Lab 7: When a guest arrives they will count how many sides it has on

Claire Jellison

11/8/2019

```
library(randomForest)
```

```
## randomForest 4.6-14
```

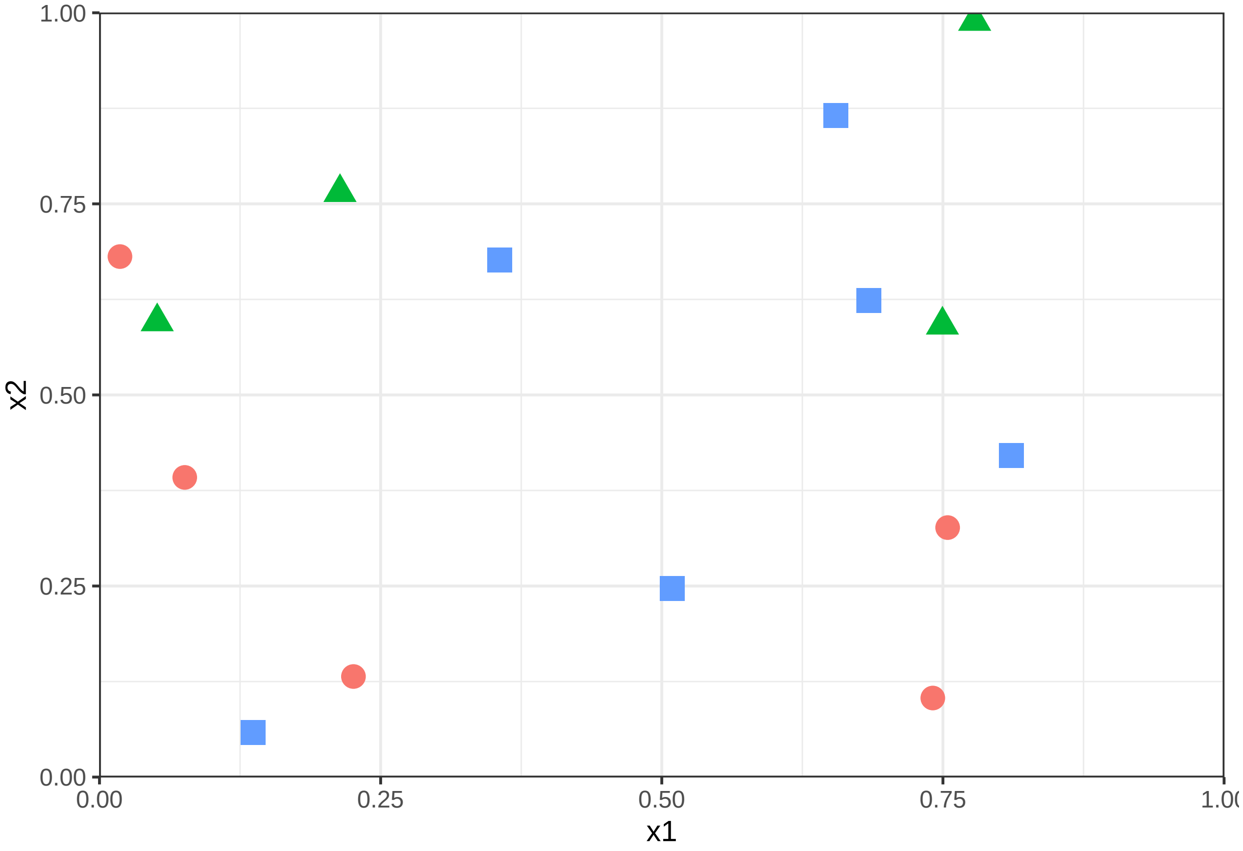
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(75)
n <- 16
x1 <- runif(n)
x2 <- runif(n)
group <- as.factor(sample(1:3, n, replace = TRUE))
levels(group) <- c("circle", "triangle", "square")
df <- data.frame(x1, x2, group)
df[1, 2] <- .765 # tweaks to make a more interesting configuration
df[9, 1] <- .741
df <- df[-7, ]
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

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

```
ggplot(df, aes(x = x1, y = x2, col = group, shape = group)) +
  geom_point(size = 4) +
  scale_x_continuous(expand = c(0, 0) , limits = c(0, 1)) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1)) +
  scale_color_discrete(guide = FALSE) +
  scale_shape_discrete(guide = FALSE) +
  theme_bw()
```



1. Growing the full classification tree

Use the trees package in R to fit a full unpruned tree to this data set, making splits based on the Gini index. You can find the code to do this in the slides from week 8 or in the lab at the end of Chapter 8 in the book. Please plot the resulting tree.

