ECON 4460 Assignment

Vai Suliafu u0742607

November 9, 2016

PRELIMINARIES

# Loading in the prison data set  
cd <- read.csv(file = "crime506.csv", header = TRUE)  
  
# Loading the required packages  
# install.packages("corrplot")  
# install.packages("BMA")  
# install.packages("tree")  
# install.packages("party")  
library(BMA)

## Loading required package: survival

## Warning: package 'survival' was built under R version 3.2.5

## Loading required package: leaps

## Loading required package: robustbase

## Warning: package 'robustbase' was built under R version 3.2.5

##   
## Attaching package: 'robustbase'

## The following object is masked from 'package:survival':  
##   
## heart

## Loading required package: inline

## Loading required package: rrcov

## Warning: package 'rrcov' was built under R version 3.2.5

## Scalable Robust Estimators with High Breakdown Point (version 1.4-3)

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.2.5

## corrplot 0.84 loaded

library(lmtest)

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.2.5

##   
## Attaching package: 'zoo'

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

library(leaps)  
library(car)  
library(sandwich)  
library(tree)  
library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

library(leaps)  
library(corrplot)

Let's first examine the structure of the crime data (cd) so that we can create our test data and our sample data

str(cd)

## 'data.frame': 506 obs. of 44 variables:  
## $ anyreturn : int 0 1 1 1 0 0 0 0 1 1 ...  
## $ newcrime : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ techviol : int 0 1 1 1 0 0 0 0 1 1 ...  
## $ cons : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ agein06 : num 46.4 44.3 32.2 24.2 28.8 ...  
## $ male : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ priorinc : int 1 2 1 1 2 1 1 3 3 2 ...  
## $ racewh : int 1 1 1 0 1 1 1 0 1 1 ...  
## $ racehi : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ racepi : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ racena : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ raceaa : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ raceas : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ drug : int 0 0 0 0 1 1 0 0 0 1 ...  
## $ driving : int 0 0 1 0 0 0 1 0 0 0 ...  
## $ murder : int 1 0 0 0 0 0 0 0 0 0 ...  
## $ othercrime: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ person : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ property : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ sex : int 0 1 0 0 0 0 0 1 0 0 ...  
## $ weapons : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ hsdeg : int 1 1 1 0 0 1 1 1 1 1 ...  
## $ ged : int 0 0 1 0 1 0 0 0 0 0 ...  
## $ colldeg : int 1 0 0 0 0 0 0 1 0 0 ...  
## $ voccert : int 0 1 0 0 0 1 0 1 0 0 ...  
## $ prednone : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ predged : int 0 0 0 0 1 0 0 0 0 0 ...  
## $ predhs : int 0 0 1 1 0 1 1 0 0 1 ...  
## $ predcuv : int 1 1 0 0 0 1 0 1 0 0 ...  
## $ employed : int 1 1 0 0 1 1 1 1 1 1 ...  
## $ hrsweek : num 0 55 0 0 40 60 40 40 40 40 ...  
## $ wage : num 20 12 0 0 18 25 11 15 13 10.5 ...  
## $ healthben : int 1 0 0 0 0 0 0 1 0 0 ...  
## $ jtmanage : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ jtbuild : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ jtsales : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ jtoffice : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ jtconstr : int 0 0 0 0 1 1 1 0 0 1 ...  
## $ jtinstall : int 0 1 0 0 0 0 1 0 0 0 ...  
## $ jtprod : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ jttrans : int 1 0 0 0 0 0 0 1 1 0 ...  
## $ restdum : int 0 0 1 1 0 0 1 0 1 1 ...  
## $ chsupdum : int 0 1 1 0 0 0 1 1 1 0 ...  
## $ owncar : int 1 1 1 0 1 1 1 1 1 0 ...

We note that there are 506 observations and 44 variables. I will do a standard 80/20 split for training data and testing data. A simple calculation shows that 80% of a 506 sample is 405 observations, so my training data will the first 405 observations and my test data will be the remaining 101 observations.

train\_cd <- cd[1:405,]  
test\_cd <- cd[405:506,]

I will first look at linear models. I could "go into the kitchen" and cook up all kinds of models using any theories I might have, but this would be tedious. I prefer to take an exploratory approach and let the machine tell me what the best predictive linear model is. Therefore, I shall use the best subsets approach from the leap package to determine the best predictive linear model.

