Final_Report

Tyler Celleghin (Celleghin.2)

2026-01-25

Introduction

Flight delays can be detrimental to commercial airlines and the governing bodies who operate the industry. Departure delays can be extremely detrimental to operational efficiency, cost a significant amount of money and resources and be damaging to the reputations of the airlines and airports. If the Port Authority could predict which flights may be at risk for delay, it is possible to allocate resources to those flights to minimize or eliminate delays. The primary goal of this analysis is to provide insight into information which may indicate an upcoming departure delay and provide recommendations to the Port Authority of New York and New Jersey for improvements in their operations. The Greater New York City area is serviced by three major international airports, JFK, LGA, and EWR. Data which included flights from all of these airports was provided to us. We performed a full univariate and bivariate exploration of the data before expanding our analysis into more complex statistical models. Linear models were prioritized for their ease of interpretation and PCR, PLS and trees were attempted for use in prediction.

Data Cleaning

To begin our analysis, we performed a preliminary examination of the data provided to us and performed cleaning on said data to ensure it was in a suitable format for our analysis. We were provided with five separate but related datasets relating to flights in the Greater New York City Area in 2013:

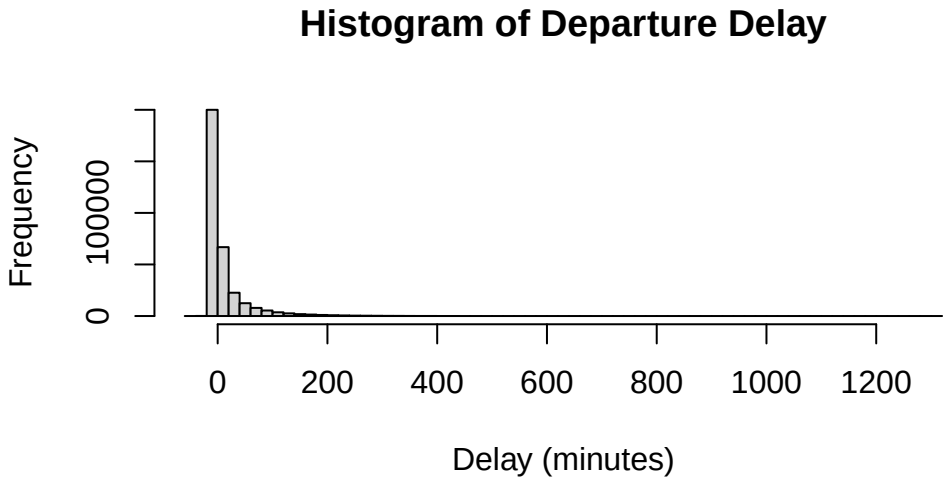
flights - containing the records of each individual flight from the three NYC airports in 2013
weather - containing records of various weather conditions in the NYC area in 2013
planes - containing attributes relating to each individual plane involved in 2013 flights
airlines - containing names of airlines corresponding to each carrier code
airports - containing location information for each airport in the dataset

For the purposes of this analysis, we focused on three datasets: flights, weather, and planes as those likely to contain insights into flight delays. Our data cleaning process began with the merging of these three datasets into one that could be used for analysis. First, flights and planes were merged to include all the plane attributes in each individual flight record, joining flights and planes by tail number. Then, we merged this dataset with weather to include ambient weather information with each individual flight by joining the flights/planes dataset and weather by time_hour and origin. This ensured that each flight would now include accompanying information on the attributes of the plane involved and the weather at the time of departure at the airport involved. At each step, duplicated columns created from merging were removed.

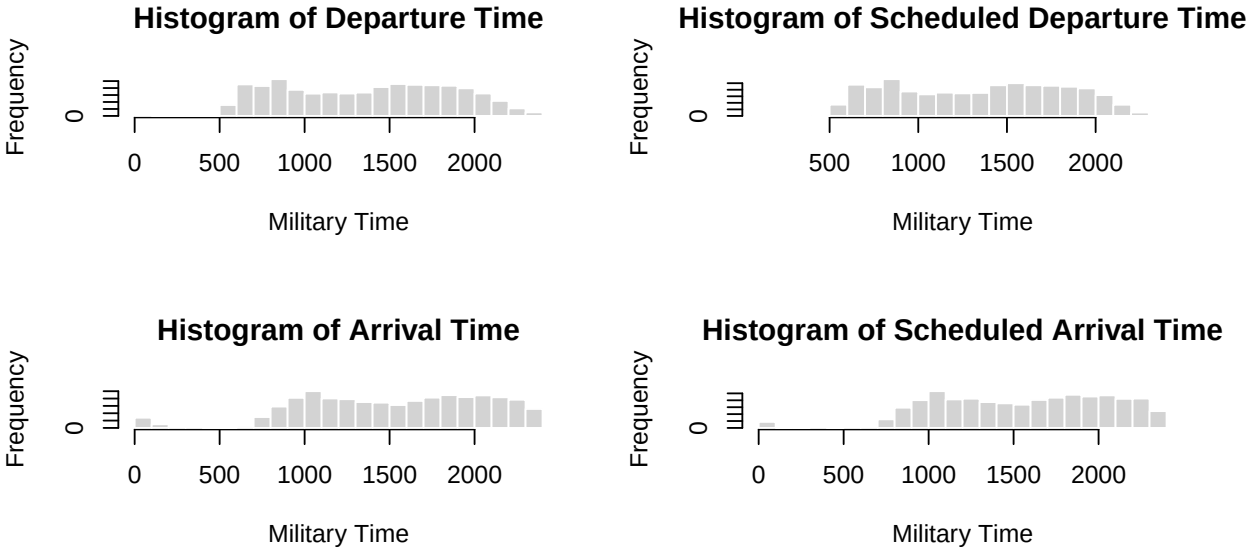
Following the creation of the merged dataset. The last step for data cleaning was examination of the data to remove superfluous information and remove null values. The day, time_hour, and hour columns were removed due to repeating time information included in other variables. flight and tailnum were removed as they were simply index variables with little analytical insight. Finally, wind_gust was removed due to most of the data being missing, and null values were omitted from the dataset at large. All of this cleaning reduced the initial number of observations from 336,776 to 60,448.

Exploratory Data Analysis

After data cleaning, the first step in Exploratory Data Analysis (EDA) is an examination of univariate distributions of the variables in our dataset. The first variable examined was our response variable, dep_delay. We first examined the summary statistics of dep_delay, finding that the mean departure delay for 2013 flights was 12.79 minutes, and the median delay -1 minutes. Departure delays in our dataset ranged from -23 to 747 minutes. We also examined a plot of the univariate distribution of departure delays, as shown below. Dep_delay exhibited a highly right skewed distribution with most delays around zero, and a small number of very high delay times. This suggests a log transformation would be appropriate, but the presence of negative values (flights leaving early) complicated this process and we ultimately decided against it.



Examination of the distributions of the time related variables relating to flights (departure time, arrival time, scheduled departure time, scheduled arrival time) reveal similar patterns with each. All four exhibit bimodal distributions with more flights centered around 1000 (10:00 AM) and 1600 (4:00 PM). Logically, the arrival times are shifted slightly later compared to departure times.



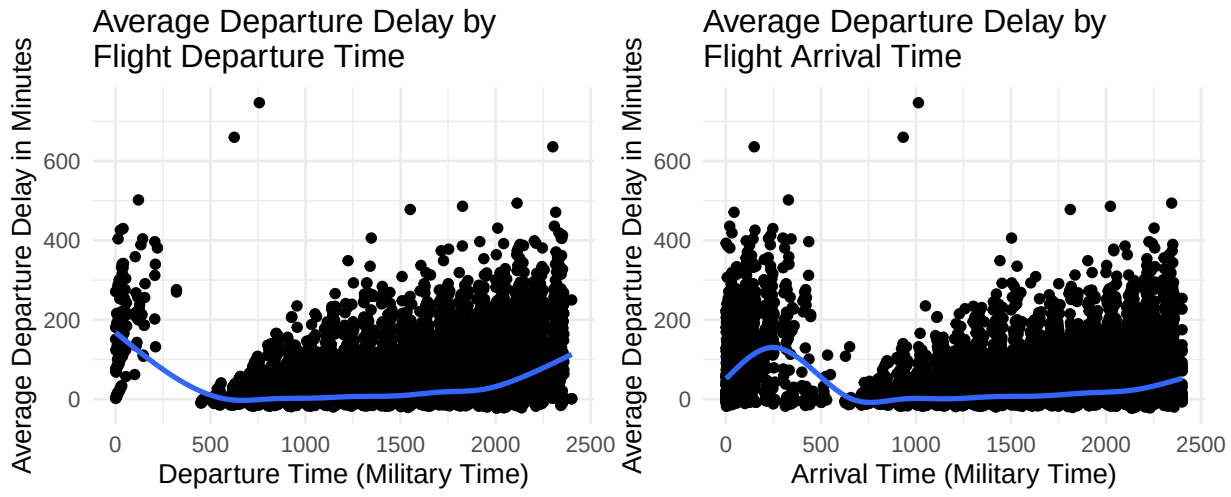
Looking at the other variables relating to flights, arrival delay exhibited a right-skewed distribution similar to departure delays. Carrier and Destination exhibited highly varying numbers of flights depending on airline and destination airport. Origin showed a relatively equal number of flights in each of the three NYC area airports. Airtime and Distance showed very similar relationships with both being somewhat right skewed but bimodal, with a concentration of shorter flights and then a second peak of long flights, followed by a handful of very high outliers (Flights to Hawaii).

