Business Improvement Recommendations for a Coffee Shop

Contents

Introduction

Saturation makes it increasingly difficult for a coffee shop to capture a consumer's attention. Our goal is to investigate Central Perk's hypothesis that their current customer base is fairly loyal and business is consistent year over year. Based on this analysis and the current customer's buying patterns (influenced by seasonality and favorite categories), we will design recommendations that will help the business achieve two goals; normalize demand and generate additional revenue.

Analysis approach

Central perk hypothesised that their customer base is fairly loyal and their sales are fairly conctant year over year. We wanted to back this with data. We also wanted to find any patterns in the items purchased together which could be used to cross-sell items and inturn increase revenue. To smoothen the demand evenly throughout the day and week, we want to analyse the distribution of sales of various category across time and get insights to come up with recommendations to smoothen the demand. To increase the revenue from our existing customer base, we want to do a price sensitivity analysis to understand the customer behavious and inturn suggest price changes to increase revenue

To identify patterns in the items bought together we used association rules to get any insights to cross-sell items to increase sales.

To identify potential groups of customers we characterized the different types of people that come to Central Perk. We singled out customers based on how often they purchase from our shop, how much they spend, and the last time they visited our shop. With these unique fields we were able to cluster the data into 5 broad segments. The segments can be described as 'Non Members' - customers who do not have membership and we are unable to track, 'Tourists' - Low frequency of purchases and they have not visited recently 'Potential Loyal Customers' - low visit frequency and they have visited recently 'Ex Loyal Customers' - High visit frequency but they have not visited recently 'Loyal Customers' - High visit frequency, higher spend, and have visited recently

Assumptions and Limitations

- Our analysis limited to data provided and not accounting for external factors that influence sales, or amount of item sold such as competitors, store location, etc.
- We have assumed that the items bought at the same time as one transaction. We have grouped the item level data at the time and Customer Id level to get transaction level dataset.
- We hae removed the Event type Refund from the dataset assuming this wont have any effect on the number of items sold or the net revenue from customers.
- We assumed that there were no marketing promotions or any internal factors that could directly or indirectly influence the demand thereby influencing our analysis.
- We are unaware of any composition changes in items that could have triggered any change in demand of each items. We have assumed the composition to have remained the same for our analysis.
- We dont know if any promotional activities are driving repeat purchases among our registered customers.We assumed that there are no external factors influencing the registered customers.

Apart from these assumptions, the assumptions for each analysis have been listed in the respective sections.

Data import

Import the data and explore the same to understand its structure, and clean the data to make it ready for analyses.

```
# Loading required libraries
library(data.table)
library(stringr)
library(lubridate)
library(esquisse)
library(ggplot2)
library(reshape2)
library(chron)
library(ggfortify)
library(stats)
library(factoextra)
library(cluster)
library(tidyverse)
library(naniar)
library(ggpubr)
library(dplyr)
library(readr)
library(Rtsne)
library(kmed)
library(arules)
library(RSQLite)
library(knitr)
library(scales)
library(ggthemes)
library(ggrepel)
library(magrittr)
library(Rlof)
library(fpp)
library(forecast)
```

Importing data and merging the three files into one single dataframe.

```
# Data import

file1 <- fread('Central Perk Item Sales Summary 2017.csv',header = TRUE,
               stringsAsFactors = TRUE)
file2 <- fread('Central Perk Item Sales Summary 2016.csv',header = TRUE,
               stringsAsFactors = TRUE)
file3 <- fread('Central Perk Item Sales Summary 2018.csv',header = TRUE,
               stringsAsFactors = TRUE)


data <- rbind(file1,file2,file3)

# checking for row level duplicates
data <- data[!duplicated(data), ]
```

Data Cleaning

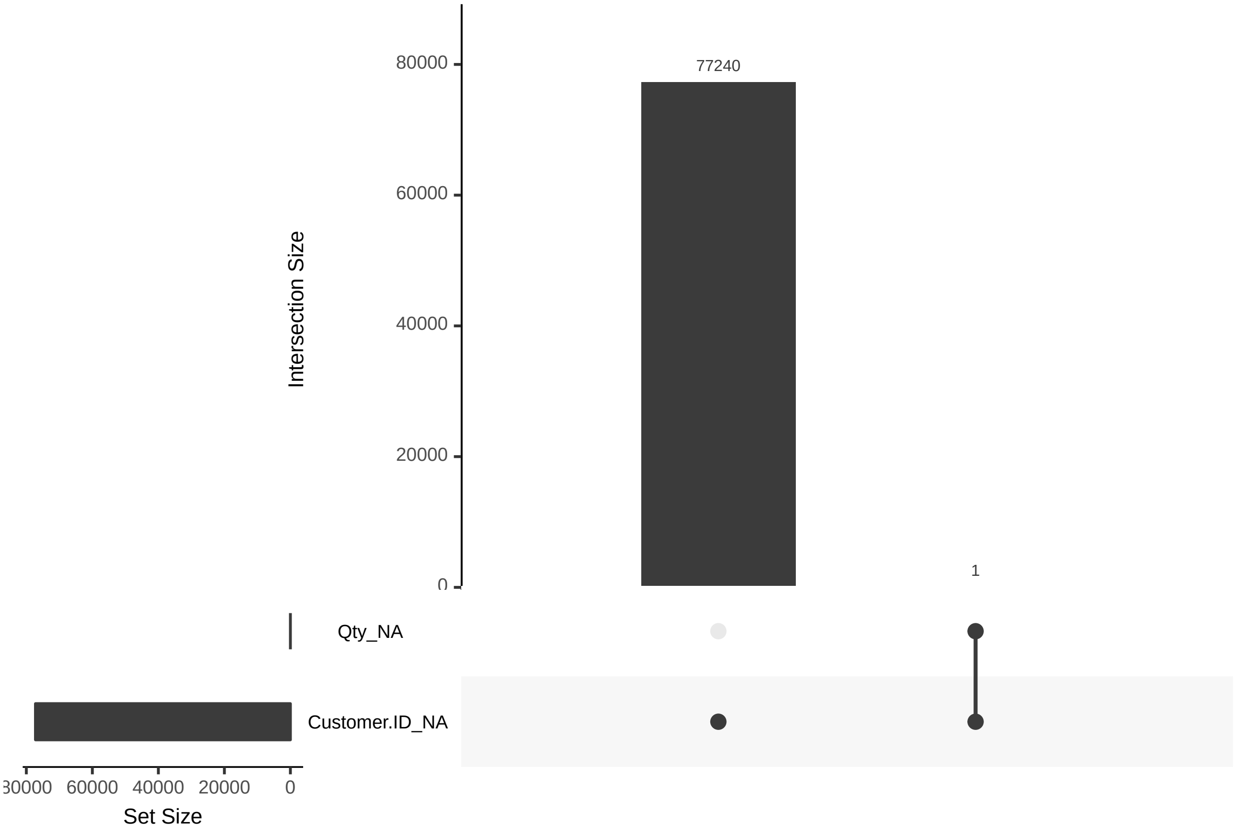
Checking missing values in each columns.

```r
colSums(is.na(data))
```

```
##             Date             Time         Category             Item 
##                0                0                0                0 
##              Qty Price.Point.Name      Gross.Sales        Discounts 
##                1                0                0                0 
##        Net.Sales              Tax            Notes       Event.Type 
##                0                0                0                0 
##      Customer.ID 
##            77241
```

```r
gg_miss_upset(data, nsets = 2)
```

```
## Warning: `lgl_len()` is deprecated as of rlang 0.2.0.
## Please use `new_logical()` instead.
## This warning is displayed once per session.

## Warning: `is_lang()` is deprecated as of rlang 0.2.0.
## Please use `is_call()` instead.
## This warning is displayed once per session.

## Warning: `lang()` is deprecated as of rlang 0.2.0.
## Please use `call2()` instead.
## This warning is displayed once per session.

## Warning: `mut_node_car()` is deprecated as of rlang 0.2.0.
## This warning is displayed once per session.

## Warning: The `printer` argument is deprecated as of rlang 0.3.0.
## This warning is displayed once per session.

## Warning: package 'bindrcpp' was built under R version 3.5.1
```



Description:

The graph above captures the number of missing records in the data, and also captures the interaction between the missing values. The linkages at the bottom capture which variables are missing in common for the corresponding bar. For example, the first bar has around 77K missing values for Customer ID and second column has only one record having missing values in Quantity and Customer ID.

Interpretation:

We observe that the bulk of missing values are present for 1 columns only. We assumed these to be the Non-registered customers.

Treating for Missing values in Customer Id field

We replace the Missing values in Customer ID column as "Non-Members". And removed the row having missing value in Quantity column.

```
data <- data[rowSums(is.na(data[ , 5])) == 0, ]
data$Customer.ID <- as.character(data$Customer.ID)
data$Customer.ID <- ifelse(is.na(data$Customer.ID),
             'Non-Member', data$Customer.ID)

# item name cleaning
data$Item <- as.character(data$Item)
```

```
data$Item[which(data$Item == '\U0001f34bLemonade\U0001f34b')] <- 'Lemonade'
```

Treating for Special characters in the data

We removed the '$' sign present in Net sales, Gross sales, Discount and Tax columns. We obsered that few fields had negetive values represented with enclosed brackets. We replaced these brackets with a negetive symbol.

```
data$Qty <- as.integer(data$Qty)
#Removing brackets in net sales and gross sales columns and adding a negetive sign
convert.brackets <- function(x){
  if(grepl("\\(.*\\)", x)){
    paste0("-", gsub("\\(|\\)", "", x))
  } else {
    x
  }
}
data$Net.Sales <- sapply(as.character(data$Net.Sales), convert.brackets, USE.NAMES = F)
data$Gross.Sales <- sapply(as.character(data$Gross.Sales), convert.brackets, USE.NAMES = F)
data$Tax <- sapply(as.character(data$Tax), convert.brackets, USE.NAMES = F)
data$Discounts <- sapply(as.character(data$Discounts), convert.brackets, USE.NAMES = F)
#removing the $ sign and converting all values to numeric type
data$Net.Sales = as.numeric(gsub("\\$", "", data$Net.Sales))
data$Gross.Sales = as.numeric(gsub("\\$", "", data$Gross.Sales))
data$Discounts = as.numeric(gsub("\\$", "", data$Discounts))
data$Tax = as.numeric(gsub("\\$", "", data$Tax))
```

We observed few inconsistancies in the data like 'Oat' categorized as None instead of Extras. We also removed the Event type Refund as we are more interested in items sold and want our analysis to be confined to the Event type Payment. We also observed a category 'None', which is of no use for our analysis. Hence we remoed rows with category None.

```
# all the oat should be in the category 'Extras'
data[data$Item == 'Oat', 'Category'] <-  'Extras'
# delete Refund record
data <- data[data$Event.Type == 'Payment',]
# delete non-product category
data <- data[data$Category != 'None',]
# add new column to calculate the profit, profit = gross sales * gross margin(20%) +
#                                                  discount(negative number).
data$profit <- data$Gross.Sales * .2 + data$Discounts
```

Summarization of data cleaning

In the data cleaning process, we have removed missing values from quantity column. We have removed '$' sign from the price columns. And corrected few inconsistancies in the data, like categorizing oats under extras, considering only 'Payment' under event type as we believe records under 'Refund' is not useful for customer analysis. We then removed the records belonging to category 'None' as this has no value to us.

Data Exploration

Combining Date and Time into one column.

```
data$time = mdy_hms(paste(data$Date,data$Time))
```

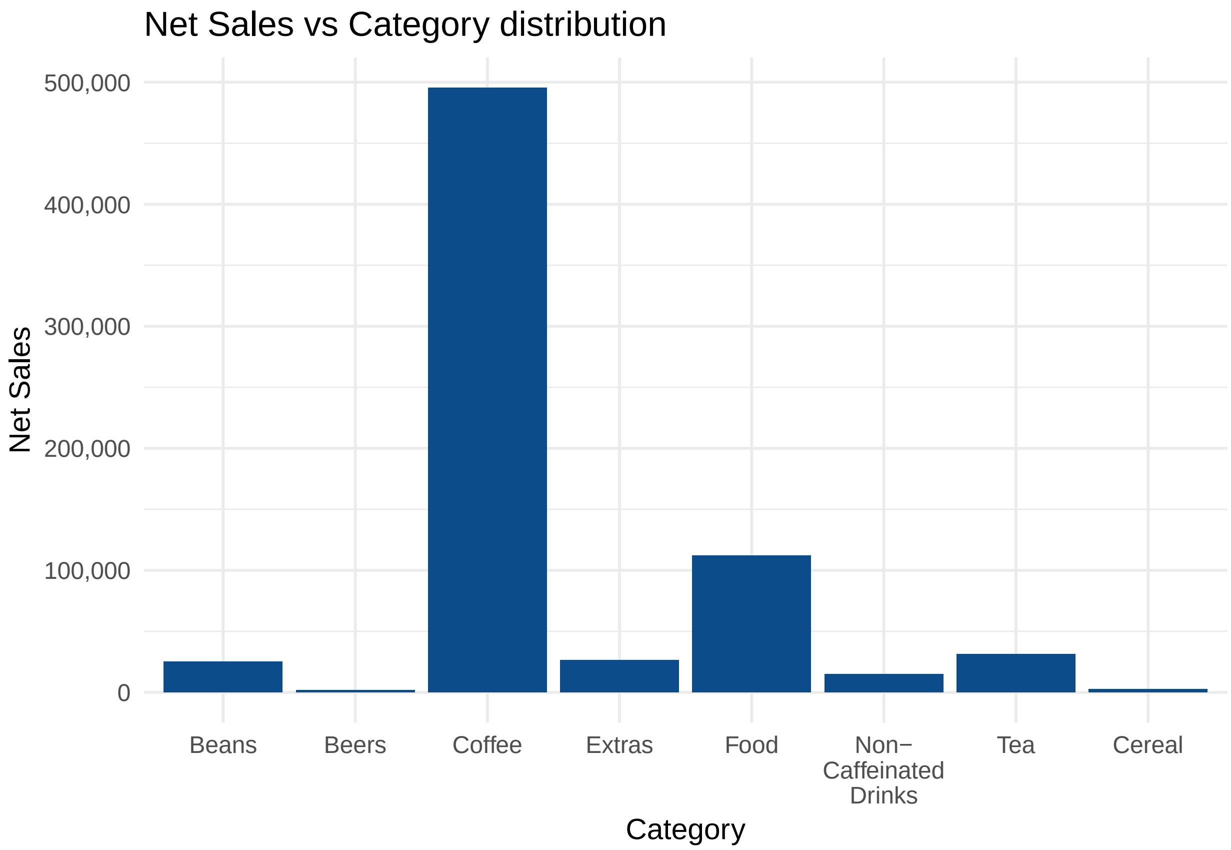
Adding a new column to calculate the profit, profit = gross sales * gross margin(20%) + discount(negative number).

```
data$profit <- data$Gross.Sales * .2 + data$Discounts
```

Exploring Category level data

The data we have is in the item level where every row contains description of each item. We wanted to understand how the data is distributed across each category through visualisation.

```
#Net sales vs category distribution:
ggplot(data, aes(x = Category, weight = Net.Sales)) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Net Sales vs Category distribution",
    x = "Category",
    y = "Net Sales") + scale_y_continuous(labels = comma) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  theme_minimal()
```