# using "anyreturn" as my target, I will allow up to 10 predictor variables, and take the singe best models with 10 predictor variables  
  
## choosing a max of 10 predictor variables is somewhat arbitrary. I would like consider all the variables, but this is too large of a calculation for my computer to handle.   
  
sub1 <- regsubsets(anyreturn ~ ., data = train\_cd, nbest = 1, nvmax = 10)

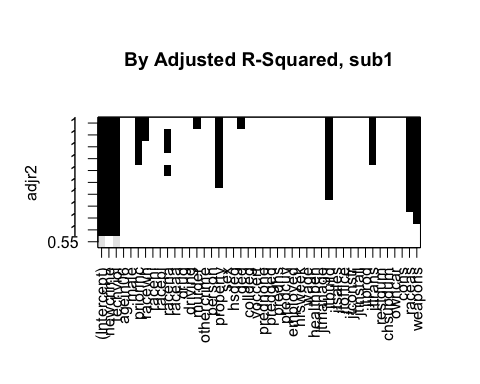
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,  
## force.in = force.in, : 3 linear dependencies found

## Reordering variables and trying again:

# Viewing a summary of the best subsets   
summary(sub1)

## Subset selection object  
## Call: regsubsets.formula(anyreturn ~ ., data = train\_cd, nbest = 1,   
## nvmax = 10)  
## 43 Variables (and intercept)  
## Forced in Forced out  
## newcrime FALSE FALSE  
## techviol FALSE FALSE  
## agein06 FALSE FALSE  
## male FALSE FALSE  
## priorinc FALSE FALSE  
## racewh FALSE FALSE  
## racehi FALSE FALSE  
## racepi FALSE FALSE  
## racena FALSE FALSE  
## raceaa FALSE FALSE  
## drug FALSE FALSE  
## driving FALSE FALSE  
## murder FALSE FALSE  
## othercrime FALSE FALSE  
## person FALSE FALSE  
## property FALSE FALSE  
## sex FALSE FALSE  
## hsdeg FALSE FALSE  
## ged FALSE FALSE  
## colldeg FALSE FALSE  
## voccert FALSE FALSE  
## prednone FALSE FALSE  
## predged FALSE FALSE  
## predhs FALSE FALSE  
## predcuv FALSE FALSE  
## employed FALSE FALSE  
## hrsweek FALSE FALSE  
## wage FALSE FALSE  
## healthben FALSE FALSE  
## jtmanage FALSE FALSE  
## jtbuild FALSE FALSE  
## jtsales FALSE FALSE  
## jtoffice FALSE FALSE  
## jtconstr FALSE FALSE  
## jtinstall FALSE FALSE  
## jtprod FALSE FALSE  
## jttrans FALSE FALSE  
## restdum FALSE FALSE  
## chsupdum FALSE FALSE  
## owncar FALSE FALSE  
## cons FALSE FALSE  
## raceas FALSE FALSE  
## weapons FALSE FALSE  
## 1 subsets of each size up to 11  
## Selection Algorithm: exhaustive  
## newcrime techviol cons agein06 male priorinc racewh racehi  
## 1 ( 1 ) " " "\*" " " " " " " " " " " " "   
## 2 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 3 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 4 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 5 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 6 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 7 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 8 ( 1 ) "\*" "\*" " " " " "\*" " " " " " "   
## 9 ( 1 ) "\*" "\*" " " " " "\*" " " " " " "   
## 10 ( 1 ) "\*" "\*" " " " " "\*" "\*" " " " "   
## 11 ( 1 ) "\*" "\*" " " " " "\*" "\*" " " " "   
## racepi racena raceaa raceas drug driving murder othercrime  
## 1 ( 1 ) " " " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " " " " " " " " " " "   
## 4 ( 1 ) " " " " " " " " " " " " " " " "   
## 5 ( 1 ) " " " " " " " " " " " " " " " "   
## 6 ( 1 ) " " " " " " " " " " " " " " "\*"   
## 7 ( 1 ) "\*" " " " " " " " " " " " " "\*"   
## 8 ( 1 ) " " " " " " " " " " " " " " "\*"   
## 9 ( 1 ) "\*" " " " " " " " " " " " " "\*"   
## 10 ( 1 ) "\*" " " " " " " " " " " " " "\*"   
## 11 ( 1 ) " " " " " " " " "\*" " " " " "\*"   
## person property sex weapons hsdeg ged colldeg voccert prednone  
## 1 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 4 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 5 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 6 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 7 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 8 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 9 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 10 ( 1 ) " " " " " " " " " " " " " " " " " "   
## 11 ( 1 ) " " " " "\*" " " " " " " " " " " " "   
## predged predhs predcuv employed hrsweek wage healthben jtmanage  
## 1 ( 1 ) " " " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " " " " " " " " " " "   
## 4 ( 1 ) " " " " " " " " " " " " " " " "   
## 5 ( 1 ) " " " " " " " " " " "\*" " " " "   
## 6 ( 1 ) " " " " " " " " " " "\*" " " " "   
## 7 ( 1 ) " " " " " " " " " " "\*" " " " "   
## 8 ( 1 ) " " " " " " " " " " "\*" " " " "   
## 9 ( 1 ) " " " " " " " " " " "\*" " " " "   
## 10 ( 1 ) " " " " " " " " " " "\*" " " " "   
## 11 ( 1 ) " " " " " " " " " " "\*" " " " "   
## jtbuild jtsales jtoffice jtconstr jtinstall jtprod jttrans  
## 1 ( 1 ) " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " " " " " " " " "   
## 4 ( 1 ) " " " " " " " " " " " " " "   
## 5 ( 1 ) " " " " " " " " " " " " " "   
## 6 ( 1 ) " " " " " " " " " " " " " "   
## 7 ( 1 ) " " " " " " " " " " " " " "   
## 8 ( 1 ) " " " " " " "\*" " " " " " "   
## 9 ( 1 ) " " " " " " "\*" " " " " " "   
## 10 ( 1 ) " " " " " " "\*" " " " " " "   
## 11 ( 1 ) " " " " " " "\*" " " " " " "   
## restdum chsupdum owncar  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " "\*" "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) " " "\*" "\*"   
## 9 ( 1 ) " " "\*" "\*"   
## 10 ( 1 ) " " "\*" "\*"   
## 11 ( 1 ) " " "\*" "\*"

