

Tools to Transfer Data between SAS® and Stata®

Dan Blanchette, Carolina Population Center, UNC-CH, Chapel Hill, NC

ABSTRACT

This paper demonstrates how to transfer data between SAS and Stata by running Stata from within SAS or SAS from within Stata using any of the following programs I have written:

- SAVASTATA: SAS macro to save a SAS data set as a Stata data set
- SAVASAS: Stata program to save a Stata data set as a SAS data set
- USESAS: Stata program to use a SAS data set while in Stata
- SAVAS (c-shell) script: Invoked at a Unix or Linux prompt makes a Stata data file copy of a SAS data file or a SAS data file copy of a Stata data file.

Since SAVASTATA runs during a SAS session and SAVASAS runs during a Stata session, you may choose to save only the converted file avoiding the unnecessary creation of a data file just to run through a conversion software program. USESAS simply allows you to open a SAS data set into Stata. Space is saved, file management is decreased, and documentation of the creation of the converted file is saved in the log of your program. User-defined formats are saved as value labels in Stata and vice-versa. Special missing data (.a - .z) are maintained.

These programs are free and can be downloaded from the internet:

http://www.cpc.unc.edu/services/computer/presentations/sas_to_stata

INTRODUCTION

There are times when you need to exchange data with someone who requires a Stata data file or when you receive a Stata data file and need to do analysis and/or data management with that file. If you have both SAS and Stata available to you on the same machine, it is possible to use both SAS and Stata to transfer the data between them with the transfer tools I have written. These tools work on Windows, Unix and Linux platforms. The c-shell script SAVAS works only on Unix and Linux.

SAVASTATA

The SAVASTATA SAS macro is pronounced "save-a-Stata" but means "save as Stata." This tool is available to you in SAS and lets you save your most recently created SAS data set in the WORK library as a permanent Stata data set on your hard drive. SAVASTATA first takes a look at your data, figures out the type of variables that are present and the optimal storage type for each variable to store the data in a Stata data file, writes a SAS program to output the data to a temporary ASCII file, and then runs that program. SAVASTATA also writes a Stata input do-file to read in the ASCII data as well as to preserve SAS formats and any user-defined formats assigned to numeric variables. Not all SAS formats are possible to preserve in Stata. At this point SAVASTATA calls Stata from your machine's operating system prompt to run the Stata input program which reads the data into Stata and saves the data set using Stata's "save" command to store the data as a Stata data file. It then reports back to SAS and documents in the SAS log how the data transfer process went, as well as how many variables and observations were saved.

In order to preserve the accuracy of the data, SAVASTATA uses the BEST18 format to write numeric data to the intermediary ASCII data file for any variable that contains decimal values.

FEATURES OF SAVASTATA

- 1) It directly saves a data set from SAS to Stata, cutting out the middle step of having to save a SAS data set to your hard drive just to run it through a conversion software program to make a Stata copy.
- 2) The resulting Stata data file has its variables optimally stored, which may save disk space.
- 3) By default user-defined formats assigned to numeric variables in the SAS data set are preserved without your having to make any special effort.
- 4) Documentation of the transfer process is saved in your SAS log.
- 5) By default the version of the saved Stata data file is the version of Stata that you are currently using.

6) With the “check” option SAVASTATA generates basic descriptive output from SAS and Stata for you to compare for yourself that the data were accurately preserved through the transfer process.

HOW TO USE SAVASTATA

Before using the SAVASTATA macro, you need to let SAS know about SAVASTATA. One way to do that is to use the %INCLUDE statement. For example, if you have saved the savastata.mac file to “c:\SASmacros”, tell SAS about this macro by adding the following line to your SAS program:

```
%include "c:\SASmacros\savastata.mac";
```

To use the SAVASTATA macro you have to make a call to it by adding a line such as the following to your SAS program:

```
%savastata("c:\mySASdir",-old);
```

The information inside the parentheses is passed on to the SAVASTATA macro. The first string of information is the location where you want to save your SAS data set as a Stata data set. This is something like a LIBNAME statement. Following the comma is the second string of information which contains the options you wish to pass on to the SAVASTATA macro. You can use as many options as you like or none at all. Documentation of all the options is in the savastata.mac text file.

EXAMPLE USE OF SAVASTATA

```
%include "c:\SASmacros\savastata.mac";
data work.ToBeStata;    ** This makes a copy of the SAS data set in the in
                        * the work library. **;

    set in.mySASfile;
run;

%savastata("c:\mydata\",-old);    ** Saves the data set in the c:\mydata\ directory if
                                * it does not already exist in that directory. **;

%savastata("c:\data\",-old -replace);    ** Saves the data set as the previous
                                        * version of Stata in c:\data\
                                        * directory, overwriting it if it
                                        * already exists. **;
```

SAVASAS

The SAVASAS Stata command is pronounced “save-a-SAS” but means “save as SAS”. You use SAVASAS while in Stata to save your Stata data set as a SAS data set. This program works just like SAVASTATA in that it first figures out what types of variables you have and what variable lengths are optimal for storing your data as a SAS data set, and then it writes a Stata do-file to save the data in a SAS data file. If requested, the value labels are stored in a SAS formats catalog file. SAVASAS writes a SAS program to read the data into SAS and save the data set. SAS is then called by Stata at your machine’s operating system prompt and runs this input program. When the program is completed, SAVASAS reports back to Stata on how the process went and how many variables and observations were saved, and documents all this in the Stata results window/log file.

