p8106_stl2137_hw4

Question 1

```r
### Loading in prostate data
data("Prostate")
dat_prostate <- Prostate
```

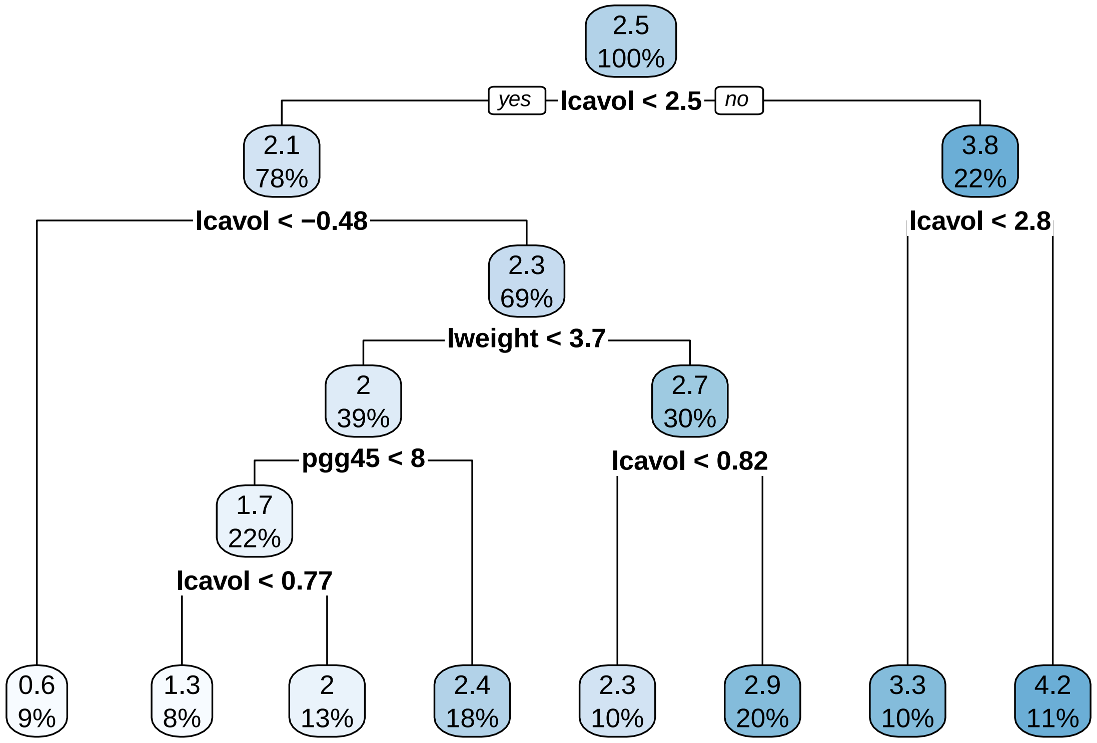
Part A

```r
set.seed(1)

tree_prost_1 <- rpart(formula = lpsa ~., data = dat_prostate)
rpart.plot(tree_prost_1)

### Finding lowest cp
cp_table <- printcp(tree_prost_1)
```

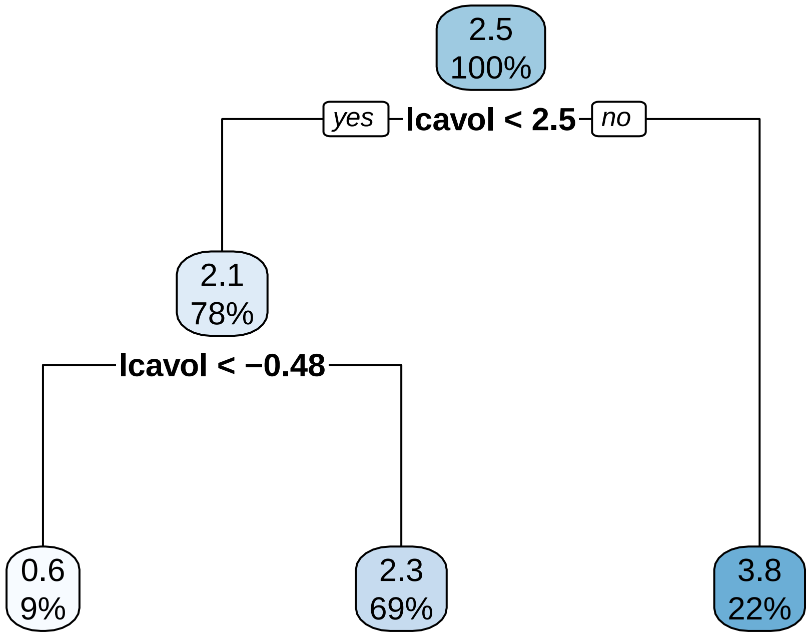
```
##
## Regression tree:
## rpart(formula = lpsa ~ ., data = dat_prostate)
##
## Variables actually used in tree construction:
## [1] lcavol  lweight pgg45
##
## Root node error: 127.92/97 = 1.3187
##
## n= 97
##
##          CP nsplit rel error  xerror     xstd
## 1 0.347108      0   1.00000 1.01323 0.162162
## 2 0.184647      1   0.65289 0.88779 0.111915
## 3 0.059316      2   0.46824 0.59168 0.066102
## 4 0.034756      3   0.40893 0.61359 0.069269
## 5 0.034609      4   0.37417 0.58640 0.067630
## 6 0.021564      5   0.33956 0.57853 0.068772
## 7 0.021470      6   0.31800 0.56398 0.067155
## 8 0.010000      7   0.29653 0.54721 0.068034
```

```r
#plotcp(tree_prost_1)

minErr <- which.min(cp_table[,4])

# minimum cross-validation error
tree_prost_3 <- prune(tree_prost_1, cp = cp_table[minErr, 1])

# 1SE rule
tree_prost_4 <- prune(tree_prost_1, cp =
                        cp_table[cp_table[,4] < cp_table[minErr, 4] + cp_table[minErr, 5], 1][1])
```

```r
rpart.plot(tree_prost_3)
```



