ARBSRM for R

ARBSRM is an R program written for the bivariate Social Relations analysis of arbitrary dyadic data structures. It is assumed that the reader is familiar with Social Relations Analysis. If not, see David Kenny's Web page at http://davidakenny.net/kenny.htm or the book on *Dyadic Data Analysis* by Kenny, Kashy, and Cook (2006).

ARBSRM is not intended for routine Social Relations analysis. For routine analysis of round robin data, use David Kenny's SOREMO program or the TripleR package by Schonbrodt, Back, and Schmukle. For routine analysis of block data, use Kenny's BLOCKO program. It is possible to do some Social Relations analyses with conventional software. See David Kenny's Web page for more information.

ARBSRM is intended for the bivariate Social Relations analysis of non-standard dyadic data structures. It estimates the parameters of the bivariate Social Relations Model from non-standard data structures. It also yields standard errors for those parameter estimates. It is designed to analyze data from a single group.

Bond and Malloy (2007) have provided the mathematical justification for ARBSRM. They also analyze a non-standard data set with ARBSRM.

Bond wrote the program and made every effort to debug it. However, Bond cannot guarantee that it is bug-free.

Get a copy of the ARBSRM program from Malloy and save it as a text file. Get also a copy of the data.csv file. Open ARBSRM and you will see the program. At the beginning of the program is a command to read the data.csv file. Note that in data.csv missing data are represented by the value -999. Note that data on the variable X is treated as missing when the corresponding data on Y is missing (and vice versa).

You will need to create your own data file. Create it in EXCEL, if you like. First, put in all of your data for the X variable. Put in a complete round robin – filling in whatever is absent from such a design with missing values. Then put in all of your data for the Y variable, using the same convention. Rows function as actors; columns function as partners. Feel free to use a missing value other than -999. If you want to use another value, change the second statement in ARBSRM. Assign your missing value to MISS. Be sure to code all values on the diagonal of each matrix as missing. Once you have input all your data, save it as a csv file. Change the first line of ARBSRM save that it READs your data.

ARBSRM can take a long time to run. The bivariate parameter estimates are trivial to compute, but the standard errors are not. It may not be feasible to get standard errors for huge data sets, and for extremely small data sets (like the block design with N=4) these standard errors cannot be computed. In such cases, you should suppress the computation of standard errors. The third statement in ARBSRM is now SERROR=1. To suppress computation of standard errors, change that statement to SERROR=0.

ARBSRM is intended for bivariate Social Relations analysis. If you want to do a univariate analysis, simply input data from your variable twice – once as X and once as Y.

With the default option SERROR=1, ARBSRM yields standard errors for univariate and bivariate parameter estimates.

ARBSRM provides output in matrix form. Let A, B, and C be actor, partner, and relationship effects on X. Let F,G, and H be actor, partner, and relationship effects on Y. ARBSRM output follows the following format:

Var(A)	Var(F)	Cov(A,F)
Var(B)	Var(G)	Cov(B,G)
Cov(A,B)	Cov(F,G)	Cov(A,G)
Cov(A,B)	Cov(F,G)	Cov(B,F)
Var(C)	Var(H)	Var(C,H)
Cov(C,C')	Cov(H,H')	Cov(C,H')

The first matrix output is a matrix of parameter estimates. The second matrix is a matrix of estimated standard errors for those parameter estimates.

ARBSRM is intended for the Social Relations analysis of data from a single group. Although it could be used to analyze data from multiple groups, the procedure would be cumbersome. Analyses would need to be done on one group at a time and then pooled. To pool univariate and bivariate SRM parameter estimates, simply average them across groups. The procedure for pooling standard errors is slightly more complex. Suppose there are *g* groups. Get the standard errors for each group, and convert each standard error to a variance by squaring it. Take the average of the variances across groups. Divide that average by *g*. The square root of that value is the pooled standard error. If the groups differ radically in size, you may want to take weighted averages.