CUSTOMER SATISFACTION DATA ANALYSIS FOR

Cool & Young Airlines, Inc.

Prepared by: Group 1
Brent C. Schuler /Justin T. Hesley
Matthew W. Sutherland /Meng Cai

Prepared for: IST687 | Summer 2019
Applied Data Science

Date: September 16, 2019

Table of Contents

Introduction

"There may be a million reasons for high customer satisfaction, the unsatisfied share only a few commons, and we need to identify them."

- group motto

PROJECT SCOPE

High customer satisfaction drives every company's potential revenue streams upward. With this in mind, it is vital that Cool & Young Airlines, Inc. stays up to date on what propels customers to their business and what areas of their service demands to improve, for profitability to continue to grow through high customer satisfaction ratings.

By analyzing the satisfaction survey data from 129,889 customers, ranging from 15 to 85 years old, with flights within United States from January 1, 2014 through March 31, our analytical team took a comparison approach that successfully identified the key characteristics of customers with low satisfaction, and then tested the assumption with 3 different analytical models. Within our report, we will provide Cool & Young Airlines, Inc. with actionable insights on the data we analyzed and recommendations for improvements on services to better customer satisfaction.

PROJECT CONTEXT & BACKGROUND

This is an analytical project for a hypothetical corporate customer in the aviation industry. The data set was provided to the research team via an airline wide satisfaction survey, with requirements for identifying the key drivers to low customer satisfaction within a specific airline company and then provide that company with strategies towards improvement. After an overview of the full data set, our team decided to pick the airline with the least amount of data, Cool & Young Airlines Inc ("VX"[1]), because we wanted to challenge ourselves in identifying data patterns with a small data set.

The challenge took hold from the start of our work, as quite a few of the analytical results were counterintuitive. For example, flight cancellation does not have an apparent correlation with customer satisfaction of VX. We quickly adjusted our approach to establishing industrial baselines, which incorporates all survey data. Within the confines of the industrial baseline, we

[1] VX will be used for Cool & Young Airlines, Inc. for the rest of the report.

were able to gauge the outcomes of descriptive analysis for the data of solely VX, to filter the key attributes for low satisfaction. This will become our road map for success:

Broad Picture of Industry -> Identity Baselines -> Compare Target Airline -> Actionable Insights

Business Questions

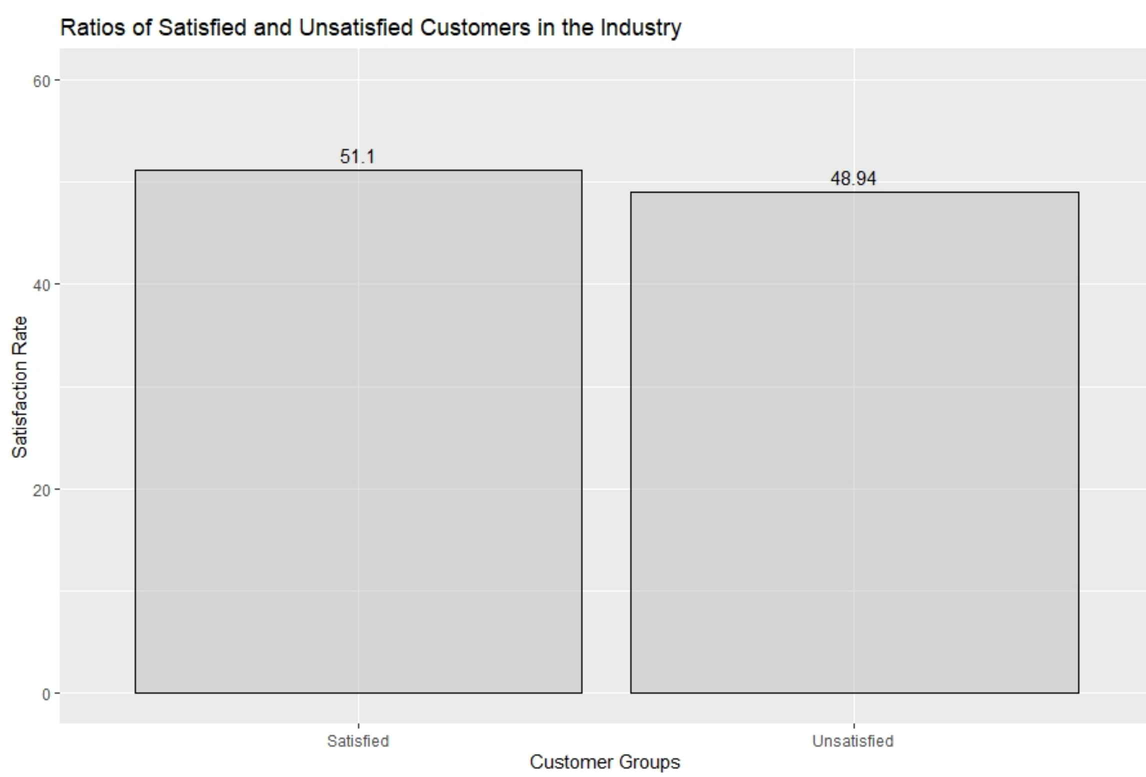
1) **What are the satisfaction ratings in the data set? What is the average satisfaction rate in the industry?**

```
> sort(unique(survey$satisfaction))
[1] 1.0 2.0 2.5 3.0 3.5 4.0 4.5 5.0

> # compute overall satisfaction rate of the industry as a baseline:
> AvgSat <- round(mean(vis$satisfaction), 3)
> AvgSat
[1] 3.379
```

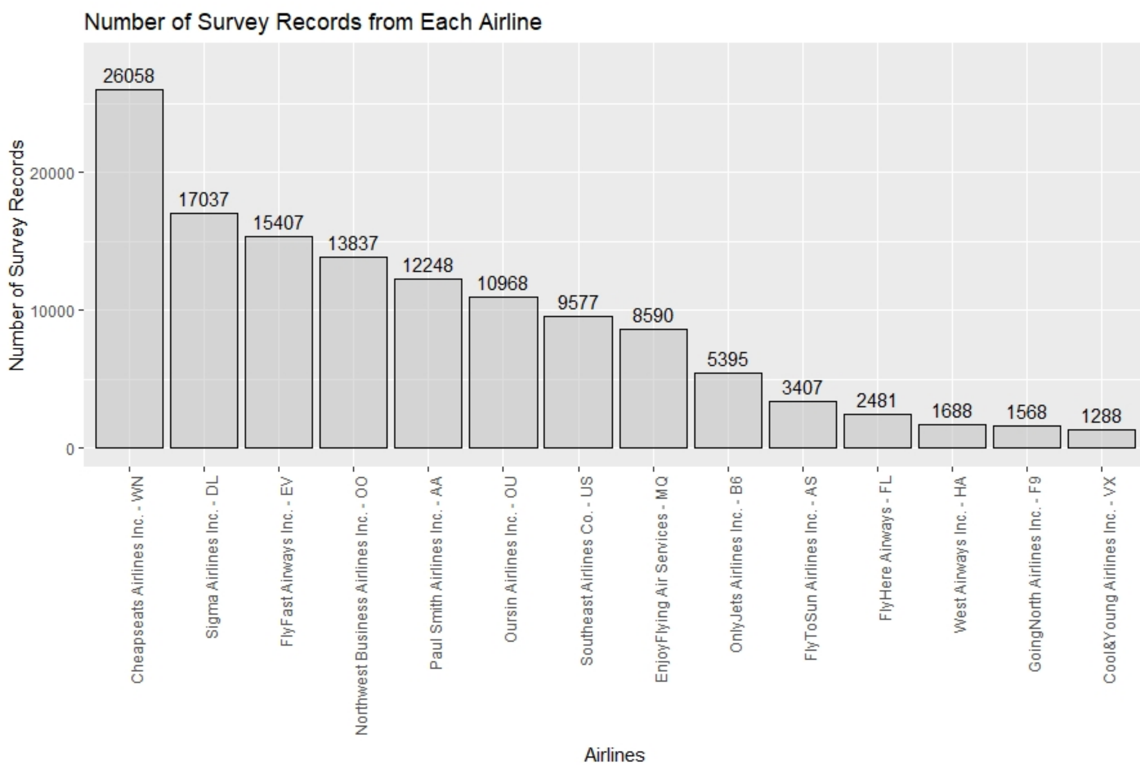
- The satisfaction ratings range from 1 to 5, with an increment at 0.5. The industrial average satisfaction is 3.379, which is below our lower limit for the satisfied.

2) **What are the satisfied and unsatisfied ratios of all customers? Assuming 0-3 as unsatisfied, and 3.5 – 5 as satisfied.**



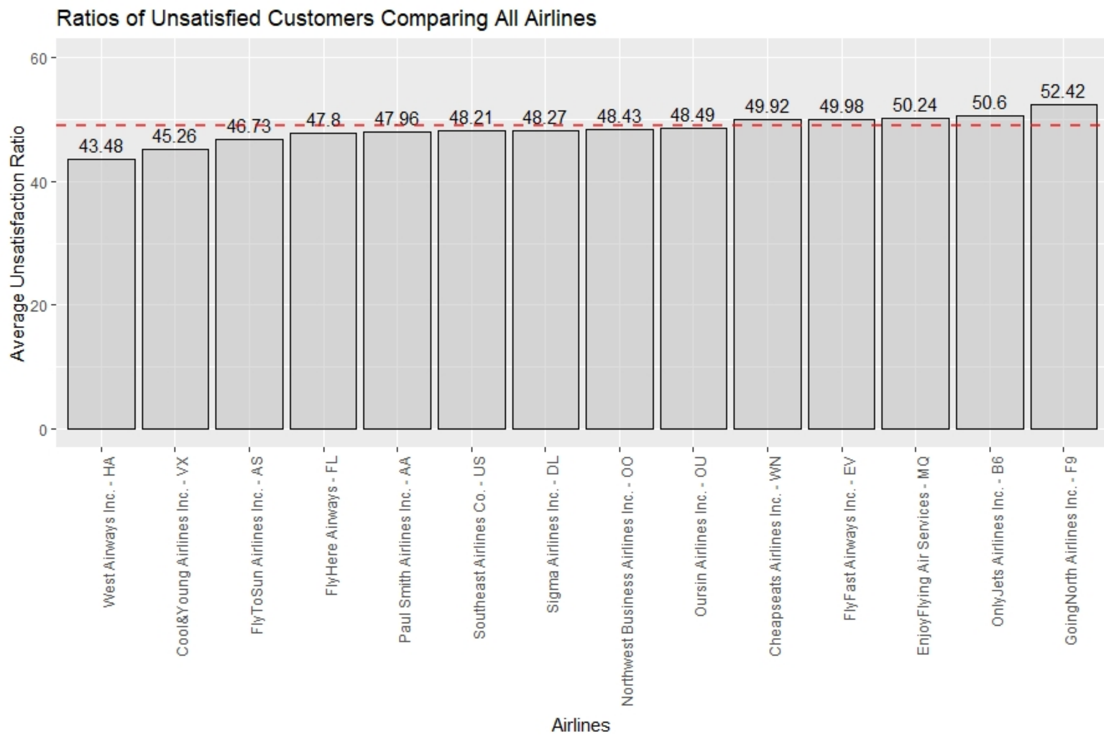
Out of the 129,889 customers, there are 51.1% satisfied and 48.9% unsatisfied. The numbers are very even, with a bit more satisfied customers than the unsatisfied.

3) **What is the airline rank for a number of surveys, customer satisfaction, and unsatisfied ratios?**

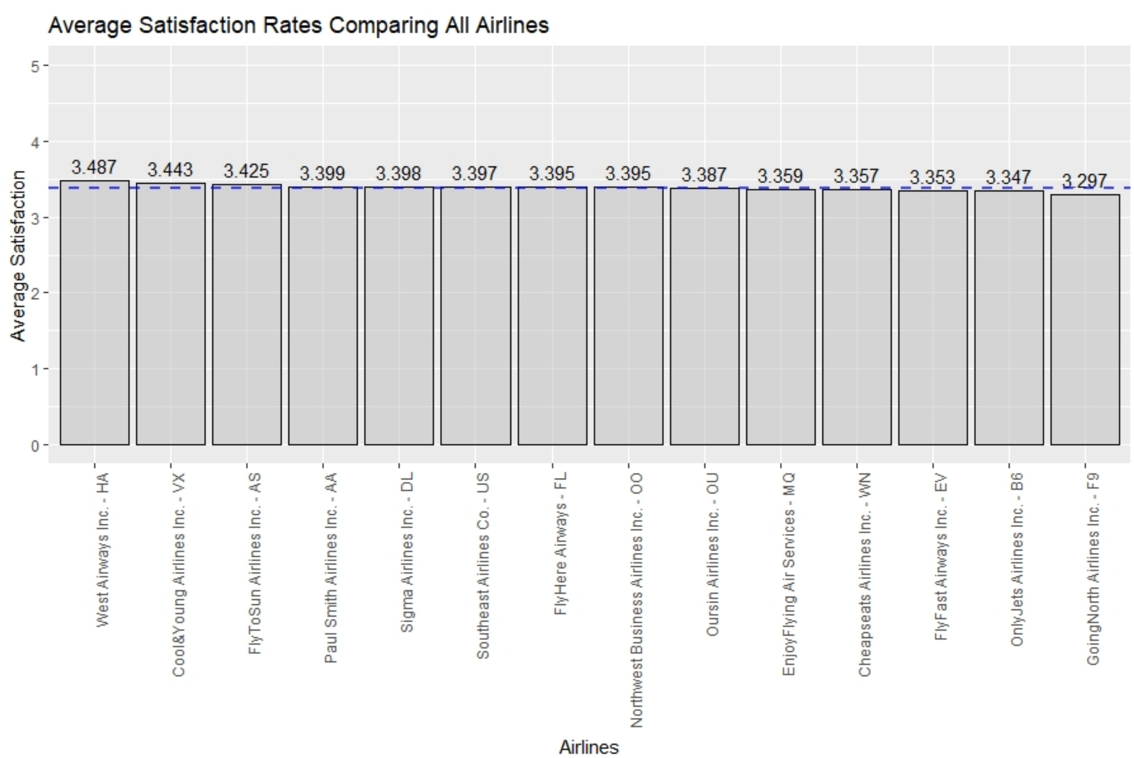


- VX has 1288 data records, the least data volume.

- VX has the second-highest satisfaction rate at 3.443, slightly above the industrial average at 3.379 (blue dashed line).



- VX has the second-lowest unsatisfied ratio at 45.26%, lower than the industrial average of 48.9%.



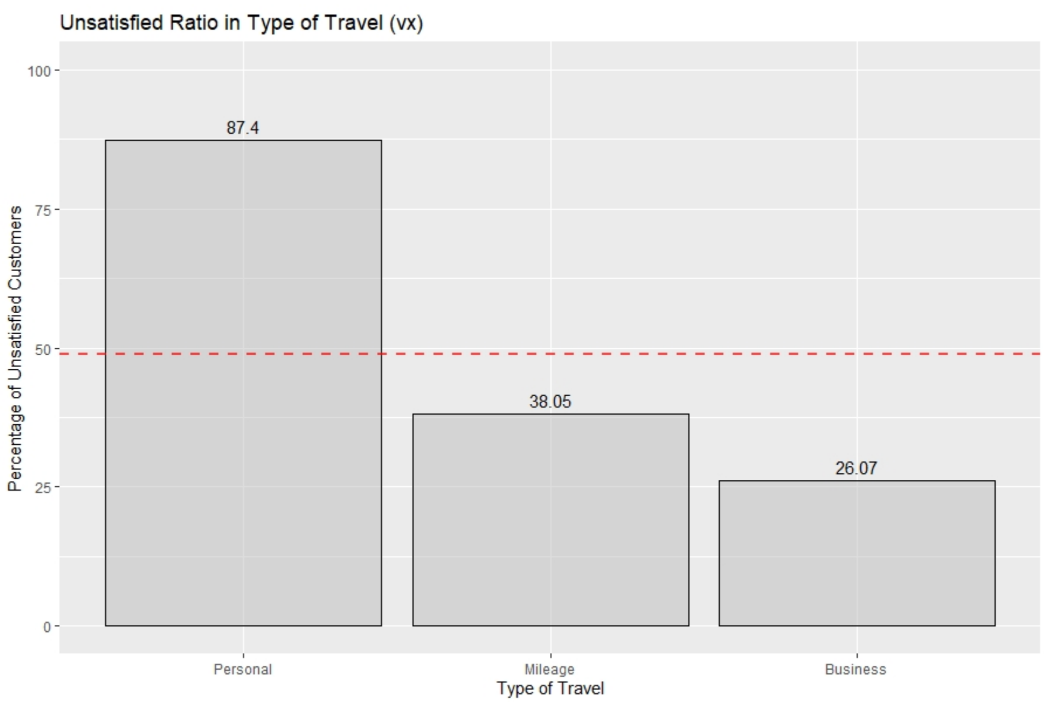
4) **<u>How does VX do in the above ranks?</u>**

- Overall, VX is doing a better job than most of its competitors in terms of customer satisfaction.

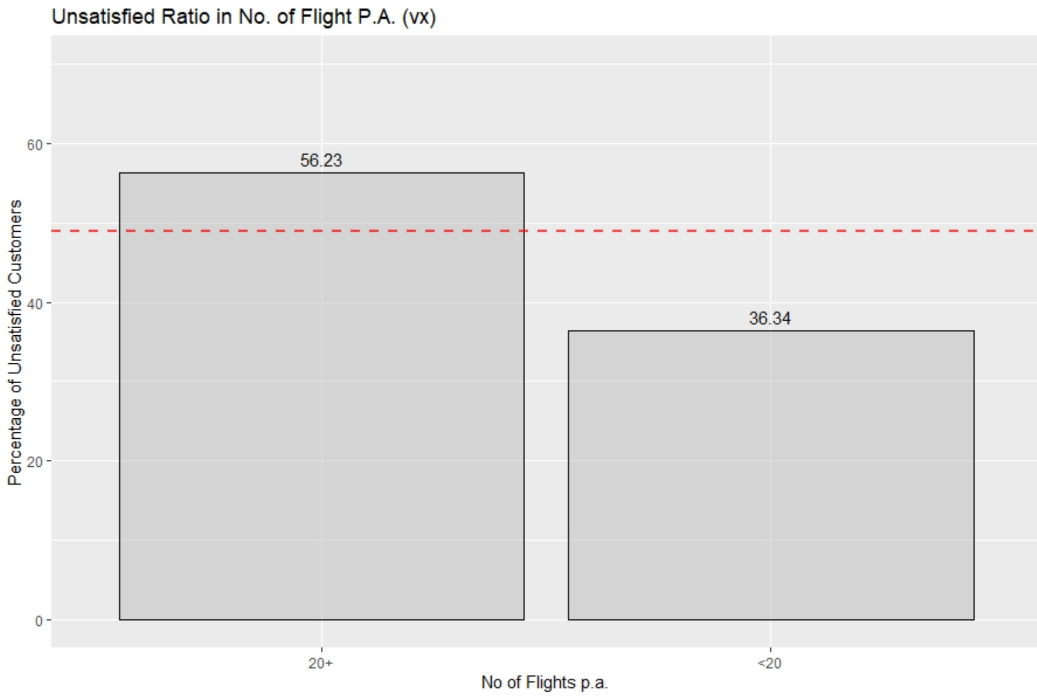
5) **<u>What are the potential characteristics of unsatisfied customers?</u>**

To answer this question, we ran a descriptive analysis of the related customer satisfaction or the unsatisfied ratio for each attribute, compared the results to the industrial average, which is our baseline benchmark. Before introduction of linear regression model, which is effective in testing the statistical significance of independent variables, we identified the following characteristics of customers who gave low satisfaction ratings for VX:

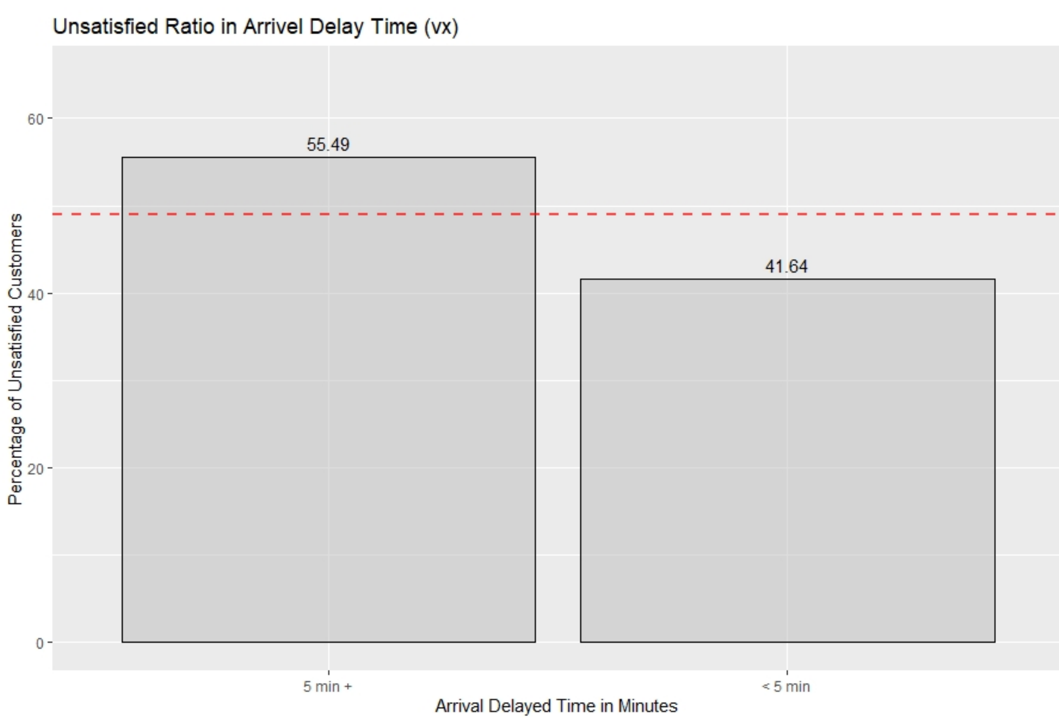
- <u>Customers travel for personal reasons</u> ("Type of Travel"). 87.4% of personal travelers are not satisfied



