Report

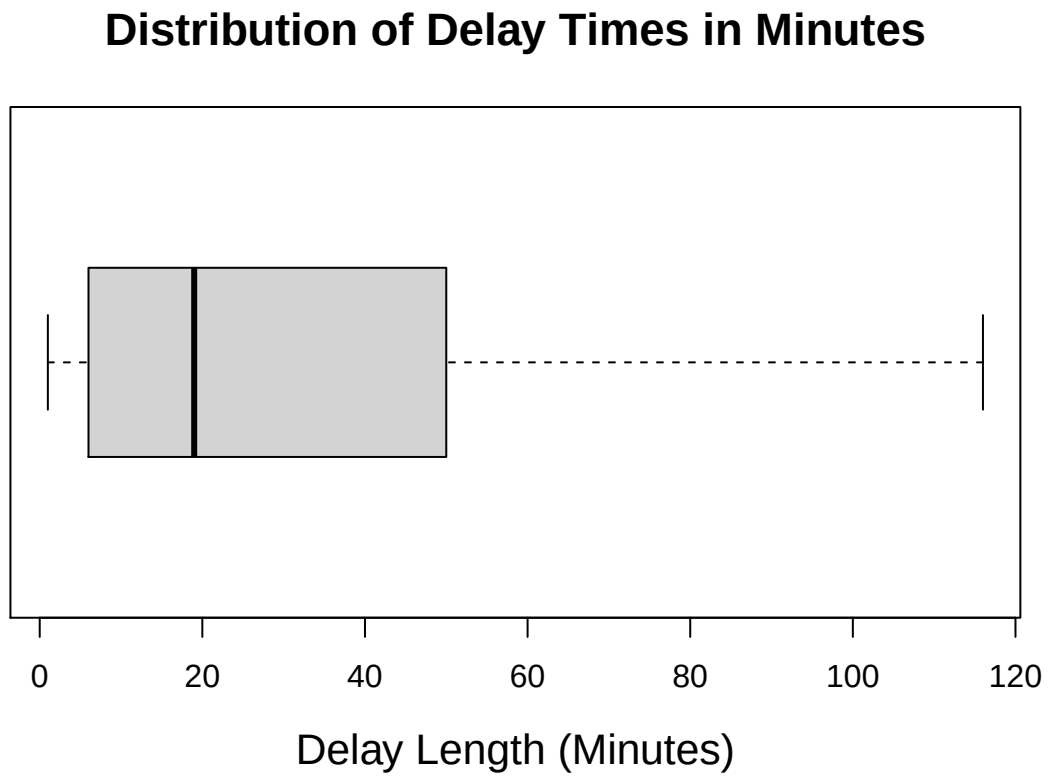
Cameron Lucas

1/22/2023

Exploratory Data Analysis

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)
```




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

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

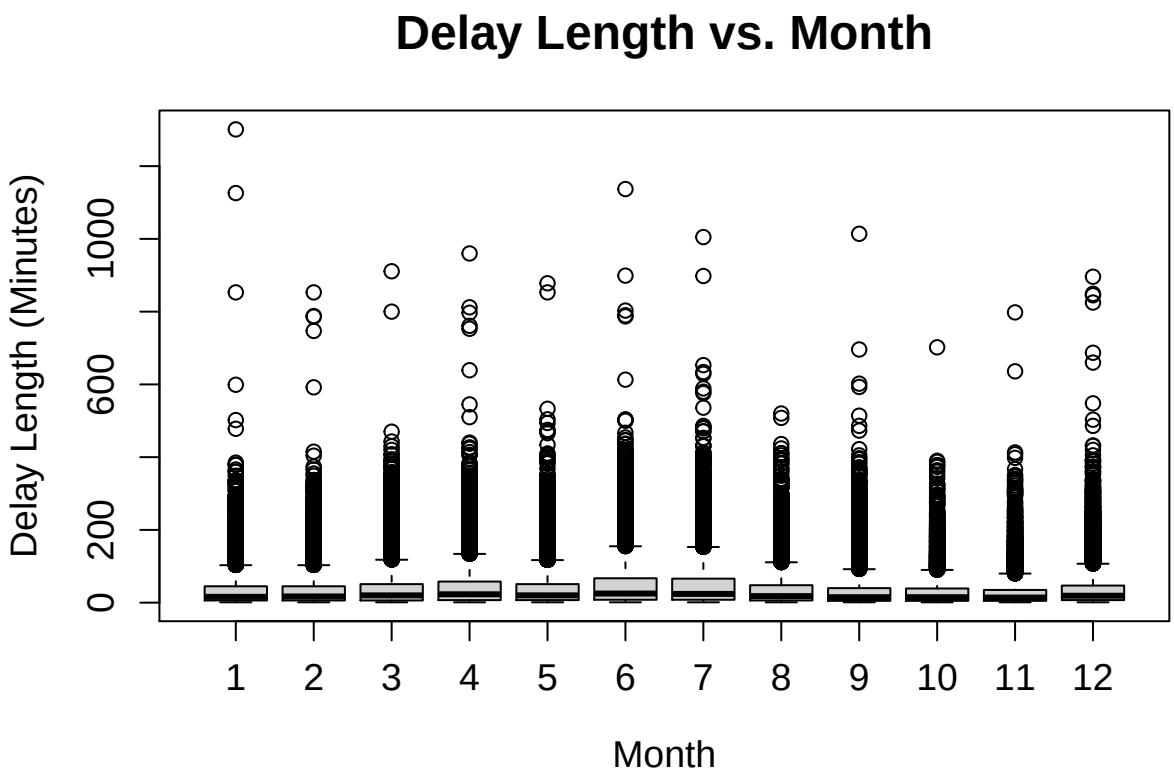
Looking at the distribution of delays by itself (omitting outliers for formatting, though they are still used in calculation), it seems quite skewed. 75% of delays are under 45 minutes, and 25% are just 5 minutes or less. Additionally, despite the median being around 