R Notebook

Gus Lipkin ~ glipkin6737@floridapoly.edu

You are provided the data on home sales in a mid-western city that includes 522 observations. The variables in the dataset in order are sales price ($), finished area of the residence (square feet), number of bedrooms and bathrooms, presence of air conditioning and pool, number of cars the garage will hold, quality of construction (low, medium or high), architectural style, lot size (square feet) and presence or absence of adjacency to a highway.

Please use the following methods to determine the predictors that drive the sale price of a home

```
library(data.table)
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
library(rpart.plot)
```

```
## Loading required package: rpart
```

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

```
## randomForest 4.7-1
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
dt <- fread("APPENC07_Age.txt", col.names = c("salePrice", "sqFt", "bedrooms", "bathrooms", "ac", "cars

intToBool <- function(x) {
  ifelse(x == 1, TRUE, FALSE)
}

dt$ac <- sapply(dt$ac, intToBool)
dt$pool <- sapply(dt$pool, intToBool)

dt$bedrooms <- as.factor(dt$bedrooms)
dt$bathrooms <- as.factor(dt$bathrooms)
dt$cars <- as.factor(dt$cars)
dt$quality <- as.factor(dt$quality)
dt$style <- as.factor(dt$style)
dt$highway <- as.factor(dt$highway)

head(dt)
```

```
##    salePrice sqFt bedrooms bathrooms   ac cars  pool quality style lotSize
## 1:    360000 3032        4         4 TRUE    2 FALSE       2     1   22221
## 2:    340000 2058        4         2 TRUE    2 FALSE       2     1   22912
## 3:    250000 1780        4         3 TRUE    2 FALSE       2     1   21345
## 4:    205500 1638        4         2 TRUE    2 FALSE       2     1   17342
## 5:    275500 2196        4         3 TRUE    2 FALSE       2     7   21786
## 6:    248000 1966        4         3 TRUE    5  TRUE       2     1   18902
##    highway houseAge
## 1:       0       30
## 2:       0       26
## 3:       0       22
## 4:       0       39
## 5:       0       34
## 6:       0       30
```

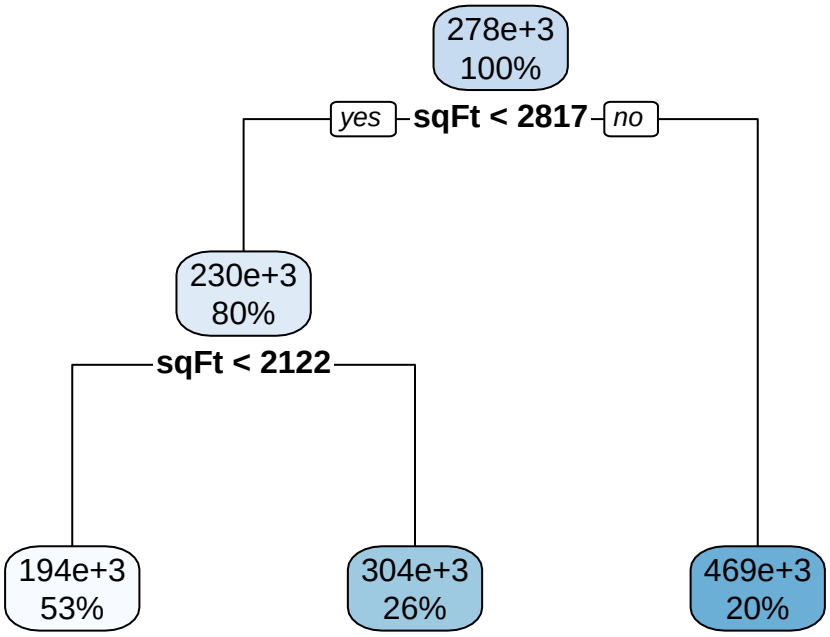
```
set.seed(2022)
rowPicker <- createDataPartition(y=dt$salePrice, p=0.8, list=FALSE)
train <- dt[rowPicker]
test <- dt[-rowPicker]

control <- trainControl(method = "cv", number = 10)
```