# Making a visualization to compare the adjusted R-Squares of each suggested model  
plot(sub1, scale = "adjr2", main = "By Adjusted R-Squared, sub1")



# Now building an acutal linear model using the suggested predictor variables  
fit1 <- lm(anyreturn ~ newcrime + techviol + male + priorinc + drug + othercrime + sex + jtconstr + chsupdum + owncar, data = train\_cd)  
  
# Viewing the summary of this linear model that was built by the suggestions of the best subset method  
summary(fit1)

##   
## Call:  
## lm(formula = anyreturn ~ newcrime + techviol + male + priorinc +   
## drug + othercrime + sex + jtconstr + chsupdum + owncar, data = train\_cd)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.949e-15 -1.340e-16 -5.070e-17 4.730e-17 2.314e-14   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.589e-15 2.193e-16 7.245e+00 2.29e-12 \*\*\*  
## newcrime 1.000e+00 1.982e-16 5.045e+15 < 2e-16 \*\*\*  
## techviol 1.000e+00 1.498e-16 6.675e+15 < 2e-16 \*\*\*  
## male 8.436e-17 2.116e-16 3.990e-01 0.690   
## priorinc -1.373e-17 4.478e-17 -3.070e-01 0.759   
## drug -6.363e-17 1.610e-16 -3.950e-01 0.693   
## othercrime -7.057e-17 4.862e-16 -1.450e-01 0.885   
## sex -1.149e-16 1.627e-16 -7.060e-01 0.480   
## jtconstr -7.109e-17 1.397e-16 -5.090e-01 0.611   
## chsupdum -1.300e-16 1.369e-16 -9.500e-01 0.343   
## owncar 8.333e-17 1.297e-16 6.430e-01 0.521   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.261e-15 on 394 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 6.136e+30 on 10 and 394 DF, p-value: < 2.2e-16

Now, the adjusted R-squared of this linear model is 1, which suggest that it can perfectly predict the possibility that a previous inmate will return to prison. This 'perfection' of the model sounds extremely suspicious, so of course we have to test it on our training data.

