

What is happening when DATA step processing?

Bingrong Li Innoventbio

ABSTRACT

As a beginner, it is easy to produce results by using SAS DATA step. But it doesn't work sometimes. You don't know why it is not doing what you want.

The DATA step has been well designed. It is necessary to understanding the internals of the processing. This paper will illustrate what is happening when DATA step processing. Several examples will be given to help understand the concepts of Program Data Vector (PDV), compile time and execution time. Understanding this enables you to take control of the DATA step, and interpret your results with confidence.

INTRODUCTION

The DATA step is the most powerful tool in SAS. It is essential for a good SAS programmer to understand the process of operation. As some data manipulation and creation may only be completed via DATA step. It may fail occasionally without well-understanding of DATA step.

There are two distinct phases for each DATA step and PROC step. One is compilation phase, SAS will check the syntax of the SAS statements and compiles them when a DATA step for execution is submitted. The statements are automatically translated into machine code in this phase. Another is execution phase, a simple DATA step iterates once for each observation that is being created by default.

When a DATA step for execution is submitted in your code, it is first compiled and then executed. The Figure 1 from SAS help is provided to describe the processing of the DATA Step.

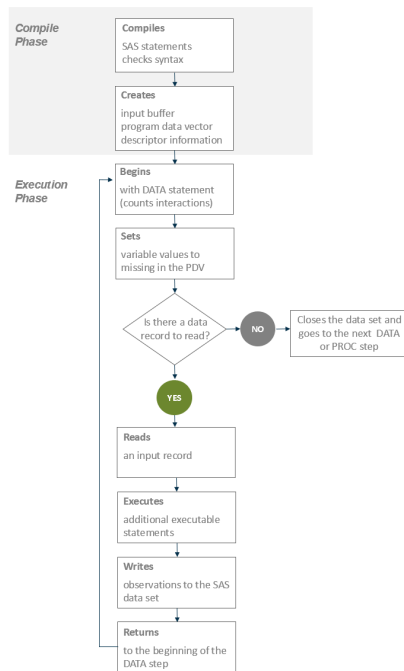


Figure 1. Flow of Action in the DATA Step

THE COMPILATION PHASE

SAS performs the following steps during this phase:

- translates the SAS statements into machine code automatically, which will be executed later.
- creates input buffer if there is an INPUT statement, which is a logical area in memory into which SAS reads each record of raw data.
- creates the Program Data Vector (PDV), which is a logical area in memory where SAS builds a data set, one observation at a time. SAS reads data values from the input buffer or creates them by executing SAS language statements. The data values are assigned to the appropriate variables in the PDV. From here, SAS writes the values to a SAS dataset as a single observation.
- sets up two automatic variables in PDV, `_N_` and `_ERROR_`. The `_N_` variable counts the number of times the DATA step begins to iterate. The `_ERROR_` variable signals the occurrence of an error caused by the data during execution. The value of `_ERROR_` is either 0 (indicating no errors exist), or 1 (indicating that one or more errors have occurred). These variables will not be written to the output data set.
- creates descriptor information for data sets and variable attributes.

THE EXECUTION PHASE

After the compilation phase, it comes into picture and performs the following steps as below order during the execution phase:

- A new iteration of the DATA step begins, and the `_N_` automatic variable is incremented by 1 for each time the DATA statement executes. This is a sample bulleted list item.
- sets the newly created program variables to missing in the PDV.
- reads a data record from a raw data file into the input buffer, or it reads an observation from a SAS dataset directly into the PDV. An INPUT, MERGE, SET, MODIFY, or UPDATE statement can also be used to read a record.
- executes any subsequent programming statements for the current record.
- writes an observation to the SAS data set at the end of a statement, the system automatically returns to the top of the DATA step, and the values of variables created by INPUT and assignment statements are reset to missing in the PDV. Note that variables that you read with a SET, MERGE, MODIFY, or UPDATE statement are not reset to missing here. The difference will be explained in the later examples.
- counts another iteration, reads the next record or observation, and executes the subsequent programming statements for the current observation.
- terminates when SAS encounters the end-of-file in a SAS data set or a raw data file.

