APPENDIX

* **Software Details**: R-version- 3.6.1
* **Names of packages used:**

library(tidyverse)

library(lubridate)

library(randomForest)

library(gam)

library(tree)

library(gbm)

library(xgboost)

* **Time to knit : 3-4 minutes**

STAT 652 Project

Ria Gupta

06/12/2019

## Reading flight data

The data are on flights from three New York City airports in 2013, from the nycflights13 package. Data were combined from four datasets from this package:

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(nycflights13)  
#help(flights)  
#help(weather)  
#help(airports)  
#help(planes)  
fltrain <- read.csv("fltrain.csv.gz")

dim(fltrain)

## [1] 200000 43

There are 43 variables measured on 200,000 flights.

##Converting character to factor variables

fl <- fltrain  
for(i in 1:ncol(fl)) {  
 if(typeof(fl[[i]]) == "character") {  
 fl[[i]] <- factor(fl[[i]])  
 }  
}

## Missing data

### Calculating percentage of missing values for each column

p<-function(x) {sum(is.na(x))/length(x)\*100}  
apply(fl,2,p)

## year.x month day dep\_time sched\_dep\_time   
## 0.0000 0.0000 0.0000 2.4490 0.0000   
## dep\_delay arr\_time sched\_arr\_time arr\_delay carrier   
## 2.4490 2.5845 0.0000 2.7920 0.0000   
## flight tailnum origin dest air\_time   
## 0.0000 0.7460 0.0000 0.0000 2.7920   
## distance hour minute time\_hour temp   
## 0.0000 0.0000 0.0000 0.0000 0.4740   
## dewp humid wind\_dir wind\_speed wind\_gust   
## 0.4740 0.4740 2.9310 0.4910 76.1300   
## precip pressure visib name lat   
## 0.4685 11.5460 0.4685 2.2420 2.2420   
## lon alt tz dst tzone   
## 2.2420 2.2420 2.2420 2.2420 2.2420   
## year.y type manufacturer model engines   
## 17.1490 15.5815 15.5815 15.5815 15.5815   
## seats speed engine   
## 15.5815 99.7075 15.5815

### Action:

From above missing values percentage we observe, that predictors such as wind\_gust, pressure, type, manufacturer, model,engines,seats, engine, speed have more than 5% values missing. From thumb of rule, we remove these variables.

fl <- fl%>%   
 select(-type,-manufacturer,-model,-engines,-seats, -speed, -engine,-wind\_gust,-pressure,-year.y)

###to omit the rows with missing data

fl <- na.omit(fl)  
summary(fl)