Looking at the variables relating to aircraft attributes, year manufactured exhibits a roughly normal distribution centered around 2000, but with a number of older outliers. Looking at engine type and number of engines, nearly 100% of flights involved a fixed-wing multi-engine aircraft and two engines. Flights were highly concentrated into just a handful of manufacturers. The number of seats also exhibited a bimodal distribution, with most planes having 100 to 200 seats, but a much smaller peak around 350 as well.

We also examined the univariate distribution for the weather related variables. Temperature exhibited a wide distribution from about 0 to 100, with a concentration in the range of 30 to 85 and a notable peak around 40 degrees. Dewpoint showed a very similar pattern, but with a concentration between about 10 and 70, and the peak at about 20. Humidity had a somewhat right skewed distribution with a peak at about 40 percent, while wind direction exhibited a highly left skewed distribution with a large peak at about 310 degrees. Pressure exhibited a roughly normal distribution centered at about 1015 millibars. The vast majority of observations had a precipitation of zero, with the rest of flights that did observe precipitation ranging to 0.53 inches of precipitation. Similarly, visibility was 10 (the maximum) in the vast majority of flights as well, and the distribution was otherwise right skewed with visibility concentrated in the higher levels.

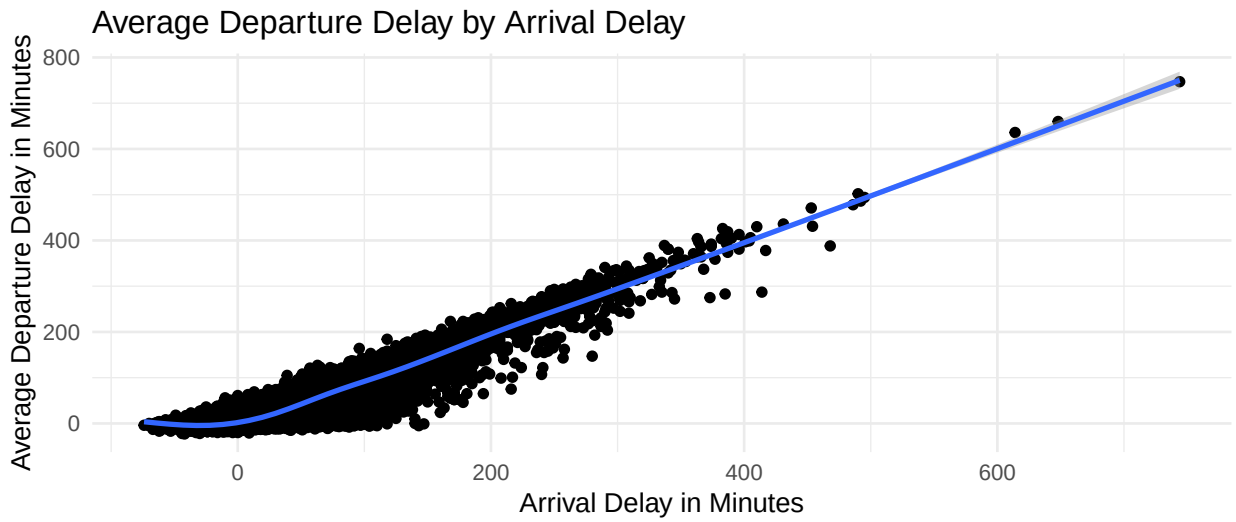
Bivariate Continuous Variables

To investigate the bivariate relationship between departure delay and continuous variables in our dataset, we created a series of scatterplots for dep_delay versus each continuous variable, including a smoothed average line for dep_delay. We began with examination of the schedule related variables departure time, arrival time, scheduled departure time, and scheduled arrival time. Looking at average departure delay versus departure time, we found that there were significantly elevated departure delays late at night and in the early morning. This was paired with corresponding elevated departure delays by arrival time, logically a few hours later. For the scheduled departure and arrival time variables, the same patterns were observed as their actual counterparts, only with a significantly weaker pattern.



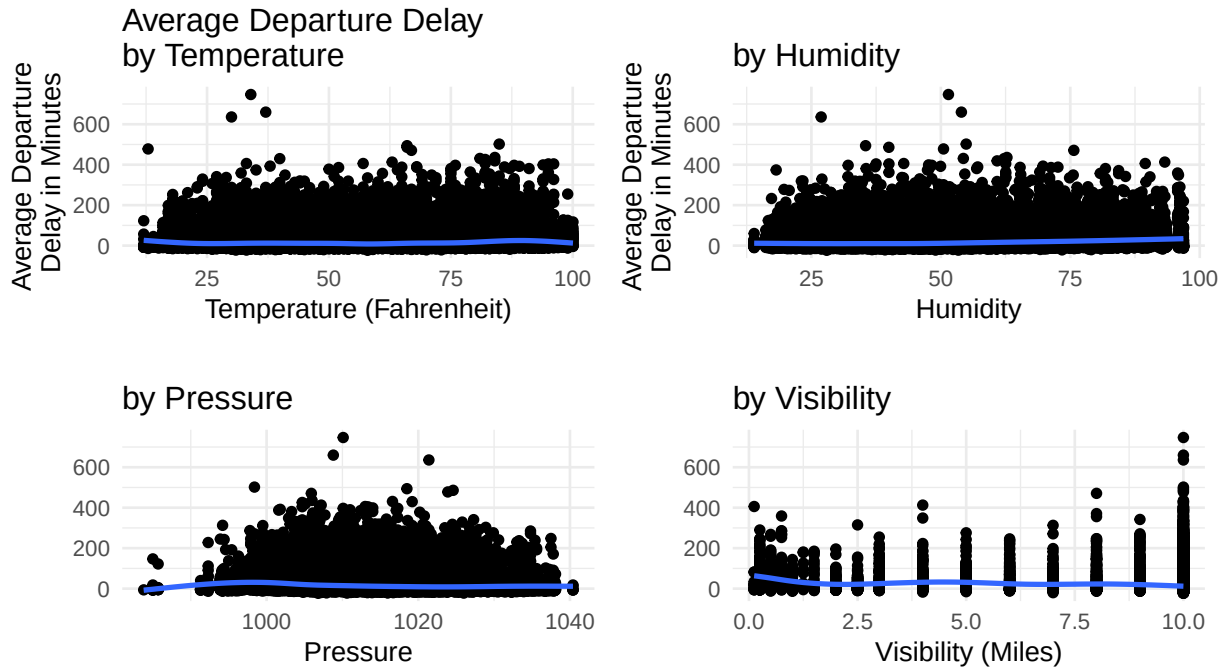
Looking at average departure delay by arrival delay, we found an incredibly strong association between the two. This logically makes sense, as arrival delays would directly depend on departure delays. However, this

has little practical benefit for providing insights into how to help prevent departure delays, since we would not be able to know arrival delay as a characteristic of a flight in advance.



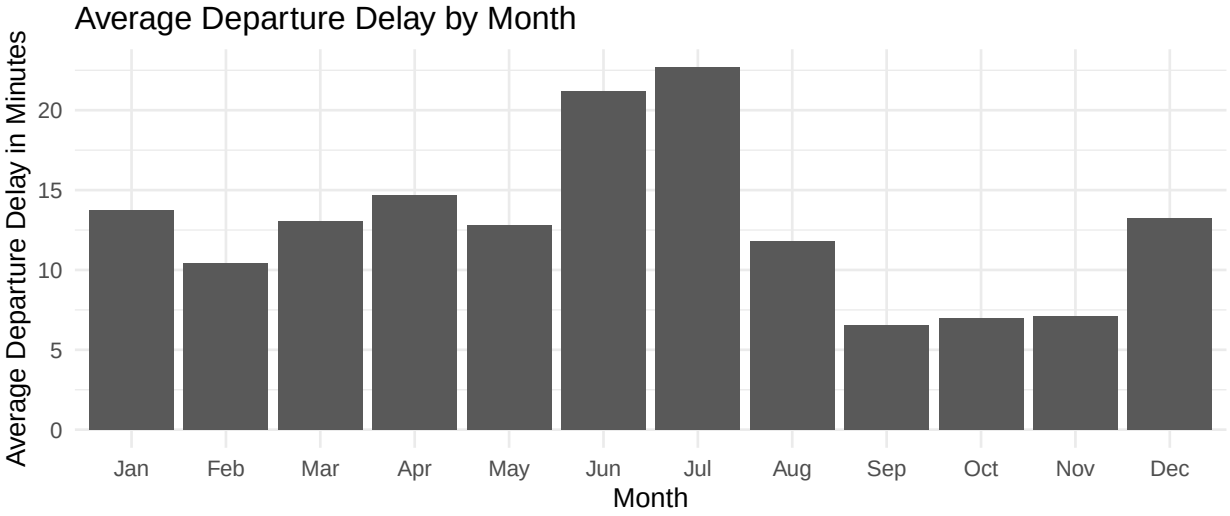
Departure delays did not appear to meaningfully differ based on flight airtime, distance, or aircraft seat number. The same was true for year constructed except for a small rise in departure delays in planes constructed before approximately 1975, although data was scarce in this range.