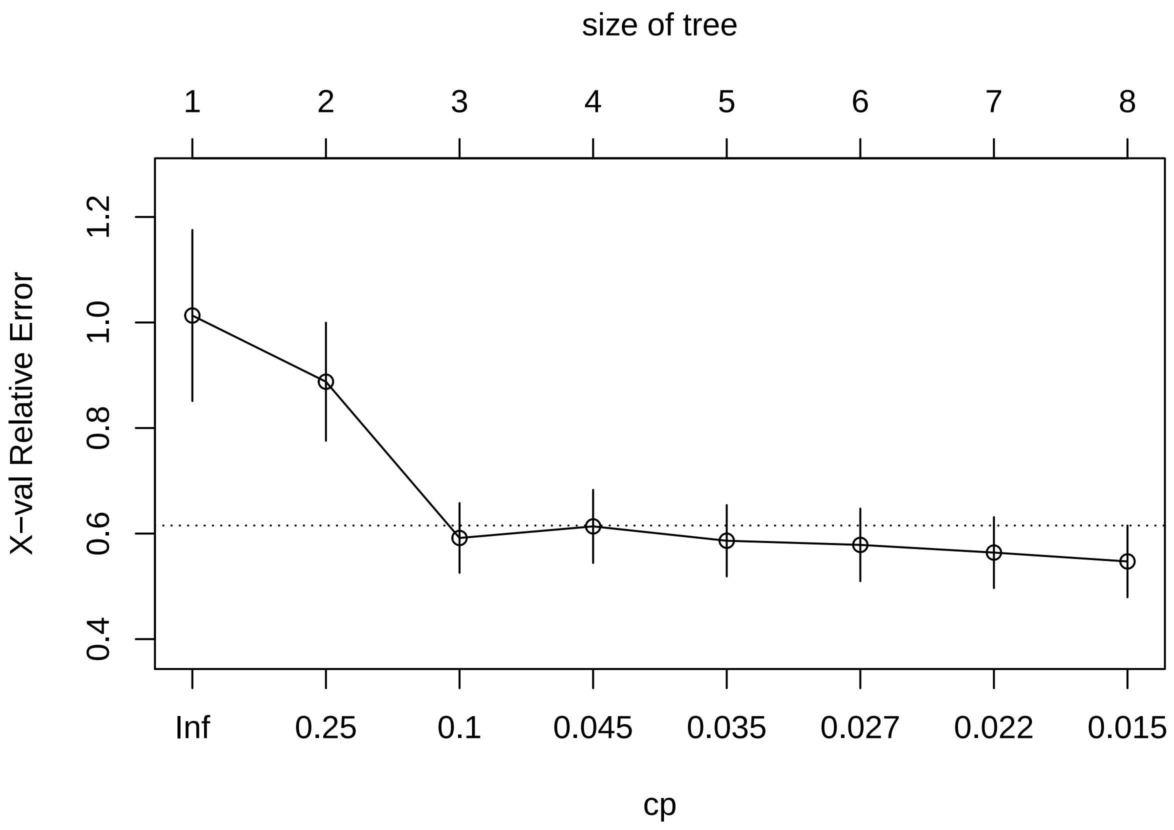
```r
rpart.plot(tree_prost_4)
```



The tree size that corresponds to the lowest cross-validation error is 8. This is not the same tree size as the one obtained using the 1 SE rule, as the tree size is 3.

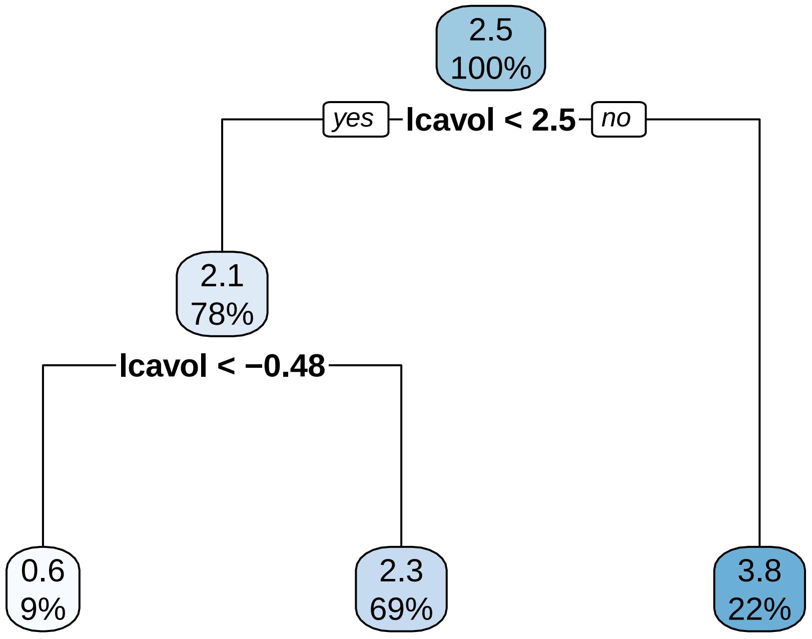
Part B

```r
plotcp(tree_prost_1)
```



Based off the leftmost value for which the mean lies below the horizontal line in the cp plot, a cp = 0.1 and a tree size of 3 should be utilized.

```r
set.seed(1)

final_tree_prost <- rpart(lpsa ~ ., data = dat_prostate,
                          control = rpart.control(cp = 0.1))

rpart.plot(final_tree_prost)
```



If you have a log cancer volume greater than 2.5, we predict that you will have a log prostate specific antigen level of 3.8.

