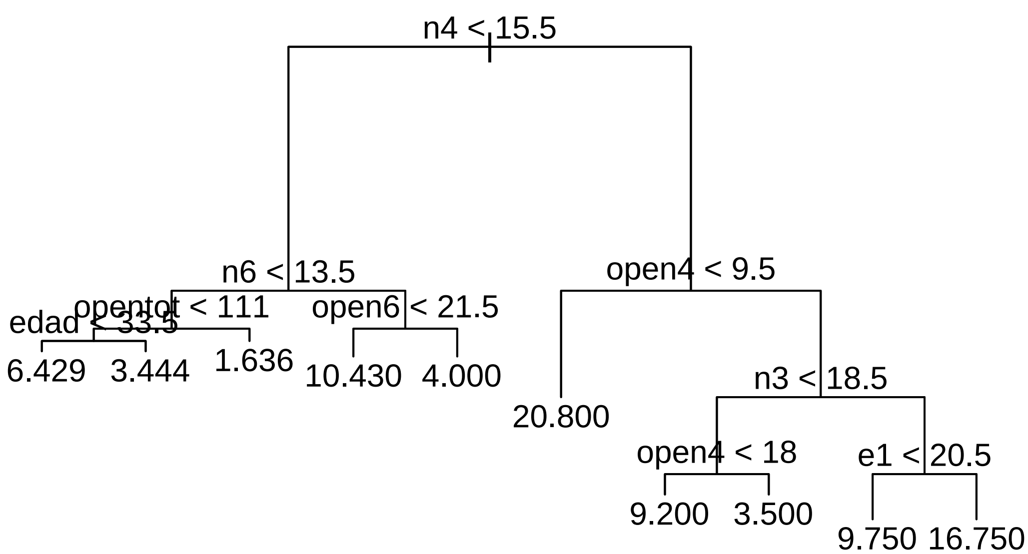
Predicting depression based on big five personality scales

Load dataset and select training data:

```
library(foreign)
cardata <- read.spss("data Carillo et al.sav", to.data.frame = TRUE)
set.seed(42)
train <- sample(1:112, 80)
```

Regression tree for predicting depression

```
library(tree)
car.tree <- tree(bdi ~ ., data = cardata[train,])
plot(car.tree)
text(car.tree, pretty = 0)
```




```
car.pred <- predict(car.tree, cardata[-train,])
mean((cardata$bdi[-train] - car.pred)^2)
```

```
## [1] 87.56231
```

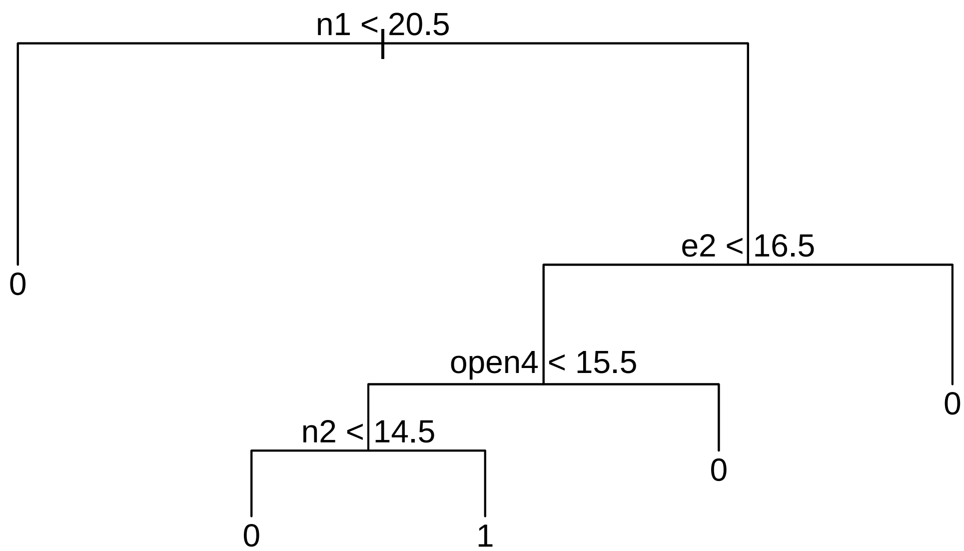
```
cor(cardata$bdi[-train], car.pred)
```

```
## [1] 0.4378805
```

The test MSE for the regression tree is 87.56.

Classification tree for predicting depression

```
cardata2 <- cardata
cardata2$bdi <- factor(ifelse(cardata2$bdi > 16, 1, 0))
car.tree2 <- tree(bdi ~ ., data = cardata2[train,])
plot(car.tree2)
text(car.tree2, pretty = 0)
```




```
car.pred2 <- predict(car.tree2, cardata2[-train,])
prop.table(table(cardata2$bdi[-train], round(car.pred2[,2])))
```

```
##
##           0       1
##   0 0.75000 0.03125
##   1 0.09375 0.12500
```

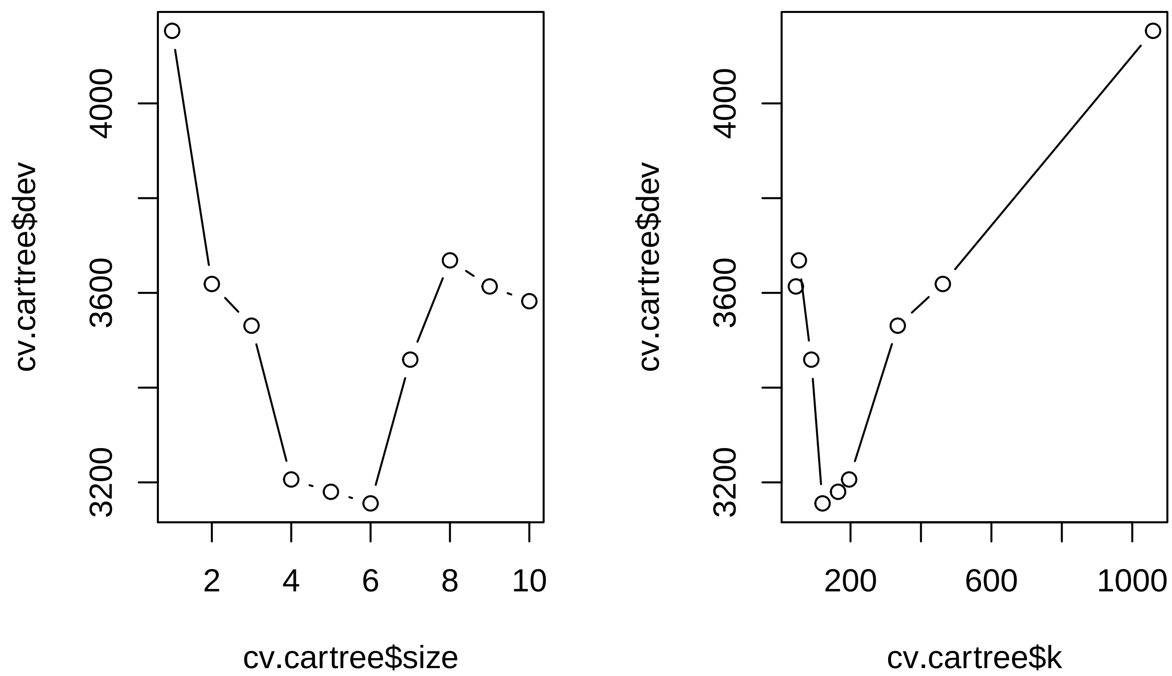
The correct classification rate in test data for the classification tree is $.75 + .125 = .875$

