Chapters 1-2 BASIC CONCEPTS AND DEFINITIONS

There are two basic steps in every SAS program: a DATA step and a PROC (procedure) step. A DATA step begins with the keyword DATA. A PROC step begins with the keyword PROC. Each step consists of statements. A SAS statement has two important characteristics: ➀ it typically begins with SAS **keyword** (that is, a **reserved** word) and ➁ it always ends with a **semicolon (;)**.

For example, the following DATA step

data example;

input id $ age actlevel $ gender $;

cards;

2810 61 MOD F

2804 38 HIGH F

2807 42 LOW M

2816 26 HIGH M

2833 32 MOD F

2823 29 HIGH M

;

consists of

DATA statement data example;

INPUT statement input id $ age actlevel $ gender $;

CARDS statement cards;

and

DATA ENTRY statement 2810 61 MOD F

2804 38 HIGH F

2807 42 LOW M

2816 26 HIGH M

2833 32 MOD F

2823 29 HIGH M

;

The following PROC step

proc print data=example;

run;

consists of

PROC PRINT statement proc print data=example;

RUN statement run;

Exercise. How many steps and how many statements are executed when the program below is processed?

data exercise;

infile ‘C:\Users\mstudent\Desktop\exercise.dat';

input id $ age actlevel $ gender $;

run;

proc sort data=exercise;

by id;

run;

proc print data=exercise;

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SAS LIBRARIES

Every SAS file is stored in **SAS library**. The library named **Work** is a **temporary** library, where files are stored until the end of SAS session. Once SAS window is closed, the files from this library are deleted. All libraries with names other than **Work** are **permanent** libraries, where files are available during subsequent SAS sessions, and are stored until deleted by programmer.

Exercise. Run the code below and verify that the data file is stored in the library Work. Check that it is a temporary library.

data exercise1;

input id $ age actlevel $ gender $;

cards;

2810 61 MOD F

2804 38 HIGH F

2807 42 LOW M

2816 26 HIGH M

2833 32 MOD F

2823 29 HIGH M

;

Exercise. Where will the following data file be stored, in a temporary or a permanent library? What is the name of the library?

data exercise2;

infile 'C:\Users\mstudent\Desktop\exercise.dat';

input id$ age actlevel $ gender $;

run;

To define a library, use a libname statement. The syntax is

libname libref 'C:\Users\mstudent\Desktop\';

where libref is a given library name, for example, MyLib.

It will be a permanent library, but the files in this library will be accessible for the current SAS session only. Each time a SAS session begins, the library must be reassigned. When SAS session ends, the files are no longer available, however, the contents of the library still exist on the operating system.

Exercise. Assign the library by typing libname MyLib 'C:\Users\mstudent\Desktop\'; Import exercise.xlsx file stored on G drive. Then close SAS session and open it again. Verify that the library must be reassigned before the file exercise becomes available.

By default, SAS defines two additional libraries besides **Work**.

* **Sashelp** is a read-only library which contains sample data and other files that control how SAS works.
* **Sasuser** is a permanent library which is a convenient place to store files that need to be saved from session to session. This library stores some files that deal with user’s personal settings.

RULES FOR SAS DATA SET NAMES

TWO-LEVEL NAMES

When referencing a permanent SAS file (the one that is stored in a permanent library), for example, a file named exercise stored in library MyLib, type

data=MyLib.exercise

The general format is always data=libref.filename

ONE-LEVEL NAMES

When referencing a temporary SAS file (the one that is stored in library **Work**), for example, exercise1, write

data=exercise1 or data=work.exercise1

The general format is data=work.filename or data=filename

RULES FOR ASSIGNING LIBREF, FILENAME, AND

SAS VARIABLE NAMES

The rules for assigning libref,filename, and SAS variable names are:

* Libref must be at most 8 characters in length.
* Filename and variable name must be at most 32 characters in length.
* must start with a letter or underscore (\_).
* subsequent characters can be letters, numeric digits, or underscores.
* cannot contain blanks.
* cannot contain any special and SAS reserved characters other than the underscore.

Exercise. Which of the following are valid SAS libref names? Which are valid SAS filename’s? Which are valid SAS variable names? Explain your choice.

Age flight.arrivals proc

\_score\_ flightarrivals myvariable

1response US state patient id

Work.resp thumb.file.name revenue\_$$$

response1 AGE\* clinic\_admit

TYPES OF VARIABLES

* Variables in SAS can be of two types: **numeric** or **character** (not considering dates and times). **Numeric** variables assume only numeric values (written using digits 0,…,9), while **character** variables assume any values. For example, in the data set below, variable Age is numeric, whereas id, actlevel, and gender are character. Note that arithmetic operations make sense on numeric variables, whereas on character variables, even if they assume numeric values, arithmetic operations don’t make sense.

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **age** | **actlevel** | **gender** |
| 2810 | 61 | MOD | F |
| 2804 | 38 | HIGH | F |
| 2807 | 42 | LOW | M |
| 2816 | 26 | HIGH | M |
| 2833 | 32 | MOD | F |
| 2823 | 29 | HIGH | M |

Exercise. Which of the following variables are naturally numeric? Which are character? Explain why.

Age Length of hospital stay Date of birth

Income Presence of disease Date of surgery

Name Number of children Employment status

Gender Insurance status Health status

Weight Height Education

* When data are entered into SAS program, names of character variables must follow by a dollar sign ($). For example, in INPUT statement, id, actlevel, and gender must follow by $.

data exercise1;

input id $ age actlevel$ gender $;

cards;

2810 61 MOD F

2804 38 HIGH F

2807 42 LOW M

2816 26 HIGH M

2833 32 MOD F

2823 29 HIGH M

;

Exercise. Where must dollar signs ($) be placed, so SAS would read the data set correctly? Verify by running SAS.

data overweight;

input patient age gender race bmi;

cards;

1 71 M AA 25.7

2 44 F AA 20.3

3 73 F AA 29.7

4 87 M AA 31.4

5 87 F C 34.8

6 75 F C 34.4

7 59 F AA 26.8

8 62 F AA 28.1

;

* If a value of a variable is **missing**, then enter a period (.) in its place. When the data are printed, SAS outputs a **period** instead of a missing **numeric** variable value, and a **blank** instead of a missing **character** variable value.

Example. The data entered as

data exercise1;

input id $ age actlevel $ gender $;

cards;

2810 61 MOD F

2804 38 HIGH F

2807 . LOW M

2816 26 HIGH M

2833 32 MOD .

2823 29 HIGH M

;

are understood by SAS as



SOME USEFUL BASIC FACTS ABOUT SAS

* SAS doesn’t differentiate between upper and lower case letters. For example, the statement PROC PRINT data=EXERCISE3; is understood the same way as proc print data=exercise3; One exception is if a word is inside a quotation mark. For example, gender=’M’ is not the same as gender=’m’.
* SAS statements can begin and end anywhere on a line. One statement may continue over several lines. Several statements can be on one line. For example, the following lines of code are read by SAS correctly (Verify!).

proc sort data=overweight;

by gender;

run;

proc print

data=overweight; run;

* A comment can be put in SAS code. It must start with /\* and end with \*/

or start with \* and end with a semicolon (;). For example,

/\* These data were collected at VA hospital in Long Beach on May 15, 2017 \*/

data overweight;

input patient $ age gender $ race $ bmi; \*bmi = body mass index;

cards;

1 71 M AA 25.7

2 44 F AA 20.3

3 73 F AA 29.7

4 87 M AA 31.4

5 87 F C 34.8

6 75 F C 34.4

7 59 F AA 26.8

8 62 F AA 28.1

;