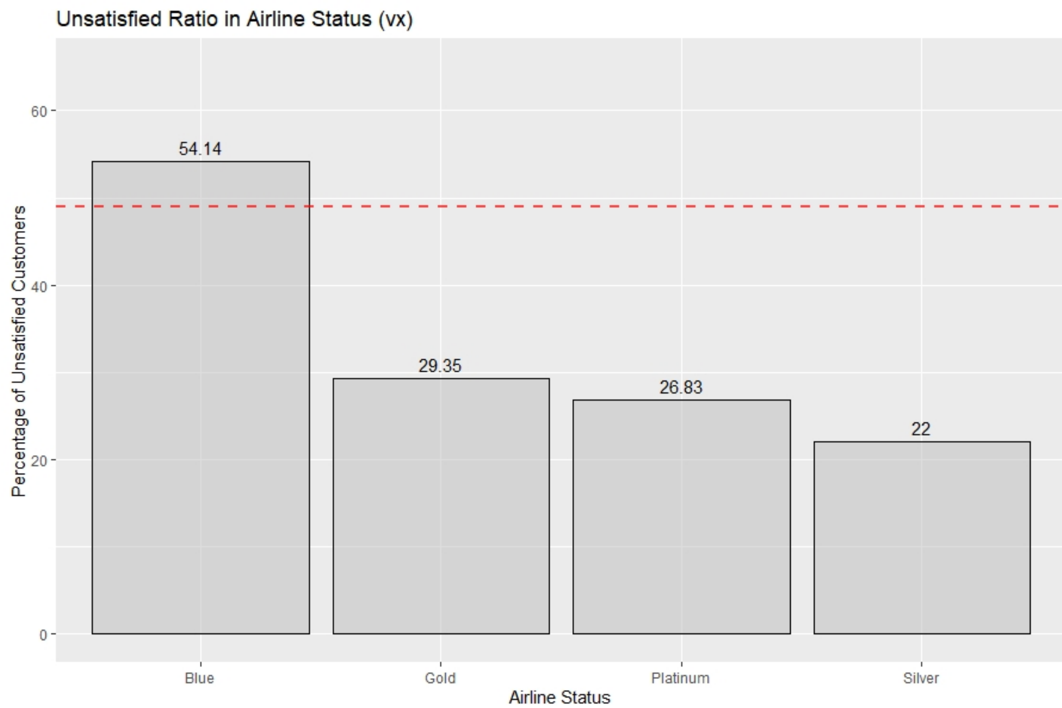
- <u>Customers with more than 20 past flights</u> ("No. of Flight p.a."). 56.23% of customers with 20+ past flights are not satisfied



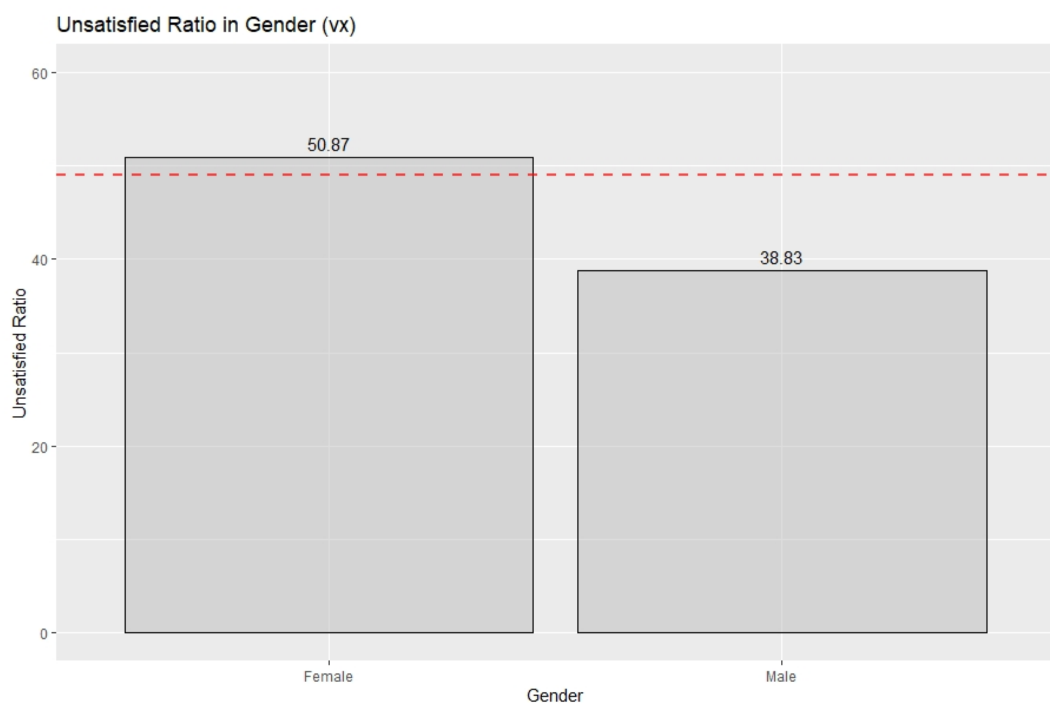
- <u>Customers experienced delay more than 5 minutes</u> ("Arrival Delay Time"). 55.49% of customers with 5+ min. delay is not satisfied



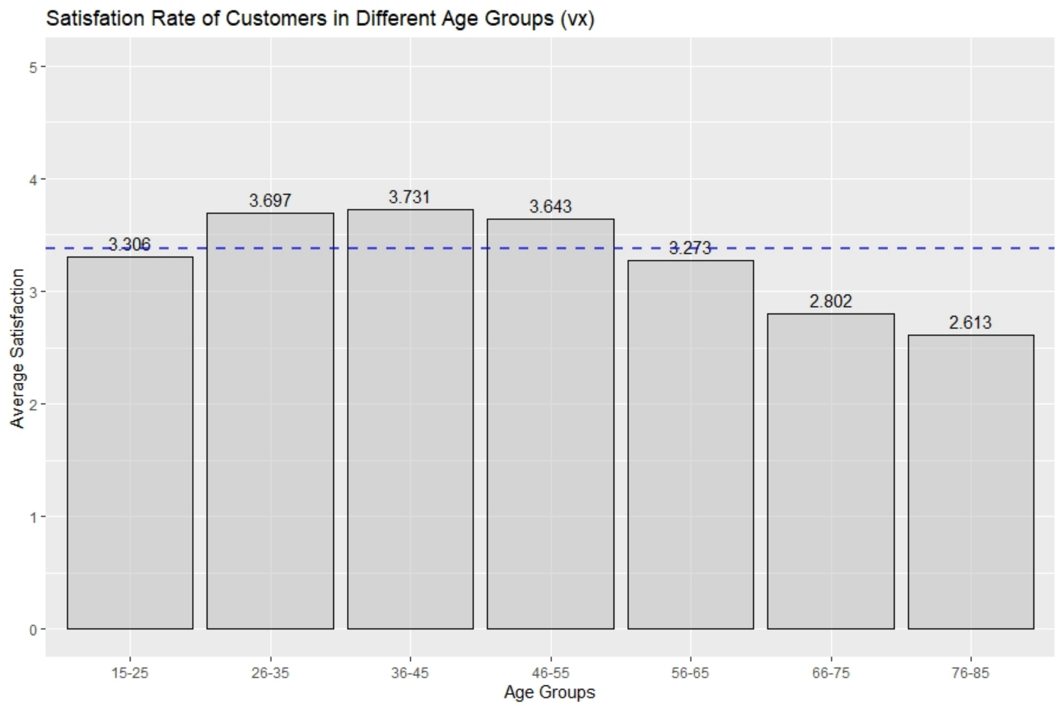
- Customers in Blue status (“Airline Status”). 54.14% of customers in blue status are not satisfied



- Female customers (“Gender”). 50.87% of female customers are not satisfied vs 42.2% of male customers



- <u>Customers who are either young or old.</u> ("Age"). The age group of 15-25 and 56+ gave lower than average satisfaction ratings



6) <u>What are the attributes that drive customer satisfaction?</u>

With help of linear regression modeling, the analytical team found the key attributes with statistical significance are "Airline Status", "Type of Travel", "Age", "Gender", and "No. of Flight p.a.". These 5 variables combined account for 42% of the variability of the VX survey satisfaction (see the Adjusted R-squared value in below linear model summary).

A summary of linear regression of satisfaction vs selected input variables for VX.

```
> summary(lm_vx)

Call:
lm(formula = satisfaction ~ ., data = sv_vx)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2294 -0.4663  0.2003  0.4866  2.5060 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.706e+00  1.560e-01  17.351  < 2e-16 ***
al_status2   5.946e-01  5.201e-02  11.432  < 2e-16 ***
al_status3   5.022e-01  8.011e-02   6.269 4.97e-10 ***
al_status4   6.917e-01  1.157e-01   5.976 2.97e-09 ***
age         -4.422e-03  1.420e-03  -3.115  0.00188 ** 
gender1      1.186e-01  4.186e-02   2.834  0.00467 ** 
sensitivity  8.900e-03  3.944e-02   0.226  0.82151    
fly_yrs      9.250e-03  6.810e-03   1.358  0.17463    
fly_pa      -4.061e-03  1.522e-03  -2.667  0.00775 ** 
fly_other    7.936e-02  2.647e-01   0.300  0.76441    
type2        8.829e-01  8.020e-02  11.008  < 2e-16 ***
type3        1.031e+00  4.994e-02  20.645  < 2e-16 ***
cards       -2.559e-02  2.036e-02  -1.257  0.20897    
shop        -5.170e-04  3.726e-04  -1.388  0.16545    
eat_drink   -2.011e-04  4.349e-04  -0.462  0.64394    
class2       1.914e-02  7.110e-02   0.269  0.78786    
class3      -9.428e-03  6.976e-02  -0.135  0.89252    
days2       -3.889e-02  7.532e-02  -0.516  0.60572    
days3        8.878e-02  7.359e-02   1.206  0.22789    
days4        4.489e-02  7.398e-02   0.607  0.54409    
days5        3.261e-02  7.397e-02   0.441  0.65945    
days6        1.619e-02  8.257e-02   0.196  0.84463    
days7       -1.895e-02  7.482e-02  -0.253  0.80006    
cancel1     -9.178e-01  7.229e-01  -1.270  0.20447    
fly_x       -3.748e-08  5.652e-08  -0.663  0.50729    
delay       -2.084e-03  2.552e-03  -0.817  0.41431    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7136 on 1262 degrees of freedom
Multiple R-squared:  0.4332,	Adjusted R-squared:  0.4219 
F-statistic: 38.58 on 25 and 1262 DF,  p-value: < 2.2e-16
```

This finding confirms and refines our descriptive analysis result, which excludes "Arrival Delay Time", a variable that is not statistically significant.

A similar inspection was performed on the origin and destination cities, with results showing very high P values, indicating no statistically significant results. The studies of these variables will be covered later in the section of Descriptive Analysis within this report.

7) **<u>Can the customer satisfaction rating be predicted?</u>**

The short answer is yes, but not perfect. The analytical team explored, tested, and refined 3 different predictive models by applying techniques of linear multiple regression (the linear model), support vector machine with Naïve Bayes algorithm for classification prediction (the NB model), and support vector machine with ksvm algorithm for value prediction (the ksvm model).

The NB model predicts whether a customer is satisfied or not satisfied, and it yields an accuracy at 77%, while the linear model yields a root mean square error (RMSE) at 0.71, and the ksvm model yields a RMSE at 0.93. The details of predictive modeling will be elaborated later in the section of Predictive Models in this report.

Data Preparation

DATA ACQUISITION

The original survey data was distributed to the analytical team within an Excel Spreadsheet, at the beginning of this academic term. This original data was read into R and stored in a data frame called "AirSurvey" for later use. A quick inspection of the data frame reveals 129,889

```
> str(AirSurvey)
Classes 'tbl_df', 'tbl' and 'data.frame':	129889 obs. of  28 variables:
 $ Satisfaction                  : num  4.5 4 2.5 4 5 5 3.5 4 4 4 ...
 $ Airline Status                : chr  "Blue" "Blue" "Blue" "Blue" ...
 $ Age                           : num  31 56 21 43 49 49 35 33 44 51 ...
 $ Gender                        : chr  "Male" "Male" "Female" "Male" ...
 $ Price Sensitivity             : num  1 2 2 1 1 1 1 1 1 1 ...
 $ Year of First Flight          : num  2007 2006 2006 2007 2006 ...
 $ No of Flights p.a.            : num  28 41 8 9 14 0 15 4 8 12 ...
 $ % of Flight with other Airlines: num  7 3 7 9 10 4 5 17 6 7 ...
 $ Type of Travel                : chr  "Business travel" "Business travel" "Personal Travel" "Business travel" ...
 $ No. of other Loyalty Cards    : num  2 0 0 2 0 1 0 2 0 0 ...
 $ Shopping Amount at Airport    : num  0 15 0 10 8 0 0 0 0 25 ...
 $ Eating and Drinking at Airport : num  75 60 135 45 26 65 60 90 90 80 ...
 $ Class                         : chr  "Business" "Business" "Business" "Eco" ...
 $ Day of Month                  : num  18 11 25 20 25 16 6 5 21 19 ...
 $ Flight date                   : POSIXct, format: "2014-03-18" "2014-01-11" "2014-01-25" "2014-02-20" ...
 $ Airline Code                  : chr  "MQ" "MQ" "MQ" "MQ" ...
 $ Airline Name                  : chr  "EnjoyFlying Air Services" "EnjoyFlying Air Services" "EnjoyFlying Air Servi
 $ Orgin City                    : chr  "Madison, WI" "Madison, WI" "Milwaukee, WI" "Madison, WI" ...
 $ Origin State                  : chr  "Wisconsin" "Wisconsin" "Wisconsin" "Wisconsin" ...
 $ Destination City              : chr  "Dallas/Fort Worth, TX" "Dallas/Fort Worth, TX" "Dallas/Fort Worth, TX" "Dal
 $ Destination State             : chr  "Texas" "Texas" "Texas" "Texas" ...
 $ Scheduled Departure Hour      : num  15 11 12 11 12 18 6 18 12 18 ...
 $ Departure Delay in Minutes    : num  0 2 34 26 0 0 0 0 0 0 ...
 $ Arrival Delay in Minutes      : num  3 5 14 39 0 0 0 1 0 0 ...
 $ Flight cancelled              : chr  "No" "No" "No" "No" ...
 $ Flight time in minutes        : num  134 120 122 141 144 123 119 138 114 118 ...
 $ Flight Distance               : num  821 821 853 821 853 821 821 821 853 821 ...
 $ Arrival Delay greater 5 Mins  : chr  "no" "no" "yes" "yes" ...
```

A summary of the original data structure.

observations (rows) and 28 variables (columns), containing numeric, characters, and date values in 'tbl', 'tbl.df', and 'data.frame' classes.

A summary of each variable in the original data.

```
> summary(AirSurvey)
  Satisfaction   Airline Status          Age           Gender          Price Sensitivity
 Min.   :1.000   Length:129889      Min.   :15.0   Length:129889      Min.   :0.000
 1st Qu.:3.000   Class :character   1st Qu.:33.0   Class :character   1st Qu.:1.000
 Median :4.000   Mode  :character   Median :45.0   Mode  :character   Median :1.000
 Mean   :3.379                      Mean   :46.2                      Mean   :1.276
 3rd Qu.:4.000                      3rd Qu.:59.0                      3rd Qu.:2.000
 Max.   :5.000                      Max.   :85.0                      Max.   :5.000
 NA's   :3
 Year of First Flight No of Flights p.a. % of Flight with other Airlines Type of Travel
 Min.   :2003         Min.   :  0.00     Min.   :  1.000                Length:129889
 1st Qu.:2004         1st Qu.:  9.00     1st Qu.:  4.000                Class :character
 Median :2007         Median : 17.00     Median :  7.000                Mode  :character
 Mean   :2007         Mean   : 20.08     Mean   :  9.314
 3rd Qu.:2010         3rd Qu.: 29.00     3rd Qu.: 10.000
 Max.   :2012         Max.   :100.00     Max.   :110.000

 No. of other Loyalty Cards Shopping Amount at Airport Eating and Drinking at Airport    Class
 Min.   : 0.0000            Min.   :  0.00             Min.   :  0.00                 Length:129889
 1st Qu.: 0.0000            1st Qu.:  0.00             1st Qu.: 30.00                 Class :character
 Median : 0.0000            Median :  0.00             Median : 60.00                 Mode  :character
 Mean   : 0.8838            Mean   : 26.55             Mean   : 68.24
 3rd Qu.: 2.0000            3rd Qu.: 30.00             3rd Qu.: 90.00
 Max.   :12.0000            Max.   :879.00             Max.   :895.00

  Day of Month    Flight date                    Airline Code       Airline Name        Orgin City
 Min.   : 1.00   Min.   :2014-01-01 00:00:00   Length:129889      Length:129889      Length:129889
 1st Qu.: 8.00   1st Qu.:2014-01-24 00:00:00   Class :character   Class :character   Class :character
 Median :16.00   Median :2014-02-17 00:00:00   Mode  :character   Mode  :character   Mode  :character
 Mean   :15.72   Mean   :2014-02-15 13:29:25
 3rd Qu.:23.00   3rd Qu.:2014-03-10 00:00:00
 Max.   :31.00   Max.   :2014-03-31 00:00:00

 Origin State       Destination City   Destination State  Scheduled Departure Hour
 Length:129889      Length:129889      Length:129889      Min.   : 1.00
 Class :character   Class :character   Class :character   1st Qu.: 9.00
 Mode  :character   Mode  :character   Mode  :character   Median :13.00
                                                          Mean   :12.99
                                                          3rd Qu.:17.00
                                                          Max.   :23.00

 Departure Delay in Minutes Arrival Delay in Minutes Flight cancelled   Flight time in minutes
 Min.   :   0.00            Min.   :   0.00          Length:129889      Min.   :  8.0
 1st Qu.:   0.00            1st Qu.:   0.00          Class :character   1st Qu.: 59.0
 Median :   0.00            Median :   0.00          Mode  :character   Median : 92.0
 Mean   :  14.98            Mean   :  15.37                             Mean   :111.5
 3rd Qu.:  13.00            3rd Qu.:  13.00                             3rd Qu.:142.0
 Max.   :1592.00            Max.   :1584.00                             Max.   :669.0
 NA's   :2345               NA's   :2738                                NA's   :2738
 Flight Distance  Arrival Delay greater 5 Mins
 Min.   :  31.0   Length:129889
 1st Qu.: 362.0   Class :character
 Median : 630.0   Mode  :character
 Mean   : 793.8
 3rd Qu.:1024.0
 Max.   :4983.0
```

DATA CLEANSING

The data cleansing process started with understanding the "existing condition" of the data itself. Inspected with the str() & summary() functions, the original data shows issues with NAs in 4 variables, inconsistent naming and letter capitalization, typos, and data that is not desired for the analysis. The main goal in this data cleansing process is dealing with these NAs.

There are 4 variables containing NAs, 3 NAs in 'Satisfaction', 2345 NAs in 'Departure Delay in Minutes', 2738 NAs in 'Arrival Delay in Minutes', and 2738 NAs in 'Flight time in minutes'. By looking closely at some of the typical rows with NAs, the analytical team found the 3 NAs in 'Satisfaction' are missing values, the NAs in the other 3 variables have a close relationship with the variable 'Flight canceled.' It seems reasonable that the majority of NAs in those 3 variables are actually from the 2401 canceled flights, leaving 337 of the NAs in both 'Arrival Delay in Minutes' and 'Flight time in minutes' as missing values. Therefore, the analytical team decided to remove the rows with missing data and convert the NAs associated with canceled flights into 0.

DATA TRANSFORMING

The data transforming process includes actions that convert data values into the desired format, renames columns and rows for consistency and easy access, drops undesired variables, adds transformed variables derived from the original data, and eventually, when necessary, creates new data frames that are easy to use for the following analysis. Below is a specific list of items performed to the data set:

- <u>Values converted</u>:
  - 'Airline Status', 'Gender', 'Type of Travel', 'Class', 'Flight canceled' are converted into Factors.
- <u>Columns renamed</u>:
  - All columns are renamed with abbreviations in lower case for easier access in later analysis.
- <u>Transformed variables</u>:
  - 'Flight date' converted into days of the week and then transformed into Factors 1 to 7 representing Monday to Sunday.

- ‘Airline Code’ and ‘Airline Name’ are combined into the ‘als’ column, which stands for ‘airlines’.

- Variables dropped:
  - ‘Day of Month’, ‘Scheduled Departure Hour’, and ‘Arrival Delay greater 5 Mins’ are dropped as they are undesired or redundant for the later analysis
- New data frame:
  - At the end of the data transforming process, a new data frame containing only the records of VX is created for further analysis.