20, the mean is 40, which is closer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

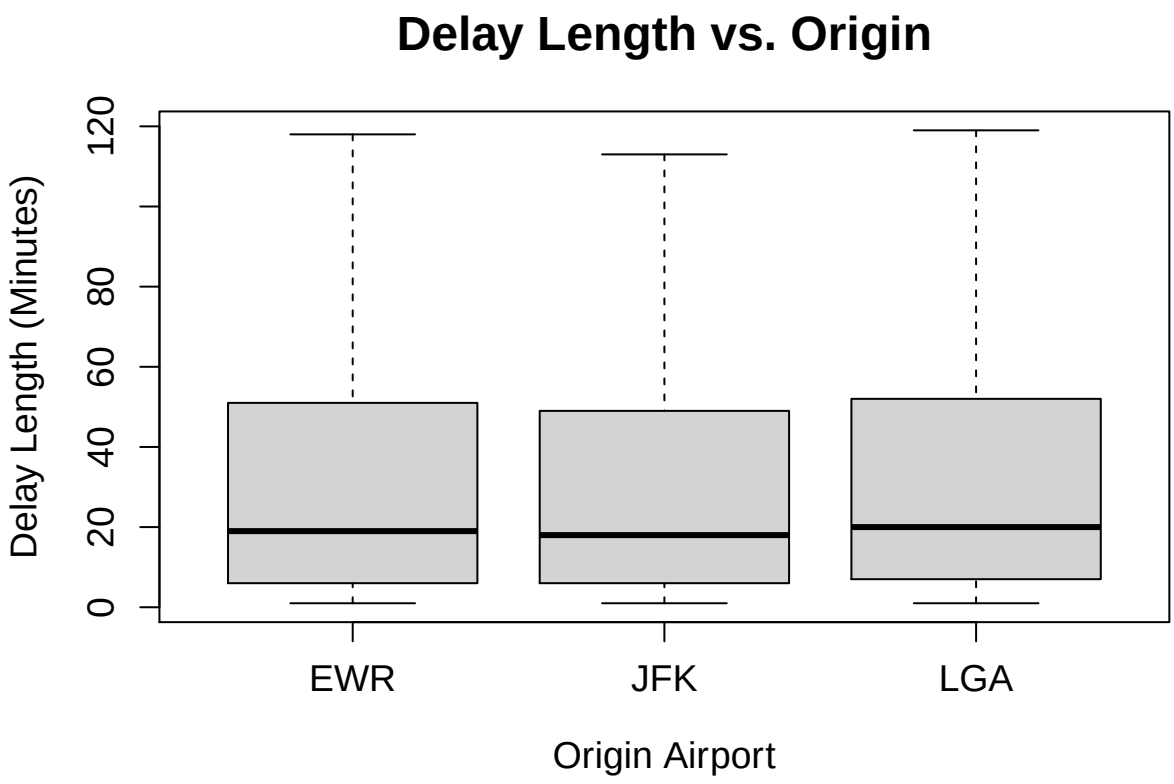
By our understanding, each observation is a record of a single flight. This would imply that variables such as arrival delay would be the delay of a plane arriving at the destination after it departed, not the arrival of the same plane but from the previous flight. Thus, we think arr_time, arr_delay, 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", yla
        main="Delay Length vs. Month", cex.lab=1.2, cex.main=1.5, cex.axis=1.2)
```



