John F. Kennedy Airport and Simulation Projects

Michaela, Connor, Yitian

Abstract

Flight delay is a serious problem in the world, so we wanted to use some statistical methods to analyze the reason for that. In this project, we used the 2019 data set of flights for New York JFK airport to make an analysis. Before we took a look at the data set, we used our experience and some online references to make some assumptions for the reasons of flight delays. Then, we used visualization to explore the data and some basic analysis. After that, we built some statistical models to do more analysis by using logistic regression, stepwise regression, ridge regression and lasso regression. Simulation data was also an important part here, so we simulated two datasets; one is an easy model with high area under the curve (AUC) and another is a difficult model with low AUC which is close to 0.5. Furthermore, we gave these two datasets to group one and we also received similar data sets from the same group. The groups needed to analyze the data sets that were created by the other group. Later, we needed to compare the original data we made with the predicted data which was made by the other group and made the confusion matrix for them. We also built a R package with one function about using cross-validation to choose the parameter k for K-NN method and making predictions. Furthermore, we built a shiny app to make some interactive plots for flight data specifically similar to the data sets we used in class.

Introduction/Background

Starting with the first assignment, this assignment was meant for us to explore the John F. Kennedy Airport as well as general airport questions. This assignment did not include any data analysis to be discussed, but involved research instead and the findings from this assignment will be discussed a little later on in the paper. Going on to the second assignment, this was one where we had to perform an exploratory data analysis on the JFK Airport data. We discovered here that most of the demographics of the flights range from about twenty to twenty-five percent delayed. We also discovered that most flights are less than seven hours with major outliers showing that some flights last longer than ten hours. We also found that, at our airport, 24,835 flights are on time versus the 6,815 flights that are delayed. For our data, delayed flights that start at 3:51 PM or later are more likely to be delayed than the flights that start on time at 1:23 PM or earlier. The proportion of delays by day for our airport also range from 16.8 percent on Sunday to 30.7 percent on Thursday. The proportion of delays by month range from 13.7 percent starting in September to 27.8 percent in August. The proportion of delays by airline range from 16.5 percent with Republic Airlines to 24.6 percent with Skywest

Airlines. JetBlue Airlines has the most flights in our data with a total of 10,774 flights and Hawaiian Airlines has the fewest flights with a total of 92 flights. There are also a total of 31,650 flights within our data set here. From this exploratory data analysis in this assignment, we made all of the important findings listed above in order to know the data that we are working with.

Next, for assignment three, we talked about logistic regression along with receiver operating characteristics (ROC) curves and the corresponding area under the curve (AUC). We also used this knowledge to do a data analysis on the JFK Airport data set. In other words, we used logistic regression and found an ROC curve with its AUC for our data after doing our own research to do a background study on the topics. Starting with logistic regression, it is the same as linear regression except the response variable in the model has to be binary or has to be made to be binary meaning that there are only two responses for that specific variable. What the logistic regression model is meant to do is that it is meant to model the possibility of a certain result occurring based on the singular characteristics within it. It will be discussed further in the next section. For the ROC curve, it is a technique for visualizing, organizing and selecting classifiers based on their performance (Fawcett, 2006). The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). More detail having to do with the ROC curve along with the AUC will also be discussed further in the next section. These topics were also explored further using the JFK Airport data set. The logistic regression model that we used for this data had an outcome variable of “delay” with the predictor variables being “day”, “carrier”, “depart”, “duration”, and “month”. The ROC curve was also generated along with the AUC and it had an AUC of 0.702. These findings will also be discussed later on.

For assignment four, we had to discuss stepwise regression, including backwards stepwise regression, along with doing an example of it using the JFK Airport flight data set. Stepwise logistic selection is known for being a step-by-step process where a logistic regression model is built and where separate variables are chosen based on statistical significance to be used in the final model. An explanatory variable either gets added or removed depending on which stepwise selection method is being utilized and the model gets tested again to see if that variable is statistically significant within that specific model. The model gets tested after every variable is either added or removed and this process repeats until all variables have been checked within the model. The goal of stepwise regression is to find a combination of independent variables that influence the dependent variable considerably which is done by using several tests such as, for example, the F-test. For this assignment, we had to do three different models.

For assignment five, we had to do research to figure out what ridge regression is along with least absolute shrinkage and selection operator, also known as lasso. Ridge

regression is a method of analysis in statistics for looking at and analyzing multiple regression models that struggle from multicollinearity. When there is an issue of multicollinearity, the least squares estimates are unbiased, but the variances for them end up being huge which, in turn, can cause the estimates to potentially be far from their true value. What ridge regression ends up doing to fix this issue is that it adds a degree of bias to the estimates from the regression which ends up decreasing the value of the standard errors. The goal, by doing this, is that it will hopefully produce estimates that are much more dependable. On the other hand, lasso is a type of linear regression that uses shrinkage. Shrinkage is when the values of the data are shrunk to a certain point towards the center such as the mean. These two methods will be discussed further in the next section. We used the method of lasso in practice on the JFK Airport data set using three different models. After setting up the three different models, finally, ROC curves along with the AUC values were calculated along with confusion matrices so that we could report the accuracy, positive predictive value, sensitivity, and specificity for each of the models. Detailed information about each model along with the results from this analysis will be discussed in the next section.

For assignment six, we had to simulate two different models with binary outcomes for logistic regression that were going to be analyzed by group one. For the first model, the data created from that model were that where group one was expected to find a rather high AUC where the second model was created where group one was expected to find a rather low AUC. Both models included six predictors and overall, there were twelve total simulated terms within the model all of which could be combinations of the six predictors. This is the basis of this assignment and more detail will be provided in the next section.

For assignment seven, we had to perform an analysis on the simulated data of group one that they produced for assignment six. We used stepwise regression along with lasso to do this data analysis on their two “training” simulated data sets. During that analysis, we created the predictor along with all of the estimated betas from the two “training” data sets. We, then, estimated the “y’s” using the “testing” data sets based on the two models for the “training” data sets. That is the basis of the data analysis that we had to do for this assignment. Along with the data analysis, we also had to come up with an idea on an R function that we wanted to create. Just like the rest of the assignments, more on this data analysis and R function idea can be read about below.

For assignment eight, we started the assignment by comparing the beta values from our original with the final model that was estimated by group one. We wanted to see how close the estimated beta values that were estimated by group one were to the true beta values that we created for both models. We also wanted to see how similar group one’s predictors were to the predictors that we used in our models that we had created for assignment six. The last thing that we did for this assignment was that we wanted to compare the predicted “y” outcomes for both the “easy” and “difficult”

simulated data sets that we did in assignment six to the true “y” outcomes that we simulated in assignment six. To do this, we created a confusion matrix for both the “easy” simulated data set and the “difficult” one in R and focused primarily on the accuracy value while discussing the other values that were given to see how well group one had done in predicting our “y” outcomes. The results of this will be discussed further in the next section.

For assignment nine, we had to complete only one R function that is a part of an overall R package. The R function that we created is one that uses ten fold cross validation to find the best k for the K nearest neighbors method, and then, we plugged this k into the model to make predictions for the test data. K nearest neighbor is a very useful method in machine learning, and it can be used for both regression and classification. This will get discussed in a bit more detail in the next section. Cross validation is also an important method to choose the best parameter. Because we never know the test data’s response, we would use cross validation to estimate the test error. This will also get discussed a little later in the next section.

Lastly, for assignment ten, we had to finish the Shiny App that we had been working on. Our completed R package was also due with this assignment. Finally, we had to summarize our project over the entirety of the semester. When we did this, we talked about some unsurprising discoveries that we made along with one surprising discovery that stood out to us the most. We also talked about some points that stood out to us the most from our project over the course of the semester. The Shiny App, R package, and project summary will get discussed in more detail in the next section.

Methods/Results

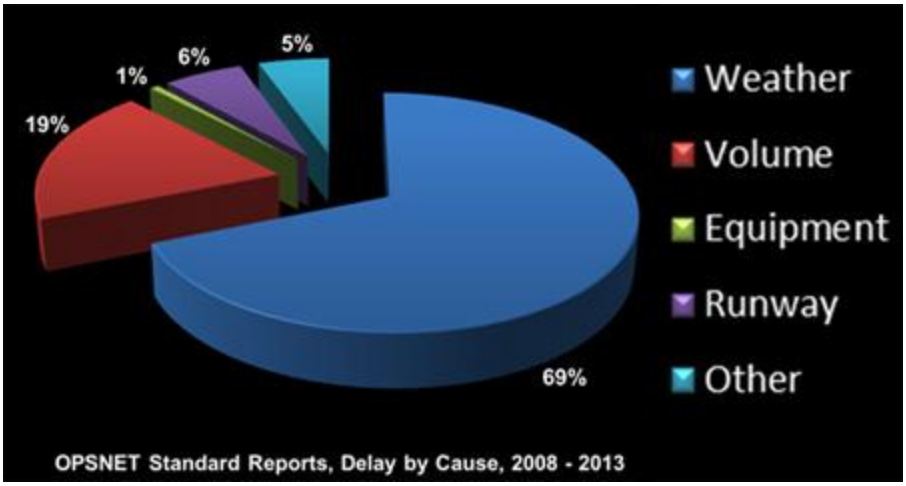
As we said in the section above, assignment one did not have any data analysis. The only method we needed to use to complete this assignment was research to answer the questions that were being posed. The first questions are about New York and the JFK Airport. When asked how many average flights per day are at the JFK Airport, we discovered that it is roughly 1100 per day. We, then, looked up what rank the airport is for the United States in terms of size and number of flights per day and we discovered that it is ranked sixth in the United States by the passengers. When asked to find other background information about the JFK Airport or about New York, we were able to find an airline by the passengers table which is listed below.

Carrier	Passengers (in thousands)	Share
JetBlue	10,277	36.33%
Delta	9,379	33.16%

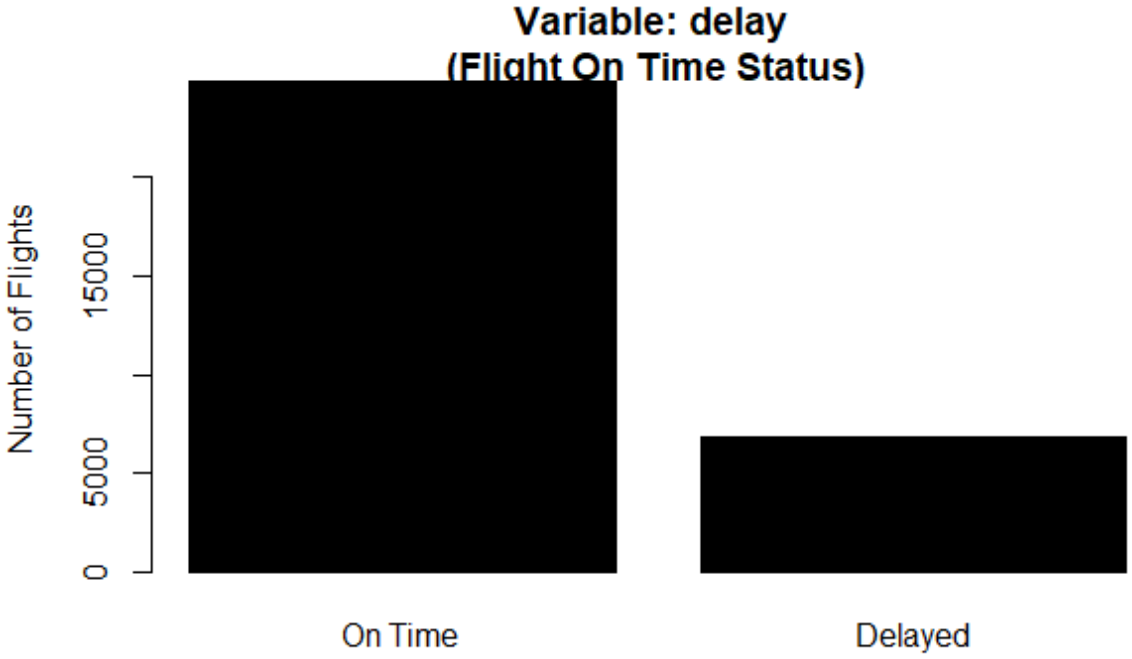
American	3,767	13.32%
Endeavor	2,277	8.05%
Alaska	1,491	5.27%
Other	1,093	3.86%

