Homework 4

Na Yun Cho

```
library(ISLR)
library(mlbench)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(rpart)
library(rpart.plot)
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
library(partykit)
```

```
## Loading required package: libcoin
```

```
##
## Attaching package: 'partykit'
```

```
## The following objects are masked from 'package:party':
##
##     cforest, ctree, ctree_control, edge_simple, mob, mob_control,
##     node_barplot, node_bivplot, node_boxplot, node_inner, node_surv,
##     node_terminal, varimp
```

```
library(plotmo)
```

```
## Loading required package: Formula

## Loading required package: plotrix

## Loading required package: TeachingDemos
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ranger)
```

```
##
## Attaching package: 'ranger'

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

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(pdp)
library(lasso2)
```

```
## R Package to solve regression problems while imposing
##   an L1 constraint on the parameters. Based on S-plus Release 2.1
## Copyright (C) 1998, 1999
## Justin Lokhorst   <jlokhors@stats.adelaide.edu.au>
## Berwin A. Turlach <bturlach@stats.adelaide.edu.au>
## Bill Venables     <wvenable@stats.adelaide.edu.au>
##
## Copyright (C) 2002
## Martin Maechler <maechler@stat.math.ethz.ch>
```

```
library(tidyverse) # data manipulation
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x stringr::boundary()    masks strucchange::boundary()
## x dplyr::combine()       masks randomForest::combine()
## x dplyr::filter()        masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::lift()          masks caret::lift()
## x randomForest::margin() masks ggplot2::margin()
## x purrr::partial()       masks pdp::partial()
```

```
library(ISLR) # data Problem 2
library(patchwork)
```

1(a)

Fit a regression tree with lpsa as the response and the other variables as predictors.

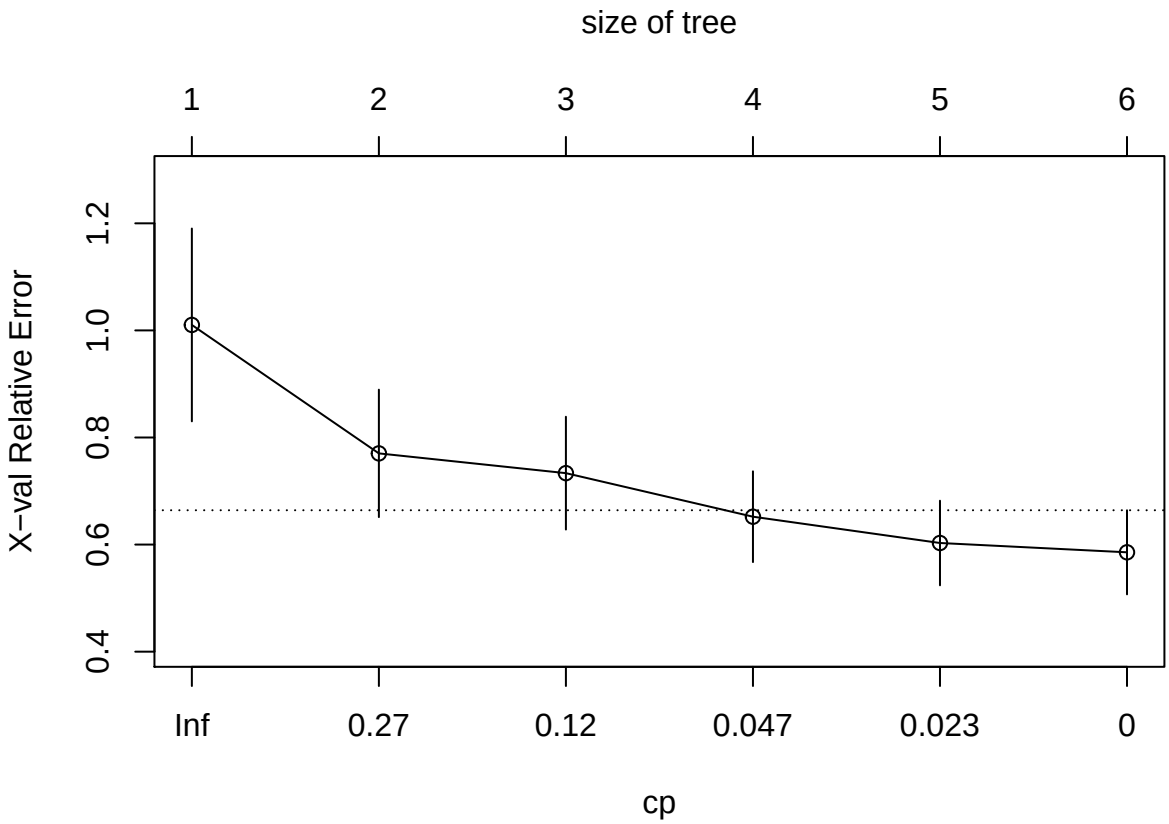
```
set.seed(1)
data(Prostate)
Prostate <-na.omit(Prostate)

# partition the dataset
trRows <- createDataPartition(Prostate$lpsa,
                              p =0.75, list =F)
```

