

NYC Flight Delays

Cameron Lucas, Chris Holman, Irfan Fazdane

2023-01-21

Importing libraries

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=40), tidy=TRUE)
library(nycflights13)
library(dplyr)
library(ggplot2)
library(car)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)
```

Introduction

Using the nycflights13 dataset we found data to back up insights that relate to flight delay times at JFK, Newark, and La Guardia airports in New York. To do this, we first inspected the data through analysis of the various factors included to better understand what information we have available to us. After this, we used statistical modeling to find the most important variables in the dataset that relate to the flight delay times in order to back up our findings. Using the results, we developed insights on the most important variables that need to be addressed in order to decrease delay times in the future.

Getting data

```
data(flights)
data(weather)
```

EDA

Checking for na in weather

```
# number of na per column
cbind(lapply(lapply(weather, is.na), sum))

## [,1]
## origin      0
## year        0
## month       0
## day         0
## hour        0
## temp        1
## dewp        1
## humid       1
## wind_dir    460
## wind_speed  4
## wind_gust   20778
## precip      0
## pressure    2729
## visib       0
## time_hour   0

# removing wind gust (too many NA and just a confusing var)
if ("wind_gust" %in% colnames(weather)) {
  # cleaned weather
  w = subset(weather, select = -c(wind_gust))
}
```

We decided to remove the wind gust variable because it had a large amount of NA values. About three quarters of the data do not have values for this predictor. Also, from reading the data dictionary it was hard to understand what the wind gust variable meant.

Cleaning flights data

```
cbind(lapply(lapply(flights, is.na), sum))

## [,1]
## year        0
## month       0
## day         0
## dep_time    8255
## sched_dep_time 0
## dep_delay   8255
## arr_time    8713
## sched_arr_time 0
## arr_delay   9430
## carrier     0
```

```

## flight      0
## tailnum    2512
## origin     0
## dest       0
## air_time   9430
## distance   0
## hour       0
## minute     0
## time_hour  0

# cleaned flights
f = flights[!is.na(flights$air_time), ]

f$sigDelay = f$dep_delay > 15

```

Within the flights data set, there is another issue of NA values with the air time variable having the most. We interpreted NA for air time as meaning the flights was cancelled. Since cancellation is different from departure delays, we removed all observations with out an air time. This gets rid of the rest of NA values for the whole flights data set.

We also created a variable named “sigDelay” which is a binary variable that encodes whether or not a flight was delayed according to the FAA’s definition of a 15 minute threshold for if a flight is delayed.

Checking for multicollinearity

Weather Correlation

```

# getting only numeric columns and removing na's
w_numeric <- w[, sapply(w, is.numeric)]
w_numeric <- na.omit(w_numeric[, c(5:12)])

cor(w_numeric)

##          temp      dewp      humid wind_dir wind_speed
## temp 1.00000000 0.90162405 0.1071538 -0.1290057 -0.10894880
## dewp 0.90162405 1.00000000 0.5193125 -0.2460956 -0.18136793
## humid 0.10715384 0.51931252 1.0000000 -0.3249786 -0.20582870
## wind_dir -0.12900567 -0.24609556 -0.3249786 1.0000000 0.25445501
## wind_speed -0.10894880 -0.18136793 -0.2058287 0.2544550 1.00000000
## precip -0.02950825 0.04208537 0.1865167 -0.0780323 0.03005549
## pressure -0.25366597 -0.28858075 -0.1803573 -0.1988064 -0.13249274
## visib 0.04323954 -0.12773124 -0.4523975 0.1873828 0.04883138
##          precip pressure visib
## temp -0.02950825 -0.2536660 0.04323954
## dewp 0.04208537 -0.2885808 -0.12773124
## humid 0.18651675 -0.1803573 -0.45239754
## wind_dir -0.07803230 -0.1988064 0.18738277
## wind_speed 0.03005549 -0.1324927 0.04883138
## precip 1.00000000 -0.1079932 -0.34709198
## pressure -0.10799318 1.0000000 0.12277393
## visib -0.34709198 0.1227739 1.00000000

```

Before we look for trends with delays, it's important to get a feel for the data and understand how variables related to each other. There seems to be some high correlations within the weather data set which could lead to come multicollinearity issues. This means that multiple variables encode similar information. When modeling, this can reduce the model's effectiveness. The variables with the highest correlations are temperature, dew point, and humidity.

Flights Correlation

```
# getting only numeric columns and removing na's
f_numeric <- f[, sapply(f, is.numeric)]
f_numeric <- na.omit(f_numeric[, c(4:12)])

cor(f_numeric)

##          dep_time sched_dep_time dep_delay arr_time sched_arr_time
## dep_time      1.0000000    0.95482687  0.25961272 0.66250900   0.78444199
## sched_dep_time 0.95482687     1.00000000  0.19892350 0.64438677   0.78058744
## dep_delay      0.25961272    0.19892350  1.00000000 0.02942101   0.16049724
## arr_time       0.66250900    0.64438677  0.02942101 1.00000000   0.79078877
## sched_arr_time 0.78444199    0.78058744  0.16049724 0.79078877   1.00000000
## arr_delay       0.23230573    0.17389620  0.91480276 0.02448214   0.13326129
## flight          0.04153017    0.02840127  0.05396975 0.02500740   0.01394723
## air_time        -0.01461948   -0.01553213 -0.02240508 0.05429603   0.07891830
## distance        -0.01413373   -0.01293250 -0.02168090 0.04718917   0.07361354
##          arr_delay     flight    air_time   distance
## dep_time        0.23230573  0.04153017 -0.01461948 -0.01413373
## sched_dep_time  0.17389620  0.02840127 -0.01553213 -0.01293250
## dep_delay        0.91480276  0.05396975 -0.02240508 -0.02168090
## arr_time         0.02448214  0.02500740  0.05429603  0.04718917
## sched_arr_time  0.13326129  0.01394723  0.07891830  0.07361354
## arr_delay        1.00000000  0.07286208 -0.03529709 -0.06186776
## flight           0.07286208  1.00000000 -0.47283836 -0.48146018
## air_time        -0.03529709 -0.47283836  1.00000000  0.99064965
## distance        -0.06186776 -0.48146018  0.99064965  1.00000000
```

The flights data set looks to have more cases of multicollinearity. This makes sense when looking at the variables and what they represent. Since most of the variables are related to departure and arrival times, the scheduled and actual times for each particular flight are similar. Also, variables like departure delay and arrival delay are derived from the difference between the actual and scheduled times. This means that only a few of these variables will prove to be useful when we begin modeling. Finally, variables like arrival delay or arrival time should not be used because in practice those will not be known before a flight takes off from New York.

```

# Looking for trends with Departure Delay

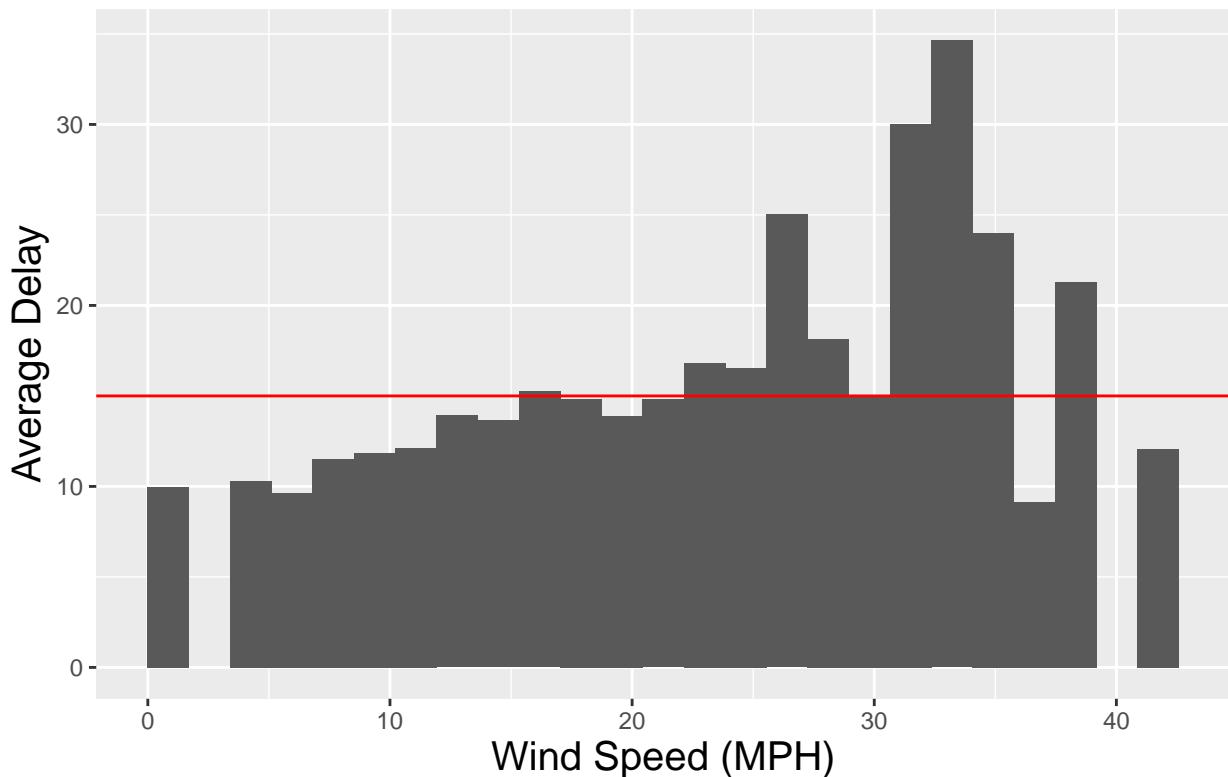
# merging together cleaned weather and flights data
wf <- merge(f, w, by = c("time_hour", "origin"))

