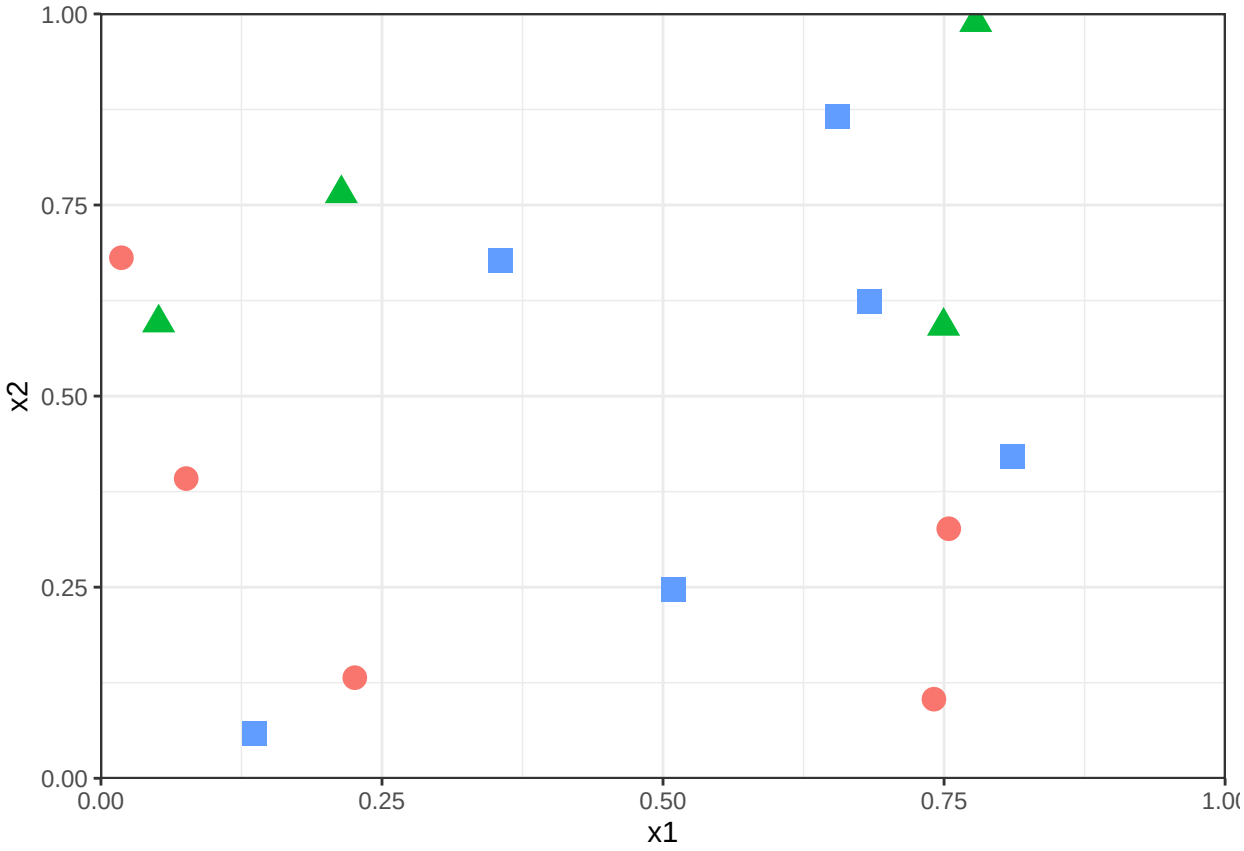
Lab 7: When a guest arrives they will count how many sides it has on

Paul Nguyen

11/7/2019

```
library(tree)
library(dplyr)
library(randomForest)
library(ggplot2)
```

```
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()
```

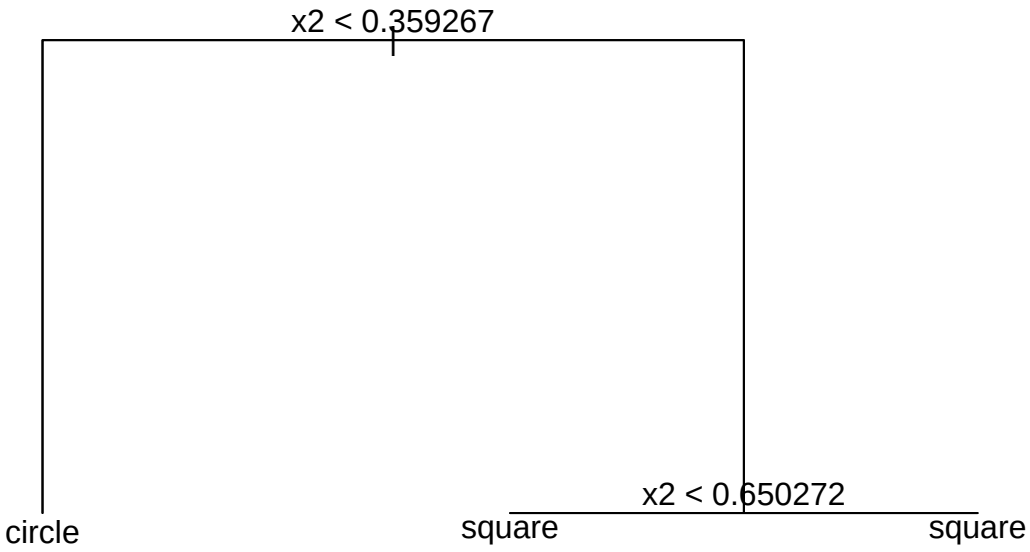



```
treeshape <- tree(group ~ x1 + x2, data = df, split = "gini")
summary(treeshape)
```