A summary of cleansed and transformed data structure.

```
> str(survey)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':       129549 obs. of  24 variables:
 $ satisfaction: num  4.5 4 2.5 4 5 5 3.5 4 4 4 ...
 $ al_status   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 2 3 3 2 1 1 ...
 $ age         : num  31 56 21 43 49 49 35 33 44 51 ...
 $ gender      : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 2 1 1 ...
 $ sensitivity : num  1 2 2 1 1 1 1 1 1 1 ...
 $ fly_yrs     : num  12 13 13 12 13 9 8 9 16 14 ...
 $ fly_pa      : num  28 41 8 9 14 0 15 4 8 12 ...
 $ fly_other   : num  0.07 0.03 0.07 0.09 0.1 0.04 0.05 0.17 0.06 0.07 ...
 $ type        : Factor w/ 3 levels "1","2","3": 3 3 1 3 3 3 3 3 3 3 ...
 $ cards       : num  2 0 0 2 0 1 0 2 0 0 ...
 $ shop        : num  0 15 0 10 8 0 0 0 0 25 ...
 $ eat_drink   : num  75 60 135 45 26 65 60 90 90 80 ...
 $ class       : Factor w/ 3 levels "1","2","3": 3 3 3 1 1 1 1 1 1 1 ...
 $ days        : Factor w/ 7 levels "1","2","3","4",..: 2 6 6 4 2 4 4 3 2 7 ...
 $ delay_dept  : num  0 2 34 26 0 0 0 0 0 0 ...
 $ delay_arvl  : num  3 5 14 39 0 0 0 1 0 0 ...
 $ cancel      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ fly_time    : num  134 120 122 141 144 123 119 138 114 118 ...
 $ fly_dist    : num  821 821 853 821 853 821 821 821 853 821 ...
 $ origin_city : chr  "madison, wi" "madison, wi" "milwaukee, wi" "madison, wi" ...
 $ origin_state: chr  "wisconsin" "wisconsin" "wisconsin" "wisconsin" ...
 $ destin_city : chr  "dallas/fort worth, tx" "dallas/fort worth, tx" "dallas/fort worth, tx"
 $ destin_state: chr  "texas" "texas" "texas" "texas" ...
 $ als         : chr  "EnjoyFlying Air Services - MQ" "EnjoyFlying Air Services - MQ" "EnjoyFl
```

A summary of cleansed and transformed data structure containing only records of VX.

```
> str(sv_vx)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':      1288 obs. of  19 variables:
 $ satisfaction: num  4 4 4 4 2 3 5 4 4 5 ...
 $ al_status   : Factor w/ 4 levels "1","2","3","4": 1 2 3 3 1 2 2 1 1 2 ...
 $ age         : num  34 48 51 43 74 80 35 30 38 39 ...
 $ gender      : Factor w/ 2 levels "0","1": 2 1 1 2 1 2 2 2 2 2 ...
 $ sensitivity : num  1 1 1 1 1 2 1 1 1 1 ...
 $ fly_yrs     : num  13 10 16 15 10 15 10 16 13 16 ...
 $ fly_pa      : num  30 10 10 29 33 29 21 4 16 18 ...
 $ fly_other   : num  0.01 0.07 0.12 0.09 0.02 0.03 0.02 0.13 0.1 0.35 ...
 $ type        : Factor w/ 3 levels "1","2","3": 3 3 3 2 1 1 3 3 3 3 ...
 $ cards       : num  0 1 1 0 0 0 2 3 2 1 ...
 $ shop        : num  0 60 25 0 0 0 15 0 0 45 ...
 $ eat_drink   : num  15 15 75 90 16 120 110 30 75 105 ...
 $ class       : Factor w/ 3 levels "1","2","3": 3 3 3 3 3 3 1 1 1 1 ...
 $ days        : Factor w/ 7 levels "1","2","3","4",..: 3 7 1 2 7 6 2 5 7 3 ...
 $ delay_dept  : num  57 0 0 0 0 0 1 27 0 0 ...
 $ delay_arvl  : num  71 0 0 0 0 0 0 56 0 5 ...
 $ cancel      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ fly_time    : num  123 128 130 98 91 98 104 150 98 117 ...
 $ fly_dist    : num  679 954 954 679 679 679 679 954 679 679 ...
```

A summary of each variable in the cleansed and transformed data.

```
> summary(survey)
  satisfaction   al_status      age         gender     sensitivity       fly_yrs          fly_pa      
 Min.   :1.000   1:88680   Min.   :15.0   0:73171   Min.   :0.000   Min.   : 7.00   Min.   :  0.00  
 1st Qu.:3.000   2:25904   1st Qu.:33.0   1:56378   1st Qu.:1.000   1st Qu.: 9.00   1st Qu.:  9.00  
 Median :4.000   3:10802   Median :45.0             Median :1.000   Median :12.00   Median : 17.00  
 Mean   :3.379   4: 4163   Mean   :46.2             Mean   :1.276   Mean   :11.79   Mean   : 20.08  
 3rd Qu.:4.000             3rd Qu.:59.0             3rd Qu.:2.000   3rd Qu.:15.00   3rd Qu.: 29.00  
 Max.   :5.000             Max.   :85.0             Max.   :5.000   Max.   :16.00   Max.   :100.00  

   fly_other        type         cards             shop          eat_drink        class         days    
 Min.   :0.01000   1:40089   Min.   : 0.0000   Min.   :  0.00   Min.   :  0.00   1:105467   1:19715  
 1st Qu.:0.04000   2:10051   1st Qu.: 0.0000   1st Qu.:  0.00   1st Qu.: 30.00   2: 13563   2:17175  
 Median :0.07000   3:79409   Median : 0.0000   Median :  0.00   Median : 60.00   3: 10519   3:18910  
 Mean   :0.09314             Mean   : 0.8838   Mean   : 26.56   Mean   : 68.25              4:19550  
 3rd Qu.:0.10000             3rd Qu.: 2.0000   3rd Qu.: 30.00   3rd Qu.: 90.00              5:19878  
 Max.   :1.10000             Max.   :12.0000   Max.   :879.00   Max.   :895.00              6:15799  
                                                                                            7:18522  
   delay_dept        delay_arvl        cancel        fly_time        fly_dist      origin_city       
 Min.   :   0.00   Min.   :   0.00   0:127156   Min.   :  0.0   Min.   :  31.0   Length:129549     
 1st Qu.:   0.00   1st Qu.:   0.00   1:  2393   1st Qu.: 57.0   1st Qu.: 363.0   Class :character  
 Median :   0.00   Median :   0.00              Median : 91.0   Median : 630.0   Mode  :character  
 Mean   :  14.72   Mean   :  15.04              Mean   :109.3   Mean   : 794.7                     
 3rd Qu.:  12.00   3rd Qu.:  13.00              3rd Qu.:141.0   3rd Qu.:1024.0                     
 Max.   :1592.00   Max.   :1584.00              Max.   :669.0   Max.   :4983.0                     

 origin_state       destin_city        destin_state           als           
 Length:129549      Length:129549      Length:129549      Length:129549     
 Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character  
```

DATA MUNGING

Most of our efforts in data munging took place in the descriptive analysis phase. Each different analytical task requires specific values which necessitates further tweaks, such as adding, deleting, converting, or even re-creating the data set. In general, most munging work involved steps to include, pulling desired data in categories or groups, calculating count, mean, and summation of values, sorting these derived values in ascending or descending orders, and finally creating graphs and plots for visual output. This process worked well in generating bar plots, uncovering patterns of unsatisfied customers ratios in personal and flight-related attributes, when comparing VX data and the overall data.

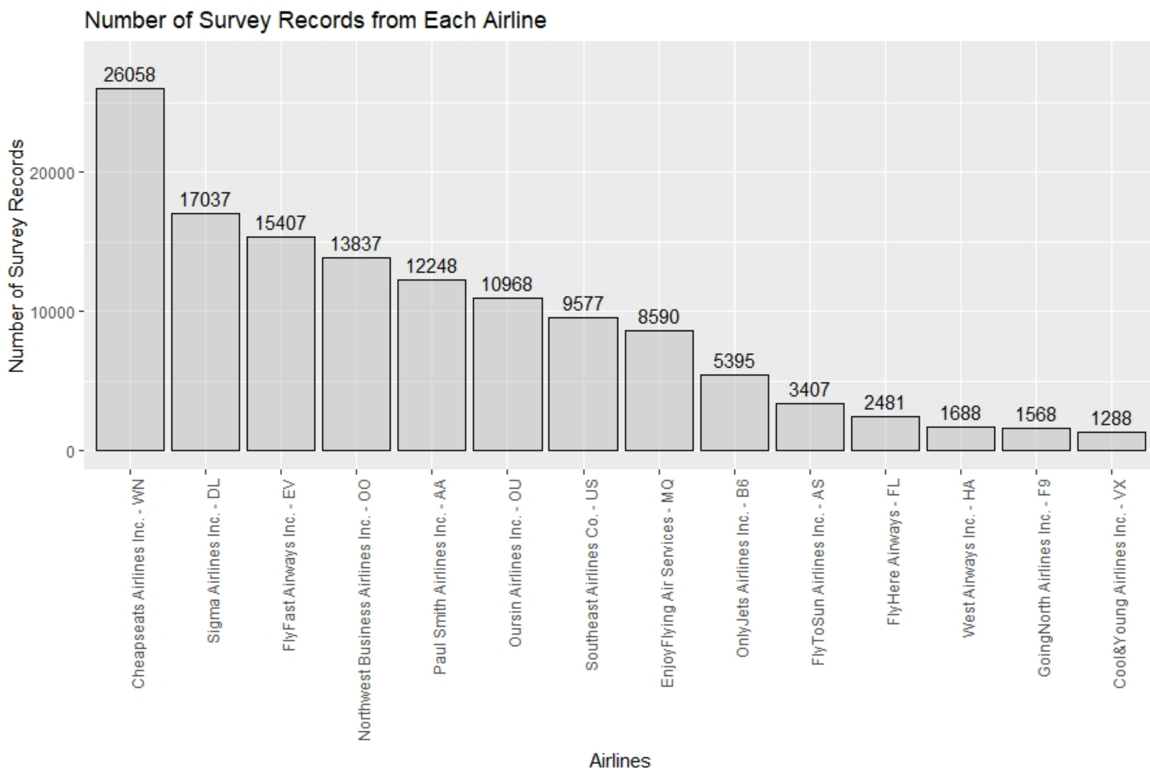
When examining location-related data, 'Origin City' and 'Destination City', the analytical team adapted the Nominatim API tool to retrieve geo-coordinates for data mapping. Some challenges we faced included: multiple location names in individual records, large number of cities with a wide range in flight numbers, and most importantly, deciding the mechanism that measures locations relevant to customer satisfaction. As the team decided to test whether there was any correlation in volumes of flight, canceled flights, and accumulated delay minutes to customer satisfaction for origin and destination cities, a process of grouping, converting, categorizing, and value-computing were tightly integrated and applied in order to mung the data.

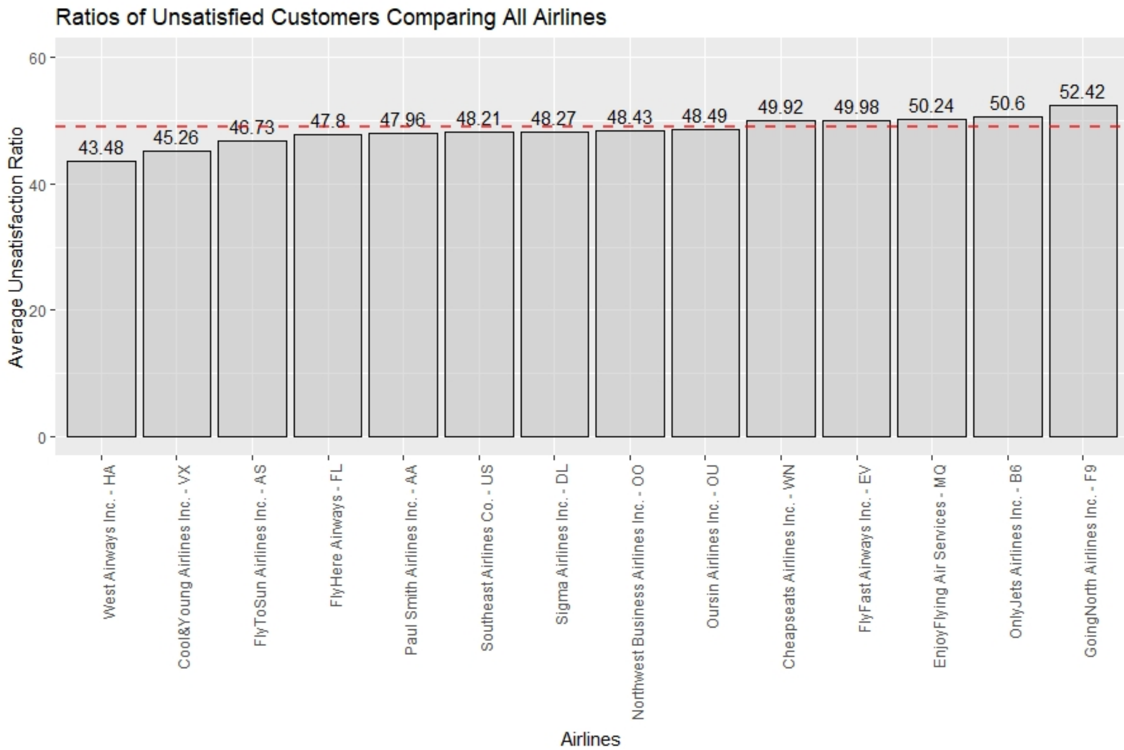
For detailed methods, steps, and codes involved with data preparation, please refer to the appendix – data preparation: importing, cleansing, and munging.

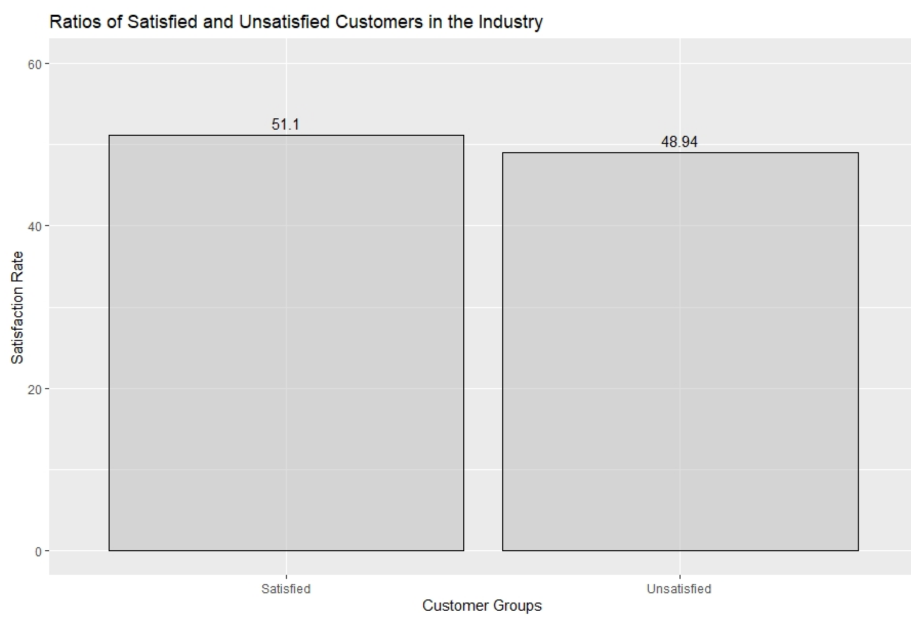
Descriptive Statistics

The data contains 28 variables and a total of 129,889 observations, offering abundant information at a macro level for a wide range of possible analytical activities. The analytical method that the investigative team adopted emphasizes on a top-down, large-to-small, and overall-to-specific approach. Thus, the team started by looking at the broad picture of the industry.

The charts below represent an overview of the overall airline industry. The target airline, Cool & Young (VX), has the lowest number of surveys records as compared with the other airlines. This is a challenge for the analytical team since fewer data makes it more difficult to determine variable influence and correlation. Using creative modeling and the descriptive analysis of the overall data of the industry as a comparison, the team later identified key attributes in correlation with customer satisfaction.







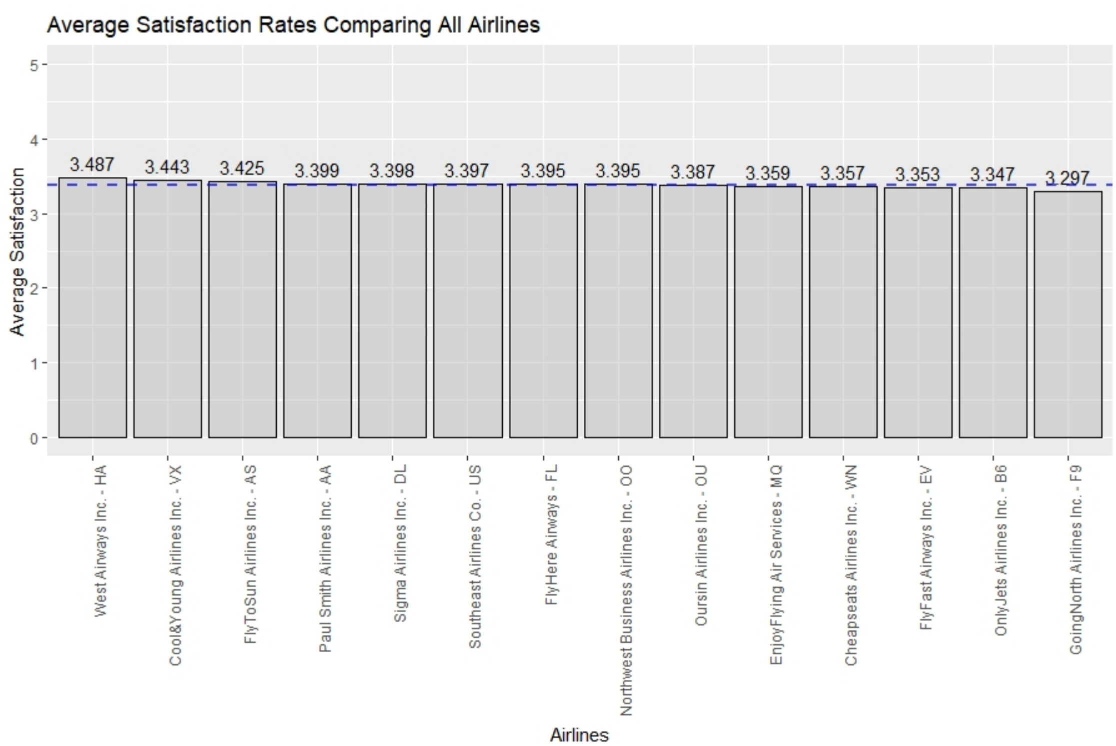
After looking at the number of survey records of each airline, the analytical team found the ratio of overall satisfied and unsatisfied customers in the industry are quite equal at 51.1% to 48.9%, assuming 1-3 as unsatisfied and 3.5-5 as satisfied. This indicates a neutral and less biased survey. These two numbers then became the baseline benchmarks for further analysis.

Given the industrial benchmarks, the analytical team compared both the unsatisfied ratio and the average satisfaction rating among individual airlines. The results of these two comparisons are considered congruent in two points:

1. Higher average satisfaction seems to be associated with low unsatisfied ratio;
2. The above-average satisfaction ratings seem to be associated with the below-average unsatisfied ratios.

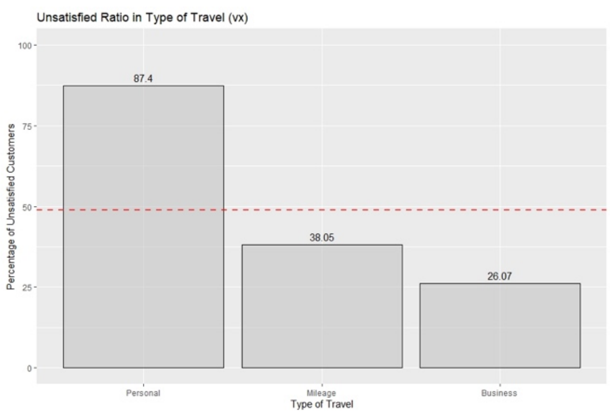
When comparing with other 13 airlines and the industrial benchmarks, overall VX is providing better services for their customers, reflected in their higher than average satisfaction rating and lower than benchmarked unsatisfied ratio.

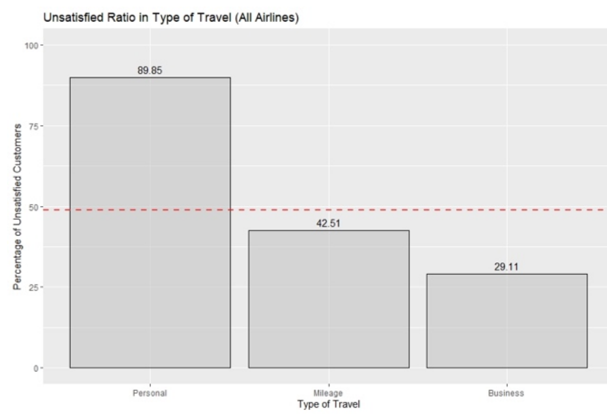
Having a broad picture of the industry is very helpful for the analytical team to immerse in the situation and keep brainstorming potential methods and specific analytical processes to identify key attributes correlate to customer satisfaction. After iterations of attempts and failure, <u>a group of variables was selected as patterns emerged, which outlines the potential "congregation" of the unsatisfied customers.</u>



1. Type of Travel

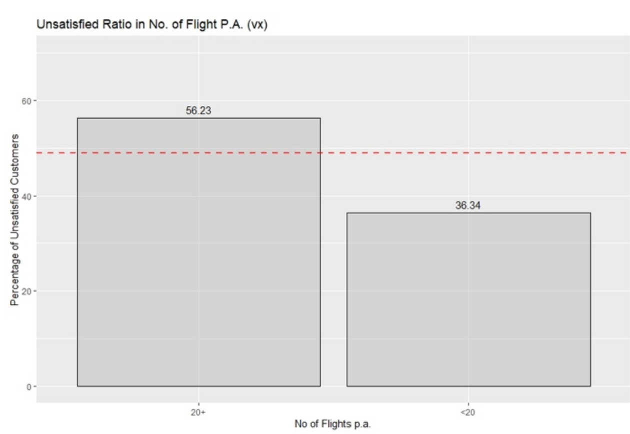
This survey contains 3 types of travels: personal, mileage, and business. The comparison study of both the VX data and the overall data shows a matched pattern that the ratio of unsatisfied personal travelers is almost twice as the industrial baseline, 87.4% vs 48.9%. While the unsatisfied ratios of customers in mileage and business travels are below the industrial baseline.

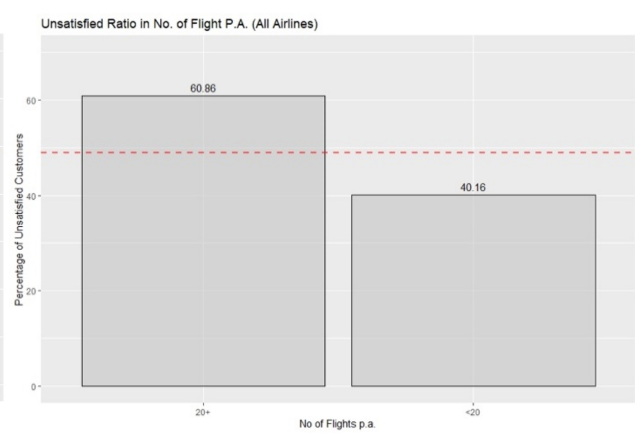




2. No. of Flight p.a.

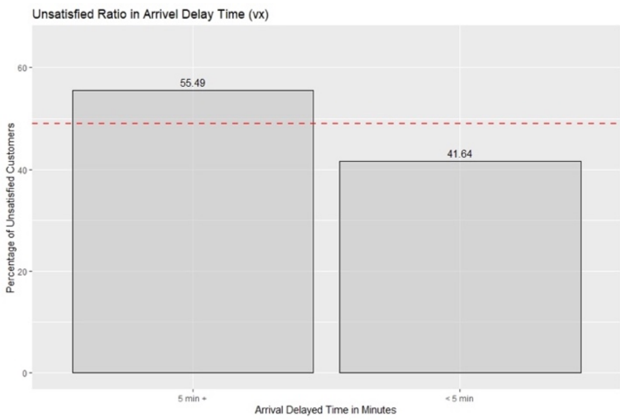
It is significant to take the number of past flights into consideration when evaluating customer satisfaction. The comparison study below shows the unsatisfied ratio higher than the industrial average, indicating that the more flights travelers take in the past, the harder to get them satisfied. It may due to the reason that customers become experienced and fussier.

