CUSTOMIZATION TO RETAIN CUSTOMERS

Group 5 project by Liya Zhou, Siyun Ding, Renjie Yin, Haolin Liu, Siddhant Prashant Bandiwadekar

Content

1. Project Introduction

Taking airlines has been a more and more common and convenient way for people to travel around the US, a country which has a vast territory, huge population and increasingly large purchasing power, thus there is an important concern for an airline company to develop high quality of service by making improvements in their advertising, product and service features, value-added products and services, in their customers' ease of use, reliability and loyalty. By improving customer experience and reduce customer churn, Southeast Airlines would become more competitive and successful in the market.

With the fact that customers were valuing the loyalty program less, just relying on their loyalty program was not enough in keeping low customer churn, thus in this project, we try to extract valuable attributes and generate models to help analyze other important influencers that companies should pay attention to.

1.1 Business Questions

According to the existing data, the specific business questions we want to explore are:

What's the relationship between loyalty and likelihood to recommend?

From partner analysis:

- Which airlines cooperate with their partners are being taken frequently by customers?

From customer analysis:

- Which characters of customers themselves influence the recommendation level?
- What's the relationship among Shopping Amount at Airport, Eating, Drinking at Airport and Class?

From flight analysis:

- Will flight distance influence customer's recommendation?
- From which origin city/state to which destination city/state has dense flight and belongs to which partner company? What's the travel type of customers on those flights?

From feedback analysis:

- What's the correlation between the most frequent words with the customers' loyalty and LTR?

By finding the answer to those questions, we would figure out and analyze relations between Net Promoter Scores and other possible factors to get insights on how to improve customer experience.

1.2 Programming methods

There are totally 10,282 observations and variables in the document. To make the investigation of data clear and concise, we decided to give up some attributes such as Year of First Flight, Total Frequent Flyer Accounts, Day of the Month. We divided the rest variables into 3 parts, that is, customer and airline information.

- **Place**: Destination.City, Origin.City, Origin.State, Destination.State, olong, olat, dlong, dlat.
- **Customer**: Age, Gender, Price.Sensitive, Flights.Per.Year, Type of Travel, Shopping.Amount.at.Airport, Eating.and.Drinking.at.Airport, Class, Year.Length, Airline.Status, Freetext.
- **Airline**: Day.of.Month, Partner.Name, Scheduled.Departure.Hour, Departure.Delay.in.Minutes, Arrival.Delay.in.Minutes, Flight.cancelled, Flight.time.in.minutes, Flight.Distance, Flight.Month.

We decided to analyze these 31 variables and 10,282 samples, which we think are more likely to influence customer experience. By using Data Acquisition, Cleansing, Transformation, Munging, we preprocess our data for later use. Then using descriptive statistic and Data Visualization to make the data more vivid and clear to present. Data Modeling are used to further find out factors related to the customer churn and the extent they influence the customer experience. We will then provide some insights and recommendations to the Southeast Airlines. Lastly, the appendix of our codes would also be added in our word document of project.

2. Data Acquisition, Cleansing, Transformation, Munging

Before the analysis, the most important thing is to extract and pre-process data. In the real world, data is usually incomplete with missing some attribute values of interest, inconsistent including differences in code or names, and very vulnerable to noise. Because the database is too large and the datasets often come from multiple heterogeneous data sources, low-quality data will lead to low-quality mining results. Data preprocessing is a reliable way to solve the problem. So, cleaning the data and doing some basic analysis will help us find the relationship among different variables to contribute to further analysis. Facing the new data set, we will decide which variables will be kept with data munging.

2.1 Data Extraction

The first step of our project is to extract data from a JSON document. Initially, we downloaded the data and it consisted of 10282 different observations of 32 variables. After the initial analysis and observation of the data set, we divided the variables into three parts and decided to give up some of the variables. Following is the code for Data Extraction:

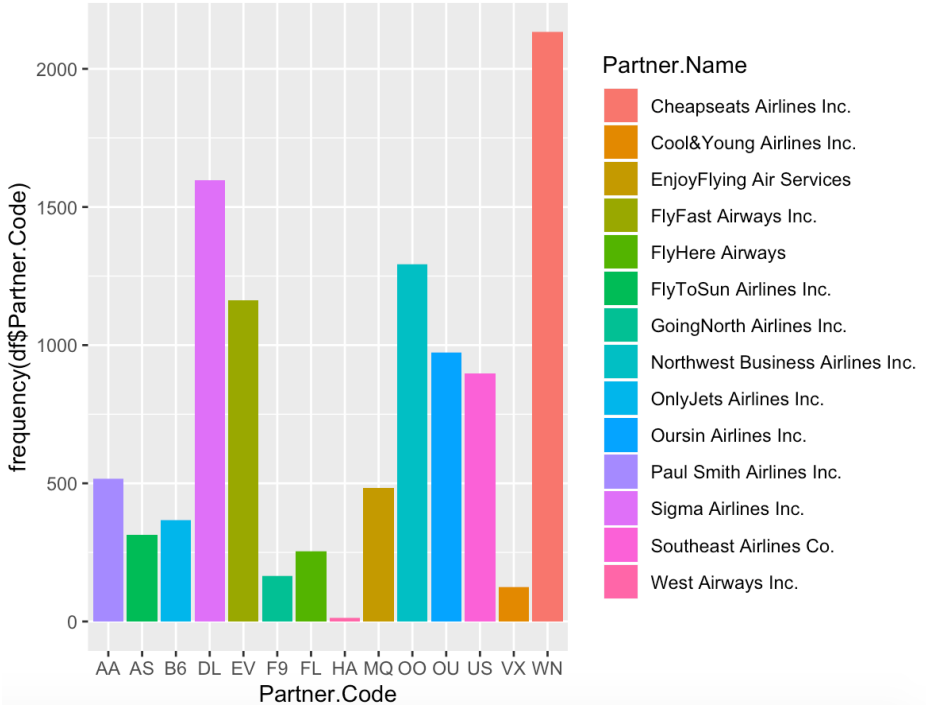
```
############################ DATA EXTRACTION ###################################
#read JSON document and save as a data frame
library(jsonlite)
df <- jsonlite::fromJSON("~/Desktop/fall2019-survey-M07.json")
View(df)
str(df)
```

2.2 Data Preprocessing

Before further analyzing the data, the priority is to clean the data. First, we deleted columns that we would not use in our analysis. Through our rough analysis, the variable "Total.Freq.Flyer.Accts" was deleted. Then, we found out columns with a lot of NA values, such as "Departure.Delay.in.Minutes", "Arrival.Delay.in.Minutes" and "Flight.time.in.minutes". We also changed data type such as changing integer into numeric data. Next, we judged whether the column is the normal distribution to decide the way to fill NAs. The average value can be used when the distribution of variables approximates the normal distribution. Skewness distribution

generally uses the median to represent the trend of the data center. So, we filled NAs with median. To make the data clearer, we deleted the state abbreviation in destination and origin city. To fulfill our further analysis, we extracted flight months and year as two new variables. Then the flight date was deleted.

From Figure 2-1, we discovered that each Partner Code is on behalf of a company. So we deleted the Partner Code to make the data set more concise.



Then we added a new variable of the length of the year in which a customer had taken the flight. After that, we deleted the used variables “Year.of.First.Flight” and “Flight.Year”. According to the definition of Net Promoter Score, we created a new variable “recommend.level” which was created based on the likelihood to recommend. If the likelihood was above eight, we recognized the customer that he was willing to recommend. On the opposite, if the likelihood was no more than eight, the customer was not able to recommend the airline. Thus, the variable was divided into “yes” and “no”, served as a categorical variable. We recorded the new data set as a basic one for further analysis.

2.3 Summary of Data Processing

In summary, the data preprocessing provides a clearer view for us to analyze and solve the business questions, which contributes to better presentation in building models. We then roughly

divided variables into four parts: customers, flights, partners, and feedback. For each part, we developed an independent analysis.

3. DESCRIPTIVE STATISTICS & VISUALIZATION

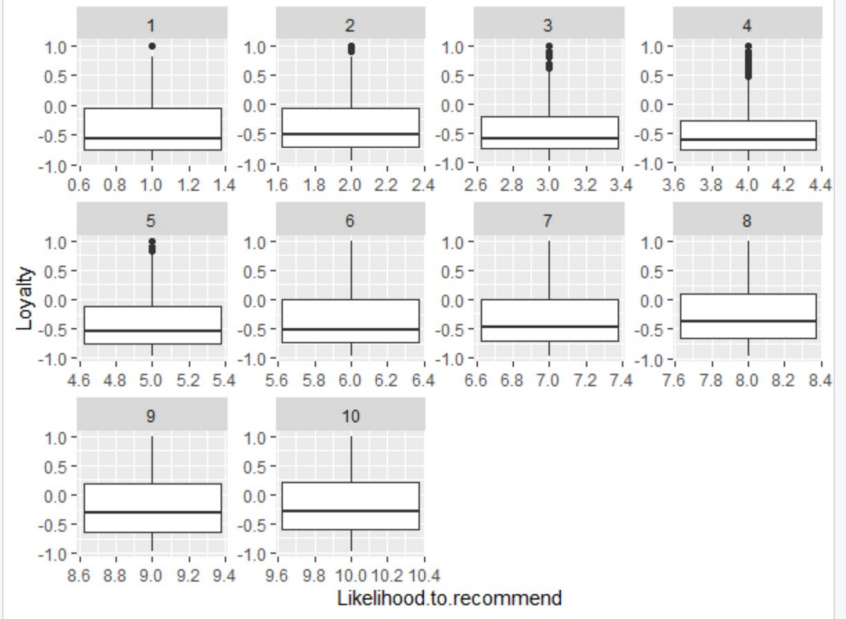
3.1 Visualization

With the fact that customers were valuing the loyalty program less, relying on their loyalty program was not enough in keeping low customer churn. While “Likelihood to Recommend” is a new indicator which has often been suggested that it is good for understanding how likely a customer is to churn. And as it has mentioned in the overview, according to research, Net Promoter Score is nearly three times more sensitive at predicting customer churn than customer satisfaction. We want to test the relationship between loyalty and “likelihood to recommend” to prove that this factor can also be set as a powerful factor in reflecting customer’s attitude and predicting their future behavior.

Since we want to see the dynamic relation trend between those two factors to ensure the accuracy of their correlation, we create several boxplots to see the distribution within a list of numbers. In the following code, we wanted to use data in the dataframe called df (whose data has been acquired, cleaned, transformed and munged in the last step) and we use the facet_wrap function to generate some facet panels of different range in this distribution and present them in the same page with the order of ascending, which is really easy to notice the difference and changes between different domains. (“free” in scales means that each small graph is free to adjust the coordinate scale range according to its own data range.)

```
# explore the relationship between likelihood and loyalty
ggplot(data = df, aes(x = Likelihood.to.recommend, y = Loyalty)) + geom_boxplot() + facet_wrap(~ Likelihood.to.recommend, scales="free")
```

As we can see from the following screenshot, with the growth of likelihood to recommend, the average of loyalty is increasing. Besides, with the growth of likelihood, volatility of loyalty is relatively greater.

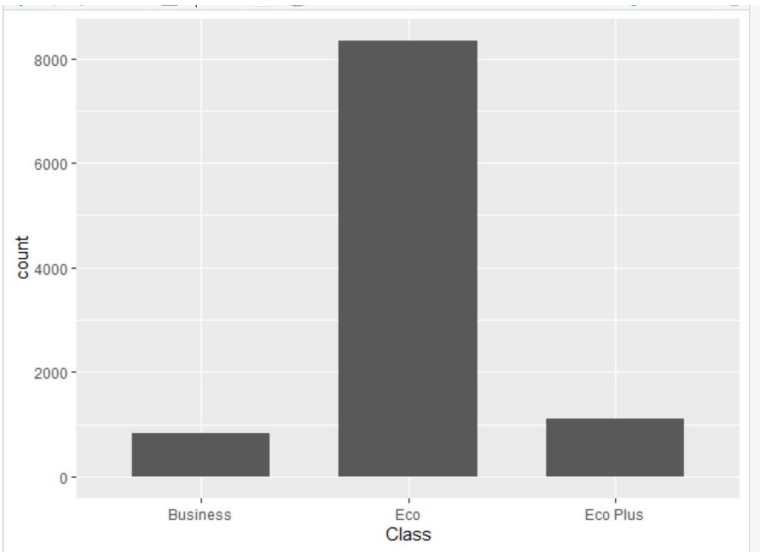


Here we start our **customer analysis**, remember that class, travel type, airline status, year length, age, gender, price sensitivity, flight per year, shopping, eating and drinking amount are what we classified as characters of customers at the beginning, we would try to explore them and get more insights from the customer analysis.

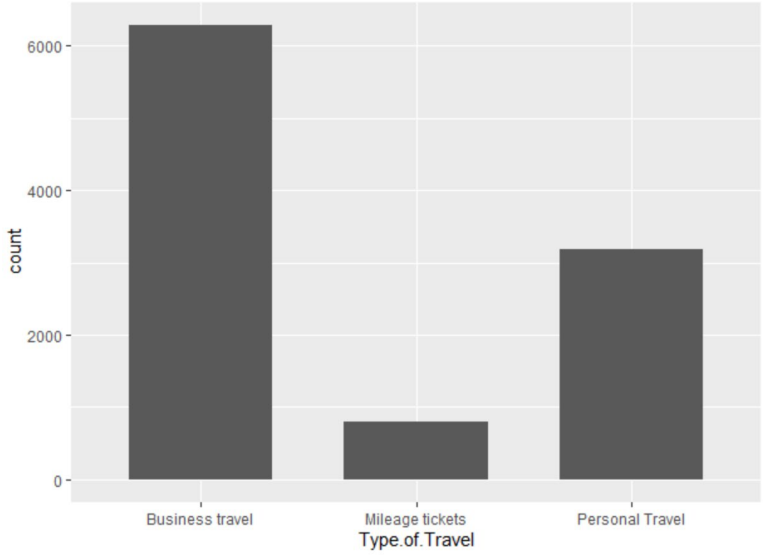
In order to explore class, travel type and status distribution of customers, we used ggplot and geom_bar function and by default, the height of each bar is equal to the number of data in each group that it is clear for us to make comparisons with different classes. To better visualize the data, we tested the ratio and finally set the width of the bar chart at 2/3.

We can see from the graph that the number of people from economic class is more than 8,000 with the total amount of this survey at 10282 people.

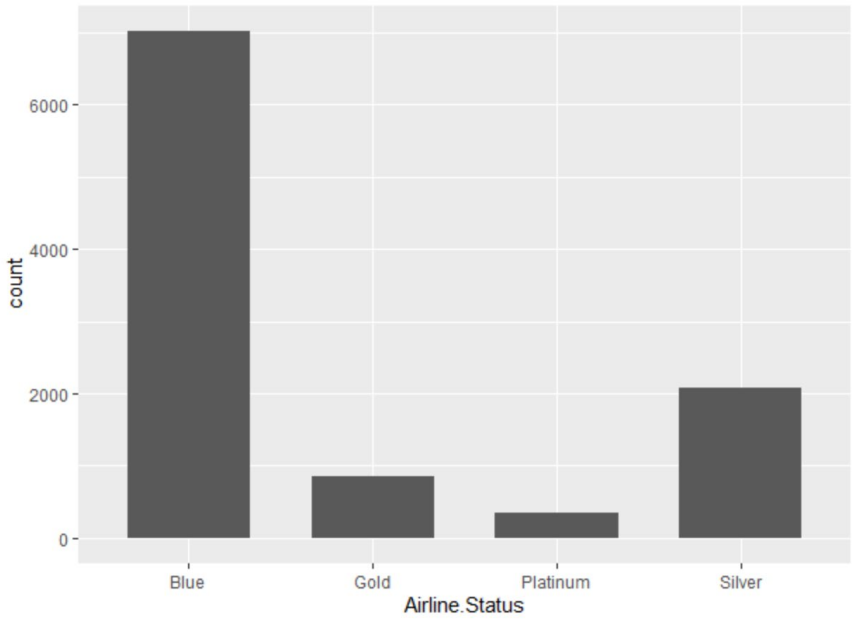
```
# explore class, travel type and status distribution
ggplot(df_customer, aes(x = Class)) + geom_bar(width = 2/3) # most economic
```



We did the same function when it comes to type of travel and Airline Status. From the analysis of type of travel, we found that business travel was the most common purpose among customers at above 6,000 people, personal travel ranked second at about 3,000 people and mileage tickets were the least.



