Citibike_ML

Hongyang Yang

12/4/2017

Purpose: Is there a linear relationship bewteen the number of rents per day and the weather data in NYC?

```
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13
```

```
library(ISLR)
library(tree)
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(e1071)
library(MASS)
library(caret)
```

```
## Loading required package: lattice
```

```
library(gbm)
```

```
## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3
```

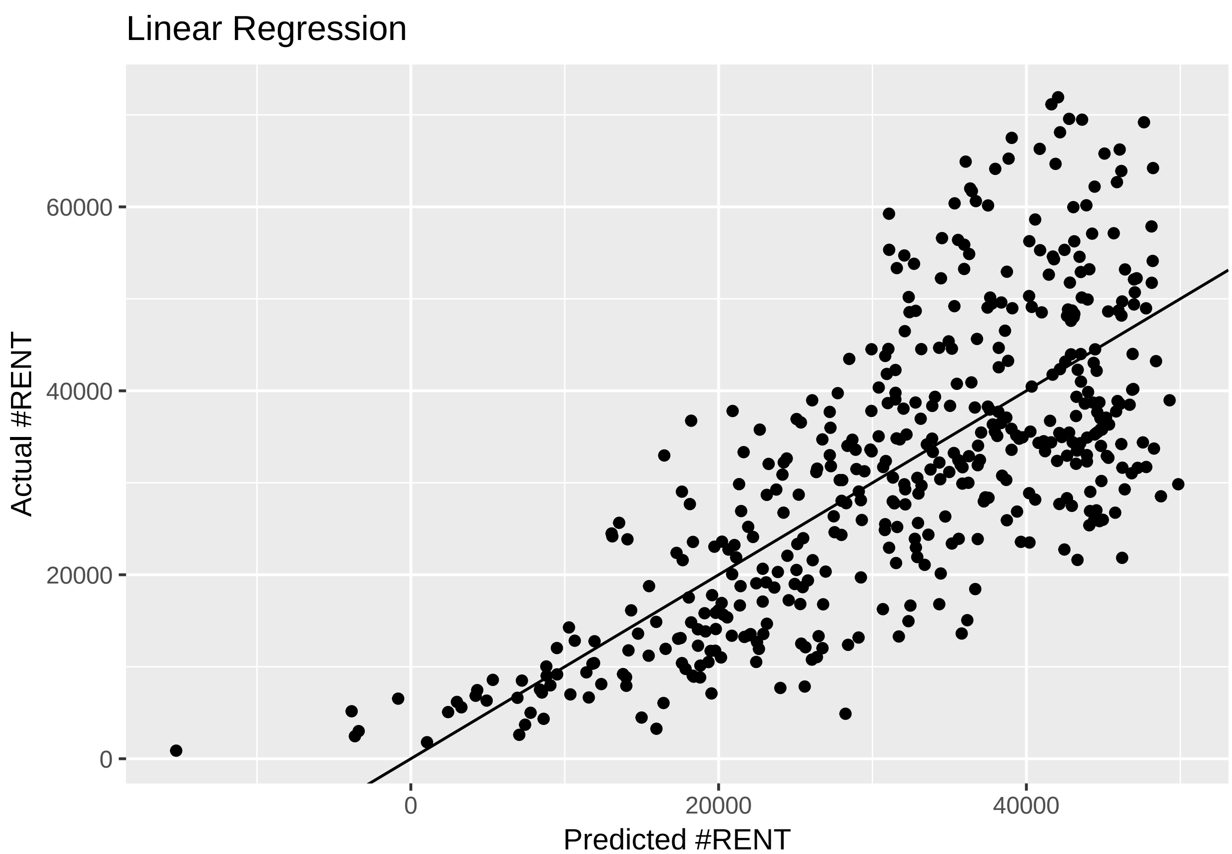
```
citibike_daily_weather=read.csv("citibike_daily_weather.csv")

sum(is.na(citibike_daily_weather))
```

```
## [1] 79
```

```r
citibike_daily_weather=na.omit(citibike_daily_weather)


train_ind=sample(1:nrow(citibike_daily_weather),0.7*nrow(citibike_daily_weather))
train=citibike_daily_weather[train_ind,c(2,5:14)]
test=citibike_daily_weather[-train_ind,c(2,5:14)]

test_x=test[,-1]
test_y=test[,1]
```

Model 1: Linear Regression

```r
###################################
#######Linear Regression##########
###################################

linear_model=lm(RENT~.,data=train)
linear_pre_y=predict(linear_model,test_x)

MSE_linear=mean((linear_pre_y-test_y)^2,na.rm = TRUE)

p.linear<-qplot((linear_pre_y), (test_y), xlab='Predicted #RENT',
              ylab='Actual #RENT', main='Linear Regression')
p.linear + geom_abline(slope=1, intercept=0)
```



