Lab 6: Model Selection in SAS
STT 422: Summer, 2004
Vince Melfi

It's relatively easy to use automatic model selection procedures in SAS. A large number of such procedures are available; in this lab we'll learn how to investigate forward selection, backward elimination, and stepwise model selection methods. We will also investigate some of the potential pitfalls of model selection procedures.

For the most part we'll be using a data set on homicide rates in Detroit from 1961–1973. The variables are

`HOMICIDE`: The homicide rate per 100000 residents. (The response variable.)
`POLICE`: Number of full-time police per 100000 residents.
`UNEMP`: Percent of unemployed people in the population.
`MFG_WRK`: Number of manufacturing workers (in thousands).
`GUN_LIC`: Handgun licenses per 100000 residents.
`GUN_REG`: Handgun registrations per 100000 residents.
`H_ARREST`: Percent of homicides cleared by arrests.
`W_MALE`: Number of white male residents.
`NMFG_WRK`: Number of non-manufacturing workers (in thousands).
`GOV_WRK`: Number of government workers (in thousands).
`H_EARN`: Average hourly earnings in dollars.
`W_EARN`: Average weekly earnings in dollars.
`ACC`: Death rate from accidents per 100000 residents. This variable won't be used in the model.
`ASSAULTS`: Number of assaults per 100000 residents. This variable won't be used in the model.

The following SAS program reads the data into SAS and displays it.

```
data detroit;
infile 'u:\msu\course\stt\422\summer04\detroit.dat';
input
       POLICE UNEMP MFG_WRK GUN_LIC GUN_REG H_ARREST W_MALE NMFG_WRK GOV_WRK
       H_EARN W_EARN HOMICIDE ACC ASSAULTS;

run;

proc print data = detroit;

run;
```

Forward selection

First we'll fit a model including all of the predictors.

```
proc reg data = detroit;
```

```
fullmodel:   model homicide = police--w_earn;

run;
```

Note that we didn't have to write the names of all the predictors explicitly. The specification `police--w_earn` tells SAS to include the variables starting with `police` and ending with w_earn as predictors. Note also that we've included the label `fullmodel` before the model statement. This label appears in the **proc reg** output. When we're fitting several models, labels help us to figure out which model generated the output.

Take a look at the output of this regression fit. Note that in this full model, none of the predictors are significant at the 5% level. Now we'll perform forward selection.

```
proc reg data = detroit;
forwardselect: model homicide = police--w_earn /selection = forward;

run;
```

Which variable entered first? What was the value of its F statistic? Which variable entered second? What was the value of its F statistic? Which of the 11 models obtained by forward selection seems best to you?

We can control how many models are considered by specifying a cutoff p-value for the F statistic for adding the variable to the model. If no variables have p-values less than the cutoff, the selection procedure is stopped. By default SAS sets the cutoff to 0.5. In the following example, the cutoff is set to 0.05.

```
proc reg data = detroit;
forwardselect2: model homicide = police--w_earn /selection = forward
     slentry = 0.05;

run;
```

How many models were considered in this case?

Backward elimination

Now we use backward elimination to select a model. Again, the user can specify a cutoff p-value, this time called `SLSTAY`. In this case, backward elimination stops when all the variables remaining in the model produce F statistics with p-values less than the cutoff. The default cutoff, which we use below, is 0.10.

```
proc reg data = detroit;
backwardelim: model homicide = police--w_earn /selection = backward;

run;
```

Note that the variable `h_arrest`, which was the first to enter the model when using forward selection, is the first to exit the model when using backward elimination!

Stepwise selection

Recall that in stepwise selection, variables are added as in forward selection, but after a variable is added, all the variables in the model are candidates for removal. There are two cutoffs to be specified, SLENTRY and SLSTAY. In the following example we specify cutoffs of 0.10 and 0.15 respectively.

```
proc reg data = detroit;
stepwise: model homicide = police--w_earn /selection = stepwise
  slentry = 0.10 slstay = 0.15;

run;
```

A better model?

There are several reasons to be cautious when using model selection techniques. Good texts on linear regression have discussions of these. We'll investigate one of these next, and one more in the exercise below.

It is tempting to assume that, for example, the three-predictor model chosen by the forward selection or stepwise procedure is the best three-predictor model. The following shows that this isn't true, at least in terms of the R^2 value for the model.

```
proc reg data = detroit;
   model homicide = unemp gun_lic w_earn;

run;
```

Compare the fit of this model to the three-predictor model chosen by stepwise.

Exercise

In this exercise you will learn that model selection methods, especially with a small number of observations relative to the number of potential predictors, can "discover" spurious relationships. We will generate an artificial dataset in which all the predictors are independent of the response, but the model selection methods will find models that indicate rather strong relationships between the predictors and the response.

We'll start by generating a small dataset, so you'll understand what's being done in the full simulation. In this case we'll generate 5 predictors x1 through x5, all of which are i.i.d. standard normal and independent of one another. We'll also generate a response y which is itself standard normal and is independent of all the predictors.

```
data smallsim;
   array x{5} x1-x5;
   do n=1 to 20;
         do i=1 to 5;
            x(i) = normal(-1);
         end;
```

```
            y = normal(-1);
            output;
   end;

run;

proc print data = smallsim;

run;
```

Here's an explanation.

1. We name the dataset `smallsim` and declare an array that will have 5 variables named `x1, x2, x3, x4, x5`.
2. We then start a loop, which lets `n` run from 1 through 20. This will generate 20 observations for each of the variables.
3. Next comes a loop which lets `i` run from 1 through 5. This controls which of the variables `x1`–`x5` we'll be creating.
4. The line `x(i) = normal(-1);` generates one standard normal observation. The value of `i` controls which of the predictors we're creating, and the value of `n` controls which of the 20 observations we're creating. The argument `-1` of `normal` is related to the "seed" of the random number genearator. Don't worry about this.
5. We end the inner loop when we're done with the predictors, then generate the response `y`.
6. We output all the variables we've created, then end the outer loop.

Question (a): Take a look at the data set you've generated. *Print the results of* **proc print** *and turn in the printout.*

Now we'll increase the number of predictors and the number of observations and use variable selection. Here is the code to generate the data set with 75 potential predictors and 100 observations.

```
data sim;
   array x{75} x1-x75;
   do n=1 to 100;
            do i=1 to 75;
               x(i) = normal(-1);
            end;
            y = normal(-1);
            output;
   end;

run;
```

Question (b): Run backward elimination on the data set using the default value for `slstay`. What model is chosen at the end of the backward elimination procedure? What is the F statistic and p-value for testing the overall fit of this model?

Question (c): Run stepwise selection on the data set using the default values for `slstay` and `slentry`. What model is chosen at the end of the stepwise selection procedure? What is the F statistic and p-value for testing the overall fit of this model?