Homework 3 An Exploration of Sales Data For CentralPark

Vijay R Dhulipala
Qian Fu
Yidan Gao
Arivarasu Perumal
Yi-Li Yu
Chuchen Xiong

20th October 2019

Contents

Background and The Business Problem

Central Park is a boutique coffee shop located in the heart of New York City. It serves a high-quality selection of coffee, food, tea and etc for customers. They believe their current customer base is fairly loyal and revenue are generating steadily year over year. Now, as the size of its customer base has reached more than 30K+, the store would like to explore their sales data and have a sense of their demands and customers. To make itself set apart from others in a unique way, Central Park aims to put a priority on drawing actionable insights to normalize demands and increase revenue.

Hired as the data scientist consultants, our motivation is to better understand the customer purchase patterns of volume and items sold in the store, confirm whether their customer base is loyal with support from sales data, and smooth demands out over time.

Our Task

Tasked by Central Park, our job is to normalize demand and generate additional revenue by finding appropriate gap in the business and improve the business operation of our client.

Our Approach

To better drive the analysis, we narrow down our focus into two specific questions. Firstly, we are interested in understanding the sales patterns across items, categories, time (hourly, daily, weekly), and how could we smooth demand out.

To address the first question, we explore data and draw plots to demonstrate interesting sales patterns, and furthermore apply the association rule to identify the appropriate items that could be bundled together to boost sales.

In terms of our second question to examine the loyalty of customer base, we do a clustering analysis on the customer-level data including key features like purchase frequency, recency, total sales amount and etc. In this way, we could furthermore explore which segments are suitable for converting into loyal customers and offers recommendations accordingly.

Understanding Data

Data Filtering and Cleaning

Load dependencies

```
# Importing reqired libraries
library(ggplot2)
theme_set(theme_classic())

library(reshape2) ; library(ggrepel) ; library(RSQLite)
library(XML) ; library(knitr) ; library(dplyr)
library(naniar) ; library(gridExtra) ; library(magrittr)
library(lubridate) ; library(RSQLite) ; library(tidyr)
library(data.table) ; library(RColorBrewer) ; library(stringr);
library(clustMixType) ;library(cluster); library(cowplot)
library(fpc);library(factoextra);library(xts)
library(factoextra);library(arules);library(arulesViz)
# My theme
my_theme <- theme_classic() +
    theme(text=element_text(size=9,  family="serif"))
```

Reading Data

```
cp2016 <- read.csv('Central Perk Item Sales Summary 2016.csv')
cp2017 <- read.csv('Central Perk Item Sales Summary 2017.csv')
cp2018 <- read.csv('Central Perk Item Sales Summary 2018.csv')
centralPark <- rbind(cp2016,cp2017,cp2018)
```

Data Glimpse

```r
# Look at the data types and head values in every column
glimpse(centralPark)
```

```
## Observations: 221,561
## Variables: 13
## $ Date             <fct> 12/31/17, 12/31/17, 12/31/17, 12/31/17, 12/31...
## $ Time             <fct> 16:57:33, 16:37:29, 16:33:08, 16:33:08, 16:31...
## $ Category         <fct> Coffee, Tea, Coffee, Coffee, Coffee, Food, Co...
## $ Item             <fct> Cappucino, Tea SM, Cappucino, Espresso, Espre...
## $ Qty              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ Price.Point.Name <fct> Regular, SM, Regular, Regular, Regular, Regul...
## $ Gross.Sales      <fct> $3.90 , $2.30 , $3.90 , $3.22 , $3.22 , $4.13...
## $ Discounts        <fct> $0.00 , $0.00 , $0.00 , $0.00 , $0.00 , $0.00...
## $ Net.Sales        <fct> $3.90 , $2.30 , $3.90 , $3.22 , $3.22 , $4.13...
## $ Tax              <fct> $0.35 , $0.20 , $0.35 , $0.28 , $0.29 , $0.37...
## $ Notes            <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ Event.Type       <fct> Payment, Payment, Payment, Payment, Payment, ...
## $ Customer.ID      <fct> NA, NA, 2a90ca0a266d3e357b693cc366e59e9156972...
```

What does the data look like? • 200K+ records containing a span of two years' historical sales data • For the customerID: 31821 unique customerID and 78077 NA records • Transaction-level details about sales include date, time, category, item, quantity, discount, net sales and etc. • Transaction-level details about customers, such as customerid

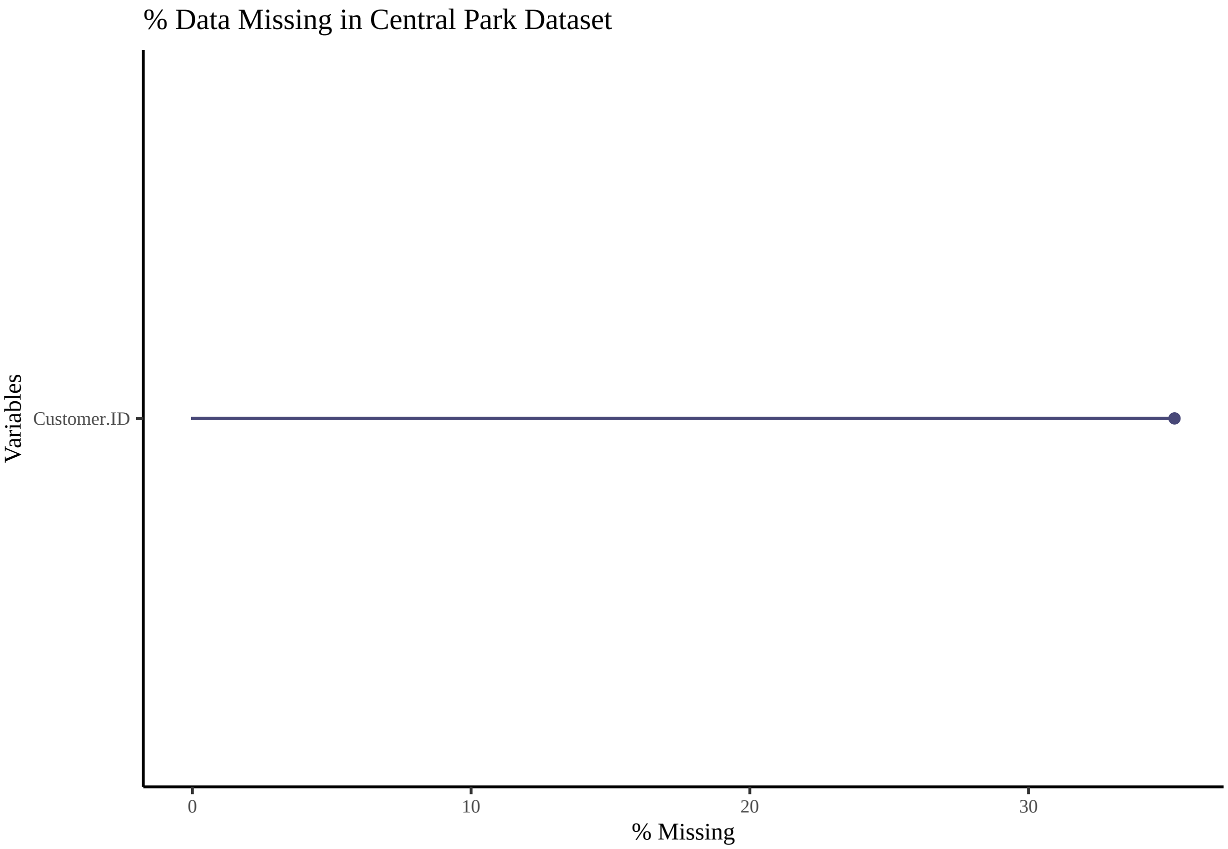
Data Transformation Firstly, there is an unknown error in the field of "Date" column, so we remove the invalid row from the dataset. Then, we create a new column "DateTime" combining date and time together as the unique identifier for each transaction, and converted the Date and Datetime string into standard data formats. For the discount, sales and tax columns, we remove the "$" in the field and transform the string into numeric format for the convenience of further analysis. Some fields in the item column contain both itemname and the size of itme like 'LG' and 'SM', therefore we clean the item column and keep the original itemname.

```r
# Filter out errors
centralPark <- centralPark %>% filter(Date != 'Unknown Error')

# Data Conversion
centralPark$DateTime <- paste(as.character(centralPark$Date) , as.character(centralPark$Time))
centralPark$Date <- mdy(centralPark$Date)
centralPark$DateTime <- mdy_hms(centralPark$DateTime)
centralPark$Hour <- hour(centralPark$DateTime)
centralPark$Year_month <- paste(format(centralPark$Date, "%Y") , format(centralPark$Date, "%m"), sep =
centralPark$Item <- as.character(centralPark$Item)
centralPark$Item <- as.factor(str_replace_all(word(centralPark$Item), c('LG', 'SM'), ''))


fixData <- function(x)
{
  x <- gsub('[$,)]', '', x)
  x <- gsub('\\(', '-', x)
  x <- as.numeric(x)
  return(x)
}
```

```
centralPark$Discounts <- sapply(centralPark$Discounts , fixData)
centralPark$Gross.Sales <- sapply(centralPark$Gross.Sales , fixData)
centralPark$Net.Sales <- sapply(centralPark$Net.Sales , fixData)
centralPark$Tax <- sapply(centralPark$Tax , fixData)
centralPark$Membership <- ifelse(is.na(centralPark$Customer.ID),'Non Member','Member')
```

Exploring data for erroneous and missing values

```
gg_miss_var((centralPark)[colSums(is.na(centralPark)) > 1], show_pct = TRUE) +
  labs(title = "% Data Missing in Central Park Dataset") + my_theme
```



We observed that more than 30% of customerID are missing. A new column "Membership" (member vs. non-member) has been created based on whether there is valid value in the customerID column. For those who have recorded the customerId, they are regarded as the members of this coffee shop.

```
table(centralPark$Membership)
```

```
##
##     Member Non Member
##     143484      78076
```

We found that 64.8% of customers are members and the rest of them are customers with no customerid, who are mon-members.

```r
table(centralPark$Category)
```

```
##
##                   Beans                   Beers                  Cereal
##                    1468                     303                     546
##                  Coffee                  Extras                    Food
##                  129648                   46022                   27852
## Non-Caffeinated Drinks                    None                     Tea
##                    4853                     119                   10749
##
##                       0
```

From this graph, we could catch a gmplise of all items sold in the coffee shop, including beers, coffee, drinks, cerals and etc, which could help us understand the business of coffee shop better.

