

A Brief Introduction to GAUSS Light

GAUSS Light is the student version of the matrix programming language GAUSS (product of Aptech Systems Inc.). GAUSS Light is fully compatible with GAUSS within certain matrix size and workspace limitations.

1 Installation

1. Go to Windows Explorer (or File Manager) and locate the file `GAUSS_3.6_Win_light.zip`.
2. Double click on `GAUSS_3.6_Win_light.zip` to unzip the file (this requires a version of WinZip).
3. Double click on `setup.exe` and follow the instructions on the screen (use the default options).
4. Create (using Windows Explorer or File Manager) a subdirectory `c:\gausslt36\work` and copy the files `ols.prc`, `ols_ex.txt` and `wage.dat` in this directory.
5. Unzip the file `GAUSS_3.6_Manual.zip` and extract the GAUSS manuals to your hard disk.

2 How to start and exit GAUSS

1. Double click on the icon **GAUSS Light 3.6** on your Desktop. The GAUSS prompt `>>` will appear on your screen. You can use GAUSS either in "interactive" or "batch" mode. In this course, we will use only the "batch" mode. This means that we first write and save a sequence of GAUSS commands in a program (for example, `ols_ex.txt`) using the text editor that comes with GAUSS Light or an external text editor (such as Notepad). Then, we can execute the program by typing `run ols_ex.txt` in the command line and pressing the **Enter** key. Before executing the program, one should go to **File/Change Working Directory** locate the directory (for example, `c:\gausslt36\work`) that contains the program and the other supplementary and data files.
2. To exit GAUSS, go to **File** in the main menu and then choose **Exit**.

3 How to create and edit GAUSS programs

From the GAUSS menu go to **File/Open** and select the file to be edited (alternatively, you can click on the second icon at the top left corner). To create a new program go to **File/New** or click on the first icon at the top left corner. After the changes in the file have been saved (use **File/Save** or the third icon at the top left corner), the file can be executed by clicking on the icon "Run current file" in the top middle part of the screen. The GAUSS editor can be used also to view the GAUSS output files.

4 Some useful GAUSS commands and operations

The most commonly used GAUSS commands and operations are the following:

<code>x=rndn(n,k);</code>	Creates an $n \times k$ matrix x of iid $N(0,1)$ random numbers
<code>x=2+sqrt(10)*rndn(n,1);</code>	Creates an $n \times 1$ vector of iid $N(2,10)$ random numbers
<code>x=rndn(n,2)*chol((1~.5) (.5~1));</code>	$n \times 2$ matrix of correlated (corr.=.5) $N(0,1)$ random numbers
<code>x=rndu(n,k);</code>	Creates an $n \times k$ matrix of iid Uniform $[0,1]$ random numbers
<code>load x[n,k]=data.dat;</code>	Loads the ASCII file "data.dat" into the $n \times k$ matrix x
<code>xi=invpd(x'x);</code>	Inverts the positive definite matrix $X'X$
<code>z=x.*y;</code>	Element by element multiplication of matrices x and y
<code>z=x./y;</code>	Element by element division of matrices x and y
<code>+, -, *</code>	Addition, subtraction and multiplication of matrices
<code>+, -, *, /</code>	Addition, subtraction, multiplication and division of scalars
<code>y=meanc(x);</code>	Computes the mean of x
<code>y=stdc(x);</code>	Computes the standard deviation of x
<code>y=sqrt(x);</code>	Takes the square root of all elements of x
<code>y=diag(x);</code>	Creates a vector from the diagonal elements of matrix x
<code>y=sortc(x,1);</code>	Sorts the vector x in ascending order
<code>x=y~z;</code>	Horizontally concatenates matrices y and z
<code>x=y z;</code>	Vertically concatenates matrices y and z
<code>x=ones(n,1);</code>	Creates an $n \times 1$ vector of ones
<code>x=zeros(n,1);</code>	Creates an $n \times 1$ vector of zeros
<code>x=eye(k);</code>	Creates an $k \times k$ identity matrix
<code>x=seqa(1,1,100);</code>	Creates a sequence of numbers 1, 2, 3, ..., 99, 100
<code>x;</code>	Prints matrix x on the screen (and in the output file)
<code>x=ones(100,1);</code>	This "do-loop" stores the numbers from 1 to 100 in matrix x
<code>i=1; do while i<=100;</code>	
<code>x[i]=i;</code>	
<code>i=i+1; endo;</code>	

5 Sample GAUSS program

@ The execution of this program requires the file `ols.prc` supplied separately @

```
new;
output file=ols_ex.out reset;
#include ols.prc;
load data[526,4]=wage.dat;
y=ln(data[.,1]);
x=data[.,2:4];
{beta_hat,se_hat,u_hat}=ols(y,x);
call stats(y);
output off;
end;
```

6 How to read and write a GAUSS program (some basic rules)

As an example, we will use the sample program provided in Section 4.

1. All GAUSS statements end with semi-colon ;.
2. The first line of the program is a comment. Comments may be added either as `@ @` or `/* */`.
3. It is useful to start the program with the command `new` which will erase everything stored in the memory from previous runs.
4. The command `output file=ols_ex.out reset` gives the name of the output file that will store the estimation results and overwrites the previous version with the same name (if there is any). To append to a previous version of the output file, the command `output file=ols_ex.out on` should be used instead.
5. To read in data from the ASCII (plain text) data file `wage.dat`, we use the command `load data[526,4]=wage.dat` which creates a matrix with name `data` (any name can be used) with 526 rows and 4 columns.
6. Then we define `y` (later used as a dependent variable) to be the natural logarithm of the first column of `data` and `x` to be a matrix (whose columns are later used as explanatory variables) that consists of column 2, 3 and 4 of `data`. The dot in the first entry of the square brackets is used to denote that all the rows of the matrix `data` should be used. If we want to use only the first 200 observations, we should replace the corresponding lines by `y=ln(data[1:200,1])` and `x=data[1:200,2:4]`.
7. The command `#include ols.prc` indicates that we will use the external program `ols.prc`. The program `ols.prc` contains two GAUSS procedures: `ols` and `stats`. The procedure `ols` contains the code for OLS estimation and is called with the command `{beta_hat,se_hat,u_hat}=ols(y,x)`, where the variables in the curly brackets are the output variables (in this case: the parameter estimates, their standard errors and the residuals, respectively) and the variables in the round brackets on the right-hand side are the input variables (in this case: the dependent variable and the matrix of explanatory variables). After the execution of the program, you can print on the screen any of the output or the input variables by typing its name, for example `beta_hat`. The procedure `stats` does not return any output variables (just prints the output on the screen) and is invoked with the command `call`. The GAUSS procedures are a very convenient tool in executing computationally large and repetitive tasks and we will be using them quite often. For some guidelines on how to write a GAUSS procedure, see the GAUSS manual.
8. The last two command lines of the program, `output off` and `end`, are used to close the output file and terminate the program, respectively.

A more detailed introduction to GAUSS can be found in the on-line tutorial that can be accessed from the course webpage and the documentation accompanying `GAUSS Light 3.6`.