INDE498_HW2

Steven Hwang, Haena Kim, Victoria Diaz

Chapter 2, Exercise 3

Pick up any dataset you have used, and randomly split the data into two halves. Use one half to build the tree model and the regression model. Test the models' prediction performances on the second half. Report what you have found, adjust your way of model building, and suggest a strategy to find the model you consider as the best.

```
#data set from Chapter 2, Exercise 1
library(RCurl)
```

```
## Loading required package: bitops
```

```
AD <- read.csv(text=getURL("https://raw.githubusercontent.com/shuailab/ind_498/master/resource/data/AD2
AD$ID = c(1:dim(AD)[1])
str(AD)
```

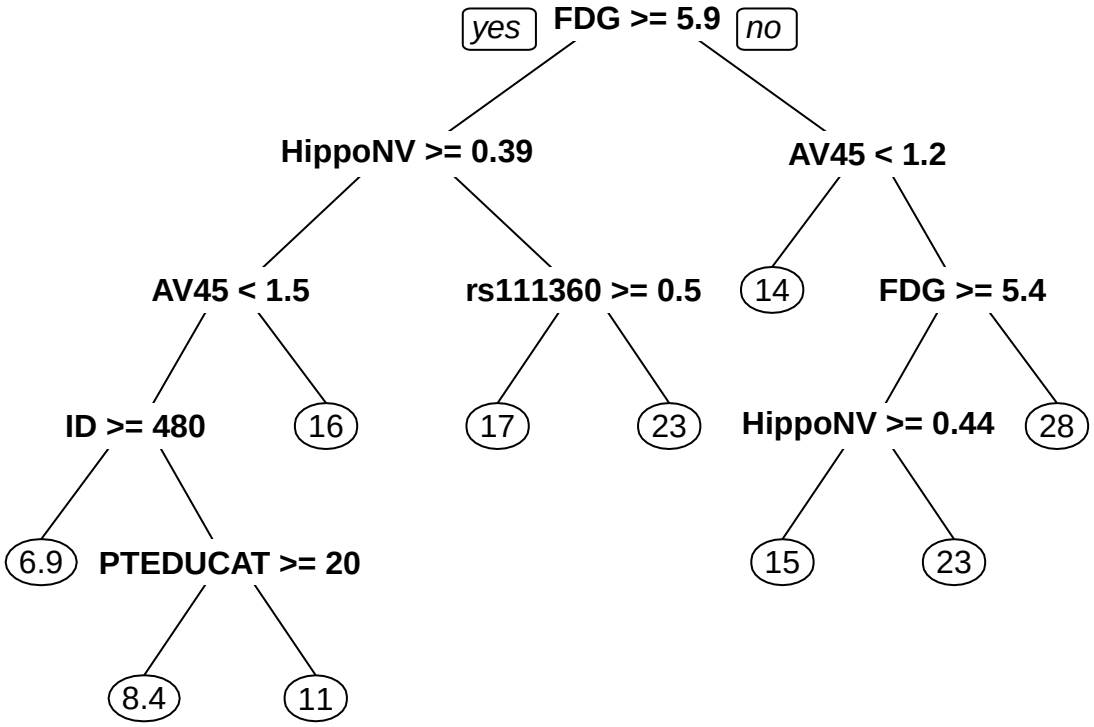
```
## 'data.frame':    517 obs. of  18 variables:
##  $ AGE       : num  71.7 77.7 72.8 69.6 70.9 65.1 79.6 73.6 60.7 70.6 ...
##  $ PTGENDER  : int  2 1 2 1 1 2 2 2 1 2 ...
##  $ PTEDUCAT  : int  14 18 18 13 13 20 20 18 19 18 ...
##  $ FDG       : num  6.82 6.37 6.37 6.37 6.37 ...
##  $ AV45      : num  1.11 1.11 1.11 1.11 1.11 ...
##  $ HippoNV   : num  0.529 0.538 0.269 0.576 0.601 ...
##  $ e2_1      : int  1 0 0 0 1 0 0 0 0 1 ...
##  $ e4_1      : int  0 0 1 0 0 1 1 1 1 1 ...
##  $ rs3818361 : int  1 1 1 1 1 1 1 1 0 0 ...
##  $ rs744373  : int  1 0 1 1 1 0 1 1 0 1 ...
##  $ rs11136000: int  1 1 1 1 1 0 0 1 0 0 ...
##  $ rs610932  : int  1 1 0 1 0 1 1 1 0 1 ...
##  $ rs3851179 : int  1 0 1 0 0 1 0 0 1 0 ...
##  $ rs3764650 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ rs3865444 : int  0 1 1 0 0 0 1 1 1 0 ...
##  $ MMSCORE   : int  26 30 30 28 29 30 30 27 28 30 ...
##  $ TOTAL13   : num  8 1.67 12 3 10 3.67 4 11 3 9 ...
##  $ ID        : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
# try full-scale model - exclude MMSCORE as it is other output, trying to predict TOTAL13

data <- AD[,c(1:18)]
data <- subset(data, select = -c(MMSCORE) )
names(data)
```

```
##  [1] "AGE"        "PTGENDER"   "PTEDUCAT"   "FDG"        "AV45"
##  [6] "HippoNV"    "e2_1"       "e4_1"       "rs3818361"  "rs744373"
## [11] "rs11136000" "rs610932"   "rs3851179"  "rs3764650"  "rs3865444"
## [16] "TOTAL13"    "ID"
```

```
data$TOTAL13<- floor(data$TOTAL13)

#splitting the data by half
set.seed(1)
```

```r
sample_first_half <- sample(nrow(data),floor( nrow(data)/2) )
set.seed(1)
check<-data[sample_first_half,]
set.seed(1)
check_2<-data[-sample_first_half,]
#because the nrow(check) = 258 and nrow(check_2) = 259, take one row out from check_2 to make both data
check_2<-check_2[1:(nrow(check_2)-1),]


#tree model - no model selection
tree <- rpart( TOTAL13 ~ ., data = check)
prp(tree, nn.cex = 1)
```




```r
#regression model - no model selection
lm.AD <- lm(TOTAL13 ~ ., data = check)
summary(lm.AD)
```

```
##
## Call:
## lm(formula = TOTAL13 ~ ., data = check)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.9746  -3.6917  -0.3742   3.0324  24.5560
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.445427   8.603695   4.933 1.51e-06 ***
## AGE          -0.050038   0.061936  -0.808   0.4199
## PTGENDER      0.883919   0.807885   1.094   0.2750
## PTEDUCAT     -0.289126   0.144124  -2.006   0.0460 *
```

```
## FDG          -3.292122   0.733168  -4.490 1.10e-05 ***
## AV45          9.891883   2.322639   4.259 2.95e-05 ***
## HippoNV     -26.988253   6.063061  -4.451 1.31e-05 ***
## e2_1         -1.919871   1.468938  -1.307   0.1925
## e4_1         -1.536581   0.912739  -1.683   0.0936 .
## rs3818361    -0.690379   0.828054  -0.834   0.4053
## rs744373      1.099919   0.764651   1.438   0.1516
## rs11136000   -0.784515   0.813240  -0.965   0.3357
## rs610932      1.007857   0.808630   1.246   0.2138
## rs3851179    -1.006268   0.782695  -1.286   0.1998
## rs3764650    -0.310024   1.006457  -0.308   0.7583
## rs3865444     0.659019   0.774287   0.851   0.3955
## ID            0.002176   0.002714   0.802   0.4236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.032 on 241 degrees of freedom
## Multiple R-squared:  0.4002, Adjusted R-squared:  0.3604
## F-statistic: 10.05 on 16 and 241 DF,  p-value: < 2.2e-16
```

```
#prediction - tree
tree_pred_with_second_half<-floor(predict(tree, check_2))

current_error_train <- length(which(tree_pred_with_second_half != check$TOTAL13))/length(tree_pred_with

MSE_tree<-mean((check$TOTAL13-tree_pred_with_second_half)^2)
print(paste("MSE_tree is ",MSE_tree))
```

```
## [1] "MSE_tree is  88.5077519379845"
```

```
#prediction - regression
regression_pred_with_second_half<-floor(predict(lm.AD, check_2))
MSE_re<-mean((check$TOTAL13-regression_pred_with_second_half)^2)
print(paste("MSE_re is ",MSE_re))
```

```
## [1] "MSE_re is  81.2596899224806"
```

Mean square error for regression is smaller than tree model. Therefore, we chose regression model over the tree model. We tried to farther improve the model in the next following sections.