ggplot(wf, aes(x = wind_speed, y = dep_delay)) + stat_summary_bin(fun = "mean",
  geom = "bar", bins = 25) + ylab("Average Delay") + xlab("Wind Speed (MPH)") +
  ggtitle("Effects of Wind Speed on Flight Delays ") + geom_hline(yintercept = 15,
  color = "red") + theme(axis.title = element_text(size = 15),
  plot.title = element_text(size = 20))

## Warning: Removed 78 rows containing non-finite values ('stat_summary_bin()').

```

Effects of Wind Speed on Flight Delays



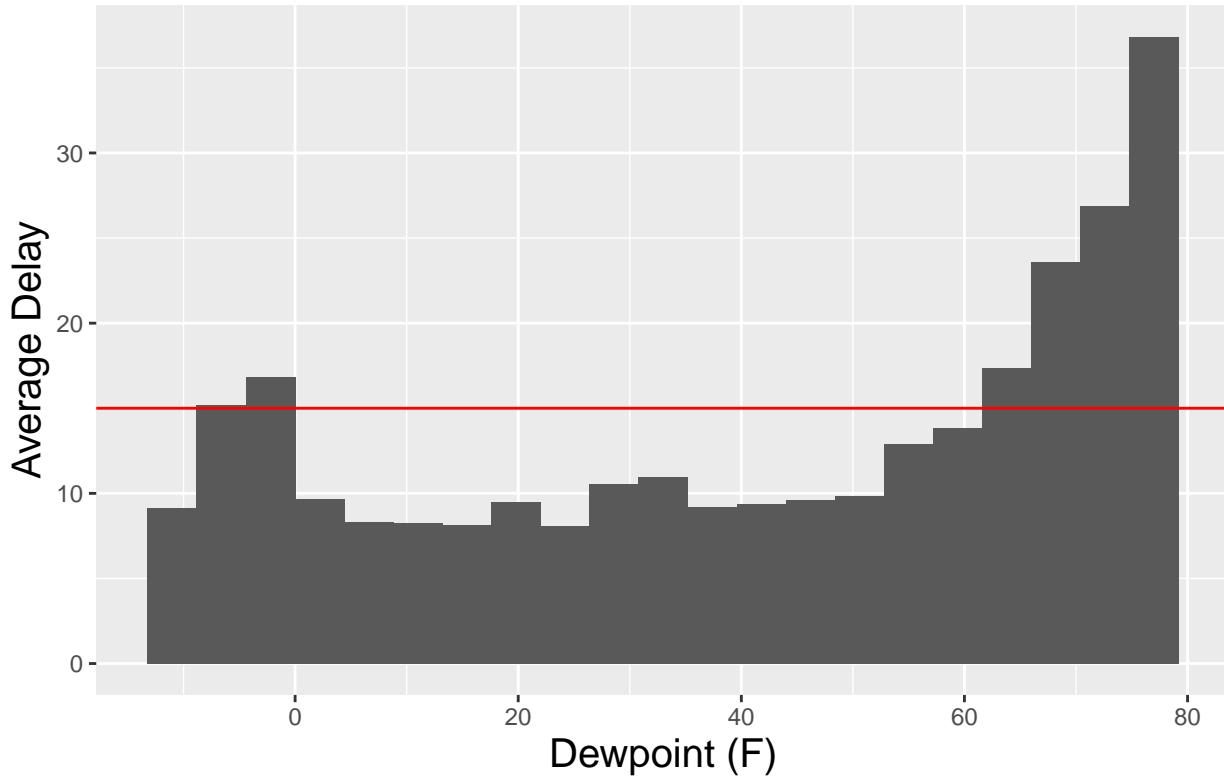
```

ggplot(wf, aes(x = temp, y = dep_delay)) + stat_summary_bin(fun = "mean",
  geom = "bar", bins = 20) + ylab("Average Delay") + xlab("Temperature (F)") +
  ggtitle("Effects of Temperature on Flight Delays ") + geom_hline(yintercept = 15,
  color = "red") + theme(axis.title = element_text(size = 15),
  plot.title = element_text(size = 20))

## Warning: Removed 17 rows containing non-finite values ('stat_summary_bin()').

```

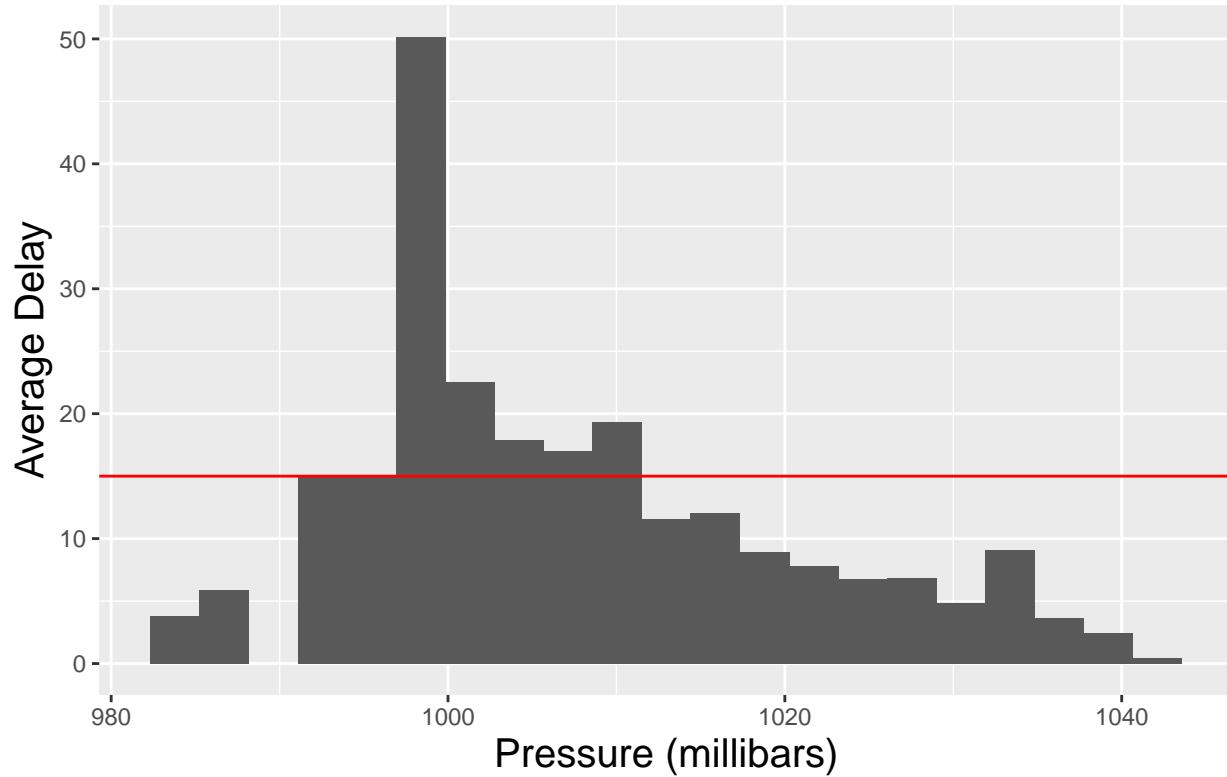
Effect of Dewpoint on Flights Delays