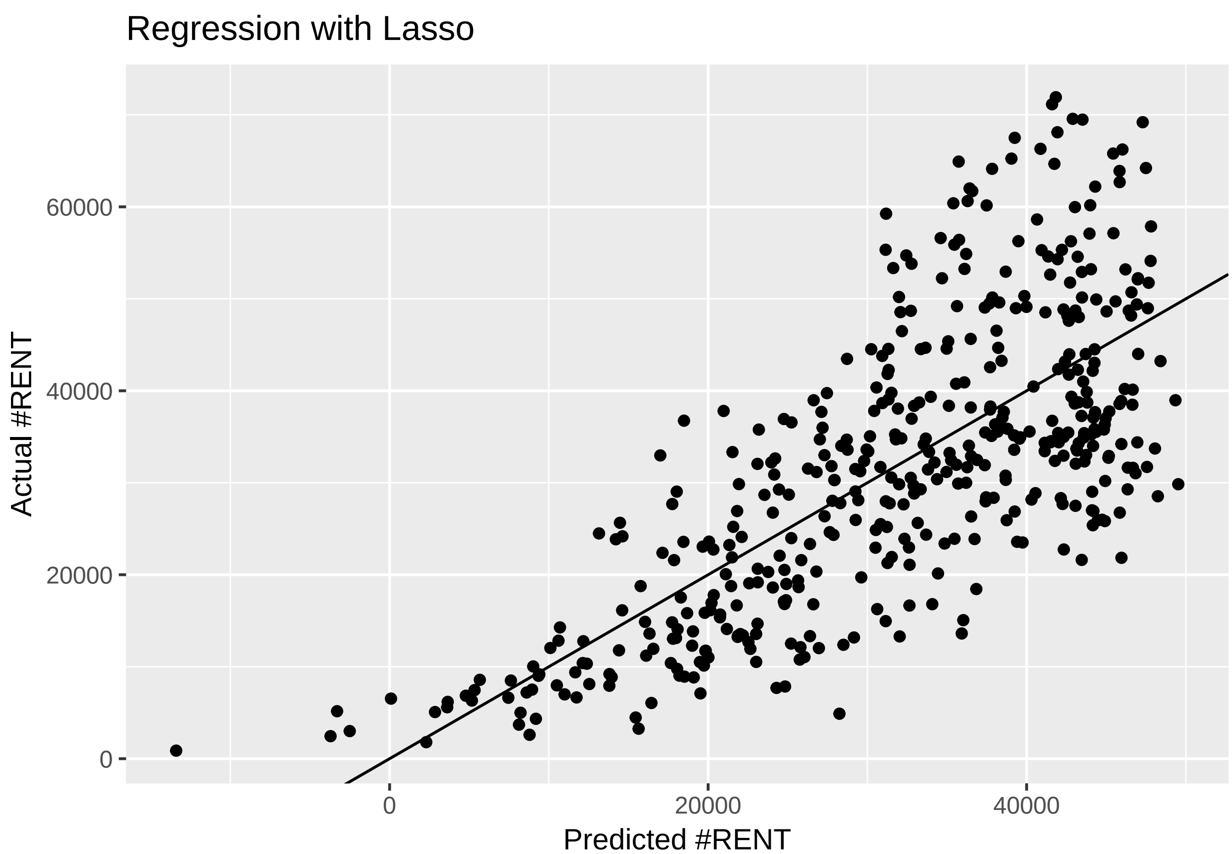
Model 2: Regression with Lasso

```
x_train_lasso=model.matrix(RENT~., train)[,1:(ncol(train)-1)]
y_train_lasso=train$RENT


x_test_lasso=model.matrix(RENT~.,test)[,1:(ncol(test)-1)]
y_test_lasso=test$RENT
grid=10^(-3:3)
#first run lasso on training set and pick the best lambda
cv.out=cv.glmnet(x_train_lasso,y_train_lasso,alpha=1,lambda = grid,nfolds = 5)

bestlam=cv.out$lambda.min

lasso_model=glmnet(x_train_lasso,y_train_lasso,alpha = 1,lambda = bestlam)
lasoo_pred_y=predict(lasso_model,x_test_lasso)

MSE_lasso=mean((lasoo_pred_y-y_test_lasso)^2,na.rm=TRUE)
#summary(lasso.mod)

coef(lasso_model)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
```

```
## (Intercept)  4486.48159
## (Intercept)       .
## AWND         -472.53318
## PRCP        -9338.25035
## SNOW         -277.43779
## SNWD         -866.86534
## TMAX          491.47821
## TMIN           18.52693
## WDF2           -7.38687
## WDF5             .
## WSF2             .
```

```
p.lasso=qplot(as.numeric(lasoo_pred_y),y_test_lasso,xlab = 'Predicted #RENT',
              ylab = 'Actual #RENT', main = 'Regression with Lasso' )
p.lasso + geom_abline(slope=1, intercept=0)
```