Improvement for regression model

```
# model selection
lm.AD.F <- step(lm.AD, direction="backward", test="F")
```

```
## Start:  AIC=943.73
## TOTAL13 ~ AGE + PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV +
##     e2_1 + e4_1 + rs3818361 + rs744373 + rs11136000 + rs610932 +
##     rs3851179 + rs3764650 + rs3865444 + ID
##
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - rs3764650   1      3.45 8773.0 941.83  0.0949   0.75832
## - ID          1     23.38 8792.9 942.42  0.6426   0.42356
## - AGE         1     23.75 8793.3 942.43  0.6527   0.41994
## - rs3818361   1     25.29 8794.8 942.47  0.6951   0.40526
## - rs3865444   1     26.36 8795.9 942.50  0.7244   0.39554
## - rs11136000  1     33.86 8803.4 942.72  0.9306   0.33567
```

```
## - PTGENDER    1     43.56 8813.1 943.01  1.1971   0.27500
## - rs610932    1     56.53 8826.1 943.39  1.5535   0.21384
## - rs3851179   1     60.15 8829.7 943.49  1.6529   0.19980
## - e2_1        1     62.16 8831.7 943.55  1.7082   0.19247
## <none>                    8769.5 943.73
## - rs744373    1     75.29 8844.8 943.93  2.0692   0.15160
## - e4_1        1    103.13 8872.7 944.74  2.8341   0.09358 .
## - PTEDUCAT    1    146.44 8916.0 946.00  4.0244   0.04596 *
## - AV45        1    660.02 9429.6 960.45 18.1382 2.946e-05 ***
## - HippoNV     1    720.98 9490.5 962.11 19.8137 1.306e-05 ***
## - FDG         1    733.68 9503.2 962.46 20.1625 1.103e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=941.83
## TOTAL13 ~ AGE + PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV +
##     e2_1 + e4_1 + rs3818361 + rs744373 + rs11136000 + rs610932 +
##     rs3851179 + rs3865444 + ID
##
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - ID          1     23.35 8796.3 940.52  0.6441   0.42302
## - AGE         1     24.35 8797.3 940.55  0.6717   0.41327
## - rs3818361   1     25.31 8798.3 940.57  0.6983   0.40419
## - rs3865444   1     25.70 8798.7 940.58  0.7090   0.40062
## - rs11136000  1     33.29 8806.3 940.81  0.9184   0.33886
## - PTGENDER    1     43.03 8816.0 941.09  1.1870   0.27702
## - rs610932    1     56.77 8829.8 941.49  1.5659   0.21201
## - rs3851179   1     58.79 8831.8 941.55  1.6218   0.20406
## - e2_1        1     63.27 8836.3 941.68  1.7452   0.18773
## <none>                    8773.0 941.83
## - rs744373    1     74.16 8847.2 942.00  2.0458   0.15392
## - e4_1        1    102.55 8875.5 942.83  2.8288   0.09388 .
## - PTEDUCAT    1    149.53 8922.5 944.19  4.1249   0.04335 *
## - AV45        1    677.89 9450.9 959.03 18.6994 2.238e-05 ***
## - FDG         1    730.37 9503.4 960.46 20.1470 1.110e-05 ***
## - HippoNV     1    731.93 9504.9 960.50 20.1901 1.087e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=940.52
## TOTAL13 ~ AGE + PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV +
##     e2_1 + e4_1 + rs3818361 + rs744373 + rs11136000 + rs610932 +
##     rs3851179 + rs3865444
##
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - rs3865444   1     22.75 8819.1 939.18  0.6285   0.42868
## - AGE         1     22.78 8819.1 939.18  0.6292   0.42842
## - rs3818361   1     23.05 8819.4 939.19  0.6368   0.42566
## - rs11136000  1     35.25 8831.6 939.55  0.9737   0.32474
## - PTGENDER    1     41.27 8837.6 939.72  1.1400   0.28672
## - rs3851179   1     52.94 8849.3 940.06  1.4625   0.22771
## - e2_1        1     55.51 8851.8 940.14  1.5334   0.21679
## - rs610932    1     56.23 8852.6 940.16  1.5533   0.21386
## <none>                    8796.3 940.52
```

```
## - rs744373    1     77.18 8873.5 940.77  2.1320   0.14554
## - e4_1        1    104.05 8900.4 941.55  2.8744   0.09128 .
## - PTEDUCAT    1    141.29 8937.6 942.63  3.9031   0.04933 *
## - AV45        1    706.32 9502.7 958.44 19.5123 1.506e-05 ***
## - HippoNV     1    718.35 9514.7 958.77 19.8446 1.282e-05 ***
## - FDG         1    744.80 9541.1 959.49 20.5752 9.013e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=939.18
## TOTAL13 ~ AGE + PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV +
##     e2_1 + e4_1 + rs3818361 + rs744373 + rs11136000 + rs610932 +
##     rs3851179
##
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - AGE         1     24.46 8843.5 937.90  0.6767   0.41153
## - rs3818361   1     28.70 8847.8 938.02  0.7942   0.37372
## - rs11136000  1     33.64 8852.7 938.16  0.9307   0.33565
## - PTGENDER    1     48.66 8867.7 938.60  1.3462   0.24707
## - e2_1        1     52.12 8871.2 938.70  1.4420   0.23097
## - rs3851179   1     52.33 8871.4 938.71  1.4478   0.23004
## - rs610932    1     54.95 8874.0 938.78  1.5202   0.21877
## <none>                    8819.1 939.18
## - rs744373    1     73.77 8892.9 939.33  2.0409   0.15440
## - e4_1        1     99.19 8918.3 940.07  2.7443   0.09888 .
## - PTEDUCAT    1    153.72 8972.8 941.64  4.2530   0.04024 *
## - AV45        1    685.96 9505.1 956.51 18.9788 1.948e-05 ***
## - HippoNV     1    720.89 9540.0 957.45 19.9451 1.219e-05 ***
## - FDG         1    755.08 9574.2 958.38 20.8909 7.729e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=937.9
## TOTAL13 ~ PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV + e2_1 +
##     e4_1 + rs3818361 + rs744373 + rs11136000 + rs610932 + rs3851179
##
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - rs3818361   1     24.90 8868.5 936.62  0.6899    0.4070
## - rs11136000  1     39.93 8883.5 937.06  1.1061    0.2940
## - PTGENDER    1     42.49 8886.0 937.13  1.1772    0.2790
## - rs3851179   1     45.56 8889.1 937.22  1.2622    0.2623
## - e2_1        1     50.03 8893.6 937.35  1.3859    0.2402
## - rs610932    1     51.49 8895.0 937.39  1.4264    0.2335
## <none>                    8843.5 937.90
## - rs744373    1     74.91 8918.5 938.07  2.0754    0.1510
## - e4_1        1     80.77 8924.3 938.24  2.2376    0.1360
## - PTEDUCAT    1    142.30 8985.9 940.02  3.9423    0.0482 *
## - AV45        1    661.63 9505.2 954.51 18.3297 2.668e-05 ***
## - HippoNV     1    720.62 9564.2 956.11 19.9640 1.206e-05 ***
## - FDG         1    782.17 9625.7 957.76 21.6692 5.311e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=936.62
```

```
## TOTAL13 ~ PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV + e2_1 +
##     e4_1 + rs744373 + rs11136000 + rs610932 + rs3851179
##
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - rs11136000  1     34.70 8903.2 935.63  0.9626   0.32750
## - PTGENDER    1     40.14 8908.6 935.79  1.1134   0.29238
## - rs3851179   1     43.95 8912.4 935.90  1.2190   0.27063
## - e2_1        1     44.14 8912.6 935.90  1.2243   0.26960
## - rs610932    1     44.62 8913.1 935.92  1.2376   0.26702
## <none>                    8868.5 936.62
## - rs744373    1     77.28 8945.7 936.86  2.1437   0.14444
## - e4_1        1     87.95 8956.4 937.17  2.4395   0.11960
## - PTEDUCAT    1    134.30 9002.8 938.50  3.7253   0.05474 .
## - AV45        1    668.28 9536.7 953.37 18.5372 2.408e-05 ***
## - HippoNV     1    703.74 9572.2 954.32 19.5209 1.492e-05 ***
## - FDG         1    797.43 9665.9 956.84 22.1197 4.273e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=935.63
## TOTAL13 ~ PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV + e2_1 +
##     e4_1 + rs744373 + rs610932 + rs3851179
##
##             Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - PTGENDER   1     40.86 8944.0 934.81  1.1336   0.28805
## - e2_1       1     44.46 8947.6 934.92  1.2334   0.26782
## - rs3851179  1     47.56 8950.7 935.00  1.3194   0.25181
## - rs610932   1     50.39 8953.5 935.09  1.3979   0.23821
## <none>                   8903.2 935.63
## - rs744373   1     79.00 8982.2 935.91  2.1916   0.14004
## - e4_1       1     86.45 8989.6 936.12  2.3983   0.12275
## - PTEDUCAT   1    129.93 9033.1 937.37  3.6046   0.05878 .
## - HippoNV    1    693.41 9596.6 952.98 19.2372 1.710e-05 ***
## - AV45       1    720.18 9623.3 953.70 19.9799 1.193e-05 ***
## - FDG        1    805.09 9708.2 955.96 22.3356 3.847e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=934.81
## TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + e2_1 + e4_1 + rs744373 +
##     rs610932 + rs3851179
##
##             Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - rs610932   1     44.59 8988.6 934.09  1.2364   0.26725
## - rs3851179  1     49.42 8993.4 934.23  1.3704   0.24287
## - e2_1       1     50.56 8994.6 934.27  1.4019   0.23755
## <none>                   8944.0 934.81
## - e4_1       1     80.70 9024.7 935.13  2.2377   0.13595
## - rs744373   1     80.81 9024.8 935.13  2.2406   0.13570
## - PTEDUCAT   1    105.52 9049.5 935.84  2.9259   0.08842 .
## - AV45       1    687.56 9631.6 951.92 19.0647 1.857e-05 ***
## - HippoNV    1    777.67 9721.7 954.32 21.5633 5.556e-06 ***
## - FDG        1    847.69 9791.7 956.17 23.5048 2.200e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=934.09
## TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + e2_1 + e4_1 + rs744373 +
##     rs3851179
##
##             Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - e2_1       1     51.81 9040.4 933.58  1.4353   0.23204
## - rs3851179  1     52.36 9041.0 933.59  1.4506   0.22958
## <none>                   8988.6 934.09
## - rs744373   1     79.27 9067.9 934.36  2.1960   0.13963
## - e4_1       1     79.49 9068.1 934.37  2.2019   0.13911
## - PTEDUCAT   1    103.75 9092.4 935.06  2.8740   0.09127 .
## - AV45       1    669.03 9657.6 950.62 18.5332 2.403e-05 ***
## - HippoNV    1    778.94 9767.5 953.54 21.5781 5.506e-06 ***
## - FDG        1    870.14 9858.7 955.93 24.1043 1.652e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=933.58
## TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + e4_1 + rs744373 +
##     rs3851179
##
##             Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - rs3851179  1     43.24 9083.7 932.81  1.1957   0.27524
## - e4_1       1     61.63 9102.0 933.33  1.7042   0.19294
## <none>                   9040.4 933.58
## - rs744373   1     79.32 9119.7 933.83  2.1934   0.13986
## - PTEDUCAT   1    105.43 9145.9 934.57  2.9156   0.08897 .
## - AV45       1    688.69 9729.1 950.52 19.0447 1.870e-05 ***
## - HippoNV    1    803.74 9844.2 953.55 22.2263 4.029e-06 ***
## - FDG        1    873.44 9913.9 955.37 24.1538 1.610e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=932.81
## TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + e4_1 + rs744373
##
##            Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - e4_1      1     54.87 9138.5 932.36  1.5161   0.21937
## <none>                  9083.7 932.81
## - rs744373  1     78.74 9162.4 933.03  2.1757   0.14146
## - PTEDUCAT  1    104.78 9188.4 933.77  2.8954   0.09007 .
## - AV45      1    719.63 9803.3 950.48 19.8849 1.241e-05 ***
## - HippoNV   1    829.47 9913.1 953.35 22.9199 2.888e-06 ***
## - FDG       1    839.81 9923.5 953.62 23.2057 2.520e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=932.36
## TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + rs744373
##
##            Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>                  9138.5 932.36
```

```
## - rs744373  1     73.06  9211.6 932.42  2.0146   0.15703
## - PTEDUCAT  1    117.09  9255.6 933.65  3.2287   0.07356 .
## - AV45      1    667.81  9806.3 948.56 18.4152 2.535e-05 ***
## - FDG       1    790.91  9929.4 951.78 21.8099 4.900e-06 ***
## - HippoNV   1    893.94 10032.5 954.44 24.6510 1.267e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.AD.F)
```

```
##
## Call:
## lm(formula = TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + rs744373,
##     data = check)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3744  -4.0272  -0.3828   3.2256  27.4873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.2172     6.4348   6.250 1.74e-09 ***
## PTEDUCAT     -0.2472     0.1376  -1.797   0.0736 .
## FDG          -3.2908     0.7047  -4.670 4.90e-06 ***
## AV45          8.8541     2.0633   4.291 2.53e-05 ***
## HippoNV     -27.1304     5.4644  -4.965 1.27e-06 ***
## rs744373      1.0791     0.7602   1.419   0.1570
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.022 on 252 degrees of freedom
## Multiple R-squared:  0.375,  Adjusted R-squared:  0.3626
## F-statistic: 30.24 on 5 and 252 DF,  p-value: < 2.2e-16
```

```
anova(lm.AD.F ,lm.AD)
```

```
## Analysis of Variance Table
##
## Model 1: TOTAL13 ~ PTEDUCAT + FDG + AV45 + HippoNV + rs744373
## Model 2: TOTAL13 ~ AGE + PTGENDER + PTEDUCAT + FDG + AV45 + HippoNV +
##     e2_1 + e4_1 + rs3818361 + rs744373 + rs11136000 + rs610932 +
##     rs3851179 + rs3764650 + rs3865444 + ID
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    252 9138.5
## 2    241 8769.5 11    368.98 0.9218 0.5199
```

Improvement for Tree

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:RCurl':
##
##     complete
```

```
library(ggplot2)


set.seed(1)
errintrain <- NULL
errintest <- NULL
leaf.v <- NULL
cp<- NULL

for(i in seq(0.2,0,by=-0.005) ){
  tree <- rpart( TOTAL13 ~ ., data = check, cp= i  )
  pred.train <- floor(predict(tree, check))
  pred.test <- floor(predict(tree, check_2))
  current_error_train <- length(which(pred.train != check$TOTAL13))/length(pred.train)
  current_error_test <- length(which(pred.test != check_2$TOTAL13))/length(pred.test)
  errintrain <- c(errintrain, current_error_train)
  errintest <- c(errintest, current_error_test)
  leaf.v <- c(leaf.v, length(which(tree$frame$var == "<leaf>")))
  cp <- c(cp,i)
}
err.mat <- as.data.frame( cbind( train_err = errintrain, test_err = errintest , leaf_num = leaf.v ,cp_ta
err.mat$leaf_num <- as.factor( err.mat$leaf_num  )
err.mat <- unique(err.mat)
err.mat <- err.mat %>% gather(type, error, train_err,test_err)
print(err.mat)
```

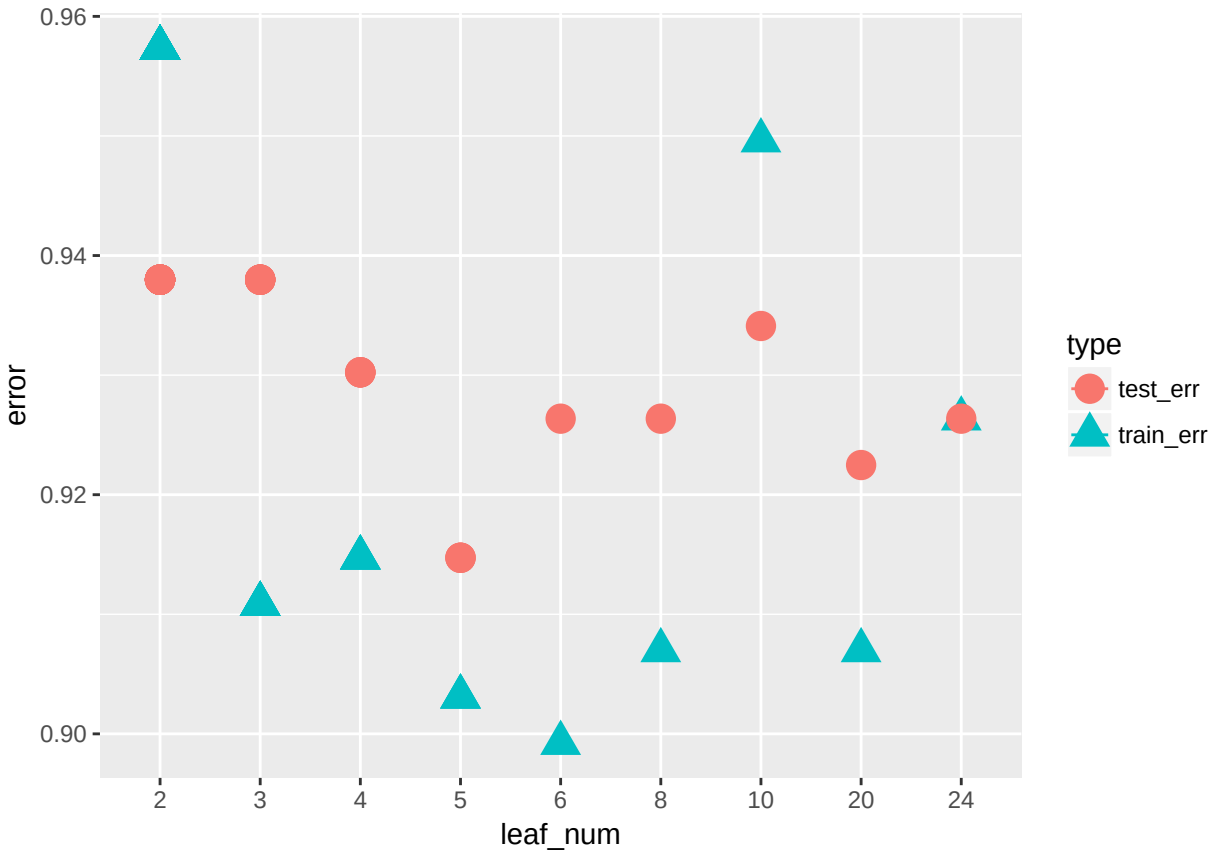
```
##    leaf_num cp_table      type     error
## 1         2    0.200 train_err 0.9573643
## 2         2    0.195 train_err 0.9573643
## 3         2    0.190 train_err 0.9573643
## 4         2    0.185 train_err 0.9573643
## 5         2    0.180 train_err 0.9573643
## 6         2    0.175 train_err 0.9573643
## 7         2    0.170 train_err 0.9573643
## 8         2    0.165 train_err 0.9573643
## 9         2    0.160 train_err 0.9573643
## 10        2    0.155 train_err 0.9573643
## 11        2    0.150 train_err 0.9573643
## 12        2    0.145 train_err 0.9573643
## 13        2    0.140 train_err 0.9573643
## 14        2    0.135 train_err 0.9573643
## 15        2    0.130 train_err 0.9573643
## 16        2    0.125 train_err 0.9573643
## 17        2    0.120 train_err 0.9573643
```

```
## 18        2    0.115 train_err 0.9573643
## 19        2    0.110 train_err 0.9573643
## 20        2    0.105 train_err 0.9573643
## 21        2    0.100 train_err 0.9573643
## 22        2    0.095 train_err 0.9573643
## 23        3    0.090 train_err 0.9108527
## 24        3    0.085 train_err 0.9108527
## 25        3    0.080 train_err 0.9108527
## 26        3    0.075 train_err 0.9108527
## 27        3    0.070 train_err 0.9108527
## 28        3    0.065 train_err 0.9108527
## 29        3    0.060 train_err 0.9108527
## 30        4    0.055 train_err 0.9147287
## 31        4    0.050 train_err 0.9147287
## 32        4    0.045 train_err 0.9147287
## 33        4    0.040 train_err 0.9147287
## 34        5    0.035 train_err 0.9031008
## 35        5    0.030 train_err 0.9031008
## 36        5    0.025 train_err 0.9031008
## 37        6    0.020 train_err 0.8992248
## 38        8    0.015 train_err 0.9069767
## 39       10    0.010 train_err 0.9496124
## 40       20    0.005 train_err 0.9069767
## 41       24    0.000 train_err 0.9263566
## 42        2    0.200  test_err 0.9379845
## 43        2    0.195  test_err 0.9379845
## 44        2    0.190  test_err 0.9379845
## 45        2    0.185  test_err 0.9379845
## 46        2    0.180  test_err 0.9379845
## 47        2    0.175  test_err 0.9379845
## 48        2    0.170  test_err 0.9379845
## 49        2    0.165  test_err 0.9379845
## 50        2    0.160  test_err 0.9379845
## 51        2    0.155  test_err 0.9379845
## 52        2    0.150  test_err 0.9379845
## 53        2    0.145  test_err 0.9379845
## 54        2    0.140  test_err 0.9379845
## 55        2    0.135  test_err 0.9379845
## 56        2    0.130  test_err 0.9379845
## 57        2    0.125  test_err 0.9379845
## 58        2    0.120  test_err 0.9379845
## 59        2    0.115  test_err 0.9379845
## 60        2    0.110  test_err 0.9379845
## 61        2    0.105  test_err 0.9379845
## 62        2    0.100  test_err 0.9379845
## 63        2    0.095  test_err 0.9379845
## 64        3    0.090  test_err 0.9379845
## 65        3    0.085  test_err 0.9379845
## 66        3    0.080  test_err 0.9379845
## 67        3    0.075  test_err 0.9379845
## 68        3    0.070  test_err 0.9379845
## 69        3    0.065  test_err 0.9379845
## 70        3    0.060  test_err 0.9379845
## 71        4    0.055  test_err 0.9302326
```