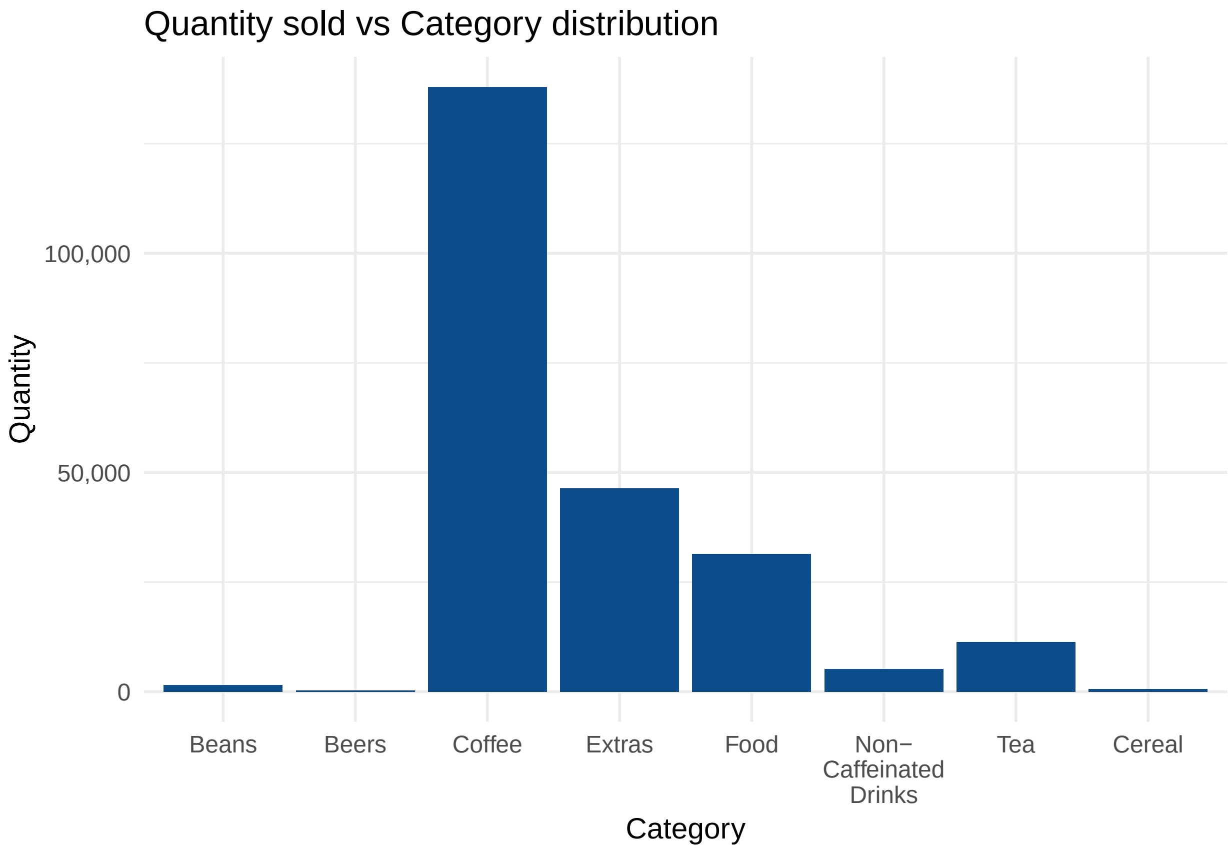
Description:

The graphs above captures the Distribution of Net sales across all the categories over the two years of data we have.

Interpretation:

We observe that the bulk of Net sales if from Coffe, as we would expect from a coffee shop. We also see that there are a lot of categories like beer, cereal, non-caffinated drinks which do not contribute a lot to net sales which could lead to further analysis and help re-think our category distribution.

```r
# Quantity sold vs category distribution:

# filtering out none in category

data <- data %>% filter(Category !="None")

ggplot(data = data) +
  aes(x = Category, weight = Qty) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Quantity sold vs Category distribution",
    x = "Category",
    y = "Quantity") + scale_y_continuous(labels = comma) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  theme_minimal()
```



Description:

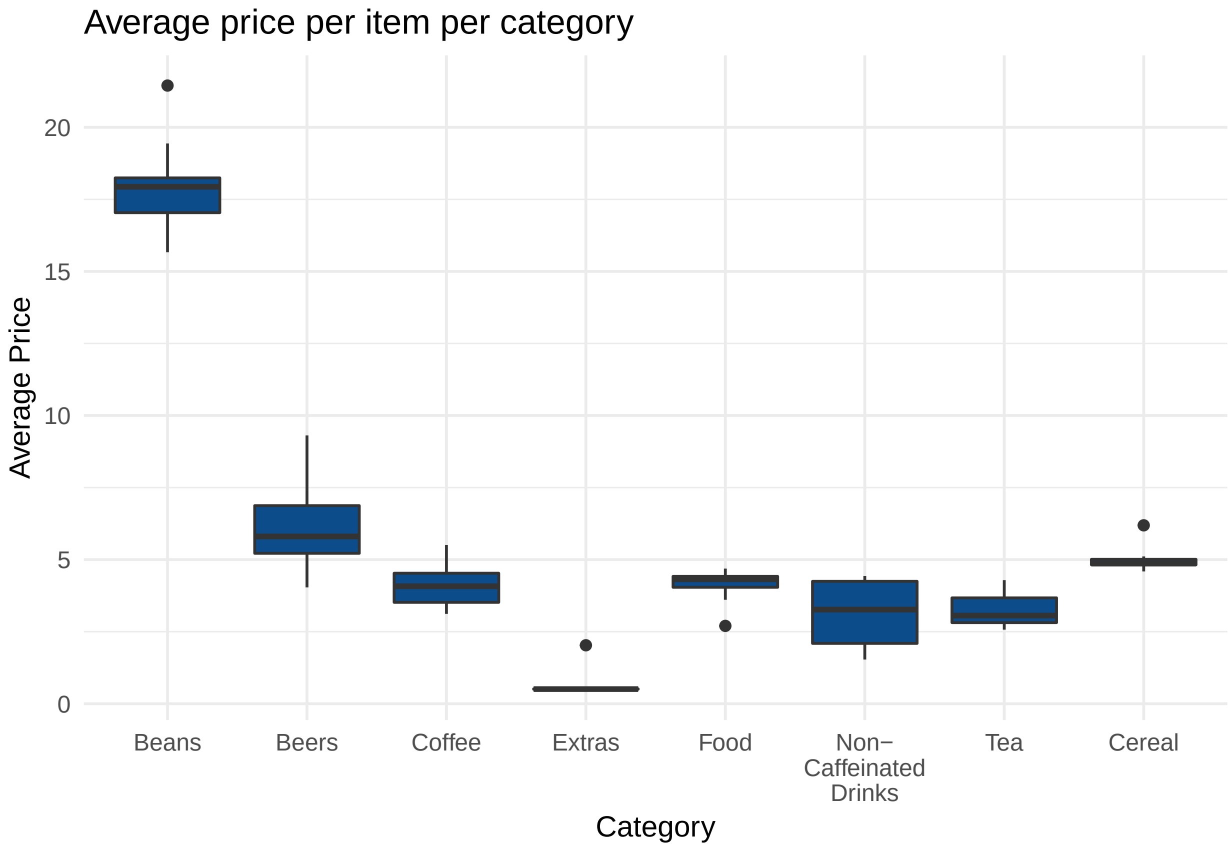
Now that we have seen the sales distribution, we wan tto see the amount of units that go behind driving the sales of the individual categories we saw in the previous graph. The graphs above captures the Distribution of Quatity across all the categories.

Interpretation:

We observe that coffee naturally lead to high quantity sold. On contrary we see that extras, even though they bring less revenue the quantity sold is quite high and on the otherside we see that the amount of beans sold is less and they generate comparitively higher sales.

Now that we know the general net sales distribution, we wanted to explore how the price per item in each category is distributed. we want to see the price of units that go behind driving the sales of the individual categories we saw in the previous graph.

```r
# average price per item per category
avg_prc <- data %>% group_by(Category,Item) %>% summarise(avg = mean(Net.Sales))

ggplot(data = avg_prc) +
  aes(x = Category, y = avg) +
  geom_boxplot(fill = "#0c4c8a") +
  labs(title = "Average price per item per category",
    y = "Average Price") + scale_y_continuous(labels = comma) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10))+
  theme_minimal()
```



Description:

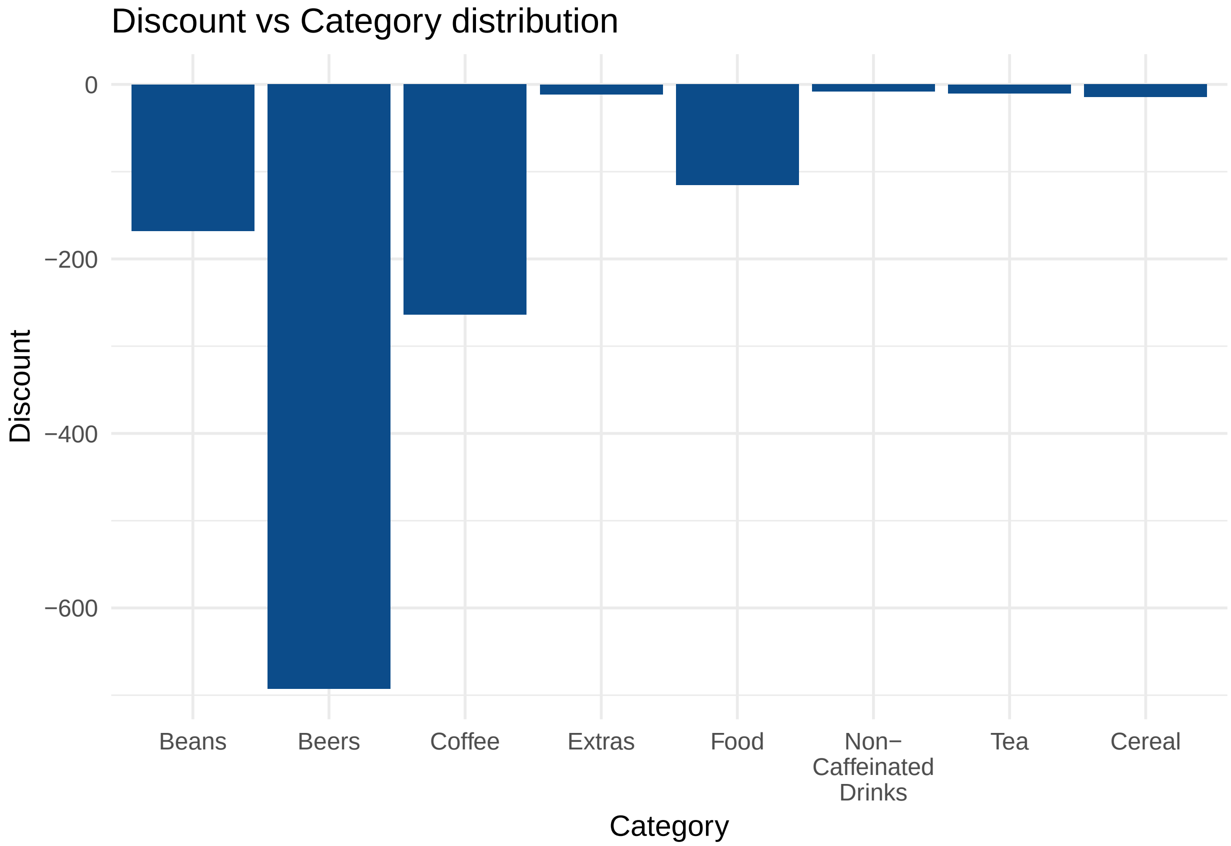
The above graph is a barplot which shows the variance in price among the items in each category.

Interpretation:

We observe that beans is driving more earning of all the categories. We see an opportunity here to upsell beans which has higher potential to increase the revenue.

We see that price per items in beer is pretty high on contrary to what we expected, but the net revenue from beans is low compared to items sold. Our hypothesis was that discounts could be the reason for this. We wanted explore deeper to understand the reason for this.

```
#Discount vs category

ggplot(data = data) +
  aes(x = Category, weight = Discounts) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Discount vs Category distribution",
    x = "Category",
    y = "Discount") + scale_y_continuous(labels = comma) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10))+
  theme_minimal()
```



Description:

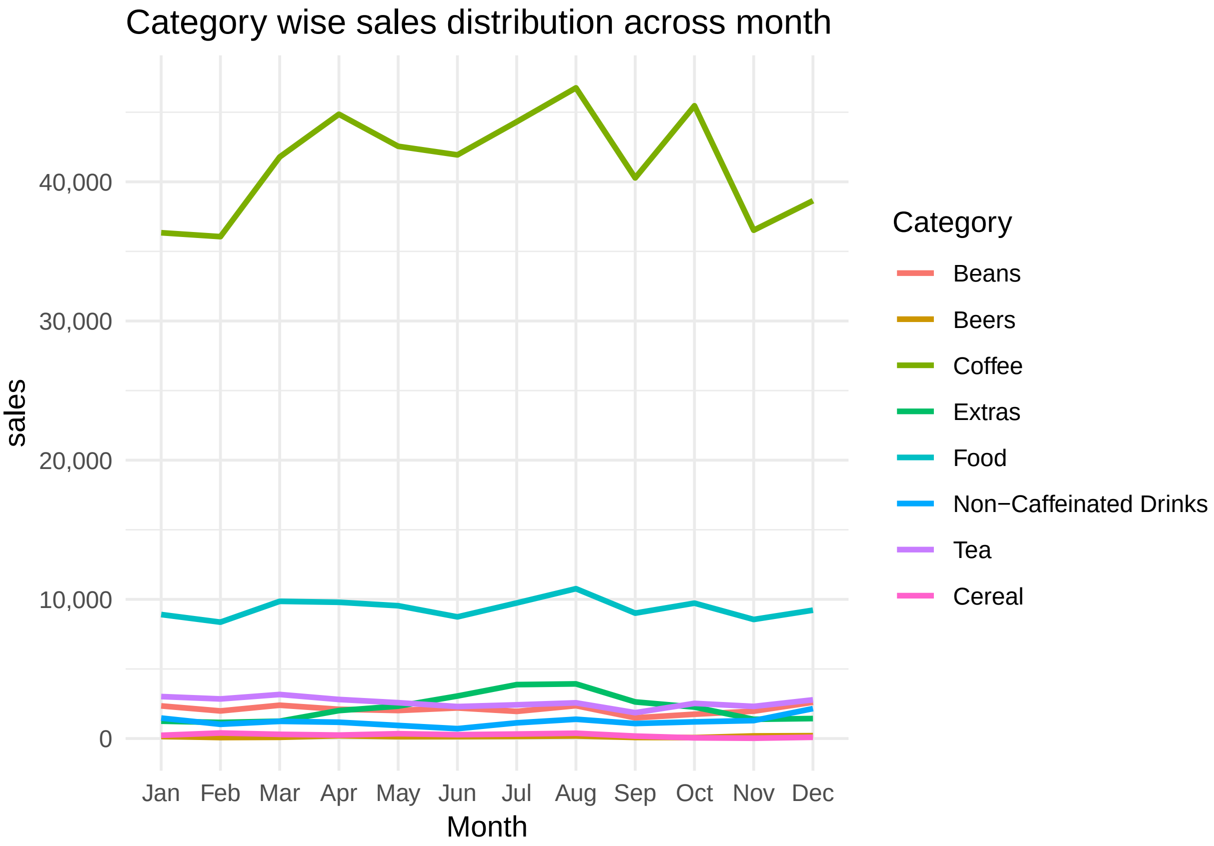
The graphs above captures the total discounts across all the categories.

Interpretation:

We observe that the e=very high level of discount given to beer. This is generating less sales and quantity sold. this could be a point of concern for us. This could mean that our customers are not interested in buying beer. we might have to rethink obout the stopping beer allltogether.

TIll now we agregating information across 2 years. Now we wanted to understand the pattern of sales across time. We wanted to plot sales of each catgory across month.

```
#category wise sales trend across the 2 years

cat_trend <- data %>% group_by(month=lubridate::month(time), Category) %>%
  summarise(sales = sum(Net.Sales))
cat_trend$month <- month.abb[cat_trend$month]
cat_trend$month = factor(cat_trend$month, levels = month.abb)

plotC <- ggplot(cat_trend, aes(x=month, y=sales, group=factor(Category)))
plotC + geom_line(data=cat_trend, aes(x=month, y=sales, group=factor(Category),
                                       color= Category),size=1) +
  labs(title = "Category wise sales distribution across month",
  x = "Month",y = "sales") + scale_y_continuous(labels = comma) +
  theme_minimal()
```



Description:

The graphs above captures the distributio of sales across different months of the year.

