Spreadsheets as Tools for Statistical Computing and Statistics Education

Erich Neuwirth[1]

[1] Department of Statistics and Decision Support Systems, University of Vienna, Austria, A-1010

Abstract. Spreadsheets are an ubiquitous program category, and we will discuss their use in statistics and statistics education on various levels, ranging from very basic examples to extremely powerful methods. Since the spreadsheet paradigm is very familiar to many potential users, using it as the interface to statistical methods can make statistics more easily accessible.

Keywords. Spreadsheets, Interfacing software, Statistical Education

1 Spreadsheets as tools

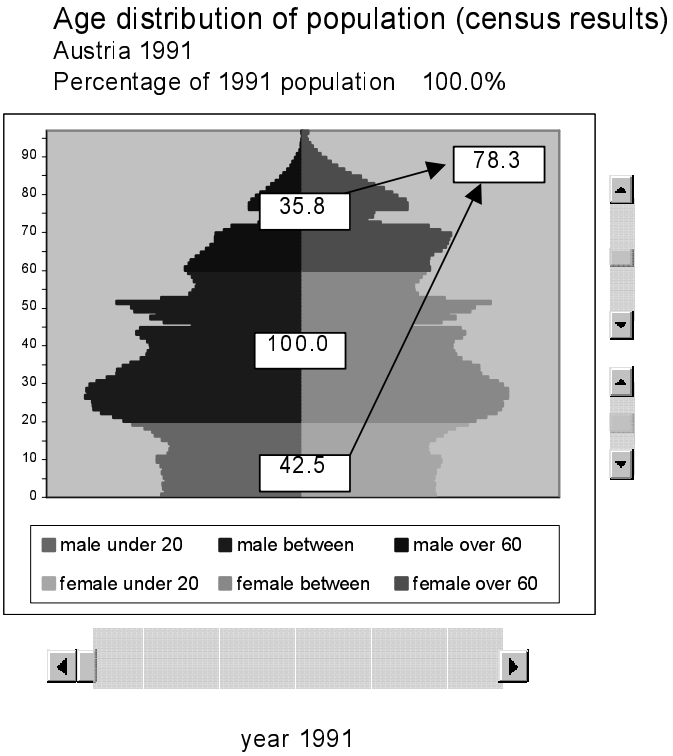
Except for word processing software, spreadsheet programs probably are the most popular software category. In a certain sense spreadsheet programs are the paradigm for numerical software for most users of desktop PCs. As a consequence, many people becoming "clients" for statistical education already know how to handle this kind of software, and therefore statistical education can build upon the pre-existing knowledge of learners.

In this paper, we will deal with different ways of using spreadsheet programs as the central tool for very different kinds of statistical work. Some examples are

- Didactical applications (including animation)
- Probability and combinatorics
- Simple database applications
- Elementary statistical methods (including cross table analysis)
- Extending spreadsheets programs through built in programming languages
- Interfacing spreadsheet programs with heavy duty statistical software (locally and over the Internet)

2 Didactical applications

Spreadsheets are quite powerful tools for numerical work and additionally have easy to use graphical tools (e.g. for plotting functions, histograms). Using these facilities, one can easily design interactive didactical applications to illustrate statistical concepts or mathematical tools for statistics. These range of complexity for these applications extends from simple to possibly rather complex. A more complex example can be seen at http://sunsite.univie.ac.at/Projects/demography.



This example calculates a prediction of the age structure of populations over time, and it presents results as a user controlled “movie”, i.e. as a graphic where the user can use the “slider” at the bottom of the graphic to move through time. As an aside, it is worth mentioning that this model has been implemented in Microsoft Excel by using “in sheet” spreadsheet techniques only, no VBA programming code was needed.

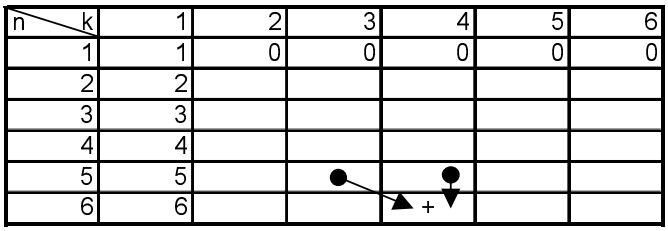
This application tries to demonstrate that spreadsheets can be used as powerful didactical devices allowing users to interact with models. It is downloadable from the WWW, and so we see that spreadsheet programs can be used as numerical engines for distributing didactical applications or “special purpose statistical labs” on the Internet. The advantage of this approach is that teachware authors do not have to learn new tools for Web programming. It is quite easy to configure a Web server in way that when the user clicks a link, the spreadsheet program on the local PC immediately starts, and so the concept of an interactive textbook can easily be implemented using software tools which are well established and for which a large knowledge base exists.

