Tree Regression - Disease Outbreak - Table 14.3

Survey data on outbreak of a disease carried by mosquitos.

Y = Person contracted disease or not X variables: Age, socioeconomic status (3 levels), sector of city (2 levels)

col3=1 if socio is Middle.

col4=1 if socio is Lower.

```
disout.data <- read.table(file="C:/Users/iverph01/Documents/Stat
327/KutnerData/Chapter 14 Data Sets/CH14TA03.txt",header=FALSE, col.names =
c('obsnum', 'age', 'col3', 'col4', 'sector', 'disease'))
attach (disout.data)
```

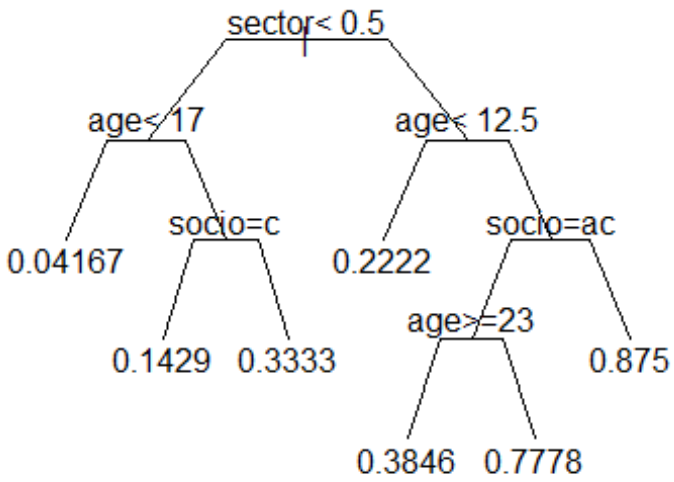
Fit tree regression model.

```
# Fitting the regression model:
disout.data$socio = ifelse (col3==0, ifelse (col4==0, "Upper", "Lower"),
"Middle")

attach (disout.data)

## The following objects are masked from disout.data (pos = 3):
##
##     age, col3, col4, disease, obsnum, sector

library (rpart)
disease.tree <- rpart(disease ~ age + socio + sector)
print(disease.tree)

## n= 98
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 98 21.1938800 0.31632650
##    2) sector< 0.5 59  8.3050850 0.16949150
##      4) age< 17 24  0.9583333 0.04166667 *
##      5) age>=17 35  6.6857140 0.25714290
##       10) socio=Upper 14  1.7142860 0.14285710 *
##       11) socio=Lower,Middle 21  4.6666670 0.33333330 *
##    3) sector>=0.5 39  9.6923080 0.53846150
##      6) age< 12.5 9  1.5555560 0.22222220 *
##      7) age>=12.5 30  6.9666670 0.63333330
##       14) socio=Lower,Upper 22  5.4545450 0.54545450
##         28) age>=23 13  3.0769230 0.38461540 *
```

```
##         29) age< 23 9  1.5555560 0.77777780 *
##       15) socio=Middle 8  0.8750000 0.87500000 *
```

Plot the tree:

```
par (mfrow=c(1,1))
plot(disease.tree, uniform = TRUE, margin = 0.1, branch = 0.5,
     compress = TRUE)
text(disease.tree)
```

