IST 687

Winter 2021

Group 2

Andrew Klassen

Mark Scholz

Zachary Chipman

Table of Contents

3	BACKGROUND
4	BUSINESS QUESTIONS
5	DATA CLEANING, TRANSFORMATION, ARCHITECTURE
7	PROFILE VISUALIZATIONS
13	SATISFACTION VISUALIZATIONS
20	MODEL DEVELOPMENT
28	INTERPRETATION OF RESULTS
29	CODE

Background

Our team was tasked to review a survey of 129,543 responses taken from January to March of 2014 to determine the factors that most impact customer satisfaction across 14 different airlines. The primary objective was to identify areas for industry-level improvement, which will have the greatest impact on improving long-term satisfaction.

We started by asking certain business questions that we felt were relevant to our research and proceeded to clean the data. We then explored the data by graphing satisfaction across the different demographics/airlines and developed models to further analyze them. Finally, we interpreted our results and provided our conclusions/recommendations to the industry.

Business Questions

- What is the profile of an airline customer?
- How do each of the airlines compare to each other on satisfaction?
- Are personal travelers more or less satisfied when compared to business travelers? How about when customers use their rewards? Is the satisfaction of these different customer segments driven by different factors?
- Which attributes are key drivers of airline customer satisfaction? How can we use this information to improve satisfaction ratings over time?
    - How impactful are departure and arrival delays in reduced customer satisfaction? Which is more impactful? At approximately what minute delay does satisfaction fall dramatically?
    - Is airline status impactful in driving satisfaction? Should airlines continue investment in this area?

Data Cleaning, Transformation, Architecture

1. Replace blanks with NAs:

   Records with NAs have relevant data elsewhere, particularly on satisfaction. Most of these records with NAs are customers with a cancelled flight. We know they are likely highly dissatisfied, and feel we should keep them in the data to not artificially increase the satisfaction rating at the overall level.

2. Trim the white space:

   Helps when referencing the "tapply" function.

3. Transform column names:

   New names without spaces are easier to reference.

   (Place images of revised column names here)

4. Remove 9 cases with misrepresented satisfaction ratings:

   They were removed since satisfaction is our key variable (ask group about this).

5. Impute the values of the "Percent of Flight with other Airlines" column that were greater than 100%:

   They now read as 100% because, in this instance, numbers over 100% are nonsensical.

6. Remove records that do not have a flight time or arrival delay metric AND do not say their flight was cancelled:

   These records are not intuitive and therefore not useful.

7. Added a "Unique Identifier" column to the dataframe:

   A column of numbers from 1 to 129,543

8. Added an initial set of NEW variables for analysis:

   - Loyalty Member / Non-Member
   - Shopper / Non-Shopper
   - Diner / Non-Diner

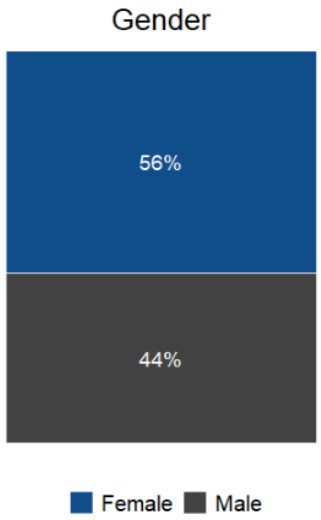
- Satisfaction coded (4-5, 3, 1-2)
- Arrival delay coded (0, 1-9, 10-59, 60+)
- Departure delay coded (0, 1-9, 10-59, 60+)

9. For modeling we converted categorical variables to binary, and satisfaction to binary (4-5) vs. (1-3) for our classification models.

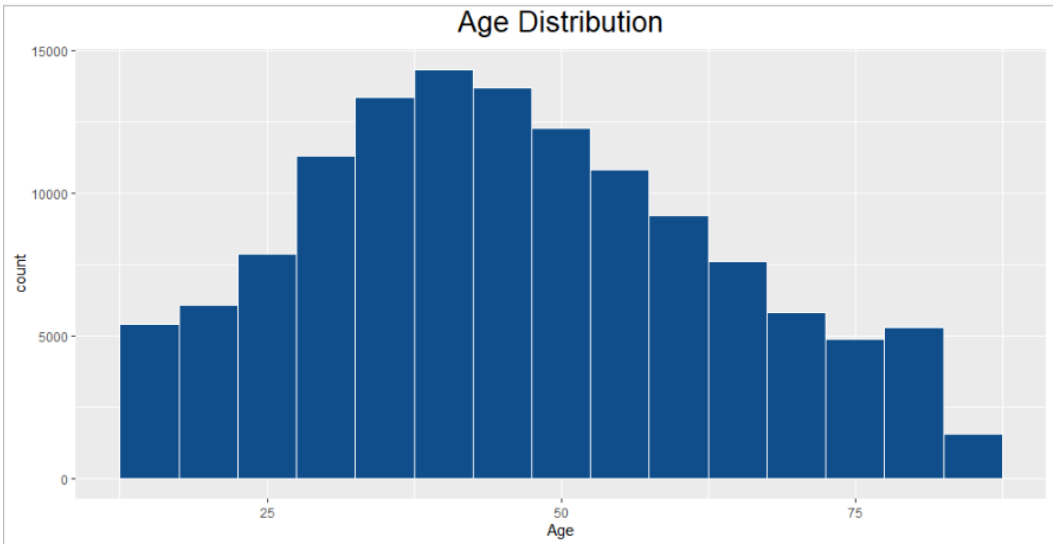
Profiling Visualizations

Demographic Profile:

A higher proportion of airline customers are female (56%).

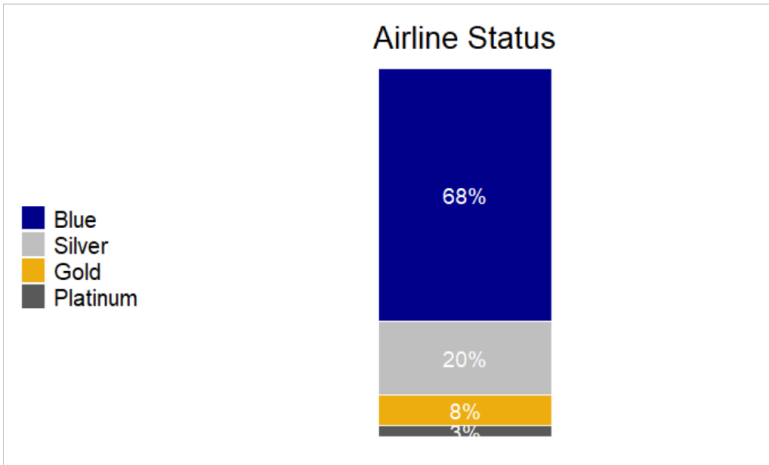


Ages 35-45 are the most common.

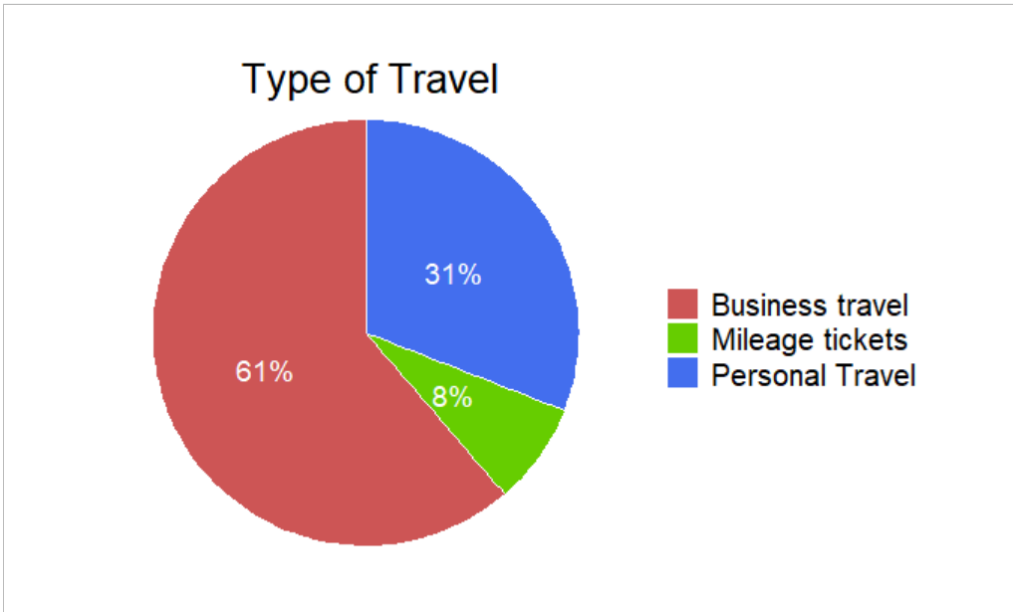


Airline Usage Profile:

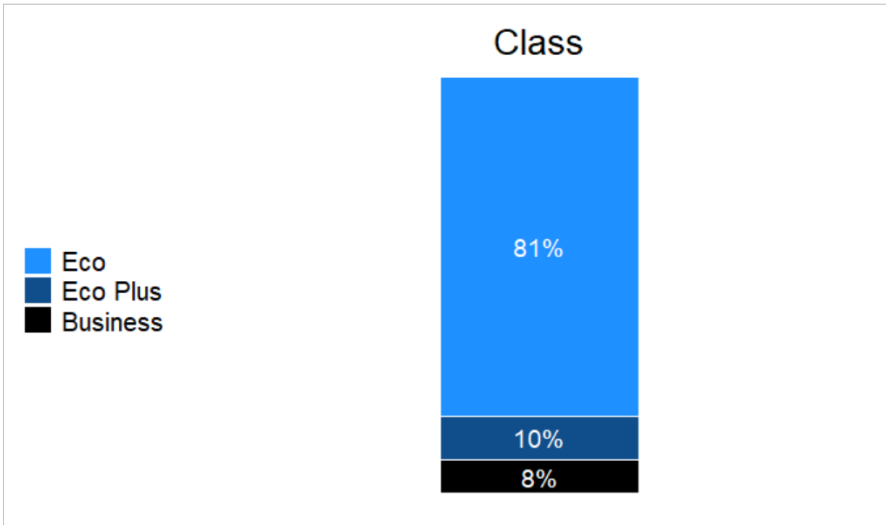
Over two-thirds (68%) of customers are “Blue” status; the next most common status is “Silver” at 20%.



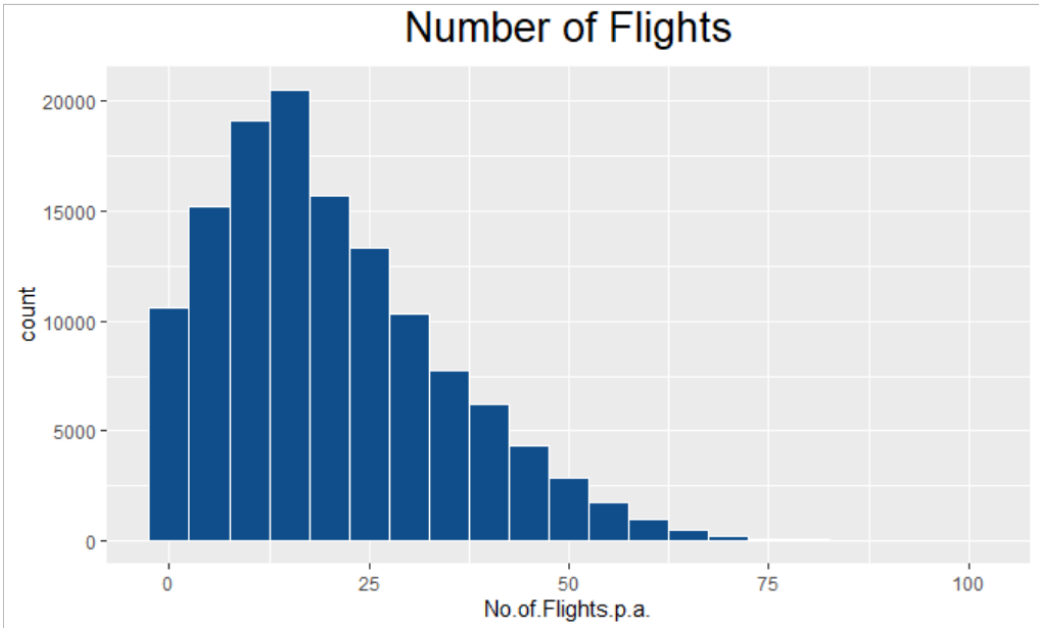
Business travel makes up the majority of flights.