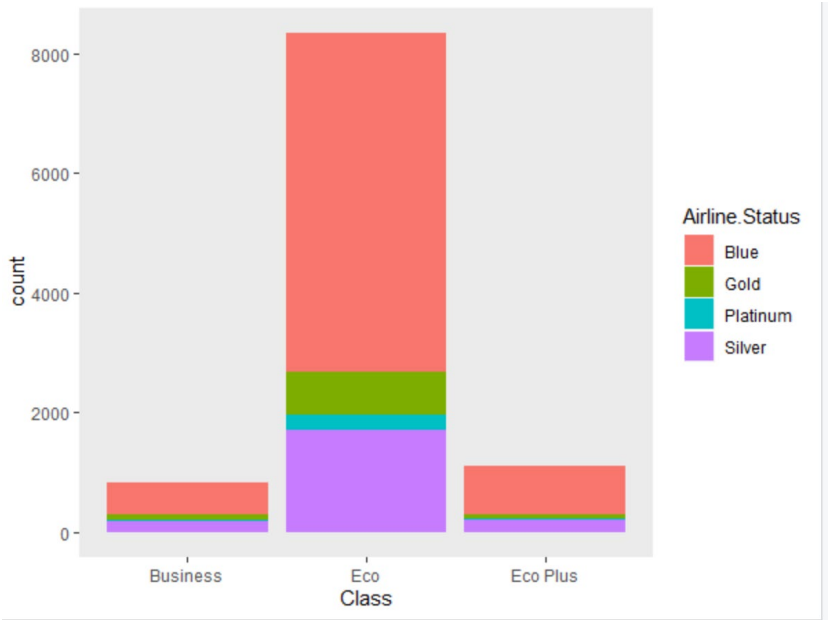
From the analysis of airline status, the difference between blue and other status was so great that it had the most customers in their survey at about 7,000, followed by silver status at about 2,000 people. Gold and platinum was at a small number.



In order to take account of class and airline status in the same time, we used the following code. To remove the default background and grid of ggplot (since there would be several indicators and colors in one graph, it's better to remove the default grid to make the outcome clear), we used the theme function and command of setting panel.border, panel.grid.major and panel.grid.minor as blank. We used data from the dataframe we have created called df_customer, and we used class as the x-axis. Then filled the bar with the number of different airline status in different classes.

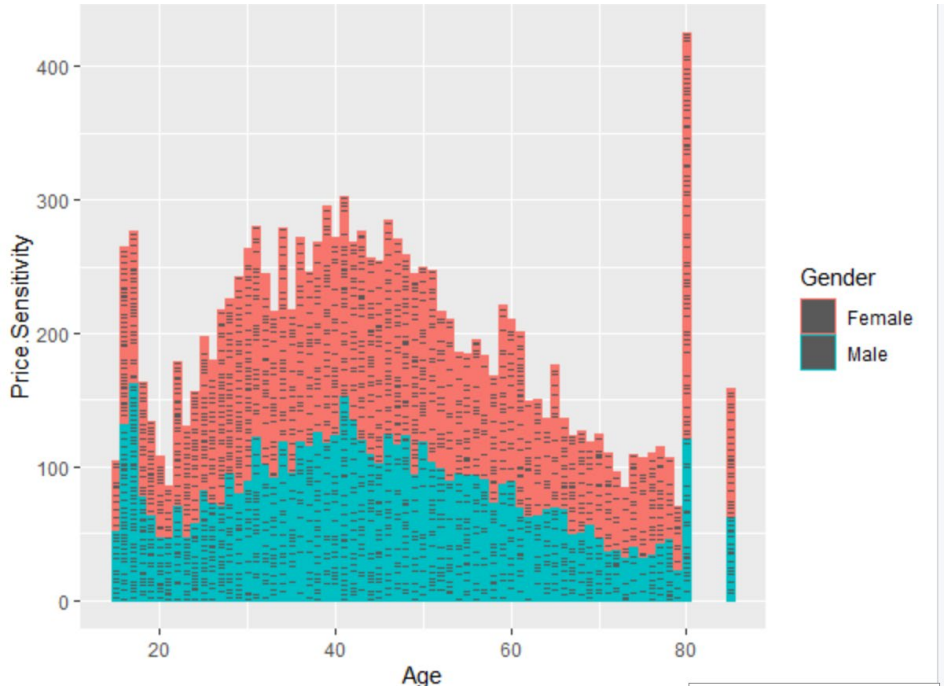
```
ggplot(df_customer, aes(x = Class)) +
  geom_bar(aes(fill = Airline.Status)) +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

From the following graph, we can see that blue airline have a greater share in all 3 classes compared to other status. The number of people in economic class are most and there is a great number of people of blue status at about 5,000 in this class.

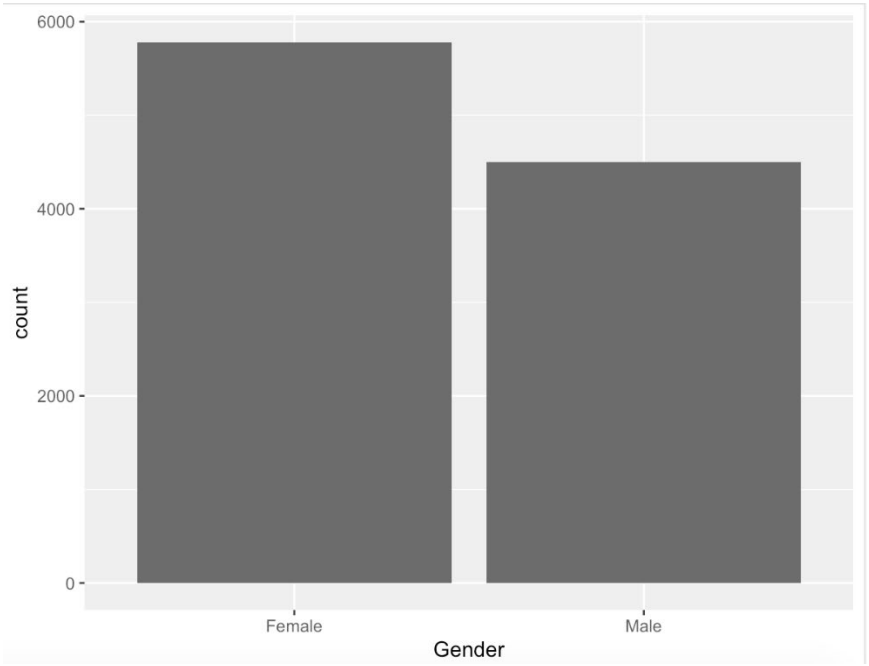


After that, we tried to explore the relationship among year length, age, gender and price sensitivity. We generated a ggplot and used geom_col function to represent the number of male or female in corresponding age with the height of its bars.

```
# explore year length, age, gendar and price sensitive
ggplot(df_customer, aes(x = Age, y = Price.Sensitivity)) +
  geom_col(aes(col = Gender))
```

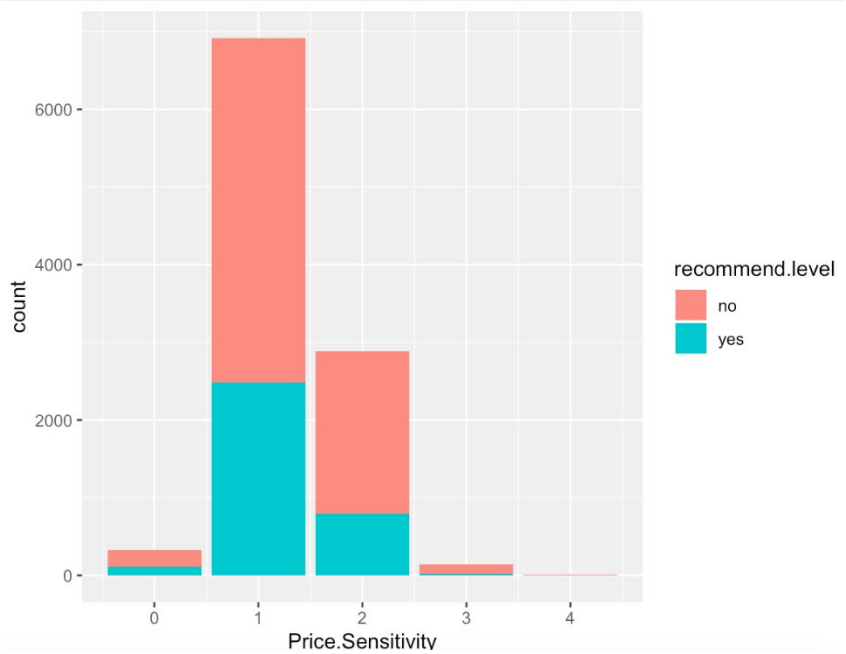


From the ggplot, we got to know that female is more sensitive to price than male. Besides, people above 80 years old and females aged 30 to 60 are more sensitive to the price.



To prove the importance of price sensitivity to make this test more meaningful, we decided to talk about the relationship between price sensitivity and likelihood to recommend. Since we have created our own indicator called recommend level that it turns to be “yes” when the value of likelihood to recommend is greater than 8, otherwise it will turn out to be “no”. We filled the bar with the recommended level to see the relation between price sensitivity and whether customers would recommend. In the value of price sensitivity at 1 and 2, the recommend level would be strongly influenced compared to other extent of price sensitivity. In the price sensitivity at 1, the number of people who are not likely to recommend (around 4,000 people) is greater than those who want to recommend.

```
# explore price sensitivity and loyalty or likelihood
df_customer %>%
  ggplot() +
  aes(x = Price.Sensitivity) +
  geom_bar(aes(fill = recommend.level))
```

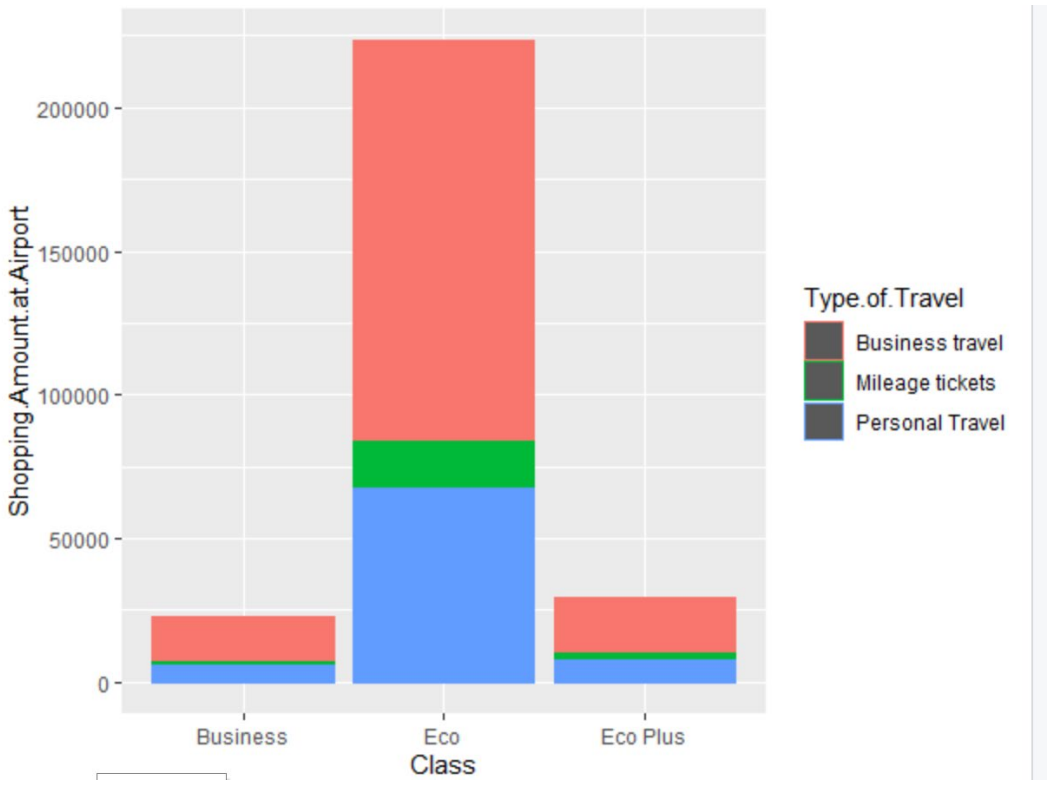


Since the number of females is larger than that of males and price sensitivity is related to the recommended level, the company shouldn't ignore the price changes.

Next, we will explore shopping amount, eating & drinking amount, class and travel type in characters of customers. Firstly, we assign the data Departure.Delay.in.Minutes in df to a column called df_customer$Departure.Delay.in.Minutes in our customer dataframe.

```
df_customer$Departure.Delay.in.Minutes <- df$Departure.Delay.in.Minutes
```

Firstly, we want to explore the relationship between the two characters of customers (type of travel and class) and shopping amount at airport. we created a ggplot and used class as x-axis, used different type of travel to represent different color.



Next, we created a geom_point graph to incorporate three characters of customers (class, type of travel, airline status) and showed the relationship of these characters with two dependent factors (those are shopping, eating and drinking at the airport). To make the outcome more powerful and make the analysis more efficient, we select shopping amount and eating and drinking amount below 200. To make the difference of those points more evident, in the aes code, we represent Type.of.Travel in different color, Class in different shape.