Looking at notable weather variables, Average Departure delays appeared slightly elevated in cold temperatures, less than 25 degrees Fahrenheit, and hot temperatures, greater than 80 degrees. Dewpoint exhibited a similar pattern to temperature. We observed a slight positive relationship between humidity and average departure delay. There also appeared to be somewhat elevated departure delays when pressure was low. Lastly, when visibility was very low, there were also somewhat elevated departure delays. The remaining variables, wind speed, wind direction, and precipitation, did not appear to have any meaningful effects on departure delay times.

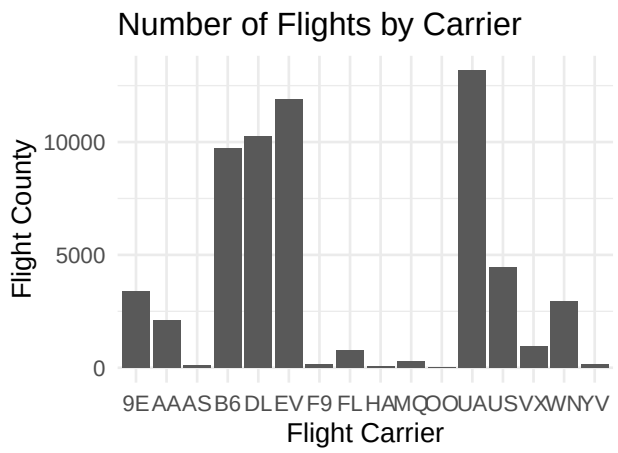


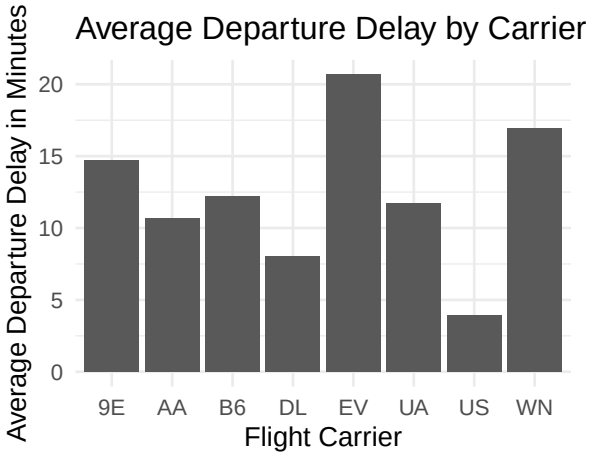
Bivariate Categorical Variables

To investigate the bivariate relationship between departure delay and the categorical variables in our dataset, we created a series of bar charts for each categorical variable displaying the average departure delay by category. Looking at average departure delay by month, there are notably peaks in June and July as well as January and February. Departure delays are also notably lower in the fall.



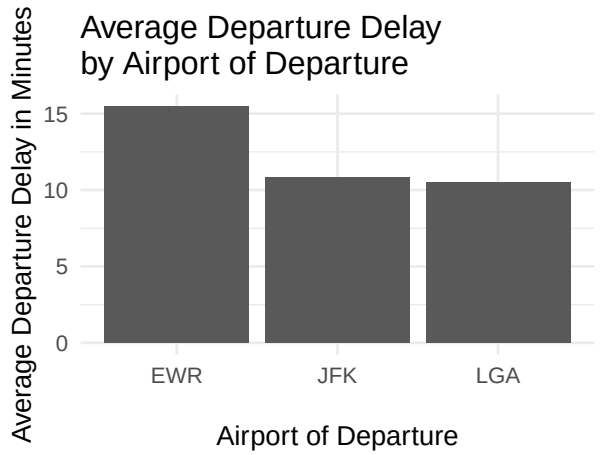
In examining the relationship between departure delay and carrier, it is important to note that the number of observations per carrier varies considerably. Many of the individual carriers have a very low number of observations. In these cases, we cannot make reasonable conclusions about the apparent variation in their average departure delays due to low sample sizes, which can lead to highly varying statistics because of just random chance. As such, in these cases of categorical variables with certain categories having low sample size, we filtered out categories that were involved in less than 1000 flights. Looking at the 8 carriers that had at least 1000 flights in 2013, we found that there was considerable variation in departure delays for a number of carriers. Most notably, ExpressJet (EV) and Southwest (WN) airlines had significantly elevated average delay times, while US Airways (US) had a very low average delay time.

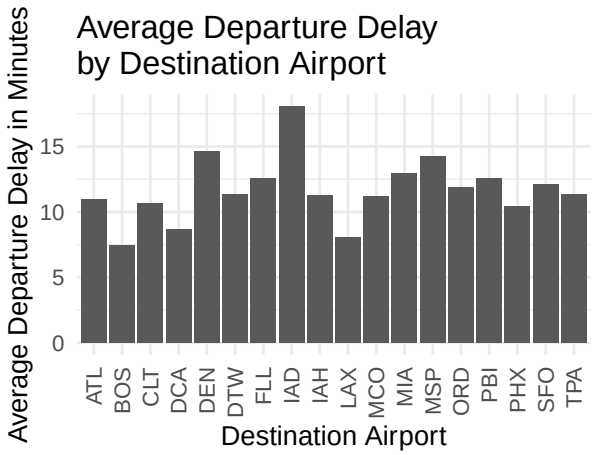




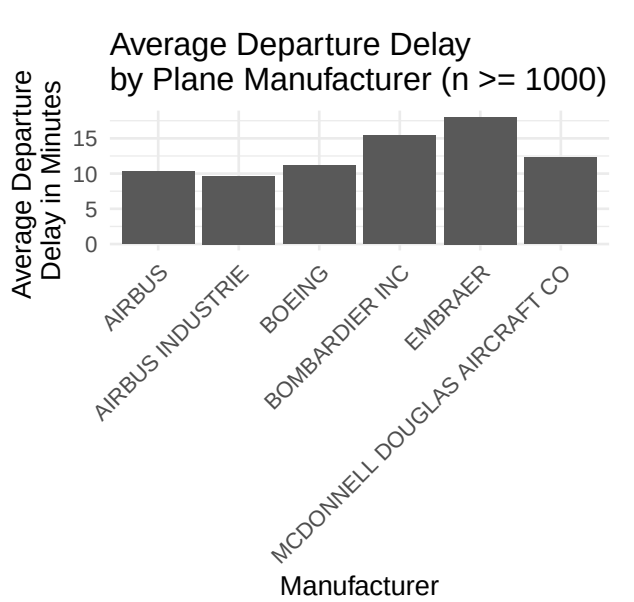
Next, we examined departure delays by airport of origin and destination. For airport of origin, that is, which of the three NYC area airports the flight departed from, we found elevated average departure delays for Newark Liberty International Airport (EWR), with an average of 15.5 minutes. Looking at John F. Kennedy and LaGuardia International Airports, we found that they both exhibited similar average departure delays at 10.9 and 10.5 respectively. For destination airports, it was necessary to filter out 86 of

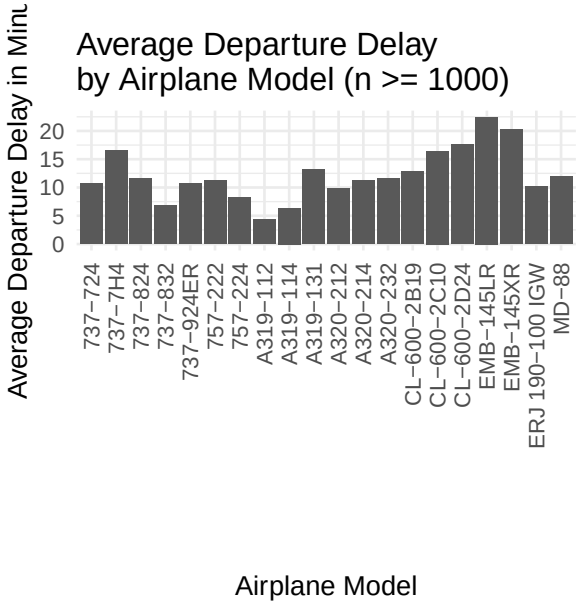
the 104 destination airports with less than 1000 observations to avoid small sample sizes. Examining the average delay for the remaining 18 destinations, we see that there does seem to be some notable variation in departure delay depending on destination airport.





We also examined how average departure delay was influenced by aircraft manufacturer and aircraft model. Again, both of these categories required filtering to only those with over 1000 observations to avoid small sample sizes. Looking at the resulting plots, we found that there seemed to be some variation in sample size by manufacturer. Notably, EMRAER exhibited higher than average departure delay times. There also appeared to be some significant variation in average departure delays by model, with some aircraft models having quite high average delays and some very low.





