Paper 142-29

SAS® Backwards: Helping SAS and SPSS® Students Speak the Same Language

Joseph J Guido, MS, University of Rochester Medical Center, Rochester, NY
Adam B Rains, MSc, University of Rochester Medical Center, Rochester, NY

ABSTRACT

While dedicated data analysts have always been, by necessity, conversant in multiple applications many casual users of SAS and SPSS remain divided into camps based upon the software on which they were trained. Because of the need to focus on methodological instruction (over software usage) and cost reduction, many academic institutions have adopted the exclusive use of a single analytic package (typically SAS or SPSS). While this strategy has obvious advantages, the necessary focus placed on software-specific instruction, when paired with the fact that most students only ever get one or two courses of this nature, can result in crops of ill-prepared data analysts. Though these casual users are methodologically savvy, their forced reliance on a single analytic package is tantamount to functional statistical illiteracy. In failing to address the needs of these casual users, we are doing a disservice to both our students, who may one day have to use another package, and our respective areas of research, by erecting barriers to interdisciplinary collaboration.

Though the authors' primary goal is to promote conversancy with a variety of statistical packages, the following paper focuses on providing examples of their techniques with specific reference to the analytic methods most commonly used in introductory statistics courses. This necessarily abbreviated overview is intended to (a) provide a rationale for the authors' strategy of promoting software-independence, (b) demonstrate several salient take-home examples, and (c) reacquaint users with the fundamental similarities and strengths of these two, commonly used, statistical packages.

INTRODUCTION

The problem, if we indeed agree there is one, seems to be that some students have a perception of "helplessness" in having to switch from one package to the other. The problem exists or persists because of certain prejudices or preferences of a visual versus a syntax-oriented mode of statistical analysis. It is difficult to "fix" the problem as some students may become deeply entrenched in one package or another based on preferences of their instructors or the current state of affairs in their academic discipline or field.

This paper is an outgrowth of the authors' experiences teaching SAS and SPSS to graduate and medical students of a variety of levels of statistical expertise. The question which most frequently arises in these settings is "Why do I have to take this course?" The answer invariably falls back to ensuring that these students leave their respective degree programs as competent data analysts and consumers. Invariably, students will then fall back to the standard "Why do I have to use SAS when I know SPSS or some other statistical package". This pragmatic and seemingly straightforward question is less easily answered.

As alluded to above, both authors have had occasion to use the SAS and SPSS systems in both academic and strictly analytic settings. This puts them in a unique frame of reference to attempt to answer the questions above. Through the course of this paper, they will focus on similarities in the two packages as a way of combating student frustration and to assist users of other statistical packages in making the switch to SAS.

THE METHOD

The authors' instructional methodology relies on a user's existing knowledge regarding the software they were trained on to help ease their use (or transition to) other statistical analysis packages. Rather than forcing users to learn a new piece of software from the ground up, the authors instead encourage new users to focus on determining how to accomplish familiar tasks in a new environment. This strategy, which we term "mapping", involves drawing parallels between procedures, concepts, vocabulary, and output to help underscore the similarities between applications. This functionalist approach to software instruction is in keeping with the pragmatic needs of analysts and students. Good examples of this needs-based approach to statistical instruction abound in the SAS Books by User catalogue. A favorite of the authors is Glenn Walker's recent book *Common Statistical Methods for Clinical Research with SAS Examples, Second Edition* (Walker, 2002). Each of this book's chapters focuses on a single mode of testing and walks the user through them by providing relevant, real world and SAS examples. A similarly functional approach can be seen in the layout of the perennial classic and student favorite *The Little SAS Book A Primer, Revised Second Edition* by Lora Delwiche and Susan Slaughter (2002). Both the Walker and Delwiche and Slaughter books are used in the two graduate level courses taught by the authors. As we have previously opined, a user already well versed in statistics and trained on either SAS or SPSS should be able to accomplish most basic tasks and a good deal of higher-level analysis using either system. The real distinctions tend to come out in complex analyses or in situations where the users wish to have an enormous amount of control over the output and analysis.

COMPARING THE TWO SYSTEMS

The story of comparability and compatibility of SAS and SPSS is a long one fraught with numerous changes borne of the various updates each package has seen since first being released. While both started as predominantly code-driven analytic packages, SPSS, over time, migrated towards a predominantly graphical user interface (GUI) while SAS focused on improvements to the implementation of its code-driven metaphor. As of the latest release, SPSS exists as a GUI program, which generates code in the background, whereas SAS retains a code-driven interface, but with an optional menu-driven front end called ANALYST Application (which is part of the SAS/STAT software). The creation of the ANALYST Application is, no doubt, a response to demands from users who learned on menu-driven applications and wished to maintain this intuitive interface in the SAS environment.

