Introduction to GAUSS (II)

Hiu Man Chan

Econ551b, Spring 1999

I. Saving and Loading Data

1. Writing to an Output file

```
/* Write the following as a batch file */
(gauss)output file=session.out reset;@name the output file as ”session.out”. By ”reset”, we write over the existing file ”session.out”, if it exists. If we want to append to the old file, use ”on” instead of ”reset”.@
(gauss)”Printing line 1”;
(gauss)output off;@output won’t be printed to the output file from now on@
(gauss)”Printing line 2”;@so you won’t see this line in the output file.@
(gauss)output on;@output is printed to output file again@
(gauss)”Printing line 3”;
(gauss)output off;
```

2. Saving and loading a matrix

```
/* Saving a matrix */
(gauss)new;@start by clearing memory@
(gauss)closeall;@start also by making sure all files are closed@
(gauss)x={1 2 3, 4 5 6};@define a variable x@
(gauss)save path=c:\temp x;@save x at c:\temp, gives x.fmt@

/* Loading a matrix */
(gauss)new;@clear memory@
(gauss)x; @now the variable is not in memory@
(gauss)load anyname=c:\temp\x;@load variable x and call it anyname@
(gauss)anyname;
```

3. Loading data from an ascii data set

```
(gauss)new;
(gauss)load x[]=c:\temp\hrsdat.asc;@load data as a column vector@
(gauss)rows(x);
(gauss)rows(x’);@number of columns should be 1@
(gauss)x=reshape(x,rows(x)/9,9);@reshape x into a new variable, also called x, which has rows(x)/9 rows and 9 columns@
(gauss)rows(x);
(gauss)rows(x’);
```

/* If we know the dimension a priori:*/
(gauss)new;
(gauss)load x[6851,9]=c:\temp\hrsdat.asc;@load data as a 6851x9 matrix@
(gauss)sex=x[.,1];@define the first column as a variable called sex@
(gauss)sex[1:10];@gives you an idea how sex looks@
(gauss)unique(sex,1);@prints out uniqu values in sex; "1" means sex is numeric. Put in "0" instead of "1" if it is character data@

4. Saving and loading GAUSS data set

/* Let us first create a GAUSS data set using x we loaded from ascii file */
(gauss)let vnames=sex age race mstat voctrn ba educ earn hwkd; @gauss data set differs from ascii file as it also contains names of variables. This line specifies the names of the variables and store it as the variable "vnames"@
(gauss)outname="hrsdat";@This specifies the name of data set we want to write to, and stored as a variable called "outname"@
(gauss)create outfile=^outname with ^vnames,0,4;@This creates and open the data file with file name as specified in outname, and with variable names as specified in outname. The "0" position specifies the numer of columns in the data set. When "0" is used, number of columns will be determined by number of elements in vnames. The "4" specifies the precision of storing the data. It can be "2", "4" or "8". "2" is for storing integers, while "4" will give precision in the order of e-37 to e+38, and "8" will be precise for anything from e-307 to e+308. In general, "4" should be used@
Note: outfile is a file handler. It is a scaler, a number that gauss assigns so that it can uniquely refer to the file specified under the file handler.
(gauss)nrw=writer(outfile,x);@writer writes the matrix x into outfile, and returns the number of rows written. By putting "nrw" on the left hand side of "=", we are storing the return, i.e., number of rows written, to a variable called "nrw"@
(gauss)nrw;;"rows written";@one way of using the variable "nrw"@

Note: if you do the above in unix, one single file hrsdat.dat will be created. But if you do it in windows, two files will be created, one is hrsdat.dat that contains only the matrix x, and the other is hrsdat.dht that contains a 9x1 vector of variable names.

/* Now try to read from the GAUSS data set we've just created */
(gauss)new;
(gauss)closeall;
(gauss)infile="hrsdat";@creating a variable that stores the name of input file@
(gauss)open inf_h=^infile;@opens the file stored under the variable infile. inf_h is again a file handler, a pointer pointing to the file named under infile.@
(gauss)infnms=getname(infile);@"getname" command gets the names of the variables and store store the names in infnms@
(gauss)$infnms;@Put $ in front of the variable when you want to print out a character@
(gauss)x=readr(inf_h,rowsf(inf_h));@"rowsf" gives the total number of rows in the file pointed by the file handler inf_h. "readr" then reads in rowsf(inf_h) rows of data from the file pointed to by inf_h@
(gauss)inf_h=close(inf_h);@"close" closes the file pointed by inf_h, and returns "0" if the closing is successful, "-1" otherwise. The return (0 or -1) is stored as inf_h. The file

```
handler is set to zero when the file is closed.@
(gauss)earn=x[.,loc(”EARN”,infnms)];@”loc” is a procedure written by Prof. John Rust
that returns the position of ”EARN” in the vector infnms. Using loc, we can identify
the column position of the variable, and extract it accordingly.@
(gauss)sex=x[.,loc(”SEX”,infnms)];

/* Alternatively, can read in by blocks when data set is very large */
/* Type in the following lines as a batch file */
(gauss)open inf_h=^infile;
(gauss)nr=500;
(gauss)earn2=0;@initialize variable@
(gauss)do until eof(inf_h);
(gauss)      x=readr(inf_h,nr);
(gauss)      earn2=earn2|x[.,loc(”EARN”,infnms)];
(gauss)endo;
(gauss)earn2=earn2[2:rows(earn2)];@the first element is just for initialization@
(gauss)inf_h=close(inf_h);
/* End of batch file – now can run the batch file */
earn==earn2;
```

II. Missing Operator

```
(gauss)sex[1:10];
(gauss)sex2=miss(sex,1);@set everything that is ”1” in variable sex to missing, and store the
new variable as sex2@
(gauss)sex2[1:10];
(gauss)sex3=packr(sex2);@pack out all rows with missing values@
(gauss)sex3[1:10];
(gauss)sex4=missrv(sex2,-9999);@substitue the missing values with -9999, and store the new
variable as sex4@
(gauss)sex4[1:10];
(gauss)tmp=sex2.*earn;@operations involving any missing value will return a missing
(gauss)sex2[1:10]~earn[1:10]~tmp[1:10];
```

III. Selecting or deleting observations

```
(gauss)sex5=selif(sex,sex.==2);@select out elements in sex where the corresponding element
in (sex.==2) is ”1”@
(gauss)sex5[1:10];
(gauss)sex3==sex5;
(gauss)earn5=selif(earn[1:10],sex[1:10].==2);
(gauss)sex[1:10]~earn[1:10];
(gauss)earn5;
(gauss)sex6=delif(sex,sex.==2);@delete elements in sex where the corresponding element in
(sex.==2) is ”1”@
(gauss)sex6[1:10];
```