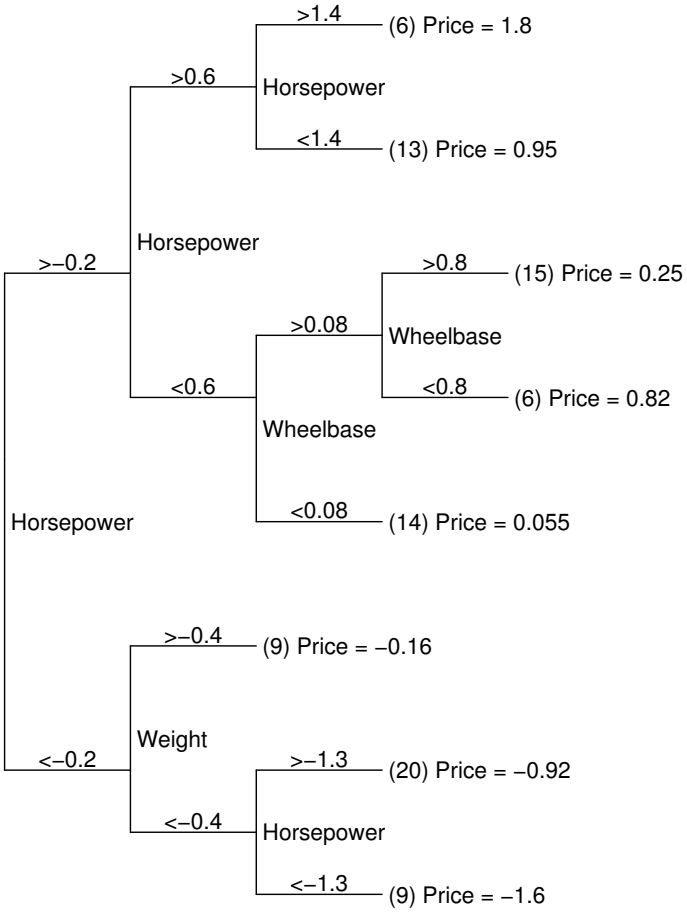
36-350: Data Mining

Handout 14
October 13, 2003

Modeling interactions via trees

If a dataset has many interactions, regression equations can become complex and unmanageable. Another approach is to split the data into subgroups, and use a simple model in each group.

A **regression tree** makes predictions by asking a series of questions, each one dependent on the answers to the previous questions. The questions are chosen to be the **most informative** given the answers so far. An estimate of the response is available at each step.