```
## 72        4    0.050  test_err 0.9302326
## 73        4    0.045  test_err 0.9302326
## 74        4    0.040  test_err 0.9302326
## 75        5    0.035  test_err 0.9147287
## 76        5    0.030  test_err 0.9147287
## 77        5    0.025  test_err 0.9147287
## 78        6    0.020  test_err 0.9263566
## 79        8    0.015  test_err 0.9263566
## 80       10    0.010  test_err 0.9341085
## 81       20    0.005  test_err 0.9224806
## 82       24    0.000  test_err 0.9263566
```

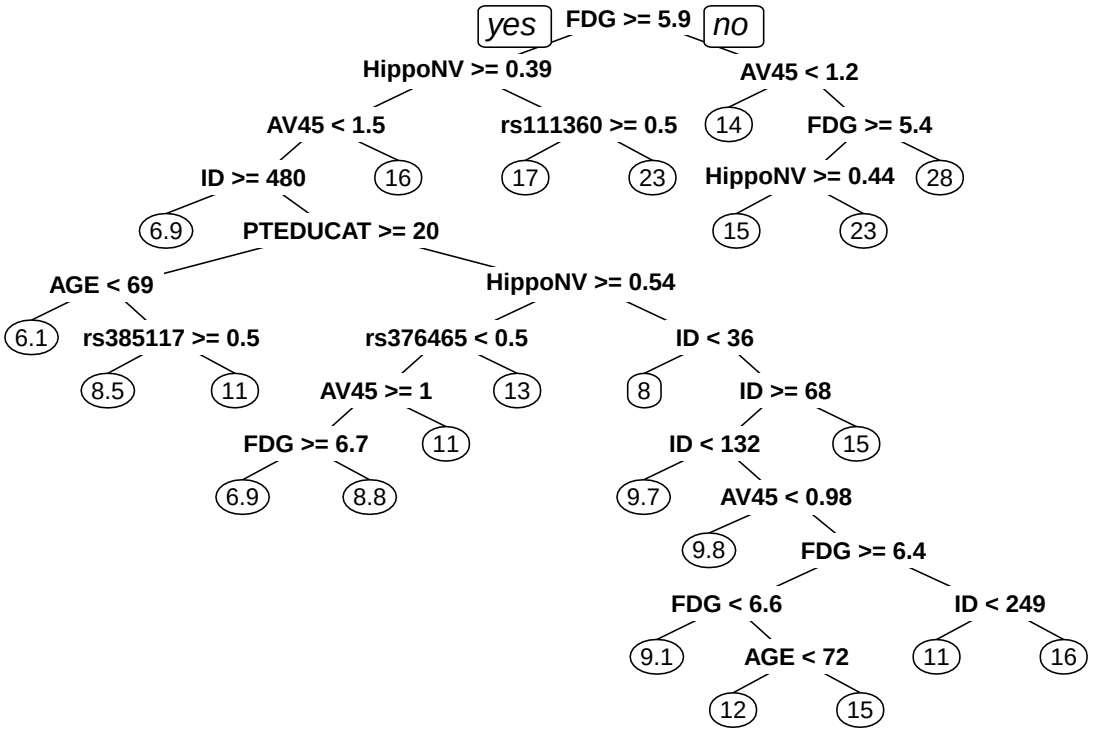
As the gap between test and train error data is the smallest at leaf number equal to 24, the adequate number of leaf node would be 24. Other leaf numbers may result overfitting or underfitting of predicted data more.

```
data.plot <- err.mat %>% mutate(type = type)
ggplot(data.plot, aes(x=leaf_num, y=error, shape = type, color=type)) + geom_line() +
  geom_point(size=5)
```



Final decision tree model can be selected with the 24 decision (leaf) nodes WITH cp = 0

```
tree_0.05 <- prune(tree, cp = 0)
prp(tree_0.05, nn.cex = 1)
```




```
#MSE for improved tree
tree_pred_with_second_half_improved<-floor(predict(tree_0.05, check_2))

current_error_train <- length(which(tree_pred_with_second_half_improved != check$TOTAL13))/length(tree_

MSE_tree_improved<-mean((check$TOTAL13-tree_pred_with_second_half_improved)^2)
print(paste("MSE_tree is ",MSE_tree))
```

```
## [1] "MSE_tree is  88.5077519379845"
```

```
print(paste("MSE_tree_improved is ",MSE_tree_improved))
```

```
## [1] "MSE_tree_improved is  91.8682170542636"
```

```
#MSE for improved regression model
regression_pred_with_second_half_improved<-floor(predict(lm.AD.F, check_2))
MSE_re_improved<-mean((check$TOTAL13-regression_pred_with_second_half_improved)^2)
print(paste("MSE_re is ",MSE_re))
```

```
## [1] "MSE_re is  81.2596899224806"
```

```
print(paste("MSE_re_improved is ",MSE_re_improved))
```

```
## [1] "MSE_re_improved is  80.4961240310078"
```

After improving the both regression and tree models, mean square error for regression model decreased indicating the improvement of the model predictions. However, for tree model MSE increased. we chose regression model over tree model because it had lower MSE. The MSE differences between original model and improved model in tree model was bigger than those in regression model.

Chapter 2, Exercise 4

Consider the case that, in building linear regression models, there is a concern that some data points may be more important (or more trustable). Thus, it is not uncommon to assign a weight to each data point. Denote the weight for the ith data point as w_i. We still want to estimate the regression parameters in the least squares framework. Follow the process of the derivation of the least squares estimator and propose your new estimator of the regression parameters.

The weighted mean square error in matrix is:

$$min(WMSE(b)) = \frac{1}{n}\sum_{i=1}^{n} w_i(y_i - x_i b)^2$$

We can rewrite w_i in matrix form as W, which is a diaganol matrix where the i_{th} diaganol element is the weight for the x_i observation. In matrix form this is:

$$min(WMSE(b)) = \frac{1}{n}(Y - X\beta)^T W(Y - X\beta)$$

Expanding the terms:

$$min(WMSE(b)) = \frac{1}{n}(Y^T WY - Y^T WX\beta - \beta^T X^T WY + \beta^T X^T WX\beta)$$

Differentiating with respect to β and setting equal to zero:

$$min(WMSE(b)) = \frac{2}{n}(-X^T WY + X^T WX\beta)$$

Setting this equal to 0, we get:

$$\hat{\beta} = (X^T WX)^{-1}(X^T WY)$$

Chapter 3, Exercise 1

Create a new binary variable based on AGE, by labeling the subjects whose age is above the mean of AGE to be class "1" and labeling the subjects whose age is below the mean of AGE to be class "0". Then, repeat the analysis shown in the R lab of this chapter for the logistic regression model and the analysis shown in the R lab of Chapter 2 for decision tree model. Identify the final models you would select, evaluate the models, and compare the regression model with the tree model.

We will use all of the predictors (except for AGE, MMSCORE, TOTAL13, and DX_bl) to predict where a person's age is above or below the mean age.

Logistic Regression Model

We begin by loading the data and creating new column named AGE_bin. AGE_bin: Contains "1" if the subject's age is >= mean(AGE); Contains "0" if the subject's age is < mean(AGE).

```
library(RCurl)
AD <- read.csv(text=getURL("https://raw.githubusercontent.com/shuailab/ind_498/master/resource/data/AD.
AD$ID = c(1:dim(AD)[1])
AD$AGE_bin = ifelse(AD$AGE >= mean(AD$AGE), 1, 0)
AD = AD[ , !(names(AD) %in% c("AGE", "MMSCORE", "TOTAL13", "DX_bl"))]
```

Fitting our model using all of the predictors yields the significant predicators as HippoNV, e4_1, and PTEDUCAT.

```r
logit.AD.1 <- glm(AGE_bin ~ ., data = AD[, -c(15)], family = "binomial")
summary(logit.AD.1)
```

```
##
## Call:
## glm(formula = AGE_bin ~ ., family = "binomial", data = AD[, -c(15)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0397  -1.0028   0.4678   1.0370   2.0055
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.204405   1.649330   3.155  0.00160 **
## PTGENDER     0.375794   0.202412   1.857  0.06337 .
## PTEDUCAT    -0.075191   0.036717  -2.048  0.04057 *
## FDG         -0.064370   0.164254  -0.392  0.69514
## AV45         0.647951   0.536258   1.208  0.22694
## HippoNV     -9.621050   1.524091  -6.313 2.74e-10 ***
## e2_1        -0.397114   0.340758  -1.165  0.24386
## e4_1        -0.575043   0.218929  -2.627  0.00862 **
## rs3818361   -0.261183   0.203489  -1.284  0.19931
## rs744373    -0.037297   0.194075  -0.192  0.84760
## rs11136000   0.189847   0.204704   0.927  0.35371
## rs610932     0.002931   0.201152   0.015  0.98837
## rs3851179    0.025735   0.195006   0.132  0.89501
## rs3764650   -0.301348   0.243323  -1.238  0.21554
## rs3865444   -0.123676   0.192755  -0.642  0.52112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 716.67  on 516  degrees of freedom
## Residual deviance: 633.62  on 502  degrees of freedom
## AIC: 663.62
##
## Number of Fisher Scoring iterations: 4
```

Fitting the model based on the significant predicators of the last model gives us that only two out of the three predictors (HippoNV and e4_1) are actually significant.

```r
logit.AD.2 <- glm(AGE_bin ~ HippoNV + e4_1 + PTEDUCAT , data = AD[, -c(15)], family = "binomial")
summary(logit.AD.2)
```

```
##
## Call:
## glm(formula = AGE_bin ~ HippoNV + e4_1 + PTEDUCAT, family = "binomial",
##     data = AD[, -c(15)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1043  -1.0323   0.5049   1.0557   1.9744
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   6.2949     0.9368   6.720 1.82e-11 ***
## HippoNV     -10.6819     1.4060  -7.597 3.03e-14 ***
## e4_1         -0.4556     0.1960  -2.325   0.0201 *
## PTEDUCAT     -0.0646     0.0350  -1.846   0.0649 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 716.67  on 516  degrees of freedom
## Residual deviance: 644.38  on 513  degrees of freedom
## AIC: 652.38
##
## Number of Fisher Scoring iterations: 4
```

We will use the following visualization of the relationships between some of the predicators and the outcome in order to make an educated guess on which other predictors should be considered.

None of the ploted variables seem to be able to properly classify the data (since all of the pairs of box overlap significantly).

```
require(reshape2)
```

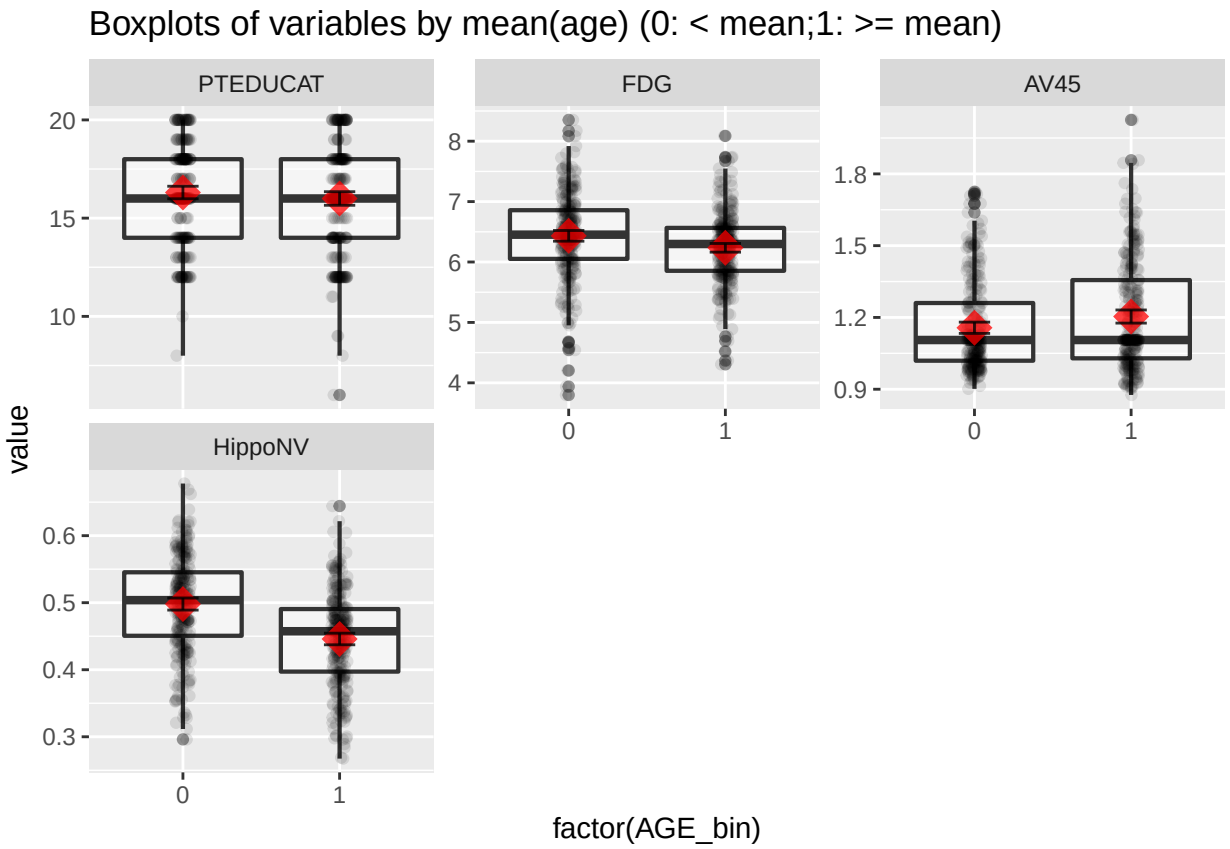
```
## Loading required package: reshape2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
AD.long <- melt(AD[,c(2:5, 15, 16)], id.vars = c("ID", "AGE_bin"))

# Plot the data using ggplot
require(ggplot2)
p <- ggplot(AD.long, aes(x = factor(AGE_bin), y = value))

# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)

# points for observed data
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0), alpha = 0.1)

# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
alpha = 0.75, colour = "red")

# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
width = .2, alpha = 0.8)

p <- p + facet_wrap( ~ variable, scales = "free_y", ncol = 3)


p <- p + labs(title = "Boxplots of variables by mean(age) (0: < mean;1: >= mean)")
```

```
print(p)
```



We will use the step() function to automatically choose the best model. The significant variables are HippoNV, e4_1, and PTEDUCAT. This model explains all but 75.13 of the total deviance with 4 less degrees of freedom.