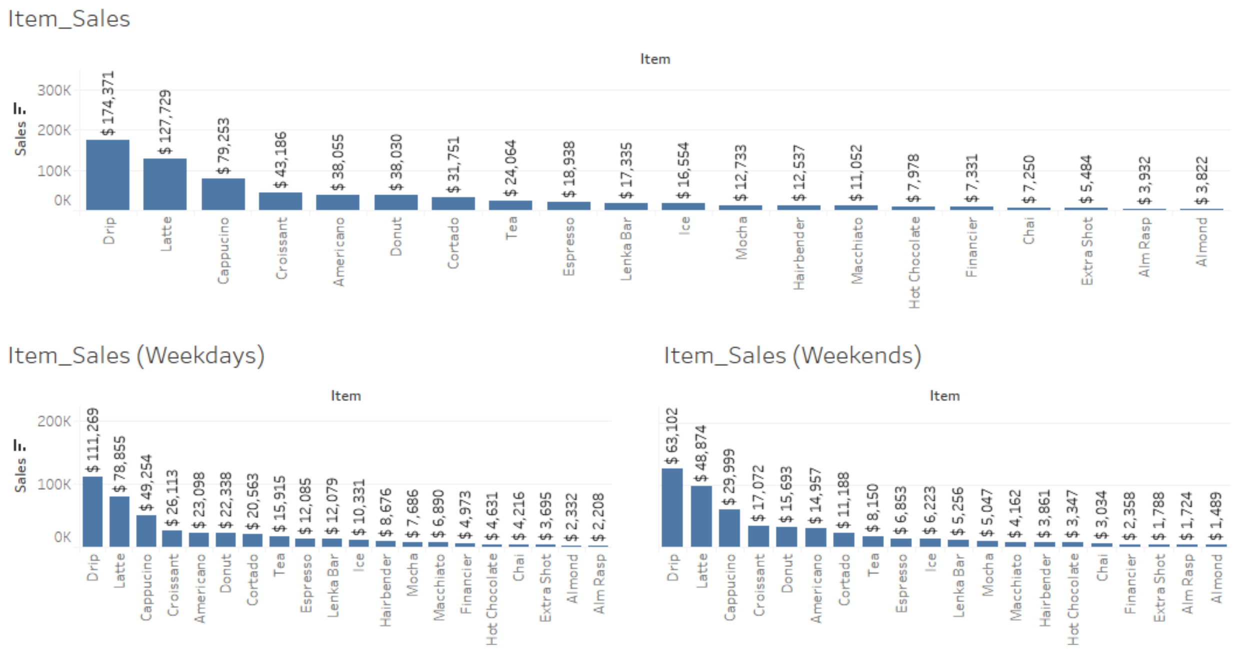
An Overview of Sales Patterns in the CentralPark Coffee Shop

Sales patterns accross different items and categories

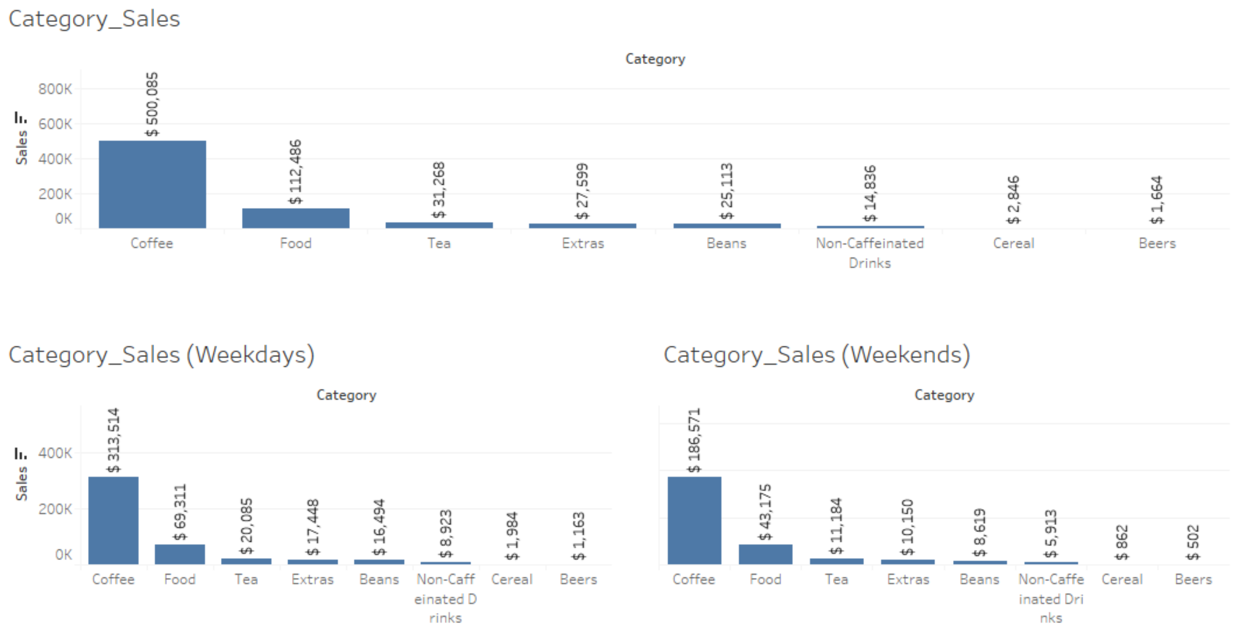
Description and Rationale for the Chosen Analysis In this part, we would like to demonstrate some sales patterns to figure out which item and category contribute the most to sales of this coffee shop. Therefore, we draw some exploratory plots showing the top selling items on weekdays and weekends.

Execution and Results (without code)
[Graphs generated from Tableau]

```r
knitr::include_graphics('Item_Sales.png')
```



```r
knitr::include_graphics('Category_Sales.png')
```



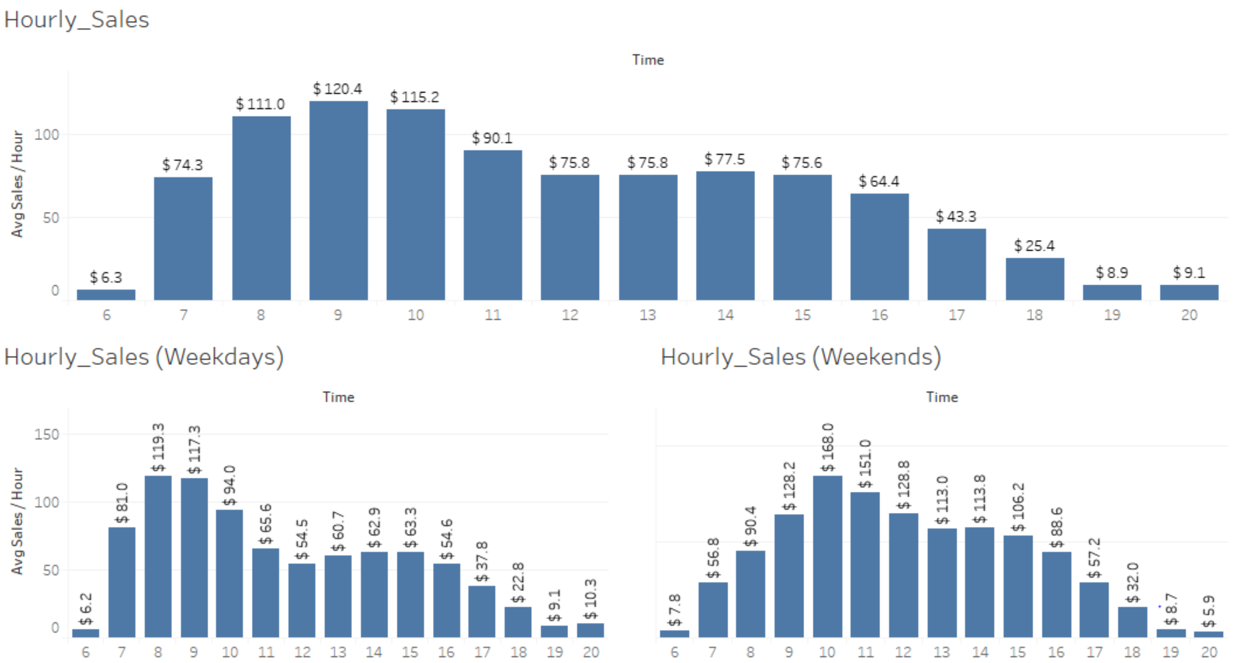
Interpretation
Above graphs clearly show that Drip, Latte and Cappucino contribute the most to sales all the time, which are much higher than other items sold in the coffee shop. Furthermore, Coffee and food are two main categories accounting for a large proportion of sales on both weekdays and weekends.

Variations in the pattern of the sales across hours, days and months

Description and Rationale for the Chosen Analysis
As one of our goals is to understand the general colume distribution based on the historical sales data, we draw some plots to present the variation of sales in terms of hour, day and week. In this way, we can have a better idea of customers' buying patterns accross different time periods.

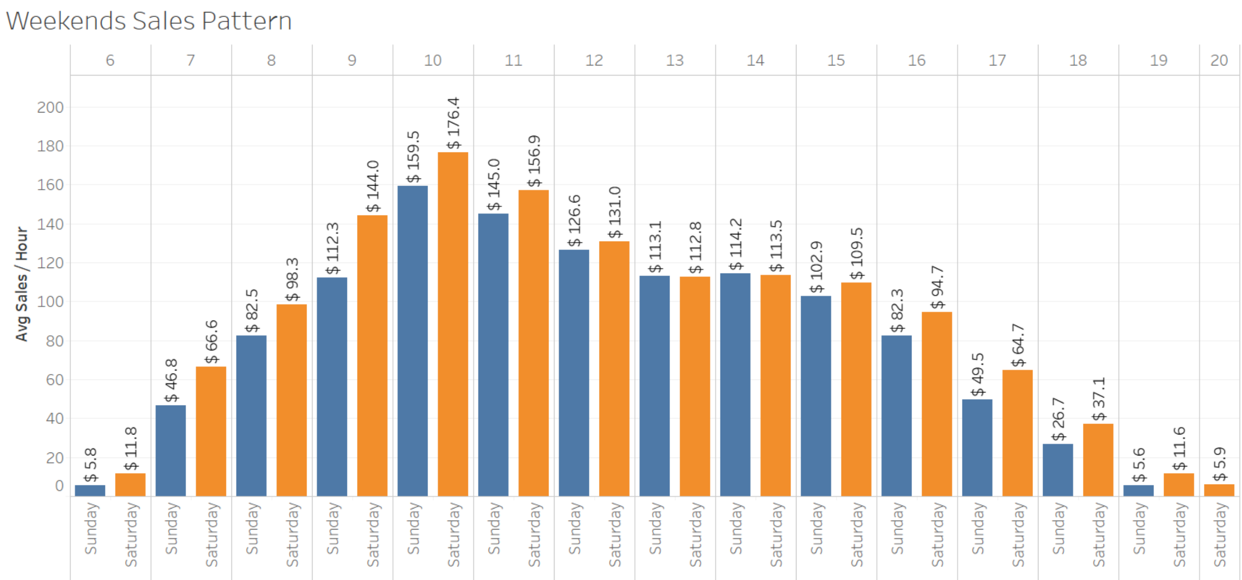
Execution and Results (without code)

```r
knitr::include_graphics('Hourly_Sales.PNG')
```