Part C

```r
set.seed(1)

bagging_prost <- randomForest(lpsa ~ ., data = dat_prostate,
                              mtry = 8)

bagging_prost$importance
```

```
##         IncNodePurity
## lcavol      76.557359
## lweight     16.761566
## age          5.875410
## lbph         5.123664
## svi          6.534788
## lcp          5.937665
## gleason      1.096503
## pgg45        5.776304
```

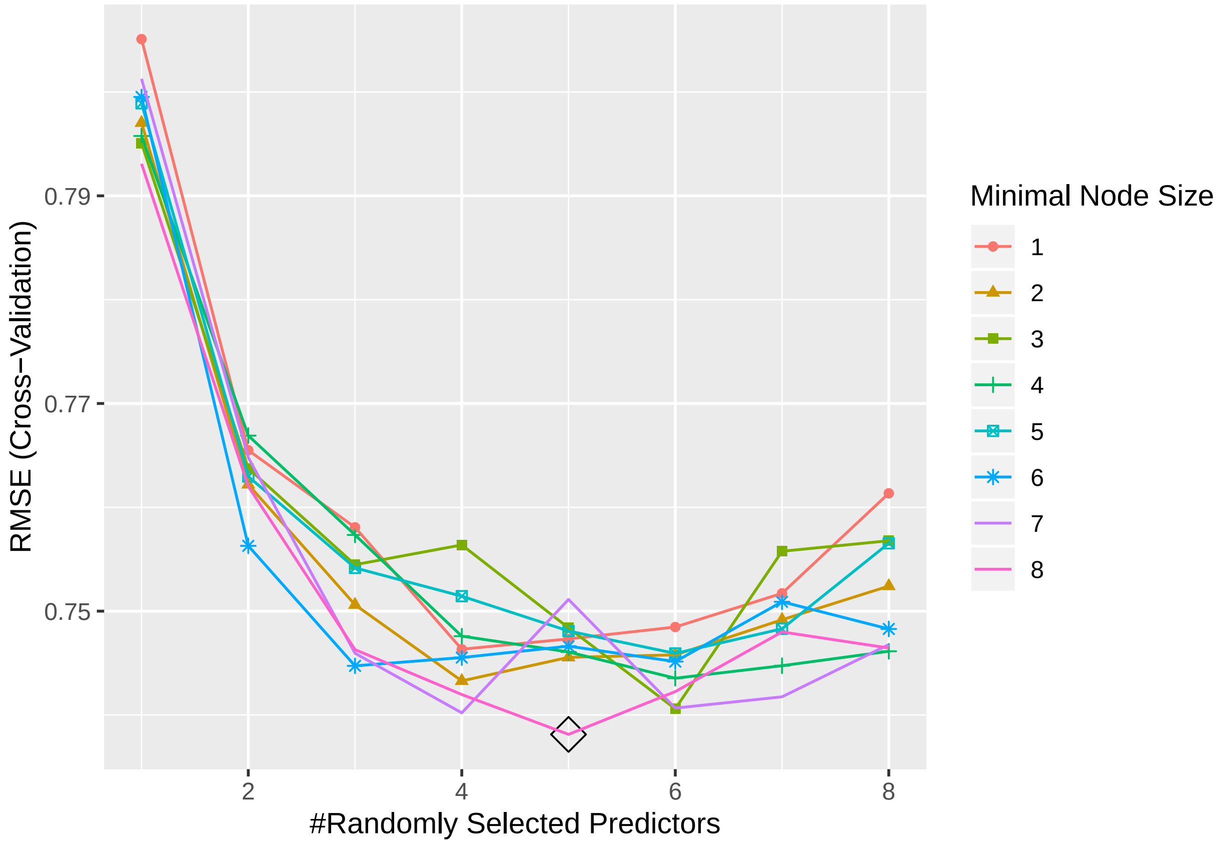
The variable for log cancer volume, at a value of 76.557359, has the highest variable importance. The variable for log of prostate weight, at the value of 16.7615664, has the 2nd highest variable importance.

Using caret for RF

```r
control <- trainControl(method = "cv")

rf_grid_prost <- expand.grid(mtry = 1:8,
                       splitrule = "variance",
                       min.node.size = 1:8)
```

```r
set.seed(1)
rf_fit_prost <- train(lpsa ~., dat_prostate,
                  method = "ranger",
                  tuneGrid = rf_grid_prost,
                  trControl = control)

ggplot(rf_fit_prost, highlight = TRUE)
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 8.
## Consider specifying shapes manually if you must have them.

## Warning: Removed 16 rows containing missing values (geom_point).
```