8 in 10 passengers opt for economy class.

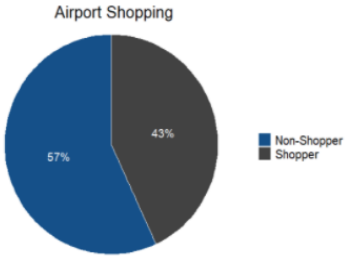


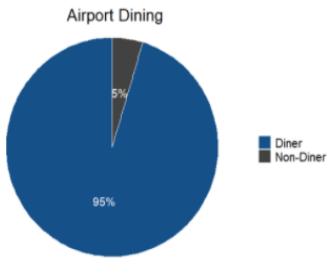
The total number of flights these customers have taken has a positive skew, centering on 15-20.



Additional Feature Usage Profile:

We see airline customers are split when it comes to loyalty card membership and shopping at the airport. Non-diners are rare, a subgroup that is less important for analysis given its lower base size and share of customers.

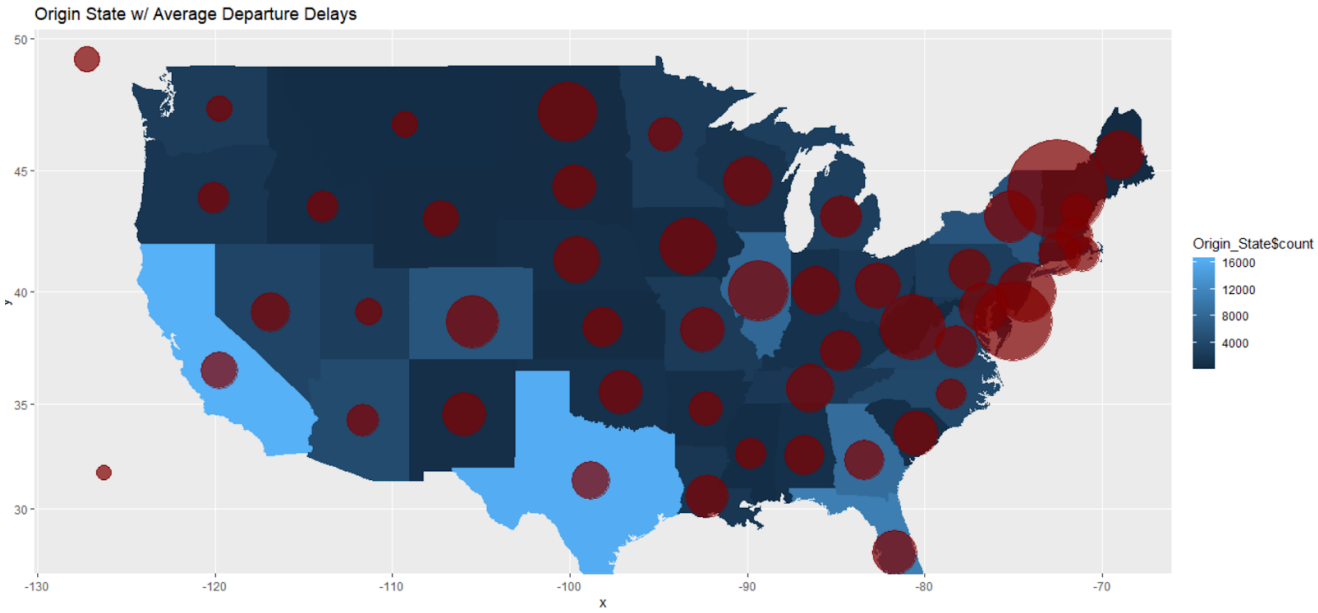




Origin/Destination State Frequency & Average Departure/Arrival Delays:

One of our data questions asked how impactful departure and arrival delays were to customer satisfaction. To help answer this question, we wanted to pinpoint the locations that have the highest average delays. On the maps below, delays are marked by red circles, with the larger circles being states with the highest average delays.

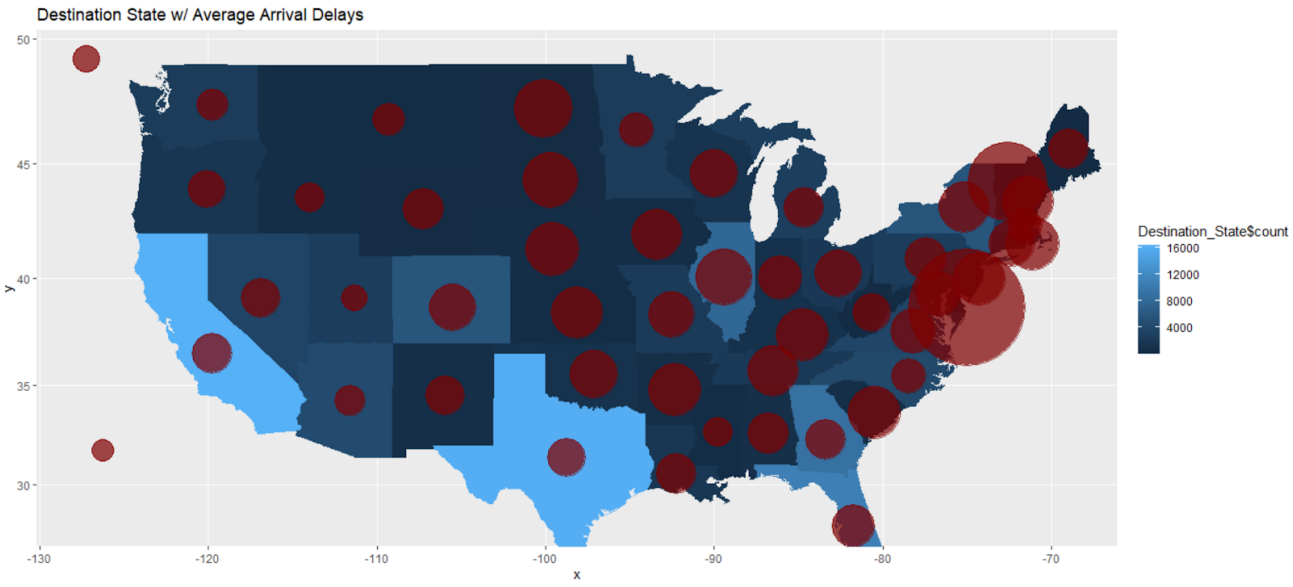
CA & TX have the most flights originating, followed by FL. VT, DE, and WV have significant room for improvement on departure delays. The Northeast is more likely to have delays, perhaps impacted by weather conditions.



On average, the most delayed departure states are as follows...

	Departure.Delay.in.Minutes	Origin.State
Vermont	35.740741	Vermont
Delaware	28.400000	Delaware
West Virginia	23.581818	West Virginia
Illinois	21.945026	Illinois
North Dakota	21.389121	North Dakota
New Jersey	21.305404	New Jersey
Iowa	20.425134	Iowa
Colorado	18.863075	Colorado
New York	18.680824	New York
Maryland	18.084331	Maryland

A similar story emerges for destination states and arrival delays, with DE demonstrated the most delays among flights headed there.



On average, the most delayed arrival states are as follows...

	Arrival.Delay.in.Minutes	Destination.State
Delaware	43.125000	Delaware
Vermont	29.281690	Vermont
North Dakota	21.541322	North Dakota
Illinois	20.784443	Illinois
South Dakota	20.466942	South Dakota
Nebraska	19.836735	Nebraska
Rhode Island	19.590226	Rhode Island
Arkansas	19.518639	Arkansas
Kentucky	19.455682	Kentucky
South Carolina	19.300855	South Carolina

Satisfaction Visualizations

Satisfaction was captured on a 5 point scale, with an overall average of 3.379.

Airline Sample Size:

This confirms we have sufficient data on each airline for analysis.

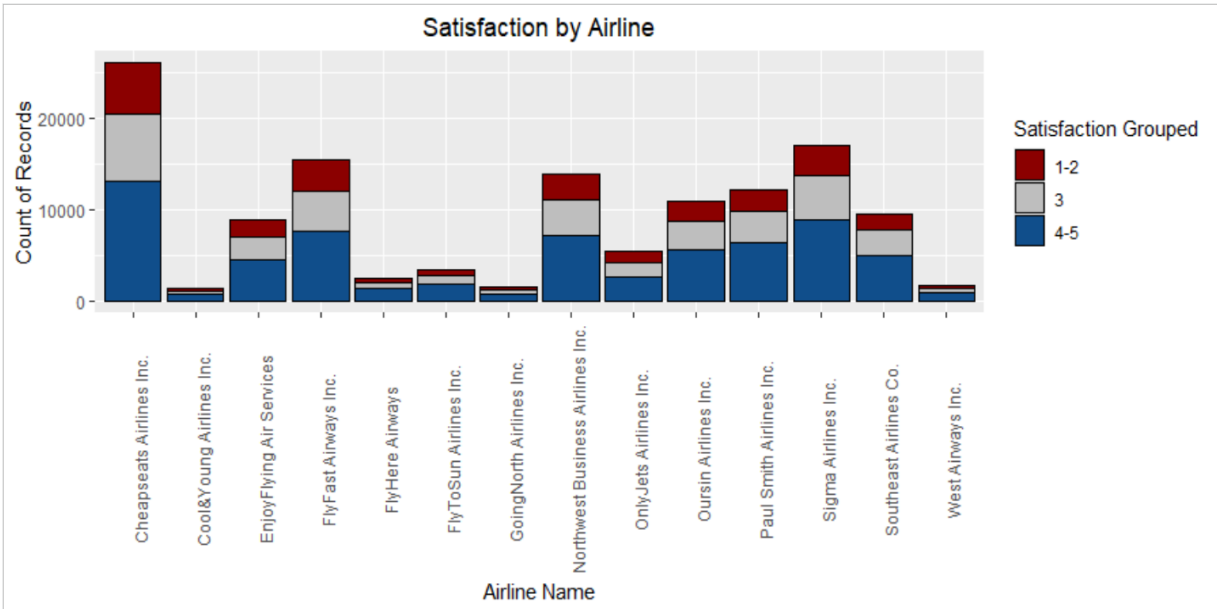
```
> stack(tapply(df$Unique, df$Airline.Name, length))
   values                              ind
1   25985         Cheapseats Airlines Inc.
2    1281         Cool&Young Airlines Inc.
3    8903         EnjoyFlying Air Services
4   15356             FlyFast Airways Inc.
5    2474                   FlyHere Airways
6    3392           FlyToSun Airlines Inc.
7    1568         GoingNorth Airlines Inc.
8   13787 Northwest Business Airlines Inc.
9    5382           OnlyJets Airlines Inc.
10  10950             Oursin Airlines Inc.
11  12207         Paul Smith Airlines Inc.
12  17018              Sigma Airlines Inc.
13   9555           Southeast Airlines Co.
14   1685                West Airways Inc.
```

Mean Airline Satisfaction:

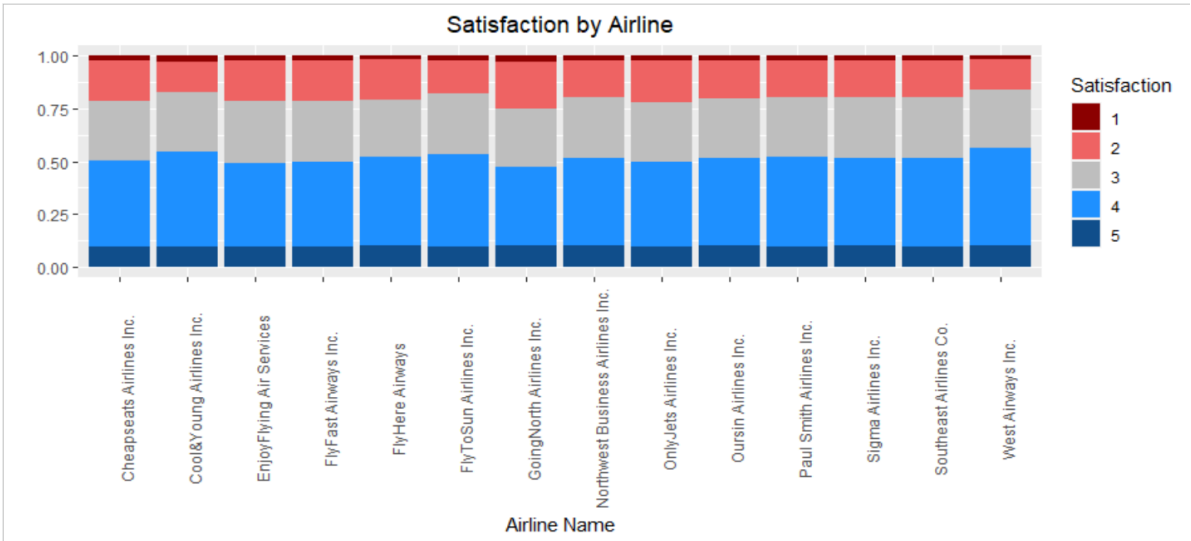
```
> stack(tapply(df$Satisfaction, df$Airline.Name, round.mean))
   values                              ind
1    3.36         Cheapseats Airlines Inc.
2    3.44         Cool&Young Airlines Inc.
3    3.36         EnjoyFlying Air Services
4    3.35             FlyFast Airways Inc.
5    3.40                   FlyHere Airways
6    3.42           FlyToSun Airlines Inc.
7    3.30         GoingNorth Airlines Inc.
8    3.39 Northwest Business Airlines Inc.
9    3.35           OnlyJets Airlines Inc.
10   3.39             Oursin Airlines Inc.
11   3.40         Paul Smith Airlines Inc.
12   3.40              Sigma Airlines Inc.
13   3.40           Southeast Airlines Co.
14   3.49                West Airways Inc.
```

Satisfaction By Airline:

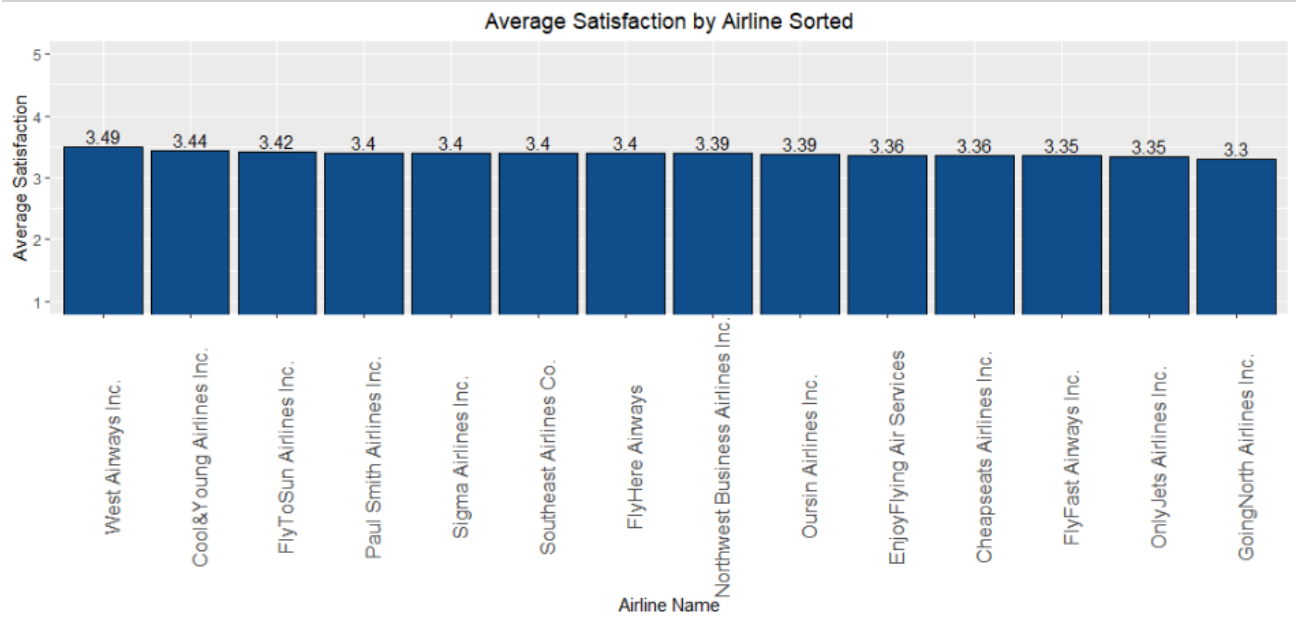
Cheapseats has the largest number of records, while others such as Cool&Young & GoingNorth have fewer survey completes. Customer satisfaction is roughly similar (in proportion) across airlines, each showing room for improvement.



Viewing the data proportionally, we see that among the customer base for each airline, ~50% are at least somewhat satisfied (4-5) across the set. That said, we can note that GoingNorth customers are most dissatisfied, while West Airways and Cool&Young outperform the competition.

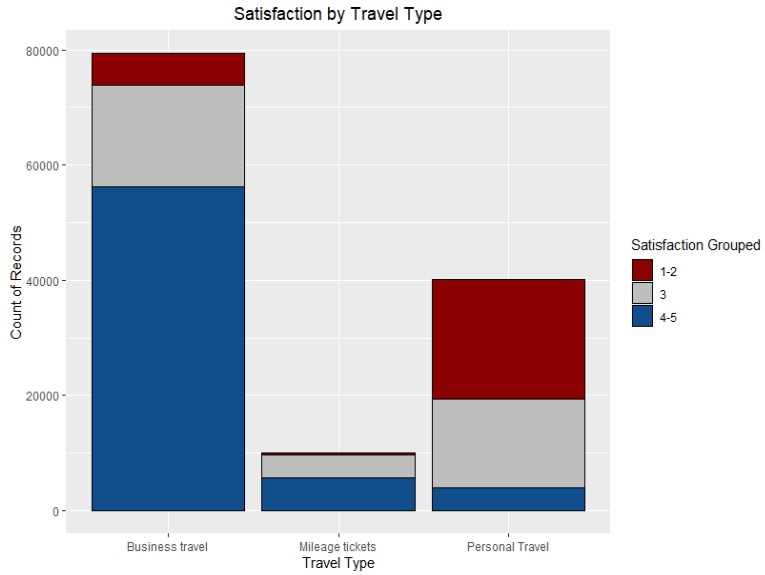


This visualization shows the average satisfaction across airlines sorted by performance confirms significant differences are not present. Therefore, the analysis that follows is on the airline industry as a whole, keeping these airlines aggregated.



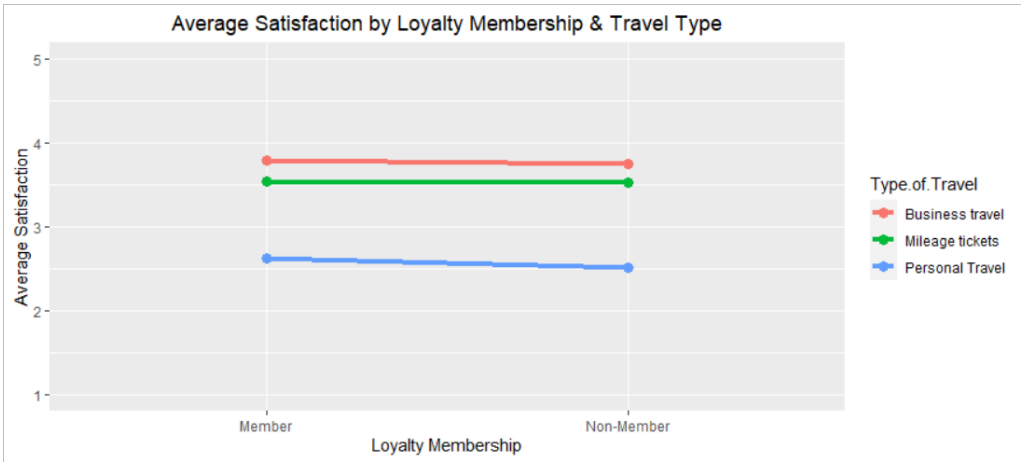
Satisfaction by Traveler Type:

Mileage Tickets and Business Travel have proportionally greater satisfaction; with those using Mileage Tickets having the lowest proportion of dissatisfied customers. This chart also shows that the largest customer base across all airlines is Business Travelers.



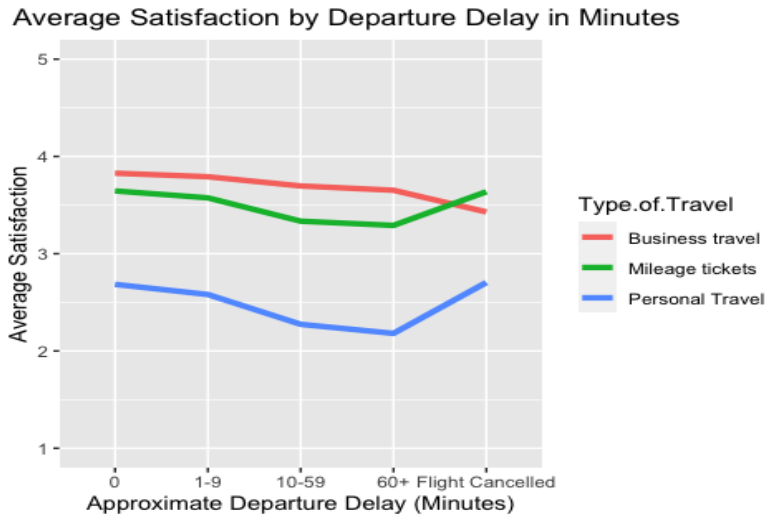
Satisfaction by Loyalty Membership:

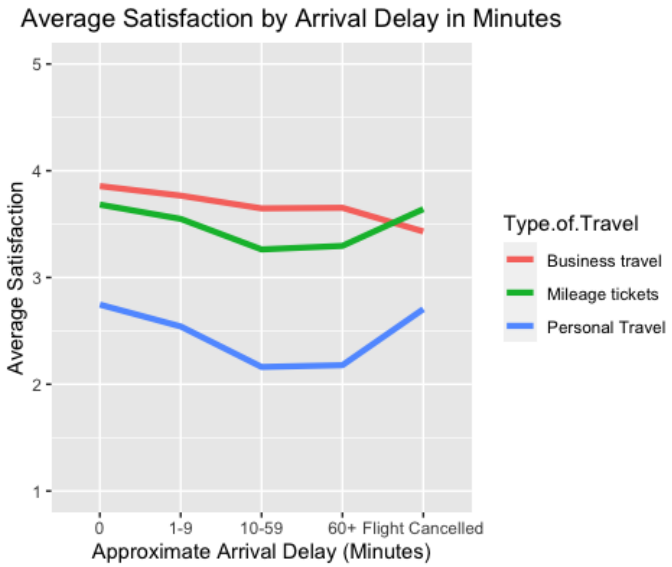
We see again business travels are most satisfied, followed closely by customers using mileage tickets. Personal travelers are considerably less satisfied; *airlines may want to consider focused efforts on improving the experience among this group.* It's also important to note customers with a loyalty membership tend to be slightly more satisfied across travel types. *Continued investments in these types of programs appear to be justified.*



Satisfaction by Departure/Arrival:

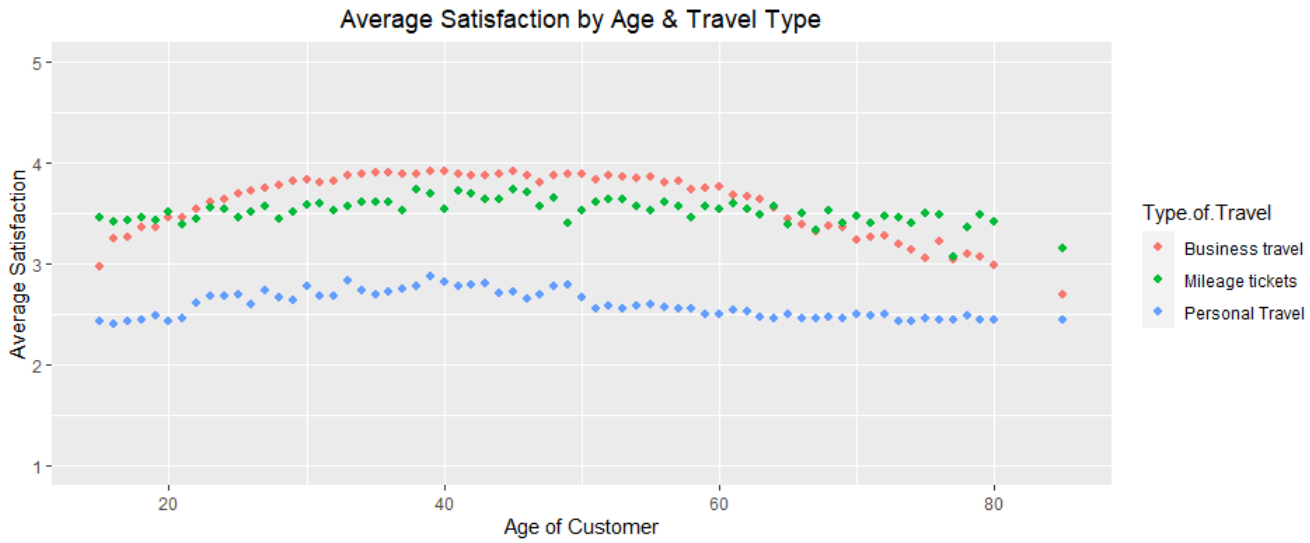
The following charts reinforce the idea that on average business travellers and mileage tickets have a higher satisfaction than personal travellers. As expected, satisfaction goes down as the delay time increases, with the exception of cancelled flights where we see a significant increase of satisfaction among mileage and personal travellers. Business travelers with a cancelled flight see a downturn in satisfaction, likely because they are missing an important commitment.





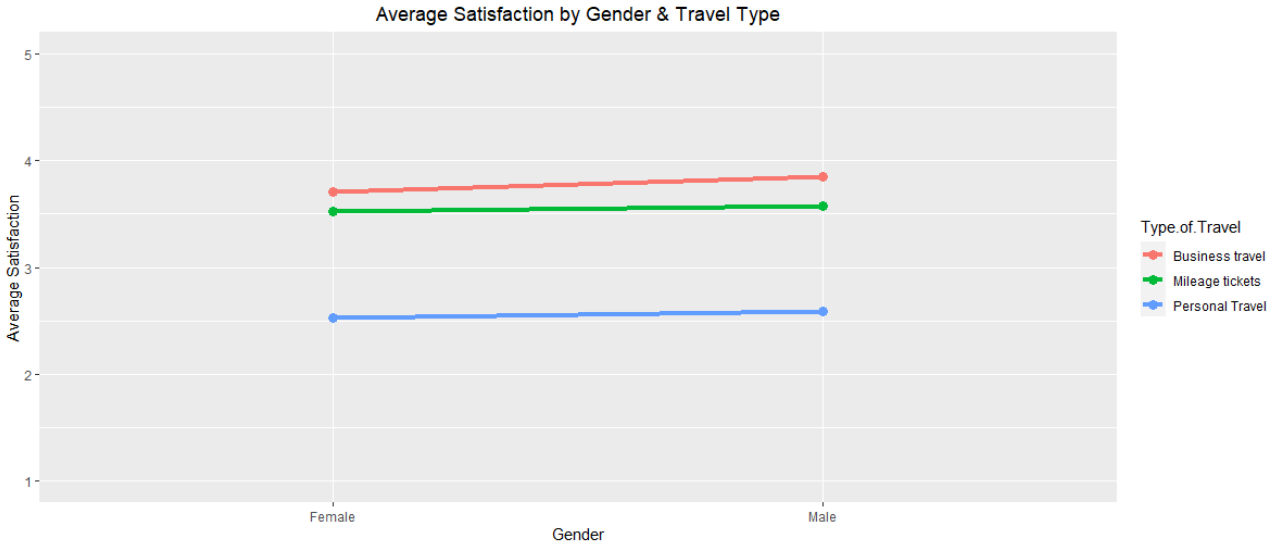
Satisfaction by Age:

Middle-aged customers are most satisfied. Business travelers become less satisfied as they grow older, maybe this is a result of the experience no longer being “new and different”.



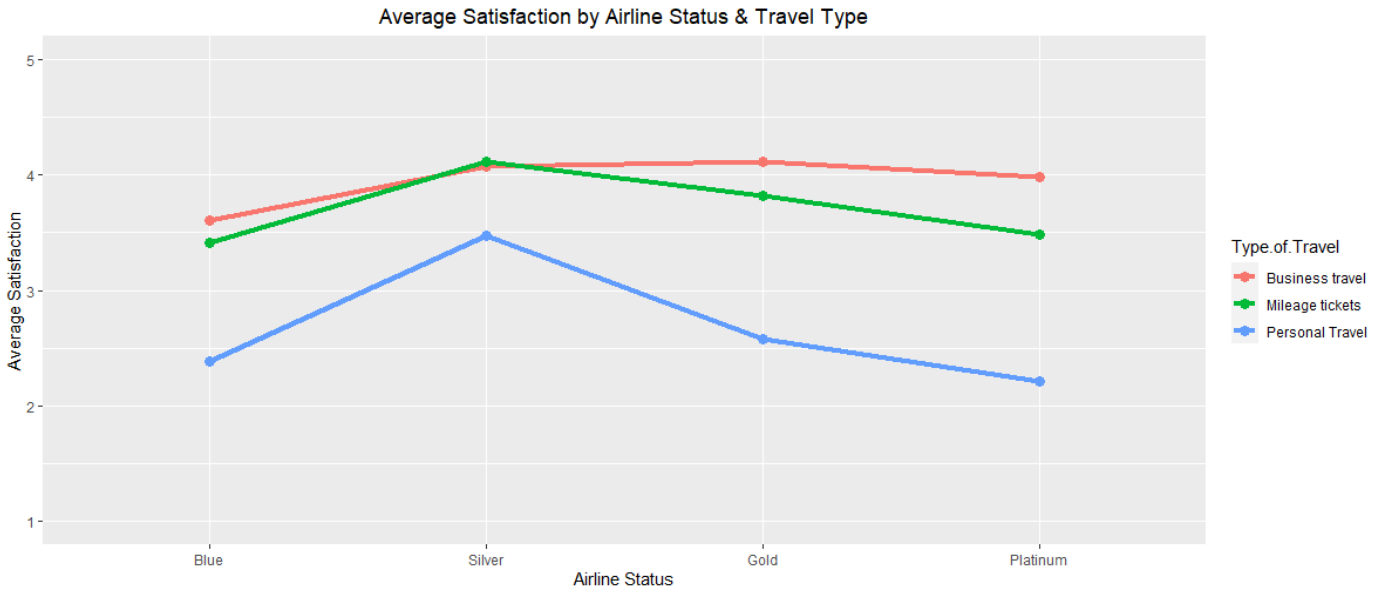
Satisfaction by Gender:

Males are directionally more satisfied across traveler types.



Satisfaction by Airline Status:

Airline performance on satisfaction appears to peak at “Silver” status. Personal Platinum status customers are the least satisfied. Are Gold and Platinum tiers offering perks that appeal to these types of customers? Are all 4 status types necessary? Personal Platinum status customers are the LEAST satisfied.



Model Development

Our initial linear model includes all independent variables our team compiled using business sense combined with the previous visualizations shown. This model returns an adjusted R2 of 42.66%. Please note, categorical variables were transformed into binary, leaving one of the options out for the model to handle. Also, we removed NAs from this data frame since we need data across these variables.

```
Coefficients:
                                   Estimate Std. Error  t value Pr(>|t|)
(Intercept)                       3.963e+00  1.896e-02  208.958  < 2e-16 ***
df.Age                           -2.292e-03  1.435e-04  -15.974  < 2e-16 ***
df.Price.Sensitivity             -4.010e-02  3.822e-03  -10.492  < 2e-16 ***
df.No.of.Flights.p.a.            -3.277e-03  1.575e-04  -20.802  < 2e-16 ***
df.No..of.other.Loyalty.Cards    -2.340e-03  2.036e-03   -1.149   0.2504
df.Shopping.Amount.at.Airport     1.651e-04  3.900e-05    4.234 2.30e-05 ***
df.Eating.and.Drinking.at.Airport -7.964e-05  4.032e-05   -1.975   0.0483 *
df.Departure.Delay.in.Minutes     1.767e-03  2.148e-04    8.224  < 2e-16 ***
df.Arrival.Delay.in.Minutes      -3.749e-03  2.126e-04  -17.634  < 2e-16 ***
df.Flight.time.in.minutes        -1.146e-03  1.403e-04   -8.168 3.16e-16 ***
df.Flight.Distance                1.402e-04  1.696e-05    8.267  < 2e-16 ***
df.Airline.Status.Blue1          -2.675e-01  1.184e-02  -22.587  < 2e-16 ***
df.Airline.Status.Gold1           1.736e-01  1.349e-02   12.868  < 2e-16 ***
df.Airline.Status.Silver1         3.547e-01  1.237e-02   28.669  < 2e-16 ***
df.Type.of.Travel.Business1       1.436e-01  7.921e-03   18.128  < 2e-16 ***
df.Type.of.Travel.Personal1      -9.322e-01  8.495e-03 -109.727  < 2e-16 ***
df.Class.Eco1                    -7.527e-02  7.518e-03  -10.012  < 2e-16 ***
df.Class.EcoPlus1                -6.970e-02  9.655e-03   -7.219 5.26e-13 ***
df.GenderBin1                     1.312e-01  4.279e-03   30.676  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7319 on 127124 degrees of freedom
Multiple R-squared:  0.4267,    Adjusted R-squared:  0.4266
F-statistic:  5256 on 18 and 127124 DF,  p-value: < 2.2e-16
```

We found that the number of loyalty cards is not significant. Departure and arrival delays likely have a lot of multicollinearity. We see similar trends when it comes to the magnitude of impact across variables when comparing to our visualizations. We iterated a few times to improve the model based on these notes.

