

# STA 325 Final Project Code

Calleigh Smith, Hannah Bogomilsky, Hugh Esterson, Maria Henriquez, Mariana Izon

November 23, 2020

```
library(readr)
library(dplyr)
library(tidyverse)
library(gridExtra)
library(mgcv)
library(patchwork)
library(MASS)
library(EnvStats)
library(tree)
library(randomForest)
library(gbm)
library(bst)
library(plyr)
library(caret)
```

## Data Load-In and Initial Cleaning

```
# read data
flights <- read_csv("data/flights.csv")

# find unique airlines, destinations, and types of delays
unique(flights$OP_CARRIER)

## [1] "AA" "DL" "B6" "AS"

unique(flights$DEST)

## [1] "LAX" "SFO" "SJC" "SAN" "PSP" "SMF" "OAK" "LGB" "ONT" "BUR"

# mutate delays and filter out NA arrival delays
flights <- flights %>%
  mutate(CARRIER_DELAY = case_when(CARRIER_DELAY > 0 ~ 1,
                                     TRUE ~ 0),
         WEATHER_DELAY = case_when(WEATHER_DELAY > 0 ~ 1,
                                    TRUE ~ 0),
         NAS_DELAY = case_when(NAS_DELAY > 0 ~ 1,
                                TRUE ~ 0),
         SECURITY_DELAY = case_when(SECURITY_DELAY > 0 ~ 1,
                                    TRUE ~ 0),
         LATE_AIRCRAFT_DELAY = case_when(LATE_AIRCRAFT_DELAY > 0 ~ 1,
                                           TRUE ~ 0)) %>%
  filter(!is.na(ARR_DELAY))
```

```
# glimpse data
flights
```

```
## # A tibble: 2,033 x 34
##   YEAR MONTH DAY_OF_MONTH DAY_OF_WEEK FL_DATE   OP_CARRIER TAIL_NUM
##   <dbl> <dbl>         <dbl>         <dbl> <date>     <chr>      <chr>
## 1  2020     1             1             3 2020-01-01 AA        N110AN
## 2  2020     1             2             4 2020-01-02 AA        N111ZM
## 3  2020     1             3             5 2020-01-03 AA        N108NN
## 4  2020     1             4             6 2020-01-04 AA        N102NN
## 5  2020     1             5             7 2020-01-05 AA        N113AN
## 6  2020     1             6             1 2020-01-06 AA        N103NN
## 7  2020     1             7             2 2020-01-07 AA        N113AN
## 8  2020     1             8             3 2020-01-08 AA        N106NN
## 9  2020     1             9             4 2020-01-09 AA        N102NN
## 10 2020     1            10             5 2020-01-10 AA        N117AN
## # ... with 2,023 more rows, and 27 more variables: OP_CARRIER_FL_NUM <dbl>,
## #   ORIGIN <chr>, ORIGIN_CITY_NAME <chr>, DEST <chr>, DEST_CITY_NAME <chr>,
## #   CRS_DEP_TIME <dbl>, DEP_TIME <dbl>, DEP_DELAY <dbl>, TAXI_OUT <dbl>,
## #   WHEELS_OFF <dbl>, WHEELS_ON <dbl>, TAXI_IN <dbl>, CRS_ARR_TIME <dbl>,
## #   ARR_TIME <dbl>, ARR_DELAY <dbl>, CANCELLED <dbl>, CANCELLATION_CODE <lgl>,
## #   DIVERTED <dbl>, CRS_ELAPSED_TIME <dbl>, ACTUAL_ELAPSED_TIME <dbl>,
## #   AIR_TIME <dbl>, DISTANCE <dbl>, CARRIER_DELAY <dbl>, WEATHER_DELAY <dbl>,
## #   NAS_DELAY <dbl>, SECURITY_DELAY <dbl>, LATE_AIRCRAFT_DELAY <dbl>
```

# Exploratory Data Analysis

## Individual Predictor Variable EDA

### Taxi Histograms

```
# plot untransformed predictor taxi_in
pTAXI_IN <- ggplot(data = flights, aes(x = TAXI_IN)) +
  geom_histogram(binwidth = 5, fill = "#FFFF00", color = "#002D72", alpha = .7) +
  labs(x = "Time to Taxi In",
       y = "Frequency",
       title = "Histogram of TAXI_IN") +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

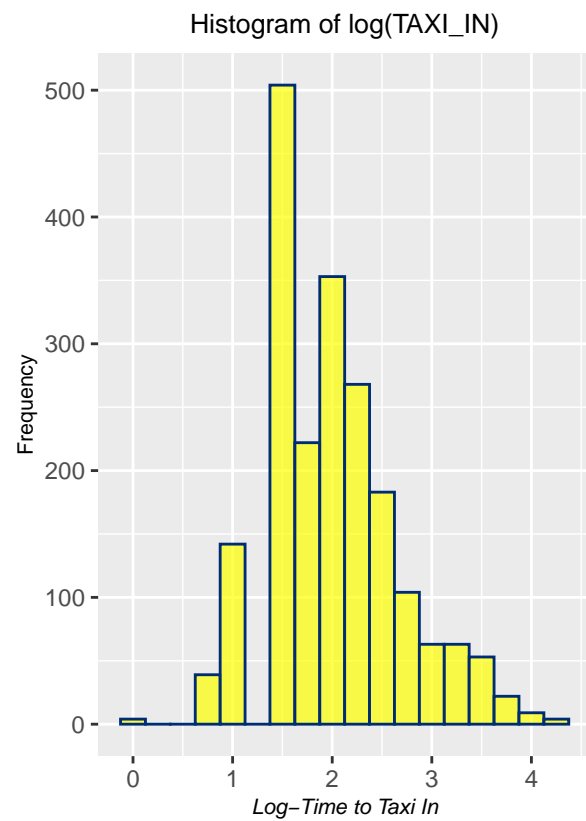
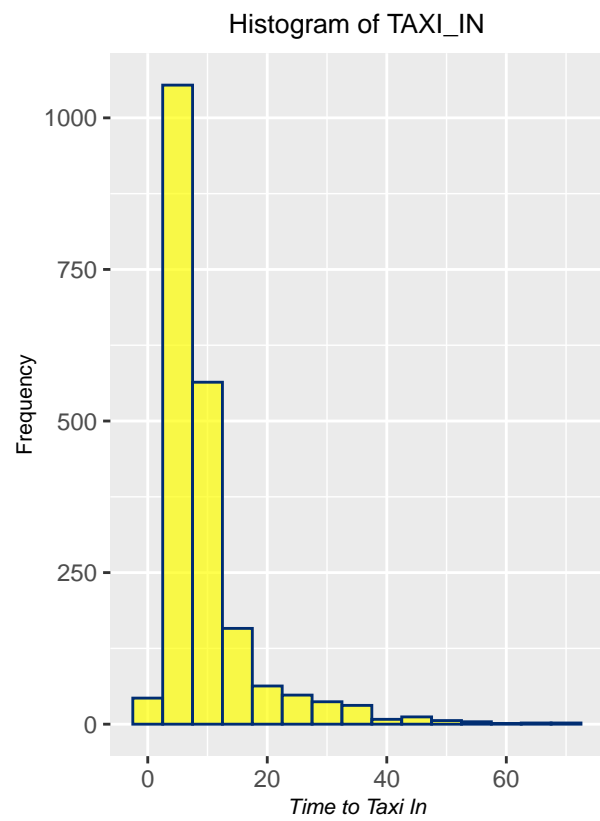
# plot untransformed predictor taxi_out
pTAXI_OUT <- ggplot(data = flights, aes(x = TAXI_OUT)) +
  geom_histogram(binwidth = 5, fill = "#FFFF00", color = "#002D72", alpha = .7) +
  labs(x = "Time to Taxi Out",
       y = "Frequency",
       title = "Histogram of TAXI_OUT") +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# log transform taxi_in and taxi_out
flights$log_TAXI_OUT <- log(flights$TAXI_OUT)
flights$log_TAXI_IN <- log(flights$TAXI_IN)

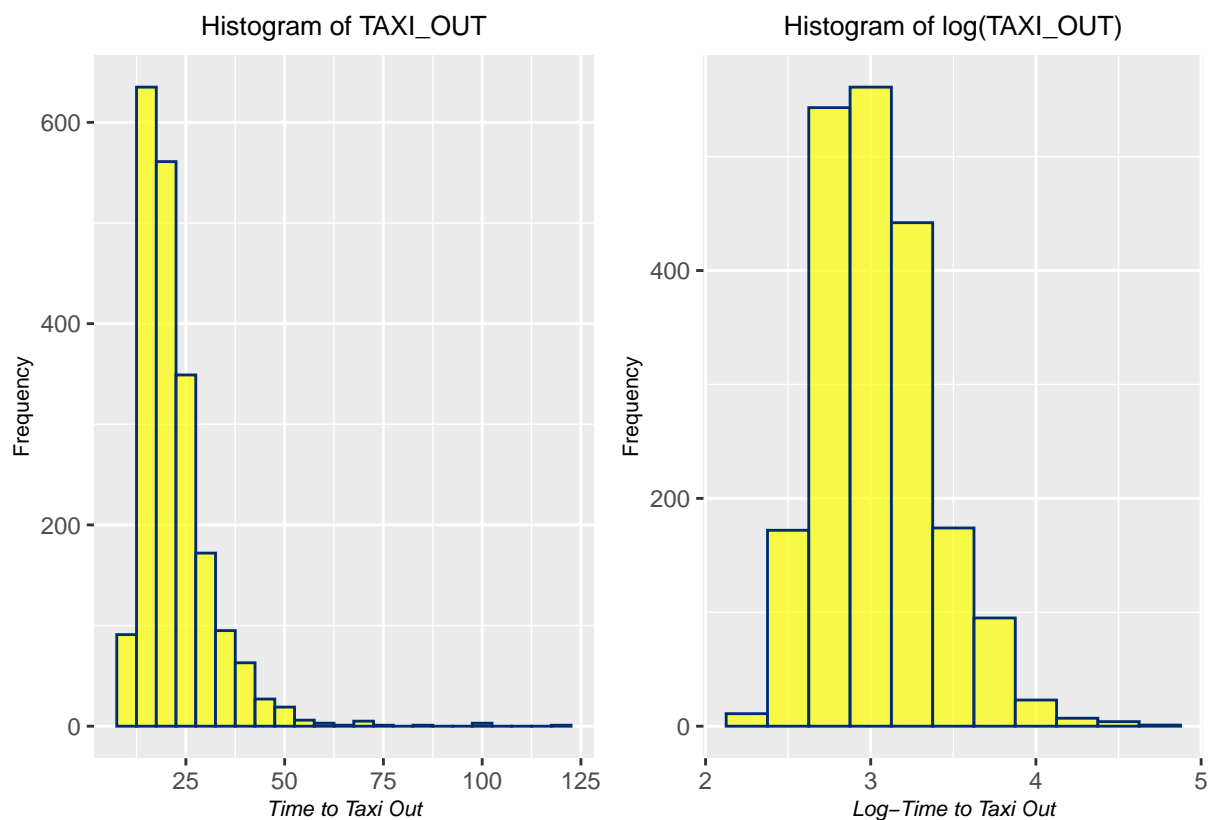
# plot log transformed taxi_out
plog_TAXI_OUT <- ggplot(data = flights, aes(x = log_TAXI_OUT)) +
  geom_histogram(binwidth = .25, fill = "#FFFF00", color = "#002D72", alpha = .7) +
  labs(x = "Log-Time to Taxi Out",
       y = "Frequency",
       title = "Histogram of log(TAXI_OUT)") +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# plot log transform taxi_in
plog_TAXI_IN <- ggplot(data = flights, aes(x = log_TAXI_IN)) +
  geom_histogram(binwidth = .25, fill = "#FFFF00", color = "#002D72", alpha = .7) +
  labs(x = "Log-Time to Taxi In",
       y = "Frequency",
       title = "Histogram of log(TAXI_IN)") +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))
```

pTAXI\_IN + plog\_TAXI\_IN



pTAXI\_OUT + plog\_TAXI\_OUT

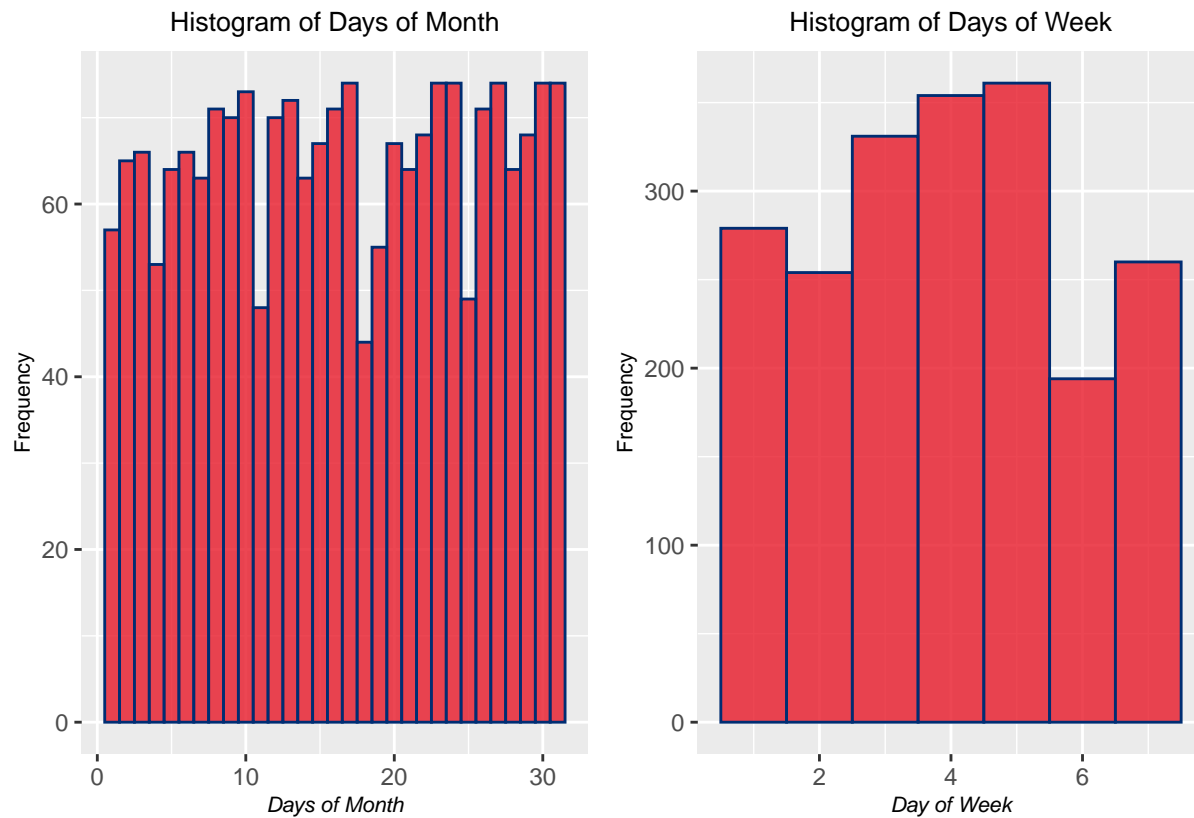


### Days of Month and Week

```
# plot predictor DAYS_OF_MONTH
pDOM <- ggplot(data = flights, aes(x = DAY_OF_MONTH)) +
  geom_histogram(binwidth = 1, fill = "#E81828", color = "#002D72", alpha = .8) +
  labs(x = "Days of Month",
       y = "Frequency",
       title = "Histogram of Days of Month") +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# plot predictor DAY_OF_WEEK
pDOW <- ggplot(data = flights, aes(x = DAY_OF_WEEK)) +
  geom_histogram(binwidth = 1, fill = "#E81828", color = "#002D72", alpha = .8) +
  labs(x = "Day of Week",
       y = "Frequency",
       title = "Histogram of Days of Week") +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

pDOM + pDOW
```

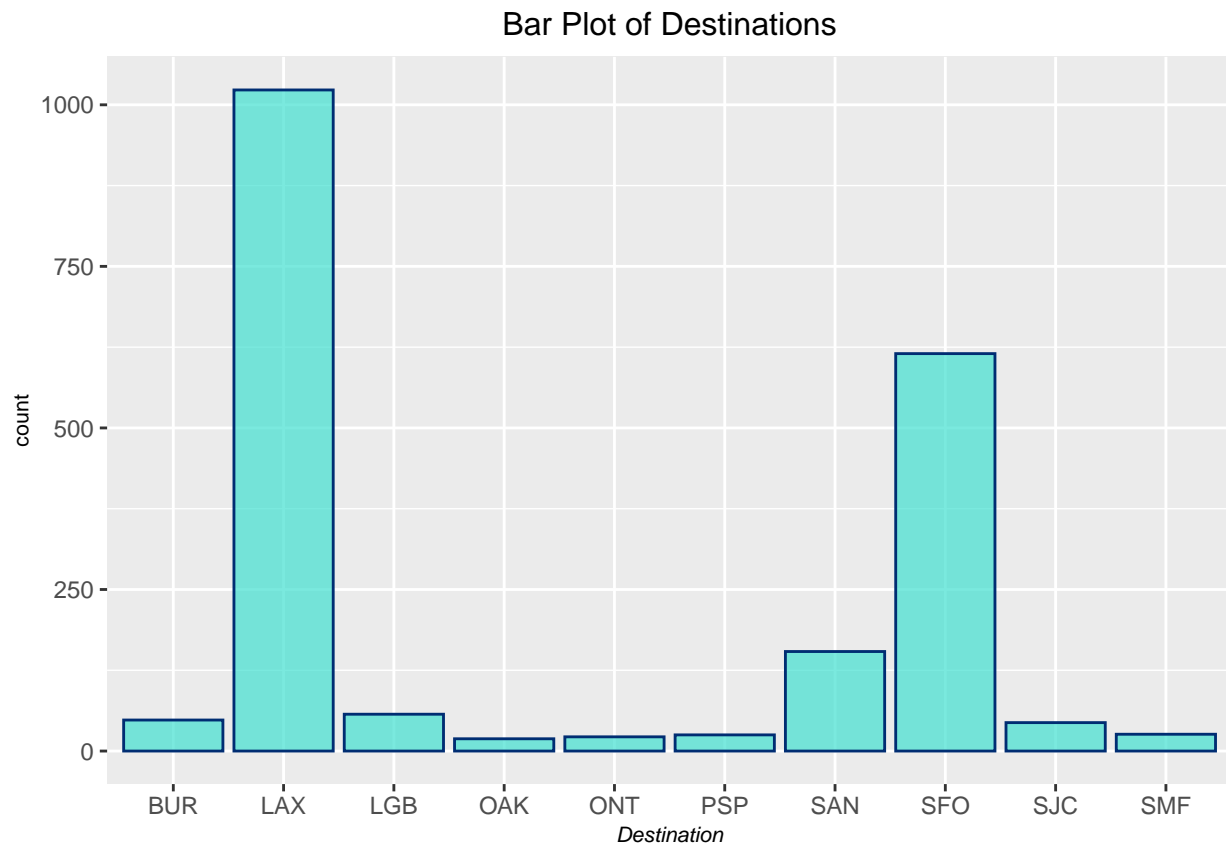


## Destination Locations

Origin is all JFK, but we could consider the different destination locations.

```
# plot destinations in CA
pDEST <- ggplot(data = flights, aes(x = DEST)) +
  geom_bar(fill = "#40E0D0", color = "#002D72", alpha = .7) +
  labs(x = "Destination",
       title = "Bar Plot of Destinations") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))
```

pDEST

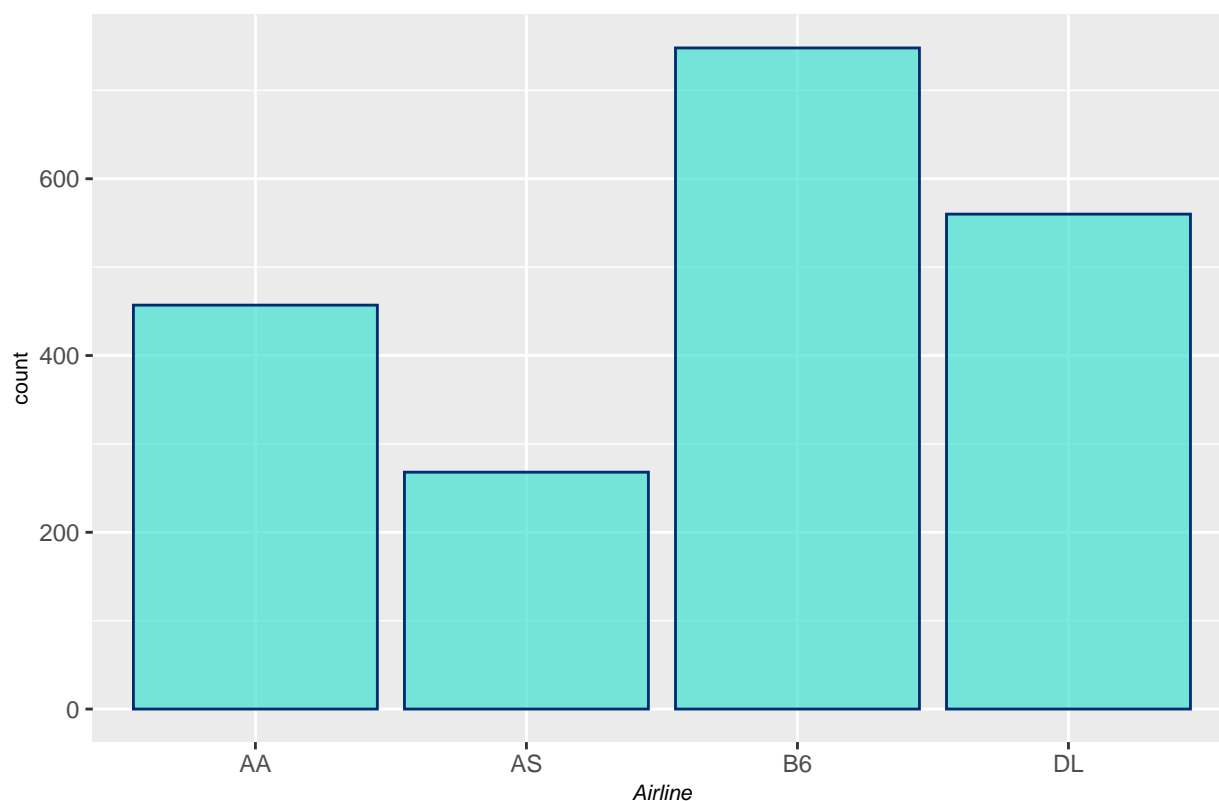


#### Airlines

```
# plot airline carriers
pLINE <- ggplot(data = flights, aes(x = OP_CARRIER)) +
  geom_bar(fill = "#40E0D0", color = "#002D72", alpha = .7) +
  labs(x = "Airline",
       title = "Bar Plot of Airlines") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))
```

pLINE

Bar Plot of Airlines



```
# plot airlines by destination
pLINEDEST <- ggplot(data = flights, aes(x = DEST, fill = OP_CARRIER)) +
  geom_bar() +
  labs(x = "Destination",
       title = "Bar Plot of Airlines by Destination",
       fill = "Airline")
  theme(plot.title = element_text(size = 12,hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))
```

```
## List of 4
## $ axis.title.x.bottom:List of 11
## ..$ family      : NULL
## ..$ face        : chr "italic"
## ..$ colour      : NULL
## ..$ size        : num 8
## ..$ hjust       : NULL
## ..$ vjust       : NULL
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : NULL
## ..$ debug       : NULL
## ..$ inherit.blank: logi FALSE
## .. attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.y.left :List of 11
## ..$ family      : NULL
```



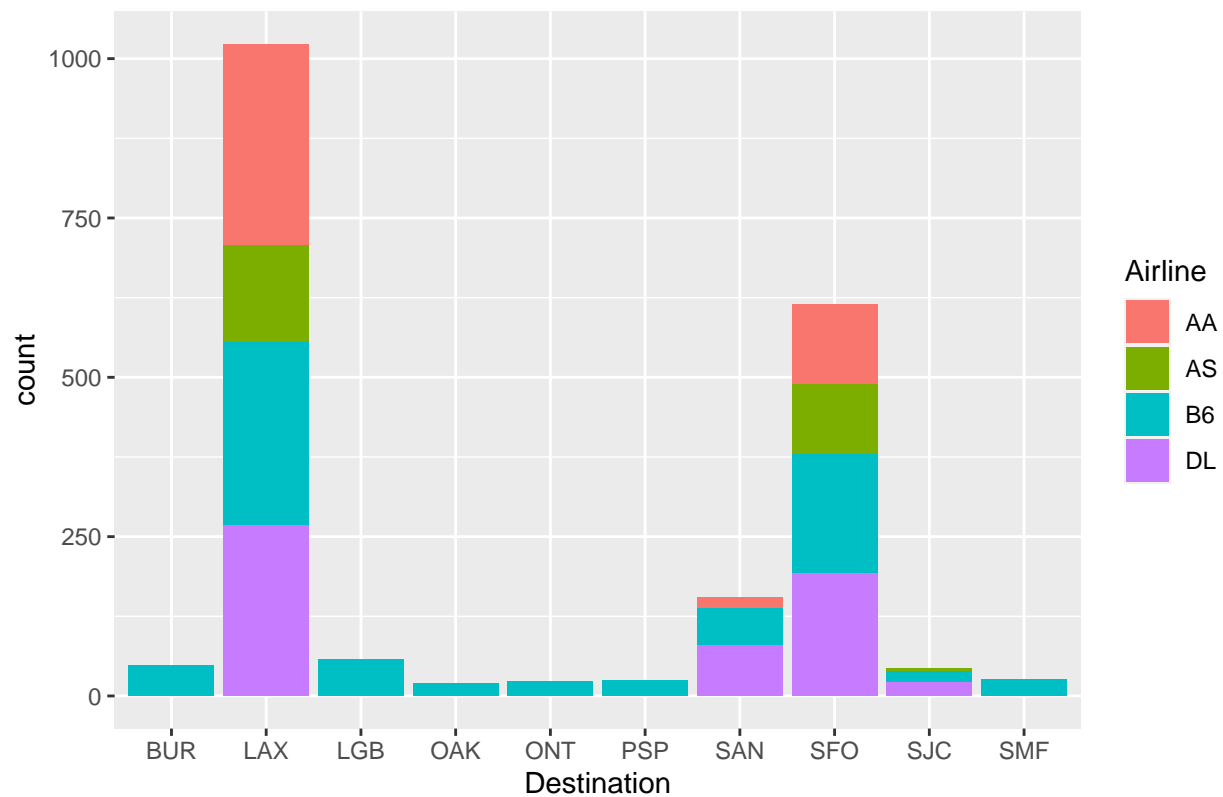
```

## ..$ face          : NULL
## ..$ colour        : NULL
## ..$ size          : num 8
## ..$ hjust         : NULL
## ..$ vjust         : NULL
## ..$ angle         : NULL
## ..$ lineheight    : NULL
## ..$ margin        : NULL
## ..$ debug         : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.title      :List of 11
## ..$ family        : NULL
## ..$ face          : NULL
## ..$ colour        : NULL
## ..$ size          : num 12
## ..$ hjust         : num 0.5
## ..$ vjust         : NULL
## ..$ angle         : NULL
## ..$ lineheight    : NULL
## ..$ margin        : NULL
## ..$ debug         : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.subtitle   :List of 11
## ..$ family        : NULL
## ..$ face          : NULL
## ..$ colour        : NULL
## ..$ size          : NULL
## ..$ hjust         : num 0.5
## ..$ vjust         : NULL
## ..$ angle         : NULL
## ..$ lineheight    : NULL
## ..$ margin        : NULL
## ..$ debug         : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi FALSE
## - attr(*, "validate")= logi TRUE

