```r
# bagging.r
RNGkind(sample.kind = 'Rounding')
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
library(MASS)      # Boston dataset
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
dim(Boston)
```

```
## [1] 506  14
```

```r
# response is medv
# p=13 predictors
#
n = nrow(Boston)
set.seed(1)
train = sample(n,n/2)   # 253 train rows

# BAGGING   -all 13 predictors should be considered at each split (mtry=p)
set.seed(1)
bag1=randomForest(medv~.,data=Boston,subset=train,mtry=13,importance = T)
#
# train performance
bag1
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = Boston, mtry = 13, importance = T,      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
```

```r
# train MSE shown as Mean of squared residuals
# p = 13 predictors
# default is B=500 trees
# will use
# importance(bag1) to ask for the importance of predictors

summary(bag1)
```

```
##                 Length Class  Mode
## call              6    -none- call
## type              1    -none- character
## predicted       253    -none- numeric
## mse             500    -none- numeric
## rsq             500    -none- numeric
## oob.times       253    -none- numeric
## importance       26    -none- numeric
```

```
## importanceSD     13    -none- numeric
## localImportance   0    -none- NULL
## proximity         0    -none- NULL
## ntree             1    -none- numeric
## mtry              1    -none- numeric
## forest           11    -none- list
## coefs             0    -none- NULL
## y               253    -none- numeric
## test              0    -none- NULL
## inbag             0    -none- NULL
## terms             3    terms  call
```

```r
#
# 500 train MSEs

# times train obs was OOB
table(bag1$oob.times)
```

```
##
## 152 154 157 160 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178
##   2   1   1   2   1   2   5   3   4   2   3   4   1   6   4   5   9   3  12   6
## 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##  12   9   9   4  20   5   8   8  12   6   6   2  13  12   9   6   7   4   4   3
## 199 200 201 203 204 205 206 207 208 215
##   1   2   3   3   1   2   3   1   1   1
```

```r
#
# compare predictions vs actual prices
#
head(bag1$predicted)
```

```
##      135      188      289      457      102      451
## 16.24355 28.00107 22.70232 16.58221 25.98038 14.62453
```

```r
# actual values
head(bag1$y)
```

```