Pruning

We are going to prune the regression tree. First, we determine the optimal tree size by 10-fold cross validation:

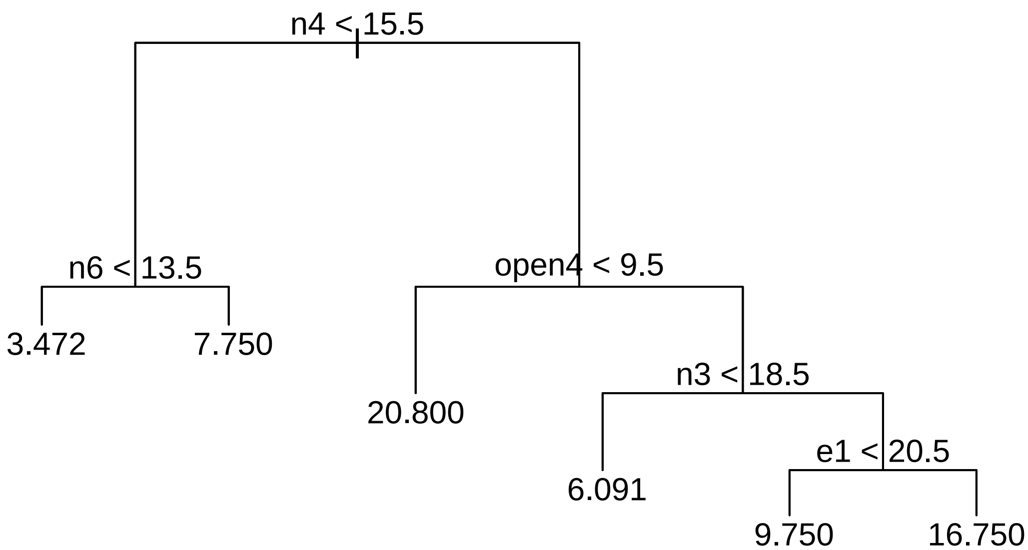
```
set.seed(3)
cv.cartree <- cv.tree(car.tree)
par(mfrow = c(1,2))
plot(cv.cartree$size, cv.cartree$dev, type = "b")
```

```r
plot(cv.cartree$k, cv.cartree$dev, type= "b")
```



Optimal tree size is 6. Now we prune the tree:

```r
cartree.pruned <- prune.tree(car.tree, best = 6)
plot(cartree.pruned)
text(cartree.pruned, pretty = 0)
```




```
car.pruned.pred <- predict(cartree.pruned, cardata[-train,])
mean((cardata$bdi[-train] - car.pruned.pred)^2)
```

```
## [1] 88.03171
```

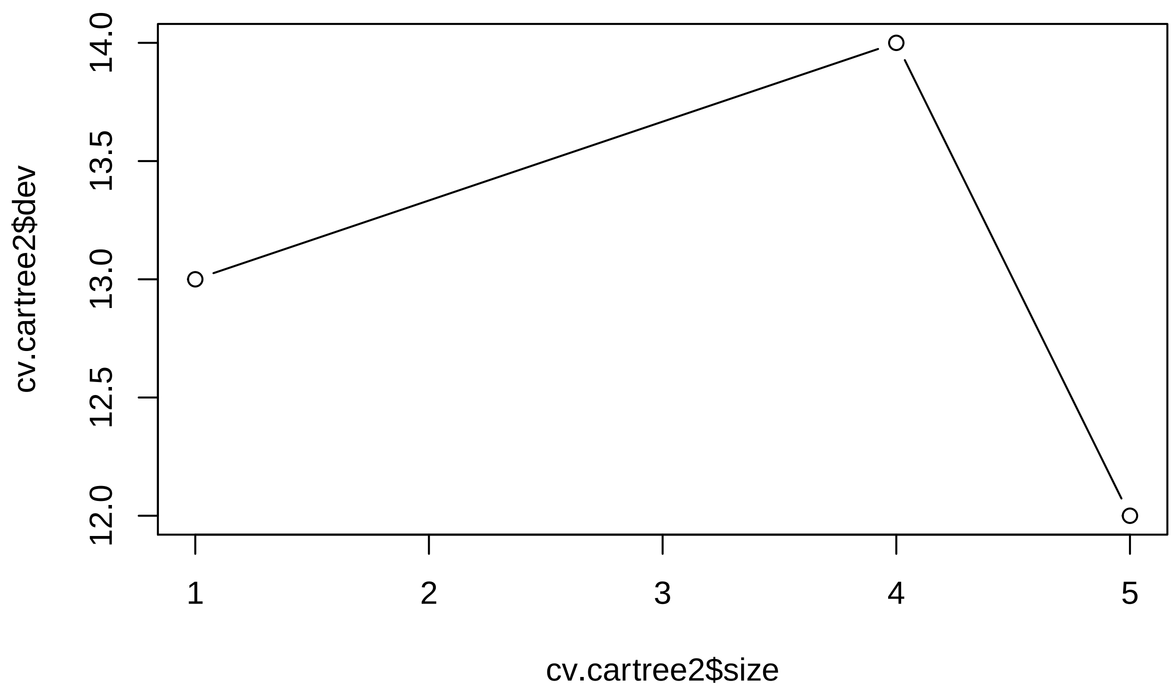
```
cor(cardata$bdi[-train], car.pruned.pred)
```

```
## [1] 0.4253286
```

We got a smaller tree, but in this case, pruning did not improve predictive accuracy on test data. The MSE for the pruned tree is 88.03.

We also determine the optimal size for the classification tree:

```
cv.cartree2 <- cv.tree(car.tree2, FUN = prune.misclass)
plot(cv.cartree2$size, cv.cartree2$dev, type = "b")
```

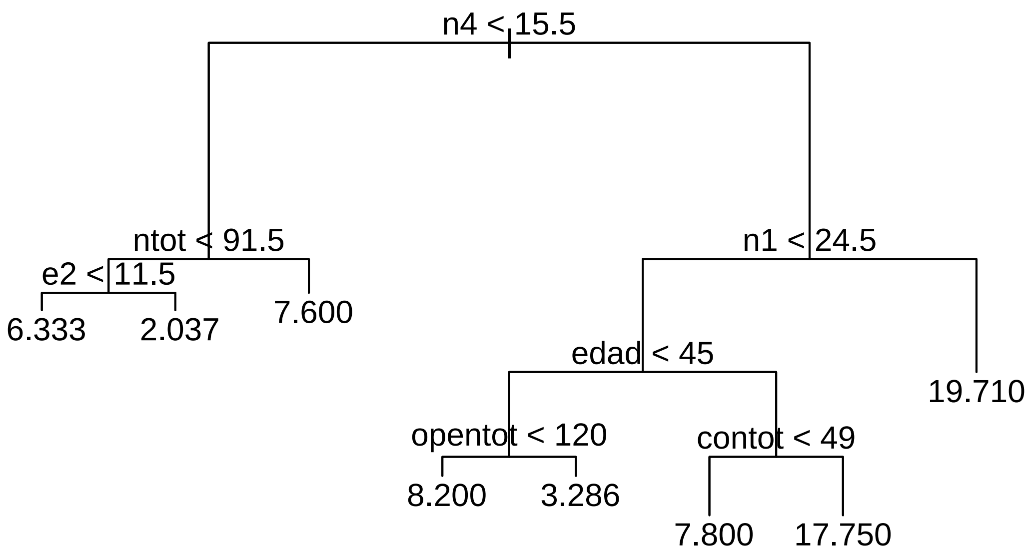


The classification tree does not need pruning.

