Airplane Delay Analysis

Data 606 - Team 8

Tejendra Naga Pavan Gottumukkala
Dennis Felson
Vishnu Regimon Nair
Anna Iuferova
Romi Punian
Charles Roberts

Introduction

In the United States airline travel can be one of the most stressful aspects of any trip, however, it is something that millions of people do every day. According to the Bureau of Transportation Statistics a total of 1.1 billion flights occurred domestically during 2019 (www.bts.gov, 2019). Furthermore, while flying numbers have continued to increase, there has not been a simultaneous increase in airline effectiveness and timeliness. In 2019, 21 percent of flights had significant delays which exceeded previous years (Chokshi, N). As delayed or canceled flights are a perpetual and growing problem facing the airline industry we have decided to research the factors that affect arrival delays.

Due to the sheer volume of data and information that exists on the subject we decided to narrow the focus of our project. This paper will investigate the factors that affect airplane arrival delays at JFK airport in December 2019. In 2019, JFK airport received over 60 million flights and historically is one of the busiest airports in the United States (DMR, 2019). Further, by focusing on December (one of the busiest travel months of the year) we can ascertain the impact of heavy traffic.

Methodology

First and foremost, the dataset and its source will be acknowledged. Then the dependent and independent variables that are significant to this project will be listed, along with the units of the variables and whether they are continuous or discrete.

In order to highlight the specific dataset that will actually be worked on, the wrangling and cleaning procedures will be detailed. Once the data has been treated and organized, we review and discuss four exploratory analysis questions which increase our knowledge of the dataset and illustrate significant details of JFK airport.

After familiarizing ourselves with the dataset we will try three different sampling methods, these methods are simple random sampling, stratified sampling and cluster sampling. The purpose is to see which method seems to lead to the subset of data that best reflects the population dataset will be utilized. This could inform future studies on subject where the targeted population is too large to test directly.

Following these essential introductory tasks, four statistical tests will be conducted:

1. Linear Regression Analysis
2. Logistic Regression Analysis
3. Classification/Regression Tree
4. Cross Validation

Once the necessary tests are completed, there will be a conclusion which summarizes our findings, discusses the limitations of our analysis and explores what future research might be appropriate.

Data Source

We found the data for our project from Kaggle which was sourced from the U.S. Department of Transportation's Bureau of Transportation Statistics which tracks the on-time performance of domestic flights operated by large air carriers.

The datasets contain daily airline information from 2009 to 2019 including flight information, carrier company, taxing-in and taxing-out time, and generalized delay reasons.

The original datasets are Open Data Commons and consist of 22 columns and over 7 million rows. We chose to look at the 2019 dataset for December at JFK airport where we have just over 10 thousand rows.

Metadata

In this section the variables essential to our research will be highlighted along with their units of measure (if applicable) and whether the variable is continuous or discrete. Also note that new variables (such as dummy variables) may be created if our test calls for it.

Dependent Variable:

- Airplane delay - Total time of delayed flight (Units: Minutes, Type: Continuous).

Independent Variables (12 total):

- TAXI_OUT - The time elapsed between departure from the origin airport gate and wheels off. (Units: Minutes, Type: Continuous)
- TAXI_IN - The time elapsed between wheels down and arrival at the destination airport gate. (Units: Minutes, Type: Continuous)
- WHEELS_OFF - time an aircraft takes off. (Units: Measure of time, 24-hour clock values. Type: Continuous)

- WHEELS_ON - time an aircraft lands on the runway (Units: Measure of time, 24-hour clock values. Type: Continuous)
- AIR_TIME - airborne hours of aircraft (Units: Minutes, Type: Continuous)
- DISTANCE - The distance between origin and destination (Units: Miles, Continuous variable)
- CARRIER_DELAY - delay as due to circumstances within the airline's control (Units: Minutes, Type: Continuous)
- WEATHER_DELAY - total time of delay due to weather (Units: Minutes, Type: Continuous)
- NAS_DELAY - delay due to National Aviation System (Units: Minutes, Type: Continuous)
- SECURITY_DELAY - security delays or cancellations (Units: Minutes, Type: Continuous)
- LATE_AIRCRAFT_DELAY - A previous flight with the same aircraft arrived late, causing the present flight to depart late (Units: Minutes, Type: Continuous)
- OP_UNIQUE_CARRIER - Unique Carrier Code (Units: N/A, Type: Discrete)

Data Cleaning and Wrangling

Our process for cleaning and wrangling our data was as follows:

- Read the CSV for 2019 flight data into r
- Discovered that the Dataset is extremely large (7.4 million rows) and so we needed to create a new dataset that was a subset of this data
- Created a new dataset from the original dataset in which we filtered for dates from December 1st, 2019, to December 31st, 2019
- Filtered our dataset to look specifically at flights arriving at JFK airport
- Removed all NA values and converted NA values in our delay columns (CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY and SECURITY_DELAY) to 0 since those NA values represented days where there were no delays and thus 0 minutes delay is an equivalent value

Consequently, we effectively managed to create a new dataset labeled ‘air_delay’ that contained all flights to JFK for in the month of December, with the same 22 columns as the original dataset and a little over 10,000 rows.

Exploratory Analysis: Guiding Questions

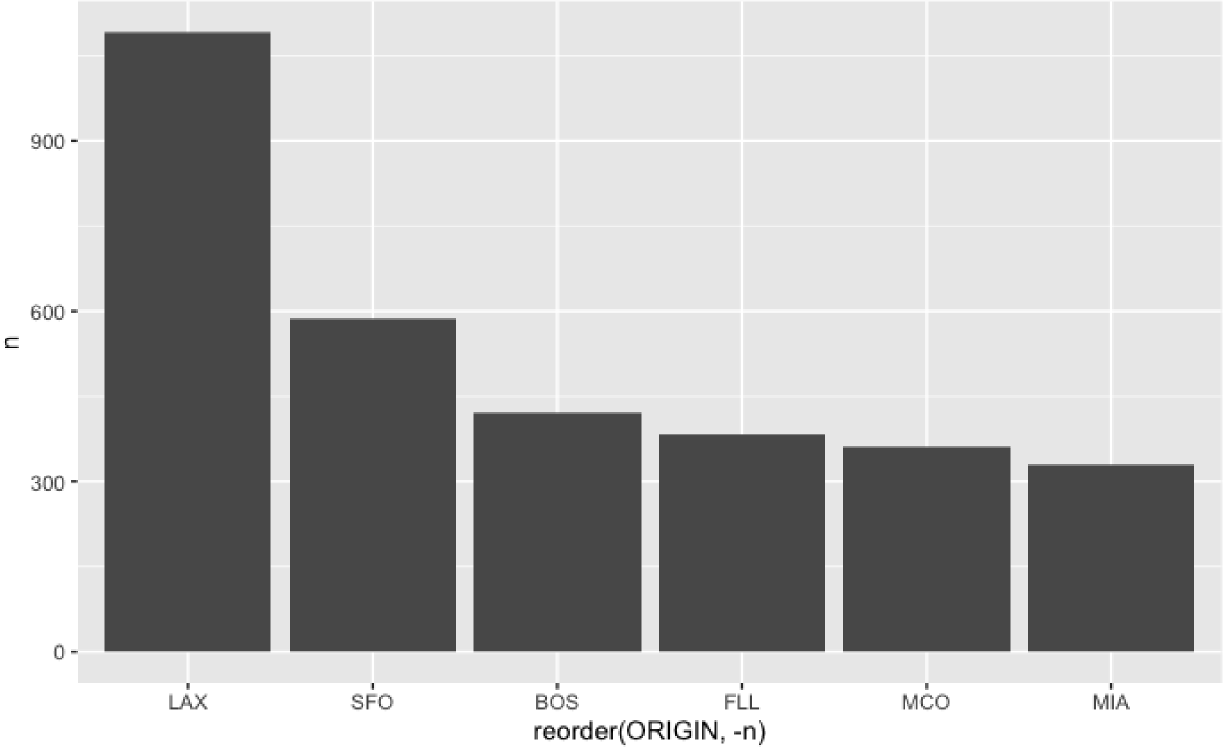
We decided to create and research four guiding questions to guide our exploratory analysis. These questions are:

1. From which airport do most flights come to JFK in December 2019?
2. Which airports have the most delay to JFK in December 2019?

3. Which airline has the highest percentage of flights delayed to JFK in December 2019?
4. Which Carrier has the highest and lowest average delay of its flights to JFK in December 2019?

Question 1: From which airport do most flights come to JFK in December 2019?

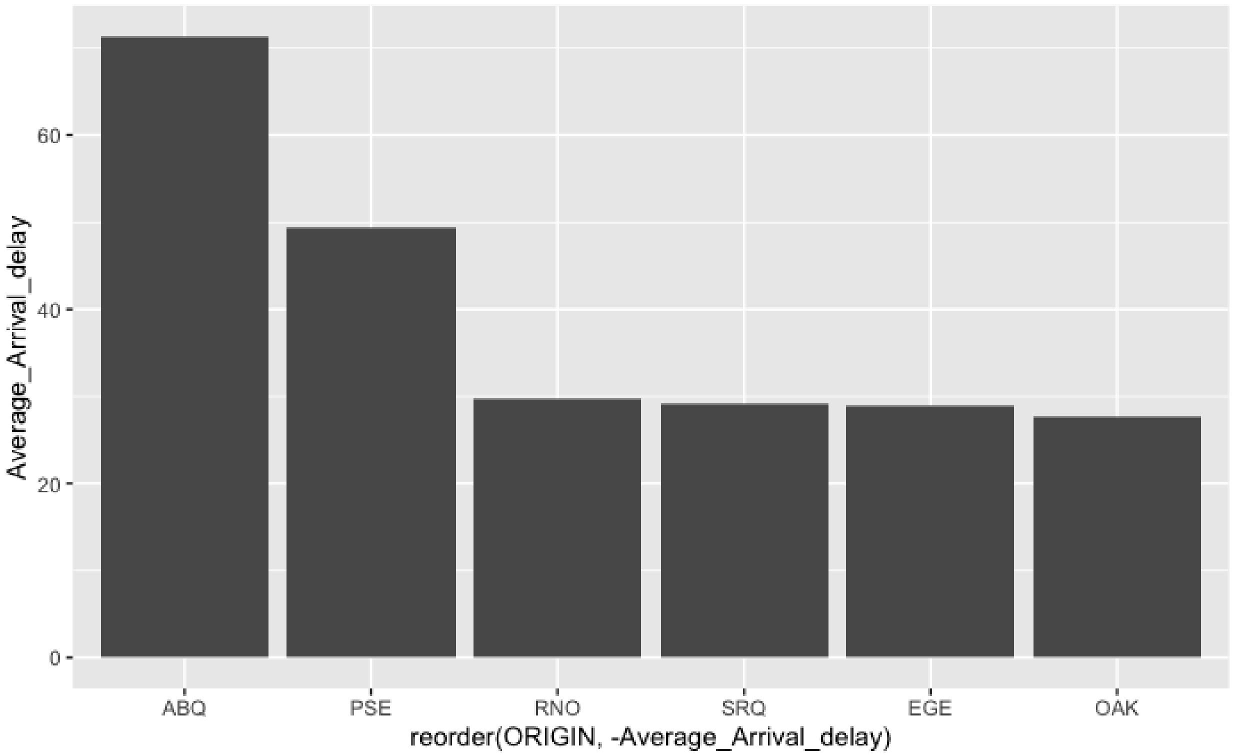
Knowing the airport that flies the most to JFK is a foundational question for this analysis because it details the airport that makes up the highest percentage of our data points. To illustrate the answer to this question, we constructed the bar-chart below. Where the x-axis represents the name of the origin airports and the y-axis represents the number of flights that airport had to JFK.



