# Homework 4

*Xinyi Lin*

*4/19/2019*

```
library(lasso2) # only for data
library(rpart) # for cart model
library(rpart.plot)
library(randomForest)
library(ranger)
library(caret)
library(gbm) # for boosting model
```

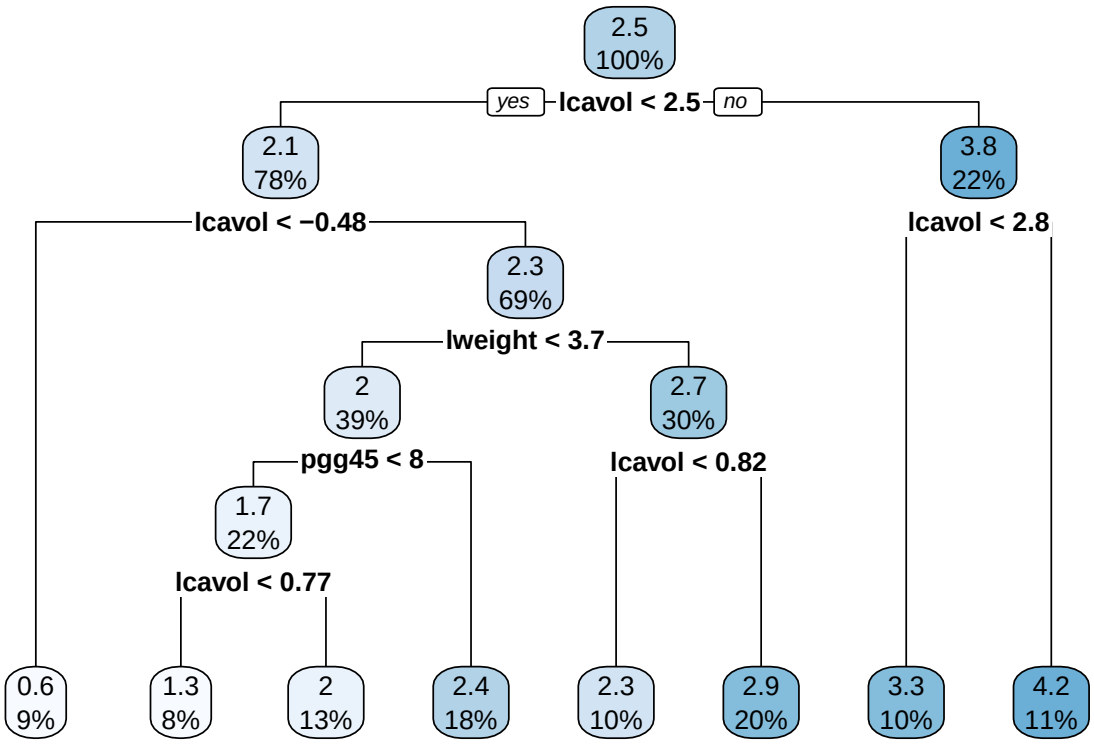
```
data(Prostate)
```

## Problem 1

### Question 1

Fit the regression tree.

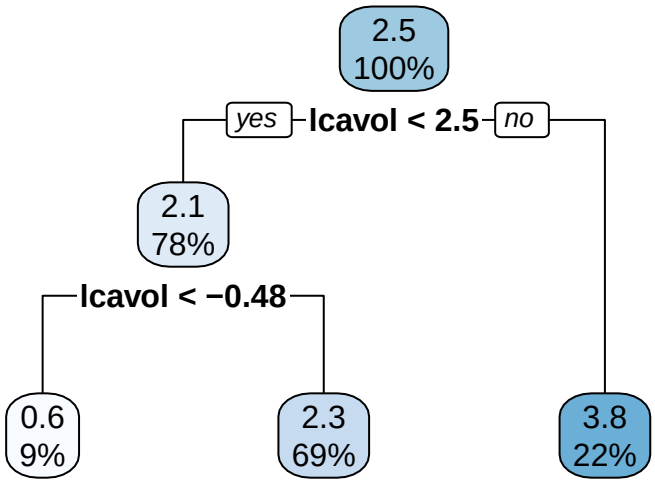
```
set.seed(123)
tree1 <- rpart(formula = lpsa~., data = Prostate)
rpart.plot(tree1)
```