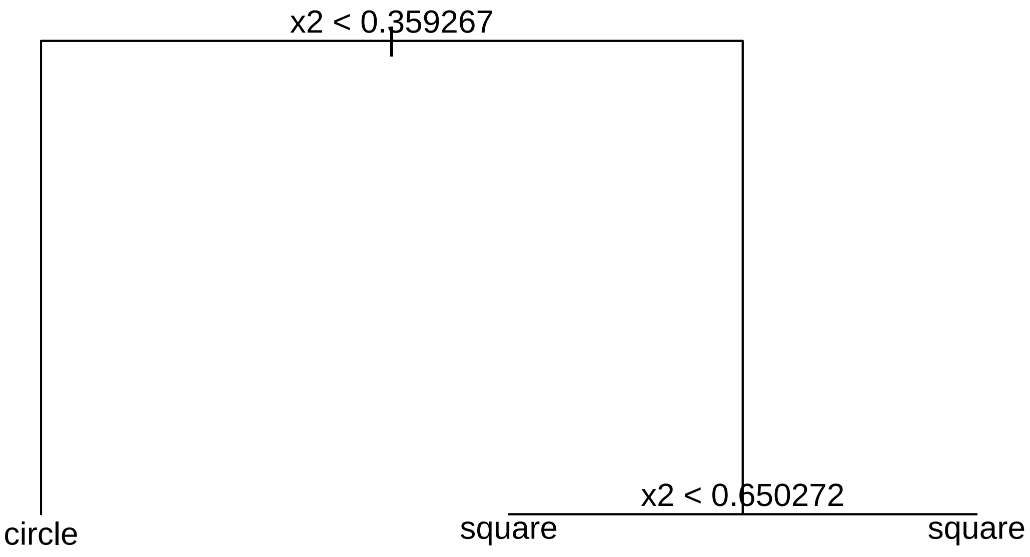
```
summary(df)
```

```
##        x1                x2                group
##  Min.   :0.01819   Min.   :0.05825   circle  :5
##  1st Qu.:0.17521   1st Qu.:0.28666   triangle:4
##  Median :0.50911   Median :0.59154   square  :6
##  Mean   :0.45061   Mean   :0.49790
##  3rd Qu.:0.74527   3rd Qu.:0.67890
##  Max.   :0.81100   Max.   :0.98897
```

```
library(tree)
t1 <- tree(group ~ x1 + x2,
           data = df, split = "gini")
class(t1)
```

```
## [1] "tree"
```

```
plot(t1)
text(t1, pretty = 0)
```



```r
set.seed(39)
```

1) The two most common splits that we saw in class were a horizontal split around $X_2 \approx 0.50$ and a vertical split around $X_1 \approx 0.30$. Was either of these the first split decided upon by your classification tree?

No, the tree did not use either of those as the first split. Instead the first split was $X_12 \approx 0.36$.

2) What is the benefit of the second split in the tree?

I see no benefit to the second split in the tree, since it predicts the same thing either way.

3) Which class would this model predict for the new observation with $X_1 = 0.21, X_2 = 0.56$?

This model uses none of the infomation from X_1 to classify so it would just depend on X_2. Given that $X_2 = 0.56 > 0.36$, it will predict a square.

2. An alternate metric

Now refit the tree based on the deviance as the splitting criterion (you set this as an argument to the tree() function). The deviance is defined for the classification setting as:

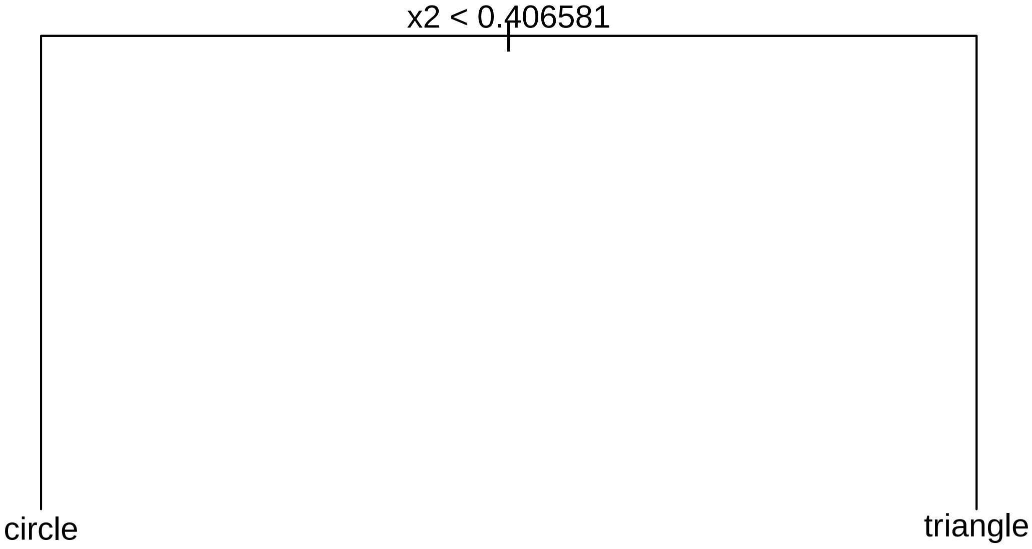
$$-2\sum_m \sum_k n_{mk} \log \hat{p}_{mk}$$

