IST 687 - Intro to Data Science
Lab - M008

Latent Analytics Pvt Ltd

Team 1:
Sanjeev Ramasamy Seenivasagamani, Woojin Park, Xiaoyan Zhang, Jeet Ganatra, Christy Sato, and Aniruddh Garge

Table of Contents

Introduction/Business Questions……...……..……... 2

Data Cleaning..……..... 3

Descriptive Statistics & Visualizations…… 5

Code for Descriptive Statistics & Visualizations…...…… 34

Linear Modelling…...……... 42

Code for Linear Modelling……... 48

Logistic Regression …...…….. 50

Code for Logistic Regression …….…….. 57

Support Vector Machines……..
61

Code for Support Vector Machines……...
63

Association Rules Mining……. 85

Code for Association Rules……...……... 89

Actionable Insights…….……...…….. 91

MIDST Link……... 92

Introduction

The given dataset was based on a flight survey that had sample data on various attributes like satisfaction, age, gender, and so much more. With a total of 27 variables and 129,889 observations, we understood that it was a relatively big dataset that needed some cleaning, so that eventually we could focus on analyzing and understanding what we can get from the dataset. With this big dataset, we are able to view many different angles of information to help solve various problems, strategies, and business decisions.

The project will use various data analysis models like SVM, prediction analysis, logistic regression, etc. We will predict the correlation trends of the variables in the dataset with the satisfaction of every customer by showcasing the variables visually to represent the data and provide accurate results for the business questions that will be formulated subsequently.

Our first step before approaching our analysis was the data acquisition process. Through this phase, we can conclude that the data was collected through a survey that randomly sampled 129,889 customers. Then, we viewed all the variables given to us and determined what the different values mean for each column. For example, looking at the first column, Satisfaction, you can see numbers ranging from 1-5. Without understanding the values, there is not much we can do. But after taking a closer look, we now understand that the column holds values that customers rate their satisfaction from their flight, which ranges from 1, being the least satisfied, to 5, being the most satisfied. We followed this process for each and every column.

Once that is completed, we began to check for any missing values or other ways to clean the dataset given to us. Then we discussed as a group, what questions that has come up based on the data, and how do we want to help airlines improve overall customer satisfaction.

Business Questions

The aim of the project is to improve the satisfaction rate of customers flying with their respective airlines. By reviewing and analyzing the different variables present in the dataset, we answered several business questions to improve the satisfaction of customers. The business questions are as follows:

1. What variables are good predictors of high customer satisfaction?
2. How can airline companies improve customer satisfaction?
3. Does class, gender, or age correlate with high satisfaction rate?
4. Does the status of an airline affect the overall satisfaction rate of customers?
5. How does flight arrival delay affect the customer satisfaction rate?

6. What is the difference between the satisfaction rating provided by males and females?

Data Cleaning

We first had to read in the dataset:

```
#installing the required packages and libraries
install.packages("readxl")
install.packages("rjson")
library(rjson)
library(readxl)

unclean_data<-data.frame(read.csv(file.choose()), stringsAsFactors=FALSE)

#make Satisfaction column as numeric values instead of characters
unclean_data$Satisfaction<-as.numeric(as.character(unclean_data$Satisfaction))
```

We then wanted to see if there were any missing values in the dataset:

```
#Find the Columns where the NAs are
colnames(unclean_data)[colSums(is.na(unclean_data))>0]
```

We notice that Departure Delay in Minutes, Arrival Delay in Minutes, Flight Time in Minutes, and Satisfaction columns have NAs. We looked into how many missing values are in each of these columns:

```
#Number of missing values
sum(is.na(unclean_data$Departure.Delay.in.Minutes))
#2345
sum(is.na(unclean_data$Arrival.Delay.in.Minutes))
#2738
sum(is.na(unclean_data$Flight.time.in.minutes))
#2738
sum(is.na(unclean_data$Satisfaction))
#3
```

Using the summary function, we decided our best approach to fixing the NAs are replacing the NAs with the medians of each column. Medians are not sensitive to outliers which can help with our analysis:

```
summary(unclean_data$Departure.Delay.in.Minutes)
   Min.   1st Qu.    Median    Mean   3rd Qu.    Max.       NA's
   0.00    0.00      0.00      14.98  13.00    1592.00    2345

summary(unclean_data$Arrival.Delay.in.Minutes)
 Min.    1st Qu.  Median    Mean    3rd Qu.    Max.       NA's
 0.00    0.00     0.00      15.37     13.00   1584.00    2738
summary(unclean_data$Flight.time.in.minutes)
```

```
 Min.   1st Qu.    Median    Mean    3rd Qu.      Max.       NA's
 8.0      59.0       92.0     111.5     142.0      669.0      2738

summary(unclean_data$Satisfaction)
 Min.       1st Qu.     Median     Mean      3rd Qu.       Max.      NA's
1.000       3.000       4.000      3.379      4.000       5.000       3
```

Now we can replace the NA's:

```
unclean_data$Departure.Delay.in.Minutes[is.na(unclean_data$Departure.Delay.in.Minutes)] <-
median(unclean_data$Departure.Delay.in.Minutes, na.rm=TRUE)
unclean_data$Arrival.Delay.in.Minutes[is.na(unclean_data$Arrival.Delay.in.Minutes)] <-
mean(unclean_data$Arrival.Delay.in.Minutes, na.rm=TRUE)
unclean_data$Flight.time.in.minutes[is.na(unclean_data$Flight.time.in.minutes)] <-
median(unclean_data$Flight.time.in.minutes, na.rm=TRUE)
unclean_data$Satisfaction[is.na(unclean_data$Satisfaction)] <- mean(unclean_data$Satisfaction, na.rm=TRUE)

#Summary of the three variables after removing the missing variables
summary(unclean_data$Departure.Delay.in.Minutes)
 Min.    1st Qu.    Median    Mean    3rd Qu.       Max.
 0.00      0.00       0.00      14.71    12.00       1592.00
summary(unclean_data$Arrival.Delay.in.Minutes)
  Min.    1st Qu.    Median    Mean     3rd Qu.       Max.
  0.00    0.00       0.00      15.37     15.00       1584.00
summary(unclean_data$Flight.time.in.minutes)
   Min.    1st Qu.    Median    Mean      3rd Qu.      Max.
    8.0      60.0       92.0      111.1      141.0      669.0
summary(unclean_data$Satisfaction)
   Min.      1st Qu.    Median    Mean    3rd Qu.    Max.
  1.000      3.000      4.000      3.379     4.000      5.000
```

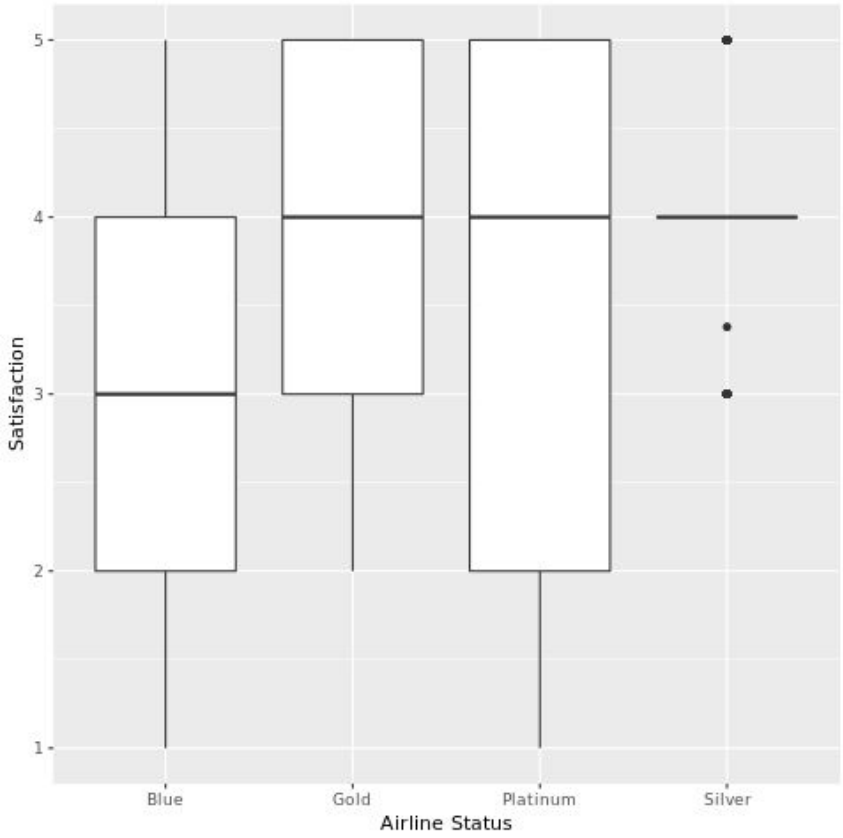
Now that data has been cleaned and corrected, can make the dataset as a data frame:

```
data <- as.data.frame(unclean_data)
```

Descriptive Statistics & Visualizations

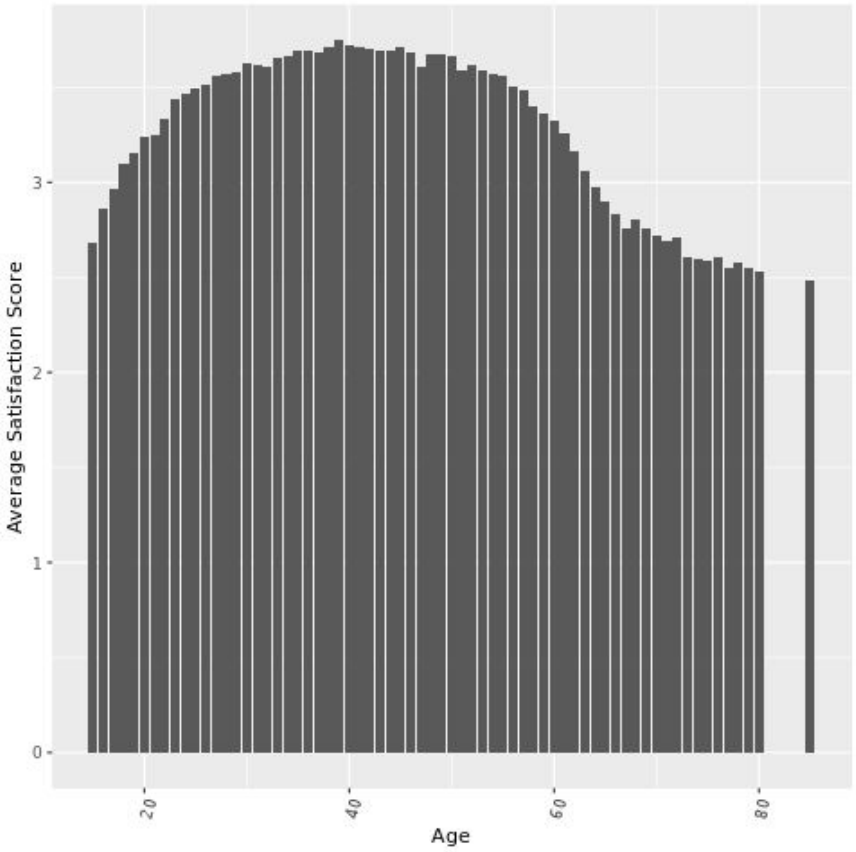
1. Airline Status

Airline Status variable shows which package or status they chose to fly in. These status are of four types namely Gold, Platinum, Silver & Blue.



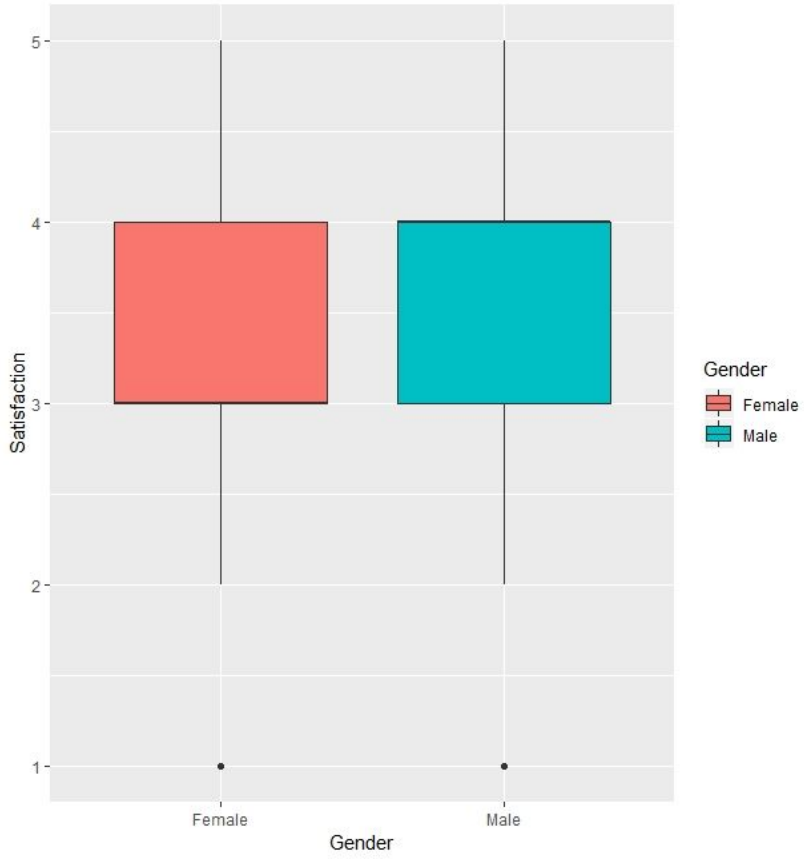
2. **Age**

In this variable we can find the age of the customers and the distribution of this variable can help us understand the customers exhaustively. As we can see, customers within the age group of 30 - 50 provide the maximum satisfaction score.

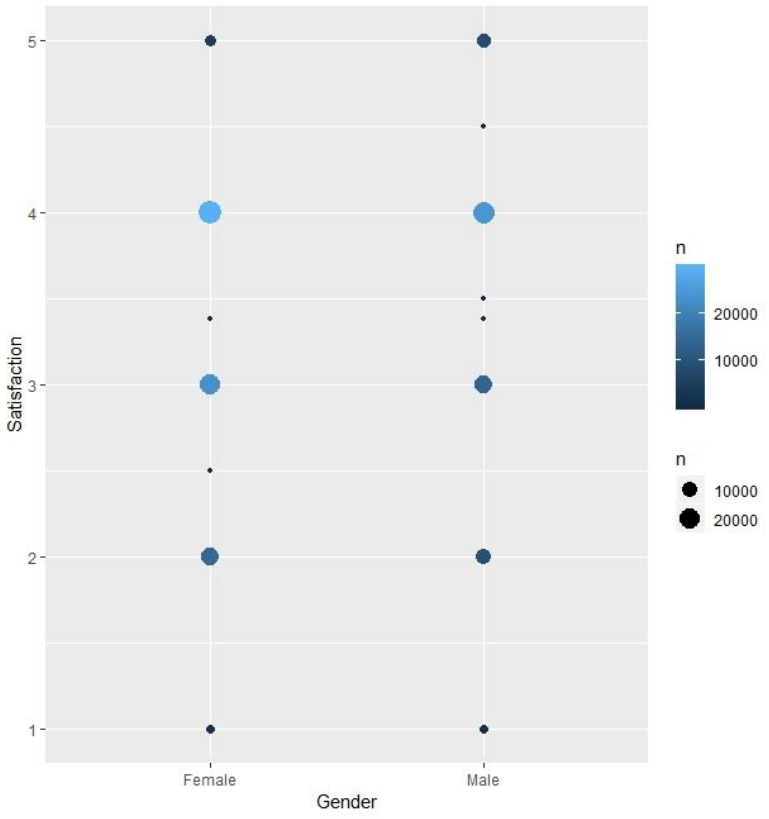


3. Gender

We can use the Gender variable of the dataset to see the gender-wise distribution of the satisfaction score.

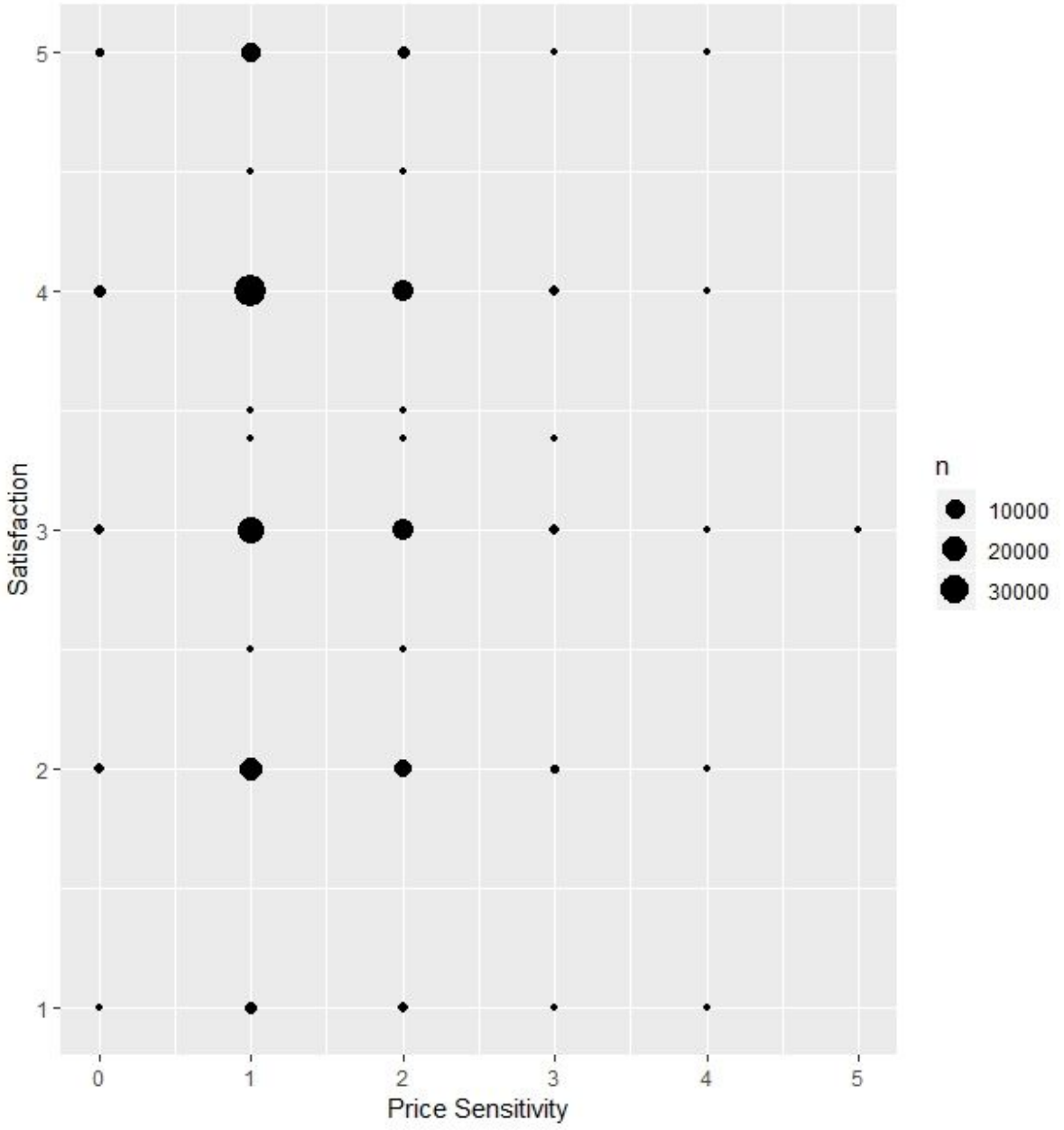


In a different plot below, we can see that most of male customers provide higher satisfaction scores than their counterparts.

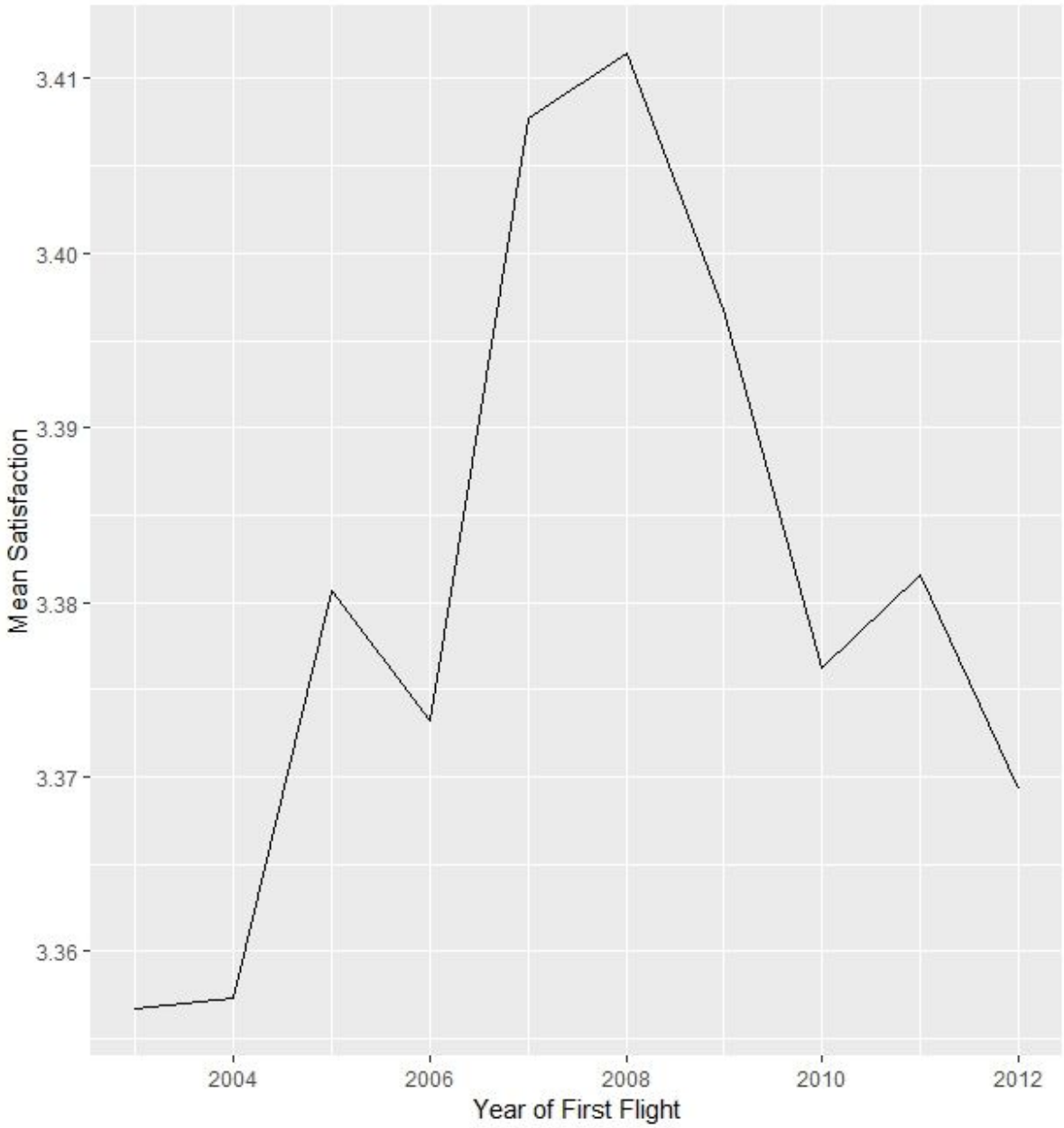

5
4
3
2
1
Satisfaction
Female
Male
Gender
n
20000
10000
n
10000
20000

4. Price Sensitivity

This variable provides the grade to which the price affects to customers purchasing with a range of 0 to 5.



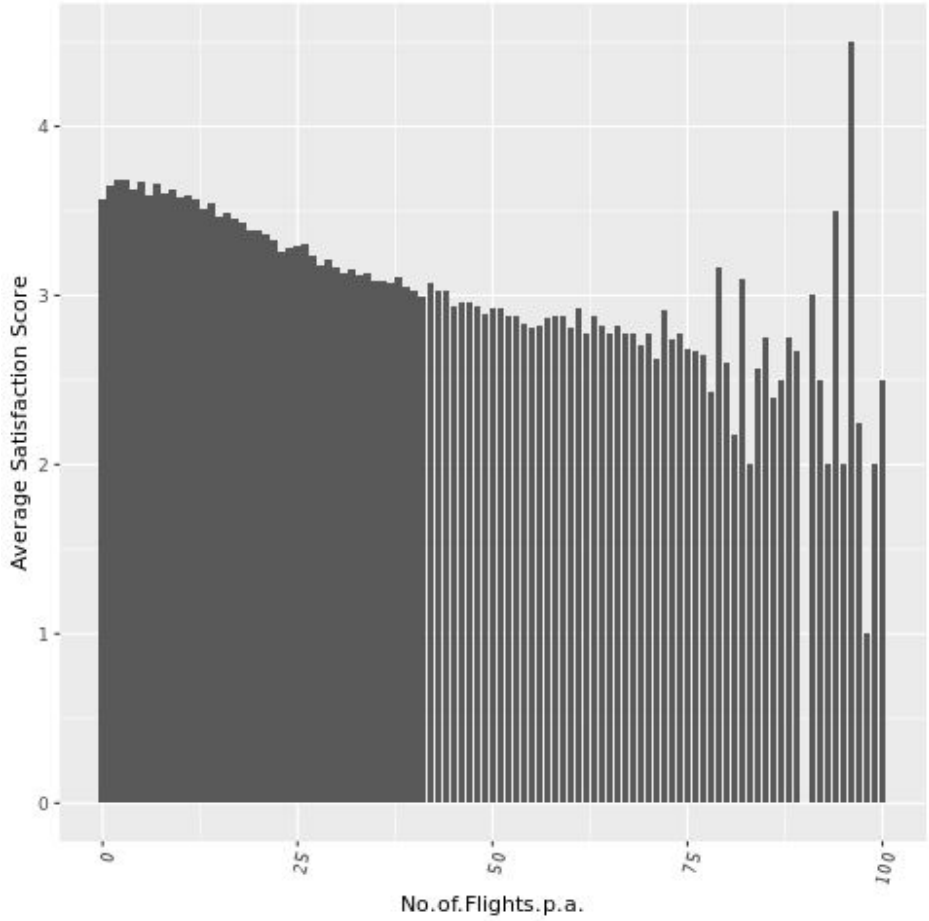
5. Year of First Flight



This variable indicates the year in which the customer has ever flew. The years range from 2002 till 2012. We can clearly witness that the peak is in 2008.

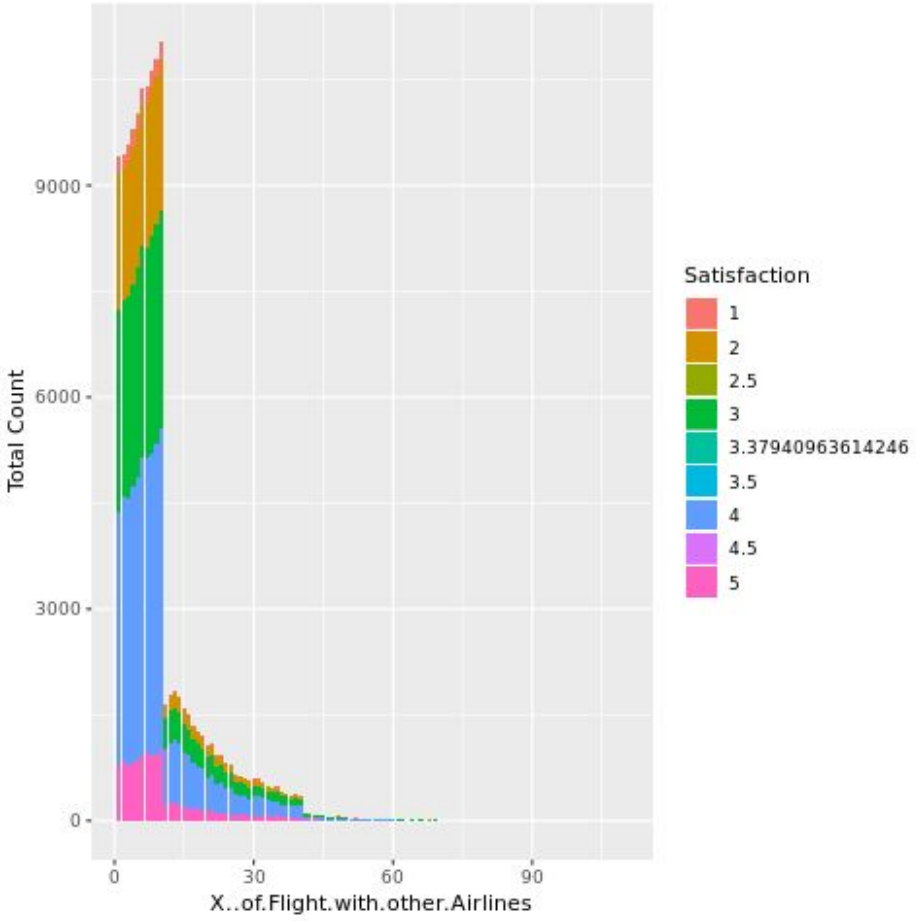
6. Number of Flights P.A

This variable gives us an idea how customers with the given number of flights per annum rate the satisfaction.