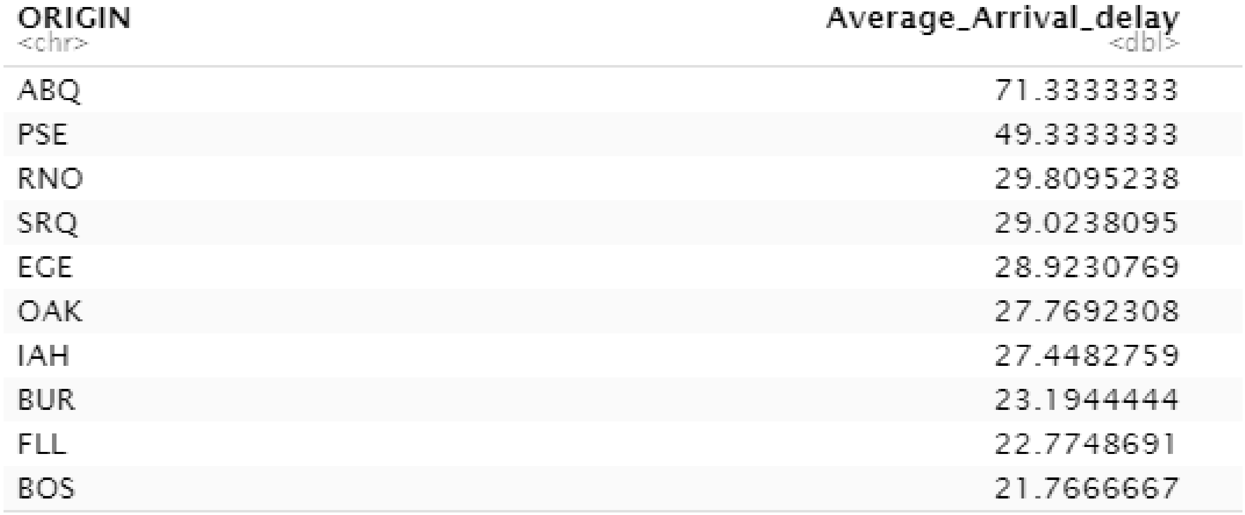
It is evident that most planes come from Los Angeles (LAX) with more than 1000 flights during December 2019. The second airport is San Francisco with around 600 flights.

Question 2: Which airport has the highest average delay time to JFK?

Now that we know which airport has the most flights to JFK airport we can see which airport has the highest average delay time to JFK. To illustrate the answer to this question, we constructed the bar-chart below. Where the x-axis represents the name of the origin airport and the y-axis represents the average delayed arrival time (in minutes) that flights from that airport had to JFK.



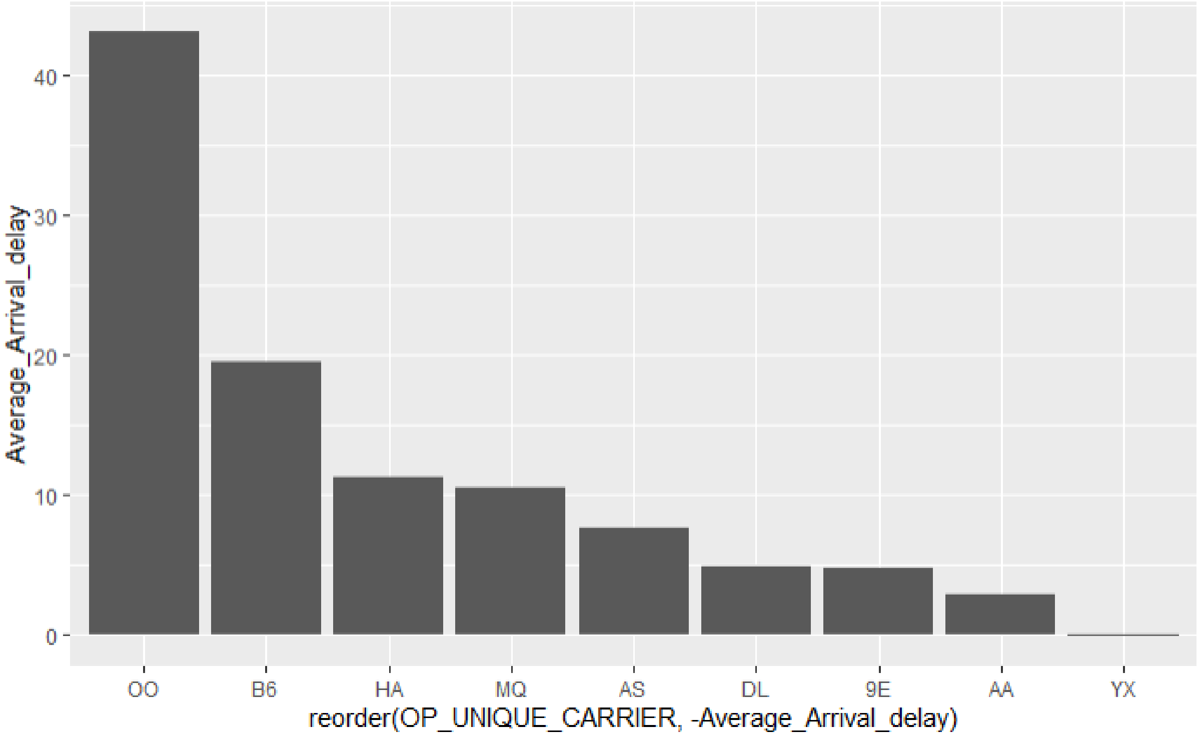
Looking at the top five airports we can see that Albuquerque (ABQ) from New Mexico has the most average delay time, with on average a 71 minute delayed arrival time and the second one is PSE (Puerto Rico) with 49 minute delayed arrival time.



ORIGIN <chr>	Average_Arrival_delay <dbl>
ABQ	71.3333333
PSE	49.3333333
RNO	29.8095238
SRQ	29.0238095
EGE	28.9230769
OAK	27.7692308
IAH	27.4482759
BUR	23.1944444
FLL	22.7748691
BOS	21.7666667

Question 3: Which Carrier has the highest and lowest average delay of its flights to JFK in December 2019?

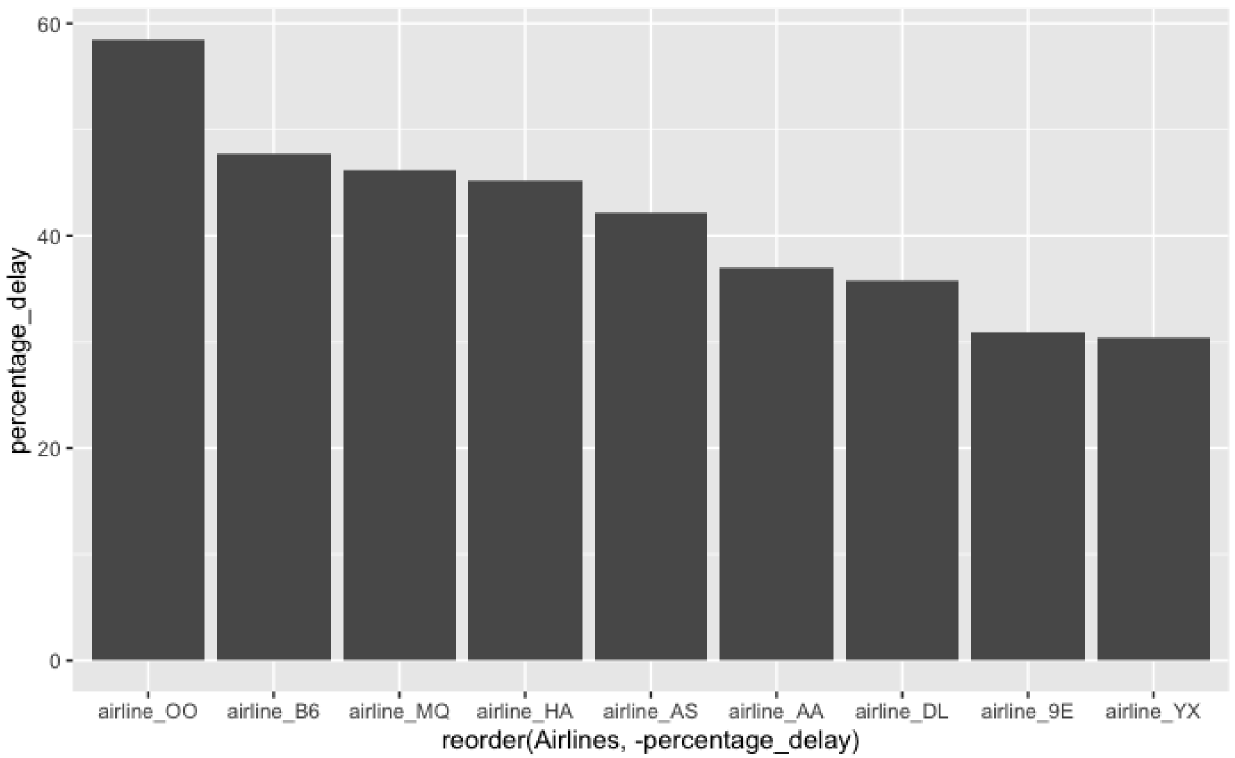
Before conducting any advanced statistical tests, it would be interesting and informative to view the airline that has the highest and average delay time to JFK airport. To illustrate the answer to this question, we constructed the bar-chart below. Where the x-axis represents the name of the carrier and the y-axis represents the average arrival delay time (in minutes) to JFK airport.



From the bar chart we created we were able to see that Skywest (OO) airlines had the highest average arrival delay to JFK airport and Midwest airlines(YX) had the lowest average arrival delay to JFK airport.

Therefore, next time you find yourself in the USA and wanting to go to New York, we would recommend using Midwest Airlines. However, these values might be influenced by outliers (for example, if one flight was delayed multiple days, that would increase the average significantly) which leads into our third question.

Question 4: Which airline has the highest percentage of flights delayed to New York?
As aforementioned we want to be able to account for outliers so we want to look at the airline that had the highest percentage of flights delayed. We counted the number of flights that had an ‘Arrival_Delay’ greater than 0 (count of delayed flights) and divided it by the total number of flights (Delayed + Not Delayed flights) to get the delayed flight percentage. To illustrate the answer to this question, we constructed the bar-chart below. Where the x-axis represents the name of the carrier and the y-axis represents the percentage of flights that experience arrival delays to JFK airport.



Our bar chart backs up our initial findings, as Skywest (OO) airlines has the highest percentage of flights delayed to New York and Midwest airlines (YX) has the lowest percentage arrival delay.

However, we found one difference, which is that American Airlines (AA) flights might not have a long delay time on average but it has a relatively higher percentage of flights that are delayed.

