# R Exercise 3: INTA 6450 - Data Analytics and Security

## R Exercise 3

**Prompt**

1. Complete the following exercise:

- RStudio exercise `trees.r`

2. You may make changes to the code to explore how they work
3. Then, submit the output for running these commands. You may copy/paste output from your terminal into a word or text document. You can convert the notebook to PDF submission:File->Compile Report->Report output format as HTML/PDF.
4. Write 100-150 words describing one extension you made to the code. You may post the change description as a comment at the beginning of your code, or submit the write-up in a separate document. Make sure to clearly mark where in your code the changes were made either by commenting, highlighting, or noting the command lines where the changes occur.

**Submissions**

You will submit the following:

1. Original code output (R file code and output)
2. Your code change and output from that code change (comment where your code changes start and end)
3. Your code change description (this should be between 100 to 150 words)

An example of what the change description should look like:

```
##############################
# CHANGE DESCRIPTION
# 100-150 words on the changes I've made, including a reference to which
# initial command I've changed >
# END OF CHANGE DESCRIPTION
##############################
```

Files must be in one of the following formats: pdf, doc, html

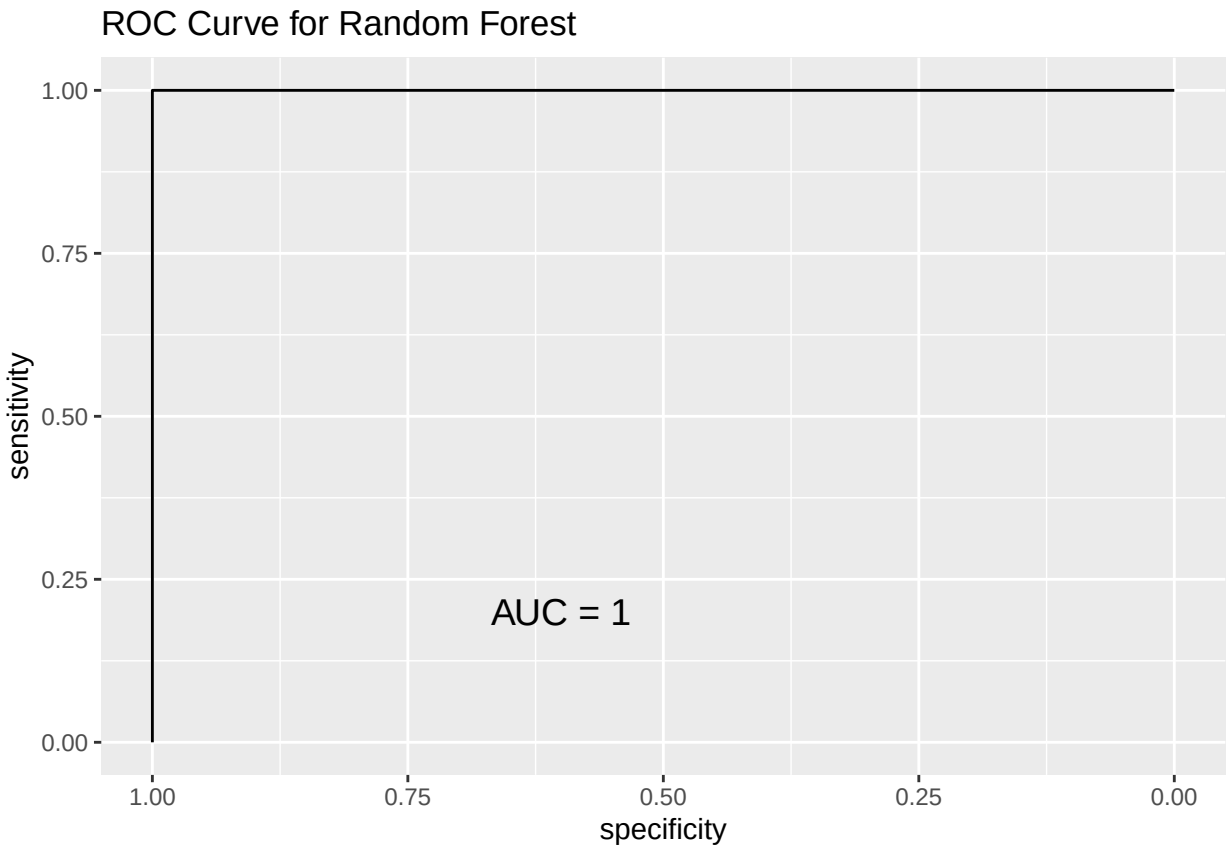
*Please note some of these exercises have graphs/charts - be sure when you export the original output to also check that the file includes graphs/charts that may have been present. If it does not, you will need to re-export or manually include the graphs in your final pdf, doc or html file*