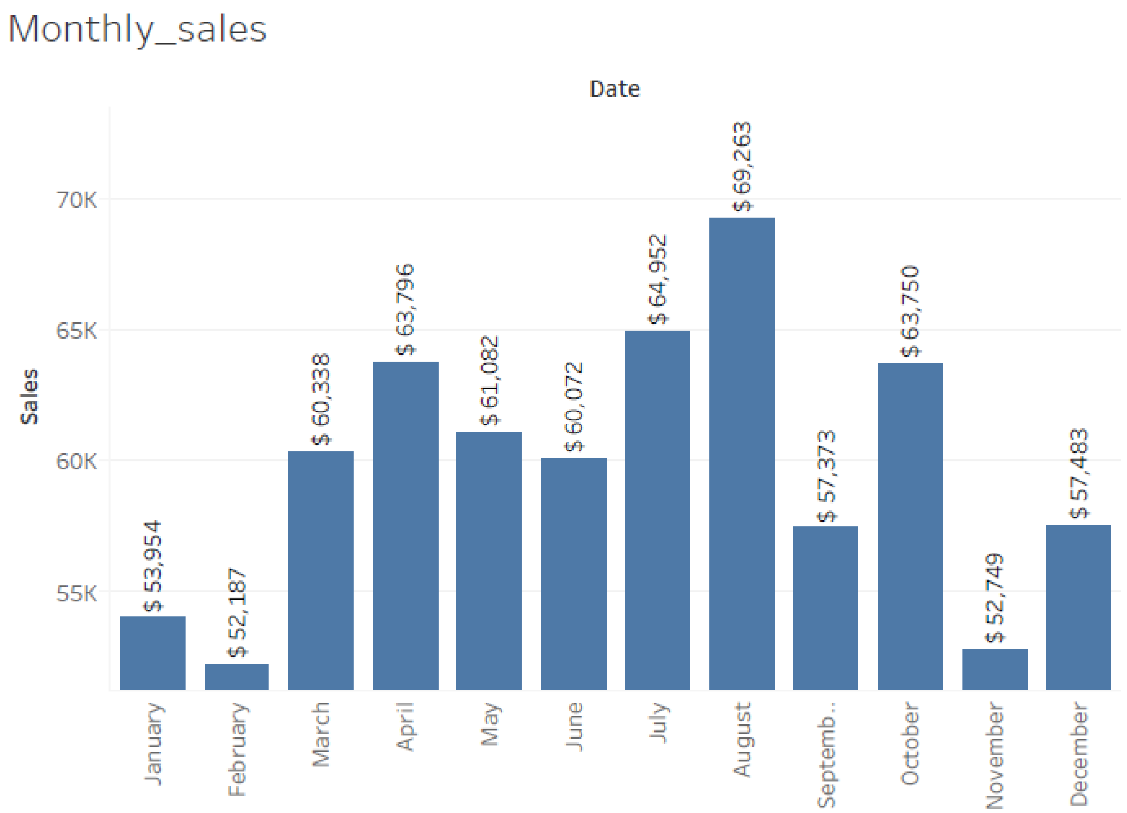
Interpretation The above graphs clearly show the 8-11am in the morning is the peak time when store makes the most sales on average among its operation hours (6am – 9pm) each day. After 3pm, the average hourly sales start to decline till its closing. For weekdays, the peak time is 8-10am, even earlier than the peak time of weekends about 10am in the morning. On both weekdays and weekends, the average hour sales drop in the afternoon and evening. Most importantly, there is a small increase in the noon time during weekdays, which is different from the purchase patterns on weekends.

```
knitr::include_graphics('Sat and Sun sales pattern.PNG')
```



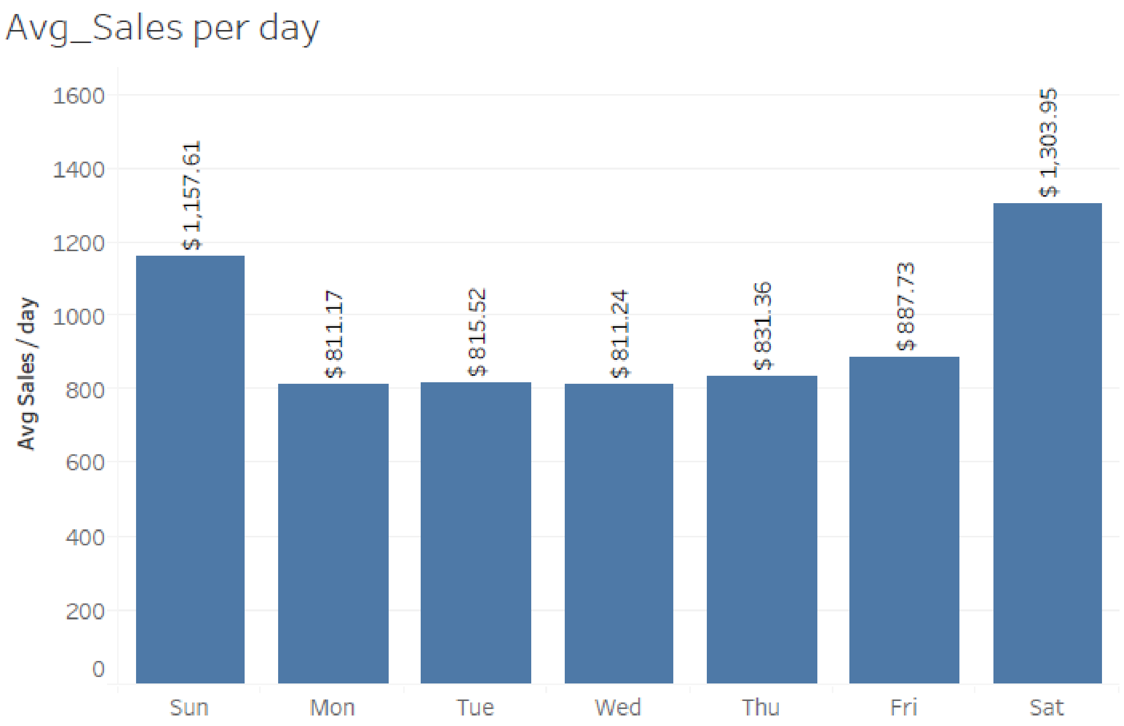
Interpretation
Furthermore, this graph illustrates that shifting peak time pattern (at about 10am) appear on both Saturaday and Sunday.

```r
knitr::include_graphics('Avg_Monthly_Sales.PNG')
```



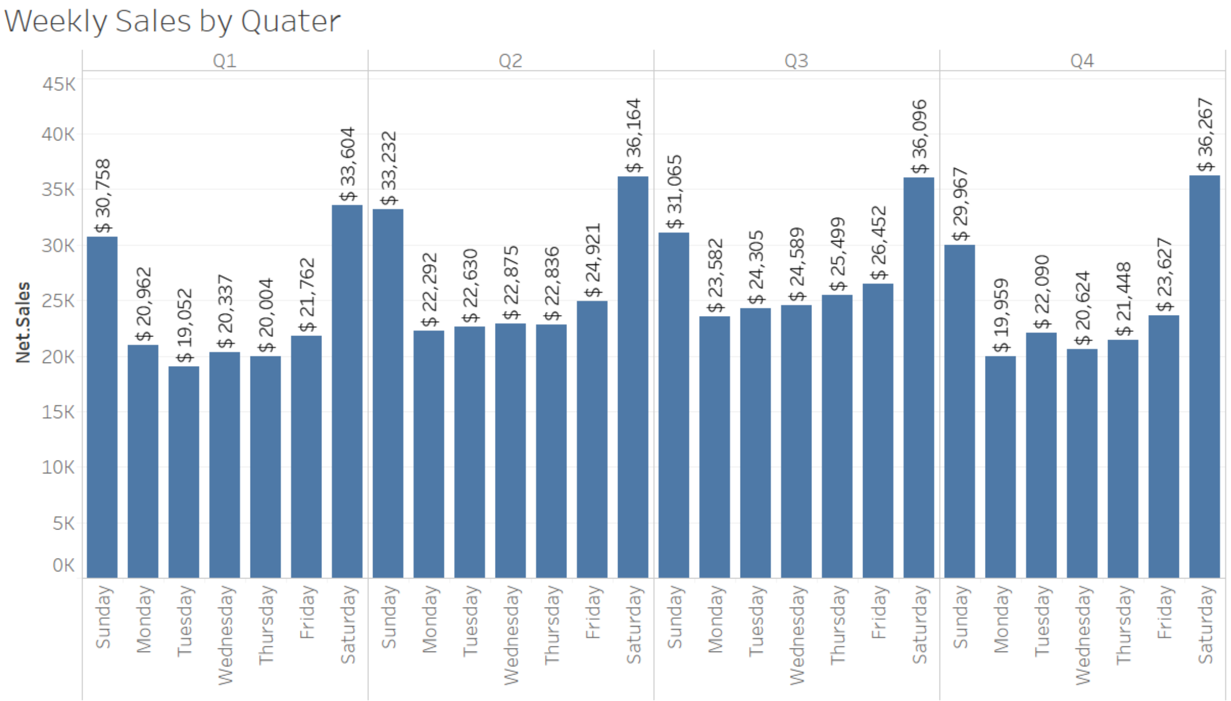
Interpretation
The graph suggests some variation in sales accross different months. In August, the coffee shop makes the most sales with 69.263k dollars. January and November are time when the store makes the least sales, with only about 52k dollars.

```r
knitr::include_graphics('Avg_Daily_Sales.PNG')
```



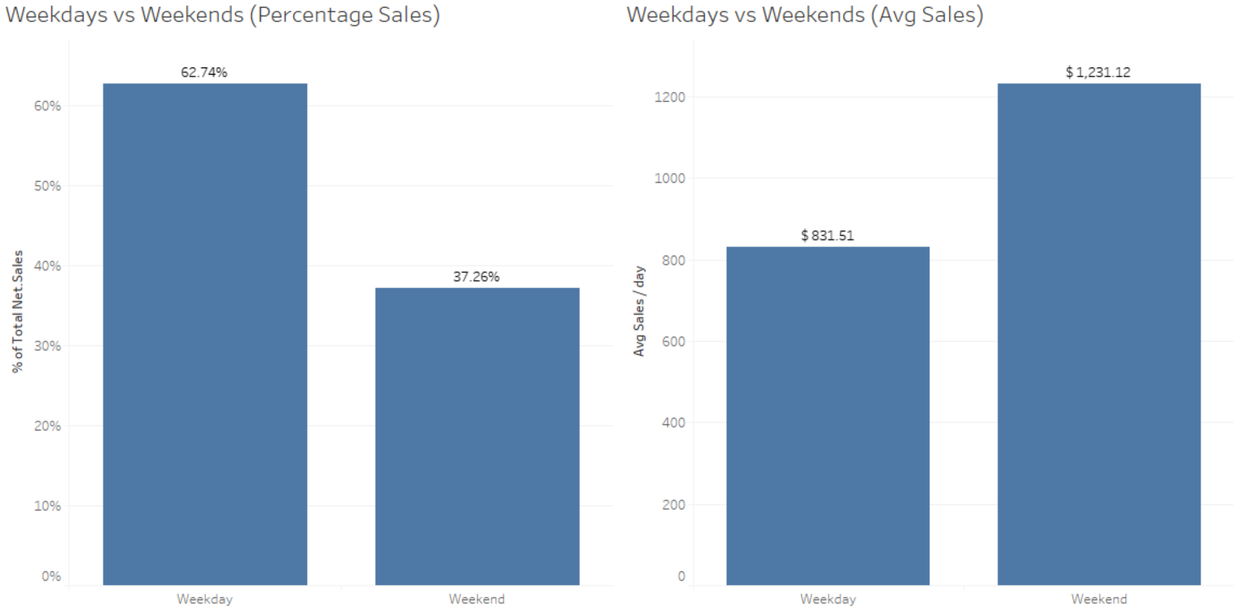
Interpretation
The graph indicates that Saturaday and Sunday makes the most revenue among all seven days every week, which are much higher than sales on weekdays. What is more, sales among weekdays are stable with few variation on average.

```
knitr::include_graphics('weekly sales pattern.PNG')
```



Interpretation For each quarter, the weekly patterns are quite similar. Weekends are more profitable than normal weekdays, and the average sales of Friday is a little higher than other weekdays.

```
knitr::include_graphics('Weekdays vs Weekends.PNG')
```



Interpretation

Need to come up

Summary

From this data exploratory process, we can clearly conclude a remarkable trend in the volume distribution, which is the sales drop in the afternoon (compared with morning) on both weekends and weekdays. Accordingly, we would like to figure out why sales drop in the afternoon, and attempt to smooth sales across different time periods in the next section.