Interpretation:

As expected, we observe coffee being the main category whcih brings us good sales across all months. We observe demand fluctuations in coffee but other categories have a fairly linear trend across different months. We observe a some seasonality in extras which includes ice, which increases during June, July and august which may be due to the hot weather in these months. call out a category We believe there is a lot more information hidded in this graph and a potential for further time series analysis. We have devoted part 2 to further explore the trend of sales.

conclusion of category level insights

- We observed that coffee contributed highest to the overall revenue the most as expected followed by food and beans.
- We observed that the discount on beer wasnt reflecting in the increased sales and would recommend to remove beer all together from the menu.
- We saw that the price distribution for all items in beas was the highest and we beieve we should market beans more. We believe there is a lot of potential to increase revenue by upselling beans.

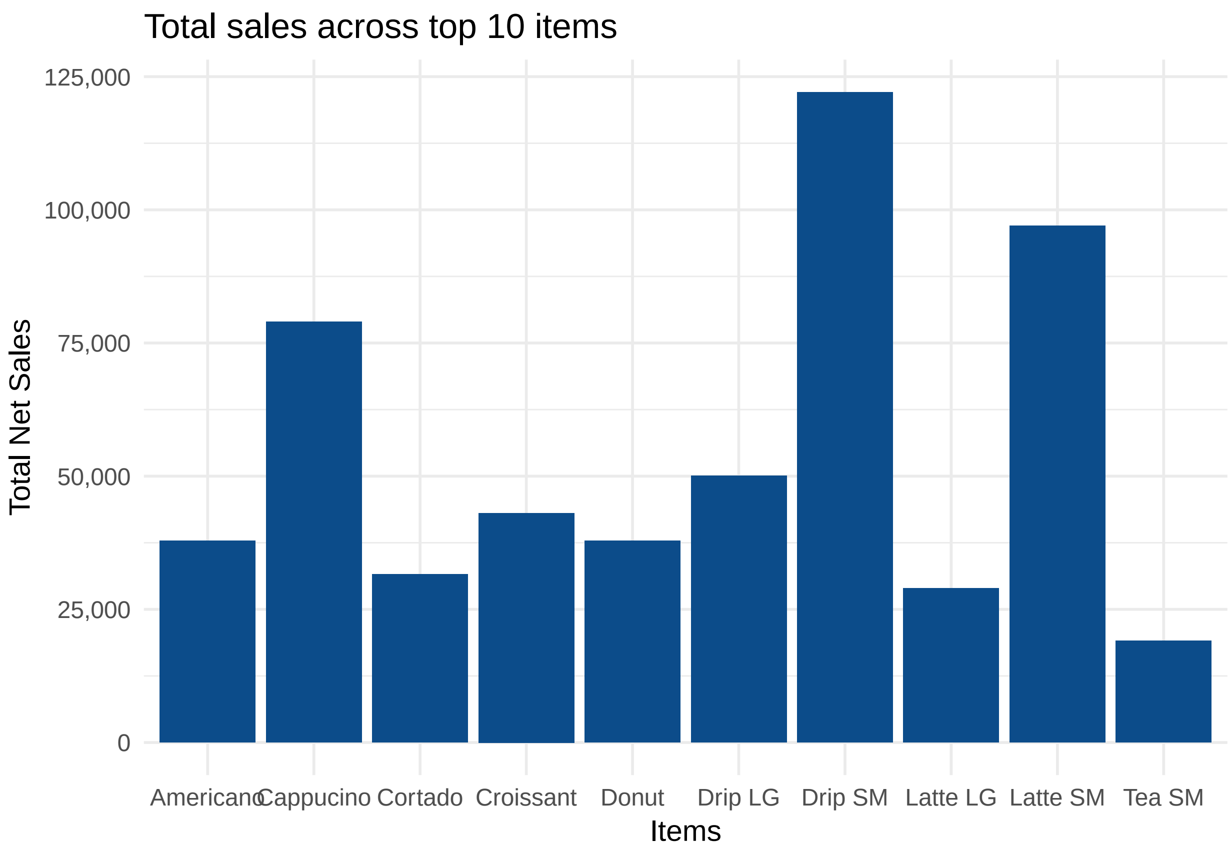
Item level data exploration

At a high level we have been exploring 7 distince category. As we are aware there are a lot of items that compose each category. Behavior of each item within a catgory might vary. Therefore we want to understand the behaviour(contribution) of the top selling items and botomn selling items Now we wanted to understand

We wanted to explore the top 10 items according to revenue and bottomn 10 items which yield us the least revenue cross category.

```
item_grp <- data %>% group_by(Item) %>% summarise(net = sum(Net.Sales)) %>% arrange(desc(net))
```

```
top_sold_item <- head(item_grp,10)
ggplot(data = top_sold_item) +
  aes(x = Item, weight = net) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Total sales across top 10 items",
    x = "Items",
    y = "Total Net Sales") + scale_y_continuous(labels = comma)  +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10))+
  theme_minimal()
```



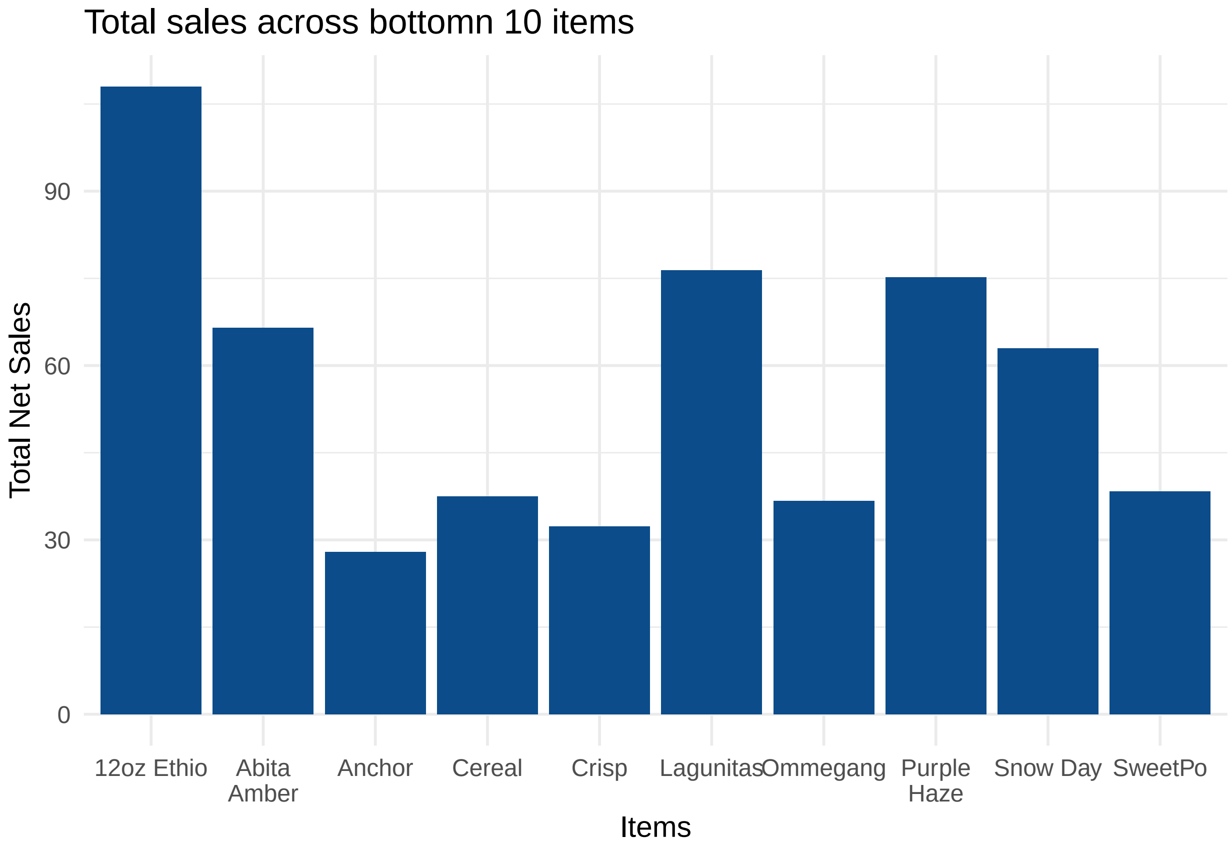
Description:

The graphs above captures total net sales for top 10 items cross all categories.

Interpretation:

We observe that highest net sales is from dfferent varieties of coffee, team and only 2 food variants.

```
bottomn_sold_item <- tail(item_grp,10)
ggplot(data = bottomn_sold_item) +
  aes(x = Item, weight = net) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Total sales across bottomn 10 items",
    x = "Items",
    y = "Total Net Sales") + scale_y_continuous(labels = comma) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10))+
  theme_minimal()
```



Description:

The graphs above captures total net sales for bottomn 10 items cross all categories.

Interpretation:

We observe that there are so may items in the menu that are rarely bought and have insignificant contribution to the overall revenue. We believe we can reduce cost by removing few of these items such as cereals and types of beer from the menu.

Conclusion of item level

- We observed that there were only 2 food variants that contributed to revenue and recommend to introduce more items under food category.
- We can remoe few items such as cereals and beer from the menu which are least sold and save on the inventory cost and other cost incurred.

Currently we have considerd items as a whole and did all our analysis. We wanted to see the distribution of transactions across different time of day and days of week. For this analysis we have to transform data to a transaction level.

Data Transformation

Creating transaction level data

Exploring the data, we saw that there is no unique transaction id for each timestanp. lot of value -> grp of items -> purchased in each transaction -> which prodecuts sell well with each other and average value of purchase per customer etc. To achieve this we transformed the data to a transactionlevel. We transformed the item level data to transaction level to understand when the sales were the highest during different time of day and different days of the week.

We grouped the dataset on a time and Customer ID level to get the unique transactions.

```
#grouping data by trasaction data high level
transaction_data <- data %>% group_by(time,Customer.ID) %>% summarise(total_qty=sum(Qty),
                                                                      transactions = n())
```

Assumption * We assumed that all items bought at a timestand as one transaction to transform the data into transaction level.

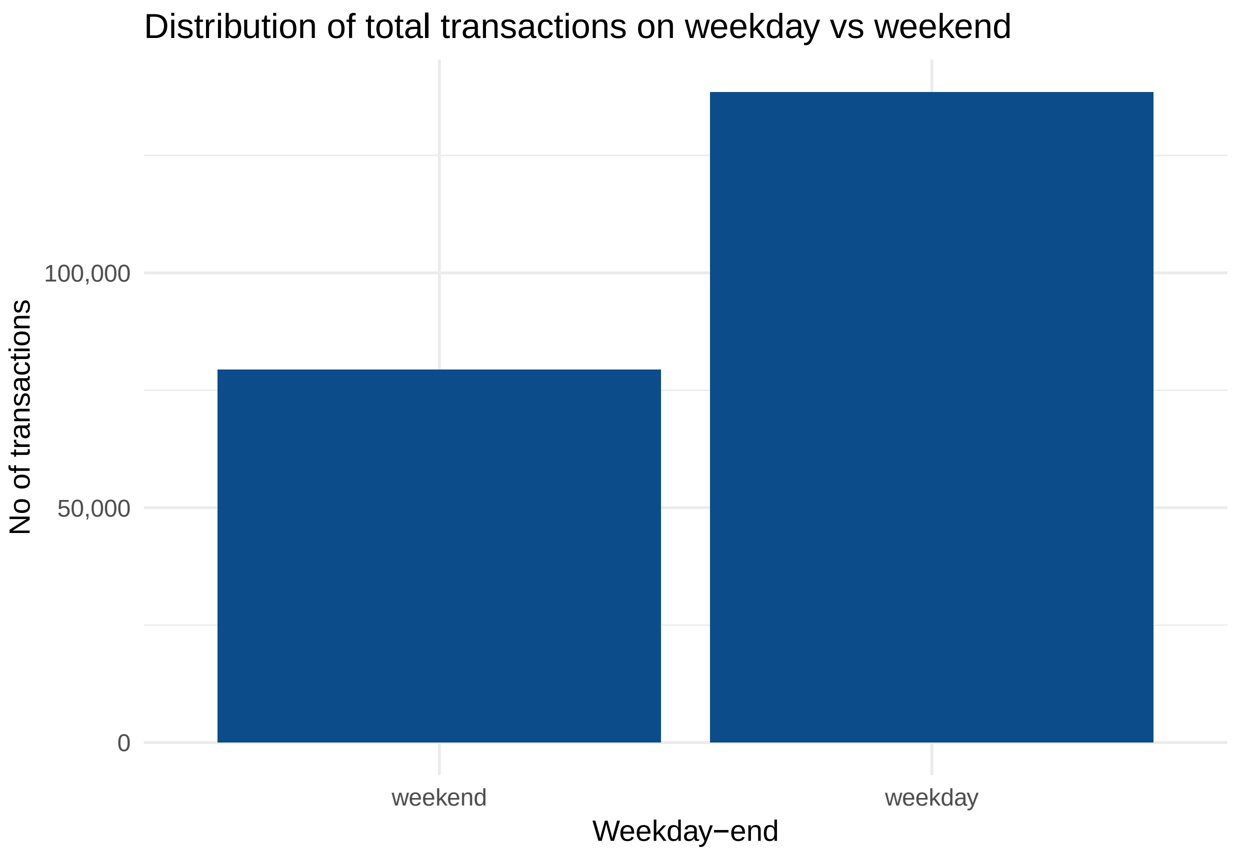
Adding 3 new columns to the transaction level dataset to bucket the transactions into different days of the week, weekends and weekdays and different time of the day. We have binned time into 3 groups, morning(6AM to 11AM), noon(11Am to 4PM) and night($PM to end of day).

```
weekdays1 <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
transaction_data$weekday_end <- factor((weekdays(transaction_data$time) %in% weekdays1),
                          levels=c(FALSE, TRUE), labels=c('weekend', 'weekday'))
transaction_data$time_of_day <- cut(lubridate::hour(transaction_data$time),
                                    breaks = c(4,11,16,23),
                                    labels = c('morning', 'noon', 'night'))
transaction_data$day <- factor((weekdays(transaction_data$time)))
```

Part 1 - Buying Pattern Analysis

Exploratory analysis of Transactional level data.

```
#Weekday,weekend distribution of sales

ggplot(data = transaction_data) +
  aes(x = weekday_end, weight = transactions) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Distribution of total transactions on weekday vs weekend",
    x = "Weekday-end",
    y = "No of transactions") + scale_y_continuous(labels = comma) +
  theme_minimal()
```



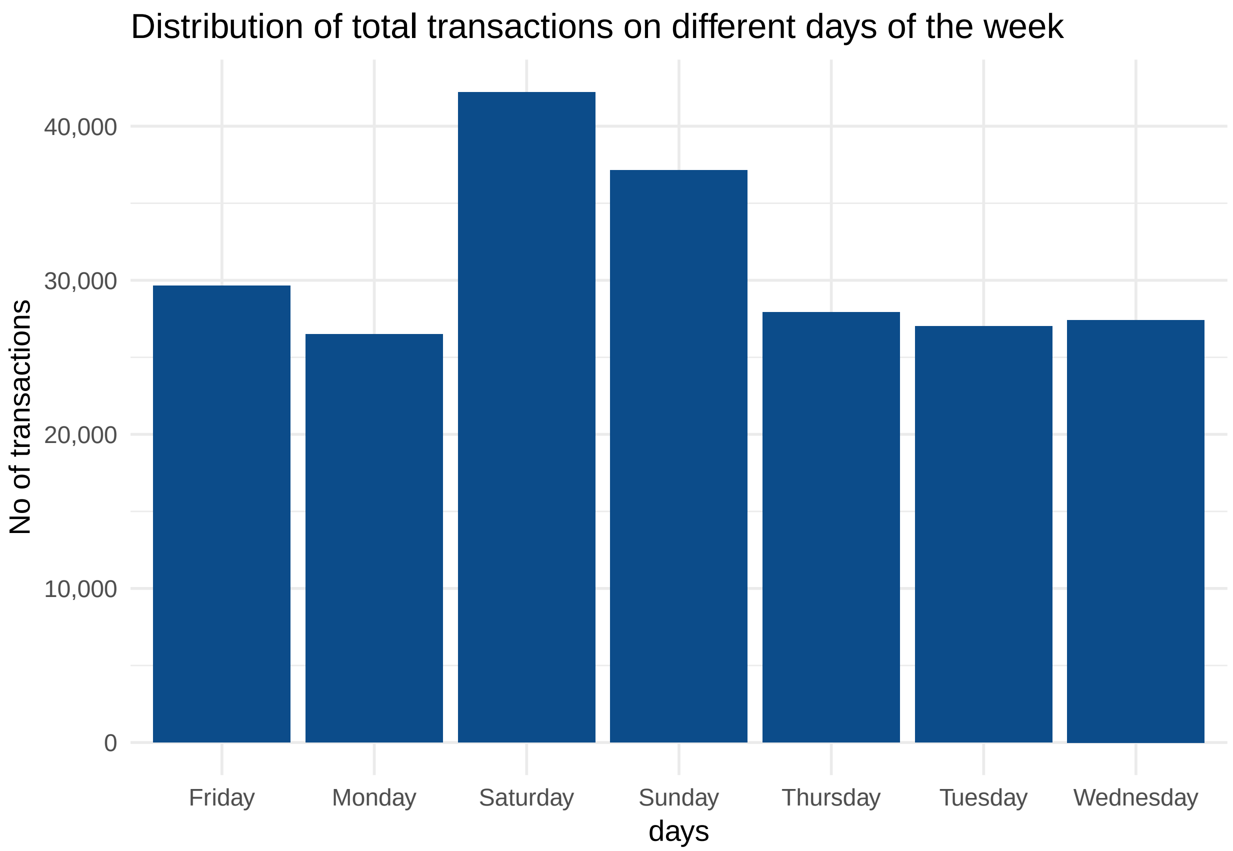
Description:

The graphs above captures the Distribution of total transactions across weekends and weekdays.