Sampling Techniques

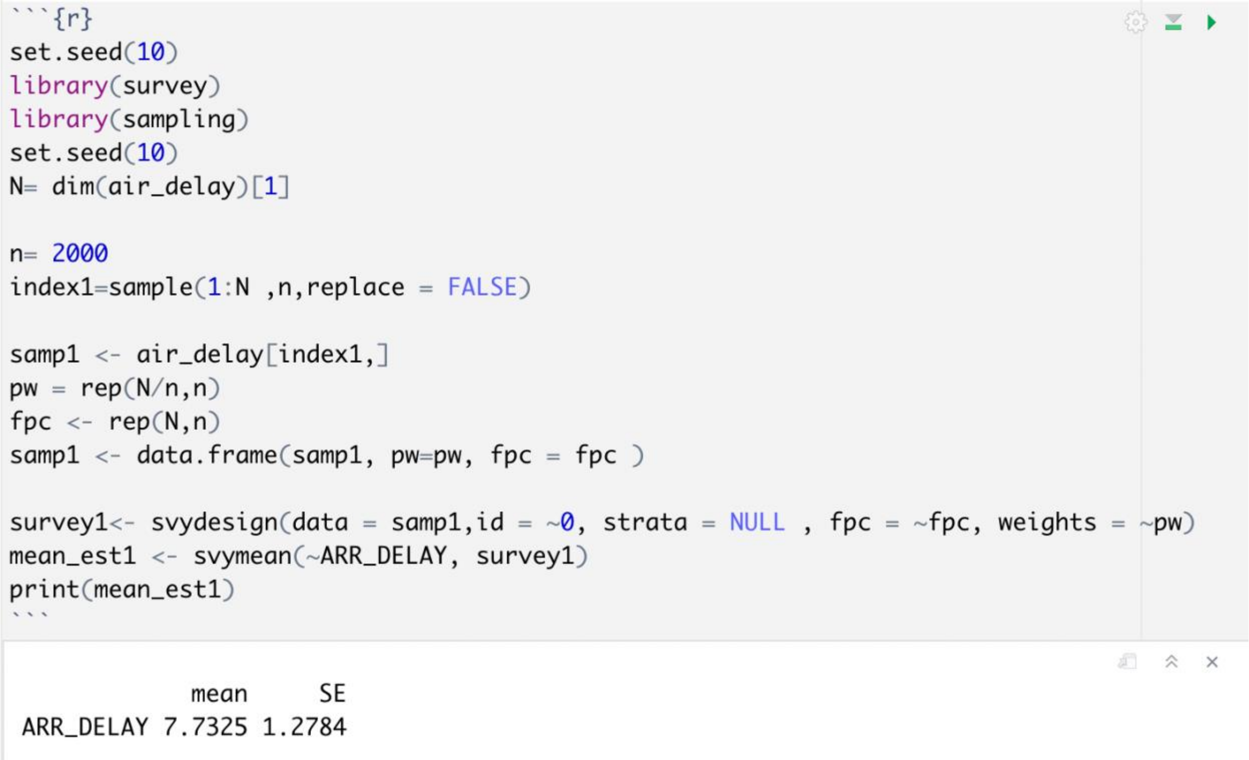
While we have greatly reduced our dataset when compared to the original, it is still valuable to see which sampling method would be best for our dataset as it could inform future studies on the subject. Consequently, before proceeding with testing it's essential to acquire the sample that best represents our population dataset. Three different sampling techniques will be introduced and compared, with these techniques being simple random sampling, stratified sampling and cluster sampling.

We calculated the average arrival delay of flights to JFK airport in December 2019 to be 9.802696 minutes for the population data which has 10,385 values. Whichever of the sampling techniques is closest to the population mean will be the method and subset of data that is best to be tested.

Part 1: Simple Random Sampling

When conducting simple random sampling (SRS) a specified number of samples are picked, with each data point having an equal probability of being chosen. In this case 2000 rows are randomly chosen, as we believe it to be both a large enough size but exceedingly more manageable than the original 10,385 values.

Simple Random Sampling with sample size = 2000



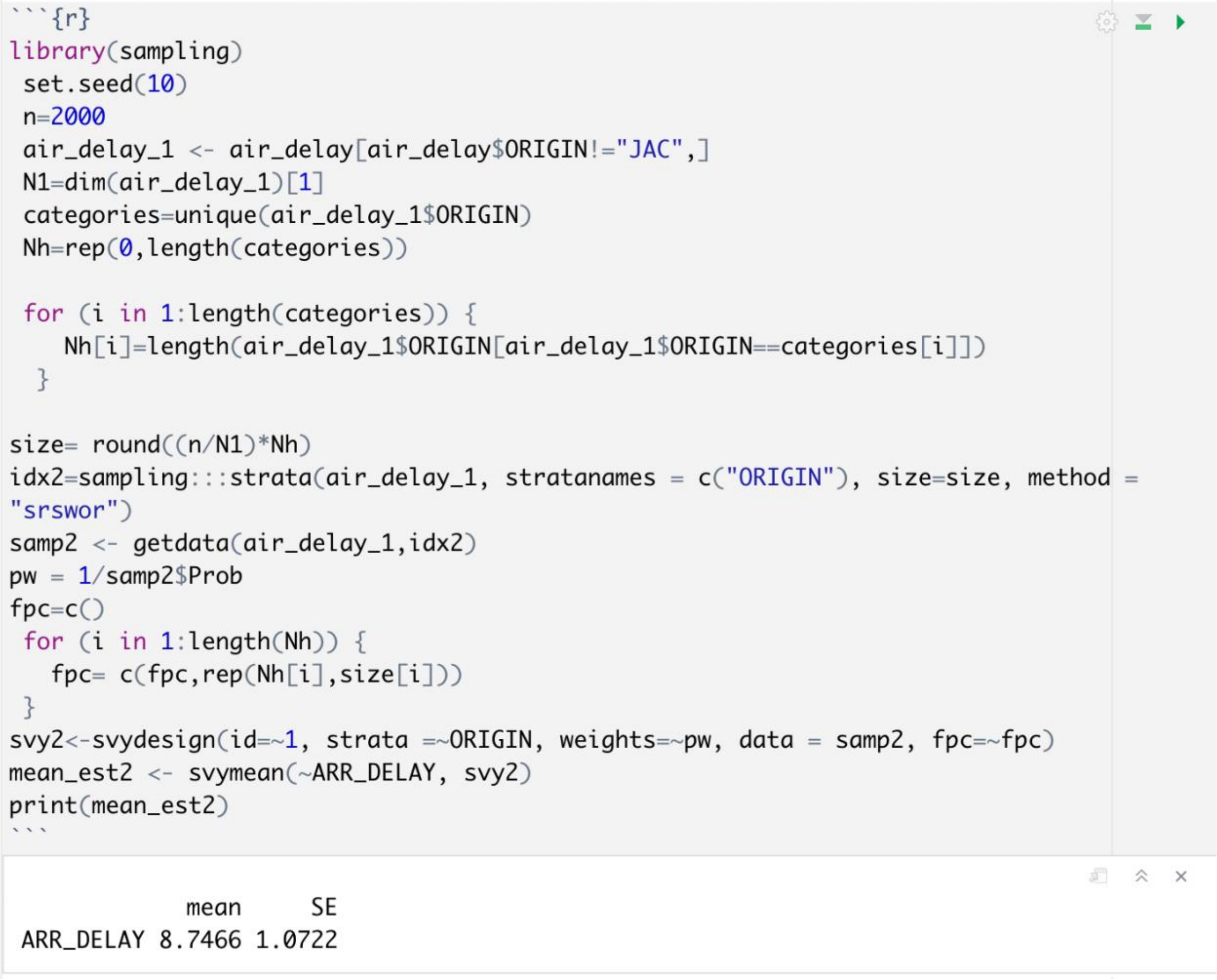
~~~
```{r}
set.seed(10)
library(survey)
library(sampling)
set.seed(10)
N= dim(air_delay)[1]

n= 2000
index1=sample(1:N ,n,replace = FALSE)

samp1 <- air_delay[index1,]
pw = rep(N/n,n)
fpc <- rep(N,n)
samp1 <- data.frame(samp1, pw=pw, fpc = fpc )

survey1<- svydesign(data = samp1,id = ~0, strata = NULL , fpc = ~fpc, weights = ~pw)
mean_est1 <- svymean(~ARR_DELAY, survey1)
print(mean_est1)
```
~~~

```
            mean     SE
ARR_DELAY 7.7325 1.2784
```

Our sample estimate is 7.73 minutes which is around 2 minutes lower than the population mean and its standard error is 1.2784 minutes.

Part 2: Stratified Sampling

Stratified sampling requires splitting up the data into ‘strata’ that contain groupings of data that exhibit similar characteristics. Since our second exploratory question seemed to indicate that the origin airport for the flight could have an impact on arrival delay times we decided to use ‘ORIGIN’ as our stratum. We believe that this stratum should give the highest sum of squares between the strata compared to other variables. For the same reasons as our SRS analysis we decided to take a sample size of 2000 rows.



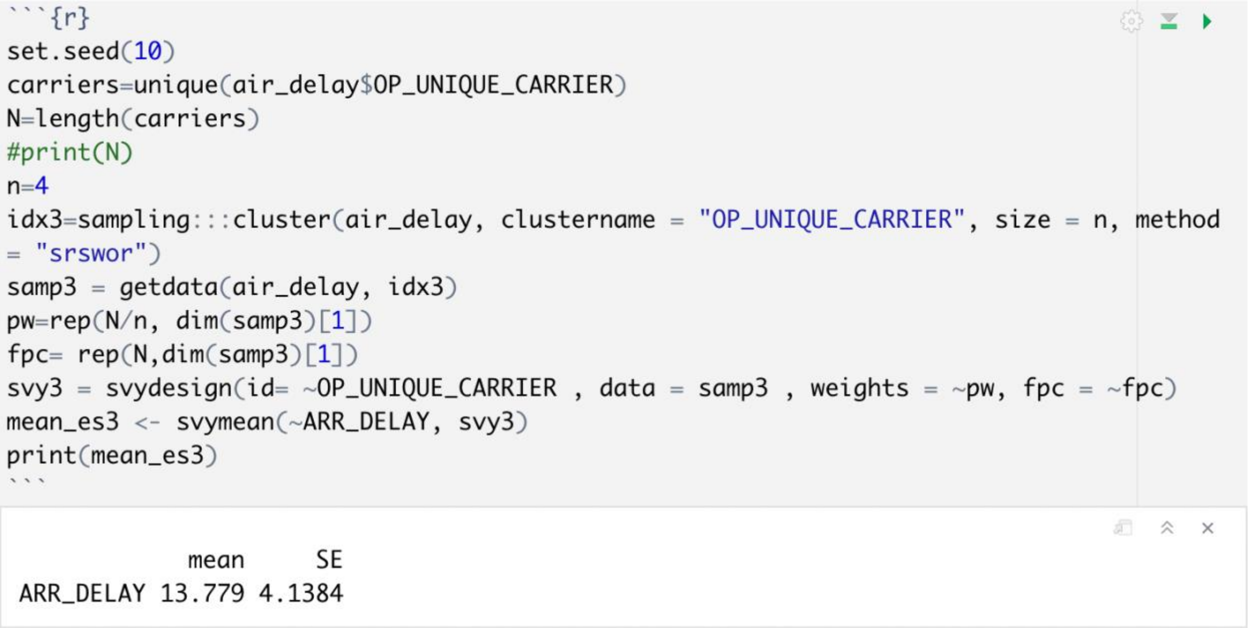
~~~
```{r}
library(sampling)
 set.seed(10)
 n=2000
 air_delay_1 <- air_delay[air_delay$ORIGIN!="JAC",]
 N1=dim(air_delay_1)[1]
 categories=unique(air_delay_1$ORIGIN)
 Nh=rep(0,length(categories))

 for (i in 1:length(categories)) {
    Nh[i]=length(air_delay_1$ORIGIN[air_delay_1$ORIGIN==categories[i]])
  }

size= round((n/N1)*Nh)
idx2=sampling:::strata(air_delay_1, stratanames = c("ORIGIN"), size=size, method =
"srswor")
samp2 <- getdata(air_delay_1,idx2)
pw = 1/samp2$Prob
fpc=c()
 for (i in 1:length(Nh)) {
   fpc= c(fpc,rep(Nh[i],size[i]))
 }
svy2<-svydesign(id=~1, strata =~ORIGIN, weights=~pw, data = samp2, fpc=~fpc)
mean_est2 <- svymean(~ARR_DELAY, svy2)
print(mean_est2)
```
~~~

```
            mean     SE
ARR_DELAY 8.7466 1.0722
```

We found the mean average delay of stratified sampling is 8.7466 minutes which is closer to the population mean than simple random sampling. The standard error of this sample is 1.0722 minutes.

