
Statistical software programs

Steven A. Conrad, MD, PhD,* and Runhua Shi, MD, PhD†

INTRODUCTION

A number of software applications are available for storage, analysis, and presentation of data, ranging from productivity tools such as Microsoft Excel to comprehensive analysis and reporting packages such as SAS (Statistical Analysis System). Frequently, the tasks of data entry, analysis, and presentation are each performed with separate tools based on the familiarity of the user, particularly when multiple individuals are involved in individual aspects of data handling. It is thus helpful to gain an understanding of the commonly used tools, particularly with respect to their limitations and their ability to exchange data with other applications. With modern computers and powerful software, it is possible to perform analyses that were historically impractical because of computational intensity. This article reviews some of the commonly used programs with the goal of guiding the selection of applications for use in projects. Emphasis will be placed on data types, entry, and exchange, as well as some of the major features of several individual applications.

DATA TYPES

All data analysis applications assume data intended for analysis is in one of several predefined formats. Before any data collection is initiated, the experimental design should be reviewed so that the data entry application can be set up for the required data types. Some applications, such as Excel, by default are very forgiving with data entry, allowing mixed data types that the application attempts to interpret. This can lead to incompatibilities within data sets that can impair data exchange or even worse can result in analyses that are silently inaccurate. This section provides an overview of computational data types (how data are stored and handled by computer systems) and statistical data types (how data are viewed for analysis by statistical software). Further guidelines will be provided under the discussion of individual software applications.

Computational Data Types

Computer software represents and operates on numbers in 1 of 2 basic formats: integer and floating point. *Integer* values, as the name suggests, have no fractional component and take on only integral values of 0, 1, 2, 3, etc. The maximum integer value available is dependent on the computer archi-

tecture, with modern systems accommodating the ranges of $\pm 2^{31}$ ($\pm 2,147,483,648$) for 32-bit systems or $\pm 2^{63}$ ($\pm 223,372,036,854,775,808$) for 64-bit systems. It is unlikely that any analysis problem will ever approach these limits. *Floating point* includes the storage of numbers in a format similar to scientific notation, with both a number component and an exponent. The term *floating point* indicates that there is a constant number of significant digits as a result of “floating” the decimal point while correspondingly altering the exponent. Current systems use the floating point standards established by the Institute of Electrical and Electronics Engineers. The precision of the number component and the range of the exponent vary with the computer architecture, with *single precision* having approximately 7 digits of precision with exponents ranging from 10^{-38} to 10^{+38} . Although data observations will only rarely exceed this range, the limited precision can be problematic when intermediate values are generated during computations. Double precision floating point handles approximately 17 digits of precision and an exponent range of 10^{-308} to 10^{+308} , assuring both improved accuracy and precision of intermediate results during computation. Fortunately, floating point computations on essentially all computer systems default to double precision so that entry and storage of data in single precision floating point format can save space without any computational penalties. Some vendors provide software emulation of larger (128 bit) numbers such as the *decimal* type, which have a much higher precision (approximately 29 digits) but lower exponent range (10^{-28} to 10^{+28}). It should be recognized that floating point computations technically are only approximate due to the fixed precision, but the number of digits in double precision format makes this approximation of little practical consequence in most cases.

Dates and times in software applications are represented internally as a *date-time* type, that is, a binary interpretation of the date and time. Unlike floating point numbers, there is no widely accepted standard for storage of date-time values, making data interchange among systems difficult. This is usually accomplished through the conversion to a standard character format such as “2006–08–23 15:45” that can be produced and interpreted across systems. With software programs, such as SAS or SPSS, that have explicit date data types, the variables can be manipulated arithmetically to calculate time elapsed between dates, for example.

Statistical Data Types

Statistical analysis applications view data in 1 of 3 contexts. *Observational data types* include data that typically represent the dependent variables in an experimental plan. Parametric analyses expect floating point values. Integer values can be

Affiliations: * Department of Medicine, Louisiana State University Health Sciences Center, Shreveport, Louisiana; †Department of Medicine, Feist-Weiller Cancer Center, Louisiana State University Health Sciences Center, Shreveport, Louisiana.

