STA 302 / 1001
Introduction to SAS for Regression
(on CQUEST)

A.L. Gibbs

September 2011

Some Basics of CQUEST

- The operating system in the RW labs (107/109 and 211) is Windows XP. There are a few terminals in RW 213 running Linux. SAS is run from a Linux server. Your account is the same and files are accessible in all labs / operating systems.
- To logout:
  - Windows: look under *Start*
  - Linux: look under *System*
- To open a web browser:
  - Windows: *Firefox* is an icon on your desktop.
  - Linux: *Firefox* is the mouse on a globe icon on the toolbar.
- If `backspace` doesn't work in SAS, turn off `Num Lock`.
- Use the `Insert` key to toggle between insert and type-over modes.
- Linux: If you want a terminal window you can either right click on the desktop or click on *Applications* and then *Accessories*.

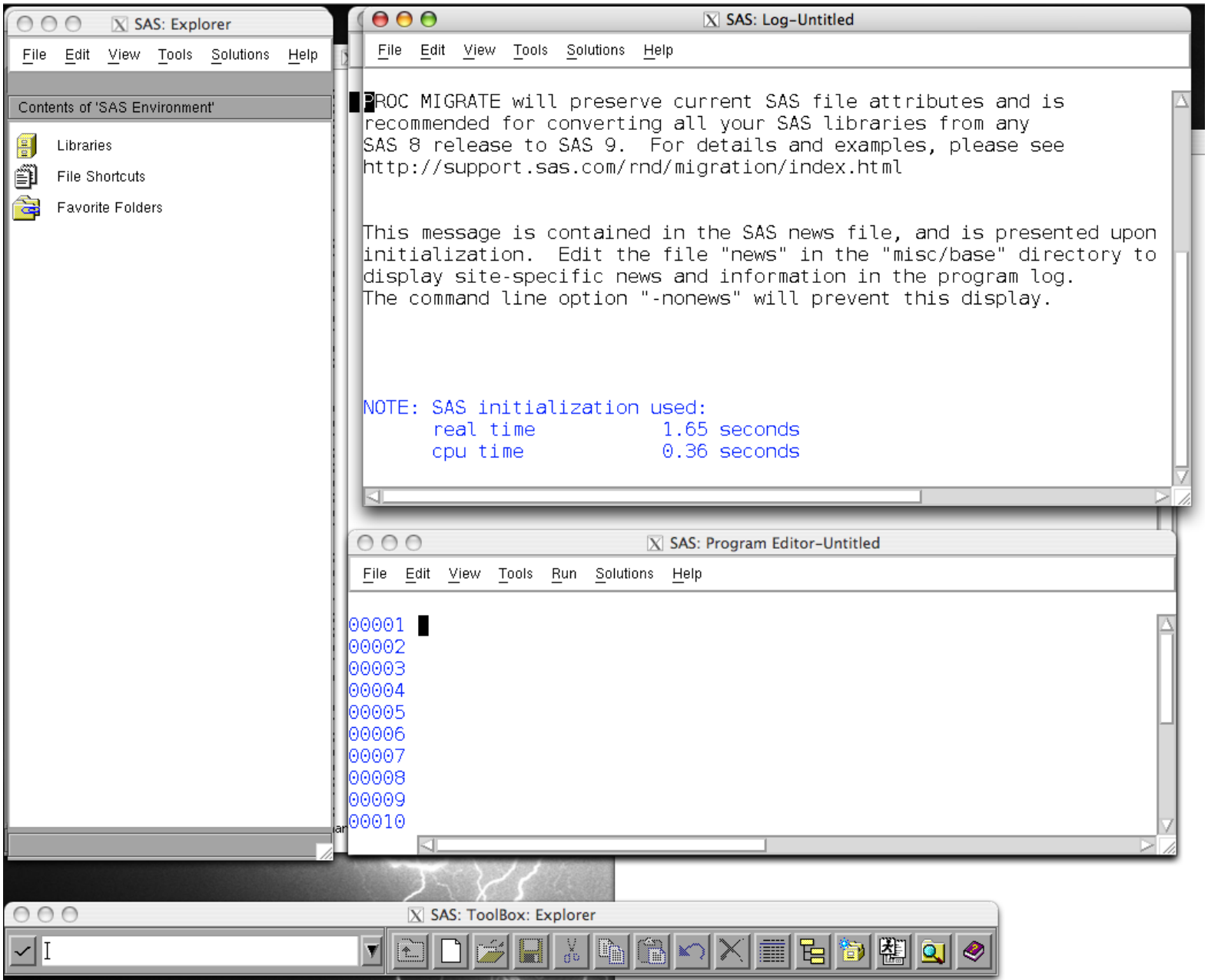
Accessing CQUEST Remotely

- Mac: Use an X11 terminal window. Command:
  `ssh -X -l cquestuserid login.cquest.utoronto.ca`
- Windows: need PuTTY and Xming (free downloads). See the course announcements on Blackboard for a link to complete instructions to set this up.
- All files will be on CQUEST and **not** on the computer you're working on. So the data file needs to be saved on your CQUEST account and SAS output will be on CQUEST. Use an `sftp` program to transfer