Statistics 452: Statistical Learning and Prediction

Chapter 8, Part 2: Bagging and Random Forests

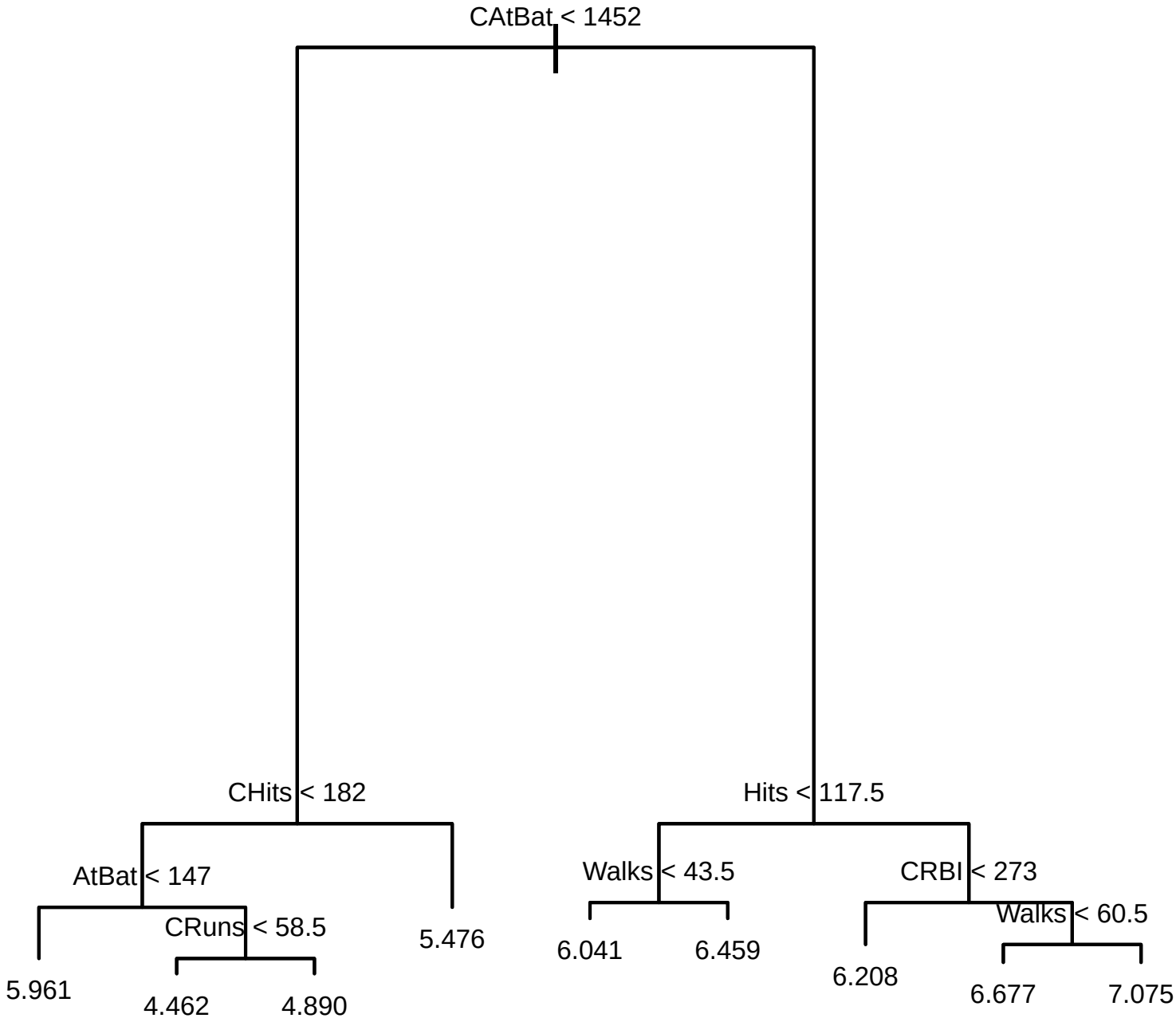
Brad McNeney

Example: The Hitters Data Set

- Example tree grown, but not pruned.

```
library(ISLR)
data(Hitters)
library(dplyr)
Hitters <- mutate(Hitters,lSalary = log(Salary)) %>%
  select(-Salary) %>%
  na.omit()
library(tree)
t1 <- tree(lSalary ~ .,data=Hitters)
```

```
plot(t1)
text(t1,cex=.5)
```