Decision Trees

```
rpartTree <- train(salePrice ~ ., train, method = "rpart")
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

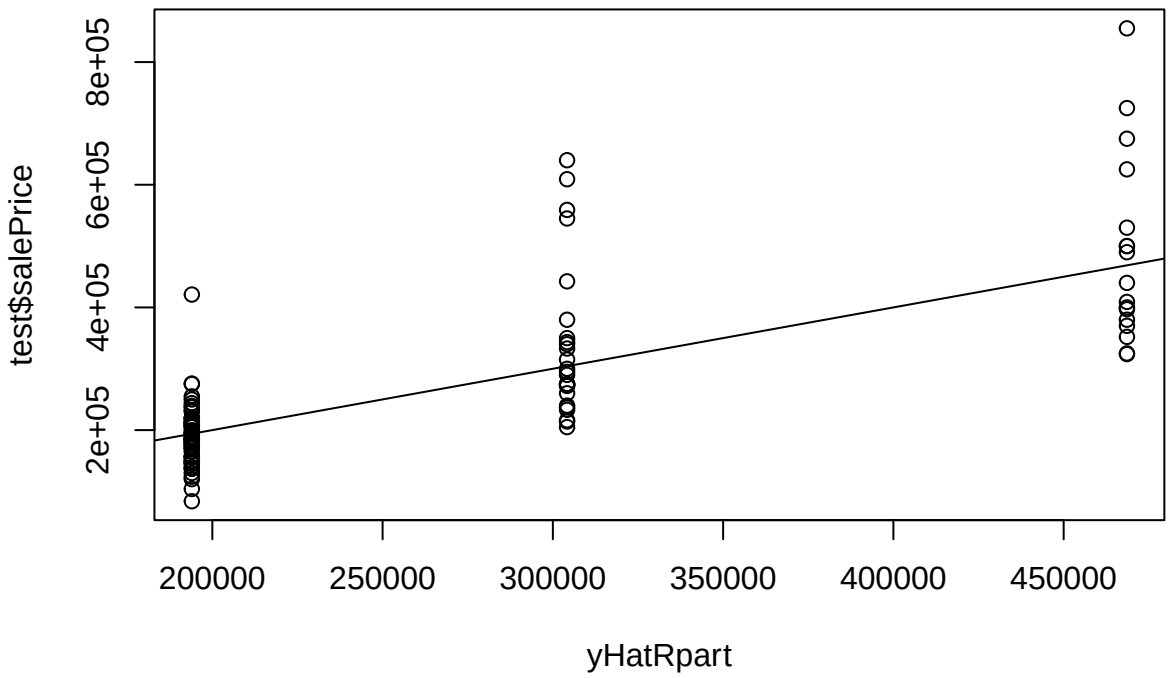
```
rpart.plot(rpartTree$finalModel)
```




```
yHatRpart <- predict(rpartTree, test)
summary(yHatRpart)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  193980  193980  193980  268189  304173  468587
```