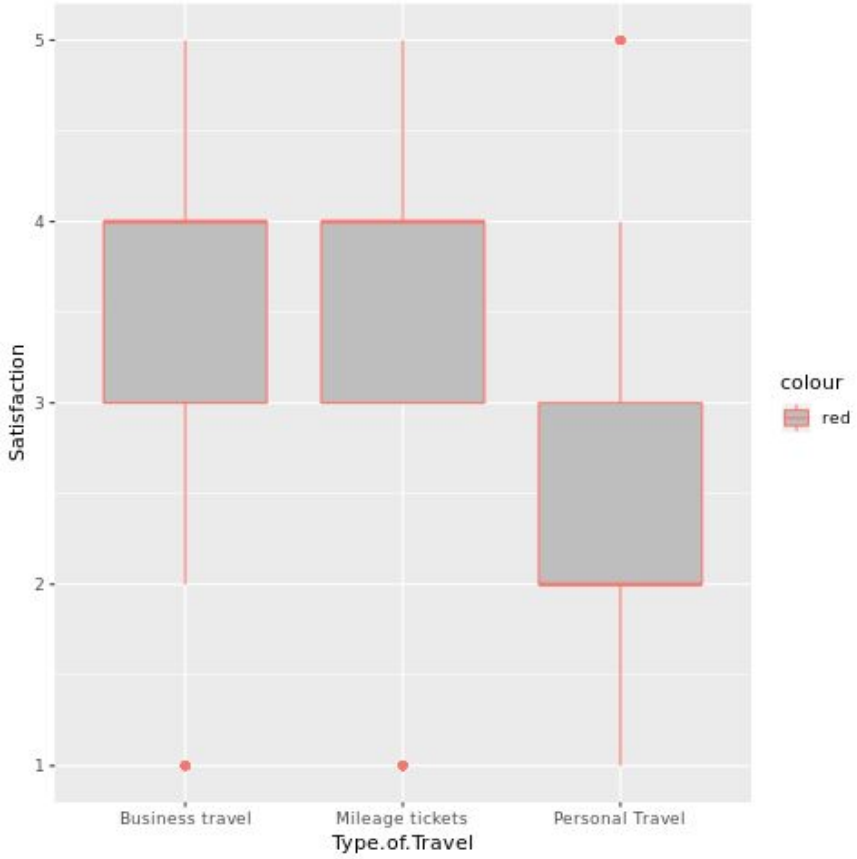
7. **Flight with Other Airlines**

This variable indicates that customer who had low number of flight with other airlines has tendency to have higher satisfaction.



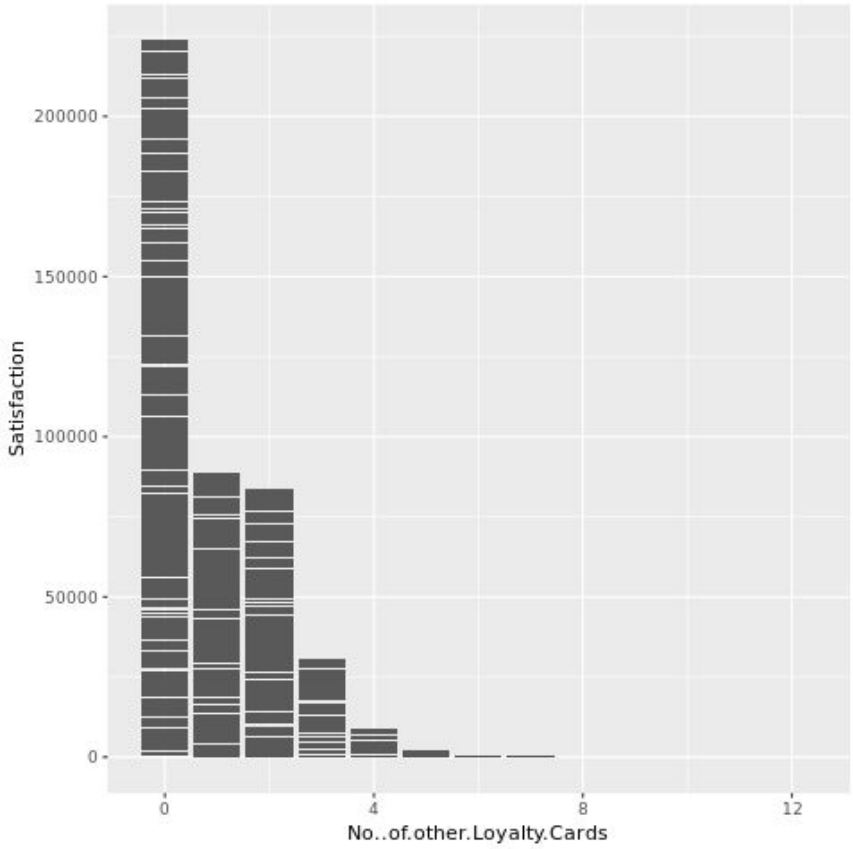
8. Type of Travel

This variable gives us the insight that customer who have experienced with Business travel and Mileage tickets has higher satisfaction rating than Personal Travel group.



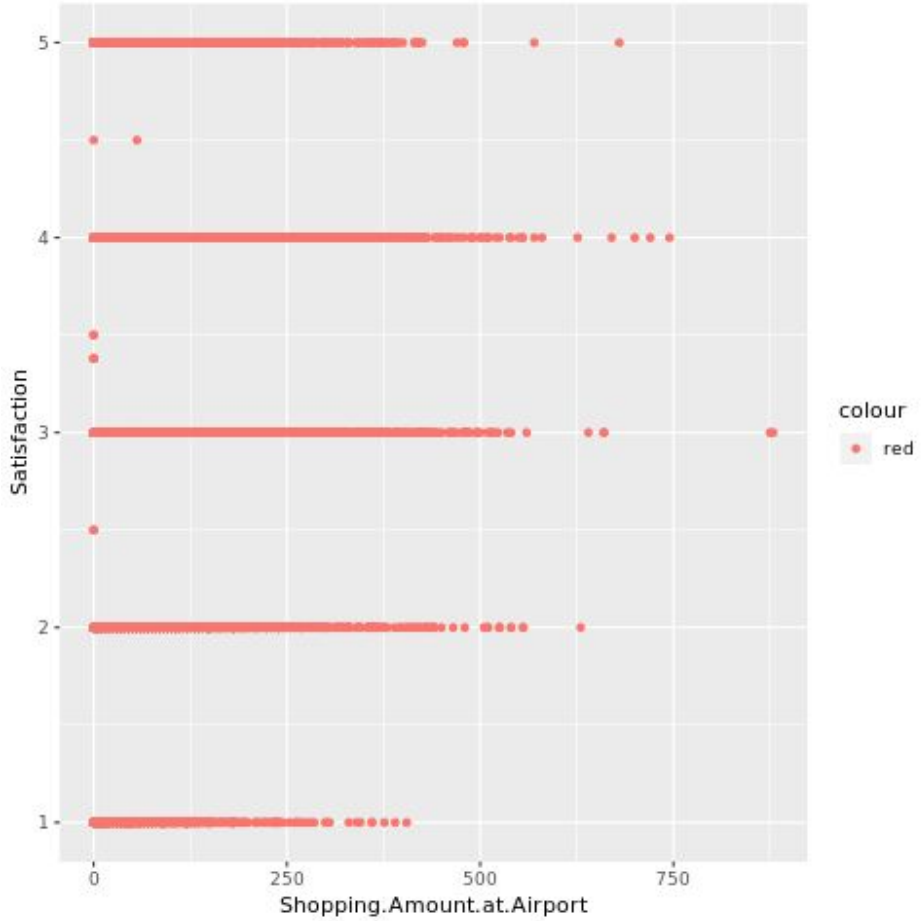
9. Number of Loyalty Cards

This variable gives us clear idea that almost half of customers have 0 different loyalty cards and another half have lower than 4 different loyalty cards.



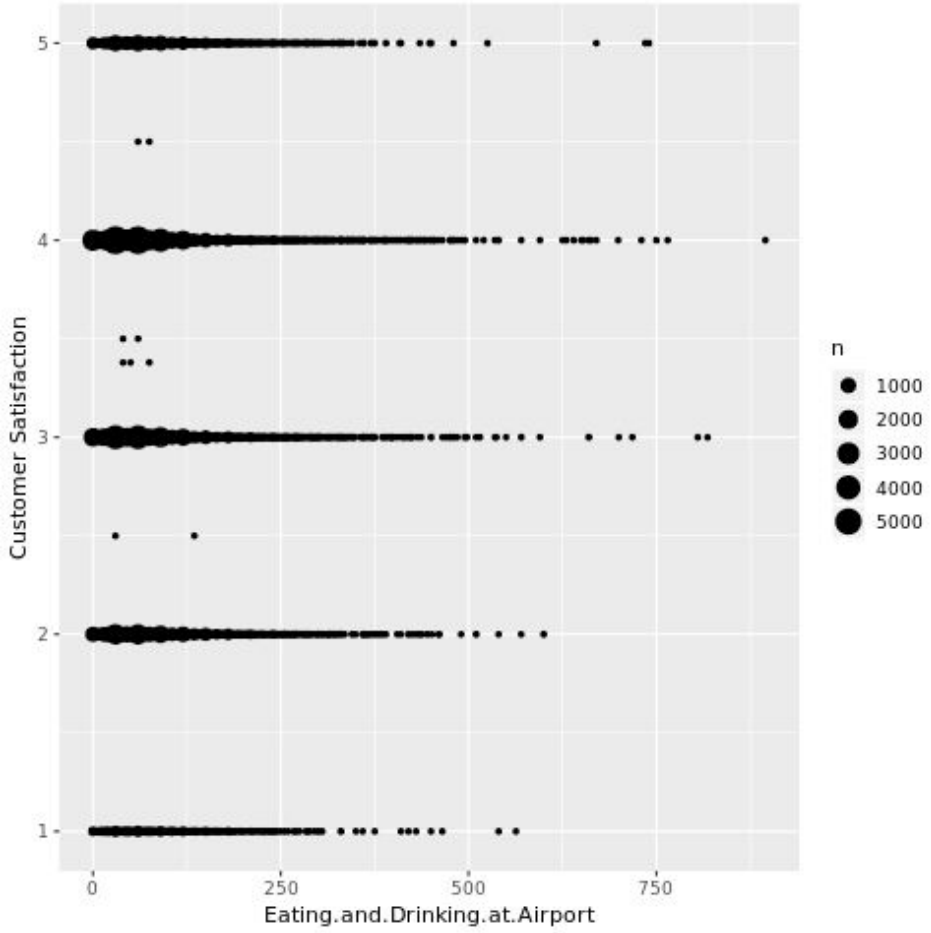
10. Shopping Amount at Airport

This variable gives an idea that how many items are purchased by customers and the range of it is mostly between 0-500. And the purchase number doesn't seem to have any effect on satisfaction rating.

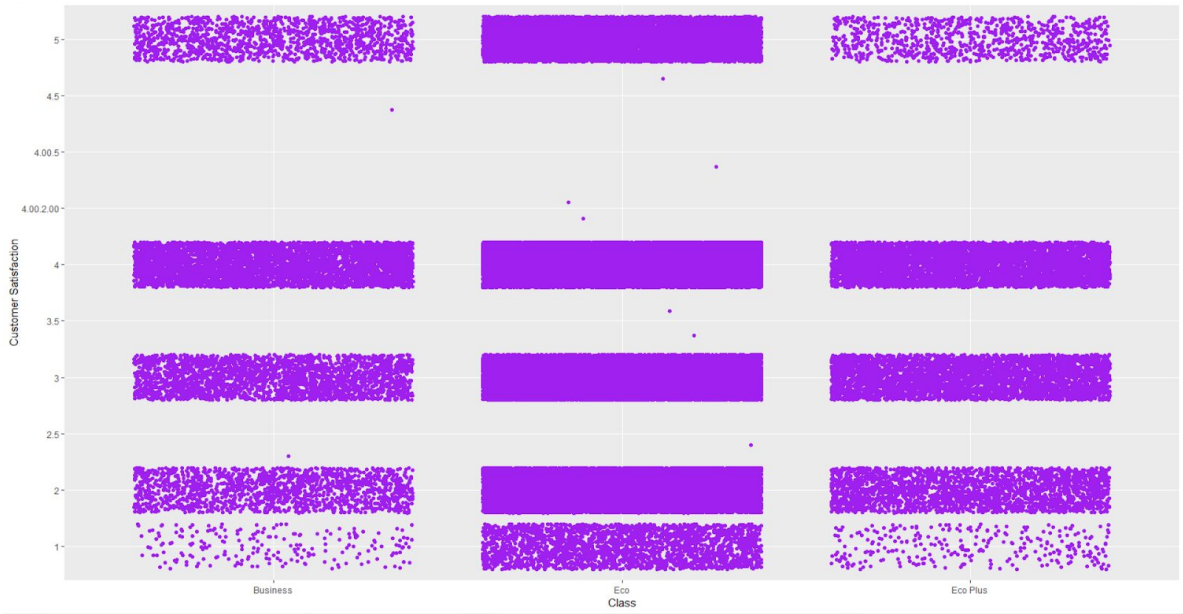


11. Eating and Drinking at Airport

This variable indicates that quantity eating and drinking per each consumers at the most of the customers are located between 0 to 250 frequency and they have tendency to have slightly high satisfaction late which is around 3-4.



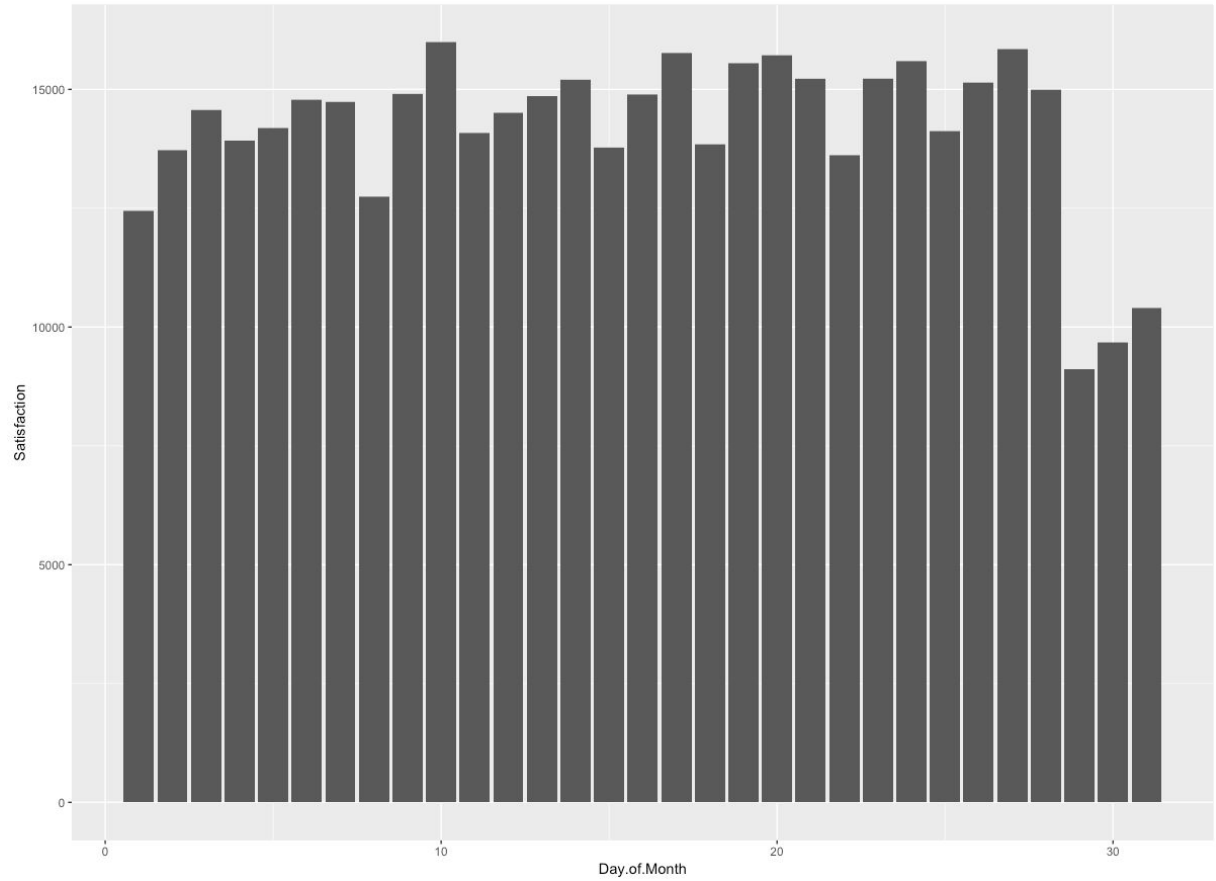
12. Class



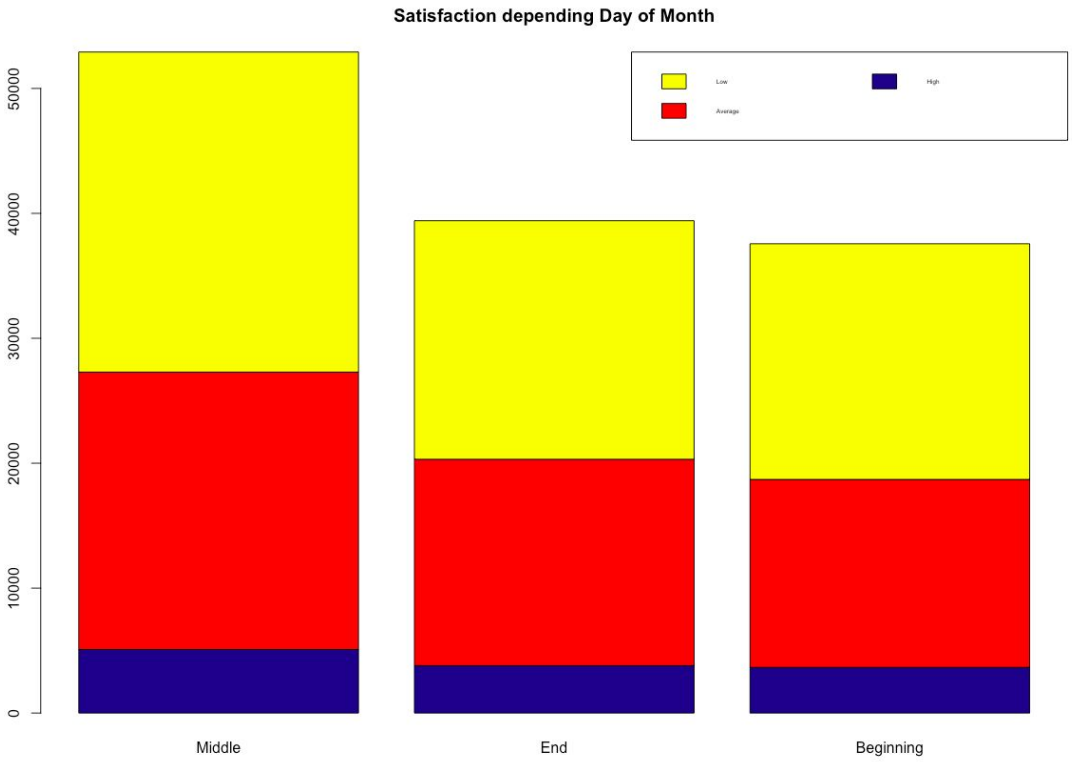
Customers travelling in business class and economy plus class majorly have a satisfaction rate of 3 to 5 while customers travelling in economy class have significant numbers in every batch of satisfaction rate. So one inference we can make is more seats of business and economy class should be introduced to get an overall high satisfaction rate and the services in economy class should be improved.

13. Day of Month

The plot below provides insight on satisfaction ratings based on the day of the month.



Decided to also try looking at the day of month as buckets, separating the day of months into three buckets. The buckets will hold the beginning, middle, and end of the month, to see what if customer satisfaction depends on when they travelled during the month. Customer satisfaction is also split into three separate buckets; high, low, and average satisfaction ratings. Unfortunately, the plot shows that the middle of the month has larger yellow (low), red (average), and blue (high) sections, which does not prove that a certain time of the month has a better satisfaction.

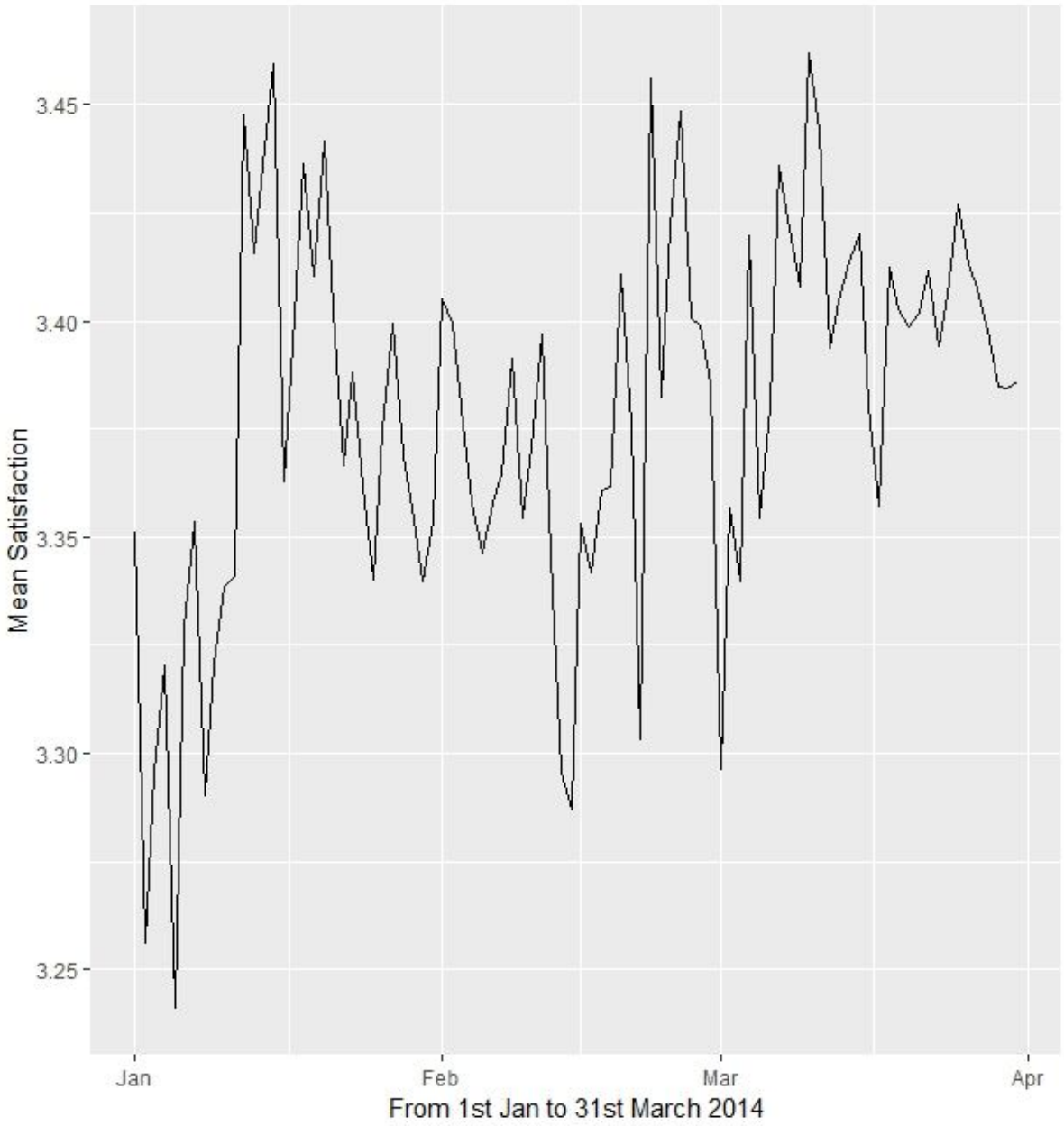


14. Flight Date

A summary look at Flight date:

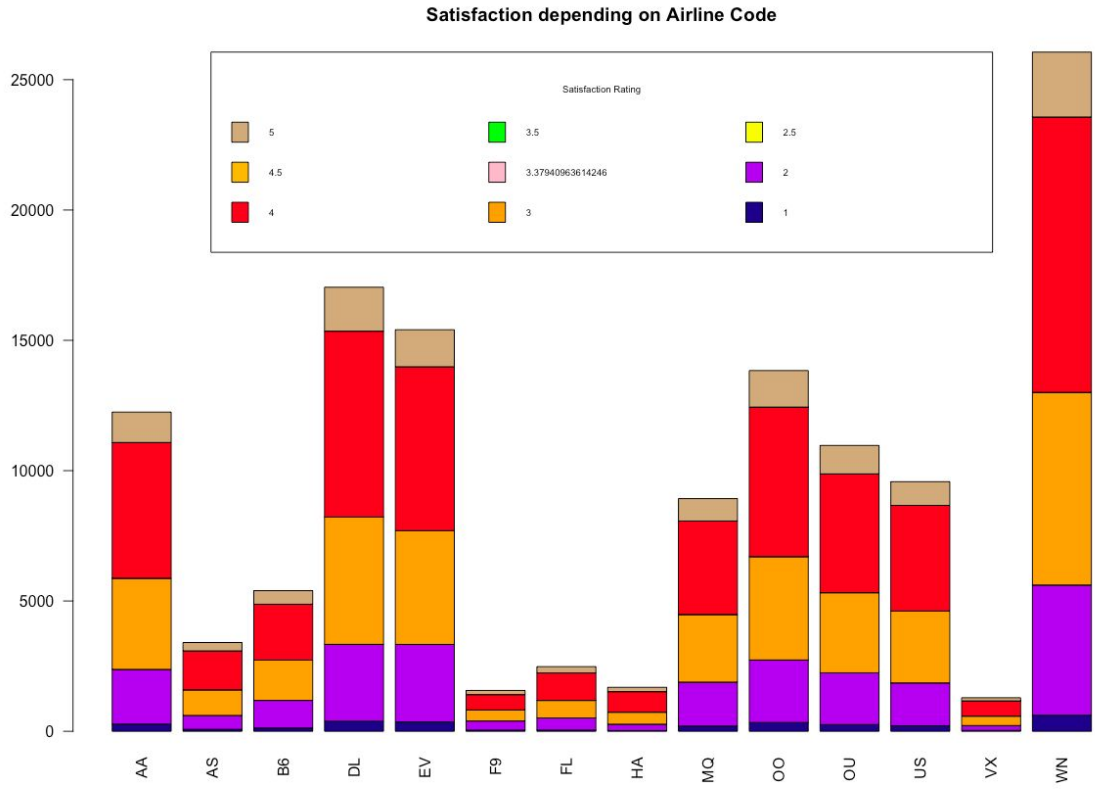
1/1/14	1/10/14	1/11/14	1/12/14	1/13/14	1/14/14	1/15/14	1/16/14	1/17/14	1/18/14	1/19/14	1/2/14	1/20/14
1349	1571	1147	1387	1494	1459	1469	1496	1553	1066	1431	1437	1469
1/21/14	1/22/14	1/23/14	1/24/14	1/25/14	1/26/14	1/27/14	1/28/14	1/29/14	1/3/14	1/30/14	1/31/14	1/4/14
1316	1425	1485	1433	1125	1374	1495	1268	1368	1431	1402	1510	1265
1/5/14	1/6/14	1/7/14	1/8/14	1/9/14	2/1/14	2/10/14	2/11/14	2/12/14	2/13/14	2/14/14	2/15/14	2/16/14
1397	1378	1304	1379	1540	1121	1503	1428	1313	1262	1471	1201	1445
2/17/14	2/18/14	2/19/14	2/2/14	2/20/14	2/21/14	2/22/14	2/23/14	2/24/14	2/25/14	2/26/14	2/27/14	2/28/14
1505	1465	1536	1292	1557	1564	1228	1459	1537	1476	1453	1539	1587
2/3/14	2/4/14	2/5/14	2/6/14	2/7/14	2/8/14	2/9/14	3/1/14	3/10/14	3/11/14	3/12/14	3/13/14	3/14/14
1494	1400	1364	1405	1468	1094	1392	1245	1640	1567	1570	1641	1564
3/15/14	3/16/14	3/17/14	3/18/14	3/19/14	3/2/14	3/20/14	3/21/14	3/22/14	3/23/14	3/24/14	3/25/14	3/26/14
1362	1488	1615	1538	1595	1384	1588	1638	1339	1547	1619	1537	1628
3/27/14	3/28/14	3/29/14	3/3/14	3/30/14	3/31/14	3/4/14	3/5/14	3/6/14	3/7/14	3/8/14	3/9/14	
1622	1572	1333	1436	1473	1575	1466	1517	1618	1576	1313	1501	

As you can see, the dates are not in their standard format, hence we use the mdy() function from the lubridate library to convert the date to the mm-dd-yyyy format.