Bootstrapping

- A bootstrap sample from n observations is obtained by drawing n observations, with replacement, from the sample

```
set.seed(543)
n <- 6
x <- round(rnorm(n),1)
sort(x)
```

```
## [1] -1.0 -0.2  0.2  0.4  0.8  1.4
```

```
sort(sample(x,size=n,replace=TRUE)) # bootstrap sample 1,
```

```
## [1] -1.0 -0.2 -0.2 -0.2  0.2  0.4
```

```
sort(sample(x,size=n,replace=TRUE)) # bootstrap sample 2
```

```
## [1] -1.0 -0.2 -0.2  0.2  0.4  0.8
```

```
sort(sample(x,size=n,replace=TRUE)) # bootstrap sample 3
```

```
## [1] -1.0 -1.0 -0.2 -0.2  0.4  0.8
```

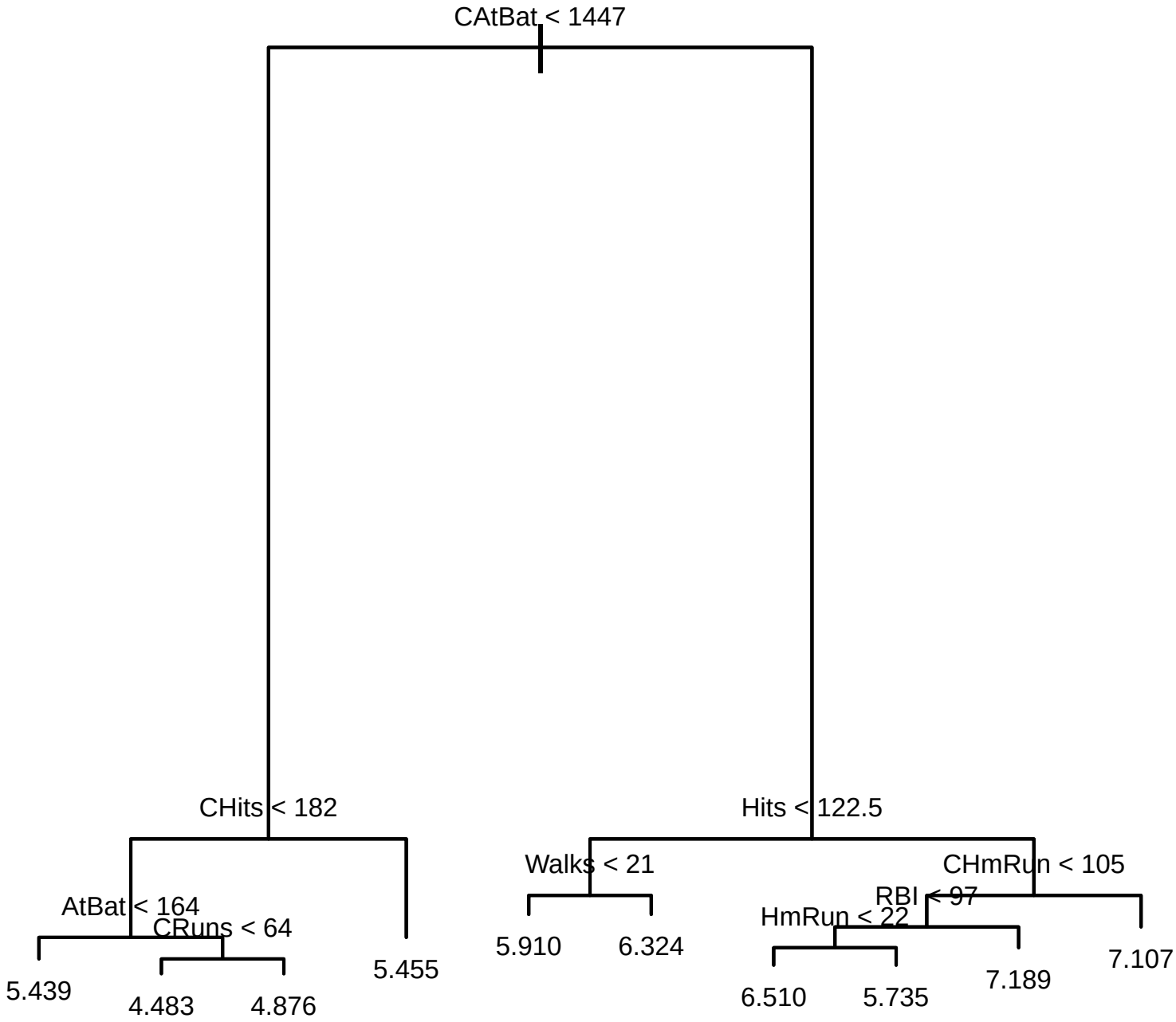
Empirical Distribution

- The data-generating process in our example is a normal distribution.
- The empirical distribution of our sample is the distribution under which each sampled value is equally likely.
- Resampling amounts to sampling from the empirical distribution.
- If empirical close to data-generating distribution, resampled samples are like new samples from the data-generating distribution.

Example: Tree from a Bootstrap Sample

```
# Use sample_n() from dplyr
n <- nrow(Hitters)
bHitters <- sample_n(Hitters,size=n,replace=TRUE)
t1b <- tree(lSalary ~ .,data=bHitters)
```

```
plot(t1b)
text(t1b,cex=.5)
```



Bootstrap Aggregation (Bagging)

- For a general statistical learning method that provides an estimate, $\hat{f}(x)$ of the model $f(x)$, we can aggregate over estimates fit to bootstrap samples.
- If $\hat{f}^b(x)$ is the fitted model from the bth bootstrap sample, the aggregate estimator is

$$\hat{f}_{\text{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x),$$

  where B is the number of bootstrap samples.
- Bagging has been found to be useful for improving predictions from decision trees.
  - Grow each tree "deep" and don't prune.
  - Average predictions over trees.