One of the issues, which present a problem to users switching between SAS and SPSS, is that of vocabulary. The specialized terminology of each application, while describing frequently similar processes, is a particular source of confusion for new users and those attempting to collaborate across applications. Users of either package tend to adopt the language of their chosen program and can, over time, become so dependent on it that they have a hard time expressing themselves not using this lingo. This situation is a recurrent source of frustration to those trying to collaborate or switch from SPSS to SAS. While attempts (Table 1) have been made to create a shared lexicon or glossary by mapping lingo across applications, this issue continues to be problem for numerous users.

Table 1. Mapping vocabulary between SPSS and SAS

SPSS term	Analogous SAS term
active file	*no analogous term*
no analogous term	temporary SAS data set (also called a table)
case	observation (also called a row)
command	statement
file handle	libref
function	function
input format	informat
numeric data	numeric data
output format	format
procedure	procedure
save file	permanent SAS data set (also called a table)
SPSS data file	permanent SAS data set (also called a table)
string data	character data
syntax	statements
syntax file	program
system file	permanent SAS data set (also called a table)
value label	user-defined format
variable	variable
variable label	label
no analogous term	DATA step
no analogous term	PROC step

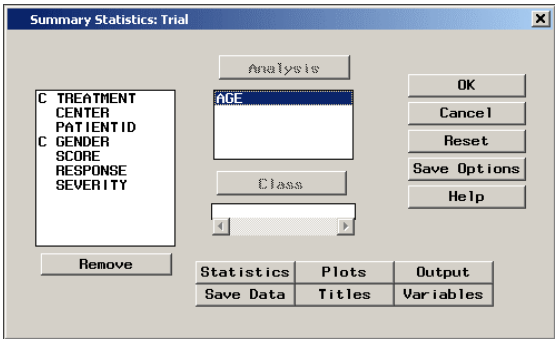
(Reproduced from Delwiche and Slaughter, 2002)

While vast differences still exist between the systems, the creation of new elements, like the SAS ANALYST Application, are testaments to the increasing rapprochement of these two systems. Other examples of the increasing similarities between these applications can be seen in their read/write support for each others data files and formats, and implementation of standard export / interchange formats (XML, HTML, RTF, PDF…).

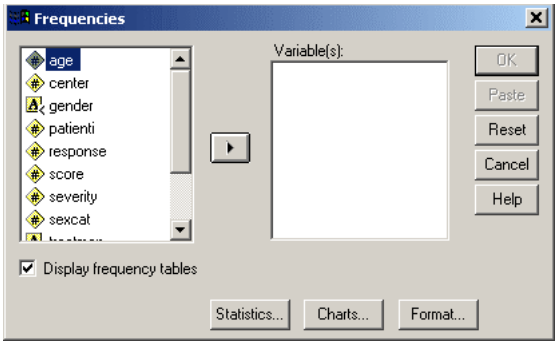
The GUI similarities between SAS (Analyst Application) and SPSS in a simple descriptive statistics example are shown in Figure 1. Through the point and click method, both users of SPSS and SAS can get an appreciation for the feel and ease of use of both packages and realize that it may not be that difficult to switch between them.

Figure 1. Descriptive Statistics: Similarities between SAS (Analyst Application) & SPSS

SAS



SPSS



STRENGTHS AND WEAKNESSES

Though a complete evaluation of the strengths and weaknesses of each of these applications is beyond the scope of the present paper we will endeavor to cover the most salient distinctions from a new user's perspective. The fundamental differences between these two system can be summed up in simplest terms by the phrase "SAS is more flexible but harder to use, SPSS is easier to use but lacks flexibility". In the final analysis, the ideal package for any given user is determined by a number of factors, not the least of which being ease-of-use.

Among the strengths of the SAS system most commonly cited by dedicated users are its flexibility and the degree of control it gives users. When boiled down to specifics this generally means a greater diversity of statistical tests, greater control of what optional procedures are run as part of a given statistical test, better data management tools, and better handling of very large datasets. As Lora Delwiche and Susan Slaughter point out in *The Little SAS Book A Primer: Second Revised Edition* "people who do really complex programming find they can do things with SAS that would be impossible to do with SPSS" (Delwiche and Slaughter, 2002).

