**##Part A**

> d <- Hitters[is.na(Hitters$Salary) == F,]

> d$logSal <- log(d$Salary)

> mean(d$logSal)

[1] 5.927222

**##Part B**

> i <- 1:200

> d.train <- d[i,]; d.test <- d[-i,]

> list(train.mean.log.Salary = mean(d.train$logSal), test.mean.log.Salary = mean(d.test$logSal))

$train.mean.log.Salary

[1] 5.940142

$test.mean.log.Salary

[1] 5.886203

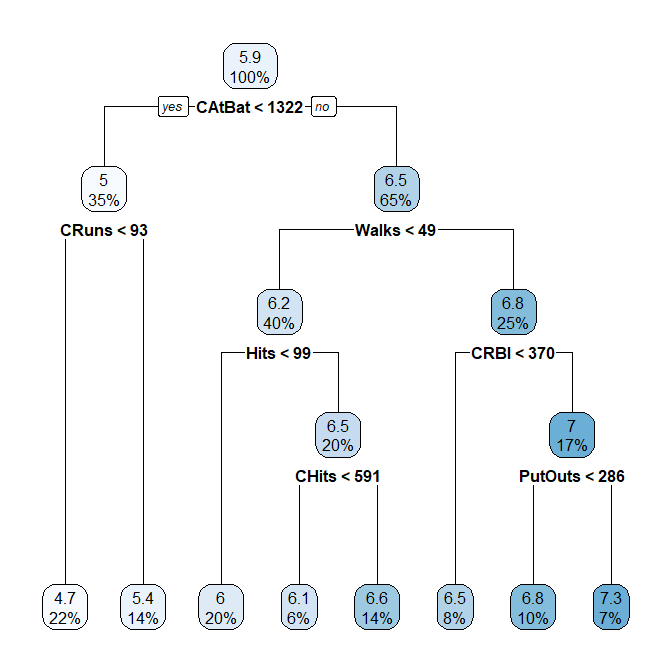
**##Part C [Note: switching to rpart package for better plots]**

> library(rpart)

> tree.train <- rpart(logSal ~ .-Salary, data = d.train)

> library(rpart.plot)

> rpart.plot(tree.train)



#"Residual Mean Deviance" on training set; RMSE on test

> TRD <- sum((predict(tree.train) - d.train$logSal)^2) #total residual dev

> num.term.nodes <- 8

> n.train <- nrow(d.train)

> RMD <- TRD/(n.train - num.term.nodes)

> RMSE.test <- sqrt(mean((predict(tree.train, newdata = d.test) - d.test$logSal)^2))

> list(RMD.on.train = RMD, RMSE.on.test = RMSE.test)  
**$RMD.on.train**

**[1] 0.1855452**

**$RMSE.on.test**

**[1] 0.5700177**

**Part D [Switch to {tree} package for CV.tree function]**

#library("tree")

> tree.train2 <-tree(logSal ~ .-Salary, data =d.train)

> pruned1 = cv.tree(tree.train2,FUN = prune.tree)

> cbind(pruned1$size,pruned1$dev)

# [,1] [,2]

# [1,] 10 63.05655

# [2,] 9 64.03083

# [3,] 8 62.39457

# [4,] 7 62.08164

# [5,] 6 63.24531

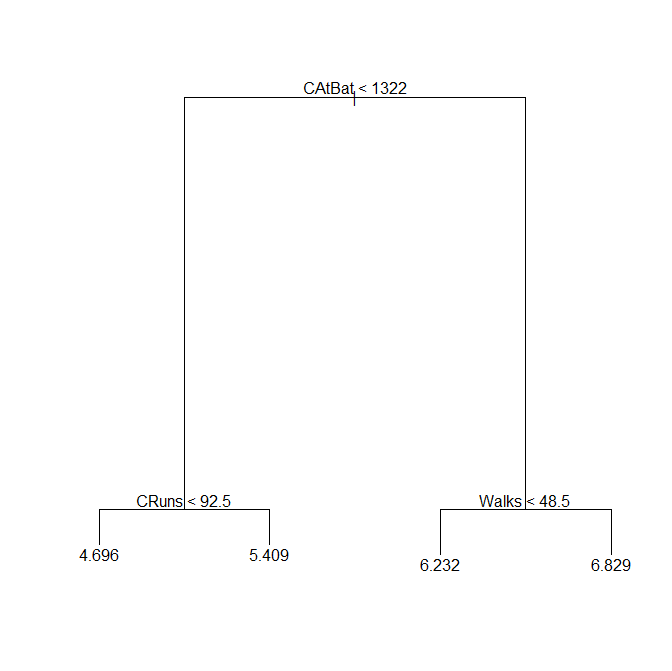
# [6,] 5 60.17142

# [7,] 4 58.64906

# [8,] 3 63.90342

# [9,] 2 69.81360

# [10,] 1 170.81099

> tree.best4 <- prune.tree(tree.train2, best = 4); plot(tree.best4); text(tree.best4, pretty = 0)

