Last Lecture

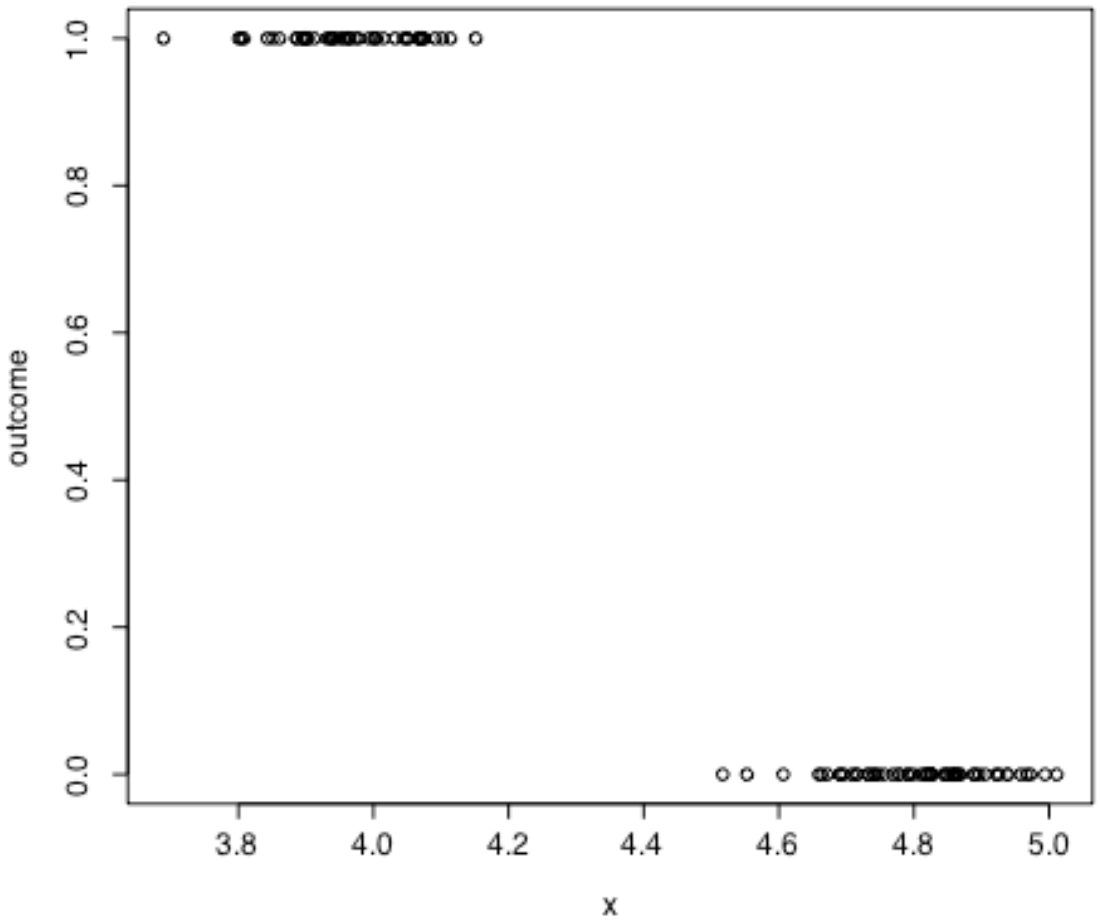
CART: Classification and Regression Tree

This is a technique used for classifying responses. Our goal is to make a prediction about an object based on some observations.

For example, an alcohol researcher wants to know if the amount of alcohol consumed predicts mortality. We have observations on a random sample of a large number of people (n = 1603). At one point in time (early 1970) various characteristics were recorded about this people: their average alcoholic intake, their age, whether or not they smoke, have high blood pressure, etc. We also know, 25 years later, whether they are alive or dead. The object is to see which variables predict death.