We also discovered the proportion of delays for 2019 with departures at twenty percent and arrivals at twenty-two percent. The top ten destination airports for 2019 are Los Angeles, San Francisco, Orlando, Fort Lauderdale, Miami, Las Vegas, Atlanta, Seattle, San Juan, and Phoenix. Next are the general airport questions that do not relate just to the JFK Airport but, instead, relate to all airports. The first question that we were asked was are flight delays affected by the time of day of departure of the flight and the answer to that was yes. The early morning flights are the best flights to try and secure since they give passengers a higher chance of having a flight that is on-time, rather than delayed. This is because there is less air traffic in the morning versus later on and throughout the day. It is also best because no other flights have taken off yet to cause a delay which also means that delays have not had a chance to pile up yet with the early morning flights. What “pile up” means is that in the mornings when the first flight leaves, it will not be delayed since no other flights have left yet, but as the day goes on, all it takes is for one flight to leave later than expected causing a domino effect of flights getting delayed because of that one flight which causes this “pile up” effect to happen. The next question that was asked was are flight delays affected by the length of the flight and the answer to that is yes. For the long trip flight, their frequencies are less than others, and the flight would be larger; they have more passengers. It would cost much more money if they were delayed, so the airport may give these flights priority to depart. When these long trip flights arrive at the airport, they also would have less fuel so they cannot fly in the sky for a long time. Hence, the long trip flights also have priority for landing compared with short distance flights. The third general airport question that was asked was are there trends that flight delays are becoming more common in recent years and the answer to that is yes. In 2019, it was reported that there were 302 tarmac delays that lasted for longer than three hours as opposed to 202 reported in 2018 and 193 in 2017. A tarmac delay is when an airplane on the ground is either waiting to take off or has just landed where the passengers have not had the opportunity to exit the plane. The next question that was posed was are flight delays affected by the day of the week and the answer to that question is once again yes. From 2014-2019, Friday has been the day of the week with the most flight delays followed by Monday with Saturday experiencing the least amount of delays. This is potentially the case because Mondays generally have cheaper flights leaving everyone to want to travel on a Monday in order to pay less while many people probably travel on a Friday in

order to spend the whole weekend where they are traveling to. There really is not a good guess on why Saturdays experience the least delays considering flights are generally the cheapest on Saturdays. The best guess would be that travelers might only be able to fly out on Saturday and not want to have to pay higher prices to fly back in on Sunday or maybe they find it pointless to fly out for only one day before having to fly back in for things like work and whatnot. Overall, we were surprised that Saturdays seem to have the least amount of flight delays. The next question asked was are flight delays affected by the month of the flight and the answer to that question is yes. From 2014-2019, August is seen as the month with the most flight delays followed closely by February. August probably sees the most delays because of school returning where students have to potentially fly back into school or back into home due to their vacations being over because of school starting shortly. February could see quite a bit of delays due to weather such as snow or because of Valentine's Day. Even though those two months experience the most delays, it is seen that April and June are the two months that experience the longest delays. The next question asked was are flight delays affected by departure and destination airports and the answer to that is once again yes. For some big and famous cities, such as New York, Chicago, and San Francisco, they are always busy because many people come to these cities for traveling, business and transit. Hence, it is easy for them to have flight delays. The next question asked was how much do flight delays cost airlines. The answer to that question comes from a quote that says "The portion of delay due to weather represented nearly 10 million minutes in 2013. Delays translate into real costs for the operators and passengers. Currently, the cost to the air carrier operators for an hour of delay ranges from about $1,400 to $4,500, depending on the class of aircraft and if the delay is on the ground or in the air. If the value of passenger time is included, the cost increases another $35 per hour for personal travel or $63 per hour for business travel for every person on board" (Federal Aviation...2021). The next question posed was what is the largest cause of flight delays and it turns out that there are several. The largest causes of flight delays are weather, volume, equipment, runway, and others which is shown by the photo of the pie chart below.

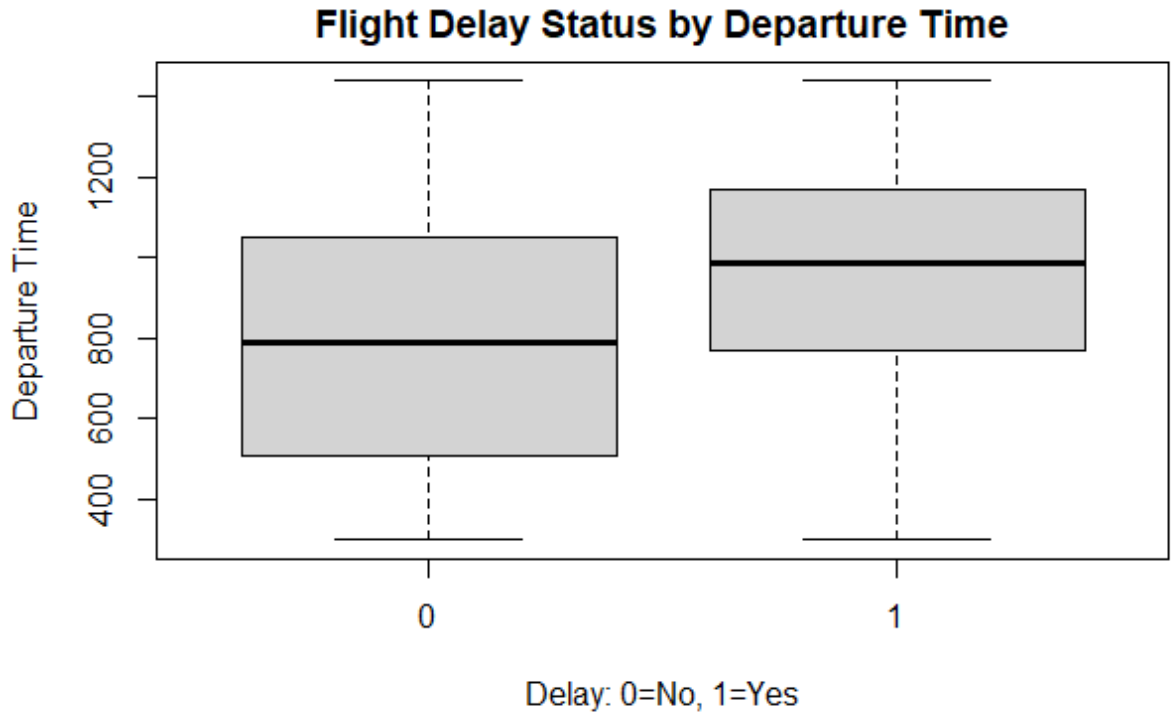


Next, as was stated above, assignment two was mainly doing some exploratory data analysis through different plots that were generated and whatnot to see what exactly the data was that we were working with for the JFK Airport. In the section above, we described what it was that we discovered through this exploratory data analysis since it had to do with important facts of the data itself, and now, we will show some of the plots generated and interpret each of them.

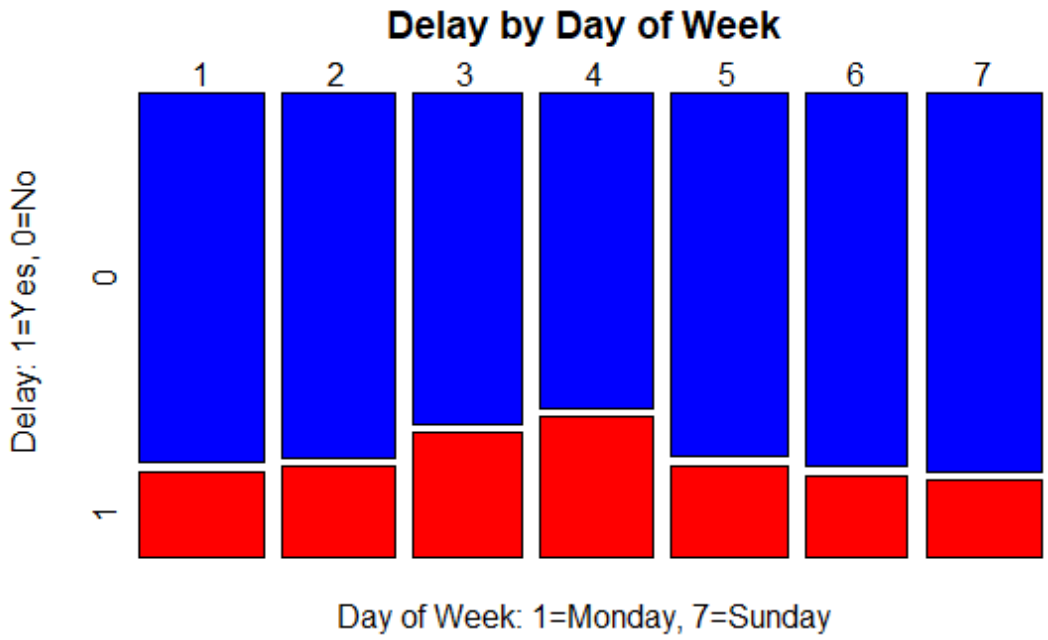


The first question that we wanted to explore when looking at the data was that we wanted to answer whether more flights were delayed or on time especially since delays in general were a major part in the first assignment that we did. In the barplot above, it can be easily seen that, for the JFK Airport specifically, there are more flights that are on time than there are flights that are delayed. There is quite a big difference between the number of flights

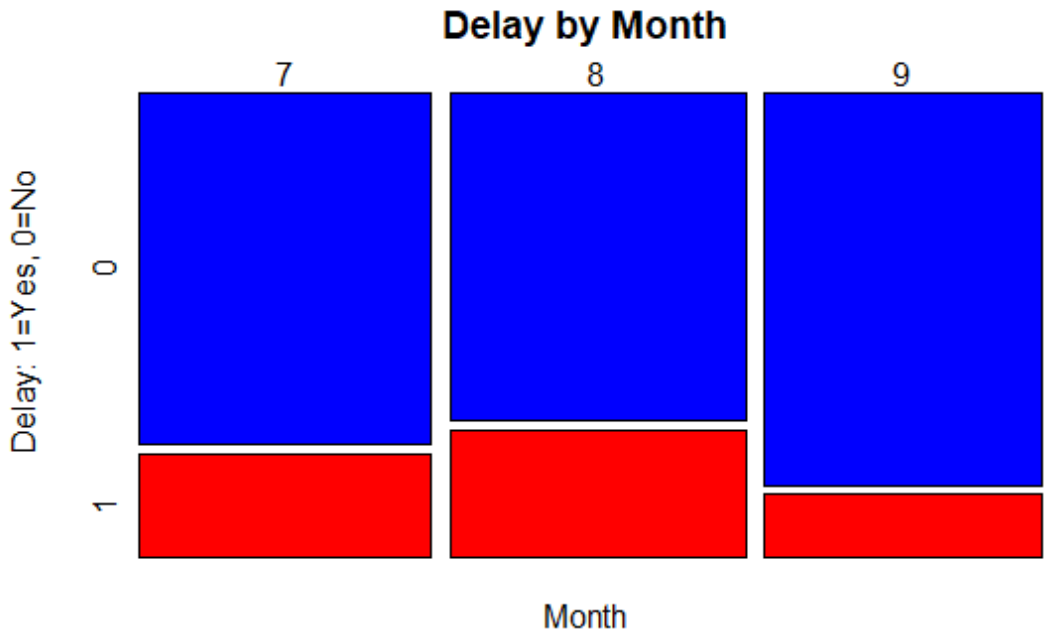
that are on time versus delayed with 24,835 flights that are on time and 6,815 flights that are delayed.