```
ggplot(wf, aes(x = humid, y = dep_delay)) + stat_summary_bin(fun = "mean",
  geom = "bar", bins = 20) + ylab("Average Delay") + xlab("Relative Humidity") +
  ggtitle("Effect of Humidity on Flights Delays") + geom_hline(yintercept = 15,
  color = "red") + theme(axis.title = element_text(size = 15),
  plot.title = element_text(size = 20))

## Warning: Removed 17 rows containing non-finite values ('stat_summary_bin()').
```

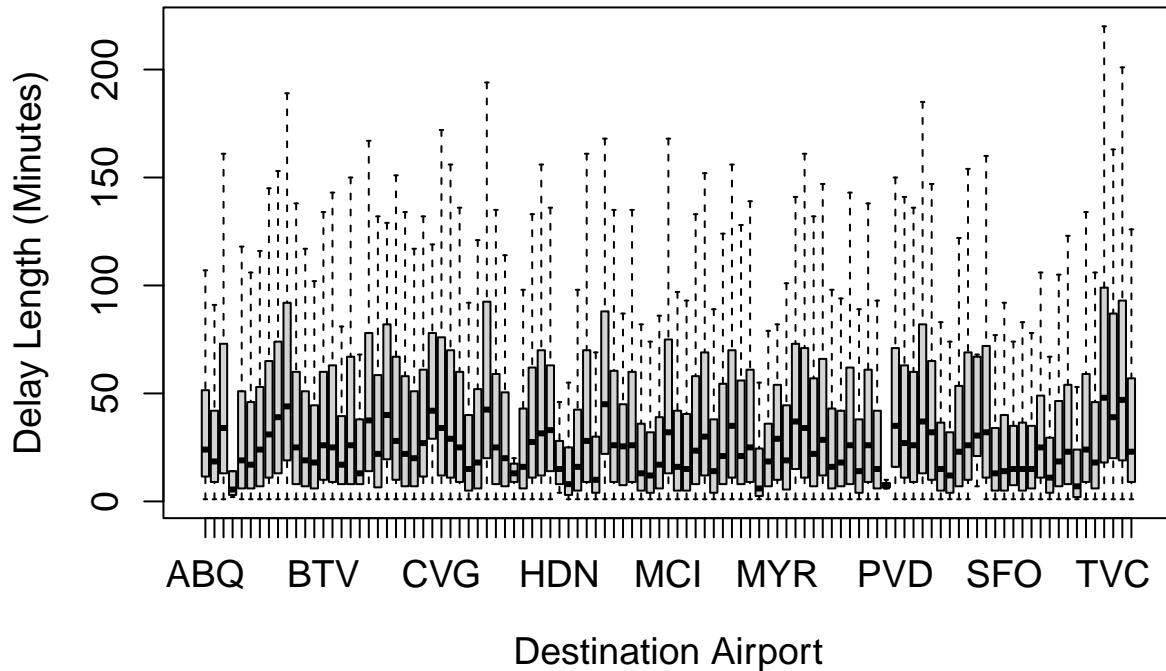
Effects of Air Pressure on Flight Delays



Takeways

It seems like wind speed, temperature, dew point, pressure, and humidity are the variables with strongest relationship with departure delay.

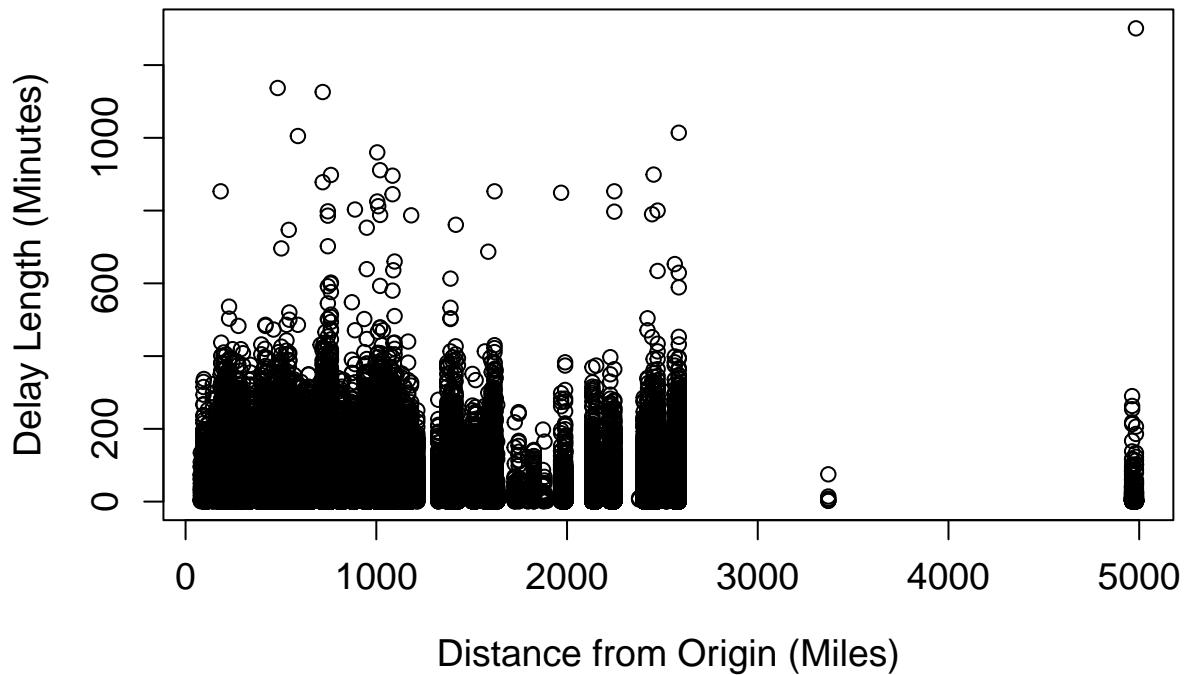
Delay Length vs. Destination



Frankly it's hard to tell which destinations are which, but it's enough to tell that there are clearly some with much better delay times than others, so this is definitely a variable we will want to use, at least during variable selection.

```
plot(flights$dep_delay[which(flights$dep_delay > 0)] ~ flights$distance[which(flights$dep_delay > 0)], xlab = "Distance from Origin (Miles)", ylab = "Delay Length (Minutes)", main = "Delay Length vs. Distance", cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

