IST 687 Introduction to Data Science

Analysis of United States Airlines

AMIT NAIK

1. Introduction

The dataset is about a survey of customers flying within the United States. It has 129889 rows with 28 variables which includes data for three months(January,February,March). The main goal of this project is to increase customer satisfaction for the airlines.

The dependent variable in this dataset is the Customer Satisfaction whereas there are 27 independent variables. These are in various subcategories such as personal attributes (age, gender, price sensitivity,…), flight planned status (class, date, Airline, Origin City, …) and how the flight went (departure delay, flight time,...)

2. Data cleanse/munge/preparation

The cleaning of data started with finding out the NA’s available in the dataset. It was found that the NA’s were available in three columns: Departure Delay, Arrival Delay and the Flight time. The three steps we followed are:

- Replaced the NA’s in departure delay, arrival delay and the flight time with a large value when the flight was canceled. (Flight.cancelled =”Yes”). This means that because we had a larger delay, the flight was cancelled.
- Deleted the 337 rows which had NA’s in the departure delay, arrival delay and the flight time when the flight was not cancelled. Because, in a large dataset with 129889 rows, 337 has a percentage of 0.2.
- There are three satisfaction values with a discrepancy which is deleted. Therefore there are 340 rows deleted in total.

```r
# ---- clean data ---- #

sw <- read.csv("~/Desktop/Satisfaction\ Survey.csv")

# give these NA data a very big value. (The rows which flights are cancelled)
sw$Departure.Delay.in.Minutes <- ifelse(is.na(sw$Departure.Delay.in.Minutes) & sw$Flight.cancelled == "Yes", 9999, sw$Departure.Delay.in.Minutes)
sw$Arrival.Delay.in.Minutes <- ifelse(is.na(sw$Arrival.Delay.in.Minutes) & sw$Flight.cancelled == "Yes", 9999, sw$Arrival.Delay.in.Minutes)
sw$Flight.time.in.minutes <- ifelse(is.na(sw$Flight.time.in.minutes) & sw$Flight.cancelled == "Yes", 9999, sw$Flight.time.in.minutes)

# There are still 337 rows we haven't deal with. Since the number is small, we decided to delete them.
# delete the last 337 rows
sw <- na.omit(sw)

# There are 3 different Satisfaction value.
sw$Satisfaction <- as.numeric(as.character(sw$Satisfaction))

# After we use this function, they change to NA data. We need delete these 3 rows.
df<- na.omit(sw)

# try to create new columns, TotalDelayTime & DelayedFlight & df$FlightSpeed, but examined no use in linear model

# df$TotalDelayTime <- df$Departure.Delay.in.Minutes + df$Arrival.Delay.in.Minutes
# df$DelayedFlight <- ifelse(df$TotalDelayTime == 0, "No", "Yes")
# df$FlightSpeed <- df$Flight.Distance / df$Flight.time.in.minutes
```

Why CheapSeats?

Of all the airlines present in the dataset we choose, Cheapseats as it is one of the airlines with low customer satisfaction. We came to this conclusion by using a word cloud & also finding the ratio of customer satisfaction of the airlines.

Word Cloud

We generated a word cloud to determine which airline has the highest number of customers. We used the customer count as the frequency to generate the word cloud. We found that the
Cheapseats Airline Inc. has the maximum number of customers. Hence, we conduct the analysis on the Cheapseats only.

CODE:

```
install.packages("wordcloud")
library(wordcloud)
createWordCounts<- function(vFtext)
{
  words.vec <- VectorSource(vFtext) #create a Corpus, a "Bag of Words"
  words.corpus <- Corpus(words.vec)
  words.corpus
  words.corpus <- tm_map(words.corpus,content_transformer(tolower))
  words.corpus <- tm_map(words.corpus, removePunctuation)
  words.corpus <- tm_map(words.corpus, removeNumbers)
  words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))
  words.corpus <- tm_map(words.corpus, removeWords, c("airlines","inc"))
  tdm<- TermDocumentMatrix(words.corpus)
  tdm
  m<- as.matrix(tdm)# create a matrix
  wordCounts <- rowSums(m)
  wordCounts<- sort(wordCounts, decreasing=TRUE)
  return(wordCounts)
}
wordCounts<- createWordCounts(cleanedDataset$Airline.Name)
View(wordCounts)

genWordCloud <- function(wordCounts)
{
  cloudFrame <- data.frame( word= names(wordCounts), frequency = wordCounts)
  wordcloud(names(wordCounts),wordCounts, min.freq = 2, max.words=30, rot.per=0.35,
          colors= brewer.pal(8,"Dark2"))

}
genWordCloud(wordCounts)
happyCust <- cleanedDataset[cleanedDataset$Satisfaction>3,]
```

```
View(happyCust)
unhappyCust <-cleanedDataset[cleanedDataset$Satisfaction<=3,]
View(unhappyCust)
wordCounts1<- createWordCounts(happyCust$Airline.Name)
wordCounts1
genWordCloud(wordCounts1)
wordCounts2<- createWordCounts(unhappyCust$Airline.Name)
wordCounts2
genWordCloud(wordCounts2)
```

Table displaying ratio of the cancelled flights

Airline_Name	**flight_num**	**flight_cancelled_num**	**Cancellation Ratio**
FlyFast Airways Inc	15356	661	4.30%
EnjoyFlying Air Services	8906	319	3.58%
OnlyJets Airline Inc	5382	123	2.29%
FlyHere Airways	2474	51	2.06%
Northwest Business Airlines	13787	248	1.80%
SouthEast Airlines	9555	132	1.38%
Oursin Airlines Inc.	10953	151	1.38%
Paul Smith Airlines Inc.	12207	156	1.28%
Sigma Airlines Inc.	17018	217	1.28%
Cheapseats Airlines Inc.	25985	316	1.22%
FlyToSun Airlines Inc.	3392	20	0.59%
GoingNorth Airlines Inc.	1568	6	0.38%
Cool&Young Airlines Inc.	1281	1	0.08%

From the above table it is quite clear that cheapSeats has the high cancellation ratio.

Net Promoter Score:

The Net Promoter Score(NPS) is used to determine the loyalty of the customers towards its provider. The NPS is based on a simple question i.e how likely is it for the customers to recommend the product or company to others. There are three components of the Net promoter score namely the promoters,passive & detractors.

Promoters(9 or 10):
The promoters are very likely to recommend the company/products to others

Passive (7 or 8):
These customers are satisfied with the service but might not or not enthusiastic about recommending the product/company

Detractors(0 to 6):
These customers are not satisfied and may not recommend the service/company to others.

We used the NPS package to find out the net promoter score for the dataset.

Code:

```
install.packages("NPS")
library("NPS")
vector1 <- as.numeric(as.character( cleanedDataset$Satisfaction))
summary(vector1)
ss <-npc(vector1, breaks = list(1:2.5, 3, 3.5:5)) #Based on our customer satisfaction
ss # We get 3 levels: Detractor Passive Promoter
sum(as.numeric(as.character(ss)))#calculate the total number of rows
table(ss)
```

Output for detractors:

Cheapseats Airlines Inc. 5602	Cool&Young Airlines Inc. 222	EnjoyFlying Air Services 1884
FlyFast Airways Inc. 3327	FlyHere Airways 512	FlyToSun Airlines Inc. 608
GoingNorth Airlines Inc. 396	Northwest Business Airlines 2731	OnlyJets Airlines Inc. 1183
Oursin Airlines Inc. 2241	Paul Smith Airlines Inc. 2369	Sigma Airlines Inc. 3334
Southeast Airlines Co. 1852	West Airways Inc. 272	

As you can see Cheapseats has the highest number of detractors.