## Solution Summary

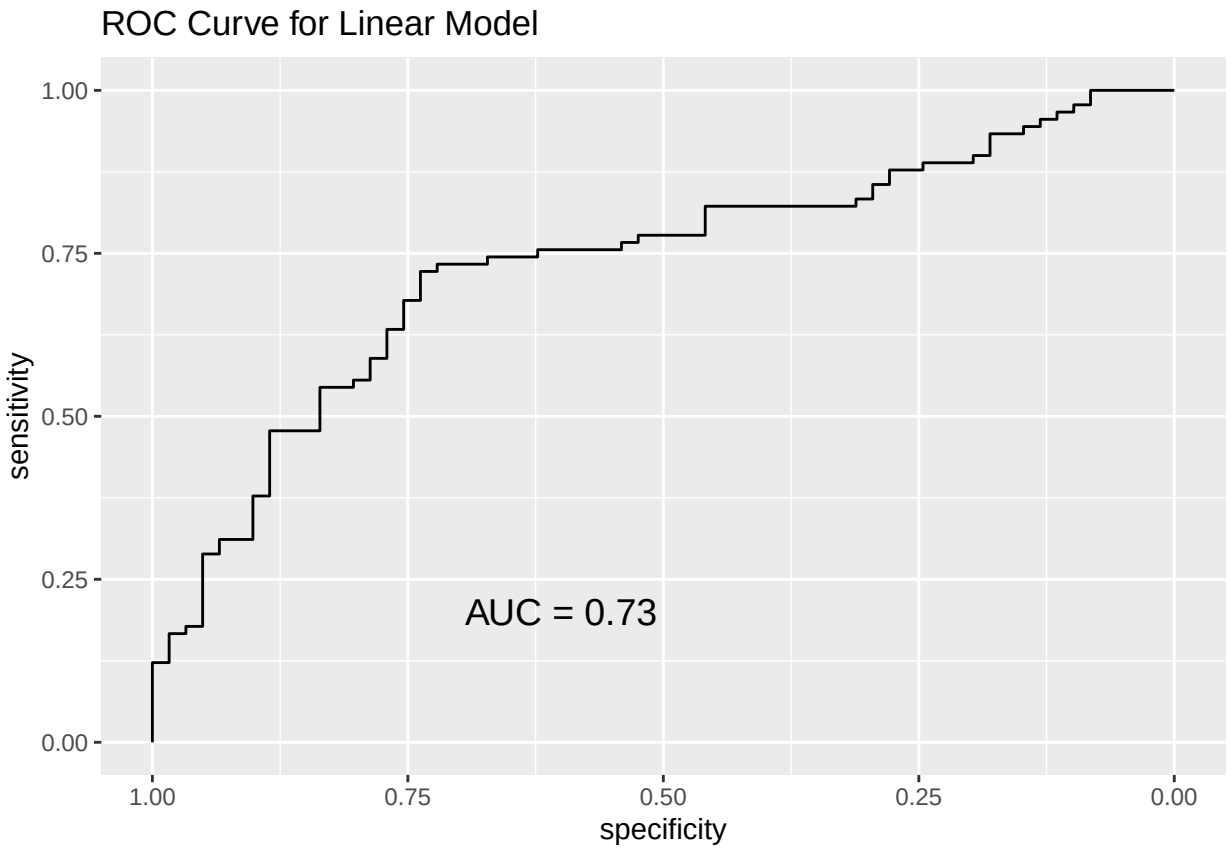
Receiver Operating Characteristic (ROC) curves are useful to evaluate performance of binary classification models. Code changes allowed for plotting ROC curves for different models to compare their performance visually. An additional function is added to allow for creating a ROC curve plot using `ggplot2`. The function takes the true and predicted data as arguments and generates a ROC curve plot. Area Under the Curve (AUC) is a single scalar value that summarizes the ROC curve. The AUC value is calculated and displayed on the plot. The plot is then saved to a file if a save path is provided. This allows for comparing models by looking at their AUC scores or by observing which curve is closer to the top-left corner. The code changes are attached below and followed up with the original code output.