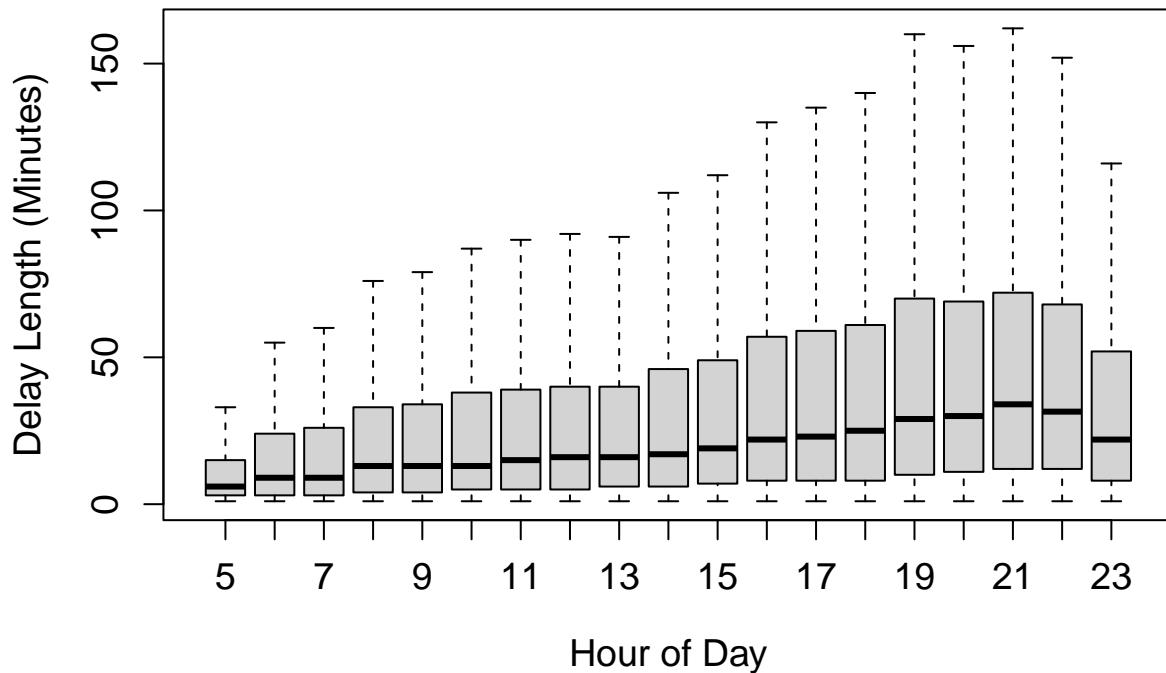
Delay Length vs. Distance



In our opinion, destination does not seem useful. The only real difference visually is at the ~3300 mile mark, but that has very few observations which makes it hard to say anything for sure.

```
boxplot(flights$dep_delay[which(flights$dep_delay > 0)] ~ flights$hour[which(flights$dep_delay > 0)], outline = F, xlab = "Hour of Day",
ylab = "Delay Length (Minutes)", main = "Delay Length vs. Hour",
cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

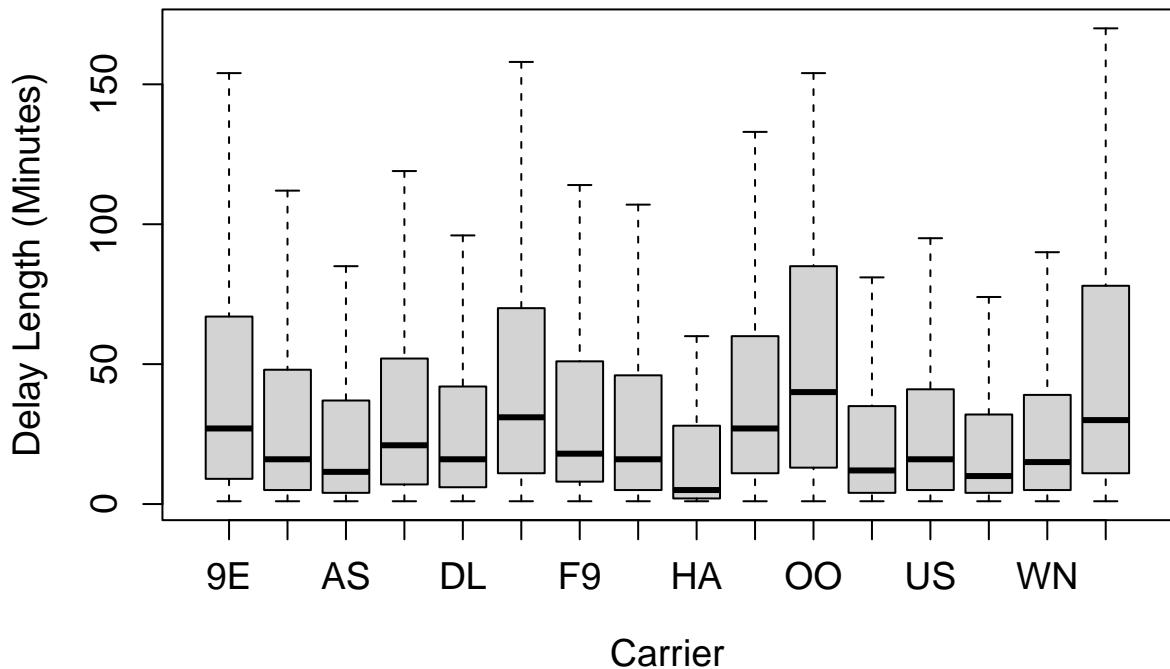
Delay Length vs. Hour



It seems pretty cut and dry that delay length goes up throughout the day until flights stop being scheduled. This makes perfect sense as delays are going to compound as delayed flights start getting in the way of other flights.

```
boxplot(flights$dep_delay[which(flights$dep_delay > 0)] ~ flights$carrier[which(flights$dep_delay > 0)], outline = F, xlab = "Carrier", ylab = "Delay Length (Minutes)", main = "Delay Length vs. Carrier", cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

Delay Length vs. Carrier



There does appear to be significant differences between carrier; we will continue with carrier into variable selection.

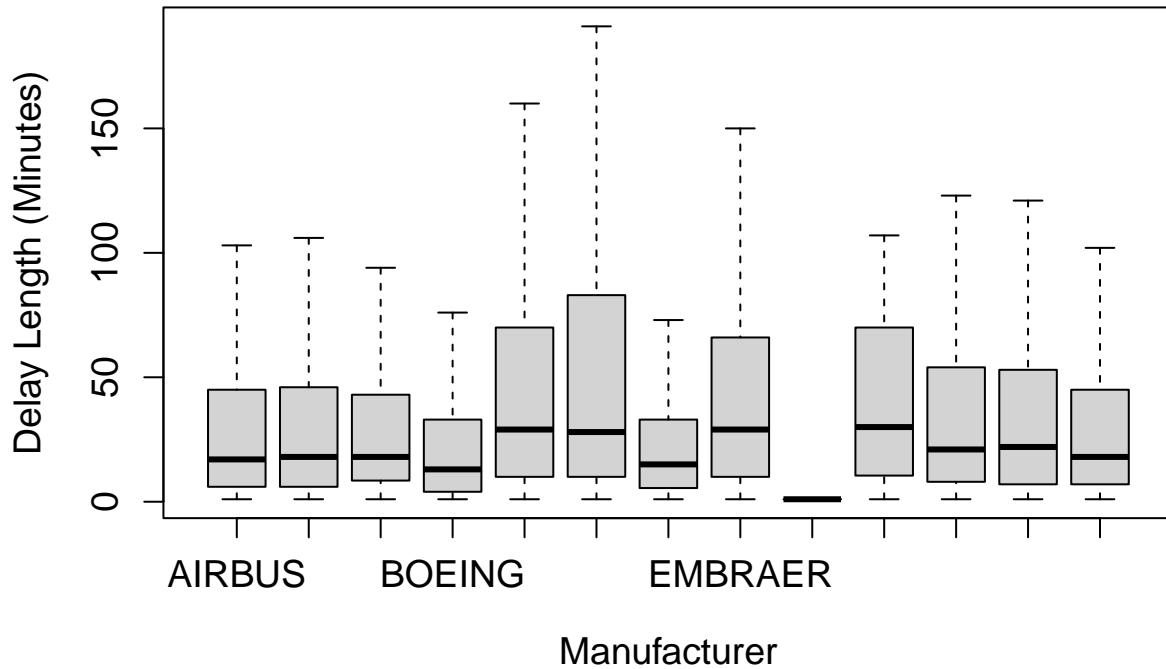
Potential Predictors from Planes Dataset