The flight date seems to have no correlation to the satisfaction ratings and is not a good predictor for the dependent variable.

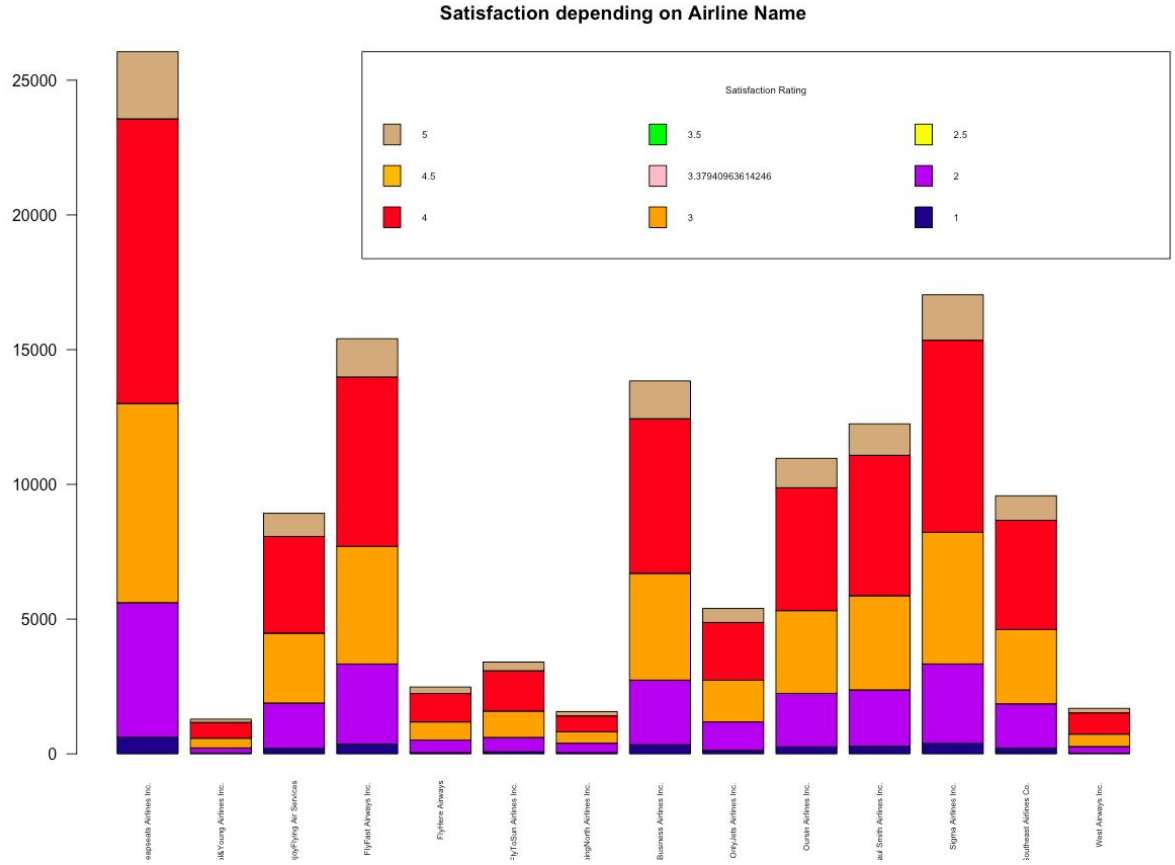
15. Airline Code



Seems a few people flew with the airline code; AS, F9, FL, HA, and VX. While, a lot of people flew with the airline code WN and it seems that most of their ratings was at 4.

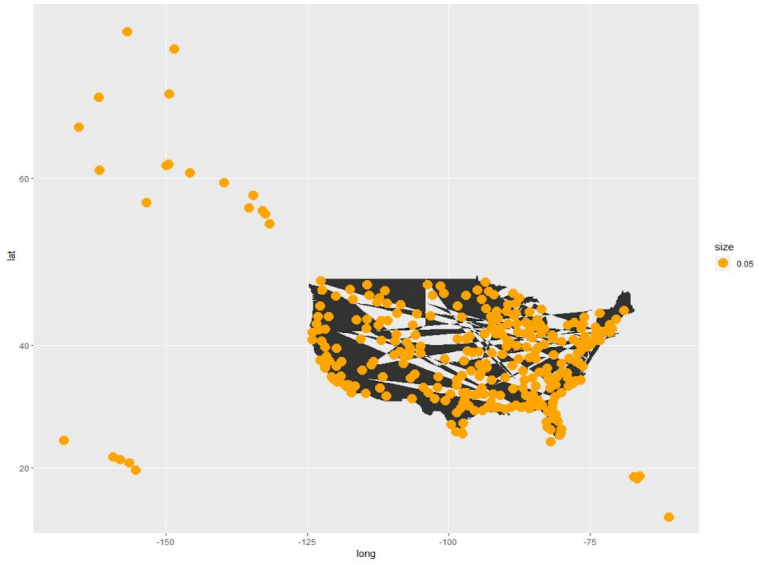
16. Airline Name

This plot illustrates that most of the customers flew with CheapSeats Airlines Inc. and it seems that they were mostly rated it with a 4. The second airline used the most was Sigma Airlines Inc. and that too had mostly ratings of 4.



17. Origin City

This variable refers to the city the customers depart from. There are 295 unique cities in the dataset. From the following map and table, we can see that the mean satisfaction rating for every origin city is more or less the same. Hence, we cannot use this variable as a good predictor to improve the overall customer satisfaction rating of the airline companies.



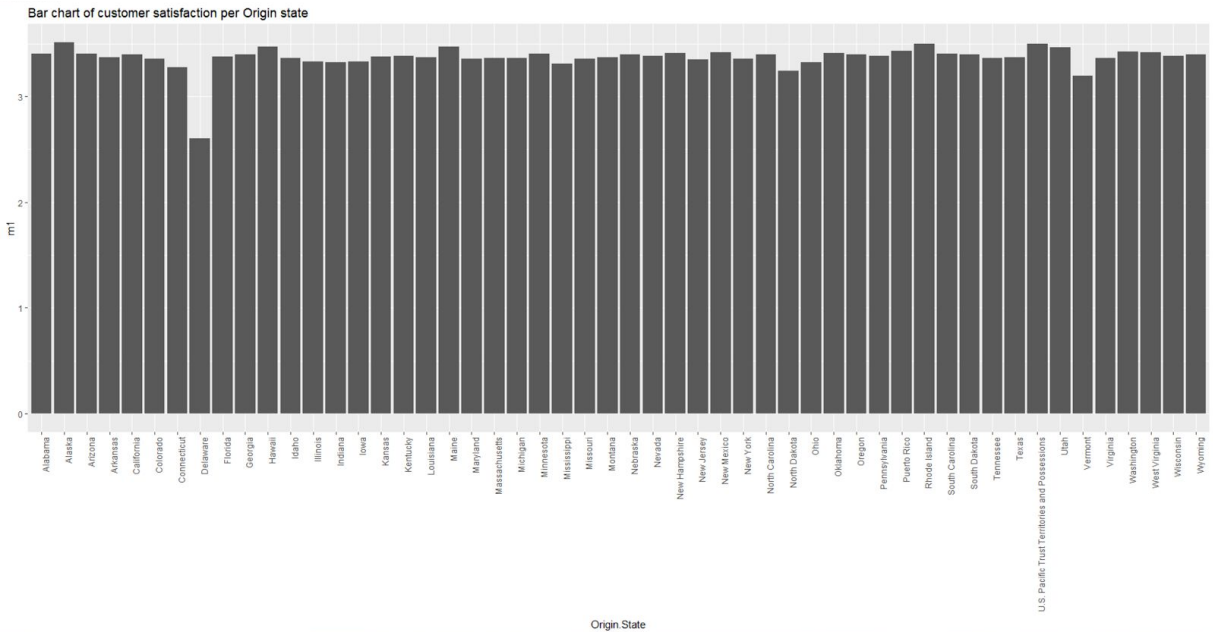
18. Origin State

Refers to the actual US state and its Pacific territories the customers have departed from. There are 52 unique values of Origin State in the dataset.

	Origin State	Mean
1	Alabama	3.406
2	Alaska	3.509
3	Arizona	3.401
4	Arkansas	3.371
5	California	3.395
6	Colorado	3.359
7	Connecticut	3.274
8	Delaware	2.600
9	Florida	3.375
10	Georgia	3.398
11	Hawaii	3.471
12	Idaho	3.364
13	Illinois	3.332

14	Indiana	3.325
15	Iowa	3.331
16	Kansas	3.378
17	Kentucky	3.381
18	Louisiana	3.371
19	Maine	3.474
20	Maryland	3.356
21	Massachusetts	3.365
22	Michigan	3.363
23	Minnesota	3.404
24	Mississippi	3.309
25	Missouri	3.355
26	Montana	3.372
27	Nebraska	3.398
28	Nevada	3.383
29	New Hampshire	3.407
30	New Jersey	3.346
31	New Mexico	3.419
32	New York	3.354
33	North Carolina	3.396
34	North Dakota	3.241
35	Ohio	3.325
36	Oklahoma	3.410
37	Oregon	3.399
38	Pennsylvania	3.380
39	Puerto Rico	3.430
40	Rhode Island	3.496
41	South Carolina	3.403
42	South Dakota	3.398

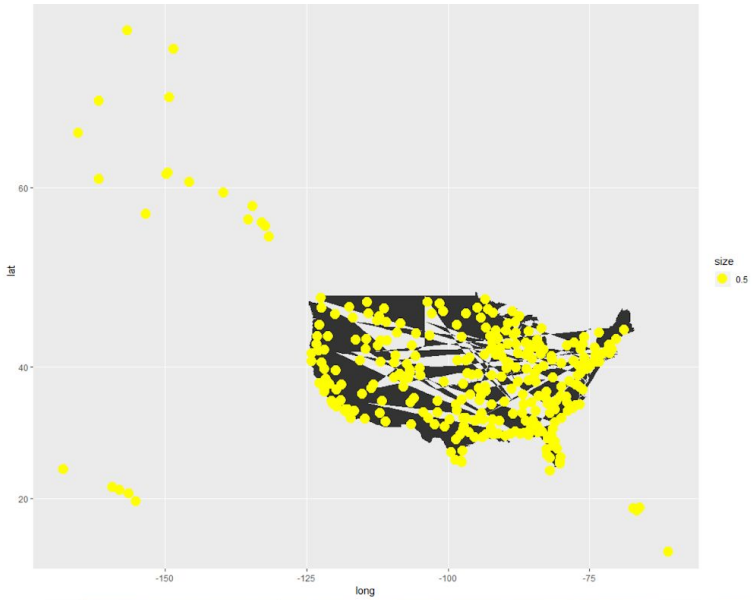
43	Tennessee	3.362
44	Texas	3.372
45	US Pacific Territories	3.500
46	Utah	3.461
47	Vermont	3.193
48	Virginia	3.362
49	Washington	3.420
50	West Virginia	3.417
51	Wisconsin	3.382
52	Wyoming	3.396



From the following plot, we can infer that the mean satisfaction rating is fairly constant for every origin state except Delaware. Still, this inference will not be enough to include this variable as a very effective predictor to increase the overall customer satisfaction rating for the airline companies.

19. Destination City

Refers to the actual cities the customers have travelled to. There are 296 unique cities in the satisfaction survey dataset.



From the following map, we infer that the mean satisfaction rating for every destination city is probably constant. We cannot use this variable as a good predictor to improve the overall customer satisfaction rating of the airline companies.

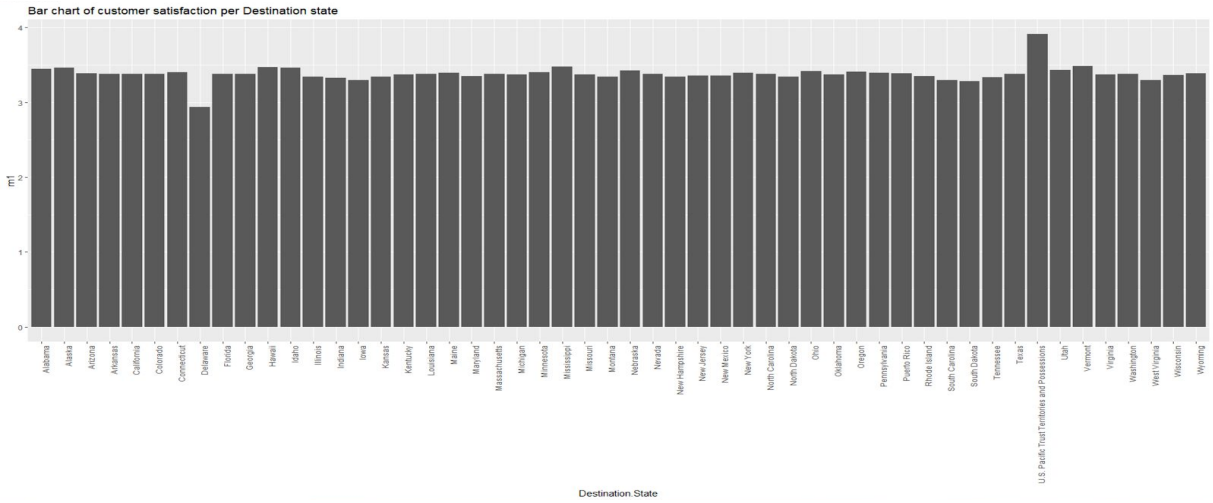
20. Destination State

Refers to the actual US states and its Pacific territories the customers have travelled to. There are 52 unique values in the dataset.

	Destination State	Mean
1	Alabama	3.442
2	Alaska	3.463
3	Arizona	3.384
4	Arkansas	3.375
5	California	3.380
6	Colorado	3.375
7	Connecticut	3.404
8	Delaware	2.937
9	Florida	3.381
10	Georgia	3.378

11	Hawaii	3.467
12	Idaho	3.458
13	Illinois	3.338
14	Indiana	3.327
15	Iowa	3.297
16	Kansas	3.343
17	Kentucky	3.370
18	Louisiana	3.375
19	Maine	3.396
20	Maryland	3.350
21	Massachusetts	3.380
22	Michigan	3.373
23	Minnesota	3.399
24	Mississippi	3.476
25	Missouri	3.373
26	Montana	3.338
27	Nebraska	3.424
28	Nevada	3.381
29	New Hampshire	3.340
30	New Jersey	3.356
31	New Mexico	3.355
32	New York	3.391
33	North Carolina	3.377
34	North Dakota	3.340
35	Ohio	3.413
36	Oklahoma	3.368
37	Oregon	3.406
38	Pennsylvania	3.390
39	Puerto Rico	3.386

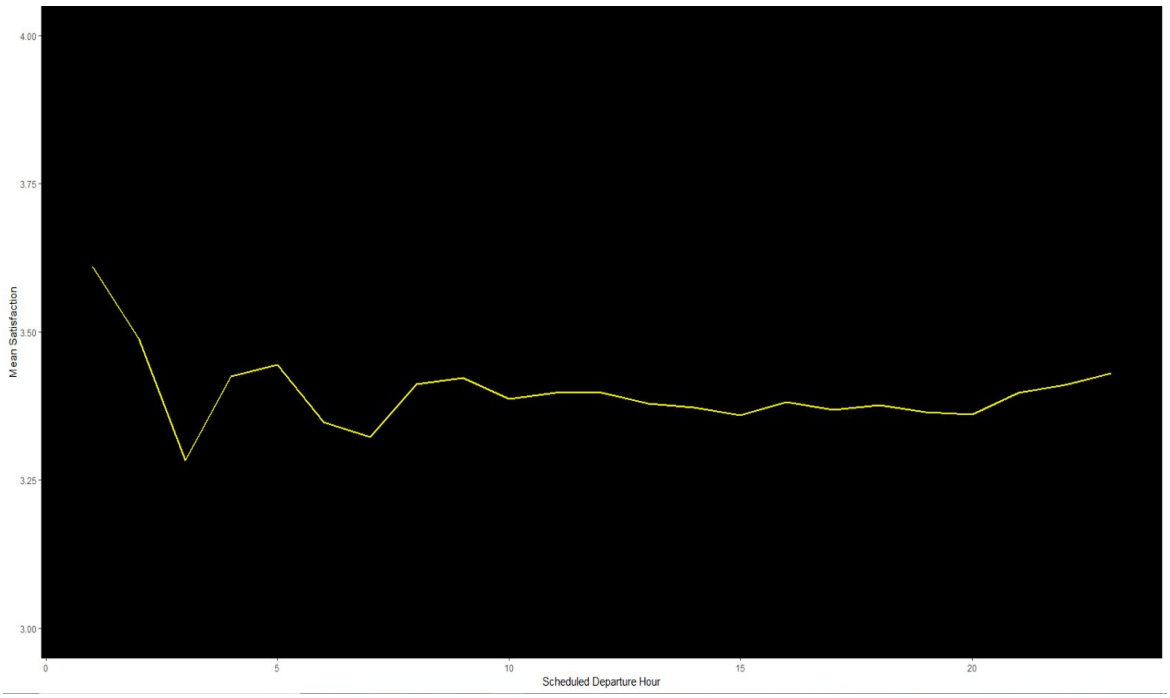
40	Rhode Island	3.351
41	South Carolina	3.295
42	South Dakota	3.283
43	Tennessee	3.336
44	Texas	3.382
45	US Pacific Territories	3.909
46	Utah	3.430
47	Vermont	3.480
48	Virginia	3.371
49	Washington	3.381
50	West Virginia	3.294
51	Wisconsin	3.367
52	Wyoming	3.384



From the following plot, we can infer that the mean satisfaction rating is fairly constant for every destination state. Hence, this variable will not be a very effective predictor to increase the overall customer satisfaction rating for the airline companies.

21. Scheduled Departure Hour

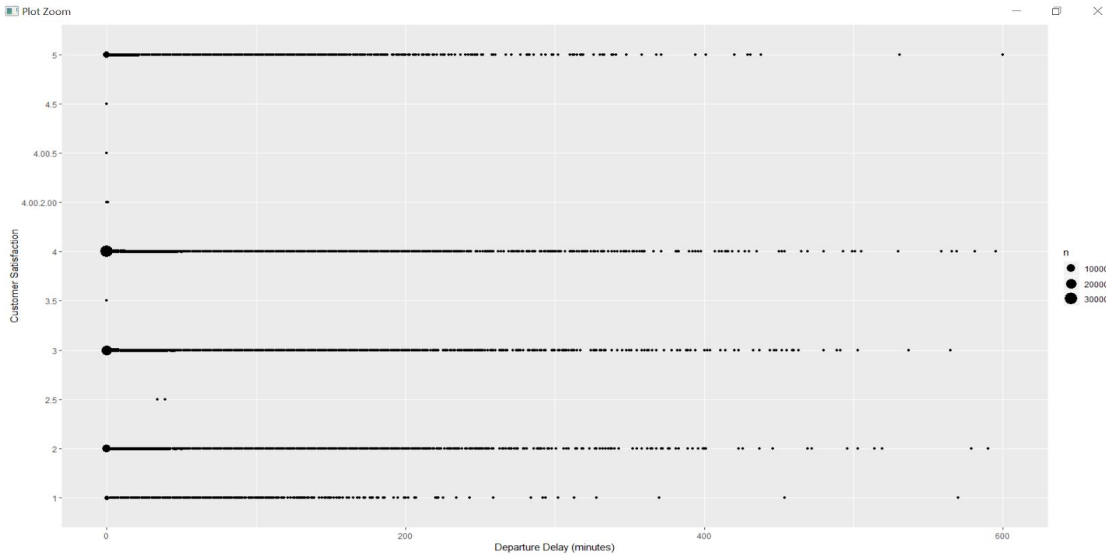
Refers to the specific time at which the customers are scheduled to depart. The departure hour is scheduled to start at 1AM and end at 23PM.



From the following plot, we can see that customers provide the highest satisfaction rate when the scheduled departure hour is between 1 and 2 AM which averages 3.6. While the lowest rating is provided when the scheduled departure hour is approximately between 3 and 4 AM which averages 3.3. After 5 AM the rating is roughly constant throughout the other departure hours.

22. Departure Delay in Minutes

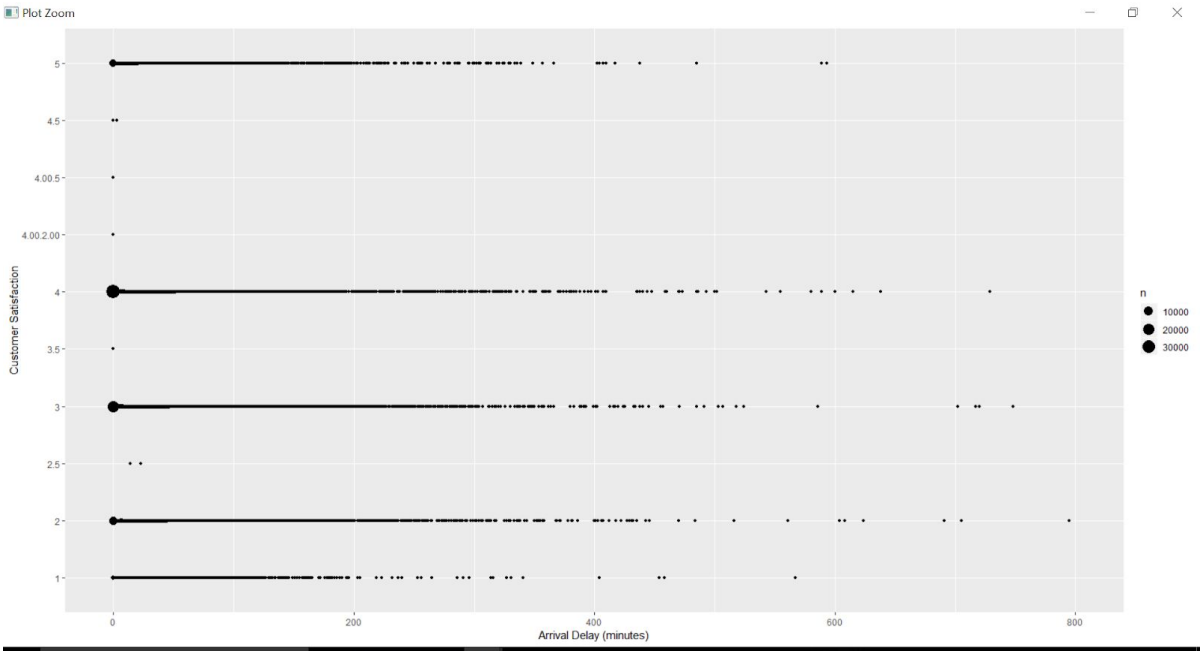
Over 90 % of the data is concentrated between 0 and 550 minutes**.**



Clearly the flights with a delay of zero minutes have a satisfaction rate majorly in the 3 to 5 batch and if the delay is of more than 200 minutes major data is spread across the 1 to 4 batch.

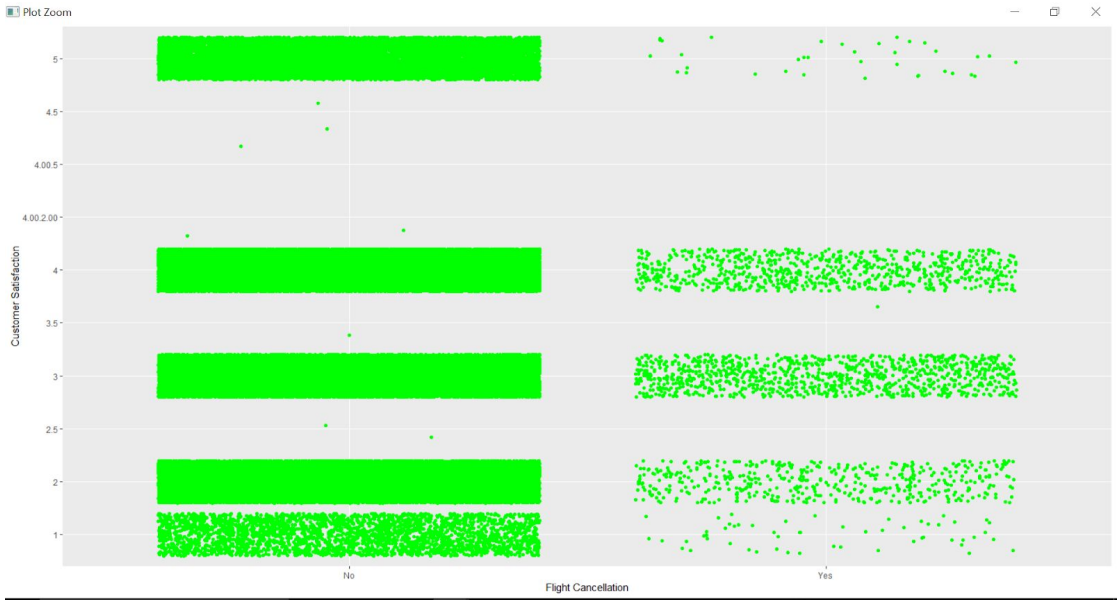
23. Arrival Delay in Minutes

Over 95 % of the data is concentrated between 0 and 800 minutes**.**



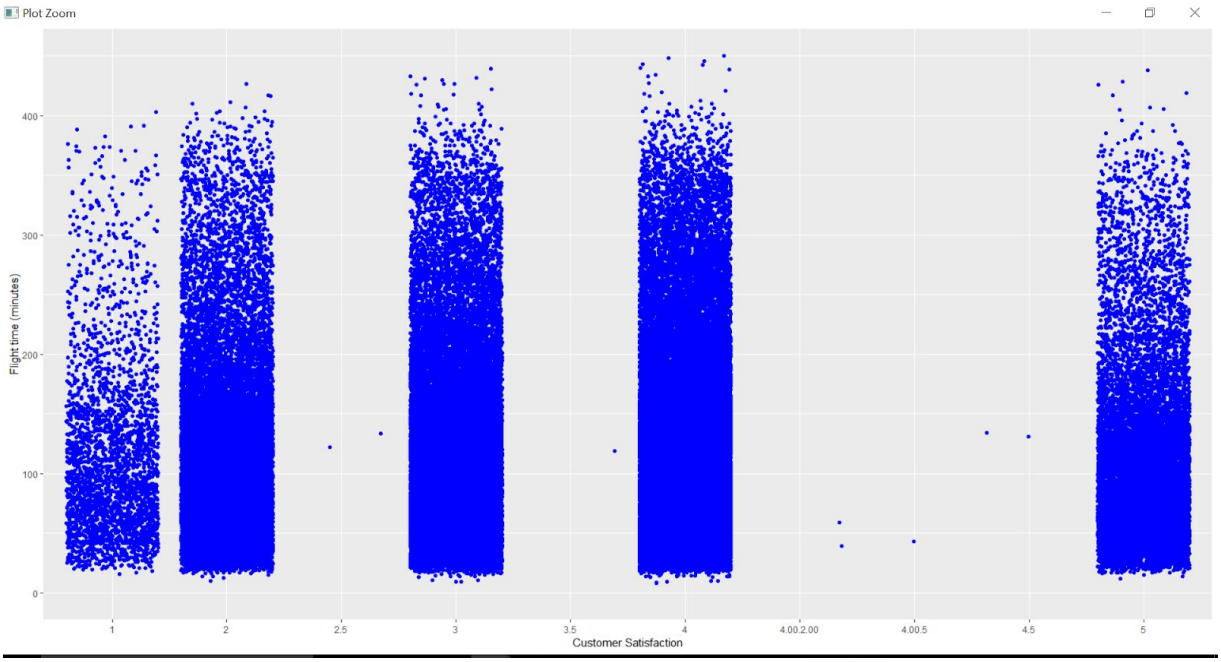
When the arrival delay is 0 minutes, around 65,000 observations suggest that customers give a high satisfaction rate to the airlines. So improving the arrival delay would also bring in a high satisfaction rate. Also, when delay goes beyond the mark of 400 minutes even if there is less amount of data we can infer that a major part lies in the batch of 1 to 2 while there is even less data with the satisfaction rate of 5.

24. Flight Cancelled



As the number of overall flight cancellations are very less, if the flight is cancelled the satisfaction rate is majorly concentrated in the 1 to 3 batch while if the flight is not cancelled 70% of the data is spread across 3 to 5 batch of the satisfaction rate. So if the number of cancellations reduce for a particular airline it would bring in a higher satisfaction rate as the data would start moving from the left of the graph to the right.