Smoothing Demand Out

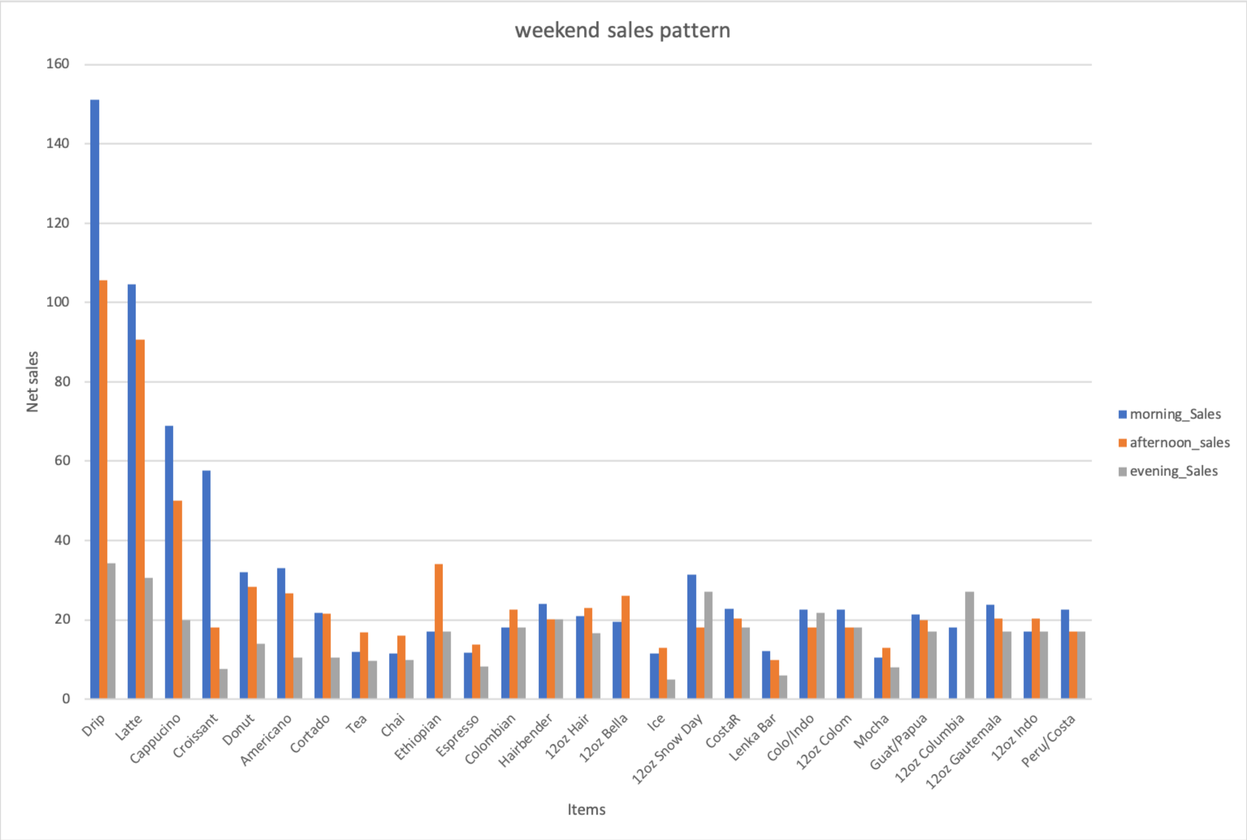
Which items contribute to sales difference between morning and afternoon?

Description and Rationale for the Chosen Analysis As we discussed above, the demand is very high on mornings and weekends, while very low at other times. So we would like to smooth the sales distribution over time. Here, we attempt to explore the sale performance of different items across various time periods on weekdays and weekends.

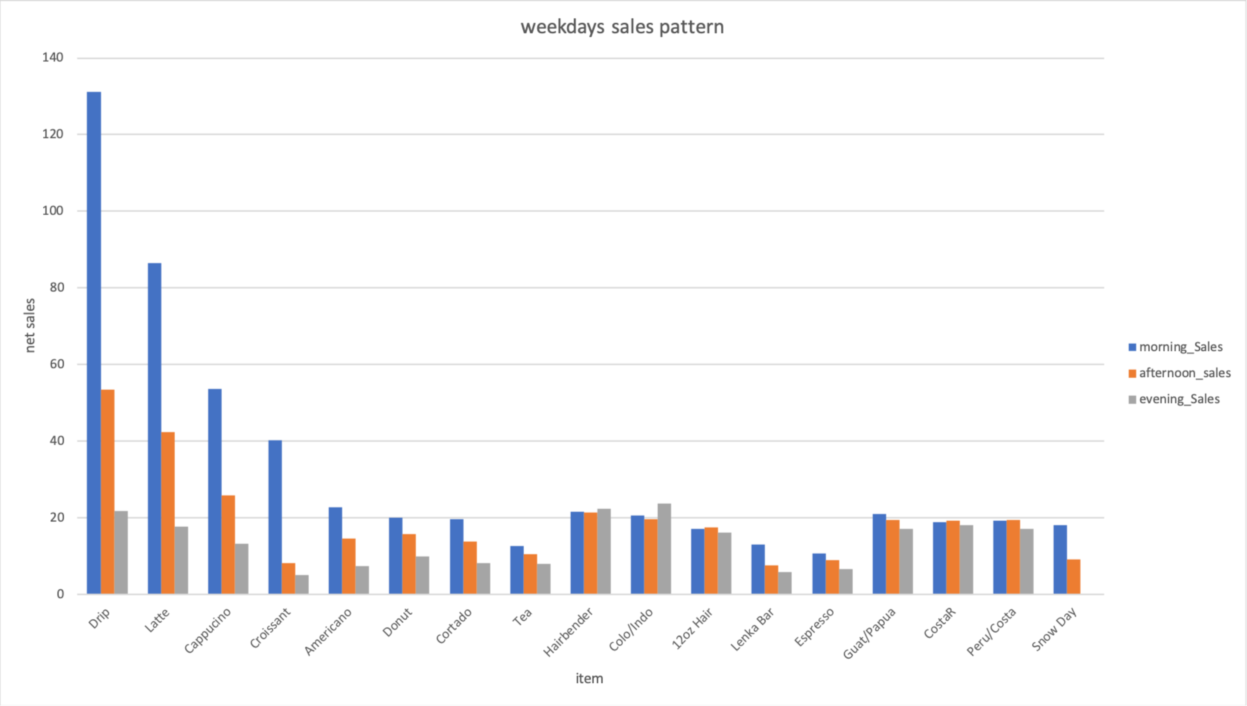
```
# clean and transform transaction data for association rules
sales <- read.csv('df1.csv')
sales$merchandise <- as.character(sales$merchandise)
sales$merchandise <- sapply(sales$merchandise, function(x) paste(unique(unlist(str_split(x,", "))), coll

sales$merchandise<-sapply(lapply(strsplit(sales$merchandise,","), sort), paste, collapse=",")
sales$merchandise <- as.character(sales$merchandise)
sales$hour <- as.character(sales$hour)
sales$hour <- as.numeric(sales$hour)
sales$X <- NULL
```

The following graph shows the distribution of items with net sales more than $20 on the morning, afternoon and evening of weekdays and weekends respectively. Here, we define the time before 12 p.m. as morning, 12 p.m – 4 p.m. as afternoon, and after 4pm as evening.

```
knitr::include_graphics('weekend.png')
```



```r
knitr::include_graphics('weekday.png')
```



Interpretation At weekends and weekdays, Drip, Latte, Cappucino and Croissant are top four selling items in the coffee store. However, sales pattern across time is not very smooth. Among all items, Drip and Latte drop the most in the afternoon. Moreover, it is apparent that the gap among top picked items on weekday morning and afternoon is wider than that on weekends.

How to smooth demands out over different time periods?

Description and Rationale for the Chosen Analysis As the graph shows above, during weekends, Drip and Croissant do not maintain their performance in the afternoon while Latte still performs quite well in both morning and afternoon. For weekdays, sales of top picked items drop dramatically in the afternoon, however, Americano and Donut perform comparably well in the afternoon.

Hence, we apply the association rule to these items (Latte, Americano and Donut) to examine the relevant items sold in this store, based on which, we could offer bundle recommendations to boost sales and smooth demands.

```
sales_weekend <- sales %>% filter(weekday == 6 | weekday == 7)
arule_weekend <- strsplit(sales_weekend$merchandise, ',')
trans_weekend <- as(arule_weekend, 'transactions')
rule_weekend <- apriori(trans_weekend, parameter = list(supp=0.001, 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   0.001      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: 42
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[69 item(s), 42787 transaction(s)] done [0.01s].
## sorting and recoding items ... [35 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [389 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rule_weekend %>% subset((subset = lhs %pin% 'Latte')) %>%
  sort(by = c("lift"),decreasing=TRUE) %>%
  head(n=5) %>% inspect()
```

```
##     lhs                     rhs      support     confidence lift     count
## [1] {Ice,Latte }       => {Oat}    0.002126814 0.04948341 6.049277  91
## [2] {Ice,Latte}        => {Soy}    0.001004978 0.08382066 4.175128  43
## [3] {Latte}            => {Soy}    0.003809568 0.08354690 4.161491 163
## [4] {Latte }           => {Oat}    0.005398836 0.03342981 4.086747 231
## [5] {Latte ,Lenka Bar} => {Almond} 0.001075093 0.21800948 3.638054  46
```

```
sales_weekday <- sales %>% filter(weekday != 6 & weekday != 7)
arule_weekday <- strsplit(sales_weekday$merchandise, ',')
trans_weekday <- as(arule_weekday, 'transactions')
rule_weekday <- apriori(trans_weekday, parameter = list(supp=0.001, 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   0.001      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: 90
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[71 item(s), 90283 transaction(s)] done [0.02s].
## sorting and recoding items ... [38 item(s)] done [0.00s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [319 rule(s)] done [0.00s].
## creating S4 object  ... done [0.02s].
```

```
rule_weekday %>%
  subset((subset = lhs %pin% 'Americano')|(subset = lhs %pin% 'Donut')) %>%
  sort(by = c("lift"), decreasing=TRUE) %>%
  head(n=10) %>% inspect()
```

```
##      lhs                   rhs             support     confidence lift
## [1]  {Almond,Donut}     => {Latte }        0.001462069 0.51162791 3.355950
## [2]  {Donut,Latte }     => {Almond}        0.001462069 0.15331010 2.875217
## [3]  {Americano,Drip }  => {Ice}           0.002093417 0.50000000 2.220001
## [4]  {Donut,Ice}        => {Drip}          0.001639290 0.13780261 1.805956
## [5]  {Americano,Latte } => {Ice}           0.001561756 0.38524590 1.710493
## [6]  {Donut,Drip}       => {Ice}           0.001639290 0.38046272 1.689255
## [7]  {Donut,Drip }      => {Ice}           0.005571370 0.35749822 1.587293
## [8]  {Donut,Ice}        => {Drip }         0.005571370 0.46834264 1.521258
## [9]  {Donut}            => {Hot Chocolate} 0.001196238 0.01879568 1.424795
## [10] {Donut,Ice}        => {Latte }        0.002436782 0.20484171 1.343630
##      count
## [1]  132
## [2]  132
## [3]  189
## [4]  148
## [5]  141
## [6]  148
## [7]  503
## [8]  503
## [9]  108
## [10] 220
```

Interpretation 1. Results of the first association rule show that when people buy Latte at weekends, they are more likely to add extras in their drink, supported by the high lift and high counts. Hence, we could offer some discounts on extras such as Oat, Almond and Soy when buying Latte with other items that do not perform very well in the afternoon, such as the Croissant and Drip.

