IST687

DATA ANALYSIS ON HYATT HOTELS IN UNITED STATES

GROUP 1
Anchal Bansal

Ashmin Swain

Radhika Arora

Sanchit Mehra

Suchitra Deekshitula

Varsha Fotedar

CONTENTS

TABLE OF FIGURES

1. Introduction

Hyatt is a hotel chain, founded in 1957. Having spread all over the world with different brands and having different Net Promoter Score (NPS) ratings, the objective of the project is to provide recommendations to Hyatt Hotel Management to increase their NPS* score. The NPS is based on the customer satisfaction, which is recorded, based on the survey data of twelve months, which was provided to the project group. This data includes the feedbacks of the customers who made bookings with Hyatt hotel brands present in the entire world and also the features of the hotels they made their bookings in.

NPS gives us an idea of the following:

- Detractors: Customers who give a score of 0 to 6. They are dissatisfied with the hotel. These have a higher tendency of discouraging friends or colleagues from making future hotel bookings or using the facilities.
- Passives: Customers who gave a score of either 7 or 8. These are not likely to recommend hotel to others.
- Promoters: Customers who gave a score of 9 or 10. These are likely to actively recommend hotel to others.

Hyatt has following brands: Hyatt Regency, Grand Hyatt, Hyatt, Park Hyatt, Hyatt Zilara, Hyatt Place, Hyatt Ziva, Andaz, Hyatt House, Hyatt Centric.

2. Scope

Due to our computer's limitation on processing large data sets, we decided to narrow down our focus to one region with maximum population in United States and we decided to go with California as larger data generally gives better insights. During our research, we saw that Hyatt Regency has many promoters as well as detractors than other brands in California. The scope was then defined to increasing the promoters and decreasing the detractors for the Hyatt Regency to increase the brand's NPS.

3. Analysis

Through analysis of data, we analyzed the factors, which were affecting the customer ratings. Recommendations were made on how Hyatt Regency hotel in California can improve its guests' experiences.

As the data given was overwhelming with lot of monthly data, we first aggregated the data of all the months in one file and then cleaned the file. After this, we segregated the data based on the business and leisure users and then looked for the detractors with respect to other factors in each scenario.

**NPS - NPS score or net promoter score is the measure of willingness of customer to recommend a company's services to others. It also represents the customer's loyalty to the brand.*

3.1. DATA ACQUISITION

The data we got was in 12 files based on the reservations made in different months of a year. Since a particular month does not serve as a best means to analyze data (seasonal effect) we decided to analyze data of all the 12 months. To narrow down our dataset we decided to pursue hotels only in United States. Thus, we started with all 12 csv files, imported them and made 12 RDS files with just hotels in United States location and where the reservation made was not cancelled.

```
july <- read.csv("out-201407.csv")
july_us <- july[june$Country_PL=="United States",]
july_us <- july_us[july_us$RESERVATION_STATUS_R!="CANCEL",]
july_us$Month = "july"
saveRDS(july_us,"july_us.rds")
```

Similarly, we acquired files for the remaining 11 months. We then merged all 12 RDS files using rbind as finaldf.

3.2. DATA FOCUSING

The data was still too large and thus we decided to focus on one particular state. We used ggmaps to find the US state with the highest number of reservations which came out to be California. We continued the rest of our analysis on the data of this single state only.

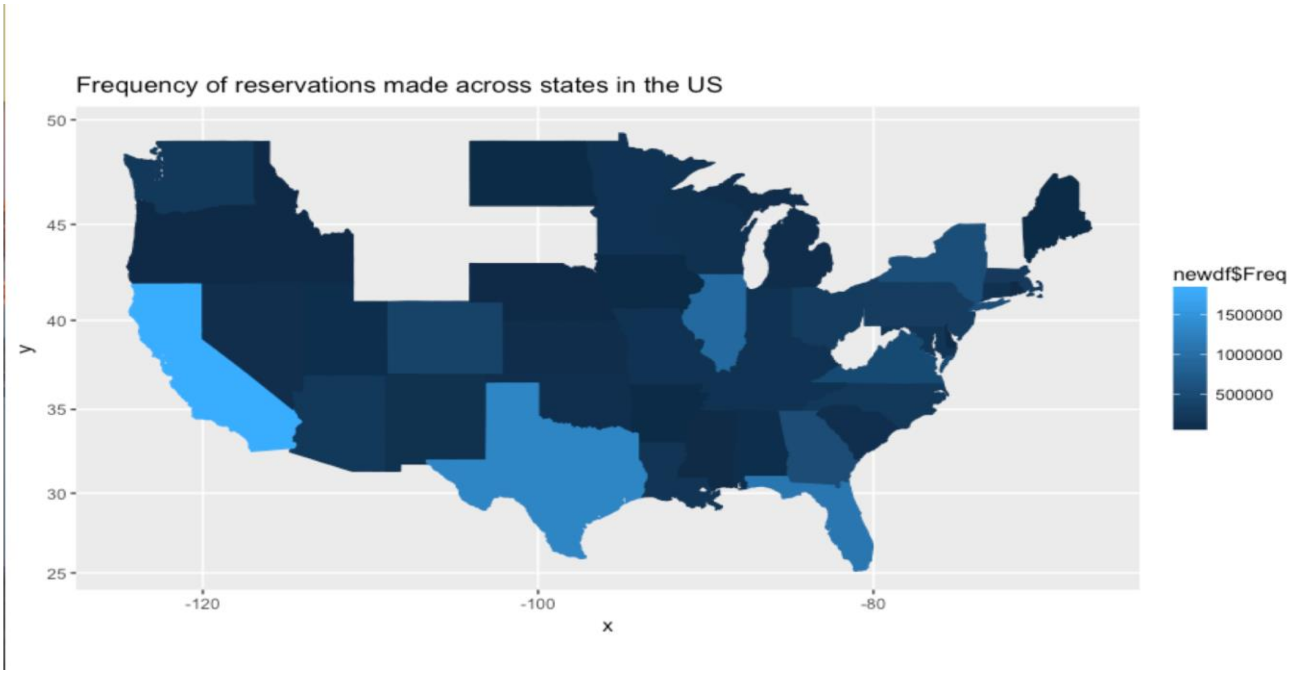

Fig. 1: Map representing highest number of reservations

We then created a subset of finaldf with records of only California state.

```
finaldf_cali = finaldf[finaldf$State_PL=="California",]
```

3.3. FORMULATION OF BUSINESS QUESTIONS

We performed descriptive statistics to see the behavior of the factors affecting the Net Promoter Score.

The next step was to finalize the business questions based on these factors and to prove the hypothesis that changes in these behaviors will have a good impact in increasing the NPS for Hyatt Hotels.

We have formulated the following business questions:

1. **Which brand of Hyatt can be referred as a benchmark to improve the NPS score of other brands of Hyatt chain?**

With a smaller subset, it was now easy for us to analyze trends. We started with analyzing the trends in our dataset based on the different brands of Hyatt. With a focus on NPS we analyzed the number of promoters and detractors in every brand type category of the Hyatt Hotels. We removed all the passives by converting all the NA's to "passive" and then eliminated all those records. The data frame thus formed contains only promoters and detractors.

```
#code for plot
barplot(m_p_normalize, col = colors, title("Number of promoters"))
```

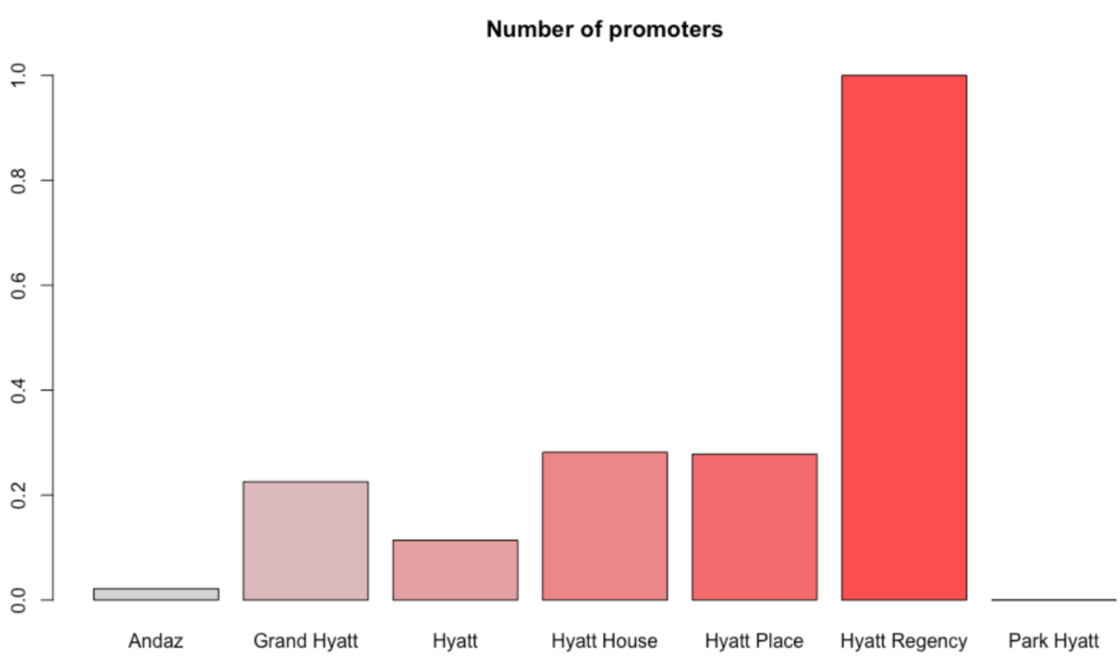

Fig. 2: Number of promoters based on brand Type

```
barplot(m_d_normalize, col = colors, title("Number of detractors"))
```

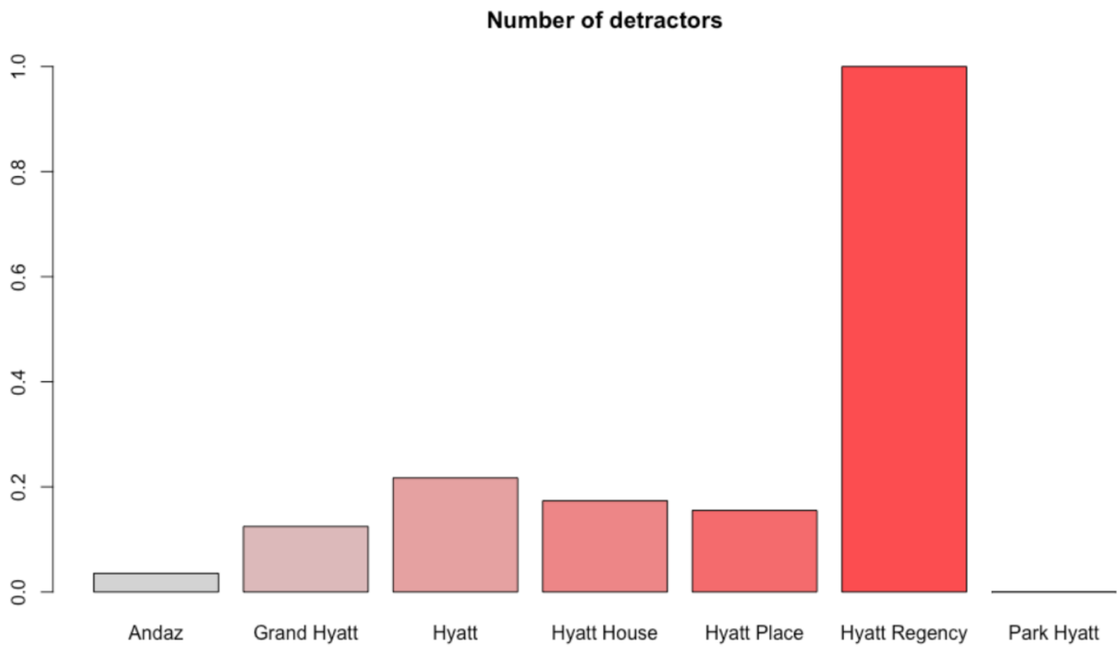

Fig. 3: Number of detractors based on brand Type

From both the graphs it is clearly evident that the number of promoters as well as detractors are highest in Hyatt Regency and thus should be considered as a benchmark to improve their facilities by other brands of Hyatt.

Recommendation: All the brands of Hyatt can refer to the facilities that Hyatt regency provides and which makes it the most preferred brand and can work on improving or providing those facilities.

2. **What facilities can a specific brand of Hyatt brand of Hyatt hotel at a given Zip Code improve to be at par with other Hyatt hotels of same brand.**

Hyatt Hotels have 9 brand of hotels namely Andaz, Grand Hyatt, Hyatt, Hyatt House, Hyatt Place, Hyatt Regency, Hyatt Zilara, Park Hyatt and Hyatt Ziva according to the dataset. Each hotel can compare itself with other Hyatt hotels of similar brand to improve its features and gain better ratings from future customers based on the survey data of previous customers of other hotels.
Based on such an ideology, a function was written which takes the survey data, the zipcode of a hotel along with the desired brand of hotel. The function returns back the average feature values of various scores given by customers for the hotel and the average feature values of other similar

hotels present in the data. The function also tells the user what features are absent in the hotel and are responsible for good ratings in other hotels. These features can be looked upon by the hotel management and improvements in these features would lead to increase in the net NPS score for the hotel.

The following features of the survey data is used for comparison of the selected hotel with rest of the hotels.

- Overall Satisfaction
- Tranquility
- Customer Service
- Internet Satisfaction
- F&B Experience Quality
- Boutique Presence
- Casino Presence
- Convention Presence
- Elevator Presence
- Fitness Trainer Presence
- Indoor Corridors Presence
- Limo Service Presence
- Indoor Pool Presence
- Regency Grand Club Presence
- Restaurant Presence
- Shuttle Service Presence
- Spa Presence
- Spa Online Booking Presence
- Valet Parking Presence
- Guest Room Satisfaction
- Hotel Condition
- Staff Caring
- Check In Quality
- Bell Staff Presence
- Business Center Presence
- Conference Presence
- Dry Cleaning Presence
- Fitness Center Presence
- Golf Space Presence
- Laundry Presence
- Mini Bar Presence
- Outdoor Presence
- Resort Presence
- Self-Parking Presence
- Ski Presence
- Spa in Fitness Center Presence
- Spa F&B Offering

For e.g. a random Hyatt Regency hotel was selected in Los Angeles, CA with zip code 90071. The hotel has a net NPS score of 7.92, the Check-In quality rated by the customers is 9.05, the internet satisfaction as scored by customers is 7.39, etc as shown in the result below. The selected hotel lacks various features like Dry Cleaning, Outdoor Pool, Valet Parking, Fitness Center, etc.

Other Hyatt Regency hotels in the rest of California have a net NPS score of 8.36, the Check In quality is almost similar to the selected LA hotel but the Internet Satisfaction is 8.02. These hotels on an average also have Dry Cleaning, Outdoor Pool, Valet Parking, Fitness Center, etc.

Based on these finding, the hotel management of Hyatt Regency, Los Angeles can work on to increase the internet satisfaction, can plan for introducing Dry Cleaning, Outdoor Pool, Valet Parking etc which in turn would attract customers, increase their net NPS Score and increase their revenue.

The function is the tool which is deliverable to the client. The client can enter the survey data along with the zip code and hotel brand. Then an output like the one shown below would be generated based on computations and in this way the client can determine which areas of the hotel need to be fixed in order to improve the NPS.

Sample Output:

label	currentValue	meanValue
Likelihood to Recommend	7.92611464968153	8.36222547021039
Overall Satisfaction	7.98243412797992	8.31412837067983
Guest Room Satisfaction	8.45108005082592	8.34404515088689
Tranquility	8.48898678414097	8.4144852526459
Hotel Condition	8.11450381679389	8.55641232209307
Customer Service Quality	8.53767560664112	8.83317563000869
Staff Caring	8.48689956331878	8.76243548015912
Internet Satisfaction	7.390625	8.02718988098202
Check In Quality	9.05032822757112	9.08493942279279
F&B Experience Quality	8.04166666787424	8.53094511027277
Bell Staff Presence	N	Y
Business Center Presence	N	Y
Dry Cleaning Presence	N	Y
Elevators Presence	N	Y
Fitness Center Presence	N	Y
Laundry Presence	N	Y
Outdoor Pool Presence	N	Y
Regency Grand Club Presence	N	Y
Self Parking Presence	N	Y
Spa Online Booking Presence	N	Y
Spa F&B Offering Presence	N	Y
Valet Parking Presence	N	Y

For further analysis, we divided the data of California based on the brand type Hyatt Regency into 2 categories namely Business and Leisure describing the purpose of visit. The following 3D pie chart explains the number of users according to their purpose of visit:

Fig. 4: Pie Chart on Purpose of Visit

This gives an insight about how majority of the customers in Hyatt Regency check in for Business related activities. This also tells about features of the hotels that account for business visits like Conference hall, Convention hall, etc which should be improved and maintained at a good standard by the hotel management to attract more business class customers in their hotels.