The greater degree of control and flexibility in SAS come, however, at the price of ease of use. The most common complaints from new users of SAS usually relate to the application's rigid program syntax and lack of sufficiently detailed help menus. Effective use of this program involves successfully assimilating a diverse variety of information including: existing statistical knowledge, SAS program syntax, and SAS output layout. This assimilation process can be a particularly frustrating one for experienced SPSS users who may feel they need to relearn the new software from the ground-up. Likewise, new statistics students may feel overtaxed by the need to integrate learning SAS syntax into an already (for some) daunting topic.

It should by now be readily clear that SPSS's true strengths lie in its ease-of-use and instructive documentation. Since almost every modern computer user is familiar and comfortable with some sort of menu-driven GUI (Windows, Mac OS, Linux,…), SPSS's menu layout and screens have an intuitive appeal and readily comprehensible appearance. In addition, the software's opening screen, the Data Editor (which displays the variables as columns and observations as rows), is visually comparable to the user interface employed in Microsoft's widely used Excel software. Since SPSS's interface is based on visual metaphors common to all modern computing (files and folders, columns and rows, menus and checkboxes), users tend to feel more welcome in the environment and are consequently less easily frustrated and less willing to switch other programs except by necessity.

One, unarguable, limitation of the SPSS package is the limited set of procedures and tests available to the user. The SPSS system was originally developed with the social sciences in mind and thus has always favored analytic techniques common to these disciplines. As this software is targeted at a social sciences audience, it necessarily comprises a more circumscribed set of procedures than SAS whose intended audience is considerably broader. While this contingent of tests has grown in recent years to encompass techniques employed in allied disciplines there still remains a significant gap between the two applications.

One point of contention in the endless debate over the strengths and weaknesses of the SPSS system (and the SAS ANALYST Application for that matter) is the robustness of its analytic procedures and the problems this can create for new statistics students. While novice SAS students are limited in what they are able to do and the complexity of the analyses they are able to run by complexity of the code required, new users to SPSS may simply select a dataset and start requesting tests. Though both packages contain crosschecks to prevent inappropriate testing and post-hoc warnings alerting the user to potential problems with their results, the sheer simplicity of testing in SPSS can lead to students overwhelming themselves in useless output. It is the opinion of the authors that while both packages have strengths and weaknesses, students educated on either system will never be as well prepared or conversant in statistics as students educated on both. While veteran users of either system are better able to migrate across applications than new students, the steep learning curve for new users of either system (SAS in particular) present needless barriers to collaboration. Therefore, we propose a method that will (a) capitalize upon the strengths of both applications, (b) capitalize on the veteran users existing knowledge to speed migration and collaboration, and (c) improve the depth and applicability of learned statistical knowledge among new students by helping them be software-independent.

MOVING DATA FILES

While the authors will not get into the "nuts and bolts" of data management per se, a word regarding importing, exporting and exchanging data between packages seems in order. The table below shows the SAS and SPSS equivalents. The SAS column is based on the Import/Export Wizard (or PROC IMPORT, PROC EXPORT) while the SPSS column is based on the FILE drop down menu, followed by clicking OPEN, then DATA, and in the OPEN FILE window clicking on the FILES OF TYPE box and selecting the appropriate data source.

SPSS version 11.5 seems to have an advantage over SAS version 8.2 in the sense that one can import and export SAS datasets directly with SPSS. Note that formatted variables are not converted to value labels in SPSS when you import the data, but are saved as a *.sas file along with the SAS dataset upon exporting a SAS dataset from SPSS. Conversely, SAS version 8.2 allows you to use an SPSS file directly (i.e., avoiding the Import/Export Wizard) by invoking the SPSS engine option in the LIBNAME statement, ANALYST Application, or PROC CONVERT procedure. For this to work effectively the SPSS file must, however, be saved as an SPSS Portable file (*.por).

The take home message here is that either package will read the other's data structures either directly or with a little coaxing or creative programming. Once the data are in an analyzable format, the two packages look more similar than not, especially if one compares SAS's ANALYST Application to SPSS's standard menu-driven interface.