> RMSE.best4 <- sqrt(mean(resid(tree.best4, newdata = d.test)^2))

> RMSE.best4

[1] 0.479202

**PART E)**

> Bag.tree <- randomForest(logSal ~ .-Salary, data = d.train, ntree = 1000, mtry = 10)

> RMSE.btree <- sqrt(mean((predict(bag.tree, newdata = d.test)-d.test$logSal)^2))

**> RMSE.btree**

**[1] 0.4697587**

**PART F)**

> RMSE.on.test <- function(tree) {

RMSE.test <- sqrt(mean((predict(tree, newdata = d.test)-d.test$logSal)^2))

return(RMSE.test)

}

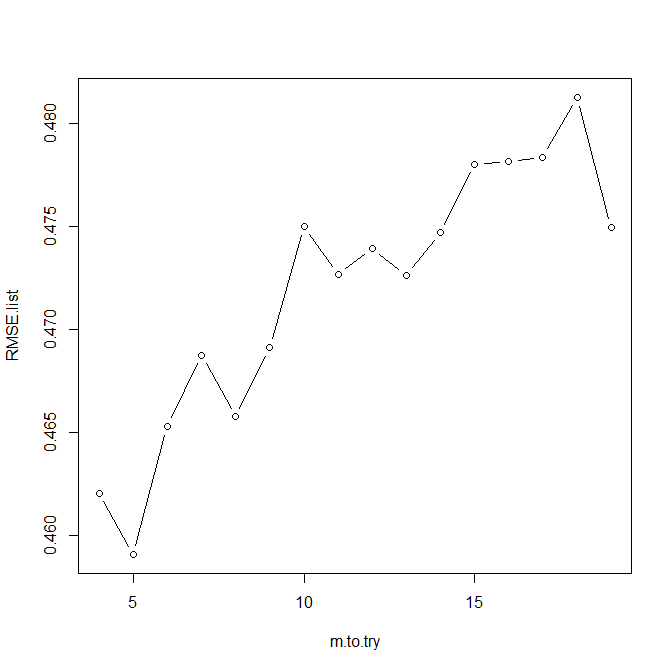
> RMSE.list <- c()

> m.to.try <- 4:19

for (i in 1:length(m.to.try)) {

tree <- randomForest(logSal ~ .-Salary, data = d.train, ntree = 1000, mtry = m.to.try[i])

RMSE.list[i] <- RMSE.on.test(tree)

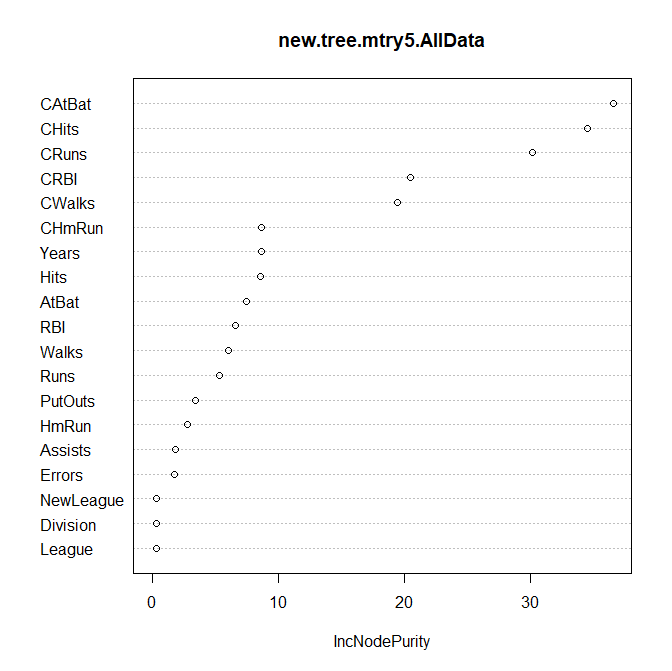
}

> plot(m.to.try,RMSE.list)

**PART G)**

> new.tree.mtry5.AllData <- randomForest(logSal ~ .-Salary, data = d, ntree = 5000, mtry = 5)

> varImpPlot(new.tree.mtry5.AllData)



**Part H)** The methods show some similarities, suggesting some convergent validity, however there are some key differences. For example, the random forest method did identify “Walks” as an important predictor of Salary, however, the other methods using the Cp criteria found “Walks” to be within the top 3 (a parent of two terminal nodes). The other top variables among the methods were largely the same (“AtBat” and “Hits”). As far as prediction error, all methods seemed to agree that the best prediction RMSE on the test set was around .46. However, in this regard, the random forest was far superior, for which, at certain levels of “mtry”, the RMSE was **below** .46.