ANALYSIS OF HYATT REGENCY CUSTOMER DATA

GUIDED BY:
PROF. JEFF SALTZ
PROF. ERIK ANDERSON
MR. IVAN SHAMSHURIN

PRESENTED BY:
ADITYA CHAUHAN
AKSHIT SALIAN
NISHITA SHETH
QIYI WU
ZHAOHENG LI

PRESENTED FOR:
IST 687-APPLIED DATA SCIENCE

Project Objective

The objective of this project is to analyze the data provided by the customers, who have visited Hyatt Regency hotels and provided feedback for their services and facilities, in order to improve customer satisfaction by providing better facilities and thereby increasing their revenue.

Project Scope

The scope of the project is limited to the chain of Hyatt Regency hotel which are located within the United States of America. For analysis we have produced a subset from existing dataset which consists of data from alternate months for the year 2014.

Deliverables

1. Identifying Key Performance Indicators (KPI) in Hyatt Regency hotels and distributing the relevance in R-programming to identify patterns.
2. Performing data munging to form clusters of data.
3. Performing data analytics on information calculate the Net Promoter Score (NPS) and represent the data in form of graphs using data visualization tools.

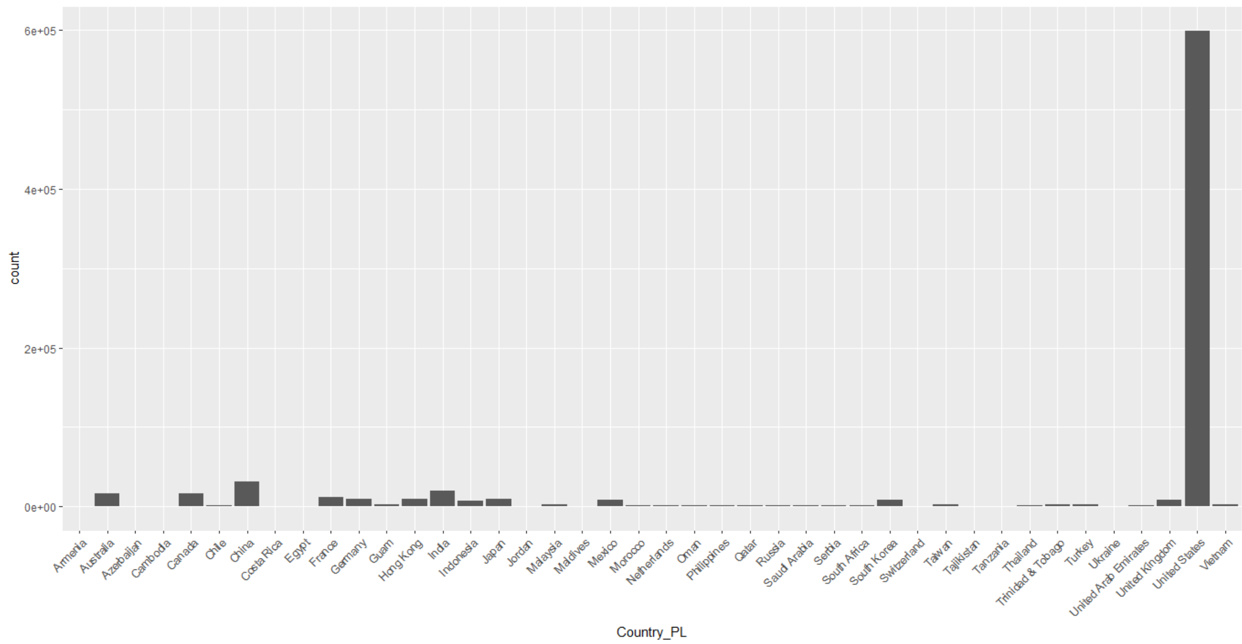
Data Requisition

Initially, we started with 55 attributes and 6 Lakh records for each alternate month, starting from February which sums up to 360,000 records. However, after munging the data set we designed a final data set which consisted of 24167 obs. With 47 variables:

Data Preprocessing

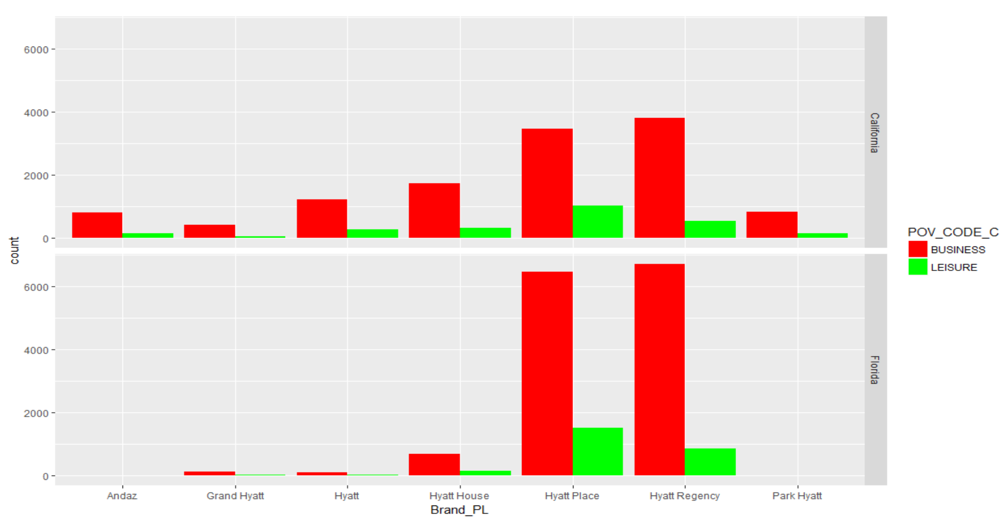
The initial data set has 360,000 instances of customer data and 55 columns describing various factors related to customers and hotels. However, there were many records where the "Likelihood to Recommend" column had null values. After eliminating these records, the data set then consisted of 47,300 records and 55 attributes. We later focused on the top states which contributed to the maximum portion of the data set i.e. Florida and California, which created our final data set consisted of 24167 records and 47 attributes.

We checked the dataset of February Month and figured out that about 90 % of the data was of United States. Hence, we proceeded to use United States data for the rest of the months as well and shortlist data of California and Florida



We determined the count of customers who visited hotels in Florida and California (Hotel Wise)

```
library(ggplot2)

HotelPlot <- ggplot(ModellingData, aes(x=Brand_PL)) + geom_bar(aes(fill=POV_CODE_C), position="dodge") + facet_grid(State_PL ~ .) + scale_fill_manual(values=c("red", "green"))

HotelPlot
```



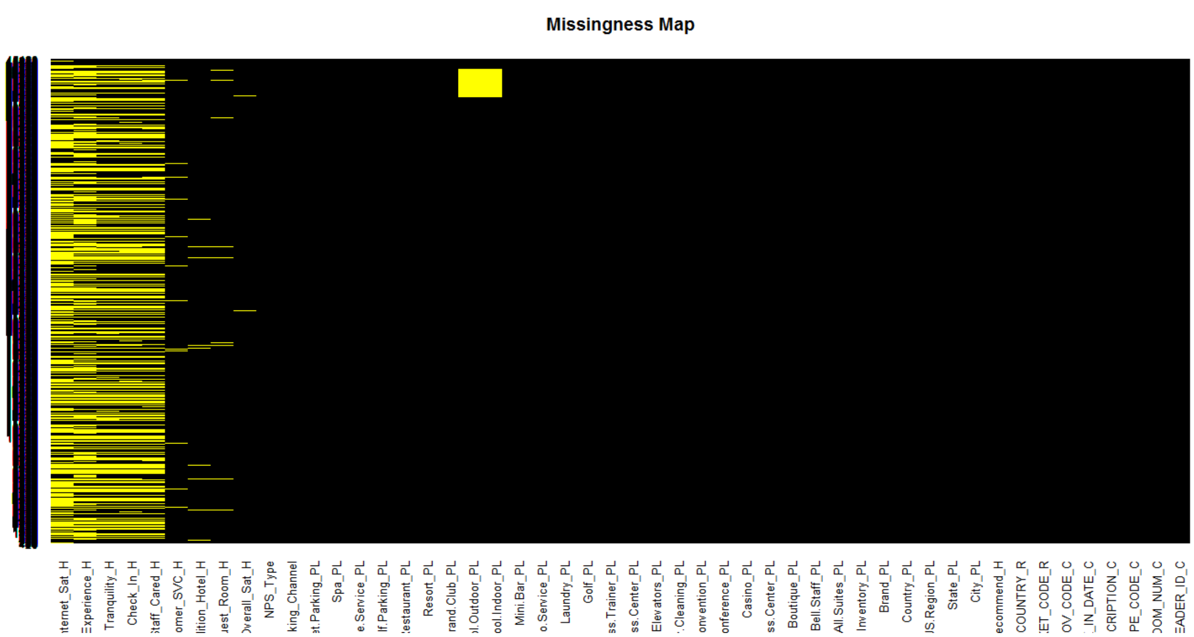
Business Questions

Feedback is provided by each and every customer who resides at one of the Hyatt Regency. This feedback consisted of various parameters such as likelihood to recommend, overall satisfaction, tranquility, customer service, overall experience, guestroom, etc.
Analyzing this data based on their feedback is crucial to Hyatt Regency to improve their services before it becomes and major issue.
We have used various descriptive statistics, modelling techniques and visualization techniques to address the following the business questions:

1. What is the count of customers who visited hotels in Florida and California (hotel wise)?
2. State the hotel with maximum promoters and detractors.
3. What are the various services and parameters affect the promoters and detractors?
4. Which are the top two cities within the top two states with maximum number of customers?
5. Which are the top two cities within the top two states with maximum number of promoters and detractors?
6. In which areas (overall satisfaction, tranquility, guestroom, customer service, etc.) other Hyatt hotels lag with respect to the best Hyatt hotels?
7. Calculate hotel wise Net Promoter Score (NPS) for each hotel within the city of California and Florida.

