IST 687: Introduction to Data Science
(Spring, 2019)

Final Project: Hyatt Hotels Customer Survey Analysis

Team West Coast:
Steven Mochalski
Ryan Timbrook
Bhavya Madhavan
Fumika Matsushita

Contents:

Background:

Hyatt Hotels Corporation is one of the largest hotel chains with locations across the world. Customer feedback is very important in understanding and evaluating the drivers of performance in their business. In 2013-2014, they conducted a customer survey, and over 19,000 customers who stayed in any Hyatt hotel participated. We analyzed this survey data using methods to create visual analysis with RStudio to identify customers' characteristics, performance and satisfaction. We also provided potential business recommendations based on our findings. This analysis is for Hyatt Hotels executives and anyone looking to gain a better understanding of the company's business.

Context:

The data originally consisted of 237 questionnaires. However, over two-thirds of the variables were not available due to empty columns. After cleaning the data, we decided to use the various attributes to answer our business questions. We also focused on analysis by Net Promoter Score (NPS) type, which labeled customers with high satisfaction scores 9-10 as promoters, customers with average scores as passives 7-8, and customers with low scores 0-6 as detractors.

Hotel Locations, Conditions and Revenue
ROOM_TYPE_CODE_R, Guest_Room_H, Condition_Hotel_H Condition_Hotel_H City_PL LENGTH_OF_STAY_C REVENUE_USD_R Property.Latitude_PL Property.Longitude_PL
Customer Informations
Gender_H Age_Range_H ADULT_NUM_C CHILDREN_NUM_C POV_CODE_C
Customer Reviews
Likelihood_Recommend_H (NPS_Type)

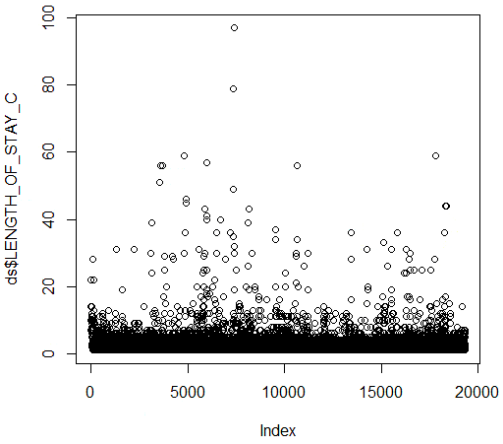
Business Questions:

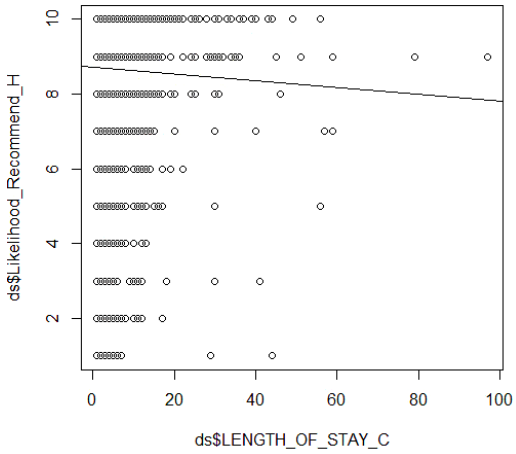
How does the length of stay impact NPS?

Our team evaluated the length of stay to understand the profile of the hotel and identify if there were any identified trends or correlation with the length of stay and NPS. We learned the average stay is very short with a median stay of 2 days and average stay of 2.4 days with a standard deviation of 2.99. 75% of guests stay 3 days or less and 95% of guests stay 5 days or less. 99% of the stays are less than 2 weeks long.

Because of the shape of the data, using linear regression was not a useful model. However, when looking at NPS type the Promoters have the lowest average stay of the NPS types, but it is only slightly shorter and not stat sig. (Note: only evaluated the top 99% of data to eliminate the length of stay outliers)

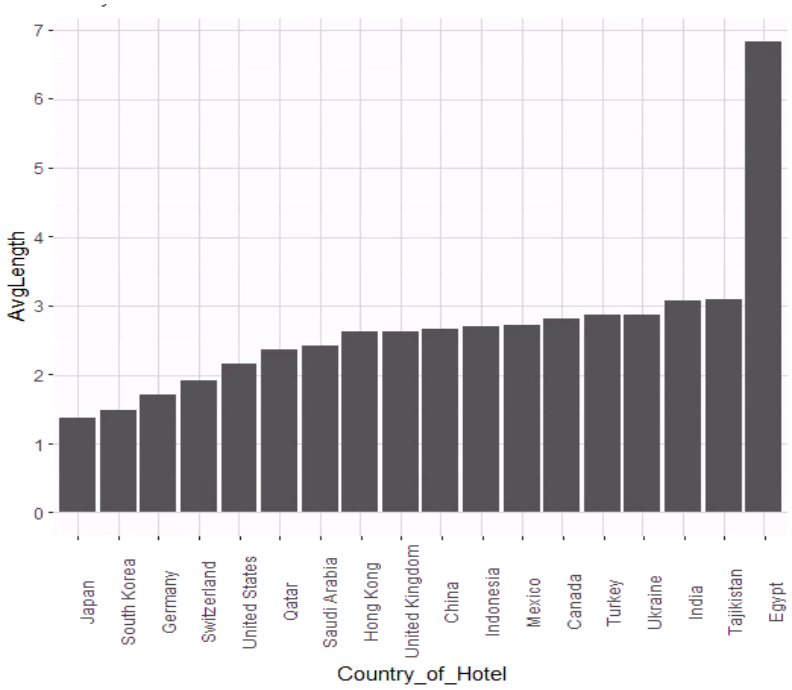
It would be worth understanding why a customer with the shortest length of stays have a slightly better NPS score. Are there issues with the hotel maintaining the room after 2 days that would increase the likelihood for detractors? Do we need to look into the cleaning service or customer service we provide on the 3rd or 4th day of a stay?





NPS	Avg. Length of Stay
Detractor	2.23
Passive	2.32
Promoter	2.16

When evaluating by the country where the hotel is located, 15 of the 18 countries had the average length of stay under 3 days, with two more countries (India and Tajikistan) under 3.1 days. Egypt clearly had the longest average length of stay at almost 7 days, which could lead to specific recommendations for that market. Additionally, 15 of the 18 countries all had an average stay that rounded to 2 or 3 days, which shows most of the hotels have very similar behavior. Japan and South Korea both rounded to 1 day stays, so we could look into handling them separately as it may be a geographic trait for Asian markets.



How do the different demographic components (age, gender) relate to NPS?

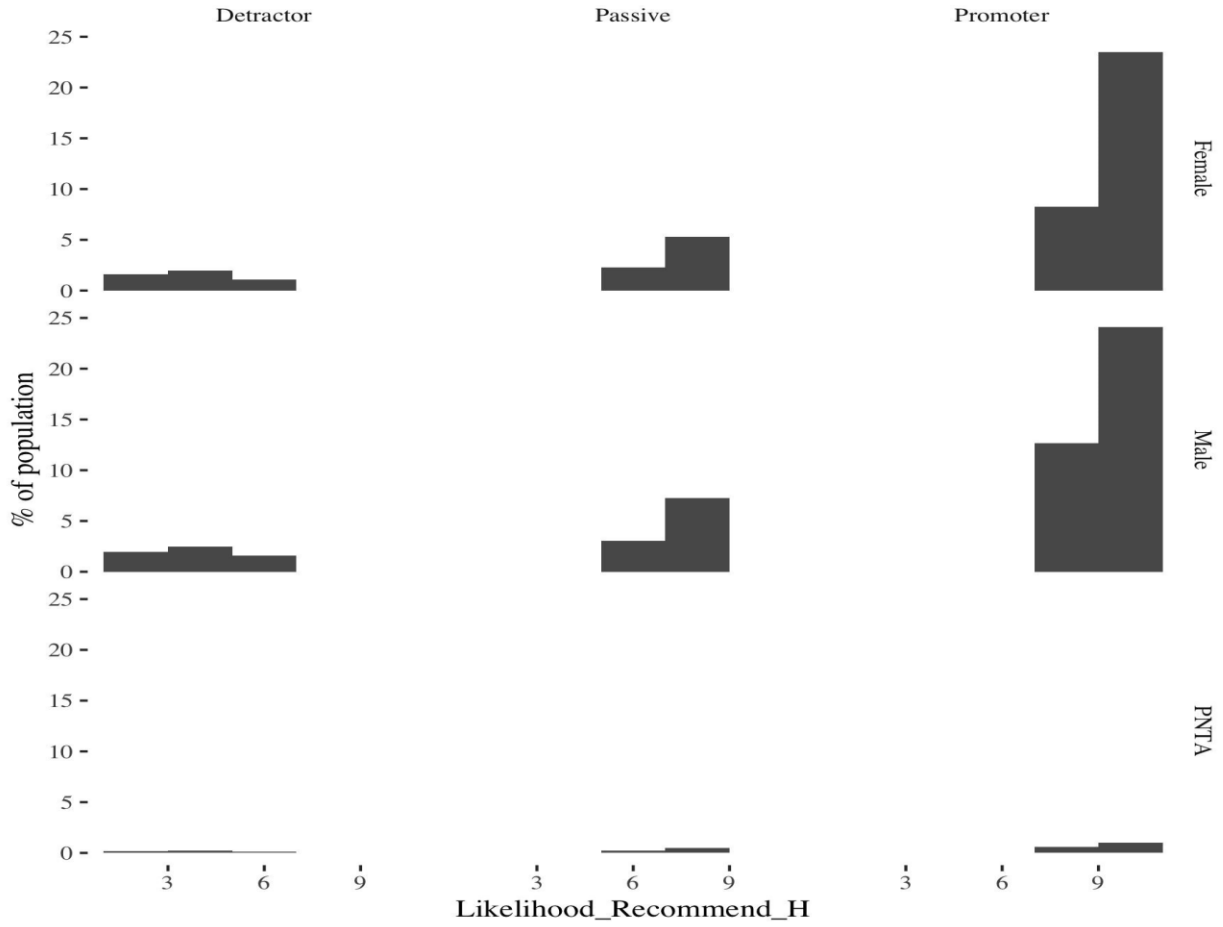
In this section, our team focused on how the demographic components like age and gender affected the NPS or likelihood to recommend score. We did both individual analysis i.e considering the age and gender as standalone and also by combining them.

As we can see from the gender plots below, the rough distribution of promoters, passives and detractors across male and female category remains the same. As a next step, we need to study how we can improve the service quality for male passives to move them into promoters category as well as for the female passives.

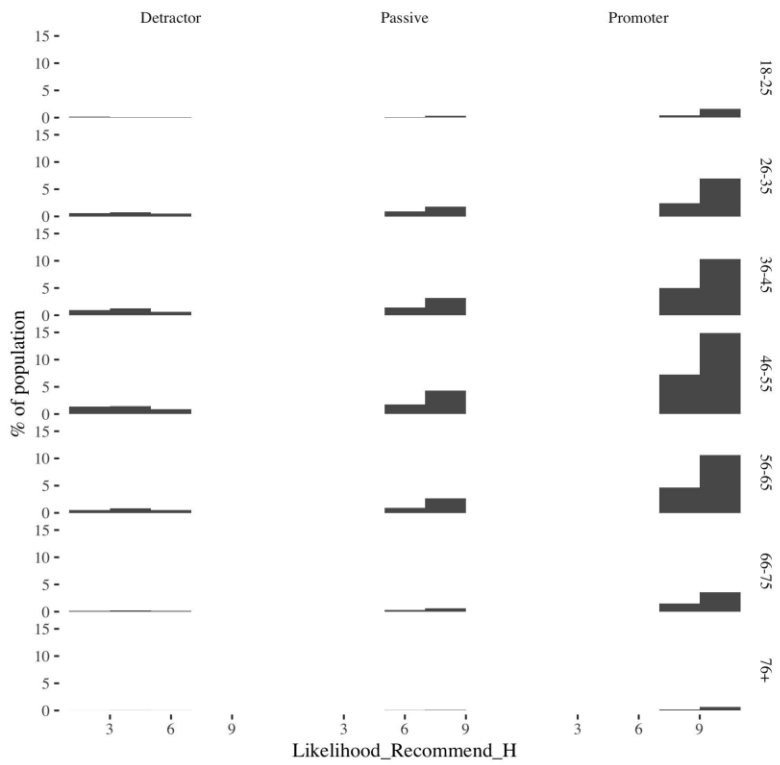
When we look at the age plots, we see that guests in the age range of 46-55 years are the most promoters. As with gender, we need to study how we can improve the service quality for guests in the age range of 36-45 years so as to move them from passives to promotor category.

The third plot gives a better visual display by combining the age and gender category into a single plot. When combining the demographics together, we learned that gender does not play a significant role in determining the NPS score. When we look at the percentage of passives across age groups, we find that there is opportunity to improve the hotel's services so that the older passive guest population could be potentially converted to promoter category

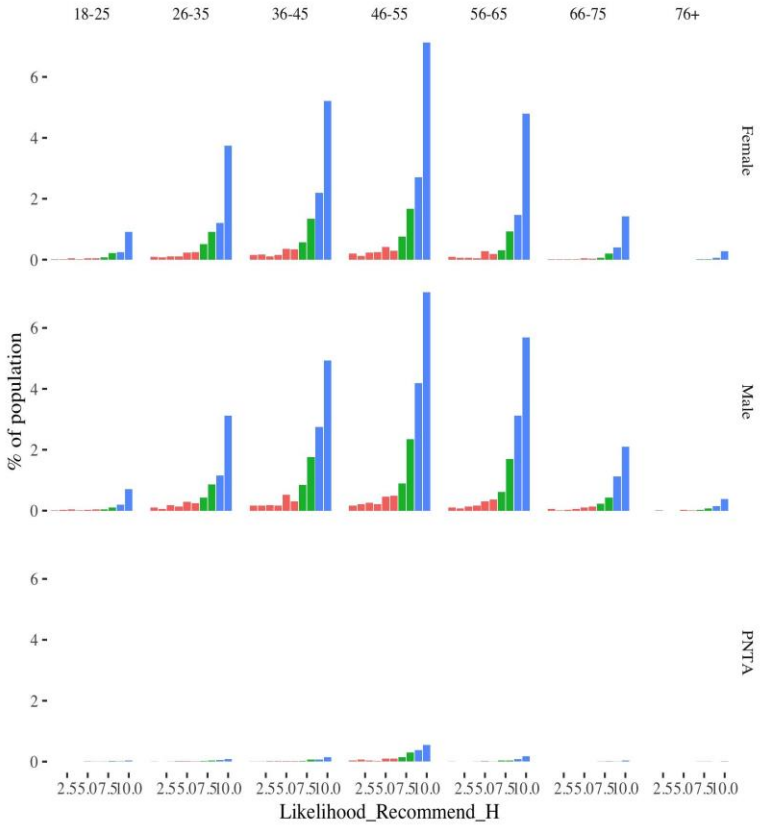
Gender Plot



Age Plot



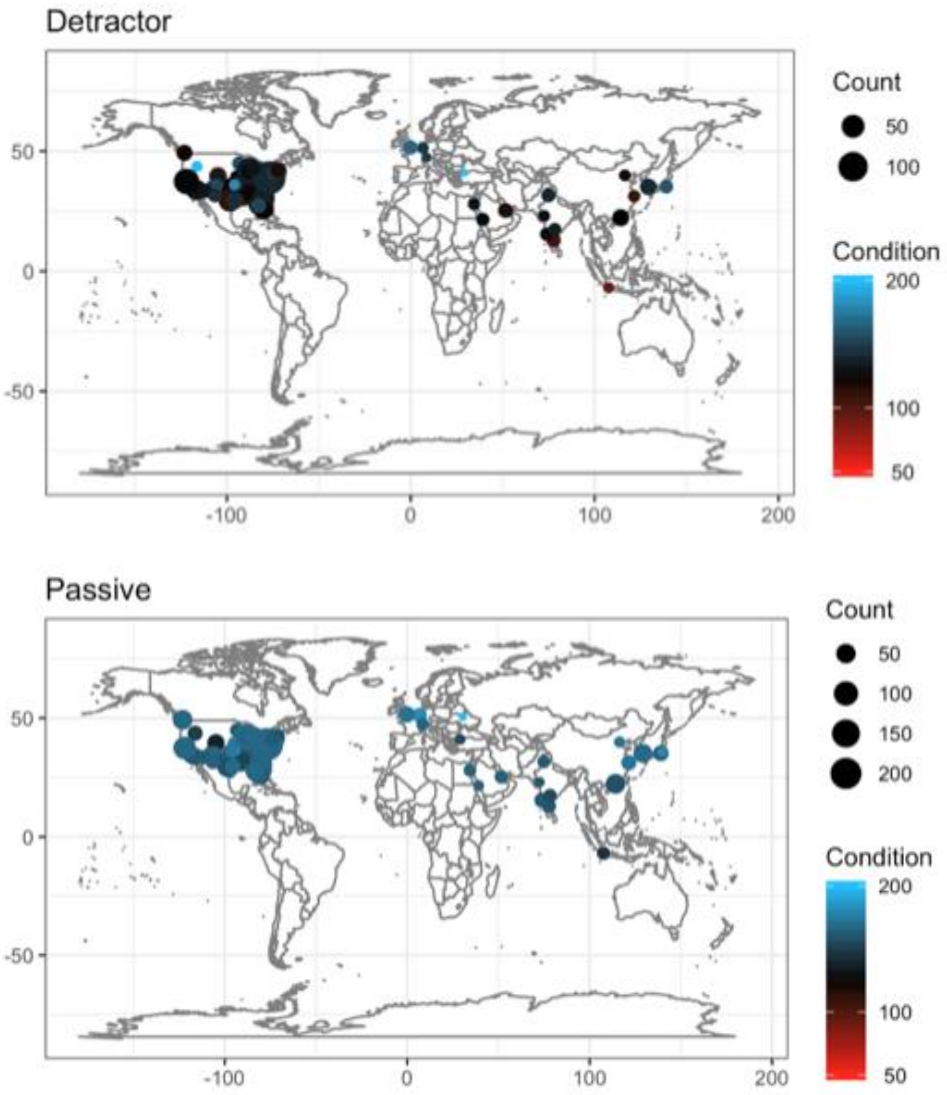
Gender/Age Plot



Which hotels/locations are performing better?

We were interested in understanding relationships between hotel conditions and customer satisfactions by geographic locations. First, we created new data frames by each NPS category (detractor, passive & promoter). In each NPS category, we counted the number of survey participants by each location. Then, we calculated average hotel condition scores by each location. Lastly, we created three world maps to display the location of detractors, passives and promoters and their average hotel condition scores.

Based on these observations, we found there were strong relationships between hotel conditions and customer satisfaction in any locations. Although a few locations in the US and Europe had reversed relationships (high average hotel condition scores in the detractor group), in the most locations, the promoter group had the highest average hotel condition scores, the passive group was the next, and the detractor group was the lowest.



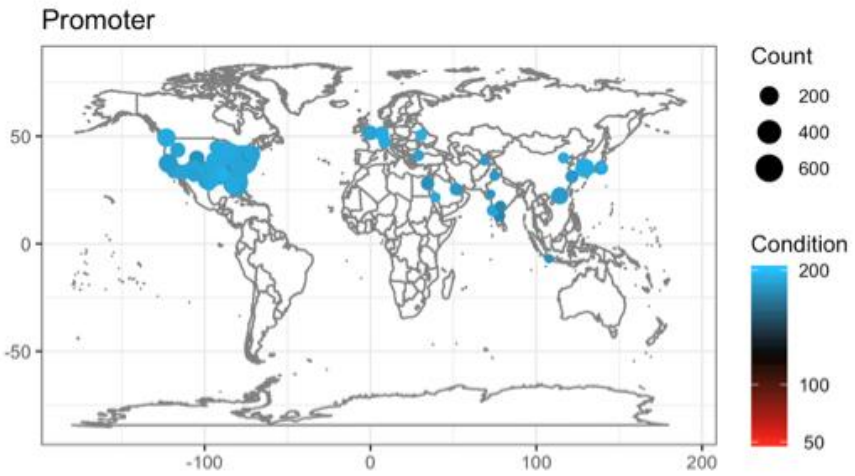


Table A: Top 3 cities with Highest Numbers of Promoters.