Part D

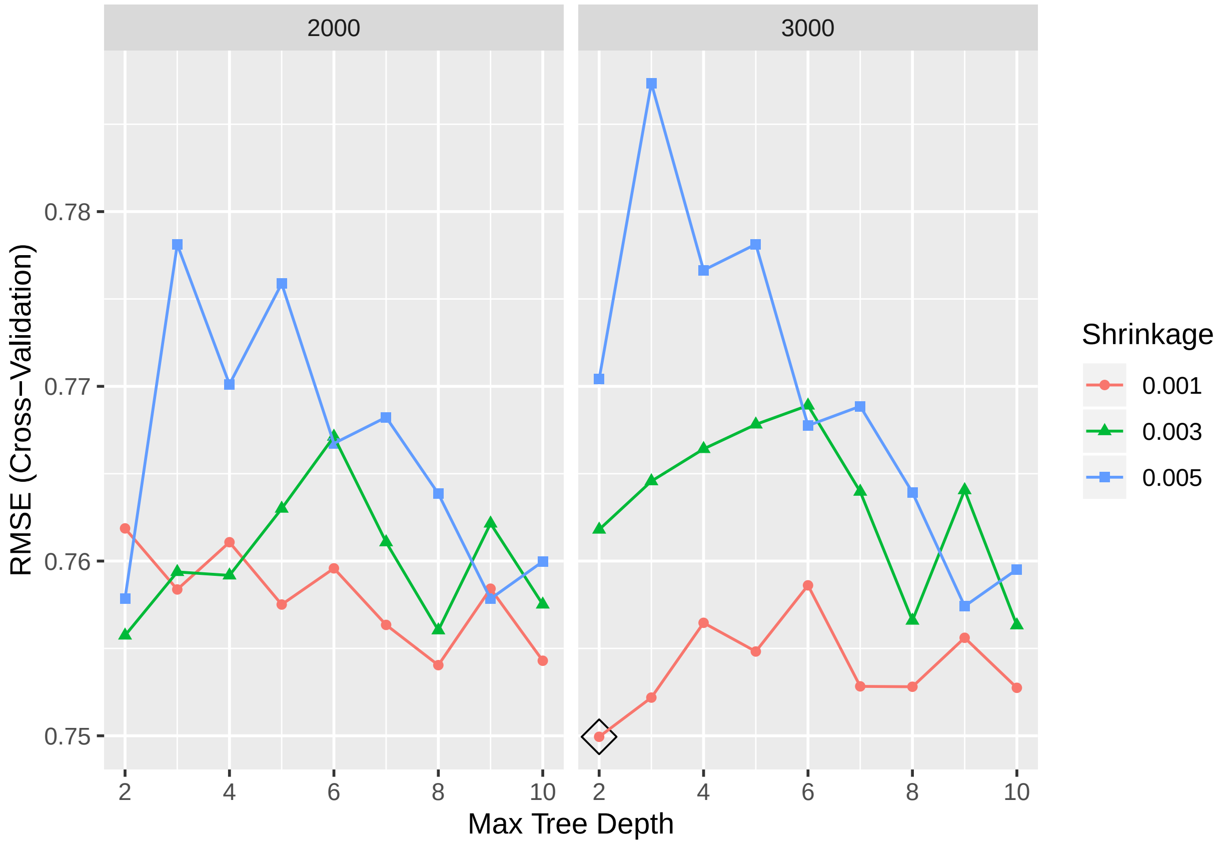
```r
set.seed(1)
rf_prost <- randomForest(lpsa ~ ., data = dat_prostate,
                         mtry = 3)

rf_prost$importance
```

```
##         IncNodePurity
## lcavol      44.037365
## lweight     20.439936
## age          7.901313
## lbph          6.974511
## svi          12.779901
## lcp          13.149594
## gleason       4.205354
## pgg45        11.130664
```

The variable for log cancer volume, at a value of 44.037365, has the highest variable importance. The variable for log of prostate weight, at the value of 20.439936, has the 2nd highest variable importance.

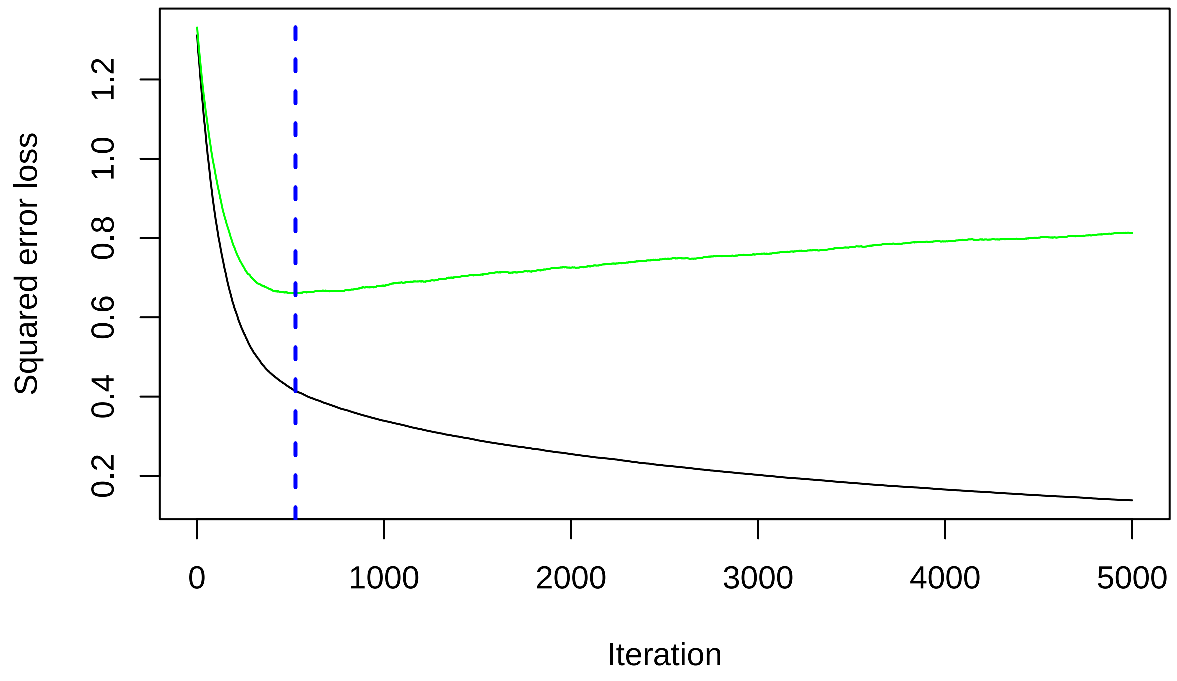
Caret for Boosting

```r
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_prost <- train(lpsa ~ ., dat_prostate,
                       method = "gbm",
                       tuneGrid = gbm_grid,
                       trControl = control,
                       verbose = FALSE)

ggplot(gbm_fit_prost, highlight = TRUE)
```