Data Cleaning

In order to clean the data and get rid of the NA values we initially checked for NA values using the missingness map created using "Amelia" Package in R



In order to get rid of the NA values, we initially checked for percentage of NA values in the respective columns. If the percentage of NA value is greater than 55%, then we have decided to omit that column, else we have replaced the NA values with the mean value of the column.

```
# for Guest_Room_H
CountOfNA_GuestRoom <- sum(is.na(HotelData$Guest_Room_H))
CountOfNA_GuestRoom
NaGuestRoom_H <- (CountOfNA_GuestRoom/nrow(HotelData)) * 100
NaGuestRoom_H

# Percentage of NA Value : 18.36454
# for Tranquility_H
CountOfNA_Tranquility_H<- sum(is.na(HotelData$Tranquility_H))
CountOfNA_Tranquility_H
NaTranquility_H <- (CountOfNA_Tranquility_H/nrow(HotelData)) * 100
NaTranquility_H
# Percentage of NA Value : 54.61132

#Condition_Hotel_H
CountOfNA_Condition_Hotel_H<- sum(is.na(HotelData$Condition_Hotel_H))
CountOfNA_Condition_Hotel_H
NaCondition_Hotel_H <- (CountOfNA_Condition_Hotel_H/nrow(HotelData)) * 100
NaCondition_Hotel_H
#Percentage of NA Value : 18.6046

#Customer_SVC_H
CountOfNA_Customer_SVC_H<- sum(is.na(HotelData$Customer_SVC_H))
CountOfNA_Customer_SVC_H
NaCustomer_SVC_H <- (CountOfNA_Customer_SVC_H/nrow(HotelData)) * 100
NaCustomer_SVC_H
#Percentage of NA Value : 18.9516
```

```
#Staff_Cared_H
CountOfNA_Staff_Cared_H<- sum(is.na(HotelData$Staff_Cared_H))
CountOfNA_Staff_Cared_H
NaStaff_Cared_H <- (CountOfNA_Staff_Cared_H/nrow(HotelData)) * 100
NaStaff_Cared_H

##Percentage of NA Value : 54.45637

#Internet_Sat_H

CountOfNA_Internet_Sat_H<- sum(is.na(HotelData$Internet_Sat_H))
CountOfNA_Internet_Sat_H
NaInternet_Sat_H<- (CountOfNA_Internet_Sat_H/nrow(HotelData)) * 100
NaInternet_Sat_H

###Percentage of NA Value : 66.90018

#Check_In_H
CountOfNA_Check_In_H<- sum(is.na(HotelData$Check_In_H))
CountOfNA_Check_In_H
NaCheck_In_H<- (CountOfNA_Check_In_H/nrow(HotelData)) * 100
NaCheck_In_H

##Percentage of NA Value : 54.4782

#F.B_Overall_Experience_H
CountOfNA_F.B_Overall_Experience_H<- sum(is.na(HotelData$F.B_Overall_Experience_H))
CountOfNA_F.B_Overall_Experience_H
NaF.B_Overall_Experience_H<- (CountOfNA_F.B_Overall_Experience_H/nrow(HotelData)) * 100
NaF.B_Overall_Experience_H
```

```
##Percentage of NA Value : 63.7183

#Overall_Sat_H
CountofNA_Overall_Sat_H<- sum(is.na(HotelData$Overall_Sat_H))
CountofNA_Overall_Sat_H
NaOverall_Sat_H<- (CountofNA_Overall_Sat_H/nrow(HotelData)) * 100
NaOverall_Sat_H

##Percentage of NA Value : 17.58107
```

We shall be removing F.B_Overall_Experience_H and #Internet_Sat_H

Also, there were some NULL Values in the categorical columns where the answer is given in Y or No. To solve that problem, we initially replaced the NULL values with NA values we followed the following steps:

- ✓ Exported the dataset to the excel file
- ✓ Called the exported file again in R and passed na.strings=c("","NA")) as the parameter while reading the file

After that we checked the percentage of NA value in each column using the same as above. Below is the data frame of the Percentage of NA Values in the columns

```
> naCategorical
                Columns PercentageOFnaVAlues
1          All.Suites_PL           0.3382655
2           Bell.Staff_PL          29.8699315
3             Boutique_PL           0.3382655
4      Business.Center_PL           0.3382655
5               Casino_PL           0.3382655
6           Conference_PL           0.3382655
7           Convention_PL           0.3382655
8         Dry.Cleaning_PL          29.8699315
9            Elevators_PL          29.8699315
10      Fitness.Center_PL          29.8699315
11     Fitness.Trainer_PL          31.3823054
12                Golf_PL           0.3382655
13             Laundry_PL          29.8699315
14        Limo.Service_PL          29.8699315
15            Mini.Bar_PL          29.8699315
16         Pool.Indoor_PL          29.8699315
17        Pool.Outdoor_PL          29.8699315
18 Regency.Grand.Club_PL          29.8699315
19              Resort_PL           0.3382655
20          Restaurant_PL           0.3382655
21        Self.Parking_PL          29.8699315
22     Shuttle.Service_PL          29.8699315
23                 Spa_PL           0.3382655
24        Valet.Parking_PL          29.8699315
```

Further, we decided to remove the column with 31 % of NA values and omit the rest

Also there were some NA values in NPS_Type as well whose Likelihood to recommend was 8.686550976. We were not able to decide whether to consider it as Promoter or Passive. Hence, we decided to omit these columns.

Initial Phase

The team decided to project "Likelihood to Recommend" column as the independent attribute, which contributes to improving the business of the hotel. Therefore, we decided to calculate the Net Promoter Score (NPS) for "Likelihood to Recommend".

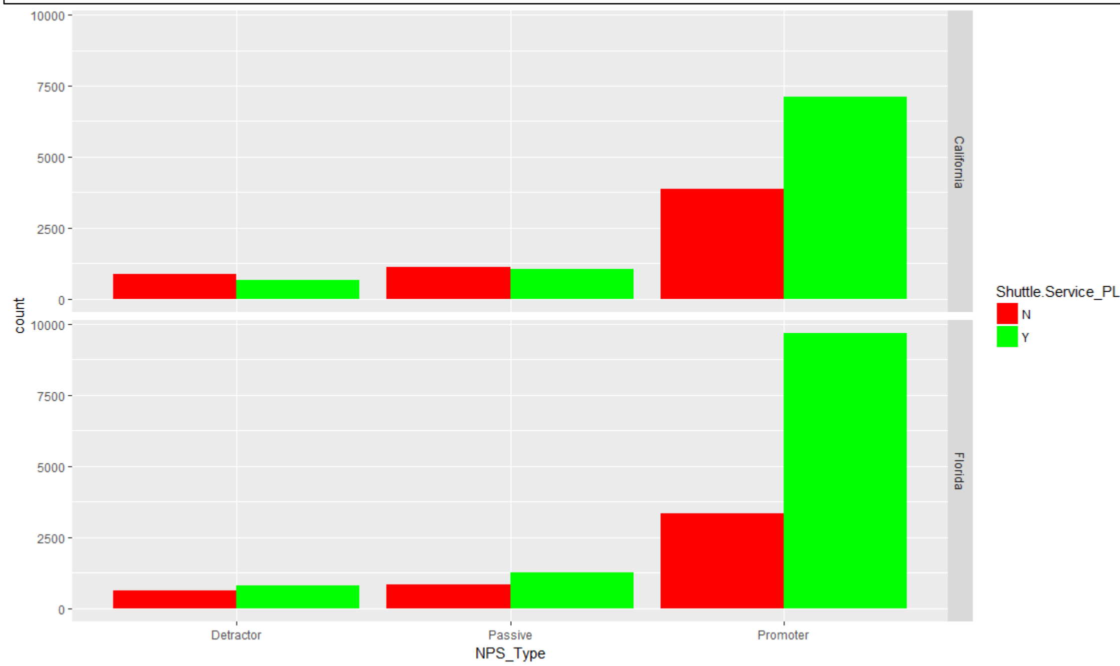
The analysis is carried considering the following the assumptions:

1. Rating of 9 or 10: Promoters
2. Rating of 7 or 8: Passive
3. Rating of 6 or below: Detractors

We have identified various parameters which could the possible reasons for customers being either detractors or promoters.