Bias-Variance Tradeoff

- The deeply-grown trees have high variance, but low bias.
- Averaging many deep trees is an attempt to reduce the variance.
- However, if trees are highly correlated (e.g., similar first few splits) then averaging does not reduce variance.
  - Return to this point when we discuss random forests.

Bagging for Classification

- When the outcome Y is categorical, one approach to the aggregate prediction of category is the "majority vote":
  - Record the predicted class from each tree (each bootstrap sample) and take the most common prediction as the aggregate prediction.

Estimating Test Error with Out-of-Bag (OOB) Observations

- On average about 2/3 are and 1/3 of observations are not resampled
  - See small bootstrap samples of size 6 above.
- Those not resampled are called out-of-bag (OOB)
- Regression trees: The OOB prediction for the ith observation is the average of its predicted values from trees grown on boostrap samples that did not include i.
  - The OOB MSE is the average squared error between observed values and their OOB predictions.
- Classification trees: The OOB prediction for the ith observation is the majority vote from trees grown on bootstrap samples that did not include i.
  - The OOB classification error is the proportion of misclassified OOB predictions.
- OOB errors are valid because the predictions use only trees not fit to the corresponding observation.

Variable Importance

- There isn't one tree to interpret now.
- Need an overall measure of the importance of a variable.
- For a single tree, a variable's importance can be measured by the amount that the node homogeneity (RSS for regression, Gini for classification) is decreased by splitting on the variable.
  - Homogeneity of tree including variable minus homogeneity if variable was not available to split on.