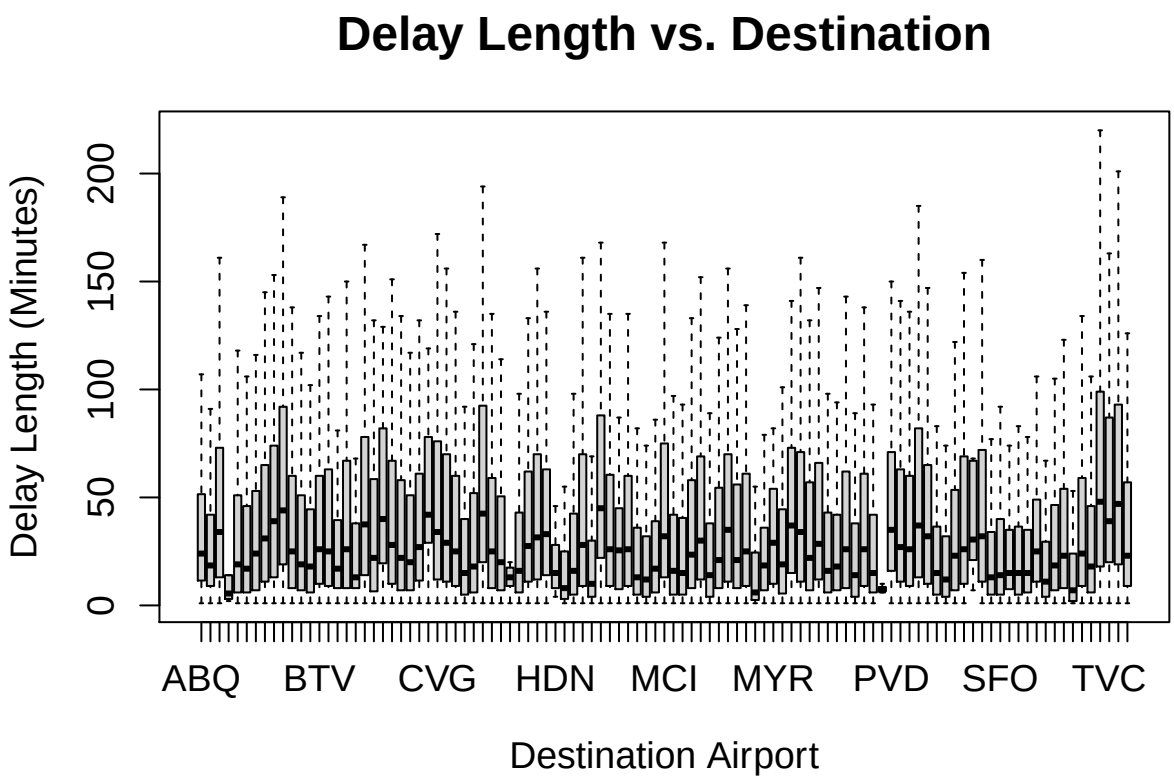
At an initial glance, it seems like month will be a helpful variable. There is a fairly clear upward trend from the beginning of year into summer which peaks around July, and then falls again until December. This is intuitive, as one would expect more traffic and thus more delays in the summer and holidays.

