Macro Problems

Exercise 1 *Use a macro variable instead of a* `libname`*. Use the nlswages data set in "y:\sas" and write a* `proc means` *that calculates the mean value of all numeric variables.*

In solving this problem, you will encounter a technical problem with SAS scanning your macro variable name. For example, if you coded

```
%let text = log;
proc &text istic data=deathpenalty;
```

SAS correctly interprets `&text` as a macro variable name, but after it is resolved you have `proc log istic`, and SAS complains that it cannot find "proc log." However, if you try `proc &textistic` in order to eliminate the space, SAS complains that it cannot resolve the symbolic reference (macro variable name) "textistic." What you need to know is that when you are calling a macro variable, the end of the macro variable name can be demarcated with a period. In this example `proc &text.istic` resolves correctly.

Exercise 2 *Both Stata and SPSS have commands that allow you to easily recode continuous data into categories. Write a SAS macro with similar functionality. Your macro should allow you to specify what variable you want to recode, should specify the number of categories you want your recoded variable to have, and should preserve your original variable in the final data set. (You might include other parameters as well.)*

To solve this problem, you will at some point have to figure out the range of your variable. You could do this as a separate step, and feed the appropriate values to your macro, or you can have SAS do this for you. To have SAS automate this, you need to know how to turn data values into macro variable values. Because we will be pulling specific data values out of a SAS data set, this will happen in the context of a data step.

The key element here is a data step statement, `call symput`. For example

```
data _null_;
    set y.nlswages;
    call symput('nobs', _n_);
run;
```

would create a new macro variable, `&nobs`, which had as its value the number of observations in the data set y.nlswages. As a second example

```
data _null_;
    set y.nlswages;
    if _n_ eq 1 then call symput('firstvalue', R0043600);
run;
```

would create a new macro variable, &firstvalue, which has the data value of the variable R0043600 from the first observation in the data set.