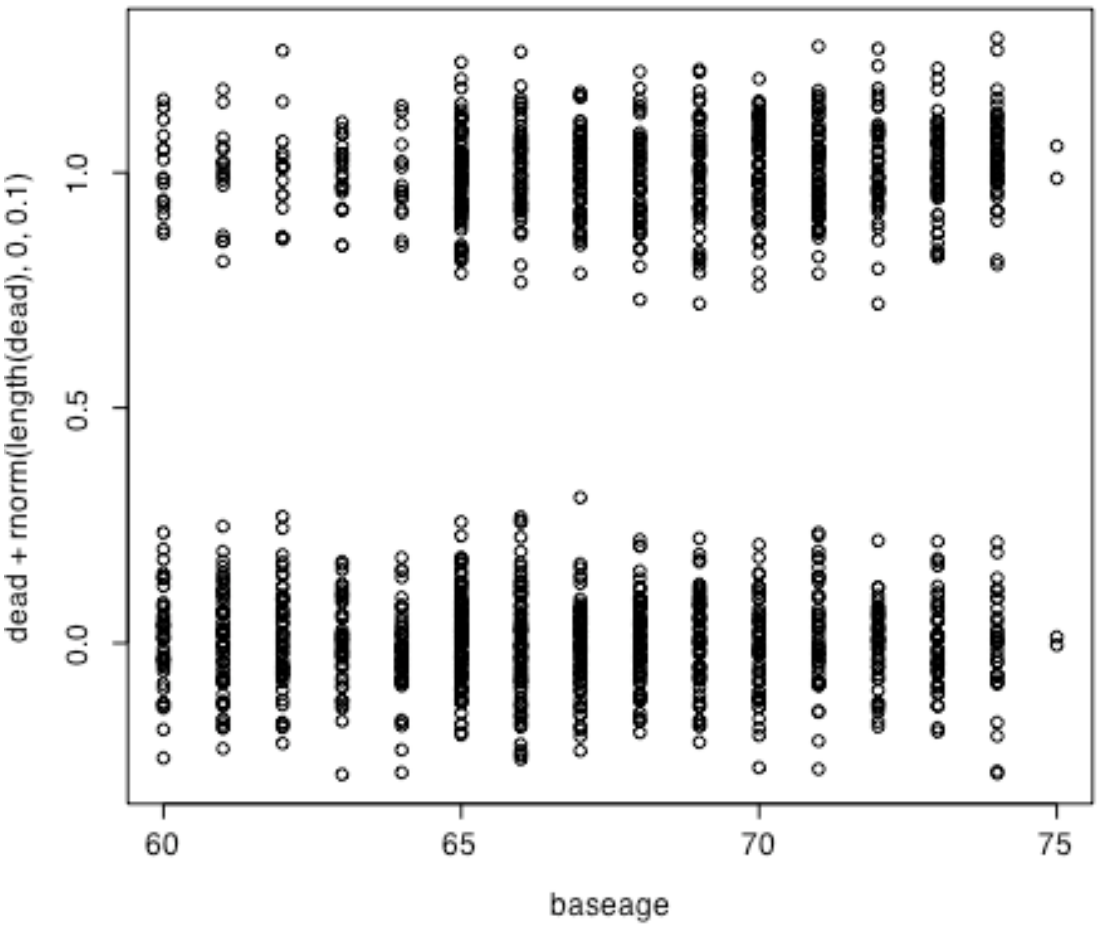
The predictors are continuous and categorical. The response, in this case, is categorical (indeed, simply yes/no). The technique can be easily extended to continuous response variables.

Here's the basic idea. Suppose we have just one continuous predictor. Maybe it's the amount of drinks per week consumed. We want to use it to divide the subjects into two groups. If we're lucky, there's a clean separation:



If this were the case, we would simply say "If x is less than 4.4, classify as 1, if bigger than 4.4 classify as 0."

But usually, things are not so clean:



We'd like to find the "baseage" (age at first interview) that separates the people into two groups. (A "1" means they were dead 20 years later.) It's hard to tell if there's any difference at all, much less to see where to put the dividing line.

Suppose we decide to split into two groups based on whether baseage < 65. We'll decide that those whose age is less than 65 will be "0", and those bigger will be 1. Clearly we'll make mistakes. But how many will be incorrectly classified?