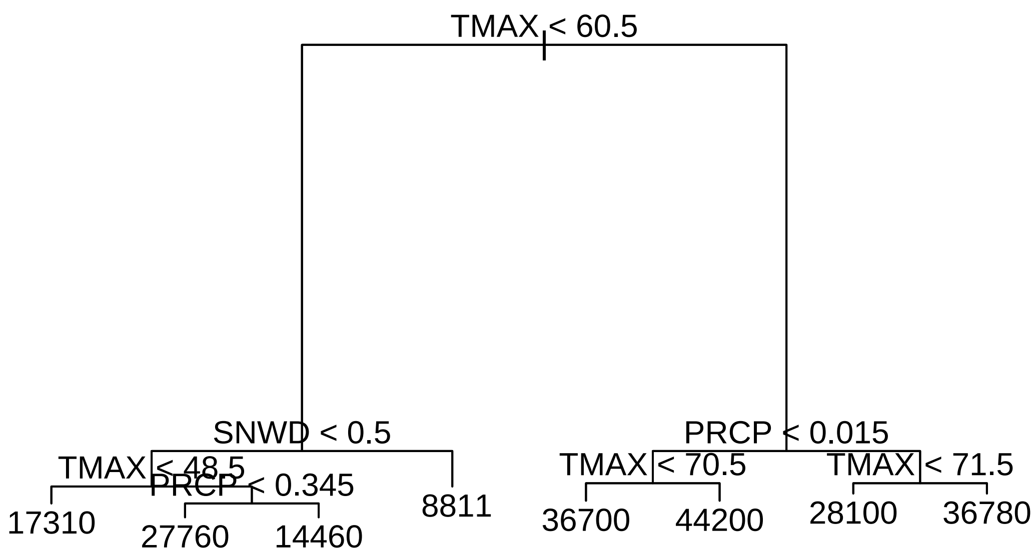



```
#MSE_lasso
```

Model 3: Regression Tree

```
library(ISLR)
library(tree)
#set.seed(1)
tree_model=tree(RENT~.,data=train)
plot(tree_model)
```

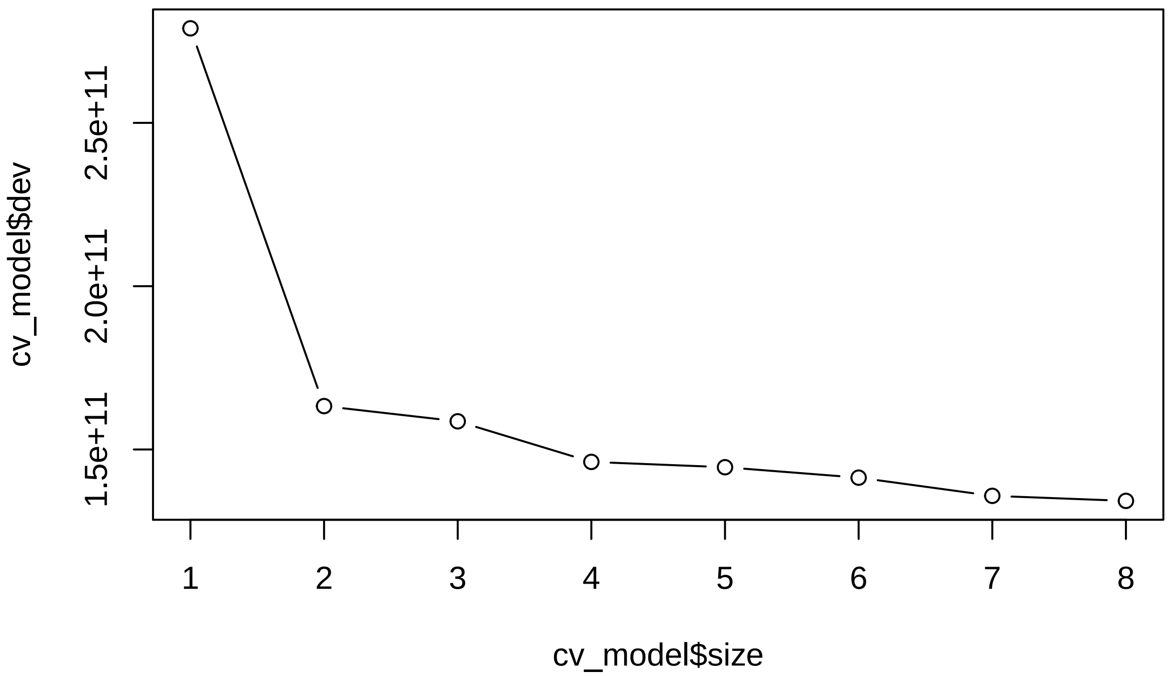
```r
text(tree_model,pretty=1)
```




```r
tree_pred_y=predict(tree_model, test_x)

MSE_tree=mean((test_y-tree_pred_y)^2,na.rm=TRUE)


MSE_tree
```

```
## [1] 126702446
```