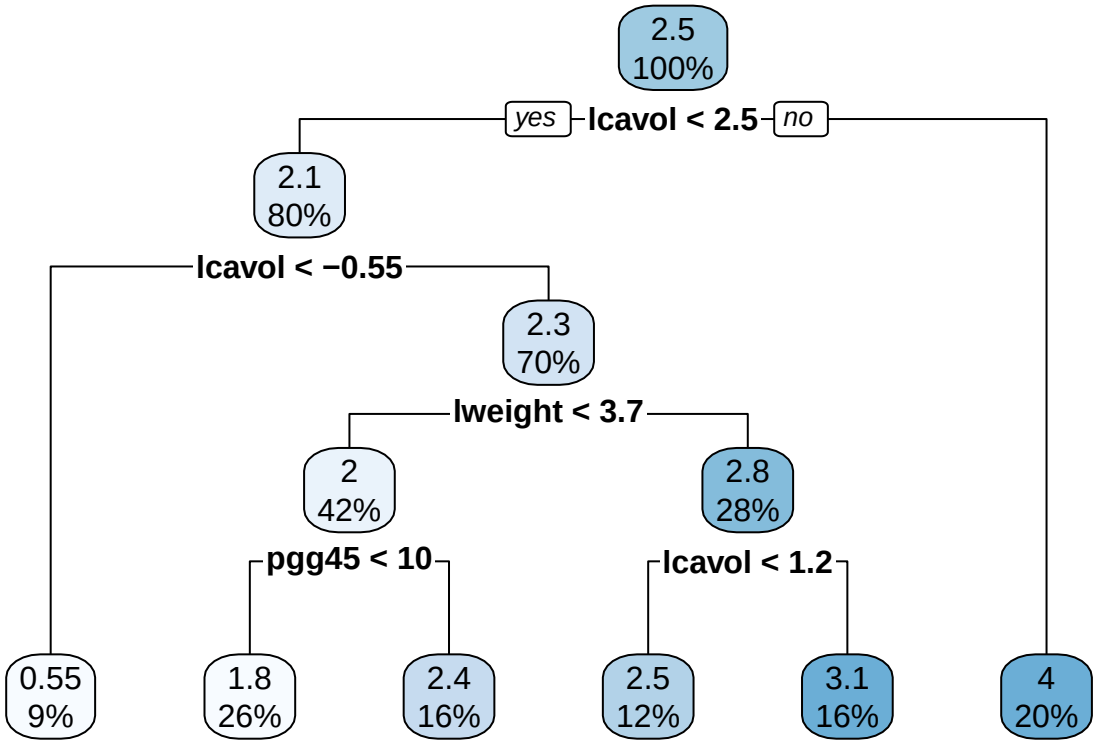
```
tree1 <- rpart(formula = lpsa ~ . ,
               data = Prostate, subset = trRows,
               control = rpart.control(cp = 0))
printcp(tree1)
```

```
##
## Regression tree:
## rpart(formula = lpsa ~ ., data = Prostate, subset = trRows, control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] lcavol  lweight pgg45
##
## Root node error: 106.7/74 = 1.4418
##
## n= 74
##
##         CP nsplit rel error  xerror     xstd
## 1 0.383415      0   1.00000 1.01023 0.180056
## 2 0.183917      1   0.61659 0.77032 0.119016
## 3 0.076423      2   0.43267 0.73339 0.105366
## 4 0.029412      3   0.35624 0.65205 0.084753
## 5 0.018601      4   0.32683 0.60292 0.079044
## 6 0.000000      5   0.30823 0.58555 0.078534
```