```
plot(yHatRpart, test$salePrice)
abline(0, 1)
```

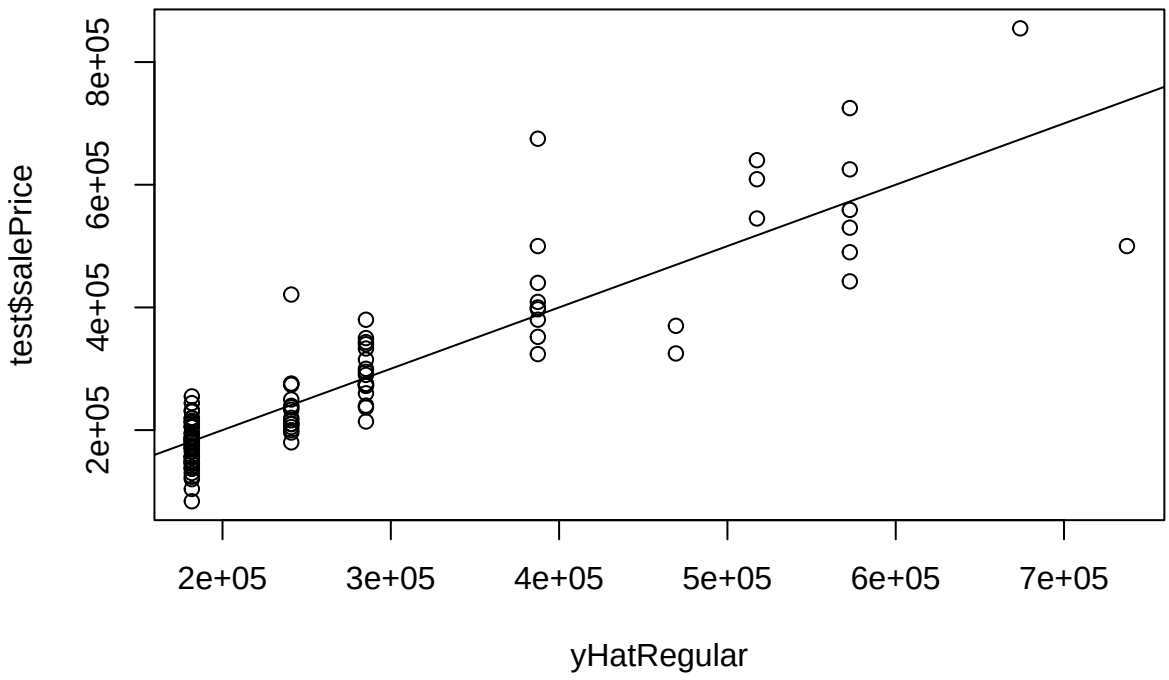



```
mseRpart <- mean((yHatRpart - test$salePrice)^2)
```

```
regularTree <- tree(salePrice ~ ., train)
yHatRegular <- predict(regularTree, test)
summary(yHatRegular)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  181770  181770  240842  275876  285221  737400
```

```
plot(yHatRegular, test$salePrice)
abline(0, 1)
text(regularTree, pretty = 0)
```



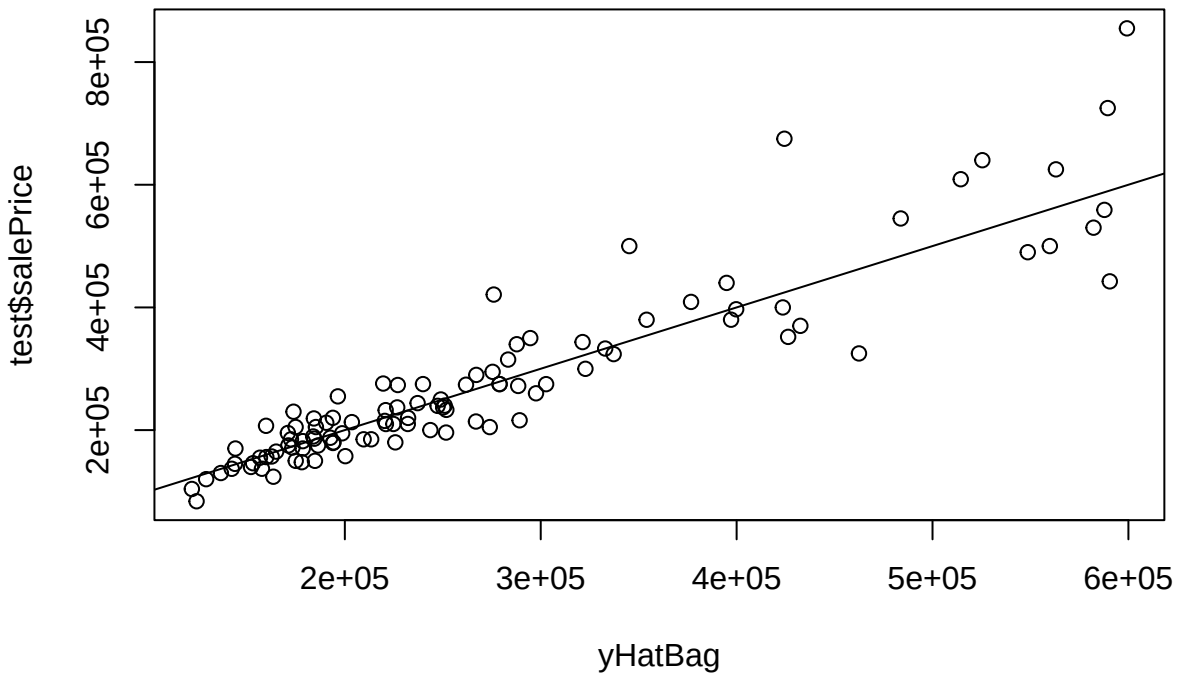