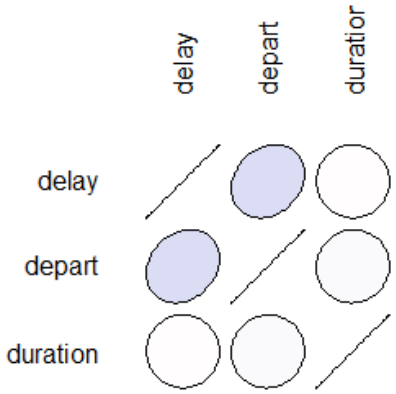
The next question that we wanted to address was whether flight delays are related to departure time. Judging from the boxplot above, the answer to this question is yes. It appears that, for the JFK Airport, if the plane leaves later in the day, there is more likely to be a delay and vice versa meaning that if the plane leaves earlier in the day, it is less likely that the flight will experience a delay. This makes perfect sense to us because if a plane leaves first thing in the morning, there are no other planes yet to cause a delay unlike later in the day after many flights have already taken off causing more air traffic as well as airports waiting for planes to arrive properly to send them out again.



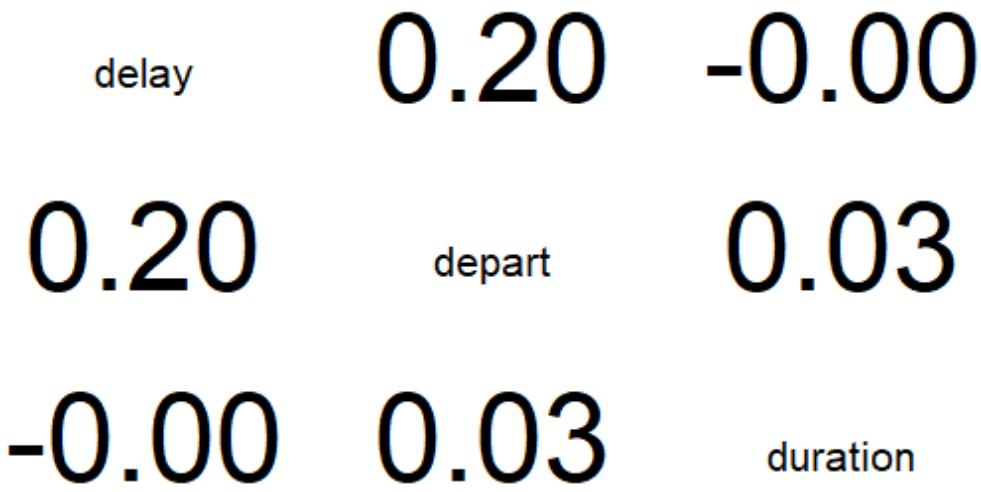
As can be seen by the mosaic plot above, there appears to be more flight delays on Thursdays followed closely by Wednesdays at the JFK Airport. This is a surprising discovery for many reasons. First and foremost, we did not expect that many people to fly out randomly in the middle of the week which causes us to not expect that amount of delays for those two days. It also surprises us because, when we were researching in general as to what days of the week experience the most delays, we discovered then that it was Friday and Monday. Friday made sense to us since people would rather fly out on Friday after work potentially for the weekend to, hopefully, avoid the outrageous amount of people at the airport on Saturday since everyone is off usually on Saturdays and the tickets on Saturdays are generally the cheapest. Monday also made sense because of people returning back to work or even potentially flying out for a business trip or something along those lines. Needless to say, this was a surprising discovery after what our general research told us as well as the fact that these delays are happening right in the middle of the week.



We would, first, like to point out that there are only three months given in this dataset, which is shown by the mosaic plot above. The month that experiences the most delays is August followed closely by July for the JFK Airport. In our general research for which month experiences the most delays, we discovered something similar that we talked about for the first assignment. In our opinion, August experiencing the most delays makes sense since that is around the time that colleges are going back into session and a lot of people are either leaving for or coming back from vacation. Colleges and schools in general starting in or around August is important to note here since that means that students either have to come back home from vacation to get ready for school or they have to potentially fly out to the college that they are going to which is a good reason for why August experiences the most delays. We would imagine that July follows closely since people are, again, leaving for vacation and the fourth of July is also happening at the beginning, so they could also be travelling for that and to potentially see family to celebrate with each other for that holiday. All around, we feel that the results from this data makes sense.



It can also be seen by the correlogram above that, for the JFK Airport, depart and delay appear to be correlated with each other which could potentially be important to take note of. In order to see just how correlated they are, the second correlogram is posted below.



It can be seen above that depart and delay are not highly correlated which is a good thing since a lot of correlation within a model is not a good thing. It can also be more clearly seen that depart and duration are slightly correlated, but not enough to the

point where it shows in the first correlogram. Both of these images just show what could be important information, so we felt like they should be included here as well.

Now, for assignment three, as we said in the previous section, this assignment consisted of learning and using logistic regression as well as receiver operating characteristics (ROC) curves and the corresponding area under the curve (AUC). These methods were then used for our JFK Airport data set to create a specific model and then generate an ROC curve along with the AUC for that curve. Starting with a little bit more detail about logistic regression than what was said in the previous section, this method of statistical analysis is used for predictive analytics and modeling. It is often used to understand the relationship between the dependent variable and at least one independent variable by estimating the probabilities using the proper logistic regression equation. A question that may come to light when talking about logistic regression is what to do if the explanatory variables are not binary. The answer to that is pretty simple. If the explanatory variable is not binary, meaning that it is a multinomial variable, then n-1 binary variables need to be constructed where n is the number of levels of the specific variable. These variables are also called dummy variables. A dummy variable is a variable that will take over a value of either one or zero depending on what the specific category listed is. We should also cover reference levels since that is a topic usually not thought of since there usually is a clear, automatic reference level to use for the explanatory variables. However, that is not always the case. There is no clear route to take when it comes to determining a reference level if it is not automatically clear. There are a few suggestions though. One can choose a reference level that has a minimum sample size with the thought of there being enough “statistical power” to hopefully help the model. The other recommendation is to select “categories” that exhibit a similar “relationship” with the “event of interest” in the model. Now, for the ROC curve, it, first and foremost, considers the classification problem with two classes and we can use the label for that to be {0, 1}. Then we can set threshold z, for example, if z = 0.5, then we compare the predicted value p(x) with the threshold. If p(x) is greater than 0.5, then the label of x is 1; if p(x) is smaller than 0.5, then the label of x is 0. Then we can use the predicted labels and true labels to construct a confusion matrix (Table 1). A confusion matrix is a table representing the details about the performance of algorithms on each label. Then we need to count the number of TN, FP, FN, and TP which will be defined next. TN is called True Negative, that means both the predicted labels and true labels are 0. FP is called False Positive, that means the predicted labels are 1, but the true labels are actually 0.FN is called False Negative, that means the predicted labels are 0, but the true labels are actually 1. TP is called True Positive, that means both the predicted labels and true labels are 1. After that, we need to find FPR and TPR which will be defined next. FPR is called False Positive Rate, which is equal to FP/(TN+FP). TPR is called True Positive Rate, which is equal to TP/(FN+TP). If we select different thresholds, then we can also get different pairs of (FPR, TPR). ROC

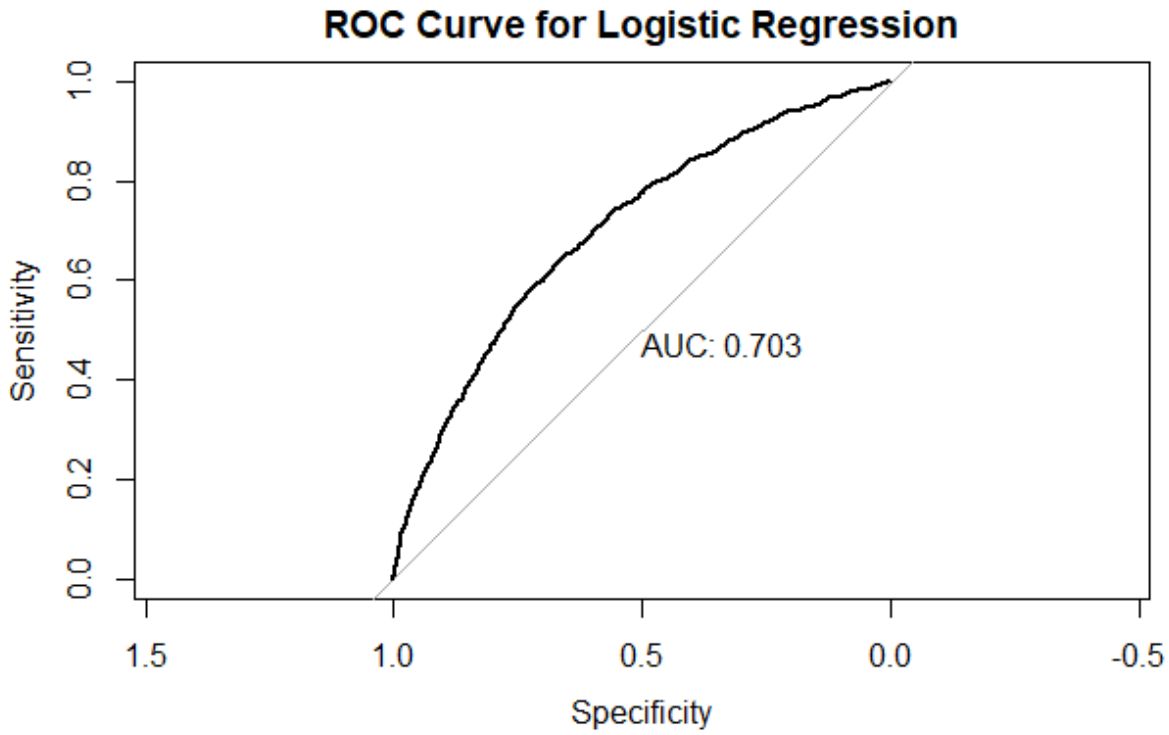
graphs are two-dimensional graphs in which TP rate is plotted on the y-axis and FP rate is plotted on the x-axis (Fawcett, 2006).

N	Predicted as 0	Predicted as 1
True label 0	True Negative (TN) True Negative Rate(TRN)=TN/(TN+FP)	False Positive (FP) False Positive Rate(FPR)=FP/(TN+FP)
True label 1	False Negative (FN) False Negative Rate(FNR)=FN/(FN+TP)	True Positive (TP) True Positive Rate(TPR)=TP/(FN+TP)

Table 1

Above, it was said as to what the AUC is and now, it is time to discuss why calculating the AUC is helpful. The AUC is used as a summary of the ROC curve. The higher the AUC, the better the performance is of the model at distinguishing between the positive and negative classes which is why having the AUC number is helpful for the ROC curve.

Using the knowledge of logistic regression as well as the knowledge of the ROC curve and AUC, we performed an example using the JFK Airport. Before forming the model, we sampled randomly from the JFK Airport data set. We then did a random shuffle which we used to create a subset of the full data set. We then split the subset of the data into an eighty percent training data set and a twenty percent testing data set. After doing that, it was time to build the model based on the training data set. The model that we performed consisted of the outcome variable being ‘delay’ with the predictor variables being ‘day’, ‘carrier’, ‘depart’, ‘duration’, and ‘month’. After putting that model into R, we had to generate an ROC curve along with its AUC based on the testing data set. The ROC curve with its AUC is shown below.