LM Iteration 1: Remove Loyalty Cards as they are not significant.

```
> #remove loyalty cards not significant
> iter.1 <-dfLMModel[,-5]
> str(iter.1)
'data.frame':   127143 obs. of  18 variables:
 $ df.Satisfaction                  : num  4 4 5 5 4 4 4 4 2 5 ...
 $ df.Age                           : num  56 43 49 49 33 44 51 28 39 46 ...
 $ df.Price.Sensitivity             : num  2 1 1 1 1 1 1 1 1 1 ...
 $ df.No.of.Flights.p.a.            : num  41 9 14 0 4 8 12 37 17 29 ...
 $ df.Shopping.Amount.at.Airport    : num  15 10 8 0 0 0 25 130 0 0 ...
 $ df.Eating.and.Drinking.at.Airport: num  60 45 26 65 90 90 80 60 75 75 ...
 $ df.Departure.Delay.in.Minutes    : num  2 26 0 0 0 0 0 0 0 13 ...
 $ df.Arrival.Delay.in.Minutes      : num  5 39 0 0 1 0 0 3 0 0 ...
 $ df.Flight.time.in.minutes        : num  120 141 144 123 138 114 118 145 156 114 ...
 $ df.Flight.Distance               : num  821 821 853 821 821 853 821 853 853 853 ...
 $ df.Airline.Status.Blue           : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 2 2 1 ...
 $ df.Airline.Status.Gold           : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ df.Airline.Status.Silver         : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
 $ df.Type.of.Travel.Business       : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ df.Type.of.Travel.Personal       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.Class.Eco                     : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ df.Class.EcoPlus                 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.GenderBin                     : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 2 1 1 ...
> model.iter1<-lm(df.Satisfaction~.,data=iter.1)
> summary(model.iter1)

Call:
lm(formula = df.Satisfaction ~ ., data = iter.1)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0780 -0.4487  0.0249  0.4941  3.1723

Coefficients:
                                    Estimate Std. Error  t value
(Intercept)                        3.956e+00  1.814e-02  218.047
df.Age                            -2.224e-03  1.307e-04  -17.020
df.Price.Sensitivity              -3.985e-02  3.816e-03  -10.444
df.No.of.Flights.p.a.             -3.254e-03  1.562e-04  -20.828
df.Shopping.Amount.at.Airport      1.651e-04  3.900e-05    4.233
df.Eating.and.Drinking.at.Airport -7.900e-05  4.032e-05   -1.959
df.Departure.Delay.in.Minutes      1.766e-03  2.148e-04    8.222
df.Arrival.Delay.in.Minutes       -3.749e-03  2.126e-04  -17.632
df.Flight.time.in.minutes         -1.146e-03  1.403e-04   -8.168
df.Flight.Distance                 1.402e-04  1.696e-05    8.266
df.Airline.Status.Blue1           -2.673e-01  1.184e-02  -22.573
df.Airline.Status.Gold1            1.738e-01  1.349e-02   12.882
df.Airline.Status.Silver1          3.549e-01  1.237e-02   28.690
df.Type.of.Travel.Business1        1.437e-01  7.920e-03   18.147
df.Type.of.Travel.Personal1       -9.324e-01  8.492e-03 -109.795
df.Class.Eco1                     -7.529e-02  7.518e-03  -10.015
df.Class.EcoPlus1                 -6.926e-02  9.647e-03   -7.179
df.GenderBin1                      1.314e-01  4.276e-03   30.735
                                  Pr(>|t|)
(Intercept)                        < 2e-16 ***
df.Age                             < 2e-16 ***
df.Price.Sensitivity               < 2e-16 ***
df.No.of.Flights.p.a.              < 2e-16 ***
df.Shopping.Amount.at.Airport     2.31e-05 ***
df.Eating.and.Drinking.at.Airport   0.0501 .
df.Departure.Delay.in.Minutes      < 2e-16 ***
df.Arrival.Delay.in.Minutes        < 2e-16 ***
df.Flight.time.in.minutes         3.16e-16 ***
df.Flight.Distance                 < 2e-16 ***
df.Airline.Status.Blue1            < 2e-16 ***
df.Airline.Status.Gold1            < 2e-16 ***
df.Airline.Status.Silver1          < 2e-16 ***
df.Type.of.Travel.Business1        < 2e-16 ***
df.Type.of.Travel.Personal1        < 2e-16 ***
df.Class.Eco1                      < 2e-16 ***
df.Class.EcoPlus1                 7.05e-13 ***
df.GenderBin1                      < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7319 on 127125 degrees of freedom
Multiple R-squared:  0.4267,    Adjusted R-squared:  0.4266
F-statistic:  5566 on 17 and 127125 DF,  p-value: < 2.2e-16
```

LM Iteration 2: Remove Eating and Drinking at Airport as the p value is above 0.05 threshold.

```
> #remove Eating only significant at 90%
> iter.2 <-iter.1[,-6]
> str(iter.2)
'data.frame':   127143 obs. of  17 variables:
 $ df.Satisfaction              : num  4 4 5 5 4 4 4 4 2 5 ...
 $ df.Age                       : num  56 43 49 49 33 44 51 28 39 46 ...
 $ df.Price.Sensitivity         : num  2 1 1 1 1 1 1 1 1 1 ...
 $ df.No.of.Flights.p.a.        : num  41 9 14 0 4 8 12 37 17 29 ...
 $ df.Shopping.Amount.at.Airport: num  15 10 8 0 0 0 25 130 0 0 ...
 $ df.Departure.Delay.in.Minutes: num  2 26 0 0 0 0 0 0 0 13 ...
 $ df.Arrival.Delay.in.Minutes  : num  5 39 0 0 1 0 0 3 0 0 ...
 $ df.Flight.time.in.minutes    : num  120 141 144 123 138 114 118 145 156 114 ...
 $ df.Flight.Distance           : num  821 821 853 821 821 853 821 853 853 853 ...
 $ df.Airline.Status.Blue       : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 2 2 1 ...
 $ df.Airline.Status.Gold       : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ df.Airline.Status.Silver     : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
 $ df.Type.of.Travel.Business   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ df.Type.of.Travel.Personal   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.Class.Eco                 : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ df.Class.EcoPlus             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.GenderBin                 : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 2 1 1 ...
> model.iter2<-lm(df.Satisfaction~.,data=iter.2)
> summary(model.iter2)

Call:
lm(formula = df.Satisfaction ~ ., data = iter.2)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0763 -0.4486  0.0250  0.4938  3.1676

Coefficients:
                               Estimate Std. Error  t value
(Intercept)                   3.951e+00  1.792e-02  220.445
df.Age                       -2.255e-03  1.297e-04  -17.386
df.Price.Sensitivity         -3.945e-02  3.811e-03  -10.353
df.No.of.Flights.p.a.        -3.217e-03  1.551e-04  -20.742
df.Shopping.Amount.at.Airport 1.616e-04  3.896e-05    4.147
df.Departure.Delay.in.Minutes 1.766e-03  2.148e-04    8.222
df.Arrival.Delay.in.Minutes  -3.748e-03  2.126e-04  -17.629
df.Flight.time.in.minutes    -1.146e-03  1.403e-04   -8.168
df.Flight.Distance            1.402e-04  1.696e-05    8.267
df.Airline.Status.Blue1      -2.662e-01  1.183e-02  -22.508
df.Airline.Status.Gold1       1.739e-01  1.349e-02   12.894
df.Airline.Status.Silver1     3.553e-01  1.237e-02   28.718
df.Type.of.Travel.Business1   1.439e-01  7.919e-03   18.168
df.Type.of.Travel.Personal1  -9.329e-01  8.490e-03 -109.884
df.Class.Eco1                -7.531e-02  7.518e-03  -10.016
df.Class.EcoPlus1            -6.959e-02  9.646e-03   -7.214
df.GenderBin1                 1.308e-01  4.265e-03   30.673
                              Pr(>|t|)
(Intercept)                    < 2e-16 ***
df.Age                         < 2e-16 ***
df.Price.Sensitivity           < 2e-16 ***
df.No.of.Flights.p.a.          < 2e-16 ***
df.Shopping.Amount.at.Airport 3.37e-05 ***
df.Departure.Delay.in.Minutes  < 2e-16 ***
df.Arrival.Delay.in.Minutes    < 2e-16 ***
df.Flight.time.in.minutes     3.15e-16 ***
df.Flight.Distance             < 2e-16 ***
df.Airline.Status.Blue1        < 2e-16 ***
df.Airline.Status.Gold1        < 2e-16 ***
df.Airline.Status.Silver1      < 2e-16 ***
df.Type.of.Travel.Business1    < 2e-16 ***
df.Type.of.Travel.Personal1    < 2e-16 ***
df.Class.Eco1                  < 2e-16 ***
df.Class.EcoPlus1             5.45e-13 ***
df.GenderBin1                  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7319 on 127126 degrees of freedom
Multiple R-squared:  0.4267,    Adjusted R-squared:  0.4266
F-statistic:  5913 on 16 and 127126 DF,  p-value: < 2.2e-16
```

LM Iteration 3: Remove Departure delays due to collinearity with arrival delay.

```
> #remove departure delay due to collinearity with arrival delay
> iter.3 <-iter.2[,-6]
> str(iter.3)
'data.frame':   127143 obs. of  16 variables:
 $ df.Satisfaction              : num  4 4 5 5 4 4 4 4 2 5 ...
 $ df.Age                       : num  56 43 49 49 33 44 51 28 39 46 ...
 $ df.Price.Sensitivity         : num  2 1 1 1 1 1 1 1 1 1 ...
 $ df.No.of.Flights.p.a.        : num  41 9 14 0 4 8 12 37 17 29 ...
 $ df.Shopping.Amount.at.Airport: num  15 10 8 0 0 0 25 130 0 0 ...
 $ df.Arrival.Delay.in.Minutes  : num  5 39 0 0 1 0 0 3 0 0 ...
 $ df.Flight.time.in.minutes    : num  120 141 144 123 138 114 118 145 156 114 ...
 $ df.Flight.Distance           : num  821 821 853 821 821 853 821 853 853 853 ...
 $ df.Airline.Status.Blue       : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 2 2 1 ...
 $ df.Airline.Status.Gold       : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ df.Airline.Status.Silver     : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
 $ df.Type.of.Travel.Business   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ df.Type.of.Travel.Personal   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.Class.Eco                 : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ df.Class.EcoPlus             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.GenderBin                 : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 2 1 1 ...
> model.iter3<-lm(df.Satisfaction~.,data=iter.3)
> summary(model.iter3)

Call:
lm(formula = df.Satisfaction ~ ., data = iter.3)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0591 -0.4486  0.0248  0.4943  3.3016

Coefficients:
                               Estimate Std. Error  t value    Pr(>|t|)
(Intercept)                   3.957e+00  1.791e-02  220.948    < 2e-16 ***
df.Age                       -2.248e-03  1.297e-04  -17.333    < 2e-16 ***
df.Price.Sensitivity         -3.962e-02  3.812e-03  -10.394    < 2e-16 ***
df.No.of.Flights.p.a.        -3.222e-03  1.552e-04  -20.765    < 2e-16 ***
df.Shopping.Amount.at.Airport 1.616e-04  3.897e-05    4.147   3.37e-05 ***
df.Arrival.Delay.in.Minutes  -2.055e-03  5.315e-05  -38.672    < 2e-16 ***
df.Flight.time.in.minutes    -1.486e-03  1.342e-04  -11.073    < 2e-16 ***
df.Flight.Distance            1.809e-04  1.623e-05   11.144    < 2e-16 ***
df.Airline.Status.Blue1      -2.663e-01  1.183e-02  -22.504    < 2e-16 ***
df.Airline.Status.Gold1       1.740e-01  1.349e-02   12.894    < 2e-16 ***
df.Airline.Status.Silver1     3.553e-01  1.237e-02   28.710    < 2e-16 ***
df.Type.of.Travel.Business1   1.436e-01  7.921e-03   18.128    < 2e-16 ***
df.Type.of.Travel.Personal1  -9.333e-01  8.492e-03 -109.902    < 2e-16 ***
df.Class.Eco1                -7.552e-02  7.520e-03  -10.042    < 2e-16 ***
df.Class.EcoPlus1            -7.003e-02  9.648e-03   -7.258   3.94e-13 ***
df.GenderBin1                 1.308e-01  4.266e-03   30.656    < 2e-16 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7321 on 127127 degrees of freedom
Multiple R-squared:  0.4264,    Adjusted R-squared:  0.4263
F-statistic:  6299 on 15 and 127127 DF,  p-value: < 2.2e-16
```

All variables are significant within our third iteration, AR2=43.43%. This is our final set of independent variables for regression. And finally, SVM is the best fit within this (iteration 3) model's structure when compared to LM and KSVM.