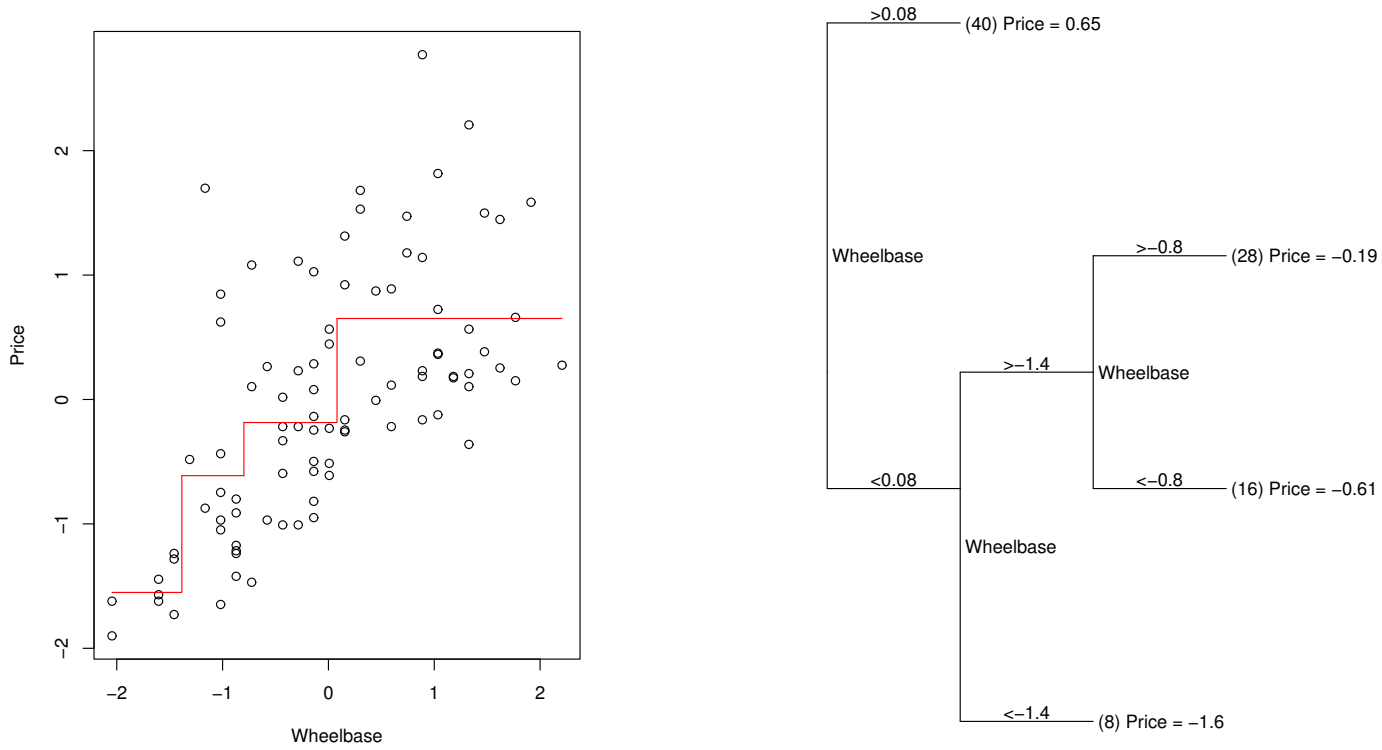
The questions are usually about one attribute, and have a yes/no answer, such as "Is `Horsepower` > -0.2?" Thus they repeatedly split the data into subgroups.

This tree has $R^2 = 0.85$. Linear regression has $R^2 = 0.8$.

Let c be the outcome of the question, and y the response. The best question to ask maximizes $\mathcal{I}(c, y)$. Usually this is computed by assuming the response is normal and has the *same variance* in each group, leading to the simple formula given in handout 10. Equivalently, the best question explains as much of the variance in the response as possible (has highest R^2).