3. Which parameter/column (Business Specific) affects the NPS most?

After understanding the previous analysis, following are the columns, which we think, are important for the analysis:

Business Specific: convention center, conference room, total meeting place, business center, dry cleaning, fitness center

Following is the description of the selected features according to the survey dataset:

Business Center_PL	Flag indicating if the hotel has a business center
Conference_PL	Flag indicating if the hotel has a conference center nearby

Convention_PL	Flag indicating if the hotel has convention space
Dry.Cleaning_PL	Flag indicating if the hotel has dry-cleaning
Fitness.Center_PL	Flag indicating if the hotel has a fitness center

*** In the following plots, Y represents the presence of mentioned facility in the hotels and N represents hotels that lack the facility.**

- **Business Center**

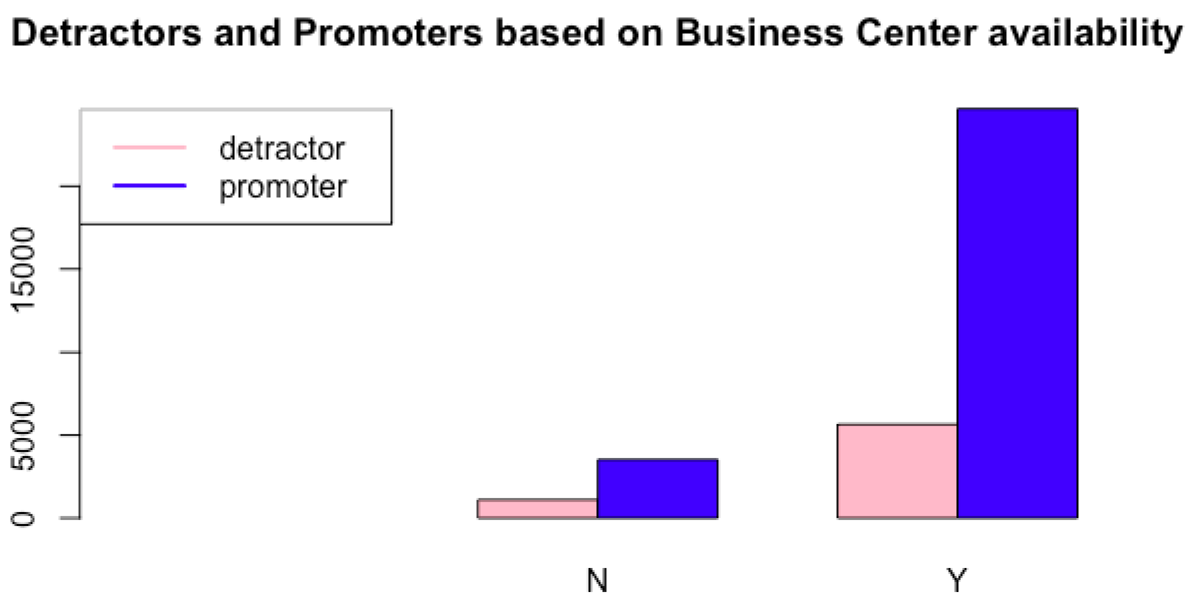

Fig. 5: Bar graph on detractor and promoters versus Business Center availability

- **Convention space**

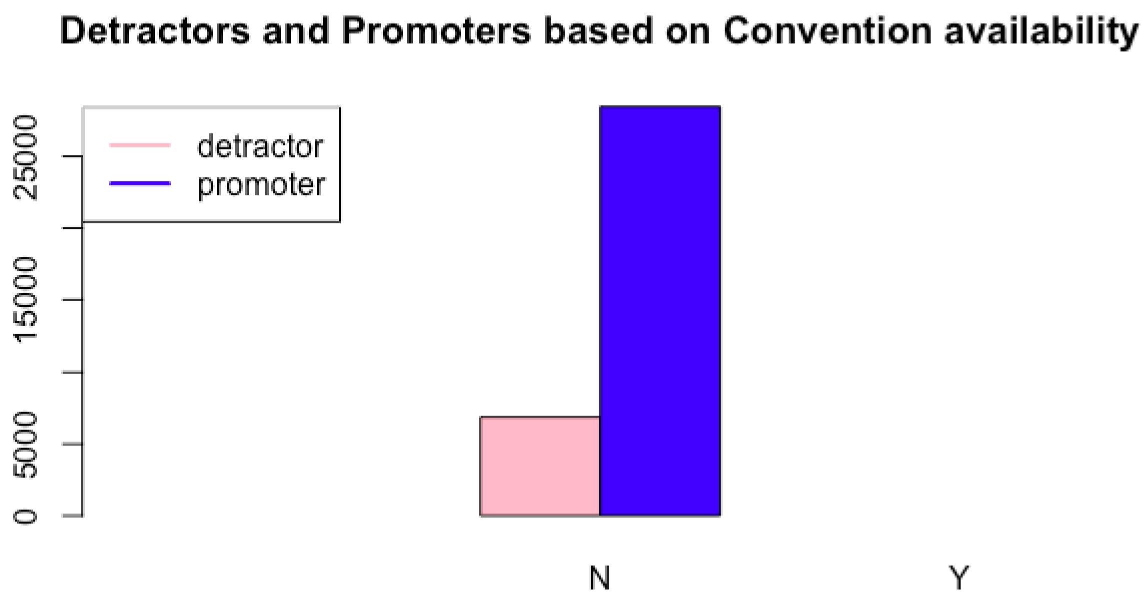

Fig. 6: Bar graph on detractor and promoters versus convention space availability

- **Dry Cleaning**

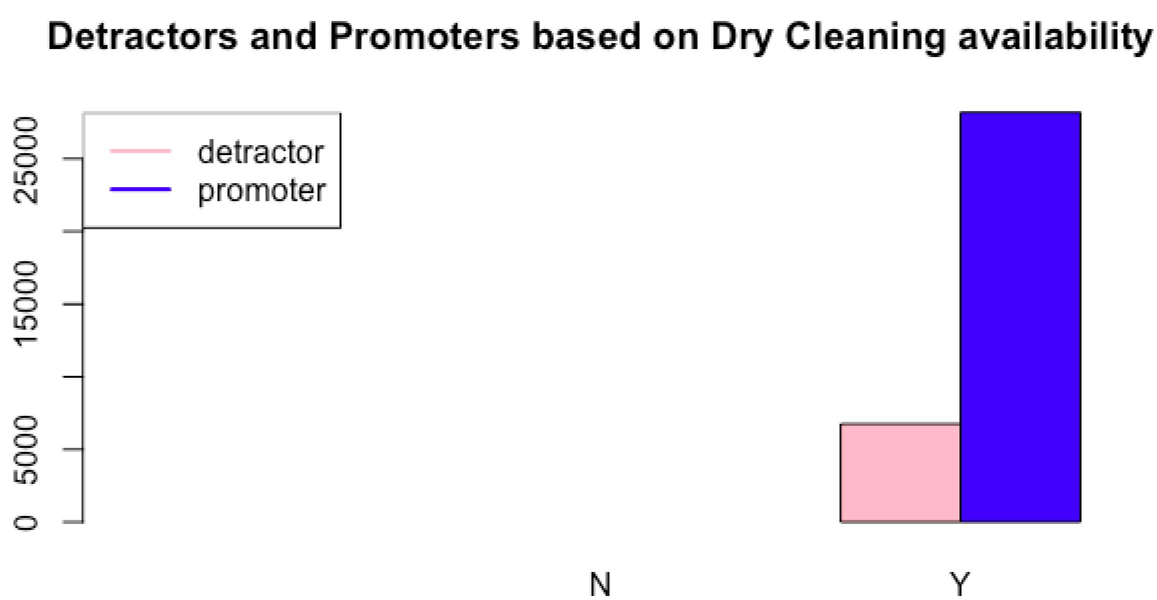

Fig. 7: Bar graph on detractor and promoters versus dry cleaning availability

- **Conference Center**

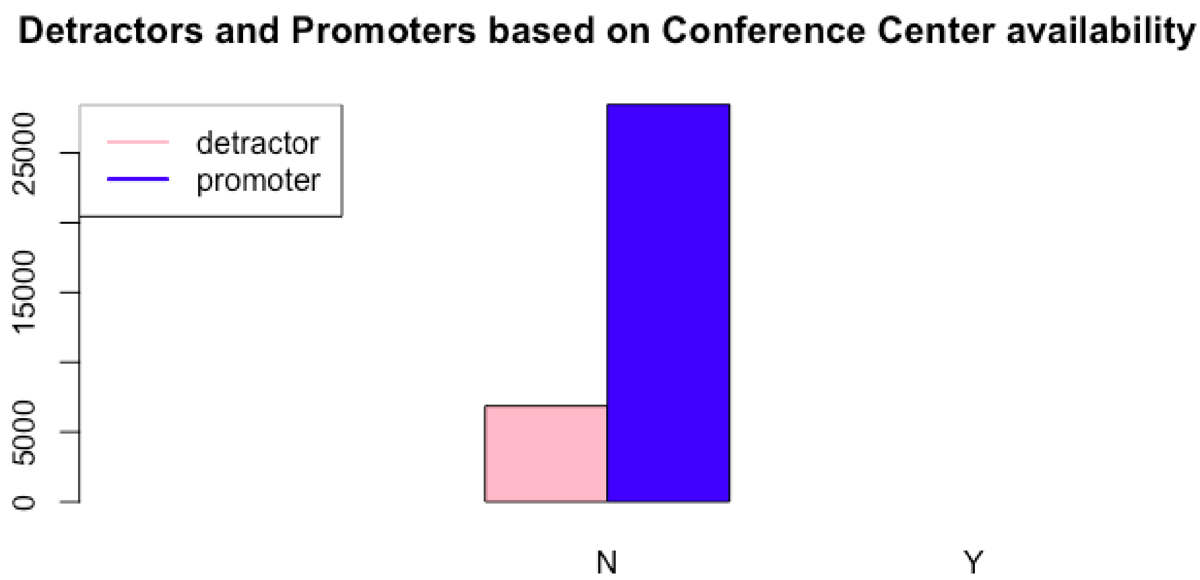

Fig. 8: Bar graph on detractor and promoters versus conference center availability

- **Fitness Center**

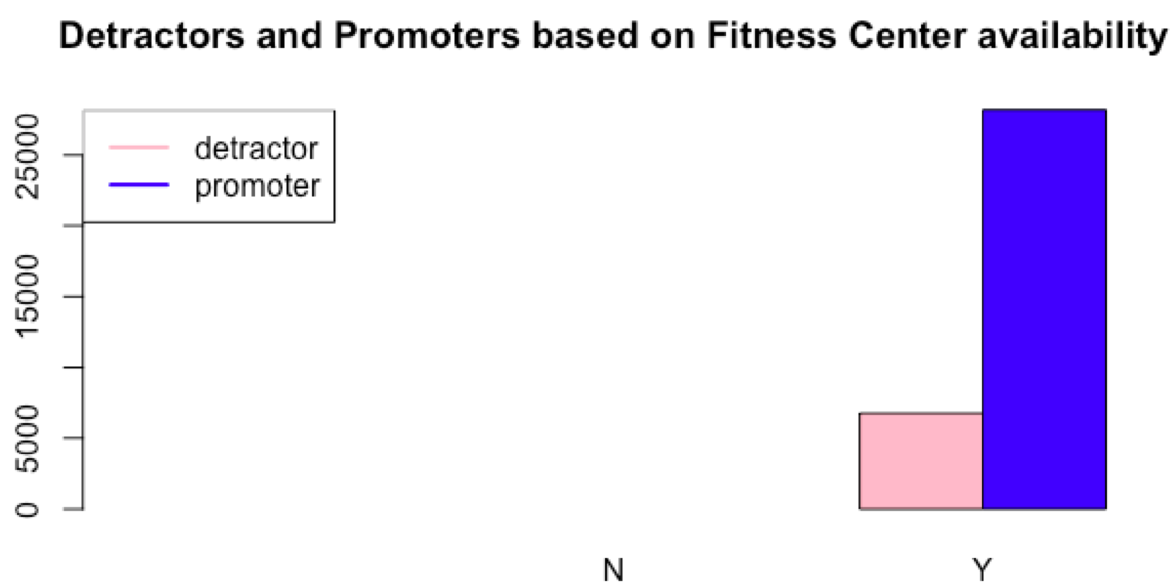

Fig. 9: Bar graph on detractor and promoters versus fitness center availability

Recommendation: People who travel and check in Hyatt on business purpose look forward to having facilities like business centers and convention centers. To have business meetings dedicated centers like these are always preferred and if such services are not provided to the guests they might rate a hotel low and may become the detractors. Thus, every hotel should provide these business services to attract more business users and increase the number of promoters resulting in the overall increase of NPS and revenue.

4. Which parameter/column (Leisure Specific) affects the NPS most?

Leisure Specific: casino_pl, spa, boutique_service, golf, fitness center, outdoor pool, indoor pool

Casino_PL	Flag indicating if the hotel has a casino
Spa_PL	Flag indicating if the hotel has a spa
Boutique_PL	Flag indicating if the hotel is boutique
Golf_PL	Flag indicating if the hotel is near a golf space
Pool.Indoor_PL	Flag indicating if the hotel has an indoor pool
Pool.Outdoor_PL	Flag indicating if the hotel has an outdoor pool

We decided on the factors affecting the purpose of visit of users based on the number of promoters and detractors present in the Regency brand Hyatt hotel with or without the facilities.