### Code Changes

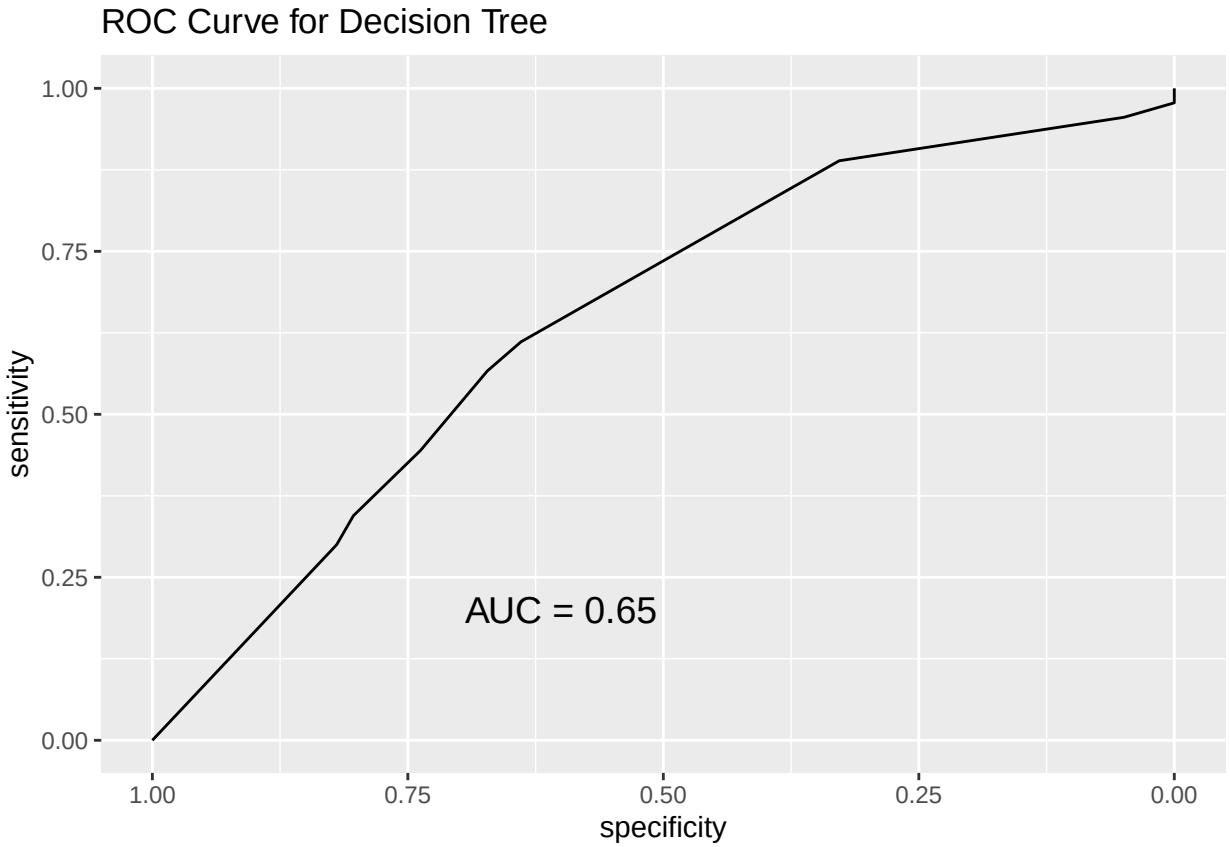
```
library(ggplot2)

plot_roc_curve <- function(y_true, y_predicted, model_name, save_path=NULL) {
  # Compute the ROC curve
  curve <- roc(y_true, y_predicted)

  # Calculate the AUC value
  auc_value <- auc(curve)

  # Create the ROC plot using ggplot2
  plot <- ggroc(curve) +
    ggtitle(paste("ROC Curve for", model_name)) +
    annotate("text", x = 0.6, y = 0.2, label = paste("AUC =", round(auc_value, 2)), size = 5)

  # Print the plot to the console
  print(plot)

  # Save the plot to a file if a save path is provided
  if (!is.null(save_path)) {
    ggsave(filename = save_path, plot = plot)
  }
}

# Plot ROC curves for the models and save them to files
plot_roc_curve(lfp.out$inlf, predict(rf, newdata=lfp.out, type="prob")[,1],
  "Random Forest",
  "random_forest_roc.png")
```




```
plot_roc_curve(lfp.out$inlf, predict(inlf.lm, newdata=lfp.out),
  "Linear Model",
  "linear_model_roc.png")
```




```
plot_roc_curve(lfp.out$inlf, predict(inlf.tree, newdata=lfp.out),
  "Decision Tree",
  "decision_tree_roc.png")
```