Starting in version 8.2 Stata introduced the “fdasave” command which allows you to save your Stata data set as a SAS XPORT Transport format data file. SAVASAS takes advantage of this by using intermediary SAS XPORT files to transfer the data to SAS.

FEATURES OF SAVASAS

- 1) It directly saves a data set from Stata to SAS, cutting out the middle step of having to save a Stata data set to your hard drive just to run it through a conversion software program to make a SAS copy.
- 2) By default the resulting SAS data set has the lengths of its variables optimally set, which may save a lot of disk space.
- 3) If specified with the “formats” option, value labels used in the Stata data set are preserved in a SAS formats catalog file.
- 4) Documentation of the transfer process is saved in your Stata results window/log file.
- 5) By default the version of the saved SAS data file is the version of SAS that you are currently using.
- 6) With the “check” option SAVASAS generates basic descriptive output from Stata and SAS for you to compare to check for yourself that the data were accurately preserved through the transfer process.

HOW TO USE SAVASAS

Before using SAVASAS you need to install it. The easiest way to do this is to run the "net install" command in Stata:

```
net install savasas , from ///
(http://www.cpc.unc.edu/services/computer/presentations/sas_to_stata/) replace
```

The word "savasas" has to be typed in all lower case. The "net install" command requires that your machine be connected to the internet because it downloads the files required to run SAVASAS from the specified internet site and saves them to the directory where your personal downloads of Stata commands are stored. Once installed, SAVASAS can be run by Stata when the "savasas" command is issued.

To use the "savasas" command all you have to do is type "savasas" at a Stata command line or add it to your Stata do-file. By default your Stata data set will be saved as a SAS data set using the same name as the Stata data set but with the ".sas7bdat" file extension. You can specify a different name and location to save the SAS data file as well as incorporate any combination of SAVASAS's options. A call to the "savasas" command should look like the following. What is in brackets is not required to run the command:

```
savasas [using "myfilename.sas7bdat"] [, options]
```

Documentation of all the options is in the SAVASAS help which can be found in Stata's help for all commands.

EXAMPLE USE OF SAVASAS

To save the data set in memory to the same directory with the same name but as a .sas7bdat file, simply type:

```
savasas
```

To save the data set in memory to the current directory with the name "mySASdata" as a .sas7bdat file, type:

```
savasas using "mySASdata"
```

To save the data set in memory to the c:\data\ directory with the name "mySASdata" as a .sas7bdat file and create two check files in the same directory as the output data file, type:

```
savasas using "c:\data\mySASdata", check
```

To save only variables between "gender" and "q32_b" from the data set in memory to the current directory with the name "mySASdata" as a .sas7bdat file and create the formats catalog file "mySASdata.sas7bcat" containing SAS formats of any Stata value label, replacing an existing file if necessary, type:

```
savasas gender-q32_b using "mySASdata", formats replace
```

More examples:

```
savasas using "mySASdata.sas7bdat", replace
savasas in 1/30 using "mySASdata", type(sas6) check
savasas using "mySASdata", messy type(sasx) rename
```

USESAS

The USESAS Stata command is pronounced "use SAS" and basically all it does is write a SAS program that runs SAVASTATA. The only difference is that USESAS does not save your data set; it just loads your data set into memory for you to use in Stata. Once the data set is loaded into Stata, it is a Stata data set. Any Stata command can be executed on the data including the "save" command to save the file as a Stata data file, or even the SAVASAS command to save it as a SAS data file. With USESAS you may decide to keep all your data in SAS data files. This is extremely helpful if you prefer to do all your data management in SAS yet have to do special analyses in Stata.

FEATURES OF USESAS

- 1) It directly reads a SAS data set into Stata.
- 2) It has all the features of SAVASTATA.
- 3) It can use a SAS program (for example: usesas using mySASprog.sas) that creates a SAS data set. The last data set created by the program is the one loaded into Stata.
- 4) It figures out how much memory you need to have set in Stata to load the data.

HOW TO USE USESAS

Before using USESAS you need to install it. The easiest way to do this is to run the "net install" command in Stata:

```
net install usesas , from ///
(http://www.cpc.unc.edu/services/computer/presentations/sas_to_stata/) replace
```

The word "usesas" has to be typed in all lower case. The "net install" command requires that your machine be connected to the internet because it downloads the files required to run USESAS from the specified internet site and saves them to the directory where your personal downloads of Stata commands are stored. Once installed, USESAS can be run by Stata when the "usesas" command is issued.