## year.x month day dep\_time   
## Min. :2013 Min. : 1.000 Min. : 1.00 Min. : 1   
## 1st Qu.:2013 1st Qu.: 4.000 1st Qu.: 8.00 1st Qu.: 910   
## Median :2013 Median : 7.000 Median :16.00 Median :1408   
## Mean :2013 Mean : 6.553 Mean :15.67 Mean :1353   
## 3rd Qu.:2013 3rd Qu.:10.000 3rd Qu.:23.00 3rd Qu.:1747   
## Max. :2013 Max. :12.000 Max. :31.00 Max. :2400   
##   
## sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
## Min. : 500 Min. : -43.00 Min. : 1 Min. : 1   
## 1st Qu.: 905 1st Qu.: -5.00 1st Qu.:1106 1st Qu.:1123   
## Median :1359 Median : -2.00 Median :1545 Median :1602   
## Mean :1342 Mean : 12.67 Mean :1511 Mean :1544   
## 3rd Qu.:1729 3rd Qu.: 11.00 3rd Qu.:1946 3rd Qu.:1950   
## Max. :2345 Max. :1301.00 Max. :2400 Max. :2359   
##   
## arr\_delay carrier flight tailnum   
## Min. : -79.000 UA :32252 Min. : 1 N725MQ : 322   
## 1st Qu.: -17.000 B6 :29282 1st Qu.: 544 N723MQ : 271   
## Median : -5.000 EV :29137 Median :1499 N711MQ : 268   
## Mean : 7.014 DL :26998 Mean :1966 N722MQ : 268   
## 3rd Qu.: 14.000 AA :17742 3rd Qu.:3448 N351JB : 247   
## Max. :1272.000 MQ :14382 Max. :8500 N258JB : 244   
## (Other):34523 (Other):182696   
## origin dest air\_time distance   
## EWR:65512 ATL : 9726 Min. : 20.0 Min. : 80   
## JFK:60327 ORD : 9443 1st Qu.: 81.0 1st Qu.: 502   
## LGA:58477 LAX : 9185 Median :127.0 Median : 866   
## BOS : 8674 Mean :149.5 Mean :1035   
## MCO : 8131 3rd Qu.:184.0 3rd Qu.:1372   
## CLT : 7822 Max. :695.0 Max. :4983   
## (Other):131335   
## hour minute time\_hour   
## Min. : 5.00 Min. : 0.00 2013-09-17T12:00:00Z: 61   
## 1st Qu.: 9.00 1st Qu.: 8.00 2013-08-05T10:00:00Z: 58   
## Median :13.00 Median :29.00 2013-10-04T12:00:00Z: 58   
## Mean :13.15 Mean :26.06 2013-10-23T12:00:00Z: 58   
## 3rd Qu.:17.00 3rd Qu.:43.00 2013-09-19T12:00:00Z: 57   
## Max. :23.00 Max. :59.00 2013-10-25T12:00:00Z: 57   
## (Other) :183967   
## temp dewp humid wind\_dir   
## Min. : 10.94 Min. :-9.94 Min. : 13.00 Min. : 0.0   
## 1st Qu.: 42.08 1st Qu.:26.06 1st Qu.: 43.71 1st Qu.:130.0   
## Median : 57.02 Median :42.08 Median : 57.14 Median :220.0   
## Mean : 56.86 Mean :41.33 Mean : 59.12 Mean :201.9   
## 3rd Qu.: 71.96 3rd Qu.:57.20 3rd Qu.: 74.29 3rd Qu.:290.0   
## Max. :100.04 Max. :78.08 Max. :100.00 Max. :360.0   
##   
## wind\_speed precip visib   
## Min. : 0.000 Min. :0.000000 Min. : 0.000   
## 1st Qu.: 6.905 1st Qu.:0.000000 1st Qu.:10.000   
## Median :10.357 Median :0.000000 Median :10.000   
## Mean :11.202 Mean :0.004059 Mean : 9.294   
## 3rd Qu.:14.960 3rd Qu.:0.000000 3rd Qu.:10.000   
## Max. :42.579 Max. :1.210000 Max. :10.000   
##   
## name lat   
## Hartsfield Jackson Atlanta Intl : 9726 Min. :21.32   
## Chicago Ohare Intl : 9443 1st Qu.:32.90   
## Los Angeles Intl : 9185 Median :36.08   
## General Edward Lawrence Logan Intl: 8674 Mean :35.97   
## Orlando Intl : 8131 3rd Qu.:41.41   
## Charlotte Douglas Intl : 7822 Max. :61.17   
## (Other) :131335   
## lon alt tz dst   
## Min. :-157.92 Min. : 3.0 Min. :-10.000 A:181313   
## 1st Qu.: -95.34 1st Qu.: 26.0 1st Qu.: -6.000 N: 3003   
## Median : -83.35 Median : 433.0 Median : -5.000   
## Mean : -89.58 Mean : 582.3 Mean : -5.757   
## 3rd Qu.: -80.15 3rd Qu.: 748.0 3rd Qu.: -5.000   
## Max. : -68.83 Max. :6602.0 Max. : -5.000   
##   
## tzone   
## America/Anchorage : 3   
## America/Chicago : 41444   
## America/Denver : 5819   
## America/Los\_Angeles: 26461   
## America/New\_York :107586   
## America/Phoenix : 2629   
## Pacific/Honolulu : 374

### Summaries of the response variable dep\_delay

The departure delays variable is highly right-skewed.

range(fl$dep\_delay)

## [1] -43 1301

fivenum(fl$dep\_delay)

## [1] -43 -5 -2 11 1301

quantile(fl$dep\_delay,probs = c(0.01,0.05,0.1,0.25,.5,.75,.90,.95,.99))

## 1% 5% 10% 25% 50% 75% 90% 95% 99%   
## -12 -9 -7 -5 -2 11 49 88 193

mean(fl$dep\_delay >= 60) # about 15,000 or 8% of flights

## [1] 0.08210356

###Summaries of departure delay by NYC airport:

Q3 <- function(x) { quantile(x,probs=.75) }  
fl %>% group\_by(origin) %>%   
 summarize(n=n(),med\_d = median(dep\_delay),Q3\_d = Q3(dep\_delay), max\_d = max(dep\_delay)) %>%   
 arrange(desc(Q3\_d)) %>% head(10)

## # A tibble: 3 x 5  
## origin n med\_d Q3\_d max\_d  
## <fct> <int> <dbl> <dbl> <int>  
## 1 EWR 65512 -1 15 896  
## 2 JFK 60327 -1 10 1301  
## 3 LGA 58477 -3 7 911

###Summaries of departure delay by airline (carrier).

fl %>% group\_by(carrier) %>%   
 summarize(n=n(),med\_d = median(dep\_delay),Q3\_d = Q3(dep\_delay), max\_d = max(dep\_delay)) %>%   
 arrange(desc(Q3\_d)) %>% head(10)

## # A tibble: 10 x 5  
## carrier n med\_d Q3\_d max\_d  
## <fct> <int> <dbl> <dbl> <int>  
## 1 EV 29137 -1 25 536  
## 2 WN 6897 1 18 471  
## 3 F9 388 0 17.2 853  
## 4 9E 10179 -2 16 430  
## 5 FL 1832 1 16 602  
## 6 YV 312 -3 13 387  
## 7 B6 29282 -1 12 502  
## 8 UA 32252 0 11 483  
## 9 MQ 14382 -3 9 486  
## 10 VX 2991 0 7 653

fl %>% group\_by(origin,carrier) %>%   
 summarize(n=n(),med\_d = median(dep\_delay),Q3\_d = Q3(dep\_delay), max\_d = max(dep\_delay)) %>%   
 arrange(desc(Q3\_d)) %>% head(10)