Disclosures: Authors have nothing to disclose.

Received for publication March 11, 2009; Received in revised form April 22, 2009; Accepted for publication March 25, 2009.

substituted when they represent an appropriate substitute for floating point values. Nonparametric analyses can use either type, but integer variables should be used if the data represent true integral values. Programs that default to floating point values will require explicit designation of these variables as integers.

Classification data types are used in statistical models, such as analysis of variance and logistical regression. These data typically represent nominal or ordinal factor levels that are predetermined in the experimental design. Formal statistical programs, such as SAS and S-Plus, allow the designation of variables as factors. Traditionally, these data are often coded in numerical format (eg, 1 for male and 2 for female), but these programs allow the use of descriptive values (eg, male or female) that make data management far more intuitive and realistic. If the application supports descriptive factor levels or can import character strings representing factor levels, then taking advantage of this descriptive feature is advisable. Applications targeted for data entry, such as Access or Excel, can take advantage of features such as supporting data tables or automemorized dropdown lists, respectively.

Descriptive data types serve to annotate the values of variables and factors to facilitate interpretation of presentation data and graphs but are not variables analyzed directly. This type is represented by a textual data element that may be included as a property of the underlying variable or factor or assigned during reporting or graphing (eg, male = 1, female = 2).

MICROSOFT EXCEL AND OTHER SPREADSHEETS

Microsoft Excel is a productivity tool known as a *spreadsheet*, providing a grid of cells in which each cell can hold data values or computational formulas referencing other cells. The first commercially successful spreadsheet for personal computers (Apple) was Visicalc. Lotus 1-2-3 was a successful spreadsheet that ran on the IBM personal computer. Microsoft Excel competed with Lotus 1-2-3 and has become the currently dominant spreadsheet. Macro languages now extend spreadsheets by providing a programming infrastructure.

Because of its column and row grid orientation, Excel has become a popular platform for recording experimental data. Its computational capabilities also permit simple summary statistical calculations using built-in formulas (eg, AVGERAGE or STDEV) or more extensive statistical calculations using add-on packages (eg, Analysis ToolPak). A principal advantage of Excel is that it is a popular application distributed with the Microsoft Office suite, widely available on Windows and Mac platforms. The file format is interchangeable between these 2 platforms.

Excel has a number of limitations that users should consider when choosing an application for statistical analysis. It has an unstructured interface. The user has to indicate the purpose and role of each set of data cells in calculations,

leading to potential for error. It is not suitable for large data sets. The calculation formulas are interpreted cell by cell and are optimized for flexibility but not performance. The types of data that can be stored are limited, consisting of floating point numbers, date-time values, and character strings. It does not support true integer storage. Excel is not well suited for complex data sets, such as hierarchical data, that may require analysis at several levels (a very common scenario!). Storage of hierarchical data requires redundancy that can lead to errors occurring during data entry. A better data storage platform for this purpose would be a relational database application such as Microsoft Access or a database server (eg, Oracle or SQL Server). Spreadsheets are not well suited for complex analyses. Also, they are not easily adapted to data that require analysis by groups. A number of graph formats are provided, but customization of individual graphs is limited, and thus it is not well suited to generating publication quality graphs except in simple scenarios.