Interpretation:

We observe that the number of transactios on weekdays is higher comapred to weekends.

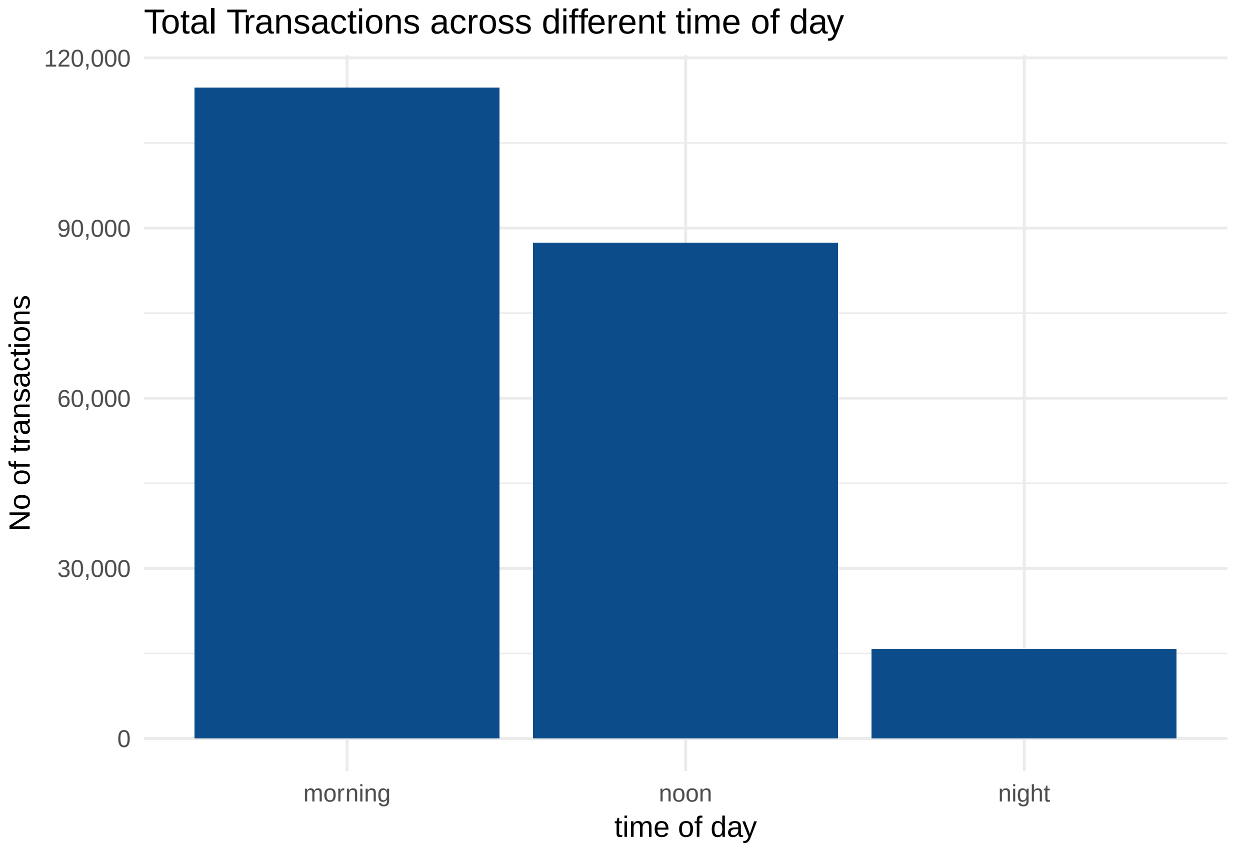
```
#distribution of sales across days of week

ggplot(data = transaction_data) +
  aes(x = day, weight = transactions) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Distribution of total transactions on different days of the week",
    x = "days",
    y = "No of transactions") + scale_y_continuous(labels = comma) +
  theme_minimal()
```



Description: The graphs above captures the Distribution of total transactions across different days of the week.

Interpretation: We observe that the No of transactions is highest on saturday and sunday. The sales is about the same across other days.

```r
#distribution of sales across time of day

ggplot(data = transaction_data) +
  aes(x = time_of_day, weight = transactions) +
  geom_bar(fill = "#0c4c8a") +
  labs(title = "Total Transactions across different time of day",
    x = "time of day",
    y = "No of transactions") + scale_y_continuous(labels = comma) +
  theme_minimal()
```



Description:

The graphs above captures the Distribution of total transactions across different time of day.

Interpretation:

We observe that the bulk of transactions are in the morning, which makes sense as morning coffee is something that customers prefer and we see a high footfall in the mornings, followed by noon and very less in the evening.

Conclusion transaction level data

- We observe that there is higher number of transaction in the mornings compared to noon and very less numberof transactions in the evening. We want to explore further on the items sold in different parts of the day to give actionable recommendation to smoothen the sales across different time of the day.
- We observed that there is a higher footfall during saturday and sunday compared to other days of the week. We want to explore further on what items are bought on these days and come up with a recommendation to smoothen the demand over all days of the week.

Association rules

We explored the data in different levels until now. We wanted to see if there are any patterns in the items purchased by cutomers in each transaction. We want to use this information to cross sell items that have high lift to increase the revenue from our existing customers. We decided to run arules on the transaction level dataset to obtain the patterns that we can utilize to come up with valid recommendations.

Data transformation into transactions for association rule

To get the transaction level data, we add another column called rank which has the same value for all rows which are bought at a particular timestamp.

```
#transaction level data for association rules
arules_input = data %>% mutate ( rank=  dense_rank (time))
# creating transaction in the format required for arules input
trans <- as(split(arules_input[,"Item"], arules_input[,"rank"]), "transactions")
```

We now have the data at a transaction level in the format we need to run arules. Running apriori algorithm on this dataset.

```
# apriori algorithm with min support and confidence as 0.01.
rules <- apriori(trans,parameter=list(supp = 0.01 , conf=0.01))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.01    0.1    1 none FALSE            TRUE       5    0.01      1
##  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: 1329
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[69 item(s), 132955 transaction(s)] done [0.02s].
## sorting and recoding items ... [23 item(s)] done [0.00s].
## creating transaction tree ... done [0.04s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [51 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#filtering the rules by lift
rules_s = sort (rules , by = 'lift' , decreasing = TRUE )
#subset the rules for lift greater than 1
rules.sub <- subset(rules_s, subset = lift > 1)
inspect(rules.sub)
```

```
##      lhs             rhs         support    confidence lift     count
## [1]  {Latte SM}   => {Almond}    0.02925050 0.18249648 3.289117  3889
## [2]  {Almond}     => {Latte SM}  0.02925050 0.52717907 3.289117  3889
## [3]  {Ice}       => {Drip SM}   0.14714753 0.64425198 2.204915 19564
## [4]  {Drip SM}   => {Ice}       0.14714753 0.50360379 2.204915 19564
## [5]  {Almond}    => {Cappucino} 0.01185363 0.21363698 1.554260  1576
## [6]  {Cappucino} => {Almond}    0.01185363 0.08623803 1.554260  1576
## [7]  {Latte SM}  => {Ice}       0.05040051 0.31445331 1.376762  6701
## [8]  {Ice}       => {Latte SM}  0.05040051 0.22066717 1.376762  6701
## [9]  {Croissant} => {Cappucino} 0.01213945 0.16183696 1.177403  1614
## [10] {Cappucino} => {Croissant} 0.01213945 0.08831737 1.177403  1614
## [11] {Tea SM}    => {Ice}       0.01292919 0.23170239 1.014456  1719
## [12] {Ice}       => {Tea SM}    0.01292919 0.05660750 1.014456  1719
```

Interpretation

From the above rules we see that the items having lift greater than 1 are genrally combination of coffee with food and extras. These rules have comparitively lower support and confidence. Hence we want to apply apriori algorothm on our loyal customer base to see any strong patterns that would give us an opportunity to increase our revenue from the existing customer base.

Recommendations

- We believe that we can cross sell food items with coffee. Since we have very less number of variety in food category, we recommend to introduce other varieties such as muffins, sandwitches under food which will increase our revenue.
- We also observe a pattern where almond comes up with coffee in many rules. We recommend to introduce hazlenut and vanilla in extras as we believe customers might preffer different flavours in their coffee.

Assumptions

We assume introducing variety of new food items and flavours for coffee will increase the customers to buy these items. We suggest to implement these recommendations in a small scale to see the margin of increase in revenue before implementing in a large scale.

Part 2 - Understanding customer base

Clustering of cutomers

Why Clustering?

We believe that by identifying segments of customers, We will be able to get a better understanding of customer behavior and loyalty . The segements could be identified in a way that ensures that the customers within the segment exhibit similar behaviour and customers across segments behave differently.

Given that there are no existing segments available, we use an unsupervised clustering alogrithm to identify the segments.

We hypothesized that we will get clusters which differentiate the loyal customer with others through this clustering algorithm. We used three metrics - recency, monetary, frequency (rmf) to categorize our customers, and then tried to find behavior pattern for each cluster.

RFM (recency, frequency, monetary) is used to determine quantitatively which customers are the best ones by examining how recently a customer has purchased (recency), how often they purchase (frequency), and how much the customer spends (monetary). We clustered customers using these variables.

Creating customer level data for clustering

Assumption

We believe customers with Customer ID represent the behavious of our loyal customers. Hence removing non-members from our analysis wont affect the outcome of analysis.

```
# Selecting only the customers with a valid Customer ID.
data_cluster <- filter(data,Customer.ID != 'Non-Member')
data_cluster$time <- as.character(data_cluster$time)

# calculate the frequecy of purchases for each custoemr
data_cluster_fre <- data_cluster %>% group_by(time, Customer.ID) %>% summarise(fre=n())
data_cluster_fre <- data_cluster_fre %>% group_by(Customer.ID) %>% summarise(fre = n())

data_cluster$time <- as.Date(data_cluster$time)
max_time <- max(data_cluster$time)
# calculating the sum of net sales for each custoemr and the last day
#since he visited the the cafe
data_cluster <- data_cluster %>% group_by(Customer.ID) %>%
  summarise(sum_sales = sum(Net.Sales),
            days_interval = as.numeric(max_time - max(time)))
# Merging the above two data frames to create the customer level dataset
#we need for the clustering algorithm.
data_cluster <- merge(data_cluster, data_cluster_fre, by = 'Customer.ID')
data_selected <- data_cluster[,c('sum_sales','days_interval','fre')]
```

Partitioning methods Partitioning clustering is used to classify observations into multiple groups based on their similarity. The paritioning algorithm works by iteratively re-allocating observations between clusters until a stable partition is reached. However, the number of clusters need to be specified by the user. Similarity is calculated based on distance calculations.

We go ahead with the partitioning clustering method, with k-means.

Rescaling data

Description

To apply distance based clustering, the first step is to rescale the numeric data columns ie., all numeric columns should have the same range of values. This process called normalization, will help us in handling columns that have varying scales. By normalizing, we rescale the data to a standardized scale, making the distance measures comparable.

There can be two ways in which the data can be rescaled: * Min-Max Normalization – The data is rescaled to a 0-1 scale * Standardization – The data is assumed to be normal and scaled to have a mean of 0 and a standard deviation of 1

Min_Max Normalization

In this normalization approach we bring all numeric columns to the range of 0 and 1 with 0 being the lowest value in the column and 1 being the highest value in the column. All other values are normalized based on the following formula:

```
**Yi = [Xi - min(X)]/[max(X) - min(X)]**
```

```r
# data normalization
minmax <- function(x){
  x <- (x - min(x)) / (max(x) - min(x))
  return(x)
}
data_normalize <- data_selected
data_normalize$sum_sales <- minmax(log(data_normalize$sum_sales))
data_normalize$fre <- minmax(log(data_normalize$fre))
data_normalize$days_interval <- minmax(data_normalize$days_interval)
```

k-means Assumptions and Limitations

- Can create clusters with a specific shape only – Since we have no idea of how the actual clusters will look like, we can assume that the clusters we obtain out of the algorithm are spherical in shape as we use the Euclidian distance measure
- Can work with numerical data only – Our dataset has only numerical clusters, and hence, there is no problem
- The number of clusters (k) needs to be specified before clustering – We will evaluate the clustering performance and choose the clusters based on the results
- Highly sensitive to outliers – Our data has been treated for outliers. Therefore, there would be no impact of outliers
- Cannot capture hierarchical structure – Since, we have not observed any significant results out of hierarchical clustering, we can infer that there is no hierarchical structure
- Hard Clustering – The customers are clustered into one group and one group only. It may be possible that a customer might belong to two different groups when his travel habits differ. But, given our original assumption that the behaviour is stable for the period under consideration, we can neglect this for the scope of this analysis. This assumption could be re-evaluated and restested in the next phase of the segmentation
- Convergence to local minima - k-means could converge to local minima instead of the global minima. The convergence should be evaluated by running multiple instances to identify whether similar results are being obtained across runs

The first step in the k-means algorithm is in choosing the value of k. To identify the value of k, we evaluate the clustering algorithm for different values of k and choose a k depending on the cluster performance.

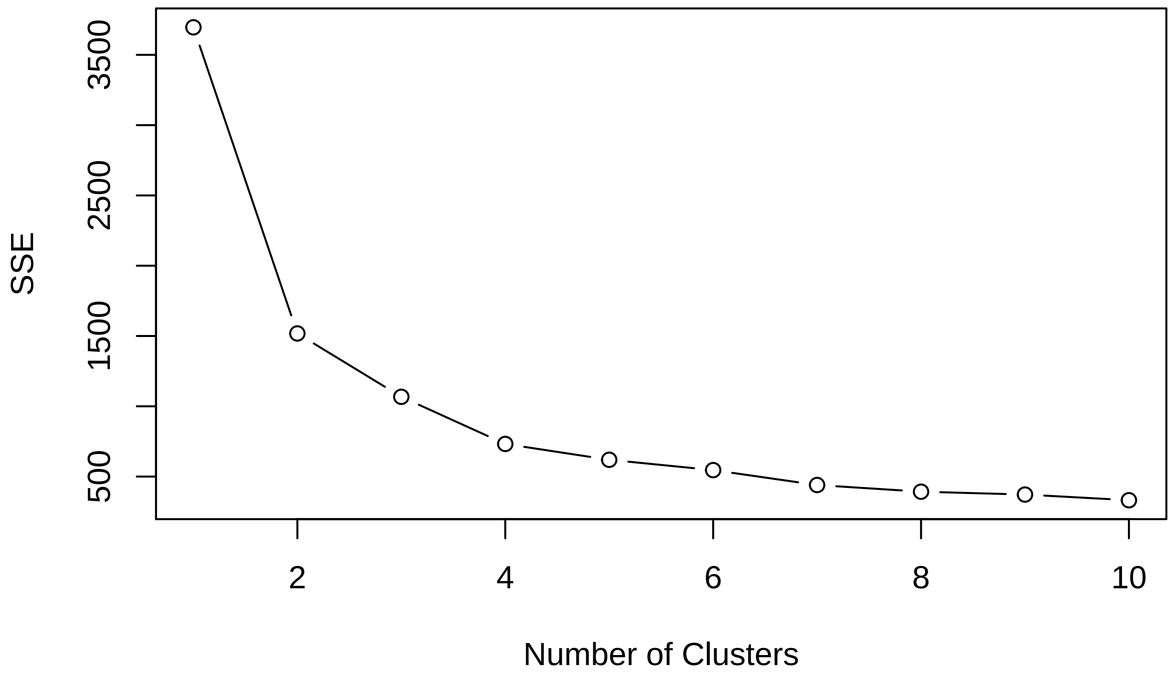
Evaluating Clustering Performance

The clustering performance depends on the number of clusters we choose. The clusters formed should be such that there is high similarity within a cluster and low similarity between the clusters.