2. Results of the second association rule suggest that people tend to buy Donuts with drinks including Latte, Americano and Drip on weekdays on the basis of their high lift and counts. Therefore, the coffee shop could take advantage of this trend and bring up the sales of other unpopular items by bundling Donuts, drink and other food as an afternoon tea. In this way, the sales in the afternoon of weekdays could be largely enhanced.

Conclusion

To address the sales drop in afternoon and weekdays, we could firstly recommend coffee store to implement reward programs in the afternoon on weekdays, in which way to boost daily sales amount and smooth the peak of the sales.

Secondly, coffee shop could offer various food combos in both weekdays and weekends. For weekends, customers could get discounts on extras/toppings in the beverage if they also buy Croissant and Drips, which do not usually perform well in the afternoon. For weekdays, offering bundled meals of Americano, donuts and other meats with low performance in the afternoon will contribute to maximizing their sales and smoothing sales.

Drive Customer Loyalty

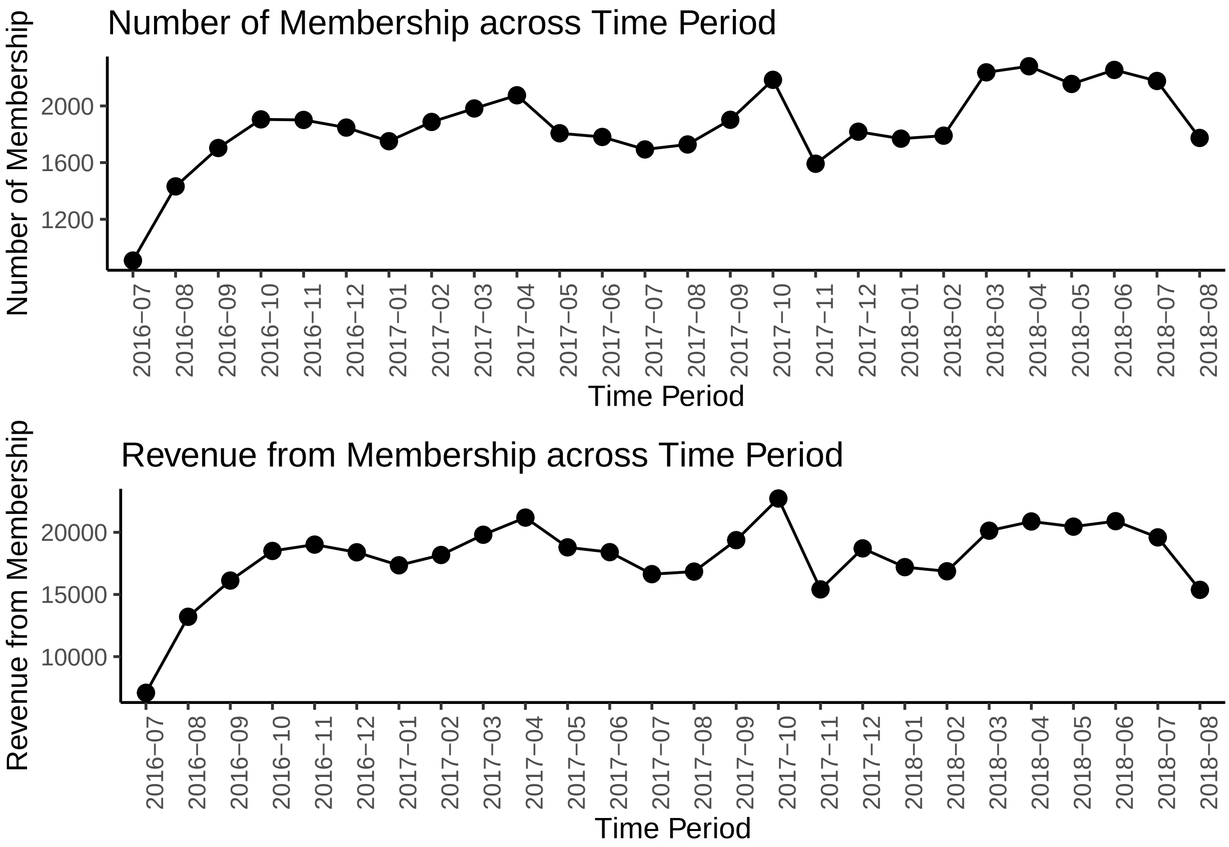
Convert One-time customer into return customers

Description and Rationale for the Chosen Analysis The coffee shop has assumed their customer base is fairly loyal. Therefore, the purpose of this analysis is to test whether their assumption is true or not based on the historical sales data. If not, we would like to drive more customer loyalty by offering insights. For this analysis, we are only focusing on customers with specific customerid, who are members of this coffee store, since data about non-members are missing in terms of customerid and scarce as well. Then, we transform the transaction-level data into customer-level ones where each row represents key features of a unique customers. Given the life span, customer base is categoried as "one-time customer" and "return customer", by which we would like to examine differences purchasing patterns between these two groups and how to transform one-time customers into return customers.

Execution and Results (without code) First, we transform the transaction-level data into customer-level data.

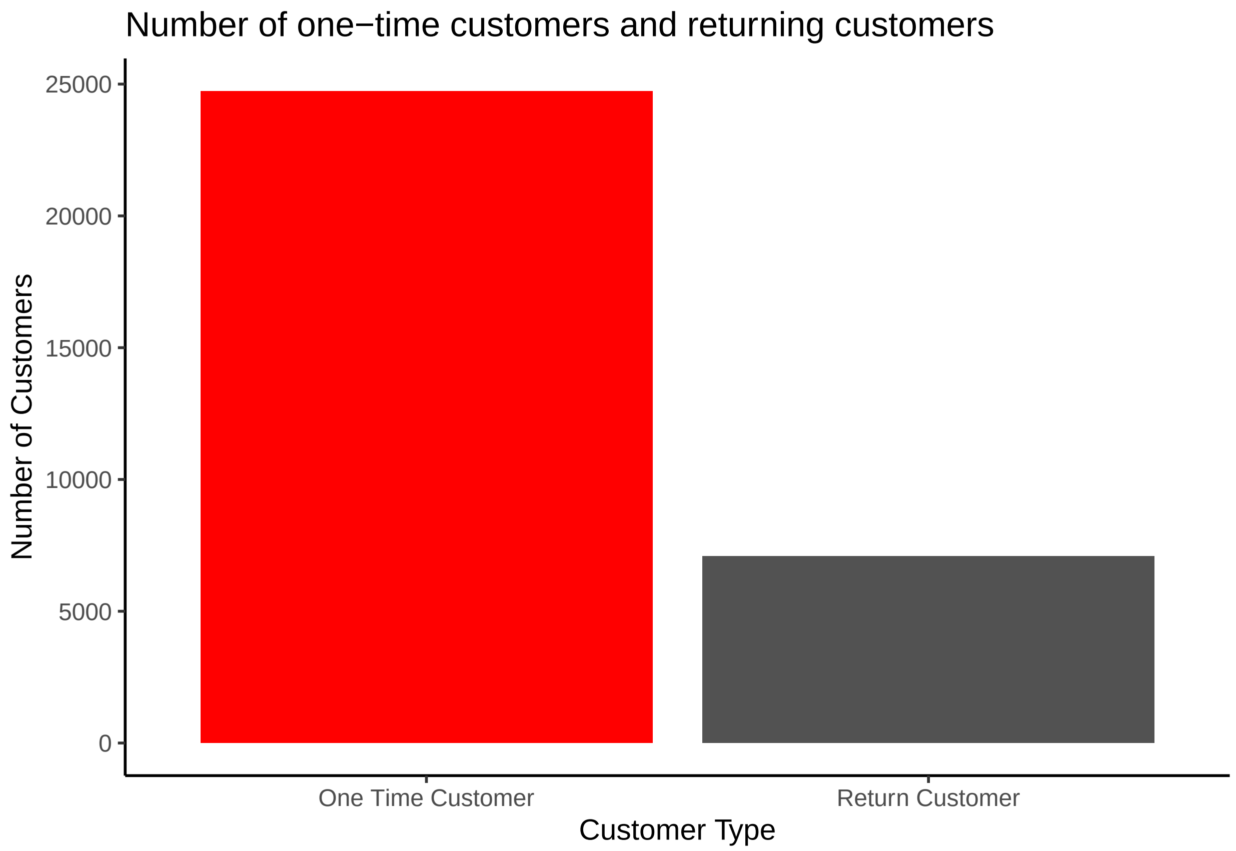
```r
# Calculate distance - daizy
newest_date <- max(centralPark$DateTime, na.rm = TRUE)
customer <- centralPark %>% drop_na() %>%
  group_by(Customer.ID) %>%
  summarise(recency = as.Date(newest_date, format="%Y/%m/%d")-
                as.Date(max(DateTime), format="%Y/%m/%d"),
            lifespan = as.Date(max(DateTime), format="%Y/%m/%d")-
                as.Date(min(DateTime), format="%Y/%m/%d"),
            total_amt = sum(Net.Sales),
            freq = length(Item))
customer$type <- ifelse(customer$lifespan == 0, 'One Time Customer', 'Return Customer')
```

```
# number of membership customer per month
p_m_number <- centralPark %>% drop_na() %>%
  group_by(Year_month)  %>%
  summarise(membership_number = length(unique(Customer.ID))) %>% ungroup() %>%
  ggplot(aes(x = Year_month, y = membership_number, group = 1))  +
  geom_line() + geom_point(size=4, shape=20) +
  labs(title = "Number of Membership across Time Period", x = 'Time Period', y = 'Number of Membership'
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# revenue of membership customer per month
p_m_revenue <- centralPark %>% drop_na() %>%
  group_by(Year_month)  %>%
  summarise(membership_revenue = sum(Net.Sales)) %>% ungroup() %>%
  ggplot(aes(x = Year_month, y = membership_revenue, group = 1))  +
  geom_line() + geom_point(size=4, shape=20) +
  labs(title = "Revenue from Membership across Time Period", x = 'Time Period', y = 'Revenue from Membe
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

plot_grid(p_m_number, p_m_revenue, ncol = 1)
```



Interpretation The number of membership in this coffee shop does not show a steadily increasing trend over two years. Given this stable existing customer base, the revenue also does not increase too much from this group of people, which is different from what the coffee shop has assumed before. Hence, we are going to explore why it happenes.

```
# Repeat purchase rate (customer buy twice / total customer)
customer$highlight <- ifelse(customer$type == 'One Time Customer',
                                          'yes', 'no')
ggplot(data = customer, aes(x = type, fill = highlight))  +
  geom_bar() +
  scale_fill_manual(values = c( "yes"="red", "no"="grey32"), guide = FALSE) +
  labs(title = "Number of one-time customers and returning customers", x = 'Customer Type', y = 'Number
```