We what need is a measure of homogeneity. If the two groups are all very much alike (all members within the group are the same), we've achieved a perfect split and our classification will work perfectly. Otherwise, we want to find the split that maximizes homogeneity.

Our plan is to go through each value of baseage and examine the homogeneity measurement assuming we split at that value. We do this for each and every value of baseage, and we keep the best one.

In practice, we use something called an impurity measure.

Some terminology: the collection of all subjects is called the "root node". We then split the rootnode into two "child" nodes. This split is chosen so that each node is "pure"-- which means that the nodes are as homogenous as possible. We have two different ways of achieving this purity:

(1) choosing the right split based on the values of a predictor x
(2) choosing the right predictor x from all of the predictors available.

We do this using a brute-force method. We try each variable, and for each variable, all possible splits. We choose the one that produces the two most pure nodes, roughly speaking.

After that, we repeat on each of the two nodes, determining where to split them.

We continue until we have a set of "terminal" nodes that are as pure as possible.

Now it's possible that this will continue until each person has his or her own node. So out of 1600 people, we could end up with 1600 nodes. This is called a "saturated" or "over fitted" tree. This isn't terribly useful, because it means it takes lots of information to make a prediction. It might also mean that the tree is too sensitive to the sample -- we found the best tree for this sample, but it might not be best for the population.

There are different procedures for knowing when to stop. One is to stop when there are 5 or more individuals in each terminal node. Another is to stop when the node is smaller than .01n (one percent of the sample size.) A more sophisticated method is called "pruning". Here you grow the tree until it is saturated, and then you "prune" off the nodes and find the best sub-tree (it's a little like backwards stepwise regression).

We've grown this tree by making the best decision for each node. Now we might want to evaluate how good the entire tree is by throwing a few individuals at it -- individuals for whom we know the true outcome, and we can determine the misclassification rate.

So let's examine the alcohol data:

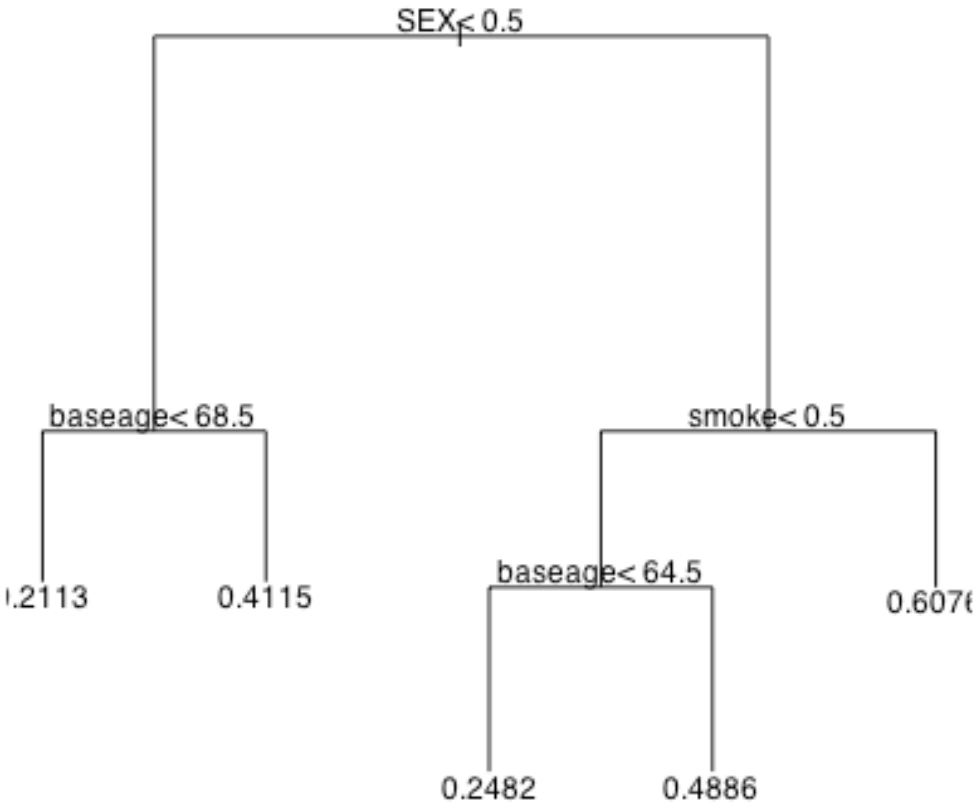
```
> names(drink.dead)
 [1] "dead"      "qfi1"      "baseage"   "SEX"       "smoke"     "hbp"       "diabetes"  "gout"
"hepatitis"
[10] "ulcer"     "anxious"   "stompain"  "heartburn"
```