We are looking at the two metrics to evaluate that the clustering performance: * SSE (Sum of Squared Errors) – SSE captures the sum of squared distance between each point and its centroid. Therefore, lower the SSE, higher the similarity between the point and its cluster

Elbow curve

```
# use kmeans to conduct cluster
SSE_curve <- c()
for (k in 1:10) {
  kcluster <- kmeans(data_normalize, k)
  sse <- sum(kcluster$withinss)
  SSE_curve[k] <- sse
}
plot(1:10, SSE_curve, type="b", xlab="Number of Clusters", ylab="SSE")
```



From the plot, we can see that the for K = 4 the SSE drop is steep and after K = 4 the SSE drop is gradual.

Creating Clusters

Applying the K-means algorithm on the transformed and normalized data with the number of clusters as 4.

```
set.seed(123)
set.seed(3) # for reproducibility
```

```
# the plot shows that best clusters should be 4 - 6, after further descriptive analysis
#, we found that 4 is the best number to categorize our customers.
num_clusters = 4
kcluster <- kmeans(data_normalize, num_clusters)

# assign the cluster information to each customer
data_cluster$cluster <- kcluster$cluster
```

From the results of the cluster , we can categorize these 4 clusters into 4 types of customers based on recency(days_interval), monetary(sum_sales), and frequency(fre). And we add the non-membership customers to form 5 types of customer segments as following:

- non-membership (we need to manage to motivate them to be memberships.) These are the customers with no customer ID present in the data.
- tourists (high recency + low monetary + low frequency, they are customers who have ever been to our store for few times during these two years but not recently) – 17908 customers
- potential new customers (low recency + low monetary + low frequency, they are customers who came to our store recently but they are relatively new to us, we need to manage to capture these customers to our loyal customers) – 10298 customers
- ex-loyal customers (high recency + high monetary + high frequency, they are customers who used to consume in our store for a few times but haven't come again for a long time, we treat them lost loyal customers, we need to try to find out the reason) – 1560 customers
- loyal customers (low recency + high monetary + high frequency, they are our most loyal customers, we need to maintain the relationship with them well) – 2045 customers

Cluster Profiling

Customer segments provide clear information with respect to which customers fall under which segment. This understanding is crucial and will be leveraged for decision making. The output of a clustering algorithm doesn't explain what each cluster comprises of. If and only if the cluster composition is explained, the mathematically-derived clusters become business-consumable customer segments.

After obtaining the clusters, it is imperative that we understand what observations fall under each cluster. This helps us in understanding the patterns that make up the cluster. Cluster profiling is the method by which we try to explain the similarity within clusters and identify patterns that make up the cluster.

Mapping clusters and raw data

The clusters identified are first mapped to the original dataset to identify what set of customers make up each cluster. We have given a name to each cluster as mentioned above.

```
# assign segment names to each item record for further analysis
data_segmented <- merge(data, data_cluster[,c('Customer.ID','cluster')],
                        by = 'Customer.ID', all.x = TRUE)
data_segmented$cluster <- ifelse(is.na(data_segmented$cluster), 'non-membership',ifelse(data_se
ifelse(data_segmented$cluster == 5,'ex_loyal_customers','new_customers'))))
```

Exploration of customers across different segments

We wanted to understand the contribution of different segments of customers to net sales. We have calculated this below.

```r
# further analyze the size of each cluster in terms of contribution to sales
contribution <- data_segmented %>% group_by(cluster) %>%
  summarise(contributions = sum(Net.Sales))
contribution$contributions <- round(contribution$contributions/
                                      sum(contribution$contributions),2)
print(contribution)
```

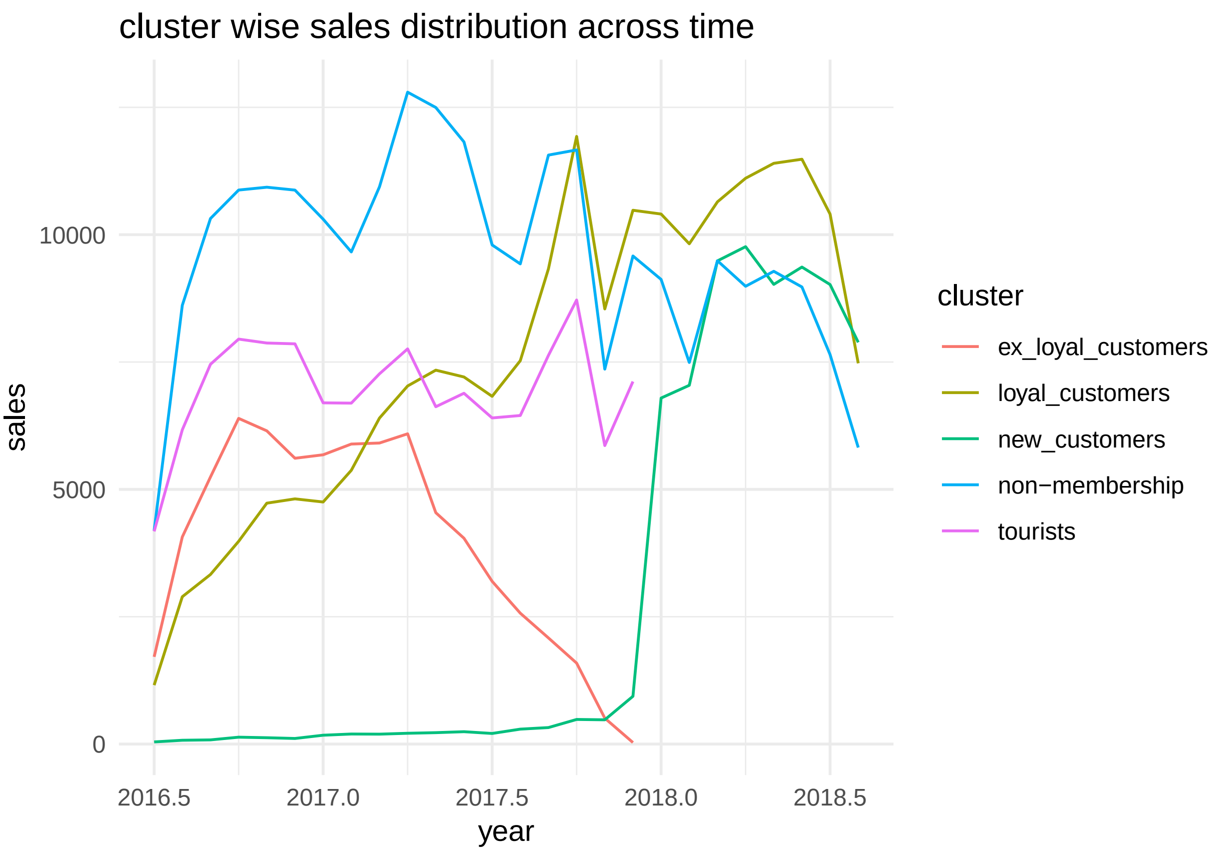
```
## # A tibble: 5 x 2
##   cluster            contributions
##   <chr>                      <dbl>
## 1 ex_loyal_customers          0.1
## 2 loyal_customers             0.27
## 3 new_customers               0.1
## 4 non-membership              0.35
## 5 tourists                    0.18
```

We can see the contribution to the total sales for each cluster, it shows that the store has a strong loyal_customers which contributed 27% of the total net sales. However, We also see that the lost loyal customers contributed 10%, which is relatively significant. Therefore, we need to work harder on relationship mantainence with loyal customers and keep developing more loyal customers from the cluster 'new customers'. We also would want to convert the non-members into our loyal customers to increase the revenue with existing customer base.

Exploring the net revenue distribution of segments across time

We saw the total net sales across different customer segments. Now we want to visualise this ditribution across time. We plot a line graph to understand this trend.

```r
library(tidyquant)
cat_trend <- data_segmented %>% group_by(cluster,time) %>%
  summarise(sales = sum(Net.Sales))
cat_trend$Month_Yr <- format(as.Date(cat_trend$time), "%Y-%m")
cat_trend$Month_Yr <- as.yearmon(cat_trend$Month_Yr)
req_data <- cat_trend %>% group_by(Month_Yr,cluster) %>% summarise(net = sum(sales))
plotC <- ggplot(req_data, aes(x=Month_Yr, y=net, color = cluster, group = cluster))
plotC +geom_line() + labs(title = "cluster wise sales distribution across time",
       x = "year",y = "sales")  +  theme_minimal()
```



Description:

The graphs above captures the Distribution of net sales across time for different customer segments we have observed. We have used this graph to name the customer segments.

Interpretation:

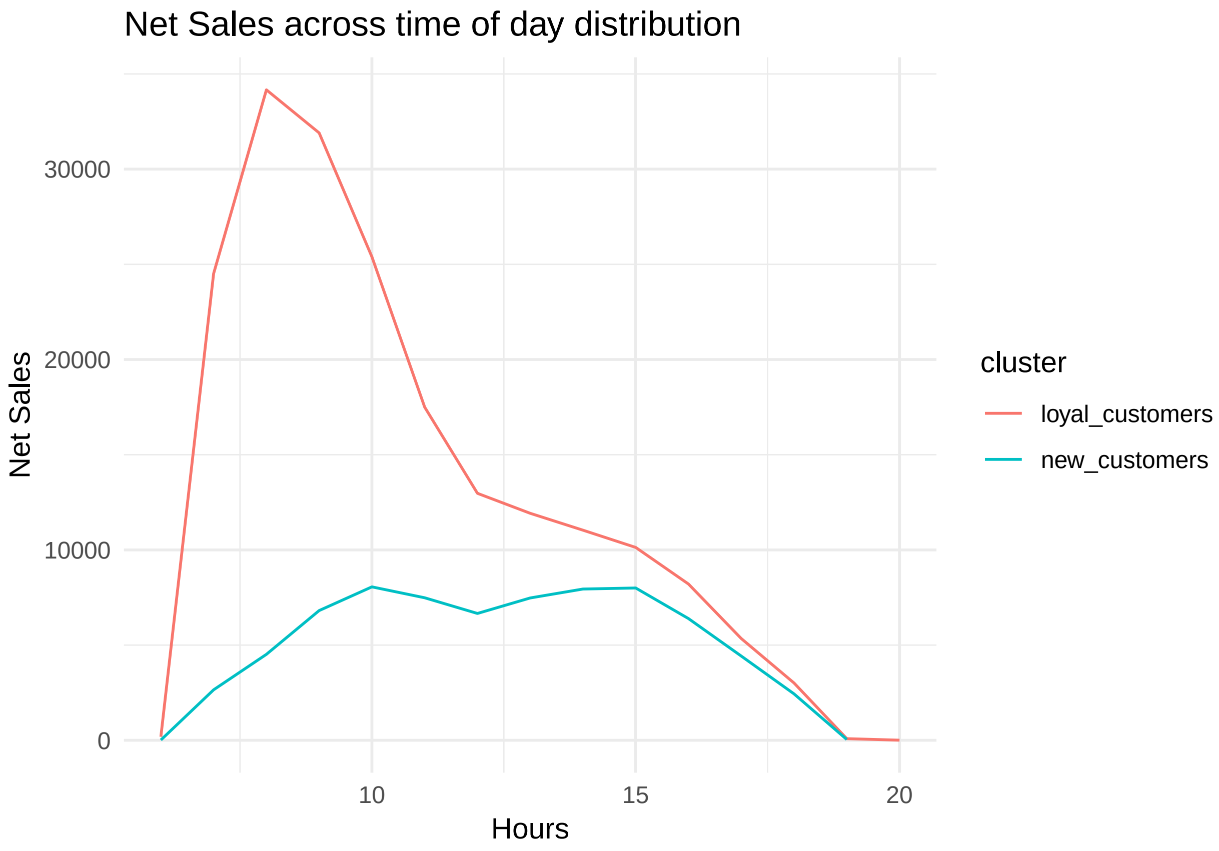
- new_customers segment - the sales have risen significantly from year 2018. We believe there is an opportunity to convert this customer segment into our loyal cutomers. We want to further analyse the behavior of this customer segment to get insights to incrase sales contribution from this segment.
- loyal customers - We see the sales for loyal customers have a significant contribution to overall sales. This customer segment has 2045 customers but generate 27% of the sales. we want to further analyse this customer segment as we see a high potential to increase revenue from this segment.
- We observe that the sales from non members are decreasing gradually. We believe that non-members are becoming members which might be the reason for this decrease in sales.
- tourists and ex-loyal customers - the segments we got from the clustering algorithm shows us that till 2018 we had good contribution to sales from these two segments. but we see that these segments have to contribution after 2018.

Therefore we want to concentrate all our analysis on new customer segment and loyal customer segment to increase sales.

We analised the volume of sales across the 2 years in the above analysis. We now want to compare the sales distribution of the two categories(new customers and loyal customers) across time of day.

Demand across time across clusters

```
hrdata <- data_segmented %>% filter(cluster %in% c('new_customers','loyal_customers'))%>%
  group_by(hr = hour(time),cluster) %>% summarise(sales = sum(Net.Sales))

ggplot(data = hrdata) +
  aes(x = hr, y = sales, color = cluster, group = cluster) +
  geom_line() +
  labs(title = "Net Sales across time of day distribution",
       x = "Hours",y = "Net Sales") +
  theme_minimal()
```



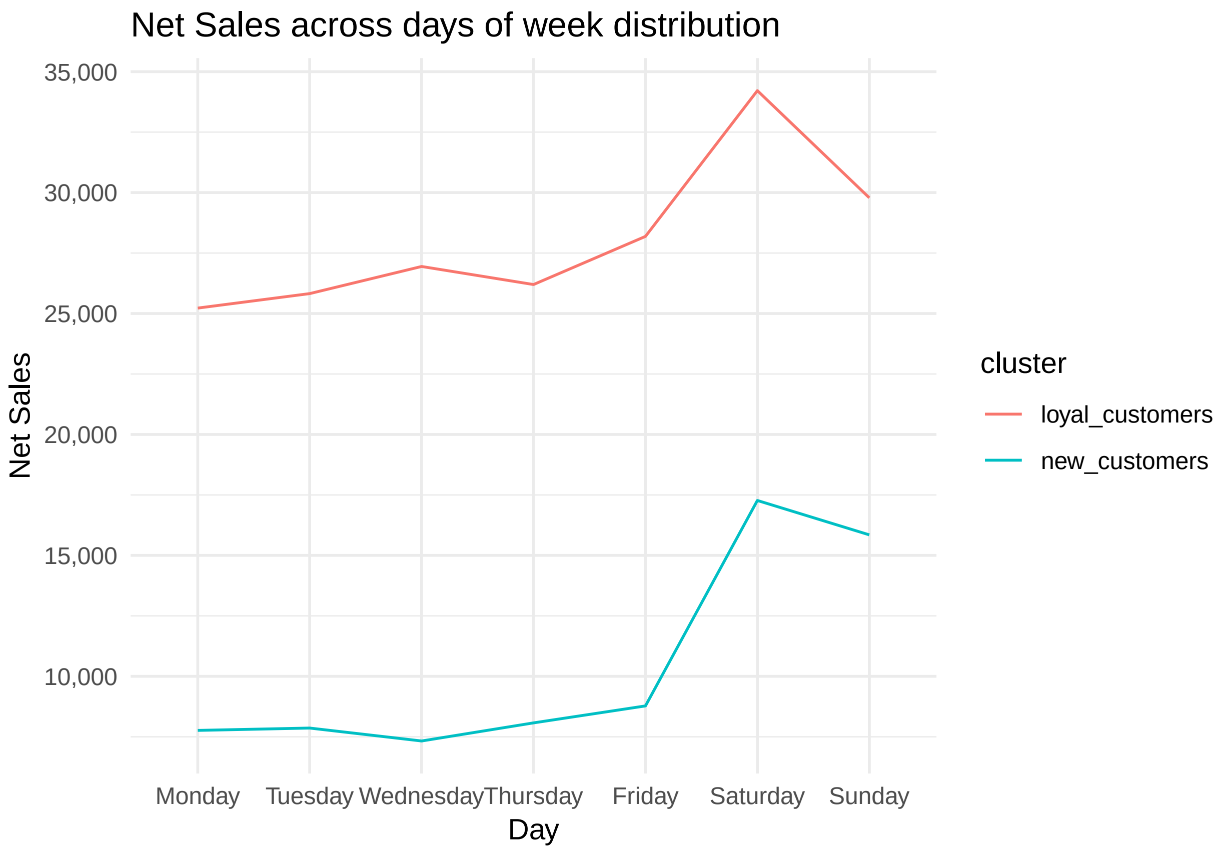
Description:

The graphs above captures the Distribution of net sales across time of day for new_customer and loyal customer segment.