Location	Numbers of Promoters	Mean Condition
Washington DC	724	190.6
Orlando, FL	465	190.2
Jacksonville, FL	438	188.4

Table B: Top 3 cities with Highest Mean Hotel Condition.

Location	Numbers of Promoters	Mean Condition
Corpus Christi, TX	62	194.4
Dallas Fort Worth Airport, TX	203	196.7
Lithonia, GA	28	196.4

We observed that the U.S. is the country with a larger number of survey completed. The top three cities with the highest numbers of promoters were Washington DC, Orlando and Jacksonville (See table A above). On the other hand, Corpus Christi, Dallas Fort Worth Airport,

and Lithonia had the highest mean condition scores. However, there were low numbers of promoters (See table B above). We concluded Washington DC was the highest performing location.

For cities with high mean hotel conditions but low number of responses and promoters, we would like to understand if they are high performing too. On the other hand, Mystic CT, Secaucus NJ, Zurich in Switzerland had the lowest hotel condition scores, but only two detractors' responses in each city when we should have expected more.

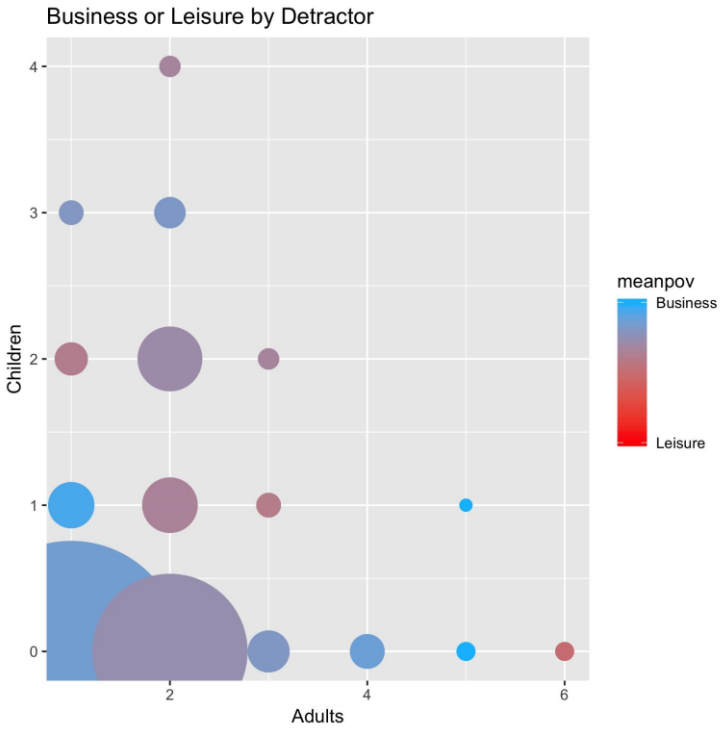
Currently, most of the completed surveys come from U.S. so we look to collect more survey data from all over the world to develop more international insights for the company. With this additional data, we may find potential reasons why several locations in the U.S. and Europe had good hotel conditions in the detractor group.

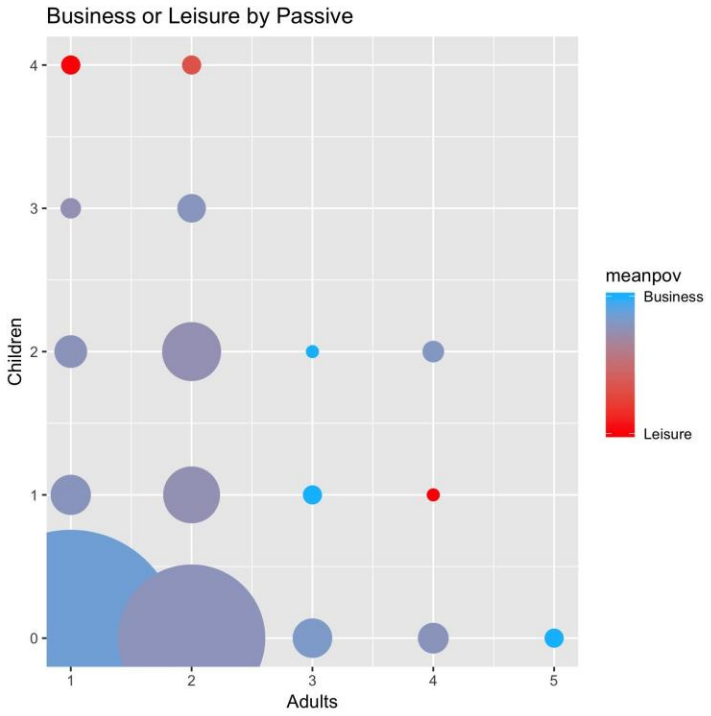
Do business and leisure travelers have different NPS type?

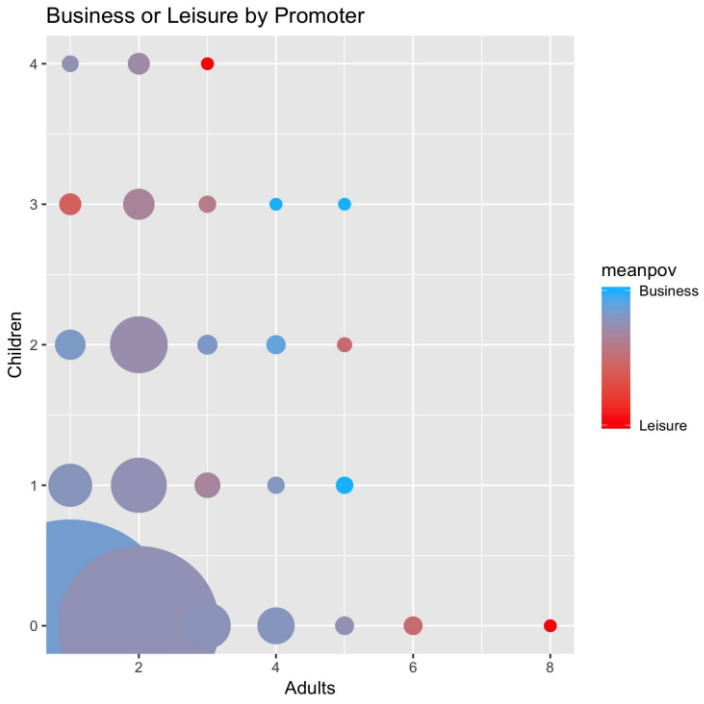
We explored the difference between business and leisure traveler by NPS type. We found in the total survey participants, over 80% of customers stayed at a hotel for business purpose. 50.0% of business travelers stayed at a hotel alone, and 6.5% of business travelers stayed at a hotel with other adults and children. We observed patterns if numbers of adults and children increased, more customers stayed at a hotel for leisure.

Based on the NPS type, 51.3% of the detractor group stayed at a hotel alone for business purposes, the passive group with 53.7% and the promoter group with 48.8%. The detractors and the passive group had slightly higher percentages than the promoter group. We observed that the percentages of leisure travelers in each NPS type were similar, the detractor group with 19.1%, the passive group with 17.4% and the promoter group with 19.3%. However, when family size increased, we saw more leisure travelers in the passive and the promoter group than in the detractor group. A family with 3 or more adults and 1 or more children, the promoter group with 37.2%, the passive group with 37.5% and the detractor group with 21.1%.

Based on these findings, we assumed that one business traveler tends to be lower satisfaction. On the other hand, leisure travelers with large numbers of family tend to be higher satisfaction. Overall, 80% of customers were business travelers and if the numbers of business and leisure travelers were closer to even we could see clear tendencies based on family size differences.



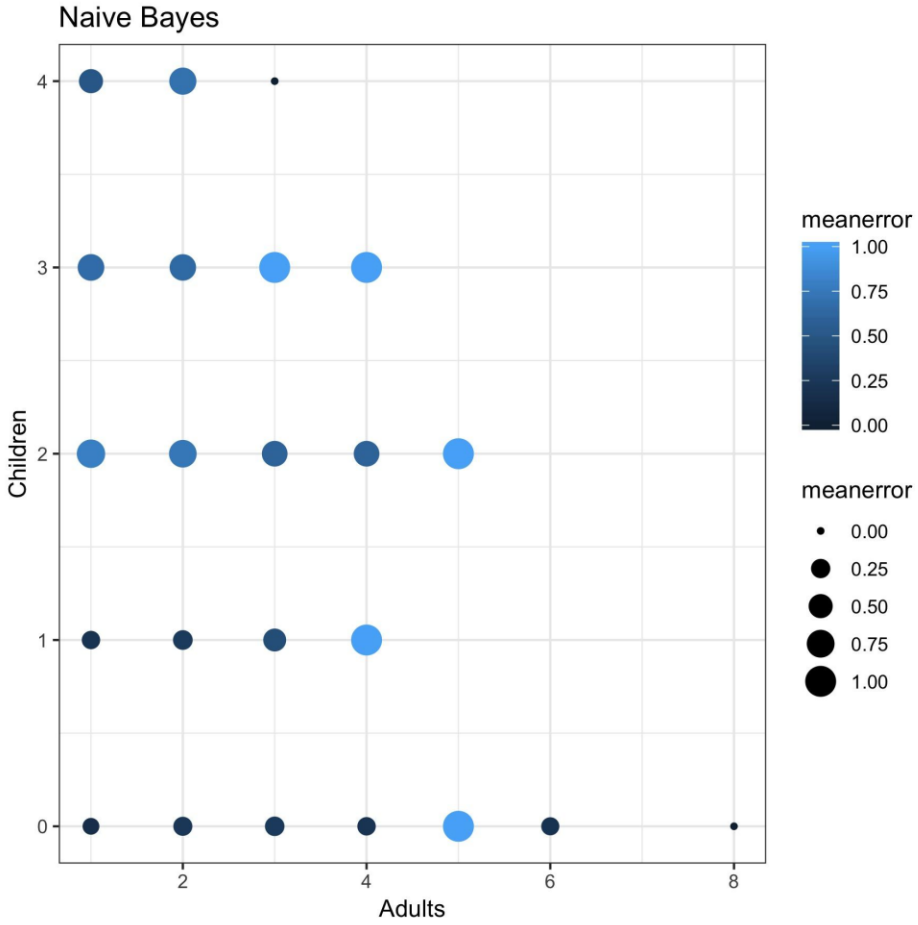




We used Naive Bayes method to do the prediction for the purpose of hotel stay based on the number of adults and children. First, we assigned numbers representing POV code business =

0 and leisure =1. We randomly chose two thirds of data as train dataset and the rest as test dataset.

We found the results with 79.2 % accuracy. The accuracy dropped when the number of children increased. Clear differences in the accuracies between zero or one child and two or more children were observed. Since 84.9% of travelers stayed at a hotel alone were business purpose, the tendency of the prediction for the travelers stayed at a hotel alone would be business as the basis for the Naive Bayes model.



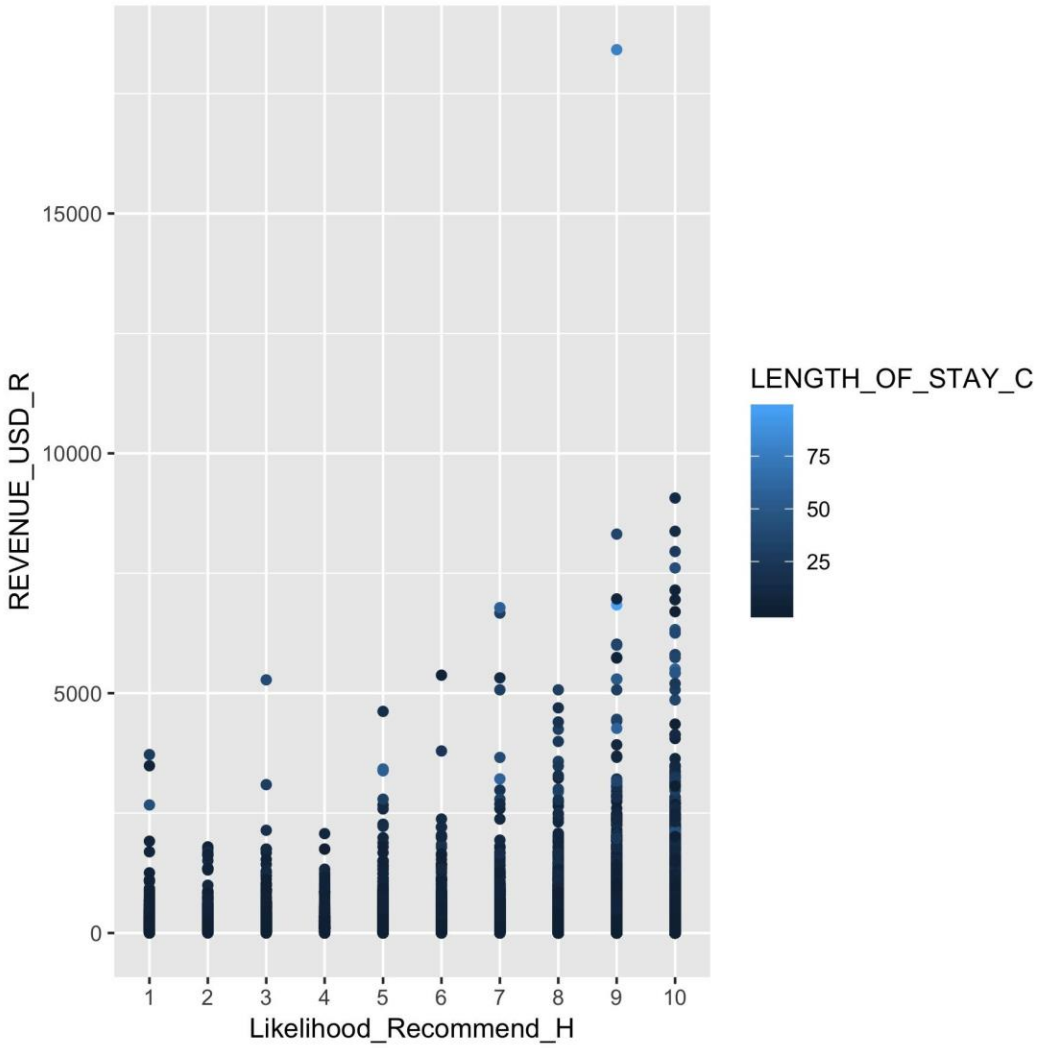
Is it possible to predict NPS score using the revenue generated?

In this section we analyzed if we could predict NPS score using the revenue generated. We did two types of analysis for this business question. First we used both revenue generated and the length of stay by the guest. Second, we used both revenue generated and room type. We did the analysis using the svm prediction technique.

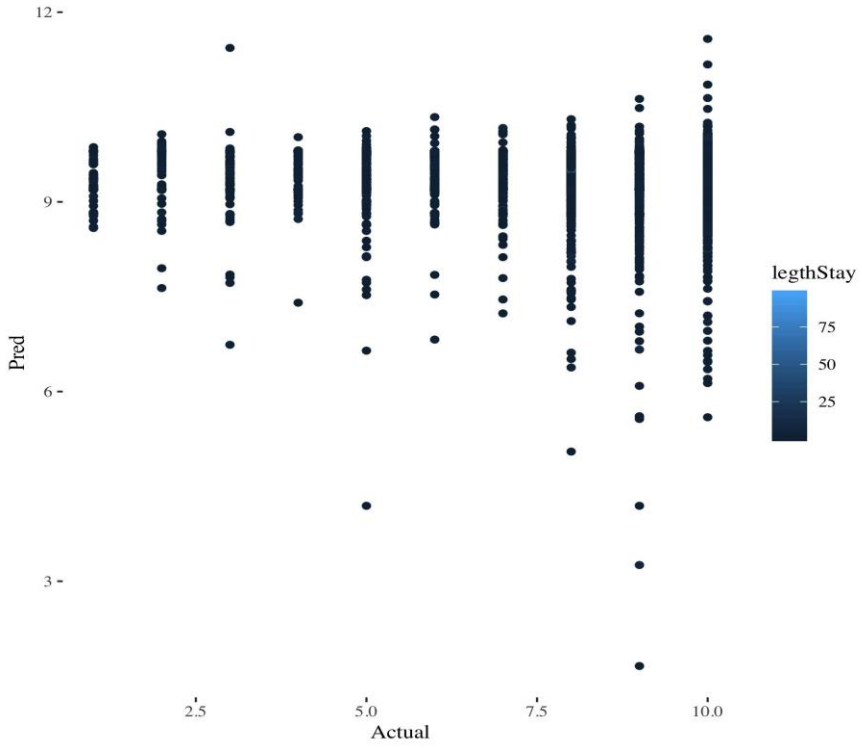
When we did the svm prediction based on revenue and length of stay, we found that the prediction is better for higher ranges of NPS score (promoters) and it was within 2 RMSE. Similarly results were seen for prediction based on revenue and room type.

When we look at the plots below, we find that Higher length of stay does not mean better NPS score. Similarly the type of room is irrelevant for predicting NPS score. For the svm prediction, we created a training and testing dataset by dividing the entire dataset in a 2/3 and 1/3 ratio. After calculating the predicted value, we compared it with the actual value by using the square root mean error rate.

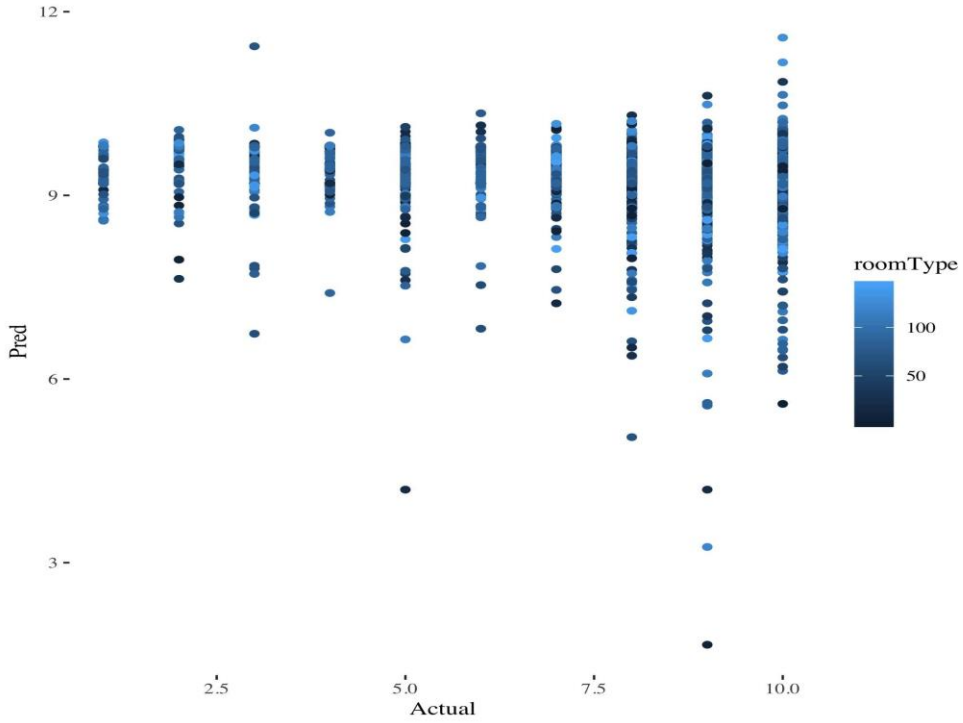
For both the models the RMSE was 2. The visual inspection shows that we can predict the NPS score better for the promoters and passives.