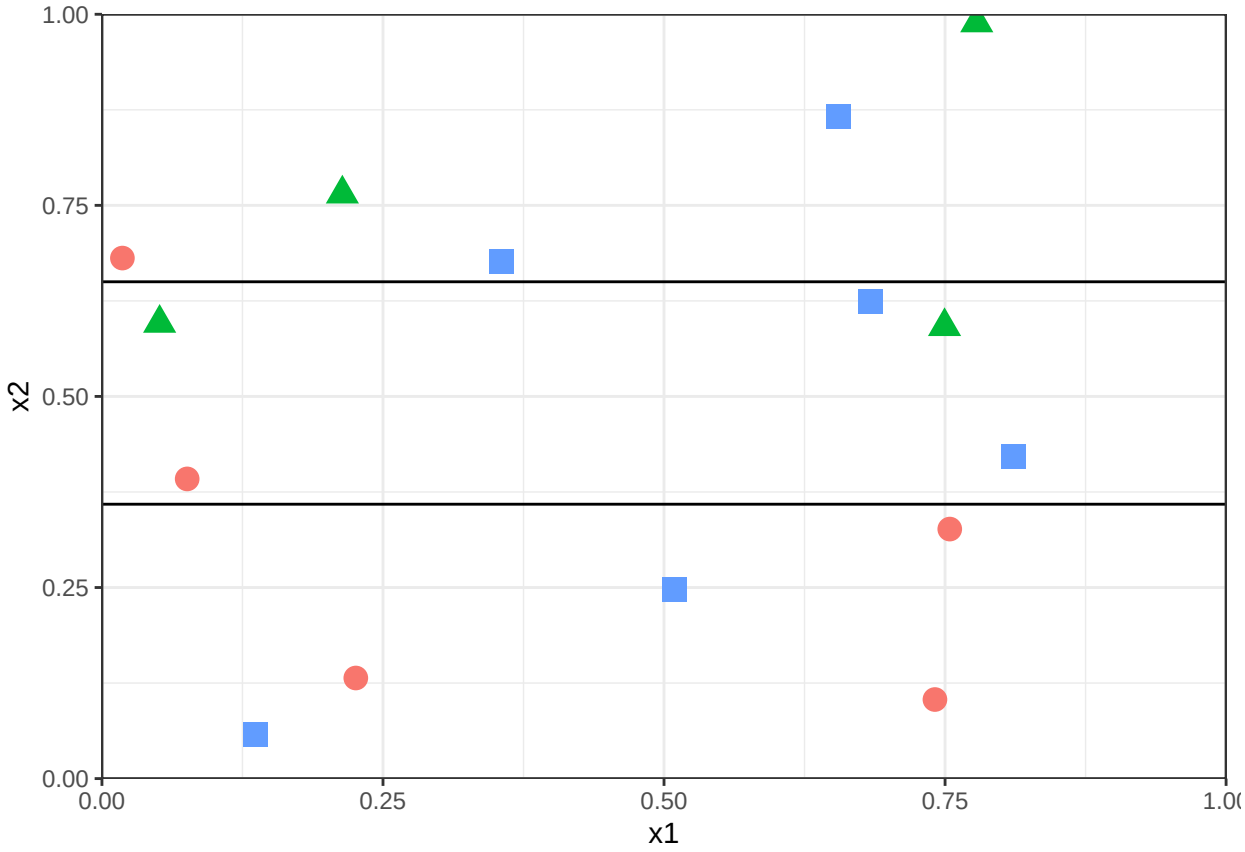
```
##
## Classification tree:
## tree(formula = group ~ x1 + x2, data = df, split = "gini")
```

```
## Variables actually used in tree construction:
## [1] "x2"
## Number of terminal nodes:  3
## Residual mean deviance:  2.319 = 27.83 / 12
## Misclassification error rate: 0.5333 = 8 / 15
```

```
plot(treeshape)
text(treeshape, pretty = 1)
```




```
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() +
  geom_hline(yintercept = .359) +
  geom_hline(yintercept = .65)
```

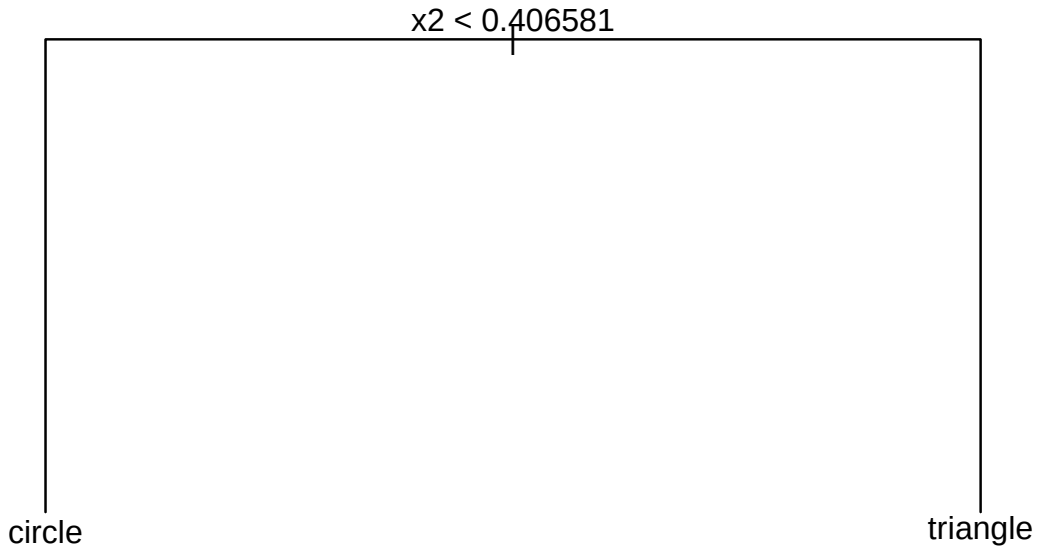


a) The split decided upon my classification tree was: if x2 $< .359$, predict circle, and if not, then predict square; this is not one of the guesses we would have made in class