```
logit.AD.full <- glm(AGE_bin ~ ., data = AD[!(names(AD) %in% c("ID"))], family = "binomial")
logit.AD.final <- step(logit.AD.full, direction="both", trace = 0)
summary(logit.AD.final)
```

```
##
## Call:
## glm(formula = AGE_bin ~ PTGENDER + PTEDUCAT + HippoNV + e4_1,
##     family = "binomial", data = AD[!(names(AD) %in% c("ID"))])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9929  -1.0272   0.4861   1.0470   2.0315
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.76903    0.98286   5.870 4.37e-09 ***
## PTGENDER      0.33253    0.19724   1.686   0.0918 .
## PTEDUCAT     -0.07674    0.03594  -2.135   0.0327 *
## HippoNV     -10.25165    1.42535  -7.192 6.37e-13 ***
## e4_1         -0.43670    0.19670  -2.220   0.0264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 716.67  on 516  degrees of freedom
## Residual deviance: 641.54  on 512  degrees of freedom
## AIC: 651.54
##
## Number of Fisher Scoring iterations: 4
```

We can find the 95% confidence intervals of the regression parameters. We notice that the largest 95% confidence interval is for HippoNV and the smallest 95% confidence interval is for PTGENDER. This tells us that the estimated coefficient of PTGENDER is more accurate than that of HippoNV.

```
## CISs of the regression parameters using profiled log-likelihood
confint(logit.AD.final)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %       97.5 %
## (Intercept)   3.88264812  7.741525498
## PTGENDER     -0.05416772  0.719837403
## PTEDUCAT     -0.14796972 -0.006846201
## HippoNV     -13.12344175 -7.527279636
## e4_1         -0.82526264 -0.053353296
```

We can also use the Wald Test to test the significance of the regression parameters.

```
library(aod)

wald.test(b = coef(logit.AD.final), Sigma = vcov(logit.AD.final), Terms = 2)
```

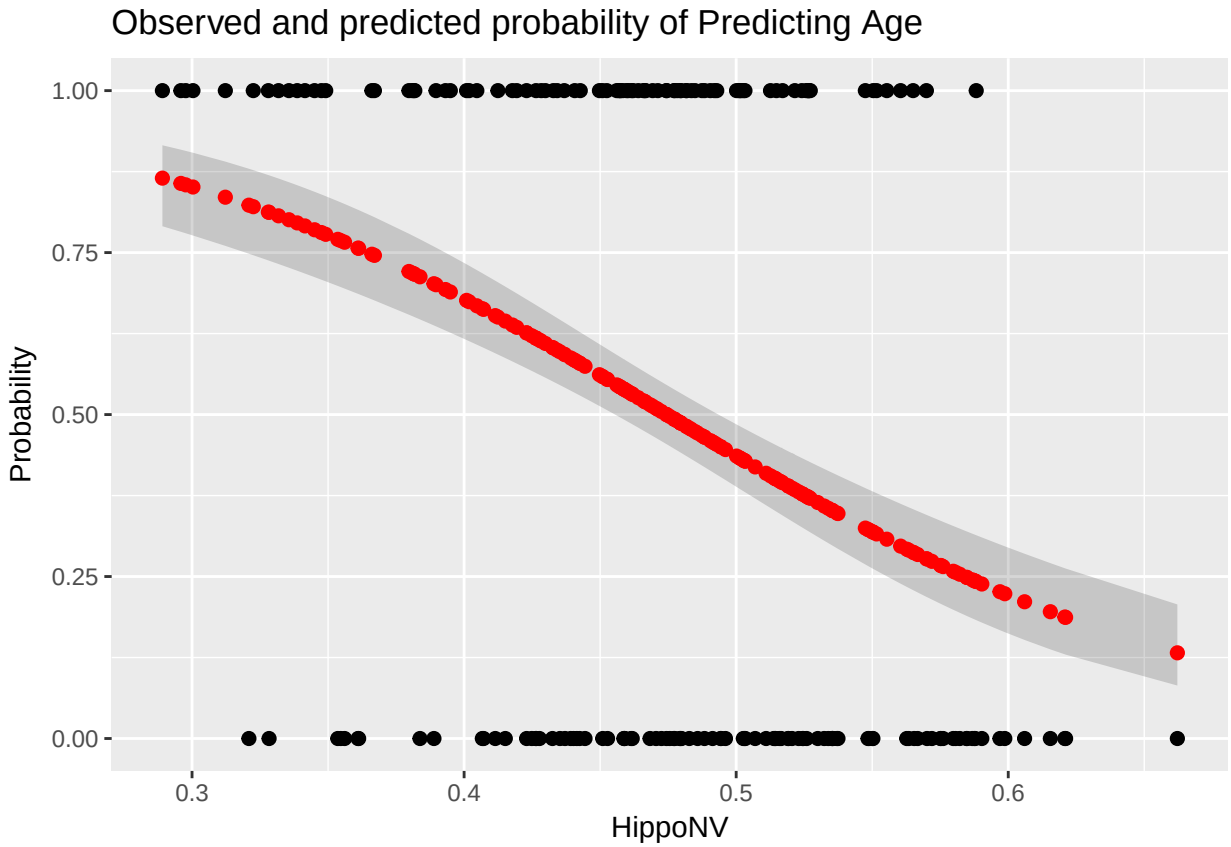
```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 2.8, df = 1, P(> X2) = 0.092
```

If our model simply depended on one predicator, say HippoNV, then we would be able to test how our model works on the data. We would do this by randomly choosing 200 samples from the AD dataset to make AD.pred. Then we would visualize these predictions and their 95% CIs.

We see that the 'tails' of the curve made by the red points do not match up with points at the end of the black lines. This means that HippoNV isn't a good predictor of AGE_bin even at the most extreme cases.

```
# Dataset that we will test a model with one predictor: HippoNV
set.seed(1)
AD.pred <- AD[sample(1:dim(AD)[1], 200),]

# pred will have our predictions
logit.HippoNV <- glm(AGE_bin ~ HippoNV, data = AD[!(names(AD) %in% c("ID"))], family = "binomial")
pred <- predict(logit.HippoNV, AD.pred, type = "link", se.fit = TRUE)
AD.pred$fit <- pred$fit
AD.pred$se.fit <- pred$se.fit

# CI for fitted values
AD.pred <- within(AD.pred, {
  # add "fitted" to make predictions at appended temp values
  fitted = exp(fit)/(1 + exp(fit))
```

```
  fit.lower = exp(fit - 1.96 * se.fit) / (1 + exp(fit - 1.96 * se.fit))
  fit.upper = exp(fit + 1.96 *se.fit) / (1 + exp(fit + 1.96 * se.fit))
})

# Visualizing the predication
library(ggplot2)
newData <- AD.pred[order(AD.pred$HippoNV),]
p <- ggplot(newData, aes(x = HippoNV, y = AGE_bin))
# predicted curve and point-wise 95% CI
p <- p + geom_ribbon(aes(x = HippoNV, ymin = fit.lower, ymax = fit.upper), alpha = 0.2)
# p <- p + geom_line(aes(x = HippoNV, y = fitted), colour="red") # take the lines off
# fitted values
p <- p + geom_point(aes(y = fitted), size=2, colour="red")
# observed values
p <- p + geom_point(size = 2)
p <- p + ylab("Probability")
p <- p + labs(title = "Observed and predicted probability of Predicting Age")
print(p)
```



Since our optimal model depends on several predictors, we can't use the above method of visualization.

We will use the following confusion matrix to see how well we can predict the output based on our optimal model. We see that our model gave 127 correct predictions and 73 incorrect predications. Our model has an accurary rate 63.5% on this randomly chosen subset of data.

```
# Dataset that we will test a model with one predictor: HippoNV
set.seed(1)
AD.pred <- AD[sample(1:dim(AD)[1], 200),]
```

```
# pred will have our predictions
pred <- predict(logit.AD.final, AD.pred, type = "link", se.fit = TRUE)
AD.pred$fit <- pred$fit
AD.pred$se.fit <- pred$se.fit

# CI for fitted values
AD.pred <- within(AD.pred, {
# add "fitted" to make predictions at appended temp values
fitted = exp(fit)/(1 + exp(fit))
fit.lower = exp(fit - 1.96 * se.fit) / (1 + exp(fit - 1.96 * se.fit))
fit.upper = exp(fit + 1.96 *se.fit) / (1 + exp(fit + 1.96 * se.fit))
})

# creating the confusion table
AD.pred$AGE_bin_predict <-  ifelse(AD.pred$fitted >= 0.5, 1, 0)
table(AD.pred$AGE_bin_predict, AD.pred$AGE_bin)
```

```
##
##      0  1
##   0 66 35
##   1 38 61
```

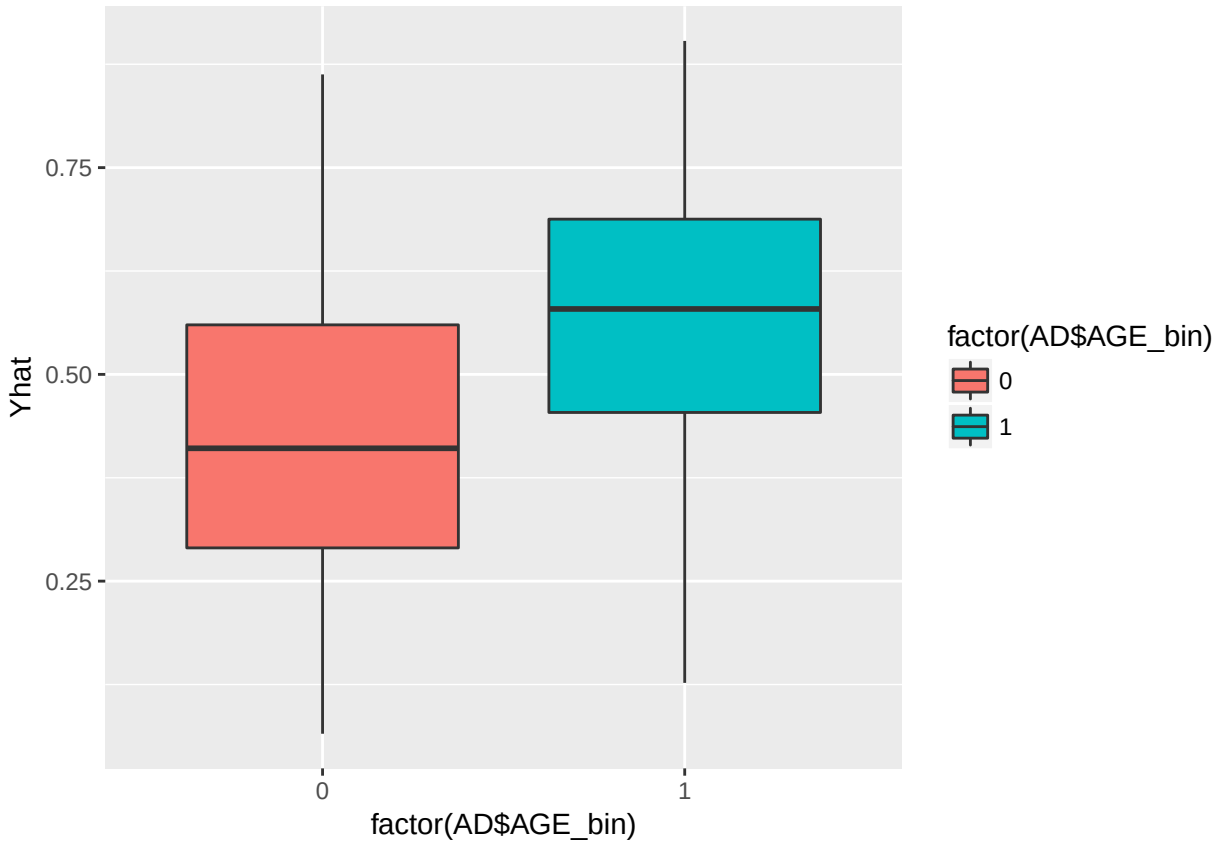
Another way to visualize our predictions would be to use the following boxplots. We see that our predictions are not super accurate. We see this since the boxes are overlapping and are not very thin.

```
# evaluate how well the model fits the data
# predicted probabilities
Yhat <- fitted(logit.AD.final)
# the observed events
YObs <- AD$AGE_bin
# calculate the correlation between the predicted and observed
cor(Yhat,AD$AGE_bin)
```

```
## [1] 0.3694917
```

```
# visualize the correlation
tempData = cbind(Yhat,AD$AGE_bin)
require(ggplot2)
qplot(factor(AD$AGE_bin), Yhat, data = AD,
geom=c("boxplot"), fill = factor(AD$AGE_bin),title="Prediction versus Observed")
```

```
## Warning: Ignoring unknown parameters: title
```



We will test whether or not there is a lack-of-fit. Since dev.p.val is 8.162904e-05, which is not greater than 0.10, there is a large lack of model fit. We conclude that the error in our predictions are coming from a lack of fit from the model.

```
# Test residual deviance for lack-of-fit (if > 0.10, little-to-no lack-of-fit)
dev.p.val <- 1 - pchisq(logit.AD.final$deviance, logit.AD.final$df.residual)
dev.p.val
```

```
## [1] 8.162904e-05
```

We will conclude by computing the odds ratios for our predicators and their corresponding 95% confidence intervals to determine the influence of the predictors. Again, the most narrow confidence interval belongs to PTEDUCAT.

```
## odds ratios and 95% CI
exp(cbind(OR = coef(logit.AD.final), confint(logit.AD.final)))
```

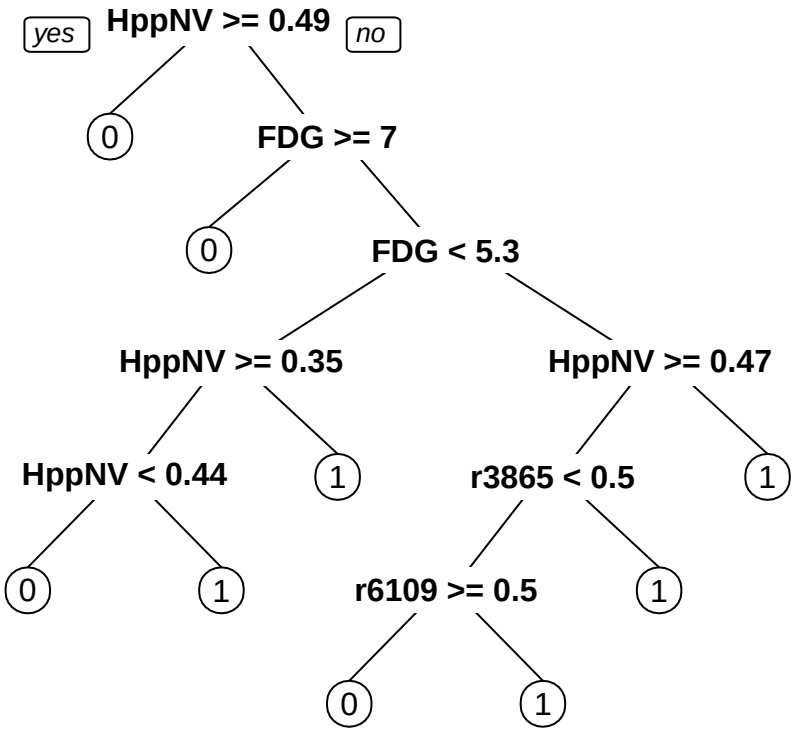
```
## Waiting for profiling to be done...
```

```
##                         OR        2.5 %       97.5 %
## (Intercept) 3.202278e+02 4.855262e+01 2.301981e+03
## PTGENDER    1.394497e+00 9.472732e-01 2.054099e+00
## PTEDUCAT    9.261333e-01 8.624572e-01 9.931772e-01
## HippoNV     3.529925e-05 1.997844e-06 5.382004e-04
## e4_1        6.461631e-01 4.381199e-01 9.480450e-01
```

Decision Tree

We now create a decision tree based on the dataset. We see that the splitting happens with regards to the predictors HippoNV, FDG, rs3865444, and rs610932.