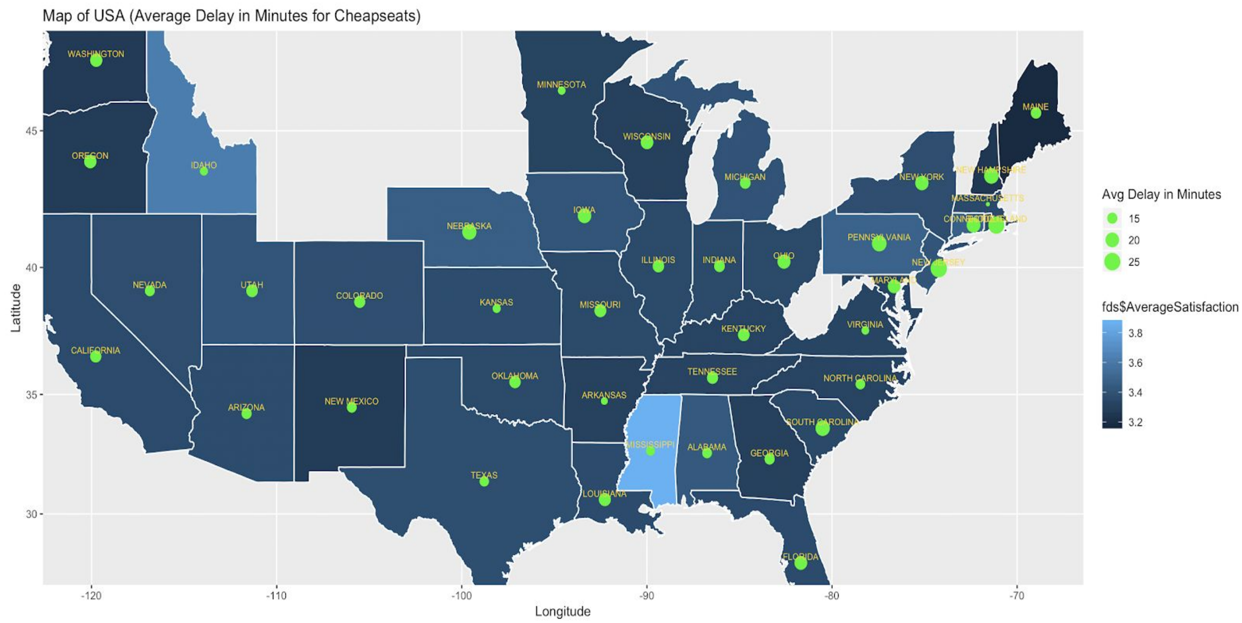
From the results of the world cloud, table displaying the cancellation ration & the Net promoter score, we choose to focus on the cheapseats airline for improving its customer satisfaction.

.

3. Descriptive Statistics & Visualization

USING GGMAPS FOR DISPLAYING AVERAGE DELAY IN MINUTES PER STATE:

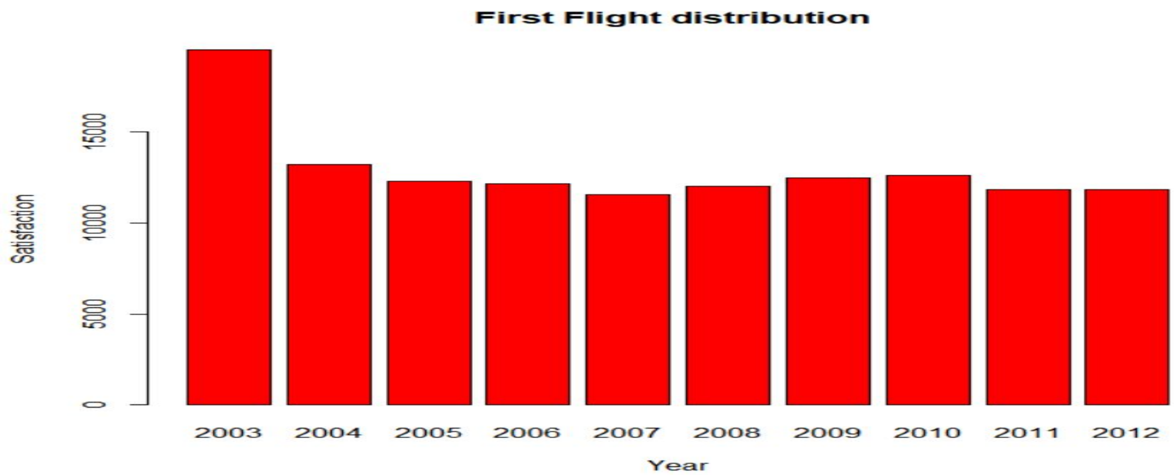
We created a map to display the satisfaction of the different states(shades depend on the level of satisfaction). The point on the map determines the average delay in minutes(the larger the dot the greater is the delay).



Code:

```
library(sqldf)
statesDelay<- sqldf('select "Destination.State" as "stateName", avg("Arrival.Delay.in.Minutes") as "adih",avg("Satisfaction") as "AverageSatisfaction" from dfAir1 group by "Destination.State"')

#taking and merging default system data with our dataset
area <- state.area
latlong<- state.center
stateName<- state.name
mergeDf<- data.frame(stateName,latlong,area)

fds<- merge(mergeDf,statesDelay, by='stateName')

#using lower case for stateName
fds$stateName<-tolower(fds$stateName)

us <- map_data("state")


#ggmaps
m.s1<-ggplot(fds , aes(map_id=stateName))
m.s1<- m.s1 + geom_map(map = us,aes(fill=fds$AverageSatisfaction),color="white")
m.s1<- m.s1 + expand_limits(x= fds$x,y=fds$y)
m.s1<- m.s1 + geom_point(data=fds, aes(x=fds$x,y=fds$y,size=fds$adih), color ="green")+ scale_size(name="Avg Delay in Minutes")
m.s1<- m.s1 + geom_text( data=fds, hjust=0.5, vjust=-0.5, aes(x=x, y=y, label=toupper(stateName)), colour="gold", size=2.5 )
m.s1<- m.s1 + coord_map() + ggtitle("Map of USA (Average Delay in Minutes for Cheapseats")+ xlab("Longitude") + ylab("Latitude")
m.s1
```

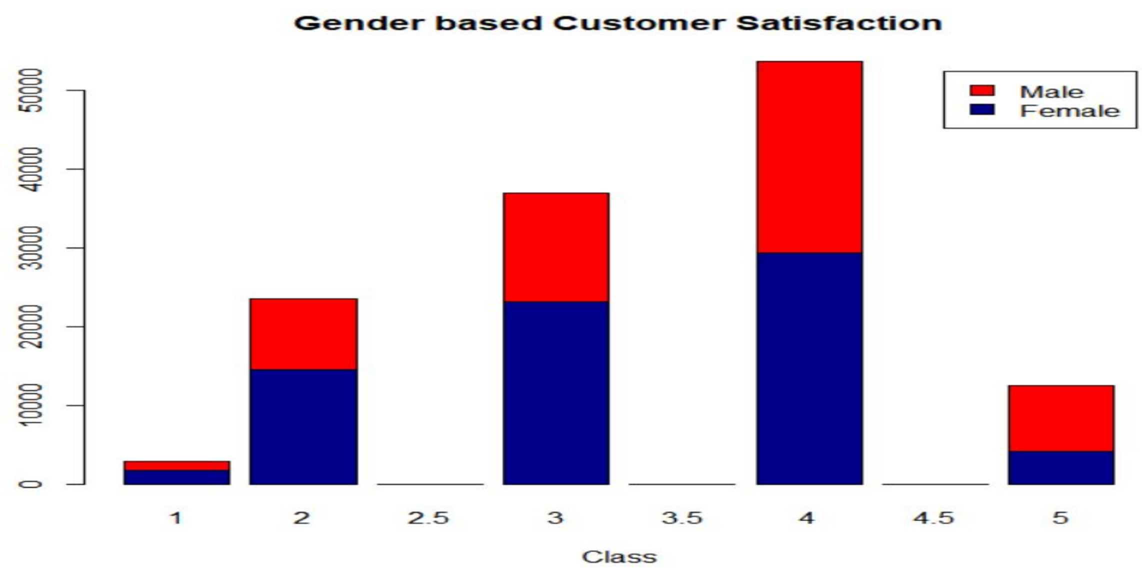
We have analyzed the first flight distribution and how has it impacted the overall customer satisfaction.