Tree instability

We grow a regression tree on a different sample of the same size, from the same data:

```
set.seed(46383)
train2 <- sample(1:112, 80)
car.tree3 <- tree(bdi ~ ., data = cardata[train2,])
plot(car.tree3)
text(car.tree3, pretty = 0)
```




```
car.pred3 <- predict(car.tree3, cardata[-train2,])
mean((cardata$bdi[-train2] - car.pred3)^2)
```

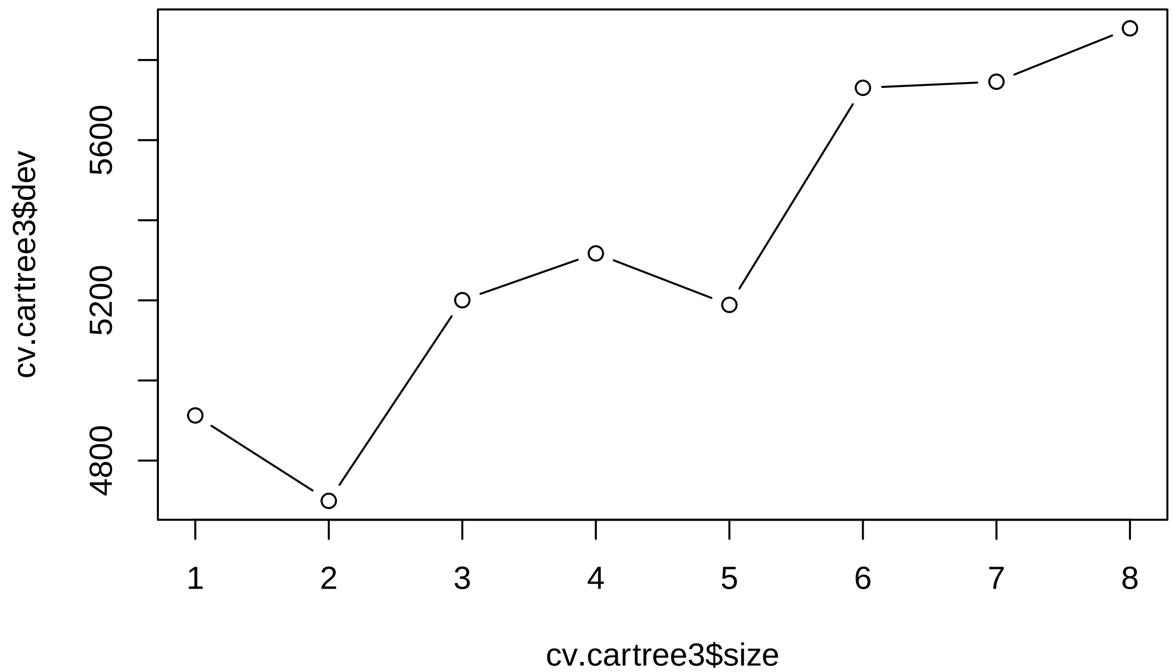
```
## [1] 69.03054
```

```
cor(cardata$bdi[-train2], car.pred3)
```

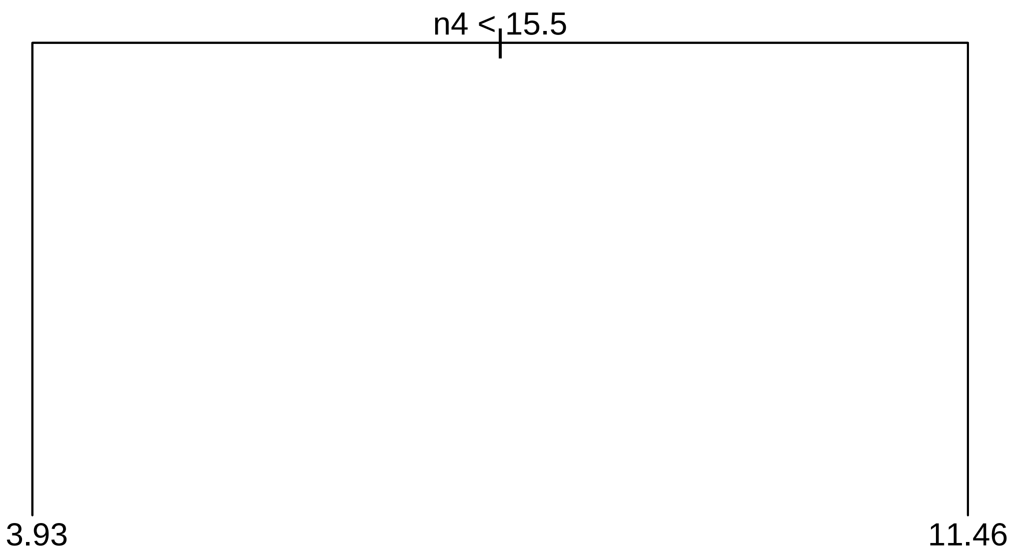
```
## [1] 0.5435759
```

In this sample, we find a different tree structure: only the first split in the tree is the same. Also, predictive accuracy in test data is different from that in the earlier sample, also indicating instability. Let's prune the tree to see if that yields a more stable tree:

```
set.seed(343545)
cv.cartree3 <- cv.tree(car.tree3)
plot(cv.cartree3$size, cv.cartree3$dev, type = "b")
```




```
cartree3.pruned <- prune.tree(car.tree3, best = 2)
plot(cartree3.pruned)
text(cartree3.pruned, pretty = 0)
```




```
car.pruned.pred3 <- predict(cartree3.pruned, cardata[-train2,])
mean((cardata$bdi[-train2] - car.pruned.pred3)^2)
```

```
## [1] 70.37825
```

```
cor(cardata$bdi[-train2], car.pruned.pred3)
```

