2003 World Cultures 14(2):144-154

Using R for Cross-Cultural Research

James W. Dow
Department of Sociology and Anthropology, Oakland University, Rochester, MI 48309; dow@oakland.edu

1. ALTERNATIVES TO SPSS

Most people utilizing the Standard Cross-Cultural Sample (SCCS) data set use the SPSS statistical package. William Divale has made this much easier by putting the SCCS data into a SPSS *sav* format that includes the variable and value labels.[1] However SPSS is expensive for students. Because they cannot afford to buy their own software, they have to use it on a university computer. The personal student versions of SPSS distributed in courses do not have enough capacity to work with all the SCCS variables at once. The speed, efficiency, and power of SPSS are often not needed, and that power can get in the way of careful and thoughtful research design. In sum, SPSS is not perfect and sometimes one might want to use another statistical package.

An alternative to SPSS is open source software. R is a significant open source statistical package that has much of the capacity of SPSS along with the capacity to read the SPSS *sav* files. R is an implementation of the UNIX S language that was developed at Bell Labs. R is freely available on the Internet from *www.r-project.org* under the GNU General Public License and is available for microcomputers running Windows, Linux, and MacOS operating systems. It does not have the click-and-go features of SPSS at the moment and requires more training. It has features that SPSS does not have, such as more extensive graphics and the capacity to be programmed. This later feature is probably its greatest asset. R is not just a set of routines but a hierarchical language in which new statistical procedures can be written.

R has been used to teach elementary statistics. I had been using R for other research, and I experimented with it in an undergraduate course on anthropological methods in which cross-cultural research techniques were included. In order to make it easier for the students I developed a basic outline for its use and wrote some programs in R that automated a few of the elementary operations that a cross-cultural researcher might use.

2. READING SPSS DATA

The basic features of R are described in the Appendix *Using R*. If one is planning to experiment with R, one should read the Appendix first. An R workspace (sccs.Rdata) with all the SCCS data accompanies this issue of *World Cultures*. When R starts, it loads the command packages shown in Table 1.[2] Others are immediately available and can be loaded as needed. To see what unloaded packages are available on your computer type the command `library()`. To load a package called `newpackage` use the command `library(newpackage)`. This loads the package into the workspace and puts all its commands in the R search path. The contents and descriptions of the commands and procedures that are available in the different packages are described in the R document *http://cran.r-project.org/doc/manuals/fullrefman.pdf*.

Table 1: Default R packages

Name	Contents
base	The R Base Package
modreg	Modern Regression: Smoothing and Local Methods
mva	Classical Multivariate Analysis
ctest	Classical Tests
methods	Formal Methods and Classes

The package that contains commands to reads SPSS files is located in a package called `foreign`. It contains functions for loading data created by other statistical programs. This package contains the command `read.spss()`. When R reads an SPSS *sav* file, it will load it into an object called a data frame residing in the workspace. A data frame contains the cases in rows, the variables in columns, and all the value and variable labels in internal substructures called factors.

Let's look at a simple problem in cross-cultural research. I have prepared a dataset containing all the SCCS data up to the present and a number of functions that can be used for simple cross-cultural research. The workspace is called `sccs.RData`. It is loaded into the R workspace with the command `load("sccs.RData")`. After loading the data set, one must attach the data frame `sccs` with the command `attach(sccs)`to put all its variables in the search path so that they can be called directly without specifying in which data frame they reside. See Illustration 9 for a list of contents of `sccs.RData`.

3. FINDING CULTURES WITH SPECIFIC TRAITS

Suppose we want to find the cultures for which a particular variable has a certain value. Typing the name of a variable, for example V1648, can print the 186 values of a single variable for all the 186 cultures. If we want to see which cultures have the value "`Warfare seems to occur every year, during particular season`" we need a vector of the culture names out of which we can select those names that have this value on variable `V1648`. We do that by first looking at the variable vector of culture names (`SOCNAME`.) contained in the `sccs` data frame. We pick out the names of the cultures that have the values we are interested in by indexing this vector with a logical vector that contains a TRUE for every culture that has a value on the variable equal to the one that we are interested in. Such a vector is written as a logical expression that says that the value is equal (==) to something.

```
V1648 == "Warfare seems to occur constantly at any time
of the year"
```

Note that non-numeric values (called 'strings') need to be enclosed in quotes. The list of cultures that have this value is sent to the screen with:

```
SOCNAME[V1648 == "Warfare seems to occur constantly at
any time of the year"]
```