Interpretation: The typical day at Central Perk sees demand peaking around 9 am and sharply falling thereafter to 12 pm where there is a slow drop off in demand until the end of the day. To help alleviate some of these peaks and to attempt to build a new group of loyal customers we believe Central Perk should rework their product offerings. new_customer segment is made up of potential loyal customers which we want to target with new offerings. This group has a greater preference for coming during the lunch hour and we recommend introducing a new line of sandwiches and salads to appeal to this group. By attracting more of the customers coming throughout the day it can help alleviate some of the volatility in demand because of potential new customers steady demand throughout the day. Not only is expanding the menu potentially beneficial for potential loyal customers we believe this can help increase sales to our loyal members. Our loyal members are two times as likely to visit us multiple times a day compared to non loyal members. By offering a 50% discount on a members second coffee in a day with the purchase of a sandwich or salad we can profit 0.95 dollar per transaction (assuming $10 sandwich and small coffee)

Demand across Day of the week across clusters

```r
data_segmented$time <- as.Date(data_segmented$time)
dydata <- data_segmented %>% filter(cluster %in% c('new_customers','loyal_customers'))%>%
  group_by(dy = as.factor(weekdays(time)),cluster) %>%
  summarise(sales = sum(Net.Sales))

dydata$dy <- ordered(dydata$dy, levels=c("Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"))
ggplot(data = dydata) +
  aes(x = dy, y = sales, color = cluster, group = cluster) +
  geom_line() +
  labs(title = "Net Sales across days of week distribution",
       x = "Day",y = "Net Sales") + scale_y_continuous(labels = comma) +
  theme_minimal()
```



Description:

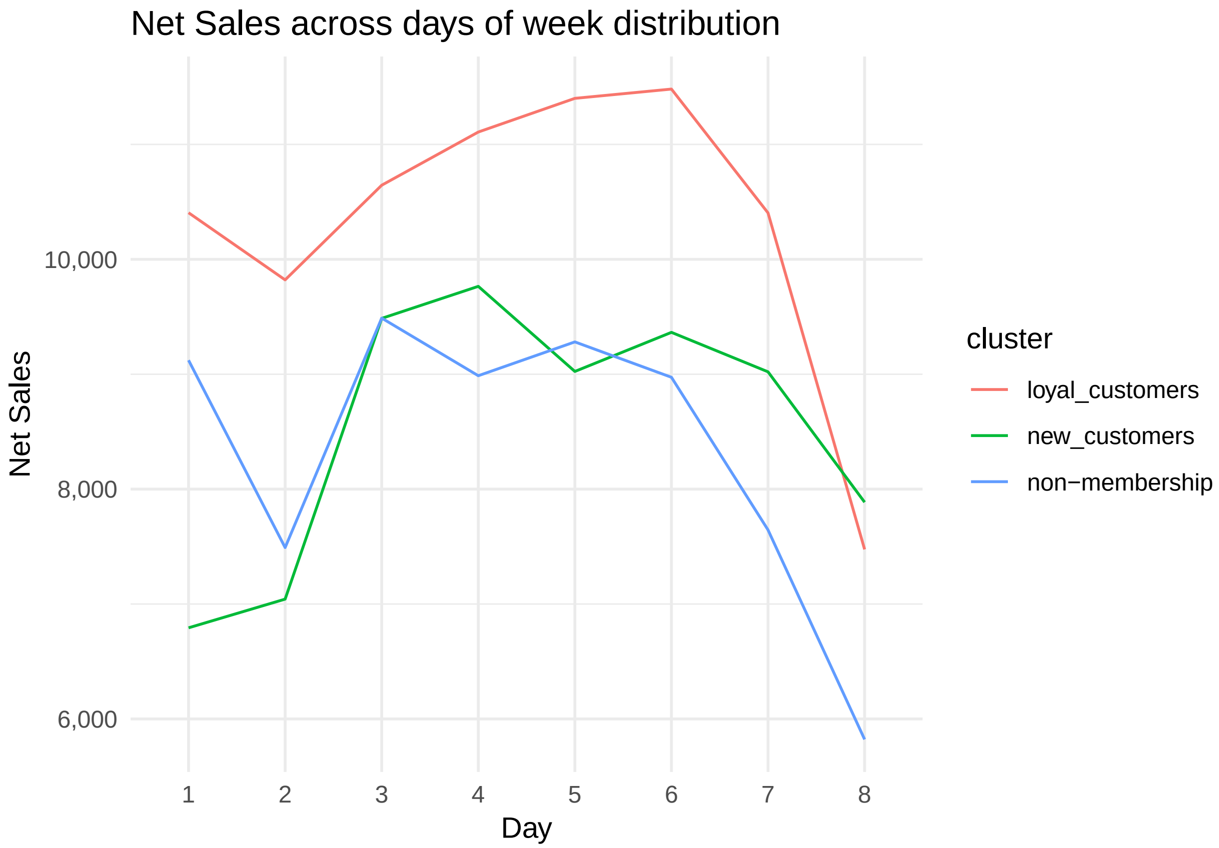
The graphs above captures the Distribution of net sales across days of the week for new_customer and loyal customer segment.

Interpretation:

A typical week at central perk sees an increase in sales towards to end of the week on saturday and sunday. To smoothen this demand, we want to introduce discounts suggested earlier on the weekdays.

To give a actinable recommendation, we would like to see how much net sales is contrinuted by our loyal customers and new customers in the year 2018.

```
data_segmented$time <- as.Date(data_segmented$time)
dydata <- data_segmented %>% filter(year(time) == 2018) %>%
  group_by(dy =as.factor(month(time)),cluster) %>% summarise(sales = sum(Net.Sales))

ggplot(data = dydata) +
  aes(x = dy, y = sales, color = cluster, group = cluster) +
  geom_line() +
  labs(title = "Net Sales across days of week distribution",
       x = "Day",y = "Net Sales") + scale_y_continuous(labels = comma) +
  theme_minimal()
```



```
calc <- data_segmented %>% filter(year(time) == 2018) %>% group_by(cluster) %>%
  summarise(sales = sum(Net.Sales))
calc <- calc %>% mutate(percent_share = (sales/sum(sales)*100))
calc
```

```
## # A tibble: 3 x 3
##   cluster          sales percent_share
##   <chr>            <dbl>         <dbl>
## 1 loyal_customers 82741.          38.0
## 2 new_customers   68382.          31.4
## 3 non-membership  66810.          30.7
```

Description:

The graphs above captures the Distribution of net sales across time for the year 2018 for our customer segments.

Interpretation:

We see that the loyal customer segment has contributed to nearly 38% of the total net sales in the year 2018. and the new customers have contributed around 31% of the total net sales. We believe all the marketing strategies should be directed towards these segments to maintain this customer base and increase sales from our existing customer base. We also observe a reduction in non-member contribution to net sales.

Assumptions: We assume there are no loyal customers in the new_customer segment and all our loyal customers are segmented under loyal customers.

Conclusion and Recommendation

Recommendation to smoothen the sales

- While we did identify a subset of customers who we would define as loyal, we had some concerns regarding relying on them for the bulk of the shop's revenue going forward. It is recommended that a shop's loyal customer base should comprise of 20% of their customers and 50% of their revenue (Loyalty Definition) and our loyal customers only make up 6% of customers with membership and they only accounted for 38% of revenue in 2018. Additionally, cluster 4 is defined as Ex Loyal Customers, customers who we believe used to regularly buy coffee from Central Perk. If we are unable to address or identify issues which may be turning Loyal Customers into Ex Loyal Customers and with percent revenue behind where we would like we may not be in a good position to solely focus on the Loyal Customers. This is why we recommend focusing Central Perk's attention on building an additional group of loyal customers and maintaining their current loyal customer base to normalize demand and generate additional revenue.

Recommendation to increase revenue from existing customer base

- One characterization that was noticed was the amount of loyal and ex loyal customers that come for coffee around 8 am. This is routinely one of the busiest times of the day (regardless of day of week or season) and we believe people may be leaving due to long wait times. To help smooth out the demand during this time we recommend that we offer a prepaid coffee card. The card will cost $32.50 and will entitle the customer to 10 small drips, the item most purchased in the morning. With this pricing we will make $4.50 per card as opposed to $7.50 if they bought each coffee individually. Customers coming in with the card will be able to have their card quickly punched and receive a coffee. With a quicker transaction time we can reduce the morning rush congestion and provide incentives for customers to prepay for coffee in advance. With this method we may see decreased revenue from loyal customers taking advantage of this deal but we can expect that this will be offset by people not redeeming the full worth of their cards (Source). By offering the prepaid card it will help smooth the demand during the morning rush and eliminate something that may be turning our loyal customers into ex loyal customers.

Part 3 - Price sensitivity

After understanding the customer behaviors of different customer segments. we now wanted to explore which customer segment are more price sensitive compared to others. And also want to see the variation of sales in each category with price per item.

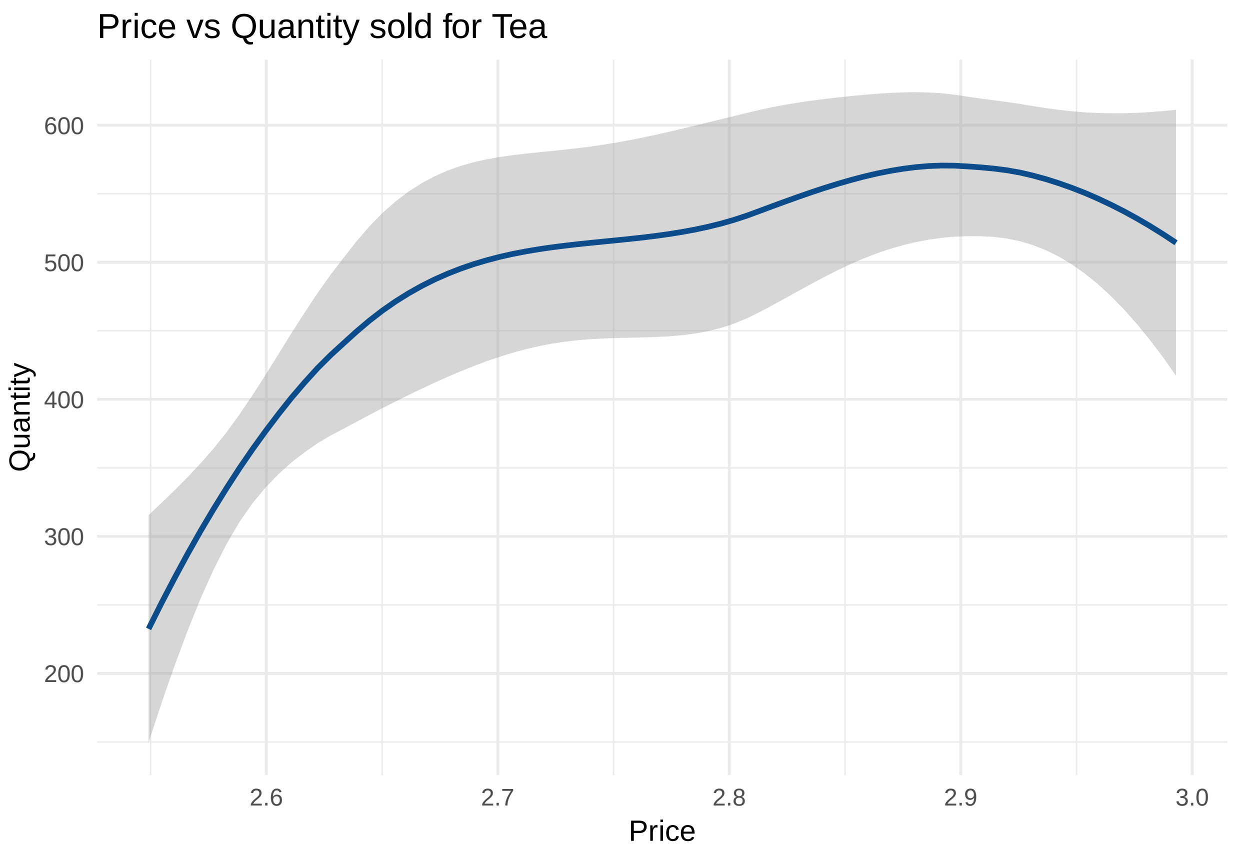
Data preparation for analysis of price with categories and among clusters.

```
pq_data <- data_segmented
pq_data$time = as.Date(pq_data$time)
pq_data$year = year(pq_data$time)
pq_data$month = month(pq_data$time)
#we first calculated the average price and total quantity sold for each
#category for each month.
pq_category <- pq_data %>% group_by(Category, year, month) %>%
  summarise(quantity = sum(Qty), price = mean((sum(Net.Sales) / sum(Qty))))
# then we calculated the average price and total quantity sold for each
#category for each month for each cluster.
pq_category_cluster <- pq_data %>% group_by(cluster,Category, year, month) %>%
  summarise(quantity = sum(Qty), price = mean((sum(Net.Sales) / sum(Qty))))
```

We plotted price against quantity sold for different categories to see how price and quantity sold are correlated.

```
ggplot(data = pq_category[pq_category$Category == 'Tea',]) +
  aes(x = price, y = quantity) +
  geom_smooth(color = '#0c4c8a') +
  labs(title = "Price vs Quantity sold for Tea",
       x = "Price",y = "Quantity") + scale_y_continuous(labels = comma)  +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Description

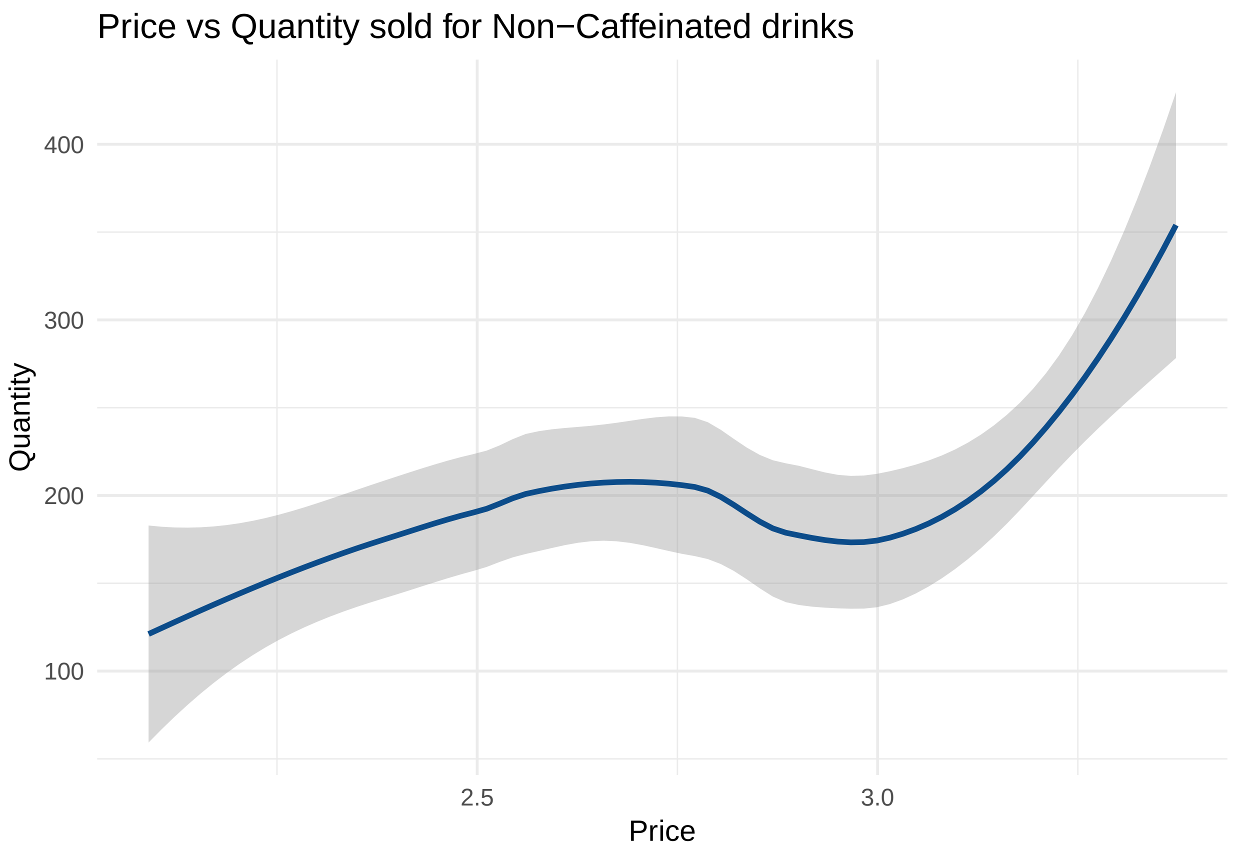
The above graph is price against quantity sold for category ‘Tea’.

Interpretation

we found that for tea, the price and quantity are positively correlated. We saw highernumber of quantity sold for items under category tea having higher price.