3 Probability and combinatorics

When teaching statistics, giving the learners an introduction to the basic concepts of probability and combinatorics is one of the harder tasks for teachers of statistics. One of the problems is that quite often the learners do not feel comfortable with mathematical techniques and notations, and therefore they concentrate on technical details instead of coping with the difficulties of the statistical concepts by themselves. Here, spreadsheets can play an important role in helping with easier access to concepts.

Let us illustrate this by an example:

One of the basic combinatorics problems is counting the number of combinations without repetitions. As a concrete example, we want to calculate the number of possible subsets of 4 elements out of a set of 6 elements. With some additional didactical explanations we can show that these subsets can be represented as increasing sequences of integers of length 4 with the last element of the sequence being not larger than 6. Therefore, the number of these sequences is the number of combinations we want to calculate. Calling these sequences 4-6-sequences we can show by a simple argument that their number is equal to the sum of the number of 4-5-sequences and the number of 3-5-sequences. This principle holds for any n-k-sequence (with n>1 and k>1), and therefore, when we write the numbers down in a spreadsheet table, we see that we can describe the structure of the table in the following graphical way:



This graphical representation expresses the recursive nature of the binomial coefficients in a way which is much easier to understand than the usual notational conventions, and recursion (which is a key concept in combinatorics) becomes very natural when expressed in sentences like “each cell except the ones in the first row and the first column contains the sum of the cell above and the cell above and to the left”.

This approach to the basic mathematical tools for statistics is quite helpful because most students already are used to work with spreadsheets, and so the mathematical ideas are less separated from the tools they normally use for any type of calculation.

4 Simple database applications

If statistics is used as a tool, almost any statistical project has to deal with data in tabular form. Since spreadsheets are “tables with smarts”, they are natural for dealing with data in the pre-analysis stage. Modern spreadsheets support database

work by integrating special tools for operating on data sets.

Programs like Microsoft Excel know the concept of a data list in the sense that a table with a first row consisting of text labels only and columns (all of the same length) of values of equal type below the header row usually represent names of variables and values of these variables for different cases. If one has such a structure in a spreadsheet, Excel will offer Sorting and Filtering in the Data menu item. Filtering will allow to easily construct boolean conditions for subsets directly on the sheet, and the data list interface will also allow to work with the data in a "questionnaire" view on a case by case basis in addition to the table view for the database.

When dealing with statistical data, ordering and selecting subsets are easily done with a "point and select" user interface, so basic operations on data are done without using a programming language (which is what is necessary when using any of the major statistical packages like SPSS or S-Plus).

Transforming and recoding data also can easily be accomplished by using standard spreadsheet techniques.

5 Elementary statistical methods (including cross table analysis)

Modern spreadsheet programs usually have quite a lot of statistical methods already built into them. Frequency counts, histograms, univariate and multiple regression, analysis of variance for one and two factor are part of the standard set of tools offered by these programs. So again, for not too complicated problems, spreadsheets offer quick ways of performing statistical analyses. It is well known, however, that some of the methods used in spreadsheet programs are ill behaved numerically. The matrix inversion algorithm in Excel for example does a very bad job when the matrix to be inverted is near singular, and therefore one should not rely on results from Excel when doing very sensitive analyses. In a later section we will discuss how to connect a spreadsheet program with special purpose statistics software to use numerically more reliable statistical algorithms.

The set of statistical tools offered by spreadsheet programs and especially by Excel is not as complete as a professional statistician would wish. Excel, for example, cannot do histograms with unequal interval widths.

Excel also has all the common statistical distribution functions and their inverse functions implemented as spreadsheet functions. Therefore significance calculations for statistical tests can be done directly in the spreadsheet. These functions also do have shortcomings, and they give imprecise results at the tails of the distributions. But by connecting spreadsheet programs with statistical software we can avoid the shortcomings of the all purpose spreadsheet program by integrating precise methods from specialized statistics software packages, and still use the well known interface of the spreadsheets.