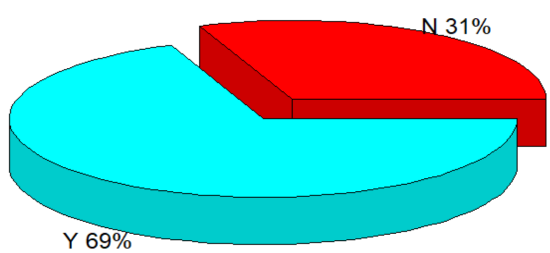
How does having Shuttle.Service_PL effect the promoter and detractor

```
HotelShuttle.Service_PL <- ggplot(ModellingData, aes(x=NPS_Type)) +
geom_bar(aes(fill=Shuttle.Service_PL), position="dodge") + facet_grid(State_PL ~ .) +
scale_fill_manual(values=c("red", "green","blue"))

HotelShuttle.Service_PL
```



Following code represents a pie chart that compares the count of promoters whose purpose of visit was leisure and hotels that do not have a shuttle service to the count of promoters whose purpose of visit was leisure and the hotel that do have a shuttle service.

```
Shuttle.Service_PL_n <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H) > 8 ) &
((ModellingData$POV_CODE_C)=="BUSINESS") &   ((ModellingData$Shuttle.Service_PL)=="N")])

Shuttle.Service_PL_y <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H) > 8 )  &
((ModellingData$POV_CODE_C)=="BUSINESS") & ((ModellingData$Shuttle.Service_PL)=="Y")])

piesShuttle.Service_PL <- c(Shuttle.Service_PL_n,Shuttle.Service_PL_y)

labels <- c("N","Y")

par(mar = rep(2, 4))

pctpiesShuttle.Service_PL <- round(piesShuttle.Service_PL/sum(piesShuttle.Service_PL)*100)

lblspiesShuttle.Service_PL <- paste(labels, pctpiesShuttle.Service_PL) # add percents to labels

lblspiesShuttle.Service_PL <- paste(lblspiesShuttle.Service_PL,"%",sep="") # ad % to labels

pie3D(piesShuttle.Service_PL, labels=lblspiesShuttle.Service_PL,explode=0.1, col =
rainbow(length(piesShuttle.Service_PL)))
```

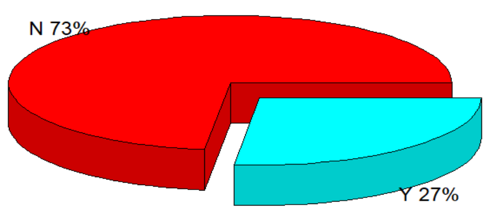


69% of the people who have visited the hotel for leisure were promoters because the hotel had shuttle service.

Similarly, we identified a few more patterns with different parameters:

1. Comparing the count of detractors whose purpose of visit was leisure and hotels that do not have a spa to the count of detractors whose purpose of visit was leisure and the hotel that do have a spa.

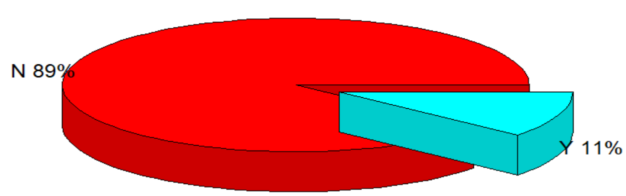
```
spa_n <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H)
< 8 ) & ((ModellingData$POV_CODE_C)=="LEISURE") & ((ModellingData$Spa_PL)=="N")])

spa_y <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H)
< 8 ) & ((ModellingData$POV_CODE_C)=="LEISURE") & ((ModellingData$Spa_PL)=="Y")])

pies <- c(spa_n,spa_y)

labels <- c("N","Y")

par(mar = rep(2, 4))

pct <- round(pies/sum(pies)*100)

lbls <- paste(labels, pct) # add percents to labels

lbls <- paste(lbls,"%",sep="") # ad % to labels

pie3D(pies, labels=lbls,explode=0.1, col = rainbow(length(pies)))
```



73% of the people who have visited the hotel for leisure were detractors because the hotel did not have spa service.

2. Comparing the count of detractors whose purpose of visit was leisure and hotels that do not have a mini bar to the count of detractors whose purpose of visit was leisure and the hotel that do have a mini bar.

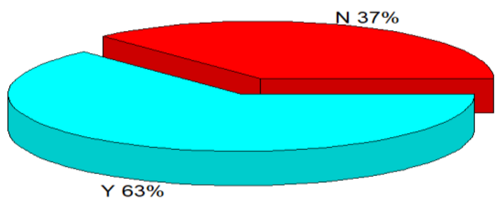
```
Mini.Bar_PL_n <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H) < 8 )
& ((ModellingData$POV_CODE_C)=="LEISURE") &   ((ModellingData$Mini.Bar_PL)=="N")])

Mini.Bar_PL_y <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H) < 8 )
& ((ModellingData$POV_CODE_C)=="LEISURE") & ((ModellingData$Mini.Bar_PL)=="Y")])

piesMini.Bar_PL <- c(Mini.Bar_PL_n,Mini.Bar_PL_y)

labels <- c("N","Y")

par(mar = rep(2, 4))

pctpiesMini.Bar_PL <- round(piesMini.Bar_PL/sum(piesMini.Bar_PL)*100)

lblsMini.Bar_PL <- paste(labels, pctpiesMini.Bar_PL) # add percents to labels

lblsMini.Bar_PL <- paste(lblsMini.Bar_PL,"%",sep="") # ad % to labels

pie3D(piesMini.Bar_PL, labels=lblsMini.Bar_PL,explode=0.1, col = rainbow(length(piesMini.Bar_PL)))
```



89% of the people who have visited the hotel for leisure were detractors because the hotel did not have a mini bar.

3. Comparing the count of promoters whose purpose of visit was leisure and hotels that do not have a valet parking to the count of promoters whose purpose of visit was leisure and the hotel that do have a valet parking.

```
Valet.Parking_PL_n <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H) > 8 )
&   ((ModellingData$Valet.Parking_PL)=="N")])

Valet.Parking_PL_y <-
length(ModellingData$Likelihood_Recommend_H[((ModellingData$Likelihood_Recommend_H) > 8 )
& ((ModellingData$Valet.Parking_PL)=="Y")])

piesValet.Parking_PL <- c(Valet.Parking_PL_n,Valet.Parking_PL_y)

labels <- c("N","Y")

par(mar = rep(2, 4))

pctValet.Parking_PL <- round(piesValet.Parking_PL/sum(piesValet.Parking_PL)*100)

lblsValet.Parking_PL <- paste(labels, pctValet.Parking_PL) # add percents to labels

lblsValet.Parking_PL <- paste(lblsValet.Parking_PL,"%",sep="") # ad % to labels

pie3D(piesValet.Parking_PL, labels=lblsValet.Parking_PL,explode=0.1, col =
rainbow(length(piesValet.Parking_PL)))
```



63% of the people who have visited the hotel for leisure were promoters because the hotel had valet parking.

Modelling

We decided to move to a model based approach to carry out analysis of data. The various models used were:

1. Linear Regression
2. Support Vector Machine (SVM)
3. K Support Vector Machine (KSVM)
4. Associative Rule Mining

Points covered under this section are:

1. The modelling is based on various parameter such as overall satisfaction, guestroom, tranquility against the independent parameter, likelihood to recommend for determining the r-square value and the accuracy of these parameters.
2. Comparing the factors of the best hotel against the same parameters of other hotels.

Following code represents the points made above and recommends the various facilities that need to be improved in order to improve customer satisfaction and profitability.

```
### We shall be using Likelihood to recommend as it is a numeric field

## Likelihood_Recommend_H vs Overall_Sat_H, Guest_Room_H,  Tranquility_H

LikeVsOverallGuestTranquility <- lm(Likelihood_Recommend_H ~ Overall_Sat_H + Guest_Room_H
+Tranquility_H,ModellingData)

summary(LikeVsOverallGuestTranquility)
```

```
> summary(LikeVsOverallGuestTranquility)

Call:
lm(formula = Likelihood_Recommend_H ~ Overall_Sat_H + Guest_Room_H +
    Tranquility_H, data = ModellingData)

Residuals:
    Min      1Q  Median      3Q     Max
-8.0552 -0.0944 -0.0552  0.0977  8.2031

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.594753   0.038704 -15.367  < 2e-16 ***
Overall_Sat_H  0.903575   0.004703 192.137  < 2e-16 ***
Guest_Room_H   0.132836   0.004777  27.806  < 2e-16 ***
Tranquility_H  0.032509   0.004763   6.825 8.99e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8616 on 24163 degrees of freedom
Multiple R-squared:  0.817,     Adjusted R-squared:  0.8169
F-statistic: 3.595e+04 on 3 and 24163 DF,  p-value: < 2.2e-16
```

```
##Multiple R-squared:  0.817,    Adjusted R-squared:  0.8169

## Removing Overall_Sat_H

LikeVsGuestTranquility <- lm(Likelihood_Recommend_H ~ Guest_Room_H
+Tranquility_H,ModellingData)

summary(LikeVsGuestTranquility)
```

```
> summary(LikeVsGuestTranquility)

Call:
lm(formula = Likelihood_Recommend_H ~ Guest_Room_H + Tranquility_H,
    data = ModellingData)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8448 -0.1227  0.1552  0.3746  8.5550

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.511664   0.060849   8.409   <2e-16 ***
Guest_Room_H  0.751355   0.005612 133.890   <2e-16 ***
Tranquility_H 0.181961   0.007471  24.355   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.37 on 24164 degrees of freedom
Multiple R-squared:  0.5373,    Adjusted R-squared:  0.5373
F-statistic: 1.403e+04 on 2 and 24164 DF,  p-value: < 2.2e-16
```

#Multiple R-squared: 0.5373,   Adjusted R-squared: 0.5373

#Likelihood_Recommend_H vs Guest_Room_H, Tranquility_H, Condition_Hotel_H

LikeVsGuestTranquilityCondition <- lm(Likelihood_Recommend_H ~ Guest_Room_H +Tranquility_H + Condition_Hotel_H,ModellingData)

summary(LikeVsGuestTranquilityCondition)

```
> summary(LikeVsGuestTranquilityCondition)

Call:
lm(formula = Likelihood_Recommend_H ~ Guest_Room_H + Tranquility_H +
    Condition_Hotel_H, data = ModellingData)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9278  0.0137  0.1387  0.4646  5.9354

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.737452   0.060724  -12.14   <2e-16 ***
Guest_Room_H       0.450979   0.007317   61.63   <2e-16 ***
Tranquility_H      0.143855   0.007016   20.50   <2e-16 ***
Condition_Hotel_H  0.471695   0.008007   58.91   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.281 on 24163 degrees of freedom
Multiple R-squared:  0.5954,    Adjusted R-squared:  0.5954
F-statistic: 1.185e+04 on 3 and 24163 DF,  p-value: < 2.2e-16
```