A WALKTHROUGH FOR DATA STEP

The below programs will be helpful to understand how does SAS run. And the difference between INPUT and SET statement will be explained. The dataset SV_1 is created in the first DATA step, which includes the weight for subject 01001 at each visit. The dataset SV_2 includes the weight for subject 01002 at each visit. The variable VSSTAT is only in dataset SV_2. In the third DATA step, these two datasets are set to the dataset SV.

```
data VS_1;
input SUBJID $ VISIT $ WEIGHT VSYN $;
datalines;
01001 Visit0 . N
```

```

01001 Visit1 55 Y
01001 Visit2 54.5 Y
01001 Visit3 55.1 Y
;
run;
data vs_2;
input SUBJID $ VISIT $ WEIGHT VSYN $ VSSTAT;
datalines;
01002 Visit2 50.1 Y .
01002 Visit3 49.9 Y .
01002 Visit4 . N .
01002 Visit5 50.2 Y .
;
run;
data VS;
set VS_1 VS_2;
if VSYN='N' then VSSTAT=1;
run;

```

The process of the first DATA step is shown in Figure 2. The input buffer, PDV, `_N_` and `_ERROR_` are generated after compilation. And the variable created by INPUT statement is set to missing before each iteration in PDV. The dataset `SV_1` is created after all iteration are completed