The WWW offers information about using spreadsheets for statistics, e.g. see the following web sites:

- http://www.mailbase.ac.uk/lists/assume/files/ and
- and http://sunsite.univie.ac.at/Spreadsite

One of the highlights for using Excel for statistical work is Pivot Tables. This

tool allows multi-dimensional cross table analyses with an extremely comfortable user interface. Since Excel knows about statistical data lists, a contingency table can be created by just dragging labels with the names of the appropriate variables in the right places on a template for the table, and one will immediately get the corresponding tables. The “point and click” interface even goes one step further. Clicking on any cell in a contingency table will automatically create a data table with the subset of cases corresponding to the stratification variables defining the clicked cell. This tool makes it very easy to identify outliers in data sets.

6 Extending spreadsheets programs through built in programming languages

Since spreadsheet programs and especially Excel also contain a programming language which allows one to extend the functionality, it is possible to implement statistical methods not available in the standard version of the program. Modules with such code, often called addins, can be loaded into Excel and this way the spreadsheet program can become the host application for a much enhanced statistical set of tools. The power of the embedded programming language (VBA = Visual Basic for Applications in the case of Excel) is comparable to C or Pascal or FORTRAN, therefore the language itself does not restrict the choice of algorithms to be implemented. The implementation of the language, however, is through an interpreter, not a compiler, and therefore execution of the added code is not very fast. The web sites already mentioned offer links to statistical addins for Excel. The option of extending Excel’s functionality has been used by a few software companies also, there are quite a few commercial addin packages offering advanced statistical methods in Excel.

As stated previously, these packages try to build on the user’s knowledge of spreadsheet programs, thereby making statistical methods convenient to use by allowing the user to access them through the familiar interface of spreadsheet programs.

7 Interfacing spreadsheet programs with heavy duty statistical software (locally and over the Internet)

As noted already, spreadsheet programs do have their shortcomings when it comes to heavy duty statistical calculations. Methods like cluster analyses or linear models with many parameters are beyond the scope and the numerical methods offered by spreadsheets. On the other hand, it would still be convenient to keep data in a spreadsheet and just transfer the data to a statistical package and then get back the results into the spreadsheet program. Spreadsheets offer very convenient ways of formatting data, therefore bringing results of analyses back into the spreadsheet can help to reduce the work needed to produce reports quite dramatically.

The integration of spreadsheet programs and statistical packages can be done on various levels.

The most simple way is having the two programs exchange data in a format

understood by both programs. Column oriented ASCII data files are understood by most programs, so this method can be used. It is, however, rather clumsy and therefore should only be used as a last resort.

Modern operating systems allow interprocess communication, an therefore the spreadsheet program and the statistics package can communicate more directly. In such a situation, one of the two programs has to play the host for the other one. Since we already discussed the convenience of spreadsheets as the statistical user interface, it seems very reasonable to have the user interact with the spreadsheet, and have the spreadsheet program use the statistics package as its "numerical library".

This kind of connection between spreadsheets and specialized statistics software has been implemented for Microsoft Excel on one side and R and XploRe on the other side, and the conference presentation will show some demonstrations of these "software bundles in action".

R is an open source statistical package (modeled somewhat after the statistical package S) developed as a joint project of a few universities and research facilities in Europe, New Zealand, and North America, available from
http://www.ci.tuwien.ac.at/R/.

XploRe is a commercial statistical package developed under the auspices of Humboldt University of Berlin, and it is available from http://www.xplore-stat.de/.

When setting up software connections between these programs, the degree of visibility for the statistics package is of key importance. Depending on the level of expertise of the intended user, one might want to hide the details of the statistics package completely from the user. In this model, the added statistical methods could appear in additional menus in the spreadsheet program, and the user would not even notice that by using a method offered by one of the menus an additional software package (different from the software package he interacts with) is called behind the scenes.

On the other end of the spectrum, the statistical method developer needs full access to the statistics package. Therefore, the interface package connecting the spreadsheet and the statistics packages has to be scalable as far as exposing the statistics package explicitly to the user is concerned.

The basic elements for such a connection package are transfers of data between the spreadsheet and the statistics package and calling statistical routines within the statistics package from Excel. So we need data transfer in both directions, and facilities for transferring code and starting execution of code for the statistics packages from within Excel. Results of calculation in the statistics program then can be transferred back to the spreadsheet via the mechanism for transferring data.

All these mechanisms have been implemented for Microsoft Excel and R, and the packages needed are freely available from http://www.ci.tuwien.ac.at/R/ as contributed extensions. The 3 example spreadsheets in
http://www.ci.tuwien.ac.at/R/contrib/extra/excel/
demonstrate 3 different user interaction models.