Plot the resulting tree. Why does this tree differ from the tree fit based on the Gini Index?

```r
t1 <- tree(group ~ x1 + x2,
           data = df, split = "deviance")
class(t1)
```

```
## [1] "tree"
```

```r
plot(t1)
text(t1, pretty = 0)
```



```
set.seed(39)
```

Yes, this tree differs quite a lot from the previous tree. The previous tree using gini for splitting criteria, only predicted circles or squares and had three leaves. This tree has two leaves and predicts circles and triangles. The split is also a higher value and close to what many groups did in class. The split is different because it is trying to minimize different criteria.

3. Growing a pruned regression tree

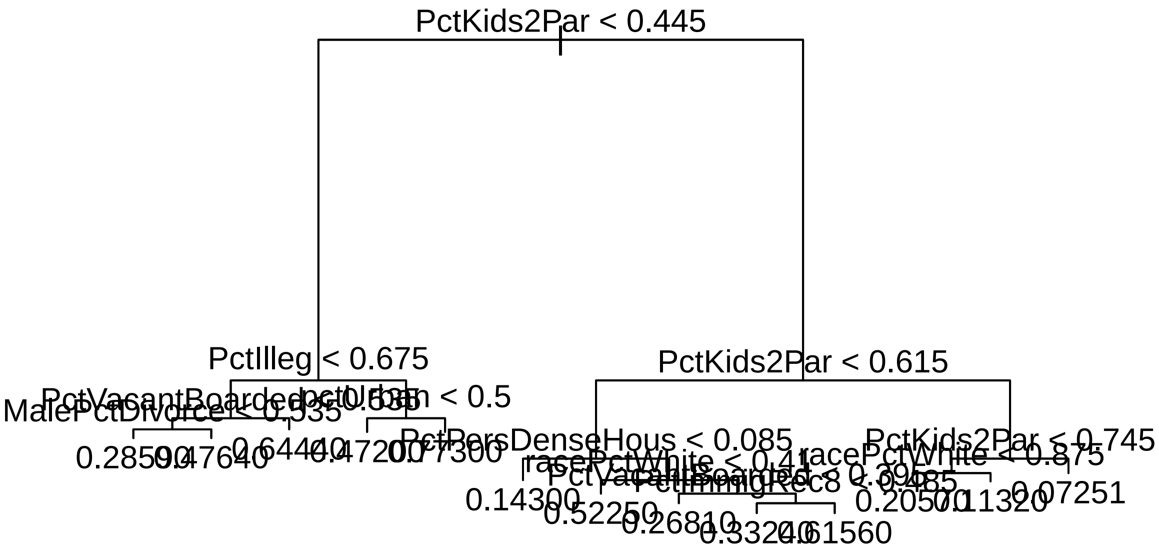
Crime and Communities, revisited

In Lab 3, you fit a regression model to a training data set that predicted the crime rate in a community as a function of properties of that community.

Fit a regression tree to the training data using the default splitting criteria (here, the deviance is essentially the RSS). Next, perform cost-complexity pruning and generate a plot showing the relationship between tree size and deviance to demonstrate the size of the best tree. Finally, construct the tree diagram for this best tree.

```
dcrime <- read.csv("http://andrewpbray.github.io/data/crime-train.csv")
treecrime <- tree(ViolentCrimesPerPop ~ racePctWhite
          + pctUrban
          + PctEmploy
          + MalePctDivorce
          + PctKids2Par
          + PctWorkMom
          + PctPersDenseHous
          + NumStreet
          + PctVacantBoarded
          + PctImmigRec8
          + PctIlleg
          + PctHousOccup
          + population
          + householdsize
          + medIncome,
           data = dcrime)
```

```r
plot(treecrime)
text(treecrime, pretty = 0)
```