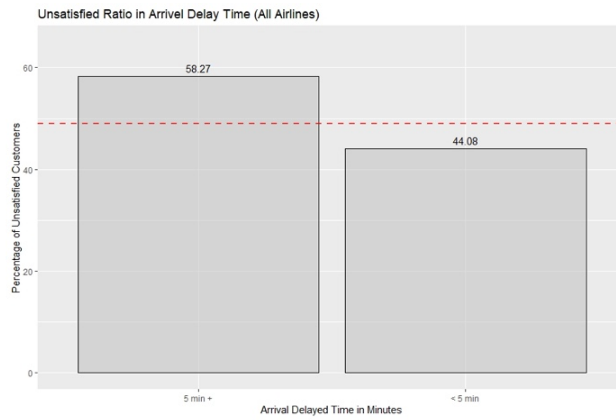




3. Arrival Delay Time

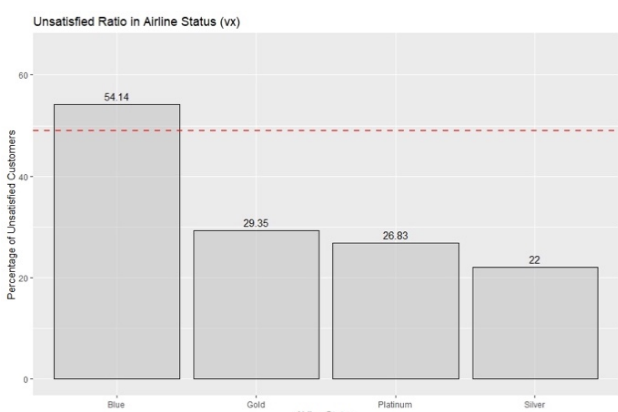
It is clear in the visuals that customers become dissatisfied even after just five minutes or more arrival delay, and the pattern from the study of VX data and the overall data is congruent. In the real-world situation, a traveler planned to arrive at a certain time and then the flight was delayed. This could have caused the customer to have to change their plans or miss a meeting or personal engagement. It is logical to hypothesize that a delay could affect customer satisfaction.

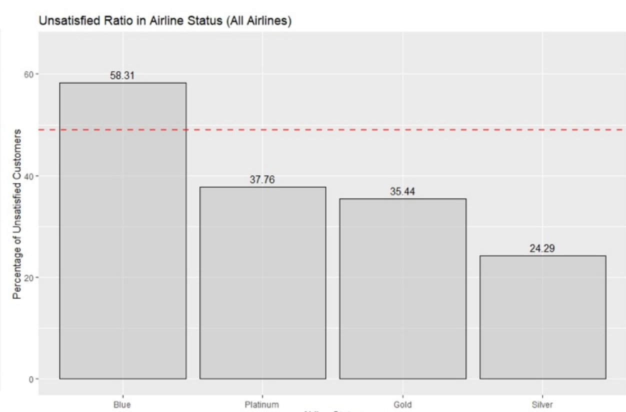




4. Airline Status

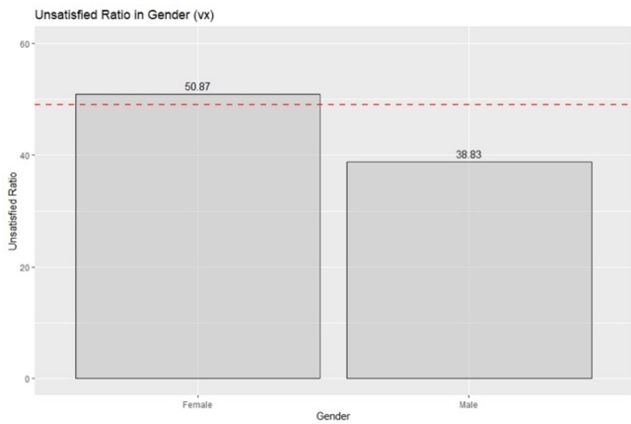
Examining these two visuals around airline status initiates some thinking around who the unsatisfied customers are. Airlines status is what a level at which a loyal cardholder is traveling at. The blue level traveler is the lowest airline status and they tend to be more unsatisfied. There is not a significant difference in satisfaction between the upper echelon statuses, gold, and platinum. Silver status travelers are the most satisfied customers.

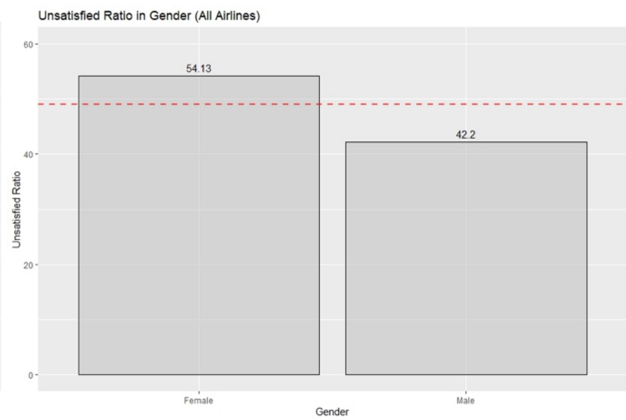




5. Gender

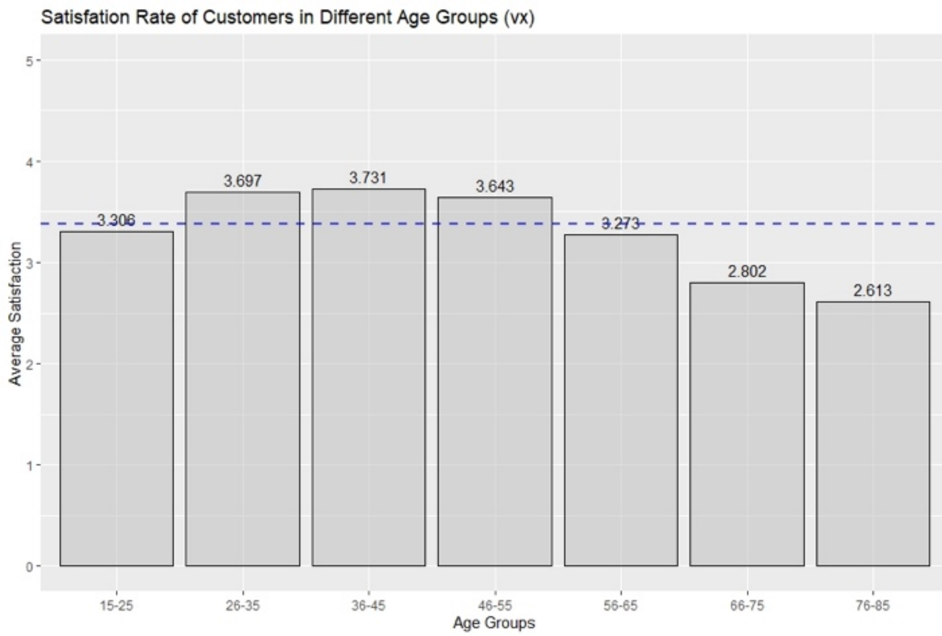
Gender is a significant variable to examine in order to determine customer satisfaction. The satisfaction rating of VX as determined by the gender variable is like that of the overall airline industry for gender. Females tend to be more unsatisfied with their airline travel experience as compared with their traveling male counterparts.

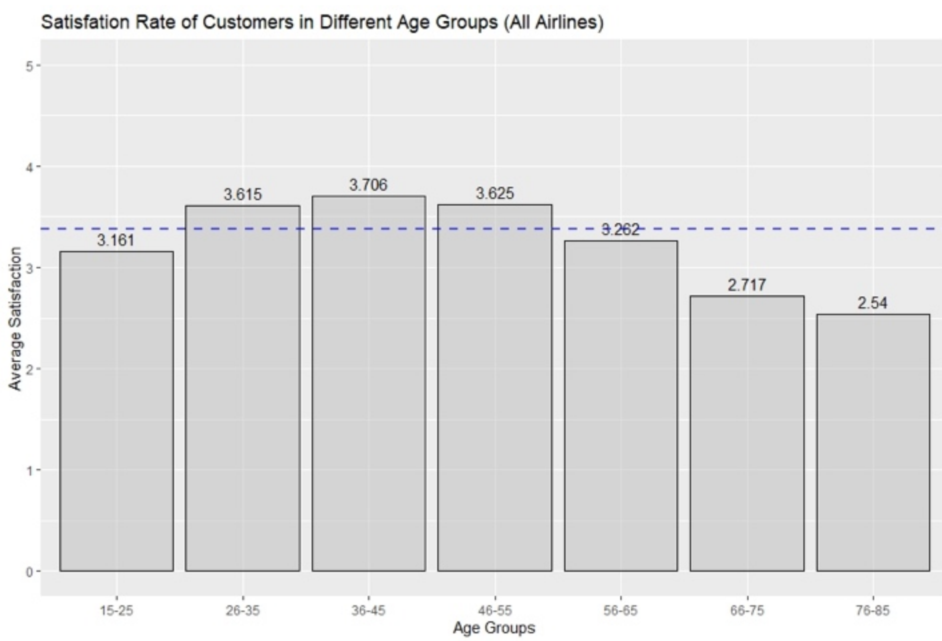




6. <u>Age Groups:</u>

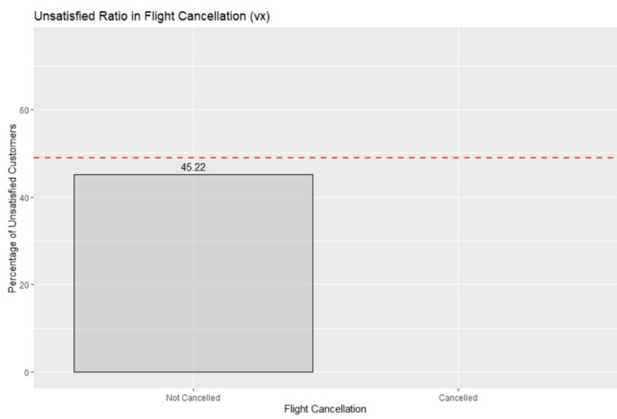
Age is an interesting variable to draw conclusions from and it is also consistent across all airlines and VX specifically. Younger people under 25 are not as satisfied with their airline travel experience as compared to the middle age groups. The visual output of the data detracts from the norm when examining the older age groups. Across the airline industry to include VX, the most satisfied customers are in the age group of 36-45. As people age, there is a trend that they will become increasingly more dissatisfied with their airline travel experience.

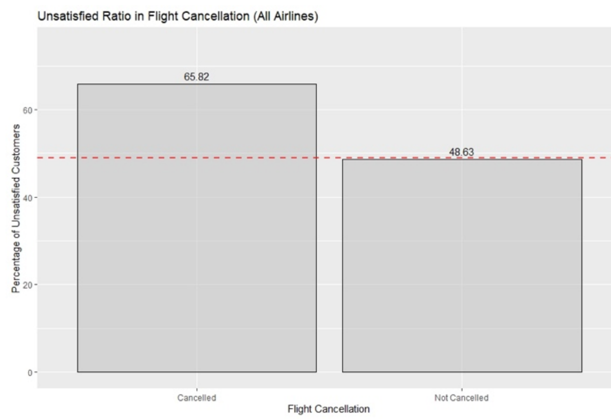




Efforts were made for inspection of other variables through the analysis process. However, either there is a <u>limitation of VX data itself</u>, or there is <u>no obvious pattern indicating outliers for customer satisfaction or dissatisfaction</u>.

One good example of the challenge of limitation of data is the flight cancelation attribute in VX data. The analytical team assumed that a canceled flight would be associated with the higher unsatisfaction ratio. At the industry-wide level, the analytical result confirms this assumption. As this variable relates to VX no good specific conclusions could be drawn because there is only one canceled flight in the 1288 records of VX. Therefore, the flight cancelation is excluded for further analysis in VX data.

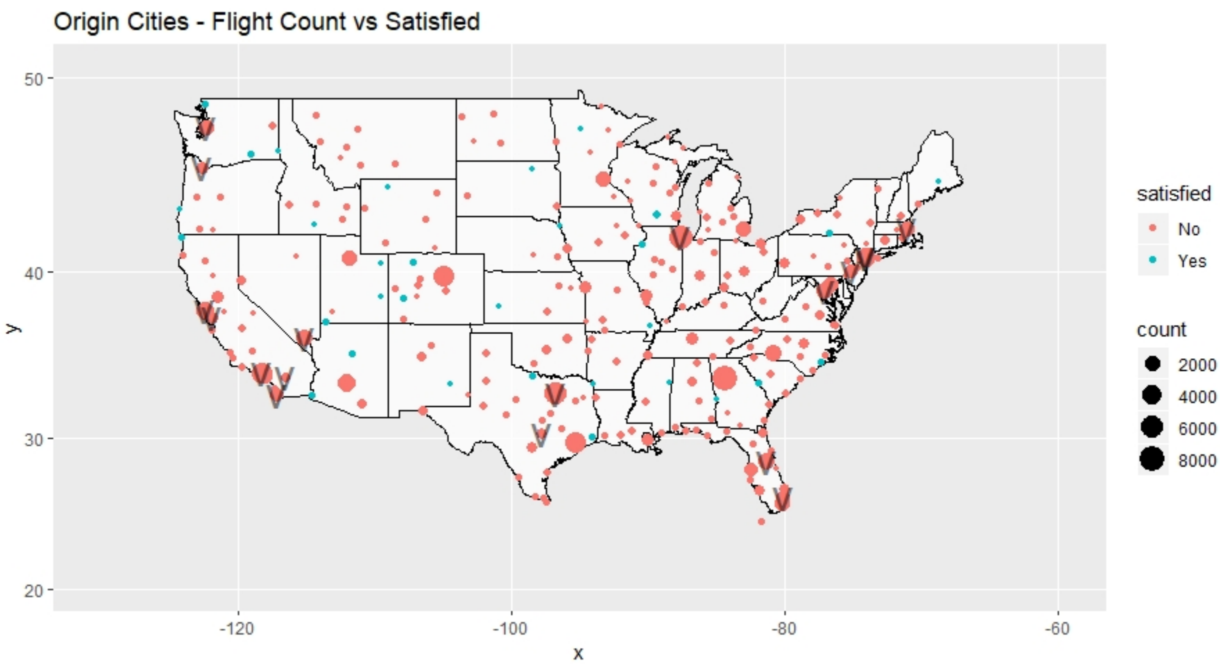


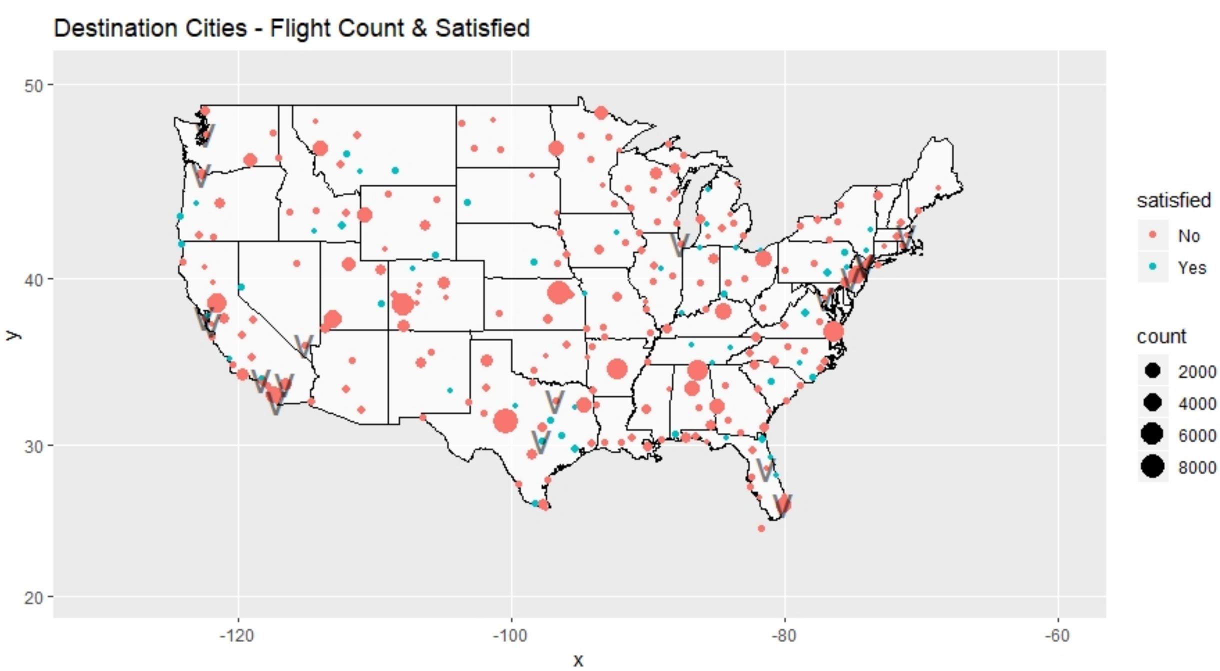


<u>Origin and Destination Cities</u>

When analyzing the origin and destination cities, the data analytical team used data mapping technique and scatter plots in an attempt to identify patterns between satisfaction and total flight count, the summation of flight delay in minutes, and the count of flight cancellations that associated with each origin and destination city. The results of the visual outputs suggest randomness with no obvious patterns of outliers. Although no correlation was identified, these findings contributed to the data analytics team focusing on the other variables that more accurately influenced customer satisfaction.

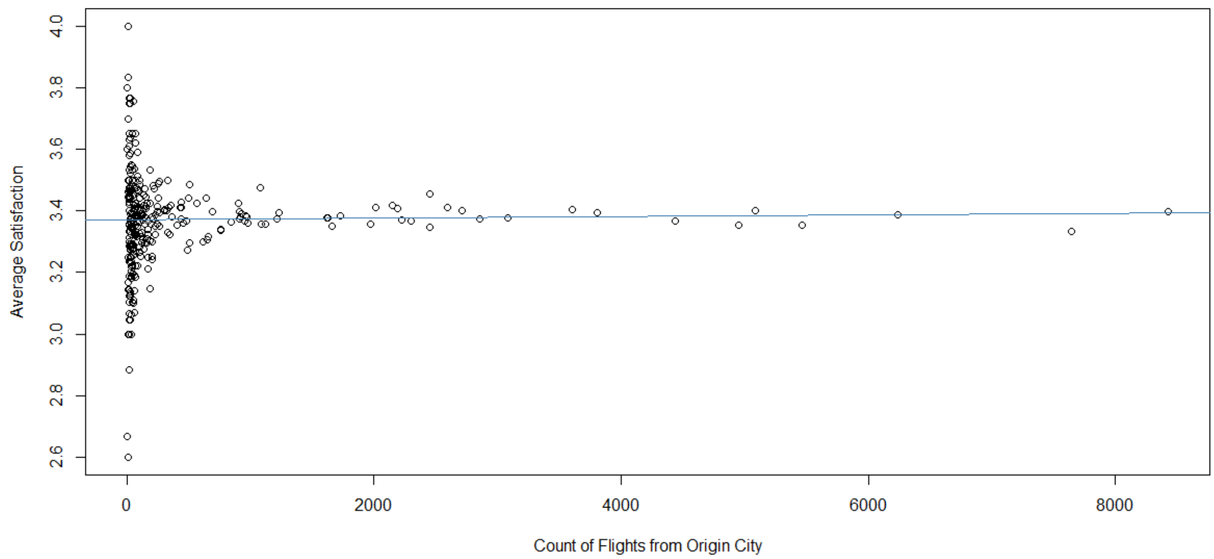
Below are a series of plots of analytical outputs. When looking at the data mapping of flight count and satisfaction status in different origin and destination cities, it seems that most of the satisfied cities have a small number of flight counts. This suggests that busy cities with a larger amount of fights have a higher unsatisfied ratio. However, a further study with scatter plots provided us with no evidence for such an assumption.