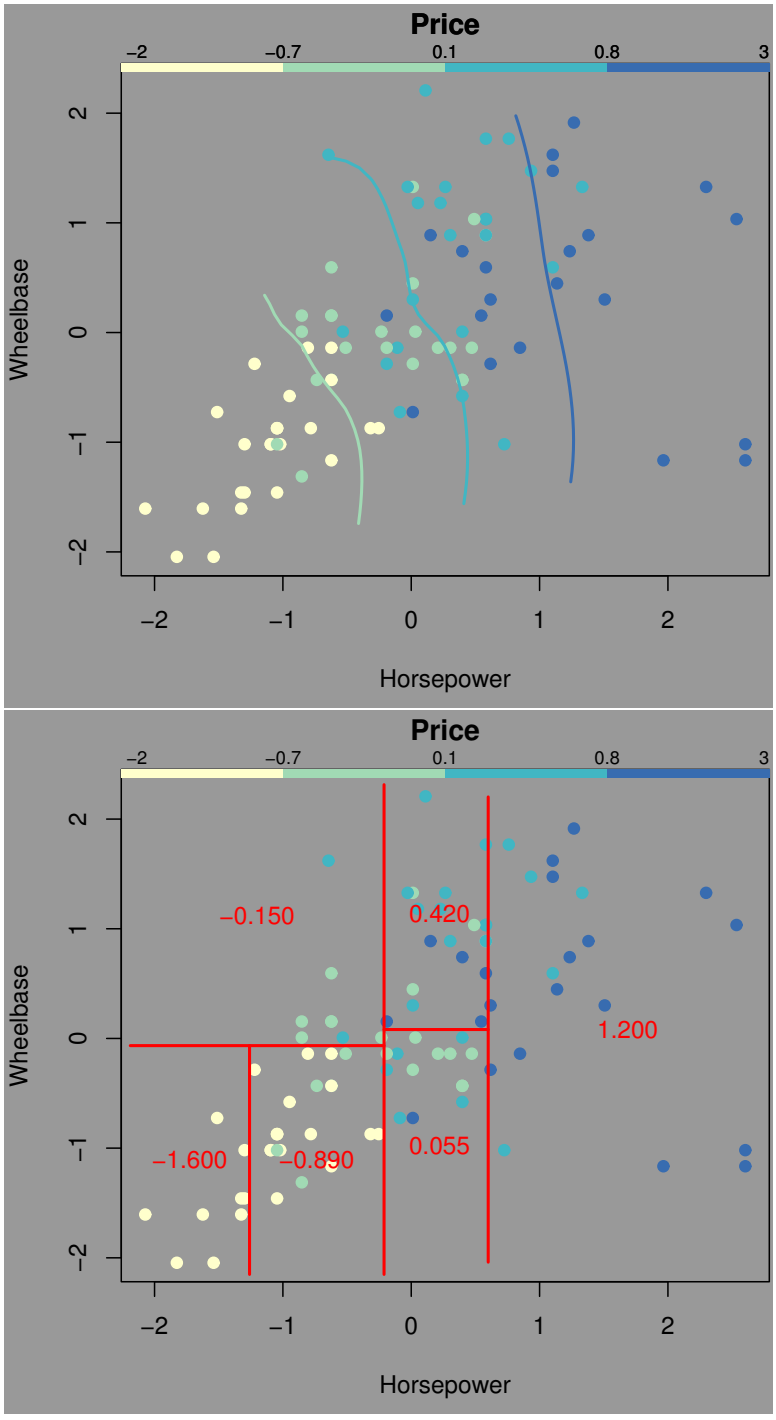
Thus the tree not only subdivides the data but also tells you the most important predictor in each group. They also make few assumptions about the regression function—they don't require linearity or smoothness.

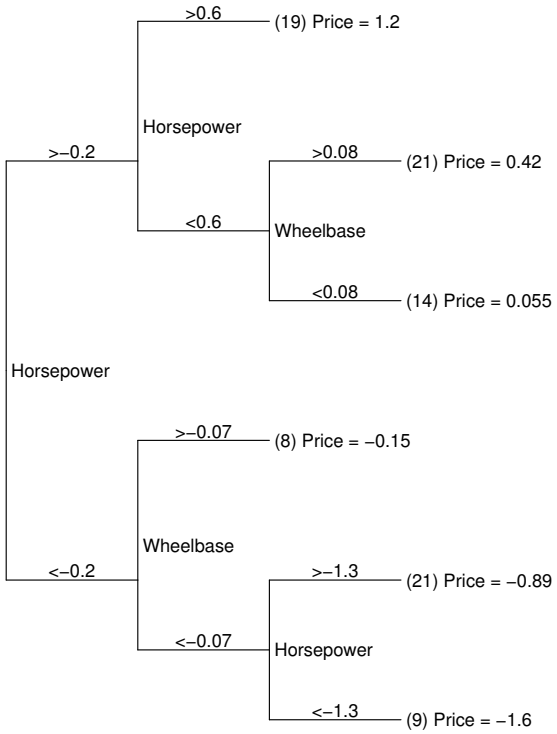
An example using one predictor:



The tree captures the nonlinearity of the function, though not the smoothness.

With two predictors, the splits of the tree can be visualized as horizontal and vertical lines.