Actual vs Predicted NPS score using revenue and length of stay



Actual vs Predicted NPS score using revenue and room type



Which Service factors are the best predictors of promoters? Focusing on improving those Service factors, what are the top 5 hotel geolocation regions to target?

To evaluate the service factors that predicted promoters, we used an SVM machine learning models to classify and find best service predictors that have a prediction value greater than 80%. Next, we focused in on the detractors with the lowest scores for those attributes in a given geo region. Based on the findings, the intent will be to make recommendations that Hyatt focus their Service improvement initiatives on the best 3 predictors of promoters and narrow their tactics to start with the top 5 geolocation regions with the highest density of hotels. This will minimize improvement costs and maximize NPS lift potential, that currently sits at 59%.

Based on our analysis, we found that of the seven different service factor types, Guest Room Condition, Customer Service, and Hotel Condition service ratings were the best predictors of NPS types, Promoters versus Detractors. Combined they yield an 86.68% accuracy rate (see Table 1.0 below for details). Our analysis further found that 93% of the hotels with these three service factors as detractors reside within the United States. When narrowing down to the Top 5 U.S. States with the highest density of hotels with detractors, we found that TX, CA, FL, NC, and NY had the most detractors (see Table 2.0 below for details). Hyatt has an opportunity to improve in these service areas across these states, as well as should focus and invest in maintaining high levels of conditions and customer service throughout their hotels.

Data Tables

Table 1.0: SVM Accuracy Rating by Service Type

SVM Accuracy Rating by Service Type				
Service Type	Guest Room	Customer Service	Hotel Condition	Combined
Accuracy Rating	83.08%	83.02%	83.46%	86.68%

Table 2.0: Top 5 States with highest density of Hotels identified as Detractors

State	Guest Room Avg	Customer Service Avg	Hotel Condition Avg	Total Hotel Count
Texas	3.86%	3.87%	4.13%	125
California	3.93%	3.98%	4.05%	86
Florida	4.28%	4.51%	4.06%	68
North Carolina	3.60%	4.45%	4.16%	45
New York	4.03%	4.34%	4.15%	37

Table 3.0: By Country, Hotel counts of Service factor Detractors

Hotel Conditions Ratings Detractors - Country, Hotel Counts

	1	2	3	4	5	6	7	8	9	10
Canada	2	0	0	1	1	2	0	0	0	0
China	0	0	1	0	6	2	0	0	0	0
Egypt	0	1	0	0	0	1	0	0	0	0
Germany	0	0	1	0	0	1	0	0	0	0
Hong Kong	0	1	3	0	3	4	0	0	0	0
India	3	3	1	3	6	1	0	0	0	0
Indonesia	0	0	0	0	0	0	0	0	0	0
Japan	0	0	0	0	1	0	0	0	0	0
Mexico	0	0	1	1	1	0	0	0	0	0
Qatar	0	0	2	1	1	1	0	0	0	0
Saudi Arabia	2	0	0	1	3	0	0	0	0	0
South Korea	0	2	3	0	2	3	0	0	0	0
Switzerland	0	0	0	0	0	1	0	0	0	0
Tajikistan	0	0	0	0	0	0	0	0	0	0
Turkey	0	0	0	0	0	1	0	0	0	0
Ukraine	0	0	0	0	0	0	0	0	0	0
United Kingdom	0	0	0	0	0	0	0	0	0	0
United States	82	80	134	123	230	194	0	0	0	0

Customer Service Detractors - Country, Hotel Counts

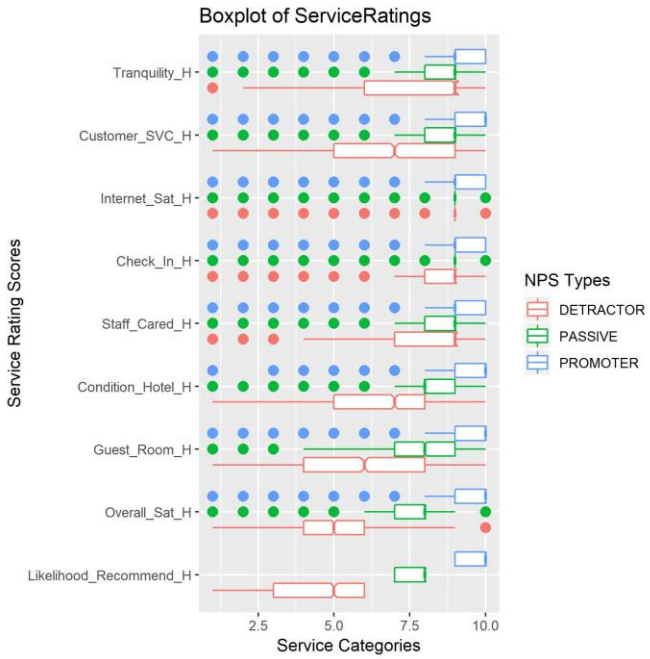
	1	2	3	4	5	6	7	8	9	10
Canada	0	2	0	1	3	2	0	0	0	0
China	0	1	2	0	3	1	0	0	0	0
Egypt	0	0	0	0	1	2	0	0	0	0
Germany	0	0	0	0	0	1	0	0	0	0
Hong Kong	0	2	1	0	3	3	0	0	0	0
India	1	1	2	0	9	4	0	0	0	0
Indonesia	0	0	0	2	0	1	0	0	0	0
Japan	0	0	1	0	0	0	0	0	0	0
Mexico	0	0	1	1	0	1	0	0	0	0
Qatar	2	1	0	1	0	1	0	0	0	0
Saudi Arabia	1	0	0	0	2	0	0	0	0	0
South Korea	0	0	0	3	1	2	0	0	0	0
Switzerland	0	0	0	0	1	0	0	0	0	0
Tajikistan	0	0	0	0	0	0	0	0	0	0
Turkey	0	0	0	0	0	0	0	0	0	0
Ukraine	0	0	0	0	0	0	0	0	0	0
United Kingdom	0	0	0	0	0	1	0	0	0	0
United States	62	100	114	131	246	213	0	0	0	0

Guest Room Condition Detractors - Country, Hotel Counts

	1	2	3	4	5	6	7	8	9	10
Canada	0	0	2	1	0	2	0	0	0	0
China	1	2	2	1	1	0	0	0	0	0
Egypt	1	1	0	1	0	0	0	0	0	0
Germany	0	0	0	0	0	1	0	0	0	0
Hong Kong	1	1	3	1	0	2	0	0	0	0
India	1	4	3	4	6	3	0	0	0	0
Indonesia	0	0	0	0	2	1	0	0	0	0
Japan	0	0	1	0	0	0	0	0	0	0
Mexico	0	0	1	1	1	0	0	0	0	0
Qatar	0	1	1	0	3	2	0	0	0	0
Saudi Arabia	0	0	0	1	2	1	0	0	0	0
South Korea	0	0	1	3	1	2	0	0	0	0
Switzerland	0	0	1	0	0	0	0	0	0	0
Tajikistan	0	0	0	0	0	0	0	0	0	0
Turkey	0	0	0	0	0	0	0	0	0	0
Ukraine	0	0	0	0	0	0	0	0	0	0
United Kingdom	1	0	0	1	1	0	0	0	0	0
United States	91	120	151	204	307	233	0	0	0	0

Visualizations, Service factors Data Analysis

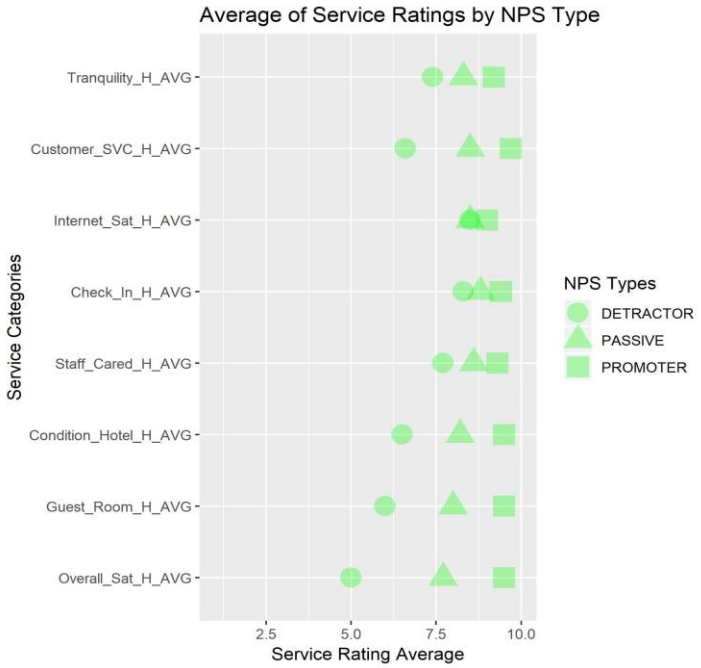
Boxplot of Service Rating Average Scores by NPS Type




```
Likelihood_Recommend Likelihood_Recommend_H Overall_Sat_H    Guest_Room_H    Condition_Hotel_H Staff_Cared_H    Check_In_H     Internet_Sat_H  Customer_SVC_H  Tranquility_H
Length:19342         Min.   : 1.0           Min.   : 1.000   Min.   : 1.000   Min.   : 1.00     Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  Min.   : 1.000  Min.   : 1.00
Class :character     1st Qu.: 8.0           1st Qu.: 8.000   1st Qu.: 8.000   1st Qu.: 8.00     1st Qu.: 9.000   1st Qu.: 9.000   1st Qu.: 9.000  1st Qu.: 9.000  1st Qu.: 9.00
Mode  :character     Median : 9.0           Median : 9.000   Median : 9.000   Median : 9.00     Median : 9.000   Median : 9.000   Median : 9.000  Median :10.000  Median : 9.00
                     Mean   : 8.7           Mean   : 8.667   Mean   : 8.794   Mean   : 8.94     Mean   : 9.016   Mean   : 9.159   Mean   : 8.856  Mean   : 9.096  Mean   : 8.86
                     3rd Qu.:10.0           3rd Qu.:10.000   3rd Qu.:10.000   3rd Qu.:10.00     3rd Qu.:10.000   3rd Qu.:10.000   3rd Qu.: 9.000  3rd Qu.:10.000  3rd Qu.:10.00
                     Max.   :10.0           Max.   :10.000   Max.   :10.000   Max.   :10.00     Max.   :10.000   Max.   :10.000   Max.   :10.000  Max.   :10.000  Max.   :10.00
```

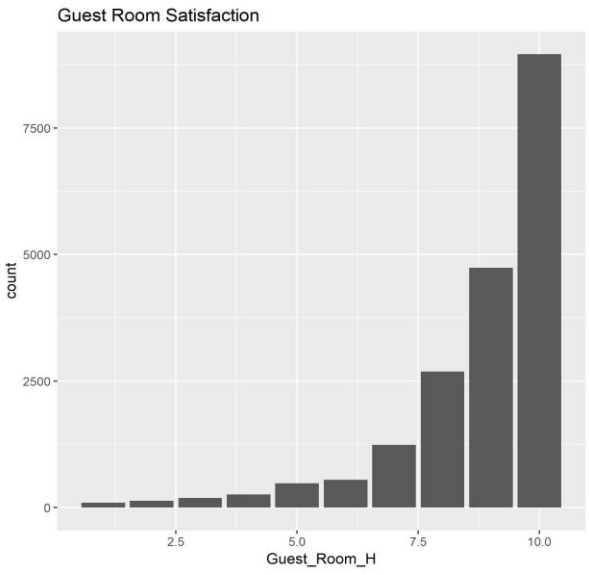
Likelihood_Recommend	Overall_Sat_H_AVG	Guest_Room_H_AVG	Condition_Hotel_H_AVG	Staff_Cared_H_AVG	Check_In_H_AVG	Internet_Sat_H_AVG	Customer_SVC_H_AVG	Tranquility_H_AVG	NPS_TYP_CNT	NPS_TYP_PERC
DETRACTOR	5.0	6.0	6.5	7.7	8.3	8.5	6.6	7.4	2169	11.21394
PASSIVE	7.7	8.0	8.2	8.6	8.8	8.5	8.5	8.3	3603	18.62786
PROMOTER	9.5	9.5	9.5	9.3	9.4	9.0	9.7	9.2	13570	70.15820

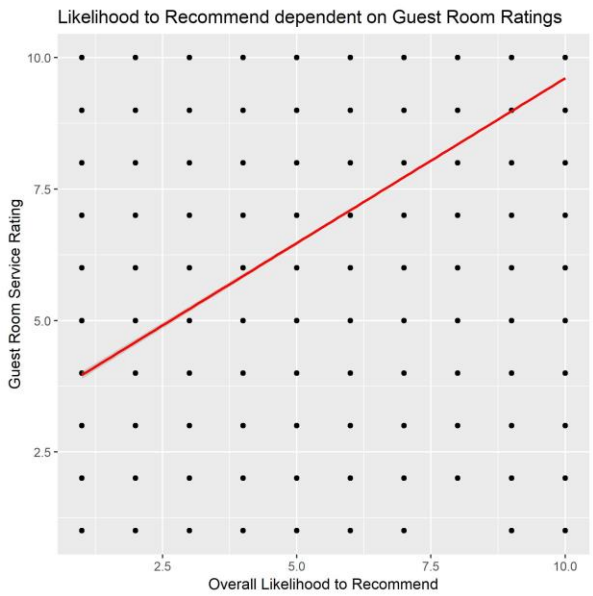
Point plot of Service Rating Averages per Service type, shaped by NPS Type



Visualizations, Best 3 Service factors for predicting promoters

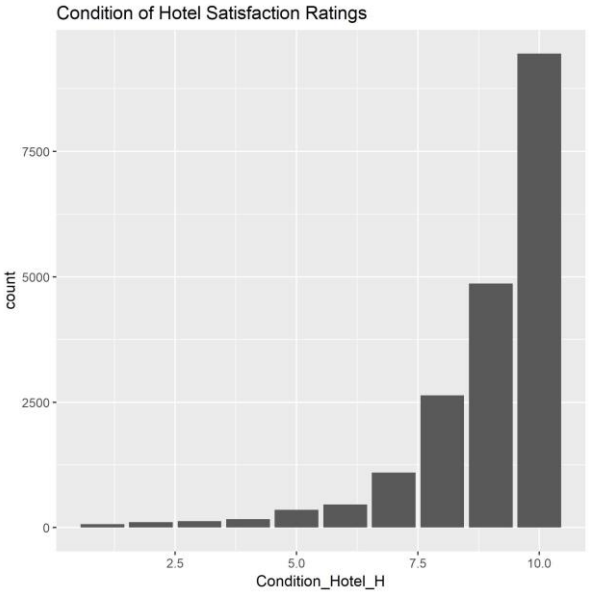
Plots of Guest Room Service Rating

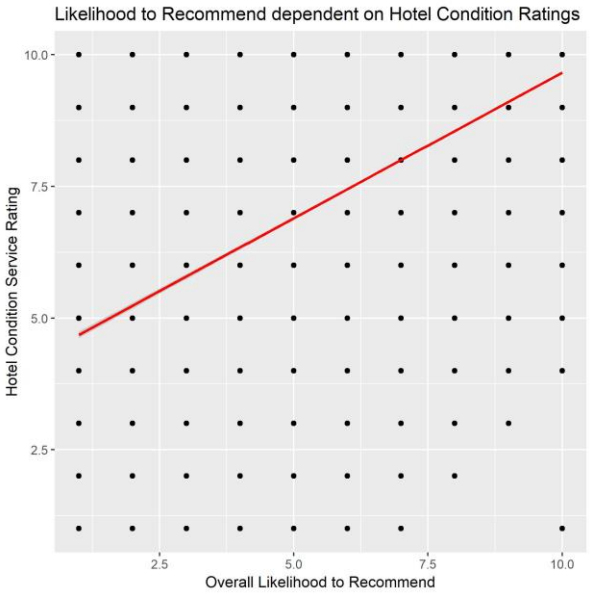




*SVM Accuracy Rate: 83.08% | Error Rate: 16.92%

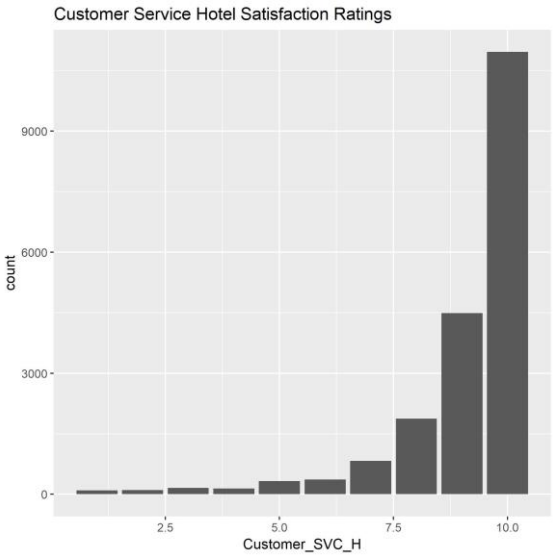
Plots of Hotel Condition Satisfactory Rating





*SVM Accuracy Rate: 83.46% | Error Rate: 16.54%

Plots of Customer Service Satisfactory Rating

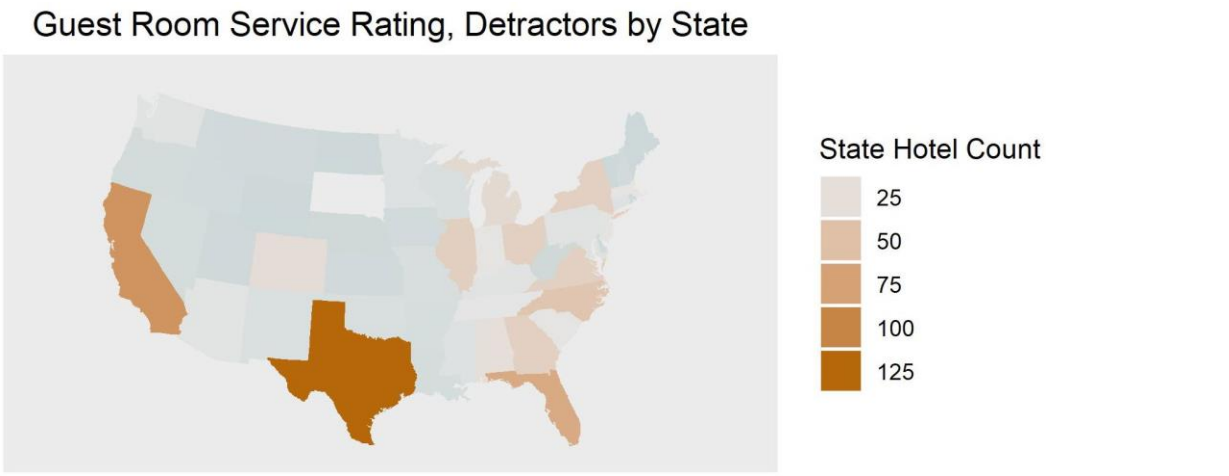


*SVM Accuracy Rate: 83.02% | Error Rate: 16.98%

Visualizations, Detractors by U.S. States

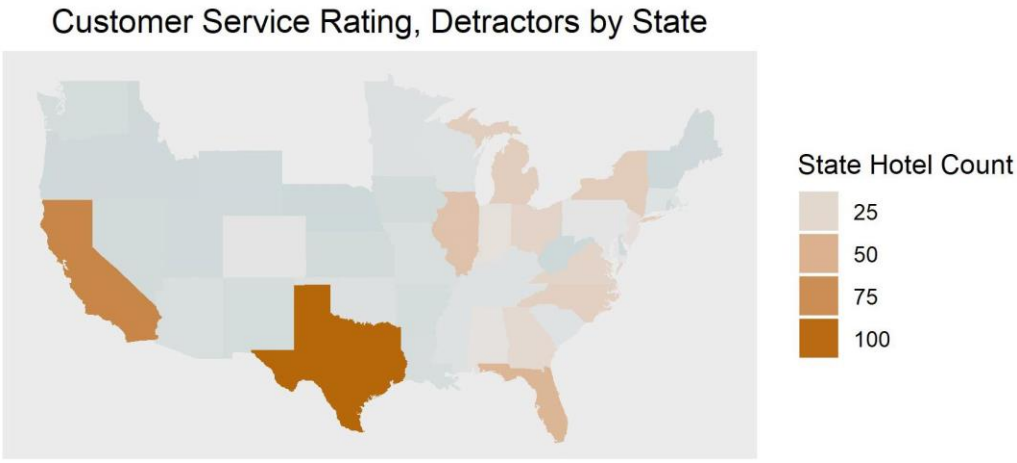
*Note: Color fill of US States uses the ‘scale_fill_gradient2’ attribute showing how the state hotel count average diverge.