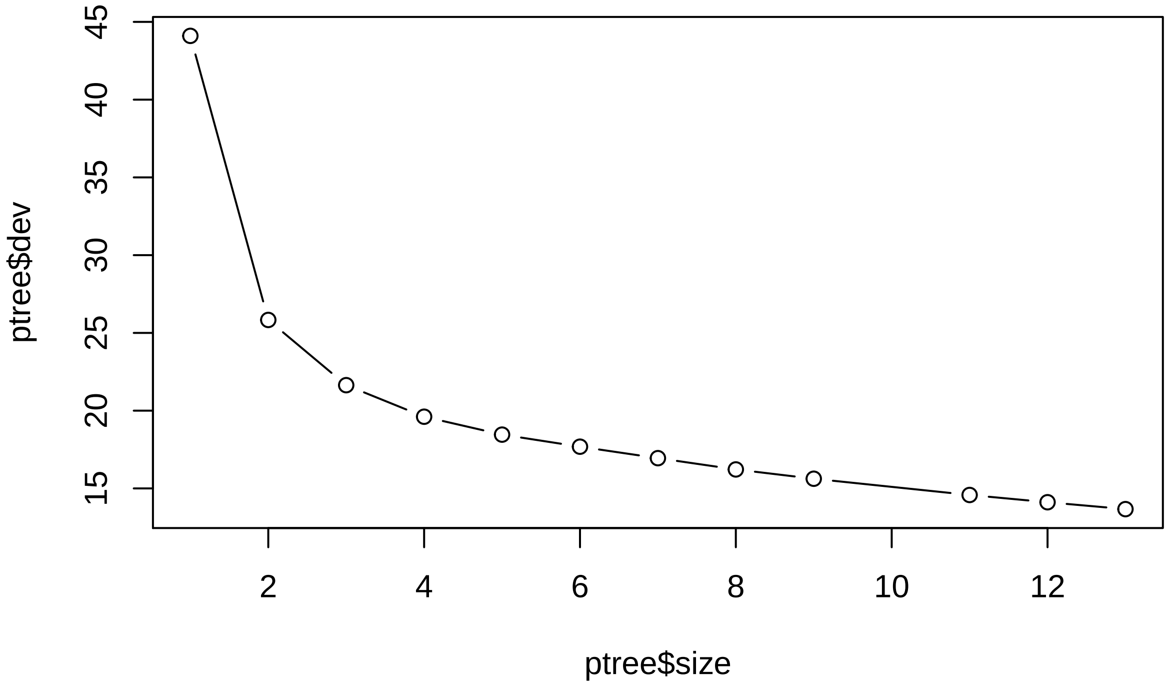


```r
set.seed(39)
```

```r
set.seed(39)
ptree <- prune.tree(treecrime, k = NULL, best = NULL, dcrime,
           method = c("deviance"), eps = 1e-3)
ptree
```

```
## $size
##  [1] 13 12 11  9  8  7  6  5  4  3  2  1
## 
## $dev
##  [1] 13.66915 14.11136 14.57863 15.62308 16.22079 16.94590 17.68391
##  [8] 18.45941 19.61170 21.63909 25.83329 44.10050
## 
## $k
##  [1]       -Inf  0.4422092  0.4672679  0.5222261  0.5977077  0.7251080
##  [7]  0.7380074  0.7755031  1.1522867  2.0273988  4.1941945 18.2672161
## 
## $method
## [1] "deviance"
## 
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```r
plot(ptree$size, ptree$dev, type = "b")
```