- **Spa**

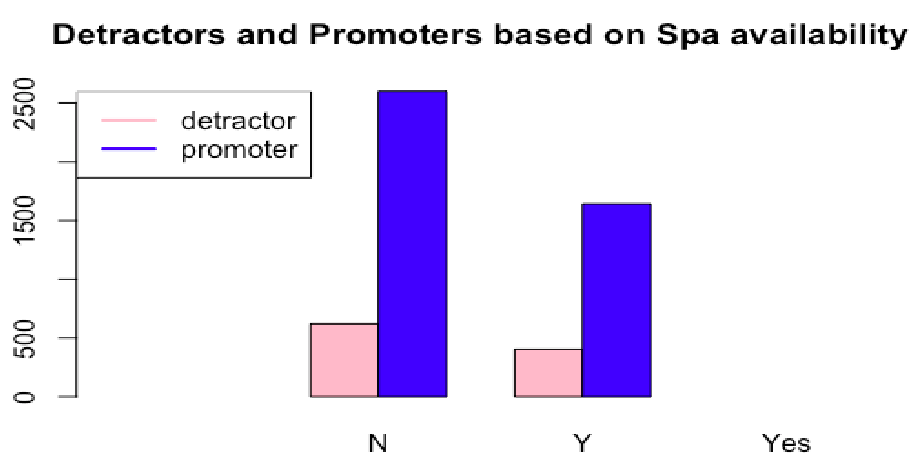

Fig. 10: Bar graph on detractor and promoters versus spa availability

- **Indoor Pool**

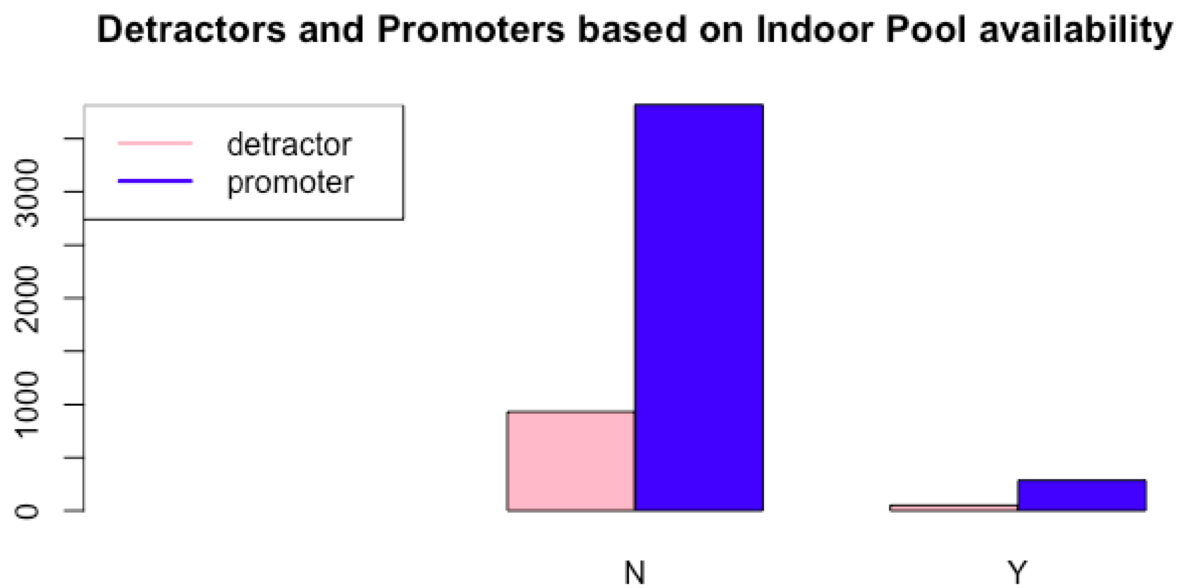

Fig. 11: Bar graph on detractor and promoters versus Indoor Pool facility

- **Outdoor Pool:**

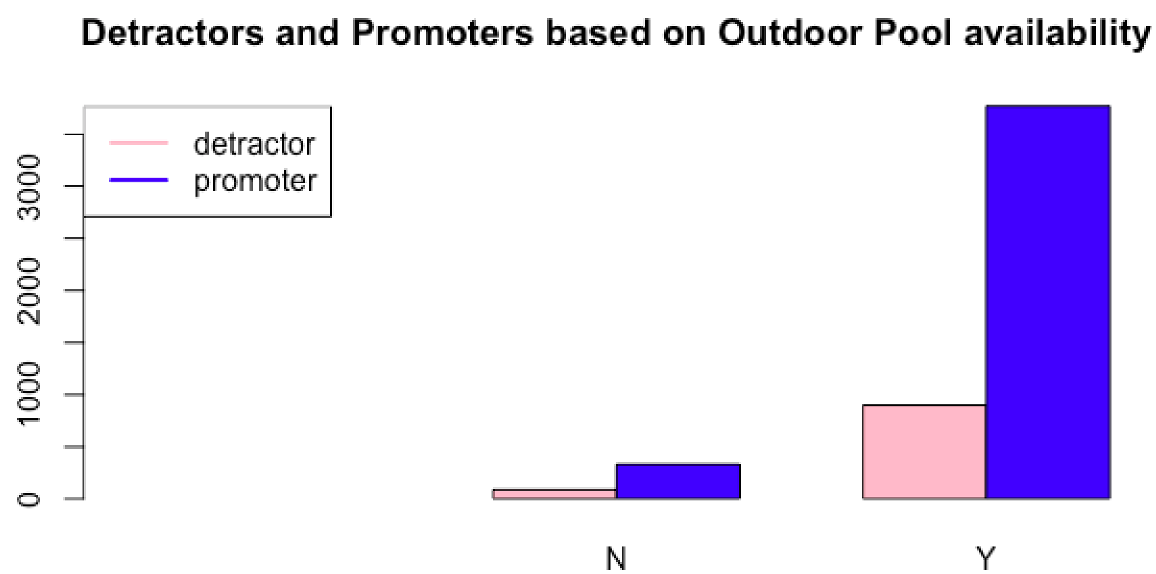

Fig. 12: Bar graph on detractor and promoters versus Outdoor Pool availability

- **Casino**

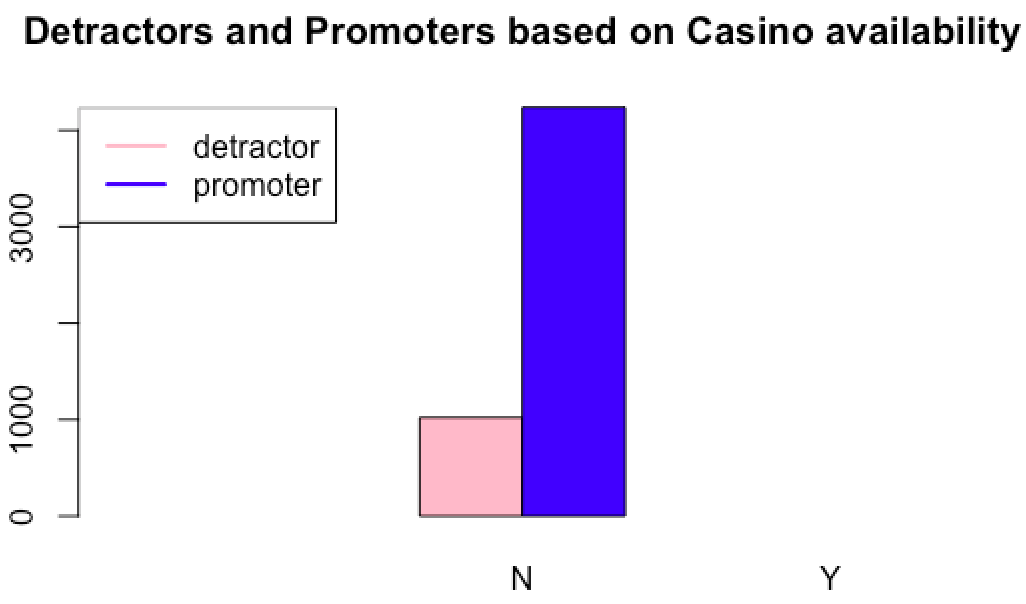

Fig. 13: Bar graph on detractor and promoters versus Casino availability

- **Boutique**

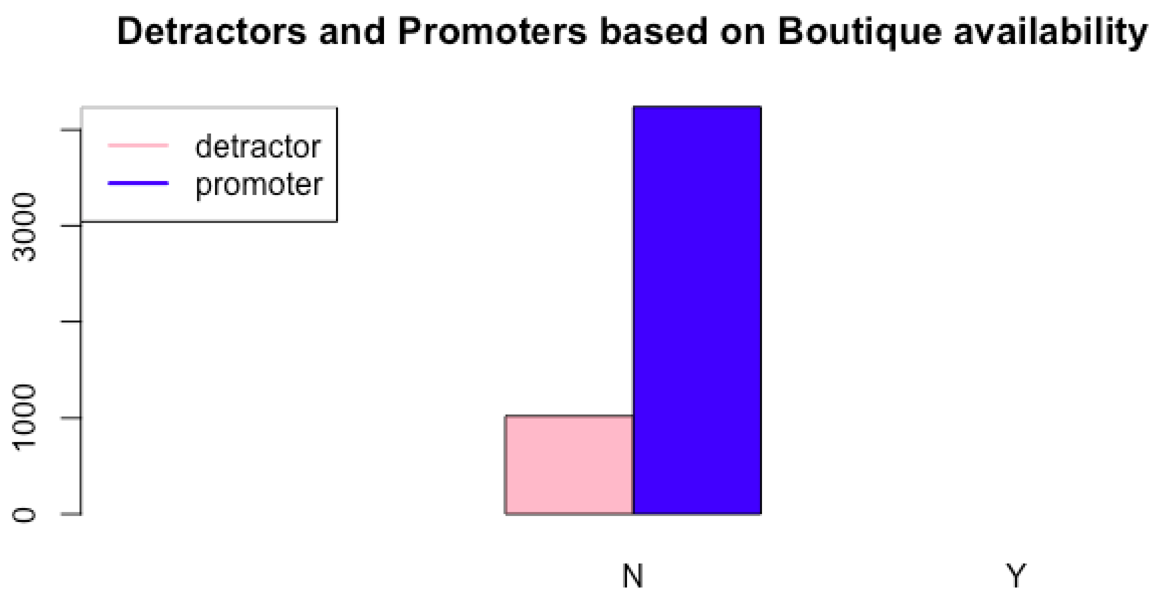

Fig. 14: Bar graph on detractor and promoters versus Boutique availability

- **Golf**

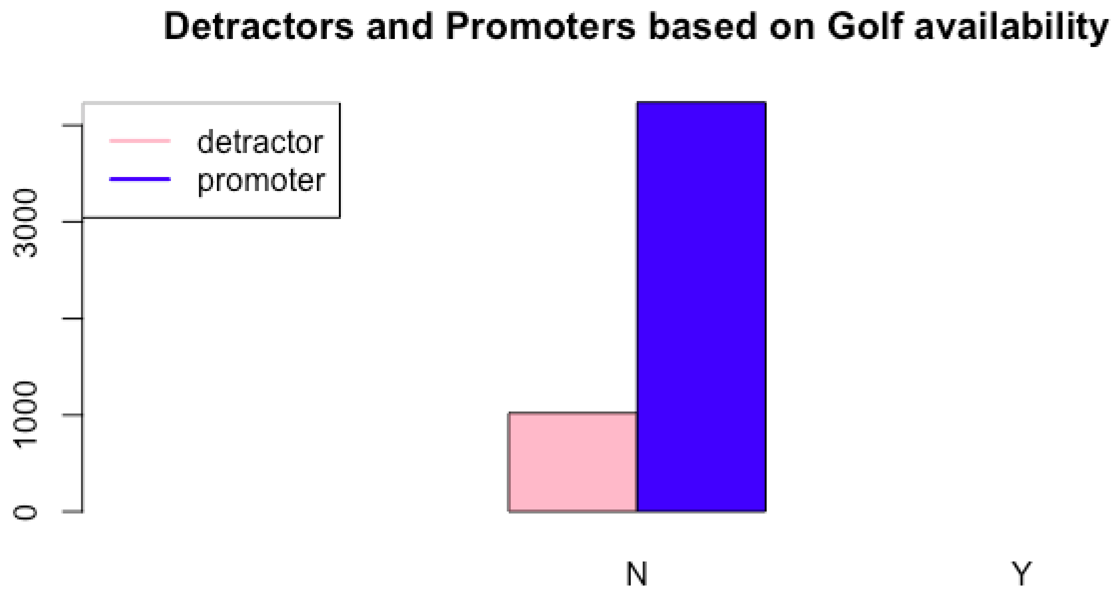

Fig. 15: Bar graph on detractor and promoters versus Golf Course availability

Recommendation: People who travel and check in Hyatt on leisure purpose look forward to having facilities like Spa, Indoor Pool, Outdoor Pool, etc to have a good experience in the hotel. To provide customers on vacations with leisure facilities like the one mentioned are always preferred and if such services are not provided to the guests they might rate a hotel low and may become the detractors. Thus, every hotel should provide the leisure services to attract more users on vacations and increase the number of promoters resulting in the overall increase of NPS and revenue.

5. **How does the monthly trend affect the NPS score of Hyatt hotels?**

We had a look at monthly trends of reservations in order to recommend a season Hyatt Regency was performing badly (had low reservations). This would help them to give out some sort of discounts to improve their overall performance.

The monthly reservations trend looks like this:

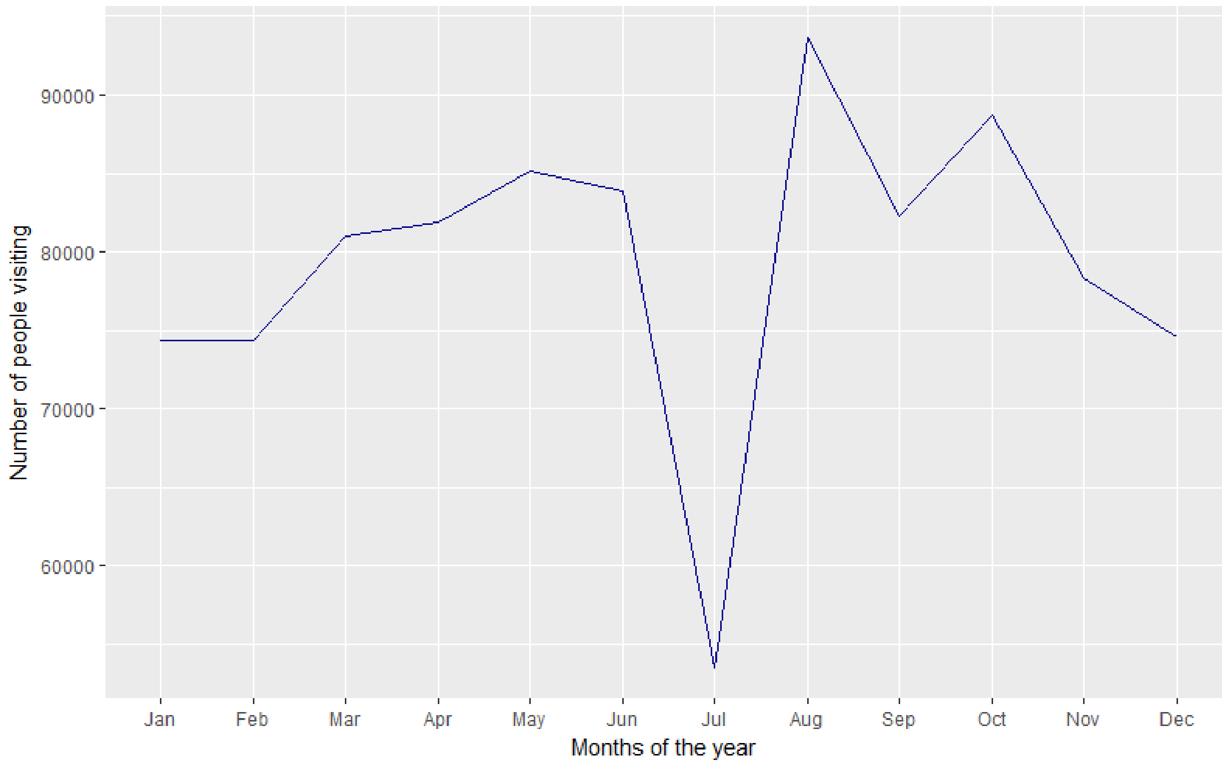

Fig. 16: Monthly Reservation Trends

We also had a look at how the likelihood to recommend varies by months of the year, the plot looks as below:

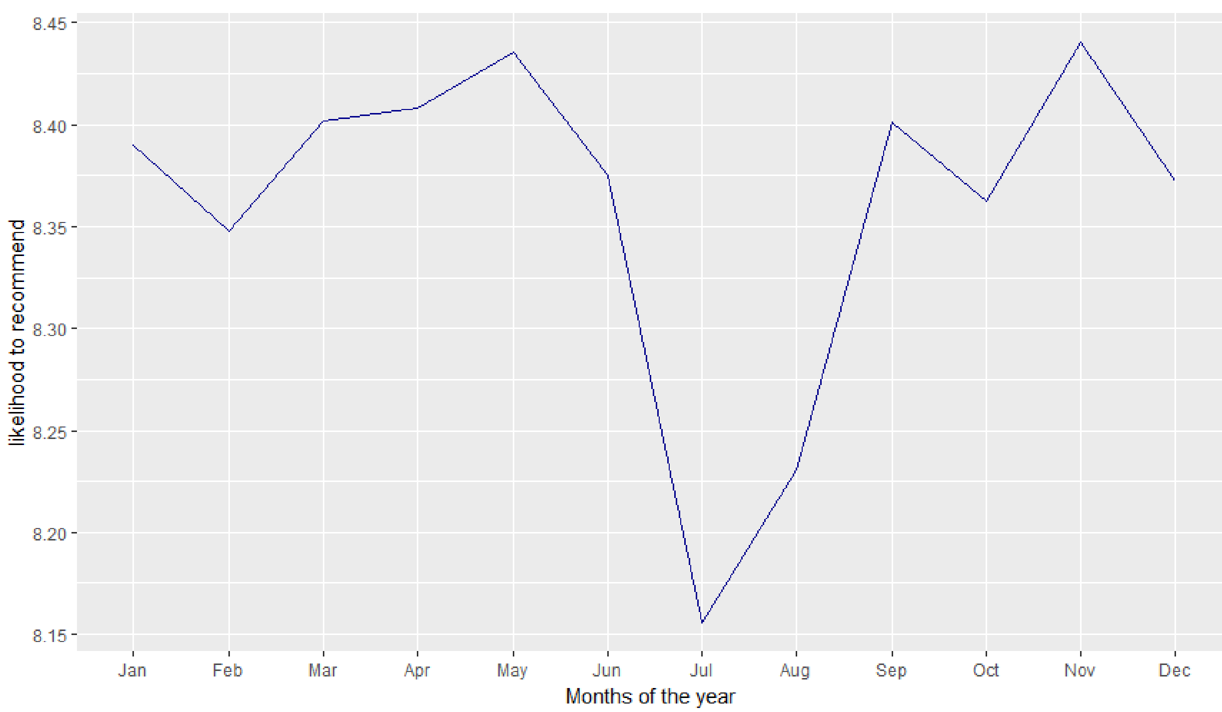

Fig. 17: Likelihood of recommend versus Months of the year

We see that July is the month where users dropped considerably and the NPS value dropped, so Hyatt regency could increase the number of people that visit them by offering some discounts in these months to increase their NPS.

3.4. ARULES

We did rule mining to generate association rules taking 'lhs' as the facilities and 'rhs' as the promoters and detractors separately for business and leisure users. Values of support and confidence for business was taken as 0.001 and 0.1 respectively. Values of support and confidence for leisure was taken as 0.3 and 0.8 respectively.
We cleaned the data by replacing the blanks and Passive to 'NA' and then omitting the NAs.
After that, Apriori algorithm was applied and rules were inspected. Good rules were then sorted taking lift value greater than 1.