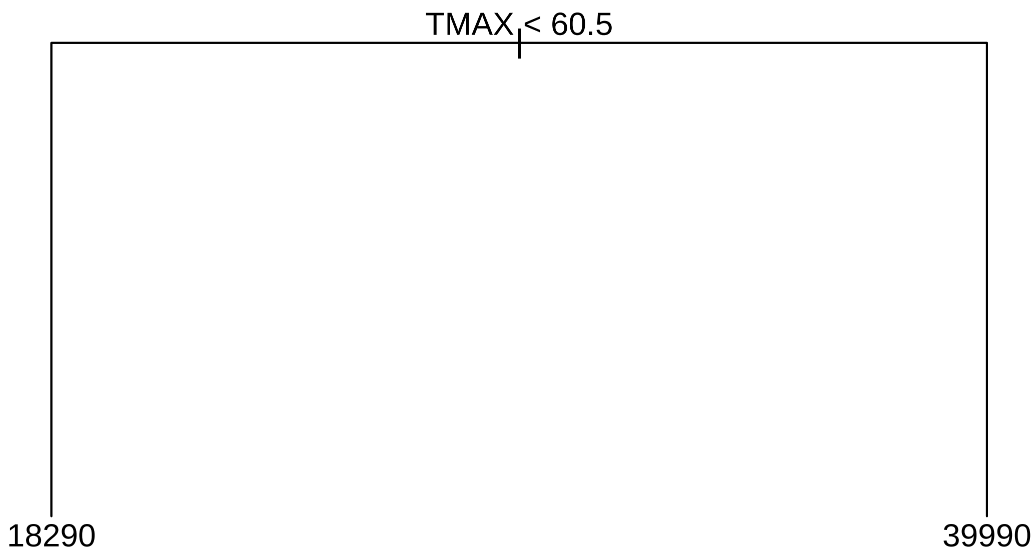
```r
##### CROSS VALIDATION #####
cv_model=cv.tree(tree_model)
plot(cv_model$size,cv_model$dev,type='b')
```




```
bestSize=which.min(cv_model$dev)
print(bestSize)
```

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

```
# Prune Tree
prune.tree=prune.tree(tree_model,best=2)
plot(prune.tree)
text(prune.tree,pretty=0)
```




```
pred.prune.tree = predict(prune.tree, newdata=test)
MSE_prune_tree=mean((test_y-pred.prune.tree)^2)
MSE_prune_tree
```

```
## [1] 157737989
```

Model 4: Random Forest

```
library(randomForest)
library(e1071)
library(MASS)
library(caret)


RF_Model=randomForest(RENT~.,data = na.omit(train) ,importance=TRUE, na.rm = TRUE)

RF_Model
```

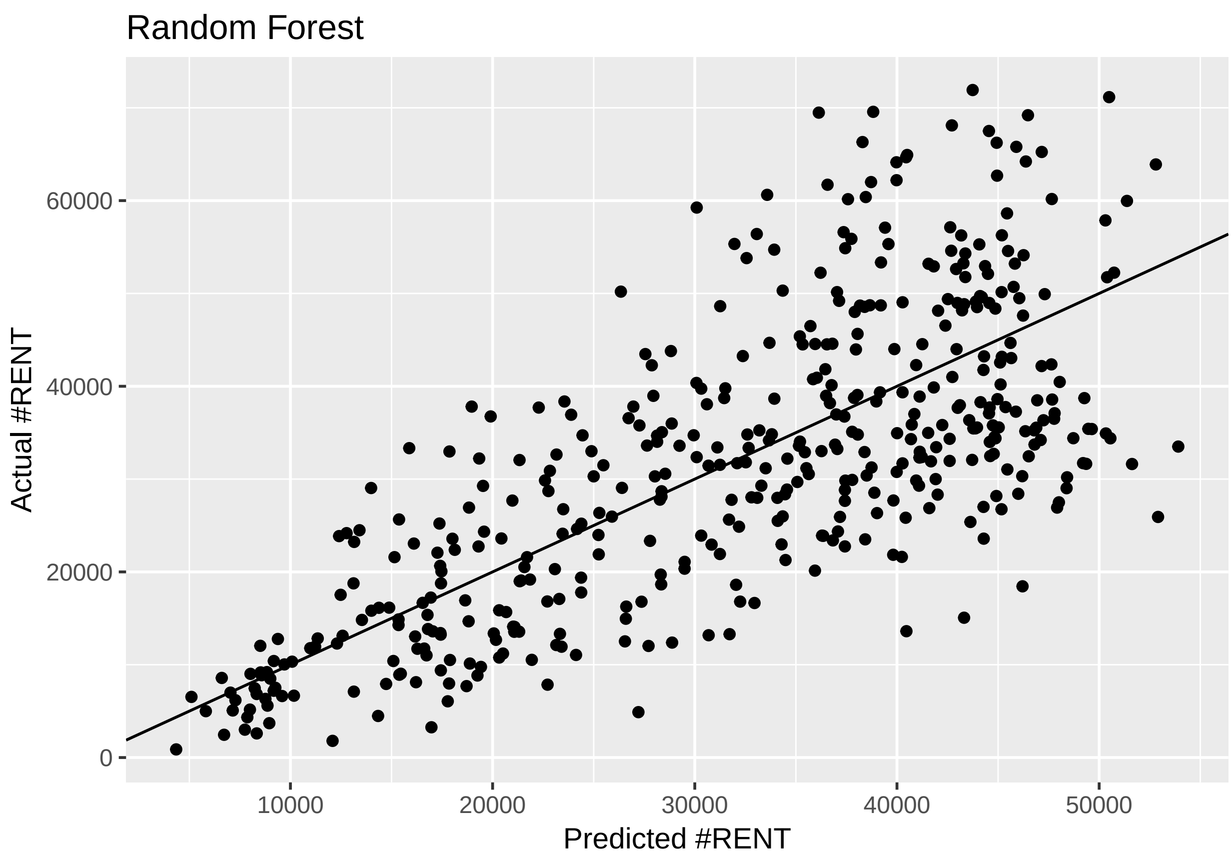
```
##
## Call:
##  randomForest(formula = RENT ~ ., data = na.omit(train), importance = TRUE,      na.rm = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
```

```
##            Mean of squared residuals: 123732835
##                      % Var explained: 53.22
```