We need to load the rpart library

```
library(rpart)
```

```
> fit <- rpart(dead~.,data=drink.dead)
> plot(fit)
```

> text(fit)

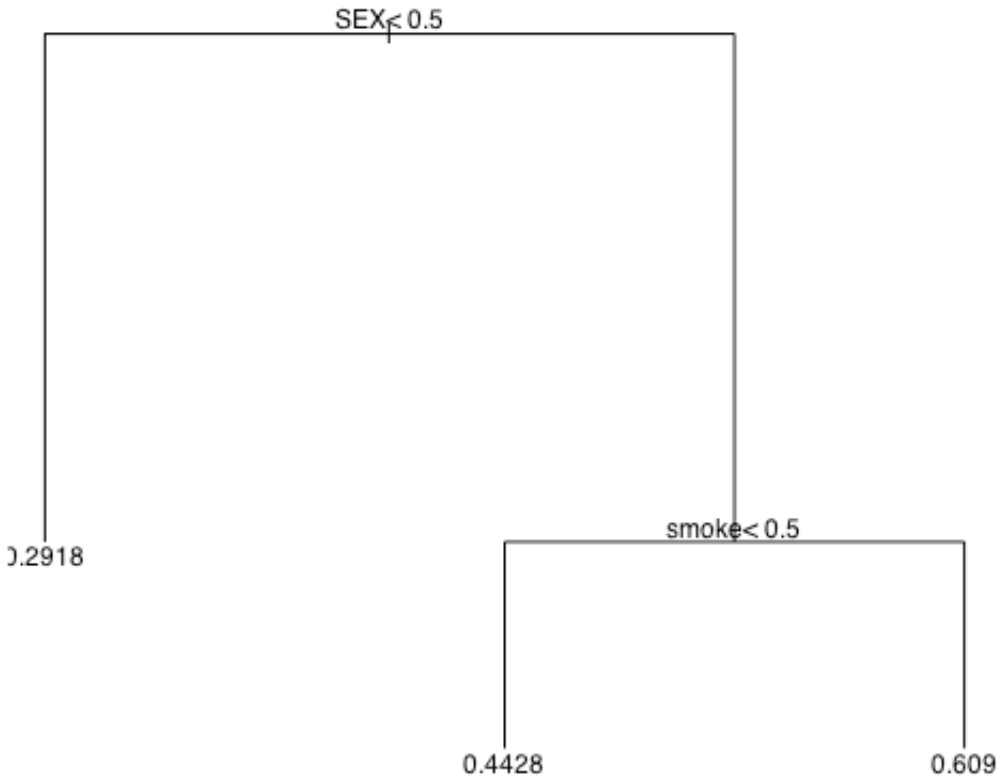


The numbers in the nodes are the average value. So the .2133 means that 21% are dead.

What this tells us is that survival depends on gender. If you are a woman (sex < .5) then the only thing that predicts survival is your age. If you were less than 68.5 at the time of the survey, you have about an 80% survival rate, otherwise about a 60%. For men, survival depends on smoking. Smokers have a 40% survival rate. Non-smokers, it depends on age, with the older smokers being less likely to survive.

Age is an irritating variable in these studies, because it is always the best predictor of survival. Let's take it out.

> fit2 <- rpart(dead~.-baseage,data=drink.dead)
> plot(fit2)
> text(fit2)



Now we see that for men, smoking is dangerous. And that seems to be all that matters.

What is interesting is that alcohol, apparently, does not matter, once we've taken into account gender and smoking.

There are other things we can play with. We can use different impurity measures. We can implement a cost function, which makes it harder or easier to split a node into two.

