

An Introduction to Using SAS on PCs

Dave Ribar

- 1) SAS is a system of software packages; some of its basic functions and uses are
 - database management – inputting, cleaning and manipulating data
 - statistical analysis – calculating simple statistics such as means, variances, correlations; running standard routines such as regressions
 - graphics – producing two- and three-dimensional plots and charts; producing maps
 - statistical programming – SAS allows you to program your own statistical routines
- 2) execution
 - after logging onto PC, invoke SAS from a desktop short-cut, from the start menu, or from a class menu
 - SAS will open five windows
 - Program editor – SAS commands are entered here; you can create and submit a new set of commands or open, edit, and submit an existing file with commands
 - Log – this window describes the commands that have been run; it also describes the resources that were used and the errors that were encountered while SAS processed your commands
 - Output – this window lists the statistical results or other program output, if any, from the commands; output will only be generated if your program ran without errors
 - Explorer – this window helps you work directly with SAS files
 - Results – this window helps you work directly with different types of SAS output
 - you can toggle between the different windows either by clicking each window's minimization and resize icons (upper right corner of the window area), clicking on the window itself, or using the **Window** menu from the top menu bar
 - operating sequence – typically when you use SAS, you will follow this sequence
 - enter program commands by either typing the commands in the Program Editor or by opening an existing program (.sas) file
 - submitting the commands using either the submit icon on the tool bar or going to the menu bar and selecting **Run • Submit**
 - reviewing the Log window for errors
 - reviewing results in the Output window
 - when the commands come from or are stored in a file, we refer to this sequence as a “batch” sequence (you submit all of your commands in a batch instead of one at a time)
 - after the program has run
 - if you want to keep a permanent copy of the material in any of the windows, open (click on) the window, go to the menu bar, and select **File • Save** or **File • Save As**
 - ◊ it's a good idea to keep copies of your programs; data processing for statistical analysis usually involves many commands; these will need to be repeated if you want to recreate or modify the data set; saving a copy of the program keeps you from having to retype these commands each time you want to correct or modify your processing steps

the TITLE command causes <title_information> to be printed at the top of each page of output; this is a great way to document your output; TITLE# changes the information for the #th title line

- RUN command

```
RUN;
```

when you submit a set of commands to be executed, SAS will execute all of the commands in every complete procedure; one way to end a procedure is to invoke a new one; another way is to put a RUN command at the end of the procedure; placing a RUN command at the end of your program ensures that SAS will run all of the commands

5) DATA step

- this is a special type of procedure that is used to input, manipulate, and create data
- if data are input or read into a DATA step, the observations are processed sequentially and the DATA step effectively loops over the observations
- DATA statement

```
DATA <sasdata>;
```

the DATA statement is the first statement in a DATA step; it creates a SAS data file called <sasdata>

- SET statement

```
SET <sasdata>;
```

the SET statement inputs/reads data from an existing SAS data file called <sasdata>

- INFILE statement

```
INFILE <fileref>;
```

the INFILE statement is used to begin reading an external (non-SAS) file <fileref>, where <fileref> has been defined using the FILENAME command; INFILE identifies the file to be read; you can skip the FILENAME and <fileref> and simply type

```
INFILE '<Windows_file_name>;'
```

- INPUT statement

```
INPUT <variable1> <variable2> ...;
```

the INPUT statement reads the data from the external file specified by the INFILE command into the variables <variable1>, <variable2>, etc.; as written, this

statement assumes that the INFILE data are all numeric with at least one blank space between the variables; alternative formatting can be specified
note: for non-SAS files you must use an INPUT statement and explicitly describe the data layout; for SAS files, this is handled automatically by the SET command

- OUTFILE statement

```
OUTFILE <fileref>;
```

the OUTFILE statement is used to begin writing to an external (non-SAS) file <fileref>, where <fileref> has been defined using the FILENAME command; OUTFILE identifies the file to be written to; as with the INFILE, you can skip the FILENAME and <fileref> and simply type

```
OUTFILE '<Windows_file_name>;'
```

- PUT statement

```
PUT <variable1> <variable2> ...;
```

the PUT statement writes the data from the variables <variable1>, <variable2>, etc. to the external file specified by the OUTFILE command; if no OUTFILE command is specified, the data are written to the SAS Log