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



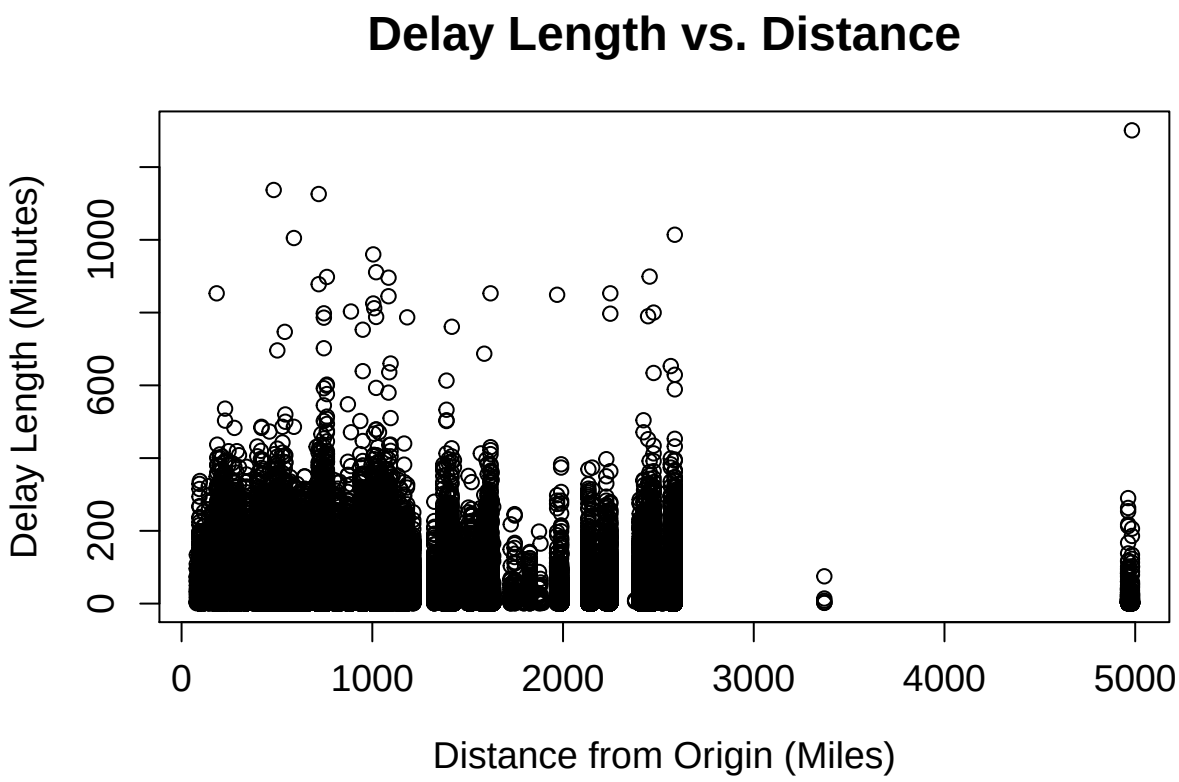
It doesn't seem like origin matters much at all, all three airports are nearly the exact same in delay time. Again this does make sense to us as the three airports are relatively close, which means if one produced worse delays than the others, many people would be able to easily go to one of the others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

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



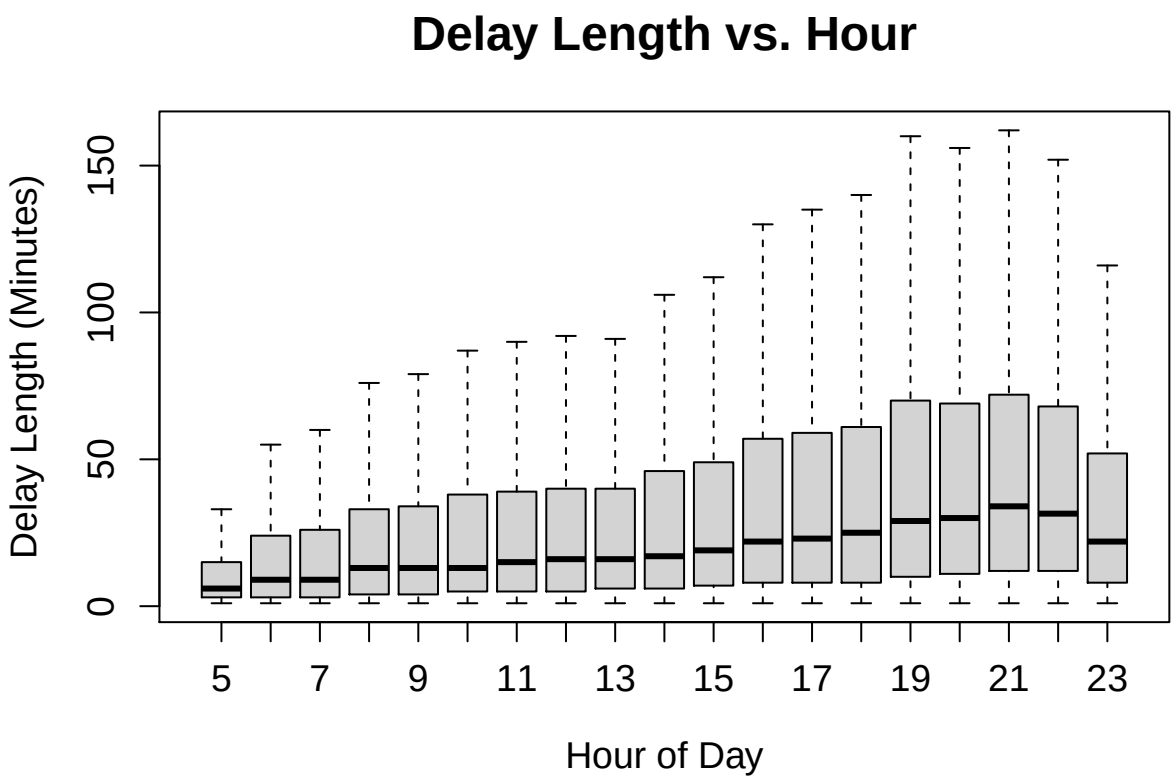
Frankly it's hard to tell which destinations are which, but it's enough to tell that there are clearly some with much better delay times than others, so this is definitely a variable we will want to use, at least during variable selection.