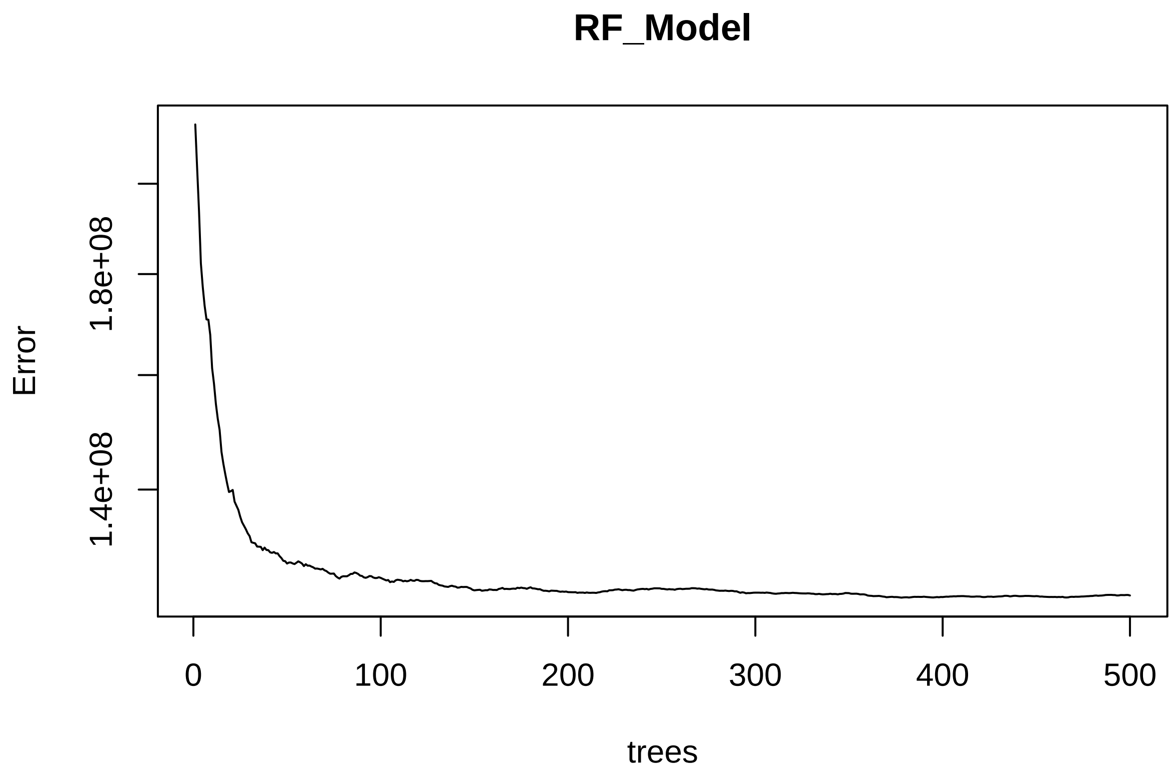
```
yhat_bag=predict(RF_Model,test_x)
MSE_RF=mean((yhat_bag-test_y)^2,na.rm=TRUE)


#running the result
MSE_RF
```

```
## [1] 120538733
```

```
p.rf<-qplot((yhat_bag), (test_y), xlab='Predicted #RENT',
            ylab='Actual #RENT', main='Random Forest')
p.rf + geom_abline(slope=1, intercept=0)
```