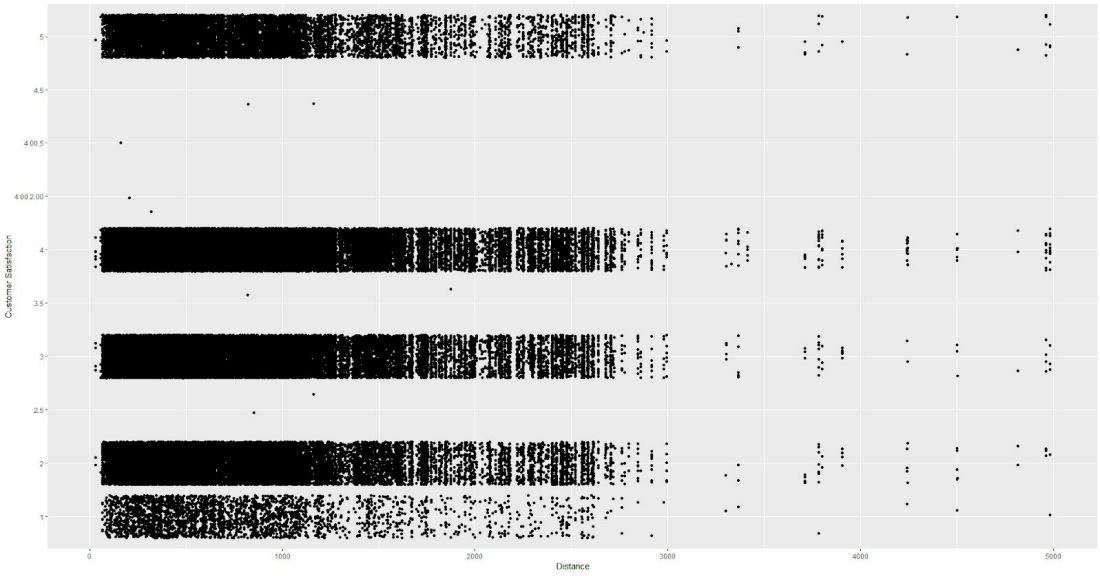
25. Flight Time in Minutes



This plot depicts the customer satisfaction as the dependant variable and flight time (minutes) as the independent variable on how flight time affects satisfaction rate. When the flight time is in between 50 minutes and 400 minutes major data lies within the 3 to 5 batch which is good but some part of the data shows low satisfaction rate. This can depend on various other factors like the quality of service during the flight time, the food etc.

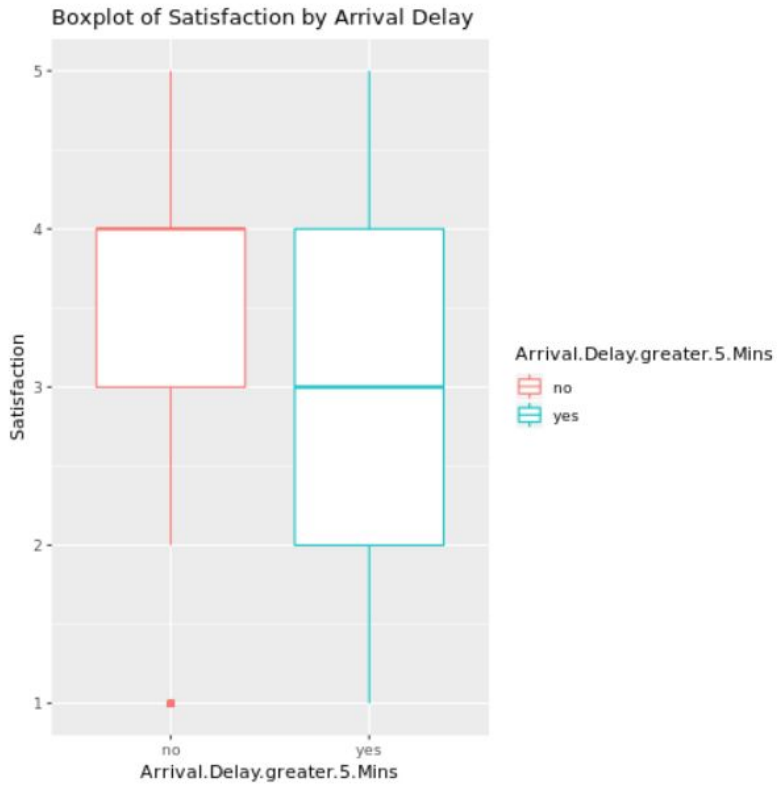
26. Flight Distance

The plot below illustrates how most of the customers did not fly past a distance of 3000. Shorter distances seemed to have a thicker density which means ratings from 2, 3, and 4 is where most of the customers rated the flights.



27. Arrival Delay Greater 5 Minutes

The boxplot can show that most of the flights have had a arrival delay greater than 5 minutes.



Code for data description and visualization:

The following packages were installed:

```
#install.packages("ggplot2")
library(ggplot2)
#install.packages("ggmap")
library(ggmap)
#install.packages("lubridate")
library(lubridate)
#install.packages("methods")
#install.packages("dplyr")
library(dplyr)
#install.packages("jtools")
library(jtools)
#install.packages('pscl')
library(pscl)
#install.packages("broom")
library(broom)
#install.packages("ggstance")
library(ggstance)
#install.packages("effects")
library(effects)
#install.packages("lattice")
```

```
#install.packages("caret")
library(caret)
#install.packages("e1071")
library(e1071)
```

1. Airline Status:

```
#Plotting Airline Status against Satisfaction
plot2<-ggplot(data, aes(Airline.Status, Satisfaction))+geom_boxplot()+labs(x="Airline Status", y="Satisfaction")
plot2
```

2. Age:

```
#Plot of Age vs Avg. Satisfaction
#Mean satisfaction for Age
satmean <- data %>%
  group_by(Age) %>%
  summarize(m1 = mean(Satisfaction))

satmean<-as.data.frame(satmean)

#Plotting average Satisfaction Score against Age
agesat<-ggplot(satmean,aes(Age,m1)) + geom_bar(stat="identity") + theme(axis.text.x = element_text(angle = 75, hjust =
1))+labs(y="Average Satisfaction Score")
agesat
```

3. Gender:

```
#Plotting Gender against Satisfaction
ggplot(data, aes(Gender, Satisfaction, fill=Gender))+geom_boxplot()+labs(x="Gender", y="Satisfaction")
```

4. Price Sensitivity

```
#Plotting Price Sensitivity against Satisfaction
ggplot(unclean_data, aes(Price.Sensitivity, Satisfaction))+geom_count()+labs(x="Price Sensitivity", y="Satisfaction")
```

5. Year of First Flight

```
#Year of First Flight
firstflight <- unclean_data %>%
  group_by(Year.of.First.Flight) %>%
  summarize(m3 = mean(Satisfaction))

firstflight<-as.data.frame(firstflight)

plot6<-ggplot(firstflight, aes(Year.of.First.Flight,m3))+geom_line()+scale_x_continuous(breaks =
c(2001,2002,2004,2006,2008,2010,2012,2014))+labs(x="Year of First Flight", y="Mean Satisfaction")
plot6
```

6. Number of Flights

```
#Mean satisfaction for Number of Flights
```

```
flightsatmean <- data %>%
  group_by(No.of.Flights.p.a.) %>%
  summarize(m1 = mean(Satisfaction))

flightsatmean<-as.data.frame(flightsatmean)

#Plotting average Satisfaction Score against No of Flights P.A
flightsat<-ggplot(flightsatmean,aes(No.of.Flights.p.a.,m1)) + geom_bar(stat="identity") + theme(axis.text.x = element_text(angle
= 75, hjust = 1))+labs(y="Average Satisfaction Score")
```

7. Flight with other airlines

```
 # make a stacked bar plot of independent variable : X..of.Flight.with.other.Airlines and dependent variable : Satisfaction
perflightsat <- ggplot(data, aes(x = X..of.Flight.with.other.Airlines, fill = factor(Satisfaction)))
geom_bar() + xlab("X..of.Flight.with.other.Airlines") + ylab("Total Count") + labs(fill = "Satisfaction")
Perflightsat
```

8. Type of Travel

```
# make a box plot of independent variable :Type.of.Travel and dependent variable : Satisfaction
typetravelsat <- ggplot(data, aes(x=Type.of.Travel, y=Satisfaction, colour= "red")) + geom_boxplot(fill="grey")
```

9. Number of Loyalty Cards

```
# make a stacked bar plot of independent variable :No..of.other.Loyalty.Cards  and dependent variable : Satisfaction
## No. Of other Loyalty Cards – it is kind of membership card of each customer, that for retail establishment to gain a benefits
such as, discounts.
loyalcardsat1 <- ggplot(data, aes(x = No..of.other.Loyalty.Cards, fill = factor(Satisfaction))) +
geom_bar() +xlab("No..of.other.Loyalty.Cards") + ylab("Total Count") +  labs(fill = "Satisfaction")
Loyalcardsat1
```

10. Shopping Amount at Airport

```
# make a scatter plot of independent variable :Shopping.Amount.at.Airport and dependent variable : Satisfaction
shoppingamtsat <- ggplot(data, aes(x=Shopping.Amount.at.Airport , y=Satisfaction, colour= "red")) +  geom_point()
```

11. Eating and Drinking at Airport

```
# make a bubble scatter plot of independent variable :Eating.and.Drinking.at.Airport and dependent variable : Satisfaction
eatdrinksat1<-ggplot(data, aes(x=data$Eating.and.Drinking.at.Airport,y=data$Satisfaction))+geom_count()
eatdrinksat1<-eatdrinksat1+labs(x="Eating.and.Drinking.at.Airport",y="Customer Satisfaction")
Eatdrinksat1
```

12. Class

```
class_plot<-ggplot(satisfactionSurvey,aes(x=satisfactionSurvey$Class,y=satisfactionSurvey$Satisfaction))+geom_jitter(color="
purple")
class_plot<-class_plot+labs(x="Class",y="Customer Satisfaction")
class_plot
```

```
class_satis <- ggplot(data, aes(x = Class, fill = factor(Satisfaction))) +
  geom_bar() +
  xlab("Class") +
  ylab("Total Count") +
  labs(fill = "Satisfaction")
class_satis
```

13. Day of Month

```
install.packages("ggthemes")
install.packages("extrafont")
install.packages("plyr")
install.packages("scales")

library("ggthemes")
library("extrafont")
library("plyr")
library("scales")

summary(data$Day.of.Month)
Min.    1st Qu.    Median    Mean     3rd Qu.      Max.
1.00      8.00       16.00      15.72      23.00       31.00
plot14 <- ggplot(data=data, aes(x=Day.of.Month, y=Satisfaction)) + geom_bar(stat='identity')
plot14

#make buckets for day of month and satisfaction
monthBucket <- replicate(length(data$Day.of.Month), "Middle")
#if values are above 21, will return end of month
monthBucket[data$Day.of.Month > 21] <- "End"
#if values are below 10, will return beginning of month
monthBucket[data$Day.of.Month < 10] <- "Beginning"

table(monthBucket)
#most of the people travel during the middle of the month
   Beginning    End    Middle
    37566      39409     52914

monthBucket <- sapply(monthBucket, as.factor)

summary(data$Satisfaction)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   4.000   3.379   4.000   5.000
#based on median created buckets for satisfaction (average, high, and low satisfaction)
satisfactionBucket <- replicate(length(data$Satisfaction), "Average")
satisfactionBucket[data$Satisfaction > 4] <- "High"
satisfactionBucket[data$Satisfaction < 4] <- "Low"

table(satisfactionBucket)
   Average    High     Low
    53758   12554   63577
#there were mostly low satisfaction rates
```

satisfactionBucket<- sapply(satisfactionBucket,as.factor)

table(satisfactionBucket, monthBucket)

satisfactionBucket	monthBucket Middle	End	Beginning
Average	22204	16512	15042
High	5089	3804	3661
Low	25621	19093	18863

#52914 people travelled in middle of the month
#39409 people travelled at end of month
#37566 people travelled at beginning of month

#look at percentages as well
prop.table(table(satisfactionBucket, monthBucket))

satisfactionBucket	monthBucket Middle	End	Beginning
Average	0.17094596	0.12712393	0.11580657
High	0.03917961	0.02928654	0.02818560
Low	0.19725304	0.14699474	0.14522400

counts3 <- table(satisfactionBucket, monthBucket)
barplot(counts3, main="Satisfaction depending Day of Month", col=c("darkblue","red", "yellow"), legend=rownames(counts3), args.legend = list(x = "topright", ncol=2, inset=c(0, 0), cex=.4))

14. Flight Date:

```
#Plotting Satisfaction against the Flight Date
flightdate <- data %>%
  group_by(Flight.date) %>%
  summarize(m2 = mean(Satisfaction))

flightdate<-as.data.frame(flightdate)

#head(flightdate)
plot15<-ggplot(flightdate, aes(Flight.date,m2, group=1))+geom_line()+labs(x="From 1st Jan to 31st March 2014", y="Mean Satisfaction")
plot15
```

15. Airline Code

summary(data$Airline.Code)

AA	AS	B6	DL	EV	F9	FL	HA	MQ	OO
12248	3407	5395	17037	15407	1568	2481	1688	8927	13840

OU	US	VX	WN
10968	9577	1288	26058

counts <- table(data$Satisfaction, data$Airline.Code)
plot16 <- barplot(counts, main="Satisfaction depending on Airline Code", col=c("darkblue","purple", "yellow", "orange", "pink", "chartreuse", "brown1", "darkgoldenrod1", "burlywood3", "darkslategray2", "azure1", "bisque1", "blueviolet", "goldenrod4"), las=2, legend=rownames(counts), args.legend = list(x = "top", ncol=2, inset=c(0, 0), cex=.4))

16. Airline Name

```
summary(data$Airline.Name)
Cheapseats Airlines Inc.      Cool&Young Airlines Inc.     EnjoyFlying Air Services
                   26058                          1288                          8927
FlyFast Airways Inc.              FlyHere Airways           FlyToSun Airlines Inc.
                   15407                          2481                          3407
              GoingNorth Airlines Inc.   Northwest Business Airlines Inc.   OnlyJets Airlines Inc.
                    1568                         13840                          5395
Oursin Airlines Inc.              Paul Smith Airlines Inc.              Sigma Airlines Inc.
                   10968                         12248                         17037
Southeast Airlines Co.            West Airways Inc.
                    9577                          1688

counts2 <- table(data$Satisfaction, data$Airline.Name)
plot17 <- barplot(counts2, main="Satisfaction depending on Airline Name",
col=c("darkblue","", "yellow", "orange", "pink", "chartreuse", "brown1", "darkgoldenrod1", "burlywood3", "darkslategray2",
"azure1", "bisque1", "blueviolet", "goldenrod4"), las=2, cex.names = 0.5, legend=rownames(counts2), args.legend = list(x =
"top", ncol=2, inset=c(0, 0), cex=.4))
```

17. Origin City

```
#calculating mean and discarding NA values
mean(data$Satisfaction,na.rm=TRUE)
#closest to mean
data$Satisfaction[is.na(data$Satisfaction)] <- 3.5
fltmean1 <- data %>% group_by(Orgin.City) %>% summarize(mean1 = mean(Satisfaction))
fltmean1 <- as.data.frame(fltmean1)
fltmean1$place <- toString(paste(data$Orgin.City,",",data$Origin.State))
fltmean1 <- as.data.frame(fltmean1)
us <- map_data("state")

lat <- data.frame()
lon <- data.frame()
#getting the latitude and longitude of NYC from the datasciencetoolkit
for(i in 1:nrow(fltmean1)){

x <- geocode(as.character(fltmean1[i,1]),source="dsk")
lon[i,1] <- x[1]
lat[i,1] <- x[2]
}
head(lat)
head(lon)

latlon <- as.data.frame(cbind(lon,lat,fltmean1))
#plotting the map for mean of satisfaction in origin city
originCity <- ggplot(us, aes(x=long, y=lat)) + expand_limits(x = us$long, y = us$lat)
originCity <- originCity + geom_polygon() + coord_map()
originCity <- originCity + geom_point(data=latlon, aes(x=lon, y=lat, size=0.05), color="orange")
originCity
```

18. Origin State

```
#Calculating mean of satisfaction and grouping it by Origin state
osmean <- data %>%
 group_by(Origin.State) %>%
 summarize(m1 = mean(as.numeric(Satisfaction)))
#Plotting graph for satisfaction vs origin state
osmean<-as.data.frame(osmean)
osmean
originState <- ggplot(osmean, aes(x=Origin.State, y=m1)) + geom_col()
originState <- originState + ggtitle("Bar chart of customer satisfaction per Origin state")
originState <- originState + theme(axis.text.x = element_text(angle = 90, hjust = 1))
originState
```

19. Destination City

```
#calculating mean and discarding NA values
mean(data$Satisfaction,na.rm=TRUE)
data$Satisfaction[is.na(data$Satisfaction)] <- 3.5
#rounding off the satisfaction value closest to mean
fltmean1 <- data %>% group_by(Destination.City) %>% summarize(mean1 = mean(Satisfaction))
fltmean1 <- as.data.frame(fltmean1)
fltmean1$place <- toString(paste(data$Destination.City,",",unclean_data$Destination.State))
fltmean1 <- as.data.frame(fltmean1)
us <- map_data("states")

lat <- data.frame()
lon <- data.frame()
#getting the latitude and longitude of NYC from the datasciencetoolkit
for(i in 1:nrow(fltmean1)){

 x <- geocode(as.character(fltmean1[i,1]),source="dsk")
 lon[i,1] <- x[1]
 lat[i,1] <- x[2]
}
head(lat)
head(lon)

latlon <- as.data.frame(cbind(lon,lat,fltmean1))
#plotting the map for mean of satisfaction in destination city
DestCity <- ggplot(us, aes(x=long, y=lat)) + expand_limits(x = us$long, y = us$lat)
DestCity <- DestCity + geom_polygon() + coord_map()
DestCity <- DestCity + geom_point(data=latlon, aes(x=lon, y=lat, size=0.5), color="yellow")
DestCity
```

20. Destination State

```
#calculating mean of satisfaction and grouping by destination state
dsmean <- data %>%
 group_by(Destination.State) %>%
 summarize(m1 = mean(as.numeric(Satisfaction)))
#plotting graph of satisfaction vs destination state
```

```
dsmean<-as.data.frame(dsmean)
dsmean
destState <- ggplot(dsmean, aes(x=Destination.State, y=m1)) + geom_col()
destState <- destState + ggtitle("Bar chart of customer satisfaction per Destination state")
destState <- destState + theme(axis.text.x = element_text(angle = 90, hjust = 1))
destState
```

21. Scheduled Departure Hour

```
#calculating mean of scheduled departure hour and grouping by Satisfaction
schdepthr <- data %>%
 group_by(Scheduled.Departure.Hour) %>%
 summarize(m3 = mean(Satisfaction))
#Transforming the scheduled departure hour variable into a data frame
schdepthr <- as.data.frame(schdepthr)
#plotting graph for scheduled departure hour vs Satisfaction
plotsdh<-ggplot(schdepthr, aes(Scheduled.Departure.Hour,m3))+geom_line(color="yellow", size = 1, linetype =
"solid")+labs(x="Scheduled Departure Hour", y="Mean Satisfaction") + ylim(3.0,4.0)+ theme(
 panel.background = element_rect(fill = "black",
                                 colour = "black",
                                 size = 0.5, linetype = "solid"),
 panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                 colour = "black"),
 panel.grid.minor = element_line(size = 0.25, linetype = 'solid',
                                 colour = "black")
)
Plotsdh
```

22. Departure Delay in Minutes

```
departure_delay__plot<-ggplot(satisfactionSurvey,aes(x=satisfactionSurvey$Departure.Delay.in.Minutes,y=satisfactionSurvey$
Satisfaction))+geom_count()
departure_delay__plot<-departure_delay__plot+labs(x="Departure Delay (minutes)",y="Customer Satisfaction") +xlim(0,550)
departure_delay__plot
```

23. Arrival Delay in Minutes

```
arrival_delay__plot<-ggplot(satisfactionSurvey,aes(x=satisfactionSurvey$Arrival.Delay.in.Minutes,y=satisfactionSurvey$Satisf
action))+geom_count()
arrival_delay__plot<-arrival_delay__plot+labs(x="Arrival Delay (minutes)",y="Customer Satisfaction")+xlim(0,800)
Arrival_delay__plot
```

24. Flight Cancelled

```
flight_cancel_plot<-ggplot(satisfactionSurvey,aes(x=satisfactionSurvey$Flight.cancelled,y=satisfactionSurvey$Satisfaction))+g
eom_jitter(color="green")
flight_cancel_plot<-flight_cancel_plot+labs(x="Flight Cancellation",y="Customer Satisfaction")
flight_cancel_plot
```

25. Flight Time in Minutes

```
flight_time_plot<-ggplot(satisfactionSurvey,aes(y=satisfactionSurvey$Flight.time.in.minutes,x=satisfactionSurvey$Satisfaction))
+geom_jitter(color="blue")
flight_time_plot<-flight_time_plot+labs(y="Flight time (minutes)",x="Customer Satisfaction")+ylim(0,450)
flight_time_plot
```

26. Flight Distance

```
dist_plot<-ggplot(satisfactionSurvey,aes(x=satisfactionSurvey$Flight.Distance,y=satisfactionSurvey$Satisfaction))+geom_jitter(
)
dist_plot<-dist_plot+labs(x="Distance",y="Customer Satisfaction")
dist_plot
```

27. Arrival Delay Greater 5 Minutes

```
bp_satisByGreater5 <- ggplot(data,
aes(Arrival.Delay.greater.5.Mins,Satisfaction))+geom_boxplot(aes(col=Arrival.Delay.greater.5.Mins))+labs(title="Boxplot of
Satisfaction by Arrival Delay")
bp_satisByGreater5
```

Modelling Techniques

Linear Modelling

1. Model building rationale

After exploring the pictorial or descriptive relationships between variables of interest, we decided to run a bivariate correlation for all numerical independent variables in the dataset before building any models, particularly linear modeling. The reason why we decided to run bivariate correlations was because when two of the independent variables were highly correlated (r=.8 and above), this would lead to a problem called multicollinearity. This would violate the assumption of running a multiple linear regression. Differently stated, we would not simultaneously include two independent variables which are moderately or highly correlated in a multiple regression, simple because it violates the assumption and these certainly don't contribute to a good regression model.

From the correlation matrix, we could see that several pairs of variables (Figure 1)are having a high correlation. For example, No. of Flights. p. a. is highly correlated with arrival

delay in minutes (r=0.96); price sensitivity and arrival delay in minutes (r=0.91); flight distance and eating and drinking at airport (r=0.91). We could also tell which variables are moderately or highly correlated with satisfaction (e.g., flight time in minutes). The correlation result helped us making the decision of which variables to be included in the model. We can also see the correlation result from the following plot. (Figure 2)

P

	Satisfaction	Age	Price.Sensitivity	No.of.Flights.p.a.	X..of.Flight.with.other.Airlines	No..of.other.Loyalty.Cards
Satisfaction		0.0000	0.0000	0.0000	0.0000	0.0000
Age	0.0000		0.0000	0.0000	0.0000	0.0000
Price.Sensitivity	0.0000	0.0000		0.0000	0.0000	0.0000
No.of.Flights.p.a.	0.0000	0.0000	0.0000		0.0000	0.0000
X..of.Flight.with.other.Airlines	0.0000	0.0000	0.0000	0.0000		0.0000
No..of.other.Loyalty.Cards	0.0000	0.0000	0.0000	0.0000	0.0000	
Shopping.Amount.at.Airport	0.0000	0.0000	0.0009	0.0000	0.0000	0.0000
Eating.and.Drinking.at.Airport	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Scheduled.Departure.Hour	0.1617	0.7386	0.0626	0.6001	0.0530	0.8318
Departure.Delay.in.Minutes	0.0000	0.7622	0.6761	0.6508	0.5674	0.7728
Arrival.Delay.in.Minutes	0.0000	0.7917	0.9060	0.9592	0.9029	0.7642
Flight.time.in.minutes	0.6948	0.5327	0.2626	0.5999	0.7353	0.7791
Flight.Distance	0.1646	0.7815	0.3734	0.4889	0.5523	0.8350

	Shopping.Amount.at.Airport	Eating.and.Drinking.at.Airport	Scheduled.Departure.Hour	Departure.Delay.in.Minutes
Satisfaction	0.0000	0.0000	0.1617	0.0000
Age	0.0000	0.0000	0.7386	0.7622
Price.Sensitivity	0.0009	0.0000	0.0626	0.6761
No.of.Flights.p.a.	0.0000	0.0000	0.6001	0.6508
X..of.Flight.with.other.Airlines	0.0000	0.0000	0.0530	0.5674
No..of.other.Loyalty.Cards	0.0000	0.0000	0.8318	0.7728
Shopping.Amount.at.Airport		0.0000	0.5620	0.6641
Eating.and.Drinking.at.Airport	0.0000		0.5378	0.0400
Scheduled.Departure.Hour	0.5620	0.5378		0.0000
Departure.Delay.in.Minutes	0.6641	0.0400	0.0000	
Arrival.Delay.in.Minutes	0.8431	0.0173	0.0000	0.0000
Flight.time.in.minutes	0.2188	0.7872	0.0000	0.0578
Flight.Distance	0.4583	0.9146	0.0000	0.0290

	Arrival.Delay.in.Minutes	Flight.time.in.minutes	Flight.Distance
Satisfaction	0.0000	0.6948	0.1646
Age	0.7917	0.5327	0.7815
Price.Sensitivity	0.9060	0.2626	0.3734
No.of.Flights.p.a.	0.9592	0.5999	0.4889
X..of.Flight.with.other.Airlines	0.9029	0.7353	0.5523
No..of.other.Loyalty.Cards	0.7642	0.7791	0.8350
Shopping.Amount.at.Airport	0.8431	0.2188	0.4583
Eating.and.Drinking.at.Airport	0.0173	0.7872	0.9146
Scheduled.Departure.Hour	0.0000	0.0000	0.0000
Departure.Delay.in.Minutes	0.0000	0.0578	0.0290
Arrival.Delay.in.Minutes		0.0000	0.8501
Flight.time.in.minutes	0.0000		0.0000
Flight.Distance	0.8501	0.0000	

Figure 1. Correlation p-values

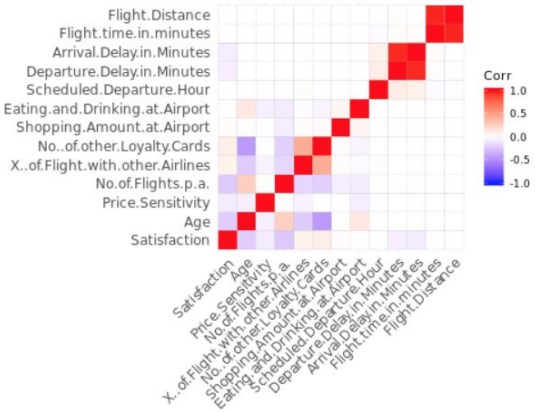


Figure 2. Correlation Matrix between Numerical Variables

Then T-tests or Anovas were conducted to see if the relationship between categorical variables and satisfaction are reaching the significant level in statistics or not. Before running t-tests or Anovas, boxplots were generated between each of the categorical variables and the outcome variable (satisfaction) to see if there is group differences. We found that there are significant differences in satisfaction between two groups by gender, flight cancelled, class, flight's arrival delay grater 5 mins, airline status and type of travel.