ROC Curve for Decision Tree
1.00
0.75
0.50
0.25
0.00
sensitivity
AUC = 0.65
1.00
0.75
0.50
0.25
0.00
specificity

## Original Code

```
# Instrumental Variables example
# install.packages("sem")
library(sem)
wages<-read.csv('http://inta.gatech.s3.amazonaws.com/wage2.csv')
iv.results<-tsls(lwage ~ educ + age + married + black, ~ feduc + age + married + black, data=wages)
# Run an instrumental variables regression. Note that father's education is
# used for an instrument, and is not in the first set of variables. Age,
# marital, status, and race are assumed to be less related to underlying
# ability and so are controlled for in the "first stage" regression too.

ols.results<-lm(lwage ~ educ + age + married + black, data=wages)
# Run the analogous OLS regression without father's education

print(summary(ols.results))
```

```
##
## Call:
## lm(formula = lwage ~ educ + age + married + black, data = wages)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.91980 -0.24114  0.01988  0.25465  1.31126
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.231197   0.159806  32.735  < 2e-16 ***
## educ         0.056040   0.005820   9.629  < 2e-16 ***
## age          0.019539   0.004062   4.810 1.76e-06 ***
## married      0.194424   0.040952   4.748 2.38e-06 ***
## black       -0.209969   0.038208  -5.495 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3834 on 930 degrees of freedom
## Multiple R-squared:  0.1747, Adjusted R-squared:  0.1711
## F-statistic: 49.21 on 4 and 930 DF,  p-value: < 2.2e-16
```

```
print(summary(iv.results))
```

```
##
##  2SLS Estimates
##
## Model Formula: lwage ~ educ + age + married + black
##
## Instruments: ~feduc + age + married + black
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.85005 -0.24061  0.02436  0.00000  0.26225  1.34065
##
```

```
##                Estimate  Std. Error  t value   Pr(>|t|)
## (Intercept)  4.62589626  0.26729720 17.30619 < 2.22e-16 ***
## educ         0.09640575  0.01586492  6.07666 1.9681e-09 ***
## age          0.02106442  0.00472040  4.46242 9.3723e-06 ***
## married      0.20134756  0.04651945  4.32824 1.7109e-05 ***
## black       -0.11997207  0.05366054 -2.23576   0.025667 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3944321 on 736 degrees of freedom
```

```r
# Note that the point estimate for education has now gone up. This suggests
# that people who have higher education in a way that's correlated with
# their father having higher education, even holding fixed age, marital
# status, and race, receive ~9% higher wages for every year of education
# Also note that the standard error on education is higher for the iv than
# ols results. The reason for this is that in IV, we're using only some of
# the variation in education: only the part related to father's education,
# so it's like there's less variation overall

# Stepwise regression and tree example
# install.packages(c('MASS','sem'))
library(MASS)
start.model<-lm(wage ~ hours + IQ + KWW + educ + exper + tenure + age + married + black + south + urban
# Give an initial model, which will be the most coefficients we'd want to ever use
summary(start.model)
```

```