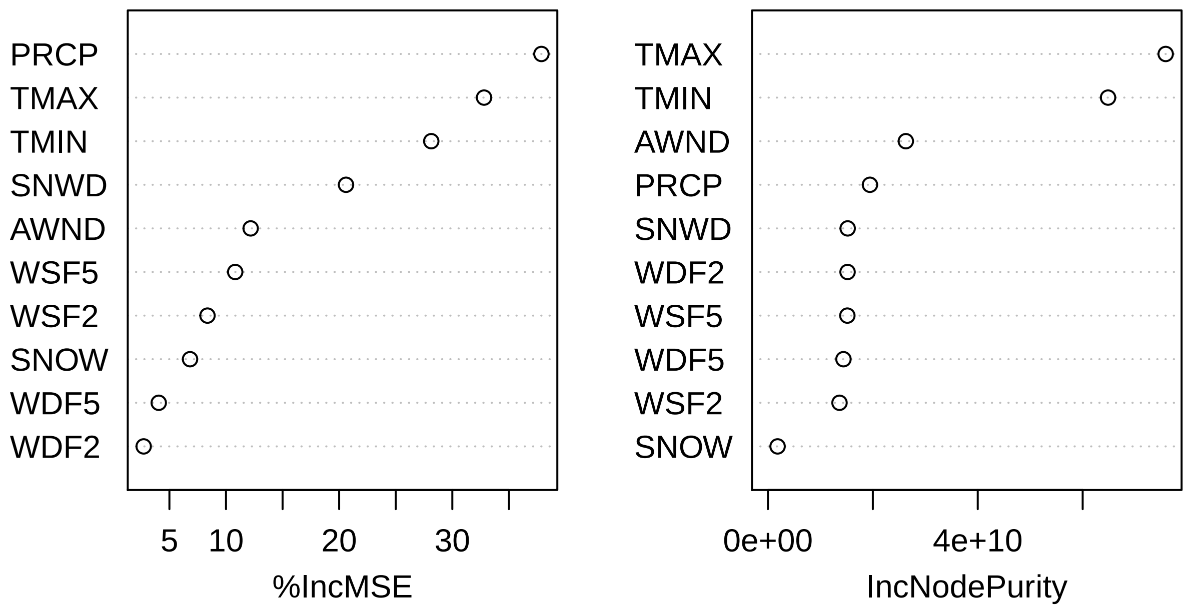



```
plot(RF_Model, log="y")
```




```
varImpPlot(RF_Model,main='Random Forest Importance Table')
```

Random Forest Importance Table

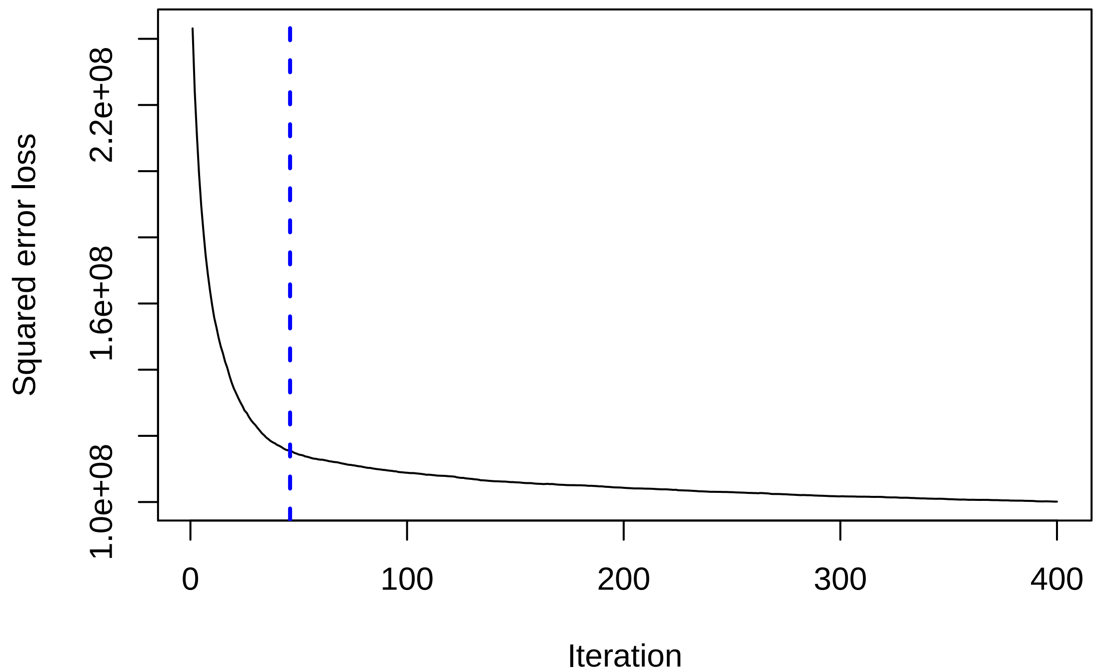



```r
varImp(RF_Model)
```

```
##        Overall
## AWND 12.178141
## PRCP 37.873969
## SNOW  6.823643
## SNWD 20.606322
## TMAX 32.804404
## TMIN 28.139469
## WDF2  2.724899
## WDF5  4.057349
## WSF2  8.363620
## WSF5 10.814144
```