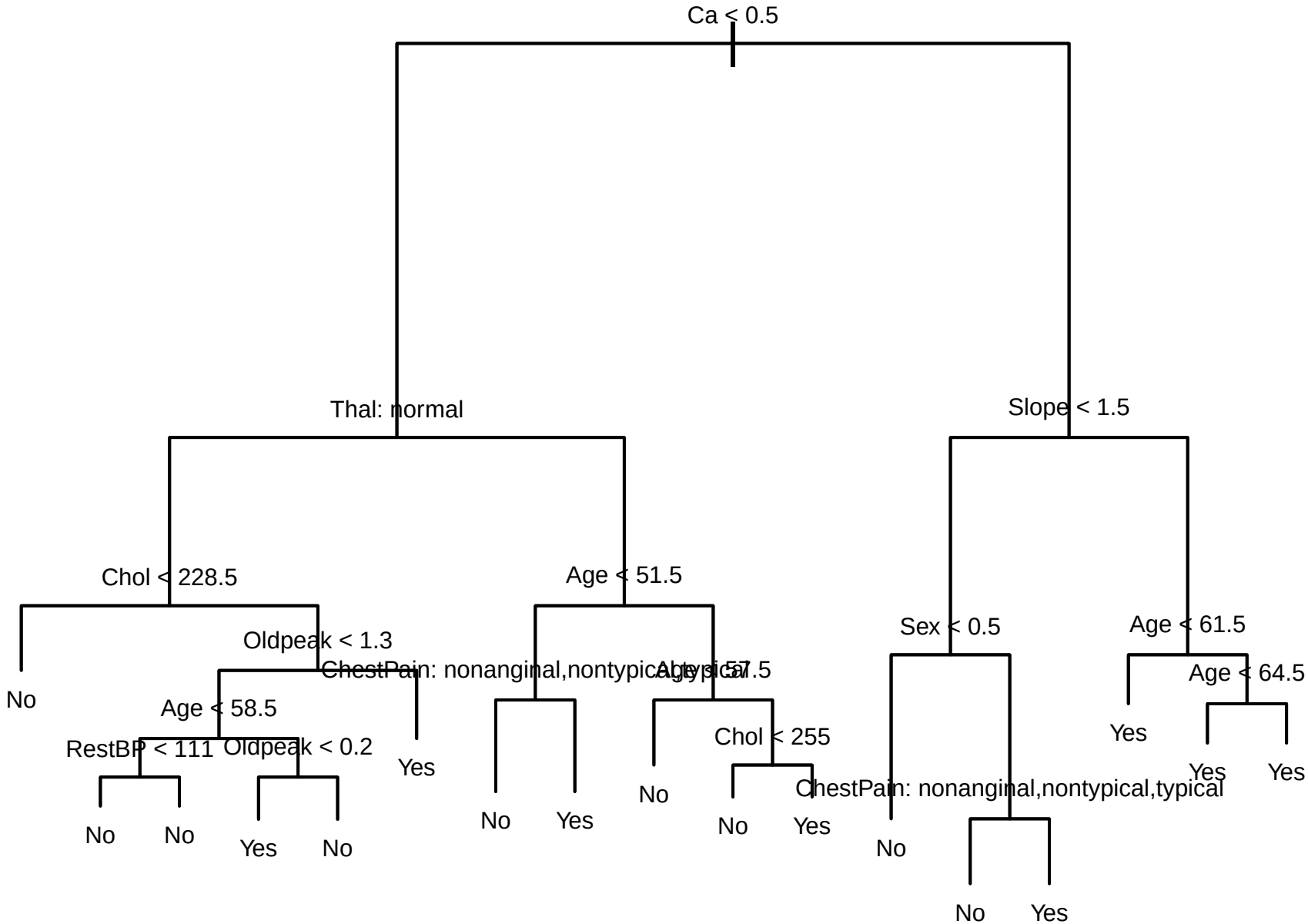
Example: Heart Data

```
uu <- url("http://faculty.marshall.usc.edu/gareth-james/ISL/Heart.csv")
Heart <- read.csv(uu,row.names=1)
Heart <- na.omit(Heart)
dim(Heart) # Train on 2/3, test on 1/3
```