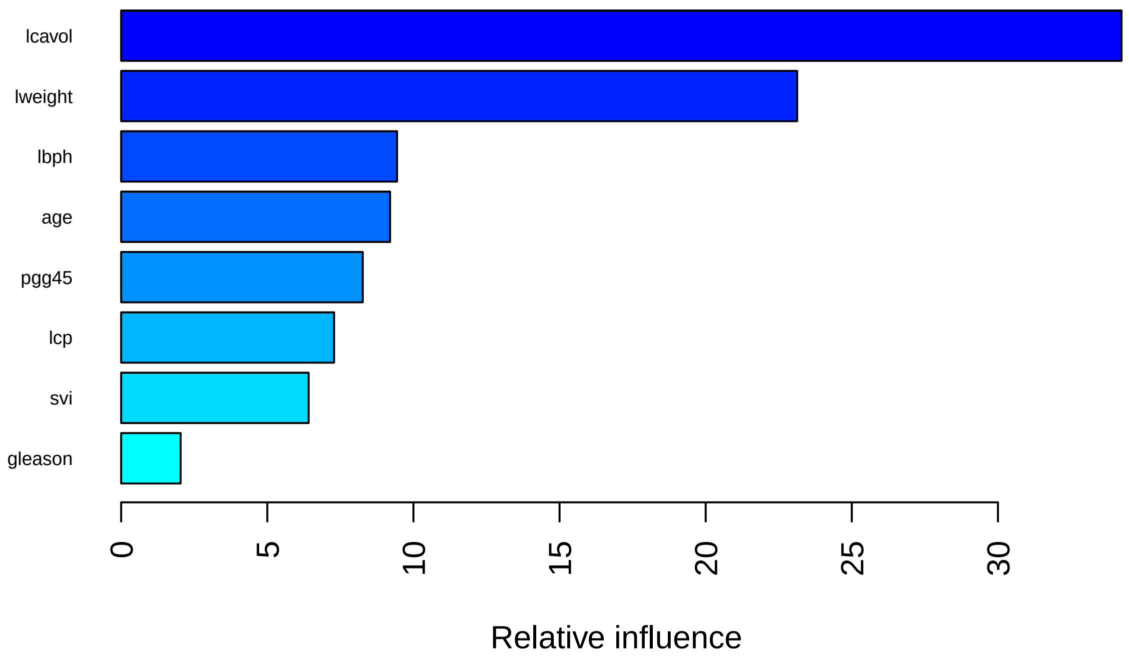


Part E

```r
set.seed(1)
boosting_prost <- gbm(lpsa ~ ., data = dat_prostate,
                      distribution = "gaussian",
                      n.trees = 5000,
                      interaction.depth = 3,
                      shrinkage = 0.005,
                      cv.folds = 10)

nt <- gbm.perf(boosting_prost, method = "cv")
```



```r
boosting_plot <- summary(boosting_prost, las = 2, cBars = 19, cex.names = 0.6)
```



The variable for log cancer volume, at a value of 34.2295525, has the highest variable importance. The variable for log of prostate weight, at the value of 23.1311753, has the 2nd highest variable importance.

Part F

```
resamp <- resamples(list(rf = rf_fit_prost, gbm = gbm_fit_prost))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: rf, gbm
## Number of resamples: 10
##
## MAE
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf  0.4737468 0.5269883 0.6122238 0.6001549 0.6672875 0.7080113    0
## gbm 0.4988605 0.5382599 0.6238535 0.6060614 0.6541746 0.7189084    0
##
## RMSE
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf  0.5831535 0.6354143 0.7101871 0.7381295 0.8276041 0.9995138    0
## gbm 0.6127814 0.6613586 0.7369285 0.7499428 0.8330904 0.9285269    0
##
## Rsquared
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf  0.3927114 0.5134677 0.5953671 0.6059044 0.7308171 0.7825008    0
## gbm 0.3912307 0.4940316 0.6253058 0.6103438 0.7211938 0.8171347    0
```

I would pick the random forest model. When comparing the RMSE of the two models when resampling, random forest has the lower mean RMSE.

Problem 2

Pulling & Creating Training/Test Datasets

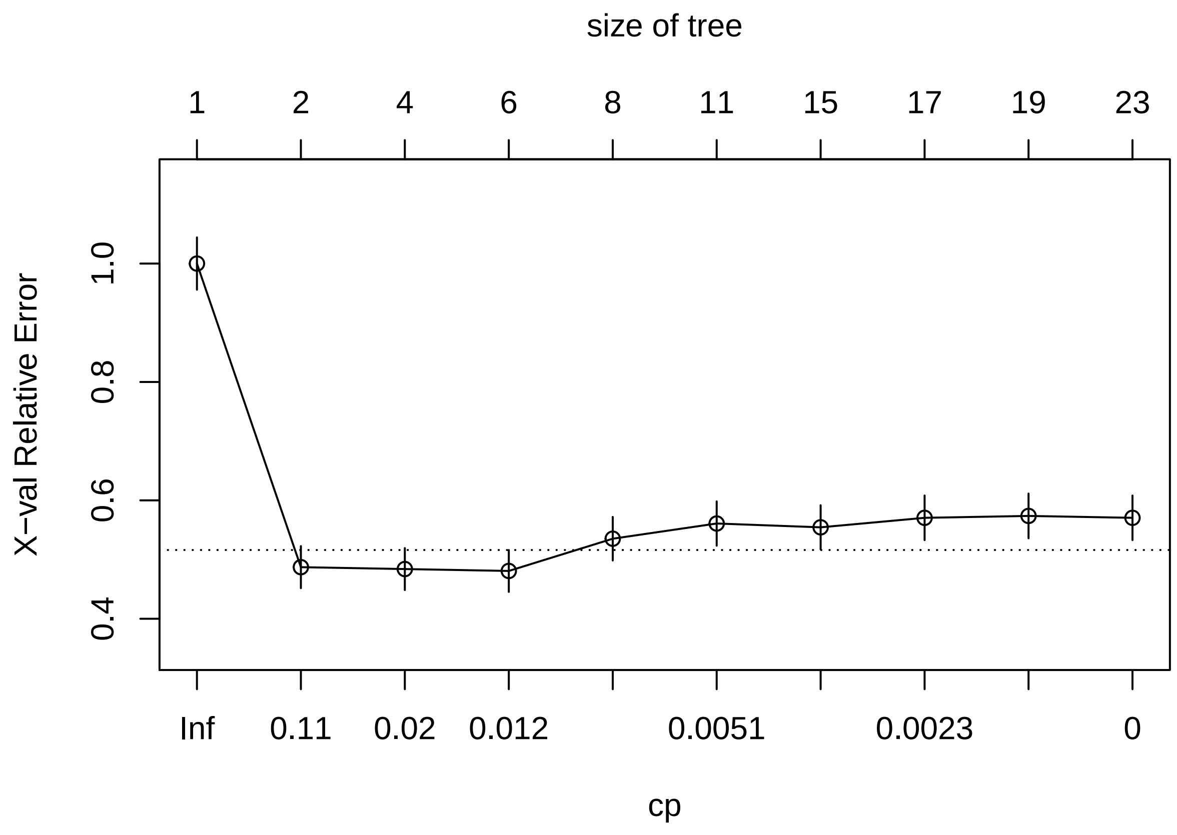
```
data("OJ")
dat_oj <- OJ %>%
  janitor::clean_names()

set.seed(1)

rowTrain <- createDataPartition(y = dat_oj$purchase,
                                p = 799/1070,
                                list = FALSE)

train_dat_oj <- as.data.frame(dat_oj[rowTrain,])
test_dat_oj <- as.data.frame(dat_oj[-rowTrain,])
```