Part 3: Cluster Sampling

Lastly, we will look at cluster sampling, where the data is divided into clusters that have no meaningful similarities and maximizes the sum of squares within each grouping. The criteria we decided to utilize for cluster sampling is the airline company as most airlines fly from a multitude of different airports and cities. If the main factors for arrival delays are distance, location (i.e. certain locations may be prone to bad weather) and the airport themselves (some might be less efficient than others) then airline is the best option. Out of the 9 clusters, we included 4 in the sample data.



~~~
```{r}
set.seed(10)
carriers=unique(air_delay$OP_UNIQUE_CARRIER)
N=length(carriers)
#print(N)
n=4
idx3=sampling:::cluster(air_delay, clustername = "OP_UNIQUE_CARRIER", size = n, method
= "srswor")
samp3 = getdata(air_delay, idx3)
pw=rep(N/n, dim(samp3)[1])
fpc= rep(N,dim(samp3)[1])
svy3 = svydesign(id= ~OP_UNIQUE_CARRIER , data = samp3 , weights = ~pw, fpc = ~fpc)
mean_es3 <- svymean(~ARR_DELAY, svy3)
print(mean_es3)
```
~~~

```
             mean     SE
ARR_DELAY 13.779 4.1384
```

Our sample had a mean of 13.779 minutes which is around 4 minutes greater than the population mean and it has a standard error of 4 minutes. This tells us that cluster sampling by airline is not suitable for predicting the population mean of arrival delay of flights to JFK in the month of December.

While there can be many potential reasons for this disparity, it could be because our clusters were poorly chosen. Though all of the airlines do travel across the country and from various airports our last two exploratory questions did indicate that individual carriers have differing numbers of arrival delays.

Regardless, it is clear that clustering is not the most suitable sampling method for our data. From our three samples we found stratified sampling based on airport origin to give the most accurate estimate of the population mean of arrival delay of flights to JFK in the month of December. This indicates that our stratum was practical and that the airport that flights originate from does have an impact on arrival delays.

Linear Regression

Part 1: Choosing Variables for Full Linear Regression Model

Prior to attempting linear regression modeling it is imperative that we consider the variables that should be used in the model. Our prediction model's response variable is arrival delay (ARR_DELAY) which represents the total time of the delayed flight (in minutes).

Originally, we planned to include the four major delay categories in our model (CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY and SECURITY_DELAY), however, we realized after further investigation that this would lead to invalid results. Whenever arrival delay recorded a delay (any value over 0 minutes) one of the delay variables would be equal to the arrival delay columns value. For example, if there was a 44-minute delay due to weather complications then ARR_DELAY would be equal to 44 minutes and the WEATHER_DELAY variable would also be equal to 44 minutes.

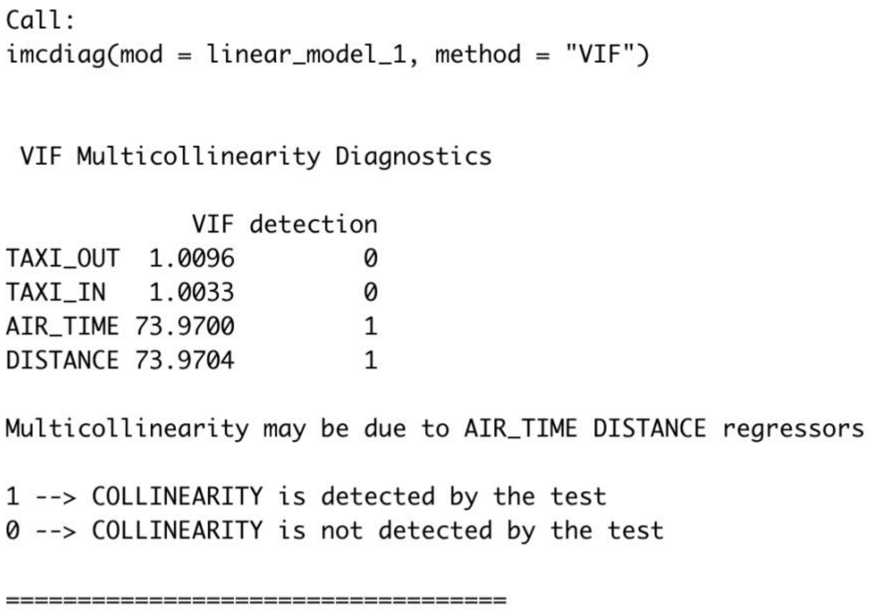
This means that predicting the delay based on these variables would lead to an adjusted r-squared value of nearly one. Consequently, DEP_DELAY, CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY and LATE_AIRCRAFT_DELAY will not be considered, and we will examine the factors outside of these variables that causes delays.

Furthermore, WHEELS OFF, WHEELS ON, DEP_TIME, ARR_TIME won't give us any information as all these variables are in time format which is not fit for linear regression analysis.

Consequently, our linear model will consist of Taxi_Out, Taxi_In, Air_Time and Distance.

Part 2: VIF Multicollinearity

The first test to conduct when conducting linear regression (once the variables are chosen) is checking for multicollinearity. When two independent variables are overly correlated with each other it can often lead to erroneous results. We conducted a Variance Inflation Factors (VIF) test which will analyze the multicollinearity of our independent variables. Multicollinearity is considered detected (note: this is not true for all VIF tests) when the VIF value of a variable is equal to one.

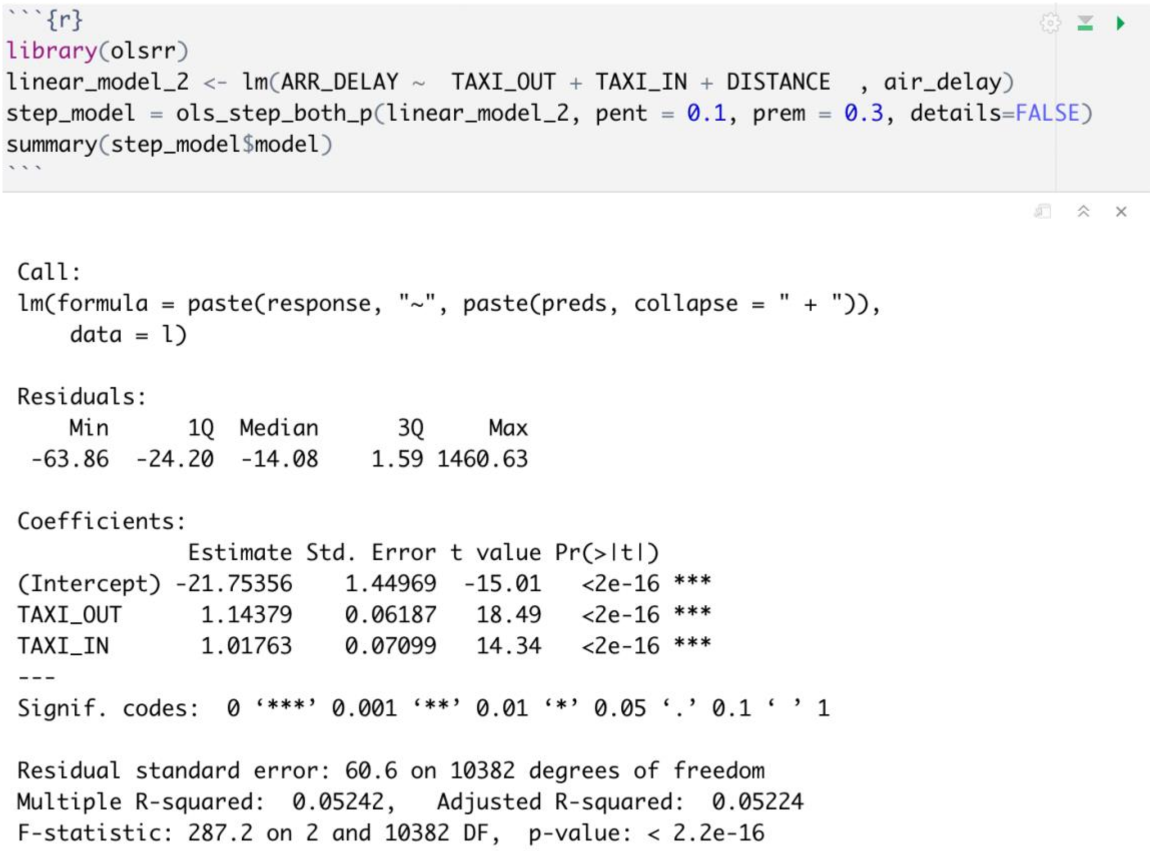


```
Call:
imcdiag(mod = linear_model_1, method = "VIF")


 VIF Multicollinearity Diagnostics

             VIF detection
TAXI_OUT  1.0096         0
TAXI_IN   1.0033         0
AIR_TIME 73.9700         1
DISTANCE 73.9704         1

Multicollinearity may be due to AIR_TIME DISTANCE regressors

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test

===================================
```

According to our results collinearity is detected between the variables Air_time and Distance, which makes a lot of sense. The longer the distance, the greater the amount of time the flight is in the air. We decided to remove Air_time rather than distance since Air_Time can vary depending on the speed of the plane while distances from airport to airport are fixed. Consequently, multicollinearity was no longer exhibited in the model.

Part 3: Model Implementation and Normality Test

Now that assumption of no multicollinearity has been established we can begin actually constructing and implementing our linear regression model. We employed step-wise regression procedure to find the grouping of variables that resulted in the best outcome (although with our number of independent variables it wasn't necessary).