```
mseTree <- mean((yHatRegular - test$salePrice)^2)
```

Bagging

```
bag <- randomForest(salePrice ~ ., data = train, mtry = 8)
bag
```

```
##
## Call:
##  randomForest(formula = salePrice ~ ., data = train, mtry = 8)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##           Mean of squared residuals: 3194249929
##                     % Var explained: 82.66
```

```
yHatBag <- predict(bag, test)
plot(yHatBag, test$salePrice)
abline(0, 1)
```



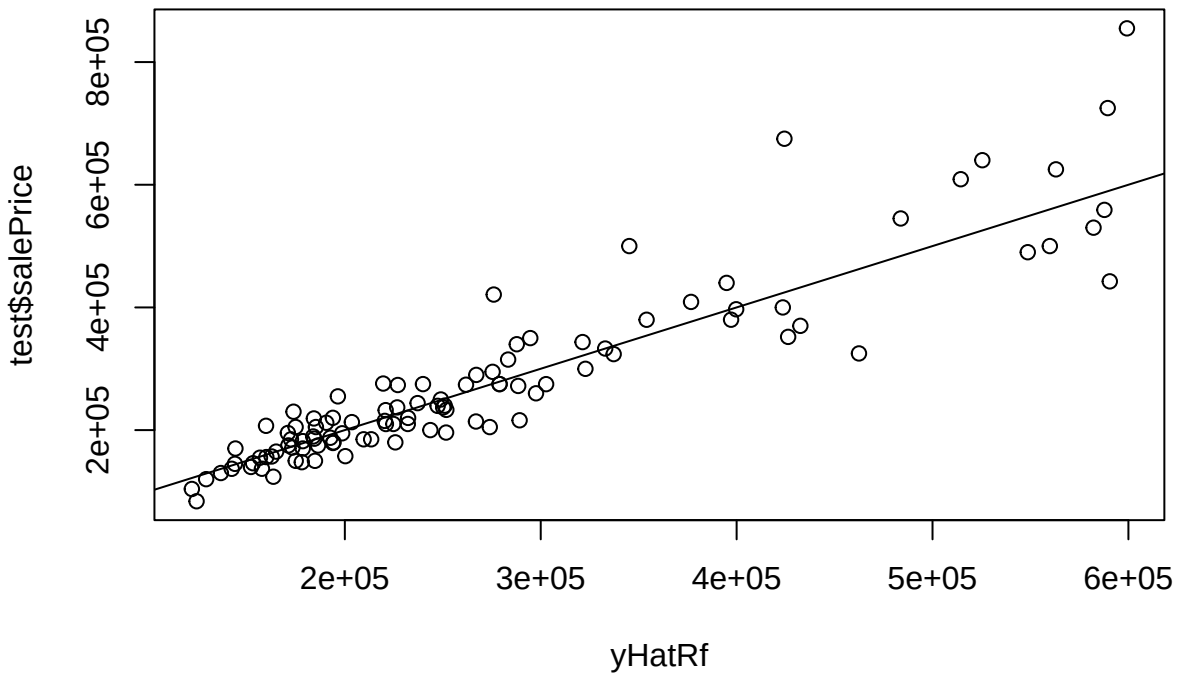

