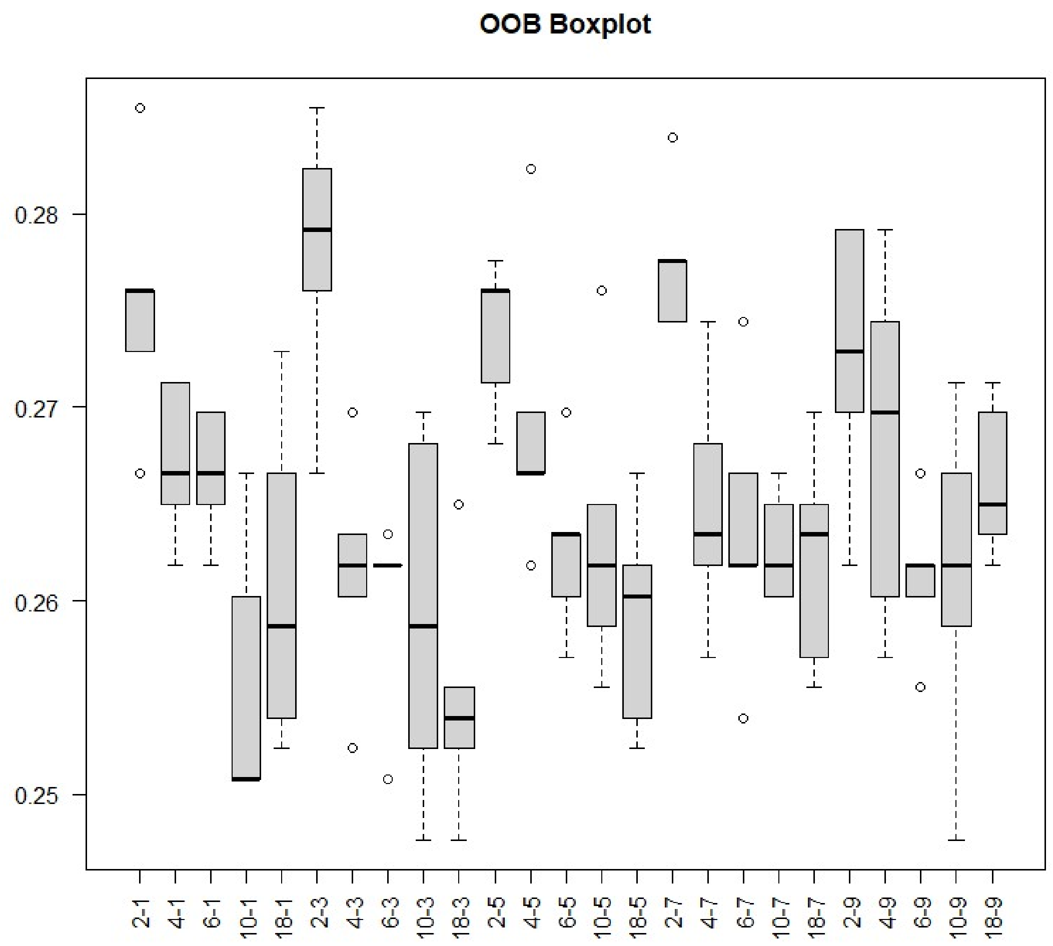
3. Tune the number of variables and the node sizes for the random forest using its internal out-of-bag error. Since *p* = 18 and the default is mtry= p18 _ 4, use a grid of 2,4,6,10,18. For node size, default is 1, so try 1,3,5,7,10. Rerun 5 random forests of 500 trees each.

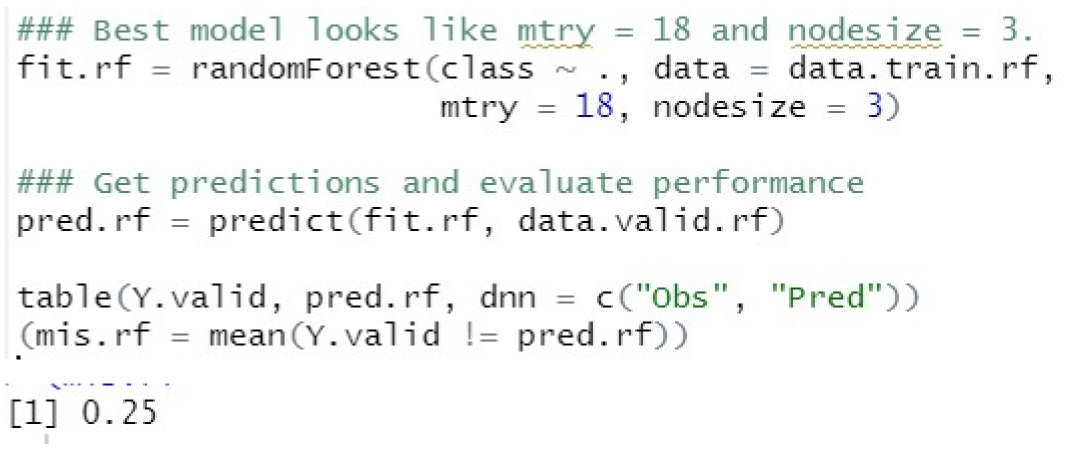
(a) **Report the average misclassification rate for each combination**

```
### Set tuning parameters
all.mtrys = c(2,4,6,10,18)
all.nodesizes = c(1, 3, 5, 7, 9)
all.pars.rf = expand.grid(mtry = all.mtrys, nodesize = all.nodesizes)
n.pars = nrow(all.pars.rf)

M = 5 # Number of times to repeat RF fitting. I.e. Number of OOB errors

### Container to store OOB errors. This will be easier to read if we name
### the columns.
all.OOB.rf = array(0, dim = c(M, n.pars))
names.pars = apply(all.pars.rf, 1, paste0, collapse = "-")
colnames(all.OOB.rf) = names.pars


for(i in 1:n.pars){
  ### Progress update
  print(paste0(i, " of ", n.pars))

  ### Get tuning parameters for this iteration
  this.mtry = all.pars.rf[i, "mtry"]
  this.nodesize = all.pars.rf[i, "nodesize"]

  for(j in 1:M){
    ### Fit RF, then get and store OOB errors
    this.fit.rf = randomForest(class ~ ., data = data.train.rf,
                               mtry = this.mtry, nodesize = this.nodesize)

    pred.this.rf = predict(this.fit.rf)
    this.err.rf = mean(Y.train.rf != pred.this.rf)

    all.OOB.rf[j, i] = this.err.rf
  }
}

### Make a regular and relative boxplot
boxplot(all.OOB.rf, las=2, main = "OOB Boxplot")
```



(c) **Which combination do you think is best?**
-> 18-3 combination looks best

(d) **Report test error for that combination. Compare it to the default RF: did it help? If so, how does it compare to other methods?**



```
### Best model looks like mtry = 18 and nodesize = 3.
fit.rf = randomForest(class ~ ., data = data.train.rf,
                      mtry = 18, nodesize = 3)

### Get predictions and evaluate performance
pred.rf = predict(fit.rf, data.valid.rf)

table(Y.valid, pred.rf, dnn = c("Obs", "Pred"))
(mis.rf = mean(Y.valid != pred.rf))
```

```
[1] 0.25
```

It shows normal performance, but I don’t know why, it doesn’t show better result than default RF.