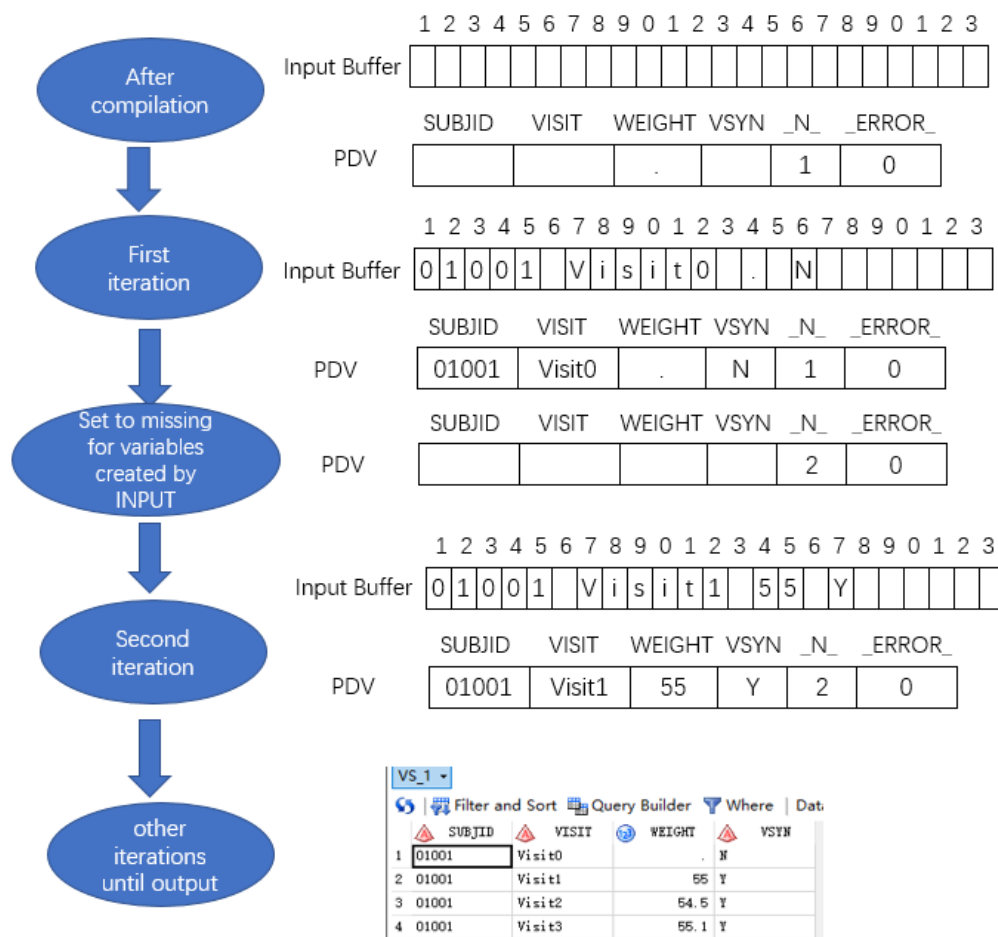


Figure 2. Process of dataset `SV_1`

As the first DATA step, the process of second DATA step is shown in Figure 3. The dataset SV_2 is created finally.