```
AD$AGE_bin <- as.factor(AD$AGE_bin)
AD.tree <- rpart(AGE_bin ~., data = AD[!(names(AD) %in% c("ID"))])
prp(AD.tree, varlen=5)
```



When we look at the variable importance of each predictor, we see that the most important variables are HippoNV and FDG.

```
print(AD.tree$variable.importance)
```

```
##    HippoNV        FDG       AV45   rs610932  rs3865444       e4_1
## 39.2980594 16.3736965  5.7736919  3.4405160  3.1850014  1.0666667
##   PTEDUCAT       e2_1 rs11136000
##  0.4498678  0.3604479  0.3555556
```

Our objective is now to prune the tree. Testing different different values for cp, we see that our decision tree is most accurate when our tree has about 3 to 4 leaves.

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

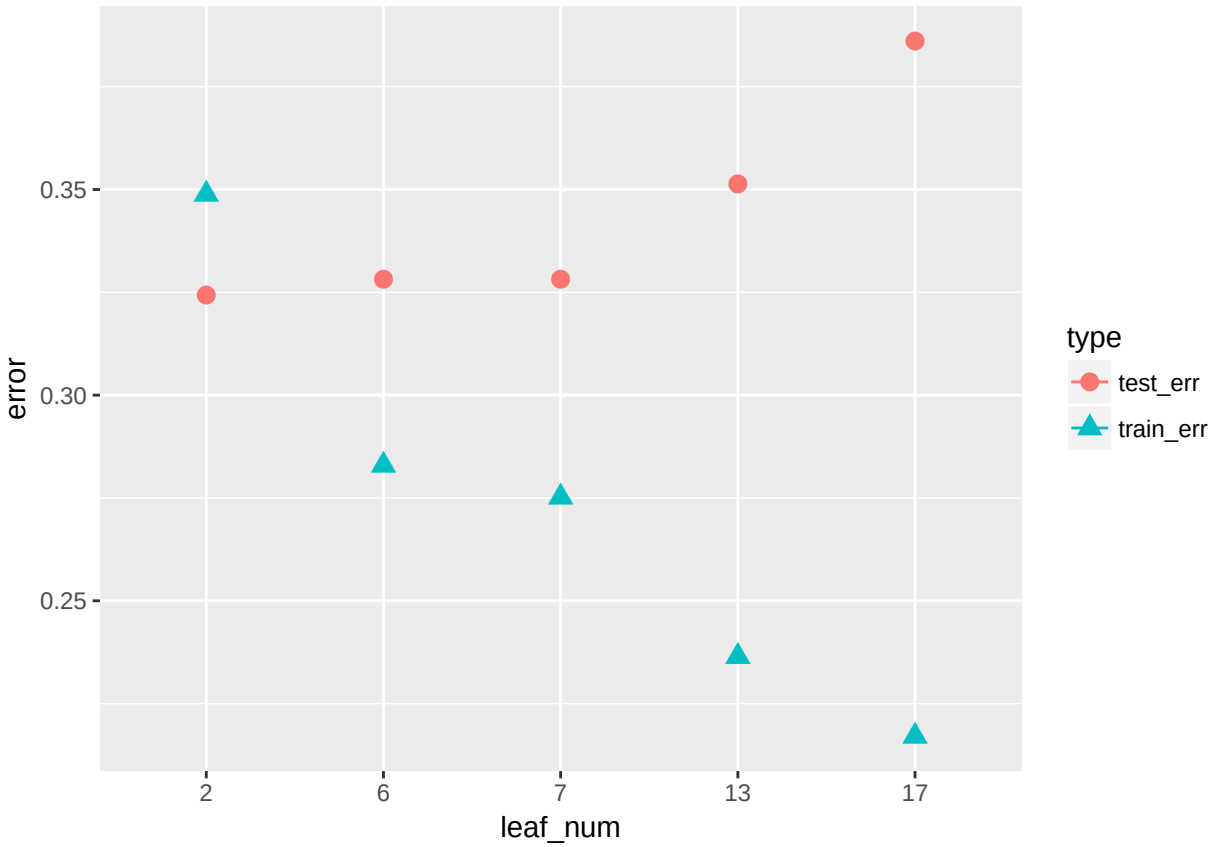
```
library(tidyr)
library(ggplot2)
library(rpart)
```

```r
library(rpart.plot)
library(dplyr)
library(partykit)
```

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

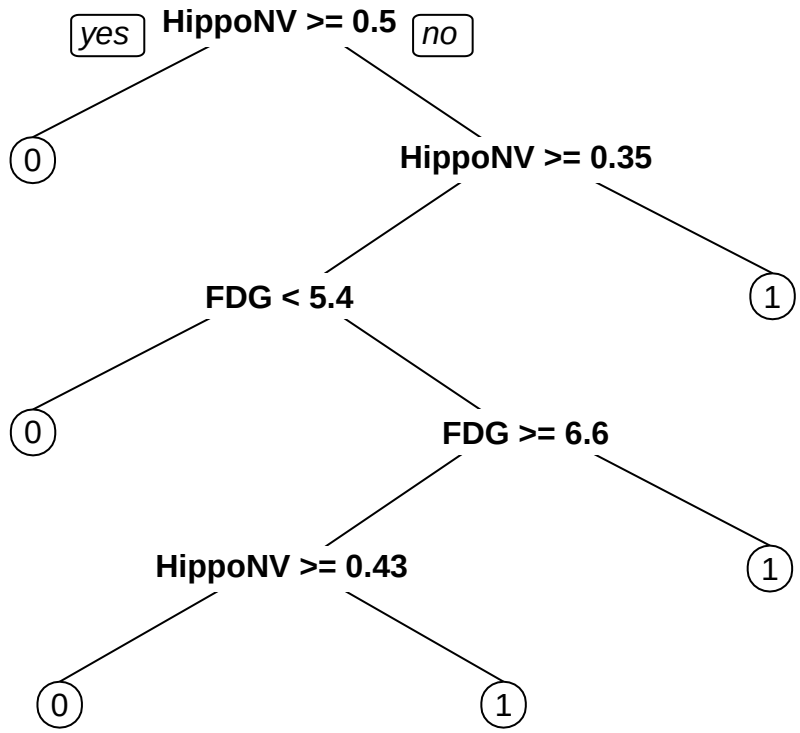
```r
set.seed(1)
train.ix <- sample(nrow(AD),floor( nrow(AD)/2) )
err.train.v <- NULL
err.test.v <- NULL
leaf.v <- NULL
for(i in seq(0.2,0,by=-0.005) ){
tree <- rpart( AGE_bin ~ ., data = AD[train.ix,], cp=i  )
pred.train <- predict(tree, AD[train.ix,],type="class")
pred.test <- predict(tree, AD[-train.ix,],type="class")
current.err.train <- length(which(pred.train != AD[train.ix,]$AGE_bin))/length(pred.train)
current.err.test <- length(which(pred.test != AD[-train.ix,]$AGE_bin))/length(pred.test)
err.train.v <- c(err.train.v, current.err.train)
err.test.v <- c(err.test.v, current.err.test)
leaf.v <- c(leaf.v, length(which(tree$frame$var == "<leaf>")))
}
err.mat <- as.data.frame( cbind( train_err = err.train.v, test_err = err.test.v , leaf_num = leaf.v ) )
err.mat$leaf_num <- as.factor( err.mat$leaf_num  )
err.mat <- unique(err.mat)
err.mat <- err.mat %>% gather(type, error, train_err,test_err)

# visualizing this
data.plot <- err.mat %>% mutate(type = factor(type))
ggplot(data.plot, aes(x=leaf_num, y=error, shape = type, color=type)) + geom_line() +
geom_point(size=3)
```

```
## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```



We plot a more optimal decision tree which is only dependent on the two most important variables.

```
tree_0.05 <- prune(tree,cp =0.0319, depth = 3)
prp(tree_0.05,nn.cex=1)
```



Both the logistic regression and the decision tree showed the importance of HippoNV as a predictor. The logistic model showed that PTGENDER, PTEDUCAT, and e4_1 are important predictors, while the decision tree showed that FDG is an important predictor. It isn't uncanny that the logistic model and decision tree identified different predicators as the most significant. However, both models showed strong evidence of the importance of the volume of the hippocampus in determining whether a person's age was above or below the mean age of those in the study.

Chapter 3, Exercise 2

Find two datasets from the UCI data repository or R datasets. Conduct a detailed analysis for both datasets using both logistic regression model and the tree model, e.g., for regression model, you may want to conduct model selection, model comparison, testing of the significance of the regression parameters, evaluation of the R-squared and significance of the model. Also comment on the application of your model on the context of the dataset you have selected.

Medical School Admission

The first dataset we chose to analyze was the MedGPA dataset from the Stat2Data package. This dataset contains data about medical school admission status and information on GPA and standardized test scores. A table that provides a description of the variables included in the data set is provided below.

Variable Name	Description
Accept	Status: A=accepted to medical school or D=denied admission
Acceptance	Indicator for Accept: 1=accepted or 0=denied
Sex	F=female or M=male
BCPM	fuel consumption miles per US gallon
GPA	College grade point average
VR	Verbal reasoning (subscore
PS	Physical sciences (subscore)
WS	Writing sample (subcore)
BS	Biological sciences (subscore)
MCAT	Score on the MCAT exam (sum of CR+PS+WS+BS)
Apps	Number of medical schools applied to

A logistic regression model was fitted using a backwards step variable selection. The final found the intercept, sex, GPA, PS, WS, and BS to be significant. Looking at the summary we can see that comparing males to females, males have a 2.84 increase in log odds of acceptance versus females. We also found that GPA, PS, and BS all have a negative log odds of admission for each unit of increase. Only WS had a positive log odds of admission for each unit of increase. This model can be used to assess a candidates probability of being accepted into medical school and can be used to give insight into what variables best increase their chance of acceptance.

```
df.ch3ex2.med <- read.csv("MedGPA.csv")
ch3ex2.med.log <- glm( Accept~.,family=binomial(link='logit'),data=df.ch3ex2.med[,-c(1,3)])
ch3ex2.med.log <- step(ch3ex2.med.log, direction = "backward", trace = 1)
```

```
## Start:  AIC=48.32
## Accept ~ Sex + BCPM + GPA + VR + PS + WS + BS + MCAT + Apps
##
##
## Step:  AIC=48.32
```

```
## Accept ~ Sex + BCPM + GPA + VR + PS + WS + BS + Apps
##
##        Df Deviance    AIC
## - VR    1   30.384 46.384
## - Apps  1   30.441 46.441
## - BCPM  1   31.158 47.158
## <none>      30.319 48.319
## - GPA   1   32.747 48.747
## - Sex   1   33.413 49.413
## - WS    1   35.568 51.568
## - PS    1   36.902 52.902
## - BS    1   44.269 60.269
##
## Step:  AIC=46.38
## Accept ~ Sex + BCPM + GPA + PS + WS + BS + Apps
##
##        Df Deviance    AIC
## - Apps  1   30.493 44.493
## - BCPM  1   31.286 45.286
## <none>      30.384 46.384
## - GPA   1   32.941 46.941
## - Sex   1   34.713 48.713
## - WS    1   35.658 49.658
## - PS    1   37.313 51.313
## - BS    1   44.281 58.281
##
## Step:  AIC=44.49
## Accept ~ Sex + BCPM + GPA + PS + WS + BS
##
##        Df Deviance    AIC
## - BCPM  1   31.321 43.321
## <none>      30.493 44.493
## - GPA   1   33.430 45.430
## - WS    1   36.171 48.171
## - Sex   1   36.854 48.854
## - PS    1   37.688 49.688
## - BS    1   44.308 56.308
##
## Step:  AIC=43.32
## Accept ~ Sex + GPA + PS + WS + BS
##
##        Df Deviance    AIC
## <none>      31.321 43.321
## - WS    1   36.644 46.644
## - PS    1   37.694 47.694
## - GPA   1   37.768 47.768
## - Sex   1   39.898 49.898
## - BS    1   44.423 54.423
```

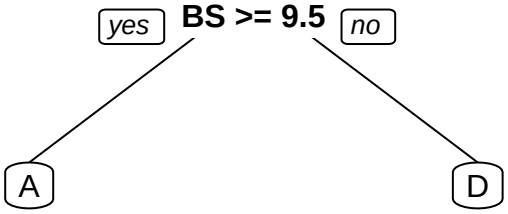
```
summary(ch3ex2.med.log)
```

```
##
## Call:
## glm(formula = Accept ~ Sex + GPA + PS + WS + BS, family = binomial(link = "logit"),
##     data = df.ch3ex2.med[, -c(1, 3)])
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.21963  -0.34653  -0.02646   0.43294   1.92771
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  39.4709    12.2145   3.231  0.00123 **
## SexM          2.8403     1.1581   2.453  0.01418 *
## GPA          -5.3344     2.4807  -2.150  0.03153 *
## PS           -1.0248     0.4723  -2.170  0.03003 *
## WS            0.7178     0.3497   2.053  0.04010 *
## BS           -1.7915     0.6435  -2.784  0.00537 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 74.192  on 53  degrees of freedom
## Residual deviance: 31.321  on 48  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 43.321
##
## Number of Fisher Scoring iterations: 7
```

Next a decision tree was fit on the same dataset. The tree only found BS to be the variable to split on. The tree does not have as much application in this context due to the limited data set size. If the data set size was larger, the tree would allow a participant to find the best values for each of the predictor variables that would best increase their chance of being accepted into medical school.

```
ch3ex2.med.tree <- rpart(Accept~., data=df.ch3ex2.med[,-c(1,3)], control = rpart.control(p = 0.0001))
prp(ch3ex2.med.tree, varlen=3)
```



Bad Health

The second dataset we chose to analyze was the BadHealth dataset from the COUNT package. This dataset contains data about a German health survey data for the year 1998. A table that provides a description of the variables included in the data set is provided below.