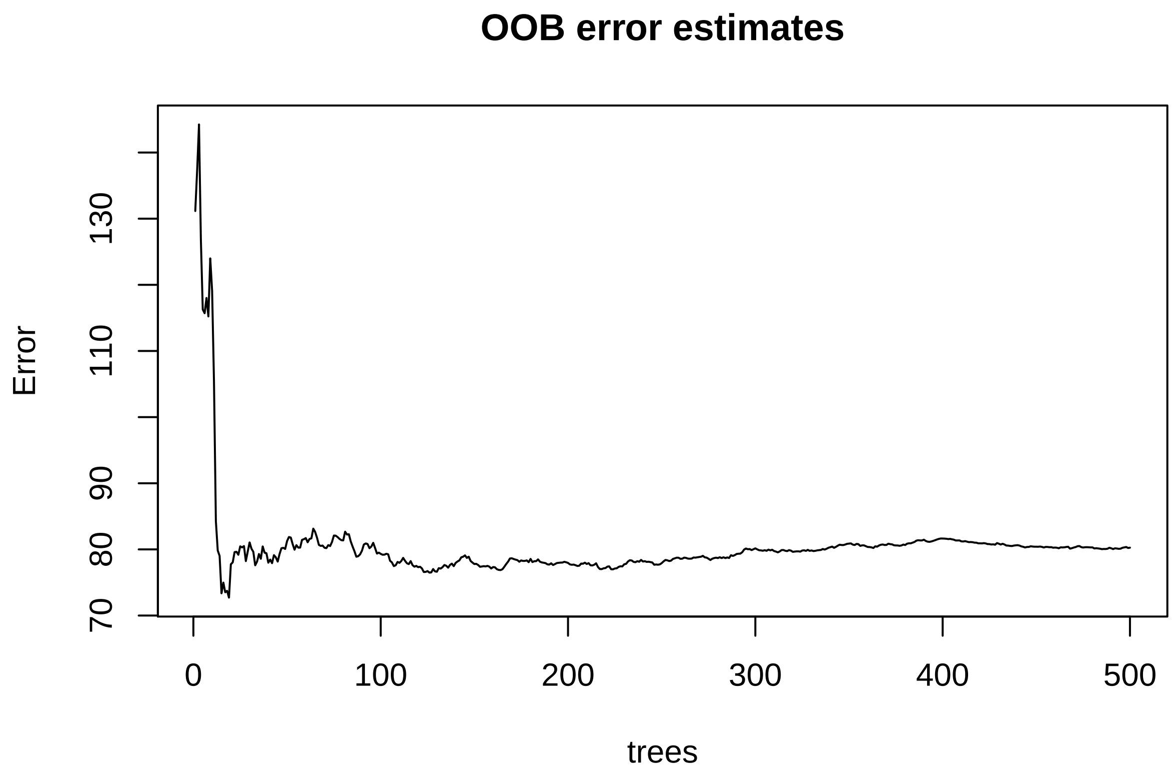
```
## [1] 0.4812077
```

Pruning does not completely eliminate the instability problem, although we do get a smaller tree and more similar predictive accuracy as with the full tree.

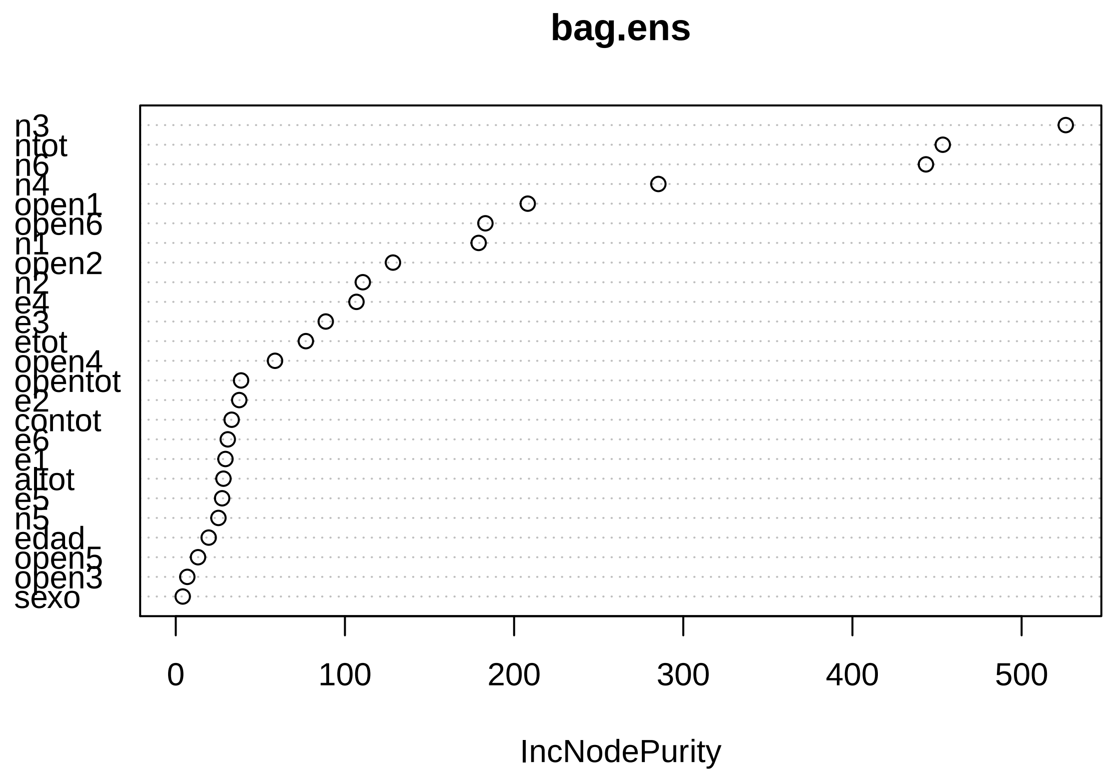
Tree ensembles

Bagging

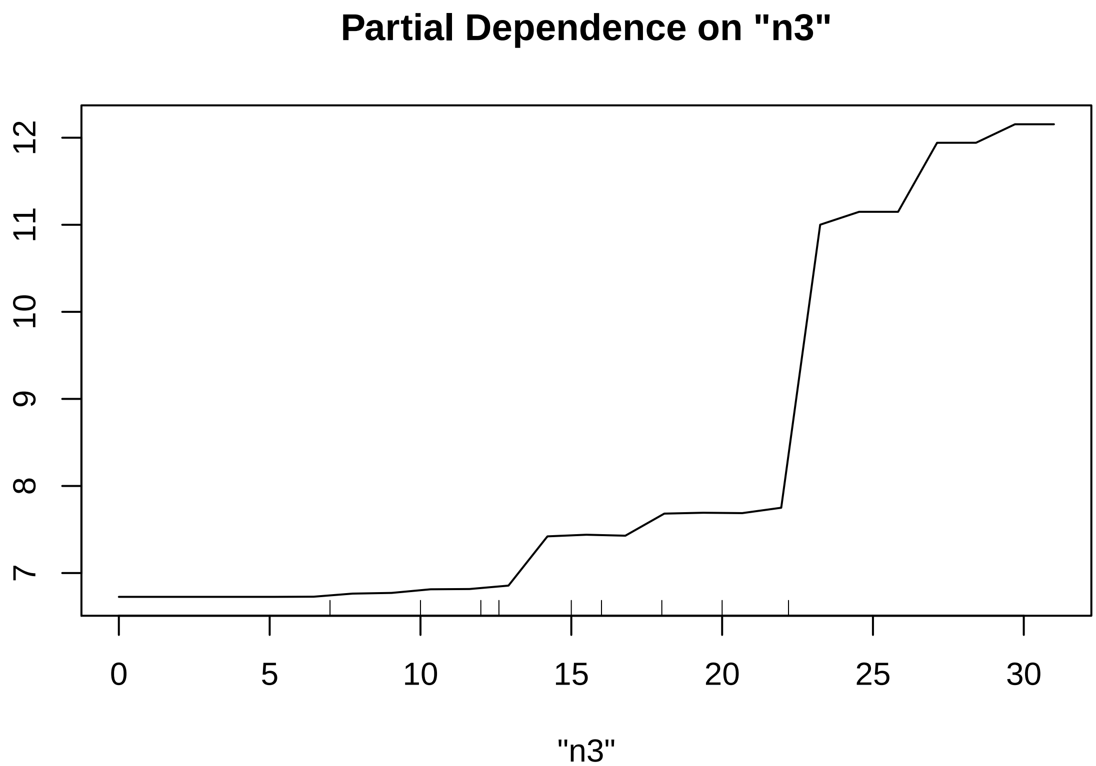
```
library(randomForest)
set.seed(45823)
bag.ens <- randomForest(bdi ~ ., mtry = ncol(cardata) - 1,
                        data = cardata[-train,])
plot(bag.ens, main = "OOB error estimates")
```