```
mseBagging <- mean((yHatBag - test$salePrice)^2)
```

Random Forests

```
rf <- randomForest(salePrice ~ ., train, mtry = 8, importance = TRUE)
rf
```

```
##
## Call:
##  randomForest(formula = salePrice ~ ., data = train, mtry = 8,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##           Mean of squared residuals: 3124118686
##                     % Var explained: 83.04
```

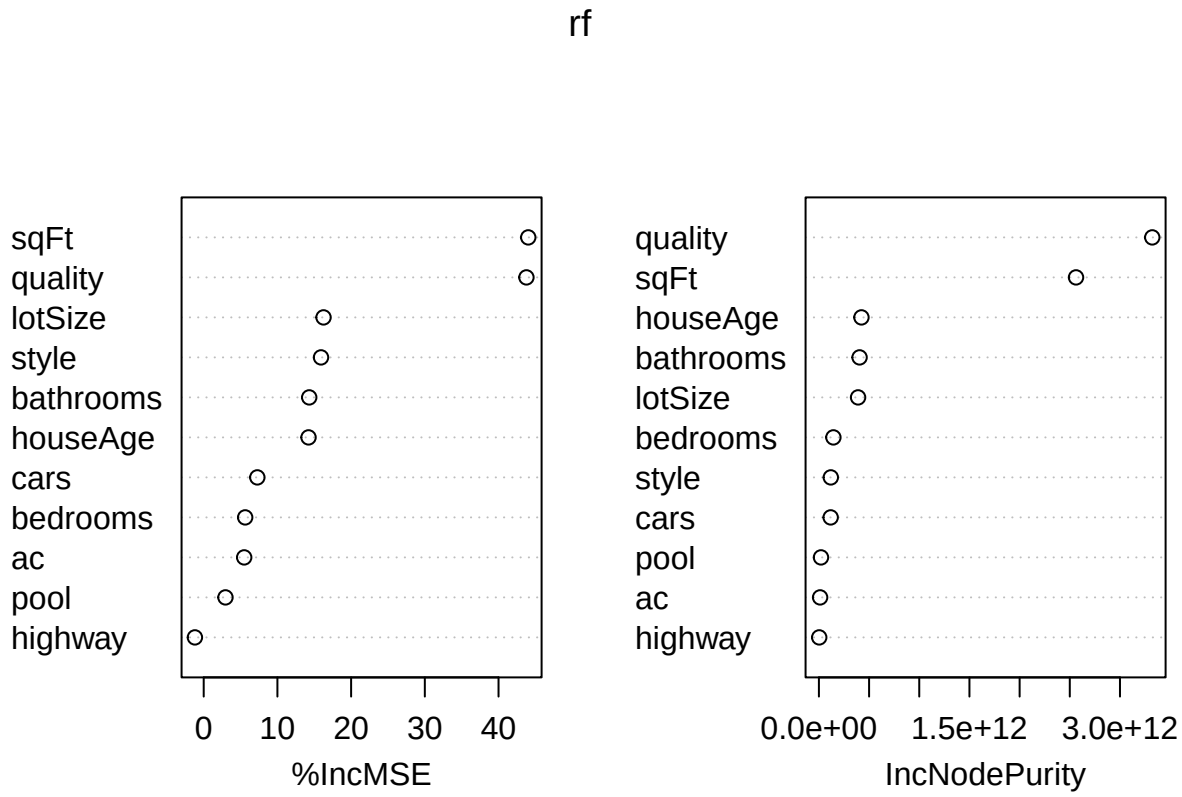
```
yHatRf <- predict(bag, test)
plot(yHatRf, test$salePrice)
abline(0, 1)
```




```
mseRf <- mean((yHatRf - test$salePrice)^2)