# Generating predictions using the test data  
pred\_lm <- predict(fit1, newdata = test\_cd)  
  
# Adding this vector of predictions to the test dataset  
test\_cd$pred\_lm <- pred\_lm  
  
# Since linear models will give you an exact prediction, it makes sense to use the function floor(), so that we have binary predictions.   
## Note that I visually inspected the data set to ensure that there were no values such as 0.99 that would be 'incorrectly' rounded by the floor function.   
  
test\_cd$pred\_lm <- floor(test\_cd$pred\_lm)  
  
# Comparing the predictions of anyreturn to the actual values of any return  
mean(test\_cd$anyreturn - test\_cd$pred\_lm)

## [1] 0

This shows that there were zero misclassification errors from the linear model suggested by best subsets. However, I also conclude that I am unsure about the meaning of some of the predictor variables. It could be that some, or even many of the predictor variables used were proxies for anyreturn. In other words, it is possible that I was basically using anyreturn to predict anyreturn. This stresses the importance of knowing the context behind the data before making any recommendations of models.

If I assume that none of the predictor variables were proxies for anyreturn, I can conclude that a linear model is extremely powerful in predicting anyreturn. If I cannot say that the predictor variables are not proxies for anyreturn, I really can't reach only conclusion about this model.

Now, lets look at the predictive power of a logit model.

# I would like to use the same predictor variables as I used for the linear model  
logit1 <- glm(anyreturn ~ newcrime + techviol + male + priorinc + drug + othercrime + sex + jtconstr + chsupdum + owncar, data = train\_cd, family = "binomial")

## Warning: glm.fit: algorithm did not converge

# Viewing a summary of this logit model  
summary(logit1)

##   
## Call:  
## glm(formula = anyreturn ~ newcrime + techviol + male + priorinc +   
## drug + othercrime + sex + jtconstr + chsupdum + owncar, family = "binomial",   
## data = train\_cd)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.409e-06 -2.409e-06 -2.409e-06 2.409e-06 2.409e-06   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.657e+01 6.193e+04 0.000 1.000  
## newcrime 5.313e+01 5.597e+04 0.001 0.999  
## techviol 5.313e+01 4.231e+04 0.001 0.999  
## male 3.569e-12 5.976e+04 0.000 1.000  
## priorinc -1.027e-11 1.264e+04 0.000 1.000  
## drug -3.259e-12 4.547e+04 0.000 1.000  
## othercrime -7.929e-12 1.373e+05 0.000 1.000  
## sex -2.437e-12 4.595e+04 0.000 1.000  
## jtconstr -7.933e-12 3.946e+04 0.000 1.000  
## chsupdum 1.827e-12 3.866e+04 0.000 1.000  
## owncar -3.963e-14 3.662e+04 0.000 1.000  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5.4672e+02 on 404 degrees of freedom  
## Residual deviance: 2.3496e-09 on 394 degrees of freedom  
## AIC: 22  
##   
## Number of Fisher Scoring iterations: 25

# Now we have to use this model to generate some actual predictions  
predict\_logit <- predict(logit1, newdata = test\_cd)  
  
# Now I have to generate some simplified predictions  
## ie, I have to make the predictions binary   
### I have decided that I will send values less than 0 to 0, and values greater than or equal to 0 to 1  
predict\_logit\_simple = ifelse(predict\_logit < 0, 0, 1)  
  
# making a cross tab of the logit model predictions and actual return to prison  
print(table(predict\_logit\_simple, test\_cd$anyreturn))

##   
## predict\_logit\_simple 0 1  
## 0 44 0  
## 1 0 58

As expected, the logistic model predicted return to prison with 100% accuracy. That being said, I reach the same conclusion that I did for the linear model. I am extremely skeptical about these results. I think there is either overfitting or I am using a proxy for anyreturn as a predictor.