For Business Users, we took following 5 facilities:

- Convention_PL
- Conference_PL
- Business.Center_PL
- Dry.Cleaning_PL
- Fitness.Center_PL

For Leisure, we concentrated on following 5 facilities:

- Casino_PL
- Spa_PL
- Boutique_PL
- Golf_PL
- Pool.Outdoor_PL

Rules for Business:

```
     lhs                          rhs                     support confidence     lift
[1]  {Business.Center_PL=N} => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[2]  {Convention_PL=Y,
      Business.Center_PL=N} => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[3]  {Conference_PL=N,
      Business.Center_PL=N} => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[4]  {Business.Center_PL=N,
      Dry.Cleaning_PL=Y}    => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[5]  {Business.Center_PL=N,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[6]  {Convention_PL=Y,
      Conference_PL=N,
      Business.Center_PL=N} => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[7]  {Convention_PL=Y,
      Business.Center_PL=N,
      Dry.Cleaning_PL=Y}    => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[8]  {Convention_PL=Y,
      Business.Center_PL=N,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[9]  {Conference_PL=N,
      Business.Center_PL=N,
      Dry.Cleaning_PL=Y}    => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[10] {Conference_PL=N,
      Business.Center_PL=N,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[11] {Business.Center_PL=N,
      Dry.Cleaning_PL=Y,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[12] {Convention_PL=Y,
      Conference_PL=N,
      Business.Center_PL=N,
      Dry.Cleaning_PL=Y}    => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[13] {Convention_PL=Y,
      Conference_PL=N,
      Business.Center_PL=N,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[14] {Convention_PL=Y,
      Business.Center_PL=N,
      Dry.Cleaning_PL=Y,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[15] {Conference_PL=N,
      Business.Center_PL=N,
      Dry.Cleaning_PL=Y,
      Fitness.Center_PL=Y}  => {NPS_Type=Detractor} 0.0315684  0.2383834 1.23303
[16] {Convention_PL=Y,
      Conference_PL=N,
      Business.Center_PL=N,
```

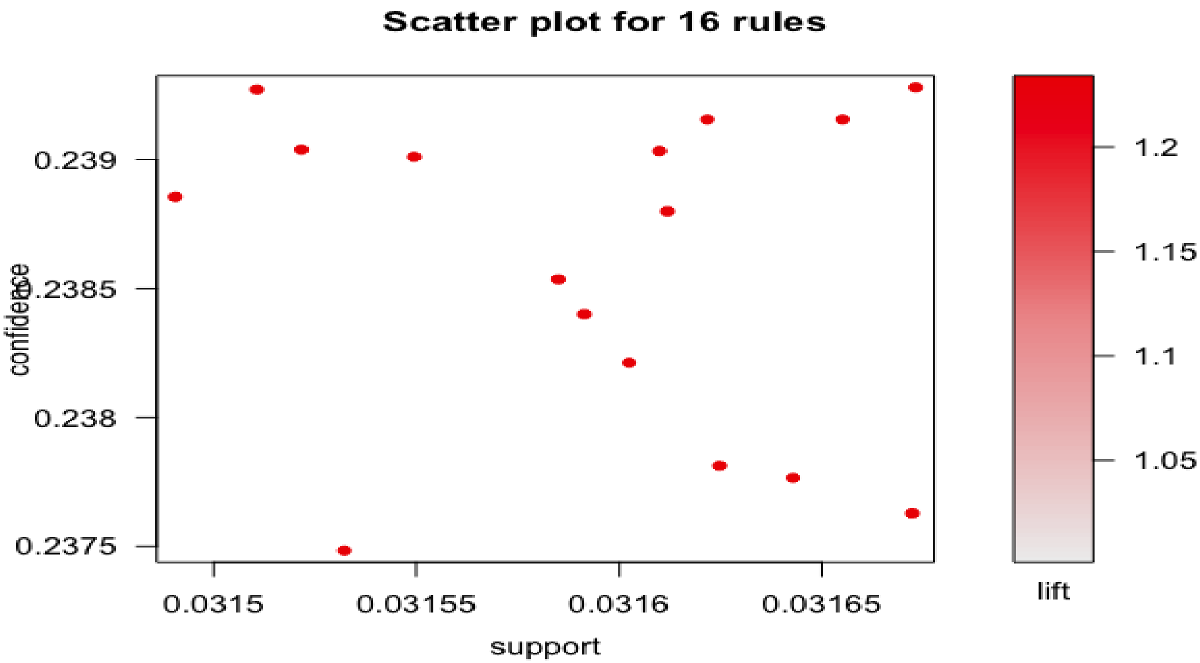

Fig. 18: Scatter Plot

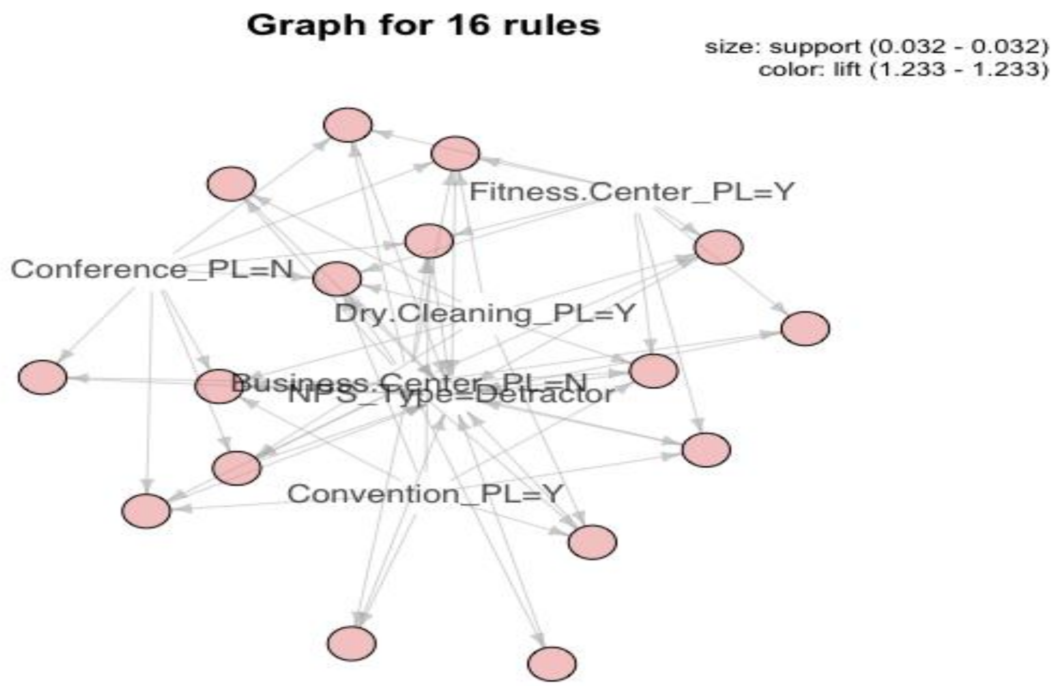

Fig. 19: Graph

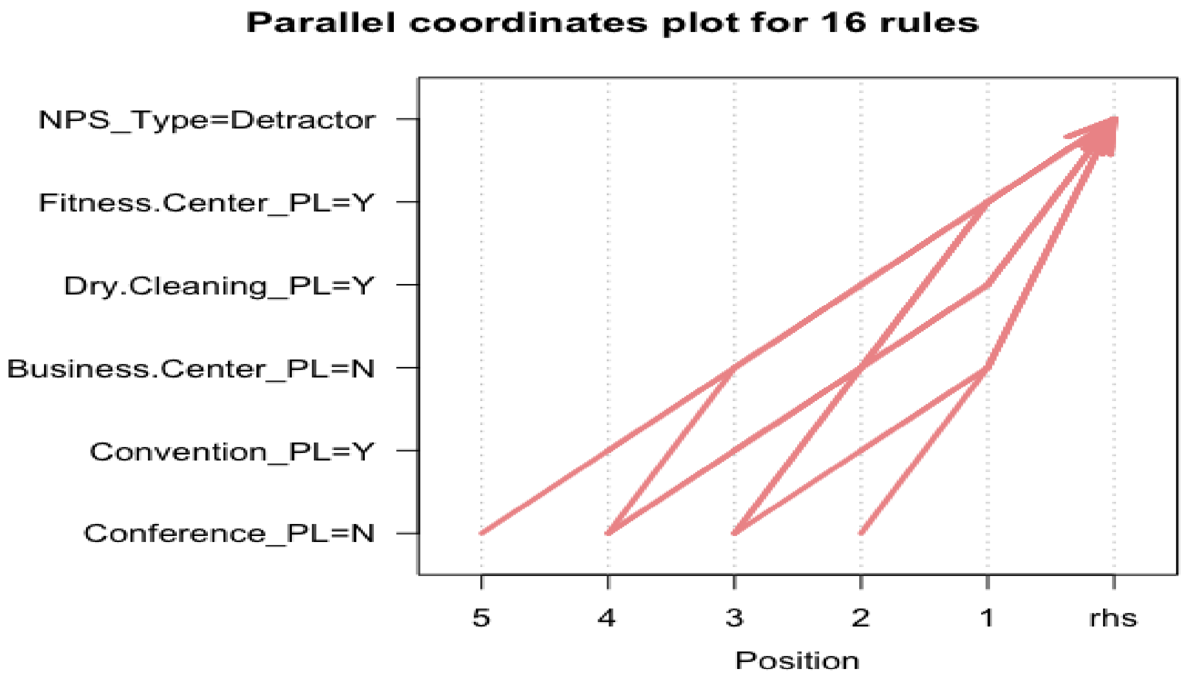

Fig. 20: Parallel Coordinates Plot

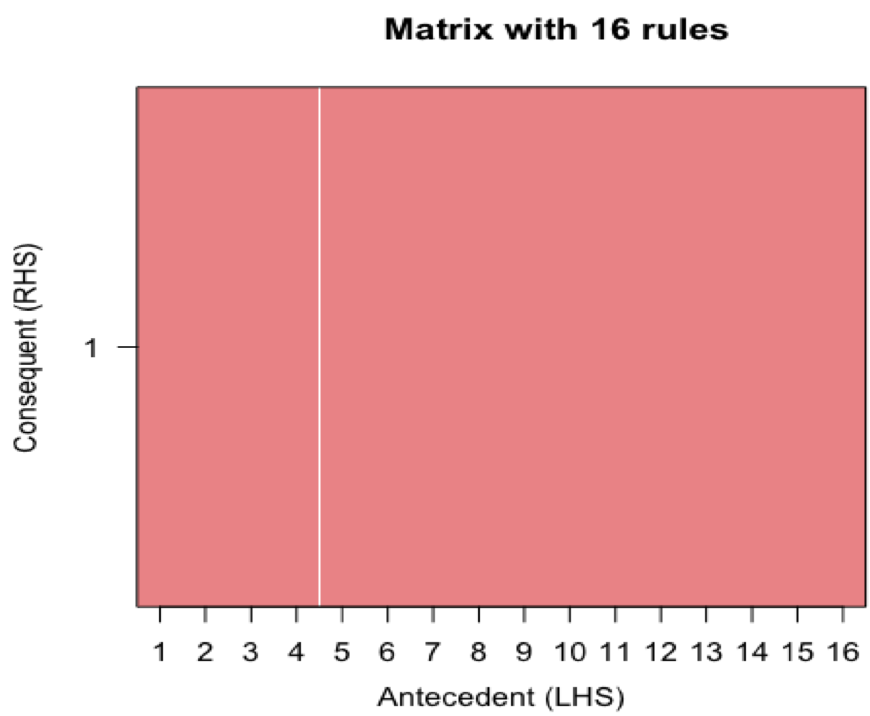

Fig. 21: Matrix

Fig. 22: Matrix 3D

Leisure:

```
Console ~/Downloads/
     lhs                      rhs                   support confidence     lift
[1]  {Spa_PL=Y}            => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[2]  {Spa_PL=Y,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[3]  {Casino_PL=N,
      Spa_PL=Y}            => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[4]  {Spa_PL=Y,
      Boutique_PL=N}       => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[5]  {Spa_PL=Y,
      Golf_PL=N}           => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[6]  {Casino_PL=N,
      Spa_PL=Y,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[7]  {Spa_PL=Y,
      Boutique_PL=N,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[8]  {Spa_PL=Y,
      Golf_PL=N,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[9]  {Casino_PL=N,
      Spa_PL=Y,
      Boutique_PL=N}       => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[10] {Casino_PL=N,
      Spa_PL=Y,
      Golf_PL=N}           => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[11] {Spa_PL=Y,
      Boutique_PL=N,
      Golf_PL=N}           => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[12] {Casino_PL=N,
      Spa_PL=Y,
      Boutique_PL=N,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[13] {Casino_PL=N,
      Spa_PL=Y,
      Golf_PL=N,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[14] {Spa_PL=Y,
      Boutique_PL=N,
      Golf_PL=N,
      Pool.Outdoor_PL=Y}   => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[15] {Casino_PL=N,
      Spa_PL=Y,
      Boutique_PL=N,
      Golf_PL=N}           => {NPS_Type=Detractor} 0.0790405  0.1969623 1.019074
[16] {Casino_PL=N,
      Spa_PL=Y,
      Boutique_PL=N
```