## # A tibble: 10 x 6  
## # Groups: origin [3]  
## origin carrier n med\_d Q3\_d max\_d  
## <fct> <fct> <int> <dbl> <dbl> <int>  
## 1 EWR OO 3 4 67.5 131  
## 2 EWR EV 23565 -1 26 443  
## 3 LGA EV 4769 -2 22 473  
## 4 JFK 9E 8126 -1 20 430  
## 5 JFK EV 803 -2 19 536  
## 6 EWR WN 3487 2 18 440  
## 7 LGA WN 3410 1 18 471  
## 8 LGA F9 388 0 17.2 853  
## 9 EWR MQ 1156 -2 17 381  
## 10 LGA FL 1832 1 16 602

fl %>% group\_by(dest,carrier) %>%   
 summarize(n=n(),med\_d = median(dep\_delay),Q3\_d = Q3(dep\_delay), max\_d = max(dep\_delay)) %>%   
 arrange(desc(Q3\_d)) %>% head(10)

## # A tibble: 10 x 6  
## # Groups: dest [10]  
## dest carrier n med\_d Q3\_d max\_d  
## <fct> <fct> <int> <dbl> <dbl> <int>  
## 1 STL UA 2 77.5 116. 155  
## 2 DTW OO 2 61 96 131  
## 3 TYS EV 183 8 68.5 285  
## 4 PBI EV 3 50 67.5 85  
## 5 ORD OO 1 67 67 67  
## 6 RDU UA 1 60 60 60  
## 7 TUL EV 185 3 53 251  
## 8 OKC EV 184 8.5 51.5 207  
## 9 BHM EV 175 3 50 325  
## 10 CAE EV 57 10 48 163

###Summaries of departure delay by date:

fl %>% group\_by(month,day) %>%   
 summarize(n=n(),med\_d = mean(dep\_delay),max\_d = max(dep\_delay)) %>%   
 arrange(desc(med\_d)) %>% head(10)

## # A tibble: 10 x 5  
## # Groups: month [7]  
## month day n med\_d max\_d  
## <int> <int> <int> <dbl> <int>  
## 1 3 8 461 79.5 470  
## 2 7 1 505 58.1 363  
## 3 7 10 471 56.6 576  
## 4 9 2 438 53.7 696  
## 5 12 5 458 52.2 896  
## 6 5 23 453 51.5 410  
## 7 4 19 511 50.4 812  
## 8 9 12 444 50.4 602  
## 9 6 13 469 50.3 388  
## 10 7 22 476 49.9 898

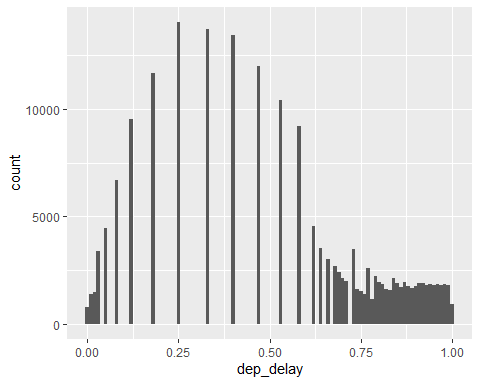
###Summaries of departure delay by precipitation:

fl %>% mutate(haveprecip = factor(precip>0)) %>% group\_by(haveprecip) %>%   
 summarize(n=n(),med\_d = median(dep\_delay),Q3\_d = Q3(dep\_delay), max\_d = max(dep\_delay)) %>%   
 arrange(desc(med\_d)) %>% head(10)

## # A tibble: 2 x 5  
## haveprecip n med\_d Q3\_d max\_d  
## <fct> <int> <dbl> <dbl> <int>  
## 1 TRUE 11804 5 41 853  
## 2 FALSE 172512 -2 9 1301

### Ranking and scaling dep\_delay

den <- nrow(fl)+1  
fl <- fl %>% mutate(dep\_delay = rank(dep\_delay)/den)  
ggplot(fl,aes(x=dep\_delay)) + geom\_histogram(binwidth=.01)



###data wrangling

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

fl <- fl %>%   
 mutate(dep\_date = make\_date(year.x,month,day)) %>%   
 select(-year.x,-month,-day,-dep\_time,-arr\_time,-arr\_delay,  
 -sched\_arr\_time,-tailnum,-flight,-name,-air\_time,  
 -hour,-minute,-time\_hour,-tz,-dst,-dest) %>%  
 mutate(precip = as.numeric(precip>0))  
fl <- mutate(fl,logalt = log(alt)) %>% select(-alt)  
fl <- mutate(fl,logdistance = log(distance)) %>% select(-distance)

