Myles J. Callan **Econ 490** **Summer 1999**

Gauss Guide

Lab Guideline for Econ 490 Students

---

1. How to Logon and start Gauss:

Step 1: Logon:

Press “Enter” key and fill-in the requested access account ID and passwords using both keyboard and mouse.

If you don't have a computer account at PSU yet, you need to contact computer center account office to get one.

Step 2: Starting Gauss:

Click the “Start” button (on the bottom left of the screen). From the list, click the icon of “Courseware”, then select the Gauss icon and, as usual, be patient to wait for the Gauss screen to show-up.

---

2. How to Run the Gauss Program.

Note: Gauss (like SAS) is mainly a code driven program. That is, to input data, and perform "Procedures" you write a “program” in text and Gauss implements all of the features in that program.)

Step 1: An overview of Gauss.

The first window that you see is the command window. You can operate in this window (for example if you type 2+2 – or any calculation at the “>>” prompt and hit “enter” key on the keyboard you will get the desire result) but it is not very useful for the type of projects that you will be concerned with.

To be truly effective in Gauss you write “programs”, just like in SAS. The “Gauss Language” is like any standard language in that it has its own “vocabulary”. Again just like SAS, a lot of the work is done for you by this language, that is the standard procedures that econometricians use are already coded into the program

you just need to know the right commands to use. You can write these programs within Gauss or you can use any text-editor to write Gauss programs (as long as you save the programs as “text only”).

Step 2: Creating a new SAS “program” or opening an already existing one.

To write a *new* program within the Gauss program type “Edit filename.G” (where “filename” is any name that you wish to call the program, you can use any “extension” you wish - the “.G” is just something I use to distinguish the Gauss programs that I write from other files). A new window will appear and this is where you will type in your “code”.

I prefer to write Gauss programs using a text editor (there are some useful text editing programs that allow you to specify certain repeatedly used words to be colored a color of your choice – for example Ultraedit. This feature may seem trivial but if you are typing thousand lines of code, it is very useful). In this case you simply open the text editor that you wish to use, type the code and save the file as text (again saving the file using any name that you wish).

Finally, a warning. When using SAS the “output window” highlighted any errors in the program that you wrote by writing in red where the perceived error occurred. This error detection is called the “debugging” routine of the program. Gauss does NOT have a very good debugging routine. It simply stops at the first error that it detects and tells you the line number associated with that error. A very similar program to Gauss, called Matlab, has an excellent debugging routine (also, it has the added feature that it will transform *your* program into C+ so that you can create “stand alone applications” (i.e. applications that run without any assistance from any other program) without ever having to learn a true computer language such as C+). So if think that you may be using these types of econometrics programs, I would advise you to opt for Matlab (even though most of the econometric world uses Gauss – which is a major advantage as you can find and use other peoples programs easily).

As with SAS, the easiest way to learn how to use Gauss is to use a pre-written program. I have supplied such a program on my web-page (called Econ_490.G) – it also at the end of this document. If you want to alter this program in any way (within Gauss) type “edit a:\Econ_490.G at the “>>” prompt – if you have the program on a disk (i.e. the “a – drive”). You can also edit the program in a text editor.

Step 3: To Run A Program.

To run a program in Gauss, simply type “Run Filename.G” at the “>>” prompt. For example, type “Run Econ_490.G” to run the program provided on the web-page.

3. A Sample Program

```
/*****************************************************************************/
/*****************************************************************************/
/*    - Econ_490.G - MYLES CALLAN - 20th of July, 1999 -                     */
/*                                                                           */
/*     Get into the habit of naming and dating your programs!!!              */
/*                                                                           */
/*     The Gauss version of the SAS program introduced earlier               */
/*                                                                           */
/* This program "loads" the mon1.dat data set and performs some              */
/* simple analysis of this data.                                             */
/*****************************************************************************/
/*****************************************************************************/

/*****************************************************************************/
/*   Naming the output file, where the results will be stored                */
/*  (and saving them to the a-drive)                                         */
/*****************************************************************************/

Output file = a:\Econ_490.OUT reset; /* "reset" tells Gauss to overwrite */
                                     /*  previous versions of the output file */

format /a1 /lo 8,6;              /* These commands tell Gauss how you want the */
outwidth 256;                    /* output to look                              */

/*****************************************************************************/
/*     The next command tells Gauss to start writing to the "output file"    */
/*****************************************************************************/

output on;

/*****************************************************************************/
/*   Loading the data to be used by this program                             */
/*****************************************************************************/

load data_1[426,5]=a:\Mon1.dat;  /* Load the data (from the "a" drive), which has 426 rows and 5 columns */
                                 /* and name this data "data_1" for use within this program */

/*****************************************************************************/
/*                                                                           */
/* Transform variables into logs and log differences, creating new variables  */
/* (note: you didn't have to name the variables when you loaded the data      */
/*  as you did in SAS although you can - you simply use                      */
/*  column numbers associated with each variable. So that CPI will be:       */
/*  data_1[.,5] in the example below, i.e. the fifth column in the dataset   */
/*****************************************************************************/

lm1     =       ln(data_1[.,1]);              /* The natural log of M1, 1st column of data_1 */
dlm1    =       lm1[1:rows(lm1),.]-lag(lm1); /* The first difference of log M1 */
lm2     =       ln(data_1[.,2]);
dlm2    =       lm2[1:rows(lm2),.]-lag(lm2);
lip     =       ln(data_1[.,4]);
dlip    =       lip[1:rows(lip),.]-lag(lip);

d0_m1   =       data_1[1:rows(data_1),1]-lag(data_1[.,1]);
d1_m1   =       lag(d0_m1);
d0_m2   =       data_1[1:rows(data_1),2]-lag(data_1[.,2]);
d0_Ip   =       data_1[1:rows(data_1),4]-lag(data_1[.,4]);
d0_pr   =       data_1[1:rows(data_1),5]-lag(data_1[.,5]);

/*****************************************************************************/
/*     compute (x) means (y) variance and (z) min of all the variables       */
/*****************************************************************************/

x = meanc(data_1);

y = vcx(data_1);

z = minc(data_1);

/*****************************************************************************/
/*          compute correlation coefficients (cor) for all variables         */
/*****************************************************************************/
```

```
cor = corrx(data_1);

/*****************************************************************************/
/*    run a regression, note that intercept is automatically included        */
/*    the output of this regression will include t-statistics                */
/*    coefficient estimates, etc.                                            */
/*    NOTE: you can name each element of the output to be used later in the  */
/*    program, this is what is done in the "{" parenthesis.                  */
/*    NOTE 2: this next command will print the output automatically to the   */
/*    output file (this is not standard as you will see later)               */
/*****************************************************************************/

{vnam,m,b,stb,vc,stderr,sigma,cx,rsq,resid,dwsat}=ols(0,dlm1,dlm2);

/*****************************************************************************/
/*     The next commands tells Gauss WHAT TO write to the "output file"      */
/*     you do this by simply typing the variable names                       */
/*****************************************************************************/

"";"";"Means of the Variables";
x;
"";"";"Variance Covariance matrix of the Variables";
y;
"";"";"Minimum of each of the Variables";
z;
"";"";"Correlation matrix of the Variables";
cor;

/*****************************************************************************/
/*     Tell Gauss that you have finished writing to the output file          */
/*****************************************************************************/

output off;
```