After generating the ROC curve along with the AUC, we wanted to find out the test error rate, the accuracy, the precision, the true positive rate, the false positive rate, the specificity, or true negative rate, and, finally, the false negative rate of the ROC curve generated from the model. To do this, we set a cutoff point of 0.5. We found that the test error rate was 0.2099526, the accuracy was 0.7900474, the precision was 0.6691729, the true positive rate was 0.06477438 , the false positive rate was 0.9352256, the specificity (true negative rate) was 0.9911219, and the false negative rate was 0.008878128. When looking at the model, we discovered that all of the predictors in the model appeared to be significant with the p-value being less than 0.05. Using the ROC curve above, we found that the AUC was 0.703 meaning that our model is good. The reason that the AUC was more useful is that a 0.5 cutoff leads to a false positive rate of over 93 percent. If we need a model with a lower false positive rate and can afford a higher false negative rate, then we would have to use a different cutoff. The cutoff threshold can be changed without detracting much from our predictive strength.

Next is assignment four. For assignment four, as was said earlier, we had to do our research on stepwise logistic regression including backwards stepwise regression and then, we had to incorporate that knowledge for our JFK Airport data set. To explain more of what stepwise selection is doing, we need to discuss subset selection since that is a part of it and represents what both forward and backwards stepwise regression is doing. Why do we need the subset selection? For example, there is a multiple linear regression model $Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$. If we don't use the subset selection method, then we need to put all predictors in the model. When p is very large, the model often

gets a poor prediction because it is easy to overfit the data. Also, less predictors are easier to interpret. Then, we also need to consider which predictors we should select and how to choose the best subset. There are two methods: best subset selection and stepwise selection. For the best subset selection, we fit all p models that contain exactly one predictor, then we fit p and choose 2 models that contain exactly two predictors, continuous until we fit the full model. We, then, look at the resulting models, with the goal of identifying the one that is best. So the total number of possible models for the best subset selection should be 2^p. If p is very large, it suffers from computational limitations. To solve that issue, we can use the stepwise methods. For the forward stepwise selection, it begins with the null model, then adds 1 predictor at a time, until all the predictors are in the model. So, the total number of possible models is $1 + \frac{p(p+1)}{2}$. This is easier to compute than the best subset selection, but it cannot guarantee to find the best subset. When the number of predictors p is greater than the number of observations n, this method can still be used. For the backward stepwise selection, in contrast, it begins with the full least squares model containing all p predictors, and then iteratively removes one least useful predictor per time until there is one predictor in the model. The total number of possible models is $1 + \frac{p(p+1)}{2}$. One thing which is different with the forward stepwise method is that the backward stepwise selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit).

Now onto the models that we did for this stepwise logistic regression assignment. The first model we fit was a logistic regression model with main effects for ‘day’, ‘carrier’, ‘depart’, ‘duration’, and ‘month’; and pairwise interaction terms for ‘day’, ‘depart’, ‘duration’, and ‘month’. We performed backwards stepwise logistic regression using an F-test to prune down the number of predictors in the model. The least significant term was the interaction between ‘depart’ and ‘duration’. However, this term was deemed significant when comparing the full model and a reduced model without this interaction. Ultimately, no terms were removed from the model. The AUC for the model was 0.718.

The second model we fit was a logistic regression model with main effects for ‘day’, ‘carrier’, ‘depart’, ‘duration’, and ‘month’; pairwise interaction terms for ‘day’, ‘depart’, ‘duration’, and ‘month’; along with 2nd and 3rd degree polynomial terms for ‘depart’ and ‘duration’. We performed backwards stepwise logistic regression using an F-test to prune down the number of predictors in the model. Similar to the first model, the least significant term was the interaction between ‘depart’ and ‘duration’. This time, the term was deemed not significant when comparing the full model and a reduced model without that interaction using an F-test. This was the only term removed from the model. Our reduced 2nd model had an AUC of 0.717.

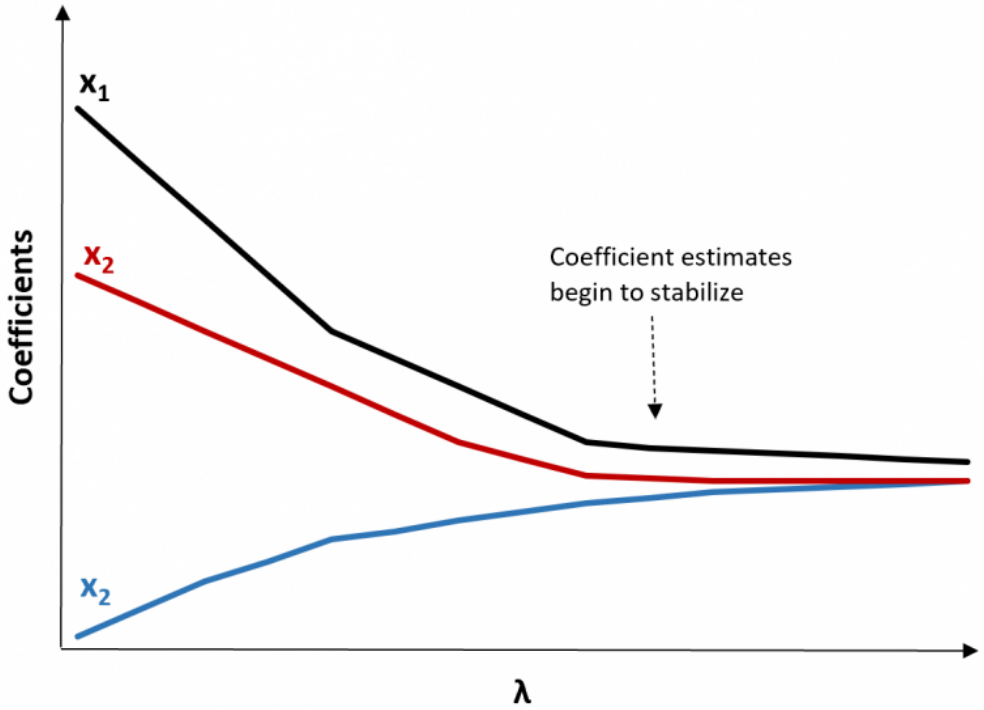
The third model we fit was a logistic regression model with main effects for ‘day’, ‘carrier’, ‘depart’, ‘duration’, and ‘month’; 3-term interaction terms for ‘day’, ‘depart’, ‘duration’, and ‘month’; along with 2nd and 3rd degree polynomial terms for ‘depart’ and ‘duration’. We performed backwards stepwise logistic regression using an F-test to prune down the number of predictors in the model. The least significant term was the 3-term interaction between ‘day’, ‘depart’, and ‘duration’. This term was deemed not significant when comparing the full model and a reduced model without that interaction using an F-test. This was the only term removed from the model. Our reduced 3rd model had an AUC of 0.717.

The three final models ended up with extremely similar AUC values. For this reason we would prefer the first model. When there is not a significant difference in how well the models perform, we should always default to the least complex model. We have always preferred more robust models when it comes to the bias-variance trade-off. We would trust the first model to overfit the data less than the other two models. Fewer terms in the model also makes it easier to explain each estimate to others as there is less to keep track of.

Next is assignment five which was the assignment, as stated earlier, that consisted of doing a background on ridge regression and lasso with an example of that knowledge being performed on the JFK Airport data set. To start talking about ridge regression and lasso, it is important to mention the shrinkage method since that is ultimately what ridge regression and lasso are doing. The shrinkage method is an alternative way of subset selection method, which can regularize or shrink the coefficient estimates towards 0. It may not be obvious to improve the model, but this method can significantly reduce the variance of the coefficient estimates. There are two techniques to shrink them and they are ridge regression and least absolute shrinkage and selection operator, also known as lasso as was also stated earlier.

Ridge regression was talked about in the section above and here, we will go into more detail about what it is doing. An equation that represents what ridge regression is doing and what it is trying to minimize is $RSS + \lambda\Sigma\beta_j^2$ where RSS stands for the sum of squared residuals, $\lambda \geq 0$, where β_j is the average effect on Y of a unit rise in X_j while holding all of the predictors fixed as j ranges from one to p where p is the number of the overall population. It is also important to discuss what the assumptions of ridge regression are since it is important that the model meets those assumptions. The assumptions are linearity, constant variance meaning no outliers, and independence meaning that the data is not connected in any way. These assumptions can also be recognized as being the ones used for regular multiple regression. There are certain steps that are followed when it comes to ridge regression. The first step is to calculate the correlation matrix and variance inflation factor, also known as VIF, values. The VIF calculates how much the variance behavior of an independent variable is influenced or inflated by its interaction or correlation with the other independent variables. When the

values are calculated, if a large amount of correlation is discovered, meaning that the VIF value is between five to ten depending on what you consider to be high enough, then that means that ridge regression is the appropriate method to use and hence why this is the first step of the process. If there is no multicollinearity present within the data, then ridge regression may not be the best choice. The second step of ridge regression is standardization. This is where the independent and dependent variables are standardized by subtracting the means and dividing by the standard deviation where each predictor variable has a mean of zero and a standard deviation of one so that no one predictor variable has so much influence during the ridge regression process. In general, all of the calculations for ridge regression are based on standardized variables. However, when the final coefficients are given, they are modified back into the scale that they originally had. This is important to note so that we know what the ridge regression is doing in R even if the final results are back on the scale that they started out with. The third and final step is to fit the ridge regression model along with choosing a value for λ from the equation above. There are two popular ways for how statisticians choose λ. The first way is to create a ridge trace plot which looks something like the plot below (Introduction to Ridge Regression … 2020 Nov 11).



.

This plot is a representation of the values of the coefficient estimates of λ as $\lambda \to \infty$. When looking at this plot, λ is usually decided as the value where the majority of the coefficient values stabilize which is also pictured on the plot above. The second method of figuring out a decent value for λ is by calculating the test mean squared error, also known as MSE, for each value of λ and then deciding the value of λ based on which value has the lowest test MSE. Those are the steps of ridge regression.

As the disadvantage of the ridge regression, lasso is an alternative technique for the shrinkage method to overcome that disadvantage. Lasso has very similar formulation as the ridge regression, but the penalty term β_j^2 in ridge regression has been replaced by $|\beta_j|$. So the lasso method's coefficient estimate $\hat{\beta}^{lasso} = arg_{\beta} min \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} p\, x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$, λ is the tuning parameter. When λ=0, lasso simply gives the least square fit; when λ is very large, some of the coefficients will be exactly equal to 0. To select the best λ for the model, we always use the cross-validation method to find the λ. Comparing ridge regression and lasso, lasso has a major advantage over ridge regression since lasso is the full model which makes it have smaller variance and makes it easier to interpret. However, neither ridge regression nor lasso will always dominate the other.

The first model we fit was a logistic regression model with main effects for 'day', 'carrier', 'depart', 'duration', and 'month'; and pairwise interaction terms for 'day', 'depart', 'duration', and 'month'. We performed lasso regularization to reduce the estimate of each predictor in the model. This makes our model more robust by trading an increase in bias for a decrease in variance. We compared different penalty terms, λ, using cross-validation. For the goal of minimizing MSE, we found that $\lambda = 1.774333e - 4$ was best. This value of λ reduced the main effects of 'day4' and 'day7', and interaction terms of 'day3:duration' and 'depart:month8' to 0. Using 0.5 as a threshold for predictions, our model has an accuracy of 79.8%, positive predictive value of 80.3%, sensitivity of 98.4%, and specificity of 12.8%. The AUC for the model was 0.717.