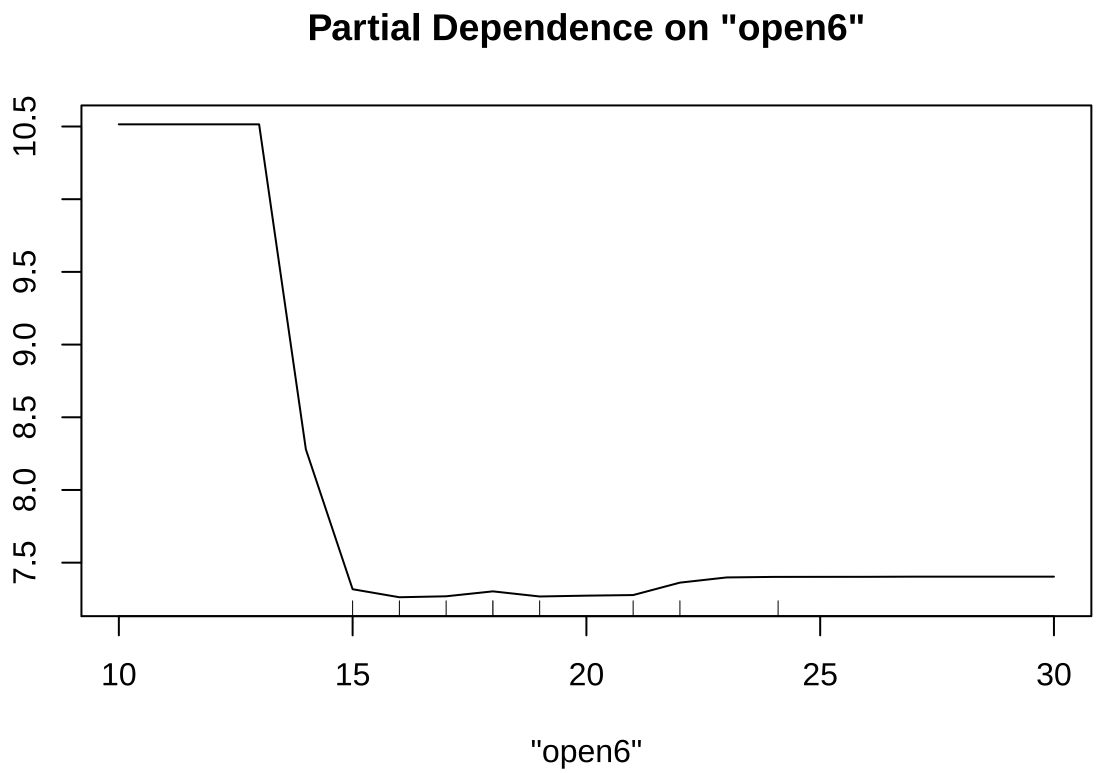

```
varImpPlot(bag.ens)
```




```r
partialPlot(bag.ens, pred.data = cardata[train,], x.var = "n3")
```




```
partialPlot(bag.ens, pred.data = cardata[train,], x.var = "open6")
```



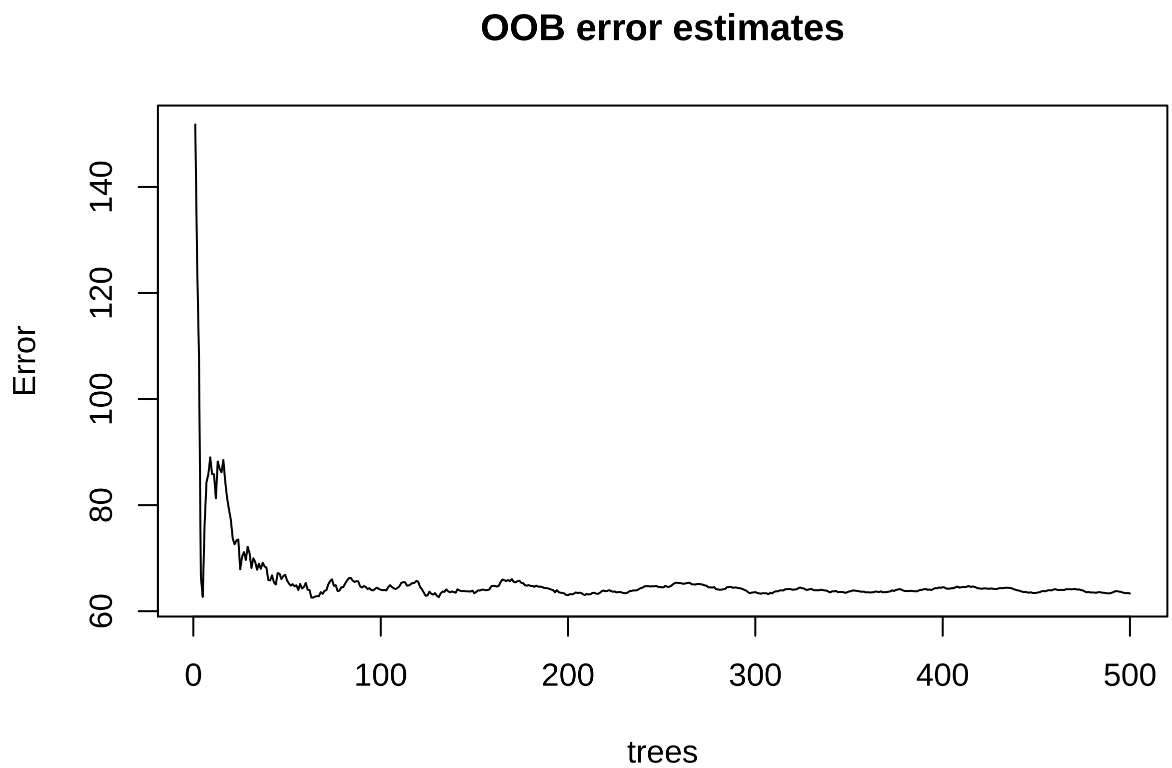

```
yhat.bag <- predict(bag.ens, newdata = cardata[-train,])
mean((yhat.bag - cardata$bdi[-train])^2)
```

```
## [1] 12.87469
```