Table 2. Data Import and Export in SAS and SPSS

SAS (via Import/Export Wizard)**	**SPSS (via Open File Drop Down)**
Microsoft Excel 97, 2000, 2002 (*.xls)	Microsoft Excel 97, 2000, 2002 (*.xls)
Microsoft Excel 5 or 95 (*.xls)	Microsoft Excel 5 or 95 (*.xls)
Microsoft Excel 4 (*.xls)	Microsoft Excel 4 (*.xls)
Microsoft Access 2000 or 2002 (*.mdb)	Microsoft Access 2000 or 2002 (*.mdb)
Microsoft Access 97 (*.mdb)	Microsoft Access 97 (*.mdb)
DBase File (*.dbf)	DBase File (*.dbf)
Lotus 1-2-3 (*.wk1)	Lotus 1-2-3 (*.wk1)
Lotus 1-2-3 (*.wk3)	Lotus 1-2-3 (*.wk3)
Lotus 1-2-3 (*.wk4)	Lotus 1-2-3 (*.wk4)
Delimited File (*.*)	Delimited File (*.*)
Comma separated values (*.csv)	Comma separated values (*.csv)
Tab delimited files (*.txt)	Tab delimited files (*.txt)
	SPSS (*.sav)
	SPSS/PC+ (*.sys)
	SYSTAT (*.sys)
	SYSTAT (*.syd)
	SPSS Portable (*.por)
	SYLK (*.slk)
	SAS Long File name (*.sas7bdat)
	SAS Short File name (*.sd7)
	SAS v6 for Windows (*.sd2)
	SAS v6 for UNIX (*.ssd01)
	SAS Transport (*.xpt)
	Data (*.dat)

** SAS is able to read SPSS (*.por) via the LIBNAME statement using the SPSS Engine option. SAS can also read the Long, Short, v6, and transport SAS files via LIBNAME statement using the appropriate SAS Engine.

DESCRIPTIVE STATISTICS

The generation of measures of frequency (value table, counts, sums,…), central tendency (mean, median,…), and distribution (variance, standard deviation, normality measures,…), hereafter collectively termed "descriptive statistics", is one of the most fundamental tasks users of any analytic package will need to accomplish. These varieties of procedure are particularly well suited to new users (or users migrating from another system) since they are easy to run, have comparable outputs, and are easy to map between systems. In addition, since all users will invariably have to conduct this sort of analysis before proceeding to more complex hypothesis testing, this makes an ideal starting point for our discussion.

While both SAS and SPSS provide a multitude of ways to generate descriptive statistics, both on their own and as

add-ons to analytic procedures, we will focus on the similarities between four procedures (two SAS and two SPSS) The procedures used to do generate these statistics are SAS's MEANS and UNIVARIATE commands and SPSS's FREQUENCIES and DESCRIPTIVES commands.
The SAS UNIVARIATE and SPSS FREQUENCIES procedures are both intended to provide in-depth summary information regarding one or more variables in the active dataset. In contrast, the SAS MEANS and SPSS DESCRIPTIVES commands are better suited to side-by-side comparison of variables. SAS, as is typical, provides a broader array of statistics and is easily customized to produce more statistics than shown by default. In contrast, SPSS, though it generates fewer measures, does so with minimal additional user input.
We could make similar comparisons for bivariate statistics (Chi-Square, T-test, etc) between the two packages, but have chosen to skip over these analyses and turn to an example of simple linear regression.

SIMPLE LINEAR REGRESSION

We will consider a simple Linear Regression test in both SAS and SPSS for comparison. In SAS we use the following syntax to predict SCORE based on AGE using the SAS dataset TRIAL.sas7bdat from Walker.

```
/* EXAMPLE OF SIMPLE LINEAR REGRESSION IN SAS VIA SYNTAX */
        LIBNAME desktop 'c:\windows\desktop';
        PROC REG DATA = desktop.trial;
        TITLE 'PROC REG FOR SCORE PREDICTED BY AGE';
        MODEL score = age;
        PLOT *age = '-';
        RUN;
```

The same analysis in SPSS could be run as a series of point and click motions:

1. Click ANALYZE, REGRESSION, LINEAR	2. Highlight SCORE in variable list
3. Click arrow to put SCORE into dependent box	4. Highlight AGE in variable list
5. Click arrow to put AGE into independent box	6. Select ENTER method, Click SAVE button
7. Select UNSTANDARDIZED RESIDUALS	8. Click CONTINUE, OK, GRAPHS, SCATTER PLOT
9. Click SIMPLE, DEFINE	10. Highlight UNSTANDARDIZED RESIDUALS
11. Click arrow & put UNSTAND RES into Y-axis box	12. Highlight AGE, Click arrow & put AGE in X-axis box

(See Appendix A for a comparison of the outputs from the SAS and SPSS Regression Procedures)

DATASET CONVERSION

Numerous tools exist to translate datasets between common file types. Some of the more popular tools include Stat/Transfer® (Circle Systems; Seattle, Washington) and DBMS/Copy® (DataFlux Corporation; Cary, NC). DataFlux is a SAS subsidiary.