```
ggplot(data = pq_category[pq_category$Category == 'Non-Caffeinated Drinks',]) +
  aes(x = price, y = quantity) +
  geom_smooth(color = '#0c4c8a') +
    labs(title = "Price vs Quantity sold for Non-Caffeinated drinks",
       x = "Price",y = "Quantity") + scale_y_continuous(labels = comma) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Description

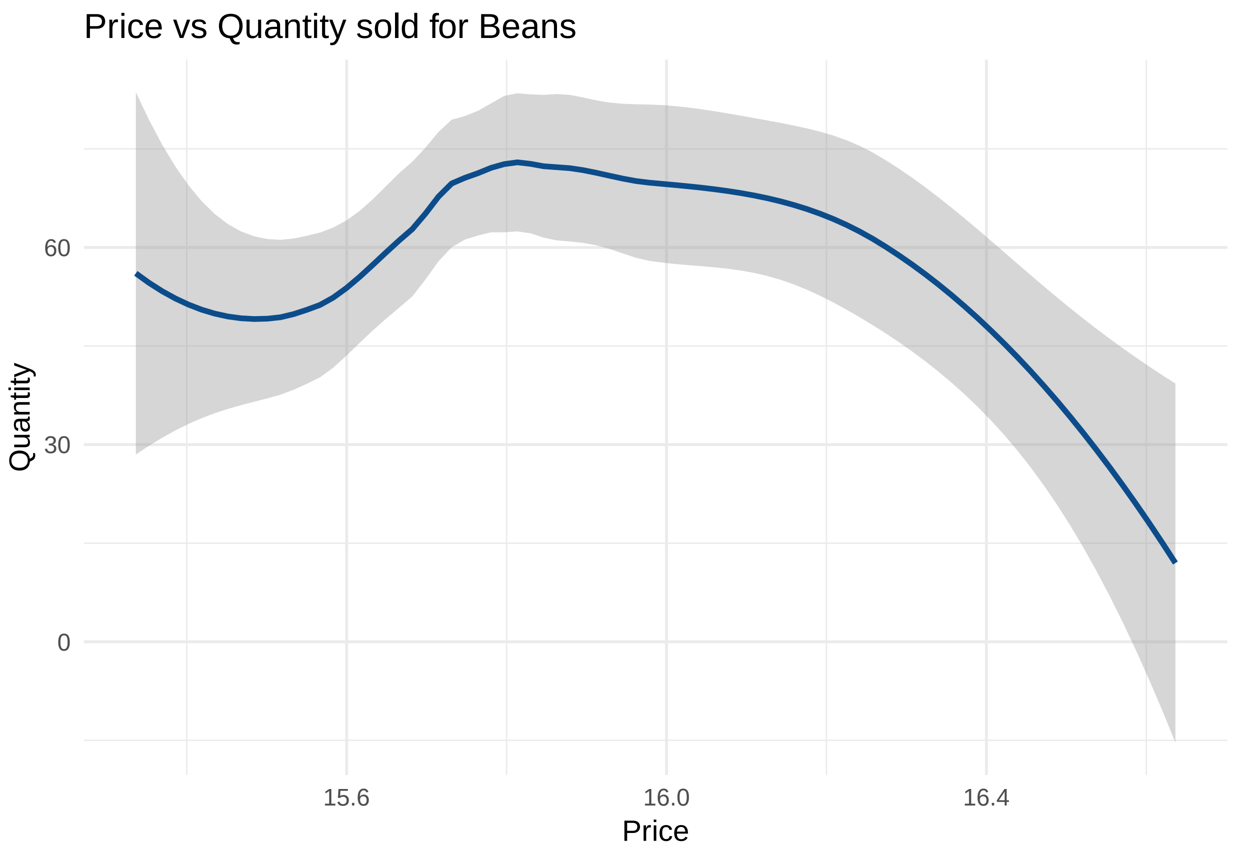
The above graph is price against quantity sold for category ‘Non-Caffeinated drinks’.

Interpretation

we found that for non-caffeinated drinks, price and quantity are positively correlated. We saw highernumber of quantity sold for items under category tea having higher price.

```
ggplot(data = pq_category[pq_category$Category == 'Beans',]) +
  aes(x = price, y = quantity) +
  geom_smooth(color = '#0c4c8a') +
    labs(title = "Price vs Quantity sold for Beans",
      x = "Price",y = "Quantity") + scale_y_continuous(labels = comma) + theme(axis.title.y =
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Description

The above graph is price against quantity sold for category 'Beans'. We observed there is a higher potential to generate more revenue through upselling beans in our earlier analysis.

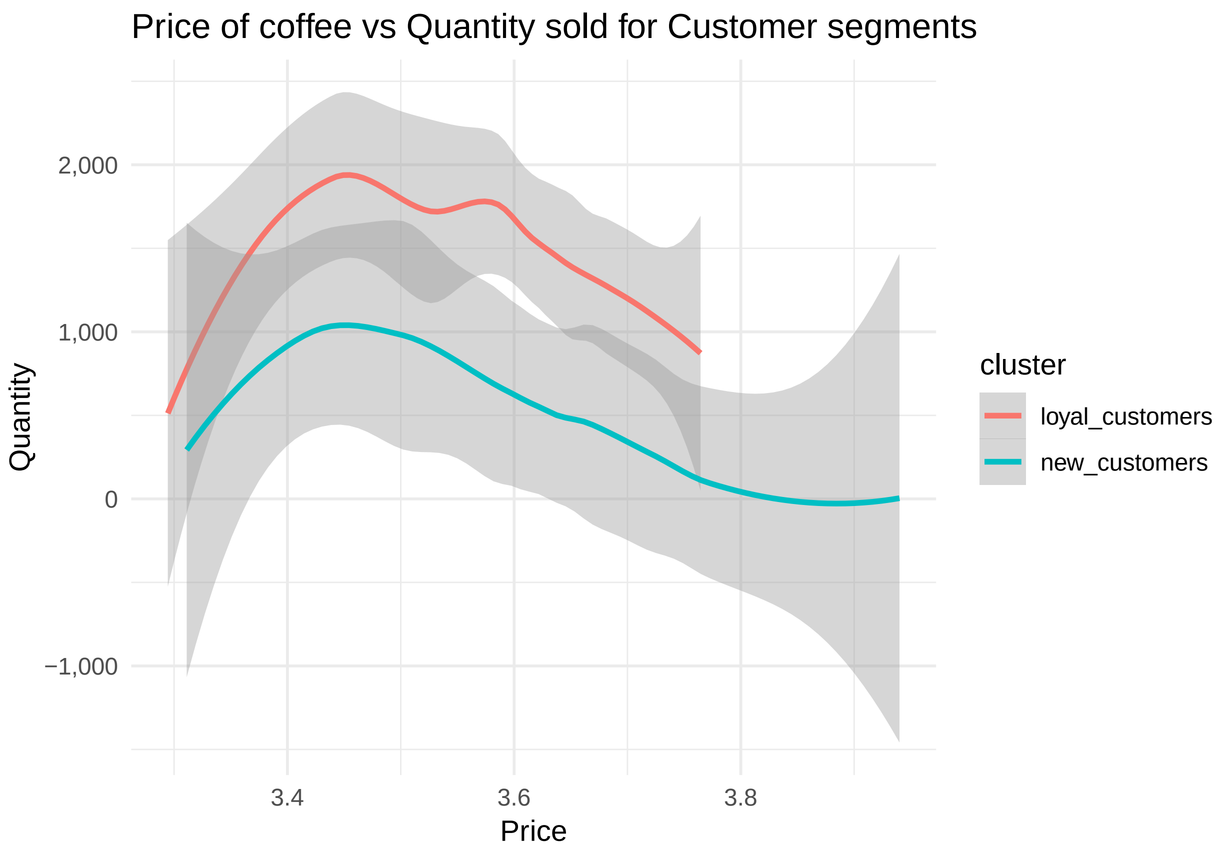
Interpretation

we observe from the trend in the above graph that the quantity sold reduces with price greater than 16.5 dollars per item. We recommend to introduce more beans variant but priced inside 16 dollars.

After analysing the corelation betwen price and categories, we wanted to understand our customer segments behaviour towards price.

```
#filter for loyal customers and new customers.
plot_data <- pq_category_cluster %>% filter(cluster %in%
                                              c('new_customers','loyal_customers'))

ggplot(data = plot_data[plot_data$Category == 'Coffee',]) +
  aes(x = price, y = quantity, color = cluster) +
  geom_smooth() +
  labs(title = "Price of coffee vs Quantity sold for Customer segments",
       x = "Price",y = "Quantity") + scale_y_continuous(labels = comma) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Description

The above graph is price of coffee against quantity sold for our loyal customer and new customer segment.

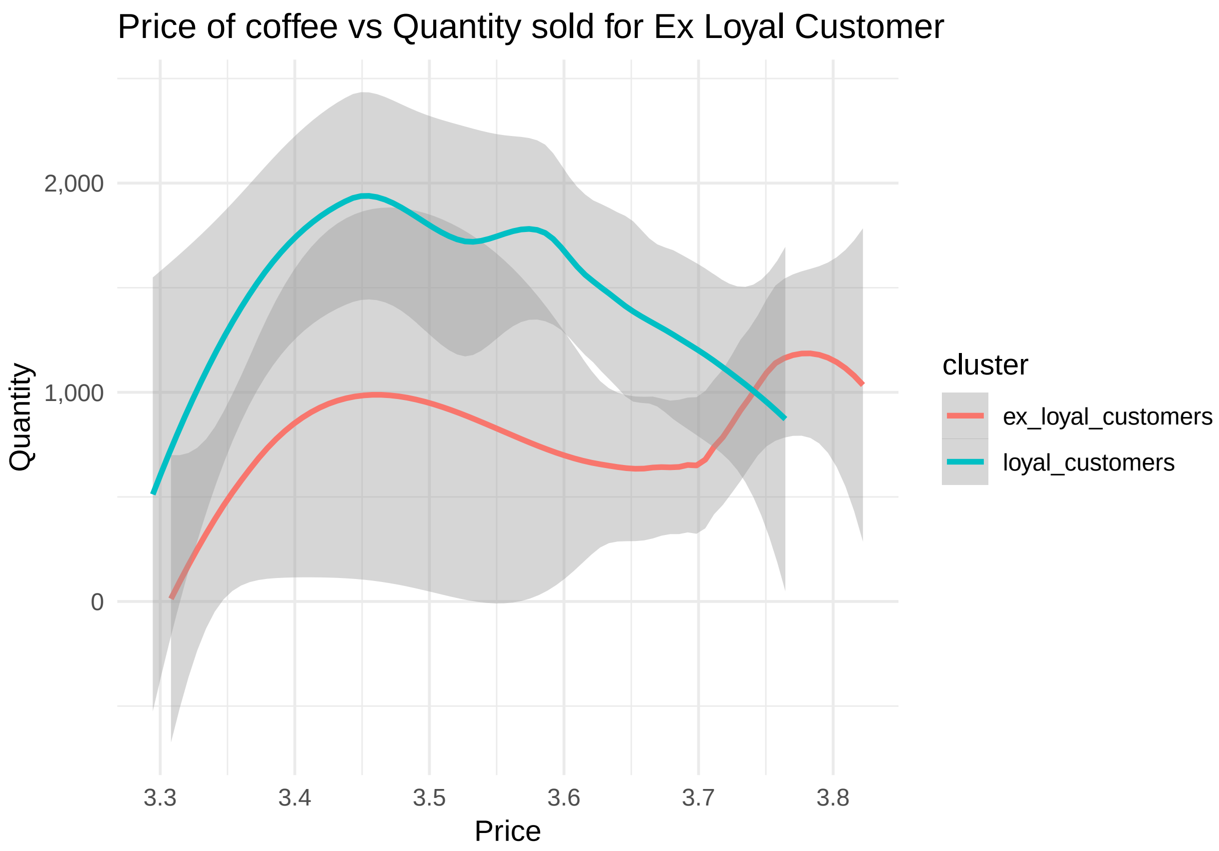
Interpretation

We see that the quantity of coffee sold across different prices is higher for our loyal customer segment.