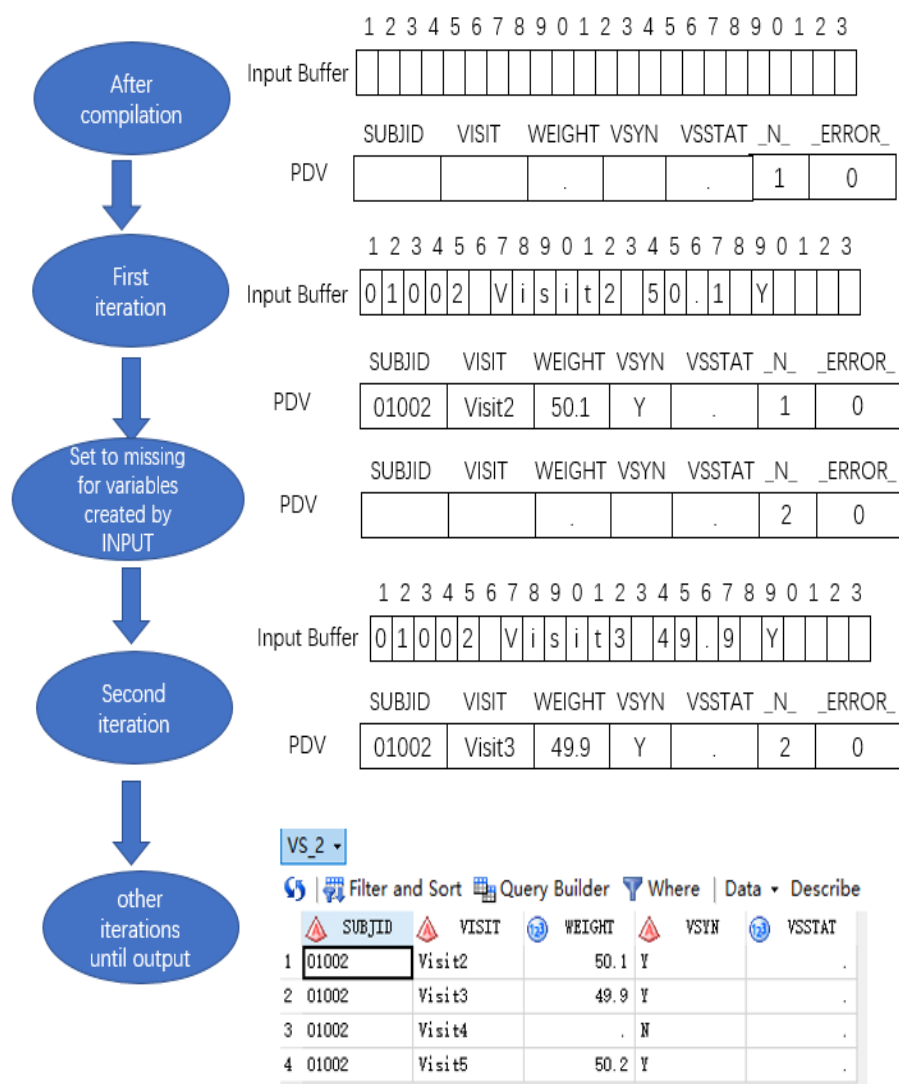


Figure 3. Process of dataset SV_2

The process of the third DATA step is shown in Figure 4. Dataset SV_1 and SV_2 are set to dataset SV. The input buffer is not created in SET statement. Only PDV, _N_ and _ERROR_ are generated. And the variable created by SET statement is not set to missing before each iteration in PDV. The variable VSSTAT is not in the first dataset SV_1, so the value of VSSTAT in observations set from SV_1 is retained from the first observation. If you want to get the right value of VSSTAT, a new variable should be created or the old variable should be renamed.

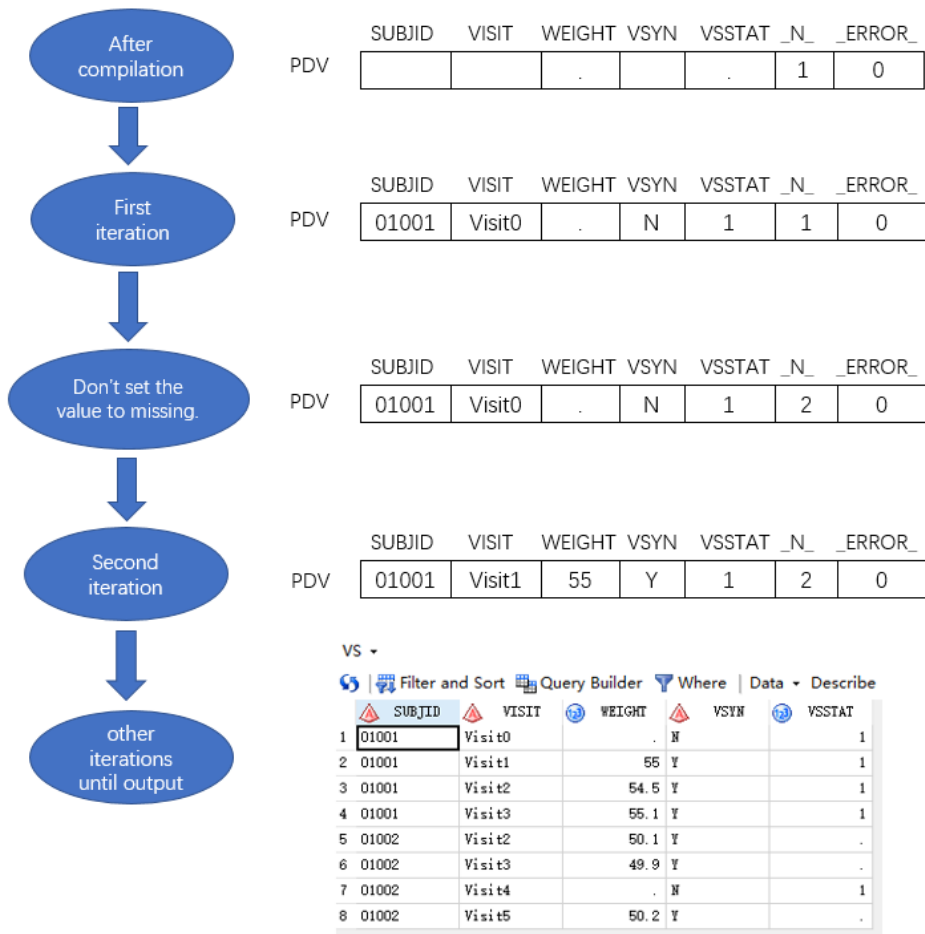


Figure 4. Process of dataset SV

CONCLUSION

To generate the write result by SAS DATA step, it is necessary to well know the internals of the processing of DATA step. This paper has instructed the process. We should also be careful when using DATA step.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Bingrong Li
libingrong1992@163.com