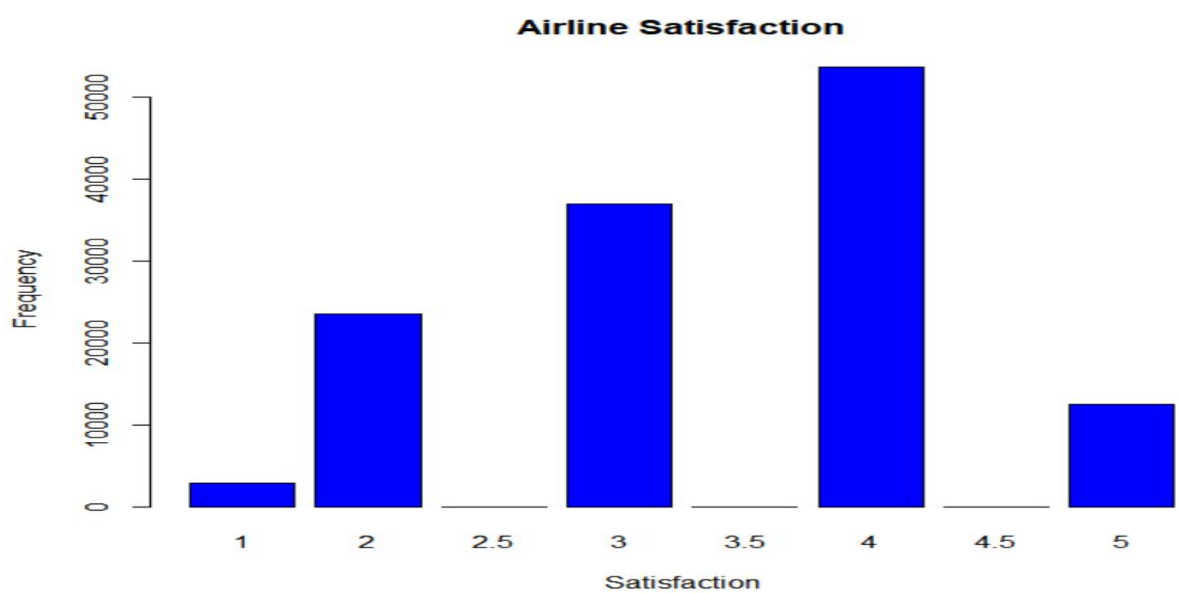
Code:

```
counts <- table( cleanedDataset$Price.Sensitivity, cleanedDataset$Age)

counts <- table(cleanedDataset$Year.of.First.Flight)
barplot(counts, main="First Flight distribution ",
      xlab="Year", col=c("RED"))
```

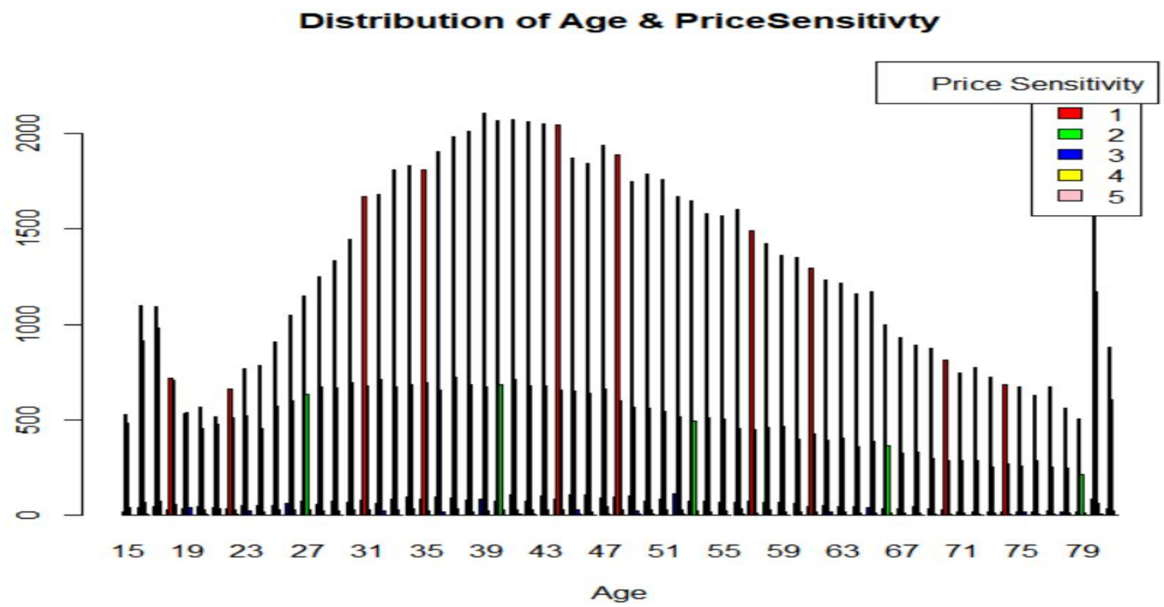
The following graph depicts the gender based customer satisfaction.



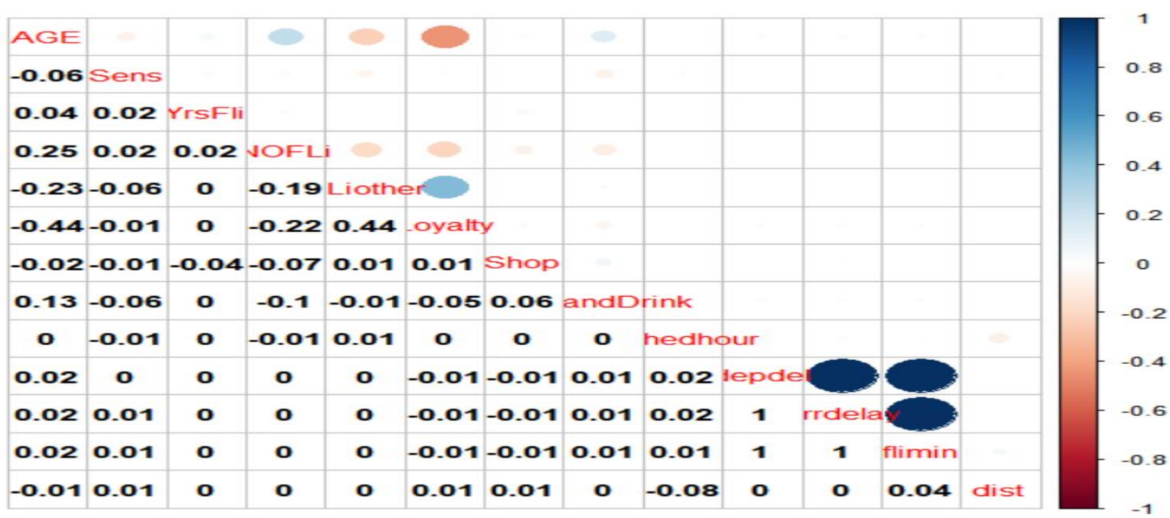
The following graph depicts the frequency of the airline distribution.



The following graph depicts the distribution of Age & Price Sensitivity



CORRELATION PLOT :



Code:

```
install.packages("corrplot")
install.packages("corrgram")
library(corrplot)
library(corrgram)
Cheapseats<- dfAir1
names(Cheapseats)
Cheapseats<- Cheapseats[,c(1,3,5:8,10:12,22:24,26,27)]
```

```
names(Cheapseats)<-c("SAT","AGE","Sens",
"YrsFli","NOFLi","FLiother","Loyalty","Shop","EatandDrink","Schedhour","depdel","arrdelay","flimin","dist")
Cheapseats<- na.omit(Cheapseats)
corr_data <- cor(Cheapseats)
corplot1 <- corrplot.mixed (corr_data,lower.col = "red",number.cex = 0.7)
```

4. BUSINESS QUESTIONS:

(1) For Cheapseats Airline, which factors have significant impact on customers satisfaction?
(2) How does flights problems affect satisfaction rating?
(3) How does person attributes affect satisfaction rating?

5. LINEAR MODELING:

Business Question 1: For Cheapseats Airline, which factors have significant impact on customers satisfaction?

Using the cleaned data, we performed Linear Modelling to determine which factors affect the satisfaction the most. The satisfaction was the dependent variable and all the other factors are independent variables.
We have created a new dataframe consisting of only Cheapseats data for carrying out the linear modelling.

