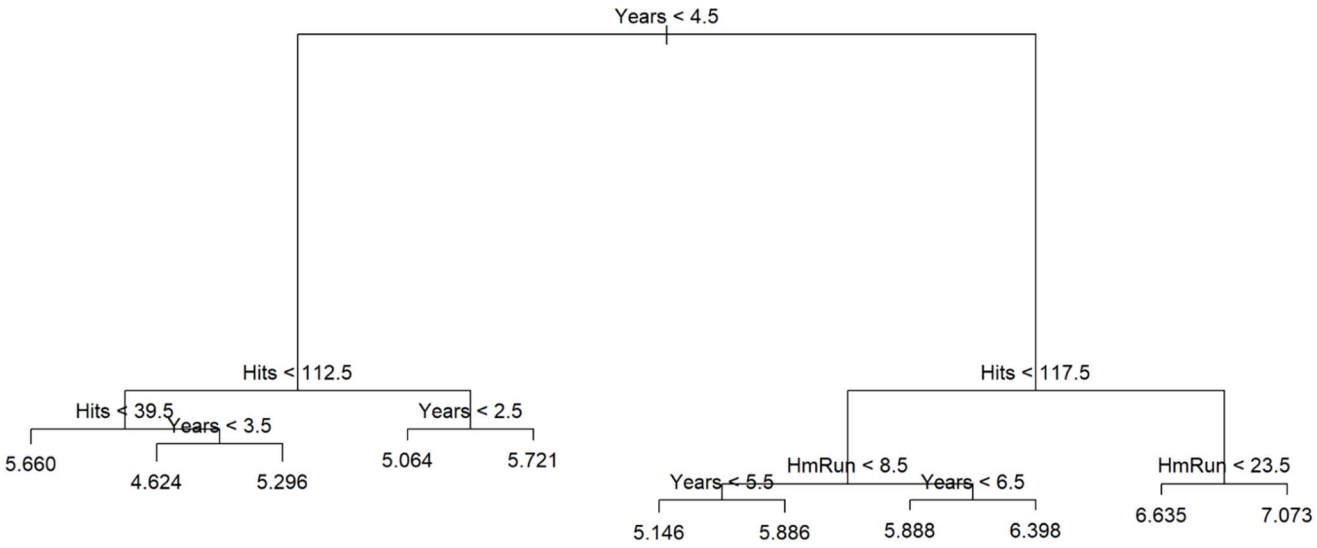
Q3: This question asks you to refer to the Hitters data we considered in Lecture 11. Separate your data into a test set (25%) and a training set (75%), then answer the following questions:

```
library(caTools)
library(caret)
library(ISLR2)
data(Hitters)
set.seed(100)
sample.data <- sample.split(Hitters, SplitRatio = 0.75)
train_data <- subset(Hitters, sample.data==T)
test_data <- subset(Hitters, sample.data==F )
```

((a) Fit a regression tree for log(salary) to the training set using the variables Years, Hits, Runs, HmRun and Errors. Plot the tree, and interpret the results. What test MSE do you obtain?

```
#(a)

library(tree)
bbsal <- subset(Hitters, !is.na(Salary), select=c("Salary","Years", "Hits", "Runs",
"HmRun", "Errors"))
bbsal$sal <- log(bbsal$Salary); bbsal$Salary <- NULL
subset_train <- bbsal[1:nrow(train_data), ]
tree_Hitters <- tree(sal ~ Years + Hits + Runs + HmRun + Errors, data=subset_train)
summary(tree_Hitters)
plot(tree_Hitters)
text(tree_Hitters, pretty = 0)
print(tree_Hitters)
```




```
Regression tree:
tree(formula = sal ~ Years + Hits + Runs + HmRun + Errors, data = subset_train)
Variables actually used in tree construction:
[1] "Years" "Hits"  "HmRun"
Number of terminal nodes:  11
Residual mean deviance:  0.247 = 57.06 / 231
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.13500 -0.28000  0.01137  0.00000  0.26630  2.00300
node), split, n, deviance, yval
      * denotes terminal node

 1) root 242 193.7000 5.906
   2) Years < 4.5 85  41.9800 5.105
     4) Hits < 112.5 55  21.8800 4.865
```

```
       8) Hits < 39.5 5   9.3320 5.660 *
       9) Hits > 39.5 50   9.0750 4.785
        18) Years < 3.5 38   3.2800 4.624 *
        19) Years > 3.5 12   1.6860 5.296 *
     5) Hits > 112.5 30  11.1000 5.546
      10) Years < 2.5 8   0.4684 5.064 *
      11) Years > 2.5 22   8.1020 5.721 *
   3) Years > 4.5 157  67.6100 6.340
     6) Hits < 117.5 83  26.3900 5.987
      12) HmRun < 8.5 44  12.5100 5.785
        24) Years < 5.5 6   1.1160 5.146 *
        25) Years > 5.5 38   8.5590 5.886 *
      13) HmRun > 8.5 39  10.0600 6.215
        26) Years < 6.5 14   2.6660 5.888 *
        27) Years > 6.5 25   5.0570 6.398 *
     7) Hits > 117.5 74  19.3100 6.736
      14) HmRun < 23.5 57  14.2800 6.635 *
      15) HmRun > 23.5 17   2.5120 7.073 *
```

```
subset_test <- bbsal[1:nrow(test_data), ]
tree_pred = predict(tree_Hitters,subset_test)
test_mse = mean((tree_pred-subset_test$sal)^2)
test_mse
[1] 0.1985394
```