##
## Call:
## lm(formula = wage ~ hours + IQ + KWW + educ + exper + tenure +
##     age + married + black + south + urban + sibs, data = wages)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -849.76 -223.19  -40.18  172.09 2091.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -562.589    185.142  -3.039  0.00244 **
## hours         -3.485      1.621  -2.150  0.03182 *
## IQ             2.828      1.004   2.817  0.00495 **
## KWW            5.261      1.981   2.656  0.00804 **
## educ          48.213      7.182   6.713 3.33e-11 ***
## exper          9.774      3.589   2.723  0.00658 **
## tenure         5.242      2.406   2.178  0.02963 *
## age            5.046      4.930   1.024  0.30634
## married      170.231     37.883   4.494 7.89e-06 ***
## black       -108.844     39.808  -2.734  0.00637 **
## south        -53.689     25.536  -2.102  0.03578 *
## urban        160.652     26.200   6.132 1.29e-09 ***
## sibs          -1.068      5.455  -0.196  0.84488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 352.6 on 922 degrees of freedom
## Multiple R-squared:  0.2493, Adjusted R-squared:  0.2395
## F-statistic: 25.51 on 12 and 922 DF,  p-value: < 2.2e-16
```

```
stepwise.model<- step(start.model)
```

```
## Start:  AIC=10981.26
## wage ~ hours + IQ + KWW + educ + exper + tenure + age + married +
##     black + south + urban + sibs
##
##           Df Sum of Sq       RSS   AIC
## - sibs     1      4763 114655843 10979
## - age      1    130265 114781346 10980
## <none>                 114651080 10981
## - south    1    549683 115200763 10984
## - hours    1    574799 115225880 10984
## - tenure   1    590101 115241181 10984
## - KWW      1    877455 115528536 10986
## - exper    1    922291 115573372 10987
## - black    1    929659 115580739 10987
## - IQ       1    986880 115637960 10987
## - married  1   2510929 117162009 11000
## - urban    1   4675218 119326298 11017
## - educ     1   5603726 120254807 11024
##
## Step:  AIC=10979.3
## wage ~ hours + IQ + KWW + educ + exper + tenure + age + married +
##     black + south + urban
##
##           Df Sum of Sq       RSS   AIC
## - age      1    128704 114784547 10978
## <none>                 114655843 10979
## - south    1    546953 115202796 10982
## - hours    1    575305 115231148 10982
## - tenure   1    590732 115246575 10982
## - KWW      1    911607 115567450 10985
## - exper    1    926789 115582632 10985
## - black    1   1000335 115656178 10985
## - IQ       1   1002450 115658294 10985
## - married  1   2508266 117164110 10998
## - urban    1   4686063 119341906 11015
## - educ     1   5673927 120329770 11022
##
## Step:  AIC=10978.35
## wage ~ hours + IQ + KWW + educ + exper + tenure + married + black +
##     south + urban
##
##           Df Sum of Sq       RSS   AIC
## <none>                 114784547 10978
## - hours    1    562954 115347501 10981
## - south    1    565737 115350283 10981
## - tenure   1    680663 115465209 10982
## - IQ       1    907667 115692213 10984
## - black    1    980306 115764853 10984
```

```
## - KWW      1   1413125 116197671 10988
## - exper    1   1678662 116463208 10990
## - married  1   2538972 117323519 10997
## - urban    1   4639497 119424043 11013
## - educ     1   6097302 120881849 11025
```

```
# The command "step" adds and subtracts coefficients to maximize a measure of
# goodness of fit, by default AIC
summary(stepwise.model)
```