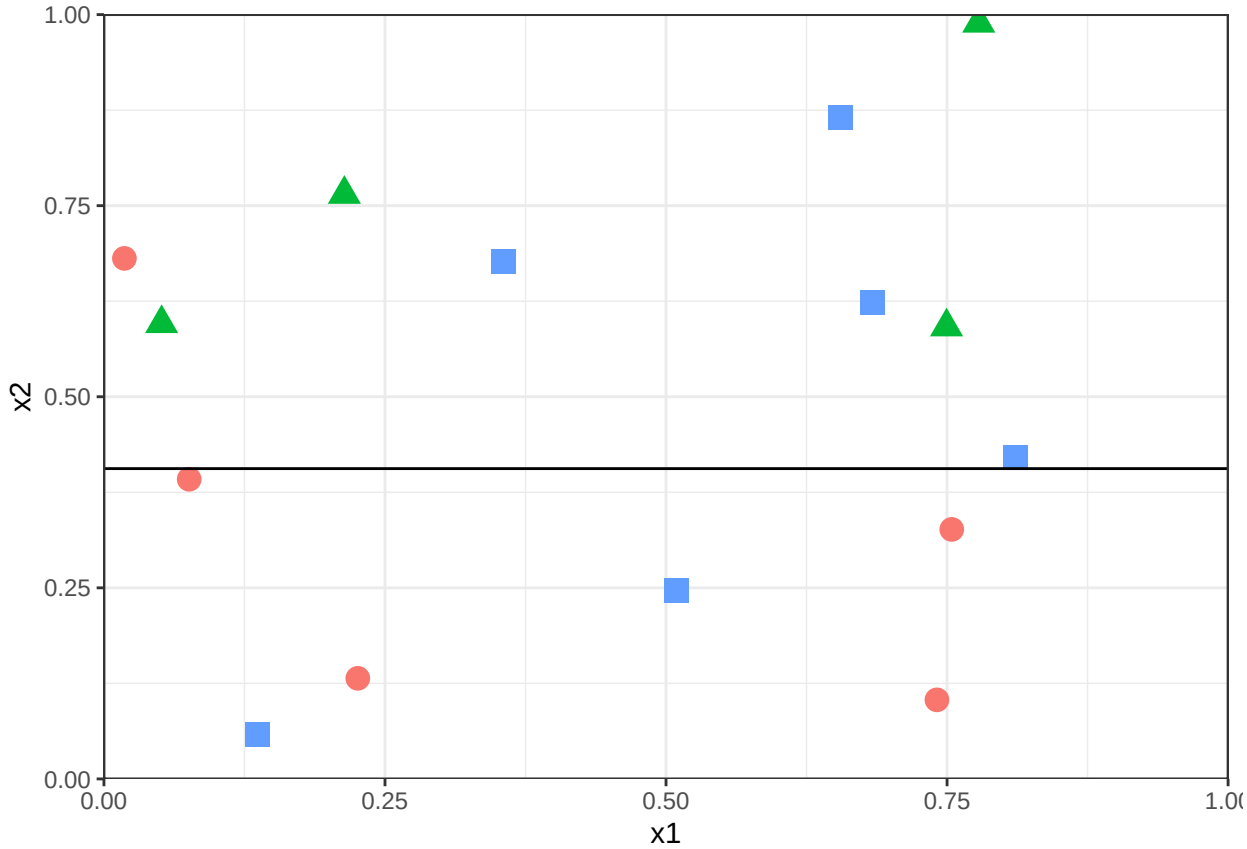
b) The reason why the tree splits at this second node is because of GINI's insistence of increasing nodal purity. While the model predicts a square in both of the regions, in the region with increased nodal purity, it is more certain that the object is a square, and in the other, a wee bit less certain. (note, I'm not that there should be a split here.. if you compare the GINI of this tree ($\frac{3}{5} * \frac{2}{5} + \frac{2}{5} * \frac{3}{5} + \frac{2}{5} * \frac{3}{5}$) vs $\frac{3}{5} * \frac{2}{5} + \frac{4}{10} * \frac{6}{10}$)

c) $(X_1 = 0.21, X_2 = 0.56)$ my model would predict that this new observation is a square

```
treeshapedeviance <- tree(group ~ . - group, data = df, split = "deviance")
plot(treeshapedeviance)
text(treeshapedeviance, pretty = 2)
```




```
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() +
  geom_hline(yintercept = .406)
```



This “deviance” tree is different than the “Gini” tree because i think that while both value nodal purity, if we

decreased the node so that the boundary was drawn at x2 = .359, the cross entropy value, which is similar to the deviance, would increase from $-4/6 \log 4/6 - 4/9 \log 4/9 = .63$ to $-3/5 \log 3/5 - 4/10 \log 4/10 = .67$

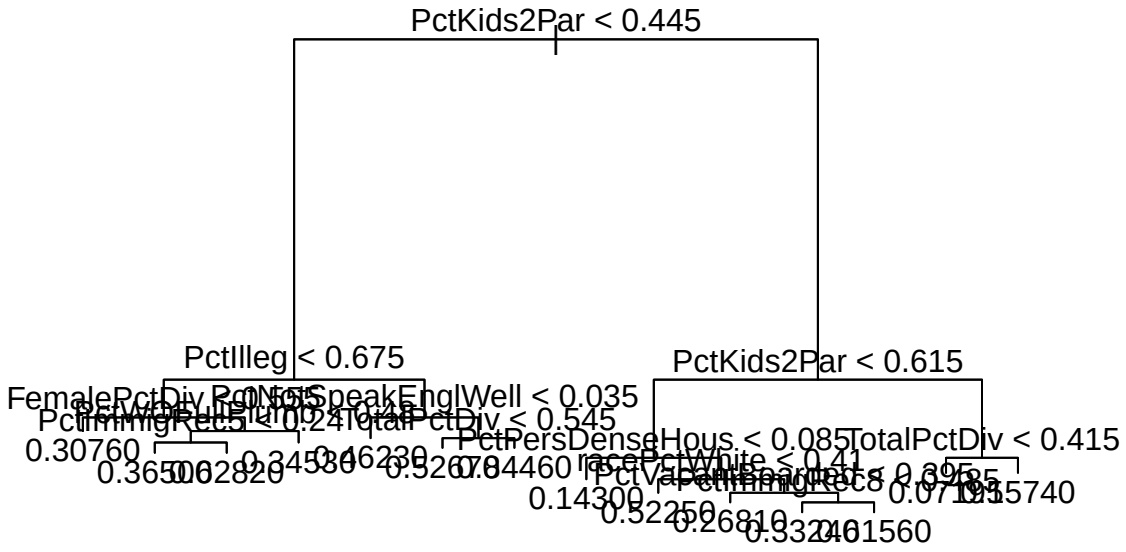
Entropy is less extreme than GINI; it would prefer being right most of the time than being really right on one side of the node and then more wrong on the other.