```

pLINEDEST

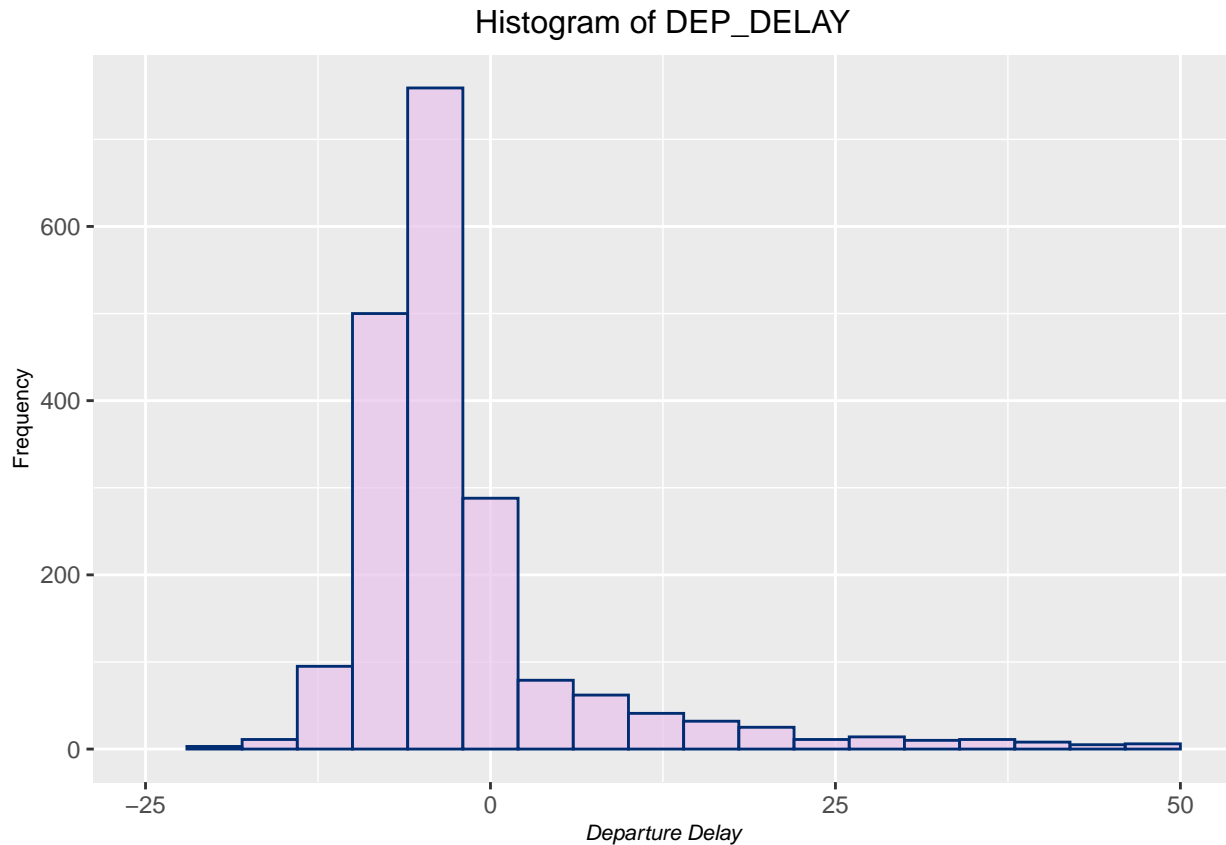
Bar Plot of Airlines by Destination



Depart Delay Histogram

```
# plot DEP_DELAY
pDEPDELAY <- ggplot(data = flights, aes(x = DEP_DELAY)) +
  geom_histogram(binwidth = 4, fill = "#e9c2ed", color = "#002D72", alpha = 0.7) +
  xlim(-25, 50) +
  labs(x = "Departure Delay",
       y = "Frequency",
       title = "Histogram of DEP_DELAY") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

pDEPDELAY
```



```
# plot types of delays
p1 <- ggplot(data = flights, aes(x = CARRIER_DELAY)) +
  geom_bar(fill = "#E81828", color = "#002D72") +
  labs(title = "Carrier Delay")

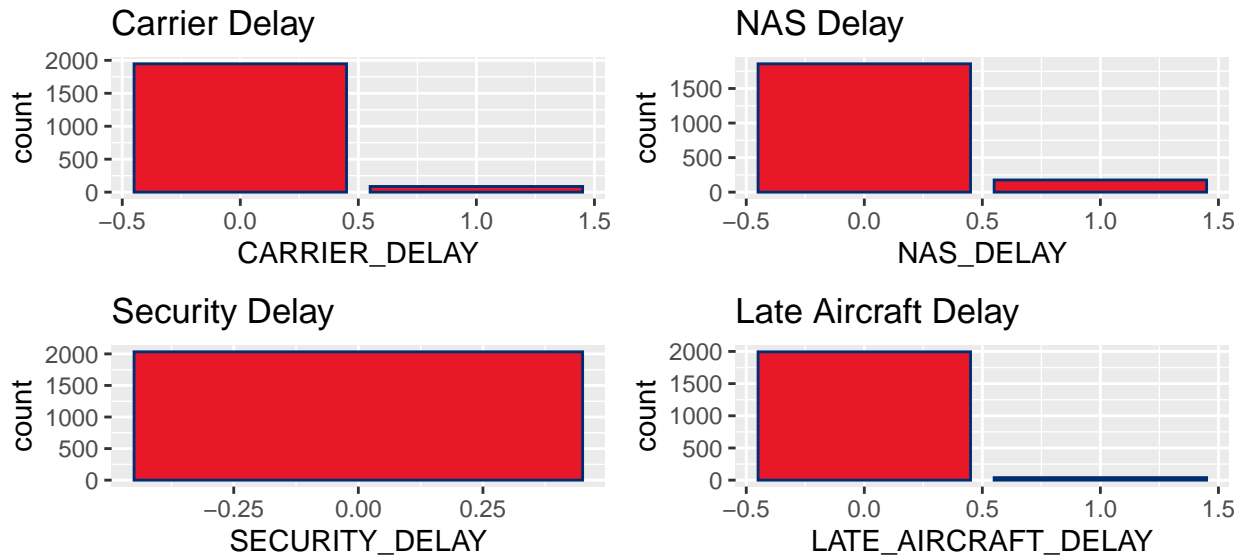
#p2 <- ggplot(data = flights, aes(x = WEATHER_DELAY)) +
#geom_bar(fill = "#E81828", color = "#002D72") +
#labs(title = "Weather Delay")

p3 <- ggplot(data = flights, aes(x = NAS_DELAY)) +
  geom_bar(fill = "#E81828", color = "#002D72") +
  labs(title = "NAS Delay")

p4 <- ggplot(data = flights, aes(x = SECURITY_DELAY)) +
  geom_bar(fill = "#E81828", color = "#002D72") +
  labs(title = "Security Delay")

p5 <- ggplot(data = flights, aes(x = LATE_AIRCRAFT_DELAY)) +
  geom_bar(fill = "#E81828", color = "#002D72") +
  labs(title = "Late Aircraft Delay")

grid.arrange(p1,p3,p4,p5, nrow = 3)
```



From this EDA of the categorical variables, we probably should not perform analysis with `SECURITY_DELAY` since all of them are classified as 0.

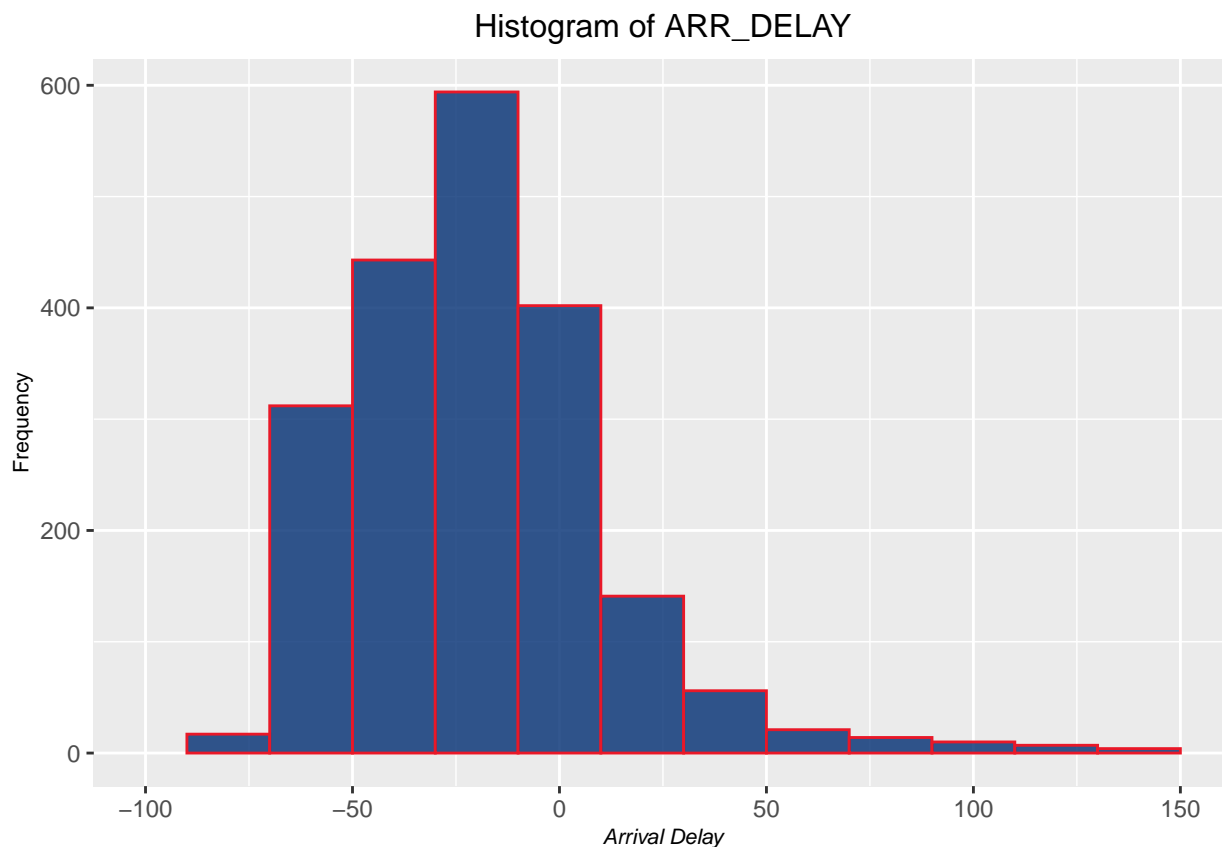
Furthermore, only 9 flights are classified with a weather delay, so it may not be good for our model to include this as a variable for right now.

Overall, the categorical delay predictors I would think we could use are: Carrier Delay, NAS Delay, and Late Aircraft Delay

## Response Variable EDA

```
# plot ARR_DELAY
pARRDELAY <- ggplot(data = flights, aes(x = ARR_DELAY)) +
  geom_histogram(binwidth = 20, fill = "#002D72", color = "#E81828", alpha = 0.8) +
  xlim(-100, 150) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Histogram of ARR_DELAY") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))
```

pARRDELAY

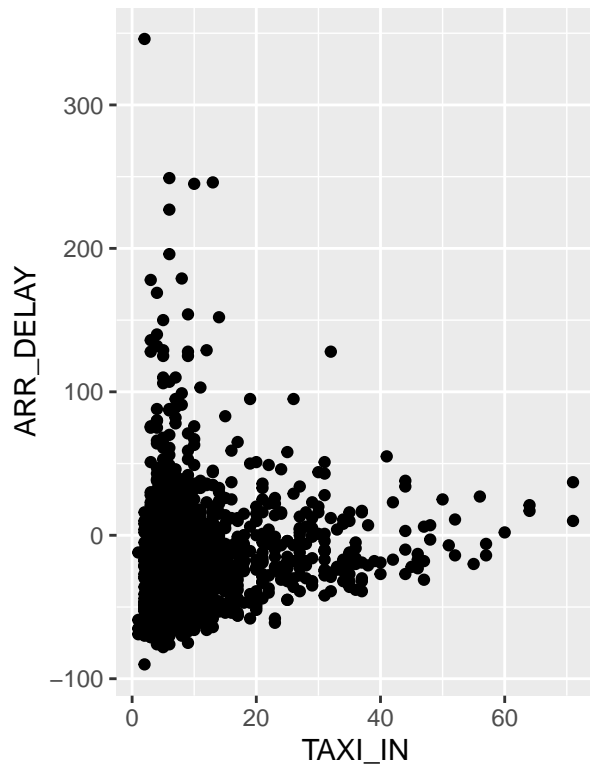


## Predictors vs. Response EDA

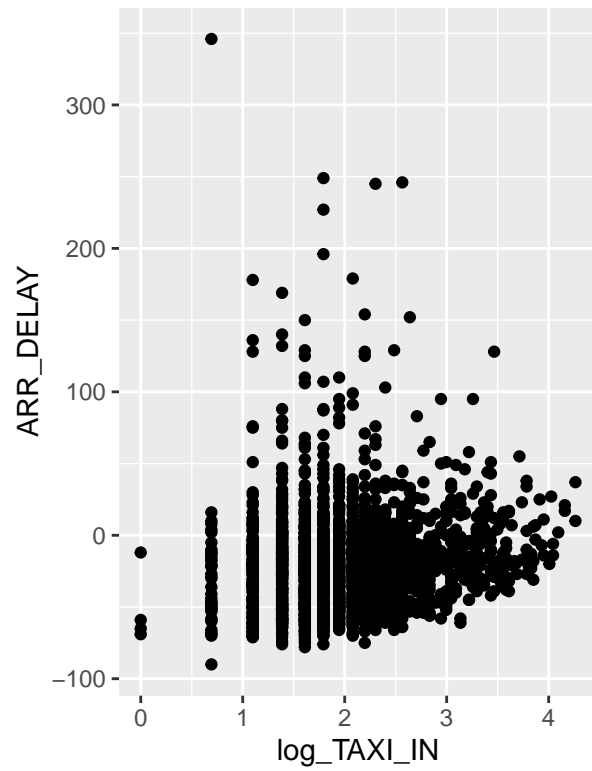
### TAXI\_IN / TAXI\_OUT vs. ARR\_DELAY

```
p6 <- ggplot(data = flights, aes(y = ARR_DELAY, x = TAXI_IN)) +  
  geom_point() +  
  labs(title = "Arrival Delay vs Taxi_In")  
  
p7 <- ggplot(data = flights, aes(y = ARR_DELAY, x = TAXI_OUT)) +  
  geom_point() +  
  labs(title = "Arrival Delay vs Taxi_Out")  
  
plog6 <- ggplot(data = flights, aes(y = ARR_DELAY, x = log_TAXI_IN)) +  
  geom_point() +  
  labs(title = "Arrival Delay vs log(Taxi_In)")  
  
plog7 <- ggplot(data = flights, aes(y = ARR_DELAY, x = log_TAXI_OUT)) +  
  geom_point() +  
  labs(title = "Arrival Delay vs log(Taxi_Out)")  
  
p6 + plog6
```

Arrival Delay vs Taxi\_In

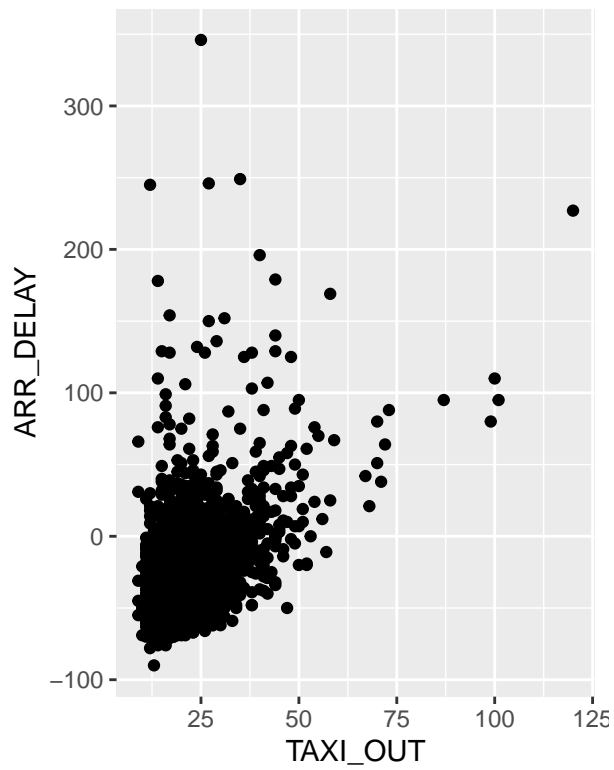


Arrival Delay vs log(Taxi\_In)

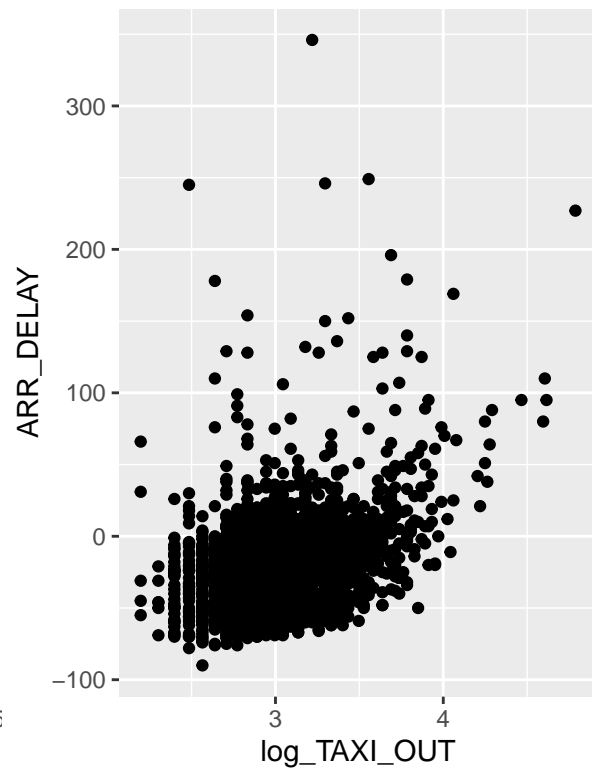


p7 + plog7

Arrival Delay vs Taxi\_Out



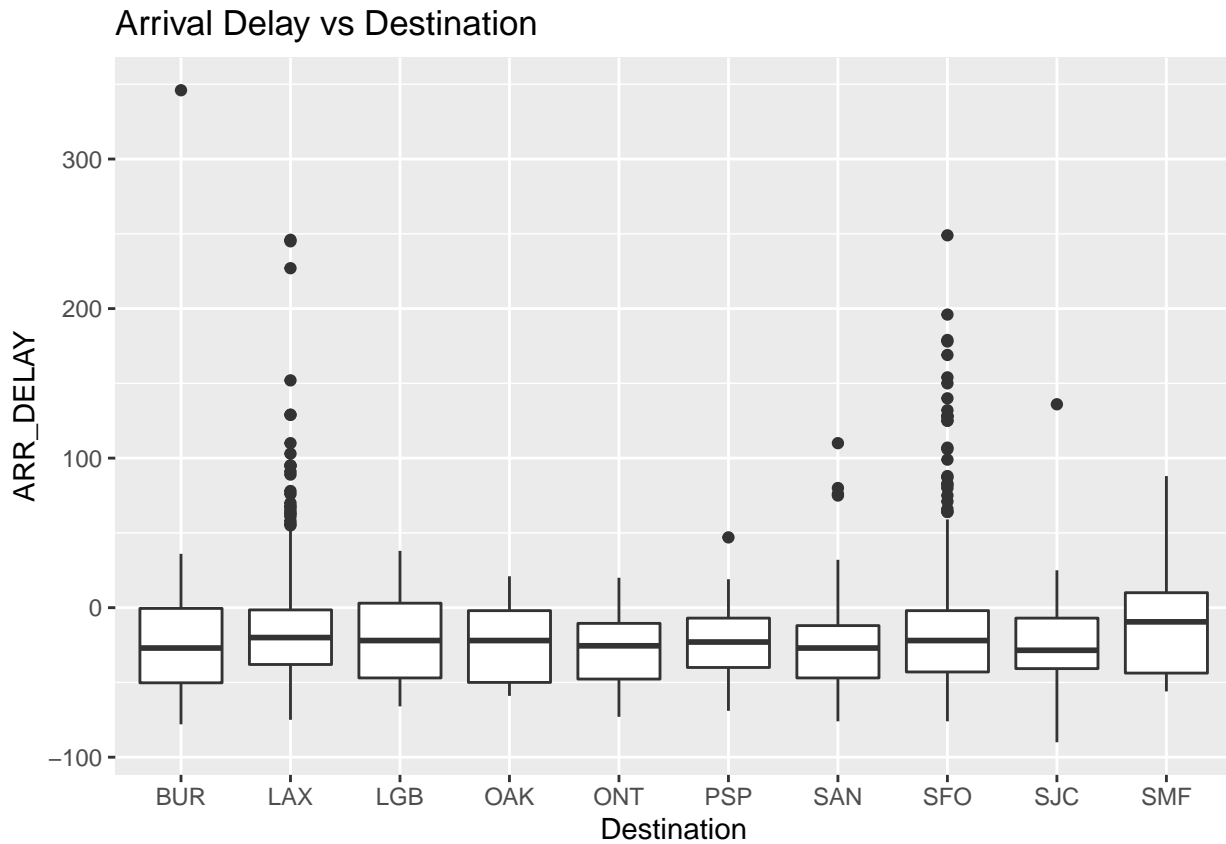
Arrival Delay vs log(Taxi\_Out)



These plots above suggest that we may want to transform the variables at some point.

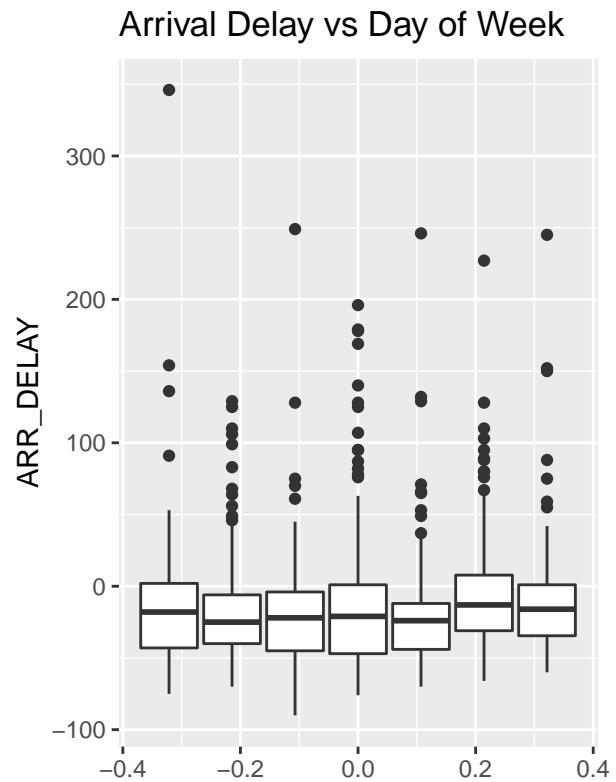
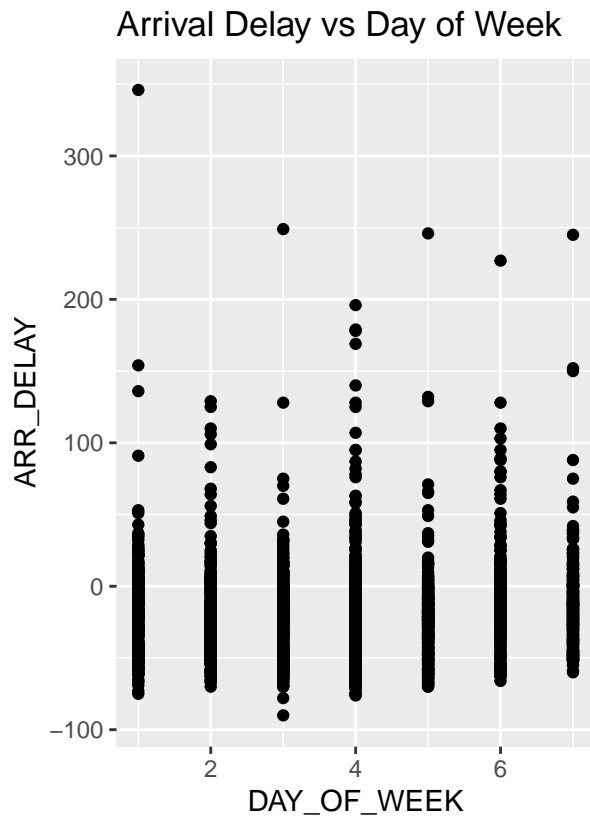
### DEST vs. ARR\_DELAY

```
ggplot(data = flights, aes(y = ARR_DELAY, x = DEST)) +  
  geom_boxplot() +  
  labs(x = "Destination",  
       title = "Arrival Delay vs Destination")
```



### DAY\_OF\_WEEK vs. ARR\_DELAY

```
p8 <- ggplot(data = flights, aes(y = ARR_DELAY, x = DAY_OF_WEEK)) +  
  geom_point() +  
  labs(title = "Arrival Delay vs Day of Week")  
  
p9 <- ggplot(data = flights, aes(y = ARR_DELAY, group = DAY_OF_WEEK)) +  
  geom_boxplot() +  
  labs(title = "Arrival Delay vs Day of Week")  
  
p8 + p9
```



