Set up the working directory as appropriate. The following code will evaluate the current working directory. One could place the data files in this default location, or set the working directory with the ‘setwd()’ command. Please see help(setwd) for more information.

getwd()

Importing the SC SAC data. As these are Microsoft Excel files, we will open using the ‘gdata’ package. To install the package, please use install.packages(‘gdata’). For more information on installing packages, please see help(install.packages).

library(gdata)
sc\_sac <- read.xls(xls = 'sc\_sac\_data.csv', header = TRUE)

Importing the ICPSR data.

icpsr <- read.xls(xls = 'icpsr.csv', header = TRUE)

Exctracting 2011 SC SAC data and South Carolina ICPSR for comparison.

sc\_sac11 <- sc\_sac[sc\_sac$year == 2011 , ]
icpsr\_sc <- icpsr[icpsr$state == 45 , ]

Creating basic density plots in ggplot2 of outcome variables of interest.

library(ggplot2)
f <- ggplot(data = sc\_sac11, aes(x = smart\_total))
f <- f + geom\_density(fill = 'blue')
f <- f + theme\_minimal()
f <- f + ylim(0, 0.007)
f <- f + labs(title = 'SC SAC Data',
 subtitle = 'Violent Crime Smart Total',
 x = 'Cases',
 y = 'Density')

g <- ggplot(data = icpsr\_sc, aes(x = agg\_assault\_arrest))
g <- g + geom\_density(fill = 'red')
g <- g + theme\_minimal()
g <- g + ylim(0, 0.007)
g <- g + labs(title = 'SC (ICPSR) Data',
 subtitle = 'Aggravated Assaults',
 x = 'Arrests',
 y = 'Density')

## making side-by-side plots
library(gridExtra)
grid.arrange(f, g, ncol = 2)

Summary / descriptive statistics of both dependent variables.

summary(sc\_sac11$violent\_crime\_smart\_total)
sd(sc\_sac1111$violent\_crime\_smart\_total)
summary(icpsr\_sc$agg\_assault\_arrest)
sd(icpsr\_sc$agg\_assault\_arrest)

Now creating a training and testing sample from the ICPSR data (75% training, 25% testing).

set.seed(8675309)
train <- sample(x = 1:nrow(icpsr), size = nrow(icpsr) \* 0.75)

Creating an example of a pairs plot with random variables of interest (substitute others you may feel are appropriate).

pairs(formula = ~ avg\_house\_size + gini\_index + avg\_hours\_worked,
 data = icpsr,
 subset = train)

Now making a correlation plot (correlelogram).

library(corrplot)
c <- cor(x = icpsr[train,])
colnames(c) <- c('Average House Size', 'GINI Index', 'Hispanic', 'Average Hours Worked', 'Drive to Work',
 'Poverty Status', 'Women with Children', 'White', 'Black', 'Receiving SSI', 'Male',
 'Population', 'Median Income', 'Education Level', 'Births Last Year', 'Working Men',
 'Working Women', 'Married', 'Divorced', 'Aggravated Assault')
rownames(c) <- c('Average House Size', 'GINI Index', 'Hispanic', 'Average Hours Worked', 'Drive to Work',
 'Poverty Status', 'Women with Children', 'White', 'Black', 'Receiving SSI', 'Male',
 'Population', 'Median Income', 'Education Level', 'Births Last Year', 'Working Men',
 'Working Women', 'Married', 'Divorced', 'Aggravated Assault')

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

par(mfrow = c(1,1))
corrplot(c,
 method = "pie", # visualization method,
 shade.col = NA, # color of shade line
 tl.col = "black", # color of text label
 tl.srt = 45, # text label rotation
 col = col(200), # color of glyphs
 order = "alphabet",
 diag = TRUE,
 type = 'upper')

Now creating a standard linear model. On inspection of residuals, apparent systematic behavior is present, leading us to determine that machine learning models seem like a valid approach.