files. For example, to print SAS output at home you'll need to transfer the files to your home computer. (I recommend the sftp programs WinSCP for MS Windows and Fugu for Mac OS X; both can be downloaded for free.)

Using SAS:
Batch Mode vs the SAS Windowing Environment

- The SAS Windowing Environment on CQUEST (Linux) is similar to the version of SAS I'll demonstrate in lecture (MS Windows).
- Alternatively you can use batch mode, creating your SAS program using the editor of your choice and running it at a command prompt.
- When accessing CQUEST remotely, the SAS Windowing Environment can be very slow so you may want to try batch mode.

A Picture of the SAS Windowing Environment on CQUEST



Starting the SAS Windowing Environment

- MS Windows in a CQUEST lab: Under *Start* choose *Statistics Apps* and then *SAS*.
- Linux machine in RW 211: Under *Applications* choose *CQUEST* and then *SAS* or type `sas` at a command prompt in a terminal window.
- When accessing CQUEST remotely: type `sas` at a command prompt.
- You will have to enter your CQUEST password.
- Double-clicking on a previously saved SAS program doesn't work. To open a previously saved program, open SAS and then choose `File > Open`.

Starting the SAS Windowing Environment continued

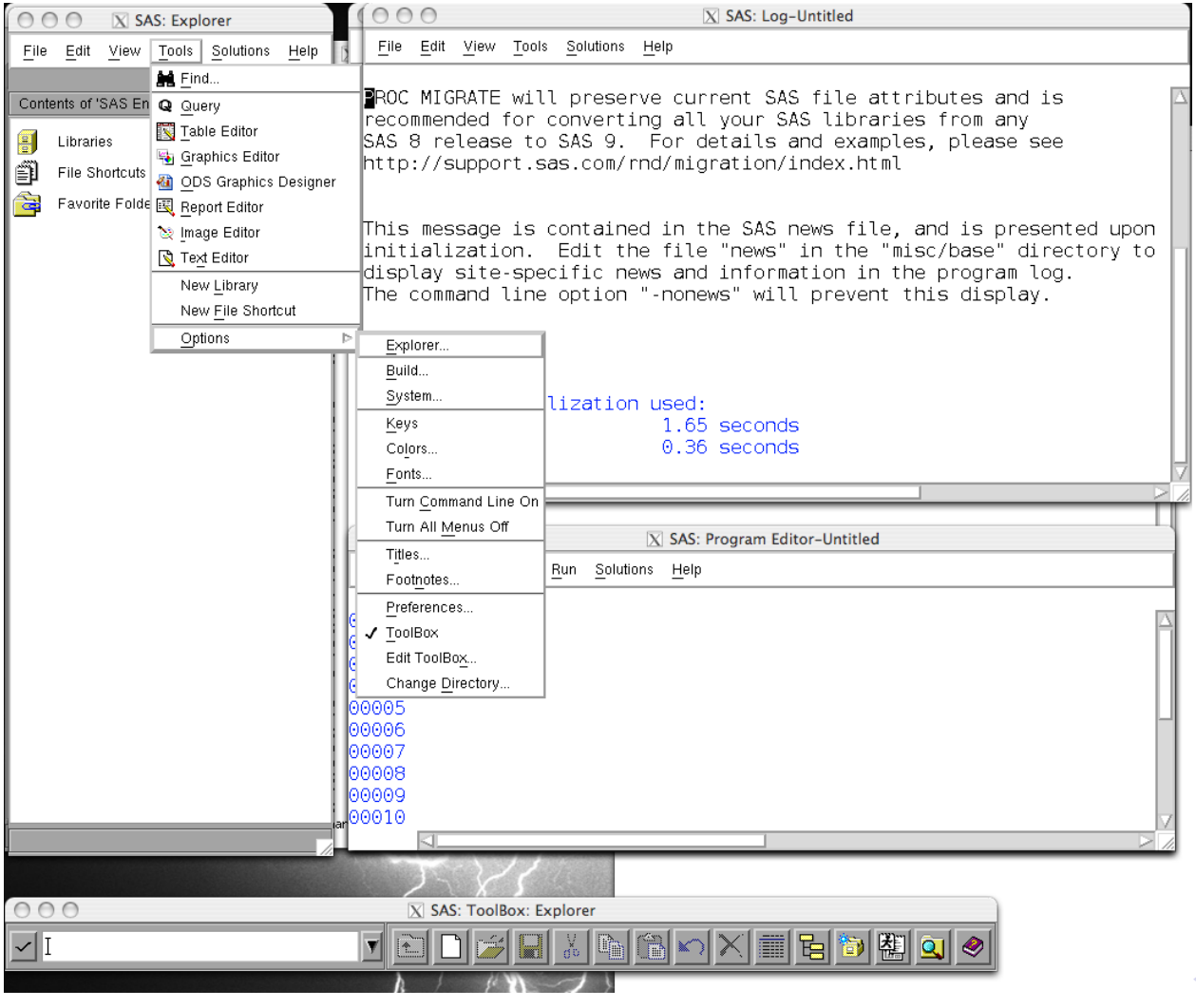
Many windows will open including:

- a program editor where you type your SAS program
- a log window
- an output window
- an explorer for navigating through your windows and output
- a graphics window will open once you've run a program that produces graphics plots (except if the plots are produced using the Output Delivery System (ODS))

IMPORTANT FIRST STEP

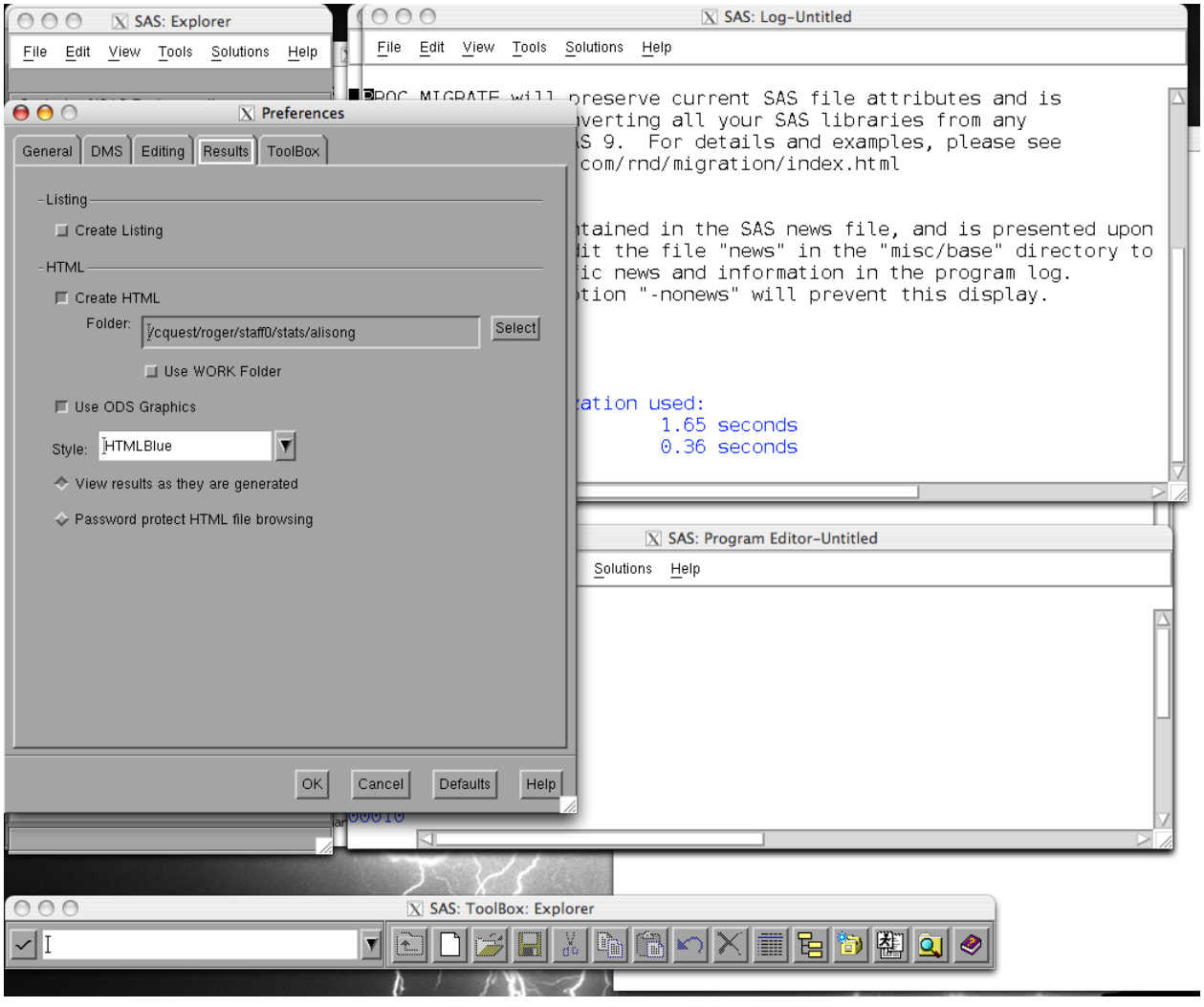
The version of SAS on CQUEST is 9.3. By default, this version of SAS uses ODS to save all graphs in graphics files (png) and output in an html file `sashtml.htm`. **TURN THIS OFF.** To do this:

`Tools > Options > Preferences...`



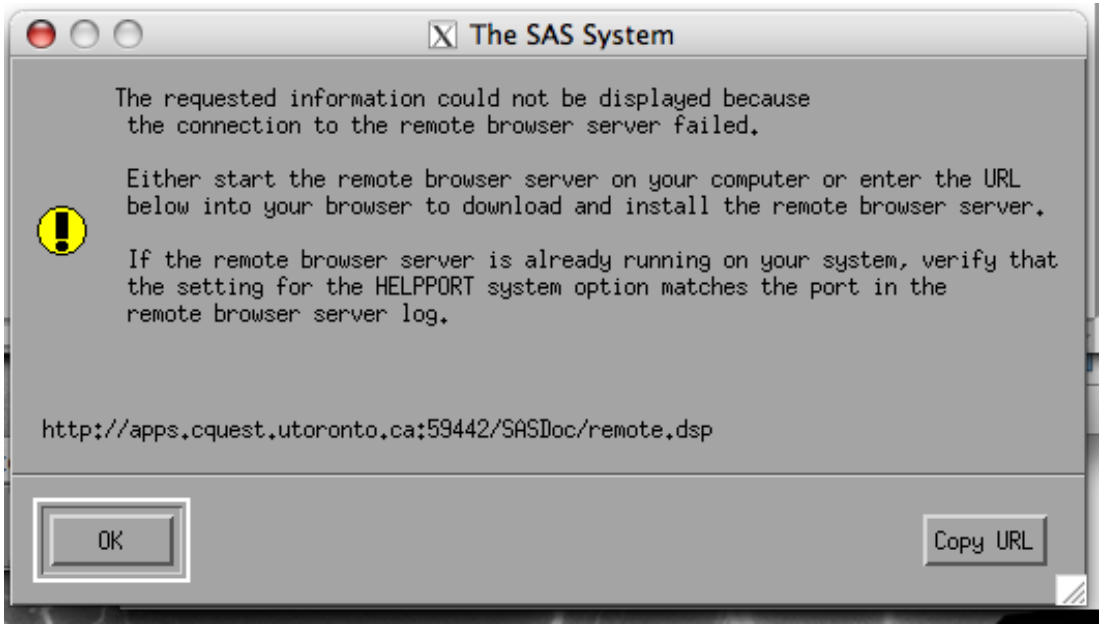
IMPORTANT FIRST STEP continued

Select: `Create Listing`



SAS Help on CQUEST

The SAS help is not completely installed on CQUEST. You may accidentally end up with this dialogue box.