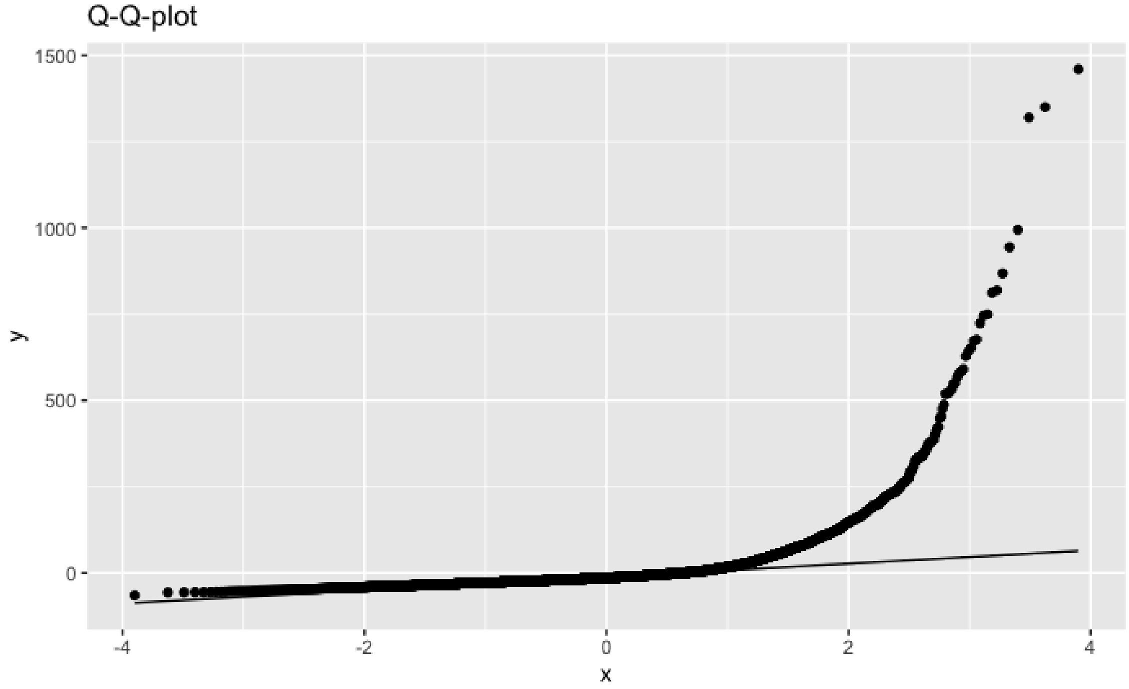
~~~
```{r}
library(olsrr)
linear_model_2 <- lm(ARR_DELAY ~  TAXI_OUT + TAXI_IN + DISTANCE  , air_delay)
step_model = ols_step_both_p(linear_model_2, pent = 0.1, prem = 0.3, details=FALSE)
summary(step_model$model)
```
~~~

```
Call:
lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
    data = l)

Residuals:
    Min      1Q  Median      3Q     Max 
 -63.86  -24.20  -14.08    1.59 1460.63 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -21.75356    1.44969  -15.01   <2e-16 ***
TAXI_OUT      1.14379    0.06187   18.49   <2e-16 ***
TAXI_IN       1.01763    0.07099   14.34   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 60.6 on 10382 degrees of freedom
Multiple R-squared:  0.05242,	Adjusted R-squared:  0.05224 
F-statistic: 287.2 on 2 and 10382 DF,  p-value: < 2.2e-16
```

The outcome of our results indicated that our linear regression model was a very poor predictor of arrival delay of flights to JFK airport. The R-squared (a statistical measure that represents the proportion of the variance for a dependent variable) value of .05224 indicates that our model predicts a measly 5 percent of the variation in the dependent variable.

Although this could be the result of our lack of independent variables (only Taxi_Out and Taxi_in ended up in the model) it seemed to indicate there was something else amiss. Thus, we checked the normality assumption which states that linear regression analysis requires that the error between observed and predicted values should be normally distributed.

This assumption can be checked by a Q-Q plot or Shapiro-wilk test. In our particular case we used the Q-Q plot due to the fact that Shapiro wilk does not work on datasets with more than 5000 rows. If the distribution is normal, the points on Q-Q plot should fall close to the diagonal reference line.



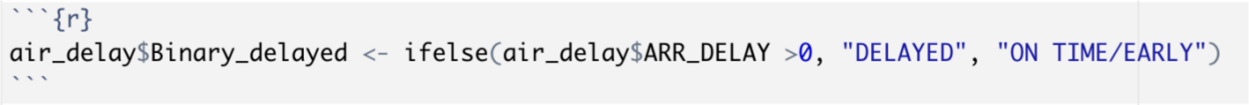
From the Q-Q plot, we can see that the majority of the points do not fall on the line. So, the normality assumption is failed by the model. It means that linear regression is not suitable at all for predicting arrival delays of flights to JFK in the month of December. The cause for this exhibited non-normality could be due to the fact that our dataset represents December, which has more flights and delays towards the end of the month due to Christmas and general holiday travel.

Consequently, we will proceed with the logistic regression and classification as well as regression tree as they are robust to non-normally distributed data.

Logistic Regression

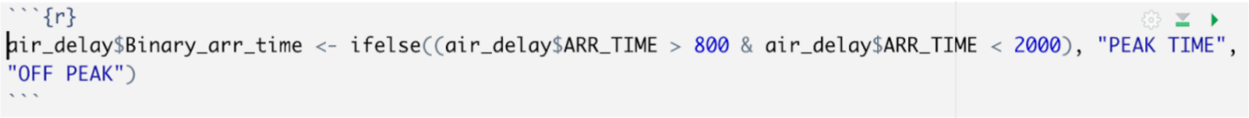
Part 1: Choosing Variables for Full Logistic Regression Model

In logistic regression analysis the dependent variable is a discrete, categorical variable. Since we still wanted to evaluate the factors that influence arrival delays at JFK airport, it was essential to create a dummy variable from ARR_DELAY. Thus, we encoded any value greater than 0 minutes as ‘DELAYED’ and any value less than 0 as ‘ON TIME/EARLY’ in the new variable ‘Binary_delayed’.



~~~
```{r}
air_delay$Binary_delayed <- ifelse(air_delay$ARR_DELAY >0, "DELAYED", "ON TIME/EARLY")
```
~~~

Now that we have our dependent variable we had to consider the independent variables that will be included. Since our previous model was limited to so few variables we decided to create another dummy variable from the column ‘ARR_Time’ which indicated the time (in 24-hour clock) that the plane arrived at JFK Airport. Any plane that landed between 800 and 2000 (8:00 am to 8:00 pm) were encoded as ‘Peak Time’ while any other time was encoded as ‘OFF PEAK’ in the new variable ‘Binary_arr_time’. This will indicate whether more flights are delayed during peak, busy hours of the day.

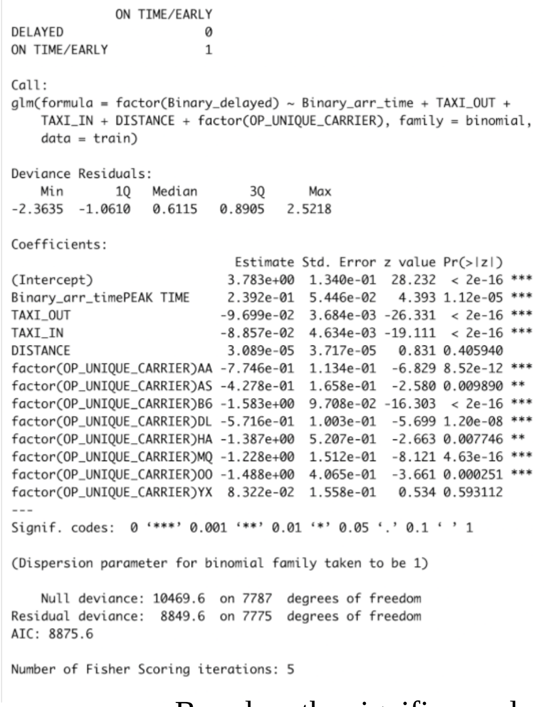


~~~
```{r}
air_delay$Binary_arr_time <- ifelse((air_delay$ARR_TIME > 800 & air_delay$ARR_TIME < 2000), "PEAK TIME",
"OFF PEAK")
```
~~~

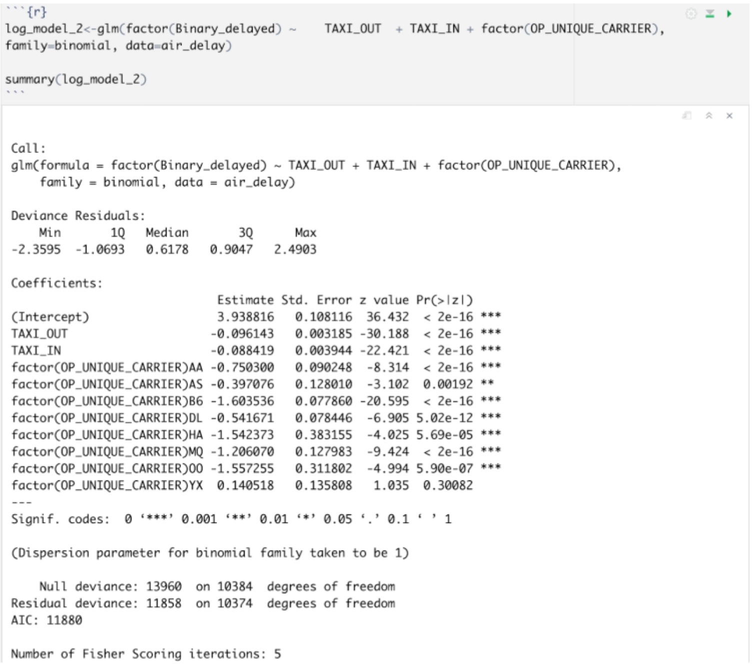
The other independent variables that were included are TAXI_OUT, TAXI_IN, DISTANCE, factor(OP_Unique_Carrier).

Part 2: Conducting Logistic Regression Analysis

Once we had our dependent and independent variables we were ready to begin the modeling process. We split 75% of the data into a training set and split the other 25% into a testing set. The trained data was employed for the first logistic regression model.



```
              ON TIME/EARLY
DELAYED                   0
ON TIME/EARLY             1

Call:
glm(formula = factor(Binary_delayed) ~ Binary_arr_time + TAXI_OUT +
    TAXI_IN + DISTANCE + factor(OP_UNIQUE_CARRIER), family = binomial,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3635  -1.0610   0.6115   0.8905   2.5218

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                3.783e+00  1.340e-01  28.232  < 2e-16 ***
Binary_arr_timePEAK TIME   2.392e-01  5.446e-02   4.393 1.12e-05 ***
TAXI_OUT                  -9.699e-02  3.684e-03 -26.331  < 2e-16 ***
TAXI_IN                   -8.857e-02  4.634e-03 -19.111  < 2e-16 ***
DISTANCE                   3.089e-05  3.717e-05   0.831 0.405940
factor(OP_UNIQUE_CARRIER)AA -7.746e-01  1.134e-01  -6.829 8.52e-12 ***
factor(OP_UNIQUE_CARRIER)AS -4.278e-01  1.658e-01  -2.580 0.009890 **
factor(OP_UNIQUE_CARRIER)B6 -1.583e+00  9.708e-02 -16.303  < 2e-16 ***
factor(OP_UNIQUE_CARRIER)DL -5.716e-01  1.003e-01  -5.699 1.20e-08 ***
factor(OP_UNIQUE_CARRIER)HA -1.387e+00  5.207e-01  -2.663 0.007746 **
factor(OP_UNIQUE_CARRIER)MQ -1.228e+00  1.512e-01  -8.121 4.63e-16 ***
factor(OP_UNIQUE_CARRIER)OO -1.488e+00  4.065e-01  -3.661 0.000251 ***
factor(OP_UNIQUE_CARRIER)YX  8.322e-02  1.558e-01   0.534 0.593112
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10469.6  on 7787  degrees of freedom
Residual deviance:  8849.6  on 7775  degrees of freedom
AIC: 8875.6

Number of Fisher Scoring iterations: 5
```



~~~
```{r}
log_model_2<-glm(factor(Binary_delayed) ~    TAXI_OUT  + TAXI_IN + factor(OP_UNIQUE_CARRIER),
family=binomial, data=air_delay)

summary(log_model_2)
```
~~~

```
Call:
glm(formula = factor(Binary_delayed) ~ TAXI_OUT + TAXI_IN + factor(OP_UNIQUE_CARRIER),
    family = binomial, data = air_delay)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3595  -1.0693   0.6178   0.9047   2.4903

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                  3.938816   0.108116  36.432  < 2e-16 ***
TAXI_OUT                    -0.096143   0.003185 -30.188  < 2e-16 ***
TAXI_IN                     -0.088419   0.003944 -22.421  < 2e-16 ***
factor(OP_UNIQUE_CARRIER)AA -0.750300   0.090248  -8.314  < 2e-16 ***
factor(OP_UNIQUE_CARRIER)AS -0.397076   0.128010  -3.102  0.00192 **
factor(OP_UNIQUE_CARRIER)B6 -1.603536   0.077860 -20.595  < 2e-16 ***
factor(OP_UNIQUE_CARRIER)DL -0.541671   0.078446  -6.905 5.02e-12 ***
factor(OP_UNIQUE_CARRIER)HA -1.542373   0.383155  -4.025 5.69e-05 ***
factor(OP_UNIQUE_CARRIER)MQ -1.206070   0.127983  -9.424  < 2e-16 ***
factor(OP_UNIQUE_CARRIER)OO -1.557255   0.311802  -4.994 5.90e-07 ***
factor(OP_UNIQUE_CARRIER)YX  0.140518   0.135808   1.035  0.30082
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13960  on 10384  degrees of freedom
Residual deviance: 11858  on 10374  degrees of freedom
AIC: 11880

Number of Fisher Scoring iterations: 5
```