Model 5: GBM

```r
#Generalized Boosted Regression Modeling
library(gbm)
gbm_model=gbm(RENT~.,data = train,dist="gaussian",n.tree = 400,shrinkage=0.1, cv.folds = 5)

best.iter <- gbm.perf(gbm_model,method="OOB")


gbm.perf(gbm_model,method="OOB")
```




```
## [1] 46
```

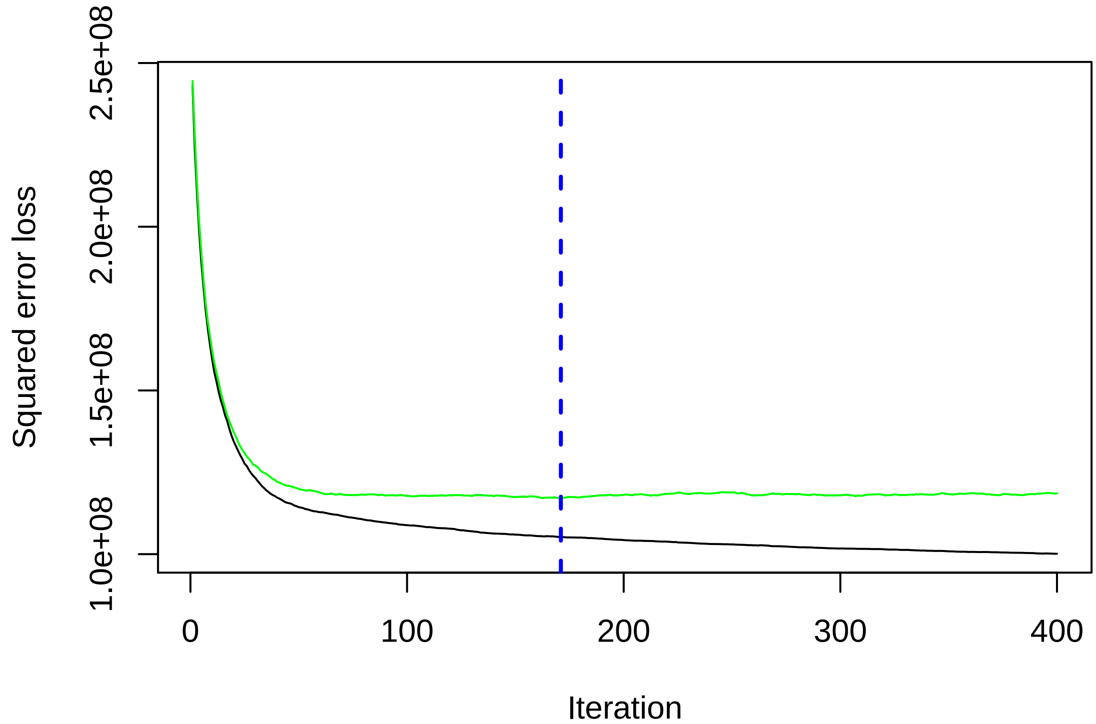
```r
print(best.iter)
```

```
## [1] 46
```

```r
best.iter <- gbm.perf(gbm_model,method="cv")
print(best.iter)
```

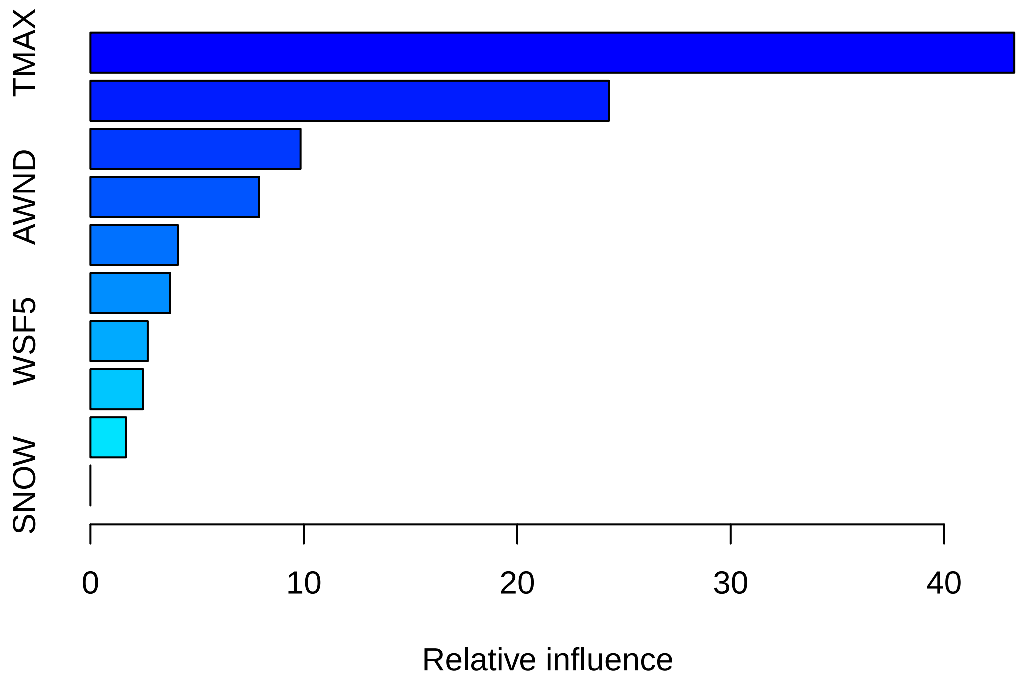
```
## [1] 171
```

```r
gbm.perf(gbm_model,method="cv")
```




```
## [1] 171
```

```
sumary_GBM=summary(gbm_model)
```