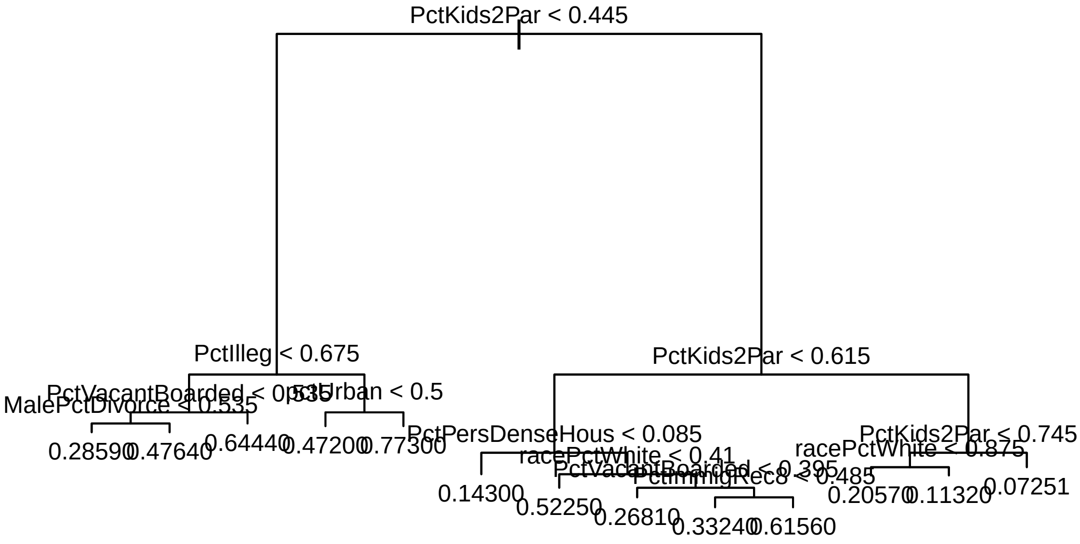


Looks like the tree with size 13 was the best at reducing the deviance.

```
ptree$size[which.min(ptree$dev)]
```

```
## [1] 13
```

```
ptreebest <- prune.tree(treecrime, k = NULL, best =13, dcrime,
           method = c("deviance"), eps = 1e-3)
plot(ptreebest)
text(ptreebest, pretty = 20, cex = .75, offset = 20)
```



4. Comparing predictive performance

Use this tree to compute the MSE for the test data set. How does it compare to the test MSE for your regression model? You can load the test data with the following code:

```
test_data <- read.csv("https://bit.ly/2PYS8Ap")

MSE <- function(model, data){
  n <- nrow(data)
  ys <- data$ViolentCrimesPerPop
  y_hats <- predict(model, data)
  residuals <- y_hats - ys
  MSE <- sum(residuals^2)/n
  MSE
}

MSEtree <- MSE(ptreebest, test_data)
MSEtree
```

```
## [1] 0.01708644
```

```
regmodel <- lm(data = dcrime, ViolentCrimesPerPop ~
            racePctWhite
          + pctUrban
          + PctEmploy
          + MalePctDivorce
          + PctKids2Par
          + PctWorkMom
          + PctPersDenseHous
          + NumStreet
          + PctVacantBoarded
          + PctImmigRec8
          + PctIlleg
          + PctHousOccup
          + population
          + householdsize
          + medIncome)

MSEreg <- MSE(regmodel, test_data)
MSEreg
```

```
## [1] 0.0182602
```

In this case the MSE for the tree model is better than the MSE for the regression model. This could be due to the fact that the functional form of our data is not well approximated by a linear model. However, the two are not that far apart.

5. Growing a random forest

We now apply methods to decrease the variance of our estimates. Fit a randomForest() model that performs only bagging and no actual random forests (recall that bagging is the special case of random forests with $m = p$). Next, fit a second random forest model that uses $m = p/3$. Compute their test MSE's. Is this an improvement over the vanilla pruned regression tree? Does it beat your regression model?

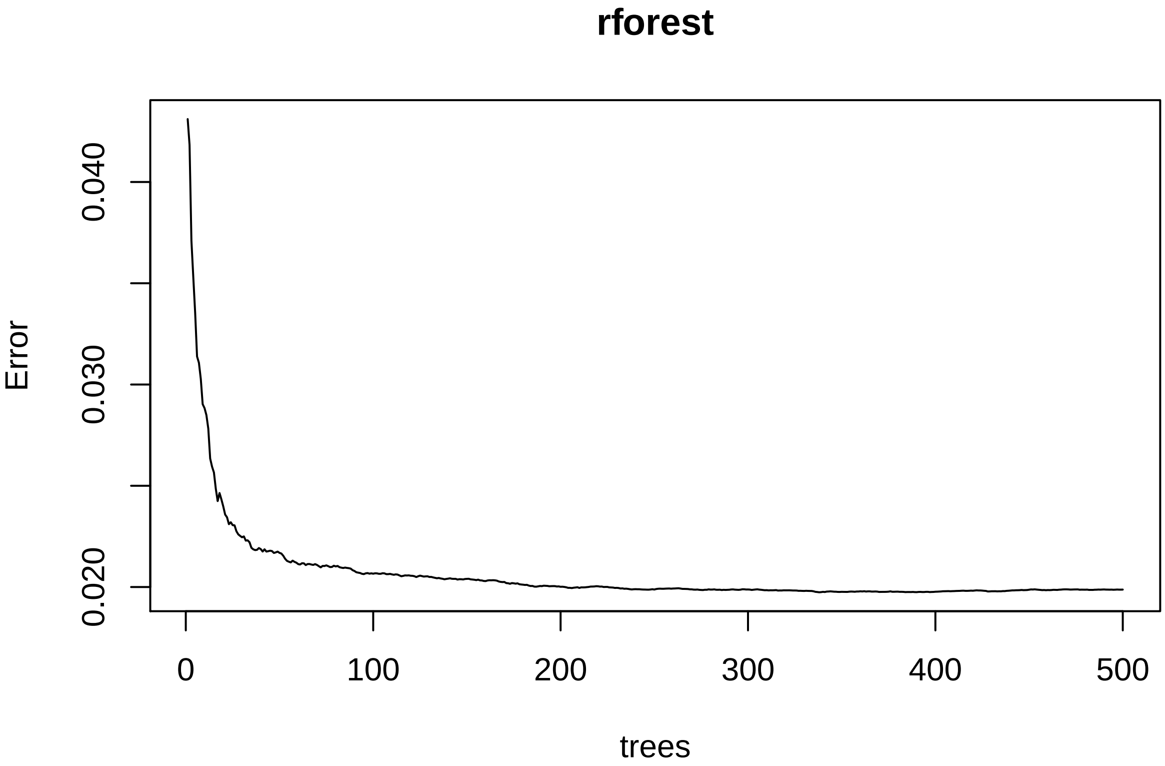
```
rforest <- randomForest(ViolentCrimesPerPop ~ racePctWhite
          + pctUrban
          + PctEmploy
          + MalePctDivorce
          + PctKids2Par
          + PctWorkMom
```

```
           + PctPersDenseHous
           + NumStreet
           + PctVacantBoarded
           + PctImmigRec8
           + PctIlleg
           + PctHousOccup
           + population
           + householdsize
           + medIncome,
            data = dcrime, mtry = 15)
```

```
rforest
```

```
##
## Call:
##  randomForest(formula = ViolentCrimesPerPop ~ racePctWhite + pctUrban +      PctEmploy + MalePctDivo
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 15
##
##           Mean of squared residuals: 0.01987385
##                     % Var explained: 63.95
```

```
plot(rforest)
```