importance(rf)
```

```
##              %IncMSE IncNodePurity
## sqFt      44.032620  2.562103e+12
## bedrooms   5.626518  1.442115e+11
## bathrooms 14.316133  4.052837e+11
## ac         5.489482  1.189983e+10
## cars       7.274059  1.172360e+11
## pool       2.950670  2.065331e+10
## quality   43.784801  3.321760e+12
## style     15.915132  1.186574e+11
## lotSize   16.253589  3.904726e+11
## highway   -1.195246  3.168845e+09
## houseAge  14.222603  4.239163e+11
```

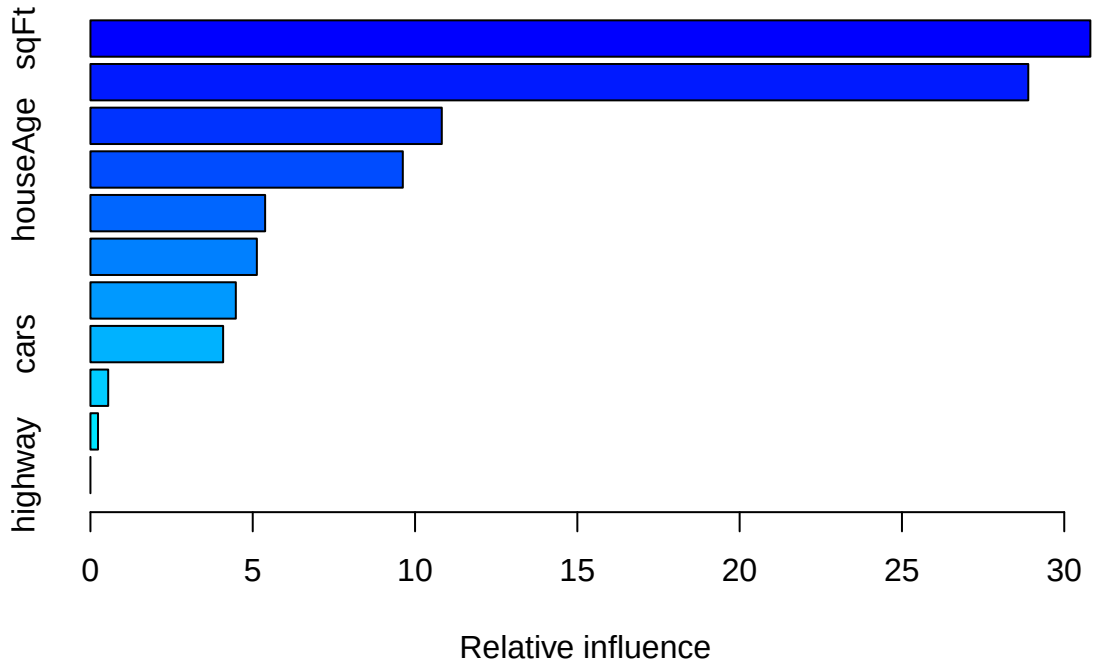
```
varImpPlot(rf)
```




```
# tunegrid <- expand.grid(mtry = c(8))
# bag2 <- train(salePrice ~ ., train, tuneGrid = tunegrid, method = "rf", importance = TRUE)
# yHatBag2 <- predict(bag2$finalModel, newdata = test)
# plot(yHatBag2, test$salePrice)
# abline(0, 1)
# mseBag2 <- mean((yHatBag2 - test$salePrice)^2)
# varImp(bag2$finalModel)
```

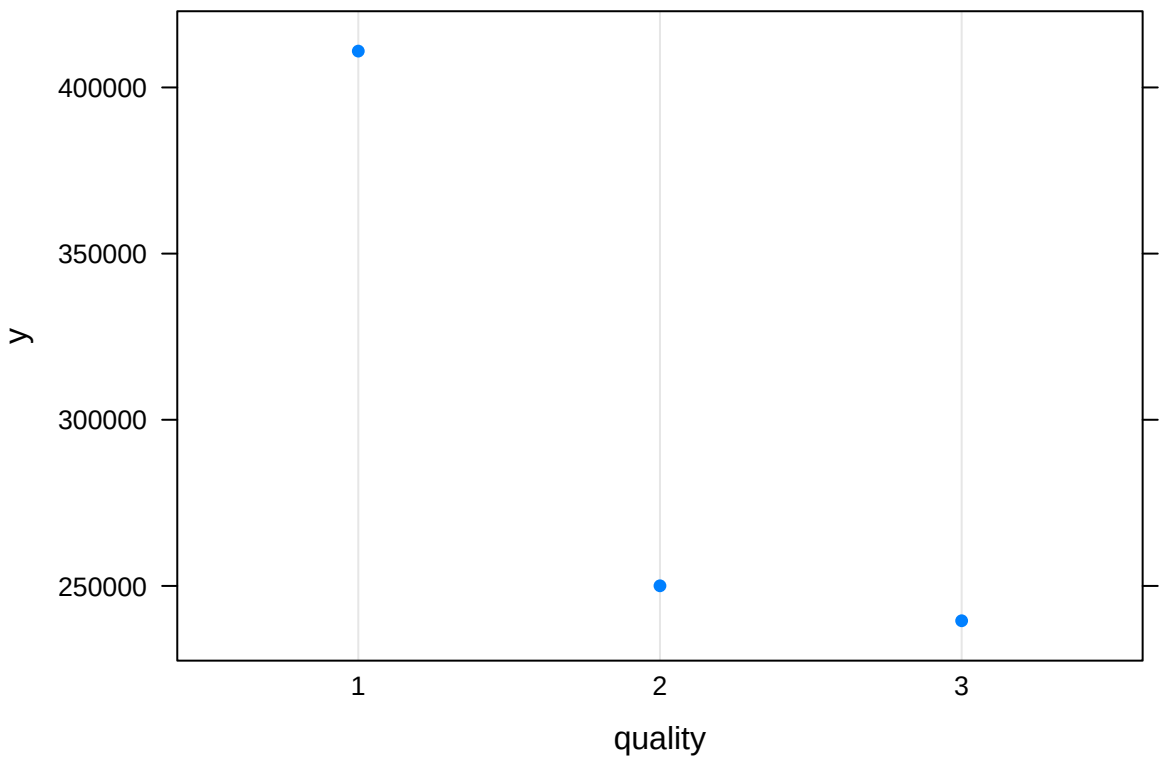
Boosting