Suggestions for using Microsoft Excel are as follows:

- Avoid Excel's attempt at data type interpretation by explicitly assigning an appropriate format to each column of data.
- Define integer values as numbers without decimal points or fractions through Excel's cell formatting capabilities.
- Do not separate date and time values into separate columns if both components are required. Use a single column that defines both and time.
- For text fields it may be important to have separate columns for subfields, such as last name, first name rather than a single name field. If it were necessary to search for someone by last name, for example, this is a difficult task if first and last names are in the same column.
- Keep data columns for data only and resist placing text notes or computational results in the same columns as data.
- Keep the data type consistent within each column. This will facilitate exportation to a statistical package if needed and avoid errors in analysis.
- Leave a cell blank if that data point is missing. If data are to be exported to a statistical analysis program, that program may allow predefined values to be interpreted as missing values on import.
- Keep column names short so that they are easy to manipulate when imported into analysis or presentation software. Some packages do not allow spaces in column names, so consider removing spaces or replacing with a character such as an underscore.

In summary, Excel is perhaps best suited as a data entry tool for simple experimental designs, either for exporting to formal statistical packages or for performing exploratory analyses.

MICROSOFT ACCESS

Access is a file-based relational database application that is well suited for data entry, structured data organization, and flexible data retrieval. As a database application, a wide variety of data types can be stored, including several variants

of integers, floating point data with different levels of precision, date-time data, character data, and fixed and variable-length textual string data. Through its relational structure, supporting tables can be used to both offer available choices for classification variables and ensure that only appropriate data can be entered. Complex data organizational structures can be imposed to support data integrity. Its query capabilities allow for flexible retrieval of hierarchical data into a flat format suitable for a statistical analysis program. These queries can be stored for ease of subsequent data retrieval as the database increases in size.

Access is based on industry-standard protocols that enable many statistical analysis programs to use its data directly for analysis without having to first import it. Data entry is facilitated by the creating of data entry forms that guide data entry and validate it. Because it is file based, the database file can be transferred easily among computers, but its multiuser capabilities allow users on different computers to enter or retrieve data from a common shared database file, eliminating the need to move the file around to different users.

Access is also part of some Microsoft Office suites and thus is widely available. It does not offer any statistical analysis capabilities, so it is only suitable for data entry and reporting. However, it offers data management capabilities not available in statistical analysis applications and deserves strong consideration as a data management tool to partner with these applications.

Client-server databases, such as Oracle and Microsoft SQL Server, offer the data handling features of Access but geared for high-availability multiuser access, such as with an online Web-based database. Data can be accessed through industry-standard protocols such as ODBC, but these systems do not offer the forms capabilities of Access. It is possible, however, to use Access forms that connect to a database server for a more custom solution.

SAS SYSTEM

The SAS System (previously known as Statistical Analysis System) is a comprehensive software package available for data handling, reporting, and statistical analysis. It was originally developed for mainframe computers but has since migrated to essentially all available platforms. It couples a procedural programming language to a fourth-generation language based on an extensive library of data handling and analysis functions. Graphical user interfaces (GUIs) have been developed to assist users who want to avoid a programming language construct, but the programming language approach remains the choice of many if not most SAS users. The ability to enter data into a dataset using a spreadsheet format is supported. One of the authors has used SAS on a desktop computer to perform 30,000 analysis of variance (ANOVA) calculations of a 3-factor linear model over a total of 12 million gene expression values with only 6 lines of SAS code and a solution time of 14 seconds.

SAS is distributed as a set of approximately 30 modules that add functionality to its core software. The core software

(Base SAS) provides data handling and reporting and basic summary statistical analyses. Perhaps the most commonly used module is SAS/STAT, which provides a large number of commonly used statistical procedures. Some other examples of the many modules include SAS/INSIGHT for exploratory data analysis, SAS/IML for matrix manipulation, SAS/GRAPH for presentation graphics, and SAS/QC for quality improvement procedures.

Analyses in SAS are performed on *datasets*, which can be created programmatically or by importing from other programs. A simple data step example program to create a dataset named ONE from inline data values is as follows:

```
DATA ONE;
INPUT DRUG $ SBP;
DATALINES;
A 140
A 128. . .
B 135
B 156. . . ;
RUN;
```

where A and B designate 1 of 2 treatments and the 3-digit numbers represent systolic blood pressure (SBP).