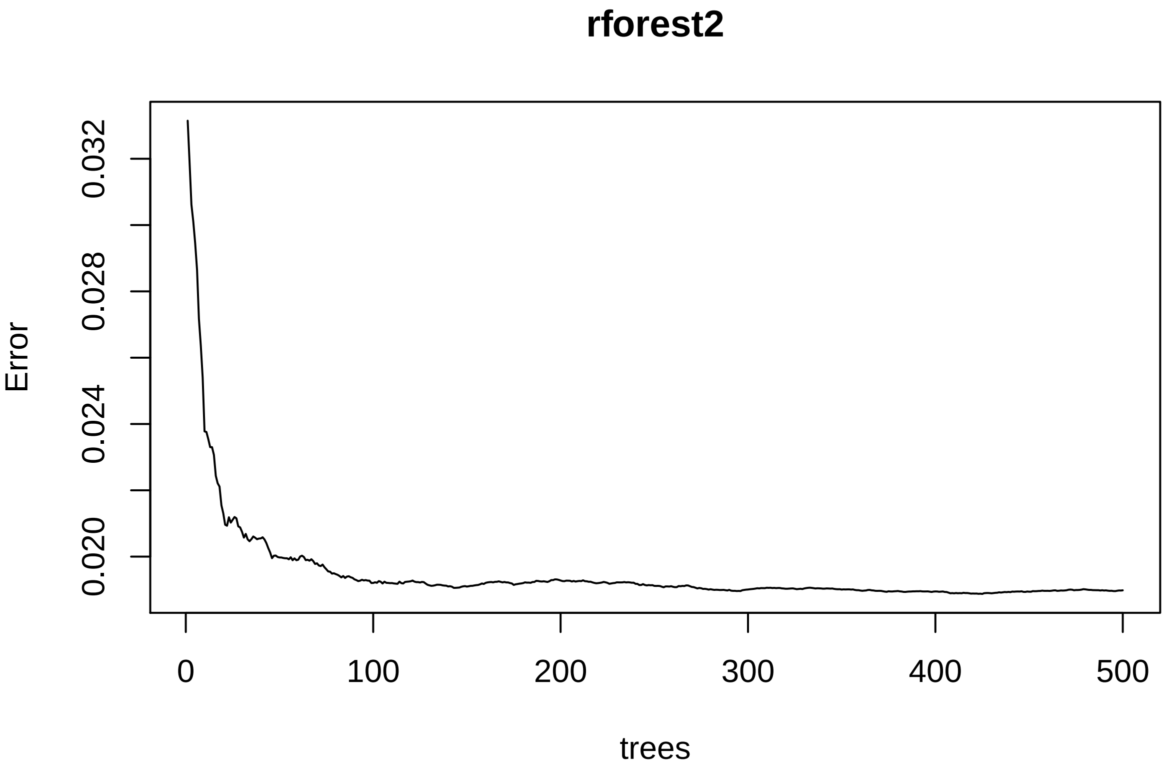
```
rforest2 <- randomForest(ViolentCrimesPerPop ~ racePctWhite
           + pctUrban
           + PctEmploy
           + MalePctDivorce
           + PctKids2Par
           + PctWorkMom
           + PctPersDenseHous
           + NumStreet
           + PctVacantBoarded
           + PctImmigRec8
           + PctIlleg
           + PctHousOccup
           + population
           + householdsize
           + medIncome,
            data = dcrime)
```