Looking at the documentation for planes, you can't merge flights with planes simply as American Airways and Envoy Air report their tail numbers in a different way than is reported in flights. As a result, we will remove these two airlines to allow us to look at variables in planes properly.

```
flightsPlanes = left_join(flights, planes,
  by = "tailnum")
flightsPlanes = flightsPlanes[-which(flightsPlanes$carrier %in%
  c("AA", "MQ")), ]

boxplot(flightsPlanes$dep_delay[which(flightsPlanes$dep_delay >
  0)] ~ flightsPlanes$manufacturer[which(flightsPlanes$dep_delay >
  0)], outline = F, xlab = "Manufacturer",
  ylab = "Delay Length (Minutes)", main = "Delay Length vs. Manufacturer",
  cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

Delay Length vs. Manufacturer



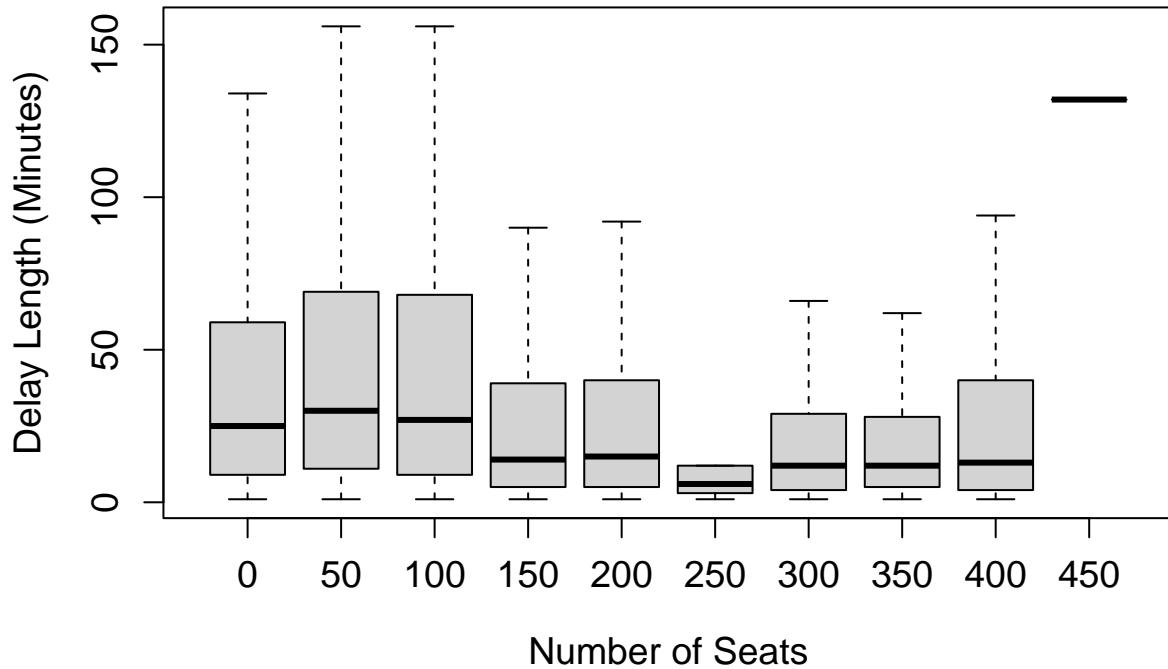
Some differences as usual, but difficult to say how significant. Will look into further.

```
table(round_any(flightsPlanes$seats[which(flightsPlanes$dep_delay > 20)], 50, f = round))

##
##      0      50     100     150     200     250     300     350     400     450
##  4406 13290   5743   9265 18530       3    243    243    362       1

boxplot(flightsPlanes$dep_delay[which(flightsPlanes$dep_delay > 0)] ~ round_any(flightsPlanes$seats[which(flightsPlanes$dep_delay > 0)], 50, f = round), outline = F, xlab = "Number of Seats",
ylab = "Delay Length (Minutes)", main = "Delay Length vs. Seats",
cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

Delay Length vs. Seats



Based on the table and the graph together, while some seat numbers don't have very many observations for them, it does seem like there could be a difference in delay between planes with fewer than 150 seats and between 150-250 seats.

```
table(flightsPlanes$engine)
```

```
##  
##        4 Cycle Reciprocating      Turbo-fan      Turbo-jet      Turbo-shaft  
##            3             543          231780         40387          286
```

Frankly, nearly every plane has a Turbo-fan engine, so there is not much to look at.

Modeling

- Goal: Create a model to help decide which variables are most useful and maybe get a hierarchy within them.

Checking Assumptions

Logistic regression has less assumptions than Linear regression. It requires a binary response, which we have with our ‘sigDelay’ variable. Also, there cannot be multicollinearity. Earlier from the correlation matrix, we saw that dew point and temperature had high correlation which could be an indication of multicollinearity. So dew point was chosen as the variable to add to the model because it had visually had the strongest relationship with the response. Finally, our sample size is still very large even after eliminating rows for missing data and removing variables.

```
wd = wf[, c(20, 26, 27, 29:32)]
sample <- sample(c(TRUE, FALSE), nrow(wd), replace = TRUE, prob = c(0.7,
  0.3))
test = wd[sample, ]
train = wd[!sample, ]
test_x = test[, -1]
test_y = test$sigDelay

weather.model = glm(sigDelay ~ ., data = train, family = "binomial")

summary(weather.model)

##
## Call:
## glm(formula = sigDelay ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q        Median        3Q        Max 
## -1.5980   -0.7048   -0.6296   -0.5181    2.2629 
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) 25.0239100  1.3047980 19.178 < 2e-16 ***
## dewp         0.0058146  0.0005364 10.841 < 2e-16 ***
## humid        0.0044429  0.0006232  7.129 1.01e-12 ***
## wind_speed   0.0234981  0.0016831 13.961 < 2e-16 ***
## precip       3.4997427  0.6023654  5.810 6.25e-09 ***
## pressure     -0.0263921  0.0012710 -20.765 < 2e-16 ***
## visib        -0.0319282  0.0064229 -4.971 6.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 88029  on 87196  degrees of freedom
## Residual deviance: 86454  on 87190  degrees of freedom
##   (10280 observations deleted due to missingness)
## AIC: 86468
```

```

## Number of Fisher Scoring iterations: 4

full_pred = predict(weather.model, newdata = test_x, type = "response")
delay_preds = rep(FALSE, length(full_pred))
delay_preds[full_pred > 0.4] = TRUE
print("TEST RESULTS")

## [1] "TEST RESULTS"

table(delay_preds, test_y)

##          test_y
## delay_preds FALSE  TRUE
##        FALSE 178974 48449
##        TRUE     553   366

mean(delay_preds == test_y)

## [1] 0.7854008

```

Model Results

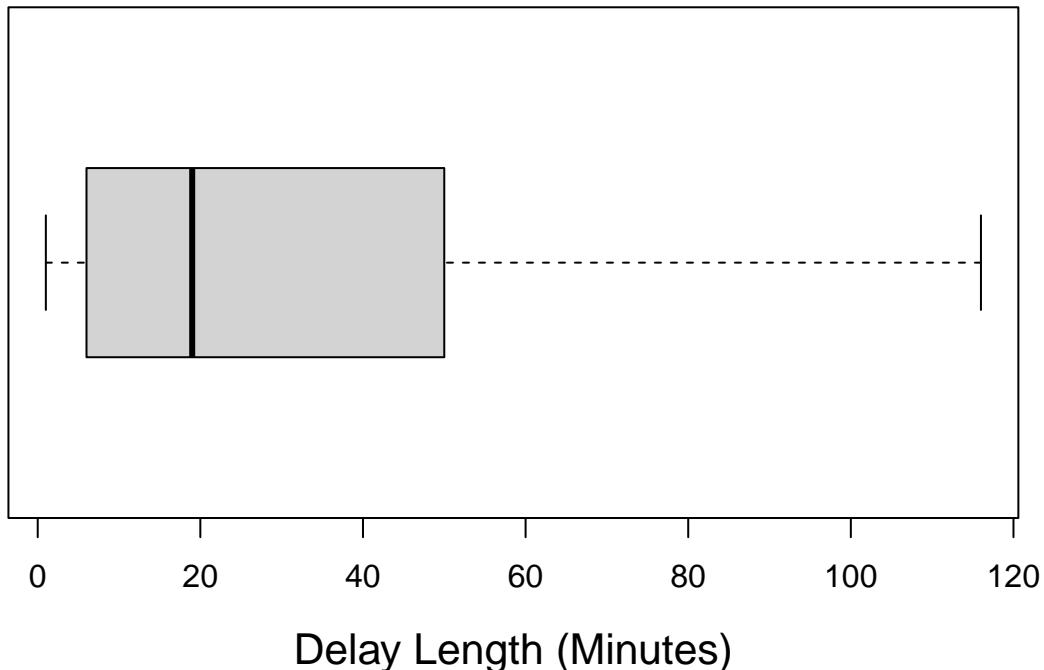
Our model shows that dew point, wind speed, and pressure seem to be the most power predictors of a delay occurring. Variables such as precipitation and visibility should be taken with a grain of salt because they have very skewed distributions of data.