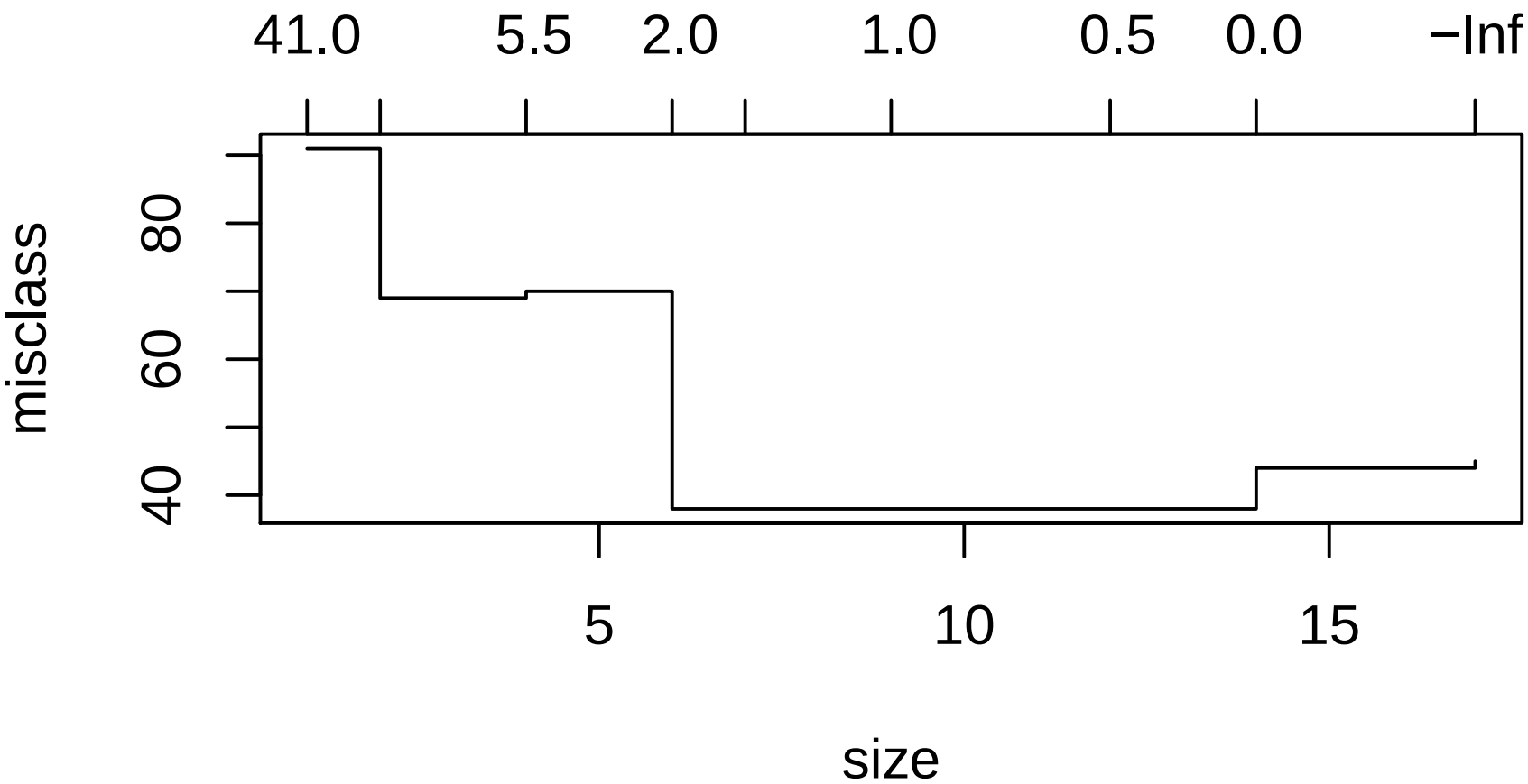
```
## [1] 297  14
```

```
set.seed(1)
train <- sample(1:nrow(Heart),size=2*nrow(Heart)/3,replace=FALSE)
t1 <- tree(AHD ~ ., data=Heart,subset=train)
```

```
plot(t1)
text(t1,cex=.5,pretty=0)
```



```
cv.t1 <- cv.tree(t1,FUN=prune.misclass) # use class'n err for CV
plot(cv.t1) # Suggest size 7
```



```r
t1.best <- prune.tree(t1,best=7)
summary(t1.best)
```

```
##
## Classification tree:
## snip.tree(tree = t1, nodes = c(7L, 13L, 11L, 4L))
## Variables actually used in tree construction:
## [1] "Ca"        "Thal"      "Age"       "ChestPain" "Slope"     "Sex"
## Number of terminal nodes:  7
## Residual mean deviance:  0.6592 = 125.9 / 191
## Misclassification error rate: 0.1162 = 23 / 198
```

```r
hpred <- predict(t1.best,newdata=Heart[-train,],type="class")
tt <- table(hpred,Heart[-train,]$AHD)
tt
```

```
##
## hpred No Yes
##   No  41  14
##   Yes 11  33
```

```r
sum(tt[row(tt) != col(tt)])/sum(tt)
```

```
## [1] 0.2525253
```