```
ggplot(df_customer[df_customer$Shopping.Amount.at.Airport<200 & df_customer$Eating.and.Drinking.at.Airport<200,],
       aes(x = Eating.and.Drinking.at.Airport, y = Shopping.Amount.at.Airport)) +
  geom_point(cex = 2, aes(alpha = Airline.Status, color = Type.of.Travel, shape = Class))
```

From the above diagram, no matter whether people shop or eat at the airport, the number of customers in economy class who travel for business or personally counts most. And blue customers usually spend more money at airports. Southeast could start with this point to set some bonus to attract customers.

4. USE OF MODELING TECHNIQUES & VISUALIZATION

4.1 Partner Analysis

4.1.1 Map Visualization

In the partner analysis, firstly, we want the name of all our partners and we used unique function to prevent the name from repeat.

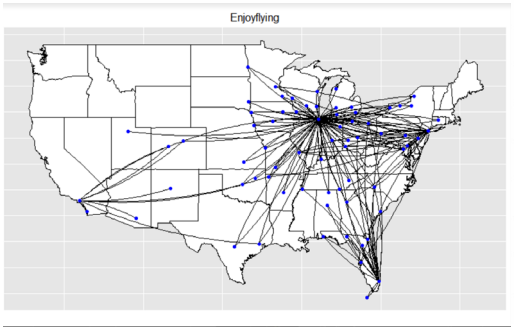
	V1
1	Cheapseats Airlines Inc.
2	GoingNorth Airlines Inc.
3	FlyToSun Airlines Inc.
4	Paul Smith Airlines Inc.
5	OnlyJets Airlines Inc.
6	Sigma Airlines Inc.
7	Northwest Business Airlines Inc.
8	FlyFast Airways Inc.
9	Oursin Airlines Inc.
10	Southeast Airlines Co.
11	FlyHere Airways
12	EnjoyFlying Air Services
13	Cool&Young Airlines Inc.
14	West Airways Inc.

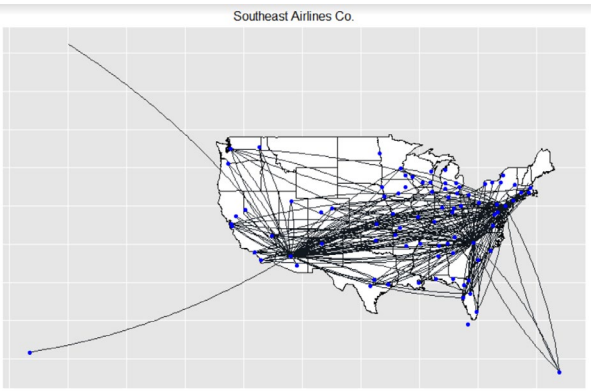
Firstly, we got the map of the united states by creating a ggplot called MAP to get the general information about the geography and the business scope the airline reached. Secondly, we used subset function to select data or related columns that meet a certain criteria from a data frame. We used it to select the partner name and firstly, we tried to explore the flights from EnjoyFlying Air Services.

geom_curve draws a curved line with the data of longitude and latitude of the place, then draws points of these destinations and original cities, at last using theme function to improve the graph visualization.

```
EFmap <- MAP +
  geom_curve(data=EF,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="black",
             size=0.70,
             curvature=0.1) +
  geom_point(data=EF,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
             ) +

  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12))+
  ggtitle("Enjoyflying")
```

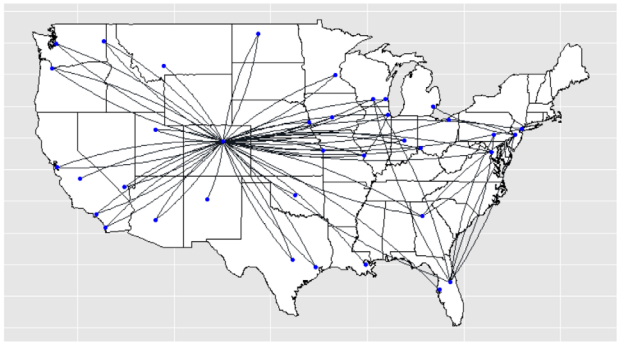
We got the graphs from other partners using the same method like the following:



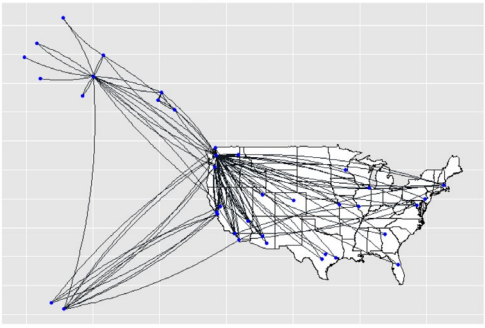


Cheapseats Airlines Inc.

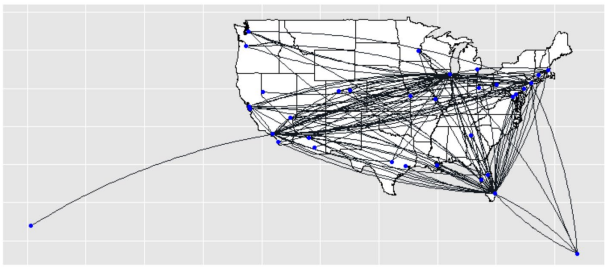
GoingNorth Airlines Inc.



FlyToSun Airlines Inc.

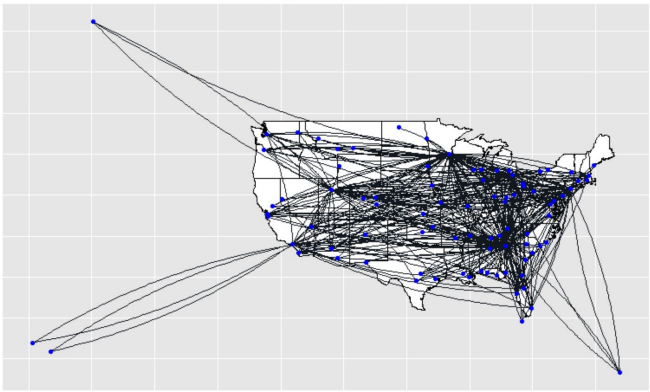


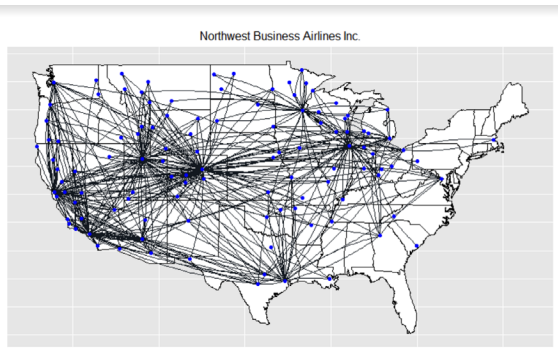
Paul Smith Airlines Inc

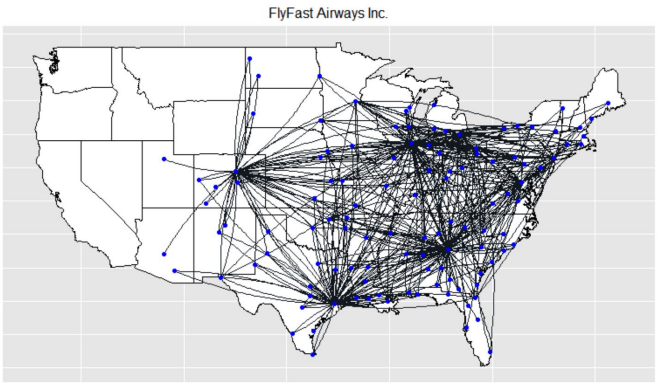


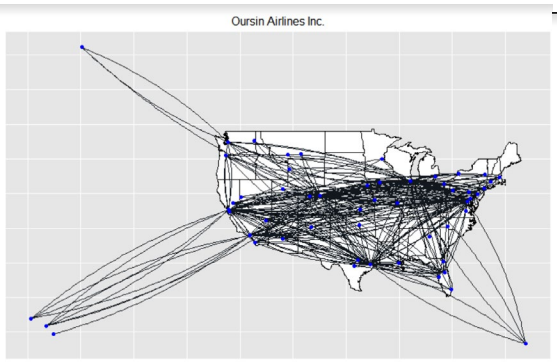
OnlyJets Airlines Inc.

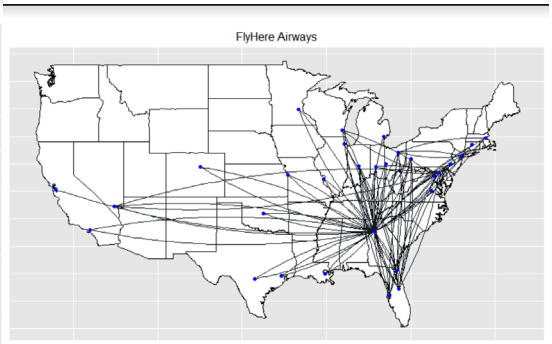
Sigma Airlines Inc.

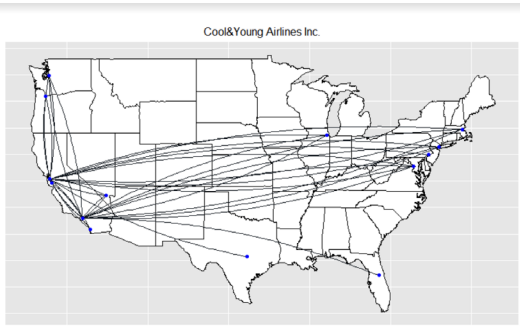


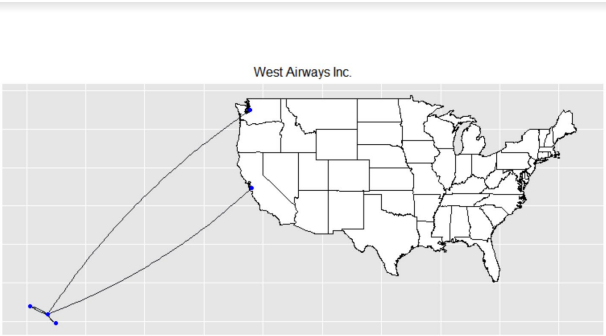












4.1.2 Result Analysis

From the analysis of other partners, we find that the lines and points are quite intense in the Cheapseats Airlines Inc. The flights are intense and there are a great number of cities that are evolved in those flights, which to some extent shows that the cooperation with Cheapseats is close and it matches customers' needs since those flights are being taken more by customers compared with that with other partners.

We can also get the number of customers for different purposes by the following code.

```
stat11 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Cheapseats Airlines Inc.")
One1<-nrow(stat11)
```

	Names	Business.Travel	Personal.Travel	Mileage.Tickets
1	Cheapseats Airlines Inc.	1336	652	145
2	GoingNorth Airlines Inc.	106	46	12
3	FlyToSun Airlines Inc.	193	93	27
4	Paul Smith Airlines Inc.	317	158	41
5	OnlyJets Airlines Inc.	225	116	25
6	Sigma Airlines Inc.	972	505	119
7	Northwest Business Airlines Inc.	817	374	101
8	FlyFast Airways Inc.	682	386	93
9	Oursin Airlines Inc.	575	310	88
10	Southeast Airlines Co.	550	279	68
11	FlyHere Airways	150	75	28
12	EnjoyFlying Air Services	287	151	44
13	Cool&Young Airlines Inc.	69	45	10
14	Cool&Young Airlines Inc.	9	1	2

From the outcome we can see that Cheapseats Airlines Inc. has most customers and the purpose of travel for business and personal accounts for a large part.

4.2 Customer Analysis

4.2.1 Linear Model

1) Meaning of Linear Model

In many practical problems of life and work, there may be more than one factor affecting the dependent variable, such as the conclusion that the higher the knowledge level, the higher the income level. To interpret these laws is complex and multi-dimensional. Multiple regression analysis methods are more suitable for interpreting the laws of life.

According to Julian, Linear models seem quite strict, but because predictors can be transformed and combined in any way, they are actually very flexible.[1] The word "linear" left casual observers with the impression that linear models can only handle small simple

data sets. However, this is a far cry from a true linear model that can be easily extended and modified to handle complex data sets. In fact, linear models can be curved rather than just straight lines.

2) Data for Linear Models

Since we set "Loyalty" and "Likelihood to recommend" to measure a customer's value, at the beginning, we built two models with variables related to customers. The data set we used was chosen from the basic data set and was named as "df_lm". The structure of the data set is displayed below.

```
> str(df_lm)
'data.frame':   10282 obs. of  13 variables:
 $ Airline.Status                : chr  "Silver" "Blue" "Blue" "Blue" ...
 $ Age                           : int  34 55 77 61 33 17 19 26 34 58 ...
 $ Gender                        : chr  "Female" "Male" "Female" "Male" ...
 $ Price.Sensitivity             : int  1 1 1 1 1 1 1 2 1 0 ...
 $ Flights.Per.Year              : int  7 17 21 33 16 4 43 19 33 4 ...
 $ Type.of.Travel                : chr  "Business travel" "Business travel" "Personal Trav
el" "Personal Travel" ...
 $ Shopping.Amount.at.Airport    : int  0 0 70 60 10 0 0 0 0 0 ...
 $ Eating.and.Drinking.at.Airport: int  50 120 30 120 20 80 120 0 95 345 ...
 $ Class                         : chr  "Eco" "Eco" "Eco" "Eco" ...
 $ Year.Lenghth                  : int  3 5 11 9 7 8 2 11 11 9 ...
 $ Loyalty                       : num  0.333 -0.7 -0.556 -0.692 -0.143 ...
 $ Likelihood.to.recommend       : int  7 10 7 4 6 8 10 6 8 8 ...
 $ recommend.level               : chr  "no" "yes" "no" "no" ...
```

Then we used these two variables "Loyalty" and "Likelihood to recommend" as dependent variables, while the remaining variables as independent variables to build a linear model. For each model, there were four plots showing model accuracy: Residuals vs Fitted, Normal QQ, Scale-Location, and Residuals vs Leverage. Followed is the meaning of each plot[2]:

- Residuals vs Fitted: The residual should be a normal distribution, independent of the estimates. If the residual is still related to the estimated value, it means that the model still has room for improvement, and the accuracy of such a model is greatly reduced.
- Normal QQ: Normal QQ-plot is used to detect whether its residuals are normally distributed. If a residual is normally distributed, the points in the graph will appear as a straight line.

- Scale-Location: This plot is used to check the equal variance assumption. If the variance is not a fixed value, then the reliability of this model is greatly reduced.
- Residuals vs Leverage: The significance of this chart is to check if there are any extreme points in the data analysis project. In the linear model, the Cook distance is used to analyze whether a point is very "influential." Generally speaking, points with a distance greater than 0.5 need attention.

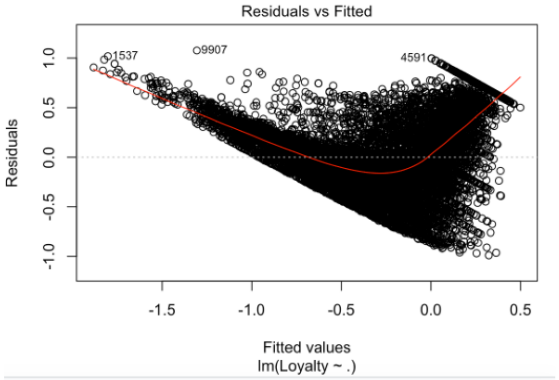
3) Results for Linear Model

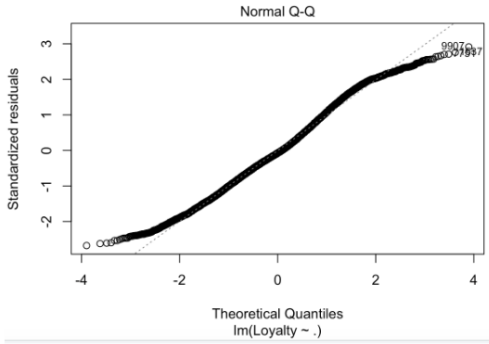
We built two linear models and regarded “Loyalty” and “Likelihood to recommend” as dependent variables. Followed are the results of each model. “lmodel0” is the model for “Loyalty” and “lmodel1” is the model for “Likelihood to recommend”. The R-squared value of lmodel0 is 0.5258, while the R-squared value of lmodel1 is 0.3554. According to the textbook, when analyzing some notoriously unpredictable behavior such as human behavior, an r-squared of .20 or .30 may be very good.[3] Based on that, those independent variables are strongly related to the dependent variables. From the result, the most related variables to Loyalty are Airline Status, Age, Gender, Price Sensitive, Flights per year, type of travel, shopping amount at airports, eating and drinking at airports. From the picture, the most related variables to likelihood are Airline Status, Age, Gender, Price Sensitive, Flights per year, type of travel, shopping amount at airports, eating and drinking at airports.

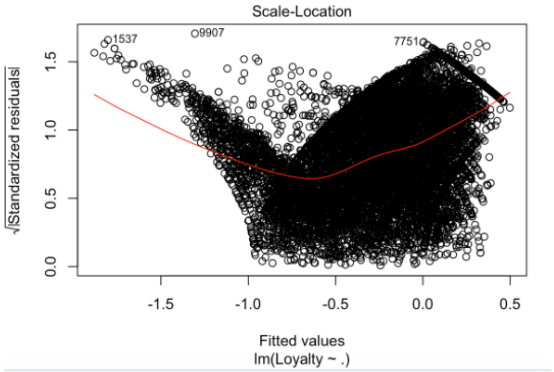
```
> summary(lmodel0)