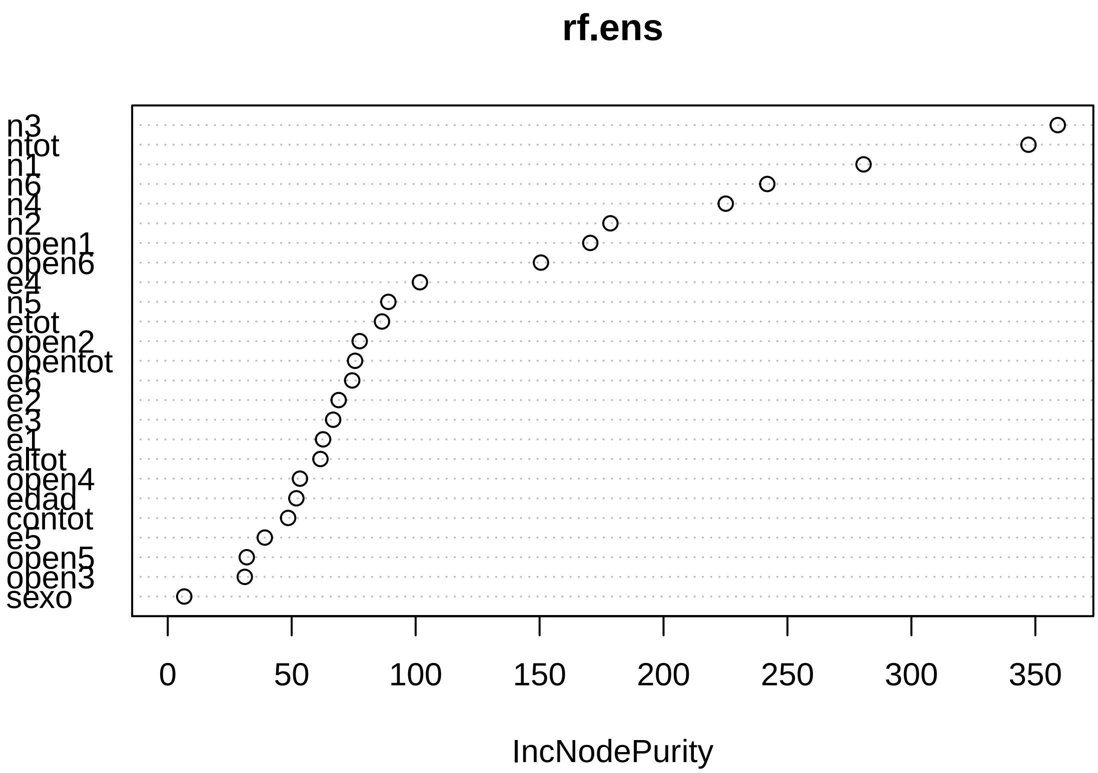
The test MSE for the bagged ensemble is 12.87, which is substantially lower than that of the single tree.

Random forest

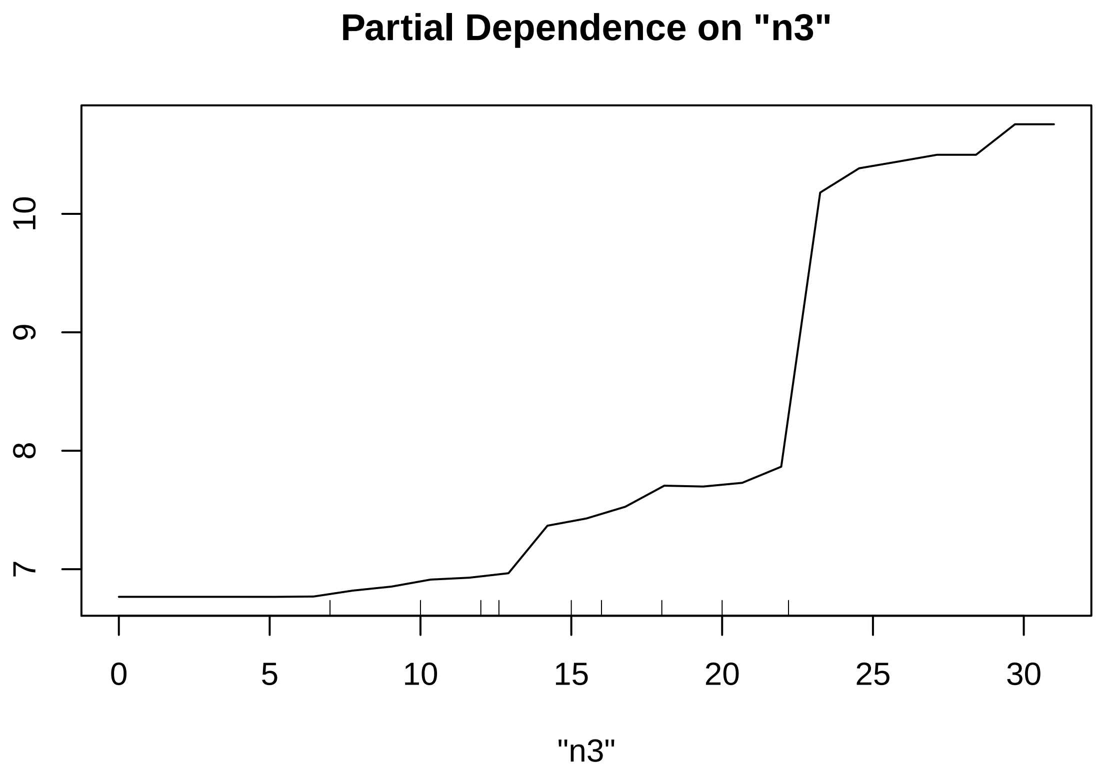
```
set.seed(45823)
rf.ens <- randomForest(bdi ~ ., mtry = sqrt(ncol(cardata) - 1),
                       data = cardata[-train,])
plot(rf.ens, main = "OOB error estimates")
```