#Multiple R-squared: 0.5954,   Adjusted R-squared: 0.5954

#Guest_Room_H +Tranquility_H + Customer_SVC_H

LikeVsGuestTranquilityCustomer <- lm(Likelihood_Recommend_H ~ Guest_Room_H +Tranquility_H + Customer_SVC_H + Condition_Hotel_H,ModellingData)

summary(LikeVsGuestTranquilityCustomer)

```
> summary(LikeVsGuestTranquilityCustomer)

Call:
lm(formula = Likelihood_Recommend_H ~ Guest_Room_H + Tranquility_H +
    Customer_SVC_H + Condition_Hotel_H, data = ModellingData)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9988 -0.0843  0.1294  0.4487  5.3332

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.104143   0.056925  -36.96   <2e-16 ***
Guest_Room_H       0.348460   0.006660   52.32   <2e-16 ***
Tranquility_H      0.106304   0.006281   16.93   <2e-16 ***
Customer_SVC_H     0.459788   0.005854   78.54   <2e-16 ***
Condition_Hotel_H  0.295745   0.007490   39.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.143 on 24162 degrees of freedom
Multiple R-squared:  0.6777,	Adjusted R-squared:  0.6777
F-statistic: 1.27e+04 on 4 and 24162 DF,  p-value: < 2.2e-16
```

#Multiple R-squared: 0.6777,   Adjusted R-squared: 0.6777

Customer Service somewhat contributes to NPS

#Adding Staff_Cared_H

Model3AndStaff <- lm(Likelihood_Recommend_H ~ Guest_Room_H +Tranquility_H + Customer_SVC_H + Staff_Cared_H + Condition_Hotel_H,ModellingData)

summary(Model3AndStaff)

```
> summary(Model3AndStaff)

Call:
lm(formula = Likelihood_Recommend_H ~ Guest_Room_H + Tranquility_H +
    Customer_SVC_H + Staff_Cared_H + Condition_Hotel_H, data = ModellingData)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0444 -0.0915  0.1607  0.4622  6.0250

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.550128   0.063817  -39.96   <2e-16 ***
Guest_Room_H       0.354361   0.006640   53.37   <2e-16 ***
Tranquility_H      0.068482   0.006729   10.18   <2e-16 ***
Customer_SVC_H     0.414691   0.006540   63.41   <2e-16 ***
Staff_Cared_H      0.123561   0.008138   15.18   <2e-16 ***
Condition_Hotel_H  0.298358   0.007456   40.01   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.138 on 24161 degrees of freedom
Multiple R-squared:  0.6808,	Adjusted R-squared:  0.6807
F-statistic: 1.03e+04 on 5 and 24161 DF,  p-value: < 2.2e-16
```

#Multiple R-squared: 0.6808,   Adjusted R-squared: 0.6807

Adding Check_In_H

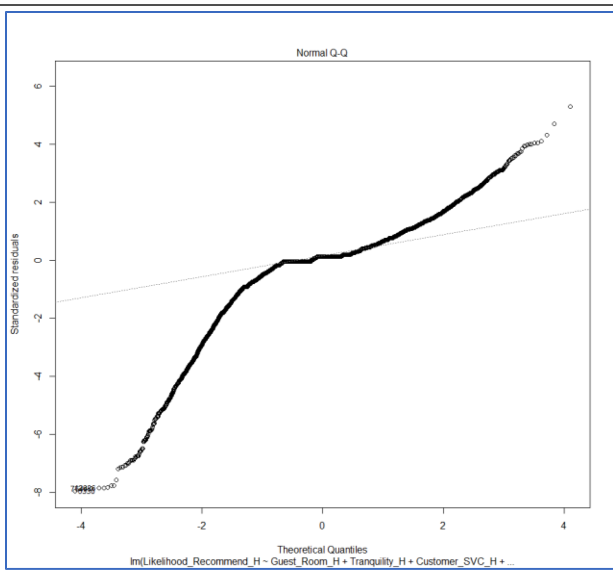
Model3AndCheckIn <- lm(Likelihood_Recommend_H ~ Guest_Room_H +Tranquility_H + Customer_SVC_H + Check_In_H,ModellingData)

summary(Model3AndCheckIn)

```
> summary(Model3AndCheckIn)

Call:
lm(formula = Likelihood_Recommend_H ~ Guest_Room_H + Tranquility_H + 
    Customer_SVC_H + Check_In_H + Condition_Hotel_H, data = ModellingData)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.0055 -0.0956  0.1339  0.4450  5.4716 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -2.232720   0.069939 -31.924  < 2e-16 ***
Guest_Room_H       0.349749   0.006671  52.427  < 2e-16 ***
Tranquility_H      0.100761   0.006519  15.456  < 2e-16 ***
Customer_SVC_H     0.454992   0.006046  75.250  < 2e-16 ***
Check_In_H         0.024363   0.007702   3.163  0.00156 ** 
Condition_Hotel_H  0.293961   0.007510  39.145  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.143 on 24161 degrees of freedom
Multiple R-squared:  0.6779,	Adjusted R-squared:  0.6778 
F-statistic: 1.017e+04 on 5 and 24161 DF,  p-value: < 2.2e-16
```

#Multiple R-squared: 0.6779,   Adjusted R-squared: 0.6778

#We shall be proceeding with Model3AndStaff and the plot is as shown below:



From the Linear Model we have shortlisted 4 attributes i.e Guest_Room_H ,Tranquility_H , Customer_SVC_H , Staff_Cared_H and Condition_Hotel_H. We received the R-squared value of 0. 6808 using these values and shall be testing to determine Net Promoter Score(NPS) using the same set of variables.

To Validate the accuracy or appropriateness of the variables shortlisted we shall be first predicting NPS using **KSVM**(**Support Vector Machines**). To run the model we have first installed and loaded “kernlab” package in R.

install.packages("kernlab")

library("kernlab")

 # #Step 1: Create train and test data sets

random.indexes <- sample(1:nrow(ModellingData),replace=TRUE)

cutpoint2_3 <- floor(nrow(ModellingData) /3 * 2)

ModellingData.train <- ModellingData[random.indexes[1:cutpoint2_3],]

ModellingData.test <- ModellingData[random.indexes[cutpoint2_3 + 1: nrow(ModellingData)],]

ModellingData.test <- na.omit(ModellingData.test)

View(ModellingData.train)

View(ModellingData.test)

 # Using KSVM

svmOutput <- ksvm(Likelihood_Recommend_H ~ Guest_Room_H +Tranquility_H + Customer_SVC_H + Condition_Hotel_H + Staff_Cared_H, data=ModellingData.train, kernel="rbfdot", kpar = "automatic", C= 10, cross= 10, prob.model=TRUE)

summary(svmOutput)

svmOutput

```
> svmOutput
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 10

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.479017211750374

Number of Support Vectors : 8116

Objective Function Value : -28538.28
Training error : 0.225469
Cross validation error : 1.191575
Laplace distr. width : 1.64302
```

#Predicting

svmPred <- predict(svmOutput,ModellingData.test, type="votes")

svmPred

Rounded.svmPred <- round(svmPred)

#Creating Data Frame of Actual and Predicted Value

CompTable1 <- data.frame(ModellingData.test$Likelihood_Recommend_H, Rounded.svmPred)

View(CompTable1)

#Renaming column names

colnames(CompTable1) <- c("Test", "Predicted")

```
View(CompTable1)
results <- table(CompTable1)
results
CompTable.ex<- CompTable1
CompTable.ex$TestNPS <- NA
CompTable.ex$PredictedNPS <- NA
```

 # Putting NPS Type for Test Column on the basis of Value

```
  for(i in 1:nrow(CompTable.ex)){
    if(CompTable.ex[i,1] >= 9){
      CompTable.ex[i,3] <- " Promoter"
    } else if (CompTable.ex[i,1]==7 | CompTable.ex[i,1]==8){
      CompTable.ex[i,3] <- " Passive"
    } else{
      CompTable.ex[i,3] <- "Detractor"
  }
}

View(CompTable.ex)
```

 # Putting NPS Type for Predicted Column on the basis of Value

```
for(i in 1:nrow(CompTable.ex)){
  if(CompTable.ex[i,2] >= 9){
    CompTable.ex[i,4] <- " Promoter"
  } else if (CompTable.ex[i,2]==7 | CompTable.ex[i,2]==8){
    CompTable.ex[i,4] <- " Passive"
  } else{
```

```
CompTable.ex[i,4] <- "Detractor"
  }
}

View(CompTable.ex)

# Creating a New Data Frame of the Predicted NPS TYPE

CompTable1.NPS <- data.frame(CompTable.ex$TestNPS,CompTable.ex$PredictedNPS)
View(CompTable1.NPS)
head(CompTable1.NPS)
colnames(CompTable1.NPS) <- c("TestNPS","PredictedNPS")
results.NPS <- table(CompTable1.NPS)
results.NPS
```