```
dt$ac <- as.factor(dt$ac)
dt$pool <- as.factor(dt$pool)
set.seed(2022)
rowPicker <- createDataPartition(y=dt$salePrice, p=0.8, list=FALSE)
train <- dt[rowPicker]
test <- dt[-rowPicker]

boost <- gbm(salePrice ~ ., train, distribution = "gaussian", n.trees = 5000, interaction.depth = 4)
summary(boost)
```

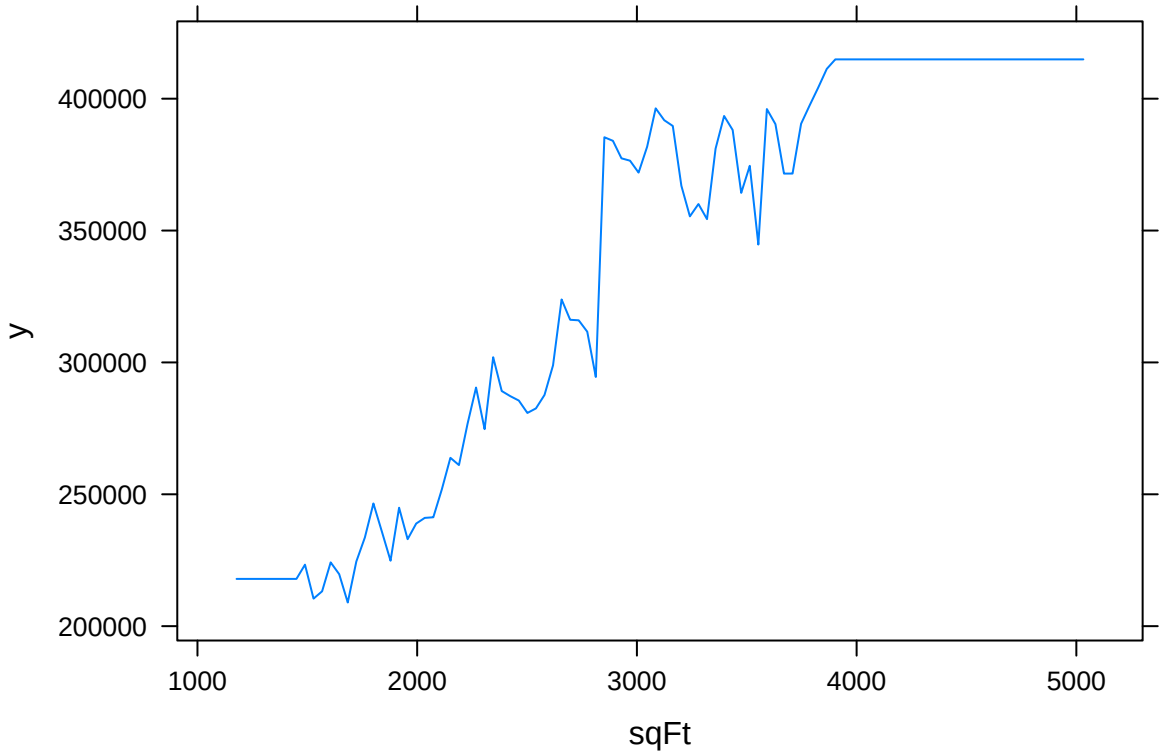