The second model we fit was a logistic regression model with main effects for 'day', 'carrier', 'depart', 'duration', and 'month'; pairwise interaction terms for 'day', 'depart', 'duration', and 'month'; along with 2nd and 3rd degree polynomial terms for 'depart' and 'duration'. We performed lasso regularization to reduce the estimate of each predictor in the model. We compared different penalty terms, λ, using cross-validation. For the goal of minimizing MSE, we found that $\lambda = 3.648937e - 5$ was best. This value of λ reduced the main effects of 'day4', the interaction terms of 'day3:duration', and polynomial terms of 'depart^2' and 'duration^2' to 0. Using 0.5 as a threshold for predictions, our model has an accuracy of 79.8%, positive predictive value of 80.3%, sensitivity of 98.5%, and specificity of 12.6%. Our reduced 2nd model had an AUC of 0.717.

The third model we fit was a logistic regression model with main effects for 'day', 'carrier', 'depart', 'duration', and 'month'; 3-term interaction terms for 'day', 'depart', 'duration', and 'month'; along with 2nd and 3rd degree polynomial terms for 'depart' and 'duration'. We performed lasso regularization to reduce the estimate of each predictor in the model. We compared different penalty terms, λ, using cross-validation. For the goal

of minimizing MSE, we found that $\lambda = 8.235722e - 6$ was best. This value of λ reduced the main effects of 'day2', the interaction terms of 'day6:depart', 'day6:duration', 'day2:month8', 'day4:month8', 'depart:month8', 'day4:depart:duration', 'day5:depart:duration', 'day2:duration:month8', 'day4:duration:month9', and polynomial terms of 'depart^2' and 'duration^2' to 0. Using 0.5 as a threshold for predictions, our model has an accuracy of 79.9%, positive predictive value of 80.5%, sensitivity of 98.2%, and specificity of 14.1%. Our reduced 3rd model had an AUC of 0.716.

The three final models ended up with extremely similar AUC values. For this reason we would prefer the first model. When there isn't a significant difference in how well the models perform, we should always default to the least complex model. We have always preferred more robust models when it comes to the bias-variance trade-off. We would trust the first model to overfit the data less than the other two models. Fewer terms in the model also makes it easier to explain each estimate to others as there is less to keep track of.

Moving on to assignment six, this assignment is the one having to do with simulation where we had to simulate two different models with binary outcomes for logistic regression that were going to be analyzed by group one which is what we stated in the section above. Starting with our first model, we simulated a data set that would be relatively easy to create an extremely accurate model. We chose to have 10,000 data points so that models could be sufficiently trained for the next group. The first 6 predictors are main effects, while the last 6 are combinations of the main effects. We decided to have X_1, X_2, and X_3 be correlated normal random variables. To do this we created a 0 vector for the means, and a covariance matrix with 1's on the diagonal and 0.2's everywhere else. The 3 correlated predictors have 0.2 covariance, but are otherwise Normal(0,1). We used a multivariate normal distribution to simulate them. X_4 and X_5 were sampled from normal distributions with means of 5 and -5 respectively. However, each only had a standard deviation of 1. X_6 is a binary predictor sampled from Bernoulli(p=0.5). X_7 is a degree 2 polynomial term of X_2. X_8 is a 3-term interaction of X_2, X_4, and X_6. X_9 is a pairwise interaction of X_1 and X_4, while X_{10} is a pairwise interaction of X_2 and X_5. X_{11} is a 3-term interaction of X_1, X_3, and X_5. Lastly, X_{12} is a degree 3 polynomial term of X_3. The coefficients were all sampled from a Normal(0,1) distribution. They are as follows: B_1 = -0.65, B_2 = 0.06, B_3 = 0.18, B_4 = 0.36, B_5 = 0.07, B_6 = -1.24, B_7 = -2.23, B_8 = -0.34, B_9 = 0.35, B_{10} = -0.07, B_{11} = 0.31, B_{12} = -0.78. When fitting a logistic regression model with all 12 predictors we find that 5 of the main effects and 11 overall predictors are significant at a 0.1 level. We removed the combination predictors from our output files while splitting the data into 67% training and 33% testing data. Since there is little overlap in the main effects aside from the correlated multivariate normal distribution, also known as MVN, random variables, we would expect methods used in previous assignments to be able to fit useful classification models.

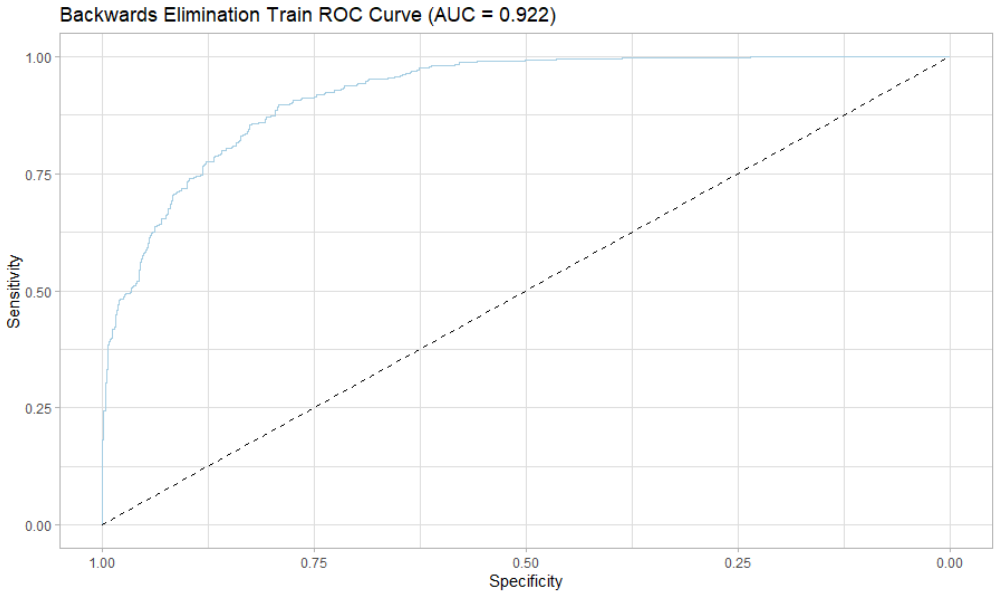
To check that the data would yield a model with a high AUC we decided to test it for ourselves. A full model of main effects, 3-term interactions as well as degree two and three polynomial terms for each of the 6 predictors would fit too much noise to be useful. Backwards elimination is a great tool when we need to prune the number of predictors in this situation. Using this stepwise approach, we would eliminate predictors with an insignificant F-statistic (p-value >= 0.05) one at a time. This was able to reduce our number of predictors from 51 to 28. While this is still fitting quite a bit of noise, it is a vast improvement. Our validation set approach of training the model on 80% of the available training data and testing on the other 20% yielded an AUC of 0.9223. When applying this model directly to the testing only data our model had an AUC of 0.9136.

While stepwise regression yielded good results, we decided to also apply LASSO to fit a model. Fitting the same full model as before, we used cross validation to find an optimal tuning parameter for our penalty term to minimize the MSE. We found that $\lambda = 0.002023364$ was ideal. This was able to shrink 32 predictors to 0, unfortunately including some of the main effects. Our validation set approach of training the model on 80% of the available training data and testing on the other 20% yielded an AUC of 0.9222. When applying this model directly to the testing only data our model had an AUC of 0.9142. Backwards elimination and LASSO may have discarded different terms, but produced nearly identical ROC curves. Either method would be useful for finding a nearly perfect model.

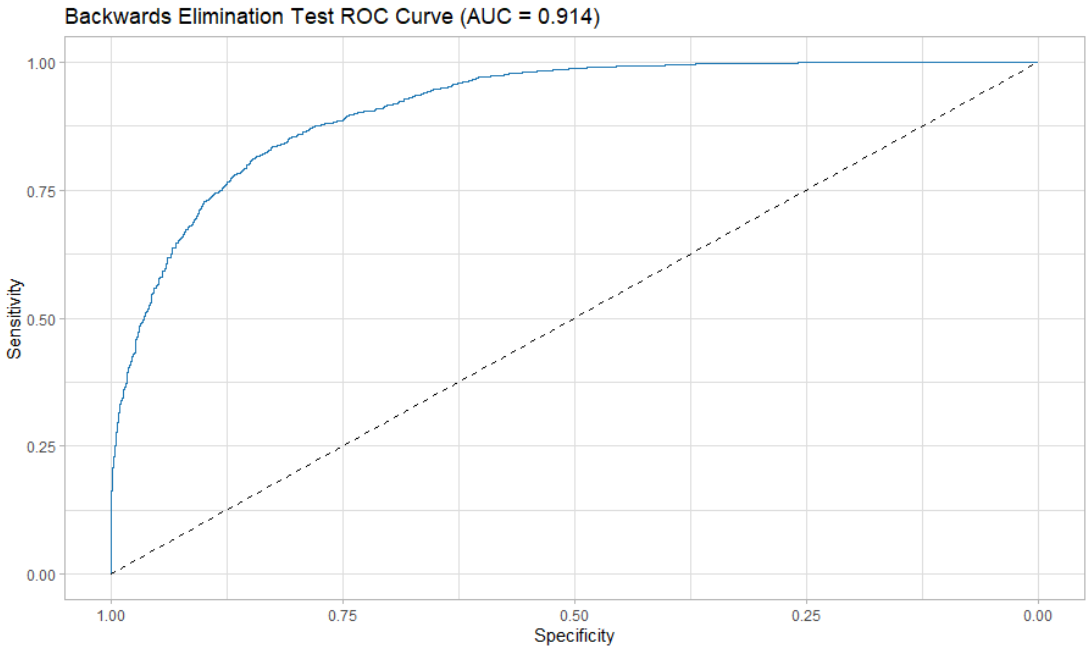
For our second model, we want to simulate the data which has a low accuracy rate. In other words, we need to find the AUC that should be as close to 0.5 as possible. In this model, we choose the sample size equal to 500. There are 12 predictors, the first six are the main effects and the other six are combinations which are interaction terms or polynomial terms. For the main effects, x1, x2, and x3 are correlated normal variables. The number of correlated predictors is 3, and the correlation between predictors is 0.15. Then we set the diagonal variance to be 1 and created a matrix of 3 correlated predictors by using the multivariate normal function “rmvnorm”. We create independent normally distributed predictors x4 and x5. x4 is a normal distribution with mean equal to 2 and standard deviation equal to 0.5, and x4 is in normal distribution with mean equal to -2 and standard deviation equal to 0.5. X6 is a binary variable with probability 0.4. X7 is the pairwise interaction of x1 and x4; x8 is a 3-term interaction of x2, x4 and x5; x9 is a degree 2 polynomial term of x1; x10 is the pairwise interaction of x3 and x2; x11 is a degree 3 polynomial term of x2; and x12 is the pairwise interaction of x2 and x6. The simulated coefficients are as follows: B_1 = 0.92, B_2 = -2.22, B_3 = 1.34, B_4 = -0.59, B_5 = 0.73, B_6 = 0.57, B_7 = -0.24, B_8 = -0.69, B_9 = 0.66, B_{10} = 1.11, B_{11} = -1.46, B_{12} = 1.09. We plugged in all coefficients and simulated x for the logistic regression model to simulate the y value. After that, we can fit the logistic regression by using the glm() function. For the main effects, five of them are significant at level of alpha≤0.10 and 10 total simulated terms in the model are significant.

After simulating the data, we randomly split the simulated data set into the training data set which contained ⅔ of the simulated data set and the last ⅓ of the simulated data set was considered the testing data set. Then, we split the training data set into a training batch and a validation batch where we used 80% for the training batch and the last 20% for the validation batch later. After using backwards stepwise regression to fit the data, we found that the AUC is 0.615. Then, we needed to create probabilities for the testing batch data by inputting the simulated testing data into the predict function as the new data with the logistic model that we created and talked about before. We, then, created the ROC curve using the y value from the simulated testing data as the dependent variable with the probabilities as the independent variable which yielded an AUC of 0.663.

