



SAS Programming Basics (Self-Study)



Objectives

- Compare SAS and R in a general way.
- List the components of a SAS program.
- Define SAS syntax rules.
- Use comments to document a program.
- Diagnose and correct a program that contains errors.

35



SAS versus R in General


SAS	R
Script compiler	Command line interpreter
Primarily driven by the data table and procedures	Object oriented
Not case sensitive	Case sensitive

SAS does have the flexibility to interact with objects.
This course focuses on procedural methods.

36

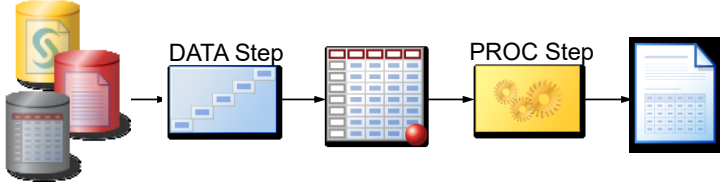
SAS does not have a command line. Code must be run in order to return results.

R is an object-oriented programming language. Results of a function are stored in an object and desired results are pulled from the object as needed. SAS revolves around the data table and uses procedures to create and print output. Results can be saved to a new data table.



SAS Programs

A SAS *program* is a sequence of one or more steps.




- *DATA steps* typically create SAS data sets.
- *PROC steps* typically process SAS data sets to generate reports and graphs, and to manage data.

37

A SAS program is a sequence of steps. There are only two types of steps in SAS: DATA and PROC steps.

- DATA steps read from an input source and create a SAS data set.
- PROC steps read and process a SAS data set, often generating an output report. Think of a PROC step as a function in R.



Step Boundaries

SAS steps begin with either of the following statements:


- a DATA statement
- a PROC statement

SAS detects the end of a step when it encounters one of the following statements:

- a RUN statement (for most steps)
- a QUIT statement (for some procedures)

38

Every step has a beginning and ending boundary. Most steps end with a RUN statement. Some PROC steps end with a QUIT statement. Think of the RUN statement as the right parentheses of an R function.



SAS Program Steps

A *step* is a sequence of SAS statements. This program has a DATA step and a PROC step.

```
data work.newemps;  
  infile "&path\newemps.csv" dlm=',';  
  input First $ Last $ Title $ Salary;  
run;  
  
proc print data=work.newemps;  
run;
```


- R uses parentheses to begin and end a function.

```
work.newemps = read.csv("C:/Users/jobake/Desktop/work.newemps.csv")  
print(work.newemps)
```

39

This program contains the following statements:

- DATA
- INFILE
- INPUT
- RUN
- PROC



SAS Syntax Rules: Statements

SAS statements

- begin with an **identifying keyword**
- always end with a **semicolon**.

```
print data=work.newsalesemps  
  
means data=work.newsalesemps  
  Salary
```

40

SAS statements usually begin with a keyword, and **always** end with a semicolon. Keywords identify the type of statement, and semicolons end the statement.

1.03 Short Answer Poll

How many statements are contained in this DATA step?

```
data work.newsalesemps;  
  length First_Name $ 12  
         Last_Name $ 18 Job_Title $ 25;  
  infile "&path\newemps.csv" dlm=',';  
  input First_Name $ Last_Name $  
        Job_Title $ Salary;  
run;
```

41

Syntax Errors

A *syntax error* is an error in the spelling or grammar of a SAS statement. SAS finds syntax errors as it compiles each SAS statement, before execution begins.


Examples of syntax errors:

- misspelled keywords
- unmatched quotation marks
- invalid options
- missing semicolons



```
proc print data=work.newsalesemp  
run;
```

43

SAS Institute Inc. SAS and the SAS logo are registered trademarks or trademarks of SAS Institute Inc. in the United States and other countries. SAS and the SAS logo are registered trademarks or trademarks of SAS Institute Inc. in the United States and other countries.


Syntax Errors

- The Enhanced Editor in SAS uses the color red to indicate a potential error in your SAS code.
- The RUN statement is bold and blue if all statements correctly end with a semicolon.

```

daat work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=';';
input First_Name $ Last_Name $
      Job_Title $ Salary;

run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
var Salary;
run;

```

44

The Enhanced Editor uses the color red to indicate a potential error in your SAS code. Notice that the misspelled word **D-A-A-T** is displayed in red. This misspelling affects other statements following it because those statements are only permitted in a DATA step, and this is not recognized as such.


The RUN statement in the PROC PRINT step is not the correct font or color, and neither is the word “average” in the PROC MEANS statement.

Code can contain incorrect keywords. For example, “average” is not recognized by the PROC MEANS statement. MEAN is the correct word.

Error messages are written to the SAS log to describe syntax errors.

These are a few of the options used in the demonstration.

- OUTPUT= names the output data set for posterior samples of parameters.
- NMC= specifies the number of MCMC iterations, excluding the burn-in iterations.
- PLOTS= specifies which plots to print.
- SEED= specifies the random number seed (default is 0) to reproduce the analysis.
- PROPCOV= specifies the method used in constructing the initial covariance matrix for the Metropolis-Hastings algorithm. Specifically, it specifies the convergence method to use for optimization.



SAS Comments

This program contains four comments.


```
data work.newsalesemps;  
  length First_Name $ 12 Last_Name $ 18  
         Job_Title $ 25;  
  infile "&path\newemps.csv" dlm=',';  
  input First_Name $ Last_Name $  
        Job_Title $ Salary ;  
run;  
/* comment */  
proc means data=work.newsalesemps;  
  ;  
run;
```

45

This program contains four comments. The first comment describes the program. The second comment is within a statement. The third comment is commenting out a step. The fourth comment is commenting out a statement. R comments do not have an end. They simply comment out everything to the right of the # symbol. SAS comments are more functional.

To comment multiple lines simultaneously, highlight the lines. Hold down the Ctrl key and press /.

To uncomment, highlight the lines. Hold down the Ctrl and Shift keys and press /.



1.04 Multiple Choice Poll

Which statement is a SAS syntax requirement?

- a. Begin each statement in column one.
- b. Put only one statement on each line.
- c. Separate each step with a line space.
- d. End each statement with a semicolon.

46