Variable Name	Description
Number of visits	Number of visits to doctor during 1998
bad health	1=patient claims to be in bad health; 0=not in bad health
age	age of patient: 20-60

A logistic regression model was fitted using a backwards step variable selection. The final found the intercept,

number of visits, and age to be significant. Looking at the summary we can see that for every unit increase in the number of visits to the doctor, there is an increase of 0.22 log odds of the patient claiming to be in bad health. For every unit increase in the age of the patient, there is an increase of 0.05 log odds of the patient claiming to be in bad health. This type of model can be used to assess the health of patients using easily accessible data and can be used in policy making.

```
df.ch3ex2.bh <- read.csv("badhealth.csv")
df.ch3ex2.bh$badh <- as.factor(df.ch3ex2.bh$badh)
ch3ex2.med.bh <- glm( badh~.,family=binomial(link='logit'),data=df.ch3ex2.bh[,-c(1)])
ch3ex2.med.bh <- step(ch3ex2.med.bh, direction = "backward", trace = 1)
```

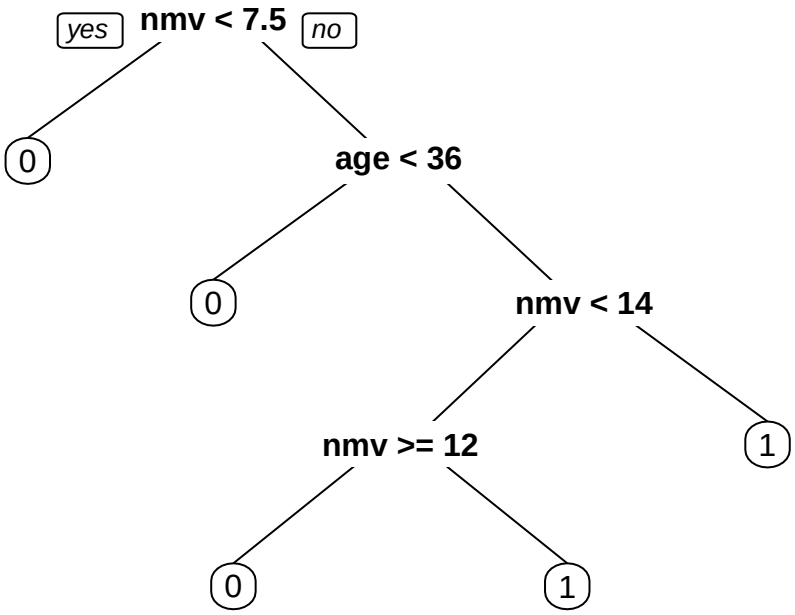
```
## Start:  AIC=609.43
## badh ~ numvisit + age
##
##            Df Deviance    AIC
## <none>          603.43 609.43
## - age       1   632.00 636.00
## - numvisit  1   687.69 691.69
```

```
summary(ch3ex2.med.bh)
```

```
##
## Call:
## glm(formula = badh ~ numvisit + age, family = binomial(link = "logit"),
##     data = df.ch3ex2.bh[, -c(1)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0553  -0.4302  -0.3258  -0.2503   2.7930
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.04184    0.44998 -11.205  < 2e-16 ***
## numvisit     0.22122    0.02628   8.419  < 2e-16 ***
## age          0.05281    0.01007   5.244 1.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 729.66  on 1126  degrees of freedom
## Residual deviance: 603.43  on 1124  degrees of freedom
## AIC: 609.43
##
## Number of Fisher Scoring iterations: 5
```

Next a decision tree was fit on the same dataset. The tree split on number of visits and on age. The tree found that if the number of visits is greater than 14 and the patients age was greater than 36, the patient most likely said they were in bad health. If the number of visits was less than 7.5 or the patient was less than the age of 36, the patient most likely reported they were not in bad health. This type of model can to identify an easy rule of assessing the overall health of a population using the number of visits to the doctor, which would be useful in policy making.

```
ch3ex2.bh.tree <- rpart(badh~., data=df.ch3ex2.bh[,-c(1)], control = rpart.control(p = 0.0001))
prp(ch3ex2.bh.tree, varlen=3)
```



Chapter 3, Exercise 3

Pick up any dataset you have used, and randomly split the data into two halves. Use one half to build the tree model and the regression model. Test the models' prediction performances on the second half. Report what you have found, adjust your way of model building, and suggest a strategy to find the model you consider as the best.

```
df.ch3ex3.bh <- read.csv("badhealth.csv")
# df.ch3ex2.bh$badh <- as.factor(df.ch3ex2.bh$badh)

#divide dataset into two
data <- df.ch3ex3.bh[,-c(1)]
sample_first_half <- sample(nrow(data),floor( nrow(data)/2) )

check<-data[sample_first_half,]
```

logistic regression model

```
check$badh <- as.factor(check$badh)

ch3ex3.bh_logit <- glm( badh~.,family=binomial(link='logit'),data=check[,-c(4)])
summary(ch3ex3.bh_logit)
```

```
##
## Call:
## glm(formula = badh ~ ., family = binomial(link = "logit"), data = check[,
##     -c(4)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9356  -0.4353  -0.3484  -0.2752   2.6888
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -4.44233    0.58716  -7.566 3.85e-14 ***
## numvisit     0.21613    0.03843   5.624 1.87e-08 ***
## age          0.03886    0.01350   2.878    0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 360.30  on 562  degrees of freedom
## Residual deviance: 312.34  on 560  degrees of freedom
## AIC: 318.34
##
## Number of Fisher Scoring iterations: 5
```

numvisit and age are significant as their p-value is less than 0.05. Out of total devience of 346.77, 346.77-294.17 = 52.6 could be explained by the predictor numvisit and age.

```
confint(ch3ex3.bh_logit)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %      97.5 %
## (Intercept) -5.64592675 -3.33608768
## numvisit     0.14180832  0.29291721
## age          0.01254417  0.06568594
```

```
library(aod)

wald.test(b = coef(ch3ex3.bh_logit),Sigma=vcov(ch3ex3.bh_logit), Terms=2)
```

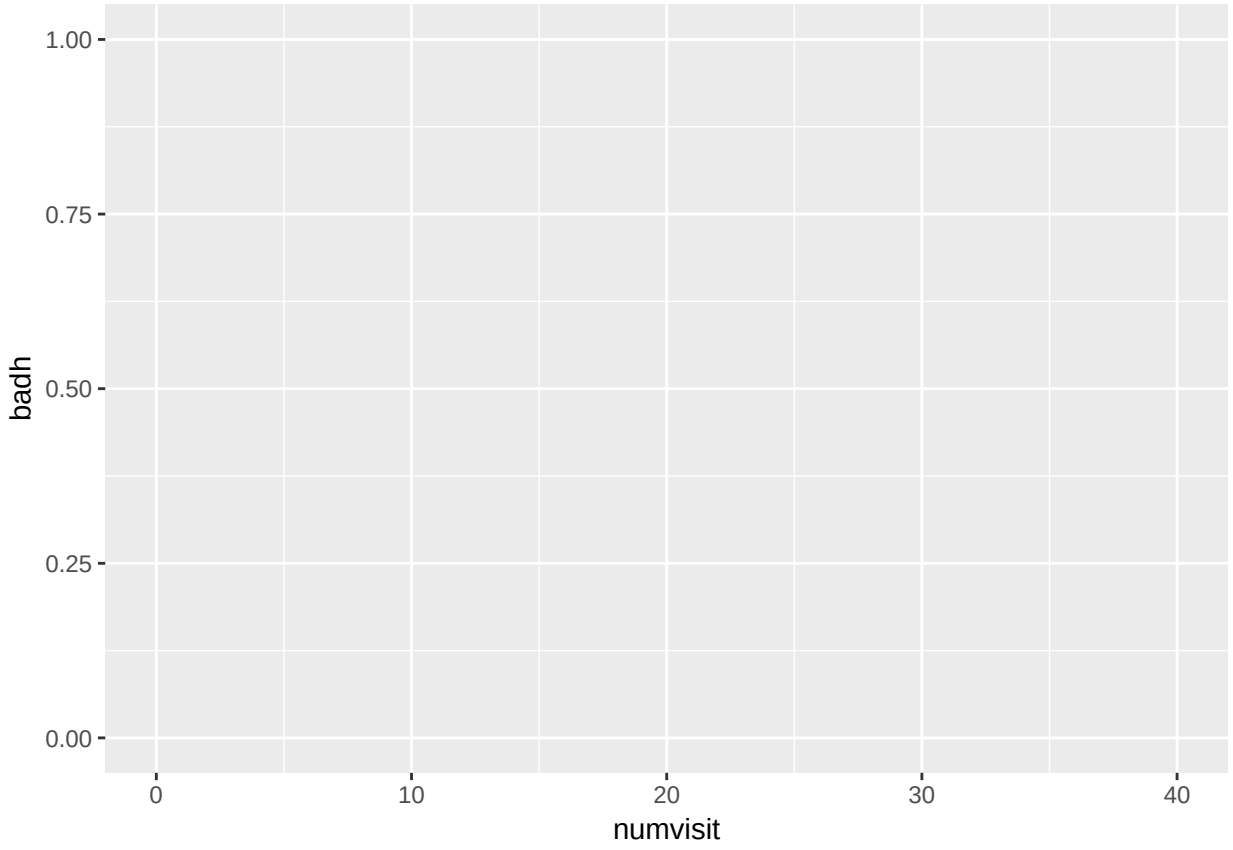
```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 31.6, df = 1, P(> X2) = 1.9e-08
```

```
check_2<-data[-sample_first_half,]
#because the nrow(check) = 563 and nrow(check_2) = 564, take one row out from check_2 to make both data
check_2<-check_2[1:(nrow(check_2)-1),]

# To predict on a given dataset
colnames(check_2) <- paste("",colnames(check_2),sep="")

# predict() uses all the temp values in dataset, including appended values
pred <- predict(ch3ex3.bh_logit, check_2, type = "link", se.fit = TRUE)
check_2$fit <- pred$fit
check_2$se.fit <- pred$se.fit
```

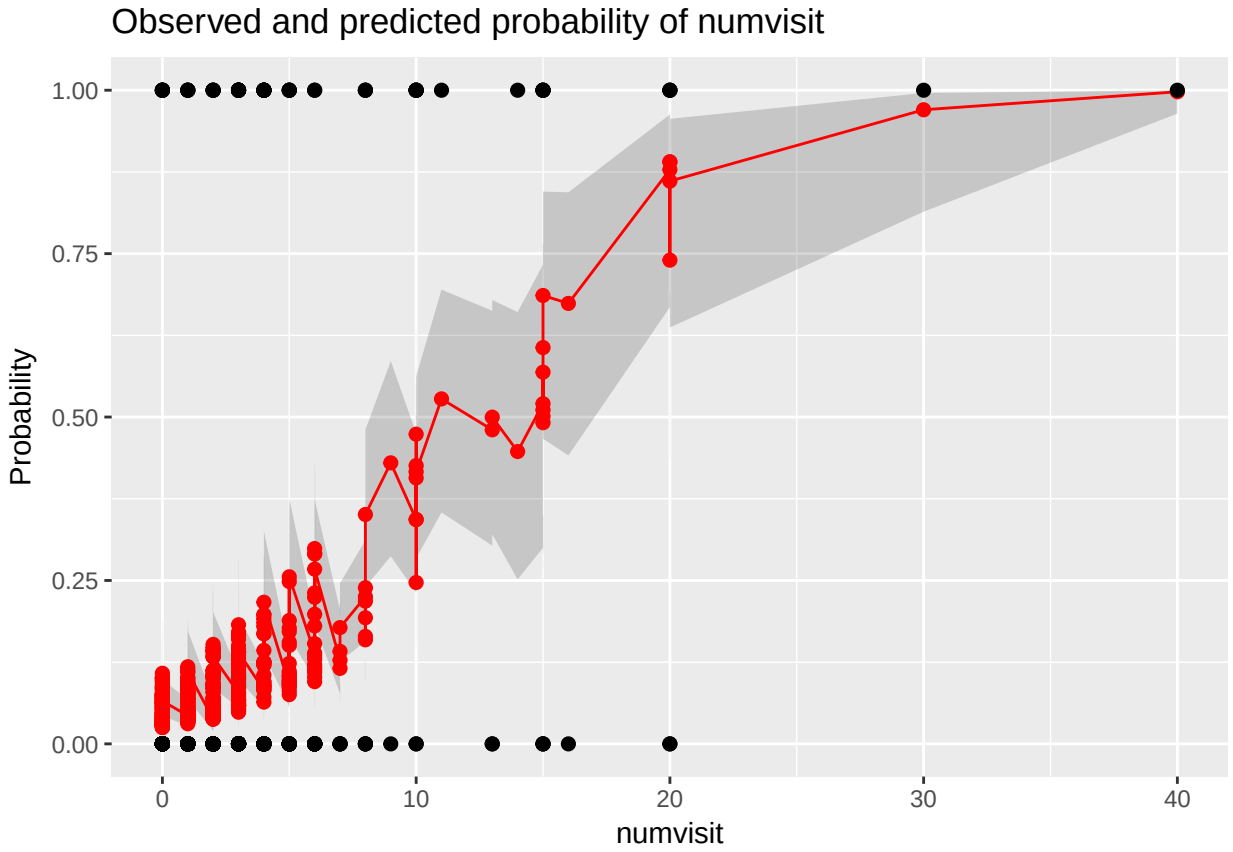
```
# CI for fitted values
check_2 <- within(check_2, {
 # added "fitted" to make predictions at appended temp values
 fitted = exp(fit) / (1 + exp(fit))
 fit.lower = exp(fit - 1.96 * se.fit) / (1 + exp(fit - 1.96 * se.fit))
 fit.upper = exp(fit + 1.96 * se.fit) / (1 + exp(fit + 1.96 * se.fit))
})
```

relationship with numvisit and badh

```r
# visualize the prediction
library(ggplot2)


newData <- check_2[order(check_2$numvisit),]
p <- ggplot(newData, aes(x = numvisit, y = badh))
print(p)
```




```r
# predicted curve and point-wise 95% CI
p <- p + geom_ribbon(aes(x = numvisit, ymin = fit.lower, ymax = fit.upper), alpha = 0.2)
p <- p + geom_line(aes(x = numvisit, y = fitted), colour="red")
# fitted values
p <- p + geom_point(aes(x = numvisit,y = fitted), size=2, colour="red")
# observed values
p <- p + geom_point(size = 2)
p <- p + ylab("Probability")
p <- p + labs(title = "Observed and predicted probability of numvisit")
print(p)
```