If you do, everything will stop until you click `OK`.

More on the SAS Windowing Enviroment

- To insert a line after the current line in the SAS program editor, type an `i` in the leftmost column over the first zero and hit enter. (For more tricks you can use in the SAS program editor, google "SAS Text Editor Line Commands".)
- To run a program: Make the program editor window your primary window (click on it) and click on the running man or select `Run` and then `Submit`. Running a program writes output to a log window, an output window (if the program ran without errors) and sometimes a graphics window. Your code will also disappear. To get it back under `Run` select `Recall last submit`.
- Note that you need to save your program, your output, and your graphics files separately if you want to keep them.

Using SAS in Batch Mode

- Store your program in a (text) file called `filename.sas` and at a command prompt type `sas filename`. This creates `filename.log` and, if your run was successful, `filename.lst`.
- If you are using batch mode in an Xwindows terminal (Mac X11 or Xming), graphics will pop up in a new window on the screen. To save graphics plots (and then print them), there is some SAS code in a later slide (or use ODS).
- If any of your SAS batch jobs have a problem and keep running, you must kill them. Type `ps` at a prompt to get the number of the SAS job (the `PID`), and then type `kill -9 thePIDnumber` to kill a job.

SAS Log Window (or `.log` file)

Messages from SAS about your program written when you run it.

Always check for ERRORs.

Basics of SAS Programming

- Every line ends with a semi-colon.
- SAS is not case sensitive; `GIbbS` and `gibbs` are the same within a program.
- Every line ends with a semi-colon.
- Anything between /* and */ is a comment.
- You can also comment out a single line by starting with a * but then it must end with a semi-colon.
- A typical SAS program has data steps (to create and manage your datasets) and procedures starting with the reserved word `proc`.

More Basics of SAS Programming

- Use `run;` after every procedure.
- You can add titles to your output with the command
  `title 'This is my title';`
  at the beginning of your program or within any procedure.
- SAS has crazy ideas about line lengths. To fix this make
  `options linesize=79;`
  the first line of your program. (There are many other options you can set, but this is the only one that is essential.)
- Every line ends with a semi-colon.

The Data Step

The code below creates a dataset called `to1993`.

```
data to1993;
  infile 'maunaloadata.txt' firstobs=33;
  input year 1-4 month 6-7 day 9-10 time1 $ date $
          time2 $ cfc11 nd sd f cs rem;
  if cfc11 < -9999 then cfc11=.;
  time=year+(month-1)/12;
  drop day time1 date time2 nd sd f cs rem;
```

- The data are read from the file `maunaloadata.txt`.
- The data file is a text file, with a new line for each observation.
- Because the first 33 lines of this file contains information and not data, we need to specify `firstobs=33`.
- Because the dates are in the form 1993-01-01, we specified the columns of the data file that contain the year, month, and day.

The Data Step continued

- The rest of the variables in the datafile are delimited by spaces. (If they were delimited by commas, you would add `dlm=','` to the `infile` line.)
- A `$` after a variable name on the `input` line indicates that the variable is character (rather than numeric) data. (I did that here since the times and dates contain special characters, but I don't care about the information in the data file for `time`, `date`, and `time2`.)
- The missing data code in SAS is `.` (a period). Values of `cfc11` less than -9999 were replaced with the missing value code.
- The variables that were read in from the data file but that we will not be using were dropped from the dataset.
- A new variable, `time`, was created from `year` and `month`.

More on the Data Step

The code below creates a new dataset which starts with the dataset `to1993` but only includes observations for which the time is before 1990.

```
data preMP;
  set to1993;
  if time < 1990;
```

The following code concatenates the two datasets. The datalines from `after1994` are appended to the datalines from `to1993` in the new dataset `all`. If a variable exists in one of the datasets and not in the other, it is created and given missing values for the other dataset.

```
data all;
  set to1993 after1994;
```

Some SAS Procedures

By default any procedure operates on the last dataset you created. To change this, add `data=datasetname` on any `proc` line before the semi-colon.

