

Quick Guide to SAS Procedures and other fun things

Procedure	Sample Syntax	Major Function(s)	Tips/Comments
proc contents	proc contents data= rmr.provdata2;	List the variables in a dataset and their key properties.	
proc print	proc print; (simplest version)	Print observations from the dataset. Used to spot-check data.	If the dataset is large, limit the output with the <i>(obs=X)</i> option or the <i>var</i> statement. Can also use <i>where</i> , but not at the same time as <i>obs</i> .
	proc print data=sampwghts (obs=20); var county census year sampwght caretype;		
proc sort	<i>Overwrite old data:</i> proc sort data=providers; by idnum;	Sort a dataset by one or more variables. Sorted datasets can be merged with one another in later datasteps.	Default syntax overwrites original dataset with sorted version. Use <i>out=datasetname</i> option to save the sorted version as a different name.
	<i>Output to new dataset:</i> proc sort data=providers out=provscopynt; by county idnum;		
proc format	proc format; value range1f (rename values) 1='high' 0='low'; value income1f (define categories) 0-10000='Poor' 10001-20000='Low Income' 20001-40000='Middle Income' 40001-high='Rich';	1. Give variable values meaningful names if they describe a category of interest. 2. Define categories of continuous variables. Categories can be changed easily to facilitate analysis	--use only one semi-colon (;) per <i>value</i> statement, after all categories have been defined. This looks different than most other lines of code in SAS. --use multiple <i>value</i> statements to define multiple formats in one proc format. --run proc format before (above) procedures that invoke the formats. --format names that end in "F" are easily identified as formats
proc freq	<i>One way freq:</i> proc freq data=rmr.rmr2001_r; table anyps unitftps;	Analyze frequencies of variable values. Make cross tabs to analyze simple relationships.	--may list many variables, table statements, and use parentheses for cross tabs -- <i>where</i> and <i>format</i> statements are useful --in cross tabs, pay attention to row and column percentages.
	<i>Two way freq:</i> proc freq data=ft; table unitftps*year; table year*(hrsftin3 hrsftps3);		

proc means	<i>Unspecified Stats:</i> proc means; class caretype census; var sampwght;	Describe numerical variables, specifically their mean, standard deviation, minimum, maximum, median, and some other percentile points.	-- <i>var</i> variables are analyzed, they must be numeric and usually they are continuous. One <i>var</i> statement with at least one variable is required. -- <i>class</i> variables determine the sub-categories of analysis. They are discrete, either numeric or character, and are often formatted. A <i>class</i> statement is optional. <i>Class</i> categories nest from left to right. --may specify statistics if do not want default.
	<i>Specified Stats:</i> proc means data=FT mean median min max; class county year; var hrsftin2;		

Other procedures you may run across:

Univariate—in recent versions of SAS, proc means can do most of what univariate does. So univariate is used less.

Tabulate—powerful tool for making frequency tables and other calculations, with pretty tables. Somewhat complicated syntax.

Corr/T-test/Reg—SAS statistical procedures. SAS has lots of these, check the documentation. Or use Stata.

3 Commands used with many Procedures:

Format—tells SAS to display and/or categorize variables via a format defined by proc format.

--format names are followed by a period (.)

--can assign a single format to multiple variables, can have multiple formats within one format statement.

Syntax:

format educ edufl. income income1f. var1 var2 yesnof.;

Where—tells SAS to only perform the procedure/display the output if certain conditions hold.

--can have lots of conditions

Syntax:

where gender=1;

Title—adds a title to the top of the output page.

--can have up to three titles, using commands title, title2, title3

--title will carry over to next procedure output unless you change it

Syntax:

title "Income versus education level, Men only";

Example: A fancy proc print:

```
proc print data=gspstudents;  
  title "IM sports participation among GSPP first years";  
  where classyear=2005;  
  format age agef. basketball soccer softball volleyball yesnof.;  
  var lastname firstname age basketball soccer softball volleyball;
```

Helen's closing thoughts:

80% of the stuff I did in SAS over 3 years was data manipulation and simple data analysis with proc freq and means. SAS is a powerful number crunching tool, but it is only as complicated as you want it to be.