Call:
lm(formula = Loyalty ~ ., data = df_lm[, c(-12, -13)])

Residuals:
     Min       1Q   Median       3Q      Max
-0.98878 -0.25018 -0.02695  0.24603  1.07716

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    5.479e-01  2.193e-02  24.980  < 2e-16 ***
Airline.StatusGold             4.422e-02  1.358e-02   3.256 0.001133 **
Airline.StatusPlatinum         7.896e-02  2.083e-02   3.791 0.000151 ***
Airline.StatusSilver           4.881e-02  9.413e-03   5.185 2.20e-07 ***
Age                           -4.070e-03  2.326e-04 -17.496  < 2e-16 ***
GenderMale                    -3.534e-02  7.594e-03  -4.654 3.29e-06 ***
Price.Sensitivity             -7.457e-02  6.777e-03 -11.003  < 2e-16 ***
Flights.Per.Year              -2.601e-02  2.804e-04 -92.739  < 2e-16 ***
Type.of.TravelMileage tickets  1.781e-02  1.399e-02   1.273 0.203080
Type.of.TravelPersonal Travel  6.334e-02  9.073e-03   6.981 3.12e-12 ***
Shopping.Amount.at.Airport    -3.221e-04  6.895e-05  -4.671 3.03e-06 ***
Eating.and.Drinking.at.Airport -3.313e-04  6.975e-05  -4.750 2.06e-06 ***
ClassEco                       4.202e-03  1.347e-02   0.312 0.755077
ClassEco Plus                 -5.410e-02  1.710e-02  -3.164 0.001563 **
Year.Lenghth                  -8.029e-04  1.229e-03  -0.653 0.513522
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3698 on 10267 degrees of freedom
Multiple R-squared:  0.5258,    Adjusted R-squared:  0.5252
F-statistic: 813.3 on 14 and 10267 DF,  p-value: < 2.2e-16
```

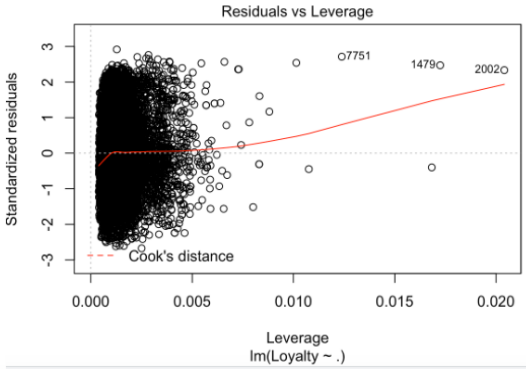
```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    8.0319831  0.1095887  73.292  < 2e-16 ***
Airline.StatusGold             0.7090117  0.0678536  10.449  < 2e-16 ***
Airline.StatusPlatinum         0.1267046  0.1040630   1.218 0.223413
Airline.StatusSilver           1.3659807  0.0470341  29.042  < 2e-16 ***
Age                           -0.0062102  0.0011624  -5.343 9.35e-08 ***
GenderMale                     0.1438563  0.0379425   3.791 0.000151 ***
Price.Sensitivity             -0.1276720  0.0338628  -3.770 0.000164 ***
Flights.Per.Year              -0.0076932  0.0014012  -5.491 4.10e-08 ***
Type.of.TravelMileage tickets -0.1541829  0.0699248  -2.205 0.027478 *
Type.of.TravelPersonal Travel -2.3381963  0.0453365 -51.574  < 2e-16 ***
Shopping.Amount.at.Airport     0.0014354  0.0003445   4.166 3.12e-05 ***
Eating.and.Drinking.at.Airport 0.0024944  0.0003485   7.157 8.80e-13 ***
ClassEco                      -0.2106511  0.0672983  -3.130 0.001752 **
ClassEco Plus                 -0.1853487  0.0854382  -2.169 0.030076 *
Year.Lenghth                  -0.0049768  0.0061404  -0.810 0.417678
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.848 on 10267 degrees of freedom
Multiple R-squared:  0.3554,    Adjusted R-squared:  0.3545
F-statistic: 404.3 on 14 and 10267 DF,  p-value: < 2.2e-16
```

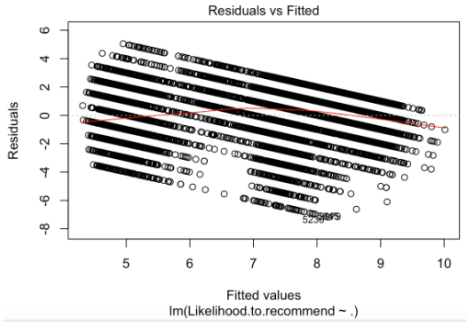
After the obtained regression model is tested for significance, it is also necessary to perform residual analysis, which shows the difference between the predicted value and the actual value, to test the correctness of the model. The residual must follow a normal distribution. Followed are pictures for residual analysis. For human activities, the residuals basically show a trend of normal distribution for “Likelihood to recommend”. However, the first picture for “Loyalty” shows that the residuals are still related to the estimated value, which means that the model still has something to improve. Thus, the accuracy of the model is not good enough.

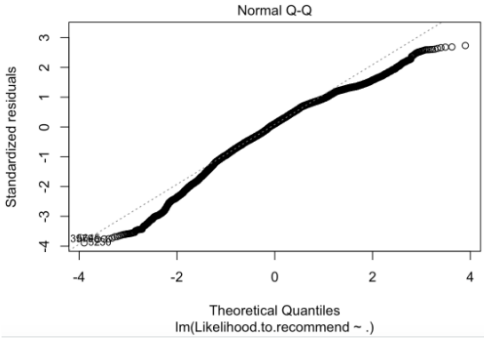

Residuals vs Fitted
Residuals
Fitted values
lm(Loyalty ~ .)
1537
9907
4591

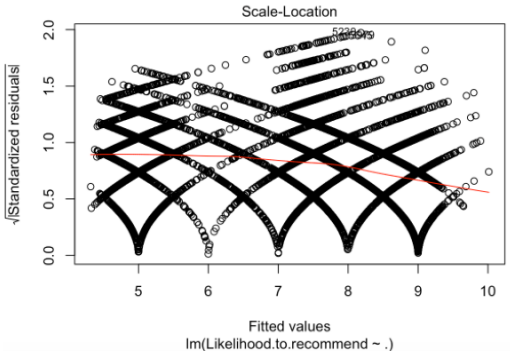

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Loyalty ~ .)

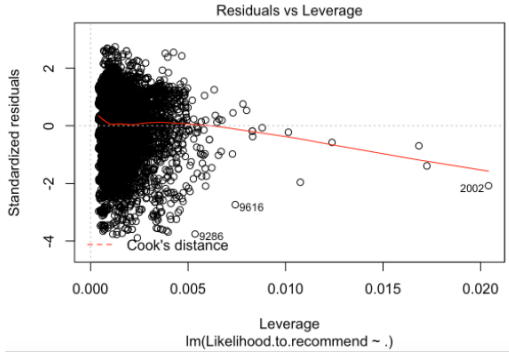

Scale-Location
√|Standardized residuals|
Fitted values
lm(Loyalty ~ .)
1537
9907
7751


Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(Loyalty ~ .)
7751
1479
2002


Residuals vs Fitted
Residuals
Fitted values
lm(Likelihood.to.recommend ~ .)

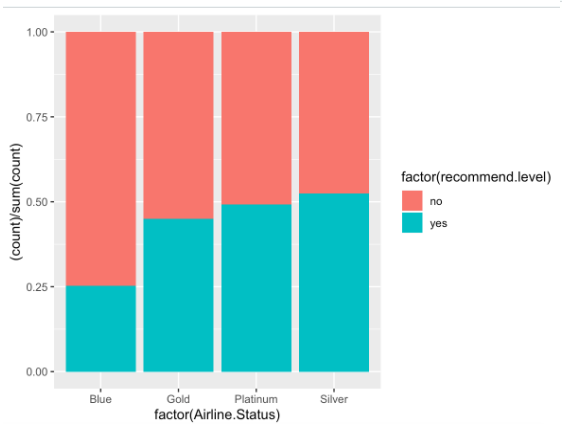

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Likelihood.to.recommend ~ .)

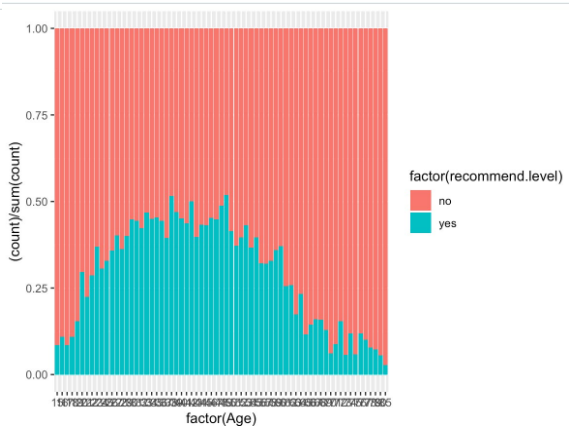

Scale-Location
√|Standardized residuals|
Fitted values
lm(Likelihood.to.recommend ~ .)

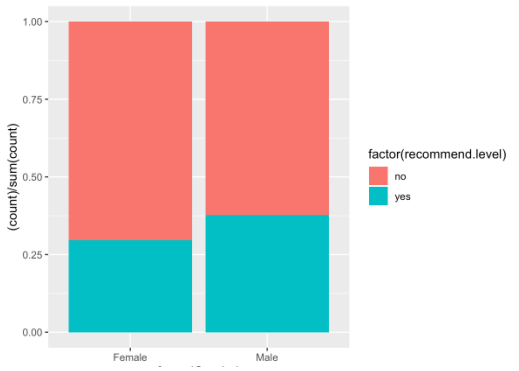

Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(Likelihood.to.recommend ~ .)
2002
9616
9286

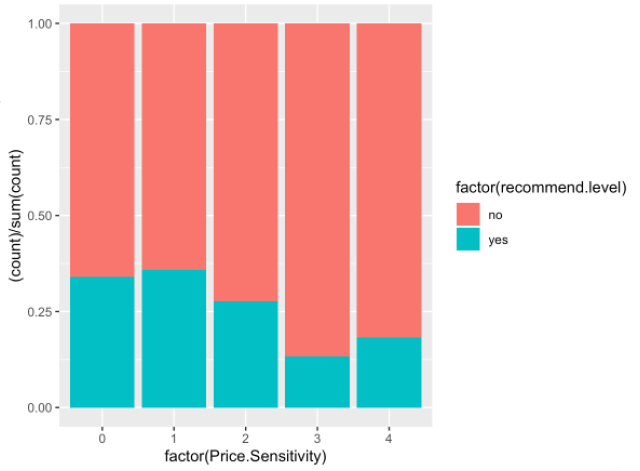
4) Analysis of Linear Model

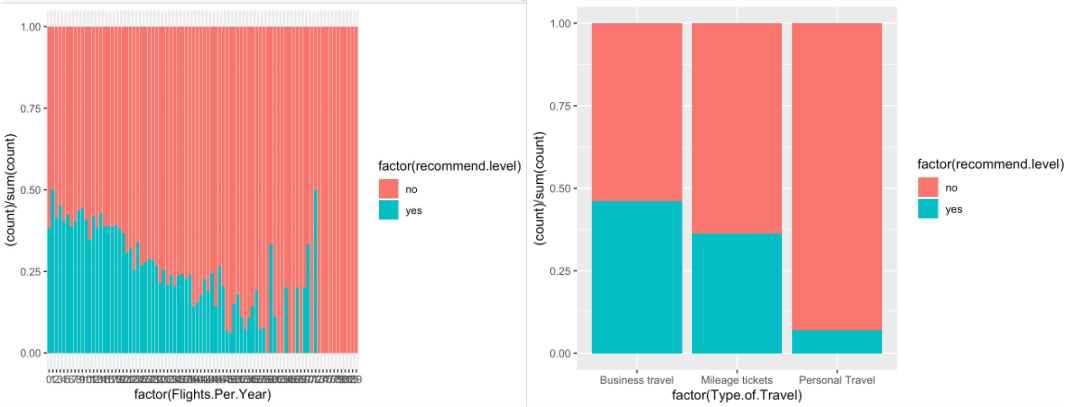
According to the results, the model of the likelihood is reliable, while the relationship to loyalty is not correct enough. Thus, we'll discuss the factors that influence the likelihood to recommend. From the summary of lmodel1, there are six factors that are strongly related to the likelihood, including Airline Status, Age, Gender, Price Sensitive, Flights per year, type of travel, shopping amount at airports, eating and drinking at airports. Southeast Airlines is supposed to pay attention to improve these services or add some coupons to retain customers based on discovery. To better describe how these six factors affect the likelihood to recommend, bar charts are displayed in the following.











According to the picture, we can find that blue customers are less likely to recommend while almost half of the silver customers would like to recommend the airline. And the preference for recommendation related to age presents in a curve, in which we can find that customers aged between 30 to 60. Besides, customers who travel for business are more likely to recommend the airline while the number of personal-traveling customers who would like to recommend the airline counts far fewer than that of those who won't recommend airlines.

4.2.2 SVM (Support Vector Machine)

1) Meaning of SVM

SVM tries to create a "hyperplane" to divide the data. It's a linear model for classification and regression problems, which can solve linear and non-linear problems and work well for many practical problems. SVM tries to make a decision boundary in a way that the separation between the two classes is as wide as possible.

2) Data for SVM

Since SVM is a dichotomy, we decided to use the "recommend level" as the variable that our model will predict to judge whether a customer would or would not recommend the airline. We added a new variable named "Old" which divided the "Age" into three parts: young, adult, old. We built two models to predict "Loyalty" as well as "recommend level". Since "Loyalty" is a continuous variable, we divided it into two parts, loyal or not loyal, based on whether the value of "Loyalty" was above 0.5. Then we selected these variables to build the SVM model: Airline Status, Age, Gender, Price Sensitivity, Flights Per Year, Type of Travel, Shopping Amount at Airport, Eating and Drinking at Airport, Class and Year Length. Since the total number of the data is large, we divided the data set into training and testing data to avoid overfitting. The code was pasted below:

```
################################## SVM to predict loyalty ###################################
library(kernlab)

df_svm <- df_customer[,c(-12,-14)]
str(df_svm)
df_svm$Loyalty[df_svm$Loyalty>=0.5] <- "loyal"
df_svm$Loyalty[df_svm$Loyalty != "loyal"] <- "not loyal"

df_svm$randIndex <- sample(1:dim(df_svm)[1])

cutPoint2_3 <- floor(2 * dim(df_svm)[1]/3)
trainData <- df_svm[df_svm$randIndex[1:cutPoint2_3],]
testData <- df_svm[df_svm$randIndex[(cutPoint2_3+1):dim(df_svm)[1]],]

library(caret)
svmmodel1 <- ksvm(Loyalty ~ ., data = trainData[,-12], kernel = "rbfdot", kpar = "automatic",
                  C = 5, cross = 3, prob.model = TRUE)
svmPred1 <- predict(svmmodel1, testData)

svmPred1 <- as.factor(svmPred1)
testData$Loyalty <- as.factor(testData$Loyalty)
install.packages("e1071")
library(e1071)
confusionMatrix(svmPred1,testData$Loyalty)

svmmodel2 <- ksvm(recommend.level ~ ., data = trainData[,-11], kernel = "rbfdot", kpar = "automatic",
                  C = 5, cross = 3, prob.model = TRUE)
svmPred2 <- predict(svmmodel2, testData)

svmPred2 <- as.factor(svmPred2)
testData$recommend.level <- as.factor(testData$recommend.level)
confusionMatrix(svmPred2,testData$recommend.level)
```

3) Results for SVM Model

The results of the SVM model that we used to predict loyalty and the likelihood of recommendation are displayed below.