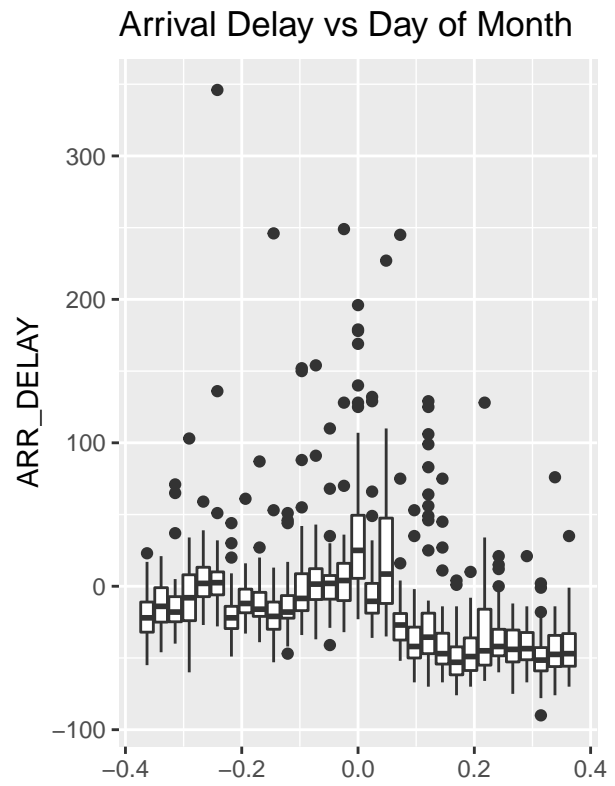
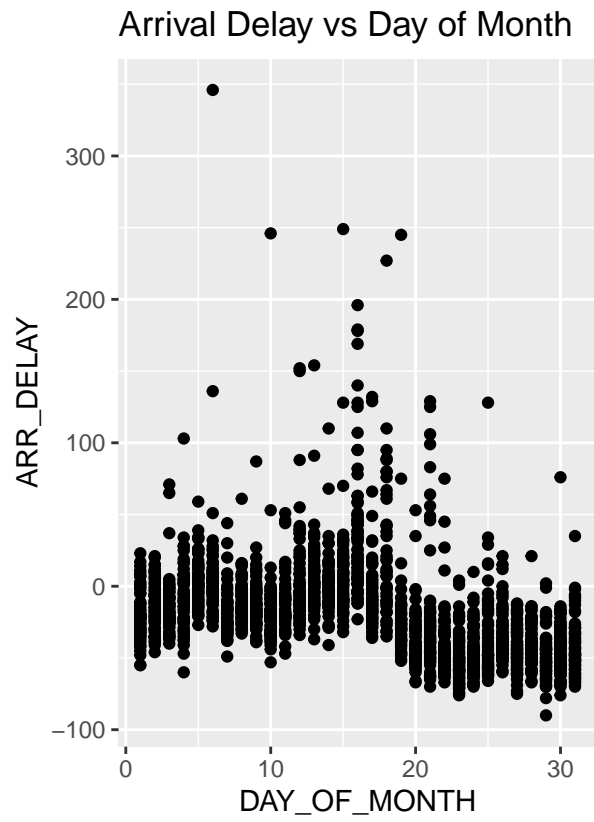
#### ARR\_DELAY and DAY\_OF\_MONTH

```
p10 <- ggplot(data = flights, aes(y = ARR_DELAY, x = DAY_OF_MONTH)) +
  geom_point() +
  labs(title = "Arrival Delay vs Day of Month")

p11 <- ggplot(data = flights, aes(y = ARR_DELAY, group = DAY_OF_MONTH)) +
  geom_boxplot() +
  labs(title = "Arrival Delay vs Day of Month")

p10 + p11
```





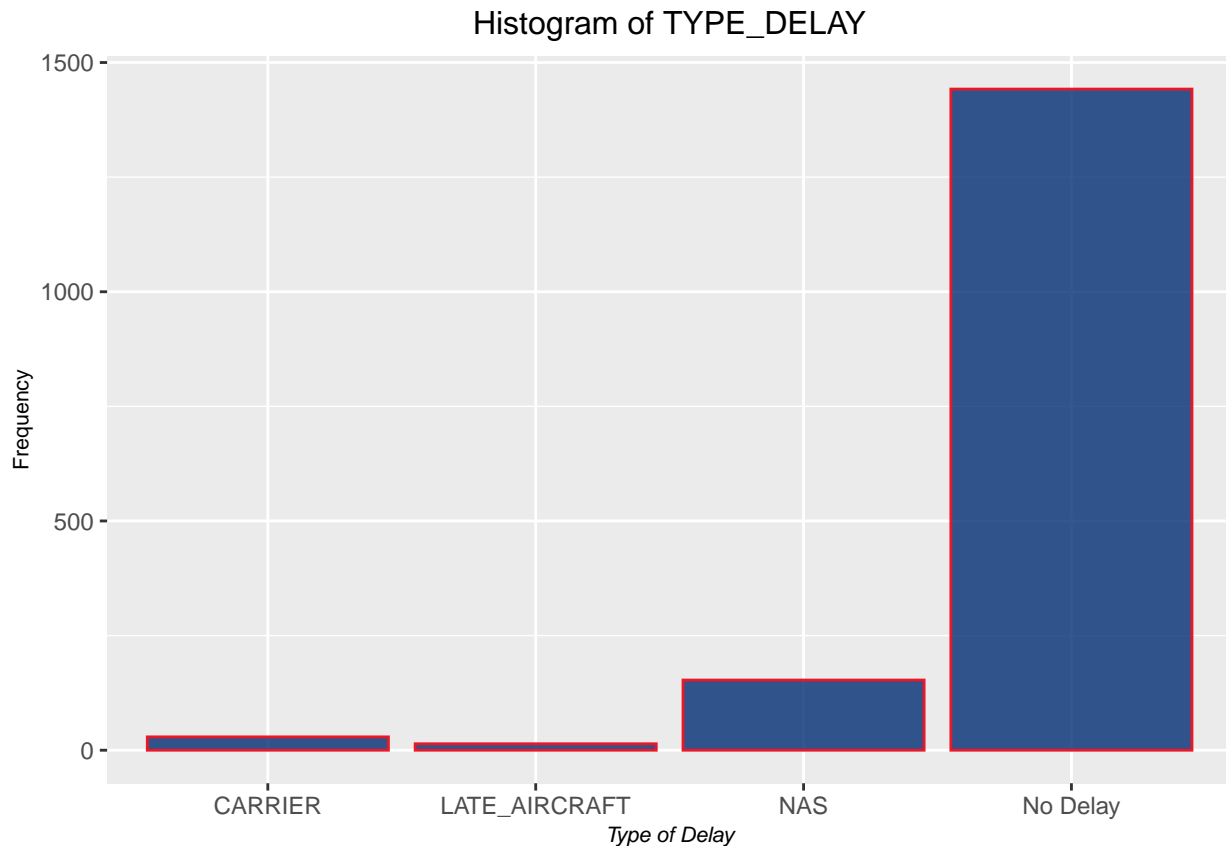
## Additional Data Cleaning

### New Bounds and Variable

```
# take only SFO/LAX since all 4 carriers fly there
# create TYPE_DELAY as a factor variable for type of delays
flights <- flights %>%
  filter(DEST == "SFO" | DEST == "LAX") %>%
  mutate(TYPE_DELAY = case_when(NAS_DELAY == 1 ~ "NAS",
                                CARRIER_DELAY == 1 ~ "CARRIER",
                                LATE_AIRCRAFT_DELAY == 1 ~ "LATE_AIRCRAFT",
                                WEATHER_DELAY == 1 ~ "WEATHER",
                                TRUE ~ "No Delay"))
```

```
pTYPE <- ggplot(data = flights, aes(x = TYPE_DELAY)) +
  geom_bar(fill = "#002D72", color = "#E81828", alpha = 0.8) +
  labs(x = "Type of Delay",
       y = "Frequency",
       title = "Histogram of TYPE_DELAY") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))
```

pTYPE



## Test and Training Set Split

We will use a 80-20 split of training and test sets.

```
set.seed(1234)

flights <- flights %>%
  dplyr::mutate(id = row_number())

train <- flights %>%
  sample_frac(0.8)
test <- anti_join(flights, train, by = "id")
```

# Modeling

## (1) Multiple Linear Regression

### (a) Baesline Linear Model (with AIC Selection)

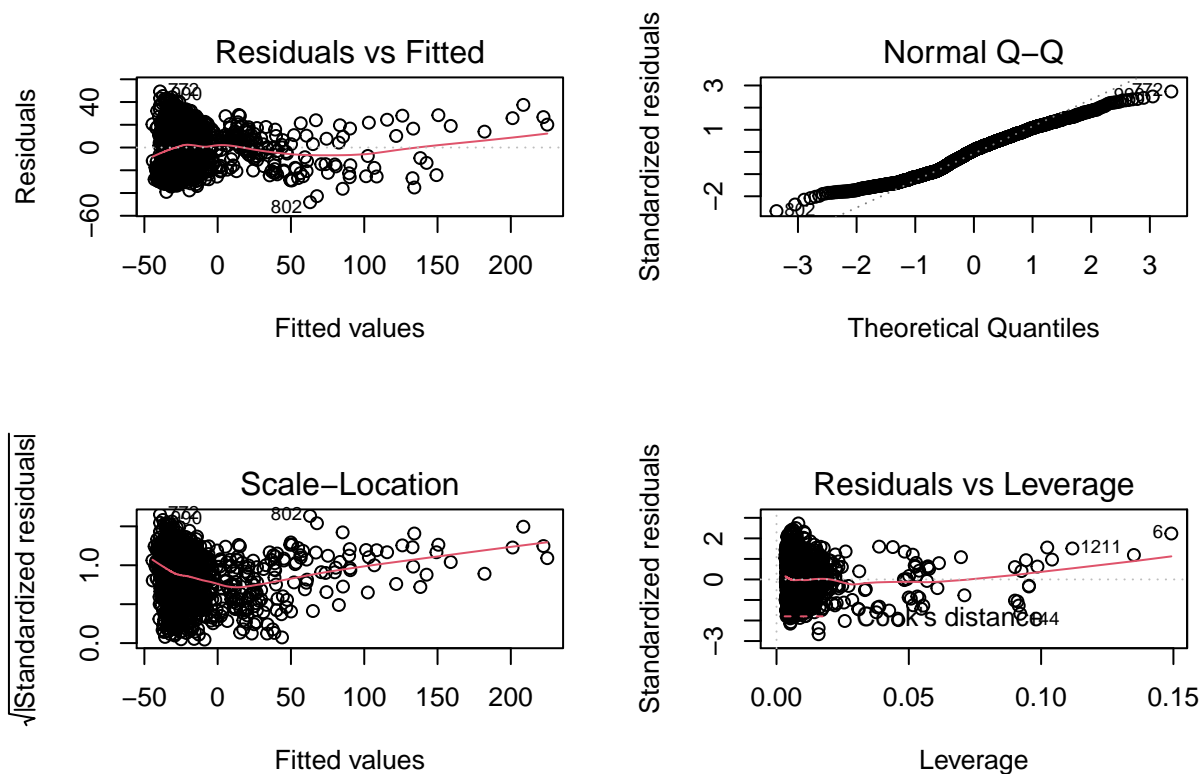
```
# create linear model with all revelant variables
full_model <- lm(ARR_DELAY ~
  DEP_DELAY +
  DAY_OF_WEEK +
  OP_CARRIER +
  DEST +
  CRS_DEP_TIME +
  CRS_ARR_TIME +
  TAXI_OUT +
  TAXI_IN +
  TYPE_DELAY, train)

# summary(full_model)

# use AIC model selection on full model
step_model <- stepAIC(full_model, trace = FALSE)
#summary(step_model)

# update full model
plain_linear_model <- step_model

# plot full model
par(mfrow = c(2,2))
plot(plain_linear_model)
```



## (b) Full Log-Transformed Model

```
# fitting models and running ANOVA tests to identify interactions

## first model
lm.01 <- lm(ARR_DELAY ~ DEP_DELAY + DAY_OF_WEEK + OP_CARRIER + DEST + CRS_DEP_TIME + CRS_ARR_TIME + log_TAXI_OUT + log_TAXI_IN, data = train)
#plot(lm.01)
#summary(lm.01)

## second model
step_model <- stepAIC(lm.01, direction = "backward", trace = FALSE)
#summary(step_model)

## third model
lm.02 <- lm(ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST + CRS_DEP_TIME + log_TAXI_OUT + log_TAXI_IN + TYPE_DELAY, data = train)
#summary(lm.02)
#anova(step_model, lm.02)

## fourth model
lm.03 <- lm(ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST + CRS_DEP_TIME + log_TAXI_OUT + log_TAXI_IN + TYPE_DELAY, data = train)
#anova(lm.02, lm.03)

## final log model
log_linear_model <- lm(ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST + CRS_DEP_TIME + log_TAXI_OUT + log_TAXI_IN + TYPE_DELAY, data = train)
anova(lm.03, log_linear_model)

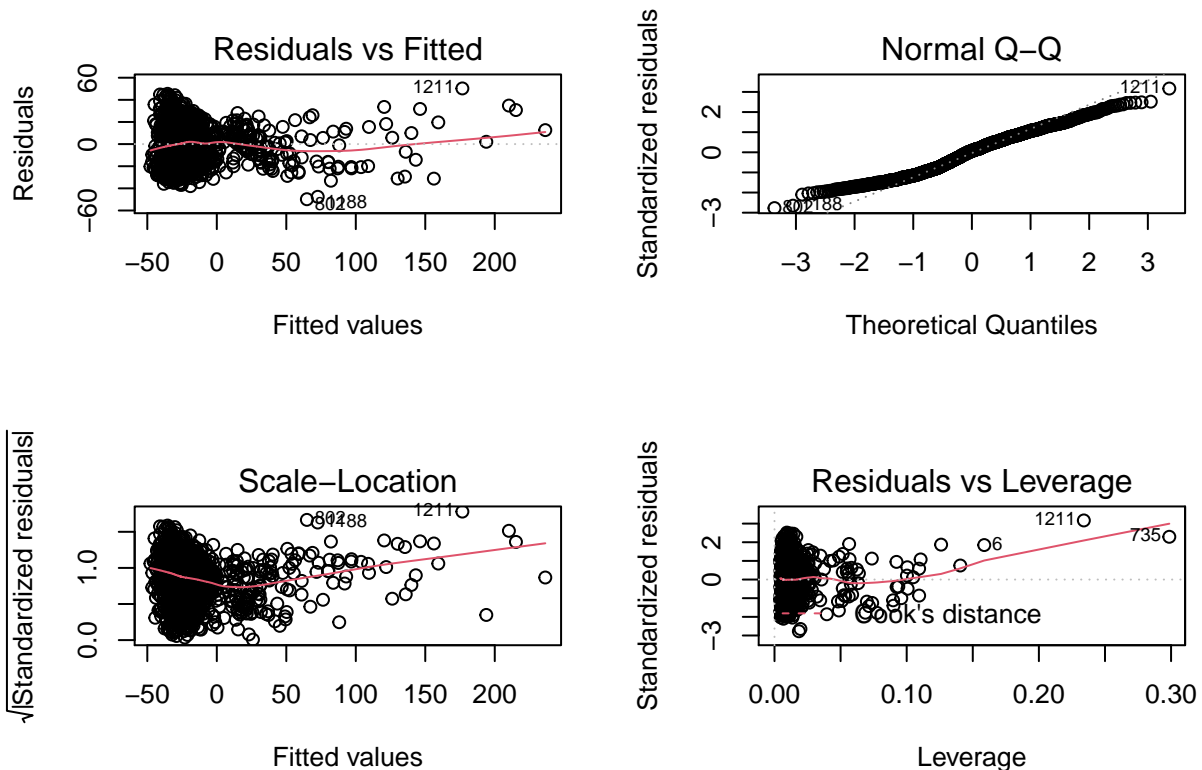
## Analysis of Variance Table
##
## Model 1: ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST + CRS_DEP_TIME + log_TAXI_OUT +
##   log_TAXI_IN + TYPE_DELAY + OP_CARRIER:DEST + DEST:log_TAXI_IN
## Model 2: ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST + CRS_DEP_TIME + log_TAXI_OUT +
##   log_TAXI_IN + TYPE_DELAY + OP_CARRIER:DEST + DEST:log_TAXI_IN +
##   log_TAXI_OUT:DEP_DELAY
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1    1294 427667
## 2    1293 425449   1      2218 6.7408 0.00953 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(log_linear_model)

##
## Call:
## lm(formula = ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST + CRS_DEP_TIME +
##   log_TAXI_OUT + log_TAXI_IN + TYPE_DELAY + OP_CARRIER:DEST +
##   DEST:log_TAXI_IN + log_TAXI_OUT:DEP_DELAY, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -49.817 -15.330   1.198  13.897  50.301
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -81.901399    7.262197  -11.278  < 2e-16 ***
```

```
## DEP_DELAY          0.524207    0.141256    3.711 0.000215 ***
## OP_CARRIERAS      -4.458249    2.087733   -2.135 0.032912 *
## OP_CARRIERB6       5.045463    1.669832    3.022 0.002564 **
## OP_CARRIERDL      -1.493672    1.717998   -0.869 0.384775
## DESTSFO             9.893184    4.366882    2.266 0.023647 *
## CRS_DEP_TIME       -0.004364    0.001070   -4.081 4.77e-05 ***
## log_TAXI_OUT        20.610508    1.617078   12.746 < 2e-16 ***
## log_TAXI_IN         8.433233    1.057392    7.976 3.32e-15 ***
## TYPE_DELAYLATE_AIRCRAFT -3.973566    6.537317   -0.608 0.543408
## TYPE_DELAYNAS       24.019795    4.598524    5.223 2.05e-07 ***
## TYPE_DELAYNo Delay  -15.676745    4.540377   -3.453 0.000573 ***
## OP_CARRIERAS:DESTSFO  6.630276    3.374581    1.965 0.049655 *
## OP_CARRIERB6:DESTSFO -4.199151    2.858830   -1.469 0.142121
## OP_CARRIERDL:DESTSFO -1.424895    2.900122   -0.491 0.623282
## DESTSFO:log_TAXI_IN  -5.261163    1.951509   -2.696 0.007110 **
## DEP_DELAY:log_TAXI_OUT 0.113332    0.043651    2.596 0.009530 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.14 on 1293 degrees of freedom
## Multiple R-squared:  0.7376, Adjusted R-squared:  0.7344
## F-statistic: 227.2 on 16 and 1293 DF, p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(log_linear_model)
```



### (c) Box-Cox-Transformed Response (w/o Log-Transformed Predictors)

```
# adjust response with vertical shift to account for negative responses
## min(train$ARR_DELAY) = -74 --> shift up by constant 75

## transform response
train$adj_ARR_DELAY = train$ARR_DELAY + 75

## fit lm for future Box-Cox use
adj_linear_model <- lm(adj_ARR_DELAY ~
  DEP_DELAY +
  OP_CARRIER +
  DEST +
  CRS_DEP_TIME +
  CRS_ARR_TIME +
  TAXI_OUT +
  TAXI_IN +
  TYPE_DELAY,
  data = train)
```

#### Box-Cox Setup

```
# box cox transform (same as case study)
## call EnvStats library
bc_model <- EnvStats::boxcox(adj_linear_model, optimize = TRUE)
# find optimal lambda parameter
bc_lambda <- bc_model$lambda
bc_lambda
```

```
## [1] 0.6981342
```

```
#plot(bc_model)
```

```
# add Box-Cox response as variable to train set
train <- train %>%
  mutate(bc_adj_ARR_DELAY = ((adj_ARR_DELAY^bc_lambda) - 1)/bc_lambda)

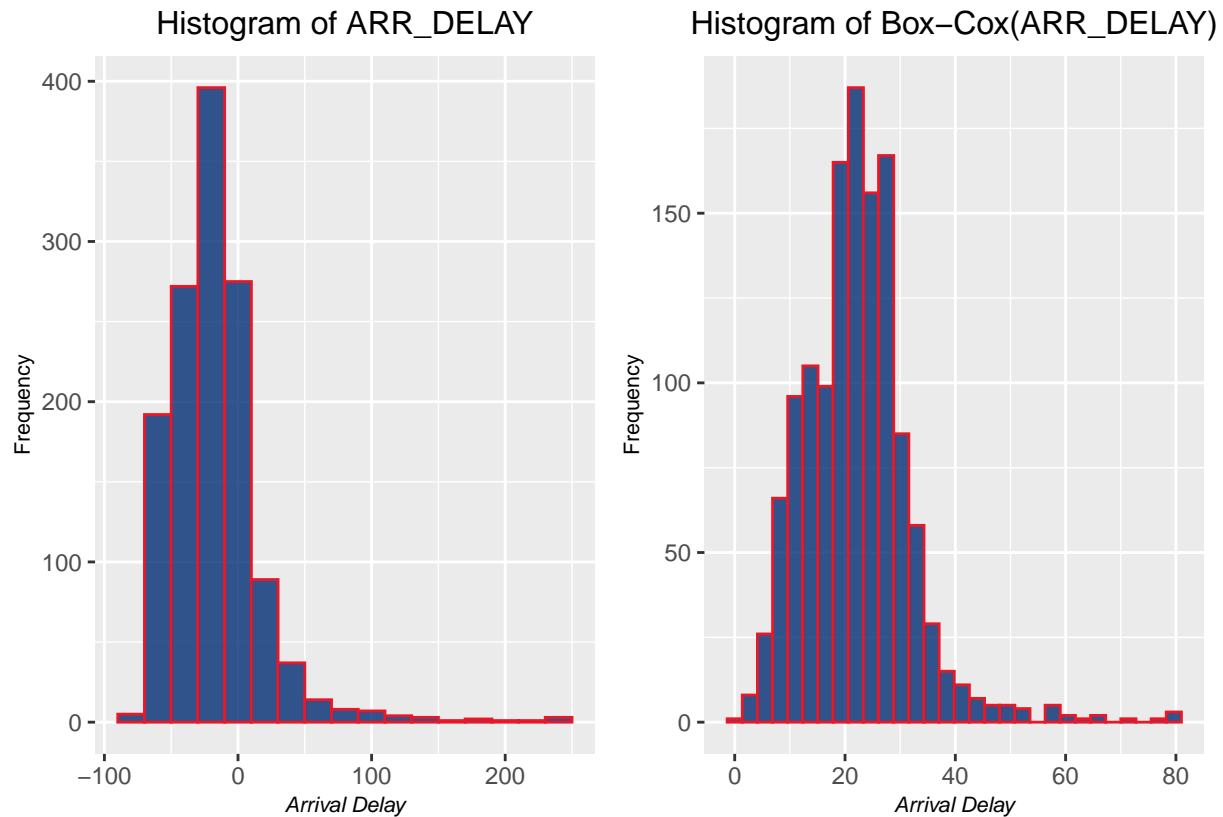
# plot training ARR_DELAY
ptrain_ARRDELAY <- ggplot(data = train, aes(x = ARR_DELAY)) +
  geom_histogram(binwidth = 20, fill = "#002D72", color = "#E81828", alpha = 0.8) +
  labs(x = "Arrival Delay",
    y = "Frequency",
    title = "Histogram of ARR_DELAY") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    axis.title.x.bottom = element_text(size = 8, face = "italic"),
    axis.title.y.left = element_text(size = 8))

# plot Box-Cox training ARR_DELAY
ptrain_bcARRDELAY <- ggplot(data = train, aes(x = bc_adj_ARR_DELAY)) +
  geom_histogram(fill = "#002D72", color = "#E81828", alpha = 0.8) +
  labs(x = "Arrival Delay",
    y = "Frequency",
    title = "Histogram of Box-Cox(ARR_DELAY)") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
```

```
plot.subtitle = element_text(hjust = 0.5),
axis.title.x.bottom = element_text(size = 8, face = "italic"),
axis.title.y.left = element_text(size = 8))
```

```
ptrain_ARRDELAY + ptrain_bcARRDELAY
```

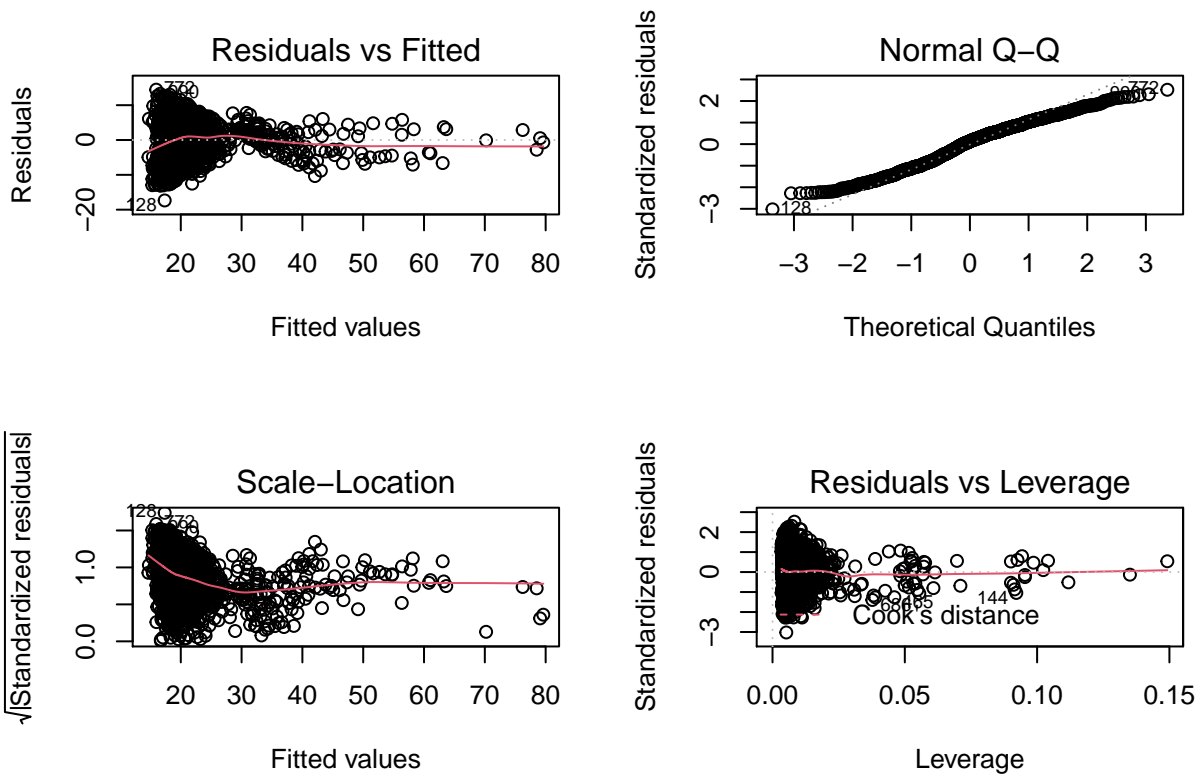
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



## Final Box-Cox Linear Model

```
bc_adj_linear_model <- lm(bc_adj_ARR_DELAY ~
  DEP_DELAY +
  OP_CARRIER +
  DEST +
  CRS_DEP_TIME +
  CRS_ARR_TIME +
  TAXI_OUT +
  TAXI_IN +
  TYPE_DELAY,
  data = train)
par(mfrow = c(2,2))
plot(bc_adj_linear_model)
```





```
summary(bc_adj_linear_model)
```

```
##
## Call:
## lm(formula = bc_adj_ARR_DELAY ~ DEP_DELAY + OP_CARRIER + DEST +
##     CRS_DEP_TIME + CRS_ARR_TIME + TAXI_OUT + TAXI_IN + TYPE_DELAY,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3550  -4.6040   0.7628   4.3532  14.4500
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    21.9473168   1.5511418   14.149 < 2e-16 ***
## DEP_DELAY       0.2019174   0.0085684   23.565 < 2e-16 ***
## OP_CARRIERAS  -0.3865039   0.5255208   -0.735 0.462187
## OP_CARRIERB6   0.4703869   0.4324730    1.088 0.276944
## OP_CARRIERDL  -0.3934861   0.4383847   -0.898 0.369575
## DESTSFO        -0.6403901   0.3426977   -1.869 0.061894 .
## CRS_DEP_TIME   -0.0012308   0.0003474   -3.543 0.000410 ***
## CRS_ARR_TIME   -0.0005199   0.0002776   -1.873 0.061325 .
## TAXI_OUT        0.2468133   0.0192961   12.791 < 2e-16 ***
## TAXI_IN         0.1470986   0.0194304    7.571 7.02e-14 ***
## TYPE_DELAYLATE_AIRCRAFT -0.9730558   2.0594731   -0.472 0.636665
## TYPE_DELAYNAS    5.1595880   1.4211386    3.631 0.000294 ***
## TYPE_DELAYNo Delay -5.5980132   1.4072197   -3.978 7.33e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5.755 on 1297 degrees of freedom
## Multiple R-squared:  0.6445, Adjusted R-squared:  0.6412
## F-statistic: 195.9 on 12 and 1297 DF,  p-value: < 2.2e-16
```