>#KSVM

```
> #KSVM
> Ksvm.model<-ksvm(df.Satisfaction~.,data=df.train)
> summary(Ksvm.model)
Length  Class   Mode
     1   ksvm     S4
```

> rmse(df.test$error3)

[1] 0.7228202

> #SVM

```
> #Svm
> svm.model<-svm(df.Satisfaction~.,data=df.train)
> summary(svm.model)

Call:
svm(formula = df.Satisfaction ~ ., data = df.train)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.0625
    epsilon:  0.1


Number of Support Vectors:  64467
```

> rmse(df.test$error2)

[1] 0.7202886

> #LM

> rmse(df.test$error1)

[1] 0.7341511

Classification models

We coded the satisfaction variable to be a binary variable where 1 equals a satisfaction of 4 or 5 and any other satisfaction is considered 0 or not satisfied. Both models (KSVM/SVM) are similarly good at predicting satisfaction, with a 79% accuracy rate on the test data set.

```
> # recode variable for statisfaction
> iter.4<-iter.3
> iter.4$SatisfactionClass<- ifelse(iter.4$df.Satisfaction==5, "1", ifelse(iter.4$df.Satisfaction==4, "1",
    ifelse(iter.4$df.Satisfaction==3, "0", ifelse(iter.4$df.Satisfaction==2, "0",
    ifelse(iter.4$df.Satisfaction==1, "0", 'NA'))))
> iter.4<-iter.4[,-1]
> str(iter.4)
'data.frame':   127143 obs. of  16 variables:
 $ df.Age                       : num  56 43 49 49 33 44 51 28 39 46 ...
 $ df.Price.Sensitivity         : num  2 1 1 1 1 1 1 1 1 ...
 $ df.No.of.Flights.p.a.        : num  41 9 14 0 4 8 12 37 17 29 ...
 $ df.Shopping.Amount.at.Airport: num  15 10 8 0 0 0 25 130 0 0 ...
 $ df.Arrival.Delay.in.Minutes  : num  5 39 0 0 1 0 0 3 0 0 ...
 $ df.Flight.time.in.minutes    : num  120 141 144 123 138 114 118 145 156 114 ...
 $ df.Flight.Distance           : num  821 821 853 821 821 853 821 853 853 853 ...
 $ df.Airline.Status.Blue       : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 2 2 1 ...
 $ df.Airline.Status.Gold       : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ df.Airline.Status.Silver     : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
 $ df.Type.of.Travel.Business   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ df.Type.of.Travel.Personal   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.Class.Eco                 : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ df.Class.EcoPlus             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ df.GenderBin                 : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 2 1 1 ...
 $ SatisfactionClass            : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 1 2 ...
> #create train/test df for Iter.4
> nrow.df<-nrow(iter.4) #total observations
> cutPoint<-floor(nrow.df/3*2)#2/3 of the count
> rand<-sample(1:nrow.df)#randomize rows
> df.train2<-iter.4[rand[1:cutPoint],]#create train dataset
> dim(df.train2)
[1] 84762    16
> df.test2<-iter.4[rand[(cutPoint+1):nrow.df],]#create test dataset
> dim(df.test2)
[1] 42381    16
```

>\# KSVM

```
> #KSvm
> ksvm.model.class<-ksvm(SatisfactionClass~.,data=df.train2)
> pred.svm.class<-predict(svm.model.class,df.test2)
> print(summary(ksvm.model.class))
Length  Class   Mode
     1   ksvm     S4
> #review predictions
> df.test2$predictSatKsvm<-predict(ksvm.model.class,df.test2)
> results<-table(df.test2$SatisfactionClass,df.test2$predictSatKsvm)
> print(results)

     0     1
0 12654  7909
1  1093 20725
> percentCorrect<-(results[1,1]+results[2,2])/(results[1,1]+results[1,2]+results[2,1]+results[2,2])*100
> cat("Percent Correct: ",round(percentCorrect),"% \n")
Percent Correct:  79 %
```

Percent Correct: 79 %

>#SVM

```
> #Svm
> svm.model.class<-svm(SatisfactionClass~.,data=df.train2)
> summary(svm.model.class)
Call:
svm(formula = SatisfactionClass ~ ., data = df.train2)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  40260

 ( 20023 20237 )


Number of Classes:  2

Levels:
 0 1

> #review predictions
> df.test2$predictSatSvm<-predict(svm.model.class,df.test2)
> print(results)

      0     1
  0 12478  8085
  1  1005 20813
> percentCorrect<-(results[1,1]+results[2,2])/(results[1,1]+results[1,2]+results[2,1]+results[2,2])*100
> cat("Percent Correct: ",round(percentCorrect),"% \n")
Percent Correct:  79 %
```

Percent Correct: 79 %

Final model analysis

While our best predictive model is our SVM, our final model is the Linear Model with the variables displayed in the below table. This model is clearest for us to evaluate as the coefficients are easy to interpret. We found whether a traveler is a personal traveler has the largest impact on the passenger's satisfaction and that impact is negative. Airline Status is the next highest, with gender, class, and prices sensitivity also being notable variables. Surprisingly, delays seem to be less impactful than we initially thought, with the caveat that as arrival delays get longer (40 minutes +), they become a more important negative variable.

Key Driver	Raw Coefficient	Absolute Value	Relative Proportion
(Intercept)	3.9572	-	-
df.Type.of.Travel.Personal1	-0.9333	0.9333	42.5%
df.Airline.Status.Silver1	0.3553	0.3553	**16.2%**
df.Airline.Status.Blue1	-0.2663	0.2663	**12.1%**
df.Airline.Status.Gold1	0.1740	0.1740	**7.9%**
df.Type.of.Travel.Business1	0.1436	0.1436	**6.5%**
df.GenderBin1	0.1308	0.1308	**6.0%**
df.Class.Eco1	-0.0755	0.0755	**3.4%**
df.Class.EcoPlus1	-0.0700	0.0700	**3.2%**
df.Price.Sensitivity	-0.0396	0.0396	**1.8%**
df.No.of.Flights.p.a.	-0.0032	0.0032	0.1%
df.Age	-0.0022	0.0022	0.1%
df.Arrival.Delay.in.Minutes	-0.0021	0.0021	0.1%
df.Flight.time.in.minutes	-0.0015	0.0015	0.1%
df.Flight.Distance	0.0002	0.0002	0.0%
df.Shopping.Amount.at.Airport	0.0002	0.0002	0.0%

Interpretation of Results

Business Question	What is the **profile of an airline customer?**	How do each of the **airlines compare to each other on satisfaction?**	**Are personal travelers more or less satisfied** when compared to business travelers?	Which attributes are **key drivers of airline customer satisfaction?** How can we use this information to improve satisfaction ratings over time?
Insights & Recommendations	Airline customers have the following characteristics... • Female (56%) • Ages 35-45 are most common • Blue status (68%); silver status (20%) • Tend to have taken ~15-20 flights on average • Business travelers make up the majority (61%) • 8 in 10 opt for standard economy • Dining in the airport is very common (95%), shopping is more split **Airlines should keep customer proportions in mind when target marketing, and when aiming to improve satisfactions scores.**	Customer satisfaction is roughly similar *(in proportion)* across airlines, each showing room for improvement. Cheapseats has the largest presence, while Cool&Young & GoingNorth have fewer customers overall. **Satisfaction is driven by factors outside of which airlines these customers use. All airlines show room for improvement, with ~50% of customers satisfied across the industry.**	Personal travelers are significantly LESS satisfied than business travelers and those using their mileage tickets to purchase. Even though personal travelers are a smaller customer base overall (31%), **airlines should consider focused efforts on improving the experience among this group.** Business travelers become less satisfied over time *(i.e., as they grow older)*. Airlines should make efforts to keep their travel experience new and different.	**FIRST STRIKE** Whether or not a customer is a personal traveler *(vs. business)*, is the most impactful driver of their satisfaction. **Airlines should conduct a deep dive analysis of this group, perhaps speaking to customers directly using a marketing research initiative.** **NEXT UP** **Examine airline status.** What is it about silver status that improves customer satisfaction over blue to a significant degree? Why are gold and platinum members not more satisfied than silver? **EFFORTS TO DELIGHT** Delays have a lower relative impact on satisfaction, but we do see a trend of lower scores among those experiencing delays. **Airlines should reduce delays to the extent possible, with particular focus on the Northeast** *(where most delays are happening today).*

