Handout #6

Lesson 5: Model Selection

To begin, let's load the dataset "mtcars." This dataset was extracted from the 1974 *Motor Trend US* magazine, and contains 11 variables on aspects of automobile design and performance for 32 automobiles (1973-74 models). Here are descriptions of the variables:

[, 1] mpg Miles/(US) gallon
[, 2] cyl Number of cylinders
[, 3] disp Displacement (cu.in.)
[, 4] hp Gross horsepower
[, 5] drat Rear axle ratio
[, 6] wt Weight (lb/1000)
[, 7] qsec 1/4 mile time
[, 8] vs V/S
[, 9] am Transmission (0 = automatic, 1 = manual)
[,10] gear Number of forward gears
[,11] carb Number of carburetors

Typically, people use this dataset to predict fuel efficiency (mpg) as a function of the other variables, but I think it's more interesting to predict "qsec": that is, what makes a car fast?

First, take a look at the data. All of the variables are numeric. Does this make sense? Should any of them be converted to factors?

Once you're more or less satisfied with the structure of the data frame, you first job is to pick one variable as the best predictor of "qsec." Take your time on this – which variable did you pick? How did you choose it? What criterion did you hope to optimize? Once you pick your favorite variable, fit the model and look at the results.

Let's see what happens if we decide to let R choose a variable for us. The "addterm" function, available through the library MASS, will consider all possible one-term additions to an existing model, based on a scope that you set. So, if we wanted to consider all possible one-variable models, we can do it this way:

```
>library(MASS)
>mod.null=lm(qsec~1,data=mtcars)
>mod.all=lm(qsec~., data=mtcars)
>addterm(mod.null,mod.all,test= "F")
```

Take a look at the output. Most of it is probably familiar, except for the column "AIC." That's the Akaike Information Criterion, which is defined by

AIC = -2(maximized log-likelihood) + 2(# parameters)

(See p.174 of Venables and Ripley to review what this means in the least-squares regression setting.)

The AIC is basically a penalized likelihood statistic, which attempts to strike a balance between the size of a model and the quality of the fit. So, the smaller the AIC, the better.

Back to the "qsec" model. Did the selected variable agree with your preference? Let's fit a model based on the automated selection process and see how it looks.

Now that you've done this, try it again. Is there another term that seems good to include? Keep going until you've picked the best model. How did you do? Are there any outliers that the model doesn't handle well?

The "update" function will save you some typing. Say you have a model (call this "mod1") that uses a single predictor var1, and you'd like to add a second variable, var2. The command

```
>mod2=update(mod1, ~. + var2)
```

will create a new model with the term added. This may not seem like a big deal, but for a model with 20 terms it'll keep you from getting a cramp…

You might also want to use the "dropterm" function at various stages, which will check whether or not a term should be removed. The syntax is easy: just type `dropterm(my.mod, test="F")` to evaluate single-term deletions from a model "my.mod."

Another alternative: let R do everything!

Now that you've got a model that you're really proud of, let's see what would've happened if we let R do all the work. The MASS function `stepAIC` will evaluate a full range of models and spit out which one seems optimum according to the AIC. It seems to work best if you start big and let it get smaller, so you could try

```
>mod.step=stepAIC(mod.all,scope=c(lower=~1,upper=~.))
```

Scroll through the output to see how the evaluation was done. How different is the model from your model? Is it better? Do you think you could improve it? Remember, the AIC can be pretty conservative, so you might decide that you're not entirely happy with the outcome.