## MLR Test Error Metrics & Predictions

LOOKING AT: - model without any interactions or transformations: ENDED UP PERFORMING THE BEST!!! - log transformed predictors (taxi\_in and taxi\_out), interactions, no boxcox transformed response - box-cox transformed response, no interactions or predicted transformations

```
# predictions
## baseline
test$plain_mlr_pred <- predict(plain_linear_model, test)
## log
test$log_linear_preds <- predict(log_linear_model, test)
## bc
test$adj_ARR_DELAY = test$ARR_DELAY + 77
test$bc_adj_linear_preds <- predict(bc_adj_linear_model, test)
test <- test %>%
  mutate(adj_linear_preds = ((bc_adj_linear_preds*(bc_lambda) + 1)^(1/bc_lambda)))
test$bc_mlr_pred = test$adj_linear_preds - 77

## histogram of predictions vs. actual
# actual ARR_DELAY in test set
#hist(test$ARR_DELAY)
ptest_actual <- ggplot(data = test, aes(x = ARR_DELAY)) +
  geom_histogram(fill = "#E81828", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Actual") +
  theme(plot.title = element_text(size = 12,hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- baseline lm
ptest_baseline_preds <- ggplot(data = test, aes(x = plain_mlr_pred)) +
  geom_histogram(fill = "#E81828", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Baseline Linear Model") +
  theme(plot.title = element_text(size = 12,hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- lm with log-trans
# hist(test$log_linear_preds)
ptest_lm_log_preds <- ggplot(data = test, aes(x = log_linear_preds)) +
  geom_histogram(fill = "#E81828", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Linear Model w/ Log-Transformations") +
  theme(plot.title = element_text(size = 12,hjust = 0.5),
```

```

plot.subtitle = element_text(hjust = 0.5),
axis.title.x.bottom = element_text(size = 8, face = "italic"),
axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- lm with Box-Cox model
#hist(test$log_linear_preds)
ptest_bc_preds <- ggplot(data = test, aes(x = bc_mlr_pred)) +
  geom_histogram(fill = "#E81828", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Linear Model w/ Box-Cox Response") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

mlr_patchwork <- ptest_actual + (ptest_baseline_preds / ptest_bc_preds / ptest_lm_log_preds)

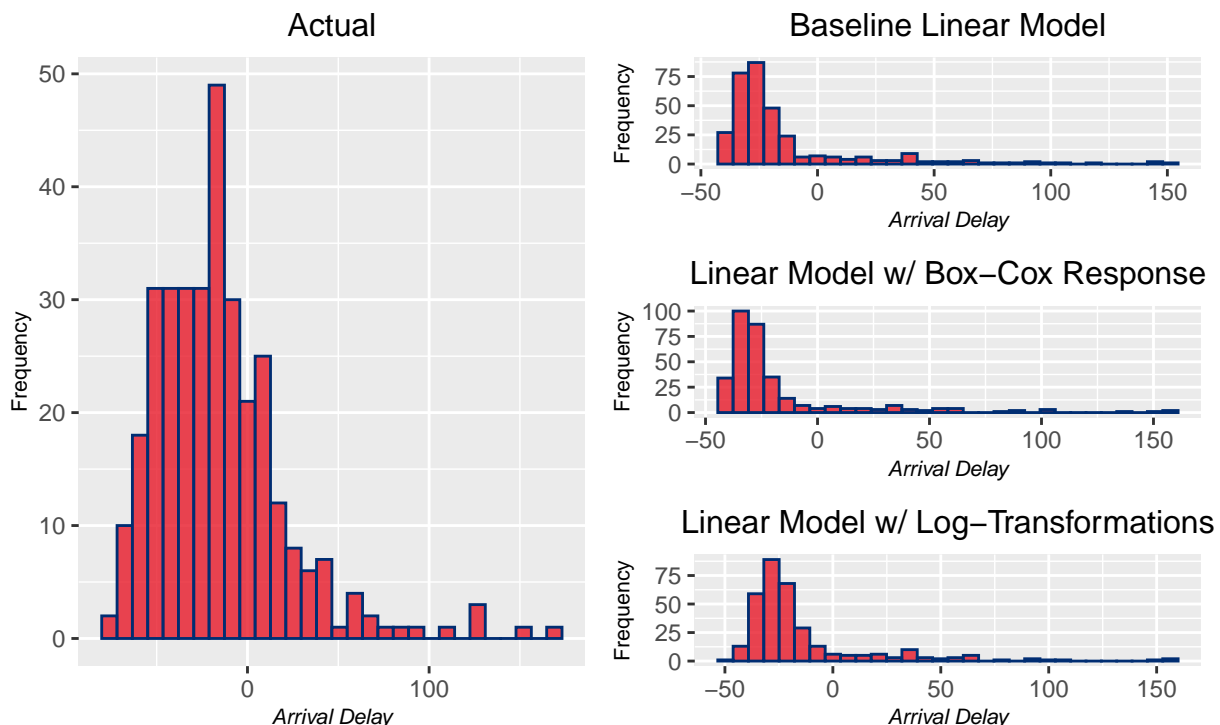
mlr_patchwork + plot_annotation(
  title = 'Comparing Distributions of ARR_DELAY',
  subtitle = 'Histograms of actual test values and MLR-predicted values'
)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```

## Comparing Distributions of ARR\_DELAY

Histograms of actual test values and MLR-predicted values



```

# test MSE calculations
plain_linear_model_MSE <- sum((test$ARR_DELAY - test$plain_mlr_pred)^2, na.rm=T)/length(test$ARR_DELAY)
plain_linear_model_MSE

## [1] 322.4588

log_linear_MSE <- sum((test$log_linear_preds-test$ARR_DELAY)^2, na.rm=T)/length(test$ARR_DELAY)
log_linear_MSE

## [1] 333.8962

bc_adj_linear_model_MSE <- sum((test$ARR_DELAY - test$bc_mlr_pred)^2, na.rm=T)/length(test$ARR_DELAY)
bc_adj_linear_model_MSE

## [1] 334.9226

```

## (2) Generalized Additive Models

### (a) Initial GAM: No Box-Cox on Response

```

gam00 <- gam(ARR_DELAY ~ DAY_OF_WEEK +
              OP_CARRIER +
              s(TAXI_IN) +
              s(TAXI_OUT) +
              DEST +
              s(DEP_DELAY) +
              s(CRS_DEP_TIME) +
              s(CRS_ARR_TIME) +
              TYPE_DELAY, data = train)

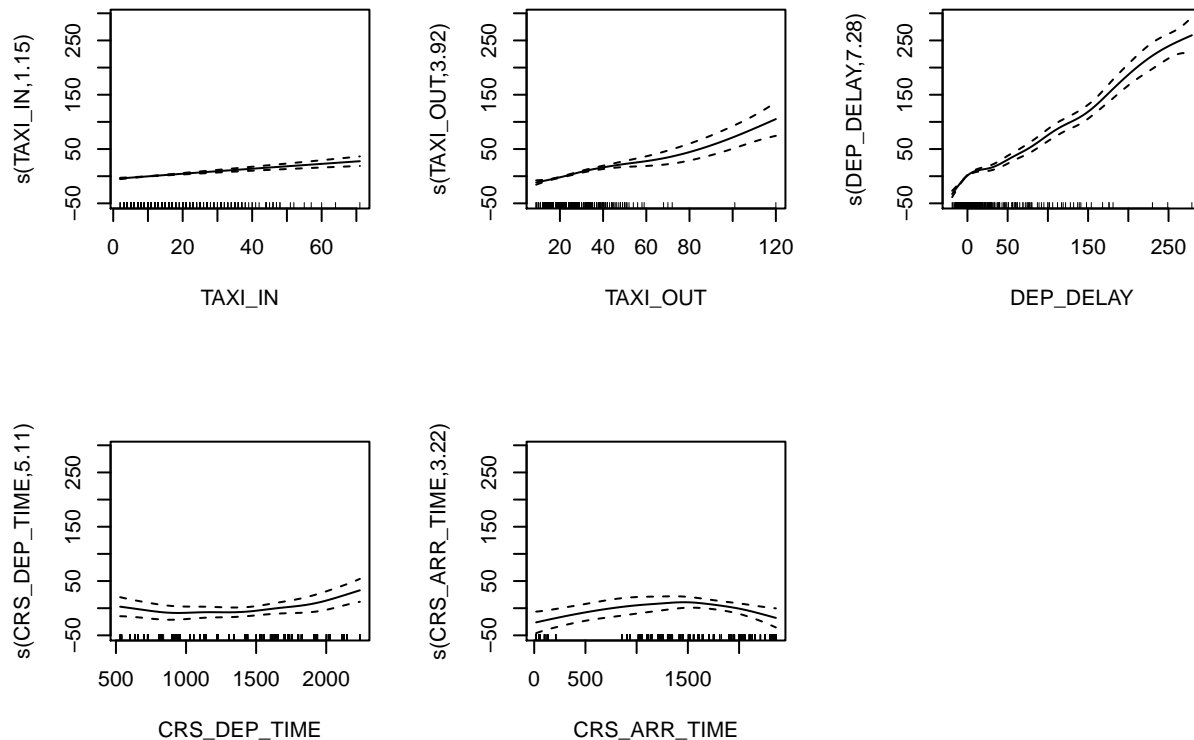
summary(gam00)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## ARR_DELAY ~ DAY_OF_WEEK + OP_CARRIER + s(TAXI_IN) + s(TAXI_OUT) +
##      DEST + s(DEP_DELAY) + s(CRS_DEP_TIME) + s(CRS_ARR_TIME) +
##      TYPE_DELAY
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.6248     4.6488   0.134  0.8931
## DAY_OF_WEEK       0.2465     0.2625   0.939  0.3479
## OP_CARRIERAS     -1.4083     1.6722  -0.842  0.3999
## OP_CARRIERB6      2.8700     1.3616   2.108  0.0352 *
## OP_CARRIERDL     -2.7519     1.3905  -1.979  0.0480 *
## DESTSFO           -0.5607     1.1267  -0.498  0.6188
## TYPE_DELAYLATE_AIRCRAFT -3.3786     6.5848  -0.513  0.6080
## TYPE_DELAYNAS      18.9194     4.5466   4.161 3.38e-05 ***
## TYPE_DELAYNo Delay -22.2130     4.5742  -4.856 1.34e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:

```

```
##              edf Ref.df      F  p-value
## s(TAXI_IN)    1.148  1.283 44.974 6.24e-13 ***
## s(TAXI_OUT)    3.922  4.851 46.982 < 2e-16 ***
## s(DEP_DELAY)   7.279  8.272 136.783 < 2e-16 ***
## s(CRS_DEP_TIME) 5.114  6.050  2.121  0.0415 *
## s(CRS_ARR_TIME) 3.221  3.897  1.845  0.1277
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.749   Deviance explained = 75.5%
## GCV = 317.86   Scale est. = 310.66      n = 1310
```

```
par(mfrow = c(2,3))
plot.gam(gam00, se=TRUE)
```



**Checking Linearity** TAXI\_IN may be linear

```
gam01 <- gam(ARR_DELAY ~ DAY_OF_WEEK +
              OP_CARRIER +
              TAXI_IN +
              s(TAXI_OUT) +
              DEST +
              s(DEP_DELAY) +
              s(CRS_DEP_TIME) +
              s(CRS_ARR_TIME) +
              TYPE_DELAY, data = train)

anova(gam00, gam01, test = "F")
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: ARR_DELAY ~ DAY_OF_WEEK + OP_CARRIER + s(TAXI_IN) + s(TAXI_OUT) +
##   DEST + s(DEP_DELAY) + s(CRS_DEP_TIME) + s(CRS_ARR_TIME) +
##   TYPE_DELAY
## Model 2: ARR_DELAY ~ DAY_OF_WEEK + OP_CARRIER + TAXI_IN + s(TAXI_OUT) +
##   DEST + s(DEP_DELAY) + s(CRS_DEP_TIME) + s(CRS_ARR_TIME) +
##   TYPE_DELAY
##   Resid. Df Resid. Dev      Df Deviance      F Pr(>F)
## 1    1276.6    397738
## 2    1276.9    397845 -0.29646  -106.39 1.1552 0.1929
```

based on anova test, the model with a smoothing spline on TAXI\_IN is a better fit

**More ANOVA Tests** DAY\_OF\_WEEK, DEST, and CRS\_ARR\_TIME have very high p-values, so let's try an anova test without including them

```
gam02 <- gam(ARR_DELAY ~ OP_CARRIER +
              s(TAXI_IN) +
              s(TAXI_OUT) +
              s(DEP_DELAY) +
              s(CRS_DEP_TIME) +
              TYPE_DELAY, data = train)

anova(gam00, gam02, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: ARR_DELAY ~ DAY_OF_WEEK + OP_CARRIER + s(TAXI_IN) + s(TAXI_OUT) +
##   DEST + s(DEP_DELAY) + s(CRS_DEP_TIME) + s(CRS_ARR_TIME) +
##   TYPE_DELAY
## Model 2: ARR_DELAY ~ OP_CARRIER + s(TAXI_IN) + s(TAXI_OUT) + s(DEP_DELAY) +
##   s(CRS_DEP_TIME) + TYPE_DELAY
##   Resid. Df Resid. Dev      Df Deviance      F Pr(>F)
## 1    1276.6    397738
## 2    1280.0    400550 -3.3459  -2811.2 2.7045 0.03822 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

based on the anova test, the model excluding these variables is a better fit

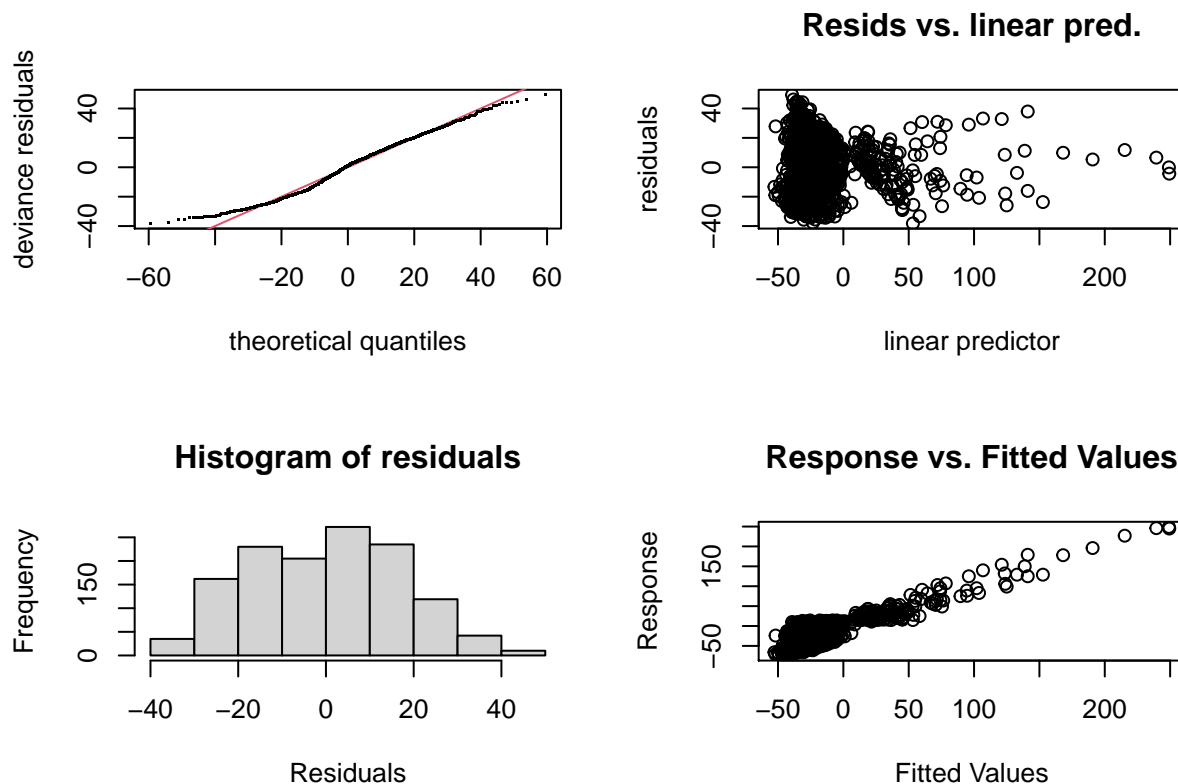
## Tuned Initial GAM

```
# final fit
summary(gam02)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## ARR_DELAY ~ OP_CARRIER + s(TAXI_IN) + s(TAXI_OUT) + s(DEP_DELAY) +
##   s(CRS_DEP_TIME) + TYPE_DELAY
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.828      4.530   0.404  0.6865
## OP_CARRIERAS    -1.676      1.664  -1.007  0.3140
```

```
## OP_CARRIERB6          2.486      1.354   1.836   0.0666 .
## OP_CARRIERDL         -3.137      1.381  -2.272   0.0233 *
## TYPE_DELAYLATE_AIRCRAFT -3.199      6.601  -0.485   0.6281
## TYPE_DELAYNAS         18.795      4.556   4.126  3.93e-05 ***
## TYPE_DELAYNo Delay    -22.413      4.588  -4.885  1.16e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(TAXI_IN)      1.260  1.478 43.326 7.39e-14 ***
## s(TAXI_OUT)      4.308  5.298 44.490 < 2e-16 ***
## s(DEP_DELAY)     7.384  8.348 134.478 < 2e-16 ***
## s(CRS_DEP_TIME)  6.781  7.883  5.229 2.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.748   Deviance explained = 75.3%
## GCV = 318.64   Scale est. = 312.13      n = 1310
```

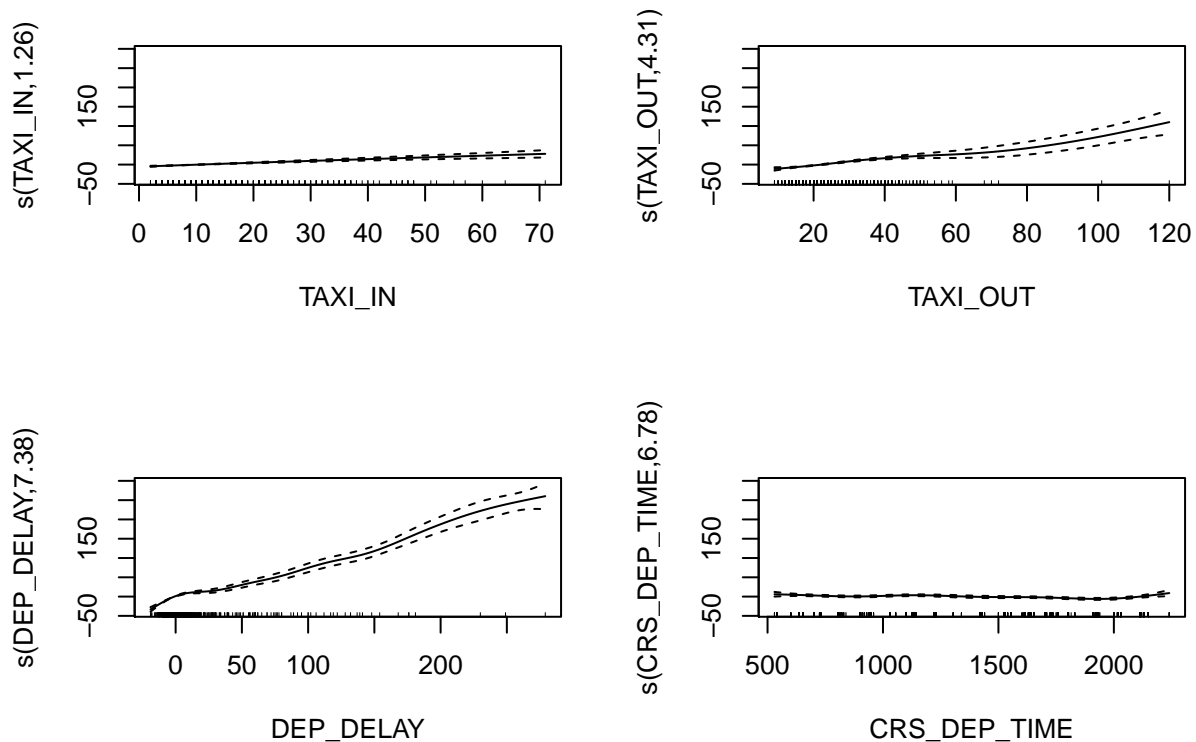
```
# diagnostic plots
par(mfrow = c(2,2))
gam.check(gam02)
```



```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 12 iterations.
## The RMS GCV score gradient at convergence was 0.0007156955 .
## The Hessian was positive definite.
## Model rank = 43 / 43
```

```
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(TAXI_IN)   9.00 1.26   0.99   0.32
## s(TAXI_OUT)   9.00 4.31   1.07   0.99
## s(DEP_DELAY)  9.00 7.38   0.99   0.31
## s(CRS_DEP_TIME) 9.00 6.78   0.97   0.14

# predictor plots
par(mfrow = c(2,2))
plot(gam02)
```



## (b) Secondary GAM: Box-Cox on Response

```
gambc <- gam(bc_adj_ARR_DELAY ~ OP_CARRIER +
             s(TAXI_IN) +
             s(TAXI_OUT) +
             s(DEP_DELAY) +
             s(CRS_DEP_TIME) +
             TYPE_DELAY, data = train)

# final BC GAM fit
summary(gambc)
```

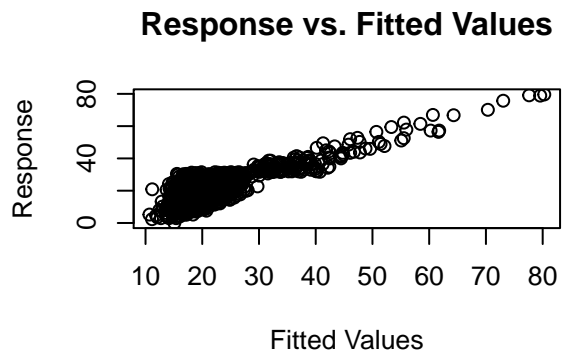
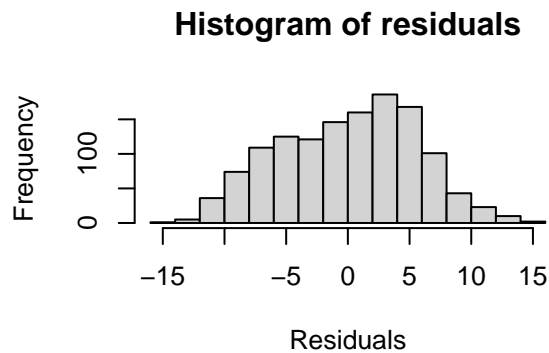
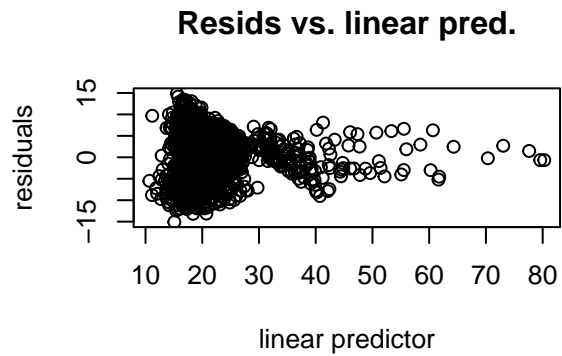
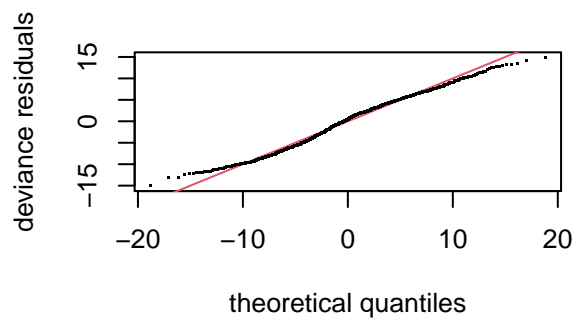
```
##
## Family: gaussian
## Link function: identity
##
```



```

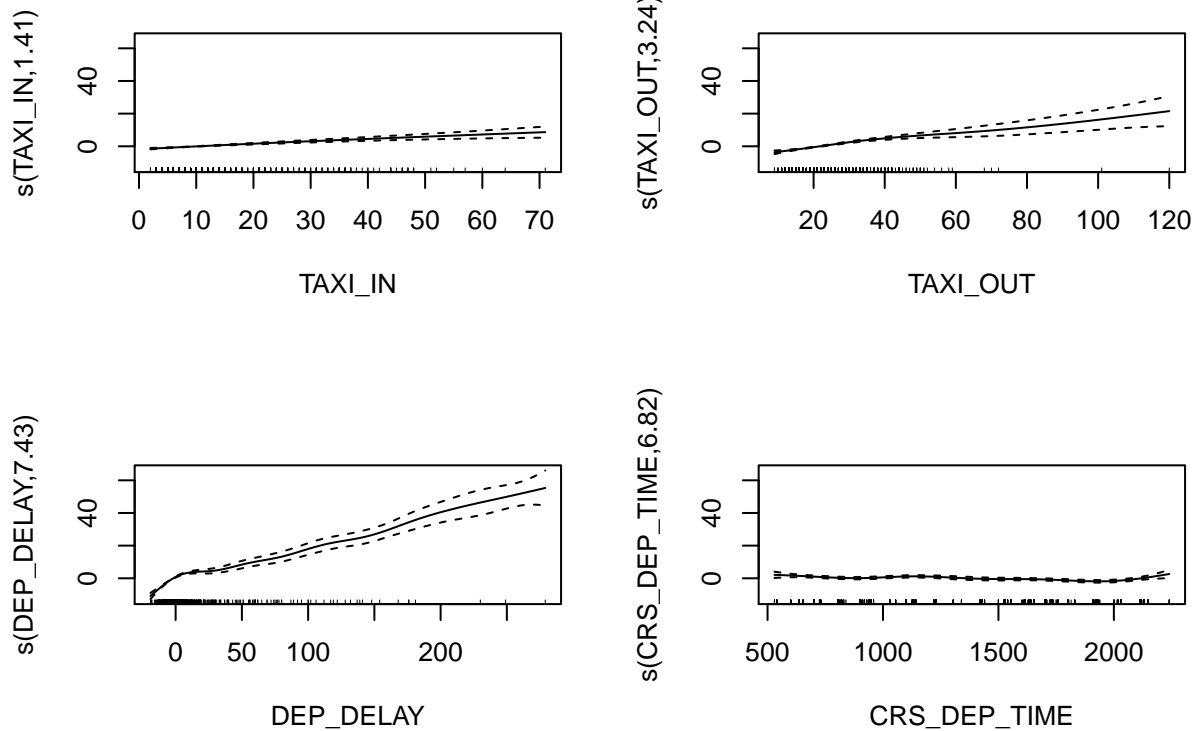
## Formula:
## bc_adj_ARR_DELAY ~ OP_CARRIER + s(TAXI_IN) + s(TAXI_OUT) + s(DEP_DELAY) +
##      s(CRS_DEP_TIME) + TYPE_DELAY
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      28.2676      1.4315  19.747 < 2e-16 ***
## OP_CARRIERAS     -0.5081      0.5258  -0.966  0.33407
## OP_CARRIERB6       0.7942      0.4274   1.858  0.06340 .
## OP_CARRIERDL     -0.9164      0.4365  -2.099  0.03598 *
## TYPE_DELAYLATE_AIRCRAFT -1.0708      2.0884  -0.513  0.60822
## TYPE_DELAYNAS       3.9305      1.4402   2.729  0.00644 **
## TYPE_DELAYNo Delay  -7.1098      1.4502  -4.903 1.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(TAXI_IN)      1.409  1.721 38.501 2.66e-14 ***
## s(TAXI_OUT)      3.241  4.048 46.660 < 2e-16 ***
## s(DEP_DELAY)     7.435  8.383 76.964 < 2e-16 ***
## s(CRS_DEP_TIME) 6.825  7.920  5.521 8.48e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.662   Deviance explained = 66.8%
## GCV = 31.856   Scale est. = 31.226     n = 1310
# diagnostic plots
par(mfrow = c(2,2))
gam.check(gambc)