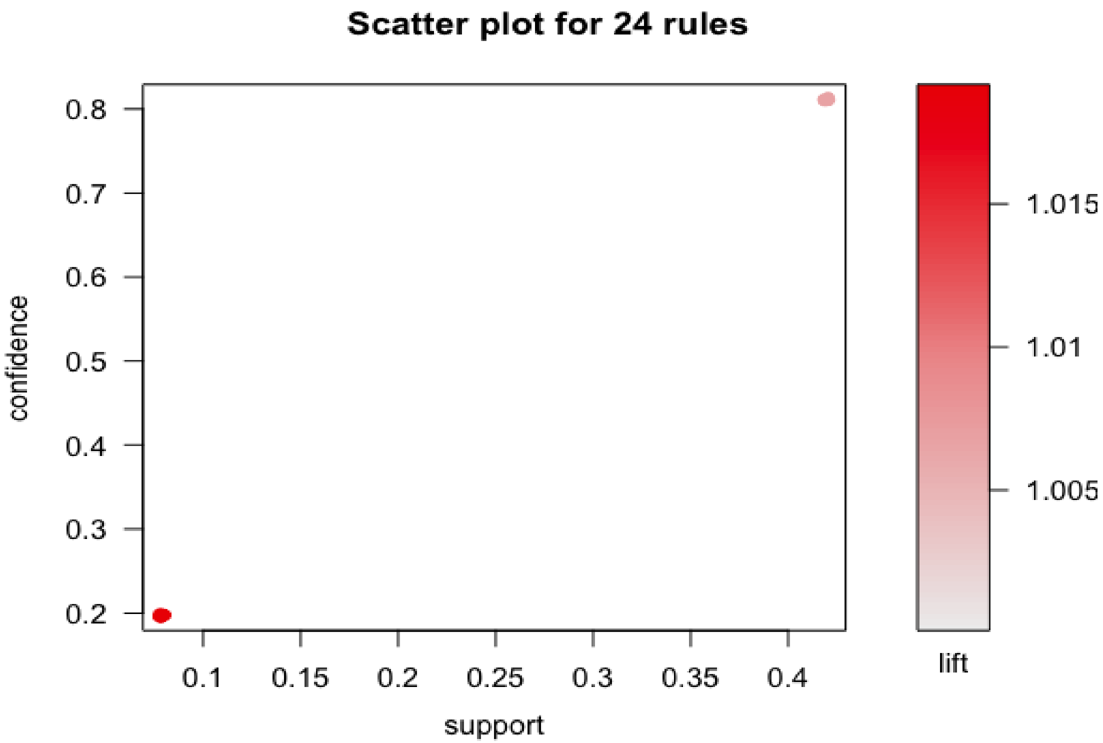

Fig. 23: Scatter Plot

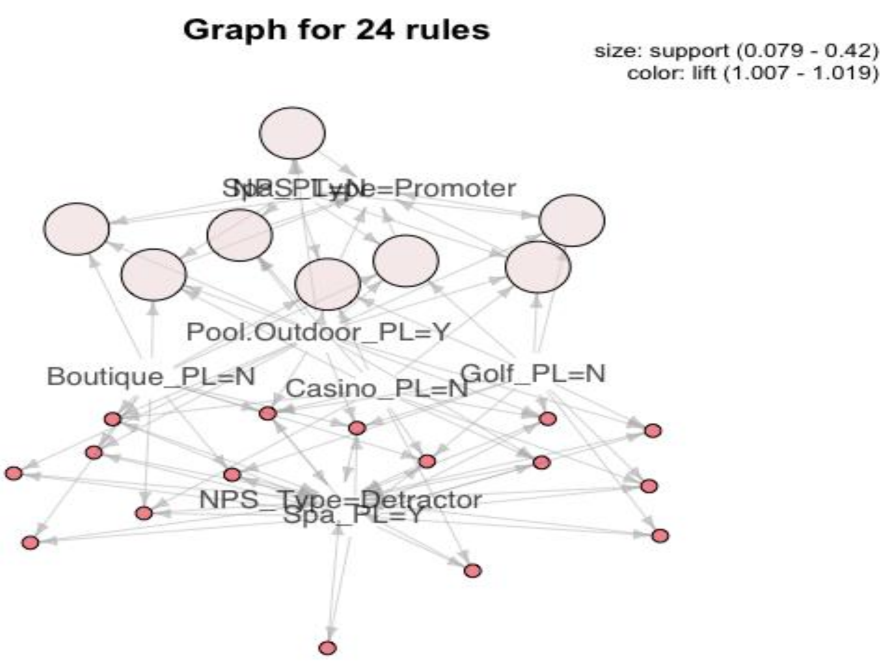

Fig. 24: Graph

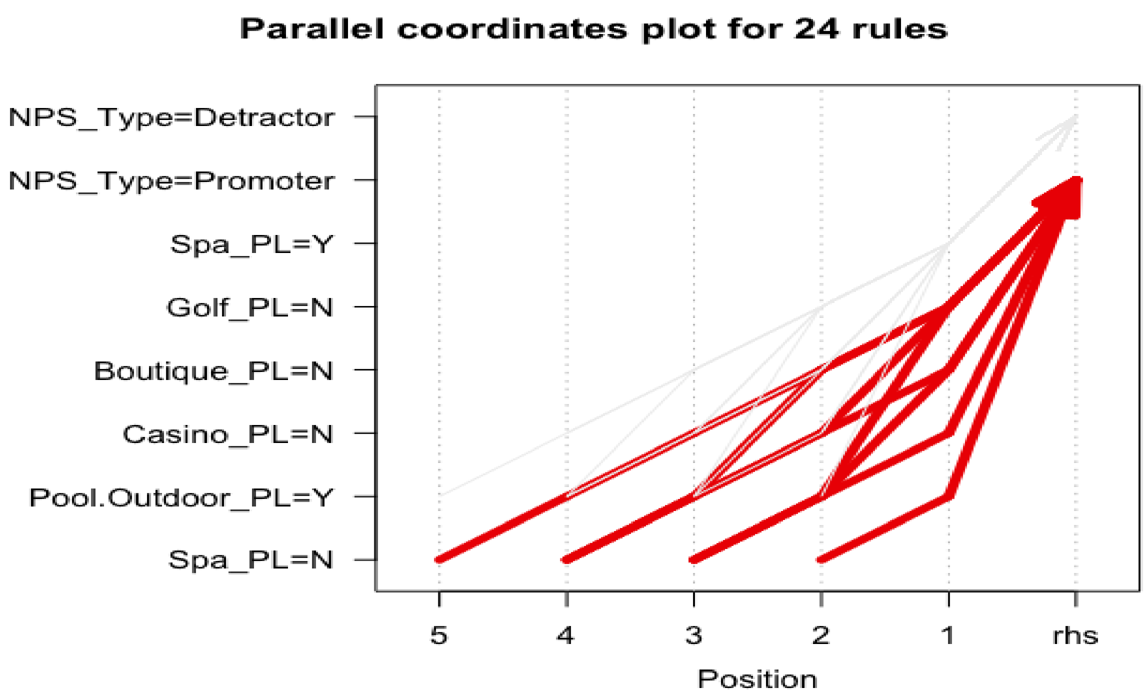

Fig. 25: Parallel Coordinated Plot

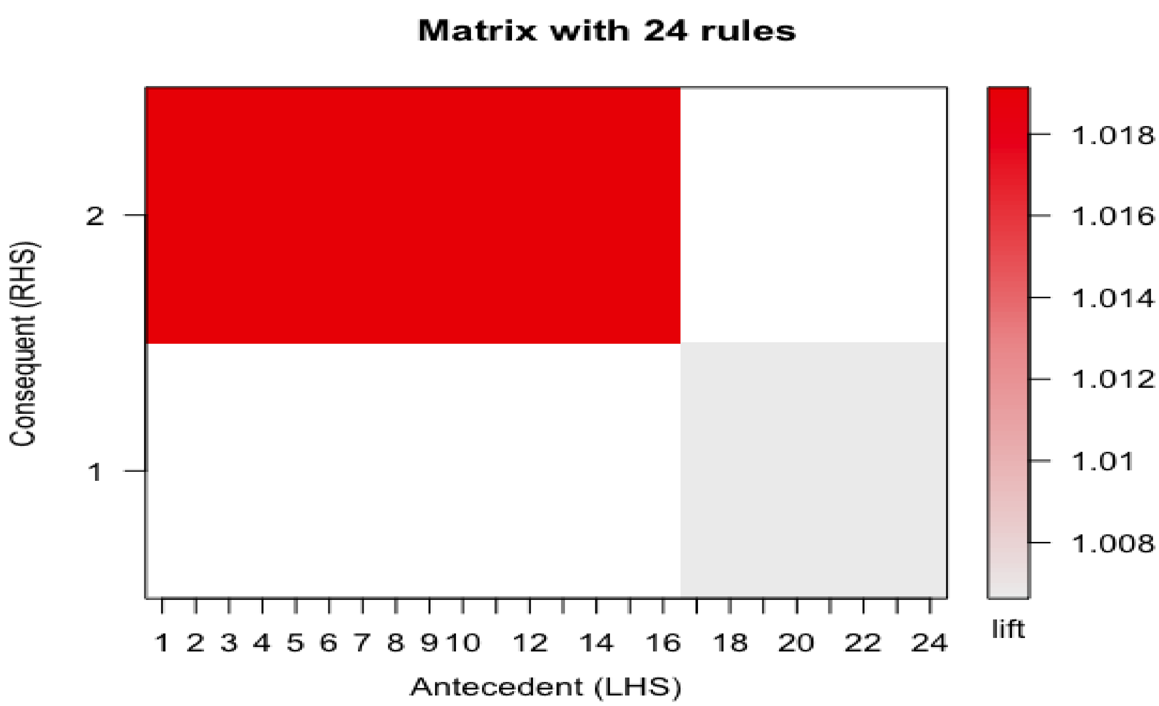

Fig. 26: Matrix

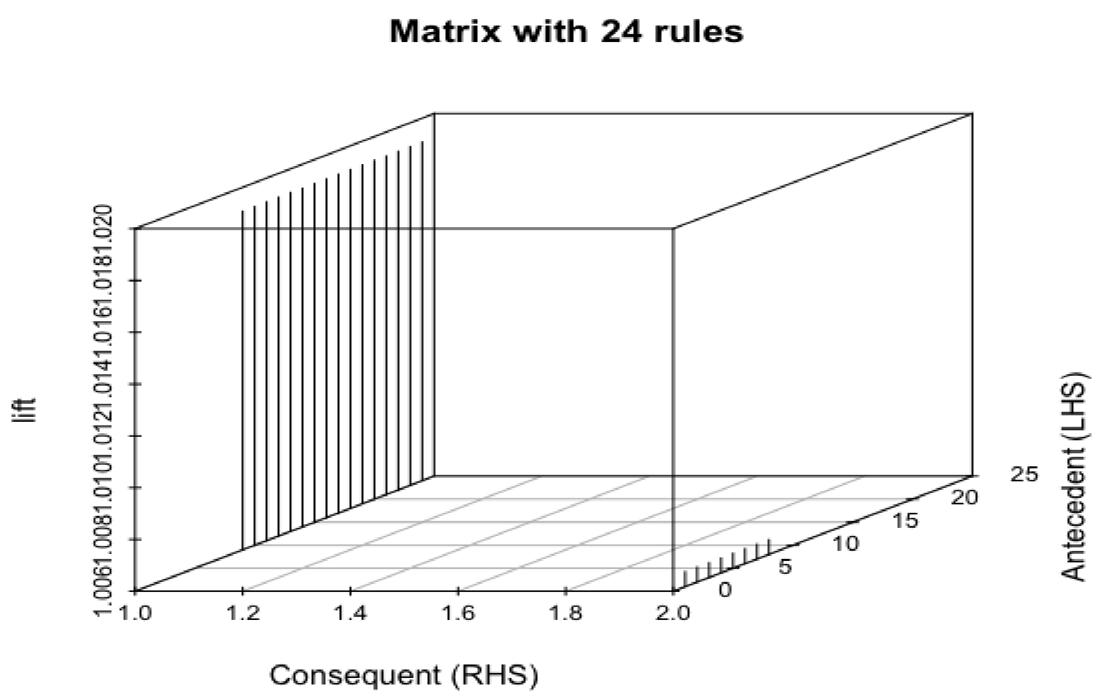

Fig. 27: Matrix 3D

4. MODELING:

4.1 Linear Modeling

We divided the dataset based on the purpose of visit namely business and leisure. We then applied linear regression modeling on likelihood to recommend against the business specific facilities that affect the NPS most. Similarly, Linear regression was applied on likelihood to recommend against the Leisure specific facilities that affect the NPS most.
Following observations were made after linear modelling:

Case 1: Likelihood to recommend against Business facilities
Features used: Business.Center_PL ,data1$Convention_PL
R-square value : 0.00054

Case 2: Likelihood to recommend against Leisure facilities
Features: Pool.Indoor_PL, data1$Pool.Outdoor_PL
R-square value: 0.0002698

Case 3: Likelihood to recommend against All facilities
Features: All
R-square value: 0.002405

No significant linear relationship was observed between likelihood to recommend and the facilities that affect the NPS most.

4.2 Logistic Modeling

We tried to analyze how the data and the selected features fit into the logistic model.

Case 1: Calculating the response(NPS type) based on business facilities.
Features: Business.Center_PL, Convention_PL, Limo.Service_PL, Valet.Parking_PL

After prediction Area Under the Curve (AUC) value was 0.5352.

Plot of performance:

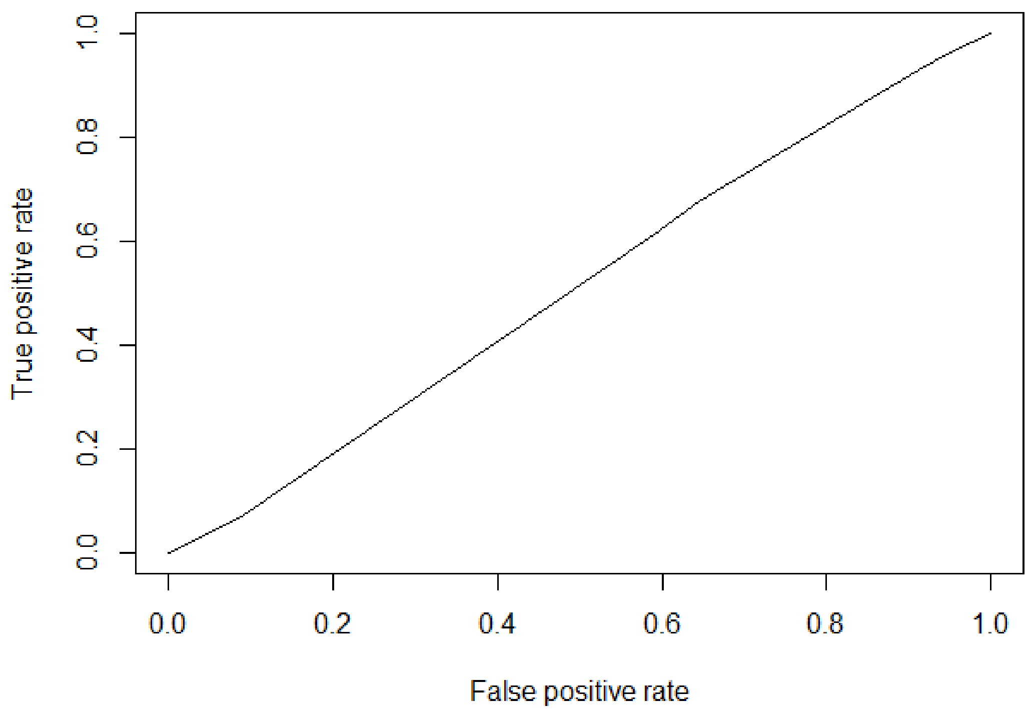

Fig. 28:Plot of Performance of Logistic Regression for Business Users

Case 2: Calculating the response(NPS type) based on leisure facilities.

Features: Pool.Indoor_PL, Pool.Outdoor_PL, Fitness.Center_PL, Casino_PL
After prediction Area Under the Curve (AUC) value was 0.5146511.

Plot of performance:

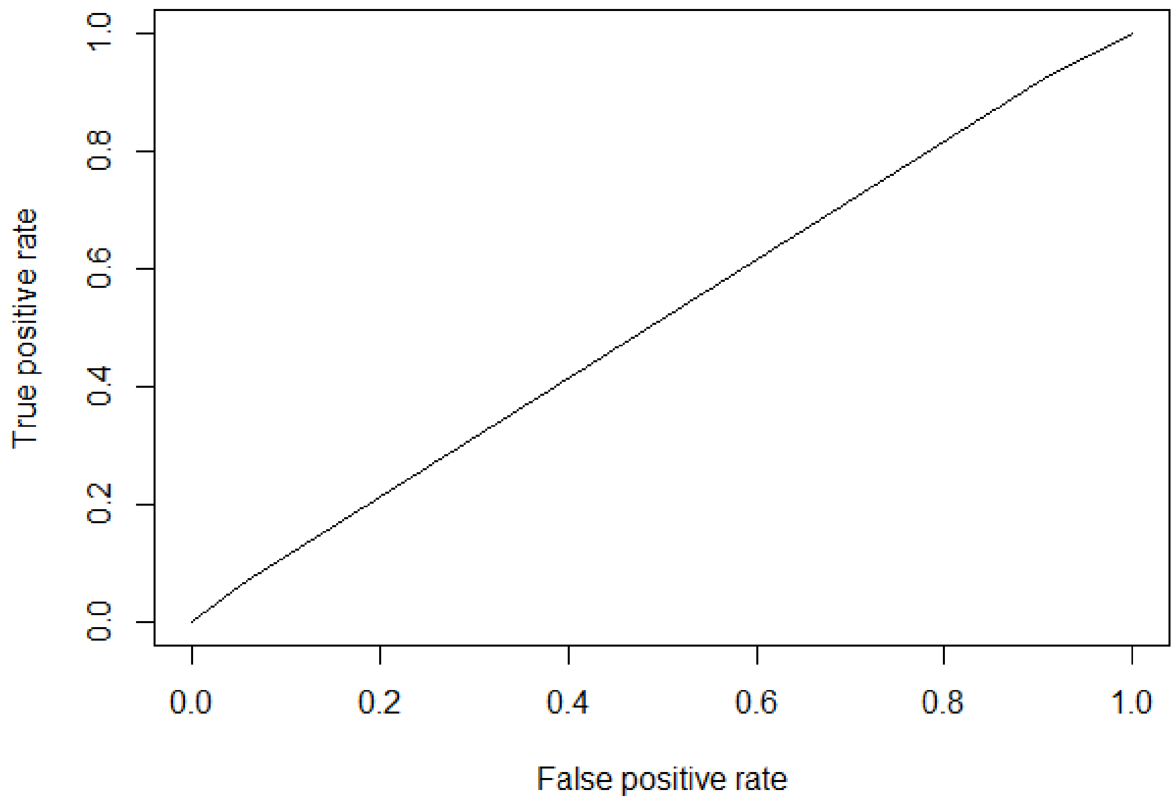

Fig. 29: Plot of Performance of Logistic Regression for Leisure Users

4.3 K- Nearest Neighbors

K nearest neighbors is a type of instance based learning. Here the input consists of k closest training examples in the feature space. The output is the average values of its k nearest neighbors.

In the case of Hyatt hotels, the input was taken as basic features of the hotel and the booking data of the customers.

The following features were used for computation.

- All Suites Presence
- Boutique Presence
- Casino Presence
- Convention Presence
- Elevator Presence
- Fitness Trainer Presence
- Indoor Corridor Presence
- Limo Service Presence
- Indoor Pool Presence
- Resort Presence
- Self-Parking Availability
- Ski Presence
- Valet Parking Presence
- Length of stay of customer
- If booking was a walk-in
- Reservation Status of booking
- Likelihood to Recommend
- Bell Staff Presence
- Business Center Presence
- Conference hall presence
- Dry Cleaning Presence
- Fitness Center Presence
- Golf Course Presence
- Laundry Presence
- Mini Bar Presence
- Outdoor Pool Presence
- Restaurant Presence
- Shuttle Service Presence
- Spa Presence
- Number of rooms in hotel
- Number of adults in a reservation
- Purpose of visit
- Types of room in hotel

Now based on these features of both hotel and customer, the booking data can be plotted in an n-dimensional space (n = 34 here). Each booking will have an point in the n-dimensions. Its most

likely that points in close proximity to others are similar. These points would have similar Net Promoter Score also.
So if a new customer makes a booking in some Hyatt hotel, then his data can be used to create a new point in the n-dimensions. Based on the distance to known data, k-nearest neighbors can be found and then these neighbors can vote on the Likelihood to Recommend Score of the new customer.
The distance metrics used in the model was Euclidian distance.

All the irrelevant data from the dataset were removed from the dataset. Categorical values in the records were factorized and converted to numerical values based on the factors. A random booking data was selected and its distance to all other booking records were calculated and then 100 nearest neighbors were selected whose average Likelihood to Recommend score was found. This is the most probable score of the new customer based on K-Nearest Neighbor Algorithm.

Results:
A random booking record was selected. The booking id for this record was 54857. The KNN logic was applied for this record and 100 nearest neighbors voted the Likelihood to Recommend Score to be 9.29 which in actual was 10.
So, the algorithm produced pretty promising results in comparison to the other regression models.

This model can be provided to the hotel as a deliverable. The hotel can then calculate the possible NPS of the booking. If the customer seems to be a detractor for the hotel, then the hotel management can make tweaks in his booking to provide him additional facilities so the customers has a good experience in the hotel and gives good rating to the hotel upon checking out and act as a promoter. The tweaks can be additional features which the model can be trained on and be decided by hotel management based on their available resources and feasibility.