Use cross-validation to determine the optimal tree size. The following is the optimal tree.

```
set.seed(123)
tree2 <- rpart(formula = lpsa~., data = Prostate,
```

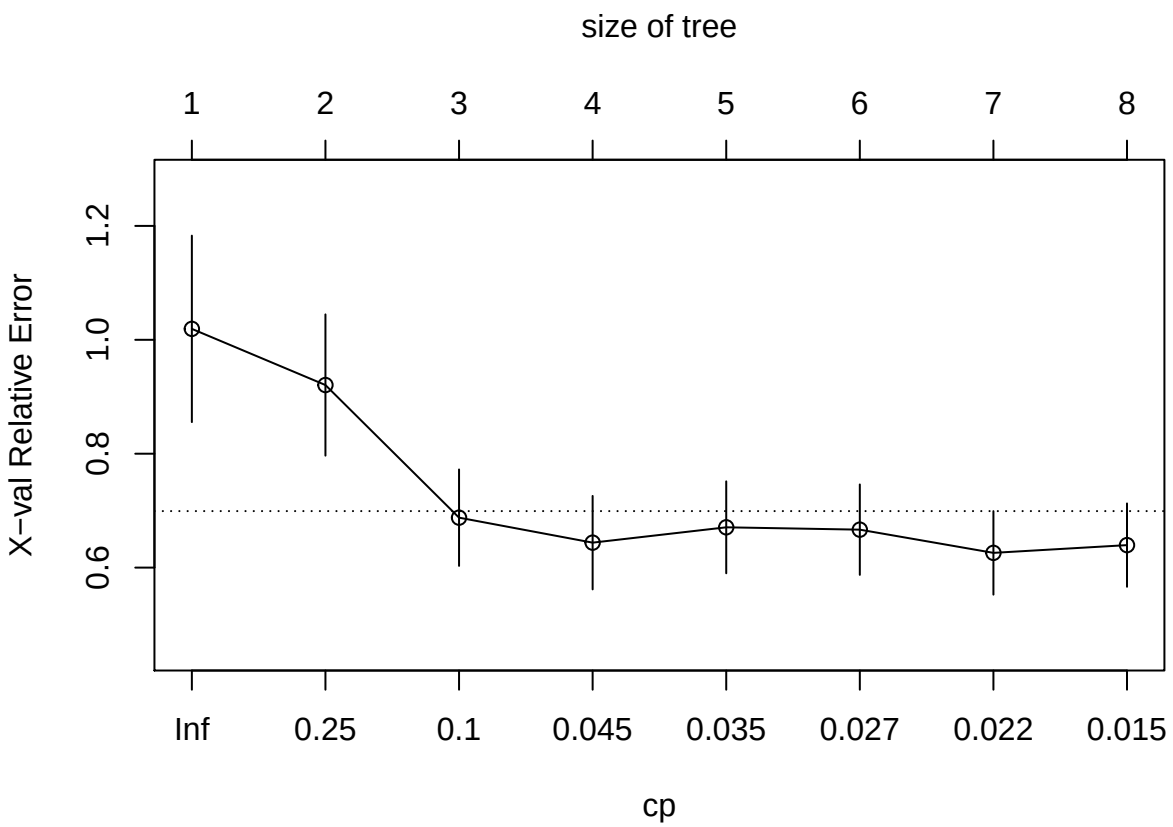
```
              control = rpart.control(cp = 0.1))
rpart.plot(tree2)
```




```
cpTable <- printcp(tree1)
```