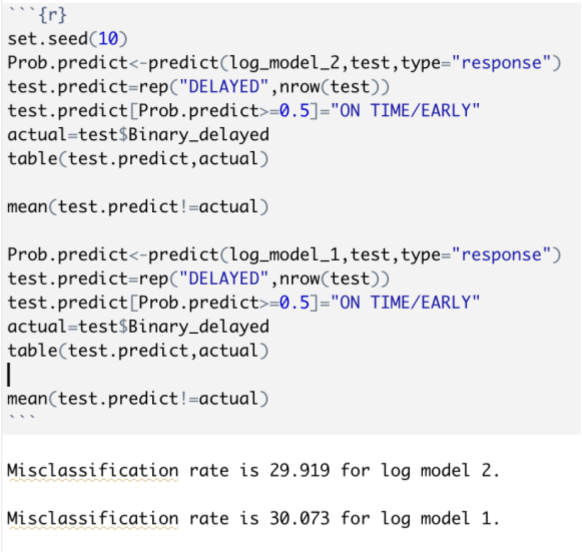
Based on the significance level of 0.05, the DISTANCE variable is insignificant. So, DISTANCE variable is removed, and the other variables are retained and used for the new logistic regression model.

Now that the new model has only significant variables we can conduct misclassification rate analysis between the old logistic regression model and the new logistic regression model.

Part 3: Old and New Model Misclassification Rate Analysis

The misclassification rate is a metric that indicates the percentage of values that we were wrongly predicted. We take the logistic regression with the testing data and use it to predict the values in the testing data. The lower the misclassification rate the more accurate the prediction.

Since the second logistic regression model contains only significant values it should be the more accurate of the two models. However, we can reaffirm this assumption by comparing the two regression models.



~~~
```{r}
set.seed(10)
Prob.predict<-predict(log_model_2,test,type="response")
test.predict=rep("DELAYED",nrow(test))
test.predict[Prob.predict>=0.5]="ON TIME/EARLY"
actual=test$Binary_delayed
table(test.predict,actual)

mean(test.predict!=actual)

Prob.predict<-predict(log_model_1,test,type="response")
test.predict=rep("DELAYED",nrow(test))
test.predict[Prob.predict>=0.5]="ON TIME/EARLY"
actual=test$Binary_delayed
table(test.predict,actual)

mean(test.predict!=actual)
```
~~~

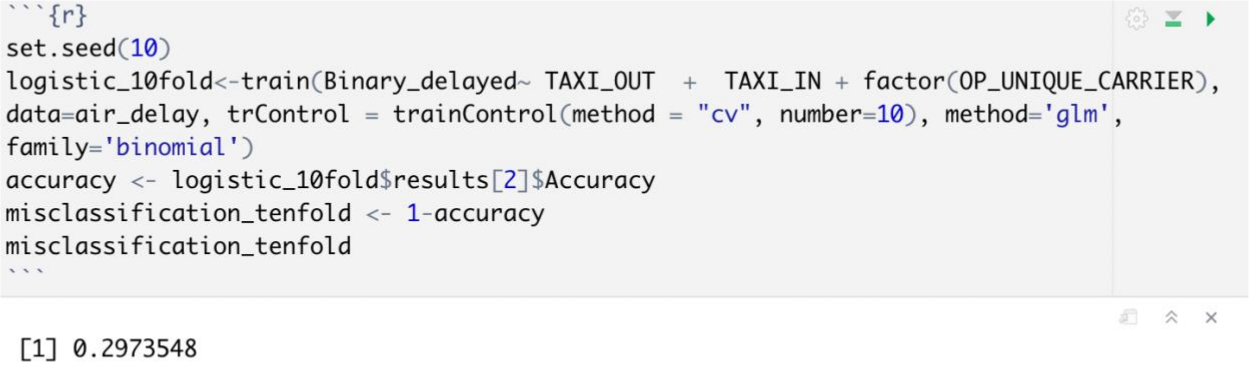
```
Misclassification rate is 29.919 for log model 2.

Misclassification rate is 30.073 for log model 1.
```

Misclassification rate for the original logistic model is 30.073 percent and the misclassification rate for the new model is 29.919 percent. As expected, the model that removed the insignificant variable has improved in performance.

Part 4: Cross Validation and Continued Misclassification Rate Analysis

After creating the newly improved logistic regression model we wanted to check what impact different resampling method would have on our misclassification rate. The three cross-validation methods we examined are k-fold cross-validation, stratified 10-fold cross validation and Leave One Out Cross Validation (LOOCV).

K-fold cross validation splits the data into a designated number of groups. Each group is used as a testing dataset once while the rest of the folds are the training dataset, this process is repeated k-times. When considering the size of the dataset, 10 folds was deemed the right amount.

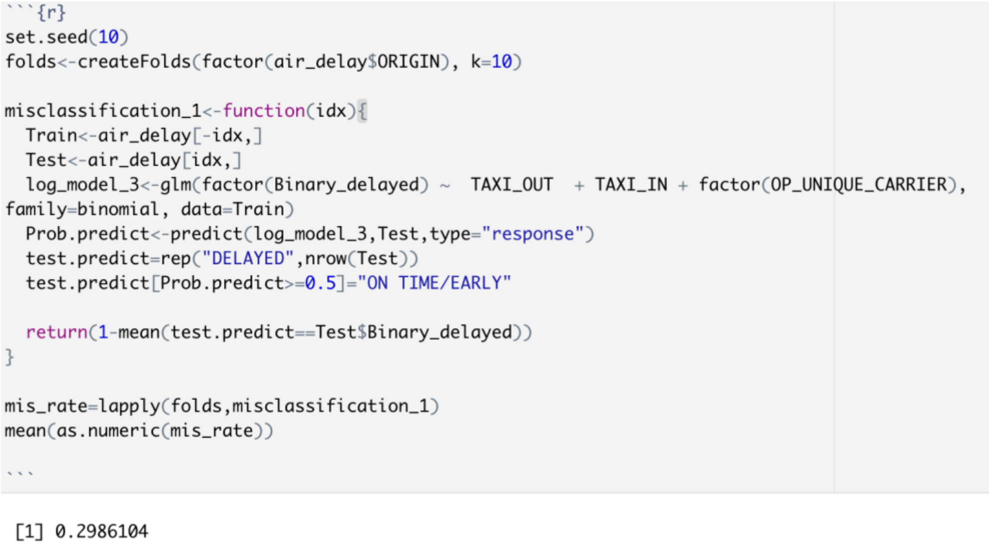


~~~
```{r}
set.seed(10)
logistic_10fold<-train(Binary_delayed~ TAXI_OUT  +  TAXI_IN + factor(OP_UNIQUE_CARRIER),
data=air_delay, trControl = trainControl(method = "cv", number=10), method='glm',
family='binomial')
accuracy <- logistic_10fold$results[2]$Accuracy
misclassification_tenfold <- 1-accuracy
misclassification_tenfold
```
~~~

```
[1] 0.2973548
```

Misclassification rate is 29.73 percent when conducting k-fold cross-validation on our logistic regression model. This is already better than the 29.913 percent that was calculated when the dataset was simply split into 75/25 percent. However, it could still be weaker than the other cross-validation methods.

The next resampling method to be tested is stratified 10-fold cross-validation. It is extremely similar to our previous approach; however, it returns folds that are stratified. Similar to how the stratified sampling method was the closest to the population dataset perhaps stratified folds would be more effective as well. The stratum chosen is origin airport due to its effectiveness during the stratified sampling process.

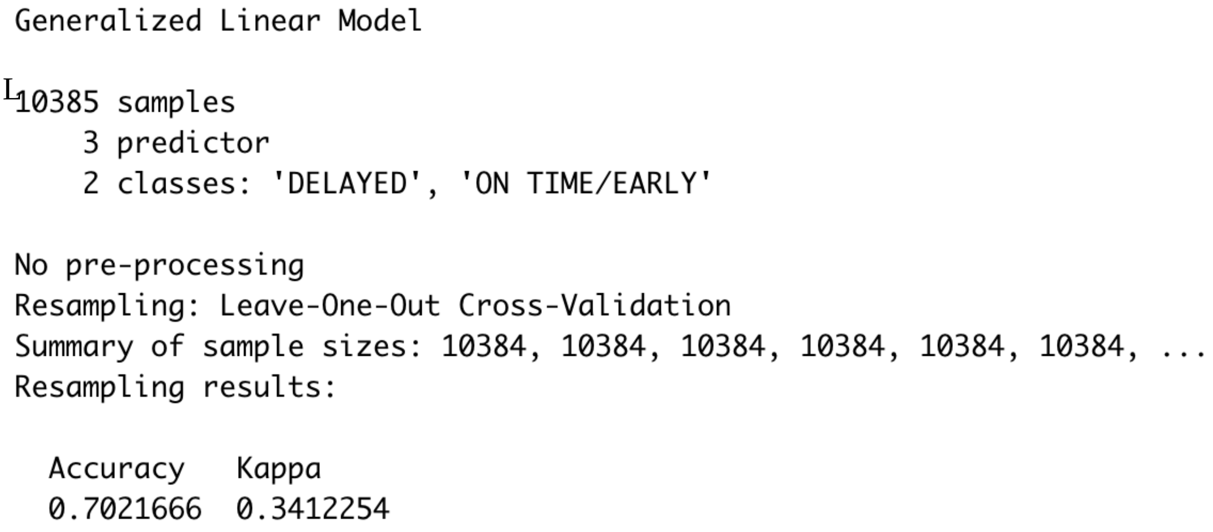


~~~
```{r}
set.seed(10)
folds<-createFolds(factor(air_delay$ORIGIN), k=10)

misclassification_1<-function(idx){
  Train<-air_delay[-idx,]
  Test<-air_delay[idx,]
  log_model_3<-glm(factor(Binary_delayed) ~  TAXI_OUT  + TAXI_IN + factor(OP_UNIQUE_CARRIER),
family=binomial, data=Train)
  Prob.predict<-predict(log_model_3,Test,type="response")
  test.predict=rep("DELAYED",nrow(Test))
  test.predict[Prob.predict>=0.5]="ON TIME/EARLY"

  return(1-mean(test.predict==Test$Binary_delayed))
}

mis_rate=lapply(folds,misclassification_1)
mean(as.numeric(mis_rate))

```
~~~

```
[1] 0.2986104
```

Misclassification rate is 29.861 percent for the stratified 10-fold cross validation, which is higher than k-fold cross validation (though not by a large amount) which indicates that it is inferior. Finally, we will attempt LOOCV.