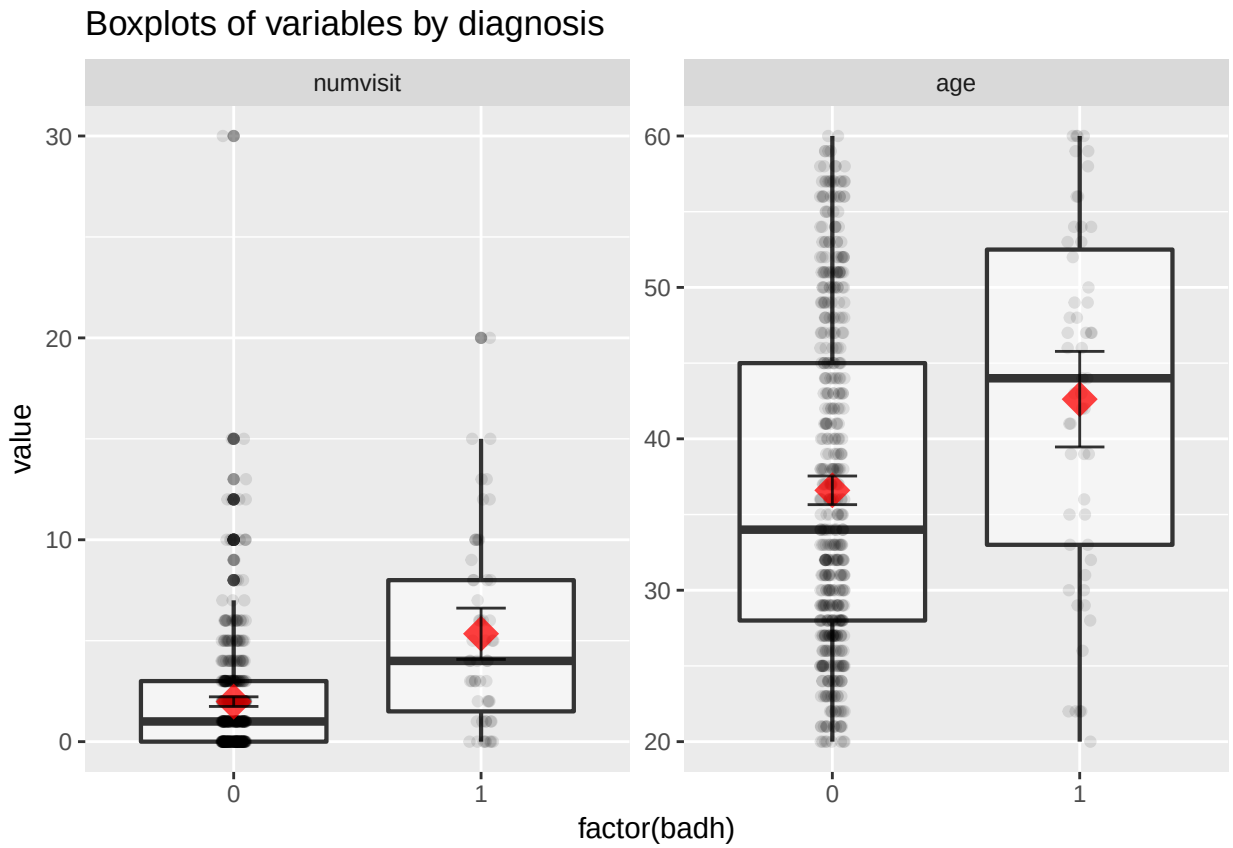


As the graph(relationship between numvisit and badh) shows a logit curve and the prediction confidences are fairly small as the graph shows tight 95% CIs.

Trying other variables

```
# install.packages("reshape2")
require(reshape2)
require(ggplot2)
check$ID = c(1:dim(check)[1])
AD.long <- melt(check[,], id.vars = c("ID", "badh"))

# Plot the data using ggplot
require(ggplot2)
p <- ggplot(AD.long, aes(x = factor(badh), y = value))
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0),
alpha = 0.1)
# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
 alpha = 0.75, colour = "red")

# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
 width = .2, alpha = 0.8)
p <- p + facet_wrap( ~ variable, scales = "free_y", ncol = 3)
p <- p + labs(title = "Boxplots of variables by diagnosis")
```

```r
print(p)
```



Both predictors numvisit and age seem to be able to classify two classes significantly.

Improvement for regression model

```r
ch3ex3.bh_logit_improved <- step(ch3ex3.bh_logit, direction = "backward", trace = 1)
```

```
## Start:  AIC=318.34
## badh ~ numvisit + age
##
##            Df Deviance    AIC
## <none>          312.34 318.34
## - age       1   320.71 324.71
## - numvisit  1   345.97 349.97
```

```r
summary(ch3ex3.bh_logit_improved)
```

```
##
## Call:
## glm(formula = badh ~ numvisit + age, family = binomial(link = "logit"),
##     data = check[, -c(4)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9356  -0.4353  -0.3484  -0.2752   2.6888
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -4.44233    0.58716  -7.566 3.85e-14 ***
## numvisit     0.21613    0.03843   5.624 1.87e-08 ***
## age          0.03886    0.01350   2.878    0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 360.30  on 562  degrees of freedom
## Residual deviance: 312.34  on 560  degrees of freedom
## AIC: 318.34
##
## Number of Fisher Scoring iterations: 5
```

Chi-square test for original logit regression model

```
# Test residual deviance for lack-of-fit (if > 0.10, little-to-no lack-of-fit)
dev.p.val <- 1 - pchisq(ch3ex3.bh_logit$deviance, ch3ex3.bh_logit$df.residual)

dev.p.val
```

```
## [1] 1
```

Chi-square test for improved logit regression model

```
# Test residual deviance for lack-of-fit (if > 0.10, little-to-no lack-of-fit)
dev.p.val_i <- 1 - pchisq(ch3ex3.bh_logit_improved$deviance, ch3ex3.bh_logit_improved$df.residual)

dev.p.val_i
```

```
## [1] 1
```

Both models show no lack of fit as the p-value is 1.

```
# coefficients and 95% CI
cbind(coef = coef(ch3ex3.bh_logit), confint(ch3ex3.bh_logit))
```

```
## Waiting for profiling to be done...
```

```
##                     coef       2.5 %      97.5 %
## (Intercept) -4.44232549 -5.64592675 -3.33608768
## numvisit     0.21613102  0.14180832  0.29291721
## age          0.03885959  0.01254417  0.06568594
```

```
cbind(coef = coef(ch3ex3.bh_logit_improved), confint(ch3ex3.bh_logit_improved))
```

```
## Waiting for profiling to be done...
```

```
##                     coef       2.5 %      97.5 %
## (Intercept) -4.44232549 -5.64592675 -3.33608768
## numvisit     0.21613102  0.14180832  0.29291721
## age          0.03885959  0.01254417  0.06568594
```

```
## odds ratios and 95% CI
exp(cbind(OR = coef(ch3ex3.bh_logit), confint(ch3ex3.bh_logit)))
```

```
## Waiting for profiling to be done...
```

```
##                      OR       2.5 %     97.5 %
## (Intercept) 0.01176854 0.003531874 0.03557587
## numvisit    1.24126500 1.152355745 1.34033182
```

```
## age         1.03962450 1.012623176 1.06789128
```

```
exp(cbind(OR = coef(ch3ex3.bh_logit_improved), confint(ch3ex3.bh_logit_improved)))
```

```
## Waiting for profiling to be done...
```

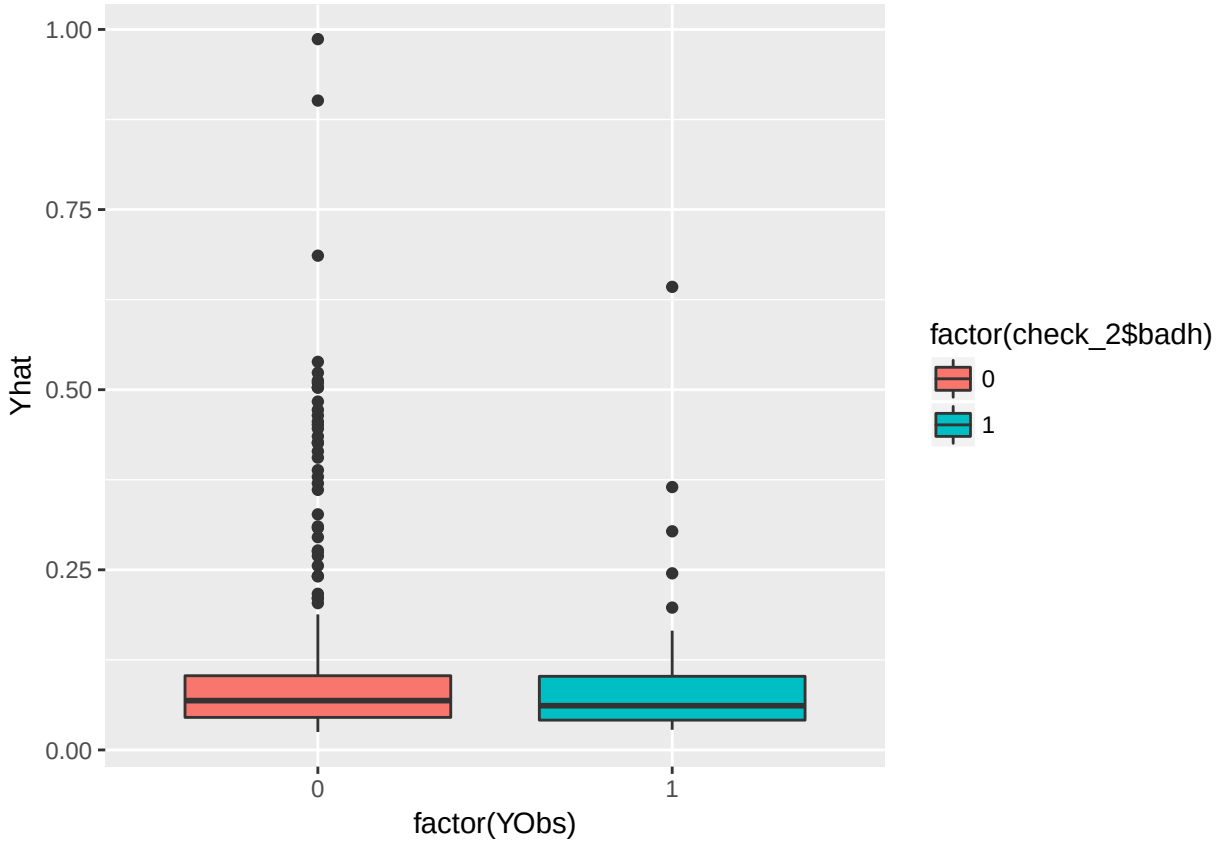
```
##                      OR       2.5 %     97.5 %
## (Intercept) 0.01176854 0.003531874 0.03557587
## numvisit    1.24126500 1.152355745 1.34033182
## age         1.03962450 1.012623176 1.06789128
```

```
# evaluate how well the model fits the data
# predicted probabilities
Yhat <- fitted(ch3ex3.bh_logit)
# the observed events
YObs <- as.numeric(check_2$badh)
# calculate the correlation between the predicted and observed
cor(Yhat,YObs)
```

```
## [1] -0.01653111
```

```
# visualize the correlation
tempData = cbind(Yhat,YObs)
require(ggplot2)
qplot(factor(YObs), Yhat, data = check_2,
      geom=c("boxplot"), fill = factor(check_2$badh),title="Prediction versus Observed")
```

```
## Warning: Ignoring unknown parameters: title
```

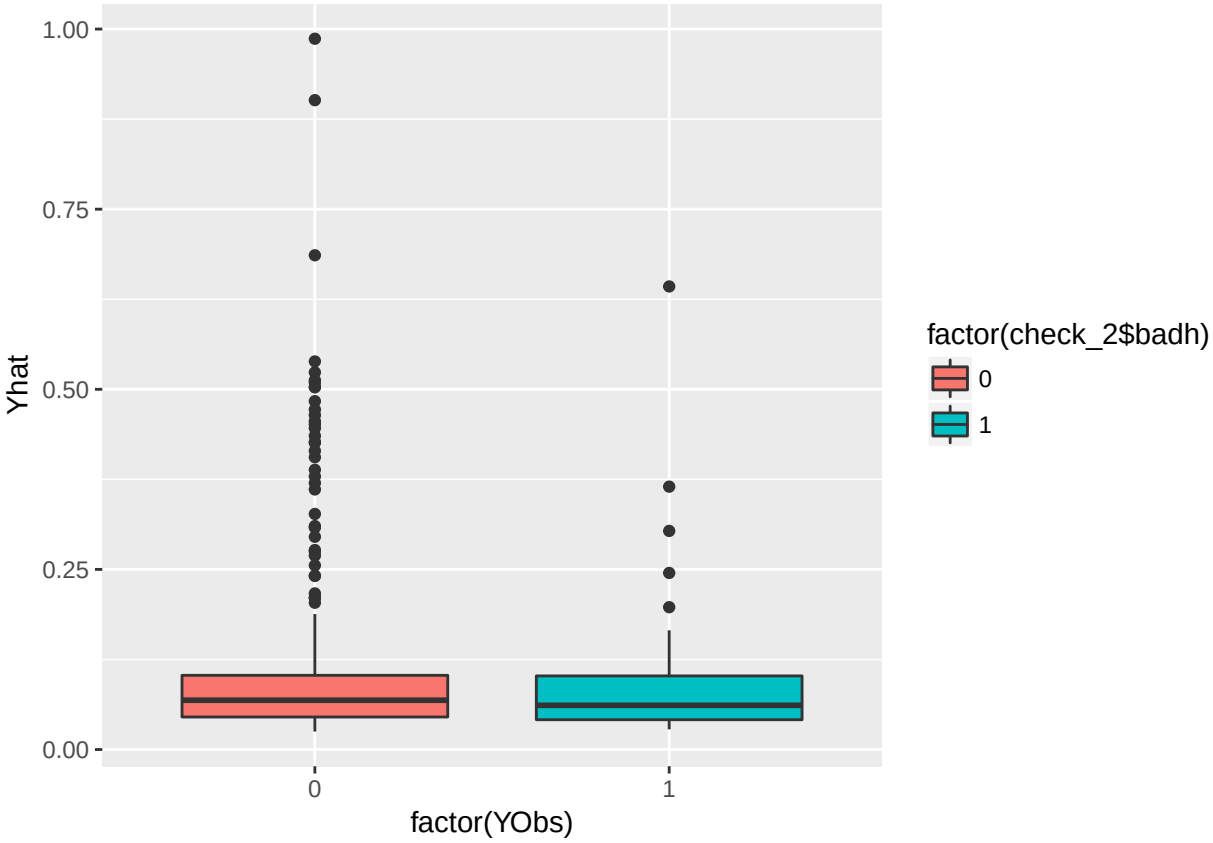


```
# evaluate how well the model fits the data
# predicted probabilities
Yhat <- fitted(ch3ex3.bh_logit_improved)
# the observed events
YObs <- as.numeric(check_2$badh)
# calculate the correlation between the predicted and observed
cor(Yhat,YObs)
```

```
## [1] -0.01653111
```

```
# visualize the correlation
tempData = cbind(Yhat,YObs)
require(ggplot2)
qplot(factor(YObs), Yhat, data = check_2,
      geom=c("boxplot"), fill = factor(check_2$badh),title="Prediction versus Observed")
```

```
## Warning: Ignoring unknown parameters: title
```



The result shows that the the model can not seperate the two classes significantly.

```
#Finding accuracy of the model
library(ROCR)
```

```
## Loading required package: gplots
```