```
> svmmodel1
Support Vector Machine object of class "ksvm"

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

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0869072710748042

Number of Support Vectors : 1175

Objective Function Value : -4368.351
Training error : 0.042165
Cross validation error : 0.066968
Probability model included.
```

```
> svmmodel2
Support Vector Machine object of class "ksvm"

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

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0890432541047363

Number of Support Vectors : 3944

Objective Function Value : -17093.64
Training error : 0.204406
Cross validation error : 0.273563
Probability model included.
```

Then we predicted the testing data based on the model and used the confusion matrix to judge the result of the prediction. As for the model of loyalty, the accuracy of the model is 0.9431, which means that the model works very well in prediction. Besides, the sensitivity and specificity are high, which means that the model is reliable when predicting. Thus, Southeast Airlines could use the model to predict the loyalty of a customer. As for the model

of the likelihood, the accuracy is 0.7415, while the sensitivity and specificity are considerable. So, using the model to predict the likelihood of recommendation is also a good way for the company to keep customers.

```
Confusion Matrix and Statistics

          Reference
Prediction  loyal not loyal
  loyal       272        77
  not loyal   118      2961

               Accuracy : 0.9431
                 95% CI : (0.9348, 0.9506)
    No Information Rate : 0.8862
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7044

 Mcnemar's Test P-Value : 0.004177

            Sensitivity : 0.69744
            Specificity : 0.97465
         Pos Pred Value : 0.77937
         Neg Pred Value : 0.96168
             Prevalence : 0.11377
         Detection Rate : 0.07935
   Detection Prevalence : 0.10181
      Balanced Accuracy : 0.83605

       'Positive' Class : loyal
```

```
> confusionMatrix(svmPred2,testData$recommend.level, positive = "yes")
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1857  455
       yes  431  685

               Accuracy : 0.7415
                 95% CI : (0.7265, 0.7561)
    No Information Rate : 0.6674
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.4147

 Mcnemar's Test P-Value : 0.4397

            Sensitivity : 0.6009
            Specificity : 0.8116
         Pos Pred Value : 0.6138
         Neg Pred Value : 0.8032
             Prevalence : 0.3326
         Detection Rate : 0.1998
   Detection Prevalence : 0.3256
      Balanced Accuracy : 0.7063

       'Positive' Class : yes
```

4.3 Flight Analysis

4.3.1 Association Rule

We decided to use Association Rule as one of our methods to discover interesting relations between the factors in the airline data set. This model intended to identify strong rules discovered in the data set using some measure of interestingness. It also generates new patterns as it analyzes more data.

```
df_customer$Old <- "adult"
df_customer$Old[which(df_customer$Age <= 21)] <- "young"
df_customer$Old[which(df_customer$Age >= 60)] <- "old"

# find out the consumption relationship
df_rule_con <- df_customer[,c(1,3,5,6,7,8,9,10,15)]

df_rule_con$shopping[df_rule_con$Shopping.Amount.at.Airport != 0] <- "yes"
df_rule_con$shopping[df_rule_con$Shopping.Amount.at.Airport == 0] <- "no"

df_rule_con$eating[df_rule_con$Eating.and.Drinking.at.Airport != 0] <- "yes"
df_rule_con$eating[df_rule_con$Eating.and.Drinking.at.Airport == 0] <- "no"

df_rule_con$Flights.Freq <- "med"
df_rule_con$Flights.Freq[which(df_rule_con$Flights.Per.Year <= 10)] <- "low"
df_rule_con$Flights.Freq[which(df_rule_con$Flights.Per.Year >= 30)] <- "high"

df_rule_con$Year.length[which(df_rule_con$Year.Lenghth < 6)] <- "new"
df_rule_con$Year.length[which(df_rule_con$Year.Lenghth >= 6)] <- "regular"

df_rule_con <- df_rule_con[,c(-3,-5,-6,-8)]

df_rule_con <- as(df_rule_con, "transactions")
ruleset_con1 <- apriori(df_rule_con[,-7], parameter = list(support = 0.005, confidence = 0.5),
                        appearance = list(default="lhs", rhs=("shopping=yes")))
ruleset_con2 <- apriori(df_rule_con[,-6], parameter = list(support = 0.005, confidence = 0.5),
                        appearance = list(default="lhs", rhs=("eating=yes")))
plot(ruleset_con1, jitter = 0)
inspect(ruleset_con1[which.max(quality(ruleset_con1)$lift)])
plot(ruleset_con2, jitter = 0)
inspect(ruleset_con2[which.max(quality(ruleset_con2)$lift)])
```

In the analysis, we divided age into three different groups, those under the age of 21 as ‘young’, those beyond the age of 60 as ‘old’, and those in between as ‘adult’. We also divided the shopping and eating into two categories, which are ‘yes’ and ‘no’. Flight frequency is separated into three groups. Those who took less than 10 flights a year are classified into the ‘low’ group. The ‘high’ group includes those who took more than 30 flights a year, while those sit in the middle are in the ‘med’ group. The year length stands for how many years since the customer first took the flight and we also choose the number 6 as a dividing point according to a diagram we get from the data. Those whose number is below 6 are classified as ‘new’, and the others are classified as ‘regular’.

```
    lhs                                rhs               support confidence     lift count
[1] {Airline.Status=Silver,
     Gender=Female,
     Type.of.Travel=Personal Travel,
     Class=Eco,
     Old=adult,
     eating=yes,
     Year.length=regular}           => {shopping=yes} 0.006418985     0.6875 1.602193    66
```

```
    lhs                                   rhs          support     confidence lift     count
[1] {Airline.Status=Gold,Class=Eco Plus} => {eating=yes} 0.007391558 1          1.048114 76
```

When using association rules to do the analysis, we can get a result like this. From the result, it showed that female adult with silver or above airline status, high loyalty who chose economic class in personal travel and have already taken this flight for more than 6 years tend to have meals and shopping at the airport. Thus, we can offer food or shopping discount to those people to improve their loyalty and attract more customers. Also, other factors like reducing the departure or arrival delay can make the customer happier.

```
df_flight <- df
df_flight$Departure.Delay <- df_flight$Departure.Delay.in.Minutes
df_flight$Departure.Delay[which(df_flight$Departure.Delay.in.Minutes > 0)] <- "yes"
df_flight$Departure.Delay[which(df_flight$Departure.Delay.in.Minutes == 0)] <- "no"

df_flight$Arrival.Delay <- df_flight$Arrival.Delay.in.Minutes
df_flight$Arrival.Delay[which(df_flight$Arrival.Delay.in.Minutes > 0)] <- "yes"
df_flight$Arrival.Delay[which(df_flight$Arrival.Delay.in.Minutes == 0)] <- "no"

df_rule <- df_flight[,c(2,22,33,34,35)]
df_rule <- data.frame(df_flight$Origin.City,df_flight$Flight.cancelled,
                      df_flight$Departure.Delay,df_flight$Arrival.Delay,
                      df_flight$recommend.level)
df_rule <- as(df_rule, "transactions")
ruleset <- apriori(df_rule,
                   parameter = list(support = 0.005, confidence = 0.5),
                   appearance = list(default="lhs", rhs=("recommend.level=yes")))
plot(ruleset,jitter = 0)
inspectDT(ruleset)
inspect(ruleset[which.max(quality(ruleset)$lift)])
```

```
    lhs                       rhs                      support confidence     lift count
[1] {Origin.City=Baltimore,
     Flight.cancelled=No,
     Arrival.Delay=no}      => {recommend.level=yes} 0.005835441  0.5940594 1.791237    60
```

This is the result from building the association rules model to analyze which attributes are related to the likelihood of recommendation. We defined those who scored above 8 as promoters according to Net Promoter Score. These people will be classified in the “yes” group of recommend level. From the result, people in the east cities like Baltimore tend to give recommendations when there was no delay during the flight. So we can put more focus on the east cities and help improve the punctuality of the flight.

```
# find out relationship to loyalty
df_rule_l <- df_customer
df_rule_l$Loyalty[df_rule_l$Loyalty>=0.5] <- "loyal"
df_rule_l$Loyalty[df_rule_l$Loyalty != "loyal"] <- "not loyal"

df_rule_l$Price.Sensitivity <- as.factor(df_rule_l$Price.Sensitivity)

ggplot(aes(x=df_rule_l$Flights.Per.Year), data = df_rule_l) + geom_histogram(col="black")
df_rule_l$Flights.Freq <- "med"
df_rule_l$Flights.Freq[which(df_rule_l$Flights.Per.Year <= 10)] <- "low"
df_rule_l$Flights.Freq[which(df_rule_l$Flights.Per.Year >= 30)] <- "high"

ggplot(aes(x=df_rule_l$Year.Lenghth), data = df_rule_l) + geom_histogram(col="black")
df_rule_l$Year.length[which(df_rule_l$Year.Lenghth < 6)] <- "new"
df_rule_l$Year.length[which(df_rule_l$Year.Lenghth >= 6)] <- "regular"

str(df_rule_l$Departure.Delay.in.Minutes)
df_rule_l$delay[which(as.numeric(df_rule_l$Departure.Delay.in.Minutes) == 0)] <- "on time"
df_rule_l$delay[is.na(df_rule_l$delay)] <- "delay"
df_rule_l <- df_rule_l[,c(-2,-5,-7,-8,-10,-12,-13,-14)]

df_rule_l <- as(df_rule_l, "transactions")
ruleset_loyal <- apriori(df_rule_l, parameter = list(support = 0.005, confidence = 0.5),
                         appearance = list(default="lhs", rhs=("Loyalty=loyal")))
inspect(ruleset_loyal)
inspect(ruleset_loyal[which.max(quality(ruleset_loyal)$lift)])
```

We defined number 0.5 as the loyalty score that whether a customer is loyal or not. Additionally, we continued to use the classification in flights per year and year length according to the diagrams we got from the data. Apart from that, we divided departure delay in minutes into two groups which are ‘on time’ and ‘delay’ based on the result that a delay happened or not.

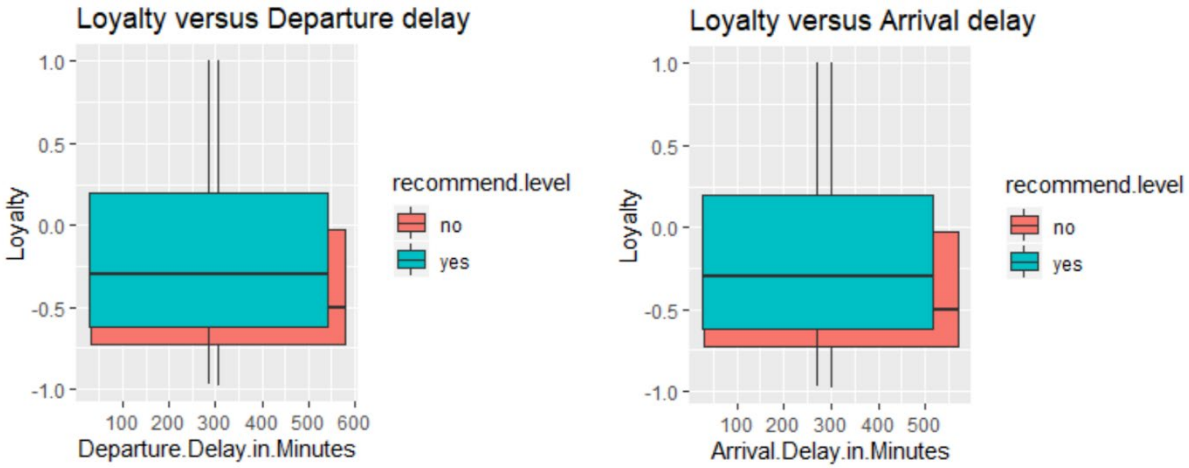
```
    lhs                                  rhs                support confidence     lift count
[1] {Airline.Status=Blue,
     Gender=Female,
     Price.Sensitivity=1,
     Type.of.Travel=Personal Travel,
     Class=Eco,
     Old=adult,
     Flights.Freq=low}              => {Loyalty=loyal} 0.005057382  0.6419753 5.357784    52
> |
```

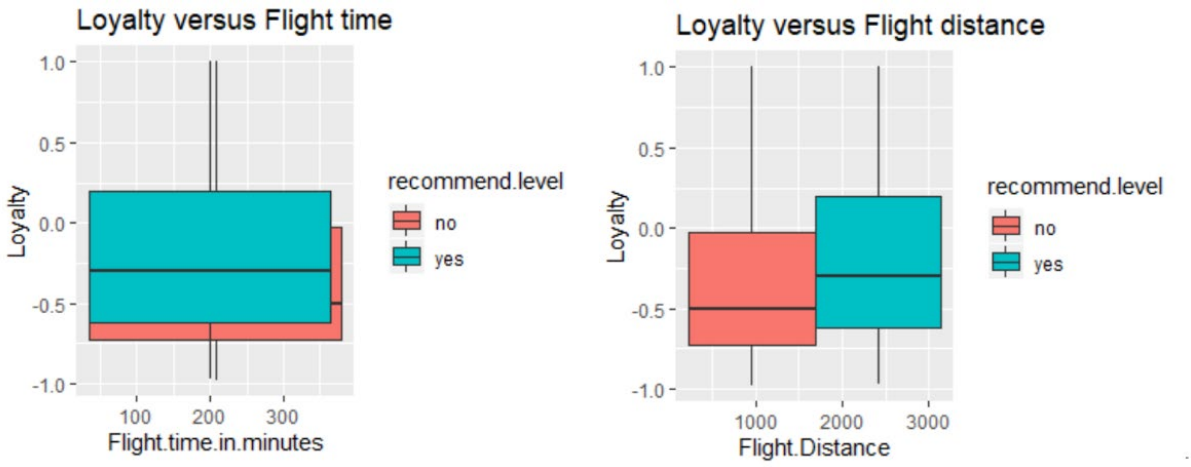
When we use the association rules to find out relations to loyalty, we get a result like this. From the result, we know that there are many factors that have effects on customer loyalty. Female customers who take the blue airline are more likely to stay loyal to the airline company. High price sensitivity customers are more likely to be unhappy, but they also show higher loyalty to one airline company. Besides, people who travel in a low flight frequency and take economic class in their personal travel also show a high tendency in their loyalty. For these kinds of people, we can offer them bones or discount on ticket in order to delight them and get better feedback.

4.3.2 Visualization

By doing data visualization, we take data and put it into a visual context, such as a map or graph. Data visualizations make complex data easier for people to understand, and visualization also makes it easier to find out trends, patterns, and differences in groups of data.