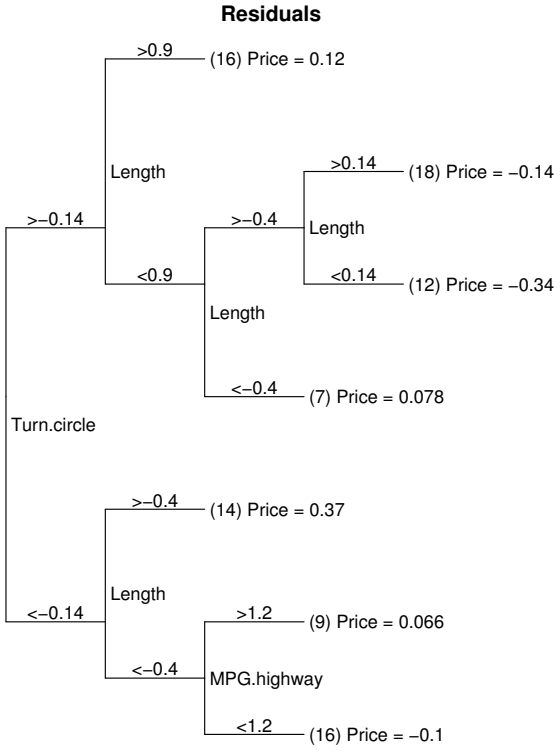
The tree correctly represents the interaction between `Horsepower` and `Wheelbase`. When `Horsepower` > 0.6, `Wheelbase` no longer matters. When both are equally important, the tree switches between them.

Trees attempt to be concise by using only one predictor at each step. If multiple predictors are equally important, the tree will switch between the predictors at random, which can give a misleading impression. (For example, `Weight` is just as good as `Wheelbase` in the previous tree.) If you want to know about all the important variables, projection is better.

It is possible to extend trees to make **diagonal splits**, involving multiple predictors at a time. How could you determine the most informative combination of variables?

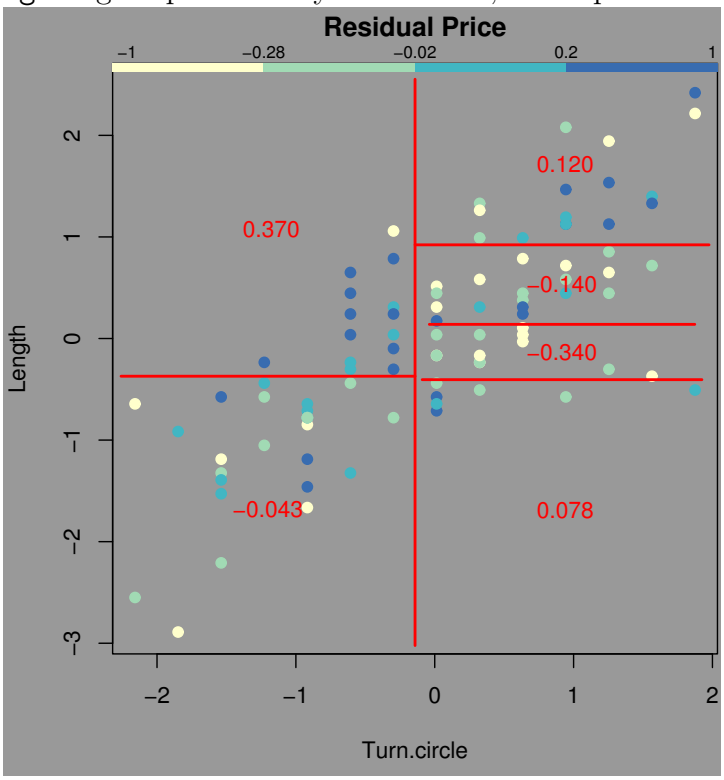
Trees give coarse predictions, because they only use the mean response in each group. This can be an advantage in data mining. The coarse predictions represent the dominant factors influencing the response, which are typically obvious and uninteresting. By removing these factors, we can get at the **second-order effects**, which are often interesting and actionable. This is done by examining **residuals** of the regression tree.

Here is a regression tree to predict the residuals:



It explains 30% of the variance in the residuals, and together with the first tree explains 90% of the variance in `Price`.

`Turn.circle` and `Length` figure prominently in this tree, so we plot them separately:



The upper left box captures 14 'overpriced' cars. Longer cars generally require more space to turn. But some long cars are maneuverable enough to make tight turns, and this is reflected in their price. This group includes the Mercedes-Benz 300E and Saab 900. (The low-priced member of the group is the Mercury Cougar.)

This is the type of second-order effect we were looking for. To see how subtle it is, note that `Turn.circle` is considered irrelevant to `Price` by the original tree, the regression projection, and the linear regression from last week.