Now, lets look at the cart/tree model.

tree1 <- tree(as.factor(anyreturn) ~ ., data = train\_cd)  
  
# viewing a summary of the tree model  
summary(tree1)

##   
## Classification tree:  
## tree(formula = as.factor(anyreturn) ~ ., data = train\_cd)  
## Variables actually used in tree construction:  
## [1] "techviol" "newcrime"  
## Number of terminal nodes: 3   
## Residual mean deviance: 0 = 0 / 402   
## Misclassification error rate: 0 = 0 / 405

Alas, the tree model has revealed the 'issues' with our previous models. It looks as if they were all relying solely on the predictor variables techviol and newcrime. The data shows that their return to prison is dependent on the classifcation of their techviolation. If it is classified as a new crime, they return to prison. If it is not classified as a new crime, they do not return to prison (from this techviol).

In context, this is not really helpful. Policymakers would likely be interested in the probability that previous inmates will return to prison BEFORE they commit a new crime. Sure, you could use this study to re-examine what kind of activities are classified as techviols, but the political feasibility of redefining these terms seems improbably. Instead, we should reevaluate our previous models and ensure that the predictor variables techviol and newcrime are not used in the model, so that the results of the study can suggest more feasible actions.

For example, a tree model without techviol and anycrime is less accurate of course, but probably more useful:

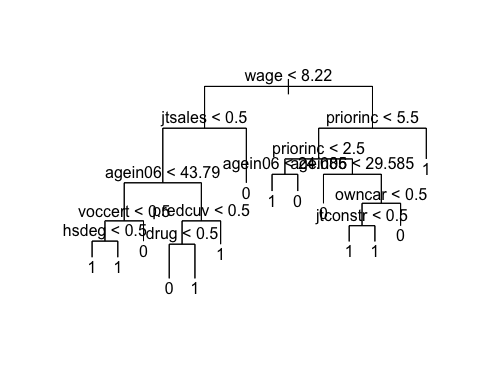
tree2 <- tree(as.factor(anyreturn) ~ .-techviol - newcrime, data = train\_cd)  
  
summary(tree2)

##   
## Classification tree:  
## tree(formula = as.factor(anyreturn) ~ . - techviol - newcrime,   
## data = train\_cd)  
## Variables actually used in tree construction:  
## [1] "wage" "jtsales" "agein06" "voccert" "hsdeg" "predcuv"   
## [7] "drug" "priorinc" "owncar" "jtconstr"  
## Number of terminal nodes: 14   
## Residual mean deviance: 1.056 = 413 / 391   
## Misclassification error rate: 0.2519 = 102 / 405

predcrime <- predict(tree2, newdata = test\_cd)  
  
predout <- cbind(predcrime, test\_cd$anyreturn)  
  
head(predout)

## 0 1   
## 405 0.1578947 0.8421053 1  
## 406 0.4285714 0.5714286 0  
## 407 1.0000000 0.0000000 0  
## 408 0.3833333 0.6166667 0  
## 409 0.3833333 0.6166667 0  
## 410 0.6923077 0.3076923 1

plot(tree2)  
text(tree2)

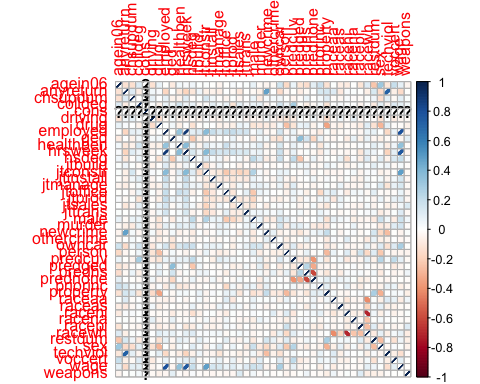


Anyway, let us move on to Bayesian Model Averaging. As a starting point, it would be wise to look at the correlations between all variables in the dataset.

c\_cors <- cor(cd)

## Warning in cor(cd): the standard deviation is zero

corrplot(c\_cors, method = "ellipse", order = "alphabet")



Now let's do our bayesian model averaging.