```



```
##
## Method: GCV  Optimizer: magic
## Smoothing parameter selection converged after 11 iterations.
## The RMS GCV score gradient at convergence was 0.0002369166 .
## The Hessian was positive definite.
## Model rank =  43 / 43
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'   edf k-index p-value
## s(TAXI_IN)   9.00 1.41   0.99   0.36
## s(TAXI_OUT)   9.00 3.24   1.06   0.98
## s(DEP_DELAY)   9.00 7.43   0.98   0.23
## s(CRS_DEP_TIME) 9.00 6.82   0.96   0.03 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# predictor plots
par(mfrow = c(2,2))
plot(gambc)
```

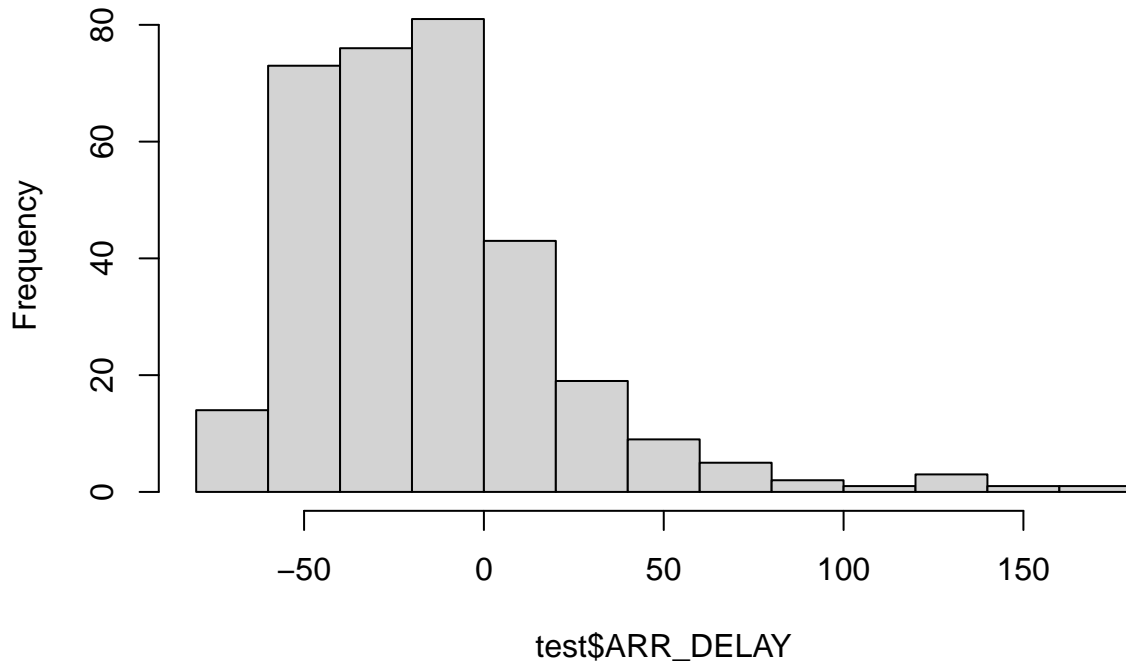


## GAM Test Error Metrics & Predictions

```
# GAM predictions
## no BC
gam_preds <- predict.gam(gam02, newdata = test)
## BC
gambc_preds <- predict.gam(gambc, newdata = test)
adjgam_preds <- ((gambc_preds*(bc_lambda) + 1)^(1/bc_lambda))
bc_gam_pred = adjgam_preds - 77

## histogram of predictions vs. actual
# actual ARR_DELAY in test set
hist(test$ARR_DELAY)
```

## Histogram of test\$ARR\_DELAY



```
pptest_actual <- ggplot(data = test, aes(x = ARR_DELAY)) +
  geom_histogram(fill = "#40E0D0", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Actual") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- baseline lm
pptest_gam <- ggplot(data = test, aes(x = gam_preds)) +
  geom_histogram(fill = "#40E0D0", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "GAM") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- lm with log-trans
pptest_gam_bc <- ggplot(data = test, aes(x = bc_gam_pred)) +
  geom_histogram(fill = "#40E0D0", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "GAM w/ Box-Cox Response") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
```

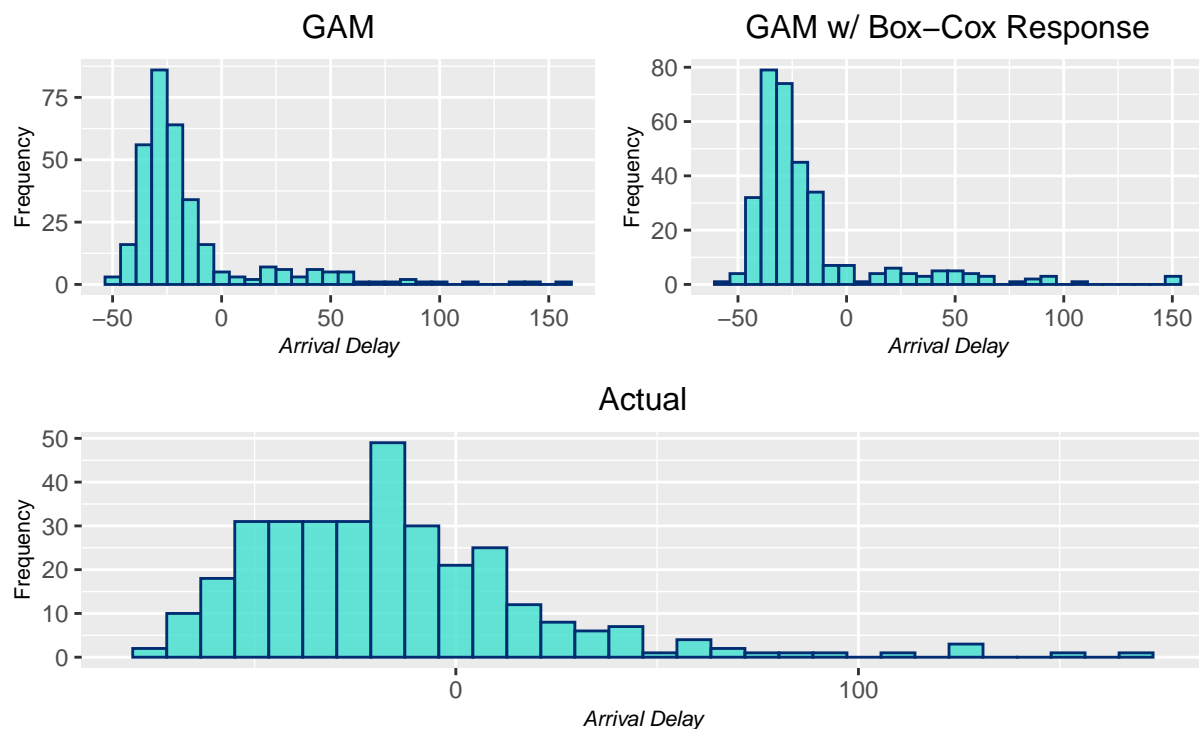
```
axis.title.x.bottom = element_text(size = 8, face = "italic"),
axis.title.y.left = element_text(size = 8))

gam_patchwork <- (ptest_gam + ptest_gam_bc) / ptest_actual
gam_patchwork + plot_annotation(
  title = 'Comparing Distributions of ARR_DELAY',
  subtitle = 'Histograms of actual test values and GAM-predicted values'
)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

## Comparing Distributions of ARR\_DELAY

Histograms of actual test values and GAM-predicted values



```
# test MSE calculations
gam_MSE <- sum((test$ARR_DELAY - gam_preds)^2, na.rm=T)/length(test$ARR_DELAY)
gam_MSE

## [1] 312.2953

gambc_MSE <- sum((test$ARR_DELAY - bc_gam_pred)^2, na.rm=T)/length(test$ARR_DELAY)
gambc_MSE

## [1] 317.4538
```

## (3) Tree-Based Models

### (a) Random Forests

By default, `randomForest()` uses  $p/3$  variables when building a random forest of regression trees.

```

set.seed(1)

# optimal number of predictors (param = mtry) used = 2 based on CV
rf.delay <- randomForest(Arr_DELAY ~ DAY_OF_MONTH +
                        TAXI_IN +
                        TAXI_OUT +
                        DEST +
                        DEP_DELAY +
                        CARRIER_DELAY +
                        NAS_DELAY,
                        data = train, na.action = na.omit, importance = TRUE,
                        ntree=10000, mtry=2)

```

Using the `importance()` function, we can view the importance of each variable.

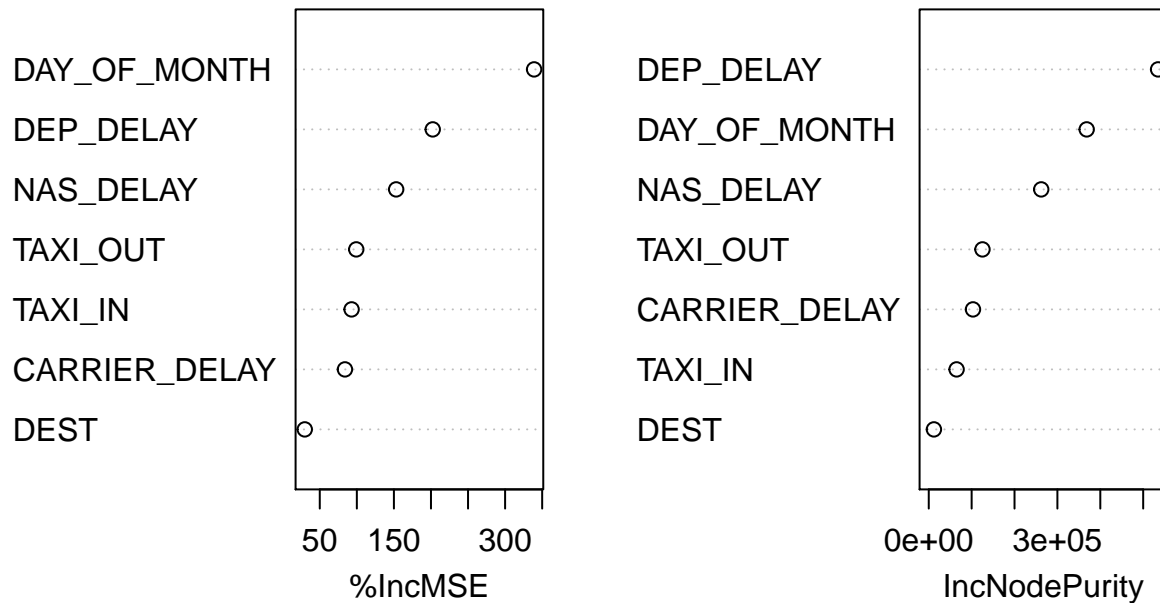
```
importance(rf.delay)
```

##		%IncMSE	IncNodePurity
##	DAY_OF_MONTH	339.08627	368261.46
##	TAXI_IN	92.93643	64905.19
##	TAXI_OUT	99.21626	125258.87
##	DEST	29.76223	12009.51
##	DEP_DELAY	202.38571	534776.35
##	CARRIER_DELAY	83.95478	103105.51
##	NAS_DELAY	153.05251	262232.63

Two measures of variable importance are reported. The former is based on the mean decrease in accuracy in predictions on the out of bag samples when a given variable is excluded from the model. The latter is a measure of the total decrease in node impurity that results from splits over that variable, averaged over all trees (this was plotted in Figure 8.9 in the text). In the case of regression trees, the node impurity is measured by the training RSS and for classification trees by the deviance. Plots of these importance measures can be produced using the `varImpPlot()` function.

```
varImpPlot(rf.delay)
```

## rf.delay



### (b) Boosting

Here we use the `gbm()` package, and within it the `gbm()` function, to fit boosted regression trees to the `train` data set. We run `gbm()` with the option `distribution = "gaussian"` since this is a regression problem. The argument `n.trees = 150` indicates that we want 150 trees, and the option `interaction.depth = 3` limits the depth of each tree.

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1150.5648	-nan	0.1000	108.0894
##	2	1062.7120	-nan	0.1000	98.0251
##	3	999.8411	-nan	0.1000	54.3611
##	4	937.5774	-nan	0.1000	53.2725
##	5	877.3090	-nan	0.1000	60.5384
##	6	834.8623	-nan	0.1000	42.6809
##	7	782.6249	-nan	0.1000	41.6720
##	8	736.9349	-nan	0.1000	47.9831
##	9	695.5806	-nan	0.1000	40.5227
##	10	657.6592	-nan	0.1000	36.4710
##	20	421.6438	-nan	0.1000	15.4628
##	40	244.7935	-nan	0.1000	1.2456
##	60	184.0732	-nan	0.1000	1.4904
##	80	155.0727	-nan	0.1000	0.9043
##	100	140.8694	-nan	0.1000	0.3294
##	120	133.2812	-nan	0.1000	0.2873
##	140	127.0999	-nan	0.1000	0.1086
##	150	124.7120	-nan	0.1000	-1.0101
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1091.1076	-nan	0.1000	130.0192

##	2	961.6851	-nan	0.1000	157.4429
##	3	858.5946	-nan	0.1000	101.8085
##	4	790.1614	-nan	0.1000	64.1696
##	5	710.8182	-nan	0.1000	78.4191
##	6	649.3869	-nan	0.1000	63.5996
##	7	598.3666	-nan	0.1000	39.1273
##	8	545.5722	-nan	0.1000	54.2270
##	9	500.1991	-nan	0.1000	34.5864
##	10	467.1205	-nan	0.1000	34.3254
##	20	287.3160	-nan	0.1000	5.7970
##	40	166.8623	-nan	0.1000	3.5853
##	60	132.0353	-nan	0.1000	0.5102
##	80	118.0411	-nan	0.1000	0.0546
##	100	110.1089	-nan	0.1000	0.7461
##	120	103.2643	-nan	0.1000	-0.5363
##	140	100.1627	-nan	0.1000	-0.0438
##	150	98.7409	-nan	0.1000	-0.4050

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1066.5079	-nan	0.1000	173.6475
##	2	923.2312	-nan	0.1000	140.7612
##	3	801.1209	-nan	0.1000	93.9866
##	4	714.0700	-nan	0.1000	88.0321
##	5	640.2316	-nan	0.1000	68.1676
##	6	569.5613	-nan	0.1000	55.4898
##	7	505.9868	-nan	0.1000	58.5620
##	8	456.4054	-nan	0.1000	49.7937
##	9	419.3526	-nan	0.1000	27.1763
##	10	388.8173	-nan	0.1000	26.1372
##	20	233.6893	-nan	0.1000	7.8144
##	40	135.9065	-nan	0.1000	1.1181
##	60	110.4129	-nan	0.1000	0.0033
##	80	101.0143	-nan	0.1000	-1.0713
##	100	93.8919	-nan	0.1000	0.3906
##	120	90.0736	-nan	0.1000	-0.2786
##	140	86.1896	-nan	0.1000	0.0213
##	150	84.9967	-nan	0.1000	-0.3078

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1038.4961	-nan	0.1000	79.2910
##	2	964.6743	-nan	0.1000	63.7220
##	3	904.4326	-nan	0.1000	63.8304
##	4	846.0564	-nan	0.1000	35.8399
##	5	796.9160	-nan	0.1000	33.8357
##	6	741.3574	-nan	0.1000	53.0141
##	7	697.1677	-nan	0.1000	42.4855
##	8	658.3158	-nan	0.1000	31.6678
##	9	620.2354	-nan	0.1000	34.9740
##	10	583.9651	-nan	0.1000	37.1194
##	20	365.3850	-nan	0.1000	16.3536
##	40	211.8031	-nan	0.1000	3.8751
##	60	162.5473	-nan	0.1000	1.5584
##	80	139.3494	-nan	0.1000	0.7306
##	100	126.2946	-nan	0.1000	0.2218



##	120	117.2882	-nan	0.1000	0.1613
##	140	110.9744	-nan	0.1000	0.3039
##	150	108.7748	-nan	0.1000	0.2844
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	977.1061	-nan	0.1000	113.4577
##	2	866.4147	-nan	0.1000	113.7672
##	3	768.6013	-nan	0.1000	79.1478
##	4	692.0857	-nan	0.1000	73.6998
##	5	626.7474	-nan	0.1000	64.8212
##	6	575.0267	-nan	0.1000	52.8815
##	7	521.4352	-nan	0.1000	40.5435
##	8	473.4024	-nan	0.1000	32.9714
##	9	435.6518	-nan	0.1000	33.5803
##	10	404.0142	-nan	0.1000	30.8877
##	20	237.3946	-nan	0.1000	9.0847
##	40	148.7742	-nan	0.1000	0.6326
##	60	116.9303	-nan	0.1000	-0.0815
##	80	103.5219	-nan	0.1000	0.3205
##	100	97.1446	-nan	0.1000	0.0267
##	120	93.7601	-nan	0.1000	-0.2853
##	140	90.1817	-nan	0.1000	-0.2491
##	150	88.4311	-nan	0.1000	0.0747
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	979.2186	-nan	0.1000	157.4003
##	2	843.8291	-nan	0.1000	140.3897
##	3	728.9358	-nan	0.1000	107.9369
##	4	657.4075	-nan	0.1000	73.6883
##	5	583.9841	-nan	0.1000	74.7565
##	6	525.9856	-nan	0.1000	47.9538
##	7	471.0073	-nan	0.1000	45.7894
##	8	421.1825	-nan	0.1000	45.8401
##	9	381.2552	-nan	0.1000	28.7010
##	10	349.2513	-nan	0.1000	25.4642
##	20	197.4749	-nan	0.1000	7.5401
##	40	118.1161	-nan	0.1000	0.2761
##	60	96.2397	-nan	0.1000	0.0293
##	80	89.0182	-nan	0.1000	0.3721
##	100	83.6028	-nan	0.1000	-0.2090
##	120	79.7778	-nan	0.1000	-0.2337
##	140	76.0114	-nan	0.1000	0.0092
##	150	74.8119	-nan	0.1000	-0.2349
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1043.1800	-nan	0.1000	65.1818
##	2	959.4253	-nan	0.1000	70.1816
##	3	896.6926	-nan	0.1000	61.6784
##	4	843.1070	-nan	0.1000	50.6219
##	5	796.5353	-nan	0.1000	40.8339
##	6	746.4385	-nan	0.1000	46.4100
##	7	702.8579	-nan	0.1000	41.7416
##	8	663.6213	-nan	0.1000	38.0004
##	9	625.0179	-nan	0.1000	39.5442

##	10	591.8549	-nan	0.1000	27.8403
##	20	378.0878	-nan	0.1000	12.1186
##	40	231.7994	-nan	0.1000	3.0440
##	60	180.3416	-nan	0.1000	1.4298
##	80	156.3828	-nan	0.1000	0.6180
##	100	144.4843	-nan	0.1000	0.4221
##	120	136.6853	-nan	0.1000	-0.0499
##	140	130.7566	-nan	0.1000	0.0179
##	150	128.6565	-nan	0.1000	-0.3876
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	988.9693	-nan	0.1000	135.6243
##	2	878.1054	-nan	0.1000	115.2549
##	3	782.7530	-nan	0.1000	68.8872
##	4	706.5929	-nan	0.1000	65.4505
##	5	647.2079	-nan	0.1000	53.1487
##	6	584.1888	-nan	0.1000	65.6355
##	7	536.8114	-nan	0.1000	43.1166
##	8	496.6526	-nan	0.1000	42.0299
##	9	460.4912	-nan	0.1000	35.7958
##	10	431.7768	-nan	0.1000	27.7186
##	20	263.1444	-nan	0.1000	9.9121
##	40	160.8285	-nan	0.1000	1.3644
##	60	131.1793	-nan	0.1000	0.8514
##	80	117.1487	-nan	0.1000	0.2811
##	100	109.9067	-nan	0.1000	-0.3520
##	120	103.3028	-nan	0.1000	-0.0456
##	140	99.4621	-nan	0.1000	-1.4216
##	150	97.8404	-nan	0.1000	-0.1779
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	963.7173	-nan	0.1000	145.6485
##	2	830.0235	-nan	0.1000	140.3019
##	3	729.2431	-nan	0.1000	112.9741
##	4	642.8786	-nan	0.1000	85.5526
##	5	579.7839	-nan	0.1000	67.1794
##	6	525.0721	-nan	0.1000	54.9593
##	7	482.4734	-nan	0.1000	44.9557
##	8	440.6059	-nan	0.1000	36.2972
##	9	405.1308	-nan	0.1000	30.6701
##	10	377.1782	-nan	0.1000	32.2158
##	20	223.1937	-nan	0.1000	8.3633
##	40	137.5210	-nan	0.1000	2.7908
##	60	115.6683	-nan	0.1000	-0.1178
##	80	103.5987	-nan	0.1000	0.1486
##	100	97.4186	-nan	0.1000	0.0468
##	120	92.1760	-nan	0.1000	-0.6028
##	140	87.8751	-nan	0.1000	-0.0790
##	150	86.0851	-nan	0.1000	-0.2352
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1114.9069	-nan	0.1000	111.0734
##	2	1026.6882	-nan	0.1000	60.7724
##	3	960.8236	-nan	0.1000	65.4554

##	4	905.6734	-nan	0.1000	56.6590
##	5	844.7358	-nan	0.1000	56.0554
##	6	790.0563	-nan	0.1000	37.1147
##	7	747.9614	-nan	0.1000	43.9995
##	8	703.2455	-nan	0.1000	37.1534
##	9	660.5436	-nan	0.1000	37.6661
##	10	620.8812	-nan	0.1000	37.8569
##	20	394.0971	-nan	0.1000	13.1611
##	40	227.3770	-nan	0.1000	3.0090
##	60	175.9880	-nan	0.1000	0.6697
##	80	153.5385	-nan	0.1000	0.4214
##	100	141.7942	-nan	0.1000	0.0400
##	120	133.9781	-nan	0.1000	0.2538
##	140	128.5789	-nan	0.1000	-0.6216
##	150	127.0912	-nan	0.1000	0.2749
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1058.4375	-nan	0.1000	179.6786
##	2	934.4213	-nan	0.1000	123.0235
##	3	826.0630	-nan	0.1000	119.9622
##	4	741.0864	-nan	0.1000	76.9391
##	5	673.9855	-nan	0.1000	59.9329
##	6	614.3471	-nan	0.1000	63.7129
##	7	565.6583	-nan	0.1000	51.6028
##	8	515.4718	-nan	0.1000	48.5894
##	9	473.3732	-nan	0.1000	39.6733
##	10	437.3717	-nan	0.1000	25.2694
##	20	255.5560	-nan	0.1000	9.7985
##	40	158.6039	-nan	0.1000	0.8082
##	60	130.2847	-nan	0.1000	1.6124
##	80	119.4850	-nan	0.1000	0.1706
##	100	113.3957	-nan	0.1000	-0.4501
##	120	107.7894	-nan	0.1000	0.0992
##	140	103.8321	-nan	0.1000	-0.2636
##	150	102.7621	-nan	0.1000	-0.2770
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1050.1920	-nan	0.1000	157.4568
##	2	908.9149	-nan	0.1000	134.2770
##	3	813.2572	-nan	0.1000	88.2433
##	4	713.7046	-nan	0.1000	97.3873
##	5	630.2140	-nan	0.1000	73.9902
##	6	555.7110	-nan	0.1000	72.0744
##	7	505.4664	-nan	0.1000	57.3740
##	8	459.7692	-nan	0.1000	50.6590
##	9	418.2648	-nan	0.1000	38.7313
##	10	381.5394	-nan	0.1000	28.0108
##	20	218.5273	-nan	0.1000	8.5399
##	40	139.6013	-nan	0.1000	1.0679
##	60	115.4817	-nan	0.1000	0.4317
##	80	105.5851	-nan	0.1000	-0.4976
##	100	98.7866	-nan	0.1000	-0.0657
##	120	94.8003	-nan	0.1000	-0.2332
##	140	92.0789	-nan	0.1000	-0.4928