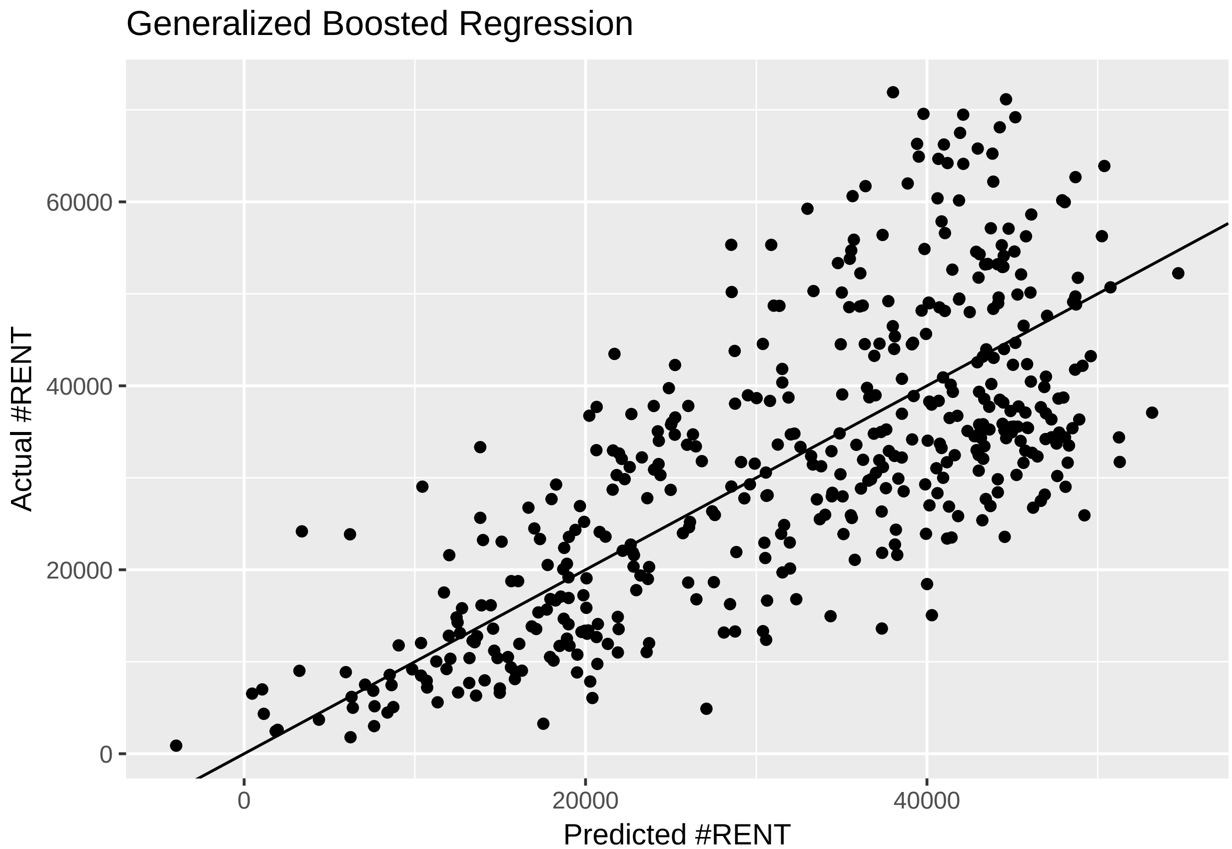

```r
sumary_GBM
```

```
##       var   rel.inf
## TMAX TMAX 43.291558
## TMIN TMIN 24.295489
## PRCP PRCP  9.847885
## AWND AWND  7.903254
## WDF2 WDF2  4.093623
## SNWD SNWD  3.736502
## WSF5 WSF5  2.685781
## WDF5 WDF5  2.471933
## WSF2 WSF2  1.673975
## SNOW SNOW  0.000000
```

```r
gbm_pred_y = predict(gbm_model, test, n.tree = 400, type = 'response')
MSE_gbm=mean((gbm_pred_y-test_y)^2,na.rm=TRUE)
MSE_gbm
```

```
## [1] 121370228
```

```r
p.rf<-qplot((gbm_pred_y), (test_y), xlab='Predicted #RENT',
            ylab='Actual #RENT', main='Generalized Boosted Regression')
p.rf +  geom_abline(slope=1, intercept=0)
```



Conclusion: Based on the 5 models we have, it can be concluded that there is a linear relationship between the number of rents per day and the weather data in NYC.