Using the SAS Import procedure to import and generate a dataset named ONE from a sheet in an Excel workbook is as follows:

```
PROC IMPORT DATAFILE= "C:\Data.xls" OUT=ONE;
SHEET="Sheet1";
GETNAMES=YES;
RUN;
```

Different versions of SAS accept only specific versions of the MS Excel spreadsheet and generally lag somewhat behind in acceptance of new formats. For example, version 9.1.3 (most recent as of this writing) will accept Microsoft Excel 2003 format but not 2007 format. Verify compatibility before saving the spreadsheet. Because Excel allows multiple sheets within a file, the sheet name will have to be specified as indicated in the above code example.

Familiarity with how SAS expects data in its datasets will facilitate data importing when data from other applications such as Excel are imported into SAS. Most SAS procedures are built on the premise that each line in the dataset represents a single observation. Each observation may consist of more than one observed variable, but each variable appears in only one column. The observation could also include other variables, such as factor levels and grouping variables. For example, data consisting of SBPs from 3 groups to be compared by an ANOVA would look like the following:

```
Drug SBP
A 140
A 128
. . .
B 135
B 156
. . .
C 122
C 118
```

...
Notice that the observed variable SBP appears only once per entry. This is distinct from how users usually record data in a program such as Excel:

```
SBP_A SBP_B SBP_C
140 135 122
128 156 118
...
```

In this case, Excel expects the data to be in 2 or more columns that will be compared. In SAS, the GROUP variable is provided to the ANOVA procedure to indicate the treatment group. To move this latter dataset structure into SAS, the Excel data would either have to be restructured to the format in the first dataset, or a SAS data step would have to be created in which the 3 data values per row from dataset ONE were exported to 3 separate observations into a new dataset TWO as follows:

```
DATA TWO; SET ONE;
Drug = 'A'; SBP = SBP_A; OUTPUT;
Drug = 'B'; SBP = SBP_B; OUTPUT;
Drug = 'C'; SBP = SBP_C; OUTPUT;
KEEP Drug SBP;
```

Statistical analyses are performed by the specification of 1 or more procedure statements that reference a dataset. A SAS program to perform a balanced 1-way ANOVA follows, where the CLASS statement identifies the classification variable:

```
PROC ANOVA DATA=ONE;
CLASS Drug;
MODEL SBP = Drug;
```

The MODEL statement is the means of specifying statistical models to all of the SAS procedures that analyze linear and nonlinear models (eg, ANOVA, REG, GLM, and others). A MODEL statement that specifies a 2-way ANOVA with interactions is as follows:

```
MODEL SBP = Drug Condition Drug*Condition;
or equivalently:
MODEL SBP = Drug|Condition;
```

The original SAS graphing capabilities were designed for line printers (SAS preceded modern output devices!). The following procedure will produce a character-based plot of these data:

```
PROC PLOT;
PLOT SBP*Drug;
```

Replacing PLOT with GPLOT will produce a high-resolution graph to the default output device (typically the display monitor).

The advantages of SAS are numerous, so only a few will be highlighted. The data manipulation capabilities are exhaustive and supported by a 4GL language based on a comprehensive procedure library. SAS works with data directly on disk; thus, it can handle datasets as large as the disk capacity on the user's computer or network drive. The code has been extensively verified and optimized during the past 40 or more years, so it is computationally fast, yielding trustworthy results. The data analysis procedures are among

the most extensive available, offering analyses not available in most other packages. It is particularly well suited when data analysis is a recurring task because the SAS programs that are developed can be easily invoked on new data.

SAS requires an initial investment of time to learn, and it is not as intuitive as many other packages. Its graphing capabilities are also somewhat more limited than many other statistical packages.

In SAS interactive GUI mode, all basic and many comprehensive statistical analyses can be performed, in addition to graphing and reporting. However, the ability to access all procedures requires use of the programming language.