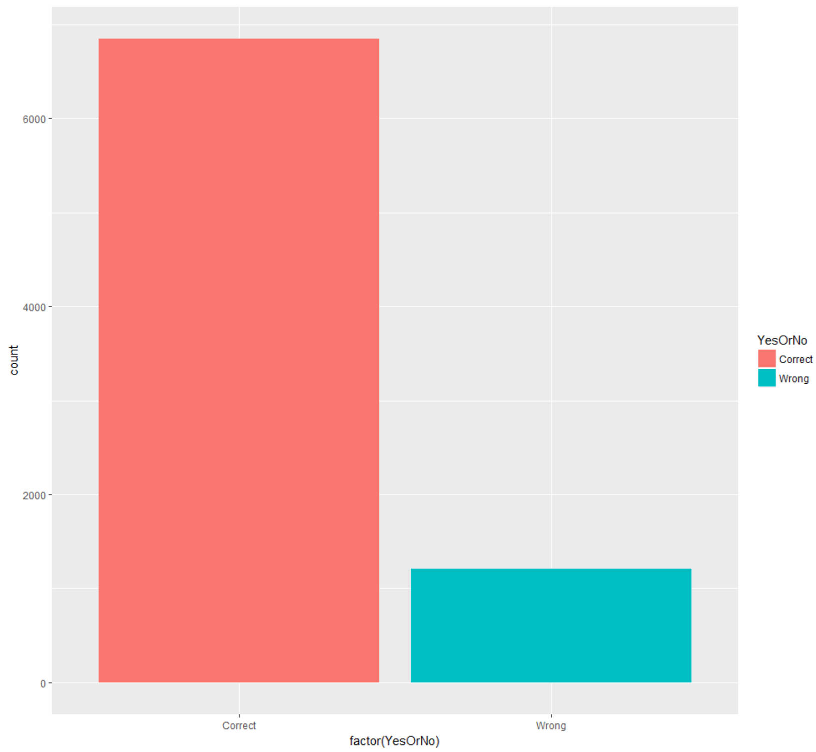
```
> results.NPS
            PredictedNPS
TestNPS      Passive  Promoter Detractor
   Passive       808       501       116
   Promoter      277      5357        33
  Detractor      181       101       682
```

```
AccuracyKSVM <-
((results.NPS[1,1]+results.NPS[2,2]+results.NPS[3,3])/(results.NPS[1,1]+results.NPS[1,2]+results.NPS[1,3]+results.NPS[2,1]+results.NPS[2,2]+results.NPS[2,3]+results.NPS[3,1]+results.NPS[3,2]+results.NPS[3,3])) * 100
AccuracyKSVM
```

```
> AccuracyKSVM
[1] 84.99255
```

```
plot(svmOutput)
CompTable1.NPS$YesOrNo <-
ifelse(CompTable1.NPS$TestNPS==CompTable1.NPS$PredictedNPS,"Correct","Wrong")
View(CompTable1.NPS)
#Plotting with respect Actual Value , Predicted Value and if predicted correctly or not
```

```
library(ggplot2)

KSVMplot <- ggplot(CompTable1.NPS, aes(x=factor(YesOrNo))) +
geom_bar(aes(fill=YesOrNo))

KSVMplot
```



Through KSVM, we received the prediction of accuracy of 92.92%. This implies that the 4 attributes or columns we determined are important to determine the Net Promoter Score(NPS) of any hotel. The Hotel can probably focus on those four attributes to increase their NPS score.

We shall check the same set of attributes using SVM Model as well. For that we shall be installing and loading "e1071" Package in R.

```
##########################SVM###################

install.packages("e1071")

library(e1071)

svmOutput2 <- svm(Likelihood_Recommend_H ~ Guest_Room_H +Tranquility_H +
Customer_SVC_H + Condition_Hotel_H + Staff_Cared_H, data=ModellingData.train)

svmOutput2
```

```
svmPred2 <- predict(svmOutput2,ModellingData.test, type="votes")

View(svmPred2)

Rounded.svmPred2 <- round(svmPred2)


#Creating Data Frame of Actual and Predicted Value

CompTable2 <- data.frame(ModellingData.test$Likelihood_Recommend_H,
Rounded.svmPred2)

View(CompTable2)


#Renaming column names

colnames(CompTable2) <- c("Test", "Predicted")

View(CompTable2)


CompTable2$TestNPS <- NA

CompTable2$PredictedNPS <- NA



# Putting NPS Type for Test Column on the basis of Value


for(i in 1:nrow(CompTable2)){
  if(CompTable2[i,1] >= 9){
    CompTable2[i,3] <- " Promoter"
  } else if (CompTable2[i,1]==7 | CompTable2[i,1]==8){
    CompTable2[i,3] <- " Passive"
  } else{
    CompTable2[i,3] <- "Detractor"
  }
}
```

```
View(CompTable2)
```

 # Putting NPS Type for Predicted Column on the basis of Value

```
for(i in 1:nrow(CompTable2)){
  if(CompTable2[i,2] >= 9){
    CompTable2[i,4] <- " Promoter"
  } else if (CompTable2[i,2]==7 | CompTable2[i,2]==8){
    CompTable2[i,4] <- " Passive"
  } else{
    CompTable2[i,4] <- "Detractor"
  }
}

View(CompTable2)
```

 # Creating a New Data Frame of the Predicted NPS TYPE

```
CompTable2.NPS <- data.frame(CompTable2$TestNPS,CompTable2$PredictedNPS)
View(CompTable2.NPS)

head(CompTable2.NPS)
colnames(CompTable2.NPS) <- c("TestNPS","PredictedNPS")
results2.NPS <- table(CompTable2.NPS)
results2.NPS
```

```
> results2.NPS
            PredictedNPS
TestNPS      Passive  Promoter Detractor
   Passive       801       487       137
   Promoter      353      5292        22
  Detractor      240        85       639
```

Using SVM Modelling technique, we got the prediction accuracy of 92.07%.

Accuracy.SVM <- ((results2.NPS[1,1]+results2.NPS[2,2]+results2.NPS[3,3])/(results2.NPS[1,1]+results2.NPS[1,2]+results2.NPS[1,3]+results2.NPS[2,1]+results2.NPS[2,2]+results2.NPS[2,3]+results2.NPS[3,1]+results2.NPS[3,2]+results2.NPS[3,3])) * 100

Accuracy.SVM

```
> Accuracy.SVM
[1] 83.56504
```

CompTable2.NPS$YesOrNo <- ifelse(CompTable2.NPS$TestNPS==CompTable2.NPS$PredictedNPS,"Correct","Wrong")

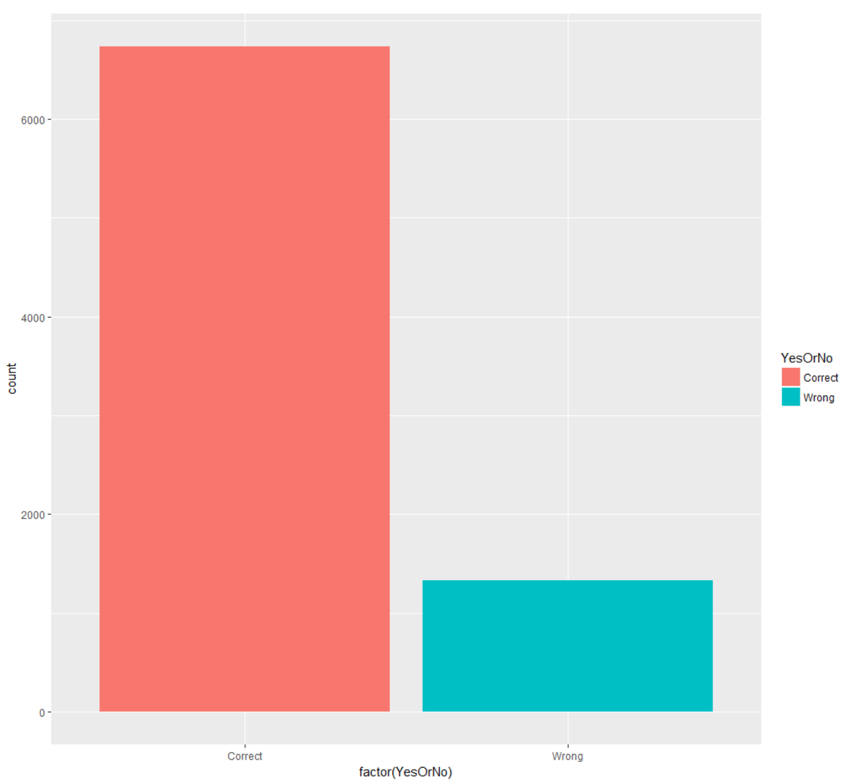
View(CompTable2.NPS)

#Plotting with respect Actual Value , Predicted Value and if predicted correctly or not

library(ggplot2)

SVMplot <- ggplot(CompTable2.NPS, aes(x=factor(YesOrNo))) + geom_bar(aes(fill=YesOrNo))

SVMplot



After determining the numeric factors, we determine columns among the list of columns which shows whether a hotel have particular amenity or not. The answer to those questions in the dataset is in the form of Y (a facility is available) or N(a facility is not available).

As those columns are factors and not numeric we shall be using Associative Rule Mining to determine which are the important amenities or facilities which contributes towards a Hotel's NPS being Promoter or Detractor.

In order to implement Associative Rule Mining we shall be requiring installation and loading of two important packages named "arules" and "arulesViz" in R.

We have created a separate column named "LikelihoodCategory" whose value will be "high", "medium" or "low" on the basis of Likelihood to Recommend score.