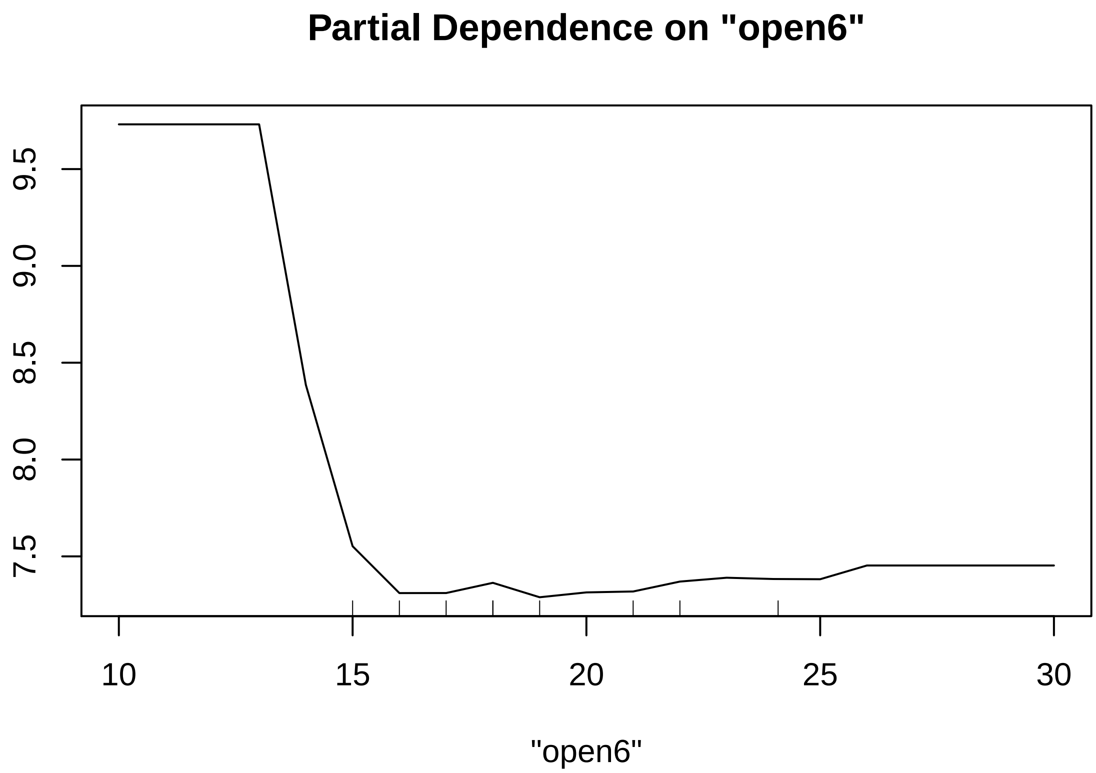

```r
varImpPlot(rf.ens)
```




```r
partialPlot(rf.ens, pred.data = cardata[train,], x.var = "n3")
```




```
partialPlot(rf.ens, pred.data = cardata[train,], x.var = "open6")
```



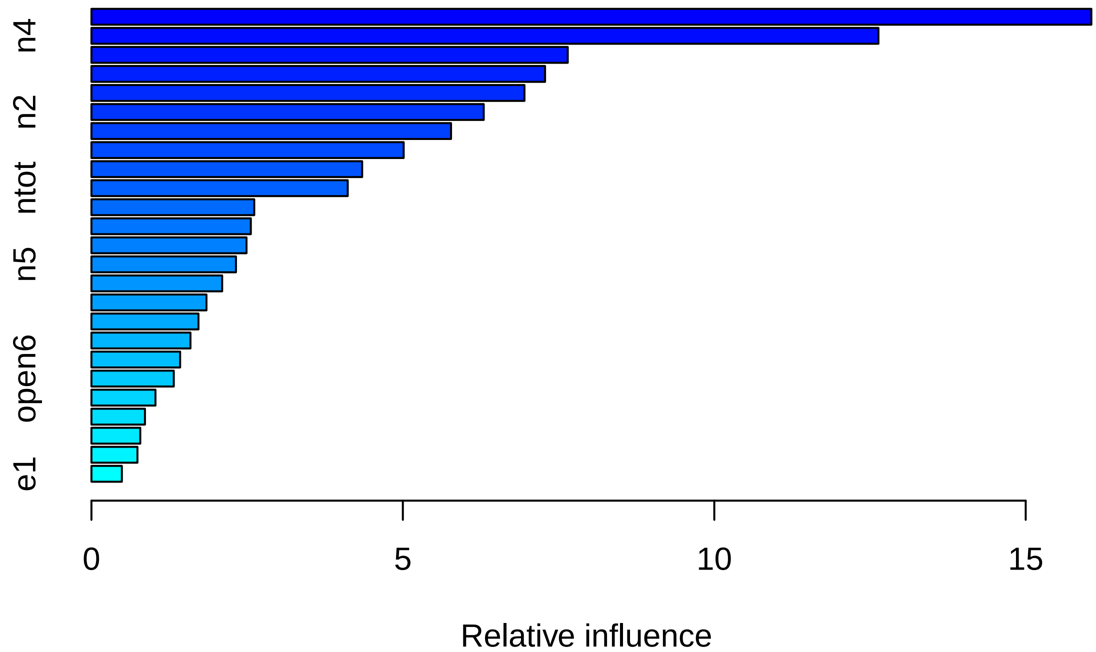

```r
yhat.rf <- predict(rf.ens, newdata = cardata[-train,])
mean((yhat.rf - cardata$bdi[-train])^2)
```

```
## [1] 12.50123
```

The test MSE for the random forest is 12.50, slightly lower than that of the boosted ensemble.

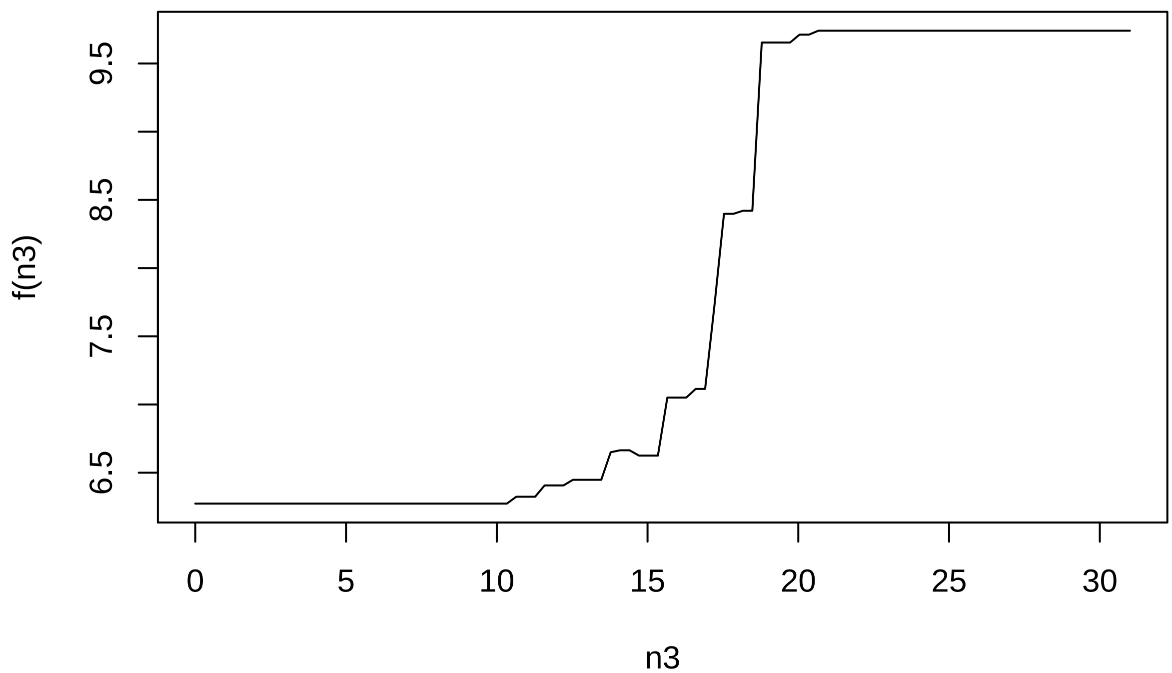
Boosting

```r
library(gbm)
set.seed(47895321)
boost.ens <- gbm(bdi ~ ., data = cardata[train,], distribution = "gaussian",
                 n.trees = 500, shrinkage = .01, interaction.depth = 4)
summary(boost.ens)
```