The following section includes the codes for correlation, t-tests and Anovas as well as some outputs

```
####### bivariate correlation for all numerical independent variables #########
#If two of the independent variables are highly related, this leads to a problem called multicollinearity
str(data)
nu_data <- select(data, Satisfaction, Age,"Price.Sensitivity","No.of.Flights.p.a.","X..of.Flight.with.other.Airlines",
"No..of.other.Loyalty.Cards", "Shopping.Amount.at.Airport",
"Eating.and.Drinking.at.Airport","Scheduled.Departure.Hour","Departure.Delay.in.Minutes", "Arrival.Delay.in.Minutes",
"Flight.time.in.minutes", "Flight.Distance")
str(nu_data)
#compute the correlation matrix
cor_1 <- round(cor(nu_data), 2)
cor_1
#the function cor() returns only the correlation coefficients between variables.
# use Hmisc R package to calculate the correlation p-values.
cor_2 <- rcorr(as.matrix(nu_data))
cor_2
cor_m<-cor(nu_data)
head(round(cor_m,2))
corPlot<- ggcorrplot(cor_m)

# significant difference in the mean of satisfaction by gender? --> yes
hist(data$Satisfaction)
bp_satisByGender <- ggplot(data, aes(Gender,Satisfaction))+geom_boxplot(aes(col=Gender))+labs(title="Boxplot of
Satisfaction by Gender")
bp_satisByGender
t.test(data$Satisfaction ~ data$Gender)
```

```
        Welch Two Sample t-test

data:  data$Satisfaction by data$Gender
t = -54.125, df = 109860, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.7175883 -0.6674337
sample estimates:
mean in group Female   mean in group Male
            4.673958            5.366469
```

```
# significant difference in the mean of satisfaction by flight cancelled? --> yes
hist(data$Satisfaction)
```

```
bp_satisByFCancelled <- ggplot(data,
aes(Flight.cancelled,Satisfaction))+geom_boxplot(aes(col=Flight.cancelled))+labs(title="Boxplot of Satisfaction by Flight Cancellation")
bp_satisByFCancelled
t.test(data$Satisfaction ~ data$Flight.cancelled)
```

```
        Welch Two Sample t-test

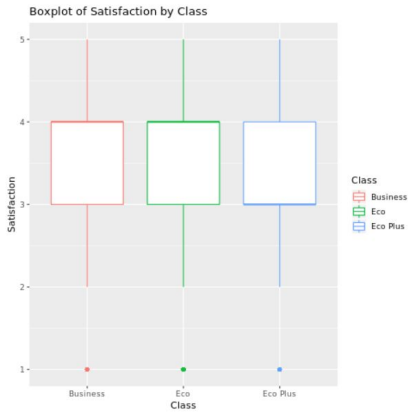
data:  data$Satisfaction by data$Flight.cancelled
t = 20.731, df = 2570.3, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.6499217 0.7857175
sample estimates:
 mean in group No mean in group Yes
         4.988540          4.270721
```

```
# significant difference in the mean of satisfaction by flight's arrival delay grater 5 mins ? --> yes
hist(data$Satisfaction)
bp_satisByGreater5 <- ggplot(data,
aes(Arrival.Delay.greater.5.Mins,Satisfaction))+geom_boxplot(aes(col=Arrival.Delay.greater.5.Mins))+labs(title="Boxplot of Satisfaction by Arrival Delay")
bp_satisByGreater5
t.test(data$Satisfaction ~ data$Arrival.Delay.greater.5.Mins)
```

```
        Welch Two Sample t-test

data:  data$Satisfaction by data$Arrival.Delay.greater.5.Mins
t = 44.445, df = 79777, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.5828900 0.6366714
sample estimates:
 mean in group no mean in group yes
         5.184201          4.574420
```

```
# significant difference in mean of satisfaction between class? --> yes
bp_satisByClass <- ggplot(data, aes(Class,Satisfaction))+geom_boxplot(aes(col=Class))+labs(title="Boxplot of Satisfaction by Class")
bp_satisByClass
#boxplot(data$Satisfaction~data$Class, data = data)
aov1<- aov(data$Satisfaction~data$Class)
summary(aov1)
TukeyHSD(aov1,conf.level=0.95)
```

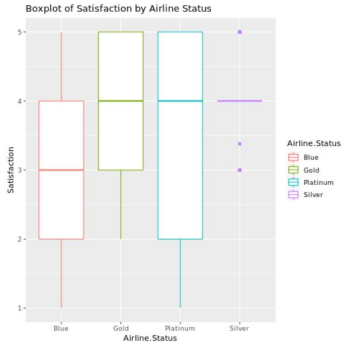



```
> aov1<- aov(data$Satisfaction~data$Class)
> summary(aov1)
                Df Sum Sq Mean Sq F value Pr(>F)
data$Class       2   1951   975.5   191.1 <2e-16 ***
Residuals   129886 662836     5.1
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TukeyHSD(aov1,conf.level=0.95)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = data$Satisfaction ~ data$Class)

$`data$Class`
                       diff        lwr        upr p adj
Eco-Business      -0.3853653 -0.4394268 -0.3313039     0
Eco Plus-Business -0.5583221 -0.6270081 -0.4896361     0
Eco Plus-Eco      -0.1729568 -0.2211786 -0.1247349     0
```

# significant difference in mean of satisfaction among customers with different airline status? --> Platinum-Gold not sig
bp_satisByStatus <- ggplot(data, aes(Airline.Stauts,Satisfaction))+geom_boxplot(aes(col=Airline.Status))+labs(title="Boxplot of Satisfaction by Airline Status")
bp_satisByStatus
aov2<- aov(data$Satisfaction~data$Airline.Status)
summary(aov2)
TukeyHSD(aov2,conf.level=0.95)

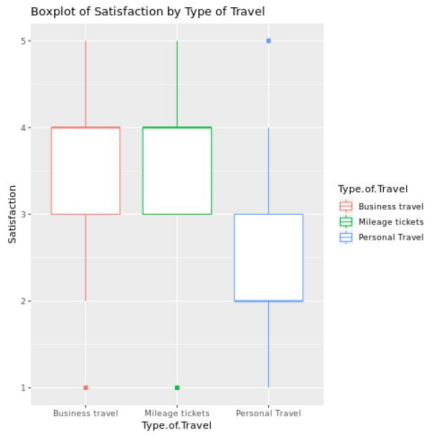



```
> aov2<- aov(data$Satisfaction~data$Airline.Status)
> summary(aov2)
                     Df Sum Sq Mean Sq F value Pr(>F)
data$Airline.Status   3  88066   29355    6611 <2e-16 ***
Residuals        129885 576721       4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TukeyHSD(aov2,conf.level=0.95)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = data$Satisfaction ~ data$Airline.Status)

$`data$Airline.Status`
                      diff       lwr       upr     p adj
Gold-Blue        1.6104252 1.55534532 1.6655050 0.0000000
Platinum-Blue    1.7368278 1.65107309 1.8225826 0.0000000
Silver-Blue      1.8384916 1.80030740 1.8766758 0.0000000
Platinum-Gold    0.1264026 0.02776972 0.2250356 0.0054876
Silver-Gold      0.2280664 0.16615832 0.2899745 0.0000000
Silver-Platinum  0.1016638 0.01137148 0.1919561 0.0199754
```

# significant difference in mean of satisfaction among customers with type of travel? --> yes
bp_satisByTravel <- ggplot(data, aes(Type.of.Travel,Satisfaction))+geom_boxplot(aes(col=Type.of.Travel))+labs(title="Boxplot of Satisfaction by Type of Travel")
bp_satisByTravel
aov3<- aov(data$Satisfaction~data$Type.of.Travel)
summary(aov3)
TukeyHSD(aov3,conf.level=0.95)



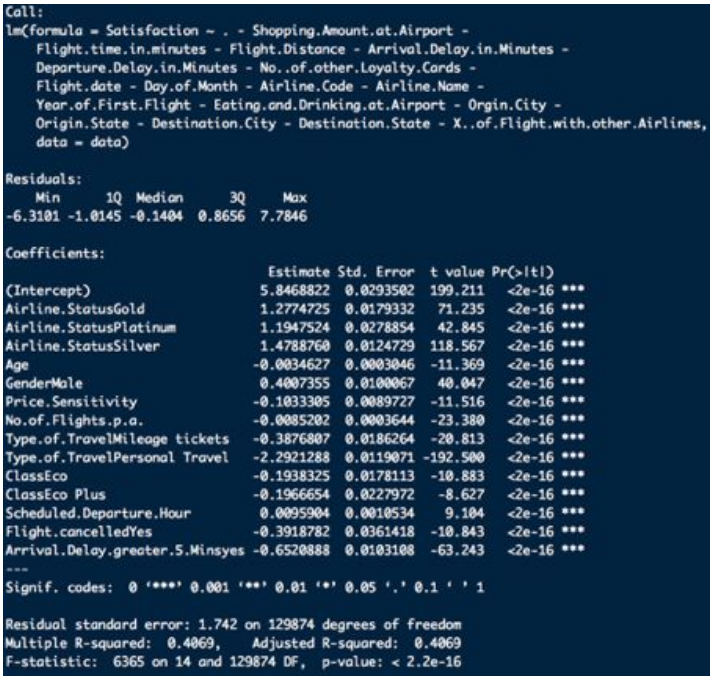

```
> aov3<- aov(data$Satisfaction~data$Type.of.Travel)
> summary(aov3)
                      Df Sum Sq Mean Sq F value Pr(>F)
data$Type.of.Travel    2 194076   97038   26776 <2e-16 ***
Residuals          129886 470710       4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TukeyHSD(aov3,conf.level=0.95)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = data$Satisfaction ~ data$Type.of.Travel)

$`data$Type.of.Travel`
                                        diff        lwr        upr p adj
Mileage tickets-Business travel -0.6483641 -0.6955531 -0.6011751     0
Personal Travel-Business travel -2.6914330 -2.7187334 -2.6641327     0
Personal Travel-Mileage tickets -2.0430689 -2.0927895 -1.9933483     0
```

2. Results

With the aforementioned findings, we run a linear model with backward deletion approach. The model was selected based on the adjust R-square and p values of each predictor. From Figure 3, we can see currently our regression model has 10 predictors which together explain 40.69 percent of the variance in the overall customer satisfaction. The 10 predictors are airline status, age, gender, price sensitivity, No of flights p.a., type of travel, class, scheduled departure hour, flight cancelled, and arrival delay greater 5 mins. We continued to use backward deletion approach to look for the best model, and we found the best three predictors were airline status, type of travel, and arrival delay greater 5 mins. These three predictors together can explain 39.46% of the variance in the overall customer satisfaction (see Figure 4), which was 1.2% less than the previous model with 10 predictors. Hence we ended up concluding that the final model with three predictors (i.e., airline status, type of travel, arrival delay greater than 5 mins) was the best model.



```
Call:
lm(formula = Satisfaction ~ . - Shopping.Amount.at.Airport -
    Flight.time.in.minutes - Flight.Distance - Arrival.Delay.in.Minutes -
    Departure.Delay.in.Minutes - No..of.other.Loyalty.Cards -
    Flight.date - Day.of.Month - Airline.Code - Airline.Name -
    Year.of.First.Flight - Eating.and.Drinking.at.Airport - Orgin.City -
    Origin.State - Destination.City - Destination.State - X..of.Flight.with.other.Airlines,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3101 -1.0145 -0.1404  0.8656  7.7846

Coefficients:
                                Estimate Std. Error  t value Pr(>|t|)
(Intercept)                    5.8468822  0.0293502  199.211   <2e-16 ***
Airline.StatusGold             1.2774725  0.0179332   71.235   <2e-16 ***
Airline.StatusPlatinum         1.1947524  0.0278854   42.845   <2e-16 ***
Airline.StatusSilver           1.4788760  0.0124729  118.567   <2e-16 ***
Age                           -0.0034627  0.0003046  -11.369   <2e-16 ***
GenderMale                     0.4007355  0.0100067   40.047   <2e-16 ***
Price.Sensitivity             -0.1033305  0.0089727  -11.516   <2e-16 ***
No.of.Flights.p.a.            -0.0085202  0.0003644  -23.380   <2e-16 ***
Type.of.TravelMileage tickets -0.3876807  0.0186264  -20.813   <2e-16 ***
Type.of.TravelPersonal Travel -2.2921288  0.0119071 -192.500   <2e-16 ***
ClassEco                      -0.1938325  0.0178113  -10.883   <2e-16 ***
ClassEco Plus                 -0.1966654  0.0227972   -8.627   <2e-16 ***
Scheduled.Departure.Hour       0.0095904  0.0010534    9.104   <2e-16 ***
Flight.cancelledYes           -0.3918782  0.0361418  -10.843   <2e-16 ***
Arrival.Delay.greater.5.Minsyes -0.6520888  0.0103108  -63.243   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.742 on 129874 degrees of freedom
Multiple R-squared:  0.4069,    Adjusted R-squared:  0.4069
F-statistic:  6365 on 14 and 129874 DF,  p-value: < 2.2e-16
```

Figure 3 lm model

```
Call:
lm(formula = Satisfaction ~ Airline.Status + Type.of.Travel +
    Arrival.Delay.greater.5.Mins, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.8284 -1.0550 -0.3522  0.9449  7.5721

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    5.543176   0.008203  675.79   <2e-16 ***
Airline.StatusGold             1.297103   0.017980   72.14   <2e-16 ***
Airline.StatusPlatinum         1.285198   0.027968   45.95   <2e-16 ***
Airline.StatusSilver           1.511788   0.012513  120.81   <2e-16 ***
Type.of.TravelMileage tickets -0.450490   0.018680  -24.12   <2e-16 ***
Type.of.TravelPersonal Travel -2.488116   0.010885 -228.57   <2e-16 ***
Arrival.Delay.greater.5.Minsyes -0.627200   0.010292  -60.94   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.76 on 129882 degrees of freedom
Multiple R-squared:  0.3947,    Adjusted R-squared:  0.3946
F-statistic: 1.411e+04 on 6 and 129882 DF,  p-value: < 2.2e-16
```

Figure 4. Final lm model

Below are the R codes for lm model

```
# model testing
model1 <- lm(Satisfaction ~ . -Flight.time.in.minutes -Departure.Delay.in.Minutes -X..of.Flight.with.other.Airlines, data = data)
summary(model1)

model2 <- lm(Satisfaction ~ . -Flight.time.in.minutes -Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards -Flight.date
-Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight, data = data)
```

```
summary(model2)

model3 <- lm(Satisfaction ~ . -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes
-No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight
-Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State, data = data)
summary(model3)

model4 <- lm(Satisfaction ~ . -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes
-No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight
-Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State -X..of.Flight.with.other.Airlines,
data = data)
summary(model4)
# ADJUSTED R-SQUARED 0.4069 with 11 predictors

model5 <- lm(Satisfaction ~ . -Shopping.Amount.at.Airport -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes
-Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name
-Year.of.First.Flight -Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State
-X..of.Flight.with.other.Airlines, data = data)
summary(model5)
# ADJUSTED R-SQUARED 0.4069 with 10 predictors

model6 <- lm(Satisfaction ~ . -Flight.cancelled -Shopping.Amount.at.Airport -Flight.time.in.minutes -Flight.Distance
-Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code
-Airline.Name -Year.of.First.Flight -Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State
-X..of.Flight.with.other.Airlines, data = data)
summary(model6)
# ADJUSTED R-SQUARED 0.4063 with 9 predictors

model7 <- lm(Satisfaction ~ . -Price.Sensitivity -Flight.cancelled -Shopping.Amount.at.Airport -Flight.time.in.minutes
-Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards -Flight.date -Day.of.Month
-Airline.Code -Airline.Name -Year.of.First.Flight -Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City
-Destination.State -X..of.Flight.with.other.Airlines, data = data)
summary(model7)
# ADJUSTED R-SQUARED 0.4057 with 8 predictors

model8 <- lm(Satisfaction ~ .-Arrival.Delay.greater.5.Mins -Price.Sensitivity -Flight.cancelled -Shopping.Amount.at.Airport
-Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards
-Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight -Eating.and.Drinking.at.Airport -Orgin.City
-Origin.State -Destination.City -Destination.State -X..of.Flight.with.other.Airlines, data = data)
summary(model8)
# ADJUSTED R-SQUARED 0.4057 --> 0.3879 with 7 predictors, so we decided to keep arrival delay greater 5 mins

model9 <- lm(Satisfaction ~ .-Scheduled.Departure.Hour -Price.Sensitivity -Flight.cancelled -Shopping.Amount.at.Airport
-Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards
-Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight -Eating.and.Drinking.at.Airport -Orgin.City
-Origin.State -Destination.City -Destination.State -X..of.Flight.with.other.Airlines, data = data)
summary(model9)
# ADJUSTED R-SQUARED 0.4057 -->0.4054 with 7 predictors
```

```
model10 <- lm(Satisfaction ~ .-No.of.Flights.p.a. -Scheduled.Departure.Hour -Price.Sensitivity -Flight.cancelled
-Shopping.Amount.at.Airport -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes
-No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight
-Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State -X..of.Flight.with.other.Airlines,
data = data)
summary(model10)
# ADJUSTED R-SQUARED 0.4054 --->0.4029 with 6 predictors

model11 <- lm(Satisfaction ~ .-Class -No.of.Flights.p.a. -Scheduled.Departure.Hour -Price.Sensitivity -Flight.cancelled
-Shopping.Amount.at.Airport -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes
-No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight
-Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State -X..of.Flight.with.other.Airlines,
data = data)
summary(model11)
# ADJUSTED R-SQUARED 0.4029 --> 0.4023 with 5 predictors

model12 <- lm(Satisfaction ~ .-Age -Class -No.of.Flights.p.a. -Scheduled.Departure.Hour -Price.Sensitivity -Flight.cancelled
-Shopping.Amount.at.Airport -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes -Departure.Delay.in.Minutes
-No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name -Year.of.First.Flight
-Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State -X..of.Flight.with.other.Airlines,
data = data)
summary(model12)
# ADJUSTED R-SQUARED 0.4023 --> 0.4013 with 4 predictors

model13 <- lm(Satisfaction ~ .-Gender -Age -Class -No.of.Flights.p.a. -Scheduled.Departure.Hour -Price.Sensitivity
-Flight.cancelled -Shopping.Amount.at.Airport -Flight.time.in.minutes -Flight.Distance -Arrival.Delay.in.Minutes
-Departure.Delay.in.Minutes -No..of.other.Loyalty.Cards -Flight.date -Day.of.Month -Airline.Code -Airline.Name
-Year.of.First.Flight -Eating.and.Drinking.at.Airport -Orgin.City -Origin.State -Destination.City -Destination.State
-X..of.Flight.with.other.Airlines, data = data)
summary(model13)
# ADJUSTED R-SQUARED 0.4013 -->39.46 with 3 predictors

# final model
m <- lm(formula = Satisfaction ~ Airline.Status + Type.of.Travel + Arrival.Delay.greater.5.Mins, data = data)
summary(m)
# ADJUSTED R-SQUARED for final model was 39.46
```

Logistic Regression

"Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables." -https://www.statisticssolutions.com/what-is-logistic-regression/

Data Manipulation:

Thus, to perform Logistic Regression we have to convert the dependent variable into a binary variable. So, we create a new binary variable based on the value of the 'Satisfaction' variable. If the satisfaction is greater than 3 then the new binary dependent variable 'Sat' is 1 which means the customer is satisfied with the service else 0 which mean the customer isn't satisfied.

Model Building:

Once we perform this conversion we remove the variables which might not make sense in consideration while building the model. Like origin city, origin state, destination city, destination state, airline code and airline name. Alongside, we also combine the variables departure delay and arrival delay into one variable called 'EffectiveDelay' by calculating the absolute value of the overall delay. We run the regression model with the remaining variable and get the following output.

```
Call:
glm(formula = Sat ~ ., family = binomial(link = "logit"), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6861  -0.5327   0.3177   0.8126   2.8705
```

```
Coefficients:
                                 Estimate Std. Error  z value Pr(>|z|)
(Intercept)                    -2.689e+01  6.686e+00   -4.022 5.78e-05 ***
Airline.StatusGold              9.059e-01  2.665e-02   33.990  < 2e-16 ***
Airline.StatusPlatinum          4.831e-01  3.895e-02   12.404  < 2e-16 ***
Airline.StatusSilver            1.719e+00  2.107e-02   81.610  < 2e-16 ***
Age                            -7.400e-03  5.141e-04  -14.395  < 2e-16 ***
GenderMale                      3.600e-01  1.477e-02   24.367  < 2e-16 ***
Price.Sensitivity              -1.582e-01  1.321e-02  -11.976  < 2e-16 ***
Year.of.First.Flight            1.193e-02  2.393e-03    4.985 6.21e-07 ***
No.of.Flights.p.a.             -1.285e-02  5.490e-04  -23.406  < 2e-16 ***
X..of.Flight.with.other.Airlines  3.195e-03  9.069e-04    3.523 0.000427 ***
Type.of.TravelMileage tickets  -4.197e-01  2.339e-02  -17.940  < 2e-16 ***
Type.of.TravelPersonal Travel  -3.066e+00  2.086e-02 -146.943  < 2e-16 ***
No..of.other.Loyalty.Cards     -6.457e-02  7.300e-03   -8.845  < 2e-16 ***
Shopping.Amount.at.Airport      6.872e-04  1.353e-04    5.080 3.77e-07 ***
Eating.and.Drinking.at.Airport -2.966e-04  1.403e-04   -2.113 0.034570 *
ClassEco                       -2.831e-01  2.644e-02  -10.707  < 2e-16 ***
ClassEco Plus                  -2.849e-01  3.363e-02   -8.472  < 2e-16 ***
Day.of.Month                   -4.500e-04  8.731e-04   -0.515 0.606290
Flight.date                     2.740e-04  2.896e-04    0.946 0.344120
Scheduled.Departure.Hour        1.865e-02  1.553e-03   12.013  < 2e-16 ***
Flight.cancelledYes            -6.952e-01  5.609e-02  -12.394  < 2e-16 ***
Flight.time.in.minutes          6.448e-04  4.103e-04    1.572 0.116020
Flight.Distance                -5.073e-05  4.935e-05   -1.028 0.304010
Arrival.Delay.greater.5.Minsyes -9.668e-01  1.744e-02  -55.423  < 2e-16 ***
EffectiveDelay                  1.846e-05  8.545e-04    0.022 0.982763
```

We can clearly see that their are some insignificant variable in the model. After removing those variables we split the dataset into training and testing set and re-run the regression model on the training dataset.

```
Coefficients:
                                Estimate Std. Error  z value Pr(>|z|)
(Intercept)                   -2.079e+01  5.868e+00   -3.544 0.000395 ***
Airline.StatusGold             8.960e-01  3.250e-02   27.568  < 2e-16 ***
Airline.StatusPlatinum         4.851e-01  4.763e-02   10.185  < 2e-16 ***
Airline.StatusSilver           1.697e+00  2.563e-02   66.203  < 2e-16 ***
Age                           -7.305e-03  6.238e-04  -11.711  < 2e-16 ***
GenderMale                     3.491e-01  1.793e-02   19.466  < 2e-16 ***
Price.Sensitivity             -1.482e-01  1.612e-02   -9.193  < 2e-16 ***
Year.of.First.Flight           1.110e-02  2.925e-03    3.797 0.000147 ***
No.of.Flights.p.a.            -1.364e-02  6.661e-04  -20.479  < 2e-16 ***
Type.of.TravelMileage tickets -4.091e-01  2.868e-02  -14.263  < 2e-16 ***
Type.of.TravelPersonal Travel -3.055e+00  2.542e-02 -120.174  < 2e-16 ***
No..of.other.Loyalty.Cards    -5.500e-02  8.389e-03   -6.556 5.53e-11 ***
Shopping.Amount.at.Airport     8.938e-04  1.658e-04    5.389 7.08e-08 ***
ClassEco                      -3.014e-01  3.239e-02   -9.306  < 2e-16 ***
ClassEco Plus                 -3.209e-01  4.102e-02   -7.823 5.17e-15 ***
Scheduled.Departure.Hour       1.987e-02  1.894e-03   10.491  < 2e-16 ***
Flight.cancelledYes           -7.287e-01  6.718e-02  -10.847  < 2e-16 ***
Arrival.Delay.greater.5.Minsyes -9.415e-01  1.853e-02  -50.804  < 2e-16 ***
```

As we can see all the variables currently in the model are significant with ‘three stars’. Now, we have to verify this model through K-fold cross-validation with k = 10.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 14744  3228
         1  6512 18813

               Accuracy : 0.775
                 95% CI : (0.7711, 0.779)
    No Information Rate : 0.5091
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5487
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6936
            Specificity : 0.8535
         Pos Pred Value : 0.8204
         Neg Pred Value : 0.7429
             Prevalence : 0.4909
         Detection Rate : 0.3405
   Detection Prevalence : 0.4151
      Balanced Accuracy : 0.7736
```

Importance of the variables:

```
> varImp(mod_fit)
glm variable importance

                                  Overall
`Type.of.TravelPersonal Travel`  100.000
Airline.StatusSilver              53.624
Arrival.Delay.greater.5.Minsyes   40.393
Airline.StatusGold                20.426
No.of.Flights.p.a.                14.335
GenderMale                        13.465
`Type.of.TravelMileage tickets`    8.994
Age                                6.800
Flight.cancelledYes                6.058
Scheduled.Departure.Hour           5.752
Airline.StatusPlatinum             5.489
ClassEco                           4.734
Price.Sensitivity                  4.637
`ClassEco Plus`                    3.459
No..of.other.Loyalty.Cards         2.371
Shopping.Amount.at.Airport         1.369
Year.of.First.Flight               0.000
```

As an experiment we go ahead and remove variables with very low importance and rerun the k-fold validation again.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 14387  3464
         1  6629 18817

               Accuracy : 0.7669
                 95% CI : (0.7629, 0.7709)
    No Information Rate : 0.5146
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5314
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6846
            Specificity : 0.8445
         Pos Pred Value : 0.8059
         Neg Pred Value : 0.7395
             Prevalence : 0.4854
         Detection Rate : 0.3323
   Detection Prevalence : 0.4123
      Balanced Accuracy : 0.7646

       'Positive' Class : 0
```

And the variable importance:

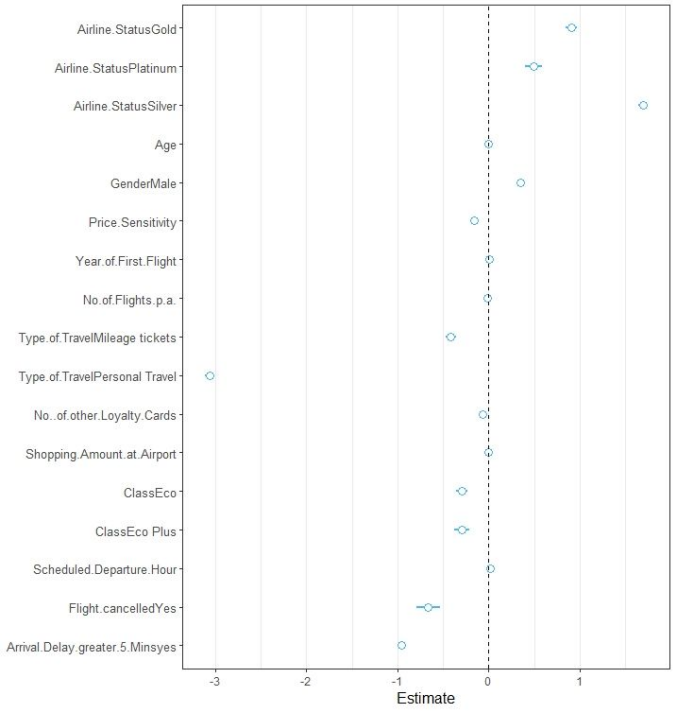
```
> varImp(mod_fit)
glm variable importance

                                  Overall
`Type.of.TravelPersonal Travel`  100.000
Airline.StatusSilver              47.481
Arrival.Delay.greater.5.Minsyes   33.121
Airline.StatusGold                13.789
No.of.Flights.p.a.                 6.229
`Type.of.TravelMileage tickets`    3.708
Airline.StatusPlatinum             0.000
```

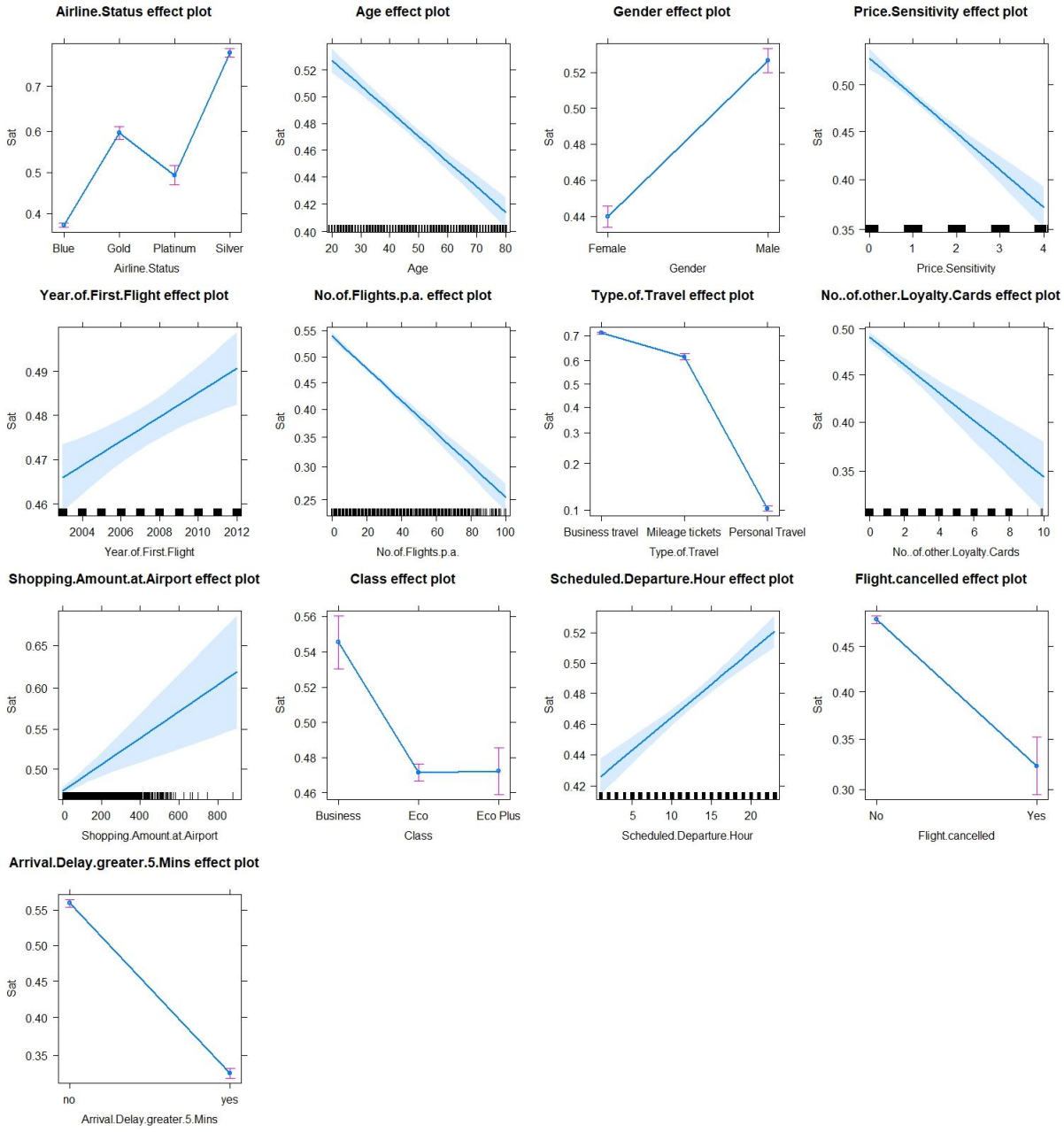
As you can see we managed to reduce the number of predictors tremendously without losing the accuracy of the model.