Lastly, there were three categorical variables where examination of average departure delay by category was deemed inappropriate or did not yield insights. First, for aircraft type, 99.39% of flights involved a fixed-wing multi-engine aircraft, so the other two categories have too little data to yield meaningful insights. The same is true for the number of engines variable, where 99.38% of flights involved a plane with two engines. Lastly, for the engine type variable, 99.32% of flights involved aircraft with turbo-fan or turbo-jet engines as opposed to the other four categories. Comparing these two categories, the average departure delay for turbo-fan planes and turbo-jet planes were 13.3 and 10.1 respectively, relatively similar to each other and the overall average departure delay of 12.79.

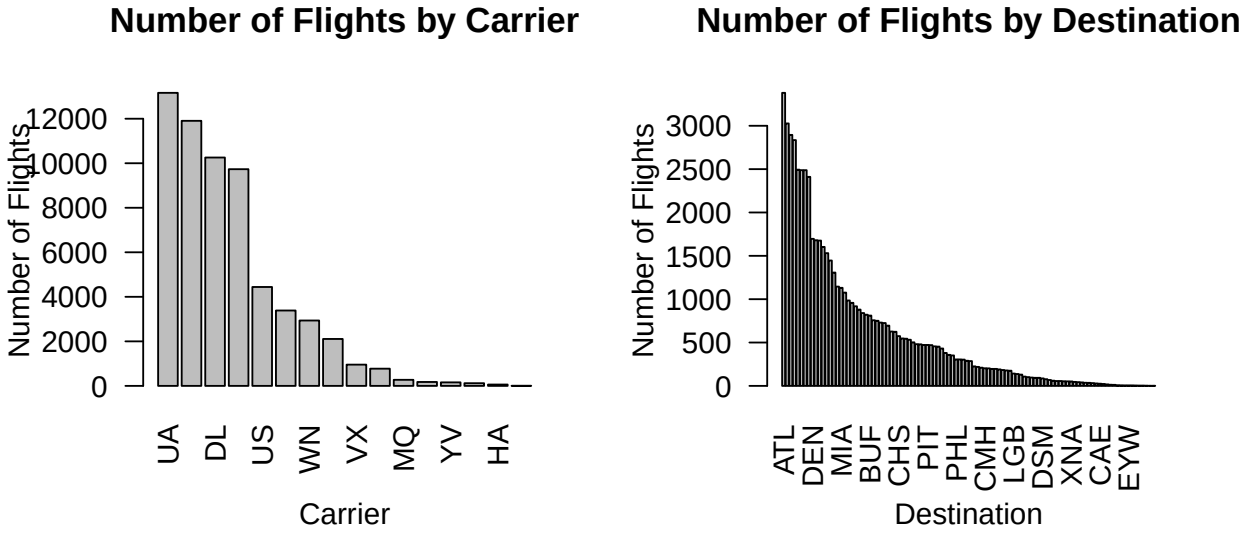
Model Fitting

The next step in our analysis was to use what we learned about the bivariate relationships of the various predictors with departure delays to explore a variety of different statistical models for predicting dep_delay from the rest of the variables in the dataset. For this project, we explored a simple linear model, Ridge and Lasso models, Principle Components Analysis (PCA) and Partial Least Squares (PLS) models, and tree-based models for predicting dep_delays. The purpose of exploring these models was not necessarily to build a highly accurate prediction model, but rather explore how they could provide insights into which variables most affect departure delays and how they do so. To prepare for model selection, we split the dataset into a training and testing set with an 80/20 split.

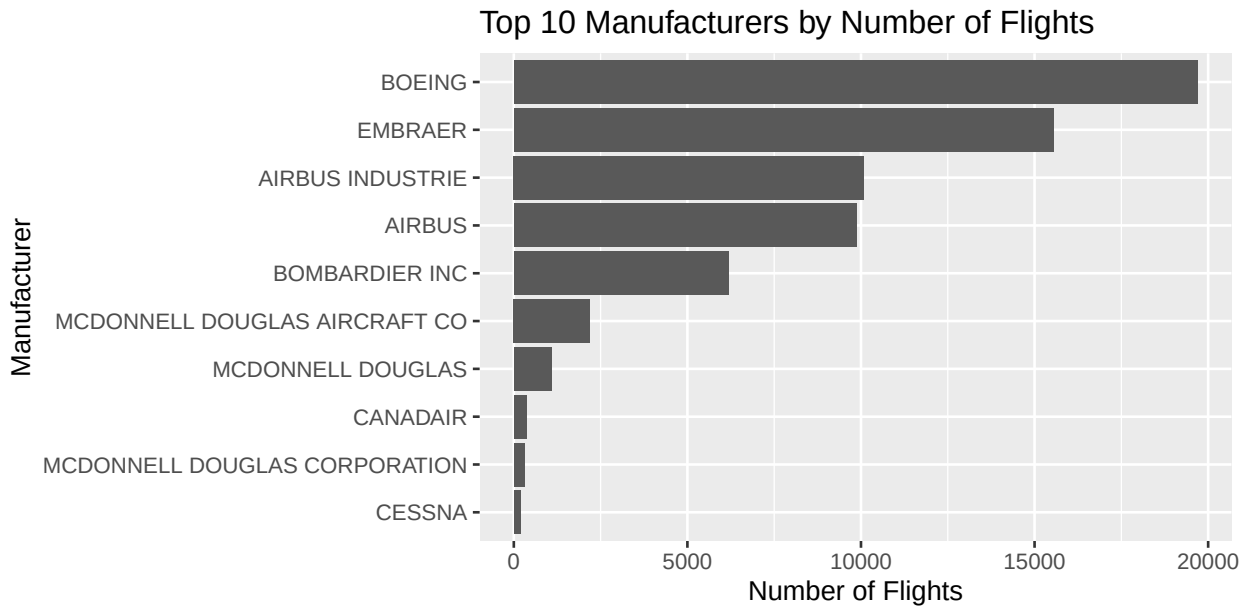
Linear

To get an initial overview of potential influences on departure delay, we made a multiple linear regression with a response variable of departure delay. The predictors included were month, scheduled departure time, scheduled arrival time, carrier, origin, destination, distance, year constructed, manufacturer, model, seats, temperature, dewpoint, humidity, wind direction, wind speed, precipitation, pressure, and visibility. The variables we chose to omit due to correlation were departure time, arrival time, scheduled arrival time, arrival delay day, and air time. The variables omitted due to irrelevance or lack of variance in data were tail number, plane type, plane model, number of engines, and engine type.

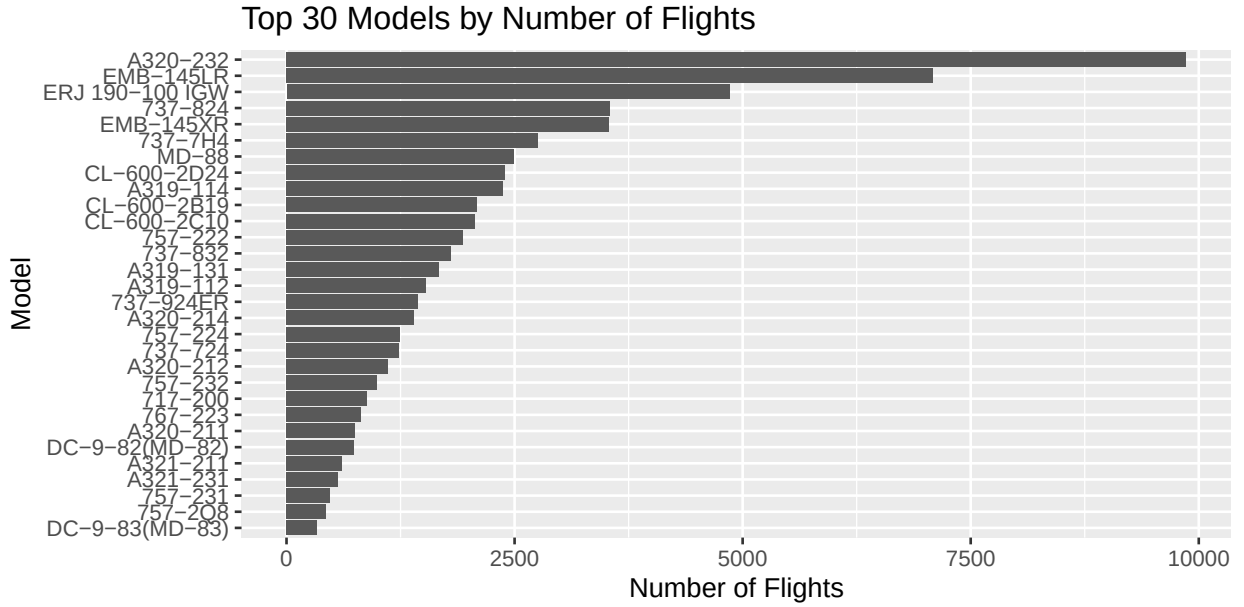
The categorical variables included were month, carrier, origin, destination, plane type, manufacturer, and model. Some of these variables had too many categorical outcomes so they had to be restricted. The values that had the highest frequency were kept as single values, the rest were put into a single categorical value called other in their specific variables.



For carrier, the top 4 values were included. For destination, the top 8 destinations were included.



For manufacturer, the top 5 manufacturers were included.



For model, the top 11 models were included.