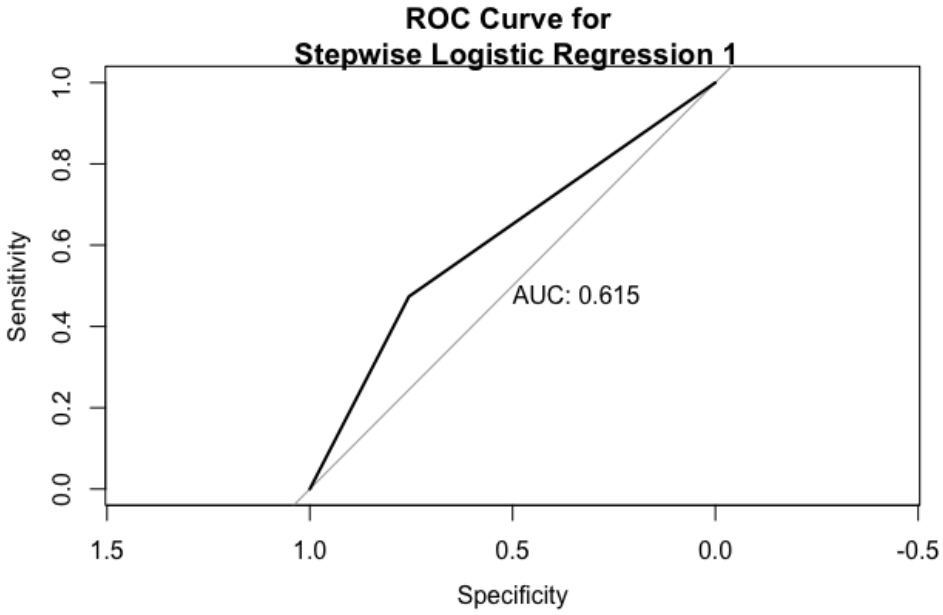
When comparing the two models, it is interesting to see how they differ. Our first model has three coefficients that are not that significant within the regular logistic model before stepwise was done on it whereas for model two, there is only one coefficient that is not significant and another that is not very significant within the model which is less insignificant coefficients than what we saw for the first model. The first model after stepwise was performed had six significant terms whereas the second model did not have any significant terms which makes sense since our second model was our difficult one and our first one was our easy one. For the first model, the significant terms are x_4, x_6, x_2^2, x_3^3, an interaction between x_1 and x_4, an interaction between x_2 and x_4, an interaction between x_2 and x_6, a three-way interaction between x_1, x_3, and x_4, and, lastly, a three-way interaction between x_1, x_3, and x_5. The ROC curve with its AUC for the first model based on the training data set is shown below.



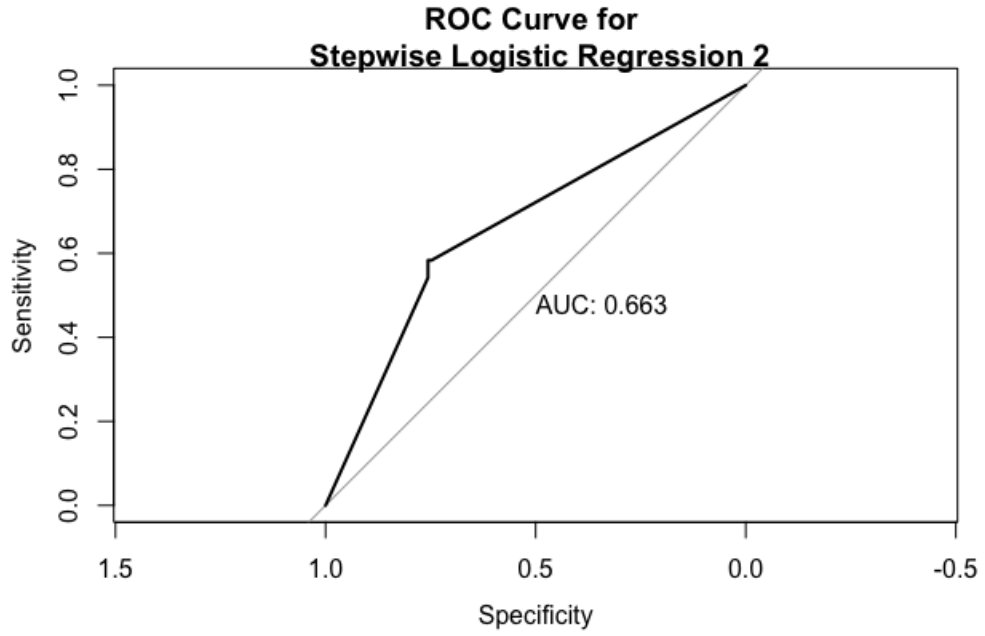
The ROC curve with its AUC for the first model based on the testing data set is shown below.



The ROC curve with its AUC for the second model based on the training data set is shown below.



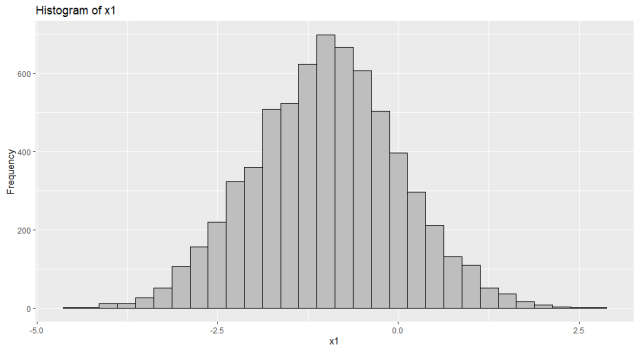
The ROC curve with its AUC for the second model based on the testing data set is shown below.

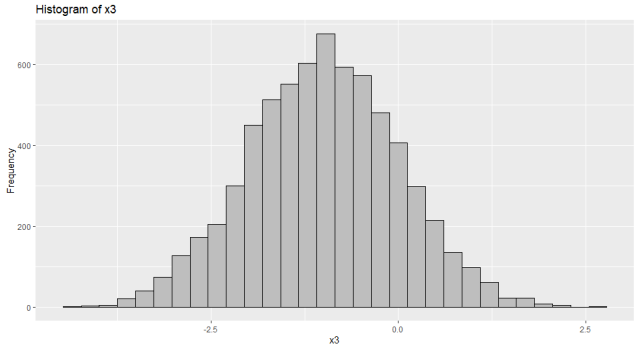


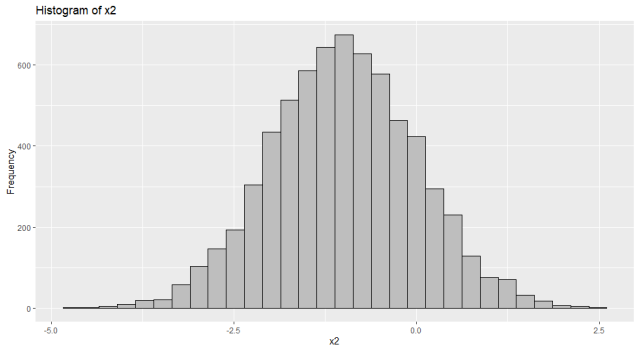
It can be seen that the AUC for the training data set for the second model is quite a bit smaller than the AUC for the testing data set for the second model. While this is interesting to see, it also makes sense since, again, the second model is the difficult one here. The AUCs for the first model for both the training and testing data set do not vary that much. Lastly, as expected and what we were supposed to do, the AUC for model one is extremely high unlike the AUC for model two which was meant to be lower and, therefore, more difficult to determine.

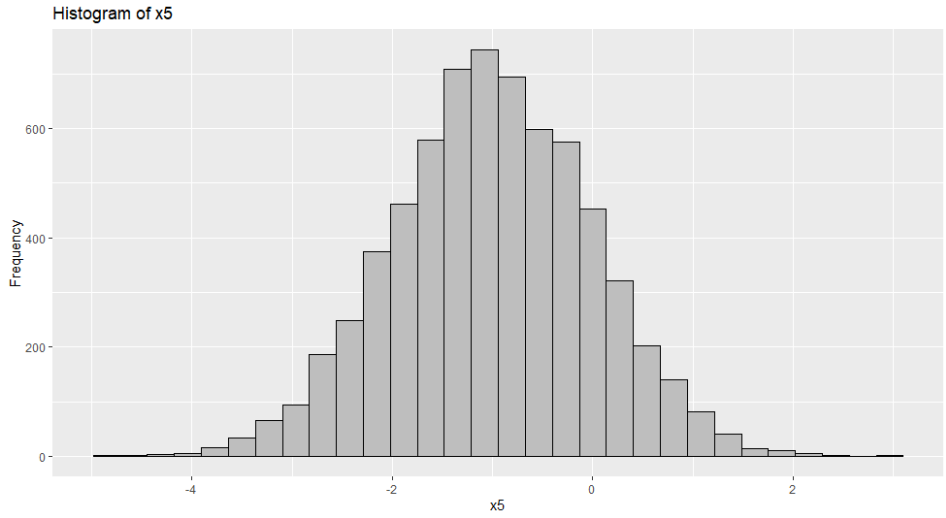
For assignment seven, we had to perform an analysis on the simulated data of group one that they created for assignment six that we stated in the section earlier as well as come up with an idea on an R function for an overall R package to create. To start, we received two sets of simulated data to analyze. Each was split into ⅔ training, where we know the outcome, and ⅓ testing, where the outcome is unknown. The two data sets were denoted as “easy” and “difficult”. The true model for the “easy” data set had an AUC of 0.999. The true model for the “difficult” data set had an AUC of 0.583. We know that these models are made up of 6 main effects, which we were directly given, and 6 combinations such as interactions or polynomial terms. We also know that the main effects are iid observations from a list of a few probability distributions. It is important to note that x_1 to x_6 are of the same values for both data sets. Before we fit our models, we decided to plot histograms of main effects to figure out the underlying distributions. Histograms of x_1, x_2, and x_3 appear to be normally distributed. When calculating means and standard deviations, they appear to be the same. Mean of -1 and standard deviation of 1. When calculating covariances, they appear to all have covariances around 0.2. We believe that these main effects have a Multivariate Normal distribution. A histogram of x_4 appears uniformly distributed. Calculating mean and standard deviation leads us to believe that x_4 has a Uniform(0,1) distribution. A

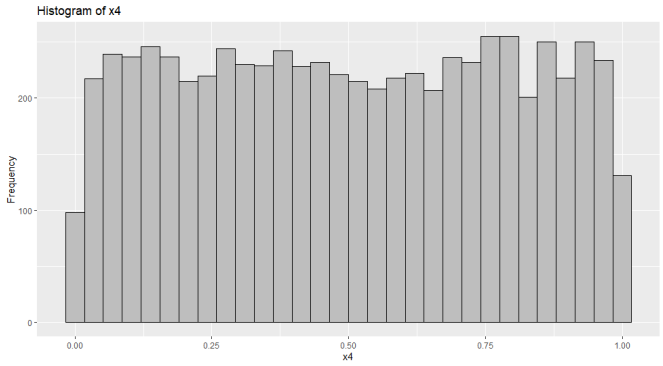
histogram of x_5 appears to be normally distributed. The mean is around -1 and the standard deviation is near 1. This is similar to x_1-x_3, so we checked the covariances. Each of the covariances was roughly 0, so we concluded that x_5 has a Normal distribution, completely independent of x_1-x_3. The x_6 histogram only shows observed data at 0 and 1. When checking the mean and standard deviation, we concluded that x_6 has a Bernoulli(0.7) distribution. The histograms being referred to are shown below.