Since originally we had 15 predictors in the model, the random forest is considering 5 at each split.

```
rforest2
```

```
##
## Call:
##  randomForest(formula = ViolentCrimesPerPop ~ racePctWhite + pctUrban +      PctEmploy + MalePctDivo
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 0.01898293
##                     % Var explained: 65.56
```

```
plot(rforest2)
```



```
MSE(rforest, test_data)
```

```
## [1] 0.003356975
```

```
MSE(rforest2, test_data)
```

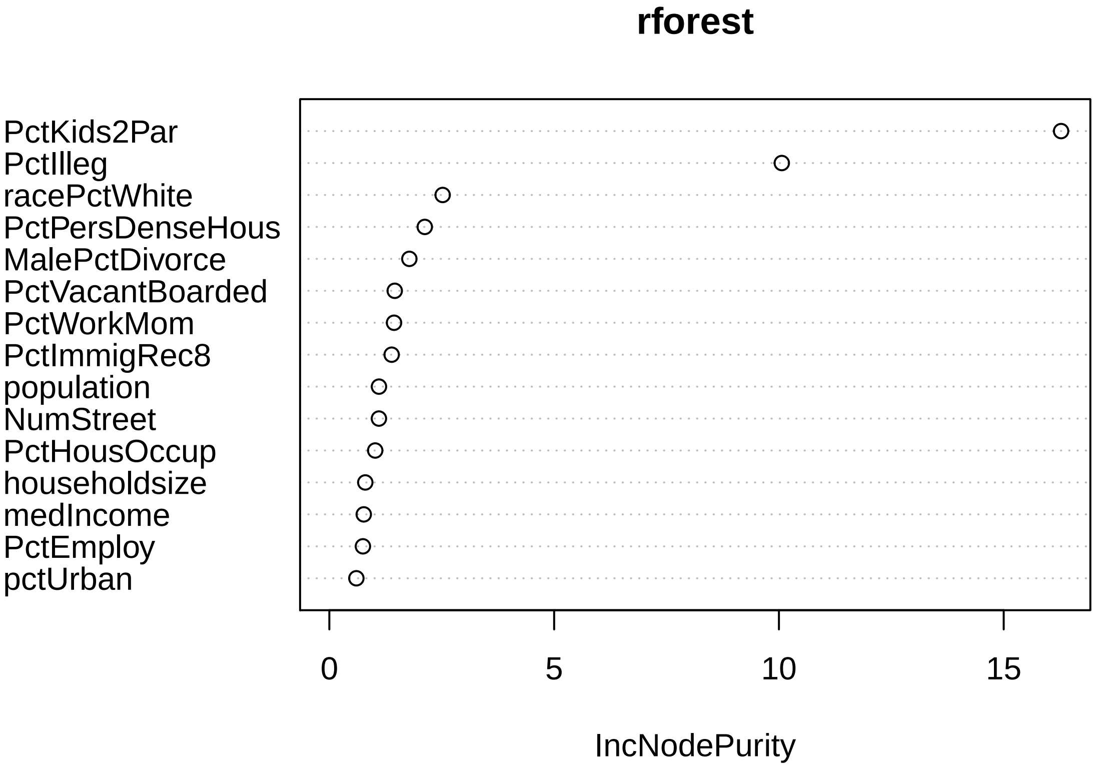
```
## [1] 0.003577114
```

The test MSE is way better than with the regression model or the pruned tree. It appears that the test MSE for the random forest where m = p is very slightly lower, perhaps if I had included more of the predictors that would not be the case.