After sorting the data, the top 10 influencing variables could be listed:

Predictor <chr>	Effect <dbl>	AbsEffect <dbl>
precip	-23.528192	23.528192
model_grpERJ 190-100 IGW	-14.194708	14.194708
model_grpCL-600-2D24	14.133164	14.133164
manu_grpBOMBARDIER INC	-13.550053	13.550053
model_grpCL-600-2C10	11.194631	11.194631
model_grpA320-232	-10.554998	10.554998
month9	-9.655858	9.655858
carrier_grpDL	-9.358663	9.358663
model_grp737-824	-8.592095	8.592095
month10	-7.785710	7.785710

From this model, it seems that precipitation has a large negative effect on flight delays. This shows an obvious problem with the precipitation variable, and upon further inspection using a frequency table we can see there is very limited data on flights that had precipitation.

```
##
##     0  0.01  0.02  0.03  0.04  0.05  0.06  0.07  0.08  0.09   0.1  0.11  0.12
## 58813   404   380   293    78    47    82    48    43     1    47    48    20
##  0.13  0.14  0.15  0.16  0.17  0.19   0.2  0.23  0.28  0.33  0.34  0.53
##     4     8    15     3    31    18     3    14     9    22    13     4
```

After precipitation, it can be seen that specific models, manufacturers, months, and destinations have fairly large effects on departure delay. While not uninteresting, such specific insights make it difficult to make actionable recommendations. For this reason emphasis was placed on other models for our recommendations.

Ridge / Lasso

Purpose and modeling context

The motivation behind the ridge and lasso modeling was to evaluate how well departure delays can be predicted using the information available prior to the flight departure. This modeling allows us to examine a realistic scenario where decisions for scheduling flights must happen without any real time operational data. This scenario arises when flight schedules are initially created, which means any post-departure operational disruptions or other same-day flight delays are completely unknown. To enforce this type of constraint, the specific variables removed from the data prior to analysis were arrival delay, arrival time, air time, and departure time. The goal of this was not to figure out a way to calculate the most exact departure delay estimates possible, but rather to assess the early predictions or risk of delay and determine whether the flight should be flagged as exhibiting signals of risk of delay.

Statistical Methodology

In order to model the departure delays under these restrictions, regularized linear regression methods were used, specifically ridge regression and lasso regression. These methods were chosen for multiple reasons, but primarily due to their ability to handle correlated predictors and perform variable selection. An ordinary design matrix for both models was created with the pre-departure variables we were given in the data set, and cross-validation was used to select lambda, the regularization parameter. There were two different values of lambda to be examined and considered: λ_{min}, which minimizes cross-validated prediction error, and λ_{1se}, the largest value of lambda within one standard error of the minimum error, which favors simpler and more stable models. The performance of the model in exact prediction was examined and represented by mean squared error (MSE) on the test set, which was separate and held out from the training set. When looking at MSE, the square root can be taken, and root MSE can be looked at as the prediction error in terms of the same units as the prediction variable, which is minutes in this instance.

Results

Both the Ridge and Lasso models showed a high prediction error in this modeling instance of being restricted to pre-departure information. When compared to the other models we examined, the RMSE values for ridge and lasso were substantially larger than the RMSE of those other models. This result was seen and consistent for all lambda values considered. Although upon first thought this level of error may seem to invalidate the model, it indicates a key observation and limitation - departure delays are strongly influenced by real-time operational events that take place on the day of the flight or very close to departure time. These events are almost never observable at the scheduling stage, which means it's justifiable to have limited prediction accuracy in the ridge and lasso models.

Regardless of the high prediction error, the lasso model performed its variable selection as hoped and identified a subset of predictors that were associated with the risk of departure delay occurring. These predictor variables showed an alignment with a consistent direction, and all were intuitively reasonable indicators.

Starting from the general scheduling side of things, we saw that a later scheduled departure time was associated with an increased risk of delay. This makes sense since there are tons of flights scheduled in the afternoon and evening of any given day, and then any earlier flight that gets delayed stacks into the afternoon and causes a chain reaction. Visibility, pressure, and humidity emerged as relevant weather-related predictors as well. Nothing too interesting there, as it's quite intuitive that better weather (good visibility, high pressure, lower humidity) indicates a lower risk of delay, and vice versa as well. Lastly were a couple of specific carrier effects. Flights operated by EV airlines were at a slightly higher risk of delay in general, and flights operated by US Airways were at a slightly lower risk of delay.

Importantly, while these variables provide useful signals, they explain only a limited portion of total delay variability when considered in isolation.

```
## Ridge lambda.min: 3.328936
```

```
## Ridge lambda.1se: 86.3867
```

```
## [1] 36.2616
```

```
## Lasso lambda.min: 0.1375521
```

```
## Lasso lambda.1se: 1.861145
```

Interpretation

The main result of this analysis is that departure delay is extremely hard to predict exactly and accurately using only pre-departure information. Many of the primary causes of departure delays are things that arise later into the operational process, such as late inbound aircraft, crew availability constraints, gate congestion, maintenance issues, evolving weather conditions, and air traffic control restrictions. Because these types of factors are very difficult to know at the time of scheduling, these models are inherently limited in prediction availability. The best that can be done and what the data is suited for is identifying relative delay risk over any type of precise estimate.

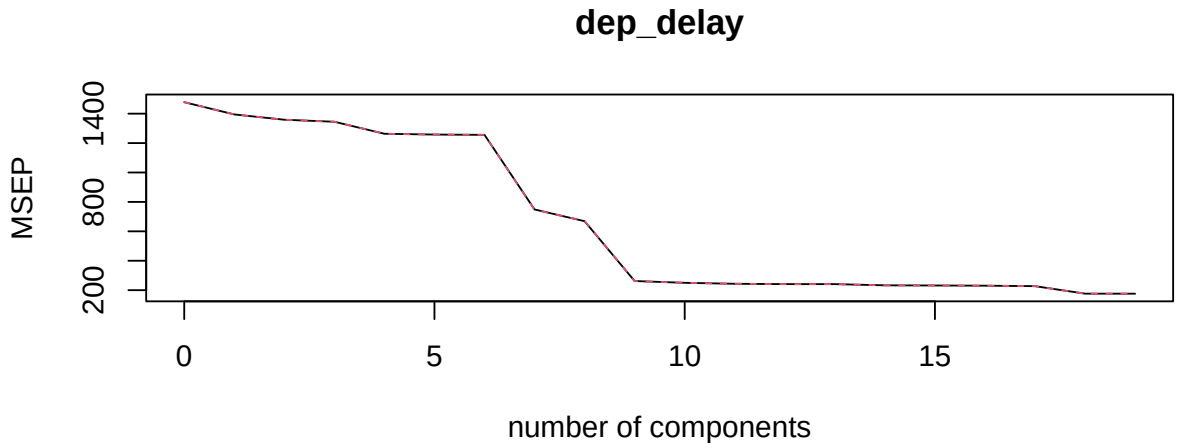
```
##                 lambda.1se
## carrierEV       5.32847460
## carrierUS      -0.51052115
## visib          -0.22226000
## humid           0.18933863
## pressure       -0.06095547
## sched_dep_time  0.01327433
```

PCA / PLS

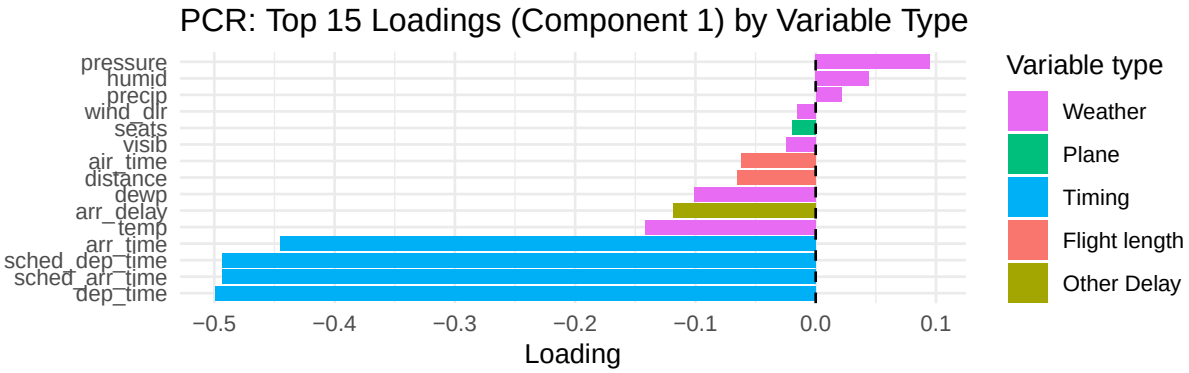
Since this dataset includes a large number of predictors with a high level of multi-colinearity, an analysis using Principal Components Regression and Partial Least Squares Regression was performed. New training and test dataframes were created which excluded all categorical variables. The PCR was included to examine how our predictors were related to each other and define groups of related predictors.