Code for creating Cheapseats datasets:

```
cleanedDataset$Airline.Name <- trim(cleanedDataset$Airline.Name)
CheapseatsAirlineDF <- cleanedDataset[which(cleanedDataset$Airline.Name=="Cheapseats Airlines Inc."),]
View(CheapseatsAirlineDF)
```

Code for linear modelling**:**

```
cheapseats<- CheapseatsAirlineDF[-16:-17]
```

```
linearModel3 <- lm(formula = as.numeric(Satisfaction) ~ ., data = cheapseats)
linearModelS <- lm(formula = as.numeric(Satisfaction) ~ Airline.Status + Age + Gender + Price.Sensitivity + Year.of.First.Flight + No.of.Flights.p.a + Type.of.Travel + Shopping.Amount.at.Airport +Eating.and.Drinking.at.Airport + Class + Scheduled.Departure.Hour + Departure.Delay.in.Minutes + Arrival.Delay.in.Minutes + Flight.cancelled + Flight.time.in.minutes + Flight.Distance + Arrival.Delay.greater.5.Mins, data = CheapseatsAirlineDF)
summary(linearModel3)
linearModel3
summary(linearModelS)
linearModelS
```

Using the linear model we found, the following significant variables:

Variables	Coefficient
Airline.StatusGold	0.06
Airline.StatusPlatinum	0.22
Airline.StatusSilver	-0.16
AgeLow	-0.07
GenderMale	0.06
Type.of.TravelMileage tickets	-0.19
Type.of.TravelPersonal Travel	0.29
ClassEco Plus	-0.04
Scheduled.Departure.hourLow	-0.03
Flights.cancelledYes	-0.13
Arrival.Delay.greater.5.minutes	0.17

6. Validation

We used Association Rules and SVM to validate the result from linear regression model.

Association Rules:

Using association rules, we determined which factors would impact the satisfaction for the Cheapseats airlines. The RHS was set to satisfaction to analyze which factors most affect the satisfaction.

Code:

```
library(arules)
library(arulesViz)

ruleDF1 <- data.frame(CheapseatsAirlineDF$Satisfaction, CheapseatsAirlineDF$Airline.Status,
CheapseatsAirlineDF$Age, CheapseatsAirlineDF$Gender, CheapseatsAirlineDF$Price.Sensitivity,
CheapseatsAirlineDF$Year.of.First.Flight,
                         CheapseatsAirlineDF$No.of.Flights.p.a, CheapseatsAirlineDF$X..of.Flight.with.other.Airlines,
CheapseatsAirlineDF$Type.of.Travel, CheapseatsAirlineDF$No..of.other.Loyalty.Cards,
                         CheapseatsAirlineDF$Shopping.Amount.at.Airport,
CheapseatsAirlineDF$Eating.and.Drinking.at.Airport, CheapseatsAirlineDF$Class,
CheapseatsAirlineDF$Day.of.Month,
                    CheapseatsAirlineDF$Scheduled.Departure.Hour, CheapseatsAirlineDF$Departure.Delay.in.Minutes,
CheapseatsAirlineDF$Arrival.Delay.in.Minutes,
                         CheapseatsAirlineDF$Flight.cancelled, CheapseatsAirlineDF$Flight.time.in.minutes,
CheapseatsAirlineDF$Flight.Distance, CheapseatsAirlineDF$Arrival.Delay.greater.5.Mins)
ruleX <- as(ruleDF1, "transactions")
ruleX

ruleset <- apriori(ruleX, parameter = list(support=0.30,confidence=0.30,maxtime=10, maxlen=30),appearance =
list(default="lhs", rhs=("CheapseatsAirlineDF.Satisfaction=High")))
ruleset <- sort(ruleset, decreasing = TRUE, by="lift")
inspect(ruleset)
summary(CheapseatsAirlineDF)


ruleDF2 <- data.frame(CheapseatsAirlineDF$Satisfaction, CheapseatsAirlineDF$Airline.Status
,CheapseatsAirlineDF$Age, CheapseatsAirlineDF$Gender, CheapseatsAirlineDF$Price.Sensitivity,
                         CheapseatsAirlineDF$Shopping.Amount.at.Airport,
CheapseatsAirlineDF$Eating.and.Drinking.at.Airport, CheapseatsAirlineDF$Class,
CheapseatsAirlineDF$Day.of.Month,
                CheapseatsAirlineDF$Flight.cancelled, CheapseatsAirlineDF$Arrival.Delay.greater.5.Mins)
ruleX <- as(ruleDF2, "transactions")
ruleX

ruleset <- apriori(ruleX, parameter = list(support=0.30,confidence=0.30,maxtime=10, maxlen=30),appearance =
list(default="lhs", rhs=("CheapseatsAirlineDF.Satisfaction=High")))
ruleset <- sort(ruleset, decreasing = TRUE, by="lift")
```

```
inspect(ruleset)
summary(CheapseatsAirlineDF)
```

The following observations were made from the arules:

- For the customers, whose flight delayed and who are low class/status tend to have low satisfaction
- For the customers, whose price sensitivity are low and are male tend to give high satisfaction.

Support Vector Machine:

With the help of SVM, we determined how various factors affect the overall likelihood to recommend. The following snippet shows the code used to run the SVM.

CODE:

```
library(kernlab)
createBucketsSurveya <- function(vec) {
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec >= 3] <- "Happy"
  vBuckets[vec < 3] <- "unHappy"
  return(vBuckets)
}
dfAir1<- CheapseatsAirlineDF1
dfAir1$SatHuH <- createBucketsSurveya(as.numeric(dfAir1$Satisfaction))

#before bucketing all the fields, add a column SatHuH which indicates the buckets
# acc to satisfaction taking <3 as Low and >=3 as high
dfAirF <- subset(dfAir1, SatHuH == "Happy" | SatHuH == "unHappy")
dfAirF
table(dfAirF$SatHuH)
dim(dfAirF)
randIndex <- sample(1:dim(dfAirF)[1])
head(randIndex)
cutPoint2_3 <- floor(2 * dim(dfAirF)[1]/3)
cutPoint2_3
trainData <- dfAirF[randIndex[1:cutPoint2_3],]
testData <- dfAirF[randIndex[(cutPoint2_3 + 1):dim(dfAir1)[1]],]
dim(trainData)
dim(testData)
modelKs1<-ksvm(SatHuH~Airline.Status+Age+Gender+Price.Sensitivity+Arrival.Delay.greater.5.Mins+Class+Fligh
t.cancelled,data=trainData, kernel="rbfdot", kpar="automatic", C=5, cross=3, prob.model =TRUE)
```

```
#age,gender, price sens,arrivaldelay>5,class,typeoftr,flight canceleld
summary(modelKs1)
modelKs1
svmPred<-predict(modelKs,testData,type="votes")
head(svmPred)
svmPred1<-predict(modelKs1,testData,type="votes")
compTable<- data.frame(testData$SatHuH,svmPred[2,])
#Confusion Matrix
cm1<-table(compTable)
```

OUTPUT:

```
# Support Vector Machine object of class "ksvm"
# SV type: C-svc  (classification)
# parameter : cost C = 5
# Gaussian Radial Basis kernel function.
# Hyperparameter : sigma =  0.283666711460145
# Number of Support Vectors : 7056
# Objective Function Value : -31257.64
# Training error : 0.1737
# Cross validation error : 0.17751
# Probability model included.

#Confusion Matrix
cm1<-table(compTable)
#
# svmPred.2...
# testData.SatHuH    0    1
# Happy   6800    0
# unHappy    1 1861
```

From the output is can be observed that the training error is 0.17 i.e(17%). The accuracy is 83% which proves that the model is good and validates the results of the linear modelling

Business Question 2: How does flights problems affect satisfaction rating?

There are 2 types of variables in the key factors we calculated. One is Flights information, the other one is personal Attributes. This business question is talking about the relationship of flights problem and satisfaction.

When it comes flights problem, the biggest impact could be the cancellation of flight.

Flight cancelled	Number of flights	Average of Satisfaction
Yes	2401	3.11

No	127148	3.38

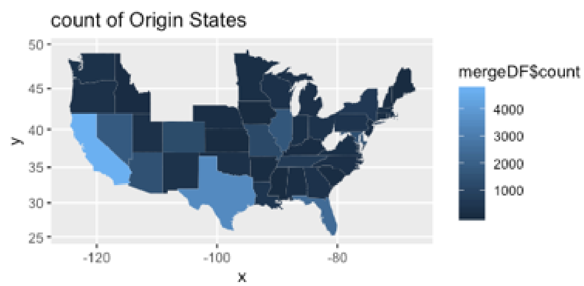
You could easily tell that, cancelled flight have a big impact on average of satisfaction. But, the number of cancelled flights is too little, which will have only a small impact on the whole picture if we prepare solution for these cancelled flights.

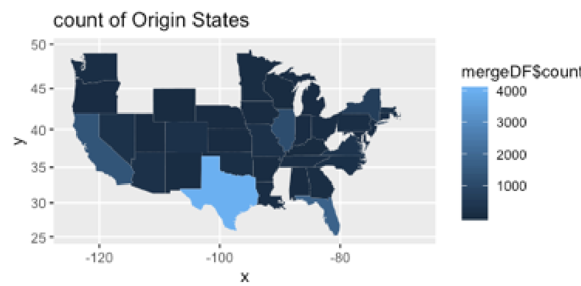
So, we begin to focus on the flights which were delayed. There are many variables describe delay information, and we believe that, we only need to focus on the arrival delay which will have a big effect on the passengers' schedule. In the end, we decide to use "Arrival.Delay.greater.5.Mins", which is dummy variable and ignore the delay is less than 5 mins.