The root node represents the entire data set, with a log(wage) mean of 5.906 out of 242 observations used in the training set. The first split is based on the variable "year". If "year" < 4.5, the tree branches to the left and if "year" > 4.5, it branches to the right. For the branch with year < 4.5, there are 85 observations. The average log (salary) in this group is 5.105. In the group with year < 4.5, the tree splits further according to the variable "Hits". If Hits<112.5 it moves to the left and if Hits>112.5 it moves to the right. In the subgroup Hits<112.5, there are 55 observations. The average log(wage) of this subgroup is 4.865. In the subgroup of Hits<112.5, the tree continues to split again according to the variable "Hits". If Hits<39.5 it moves to the left and if Hits>39.5 it moves to the right. For the subgroup Hits<39.5, there are 5 observations. The average log(wage) of this subgroup is 5.66. This is a terminal node. For the subgroup Hits>39.5, there are 50 observations. The average log(wage) for this subgroup is 4.785. In the subgroup of Hits>39.5, the tree is further split according to the variable "Years". If Years < 3.5, it moves to the left, if Years > 3.5, it moves to the right. For the subgroup with Year < 3.5, there are 38 observations. The mean log(salary) of this subgroup is 4.624. this is another terminal node. For the subgroup with years > 3.5, there are 12 observations. The mean log(salary) for this subgroup is 5.296. this is also a terminal node.

MSE is 0.1985394.

(b) Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

```
set.seed(666)
tree.T0 <- tree(sal ~ Years + Hits + Runs + HmRun + Errors, data=subset_train)
summary(tree.T0)
```

```
Regression tree:
tree(formula = sal ~ Years + Hits + Runs + HmRun + Errors, data = subset_train)
Variables actually used in tree construction:
[1] "Years" "Hits"  "HmRun"
Number of terminal nodes:  11
Residual mean deviance:  0.247 = 57.06 / 231
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.13500 -0.28000  0.01137  0.00000  0.26630  2.00300
```

```
prune.tree(tree.T0,best=5)
```

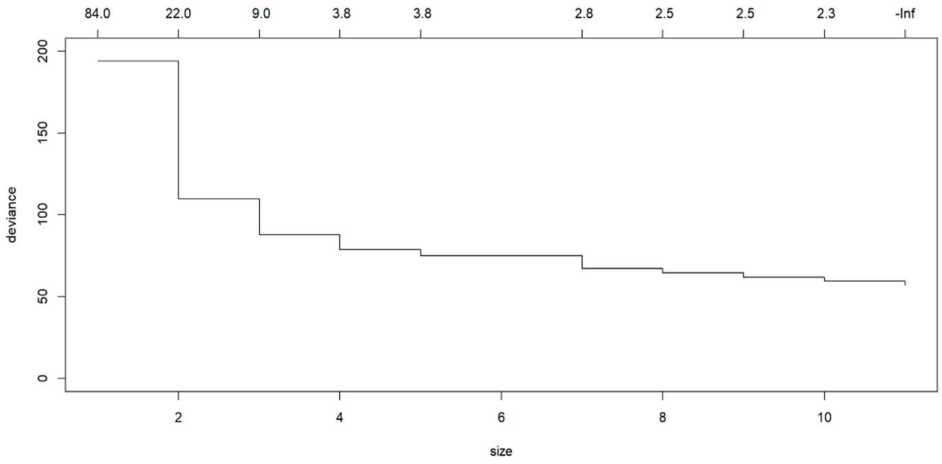
```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 242 193.70 5.906
```

```
   2) Years < 4.5 85  41.98 5.105
     4) Hits < 112.5 55  21.88 4.865 *
     5) Hits > 112.5 30  11.10 5.546 *
   3) Years > 4.5 157  67.61 6.340
     6) Hits < 117.5 83  26.39 5.987
      12) HmRun < 8.5 44  12.51 5.785 *
      13) HmRun > 8.5 39  10.06 6.215 *
     7) Hits > 117.5 74  19.31 6.736 *
```