##  135  188  289  457  102  451
## 15.6 32.0 22.3 12.7 26.5 13.4
```

```r
head(Boston[train,"medv"])
```

```
## [1] 15.6 32.0 22.3 12.7 26.5 13.4
```

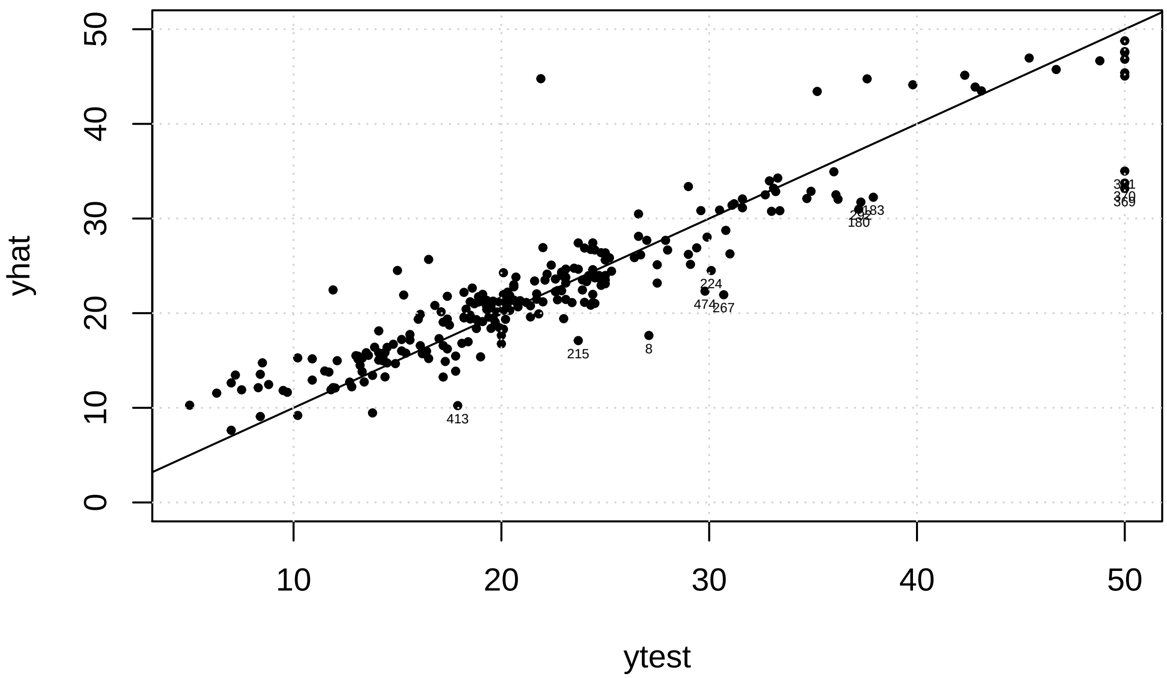
```r
#

# test set performance
ytest=Boston[-train,"medv"]         # y values in test set
yhat = predict(bag1,newdata=Boston[-train,])

# residuals and  row numbers
res = ytest - yhat
a = rownames(as.matrix(yhat))    # as.matrix is required

# plot yhat vs y
plot(yhat~ytest,pch=19,cex=0.5,ylim=c(0,50))
abline(0,1)
grid()
```

```
text(yhat~ytest,labels=ifelse(res>5,a,""),pos=1,offset=0.25,cex=0.4)
```




```
# dots seem to cluster around 45 degree line

# MSPE
mean((yhat-ytest)^2)     # this value changes
```

```
## [1] 13.50808
```

```
# large improvement over single tree MSPE

# try B=25 bagged trees
bag2=randomForest(medv~.,data=Boston,subset=train,mtry=13,ntree=25)
yhat = predict(bag2,newdata=Boston[-train,])
mean((yhat-ytest)^2)     #  this value changes
```

```
## [1] 13.94835
```

```
#
# not much different than that with B=500 trees
#
#
# RANDOM FOREST (mtry < p)
#
set.seed(1)
forest1=randomForest(medv~.,data=Boston,subset=train,mtry=6,importance=T)
#
forest1
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = Boston, mtry = 6, importance = T,      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
```

```r
#
# MSPE
yhat.rf = predict(forest1,newdata=Boston[-train,])
mean((yhat.rf-ytest)^2)
```

```
## [1] 11.66454
```

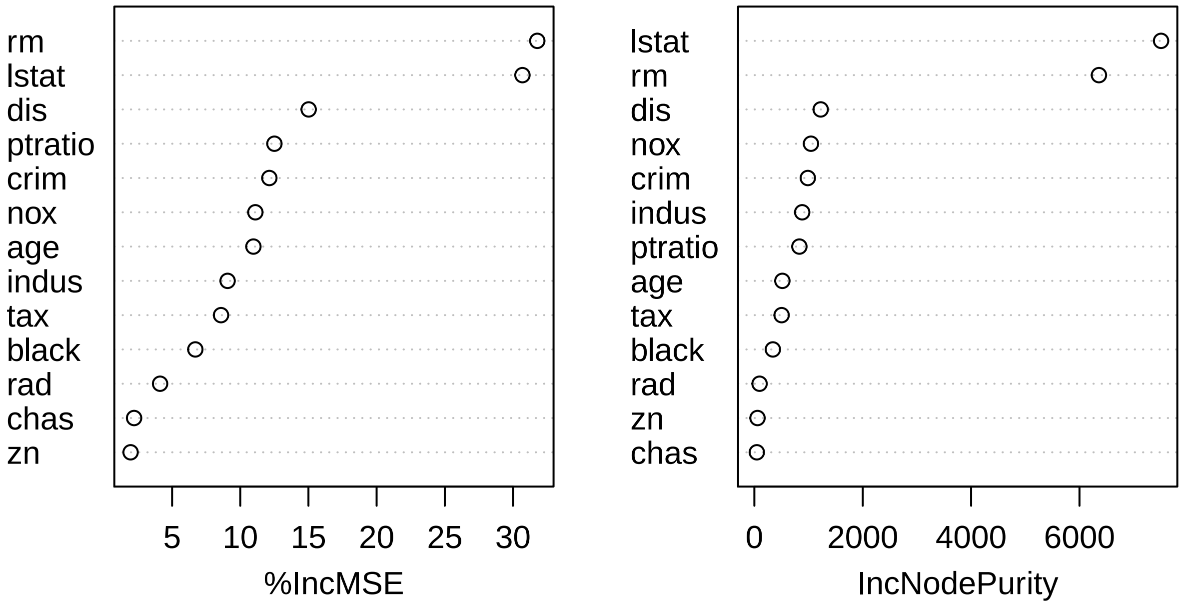
```r
# some improvement over bagging

# importance of each predictor
#
importance(forest1)
```

```
##           %IncMSE IncNodePurity
## crim    12.132320     986.50338
## zn       1.955579      57.96945
## indus    9.069302     882.78261
## chas     2.210835      45.22941
## nox     11.104823    1044.33776
## rm      31.784033    6359.31971
## age     10.962684     516.82969
## dis     15.015236    1224.11605
## rad      4.118011      95.94586
## tax      8.587932     502.96719
## ptratio 12.503896     830.77523
## black    6.702609     341.30361
## lstat   30.695224    7505.73936
```

```r
#
# IncMSE - avg increase in MSE when predictor is excluded from model
# IncNodePurity - avg increase in RSS from splits using this predictor

# plot these two columns - for convenience
varImpPlot(forest1,main="")
```




```
#
# rm and lstat most important predictors
```