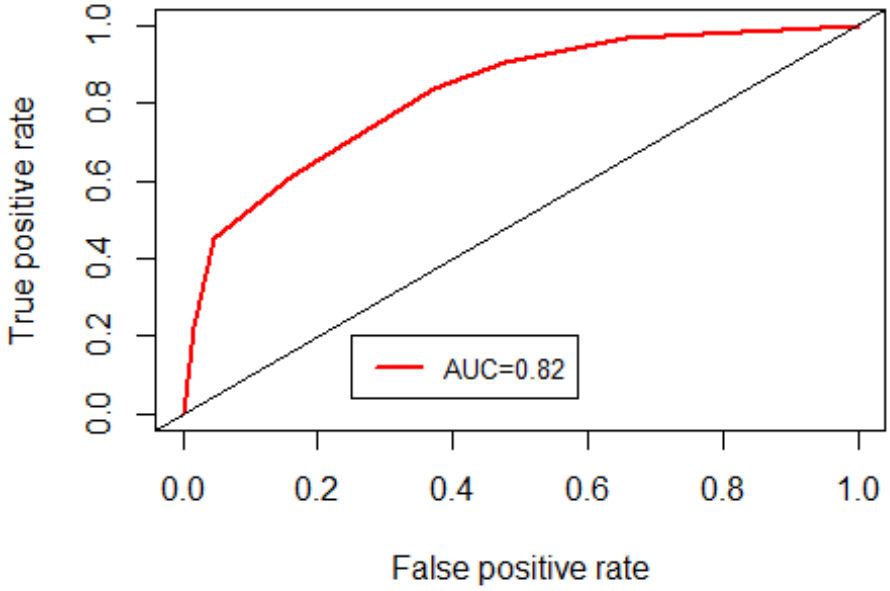


The tree above has means of disease status. Because disease status is 0-1 variable, these means are proportions of subjects in each subset that have the disease. They are also the predicted probability of disease for each subset.

ROC curve for regression tree:

```
#Put ROC curve statements in a function.

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.4.2

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.4.2
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

## This function is the same as the roc.logistic function, except for
## for how it gets the predicted values for a classification tree (example
below)

roc.tree = function (fit) {
  if (fit$method=="anova") {
    fitvals = predict(fit)
  }
  else {
    fitvals = predict(fit) [,2]
  }

  pred1 <- prediction(fitvals, fit$y)
  perf1 <- performance(pred1,"tpr","fpr")
  auc1 <- performance(pred1,"auc")@y.values[[1]]
  plot(perf1, lwd=2, col=2)
  abline(0,1)
  legend(0.25, 0.2, c(paste ("AUC=", round(auc1, 2), sep="")),
         cex=0.8, lwd=2, col=2)
  roc.table = cbind.data.frame (pred1@tn, pred1@fp, pred1@fn, pred1@tp,
                                pred1@cutoffs, perf1@x.values,
perf1@y.values)
  roc.table$spec = 1 - perf1@x.values[[1]]
  roc.table$ppv = pred1@tp[[1]] / (pred1@tp[[1]] + pred1@fp[[1]])
  roc.table$npv = pred1@tn[[1]] / (pred1@tn[[1]] + pred1@fn[[1]])
  roc.table$pctcorr = (pred1@tn[[1]] + pred1@tp[[1]]) /
    (pred1@tn[[1]] + pred1@tp[[1]] + pred1@fn[[1]] + pred1@fp[[1]])
  roc.table$optdist = sqrt ((perf1@x.values[[1]] - 0)^2 +
                            (perf1@y.values[[1]] - 1)^2)
  names (roc.table) = c("TN", "FP", "FN", "TP", "Cutoff", "FPR", "TPR",
"Spec",
                        "PPV", "NPV", "PctCorr", "OptDist")
  return (roc.table)
}

roc1 = roc.tree (disease.tree)
```



Optimal cutoffs:

```
roc1 [which.max (roc1$PctCorr)]
```

```
##    FN
##    31
## 51 24
## 56 17
## 60 12
## 98  5
## 46  3
## 74  1
## 97  0
```

```
roc1 [which.min (roc1$OptDist)]
```

```
##        Cutoff
##           Inf
## 51 0.87500000
## 56 0.77777778
## 60 0.38461538
## 98 0.33333333
## 46 0.22222222
## 74 0.14285714
## 97 0.04166667
```

Examine the cptable for this regression tree:

```
disease.tree$cptable
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.15082116      0 1.0000000 1.0153481 0.08105100
## 2 0.05520865      1 0.8491788 0.8665930 0.09610703
## 3 0.03442475      2 0.7939702 0.9370384 0.11379325
## 4 0.03119000      4 0.7251207 0.9517680 0.11423546
## 5 0.01437971      5 0.6939307 0.9511547 0.11334626
## 6 0.01000000      6 0.6795510 0.9719352 0.11600057
```