APPENDIX

2. What facilities can a specific brand of Hyatt brand of Hyatt hotel at a given Zip Code improve to be at par with other Hyatt hotels of same brand.

```
#function to find if a facility is currently present in hotel or not
#facility == yes or no is repeated in bookings made in same hotel, so no need to give
weight to yes/no
currentFunction <- function(df, zipCode, brand, feature)
{
  df = data.frame(table(feature
                        [which(df$Postal.Code_PL == zipCode &
                                 df$Brand_PL == brand)]))

  y = df$Freq[which(df$Var1 == "Y")]
  n = df$Freq[which(df$Var1 == "N")]

  if(y > n)
    return("Y")
  else
    return("N")
}

#function to check for overall facility availibility in same brand of hotel
#NPS is used to calculate weight
overallFunction <- function(df, brand, feature)
{
  #calculating mean nps for y and n across various hotels
  meanNPS_y = mean(df$Likelihood_Recommend_H
                   [df$Brand_PL == brand & feature == "Y"],
                   na.rm = T)

  meanNPS_n = mean(df$Likelihood_Recommend_H
                   [df$Brand_PL == brand & feature == "N"],
                   na.rm = T)

  #calculating their weights (% of y/n in this case)
  df = data.frame(table(feature
                        [which(df$Brand_PL == brand)]))

  y = df$Freq[which(df$Var1 == "Y")]
  n = df$Freq[which(df$Var1 == "N")]

  #handling Nans as some hotel brands don't have a facility
  #e.g. all Hyatt Regency hotels lacks Casinos
  if(is.na(meanNPS_y))
    meanNPS_y = 0
```

```
  if(is.na(meanNPS_n))
    meanNPS_n = 0

  if(is.na(y))
    y = 0

  if(is.na(n))
    n = 0

  y_percent = y / (y + n)
  n_percent = n / (y + n)

  if((meanNPS_y * y_percent) > (meanNPS_n * n_percent))
    return("Y")
  else
    return("N")
}

#function to return desirable features of hotel in comparison to hotels of same brand
based on zipcode
#more than one brand of hotel can be present in a single city, hence narrowing down
to zipcode
amazingFunction <- function(df, brand, zipCode)
{
  #creating an empty data frame to store features which are desirable
  resultDF <- setNames(data.frame(matrix(ncol = 3, nrow = 0)),
                      c("label", "currentValue", "meanValue"))

  #NPS Score
  currentNPS_Score <- mean(df$Likelihood_Recommend_H
                          [(df$Postal.Code_PL == zipCode) & (df$Brand_PL == brand)],
                          na.rm = T)

  meanNPS_Score <- mean(df$Likelihood_Recommend_H
                          [df$Brand_PL == brand],
                          na.rm = T)

  resultDF<-rbind(resultDF, data.frame(label = "Likelihood to Recommend",
                                        currentValue = currentNPS_Score,
                                        meanValue = meanNPS_Score))

  #Overall Satisfaction
  current_satisfaction <- mean(df$Overall_Sat_H
                          [(df$Postal.Code_PL==zipCode) & (df$Brand_PL==brand)],
                          na.rm=T)

  mean_satisfaction <- mean(df$Overall_Sat_H
                          [df$Brand_PL==brand],
                          na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Overall Satisfaction",
```

```
                                      currentValue = current_satisfaction,
                                      meanValue = mean_satisfaction))

  #Guest Room Satisfaction
  current_roomSatisfaction <- mean(df$Guest_Room_H
                         [(df$Postal.Code_PL==zipCode) & (df$Brand_PL==brand)],
                         na.rm=T)

  mean_roomSatisfaction <- mean(df$Guest_Room_H
                         [df$Brand_PL==brand],
                         na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Guest Room Satisfaction",
                                      currentValue = current_roomSatisfaction,
                                      meanValue = mean_roomSatisfaction))

  #Tranquility
  current_tranquility <- mean(df$Tranquility_H
                                  [(df$Postal.Code_PL==zipCode) &
(df$Brand_PL==brand)],
                                  na.rm=T)

  mean_tranquility <- mean(df$Tranquility_H
                                  [df$Brand_PL==brand],
                                  na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Tranquility",
                                      currentValue = current_tranquility,
                                      meanValue = mean_tranquility))

  #Hotel Condition
  current_hotelCondition <- mean(df$Condition_Hotel_H
                            [(df$Postal.Code_PL==zipCode) & (df$Brand_PL==brand)],
                            na.rm=T)

  mean_hotelCondition <- mean(df$Condition_Hotel_H
                            [df$Brand_PL==brand],
                            na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Hotel Condition",
                                      currentValue = current_hotelCondition,
                                      meanValue = mean_hotelCondition))

  #Customer Service
  current_customerService <- mean(df$Customer_SVC_H
                                [(df$Postal.Code_PL==zipCode) &
(df$Brand_PL==brand)],
                                na.rm=T)

  mean_customerService <- mean(df$Customer_SVC_H
                                [df$Brand_PL==brand],
```

```
                                    na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Customer Service Quality",
                                       currentValue = current_customerService,
                                       meanValue = mean_customerService))

  #Staff Caring
  current_staffCaring <- mean(df$Staff_Cared_H
                                    [(df$Postal.Code_PL==zipCode) &
(df$Brand_PL==brand)],
                                    na.rm=T)

  mean_staffCaring <- mean(df$Staff_Cared_H
                                    [df$Brand_PL==brand],
                                    na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Staff Caring",
                                       currentValue = current_staffCaring,
                                       meanValue = mean_staffCaring))

  #Internet Satisfaction
  current_internetSatisfaction <- mean(df$Internet_Sat_H
                                [(df$Postal.Code_PL==zipCode) & (df$Brand_PL==brand)],
                                na.rm=T)

  mean_internetSatisfaction <- mean(df$Internet_Sat_H
                                [df$Brand_PL==brand],
                                na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Internet Satisfaction",
                                       currentValue = current_internetSatisfaction,
                                       meanValue = mean_internetSatisfaction))

  #Check In Quality
  current_checkIn <- mean(df$Check_In_H
                        [(df$Postal.Code_PL==zipCode) & (df$Brand_PL==brand)],
                        na.rm=T)

  mean_checkIn <- mean(df$Check_In_H
                     [df$Brand_PL==brand],
                     na.rm=T)

  resultDF<-rbind(resultDF, data.frame(label = "Check In Quality",
                                       currentValue = current_checkIn,
                                       meanValue = mean_checkIn))

  #F&B Experience Quality
  current_fnb <- mean(df$F.B_Overall_Experience_H
                        [(df$Postal.Code_PL==zipCode) & (df$Brand_PL==brand)],
                        na.rm=T)
```

```
mean_fnb <- mean(df$F.B_Overall_Experience_H
                     [df$Brand_PL==brand],
                     na.rm=T)

resultDF<-rbind(resultDF, data.frame(label = "F&B Experience Quality",
                                     currentValue = current_fnb,
                                     meanValue = mean_fnb))

#Bell staff presence
current_bell <- currentFunction(df, zipCode, brand, df$Bell.Staff_PL)
overall_bell <- overallFunction(df, brand, df$Bell.Staff_PL)

#add only if current and overall values are different
if(current_bell != overall_bell)
  resultDF<-rbind(resultDF, data.frame(label = "Bell Staff Presence",
                                     currentValue = current_bell,
                                     meanValue = overall_bell))

#Boutique presence
current_boutique <- currentFunction(df, zipCode, brand, df$Boutique_PL)
overall_boutique <- overallFunction(df, brand, df$Boutique_PL)

if(current_boutique != overall_boutique)
  resultDF<-rbind(resultDF, data.frame(label = "Boutique Presence",
                                     currentValue = current_boutique,
                                     meanValue = overall_boutique))

#Business Center presence
current_bc <- currentFunction(df, zipCode, brand, df$Business.Center_PL)
overall_bc <- overallFunction(df, brand, df$Business.Center_PL)

if(current_bc != overall_bc)
  resultDF<-rbind(resultDF, data.frame(label = "Business Center Presence",
                                     currentValue = current_bc,
                                     meanValue = overall_bc))

#Casino presence
current_casino <- currentFunction(df, zipCode, brand, df$Casino_PL)
overall_casino <- overallFunction(df, brand, df$Casino_PL)

if(current_casino != overall_casino)
  resultDF<-rbind(resultDF, data.frame(label = "Casino Presence",
                                     currentValue = current_casino,
                                     meanValue = overall_casino))

#Conference presence
current_conf <- currentFunction(df, zipCode, brand, df$Conference_PL)
overall_conf <- overallFunction(df, brand, df$Conference_PL)

if(current_conf != overall_conf)
  resultDF<-rbind(resultDF, data.frame(label = "Conference Presence",
```

```
                                     currentValue = current_conf,
                                     meanValue = overall_conf))

#Convention presence
current_conv <- currentFunction(df, zipCode, brand, df$Convention_PL)
overall_conv <- overallFunction(df, brand, df$Convention_PL)

if(current_conv != overall_conv)
  resultDF<-rbind(resultDF, data.frame(label = "Convention Presence",
                                     currentValue = current_conv,
                                     meanValue = overall_conv))

#Dry Cleaning presence
current_dry <- currentFunction(df, zipCode, brand, df$Dry.Cleaning_PL)
overall_dry <- overallFunction(df, brand, df$Dry.Cleaning_PL)

if(current_dry != overall_dry)
  resultDF<-rbind(resultDF, data.frame(label = "Dry Cleaning Presence",
                                     currentValue = current_dry,
                                     meanValue = overall_dry))

#Elevator presence
current_elev <- currentFunction(df, zipCode, brand, df$Elevators_PL)
overall_elev <- overallFunction(df, brand, df$Elevators_PL)

if(current_elev != overall_elev)
  resultDF<-rbind(resultDF, data.frame(label = "Elevators Presence",
                                     currentValue = current_elev,
                                     meanValue = overall_elev))

#Fitness Center Presence
current_fc <- currentFunction(df, zipCode, brand, df$Fitness.Center_PL)
overall_fc <- overallFunction(df, brand, df$Fitness.Center_PL)

if(current_fc != overall_fc)
  resultDF<-rbind(resultDF, data.frame(label = "Fitness Center Presence",
                                     currentValue = current_fc,
                                     meanValue = overall_fc))

#Fitness Trainer Presence
current_ft <- currentFunction(df, zipCode, brand, df$Fitness.Trainer_PL)
overall_ft <- overallFunction(df, brand, df$Fitness.Trainer_PL)

if(current_ft != overall_ft)
  resultDF<-rbind(resultDF, data.frame(label = "Fitness Trainer Presence",
                                     currentValue = current_ft,
                                     meanValue = overall_ft))

#Golf Space Presence
current_golf <- currentFunction(df, zipCode, brand, df$Golf_PL)
overall_golf <- overallFunction(df, brand, df$Golf_PL)
```

```
if(current_golf != overall_golf)
  resultDF<-rbind(resultDF, data.frame(label = "Golf Space Presence",
                                       currentValue = current_golf,
                                       meanValue = overall_golf))

#Indoor Corridors Presence
current_ic <- currentFunction(df, zipCode, brand, df$Indoor.Corridors_PL)
overall_ic <- overallFunction(df, brand, df$Indoor.Corridors_PL)

if(current_ic != overall_ic)
  resultDF<-rbind(resultDF, data.frame(label = "Indoor Corridor Presence",
                                       currentValue = current_ic,
                                       meanValue = overall_ic))

#Laundry Presence
current_laun <- currentFunction(df, zipCode, brand, df$Laundry_PL)
overall_laun <- overallFunction(df, brand, df$Laundry_PL)

if(current_laun != overall_laun)
  resultDF<-rbind(resultDF, data.frame(label = "Laundry Presence",
                                       currentValue = current_laun,
                                       meanValue = overall_laun))

#Limo Service Presence
current_limo <- currentFunction(df, zipCode, brand, df$Limo.Service_PL)
overall_limo <- overallFunction(df, brand, df$Limo.Service_PL)

if(current_limo != overall_limo)
  resultDF<-rbind(resultDF, data.frame(label = "Limo Service Presence",
                                       currentValue = current_limo,
                                       meanValue = overall_limo))

#Mini Bar Presence
current_miniBar <- currentFunction(df, zipCode, brand, df$Mini.Bar_PL)
overall_miniBar <- overallFunction(df, brand, df$Mini.Bar_PL)

if(current_miniBar != overall_miniBar)
  resultDF<-rbind(resultDF, data.frame(label = "Mini Bar Presence",
                                       currentValue = current_miniBar,
                                       meanValue = overall_miniBar))

#Indoor Pool Presence
current_ip <- currentFunction(df, zipCode, brand, df$Pool.Indoor_PL)
overall_ip <- overallFunction(df, brand, df$Pool.Indoor_PL)

if(current_ip != overall_ip)
  resultDF<-rbind(resultDF, data.frame(label = "Indoor Pool Presence",
                                       currentValue = current_ip,
                                       meanValue = overall_ip))
```

```
#Outdoor Pool Presence
current_op <- currentFunction(df, zipCode, brand, df$Pool.Outdoor_PL)
overall_op <- overallFunction(df, brand, df$Pool.Outdoor_PL)

if(current_op != overall_op)
  resultDF<-rbind(resultDF, data.frame(label = "Outdoor Pool Presence",
                                       currentValue = current_op,
                                       meanValue = overall_op))

#Regency Grand Club Presence
current_rgc <- currentFunction(df, zipCode, brand, df$Regency.Grand.Club_PL)
overall_rgc <- overallFunction(df, brand, df$Regency.Grand.Club_PL)

if(current_rgc != overall_rgc)
  resultDF<-rbind(resultDF, data.frame(label = "Regency Grand Club Presence",
                                       currentValue = current_rgc,
                                       meanValue = overall_rgc))

#Resort Presence
current_resort <- currentFunction(df, zipCode, brand, df$Resort_PL)
overall_resort <- overallFunction(df, brand, df$Resort_PL)

if(current_resort != overall_resort)
  resultDF<-rbind(resultDF, data.frame(label = "Resort Presence",
                                       currentValue = current_resort,
                                       meanValue = overall_resort))

#Restaurant Presence
current_restaurant <- currentFunction(df, zipCode, brand, df$Restaurant_PL)
overall_restaurant <- overallFunction(df, brand, df$Restaurant_PL)

if(current_restaurant != overall_restaurant)
  resultDF<-rbind(resultDF, data.frame(label = "Restaurant Presence",
                                       currentValue = current_restaurant,
                                       meanValue = overall_restaurant))

#Self Parking Presence
current_sp <- currentFunction(df, zipCode, brand, df$Self.Parking_PL)
overall_sp <- overallFunction(df, brand, df$Self.Parking_PL)

if(current_sp != overall_sp)
  resultDF<-rbind(resultDF, data.frame(label = "Self Parking Presence",
                                       currentValue = current_sp,
                                       meanValue = overall_sp))

#Shuttle Service Presence
current_ss <- currentFunction(df, zipCode, brand, df$Shuttle.Service_PL)
overall_ss <- overallFunction(df, brand, df$Shuttle.Service_PL)

if(current_ss != overall_ss)
  resultDF<-rbind(resultDF, data.frame(label = "Shuttle Service Presence",
```

```
                                        currentValue = current_ss,
                                        meanValue = overall_ss))

  #Ski Presence
  current_ski <- currentFunction(df, zipCode, brand, df$Ski_PL)
  overall_ski <- overallFunction(df, brand, df$Ski_PL)

  if(current_ski != overall_ski)
    resultDF<-rbind(resultDF, data.frame(label = "Ski Presence",
                                        currentValue = current_ski,
                                        meanValue = overall_ski))

  #Spa Presence
  current_spa <- currentFunction(df, zipCode, brand, df$Spa_PL)
  overall_spa <- overallFunction(df, brand, df$Spa_PL)

  if(current_spa != overall_spa)
    resultDF<-rbind(resultDF, data.frame(label = "Spa Presence",
                                        currentValue = current_spa,
                                        meanValue = overall_spa))

  #Spa in Fitness Center Presence
  current_spaFC <- currentFunction(df, zipCode, brand,
df$Spa.services.in.fitness.center_PL)
  overall_spaFC <- overallFunction(df, brand, df$Spa.services.in.fitness.center_PL)

  if(current_spaFC != overall_spaFC)
    resultDF<-rbind(resultDF, data.frame(label = "Spa in Fitness Center Presence",
                                        currentValue = current_spaFC,
                                        meanValue = overall_spaFC))

  #Spa Online Booking Presence
  current_spaOn <- currentFunction(df, zipCode, brand, df$Spa.online.booking_PL)
  overall_spaOn <- overallFunction(df, brand, df$Spa.online.booking_PL)

  if(current_spaOn != overall_spaOn)
    resultDF<-rbind(resultDF, data.frame(label = "Spa Online Booking Presence",
                                        currentValue = current_spaOn,
                                        meanValue = overall_spaOn))

  #Spa F&B Offering Presence
  current_spaFB <- currentFunction(df, zipCode, brand, df$Spa.F.B.offering_PL)
  overall_spaFB <- overallFunction(df, brand, df$Spa.F.B.offering_PL)

  if(current_spaFB != overall_spaFB)
    resultDF<-rbind(resultDF, data.frame(label = "Spa F&B Offering Presence",
                                        currentValue = current_spaFB,
                                        meanValue = overall_spaFB))

  #Valet Presence
  current_valet <- currentFunction(df, zipCode, brand, df$Valet.Parking_PL)
```

```
  overall_valet <- overallFunction(df, brand, df$Valet.Parking_PL)

  if(current_valet != overall_valet)
    resultDF<-rbind(resultDF, data.frame(label = "Valet Parking Presence",
                                         currentValue = current_valet,
                                         meanValue = overall_valet))

  return(resultDF)
}
```

3.2.2. Highest number of reservations

(1)

```
finaldf <- readRDS("finaldf.rds")
which.max(table(finaldf$State_PL))

newdf<- data.frame(table(finaldf$State_PL))
View(newdf)
install.packages("ggplot2")
library("ggplot2")
install.packages("maps")
library("maps")

install.packages("mapproj")
library("mapproj")

us <- map_data("state")
View(us)

newdf$Var1 = tolower(newdf$Var1)
newdf = na.omit(newdf)
#newdf = newdf[-45,]
# map representing highest number of reservations
Map <- ggplot(newdf,aes(map_id = Var1))
Map <- Map + geom_map(map = us, aes(color = newdf$Freq, fill = newdf$Freq))
Map <- Map + expand_limits(x = us$long, y = us$lat)
Map <- Map + coord_map()
Map = Map + ggtitle("Frequency of reservations made across states in the US")
Map
```

3.2. Code for selecting the brand and purpose of visits

```
finaldf_cali_brand =finaldf_cali
finaldf_cali_brand$NPS_Type[which(finaldf_cali_brand$NPS_Type=="")] <- "Passive"
finaldf_cali_brand$NPS_Type <- factor(finaldf_cali_brand$NPS_Type)

table(finaldf_cali_brand$NPS_Type)

finaldf_cali_brand = finaldf_cali_brand[finaldf_cali_brand$NPS_Type != "Passive",]
finaldf_cali_brand$NPS_Type <- factor(finaldf_cali_brand$NPS_Type)
```

```
########## grouping by brand type############
m = (tapply(finaldf_cali_brand$s.no, list(finaldf_cali_brand$NPS_Type,
finaldf_cali_brand$Brand_PL ), length))
#remove null columns
m <- m[,!(colnames(m) %in% c('Hyatt Zilara','Hyatt Ziva'))]
m
View(m)

m_d = m[-2,]
m_p = m[-1,]
m_p
m_d

m_d_normalize = (m_d - min(m_d))/(max(m_d) - min(m_d))
m_p_normalize = (m_p - min(m_p))/(max(m_p) - min(m_p))

#plotting graphs
pal <- colorRampPalette(c("light gray", "red"))
colors <- pal(10)
barplot(m_p_normalize,col=colors, title("Number of promoters"))
barplot(m_d_normalize,col= colors, title("Number of detractors"))

library(plotrix)
usdata.regency <- usdata[which(usdata$Brand_PL == "Hyatt Regency"),]
mytable <- table(usdata.regency$POV_CODE_C)
lbls <- paste(levels(usdata$POV_CODE_C))
pie3D(mytable, labels = lbls, main = "Purpose of visit")
```