When a logical expression such as this one is enclosed in brackets, it indexes the preceding vector and selects only those elements for which the logical relationship is true. I am sure that this seems rather complicated, so I have written a function in R to do this. It is called `find.cname(`*`var`*`, `*`value`*`)`. *`var`* is the name of the variable you are interested in, and *`value`* is the value of that variable in the cultures you want to find. Thus you can do this in a more intuitive way by using the function:

```
find.cname(V1648, "Warfare seems to occur constantly at
any time of the year")
```

This vector indexing system and vector arithmetic is can also be used for recoding and combining variables.[3]

The function `find.cname` is quite simple. You can look at any function, or other object, simply by typing it without the parentheses. The function `find.cname` has two lines:

```
function (var, code){
SOCNAME[var == code]
}
```

You can edit functions with the command:

```
 function_name <- edit(function_name).
```

Note that the editing produces a new output that does not automatically replace the edited function unless you tell it to with the <- assignment.

You will see your screen generate a lot of program symbols when you type a command without the parentheses. What you are seeing is the underlying program that makes the command work. R is attractive to scientists because of this ability to be programmed. SAS has some of this ability, but SPSS sacrifices it to ease of use by beginners who prefer to point a mouse to get things done.

4. DEVELOPING AND TESTING CROSS-CULTURAL HYPOTHESES

The association between nominal variables should be evaluated differently than the association between ordinal variables. Ordinal variables imply a sequence that can be compared with other sequences. Nominal variables do not imply a sequence. The strength of an association between nominal variables should be a measured by its deviation from randomness, often called a test of independence. Some of the variables in the Standard Cross Cultural Sample are categories that imply an order from less to greater, so it is often possible to say that when X goes up Y also goes up, or down for that matter. However, other variables may be just nominal and not imply an order, or the association between ordinal variables may be *non-linear*, that is Y may go up, then down, or level out over the range of X. In that case, we have to test the association as a deviation from randomness.

A. Tests of Independence (nominal variables)

Take the variables V200 and V595 as an example. (See Illustration 1) The association between nominal variables can be seen in the cross tabulation of these variables. The function that creates a cross tab from *sccs* is:

```
scxt("var1", "var1")
```

in which `var1` and `var2` are the two variables of the crossstab. Remember to enclose them in quotes, because they are the names of the variables and not the variable objects themselves. You should get crosstab like the one shown in Illustration 2 (next page). If tables are larger than the width of a standard window, the columns are wrapped. The output is not as beautiful as SPSS's but it is plain text and can be copied to almost anything.

Illustration 1: Variables V200 and V595

```
200. MAJOR CROP TYPE
   28   1 = Africa
   28   2 = Circum-Mediterranean
   34   3 = East Eurasia
   31   4 = Insular Pacific
   33   5 = North America
   32   6 = South America

595. Domestic work
   94   . = Missing Data
   47   1 = Males do virtually none
   45   2 = Males do some, but mostly
         done by females
```

I wrote `scxt` to make cross-cultural crosstabs easier to produce. The source code is shown in Illustration 6. Most of the program is devoted to producing a nice-looking output. It is written in the R language itself and outputs the names of the variables, a chi-square test of independence, and a calculation of Cramer's V.[4]

Illustration 2: Cross tab output

```
           V200
"Region"
           V595
"Domestic Work"
                        V595
V200                     Males do some Males do virtually None
  South America           8             7
  North America           5            11
  Insular Pacific        11             5
  East Eurasia           10             7
  Circum-Mediterranean    4            11
  Africa                  7             6
Number of cases in table: 92
Number of factors: 2
Test for independence of all factors:
        Chisq = 8.4, df = 5, p-value = 0.1355
        Chi-squared approximation may be incorrect
[1] Cramer's V is 0.302168966262531
```

The graphics features of R are extensive. It has a crosstab plot in which the crosstab is represented by rectangles of proportional width and height. This is produced with the function `scpl("v1", "v2")`. The command `scpl(“V200”,”V595”)` produces illustration 4 (next page).

B. Rank Correlation: The Association of Ordinal Variables

Many of the variables in the SCCS have an implied order. The association between two ordinal variables can be measured by a rank correlation. Both variables need an order. Correlation measures the closeness by which variable Y will go up or down as variable X goes up or down. The correlation is positive if Y goes up when X goes up and negative when Y goes down when X goes up. If X and Y are not correlated, the correlation is zero.

