

# OVERVIEW of SAS

## What's In SAS?

SAS = Statistical Analysis System

### Main Features:

- Statistical analysis
- Programming language
- Data management
- Standardized output

### The Core of SAS:

- Base SAS - basic data handling (DATA step) and basic PROCedures
- SAS/STAT - large collection of more advanced statistical PROCedures

### Selected Specialized Features of SAS (not in all installations of SAS):

- SAS/ETS - economics and time series analysis
- SAS/Insight - interactive point-and-click data exploration (like JMP)
- Enterprise Miner - data mining
- SAS/LAB - SAS guides you through simple analyses
- SAS/IML - interactive matrix language (write math-like matrix operations)
- macro language - write your own SAS macros (for advanced programmers)
- SAS/Windows - controlling and customizing SAS in Windows
- SAS/GRAPH - color graphics

### Some SAS documentation:

- Internal help
- Online help ([www.sas.com](http://www.sas.com))
- The Little SAS Book: A Primer<sup>1</sup> - 350 pages of getting started

SAS no longer sells current printed general compendia, even for SAS/STAT, but a variety of specialized manuals are available at [www.sas.com](http://www.sas.com). Older compendia are available online from various sources and include:

- SAS 9.2 Language Reference: Concepts (2 vols) - usage for DATA step statements
- SAS 9.2 Language Reference: Dictionary (3 vols) - details for DATA step statements
- Base SAS 9.2 Procedures Guide (3 vols) - basic analyses
- SAS/STAT 9.1 Users Guide (7 vols) - advanced statistical analyses
- SAS 9.2 Companion for Windows - how SAS interacts with Windows

SAS documentation is extensive and intimidating.

But 10% of SAS does 90% of the work.

---

<sup>1</sup> *The Little SAS Book: A Primer* (5<sup>th</sup> edition, 2012) by Lora Delwiche and Susan Slaughter – well-organized, well-written introduction to the way SAS works and to writing programs in SAS (still, it has 350 pages!), takes PROCs from simplest through PROC REG, but not beyond. You will become a SAS programming expert if you learn this book well. About \$35-\$45 from online sources. Earlier editions are probably good enough.

## SAS Programming

The “standard” way to use SAS is to write a SAS program, rather than clicking on menus around a spreadsheet (as in Excel).

A SAS program is a written set of instructions to SAS.

The SAS programming mode is *interpretative*: SAS runs your program in blocks, compiling and executing each block before going on to the next block (a “block” is generally a DATA step or PROCedure step)

There are three environments of SAS programming:

Program	your instructions
LISTing (Output)	the results
LOG	SAS diagnostic report on what it did & what you did wrong

In Windows and SAS OnDemand, each of these environments is in a separate window.

## Overview of a Basic SAS Program

Example program:

```
DATA;                                -- |
  INPUT NAME $ AGE SALARY;           |
  NEWSAL = SALARY * 1.10;             |
CARDS;                                | - DATA step
Bill 36 30000                         |
Mary 48 36000                         |
Jim 32 27000                          |
RUN;                                  -- |
PROC MEANS;                           -- |
RUN;                                  |
PROC REG;                             | - PROC steps
  MODEL SALARY = AGE;                 |
RUN;                                  -- |
```

This SAS program reads employee records, gives everyone a 10% raise, produces summary statistics, and estimates a linear regression relationship  $salary = a + b * age$  between salary and age.

## Some Generalizations about SAS Programs

The two most important building blocks of SAS programs are DATA steps and PROCedure steps. A basic SAS program consists of DATA steps and PROC steps.

- DATA step reads and manipulates data.
- PROC step applies statistical procedures, one procedure per PROC.

A paradigm for a basic SAS program:

```

DATA;                                --|
    INPUT <v1> <v2> ...;            |
    ...;                            |
CARDS;                                | - DATA step
|----|  |----|                      |
|<v1>|  |<v2>|  ...                |
|____|  |____|                      |
RUN;                                  --|
PROC <proc1>;                        --|
RUN;                                  |
PROC <proc2>;                        | - PROC steps
RUN;                                  |
...                                  --|

```

So the basic paradigm for a SAS program is:

**DATA**; followed by **INPUT**, then list of data variables in order (with alphabetic variables followed by \$), then data manipulation statements (if any), then **CARDS**; , followed by the data in columns in the order listed by **INPUT**. This completes the DATA step to read in the data. Next, procedures analyze the data. Procedures begin with the word **PROC** and may be selected from the following (for example), among many others:

<b>MEANS</b>	- simple summary statistics
<b>UNIVARIATE</b>	- more simple summary statistics
<b>SORT</b>	- sorting ( <b>BY</b> required)
<b>PRINT</b>	- listing contents of dataset
<b>PLOT</b>	- plotting ( <b>PLOT</b> required, ex: <b>PLOT SALARY * AGE</b> )
<b>CHART</b>	- histograms, pie charts, etc. (ex: <b>HBAR SALARY</b> )
<b>FREQ</b>	- tables, cross-tabulation ( <b>TABLES</b> required)
<b>CORR</b>	- correlations
<b>GLM</b>	- regression, anova, anacova ( <b>MODEL</b> required, ex: <b>MODEL Y=A X</b> )
<b>REG</b>	- regression ( <b>MODEL</b> required, ex: <b>MODEL Y=X;</b> )
<b>AUTOREG</b>	- autoregressive time series ( <b>MODEL</b> required, ex: <b>MODEL Y=X;</b> )

So, for the simplest analyses:

- Have the DATA step read data into a SAS dataset (an “electronic worksheet”)
- Follow with the names of PROCs you want to apply to these data

## Basic SAS Grammar

The first word of a SAS statement tells SAS the big picture of what you want done.

Ex: “**DATA**” begins a datastep

Ex: “**PROC**” begins a procedure step

Ex: “**INPUT**” starts reading data

Ex: “**CARDS**” tells SAS the data are found on the following lines

Subsequent words provide details on how you want it done.

Extraneous words are not permitted (unlike Minitab, for example).

Comments may be added as separate statements by beginning with an asterisk and ending with semicolon. Ex: **\* This is a comment;**

Every statement must end with a semicolon.

***The most common mistake in SAS is to omit semicolons.***

A SAS statement extends from one semicolon to the next.

- You can put more than one statement on a line (not recommended).
- You can split a statement between 2 or more lines (sometimes necessary).
- I recommend writing SAS programs so that they appear neatly structured, uniform, and visually attractive. This includes indenting and spacing to set off related blocks of statements, aligning = signs in assignment blocks, etc. – this improves readability and makes error detection easier.

I recommend ending each block of code (DATA or PROC) with a **RUN;** statement. Since SAS is interpretative, SAS will not execute a block of code until it is sure that the block has ended. Some procedures (like **PROC REG**) are “interactive”, allowing continued issuance of SAS commands while the procedure is running. Such interactive procs do not end with a **RUN;** statement, but with **QUIT;**