I think that deviance is more sensitive to "outlier" regions, as it multiplies the sum of the proportion*logged proportion by 2, so it penalizes regions with low nodal purity to a greater extent. $-2 \sum_m \sum_k n_{mk} \log \hat{p}_{mk}$

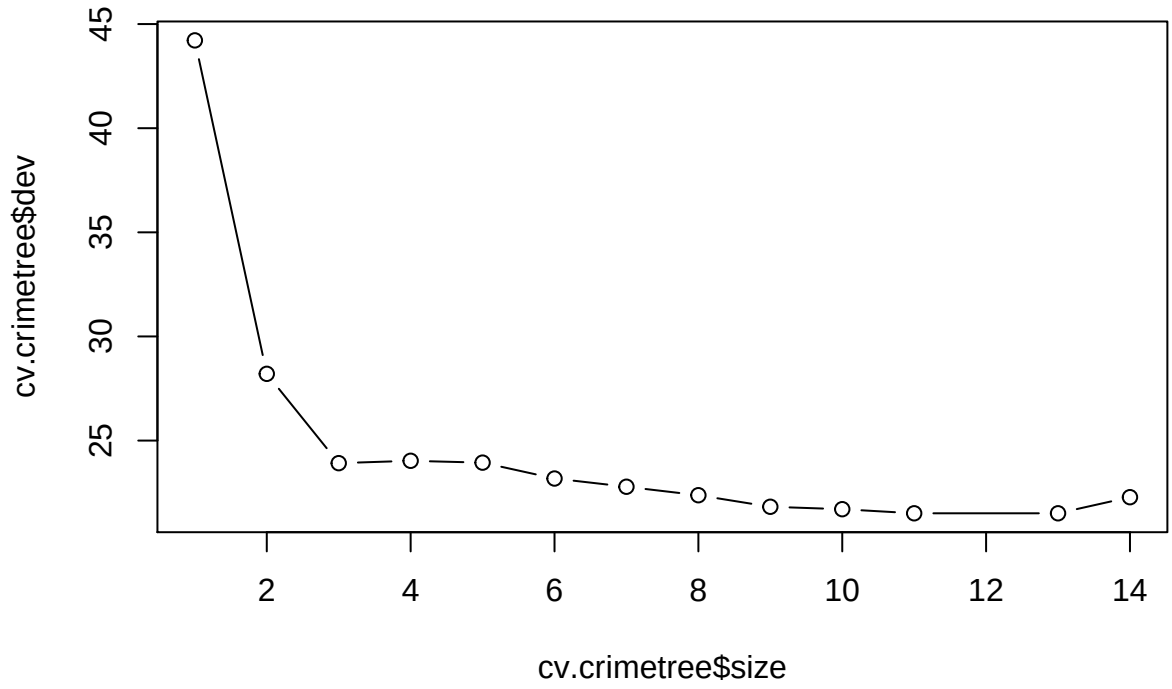
```
crime <- read.csv("http://andrewpbray.github.io/data/crime-train.csv")

dw<-select (crime,-c(state,county,community,communityname,LemasSwornFT,
 LemasSwFTPerPop,LemasSwFTFieldOps,LemasSwFTFieldPerPop,LemasTotalReq,
 LemasTotReqPerPop,PolicReqPerOffic,PolicPerPop,RacialMatchCommPol,
 PctPolicWhite,PctPolicBlack,PctPolicHisp,PctPolicAsian,PctPolicMinor
 ,OfficAssgnDrugUnits,NumKindsDrugsSeiz,PolicAveOTWorked,LandArea,PopDens,
 PctUsePubTrans,PolicCars,PolicOperBudg,LemasPctPolicOnPatr,
 LemasGangUnitDeploy,LemasPctOfficDrugUn,PolicBudgPerPop))


crimetree <- tree(ViolentCrimesPerPop ~ . - ViolentCrimesPerPop, data = dw)
summary(crimetree)
```

```
##
## Regression tree:
## tree(formula = ViolentCrimesPerPop ~ . - ViolentCrimesPerPop,
##     data = dw)
## Variables actually used in tree construction:
##  [1] "PctKids2Par"         "PctIlleg"            "FemalePctDiv"
##  [4] "PctWOFullPlumb"      "PctImmigRec5"        "PctNotSpeakEnglWell"
##  [7] "TotalPctDiv"         "PctPersDenseHous"    "racePctWhite"
## [10] "PctVacantBoarded"    "PctImmigRec8"
## Number of terminal nodes:  14
## Residual mean deviance:  0.01572 = 12.36 / 786
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.38820 -0.06195 -0.02195  0.00000  0.05261  0.55260
```

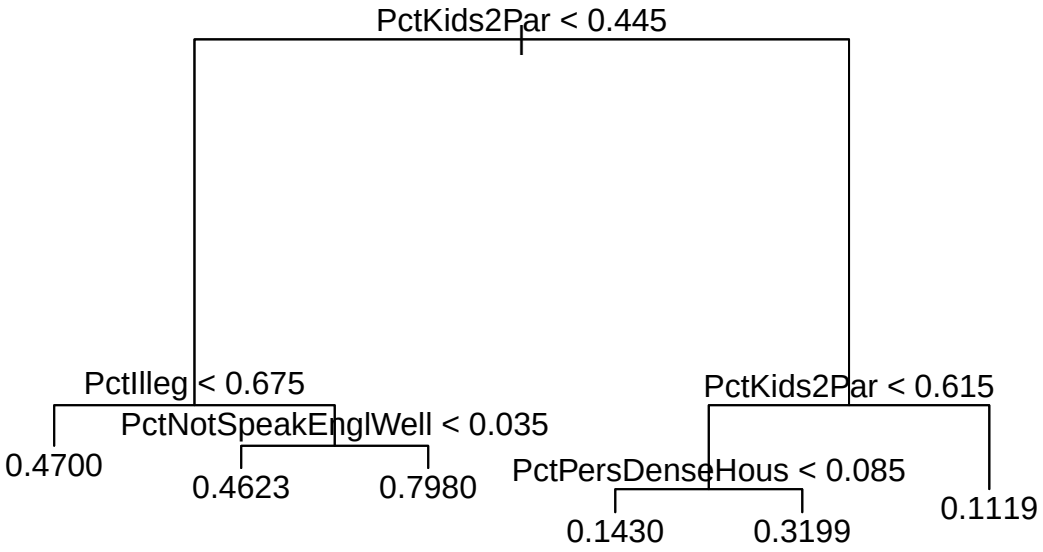
```
plot(crimetree)
text(crimetree, pretty = 3)
```