Bagging on Heart Data

```
library(randomForest)
set.seed(1)
bag.heart <- randomForest(AHD ~ ., data=Heart,subset=train,
                          mtry=13,importance=TRUE) #m=p is bagging
bag.heart # Notice OOB estimate of error rate
```

```
##
## Call:
##  randomForest(formula = AHD ~ ., data = Heart, mtry = 13, importance = TRUE,
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 13
##
##         OOB estimate of  error rate: 17.17%
## Confusion matrix:
##     No Yes class.error
## No  92  16   0.1481481
## Yes 18  72   0.2000000
```

Misclassification Error from Test Set

```
hpred <- predict(bag.heart,newdata=Heart[-train,],type="class")
tt <- table(hpred,Heart[-train,]$AHD)
tt
```

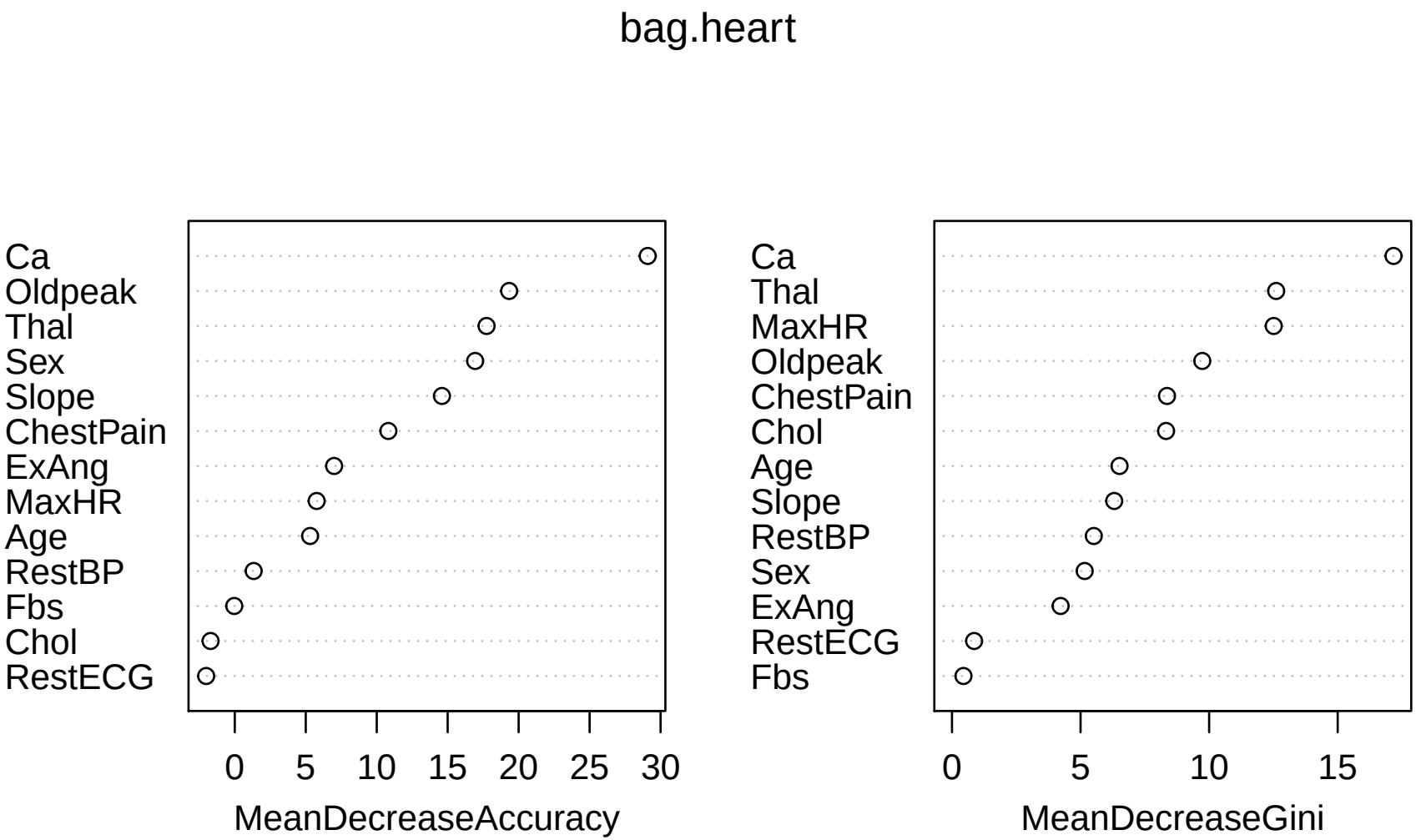
```
##
## hpred No Yes
##   No  42  13
##   Yes 10  34
```

```
sum(tt[row(tt) != col(tt)])/sum(tt)
```

```
## [1] 0.2323232
```