```

##      150      89.9621      -nan      0.1000     -0.4581
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1363.0377      -nan      0.1000    115.9742
##      2      1261.0838      -nan      0.1000     95.4869
##      3      1169.9134      -nan      0.1000     89.4782
##      4      1098.3658      -nan      0.1000     58.2659
##      5      1033.3767      -nan      0.1000     65.9647
##      6       975.8048      -nan      0.1000     57.3396
##      7       916.9678      -nan      0.1000     51.0980
##      8       866.2389      -nan      0.1000     48.7142
##      9       822.0977      -nan      0.1000     45.7419
##     10       780.3579      -nan      0.1000     40.0106
##     20       473.4706      -nan      0.1000     21.2260
##     40       261.5182      -nan      0.1000      4.9859
##     60       188.8009      -nan      0.1000      2.0729
##     80       156.3942      -nan      0.1000      0.2772
##    100       139.4622      -nan      0.1000      0.4014
##    120       130.4820      -nan      0.1000      0.2730
##    140       123.5861      -nan      0.1000     -0.1129
##    150       120.8073      -nan      0.1000      0.0414
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1313.9267      -nan      0.1000    178.6950
##      2      1154.3750      -nan      0.1000    158.1254
##      3      1021.4058      -nan      0.1000     96.9070
##      4       907.3896      -nan      0.1000     93.8171
##      5       821.2174      -nan      0.1000     79.4315
##      6       735.9728      -nan      0.1000     65.2562
##      7       672.5909      -nan      0.1000     59.5681
##      8       625.1451      -nan      0.1000     44.8287
##      9       569.7428      -nan      0.1000     37.7258
##     10       521.7478      -nan      0.1000     38.0977
##     20       296.2888      -nan      0.1000     12.0787
##     40       167.2275      -nan      0.1000      1.9542
##     60       133.7161      -nan      0.1000      0.8959
##     80       115.3132      -nan      0.1000     -0.3287
##    100       107.6759      -nan      0.1000      0.3238
##    120       102.2019      -nan      0.1000     -0.2904
##    140        96.2270      -nan      0.1000      0.0778
##    150        94.6262      -nan      0.1000     -0.3259
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1284.6787      -nan      0.1000    216.3611
##      2      1110.8225      -nan      0.1000    156.8866
##      3       957.2629      -nan      0.1000    157.3410
##      4       837.9827      -nan      0.1000    107.2450
##      5       739.2871      -nan      0.1000     94.1477
##      6       657.4487      -nan      0.1000     70.9209
##      7       592.1601      -nan      0.1000     65.7733
##      8       533.5499      -nan      0.1000     54.9894
##      9       481.1973      -nan      0.1000     54.6397
##     10       450.3253      -nan      0.1000     32.3668
##     20       242.1455      -nan      0.1000     16.5360

```

##	40	134.1906	-nan	0.1000	1.2931
##	60	107.5346	-nan	0.1000	0.6984
##	80	96.4049	-nan	0.1000	0.4291
##	100	88.7862	-nan	0.1000	0.3195
##	120	84.0447	-nan	0.1000	-0.3354
##	140	79.9662	-nan	0.1000	-0.0714
##	150	78.3658	-nan	0.1000	-0.0550
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1119.6872	-nan	0.1000	64.0268
##	2	1028.6291	-nan	0.1000	89.2680
##	3	955.2203	-nan	0.1000	51.7304
##	4	886.0096	-nan	0.1000	76.9672
##	5	833.6934	-nan	0.1000	52.1605
##	6	792.9316	-nan	0.1000	36.7743
##	7	744.9836	-nan	0.1000	38.6176
##	8	697.4934	-nan	0.1000	37.4781
##	9	656.1072	-nan	0.1000	39.4088
##	10	622.1726	-nan	0.1000	33.2767
##	20	396.7593	-nan	0.1000	17.3952
##	40	235.1882	-nan	0.1000	1.0045
##	60	179.7141	-nan	0.1000	1.2229
##	80	154.9271	-nan	0.1000	0.6423
##	100	143.1362	-nan	0.1000	0.2508
##	120	135.1987	-nan	0.1000	0.0048
##	140	129.8053	-nan	0.1000	-0.5869
##	150	127.0031	-nan	0.1000	0.1124
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1030.7566	-nan	0.1000	138.8045
##	2	931.4097	-nan	0.1000	108.5589
##	3	823.2101	-nan	0.1000	119.9196
##	4	737.1424	-nan	0.1000	83.8840
##	5	682.1829	-nan	0.1000	52.8919
##	6	617.9732	-nan	0.1000	55.2543
##	7	564.1432	-nan	0.1000	44.3180
##	8	518.3703	-nan	0.1000	41.4856
##	9	485.0086	-nan	0.1000	35.0296
##	10	459.1011	-nan	0.1000	24.7205
##	20	265.7766	-nan	0.1000	6.5308
##	40	163.6818	-nan	0.1000	1.6940
##	60	134.7479	-nan	0.1000	0.6209
##	80	121.5619	-nan	0.1000	0.4184
##	100	111.6734	-nan	0.1000	-0.3092
##	120	106.5058	-nan	0.1000	-0.0169
##	140	101.4909	-nan	0.1000	0.1760
##	150	98.6352	-nan	0.1000	0.0204
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1030.4886	-nan	0.1000	169.4195
##	2	894.3337	-nan	0.1000	147.4280
##	3	781.2207	-nan	0.1000	99.2478
##	4	692.5225	-nan	0.1000	86.5989
##	5	614.2082	-nan	0.1000	74.9347

##	6	556.0481	-nan	0.1000	53.3782
##	7	506.4840	-nan	0.1000	45.6967
##	8	456.4495	-nan	0.1000	44.0783
##	9	425.2643	-nan	0.1000	34.2144
##	10	394.9264	-nan	0.1000	30.4285
##	20	233.0086	-nan	0.1000	7.0539
##	40	142.0020	-nan	0.1000	1.1186
##	60	116.1049	-nan	0.1000	0.5247
##	80	102.4701	-nan	0.1000	-0.5115
##	100	94.1522	-nan	0.1000	0.1074
##	120	88.0607	-nan	0.1000	0.2569
##	140	84.4240	-nan	0.1000	0.1709
##	150	82.9162	-nan	0.1000	-0.0851
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1151.1140	-nan	0.1000	74.8637
##	2	1073.9338	-nan	0.1000	73.2535
##	3	1014.7446	-nan	0.1000	60.9543
##	4	947.6675	-nan	0.1000	59.7251
##	5	902.7024	-nan	0.1000	48.6272
##	6	840.8978	-nan	0.1000	61.2264
##	7	795.5811	-nan	0.1000	39.0083
##	8	748.1772	-nan	0.1000	48.2803
##	9	709.2787	-nan	0.1000	36.1646
##	10	675.0291	-nan	0.1000	35.6680
##	20	425.3133	-nan	0.1000	16.5287
##	40	244.8662	-nan	0.1000	1.7448
##	60	187.7142	-nan	0.1000	1.3188
##	80	158.9160	-nan	0.1000	0.8889
##	100	145.4590	-nan	0.1000	0.4266
##	120	136.4194	-nan	0.1000	0.2414
##	140	130.0089	-nan	0.1000	0.3067
##	150	128.3017	-nan	0.1000	-1.5994
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1089.0163	-nan	0.1000	147.1273
##	2	969.9221	-nan	0.1000	116.2787
##	3	880.5462	-nan	0.1000	78.6197
##	4	812.5230	-nan	0.1000	59.9873
##	5	726.9211	-nan	0.1000	84.5285
##	6	677.4496	-nan	0.1000	52.2021
##	7	612.9353	-nan	0.1000	50.6274
##	8	561.4688	-nan	0.1000	48.4978
##	9	512.3283	-nan	0.1000	45.8716
##	10	483.8197	-nan	0.1000	29.7787
##	20	277.0257	-nan	0.1000	10.2003
##	40	170.0154	-nan	0.1000	0.9824
##	60	138.0782	-nan	0.1000	0.1468
##	80	121.8979	-nan	0.1000	1.0083
##	100	113.0314	-nan	0.1000	0.7136
##	120	106.1635	-nan	0.1000	0.4506
##	140	101.5379	-nan	0.1000	-0.1965
##	150	99.3361	-nan	0.1000	0.0973
##					

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1076.3660	-nan	0.1000	166.0326
##	2	934.8358	-nan	0.1000	127.5496
##	3	821.5449	-nan	0.1000	106.6992
##	4	746.0042	-nan	0.1000	77.6186
##	5	662.0705	-nan	0.1000	88.5528
##	6	587.9474	-nan	0.1000	68.8832
##	7	524.7118	-nan	0.1000	57.8505
##	8	479.7467	-nan	0.1000	47.0668
##	9	436.0507	-nan	0.1000	39.3705
##	10	399.8243	-nan	0.1000	31.9315
##	20	220.4503	-nan	0.1000	6.6917
##	40	137.8457	-nan	0.1000	0.0340
##	60	111.5257	-nan	0.1000	0.4765
##	80	103.2541	-nan	0.1000	0.3290
##	100	95.8835	-nan	0.1000	0.0615
##	120	90.3473	-nan	0.1000	-1.0049
##	140	86.4337	-nan	0.1000	-0.4626
##	150	84.6927	-nan	0.1000	-0.2075
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1339.4377	-nan	0.1000	113.2613
##	2	1216.2615	-nan	0.1000	99.2026
##	3	1126.4295	-nan	0.1000	68.4633
##	4	1046.2734	-nan	0.1000	82.6301
##	5	983.9641	-nan	0.1000	62.1021
##	6	914.1832	-nan	0.1000	58.4017
##	7	852.3048	-nan	0.1000	55.8210
##	8	808.5896	-nan	0.1000	38.0359
##	9	761.6549	-nan	0.1000	46.5411
##	10	717.0981	-nan	0.1000	45.1370
##	20	436.3161	-nan	0.1000	5.0169
##	40	247.8442	-nan	0.1000	4.0908
##	60	189.6155	-nan	0.1000	1.2188
##	80	163.7345	-nan	0.1000	-0.4299
##	100	151.1381	-nan	0.1000	0.3984
##	120	143.2722	-nan	0.1000	0.3737
##	140	137.0679	-nan	0.1000	0.2390
##	150	134.8423	-nan	0.1000	-0.2984
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1293.9024	-nan	0.1000	152.8322
##	2	1141.1291	-nan	0.1000	163.8377
##	3	1004.9357	-nan	0.1000	125.6723
##	4	904.6677	-nan	0.1000	101.5808
##	5	806.7888	-nan	0.1000	95.9764
##	6	727.2010	-nan	0.1000	56.4767
##	7	671.4384	-nan	0.1000	51.6276
##	8	611.6293	-nan	0.1000	47.2025
##	9	561.1445	-nan	0.1000	48.9747
##	10	523.9407	-nan	0.1000	41.5284
##	20	290.1424	-nan	0.1000	8.3678
##	40	169.4780	-nan	0.1000	2.2472
##	60	139.4834	-nan	0.1000	0.5103

##	80	124.4535	-nan	0.1000	0.4551
##	100	116.3915	-nan	0.1000	0.0335
##	120	109.1921	-nan	0.1000	-0.3405
##	140	105.2343	-nan	0.1000	-0.4163
##	150	103.8512	-nan	0.1000	-0.3317
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1267.3354	-nan	0.1000	185.0019
##	2	1107.6952	-nan	0.1000	184.8420
##	3	963.0019	-nan	0.1000	143.9640
##	4	842.8009	-nan	0.1000	109.3526
##	5	751.5434	-nan	0.1000	106.4052
##	6	668.2674	-nan	0.1000	99.3912
##	7	596.6896	-nan	0.1000	53.4083
##	8	542.7084	-nan	0.1000	57.2947
##	9	501.0338	-nan	0.1000	42.9532
##	10	455.1269	-nan	0.1000	46.6512
##	20	255.6455	-nan	0.1000	8.0047
##	40	152.1644	-nan	0.1000	0.5435
##	60	122.5953	-nan	0.1000	0.5605
##	80	109.4411	-nan	0.1000	0.0806
##	100	102.2738	-nan	0.1000	0.0991
##	120	96.9834	-nan	0.1000	-0.6580
##	140	92.7473	-nan	0.1000	-0.6350
##	150	90.8796	-nan	0.1000	-0.2192
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1309.6891	-nan	0.1000	119.7429
##	2	1192.8718	-nan	0.1000	131.6083
##	3	1106.9567	-nan	0.1000	79.0799
##	4	1030.6708	-nan	0.1000	83.9500
##	5	960.5429	-nan	0.1000	60.8600
##	6	892.6716	-nan	0.1000	56.2380
##	7	833.7406	-nan	0.1000	62.1241
##	8	785.0503	-nan	0.1000	50.2280
##	9	731.9059	-nan	0.1000	58.2727
##	10	686.1060	-nan	0.1000	47.9452
##	20	400.7869	-nan	0.1000	13.7796
##	40	224.7410	-nan	0.1000	2.2285
##	60	170.6971	-nan	0.1000	1.0392
##	80	145.5117	-nan	0.1000	0.8516
##	100	132.2849	-nan	0.1000	0.1497
##	120	123.5844	-nan	0.1000	-0.1524
##	140	117.6299	-nan	0.1000	0.0698
##	150	115.2245	-nan	0.1000	-0.0096
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1215.0085	-nan	0.1000	178.3146
##	2	1083.5683	-nan	0.1000	136.7143
##	3	980.2517	-nan	0.1000	107.0384
##	4	875.5395	-nan	0.1000	91.5845
##	5	777.3021	-nan	0.1000	89.5319
##	6	697.5710	-nan	0.1000	81.7018
##	7	629.7203	-nan	0.1000	48.0097



##	8	574.1415	-nan	0.1000	61.9125
##	9	524.1105	-nan	0.1000	43.0415
##	10	487.4983	-nan	0.1000	38.4445
##	20	262.9601	-nan	0.1000	5.6366
##	40	159.6440	-nan	0.1000	1.8309
##	60	129.1143	-nan	0.1000	0.1300
##	80	114.9550	-nan	0.1000	0.0602
##	100	106.5549	-nan	0.1000	0.1337
##	120	101.1085	-nan	0.1000	-0.0446
##	140	96.6816	-nan	0.1000	-0.2191
##	150	94.8257	-nan	0.1000	-0.4427

##

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1202.8157	-nan	0.1000	211.6199
##	2	1043.5734	-nan	0.1000	158.7406
##	3	894.0216	-nan	0.1000	117.4613
##	4	778.1679	-nan	0.1000	113.2346
##	5	696.3307	-nan	0.1000	97.7456
##	6	617.0662	-nan	0.1000	76.7895
##	7	548.1352	-nan	0.1000	55.9634
##	8	497.3939	-nan	0.1000	48.5479
##	9	452.4059	-nan	0.1000	43.9317
##	10	409.5711	-nan	0.1000	38.9295
##	20	226.4281	-nan	0.1000	9.2007
##	40	131.8420	-nan	0.1000	1.4361
##	60	107.8706	-nan	0.1000	0.4472
##	80	97.8616	-nan	0.1000	0.1277
##	100	90.9571	-nan	0.1000	-0.1795
##	120	86.1733	-nan	0.1000	-0.4309
##	140	81.2866	-nan	0.1000	-0.3627
##	150	79.7317	-nan	0.1000	-0.1314

##

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1197.8704	-nan	0.1000	105.9670
##	2	1116.4273	-nan	0.1000	79.7892
##	3	1039.6990	-nan	0.1000	79.0002
##	4	967.4720	-nan	0.1000	69.3039
##	5	914.7650	-nan	0.1000	54.4291
##	6	861.5939	-nan	0.1000	60.7256
##	7	806.6787	-nan	0.1000	52.6622
##	8	764.8375	-nan	0.1000	42.8954
##	9	732.0972	-nan	0.1000	34.8340
##	10	682.1197	-nan	0.1000	22.3085
##	20	422.4240	-nan	0.1000	15.8138
##	40	244.9176	-nan	0.1000	2.6851
##	60	186.9602	-nan	0.1000	1.5573
##	80	160.2296	-nan	0.1000	0.5846
##	100	147.3465	-nan	0.1000	-0.0290
##	120	140.7966	-nan	0.1000	-0.7363
##	140	135.9467	-nan	0.1000	0.0404
##	150	133.1098	-nan	0.1000	0.0520

##

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1138.3148	-nan	0.1000	161.1864

##	2	1003.7702	-nan	0.1000	111.5627
##	3	885.9315	-nan	0.1000	88.6904
##	4	787.9267	-nan	0.1000	80.9427
##	5	718.3539	-nan	0.1000	73.0769
##	6	654.9425	-nan	0.1000	66.9388
##	7	593.8499	-nan	0.1000	57.3728
##	8	552.5312	-nan	0.1000	42.2236
##	9	506.1835	-nan	0.1000	32.1548
##	10	470.7422	-nan	0.1000	20.3838
##	20	280.8118	-nan	0.1000	10.4112
##	40	168.5224	-nan	0.1000	1.9261
##	60	138.6952	-nan	0.1000	0.5653
##	80	125.8292	-nan	0.1000	0.5666
##	100	116.6583	-nan	0.1000	-0.0397
##	120	110.5151	-nan	0.1000	-0.0843
##	140	104.5755	-nan	0.1000	-0.5364
##	150	102.4542	-nan	0.1000	-0.0456
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1122.9821	-nan	0.1000	178.7064
##	2	977.8391	-nan	0.1000	140.7097
##	3	857.8848	-nan	0.1000	101.6812
##	4	750.4677	-nan	0.1000	97.1855
##	5	662.0855	-nan	0.1000	56.7001
##	6	585.8205	-nan	0.1000	66.1089
##	7	536.4547	-nan	0.1000	49.5552
##	8	486.1739	-nan	0.1000	47.3448
##	9	446.9455	-nan	0.1000	40.6060
##	10	411.8919	-nan	0.1000	34.5107
##	20	234.3615	-nan	0.1000	8.2405
##	40	140.8457	-nan	0.1000	0.6081
##	60	115.9656	-nan	0.1000	-0.3337
##	80	105.0992	-nan	0.1000	-0.0829
##	100	99.1904	-nan	0.1000	-0.3598
##	120	93.7467	-nan	0.1000	-0.6929
##	140	89.7672	-nan	0.1000	-0.1951
##	150	88.6174	-nan	0.1000	-0.6462
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1023.4582	-nan	0.1000	68.3441
##	2	967.9584	-nan	0.1000	59.5204
##	3	909.0685	-nan	0.1000	63.8819
##	4	854.1288	-nan	0.1000	43.4207
##	5	810.2001	-nan	0.1000	44.7913
##	6	765.1487	-nan	0.1000	46.8162
##	7	725.9762	-nan	0.1000	37.0026
##	8	675.6022	-nan	0.1000	31.4953
##	9	644.0094	-nan	0.1000	28.9816
##	10	616.3479	-nan	0.1000	29.9801
##	20	393.2961	-nan	0.1000	8.9774
##	40	235.7948	-nan	0.1000	3.5970
##	60	179.8041	-nan	0.1000	1.5017
##	80	150.9343	-nan	0.1000	0.7221
##	100	136.3889	-nan	0.1000	-1.4436

##	120	129.0320	-nan	0.1000	0.1599
##	140	123.3141	-nan	0.1000	0.2362
##	150	121.3434	-nan	0.1000	-0.7116
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	982.6173	-nan	0.1000	147.4154
##	2	873.1261	-nan	0.1000	93.7166
##	3	780.8316	-nan	0.1000	83.4734
##	4	705.9900	-nan	0.1000	79.7419
##	5	650.7736	-nan	0.1000	59.2770
##	6	603.7625	-nan	0.1000	46.4562
##	7	557.2159	-nan	0.1000	44.0364
##	8	507.2621	-nan	0.1000	38.9514
##	9	458.6382	-nan	0.1000	36.5356
##	10	425.9856	-nan	0.1000	32.0136
##	20	268.7626	-nan	0.1000	10.8314
##	40	163.7822	-nan	0.1000	1.5170
##	60	129.1078	-nan	0.1000	-0.2815
##	80	115.6694	-nan	0.1000	-0.5982
##	100	108.3858	-nan	0.1000	0.0414
##	120	102.1804	-nan	0.1000	0.2199
##	140	97.5815	-nan	0.1000	-1.0489
##	150	95.6573	-nan	0.1000	-0.4461
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	960.5786	-nan	0.1000	154.4063
##	2	843.3030	-nan	0.1000	117.0651
##	3	741.9885	-nan	0.1000	110.7751
##	4	647.6381	-nan	0.1000	82.1691
##	5	577.1404	-nan	0.1000	57.7678
##	6	518.5489	-nan	0.1000	49.1300
##	7	471.1645	-nan	0.1000	41.3249
##	8	436.8741	-nan	0.1000	25.9718
##	9	404.6363	-nan	0.1000	25.0365
##	10	374.0788	-nan	0.1000	25.2631
##	20	221.2741	-nan	0.1000	8.9664
##	40	130.5958	-nan	0.1000	1.2295
##	60	104.8111	-nan	0.1000	-0.9402
##	80	96.0620	-nan	0.1000	-0.1090
##	100	89.7521	-nan	0.1000	-0.1655
##	120	85.2111	-nan	0.1000	-0.5734
##	140	81.8484	-nan	0.1000	-0.4011
##	150	79.9122	-nan	0.1000	0.0527
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1173.3057	-nan	0.1000	53.0337
##	2	1104.1929	-nan	0.1000	75.5040
##	3	1034.8149	-nan	0.1000	73.8075
##	4	976.3153	-nan	0.1000	57.9518
##	5	916.5336	-nan	0.1000	55.0927
##	6	871.8483	-nan	0.1000	46.3780
##	7	820.0678	-nan	0.1000	41.5526
##	8	777.0558	-nan	0.1000	52.3215
##	9	733.4139	-nan	0.1000	29.6760

##	10	698.0279	-nan	0.1000	33.0200
##	20	441.0445	-nan	0.1000	16.1338
##	40	264.6720	-nan	0.1000	2.5522
##	60	199.0969	-nan	0.1000	1.4790
##	80	170.6772	-nan	0.1000	0.8804
##	100	156.7461	-nan	0.1000	0.2365
##	120	148.7043	-nan	0.1000	-0.8564
##	140	142.8518	-nan	0.1000	0.1958
##	150	140.7197	-nan	0.1000	-0.0893
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1172.7678	-nan	0.1000	100.4373
##	2	1038.9700	-nan	0.1000	143.6029
##	3	956.5260	-nan	0.1000	68.2986
##	4	860.1925	-nan	0.1000	92.9443
##	5	776.9312	-nan	0.1000	62.7130
##	6	701.8254	-nan	0.1000	71.9763
##	7	639.9996	-nan	0.1000	59.3219
##	8	586.9620	-nan	0.1000	39.7598
##	9	540.6001	-nan	0.1000	41.4294
##	10	509.9882	-nan	0.1000	31.6544
##	20	310.5632	-nan	0.1000	11.7259
##	40	187.8151	-nan	0.1000	2.0228
##	60	150.1506	-nan	0.1000	-0.0571
##	80	133.2271	-nan	0.1000	0.3945
##	100	122.1332	-nan	0.1000	0.6666
##	120	117.1130	-nan	0.1000	-0.1163
##	140	112.4926	-nan	0.1000	-0.3512
##	150	111.4169	-nan	0.1000	-0.2453
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1098.0963	-nan	0.1000	136.6717
##	2	1001.6572	-nan	0.1000	110.3699
##	3	876.4455	-nan	0.1000	116.3827
##	4	768.7673	-nan	0.1000	89.0480
##	5	694.6607	-nan	0.1000	69.1399
##	6	628.2615	-nan	0.1000	65.2577
##	7	566.9362	-nan	0.1000	58.2404
##	8	523.7986	-nan	0.1000	44.3024
##	9	476.6739	-nan	0.1000	34.2358
##	10	441.0850	-nan	0.1000	36.1024
##	20	252.1497	-nan	0.1000	9.0238
##	40	151.3068	-nan	0.1000	1.0013
##	60	126.2079	-nan	0.1000	0.3360
##	80	115.3667	-nan	0.1000	-0.8247
##	100	106.5691	-nan	0.1000	0.0017
##	120	101.4904	-nan	0.1000	0.1990
##	140	97.1520	-nan	0.1000	-0.1910
##	150	95.2360	-nan	0.1000	-0.2821
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1201.4410	-nan	0.1000	106.1190
##	2	1112.0699	-nan	0.1000	89.6509
##	3	1043.7033	-nan	0.1000	68.3250

##	4	987.0400	-nan	0.1000	55.9010
##	5	924.1874	-nan	0.1000	53.4938
##	6	877.9404	-nan	0.1000	48.3207
##	7	823.8676	-nan	0.1000	46.5479
##	8	773.5133	-nan	0.1000	50.9513
##	9	726.6435	-nan	0.1000	41.1811
##	10	687.2881	-nan	0.1000	32.1867
##	20	428.2893	-nan	0.1000	16.4960
##	40	247.9174	-nan	0.1000	3.8608
##	60	183.2643	-nan	0.1000	1.8177
##	80	153.9239	-nan	0.1000	0.7093
##	100	139.5839	-nan	0.1000	-0.0992
##	120	130.6775	-nan	0.1000	-0.0728
##	140	124.8761	-nan	0.1000	-0.5797
##	150	122.2978	-nan	0.1000	0.0473
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1142.7927	-nan	0.1000	151.0889
##	2	1014.7561	-nan	0.1000	124.4026
##	3	908.0770	-nan	0.1000	88.4686
##	4	812.2367	-nan	0.1000	92.8222
##	5	754.5227	-nan	0.1000	55.9591
##	6	686.4518	-nan	0.1000	64.5867
##	7	626.9016	-nan	0.1000	60.8306
##	8	568.7257	-nan	0.1000	39.6262
##	9	533.7047	-nan	0.1000	38.9119
##	10	490.3844	-nan	0.1000	25.6961
##	20	280.8341	-nan	0.1000	11.5409
##	40	162.4504	-nan	0.1000	2.3763
##	60	125.9654	-nan	0.1000	-0.4606
##	80	113.8271	-nan	0.1000	0.2730
##	100	107.0324	-nan	0.1000	-0.2279
##	120	100.7149	-nan	0.1000	-0.0411
##	140	97.1166	-nan	0.1000	-0.1578
##	150	95.0111	-nan	0.1000	-0.1561
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1156.1045	-nan	0.1000	168.0392
##	2	991.6610	-nan	0.1000	178.6425
##	3	873.8349	-nan	0.1000	122.0701
##	4	775.5410	-nan	0.1000	89.8189
##	5	685.0005	-nan	0.1000	82.8261
##	6	624.5754	-nan	0.1000	57.5314
##	7	558.5024	-nan	0.1000	43.5408
##	8	503.7201	-nan	0.1000	45.5906
##	9	461.8048	-nan	0.1000	49.5252
##	10	429.3099	-nan	0.1000	32.1101
##	20	226.6858	-nan	0.1000	10.5130
##	40	129.7483	-nan	0.1000	1.1179
##	60	106.9612	-nan	0.1000	0.1356
##	80	96.7065	-nan	0.1000	0.0488
##	100	91.2641	-nan	0.1000	0.0641
##	120	87.7454	-nan	0.1000	-0.2613
##	140	84.6140	-nan	0.1000	0.0046

