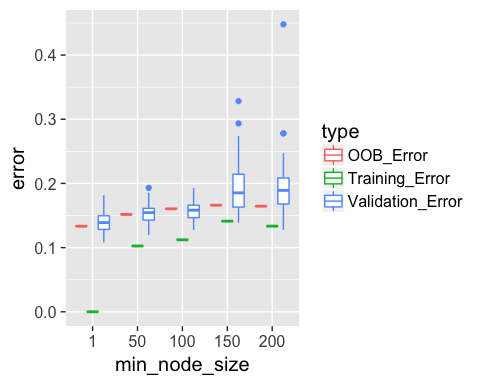
R-Lab: Random Forests OOB Error

We apply random forests to the re-admission data, with different number of minimum node sizes. The out-of-bag (OOB) error rates are obtained. Separately, we use half of the data for training random forests and use the other half to get the testing error (referred to as the validation error). This is repeated 50 times and we can get a distribution of the validation error. Lastly, we also get the training error by building random forests on the data set and use the same data set for testing. Three types of error rates are plotted in the boxplot below. Note there is only one data point at each node size for OOB error and training error, which are represented as one line instead a range. It can be seen while the OOB error rates are reasonably aligned with the validation error rates, the training error rates are much smaller.

library(dplyr)  
library(tidyr)  
library(ggplot2)  
require(randomForest)  
set.seed(1)  
  
theme\_set(theme\_gray(base\_size = 15))  
library(RCurl)  
data <- read.csv(text = getURL("https://raw.githubusercontent.com/shuailab/ind\_498/master/resource/data/AD.csv"))  
  
target\_indx <- which(colnames(data) == "DX\_bl")  
data[, target\_indx] <- as.factor(paste0("c", data[, target\_indx]))  
rm\_indx <- which(colnames(data) %in% c("ID", "TOTAL13", "MMSCORE"))  
data <- data[, -rm\_indx]  
  
para.v <- c(1, 50, 100, 150, 200)  
results <- NULL  
  
# OOB error  
for (ipara in para.v) {  
 rf <- randomForest(DX\_bl ~ ., nodesize = ipara, data = data) # nodesize = inodesize  
 results <- rbind(results, c("OOB\_Error", ipara, mean(rf$err.rate[, "OOB"])))  
}  
  
# Validation error  
for (ipara in para.v) {  
 for (i in 1:50) {  
 train.ix <- sample(nrow(data), floor(nrow(data)/2))  
 rf <- randomForest(DX\_bl ~ ., nodesize = ipara, data = data[train.ix,   
 ])  
 pred.test <- predict(rf, data[-train.ix, ], type = "class")  
 this.err <- length(which(pred.test != data[-train.ix, ]$DX\_bl))/length(pred.test)  
 results <- rbind(results, c("Validation\_Error", ipara, this.err))  
 }  
}  
  
# Training error  
for (ipara in para.v) {  
 rf <- randomForest(DX\_bl ~ ., nodesize = ipara, data = data) # nodesize = inodesize  
 pred <- predict(rf, data, type = "class")  
 this.err <- length(which(pred != data$DX\_bl))/length(pred)  
 results <- rbind(results, c("Training\_Error", ipara, this.err))  
}  
  
colnames(results) <- c("type", "min\_node\_size", "error")  
results <- as.data.frame(results)  
results$error = as.numeric(as.character(results$error))  
results$min\_node\_size <- factor(results$min\_node\_size, unique(results$min\_node\_size))  
ggplot() + geom\_boxplot(data = results, aes(y = error, x = min\_node\_size, color = type)) +   
 geom\_point(size = 3)



Let's also investigate the impact of the number of trees (ntree) on OOB errors. In particular, we compare 50 trees with 500 trees, with the OOB errors plotted in the plot below. As expected, the OOB errors from 50 trees are clearly larger than the errors from 500 trees. This is because fewer trees are used in the random forests with 50 trees.

para.v <- c(1, 50, 100, 150, 200)  
results <- NULL  
  
# OOB error with 500 trees  
for (ipara in para.v) {  
 rf <- randomForest(DX\_bl ~ ., nodesize = ipara, ntree = 500, data = data) # nodesize = inodesize  
 results <- rbind(results, c("OOB\_Error\_500trees", ipara, mean(rf$err.rate[,   
 "OOB"])))  
}  
  
# OOB error with 50 trees  
for (ipara in para.v) {  
 rf <- randomForest(DX\_bl ~ ., nodesize = ipara, ntree = 50, data = data) # nodesize = inodesize  
 results <- rbind(results, c("OOB\_Error\_50trees", ipara, mean(rf$err.rate[,   
 "OOB"])))  
}  
colnames(results) <- c("type", "min\_node\_size", "error")  
results <- as.data.frame(results)  
results$error = as.numeric(as.character(results$error))  
results$min\_node\_size <- factor(results$min\_node\_size, unique(results$min\_node\_size))  
ggplot(data = results, aes(y = error, x = min\_node\_size, fill = type)) + geom\_bar(stat = "identity",   
 position = "dodge")