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



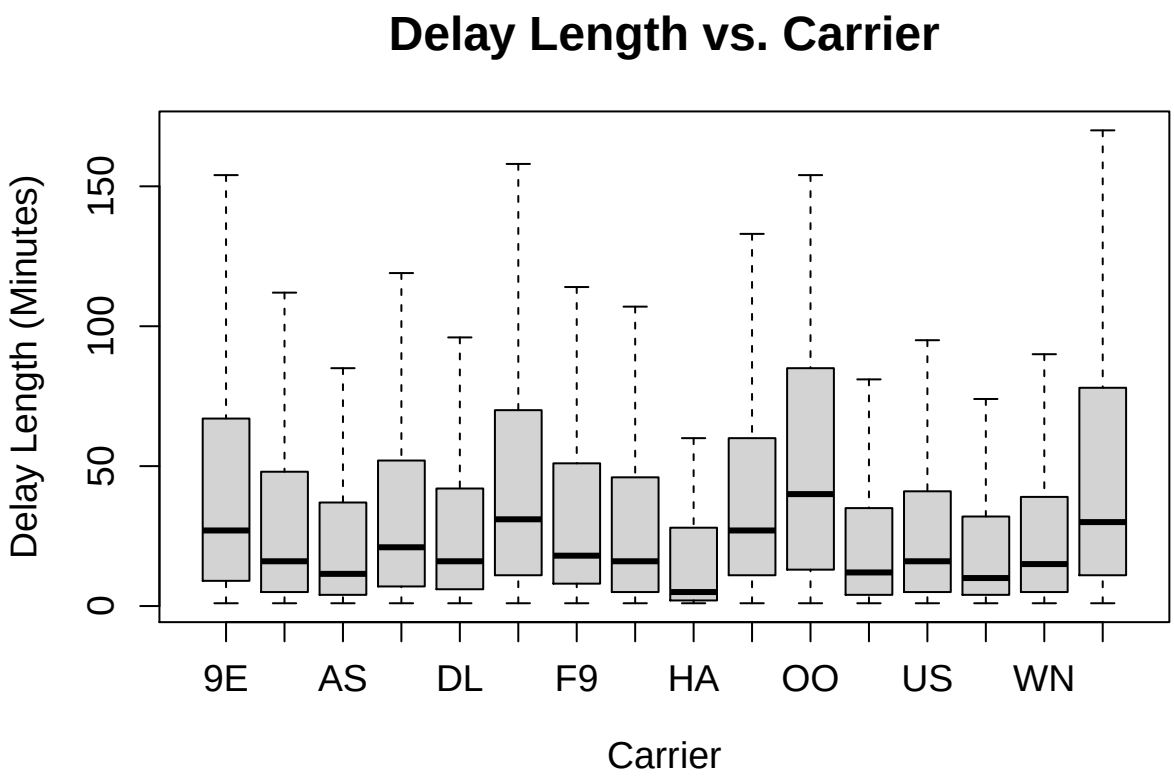
In our opinion, destination does not seem useful. The only real difference visually is at the ~3300 mile mark, but that has very few observations which makes it hard to say anything for sure.