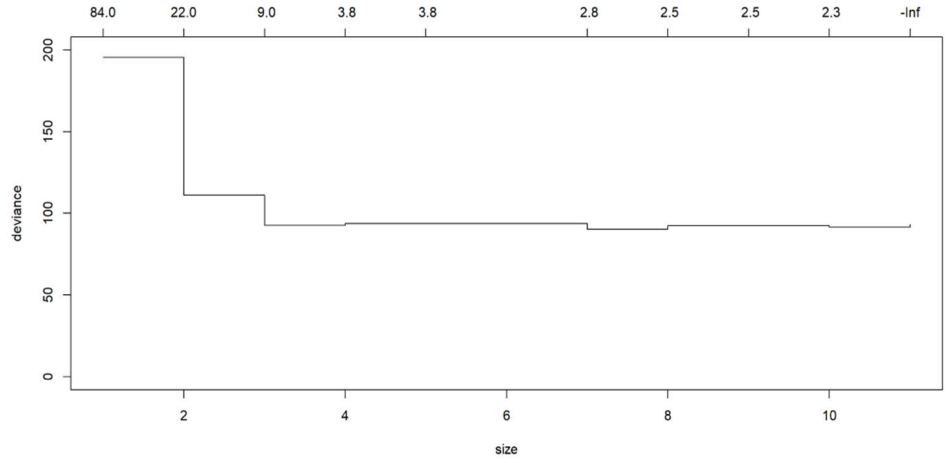
```
summary(prune.tree(tree.T0,best=5))
```

```
Regression tree:
snip.tree(tree = tree.T0, nodes = c(13L, 7L, 5L, 12L, 4L))
Variables actually used in tree construction:
[1] "Years" "Hits"  "HmRun"
Number of terminal nodes:  5
Residual mean deviance:  0.3159 = 74.86 / 237
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.2360 -0.3551 -0.0810  0.0000  0.3359  2.7980
```

```
tree.seq <- prune.tree(tree.T0)
plot(tree.seq,ylim=c(0,200))
```




```
cv.T0 <- cv.tree(tree.T0); plot(cv.T0,ylim=c(0,200))
```




```
cv.T0$size[cv.T0$dev==min(cv.T0$dev)]
```

```
[1] 7
```

```
res.tree <- prune.tree(tree.T0,best=7)
mean( (subset_test$sal - predict(res.tree,newdata=subset_test))^2 )
[1] 0.2389404

summary(res.tree)$dev/nrow(subset_train)
[1] 0.2780106
```

The optimal complexity parameter determined by cross-validation was 7.

The tested MSE of the pruned tree was 0.2389404.

The alternative measure of model fit for the pruned tree was 0.2780106.

Based on these results, we can see that pruning the tree did not significantly improve the test MSE. the MSE and bias were slightly higher for the pruned tree compared to the unpruned tree.

(c) Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important.

```
library(randomForest)
tree.bag <- randomForest(sal ~ Years + Hits + Runs + HmRun + Errors,
data=subset_train, ntree=500, mtry=2,importance = TRUE)
importance(tree.bag)
sal.pred.bag <- predict(tree.bag, subset_test)
mean((subset_test$sal - sal.pred.bag)^2)
```

```
         %IncMSE IncNodePurity
Years  80.654030      83.80740
Hits   26.321204      39.73319
Runs   19.106163      29.24741
HmRun  14.771433      20.34845
Errors  4.765347      12.29094
[1] 0.04999827
```

The MSE is 0.04999827. Years is the most important variable.