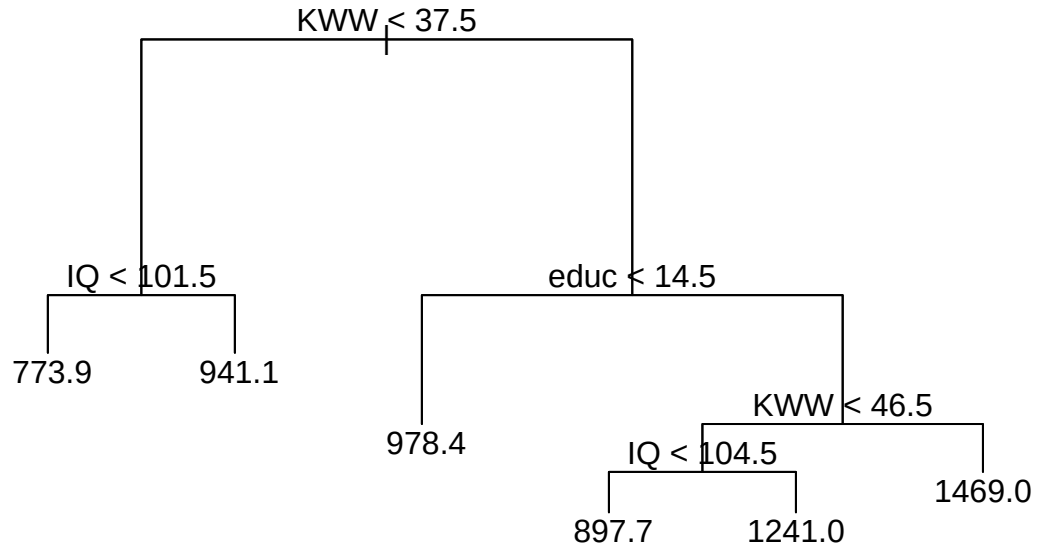
```
##
## Call:
## lm(formula = wage ~ hours + IQ + KWW + educ + exper + tenure +
##     married + black + south + urban, data = wages)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -872.52 -224.26  -41.71  171.08 2086.14
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -453.2969   141.0390  -3.214 0.001354 **
## hours         -3.4474     1.6194  -2.129 0.033536 *
## IQ             2.6621     0.9848   2.703 0.006996 **
## KWW            6.0918     1.8062   3.373 0.000775 ***
## educ          49.4804     7.0627   7.006 4.73e-12 ***
## exper         11.5519     3.1425   3.676 0.000251 ***
## tenure         5.5777     2.3828   2.341 0.019455 *
## married      171.1045    37.8476   4.521 6.96e-06 ***
## black       -109.2993    38.9083  -2.809 0.005072 **
## south        -54.4073    25.4951  -2.134 0.033103 *
## urban        159.8942    26.1639   6.111 1.46e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 352.5 on 924 degrees of freedom
## Multiple R-squared:  0.2484, Adjusted R-squared:  0.2402
## F-statistic: 30.53 on 10 and 924 DF,  p-value: < 2.2e-16
```

```
# install.packages('tree')
library(tree)
wage.tree <- tree(wage ~ married + hours + IQ + KWW + educ + tenure + exper, method="anova", data=wages)
# fit a tree
summary(wage.tree)
```

```
##
## Regression tree:
## tree(formula = wage ~ married + hours + IQ + KWW + educ + tenure +
##     exper, data = wages, method = "anova")
## Variables actually used in tree construction:
## [1] "KWW"  "IQ"   "educ"
## Number of terminal nodes:  6
## Residual mean deviance:  130000 = 120800000 / 929
```

```
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -891.70 -250.10  -27.35    0.00  188.60 2137.00
```

```
# Print the results

plot(wage.tree)
text(wage.tree)
```




```
# Plot the tree, and add the text decisions

cross.validation <- cv.tree(wage.tree)
# perform cross-validation, 10-fold.

cross.validation
```

```
## $size
## [1] 6 5 4 3 2 1
##
## $dev
## [1] 129725281 132193357 133522601 134143608 141258892 153594136
##
## $k
## [1]     -Inf  2430505  2880143  3466767  7749956 15376573
##
```

```
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
# look for the size with the lowest "dev" or deviance
# Why might this be different than the fitted tree?

pruned.wage.tree<-prune.tree(wage.tree,best=5)
# Prune this tree down to 5 terminal nodes, just to show this is how you would
# do it. Here this makes it worse though
summary(pruned.wage.tree)
```

```
##
## Regression tree:
## snip.tree(tree = wage.tree, nodes = 14L)
## Variables actually used in tree construction:
## [1] "KWW"  "IQ"   "educ"
## Number of terminal nodes:  5
## Residual mean deviance:  132500 = 123200000 / 930
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -891.70 -250.80  -26.53    0.00  190.40 2137.00