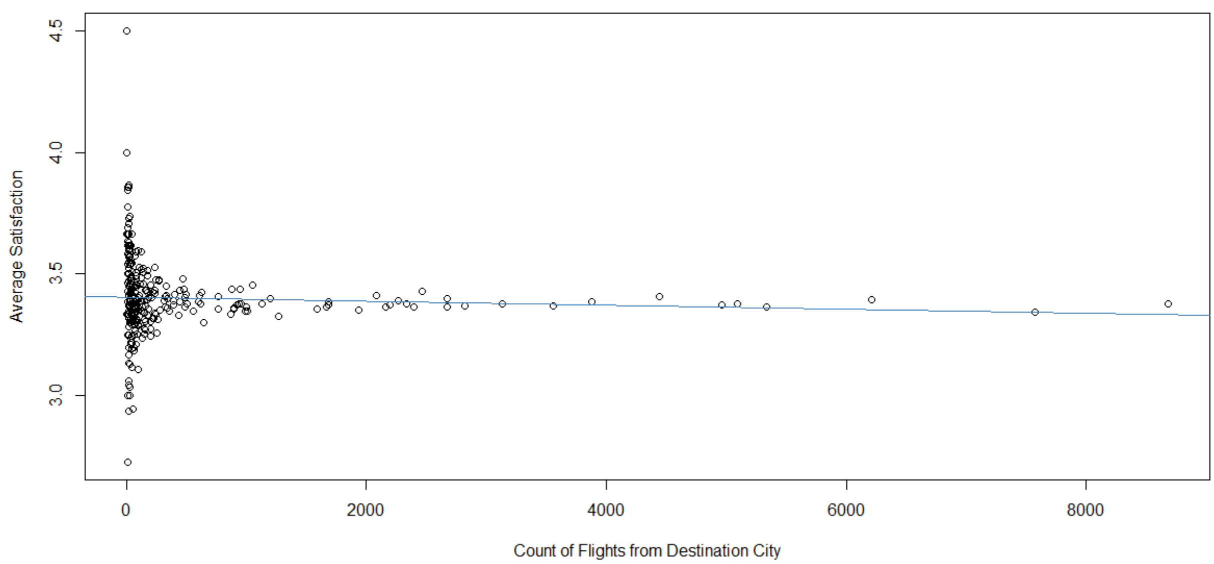




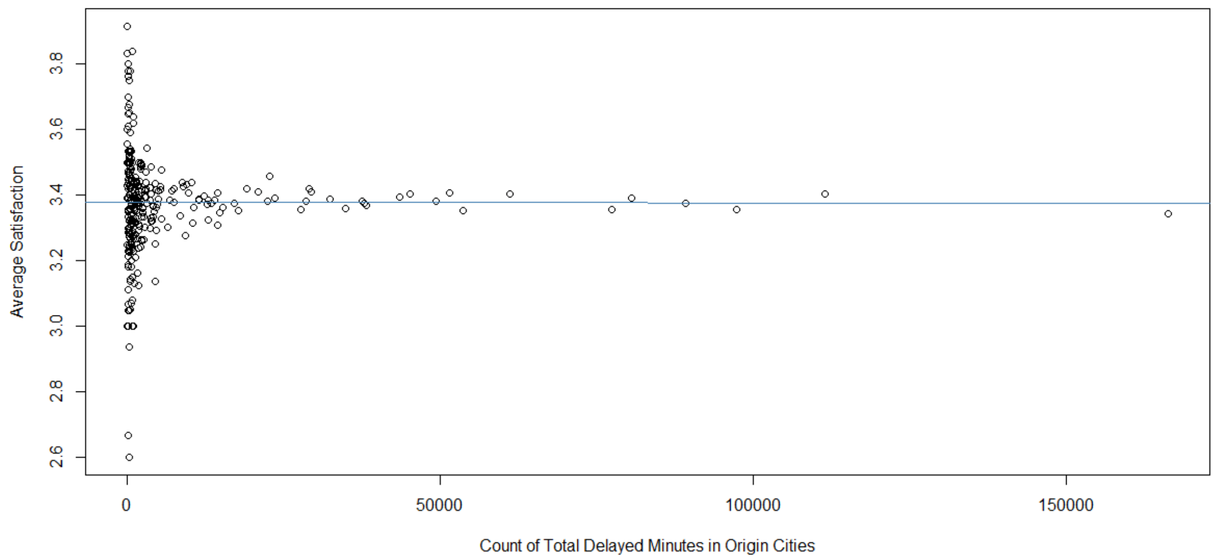
*** The "V" marks indicate cities of origin and destination in service of VX**

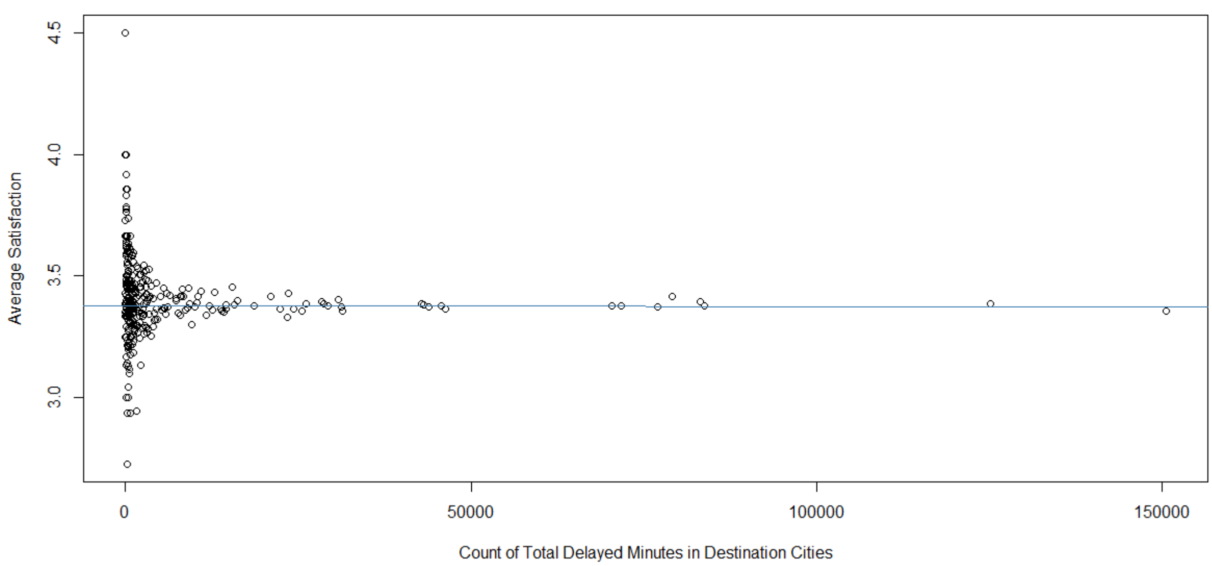
The symmetrical plots below suggest no obvious for correlation of high unsatisfied ratio with origin/ destination cities with large amount of flights.

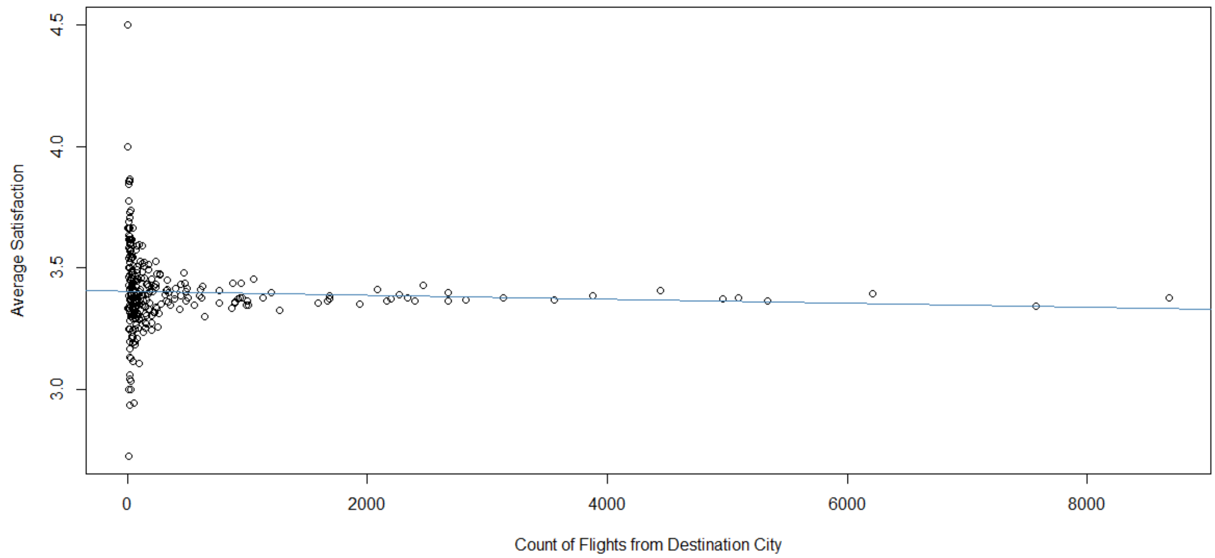


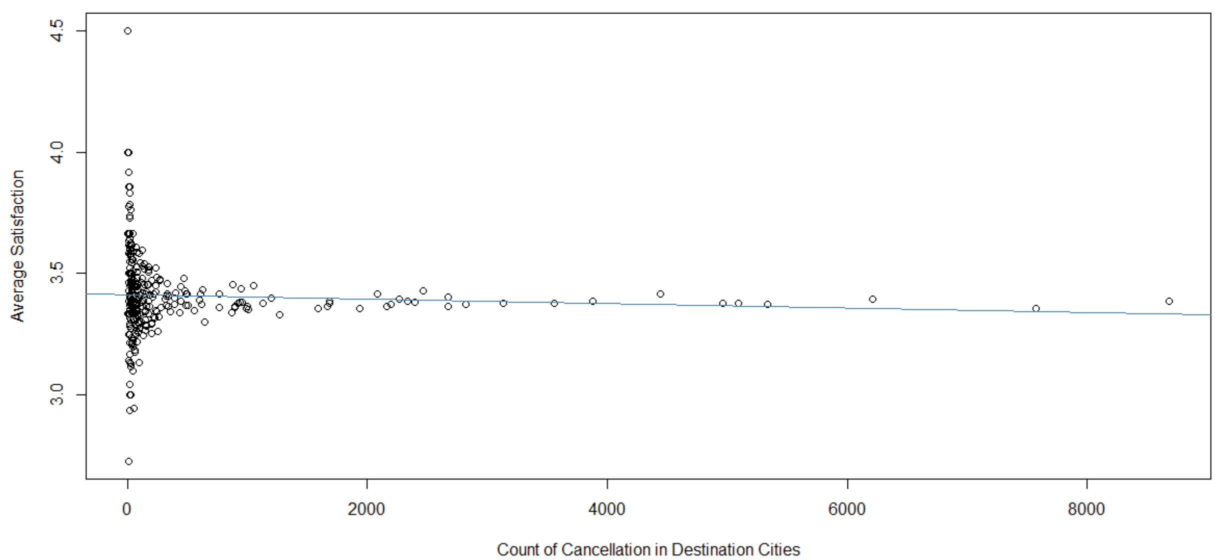


Similar analytical studies are applied to a total flight delay in minutes and count of flight cancellation vs flight count of origin/ destination cities, in order to identify any outlier patterns. Unfortunately, no apparent pattern was identified. See relevant plots below.


Average Satisfaction
3.8
3.6
3.4
3.2
3.0
2.8
2.6
0
50000
100000
150000
Count of Total Delayed Minutes in Origin Cities


Average Satisfaction
4.5
4.0
3.5
3.0
0
50000
100000
150000
Count of Total Delayed Minutes in Destination Cities



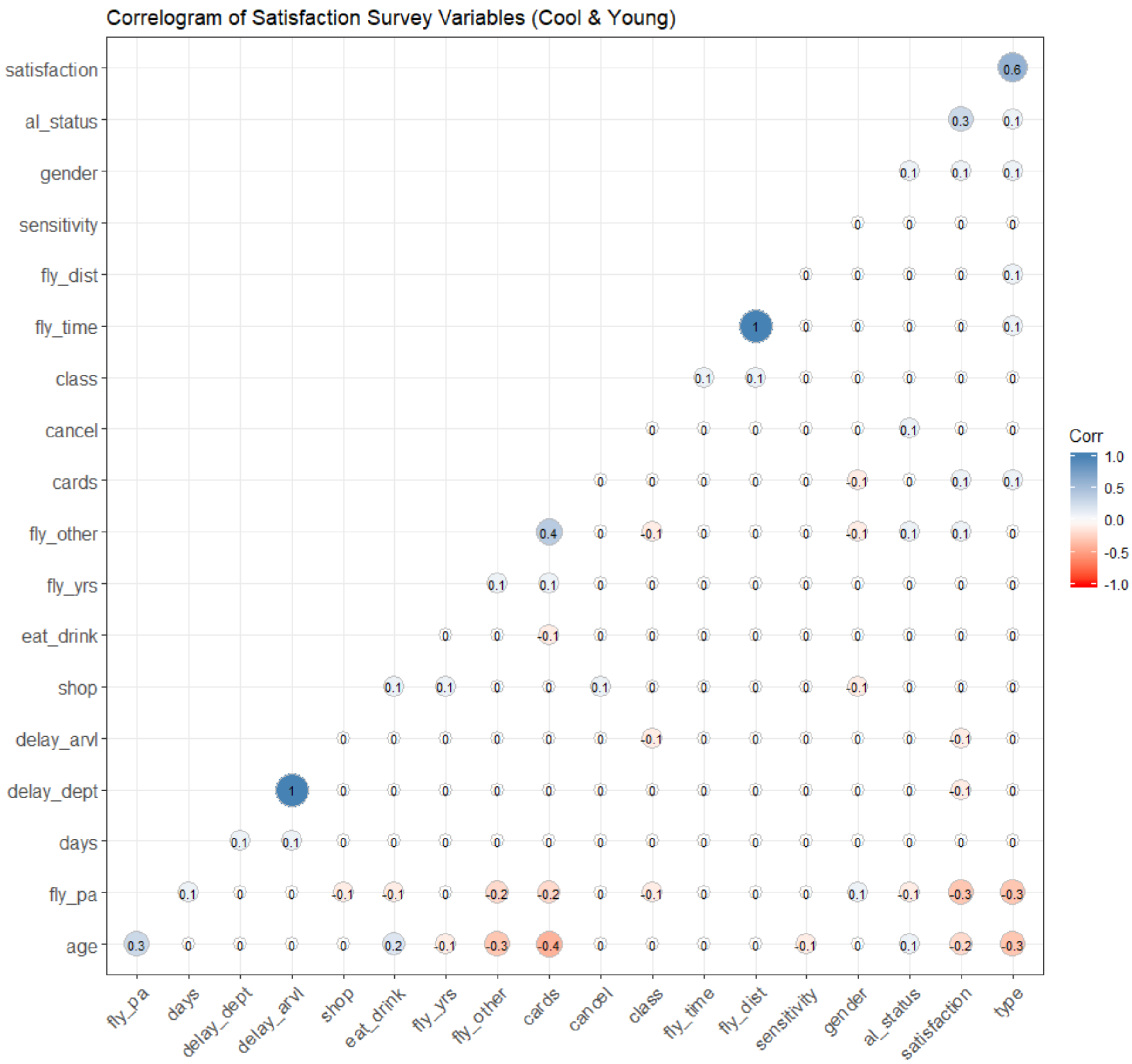


Predictive Models

In the very beginning of the modeling building phase, the team created two correlograms for inspection of correlation among all variables in the cleansed data set, one for VS, one for the overall data in comparison. This exercise helps in 2 aspects:

1. Visually and quantitively, it is easier to filter attributes that have a higher correlation with the satisfaction ratings;

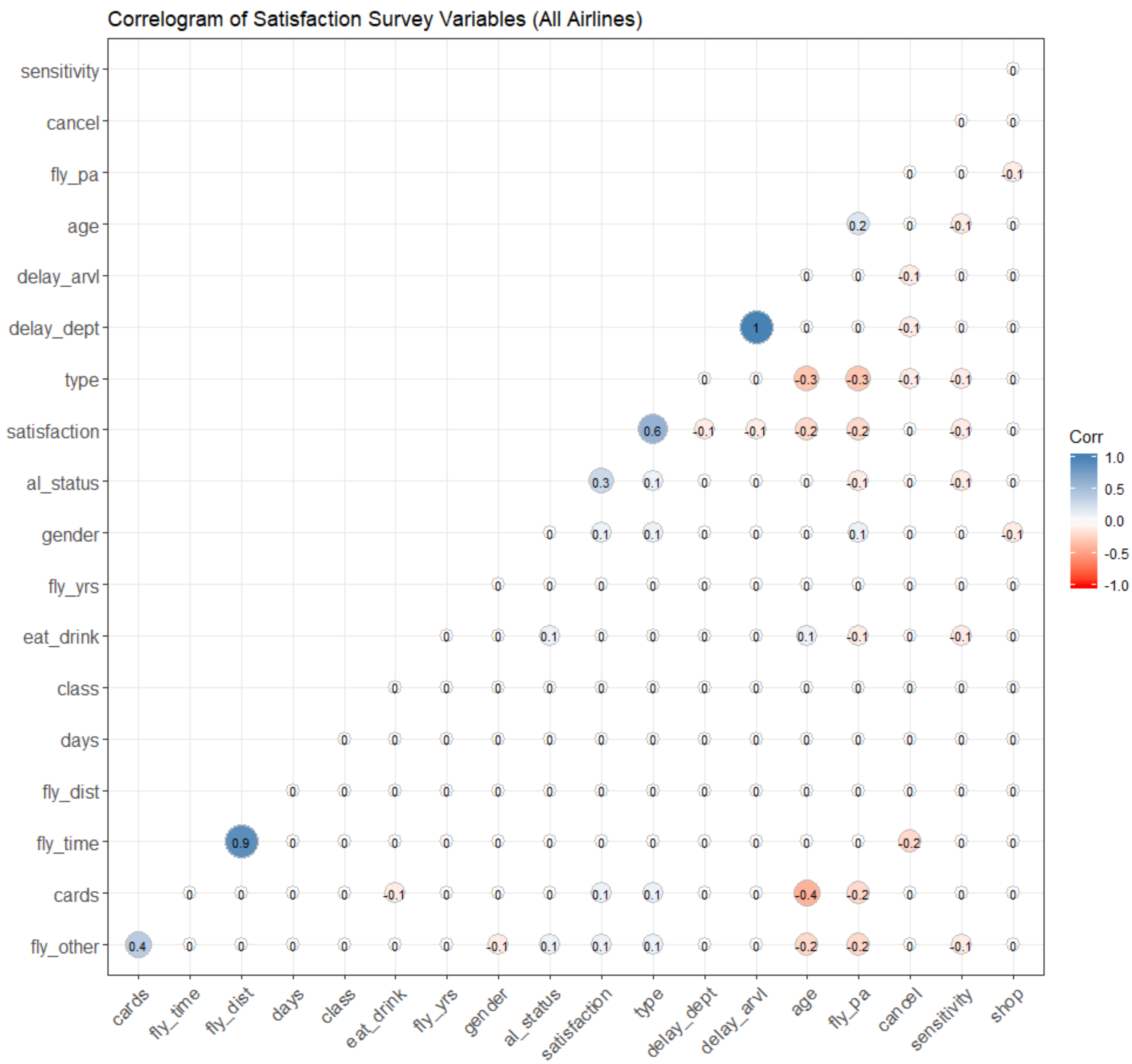
2. Visually and quantitively, it is easier to filter attributes that have a strong correlation among themselves, which distracts the performance of the models in prediction of the survey ratings.



As shown in the above correlogram of VX variables, the following attributes appear to have some correlation with satisfaction in descending order:

- Type of Travel – Correlation Coefficient at 0.6, positive correlation with satisfaction.
- Airline Status - Correlation Coefficient at 0.3, positive correlation with satisfaction.
- No. of Flight p.a. - Correlation Coefficient at -0.3, negative correlation with satisfaction.
- Age - Correlation Coefficient at -0.2, negative correlation with satisfaction.
- Gender - Correlation Coefficient at 0.1, positive correlation with satisfaction.

Another factor to be noticed is that pairs of variables “Flight Time” & “Flight Distance”, “Departure Delay in Minutes” & “Arrival Delay in Minutes” have perfect positive correlations, as both correlation coefficients are at 1. This means that these variables need to be combined before taking into the predictive models. What the team decided was to compute the product of “Flight Time” & “Flight Distance” as a new variable “fly_x” and compute the actual delay in minutes by subtracting “Departure Delay in Minutes” from “Arrival Delay in Minutes” as a new variable “delay.”



In short, this variable correlation analysis provides quantitative results strongly support for the team's assumption on key driving attributes to customer satisfaction in VX data. The team successfully identified the key drivers to unsatisfied customers in the descriptive analysis phase. Importantly, the team found similar correlations among attributes in the overall data, which again provides evidence to support the team's assumption.

```
> ## create a linear model for VX with all variables
> lm_vx <-lm(formula=satisfaction~., data=sv_vx)
> summary(lm_vx)

Call:
lm(formula = satisfaction ~ ., data = sv_vx)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2294 -0.4663  0.2003  0.4866  2.5060 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.706e+00  1.560e-01  17.351  < 2e-16 ***
al_status2   5.946e-01  5.201e-02  11.432  < 2e-16 ***
al_status3   5.022e-01  8.011e-02   6.269 4.97e-10 ***
al_status4   6.917e-01  1.157e-01   5.976 2.97e-09 ***
age         -4.422e-03  1.420e-03  -3.115  0.00188 ** 
gender1      1.186e-01  4.186e-02   2.834  0.00467 ** 
sensitivity  8.900e-03  3.944e-02   0.226  0.82151    
fly_yrs      9.250e-03  6.810e-03   1.358  0.17463    
fly_pa      -4.061e-03  1.522e-03  -2.667  0.00775 ** 
fly_other    7.936e-02  2.647e-01   0.300  0.76441    
type2        8.829e-01  8.020e-02  11.008  < 2e-16 ***
type3        1.031e+00  4.994e-02  20.645  < 2e-16 ***
cards       -2.559e-02  2.036e-02  -1.257  0.20897    
shop        -5.170e-04  3.726e-04  -1.388  0.16545    
eat_drink   -2.011e-04  4.349e-04  -0.462  0.64394    
class2       1.914e-02  7.110e-02   0.269  0.78786    
class3      -9.428e-03  6.976e-02  -0.135  0.89252    
days2       -3.889e-02  7.532e-02  -0.516  0.60572    
days3        8.878e-02  7.359e-02   1.206  0.22789    
days4        4.489e-02  7.398e-02   0.607  0.54409    
days5        3.261e-02  7.397e-02   0.441  0.65945    
days6        1.619e-02  8.257e-02   0.196  0.84463    
days7       -1.895e-02  7.482e-02  -0.253  0.80006    
cancel1     -9.178e-01  7.229e-01  -1.270  0.20447    
fly_x       -3.748e-08  5.652e-08  -0.663  0.50729    
delay       -2.084e-03  2.552e-03  -0.817  0.41431    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7136 on 1262 degrees of freedom
Multiple R-squared:  0.4332,	Adjusted R-squared:  0.4219 
F-statistic: 38.58 on 25 and 1262 DF,  p-value: < 2.2e-16
```

A summary of the linear model for VX data, including all attributes as input variables.

LINEAR MODEL

The team started with creating a linear regression model with all variables in the cleansed data set of VX. The adjusted R^2 is low at 0.4219. The linear model shows that most variables are not statistically significant in accounting for the variability of customer satisfaction variable.

The team then used the step() function and generated the most parsimonious model based on AIC suggestion. Although most of the insignificant variables are excluded from the regression model, the outcome of the adjusted R^2 at 0.4242 does not show significant improvement. The team also tried to build a linear regression model for the overall cleaned data set as a comparison, the outcome of the adjusted R^2 is very similar that plateaus at around 0.42.