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



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 Leng
        main="Delay Length vs. Carrier", cex.lab=1.2, cex.main=1.5, cex.axis=1.2)
```



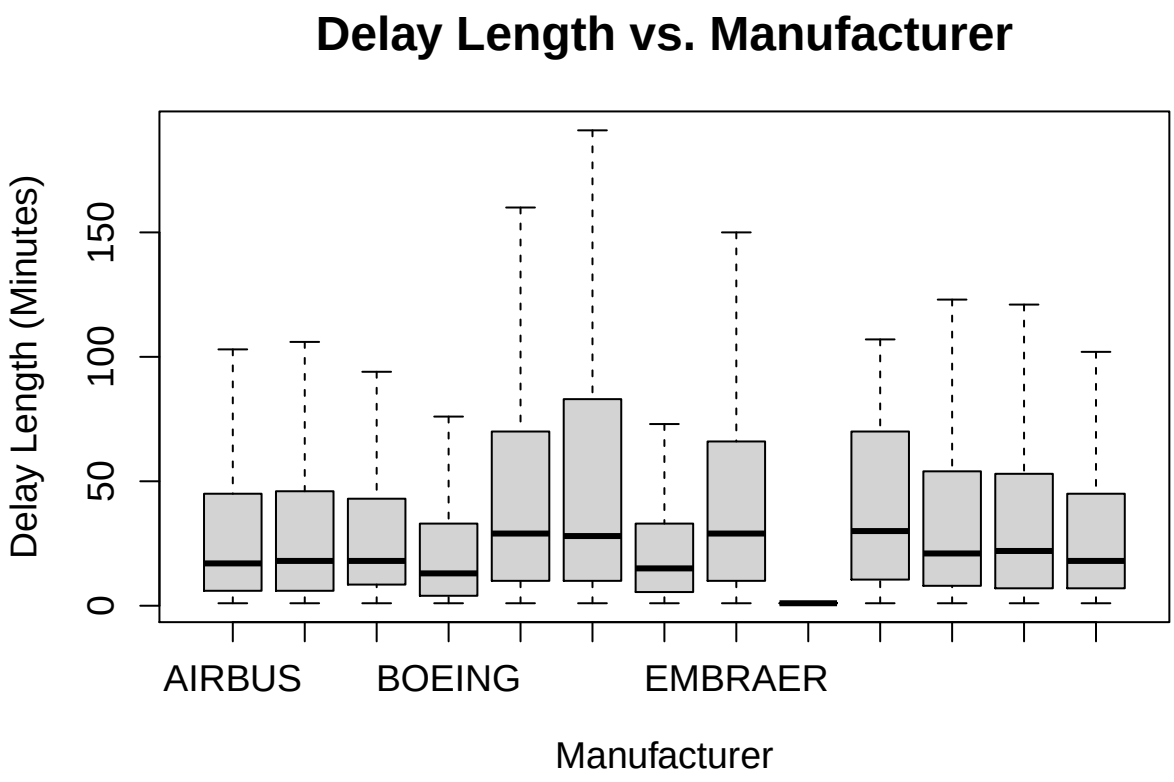
There does appear to be significant differences between carrier; we will continue with carrier into variable selection.

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 = "Manufactu
        main="Delay Length vs. Manufacturer", cex.lab=1.2, cex.main=1.5, cex.axis=1.2)
```