```
# correlation in Departure delay,loyalty and LTR
Plot_Dep.Delay <- ggplot(df,aes(x = Departure.Delay.in.Minutes,y = Loyalty))+
  geom_boxplot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Departure delay')
Plot_Dep.Delay

# correlation in Arrival delay, loyalty and LTR
Plot_Arr.Delay <- ggplot(df,aes(x = Arrival.Delay.in.Minutes, y = Loyalty))+
  geom_boxplot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Arrival delay')
Plot_Arr.Delay

# correlation in Flight time, loyalty and LTR
Plot_F.time <- ggplot(df,aes(x = Flight.time.in.minutes, y = Loyalty))+
  geom_spot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Flight time')
Plot_F.time

# correlation in Flight distance, loyalty and LTR
Plot_F.distance <- ggplot(df,aes(x = Flight.Distance, y = Loyalty))+
  geom_boxplot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Flight distance')
```





These are the boxplots show the correlation among loyalty, recommendation level, and the four flight variables, which are Departure delay, Arrival delay, Flight time, and Flight distance. We can see that when delay happened, no matter it is arrival delay or departure delay, people are not willing to give recommendations and their loyalty tends to go down. Generally, people with higher loyalty are more willing to give recommendations as well. From the third diagram, we can see flight time slightly influence the recommend level or loyalty. When we are looking at the last diagram, it showed that people who take a long-distance flight are more likely to give recommendation compared with those who take a short-distance flight. And the people who take a short-distance flight generally have lower loyalty. So how to improve the customer's experience of short-distance travel should be taken into consideration for airline companies. Offering some good entertainment packages or services to help customers pass their time would be helpful to improve their likelihood of recommendation and loyalty.

4.4 Feedback Analysis-Text Mining

4.4.1 Introduction

After the project data cleaning process, noticed there are certain numbers of free form text field of the passenger comment, with respect to the flight existed. A computer readable nature language transforming process from human readable text is desired to seek further analyzing. Text related analysis could be processed here to locate the major factors which may have certain influence to the customer "Churn". What I'm doing here is trying to pick up those free texts with negative loyalty and low NPS. The reasons for picking up negative comments only here is the content showed in bad comments are most likely something need to be fixed right away to increase customer satisfaction. It is more realistic and authentic for showing the weakness of the

airline company. Problems can be located directly based on these negative comments here. Customer "Churn" could be decrease in certain way by simply solving most "hot" problems after Text Mining analysis.

4.4.2 Package Preparation

```
#Loading the Package
library(NLP)
library(tm)
library(ggplot2)
library(tidyverse)
library(tidyr)
library(RColorBrewer)
library(wordcloud)
```

To start the text mining process, a group of R studio packages are installed and library here. The major package used here is "tm" which grouped by a bunch of functions for the content transform.

4.4.3 Text Preprocessing

1) Creating Corpus

```
#Delete the rows with positive loyalty
df1<-filter(df,df$Loyalty<=0)
#Extracting the free Text
Comment<-df1$freeText
#Cleaning the NA in Text
Comment<-na.omit(Comment)
words.vec <- VectorSource(c(Comment))
# Make a corpus of web data
words.corpus <- Corpus(words.vec)
```

The main structure of the TM package for handling the text is called Corpus which are collections of documents containing natural language. Simply filter out the positive texts only here.

2) Cleaning Corpus

```
#To lower
words.corpus <- tm_map(words.corpus,content_transformer(tolower))
#removePunctuation
words.corpus <- tm_map(words.corpus,removePunctuation)
#removeNumbers
words.corpus <- tm_map(words.corpus, removeNumbers)
#Words no nned to be used
otherwords<-c("now","well","however","always","way","much","take","took","can","poor"
#Taking out stop words
words.corpus <- tm_map(words.corpus, removeWords,c(stopwords("english"),otherwords))
```

Convert the text to lowercase, to eliminate the influence of Capitalization. Stripping any extra white space, removing numbers, removing punctuation and English stop words. “Stop words are basically just common words that were determined to be of little value for certain text analysis, such as sentiment analysis”. For a more precise and accurate analysis, some unrelated words like “Now, well, however, always, way, much, take, took, can, poor, better, even, flying, didn’t, bad, don’t, due, never, flew, airline, flight, southeast, service, good, one, Always, also, just, best, nice, great, first, will, many, ever, fly, get, got, really, like, worst” are manually deleted here. What am I trying to do here is to focus just on nouns since verbs seem confused to our analysis here.

4.4.4 Creating a Document-Term Matrix (DTM)

```
#Create a document-term-matrix
tdm <- TermDocumentMatrix(words.corpus)
#coerce text data back into a plain data matrix
m<-as.matrix(tdm)
wordCounts<-rowSums(m)
wordCounts<-sort(wordCounts,decreasing=TRUE)
```

```
<<TermDocumentMatrix (terms: 1711, documents: 214)>>
Non-/sparse entries: 4129/362025
Sparsity           : 99%
Maximal term length: 20
Weighting          : term frequency (tf)
```

A document-term matrix is a common method for the word’s comparison. The result shows there are 4129 words in transformed TDM form negative loyalty free texts with 99% sparsity. And words counting also finished here for the further analysis.

4.4.5 Visualization

1) Creating Word Cloud

```
#Create a word cloud
cloudFrame<-data.frame(word=names(wordCounts),freq=wordCounts)
wordcloud(cloudFrame$word,cloudFrame$freq,max.words=150, random.order=FALSE, rot.per=0.15)
```

Word cloud is an efficient method for people to easier understand the textual data and to do text analysis. A 150 most frequently used words are plotted here to make word cloud.

2) Creating Plot Bar Chart

```
#Creating new dataframe with 50 most frequent words
CF25<-head(cloudFrame,25)
CF25<-arrange(CF25,desc(freq))
#Plot word frequencies
CF25%>% ggplot(aes(x = reorder(CF25$word,-CF25$freq), y = CF25$freq)) +
                geom_bar(stat = "identity") +
                labs(title = "Top 25 Feature Words based on Negative Loyalty", x = "Word", y= "Frequency") +
                theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

The bar plot is used to present specific words counting with decreasing order.

4.4.6 Analysis

Negative Loyalty

Low NPS

Delayed	**Time**
Luggage	**Food**
Flights	**Seats**
Seats	**Delayed**
Customer	**Luggage**
Time	**Staff**
Hours	**Business**
People	**Flights**
Staff	**Crew**
Experience	**Plane**
Plane	Economy
Food	Comfortable
Hour	**Experience**
Gate	Trip
Terrible	Room

The above word clouds and bar charts clearly shows that “delayed”, “time”, “luggage”, “staff” are some most frequent words customer with negative loyalty and low NPS commonly used in their comments. I also listed the 25 words for both groups customers and bolded the same words which showed twice. The reason for doing that is showing their problems are significantly enough for the customer giving a negative comment and need to be fixed right away.
Based on this analysis, the customer who suffering flight delay, bad luggage and low customer service are more likely to give negative comment. The airline company could focus on improving these three aspects are more likely to improve its customer satisfaction with a lower customer “Churn”.

5. Actionable insights

5.1 From the Partner analysis

Add airlines and Cooperate more with Cheapseats Airlines Inc. Since from the survey and the ggmap, there are intense flights being taken by customers in the partnership with this airline, which means it has broad market and Southeast airline should match customers’ needs and

increase some flights. To better manage the cooperation with other partners, the company could explore more about whether this airline is popular and learn to apply it to other cooperation.

5.2 From the customer analysis

Pay attention to women and old, provide them with special and additional service. (such as providing additional service for caring and helping the old, those pregnant women and women with babies. This is a kind of warm promotion and by doing so the company can eliminate the negative feedback of these vulnerable people, win their customer trust and build their high reputation.

Increase efforts to advertise our additional service about shopping and eating in the Airport to economy class and blue customers. Since no matter whether people shop or eat at the airport, the number of customers in economy class who travel for business or personally counts most. And blue customers usually spend more money at airports. Besides, the number of blue customers and economy class customers is huge compared to others, it is important to attract and encourage them to consume, Southeast airline can give low price discounts and extra bonus.

5.3 From the flight analysis

Make flights on time to improve likelihood to recommend, especially in some east cities. From the boxplots, the longer the departure and arrival delay in minutes, the more likely customer would not recommend. Besides, from the association rule graph, the origin cities from east such as Baltimore, the rule is especially evident, people are more concerned about the delay and cancellations of flights. Their customer experience and buying behavior are influenced by negligence of service of airlines.

Improve customer’s experience of long-time and short-distance travel. In the boxplots, the minutes of flight longer, the more likely customers would not recommend. The long trip made customers feel more easily to get angry with the airline service and airline companies should pay attention to this to develop their service quality or add some other activities during their long trip. Besides, the shorter the distance of the flight, the more likely customers would not recommend. The reason of it may lie in customer dissatisfaction of the long waiting time (including waiting for luggage, security check) and they may think it does not deserve preparation time for their short trip. Company could make a survey of it and serve for customers’ need.

5.4 From the feedback analysis:

Make flights on time and improve the luggage service. In customers of low loyalty and low recommendation level from the 2 sentimental analysis graphs, “time”, “luggage”, “delayed”, were what they mentioned most frequently in both graphs, thus airline should pay more attention to making flights on time and luggage service to reduce negative feedback or customer churn.

Reference：

1. Faraway, J. J. (2005). Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. United Kingdom: CRC Press.
2. https://blog.csdn.net/qq_35837578/article/details/88357551
3. Saltz, J. S., & Stanton, J. M. (2018). An introduction to data science.
4. https://towardsdatascience.com/a-light-introduction-to-text-analysis-in-r-ea291a9865a8
5. https://rpubs.com/pjmurphy/265713