LOOCV is the most extreme of the three methods and comes with some major computational setbacks. Rather than creating large groupings LOOCV makes a single data point a test set and the rest of the sample the training set. This process is repeated equivalent to the number of observations in the dataset. Since the data has over 10,000 observations this means that the process occurs over 10,000 times! Consequently, LOOCV is incredibly time and computationally expensive, and we experienced this firsthand as the LOOCV crashed R-studio numerous times as well as took a long time.



```
Generalized Linear Model

10385 samples
    3 predictor
    2 classes: 'DELAYED', 'ON TIME/EARLY'

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 10384, 10384, 10384, 10384, 10384, 10384, ...
Resampling results:

  Accuracy   Kappa
  0.7021666  0.3412254
```

Misclassification rate is 29.78 percent for LOOCV which is between k-fold cross validation and stratified 10-fold cross validation. Even if it performed better the computational cost makes it unrealistic for big datasets. Overall, the logistic regression model with K-fold cross validation works best with a misclassification rate of 29.73. Our findings are summarized in the table below:

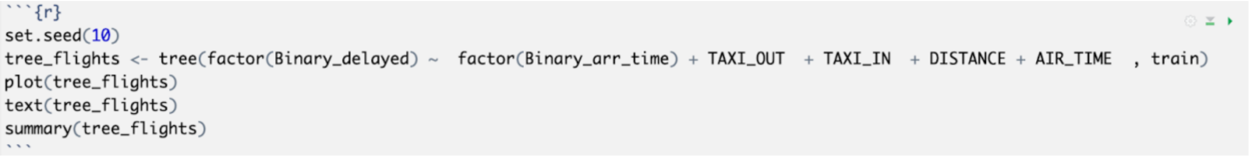
Test	Misclassification rate
K-fold cross validation	29.73
Stratified 10-fold cross validation	29.861
Leave one out cross validation	29.78

Classification Tree Model

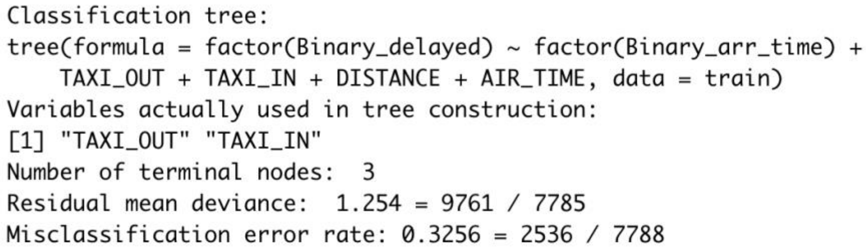
Part 1: Choosing Variables and Conducting Classification Tree Modeling

The logistic model undoubtedly performed better and was more appropriate than the linear regression model but that does not guarantee that other methods could not prove to be more accurate. Another statistical test that will be computed is the classification tree which is a mapping of binary decisions that to lead to a decision on the dependent variable. Furthermore, similar to our logistic regression model the dependent variable has to be categorical.

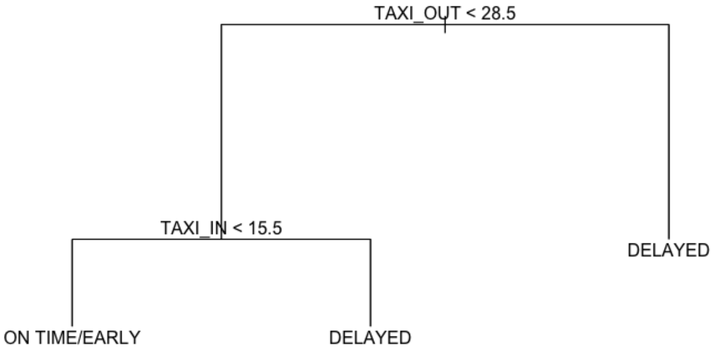
When it came to choose the variables, we kept the same model as during logistic regression, except, air_time was included since there is no assumption regarding multicollinearity for classification tree testing. Also, similar to our previous analyses we started with a basic split of the data where 75 percent was allocated to the training set and 25 percent was allocated to the test set.



~~~
```{r}
set.seed(10)
tree_flights <- tree(factor(Binary_delayed) ~  factor(Binary_arr_time) + TAXI_OUT  + TAXI_IN  + DISTANCE + AIR_TIME  , train)
plot(tree_flights)
text(tree_flights)
summary(tree_flights)
```
~~~



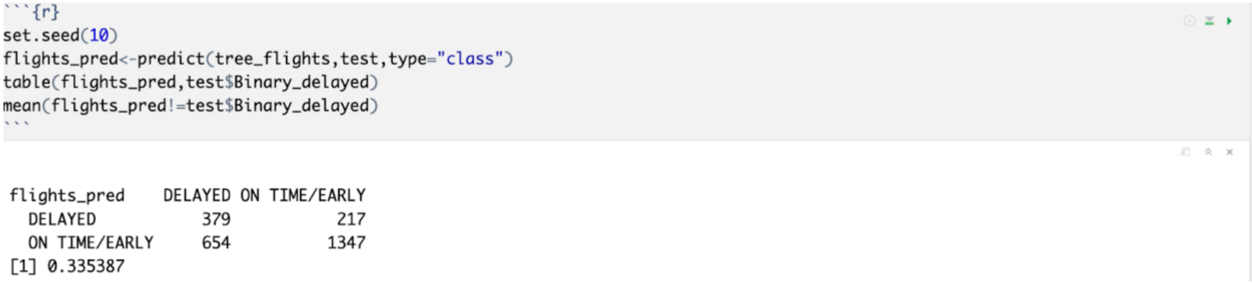
```
Classification tree:
tree(formula = factor(Binary_delayed) ~ factor(Binary_arr_time) +
    TAXI_OUT + TAXI_IN + DISTANCE + AIR_TIME, data = train)
Variables actually used in tree construction:
[1] "TAXI_OUT" "TAXI_IN"
Number of terminal nodes:  3
Residual mean deviance:  1.254 = 9761 / 7785
Misclassification error rate: 0.3256 = 2536 / 7788
```



Part 2: Analyzing Results, Misclassification Rate and Pruning

Despite the number of independent variables put into the classification tree model only taxi_in and tax_out made the final cut which indicates that they are the most significant variables when it comes to influencing airline arrival delays at JFK airport. As previously mentioned these two variables indicate how long the taxi took to take off while on track during departure and how long it took to get to the correct gate once landed. Consequently, these longer ‘taxi’ times can be significant causes of delays. Also, it’s important to note that we only have three terminal nodes.

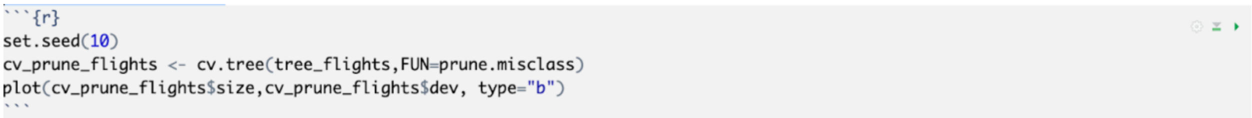
Now, it’s time to test the training data against the test set.



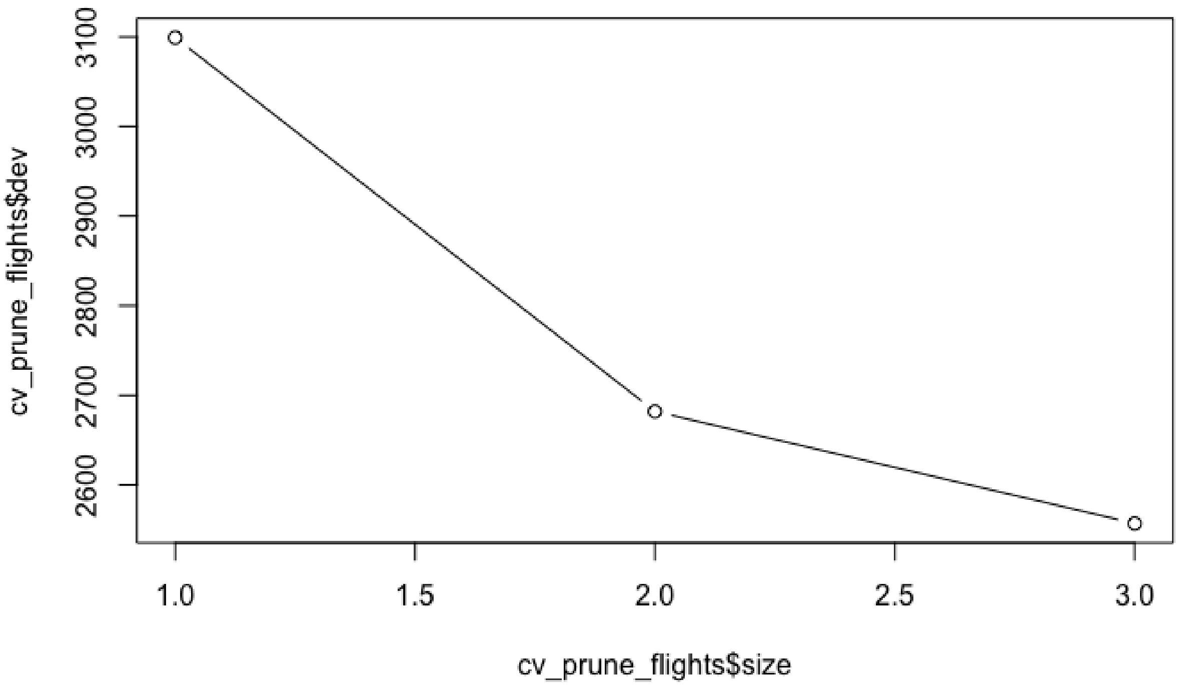
~~~
```{r}
set.seed(10)
flights_pred<-predict(tree_flights,test,type="class")
table(flights_pred,test$Binary_delayed)
mean(flights_pred!=test$Binary_delayed)
```
~~~

```
 flights_pred    DELAYED ON TIME/EARLY
   DELAYED           379           217
   ON TIME/EARLY     654          1347
 [1] 0.335387
```

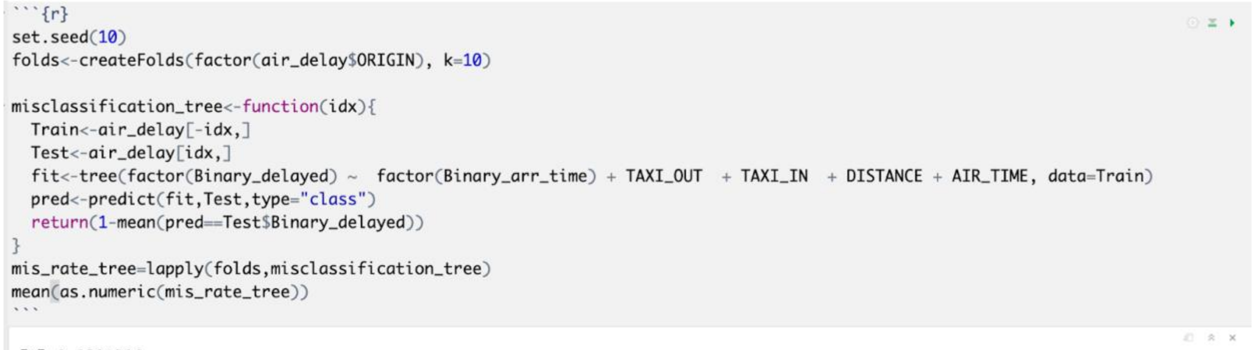
We have a misclassification rate of 33.538 percent which is higher than any of the tests done on the logistic model. Attempting to prune the tree could lead to better results, however, our tree is already so small (3 nodes) that this seems unlikely. Regardless, we will check the best number of nodes that minimizes the error through LOOCV.



~~~
```{r}
set.seed(10)
cv_prune_flights <- cv.tree(tree_flights,FUN=prune.misclass)
plot(cv_prune_flights$size,cv_prune_flights$dev, type="b")
```
~~~



We can see that when tree node size is 3, we are getting the least error. So, as expected, pruning the tree is not required. Similar to our process with logistic regression, we have applied stratified ten-fold cross validation approach based on origin to check whether it will lower the misclassification error rate or not.