The "usesas" command requires you to know the location and full name of the SAS data file that you want to use in Stata. If you want to keep user-defined formats associated with numeric variables in the SAS data set, you need to specify the "formats" option and have the formats stored in a SAS formats catalog file (.sas7bcat) with the same name and location as the input SAS data file or in a file named "formats.sas7bcat" in the same location as the SAS data file. A call to the "usesas" command should look like the following. What is in brackets is not required to run the command:

```
usesas using "myfilename.sas7bdat" [, options]
```

Documentation of all the options are in the USESAS help which can be found in Stata's help for all commands.

EXAMPLE USE OF USESAS

To load a SAS data set named "mySASdata.sas7bdat" located in the current directory into Stata, type:

```
usesas using "mySASdata.sas7bdat"
```

To load from a different directory than the current one, type:

```
usesas using "c:\data\mySASdata.sas7bdat"
```

To have both SAS and Stata generate check files, type:

```
usesas using "c:\data\mySASdata.sas7bdat", check
```

To load only selected variables and observations, type:

```
usesas using "mySASdata.sas7bdat", keep(id--qvm203a) if(1980<year<2000) ///
in(1/500)
```

SAVAS

The c-shell script SAVAS is pronounced "save as" and means "save as SAS or save as Stata". SAVAS either writes a Stata do-file to run SAVASAS or writes a SAS program that runs SAVASTATA. SAVAS figures out which program to run based on the file extension of your file. If your file ends in ".sas7bdat", then SAVAS knows that it is a SAS data file and makes a Stata copy of that file using the same name and location but the appropriate file extension. Conversely, if your file ends in ".dta" SAVAS recognizes it to be a Stata data file and makes a SAS copy of it. Savas is called from a Unix or Linux prompt usually in the same location as the input data file. Just like other data conversion programs, SAVAS needs the SAS or Stata data file to exist on the hard drive so that it can make a copy of the file in the "other" format and save it to the hard drive.

FEATURES OF SAVAS

- 1) It makes a SAS copy of a Stata data file or a Stata copy of a SAS data file.
- 2) SAVAS has all the features of SAVASTATA and SAVASAS.
- 3) It can process a series of data files with the file names typed out or with the use of wild cards.

HOW TO USE SAVAS

Before using SAVAS you need to have SAVASTATA and SAVASAS already installed, as well as the SAVAS script itself. At a Unix or Linux prompt type "savas" in all lower case, followed by the name of the file you want to create a copy, and hit Enter (the Return key):

```
$ savas [-options] mydata.ext
```

If you want to preserve user-defined formats assigned to numeric variables in your SAS data set, use the "-formats" option. Just like USESAS, SAVAS will use a SAS formats catalog file with the same name and in the same location as the data file or, if no such file exists, a file with the name "formats.sas7bcat" in the same directory as the target data file to access the formats used by the data. All the options for SAVASTATA and SAVASAS are available and can be viewed by typing "savas" at a prompt with no file name to process.

EXAMPLE USE OF SAVAS

To make a Stata copy of a SAS data set named "mySASdata.sas7bdat" located in the current directory, type:

```
savas mySASdata.sas7bdat
```

This will create the file "mySASdata.dta" which is a Stata data file.

To make a SAS copy of a Stata data set named "myStataData.dta" located in the current directory, type:

```
savas myStataData.dta
```

This will create the file "myStataData.sas7bdat" which is a SAS data file.

To make a Stata copy of a SAS data set and save SAS formats stored in a file "mySASdata.sas7bcat", type:

```
savas -formats mySASdata.sas7bcat
```

To make a copy of a data file when you know you want to replace any existing copies of that file, type:

```
savas -replace myStataData.dta
```

To create check files named "mySASdata_SAScheck.lst" and "mySASdata_STATACheck.log," type:

```
savas -check mySASdata.sas7bdat
```

CONCLUSION

These four transfer tools make use of the SAS and Stata software you have purchased and can perhaps make the transfer of data between the two easier than it is with a purchased data transfer program. When it comes time to upgrade to the next versions of SAS and Stata, these programs should continue to work and save to those new versions.

ACKNOWLEDGMENTS

The Rand Corporation's c-shell script SAS2STATA was inspiration for these programs.

CONTACT INFORMATION

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

Dan Blanchette
Carolina Population Center
University of North Carolina at Chapel Hill
CB# 8120, University Square
123 West Franklin St.
Chapel Hill NC 27516-2524

Work Phone: 919-966-1714
Fax: 919-966-6638
Email: Dan_Blanchette@unc.edu
Web: <http://www.cpc.unc.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.