```

##      150      83.3537      -nan      0.1000     -0.1391
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1035.7901      -nan      0.1000     58.7296
##      2       965.8432      -nan      0.1000     50.9052
##      3       906.2053      -nan      0.1000     65.9404
##      4       842.6515      -nan      0.1000     62.7205
##      5       795.6941      -nan      0.1000     43.9271
##      6       751.0854      -nan      0.1000     44.0143
##      7       705.3684      -nan      0.1000     43.3203
##      8       662.9367      -nan      0.1000     29.9252
##      9       628.5286      -nan      0.1000     23.8324
##     10       591.7874      -nan      0.1000     36.1055
##     20       380.0726      -nan      0.1000     13.3722
##     40       215.3355      -nan      0.1000      3.7943
##     60       164.8433      -nan      0.1000      1.1424
##     80       140.1305      -nan      0.1000      0.1823
##    100       127.3285      -nan      0.1000      0.3753
##    120       118.5926      -nan      0.1000      0.1254
##    140       112.7833      -nan      0.1000      0.2753
##    150       110.4796      -nan      0.1000      0.1521
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1       984.4417      -nan      0.1000    104.8746
##      2       876.2773      -nan      0.1000    106.6737
##      3       801.2080      -nan      0.1000     77.5352
##      4       712.2964      -nan      0.1000     79.4696
##      5       643.4845      -nan      0.1000     42.7919
##      6       579.3777      -nan      0.1000     58.4736
##      7       525.9397      -nan      0.1000     40.6531
##      8       478.4460      -nan      0.1000     44.2109
##      9       448.4120      -nan      0.1000     31.1903
##     10       418.4793      -nan      0.1000     30.5114
##     20       248.7991      -nan      0.1000      6.4246
##     40       150.1354      -nan      0.1000      0.9824
##     60       119.8076      -nan      0.1000      0.0967
##     80       107.4000      -nan      0.1000     -0.7326
##    100        99.6906      -nan      0.1000     -0.2750
##    120        94.1053      -nan      0.1000      0.2156
##    140        91.0177      -nan      0.1000     -0.1580
##    150        89.8213      -nan      0.1000      0.1896
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1       951.9572      -nan      0.1000    128.8859
##      2       842.2726      -nan      0.1000    117.5267
##      3       744.2457      -nan      0.1000     88.7939
##      4       659.7421      -nan      0.1000     79.6048
##      5       595.3616      -nan      0.1000     65.9342
##      6       547.8268      -nan      0.1000     49.5625
##      7       501.8491      -nan      0.1000     45.1066
##      8       453.5627      -nan      0.1000     53.1859
##      9       408.1017      -nan      0.1000     38.3713
##     10       376.2601      -nan      0.1000     26.7730
##     20       209.4982      -nan      0.1000      8.3822

```

##	40	123.2374	-nan	0.1000	2.6032
##	60	97.5851	-nan	0.1000	0.9637
##	80	87.1513	-nan	0.1000	0.1307
##	100	80.4227	-nan	0.1000	-0.0476
##	120	75.7581	-nan	0.1000	-0.0517
##	140	72.9355	-nan	0.1000	-0.2326
##	150	71.9447	-nan	0.1000	-0.1492
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1201.6287	-nan	0.1000	90.9115
##	2	1119.7142	-nan	0.1000	83.7937
##	3	1040.6954	-nan	0.1000	87.8484
##	4	985.5103	-nan	0.1000	58.0410
##	5	923.7542	-nan	0.1000	56.8231
##	6	870.1986	-nan	0.1000	47.7431
##	7	813.9796	-nan	0.1000	59.0307
##	8	771.3084	-nan	0.1000	45.1861
##	9	734.6957	-nan	0.1000	38.0202
##	10	691.2948	-nan	0.1000	45.7223
##	20	426.1964	-nan	0.1000	16.5375
##	40	247.8649	-nan	0.1000	3.8290
##	60	185.7757	-nan	0.1000	1.6255
##	80	156.0575	-nan	0.1000	0.7319
##	100	142.2254	-nan	0.1000	-0.8207
##	120	134.5285	-nan	0.1000	0.2423
##	140	128.5601	-nan	0.1000	0.1166
##	150	126.1168	-nan	0.1000	0.2170
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1157.7912	-nan	0.1000	143.5886
##	2	1022.0738	-nan	0.1000	138.6245
##	3	905.0947	-nan	0.1000	97.2034
##	4	809.8742	-nan	0.1000	90.6408
##	5	728.5079	-nan	0.1000	73.5228
##	6	655.3187	-nan	0.1000	60.1245
##	7	604.2588	-nan	0.1000	35.3958
##	8	556.4355	-nan	0.1000	45.5072
##	9	511.7440	-nan	0.1000	45.0474
##	10	476.4448	-nan	0.1000	27.4272
##	20	272.3418	-nan	0.1000	10.5469
##	40	165.1427	-nan	0.1000	3.5678
##	60	133.0993	-nan	0.1000	-0.6801
##	80	121.5918	-nan	0.1000	-0.0320
##	100	114.5075	-nan	0.1000	0.0147
##	120	109.9636	-nan	0.1000	-0.1104
##	140	104.0401	-nan	0.1000	0.0509
##	150	102.6555	-nan	0.1000	0.0085
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1133.5197	-nan	0.1000	172.7048
##	2	987.1972	-nan	0.1000	143.5850
##	3	872.7645	-nan	0.1000	118.7208
##	4	764.0503	-nan	0.1000	112.4616
##	5	665.5178	-nan	0.1000	92.4002

##	6	592.2673	-nan	0.1000	80.0497
##	7	536.5960	-nan	0.1000	64.6191
##	8	485.7798	-nan	0.1000	47.8213
##	9	445.0319	-nan	0.1000	41.1958
##	10	411.4633	-nan	0.1000	31.3130
##	20	232.8683	-nan	0.1000	9.9203
##	40	137.9395	-nan	0.1000	1.1614
##	60	115.5033	-nan	0.1000	-0.2852
##	80	103.5464	-nan	0.1000	-0.1662
##	100	96.6148	-nan	0.1000	0.3704
##	120	92.0743	-nan	0.1000	-0.4650
##	140	87.9470	-nan	0.1000	-0.5296
##	150	86.5364	-nan	0.1000	0.1443
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1072.0599	-nan	0.1000	83.9007
##	2	995.1452	-nan	0.1000	75.6876
##	3	930.6467	-nan	0.1000	60.1760
##	4	882.8934	-nan	0.1000	36.4684
##	5	826.8277	-nan	0.1000	57.7570
##	6	777.8295	-nan	0.1000	45.0836
##	7	738.5092	-nan	0.1000	38.4305
##	8	694.2602	-nan	0.1000	35.9653
##	9	657.8653	-nan	0.1000	14.7778
##	10	614.2662	-nan	0.1000	45.3666
##	20	381.4929	-nan	0.1000	13.7471
##	40	225.4717	-nan	0.1000	2.9268
##	60	169.7447	-nan	0.1000	1.1790
##	80	143.4027	-nan	0.1000	0.1550
##	100	130.2860	-nan	0.1000	0.3389
##	120	121.7222	-nan	0.1000	0.2099
##	140	115.8403	-nan	0.1000	-0.0558
##	150	113.2317	-nan	0.1000	0.2352
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1019.1468	-nan	0.1000	93.3516
##	2	923.9510	-nan	0.1000	96.2785
##	3	817.5182	-nan	0.1000	95.1558
##	4	746.5360	-nan	0.1000	67.5560
##	5	669.3585	-nan	0.1000	64.8072
##	6	605.2731	-nan	0.1000	61.6085
##	7	552.7118	-nan	0.1000	31.8123
##	8	509.3889	-nan	0.1000	34.3121
##	9	473.4643	-nan	0.1000	37.5760
##	10	434.2131	-nan	0.1000	32.7532
##	20	250.3380	-nan	0.1000	9.0373
##	40	149.8441	-nan	0.1000	2.3425
##	60	121.5830	-nan	0.1000	0.4301
##	80	107.6926	-nan	0.1000	-0.6998
##	100	100.4119	-nan	0.1000	0.0089
##	120	96.2707	-nan	0.1000	-0.3374
##	140	90.8735	-nan	0.1000	-0.0021
##	150	90.0521	-nan	0.1000	-0.2075
##					



##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	997.9016	-nan	0.1000	156.2244
##	2	874.7602	-nan	0.1000	136.0710
##	3	766.1950	-nan	0.1000	88.8837
##	4	673.5437	-nan	0.1000	86.0353
##	5	608.7514	-nan	0.1000	68.2482
##	6	548.6146	-nan	0.1000	57.5847
##	7	501.5420	-nan	0.1000	49.3042
##	8	456.5943	-nan	0.1000	40.2833
##	9	417.2413	-nan	0.1000	37.7478
##	10	387.6704	-nan	0.1000	27.5050
##	20	216.7646	-nan	0.1000	9.6328
##	40	132.6365	-nan	0.1000	1.6443
##	60	107.6826	-nan	0.1000	0.0679
##	80	94.8867	-nan	0.1000	0.1861
##	100	86.7963	-nan	0.1000	0.4310
##	120	80.0393	-nan	0.1000	0.0909
##	140	76.8072	-nan	0.1000	-0.1429
##	150	75.2394	-nan	0.1000	-0.0434
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1080.8189	-nan	0.1000	76.6571
##	2	1003.9710	-nan	0.1000	66.6505
##	3	942.4747	-nan	0.1000	58.5403
##	4	886.9356	-nan	0.1000	55.1046
##	5	839.6075	-nan	0.1000	47.9903
##	6	781.6543	-nan	0.1000	58.4151
##	7	735.4703	-nan	0.1000	39.6462
##	8	691.4888	-nan	0.1000	42.5628
##	9	654.6201	-nan	0.1000	36.2675
##	10	622.4645	-nan	0.1000	28.5020
##	20	401.3246	-nan	0.1000	8.4799
##	40	234.4538	-nan	0.1000	3.2475
##	60	175.0800	-nan	0.1000	0.7847
##	80	150.1060	-nan	0.1000	0.2590
##	100	137.6066	-nan	0.1000	-0.6271
##	120	128.8004	-nan	0.1000	0.3464
##	140	123.6679	-nan	0.1000	-0.3460
##	150	121.3646	-nan	0.1000	0.1333
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1032.4696	-nan	0.1000	155.1064
##	2	945.9092	-nan	0.1000	107.4908
##	3	833.1954	-nan	0.1000	98.7479
##	4	748.2650	-nan	0.1000	94.3671
##	5	680.7824	-nan	0.1000	70.5340
##	6	622.6394	-nan	0.1000	63.3733
##	7	568.8359	-nan	0.1000	61.2356
##	8	521.6860	-nan	0.1000	39.0593
##	9	482.9395	-nan	0.1000	44.5384
##	10	444.5591	-nan	0.1000	37.3012
##	20	257.2536	-nan	0.1000	9.2497
##	40	156.4679	-nan	0.1000	1.0572
##	60	124.0190	-nan	0.1000	0.7324

##	80	110.4133	-nan	0.1000	0.6957
##	100	101.9957	-nan	0.1000	0.7617
##	120	95.8166	-nan	0.1000	0.0013
##	140	93.3269	-nan	0.1000	-0.2322
##	150	91.7920	-nan	0.1000	-0.2663
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	996.8675	-nan	0.1000	155.6846
##	2	868.4262	-nan	0.1000	125.6263
##	3	766.1887	-nan	0.1000	104.5108
##	4	675.4711	-nan	0.1000	58.7944
##	5	600.3957	-nan	0.1000	68.9915
##	6	533.0195	-nan	0.1000	66.1509
##	7	477.9536	-nan	0.1000	37.5246
##	8	440.1525	-nan	0.1000	41.8343
##	9	402.6834	-nan	0.1000	37.8598
##	10	374.8409	-nan	0.1000	31.5991
##	20	210.2067	-nan	0.1000	8.9055
##	40	128.6086	-nan	0.1000	-1.2271
##	60	109.0801	-nan	0.1000	-0.2201
##	80	98.2993	-nan	0.1000	-0.1650
##	100	92.4266	-nan	0.1000	0.0777
##	120	86.3903	-nan	0.1000	-0.2130
##	140	82.5415	-nan	0.1000	-0.2469
##	150	81.0329	-nan	0.1000	-0.4306
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1222.9937	-nan	0.1000	95.5819
##	2	1145.0232	-nan	0.1000	77.1941
##	3	1077.9556	-nan	0.1000	67.2316
##	4	1009.5096	-nan	0.1000	67.9725
##	5	945.3913	-nan	0.1000	58.7099
##	6	894.1958	-nan	0.1000	54.7122
##	7	841.2722	-nan	0.1000	47.8715
##	8	798.8638	-nan	0.1000	30.1145
##	9	756.3627	-nan	0.1000	37.6571
##	10	714.7252	-nan	0.1000	42.2972
##	20	434.9873	-nan	0.1000	17.1626
##	40	248.3850	-nan	0.1000	4.5311
##	60	189.8807	-nan	0.1000	1.6512
##	80	163.0919	-nan	0.1000	0.6497
##	100	150.7166	-nan	0.1000	-0.4042
##	120	142.7196	-nan	0.1000	0.2574
##	140	136.4626	-nan	0.1000	-0.1399
##	150	133.4146	-nan	0.1000	-0.1580
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1160.9798	-nan	0.1000	137.7709
##	2	1025.0834	-nan	0.1000	130.3225
##	3	926.5422	-nan	0.1000	101.9054
##	4	835.0766	-nan	0.1000	95.1952
##	5	755.7430	-nan	0.1000	67.7841
##	6	696.4705	-nan	0.1000	59.9653
##	7	639.3003	-nan	0.1000	52.7958

##	8	578.5829	-nan	0.1000	60.5109
##	9	525.8460	-nan	0.1000	45.8725
##	10	483.9510	-nan	0.1000	44.4311
##	20	277.9536	-nan	0.1000	9.0038
##	40	169.2679	-nan	0.1000	0.9201
##	60	139.9209	-nan	0.1000	-0.2076
##	80	126.1292	-nan	0.1000	-0.7480
##	100	117.6557	-nan	0.1000	0.2180
##	120	110.9438	-nan	0.1000	-0.6432
##	140	106.7862	-nan	0.1000	0.0718
##	150	104.1988	-nan	0.1000	-0.0185

##

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1122.3496	-nan	0.1000	214.4010
##	2	977.3170	-nan	0.1000	141.5716
##	3	881.4311	-nan	0.1000	89.5670
##	4	802.2643	-nan	0.1000	73.8740
##	5	706.5898	-nan	0.1000	103.1031
##	6	638.2308	-nan	0.1000	74.1872
##	7	570.3551	-nan	0.1000	51.2075
##	8	521.9023	-nan	0.1000	47.8098
##	9	471.4959	-nan	0.1000	40.5367
##	10	430.4478	-nan	0.1000	38.1211
##	20	234.7109	-nan	0.1000	9.8324
##	40	139.1237	-nan	0.1000	1.2925
##	60	115.0282	-nan	0.1000	0.5032
##	80	103.9629	-nan	0.1000	-0.3077
##	100	96.6829	-nan	0.1000	-0.2997
##	120	91.8252	-nan	0.1000	0.0046
##	140	88.5128	-nan	0.1000	-0.2414
##	150	86.9650	-nan	0.1000	-0.2051

##

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1087.8570	-nan	0.1000	76.6095
##	2	1013.8603	-nan	0.1000	62.7495
##	3	957.5153	-nan	0.1000	55.9620
##	4	897.3468	-nan	0.1000	61.6687
##	5	836.9038	-nan	0.1000	49.7647
##	6	787.3263	-nan	0.1000	47.1981
##	7	744.5487	-nan	0.1000	36.7272
##	8	699.7224	-nan	0.1000	44.5974
##	9	664.5324	-nan	0.1000	34.5989
##	10	632.6207	-nan	0.1000	32.1634
##	20	404.4873	-nan	0.1000	14.4809
##	40	243.6616	-nan	0.1000	4.6382
##	60	187.3818	-nan	0.1000	1.8495
##	80	161.7907	-nan	0.1000	0.4776
##	100	149.1679	-nan	0.1000	0.4633
##	120	141.2548	-nan	0.1000	-0.4979
##	140	135.7785	-nan	0.1000	-0.1640
##	150	133.6173	-nan	0.1000	-0.5843

##

##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1050.9876	-nan	0.1000	112.8077

##	2	918.7068	-nan	0.1000	129.8786
##	3	808.9025	-nan	0.1000	98.3689
##	4	728.8537	-nan	0.1000	78.2674
##	5	676.8000	-nan	0.1000	55.3235
##	6	612.2399	-nan	0.1000	54.5006
##	7	559.6181	-nan	0.1000	50.6218
##	8	515.0434	-nan	0.1000	40.8312
##	9	473.0174	-nan	0.1000	36.9899
##	10	433.1366	-nan	0.1000	33.8391
##	20	272.3469	-nan	0.1000	9.4617
##	40	170.5693	-nan	0.1000	2.2894
##	60	143.1123	-nan	0.1000	0.3532
##	80	126.9598	-nan	0.1000	0.1493
##	100	118.7461	-nan	0.1000	-0.8097
##	120	113.7615	-nan	0.1000	-0.1565
##	140	109.6232	-nan	0.1000	-0.4357
##	150	107.6110	-nan	0.1000	0.4073
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1002.9726	-nan	0.1000	130.6249
##	2	874.5304	-nan	0.1000	106.8671
##	3	784.4297	-nan	0.1000	92.7181
##	4	697.1941	-nan	0.1000	88.3228
##	5	626.8668	-nan	0.1000	74.1835
##	6	573.3715	-nan	0.1000	54.5405
##	7	514.4491	-nan	0.1000	67.7620
##	8	464.2703	-nan	0.1000	44.0149
##	9	427.8894	-nan	0.1000	36.7926
##	10	396.0215	-nan	0.1000	30.5870
##	20	228.4352	-nan	0.1000	7.3448
##	40	143.6600	-nan	0.1000	0.7433
##	60	121.9373	-nan	0.1000	0.2120
##	80	111.5679	-nan	0.1000	-0.0402
##	100	103.4978	-nan	0.1000	0.1086
##	120	97.0031	-nan	0.1000	-0.1387
##	140	92.7187	-nan	0.1000	-0.0460
##	150	90.3598	-nan	0.1000	-0.0037
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1090.7159	-nan	0.1000	84.5760
##	2	1007.2289	-nan	0.1000	74.1523
##	3	946.8949	-nan	0.1000	60.3607
##	4	887.4432	-nan	0.1000	59.7682
##	5	819.2801	-nan	0.1000	70.1452
##	6	771.3496	-nan	0.1000	47.5261
##	7	720.1783	-nan	0.1000	53.3296
##	8	681.0785	-nan	0.1000	40.7998
##	9	643.3239	-nan	0.1000	30.1461
##	10	606.5380	-nan	0.1000	36.8915
##	20	377.9292	-nan	0.1000	12.3079
##	40	217.9400	-nan	0.1000	3.2746
##	60	166.1905	-nan	0.1000	1.2989
##	80	141.4818	-nan	0.1000	0.7027
##	100	128.4313	-nan	0.1000	0.1308

##	120	119.5264	-nan	0.1000	-0.1771
##	140	113.3529	-nan	0.1000	-0.0105
##	150	110.9493	-nan	0.1000	0.0478
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1033.9575	-nan	0.1000	132.8711
##	2	923.8711	-nan	0.1000	96.0836
##	3	819.2553	-nan	0.1000	97.9891
##	4	725.4915	-nan	0.1000	69.5400
##	5	664.0824	-nan	0.1000	62.4558
##	6	606.7370	-nan	0.1000	58.9446
##	7	551.1123	-nan	0.1000	55.5651
##	8	507.4330	-nan	0.1000	44.5145
##	9	467.8108	-nan	0.1000	36.8782
##	10	434.9003	-nan	0.1000	32.6813
##	20	248.0589	-nan	0.1000	12.5799
##	40	146.4711	-nan	0.1000	1.8295
##	60	114.5459	-nan	0.1000	0.7619
##	80	102.0801	-nan	0.1000	0.0312
##	100	94.9411	-nan	0.1000	0.5516
##	120	89.8323	-nan	0.1000	-0.5575
##	140	86.1568	-nan	0.1000	0.0435
##	150	84.8165	-nan	0.1000	-0.1532
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1026.8621	-nan	0.1000	142.8041
##	2	921.2654	-nan	0.1000	108.3263
##	3	795.7200	-nan	0.1000	106.8504
##	4	696.4259	-nan	0.1000	106.4723
##	5	617.6322	-nan	0.1000	72.1743
##	6	552.1506	-nan	0.1000	65.2443
##	7	494.2715	-nan	0.1000	50.5568
##	8	445.6477	-nan	0.1000	46.1631
##	9	408.5658	-nan	0.1000	39.8022
##	10	374.8992	-nan	0.1000	30.8978
##	20	208.5083	-nan	0.1000	6.9347
##	40	122.2220	-nan	0.1000	2.0757
##	60	99.9701	-nan	0.1000	0.5794
##	80	89.6386	-nan	0.1000	0.2813
##	100	83.1020	-nan	0.1000	0.1541
##	120	78.2733	-nan	0.1000	-0.2972
##	140	75.3554	-nan	0.1000	-0.1857
##	150	73.6533	-nan	0.1000	-0.1366
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1118.8115	-nan	0.1000	73.4980
##	2	1050.4907	-nan	0.1000	66.5460
##	3	978.8147	-nan	0.1000	67.2210
##	4	917.3993	-nan	0.1000	54.8446
##	5	857.4446	-nan	0.1000	58.4640
##	6	810.0807	-nan	0.1000	42.8145
##	7	764.8915	-nan	0.1000	45.2697
##	8	717.6212	-nan	0.1000	48.6299
##	9	678.6333	-nan	0.1000	38.5026

##	10	642.8427	-nan	0.1000	36.2583
##	20	400.0696	-nan	0.1000	15.5698
##	40	232.2462	-nan	0.1000	3.8881
##	60	174.7613	-nan	0.1000	1.5945
##	80	148.1350	-nan	0.1000	0.5106
##	100	135.2250	-nan	0.1000	-0.9972
##	120	126.5791	-nan	0.1000	0.3248
##	140	119.9147	-nan	0.1000	0.1349
##	150	117.7438	-nan	0.1000	0.0888
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1062.1767	-nan	0.1000	153.5348
##	2	936.4652	-nan	0.1000	115.1786
##	3	829.3559	-nan	0.1000	85.4330
##	4	757.2288	-nan	0.1000	79.2473
##	5	700.1942	-nan	0.1000	61.7006
##	6	636.2802	-nan	0.1000	66.7213
##	7	580.7349	-nan	0.1000	56.5523
##	8	534.5268	-nan	0.1000	46.2899
##	9	498.0534	-nan	0.1000	35.5052
##	10	458.5808	-nan	0.1000	36.8635
##	20	272.1561	-nan	0.1000	10.5912
##	40	163.0717	-nan	0.1000	1.0363
##	60	126.9545	-nan	0.1000	1.4551
##	80	113.7320	-nan	0.1000	-0.0505
##	100	106.1635	-nan	0.1000	-0.7638
##	120	100.3572	-nan	0.1000	0.5649
##	140	95.7085	-nan	0.1000	-0.1186
##	150	93.0152	-nan	0.1000	0.3598
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1041.6123	-nan	0.1000	154.3690
##	2	914.0338	-nan	0.1000	142.5481
##	3	801.1364	-nan	0.1000	110.1066
##	4	733.4377	-nan	0.1000	74.4354
##	5	652.7770	-nan	0.1000	91.6788
##	6	578.6338	-nan	0.1000	59.0238
##	7	526.6763	-nan	0.1000	49.7396
##	8	479.9661	-nan	0.1000	42.8199
##	9	438.8906	-nan	0.1000	41.4579
##	10	404.4224	-nan	0.1000	35.9367
##	20	229.6770	-nan	0.1000	9.7288
##	40	135.5758	-nan	0.1000	0.8887
##	60	109.8398	-nan	0.1000	-0.2751
##	80	97.5181	-nan	0.1000	-0.1177
##	100	91.6485	-nan	0.1000	-0.3932
##	120	85.1088	-nan	0.1000	-0.1882
##	140	80.8995	-nan	0.1000	-0.4833
##	150	78.4903	-nan	0.1000	-0.2523
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1119.8773	-nan	0.1000	94.3423
##	2	1036.5403	-nan	0.1000	84.1755
##	3	967.3030	-nan	0.1000	68.0679

##	4	900.4328	-nan	0.1000	51.4887
##	5	840.7201	-nan	0.1000	57.2646
##	6	789.3604	-nan	0.1000	53.1029
##	7	737.3305	-nan	0.1000	48.8495
##	8	694.6946	-nan	0.1000	46.8030
##	9	651.7255	-nan	0.1000	37.3301
##	10	615.7655	-nan	0.1000	37.5008
##	20	383.5992	-nan	0.1000	11.8737
##	40	228.2604	-nan	0.1000	2.4331
##	60	178.3682	-nan	0.1000	1.1799
##	80	152.9885	-nan	0.1000	0.4506
##	100	139.4953	-nan	0.1000	0.1046
##	120	131.4240	-nan	0.1000	0.2497
##	140	125.9992	-nan	0.1000	-0.6988
##	150	123.9954	-nan	0.1000	0.1049
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1056.6943	-nan	0.1000	152.1538
##	2	922.9485	-nan	0.1000	127.0554
##	3	812.9224	-nan	0.1000	112.0540
##	4	735.7370	-nan	0.1000	77.1514
##	5	660.4394	-nan	0.1000	77.7973
##	6	594.5295	-nan	0.1000	54.8795
##	7	538.1276	-nan	0.1000	50.0000
##	8	496.2987	-nan	0.1000	41.3740
##	9	463.3502	-nan	0.1000	29.0453
##	10	423.1220	-nan	0.1000	32.9575
##	20	254.4723	-nan	0.1000	10.8771
##	40	157.7539	-nan	0.1000	3.1032
##	60	129.3769	-nan	0.1000	0.6312
##	80	115.9494	-nan	0.1000	0.5027
##	100	109.1067	-nan	0.1000	-0.1916
##	120	103.6767	-nan	0.1000	-0.1559
##	140	100.4952	-nan	0.1000	-0.0890
##	150	98.7492	-nan	0.1000	-0.2977
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1048.6867	-nan	0.1000	169.7189
##	2	898.1875	-nan	0.1000	132.8233
##	3	779.0176	-nan	0.1000	115.3169
##	4	688.4945	-nan	0.1000	76.3549
##	5	601.5160	-nan	0.1000	76.0706
##	6	535.5268	-nan	0.1000	49.9857
##	7	491.4052	-nan	0.1000	44.9608
##	8	456.8617	-nan	0.1000	33.7188
##	9	427.1252	-nan	0.1000	27.1048
##	10	396.9068	-nan	0.1000	34.3282
##	20	227.6345	-nan	0.1000	7.7327
##	40	139.1342	-nan	0.1000	2.4652
##	60	116.2420	-nan	0.1000	0.1765
##	80	104.1271	-nan	0.1000	-0.5082
##	100	96.7440	-nan	0.1000	-0.1467
##	120	92.1058	-nan	0.1000	-0.1483
##	140	87.7991	-nan	0.1000	0.0628