Using the model to predict on training data resulted in a 78% accuracy rate. At first glance this seems like a positive, however it overwhelmingly predicts flights to not be delayed. In fact, ~80% of flights were not delayed in this data set so always predicting no delay results in ~ 80% accuracy anyway. Therefore, this model is not ideal for prediction and should just be used to determine useful variables.

The Dependent Variable

```
boxplot(flights$dep_delay[flights$dep_delay > 0], outline = F, xlab = "Delay Length (Minutes)",  
main = "Distribution of Delay Times in Minutes",  
cex.lab = 1.3, cex.main = 1.4, horizontal = T)
```

Distribution of Delay Times in Minutes



```
mean(flights$dep_delay[which(flights$dep_delay > 0)])
```

```
## [1] 39.37323
```

Looking at the distribution of delays by itself (omitting outliers for formatting, though they are still used in calculation), it seems quite skewed. 75% of delays are under 45 minutes, and 25% are just 5 minutes or less. Additionally, despite the median being around 20, the mean is 40, which is closer to the 3rd quartile than the median.

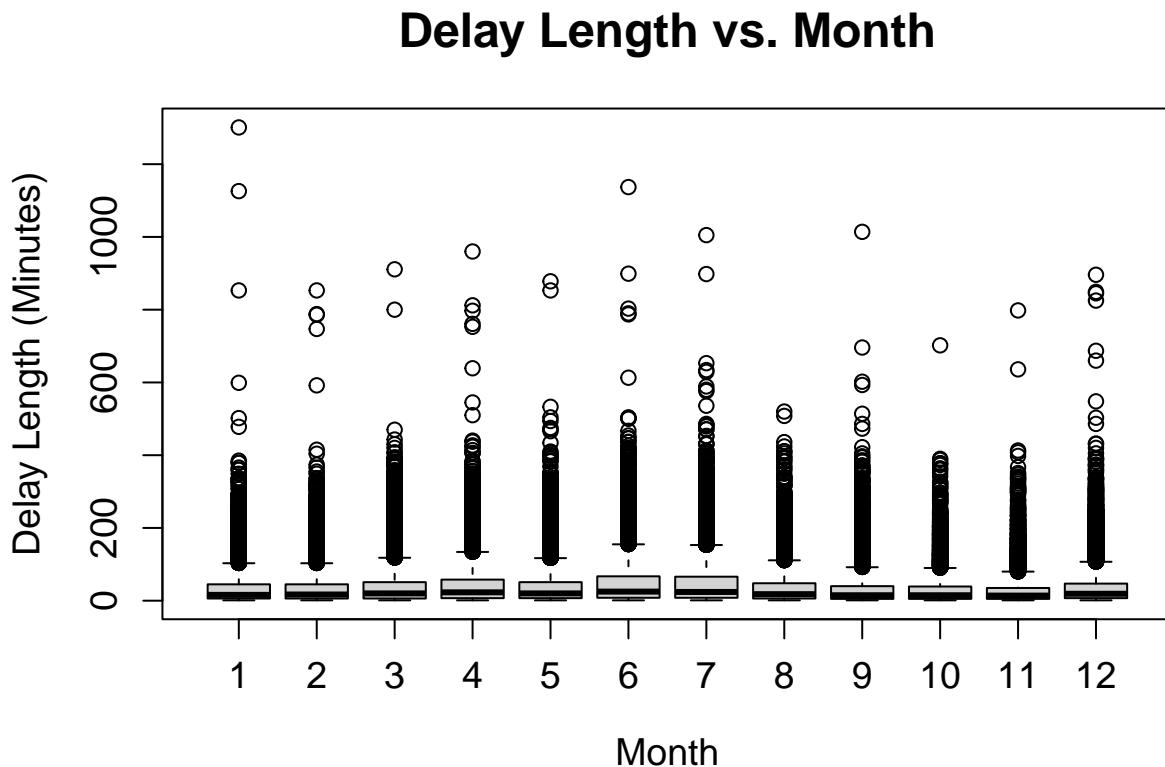
Considering the minor distinction 10 minutes makes in a delay and the relative lack of variance in this distribution, it may be unreasonable to try and model delay in minutes. Perhaps a binary dependent variable would be better.

Potential Predictors From Flights Dataset

By our understanding, each observation is a record of a single flight. This would imply that variables such as arrival delay would be the delay of a plane arriving at the destination after it departed, not the arrival of

the same plane but from the previous flight. Thus, we think arr_time, arr_delay, and air_time don't make sense to include at all, since they cannot possibly be used to predict departure delays.

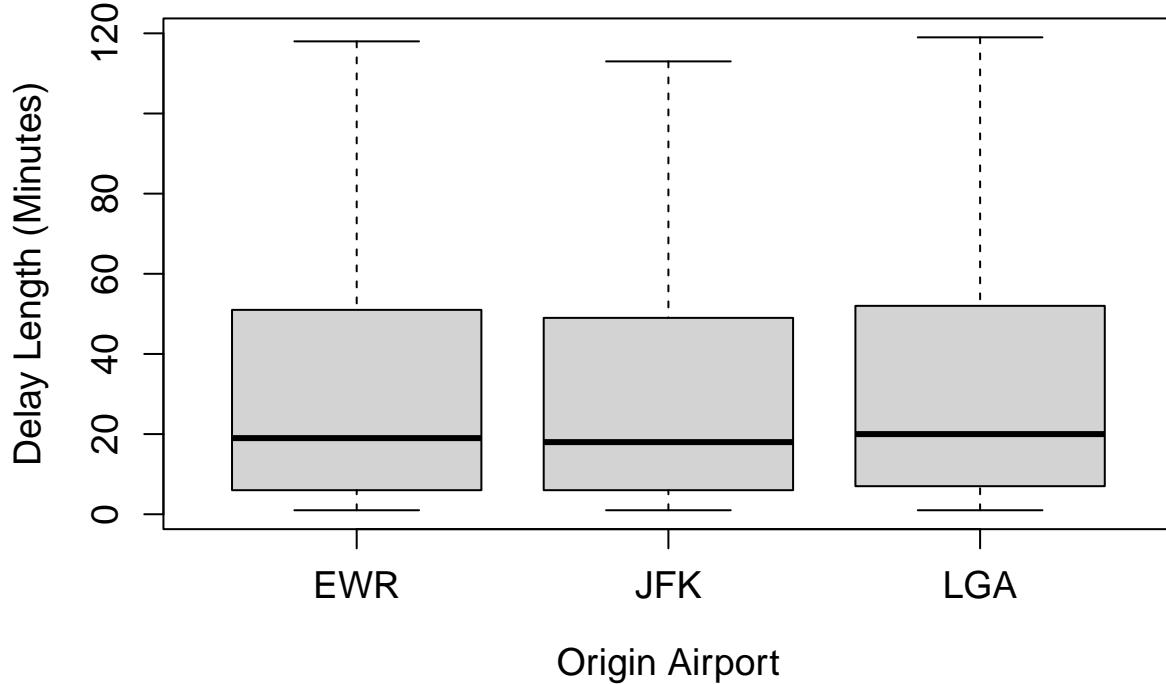
```
boxplot(flights$dep_delay[flights$dep_delay >
  0] ~ flights$month[flights$dep_delay >
  0], xlab = "Month", ylab = "Delay Length (Minutes)",
  main = "Delay Length vs. Month", cex.lab = 1.2,
  cex.main = 1.5, cex.axis = 1.2)
```



At an initial glance, it seems like month will be a helpful variable. There is a fairly clear upward trend from the beginning of year into summer which peaks around July, and then falls again until December. This is intuitive, as one would expect more traffic and thus more delays in the summer and holidays.

```
boxplot(flights$dep_delay[which(flights$dep_delay >
  0)] ~ flights$origin[which(flights$dep_delay >
  0)], outline = F, xlab = "Origin Airport",
  ylab = "Delay Length (Minutes)", main = "Delay Length vs. Origin",
  cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

Delay Length vs. Origin



It doesn't seem like origin matters much at all, all three airports are nearly the exact same in delay time. This does make sense to us as the three airports are relatively close, which means if one produced worse delays than the others, many people would be able to easily go to one of the others instead.