Flight delayed	Number of flights	Average of Satisfaction
Yes	44504	3.17
No	85045	3.49

Wow, there are a large number of flights delayed in the whole dataset! And considering the average of satisfaction, we could have a big impact if we work on delay problem. It also seems like flight delay is easier to fix than flight cancel.

The analysis for flight cancel and delay is for the whole dataset. Since we have chose Cheapseats Airlines, we love to find a competitor for Cheapseats. Considering the probability of different city/airport have different delay objective condition. This competitor we have chose is very similar than Cheapseats, which means the mean customer are not differed in states.





The left graph is Cheapseats Airlines, which delayed ratio is 41.54%.
The right graph is Paul Smith Airlines, which delayed ratio is 29.59%.

From these two pictures, we could find they have similar customers, but Paul Smith Airlines did better in delay issue.

And what needs to be include is that, Cheapseats have 25985 flights during these data collecting time, while Paul Smith has 12207 flights. Cheapseats needs be careful about this increasing competitor!

Why are we choosing a competitor for Cheapseats? Because we want to make sure the conclusion is actionable. If some airlines can do this thing that good, why Cheapseats can't?

We also investigate whether delay flights can be improved. Finally we find that, only 4.3% delayed flights is caused by extreme weather, large amount of delay issue is caused by late-arriving aircraft, maintenance, crew problems, aircraft cleaning, baggage loading and fueling.

So, Cheapseats needs to improve more on delay issue!

Business Question 3: How does person attributes affect satisfaction rating?

Gender	Number	Average of Satisfaction
Female	14624	3.24

Male	11361	3.51

Age	Number	Average of Satisfaction
Middle-aged	5475	3.67
Old	9932	3.06
Young	10578	3.48

For this issue, we researched gender and age.

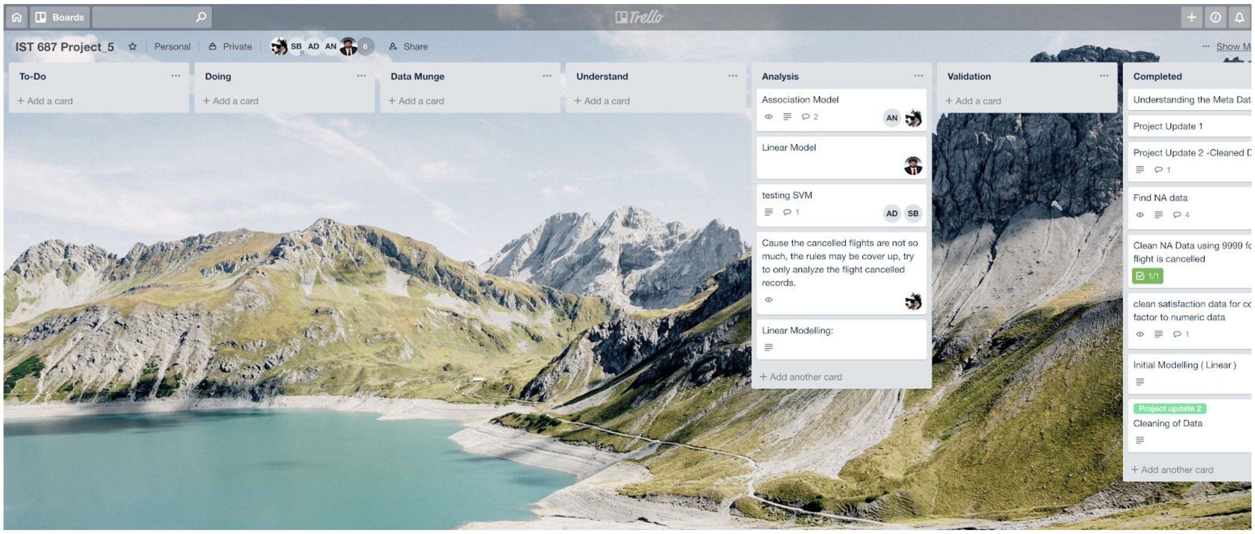
We could easily tell, that age is old and gender is female always gives a lower rating.

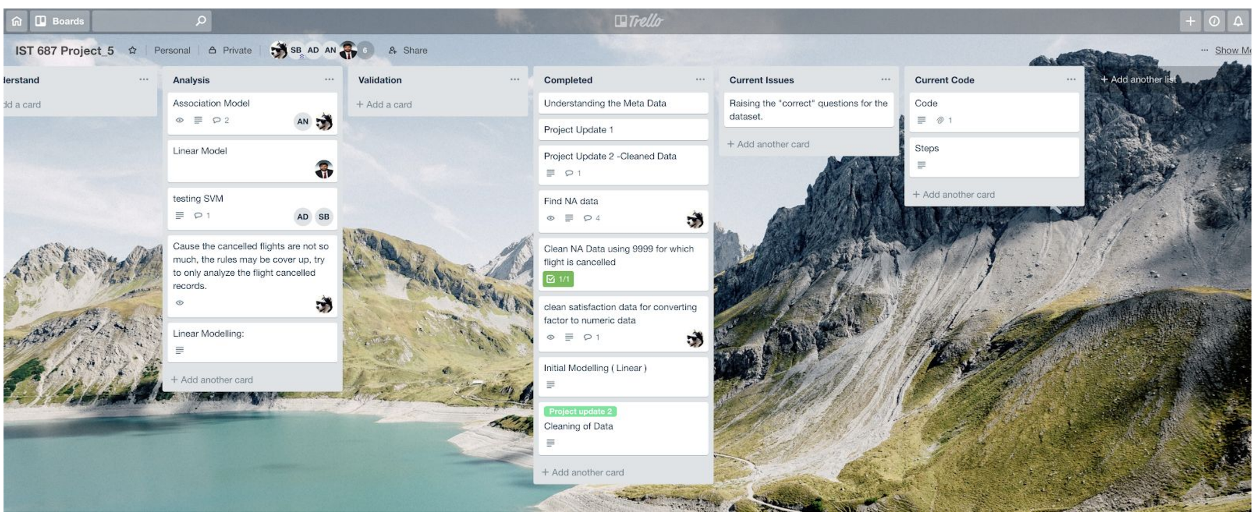
For these 2 groups of people, when the cost is limited, Cheapseats Airlines need to give their best choice and service they have, to get a better satisfaction insights from these kind of people.

7. Actionable Insights and conclusions:

1. Based on the analysis of these 3 business questions and our analysis, we have found the key factors of satisfaction are Airline.Status, Age, Gender, Type of Travel, Class, Flights.cancelled, Arrival.Delay.greater.5.minutes.
2. We research on the flights problem, especially the delay issue. We find a big competitor of Cheapseats, and explain why Cheapseat needs to have some improvement in delay issue, and why this way is actionable, in business question 2.
3. We research on the personal problem. For people who are old or female, when the cost is limited, Cheapseats Airlines need to give their best choice and service they have, to get a better satisfaction ratings from these kind of people.

