

## SAS Quick Tips

### GLOBAL SAS SYNTAX and TIPS

1. There are two types of steps in SAS: DATA and PROC.
  - a. DATA is used to read and manipulate data for analysis
  - b. PROC is used to analyze data
  - c. Each of these "steps" is made up of lines of SAS code.
2. Lines of SAS code are called "statements"
3. Each SAS statement MUST end with a semicolon ;
4. Comments are REALLY USEFUL for documenting programs so that you (and your instructors!) can quickly understand what you did.
  - a. Comments are statements that SAS passes by – that is, SAS will not read them as code or try to execute them.
  - b. Comments come in two forms
    - i. For a single line, start with an asterisk (\*) and end with a semicolon(;).
    - ii. For multiple statements that already contain semicolons, start comment section with /\* and end with \*/  
  
Note: it's easy to get really messed up if you forget to put one of the beginning or end symbols in the comment – SAS may think your entire code is comment, or try to execute the comment text.
5. Some statements show up in both DATA and PROC steps
  - a. BY <subgroup> ; indicates that operations should be performed separately for each subgroup, which is defined by a unique value in the specified *subgroup* variable
  - b. WHERE <condition> ; indicates that operations should be performed only for data meeting the condition specified
6. TITLE statements are REALLY USEFUL for documenting output so that you (and your instructors!) can quickly understand what you did. See below under PROC.
7. Always examine your LOG file to check for SAS logic/syntax errors AND other info that indicate errors in how your code (even if it passes logic/syntax checks). For example, check the name of the data set being used, number of observations in data set, problems reading data, etc.  
  
EVEN THOUGH THE PROGRAM EXECUTES, IT DOESN'T MEAN THAT YOU'VE PRODUCED THE OUTPUT YOU REALLY WANT !!! This principle is otherwise known as GIGO (garbage in, garbage out).

## DATA steps

### Purpose

1. Read data
  - a. Instream: data are included directly inside your program code as data lines (DATALINES)
  - b. External files: data are read from a file, lots of ways to do this (INFILE)
  - c. Existing SAS dataset: use an existing SAS dataset created earlier in the program to initiate the DATA step (SET)
2. Manipulate data
  - a. Create new variables
  - b. Subset the observations and/or variables in the working data set

### Logic

1. It's best to think of a SAS dataset as a rectangular array or matrix
  - a. Rows = observations (SAS uses "observations" for this concept)
  - b. Columns = variables, or single spaces when reading fixed column format using an INPUT statement
2. DATA step takes one observation at a time and performs all of the operations in the DATA step on that observation
  - a. Read in 1<sup>st</sup> observation, check condition, calculate, etc.
  - b. Read in 2<sup>nd</sup> observation, check condition, calculate, etc.
  - c. ...

### Syntax

DATA <dataset name> ;

Statements that tell SAS how to read in data (format of observations)

Statements that tell SAS how to manipulate the data once an observation has been read in

If *instream* approach is being used, then this is where data lines go

DATALINES; indicates that the next line has data in the format specified

Lines following DATALINES are data

Finish this section with semicolon on the line below the last data line

## PROC steps

### Purpose

1. Prepare data for analysis (SORT)
2. Review and explore data (PRINT, PLOT, FREQ, UNIVARIATE)
3. Calculate estimates and standard errors (SURVEYMEANS, SURVEYREG, ...)

### Logic and tips

1. If you ever have a BY statement in a PROC step, you must use a PROC SORT procedure prior to executing the PROC statement
2. Include a TITLEN statement (see below) with your PROC steps to document your output so that we can quickly discern the nature of the output from the PROC step (data, motivation, etc.)

### Syntax

PROC <PROCEDURE NAME> DATA = <dataset name> <OPTIONS using SAS keywords specific to procedure> ;

Statements that follow are specific to the procedure (see SAS reference document)

Title statements to document output (TITLEN) – example syntax:

TITLE '<text without apostrophes>' ;

( $n = 1$  is implicit and text is displayed at the top line of the page)

TITLE2 '<second title text without apostrophes>' ;

( $n = 2$  is explicit and text is displayed below the first title)

Note: TITLEN statements are persistent, and will not change until another TITLE statement with the same value for  $n$  is created.

Hint: One approach is to use the first TITLE to describe the data, and subsequent titles to describe the analysis