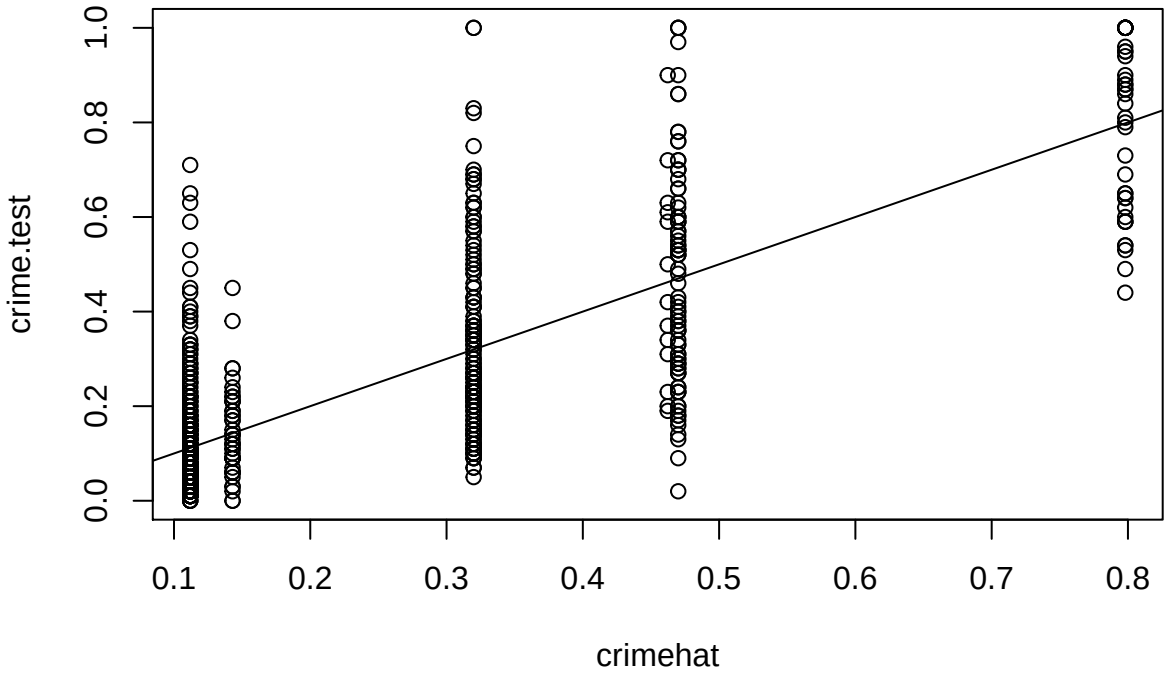

```
cv.crimetree <- cv.tree(crimetree)
plot(cv.crimetree$size, cv.crimetree$dev, type = "b")
```




```
prune.crimetree <- prune.tree(crimetree, best = 6)
plot(prune.crimetree)
text(prune.crimetree, pretty = 1)
```



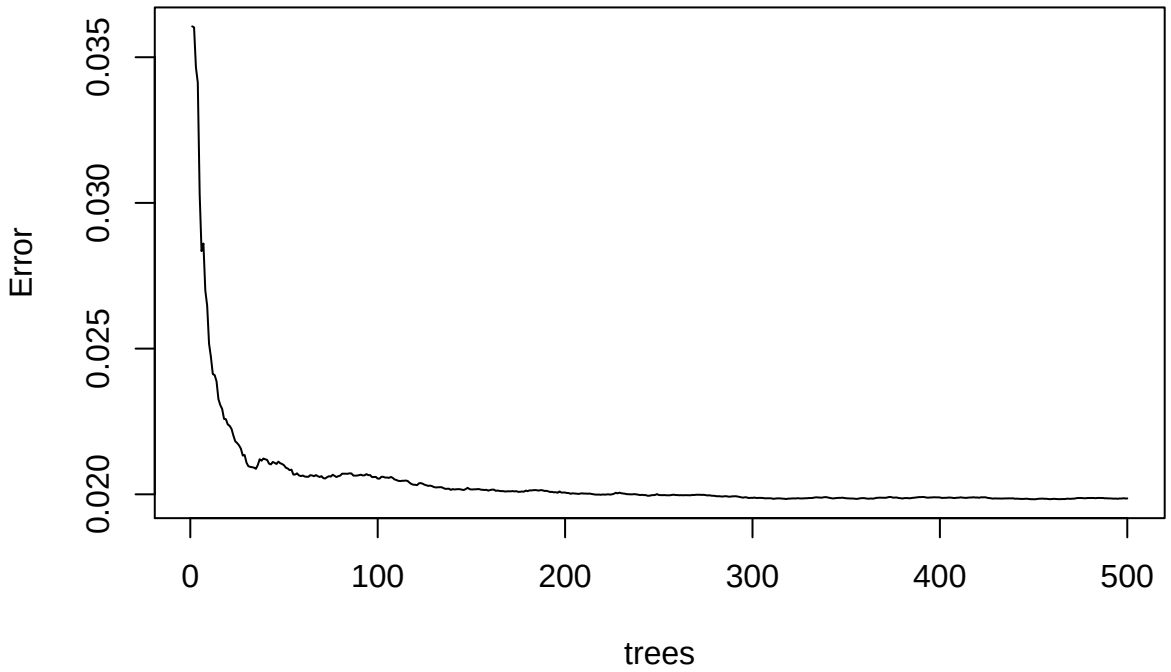

```
#need to compute test mse
test_data <- read.csv("https://bit.ly/2PYS8Ap")
crimehat <- predict(prune.crimetree, newdata = test_data)
crime.test <- test_data[, "ViolentCrimesPerPop"]
plot(crimehat, crime.test)
abline(0,1)
```




```
msepruned <- mean((crime.test - crimehat)^2)
msepruned
```

```
## [1] 0.0216835
```