Variable Importance Plot

```
varImpPlot(bag.heart) # or importance(bag.heart)
```



- See `help("importance")` for a definition of accuracy.

Random Forests

- Bootstrap datasets may yield correlated trees.
  - E.G., if we have one or two strong predictors, all trees might share the first few splits.
- Random forests avoid this by restricing the predictors to a random subset.
  - New subset at each split.
  - Subset size? Text: "... typically we choose $m \approx \sqrt{p}$".

Random Forests on Heart Data

```
rf.heart <- randomForest(AHD ~ ., data=Heart,subset=train,
                         mtry=sqrt(13),importance=TRUE) #m<p for RF
rf.heart
```

```
##
## Call:
##  randomForest(formula = AHD ~ ., data = Heart, mtry = sqrt(13),      importa
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 16.16%
## Confusion matrix:
##     No Yes class.error
## No  93  15   0.1388889
## Yes 17  73   0.1888889
```

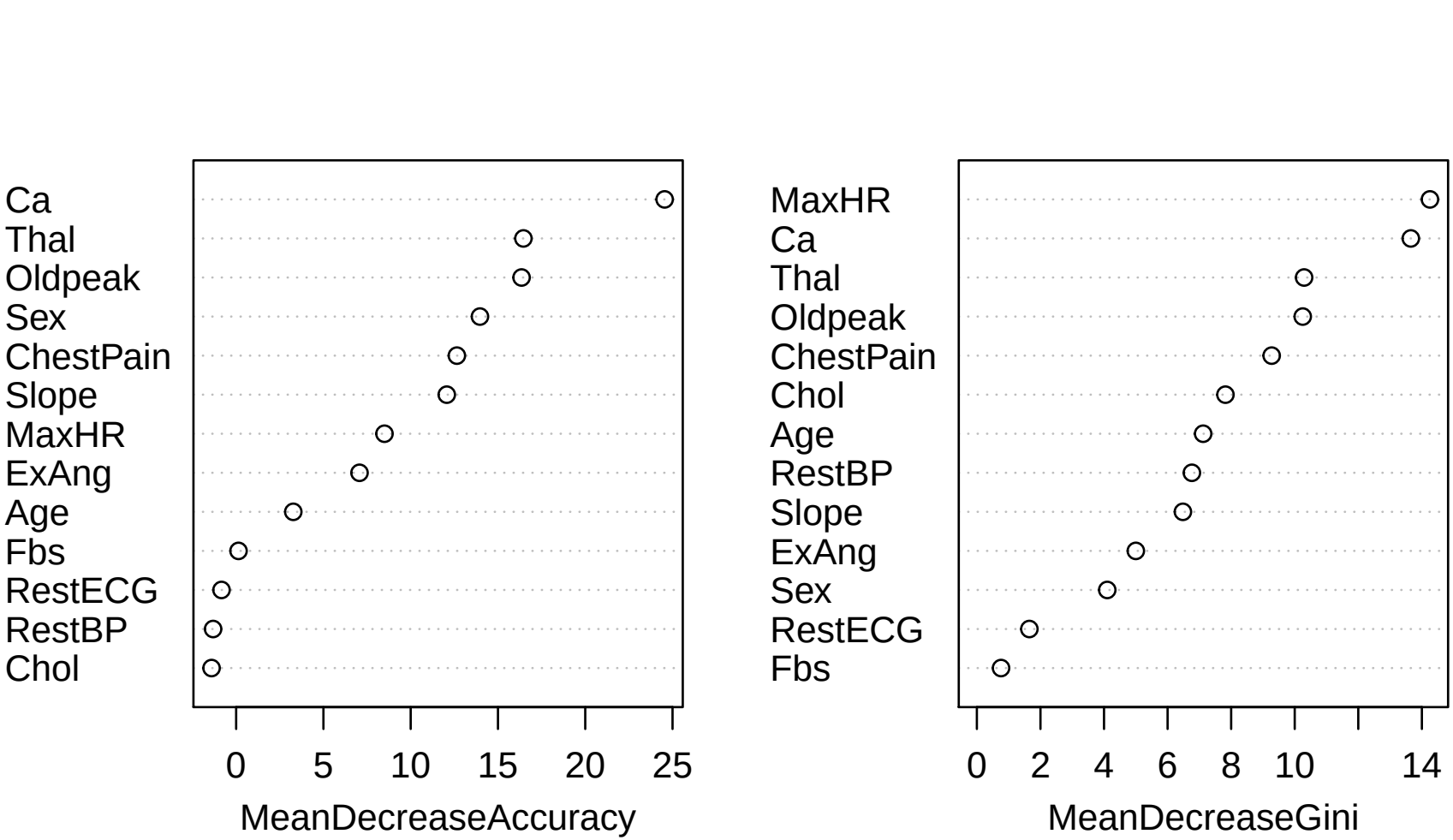
```
rf.hpred <- predict(rf.heart,newdata=Heart[-train,],type="class")
tt <- table(rf.hpred,Heart[-train,]$AHD)
tt
```

```
##
## rf.hpred No Yes
##       No  47  13
##       Yes  5  34
```

```
sum(tt[row(tt) != col(tt)])/sum(tt)
```

```
## [1] 0.1818182
```