- **gplot and plot**
  Graphics plots go into a graphics window (from which they can be printed or saved individually) or pop up in their own window if using batch mode in an Xwindow. For a `variable1` versus `variable2` scatterplot:

  ```
  proc gplot;
    plot variable1*variable2;
  ```

  For an ugly lineprinter plot that goes in the output window (or `.lst`) file:

  ```
  proc plot; plot variable1*variable2;
  ```

Some SAS Procedures continued

- `proc print;`
  Prints the contents of the last dataset. Useful to see if the data have been correctly read into SAS.
- `proc corr;`
  Prints summary statistics and pairwise correlations for all quantitative variables.
- `proc univariate plot;`
  `var variablename;`
  Gives many summary statistics and basic plots (or no plots if you leave out the word `plot`).
- The minimum commands for regression where `variable1` is the independent variable and `variable2` is the dependent variable:

```
proc reg;
  model variable2=variable1;
```

More on `proc reg`

There are many more commands you can add to all procedures, but we will focus on `proc reg`. More will arise in class as needed but here are some commonly used plotting commands:

```
proc reg simple;
  model y=x;
  plot y*x;  /* scatterplot with fitted line */
  plot r.*x;  /* plot of residuals vs x */
  plot r.*p.;  /* residuals vs predicted */
```

The keyword `simple` gives basic summary statistics for `y` and `x`. Variable names that end in a period are variables created by SAS. When ODS is on and no plot statements are given, `proc reg` produces some graphics files in your home directory (png format): `DiagnosticsPanel.png`, `ResidualPlot.png` (residuals versus explanatory variables), `FitPlot.png` (scatterplot, fitted line, and intervals around the line, for simple regression only).
(`proc glm` can also be used for regression but has fewer options.)

Saving Residuals and Predicted Values to a Dataset

(Useful if you want to do more analysis on the residuals.)

```
proc reg;
  model variable2=variable1;
  output out=outdata p=pred r=resid;
```

creates a new dataset called `outdata` that includes all the variables in the original dataset plus variables called `pred` (the predicted values) and `resid` (the residuals).

Saving Graphics Plots to a File

In the SAS Windowing Environment, choose `File` from the toolbar and then to print select `Print` or to save select `Export as Image`.

If you are working at home by using ssh to access CQUEST you will want to save graphics plots in files within the SAS program so that you can sftp them to your home computer or print them when you come to campus. The following commands save graphics plots in gif format to files in your home directory.

```
filename grafout '.';
goptions device=gif gsfname=grafout;
```

The files are named by the SAS procedure that created them. For example, if you have requested 3 plots within `proc reg` the plots will be stored in the files `reg.gif`, `reg1.gif`, and `reg2.gif`. Instead of gif you can specify png, jpeg, and other image formats.

Example: Smoking and Cancer

```
options ls=79;

/* Data within program rather than in separate file */
data smoking;
  input occupational_group $ smoking mortality ;
  datalines;
Farmers_foresters_fisherman          77       84
Miners_quarrymen                    137      116
Gas_coke_chemical_makers            117      123
Glass_ceramics_makers                94      128
;
/* Note that there is no semi-colon after each of the
data lines, but just one at the end.  (Also, to save
space for this example, I've left out most of the data.)
Also note that since occupational_group is character data
there is a $ after its name in the input statement. */

proc corr data=smoking;
```

Example: Smoking and Cancer continued

```
* Change plotting symbol to a black dot ;
symbol1 v='dot';

proc gplot;
  plot mortality*smoking;

proc reg data=smoking;
  model mortality=smoking;
  plot mortality*smoking;
  plot r.*smoking;
  plot r.*p.;
run;
```

A Few Unix Commands

These can be executed from a terminal window on the three Linux machines in RW 211 or from home when accessing CQUEST through ssh.

exit	Logs you off.
passwd	Change your password.
man command	Unix help for command.
ls	List the files in the current directory.
mkdir directoryname	Make a new directory.
cd directoryname	Change directory to directoryname.
cd	Change to your home directory.
less filename	Displays filename on your screen. (q to quit.)

A Few Unix Commands continued

`rm filename` Removes filename.

`cp filename1 filename2` Copies filename1 to filename2.

`mv filename1 filename2` Moves (renames) filename1 to filename2.

Editors: On the Linux machines in RW 211, try `gedit filename` (the default editor). (`filename` can be a new file.) (Gedit is available as `Text Editor` when you right-click on an existing file.) If you are using ssh, try `nano filename`. My favourite Unix editor is `vi` but you need to learn about it before you use it. If you google "vi unix editor" you'll find some good tutorials.