Part A

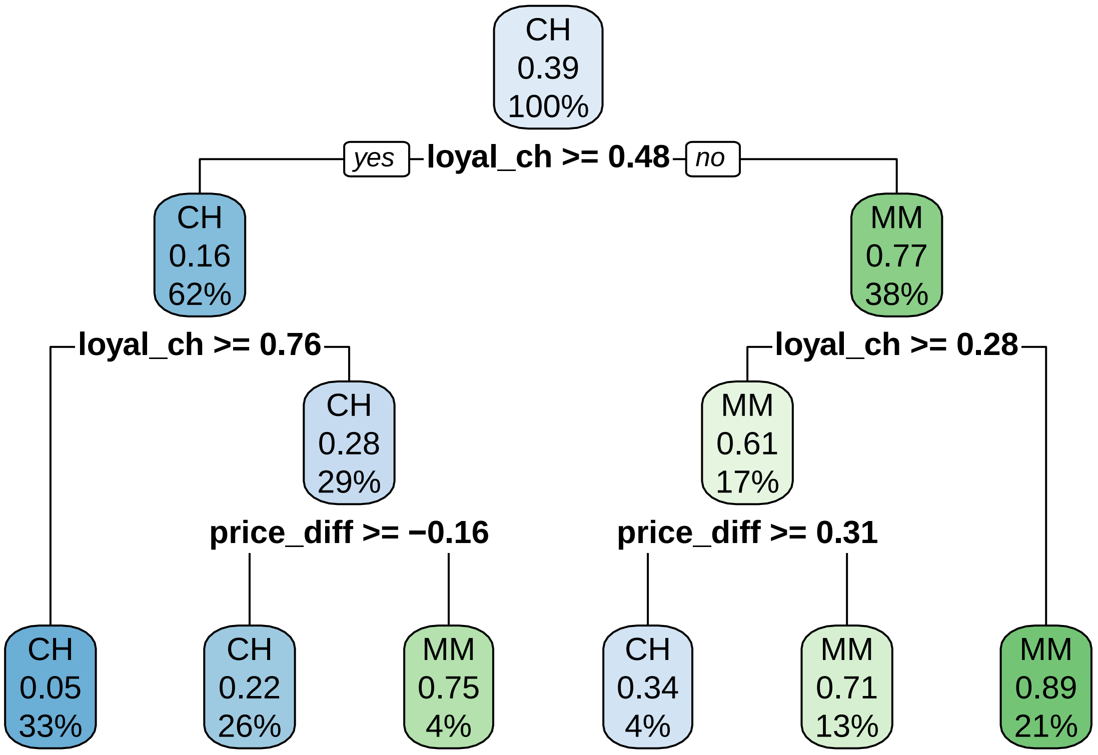
```
set.seed(1)
oj_tree <- rpart(purchase ~., data = train_dat_oj,
                 control = rpart.control(cp = 0))
#rpart.plot(oj_tree)

cp_table_oj <- printcp(oj_tree)
```

```
##
## Classification tree:
## rpart(formula = purchase ~ ., data = train_dat_oj, control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] list_price_diff loyal_ch        price_ch        price_diff
## [5] store           weekof_purchase
##
## Root node error: 312/800 = 0.39
##
## n= 800
##
##            CP nsplit rel error  xerror     xstd
## 1  0.51923077      0   1.00000 1.00000 0.044217
## 2  0.02243590      1   0.48077 0.48718 0.035564
## 3  0.01762821      3   0.43590 0.48397 0.035474
## 4  0.00801282      5   0.40064 0.48077 0.035384
## 5  0.00534188      7   0.38462 0.53526 0.036843
## 6  0.00480769     10   0.36859 0.56090 0.037477
## 7  0.00320513     14   0.34936 0.55449 0.037321
## 8  0.00160256     16   0.34295 0.57051 0.037706
## 9  0.00080128     18   0.33974 0.57372 0.037781
## 10 0.00000000     22   0.33654 0.57051 0.037706
```

```
plotcp(oj_tree)
```



```
minErr_oj <- which.min(cp_table_oj[,4])

# minimum cross-validation error
oj_tree_2 <- prune(oj_tree, cp = cp_table_oj[minErr_oj, 1])
rpart.plot(oj_tree_2)
```