State Map Plot of Guest Room Service Rating, Density by State



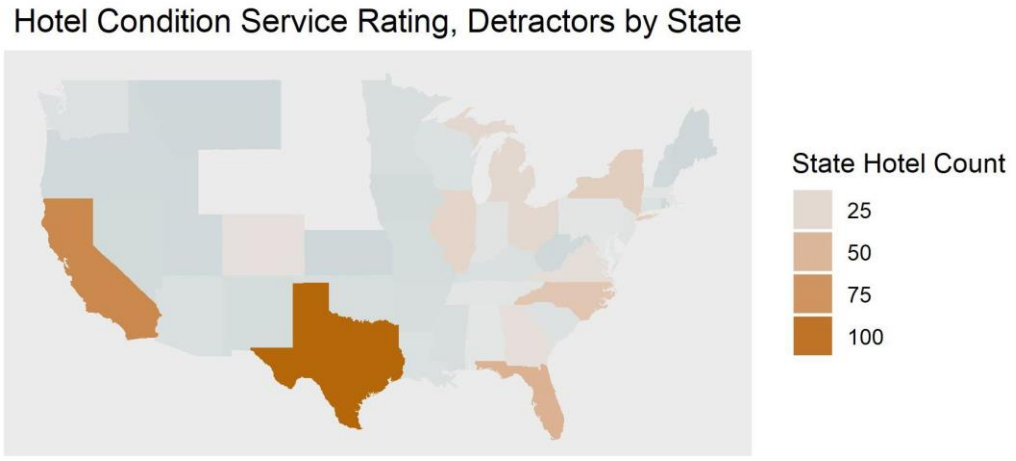
STATE_R	Hotel_Count	Guest_Room_Avg	Detractor_Low_Cnt	Detractor_Mid_Cnt	Detractor_High_Cnt
TX	125	3.86	29	41	55
CA	86	3.93	20	29	37
FL	68	4.28	8	23	37
NC	45	3.60	11	19	15
IL	37	4.54	3	12	22

State Map Plot of Customer Service Rating, Density by State



STATE_R	Hotel_Count	CustomerSVC_Avg	Detractor_Low_Cnt	Detractor_Mid_Cnt	Detractor_High_Cnt
TX	102	3.87	20	41	41
CA	80	3.98	16	28	36
FL	47	4.51	4	16	27
IL	39	3.67	13	7	19
NY	32	4.34	7	4	21

State Map Plot of Hotel Condition Service Rating, Density by State



STATE_R	Hotel_Count	Condition_Avg	Detractor_Low_Cnt	Detractor_Mid_Cnt	Detractor_High_Cnt
TX	109	4.13	19	36	54
CA	83	4.05	21	20	42
FL	53	4.06	9	20	24
NC	38	4.16	5	15	18
NY	33	4.15	6	10	17

Conclusion:

Next steps and business recommendations

From the data, we gained a variety of insights on the Hyatt business. There are areas for improvement as well as more open questions to evaluate further.

We learned that the majority of stays are short, and average 3 days or less across 17 of the 18 countries Hyatt operates in. NPS scores did not have significant variation across the length of stay. Additionally, the majority of surveys came from travelers who were on business or ones without children.

We recommend that we increase our offerings that make strong first impressions and cater to short term stays. This could include ideas such as flexible or early check in offerings, and different welcoming packages like having complimentary food or drinks when a guest arrives. For example, for morning arrivals we could offer a coffee and bagel upon check in, and with evening arrivals we could offer a complimentary drink, such as wine. An early check in service offering can cater an offering to business travelers who may arrive at a variety of times. This is

something we could do a pricing experiment with to try to maximize revenue from guests during their short stays. We could test the trade off of increasing NPS score vs. increasing revenue with an early check in offering. Also, we could make flexible check in a feature or benefit of a Hyatt Rewards program that is aimed at improving customer loyalty and retention. This way we could try to increase the frequency of how often the traveler stays throughout the year.

From an operations perspective, we can look at how can evaluate if we are maximizing our room utilization and operations. Are the utilization and efficiency in which we are using the total number of our rooms optimized for the profile of our guests? Is there something in how the hotel can focus resources to get rooms ready that increase the booking capacity since there is high guest turnover with stays of 3 days or less? We would work with the hotel operations to determine if there are any efficiencies to be gained in room utilization and booking.

For demographics, we did not see any major differences between NPS scores from males and female survey responders. However, we did see some different with age in that the 46-55 group had a higher amount of promoters than the lower age groups. We believe there may be an opportunity to penetrate the younger demographics and find opportunities to improve their NPS averages. We would want to use other data collection techniques, such as focus groups, to better understand the preferences by age. Could the hotel have different amenities that target younger demographics. Are there different brands, local or national, that would make sense to partner with. For example, do we need improved WiFi service? what if Netflix was available in each room? Can we partner with Uber to have discounted rides to hotel locations? Are there local venues or events that Hyatt could partner with?

Since hotel condition is of value to the customers, we should focus on this trait especially for hotels that have lower condition scores. Additionally, this indicates that the condition is a priority for customers so we can evaluate increasing pricing to get additional revenue to fund increasing the hotel condition. Our assumption is that a customer that is more focused on condition is going to be *less* price conscious since their NPS score is correlated with the condition. If we did not see a relation between NPS score and hotel condition, we would have looked at whether customers valued lower pricing the most.

Geographically, we saw more data in the US markets than we did internationally. There was a higher rate of response. The Washington DC location had the highest amount of promoters and it is a top performing location. After diving deeper into the US data, we saw that guest room conditions, hotel condition and customer service were the key three service factor detractors for the market to focus on. Many major markets, such as the states of CA, NY, TX, NC, and FL have opportunities to improve NPS through these service factors.

Since Hyatt is a global company, we would work on an initiative to expand data collection to understand the international markets better. The mean condition score was an important factor for NPS, but we could benefit from more data to make more accurate assessments across the markets. We would recommend more international data collection efforts.

We saw strong results from leisure travel and with family size, but a lower percentage of responses. This indicates that we could look to penetrate the family and leisure travel market more to make them a larger percentage of the customer base. Based on the NPS scores, we see that they enjoy their stay at the hotel at a higher rate. We would want to evaluate our marketing message and opportunity to increase leisure travel promotions. Additionally, we should understand if this is a seasonal trend for leisure travel. Do we see more leisure travel during certain times of the year, such as holidays, summer or weekends? Does this vary by market and country?

Overall, we gained insights on Hyatt hotels and have actions to pursue, but it will be an ongoing process of testing and analysis from our original recommendations and proposed ideas. It will be important for our data scientist and analysis team work cross functionally across the different teams at Hyatt to implement a plan and iterate on the findings from new data and analysis.

Data Cleansing, Munging, and Preparation:

Please see below for the code used to transform the data set:

```
##


# ---Preprocess Steps:------------------------------------------------------------------

### Clear objects from Memory
rm(list=ls())
### Clear Console:
cat("\014")

##---- Global Variables -------------------------------------
dataSetName <- "ProjectSurveyData.csv"
workingDir <- "C:\\workspaces\\ms_datascience_su\\IST687-
IntroDataScience\\R_workspace\\project\\"
### Set Working Directory
setwd(workingDir)

##---- Required Libraries -----------------------------------
if(!require("stringr")){install.packages("stringr")}
if(!require("dplyr")) {install.packages("dplyr")}
if(!require("lubridate")) {install.packages("lubridate")}
if(!require("sqldf")){install.packages("sqldf")}
if(!require("ggplot2")){install.packages("ggplot2")}
```

```
if(!require("psych")){install.packages("psych")}
if(!require("reshape2")){install.packages("reshape2")}

#if(!require("RColorBrewer")){install.packages("RColorBrewer")}

# ----Step 1: Data Preparation Steps----------------------------------------------------------
### Function: Read DataSet
readDataSet <- function(ds){

  survey.ds <- read.csv(ds, header = TRUE, stringsAsFactors = FALSE)
  return(survey.ds)

}

## Function: Clean DataSet
### Replace NA with column means
na.2.mean <- function(x){
  replace(x, is.na(x), mean(x, na.rm = TRUE))
}

cleanDataSet <- function(ds){

  #Make all empty cells equal to NA
  ds[ds==""] <- NA

  #Clean NA Columns from Dataframe
  ds <- ds[ ,!apply(ds,2,function(x) all(is.na(x)))]

  #Clean empty Rows from Dataframe
  ds <- ds[!apply(ds,1,function(x) all(is.na(x))),]

  #Set Guest Country NA to 'NOT_LISTED'
  ds$Guest_Country_H[is.na(ds$Guest_Country_H)] <- 'NOT_LISTED'
  ds$Guest_Country_H[ds$Guest_Country_H=="United States Minor "] <- 'USA'

  #Set Character NA to 'NOT_LISTED'
  ds$Club.Type_PL[is.na(ds$Club.Type_PL)] <- 'NOT_LISTED'
  ds$STATE_R[is.na(ds$STATE_R)] <- 'NOT_LISTED'
  ds$GP_Tier[is.na(ds$GP_Tier)] <- 'NOT_LISTED'


  #Set Gendr to 'PNTA' where it's 'Prefer not to answer' or NA
  ds$Gender_H[is.na(ds$Gender_H) | ds$Gender_H == 'Prefer not to answer'] <- 'PNTA'
```

```
#Replace column NA with mean values
ds$LENGTH_OF_STAY_C <- round(na.2.mean(ds$LENGTH_OF_STAY_C))
ds$NUMBER_OF_ROOMS_C <- round(na.2.mean(ds$NUMBER_OF_ROOMS_C))
ds$ADULT_NUM_C <- round(na.2.mean(ds$ADULT_NUM_C))
ds$F.B_Overall_Experience_H <- round(na.2.mean(ds$F.B_Overall_Experience_H))
ds$F.B_FREQ_H <- round(na.2.mean(ds$F.B_FREQ_H))
ds$Check_In_H <- round(na.2.mean(ds$Check_In_H))
ds$Internet_Sat_H <- round(na.2.mean(ds$Internet_Sat_H))
ds$Staff_Cared_H <- round(na.2.mean(ds$Staff_Cared_H))
ds$Customer_SVC_H <- round(na.2.mean(ds$Customer_SVC_H))
ds$Condition_Hotel_H <- round(na.2.mean(ds$Condition_Hotel_H))
ds$Tranquility_H <- round(na.2.mean(ds$Tranquility_H))
ds$Guest_Room_H <- round(na.2.mean(ds$Guest_Room_H))
ds$Overall_Sat_H <- round(na.2.mean(ds$Overall_Sat_H))

#Set Variables as Factors
ds$MARKET_GROUP_C <- as.factor(ds$MARKET_GROUP_C)
ds$ROOM_TYPE_CODE_C <- as.factor(ds$ROOM_TYPE_CODE_C)
ds$WALK_IN_FLG_C <- as.factor(ds$WALK_IN_FLG_C)
ds$POV_CODE_C <- as.factor(ds$POV_CODE_C)
ds$ENTRY_HOTEL_CODE_R <- as.factor(ds$ENTRY_HOTEL_CODE_R)
ds$ROOM_TYPE_CODE_R <- as.factor(ds$ROOM_TYPE_CODE_R)
ds$NPS_Type <- as.factor(ds$NPS_Type)
ds$Booking_Channel <- as.factor(ds$Booking_Channel)
ds$GP_Tier <- as.factor(ds$GP_Tier)
ds$Relationship_PL <- as.factor(ds$Relationship_PL)
ds$Location_PL <- as.factor(ds$Location_PL)
ds$Class_PL <- as.factor(ds$Class_PL)
ds$Type_PL <- as.factor(ds$Type_PL)
ds$Category_PL <- as.factor(ds$Category_PL)
ds$Region_PL <- as.factor(ds$Region_PL)
ds$Brand_PL <- as.factor(ds$Brand_PL)
ds$Dom.Int.l_PL <- as.factor(ds$Dom.Int.l_PL)
ds$Currency_PL <- as.factor(ds$Currency_PL)
ds$Ops.Region_PL <- as.factor(ds$Ops.Region_PL)
ds$Country_PL <- as.factor(ds$Country_PL)
ds$Award.Category_PL <- as.factor(ds$Award.Category_PL)
ds$F.B_Overall_Experience_H <- as.factor(ds$F.B_Overall_Experience_H)
ds$Check_In_H <- as.factor(ds$Check_In_H)
ds$Internet_Sat_H <- as.factor(ds$Internet_Sat_H)
ds$Staff_Cared_H <- as.factor(ds$Staff_Cared_H)
ds$Customer_SVC_H <- as.factor(ds$Customer_SVC_H)
ds$Condition_Hotel_H <- as.factor(ds$Condition_Hotel_H)
ds$Tranquility_H <- as.factor(ds$Tranquility_H)
```

```
  ds$Guest_Room_H <- as.factor(ds$Guest_Room_H)
  ds$Overall_Sat_H <- as.factor(ds$Overall_Sat_H)
  ds$Likelihood_Recommend_H <- as.factor(ds$Likelihood_Recommend_H)
  ds$Language_H <- as.factor(ds$Language_H)
  ds$Gender_H <- as.factor(ds$Gender_H)
  ds$Club.Type_PL <- as.factor(ds$Club.Type_PL)
  ds$STATE_R <- as.factor(ds$STATE_R)
  ds$PACE_CATEGORY_R <- as.factor(ds$PACE_CATEGORY_R)

  #Convert Character Date's to Data Type: convert date info in format 'mm/dd/yyyy'
  ds$CHECK_IN_DATE_C <- as.Date(ds$CHECK_IN_DATE_C,"%m/%d/%Y")
  ds$CHECK_OUT_DATE_C <- as.Date(ds$CHECK_OUT_DATE_C,"%m/%d/%Y")
  ds$RESERVATION_DATE_R <- as.Date(ds$RESERVATION_DATE_R,"%m/%d/%Y")
  ds$LAST_CHANGE_DATE_R <- as.Date(ds$LAST_CHANGE_DATE_R,"%m/%d/%Y")

  #Convert Character Time's to POSIXct format - Use lubridate later for analysis
  ds$ENTRY_TIME_R <- as.POSIXct(paste("2019-01-01",ds$ENTRY_TIME_R, sep = " "),
tz="UTC")

  #Set CHILDREN NUM Count to zero where it's NA
  ds$CHILDREN_NUM_C[is.na(ds$CHILDREN_NUM_C)] <- 0


  #Replace outliers


  return(ds)
}


# ---Step 1: Data Preparation Processing--------------------------------------------------------------------
### Read Data Set
surveyDataSet <- readDataSet(paste(workingDir,dataSetName, sep=""))
str(surveyDataSet)

### Clean Data Set
surveyDataSetCleaned <- cleanDataSet(surveyDataSet)
str(surveyDataSetCleaned)
head(surveyDataSetCleaned)

## Step 2: Data Exploration, Transformation and Feature Selection

## ----------------------------------------------------------------------------------------
## ----- Transformation Functions ----
```

```
## Classify Age Ranges, 1 through 8 -> NAs have been transformed to mean value of all age
range observations
normNAAgeRange <- function(age.class){
  range <- ""
  if(age.class==1){
    range <- "00-17"
  }else if(age.class==2){
    range <- "18-25"
  }else if(age.class==3){
    range <- "26-35"
  }else if(age.class==4){
    range <- "36-45"
  }else if(age.class==5){
    range <- "46-55"
  }else if(age.class==6){
    range <- "56-65"
  }else if(age.class==7){
    range <- "66-75"
  }else if(age.class==8){
    range <- "76+"
  }else{
    range <- "00-00"
  }
  return(range)
}

classifyAgeRanges <- function(ds){
  ds$Age_Range_H_Class <- NA

  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "00-17", 1,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "18-25", 2,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "26-35", 3,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "36-45", 4,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "46-55", 5,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "56-65", 6,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "66-75", 7,
Age_Range_H_Class))
```

```
  ds <- transform(ds, Age_Range_H_Class = ifelse(Age_Range_H == "76+", 8,
Age_Range_H_Class))
  ds <- transform(ds, Age_Range_H_Class = ifelse(is.na(Age_Range_H),
round(mean(ds$Age_Range_H_Class, na.rm = TRUE)), Age_Range_H_Class))

  #replace NA values in the Age_Range_H variable
  ds$Age_Range_H[is.na(ds$Age_Range_H)] <-
normNAAgeRange(round(mean(ds$Age_Range_H_Class)))

  return(ds)
}
## Classify Likely Hood To Recommend by NPS grouping -> Transforms DataSet by adding
classifier attributes for Promotor, Passive, Detractor
classifyLikelyhoodToRecommend_Class <- function(ds){
  ds$Likelihood_Recommend_Promotor <- NA
  ds$Likelihood_Recommend_Passive <- NA
  ds$Likelihood_Recommend_Detractor <- NA

  ds <- transform(ds, Likelihood_Recommend_Promotor = ifelse(Likelihood_Recommend_H ==
9 | Likelihood_Recommend_H == 10,1,0))
  ds <- transform(ds, Likelihood_Recommend_Passive = ifelse(Likelihood_Recommend_H == 7
| Likelihood_Recommend_H == 8,1,0))
  ds <- transform(ds, Likelihood_Recommend_Detractor =
ifelse(as.numeric(Likelihood_Recommend_H) <= 6,1,0))

  return(ds)
}
classifyLikelyhoodToRecommend_Type <- function(ds){
  ds$Likelihood_Recommend <- NA

  ds <- transform(ds, Likelihood_Recommend = ifelse(Likelihood_Recommend_H == 9 |
Likelihood_Recommend_H == 10,'PROMOTOR',ds$Likelihood_Recommend))
  ds <- transform(ds, Likelihood_Recommend = ifelse(Likelihood_Recommend_H == 7 |
Likelihood_Recommend_H == 8,'PASSIVE',ds$Likelihood_Recommend))
  ds <- transform(ds, Likelihood_Recommend = ifelse(as.numeric(Likelihood_Recommend_H)
<= 6,'DETRACTOR',ds$Likelihood_Recommend))

  return(ds)
}

## ----- Data Exploration Plotting Functions ----
singleBarPlot <- function(v){

}
```

```
#------ Data Exploration --------------------------------------------------------------------

### Print Column Names
colnames(surveyDataSetCleaned)
str(surveyDataSetCleaned)
head(surveyDataSetCleaned)

surveyDataSetCleaned <- classifyAgeRanges(surveyDataSetCleaned)

# Evaluate Attribute Values

##---- NPS - Key Driver ---------------------------
### Likelihood_Recommend_H
length(unique(surveyDataSetCleaned$Likelihood_Recommend_H))
sort(table(surveyDataSetCleaned$Likelihood_Recommend_H), decreasing = TRUE)
summary(surveyDataSetCleaned$Likelihood_Recommend_H)
describe(as.numeric(surveyDataSetCleaned$Likelihood_Recommend_H))

## Visualizations
ggplot(data=surveyDataSetCleaned) +
  geom_bar(mapping = aes(Likelihood_Recommend_H)) +
  ggtitle('Likelihood to Recommend')
ggsave(filename='Bar_of_Likelihood_Recommend_H.jpg', width = 6, height = 6)

surveyDataSetCleaned <- classifyLikelyhoodToRecommend_Type(surveyDataSetCleaned)
surveyDataSetCleaned$Likelihood_Recommend

ggplot(data=surveyDataSetCleaned) +
  geom_bar(mapping = aes(Likelihood_Recommend)) +
  ggtitle('Likelihood to Recommend by NPS Type')
ggsave(filename='Bar_of_Likelihood_Recommend_Type.jpg', width = 6, height = 6)

##---- SERVICE - Key Drivers ----------------------
service.attributes <-
c('Likelihood_Recommend','Likelihood_Recommend_H','Overall_Sat_H','Guest_Room_H','Cond
ition_Hotel_H','Staff_Cared_H','Check_In_H','Internet_Sat_H','Customer_SVC_H','Tranquility_H'
)
serviceDf <- surveyDataSetCleaned[service.attributes]
head(serviceDf)
summary(serviceDf)

serviceDf <- serviceDf %>%

mutate(Likelihood_Recommend_H=as.numeric(Likelihood_Recommend_H),Overall_Sat_H=as.
```

```
numeric(Overall_Sat_H), Guest_Room_H=as.numeric(Guest_Room_H),
Condition_Hotel_H=as.numeric(Condition_Hotel_H),
        Staff_Cared_H=as.numeric(Staff_Cared_H), Check_In_H=as.numeric(Check_In_H),
Internet_Sat_H=as.numeric(Internet_Sat_H),
Customer_SVC_H=as.numeric(Customer_SVC_H),
        Tranquility_H=as.numeric(Tranquility_H))
summary(serviceDf)

## Visualize Descriptive Statistics via Boxplot
ggplot(data=serviceDf, aes(x=factor(0), y=Likelihood_Recommend_H)) +
  geom_boxplot(fill="gray",outlier.colour="red",outlier.shape=16 ,outlier.size=3, notch=TRUE) +
  coord_flip(ylim=c(0,10)) +
  labs(title="Boxplot of Likelihood_Recommend_H", x="", y="Rating Score")
#theme_classic()
ggsave(filename="Boxplot_of_Likelihood_Recommend_H.jpg", width = 6, height = 6)

#service.ratings.df <- serviceDf[,-which(colnames(serviceDf)=="Likelihood_Recommend")]
service.ratings.melt <- melt(serviceDf,id="Likelihood_Recommend" )
g.box.service.ratings <- ggplot(service.ratings.melt) +
  geom_boxplot(aes(x=variable, y=value, color=Likelihood_Recommend), outlier.shape=16
,outlier.size=3, notch=TRUE) +
  coord_flip(ylim=c(1,10)) +
  labs(title="Boxplot of ServiceRatings", x="Service Rating Scores", y="Service Categories")
g.box.service.ratings
ggsave(filename="Boxplot_of_ServiceRatings.jpg", width = 6, height = 6)
##

# Promotors
promotorServiceDf <- sqldf("select * from serviceDf where Likelihood_Recommend =
'PROMOTOR'")
summary(promotorServiceDf)

# Passives
passiveServiceDf <- sqldf("select * from serviceDf where Likelihood_Recommend = 'PASSIVE'")
summary(passiveServiceDf)

# Detractors
detractorServiceDf <- sqldf("select * from serviceDf where Likelihood_Recommend =
'DETRACTOR'")
summary(detractorServiceDf)

## Plot Average scoring by NPS Type
service.ratings.avgs <- serviceDf %>%
  group_by(Likelihood_Recommend) %>%
```

```
  summarise(Overall_Sat_H_Avg=mean(Overall_Sat_H),
Guest_Room_H_AVG=mean(Guest_Room_H),
Condition_Hotel_H_AVG=mean(Condition_Hotel_H),
Staff_Cared_H_AVG=mean(Staff_Cared_H), Check_In_H_AVG=mean(Check_In_H),
Internet_Sat_H_AVG=mean(Internet_Sat_H),
Customer_SVC_H_AVG=mean(Customer_SVC_H), Tranquility_H_AVG=mean(Tranquility_H))
service.ratings.avgs

service.ratings.avgs.melt <- melt(service.ratings.avgs,id="Likelihood_Recommend")
g.point.service.avgs.ratings <- ggplot(service.ratings.avgs.melt) +
  geom_point(aes(x=variable, y=value, color=Likelihood_Recommend), size=6) +
  coord_flip(ylim=c(1,10)) +
  labs(title="Average of Service Ratings by NPS Type", x="Service Rating Average", y="Service
Categories")
g.point.service.avgs.ratings
##

###----- ROOM_TYPE_CODE_C: Hyatt standard room type code of the guest's room upon
checkout ----
# 164 Unique codes ex: Top 10:
(KING,DDBL,QNQN,1BKN,QNQN,QUEN,DLXK,DLXN,CLBK,2BKN,VW1K)
length(unique(surveyDataSetCleaned$ROOM_TYPE_CODE_C))
sort(table(surveyDataSetCleaned$ROOM_TYPE_CODE_C), decreasing = TRUE)
head(sort(table(surveyDataSetCleaned$ROOM_TYPE_CODE_C), decreasing = TRUE), 10)

###----- ROOM_TYPE_CODE_R: Hyatt standard room type code of the guest's room as per the
booking----
# 144 Unique codes ex: Top 10:
(KING,DDBL,1BKN,QNQN,QUEN,DLXN,VW1K,KNGT,2BKN,DLXK)
length(unique(surveyDataSetCleaned$ROOM_TYPE_CODE_R))
sort(table(surveyDataSetCleaned$ROOM_TYPE_CODE_R), decreasing = TRUE)
head(sort(table(surveyDataSetCleaned$ROOM_TYPE_CODE_R), decreasing = TRUE), 10)

###----- Guest_Room_H: Guest room satisfaction metric ------------------------
# 10 Unique value on a 1 to 10 scale
length(unique(serviceDf$Guest_Room_H))
sort(table(serviceDf$Guest_Room_H), decreasing = TRUE)
describe(as.numeric(serviceDf$Guest_Room_H))

## Visualizations
ggplot(data=serviceDf) +
  geom_bar(mapping = aes(Guest_Room_H)) +
  ggtitle('Guest Room Satisfaction')
ggsave(filename='Guest_Room_Satisfaction.jpg', width = 6, height = 6)
```

```
ggplot(data=serviceDf) +
  geom_bar(mapping = aes(Guest_Room_H)) +
  geom_bar(mapping = aes(Likelihood_Recommend_H)) +
  ggtitle('Guest Room Satisfaction')
ggsave(filename='Guest_Room_Satisfaction.jpg', width = 6, height = 6)


###----- Condition_Hotel_H: Condition of hotel metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Condition_Hotel_H))
sort(table(surveyDataSetCleaned$Condition_Hotel_H), decreasing = TRUE)

###----- Staff_Cared_H: Staff cared metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Staff_Cared_H))
sort(table(surveyDataSetCleaned$Staff_Cared_H), decreasing = TRUE)


###----- Check_In_H: Quality of the check in process metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Check_In_H))
sort(table(surveyDataSetCleaned$Check_In_H), decreasing = TRUE)


###----- Internet_Sat_H - Internet satisfaction metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Internet_Sat_H))
sort(table(surveyDataSetCleaned$Internet_Sat_H), decreasing = TRUE)


###----- Customer_SVC_H: Quality of customer service metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Customer_SVC_H))
sort(table(surveyDataSetCleaned$Customer_SVC_H), decreasing = TRUE)


###----- Tranquility_H: Tranquility metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Tranquility_H))
sort(table(surveyDataSetCleaned$Tranquility_H), decreasing = TRUE)


###----- Overall_Sat_H: Overall satisfaction metric ----------
```

```
# 10 Unique value on a 1 to 10 scale
length(unique(surveyDataSetCleaned$Overall_Sat_H))
sort(table(surveyDataSetCleaned$Overall_Sat_H), decreasing = TRUE)



### Visualizations - Grouped

##---- GEOGRAPHY - Key Drivers --------------------
### Country_PL
length(unique(table(surveyDataSetCleaned$Country_PL)))
sort(table(surveyDataSetCleaned$Country_PL), decreasing = TRUE)

### Country_PL - Visualizations

### City_PL
length(unique(surveyDataSetCleaned$City_PL))
sort(table(surveyDataSetCleaned$Country_PL), decreasing = TRUE)

### City_PL - Visualizations
boxplot(table(surveyDataSetCleaned$City_PL))
barplot(table(surveyDataSetCleaned$City_PL))

### Location_PL
length(unique(surveyDataSetCleaned$Location_PL))
sort(table(surveyDataSetCleaned$Location_PL), decreasing = TRUE)

### Location_PL - Visualizations

### GUEST_COUNTRY_R
length(unique(surveyDataSetCleaned$GUEST_COUNTRY_R))
sort(table(surveyDataSetCleaned$GUEST_COUNTRY_R), decreasing = TRUE)

### GUEST_COUNTRY_R - Visualizations

### Guest_Country_H
length(unique(surveyDataSetCleaned$Guest_Country_H))
sort(table(surveyDataSetCleaned$Guest_Country_H), decreasing = TRUE)

### Guest_Country_H - Visualizations

### STATE_R
length(unique(surveyDataSetCleaned$STATE_R))
sort(table(surveyDataSetCleaned$STATE_R), decreasing = TRUE)
```

```
### STATE_R - Visualizations
##---- GENDER - Key Drivers ----------------------
### Gender_H
length(unique(table(surveyDataSetCleaned$Gender_H)))
gender.dist <- sort(table(surveyDataSetCleaned$Gender_H), decreasing = TRUE)
gender.dist <- data.frame(gender.dist)
colnames(gender.dist) <- c('Gender','No.Of.People')
gender.dist

### Gender_H - Visualizations
ggplot(surveyDataSetCleaned, aes(x=Gender_H)) +
  geom_histogram(binwidth = 1, stat='count')

ggplot(surveyDataSetCleaned, aes(x=factor(1), fill =
factor(surveyDataSetCleaned$Gender_H))) +
  geom_bar(width = 1) + coord_polar(theta = "y")


##---- AGE - Key Drivers ------------------------
### Age_Range_H
length(unique(table(surveyDataSetCleaned$Age_Range_H)))
age.dist <- sort(table(surveyDataSetCleaned$Age_Range_H), decreasing = TRUE)
age.dist <- data.frame(age.dist)
colnames(age.dist) <- c('Age.Range','No.Of.People')
age.dist

sum(is.na(surveyDataSetCleaned$Age_Range_H))

surveyDataSetCleaned$Age_Range_H[is.na(surveyDataSetCleaned$Age_Range_H)] <-
normNAAgeRange(4)

### Age_Range_H - Visualizations
ggplot(surveyDataSetCleaned, aes(Age_Range_H)) +
  geom_bar()

#Mat[which(Mat['A'] == 10), ]
surveyDataSetCleaned[which(surveyDataSetCleaned['Location_PL'] == 'Airport'), ]


#------ Data Transformation ----------------------------------------------------------------

### Transformation - Classify Age Ranges
transf.survey.ds <- classifyAgeRanges(surveyDataSetCleaned)
```