fit1 <- lm(formula = agg\_assault\_arrest ~ ., data = icpsr, subset = train)
summary(fit1)
plot(fit1)

library(MASS)
fit2 <- glm.nb(formula = agg\_assault\_arrest ~ ., data = icpsr, subset = train)
summary(fit2)

yhat\_reg <- predict(object = fit1, newdata = icpsr[-train , ])

icpsr\_test <- icpsr[-train, 'agg\_assault\_arrest'] ## we will call this multiple times in other functions below

plot(x = icpsr\_test, y = yhat\_reg)
abline(0,1)

mean((yhat\_reg - icpsr\_test)^2) # mean square error
sqrt(mean((yhat\_reg - icpsr\_test)^2)) # standard deviation

Now moving forward with a regression tree.

library(tree)
train\_tree <- tree(formula = agg\_assault\_arrest ~ . , data = icpsr, subset = train)
summary(train\_tree)
plot(train\_tree)
text(train\_tree, pretty = 0)

complex <- cv.tree(train\_tree) # cross-fold validation to determine optimal level of complexity
complex
plot(complex$size, complex$dev,type = 'b')

## pruning for interpretation
train\_prune <- prune.tree(train\_tree, best = 6) # have your best match the above plot for number
summary(train\_prune)
plot(train\_prune)
text(train\_prune)

## let's look the prediction aspect
yhat\_tree <- predict(object = train\_prune, newdata = icpsr[-train , ])
plot(x = icpsr\_test, y = yhat\_tree)
abline(0,1)

mean((yhat\_tree - icpsr\_test)^2) # mean square error
sqrt(mean((yhat\_tree - icpsr\_test)^2)) # standard deviation

Now a gradient boosted model.

library(gbm)
train\_boost <- gbm(formula = agg\_assault\_arrest ~ . ,
 data = icpsr[train , ], # there is no subset command in this function
 distribution = 'gaussian',
 n.trees = 5000,
 shrinkage = 0.001,
 interaction.depth = 3)
summary(train\_boost)
yhat\_boost <- predict(object = train\_boost,
 newdata = icpsr[-train , ],
 n.trees = 5000,
 interaction.depth = 3)
plot(x = icpsr\_test, y = yhat\_boost)
abline(0,1)

mean((yhat\_boost - icpsr\_test)^2)
sqrt(mean((yhat\_boost - icpsr\_test)^2))

Random forest model with output of figure showing variable importance. The random forest model is what will be applied to the prediction aspect for identifying potential outlier counties.

library(randomForest)
train\_rf <- randomForest(formula = agg\_assault\_arrest ~ . ,
 data = icpsr,
 subset = train,
 mtry = 19,
 n.trees = 100,
 importance = TRUE)
train\_rf
yhat\_rf <- predict(object = train\_rf, newdata = icpsr[-train , ])
plot(x = icpsr\_test, y = yhat\_rf)
abline(0,1)

mean((yhat\_rf - icpsr\_test)^2) # mean square error
sqrt(mean((yhat\_rf - icpsr\_test)^2)) # standard deviation

out <- as.data.frame(importance(train\_rf))
out2 <- cbind(rownames(out), data.frame(out, row.names = NULL))
names(out2) <- c('var', 'mse', 'purity')
out2
variable\_full <- c('Average House Size', 'GINI Index', 'Hispanic', 'Average Hours Worked',
 'Drive to Work', 'Poverty Status', 'Women with Children', 'White', 'Black',
 'Receiving SSI', 'Male', 'Population', 'Median Income', 'Education Level',
 'Births Last Year', 'Working Men', 'Working Women', 'Married', 'Divorced')
out3 <- cbind(out2, variable\_full)
out3

library(ggplot2)
j <- ggplot(data = out3, aes(x = mse, y = reorder(variable\_full, mse)))
j <- j + geom\_point(color = 'blue', size = 3.5)
j <- j + theme\_minimal()
j <- j + labs(x = '% Increase Mean Square Error (MSE)', y = '')
j <- j + theme(axis.text = element\_text(size = 12))
j

varImpPlot(x = train\_rf, main = 'Variable Importance Plot')

Now the random forest model is applied to the SC SAC data to identify potential outlier counties.