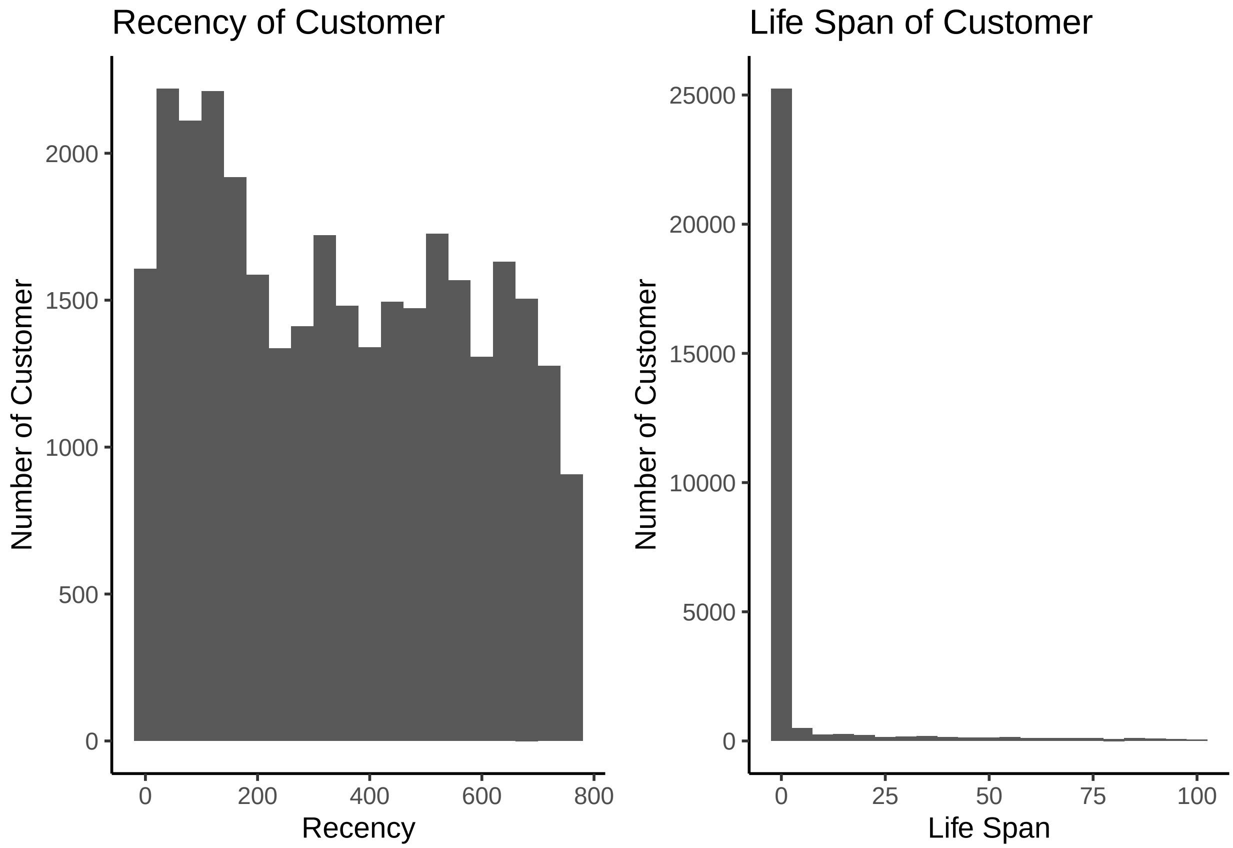


```
customer$highlight <- NULL
# customer value
average_value = sum(customer$total_amt) / sum(customer$freq) # (average_value = total revenue / number
purchase_freq = sum(customer$freq) / nrow(customer) # (purchase_freq = number of orders / number of cus
customer_value = average_value * purchase_freq
# https://blog.smile.io/easy-way-to-calculate-and-increase-customer-lifetime-value
```

Interpretation Given the life span, our customer base is categoried as "one-time customer" and "return customer". From this graph, we could see one-time customers largely outnumber the return customers, which might explain the stable revenue generated by membership group from 2016 to 2018. Considering the imbalanced ratio between one-time customers and return customers, our objective here is to convert one-time customers to return customers, which could improve sales of coffee shop.

```
# Recency
p_recency <- ggplot(data = customer, aes(x = recency))  +
  geom_histogram(binwidth=40) +
  labs(title = "Recency of Customer", x = 'Recency', y = 'Number of Customer')
```

```r
# lifespan
p_lifespan <- customer %>% filter(lifespan < 100) %>%
  ggplot(aes(x = lifespan))  +
  geom_histogram(binwidth=5) +
  labs(title = "Life Span of Customer", x = 'Life Span', y = 'Number of Customer')

plot_grid(p_recency, p_lifespan)
```

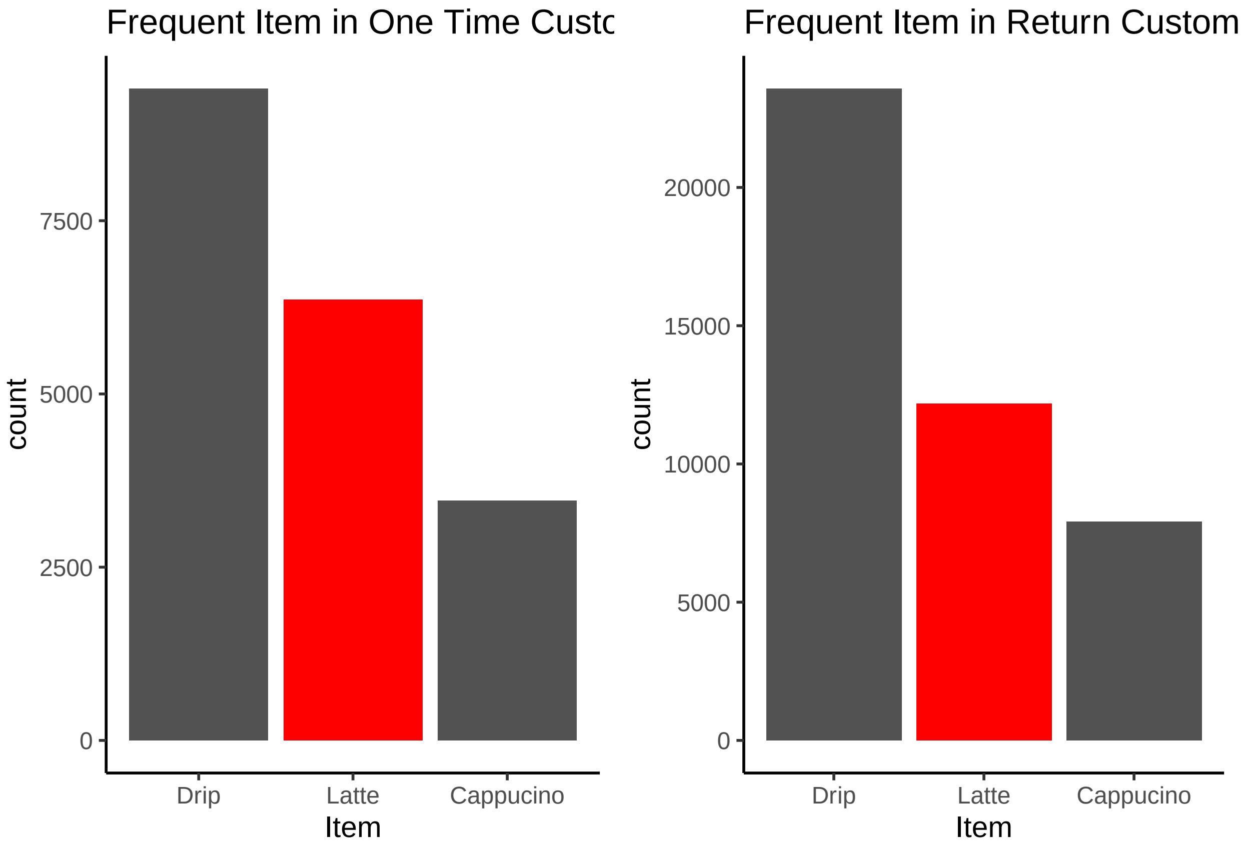


Interpretation Here are more further graphs demonstrating that "assumed loyalty customers" by coffee shop are not loyal at all. For all coffee shop members, the left graph suggests a large proportion of customers bought items from this coffee shop a long time ago, even up to 400 days. From the right graph, we could clearly see the majority of customers are one-time customers with only 1 day of life span.

```r
# divide the table into one-time and return customers respectively
centralPark_membership <- centralPark
centralPark_membership <- centralPark[centralPark$Membership == 'Member',]
return_list <- customer$Customer.ID[customer$type == 'Return Customer']
centralPark_membership$type <- ifelse(centralPark_membership$Customer.ID %in% return_list,
                                      'Return Customer', 'One Time Customer')
centralPark_return <- centralPark_membership[centralPark_membership$type == 'Return Customer',]
centralPark_onetime <- centralPark_membership[centralPark_membership$type == 'One Time Customer',]
```

```r
# Frequent items in one time customer vs return customer

centralPark_onetime_coffee <- centralPark_onetime[centralPark_onetime$Category == 'Coffee',]
centralPark_return_coffee <- centralPark_return[centralPark_return$Category == 'Coffee',]
```

```
centralPark_onetime_coffee$highlight <- ifelse(centralPark_onetime_coffee$Item == 'Latte',
                                      'yes', 'no')
centralPark_return_coffee$highlight <- ifelse(centralPark_return_coffee$Item == 'Latte',
                                      'yes', 'no')
max_freq <- 4
onetime_freq_item <- row.names(as.data.frame(summary(centralPark_onetime_coffee$Item, max=max_freq)))
p_onetime_freq_item <- centralPark_onetime_coffee %>% filter(Item %in% onetime_freq_item) %>%
  ggplot(aes(x = factor(Item,levels=names(sort(table(Item),decreasing = TRUE))), fill = highlight)) +
  geom_bar()+
  scale_fill_manual(values = c( "yes"="red", "no"="grey32"), guide = FALSE) +
  labs(title = 'Frequent Item in One Time Customer', x = 'Item')

return_freq_item <- row.names(as.data.frame(summary(centralPark_return_coffee$Item, max=max_freq)))
p_return_freq_item <- centralPark_return_coffee %>% filter(Item %in% return_freq_item) %>%
  ggplot(aes(x = factor(Item,levels=names(sort(table(Item),decreasing = TRUE))), fill = highlight)) +
  geom_bar()+
  scale_fill_manual(values = c( "yes"="red", "no"="grey32"), guide = FALSE) +
  labs(title = 'Frequent Item in Return Customer', x = 'Item')

plot_grid(p_onetime_freq_item, p_return_freq_item, ncol = 2)
```



Interpretation Latte is the item which shows much difference between one-time customers and return customers. We make an assumption that latte has some associations with the type of customers. To further test this hypothesis, we run association rule as shown below:

```
inspect(sort(rules_draw, by = 'lift', decreasing = TRUE)[1:3])
```

```
##     lhs                   rhs                       support    confidence
## [1] {Item=Drip}        => {type=Return Customer}   0.21062503 0.7148010
## [2] {Item=Cappucino}   => {type=Return Customer}   0.07070256 0.6959733
## [3] {Item=Latte}       => {type=One Time Customer} 0.05680499 0.3430421
##     lift     count
## [1] 1.073054 23582
## [2] 1.044790  7916
## [3] 1.027493  6360
```

Conclusion

From this analysis, we can clearly see drio and cappucino appear frequently with return customers, while for non-return customers, Latte is a common item for them. Based on what we discussed above, we would like to make suggestions about the latte on menu. Therefore, the store could design a survey for new customers, by which to figure out what favor of Latte do they prefer the most. Furthermore, the coffee shop could polish up the favor of Latte and try to get back more one-time customers.