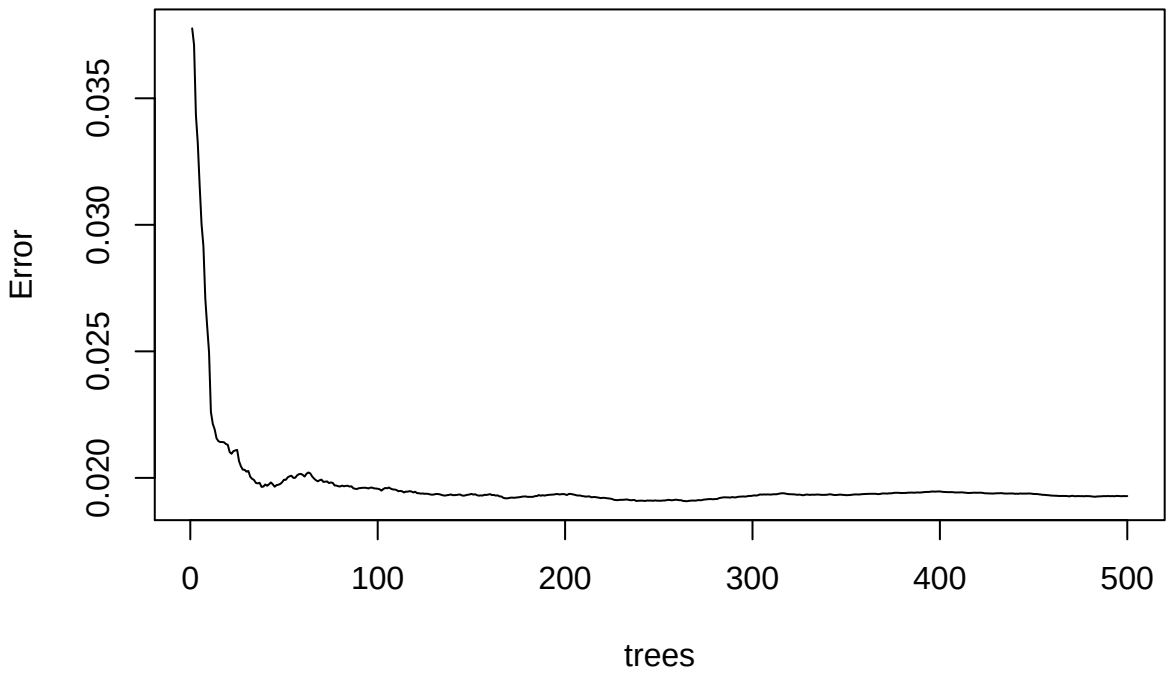
```
bag.forest97 <- randomForest(ViolentCrimesPerPop ~ . - ViolentCrimesPerPop, data = dw,
                             mtry = 96)
plot(bag.forest97)
```




```
bag.forest97
```

```
##
## Call:
##  randomForest(formula = ViolentCrimesPerPop ~ . - ViolentCrimesPerPop,      data = dw, mtry = 96)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 96
##
##           Mean of squared residuals: 0.01985777
##                     % Var explained: 63.98
```

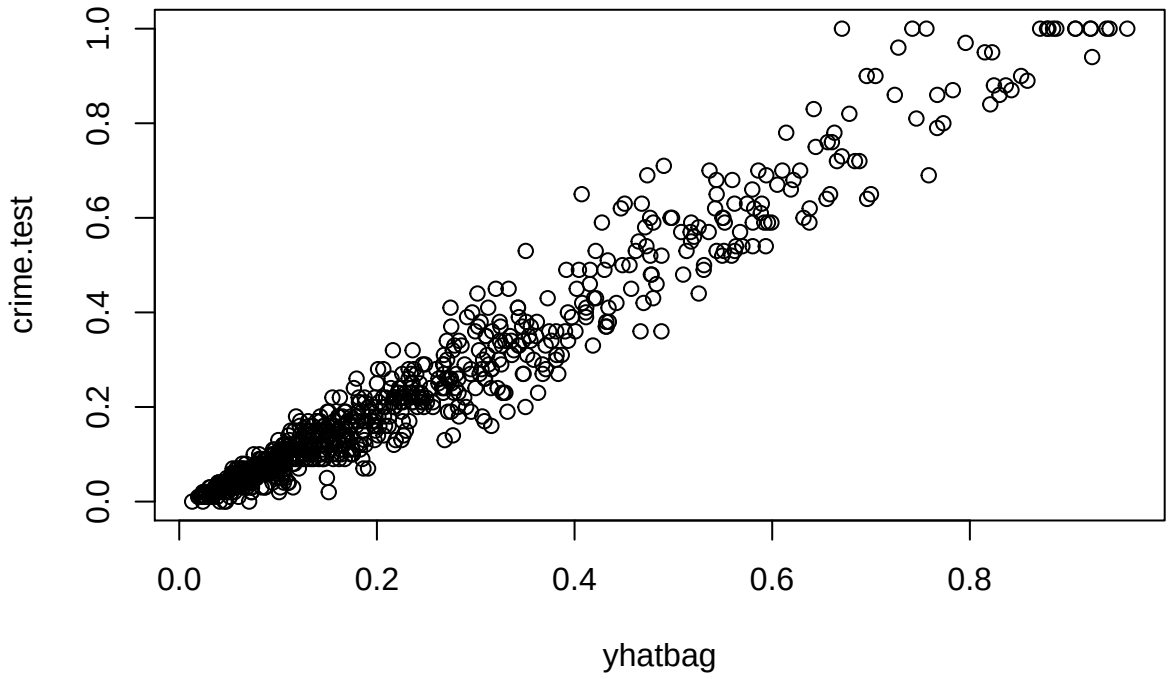
```
bag.forest32 <- randomForest(ViolentCrimesPerPop ~ . - ViolentCrimesPerPop, data = dw,
                             mtry = 32)
plot(bag.forest32)
```




```
bag.forest32
```

```
##
## Call:
##  randomForest(formula = ViolentCrimesPerPop ~ . - ViolentCrimesPerPop,      data = dw, mtry = 32)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 32
##
##           Mean of squared residuals: 0.01928098
##                     % Var explained: 65.02
```