- Score of 9 or more will be considered as High
- Score of 7 or 8 will be considered as Medium
- Score of less than 7 will be considered as Low

```
install.packages("arules")

install.packages("arulesViz")

library(arulesViz)

library(arules)

 ModellingData2 <- ModellingData

 ModellingData2$LikelihoodCategory <- NA


 # Creating Category " High", " Medium"," Low"


 for(i in 1:nrow(ModellingData2)){

  if(ModellingData2[i,6] >= 9){

   ModellingData2[i,ncol(ModellingData2)] <- "High"

  } else if (ModellingData2[i,6]==7 | ModellingData2[i,6]==8){

   ModellingData2[i,ncol(ModellingData2)] <- "Medium"

  } else{

   ModellingData2[i,ncol(ModellingData2)] <- "Low

}

}
```

```
ModellingData2$LikelihoodCategory <- as.factor(ModellingData2$LikelihoodCategory)
View(ModellingData2)

# Creating a new Data Set with Factor Variables

ModellingData2.factor <- ModellingData2[,23:46]

ModellingData2.factor$LikelihoodCategory  <- ModellingData2$LikelihoodCategory
View(ModellingData2.factor)
str(ModellingData2.factor)
summary(ModellingData2.factor)
```

```
>   summary(ModellingData2.factor)
 All.Suites_PL Bell.Staff_PL Boutique_PL Business.Center_PL Casino_PL Conference_PL Convention_PL Dry.Cleaning_PL Elevators_PL
 N:14111       N:10426       N:23220     N: 1220            N:24167   N:24167       N:13473       Y:24167         N:  343
 Y:10056       Y:13741       Y:  947     Y:22947                                    Y:10694                       Y:23824



 Fitness.Center_PL Golf_PL   Laundry_PL Limo.Service_PL Mini.Bar_PL Pool.Indoor_PL Pool.Outdoor_PL Regency.Grand.Club_PL Resort_
 N:  566           N:20725   N: 5195    N:16991         N:20868     N:23357        N:  934         N:15116               N:20482
 Y:23601           Y: 3442   Y:18972    Y: 7176         Y: 3299     Y:  810        Y:23233         Y: 9051               Y: 3685



 Restaurant_PL Self.Parking_PL Shuttle.Service_PL Spa_PL    Valet.Parking_PL              Booking_Channel  LikelihoodCategory
 N: 8803       N: 2624         N:10702            N:16977   N:12138          Digital Channels     :12941   High  :16933
 Y:15364       Y:21543         Y:13465            Y: 7190   Y:12029          GDS                  :  444   Low   : 5843
                                                                             Global Contact Center: 7984   Medium: 1391
                                                                             Hotel                : 2306
                                                                             Internet             :  482
                                                                             Opaque               :   10
>
```

##############Reducing Columns on the basis on of the summary##########

 # All the Values in Casino_PL and Conference_PL are N, hence removing it

ModellingData2.factor <- ModellingData2.factor[,-5:-6]

 # All the Values in Dry.Cleaning_PL are Y, hence removing it

ModellingData2.factor <- ModellingData2.factor[,-6]

.

#Removing Booking Channel

ModellingData2.factor <- ModellingData2.factor[,-21]

#Removing Elevators_PL as the count is too one sided. Most of the people counted it as Y

ModellingData2.factor <- ModellingData2.factor[,-6]

 # Choosing the one with mixed Reviews

#All.Suites_PL , Bell.Staff_PL, Convention_PL, Limo.Service_PL, Mini.Bar_P, Regency.Grand.Club_PL, Restaurant_PL, Shuttle.Service_PL, Valet.Parking_PL, LikelihoodCategory

ModellingData2.Selected <- ModellingData2.factor

ModellingData2.Selected <- ModellingData2.Selected[,c(1,2,5,9,10,13,15,17,18,19,20)]

View(ModellingData2.Selected)

 # Use arules to calculate some rules (clusters) for the dataset

ruleset <- apriori(ModellingData2.Selected,parameter=list(support=0.01,confidence=0.5),appearance=list(default="lhs",rhs=("LikelihoodCategory=High")))

#Summary of the plot

summary(ruleset)

```
> summary(ruleset)
set of 178 rules

rule length distribution (lhs + rhs):sizes
 1  2  3  4  5  6
 1 10 39 65 49 14

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   4.000   4.084   5.000   6.000

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.01370   Min.   :0.5460   Min.   :0.7090   Min.   :  427
 1st Qu.:0.04991   1st Qu.:0.6790   1st Qu.:0.8818   1st Qu.: 1555
 Median :0.11700   Median :0.7307   Median :0.9489   Median : 3646
 Mean   :0.13843   Mean   :0.7569   Mean   :0.9830   Mean   : 4313
 3rd Qu.:0.18585   3rd Qu.:0.7738   3rd Qu.:1.0050   3rd Qu.: 5790
 Max.   :0.77000   Max.   :1.0000   Max.   :1.2987   Max.   :23991

mining info:
                  data ntransactions support confidence
 ModellingData2.Selected         31157    0.01        0.5
```

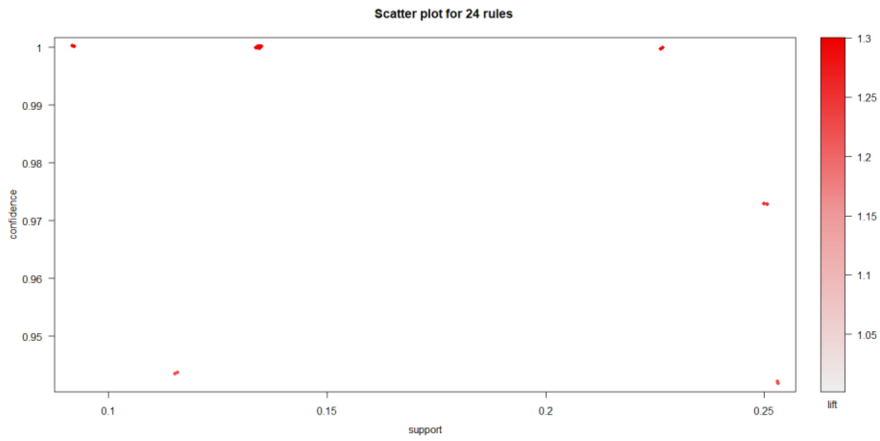
```
goodrules <- ruleset[quality(ruleset)$lift > 1.2 ]

goodrules

inspect(goodrules)

goodrules <- sort(goodrules,by='lift',decreasing=T)

summary(goodrules)

inspect(goodrules)

plot(goodrules)
```

```
#rule length distribution (lhs + rhs):sizes
#3   4   5   6   7   8   9  10
#16  84 185 252 223 123  38   5
```

This is the rule length distribution we got. This implies that columns 5,6,7,8 comes in most of the rules create. Thus we proceed with those rules and we decided to trim the variables and determine the rules for the LikelihoodCategory = High i.e Promoters.

```
ModellingData2.Selected <-  ModellingData2.Selected[,c(5,6,7,8,9,10)]

 View(ModellingData2.Selected)

   ruleset2 <-
apriori(ModellingData2.Selected,parameter=list(support=0.01,confidence=0.5),appearance=li
st(default="lhs",rhs=("LikelihoodCategory=High")))

 #Summary of the plot

 summary(ruleset2)

 #visualize the plot

 plot(ruleset2)


 goodrules2 <- ruleset2[quality(ruleset2)$lift > 1.2 ]

 goodrules2

 goodrules2 <- sort(goodrules2,by='lift',decreasing=T)

 summary(goodrules2)

 inspect(goodrules2)
```

plot(goodrules2)



From the Association Rule Mining we got the findings that hotels having MiniBar, Valet Parking, Shuttle Service and Spa Service have the likelihood Category = High i.e Promoter

This aligns with our descriptive analysis mentioned above which states that:

- 69% of the people who have visited the hotel for leisure were promoters because the hotel had shuttle service.
- 73% of the people who have visited the hotel for leisure were detractors because the hotel did not have spa service.
- 89% of the people who have visited the hotel for leisure were detractors because the hotel did not have a mini bar.
- 63% of the people who have visited the hotel for leisure were promoters because the hotel had valet parking.

Comparison of Hotels and Different Location

We decided that it would be a good practice to compare two set of hotels on the basis of possible factors. The model is based on the following set of rules:

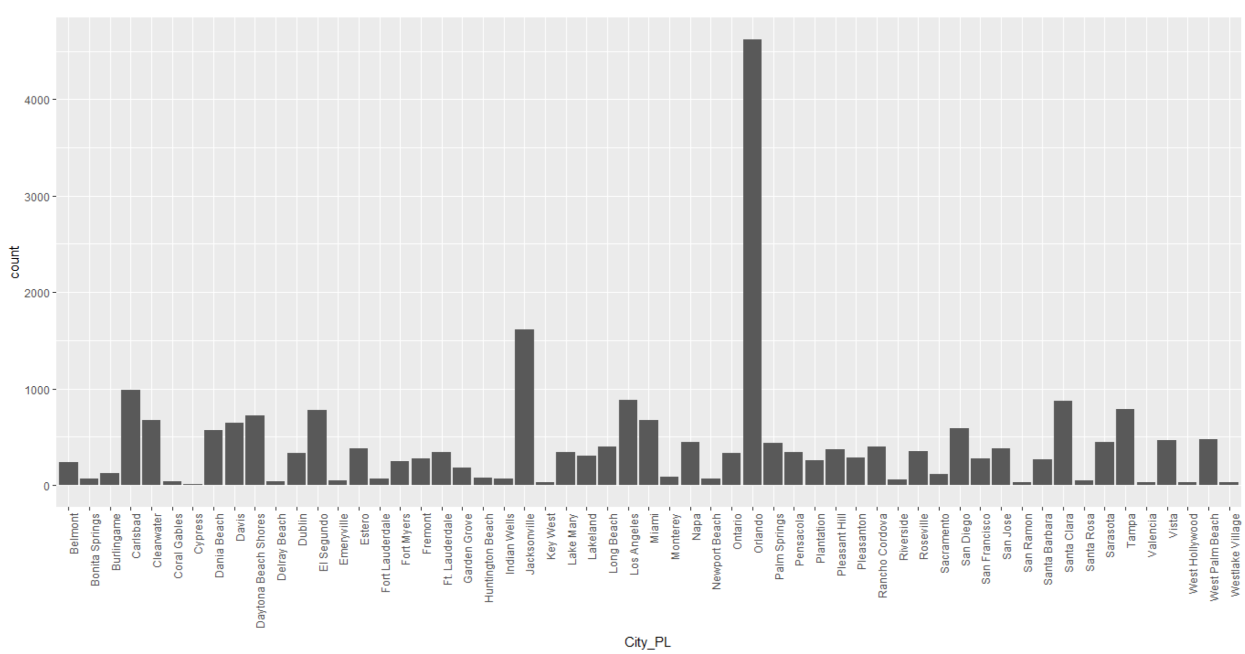
- We found out the location with maximum number of promoters and within that state we determine the hotel with maximum NPS value. We determine the values of the ratings by taking mean and set it as the benchmark for the other hotels.
- Secondly, we compared the same set of factors for the hotel at other location, so that they can get an idea of which factors to focus on to reach the benchmark set by the best hotel.