Code:

```
#-------Read in the RAW data-------
library("kernlab")
library(e1071)
library(ggplot2)
library(gridExtra)
library(caret)
library(readr)

#-------------------------------------
#Data Frame Prep
#-------------------------------------

#Select file name
filename<-file.choose()
#Function to read file
readCSV <- function() #read in data, replace blanks with NAs, trim white space
{
  x <- read_csv(file=filename, na = "NA", trim_ws = TRUE) #note, this function replaces 3 gibberish
satisfaction ratings with "NA"
  return(x)
}

#Create Raw Dataframe
SatSurveyRaw <- readCSV()

#-------Clean the data within a function-------

CleanData <- function() # function to clean satisfaction survey data set
{
  d <- SatSurveyRaw
```

```
  # transform column names to new names without spaces
  names(d)<-make.names(names(d),unique = TRUE)
  names(d)[8]<-paste("Percent.of.Flight.with.Other.Airlines")

  # Remove cases with misrepresented satisfaction ratings (i.e., NOT 1:5 integers)
  d <- d[!is.na(d$Satisfaction), ]
  d <- d[d$Satisfaction==1 | d$Satisfaction==2 | d$Satisfaction==3 | d$Satisfaction==4 |
d$Satisfaction==5, ]

  #Code down flight with other airlines values over 100%, down to 100%
  d$Percent.of.Flight.with.Other.Airlines[d$Percent.of.Flight.with.Other.Airlines>100] <- 100

  #remove records with no flight time AND does not say their flight was canceled
  #we feel these records aren't intuitive and therefore not useful, perhaps captured incorrectly
  d <- d[!(is.na(d$Arrival.Delay.in.Minutes) & is.na(d$Flight.time.in.minutes) &
d$Flight.cancelled=="No"),]

  #add a unique identifier
  d$Unique <- c(1:129543)

  #convert flight.date to date format
  d$Flight.date <- as.Date(d$Flight.date , format = "%m/%d/%Y")

  return(d)
}

df <- CleanData()

#-------Let's view the clean data file specs-------

#display the specs of the columns
str(df)
```

```
#------------Create new variables----------
#variable creation
length(df$No..of.other.Loyalty.Cards[df$No..of.other.Loyalty.Cards=='NA']) # check for NAs
df$LoyaltyCardCat <- ifelse(df$No..of.other.Loyalty.Cards>0, "Member", "Non-Member")
length(df$Shopping.Amount.at.Airport[df$Shopping.Amount.at.Airport=='NA']) # check for NAs
df$ShoppingCat <- ifelse(df$Shopping.Amount.at.Airport>0, "Shopper", "Non-Shopper")
length(df$Eating.and.Drinking.at.Airport[df$Eating.and.Drinking.at.Airport=='NA']) # check for NAs
df$DinerCat <- ifelse(df$Eating.and.Drinking.at.Airport>0, "Diner", "Non-Diner")
#Create satisfaction variable coded (4-5, 3, 1-2)
df$Satisfaction.Coded <- ifelse(df$Satisfaction==5, "4-5", ifelse(df$Satisfaction==4, "4-5",
ifelse(df$Satisfaction==3, "3", ifelse(df$Satisfaction==2, "1-2",ifelse(df$Satisfaction==1, "1-2", 'NA')))))


#-------convert to binary----------

#No. of other Loyalty Cards:  add a column with (0=none) and (1=loyalty member)

df$LoyaltyBin<- ifelse(df$No..of.other.Loyalty.Cards>0, 1, 0)
df$LoyaltyBin<-as.factor(df$LoyaltyBin)
#Shopping Amount at Airport:  add a column with (0=non-shopper) and (1=shopper)
df$ShopperBin<- ifelse(df$Shopping.Amount.at.Airport>0, 1, 0)
df$ShopperBin<-as.factor(df$ShopperBin)
#Eating and Drinking at Airport:  add a column with (0=non-diner) and (1=diner)
df$DinerBin<- ifelse(df$Eating.and.Drinking.at.Airport>0, 1, 0)
df$DinerBin<-as.factor(df$DinerBin)
#gender bin male=1
df$GenderBin<- ifelse(df$Gender=="Male", 1, 0)
df$GenderBin<-as.factor(df$GenderBin)
#canceled bin
df$Flight.canceledBin<- ifelse(df$Flight.cancelled=="yes", 1, 0)
df$Flight.canceledBin<-as.factor(df$Flight.canceledBin)
```

```
#convert Travel type
df$Type.of.Travel.Business<- ifelse(df$Type.of.Travel=="Business travel", 1, 0)
df$Type.of.Travel.Business<-as.factor(df$Type.of.Travel.Business)
df$Type.of.Travel.Personal<- ifelse(df$Type.of.Travel=="Personal Travel", 1, 0)
df$Type.of.Travel.Personal<-as.factor(df$Type.of.Travel.Personal)
#Convert Status
df$Airline.Status.Blue<- ifelse(df$Airline.Status=="Blue", 1, 0)
df$Airline.Status.Blue<-as.factor(df$Airline.Status.Blue)
df$Airline.Status.Silver<- ifelse(df$Airline.Status=="Silver", 1, 0)
df$Airline.Status.Silver<-as.factor(df$Airline.Status.Silver)
df$Airline.Status.Gold<- ifelse(df$Airline.Status=="Gold", 1, 0)
df$Airline.Status.Gold<-as.factor(df$Airline.Status.Gold)

#Convert Class
df$Class.Eco<- ifelse(df$Class=="Eco", 1, 0)
df$Class.Eco<-as.factor(df$Class.Eco)
df$Class.EcoPlus<- ifelse(df$Class=="Eco Plus", 1, 0)
df$Class.EcoPlus<-as.factor(df$Class.EcoPlus)

#------------Create Linear Model Dataframe----------
dfLMModel<-data.frame(df$Satisfaction,df$Age,df$Price.Sensitivity,df$No.of.Flights.p.a.,

df$No..of.other.Loyalty.Cards,df$Shopping.Amount.at.Airport,df$Eating.and.Drinking.at.Airport,
            df$Departure.Delay.in.Minutes,df$Arrival.Delay.in.Minutes,
            df$Flight.time.in.minutes,df$Flight.Distance,df$Airline.Status.Blue,df$Airline.Status.Gold,
            df$Airline.Status.Silver,df$Type.of.Travel.Business,df$Type.of.Travel.Personal,
            df$Class.Eco,df$Class.EcoPlus,df$GenderBin)


dfLMModel <- na.omit(dfLMModel) #Remove NAs from linear model df, we want data across these
variables
str(dfLMModel)
```

```
#-------------------------------------------------
#Modeling
#-------------------------------------------------


model.all<-lm(df.Satisfaction~.,data=dfLMModel)


summary(model.all)

#remove loyalty cards not significant
iter.1 <-dfLMModel[,-5]
str(iter.1)
model.iter1<-lm(df.Satisfaction~.,data=iter.1)
summary(model.iter1)

#remove Eating only significant at 90%
iter.2 <-iter.1[,-6]
str(iter.2)
model.iter2<-lm(df.Satisfaction~.,data=iter.2)
summary(model.iter2)


#remove departure delay due to collinearity with arrival delay
iter.3 <-iter.2[,-6]
str(iter.3)
model.iter3<-lm(df.Satisfaction~.,data=iter.3)
summary(model.iter3)
#export to file for easier copy/paste
sink("iter3.txt")
```

```
print(summary(model.iter3))
sink()  # returns output to the console

iter.3_coef <- summary(model.iter3)$coefficients
iter.3_coef



#create train/test df for Iter.3
nrow.df<-nrow(iter.3) #total observations
cutPoint<-floor(nrow.df/3*2)#2/3 of the count
rand<-sample(1:nrow.df)#randomize rows
df.train<-iter.3[rand[1:cutPoint],]#create train dataset
dim(df.train)
df.test<-iter.3[rand[(cutPoint+1):nrow.df],]#create test dataset
dim(df.test)



####root mean squared error
rmse <- function(error)
{
  sqrt(mean(error^2))
}
# LM Model
lm.model<-lm(df.Satisfaction~.,data=df.train)
summary(lm.model)
#export to file for easier copy/paste
sink("lm.txt")
print(summary(lm.model))
sink()  # returns output to the console
pred.lm<-predict(lm.model,df.test)
df.test$error1 <- df.test$df.Satisfaction - pred.lm
head(df.test)
```

```
rmse(df.test$error1)

#Svm
svm.model<-svm(df.Satisfaction~.,data=df.train)
summary(svm.model)
#export to file for easier copy/paste
sink("svm.txt")
print(summary(svm.model))
sink()  # returns output to the console
pred.svm<-predict(svm.model,df.test)
df.test$error2 <- df.test$df.Satisfaction - pred.svm
head(df.test)
rmse(df.test$error2)

#KSvm
Ksvm.model<-ksvm(df.Satisfaction~.,data=df.train)
summary(Ksvm.model)
#export to file for easier copy/paste
sink("ksvm.txt")
print(summary(Ksvm.model))
sink()  # returns output to the console
pred.ksvm<-predict(Ksvm.model,df.test)
df.test$error3 <- df.test$df.Satisfaction - pred.ksvm
head(df.test)
rmse(df.test$error3)


# recode variable for statisfaction
iter.4<-iter.3
iter.4$SatisfactionClass<- ifelse(iter.4$df.Satisfaction==5, "1", ifelse(iter.4$df.Satisfaction==4, "1",
ifelse(iter.4$df.Satisfaction==3, "0", ifelse(iter.4$df.Satisfaction==2, "0",ifelse(iter.4$df.Satisfaction==1,
"0", 'NA')))))
```

```
str(iter.4)
iter.4<-iter.4[,-1]
iter.4$SatisfactionClass<-as.factor(iter.4$SatisfactionClass)

#create train/test df for Iter.4
nrow.df<-nrow(iter.4) #total observations
cutPoint<-floor(nrow.df/3*2)#2/3 of the count
rand<-sample(1:nrow.df)#randomize rows
df.train2<-iter.4[rand[1:cutPoint],]#create train dataset
dim(df.train2)
df.test2<-iter.4[rand[(cutPoint+1):nrow.df],]#create test dataset
dim(df.test2)

#Svm
svm.model.class<-svm(SatisfactionClass~.,data=df.train2)
#export to file for easier copy/paste
sink("svmclass.txt")
print(summary(svm.model.class))
sink()  # returns output to the console
#review predictions
df.test2$predictSatSvm<-predict(svm.model.class,df.test2)
str(df.test2)
results<-table(df.test2$SatisfactionClass,df.test2$predictSatSvm)
print(results)
percentCorrect<-(results[1,1]+results[2,2])/(results[1,1]+results[1,2]+results[2,1]+results[2,2])*100
cat("Percent Correct: ",round(percentCorrect),"\n")


#KSvm
ksvm.model.class<-ksvm(SatisfactionClass~.,data=df.train2)
#export to file for easier copy/paste
sink("ksvmclass.txt")
```

```
print(summary(ksvm.model.class))
sink()  # returns output to the console
#review predictions
df.test2$predictSatKsvm<-predict(ksvm.model.class,df.test2)

results<-table(df.test2$SatisfactionClass,df.test2$predictSatKsvm)
print(results)
percentCorrect<-(results[1,1]+results[2,2])/(results[1,1]+results[1,2]+results[2,1]+results[2,2])*100
cat("Percent Correct: ",round(percentCorrect),"% \n")



#-------Creating variables for analysis-------

#No. of other Loyalty Cards:  add a column with (0=none) and (1=loyalty member)
length(df$No..of.other.Loyalty.Cards[df$No..of.other.Loyalty.Cards=='NA']) # check for NAs
df$LoyaltyCardCat <- ifelse(df$No..of.other.Loyalty.Cards>0, "Member", "Non-Member")
g <- ggplot(df, aes(x=LoyaltyCardCat))
g <- g + geom_bar(color='black', fill='gray')
g # visualize result

#Shopping Amount at Airport:  add a column with (0=non-shopper) and (1=shopper)
length(df$Shopping.Amount.at.Airport[df$Shopping.Amount.at.Airport=='NA']) # check for NAs
df$ShoppingCat <- ifelse(df$Shopping.Amount.at.Airport>0, "Shopper", "Non-Shopper")
g <- ggplot(df, aes(x=ShoppingCat))
g <- g + geom_bar(color='black', fill='gray')
g # visualize result

#Eating and Drinking at Airport:  add a column with (0=non-diner) and (1=diner)
length(df$Eating.and.Drinking.at.Airport[df$Eating.and.Drinking.at.Airport=='NA']) # check for NAs
df$DinerCat <- ifelse(df$Eating.and.Drinking.at.Airport>0, "Diner", "Non-Diner")
g <- ggplot(df, aes(x=DinerCat))
```

```
g <- g + geom_bar(color='black', fill='gray')
g # visualize result

table(df$DinerCat) # we see there aren't many non-diners for analysis
table(df$DinerCat) / length(df$DinerCat) # in fact, less than 5% are non-diners

#-------Additional Feature Profile------

# LOYALTY CARD
TempDf <- data.frame(table(df$LoyaltyCardCat) / length(df$LoyaltyCardCat))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", width=1, color="white")
g <- g + coord_polar("y", start=0) + geom_text(aes(label=paste0(round(Freq*100), "%")),
position=position_stack(vjust=0.5), color="white", size=5)
g <- g + scale_fill_manual(values=c("dodgerblue4", "gray26"))
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Loyalty Membership")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank(),
                        plot.title = element_text(hjust = 0.52, vjust=-5, color = "black", size=20),
                        legend.text = element_text(size=15))
g

# SHOPPER
TempDf <- data.frame(table(df$ShoppingCat) / length(df$ShoppingCat))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", width=1, color="white")
g <- g + coord_polar("y", start=0) + geom_text(aes(label=paste0(round(Freq*100), "%")),
position=position_stack(vjust=0.5), color="white", size=5)
g <- g + scale_fill_manual(values=c("dodgerblue4", "gray26"))
```

```
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Airport Shopping")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank(),
                        plot.title = element_text(hjust = 0.52, vjust=-5, color = "black", size=20),
                        legend.text = element_text(size=15))
g

# DINER
TempDf <- data.frame(table(df$DinerCat) / length(df$DinerCat))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", width=1, color="white")
g <- g + coord_polar("y", start=0) + geom_text(aes(label=paste0(round(Freq*100), "%")),
position=position_stack(vjust=0.5), color="white", size=5)
g <- g + scale_fill_manual(values=c("dodgerblue4", "gray26"))
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Airport Dining")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank(),
                        plot.title = element_text(hjust = 0.52, vjust=-5, color = "black", size=20),
                        legend.text = element_text(size=15))
g

#-------Demographic Profile-------
#GENDER