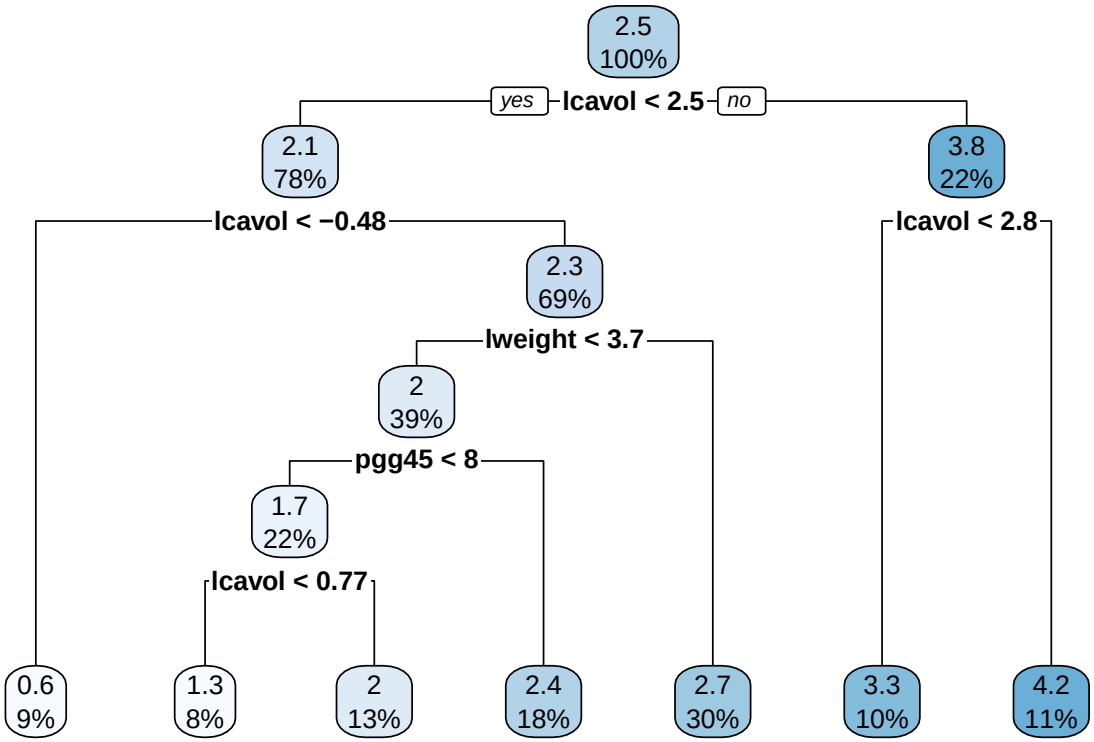
```
##
## Regression tree:
## rpart(formula = lpsa ~ ., data = Prostate)
##
## Variables actually used in tree construction:
## [1] lcavol  lweight pgg45
##
## Root node error: 127.92/97 = 1.3187
##
## n= 97
##
##         CP nsplit rel error  xerror     xstd
## 1 0.347108      0   1.00000 1.01919 0.163712
## 2 0.184647      1   0.65289 0.92059 0.124004
## 3 0.059316      2   0.46824 0.68769 0.084684
## 4 0.034756      3   0.40893 0.64380 0.082051
## 5 0.034609      4   0.37417 0.67073 0.080740
## 6 0.021564      5   0.33956 0.66664 0.079424
## 7 0.021470      6   0.31800 0.62587 0.073369
## 8 0.010000      7   0.29653 0.63949 0.073227
```

```
plotcp(tree1)
```