Appendix

```
# IST 687, Standard Homework Heading
#
# Student name: Liya Zhou, Siyun Ding, Renjie Yin, Haolin Liu, Siddhant Prashant
Bandiwadekar
# Homework number: Group 5 Final Project
# Date due: 12/10/2019
#
# Attribution statement: 1 (choose the statements that are true)
# 1. I did this work by myself, with help from the book and the professor

# Run these three functions to get a clean test of homework code
dev.set(dev.next())
dev.off() # Clear the graph window
cat('\014')  # Clear the console
rm(list=ls()) # Clear all user objects from the environment!!!


####################Data Acquisition, Cleansing, Transformation, Munging#############
################################################################################

############## DATA EXTRACTION #################
#read JSON document and save as a data frame
library(jsonlite)
```

```
df <- jsonlite::fromJSON("C:/Users/yinre/Desktop/IST 687/Final Project/fall2019-survey-M07.json")

#################### DATA PREPROCESSING ####################
# delete variables that will not be used in data analysis
df <- df[,-11]

# find out NA's
summary(df)

# change integrate data into numeric ones
df$Age <- as.numeric(df$Age)
df$Price.Sensitivity <- as.numeric(df$Price.Sensitivity)
df$Year.of.First.Flight <- as.numeric(df$Year.of.First.Flight)
df$Flights.Per.Year <- as.numeric(df$Flights.Per.Year)
df$Shopping.Amount.at.Airport <- as.numeric(df$Shopping.Amount.at.Airport)
df$Eating.and.Drinking.at.Airport <- as.numeric(df$Eating.and.Drinking.at.Airport)
df$Scheduled.Departure.Hour <- as.numeric(df$Scheduled.Departure.Hour)
df$Departure.Delay.in.Minutes <- as.numeric(df$Departure.Delay.in.Minutes)
df$Arrival.Delay.in.Minutes <- as.numeric(df$Arrival.Delay.in.Minutes)
df$Flight.time.in.minutes <- as.numeric(df$Flight.time.in.minutes)
df$Flight.Distance <- as.numeric(df$Flight.Distance)

# judge data distribution--not normal distribution
hist(df$Departure.Delay.in.Minutes)
hist(df$Arrival.Delay.in.Minutes)
hist(df$Flight.time.in.minutes)
# fill Na's with median
df[is.na(df$Departure.Delay.in.Minutes),"Departure.Delay.in.Minutes"] <-
median(df$Departure.Delay.in.Minutes, na.rm = T)
df[is.na(df$Arrival.Delay.in.Minutes),"Arrival.Delay.in.Minutes"] <-
median(df$Arrival.Delay.in.Minutes, na.rm = T)
df[is.na(df$Flight.time.in.minutes),"Flight.time.in.minutes"] <- median(df$Flight.time.in.minutes,
na.rm = T)

# delete the state abbreviation in destination and origin city
library(stringr)
df$Destination.City <- gsub("\\,", "", str_extract_all(df$Destination.City, ".*\\,"))
df$Origin.City <- gsub("\\,", "", str_extract_all(df$Origin.City, ".*\\,"))

# extract flight month
df$Flight.Month <- as.numeric(gsub("\\/", "", str_extract_all(df$Flight.date, "^.{1,2}\\/")))
df$Flight.Year <- 2014
# delete flight date
```

```
df <- df[,-15]
library(ggplot2)
ggplot(df, aes(x = Partner.Code, y = frequency(df$Partner.Code), fill = Partner.Name))+
geom_col()

# delete Partner Code since each code correspond to the company.
df <- df[, -15]

# add a new variable of the length of year that a customer have taken the flight
df$Year.Lenghth <- df$Flight.Year-df$Year.of.First.Flight
# delete the used variables
df <- df[, -c(7,31)]

# classify the likelihood recommend
df$recommend.level[which(df$Likelihood.to.recommend > 8)] <- "yes"
df$recommend.level[which(df$Likelihood.to.recommend <= 8)] <- "no"




####################DESCRIPTIVE STATISTICS & VISUALIZATION###################
##############################################################################
# read updated data frame
library(jsonlite)
df <- jsonlite::fromJSON("~/Desktop/df.json")
df_customer <- df[, c(3,4,5,6,7,9,10,11,12,29,8,23,30)]
View(df_customer)

################ Visualization #############
library(ggplot2)
# explore the relationship between likelihood and loyalty
ggplot(data = df, aes(x = Likelihood.to.recommend, y = Loyalty)) + geom_boxplot() +
facet_wrap(~ Likelihood.to.recommend, scales="free")
# As we can see, with the growth of likelihood to recommend, the average of loyalty is
increasing. Withe the growth of likelihood, volatility of lotalty is relatively greater

# explore class, travel type and status distribution
ggplot(df_customer, aes(x = Class)) + geom_bar(width = 2/3) # most economic
ggplot(df_customer, aes(x = Type.of.Travel)) + geom_bar(width = 2/3) # most for business
travel
ggplot(df_customer, aes(x = Airline.Status)) + geom_bar(width = 2/3) # most blue
ggplot(df_customer, aes(x = Class)) +
  geom_bar(aes(fill = Airline.Status)) +
```

```
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor =
element_blank())

# explore year length, age, gendar and price sensitive
ggplot(df_customer, aes(x = Age, y = Price.Sensitivity)) +
  geom_col(aes(col = Gender))
# Female is more sensitive to price than male. People above 80 and between 30 to 50 seem
more sensitive.

# explore flight per year, year length and airline status
ggplot(df_customer, aes(x=Flights.Per.Year)) +
  geom_density(aes(fill=Airline.Status), position = "fill")

# explore price sensitivity and loyalty or likelihood
df_customer %>%
  ggplot() +
  aes(x = Price.Sensitivity) +
  geom_bar(aes(fill = recommend.level))

# explore shopping amount, eating & drinking amount, class and travel type
df_customer$Departure.Delay.in.Minutes <- df$Departure.Delay.in.Minutes
ggplot(df_customer, aes(x = Class, y = Shopping.Amount.at.Airport)) +
  geom_col(aes(color = Type.of.Travel))

ggplot(df_customer, aes(x = Class, y = Shopping.Amount.at.Airport)) +
  geom_col(aes(color = Type.of.Travel))

ggplot(df_customer[df_customer$Shopping.Amount.at.Airport<200 &
df_customer$Eating.and.Drinking.at.Airport<200,], aes(x = Eating.and.Drinking.at.Airport, y =
Shopping.Amount.at.Airport)) +
  geom_point(cex = 2, aes(alpha = Airline.Status, color = Type.of.Travel, shape = Class))

########################## linear model
df_lm <- df_customer
lmodel0 <- lm(formula = Loyalty ~., data = df_lm[,c(-12,-13)])
summary(lmodel0)
lmodel1 <- lm(formula = Likelihood.to.recommend ~ ., data = df_lm[,c(-11,-13)])
summary(lmodel1)
lmodel2 <- lm(formula = recommend.level ~ ., data = df_lm[,c(-11,-12)])

# select remarkable variable
lmodel2 <- lm(formula = Likelihood.to.recommend ~ Airline.Status + Age + Flights.Per.Year +
Type.of.Travel + Shopping.Amount.at.Airport + Eating.and.Drinking.at.Airport +
Departure.Delay.in.Minutes, data = df_customer)
```

```
summary(lmodel2)

lmodel3 <- lm(formula = Likelihood.to.recommend ~ Age, data = df_customer)
summary(lmodel3)




###################### SVM to predict loyalty ####################
install.packages("kernlab")
library(kernlab)

df_svm <- df_customer[,c(-12,-14)]
df_svm$Loyalty[df_svm$Loyalty>=0.5] <- "loyal"
df_svm$Loyalty[df_svm$Loyalty != "loyal"] <- "not loyal"

df_svm$randIndex <- sample(1:dim(df_svm)[1])

cutPoint2_3 <- floor(2 * dim(df_svm)[1]/3)
trainData <- df_svm[df_svm$randIndex[1:cutPoint2_3],]
testData <- df_svm[df_svm$randIndex[(cutPoint2_3+1):dim(df_svm)[1]],]

install.packages("caret")
library(caret)
svmmodel <- ksvm(Loyalty ~ ., data = trainData[,-12], kernel = "rbfdot", kpar = "automatic",
                 C = 5, cross = 3, prob.model = TRUE)
svmPred <- predict(svmmodel, testData)

svmPred <- as.factor(svmPred)
testData$Loyalty <- as.factor(testData$Loyalty)
install.packages("e1071")
library(e1071)
confusionMatrix(svmPred,testData$Loyalty)

svmmodel2 <- ksvm(recommend.level ~ ., data = trainData[,-11], kernel = "rbfdot", kpar =
"automatic",
                  C = 5, cross = 3, prob.model = TRUE)
svmPred2 <- predict(svmmodel2, testData)

svmPred2 <- as.factor(svmPred2)
testData$recommend.level <- as.factor(testData$recommend.level)
confusionMatrix(svmPred2,testData$recommend.level)

################## cart tree to predict recommend #################
```

```
install.packages("tree")
library(tree)
cartTree <- train(recommend.level ~., data = trainData[,-11], method = "treebag", preProc =
c("center","scale"))
preoutput <- predict(cartTree, testData)
preoutput <- as.factor(preoutput)
testData$recommend.level <- as.factor(testData$recommend.level)
confusionMatrix(preoutput,testData$recommend.level)
```

```
##############USE OF MODELING TECHNIQUES & VISUALIZATIONHHHHHHHHHHHHHHHHH
# SVM: Delay and flight time in minutes that may lead to the change of loyalty/likelihood to
recommend
dfnew <- df
View(dfnew)

dfnew$True.Arrival.Delay <- dfnew$Arrival.Delay.in.Minutes
dfnew$True.Arrival.Delay[which(df$Arrival.Delay.in.Minutes < 6)] <- 0
trainindex <- sample(c(1,2), nrow(dfnew),replace= T,prob = c(0.67,0.33))
traindata <- dfnew[trainindex==1,]
testdata <- dfnew[trainindex==2,]
svmOutput1 <- ksvm(Loyalty ~
Departure.Delay.in.Minutes+True.Arrival.Delay+Flight.time.in.minutes,
                   data=traindata,kernel="rbfdot", kpar="automatic",C=40,cross=4, prob.model=TRUE)
svmOutput1

svmresult1 <- predict(svmOutput,testdata,type="votes")
View(svmresult1)
comparable1 <- (testdata[,8]-svmresult)<0.8&(testdata[,8]-svmresult)>-0.7
result1 <- table(comparable)
result1
accuraryratio1 <- result[2]/(sum(result))
accuraryratio1

svmOutput2 <- ksvm(Likelihood.to.recommend ~
Departure.Delay.in.Minutes+True.Arrival.Delay+
                     Flight.time.in.minutes+Flight.Distance,
                   data=traindata,kernel="rbfdot", kpar="automatic",C=40,cross=4, prob.model=TRUE)
```

```
svmOutput2

svmresult2 <- predict(svmOutput,testdata,type="votes")
View(svmresult2)
comparable2 <- (testdata[,25]-svmresult)<6.5&(testdata[,25]-svmresult)>3.5
result2 <- table(comparable)
result2
accuraryratio2 <- result[2]/(sum(result))
accuraryratio2

# correlation in Departure delay,loyalty and LTR
Plot_Dep.Delay <- ggplot(df,aes(x = Departure.Delay.in.Minutes,y = Loyalty))+
  geom_boxplot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Departure delay')
Plot_Dep.Delay

# correlation in Arrival delay, loyalty and LTR
Plot_Arr.Delay <- ggplot(df,aes(x = Arrival.Delay.in.Minutes, y = Loyalty))+
  geom_boxplot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Arrival delay')
Plot_Arr.Delay

# correlation in Flight time, loyalty and LTR
Plot_F.time <- ggplot(df,aes(x = Flight.time.in.minutes, y = Loyalty))+
  geom_spot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Flight time')
Plot_F.time

# correlation in Flight distance, loyalty and LTR
Plot_F.distance <- ggplot(df,aes(x = Flight.Distance, y = Loyalty))+
  geom_boxplot(aes(fill=recommend.level),position = 'dodge')+
  ggtitle('Loyalty versus Flight distance')
Plot_F.distance

################################## Association rules ##############################
install.packages("arules")
install.packages("arulesViz")
install.packages("arules")
install.packages("arulesViz")
library("arules")
library(arulesViz)
df_customer$Old <- "adult"
df_customer$Old[which(df_customer$Age <= 21)] <- "young"
df_customer$Old[which(df_customer$Age >= 60)] <- "old"
```

```
# find out the consumption relationship
df_rule_con <- df_customer[,c(1,3,5,6,7,8,9,10,15)]

df_rule_con$shopping[df_rule_con$Shopping.Amount.at.Airport != 0] <- "yes"
df_rule_con$shopping[df_rule_con$Shopping.Amount.at.Airport == 0] <- "no"

df_rule_con$eating[df_rule_con$Eating.and.Drinking.at.Airport != 0] <- "yes"
df_rule_con$eating[df_rule_con$Eating.and.Drinking.at.Airport == 0] <- "no"

df_rule_con$Flights.Freq <- "med"
df_rule_con$Flights.Freq[which(df_rule_con$Flights.Per.Year <= 10)] <- "low"
df_rule_con$Flights.Freq[which(df_rule_con$Flights.Per.Year >= 30)] <- "high"

df_rule_con$Year.length[which(df_rule_con$Year.Lenghth < 6)] <- "new"
df_rule_con$Year.length[which(df_rule_con$Year.Lenghth >= 6)] <- "regular"

df_rule_con <- df_rule_con[,c(-3,-5,-6,-8)]

df_rule_con <- as(df_rule_con, "transactions")
ruleset_con1 <- apriori(df_rule_con[,-7], parameter = list(support = 0.005, confidence = 0.5),
                        appearance = list(default="lhs", rhs=("shopping=yes")))
ruleset_con2 <- apriori(df_rule_con[,-6], parameter = list(support = 0.005, confidence = 0.5),
                        appearance = list(default="lhs", rhs=("eating=yes")))
plot(ruleset_con1, jitter = 0)
inspect(ruleset_con1[which.max(quality(ruleset_con1)$lift)])
plot(ruleset_con2, jitter = 0)
inspect(ruleset_con2[which.max(quality(ruleset_con2)$lift)])

# find out the most related factor to recommendation
df_rule_r <- df_customer

df_rule_r$Price.Sensitivity <- as.factor(df_rule_r$Price.Sensitivity)

ggplot(aes(x=Flights.Per.Year), data = df_rule_r) + geom_histogram(col="black")
df_rule_r$Flights.Freq <- "med"
df_rule_r$Flights.Freq[which(df_rule_r$Flights.Per.Year <= 10)] <- "low"
df_rule_r$Flights.Freq[which(df_rule_r$Flights.Per.Year >= 30)] <- "high"

ggplot(aes(x=df_rule_r$Year.Lenghth), data = df_rule_r) + geom_histogram(col="black")
df_rule_r$Year.length[which(df_rule_r$Year.Lenghth < 6)] <- "new"
df_rule_r$Year.length[which(df_rule_r$Year.Lenghth >= 6)] <- "regular"

df_rule_r$delay[which(as.numeric(df_rule_r$Departure.Delay.in.Minutes) == 0)] <- "on time"
```

```
df_rule_r$delay[is.na(df_rule_r$delay)] <- "delay"

df_rule_r <- df_rule_r[,c(-2,-5,-7,-8,-10,-11,-12,-14)]

df_rule_r <- as(df_rule_r, "transactions")
ruleset_recom <- apriori(df_rule_r, parameter = list(support = 0.005, confidence = 0.5),
                         appearance = list(default="lhs", rhs=("recommend.level=yes")))
plot(ruleset_recom)
inspect(ruleset_recom)
inspect(ruleset_recom[which.max(quality(ruleset)$lift)])

# find out relationship to loyalty
df_rule_l <- df_customer
df_rule_l$Loyalty[df_rule_l$Loyalty>=0.5] <- "loyal"
df_rule_l$Loyalty[df_rule_l$Loyalty != "loyal"] <- "not loyal"

df_rule_l$Price.Sensitivity <- as.factor(df_rule_l$Price.Sensitivity)

ggplot(aes(x=df_rule_l$Flights.Per.Year), data = df_rule_l) + geom_histogram(col="black")
df_rule_l$Flights.Freq <- "med"
df_rule_l$Flights.Freq[which(df_rule_l$Flights.Per.Year <= 10)] <- "low"
df_rule_l$Flights.Freq[which(df_rule_l$Flights.Per.Year >= 30)] <- "high"

ggplot(aes(x=df_rule_l$Year.Lenghth), data = df_rule_l) + geom_histogram(col="black")
df_rule_l$Year.length[which(df_rule_l$Year.Lenghth < 6)] <- "new"
df_rule_l$Year.length[which(df_rule_l$Year.Lenghth >= 6)] <- "regular"

str(df_rule_l$Departure.Delay.in.Minutes)
df_rule_l$delay[which(as.numeric(df_rule_l$Departure.Delay.in.Minutes) == 0)] <- "on time"
df_rule_l$delay[is.na(df_rule_l$delay)] <- "delay"
df_rule_l <- df_rule_l[,c(-2,-5,-7,-8,-10,-12,-13,-14)]

df_rule_l <- as(df_rule_l, "transactions")
ruleset_loyal <- apriori(df_rule_l, parameter = list(support = 0.005, confidence = 0.5),
                         appearance = list(default="lhs", rhs=("Loyalty=loyal")))
inspect(ruleset_loyal)
inspect(ruleset_loyal[which.max(quality(ruleset_loyal)$lift)])

df_flight <- df
df_flight$Departure.Delay <- df_flight$Departure.Delay.in.Minutes
df_flight$Departure.Delay[which(df_flight$Departure.Delay.in.Minutes > 0)] <- "yes"
df_flight$Departure.Delay[which(df_flight$Departure.Delay.in.Minutes == 0)] <- "no"

df_flight$Arrival.Delay <- df_flight$Arrival.Delay.in.Minutes
```

```
df_flight$Arrival.Delay[which(df_flight$Arrival.Delay.in.Minutes > 0)] <- "yes"
df_flight$Arrival.Delay[which(df_flight$Arrival.Delay.in.Minutes == 0)] <- "no"

df_rule <- df_flight[,c(2,22,33,34,35)]
df_rule <-
data.frame(df_flight$Origin.City,df_flight$Flight.cancelled,df_flight$Departure.Delay,df_flight$Arr
ival.Delay,df_flight$recommend.level)
df_rule <- as(df_rule, "transactions")
ruleset <- apriori(df_rule,
                   parameter = list(support = 0.005, confidence = 0.5),
                   appearance = list(default="lhs", rhs=("recommend.level=yes")))
plot(ruleset,jitter = 0)
inspect(ruleset)
inspect(ruleset[which.max(quality(ruleset)$lift)])

df_rule_l <- df_flight
df_rule_l$Loyalty[df_rule_l$Loyalty>=0.5] <- "loyal"
df_rule_l$Loyalty[df_rule_l$Loyalty != "loyal"] <- "not loyal"
df_rule_l <- df_rule_l[,c(2,9,22,33,34,35)]
df_rule_l <- as(df_rule_l, "transactions")
colnames(df_rule_l$Loyalty) <- "loyalty"
ruleset_l <- apriori(df_rule_l, parameter = list(support = 0.005, confidence = 0.5),
                     appearance = list(default="lhs", rhs=("Loyalty = loyal")))

# flight from or to which place has possibility to cancel or delay
possible_cancel <- df_flight %>%
  filter(Flight.cancelled == 'Yes') %>%
  group_by(Origin.City) %>%
  summarise(count = n())
possible_D.Delay <- df_flight %>%
  filter(Departure.Delay == 'yes') %>%
  group_by(Origin.City) %>%
  summarise(count = n())
possible_A.Delay <- df_flight %>%
  filter(Arrival.Delay == 'yes') %>%
  group_by(Origin.City) %>%
  summarise(count = n())
possible_cancel <- data.frame(possible_cancel)
possible_cancel$count <- sort(possible_cancel$count,decreasing = TRUE)
possible_D.Delay <- data.frame(possible_D.Delay)
possible_D.Delay$count <- sort(possible_D.Delay$count,decreasing = TRUE)
possible_A.Delay <- data.frame(possible_A.Delay)
possible_A.Delay$count <- sort(possible_A.Delay$count,decreasing = TRUE)
head(possible_cancel,10)
```

```
head(possible_D.Delay,10)
head(possible_A.Delay,10)



##############################Partner Analysis##################################
########################################################################


a<-unique(df$Partner.Name)
View(a)

states <- map_data("state")

MAP<- ggplot(data = states) +
  borders("state",colour = "black",fill = "white") +
  coord_fixed(1.3) +
  guides(fill=FALSE)
MAP


EF <- subset(df,Partner.Name=="EnjoyFlying Air Services")
EFmap <- MAP +
  geom_curve(data=EF,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="black",
             size=0.70,
             curvature=0.1) +
  geom_point(data=EF,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  geom_point(data=EF,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
```

```
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12))+
  ggtitle("Enjoyflying")

EFmap



##############################################################################


SE <- subset(df,Partner.Name=="Southeast Airlines Co.")
SEmap <- MAP+
  geom_curve(data=SE,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=SE,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=enjoyflying,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
```

```
      axis.ticks = element_blank(),
      plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("Southeast Airlines Co.")
SEmap

################################################################################


#Cheapseats Airlines Inc.


CA <- subset(df,Partner.Name=="Cheapseats Airlines Inc.")

CAmap <- MAP+
  geom_curve(data=CA,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=CA,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=CA,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("CheapseatsÂ AirlinesÂ Inc.")

CAmap
```

```
#############################################################################
#GoingNorth Airlines Inc.

GN <- subset(df,Partner.Name=="GoingNorth Airlines Inc.")

GNmap <- MAP+
  geom_curve(data=GN,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=GN,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=GN,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("GoingNorth Airlines Inc.")
GNmap

#############################################################################
#FlyToSun Airlines Inc.

FA <- subset(df,Partner.Name=="FlyToSun Airlines Inc.")

FAmap <- MAP+
```

```
  geom_curve(data=FA,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=FA,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=FA,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("FlyToSun Airlines Inc.")
FAmap

################################################################################


#Paul Smith Airlines Inc.

PS <- subset(df,Partner.Name=="Paul Smith Airlines Inc.")

PSmap <- MAP+
  geom_curve(data=PS,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=PS,
```

```
          aes(x=olong,y=olat),
          size = 2.5,
          alpha = 1,
          na.rm = T,
          shape = 20,
          colour = "blue" ) +
  geom_point(data=PS,
          aes(x=dlong,y=dlat),
          size = 2.5,
          alpha = 1,
          na.rm = T,
          shape = 20,
          colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("Paul Smith Airlines Inc")

PSmap

##############################################################################

#OnlyJets Airlines Inc.

OJ <- subset(df,Partner.Name=="OnlyJets Airlines Inc.")
OJmap <- MAP+
  geom_curve(data=OJ,
          aes(x=olong,y=olat,xend=dlong,yend=dlat),
          color="#17202A",
          size=0.70,
          curvature=0.1) +
  geom_point(data=OJ,
          aes(x=olong,y=olat),
          size = 2.5,
          alpha = 1,
          na.rm = T,
          shape = 20,
          colour = "blue" ) +
  geom_point(data=OJ,
```

```
          aes(x=dlong,y=dlat),
          size = 2.5,
          alpha = 1,
          na.rm = T,
          shape = 20,
          colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("OnlyJets Airlines Inc.")

OJmap

##############################################################################

#Sigma Airlines Inc.

SA <- subset(df,Partner.Name=="Sigma Airlines Inc.")
SAmap <- MAP+
  geom_curve(data=SA,
          aes(x=olong,y=olat,xend=dlong,yend=dlat),
          color="#17202A",
          size=0.70,
          curvature=0.1) +
  geom_point(data=SA,
          aes(x=olong,y=olat),
          size = 2.5,
          alpha = 1,
          na.rm = T,
          shape = 20,
          colour = "blue" ) +
  geom_point(data=SA,
          aes(x=dlong,y=dlat),
          size = 2.5,
          alpha = 1,
          na.rm = T,
          shape = 20,
          colour = "blue"
  ) +
```

```
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("Sigma Airlines Inc.")

SAmap


##############################################################################
##

#Northwest Business Airlines Inc.


NBA <- subset(df,Partner.Name=="Northwest Business Airlines Inc.")
NBAmap <- MAP+
  geom_curve(data=NBA,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=NBA,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=NBA,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
```

```
      axis.title.y = element_blank(),
      axis.ticks = element_blank(),
      plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("Northwest Business Airlines Inc.")

NBAmap

##############################################################################


#FlyFast Airways Inc.


FF <- subset(df,Partner.Name=="FlyFast Airways Inc.")
FFmap <- MAP+
  geom_curve(data=FF,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=FF,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=FF,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("FlyFast Airways Inc.")
```