X <- cd[,1]  
Y <- cd[,2:44]  
  
cdbma <- bicreg(Y,X, OR = 250)

## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense  
  
## Warning: attempting model selection on an essentially perfect fit is  
## nonsense

summary(cdbma)

##   
## Call:  
## bicreg(x = Y, y = X, OR = 250)  
##   
##   
## 29 models were selected  
## Best 5 models (cumulative posterior probability = 0.5247 ):   
##   
## p!=0 EV SD model 1 model 2 model 3   
## Intercept 100 2.811e-16 1.465e-16 2.764e-16 4.343e-16 3.554e-16  
## newcrime 100 1.000e+00 3.466e-16 1.000e+00 1.000e+00 1.000e+00  
## techviol 100 1.000e+00 2.363e-16 1.000e+00 1.000e+00 1.000e+00  
## agein06 2 -7.335e-20 1.337e-18 . -3.704e-18 .   
## priorinc 2 1.070e-18 1.178e-17 . . .   
## racehi 2 3.557e-18 4.358e-17 . . .   
## racena 2 2.952e-18 9.441e-17 . . .   
## raceaa 2 4.840e-18 6.442e-17 . . .   
## drug 2 4.633e-18 4.426e-17 . . .   
## driving 2 -2.215e-17 1.646e-16 . . .   
## murder 2 -1.412e-17 1.283e-16 . . .   
## property 2 3.988e-18 4.067e-17 . . .   
## sex 2 -2.969e-18 3.864e-17 . . .   
## hsdeg 2 -4.483e-18 4.234e-17 . . .   
## ged 2 -2.097e-18 3.052e-17 . . .   
## colldeg 2 -2.090e-18 4.704e-17 . . .   
## voccert 2 -3.109e-18 3.857e-17 . . .   
## prednone 2 3.609e-18 3.849e-17 . . .   
## predged 2 3.764e-18 4.058e-17 . . .   
## predhs 2 -7.897e-20 2.566e-17 . . .   
## predcuv 2 -2.261e-18 3.224e-17 . . .   
## jtmanage 2 9.253e-19 6.230e-17 . . .   
## jtbuild 2 4.035e-18 7.325e-17 . . .   
## jtsales 2 2.320e-18 5.216e-17 . . .   
## jtoffice 2 2.896e-18 4.626e-17 . . .   
## jtconstr 2 3.804e-18 3.869e-17 . . .   
## jtinstall 2 -8.007e-18 7.017e-17 . . .   
## jtprod 2 2.907e-18 4.487e-17 . . .   
## restdum 2 2.185e-20 2.598e-17 . . .   
## chsupdum 2 -6.164e-18 5.123e-17 . . -3.113e-16  
## owncar 2 -7.857e-18 6.117e-17 . . .   
##   
## nVar 2 3 3   
## r2 0.999 0.999 0.999   
## BIC -3.483e+03 -3.477e+03 -3.477e+03  
## post prob 0.445 0.020 0.020   
## model 4 model 5   
## Intercept 2.764e-16 3.554e-16  
## newcrime 1.000e+00 1.000e+00  
## techviol 1.000e+00 1.000e+00  
## agein06 . .   
## priorinc . .   
## racehi . .   
## racena . .   
## raceaa . .   
## drug . .   
## driving . -1.119e-15  
## murder . .   
## property . .   
## sex . .   
## hsdeg . .   
## ged . .   
## colldeg -1.055e-16 .   
## voccert . .   
## prednone . .   
## predged . .   
## predhs . .   
## predcuv . .   
## jtmanage . .   
## jtbuild . .   
## jtsales . .   
## jtoffice . .   
## jtconstr . .   
## jtinstall . .   
## jtprod . .   
## restdum . .   
## chsupdum . .   
## owncar . .   
##   
## nVar 3 3   
## r2 0.999 0.999   
## BIC -3.477e+03 -3.477e+03  
## post prob 0.020 0.020

Now let's do our bayesian model averaging, excluding techviol and newcrime.