One of the challenges is to create a tree that will be useful for predicting for the population, and not just for the sample.

In recent years there has been lots of activity in this area. This has resulted in something called "forests" and "bagging" and "boosting". All our various computationally intensive strategies for getting robust predictive models.

A summary command gives you the blow-by-blow account:
> summary(fit)
Call:

```
rpart(formula = dead ~ ., data = drink.dead)
  n= 1603

         CP nsplit rel error    xerror        xstd
1 0.03632701      0 1.0000000 1.0019996 0.008145306
2 0.01565860      1 0.9636730 0.9902093 0.011202793
3 0.01386925      3 0.9323558 0.9740094 0.014271803
4 0.01000000      4 0.9184865 0.9576297 0.014761045

Node number 1: 1603 observations,    complexity param=0.03632701
  mean=0.4198378, MSE=0.243574
  left son=2 (562 obs) right son=3 (1041 obs)
  Primary splits:
      SEX      < 0.5  to the left,  improve=0.036327010, (0 missing)
      smoke    < 0.5  to the left,  improve=0.034694560, (190 missing)
      baseage  < 64.5 to the left,  improve=0.031258980, (0 missing)
      diabetes < 0.5  to the left,  improve=0.008468137, (0 missing)
      hbp      < 0.5  to the left,  improve=0.008171825, (0 missing)
  Surrogate splits:
      anxious < 0.5  to the right, agree=0.651, adj=0.005, (0 split)

Node number 2: 562 observations,    complexity param=0.01386925
  mean=0.2918149, MSE=0.206659
  left son=4 (336 obs) right son=5 (226 obs)
  Primary splits:
      baseage   < 68.5 to the left,  improve=0.0466258600, (0 missing)
      hbp       < 0.5  to the left,  improve=0.0120213100, (0 missing)
      anxious   < 0.5  to the left,  improve=0.0055831710, (0 missing)
      qfi1      < 24.5 to the left,  improve=0.0030723750, (4 missing)
      heartburn < 0.5  to the right, improve=0.0008747111, (0 missing)

Node number 3: 1041 observations,    complexity param=0.0156586
  mean=0.4889529, MSE=0.249878
  left son=6 (753 obs) right son=7 (288 obs)
  Primary splits:
      smoke     < 0.5  to the left,  improve=0.06043668, (145 missing)
      baseage   < 64.5 to the left,  improve=0.03014446, (0 missing)
      hbp       < 0.5  to the left,  improve=0.01413455, (0 missing)
      diabetes  < 0.5  to the left,  improve=0.01383345, (0 missing)
      heartburn < 0.5  to the right, improve=0.00575115, (0 missing)
  Surrogate splits:
      baseage < 74.5 to the left,  agree=0.681, adj=0.007, (145 split)
      qfi1    < 66.5 to the left,  agree=0.680, adj=0.003, (0 split)

Node number 4: 336 observations
  mean=0.2113095, MSE=0.1666578
```

```
Node number 5: 226 observations
  mean=0.4115044, MSE=0.2421685

Node number 6: 753 observations,    complexity param=0.0156586
  mean=0.4435591, MSE=0.2468144
  left son=12 (141 obs) right son=13 (612 obs)
  Primary splits:
      baseage   < 64.5 to the left,  improve=0.035615960, (0 missing)
      hbp       < 0.5  to the left,  improve=0.014098680, (0 missing)
      diabetes  < 0.5  to the left,  improve=0.011016010, (0 missing)
      anxious   < 0.5  to the left,  improve=0.007070408, (0 missing)
      heartburn < 0.5  to the right, improve=0.005765263, (0 missing)
  Surrogate splits:
      hepatitis < 0.5  to the right, agree=0.815, adj=0.014, (0 split)

Node number 7: 288 observations
  mean=0.6076389, MSE=0.2384139

Node number 12: 141 observations
  mean=0.248227, MSE=0.1866103

Node number 13: 612 observations
  mean=0.4885621, MSE=0.2498692
```

This is useful because we don't always want ot trust the machine. This tells us, for example, that at several stages, high blood presure (hbp) is a possibly useful predictor, and the drinking variable (qfi) plays a role, too.