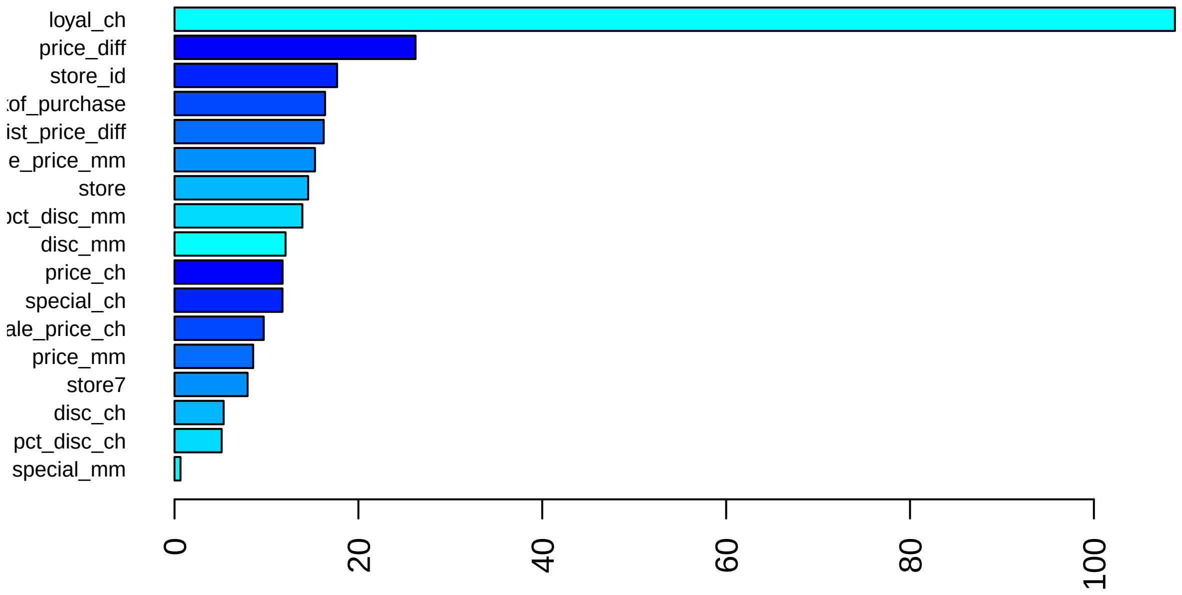


```r
### Building Confusion Matrix
test_oj_tree_prob <- predict(oj_tree_2, newdata = test_dat_oj, type = "prob")
test_oj_tree_prob <- test_oj_tree_prob[,1]
test_oj_tree_pred <- rep("CH", 270)
test_oj_tree_pred[test_oj_tree_prob > 0.5] <- "MM"

oj_tree_matrix <- caret::confusionMatrix(data = as.factor(test_oj_tree_pred),
                reference = test_dat_oj$purchase,
                positive = "MM")
```

Based off the confusion matrix, it has an accuracy of 0.1740741. Thus, it has an error of 0.8259259.

Part B

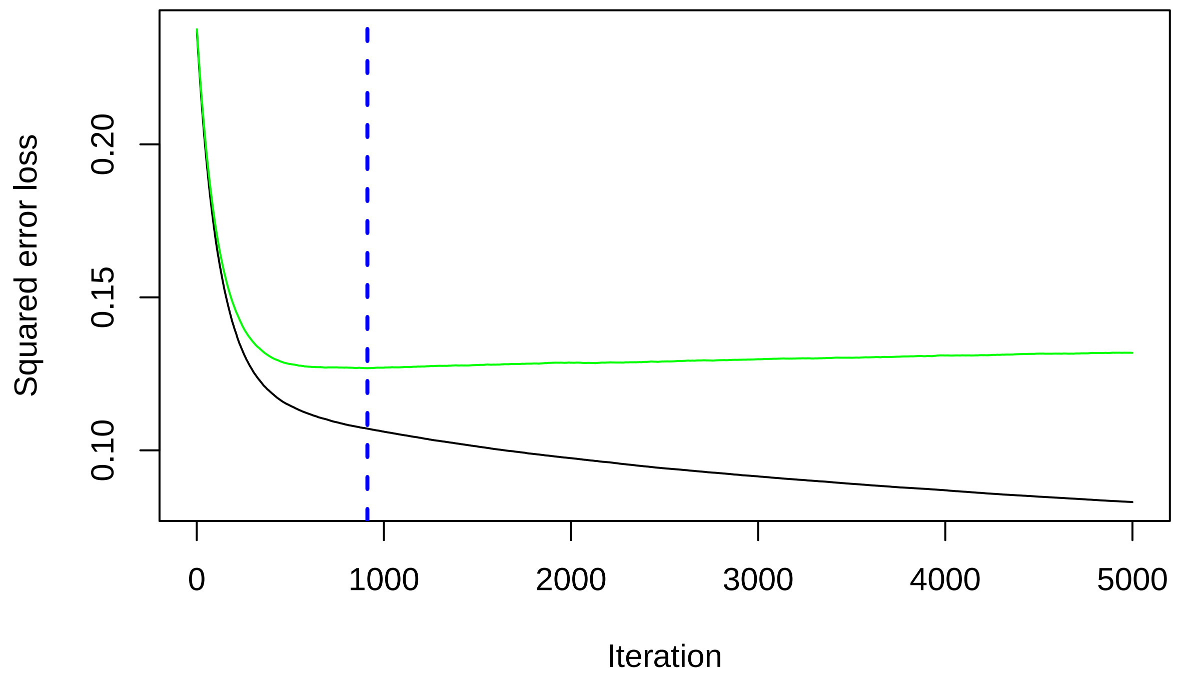
```r
set.seed(1)
rf_oj <- ranger(purchase ~., train_dat_oj,
                mtry = 6,
                min.node.size = 5,
                splitrule = "gini",
                importance = "permutation",
                scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf_oj), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(8))
```