## Associations between dep\_delay and quantitative predictors

ggplot(fl,aes(x=dep\_date,y=dep\_delay)) + geom\_point(alpha=.01) + geom\_smooth()  
  
ggplot(fl,aes(x=sched\_dep\_time,y=dep\_delay)) + geom\_point(alpha=0.01) + geom\_smooth()  
  
  
ggplot(fl,aes(x=log(distance),y=dep\_delay)) + geom\_point(alpha=0.01) + geom\_smooth()  
  
  
ggplot(fl,aes(x=temp,y=dep\_delay)) + geom\_point(alpha=0.01) + geom\_smooth()  
  
ggplot(fl,aes(x=dewp,y=dep\_delay)) + geom\_point(alpha=0.01) + geom\_smooth()

## Split training set in two for tuning

set.seed(123)  
tr\_size <- ceiling(2\*nrow(fl)/3)  
train <- sample(1:nrow(fl),size=tr\_size)  
fl\_tr <- fl[train,]  
fl\_te <- fl[-train,]

## Learning methods

###Random Forest

library(randomForest)  
rf.fit=randomForest(dep\_delay∼.,data=fl\_tr[-5], mtry=6,importance=TRUE,ntree=100)

###GAM

library(gam)

## Loading required package: splines

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded gam 1.16.1

form <- formula(dep\_delay ~ s(dep\_date) + s(sched\_dep\_time) + carrier + origin +tzone + s(logdistance) +  
 s(temp) + s(dewp) + s(humid) + s(wind\_dir) + s(wind\_speed) + precip + s(visib))  
gam\_fit <- gam(form, data=fl\_tr,family=gaussian)