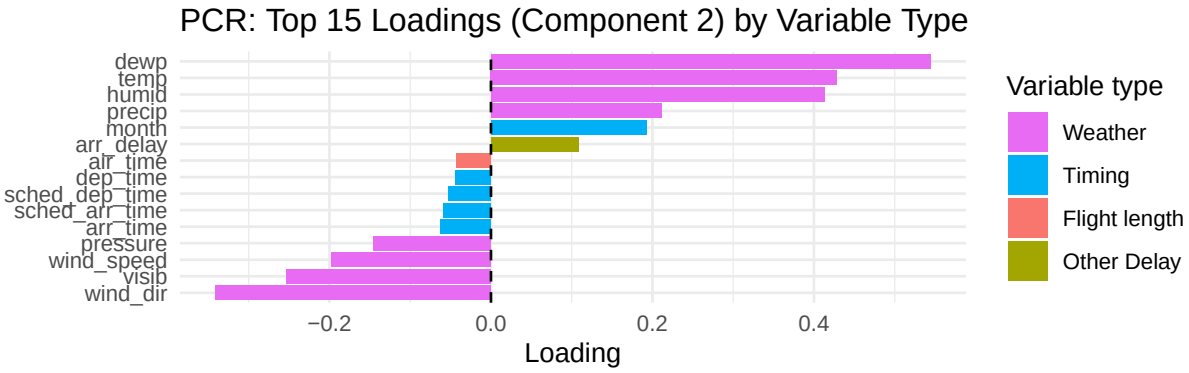
PCR

The PCR model was fit using the new training dataframe with dep_delay as the response, using leave one out cross validation. The best RMSEP for this model was achieved using 9 components. With a lower number of components, the models performed poorly with regards to the response but with 9 components, this model explained 81% of the variation in the predictors and 82% of the variation in the response. When using the model to predict on our test data, it produced the lowest MSE, equivalent to that of the PLS model below, of any of the models we tested at 171.72.



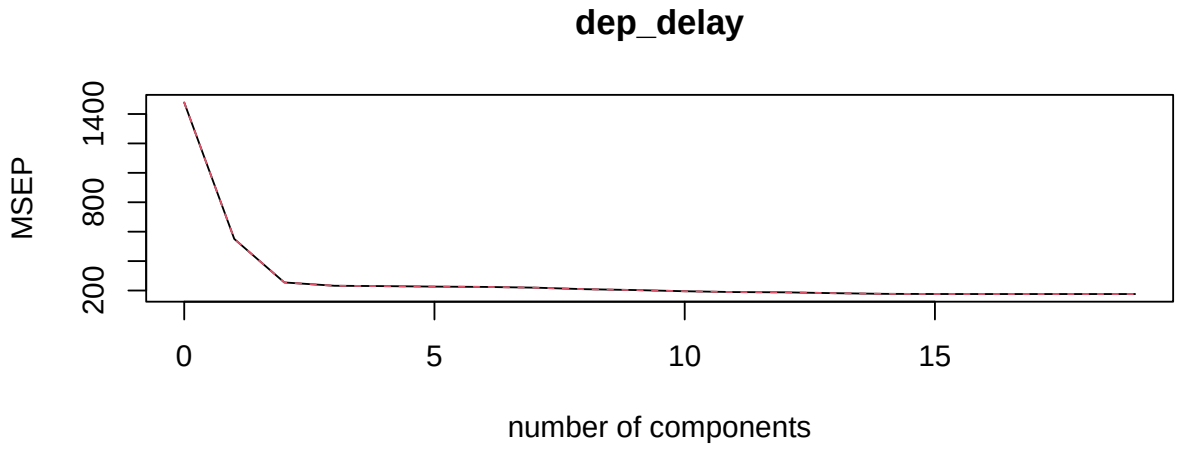
Upon examination of the loadings, the first three components appeared to have logically grouped predictors, with the first component using primarily timing variables (scheduled departure time, actual departure time, arrival time...), the second component using primarily weather variables (dewpoint, temperature, humidity, wind direction...) and the third component adding in some flight length variables (distance, airtime, seats). Arrival delay is present in many of the components with lesser magnitude. The rest of the components are composed from similar variables. At component 6, some plane specific variables are introduced including engines and year constructed. This suggests that the first three groups of variables dominate the relationship with departure delays, with plane specific variables also playing a more minor role. The first two components are shown below.





PLS

The PCR model was fit the same as for the PCR model above. The best RMSEP for this model was achieved using just 2 components. This model explained 83% of the variation in the predictors with just these two components. When using the model to predict on our test data, it produced the lowest MSE of any of the models we tested, equivalent to the PCR model at 171.72.



Upon examination of the loadings, with the exception of arrival delay playing a larger role, we found that both components are predictors of a very similar nature to those in the PCR model above. They are both composed primarily of flight timing variables and weather variables, with some flight distance variables

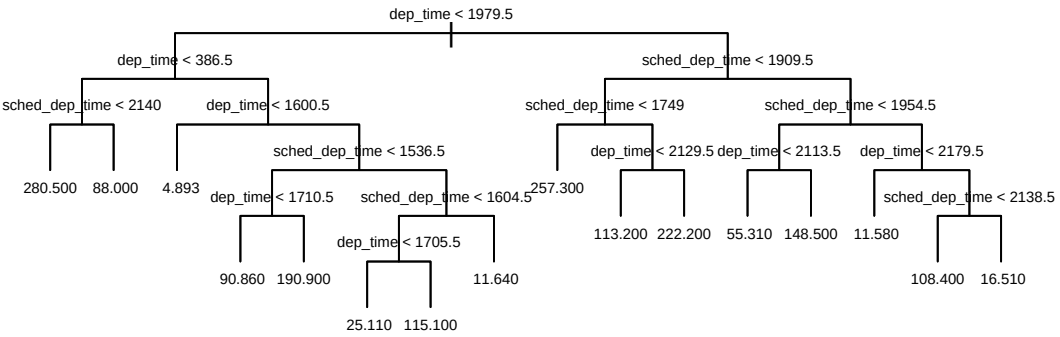
present. The only plane specific variable that appears strongly is seats, which is also correlated with flight timings. Since both the PCR and PLS models explained flight delays well, it suggests that arrival delay is not completely necessary to predict departure delays, despite its obvious association. This is consistent with the groupings seen in the PCR analysis and confirms that the other variables can comprise most of the predictive power.

Operational Implications

The PCR and PLS models were the best performing models of all those we tried. This is unsurprising given the large number of predictors in the dataset and the large amount of collinearity. While PLS is intended to explain the response more directly than the PCR, the PCR model also performed very well for prediction. The interpretation of both models reveals that flight timing is the primary predictor of flight departure delays, with weather playing a large secondary role, and flight length and plane model specific variables playing a lesser role. While arrival delay is seen playing a major role in the PLS model, it is used much less in the PCR model, proving that it is possible to make good flight predictions without it. Since flight timings and weather are difficult for the Port Authority to control, using the other variables as guidance for which plane might also be prone to delays in planning should form the basis for our recommendation, however the active usage of predictions using real time weather and departure and arrival destination may prove to be of huge benefit to the Port Authority. The ability to mitigate or eliminate any part or the whole of a delay in real time may be within reach if the technology to predict them in real time is available.

Trees

We also explored tree-based methods for predicting departure delays based on the dataset. First, we fit a regression tree to predict dep_delay based on all variables in the dataset besides arrival_time, arr_delay, model, and dest. The first two were removed because they are not information available before a flight takes off, so they would not be usable to predict a departure delay ahead of time. The second two were removed due to their large number of categories which exceeded the amount possible for categorical variables in the package used. The resulting regression tree returned a test MSE of 687.5693. However, regression trees are generally vulnerable to overfitting, so we also tried pruning the tree to address this. Pruning a regression tree involves applying a penalty parameter on additional branches using cross validation to select an appropriate penalty. In our case, likely due to the very large sample size, the pruning the tree did not result in any differences compared to the original tree. Although the test MSE is high, suggesting a poor fit, examining the variables used in the tree can inform us about what variables appear to influence departure delays the most. The splits in the tree grown involved almost exclusively schedule and time related variables, with various splits of departure time and scheduled departure time being used in the tree.



We also explored using two models for predicting dep_delay that involve the creation of a large number of trees to fit an overall combined model, bagging and random forest methods. In the bagging method, a tree is fitted for each of a given number of bootstrap samples of the data and a majority vote from this set of trees is utilized to create a combined model. Random forest works in the same way, but adds an additional step by limiting the parameters available for each split in each tree to a randomly selected subset of all parameters. This helps to reduce correlation. However, for our fitting of bagging and random forest models, the large sample size caused performance issues in successfully creating these models. We addressed this issue by lowering the number of trees created for each model to just 50 so that the models would successfully be created. The bagging model returned a test MSE of 8.677 and the random forest a test MSE of 165.4381. However, it is very important to point out that limiting the number of trees created to just 50 is highly inadvised, as that is too small a number to get reliable results. As such, very little insights can be gained from these two models.

Insights and Recommendations

Recommendations

Based on these findings, we have several recommendations:

- Early predictions can be employed to flag high-risk flights instead of predicting the lengths of the actual delays. These early models perform best for prioritization and preparedness.
- Specific operational resources can be allocated proactively by focusing on the higher-risk flights, for example, allocating more gate staff or grounds crew to the late-day departures and those operating under adverse weather conditions.
- Real-time integration of operational information to enhance prediction accuracy as it approaches departure would be very useful. The real-time data to be integrated will increase prediction accuracy as it approaches departure, and it can focus on various areas such as the status of inbound arrivals, crew status, congestion levels, the status of the plane's maintenance, and dynamic changes to weather patterns.