```

##      150      86.2169      -nan      0.1000     -0.0998
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1189.0101      -nan      0.1000     86.6900
##      2      1114.6414      -nan      0.1000     71.3947
##      3      1047.4181      -nan      0.1000     57.8647
##      4       977.1987      -nan      0.1000     66.0326
##      5       907.3896      -nan      0.1000     67.8968
##      6       851.5658      -nan      0.1000     54.0091
##      7       800.4883      -nan      0.1000     43.4781
##      8       753.3136      -nan      0.1000     40.4555
##      9       709.8473      -nan      0.1000     39.5049
##     10       668.1243      -nan      0.1000     37.6030
##     20       423.7775      -nan      0.1000     18.7096
##     40       248.4562      -nan      0.1000      3.1495
##     60       192.5803      -nan      0.1000      1.5092
##     80       169.4225      -nan      0.1000      0.4061
##    100       156.9656      -nan      0.1000      0.2653
##    120       149.3569      -nan      0.1000      0.0663
##    140       145.4897      -nan      0.1000     -0.1711
##    150       143.3884      -nan      0.1000     -0.4252
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1129.1701      -nan      0.1000    153.7598
##      2       996.7433      -nan      0.1000     97.6723
##      3       909.4208      -nan      0.1000     88.9440
##      4       821.4093      -nan      0.1000     82.8959
##      5       755.1422      -nan      0.1000     67.1834
##      6       678.6369      -nan      0.1000     67.5101
##      7       618.9197      -nan      0.1000     56.0494
##      8       569.8699      -nan      0.1000     52.9390
##      9       533.0902      -nan      0.1000     36.0775
##     10       490.8521      -nan      0.1000     41.1104
##     20       279.8185      -nan      0.1000      8.8339
##     40       181.4204      -nan      0.1000      1.5549
##     60       150.0737      -nan      0.1000      1.6910
##     80       137.2608      -nan      0.1000      0.7824
##    100       129.9877      -nan      0.1000     -0.4570
##    120       123.8775      -nan      0.1000     -0.3129
##    140       118.0144      -nan      0.1000     -0.7412
##    150       115.0476      -nan      0.1000     -0.3169
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1106.8309      -nan      0.1000    137.4255
##      2       983.8373      -nan      0.1000    111.8664
##      3       876.5559      -nan      0.1000    110.1780
##      4       766.9048      -nan      0.1000    109.8097
##      5       682.1685      -nan      0.1000     80.4737
##      6       622.5092      -nan      0.1000     66.1048
##      7       550.4590      -nan      0.1000     62.3545
##      8       498.6682      -nan      0.1000     46.0724
##      9       457.5230      -nan      0.1000     46.5795
##     10       421.0962      -nan      0.1000     37.5097
##     20       236.7550      -nan      0.1000      9.0588

```



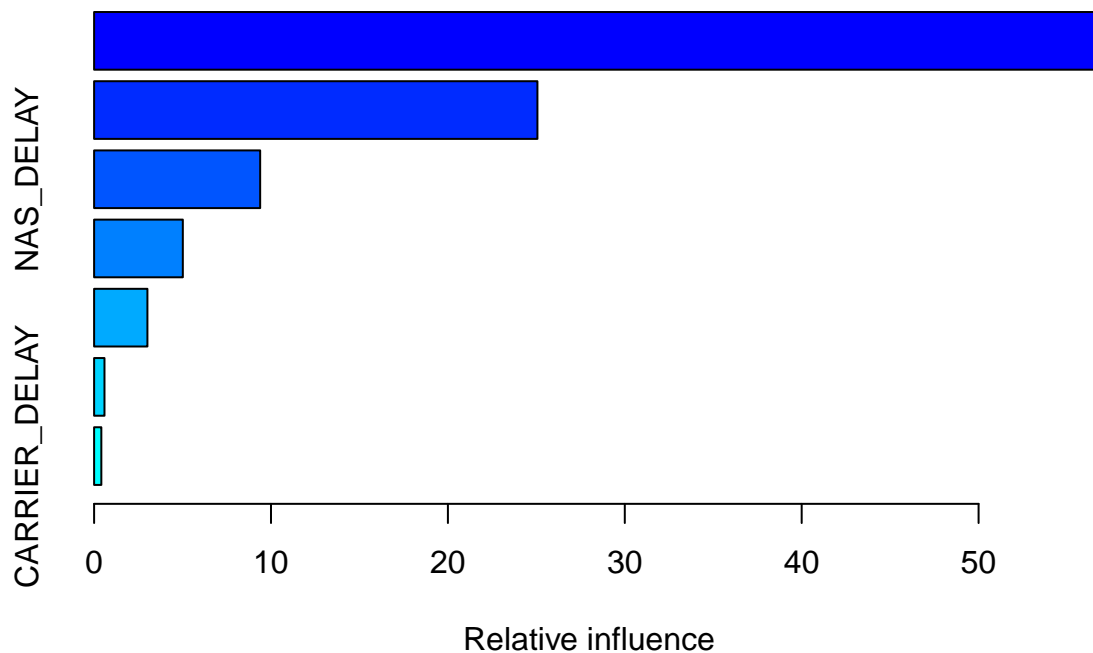
##	40	151.0315	-nan	0.1000	0.7579
##	60	127.0778	-nan	0.1000	-0.1092
##	80	116.4214	-nan	0.1000	-0.0738
##	100	107.3310	-nan	0.1000	-0.5775
##	120	102.4719	-nan	0.1000	-0.3353
##	140	96.2524	-nan	0.1000	-0.0157
##	150	94.1987	-nan	0.1000	0.1255
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1283.6954	-nan	0.1000	73.1715
##	2	1182.6265	-nan	0.1000	84.5721
##	3	1092.3751	-nan	0.1000	87.0078
##	4	1020.7277	-nan	0.1000	66.1521
##	5	954.5623	-nan	0.1000	65.1395
##	6	892.8784	-nan	0.1000	38.6726
##	7	839.0687	-nan	0.1000	52.9767
##	8	785.7557	-nan	0.1000	57.1875
##	9	743.5452	-nan	0.1000	39.0069
##	10	695.9698	-nan	0.1000	48.5670
##	20	422.0979	-nan	0.1000	15.6932
##	40	243.9402	-nan	0.1000	4.0555
##	60	186.8147	-nan	0.1000	0.8337
##	80	161.0412	-nan	0.1000	0.2119
##	100	147.5159	-nan	0.1000	0.2091
##	120	138.3987	-nan	0.1000	0.0832
##	140	131.6827	-nan	0.1000	0.1823
##	150	128.7122	-nan	0.1000	-0.0164
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1207.7910	-nan	0.1000	180.1243
##	2	1063.2926	-nan	0.1000	153.4980
##	3	965.0687	-nan	0.1000	85.7630
##	4	864.6020	-nan	0.1000	91.2689
##	5	773.1352	-nan	0.1000	82.4426
##	6	700.6816	-nan	0.1000	69.2203
##	7	631.2889	-nan	0.1000	68.3590
##	8	573.0636	-nan	0.1000	51.0187
##	9	528.0273	-nan	0.1000	45.9456
##	10	476.1582	-nan	0.1000	44.3333
##	20	278.1279	-nan	0.1000	9.1232
##	40	169.9533	-nan	0.1000	1.3212
##	60	137.5625	-nan	0.1000	0.8528
##	80	123.0038	-nan	0.1000	-0.2596
##	100	112.5488	-nan	0.1000	0.2349
##	120	105.2482	-nan	0.1000	-0.0700
##	140	101.6530	-nan	0.1000	-0.3782
##	150	100.3467	-nan	0.1000	-0.7668
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1187.7043	-nan	0.1000	214.3587
##	2	1020.9794	-nan	0.1000	181.5930
##	3	893.9061	-nan	0.1000	127.2004
##	4	778.8480	-nan	0.1000	120.6036
##	5	695.0268	-nan	0.1000	74.5797

##	6	629.3644	-nan	0.1000	77.4514
##	7	561.2605	-nan	0.1000	67.4979
##	8	510.4674	-nan	0.1000	53.8901
##	9	459.3352	-nan	0.1000	37.7979
##	10	416.3479	-nan	0.1000	41.0710
##	20	227.3097	-nan	0.1000	6.8951
##	40	136.2936	-nan	0.1000	1.5643
##	60	115.3820	-nan	0.1000	0.0036
##	80	103.7335	-nan	0.1000	-0.0722
##	100	96.6125	-nan	0.1000	0.0211
##	120	92.4467	-nan	0.1000	-0.1759
##	140	88.4003	-nan	0.1000	-0.2170
##	150	86.1169	-nan	0.1000	-0.0322
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1210.0056	-nan	0.1000	77.0098
##	2	1131.9875	-nan	0.1000	77.4601
##	3	1050.1754	-nan	0.1000	78.9251
##	4	985.7537	-nan	0.1000	63.0267
##	5	929.2235	-nan	0.1000	51.8651
##	6	868.4312	-nan	0.1000	66.4035
##	7	813.4592	-nan	0.1000	52.3478
##	8	772.3243	-nan	0.1000	43.1009
##	9	738.1129	-nan	0.1000	33.0682
##	10	699.9654	-nan	0.1000	40.1154
##	20	434.8572	-nan	0.1000	14.7241
##	40	252.9659	-nan	0.1000	4.0820
##	60	190.3498	-nan	0.1000	2.0213
##	80	162.8234	-nan	0.1000	0.6628
##	100	150.7828	-nan	0.1000	0.5502
##	120	142.9481	-nan	0.1000	0.0537
##	140	137.5727	-nan	0.1000	-0.5494
##	150	135.5533	-nan	0.1000	0.0043
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1144.5054	-nan	0.1000	171.3193
##	2	1026.1129	-nan	0.1000	122.7750
##	3	929.5790	-nan	0.1000	92.9228
##	4	835.4790	-nan	0.1000	75.4115
##	5	752.1271	-nan	0.1000	78.2724
##	6	683.9247	-nan	0.1000	71.9112
##	7	620.0264	-nan	0.1000	54.5971
##	8	565.8734	-nan	0.1000	40.3096
##	9	516.9392	-nan	0.1000	38.0251
##	10	484.2238	-nan	0.1000	36.4699
##	20	292.2535	-nan	0.1000	10.8410
##	40	165.6017	-nan	0.1000	2.9954
##	60	134.2776	-nan	0.1000	0.5575
##	80	120.1004	-nan	0.1000	0.0713
##	100	111.6263	-nan	0.1000	-1.0180
##	120	105.9832	-nan	0.1000	-0.2969
##	140	102.5158	-nan	0.1000	0.0533
##	150	100.7794	-nan	0.1000	-0.2743
##					

```
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1      1163.8655          -nan      0.1000    156.2903
##      2      1016.5769          -nan      0.1000    143.8181
##      3       882.9052          -nan      0.1000    133.6532
##      4       772.9999          -nan      0.1000     83.3790
##      5       688.3052          -nan      0.1000     60.3592
##      6       613.6813          -nan      0.1000     68.7803
##      7       550.7211          -nan      0.1000     64.4275
##      8       497.8653          -nan      0.1000     47.6486
##      9       458.5626          -nan      0.1000     40.8449
##     10       420.3436          -nan      0.1000     30.5602
##     20       229.2189          -nan      0.1000     15.5717
##     40       134.3119          -nan      0.1000      0.8417
##     60       112.1139          -nan      0.1000      1.0587
##     80        99.2406          -nan      0.1000      0.1316
##    100        92.5674          -nan      0.1000     -0.1656
##    120        87.5723          -nan      0.1000      0.1431
##    140        83.9197          -nan      0.1000     -0.3731
##    150        81.6334          -nan      0.1000      0.2738
##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1      1066.3687          -nan      0.1000    169.8292
##      2       927.1047          -nan      0.1000    114.6841
##      3       807.6555          -nan      0.1000    121.2462
##      4       718.3953          -nan      0.1000     99.4117
##      5       634.5350          -nan      0.1000     79.5987
##      6       573.8356          -nan      0.1000     69.3062
##      7       515.3659          -nan      0.1000     47.3729
##      8       470.2866          -nan      0.1000     36.9176
##      9       429.8505          -nan      0.1000     39.8279
##     10       400.3352          -nan      0.1000     30.4731
##     20       229.6228          -nan      0.1000      8.5911
##     40       144.8718          -nan      0.1000      1.4341
##     60       123.6587          -nan      0.1000      1.1849
##     80       111.3196          -nan      0.1000      0.1940
##    100       105.9478          -nan      0.1000     -0.0948
##    120       101.1540          -nan      0.1000     -0.2309
##    140        97.2260          -nan      0.1000     -0.1688
##    150        95.4646          -nan      0.1000     -0.2985
```

The `summary()` function also provides a relative influence plot and also outputs the relative influence statistics.

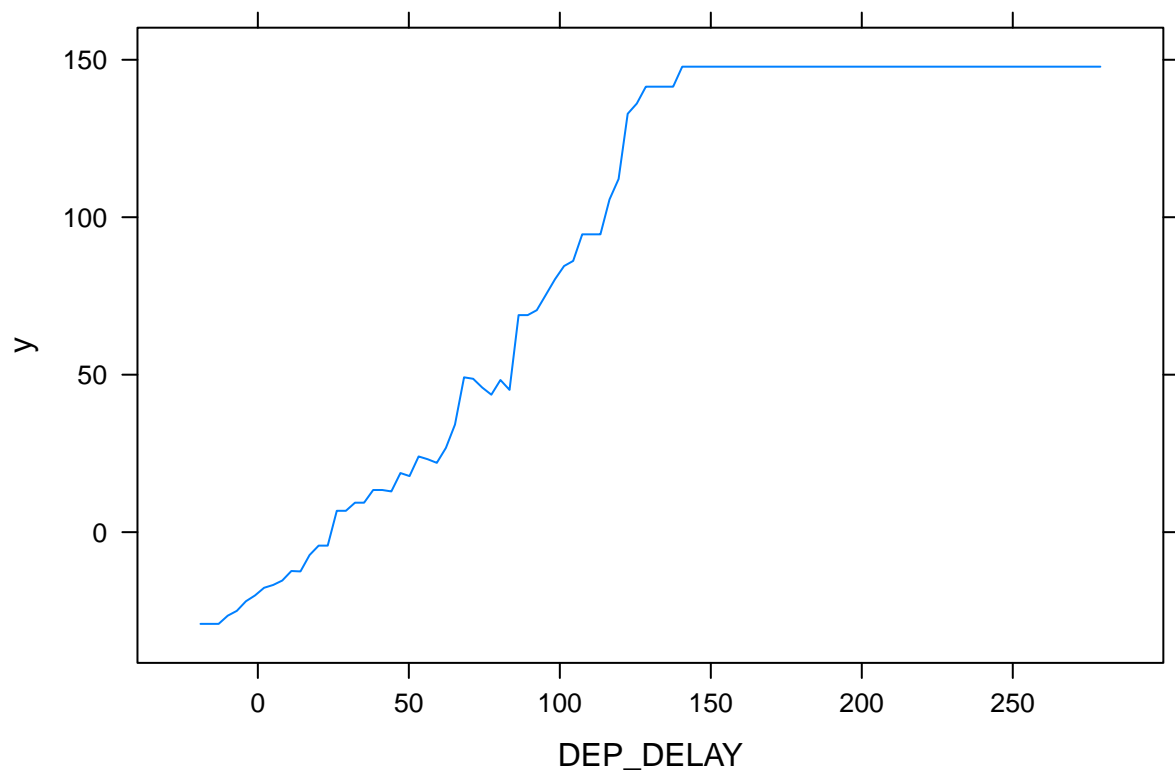
```
summary(boost.delay)
```



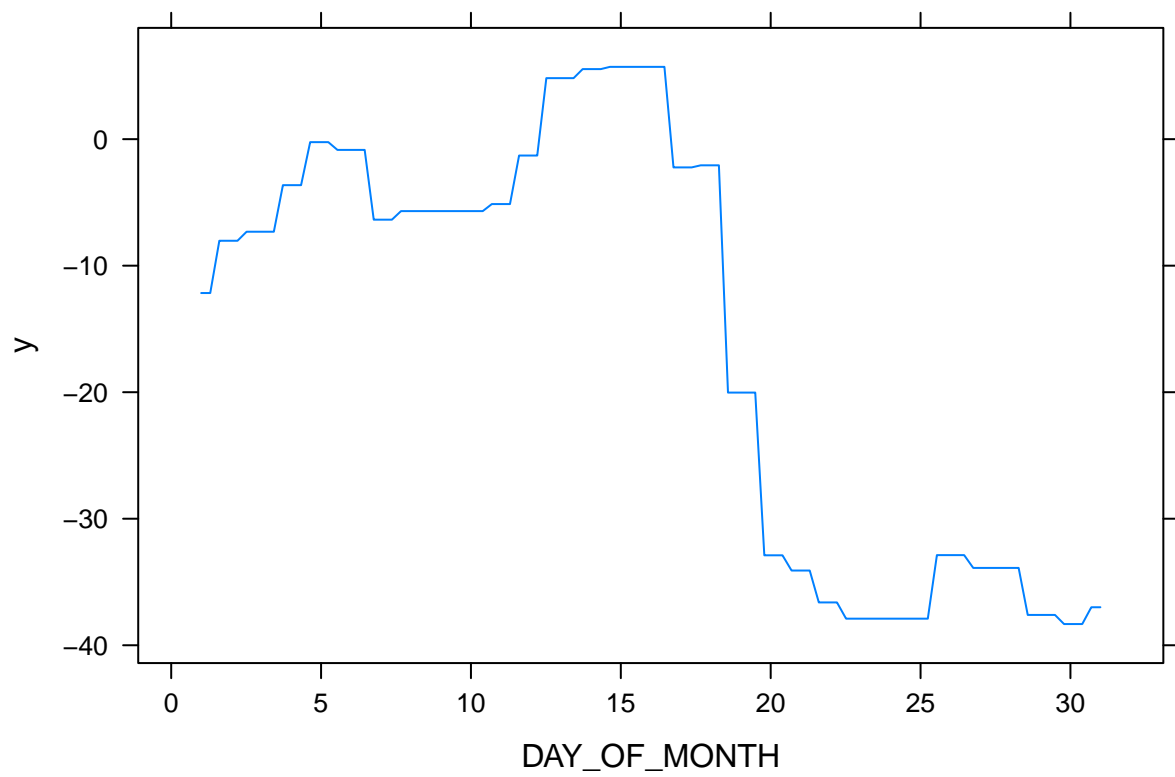
```
##           var    rel.inf
## DEP_DELAY      DEP_DELAY 56.5279213
## DAY_OF_MONTH    DAY_OF_MONTH 25.0608132
## NAS_DELAY       NAS_DELAY  9.3867001
## TAXI_OUT        TAXI_OUT  5.0157480
## TAXI_IN         TAXI_IN   3.0135950
## LATE_AIRCRAFT_DELAY LATE_AIRCRAFT_DELAY 0.5862760
## CARRIER_DELAY   CARRIER_DELAY 0.4089464
```

We see that `DEP_DELAY` and `DAY_OF_MONTH` are by far the most important variables. We can also produce *partial dependence plots* for these two variables. These plots illustrate the marginal effect of the selected variables on the response after *integrating* out the other variables.

```
par(mfrow = c(1,2))
plot(boost.delay, i = "DEP_DELAY")
```



```
plot(boost.delay, i = "DAY_OF_MONTH")
```



## Trees Test Error Metrics & Predictions

```
# predictions
## random forest
yhat.rf <- predict(rf.delay, newdata = test)
## boosting
yhat.boost <- predict(boost.delay, newdata = test,
                      n.trees = 150)

## histogram of predictions vs. actual
# actual ARR_DELAY in test set
ptest_actual <- ggplot(data = test, aes(x = ARR_DELAY)) +
  geom_histogram(fill = "#FFFF00", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Actual") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- baseline lm
ptest_rf <- ggplot(data = test, aes(x = yhat.rf)) +
  geom_histogram(fill = "#FFFF00", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Random Forest") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

# predicted ARR_DELAY in test set -- lm with log-trans
ptest_boost <- ggplot(data = test, aes(x = yhat.boost)) +
  geom_histogram(fill = "#FFFF00", color = "#002D72", alpha = 0.8) +
  labs(x = "Arrival Delay",
       y = "Frequency",
       title = "Boosting") +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5),
        axis.title.x.bottom = element_text(size = 8, face = "italic"),
        axis.title.y.left = element_text(size = 8))

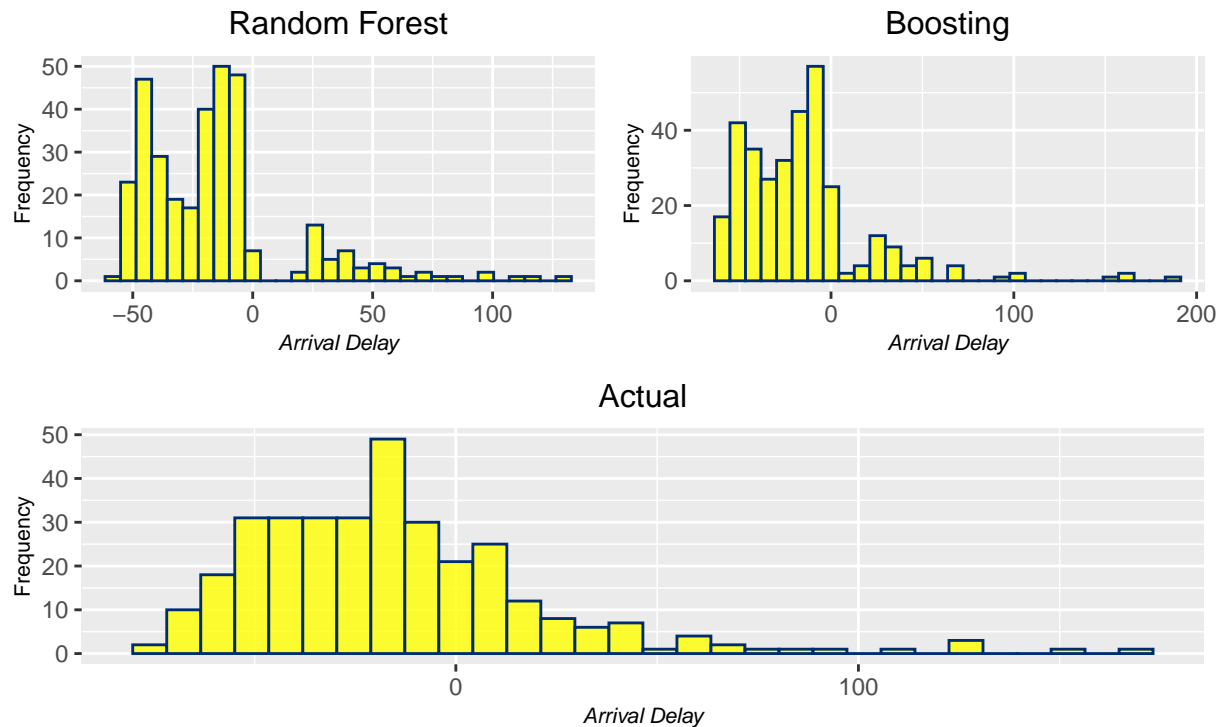
# grid.arrange(ptest_actual, ptest_rf, ptest_boost, ncol = 2,
#               width = c(2,1,1),
#               layout_matrix = rbind(c(1,2),
#                                     c(1, 3)))

trees_patchwork <- (ptest_rf | ptest_boost) / ptest_actual
trees_patchwork + plot_annotation(
  title = 'Comparing Distributions of ARR_DELAY',
  subtitle = 'Histograms of actual test values and tree-based predicted values'
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

## Comparing Distributions of ARR\_DELAY

Histograms of actual test values and tree-based predicted values



```
# test MSE calculations
```

```
rf.MSE <- sum((test$ARR_DELAY - yhat.rf)^2, na.rm=T)/length(test$ARR_DELAY)
rf.MSE
```

```
## [1] 155.0148
```

```
boost_MSE <- sum((test$ARR_DELAY-yhat.boost)^2, na.rm = T)/length(test$ARR_DELAY)
boost_MSE
```

```
## [1] 129.7965
```

## Test Error Tables

```
customGreen = "#71CA97"
customRed = "#ff7f7f"

library(formattable)

##
## Attaching package: 'formattable'
##
## The following object is masked from 'package:bst':
##
##     gradient
##
## The following object is masked from 'package:MASS':
##
##     area
##
## The following object is masked from 'package:patchwork':
##
##     area

options(scipen = 5, digits = 4)
model.names <- c("Baseline Linear", "Selected Linear w/ Log-Transformed Predictors", "Selected Linear w/
model.types <- c("Multiple Linear Regression", "Multiple Linear Regression", "Multiple Linear Regression")

#model4.ints <- c("FALSE", "FALSE", "FALSE", "FALSE")

model.mse <- c(plain_linear_model_MSE, log_linear_MSE,
               bc_adj_linear_model_MSE, gam_MSE,
               gambc_MSE, rf.MSE, boost_MSE)
model.mse.char <- c("322.46", "333.90", "334.92", "312.30", "317.45", "155.01", "129.80")

#model4.cumse_var <- c(mlr4_1_cv, mlr4_4_bc_cv, ridge.mom4.cumse, gam4_bc_gcv)
#model4.cumse <- c(2.284e+20, "25.66", 25.62, 25.79)

pctchange_1 <- round(-((log_linear_MSE - plain_linear_model_MSE)/plain_linear_model_MSE)*100, digits = 4)
pctchange_2 <- round(-((bc_adj_linear_model_MSE - plain_linear_model_MSE)/plain_linear_model_MSE)*100, digits = 4)
pctchange_3 <- round(-((gam_MSE - plain_linear_model_MSE)/plain_linear_model_MSE)*100, digits = 4)
pctchange_4 <- round(-((gambc_MSE - plain_linear_model_MSE)/plain_linear_model_MSE)*100, digits = 4)
pctchange_5 <- round(-((rf.MSE - plain_linear_model_MSE)/plain_linear_model_MSE)*100, digits = 4)
pctchange_6 <- round(-((boost_MSE - plain_linear_model_MSE)/plain_linear_model_MSE)*100, digits = 4)

model.pctchange <- c("---", pctchange_1, pctchange_2, pctchange_3, pctchange_4, pctchange_5, pctchange_6)

errors.df <- data.frame(model.names,
                        model.types,
                        model.mse.char,
                        model.pctchange
                        )

#model4.ints,
#model4.cumse,
#model4.pctchange
```



```
#colnames(errors.df4) <- c("Model Name", "Model Type", "Interactions?", "Model MSE", "Model CV MSE", "Model AIC")
errors.df
```

```
##               model.names               model.types
## 1               Baseline Linear Multiple Linear Regression
## 2 Selected Linear w/ Log-Transformed Predictors Multiple Linear Regression
## 3               Selected Linear w/ Box-Cox Multiple Linear Regression
## 4                               GAM Generalized Additive Model
## 5               GAM w/ Box-Cox Generalized Additive Model
## 6               Random Forest           Tree-Based Regression
## 7               Boosting           Tree-Based Regression
##  model.mse.char model.pctchange
## 1           322.46           ---
## 2           333.90          -3.5469
## 3           334.92          -3.8652
## 4           312.30           3.1519
## 5           317.45           1.5521
## 6           155.01          51.9272
## 7           129.80          59.7479
```

```
mlr.model.names <- c("Baseline Linear", "Selected Linear w/ Log-Transformed Predictors", "Selected Linear w/ Box-Cox")
mlr.mse.char <- c("322.46", "333.90", "334.92")
```

```
mlr.error.df <- data.frame(mlr.model.names, mlr.mse.char)
```

```
formattable(mlr.error.df,
  col.names = c("Model Name", "Model MSE"),
  list(
    mlr.model.names = formatter("span", style = x ~ ifelse(x == "Baseline Linear",
      style(color = "purple", font.weight = "bold"), NA)),
    mlr.mse.char = formatter("span", style = x ~ ifelse(x == "322.46",
      style(color = "purple", font.weight = "bold"), NA))
  ))
```

Model Name

Model MSE

Baseline Linear

322.46

Selected Linear w/ Log-Transformed Predictors

333.90

Selected Linear w/ Box-Cox

334.92

```
gam.model.names <- c("GAM", "GAM w/ Box-Cox")
gam.mse.char <- c("312.30", "317.45")
```

```
gam.error.df <- data.frame(gam.model.names, gam.mse.char)
```

```
formattable(gam.error.df,
  col.names = c("Model Name", "Model MSE"),
```

```

      list(
gam.model.names = formatter("span", style = x ~ ifelse(x == "GAM",
  style(color = "purple", font.weight = "bold"), NA)),
gam.mse.char = formatter("span", style = x ~ ifelse(x == "312.30",
  style(color = "purple", font.weight = "bold"), NA))
))

```

Model Name

Model MSE

GAM

312.30

GAM w/ Box-Cox

317.45

```

tree.model.names <- c("Random Forest", "Boosting")
tree.mse.char <- c("155.01", "129.80")

tree.error.df <- data.frame(tree.model.names, tree.mse.char)

formattable(tree.error.df,
  col.names = c("Model Name", "Model MSE"),
  list(
    tree.model.names = formatter("span", style = x ~ ifelse(x == "Boosting",
      style(color = "purple", font.weight = "bold"), NA)),
    tree.mse.char = formatter("span", style = x ~ ifelse(x == "129.80",
      style(color = "purple", font.weight = "bold"), NA))
  ))

```

Model Name

Model MSE

Random Forest

155.01

Boosting

129.80

```

formattable(errors.df,
  col.names = c("Model Name", "Model Type", "Model MSE", "Model Percent Improvement"),
  list(
    model.names = formatter("span", style = x ~ ifelse(x == "Boosting",
      style(color = "purple", font.weight = "bold"), NA)),
    model.mse.char = formatter("span", style = x ~ ifelse(x == "129.80",
      style(color = "purple", font.weight = "bold"), NA)),
    model.types = formatter("span", style = x ~ ifelse(x == "Tree-Based Regression ",
      style(color = "purple", font.weight = "bold"), NA)),
    model.pctchange = formatter("span",
      style = x ~ style(font.weight = "bold",
        color = ifelse(x == "---", "black",
          ifelse(x > 0, customGreen, ifelse(x < 0,
            x ~ icontext(ifelse(x>0, "arrow-up", "arrow-down"), x)

```

Model Name	Model Type	Model MSE	Model Percent Improvement
Baseline Linear	Multiple Linear Regression	322.46	
Selected Linear w/ Log-Transformed Predictors	Multiple Linear Regression	333.90	-3.5469
Selected Linear w/ Box-Cox	Multiple Linear Regression	334.92	-3.8652
GAM	Generalized Additive Model	312.30	3.1519
GAM w/ Box-Cox	Generalized Additive Model	317.45	1.5521
Random Forest	Tree-Based Regression	155.01	51.9272
Boosting	Tree-Based Regression	129.80	59.7479