SPSS (STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES)

SPSS was developed at Stanford University in 1968 and further developed at the University of Chicago. It was so widely received that it quickly became a small business and had to separate from the university and become incorporated in 1975. Although developed initially for social scientists, the statistical capabilities of SPSS and its relative user-friendliness led to its adoption by statisticians and scientists practicing in many fields. Like SAS, SPSS has a command language syntax that allows programs to be written and saved for later use. Initial implementations were on large mainframe computers. For many years, the command language was the only way to use SPSS. The advent of personal computers dramatically changed the way statistical analysis was performed. In the mid-1980s, SPSS was the first major statistical package to be ported to personal computers; it ran under DOS, the original Microsoft operating system, and still used the command language interface. In 1992, SPSS was released in a Windows compatible format and became the first major statistical package to have a GUI. Today SPSS is available for most operating systems, including Windows, Macintosh, and Linux.

SPSS retains its command language interface capabilities, but most of its features can be used in an interactive graphical interface using extensive pull-down menus. For the interactive use, data are entered through a spreadsheet-like interface or imported from a variety of source files, such as Excel, ASCII text files, and databases via ODBC and SQL. The data entry screen has 2 options, data and variable modes. Variable mode allows the addition of labels, missing values, and data type specification (eg, date, string, categorical or floating point). Data mode displays the data in a spreadsheet format. A number of sorting, aggregation, case selection, and data transformation functions are available. Data can be saved in a special SPSS format containing all the data entry, labels, transformations, and other defined characteristics, which greatly simplify future analysis. Once data entry and data manipulations are complete, statistical procedures can be selected through a pull-down menu. For each statistical procedure chosen, a window is shown that allows variable selection and specification of various options particular to the chosen analysis. Output is immediately shown on the com-

puter screen and can be saved in a text format for future reference. SPSS also contains a powerful graphing capability that can be used with either command language or graphical interface. The graphs, usually after some customizing, are publication quality and can be exported or cut and pasted into other applications. Some of the graphs for this series of articles were created with SPSS.

An example of an ANOVA calculation in the command language follows:

```
UNIANOVA FVC BY Smoke Race WITH Age BMI
/METHOD=SSTYPE(3)
/INTERCEPT=INCLUDE
/CRITERIA=ALPHA(0.05)
/DESIGN=Age BMI Smoke Race Smoke*Race.
```

BMI indicates body mass index, and FVC indicates forced vital capacity. Although this is straightforward for the experienced statistician, the typical user would have difficulty remembering the commands, options, and syntax. The scripts corresponding to the graphical commands can be saved for duplication or modification at a future time.

For the typical user, the number and size of records that can be processed with SPSS are virtually unlimited, restricted only by the hard disk, rapid access memory, and speed of their processor. One of the authors has processed analyses with tens of thousands of large records on a Windows personal computer without problem. SPSS comes in several modules. The SPSS base module contains the most commonly used statistical tools; specialized modules can be added for different applications, such as classification trees or neural networks. Student versions are relatively inexpensive and have the same capabilities as the full SPSS has but with limitations on the number of cases that can be read (1,500). For most students, this is not a serious limitation.

In summary, SPSS is a powerful, mature statistical package that is easy to use. Although SAS is, arguably, the most widely used by professional statisticians and data analysts, SPSS at this time has a much more attractive and easy to use graphical interface. There are few common statistical procedures and data manipulation capabilities not available in SPSS. Of course, there is always the danger of easy-to-use statistical programs, not limited to SPSS, for which naive users may use the wrong analysis or misinterpret the results.