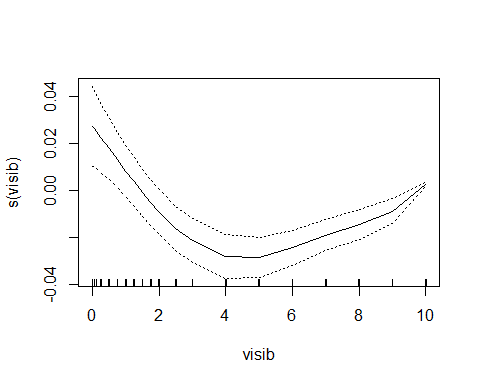
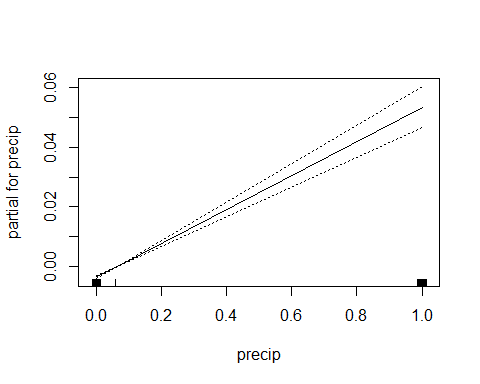
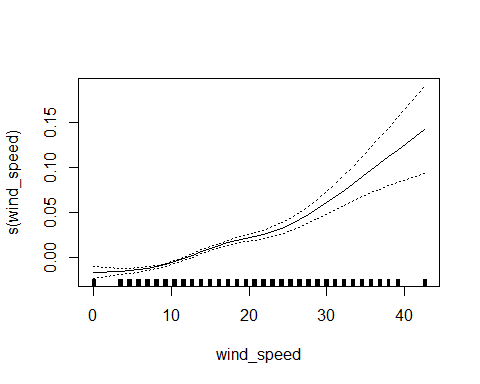
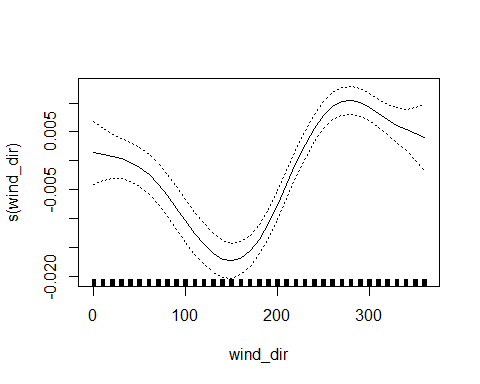
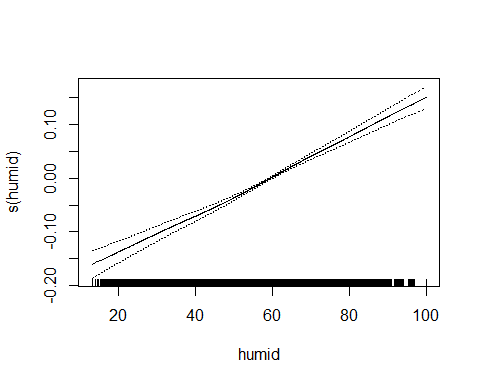
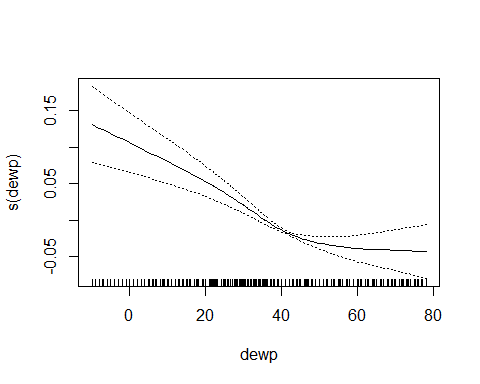
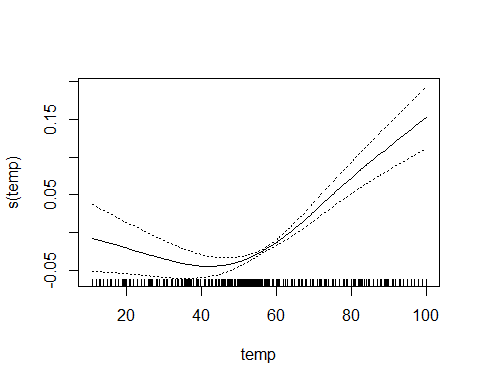
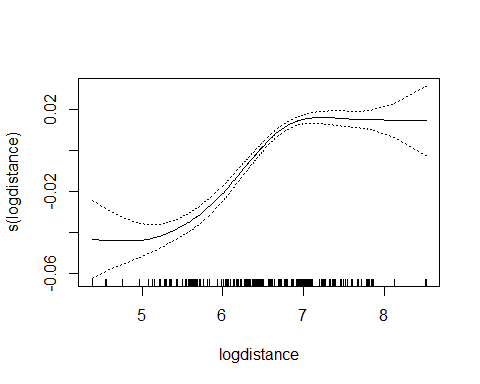
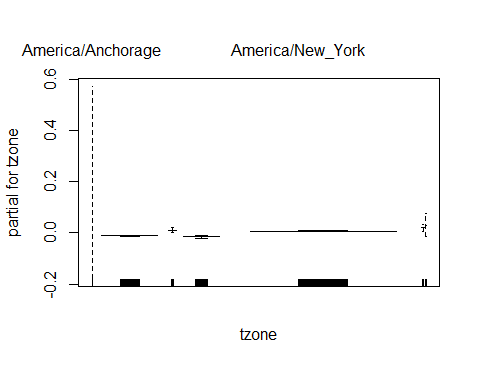
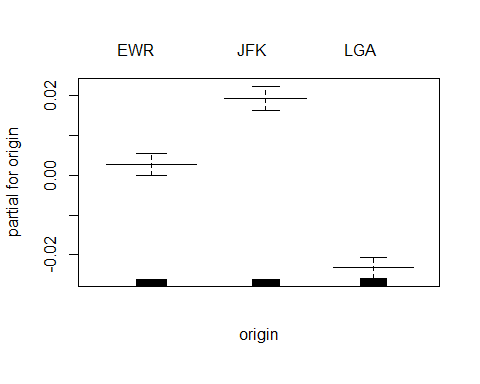
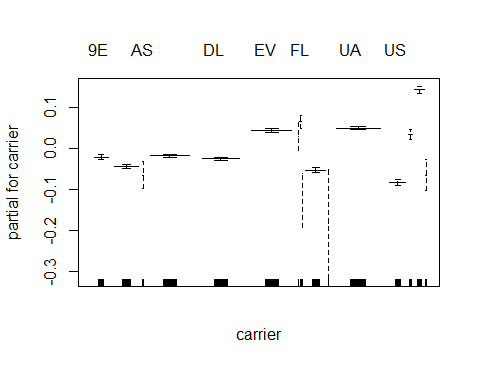
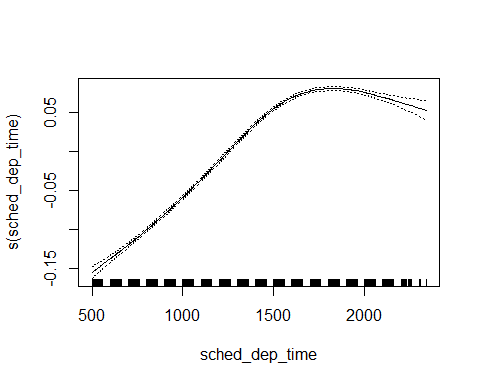
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts  
## argument ignored

summary(gam\_fit)

