sat.sas: Explanation of code

Goals of code:

- Reading an excel worksheet
- Model selection statistics
- Diagnostics for multiple regression
- Storing model selection results in a data set

Reading an excel worksheet: `proc import`; Proc Import can read .xslx or .xls files. Just provide the appropriate filename and give a name for the resulting SAS data set. Remember `replace` allows SAS to overwrite an old copy of the data set. If you only have one sheet in the workbook, that is what is read. Doesn't matter whether that sheet has the default name of Sheet1 or some other name.

When there are two or more sheets, you have to tell proc import which sheet you want to read. That is done by `sheet=' name ';`. This is a separate command, unlike `out=` or `replace`. `sheet=' name ';` goes after the `;` at the end of `proc import` and before the `run;`. Other commands, such as `guessingrows=` also go between the proc statement and run statements. The SAT file has one sheet, called sat, so `sheet='sat';` will read that sheet by name. It is commented out because it isn't necessary here.

All the rest of this week's information concerns `proc reg`. Although both proc glm and proc reg fit regression models, proc reg provides the better set of tools for multiple regression. The only time I revert to proc glm for multiple regression is when I want SAS to create indicator variables automatically.

Model selection statistics: `prog reg; model / selection = cp`
Proc reg does all subsets variable selection by adding the option `/ selection = cp` to the model statement. The variable list (right-hand side of the model equation) lists the variables to be considered. Variables not on this list are ignored. By default, you will get information about every possible model (not recommended!). The `best=10` option requests information about the 10 best models. Changing the 10 changes the number of models that are printed.

The AIC and sbc options request that the AIC and BIC statistic are printed for each model. There are a couple of versions of the BIC criterion. SBC is SAS's name what I (and most others) call BIC.

Proc reg ranks models by the Cp criterion. To find the best model by AIC or BIC, look at the top 10 (or perhaps top 20) models using Cp and look for the best AIC or BIC. This doesn't guarantee finding the best AIC or BIC, but it almost always works. In my experience, the only time there is a risk of the best AIC/BIC model not in the top 10 or 20 Cp models is when there are many

variables and many models close to the best. If this is a potential concern, see the optional code at the end of the file (and explained at the end of this document).

Diagnostics for multiple regression
The `/ vif` option requests the VIF (variance inflation factor) statistic for each regression parameter.

Diagnostics for individual points are available using the output statement. Keywords request specific values: `p=` gives the predicted values (used frequently before), `rstudent=` gives the studentized residuals, and `cookd=` gives the Cook's distance value. The `data sat2;` data step at the beginning of the example code creates the variable i containing the observation number.

Model selection statistics - version 2: `ods html exclude all;` etc.
If you really want to make sure you have the best model by AIC or BIC, an alternative strategy is to store all the model selection results in a data set, then sorting and printing those results.

The `ODS` commands provide the ability to store any table of results in a data set. More generally, ODS provides considerable control over how SAS saves output. `ODS` stands for Output Delivery System. We've already used two features of ODS:

- `ods html close; ods html;` The `html` destination describes what goes in the Results Viewer window. This pair of commands closes the output and restarts it. The result is that the previous contents of the Results Viewer window are erased.
- `ods rtf file='name.rtf';` This starts the `rtf` destination in addition to the `html` destination. The result is that all results go to both the Results Viewer window and to the file you specify. The `ods rtf close;` closes that destination and saves the file. The result is a file that Word can read.

The example code uses two more features of the ODS.

- `ods html exclude all;` tells SAS to not put any output into the results viewer window. This takes effect the next time output would be generated (e.g., the next proc step) and lasts until rescinded by `ods html select all;`.
- `ods output subsetselsummary= dsname;` tells SAS to save a copy of one specific piece of output in a SAS data set. You provide the name of that data set after the `=` sign. The example code creates a data set called models. The `subsetselsummary=` piece tells SAS which piece of output you want. Each separate block of output in the Results Viewer has a name. `subsetselsummary` is the name of the piece of output with the AIC, BIC, and Cp statistics.

  The hardest part about using `ods output` is finding the SAS name for the desired piece of output. I will tell you the names of the pieces that we will use in class. If you want something else, there are a couple of ways to get the appropriate name. `ods html trace on;` will print

the name for each piece of output, but the easiest way to get one specific name is to go to the SAS online help system. The link is at the top of the SAS lab page. You want the procedures guide, then by name. Click on the procedure (e.g. reg), then look under the Details tab. One menu item, usually at or very near the bottom of the drop-down list, is `ODS Table names`. Select that and you get a list of every piece of output, many produced only by specific statements or specific options. The first column gives the SAS name for each piece of output.

Once the numbers are saved in a data set, you can sort observations by the desired statistic (proc sort). If you then print the first 5 or first 10 (proc print data=models(obs=10)) observations, you get the best models by the criterion used in the sort.