In developer mode, the user is able to select rectangular areas in the spreadsheet and transfer these data to R and assign the values to a matrix object in

R. The developer can use spreadsheet regions as scratchpad for R code, write the code, and then transfer it to R and execute it in R. Getting data from R is done by creating an object with all the data needed within R (with assignment statements) and then transferring all the data in contained in the R object into a rectangular region in Excel.

In end user mode, the transfer mechanism can be hidden completely. In this mode, Excel offers some additional toolbars or menu items, the user selects data in a spreadsheet region, clicks the menu item, and then the analyses are performed and the results put in appropriate places in the spreadsheet.

There is a third mode of connecting Excel and R. Excel allows to define functions which call external functions and routines, but are executed as part of the automatic recalculation procedure of Excel. This way, one can define functions which look indistinguishable from functions built in Excel, but use the much more reliable algorithms in R. This way Excel can access distribution functions of noncentral distributions in R which are not available in Excel directly, but for the user it still will look like these functions are functions supplied by Excel itself. This mechanism has very important didactical applications, it allows us to combine "the best of both worlds". We can use the well known spreadsheet interface, and enhance the method base by the full range of algorithms supplied by the statistics package. Especially, we can shield the learners from the intricacies of the programming language of the statistics package. Of course, it is not mandatory to hide the statistics package completely. It is the strength of this approach that the visibility of the statistics package is scalable, and we can set up the system in a way close to developer mode, so the learner can see and even change the code for the statistics package.

The package we just described uses two programs, Excel and R, which run on the same computer. Additional possibilities arise when we separate these two programs. As a interface package between Excel and XploRe we have implemented a setup where Excel and XploRe reside on different machines, but XploRe still seems to be part of Excel. The implementation uses socket technology available in Microsoft Windows and in UNIX. On a UNIX server, XploRe runs as a server process listening to a port and reacting to calls from users from other machines. Technically, this can be implemented by using inetd on the UNIX server to pipe incoming XploRe calls to an XploRe process and send the results back to the calling program (in our case Excel). From the user's point of view, this is not very much different from what we have seen for the connection between R and Excel. Like in the one-computer model there is a developer mode, and there is an end user mode. The difference to the one-computer model is that data are transferred to a program running on a different computer, and that the program code transferred to a different program is also executed on a computer different from the one running Excel. Developer mode and end user mode behave exactly like in the case with two programs running on the same machine. The only difference is timing. Since the data transfer is done over the internet, transfer time may vary, and therefore one has to expect much longer answering times. Therefore, executing remote calls of statistical procedures in XploRe as part of Excel's automatic recalculation is theoretically possible, but extremely infeasible

and has not been implemented.

Separation of the Excel side and the XploRe side makes this setup a client server configuration. Therefore, XploRe can be considered as a statistical method server with Excel as the user front end. Since XploRe resides on a different machine, all the details of the code can be completely shielded from the user. The user himself does not even have to have access to the UNIX machine. This way, a complete method base can be set up on the server machine (which does not have to be UNIX, this is just a detail of the current implementation), and access to this method base is controlled by the facilities the Excel interface package (implemented as a VBA-based addin) offers the user.

Similar to the very first example in our paper such an Excel sheet can be embedded in a web page. Therefore the user can go to a web page, get the details and explanations of a statistical method in written form (like in a textbook), and then by just clicking a link on the page open an Excel sheet which connects to the remote statistics server on the internet. This spreadsheet then executes statistical programs on the internet, and transfers the results back to Excel on the user's machine.

It is important to note that the computational architecture we have been describing is quite adaptable to different needs. The one fixed point in our configurations is that the spreadsheet program is the hub for the user to interact with the statistical data and with the specialized statistics program. There is a choice of how much of the work is to be done by the spreadsheet program, and how much by the statistics program. There also is a choice for having the statistics program either reside on the same machine as the spreadsheet program, or on a different machine. Finally, there is a choice on how much of the statistics program to expose to the user, and how much convert into an integral part of the spreadsheet program (at least for the spreadsheet user).

References

Evans, I G (1997). A Note on p-values. *Teaching Statistics* **19**, 22-23.

Donald Piele (1990). *Introductory Statistics With Spreadsheets*. Reading: Addison-Wesley.

Erich Neuwirth (1990). Visualizing Correlation with Spreadsheets. *Teaching Statistics* **12**