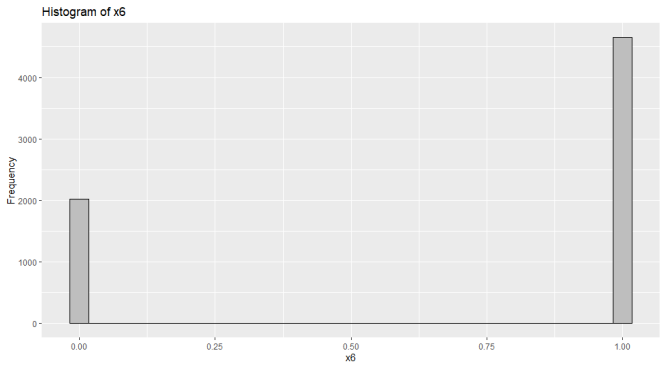












For the first data set we decided to fit a full model of polynomial terms up to degree 3 and up to 3 term interactions. We would evaluate the predictors from the full model with backwards stepwise elimination and LASSO. This model, as discussed above, was split into an eighty percent training data set and a twenty percent testing data set where the training data was used to fit the model on and the testing data was used to discover the ROC curve as well as the AUC. The final model output using backwards stepwise selection for the first training data set contains x_1 through x_6, x_3^2, x_3^3, x_4^2, x_4^3, an interaction between x_1 and x_5, an interaction between x_2 and x_3, an interaction between x_2 and x_4, an interaction between x_2 and x_5, an interaction between x_2 and x_6, an interaction between x_3 and x_4, an interaction between x_4 and x_5, an

interaction between x_4 and x_6, a three-way interaction between x_2, x_3, and x_4, and, lastly, another three-way interaction between x_2, x_4, and x_6. The AUC for this model turned out to be 0.595 which is close to what the AUC is expected to be which is 0.583. This leaves us to believe, of course, that this data is the "difficult" data set for finding the AUC. For the LASSO model we used cross validation to find an optimal tuning parameter for our penalty term to minimize the MSE. This was able to shrink down to 23 predictors including some of the main effects. Our validation set approach of training the model on 80% of the available training data and testing on the other 20% yielded an AUC of 0.601. We discussed common significant terms and were able to parse it down to x_3^2, x_4^3, x_1*x_5, $x_2*x_3*x_4$, $x_2*x_4*x_6$, and x_2*x_5.

For the second data set we decided to fit a full model of polynomial terms up to degree 3 and up to 3 term interactions. We would evaluate the predictors from the full model with backwards stepwise elimination and LASSO. Starting for the second model, backwards stepwise selection was also used along with the same use of the training and testing data sets that were also mentioned above. The model also started the same way as the one prior. The final output and, therefore, final model, consisted of x_1 through x_6, x_3^2, x_3^3, x_4^2, x_4^3, an interaction between x_1 and x_2, an interaction between x_1 and x_5, an interaction between x_1 and x_6, an interaction between x_2 and x_3, an interaction between x_2 and x_4, an interaction between x_2 and x_5, an interaction between x_2 and x_6, an interaction between x_3 and x_4, an interaction between x_3 and x_5, an interaction between x_3 and x_6, an interaction between x_4 and x_5, an interaction between x_4 and x_6, an interaction between x_5 and x_6, a three-way interaction between x_2, x_4, and x_5, a three-way interaction between x_2, x_4, and x_6, a three-way interaction between x_2, x_5, and x_6, a three-way interaction between x_3, x_4, and x_5, a three-way interaction between x_3, x_4, and x_6, a three-way interaction between x_4, x_5, and x_6. Overall, this model yielded an AUC of 0.999 which was exactly the AUC that it should have gotten, leading us to believe for sure that this data was the "easy" data for discovering the AUC. For the LASSO model we used cross validation to find an optimal tuning parameter for our penalty term to minimize the MSE. This was able to shrink down to 33 predictors including some of the main effects. Our validation set approach of training the model on 80% of the available training data and testing on the other 20% yielded an AUC of 0.999. We discussed common significant terms and were able to parse it down to x_3^2, x_4^3, x_1*x_5, $x_2*x_4*x_5$, x_2*x_6, and x_4*x_6.

We plan to write a R package which consists of two functions. The first function is to plot an animated plot. In this plot, we can see the plot has distance as the x-axis, delay as the y-axis, and the color of the scatters are classified by the flights' carriers. There is an animated bar and when you move that bar, you can see different days' information.The second function is using the cross validation to choose the best k, and then plug in this k to the K-nearest neighbors method to make a prediction where the "best k" is discovered by the points that came before which will determine what the

majority point is and that will be what the k is. Overall, we have done and learned many things this semester. These are all of the assignments that we have covered so far this semester.

To start off assignment eight, we will, first, show the true beta values and what our coefficients were for both our “easy” and “difficult” data sets from assignment six and then, we will show the estimated betas and coefficients that group one came up with for both the “easy” and “difficult” data sets we had done in assignment six. Then, we will compare the two. After that, we will discuss the results from the confusion matrix along with what those results mean. Lastly, we will talk about what our initial plan was for our Shiny App since that was also a part of the original assignment.

Predictor (main effects)	Simulated Beta	Predictor (interaction terms)	Simulated Beta
x1	-0.65	(x2)^2	-2.23
x2	0.06	x2*x4*x6	-0.34
x3	0.18	x1*x4	0.35
x4	0.36	x2*x5	-0.07
x5	0.07	x1*x3*x5	0.31
x6	-1.24	(x3)^3	-0.78

The table above is the original easy model that we did.

Predictor (main effects)	Simulated Beta	Predictor (interaction terms)	Simulated Beta
x1	0.92	x1*x4	-0.24
x2	-2.22	x2*x4*x5	-0.69
x3	1.34	(x1)^2	0.66
x4	-0.59	x3*x2	1.11

x5	0.73	(x2)^3	-1.46
x6	0.57	x2*x4	1.09

This table right above is the original difficult model that we did.

```
## Coefficients:
##                                      Estimate
## (Intercept)                           2.31624
## x1                                    0.02683
## x2                                    0.44738
## x3                                   -2.89093   *
## x4                                   -0.14181
## x5                                    0.38598
## x6                                   -5.49278  **
## poly(x2, degree = 3, raw = TRUE)[, 2:3]2 -2.20812  ***
## poly(x2, degree = 3, raw = TRUE)[, 2:3]3  0.07418
## poly(x3, degree = 3, raw = TRUE)[, 2:3]2 -0.06876
## poly(x3, degree = 3, raw = TRUE)[, 2:3]3 -0.80241  ***
## x1:x3                                -0.44585
## x1:x4                                 0.26925  ***
## x1:x5                                 0.06446
## x1:x6                                -0.62179
## x2:x3                                 0.59239
## x2:x4                                -0.10434  .
## x2:x5                                -0.07632
## x2:x6                                -1.71222  ***
## x3:x4                                 0.54051  *
## x3:x5                                -0.63070  *
## x3:x6                                 0.02413
## x4:x5                                -0.07132
## x4:x6                                 0.85925  *
## x5:x6                                -0.68879  .
## x1:x3:x4                              0.15329  *
## x1:x3:x5                              0.36165  ***
## x1:x5:x6                             -0.13920
## x2:x3:x5                              0.10171
## x2:x3:x6                             -0.23207
## x3:x4:x5                              0.10811   *
## x4:x5:x6                              0.14103   .

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

In the screenshot above is what group one had come up with for the “easy” model.

```
## Coefficients:
##                                          Estimate
## (Intercept)                               -6.0754 ***
## x1                                         1.9471
## x2                                        -2.7327
## x3                                         0.3196
## x4                                         1.4210
## x5                                        -0.9230
## x6                                        -2.1042 ***
## poly(x1, degree = 3, raw = TRUE)[, 2:3]2   0.9232 *
## poly(x1, degree = 3, raw = TRUE)[, 2:3]3  -0.3345
## poly(x2, degree = 3, raw = TRUE)[, 2:3]2   0.2019 ***
## poly(x2, degree = 3, raw = TRUE)[, 2:3]3  -2.4241 **
## x2:x3                                      6.8366
## x2:x4                                     -4.3605
## x2:x5                                      0.4399 **
## x2:x6                                      9.5404
## x3:x5                                     -0.7017
## x3:x6                                      6.7138 .
## x4:x5                                      0.8504
## x5:x6                                     -1.8548
## x2:x3:x5                                   2.7283 **
## x2:x4:x5                                  -3.2803
## x2:x5:x6                                   3.9067 *
## x3:x5:x6                                   3.4406 .

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

In the screenshot above is what group one had come up with for the difficult model.

In this assignment, we got the predicted y values and their estimated betas for two data sets from group one. Then, we compared our original 12 betas with the estimated betas which were made by group one. Unfortunately, both two models' betas are not close to the true betas. The only difference for the two models is that the difference between estimated betas and the original betas for the difficult model is larger than for the easy model. We think that is because the difficult data only contains 500 observations. When we split the whole data to train and test data, there are only 333 observations in the train data that the model can fit, so we think these y observations may not be enough for the logistic regression to fit a good model.

The accuracy for the prediction for the easy model is 82.18 percent which is not a bad percentage. It would be better if it were higher, of course, but it is not a bad accuracy rating either. The higher the accuracy rating, the better the model is at predicting the y values. For the sensitivity rating, the higher the rating, the better the model did at positively predicting the y values. This rating ended up being lower than the accuracy rating with it having a percentage of 79.9 percent. This rating also could have been better by being higher, but overall, it could have been worse. Specificity, as a reminder, determines the proportion of actual negatives that are correctly identified which is the true negative rating. The specificity rating is higher than the accuracy rating as well as sensitivity with a percentage of 85.81 percent meaning that the model was really good at predicting the negatives here, also known as the zeros, and then correctly

identifying them which is a good thing. The positive predictive value, also known as precision, is a representation of when it predicted yes, how often is that prediction correct? When looking at this, the higher the number, the better the model is at being correct when it predicts a yes. The percentage is the highest with it being at 90 percent which is a great thing for the model. On the contrary of that, when it was predicted as a "no", how often is that prediction correct? This rating is one that you ideally want to be high in value since you want the negatives to be determined as negatives appropriately. However, this rating is the lowest with it equalling 72.77 percent. Now, to discuss the actual confusion matrix table. It can be seen that the true negative amount is 1638 meaning that this amount was correctly identified as being negative. The true positive value is 1101 which is the number that was correctly identified as being positive here. The false negative value is 412 meaning that those were the ones identified as being negative that should not have been. And lastly, the false positive value here is 182 meaning that those were the ones labeled positive when they actually are not. Overall, the model could have done better at predicting, but it was relatively accurate with what it was predicting and it had a relatively high specificity percentage as well as a really high positive predictive value all of which are really good things to have high percentages for.

The accuracy now for the difficult model is 82.04 percent which, again, is not a terrible number for accuracy. Again, the higher the percentage in terms of accuracy, the better the model is at predicting the true y values and this model has a relatively decent accuracy rating similar to the accuracy rating for the easy model. The sensitivity rating for this difficult model actually ended up being higher than the overall accuracy rating with it having a percentage of 84.03 percent. As we said earlier, the higher the rating, the better it is in terms of predicting the true values of the y's. Specificity, though, has a bit of a lower rating here for the difficult model with it being 77.08 percent. This, again, is meant to identify the actual negatives, so, again, the higher the percentage, the better the model is here at predicting and this percentage is a bit lower than the other two percentages. Next is the positive predictive value. This is once again the highest value with it having a percentage of 90.09 percent which is, again, great for the model. The negative predictive value again has the lowest here as well with it having a percentage of 66.07 percent. That is quite low all things considered and it would be better if it were a bit of a higher percentage. Now for the confusion matrix. The true negative has a value of 100. The true positive has a value of 37. The false negative has a value of 11 and the false positive has a value of 19 here.