##   
## Call: gam(formula = form, family = gaussian, data = fl\_tr)  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -0.837858 -0.210733 0.004967 0.213007 0.767894   
##   
## (Dispersion Parameter for gaussian family taken to be 0.0704)  
##   
## Null Deviance: 10216.89 on 122877 degrees of freedom  
## Residual Deviance: 8640.651 on 122817 degrees of freedom  
## AIC: 22630.71   
##   
## Number of Local Scoring Iterations: 3   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## s(dep\_date) 1 0.3 0.25 3.5817 0.0584228 .   
## s(sched\_dep\_time) 1 582.7 582.71 8282.5374 < 2.2e-16 \*\*\*  
## carrier 15 339.4 22.63 321.6176 < 2.2e-16 \*\*\*  
## origin 2 63.2 31.60 449.1553 < 2.2e-16 \*\*\*  
## tzone 6 8.6 1.44 20.4176 < 2.2e-16 \*\*\*  
## s(logdistance) 1 12.9 12.91 183.5427 < 2.2e-16 \*\*\*  
## s(temp) 1 5.1 5.14 73.0613 < 2.2e-16 \*\*\*  
## s(dewp) 1 288.8 288.78 4104.6842 < 2.2e-16 \*\*\*  
## s(humid) 1 36.2 36.22 514.8924 < 2.2e-16 \*\*\*  
## s(wind\_dir) 1 9.7 9.73 138.2674 < 2.2e-16 \*\*\*  
## s(wind\_speed) 1 22.9 22.86 324.9140 < 2.2e-16 \*\*\*  
## precip 1 15.9 15.86 225.4495 < 2.2e-16 \*\*\*  
## s(visib) 1 0.9 0.87 12.3623 0.0004383 \*\*\*  
## Residuals 122817 8640.7 0.07   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)   
## (Intercept)   
## s(dep\_date) 3 492.04 < 2.2e-16 \*\*\*  
## s(sched\_dep\_time) 3 295.86 < 2.2e-16 \*\*\*  
## carrier   
## origin   
## tzone   
## s(logdistance) 3 17.82 1.479e-11 \*\*\*  
## s(temp) 3 253.61 < 2.2e-16 \*\*\*  
## s(dewp) 3 97.38 < 2.2e-16 \*\*\*  
## s(humid) 3 3.50 0.01482 \*   
## s(wind\_dir) 3 51.19 < 2.2e-16 \*\*\*  
## s(wind\_speed) 3 12.34 4.597e-08 \*\*\*  
## precip   
## s(visib) 3 26.77 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(gam\_fit,se=TRUE)

## Warning in gplot.default(x = structure(c(15823, 15947, 15822, 16027,  
## 15898, : The "x" component of "s(dep\_date)" has class "Date"; no gplot()  
## methods available



gam\_pred <- predict(gam\_fit,newdata=fl\_te)  
mse\_gam <- mean((fl\_te$dep\_delay-gam\_pred)^2)  
mse\_gam

## [1] 0.07066494

###Gradient Boost running with default hyperparameters

library(gbm)  
dep\_date\_numeric <- as.numeric(fl\_tr$dep\_date)  
dep\_date\_numeric <- dep\_date\_numeric - mean(dep\_date\_numeric)  
fl\_tr\_tem <- mutate(fl\_tr,dep\_date = dep\_date\_numeric)  
gbm\_fit <-gbm(dep\_delay ~ .,data=fl\_tr\_tem,distribution="gaussian",  
 n.trees = 1000, shrinkage = 0.01,interaction.depth = 1)  
  
dep\_date\_numeric <- as.numeric(fl\_te$dep\_date)  
dep\_date\_numeric <- dep\_date\_numeric - mean(dep\_date\_numeric)  
fl\_te\_tem <- mutate(fl\_te,dep\_date = dep\_date\_numeric)  
  
gbm\_pred <- predict(gbm\_fit,newdata=fl\_te\_tem,n.trees = 1000)  
mse\_gbm <- mean((fl\_te$dep\_delay-gbm\_pred)^2)  
mse\_gbm

##Hyper Tuning of GBM

dvalues <- matrix(ncol=3,nrow = 4)  
ntrees\_list<-c(1000,2000)  
list\_shrink<-c(0.1,0.2)  
j<-1  
for(tree\_i in ntrees\_list)  
{  
 for(shrinkage\_k in list\_shrink){  
 gbm\_fit <-gbm(dep\_delay ~ .,data=fl\_tr\_tem,distribution="gaussian",  
 n.trees = tree\_i, shrinkage = shrinkage\_k, interaction.depth = 3)  
   
 gbm\_pred <- predict(gbm\_fit,newdata=fl\_te\_tem,n.trees =tree\_i)  
 mse\_gbm <- mean((fl\_te$dep\_delay-gbm\_pred)^2)  
 dvalues[j,1] <- tree\_i  
 dvalues[j,2] <- shrinkage\_k  
 dvalues[j,3] <- mse\_gbm  
 j<-j+1  
   
 }  
 }  
 index<-which(dvalues[,3] == min(dvalues[,3]), arr.ind = T)   
 best\_tree<-dvalues[index,1]  
 best\_shrink<-dvalues[index,2]

###Analyzing mse\_list, the last value of parameters, shrinkage=0.2 and trees=2000 give best result.

###Training with tuned paramaters

library(gbm)

## Loaded gbm 2.1.5

dep\_date\_numeric <- as.numeric(fl\_tr$dep\_date)  
dep\_date\_numeric <- dep\_date\_numeric - mean(dep\_date\_numeric)  
fl\_tr\_tem <- mutate(fl\_tr,dep\_date = dep\_date\_numeric)  
gbm\_fit <-gbm(dep\_delay ~ .,data=fl\_tr\_tem,distribution="gaussian",  
 n.trees = 2000, shrinkage = 0.2,interaction.depth = 3)  
  
dep\_date\_numeric <- as.numeric(fl\_te$dep\_date)  
dep\_date\_numeric <- dep\_date\_numeric - mean(dep\_date\_numeric)  
fl\_te\_tem <- mutate(fl\_te,dep\_date = dep\_date\_numeric)  
  
gbm\_pred <- predict(gbm\_fit,newdata=fl\_te\_tem,n.trees = 2000)  
mse\_gbm <- mean((fl\_te$dep\_delay-gbm\_pred)^2)  
mse\_gbm