CODE CONVERSION

Emacs Speaks Statistics (ESS) is an universal interface for statistical analysis, which generates program specific analytical code based on simple English statements. ESS is able to control most publicly available statistical packages including SAS and SPSS. For further information see Rossini et al. (2001) UW Working Paper Series.

ADDITIONAL SOURCES

Students of SAS and SPSS, as well as those migrating between packages, should be encouraged to investigate the many listservers that have been created to support users of these programs. Prominent sources of information include SAS-L, SPSS-L, and STAT-L listservs. A good source for additional information on these programs is found at their respective home pages (www.sas.com and www.spss.com).

CONCLUSION

The authors hope that this brief introduction to similarities and differences of SAS and SPSS has provided useful information for those who may have to make the switch or collaborate across applications. When data analysts, students, and professors remain divided into exclusive camps based on their choice of software it is their research that suffers. For this reason the authors' hope that their contribution has helped to bridge the gap between these camps and break down some of the prejudices each camp has about the others' software. The practical approach of "mapping" has been shown, in our experience, to be an effective teaching methodology that can assist both students and veteran users make the most of their analytic resources.

ACKNOWLEDGEMENTS

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 registered trademarks or trademarks of their respective companies.

SPSS® for Windows, Release 11.5, 2002. SPSS Base & all other SPSS products or services are registered US trademarks of SPSS Incorporated (Chicago, IL)

REFERENCES

- Walker, Glenn A. Common Statistical Methods for Clinical Research with SAS Examples, 2nd Edition. Cary, NC: SAS Institute Inc., 2002.
- Delwiche, Lora D. and Susan J. Slaughter. The Little SAS Book, Edition. Cary, NC: SAS Institute Inc., 2002.

CONTACT INFORMATION

Your comments and suggestions are valued and encouraged. Please contact the authors at:

Joseph J Guido, MS University of Rochester Medical Center Department of Community and Preventive Medicine Division of Social and Behavioral Medicine 601 Elmwood Avenue, Box 644 Rochester, New York 14642 Tel: (585) 273-2671 Fax: (585) 461-4532 Email: Joseph_Guido@urmc.rochester.edu	Adam B Rains, MSc University of Rochester Medical Center Department of Community and Preventive Medicine Division of Public Health Practice 601 Elmwood Avenue, Box 324 Rochester, New York 14642 Tel: (585) 275-3705 Fax: (585) 756-7656 Email: Adam_Rains@urmc.rochester.edu

Appendix A.

SPSS

Variables Entered/Removed[b]

Model	Variables Entered	Variables Removed	Method
1	AGE[a]	.	Enter

a. All requested variables entered.

b. Dependent Variable: SCORE

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.164[a]	.027	.017	28.51450

a. Predictors: (Constant), AGE

b. Dependent Variable: SCORE

ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2196.183	1	2196.183	2.701	.103[a]
	Residual	79681.527	98	813.077		
	Total	81877.710	99			

a. Predictors: (Constant), AGE

b. Dependent Variable: SCORE

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	15.581	10.547		1.477	.143	-5.350	36.512
	AGE	.392	.238	.164	1.643	.103	-.081	.865

a. Dependent Variable: SCORE

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	23.0280	43.0170	32.2700	4.70995	100
Residual	-40.6654	64.2284	.0000	28.37012	100
Std. Predicted Value	-1.962	2.282	.000	1.000	100
Std. Residual	-1.426	2.252	.000	.995	100

a. Dependent Variable: SCORE

SAS

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	2196.18275	2196.18275	2.70	0.1035
Error	98	79682	813.07681		
Corrected Total	99	81878			

Root MSE	28.51450	R-Square	0.0268
Dependent Mean	32.27000	Adj R-Sq	0.0169
Coeff Var	88.36226		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	15.58112	10.54726	1.48	0.1428
AGE	1	0.39194	0.23848	1.64	0.1035

PROC REG for Score predicted by AGE