The accuracy for both the predicted easy model and the predicted difficult model were really close, but the accuracy for the easy model was slightly higher. The difficult model sensitivity was a little bit higher than the sensitivity for the easy model which we thought was interesting. The easy model specificity was quite a bit higher than the difficult model specificity. The positive predictive value for both models were almost equal with the positive predictive value being slightly higher for the difficult model. The

negative predictive value for the easy model was also higher than the negative predictive value of the difficult model. Overall, because the predicted easy model had higher ratings for everything other than sensitivity, it seems to have performed slightly better which is what we expected since the model for the difficult data was supposed to be difficult to predict in the first place and, therefore, may struggle a bit more with predicting the y's.

Finally, for the R shiny app, we decided to plot the interactive plot. So we plan to set some interactive bars, such as the x variable, y variable and the group. Then the users can change those variables to make the plot they want.

For assignment nine, as we said earlier, we finished our R function for this assignment. As was said earlier, this R function uses the 10 fold cross validation to find the best k for the K nearest neighbors method, and then we plugged in this k to the model to make predictions for the test data.

Now, we will expand more on what the K nearest neighbors method is than what we said earlier. The idea for this method is very simple. We need to find the k nearest neighbors for the test point by using the standard Euclidean Distance. If this is a regression problem, then we can predict our test data is equal to the mean of the k nearest neighbors' responses. If this is a classification problem, then we need to compare which kind of variable has larger probability in the k nearest neighbors. For example, if k is equal to 3, and we found the nearest 3 neighbors to be two triangles and 1 square. Hence we may think the test also should be a triangle.

Now, we will expand a little bit more on what cross validation is since it is also used for this method. In this method, it randomly divides the train data into two parts: a training set and a validation set. The train data randomly separates k folds for the data. For each fold of data, they would be considered the validation set and others as the training set. The model is fitted on the training set and the validation set is plugged into this model to make predictions for the response value. After comparing the predicted response value and the true response value, it is easy to find the mean square error which is an estimate of the test error rate.

Our R function is called KNN_10CV. There are four things that are needed to be inputted to this function; they are Xtrain , ytrain, Xtest and method. Xtrain is the train data without the response value; ytrain is the response value for the train; Xtest is the test data, and there are two options for the method: regression and classification. For example, our function can be used like the following: KNN_10CV(Xtrain=train[,-1] , ytrain$Y, Xtest, method= “regression”). After the user inputs this data, the function would automatically combine the data first. This is because we wanted to have the same variables after we scaled the data. Then we use the model.matrix() function to expand variables to dummy variables' sets. For example, if there is a factor variable in the data called gender with two variables “F” for female and “M” for male, then, after using the model.matrix() function, they would be expanded to gender_F and gender_M with 0 and

1. Then we can scale the data and separate them as Xtrain and Xtest data. In this function, it has 100 K (from 1 to 100) for K-NN methods. Then we use 10 fold cross validation to find which k has the smallest mean square error, and we think this k is the best parameter for this model in the K-NN method. Finally, we use the K-NN method to make predictions for the test data and print the test data.

Now, we will talk about assignment ten. Our R Shiny App focuses on exploratory data analysis (EDA) for the airport flights data earlier in this course. EDA can often be viewed as a tedious (but necessary) step before moving on to more interesting modeling techniques. Many of the plots and numerical summaries involve copying chunks of code over and over again. While many packages exist to help automate this process, some of the underlying trends can be missed by exclusively referring to these “all-in-one” summaries. Our shiny app allows future students of STAT 691P to perform a few basic analysis steps through an interactive application. Students can select up to two variables at a time to view plots and summary statistics. This is dynamic and will update to appropriately reflect the number of variables and types of data. Visualizations for single variables will either produce a histogram (numerical data) or bar plot (categorical data). Visualizations for multiple variables will either produce a scatter plot (both numerical data) or boxplots (one numerical and one categorical). Numerical data will display two measures of central tendency (mean and median) along with two measures of spread (1st and 3rd quartiles, and standard deviation). Categorical data will display counts. Summary statistics for multiple variables will include correlation for two numerical variables or the mean of the numerical variable for each category if it is categorical.

Our R Package is uploaded to github. When people want to use this package, they can use this code to download: install_github('Yitian1349/KNN-10CV'); library(RPackage). If people type “?KNN_10CV” to the Rstudio console, they can see the description of the R function in this RPackage. Our group also put an example in the description. In this example, we used the data from the titanic package. Because there are some missing values in both the train and test data set, we deleted the missing values in the train data and we used the mean value to fill the missing values in the test data. We know this is not the best way to clean the data, but we just want to give a simple example for people to understand the function. After we use the KNN_10CV(), this function will report all the prediction values for the test data. Here is the link for R Package on Github: https://github.com/Yitian1349/KNN-10CV.git.

There were some unsurprising discoveries over the course of the semester that we discovered. The first was that we discovered that August experienced the most delays within our JFK Airport data set. As we said then, this was not surprising since that is the time that colleges are going back into session and people are either leaving for or coming back from vacation, so that discovery made sense to us. Another unsurprising find was that more flights for the JFK Airport were on time than the number

of flights that were delayed. While delayed flights are fairly common to happen, it makes sense that delayed flights would not happen even close to as often as the on-time flights. For the JFK Airport specifically, 24,835 flights were on time and 6,815 flights were discovered to be delayed. The last unsurprising discovery that stood out to us was that flight delays are affected by the time of the day of the flight. For this, the early morning flights are the best to secure to lower the possibility of a flight having a delay. It makes sense because flights are just starting up early in the morning, so there is less air traffic, unlike the air traffic as the day goes on. It is also best because in the early morning when the flight leaves, no other flights have taken off yet other than those early morning flights, but as the day goes on, all it takes is for one flight to leave later than expected to, then, cause a domino effect to happen which could mess up the flights that happen later in the day. That is why it was unsurprising that the early morning flights are the best flights to try and secure to avoid delays.

The only surprising find that stood out to us was that, for our JFK Airport data set, Thursdays actually experienced the most flight delays. We did not expect that many people to fly out in the middle of the week and, therefore, did not expect that many delays to happen. Thursdays only make sense to have the most delays since the flights that happen then are some of the cheapest, but with many people working on Thursdays, it was surprising that Thursday does experience the most delays and that many people were flying out then causing the air traffic to be high despite the tickets being on the cheaper side.

There were a few points that stood out to us throughout the semester. When we did both stepwise logistic regression and lasso regression on our three models from our JFK Airport data set, we discovered both times that all three of our models had very similar AUC values which we thought was a pretty interesting discovery. What made this interesting was the fact that, for two completely different statistical methods and for three different models that were created for each method (six in total), all models used for both methods derived very similar AUC values even though the models used were different and the methods used were totally different. We also noted both times that, since the AUC values were very similar, the simplest model would be the one that we would choose if we had to choose one. The next point we would like to make is that it takes a lot of trial and error to simulate data and find a model from selected x values, beta values, polynomial terms, interaction terms, etcetera for both the high AUC and the low AUC with the low AUC being the more difficult one to come up with. It also was, as expected, more difficult to find the low AUC accurately when going off of the simulated data from the other group, which is also an important point from the semester. Overall, we learned a lot of different topics over the course of the semester which are all important in different ways. This is just a brief summary of a few of those topics.

References

Airline Flight Delays Got Worse in 2019. c2020. New York. [accessed 2021 Sep 8]. https://www.nytimes.com/2020/02/19/business/air-travel-delays-airlines.html

Airports Council International. 19 May 2020. https://aci.aero/news/2020/05/19/aci-reveals-top-20-airports-for-passenger-traffic-cargo-and-aircraft-movements/. ACI Media Releases

An analysis of Flight Delays in the United States. c2018. Bookit.com blogger; [accessed 2021 Sep 8]. https://blog.bookit.com/an-analysis-of-flight-delays-in-the-united-states/

Deane, Steve. 21 February 2021. https://www.stratosjets.com/blog/busiest-us-airports/. Stratos Jet Charters, Inc.

Fawcett, T. (2006) An Introduction to ROC Analysis. Pattern Recognition Letters. 27 (8): 861–874.

Federal Aviation Administration. 2021. *FAQ: Weather Delay*. [online] Available at: <https://www.faa.gov/nextgen/programs/weather/faq/#faq3> [Accessed 11 September 2021].

Introduction to Ridge Regression. 2020 Nov 11. Statology: Nick; [accessed 2021 Oct 7]. https://www.statology.org/ridge-regression/.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R (sixth printing). New York: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). 6.2 Shrinkage Methods. *An Introduction to Statistical Learning with Applications in R* (p. 224). Springer. [accessed 2021 Oct 7].

JFK Airport. 10 September 2021. https://jfkairport.net/statistics/. JFK Airport Essential Airport Information and Services.

Port Authority of New York and New Jersey. 10 September 2021. https://www.jfkairport.com/flight/airlines. John F. Kennedy International Airport.

Ridge Regression. NCSS; [accessed 2021 Oct 7]. https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf.

Sperandei, S. (2014) Understanding logistic regression analysis. Biochemia Medica 24: 12-18.

Stepwise Regression. 2021 February 02. Investopedia: Adam Hayes; [accessed 2021 October 01]. https://www.investopedia.com/terms/s/stepwise-regression.asp.

Stepwise Regression. 2015 September 24. From StatisticsHowTo.com Elementary Statistics for the rest of us: Stephanie Glen; [accessed 2021 October 01]. https://www.statisticshowto.com/stepwise-regression/.
The Best Days and Worst Days and Times to Fly to Avoid Airport Delays. 2019 Nov 15. New York. Lynda Baquero; [updated 2019 Nov 19; accessed 2021 Sep 8]. https://www.nbcnewyork.com/news/local/airport-delays-ewr-jfk-lga/2205292/.

US. Department of Transportation. 10 September 2021. https://www.transtats.bts.gov/airports.asp?20=E. Bureau of Transportation Statistics.

Contributions

In the first assignment, Connor introduced New York's JFK airport's information; Michaela and Yitian answered some questions about whether some variables would affect flight delays. In the second assignment, Connor and Michaela made some plots to analyze the data set and answered questions about the data set. In the third assignment, Michaela introduced the definition of logistic regression; Yitian introduced the definition of ROC and AUC; Connor applied logistic regression on our JFK Airport data and found the AUC value and ROC curve by R. In the fourth assignment, Michaela introduced the goal, advantages and disadvantages about the stepwise regression; Yitian introduced best subset selection, forward stepwise regression and backward stepwise regression; Connor applied stepwise regression on the JFK Airport data set and then found the ROC curve and AUC value. In the fifth assignment, Michaela introduced ridge regression; Yitian introduced lasso regression; Connor applied both lasso and ridge regression on the JFK Airport data set and found the ROC curve and AUC value. In the sixth assignment, Connor simulated the data for the easy model with high AUC; Yitian simulated the data for the difficult model with the AUC that should be as close to 0.5 as possible. In the seventh assignment, Connor and Michaela worked together to do some visualization and figure out which data set is the easy model and which is the difficult model; Yitian thought of a topic of the R function. For the preliminary report, Michaela finished most of it; Yitian finished the Abstract and Contributions in the report. For assignment eight, Yitian did the comparison of the betas along with the comparison of the coefficients for our "easy" and "difficult" models and data sets that we did for assignment six with that of what group one came up with for each; Michaela did the confusion matrices and interpreted the results; Yitian also came up with the idea for the Shiny App. For assignment nine, Yitian finished the R function in our R package and Connor gave a Shiny App update. For assignment ten, Connor finished the Shiny App, Yitian finished the R package, and Michaela did the summary of the project over the course of the semester. For the final project report, Michaela added

in assignments eight, nine, and ten into the “introduction/background” section as well as added them into the “methods/results” section. Michaela also did the corrections needed on the preliminary report as well as corrected the mistakes on those three assignments before adding them in.