## [1] 0.06373986

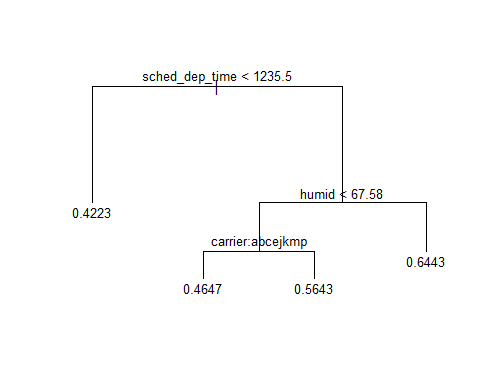
##Tree

library(tree)

## Registered S3 method overwritten by 'tree':  
## method from  
## print.tree cli

tree.pred<-tree(dep\_delay ~., data=fl\_tr\_tem)  
yhat=predict(tree.pred ,newdata=fl\_te\_tem)   
tree.test=fl\_te\_tem$dep\_delay

plot(tree.pred)  
text(tree.pred,cex=.8)



mse\_t<-mean((yhat -tree.test)^2)  
mse\_t

## [1] 0.07628095

##Using xgboost

library(xgboost)

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

classifier = xgboost(data = data.matrix(fl\_tr\_tem[-2]), label = fl\_tr\_tem$dep\_delay, nrounds =100)

## [1] train-rmse:0.277630   
## [2] train-rmse:0.271640   
## [3] train-rmse:0.267499   
## [4] train-rmse:0.264834   
## [5] train-rmse:0.262871   
## [6] train-rmse:0.261236   
## [7] train-rmse:0.260135   
## [8] train-rmse:0.259173   
## [9] train-rmse:0.258136   
## [10] train-rmse:0.257447   
## [11] train-rmse:0.256983   
## [12] train-rmse:0.256286   
## [13] train-rmse:0.255472   
## [14] train-rmse:0.255007   
## [15] train-rmse:0.254648   
## [16] train-rmse:0.254374   
## [17] train-rmse:0.253628   
## [18] train-rmse:0.253014   
## [19] train-rmse:0.252776   
## [20] train-rmse:0.252459   
## [21] train-rmse:0.251957   
## [22] train-rmse:0.251784   
## [23] train-rmse:0.251357   
## [24] train-rmse:0.250896   
## [25] train-rmse:0.250390   
## [26] train-rmse:0.250216   
## [27] train-rmse:0.249991   
## [28] train-rmse:0.249665   
## [29] train-rmse:0.249394   
## [30] train-rmse:0.249226   
## [31] train-rmse:0.248736   
## [32] train-rmse:0.248459   
## [33] train-rmse:0.248196   
## [34] train-rmse:0.248009   
## [35] train-rmse:0.247643   
## [36] train-rmse:0.247277   
## [37] train-rmse:0.247164   
## [38] train-rmse:0.246994   
## [39] train-rmse:0.246787   
## [40] train-rmse:0.246587   
## [41] train-rmse:0.246423   
## [42] train-rmse:0.246117   
## [43] train-rmse:0.246000   
## [44] train-rmse:0.245854   
## [45] train-rmse:0.245506   
## [46] train-rmse:0.245402   
## [47] train-rmse:0.245287   
## [48] train-rmse:0.245017   
## [49] train-rmse:0.244816   
## [50] train-rmse:0.244688   
## [51] train-rmse:0.244420   
## [52] train-rmse:0.244262   
## [53] train-rmse:0.244062   
## [54] train-rmse:0.243899   
## [55] train-rmse:0.243737   
## [56] train-rmse:0.243636   
## [57] train-rmse:0.243396   
## [58] train-rmse:0.243202   
## [59] train-rmse:0.243097   
## [60] train-rmse:0.243037   
## [61] train-rmse:0.242965   
## [62] train-rmse:0.242808   
## [63] train-rmse:0.242596   
## [64] train-rmse:0.242454   
## [65] train-rmse:0.242283   
## [66] train-rmse:0.242066   
## [67] train-rmse:0.241837   
## [68] train-rmse:0.241718   
## [69] train-rmse:0.241453   
## [70] train-rmse:0.241198   
## [71] train-rmse:0.240918   
## [72] train-rmse:0.240773   
## [73] train-rmse:0.240618   
## [74] train-rmse:0.240405   
## [75] train-rmse:0.240165   
## [76] train-rmse:0.240054   
## [77] train-rmse:0.239913   
## [78] train-rmse:0.239757   
## [79] train-rmse:0.239703   
## [80] train-rmse:0.239495   
## [81] train-rmse:0.239360   
## [82] train-rmse:0.239212   
## [83] train-rmse:0.239109   
## [84] train-rmse:0.239080   
## [85] train-rmse:0.238946   
## [86] train-rmse:0.238727   
## [87] train-rmse:0.238544   
## [88] train-rmse:0.238341   
## [89] train-rmse:0.238091   
## [90] train-rmse:0.237981   
## [91] train-rmse:0.237821   
## [92] train-rmse:0.237658   
## [93] train-rmse:0.237433   
## [94] train-rmse:0.237322   
## [95] train-rmse:0.237085   
## [96] train-rmse:0.236925   
## [97] train-rmse:0.236797   
## [98] train-rmse:0.236629   
## [99] train-rmse:0.236448   
## [100] train-rmse:0.236409

xgb\_pred<-predict(classifier,data.matrix(fl\_te\_tem[-2]))  
mse\_xgb<-mean((xgb\_pred- fl\_te\_tem[,2])^2)  
mse\_xgb