```

```
lmfit<- lm(wage ~ married + hours + IQ + KWW + educ + tenure + exper, data=wages)
# Compute a regression using those same variables:
anova(lmfit)
```

```
## Analysis of Variance Table
##
## Response: wage
##            Df    Sum Sq  Mean Sq  F value    Pr(>F)
## married     1   2848893  2848893  21.7647 3.535e-06 ***
## hours       1     29759    29759   0.2273 0.6336108
## IQ          1  14964264 14964264 114.3225 < 2.2e-16 ***
## KWW         1   6619303  6619303  50.5695 2.297e-12 ***
## educ        1   4117977  4117977  31.4601 2.687e-08 ***
## tenure      1   1277563  1277563   9.7602 0.0018387 **
## exper       1   1518568  1518568  11.6014 0.0006873 ***
## Residuals 927 121339841   130895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(wage.tree)
```

```
##
## Regression tree:
## tree(formula = wage ~ married + hours + IQ + KWW + educ + tenure +
##     exper, data = wages, method = "anova")
## Variables actually used in tree construction:
## [1] "KWW"  "IQ"   "educ"
```

```
## Number of terminal nodes:  6
## Residual mean deviance:  130000 = 120800000 / 929
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -891.70 -250.10  -27.35    0.00  188.60 2137.00
```

```
# Notice that the mean sum of squared residuals was 130895 using the linear
# model, compared to the residual mean deviance of 130000 using trees. So the
# tree method has less residual error, which means it's a better predictor.
# This is especially striking given that the tree only uses KWW, educ, IQ,
# While the regression needs substantial contributions from tenure, experience,
# and marital status too.

predict(wage.tree, newdata=data.frame(IQ=100, KWW=50, educ=16, married=1, hours=40, tenure=3, exper=2))
```

```
##        1
## 1468.696
```

```
predict(lmfit, newdata=data.frame(IQ=100, KWW=50, educ=16, married=1, hours=40, tenure=3, exper=2))
```

```
##        1
## 1089.293
```

```
# These predictors give different estimates for the expected wage of a
# particular individual, relatively young, married, well educated, with good
# knowledge of the world of work. Which would you trust and why?

# Extension: try to build another tree, and see what it looks like.
# You can sample the data with the command:
# install.packages('dplyr')
# library('dplyr')
# wages.sample <- sample(wages, n=500)

# Takehome exercise from last class
lfp <- read.csv('http://inta.gatech.s3.amazonaws.com/mroz_train.csv')
lfp$inlf<-as.factor(lfp$inlf) # We need the outcome variable to be a 'factor'