S-PLUS, 'S,' AND 'R'

The S-Plus statistical analysis package is a comprehensive application based on the 'S' language. This language is a functional and procedural programming language that incorporates an extensive library of several thousand functions for performing data manipulation, mathematical computations, statistical analysis, and graphing and is well suited to scientific and engineering computations. Complex data objects can be created and manipulated in S-Plus, and it provides support for matrix and vector operations, including symbolic processing, such as those seen in mathematical packages such as MATLAB. 'S' now fully supports an object-oriented programming paradigm, a feature not available in most statistical

packages. The software is available on common operating system platforms (Windows, Macintosh, and Linux)

S-Plus has a powerful user interface that provides an intuitive approach to data management and statistical analysis. As a result, 1-time analyses can be performed without the need to program in the 'S' language. Data are stored in datasets much like SAS, but some functions expect multiple observations in each data row. Data can be imported from a variety of sources or can be directly entered into the dataset in a spreadsheet-like manner. Functions to transform and reshape data are provided. Earlier versions operated on data residing in memory that limited the size of datasets, but current versions have large dataset capabilities by sharing data between memory and disk. S-Plus supports a number of data types, including factors, character strings, integers, single and double precision floating point, logical, date-time, and complex numbers.

The graphing capabilities in S-plus are extensive and support interactive modification. For example, graphs can be easily annotated, the axes can be moved by dragging, and plot types can be easily changed. It is simple to create multiple plots per graph and to have multiple graphs per page.

The 'S' language was developed by Bell Laboratories (now Lucent Technologies) in the 1970s and commercially licensed since the 1980s. It is currently licensed by Insightful Corporation (now TIBCO Software Inc) as part of the S-Plus package. It is an open language in that it can be easily extended with additional functions and existing functions can be modified. The 'R' language is an open source implementation of the 'S' language licensed under the GNU General Public License and is largely compatible with the 'S' language. Being open source, it is widely used, but support is not available except through public forums. S-Plus can now import packages written in 'R'.

The 'S' language follows a compact functional notation. Sample 'S' code to import data from an Excel spreadsheet and perform an ANOVA follows (assumes Excel has the data as a single observation per row with columns named SBP and Drug):

```
> dataone <- importData("a:\\examples\\bp.dat")
> aov(formula = SBP ~ Drug, data = dataone)
```

Numerous options to the functions performing analyses are available, but they default to commonly used values and thus need not be included unless different options are needed. S-Plus uses a different notation than SAS for specifying statistical models. In this example, the model specification $SBP \sim Drug$ indicates that SBP is the dependent variable and Drug the single independent variable. Polynomial regression (quadratic) would be specified as $SBP \sim Drug + Drug^2$.

A model with main effects and interaction would be specified as $SBP \sim Drug + Condition + Drug:Condition$, or equivalently $SBP \sim Drug * Condition$. Because S is a functional language, function arguments can be replaced with functions, allowing this to be expressed as a compact, single-line program:

```
> aov(formula = SBP ~ Drug, data = importData("a:\\examples\\bp.dat"))
```

S-Plus is popular because of its refined user interface and interactive graphing capabilities and is especially well suited for scientific data analysis because it incorporates extensive support for scientific functions and calculations. Several add-in modules are available for performing microarray data analysis, wavelet and signal analysis, environmental statistics, and optimization, among others. Using S-Plus from the GUI is relatively intuitive and supports most statistical tasks. Learning the 'S' language for more advanced analysis requires some investment in time.

CONCLUSION

In addition to a working knowledge of the statistical tests to be used, optimal use of data management and statistical software requires an understanding of how data are stored and manipulated, as well as the strengths and weaknesses of the analysis software in use. Each statistical software package offers advantages and disadvantages, and it is not uncommon

to use more than one package for data manipulation and analysis. Most statistical analysis packages can also be used to produce tables and graphs in addition to performing analyses. Whatever software package is used, an investment of time in learning the details of its use will yield many returns, reduce errors, and improve productivity.

REFERENCES

Included in a comprehensive list at the end of the Supplement.

Requests for reprints should be addressed to:
Steven A. Conrad, MD, PhD
Department of Medicine
Louisiana State University Health Sciences Center
1501 Kings Highway
Shreveport, LA 71103-4228
E-mail: SCONRAD@LSUHSC.EDU