## [1] 0.06364874

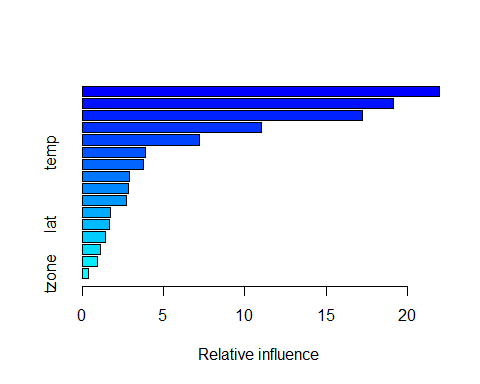
##Running on Test Set

###Prepare Test Data and perform same preprocessing as while during training ###Predicting Fitted Tuned Model on Test Set

fltest <- read.csv("fltest.csv.gz")  
flt <- fltest  
for(i in 1:ncol(fl)) {  
 if(typeof(fl[[i]]) == "character") {  
 flt[[i]] <- factor(flt[[i]])  
 }  
}  
  
p<-function(x) {sum(is.na(x))/length(x)\*100}  
apply(flt,2,p)

## year.x month day dep\_time sched\_dep\_time   
## 0.0000000 0.0000000 0.0000000 2.4543780 0.0000000   
## dep\_delay arr\_time sched\_arr\_time arr\_delay carrier   
## 2.4543780 2.5910979 0.0000000 2.8118968 0.0000000   
## flight tailnum origin dest air\_time   
## 0.0000000 0.7457449 0.0000000 0.0000000 2.8118968   
## distance hour minute time\_hour temp   
## 0.0000000 0.0000000 0.0000000 0.0000000 0.4569515   
## dewp humid wind\_dir wind\_speed wind\_gust   
## 0.4569515 0.4569515 2.8762356 0.4766918 76.1325086   
## precip pressure visib name lat   
## 0.4525648 11.4756975 0.4525648 2.2796397 2.2796397   
## lon alt tz dst tzone   
## 2.2796397 2.2796397 2.2796397 2.2796397 2.2796397   
## year.y type manufacturer model engines   
## 17.2647248 15.6774580 15.6774580 15.6774580 15.6774580   
## seats speed engine   
## 15.6774580 99.7236357 15.6774580

flt <- flt%>%   
 select(-type,-manufacturer,-model,-engines,-seats, -speed, -engine,-wind\_gust,-pressure,-year.y)  
  
flt <- na.omit(flt)  
  
dent <- nrow(flt)+1  
flt <- flt %>% mutate(dep\_delay = rank(dep\_delay)/dent)  
  
flt <- flt %>%   
 mutate(dep\_date = make\_date(year.x,month,day)) %>%   
 select(-year.x,-month,-day,-dep\_time,-arr\_time,-arr\_delay,  
 -sched\_arr\_time,-tailnum,-flight,-name,-air\_time,  
 -hour,-minute,-time\_hour,-tz,-dst,-dest) %>%  
 mutate(precip = as.numeric(precip>0))  
  
flt <- mutate(flt,logdistance = log(distance)) %>% select(-distance)  
flt <- mutate(flt,logalt = log(alt)) %>% select(-alt)  
flt <- flt %>% mutate(dep\_delay = rank(dep\_delay)/dent)  
dep\_date\_numeric <- as.numeric(flt$dep\_date)  
dep\_date\_numeric <- dep\_date\_numeric - mean(dep\_date\_numeric)  
flt <- mutate(flt,dep\_date = dep\_date\_numeric)  
  
gbm\_pred\_t <- predict(gbm\_fit,newdata=flt,n.trees = 2000)  
summary(gbm\_fit)



## var rel.inf  
## sched\_dep\_time sched\_dep\_time 21.9949294  
## dep\_date dep\_date 19.1624649  
## carrier carrier 17.2504658  
## humid humid 11.0364360  
## dewp dewp 7.1896397  
## temp temp 3.8839357  
## logdistance logdistance 3.7843132  
## wind\_dir wind\_dir 2.8849586  
## wind\_speed wind\_speed 2.8735125  
## origin origin 2.7473873  
## logalt logalt 1.7297329  
## lat lat 1.6778932  
## lon lon 1.3992398  
## precip precip 1.1044229  
## visib visib 0.9288953  
## tzone tzone 0.3517728

mse\_gbm <- mean((flt$dep\_delay-gbm\_pred\_t)^2)  
mse\_gbm

## [1] 0.06347855