Following are the attributes we considered:

- Check_In_H
- Condition_Hotel_H
- Customer_SVC_H
- F.B_Overall_Experience_H
- Guest_Room_H
- Internet_Sat_H
- Staff_Cared_H
- Tranquility_H

In order to determine the “Best Hotel” we initially figured out the place with maximum number of promoters.

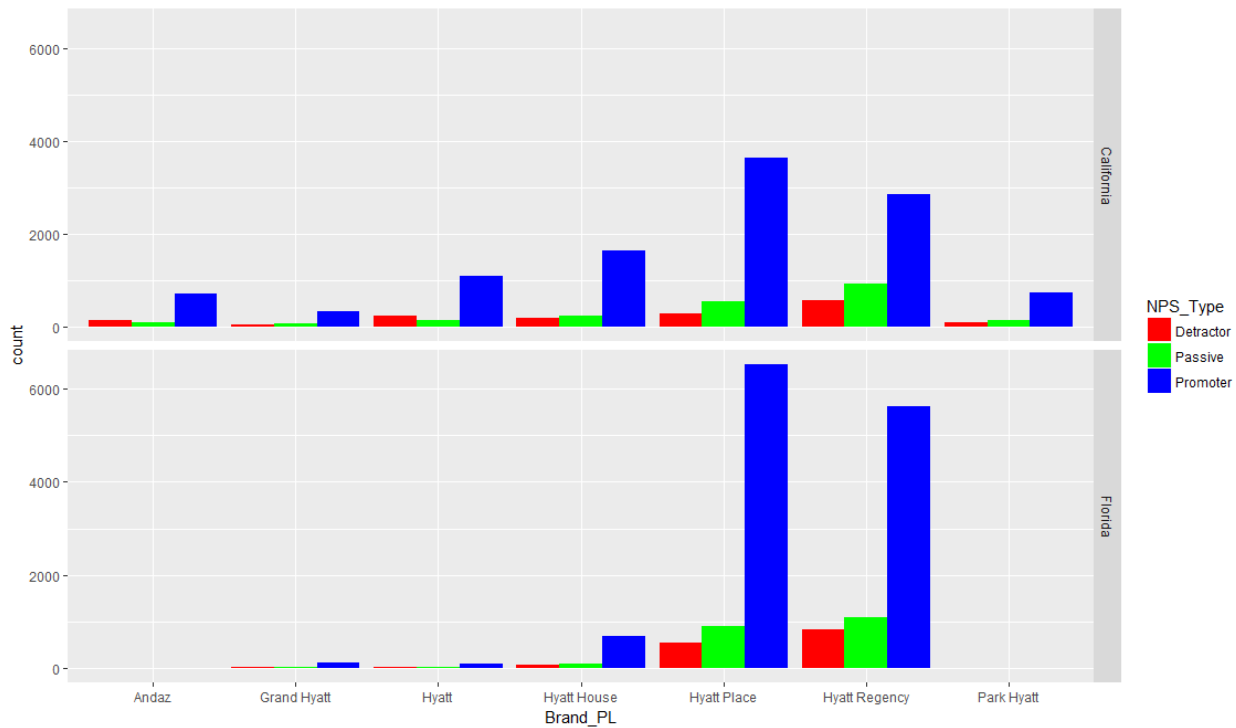
```
Promoters <- ModellingData[which(ModellingData$NPS_Type=="Promoter"),]

View(Promoters)

CityPromoters <-ggplot(Promoters, aes(x=City_PL)) + geom_bar(aes(fill=CHECKOUT_HEADER_ID_C),
position ="dodge") + theme(axis.text.x = element_text(angle = 90, hjust = 1))

CityPromoters
```



Orlando is the city with most number of Promoters.

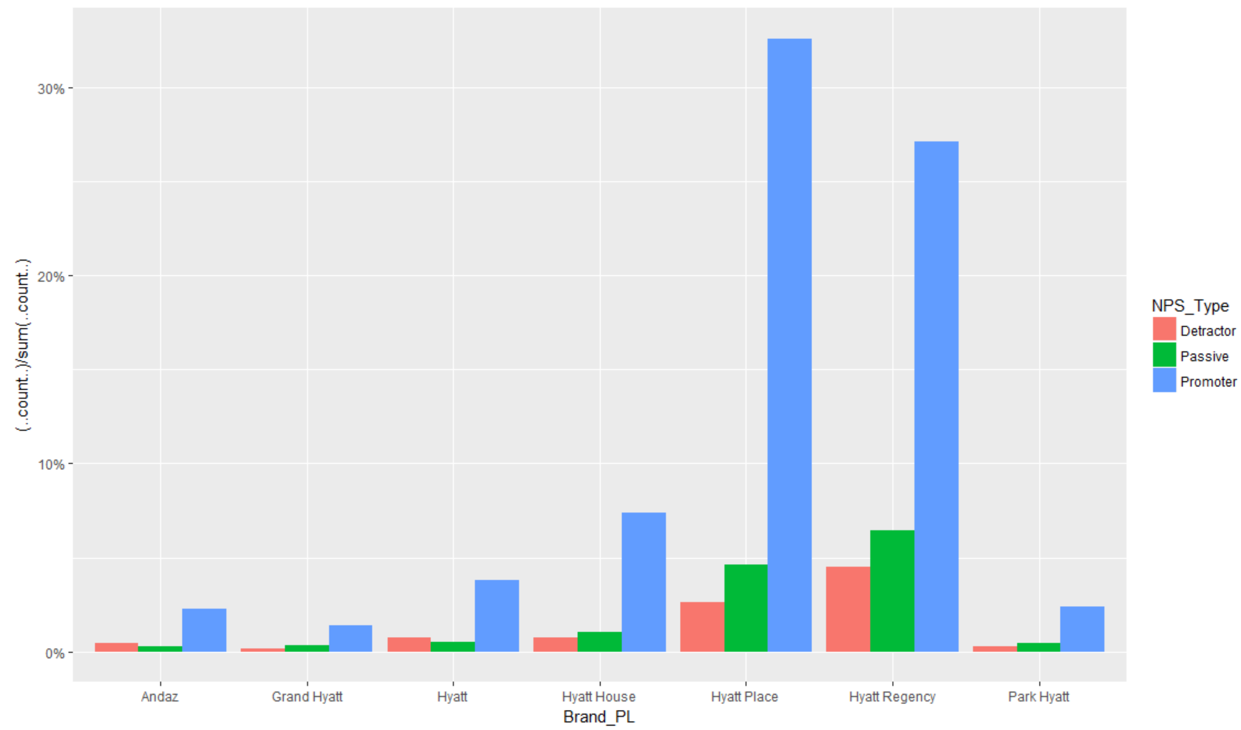
Next, we checked the count of Promoters and Detractors of each hotel within each state.

```
HotelPromoter <- ggplot(ModellingData, aes(x=Brand_PL)) + geom_bar(aes(fill=NPS_Type),
position="dodge") + facet_grid(State_PL ~ .) + scale_fill_manual(values=c("red", "green","blue"))

HotelPromoter
```



However, the insights we received from this plot were not clear enough to reach to a conclusion, hence we created a plot with percentage to determine the rough estimate of the NPS value

```
HotelNPS <- ggplot(ModellingData, aes(x=Brand_PL)) + geom_bar(aes(fill=NPS_Type,
y=(..count..)/sum(..count..)), position="dodge")  + scale_y_continuous(labels = scales::percent) +
ylab='Percentage'

HotelNPS
```



From the above plot we reached to the conclusion that Hyatt Place has the highest NPS Value as compared to other hotels in California and Florida.