Transformation - Group NPS Scoring
transf.survey.ds <- classifyLikelyhoodToRecommend_Class(transf.survey.ds)
nps.ds <- transf.survey.ds[,60:62]

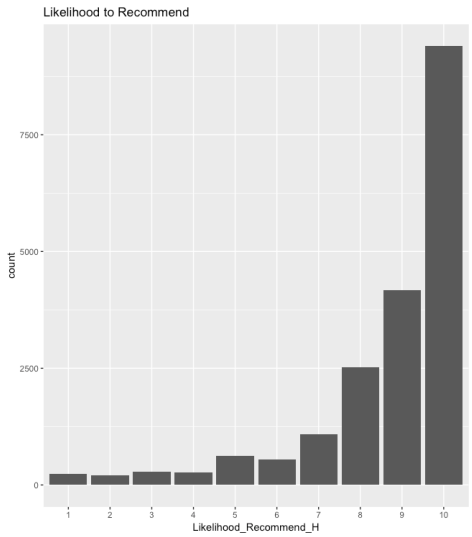
Additional Data:

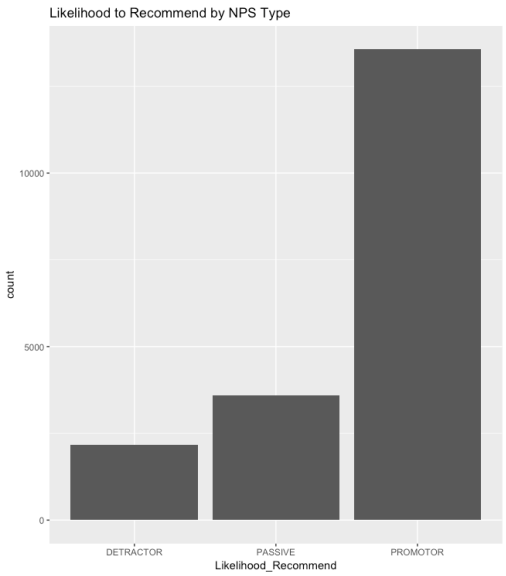
Likelihood to Recommend Scoring Stats

```
> describe(as.numeric(surveyDataSetCleaned$Likelihood_Recommend_H))
   vars     n mean   sd median trimmed  mad min max range  skew kurtosis   se
X1    1 19342  8.7 1.92      9    9.13 1.48   1  10     9 -2.02     4.02 0.01
```

Frequency Table by NPS Type

DETRACTOR	PASSIVE	PROMOTOR
2169	3603	13570



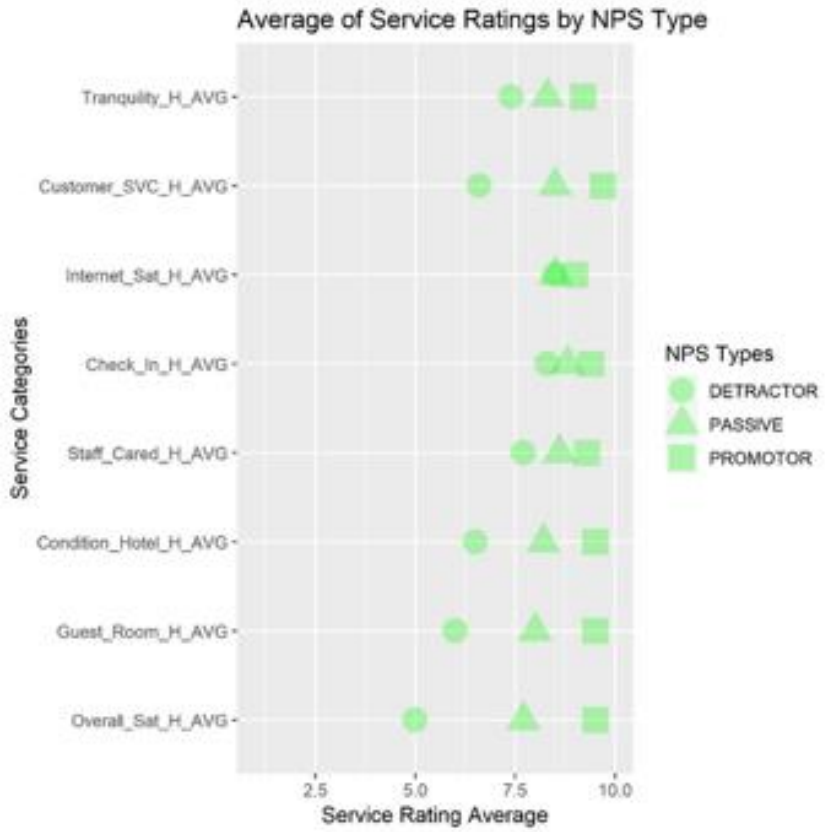


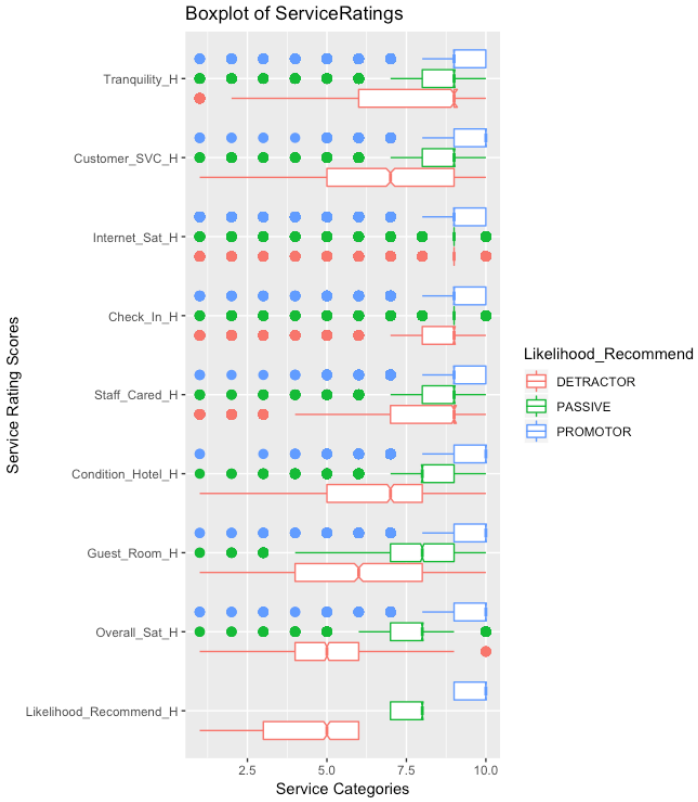
Average Service Scores by NPS Type

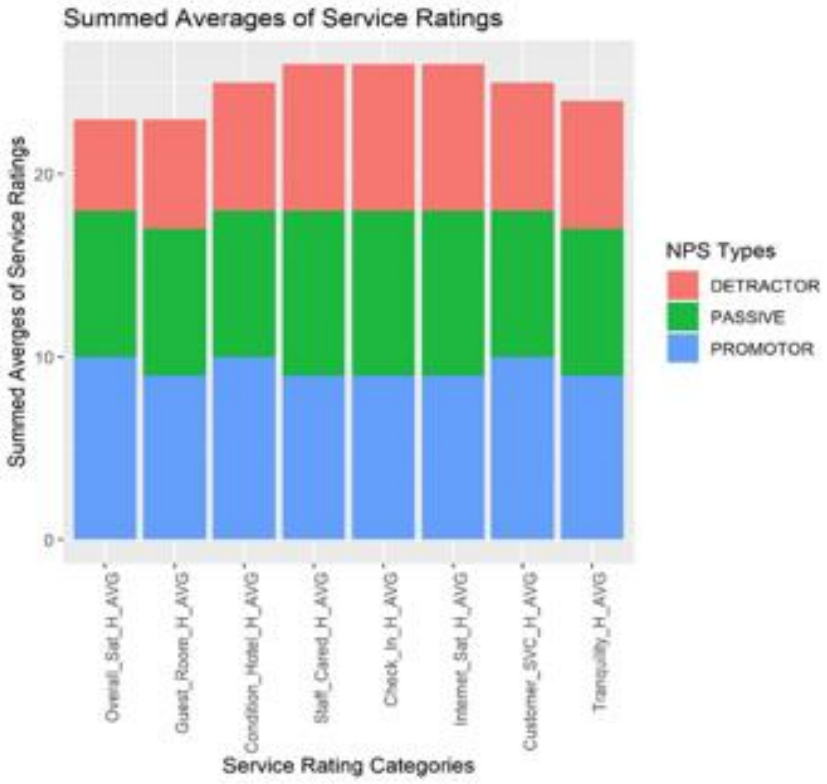
Likelihood_Recommend	Overall_Sat_H_AVG	Guest_Room_H_AVG	Condition_Hotel_H_AVG	Staff_Cared_H_AVG	Check_In_H_AVG	Internet_Sat_H_AVG	Customer_SVC_H_AVG	Tranquility_H_AVG	NPS_TYP_CNT	NPS_TYP_PERC
DETRACTOR	5.0	6.0	6.5	7.7	8.3	8.5	6.6	7.4	2169	11.21394
PASSIVE	7.7	8.0	8.2	8.6	8.8	8.5	8.5	8.5	3603	18.62786
PROMOTOR	9.5	9.5	9.5	9.3	9.4	9.0	9.7	9.2	13570	70.15820