For example as seen in Illustration 3 the variable V54 implies an ordering from distant to affectionate, and V679 measures the intensity of warfare. The order of the variables can be taken into account for a more sensitive test of the relationship between the variables. The null hypothesis that the Pearson correlation is zero is tested with a t-test that depends on assumptions about the distribution of the deviations from a linear relationship. A better non-parametric test of rank correlation is Kendall's tau. Non-parametric tests do not make assumptions about the distributions involved. A Kendal correlation and null-hypothesis test is produced with the command:

```
Cor.test(V54,V679, method = "kendall")
```

Illustration 3: Variables V54 and V679

```
54. ROLE OF FATHER, EARLY CHILDHOOD

    36    . = Missing Data
     4    1 = Distant
    18    2 = Rarely Close
    46    3 = Occasionally Close
    73    4 = Frequently Close
     9    5 = Regularly Close

679. Warfare or Fighting

    53    . = Missing data
    41    1 = absent or occasional or
              periodical
    92    2 = frequent or endemic
```

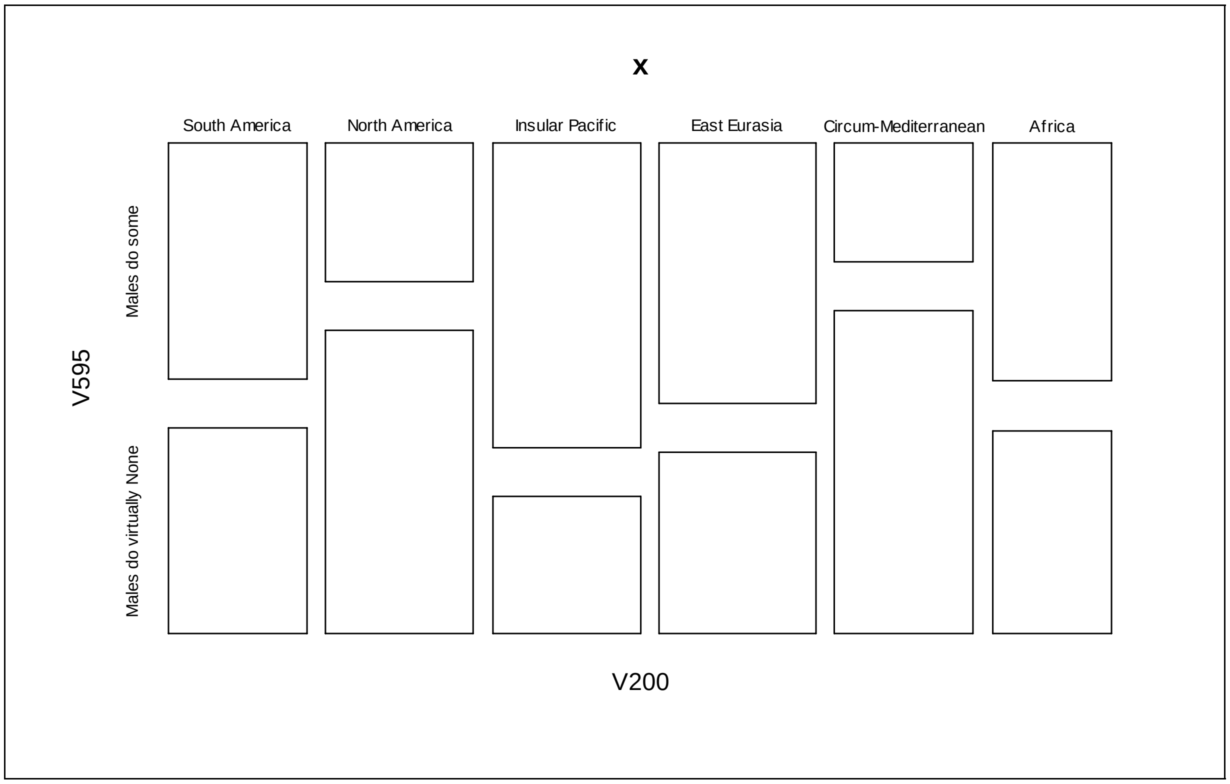


Illustration 4: Variables V200 and V595

The output is shown in Illustration 4. We can say that the result tau = -0.1489955 is significant because the p-value is less than .05. A negative correlation means that as the role of the father in early childhood increases, the amount of warfare decreases. Note that this correlation could be due to a number of things, not the least of which is that the fathers are away fighting and cannot give attention to their children. Psychological connections sometimes are not the strongest links between variables. Tau is a good choice for examining the association of cross-cultural ordinal variables. It is called a rank correlation, because it depends only on the order of the variables, which is all that we really have. No actual measure of the differences between the values has been made. Tests that make use of the ordinal property of the variable are more sensitive than a simple chi-square test of association. Another rank correlation measure and test is Spearman's rho. It can be computed with `cor.test(V54,V679, method = "spearman").`