```
boxplot(flights$dep_delay[which(flights$dep_delay > 0)] ~ flights$dest[which(flights$dep_delay > 0)], outline = F, xlab = "Destination Airport",
ylab = "Delay Length (Minutes)", main = "Delay Length vs. Destination",
cex.lab = 1.2, cex.main = 1.5, cex.axis = 1.2)
```

Model for Variable Selection

Cleaning data

```
# Keep only carriers, models, and
# destinations that have over 1% of
# flights
carrierNames = names(table(flightsPlanes$carrier))[(table(flightsPlanes$carrier)/nrow(flightsPlanes)) >
  0.01]/nrow(flightsPlanes)

flightsPlanes = flightsPlanes[flightsPlanes$carrier %in%
  carrierNames, ]

modelNames = names(table(flightsPlanes$model))[(table(flightsPlanes$model)/nrow(flightsPlanes)) >
  0.01]/nrow(flightsPlanes)

flightsPlanes = flightsPlanes[flightsPlanes$model %in%
  modelNames, ]

destNames = names(table(flightsPlanes$dest))[(table(flightsPlanes$dest)/nrow(flightsPlanes)) >
  0.01]/nrow(flightsPlanes)

flightsPlanes = flightsPlanes[flightsPlanes$dest %in%
  destNames, ]

# Creating binary variable
flightsPlanes$sigDelay = ifelse(flightsPlanes$dep_delay >=
  15, 1, 0)

# For modeling purposes, month should
# not be continuous
flightsPlanes$month <- as.factor(flightsPlanes$month)

# Check for NAs and remove
sum(is.na(flightsPlanes)) # 219496, that's a lot

## [1] 219496

# Many have multiple columns with all
# NAs, we can try to remove those rows
sum(is.na(flightsPlanes[26])) # Almost all entries in speed

## [1] 200699

# Speed column basically all NA, remove
# it
flightsPlanes = flightsPlanes %>%
  select(-"speed")

sum(is.na(flightsPlanes[6]))
```

```

## [1] 2373

# Some have no dependent variable, so
# might as well remove these
# observations
flightsPlanes = flightsPlanes[which(!is.na(flightsPlanes[6])),
]
# Many are in year
flightsPlanes = flightsPlanes[which(!is.na(flightsPlanes[20])),
]
# The rest I'll just remove ad hoc
flightsPlanes = flightsPlanes[which(!is.na(flightsPlanes[7])),
]
flightsPlanes = flightsPlanes[which(!is.na(flightsPlanes[9])),
]

```

Now we can remove variables that we considered unusable, unhelpful or too big (in the case of tailnum – it has too many levels to run anything in reasonable time).

```

flightsPlanes = flightsPlanes %>%
  select(-c(4, 6, 7, 9, 12, 13, 15, 16,
  18, 21, 24, 26))

```

Variable Selection With Flights and Planes

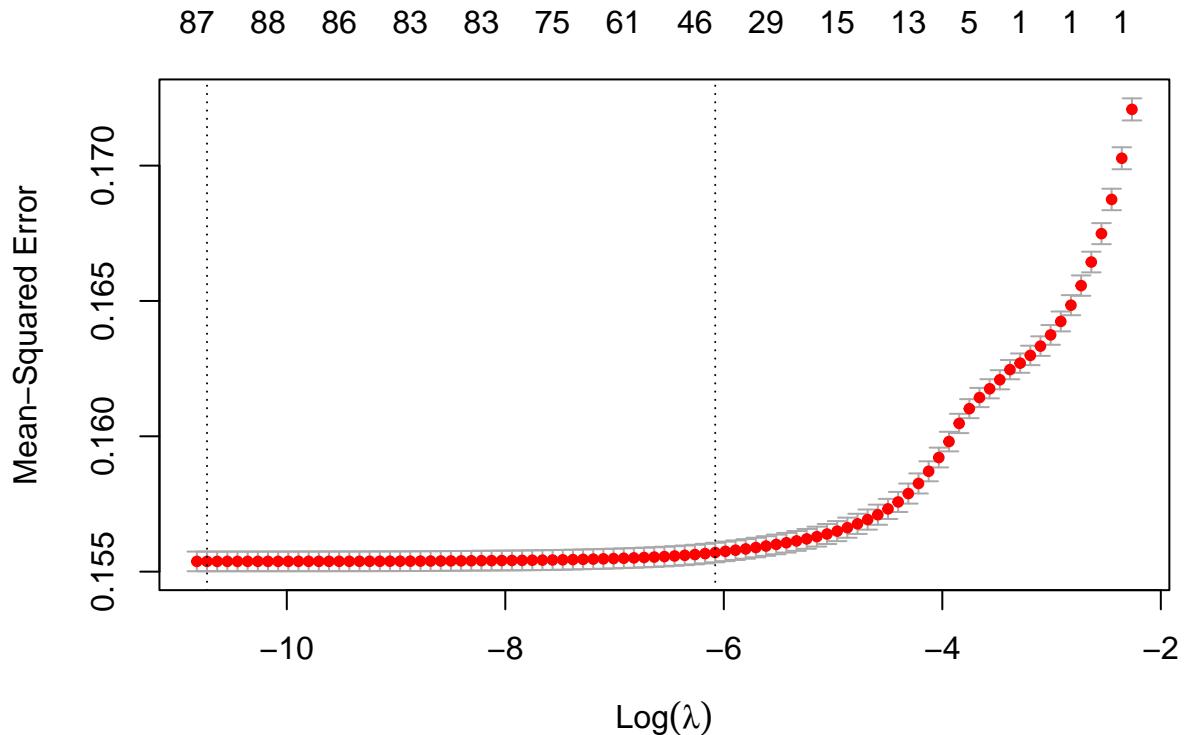
I will run an initial LASSO model to select which variables from the combination of the flights and planes datasets are important. Since many of these variables are factors, we would expect many to have coefficients of zero, making LASSO a reasonable option. We don't plan on doing inference, so the lack of a closed form solution for LASSO is not an issue for us.

```

xInit = model.matrix(sigDelay ~ ., flightsPlanes)[,
-1]
yInit = flightsPlanes$sigDelay

lasso.cvInit = cv.glmnet(xInit, yInit, alpha = 1)
plot(lasso.cvInit)

```



```

lambda.cvInit = lasso.cvInit$lambda.min
lambda.cvInit

fit.lassoInit = glmnet(xInit, yInit, alpha = 1,
                      lambda = lambda.cvInit)

# Let's check which variables seem
# important
CFInit = as.matrix(coef(fit.lassoInit, fit.lassoInit$lambda.cv))
CFInit[CFInit != 0, ][order(CFInit[CFInit != 0])]

# Month, carrier, model, destination,
# manufacturer, and hour Some variables
# have coefficients that aren't zero,
# but are below 0.001. In the interest
# of creating a very interpretable
# model, we will leave these out for
# the full model.

```

Final Model

Now that we have selected variables from flights, planes, and weather, we can create a final model to get the best interpretability.

```

# Pressure, wind speed, dew point

wf <- merge(flights, weather, by = c("time_hour",
  "origin"))

# Removing the unnecessary variables
# except for those needed to combine,
# just because combining everything
# took too much memory

wf = wf[c(4, 8, 12, 14, 15, 18, 25, 28, 31)]

wfp = left_join(wf, planes, by = "tailnum")

wfp = wfp[c(1, 2, 3, 5, 6, 7, 8, 9, 12, 13)]

# Remove AA and MQ because they don't
# join correctly
wfp = wfp[-which(wfp$carrier %in% c("AA",
  "MQ")), ]

# Keep only carriers, models, and
# destinations that have over 1% of
# flights
carrierNames = names(table(wfp$carrier))[(table(wfp$carrier)/nrow(wfp)) >
  0.01]/nrow(wfp))
wfp = wfp[wfp$carrier %in% carrierNames,
  ]

modelNames = names(table(wfp$model))[(table(wfp$model)/nrow(wfp)) >
  0.01]/nrow(wfp))
wfp = wfp[wfp$model %in% modelNames, ]

destNames = names(table(wfp$dest))[(table(wfp$dest)/nrow(wfp)) >
  0.01]/nrow(wfp))
wfp = wfp[wfp$dest %in% destNames, ]

wfp$sigDelay = ifelse(wfp$dep_delay >= 15,
  1, 0)
wfp$month.x <- as.factor(wfp$month.x)