Results:

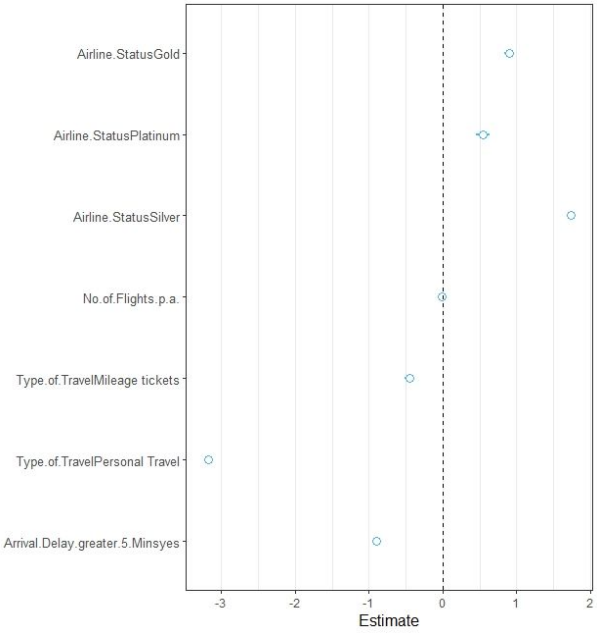
With the help of the 'jtool' library we can plot the effects of the logistic regression model.
Coefficient Plot:



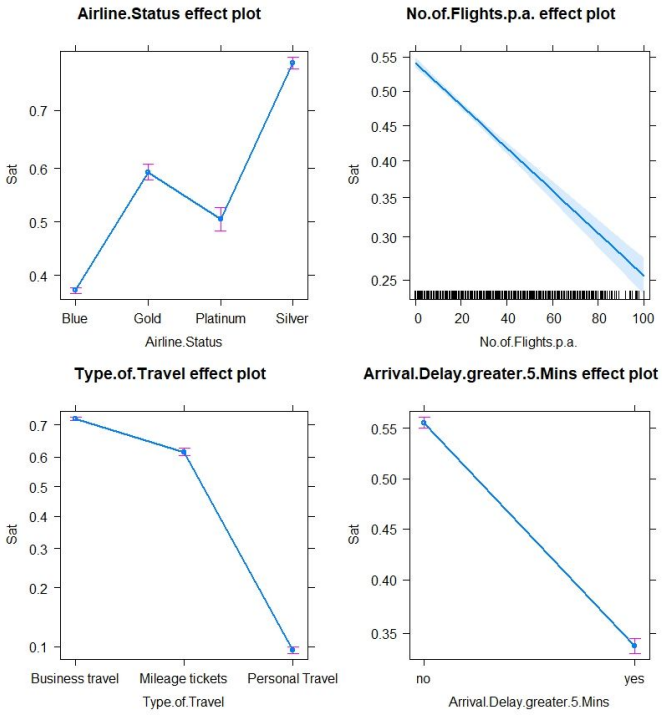
Regression prediction plot:



Results of the smaller model:



Regression prediction result:



Inference:

As we can see that both the larger and smaller model help predict the satisfaction score with 77% accuracy. The larger model can be used by the airline companies with good rating to boost their rating using the variables with smaller significance and the smaller model can be used by the companies with poor ratings to drive them up. Hence, to improve the satisfaction score companie are recommended to increase the quality of the flight travel for the passengers travelling in the economy and the economy plus category and also avoid any delays except under unavoidable circumstances like weather. Also, they'll have to make sure that the women in the age group of 30-50 get amenities and better care during their flight as they tend to give lower ratings. This will help the companies to achieve a great overall satisfaction score.

Code for Logistic Regression:

```
unclean_data<-data.frame(read.csv(file.choose()), stringsAsFactors=FALSE)
#Converting the dependent variable into numeric
unclean_data$Satisfaction<-as.numeric(as.character(unclean_data$Satisfaction))

#Number of missing values
sum(is.na(unclean_data$Departure.Delay.in.Minutes))
sum(is.na(unclean_data$Arrival.Delay.in.Minutes))
sum(is.na(unclean_data$Flight.time.in.minutes))
sum(is.na(unclean_data$Satisfaction))


#Summary of the three variables with missing variables
summary(unclean_data$Departure.Delay.in.Minutes)
summary(unclean_data$Arrival.Delay.in.Minutes)
summary(unclean_data$Flight.time.in.minutes)
summary(unclean_data$Satisfaction)

#Replacing the NA's
unclean_data$Departure.Delay.in.Minutes[is.na(unclean_data$Departure.Delay.in.Minutes)] <-
median(unclean_data$Departure.Delay.in.Minutes, na.rm=TRUE)
unclean_data$Arrival.Delay.in.Minutes[is.na(unclean_data$Arrival.Delay.in.Minutes)] <-
mean(unclean_data$Arrival.Delay.in.Minutes, na.rm=TRUE)
unclean_data$Flight.time.in.minutes[is.na(unclean_data$Flight.time.in.minutes)] <-
median(unclean_data$Flight.time.in.minutes, na.rm=TRUE)
unclean_data$Satisfaction[is.na(unclean_data$Satisfaction)] <- mean(unclean_data$Satisfaction, na.rm=TRUE)

#Summary of the three variables after removing the missing variables
summary(unclean_data$Departure.Delay.in.Minutes)
summary(unclean_data$Arrival.Delay.in.Minutes)
summary(unclean_data$Flight.time.in.minutes)
summary(unclean_data$Satisfaction)

#Dealing with the Flight Data variable
```

```
unclean_data$Flight.date<-mdy(unclean_data$Flight.date)
head(unclean_data$Flight.date)
#Calculating the Effective Delay
unclean_data$EffectiveDelay<-abs(unclean_data$Departure.Delay.in.Minutes-unclean_data$Arrival.Delay.in.Minutes)
unclean_data$Departure.Delay.in.Minutes<-NULL
unclean_data$Arrival.Delay.in.Minutes<-NULL
data<-unclean_data

#Plot of Age vs Avg. Satisfaction
#Mean satisfaction for Age
satmean <- data %>%
  group_by(Age) %>%
  summarize(m1 = mean(Satisfaction))

satmean<-as.data.frame(satmean)

#Plotting Satisfaction against the Flight Date
flightdate <- data %>%
  group_by(Flight.date) %>%
  summarize(m2 = mean(Satisfaction))

flightdate<-as.data.frame(flightdate)

#head(flightdate)
plot15<-ggplot(flightdate, aes(Flight.date,m2, group=1))+geom_line()+labs(x="From 1st Jan to 31st March 2014", y="Mean
Satisfaction")
plot15

#Plotting Airline Status against Satisfaction
plot2<-ggplot(data, aes(Airline.Status, Satisfaction))+geom_boxplot()+labs(x="Airline Status", y="Satisfaction")
plot2
#Plotting Gender against Satisfaction
ggplot(data, aes(Gender, Satisfaction, fill=Gender))+geom_boxplot()+labs(x="Gender", y="Satisfaction")

#Plotting Price Sensitivity against Satisfaction
ggplot(unclean_data, aes(Price.Sensitivity, Satisfaction))+geom_count()+labs(x="Price Sensitivity", y="Satisfaction")

#Year of First Flight
firstflight <- unclean_data %>%
  group_by(Year.of.First.Flight) %>%
  summarize(m3 = mean(Satisfaction))

firstflight<-as.data.frame(firstflight)

plot6<-ggplot(firstflight, aes(Year.of.First.Flight,m3))+geom_line()+scale_x_continuous(breaks =
c(2001,2002,2004,2006,2008,2010,2012,2014))+labs(x="Year of First Flight", y="Mean Satisfaction")
plot6

#Creating a Categoricall Variable from the dependent variable
data$Sat<-replicate(length(data$Satisfaction),0)
```

```
data$Sat[data$Satisfaction>3]<-1
data$Sat<-as.factor(data$Sat)
str(data)

#Removing the unwanted variables
data<-data[,-c(1)]
data$Airline.Code<-NULL
data$Airline.Name<-NULL
data$Orgin.City<-NULL
data$Origin.State<-NULL
data$Destination.City<-NULL
data$Destination.State<-NULL

#Initial model
model<-glm(Sat~., family=binomial(link="logit"),data=data)
summary(model)

#Removing insignificant Variables
data$Day.of.Month<-NULL
data$Flight.date<-NULL
data$Flight.time.in.minutes<-NULL
data$Flight.Distance<-NULL
data$EffectiveDelay<-NULL
data$Eating.and.Drinking.at.Airport<-NULL
data$X..of.Flight.with.other.Airlines<-NULL

#Train and Test data split
rand<-sample(1:dim(data)[1])
cutpoint2_3<-floor(2*dim(data)[1]/3)
cutpoint2_3

traindata<-data[rand[1:cutpoint2_3],]
testdata<-data[rand[(cutpoint2_3+1):dim(data)[1]],]

#Second model
model2<-glm(Sat~., family=binomial(link="logit"),data=traindata)
summary(model2)

#K-Fold Cross Validation
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)
mod_fit <- train(Sat ~.,  data=traindata, method="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)

#Predicting the outcome using the model and the test data
pred = predict(mod_fit, newdata=testdata)

#Confusion Matrix
confusionMatrix(data=pred, testdata$Sat)

#Model Summary
```

```
summary(mod_fit)

#Variable Importance
varImp(mod_fit)

#Plotting the model and its effects
plot_summs(model2)
plot(allEffects(model2))

#Creating a copy of the dataset for the smaller model
data2<-data

#Removing variable with low significance
data2$Year.of.First.Flight<-NULL
data2$No..of.other.Loyalty.Cards<-NULL
data2$Class<-NULL
data2$Price.Sensitivity<-NULL
data2$Scheduled.Departure.Hour<-NULL
data2$Age<-NULL
data2$Flight.cancelled<-NULL
data2$Gender<-NULL
data2$Shopping.Amount.at.Airport<-NULL

#Data split
rand<-sample(1:dim(data2)[1])
cutpoint2_3<-floor(2*dim(data2)[1]/3)
cutpoint2_3

traindata<-data2[rand[1:cutpoint2_3],]
testdata<-data2[rand[(cutpoint2_3+1):dim(data2)[1]],]

#Model Bulding
tiny<-glm(Sat~., family=binomial(link="logit"),data=traindata)

#K-Fold cross validation
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)
mod_fit <- train(Sat ~.,  data=traindata, method="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)

#Variable importance
varImp(mod_fit)

#Predicting the result
pred = predict(mod_fit, newdata=testdata)

#Confusion Matrix
confusionMatrix(data=pred, testdata$Sat)

#PLotting the result and its effects
plot_summs(tiny)
```

plot(allEffects(tiny))

Support Vector Machine

In this step, we used a supervised learning technique called a support vector machine (SVM). First, I train the algorithm on an initial set of data (the supervised phase) and then test out on a brand-new set of data.

SVM uses a technique called the kernel trick to transform my dataset and then based on these transformations it finds an optimal boundary between the possible outputs. So simply, it helps to separate our data based on the output have been defined. Based-on T-test result, the 7 most important variables are used in SVM analysis, 'Age, Gender, Price.Sensitivity, No.of.Flights.p.a, Type.of.Travel, Class, Arrival.Delay.greater.5.Mins' .

In the SVM prediction model the percentage of mistakes explains the error rate, and we tried to detect the model having the error rate to be as low as possible. After all analysis work, we found the best model with the smallest error rate to distinguish our customer's Overall satisfaction.

Analysis step & Finding

First, after the cleaning step of data, made 'OverallSatisfaction' column to define "Satisfied" and "Not Satisfied" customer. And then using dim function to detect the size of the whole data, which is

[dim(unclean_data) # [1] 129889 30]

But there was an issue running Rstudio because of the extremely huge dataset, so I changed the approach with random sampling to detect the error rate first of the sample data set. According to the sampling we used 'sample function()' like below

sample(1:nrow(unclean_data), size =, replace =FALSE)

dummyData<-unclean_data[Index,]

And then, we did the sampling approach with 13000 rows of data which is 10% of the whole data set '129889'. The basic step of having substantial number of training cases that algorithm can use to discover and mimic the underlying pattern is fulfilled, set the proper 'cross-validation' number for avoiding the problem of overfitting and 'C argument (so-called cost of constraints)' to have less classification mistakes. Then, simply start the algorithm work and also make the prediction model start from using one of the 7 most variables, and then move forward to input 2,3,4,5, and eventually all 7 variables to input and then compare the each model's error rate.

After finalizing all the single step, we compared the error rate and found the best model included all 7 variables in the model.

Finally, I used whole dataset into the model to eventually find out the best prediction model with the 7 important variables 'Age, Gender, Price.Sensitivity, No.of.Flights.p.a, Type.of.Travel, Class, Arrival.Delay.greater.5.Mins' which has training error of 0.215593 at its

SVM output. And moved on to prediction model, at this step most important part was calculating the percentage of confusion matrix which is Not Satisfied(49%) and Satisfied(51%) first, and then see the overall error rate 0.2368755. This model finally enable to predict almost with 80% accuracy to distinguish Satisfied and not satisfied customers.

Final SVM model output

```
> ksvm(overallSatisfaction ~., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3, prob.m
odel=TRUE)
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 5

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.206272679803563

Number of Support Vectors : 40013

Objective Function Value : -191525.6
Training error : 0.214593
Cross validation error : 0.217249
Probability model included.
```

```
> # Create a confusion matrix (a 2 x 2 table)
> comTable <- data.frame(newtestData[ ,8], svmPred[1, ])
> table(comTable)
                svmPred.1...
newtestData...8.     0     1
   Not Satisfied  7693 13473
   Satisfied     19568  2563
> # 11. Calculate an error rate based on what you see in the confusion matrix.
> t<-table(comTable)
> sum(t[1,1]+t[2,2])/sum(t)
[1] 0.2368755
```

SVM analysis Output Summary

1) Best model output (using whole dataset) :

The best model which is using whole dataset and

include all most important 7 variables;

'Age, Gender, Price.Sensitivity, No.of.Flights.p.a, Type.of.Travel,

Class, Arrival.Delay.greater.5.Mins'

Error rate : 0.2368755

All other model output (using 10% of whole dataset) :

with one important variables : Age and overallSatisfaction

error rate : 0. 0.3230272

with one important variables : Gender and overallSatisfaction

```
# error rate : 0.44024
# with one important variables : Price.Sensitivity and overallSatisfaction
# error rate : 0.443701
# with one important variables : No.of.Flights.p.a and overallSatisfaction
# error rate : 0.3936317
# with one important variables : Type.of.Travel and overallSatisfaction
# error rate : 0.2464236
# with one important variables : Class and overallSatisfaction
# error rate : 0.4755422
# with one important variables : Arrival.Delay.greater.5.Mins and overallSatisfaction
# error rate : 0.4257037
# with 3 important variables : Age,No.of.Flights.p.a,Type.of.Traveland overallSatisfaction
# error rate : 0.2316567
# with 4 important variables : Age,No.of.Flights.p.a, Type.of.Traveland overallSatisfaction,Arrival.Delay.greater.5.Mins
# error rate : 0.2185048
# with 5 important variables :
#Age,No.of.Flights.p.a,Gender,Type.of.Traveland overallSatisfaction
# Arrival.Delay.greater.5.Mins
# error rate : 0.2198892
# with 7 important variables :
#Age,No.of.Flights.p.a,Gender,Type.of.Traveland overallSatisfaction
# Arrival.Delay.greater.5.Mins
# error rate : 0.2113521
```

Lesson-learned

- SVM's are very good when we have no idea on the data.
- It has a regularisation parameter, which makes the user think about avoiding 'overfitting'.
- Choosing a "good" kernel function is not easy.
- Need to be patient while building SVMs on large datasets. They take a lot of time for training.

Code for Support Vector Machine

```
library(ggplot2)
library(lubridate)
library(dplyr)
library(jsonlite)
library(kernlab)
library(curl)
View(unclean_data)

#############################################################
# update new 2 columns
unclean_data$overallSatisfaction <-ifelse(unclean_data$Satisfaction>3,"Satisfied","Not Satisfied")
unclean_data$effectiveDelay <-abs(unclean_data$Departure.Delay.in.Minutes-unclean_data$Arrival.Delay.in.Minutes)
# show how many rows and columns
str(unclean_data)
dim(unclean_data) # [1] 129889     30
```

```
############################################################
# 2) Build a support vector model using the ksvm( ) function
# with one important variables : Age and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)
#show how many rows and columns
dim(dummyData) # 130000 rows, 30 columns
# can use table() in factor

randIndex <- sample(1:dim(dummyData)[1])
randIndex
# floor() chops off any decimal part of calculation
cutPoint2_3 <-floor(2*dim(dummyData)[1]/3)
trainData <- dummyData[1:cutPoint2_3,]
str(trainData)
summary(trainData)
testData <- dummyData[randIndex[(cutPoint2_3+1):dim(dummyData)[1]],]
testData
# Use the dim( ) function to demonstrate that the resulting training data set and test data set contain the appropriate number of
cases.
dim(trainData)
dim(testData)

# Build a support vector model using the ksvm( ) function
# Age, overallSatisfaction
newtrainData <-trainData[,c(3,29)]
newtestData<-testData[,c(3,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

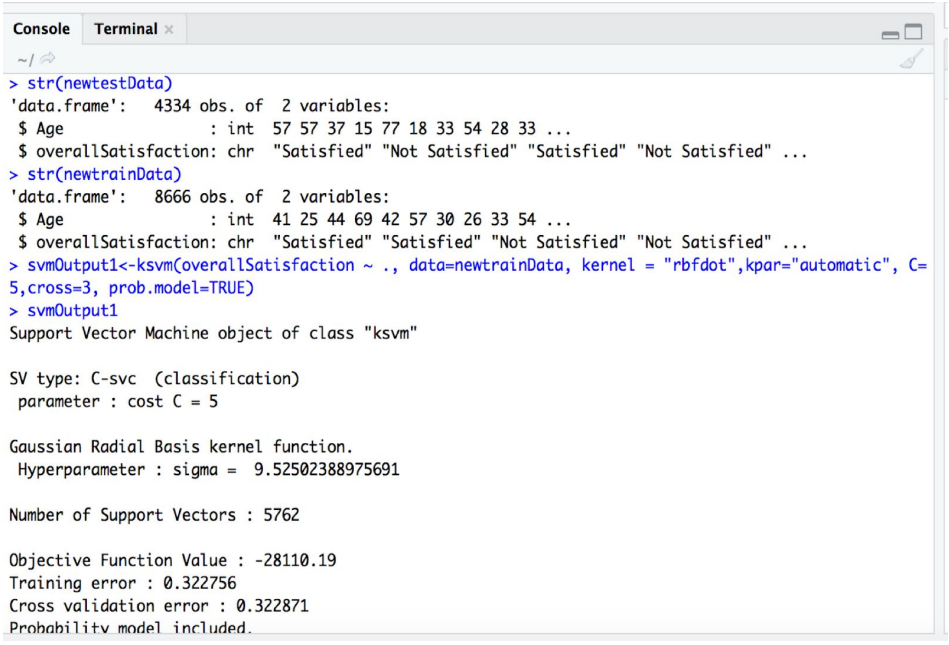
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

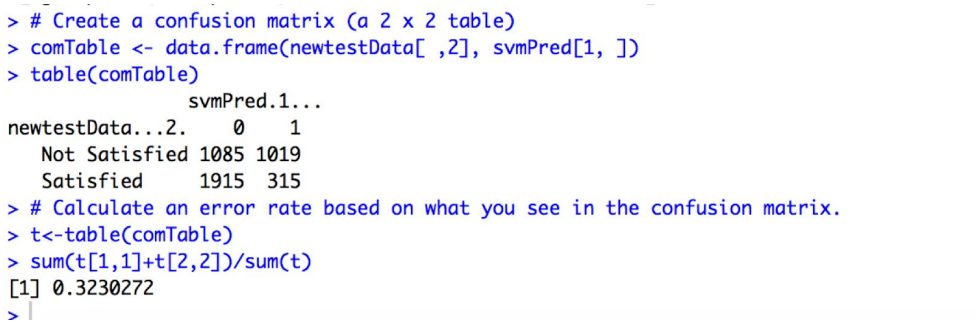
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
```

t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# with one important variables : Age and overallSatisfaction
# error rate : 0. 0.3230272





##
# 3) Build a support vector model using the ksvm( ) function
# with one important variables : Gender and overallSatisfaction
# made dummyData to cut down the size!!
# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(4,29)]
newtestData<-testData[,c(4,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3, prob.model=TRUE)
svmOutput1

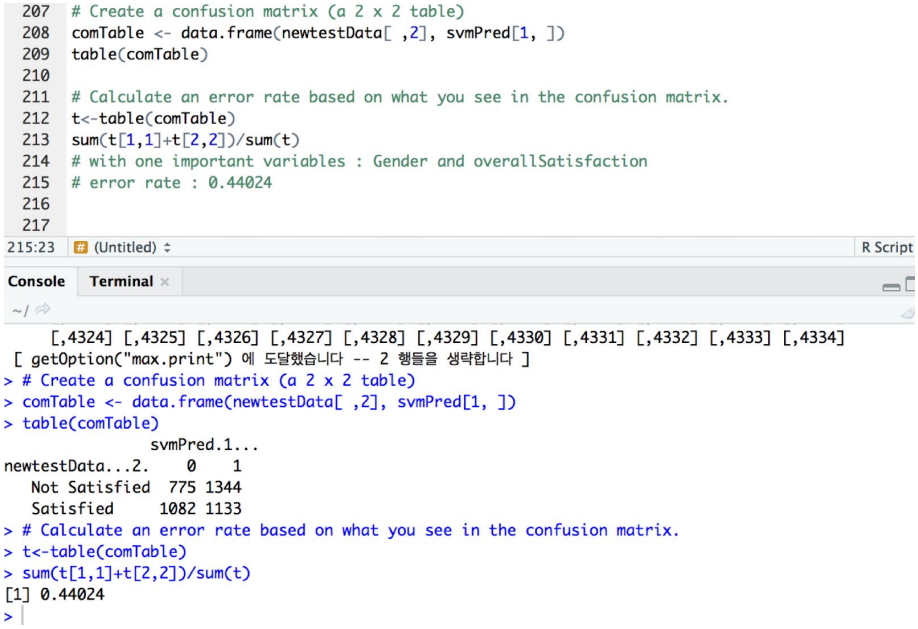
See the output of Support vector histogram to understand outcome

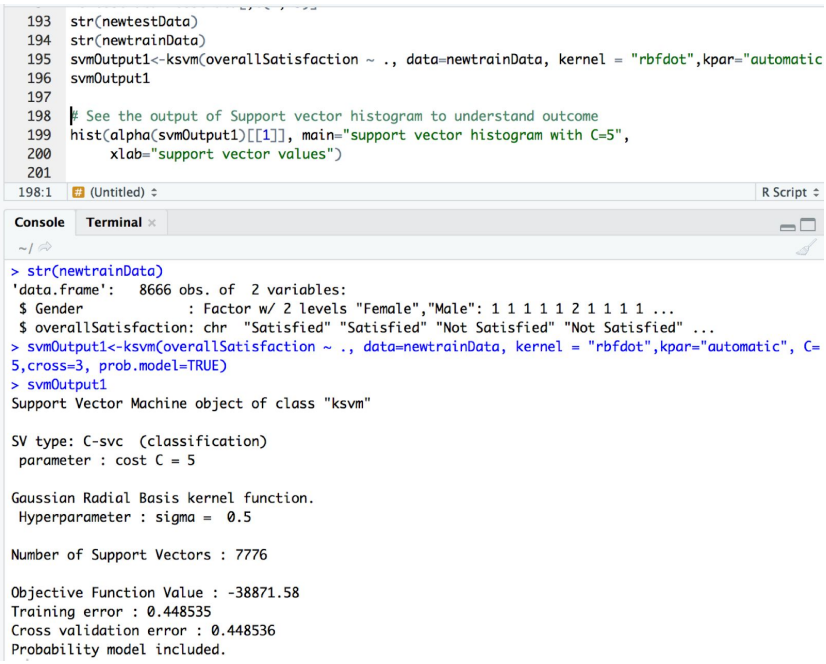
```
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# with one important variables : Gender and overallSatisfaction
# error rate : 0.44024
```





```
193  str(newtestData)
194  str(newtrainData)
195  svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic
196  svmOutput1
197
198  # See the output of Support vector histogram to understand outcome
199  hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
200       xlab="support vector values")
201
198:1  (Untitled)                                                    R Script

Console  Terminal
~/

> str(newtrainData)
'data.frame':   8666 obs. of  2 variables:
 $ Gender            : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 2 1 1 1 1 ...
 $ overallSatisfaction: chr  "Satisfied" "Satisfied" "Not Satisfied" "Not Satisfied" ...
> svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=
5,cross=3, prob.model=TRUE)
> svmOutput1
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 5

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.5

Number of Support Vectors : 7776

Objective Function Value : -38871.58
Training error : 0.448535
Cross validation error : 0.448536
Probability model included.
```

```
#################################################################
# 4) Build a support vector model using the ksvm( ) function
# with one important variables : Price.Sensitivity and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)


# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(5,29)]
newtestData<-testData[,c(5,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

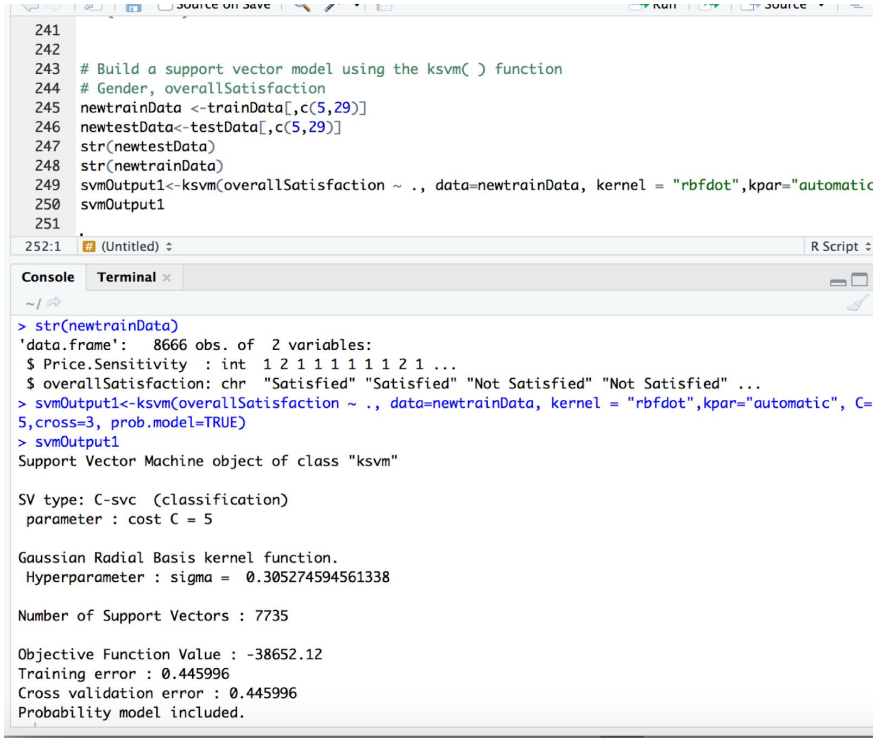
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

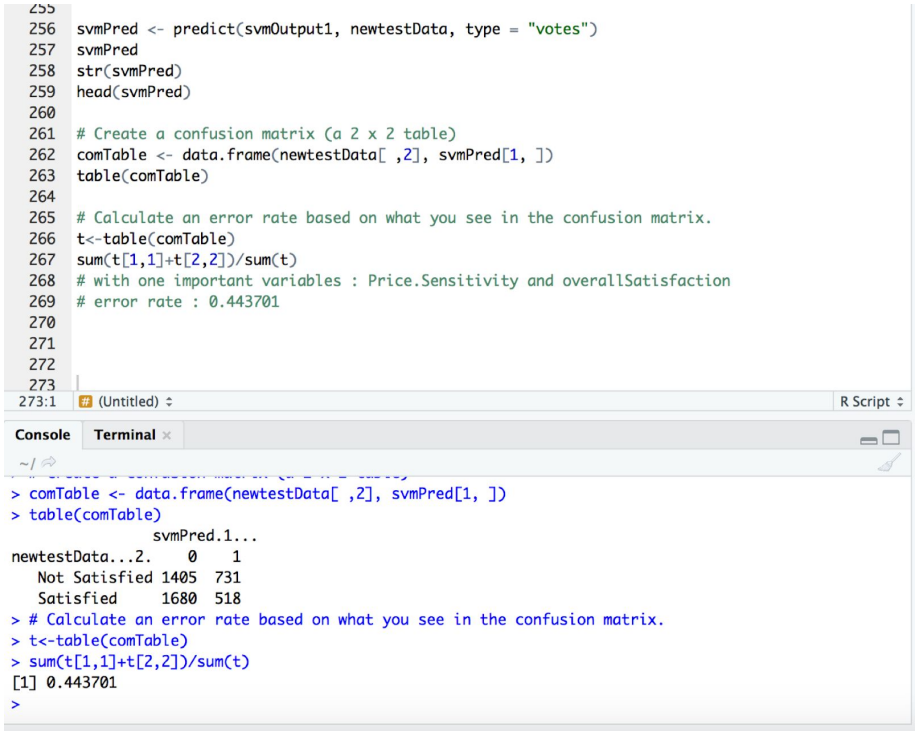
# Create a confusion matrix (a 2 x 2 table)
```

```
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

# with one important variables : Price.Sensitivity and overallSatisfaction
# error rate : 0.443701
```






```
############################################################
# 5) Build a support vector model using the ksvm( ) function
# with one important variables : No.of.Flights.p.a and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(7,29)]
newtestData<-testData[,c(7,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

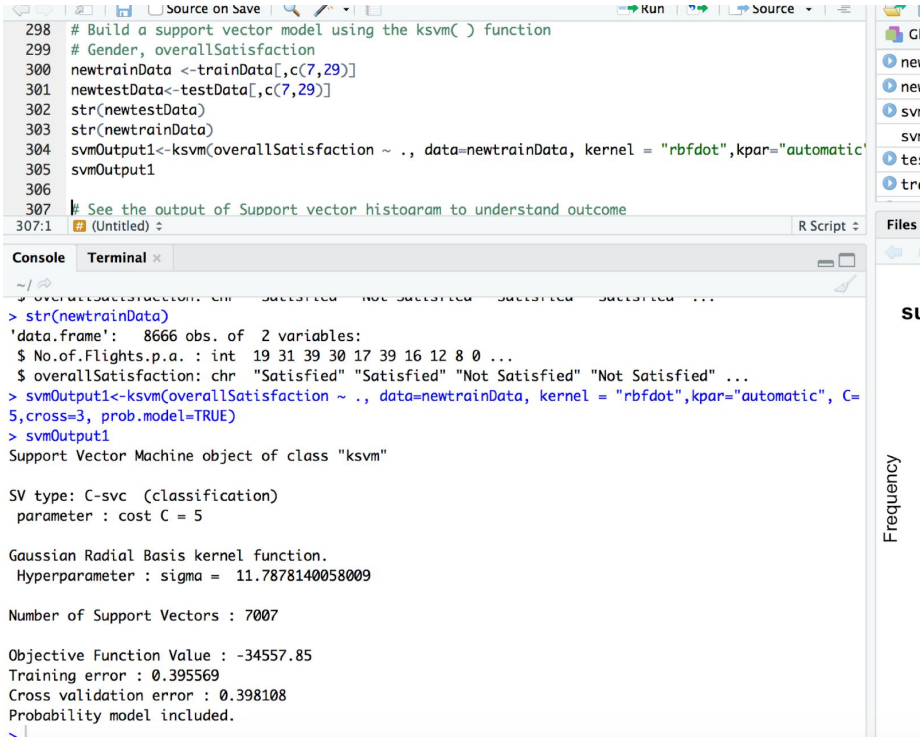
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

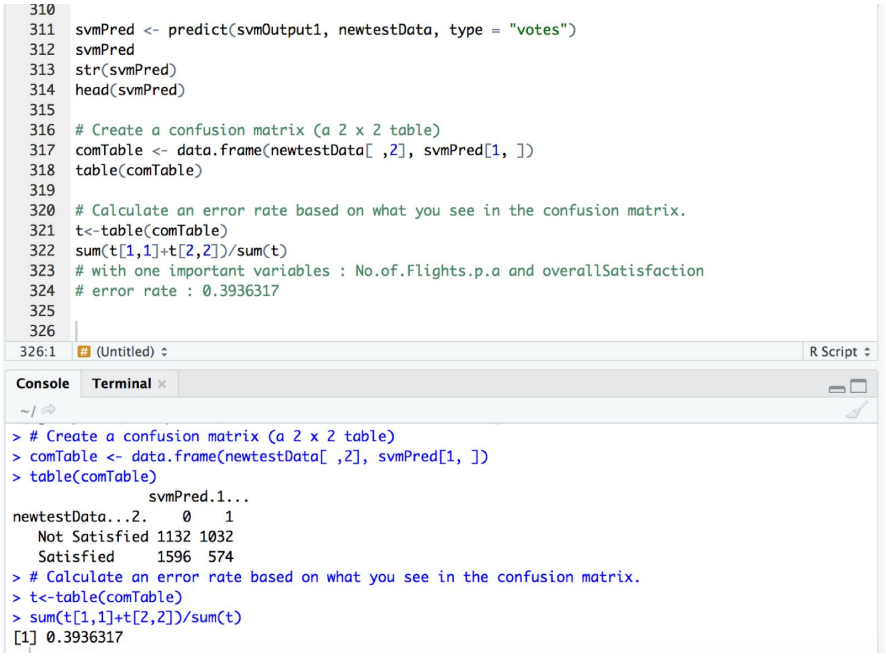
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)
```