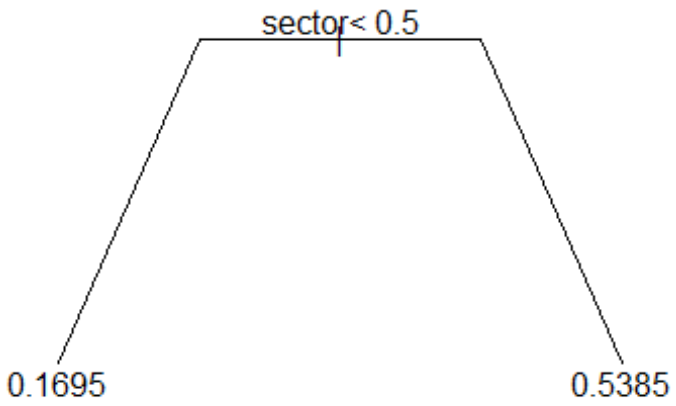
The xerror column has cross-validated prediction error for each value of nsplit (number of splits for each step in the tree-building process). The best tree in terms of prediction error is the one with the smallest xerror. This occurs in row 2, with just 1 split. In general, we can find this row as follows:

```
opt1 = which.min (disease.tree$cptable[,"xerror"])
opt1
```

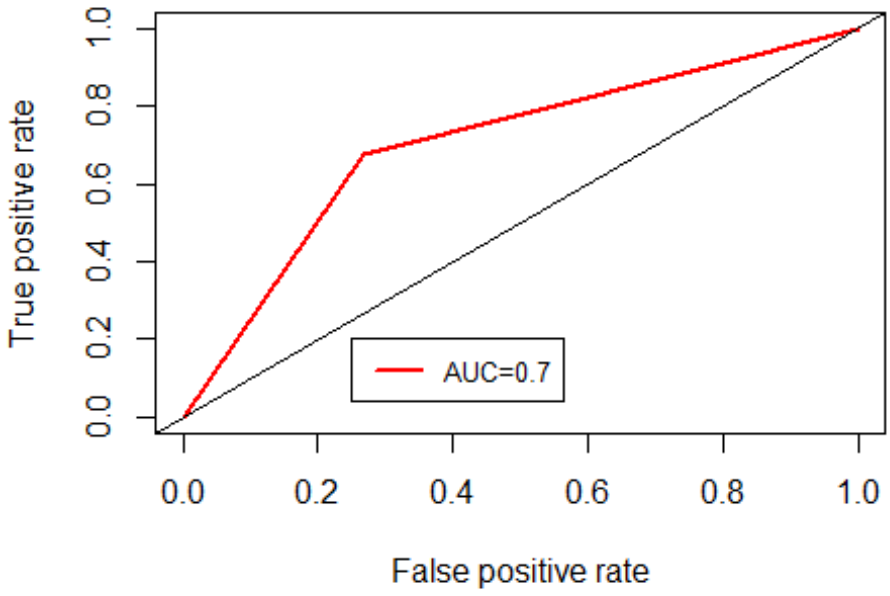
```
## 2
## 2
```

Then we can prune the tree using the CP value that corresponds to the minimum xerror tree:

```
disease.tree.pr = prune (disease.tree, cp= disease.tree$cptable [opt1, "CP"])
plot(disease.tree.pr, uniform = TRUE, margin = 0.1, branch = 0.5,
     compress = TRUE)
text(disease.tree.pr)
```




```
roc2 = roc.tree (disease.tree.pr)
```