We saw the difference between our loyal and new customer segments above. Now we wanted to understand what was the reason for losing out customer(ex_loyal_customers) who were our loyal customers previously.

```r
plot_data <- pq_category_cluster %>% filter(cluster %in%
                                              c('ex_loyal_customers','loyal_customers'))

ggplot(data = plot_data[plot_data$Category == 'Coffee',]) +
  aes(x = price, y = quantity, color = cluster) +
  geom_smooth(method = "auto") +
  labs(title = "Price of coffee vs Quantity sold for Ex Loyal Customer",
       x = "Price",y = "Quantity") + scale_y_continuous(labels = comma) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Description

The above graph is price of coffee against quantity sold for our loyal customer and new customer segment.

Interpretation

We can conclude that price was not the reason for losing out on these customers, as the sales was high even for higher priced coffee.

Assumptions:

We assume there are no loyal customers in the new_customer segment and all our loyal customers are segmented under loyal customers.

Conclusion

From this analysis we see that our loyal customer base is not very price sensitive. Where as our new customre segment is price sensitive. To get a sweet pricing to help increase sales from both these segments we can price our coffee between 2 to 5 dollars. We can introduce different flavoured coffee that would cater for our loyal customers who are not price sensitive.

We observed that beans have a high margin compared to other categories. from this analysis we see that the sales of beans decreases for varients priced above 16 dollars. We recommend to introduce new varieties of beans that cost around 16 dollars for more profits.

As we are unable to address or identify issues which may be turning Loyal Customers into Ex Loyal

Customers we may not be in a good position to solely focus on the Loyal Customers. This is why we recommend focusing Central Perk's attention on building an additional group of loyal customers and maintaining their current loyal customer base to normalize demand and generate additional revenue. Our first concern is that loyal customers may be leaving for one reason or another. We explored the thought that they are leaving for cheaper coffee but we noticed that our ex loyal customers were very inelastic towards pricing of the coffee, because of this distinction we do not think price plays a role in these customers leaving.

Part 4 - Time Series Analysis

In this part of analysis we want to determine if the belief of Central perk regarding their business being consistant year over yearis true or false. To determine this we used a time series analysis to decompose the series to see the trend and inturn tried to find out in there are any outliers that exist in the dataset.

We also want to analyse the sales distribution of categories such as beans, which have a greater margin to increase revenue.

We are also curious to see how the net revenue is distributed across different time of day for the two customer segments (new customers and loyal customers) we are targetting. We would like to come up with a recommendation to increase sales and also smoothen the distribution of sales acrossthe day.

Analysis for Business consistancy

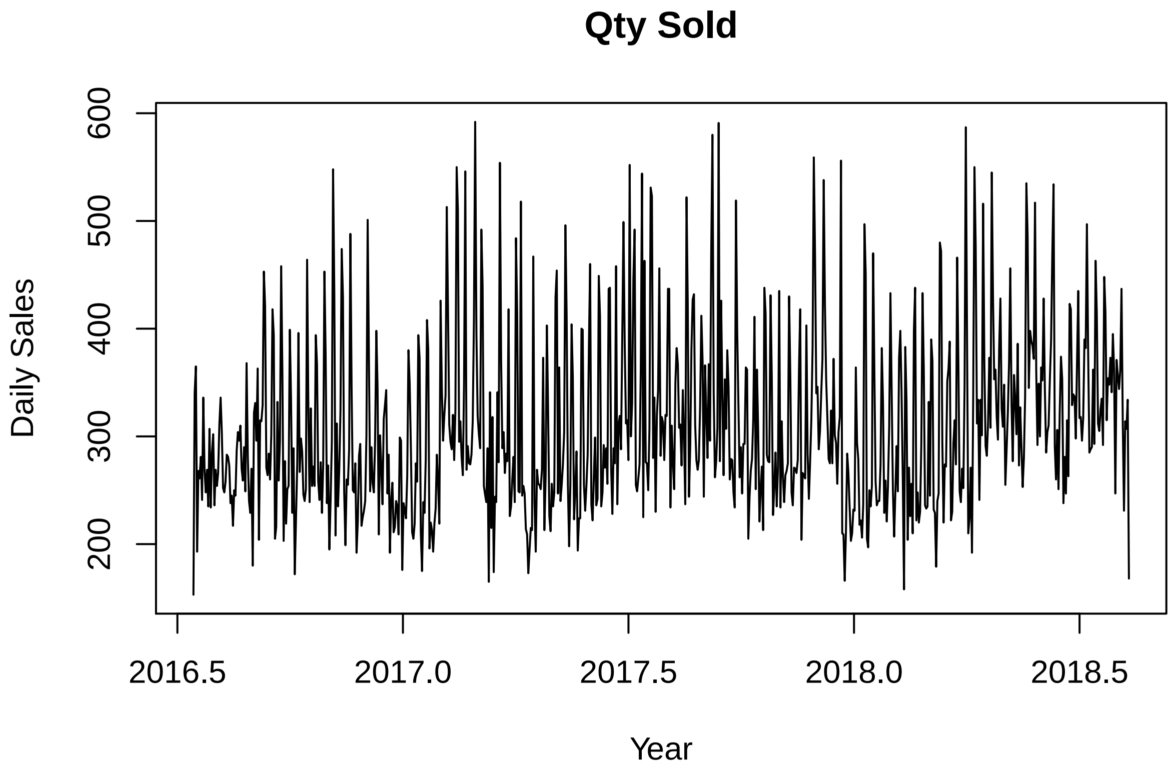
To create a time series object we aggregate the data at a day level.

```
df_monthly = data %>% select ( Date, Qty) %>%group_by(Date) %>%
  summarise( sum_qty = sum(Qty)) %>% arrange(Date)
```

```
## Warning: `new_overscope()` is deprecated as of rlang 0.2.0.
## Please use `new_data_mask()` instead.
## This warning is displayed once per session.
```

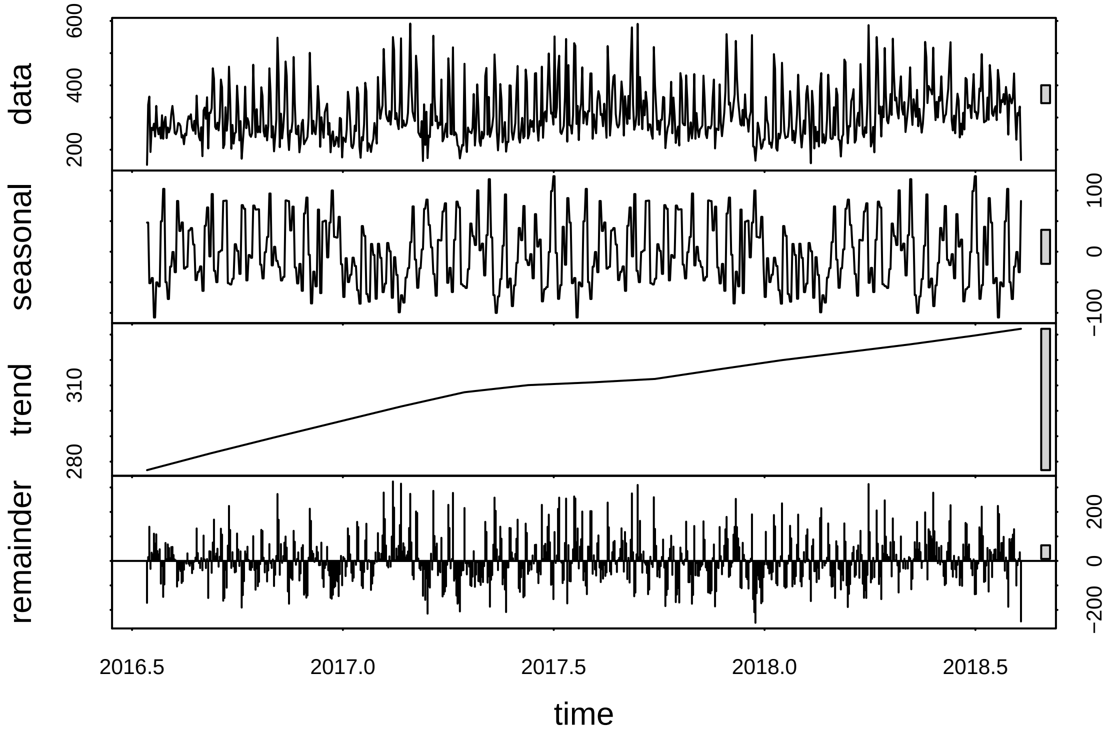
```
## Warning: `overscope_eval_next()` is deprecated as of rlang 0.2.0.
## Please use `eval_tidy()` with a data mask instead.
## This warning is displayed once per session.
```

```
msts = msts(df_monthly$sum_qty,seasonal.periods = c(7,30.4,365.25/7,365.25),
            start=decimal_date((as.Date("2016-07-15"))))
plot(msts, main="Qty Sold", xlab="Year", ylab="Daily Sales")
```



Above is the plot of daily data across the two years. We see a lot of movement but it is hard to understand if there are any seasonal elements in play based on this graph. Going forward we will try to decompose the time series to understand the seaonsal components.

```
data_fit = ts( df_monthly$sum_qty , frequency=365.25,
               start=decimal_date((as.Date("2016-07-15"))))
fit = stl(data_fit, s.window = 'periodic')
plot(fit)
```



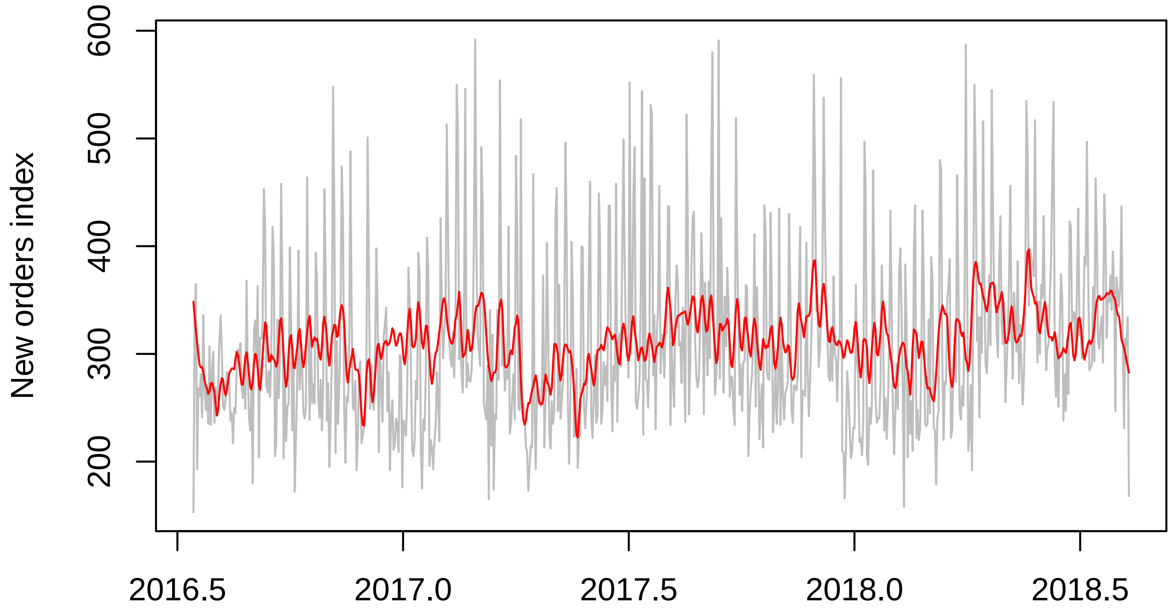
Description

The above graph shows the decompostion of the time series data we created

Interpretation

We observe that volume of sales has an increasing trend.Considering that it is daily data decomposed it is hard to understand the seasonal patterns but we can see that there is a very clear rising trend in the data which indicates that our stores sales volumne has been increasing over time.

```
fit <- stl(data_fit, t.window=9, s.window="periodic", robust=TRUE)
plot(data_fit, col="gray", main="", ylab="New orders index", xlab="")
lines(fit$time.series[,2],col="red",ylab="Trend")
```



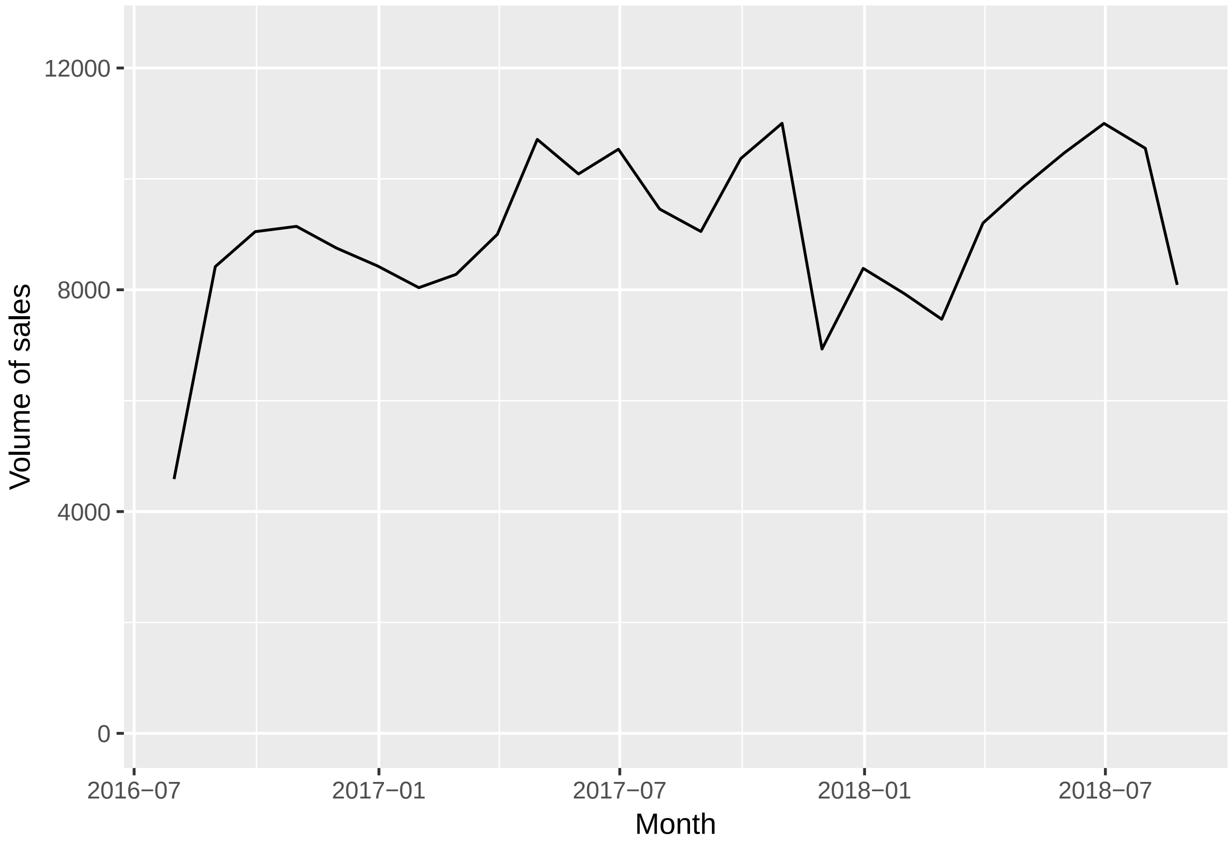
Description

The plot shows the trend component in red and the original data in grey. The trend shows the overall movement in the series, ignoring the seasonality and noise components.

Interpretation

This graph furthur supports the conclusion that there is a very clear rising trend in the data which indicates that our stores sales volumne has been increasing over time. This increase is not very drastic.

```r
library(zoo)
y=read.zoo(df_monthly, format = "%m/%d/%y")
library(xts)
y1=to.monthly(y)

ts_m = apply.monthly(y, FUN= sum)

autoplot(ts_m, geom = "line", ylim=c(0,12500)) + xlab("Month")+ylab("Volume of sales")
```



Description

This is the monthly plot of volume of sales distribution our data,

Interpretation

we again see a slight increasing trend in the data but what we see most interestingly from this part is that there seems to be seasonality in the volume of sales. Just based on this graph it seems like the sales are lower in the winter months in the begeninig of the year and then it rising during the middle of the year.

Conclusion

We see an increasing trend in the volume of sales year over year. Hence we agree with the coffee shop regarding the business being consistant year over year. Even though we see a trend in sales, it is minimum. we can say its more or less consistent.

Conclusion

Customer Loyalty

While we did identify a subset of customers who we would define as loyal, we had some concerns regarding relying on them for the bulk of the shop's revenue going forward. It is recommended that a shop's loyal customer base should comprise of 20% of their customers and 50% of their revenue (Loyalty Definition) and our loyal customers only make up 6% of customers with membership and they only accounted for 38% of revenue in 2018. Additionally, cluster 4 is defined as Ex Loyal

Customers, customers who we believe used to regularly buy coffee from Central Perk. If we are unable to address or identify issues which may be turning Loyal Customers into Ex Loyal Customers and with percent revenue behind where we would like we may not be in a good position to solely focus on the Loyal Customers. This is why we recommend focusing Central Perk's attention on building an additional group of loyal customers and maintaining their current loyal customer base to normalize demand and generate additional revenue.

Overall Business

The over all business of the coffee shop has an increasing trend over the past two years. The business seems to be more or less consistent year over year.

Recommendations

Short term recommendations

We recommend the following Short term recommendations leveraging the insights from our analysis:

- Prepaid coffee card to help normalize demand during the morning rush and maintain loyal customers
- New Offerings such as sandwiches and salads to reach new potential loyal customers
- Discounting a 2nd coffee to promote additional purchases from our loyal members
- Removing beer and cereal because of their lack of revenue generation
- Introducing a new bean of the month to excite loyal customers and spark purchase interest

Long term recommendations

We recommend the following long term recommendations leveraging the insights from our analysis:

- Exploring the possibility of a Central Perk themed RTD which can further increase sales to our loyal customers
- Surveying our loyal customer base to proactively identify trends and make changes to drink offering to meet the needs of customers