- Lastly, you can implement a layered prediction system where risk assessment is performed early while the system learns to update the delay calculations accordingly with emerging trends. This would be the best way to have an accurate prediction and have it updated in real time.

Conclusion

Overall, this analysis provides evidence that although early prediction of departure delay using strictly pre-departure information is not highly accurate in itself, it still provides us with valuable operational insight. We can conclude that departure delays in NYC are primarily driven by downstream operational events, and since these factors are not easily measurable at the scheduling stage, early prediction models should not be utilized as an aid in calculating these delays. Nonetheless, the greatest operational value is gained when early risk signals are combined with real-time data in a merged system to support dynamic decision making.

Appendix - Relevant Code

```
##Loaded Libraries
library(nycflights13)
library(ggplot2)
library(tree)
library(dplyr)
library(patchwork)
library(glmnet)
library(pls)
```

```
##LOAD DATA
data("flights")
data("weather")
data("planes")

##MERGE DATASETS
##join flights and planes datasets
dataset = left_join(flights, planes, by = "tailnum")
#dataset = merge(dataset, weather, by = "time_hour")
#remove observations without matching planes
dataset = dataset %>% filter(!is.na(year.y))
#Remove unnecessary columns
dataset = dataset %>% dplyr::select(-year.x, -hour, -minute, -speed)
#Rename year.y to provide more context
dataset = dataset %>% rename(year_constructed = year.y)

##join previous dataset with weather
dataset = left_join(dataset, weather, by = c("time_hour", "origin"))
#Check number of non-NA columns in combined dataset
dataset = dataset %>% filter(!is.na(wind_gust))
#remove duplicate columns
dataset = dataset %>% dplyr::select(-year, -month.y, -day.y, -wind_gust)
#rename columns to more intuitive names
dataset = dataset %>% rename(month = month.x,
                             day = day.x)
```

```r
##Additional Data Cleaning
#Remove all rows w/ NAs
dataset = na.omit(dataset)
#Remove unnecessary columns
dataset = dataset %>% dplyr::select(-day, -flight, -tailnum, -time_hour, -hour)
```

EDA Plots

```r
hist(na.omit(flights$dep_delay),
     breaks = 50,
     main = "Histogram of Departure Delay",
     xlab = "Delay (minutes)")
```

```r
#Other flights dataset variables
par(mfrow = c(2,2))

hist(flights$dep_time,
     main = "Histogram of Departure Time",
     xlab = "Military Time",
     col = "lightgray",
     border = "white")

hist(flights$sched_dep_time,
     main = "Histogram of Scheduled Departure Time",
     xlab = "Military Time",
     col = "lightgray",
     border = "white")

hist(flights$arr_time,
     main = "Histogram of Arrival Time",
     xlab = "Military Time",
     col = "lightgray",
     border = "white")

hist(flights$sched_arr_time,
     main = "Histogram of Scheduled Arrival Time",
     xlab = "Military Time",
     col = "lightgray",
     border = "white")
```

```r
#Departure and Arrival time
p1 = ggplot(dataset, aes(x = dep_time, y = dep_delay)) + geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "Average Departure Delay by
Flight Departure Time",
       x = "Departure Time (Military Time)",
       y = "Average Departure Delay in Minutes") + theme_minimal()

p2 = ggplot(dataset, aes(x = arr_time, y = dep_delay)) + geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "Average Departure Delay by
```

```
Flight Arrival Time",
       x = "Arrival Time (Military Time)",
       y = "Average Departure Delay in Minutes") + theme_minimal()

wrap_plots(p1, p2, nrow = 1)
```

```
##Arrival Delay
ggplot(dataset, aes(x = arr_delay, y = dep_delay)) + geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "Average Departure Delay by Arrival Delay",
       x = "Arrival Delay in Minutes",
       y = "Average Departure Delay in Minutes") + theme_minimal()
```

```
##Weather continuous plots
p1 = ggplot(dataset, aes(x = temp, y = dep_delay)) + geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "Average Departure Delay
by Temperature",
       x = "Temperature (Fahrenheit)",
       y = "Average Departure
Delay in Minutes") + theme_minimal()

p2 = ggplot(dataset, aes(x = humid, y = dep_delay)) + geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "
by Humidity",
       x = "Humidity",
       y = "Average Departure
Delay in Minutes") + theme_minimal()

p3 = ggplot(dataset, aes(x = pressure, y = dep_delay))+ geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "
by Pressure",
       x = "Pressure",
       y = "") + theme_minimal()

p4 = ggplot(dataset, aes(x = visib, y = dep_delay)) + geom_point() +
  geom_smooth(method = "gam") +
  labs(title = "
by Visibility",
       x = "Visibility (Miles)",
       y = "") + theme_minimal()

wrap_plots(p1, p2, p3, p4, nrow = 2, ncol = 2)
```

```
#Bivariate barplots for all categorical predictors with the response of dep_delay

#By month
avg_month = dataset %>% group_by(month) %>%
  summarize(
    avg_dep_delay = mean(dep_delay)
    )
```

```
ggplot(avg_month, aes(x = factor(month, levels = 1:12, labels = month.abb),
                      y = avg_dep_delay)) +
  geom_col() + labs(title = "Average Departure Delay by Month",
                    x = "Month",
                    y = "Average Departure Delay in Minutes") +
  theme_minimal()
```

```
#By carrier

p1 = ggplot(dataset, aes(x = carrier)) + geom_bar() +
  labs(title = "Number of Flights by Carrier",
       x = "Flight Carrier",
       y = "Flight County")  + theme_minimal()

avg_carrier = dataset %>% group_by(carrier) %>% filter(n() >= 1000) %>%
  summarize(
    avg_dep_delay = mean(dep_delay)
    )

p2 = ggplot(avg_carrier, aes(x = carrier, y = avg_dep_delay)) + geom_col() +
  labs(title = "Average Departure Delay by Carrier",
       x = "Flight Carrier",
       y = "Average Departure Delay in Minutes") + theme_minimal()

wrap_plots(p1, p2, nrow = 1)
```

```
#By origin
avg_origin = dataset %>% group_by(origin) %>%
  summarize(
    avg_dep_delay = mean(dep_delay)
    )

p1 = ggplot(avg_origin, aes(x = origin, y = avg_dep_delay)) + geom_col() +
  labs(title = "Average Departure Delay
by Airport of Departure",
       x = "Airport of Departure",
       y = "Average Departure Delay in Minutes") + theme_minimal()

#By destination
avg_dest = dataset %>% group_by(dest) %>% filter(n() >= 1000) %>%
  summarize(
    avg_dep_delay = mean(dep_delay)
    )

p2 = ggplot(avg_dest, aes(x = dest, y = avg_dep_delay)) + geom_col() +
  labs(title = "Average Departure Delay
by Destination Airport",
       x = "Destination Airport",
       y = "Average Departure Delay in Minutes") + theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

wrap_plots(p1, p2, nrow = 1)
```

```r
#By plane manufacturer
avg_manufact1000 = dataset %>% group_by(manufacturer) %>% filter(n() >= 1000) %>%
  summarize(
    avg_dep_delay = mean(dep_delay)
    )

p3 = ggplot(avg_manufact1000, aes(x = manufacturer, y = avg_dep_delay)) + geom_col() +
  labs(title = "Average Departure Delay
by Plane Manufacturer (n >= 1000)",
       x = "Manufacturer",
       y = "Average Departure
Delay in Minutes") + theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#By model
avg_model = dataset %>% group_by(model) %>% filter(n() >= 1000) %>%
  summarize(
    avg_dep_delay = mean(dep_delay)
    )

p4 = ggplot(avg_model, aes(x = model, y = avg_dep_delay)) + geom_col() +
  labs(title = "Average Departure Delay
by Airplane Model (n >= 1000)",
       x = "Airplane Model",
       y = "Average Departure Delay in Minutes") + theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

wrap_plots(p3, p4, nrow = 1)
```

Model Selection Setup

```r
#Preparation
set.seed(4911) #For reproducibility

ix = sample(1:nrow(dataset),nrow(dataset)*0.8) # 80/20 split
dataset.train = dataset[ix, ]
dataset.test = dataset[-ix, ]
```

Model Selection - Linear

```r
carrier_counts <- sort(table(dataset$carrier), decreasing = TRUE)

barplot(carrier_counts,
        las = 2,                  # rotate labels
        main = "Number of Flights by Carrier",
        xlab = "Carrier",
        ylab = "Number of Flights")

dest_counts <- sort(table(dataset$dest), decreasing = TRUE)
```

```r
barplot(dest_counts,
        las = 2,                  # rotate labels
        main = "Number of Flights by Destination",
        xlab = "Destination",
        ylab = "Number of Flights")

manu_counts <- sort(table(datasetb$manufacturer), decreasing = TRUE)
manu_df <- as.data.frame(manu_counts)
colnames(manu_df) <- c("manu", "count")

ggplot(manu_df[1:10, ], aes(x = reorder(manu, count), y = count)) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Top 10 Manufacturers by Number of Flights",
    x = "Manufacturer",
    y = "Number of Flights"
  )

model_counts <- sort(table(datasetb$model), decreasing = TRUE)
model_df <- as.data.frame(model_counts)
colnames(model_df) <- c("model", "count")

ggplot(model_df[1:30, ], aes(x = reorder(model, count), y = count)) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Top 30 Models by Number of Flights",
    x = "Model",
    y = "Number of Flights"
  )
```