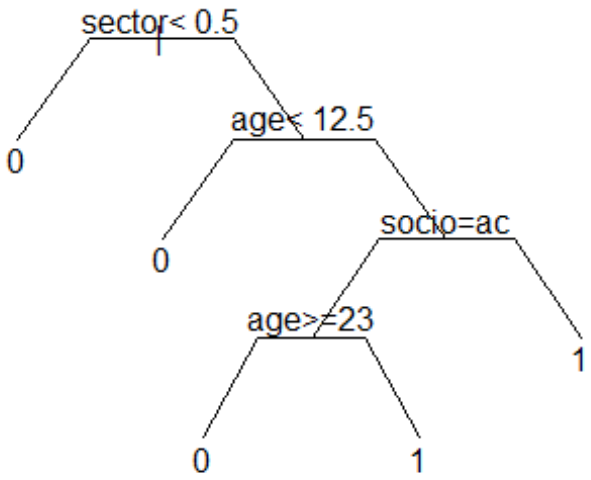
Classification tree:

```
dis.ctree <- rpart(disease ~ age + socio + sector, method='class')
print(dis.ctree)
```

```
## n= 98
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 98 31 0 (0.6836735 0.3163265)
##    2) sector< 0.5 59 10 0 (0.8305085 0.1694915) *
##    3) sector>=0.5 39 18 1 (0.4615385 0.5384615)
##      6) age< 12.5 9  2 0 (0.7777778 0.2222222) *
##      7) age>=12.5 30 11 1 (0.3666667 0.6333333)
##       14) socio=Lower,Upper 22 10 1 (0.4545455 0.5454545)
##         28) age>=23 13  5 0 (0.6153846 0.3846154) *
##         29) age< 23 9  2 1 (0.2222222 0.7777778) *
##       15) socio=Middle 8  1 1 (0.1250000 0.8750000) *
```

Plot the tree:

```
par (mfrow=c(1,1))
plot(dis.ctree, uniform = TRUE, margin = 0.1, branch = 0.5,
     compress = TRUE)
text(dis.ctree)
```



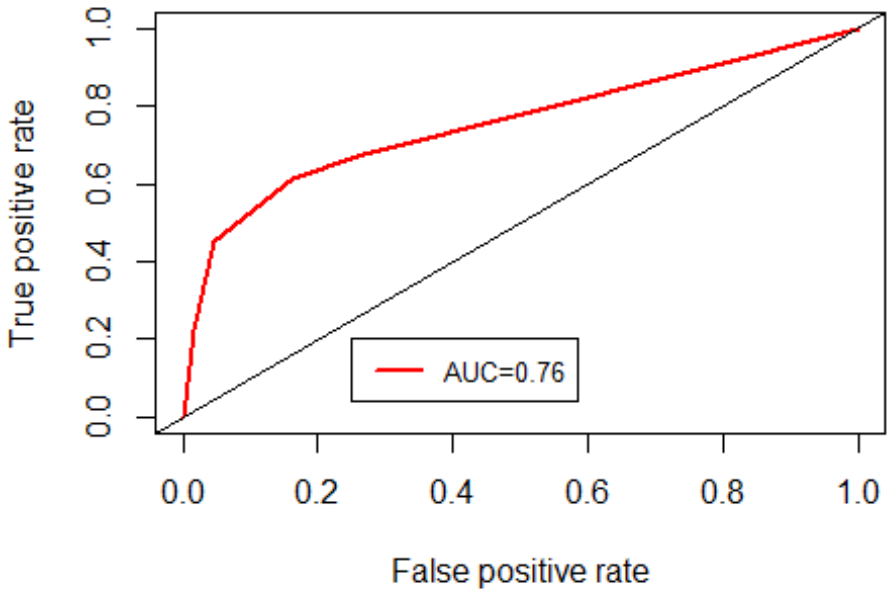
Examine the cptable for this classification tree:

```
dis.ctree$cptable
```

```
##          CP nsplit rel error    xerror      xstd
## 1 0.1290323      0 1.0000000 1.0000000 0.1485058
## 2 0.0483871      2 0.7419355 0.7741935 0.1373241
## 3 0.0100000      4 0.6451613 0.8064516 0.1392056
```

ROC curve for the classification tree

```
roc.results = roc.tree (dis.ctree)
```