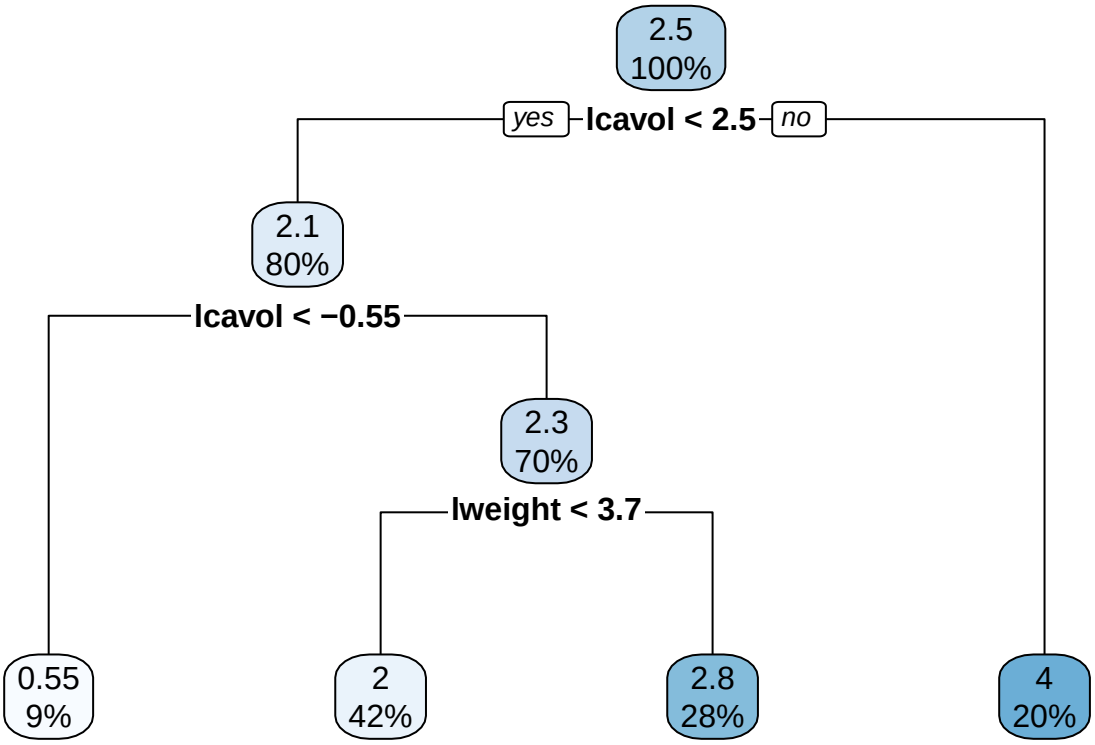
```
cpTable <- tree1$cptable
plotcp(tree1)
```



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

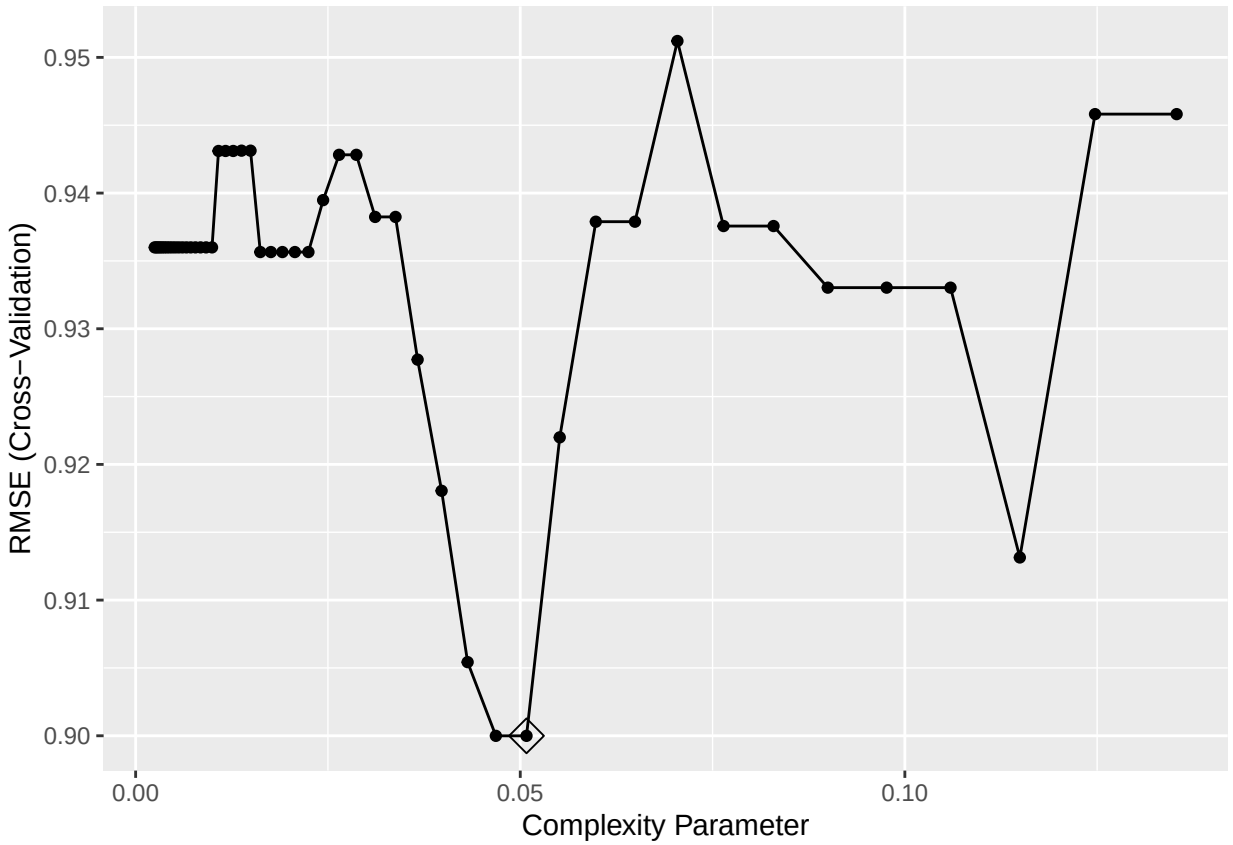



```
# tree using the 1SE rule
tree4 <- prune(tree1, cp = cpTable[cpTable[,4]<cpTable[minErr,4]+cpTable[minErr,5],1][1])
rpart.plot(tree4)
```