inlf.tree <-tree(as.numeric(inlf) - 1 ~huseduc + husage + kidslt6 + kidsge6 + nwifeinc + educ + age, da
inlf.lm <- lm(as.numeric(inlf) - 1 ~huseduc + husage + kidslt6 + kidsge6 + nwifeinc + educ + age, data=
print(inlf.tree)
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 602 148.2000 0.56150
##    2) kidslt6 < 0.5 478 113.8000 0.60880
##      4) husage < 48.5 273  55.2800 0.71790
##        8) nwifeinc < 27.936 224  40.4600 0.76340
##         16) educ < 9.5 19   4.7370 0.47370 *
##         17) educ > 9.5 205  33.9800 0.79020
##           34) kidsge6 < 2.5 153  20.2400 0.84310 *
##           35) kidsge6 > 2.5 52  12.0600 0.63460 *
```

```
##        9) nwifeinc > 27.936 49  12.2400 0.51020
##         18) huseduc < 12.5 11   0.9091 0.09091 *
##         19) huseduc > 12.5 38   8.8420 0.63160
##           38) nwifeinc < 31.85 11   2.1820 0.27270 *
##           39) nwifeinc > 31.85 27   4.6670 0.77780 *
##      5) husage > 48.5 205  50.9800 0.46340
##       10) educ < 12.5 155  37.2000 0.40000 *
##       11) educ > 12.5 50  11.2200 0.66000 *
##    3) kidslt6 > 0.5 124  29.1900 0.37900 *
```

```
# compute predictions manually, and save them as lfp$predicted.inlf
lfp$predicted.inlf<-predict(inlf.tree)


library(pROC)
evaluate <- function(y_true, y_predicted, detail=FALSE) {
  curve<-roc(y_true, y_predicted)
  if (detail) {
  print(ci.auc(curve))
  print('Sensitivity (True Positive Rate)')
  tpr<-ci.se(curve)
  print(tpr)
  print('Specificity (True Negative Rate)')
  tnr<-ci.sp(curve)
  print(tnr)
  }
  cat('Point estimate (final word on effectiveness)\n')
  print(auc(curve))
  #print('Point estimate of AUCROC: ' + auc(curve))
}
evaluate(lfp$inlf, lfp$predicted.inlf)
```

```
## Point estimate (final word on effectiveness)
## Area under the curve: 0.7283
```

```
library(randomForest)
lfp <- read.csv('http://inta.gatech.s3.amazonaws.com/mroz_train.csv')
lfp[is.na(lfp)] <- 0
rf <-randomForest(as.factor(inlf) ~ hours  +  kidslt6  , data=lfp)
evaluate(as.factor(lfp$inlf), predict(rf,type="prob")[,1])
```

```
## Point estimate (final word on effectiveness)
## Area under the curve: 1
```

```
lfp.out <- read.csv('http://inta.gatech.s3.amazonaws.com/mroz_test.csv')
lfp.out[is.na(lfp.out)] <- 0
evaluate(as.factor(lfp.out$inlf), predict(rf, newdata=lfp.out, type="prob")[,1])
```

```
## Point estimate (final word on effectiveness)
## Area under the curve: 1
```

```
evaluate(lfp.out$inlf, predict(inlf.lm, newdata=lfp.out))
```

```
## Point estimate (final word on effectiveness)
## Area under the curve: 0.7322
```

```
evaluate(lfp.out$inlf, predict(inlf.tree, newdata=lfp.out))
```

```
## Point estimate (final word on effectiveness)
## Area under the curve: 0.6488
```