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

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

```
##     lowess
```

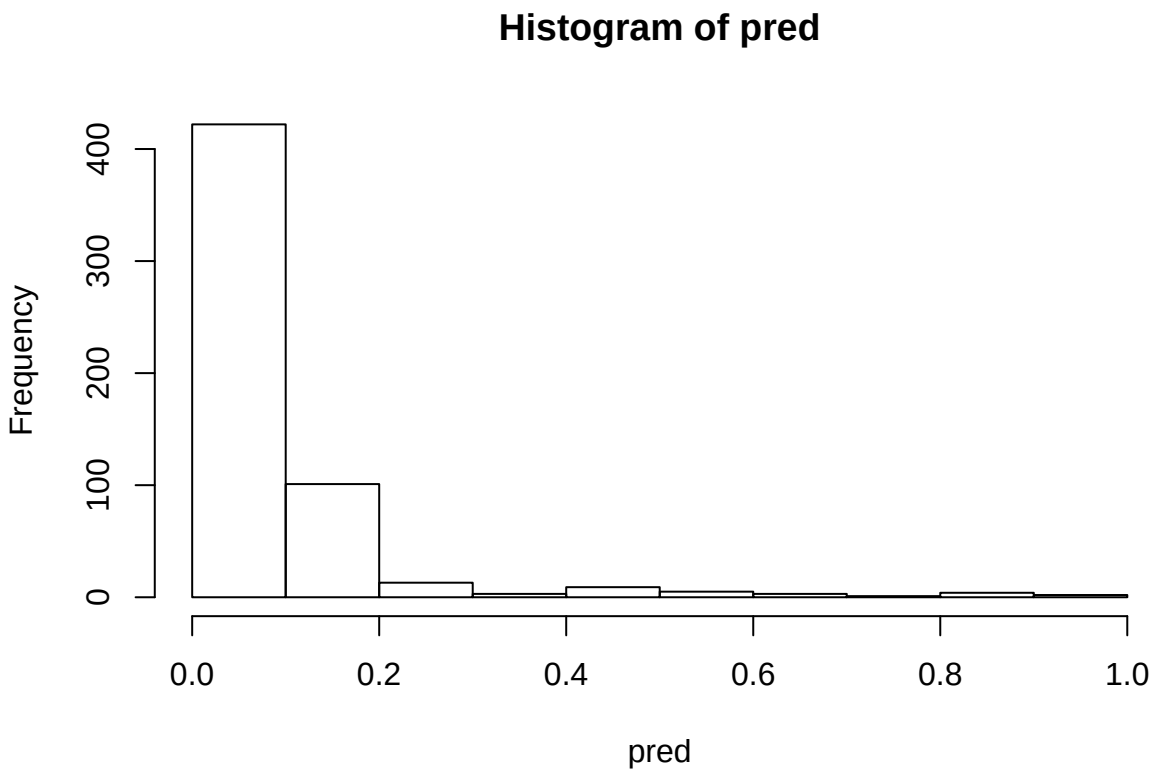
```
pred<- check_2$fitted
head(pred)
```

```
## [1] 0.8785439 0.6738374 0.8904447 0.5204901 0.5010718 0.4913577
```

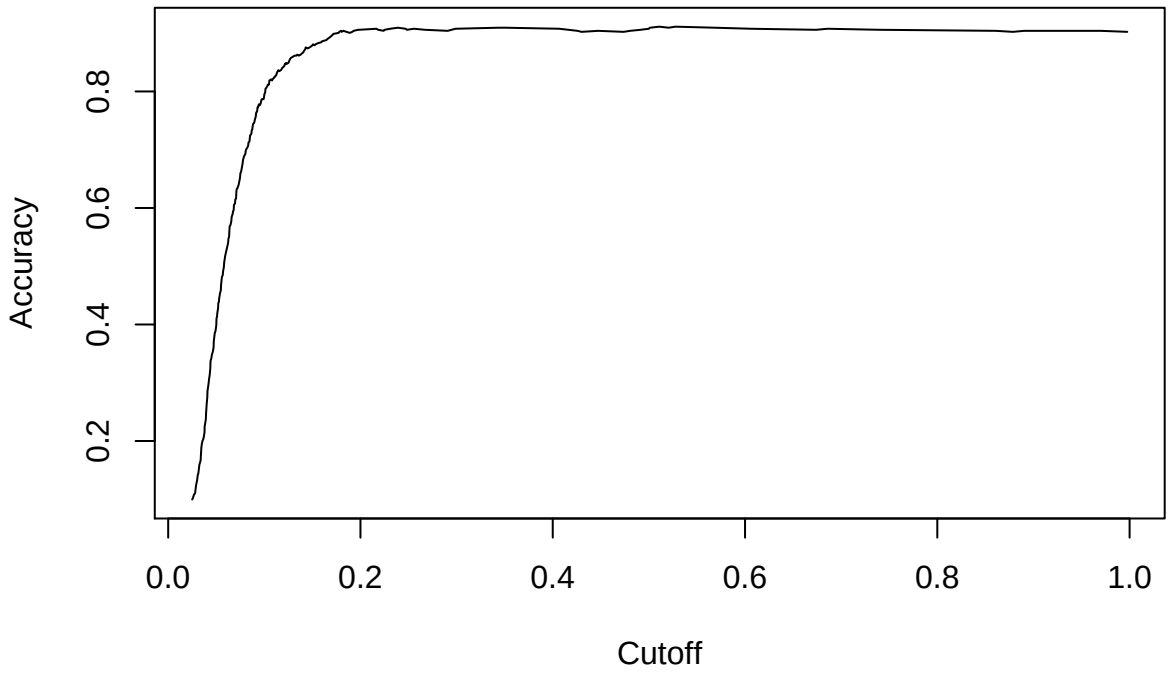
```
check_2$badh
```

```
##   [1] 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1
##  [36] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
##  [71] 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
## [141] 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [211] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0
## [246] 0 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [281] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
## [316] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0
## [351] 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1
## [386] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [421] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [456] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [491] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [526] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
## [561] 1 1 1
```

```
hist(pred)
```




```
pred <- prediction(pred, check_2$badh)
eval <- performance(pred, "acc")
plot(eval)
```



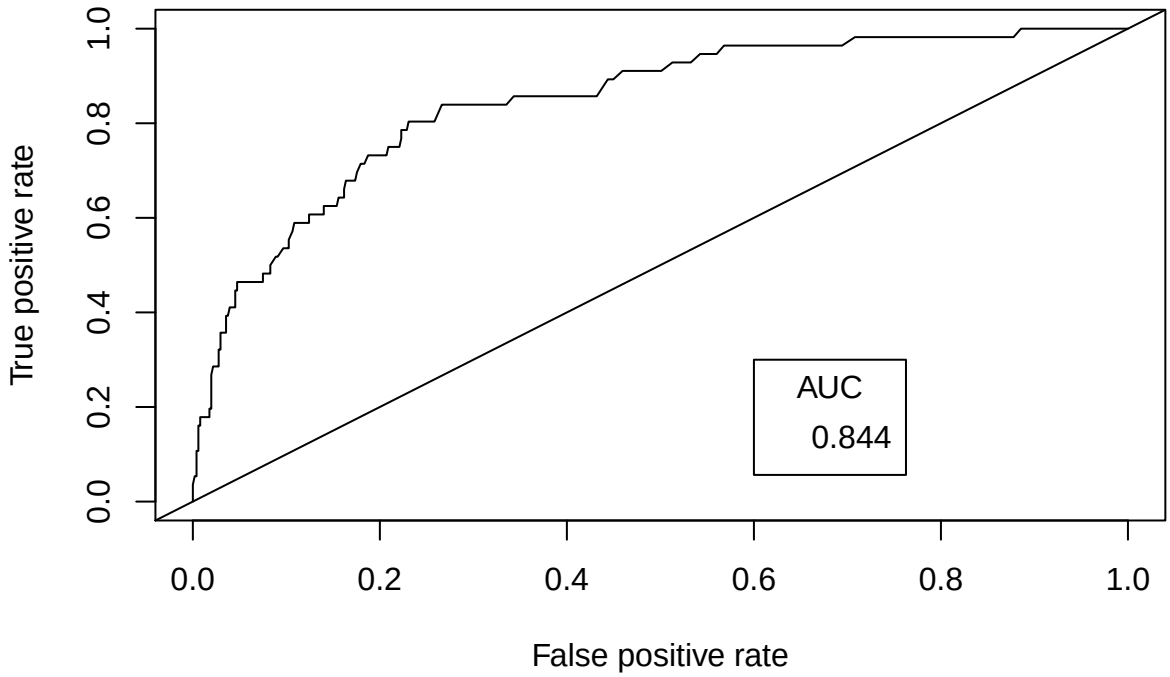

```
max <- which.max(slot(eval,"y.values")[[1]])
acc <- slot(eval,"y.values")[[1]][max]
cut <- slot(eval,"x.values")[[1]][max]

print(c(Accuracy = acc, Cuttoffvalue=cut))
```

```
##     Accuracy Cuttoffvalue
##    0.9111901    0.5277846
```

```
#Recievер Operating Chatasteristic(ROC) Curve

roc <- performance(pred, "tpr","fpr")
plot(roc)
abline(0,1)

#Area Under Curve(AUC)
auc <- performance(pred,"auc")
auc <- unlist(slot(auc,"y.values"))
auc
```

```
## [1] 0.8443047
```

```
legend(.6,.3,round(auc,digits=3),title="AUC")
```

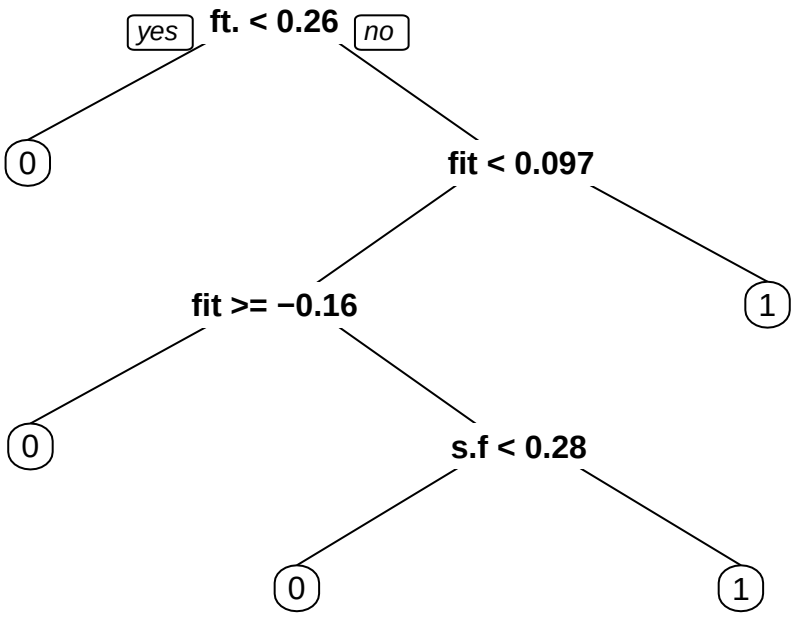



```
x <- c(1:nrow(check_2))
for (i in x){
if (check_2[i,]$se.fit >= cut){
  check_2[i,"pred"]  <- 1
}else{
  check_2[i,"pred"]  <- 0
}
}
```

Our step function didn't improve the

Tree model

```
check_2$badh <- as.factor(check_2$badh)
ch3ex3.bh.tree <- rpart(badh~., data=check_2)
prp(ch3ex3.bh.tree, varlen=3)
```




```
#prediction - tree
tree_pred_with_second_half<-predict(ch3ex3.bh.tree , check_2,type="class")
t <- table (predictions = tree_pred_with_second_half, actual = check_2$badh)
t
```

```
##            actual
## predictions   0   1
##           0 502  39
##           1   5  17
```

```
#accuracy matric
sum(diag(t))/sum(t)
```

```
## [1] 0.9218472
```

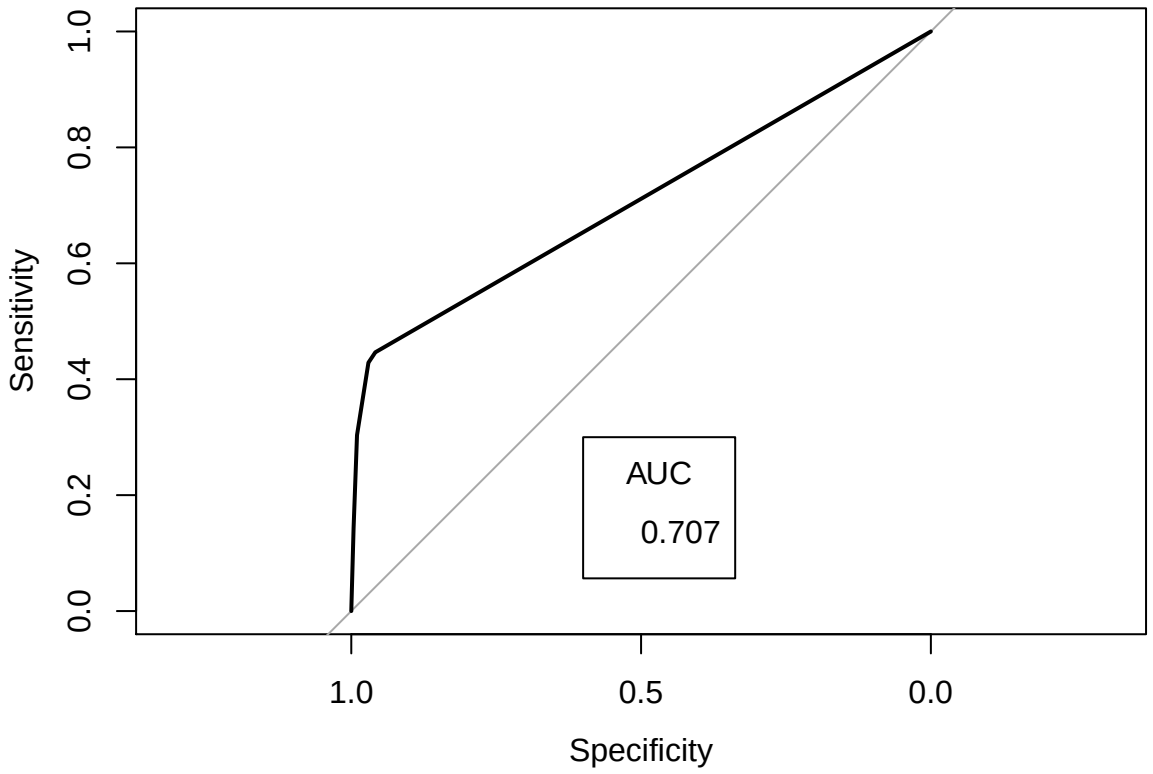
```
library(ROCR)
library(pROC)
```

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

```
##
## Attaching package: 'pROC'
```

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

```
tree_pred_with_second_half<-predict(ch3ex3.bh.tree,newdata = check_2,type ='prob')
auc <- auc(check_2$badh,tree_pred_with_second_half[,2])
plot(roc(check_2$badh,tree_pred_with_second_half[,2]))
legend(.6,.3,round(auc,digits=3),title="AUC")
```



Since Area under the curve for logit regression model is bigger, we chose logit regression model.