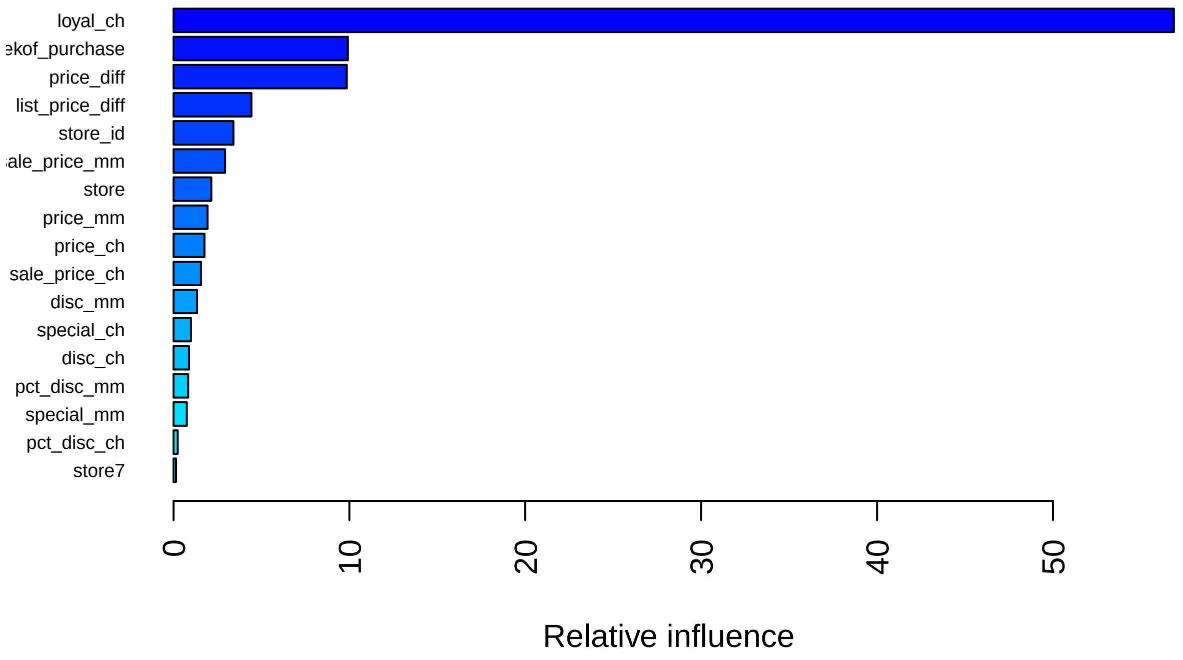
The customer brand loyalty for Citrus Hill orange juice, at a value of 108.8103408, has the highest variable importance. As you can see in the bar plot, it a much higher variable importance compared to the other variables.

Part C

```r
set.seed(1)

boosting_oj <- gbm(purchase ~., data = train_dat_oj,
                   distribution = "gaussian",
                   n.trees = 5000,
                   interaction.depth = 3,
                   shrinkage = 0.005,
                   cv.folds = 10)

nt_oj = gbm.perf(boosting_oj, method = "cv")
```

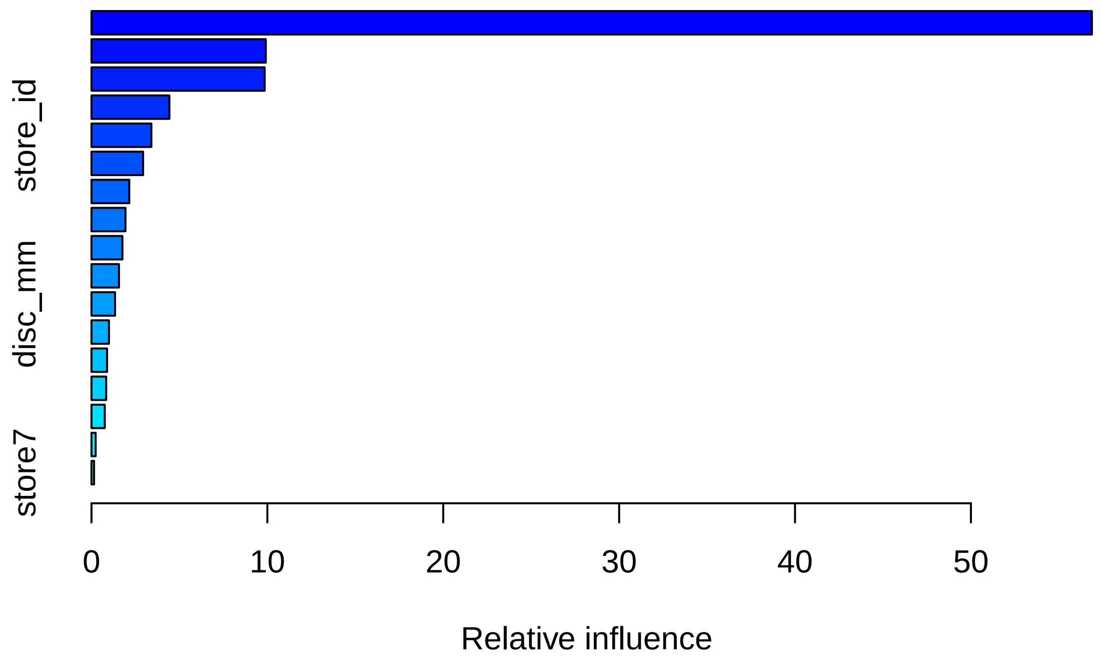


```r
summary(boosting_oj, las = 2, cBars = 19, cex.names = 0.6)
```



```
##                             var    rel.inf
## loyal_ch               loyal_ch 56.8781742
## weekof_purchase weekof_purchase  9.9104369
## price_diff           price_diff  9.8463273
## list_price_diff list_price_diff  4.4288931
## store_id               store_id  3.4056419
## sale_price_mm     sale_price_mm  2.9345853
## store                     store  2.1489733
## price_mm               price_mm  1.9319370
## price_ch               price_ch  1.7584702
## sale_price_ch     sale_price_ch  1.5642614
## disc_mm                 disc_mm  1.3409564
## special_ch           special_ch  0.9929761
## disc_ch                 disc_ch  0.8858828
## pct_disc_mm         pct_disc_mm  0.8351668
## special_mm           special_mm  0.7550823
## pct_disc_ch         pct_disc_ch  0.2361005
## store7                   store7  0.1461343
```

```r
boosting_imp_oj <- summary(boosting_oj)
```



The customer brand loyalty for Citrus Hill orange juice, at a value of 56.8781742, has the highest variable importance. As you can see in the bar plot, it a much higher variable importance compared to the other variables.The week of purchase variable, at a value of 9.9104369, has the 2nd highest variable importance.