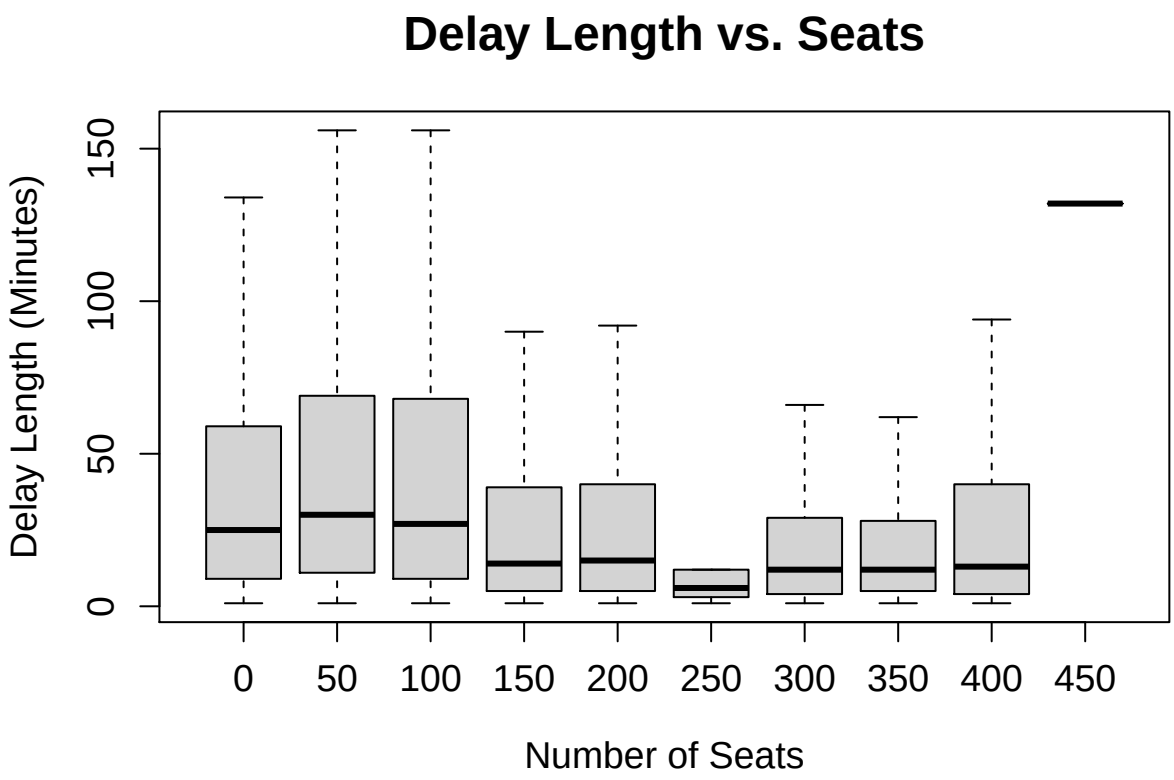


Some differences as usual, but difficult to say how significant. Will look into further.

```
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,  xl
```



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.

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

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

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

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

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

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])) # Some have no dependent variable, so might as well remove these observati
```

```
## [1] 2373
```

```
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).

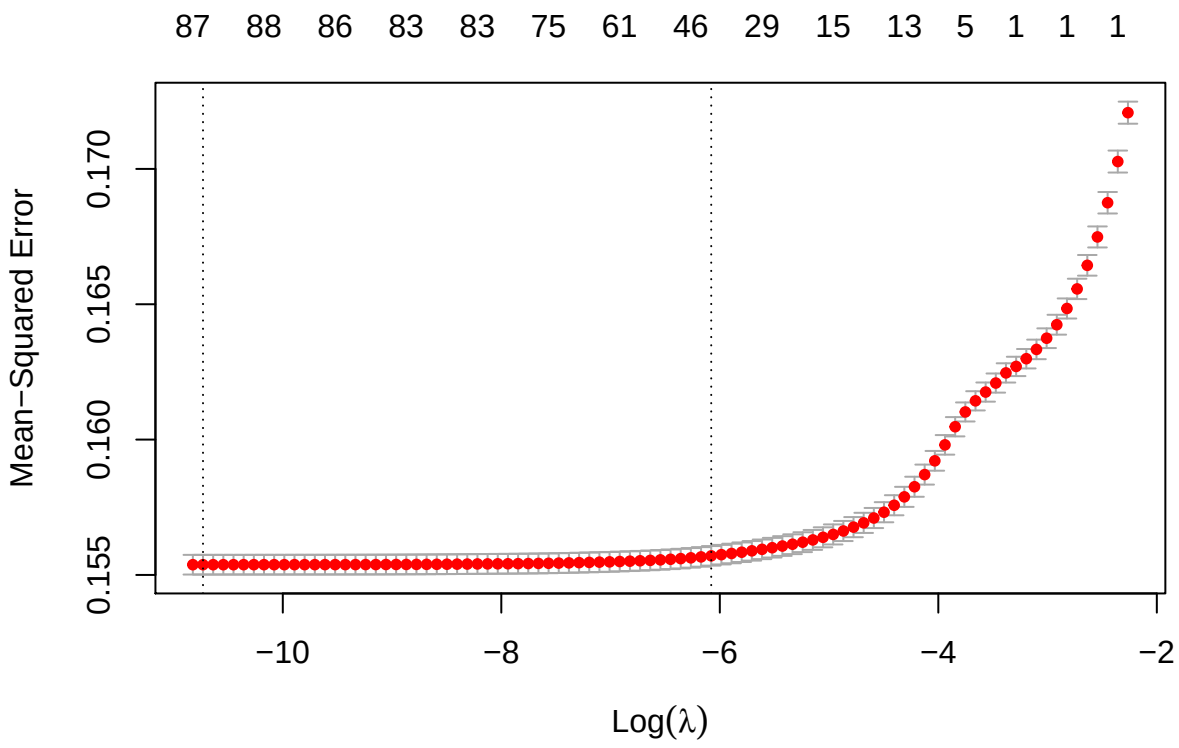
```
flightsPlanes = flightsPlanes %>% select(-c(4,6,7,9,12,13,15,16,18,21,24,26))
```