TempDf <- data.frame(table(df$Gender) / length(df$Gender))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", color="white", width=0.35)
g <- g + geom_text(aes(label=paste0(round(Freq*100), "%")), position=position_stack(vjust=0.5),
color="white", size=5)
```

```
g <- g + scale_fill_manual(values=c("dodgerblue4", "gray26"))
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Gender")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank(),
                        plot.title = element_text(hjust = 0.5, vjust=-1, color = "black", size=20),
                        legend.text = element_text(size=15),
                        legend.position = "bottom")
g

#AGE
g <- ggplot(df, aes(x=Age))
g <- g + geom_histogram(color="white", fill="dodgerblue4", binwidth=5)
g <- g + labs(title = "Age Distribution") + theme(plot.title = element_text(hjust=0.5, vjust=2, size=20))
g

#-------Usage Profile-------
#AIRLINE STATUS
TempDf <- data.frame(table(df$Airline.Status) / length(df$Airline.Status))
TempDf$Var1 <- factor(TempDf$Var1, levels=c("Blue", "Silver", "Gold", "Platinum"))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", color="white", width=0.35)
g <- g + geom_text(aes(label=paste0(round(Freq*100), "%")), position=position_stack(vjust=0.5),
color="white", size=5)
g <- g + scale_fill_manual(values=c("blue4", "grey75","darkgoldenrod2", "gray35"))
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Airline Status")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank(),
                        plot.title = element_text(hjust = 0.5, vjust=-1, color = "black", size=20),
                        legend.text = element_text(size=15),
```

```
                    legend.position = "right")
g

#NUMBER OF FLIGHTS
g <- ggplot(df, aes(x=No.of.Flights.p.a.))
g <- g + geom_histogram(color="white", fill="dodgerblue4", binwidth=5)
g <- g + labs(title = "Number of Flights") + theme(plot.title = element_text(hjust=0.5, vjust=2, size=20))
g

#TYPE OF TRAVELER
TempDf <- data.frame(table(df$Type.of.Travel) / length(df$Type.of.Travel))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", width=1, color="white")
g <- g + coord_polar("y", start=0) + geom_text(aes(label=paste0(round(Freq*100), "%")),
position=position_stack(vjust=0.5), color="white", size=5)
g <- g + scale_fill_manual(values=c("indianred3", "chartreuse3", "royalblue2"))
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Type of Travel")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                    axis.text = element_blank(),
                    axis.ticks = element_blank(),
                    plot.title = element_text(hjust = 0.52, vjust=-5, color = "black", size=20),
                    legend.text = element_text(size=15))
g

#CLASS
TempDf <- data.frame(table(df$Class) / length(df$Class))
TempDf$Var1 <- factor(TempDf$Var1, levels=c("Eco", "Eco Plus", "Business"))
TempDf
g <- ggplot(TempDf, aes(x="", y=Freq, fill=Var1))
g <- g + geom_bar(stat="identity", color="white", width=0.35)
```

```
g <- g + geom_text(aes(label=paste0(round(Freq*100), "%")), position=position_stack(vjust=0.5),
color="white", size=5)
g <- g + scale_fill_manual(values=c("dodgerblue1", "dodgerblue4","black"))
g <- g + labs(x = NULL, y = NULL, fill = NULL, title = "Class")
g <- g + theme_classic() + theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank(),
                        plot.title = element_text(hjust = 0.5, vjust=-1, color = "black", size=20),
                        legend.text = element_text(size=15),
                        legend.position = "left")
g


#Create satisfaction variable coded (4-5, 3, 1-2)
df$Satisfaction.Coded <- ifelse(df$Satisfaction==5, "4-5", ifelse(df$Satisfaction==4, "4-5",
ifelse(df$Satisfaction==3, "3", ifelse(df$Satisfaction==2, "1-2",ifelse(df$Satisfaction==1, "1-2", 'NA')))))

#Satisfaction descriptives
summary(df$Satisfaction)

#Let's check sample sizes by airline
stack(tapply(df$Unique, df$Airline.Name, length))

#View mean satisfaction by airline

round.mean <- function(x)
{
  y <- round(mean(x),digits=2)
  return(y)
}

stack(tapply(df$Satisfaction, df$Airline.Name, round.mean))
```

```
#-------Satisfaction by Airline Visualized as Counts-------
g <- ggplot(df, aes(x=Airline.Name))
g <- g + geom_histogram(stat="count", color="black", aes(fill=Satisfaction.Coded))
g <- g + theme(axis.text.x = element_text(angle = 90))
g <- g + ggtitle("Satisfaction by Airline") + theme(plot.title=element_text(hjust=0.5))
g <- g + xlab("Airline Name") + ylab("Count of Records") + labs(fill="Satisfaction Grouped")
g <- g + scale_fill_manual(values = c("darkred", "gray", "dodgerblue4"))
g

#-------Satisfaction by Airline Visualized as Percentages-------
g <- ggplot(df, aes(fill=factor(Satisfaction), y=Unique, x=Airline.Name))
g <- g + geom_bar(position="fill", stat="identity")
g <- g + theme(axis.text.x = element_text(angle = 90))
g <- g + ggtitle("Satisfaction by Airline") + theme(plot.title=element_text(hjust=0.5))
g <- g + xlab("Airline Name") + ylab("") + labs(fill="Satisfaction")
g <- g + scale_fill_manual(values = c("darkred", "indianred2", "gray", "dodgerblue1", "dodgerblue4"))
g

#-------Satisfaction by Airline Visualized as Y=Average-------
TempDf <- data.frame(tapply(df$Satisfaction, df$Airline.Name, mean))
TempDf$Airline.Name <- row.names(TempDf)
names(TempDf)[1] <- "Value"
g <- ggplot(TempDf, aes(x=reorder(Airline.Name, -Value), y=Value))
g <- g + geom_bar(stat="identity", color="black",fill="dodgerblue4")
g <- g + theme(axis.text.x = element_text(angle = 90, size=12))
g <- g + ggtitle("Average Satisfaction by Airline Sorted") + theme(plot.title=element_text(hjust=0.5))
g <- g + xlab("Airline Name") + ylab("Average Satisfaction")
g <- g + coord_cartesian(ylim = c (1, 5))
g <- g + geom_text(aes(label=round(Value,2)), position=position_dodge(width=0.9), vjust=-0.25)
g
```

```
#-------Airline Average Arrival Delay Comparison-------
g <- ggplot(df, aes(x=Airline.Name, y=Arrival.Delay.in.Minutes))
g <- g + geom_bar(stat="summary", fun="mean", color="black", fill="dodgerblue4")
g <- g + theme(axis.text.x = element_text(angle = 90))
g <- g + xlab("Airline Name") + ylab("Arrival Delay (minutes)")
g

#-------Satisfaction by Loyalty Membership (Split Business vs. Personal) with y=Average-------
g <- ggplot(df, aes(x=LoyaltyCardCat, y=Satisfaction, group=Type.of.Travel, color=Type.of.Travel))
g <- g + geom_line(stat="summary", fun="mean", size=1.5)
g <- g + ggtitle("Average Satisfaction by Loyalty Membership & Travel Type") +
theme(plot.title=element_text(hjust=0.5))
g <- g + xlab("Loyalty Membership") + ylab("Average Satisfaction")
g <- g + coord_cartesian(ylim = c (1, 5))
g <- g + geom_point(stat="summary", fun="mean", size=3)
g

#-------Satisfaction by Age & Traveler Type-------
g <- ggplot(df, aes(x=Age, y=Satisfaction, group=Type.of.Travel, color=Type.of.Travel))
g <- g + geom_point(stat="summary", fun="mean", size=1.75)
g <- g + ggtitle("Average Satisfaction by Age & Travel Type") + theme(plot.title=element_text(hjust=0.5))
g <- g + xlab("Age of Customer") + ylab("Average Satisfaction")
g <- g + coord_cartesian(ylim = c (1, 5))
g

#-------Satisfaction by Airline Status & Traveler Type-------
g <- ggplot(df, aes(x=factor(Airline.Status, level = c("Blue", "Silver", "Gold", "Platinum")), y=Satisfaction,
group=Type.of.Travel, color=Type.of.Travel))
g <- g + geom_line(stat="summary", fun="mean", size=1.5)
g <- g + geom_point(stat="summary", fun="mean", size=3)
g <- g + ggtitle("Average Satisfaction by Airline Status & Travel Type") +
theme(plot.title=element_text(hjust=0.5))
```

```
g <- g + xlab("Airline Status") + ylab("Average Satisfaction")
g <- g + coord_cartesian(ylim = c (1, 5))
g

#-------Satisfaction by Gender & Traveler Type-------
g <- ggplot(df, aes(x=Gender, y=Satisfaction, group=Type.of.Travel, color=Type.of.Travel))
g <- g + geom_line(stat="summary", fun="mean", size=1.5)
g <- g + geom_point(stat="summary", fun="mean", size=3)
g <- g + ggtitle("Average Satisfaction by Gender & Travel Type") +
theme(plot.title=element_text(hjust=0.5))
g <- g + xlab("Gender") + ylab("Average Satisfaction")
g <- g + coord_cartesian(ylim = c (1, 5))
g

# Show the U.S. map,
# Thus code was modified from the Sample Project Report
OriginState <- state.name
area <- state.area
center <- state.center
state_1 <- data.frame(OriginState, area, center)


UniqueOS <- sort(table(df$Origin.State))
UniqueOS <- data.frame(UniqueOS)
colnames(UniqueOS) <- c("OriginState","count")

Origin_State <- merge(UniqueOS, state_1, by = "OriginState")
Origin_State$OriginState <- tolower(Origin_State$OriginState)

library(ggplot2)
library(ggmap)
us <- map_data("state")
```

```
Origin_State_Area <- ggplot(Origin_State, aes(map_id = OriginState))
Origin_State_Area <- Origin_State_Area +
  geom_map(map = us, aes(fill = Origin_State$area))
Origin_State_Area <- Origin_State_Area +
  expand_limits(x = Origin_State$x, y = Origin_State$y)
Origin_State_Area <- Origin_State_Area + coord_map() + ggtitle("Area")

OS <- ggplot(Origin_State, aes(map_id = OriginState))
OS <- OS + geom_map(map = us, aes(fill = Origin_State$count))
OS <- OS + expand_limits(x = Origin_State$x, y = Origin_State$y)
OS <- OS + coord_map() + ggtitle("Origin State")
OS

DestinationState <- state.name
area <- state.area
center <- state.center
state_2 <- data.frame(DestinationState, area, center)

UniqueDS <- sort(table(df$Destination.State))
UniqueDS <- data.frame(UniqueDS)
colnames(UniqueDS) <- c("DestinationState","count")


Destination_State <- merge(UniqueDS, state_2, by = "DestinationState")
Destination_State$DestinationState <-
  tolower(Destination_State$DestinationState)

Destination_State_Area <- ggplot(Destination_State, aes(map_id = DestinationState))
Destination_State_Area <- Destination_State_Area +
  geom_map(map = us, aes(fill = Destination_State$area))
Destination_State_Area <- Destination_State_Area +
```

```
  expand_limits(x = Destination_State$x, y = Destination_State$y)
Destination_State_Area <- Destination_State_Area + coord_map() + ggtitle("Area")

DS <- ggplot(Destination_State, aes(map_id = DestinationState))
DS <- DS + geom_map(map = us, aes(fill = Destination_State$count))
DS <- DS + expand_limits(x = Destination_State$x, y = Destination_State$y)
DS <- DS + coord_map() + ggtitle("Destination State")
DS

# Too much to geocode we plan to comment out
# UniqueOC <- sort(table(df$Orgin.City))
# UniqueOC <- data.frame(UniqueOC)
# colnames(UniqueOC) <- c("OriginCity","count")
# View(UniqueOC)

# UniqueDC <- sort(table(df$Destination.City))
# UniqueDC <- data.frame(UniqueDC)
# colnames(UniqueDC) <- c("DestinationCity","count")
# View(UniqueDC)

# MS Additions....

# Average Departure Delay by Origin State Data Frame
dfNoNA <- na.omit(df) #Remove NAs from df
TempDf <- data.frame(tapply(dfNoNA$Departure.Delay.in.Minutes, dfNoNA$Origin.State, mean))
TempDf$Origin.State <- row.names(TempDf)
names(TempDf)[1] <- "Departure.Delay.in.Minutes"
TempDf <- TempDf[-39,] #Remove Puerto Rico
TempDf <- TempDf[-44,] #Remove U.S Pacific Trust Territories
TempDf[order(-TempDf$Departure.Delay.in.Minutes),]
# View(TempDf)
Origin_State$Mean.Departure.Delay <- TempDf$Departure.Delay.in.Minutes
```

```
# View(Origin.State)

OS <- ggplot(Origin_State, aes(map_id = OriginState))
OS <- OS + geom_map(map = us, aes(fill = Origin_State$count))
OS <- OS + expand_limits(x = Origin_State$x, y = Origin_State$y)
OS <- OS + coord_map() + ggtitle("Origin State w/ Average Departure Delays")
OS <- OS + geom_point(data=Origin_State, aes(x=Origin_State$x, y=Origin_State$y),
size=Origin_State$Mean.Departure.Delay, color="#800000b5")
OS

# Average Arrival Delay by Destination State Data Frame
TempDf <- data.frame(tapply(dfNoNA$Arrival.Delay.in.Minutes, dfNoNA$Destination.State, mean))
TempDf$Destination.State <- row.names(TempDf)
names(TempDf)[1] <- "Arrival.Delay.in.Minutes"
TempDf <- TempDf[-39,] #Remove Puerto Rico
TempDf <- TempDf[-44,] #Remove U.S Pacific Trus Territories
TempDf[order(-TempDf$Arrival.Delay.in.Minutes),]
# View(TempDf)
Destination_State$Mean.Arrival.Delay <- TempDf$Arrival.Delay.in.Minutes
# View(Destination_State)

DS <- ggplot(Destination_State, aes(map_id = DestinationState))
DS <- DS + geom_map(map = us, aes(fill = Destination_State$count))
DS <- DS + expand_limits(x = Destination_State$x, y = Destination_State$y)
DS <- DS + coord_map() + ggtitle("Destination State w/ Average Arrival Delays")
DS <- DS + geom_point(data=Destination_State, aes(x=Destination_State$x, y=Destination_State$y),
size=Destination_State$Mean.Arrival.Delay, color="#800000b5")
DS
```