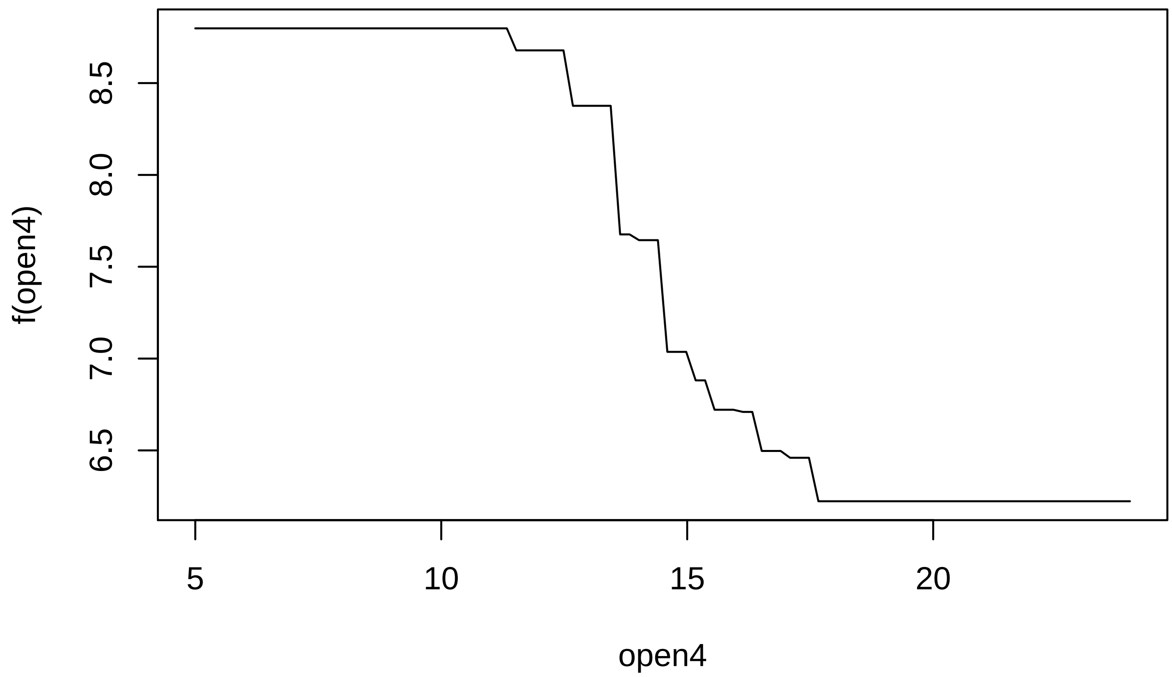



```
##                 var    rel.inf
## n3               n3 16.0541454
## n4               n4 12.6355765
## open4         open4  7.6476624
## n6               n6  7.2831647
## open3         open3  6.9522351
## n2               n2  6.2984536
## e2               e2  5.7735845
## n1               n1  5.0132382
## e6               e6  4.3465808
## ntot           ntot  4.1149076
## opentot opentot  2.6137953
## etot           etot  2.5589454
## edad           edad  2.4897261
## n5               n5  2.3205583
## open5         open5  2.0987289
## open1         open1  1.8464558
## altot         altot  1.7182326
## e5               e5  1.5898183
## contot       contot  1.4239046
## open6         open6  1.3221283
## open2         open2  1.0278423
## sexo           sexo  0.8591599
## e4               e4  0.7841236
## e3               e3  0.7386340
## e1               e1  0.4883978
```

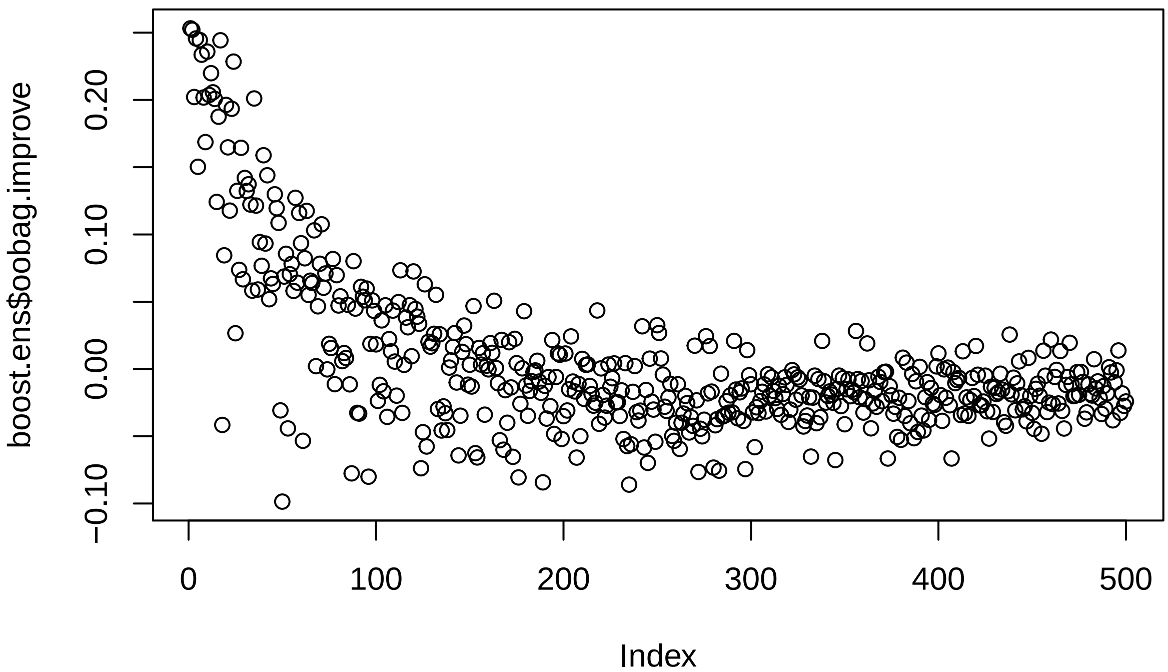
```r
plot(boost.ens, i = "n3")
```




```r
plot(boost.ens, i = "open4")
```




```r
plot(boost.ens$oobag.improve)
```




```
yhat.boost <- predict(boost.ens, newdata = cardata[-train,], n.trees = 500)
mean((yhat.boost - cardata$bdi[-train])^2)
```

```
## [1] 65.23001
```

The test MSE for the boosted ensemble is 65.23; lower than that of the single regression tree, but higher than that of the random forest and bagged ensemble.

Note that I selected the parameter values above, after performing some cross validation using the `train()` function from package `caret`:

```
library(caret)
set.seed(4349493)
tunegrid <- expand.grid(n.trees = c(500, 1000),
                        shrinkage = c(.001, .01, .1),
                        interaction.depth = 3:4,
                        n.minobsinnode = 10)
bdi_ind <- which(names(cardata) == "bdi")
cvpars <- train(x = cardata[train, -bdi_ind], y = cardata[train, bdi_ind],
                method = 'gbm', distribution = "gaussian", tuneGrid = tunegrid)
cvpars
```