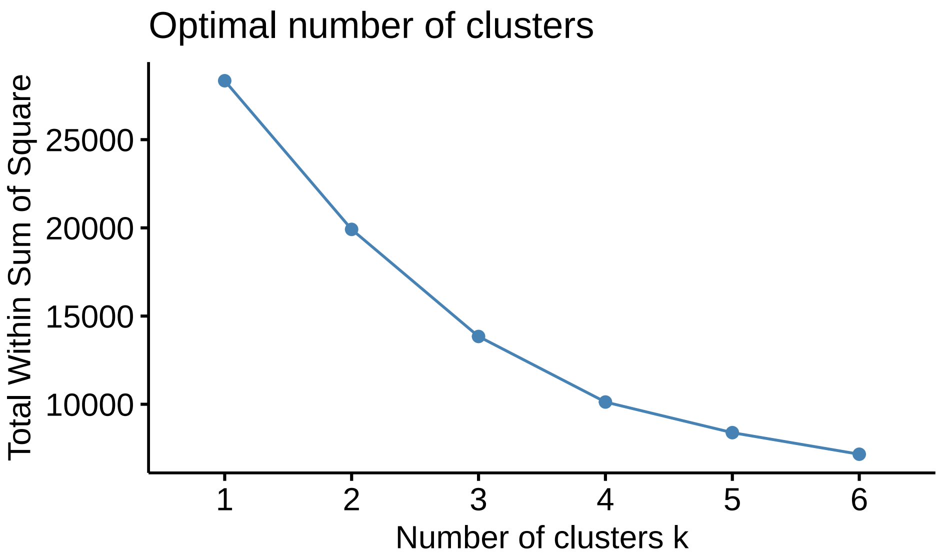
As it also shows, Drip and Cappucino appear frequently with return customers, as a result of which, our recommendation is to market these two items to new-customers through in-store advertisement (list as the "top drinks recommended") or offering in-store promotions targeted at these two drinks.

Enhance loyalty among return customers

Description and Rationale for the Chosen Analysis To drive more customer loyalty, the second approach is to enhance more loyalty among return customers. Here, we do a clustering for return customers to identify which segement of customers we should focus on, based on which, we are going to offer insights in converting more potential loyalty customers into loyalty ones.

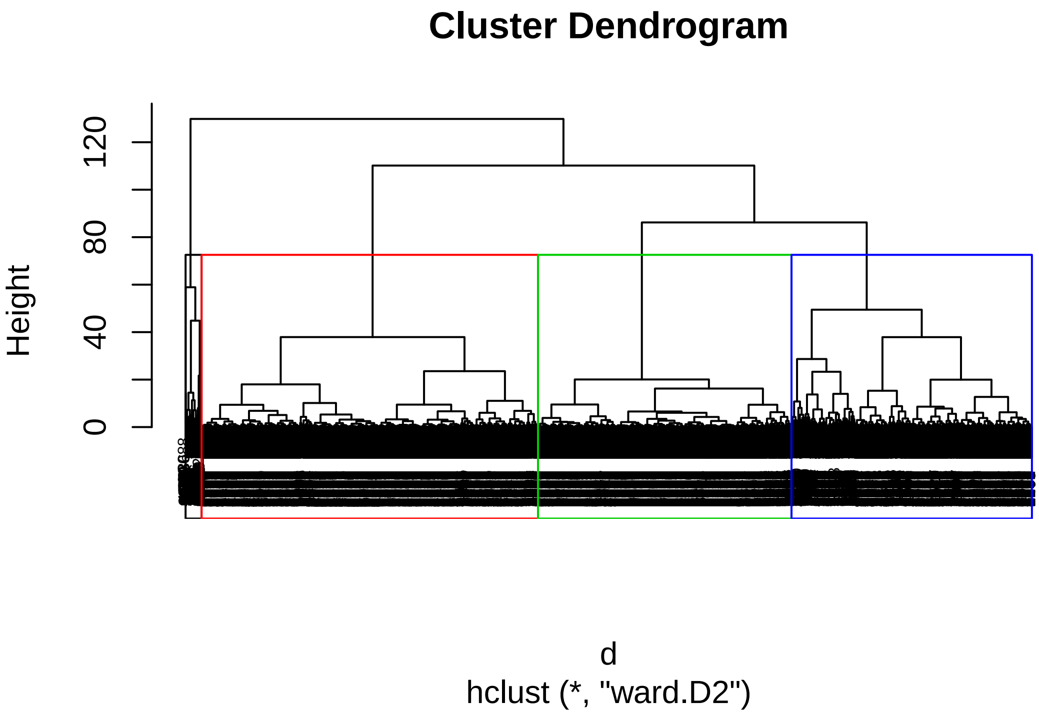
Execution and Results (without code)

```
# filter the return customers and do clustering
customer_return <- customer[customer$type == 'Return Customer',]
RFM <- customer_return
RFM$type <- NULL
RFM$Customer.ID <- NULL
RFM$recency <- as.numeric(RFM$recency)
RFM$lifespan <- as.numeric(RFM$lifespan)
RFM_scale <- scale(RFM)
fviz_nbclust(RFM_scale, FUN = hcut, method = "wss", k.max = 6)
```



We select the cluster k =4 and do clustering

```
# show the process of clustering
d <- dist(RFM_scale)
c <- hclust(d, method = 'ward.D2')
members <- cutree(c,k = 4)
plot(x = c, labels =  row.names(c), cex = 0.5)
rect.hclust(tree = c, k = 4, which = 1:4, border = 1:4, cluster = members)
```



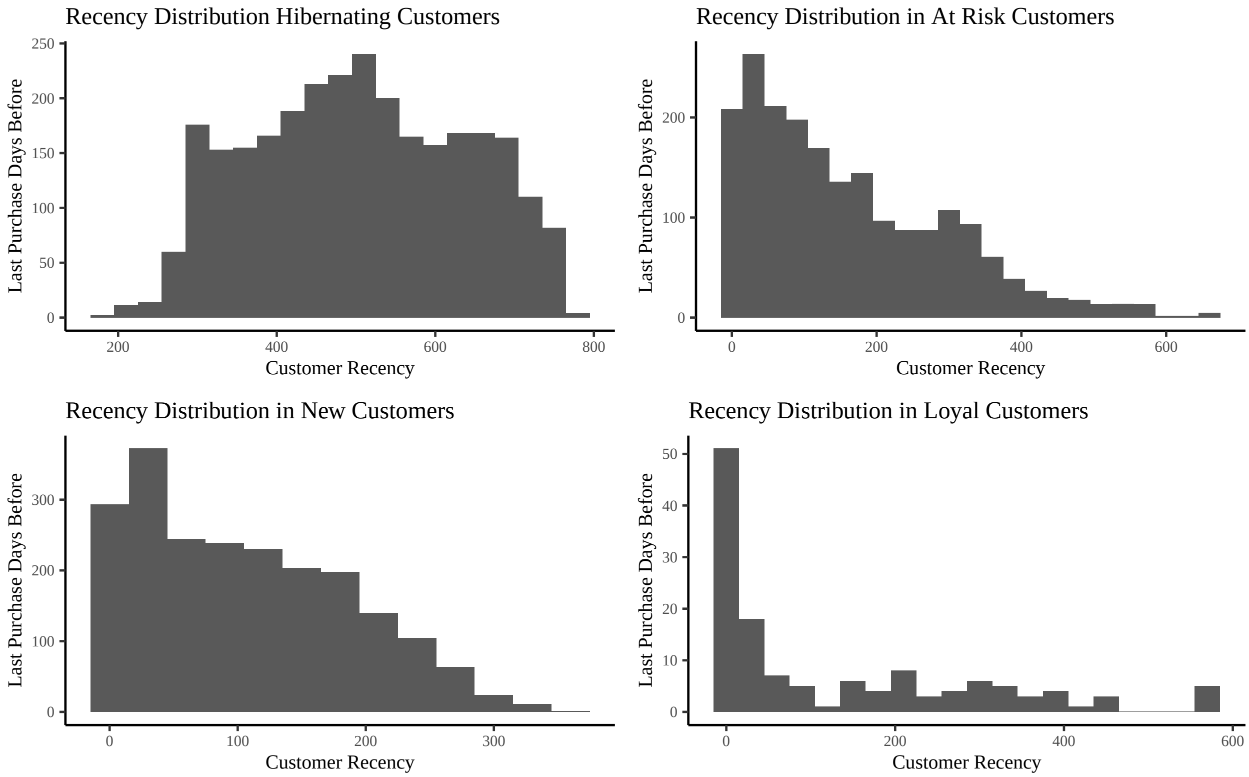
```
# show the count of each cluster
table(members)
```

```
## members
##    1    2    3    4 
## 2817 2013 2122  134
```

```
# show the mean feature of each cluster
RFM_result <- RFM
RFM_result$cluster <- members
aggregate(RFM, by=list(members), mean)
```

```
##   Group.1  recency  lifespan total_amt       freq
## 1       1 504.3685  72.59993  20.77627   6.424565
## 2       2 162.5618 381.90363  66.32392  19.939394
## 3       3 106.5650  65.00330  20.44943   6.525919
## 4       4 130.8806 480.70896 558.29075 168.835821
```

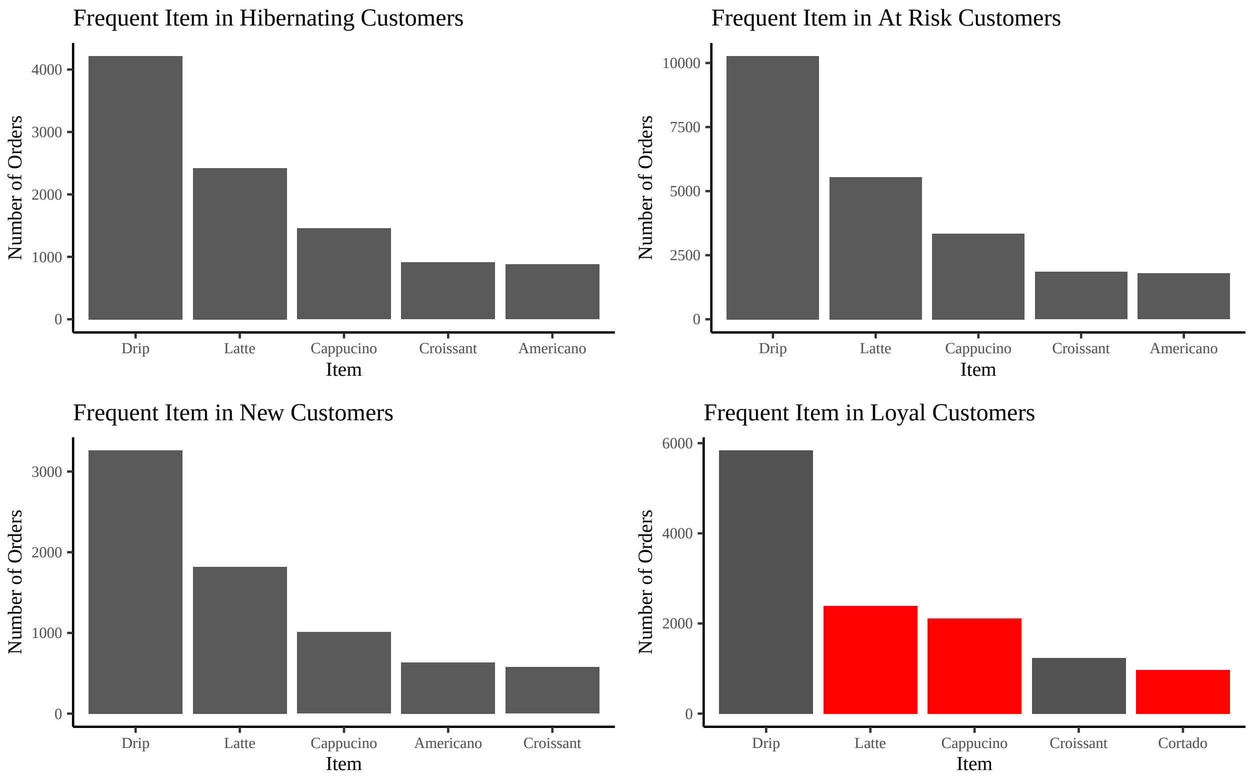
```
grid.arrange(recency1,recency2,recency3, recency4, nrow = 2)
```



Interpretation The analysis provides four customer segmentations. For cluster 1, customers' recency is high while total amount and frequency is low. These customers have low value and are unprofitable. We define them as hibernating customers. For cluster 2, customers' recency is high but their frequency and life span is also high, which means that they are loyal before but do not purchase for a long time. We define them as at risk customers. For cluster 3, customers' recency is low while frequency is also low, which means that they are new customers. For cluster 4, customers' lifspan is longest. Both total amount and frequency is the highest. Therefore, we define them as "loyal customers".

