Linear Regression vs Random Forests on the Boston housing dataset

```r
set.seed(1)
train <- sample(1:nrow(Boston), nrow(Boston) * 0.8)
boston.test <- Boston[-train, "medv"]
boston.train <- Boston[train, "medv"]
```

Linear Regression

```r
lm.boston <- lm(medv~., data=Boston, subset = train)
```

Train error:

```r
yhat.lm_train <- predict(lm.boston, newdata = Boston[train,])
mean((yhat.lm_train - boston.train)^2)
```

```
## [1] 22.42843
```

Test error:

```r
yhat.lm <- predict(lm.boston, newdata = Boston[-train,])
mean((yhat.lm - boston.test)^2)
```

```
## [1] 21.40796
```

Random Forests

Let's train a random forest:

```r
rf.boston <- randomForest(medv ~ ., data = Boston, subset = train, mtry=6, ntree=200)
```

Now test the prediction accuracy on the test set:

```r
yhat.rf <- predict(rf.boston, newdata =  Boston[-train,])
mean((yhat.rf - boston.test)^2)
```

```
## [1] 8.110461
```

Random Forests deliver much more accurate results right away with the test MSE of 7.9 vs 21.4 for a Linear Regression. The likely reason is the relationship between features and the label is non-linear. Random Forests accomodate the non-linearity, while Linear Regression doesn't.

We can address this issue with the Linear Regression with additional feature engineering by adding non-linear terms.

Adding Non-Linear Terms to the Boston dataset

TO-DO