(d) Use random forests to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

```
tree.rf <- randomForest(sal ~ Years + Hits + Runs + HmRun + Errors, data=subset_train,
ntree=500, mtry=1,importance = TRUE)
importance(tree.rf)
sal.pred.rf <- predict(tree.rf, subset_test)
mean((subset_test$sal - sal.pred.rf)^2)
```

```
         %IncMSE IncNodePurity
Years  53.222246      63.86092
Hits   23.480599      37.60873
Runs   16.362128      31.51740
HmRun  13.560877      26.12134
Errors  8.499749      17.39803
[1] 0.07070855
```

```
tree.rf <- randomForest(sal ~ Years + Hits + Runs + HmRun + Errors, data=subset_train,
ntree=500, mtry=2/3,importance = TRUE)
importance(tree.rf)
sal.pred.rf <- predict(tree.rf, subset_test)
mean((subset_test$sal - sal.pred.rf)^2)
```

```
Years  53.104092      64.57615
Hits   24.095230      36.24654
Runs   14.469188      31.09787
HmRun  11.869685      26.92423
Errors  5.912929      17.15008
[1] 0.07195809
```

We can see that decreasing the value of mtry from 1 to 2/3 results in a slight increase in the test MSE, indicating a slightly higher error rate.

Q4. Separate your data into a test set (25%) and a training set (75%), then answer the following questions:

(a) Does how we measure the outcome matter? Estimate a linear model (OLS) using LIST PRICE as your outcome variable and the following as covariates: NBHD, YEAR, SQ FT, BEDS, BATHS, BUILT YEAR. Compute a second OLS regression using PPSF as your outcome variable and omitting SQ FT as a covariate. What do you make of the negative coefficient on BEDS when the price is denominated in square feet? Is anything else noteworthy when comparing the two sets of results?

```
setwd('C:\\Users\\yuerl\\Desktop')
set.seed(666)
load("fmls.rda")
sample.data <- sample.split(fmls, SplitRatio = 0.75)
train_data <- subset(fmls, sample.data==T)
test_data <- subset(fmls, sample.data==F )
model_list_price <- lm(`LIST_PRICE` ~ NBHD + YEAR + `SQ_FT` + BEDS + BATHS +
`BUILT_YEAR`, data = train_data)
summary(model_list_price)

model_ppsf <- lm(PPSF ~ NBHD + YEAR + BEDS + BATHS + `BUILT_YEAR`, data = train_data)
summary(model_ppsf)


Call:
lm(formula = LIST_PRICE ~ NBHD + YEAR + SQ_FT + BEDS + BATHS +
    BUILT_YEAR, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-773378  -38703    -651   36134  845249

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.654e+07  4.297e+06   3.850 0.000123 ***
NBHDCOLUMBIA HEIGHTS -9.165e+04  7.261e+03 -12.622  < 2e-16 ***
NBHDKALORAMA         -1.928e+04  6.941e+03  -2.778 0.005550 **
NBHDADAMS MORGAN     -6.835e+04  8.161e+03  -8.375  < 2e-16 ***
NBHDLOGAN CIRCLE     -2.327e+03  1.085e+04  -0.215 0.830184
NBHDMOUNT PLEASANT   -5.394e+04  8.870e+03  -6.080 1.55e-09 ***
NBHDU STREET         -5.546e+04  9.282e+03  -5.975 2.92e-09 ***
YEAR                 -8.550e+03  2.139e+03  -3.997 6.76e-05 ***
SQ_FT                 4.060e+02  1.008e+01  40.266  < 2e-16 ***
BEDS                  4.214e+03  5.442e+03   0.774 0.438899
BATHS                 5.066e+04  6.577e+03   7.702 2.53e-14 ***
BUILT_YEAR            3.409e+02  6.755e+01   5.047 5.08e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 83830 on 1387 degrees of freedom
Multiple R-squared:  0.8044, Adjusted R-squared:  0.8028
F-statistic: 518.5 on 11 and 1387 DF,  p-value: < 2.2e-16


Call:
lm(formula = PPSF ~ NBHD + YEAR + BEDS + BATHS + BUILT_YEAR,
    data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-277.979  -48.256    2.433   45.738  309.254

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.008e+04  3.846e+03   2.620 0.008895 **
NBHDCOLUMBIA HEIGHTS -9.497e+01  6.496e+00 -14.620  < 2e-16 ***
NBHDKALORAMA         -3.475e+01  6.215e+00  -5.592 2.71e-08 ***
NBHDADAMS MORGAN     -8.928e+01  7.307e+00 -12.220  < 2e-16 ***
NBHDLOGAN CIRCLE     -2.902e+01  9.603e+00  -3.022 0.002556 **
NBHDMOUNT PLEASANT   -6.786e+01  7.930e+00  -8.557  < 2e-16 ***
NBHDU STREET         -6.999e+01  8.276e+00  -8.457  < 2e-16 ***
```

```
YEAR                -5.036e+00  1.915e+00  -2.630 0.008642 **
BEDS                -1.976e+01  4.145e+00  -4.766 2.07e-06 ***
BATHS                2.127e+01  5.469e+00   3.889 0.000105 ***
BUILT_YEAR           2.976e-01  6.049e-02   4.919 9.72e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 75.07 on 1388 degrees of freedom
Multiple R-squared:  0.1974, Adjusted R-squared:  0.1916
F-statistic: 34.14 on 10 and 1388 DF,  p-value: < 2.2e-16
```

