

SAS Syntax Programming

1. SAS PROGRAM STEPS	1
2. SAS STATEMENTS	1
2.1 THE THREE MOST COMMON ERRORS	2
3. SAS PROCEDURES	2
3.1. DESCRIPTIVE STATISTICS	2
3.2. LINEAR MODELS	3
3.3. PLOTS AND CHARTS	4
3.4. UTILITY PROCEDURES.....	4

1. SAS program steps

All SAS programs consist of a sequence of "steps". There are only two kinds of steps:

DATA step

A DATA step creates a SAS dataset (a collection of data together with a "data dictionary", which defines the variables and their properties). Data must be in the form of a SAS dataset before it can be analyzed by SAS procedures.

In the example SAS program, these lines create the dataset CLASS from raw data input:

```
DATA CLASS;  
  INPUT NAME $ SEX $ AGE HEIGHT WEIGHT;  
  CARDS;  
JOHN   M 12 59.0 99.5  
JAMES   M 12 57.3 83.0  
... (more data lines)
```

PROC step

A PROCedure step calls a SAS procedure to analyse or process a SAS dataset.

In the example SAS program, these lines call two SAS procedures to analyze the CLASS dataset:

```
PROC PRINT;  
PROC MEANS;  
  VARIABLES HEIGHT WEIGHT;
```

A SAS program can contain any number of DATA and PROC steps. The SAS statements in each step are executed all together. Once a dataset has been created, it can be processed by any subsequent DATA or PROC step.

2. SAS statements

- SAS programs are composed of SAS statements. All SAS statements start with a keyword (DATA, INPUT, PROC, etc.).
- All SAS statements end with a semicolon (;)
- Statements preceded by an asterisk (*) are comments.
- Uppercase and lowercase are equivalent, except inside quote marks (sex = 'm'; is not the same as sex ='M';).
- SAS variable names can be 1-8 characters but must begin with a letter.
- All SAS programs consist of 3 parts-a beginning, a middle and an end.

- SAS statements can be entered in free-format : You can begin in any column, type several statements on one line or split a single statement over several lines (as long as no word is split.).
- The structure of SAS program
 1. The Beginning is to create a SAS data set containing data.
 2. The Middle is to work with the data using SAS PROC'S, or procedures.
 3. The End is to RUN the program.

2.1 The three most common errors

1. Leaving off a semi-colon: this prevents SAS from understanding the command. SAS thinks that two statements are just one. That is the first place to look when you do not get all the output you expect. Syntax highlights of SAS 8.0 are very useful avoiding this error. Make sure your are using enhanced editor.
2. Misspelling: must correctly type the SAS commands and variable names.
3. Omitting one quote (') in a title or infile statement: this is very fatal SAS interprets everything between quotes as a character string. It will not recognize any more statements as commands. SAVE your program, exit SAS and begin a new session. You will be amazed by the variety of errors you commit.

3. SAS Procedures

SAS Procedures exist to carry out all the forms of statistical analysis. As the above examples indicate, a procedure is invoked in a "PROC step" which starts with the keyword PROC, such as:

```
PROC MEANS DATA=CLASS;
  VAR HEIGHT WEIGHT;
```

The VAR or VARIABLES statement can be used with all procedures to indicate which variables are to be analyzed. If this statement is omitted, the default is to include all variables of the appropriate type (character or numeric) for the given analysis.

Some other statements that can be used with most SAS procedure steps are:

BY variable(s);

Causes the procedure to be repeated automatically for each different value of the named variable(s). The data set must first be sorted by those variables.

ID variable(s);

Give the name of a variable to be used as an observation IDentifier.

LABEL var='label';

Assign a descriptive label to a variable.

WHERE (expression);

Select only those observations for which the expression is true.

For example, the following lines produce separate means for males and females, with the variable SEX labelled 'Gender'. (An ID statement is not appropriate, because PROC MEANS produces only summary output.)

```
PROC SORT DATA=CLASS;
  BY SEX;
PROC MEANS DATA=CLASS;
  VAR HEIGHT WEIGHT;
  BY SEX;
  LABEL SEX='Gender';
```

If the DATA= option is not used, SAS procedures process the most recently created dataset. In the brief summaries below, the required portions of a PROC step are shown in bold. Only a few representative options are shown.

3.1. Descriptive statistics

```
PROC CORR
```

Correlations among a set of variables.
PROC CORR DATA=SASdataset options;
 options:NOMISS ALPHA
 VAR variable(s);
 WITH variable(s);

PROC FREQ
 Frequency tables, chi ² tests

PROC FREQ DATA=SASdataset;
 TABLES variable(s) / options;
 options:NOCOL NOROW NOPERCENT
 OUTPUT OUT=SASdataset;

PROC MEANS
 Means, standard deviations, and a host of other univariate statistics for a set of variables.

PROC MEANS DATA=SASdataset options;
 options:N MEAN STD MIN MAX SUM VAR CSS USS
 VAR variable(s);
 BY variable(s);
 OUTPUT OUT=SASdataset keyword=variablename ... ;

Statistical options on the PROC MEANS statement determine which statistics are printed. The (optional) OUTPUT statement is used to create a SAS dataset containing the values of these statistics.

PROC UNIVARIATE
 Univariate statistics and displays for a set of variables.

PROC UNIVARIATE DATA=SASdataset options;
 options:PLOT
 VAR variable(s);
 BY variable(s);
 OUTPUT OUT=SASdataset keyword=variablename ... ;

3.2. Linear models

SAS statements and options for regression (PROC REG) are described in more detail in the document PROC REG Summary.

SAS statements and options for analysis of variance (PROC ANOVA and PROC GLM) described in the document PROC

ANOVA and PROC GLM.

PROC ANOVA
 Analysis of variance (balanced designs)
 PROC ANOVA DATA=SASdataset options;
 CLASS variable(s);
 MODEL dependent(s)= effect(s);

PROC GLM
 General linear models, including ANOVA, regression and analysis of covariance models.

PROC GLM DATA=SASdataset options;
 CLASS variable(s);
 MODEL dependent(s)= effect(s);
 OUTPUT OUT=SASdataset keyword=variablename ... ;

PROC REG
 Regression analysis

```
PROC REG DATA=SASdataset options;  
  MODEL dependent(s) = regressors  
  / options;  
  PLOT variable | keyword. *  
    variable | keyword. = symbol ;  
  OUTPUT OUT=SASdataset P=name R=name ... ;
```

1.1 3.3. Plots and charts

```
PROC CHART  
  Histograms and bar charts  
  PROC CHART DATA=SASdataset options;  
    VBAR variable / options;  
    HBAR variable / options;  
    options: MIDPOINTS= GROUP= SUMVAR=
```

```
PROC PLOT  
  Scatter plots  
  
  PROC PLOT DATA=SASdataset options;  
    options: HPERCENT= VPERCENT=  
  PLOT yvariable *  
    xvariable = symbol / options;  
  PLOT (yvariables) *  
    (xvariables) = symbol / options ;  
  PLOT options: BOX OVERLAY VREF= HREF=  
  BY variable(s) ;
```

Note that the parenthesized form in the PLOT statement plots each y-variable listed against each x-variable.

3.4. Utility procedures

```
PROC PRINT  
  Print a SAS data set  
  PROC PRINT DATA= SASdataset options;  
    options: UNIFORM LABEL SPLIT='char'  
  VAR variable(s);  
  BY variable(s);  
  SUM variable(s);
```

```
PROC SORT  
  Sort a SAS data set according to one or more variables.  
  PROC SORT DATA=SASdataset options;  
    options: OUT=  
  BY variable(s);
```

◆ THE END ◆