8. Trello board Screenshots





APPENDIX:

```
#Final Code


#Getting Data
dataset <- read.csv("SatisfactionSurvey.csv")

#Cleaning Data
cleanedDataset <- dataset
```

```
View(cleanedDataset)
which(is.na(dataset$Departure.Delay.in.Minutes))

#The Case of replacing with Highest number
cleanedDataset$Departure.Delay.in.Minutes <- ifelse(is.na(cleanedDataset$Departure.Delay.in.Minutes) & cleanedDataset$Flight.cancelled == "Yes", 9999, cleanedDataset$Departure.Delay.in.Minutes)
cleanedDataset$Arrival.Delay.in.Minutes <- ifelse(is.na(cleanedDataset$Arrival.Delay.in.Minutes) & cleanedDataset$Flight.cancelled == "Yes", 9999, cleanedDataset$Arrival.Delay.in.Minutes)
cleanedDataset$Flight.time.in.minutes <- ifelse(is.na(cleanedDataset$Flight.time.in.minutes) & cleanedDataset$Flight.cancelled == "Yes", 9999, cleanedDataset$Flight.time.in.minutes)
cleanedDataset <- na.omit(cleanedDataset)
summary(cleanedDataset)
str(dataset)
str(cleanedDataset)
which(is.na(cleanedDataset$Departure.Delay.in.Minutes))

cleanedDataset$Satisfaction <- as.numeric(as.character(cleanedDataset$Satisfaction))

# After we use this function, they change to NA data. We need delete these 3 rows.
cleanedDataset<- na.omit(cleanedDataset)
str(cleanedDataset$Satisfaction)

################################### NPS ################################
################# Using NPS to find which Airlines to use #############

install.packages("NPS")
library("NPS")
vector1 <- as.numeric(as.character( cleanedDataset$Satisfaction))
summary(vector1)
```