To determine the "Best Hotel" we picked the Hyatt Place (Highest NPS) in Orlando Location (maximum number of promoters) and set the parameters as the benchmark for other hotels and compared it with all the hotels at Dania Beach.

```
ModellingData.Orlando <- ModellingData[which(ModellingData$City_PL=="Orlando" &
ModellingData$Brand_PL=='Hyatt Place'),]

View(ModellingData.Orlando)

ModellingData.DaniaBeach <- ModellingData[which(ModellingData$City_PL=="Dania
Beach"),]

ModellingData.compare <- rbind(ModellingData.Orlando,ModellingData.DaniaBeach)

View(ModellingData.compare)

#Checking the Factors
```

```
# Overall Satisfaction:

BestFactors1 =
mean(ModellingData.compare$Overall_Sat_H[(ModellingData.compare$City_PL=="Orland
o")], na.rm=T)

print(paste("Desired Overall Satisfaction: ",BestFactors1))


#Guest_Room_H:


BestFactors2 =
mean(ModellingData.compare$Guest_Room_H[(ModellingData.compare$City_PL=="Orlan
do")], na.rm=T)

print(paste("Desired Guest Room Satisfaction: ",BestFactors2))


#Tranquility_H


BestFactors3 =
mean(ModellingData.compare$Tranquility_H[(ModellingData.compare$City_PL=="Orlando
")], na.rm=T)

print(paste("Desired Tranquility Satisfaction: ",BestFactors3))


#Condition_Hotel_H

BestFactors4 =
mean(ModellingData.compare$Condition_Hotel_H[(ModellingData.compare$City_PL=="Or
lando")], na.rm=T)

print(paste("Desired Condition Satisfaction: ",BestFactors4))


#Customer_SVC_H


BestFactors5 =
mean(ModellingData.compare$Customer_SVC_H[(ModellingData.compare$City_PL=="Orl
ando")], na.rm=T)

print(paste("Desired Customer Service Satisfaction: ",BestFactors5))
```

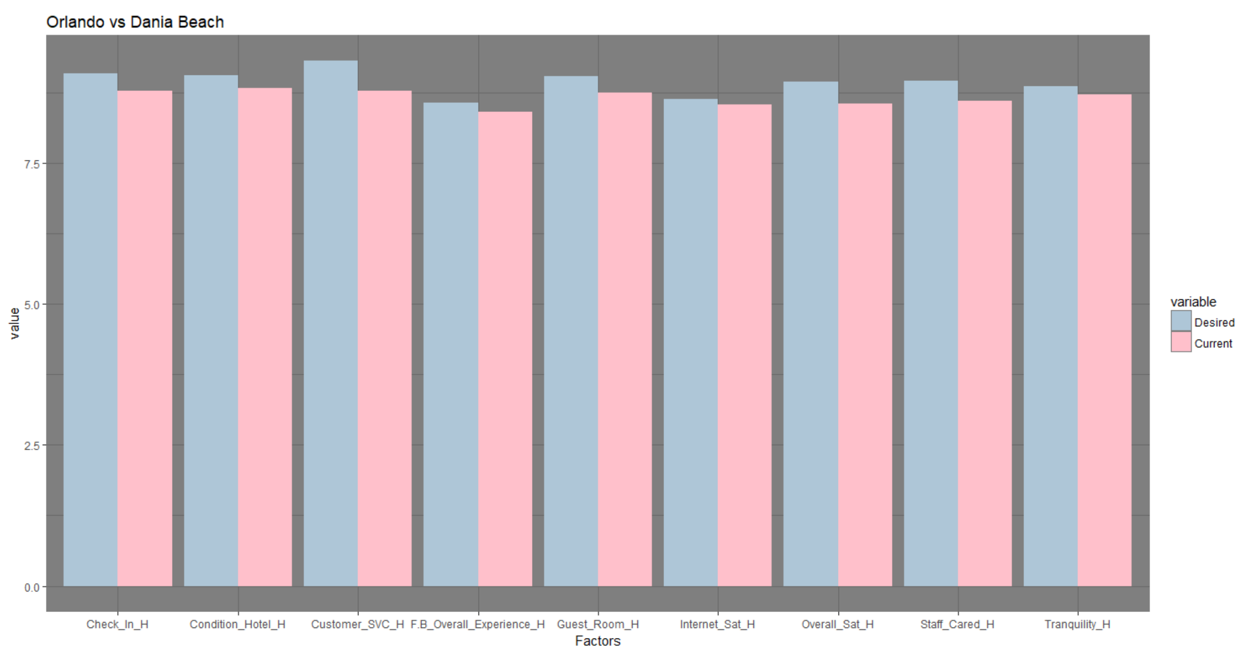
```
#Staff_Cared_H


BestFactors6 =
mean(ModellingData.compare$Staff_Cared_H[(ModellingData.compare$City_PL=="Orland
o")], na.rm=T)

print(paste("Desired Staff Service Satisfaction: ",BestFactors6))


#Internet_Sat_H


BestFactors7 =
mean(ModellingData.compare$Internet_Sat_H[(ModellingData.compare$City_PL=="Orland
o")], na.rm=T)

print(paste("Desired Internet Satisfaction: ",BestFactors7))


#Check_In_H

BestFactors8 =
mean(ModellingData.compare$Check_In_H[(ModellingData.compare$City_PL=="Orlando")
], na.rm=T)

print(paste("Desired Check In Satisfaction: ",BestFactors8))


#F.B_Overall_Experience_H

BestFactors9 =
mean(ModellingData.compare$F.B_Overall_Experience_H[(ModellingData.compare$City_P
L=="Orlando")], na.rm=T)

print(paste("Desired Food and Beverages Satisfaction: ",BestFactors9))
```

```
#########Hotels in Dania Beach#####


DBFactors1 = mean(ModellingData.compare$Overall_Sat_H[(ModellingData.compare$City_PL=="Dania Beach")], na.rm=T)

print(paste("Current Overall Satisfaction: ",DBFactors1))


#Guest_Room_H:


DBFactors2 = mean(ModellingData.compare$Guest_Room_H[(ModellingData.compare$City_PL=="Dania Beach")], na.rm=T)

print(paste("Current Guest Room Satisfaction: ",DBFactors2))


#Tranquility_H


DBFactors3 = mean(ModellingData.compare$Tranquility_H[(ModellingData.compare$City_PL=="Dania Beach")], na.rm=T)

print(paste("Current Tranquility Satisfaction: ",DBFactors3))


#Condition_Hotel_H

DBFactors4 = mean(ModellingData.compare$Condition_Hotel_H[(ModellingData.compare$City_PL=="Dania Beach")], na.rm=T)

print(paste("Current Condition Satisfaction: ",DBFactors4))


#Customer_SVC_H


DBFactors5 = mean(ModellingData.compare$Customer_SVC_H[(ModellingData.compare$City_PL=="Dania Beach")], na.rm=T)

print(paste("Current Customer Service Satisfaction: ",DBFactors5))
```

```
#Staff_Cared_H


DBFactors6 =
mean(ModellingData.compare$Staff_Cared_H[(ModellingData.compare$City_PL=="Dania
Beach")], na.rm=T)

print(paste("Current Staff Service Satisfaction: ",DBFactors6))


#Internet_Sat_H


DBFactors7 =
mean(ModellingData.compare$Internet_Sat_H[(ModellingData.compare$City_PL=="Dania
Beach")], na.rm=T)

print(paste("Current Internet Satisfaction: ",DBFactors7))


#Check_In_H

DBFactors8 =
mean(ModellingData.compare$Check_In_H[(ModellingData.compare$City_PL=="Dania
Beach")], na.rm=T)

print(paste("Current Check In Satisfaction: ",DBFactors8))


#F.B_Overall_Experience_H

DBFactors9 =
mean(ModellingData.compare$F.B_Overall_Experience_H[(ModellingData.compare$City_P
L=="Dania Beach")], na.rm=T)

print(paste("Current Food and Beverages Satisfaction: ",DBFactors9))
```

Now we plotted the graph to compare the parameters from the two locations.

```
##########Plotting Graph###

Factors <-
c("Overall_Sat_H","Guest_Room_H","Tranquility_H","Condition_Hotel_H","Customer_SVC_H","Staff_Cared_H","Internet_Sat_H","Check_In_H","F.B_Overall_Experience_H")

Desired <-
c(BestFactors1,BestFactors2,BestFactors3,BestFactors4,BestFactors5,BestFactors6,BestFactors7,BestFactors8,BestFactors9)

Current <-
c(DBFactors1,DBFactors2,DBFactors3,DBFactors4,DBFactors5,DBFactors6,DBFactors7,DBFactors8,DBFactors9)

ComparisonDF <- data.frame(Factors,Desired,Current)


library(ggplot2)

library(reshape2)

meltedComparisonDF = melt(ComparisonDF, id = "Factors")

ggplot(meltedComparisonDF, aes(Factors, value)) +geom_bar(aes(fill = variable), position = "dodge", stat="identity")+ggtitle("Orlando vs Dania Beach") + theme(axis.text.x = element_text(angle = 90, hjust = 1)) + scale_fill_manual(values=c("#aec6d7","#ffc0cb")) + theme_dark()
```



In order to increase the net profitability and customer satisfaction, the hotels need to increase their current service ratings to the desired service ratings.

For example, for the above code, the average ratings of the Hyatt Place at Orlando is

- “Desired Overall Satisfaction” is 8.9464,
- “Desired Guestroom Satisfaction” is 9.0344,
- “Desired Tranquility Satisfaction” is 8.86514,
- “Desired Condition “Satisfaction” is 9.0468,
- “Desired Customer Service Satisfaction” is 9.30911,
- “Desired staff Service Satisfaction” is 8.9581,
- “Desired internet service satisfaction” is 8.6299,
- “Desired food and Beverages Satisfaction” is 8.56691,
- “Desired Check-in Satisfaction” is 9.0874.

Whereas, for the hotels at Dania Beach

- “Current Overall satisfaction” is 8.54866,
- “Current Overall Satisfaction: 8.5486618004866”
- "Current Guest Room Satisfaction: 8.74574209245742"
- "Current Tranquility Satisfaction: 8.71046228710462"
- Current Condition Satisfaction: 8.83211678832117"
- "Current Customer Service Satisfaction: 8.78467153284672
- "Current Staff Service Satisfaction: 8.60827250608273"
- Current Internet Satisfaction: 8.52919708029197
- Current Check in Satisfaction: 8.78467153284672
- "Current Food and Beverages Satisfaction: 8.40916463706569

Based on the insights provided above and the Net Promoter Score calculated, Hyatt Regency can work towards implementing necessary changes and thus improving customer satisfaction and thus their profitability.