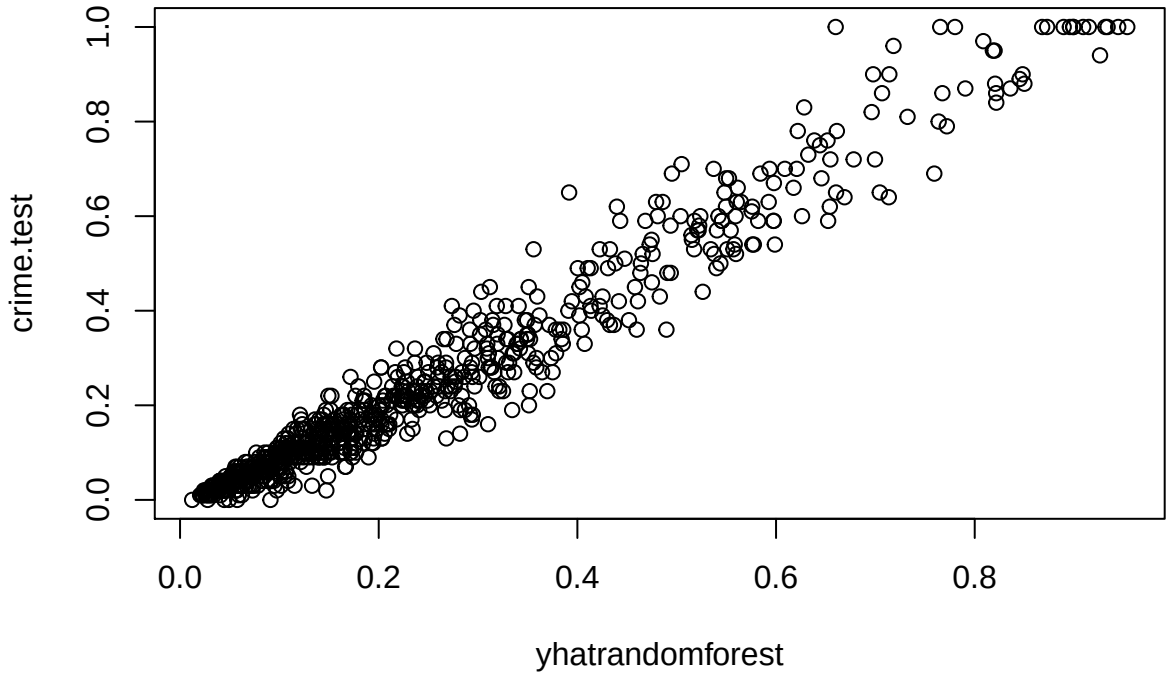
```
#computing mse's
yhatbag <- predict(bag.forest97, newdata = test_data)
plot(yhatbag, crime.test)
```




```
msebag <- mean((crime.test - yhatbag)^2)
msebag
```

```
## [1] 0.003137841
```

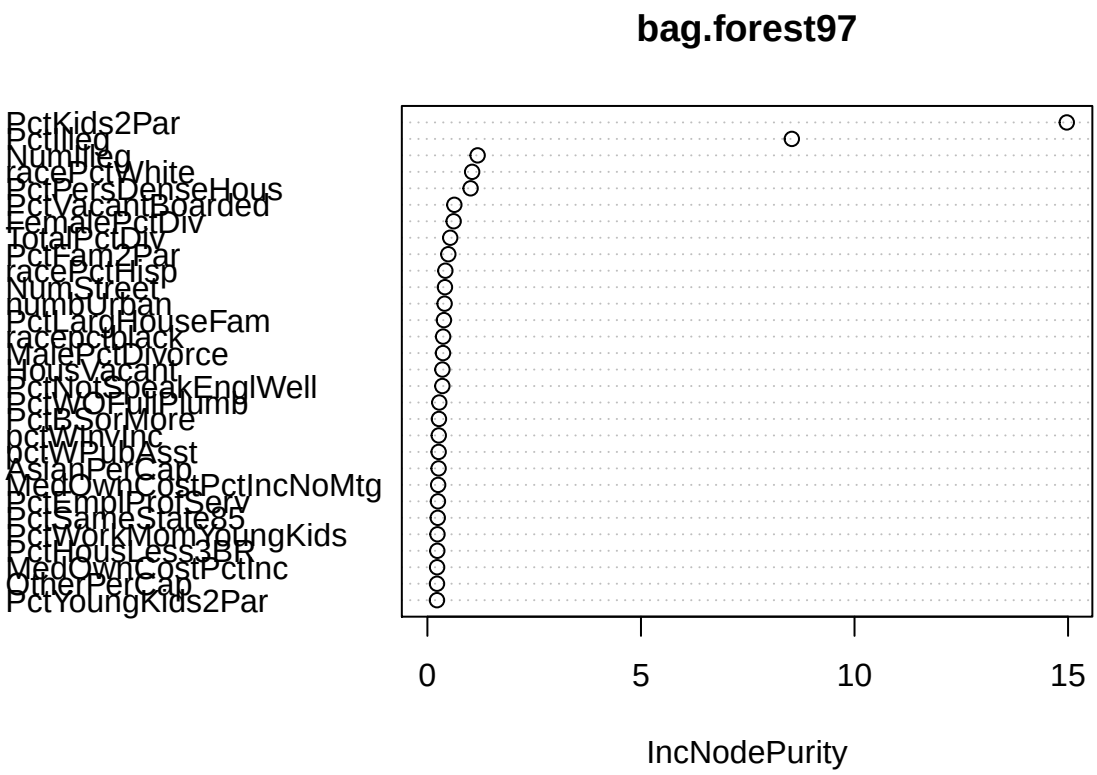
```
yhatrandomforest <- predict(bag.forest32, newdata = test_data)
plot(yhatrandomforest, crime.test)
```