ss <-npc(vector1, breaks = list(1:2.5, 3, 3.5:5)) #Based on our customer satisfaction
ss # We get 3 levels: Detractor Passive Promoter
sum(as.numeric(as.character(ss)))#calculate the total number of rows
table(ss)
#Detractor Passive Promoter
26533 36888 53608
#Now find the Airline to chose from a Detractor and a Promoter
PromoterAirlines <- cleanedDataset[which(ss=="Promoter"),] #creating a dataset with highest customer satisfaction
View(PromoterAirlines)
DetractorAirlines <- cleanedDataset[which(ss=="Detractor"),]#creating a dataset with highest customer satisfaction
summary(PromoterAirlines$Airline.Name)
# Cheapseats Airlines Inc. Cool&Young Airlines Inc. EnjoyFlying Air Services
# 10528 581 3577
# FlyFast Airways Inc. FlyHere Airways FlyToSun Airlines Inc.
# 6262 1051 1489
# GoingNorth Airlines Inc. Northwest Business Airlines Inc. OnlyJets Airlines Inc.
# 586 5710 2142
# Oursin Airlines Inc. Paul Smith Airlines Inc. Sigma Airlines Inc.
# 4551 5187 7118
# Southeast Airlines Co. West Airways Inc.
# 4039 787
#Thus we select Cheapseats Airline
summary(DetractorAirlines$Airline.Name)
# Cheapseats Airlines Inc. Cool&Young Airlines Inc. EnjoyFlying Air Services
# 5602 222 1884
# FlyFast Airways Inc. FlyHere Airways FlyToSun Airlines Inc.
# 3327 512 608
# GoingNorth Airlines Inc. Northwest Business Airlines Inc. OnlyJets Airlines Inc.
# 396 2731 1183

```
#      Oursin Airlines Inc.           Paul Smith Airlines Inc.                Sigma
Airlines Inc.
#                   2241                             2369                              3334
#    Southeast Airlines Co.                  West Airways Inc.
#                   1852                              272
# Thus we select Cheapseats Airlines


################Using wordcount to find the airline which has low
satisfation########
install.packages("tm")
library(tm)
install.packages("wordcloud")
library(wordcloud)
createWordCounts<- function(vFtext)
{
  words.vec <- VectorSource(vFtext) #create a Corpus, a "Bag of Words"
  words.corpus <- Corpus(words.vec)
  words.corpus
  words.corpus <- tm_map(words.corpus,content_transformer(tolower))
  words.corpus <- tm_map(words.corpus, removePunctuation)
  words.corpus <- tm_map(words.corpus, removeNumbers)
  words.corpus <- tm_map(words.corpus, removeWords,
stopwords("english"))
  words.corpus <- tm_map(words.corpus, removeWords, c("airlines","inc"))
  tdm<- TermDocumentMatrix(words.corpus)
  tdm
  m<- as.matrix(tdm)# create a matrix
  wordCounts <- rowSums(m)
  wordCounts<- sort(wordCounts, decreasing=TRUE)
  return(wordCounts)
}
wordCounts<- createWordCounts(cleanedDataset$Airline.Name)
View(wordCounts)

genWordCloud <- function(wordCounts)
{
   cloudFrame <- data.frame( word= names(wordCounts), frequency =
wordCounts)
```

```
  wordcloud(names(wordCounts),wordCounts, min.freq = 2, max.words=30,
rot.per=0.35,
        colors= brewer.pal(8,"Dark2"))

}
genWordCloud(wordCounts)
happyCust <- cleanedDataset[cleanedDataset$Satisfaction>3,]
View(happyCust)
unhappyCust <-cleanedDataset[cleanedDataset$Satisfaction<=3,]
View(unhappyCust)
wordCounts1<- createWordCounts(happyCust$Airline.Name)
wordCounts1
genWordCloud(wordCounts1)
wordCounts2<- createWordCounts(unhappyCust$Airline.Name)
wordCounts2
genWordCloud(wordCounts2)

############################## Creating Buckets For
Complete Dataset #####################
#Sattisfaction Grouping: High >3 , Average = 3, Low < 3
createBucketsSurvey <- function(vec) {
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec > 3] <- "High"
  vBuckets[vec <= 2] <- "Low"
  return(vBuckets)
}
#arbitary selection of quantile to be 40% and 60%
createBuckets <- function(vec) {
  q <- quantile(vec, c(0.4, 0.6))
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec <= q[1]] <- "Low"
  vBuckets[vec > q[2]] <- "High"
  return(vBuckets)
}

Bucketing <- function(a){
  Satisfaction <- createBucketsSurvey(a$Satisfaction)
  head(Satisfaction)
  Airline.Status <- a$Airline.Status
```

```
Age <- createBuckets(a$Age)
head(Age)
Gender <- a$Gender
Price.Sensitivity <- createBucketsSurvey(a$Price.Sensitivity)
head(Price.Sensitivity)
Year.of.First.Flight <- createBuckets(a$Year.of.First.Flight)
head(Year.of.First.Flight)
No.of.Flights.p.a <- createBuckets(a$No.of.Flights.p.a.)
X..of.Flight.with.other.Airlines <- createBuckets(a$X..of.Flight.with.other.Airlines)
Type.of.Travel <- a$Type.of.Travel
No..of.other.Loyalty.Cards <- createBuckets(a$No..of.other.Loyalty.Cards)
Shopping.Amount.at.Airport <- createBuckets(a$Shopping.Amount.at.Airport)
Eating.and.Drinking.at.Airport <- createBuckets(a$Eating.and.Drinking.at.Airport)
Class <- a$Class
Day.of.Month <- createBuckets(a$Day.of.Month)
# Flight.date, don't know how to examine
Flight.date <- a$Flight.date
Airline.Code <- a$Airline.Code
head(Airline.Code)
Airline.Name <- a$Airline.Name
# Orgin.City, Origin.State, Destination.City, Destination.State don't know
Origin.City <- a$Orgin.City
Origin.State <- a$Origin.State
Destination.State <- a$Destination.State
Destination.City <- a$Destination.City

Scheduled.Departure.Hour <- createBuckets(a$Scheduled.Departure.Hour)
head(Scheduled.Departure.Hour)
Departure.Delay.in.Minutes <- createBuckets(a$Departure.Delay.in.Minutes)
head(Departure.Delay.in.Minutes)
Arrival.Delay.in.Minutes <- createBuckets(a$Arrival.Delay.in.Minutes)
head(Arrival.Delay.in.Minutes)
Flight.cancelled <- a$Flight.cancelled
Flight.time.in.minutes <- createBuckets(a$Flight.time.in.minutes)
head(Flight.time.in.minutes)
```

```
  Flight.Distance <- createBuckets(a$Flight.Distance)
  head(Flight.Distance)
  Arrival.Delay.greater.5.Mins <- a$Arrival.Delay.greater.5.Mins

      df <- data.frame(Satisfaction, Airline.Status, Age, Gender,
Price.Sensitivity, Year.of.First.Flight,
                       No.of.Flights.p.a, X..of.Flight.with.other.Airlines,
Type.of.Travel, No..of.other.Loyalty.Cards,
                       Shopping.Amount.at.Airport, Eating.and.Drinking.at.Airport,
Class, Flight.date, Day.of.Month,
                       Airline.Code, Airline.Name, Scheduled.Departure.Hour,
Departure.Delay.in.Minutes, Arrival.Delay.in.Minutes,
                       Flight.cancelled, Flight.time.in.minutes, Flight.Distance,
Arrival.Delay.greater.5.Mins,Origin.City,Origin.State,Destination.City,Destina
tion.State)
  #View(df)
  return(df)
}



###################### Creating Subset Dataset
###################
install.packages("gdata")
library(gdata)


#Removing whitespace
cleanedDataset$Airline.Name <- trim(cleanedDataset$Airline.Name)
CheapseatsAirlineDF <-
cleanedDataset[which(cleanedDataset$Airline.Name=="Cheapseats Airlines
Inc."),]
View(CheapseatsAirlineDF)
CheapseatsAirlineDF1<-CheapseatsAirlineDF
dfAir1<-CheapseatsAirlineDF1
str(CheapseatsAirlineDF)
summary(CheapseatsAirlineDF)

####CheapseatsAirlineDF
```

```
#Verifying the number of rows to be 25,985
table(cleanedDataset$Airline.Name)

CheapseatsAirlineDF <- Bucketing(CheapseatsAirlineDF)
View(CheapseatsAirlineDF)


FullDataset <- Bucketing(cleanedDataset)
View(FullDataset)

CheapSeatsdataset <- Bucketing(CheapseatsAirlineDF)
View(CheapseatsAirlineDF)


################################## Linear Model For complete Data Set ##############################
#Find significant Columns
linearModel1 <- lm(formula = as.numeric(Satisfaction) ~ . , data = FullDataset)
linearModel2 <- lm(formula = as.numeric(Satisfaction) ~ Airline.Status + Age + Gender + Price.Sensitivity + Year.of.First.Flight + No.of.Flights.p.a + Type.of.Travel + Shopping.Amount.at.Airport +Eating.and.Drinking.at.Airport + Class + Scheduled.Departure.Hour + Departure.Delay.in.Minutes + Arrival.Delay.in.Minutes + Flight.cancelled + Flight.time.in.minutes + Flight.Distance + Arrival.Delay.greater.5.Mins, data = FullDataset)
summary(linearModel1)
linearModel1
summary(linearModel2)
linearModel2

################################## Linear Model For Cheapseat Airline Data Set ##############################
#Find significant Columns
#Removing 2 factor data
cheapseats<- CheapseatsAirlineDF[-16:-17]
```

```
linearModel3 <- lm(formula = as.numeric(Satisfaction) ~ ., data = cheapseats)
linearModelS <- lm(formula = as.numeric(Satisfaction) ~ Airline.Status + Age + Gender + Price.Sensitivity + Year.of.First.Flight + No.of.Flights.p.a + Type.of.Travel + Shopping.Amount.at.Airport +Eating.and.Drinking.at.Airport + Class + Scheduled.Departure.Hour + Departure.Delay.in.Minutes + Arrival.Delay.in.Minutes + Flight.cancelled + Flight.time.in.minutes + Flight.Distance + Arrival.Delay.greater.5.Mins, data = CheapseatsAirlineDF)
summary(linearModel3)
linearModel3
summary(linearModelS)
linearModelS

########################### Observation from Linear Modelling ###########################

## Following columns play a Significant role in improving Customer Satisfaction for Cheapseat Airlines.
#CheapseatsAirlineDF$Age CheapseatsAirlineDF$Gender , CheapseatsAirlineDF$Price.Sensitivity, CheapseatsAirlineDF$Flight.cancelled, CheapseatsAirlineDF$Departure.Delay.in.Minutes
#1.
l1 <- lm(formula = as.numeric(Satisfaction) ~ Age, data = CheapseatsAirlineDF)
summary(l1)
#p-value: < 2.2e-16 which is less that 0.5
#Thus we reject Null Hypothesis
#Conclusion: Age affects the Customer Satisfaction for the CheapseatsAirline


lmcs1 <- lm(formula= as.numeric(Satisfaction)~ Age, data = dfAir1)
summary(lmcs1)
lmcs1
lmcs2 <- lm(formula= as.numeric(Satisfaction)~ Gender, data = dfAir1)
summary(lmcs2)
lmcs2
```

```
lmcs3 <- lm(formula= as.numeric(Satisfaction)~ Price.Sensitivity, data =
dfAir1)
summary(lmcs3)
lmcs3
lmcs4 <- lm(formula=
as.numeric(Satisfaction)~Eating.and.Drinking.at.Airport , data = dfAir1)
summary(lmcs4)
lmcs4 # not significant at all
lmcs6 <- lm(formula= as.numeric(Satisfaction)~Arrival.Delay.greater.5.Mins
, data = dfAir1)
summary(lmcs6)
lmcs6
lmcs7 <- lm(formula= as.numeric(Satisfaction)~Airline.Status , data =
dfAir1)
summary(lmcs7)
lmcs7
lmcs8 <- lm(formula= as.numeric(Satisfaction)~ Class , data = dfAir1)
summary(lmcs8)
lmcs8
lmcs9 <- lm(formula= as.numeric(Satisfaction)~ Type.of.Travel , data =
dfAir1)
summary(lmcs9)
lmcs9

summary(dfAir1)

############################## KSVM
MODEL##################################

# any variable starting with dfAir*
library(kernlab)

createBucketsSurveya <- function(vec) {
  vBuckets <- replicate(length(vec), "Average")
  vBuckets[vec >= 3] <- "Happy"
  vBuckets[vec < 3] <- "unHappy"
  return(vBuckets)
}
dfAir1<- CheapseatsAirlineDF1
```

```
dfAir1$SatHuH <- createBucketsSurveya(as.numeric(dfAir1$Satisfaction))

#before bucketing all the fields, add a column SatHuH which indicates the buckets
# acc to satisfaction taking <3 as Low and >=3 as high

dfAirF <- subset(dfAir1, SatHuH == "Happy" | SatHuH == "unHappy")
dfAirF
table(dfAirF$SatHuH)

dim(dfAirF)
randIndex <- sample(1:dim(dfAirF)[1])
head(randIndex)

cutPoint2_3 <- floor(2 * dim(dfAirF)[1]/3)
cutPoint2_3

trainData <- dfAirF[randIndex[1:cutPoint2_3],]
testData <- dfAirF[randIndex[(cutPoint2_3 + 1):dim(dfAir1)[1]],]

dim(trainData)
dim(testData)

modelKs<-ksvm(SatHuH ~.,data=trainData, kernel="rbfdot", kpar="automatic", C=5, cross=3, prob.model =TRUE)
modelKs
# Support Vector Machine object of class "ksvm"
#
# SV type: C-svc  (classification)
# parameter : cost C = 5
#
# Gaussian Radial Basis kernel function.
# Hyperparameter : sigma =  0.0304469805112404
#
# Number of Support Vectors : 960
#
# Objective Function Value : -220.7832
# Training error : 0
```

```
# Cross validation error : 0.00052
# Probability model included.

modelKs1<-ksvm(SatHuH ~
Airline.Status+Age+Gender+Price.Sensitivity+Arrival.Delay.greater.5.Mins+
Class+Flight.cancelled,data=trainData, kernel="rbfdot", kpar="automatic",
C=5, cross=3, prob.model =TRUE)
#age,gender, price sens,arrivaldelay>5,class,typeoftr,flight canceleld
summary(modelKs1)
modelKs1
# Support Vector Machine object of class "ksvm"
#
# SV type: C-svc  (classification)
# parameter : cost C = 5
#
# Gaussian Radial Basis kernel function.
# Hyperparameter : sigma =  0.283666711460145
#
# Number of Support Vectors : 7056
#
# Objective Function Value : -31257.64
# Training error : 0.1737
# Cross validation error : 0.17751
# Probability model included.


svmPred<-predict(modelKs,testData,type="votes")
head(svmPred)
svmPred1<-predict(modelKs1,testData,type="votes")
compTable<- data.frame(testData$SatHuH,svmPred[2,])
#Confusion Matrix
cm1<-table(compTable)
#
# svmPred.2...
# testData.SatHuH    0    1
# Happy   6800    0
# unHappy    1 1861

compTable1<- data.frame(testData$SatHuH,svmPred1[2,])
```

```
#Confusion Matrix
cm2<-table(compTable1)
# testData.SatHuH    0    1
# Happy   6685  115
# unHappy 1399  463


#Confusion Matrix
ES1<- cm1[1,2]+cm1[2,1]
ES2<- cm2[1,2]+cm2[2,1]

er1<- ES1/sum(cm1)*100
# 0
er2<- ES2/sum(cm2)*100
# 17.55946

################################## Validating Our Selected Columns using Association Model ###################
library(arules)
library(arulesViz)

ruleDF1 <- data.frame(CheapseatsAirlineDF$Satisfaction, CheapseatsAirlineDF$Airline.Status, CheapseatsAirlineDF$Age, CheapseatsAirlineDF$Gender, CheapseatsAirlineDF$Price.Sensitivity, CheapseatsAirlineDF$Year.of.First.Flight, CheapseatsAirlineDF$No.of.Flights.p.a, CheapseatsAirlineDF$X..of.Flight.with.other.Airlines, CheapseatsAirlineDF$Type.of.Travel, CheapseatsAirlineDF$No..of.other.Loyalty.Cards, CheapseatsAirlineDF$Shopping.Amount.at.Airport, CheapseatsAirlineDF$Eating.and.Drinking.at.Airport, CheapseatsAirlineDF$Class, CheapseatsAirlineDF$Day.of.Month, CheapseatsAirlineDF$Scheduled.Departure.Hour, CheapseatsAirlineDF$Departure.Delay.in.Minutes, CheapseatsAirlineDF$Arrival.Delay.in.Minutes, CheapseatsAirlineDF$Flight.cancelled, CheapseatsAirlineDF$Flight.time.in.minutes, CheapseatsAirlineDF$Flight.Distance, CheapseatsAirlineDF$Arrival.Delay.greater.5.Mins)
```

```
ruleX <- as(ruleDF1, "transactions")
ruleX

ruleset <- apriori(ruleX, parameter = list(support=0.30,confidence=0.30,maxtime=10, maxlen=30),appearance = list(default="lhs", rhs=("CheapseatsAirlineDF.Satisfaction=High")))
ruleset <- sort(ruleset, decreasing = TRUE, by="lift")
inspect(ruleset)
summary(CheapseatsAirlineDF)


ruleDF2 <- data.frame(CheapseatsAirlineDF$Satisfaction, CheapseatsAirlineDF$Airline.Status ,CheapseatsAirlineDF$Age, CheapseatsAirlineDF$Gender, CheapseatsAirlineDF$Price.Sensitivity, CheapseatsAirlineDF$Shopping.Amount.at.Airport, CheapseatsAirlineDF$Eating.and.Drinking.at.Airport,
CheapseatsAirlineDF$Class, CheapseatsAirlineDF$Day.of.Month, CheapseatsAirlineDF$Flight.cancelled, CheapseatsAirlineDF$Arrival.Delay.greater.5.Mins)
ruleX <- as(ruleDF2, "transactions")
ruleX

ruleset <- apriori(ruleX, parameter = list(support=0.30,confidence=0.30,maxtime=10, maxlen=30),appearance = list(default="lhs", rhs=("CheapseatsAirlineDF.Satisfaction=High")))
ruleset <- sort(ruleset, decreasing = TRUE, by="lift")
inspect(ruleset)
summary(CheapseatsAirlineDF)

#OBSERVATIONS
#For the customers **whose flight delayed and who are low class/status** tend to give **low satisfaction**
#For the customers **whose price sensitivity are low and Male** tend to give **high satisfaction**
```