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[,2], svmPred[1,])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

with one important variables : No.of.Flights.p.a and overallSatisfaction
error rate : 0.3936317





```
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# with one important variables : No.of.Flights.p.a and overallSatisfaction
# error rate : 0.3936317
```

```
> # Create a confusion matrix (a 2 x 2 table)
> comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
> table(comTable)
                svmPred.1...
newtestData...2.    0    1
   Not Satisfied 1132 1032
   Satisfied     1596  574
> # Calculate an error rate based on what you see in the confusion matrix.
> t<-table(comTable)
> sum(t[1,1]+t[2,2])/sum(t)
[1] 0.3936317
```

```
############################################################
# 6) Build a support vector model using the ksvm( ) function
# with one important variables : Type.of.Travel and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(9,29)]
newtestData<-testData[,c(9,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

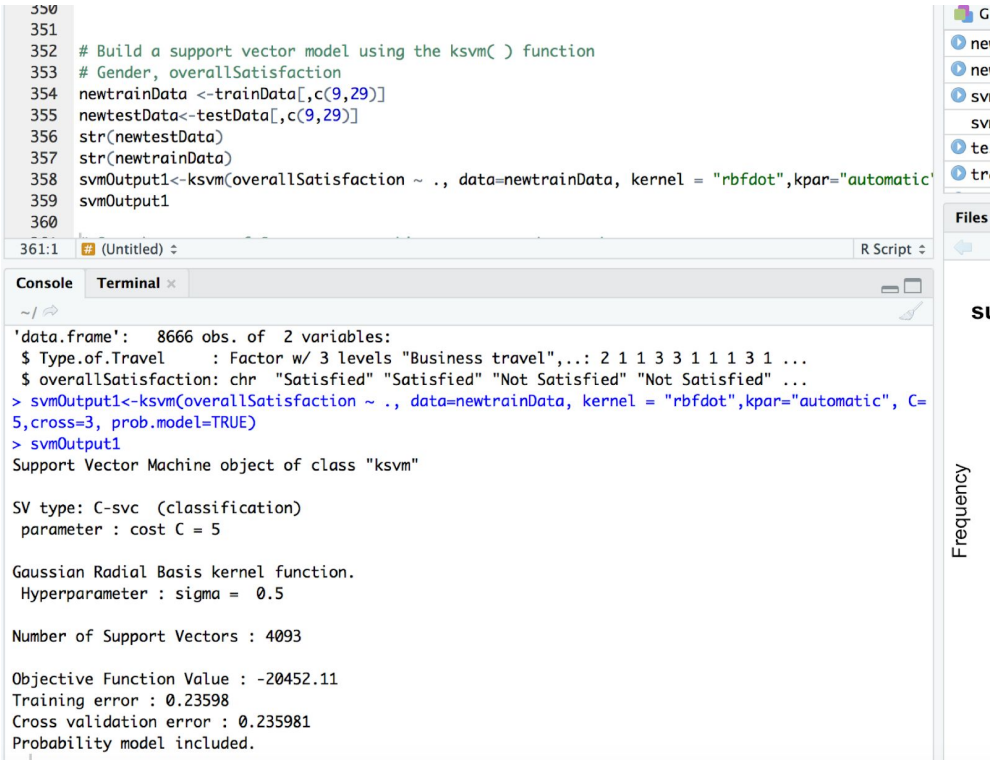
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

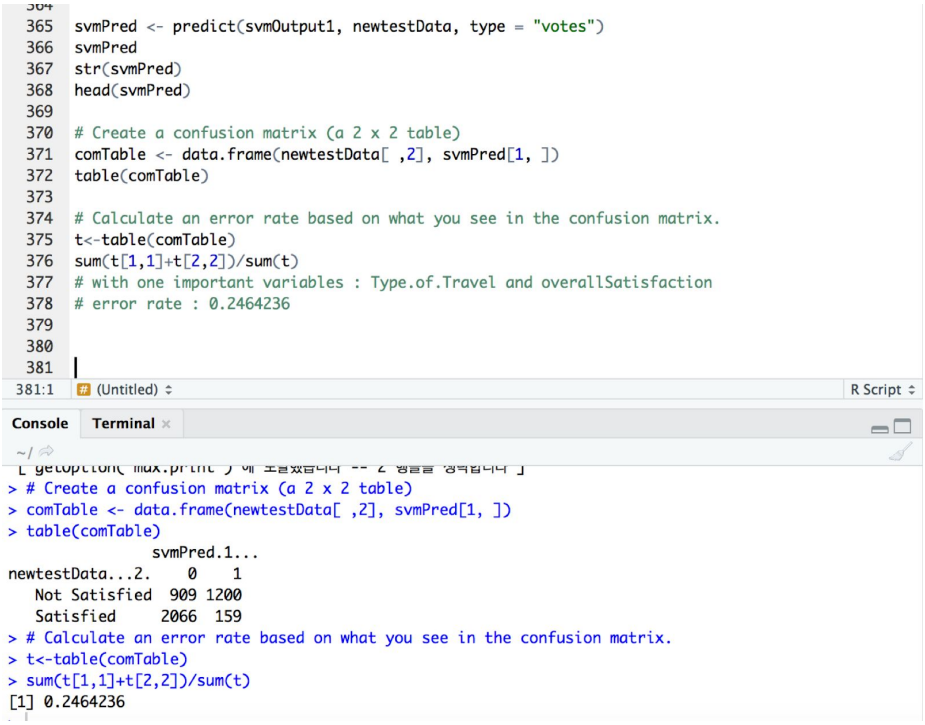
# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
```

table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

with one important variables : Type.of.Travel and overallSatisfaction
error rate : 0.2464236






```
############################################################
# 7) Build a support vector model using the ksvm( ) function
# with one important variables : Class and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(13,29)]
newtestData<-testData[,c(13,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

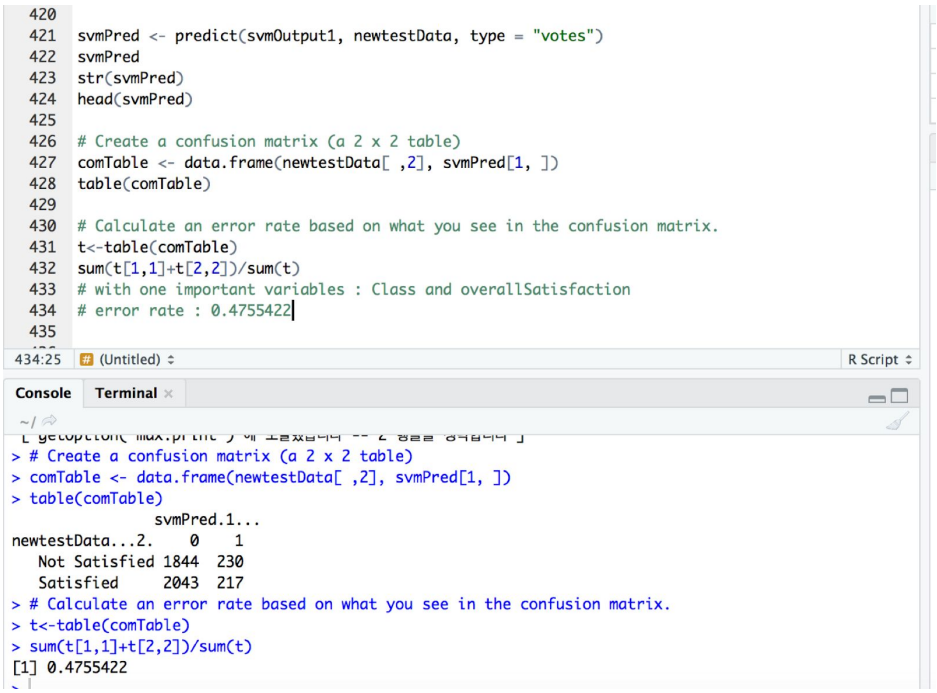
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)
```

```
# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

# with one important variables : Class and overallSatisfaction
# error rate : 0.4755422
```




```
#################################################################
# 8) Build a support vector model using the ksvm( ) function
# with one important variables : Arrival.Delay.greater.5.Mins and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)


# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(28,29)]
newtestData<-testData[,c(28,29)]
str(newtestData)
str(newtrainData)
```

```
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

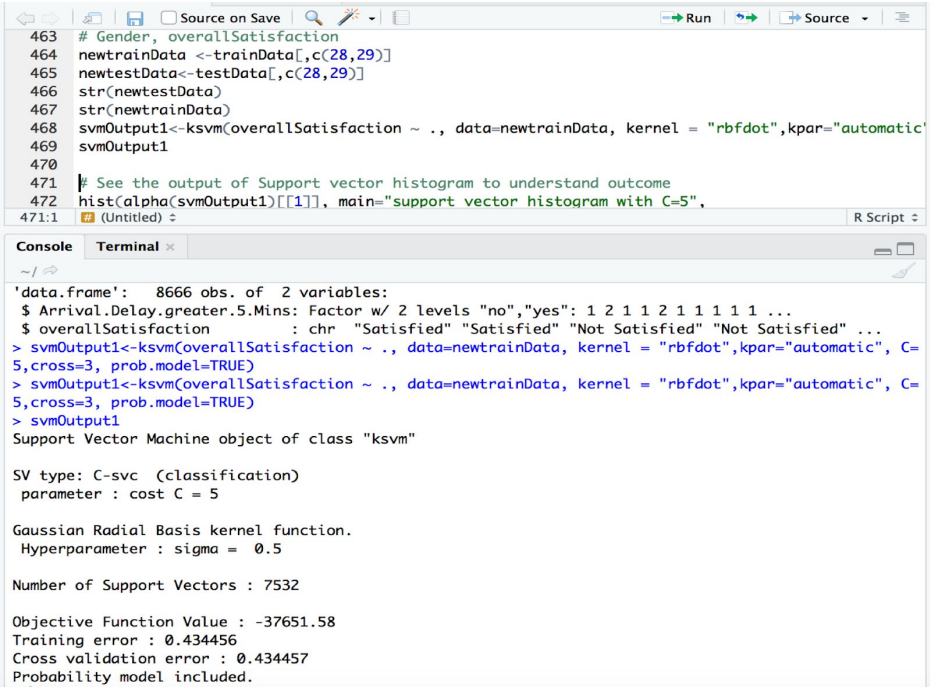
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

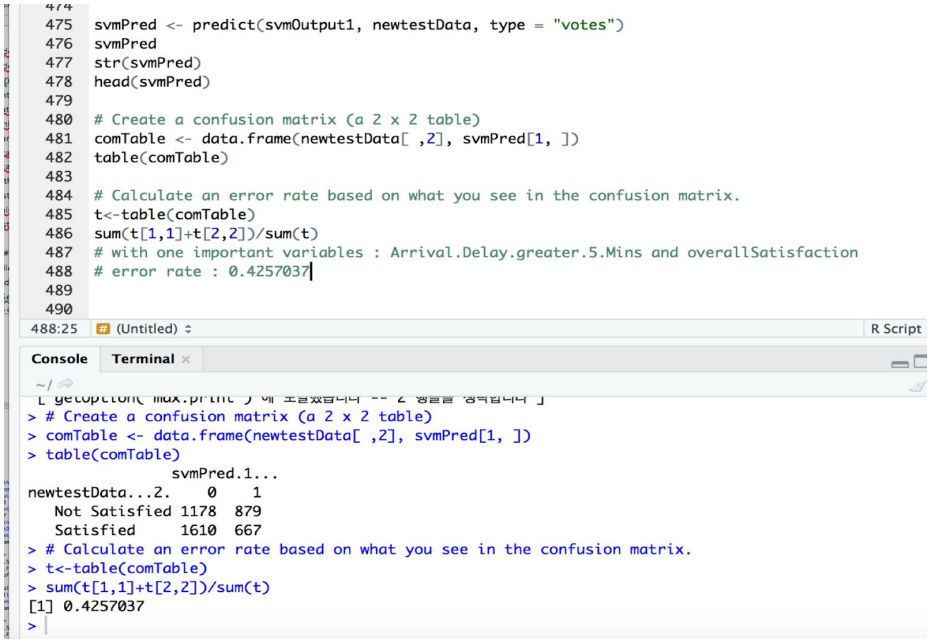
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,2], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

# with one important variables : Arrival.Delay.greater.5.Mins and overallSatisfaction
# error rate : 0.4257037
```






```
############################################################
# 9) Build a support vector model using the ksvm( ) function
# with 3 important variables :Age,No.of.Flights.p.a,Type.of.Traveland overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

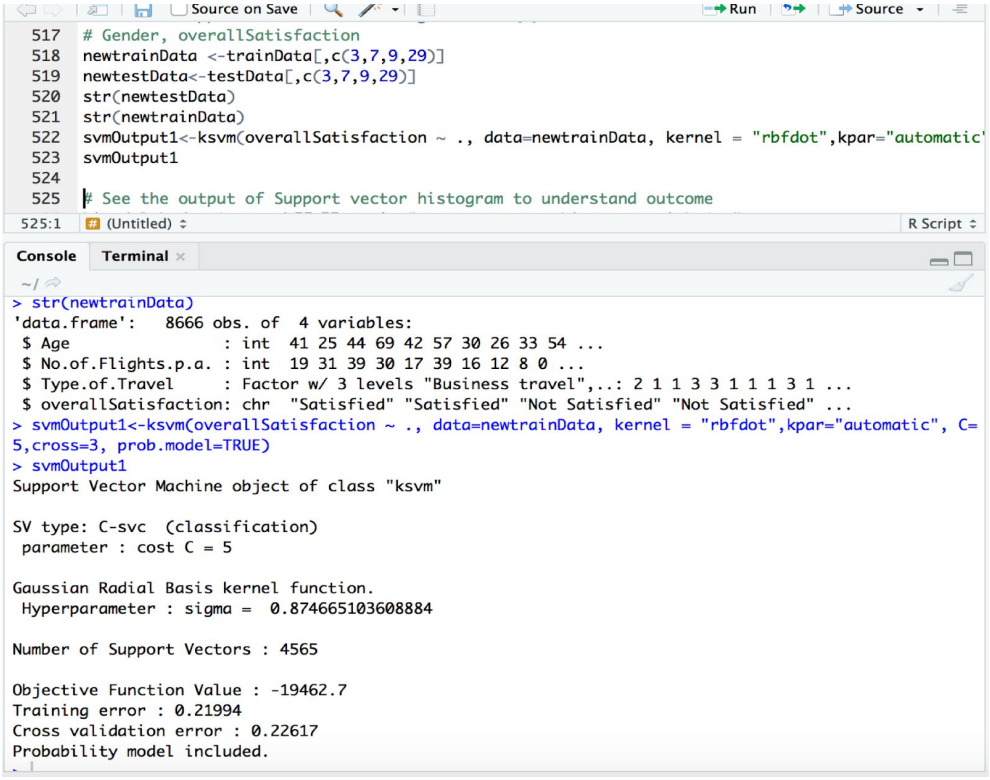
# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(3,7,9,29)]
newtestData<-testData[,c(3,7,9,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

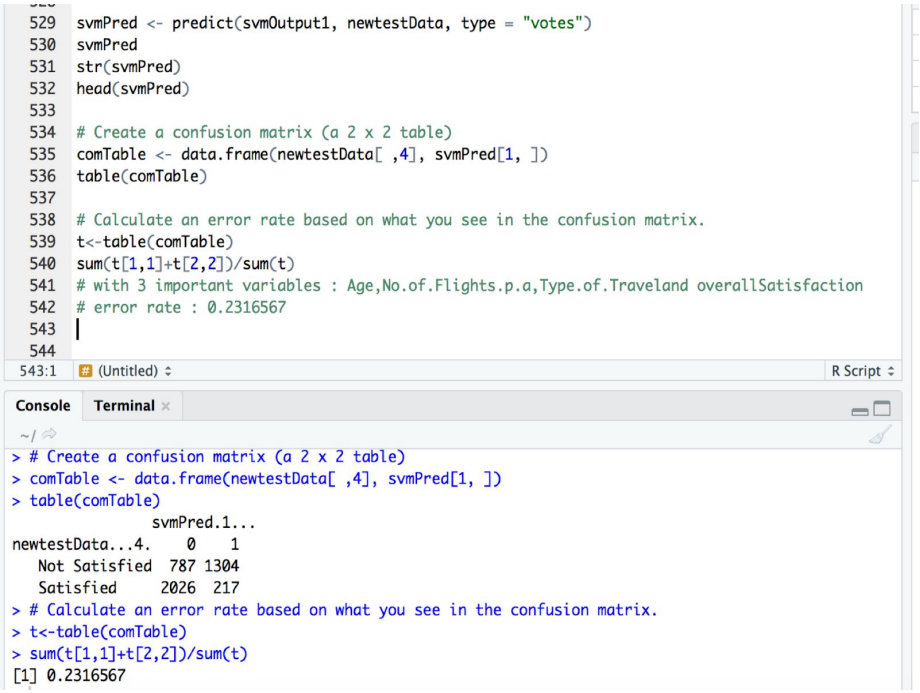
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,4], svmPred[1, ])
table(comTable)
```

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# with 3 important variables : Age,No.of.Flights.p.a,Type.of.Traveland overallSatisfaction
# error rate : 0.2316567






```
############################################################
# 10) Build a support vector model using the ksvm( ) function
# with 4 important variables :
#Age,No.of.Flights.p.a,Type.of.Travel,Arrival.Delay.greater.5.Mins
# and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(3,7,9,28,29)]
newtestData<-testData[,c(3,7,9,28,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

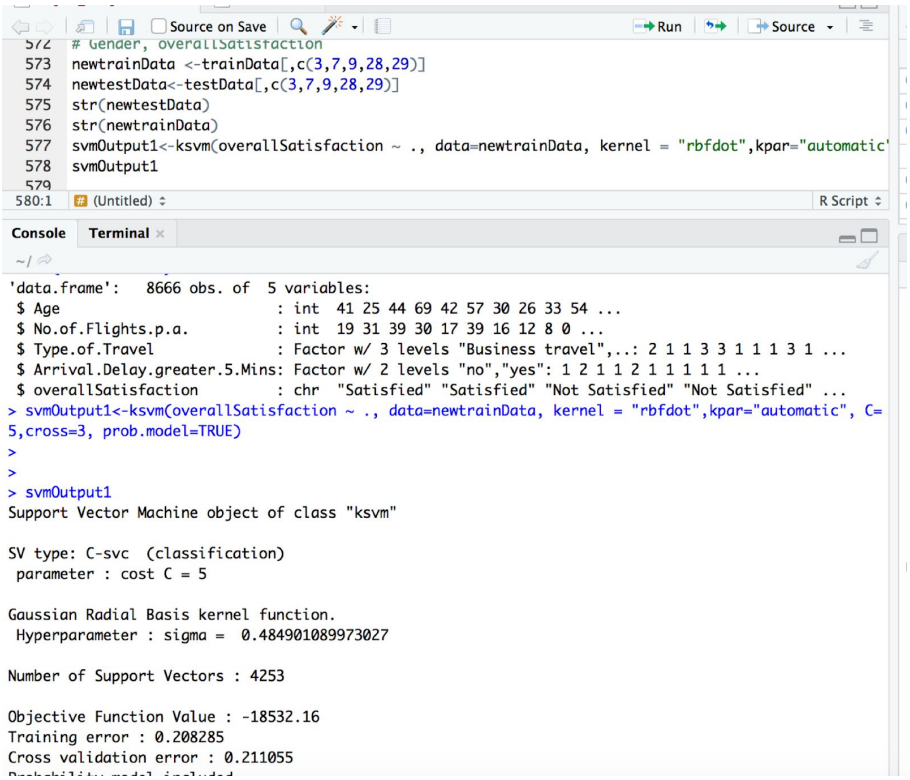
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

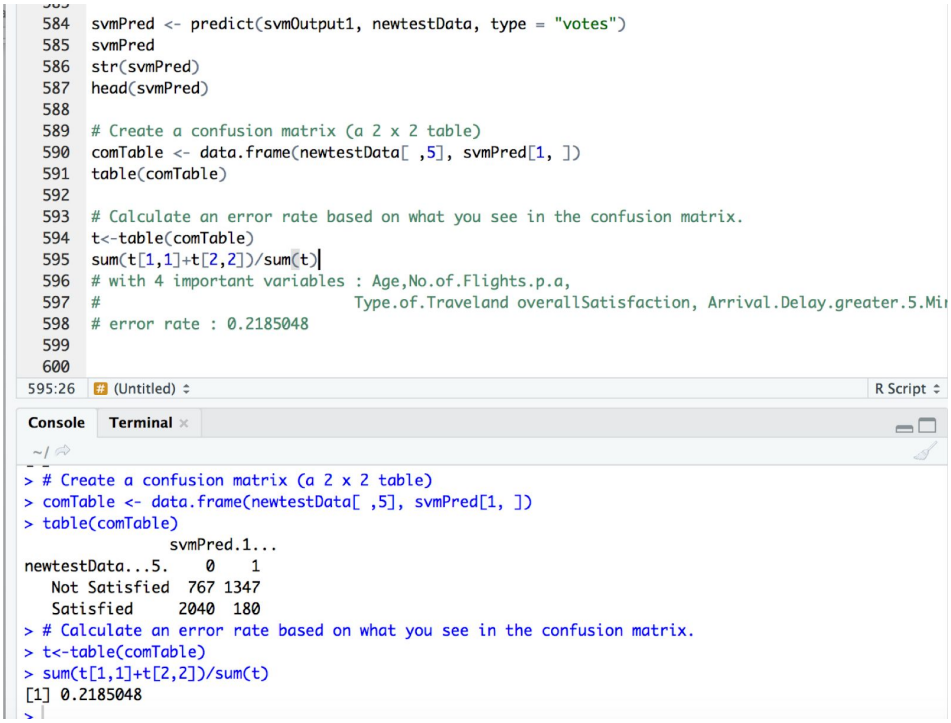
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
```

```
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,5], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# with 4 important variables : Age,No.of.Flights.p.a, Type.of.Traveland overallSatisfaction,Arrival.Delay.greater.5.Mins
# error rate : 0.2185048
```






```
############################################################
# 11) Build a support vector model using the ksvm( ) function
# with 5 important variables :
# Age,Gender,No.of.Flights.p.a,Type.of.Travel,Arrival.Delay.greater.5.Mins
# and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(3,4,7,9,28,29)]
newtestData<-testData[,c(3,4,7,9,28,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

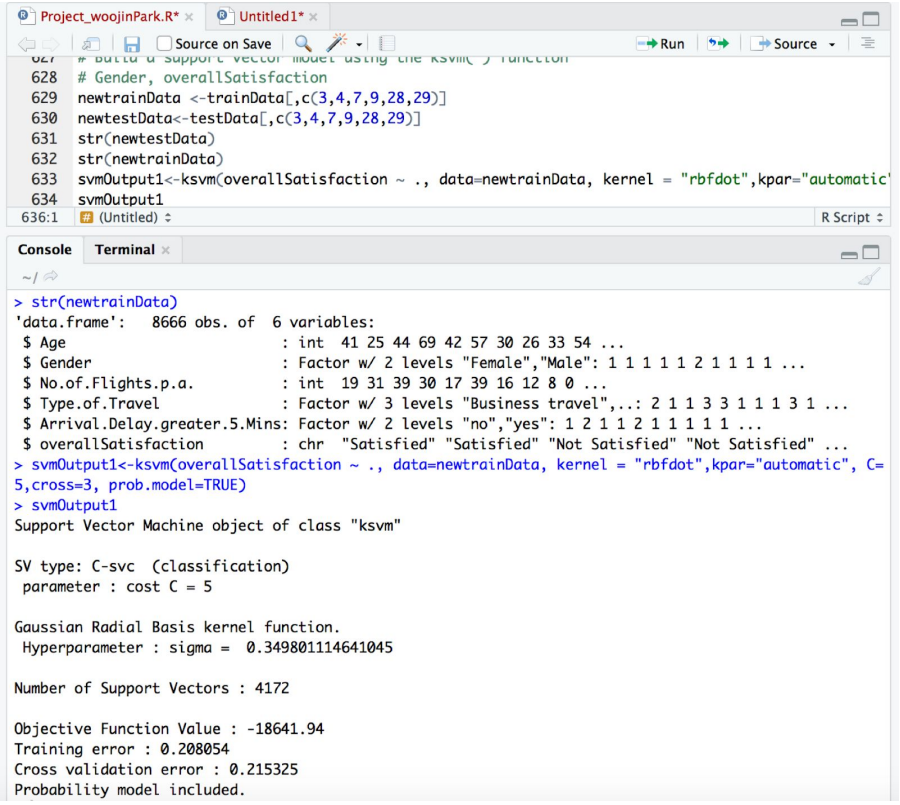
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

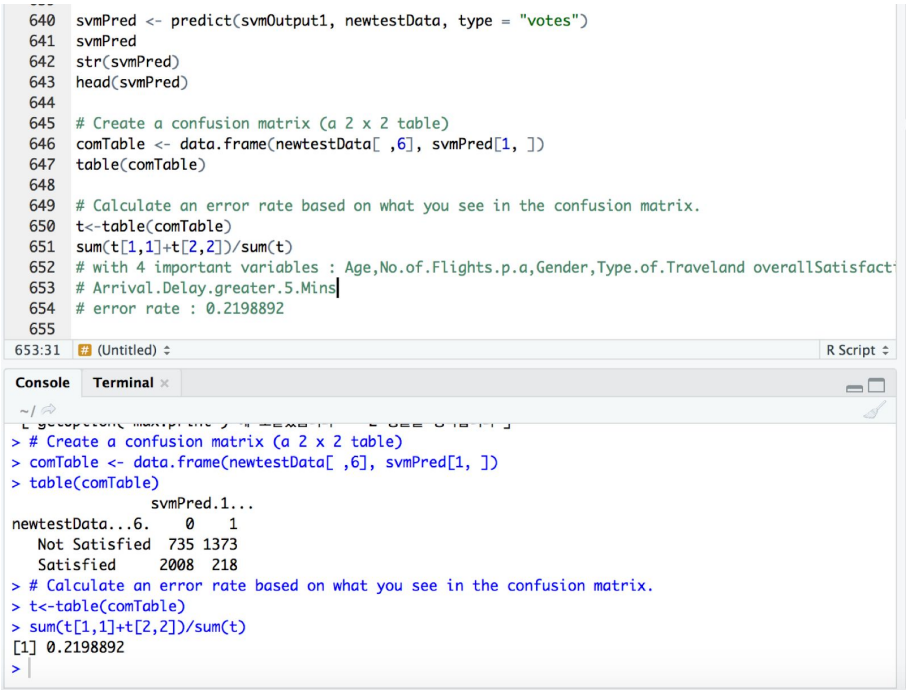
svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
```

```
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,6], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# with 5 important variables :
#Age,No.of.Flights.p.a,Gender,Type.of.Traveland overallSatisfaction
# Arrival.Delay.greater.5.Mins
# error rate : 0.2198892
```






```
##################################################################
# 12) Build a support vector model using the ksvm( ) function
# with 7 important variables :
Age,Gender,No.of.Flights.p.a,Type.of.Travel,Arrival.Delay.greater.5.Mins
# and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 130000, replace =FALSE)
dummyData<-unclean_data[Index,]
str(dummyData)

# Build a support vector model using the ksvm( ) function
# Gender, overallSatisfaction
newtrainData <-trainData[,c(3,4,5,7,9,13,28,29)]
newtestData<-testData[,c(3,4,5,7,9,13,28,29)]
str(newtestData)
str(newtrainData)
svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1

# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)
```

```
# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,8], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

# with 7 important variables :
#Age,No.of.Flights.p.a,Gender,Type.of.Traveland overallSatisfaction
# Arrival.Delay.greater.5.Mins
# error rate : 0.2113521

#############################################################
#13) Build a support vector model using the ksvm( ) function
# include all most important 7 variables 3,4,5,7,9,13,28
#Age,Gender,Price.Sensitivity,No.of.Flights.p.a,Type.of.Travel,Class,Arrival.Delay.greater.5.Mins and overallSatisfaction
# made dummyData to cut down the size!!
Index<- sample(1:nrow(unclean_data), size = 129889, replace =FALSE)
dummyData<-unclean_data[Index,]
View(dummyData)
str(dummyData)

#show how many rows and columns
dim(dummyData) # 129889 rows, 30 columns
# can use table() in factor

randIndex <- sample(1:dim(dummyData)[1])
randIndex
# floor() chops off any decimal part of calculation
cutPoint2_3 <-floor(2*dim(dummyData)[1]/3)
trainData <- dummyData[1:cutPoint2_3,]
str(trainData)
summary(trainData)
testData <- dummyData[randIndex[(cutPoint2_3+1):dim(dummyData)[1]],]
testData
# Use the dim( ) function to demonstrate that the resulting training data set and test data set contain the appropriate number of
cases.
dim(trainData)
dim(testData)
newtrainData <-trainData[,c(3,4,5,7,9,13,28,29)]
newtestData<-testData[,c(3,4,5,7,9,13,28,29)]
str(newtestData)
str(newtrainData)
ksvm(overallSatisfaction ~., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3, prob.model=TRUE)

svmOutput1<-ksvm(overallSatisfaction ~ ., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3,
prob.model=TRUE)
svmOutput1
```

```
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
    xlab="support vector values")

svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,8], svmPred[1, ])
table(comTable)

# Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)

# include all most important 7 variables 3,4,5,7,9,13,28
# error rate : 0.2368755
```