A summary of the most parsimonious linear model for VX data.

```
> summary(lmp_vx)

Call:
lm(formula = satisfaction ~ al_status + age + gender + fly_pa +
    type, data = sv_vx)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2605 -0.4723  0.2349  0.4766  2.5242

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.769280   0.082246  33.671  < 2e-16 ***
al_status2   0.597592   0.051396  11.627  < 2e-16 ***
al_status3   0.494808   0.078968   6.266 5.05e-10 ***
al_status4   0.702707   0.114614   6.131 1.16e-09 ***
age         -0.004094   0.001271  -3.221  0.00131 **
gender1      0.118195   0.040597   2.911  0.00366 **
fly_pa      -0.003813   0.001462  -2.608  0.00921 **
type2        0.874558   0.079595  10.988  < 2e-16 ***
type3        1.030566   0.049107  20.986  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7122 on 1279 degrees of freedom
Multiple R-squared:  0.4278,	Adjusted R-squared:  0.4242
F-statistic: 119.5 on 8 and 1279 DF,  p-value: < 2.2e-16
```

Since this linear regression model contains 3 factorial variables, the coefficients for the factorial variables at different levels need to be taken into consideration. Below are a few lines of code that demonstrates the method and the final equation:

After the linear model created, the predicted values are stored in the 'prd' column in data frame 'sv_vx_lp'. This helps the team to compare them with the actual observations and calculate the root mean square errors (RMSE), which gauges the average residual size of the prediction.

The code for creating the linear regression equation for VX data.

```
> # create a linear regression model for Cool & Young Ailines, Inc.
> # create vectors to properly store Coefficients for categorical variables of the linear model
> coef_status <- c(0,lmp_vx$coefficients[2],lmp_vx$coefficients[3],lmp_vx$coefficients[4])
> coef_type <- c(0,lmp_vx$coefficients[8],lmp_vx$coefficients[9])
> coef_gender <- c(0,lmp_vx$coefficients[6])
> # ceate a new data frame for regression prediction
> sv_vx_lp <- sv_vx
> sv_vx_lp$prd <- lmp_vx$coefficients[1] + coef_status[as.numeric(sv_vx_lp$al_status)] + lmp_vx$coefficie
nts[5]*sv_vx_lp$age +  coef_gender[as.numeric(sv_vx_lp$gender)] + lmp_vx$coefficients[7] * sv_vx_lp$fly_p
a + coef_type[as.numeric(sv_vx_lp$type)]
```

The result of RMSE from linear model prediction for VX data.

```
> # check the value of root mean square error
> rmse_val <- rmse(sv_vx_lp$satisfaction, sv_vx_lp$prd)
> rmse_val
[1] 0.7096927
```

As shown in the above code, the RMSE is 0.71, considering the actual satisfaction rating has an increment at 0.5, 0.71 is within 2 increments. However, the team finds it complicated to think this way. In searching for a better and practical way of measuring the accuracy, the team came up with a way to gauge the relative accuracy of the predicted value by adding a tolerance when comparing. Below are the method and specific steps the team adopted to test the relative accuracy of the prediction yielded from the linear model.

First, the team took the predicted value and round them to only one decimal, then the rounded values are compared to the actual observations, if the difference is within 0.5, one increment of the survey rating, it is marked as accurate, and vice versa. This is an intuitive way for comparison, and it is much easier to understand the results. As it is shown, the linear model reaches a very high relative accuracy of over 97% correct.

```
> # 1) round the predicted value up to one decimal;
> sv_vx_lp$prd_rnd <- round(sv_vx_lp$prd, 1)
> # 2) set a tolerance that if the predicted value is within 0.5, it
is considered accurate
> sv_vx_lp$correct <- ifelse(abs(sv_vx_lp$prd_rnd - sv_vx_lp$satisfac
tion) >!0.5, 1,0)
> # 3) caculate the accuracy ratio
> accuracy_lm <- sum(sv_vx_lp$correct==1)/nrow(sv_vx_lp)
> accuracy_lm
[1] 0.9743789
```

In short, the team is very confident that attributes of "Type of Travel", "Airline Status", "No. of Flights p.a.", "Age", and "Gender" is correlated to customer satisfaction rating, which may be the key drivers for unsatisfaction.

NAIVE BAYES MODEL

The SVM model with Naïve Bayes algorithm is the second predictive model that the analytical team attempted, in hope to see if a classification prediction on whether a customer is satisfied or not would yield better accuracy.

A summary of the Naïve Bayes model and result.

```
> # train the algorithm to generate output
> nb_vx_out <-naiveBayes(satisfied~al_status + age + gender + sensitivity + fly_yrs + fly_pa + fly_other
 + type + cards + shop + eat_drink + class + days + delay_dept + delay_arvl + cancel + fly_time + fly_dis
t,
+                          data=vx_tr)
> nb_vx_out

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1 
0.4568765 0.5431235 

Conditional probabilities:
   al_status
Y            1          2          3          4
  0 0.85714286 0.08928571 0.03571429 0.01785714
  1 0.59227468 0.26824034 0.09656652 0.04291845

   age
Y       [,1]     [,2]
  0 49.71173 19.84817
  1 42.53219 13.35971

   gender
Y           0         1
  0 0.6173469 0.3826531
  1 0.4849785 0.5150215
```

...

```
> table(vx_pr[ ,c(20, 21)])
         prd
satisfied   0   1
        0 123  68
        1  30 209
> # check accuracy of predicted classification
> accuracy <- sum(vx_pr$satisfied==vx_pr$prd)/nrow(vx_pr)
> accuracy
[1] 0.772093
```

This model was built with all variables in the cleansed data set of VX, which 2/3 of the data was randomly selected as training data, and the rest 1/3 was used as test data predicted by the algorithm based on the output of the training data. As shown above, it yields an accuracy at 77%. Compared with the relative accuracy of the linear model, the performance is inferior.

KSVM MODEL

For the last predictive model, the analytical team attempted the KSVM algorithm in predicting the satisfaction ratings. Similar to the Naïve Bayes model, all variables from the cleansed data set of VX was used, with randomly selected 2/3 for training, and the rest 1/3 for testing. After a few tweaking on the parameters, the team found a combination of “C=80”, and “cross=5” yields reasonable results with training error remains below 0.03 most of the times.

```
> # train the algorithm to generate output
> svm_vx_out <- ksvm(satisfaction~., data=vx_tr, kernal = "rbfdot", kpar = "automatic", C=80, cross=5, pro
b.model=TRUE)
> svm_vx_out
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 80

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.050993256746024

Number of Support Vectors : 795

Objective Function Value : -5050.301
Training error : 0.024772
Cross validation error : 0.979344
Laplace distr. width : 2.795741
> # predit satisfaction of test data based on trained data
> vx_pr$prd <- predict(svm_vx_out, vx_pr)
> # check the value of root mean square error
> rmse_val <- rmse(vx_pr$satisfaction, vx_pr$prd)
> rmse_val
[1] 0.9313642
```

However, the prediction yields a RMSE value at 0.93, which is higher than the RMSE of the linear model at 0.71. The team also computed the relative accuracy of this KSVM model, by which is at 96%.

```
> # calculate the relative accuracy
> # 1) round the predicted value up to one decimal;
> vx_pr$prd_rnd <- round(vx_pr$prd, 1)
> # 2) set a tolerance that if the predicted value is within 0.5, it is considered accurate
> vx_pr$correct <- ifelse(abs(vx_pr$prd_rnd - vx_pr$satisfaction) >!0.5, 1,0)
> # 3) caculate the accuracy ratio
> accuracy_ksvm <- sum(vx_pr$correct==1)/nrow(vx_pr)
> accuracy_ksvm
[1] 0.9604651
```

Although the KSVM model performed well with high relative accuracy at 96%, the linear regression model that the team building has a slightly better result.

***

In conclusion, with a few tweaks and modifications, the predictive models yielded outstanding results with some imperfection. Comparing the three models, the analytical team found the linear regression model as the best predictive model. In fact, the linear regression model provides sound support for the team's assumption of the five key attributes that drive customer satisfaction. In addition, the slightly lower relative accuracy of the prediction results from the KSVM model indicating the significance and effectiveness of these 5 key attributes in correlation with the overall customer satisfaction. The analytical team is very confident with the assumption based on the analytical method and application described in this report, and as well as the actionable insights suggested to improve the overall customer satisfaction of Cool & Young Airline, Inc.

Actionable Insights, Findings, and Recommendations

Cool & Young Airlines, Inc. is doing a good job overall to meet travelers' needs. In fact, the airline has the second-highest customer satisfaction scores across the industry. With any business, however, proactively trying to improve is what keeps the business doing well; e.g. what if the people at Apple rested on their laurels after they released the first Apple II? There is always somewhere to innovate and room to improve. **We identified five key attributes by investigating the congregation of the unsatisfied customers, to which the managing team at VX will need to pay close attention, in order to lower the unsatisfied customer ratio and improve the overall customer satisfaction rating.**

Customers Travel for Personal Reasons:

Customers flying for personal reasons are significantly less satisfied than those traveling on mileage or for business. Factors that could influence this could be that they are traveling as a family (which can be stressful) or simply with more people. Those people travelling to go on vacation could have more needs as compared with the solo business traveler. Also, they probably must check bags at an additional cost as compared with a business traveler that usually just brings one overnight carry-on bag.

Business travelers might be accustomed to traveling and knowing what to expect with a chosen airline, the opposite may be true with a personal traveler who may have higher services and experiences sensitivity because they have a higher expectation of services they want to be comparable to their hard-earned money. In some circumstances, the airline may receive a low mark because of the personal traveler's experience with TSA that makes them unsatisfied. Thus, the team suggests VX to consider additional programs and services to make sure the personal traveler's experience consistent quality.

Customers in Blue Status

It is evident that the blue status has an impact on satisfaction. To bring it all together these blue status travelers just do not have the same perks and experiences that lead to a high customer satisfaction score.

VX needs to do more to help make these people feel welcome and heard. There could be a sense of jealousy when Blue status flyers see other customers being treated better. Just because a traveler is not of an elite status does not mean they should be treated like a second-class citizen. These Blue status travelers could one day reach Platinum status. If they write off VX early on, they might one day have Platinum status but with another airline and never try VX again. The team suggests that the services and experiences the blue status travelers receive should be inspected and properly improved. VX may also want to try giving higher status perks to Blue customers, as a tease or promotion, to nudge them towards buying into the higher status, which will raise their satisfaction and lead to higher profits.

Customers with 20+ past flights

The frequent flying customers often tend to be less satisfied. This could be due to the increased chances to be exposed to distractors, or elevated expectations. To reward frequent flyers and help them look past some potential blunders in their experience, VX could offer them more benefits.

The team suggests VX to create additional perk programs for these frequent flyers, which will leverage incentives to keep them loyal customers to the airline. More points, early boarding, access to a VIP lounge, etc… all of which are some great ideas that can create a situation for VX to separate themselves from the airlines they compete with.

Customers in junior and senior age groups

In general, junior customers between age 15 to 25 and senior customers above age 55 tend to have lower than average satisfaction ratings. Among the two, the senior group tends to be the most dissatisfied. The team suggests VX to do additional research on the specific wants and needs of these two groups. For example, Early boarding for seniors could make them more comfortable. There could also be dedicated agents available to greet elderly travelers or help the elders navigate through the airport and ensuring they are comfortable. Senior Citizens may also like the perk of being automatically upgraded to premium seating. These are all possible improvements for VX to make a mark and separate themselves from their competitors.

Female Customers:

The facts depict female travelers are not satisfied. The results of unsatisfied ratios between female and male customer groups are close; the small gap could be tied to something specific. There could be several known and unknown reasons, and further investigation is suggested in a focus group to determine what would make a female traveler's experience better. It would be interesting to look at one of the few differences between the genders, such as a scenario that is related to carrying a child and the associated significant efforts and challenges.

In summary, the team was hoping the support vector machine would do a better job by utilizing complicated algorithm with a wider range of variables and data. However, what the team eventually found is somewhat unexpected, but overall within our recognition: sometimes quality overrules quantity, and it applies to this particular case of data analysis very well.

Later in the analytic process, the outputs of the 3 models were not very convincing, due to the low adjusted R^2, relatively high RMSEs, and flat accuracy of prediction from the Naïve Bayes model. The team felt frustrated with the results and doubted about the lack of confidence in the predictions. Not until all the findings of the predictive models gathered, members of the analytical team came up with the idea of introducing relative accuracy with a reasonable tolerance of error that shed light on the positive conclusion. The team benefited from thinking critically and solving the problem creatively.

Due to limited time, the analytical team did not get a chance to explore association rules. The team also hoped there was time left for building a neuro-network model, comparing results with the other predictive models.

Appendix

This appendix includes the R codes used in each step of the analysis process, including data preparation, descriptive analysis, predictive analysis, and various charts and plots generated for the report.

DATA PREPARATION: IMPORT, CLEANSING, AND MUNGING

```
## check, install, and load required packages
packages <- c("readxl", "plyr")
package.check <- lapply(packages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
 })
## clean up packages and package.check after checking the packages
rm(packages)
rm(package.check)
# --------------------
## import the survey data
# set workspace to drectory of the satisfaction survey file
setwd("~/OneDrive - Syracuse University/SU/Courses/IST 687/Project")
AirSurvey <- read_excel("SatisfactionSurvey.xlsx") # read.xls in gdata package is very slow
# --------------------
## data cleanse
```

```
survey <- AirSurvey # create a working data set

colnames(survey)[colnames(survey)=="Orgin City"] <- "Origin City" # correct a typo

# dealing with NAs

# check where and how many NAs in the data

View(lapply(survey, function(x) length(which(is.na(x)))))

# 3 NAs in 'Satisfactoin'

# 2345 NAs in 'Departure Delay in Minutes'

# 2738 NAs in 'Arrival Delay in Minutes'

# 2738 NAs in 'Flight time in minutes'

# remove 3 rows with NAs in 'Satisfaction'

survey <- survey[-(which(is.na(survey$Satisfaction))), ]

sum(survey$`Flight cancelled`=="Yes")

# 2401 records of cancelled flights

sum(survey[which(is.na(survey$'Departure Delay in Minutes')), 'Flight cancelled']=="No")

# all NA records of 'Departure Delay in Minutes' are associated with flight cancellation.

sum(survey[which(is.na(survey$'Arrival Delay in Minutes')), 'Flight cancelled']=="No")

# 337 NA records of 'Arrival Delay in Minutes' are not associated with flight cancellation.

sum(survey[which(is.na(survey$'Flight time in minutes')), 'Flight cancelled']=="No")

# 337 NA records of 'Flight time in minutes' are not associated with flight cancellation.

nrow(subset(survey, is.na(survey$'Arrival Delay in Minutes') & is.na(survey$'Flight time in minutes') & survey$'Flight cancelled'=="No"))

# the 337 NA records of 'Departure Delay in Minutes' & 'Flight time in minutes' that are not associated with flight cancellation

head(subset(survey, is.na(survey$'Arrival Delay in Minutes') & is.na(survey$'Flight time in minutes') & survey$'Flight cancelled'=="No"))

# a quick look at these NAs, and they seem to be missing values, thus these records need to be omitted for analysis

# remove 337 rows with NAs in 'Departure Delay in Minutes' & 'Flight time in minutes' that are not associated with flight cancellation

survey <- survey[-(which(survey[which(is.na(survey$'Flight time in minutes')), 'Flight cancelled']=="No")), ]
```

```
# convert NAs of departure/ arrival delay & flight time to 0
survey$`Departure Delay in Minutes`[is.na(survey$`Departure Delay in Minutes`)] <- 0
survey$`Arrival Delay in Minutes`[is.na(survey$`Arrival Delay in Minutes`)] <- 0
survey$`Flight time in minutes`[is.na(survey$`Flight time in minutes`)] <- 0
# check the NAs again
View(lapply(survey, function(x) length(which(is.na(x)))))
# no NA found in the data set
# --------------------
## data preparation
str(survey)
# rename satisfaction
survey$satisfaction <- survey$Satisfaction
# convert airline status
survey$al_status <- as.factor(mapvalues(survey$`Airline Status`, from=c("Blue", "Silver", "Gold", "Platinum"), to=c(1,2,3,4)))
# rename age
survey$age <- survey$Age
# convert gender
# 0 - Female
# 1 - Male
survey$gender <- as.factor(mapvalues(survey$Gender, from=c("Male", "Female"), to=c(1,0)))
# rename price sensitivity
survey$sensitivity <- survey$`Price Sensitivity`
# convert year of 1st flight to fly_years
survey$fly_yrs <- 2019-survey$`Year of First Flight`
# rename No. of flight p.a.
survey$fly_pa <- survey$`No of Flights p.a.`
# convert % of flight with other airlines
survey$fly_other <- survey$`% of Flight with other Airlines` / 100
```

```
# convert type of travel

survey$type <- as.factor(mapvalues(survey$`Type of Travel`, from=c("Personal Travel", "Mileage tickets", "Business travel"), to=c(1,2,3)))

# rename number of cards

survey$cards <- survey$`No. of other Loyalty Cards`

# rename shopping amount

survey$shop <- survey$`Shopping Amount at Airport`

# rename eating & drinking

survey$eat_drink <- survey$`Eating and Drinking at Airport`

# convert class

survey$class <- as.factor(mapvalues(survey$Class, from=c("Eco", "Eco Plus", "Business"), to=c(1,2,3)))

# convert day of month & date into day of week

# 1 - Monday

# 2 - Tuesday

# 3 - Wednesday

# 4 - Thursday

# 5 - Friday

# 6 - Saturday

# 7 - Sunday

survey$days <- weekdays(as.Date(survey$`Flight date`, '%Y-%m-%d'))

survey$days <- as.factor(mapvalues(survey$days, from=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"), to=c(1,2,3,4,5,6,7)))

# rename departure/ arrival delay in minutes

survey$delay_dept <- survey$`Departure Delay in Minutes`

survey$delay_arvl <- survey$`Arrival Delay in Minutes`

# convert flight cancellation status

survey$cancel <- as.factor(mapvalues(survey$`Flight cancelled`, from=c("Yes", "No"), to=c(1,0)))

# rename flight time/ distance
```

```
survey$fly_time <- survey$`Flight time in minutes`
survey$fly_dist <- survey$`Flight Distance`
# rename origin/ destionation cities/ states
survey$origin_city <- tolower(survey$`Origin City`)
survey$origin_state <- tolower(survey$`Origin State`)
survey$destin_city <- tolower(survey$`Destination City`)
survey$destin_state <- tolower(survey$`Destination State`)
# combine airline code and airline name into airlines (als)
survey$als <- paste(survey$"Airline Name", survey$"Airline Code",  sep=" - ")
# remove columns not needed
survey <- survey[ ,-1:-28]
str(survey)
# create data frames for all records and records for Cool&Young for further analysis.
sv_all <- survey[ ,-20:-24]
sv_vx <- survey[survey$als=="Cool&Young Airlines Inc. - VX", -20:-24]
```