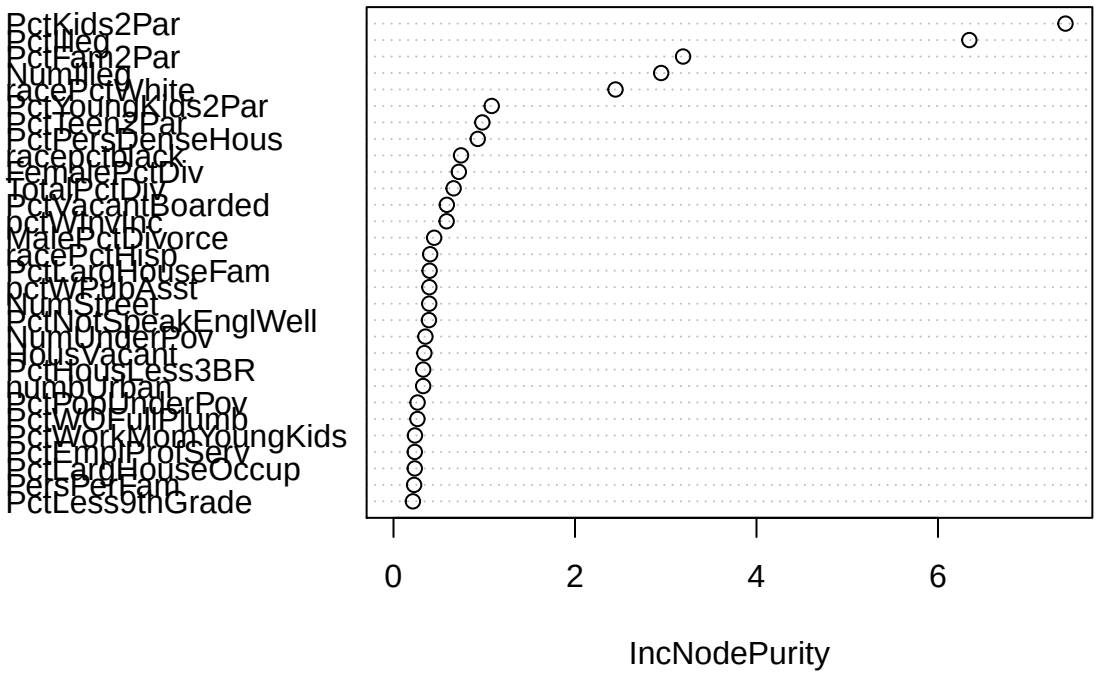

```
mserf<- mean((crime.test - yhatrandomforest)^2)
mserf
```

```
## [1] 0.003092507
```

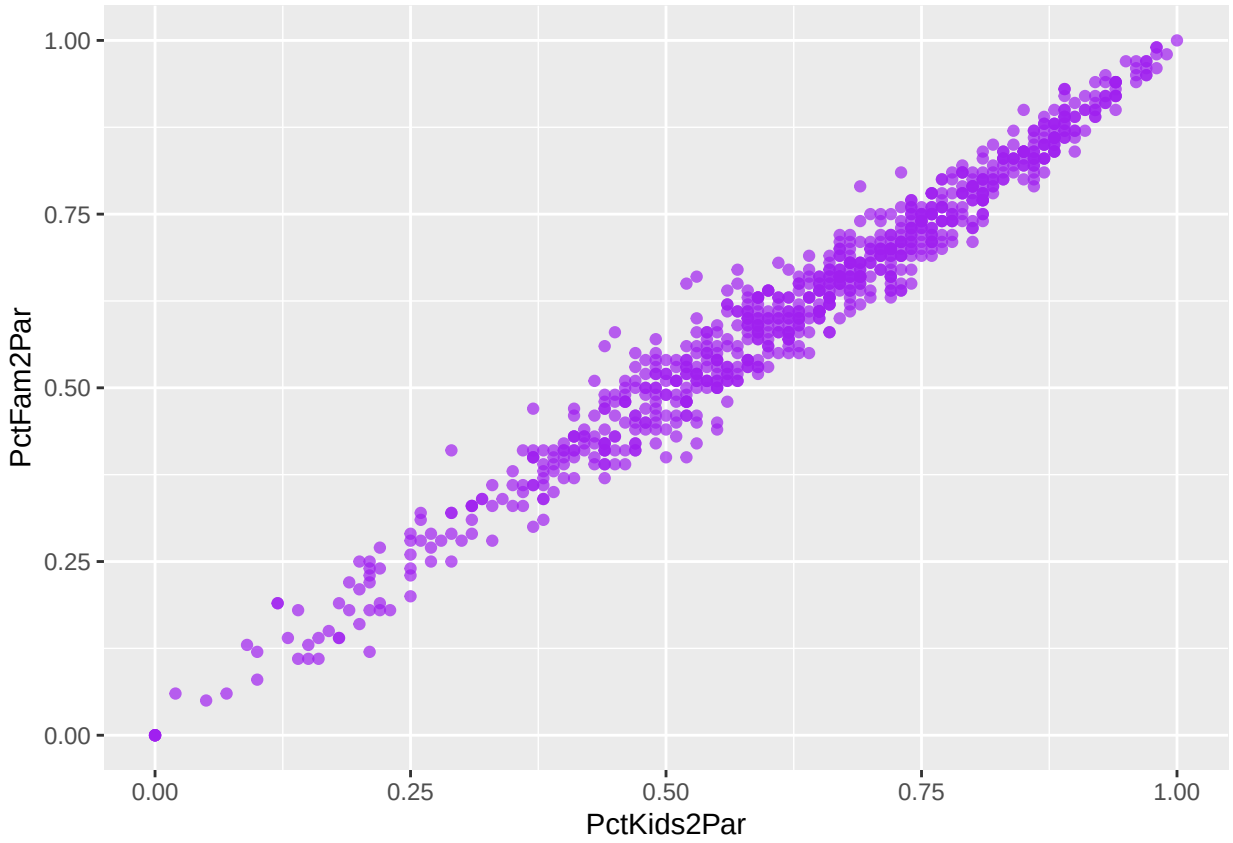
```
varImpPlot(bag.forest97)
```




```
varImpPlot(bag.forest32)
```



```
ggplot(data = crime, mapping = aes(x = PctKids2Par, y = PctFam2Par)) +
  geom_point(color = "purple", alpha = .7)
```



MSE for pruned tree is .02168 My MSE for my regression model is ~.01436, so this pruned tree did not beat it. My MSE for my bagged tree was super low actually, .00313 My MSE for my other random forest model with m = p/3 is .0031, barely beating out the bagged tree. Both of these bootstrapped trees beat my regression model and pruned tree.

I would say that these variable importance plots are pretty similar to what I found in lab 3. Variables in common include: PctKids2Par, PctIlleg, racePctWhite, PctPerDenseHous, MalePctDivorce. The only issue is that I think that these tree models do not distinguish between highly correlated variables which is something that my group kept in mind for the linear regression challenge (for example, PctKids2Par and PctYoungKids2Par)