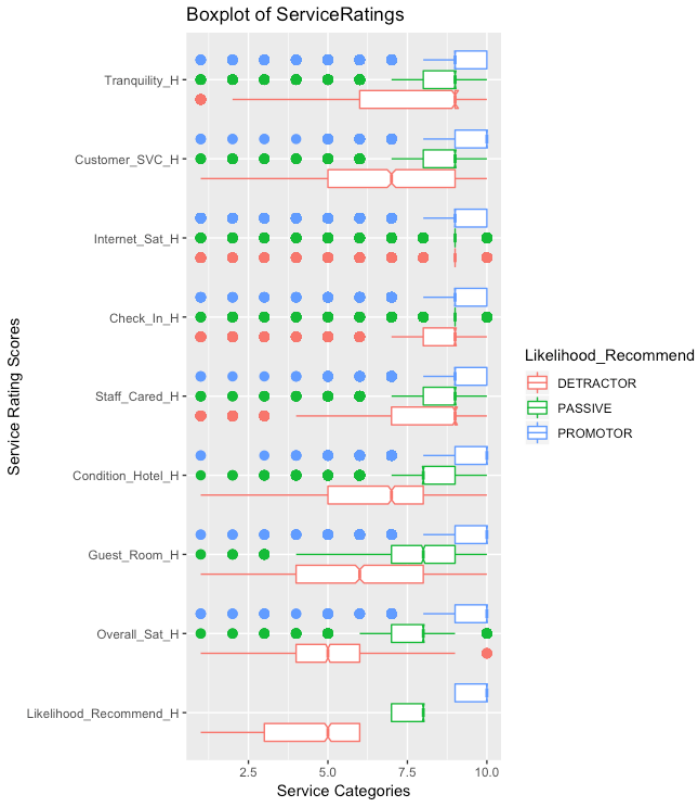
Median Service Scores by NPS Type

Likelihood_Recommend	Overall_Sat_H_MED	Guest_Room_H_MED	Condition_Hotel_H_MED	Staff_Cared_H_MED	Check_In_H_MED	Internet_Sat_H_MED	Customer_SVC_H_MED	Tranquility_H_MED	NPS_TYP_CNT	NPS_TYP_PERC
DETRACTOR	5	6	7	9	9	9	7	9	2169	11.21394
PASSIVE	8	8	8	9	9	9	9	9	3603	18.62786
PROMOTOR	10	10	10	9	9	9	10	9	13570	70.15820


Average of Service Ratings by NPS Type
Tranquility_H_AVG
Customer_SVC_H_AVG
Internet_Sat_H_AVG
Check_In_H_AVG
Staff_Cared_H_AVG
Condition_Hotel_H_AVG
Guest_Room_H_AVG
Overall_Sat_H_AVG
Service Categories
2.5
5.0
7.5
10.0
Service Rating Average
NPS Types
DETRACTOR
PASSIVE
PROMOTOR


Boxplot of ServiceRatings
Tranquility_H
Customer_SVC_H
Internet_Sat_H
Check_In_H
Staff_Cared_H
Condition_Hotel_H
Guest_Room_H
Overall_Sat_H
Likelihood_Recommend_H
Service Rating Scores
2.5
5.0
7.5
10.0
Service Categories
Likelihood_Recommend
DETRACTOR
PASSIVE
PROMOTOR


Summed Averages of Service Ratings
Summed Averges of Service Ratings
20
10
0
Overall_Sat_H_AVG
Guest_Room_H_AVG
Condition_Hotel_H_AVG
Staff_Cared_H_AVG
Check_In_H_AVG
Internet_Sat_H_AVG
Customer_SVC_H_AVG
Tranquility_H_AVG
Service Rating Categories
NPS Types
DETRACTOR
PASSIVE
PROMOTOR


Boxplot of ServiceRatings
Tranquility_H
Customer_SVC_H
Internet_Sat_H
Check_In_H
Staff_Cared_H
Condition_Hotel_H
Guest_Room_H
Overall_Sat_H
Likelihood_Recommend_H
Service Rating Scores
2.5
5.0
7.5
10.0
Service Categories
Likelihood_Recommend
DETRACTOR
PASSIVE
PROMOTOR

Code:

How does the length of stay impact NPS?

```
descriptive_stats <- function(x){
  a<-mean(x)
  b<-median(x)
  z <- max(x)
  c <-  min(x)
  d <- sd(x)

  cat("mean:",a,"\nmedian:",b, "\nmax:", z, "\nmin:", c,"\nstandard deviation",d)
}

descriptive_stats(ds$LENGTH_OF_STAY_C)
plot(ds$LENGTH_OF_STAY_C)
quantile(ds$LENGTH_OF_STAY_C,0.5)
quantile(ds$LENGTH_OF_STAY_C,0.75)
quantile(ds$LENGTH_OF_STAY_C,0.95)
quantile(ds$LENGTH_OF_STAY_C,0.99)


sqldf("SELECT case when Likelihood_Recommend_H < 7 then 'Detractor'
+ when  Likelihood_Recommend_H < 9 then 'Passive' else 'Promoter'end as 'NPS',
avg(LENGTH_OF_STAY_C) as 'AvgLength'
+       FROM ds
+       where LENGTH_OF_STAY_C < 14 group by 1  ")
       NPS  AvgLength
1 Detractor  2.228983
2  Passive  2.323340
3  Promoter  2.158239

df4 <- data.frame(sqldf("SELECT Country_PL as 'Country_of_Hotel',
avg(LENGTH_OF_STAY_C) as 'AvgLength'  FROM ds where LENGTH_OF_STAY_C < 14
group by 1 order by 2  "))
p4 <- ggplot(data=df4, aes(x= reorder(Country_of_Hotel,AvgLength), y=AvgLength)) +
geom_bar(stat="identity")
p4
p5 <-p4 + theme(axis.text.x = element_text(angle = 90))+ theme( panel.grid.major =
element_line(size = .5, colour = "grey80")) + scale_y_continuous(breaks=c(0, 1, 2, 3, 4, 5,6,7))
```

How do the different demographic components relate to NPS?

```
library("ggthemes")

##Gender
#Gender_H,Likelihood_Recommend_H,NPS_Type
gender.attributes <- c('Gender_H','Likelihood_Recommend_H','NPS_Type')
genderDf <- surveyDataSetCleaned[gender.attributes]
head(genderDf)
summary(genderDf)

genderDf <- genderDf %>%
  mutate(Likelihood_Recommend_H=as.numeric(Likelihood_Recommend_H))
summary(genderDf)

#Histogram
View(genderDf)

gender.plot <- ggplot(genderDf, aes(Likelihood_Recommend_H)) +
geom_histogram(aes(y=..count..), binwidth = 2)
gender.plot <- gender.plot + facet_grid(Gender_H ~ NPS_Type) + theme_tufte()
gender.plot
ggsave(filename='Gender_Attributes.jpg', width = 6, height = 6)

##Age
#Age_Range_H,Age_Range_H_Class,Likelihood_Recommend_H,NPS_Type
age.attributes <-
c('Age_Range_H','Age_Range_H_Class','Likelihood_Recommend_H','NPS_Type')
ageDf <- surveyDataSetCleaned[age.attributes]
head(ageDf)

ageDf <- ageDf %>%
  mutate(Likelihood_Recommend_H=as.numeric(Likelihood_Recommend_H))
View(ageDf)

#Histogram
age.plot <- ggplot(ageDf, aes(Likelihood_Recommend_H)) + geom_histogram(aes(y=..count..),
binwidth = 2)
age.plot <- age.plot + facet_grid(Age_Range_H ~ NPS_Type) + theme_tufte()
age.plot
```

```
ggsave(filename='Age_Attributes.jpg', width = 6, height = 6)


##Gender and Age analysis##
genAge.attributes <-
c('Gender_H','Likelihood_Recommend_H','NPS_Type','Age_Range_H','Age_Range_H_Class')
genAgeDf <- surveyDataSetCleaned[genAge.attributes]
head(genAgeDf)
str(genAgeDf)

genAgeDf <- genAgeDf %>%
  mutate(Likelihood_Recommend_H=as.numeric(Likelihood_Recommend_H))
View(genAgeDf)

#Plot

genAge.plot <- ggplot(data=genAgeDf) + geom_bar(aes(x=Likelihood_Recommend_H,
fill=NPS_Type)) + facet_grid(Gender_H ~ Age_Range_H)
genAge.plot <- genAge.plot + theme_tufte() + theme(legend.position ="none")
genAge.plot
ggsave(filename='GenAge_Attributes.jpg', width = 6, height = 6)

sqldf("select Gender_H as Gender ,Age_Range_H as Age_Range,
count(Age_Range_H)*100/(select count(*) from genAgeDf) as Percentage from genAgeDf
where Likelihood_Recommend_H in (9,10) group by Gender_H,Age_Range_H")

sqldf("select Gender_H as Gender ,Age_Range_H as Age_Range,
count(Age_Range_H)*100/(select count(*) from genAgeDf) as Percentage from genAgeDf
where Likelihood_Recommend_H in (7,8) group by Gender_H,Age_Range_H")
```

Which hotels/locations are performing better?

```
# surveyDataSetCleaned$City_PL
# surveyDataSetCleaned$Property.Latitude_PL
# surveyDataSetCleaned$Property.Longitude_PL
# surveyDataSetCleaned$NPS_Type
# surveyDataSetCleaned$Condition_Hotel_H

sds <- surveyDataSetCleaned
sds$NPS <- NA
```

```
sds$NPS <- ifelse(sds$NPS_Type=="Detractor", 0, ifelse(sds$NPS_Type=="Passive", 1, 2))

countPeople <- function(ds) {
  tap <- tapply(ds$NPS, ds$City_PL, length)
  return(tap)
}


meanCondition <- function(ds) {
  tap <- tapply(ds$Condition_Hotel_H, ds$City_PL, mean)
  return(tap)
}

sds$Condition_Hotel_H <- as.numeric(sds$Condition_Hotel_H)

createNPSDS <- function(ntype) {
  count <- countPeople(sds[sds$NPS_Type==ntype,])
  meancondition <- meanCondition(sds[sds$NPS_Type==ntype,])*20

  ds <- data.frame(count, meancondition)
  ds$cityname <- NA
  ds$latitude <- NA
  ds$longitude <- NA
  #View(ds_det)

  for(i in 1:nrow(ds)) {
    ds$cityname[i] <- rownames(ds)[i]
    ds$latitude[i] <- sds$Property.Latitude_PL[sds$City_PL==rownames(ds)[i]][1]
    ds$longitude[i] <- sds$Property.Longitude_PL[sds$City_PL==rownames(ds)[i]][1]
  }
  rownames(ds) <- NULL

  return(ds)
}

ds_det <- createNPSDS("Detractor")
ds_pas <- createNPSDS("Passive")
ds_pro <- createNPSDS("Promoter")

View(ds_det)
View(ds_pas)
View(ds_pro)

####################
```

```
## World Map
####################
#install.packages("maps")
world <- map_data("world")
breakC <- c(50, 100, 200)

createWorldMap <- function(ds, Title, Filename) {
  worldmap <- ggplot() +
    geom_polygon(data = world, aes(x = long, y = lat, group = group),
                 fill="white", colour="gray50", alpha=1) +
    coord_fixed(1.3)
  worldmap <- worldmap +
    geom_point(data=ds, aes(x=ds$longitude, y=ds$latitude, color=ds$meancondition,
size=ds$count))
  worldmap <- worldmap +
    scale_colour_gradientn(limits=c(50, 200), colours=c("red", "black","deepskyblue"),
                           breaks=breakC, labels=format(breakC))
  worldmap <- worldmap +
    labs(title=Title, x="", y="", color="Condition", size="Count")
  ggsave(filename=Filename, width = 6, height = 6)
  return(worldmap)
}

worldmap_det <- createWorldMap(ds_det, "Detractor", "WorldMap_Detractor.jpg")
worldmap_det
worldmap_pas <- createWorldMap(ds_pas, "Passive", "WorldMap_Passive.jpg")
worldmap_pas
worldmap_pro <- createWorldMap(ds_pro, "Promoter", "WorldMap_Promoter.jpg")
worldmap_pro

####################
## Tables
####################

head(ds_pro[order(-ds_pro$meancondition),])
head(ds_pas[order(-ds_pas$meancondition),])
head(ds_det[order(-ds_det$meancondition,])

head(ds_pro[order(-ds_pro$count),])
head(ds_pas[order(-ds_pas$count),])
head(ds_det[order(-ds_det$count),])

tail(ds_pro[order(-ds_pro$count),])
tail(ds_pas[order(-ds_pas$count),])
```

```
tail(ds_det[order(-ds_det$count),])

tail(ds_pro[order(-ds_pro$meancondition),])
tail(ds_pas[order(-ds_pas$meancondition),])
tail(ds_det[order(-ds_det$meancondition),])
```

Do business and leisure travelers have different NPS?

```
##Business or Leisure

adultMin <- min(ds_nb$ADULT_NUM_C)
adultNum <- (max(ds_nb$ADULT_NUM_C) - adultMin + 1)
childMin <- min(ds_nb$CHILDREN_NUM_C)
childMax <- max(ds_nb$CHILDREN_NUM_C)
childNum <- (childMax - childMin + 1)
totalEntries <- adultNum * childNum
dummyEntry <- replicate(totalEntries, 0)

adultEntry <- NULL
for(a in 1:adultNum) {
  entry <- replicate(childNum, adultMin+a-1)
  adultEntry <- c(adultEntry, entry)
}
adultEntry

childEntry <- NULL
for(a in 1:adultNum) {
  entry <- childMin:childMax
  childEntry <- c(childEntry, entry)
}
childEntry

meanPOV <- data.frame(dummyEntry, dummyEntry, adultEntry, childEntry)
# assign column names
colnames(meanPOV) <- c("meanpov", "total", "Adults", "Children")

for(r in 1:nrow(meanPOV)) {
  foundEntry <- ds_nb[ds_nb$ADULT_NUM_C==meanPOV$Adults[r] &
ds_nb$CHILDREN_NUM_C==meanPOV$Children[r],]
  meanPOV$meanpov[r] <- ifelse(nrow(foundEntry)==0, NA, 1 - (sum(foundEntry$POV) /
nrow(foundEntry)))
  meanPOV$total[r] <- ifelse(nrow(foundEntry)==0, NA, nrow(foundEntry))
```

```
}

breakC <- c(0, 1)
breakC2 <- c(3, 60)

# polt result using ggplot, setting "Adults" as x-axis and "Children" as y-axis
plot.pov <- ggplot(meanPOV, aes(x=Adults,y=Children)) +
  # use point size and color shade to illustrate how big is the error
  geom_point(aes(size=total, color=meanpov), na.rm = TRUE)+
  scale_colour_gradientn(limits=c(0, 1), colours=c("red", "deepskyblue"),
                         breaks=breakC, labels=c("Leisure", "Business")) +
  scale_size_continuous(range = breakC2) +
  guides(size=FALSE) +
  ggtitle("Business or Leisure")
ggsave(filename="Business_or_Leisure.jpg", width = 6, height = 6)
plot.pov

nrow(ds_nb[ds_nb$ADULT_NUM_C==1 & ds_nb$CHILDREN_NUM_C==0 &
ds_nb$POV==0,])/ nrow(ds_nb)
nrow(ds_nb[ds_nb$CHILDREN_NUM_C>0 & ds_nb$POV==0,])
nrow(ds_nb[ds_nb$CHILDREN_NUM_C>0 & ds_nb$POV==0,]) / nrow(ds_nb) * 100



## Rate for Adult 1, Child 0, Business
# Det
nrow(ds_nb_det[ds_nb_det$ADULT_NUM_C==1 & ds_nb_det$CHILDREN_NUM_C==0 &
ds_nb_det$POV==0,])/ nrow(ds_nb_det[ds_nb_det$ADULT_NUM_C==1 &
ds_nb_det$CHILDREN_NUM_C==0,])
# Pas
nrow(ds_nb_pas[ds_nb_pas$ADULT_NUM_C==1 & ds_nb_pas$CHILDREN_NUM_C==0 &
ds_nb_pas$POV==0,])/ nrow(ds_nb_pas[ds_nb_pas$ADULT_NUM_C==1 &
ds_nb_pas$CHILDREN_NUM_C==0,])
# Pro
nrow(ds_nb_pro[ds_nb_pro$ADULT_NUM_C==1 & ds_nb_pro$CHILDREN_NUM_C==0 &
ds_nb_pro$POV==0,])/ nrow(ds_nb_pro[ds_nb_pro$ADULT_NUM_C==1 &
ds_nb_pro$CHILDREN_NUM_C==0,])
# All
nrow(ds_nb[ds_nb$ADULT_NUM_C==1 & ds_nb$CHILDREN_NUM_C==0 &
ds_nb$POV==0,])/ nrow(ds_nb[ds_nb$ADULT_NUM_C==1 &
ds_nb$CHILDREN_NUM_C==0,]) * 100

#####################################################
##Naive Bayes
```

```
#####################################################
# surveyDataSetCleaned$ADULT_NUM_C
# surveyDataSetCleaned$CHILDREN_NUM_C
# surveyDataSetCleaned$POV_CODE_C
library(e1071)
library(ggplot2)

sds <- surveyDataSetCleaned

# BUSINESS --> 0, LEISURE --> 1
sds$POV <- NA
sds$POV <- ifelse(sds$POV_CODE_C=="BUSINESS", 0, 1)

summary(sds$POV_CODE_C)
tapply(sds$POV_CODE_C, sds$POV, length)

# Original Dataset
ds_nb <- data.frame(sds$ADULT_NUM_C, sds$CHILDREN_NUM_C, sds$POV)
colnames(ds_nb) <- c("ADULT_NUM_C", "CHILDREN_NUM_C", "POV")
is.numeric(ds_nb$ADULT_NUM_C)
is.numeric(ds_nb$CHILDREN_NUM_C)
is.numeric(ds_nb$POV)

# Split into Train and Test dataset
randIndex <- sample(1:nrow(ds_nb))
length(randIndex)

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

trainNBdata <- ds_nb[randIndex[1:cutpoint2_3],]
dim(trainNBdata)
head(trainNBdata)

testNBdata <- ds_nb[randIndex[(cutpoint2_3+1):nrow(ds_nb)],]
dim(testNBdata)
head(testNBdata)

# Naive Bayes

nbModel <- naiveBayes(as.factor(POV)~., data=trainNBdata)
nbPred <- predict(nbModel, testNBdata)
nbPred
```

```
compNBTable <- data.frame(testNBdata$POV, nbPred)
colnames(compNBTable) <- c("test", "Pred")
View(compNBTable)
compNBTable$Pred <- as.numeric(compNBTable$Pred)-1

compNBTable$error <- abs(compNBTable$test - compNBTable$Pred)
# create a new dataframe contains error, tempreture and wind
nbResult <- data.frame(compNBTable$error, testNBdata$ADULT_NUM_C,
testNBdata$CHILDREN_NUM_C, testNBdata$POV)
# assign column names
colnames(nbResult) <- c("error","Adults","Children", "POV")

# Correct Answer
100 - (sum(nbResult$error) / nrow(nbResult) * 100)

adultMin <- min(nbResult$Adults)
adultNum <- (max(nbResult$Adults) - adultMin + 1)
childMin <- min(nbResult$Children)
childMax <- max(nbResult$Children)
childNum <- (childMax - childMin + 1)
totalEntries <- adultNum * childNum
dummyEntry <- replicate(totalEntries, 0)

adultEntry <- NULL
for(a in 1:adultNum) {
  entry <- replicate(childNum, adultMin+a-1)
  adultEntry <- c(adultEntry, entry)
}
adultEntry

childEntry <- NULL
for(a in 1:adultNum) {
  entry <- childMin:childMax
  childEntry <- c(childEntry, entry)
}
childEntry

nbMeanError <- data.frame(dummyEntry, adultEntry, childEntry)
# assign column names
colnames(nbMeanError) <- c("meanerror", "Adults", "Children")

for(r in 1:nrow(nbMeanError)) {
  foundEntry <- nbResult[nbResult$Adults==nbMeanError$Adults[r] &
nbResult$Children==nbMeanError$Children[r],]
```

```
  nbMeanError$meanerror[r] <- ifelse(nrow(foundEntry)==0, NA, sum(foundEntry$error) /
nrow(foundEntry))
}

# polt result using ggplot, setting "Adults" as x-axis and "Children" as y-axis
plot.nb.good <- ggplot(nbMeanError, aes(x=Adults,y=Children)) +
  # use point size and color shade to illustrate how big is the error
  geom_point(aes(size=meanerror, color=meanerror), na.rm = TRUE)+
  ggtitle("Naive Bayes")
ggsave(filename="NaiveBayes_POV.jpg", width = 6, height = 6)
plot.nb.good
```