The coefficient of BEDS is positive (4.214e+03) in the model where the outcome variable is LIST_PRICE (in USD). This indicates that an increase in the number of bedrooms leads to an increase in the list price of the property, holding all other variables constant. However, in the model where the outcome variable is PPSF (price per square foot), the coefficient of BEDS is negative (-1.976e+01). This suggests that an increase in the number of bedrooms is associated with a decrease in price per square foot when the price is measured in units per square foot. However, it could mean that properties with more bedrooms tend to lower the average square footage per room, which leads to a decrease in price per square foot.

The R-squared is also a noteworthy point. In the model with LIST_PRICE as the outcome, the R-squared value is 0.8044. On the other hand, in the model with PPSF as the outcome, the R-squared value is 0.1974. This indicates that the model with LIST_PRICE as the outcome has a better overall fit compared to the model with PPSF as the outcome.

(b) Do some listing agents get better prices? Estimate a linear model using LIST PRICE as outcome, but adding AGENT NAME as a covariate. How many columns does this add to the design matrix? What happens to the regression R2? Is this a better model? Use what you have learned in class to argue either way. Would you choose a listing agent for your home based on this analysis?

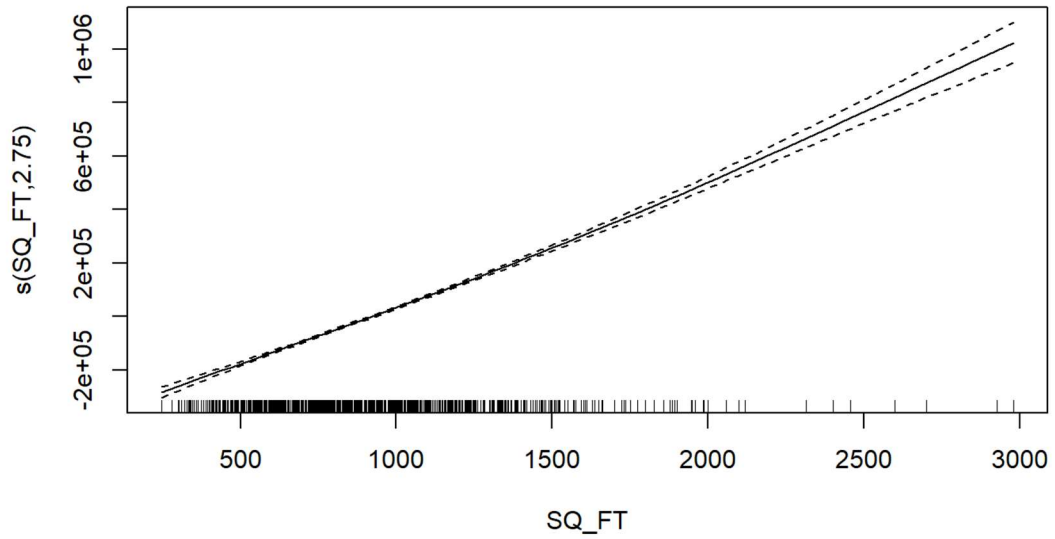
```
agent_dummies <- model.matrix(~ AGENT_NAME - 1, data = train_data)
design_matrix <- cbind(train_data[, c("LIST_PRICE", "NBHD", "YEAR", "SQ_FT", "BEDS",
"BATHS", "BUILT_YEAR")], agent_dummies)
model_agent <- lm(LIST_PRICE ~ ., data = design_matrix)
R2_agent <- summary(model_agent)$r.squared
R2_no_agent <- summary(model_list_price)$r.squared
R2_difference <- R2_agent - R2_no_agent
R2_difference
```

```
[1] 1099
[1] 0.1149534
```