FFmap

##

#Oursin Airlines Inc.

```
OA <- subset(df,Partner.Name=="Oursin Airlines Inc.")
OAmap <- MAP+
  geom_curve(data=OA,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=OA,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=OA,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("Oursin Airlines Inc.")

OAmap
```

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

#FlyHere Airways


FH <- subset(df,Partner.Name=="FlyHere Airways")
FHmap <- MAP+
  geom_curve(data=FH,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=FH,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=FH,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("FlyHere Airways")

FHmap

#########################################################################


#Cool&Young Airlines Inc.

CN <- subset(df,Partner.Name=="Cool&Young Airlines Inc.")
```

```
CNmap <- MAP+
  geom_curve(data=CN,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
  geom_point(data=CN,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=CN,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme( axis.line = element_blank(),
         axis.text.x = element_blank(),
         axis.text.y = element_blank(),
         axis.title.x = element_blank(),
         axis.title.y = element_blank(),
         axis.ticks = element_blank(),
         plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("Cool&Young Airlines Inc.")
CNmap


##############################################################################

#West Airways Inc.


WA <- subset(df,Partner.Name=="West Airways Inc.")
WAmap <- MAP+
  geom_curve(data=WA,
             aes(x=olong,y=olat,xend=dlong,yend=dlat),
             color="#17202A",
             size=0.70,
             curvature=0.1) +
```

```
  geom_point(data=WA,
             aes(x=olong,y=olat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue" ) +
  geom_point(data=WA,
             aes(x=dlong,y=dlat),
             size = 2.5,
             alpha = 1,
             na.rm = T,
             shape = 20,
             colour = "blue"
  ) +
  theme(axis.line = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 12)) +
  ggtitle("West Airways Inc.")


WAmap

#########################################################################

unique(df$Type.of.Travel)
unique(df$Partner.Name)

stat1 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Cheapseats Airlines
Inc.")
One<-nrow(stat1)

stat2 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="GoingNorth Airlines
Inc.")
Two<-nrow(stat2)

stat3 <- df %>%
```

```
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="FlyToSun Airlines
Inc.")
Three<-nrow(stat3)

stat4 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Paul Smith Airlines
Inc.")
Four <- nrow(stat4)

stat5 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="OnlyJets Airlines
Inc.")
Five <- nrow(stat5)

stat6 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Sigma Airlines Inc.")
Six <-nrow(stat6)

stat7 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Northwest Business
Airlines Inc.")
Seven <-nrow(stat7)

stat8 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="FlyFast Airways
Inc.")
Eight <-nrow(stat8)

stat9 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Oursin Airlines Inc.")
Nine <-nrow(stat9)

stat10 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Southeast Airlines
Co.")
Ten <-nrow(stat10)

stat11 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="FlyHere Airways")
Eleven <-nrow(stat11)

stat12 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="EnjoyFlying Air
Services")
```

```
Twelve <-nrow(stat12)

stat13 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="Cool&Young Airlines
Inc.")
Thirteen <-nrow(stat13)

stat14 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Business travel",(df$Partner.Name)=="West Airways Inc.")
Fourteen <-nrow(stat14)


Business.Travel <-
c(One,Two,Three,Four,Five,Six,Seven,Eight,Nine,Ten,Eleven,Twelve,Thirteen,Fourteen)
View(Business.Travel)

#############################################################################

stat11 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Cheapseats Airlines
Inc.")
One1<-nrow(stat11)

stat21 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="GoingNorth Airlines
Inc.")
Two1<-nrow(stat21)

stat31 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="FlyToSun Airlines
Inc.")
Three1<-nrow(stat31)

stat41 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Paul Smith Airlines
Inc.")
Four1 <- nrow(stat41)

stat51 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="OnlyJets Airlines
Inc.")
Five1 <- nrow(stat51)

stat61 <- df %>%
```

```
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Sigma Airlines Inc.")
Six1 <-nrow(stat61)

stat71 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Northwest Business
Airlines Inc.")
Seven1 <-nrow(stat71)

stat81 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="FlyFast Airways
Inc.")
Eight1 <-nrow(stat81)

stat91 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Oursin Airlines
Inc.")
Nine1 <-nrow(stat91)

stat101 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Southeast Airlines
Co.")
Ten1 <-nrow(stat101)

stat111 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="FlyHere Airways")
Eleven1 <-nrow(stat111)

stat121 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="EnjoyFlying Air
Services")
Twelve1 <-nrow(stat121)

stat131 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="Cool&Young
Airlines Inc.")
Thirteen1 <-nrow(stat131)

stat141 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Personal Travel",(df$Partner.Name)=="West Airways Inc.")
Fourteen1 <-nrow(stat141)

Personal.Travel <-
c(One1,Two1,Three1,Four1,Five1,Six1,Seven1,Eight1,Nine1,Ten1,Eleven1,Twelve1,Thirteen1,
Fourteen1)
```

```
View(Personal.Travel)

#############################################################################

stat111 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Cheapseats Airlines
Inc.")
One11<-nrow(stat111)

stat211 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="GoingNorth Airlines
Inc.")
Two11<-nrow(stat211)

stat311 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="FlyToSun Airlines
Inc.")
Three11<-nrow(stat311)

stat411 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Paul Smith Airlines
Inc.")
Four11 <- nrow(stat411)

stat511 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="OnlyJets Airlines
Inc.")
Five11 <- nrow(stat511)

stat611 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Sigma Airlines Inc.")
Six11 <-nrow(stat611)

stat711 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Northwest Business
Airlines Inc.")
Seven11 <-nrow(stat711)

stat811 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="FlyFast Airways
Inc.")
Eight11 <-nrow(stat811)

stat911 <- df %>%
```

```
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Oursin Airlines Inc.")
Nine11 <-nrow(stat911)

stat101 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Southeast Airlines
Co.")
Ten11 <-nrow(stat101)

stat1111 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="FlyHere Airways")
Eleven11 <-nrow(stat1111)

stat1211 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="EnjoyFlying Air
Services")
Twelve11 <-nrow(stat1211)

stat1311 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="Cool&Young Airlines
Inc.")
Thirteen11 <-nrow(stat1311)

stat1411 <- df %>%
  filter(str_trim(df$Type.of.Travel)=="Mileage tickets",(df$Partner.Name)=="West Airways Inc.")
Fourteen11 <-nrow(stat1411)

Mileage.Tickets <-
c(One11,Two11,Three11,Four11,Five11,Six11,Seven11,Eight11,Nine11,Ten11,Eleven11,Twelv
e11,Thirteen11,Fourteen11)
View(Mileage.Tickets)

CustAnalysis <- data.frame(Business.Travel,Personal.Travel,Mileage.Tickets)
View(CustAnalysis)

unique(df$Partner.Name)
Names <- c("Cheapseats Airlines Inc.",
           "GoingNorth Airlines Inc.",
           "FlyToSun Airlines Inc." ,
           "Paul Smith Airlines Inc." ,
           "OnlyJets Airlines Inc.",
           "Sigma Airlines Inc." ,
           "Northwest Business Airlines Inc.",
           "FlyFast Airways Inc.",
           "Oursin Airlines Inc." ,
```

```
        "Southeast Airlines Co.",
        "FlyHere Airways",
        "EnjoyFlying Air Services",
        "Cool&Young Airlines Inc.",
        "Cool&Young Airlines Inc.")

CustAnalysis <- data.frame(Names,Business.Travel,Personal.Travel,Mileage.Tickets)
View(CustAnalysis)


row.names(CustAnalysis[which.max(CustAnalysis$Business.Travel),])
row.names(CustAnalysis[which.max(CustAnalysis$Personal.Travel),])
row.names(CustAnalysis[which.max(CustAnalysis$Mileage.Tickets),])


row.names(CustAnalysis[which.min(CustAnalysis$Business.Travel),])
row.names(CustAnalysis[which.min(CustAnalysis$Personal.Travel),])
row.names(CustAnalysis[which.min(CustAnalysis$Mileage.Tickets),])




##########################Feedback Analysis-Text Mining#########################
################################################################################

#Loading the Package
library(NLP)
library(tm)
library(ggplot2)
library(tidyverse)
library(tidyr)
library(RColorBrewer)
library(wordcloud)


####################Loyalty#########################
#Delete the rows with positive loyalty
df1<-filter(df,df$Loyalty<=0)
#Extracting the free Text
Comment<-df1$freeText
#Cleaning the NA in Text
```

```
Comment<-na.omit(Comment)
words.vec <- VectorSource(c(Comment))
# Make a corpus of web data
words.corpus <- Corpus(words.vec)
#To lower
words.corpus <- tm_map(words.corpus,content_transformer(tolower))
#removePunctuation
words.corpus <- tm_map(words.corpus,removePunctuation)
#removeNumbers
words.corpus <- tm_map(words.corpus, removeNumbers)
#Words no nned to be used
otherwords<-
c("now","well","however","always","way","much","take","took","can","poor","better","even","flying"
,"didnt","bad","dont","due","never","flew","airline","flight","southeast","service","good","one","Alw
ays","also","just","best","nice","great","first","will","many","ever","fly","get","got","really","like","wor
st")
#Taking out stop words
words.corpus <- tm_map(words.corpus, removeWords,c(stopwords("english"),otherwords))


#Create a document-term-matrix
tdm <- TermDocumentMatrix(words.corpus)
#coerce text data back into a plain data matrix
m<-as.matrix(tdm)
wordCounts<-rowSums(m)
wordCounts<-sort(wordCounts,decreasing=TRUE)



#Create a word cloud
cloudFrame<-data.frame(word=names(wordCounts),freq=wordCounts)
wordcloud(cloudFrame$word,cloudFrame$freq,max.words=150, random.order=FALSE,
rot.per=0.15)


#Creating new dataframe with 50 most frequent words
CF25<-head(cloudFrame,25)
CF25<-arrange(CF25,desc(freq))
#Plot word frequencies
CF25%>% ggplot(aes(x = reorder(CF25$word,-CF25$freq), y = CF25$freq)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 25 Feature Words based on Negative Loyalty", x = "Word", y= "Frequency") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
####################NPS#########################
#Delete the rows with positive loyalty
df2<-filter(df,df$Likelihood.to.recommend<7)
#Extracting the free Text
Comment2<-df2$freeText
#Cleaning the NA in Text
Comment2<-na.omit(Comment2)

words.vec2 <- VectorSource(c(Comment2))
# Make a corpus of web data
words.corpus2 <- Corpus(words.vec2)
#To lower
words.corpus2 <- tm_map(words.corpus2,content_transformer(tolower))
#removePunctuation
words.corpus2 <- tm_map(words.corpus2,removePunctuation)
#removeNumbers
words.corpus2 <- tm_map(words.corpus2, removeNumbers)
#Taking out stop words
words.corpus2 <- tm_map(words.corpus2, removeWords,c(stopwords("english"),otherwords))



#Create a document-term-matrix
tdm2 <- TermDocumentMatrix(words.corpus2)
#coerce text data back into a plain data matrix
m2<-as.matrix(tdm2)
wordCounts2<-rowSums(m2)
wordCounts2<-sort(wordCounts2,decreasing=TRUE)


#Create a word cloud
cloudFrame2<-data.frame(word=names(wordCounts2),freq=wordCounts2)
wordcloud(cloudFrame2$word,cloudFrame2$freq,max.words=150, random.order=FALSE,
rot.per=0.15, color= "blue")


#Creating new dataframe with 50 most frequent words
CF252<-head(cloudFrame2,25)
CF252<-arrange(CF252,desc(freq))
#Plot word frequencies
CF252%>% ggplot(aes(x = reorder(CF252$word,-CF252$freq), y = CF252$freq)) +
  geom_bar(stat = "identity",color="blue") +
  labs(title = "Top 25 feature words based on Low NPS", x = "Word", y= "Frequency") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```