##  maxlen target  ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 93
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[61 item(s), 18691 transaction(s)] done [0.00s].
## sorting and recoding items ... [21 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [26 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rule_e <- sort(rule_e, by = "lift", decreasing = TRUE)
inspect(rule_e[0:10])
```

```
##      lhs             rhs          support     confidence lift      count
## [1]  {Cappucino} => {Financier} 0.005671179 0.04372937 1.4860831 106
## [2]  {Financier} => {Cappucino} 0.005671179 0.19272727 1.4860831 106
## [3]  {Lenka Bar} => {Cappucino} 0.006848216 0.16537468 1.2751725 128
## [4]  {Cappucino} => {Lenka Bar} 0.006848216 0.05280528 1.2751725 128
## [5]  {Donut}     => {Latte SM}  0.012465893 0.15041963 0.9674787 233
## [6]  {Latte SM}  => {Donut}     0.012465893 0.08017894 0.9674787 233
## [7]  {Donut}     => {Americano} 0.005778182 0.06972240 0.8618925 108
## [8]  {Americano} => {Donut}     0.005778182 0.07142857 0.8618925 108
## [9]  {Donut}     => {Drip SM}   0.018404580 0.22207876 0.8546169 344
## [10] {Drip SM}   => {Donut}     0.018404580 0.07082561 0.8546169 344
```

We only have two useful product bundles here, namely Cappucino and Financier, Lenka Bar and Cappucino.

```
plot(rule_e[0:4],method="graph",cex = 1)
```



Bundling

We will offer the above bundling to our loyal customers, and they can redeem the offer within the given timeframe, i.e. 11:00 to 14:00 and 16:00 to 19:00. We will bring this marketing message to those customers by email and text messages, and we will keep track of the link open rate and promotion code conversion rate. There will also be in-shop advertising posters to remind the customers.

Even though some may argue providing the most popular product bundles actually create costs to our business as people will anyhow buy these products, our group focuses more on the long-term benefits of this practice. What we are trying to do is to transfer loyal customers' buying behavior and get a smoothing demand curve across the day. Thus, in the short-term, we may not be beneficial, but in the long run, this certainly helps us reduce wasteful staffing costs.

Does Offering Bundle Sales Only to Loyal Customers Make Sense?

We only plan to offer these product bundlings to our loyal customers. We believe this makes better sense compared to offering to all customers because:

1. This is a common practice in the business world. Customers who are already a part of the business's loyalty program could have more benefits compared to random guests. From the loyal customers' perspective, this would make them feel unique and further deepen their preference for our shop.
2. This will make non-loyalty members see the benefits of membership and increase their desire to become members. Only when they see the value of membership are they more likely to convert to loyalty members and bring more profits to our shop.
3. Also, from the graph presented in the previous analysis, we can see the buying pattern of non-loyal customers is much smoother than loyal customers. Therefore, there is no need to shift their demand line.

4. Bundling creates a cost to our business, we may want to focus on a small group of people first to test the outcome and then come up with better ones to generalize to public customers.

Staff Shifting and Optimization

One of the major benefits of smoothing our demand curve is related to expected savings on staffing costs. Since it's not practical to make employees work a two-hour shift during peak demand, we prefer to bring more customers to our shop later in the day to utilize an employee's full shift. Hopefully, with bundling, we will be able to boost sales in the afternoon up to where they are in the morning and increase the sales volume in the evening to a position similar to that in the afternoon. With demand smoothing, we could optimize our staff allocation and reduce wasted costs.

Final Takeaways

- Not all customers who have enrolled for the membership program and have a Customer ID are loyal customers. Of the enrolled customers, only 5% contribute ~46% of the overall sales and they are identified as truly loyal customers. This segment which drives most of the sales has witnessed a slump in 2018, pulling down the overall sales.
- We treat customers who have come to our shop more than six times in the past few years as our loyal customers and offer them incentives to come again in the afternoon and evening.
- According to association rules, we could offer three bundle pricing strategies of Cappuccino 'and Croissant, Croissant and Latte, and Lanka Bar and Latte from 11:00 to 14:00. Two bundle sales of Cappuccino and Financier and Lenka Bar and Cappuccino will be offered from 16:00 to 19:00.