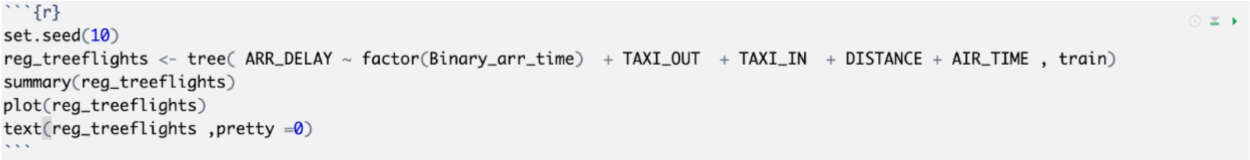
~~~
```{r}
set.seed(10)
folds<-createFolds(factor(air_delay$ORIGIN), k=10)

misclassification_tree<-function(idx){
  Train<-air_delay[-idx,]
  Test<-air_delay[idx,]
  fit<-tree(factor(Binary_delayed) ~  factor(Binary_arr_time) + TAXI_OUT  + TAXI_IN  + DISTANCE + AIR_TIME, data=Train)
  pred<-predict(fit,Test,type="class")
  return(1-mean(pred==Test$Binary_delayed))
}
mis_rate_tree=lapply(folds,misclassification_tree)
mean(as.numeric(mis_rate_tree))
```
~~~

We have a misclassification rate of 33.213 percent which is 0.3 percent better than the original misclassification rate, however, it is still much more inaccurate than our logistic regression model. Consequently, logistic regression is the better model in predicting whether a flight gets delayed to JFK in the month of December, 2019.

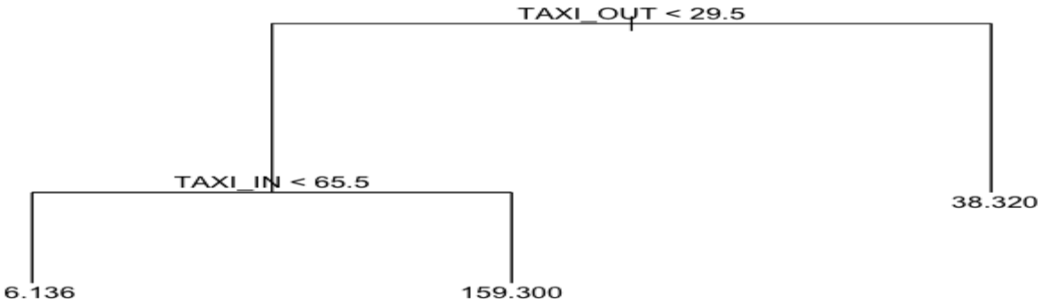
Regression Tree Model

Part 1: Choosing Variables and Conducting Regression Tree Modeling

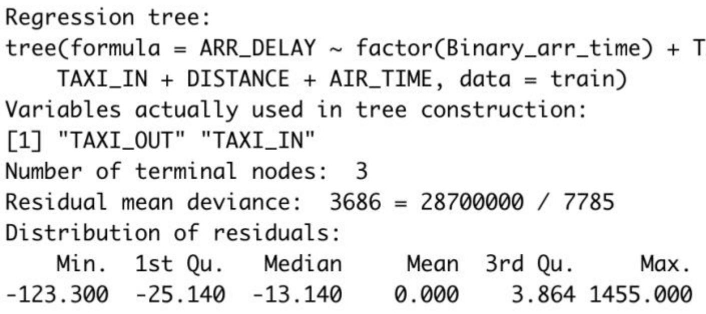
Instead of using our dummy variable which classifies whether a flight is delayed or not we want to utilize a regression tree model which requires a quantitative variable. So, we are now allowed to use the original variable ARR_DELAY without any categorical changes. The variables we utilized are exactly the same as our categorical tree model.



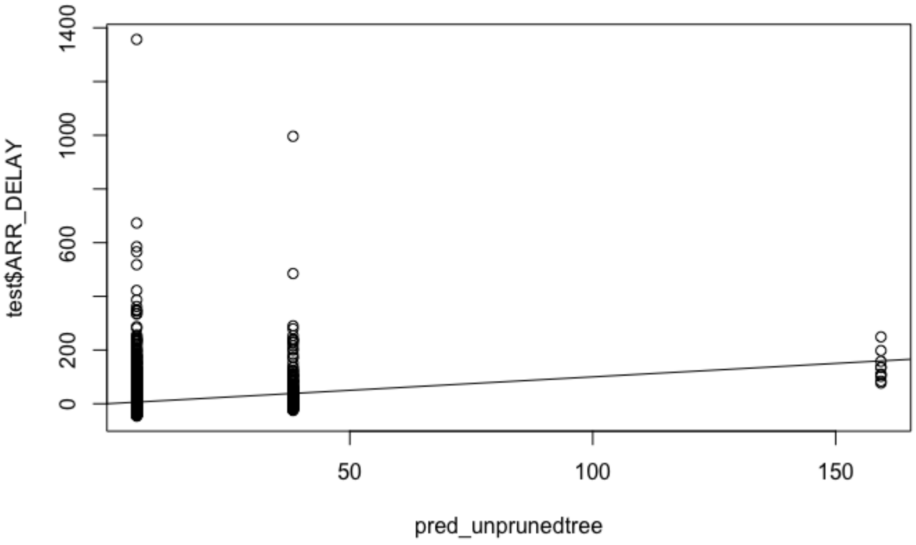
~~~
```{r}
set.seed(10)
reg_treeflights <- tree( ARR_DELAY ~ factor(Binary_arr_time)  + TAXI_OUT  + TAXI_IN  + DISTANCE + AIR_TIME , train)
summary(reg_treeflights)
plot(reg_treeflights)
text(reg_treeflights ,pretty =0)
```
~~~



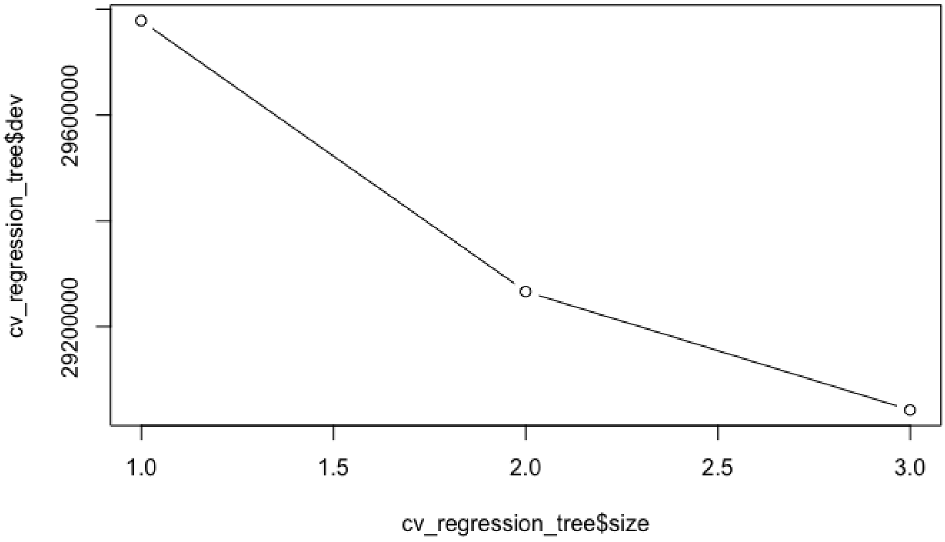
Similar to our categorical tree model only TAXI_OUT and TAXI_IN are included in the final tree and we have the same number of nodes. This time, however, the values are representing time of delay in minutes rather than just delayed or on-time. However, if you look at the results and the diagram below you'll see that our regression tree does an extremely poor job at accurately predicting the times of delays. This could be potentially due to the fact that predicting 'delayed' or 'not delayed' (logistic regression and the classification tree) is a lot more manageable than predicting specific times of delays which requires a lot more precision (and is prone to variability).



```
Regression tree:
tree(formula = ARR_DELAY ~ factor(Binary_arr_time) + TAXI_OUT +
    TAXI_IN + DISTANCE + AIR_TIME, data = train)
Variables actually used in tree construction:
[1] "TAXI_OUT" "TAXI_IN"
Number of terminal nodes:  3
Residual mean deviance:  3686 = 28700000 / 7785
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-123.300  -25.140  -13.140    0.000    3.864 1455.000
```



Typically pruning the tree could lead to clearer and more satisfactory results but that seems unlikely with the size of our current tree. Regardless, we can still evaluate whether pruning is a viable option or not.



Since the graph above indicates that 3 nodes minimize the error, there is no room for improvement with our regression tree. Furthermore, with an RMSE of 61.17271 it is evident that a regression tree model is not suitable for predicting flight delays.

Conclusion

Part 1: Limitations and Major Takeaways

Setting out on this journey our goal was to assuage the ubiquitous, encompassing stress that travelers experience by discovering the factors that affect arrival delays. However, limitations due to the variables of the dataset and characteristics of the distribution of the data made that process difficult and complicated.

While this might initially seem to limit the value of this research paper, that actually might not be the case as we did come to some valuable findings. The three significant findings (in our opinion) being that stratified sampling (by origin airport) may be the best way to sample airport data, potentially due to holiday flying patterns airport data is not normally distributed (this is certainly valuable to future researchers) and logistic modeling along with k-fold cross validation is a potentially successful testing approach.

Furthermore, approaches that favor categorical variables are more successful than quantitative variables as fluctuations in flying delays are prone to extreme variability. So, trying to find out ‘delayed’ and ‘On-time’ is a lot more reasonable than getting into specific delay amounts.

Part 2: Future Analysis

The biggest limitation of the study, and the place where it can be improved most significantly, is simply an increase in independent variables. For example, if someone can combine weather data with rain amounts, wind speeds, air quality etc. with flight information that would be extremely informative.

Another aspect to improve is actually testing the logistic regression model (or whatever best-fit model is developed) against other years and months to see the accuracy. It’s ambiguous how accurate our predictions can be from year to year and other months. It seems likely that it might be accurate predicting arrival air delays to JFK in December 2018, but as time goes on the predictability probably diminishes. Evaluating exactly how large that drop-off is by year could be very interesting and informative.

Reference List

Chokshi, N. (2020). Airline Flight Delays Got Worse in 2019. Here's a Scorecard. The New York Times. [online] 19 Feb. Available at: https://www.nytimes.com/2020/02/19/business/air-travel-delays-airlines.html.

DMR. (2019). John F. Kennedy International Airport Statistics and Facts. [online] Available at: https://expandedramblings.com/index.php/john-f-kennedy-international-airport-statistics-and-facts/ [Accessed 19 Feb. 2022].

www.bts.gov. (n.d.). 2019 Traffic Data for U.S. Airlines and Foreign Airlines U.S. Flights - Final, Full-Year | Bureau of Transportation Statistics. [online] Available at: https://www.bts.gov/newsroom/final-full-year-2019-traffic-data-us-airlines-and-foreign-airlines.