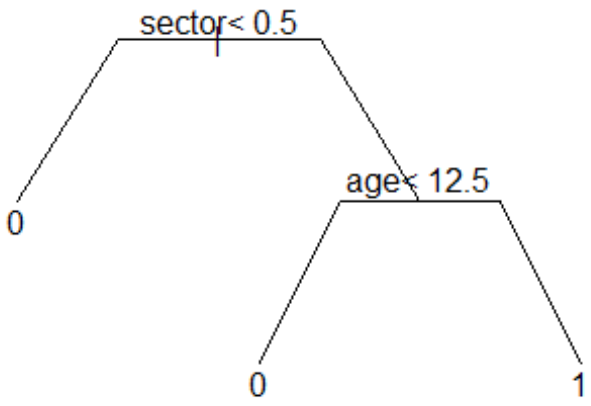


Prune the classification tree:

```
opt2 = which.min (dis.ctree$cptable [,"xerror"])
opt2
```

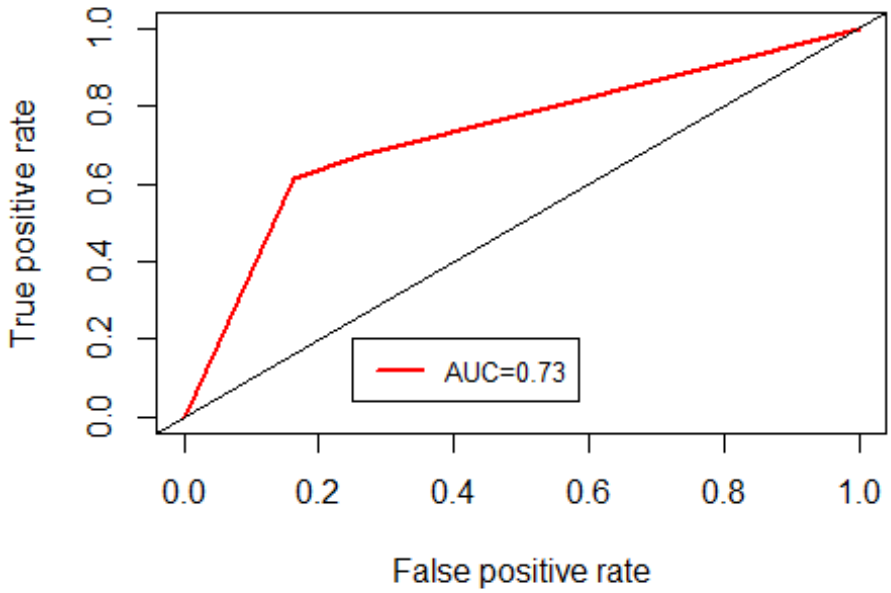
```
## 2
## 2
```

```
dis.ctree.pr = prune (dis.ctree, cp=dis.ctree$cptable [opt2, "CP"])
plot(dis.ctree.pr, uniform = TRUE, margin = 0.1, branch = 0.5,
     compress = TRUE)
text(dis.ctree.pr)
```



ROC curve for pruned classification tree:

```
roc3 = roc.tree (dis.ctree.pr)
```



```
roc3
```

```
##   TN FP FN TP    Cutoff       FPR       TPR      Spec       PPV       NPV
## 1 67  0 31  0       Inf 0.0000000 0.0000000 1.0000000       NaN 0.6836735
## 2 56 11 12 19 0.6333333 0.1641791 0.6129032 0.8358209 0.6333333 0.8235294
## 3 49 18 10 21 0.2222222 0.2686567 0.6774194 0.7313433 0.5384615 0.8305085
## 4  0 67  0 31 0.1694915 1.0000000 1.0000000 0.0000000 0.3163265       NaN
##     PctCorr   OptDist
## 1 0.6836735 1.0000000
## 2 0.7653061 0.4204744
## 3 0.7142857 0.4198032
## 4 0.3163265 1.0000000
```