6. Variance importance

One thing we lose by using these computational techniques to limit the variance is the clearly interpretable tree diagram. We can still salvage some interpretability by considering importance(). Please construct a Variable Importance Plot (varImpPlot()). Are these restults similar/different from your interpretation of your regression coefficients in Lab 3?

```
varImpPlot(rforest)
```



```
regm <- lm( ViolentCrimesPerPop ~ racePctWhite
           + pctUrban
           + PctEmploy
           + MalePctDivorce
           + PctKids2Par
           + PctWorkMom
           + PctPersDenseHous
           + NumStreet
           + PctVacantBoarded
           + PctImmigRec8
           + PctIlleg
           + PctHousOccup
           + population
```

```
          + householdsize
          + medIncome,
           data = dcrime)
summary(regm)
```

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ racePctWhite + pctUrban +
##     PctEmploy + MalePctDivorce + PctKids2Par + PctWorkMom + PctPersDenseHous +
##     NumStreet + PctVacantBoarded + PctImmigRec8 + PctIlleg +
##     PctHousOccup + population + householdsize + medIncome, data = dcrime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54500 -0.07471 -0.01576  0.04907  0.73504
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.26154    0.08965   2.918 0.003629 **
## racePctWhite     -0.16273    0.04377  -3.718 0.000215 ***
## pctUrban          0.04485    0.01283   3.495 0.000500 ***
## PctEmploy        -0.08406    0.05232  -1.607 0.108550
## MalePctDivorce    0.25233    0.05356   4.711 2.91e-06 ***
## PctKids2Par      -0.14220    0.10232  -1.390 0.164965
## PctWorkMom       -0.02854    0.03846  -0.742 0.458309
## PctPersDenseHous  0.11222    0.04359   2.574 0.010222 *
## NumStreet         0.17870    0.05002   3.573 0.000375 ***
## PctVacantBoarded  0.07257    0.02911   2.493 0.012875 *
## PctImmigRec8      0.02089    0.02742   0.762 0.446429
## PctIlleg          0.29130    0.06219   4.684 3.32e-06 ***
## PctHousOccup     -0.07475    0.02986  -2.504 0.012493 *
## population        0.01025    0.04797   0.214 0.830822
## householdsize     0.01219    0.05068   0.241 0.809935
## medIncome         0.06077    0.04539   1.339 0.181063
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1365 on 784 degrees of freedom
## Multiple R-squared:  0.6688, Adjusted R-squared:  0.6624
## F-statistic: 105.5 on 15 and 784 DF,  p-value: < 2.2e-16
```

The two appear similar however there are some notable differences. The variables racePctWhite and Pct Illeg are the second and third most important variables in the random forest and similarly are very significant with relatively large coefficients in the regression linear model. The variable PctKids2Par shows up as the most important variable but lacks significance despite having a largeish coefficient in the linear model. Overall, the two clearly correspond to some degree.

Two Cultures

1)What are the two cultures outlined by Breiman?

One culture assumes that data are generated by some stochastic data model (this includes linear and logistic regression type models). The other doesn't make that kind of assumption about knowing the true form, and rather uses algorithmic models.

2)Is the Ozone Project supervised or unsupervised? Classification or Regression? Which methods that we've seen could be used to tackle this problem?

It was supervised and used regression analysis. Perhaps, binary trees would work better for that data or some form of discriminant analysis.

3)What is the name of the model/method that is discussed in equation (R) of section 5.1?

That is the assumed functional form for linear regression.

4)In section 5.4 he states, "If the model has too many parameters, then it may overfit the data and give a biased estimate of accuracy". Where would this model be in terms of the bias-variance tradeoff?

This would lean towards high variance and low bias since the model is overly complex.

5)What is the Rashoman effect? Did you run into this effect is question 5 from the last lab?

It is possible to have a large number of models with different parameters that yield about the same RSS. This could also arise with trees when the data used is slightly different.

6)Explain how one of the techniques that we've covered could be seen to invoke Occam's Razor.

Linear regression is very easy to interpret, however it might not be the best predictor depending on the structure of the data.

7)The most illuminating point for me in this paper was…

"For a data model, this translates as: fit the parameters in your model by using the data, then, using the model, predict the data and see how good the prediction is." I hadn't thought about it like that before as going backward through the model mechanism to generate the data.

8)The most confusing point for me in this paper was…

I thought the section on Bellman and the curse of dimensionality was somewhat confusing.

9)Which of the responses (Cox, Efron, Hoadley, Parzen) do you find the most incisive? Why?

I liked Hoadley's response about how algorithmic models can be highly contextual and only have really good predictive ability on a sample of the data. It seems to me that algorithmic models may be less desirable when data collection was not carried out well (small or biased sample), since no functional form about how the variables should cooperate is assumed.

10)Which do you think is the strongest single criticism of Breiman's paper that is levelled by the commentators?

Cox seems to have the greatest disagreement with the paper and I like his point about how understanding the approximating relationship between variables is important when you want to know about a somewhat related question but have to make use of existing data sources.

11)The big ticket question: in your area of study, if you had to use methods from only one of Breiman's cultures for the rest of your life, which would it be: Data Model or Algorithmic Model?

I'd probably choose the data model because it seems much more likely I'll use it in the future, although I do agree that algorithmic models are probably generally better since they don't seem as subject to faulty assumptions of form.