Is it possible to predict NPS score using revenue generated?

```
str(surveyDataSetCleaned)
revenue.attributes <-
c('REVENUE_USD_R','LENGTH_OF_STAY_C','ROOM_TYPE_CODE_R','Likelihood_Recomm
end_H','NPS_Type')
revenueDf <- surveyDataSetCleaned[revenue.attributes]

head(revenueDf)
summary(revenueDf)

revenueDf <- revenueDf %>%
  mutate(Likelihood_Recommend_H=as.numeric(Likelihood_Recommend_H))
summary(revenueDf)

View(revenueDf)


ggplot(data=revenueDf) + geom_point(aes(y=REVENUE_USD_R,
x=Likelihood_Recommend_H, color=LENGTH_OF_STAY_C))
ggsave(filename='Rev_Likelihood.jpg', width = 6, height = 6)

roomType.plot <- ggplot(data = revenueDf, aes (x=ROOM_TYPE_CODE_R, y=
Likelihood_Recommend_H)) + geom_point()
roomType.plot

revenueDf$REVENUE_USD_R <- round(na.2.mean(revenueDf$REVENUE_USD_R))


#Predictions
```

```
dim(revenueDf)
#air[1:5,]
randIndex <- sample(1:dim(revenueDf)[1])
length(randIndex)

# Split of data
cutpoint2_3 <- floor(2*dim(revenueDf)[1]/3)
# check the 2/3 cutpoint
cutpoint2_3
# train data set
#
trainData <- revenueDf[randIndex[1:cutpoint2_3],]
dim(trainData)
head(trainData)
# test data set
#
testData <- revenueDf[randIndex[(cutpoint2_3+1):dim(revenueDf)[1]],]
dim(testData)
head(testData)


#svm model for revenue and length of stay
svmRecommend <-
ksvm(Likelihood_Recommend_H~REVENUE_USD_R+LENGTH_OF_STAY_C,
                data = trainData,
                kernel = "rbfdot",
                kpar = "automatic",
                 C = 10,
                cross = 10,
                scaled = c(TRUE),
                prob.model = TRUE
)
svmPred <- predict(svmRecommend,
                 testData,
                 type = "votes"
)

compTable <- data.frame(testData$Likelihood_Recommend_H, svmPred[,1])
colnames(compTable) <- c("Actual","Pred")
compTable$legthStay <- testData$LENGTH_OF_STAY_C

# compute the Root Mean Squared Error
sqrt(mean((compTable$Actual-compTable$Pred)^2))
```

```
g <- ggplot(data=compTable, aes(x=Actual)) + geom_point(aes(y=Pred, color=legthStay)) +
ggtitle("Length of stay", subtitle = waiver())
g <- g + theme_tufte()
ggsave(filename='Revenue-NPS.jpg', width = 6, height = 6)


##Room Type and Revenue#


#Converting all data types to numeric
newR <- revenueDf
newR$NPS_Type <- NULL
newR$RecommendLikelihood <- NULL
str(newR)


#convert the factor value to numeric value.
newR$RoomType <- as.numeric(newR$ROOM_TYPE_CODE_R)
newR$ROOM_TYPE_CODE_R <- NULL
newR$RoomType <- round(na.2.mean(newR$RoomType))
newR$REVENUE_USD_R <- round(na.2.mean(newR$REVENUE_USD_R))
newR$Likelihood_Recommend_H <- as.numeric(newR$Likelihood_Recommend_H)

#Predictions
dim(newR)
#air[1:5,]
randIndex <- sample(1:dim(newR)[1])
length(randIndex)

# Split of data
cutpoint2_3 <- floor(2*dim(newR)[1]/3)
# check the 2/3 cutpoint
cutpoint2_3
# train data set
#
trainData <- newR[randIndex[1:cutpoint2_3],]
dim(trainData)
head(trainData)
# test data set
#
testData <- newR[randIndex[(cutpoint2_3+1):dim(newR)[1]],]
dim(testData)
head(testData)
```

```
#svm model
svmRecommend <- ksvm(Likelihood_Recommend_H~REVENUE_USD_R+RoomType,
                     data = trainData,
                     kernel = "rbfdot",
                     kpar = "automatic",
                     C = 10,
                     cross = 10,
                     scaled = c(TRUE),
                     prob.model = TRUE
)
svmPred <- predict(svmRecommend,
                   testData,
                   type = "votes"
)

compTable <- data.frame(testData$Likelihood_Recommend_H, svmPred[,1])
colnames(compTable) <- c("Actual","Pred")
compTable$roomType <- testData$RoomType

# compute the Root Mean Squared Error
sqrt(mean((compTable$Actual-compTable$Pred)^2))


g <- ggplot(data=compTable, aes(x=Actual)) + geom_point(aes(y=Pred,color=roomType)) +
ggtitle("Room Type", subtitle = waiver())
g <- g + theme_tufte() + theme(legend.position ="none")
ggsave(filename='RoomType-NPS.jpg', width = 6, height = 6)
```

