Homework 1

Due: 3 pts – Full source code included, answers are separated and included on final section as well, for ease of viewing.

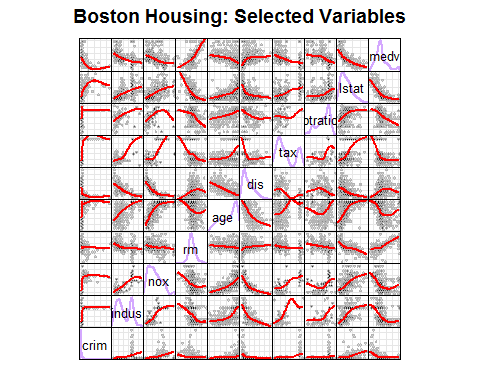
* 3.1 Last plot
* 6.1 Compute the % of explained for the test set and show the script. Compare this to the result in 5.1. Which is higher? (2 pts)
* 6.1 Is the ordering variable in terms of variable importance different from 5.1

## 3.1 Sploms with hexagon bin counts smooths for the more important variables

#Remove four lowest important variables  
varNum(Boston)

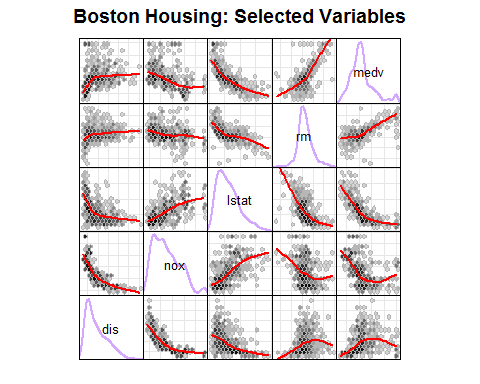
## crim zn indus chas nox rm age dis rad   
## 1 2 3 4 5 6 7 8 9   
## tax ptratio black lstat medv   
## 10 11 12 13 14

Boston2 <- Boston[,-c(2,4,9,12)]  
  
# hexbin modified to 15 xbins and trans power set to 0.5.   
offDiag <- function(x,y,...){  
 panel.grid(h=-1,v=-1,...)  
 panel.hexbinplot(x,y,xbins=15,...,border=gray(.7),  
 trans=function(x)x^.5)  
 panel.loess(x , y, ..., lwd=2,col='red')  
}  
  
windows(width=9,height=9)   
splom(Boston2,  
 xlab='',main="Boston Housing: Selected Variables",  
 pscale=0, varname.cex=0.8,axis.text.cex=0.6,  
 axis.text.col="purple",axis.text.font=2,  
 axis.line.tck=.5,  
 panel=offDiag,  
 diag.panel = onDiag  
)



We might choose to focus on the four most important variables

#Focus on four most important variables  
windows(width=9,height=9)   
splom(Boston[,c(8,5,13,6,14)],  
 xlab='',main="Boston Housing: Selected Variables",  
 pscale=0, varname.cex=0.8,axis.text.cex=0.6,  
 axis.text.col="purple",axis.text.font=2,  
 axis.line.tck=.5,  
 panel=offDiag,  
 diag.panel = onDiag  
)



## 6. Random forest using 6 randomly selected variables at each node

#6.1 A- Compute % explained for test set; compare with 5.1  
##6.1  
set.seed(1)  
rf.boston=randomForest(medv~.,data=Boston,  
 subset=train,mtry=6,importance=TRUE)  
rf.boston

##   
## Call:  
## randomForest(formula = medv ~ ., data = Boston, mtry = 6, importance = TRUE, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 6  
##   
## Mean of squared residuals: 11.8888  
## % Var explained: 85.6

yhat.rf = predict(rf.boston,newdata=Boston[-train,])  
testMse <- mean((yhat.rf-boston.test)^2)  
testMse

## [1] 11.66454

den <- mean((yhat.rf-mean(boston.test))^2)  
pVar <- 100\*(1-testMse/den)  
pVar # percent of variance explained - 85.6

## [1] 80.74407

##5.1  
set.seed(1)  
bag.boston=randomForest(medv~.,data=Boston,subset=train,  
 mtry=13,importance=TRUE)  
bag.boston

##   
## Call:  
## randomForest(formula = medv ~ ., data = Boston, mtry = 13, importance = TRUE, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 13  
##   
## Mean of squared residuals: 11.15723  
## % Var explained: 86.49

yhat.bag = predict(bag.boston,newdata=Boston[-train,])  
mse <- mean((yhat.bag-boston.test)^2)  
mse

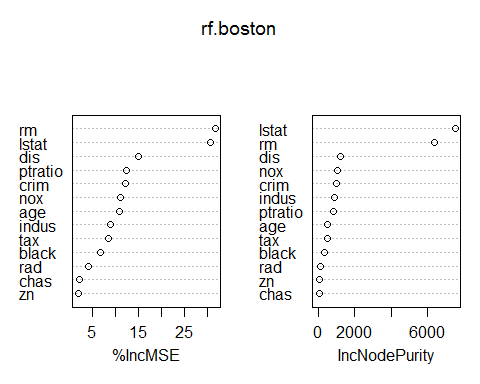
## [1] 13.50808

den <- mean( (boston.test-mean(boston.test))^2)  
pVar <- 100\*(1-mse/den)  
pVar #percent of variance explained - 86.5

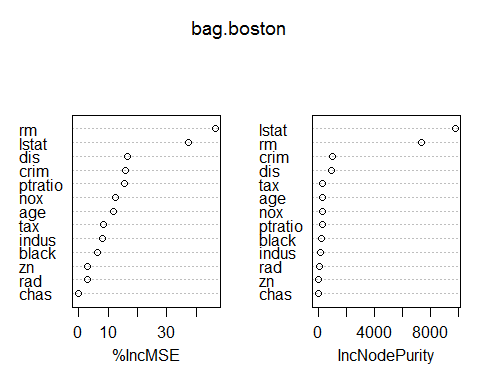
## [1] 84.33158

##5.1 is higher, at 86.5, compared to 85.6

#6.1 B - Ordering variable in terms of importance  
##6.1  
varImpPlot(rf.boston)



#Compare with 5.1  
varImpPlot(bag.boston)



##The top three variables (rm, lstat, dis) do not change for MSE within the two examples, however, only the top two are consistent in terms of the Node Purity (lstat, rm)