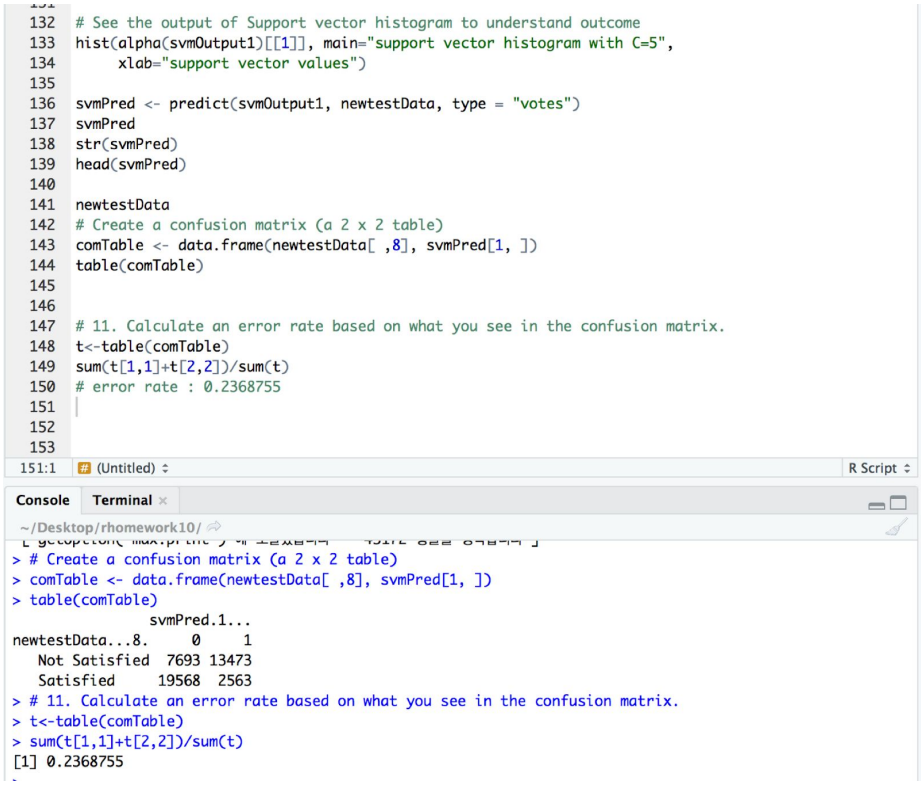
```
> ksvm(overallSatisfaction ~., data=newtrainData, kernel = "rbfdot",kpar="automatic", C=5,cross=3, prob.m
odel=TRUE)
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 5

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.206272679803563

Number of Support Vectors : 40013

Objective Function Value : -191525.6
Training error : 0.214593
Cross validation error : 0.217249
Probability model included.
```



```
# See the output of Support vector histogram to understand outcome
hist(alpha(svmOutput1)[[1]], main="support vector histogram with C=5",
     xlab="support vector values")

svmPred <- predict(svmOutput1, newtestData, type = "votes")
svmPred
str(svmPred)
head(svmPred)

newtestData
# Create a confusion matrix (a 2 x 2 table)
comTable <- data.frame(newtestData[ ,8], svmPred[1, ])
table(comTable)


# 11. Calculate an error rate based on what you see in the confusion matrix.
t<-table(comTable)
sum(t[1,1]+t[2,2])/sum(t)
# error rate : 0.2368755
```

151:1 (Untitled) R Script

Console Terminal

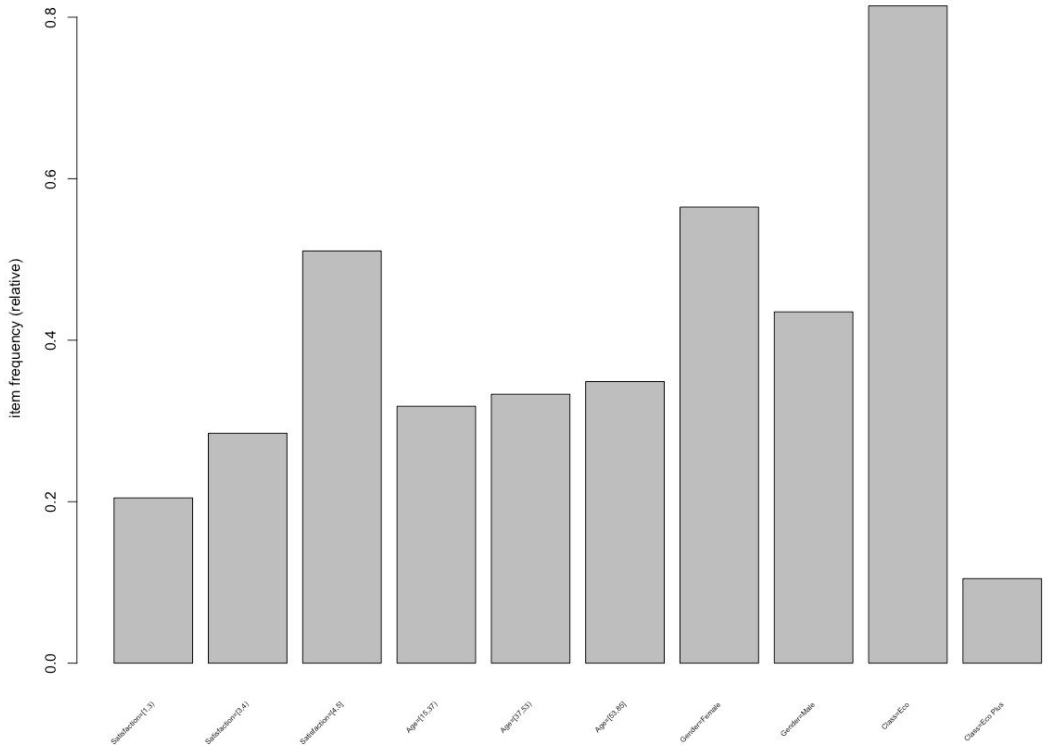
~/Desktop/rhomework10/

```
> # Create a confusion matrix (a 2 x 2 table)
> comTable <- data.frame(newtestData[ ,8], svmPred[1, ])
> table(comTable)
                svmPred.1...
newtestData...8.     0     1
   Not Satisfied  7693 13473
   Satisfied     19568  2563
> # 11. Calculate an error rate based on what you see in the confusion matrix.
> t<-table(comTable)
> sum(t[1,1]+t[2,2])/sum(t)
[1] 0.2368755
```

Association Rules Mining

Association mining is commonly used to make product recommendations by identifying products that are frequently bought together. But what we wanted to see using arules is what are some frequent factors that result to higher satisfaction ratings from customers. With the amount of data we have, we have decided just to make rules based on a few columns that we believe are good predictors for customer satisfaction which we concluded from previous models. Our RHS will represent satisfaction, while LHS, we wanted to just focus on 3 other variables; age, gender, and class.

The first step to this process is looking at the frequency of the items/columns we are looking at which is shown in the plot below.



The plot shows that out of satisfaction ratings, ratings 4-5 are most frequent, then ages between 53 to 85 is most frequent, most of the survey consists of more females than males, and most customers flew in the Economy Class.

We can also look at the summary of the transactions:

```
> summary(flightSurvey)
transactions as itemMatrix in sparse format with
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.5    0.1    1 none FALSE            TRUE       5   0.005      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 649

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 129889 transaction(s)] done [0.09s].
sorting and recoding items ... [11 item(s)] done [0.01s].
creating transaction tree ... done [0.06s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [141 rule(s)] done [0.00s].
creating S4 object  ... done [0.03s].
set of 141 rules
```

Using the apriori function, we can look at how many rules were set. We can see there are 141 rules, which is still too many to view, so want to make it lower amount of rules. To do so, we can change the support amount to a higher number. The below images can show that now, there are only 25 rules, which is a good amount to analyze the data.

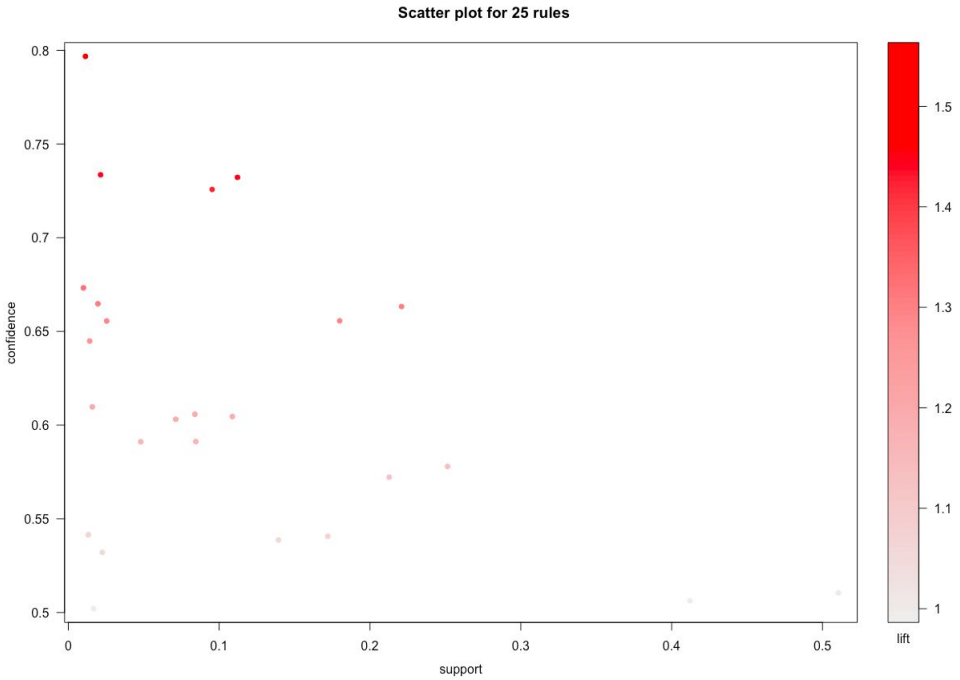
```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
        0.5    0.1    1 none FALSE            TRUE       5    0.01
 minlen maxlen target   ext
      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 1298

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[11 item(s), 129889 transaction(s)] done [0.10s].
sorting and recoding items ... [11 item(s)] done [0.01s].
creating transaction tree ... done [0.06s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [25 rule(s)] done [0.00s].
creating S4 object  ... done [0.03s].
```



Using the apriori() function, we can generate the most relevant set of rules which will show the support, confidence, and lift of those rules. What we want to see is high lift, because it represents that the chance of LHS rule occurring together with RHS rule.

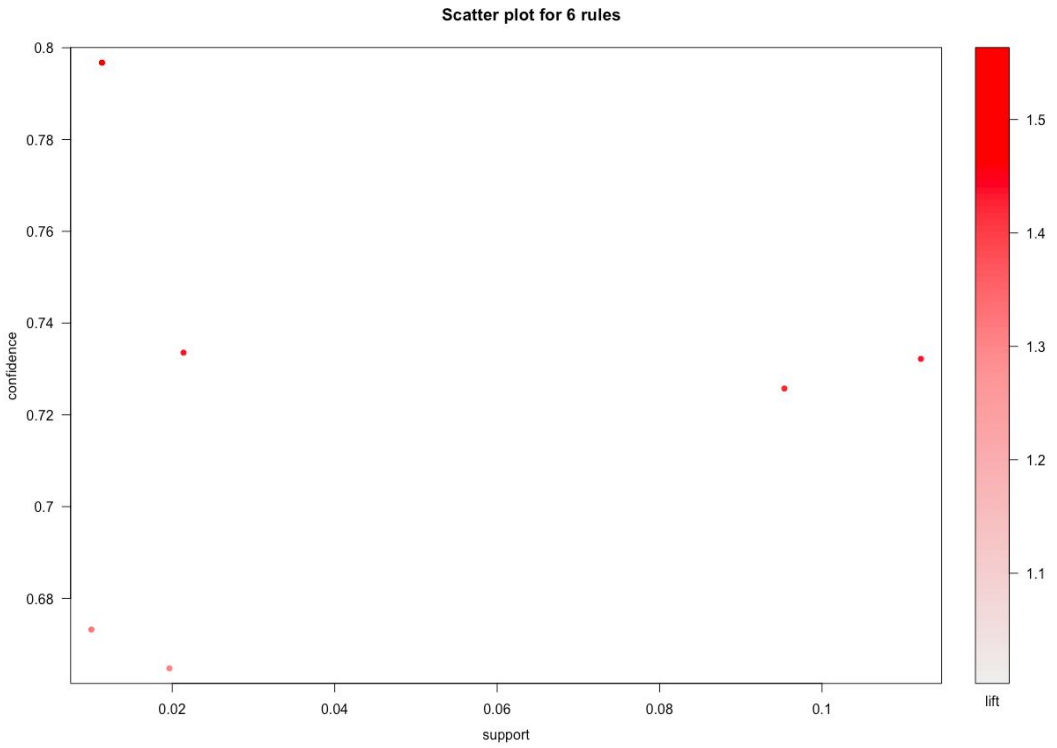
```
     lhs                                                rhs                  support    confidence lift      count
[1]  {}                                              => {Satisfaction=[4,5]} 0.51052822 0.5105282  1.0000000 66312
[2]  {Class=Business}                                => {Satisfaction=[4,5]} 0.04800253 0.5911073  1.1578348  6235
[3]  {Age=[15,37)}                                   => {Satisfaction=[4,5]} 0.17203150 0.5406746  1.0590494 22345
[4]  {Age=[37,53)}                                   => {Satisfaction=[4,5]} 0.22092710 0.6632766  1.2991968 28696
[5]  {Gender=Male}                                   => {Satisfaction=[4,5]} 0.25145316 0.5779174  1.1319989 32661
[6]  {Class=Eco}                                     => {Satisfaction=[4,5]} 0.41212112 0.5062657  0.9916507 53530
[7]  {Age=[15,37),Class=Business}                    => {Satisfaction=[4,5]} 0.01591359 0.6097345  1.1943209  2067
[8]  {Age=[37,53),Class=Business}                    => {Satisfaction=[4,5]} 0.02138749 0.7335622  1.4368690  2778
[9]  {Gender=Male,Class=Business}                    => {Satisfaction=[4,5]} 0.02545250 0.6555622  1.2840860  3306
[10] {Gender=Female,Class=Business}                  => {Satisfaction=[4,5]} 0.02255002 0.5320618  1.0421789  2929
[11] {Age=[15,37),Class=Eco Plus}                    => {Satisfaction=[4,5]} 0.01679896 0.5020709  0.9834341  2182
[12] {Age=[37,53),Class=Eco Plus}                    => {Satisfaction=[4,5]} 0.01964755 0.6647564  1.3020954  2552
[13] {Gender=Male,Class=Eco Plus}                    => {Satisfaction=[4,5]} 0.01325747 0.5415094  1.0606846  1722
[14] {Age=[15,37),Gender=Male}                       => {Satisfaction=[4,5]} 0.08389471 0.6057928  1.1865999 10897
[15] {Age=[15,37),Class=Eco}                         => {Satisfaction=[4,5]} 0.13931896 0.5386997  1.0551810 18096
[16] {Age=[37,53),Gender=Male}                       => {Satisfaction=[4,5]} 0.11217270 0.7321976  1.4341961 14570
[17] {Age=[37,53),Gender=Female}                     => {Satisfaction=[4,5]} 0.10875440 0.6045795  1.1842235 14126
[18] {Age=[37,53),Class=Eco}                         => {Satisfaction=[4,5]} 0.17989206 0.6556485  1.2842551 23366
[19] {Gender=Male,Class=Eco}                         => {Satisfaction=[4,5]} 0.21274319 0.5722066  1.1208128 27633
[20] {Age=[37,53),Gender=Male,Class=Business}        => {Satisfaction=[4,5]} 0.01134815 0.7967568  1.5606517  1474
[21] {Age=[37,53),Gender=Female,Class=Business}      => {Satisfaction=[4,5]} 0.01003934 0.6732060  1.3186460  1304
[22] {Age=[37,53),Gender=Female,Class=Eco Plus}      => {Satisfaction=[4,5]} 0.01418904 0.6448565  1.2631164  1843
[23] {Age=[15,37),Gender=Male,Class=Eco}             => {Satisfaction=[4,5]} 0.07122235 0.6031032  1.1813318  9251
[24] {Age=[37,53),Gender=Male,Class=Eco}             => {Satisfaction=[4,5]} 0.09536604 0.7257441  1.4215553 12387
[25] {Age=[37,53),Gender=Female,Class=Eco}           => {Satisfaction=[4,5]} 0.08452602 0.5912224  1.1580602 10979
```

The above shows the 25 rules and with this, we want to focus on high support and confidence, while higher lift implies that the rules is more likely to happen together compared to other rules that are assumed to be unrelated which makes it interesting. I have decided to focus on highest confidence, so I sorted the data from highest to lowest:

```
    lhs                                                rhs                  support    confidence lift     count
[1] {Age=[37,53),Gender=Male,Class=Business}        => {Satisfaction=[4,5]} 0.01134815 0.7967568  1.560652  1474
[2] {Age=[37,53),Class=Business}                    => {Satisfaction=[4,5]} 0.02138749 0.7335622  1.436869  2778
[3] {Age=[37,53),Gender=Male}                       => {Satisfaction=[4,5]} 0.11217270 0.7321976  1.434196 14570
[4] {Age=[37,53),Gender=Male,Class=Eco}             => {Satisfaction=[4,5]} 0.09536604 0.7257441  1.421555 12387
[5] {Age=[37,53),Gender=Female,Class=Business}      => {Satisfaction=[4,5]} 0.01003934 0.6732060  1.318646  1304
[6] {Age=[37,53),Class=Eco Plus}                    => {Satisfaction=[4,5]} 0.01964755 0.6647564  1.302095  2552
```

The highest confidence shows that about 80% of the time that males from the ages between 37 to 53 that flew in the business class will give a high satisfaction rating. We can understand that even though there were less males in the survey, that they were generally more satisfied with their flight.

Can look at a plot with rules that only have a lift over 1.3:



We also decided to take a look at what factors would most likely lead to a lower satisfaction rating.

```
    lhs                                          rhs                    support    confidence lift     count
[1] {Age=[53,85],Gender=Female,Class=Eco}      => {Satisfaction=[1,3)} 0.05668686 0.3570805  1.744427  7363
[2] {Age=[53,85],Gender=Female}                => {Satisfaction=[1,3)} 0.07233099 0.3522817  1.720984  9395
[3] {Age=[53,85],Class=Eco}                    => {Satisfaction=[1,3)} 0.09536604 0.3393234  1.657679 12387
[4] {Age=[53,85],Gender=Female,Class=Eco Plus} => {Satisfaction=[1,3)} 0.01100170 0.3384652  1.653487  1429
[5] {Age=[53,85]}                              => {Satisfaction=[1,3)} 0.11703839 0.3356072  1.639525 15202
[6] {Age=[53,85],Class=Eco Plus}               => {Satisfaction=[1,3)} 0.01381949 0.3311197  1.617602  1795
```

Results:

This was very interesting to look at, because it showed that about 36% of the time that females who were 53 to 85 years old that flew in the economy class, were going to give a bad satisfaction rating. While the confidence level is a bit lower than the previous one, the lift was a lot higher which means it is an interesting rule to look into. Definitely a factor that should be looked into for further insight to make suggestions for airlines to improve upon.

Code for Association Rule Mining:

```
#So the first step of creating arules is installing the necessary packages:
install.packages("arules")
install.packages("arulesViz")

library(arules)
```

```
library(arulesViz)

#Then, I wanted to focus on just satisfaction, age, class, and gender. So i created a new data set with just those columns:
newdata <- data[,1:13]
newdata <- newdata[,-2]
newdata <- newdata[,-4:-11]

#Must make the continuous variables into factors:
newdata$Satisfaction <- discretize(newdata$Satisfaction, methods = NULL, default = NULL)
newdata$Age <- discretize(newdata$Age, methods = NULL, default = NULL)

#Once this is completed, for arules to work, must make the data set into transactions:
flightSurvey <- as(newdata, "transactions")

#can inspect the data, but there is still too many variable to look at, so we want to just see the items that are most frequent:
inspect(flightSurvey)

itemFrequency(flightSurvey)
Satisfaction=[1,3)   Satisfaction=[3,4)   Satisfaction=[4,5]
         0.2046979          0.2847739          0.5105282
      Age=[15,37)        Age=[37,53)        Age=[53,85]
         0.3181794          0.3330844          0.3487362
   Gender=Female      Gender=Male     Class=Business
         0.5648977          0.4351023          0.0812078
      Class=Eco     Class=Eco Plus
         0.8140412          0.1047510

itemFrequencyPlot(flightSurvey, support = 0.1, cex.names=0.5)
summary(flightSurvey)

apriori(flightSurvey, parameter = list(support = 0.005, confidence = 0.5))

ruleset <- apriori(data=flightSurvey, parameter = list(support = .01, confidence = 0.5), appearance = list(default="lhs",
rhs="Satisfaction=[4,5]"))
inspect(ruleset)
plot(ruleset)

#organized it to see high confidence to lower confidence levels
rules_conf <- sort(ruleset, by="confidence", decreasing = TRUE)
inspect(head(rules_conf))
#Can look at a plot with rules that only have a lift over 1.3
goodrules <- ruleset[quality(ruleset)$lift > 1.3]
plot(goodrules)
```

Actionable Insights

We concluded the following insights after analyzing the dataset:

Insight 1:

- We found that gender played a significant role in determining the customer's satisfaction rating. More precisely, based on our analysis, we found that females in general provided low satisfaction ratings than males.
- We recommend the airline companies to focus more on providing better amenities and assistance to females. We also found out that females between the ages of 53 and 85 travelling in economy class rated satisfaction rating the lowest among all the age groups. We recommend the airline companies to create a strategy that cater to that specific age group, for example providing additional discounts on flights or provide them with earlier check-in than others.

Insight 2:

- After analysis, we found that the class in which the customers travel was significant but not the deciding factor.
- Customers travelling in business class and economy plus class majorly had a satisfaction rate of 3 to 5 while customers travelling in economy class had significant numbers in every batch of satisfaction rate.
- So one recommendation we would make is more seats of business and economy class should be introduced to get an overall high satisfaction rate and the services in economy class should be improved.

Insight 3:

- We also found that the amount of flight delay was negatively impact customer satisfaction rating.
- Given that there are a lot of factors can lead to flight delay, and some of them are uncontrollable ones, such as bad weather, namely snow or heavy rainfall, and air travel control, the airline companies can actually do very little regarding avoiding the flight delay. As such, we recommend them to provide certain extra amenities to ease the irritation and anger of the passengers.

- Companies can provide passengers with discounts if the flight is delayed beyond a certain number of hours or make the lounge area accessible to passengers during this special time without extra charge.

Insight 4:

- The Airline Status of a company has also been found to play a significant role in the overall customer satisfaction rating.
- Based on our analysis, we found that Silver status has been predicted to have the highest satisfaction rate followed by Gold.
- We recommend the airline companies to make promotion of Bronze and Blue status to Silver easier by providing significant promotions on status upgrade. We also recommend to provide more discounts and promotions to the Bronze and Blue status to help improve the satisfaction rate.

MIDST Link: https://midst.syr.edu/project/cs_project1