3.3.1.1. Code for the bar plots for the facilities for business users

```
cali_regency <- readRDS("cali_regency.rds")

#### for business users
#Business.Center_PL

regency_b_business_center = cali_regency[cali_regency$POV_CODE_C == "BUSINESS",]
table(regency_b_business_center$NPS_Type)
View(regency_b_business_center)

nrow(regency_b_business_center)
regency_b_business_center[regency_b_business_center==""] <- NA
regency_b_business_center[regency_b_business_center=="Passive"] <- NA
table(regency_b_business_center$NPS_Type)

regency_b_business_center                                                     <-
regency_b_business_center[!is.na(regency_b_business_center$NPS_Type),]
table(regency_b_business_center$NPS_Type)

regency_b_business_center$NPS_Type <- factor(regency_b_business_center$NPS_Type)
table(regency_b_business_center$NPS_Type)
```

```
d2<-regency_b_business_center[,c("Business.Center_PL","NPS_Type")]
View(d2)

d2$NPS_Type <- as.numeric(d2$NPS_Type)
#d2 <- lapply(d2,as.numeric)
d2_df <- data.frame(d2)

d2$s.no2 = seq.int(nrow(d2_df))

m2 = tapply(d2$s.no2, list(d2$NPS_Type, d2$Business.Center_PL), length)
m2
#m2= m1[,-1]

barplot(m2, beside = TRUE, col =c ( "pink", "blue"), main = "Detractor and Promoters
based on Business Center availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd =
2, lty=1)



#Conference PL

regency_b_Conference = cali_regency[cali_regency$POV_CODE_C == "BUSINESS",]
table(regency_b_Conference$NPS_Type)
View(regency_b_Conference)

nrow(regency_b_Conference)
regency_b_Conference[regency_b_Conference==""] <- NA
regency_b_Conference[regency_b_Conference=="Passive"] <- NA
table(regency_b_Conference$NPS_Type)

regency_b_Conference <- regency_b_Conference[!is.na(regency_b_Conference$NPS_Type),]
table(regency_b_Conference$NPS_Type)

regency_b_Conference$NPS_Type <- factor(regency_b_Conference$NPS_Type)
table(regency_b_Conference$NPS_Type)
d2<-regency_b_Conference[,c("Convention_PL","NPS_Type")]
View(d2)

d2$NPS_Type <- as.numeric(d2$NPS_Type)
#d2 <- lapply(d2,as.numeric)
d2_df <- data.frame(d2)

d2$s.no2 = seq.int(nrow(d2_df))

m2 = tapply(d2$s.no2, list(d2$NPS_Type, d2$Conference_PL), length)
m2
#m2= m1[,-1]

barplot(m2, beside = TRUE, col =c ( "pink", "blue"), main = "Detractor and Promoters
based on Conference availability")
```

```
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd =
2, lty=1)


#Convention PL

regency_b_Convention = cali_regency[cali_regency$POV_CODE_C == "BUSINESS",]
table(regency_b_Convention$NPS_Type)
View(regency_b_Convention)

nrow(regency_b_Convention)
regency_b_Convention[regency_b_Convention==""] <- NA
regency_b_Convention[regency_b_Convention=="Passive"] <- NA
table(regency_b_Convention$NPS_Type)

regency_b_Convention <- regency_b_Convention[!is.na(regency_b_Convention$NPS_Type),]
table(regency_b_Convention$NPS_Type)

regency_b_Convention$NPS_Type <- factor(regency_b_Convention$NPS_Type)
table(regency_b_Convention$NPS_Type)
d2<-regency_b_Convention[,c("Convention_PL","NPS_Type")]
View(d2)

d2$NPS_Type <- as.numeric(d2$NPS_Type)
#d2 <- lapply(d2,as.numeric)
d2_df <- data.frame(d2)

d2$s.no2 = seq.int(nrow(d2_df))

m2 = tapply(d2$s.no2, list(d2$NPS_Type, d2$Convention_PL), length)
m2
#m2= m1[,-1]

barplot(m2, beside = TRUE, col =c ( "pink", "blue"), main = "Detractor and Promoters
based on Convention availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd =
2, lty=1)


#Dry.Cleaning_PL

regency_b_Dry_Cleaning = cali_regency[cali_regency$POV_CODE_C == "BUSINESS",]
table(regency_b_Dry_Cleaning$NPS_Type)
View(regency_b_Dry_Cleaning)

nrow(regency_b_Dry_Cleaning)
regency_b_Dry_Cleaning[regency_b_Dry_Cleaning==""] <- NA
regency_b_Dry_Cleaning[regency_b_Dry_Cleaning=="Passive"] <- NA
table(regency_b_Dry_Cleaning$NPS_Type)
```

```
regency_b_Dry_Cleaning                                              <-
regency_b_Dry_Cleaning[!is.na(regency_b_Dry_Cleaning$NPS_Type),]
table(regency_b_Dry_Cleaning$NPS_Type)

regency_b_Dry_Cleaning$NPS_Type <- factor(regency_b_Dry_Cleaning$NPS_Type)
table(regency_b_Dry_Cleaning$NPS_Type)
d2<-regency_b_Dry_Cleaning[,c("Dry.Cleaning _PL","NPS_Type")]
View(d2)

d2$NPS_Type <- as.numeric(d2$NPS_Type)
#d2 <- lapply(d2,as.numeric)
d2_df <- data.frame(d2)

d2$s.no2 = seq.int(nrow(d2_df))

m2 = tapply(d2$s.no2, list(d2$NPS_Type, d2$Dry.Cleaning_PL), length)
m2
#m2= m1[,-1]

barplot(m2, beside = TRUE, col =c ( "pink", "blue"), main = "Detractor and Promoters
based on Dry Cleaning availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd =
2, lty=1)


#Fitness.Center_PL

regency_b_Fitness_Center = cali_regency[cali_regency$POV_CODE_C == "BUSINESS",]
table(regency_b_Fitness_Center$NPS_Type)
View(regency_b_Fitness_Center)

nrow(regency_b_Fitness_Center)
regency_b_Fitness_Center[regency_b_Fitness_Center==""] <- NA
regency_b_Fitness_Center[regency_b_Fitness_Center=="Passive"] <- NA
table(regency_b_Fitness_Center$NPS_Type)

regency_b_Fitness_Center                                            <-
regency_b_Fitness_Center[!is.na(regency_b_Fitness_Center$NPS_Type),]
table(regency_b_Fitness_Center$NPS_Type)

regency_b_Fitness_Center$NPS_Type <- factor(regency_b_Fitness_Center$NPS_Type)
table(regency_b_Fitness_Center$NPS_Type)
d2<-regency_b_Fitness_Center[,c("Fitness.Center _PL","NPS_Type")]
View(d2)

d2$NPS_Type <- as.numeric(d2$NPS_Type)
#d2 <- lapply(d2,as.numeric)
d2_df <- data.frame(d2)

d2$s.no2 = seq.int(nrow(d2_df))
```

```
m2 = tapply(d2$s.no2, list(d2$NPS_Type, d2$Fitness.Center_PL), length)
m2
#m2= m1[,-1]

barplot(m2, beside = TRUE, col =c ( "pink", "blue"), main = "Detractor and Promoters
based on Fitness Center availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd =
2, lty=1)
```

3.3.1.2. Code for the bar plots for the facilities for leisure users

```
cali_regency <- readRDS("cali_regency.rds")

#### for leisure users
#Casino pl

regency_b_casino = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_casino$NPS_Type)
View(regency_b_casino)

nrow(regency_b_casino)
regency_b_casino[regency_b_casino==""] <- NA
regency_b_casino[regency_b_casino=="Passive"] <- NA
table(regency_b_casino$NPS_Type)

regency_b_casino <- regency_b_casino[!is.na(regency_b_casino$NPS_Type),]
table(regency_b_casino$NPS_Type)

regency_b_casino$NPS_Type <- factor(regency_b_casino$NPS_Type)
table(regency_b_casino$NPS_Type)
d2<-regency_b_casino[,c("Casino_PL","NPS_Type")]
View(d2)

d2$NPS_Type <- as.numeric(d2$NPS_Type)
#d2 <- lapply(d2,as.numeric)
d2_df <- data.frame(d2)

d2$s.no2 = seq.int(nrow(d2_df))

m2 = tapply(d2$s.no2, list(d2$NPS_Type, d2$Casino_PL), length)
m2
#m2= m1[,-1]

barplot(m2, beside = TRUE, col =c ( "pink", "blue"), main = "Detractor and promotors
based on casino availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2,
lty=1)
```

```
#spa

regency_b_spa = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_spa$NPS_Type)
View(regency_b_spa)

nrow(regency_b_spa)
regency_b_spa[regency_b_spa==""] <- NA
regency_b_spa[regency_b_spa=="Passive"] <- NA
table(regency_b_spa$NPS_Type)

regency_b_spa <- regency_b_spa[!is.na(regency_b_spa$NPS_Type),]
table(regency_b_spa$NPS_Type)

regency_b_spa$NPS_Type <- factor(regency_b_spa$NPS_Type)
table(regency_b_spa$NPS_Type)
d3<-regency_b_spa[,c("Spa_PL","NPS_Type")]
View(d3)

d3$NPS_Type <- as.numeric(d3$NPS_Type)
#d3 <- lapply(d3,as.numeric)

d3_df <- data.frame(d3)

d3$s.no3 = seq.int(nrow(d3_df))

m3 = tapply(d3$s.no3, list(d3$NPS_Type, d3$Spa_PL), length)
m3
#m2= m1[,-1]

barplot(m3, beside = TRUE, col =c ( "pink", "blue"),main = "Detractor and promotors
based on Spa availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2,
lty=1)


#boutique_service

regency_b_boutique = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_boutique$NPS_Type)
View(regency_b_boutique)

nrow(regency_b_boutique)
regency_b_boutique[regency_b_boutique==""] <- NA
regency_b_boutique[regency_b_boutique=="Passive"] <- NA
table(regency_b_boutique$NPS_Type)

regency_b_boutique <- regency_b_boutique[!is.na(regency_b_boutique$NPS_Type),]
table(regency_b_boutique$NPS_Type)

regency_b_boutique$NPS_Type <- factor(regency_b_boutique$NPS_Type)
```

```
table(regency_b_boutique$NPS_Type)
d4<-regency_b_boutique[,c("Boutique_PL","NPS_Type")]
View(d4)

d4$NPS_Type <- as.numeric(d4$NPS_Type)
#d4 <- lapply(d4,as.numeric)

d4_df <- data.frame(d4)

d4$s.no4 = seq.int(nrow(d4_df))

m4 = tapply(d4$s.no4, list(d4$NPS_Type, d4$Boutique_PL), length)
m4
#m2= m1[,-1]

barplot(m4, beside = TRUE, col =c ( "pink", "blue"), main="Detractor and promotors
based on Boutique availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2,
lty=1)


#Golf

regency_b_golf = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_golf$NPS_Type)
View(regency_b_golf)

nrow(regency_b_golf)
regency_b_golf[regency_b_golf==""] <- NA
regency_b_golf[regency_b_golf=="Passive"] <- NA
table(regency_b_golf$NPS_Type)

regency_b_golf <- regency_b_golf[!is.na(regency_b_golf$NPS_Type),]
table(regency_b_golf$NPS_Type)

regency_b_golf$NPS_Type <- factor(regency_b_golf$NPS_Type)
table(regency_b_golf$NPS_Type)
d5<-regency_b_golf[,c("Golf_PL","NPS_Type")]
View(d5)

d5$NPS_Type <- as.numeric(d5$NPS_Type)
#d5 <- lapply(d5,as.numeric)

d5_df <- data.frame(d5)

d5$s.no5 = seq.int(nrow(d5_df))

m5 = tapply(d5$s.no5, list(d5$NPS_Type, d5$Golf_PL), length)
m5
#m2= m1[,-1]
```

```
barplot(m5, beside = TRUE, col =c ( "pink", "blue"), main="Detractor and promotors based on Golf availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2, lty=1)



#fitness center

regency_b_FC = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_FC$NPS_Type)
View(regency_b_FC)

nrow(regency_b_FC)
regency_b_FC[regency_b_FC==""] <- NA
regency_b_FC[regency_b_FC=="Passive"] <- NA
table(regency_b_FC$NPS_Type)

regency_b_FC <- regency_b_FC[!is.na(regency_b_FC$NPS_Type),]
table(regency_b_FC$NPS_Type)

regency_b_FC$NPS_Type <- factor(regency_b_FC$NPS_Type)
table(regency_b_FC$NPS_Type)
d6<-regency_b_FC[,c("Fitness.Center_PL","NPS_Type")]
View(d6)

d6$NPS_Type <- as.numeric(d6$NPS_Type)
#d6 <- lapply(d6,as.numeric)

d6_df <- data.frame(d6)
View(d6)
d6$s.no6 = seq.int(nrow(d6_df))

m6 = tapply(d6$s.no6, list(d6$NPS_Type, d6$Fitness.Center_PL), length)
m6
#m2= m1[,-1]

barplot(m6, beside = TRUE, col =c ( "pink", "blue"), main= "Detractor and promotors based on Fitness center availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2, lty=1)



#Outdoor Pool

regency_b_OP = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_OP$NPS_Type)
View(regency_b_OP)

nrow(regency_b_OP)
```

```
regency_b_OP[regency_b_OP==""] <- NA
regency_b_OP[regency_b_OP=="Passive"] <- NA
table(regency_b_OP$NPS_Type)

regency_b_OP <- regency_b_OP[!is.na(regency_b_OP$NPS_Type),]
table(regency_b_OP$NPS_Type)

regency_b_OP$NPS_Type <- factor(regency_b_OP$NPS_Type)
table(regency_b_OP$NPS_Type)
d7<-regency_b_OP[,c("Pool.Outdoor_PL","NPS_Type")]
View(d7)
d7$NPS_Type <- as.numeric(d7$NPS_Type)
#d7 <- lapply(d7,as.numeric)

d7_df <- data.frame(d7)

d7$s.no7 = seq.int(nrow(d7_df))

m7 = tapply(d7$s.no7, list(d7$NPS_Type, d7$Pool.Outdoor_PL), length)
m7
#m2= m1[,-1]

barplot(m7, beside = TRUE, col =c ( "pink", "blue"), main="Detractor and promotors
based on Outdoor pool availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2,
lty=1)




#Indoor Pool


regency_b_IP = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]
table(regency_b_IP$NPS_Type)
View(regency_b_IP)

nrow(regency_b_IP)
regency_b_IP[regency_b_IP==""] <- NA
regency_b_IP[regency_b_IP=="Passive"] <- NA
table(regency_b_IP$NPS_Type)

regency_b_IP <- regency_b_IP[!is.na(regency_b_IP$NPS_Type),]
table(regency_b_IP$NPS_Type)

regency_b_IP$NPS_Type <- factor(regency_b_IP$NPS_Type)
table(regency_b_IP$NPS_Type)
View(regency_b_IP)
d8<-regency_b_IP[,c("Pool.Indoor_PL","NPS_Type")]
View(d8)
```

```
d8$NPS_Type <- as.numeric(d8$NPS_Type)
#d8$Pool.Indoor_PL <- as.numeric(d8$Pool.Indoor_PL)

#d8 <- lapply(d8,as.numeric)

d8_df <- data.frame(d8)

d8$s.no8 = seq.int(nrow(d8_df))

m8 = tapply(d8$s.no8,list(d8$NPS_Type, d8$Pool.Indoor_PL), length)
m8

barplot(m8, beside = TRUE, col =c ( "pink", "blue"), main="Detractor and promotors based on Indoor pool availability")
legend('topleft', legend= c("detractor", "promoter"), col = c("pink","blue"), lwd = 2, lty=1)
```