It is difficult to convert those hibernating customers, so our priority is to convert at risk customers (cluster2) and new customers (cluster3) into loyal customers.

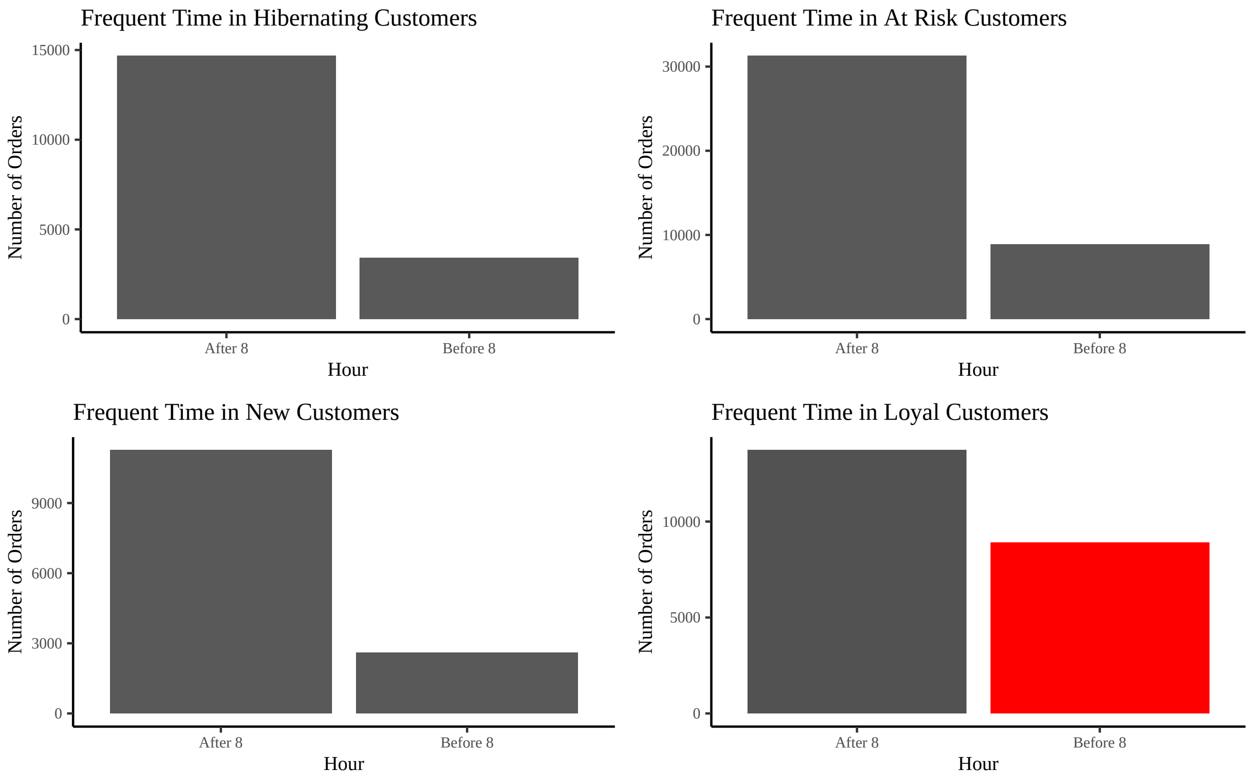
```r
# Frequent items in different return customer cluster
plot_grid(p_cluster_1_freq_item, p_cluster_2_freq_item,
          p_cluster_3_freq_item, p_cluster_4_freq_item)
```



Interpretation When we look at the frequent bought items amomg different customer segements, Latte is the item showing much difference between loyal customers and non-loyal customers. Non-loyal customers tend to buy more latte than loyalty customers. It aligns with our previous findings that the store need to polish polish up the favor of Latte.

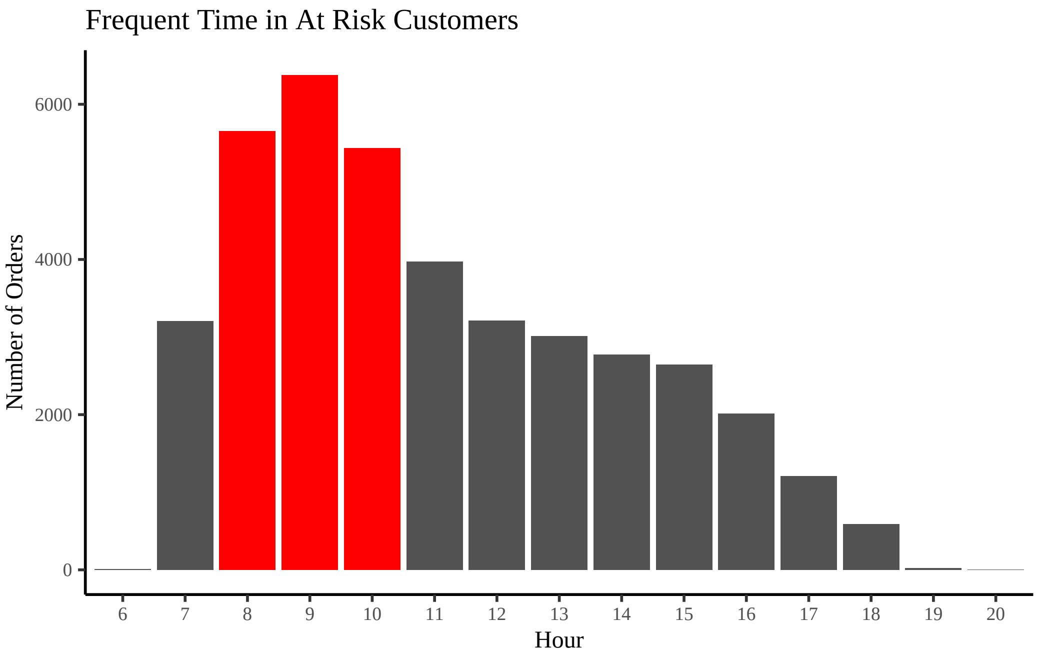
Cappucino is more favored by loyal customers than non-loyal ones. Most interestingly, Cortado is the unique item belonging to the loyal members' frequent bought items, which does not appear in non-loyal customers' lists. Therefore, we could offer coupons for those customers who try the Cortado and Cappucino for the first time, in which way to introduce our secure products to non-loyal customers.

```r
# Frequent time in different return customer cluster
plot_grid(p_cluster_1_freq_time, p_cluster_2_freq_time,
          p_cluster_3_freq_time, p_cluster_4_freq_time)
```



Interpretation Loyal customers often purchase coffee before 8 am, which means loyal customers might have habits to have breakfast in our store. Accordingly, we want to cultivate the habit among more customers by setting up a special 'breakfast time' when customers could got a discounted breakfast combo (e.g. extra 2 dollars to get a donut when buying coffee at store before 8am)

```
# Frequent time in at risk customer
centralPark_return_cluster_2  %>%
  ggplot(aes(x = Hour, fill = highlight)) +
  geom_bar()+
  scale_fill_manual(values = c( "yes"="red", "no"="grey32"), guide = FALSE) +
  labs(title = 'Frequent Time in At Risk Customers',
       x = 'Hour', y = 'Number of Orders') +
  my_theme
```



Interpretation For "at risk" customers, our goal is to remind them coming back to our store. Given their relatively high frequency, we assume there should be certain time period they usually purchase items from coffee store each day. This graph indicates their frequent time at the store are 7:30-10:30 am on morning, based on which, we could offer suggestions like sending reminder messages to them and give them discounts between that time period (7:30-10:30 am).

Conclusion

For all four customer segments we got from clustering, we only need to focus on "at risk" customers and new customers considering their potential in becoming loyalty members.

For at risk cusotmers, our goal is to remind them coming back. As what we discussed above, sending them reminder emails/texts and offering them coupon at certain time could stimulate them to come back to our coffee shop.

For new customers, our goal is to increase frequency and cultivate them to become our loyal customers. Therefore, we recommend them to introduce credit points program where customer could collect 1 point per purchase and get a free drink after gaining 10 points.

Final Takeaways

Smooth Demands Out

For CentralPark, two remarkable customer purchase patterns are: 1) peak time on morning and sales drop in afternoon after 3pm 2) sales drop on weekdays and increase on weekends. To smooth sales accross different time periods, coffee shop could offer various food combos in both weekdays and weekends. For weekends, customers could get discounts on extras/toppings in the beverage if they also buy Croissant and Drips, which do not usually perform well in the afternoon. For weekdays, offering bundled meals of Americano, donuts and other meats with low performance in the afternoon will contribute to maximizing their sales and smoothing sales.

Drive Customer Loyal

The coffee shop believes it has a loyal customer base and could generate more revenue from the existing base. However, it is not the case. The majority of customer base are one-time customers whose life span is one day, while return customers make up only about 20% of all customers with specific customerid (members). Also, not every one of return customers is loyal. Our clustering analysis shows, quite a large number of customers have not purchase any item for a long time, nor come to our store frequently. Further we have identified 4 segments and select 2 of them as our targeted customers.

Our recommendations are summarized as below:

1.Market Drip and Cappucino, which are mostly favored by return customers, to new customers through in-store advertisement ("top recommended drinks") and 10% off in-store promotions 2.Latte is the item which shows much difference among return customers vs one-time customers and loyal customers vs non-loyal customers as well. One-time customers and non-loyal customers tend to buy more latte than return and loyal customers. Therefore, the store could polish up the favor of Latte by surveying customers about their prefered ones 3.Cortado and Cappucino are products that are favored by loyal customers, therefore the store could offer coupons (10% off) for those customers who try the Cortado and Cappucino for the first time, in which way to introduce our secure products to non-loyal customers. 4.Cultivate the breakfast habit among more customers by setting up a special 'breakfast time' when customers could got a discounted breakfast combo (e.g. extra 2 dollars to get a donut when buying coffee at store before 8am) 5.For at risk cusotmers, sending them reminder emails/texts and offering them coupon at their habitual time (7:30-10:30am) could stimulate them to come back to our coffee shop. 6.For new customers, we could recommend them to introduce credit points program where customer could collect 1 point per purchase and get a free drink after gaining 10 points, in which way to increase their purchase frequency

With these measures, it would help CentralPark to smooth demands, boost sales and convert more non-loyal customers into loyal ones, which will eventually yield constant revenue over years.