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 and 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
```

Final Model

#Irfan – this or before is where you can/should add Chris' stuff, since I'm using his weather choices now

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 everyth

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))
```

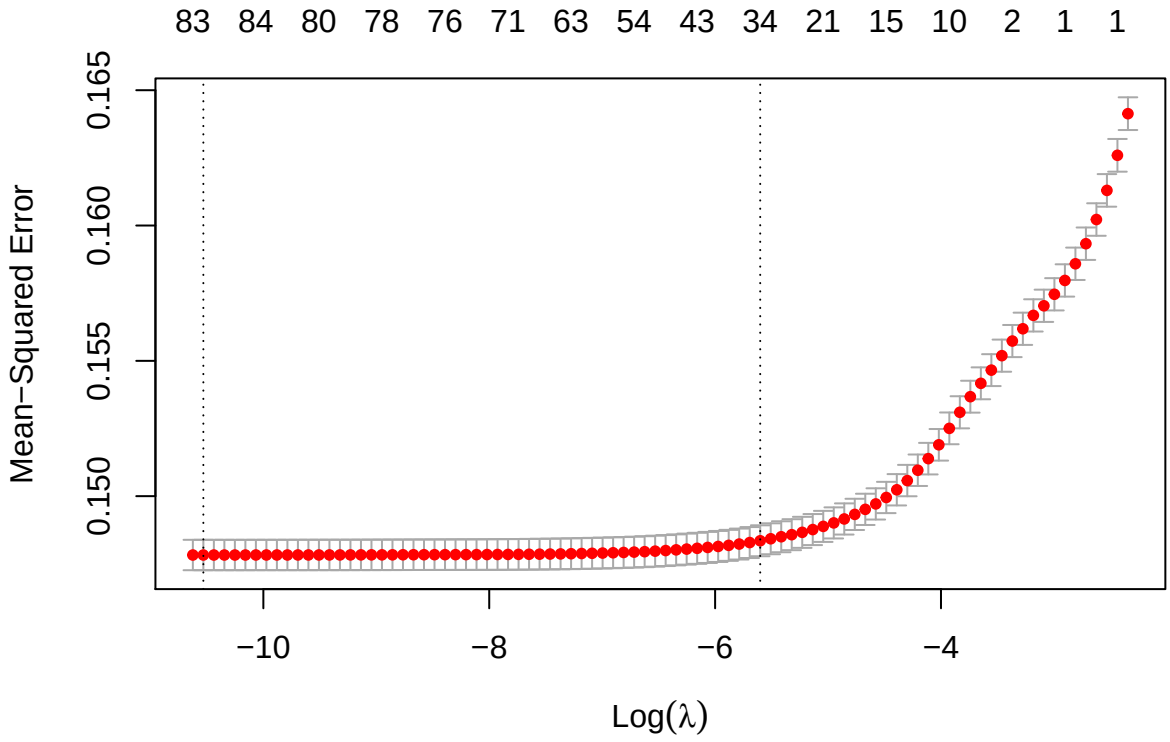
```
##      month.x    dep_delay      carrier         dest       hour.x         dewp
##            0         2370            0            0            0           13
##   wind_speed     pressure manufacturer        model     sigDelay
##           60        22040            0            0         2370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

```
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])]
```

#Irfan, please add whatever interpretation you think is necessary