#####ggplot2 for age########

############################ PROPOSALS ####################################

1

We propose to reduce the delay in arrival of flights at major cities like Los Angeles, San Jose, Seattle, San Diego, Phoenix, Flint, Norfolk, Rochester, West Palm Beach/Palm Beach, New York, NY.

Proposal is made on an analysis of the airline with highest and lowest average arrival delay time and the major cities being affected in those were then segregated.

Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify

```
library(ggplot2)
# g <- ggplot(c) + aes(x= reorder(c$Orgin.City,c$Airline.Code), y=c$count
) + geom_col(aes(fill = c$Airline.Code))
# g <- g + theme(axis.text.x = element_text(angle = 45, hjust = 1))
# g <- g + ggtitle("Average Arrival Delay Count greater than 5 Minutes
across cities")
# g
```

2

```
#Using Arules prove that Airline.Status=Blue,Class=Eco,Price.Sensitivity=Low,Flight.cancelled=No together gives a negative impact i.e. gives a lower Satisfaction
ruleDF3 <- data.frame(CheapseatsAirlineDF$Satisfaction, CheapseatsAirlineDF$Airline.Status ,CheapseatsAirlineDF$Age, CheapseatsAirlineDF$Gender, CheapseatsAirlineDF$Price.Sensitivity,
                        CheapseatsAirlineDF$Shopping.Amount.at.Airport, CheapseatsAirlineDF$Eating.and.Drinking.at.Airport,
CheapseatsAirlineDF$Class, CheapseatsAirlineDF$Day.of.Month,
                        CheapseatsAirlineDF$Flight.cancelled, CheapseatsAirlineDF$Arrival.Delay.greater.5.Mins)
ruleX <- as(ruleDF3, "transactions")
ruleX

ruleset <- apriori(ruleX, parameter = list(support=0.15,confidence=0.20,maxtime=10, maxlen=30),appearance = list(default="lhs", rhs=("CheapseatsAirlineDF.Satisfaction=Low")))
ruleset <- sort(ruleset, decreasing = TRUE, by="lift")
inspect(ruleset)
summary(CheapseatsAirlineDF)

##### age vs satisfaction ###########









################################## GGMAP
#ggmaps code - Avg Delay in Minutes - Average Satisfaction - Displayed on Map

#for ggmaps

# View(dfAir1)
```

```
summary(dfAir1)


#Removing 9999 Values which were introduced for removing Na's
sqldf(' select count("Arrival.Delay.in.Minutes") from dfAir1 where "Arrival.Delay.in.Minutes" = 9999')
dfAirif<- dfAir1[dfAir1$Arrival.Delay.in.Minutes!=9999,]
dfAir1<- dfAirif

#extracting data form the dataset for destination and average arrival in delay
library(sqldf)
statesDelay<- sqldf('select "Destination.State" as "stateName", avg("Arrival.Delay.in.Minutes") as "adih",avg("Satisfaction") as "AverageSatisfaction"  from dfAir1 group by "Destination.State"')

#taking and merging default system data with our dataset
area <- state.area
latlong<- state.center
stateName<- state.name
mergeDf<- data.frame(stateName,latlong,area)

fds<- merge(mergeDf,statesDelay, by='stateName')

#using lower case for stateName
fds$stateName<-tolower(fds$stateName)

us <- map_data("state")



#ggmaps
m.s1<-ggplot(fds , aes(map_id=stateName))
m.s1<- m.s1 + geom_map(map = us,aes(fill=fds$AverageSatisfaction),color="white")
m.s1<- m.s1 + expand_limits(x= fds$x,y=fds$y)
m.s1<- m.s1 + geom_point(data=fds, aes(x=fds$x,y=fds$y,size=fds$adih), color ="green")+ scale_size(name="Avg Delay in Minutes")
m.s1<- m.s1 + geom_text( data=fds, hjust=0.5, vjust=-0.5, aes(x=x, y=y, label=toupper(stateName)), colour="gold", size=2.5 )
```

```
m.s1<- m.s1 + coord_map() + ggtitle("Map of USA (Average Delay in Minutes for Cheapseats)")+ xlab("Longitude") + ylab("Latitude")
m.s1


#####BAR CHART#######
counts <- table( cleanedDataset$Price.Sensitivity, cleanedDataset$Age)

counts <- table(cleanedDataset$Year.of.First.Flight)
barplot(counts, main="First Flight distribution ",
        xlab="Year", col=c("RED"))




##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify ##### Rectify

### SEVITHA PLEASE RECTIFY THIS CODE

####GGRIDGES###########GGRIDGES###########GGRIDGES##########GGRIDGES###########GGRIDGES###########GGRIDGES###########GGRIDGES#######
install.packages("ggridges")
library(ggridges)
ggplot(ss1) + geom_density_ridges(aes( x = ss1$Satisfaction, y = ss1$Type.of.Travel, fill = ss1$Type.of.Travel), scale = 3) +
  scale_fill_brewer("Status", palette = "Set1")

#################Corrolation Plot##########################Corrolation Plot##########################Corrolation Plot####################
install.packages("corrplot")
install.packages("corrgram")
library(corrplot)
library(corrgram)
Cheapseats<- dfAir1
```

```
names(Cheapseats)
Cheapseats<- Cheapseats[,c(1,3,5:8,10:12,22:24,26,27)]
names(Cheapseats)<-c("SAT","AGE","Sens",
"YrsFli","NOFLi","FLiother","Loyalty","Shop","EatandDrink","Schedhour","dep
del","arrdelay","flimin","dist")
Cheapseats<- na.omit(Cheapseats)
corr_data <- cor(Cheapseats)
corplot1 <- corrplot.mixed (corr_data,lower.col = "red",number.cex = 0.7)
```