- assignment statement

```
<variable1> = <math_expression>;
```

the assignment statement places the value from <math_expression> into <variable1>; a math expression consists of a numeric (or character) constant, a variable, or a formula involving constants and variables

mathematical operators: +, -, /, *, **

- IF THEN ELSE statements

```
IF <condition1> THEN  
    <command1>;  
ELSE IF <condition2> THEN  
    <command2>;
```

these statements are used to call commands conditionally; the logical condition usually tests a variable against another variable or constant; commands are executed if the condition is true

the ELSE clause is optional; if ELSE is used without the IF <condition2> THEN, <command2> is executed anytime that <condition1> is not true

logical operators: =, <, >, <=, >=, ^=, AND, OR (note: the symbols can be replaced by EQ, LT, GT, LE, GE, NE; this sometimes improves the readability of programs)

if you want to execute multiple commands for the same condition, you need to use a DO END construct as follows

```
IF <condition1> THEN DO;  
    <command1>;  
    <command2>;  
    ...  
    <commandn>;  
END;
```

- subsetting IF statement

```
IF <condition>;
```

this tells SAS to keep only the observations that satisfy <condition>; this statement is useful for cleaning data and removing out-of-range observations; it is included here because its syntax is similar to an IF THEN statement but it produces very different results

- KEEP and DROP statements

```
KEEP <variable1> <variable2> ...;
```

```
DROP <variable1> <variable2> ...;
```

the KEEP statement keeps the listed variables in the SAS data file that is created by the DATA step; the DROP statement drops the listed variables from the data file; these statements are useful in paring down the “width” (number of variables) of a file

- the DATA step is terminated by a RUN statement, the start of another DATA step, or by a procedure

6) some utility procedures

- CONTENTS procedure

```
PROC CONTENTS DATA = <sasdata>;
```

- this procedure lists the variables in and other characteristics of the SAS data file <sasdata>
- the procedure is useful for examining unfamiliar external SAS datasets and for checking the structure of datasets that you have created

- SORT procedure

```
PROC SORT;  
    BY <variable1>;
```

- this procedure sorts the preceding SAS data set in order of <variable1>
- for large files, this procedure can be very time consuming

7) procedures to generate simple statistics

- MEANS procedure

```
PROC MEANS;
```

- despite the name this procedure actually calculates several statistics—means, standard deviations, minimum and maximum values, and counts of non-missing values
- unless you specify otherwise, PROC MEANS produces statistics for all of the variables in the preceding dataset; if you want to produce statistics for just a subset of variables, add the following statement:

```
VAR <variable1> <variable2> ...;
```

- UNIVARIATE procedure

```
PROC UNIVARIATE;
```

```
VAR <variable1> <variable2> ...;
```

- this procedure generates a very detailed set of univariate statistics; the statistics fill up an entire page of output
- because of the large amount of output, you should use a VAR statement to limit the number of variables

- CORRelation procedure

```
PROC CORR;
```

```
VAR <variable1> <variable2> ...;
```

- the procedure calculates correlation coefficients for pairs of variables from the preceding SAS dataset
- it also calculates simple statistics
- again, because of the extensive amount of output, the use of a VAR statement is recommended

- Frequency procedure

```
PROC FREQ;
```

```
TABLE <variable1>;
```

- this procedure reports the numbers of observations with different values of the variable listed in the TABLE statement; it is very valuable for checking the values of discrete variables
- the procedure can also be used to calculate cross-tabulations using the following TABLE statement

```
TABLE <variable1> * <variable2>;
```

8) running regressions, the REG procedure

- the syntax of the REG procedure is

```
PROC REG;  
MODEL <dependent_variable> = <list_of_independent_variables>;
```

- the <dependent_variable> is the name of the SAS variable corresponding to the outcome of interest
- the list of independent variables contains the name(s) of the SAS variable(s) corresponding to the explanatory variable(s)
- if more than one explanatory variable is used, the REG procedure performs multiple regression
- unless asked to do otherwise, the REG procedure automatically includes an intercept term (including an intercept is the default and SAS does not need to be told to include one)
- the REG procedure outputs estimated coefficients, coefficient standard errors, as well as test statistics and p -values (under the null hypothesis that the true coefficient is zero) for the intercept and independent variables
- the REG procedure also calculates the residual sum of squares, the R^2 and adjusted R^2 statistics, the mean square error, F -test statistics and other statistics

9) additional reading and information

- SAS has on-line help and tutorials
- a variety of documentation exists including Language References and User Guides for different SAS components