```
##                 var    rel.inf
## sqFt           sqFt 30.8041439
## quality     quality 28.8919750
## lotSize     lotSize 10.8245834
## houseAge   houseAge  9.6233434
## bathrooms bathrooms  5.3816282
## bedrooms   bedrooms  5.1266544
## style         style  4.4790674
## cars           cars  4.0881380
## pool           pool  0.5470986
## ac               ac  0.2333677
## highway     highway  0.0000000
```

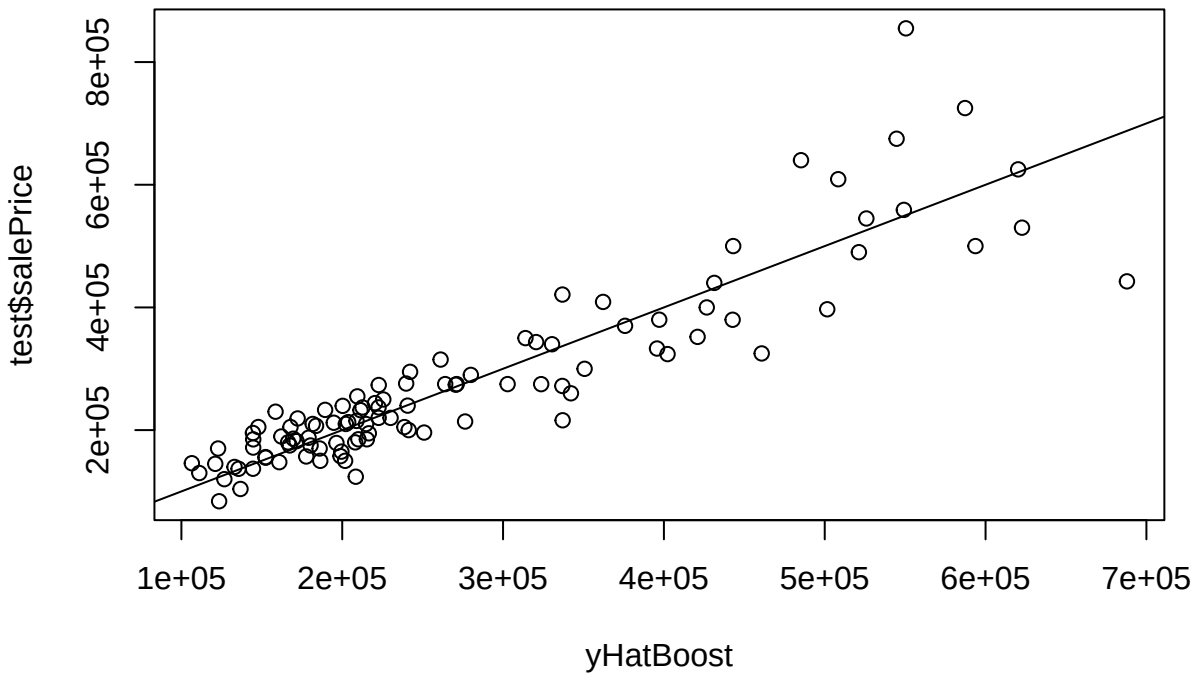
```
plot(boost, i = "quality")
```




```
plot(boost, i = "sqFt")
```




```
yHatBoost <- predict(boost, test, n.trees = 5000)
plot(yHatBoost, test$salePrice)
abline(0, 1)
```




```
mseBoost <- mean((yHatBoost - test$salePrice)^2)
```

Provide a comparison of the test MSE for the above methods.

```
mse <- c("Bagging" = mseBagging,
  "Boosting" = mseBoost,
  "Random Forest" = mseRf,
  "RPart Tree" = mseRpart,
  "Tree" = mseTree)
mse[order(mse)]
```

```
##       Bagging Random Forest      Boosting          Tree    RPart Tree
##    3416048487    3416048487    3982447471    4320450794    9047087086
```