

## Working with Datetime Variable from Stata

Haiyin Liu, University of Michigan School of Nursing, Ann Arbor, MI  
Wei Ai, University of Michigan School of Information, Ann Arbor, MI

### ABSTRACT

Most SAS<sup>®</sup> users have to transfer Stata data files into SAS frequently. However, you must be careful when converting Stata Datetime to SAS Datetime. In this paper, we examine how SAS or Stata stores Datetime variables differently. We propose a correction function to accurately transfer Stata Datetime into SAS.

### INTRODUCTION

There are times when you receive data files from your Stata collaborator and need to perform analysis with that file in SAS. The most common method to read a Stata data set into SAS is the IMPORT procedure. While the IMPORT procedure is user friendly and relatively simple, problems may occur when applying PROC IMPORT to Datetime variables.

### AN EXAMPLE

#### THE STATA DATA SET

The following Stata data set TEST.DTA is stimulated for this paper. This data set contains unique identifier for patients and their hospital visits. The RECORDED\_DTM is the date and time when healthcare providers enter medical records of the patients into the electronic system. The data set is stored as a Stata file, and we hope to analyze it in SAS.

TEST.DTA		
PATIENT_ID	VISIT_ID	RECORDED_DTM
1	1	02jul2011 10:27:00
1	2	21oct2011 08:33:00
2	1	05mar2011 12:10:00
2	2	16sep2011 09:50:00
2	3	20dec2011 18:06:00

#### IMPORT INTO SAS

We use the IMPORT procedure to read TEST.DTA into SAS. We write the following SAS code:

```
PROC IMPORT DATAFILE='C:\TEST.dta'  
            OUT=OUT.TEST DBMS=STATA REPLACE;  
RUN;
```

SAS will detect the Stata Datetime variable RECORDED\_DTM and store it as an integer. We need to assign a desired SAS Datetime format to RECORDED\_DTM in the DATA step.

```
DATA OUT.TEST;  
    SET OUT.TEST;  
    FORMAT RECORDED_DTM DATETIME20.;  
RUN;
```

Clearly, something goes wrong with RECORDED\_DTM in the exported SAS data set OUT.TEST. We need to take a further look at how Datetime variables are handled by SAS and STATA to explain this unexpected result.

## OUT.TEST

Obs	PATIENT_ID	VISIT_ID	RECORDED_DTM
1	1	1	31DEC9999:23:59:59
2	1	2	31DEC9999:23:59:59
3	2	1	31DEC9999:23:59:59
4	2	2	31DEC9999:23:59:59
5	2	3	31DEC9999:23:59:59

## LOOK UNDER THE HOOD

We want to examine how SAS and Stata store Datetime variables.

In SAS, Datetime variables are stored as integers. The integer value represents the number of seconds that have elapsed since midnight January 1, 1960. That means, 01JAN1960 00:00:00 is SAS Datetime 0, which will be referred to as the sentinel in the rest of the paper. Positive numbers represent dates and times after the sentinel, and negative numbers represent dates and times before. Integer value 1 is one second after the sentinel: 01JAN1960 00:00:01; and integer value -1 is one second before that: 31DEC1959 23:59:59.

In Stata, Datetime variables are also stored as integers and represent the elapsed time since January 1, 1960. Stata uses the same sentinel as SAS, but it differs from SAS in the units in which the elapsed time is calculated. Instead of using second as the unit, Stata uses millisecond. The integer value now represents the number of milliseconds since the sentinel. In Stata, integer value 0 represents 01JAN1960 00:00:00.000. Integer value 1 represents the first milliseconds of January 1, 1960: 01JAN1960 00:00:00.001. Similarly, integer value -1 represents 31DEC1959 23:59:59.999.

This explains why we get the undesired result. When SAS received the imported Stata file TEST.DTA, it ends up with a numeric value recording a Datetime in Stata's encoding. For example, 02JUL2011 10:27:00 is stored as 1,625,221,620,000 in Stata, meaning 1,625,221,620,000 milliseconds since the first millisecond of January 1, 1960. SAS receives the numeric value 1,625,221,620,000 and interprets it as 1,625,621,620,000 seconds since the first second of January 1, 1960, resulting in the enormous Datetime in the output file.

## CORRECTION

From the above discussion, we can see that although both SAS and Stata store dates and times as integer values, they differ in the units being used. SAS record Datetime values as the numbers of seconds from midnight 01JAN1960, while Stata record Datetime values as the numbers of milliseconds since the same sentinel. We need to convert milliseconds to seconds to fix the problematic Datetime values when converting Stata data set into SAS. To achieve that, we present a correction DATA step as following:

```
DATA OUT.TEST_1 (DROP=RECORDED_DTM) ;  
    SET OUT.TEST;  
    CORRECTED_DTM = RECORDED_DTM/1000;  
    FORMAT RECORDED_DTM CORRECTED_DTM DATETIME20. ;  
RUN ;
```

As shown below, the correction code successfully produces the expected results. The RECORDED\_DTM has accurate values in the output data set OUT.TEST\_1.

## OUT.TEST\_1

Obs	PATIENT_ID	VISIT_ID	CORRECTED_DTM
1	1	1	02JUL2011:10:27:00
2	1	2	21OCT2011:08:33:00
3	2	1	05MAR2011:12:10:00
4	2	2	16SEP2011:09:50:00
5	2	3	20DEC2011:18:06:00

## CONCLUSION

The commonly used IMPORT procedure may generate unexpected result when importing Stata Datetime variables into SAS. In this paper, we look deeply into the representations of Datetime variables in both software, and show that the unexpected result is due to the different time unit Stata and SAS use to measure elapsed time. Though both Stata and SAS store Datetime variables numerically as the time that has elapsed from the same sentinel, SAS uses the number of seconds while Stata uses the number of milliseconds. It is important to convert the milliseconds to seconds to avoid mistakes. In the last section of this paper, we provide a correction DATA step that is able to convert Stata Datetime into SAS Datetime accurately.

## REFERENCES

Andrew H. Karp. "Working with SAS® Date and Time Functions: Foundations & Fundamentals." Proceedings of the SAS® Global Forum 2010 Conference. Paper 134.

Stata Manual. "Working with Dates and Times". Stata Manual 2013. U24.

## CONTACT INFORMATION

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

Haiyin Liu  
University of Michigan School of Nursing  
[haiyinl@umich.edu](mailto:haiyinl@umich.edu)

Wei Ai  
University of Michigan School of Information  
[aiwei@umich.edu](mailto:aiwei@umich.edu)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.