Illustration 6: Kendall Correlation

```
        Kendall's rank correlation tau

data: V54 and V679
z = -2.3073, p-value = 0.02104
alternative hypothesis: true tau is
not equal to 0
sample estimates:
      tau
-0.1489955
```

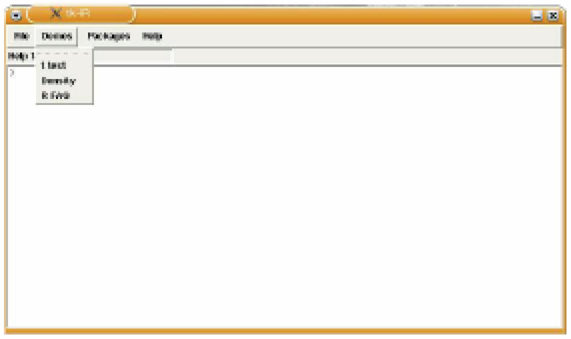
5. CONCLUSION

R is a viable alternative to SPSS for cross-cultural research. It is not as easy to use as SPSS, but it has some advantages: It is an open source program that is free, it is widely available for common microcomputer operating systems, it is programmable, and it has an excellent graphics system. The open and programmable procedures in R make possible more complex and sophisticated statistical data processing by users. This opens a door to more creative and innovative statistical analysis in cross-cultural research.

6. APPENDIX: USING R

When R is started, the user sees a window in which commands can be typed. The input and output within this window is plain text. On UNIX systems, all the editing features of an X terminal window are available in the R window. On Windows systems, the user is provided with a window that has most of these features. Another window for the graphics output is opened automatically as needed. This graphics output can be diverted to a file when copies are needed for publication. The R users group reports that click-and-go graphics user interfaces (GUIs) for R are being developed, but, so far, I have not encountered any that I can heartily recommend. When they are developed, they probably will resemble the S-Plus interface. There is a Tcl/Tk GUI interface for Linux systems (see Figure 1). It is started with the commands (1) `library(tcltk)` to load the package containing the Tcl/Tk programming and (2) `tkStartGUI()`. I have not used it much, but it makes R look more like SPSS with basic menus. As with most open-source packages there is a slow, halting, but creative development process going on at all times. A good introduction to R is found in *http://cran.r-project.org/doc/manuals/R-intro.pdf*.[5]

Figure 1: The Tcl/Tk window



Like many UNIX developed programs, R pays attention to the case of letters. When writing commands and variables, capital letters will be interpreted differently than small letters. Blanks, end of lines, and tabs are ignored by the interpreter and are used only to make the text more readable. Also, anything after a # on a line is interpreted as a comment, and no action is taken. There is no difference between commands that are written in the R language and functions. Most of the commands are written in the R language, which can refer to other functions. The object on which the command or function works is placed inside following parentheses. However, some commands do not have objects, yet the empty parentheses must still be included to show it is a command. Some commands have alternative symbols such as <- and -> for assign.

R operates in a workspace that you cannot see. `ls()` lists the objects that are in the workspace. When you exit R with the command `q()`, it will ask you if you want to save the workspace, which will be loaded the next time you start R. It is a good idea to save it. The command `save.image()` will save the workspace at any time. The workspace is kept in a file called `.RData`.

The workspace contains objects that you cannot see. Objects are commands, functions, and data. R executes the commands and functions when they are typed in the window and followed by a return, and they operate on the data. The most important types of data are numbers, character strings, vectors, and data frames, a type of matrix. You can see the value of an object at any moment by typing it on a line. You can also use an object in various commands, functions, and mathematical expressions. An object is assigned a value with the assign operator, <- or ->, which itself is a type of function. Note that this is different than the equal operator. Separate commands can be placed on a single line if they are separated by a semicolon. The current value of anything can be sent to your screen by typing its name on a line.