3.4. How does the monthly trend affect the NPS score of Hyatt hotels?

```
#monthly trend plot
install.packages("ggplot2")
library(ggplot2)
cali_regency = readRDS("cali_regency.rds")

#insert a column with sequence
cali_regency$s.no = seq.int(nrow(cali_regency))

#using tapply to group by month and frequency
cali_regency_tapply = tapply(cali_regency$s.no, cali_regency$Month, length)

cali_regency_tapply = as.data.frame(cali_regency_tapply)

cali_regency_tapply$month<-                                        data.frame(rep(NA, length(cali_regency_tapply$cali_regency_tapply)))
View(cali_regency_tapply)
colnames(cali_regency_tapply)=c("value","month")
cali_regency_tapply$month = rownames(cali_regency_tapply)
View(cali_regency_tapply)

rownames(cali_regency_tapply)= NULL

#creating a vector to hold the levels of months
mymonths<- c("Jan","Feb","Mar",
              "Apr","May","Jun",
              "Jul","Aug","Sep",
              "Oct","Nov","Dec")

cali_regency_tapply$month = factor(cali_regency_tapply$month,levels = mymonths)
str(cali_regency_tapply)
View(cali_regency_tapply)
```

```
#plotting the montly trend using ggplot
montly_trend_plot = ggplot(data=cali_regency_tapply,
                           aes(x=cali_regency_tapply$month,
                               y=     cali_regency_tapply$value,     group     =1))
+geom_line(color = "darkblue" )+
  xlab("Months of the year ") +
  ylab("Number of people visiting")

montly_trend_plot

#median user likelihood to recommend by months plot

#performing basic data cleaning
cali_regency$Likelihood_Recommend_H = as.numeric(cali_regency$Likelihood_Recommend_H)
str(cali_regency$mon)
cali_regency_likelihood_by_month  =  cali_regency[,c("Likelihood_Recommend_H","Month",
"s.no")]
View(cali_regency_likelihood_by_month)

#replacing empty cells with NA
cali_regency_likelihood_by_month[ cali_regency_likelihood_by_month == "" ] = NA

#deleting records having NA from the dataset
cali_regency_likelihood_by_month<-na.omit(cali_regency_likelihood_by_month)
cali_regency_likelihood_by_month$s.no                                            =
seq.int(nrow(cali_regency_likelihood_by_month))

#grouping likelihood to recommend by month and calculating mean
cali_regency_likelihood_by_month_1                                               =
tapply(cali_regency_likelihood_by_month$Likelihood_Recommend_H,
cali_regency_likelihood_by_month$Month, mean)

cali_regency_likelihood_by_month_1 = as.data.frame(cali_regency_likelihood_by_month_1)
View(cali_regency_likelihood_by_month_1)
cali_regency_likelihood_by_month_1$month                                         =
rownames(cali_regency_likelihood_by_month_1)

colnames(cali_regency_likelihood_by_month_1)[1] = c("values")

rownames(cali_regency_likelihood_by_month_1)= NULL

#creating a vector to hold the levels of months
mymonths<- c("Jan","Feb","Mar",
             "Apr","May","Jun",
             "Jul","Aug","Sep",
             "Oct","Nov","Dec")

cali_regency_likelihood_by_month_1$month                                         =
factor(cali_regency_likelihood_by_month_1$month,levels = mymonths)
str(cali_regency_tapply)
View(cali_regency_likelihood_by_month_1)
```

```
#plotting using ggplot
montly_trend_plot = ggplot(data=cali_regency_likelihood_by_month_1,
                           aes(x=cali_regency_likelihood_by_month_1$month,
                               y=   cali_regency_likelihood_by_month_1$values,   group
=1)) +geom_line(color = "darkblue" )+
  xlab("Months of the year ") +
  ylab("likelihood to recommend")

montly_trend_plot
```

3.6. ARULES:

```
cali_regency <- readRDS("cali_regency.rds")
View(cali_regency)

install.packages("ggplot2")
library("ggplot2")

install.packages("arules")
library("arules")

install.packages("arulesViz")
library("arulesViz")

regency_business = cali_regency[cali_regency$POV_CODE_C == "BUSINESS",]
b                                                                                    <-
regency_business[,c("Convention_PL","Conference_PL","Business.Center_PL","Dry.Cleanin
g_PL","Fitness.Center_PL","NPS_Type")]
View(b)

b[b==""] <- NA
b[b=="Passive"] <- NA
b<- na.omit(b)

rules_b<-apriori(c, parameter = list(support = 0.001, confidence = 0.1), appearance =
list(rhs = c("NPS_Type=Promoter","NPS_Type=Detractor"), default ="lhs"))

inspect(rules_b)
rules.sorted_b <- sort(rules_b, by="lift")
inspect(rules.sorted_b)

Good.Rules <- rules.sorted_b[quality(rules.sorted_b)$lift>1.2]
inspect(Good.Rules)

plot(Good.Rules)
plot(Good.Rules, method = "graph", control=list(type="items"))
plot(Good.Rules, method = "paracoord", control=list(reorder=TRUE))
plot(Good.Rules, method = "matrix", measure = "lift", control=list(reorder=TRUE))
plot(Good.Rules, method = "matrix3D", measure = "lift", control=list(reorder=TRUE))
```

```
regency_liesure = cali_regency[cali_regency$POV_CODE_C == "LEISURE",]

l[l==""] <- NA
l[l=="Passive"] <- NA
#regency_liesure <- regency_liesure[!is.na(regency_liesure$NPS_Type),]
l<-na.omit(l)

l<-
regency_liesure[,c("Casino_PL","Spa_PL","Boutique_PL","Golf_PL","Pool.Outdoor_PL","NP
S_Type")]
View(l)

rules_l<-apriori(l,parameter = list(support = 0.03, confidence = 0.08), appearance =
list(rhs = c("NPS_Type=Promoter","NPS_Type=Detractor"), default ="lhs"))
inspect(rules_l)
rules.sorted_l<- sort(rules_l, by="lift")
inspect(rules.sorted_l)

Good.Rules_l <- rules.sorted_l[quality(rules.sorted_l)$lift>1.006]
inspect(Good.Rules_l)

plot(Good.Rules_l)
plot(Good.Rules_l, method = "graph", control=list(type="items"))
plot(Good.Rules_l, method = "paracoord", control=list(reorder=TRUE))
plot(Good.Rules_l, method = "matrix", measure = "lift", control=list(reorder=TRUE))
plot(Good.Rules_l, method = "matrix3D", measure = "lift", control=list(reorder=TRUE))
```

4.1 Linear Modelling:

```
#Linear modeling


  #loading the dataset
  data1 <-readRDS("cali_regency.rds")

  #View(datag)
  #summary(datag)

  #considering only the folllowing columns in the dataset
  data1  <-  data1[,which(names(data1)  %in%  c("Likelihood_Recommend_H","Boutique_PL",
"Business.Center_PL", "Casino_PL",
                                            "Conference_PL",         "Convention_PL",
"Dry.Cleaning_PL",
                                            "Elevators_PL",      "Fitness.Center_PL",
"Fitness.Trainer_PL",

"Golf_PL","Indoor.Corridors_PL","Laundry_PL", "Limo.Service_PL",
                                            "Mini.Bar_PL","Pool.Indoor_PL",
"Pool.Outdoor_PL",
```

```
"Regency.Grand.Club_PL","Resort_PL","Restaurant_PL","POV_CODE_C"))]

  data1$Likelihood_Recommend_H[is.na(data1$Likelihood_Recommend_H)]               =
mean(data1$Likelihood_Recommend_H, na.rm = TRUE)
  data1$Likelihood_Recommend_H = round(data1$Likelihood_Recommend_H)
  View(data1)
  summary(data1)


  #data1$All.Suites_PL <- factor(data1$All.Suites_PL)

  #changing all "" to NA
  data1[ data1 == "" ] = NA
  #deleting records having NA from the dataset
  data1<-na.omit(data1)
  summary(data1)

  #dividing data into test and train
  require(caTools)
  set.seed(101)
  sample = sample.split(data1$Likelihood_Recommend_H, SplitRatio = 0.7)
  train = subset(data1, sample == TRUE)
  test  = subset(data1, sample == FALSE)
  View((train))
  View(test)



  nrow(data1)
  data1$Likelihood_Recommend_H = as.numeric(data1$Likelihood_Recommend_H)
  table(data1$POV_CODE_C)

  #considering only business users
  data1 = data1[data1$POV_CODE_C == "BUSINESS",]

  #### Now creating a linear model between likelihood to recommend and facilities that
matter to business users
  linearModel_1 <- lm(formula = data1$Likelihood_Recommend_H~data1$Business.Center_PL
+data1$Convention_PL, data = data1)


  # Checking the R-squared value for this model
  summary(linearModel_1) #R-Squared -  0.00054

  data1 = data1[data1$POV_CODE_C == "LEISURE",]
  #### Now creating a linear model between likelihood to recommend and  facilities
that matter to leisure users
  linearModel_2                   <-                   lm(formula                   =
data1$Likelihood_Recommend_H~data1$Pool.Indoor_PL+data1$Pool.Outdoor_PL, data = data1)
```

```
# Checking the R-squared value for this model
summary(linearModel_2) #R-Squared - 0.0002698


#### Now creating a linear model between likelihood to recommend and all features
linearModel_3 <- lm(formula = data1$Likelihood_Recommend_H~ ., data = data1)

# Checking the R-squared value for this model
summary(linearModel_3) #R-Squared - 0.002405
```

4.2. Logistic Regression:

```
#logistic regression

install.packages('kernlab')
library(kernlab)

install.packages("e1071")
library(e1071)

install.packages("SDMTools")
library(SDMTools)

library(ROCR)

SVM.Data <- readRDS("cali_regency.rds")
#View(SVM.Data)

#dividing users by purpose of visit

table(SVM.Data$POV_CODE_C)
svm_business = SVM.Data[SVM.Data$POV_CODE_C == "BUSINESS",]
svm_leisure = SVM.Data[SVM.Data$POV_CODE_C == "LEISURE",]

#considering only the folllowing columns in the dataset
svm_business <- svm_business[,which(names(svm_business) %in%
c("NPS_Type","Likelihood_Recommend_H","Boutique_PL", "Business.Center_PL",
"Casino_PL",
                                                              "Conference_PL",
"Convention_PL", "Dry.Cleaning_PL",
                                                              "Elevators_PL",
"Fitness.Center_PL", "Fitness.Trainer_PL",

"Golf_PL","Indoor.Corridors_PL","Laundry_PL", "Limo.Service_PL",

"Mini.Bar_PL","Pool.Indoor_PL", "Pool.Outdoor_PL",

"Regency.Grand.Club_PL","Resort_PL","Restaurant_PL","POV_CODE_C","Limo.Service_PL","V
alet.Parking_PL"))]
```

```
svm_leisure          <-          svm_leisure[,which(names(svm_leisure)          %in%
c("NPS_Type","Likelihood_Recommend_H","Boutique_PL",            "Business.Center_PL",
"Casino_PL",
                                                              "Conference_PL",
"Convention_PL", "Dry.Cleaning_PL",
                                                              "Elevators_PL",
"Fitness.Center_PL", "Fitness.Trainer_PL",

"Golf_PL","Indoor.Corridors_PL","Laundry_PL", "Limo.Service_PL",

"Mini.Bar_PL","Pool.Indoor_PL", "Pool.Outdoor_PL",

"Regency.Grand.Club_PL","Resort_PL","Restaurant_PL","POV_CODE_C","Limo.Service_PL","V
alet.Parking_PL"))]

nrow(svm_business) # 86k entries
#View(SVM.Data)

#changing all "" to NA

svm_business[ svm_business == "" ] = NA

#deleting records having NA from the dataset
svm_business<-na.omit(svm_business)

#converting NPS type to numeric factors
svm_business$NPS_Type <- gsub("Promoter", "Yes", svm_business$NPS_Type)
svm_business$NPS_Type <- gsub("Detractor", "No", svm_business$NPS_Type)
svm_business$NPS_Type <- gsub("Passive", "Yes", svm_business$NPS_Type)
svm_business$NPS_Type <- as.factor(svm_business$NPS_Type)
table(svm_business$NPS_Type)
svm_business$NPS_Type = as.factor(svm_business$NPS_Type )
svm_business$NPS_Type = as.numeric(svm_business$NPS_Type )

table(svm_business$NPS_Type )

svm_business <- as.data.frame(lapply(svm_business, factor))
svm_business <- as.data.frame(sapply(svm_business,  as.numeric))
sapply(svm_business, levels)

head(svm_business)

#dividing data into test and train data
require(caTools)
set.seed(101)
sample = sample.split(svm_business$NPS_Type, SplitRatio = .70)
train = subset(svm_business, sample == TRUE)
test  = subset(svm_business, sample == FALSE)
View((train))
View(test)
```

```
#### Now creating a logistic model between likelihood to recommend and facilities that
matter to business users
logistic_A <- glm(NPS_Type~Business.Center_PL+Convention_PL+Limo.Service_PL
                   +Valet.Parking_PL, data = train, family = 'binomial' )

table(svm_business$Limo.Service_PL)

svm_business$Valet.Parking_PL = svm_business$Valet.Parking_PL-1
svm_business$NPS_Type = svm_business$NPS_Type-1
svm_business$Business.Center_PL = svm_business$Business.Center_PL-1
svm_business$Convention_PL = svm_business$Convention_PL-1
svm_business$Limo.Service_PL = svm_business$Limo.Service_PL-1

summary(logistic_A)

# predicting the model
Predict.A <- predict(logistic_A, test, type= 'response')

head(as.data.frame(Predict.A))
unique(Predict.A)
unique(test$NPS_Type)

#analysing AUC
Predict.A <- predict(SVM.Model.A, test, type= 'response')
pred <- prediction(Predict.A, test$NPS_Type)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc #0.5352322

#### Now creating a logistic model between likelihood to recommend and facilities that
matter to leisure users

svm_leisure[ svm_leisure == "" ] = NA

#deleting records having NA from the dataset
svm_leisure<-na.omit(svm_leisure)

#converting NPS type to numeric factors
svm_leisure$NPS_Type <- gsub("Promoter", "Yes", svm_leisure$NPS_Type)
svm_leisure$NPS_Type <- gsub("Detractor", "No", svm_leisure$NPS_Type)
svm_leisure$NPS_Type <- gsub("Passive", "Yes", svm_leisure$NPS_Type)
svm_leisure$NPS_Type <- as.factor(svm_leisure$NPS_Type)
table(svm_leisure$NPS_Type)
svm_leisure$NPS_Type = as.factor(svm_leisure$NPS_Type )
svm_leisure$NPS_Type = as.numeric(svm_leisure$NPS_Type )

table(svm_leisure$NPS_Type )

svm_leisure <- as.data.frame(lapply(svm_leisure, factor))
```

```
svm_leisure <- as.data.frame(sapply(svm_leisure,  as.numeric))
svm_leisure$Pool.Indoor_PL = svm_leisure$Pool.Indoor_PL-1
svm_leisure$Pool.Outdoor_PL = svm_leisure$Pool.Outdoor_PL-1
svm_leisure$Fitness.Center_PL = svm_leisure$Fitness.Center_PL-1
svm_leisure$Casino_PL = svm_leisure$Casino_PL-1
svm_leisure$NPS_Type = svm_leisure$NPS_Type-1

sapply(svm_leisure, levels)

#dividing data into test and train data
require(caTools)
set.seed(101)
sample = sample.split(svm_leisure$NPS_Type, SplitRatio = .70)
train = subset(svm_leisure, sample == TRUE)
test  = subset(svm_leisure, sample == FALSE)
View((train))
View(test)

table(svm_leisure$NPS_Type)

#performing logistic regression
logistic_B <- glm(NPS_Type~Pool.Indoor_PL+Pool.Outdoor_PL+Fitness.Center_PL+Casino_PL
                  , data = train, family = 'binomial')

#predicting on the test data
Predict.B <- predict(logistic_B, test, type= 'response')

head(as.data.frame(Predict.B))
unique(Predict.B)
unique(test$NPS_Type)

#Plotting the AUC
pred <- prediction(Predict.B, test$NPS_Type)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc #0.5146511
```

5. K- Nearest Neighbors:

```
#features on which the data is to be trained
usdata.mini  <-  subset(usdata,  select  =  c("CHECKOUT_HEADER_ID_C","All.Suites_PL",
"Bell.Staff_PL","Boutique_PL",  "Business.Center_PL",  "Casino_PL",  "Conference_PL",
"Convention_PL",     "Dry.Cleaning_PL",     "Elevators_PL",     "Fitness.Center_PL",
"Fitness.Trainer_PL","Golf_PL","Indoor.Corridors_PL","Laundry_PL", "Limo.Service_PL",
"Mini.Bar_PL","Pool.Indoor_PL",    "Pool.Outdoor_PL",    "Resort_PL","Restaurant_PL",
"Self.Parking_PL",   "Shuttle.Service_PL",   "Ski_PL",   "Spa_PL","Valet.Parking_PL",
"NUMBER_OF_ROOMS_C", "Length_Stay_H", "ADULT_NUM_C", "WALK_IN_FLG_C", "POV_CODE_C",
"RESERVATION_STATUS_R", "ROOM_TYPE_CODE_R", "ADULT_NUM_R", "Likelihood_Recommend_H"))
```

```
#check levels of data
sapply(usdata.mini, levels)

#removing NAs
usdata.mini[usdata.mini ==""] <- NA

usdata.mini <- na.omit(usdata.mini)

#Factorizing and then giving numerical values to the categorical values present in
dataframe
usdata.mini <- as.data.frame(lapply(usdata.mini, factor))
usdata.mini <- sapply(usdata.mini, as.numeric)
usdata.mini <- as.data.frame(usdata.mini)

#function to find the Euclidian distance between two points
euc.dist <- function(x1, x2) sqrt(sum((x1 - x2) ^ 2))

#selecting a random row from the dataset
row1 <- usdata.mini[which(usdata.mini$CHECKOUT_HEADER_ID_C == 54857),]
#creating the n-dimensional vector
vec1          <-          as.numeric(row1[,          !(colnames(row1)          %in%
c("CHECKOUT_HEADER_ID_C","Likelihood_Recommend_H"))])

#see the vector
vec1

#creating an empty data frame to hold the distance values
distance <- data.frame(dist = rep(NA, length(usdata.mini)))

#loop in which distance between selected point and rest of the points is calculated
for( i in 1 : nrow(usdata.mini))
{
  row <- usdata.mini[i,1]
  row2 <- usdata.mini[which(usdata.mini$CHECKOUT_HEADER_ID_C == row),]
  vec2          <-          as.numeric(row2[,          !(colnames(row2)          %in%
c("CHECKOUT_HEADER_ID_C","Likelihood_Recommend_H"))])

  distance[i,1] = euc.dist(vec1, vec2) #holds the distance
  distance[i,2] = usdata.mini[i,1] #holds the index for comparison later on
  print(i) #to know how much of the data has been processed
}

#ordering the distance dataframe
distance1 <- distance[order(distance$dist),]

#calculation of net NPS
# selected k = 100 in this case, can be changed
nps=0
for(i in 1 : 100)
{
```

```
  nps = nps + usdata.mini$Likelihood_Recommend_H[usdata.mini$CHECKOUT_HEADER_ID_C ==
distance1[i,2]]
}

#variable to store the calculated nps
calculated_nps <- nps/100

#actual nps score
actual_score <- row1$Likelihood_Recommend_H
```