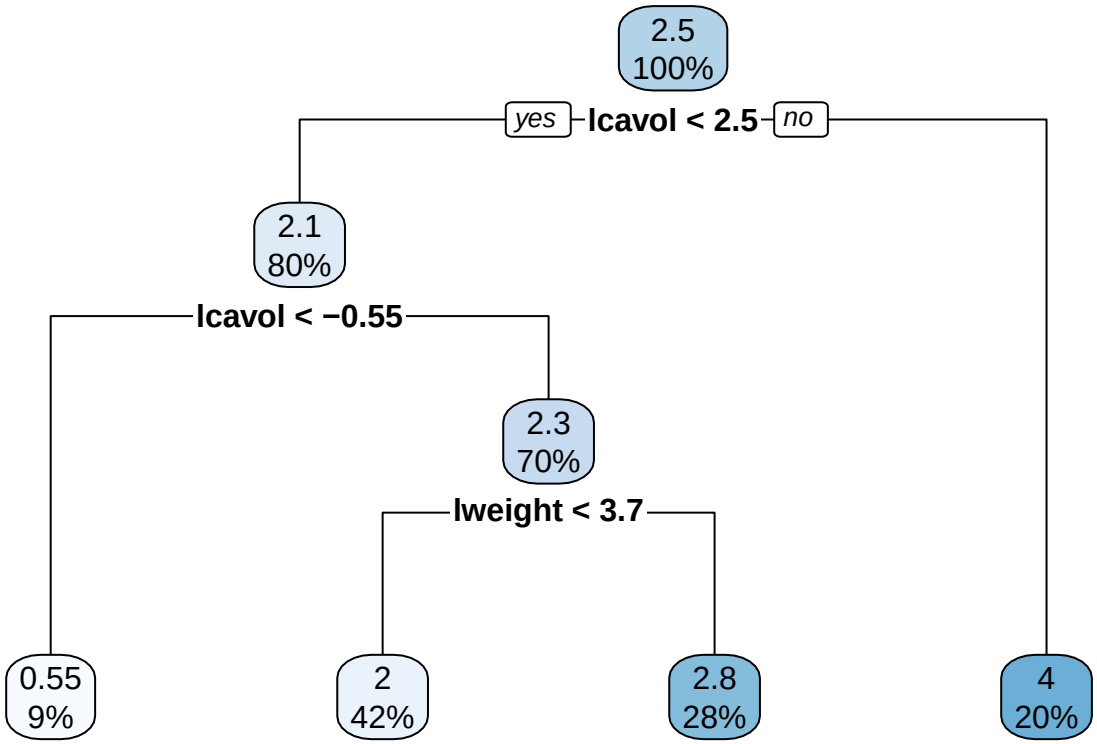


Tree size of 6 corresponds to the lowest cross-validation error. Tree size of 4 is obtained using the 1 SE rule. Thus, the sizes are different.

```
# use caret to do cross validation
set.seed(1)
ctrl <- trainControl(method = "cv")
rpart.fit <- train(lpsa~., Prostate[trRows,],
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-6,-2, length = 50))),
                   trControl = ctrl)
ggplot(rpart.fit, highlight = TRUE)
```




```
rpart.plot(rpart.fit$finalModel)
```



```
rpart.fit$bestTune
```

```
##            cp
## 38 0.05081357
```

The optimal tree size is 4.