## setting the number of standard deviations to evaluate
st\_dev <- 1 \* (sqrt(mean((yhat\_rf - icpsr\_test)^2)))

## 2011
yhat\_rf11 <- predict(object = train\_rf,
 newdata = socar[ , -c(1, 2, 22, 23)][socar$year == 2011 &
 socar$violent\_crime\_smart\_total <= 141, ] )
summary(yhat\_rf11)
summary(socar$violent\_crime\_smart\_total[socar$year == 2011 & socar$violent\_crime\_smart\_total <= 141])

one <- socar[ , c(1,22)][socar$year == 2011 & socar$violent\_crime\_smart\_total <= 141 , ]
two <- yhat\_rf11

results11 <- cbind(one, two)
names(results11) <- c('county', 'reported', 'predicted')

## 2012
 yhat\_rf12 <- predict(object = train\_rf,
 newdata = socar[ , -c(1, 2, 22, 23)][socar$year == 2012 &
 socar$violent\_crime\_smart\_total <= 141, ] )
 summary(yhat\_rf12)
 summary(socar$violent\_crime\_smart\_total[socar$year == 2012 & socar$violent\_crime\_smart\_total <= 141])

 one <- socar[ , c(1,22)][socar$year == 2012 & socar$violent\_crime\_smart\_total <= 141 , ]
 two <- yhat\_rf12

 results12 <- cbind(one, two)
 names(results12) <- c('county', 'reported', 'predicted')

## 2013
 yhat\_rf13 <- predict(object = train\_rf,
 newdata = socar[ , -c(1, 2, 22, 23)][socar$year == 2013 &
 socar$violent\_crime\_smart\_total <= 141, ] )
 summary(yhat\_rf13)
 summary(socar$violent\_crime\_smart\_total[socar$year == 2013 & socar$violent\_crime\_smart\_total <= 141])

 one <- socar[ , c(1,22)][socar$year == 2013 & socar$violent\_crime\_smart\_total <= 141 , ]
 two <- yhat\_rf13

 results13 <- cbind(one, two)
 names(results13) <- c('county', 'reported', 'predicted')

## 2014
 yhat\_rf14 <- predict(object = train\_rf,
 newdata = socar[ , -c(1, 2, 22, 23)][socar$year == 2014 &
 socar$violent\_crime\_smart\_total <= 141, ] )
 summary(yhat\_rf11)
 summary(socar$violent\_crime\_smart\_total[socar$year == 2014 & socar$violent\_crime\_smart\_total <= 141])

 one <- socar[ , c(1,22)][socar$year == 2014 & socar$violent\_crime\_smart\_total <= 141 , ]
 two <- yhat\_rf14

 results14 <- cbind(one, two)
 names(results14) <- c('county', 'reported', 'predicted')

## 2015
 yhat\_rf15 <- predict(object = train\_rf,
 newdata = socar[ , -c(1, 2, 22, 23)][socar$year == 2015 &
 socar$violent\_crime\_smart\_total <= 141, ] )
 summary(yhat\_rf15)
 summary(socar$violent\_crime\_smart\_total[socar$year == 2015 & socar$violent\_crime\_smart\_total <= 141])

 one <- socar[ , c(1,22)][socar$year == 2015 & socar$violent\_crime\_smart\_total <= 141 , ]
 two <- yhat\_rf15

 results15 <- cbind(one, two)
 names(results15) <- c('county', 'reported', 'predicted')

## OVERALL
results11$year <- 2011
results12$year <- 2012
results13$year <- 2013
results14$year <- 2014
results15$year <- 2015

dta <- rbind(results11, results12, results13, results14, results15)

## setting up the flag component
dta$flag <- 0
dta$flag[dta$predicted - st\_dev > dta$reported |
 dta$predicted + st\_dev < dta$reported] <- 1

## clean-up
dta$predicted <- round(dta$predicted,0)

## showing the difference
dta$delta\_abs <- round(abs(dta$predicted - dta$reported),0)

## whether higher or lower than expected
dta$reported\_direction <- NA
dta$reported\_direction[dta$predicted - st\_dev > dta$reported] <- 'lower'
dta$reported\_direction[dta$predicted + st\_dev < dta$reported] <- 'higher'

dta <- dta[order(dta$county, dta$year, dta$flag),]

Finally rolling up the data to identify counties by the number of years as an outlier.

library(sqldf)
dta2 <- sqldf("select county, sum(flag) as 'count', avg(delta\_abs) as 'mean\_diff', reported\_direction
 from dta
 group by county")
dta2$mean\_diff <- round(dta2$mean\_diff, 0)
dta2