Which Service factors are the best predictors of promoters? Focusing on improving those Service factors, what hotel geolocation regions should be targeted?

```
##---- SERVICE - Key Drivers ---------------------------------------------------------
# Business Questions:
# - Which Service Categories have the most significant impact on an NPS PROMOTER
scoring?
# --- Customer_SVC_H & Condition_Hotel_H & Guest_Room_H
service.attributes <-
c('Likelihood_Recommend','Likelihood_Recommend_H','Overall_Sat_H','Guest_Room_H','Cond
```

```
ition_Hotel_H','Staff_Cared_H','Check_In_H','Internet_Sat_H','Customer_SVC_H','Tranquility_H'
)
serviceDf <- surveyDataSetCleaned[service.attributes]
head(serviceDf)
summary(serviceDf)

serviceDf <- serviceDf %>%

mutate(Likelihood_Recommend_H=as.numeric(Likelihood_Recommend_H),Overall_Sat_H=as.
numeric(Overall_Sat_H), Guest_Room_H=as.numeric(Guest_Room_H),
Condition_Hotel_H=as.numeric(Condition_Hotel_H),
       Staff_Cared_H=as.numeric(Staff_Cared_H), Check_In_H=as.numeric(Check_In_H),
Internet_Sat_H=as.numeric(Internet_Sat_H),
Customer_SVC_H=as.numeric(Customer_SVC_H),
       Tranquility_H=as.numeric(Tranquility_H))
summary(serviceDf)

## Visualize Descriptive Statistics via Boxplot
ggplot(data=serviceDf, aes(x=factor(0), y=Likelihood_Recommend_H)) +
  geom_boxplot(fill="gray",outlier.colour="red",outlier.shape=16 ,outlier.size=3, notch=TRUE) +
  coord_flip(ylim=c(0,10)) +
  labs(title="Boxplot of Likelihood_Recommend_H", x="", y="Rating Score")
ggsave(filename="Boxplot_of_Likelihood_Recommend_H.jpg", width = 6, height = 6)

service.ratings.melt <- melt(serviceDf,id="Likelihood_Recommend" )
g.box.service.ratings <- ggplot(service.ratings.melt) +
  geom_boxplot(aes(x=variable, y=value, color=Likelihood_Recommend), outlier.shape=16
,outlier.size=3, notch=TRUE) +
  coord_flip(ylim=c(1,10)) +
  labs(title="Boxplot of ServiceRatings", x="Service Rating Scores", y="Service Categories") +
  guides(color=guide_legend(title="NPS Types"))
g.box.service.ratings
ggsave(filename="Boxplot_of_ServiceRatings.jpg", width = 6, height = 6)
##

## Average scoring by NPS Type
service.ratings.avgs <- serviceDf %>%
  group_by(Likelihood_Recommend) %>%
  summarise(Overall_Sat_H_AVG=round(mean(Overall_Sat_H),1),
Guest_Room_H_AVG=round(mean(Guest_Room_H),1),
Condition_Hotel_H_AVG=round(mean(Condition_Hotel_H),1),
         Staff_Cared_H_AVG=round(mean(Staff_Cared_H),1),
Check_In_H_AVG=round(mean(Check_In_H),1),
Internet_Sat_H_AVG=round(mean(Internet_Sat_H),1),
```

```
        Customer_SVC_H_AVG=round(mean(Customer_SVC_H),1),
Tranquility_H_AVG=round(mean(Tranquility_H),1),
        NPS_TYP_CNT=n(), NPS_TYP_PERC=(n()/nrow(serviceDf))*100)
View(service.ratings.avgs)


## Median scoring by NPS Type
service.ratings.med <- serviceDf %>%
  group_by(Likelihood_Recommend) %>%
  summarise(Overall_Sat_H_MED=round(median(Overall_Sat_H),1),
Guest_Room_H_MED=round(median(Guest_Room_H),1),
Condition_Hotel_H_MED=round(median(Condition_Hotel_H),1),
        Staff_Cared_H_MED=round(median(Staff_Cared_H),1),
Check_In_H_MED=round(median(Check_In_H),1),
Internet_Sat_H_MED=round(median(Internet_Sat_H),1),
        Customer_SVC_H_MED=round(median(Customer_SVC_H),1),
Tranquility_H_MED=round(median(Tranquility_H),1),
        NPS_TYP_CNT=n(), NPS_TYP_PERC=(n()/nrow(serviceDf))*100)
View(service.ratings.med)

# Melt data frame
col.remove <- which(colnames(service.ratings.avgs)=="NPS_TYP_CNT" |
colnames(service.ratings.avgs)== "NPS_TYP_PERC")
service.ratings.avgs.melt <- melt(service.ratings.avgs[,-
col.remove],id="Likelihood_Recommend")

# Point Plot - Horizontal - Service Rating Averages per Service Category
g.point.service.avgs.ratings <- ggplot(service.ratings.avgs.melt) +
  geom_point(aes(x=variable, y=value, shape=Likelihood_Recommend), size=6,
color="green",alpha=.3) +
  coord_flip(ylim=c(1,10)) +
  labs(title="Average of Service Ratings by NPS Type", y="Service Rating Average", x="Service
Categories") +
  guides(shape=guide_legend(title="NPS Types"))
g.point.service.avgs.ratings
ggsave(filename="Average_of_Service_Ratings_by_NPS_Type.jpg", width = 6, height = 6)

# Bar Plot - Vertically Stacked, Summed Averages of Service Ratings
ggplot(service.ratings.avgs.melt, aes(x=variable,y=value, fill=Likelihood_Recommend)) +
  geom_bar(stat="identity") +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title="Summed Averages of Service Ratings", y="Summed Averages of Service Ratings",
x="Service Rating Categories") +
  guides(fill=guide_legend(title="NPS Types"))
```

```
ggsave(filename="Summed_Averages_of_Service_Ratings.jpg", width = 6, height = 6)

## Create Training and Test data for SVM predictions
serviceDataSetSplits <- partitionDataSet(serviceDf,0.33)
#Training Data
serviceTrain <- serviceDataSetSplits$trainingData
dim(serviceTrain)
head(serviceTrain)
#Test Data
serviceTest <- serviceDataSetSplits$testingData
dim(serviceTest)
head(serviceTest)

#Classify dataset as being either a 'Promoter' or 'Not a Promoter'(i.e. a Detractor)
serviceTrain$Promoter <- ifelse(serviceTrain$Likelihood_Recommend=="PROMOTER", 1, 0)
serviceTest$Promoter <- ifelse(serviceTest$Likelihood_Recommend=="PROMOTER", 1, 0)
removeAttributes <- c('Likelihood_Recommend','Likelihood_Recommend_H','Overall_Sat_H')
serviceTrain <- serviceTrain[,!(names(serviceTrain) %in% removeAttributes)]
serviceTest <- serviceTest[,!(names(serviceTest) %in% removeAttributes)]
str(serviceTrain)
str(serviceTest)
serviceTrain$Promoter <- as.factor(serviceTrain$Promoter)
serviceTest$Promoter <- as.factor(serviceTest$Promoter)

##---- Primary Service Rating Predictors of Promoter ----------------------##
## Guest_Room_H | Condition_Hotel_H | Customer_SVC_H
## Guest_Room_H:
## -
##-- Compute models and plot the results for 'svm' (in the e1071 package)
formula <- Promoter ~ Guest_Room_H + Customer_SVC_H + Condition_Hotel_H
computeSVMModels(serviceTrain,serviceTest,formula)

###----- Guest_Room_H: Guest room satisfaction metric ------------------------
# 10 Unique value on a 1 to 10 scale
length(unique(serviceDf$Guest_Room_H))
sort(table(serviceDf$Guest_Room_H), decreasing = TRUE)
describe(as.numeric(serviceDf$Guest_Room_H))

## Visualizations
ggplot(data=serviceDf) +
  geom_bar(mapping = aes(Guest_Room_H)) +
  ggtitle('Guest Room Satisfaction')
ggsave(filename='Guest_Room_Satisfaction.jpg', width = 6, height = 6)
```

```
# Bivariat plots
plot.biv.guest.likelihood <- ggplot(serviceDf, aes(x=Likelihood_Recommend_H,
y=Guest_Room_H)) +
  geom_point() +
  stat_smooth(method='lm',col='red') +
  labs(title="Likelihood to Recommend dependent on Guest Room Ratings", x="Overall
Likelihood to Recommend", y="Guest Room Service Rating")
plot.biv.guest.likelihood
ggsave(filename='Likelihood_to_Recommend_dependenton_Guest_Room.jpg', width = 6,
height = 6)

# Compute SVM Prediction Accuracy Score
formula <- Promoter ~ Guest_Room_H
computeSVMModels(serviceTrain,serviceTest,formula)

###----- Condition_Hotel_H: Condition of hotel metric ----------
# 10 Unique value on a 1 to 10 scale
length(unique(serviceDf$Condition_Hotel_H))
sort(table(serviceDf$Condition_Hotel_H), decreasing = TRUE)
describe(as.numeric(serviceDf$Condition_Hotel_H))

## Visualizations
ggplot(data=serviceDf) +
  geom_bar(mapping = aes(Condition_Hotel_H)) +
  ggtitle('Condition of Hotel Satisfaction Ratings')
ggsave(filename='Condition_of_Hotel_Satisfaction.jpg', width = 6, height = 6)

# Bivariat plots
plot.biv.cond.likelihood <- ggplot(serviceDf, aes(x=Likelihood_Recommend_H,
y=Condition_Hotel_H)) +
  geom_point() +
  stat_smooth(method='lm',col='red') +
  labs(title="Likelihood to Recommend dependent on Hotel Condition Ratings", x="Overall
Likelihood to Recommend", y="Hotel Condition Service Rating")
plot.biv.cond.likelihood
ggsave(filename='Likelihood_to_Recommend_dependenton_Condition.jpg', width = 6, height =
6)

# Compute SVM Prediction Accuracy Score
formula <- Promoter ~ Condition_Hotel_H
computeSVMModels(serviceTrain,serviceTest,formula)

###----- Condition_Hotel_H: Condition of hotel metric ----------
# 10 Unique value on a 1 to 10 scale
```

```
length(unique(serviceDf$Condition_Hotel_H))
sort(table(serviceDf$Condition_Hotel_H), decreasing = TRUE)
describe(as.numeric(serviceDf$Condition_Hotel_H))

## Visualizations
ggplot(data=serviceDf) +
  geom_bar(mapping = aes(Condition_Hotel_H)) +
  ggtitle('Condition of Hotel Satisfaction Ratings')
ggsave(filename='Condition_of_Hotel_Satisfaction.jpg', width = 6, height = 6)

# Bivariat plots
plot.biv.cond.likelihood <- ggplot(serviceDf, aes(x=Likelihood_Recommend_H,
y=Condition_Hotel_H)) +
  geom_point() +
  stat_smooth(method='lm',col='red') +
  labs(title="Likelihood to Recommend dependent on Hotel Condition Ratings", x="Overall
Likelihood to Recommend", y="Hotel Condition Service Rating")
plot.biv.cond.likelihood
ggsave(filename='Likelihood_to_Recommend_dependenton_Condition.jpg', width = 6, height =
6)

# Compute SVM Prediction Accuracy Score
formula <- Promoter ~ Condition_Hotel_H
computeSVMModels(serviceTrain,serviceTest,formula)

# Show all three results (charts) in one window, using the grid.arrange function
ga3 <- grid.arrange(plot.biv.cust.likelhood, plot.biv.guest.likelihood, plot.biv.cond.likelihood,
ncol=1, nrow=3, top="Top 3, Best Service factors for predicting promotors")
ggsave(file="Grid_Arrange_biv_service.jpg", ga3, width = 8, height = 8)

##---- GEOGRAPHY - Key Drivers --------------------
##
# Question: What geolocation has the lowest scoring for service factors, Customer_SVC_H,
Condition_Hotel_H, Guest_Room_H
# Service - What Service factors are the best predictors of promoters?
#   Focusing on improving those Service factors, what hotel geolocation regions should be
targeted?
###------------------------------------------------------------------
#-> 93% of the Top 3 Service detractors are within the United States
###------------------------------------------------------------------
#-> 94% of Guest Room detractor's are within the United States
#-> 93% of Condition Hotel detractors are within the United States
#-> 92% of Customer Service detractors are within the United States
```

```
keep <-
c('Hotel.Name.Long_PL','Customer_SVC_H','Condition_Hotel_H','Guest_Room_H','Country_PL'
,'City_PL','STATE_R','Location_PL','Likelihood_Recommend_H','Property.Latitude_PL','Property
.Longitude_PL')
geo.service.detractors <- detractorDf[,(names(detractorDf) %in% keep)]
geo.service.detractors$Property.Latitude_PL <-
plyr::round_any(as.numeric(geo.service.detractors$Property.Latitude_PL), accuracy=.00001,
f=floor)
geo.service.detractors$Property.Longitude_PL <-
plyr::round_any(as.numeric(geo.service.detractors$Property.Longitude_PL), accuracy=.00001,
f=floor)
summary(geo.service.detractors)
head(geo.service.detractors)
describe(as.numeric(geo.service.detractors$Guest_Room_H))
describe(as.numeric(geo.service.detractors$Condition_Hotel_H))
describe(as.numeric(geo.service.detractors$Customer_SVC_H))

# Fix incorrect States
z <- zipcode
z$latitude <- as.character(z$latitude)
z$longitude <- as.character(z$longitude)

state <- z$state
lat <- plyr::round_any(as.numeric(z$latitude), accuracy=.00001, f=floor)
long <- plyr::round_any(as.numeric(z$longitude), accuracy=.00001, f=floor)

stateLatLong <- data.frame(state,lat,long)
head(stateLatLong)

geoDetractors <- geo.service.detractors


# break dataset down by service type, then remove observations that are 7 or heigher (i.e. focus
on detractor level scoring)
remove1 <- c('Customer_SVC_H','Condition_Hotel_H')
gsd.guestRoom <- geo.service.detractors[,!names(geo.service.detractors) %in% remove1]
summary(gsd.guestRoom)
gsd.guestRoom.detractors <-
gsd.guestRoom[as.numeric(gsd.guestRoom$Guest_Room_H)<=6,]
summary(gsd.guestRoom.detractors)
head(gsd.guestRoom.detractors)
guest.detract.tbl <-
table(gsd.guestRoom.detractors$Country_PL,gsd.guestRoom.detractors$Guest_Room_H)
guest.detract.tbl
```

```
remove2 <- c('Customer_SVC_H','Guest_Room_H')
gsd.conditionHotel <- geo.service.detractors[,!names(geo.service.detractors) %in% remove2]
summary(gsd.conditionHotel)
gsd.conditionHotel.detractors <-
gsd.conditionHotel[as.numeric(gsd.conditionHotel$Condition_Hotel_H)<=6,]
summary(gsd.conditionHotel.detractors)
head(gsd.conditionHotel.detractors)
cond.detract.tbl <-
table(gsd.conditionHotel.detractors$Country_PL,gsd.conditionHotel.detractors$Condition_Hotel
_H)
cond.detract.tbl

remove3 <- c('Guest_Room_H','Condition_Hotel_H')
gsd.customerSVC <- geo.service.detractors[,!names(geo.service.detractors) %in% remove3]
summary(gsd.customerSVC)
gsd.customerSVC.detractors <-
gsd.customerSVC[as.numeric(gsd.customerSVC$Customer_SVC_H)<=6,]
summary(gsd.customerSVC.detractors)
head(gsd.customerSVC.detractors)
cust.detract.tbl <-
table(gsd.customerSVC.detractors$Country_PL,gsd.customerSVC.detractors$Customer_SVC_
H)
cust.detract.tbl

# break dataset down by service type, then remove observations that are 7 or heigher (i.e. focus
on detractor level scoring)
remove1 <- c('Customer_SVC_H','Condition_Hotel_H')
gsd.guestRoom <- geo.service.detractors[,!names(geo.service.detractors) %in% remove1]
summary(gsd.guestRoom)
gsd.guestRoom.detractors <-
gsd.guestRoom[as.numeric(gsd.guestRoom$Guest_Room_H)<=6,]
summary(gsd.guestRoom.detractors)
head(gsd.guestRoom.detractors)
guest.detract.tbl <-
table(gsd.guestRoom.detractors$Country_PL,gsd.guestRoom.detractors$Guest_Room_H)
guest.detract.tbl

remove2 <- c('Customer_SVC_H','Guest_Room_H')
gsd.conditionHotel <- geo.service.detractors[,!names(geo.service.detractors) %in% remove2]
summary(gsd.conditionHotel)
gsd.conditionHotel.detractors <-
gsd.conditionHotel[as.numeric(gsd.conditionHotel$Condition_Hotel_H)<=6,]
summary(gsd.conditionHotel.detractors)
```

```
head(gsd.conditionHotel.detractors)
cond.detract.tbl <-
table(gsd.conditionHotel.detractors$Country_PL,gsd.conditionHotel.detractors$Condition_Hotel
_H)
cond.detract.tbl

remove3 <- c('Guest_Room_H','Condition_Hotel_H')
gsd.customerSVC <- geo.service.detractors[,!names(geo.service.detractors) %in% remove3]
summary(gsd.customerSVC)
gsd.customerSVC.detractors <-
gsd.customerSVC[as.numeric(gsd.customerSVC$Customer_SVC_H)<=6,]
summary(gsd.customerSVC.detractors)
head(gsd.customerSVC.detractors)
cust.detract.tbl <-
table(gsd.customerSVC.detractors$Country_PL,gsd.customerSVC.detractors$Customer_SVC_
H)
Cust.detract.tbl


## Subquestion: What proportion of detrctors fall inside the US versus outside?
# 94% of Guest Room detractor's are within the United States
num.us.guestR.detractors <-
nrow(gsd.guestRoom.detractors[gsd.guestRoom.detractors$Country_PL=='United States',])
us.guestR.detractor.proportion <- num.us.guestR.detractors/nrow(gsd.guestRoom.detractors)
round(us.guestR.detractor.proportion,2)*100

# 93% of Condition Hotel detractors are within the United States
num.us.conditionH.detractors <-
nrow(gsd.conditionHotel.detractors[gsd.conditionHotel.detractors$Country_PL=='United
States',])
us.conditionH.detractor.proportion <-
num.us.conditionH.detractors/nrow(gsd.conditionHotel.detractors)
round(us.conditionH.detractor.proportion,2)*100

# 92% of Customer Service detractors are within the United States
num.us.customerSVC.detractors <-
nrow(gsd.customerSVC.detractors[gsd.customerSVC.detractors$Country_PL=='United
States',])
us.customerSVC.detractor.proportion <-
num.us.customerSVC.detractors/nrow(gsd.customerSVC.detractors)
round(us.customerSVC.detractor.proportion,2)*100

#-> 93% of the Top 3 Service detractors are within the United States
```

```
total.us.detractors <-
num.us.guestR.detractors+num.us.conditionH.detractors+num.us.customerSVC.detractors
total.detractos <-
nrow(gsd.guestRoom.detractors)+nrow(gsd.conditionHotel.detractors)+nrow(gsd.customerSVC.
detractors)
total.us.detractor.proportion <- total.us.detractors/total.detractos
round(total.us.detractor.proportion,2)*100


## Subquestion: What proportion of detrctors fall inside the US versus outside?
# 94% of Guest Room detractor's are within the United States
num.us.guestR.detractors <-
nrow(gsd.guestRoom.detractors[gsd.guestRoom.detractors$Country_PL=='United States',])
us.guestR.detractor.proportion <- num.us.guestR.detractors/nrow(gsd.guestRoom.detractors)
round(us.guestR.detractor.proportion,2)*100

# 93% of Condition Hotel detractors are within the United States
num.us.conditionH.detractors <-
nrow(gsd.conditionHotel.detractors[gsd.conditionHotel.detractors$Country_PL=='United
States',])
us.conditionH.detractor.proportion <-
num.us.conditionH.detractors/nrow(gsd.conditionHotel.detractors)
round(us.conditionH.detractor.proportion,2)*100

# 92% of Customer Service detractors are within the United States
num.us.customerSVC.detractors <-
nrow(gsd.customerSVC.detractors[gsd.customerSVC.detractors$Country_PL=='United
States',])
us.customerSVC.detractor.proportion <-
num.us.customerSVC.detractors/nrow(gsd.customerSVC.detractors)
round(us.customerSVC.detractor.proportion,2)*100

#-> 93% of the Top 3 Service detractors are within the United States
total.us.detractors <-
num.us.guestR.detractors+num.us.conditionH.detractors+num.us.customerSVC.detractors
total.detractos <-
nrow(gsd.guestRoom.detractors)+nrow(gsd.conditionHotel.detractors)+nrow(gsd.customerSVC.
detractors)
total.us.detractor.proportion <- total.us.detractors/total.detractos
round(total.us.detractor.proportion,2)*100

##############################################################################
##############################################################################
## Function to classify the service rating in to levels, LOW[1-2], MID[3-4], HIGH[5-6]
```

```
classifyDetractorRange <- function(ds, serviceType){
  ds$Detractor_Level <- NA
  switch(serviceType,
         "GUEST" = {
           ds <- transform(ds, Detractor_Level = ifelse(Guest_Room_H == 1, "LOW",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Guest_Room_H == 2, "LOW",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Guest_Room_H == 3, "MID",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Guest_Room_H == 4, "MID",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Guest_Room_H == 5, "HIGH",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Guest_Room_H == 6, "HIGH",
Detractor_Level))
         },
         "CUSTOMER" = {
           ds <- transform(ds, Detractor_Level = ifelse(Customer_SVC_H == 1, "LOW",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Customer_SVC_H == 2, "LOW",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Customer_SVC_H == 3, "MID",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Customer_SVC_H == 4, "MID",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Customer_SVC_H == 5, "HIGH",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Customer_SVC_H == 6, "HIGH",
Detractor_Level))
         },
         "CONDITION" = {
           ds <- transform(ds, Detractor_Level = ifelse(Condition_Hotel_H == 1, "LOW",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Condition_Hotel_H == 2, "LOW",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Condition_Hotel_H == 3, "MID",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Condition_Hotel_H == 4, "MID",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Condition_Hotel_H == 5, "HIGH",
Detractor_Level))
           ds <- transform(ds, Detractor_Level = ifelse(Condition_Hotel_H == 6, "HIGH",
Detractor_Level))
```

```
        }

        )

  return(ds)
}
##---- Plot US Maps ------
us.map <- map_data("state")

removeThemeAxis <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank()
)
## GUEST_ROOM DETRACTORS US ##
# filter detractor data frames to focus on US
gsd.us.guestRoom.detractors <-
gsd.guestRoom.detractors[gsd.guestRoom.detractors$Country_PL=='United States',]
gsd.us.guestRoom.detractors <-
classifyDetractorRange(gsd.us.guestRoom.detractors,"GUEST")
head(gsd.us.guestRoom.detractors[order(gsd.us.guestRoom.detractors[,3]),])

# Table of unique hotels by STATE_R
guestDf <- gsd.us.guestRoom.detractors
guestDf$STATE_R <- as.character(guestDf$STATE_R)
names(guestDf)

guestDf$state_name <- tolower(state.name[match(guestDf$STATE_R,state.abb)])
guestDf <- subset(guestDf,!is.na(state_name))
head(guestDf[order(guestDf[,2],decreasing = TRUE),],5)

guestRoomDetractors.by.state.hotelCount <- sqldf("select STATE_R,
count('Hotel.Name.Long_PL') as Hotel_Count,
                                    round(avg(Guest_Room_H),2) as Guest_Room_Avg,
                                    count(case when Detractor_Level='LOW' then 1 else null end) as
Detractor_Low_Cnt,
                                    count(case when Detractor_Level='MID' then 1 else null end) as
Detractor_Mid_Cnt,
                                    count(case when Detractor_Level='HIGH' then 1 else null end)
as Detractor_High_Cnt
```

```
                                    from guestDf where STATE_R <> 'NOT_LISTED' group by
STATE_R")
head(guestRoomDetractors.by.state.hotelCount[order(guestRoomDetractors.by.state.hotelCoun
t[,2],decreasing = TRUE),],7)

# Some state abbreviations are wrong in the survey data set, remove them from the dataset
guestRoomDetractors.by.state.hotelCount$state_name <-
tolower(state.name[match(guestRoomDetractors.by.state.hotelCount$STATE_R,state.abb)])
guestRoomDetractors.by.state.hotelCount <-
subset(guestRoomDetractors.by.state.hotelCount,!is.na(state_name))


#--- Map Plotting Detractors - Guest Room Rating, US Map by State & Hotel Count
# Top 5 States with the most number of Hotels with a Guest Room Service Rating as
Detractors:
# 1: TX with 125 Hotels
# 2: CA with 86 Hotels
# 3: FL with 68 Hotels
# 4: NC with 45 Hotels
# 5: IL with 37 Hotels
#---
guestRoomDetractors.by.state.hotelCount[order(guestRoomDetractors.by.state.hotelCount[,2],d
ecreasing = TRUE),]
map.detractors.guestR <- ggplot(data=guestRoomDetractors.by.state.hotelCount,
mapping=aes(map_id=state_name))
map.detractors.guestR <- map.detractors.guestR + geom_map(map=us.map,
mapping=aes(fill=Hotel_Count))
map.detractors.guestR <- map.detractors.guestR + scale_fill_gradient2(low="#559999",
mid="grey90", high="#BB650B",midpoint =
mean(guestRoomDetractors.by.state.hotelCount$Hotel_Count))
map.detractors.guestR <- map.detractors.guestR + expand_limits(x=us.map$long,
y=us.map$lat)

map.detractors.guestR <- map.detractors.guestR + coord_map("polyconic")
map.detractors.guestR <- map.detractors.guestR + ggtitle("Guest Room Service Rating,
Detractors by State") + theme(plot.title=element_text(hjust=0.5))
map.detractors.guestR <- map.detractors.guestR + guides(fill=guide_legend(title="State Hotel
Count")) + removeThemeAxis
map.detractors.guestR
ggsave("U.S._Map_of_Service_Detractors_Guest_Room.jpg", width = 6, height = 6)


## CONDITION HOTEL DETRACTORS
```

```
gsd.us.conditionHotel.detractors <-
gsd.conditionHotel.detractors[gsd.conditionHotel.detractors$Country_PL=='United States',]
gsd.us.conditionHotel.detractors <-
classifyDetractorRange(gsd.us.conditionHotel.detractors,"CONDITION")
head(gsd.us.conditionHotel.detractors[order(gsd.us.conditionHotel.detractors[,4]),])
conditionDf <- gsd.us.conditionHotel.detractors
conditionDetractors.by.state.hotelCount <- sqldf("select STATE_R,
count('Hotel.Name.Long_PL') as Hotel_Count,
                                    round(avg(Condition_Hotel_H),2) as Condition_Avg,
                                    count(case when Detractor_Level='LOW' then 1 else null end) as
Detractor_Low_Cnt,
                                    count(case when Detractor_Level='MID' then 1 else null end) as
Detractor_Mid_Cnt,
                                    count(case when Detractor_Level='HIGH' then 1 else null end)
as Detractor_High_Cnt
                                    from conditionDf where STATE_R <> 'NOT_LISTED' group by
STATE_R")
head(conditionDetractors.by.state.hotelCount[order(conditionDetractors.by.state.hotelCount[,2],
decreasing = TRUE),],5)

#--- Map Plotting Detractors - Condition Hotel Rating, US Map by State & Hotel Count
# Top 5 States with the most number of Hotels with a Hotel Condition Service Rating as
Detractors:
# 1: TX with 109 Hotels
# 2: CA with 83 Hotels
# 3: FL with 53 Hotels
# 4: NC with 38 Hotels
# 5: NY with 33 Hotels
#---
conditionDetractors.by.state.hotelCount[order(conditionDetractors.by.state.hotelCount[,2],decre
asing = TRUE),]
# Some state abbreviations are wrong in the survey data set, remove them from the dataset
conditionDetractors.by.state.hotelCount[order(conditionDetractors.by.state.hotelCount[,2],decre
asing = TRUE),]
conditionDetractors.by.state.hotelCount$state_name <-
tolower(state.name[match(conditionDetractors.by.state.hotelCount$STATE_R,state.abb)])
conditionDetractors.by.state.hotelCount <-
subset(conditionDetractors.by.state.hotelCount,!is.na(state_name))

conditionDetractors.by.state.hotelCount[order(conditionDetractors.by.state.hotelCount[,2],decre
asing = TRUE),]
map.detractors.condition <- ggplot(data=conditionDetractors.by.state.hotelCount,
mapping=aes(map_id=state_name))
```

```
map.detractors.condition <- map.detractors.condition + geom_map(map=us.map,
mapping=aes(fill=Hotel_Count))
map.detractors.condition <- map.detractors.condition + scale_fill_gradient2(low="#559999",
mid="grey90", high="#BB650B",midpoint =
mean(conditionDetractors.by.state.hotelCount$Hotel_Count))
map.detractors.condition <- map.detractors.condition + expand_limits(x=us.map$long,
y=us.map$lat)
map.detractors.condition <- map.detractors.condition + coord_map()
map.detractors.condition <- map.detractors.condition + ggtitle("Hotel Condition Service Rating,
Detractors by State") + theme(plot.title=element_text(hjust=0.5))
map.detractors.condition <- map.detractors.condition + guides(fill=guide_legend(title="State
Hotel Count")) + removeThemeAxis
map.detractors.condition
ggsave("U.S._Map_of_Service_Detractors_Condition.jpg", width = 6, height = 6)


## CUSTOMER SERVICE DETRACTORS
gsd.us.customerSVC.detractors <-
gsd.customerSVC.detractors[gsd.customerSVC.detractors$Country_PL=='United States',]
gsd.us.customerSVC.detractors <-
classifyDetractorRange(gsd.us.customerSVC.detractors,"CUSTOMER")
head(gsd.us.customerSVC.detractors[order(gsd.us.customerSVC.detractors[,3]),])
customerSVCDf <- gsd.us.customerSVC.detractors
customerDetractors.by.state.hotelCount <- sqldf("select STATE_R,
count('Hotel.Name.Long_PL') as Hotel_Count,
                                round(avg(Customer_SVC_H),2) as CustomerSVC_Avg,
                                count(case when Detractor_Level='LOW' then 1 else null end) as
Detractor_Low_Cnt,
                                count(case when Detractor_Level='MID' then 1 else null end) as
Detractor_Mid_Cnt,
                                count(case when Detractor_Level='HIGH' then 1 else null end)
as Detractor_High_Cnt
                                from customerSVCDf where STATE_R <> 'NOT_LISTED' group
by STATE_R")
head(customerDetractors.by.state.hotelCount[order(customerDetractors.by.state.hotelCount[,2],
decreasing = TRUE),],7)

#--- Map Plotting Detractors - Customer Service Rating, US Map by State & Hotel Count
# Top 5 States with the most number of Hotels with a Hotel Condition Service Rating as
Detractors:
# 1: TX with 102 Hotels
# 2: CA with 80 Hotels
# 3: FL with 47 Hotels
# 4: IL with 39 Hotels
```

```
# 5: NY with 32 Hotels
#---
customerDetractors.by.state.hotelCount[order(customerDetractors.by.state.hotelCount[,2],decreasing = TRUE),]
# Some state abbreviations are wrong in the survey data set, remove them from the dataset
customerDetractors.by.state.hotelCount$state_name <- tolower(state.name[match(customerDetractors.by.state.hotelCount$STATE_R,state.abb)])
customerDetractors.by.state.hotelCount <- subset(customerDetractors.by.state.hotelCount,!is.na(state_name))

customerDetractors.by.state.hotelCount[order(customerDetractors.by.state.hotelCount[,2],decreasing = TRUE),]
map.detractors.customer <- ggplot(data=customerDetractors.by.state.hotelCount, mapping=aes(map_id=state_name))
map.detractors.customer <- map.detractors.customer + geom_map(map=us.map, mapping=aes(fill=Hotel_Count))
map.detractors.customer <- map.detractors.customer + scale_fill_gradient2(low="#559999", mid="grey90", high="#BB650B",midpoint = mean(customerDetractors.by.state.hotelCount$Hotel_Count))
map.detractors.customer <- map.detractors.customer + expand_limits(x=us.map$long, y=us.map$lat)
map.detractors.customer <- map.detractors.customer + coord_map()
map.detractors.customer <- map.detractors.customer + ggtitle("Customer Service Rating, Detractors by State") + theme(plot.title=element_text(hjust=0.5))
map.detractors.customer <- map.detractors.customer + guides(fill=guide_legend(title="State Hotel Count")) + removeThemeAxis
map.detractors.customer
ggsave("U.S._Map_of_Service_Detractors_Customer.jpg", width = 6, height = 6)
```