CODE FOR DESCRIPTIVE ANALYSIS - BARPLOTS

```
## check, install, and load required packages
packages <- c("data.table", "ggplot2", "maps", "ggmap", "mapproj", "sqldf")
package.check <- lapply(packages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})
## clean up packages and package.check after checking the packages
rm(packages)
rm(package.check)
```

```
# --------------------
## furhter preparation of data for plots & visuals
# create a data frame for plots & visuals
vis <- survey
# Add column 'satisfied' to label satisfaction rate - "No" [0,3], Yes" [3.5,5]
vis$satisfied <- ifelse(vis$satisfaction > 3, "Yes", "No")
# Add column 'sensitive' to label price sensitivity - "No" [0,2], Yes" [3,5]
vis$sensitive <- ifelse(vis$sensitivity > 2, "Yes", "No")
# Add column 'frequent' to label no. of flights p.a. - "No" [0, 20], "Yes" [20, ]
vis$frequent <- ifelse(vis$fly_pa > 20, "Yes", "No")
# Add column 'morecards' to label no. of other loyaty cards - "No" [0,4], "Yes" [4,]
# Add "Yes" as No of Loyalty Cards <! 4, "No" for <= 4.
vis$morecards <- ifelse(vis$cards> 4, "Yes", "No")
# Summary of each variable prepared
summary(vis)
# --------------------
# descriptive analysis
# --------------------
# satisfaction survey overview
# --------------------
# count the number of survey records from each airline and compare:
ls_rec <- data.frame(tapply(vis$satisfaction, vis$als, length))
ls_rec$Airlines <- rownames(ls_rec)
ls_rec$Entries <- ls_rec[ ,1]
ls_rec <- ls_rec[ ,-1]
rownames(ls_rec) <- seq(length=nrow(ls_rec))
# plot a bar chart comparing numbers of survey records from each airline
g <- ggplot(ls_rec, aes(x=reorder(Airlines, -Entries), y=Entries)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 28000)
```

```
g <- g + theme(axis.text.x=element_text(angle=90, hjust=1)) + geom_text(aes(label=Entries), vjust=-0.5)

g_rec <- g + ylab("Number of Survey Records") + xlab("Airlines") + ggtitle("Number of Survey Records from Each Airline")

g_rec

# --------------------

# compute overall ratios of satisfied and unsatisfied customers as a baseline

x1 <- c(sum(ct_sat_al$Satisfied), sum(ct_sat_al$Unsatisfied))

x2 <- c(round(x1[1]/(x1[1]+x1[2]), 3)*100,round(x1[2]/(x1[1]+x1[2]), 4)*100 )

ct_sat <- data.frame(x1, x2)

ct_sat$Satisfaction <- c("Satisfied", "Unsatisfied")

ct_sat$Quantity <- ct_sat[ ,1]

ct_sat$Percentage <- ct_sat[ ,2]

ct_sat <- ct_sat[ ,-1:-2]

# overall unsatisfied ratio

AvgUnsatRate <- ct_sat[2,3]

# plot a bar chart of overall ratios of satisfied and unsatisfied customers

g <- ggplot(ct_sat, aes(x=reorder(Satisfaction, -Percentage), y=Percentage)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 60)

g <- g + geom_text(aes(label=Percentage), vjust=-0.5)

g_sat <- g + ylab("Satisfaction Rate") + xlab("Customer Groups") + ggtitle("Ratios of Satisfied and Unsatisfied Customers in the Industry")

g_sat

# --------------------

# compute ratios of satisfied and unsatisfied customers for each airline:

ct_sat_al <- data.frame(tapply(vis$satisfaction, list(vis$als, vis$satisfied=="Yes"), length))

ct_sat_al$Airlines <- rownames(ct_sat_al)

ct_sat_al$Satisfied <- ct_sat_al[ ,2]

ct_sat_al$Unsatisfied <- ct_sat_al[ ,1]

ct_sat_al <- ct_sat_al[ ,-1:-2]
```

```
rownames(ct_sat_al) <- seq(length=nrow(ct_sat_al))

ct_sat_al$SatRate <- round(ct_sat_al$Satisfied/(ct_sat_al$Satisfied+ct_sat_al$Unsatisfied),4)*100

ct_sat_al$UnsatRate <- round(ct_sat_al$Unsatisfied/(ct_sat_al$Satisfied+ct_sat_al$Unsatisfied),4)*100

# plot a bar chart of the unsatisfied customers ratios comparing all airlines:

g <- ggplot(ct_sat_al, aes(x=reorder(ct_sat_al$Airlines, ct_sat_al$UnsatRate), y=ct_sat_al$UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 60)

g <- g + theme(axis.text.x=element_text(angle=90, hjust=1)) + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Average Unsatisfaction Ratio") + xlab("Airlines") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_satal <- g+ ggtitle("Ratios of Unsatisfied Customers Comparing All Airlines")

g_satal

# --------------------

# compute overall satisfaction rate of the industry as a baseline:

AvgSat <- round(mean(vis$satisfaction), 3)

# overall satisfaction rate 3.379 for the industry

# compare the average satisfaction rates among airlines:

ls_sat <- data.frame(tapply(vis$satisfaction, vis$als, mean))

ls_sat$Airlines <- rownames(ls_sat)

ls_sat$AvgSatisfaction <- round(ls_sat[ ,1], 3)

ls_sat <- ls_sat[ ,-1]

rownames(ls_sat) <- seq(length=nrow(ls_sat))

# plot a bar chart for comparison of average satisfaction rate among airlines:

g <- ggplot(ls_sat, aes(x=reorder(Airlines, -AvgSatisfaction), y=AvgSatisfaction)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 5)

g <- g + theme(axis.text.x=element_text(angle=90, hjust=1)) + geom_text(aes(label=AvgSatisfaction), vjust=-0.5)

g <- g + ylab("Average Satisfaction") + xlab("Airlines") + geom_hline(yintercept=AvgSat, linetype="dashed", color="blue", size=1, alpha=0.7)
```

```
g_sat <- g + ggtitle("Average Satisfaction Rates Comparing All Airlines")

g_sat

# --------------------

# inspect factors associated with high unsatisfied ratio

# the below codes were executed twice, the second time with "vis" as only the records of only Cool & Young, each with a proper ggtitle.

# --------------------

# 1) analyze unsatisfaction in "airline status"

ct_status <- data.frame(tapply(vis$satisfaction, list(vis$al_status, vis$satisfied=="Yes"), length))

ct_status$Status <- c("Blue", "Silver", "Gold", "Platinum")

ct_status$Satisfied <- ct_status[ ,2]

ct_status$Unsatisfied <- ct_status[ ,1]

ct_status <- ct_status[ ,-1:-2]

ct_status$UnsatRate <- round(ct_status$Unsatisfied / (ct_status$Satisfied + ct_status$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_status, aes(x=reorder(Status, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 65)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("Airline Status") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_status <- g + ggtitle("Unsatisfied Ratio in Airline Status (All Airlines)")

g_status

# --------------------

# 2) analyze satisfaction in "age"

summary(vis$age)

# sort customers in 7 age groups, and compute average satisfaction rate for each age group

attach(vis)

x1 <- round(sqldf('select avg(satisfaction) from vis where age<26'), 3)

x2 <- round(sqldf('select avg(satisfaction) from vis where age>25 and age<36'), 3)
```

```
x3 <- round(sqldf('select avg(satisfaction) from vis where age>35 and age<46'), 3)

x4 <- round(sqldf('select avg(satisfaction) from vis where age>45 and age<56'), 3)

x5 <- round(sqldf('select avg(satisfaction) from vis where age>55 and age<66'), 3)

x6 <- round(sqldf('select avg(satisfaction) from vis where age>65 and age<76'), 3)

x7 <- round(sqldf('select avg(satisfaction) from vis where age>75 and age<86'), 3)

sat_age <- rbind(x1, x2, x3, x4, x5 ,x6, x7)

sat_age$AgeGroup <- c("15-25", "26-35", "36-45", "46-55", "56-65", "66-75", "76-85")

sat_age$SatRate <- sat_age[ ,1]

sat_age <- sat_age[ ,-1]

# plot a bar chart for comparison

g <- ggplot(sat_age, aes(x=AgeGroup, y=SatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 5)

g <- g + geom_text(aes(label=SatRate), vjust=-0.5)

g <- g + ylab("Average Satisfaction") + xlab("Age Groups") + geom_hline(yintercept=AvgSat, linetype="dashed", color="blue", size=1, alpha=0.7)

g_age <- g + ggtitle("Satisfation Rate of Customers in Different Age Groups (All Airlines)")

g_age

# --------------------

# 3) analyze unsatisfaction in "gender"

ct_gender <- data.frame(tapply(vis$satisfaction, list(vis$gender, vis$satisfied=="Yes"), length))

ct_gender$Gender <- c("Female", "Male")

ct_gender$Satisfied <- ct_gender[ ,2]

ct_gender$Unsatisfied <- ct_gender[ ,1]

ct_gender <- ct_gender[ ,-1:-2]

ct_gender$UnsatRate <- round(ct_gender$Unsatisfied / (ct_gender$Satisfied + ct_gender$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_gender, aes(x=reorder(Gender, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 60)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)
```

```
g <- g + ylab("Unsatisfied Ratio") + xlab("Gender") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_gender <- g + ggtitle("Unsatisfied Ratio in Gender (All Airlines)")

g_gender

# --------------------

# 4) analyze unsatisfaction in "price sensitivity"

ct_price <- data.frame(tapply(vis$satisfaction, list(vis$sensitive, vis$satisfied=="Yes"), length))

ct_price$Sensitivity <- c("Not Sensitive", "Sensitive")

ct_price$Satisfied <- ct_price[ ,2]

ct_price$Unsatisfied <- ct_price[ ,1]

ct_price <- ct_price[ ,-1:-2]

ct_price$UnsatRate <- round(ct_price$Unsatisfied / (ct_price$Satisfied + ct_price$Unsatisfied), 4) * 100


# plot a bar chart for comparison

g <- ggplot(ct_price, aes(x=reorder(Sensitivity, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 70)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("Price Sensitivity") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_price <- g + ggtitle("Unsatisfied Ratio in Price Sensitivity (All Airlines)")

g_price

# --------------------

# 5) analyze unsatisfaction in "no. of flight p.a." (not frequent < 20, frenquent > 20)

ct_fl_pa <- data.frame(tapply(vis$satisfaction, list(vis$frequent, vis$satisfied), length))

ct_fl_pa$no_fl_pa <- c("<20", "20+")

ct_fl_pa$Satisfied <- ct_fl_pa[ ,2]

ct_fl_pa$Unsatisfied <- ct_fl_pa[ ,1]

ct_fl_pa <- ct_fl_pa[ ,-1:-2]
```

```
ct_fl_pa$UnsatRate <- round(ct_fl_pa$Unsatisfied / (ct_fl_pa$Satisfied + ct_fl_pa$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_fl_pa, aes(x=reorder(no_fl_pa, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 70)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("No of Flights p.a.") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_flpa <- g + ggtitle("Unsatisfied Ratio in No. of Flight P.A. (All Airlines)")

g_flpa



# --------------------

# 6) analyze unsatisfaction in "type of travel"

ct_type <- data.frame(tapply(vis$satisfaction, list(vis$type, vis$satisfied=="Yes"), length))

ct_type$Type <- c("Personal","Mileage","Business")

ct_type$Satisfied <- ct_type[ ,2]

ct_type$Unsatisfied <- ct_type[ ,1]

ct_type <- ct_type[ ,-1:-2]

ct_type$UnsatRate <- round(ct_type$Unsatisfied / (ct_type$Satisfied + ct_type$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_type, aes(x=reorder(Type, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 100)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("Type of Travel") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_type <- g + ggtitle("Unsatisfied Ratio in Type of Travel (All Airlines)")

g_type

# --------------------

# 7) analyze unsatisfaction in "no. of other loyalty cards"
```

```
ct_lcards <- data.frame(tapply(vis$satisfaction, list(vis$morecards, vis$satisfied), length))

ct_lcards$Loyalty_Cards <- c("<5", "5+")

ct_lcards$Satisfied <- ct_lcards[ ,2]

ct_lcards$Unsatisfied <- ct_lcards[ ,1]

ct_lcards <- ct_lcards[ ,-1:-2]

ct_lcards$UnsatRate <- round(ct_lcards$Unsatisfied / (ct_lcards$Satisfied + ct_lcards$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_lcards, aes(x=reorder(Loyalty_Cards, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 65)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("No of other Loyalty Cards") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_card <- g + ggtitle("Unsatisfied Ratio in No. of Other Loyalty Cards (All Airelines)")

g_card

# --------------------

# 8) analyze unsatisfaction in "class" - No obious pattern in unsatisfied ratio

ct_class <- data.frame(tapply(vis$satisfaction, list(vis$class, vis$satisfied), length))

ct_class$Class <- c("Eco", "EcoPlus", "Business")

ct_class$Satisfied <- ct_class[ ,2]

ct_class$Unsatisfied <- ct_class[ ,1]

ct_class <- ct_class[ ,-1:-2]

ct_class$UnsatRate <- round(ct_class$Unsatisfied / (ct_class$Satisfied + ct_class$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_class, aes(x=reorder(Class, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 65)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("Class") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_class <- g + ggtitle("Unsatisfied Ratio in Class (All Airelines)")
```

```
g_class
# --------------------
# 9) analyze unsatisfaction in "week days" - No obious pattern in unsatisfied ratio
ct_days <- data.frame(tapply(vis$satisfaction, list(vis$days, vis$satisfied=="Yes"), length))
colnames(ct_days) <- c("unsat_count","sat_count")
ct_days$days <- c("Mon","Tue","Wed","Thu","Fri","Sat","Sun")
ct_days$ttl_count <- ct_days$unsat_count + ct_days$sat_count
ct_days$avg_sat <- tapply(vis$satisfaction, vis$days, mean)
ct_days$sat_rate <- ct_days$sat_count/ct_days$ttl_count * 100
str(ct_days)
# --------------------
# 10) analyze unsatisfaction in "flight cancelled"
ct_cancel <- data.frame(tapply(vis$satisfaction, list(vis$cancel, vis$satisfied=="Yes"), length))
ct_cancel$Cancellation <- c("Not Cancelled", "Cancelled")
ct_cancel$Satisfied <- ct_cancel[ ,2]
ct_cancel$Unsatisfied <- ct_cancel[ ,1]
ct_cancel <- ct_cancel[ ,-1:-2]
ct_cancel$UnsatRate <- round(ct_cancel$Unsatisfied / (ct_cancel$Satisfied + ct_cancel$Unsatisfied), 4) * 100
# plot a bar chart for comparison
g <- ggplot(ct_cancel, aes(x=reorder(Cancellation, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 75)
g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)
g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("Flight Cancellation") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)
g_cancel <- g + ggtitle("Unsatisfied Ratio in Flight Cancellation (All Airlines)")
g_cancel
# --------------------
# 11) analyze unsatisfaction in "flight delayed > 5 min"
```

```
ct_delay <- data.frame(tapply(vis$satisfaction, list(vis$delay_arvl > 5, vis$satisfied=="Yes"), length))

ct_delay$Delayed <- c("< 5 min", "5 min +")

ct_delay$Satisfied <- ct_delay[ ,2]

ct_delay$Unsatisfied <- ct_delay[ ,1]

ct_delay <- ct_delay[ ,-1:-2]

ct_delay$UnsatRate <- round(ct_delay$Unsatisfied / (ct_delay$Satisfied + ct_delay$Unsatisfied), 4) * 100

# plot a bar chart for comparison

g <- ggplot(ct_delay, aes(x=reorder(Delayed, -UnsatRate), y=UnsatRate)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 65)

g <- g + geom_text(aes(label=UnsatRate), vjust=-0.5)

g <- g + ylab("Percentage of Unsatisfied Customers") + xlab("Arrival Delayed Time in Minutes") + geom_hline(yintercept=AvgUnsatRate, linetype="dashed", color="red", size=1, alpha=0.7)

g_delay <- g + ggtitle("Unsatisfied Ratio in Arrivel Delay Time (All Airlines)")

g_delay

## Understand origin/ destination cities and Satisfaction

# origin city - satisfaction

str(air)

grep("Origin City", colnames(air)) # 18

grep("Origin State", colnames(air)) # 19

# cleanse data in origin city & state, sort in alphabetical order

ls_origin <- data.frame(unique(air[ , 18:19]))

ls_origin <- ls_origin[order(ls_origin$Origin.City), ]

ls_origin$Origin.City <- tolower(ls_origin$Origin.City)

ls_origin$Origin.State <- tolower(ls_origin$Origin.State)

ls_origin$Origin.City <- gsub("/"," ", ls_origin$Origin.City)

sort(unique(ls_origin$Origin.State))

# show how many locations with multiple city names

library(data.table)
```

```
ls_origin_m <- ls_origin[ls_origin$Origin.City %like% "/", ]
nrow(ls_origin[ls_origin$Origin.City %like% "/", ]) # 36 locations
fix(ls_origin)
# get the geo-coordinance through nominatim API
# test osmlocale
osmlocale("allentown bethlehem easton, pa")
ct_origin <- data.frame(tapply(air$Satisfaction, air$"Origin City", length))
ct_origin$origin_city <- tolower(rownames(ct_origin))
ct_origin$count <- ct_origin[ ,1]
ct_origin <- ct_origin[ ,-1]
ct_origin[order(ct_origin$origin_city), ]
# get geo-coordinance for all origin cities
ls_origin$locale <- osmlocale(ls_origin$Origin.City)
ct_origin$state <- ls_origin$Origin.State
ct_origin.sat <- data.frame(tapply(air$Satisfaction, air$"Origin City", mean))
ct_origin$avg_sat <- ct_origin.sat[ ,1]
ct_origin$satisfied <- ifelse(ct_origin$avg_sat < 3.5, "No", "Yes")
ct_origin$locale <- ls_origin$locale
summary(ct_origin)
tapply(ct_origin$origin_city, ct_origin$satisfied=="Yes", length)
# below 295 cities in one plot is visually not readable
# g <- ggplot(ct_origin, aes(x=reorder(origin_city, avg_sat), y=avg_sat)) + geom_bar(stat="identity", color="black", fill="gray", alpha=0.5) + ylim(0, 5)
# g <- g + geom_text(aes(label=avg_sat), vjust = -0.5)
# g <- g + ylab("Average Satisfaction Rate") + xlab("Origin City") + geom_hline(yintercept=AvgSat, linetype="dashed", color="steelblue", size=1, alpha=0.7)
# g_origin <- g + theme(axis.text.x = element_text(angle = 90, hjust = 1))
# g_origin
```

```
# functions of nominatim_osm() & osmlocale() loaded from another R script

# create a simple US map

df_usa <- map_data("state")

unique(df_usa$region)

gm <- ggplot(ct_origin, aes(map_id=state)) + geom_map(map=df_usa, fill="white", color="black", alpha=0.75) + expand_limits(x=df_usa$long, y=df_usa$lat)

gm <- gm + geom_point(data=ct_origin, aes(x=locale$lon, y=locale$lat, color=satisfied, size=count)) + scale_fill_gradient(low="steelblue1", high="steelblue4")

gm_origin <- gm + coord_map() + xlim(c(-130, -60)) + ylim(c(20, 50)) + ggtitle("Origin Cities: Average Satisfaction & Flight Count")

gm_origin

# create a linear model of satisfaction vs numbers of flights among origin cities

summary(ct_origin)

m_origin <- lm(formula = avg_sat ~ count, data=ct_origin)

summary(m_origin)

plot(ct_origin$count, ct_origin$avg_sat, xlab="Count of Flights from Origin City", ylab="Average Satisfaction")

abline(m_origin, col="steelblue")

# destination city - satisfaction

grep("Destination City", colnames(air)) # 20

grep("Destination State", colnames(air)) # 21

# cleanse data in destination city & state, sort in alphabetical order

ls_destination <- data.frame(unique(air[ ,20:21]))

ls_destination <- ls_destination[order(ls_destination$Destination.City)]

ls_destination$Destination.City <- tolower(ls_destination$Destination.City)

ls_destination$Destination.State <- tolower(ls_destination$Destination.State)


# show how many locations with multiple city names

nrow(ls_destination[ls_destination$Destination.City %like% "/", ])

fix(ls_destination)
```

```
ls_destination$locale <- osmlocale(ls_destination$Destination.City)

ct_destin <- data.frame(tapply(air$Satisfaction, air$"Destination City", length))

ct_destin$destin_city <- tolower(rownames(ct_destin))

ct_destin$count <- ct_destin[ ,1]

ct_destin <- ct_destin[ ,-1]

ct_destin.sat <- data.frame(tapply(air$Satisfaction, air$"Destination City", mean))

ct_destin$state <- ls_destination$Destination.State

ct_destin$avg_sat <- ct_destin.sat[ ,1]

ct_destin$satisfied <- ifelse(ct_destin$avg_sat < 3.5, "No", "Yes")

ct_destin$locale <- ls_destination$locale

gm <- ggplot(ct_destin, aes(map_id=state)) + geom_map(map=df_usa, fill="white", color="black", alpha=0.75) + expand_limits(x=df_usa$long, y=df_usa$lat)

gm <- gm + geom_point(data=ct_destin, aes(x=locale$lon, y=locale$lat, color=satisfied, size=count)) + scale_fill_gradient(low="steelblue1", high="steelblue4")

gm_destin <- gm + coord_map() + xlim(c(-130, -60)) + ylim(c(20, 50)) + ggtitle("Origin Cities: Average Satisfaction & Flight Count")

gm_destin

# create a linear model of satisfaction vs numbers of flights among origin cities

summary(ct_destin)

m_destin <- lm(formula = avg_sat ~ count, data=ct_destin)

summary(m_destin)

plot(ct_destin$count, ct_destin$avg_sat, xlab="Count of Flights from Destination City", ylab="Average Satisfaction")

abline(m_destin, col="steelblue")

# origin city/ destination city delay in min. vs satisfaction

grep("Arrival Delay in Minutes", colnames(air)) # 24

air_delay <- air[ ,c(1,18:21,24)]

air_delay <- na.omit(air_delay)

# origin city

ct_delay_origin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Origin City`, mean))
```

```
ct_delay_origin$ttlmin <- tapply(air_delay$`Arrival Delay in Minutes`, air_delay$`Origin City`, sum)

ct_delay_origin$origin_city <- tolower(rownames(ct_delay_origin))

colnames(ct_delay_origin) <- c("avg_sat", "total_min", "origin_city")

summary(ct_delay_origin)

m_delay_origin <- lm(formula=avg_sat ~ total_min, data=ct_delay_origin)

summary(m_delay_origin)

plot(ct_delay_origin$total_min, ct_delay_origin$avg_sat, xlab="Count of Total Delayed Minutes in Origin Cities", ylab="Average Satisfaction")

abline(m_delay_origin, col="steelblue")

# destination city

ct_delay_destin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Destination City`, mean))

ct_delay_destin$ttlmin <- tapply(air_delay$`Arrival Delay in Minutes`, air_delay$`Destination City`, sum)

ct_delay_destin$destin_city <- tolower(rownames(ct_delay_destin))

colnames(ct_delay_destin) <- c("avg_sat", "total_min", "destin_city")

m_delay_destin <- lm(formula=avg_sat ~ total_min, data=ct_delay_destin)

summary(m_delay_destin)

plot(ct_delay_destin$total_min, ct_delay_destin$avg_sat, xlab="Count of Total Delayed Minutes in Destination Cities", ylab="Average Satisfaction")

abline(m_delay_origin, col="steelblue")


# origin city/ destination city cancellation vs satisfaction

# origin city

ct_cancel_origin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Origin City`, mean))

ct_cancel_origin$count <- tapply(air$`Flight cancelled`, air$`Origin City`, length)

ct_cancel_origin$origin_city <- tolower(rownames(ct_cancel_origin))

colnames(ct_cancel_origin)[1:2] <- c("avg_sat","cancel_count")

summary(ct_cancel_origin$cancel_count)
```

```
m_cancel_origin <- lm(formula=avg_sat ~ cancel_count, data=ct_cancel_origin)

summary(m_cancel_origin)

plot(ct_cancel_origin$cancel_count, ct_cancel_origin$avg_sat, xlab="Count of Cancellation in Origin Cities", ylab="Average Satisfaction")

abline(m_cancel_origin, col="steelblue")

# origin city

ct_cancel_destin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Destination City`, mean))

ct_cancel_destin$count <- tapply(air$`Flight cancelled`, air$`Destination City`, length)

ct_cancel_destin$destin_city <- tolower(rownames(ct_cancel_destin))

colnames(ct_cancel_destin)[1:2] <- c("avg_sat","cancel_count")

summary(ct_cancel_destin$cancel_count)

m_cancel_destin <- lm(formula=avg_sat ~ cancel_count, data=ct_cancel_destin)

summary(m_cancel_destin)

plot(ct_cancel_destin$cancel_count, ct_cancel_destin$avg_sat, xlab="Count of Cancellation in Destination Cities", ylab="Average Satisfaction")

abline(m_cancel_destin, col="steelblue")
```