Therefore, we can conclude that 1099 columns added. And due to the R2_difference = 0.1149534, which indicates that the model with the AGENT NAME covariate explains an additional 11.49% of the variance in the list prices compared to the model without the AGENT NAME covariate. Based on the increase in R-squared, it can be argued that the model with AGENT NAME covariates is better than the model without AGENT NAME covariates. According to the analysis, which shows that some listed proxies may be associated with better prices, I would choose a agent without considering other factors.

(c) Are old homes less expensive? Estimate an additive model using LIST PRICE as outcome as in (a), but using splines on SQ FT and BUILT YEAR. Plot the partial effects for SQ FT and BUILT YEAR (you may need to set scale=0 due to the large effect of square footage). Is the response different than what you found using OLS in part (a)?

```
library(mgcv)
model_additive <- gam(LIST_PRICE ~ s(SQ_FT) + s(BUILT_YEAR), data = train_data)
plot(model_additive, select = 1:2, scale = 0)
```



We can see that old homes shows to be less expensive.

(d) Estimate the first model from part (a) using LASSO, choosing the model flexibility, λ by cross validation.

```
library(glmnet)
x <- model.matrix(LIST_PRICE ~ NBHD + YEAR + SQ_FT + BEDS + BATHS + BUILT_YEAR,
alpha=1, data = train_data)
y <- train_data$LIST_PRICE
lasso_model <- cv.glmnet(x, y, alpha=1, nfolds = 10)
opt_lambda <- lasso_model$lambda.min
lasso_fit <- glmnet(X, y, lambda = opt_lambda)
print(coef(lasso_fit))
```

```
7 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) 21376487.5284
NBHD           -4192.8867
YEAR          -10895.3373
SQ_FT            418.9476
BEDS            -131.2643
BATHS          43954.6659
BUILT_YEAR       264.6288
```

(e) Estimate the first model from part (a) using regression forests.

```
library(randomForest)
X <- train_data[, c("NBHD", "YEAR", "SQ_FT", "BEDS", "BATHS", "BUILT_YEAR")]
y <- train_data$LIST_PRICE
rf_model <- randomForest(x = X, y = y, ntree = 500)
print(rf_model)
test_X <- test_data[, c("NBHD", "YEAR", "SQ_FT", "BEDS", "BATHS", "BUILT_YEAR")]
predictions <- predict(rf_model, newdata = test_X)
actual_prices <- test_data$LIST_PRICE
mse <- mean((actual_prices - predictions)^2)
mse_d <- mse
```

```
Call:
 randomForest(x = X, y = y, ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 6236031836
                    % Var explained: 82.49
[1] 3745261098
```

(f) Compare the test MSE from the regressions in parts (a), (c), (d) and (e). Which performed best?

```
mse_a <- mean((test_data$LIST_PRICE - predict(model_list_price, newdata =
test_data))^2)
mse_c <- mean((test_data$LIST_PRICE - predict(model_agent, newdata =
design_matrix))^2)
mse_d <- mean((test_data$LIST_PRICE - predictions)^2)
mse_list <- c(mse_a, mse_c, mse_d, mse_e)
regressions <- c("OLS", "OLS with Agent Name", "Random Forest", "Lasso")
comparison <- data.frame(Regression = regressions, Test_MSE = mse_list)
comparison
```

Regression <chr>	Test_MSE <dbl>
OLS	4901044758
OLS with Agent Name	51202888435
Random Forest	3745261098
Lasso	3745261098

4 rows

Random forest shows to be the best.