Illustration 7: Some Useful R Commands

`help(com):` Get information for command com. Also you can use ?com

`help.search(“x”):` Search for information on subject x

`x <- read.spss(“file”):` Read the SPSS file called file into data frame x

`b <- a or a -> b:` Assign in vale of object a to object b

`ls():` List objects in the workspace

`q():` Quit R

`save.image():` Saves the workspace in a file called .Rdata, which will be loaded next time R starts

`save.image(file=”w.Rdata”):` Saves the workspace in a file called w.Rdata

In order to load an SPSS *sav* file into the R workspace, you have to open an R window in the same directory as the SPSS file. You will have problems if you do not move your R window to the same directory (folder) as your data. You can do this in MS Windows from the file menu of R. If you start by clicking on the .RData file, you will be in the right directory. Next, you have to load the package that does the conversion with the command:

```
library(foreign).
```

We can read the entire SCSS data set into an a data frame called sccs in the R workspace with the command:

```
sccs <- read.spss("sccs.sav")
```

To make the work easier by making unique variable names available in the search path attach the data frame sccs with the command:

```
attach(sccs)
```

This puts the variable names into the R search path so that R can get them without knowing to which data frame they belong. The variables are in columns and the cases, the cultures, are in rows. To see the variables in a data frame `df` use the command `names(df)`. To see the labels for the different values of a variable `Vxxx` use the command

```
levels(Vxxx)
```

The SPSS missing data value "." is converted into the R missing data value "NA." Note that there is a difference in the way that a variable is indicated. For example, V56 is the object, the full set of data for that variable for each of the 186 cultures. "V56" in quotations is the character string that names that variable. R reacts differently to these two objects. The vector form, V56, is used in computations. The character string name is simply a string of characters that may be used in a variety of ways. Surrounding something with quotes indicates that it is a string of characters, not a tag for an object in R.

Illustration 8: Function `scxt`

```
scxt <- function(v1,v2){
varlab <- attr(sccs, "variable.labels")
# Prints a crosstab for two variables in the sccs SPSS data list.
# The variables must be in quotes because they are names.
v1nam <- varlab[v1]
v2nam <- varlab[v2]
print(v1nam)
print(v2nam)
tab <- table(c(sccs[v1], sccs[v2]))
print(tab)
suma <- summary(tab)
print(suma)
# Cramer's V
n <- as.numeric(suma["n.cases"])
X2 <- as.numeric(suma["statistic"])
k <- min(dim(tab))
V <- sqrt(X2/(n*(k-1)))
print(c("Cramer's V is", V), quote = FALSE, digits = 4)}
```

Illustration 9: Objects in the Workspace sccs.RData

Data

sccs - A data frame with the entire Standardized Cross Cultural Sample with1848 variables and 186 cultures.

Functions

Functions that do not require attachment

`scpl("v1", "v2")` - makes a table plot between variables v1 and v2.

`scxt("v1", "v2")` - makes a cross tab between variable v1 and variable v2. It also gives you the variable labels, chi-square, and Cramer's V.

`vlabel("var")` - prints the full label of the variable var. Several names can be retrieved at one time with `vlabel(c("v1", "v2", ...))`

Functions that require that sccs be first attached with the command `attach(sccs).`

`find.cname(var, "val")` - finds the cultures for which variable var takes on value `"val"`. Example: `find.cname(V1648, "Warfare seems to occur constantly at any time of the year")`

`scrc(v1, v2)` - computes Kendall's rank correlation between two ordinal variables v1 and v2.

7. NOTES

1. These SPSS *sav* files are distributed with the issues of *World Cultures*.
2. Because R is also a programming language, users can write their own packages. Many non-standard R packages can be downloaded from *http://cran.get-software.com/*
3. The `sccs` data frame expresses the variable values as string labels. These can be coerced into their original numeric format with the function `as.numeric()`. Then, these numeric values can be used in arithmetic or logical expressions to recode variables. The numeric representation of a TRUE value is 1, and the numeric representation of a FALSE value is zero. Thus, the result of an indexing expression can also be used in the recoding process. I have found it useful when working in an R workspace to make notes on how new objects are created. Otherwise, one can forget what one has done.
4. Editor’s note: If a row or column of a crosstabulation contains all zeros, the chi-square statistic, its probability, and Cramer’s V will all be reported as NA
5. General documentation for R is available at http://cran-r-project.org/doc/manuals/