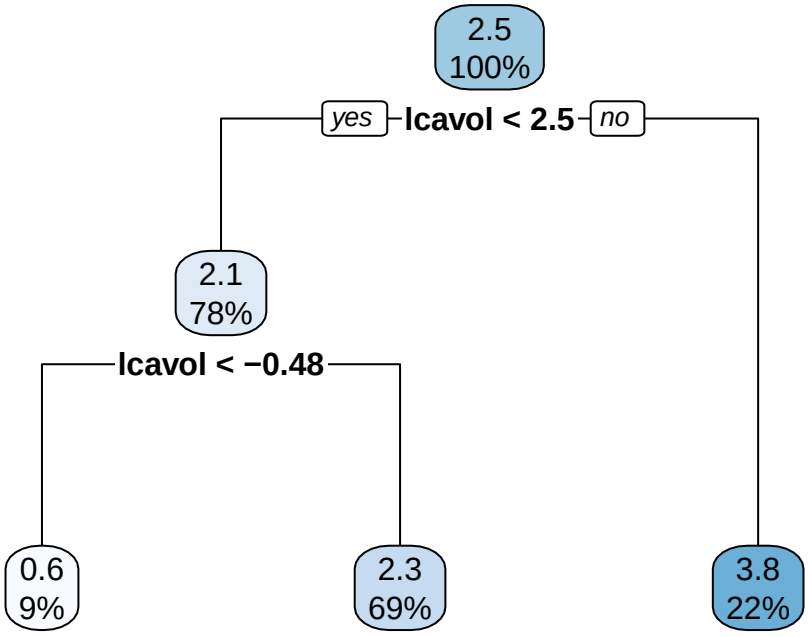

According to the plot and results given by `cpTable` function, we can find that when the number of splits equals to 6, the tree have lowest cross-validation error. The optimal tree is shown below.

```
minErr <- which.min(cpTable[,4])
# minimum cross-validation error
tree3 <- prune(tree1, cp = cpTable[minErr,1])
rpart.plot(tree3)
```




Using the 1 SE rule to obtain optimal tree size.

```
tree4 <- prune(tree1, cp = cpTable[cpTable[,4]<cpTable[minErr,4]+cpTable[minErr,5],1][1])
rpart.plot(tree4)
```

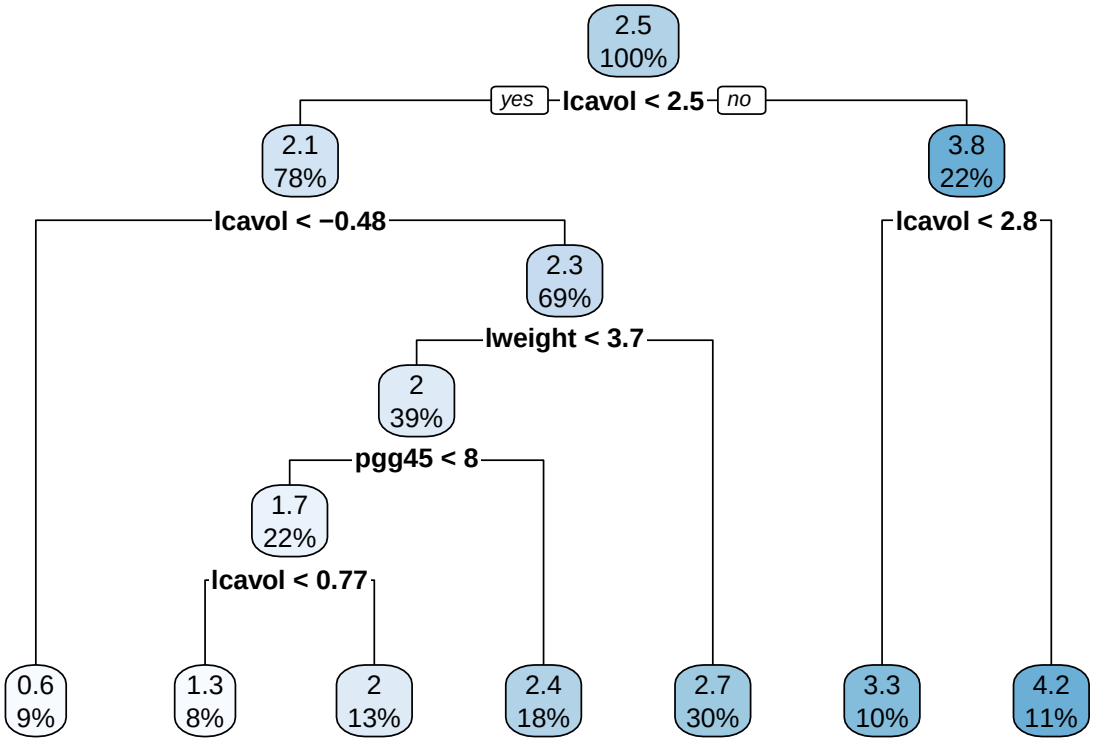



By comparing two tree, we can find that the optimal tree sizes given by cross-validation and 1 SE rule are different.