Model Select - Ridge / Lasso

```r
##Full code

# Drop rows where dep_delay is missing
train0 <- subset(dataset.train, !is.na(dep_delay))
test0  <- subset(dataset.test,  !is.na(dep_delay))

#training model with no arrival delay, arrival time, air time, or departure time
mf_train2 <- model.frame(dep_delay ~ . - arr_delay - arr_time - air_time - dep_time,
                         data = train0, na.action = na.omit)

# Pull y and X
x_train <- model.matrix(dep_delay ~ . - arr_delay - arr_time - air_time - dep_time,
                        data = mf_train2)[, -1]
y_train <- mf_train2$dep_delay

#test model with no arrival delay, arrival time, air time, or departure time
mf_test2  <- model.frame(dep_delay ~ . - arr_delay - arr_time - air_time - dep_time,
```

```
                        data = test0,  na.action = na.omit)

y_test <- mf_test2$dep_delay
x_test <- model.matrix(dep_delay ~ . - arr_delay - arr_time - air_time - dep_time,
                       data = mf_test2)[, -1]

train_cols <- colnames(x_train)
test_cols  <- colnames(x_test)

missing_in_test <- setdiff(train_cols, test_cols)
extra_in_test   <- setdiff(test_cols, train_cols)

if (length(missing_in_test) > 0) {
  x_test <- cbind(
    x_test,
    matrix(0, nrow = nrow(x_test), ncol = length(missing_in_test),
           dimnames = list(NULL, missing_in_test))
  )
}
# Keep only training columns, in the same order
x_test <- x_test[, train_cols, drop = FALSE]

# Ridge model
set.seed(4911)
cv_ridge <- cv.glmnet(
  x = x_train,
  y = y_train,
  alpha = 0,          # ridge
  standardize = TRUE
)

# lambdamin is lambda that gives lowest CV error, lambda1se is largest lambda that CV error is within o
cat("Ridge lambda.min:", cv_ridge$lambda.min, "\n")
cat("Ridge lambda.1se:", cv_ridge$lambda.1se, "\n")

# Ridge predictions on test set (lambda.min)
ridge_pred <- predict(cv_ridge, newx = x_test, s = "lambda.min")

# Test RMSE
rmse_ridge <- sqrt(mean((y_test - ridge_pred)^2))
rmse_ridge

# Lasso model
set.seed(4911)
cv_lasso <- cv.glmnet(
  x = x_train,
  y = y_train,
  alpha = 1,          # lasso
  standardize = TRUE
)

# lambdamin is lambda that gives lowest CV error, lambda1se is largest lambda that CV error is within o
cat("Lasso lambda.min:", cv_lasso$lambda.min, "\n")
```

```r
cat("Lasso lambda.1se:", cv_lasso$lambda.1se, "\n")

lasso_pred_min <- predict(cv_lasso, newx = x_test, s = "lambda.min")
rmse_lasso_min <- sqrt(mean((y_test - lasso_pred_min)^2))
rmse_lasso_min

# Ridge RMSE at lambda.1se
ridge_pred_1se <- predict(cv_ridge, newx = x_test, s = "lambda.1se")
mse_ridge_1se <- mean((y_test - ridge_pred_1se)^2)
mse_ridge_1se

# Lasso RMSE at lambda.1se
lasso_pred_1se <- predict(cv_lasso, newx = x_test, s = "lambda.1se")
rmse_lasso_1se <- sqrt(mean((y_test - lasso_pred_1se)^2))
rmse_lasso_1se

lasso_coefs <- coef(cv_lasso, s = "lambda.1se")
lasso_mat   <- as.matrix(lasso_coefs)

# Drop intercept
lasso_mat <- lasso_mat[-1, , drop = FALSE]

# Keep non-zero coefficients
nonzero <- lasso_mat[lasso_mat[,1] != 0, , drop = FALSE]

# How many variables selected?
nrow(nonzero)

# Names + coefficient values of selected predictors at lambda.1se
nonzero[order(abs(nonzero[,1]), decreasing = TRUE), , drop = FALSE]
```

Model Selection - PCA / PLS

```r
## PCA / PLS
## Create dataframes that include only numeric variables
train_dat <- dataset.train %>% select(dep_delay, where(is.numeric)) %>% as.data.frame()
test_dat  <- dataset.test  %>% select(dep_delay, where(is.numeric)) %>% as.data.frame()

# PCR
set.seed(4911)

## Fit PCR model to the training data
pcr.fit <- pcr(dep_delay ~ ., data = train_dat,
               scale = TRUE,
               validation = "CV")

# Validate the model
summary(pcr.fit)
validationplot(pcr.fit, val.type = "MSEP")

top_k <- 15
```

```r
groups <- list(
  Weather = c("temp","dewp","humid","wind_dir","wind_speed","wind_gust","precip","pressure","visib"),
  Timing = c("hour","minute","day","wday","dep_time","sched_dep_time","arr_time","sched_arr_time", "mon
  Plane = c("engines", "year_constructed","seats"),
  `Flight length` = c("distance","air_time"))

fill_cols <- c(
  "Weather"       = "#E76BF3",
  "Timing"        = "#00B0F6",
  "Plane"        = "#00BF7D",
  "Flight length" = "#F8766D",
  "Other Delay"         = "#A3A500"
)

## Create loading plots
pcr_load <- loadings(pcr.fit)

make_loading_plot <- function(comp) {
  load_df <- tibble(
    var = rownames(pcr_load),
    loading = as.numeric(pcr_load[, comp]),
    group = case_when(
      var %in% groups$Weather ~ "Weather",
      var %in% groups$Timing ~ "Timing",
      var %in% groups$Plane ~ "Plane",
      var %in% groups$`Flight length` ~ "Flight length",
      TRUE ~ "Other Delay"
    )
  ) %>%
    slice_max(abs(loading), n = top_k) %>%
    mutate(
      var = reorder(var, loading),
      group = factor(group, levels = c("Weather","Plane","Timing","Flight length","Other Delay"))
    )

  ggplot(load_df, aes(var, loading, fill = group)) +
    geom_col() +
    geom_hline(yintercept = 0, linetype = "dashed") +
    coord_flip() +
    scale_fill_manual(values = fill_cols) +
    labs(
      title = paste0("PCR: Top ", top_k, " Loadings (Component ", comp, ") by Variable Type"),
      x = NULL, y = "Loading", fill = "Variable type"
    ) +
    theme_minimal()
}

p_comp1 <- make_loading_plot(1)
p_comp2 <- make_loading_plot(2)
wrap_plots(p_comp1, p_comp2, nrow = 2, ncol = 1)
```

```r
#PLS
set.seed(4911)
```

```r
## Fit a partial least squares regression
pls.fit <- plsr(dep_delay ~ ., data = train_dat,
                scale = TRUE,
                validation = "CV")
## validate the model
summary(pls.fit)
validationplot(pls.fit, val.type = "MSEP")
```

Model Selection - Trees

```r
#Preparation
set.seed(4911) #For reproducibility

#Remove arrival time variables, since they are dependent on dep_delay
dataset = dataset %>% dplyr::select(-arr_time, -arr_delay)
#Adjust formatting of some variables so that they can work with regression code
dataset = dataset %>% mutate(across(where(is.character), as.factor))

ix = sample(1:nrow(dataset),nrow(dataset)*0.8) # 80/20 split
dataset.train = dataset[ix, ]
dataset.test = dataset[-ix, ]

#Use modified dataset without variables with too many factors to work with  tree() function
tree.train = dataset.train %>% dplyr::select(-model, -dest)
tree.test = dataset.test %>% dplyr::select(-model, -dest)

#Create regression tree
tree.fit = tree(dep_delay ~ ., data = tree.train)

#Prune the tree
cvTree.fit = cv.tree(tree.fit)
size = cvTree.fit$size[which.min(cvTree.fit$dev)]
prune.fit = prune.tree(tree.fit, best = size)
#Plot Tree
plot(prune.fit, type = "uniform")
text(prune.fit, pretty = 0, cex = 0.5)
```

```r
#Using Bagging
set.seed(4911)

#Fit Model
bag.fit = randomForest(dep_delay ~ ., data = tree.train, mtry = (ncol(tree.train) - 1),
                       ntree = 50, importance = TRUE)
#Calculate test MSE
bag.pred = predict(bag.fit, newdata = tree.test)
mean((bag.pred - tree.test$dep_delay)^2)

#Using Random Forest
set.seed(4911)

#Fit Model
```

```r
rf.fit = randomForest(dep_delay ~ ., data = tree.train,
                      ntree = 50, importance = TRUE)
#Calculate test MSE
rf.pred = predict(rf.fit, newdata = tree.test)
mean((rf.pred - tree.test$dep_delay)^2)
```