# Remove NAs
colSums(is.na(wfp))

```

	month.x	dep_delay	carrier	dest	hour.x	dewp
##	0	2370	0	0	0	13
##	wind_speed	pressure	manufacturer	model	sigDelay	
##	60	22040	0	0	2370	

```

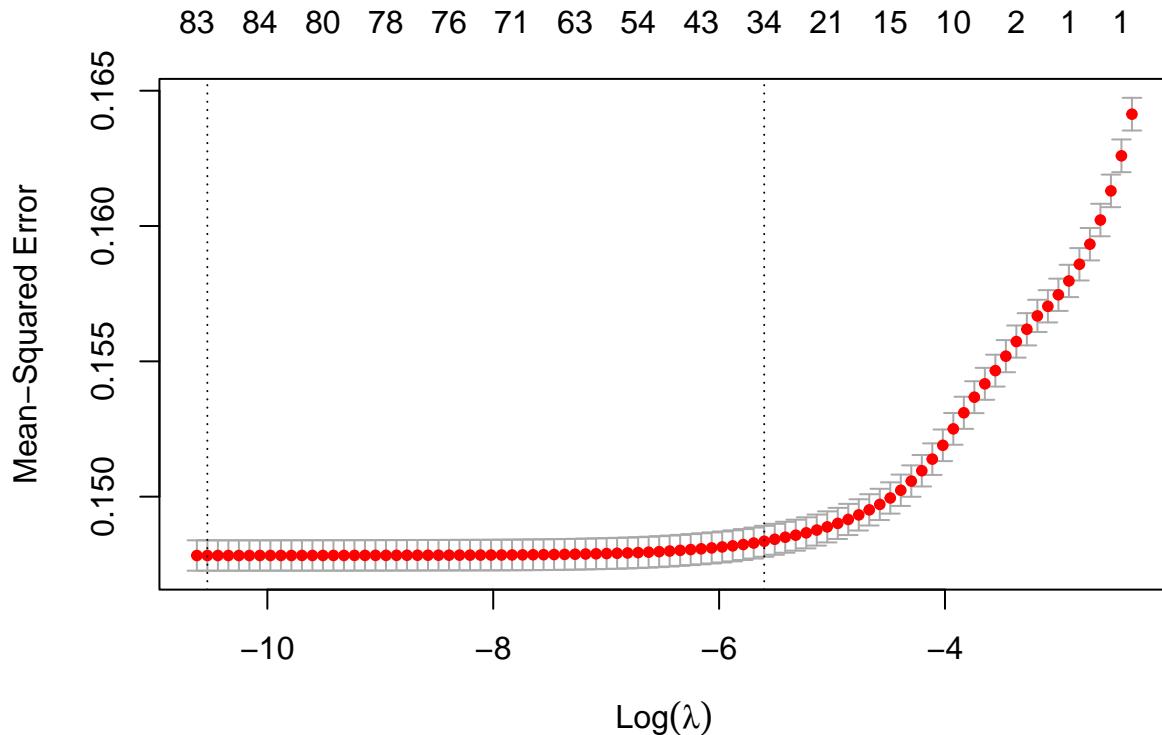
wfp = wfp[which(!is.na(wfp$dep_delay)), ]
wfp = wfp[which(!is.na(wfp$pressure)), ]
wfp = wfp[which(!is.na(wfp$wind_speed)), ]

```

```
]
```

Final Model

```
x = model.matrix(sigDelay ~ ., wfp[-2])[,  
  -1]  
y = wfp$sigDelay  
  
# LASSO  
lasso.cv = cv.glmnet(x, y, alpha = 1)  
plot(lasso.cv)
```

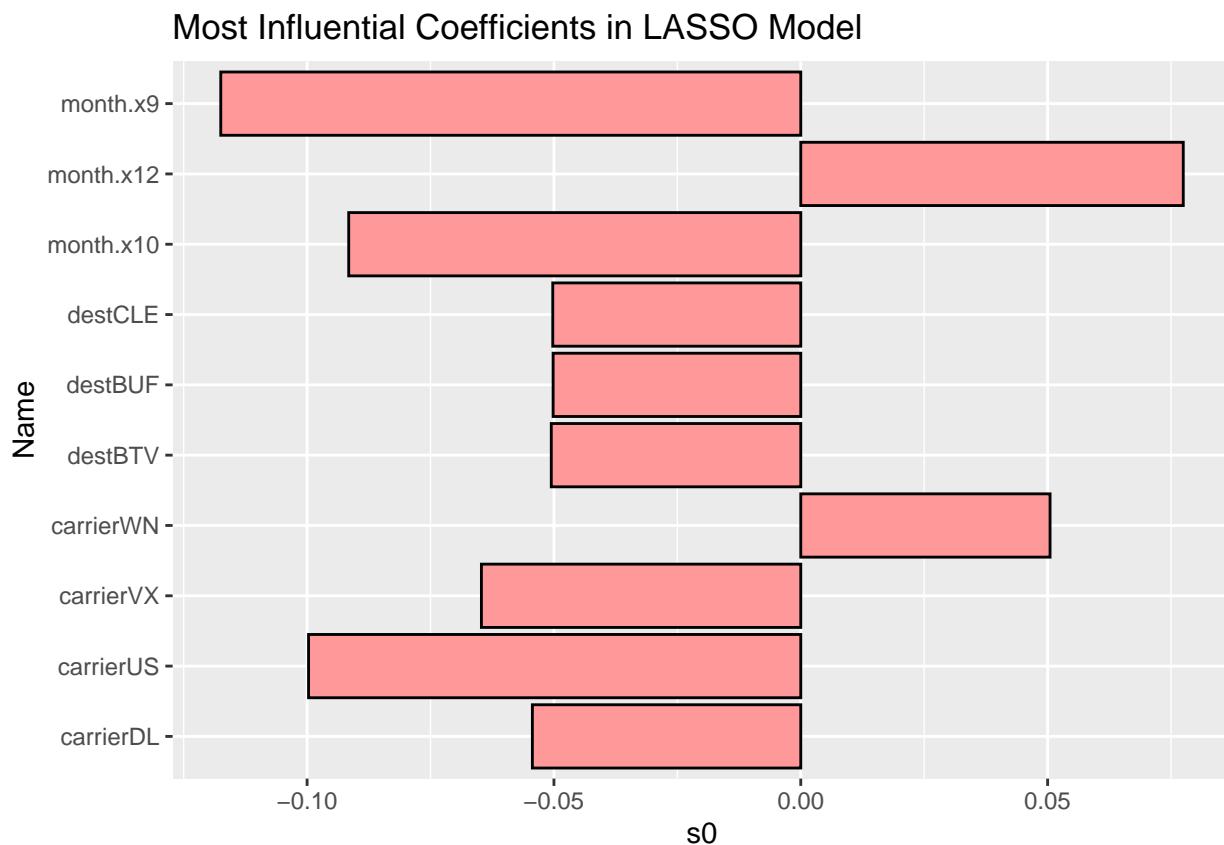


```
lambda.cv = lasso.cv$lambda.min  
lambda.cv  
  
fit.lasso = glmnet(x, y, alpha = 1, lambda = lambda.cv)  
  
CF = as.matrix(coef(fit.lasso, fit.lasso$lambda.cv))  
CF[CF != 0, ][order(CF[CF != 0])]  
  
# Plotting most influential predictors  
lasso_coef <- as.data.frame(as.matrix(coef(fit.lasso)))  
lasso_coef <- cbind(Name = rownames(lasso_coef),
```

```

    lasso_coef)
lasso_coef <- lasso_coef[-1, ]
o <- order(abs(lasso_coef$s0), decreasing = TRUE)
lasso_coef <- lasso_coef[o, ]
lasso_coef <- lasso_coef[c(1:10), ]
rownames(lasso_coef) <- 1:nrow(lasso_coef)
p <- ggplot(data = lasso_coef, aes(x = Name,
  y = s0)) + geom_bar(stat = "identity",
  fill = "#FF9999", colour = "black")
p + coord_flip() + ggtitle("Most Influential Coefficients in LASSO Model")

```



The Lasso model results show that the most influential positive variables to be Month 12 and carrierWN (Southwest Airlines) The most influential negative variables are Month 9, Month 10, and carrierUS