### Question 2

We choose the final tree based on the cross-validation error and following is the final tree.

```
rpart.plot(tree3)
```




Interpretation of the node 3.3:

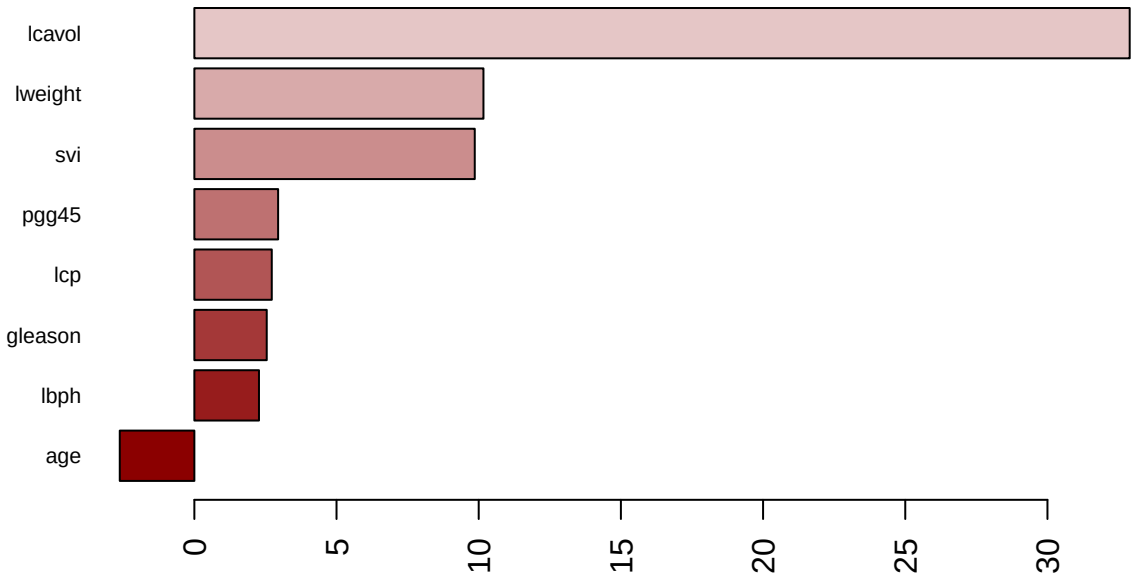
If the log of cancer volume is equals or larger than 2.5 and smaller than 3.8, than the log of prostate specific antigen is 3.3.

### Question 3

Fit the bagging model and get the variable importance.

```
set.seed(123)
bagging.final.per <- ranger(lpsa~., Prostate,
                            mtry = 8, splitrule = "variance",
                            min.node.size = 2,
                            importance = "permutation",
                            scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(bagging.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("darkred","white","darkblue"))(19))
```