Y2 <- cd[,4:44]  
  
cdbma2 <- bicreg(Y2, X, OR = 250)  
  
summary(cdbma2)

##   
## Call:  
## bicreg(x = Y2, y = X, OR = 250)  
##   
##   
## 1095 models were selected  
## Best 5 models (cumulative posterior probability = 0.048 ):   
##   
## p!=0 EV SD model 1 model 2   
## Intercept 100.0 6.049e-01 0.1424407 0.660919 0.684490  
## agein06 65.3 -4.015e-03 0.0034470 -0.006492 -0.006768  
## male 0.0 9.537e-07 0.0007635 . .   
## priorinc 82.5 3.807e-02 0.0226184 0.048016 0.048591  
## racewh 0.1 -5.463e-05 0.0023168 . .   
## racehi 0.0 6.691e-06 0.0009436 . .   
## racepi 0.0 -1.365e-05 0.0021483 . .   
## racena 0.0 7.748e-06 0.0020147 . .   
## raceaa 57.6 1.329e-01 0.1329566 0.235526 0.230324  
## drug 10.9 1.149e-02 0.0372547 . .   
## driving 0.8 9.665e-04 0.0136243 . .   
## person 8.2 -9.302e-03 0.0355703 . .   
## property 53.7 7.398e-02 0.0792836 0.137344 0.133169  
## hsdeg 10.8 -1.009e-02 0.0332217 . .   
## voccert 0.4 -2.348e-04 0.0050334 . .   
## prednone 0.0 -4.753e-06 0.0007945 . .   
## predged 22.5 -2.485e-02 0.0524577 . .   
## predhs 0.1 -2.434e-05 0.0015160 . .   
## predcuv 0.0 7.415e-06 0.0008944 . .   
## employed 50.5 -9.887e-02 0.1045609 . -0.190603  
## hrsweek 49.4 -2.241e-03 0.0024184 -0.004395 .   
## wage 0.2 -1.849e-05 0.0005227 . .   
## jtmanage 1.6 -2.271e-03 0.0219796 . .   
## jtbuild 4.6 8.415e-03 0.0453389 . .   
## jtsales 5.9 -8.007e-03 0.0376777 . .   
## jtoffice 0.6 4.904e-04 0.0084240 . .   
## jtconstr 0.0 1.772e-06 0.0006377 . .   
## jtinstall 0.0 -5.758e-06 0.0010226 . .   
## restdum 47.6 5.537e-02 0.0662579 . .   
## chsupdum 50.1 5.793e-02 0.0665227 . .   
## owncar 38.5 -4.111e-02 0.0590584 . .   
##   
## nVar 5 5   
## r2 0.098 0.097   
## BIC -20.938750 -20.624774  
## post prob 0.012 0.011   
## model 3 model 4 model 5   
## Intercept 0.700821 0.675348 0.639750  
## agein06 -0.006640 -0.006353 -0.006242  
## male . . .   
## priorinc 0.046907 0.046331 0.052515  
## racewh . . .   
## racehi . . .   
## racepi . . .   
## racena . . .   
## raceaa . . 0.235130  
## drug . . .   
## driving . . .   
## person . . .   
## property 0.138269 0.142630 .   
## hsdeg . . .   
## voccert . . .   
## prednone . . .   
## predged . . .   
## predhs . . .   
## predcuv . . .   
## employed -0.197823 . -0.197012  
## hrsweek . -0.004518 .   
## wage . . .   
## jtmanage . . .   
## jtbuild . . .   
## jtsales . . .   
## jtoffice . . .   
## jtconstr . . .   
## jtinstall . . .   
## restdum . . 0.116015  
## chsupdum . . .   
## owncar . . .   
##   
## nVar 4 4 5   
## r2 0.085 0.085 0.096   
## BIC -20.197313 -20.191781 -20.030995  
## post prob 0.009 0.009 0.008

As expected, the linear models without techviol and newcrime are significantly less powerful than the tree model without techviol and newcrime.

Our final conclusion is that it's imperative to understand the context of the data before any models can be built and any conclusions can be reached. Furthermore, the data is really the sole determinant of which model will work best, so it is wise to familiarize yourself with the various modeling techniques. Our 'fair' tree model was about 75% accurate, and our bayesian model averaging suggests that the best model is only about 10% accurate.