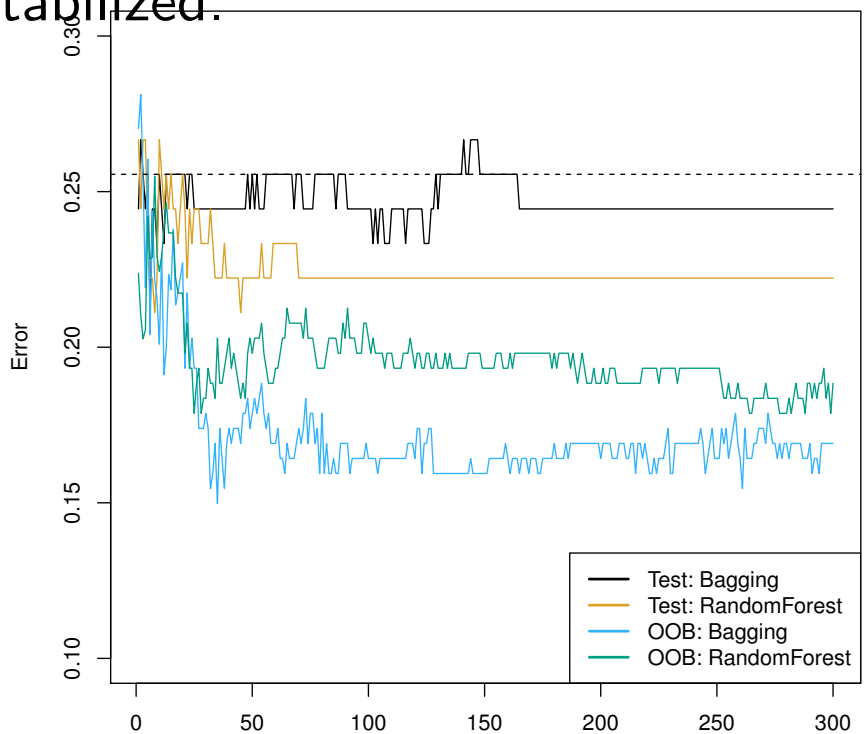
```r
varImpPlot(rf.heart) # or importance(rf.heart)
```



- Chest pain still most important, but not by such a wide margin.

Number of Trees

- Bagging and random forests are not very sensitive to the number of trees B.
- Can try different values and make sure B large enough that error has stabilized.



- Text, Fig. 8.8. Details not important, but can see that the estimated test error levels off at about $B = 75$ trees.