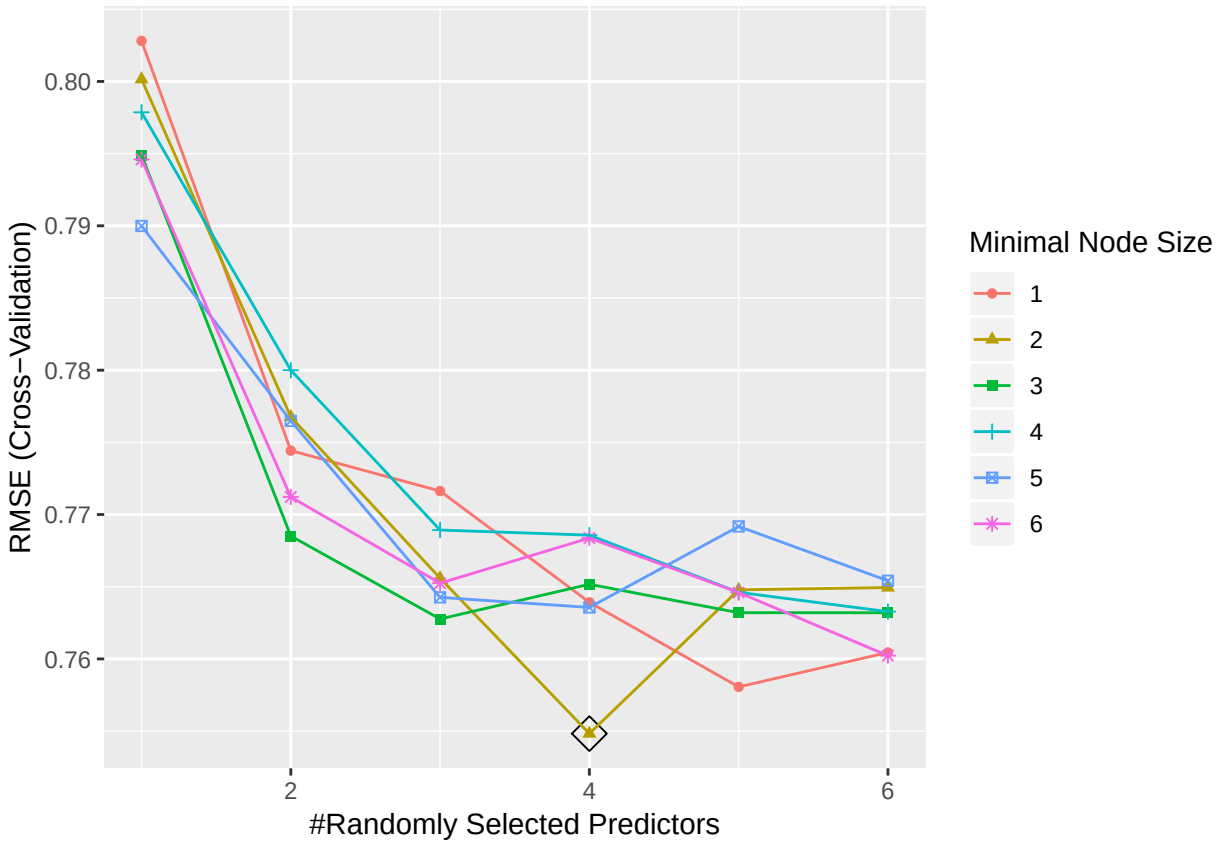



### Question 4

Using `caret` package to find out the best mtry.

```
rf.grid <- expand.grid(mtry = 1:6,
                       splitrule = "variance",
                       min.node.size = 1:6)
set.seed(123)
ctrl <- trainControl(method = "cv")
rf.fit <- train(lpsa~., Prostate,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl)

ggplot(rf.fit, highlight = TRUE)
```

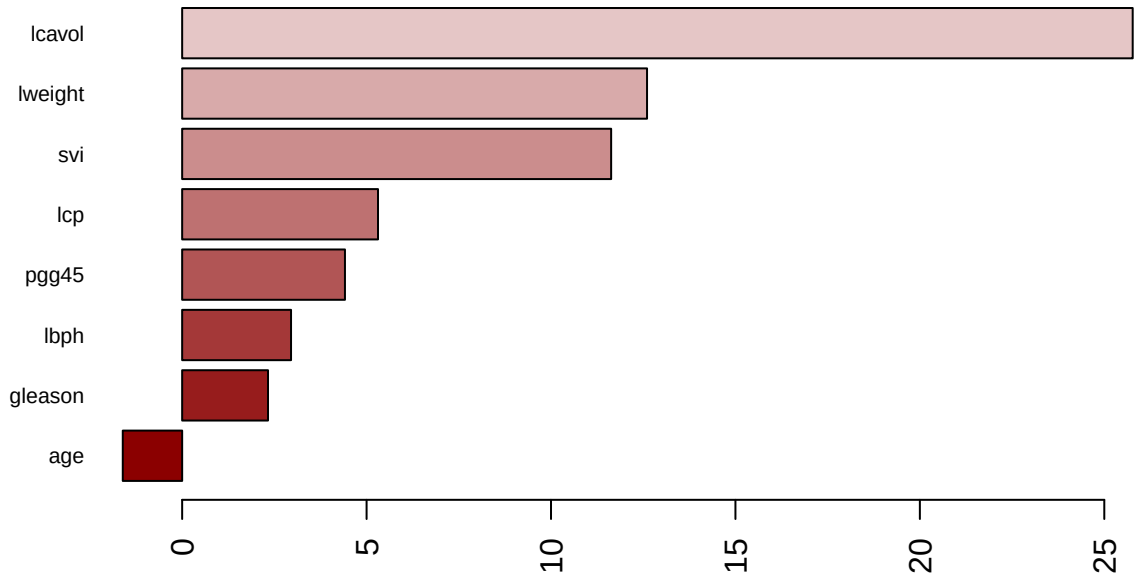



According to the result, the best mtry is 4 and best minimal node size is 2.