CODE FOR DESCRIPTIVE ANALYSIS – MAPS

```
# --------------------

## check, install, and load required packages

packages <- c("data.table", "ggplot2", "maps", "ggmap", "mapproj", "tidyverse")

package.check <- lapply(packages, FUN = function(x) {

  if (!require(x, character.only = TRUE)) {

    install.packages(x, dependencies = TRUE)

    library(x, character.only = TRUE)

  }

})

## clean up packages and package.check after checking the packages
```

```
rm(packages)
rm(package.check)
# --------------------
# load nominatim functions for geo location API
nominatim_osm <- function(address = NULL)
{
  if(suppressWarnings(is.null(address)))
    return(data.frame())
  tryCatch(
    d <- jsonlite::fromJSON(
      gsub('\\@addr\\@', gsub('\\s+', '\\%20', address),

'http://nominatim.openstreetmap.org/search/@addr@?format=json&addressdetails=0&limit=1')
    ), error = function(c) return(data.frame())
  )
  if(length(d) == 0) return(data.frame())
  return(data.frame(lon = as.numeric(d$lon), lat = as.numeric(d$lat)))
}

osmlocale<-function(addresses){
  d <- suppressWarnings(lapply(addresses, function(address) {
    #set the elapsed time counter to 0
    t <- Sys.time()
    #calling the nominatim OSM API
    api_output <- nominatim_osm(address)
    #get the elapsed time
    t <- difftime(Sys.time(), t, 'secs')
    #return data.frame with the input address, output of the nominatim_osm function and elapsed
time
```

```
    return(data.frame(address = address, api_output, elapsed_time = t))
  }) %>%
    #stack the list output into data.frame
    bind_rows() %>% data.frame())
  #output the data.frame content into console
  return(d)
}
# --------------------
# analyze factors of origin/ destination cities and customer satisfaction
# --------------------
# origin city - satisfaction
vis <- survey
str(vis)
# cleanse data in origin city & state, sort in alphabetical order
ls_origin <- data.frame(unique(vis[ , 20:21]))
ls_origin <- ls_origin[order(ls_origin$origin_city), ]
# check number of origin_city that contain multiple locations with "/"
ls_origin_m <- ls_origin[ls_origin$origin_city %like% "/", ]
nrow(ls_origin[ls_origin$origin_city %like% "/", ])
# result shows 36 origin_city with multiple locations
rm(ls_origin_m)
# fix the values manually
fix(ls_origin)
# count number of survey records of each origin city
ct_origin <- data.frame(tapply(vis$satisfaction, vis$origin_city, length))
ct_origin$count <- ct_origin[ ,1]
ct_origin$origin_city <- rownames(ct_origin)
ct_origin <- ct_origin[ ,-1]
```

```
# compute average satisfaction of each origin city
ct_origin$avg_sat <- tapply(vis$satisfaction, vis$origin_city, mean)
# sort if customers are overall satisfied from each origin city
ct_origin$satisfied <- ifelse(ct_origin$avg_sat < 3.5, "No", "Yes")
# get geo-coordinance for all origin cities
ls_origin$locale <- osmlocale(ls_origin$origin_city)
ct_origin$state <- ls_origin$origin_state
ct_origin$locale <- ls_origin$locale
# find origin cities and geo-coordinances for Cool & Young Airlines
vis_cx <- vis[vis$als=="Cool&Young Airlines Inc. - VX", ]
ls_origin_cx <- data.frame(unique(vis_cx[ ,20:21]))
colnames(ls_origin_cx)[2] <- "state"
# manually remove multiple location
fix(ls_origin_cx)

# find geo-coordinance for Cool & Young origin cities
ls_origin_cx$locale <- osmlocale(ls_origin_cx$origin_city)
# create a simple US map
df_usa <- map_data("state")
unique(df_usa$region)
# plot a map of origin city flight count vs satisfied
gm <- ggplot(ct_origin, aes(map_id=state)) + geom_map(map=df_usa, fill="white", color="black", alpha=0.75) + expand_limits(x=df_usa$long, y=df_usa$lat)
gm <- gm + geom_point(data=ct_origin, aes(x=locale$lon, y=locale$lat, color=satisfied, size=count)) + scale_fill_gradient(low="steelblue1", high="steelblue4")
gm <- gm + coord_map() + xlim(c(-130, -60)) + ylim(c(20, 50)) + geom_point(data=ls_origin_cx, aes(x=locale$lon, y=locale$lat), shape="v", size=8, alpha=0.5)
gm_origin <- gm + ggtitle("Origin Cities - Flight Count vs Satisfied")
gm_origin
```

```
# ----------
# create a linear model of satisfaction vs numbers of flights among origin cities
summary(ct_origin)
m_origin <- lm(formula = avg_sat ~ count, data=ct_origin)
summary(m_origin)
plot(ct_origin$count, ct_origin$avg_sat, xlab="Count of Flights from Origin City", ylab="Average Satisfaction")
abline(m_origin, col="steelblue")
# --------------------
# destination city - satisfaction
# cleanse data in destination city & state, sort in alphabetical order
ls_destin <- data.frame(unique(vis[ ,22:23]))
ls_destin <- ls_destin[order(ls_destin$destin_city)]


# show how many locations with multiple city names
nrow(ls_destin[ls_destin$destin_city %like% "/", ])
# result shows 36 cities with multiple locations, manually fix
fix(ls_destin)
# get geo-coordinance for all destination cities
ls_destin$locale <- osmlocale(ls_destin$destin_city)
colnames(ls_destin)[2] <- "state"
# count number of survey records of each destination city
ct_destin <- data.frame(tapply(vis$satisfaction, vis$destin_city, length))
ct_destin$destin_city <- rownames(ct_destin)
ct_destin$state <- ls_destin$state
ct_destin$count <- ct_destin[ ,1]
ct_destin <- ct_destin[ ,-1]
# compute average satisfaction of each destination city
```

```
ct_destin$avg_sat <- tapply(vis$satisfaction, vis$destin_city, mean)

ct_destin$avg_sat <- ct_destin[ ,4]

# sort if customers are overall satisfied from each destination city

ct_destin$satisfied <- ifelse(ct_destin$avg_sat < 3.5, "No", "Yes")

ct_destin$locale <- ls_destin$locale

# find origin cities and geo-coordinances for Cool & Young Airlines

ls_destin_cx <- data.frame(unique(vis_cx[ ,22:23]))

colnames(ls_destin_cx)[2] <- "state"

# manually remove multiple location

fix(ls_destin_cx)


# find geo-coordinance for Cool & Young origin cities

ls_destin_cx$locale <- osmlocale(ls_destin_cx$destin_city)

# plot a map of destination city flight count vs satisfied

gm <- ggplot(ct_destin, aes(map_id=state)) + geom_map(map=df_usa, fill="white", color="black", alpha=0.75) + expand_limits(x=df_usa$long, y=df_usa$lat)

gm <- gm + geom_point(data=ct_destin, aes(x=locale$lon, y=locale$lat, color=satisfied, size=count)) + scale_fill_gradient(low="steelblue1", high="steelblue4")

gm <- gm + coord_map() + xlim(c(-130, -60)) + ylim(c(20, 50)) + geom_point(data=ls_destin_cx, aes(x=locale$lon, y=locale$lat), shape="v", size=8, alpha=0.5)

gm_destin <- gm + ggtitle("Destination Cities - Flight Count & Satisfied")

gm_destin

# ----------

# create a linear model of satisfaction vs numbers of flights among origin cities

summary(ct_destin)

m_destin <- lm(formula = avg_sat ~ count, data=ct_destin)

summary(m_destin)

plot(ct_destin$count, ct_destin$avg_sat, xlab="Count of Flights from Destination City", ylab="Average Satisfaction")

abline(m_destin, col="steelblue")
```

```
# origin city/ destination city delay in min. vs satisfaction

grep("Arrival Delay in Minutes", colnames(air)) # 24

air_delay <- air[ ,c(1,18:21,24)]

air_delay <- na.omit(air_delay)

# origin city

ct_delay_origin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Origin City`, mean))

ct_delay_origin$ttlmin <- tapply(air_delay$`Arrival Delay in Minutes`, air_delay$`Origin City`, sum)

ct_delay_origin$origin_city <- tolower(rownames(ct_delay_origin))

colnames(ct_delay_origin) <- c("avg_sat", "total_min", "origin_city")

summary(ct_delay_origin)

m_delay_origin <- lm(formula=avg_sat ~ total_min, data=ct_delay_origin)

summary(m_delay_origin)

plot(ct_delay_origin$total_min, ct_delay_origin$avg_sat, xlab="Count of Total Delayed Minutes in Origin Cities", ylab="Average Satisfaction")

abline(m_delay_origin, col="steelblue")

# destination city

ct_delay_destin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Destination City`, mean))

ct_delay_destin$ttlmin <- tapply(air_delay$`Arrival Delay in Minutes`, air_delay$`Destination City`, sum)

ct_delay_destin$destin_city <- tolower(rownames(ct_delay_destin))

colnames(ct_delay_destin) <- c("avg_sat", "total_min", "destin_city")

m_delay_destin <- lm(formula=avg_sat ~ total_min, data=ct_delay_destin)

summary(m_delay_destin)

plot(ct_delay_destin$total_min, ct_delay_destin$avg_sat, xlab="Count of Total Delayed Minutes in Destination Cities", ylab="Average Satisfaction")

abline(m_delay_origin, col="steelblue")

# origin city/ destination city cancellation vs satisfaction

# origin city
```

```
ct_cancel_origin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Origin City`, mean))

ct_cancel_origin$count <- tapply(air$`Flight cancelled`, air$`Origin City`, length)

ct_cancel_origin$origin_city <- tolower(rownames(ct_cancel_origin))

colnames(ct_cancel_origin)[1:2] <- c("avg_sat","cancel_count")

summary(ct_cancel_origin$cancel_count)

m_cancel_origin <- lm(formula=avg_sat ~ cancel_count, data=ct_cancel_origin)

summary(m_cancel_origin)

plot(ct_cancel_origin$cancel_count, ct_cancel_origin$avg_sat, xlab="Count of Cancellation in Origin Cities", ylab="Average Satisfaction")

abline(m_cancel_origin, col="steelblue")


# origin city

ct_cancel_destin <- data.frame(tapply(air_delay$Satisfaction, air_delay$`Destination City`, mean))

ct_cancel_destin$count <- tapply(air$`Flight cancelled`, air$`Destination City`, length)

ct_cancel_destin$destin_city <- tolower(rownames(ct_cancel_destin))

colnames(ct_cancel_destin)[1:2] <- c("avg_sat","cancel_count")

summary(ct_cancel_destin$cancel_count)

m_cancel_destin <- lm(formula=avg_sat ~ cancel_count, data=ct_cancel_destin)

summary(m_cancel_destin)

plot(ct_cancel_destin$cancel_count, ct_cancel_destin$avg_sat, xlab="Count of Cancellation in Destination Cities", ylab="Average Satisfaction")

abline(m_cancel_destin, col="steelblue")
```

CODE FOR DESCRIPTIVE ANALYSIS - LINEAR MODELS & PLOTS

```
# --------------------

## check, install, and load required packages

packages <- c("data.table", "ggplot2", "maps", "ggmap", "mapproj", "tidyverse")

package.check <- lapply(packages, FUN = function(x) {
```

```
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})
## clean up packages and package.check after checking the packages
rm(packages)
rm(package.check)
# --------------------
# analyze factors of number of survey records in origin/ destination cities and customer satisfaction
# create a linear model of satisfaction vs numbers of flights among origin cities
summary(ct_origin)
m_origin <- lm(formula = avg_sat ~ count, data=ct_origin)
summary(m_origin)
plot(ct_origin$count, ct_origin$avg_sat, xlab="Count of Flights from Origin City", ylab="Average Satisfaction")
abline(m_origin, col="steelblue")
# create a linear model of satisfaction vs numbers of flights among destination cities
summary(ct_destin)
m_destin <- lm(formula = avg_sat ~ count, data=ct_destin)
summary(m_destin)
plot(ct_destin$count, ct_destin$avg_sat, xlab="Count of Flights from Destination City", ylab="Average Satisfaction")
abline(m_destin, col="steelblue")
# --------------------
# analyze factors of delay in min. in origin/ destination cities and customer satisfaction
# create a new data frame for easily access of needed records
vis_delay <- vis[ ,c(1,15:17,20:24)]
# origin city
ct_delay_origin <- data.frame(tapply(vis_delay$satisfaction, vis_delay$origin_city, mean))
```

```
ct_delay_origin$ttlmin <- tapply(vis_delay$delay_arvl, vis_delay$origin_city, sum)

ct_delay_origin$origin_city <- rownames(ct_delay_origin)

colnames(ct_delay_origin) <- c("avg_sat", "total_min", "origin_city")

# create a linear model and plot

summary(ct_delay_origin)

m_delay_origin <- lm(formula=avg_sat ~ total_min, data=ct_delay_origin)

summary(m_delay_origin)

plot(ct_delay_origin$total_min, ct_delay_origin$avg_sat, xlab="Count of Total Delayed Minutes in Origin Cities", ylab="Average Satisfaction")

abline(m_delay_origin, col="steelblue")

# destination city

ct_delay_destin <- data.frame(tapply(vis_delay$satisfaction, vis_delay$destin_city, mean))

ct_delay_destin$ttlmin <- tapply(vis_delay$delay_arvl, vis_delay$destin_city, sum)

ct_delay_destin$destin_city <- rownames(ct_delay_destin)

colnames(ct_delay_destin) <- c("avg_sat", "total_min", "destin_city")

# create a linear model and plot

m_delay_destin <- lm(formula=avg_sat ~ total_min, data=ct_delay_destin)

summary(m_delay_destin)

plot(ct_delay_destin$total_min, ct_delay_destin$avg_sat, xlab="Count of Total Delayed Minutes in Destination Cities", ylab="Average Satisfaction")

abline(m_delay_origin, col="steelblue")

# --------------------

# analyze factors of flight cancellation in origin/ destination cities and customer satisfaction

# origin city

ct_cancel_origin <- data.frame(tapply(vis_delay$satisfaction, vis_delay$origin_city, mean))

ct_cancel_origin$count <- tapply(vis$cancel, vis$origin_city, length)

colnames(ct_cancel_origin)[1:2] <- c("avg_sat","cancel_count")

summary(ct_cancel_origin$cancel_count)

# # create a linear model and plot

m_cancel_origin <- lm(formula=avg_sat ~ cancel_count, data=ct_cancel_origin)
```

```
summary(m_cancel_origin)

plot(ct_cancel_origin$cancel_count, ct_cancel_origin$avg_sat, xlab="Count of Cancellation in Origin Cities", ylab="Average Satisfaction")

abline(m_cancel_origin, col="steelblue")

# destination city

ct_cancel_destin <- data.frame(tapply(vis_delay$satisfaction, vis_delay$destin_city, mean))

ct_cancel_destin$count <- tapply(vis$cancel, vis$destin_city, length)

colnames(ct_cancel_destin)[1:2] <- c("avg_sat","cancel_count")

summary(ct_cancel_destin$cancel_count)

# # create a linear model and plot

m_cancel_destin <- lm(formula=avg_sat ~ cancel_count, data=ct_cancel_destin)

summary(m_cancel_destin)

plot(ct_cancel_destin$cancel_count, ct_cancel_destin$avg_sat, xlab="Count of Cancellation in Destination Cities", ylab="Average Satisfaction")

abline(m_cancel_destin, col="steelblue")
```

CODE FOR PREDICTIVE ANALYSIS - LINEAR MODEL

```
## check, install, and load required packages

packages <- c("ggcorrplot")

package.check <- lapply(packages, FUN = function(x) {

  if (!require(x, character.only = TRUE)) {

    install.packages(x, dependencies = TRUE)

    library(x, character.only = TRUE)

  }

})

## make a function to return RMSE

rmse <- function(t,p){

  rt <- sqrt(mean((t-p)^2))

  return(rt)
```

```
}
# find correlations of variables in data of Cool & Young
str(sv_vx)
# need to convert all factorial values into numeric
sv_vx_n <- data.frame(lapply(sv_vx, function(x) as.numeric(x)))
str(sv_vx_n)
corr <- round(cor(sv_vx_n),1)
ggcorrplot(corr, hc.order = TRUE,
    type = "lower",
    lab = TRUE,
    lab_size = 3,
    method = "circle",
    colors = c("red", "white", "steelblue"),
    title = "Correlogram of Satisfaction Survey Variables (Cool & Young)",
    ggtheme = theme_bw)
# need to convert all factorial values into numeric
sv_all_n <- data.frame(lapply(sv_all, function(x) as.numeric(x)))
str(sv_all_n)
corr <- round(cor(sv_all_n),1)
ggcorrplot(corr, hc.order = TRUE,
    type = "lower",
    lab = TRUE,
    lab_size = 3,
    method = "circle",
    colors = c("red", "white", "steelblue"),
    title = "Correlogram of Satisfaction Survey Variables (All Airlines)",
    ggtheme = theme_bw)
# The plot shows strong correlations in between input variables of fly_time & fly_dist, delay_dept
& delay_arvl
```

```
# To build better linear model, these two pairs of variables need to combined
# create variable fly_x = fly_dist * fly_time
sv_vx$fly_x <- sv_vx$fly_dist * sv_vx$fly_time
# create variable delay = delay_arvl - delay_dept
sv_vx$delay <- sv_vx$delay_arvl - sv_vx$delay_dept
# drop variables fly_time, fly_dist, delay_dept, delay_arvl
sv_vx <- sv_vx[ ,c(-15:-16, -18:-19)]
## create the most parsimonious linear regression models
## for Cool & Young Airlines, Inc - with all variables
lm_vx <-lm(formula=satisfaction~., data=sv_vx)
summary(lm_vx)
summary(lm_vx)$adj.r.squared  # 0.4219334
# apply step, backward, to pick the most parsimonious variables
step(lm_vx, data=sv_vx, direct="backward")
# AIC results:
# lm(formula = satisfaction ~ al_status + age + gender + fly_pa + type, data = sv_vx)
lmp_vx <- lm(formula = satisfaction ~ al_status + age + gender + fly_pa + type, data = sv_vx)
summary(lmp_vx)
summary(lmp_vx)$adj.r.squared
# create a linear regression model for Cool & Young Ailines, Inc.
# create vectors to properly store Coefficients for categorical variables of the linear model
coef_status <- c(0,lmp_vx$coefficients[2],lmp_vx$coefficients[3],lmp_vx$coefficients[4])
coef_type <- c(0,lmp_vx$coefficients[8],lmp_vx$coefficients[9])
coef_gender <- c(0,lmp_vx$coefficients[6])
# ceate a new data frame for regression prediction
sv_vx_lp <- sv_vx
sv_vx_lp$prd <- lmp_vx$coefficients[1] + coef_status[as.numeric(sv_vx_lp$al_status)] +
lmp_vx$coefficients[5]*sv_vx_lp$age + coef_gender[as.numeric(sv_vx_lp$gender)] +
lmp_vx$coefficients[7] * sv_vx_lp$fly_pa + coef_type[as.numeric(sv_vx_lp$type)]
```

```
# check the value of root mean square error

rmse_val <- rmse(sv_vx_lp$satisfaction, sv_vx_lp$prd)

rmse_val
```

```
# rmse value is a great measure to gauge the errors of the predictive outcomes, but not really practical in telling how accurate the model is for a non-technical person

# to make it easier to understand how accurate this model predicts, let's take the following actions:

# 1) round the predicted value up to one decimal;

sv_vx_lp$prd_rnd <- round(sv_vx_lp$prd, 1)

# 2) set a tolerance that if the predicted value is within 0.5, it is considered accurate

sv_vx_lp$correct <- ifelse(abs(sv_vx_lp$prd_rnd - sv_vx_lp$satisfaction) >!0.5, 1,0)

# 3) caculate the accuracy ratio

accuracy_lm <- sum(sv_vx_lp$correct==1)/nrow(sv_vx_lp)

accuracy_lm
```

CODE FOR PREDICTIVE ANALYSIS - NAIVE BAYES CLASSIFICATION MODEL

```
## check, install, and load required packages

packages <- c("e1071")

package.check <- lapply(packages, FUN = function(x) {

  if (!require(x, character.only = TRUE)) {

    install.packages(x, dependencies = TRUE)

    library(x, character.only = TRUE)

  }

})

## make a function to generate random row indices of an input data set

randinx <- function(df){

  rt <- sample(1:nrow(df))

  return(rt)
```

```
}
## make a function to return a cutpoint at 2/3 of input data set
cutpoint <- function(df){
  n <- nrow(df)
  cp <- floor(n*2/3)
  return(cp)
}
## prepare train/ test data sets for "Cool&Young Airlines Inc. - VX"
# label satisfaction as unsatisfied [0, 3] and satisfied [3.5, 5]
sv_vx$satisfied <- as.factor(ifelse(sv_vx$satisfaction < 3.5, 0, 1))
# randomly select 2/3 for train, 1/3 for test
idx <- randinx(sv_vx)
vx_tr <- sv_vx[idx[1:cutpoint(sv_vx)], ]
vx_pr <- sv_vx[idx[(cutpoint(sv_vx)+1):nrow(sv_vx)], ]
# train the algorithm to generate output
nb_vx_out <-naiveBayes(satisfied~al_status + age + gender + sensitivity + fly_yrs + fly_pa +
fly_other + type + cards + shop + eat_drink + class + days + delay_dept + delay_arvl + cancel +
fly_time + fly_dist,
                        data=vx_tr)
nb_vx_out
# predit satisfaction of test data based on trained data
vx_pr$prd <- predict(nb_vx_out, vx_pr)
# create a table with observed and predicted values for comparison
table(vx_pr[ ,c(20, 21)])
# check accuracy of predicted classification
accuracy <- sum(vx_pr$satisfied==vx_pr$prd)/nrow(vx_pr)
accuracy
```

CODE FOR PREDICTIVE ANALYSIS – KSVM MODEL

```
## check, install, and load required packages
packages <- c("kernlab")
package.check <- lapply(packages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})
## make a function to generate random row indices of an input data set
randinx <- function(df){
  rt <- sample(1:nrow(df))
  return(rt)
}
## make a function to return a cutpoint at 2/3 of input data set
cutpoint <- function(df){
  n <- nrow(df)
  cp <- floor(n*2/3)
  return(cp)
}
## make a function to return RMSE
rmse <- function(t,p){
  rt <- sqrt(mean((t-p)^2))
  return(rt)
}
## prepare train/ test data sets for "Cool&Young Airlines Inc. - VX"
idx <- randinx(sv_vx)
vx_tr <- sv_vx[idx[1:cutpoint(sv_vx)], ]
```

```
vx_pr <- sv_vx[idx[(cutpoint(sv_vx)+1):nrow(sv_vx)], ]

# train the algorithm to generate output

svm_vx_out <- ksvm(satisfaction~., data=vx_tr, kernal = "rbfdot", kpar = "automatic", C=80, cross=5, prob.model=TRUE)

svm_vx_out

# predit satisfaction of test data based on trained data

vx_pr$prd <- predict(svm_vx_out, vx_pr)

# check the value of root mean square error

rmse_val <- rmse(vx_pr$satisfaction, vx_pr$prd)

rmse_val


# calculate the relative accuracy

# 1) round the predicted value up to one decimal;

vx_pr$prd_rnd <- round(vx_pr$prd, 1)

# 2) set a tolerance that if the predicted value is within 0.5, it is considered accurate

vx_pr$correct <- ifelse(abs(vx_pr$prd_rnd - vx_pr$satisfaction) >!0.5, 1,0)

# 3) caculate the accuracy ratio

accuracy_ksvm <- sum(vx_pr$correct==1)/nrow(vx_pr)

accuracy_ksvm
```

*** End of R Script Code ***

*** End of Report ***