Fit the bagging model and get the variable importance.

```
set.seed(123)
bagging.final.per <- ranger(lpsa~., Prostate,
                          mtry = 4, splitrule = "variance",
                          min.node.size = 2,
                          importance = "permutation",
                          scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(bagging.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("darkred","white","darkblue"))(19))
```




### Question 5

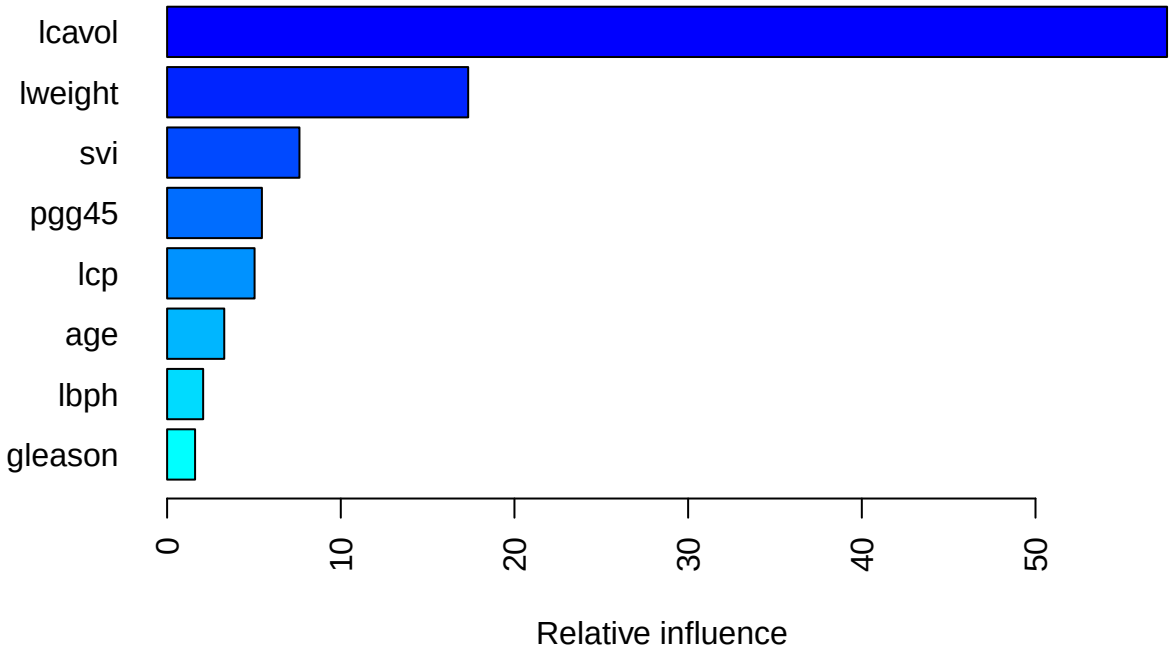
First, tune gbm model.

```
gbm.grid <- expand.grid(n.trees = c(2000,3000),
                        interaction.depth = 2:10,
                        shrinkage = c(0.001,0.003,0.005),
                        n.minobsinnode = 1)
set.seed(1)
gbm.fit <- train(lpsa~., Prostate,
                 method = "gbm",
                 tuneGrid = gbm.grid,
                 trControl = ctrl,
                 verbose = FALSE)

ggplot(gbm.fit, highlight = TRUE)
```


Get the variable importance.

```
summary(gbm.fit$finalModel, las = 2, cBars = 19, cex.names = 1)
```




```
##              var   rel.inf
## lcavol   lcavol 57.574689
## lweight lweight 17.335025
```

```
## svi          svi  7.619823
## pgg45      pgg45  5.457903
## lcp          lcp  5.036310
## age          age  3.289945
## lbph        lbph  2.075507
## gleason gleason  1.610798
```