

TripleR: An R package for social relations analyses based on round-robin designs

Felix D. Schönbrodt · Mitja D. Back ·
Stefan C. Schmukle

Published online: 10 September 2011
© Psychonomic Society, Inc. 2011

Abstract In this article, we present TripleR, an R package for the calculation of social relations analyses (Kenny, 1994) based on round-robin designs. The scope of existing software solutions is ported to R and enhanced with previously unimplemented methods of significance testing in single groups (Lashley & Bond, 1997) and handling of missing values. The package requires only minimal knowledge of R, and results can be exported for subsequent analyses to other software packages. We demonstrate the use of TripleR with several didactic examples.

Keywords Social relations analyses · Social relations model · Round-robin · R · Software

Interpersonal perceptions (e.g., liking) and behaviors (e.g., smiling) are complex and multiply determined social phenomena. The social relations model (SRM; Back & Kenny, 2010; Kenny, 1994; Kenny, Kashy, & Cook, 2006; Kenny & La Voie, 1984) is one way to understand these complexities. It is based on the analysis of interpersonal perceptions or social behaviors within dyads. The two people of a dyadic

measurement are usually denoted as the *actor* and the *partner*. The actor provides the measurement, and the partner is the other person. The terms *actor* and *partner* are generic terms, and other terms can be used in different contexts. For instance, in interpersonal perception research, the terms *perceiver* and *target* are more commonly used. In line with this terminology, we use the more general terms *actors* and *partners* if we describe social relations analyses (SRAs) on the conceptual level or if the investigated phenomenon is a behavior. If the investigated phenomenon is an interpersonal perception, the dyadic members are called *perceivers* and *targets*.

The SRM accounts for the interdependent nature of social relations and distinguishes conceptually distinct components necessarily entailed in interpersonal perceptions and social behaviors. On the basis of this componential approach, the perception that perceiver A likes target B can, for instance, be decomposed into A's general tendency to like others (a perceiver effect for liking attributed to A), B's general tendency to be liked by others (a target effect for liking attributed to B), and the unique liking of B by A (a relationship effect for liking from A to B). Similarly, actor A smiling at partner B can be decomposed into A's general tendency to smile (an actor effect for smiling attributed to A), B's general tendency to be smiled at (a partner effect attributed to B), and A's unique smiling toward B (a relationship effect for smiling from A to B). The SRM allows for a differentiated look at many important psychological topics, such as attraction, persuasion, helping, aggression, and cooperation, to name just a few (for overviews, see Back, Baumert, Denissen, Hartung, Penke, Schmukle & Wrzus, 2011a; Back & Kenny, 2010).

Statistical analyses based on the SRM (called *social relations analyses*, SRAs) cannot be conducted, however, by using an individual-focused data collection and traditional methods of data analysis. The most comprehensive

F. D. Schönbrodt (✉)
Department of Psychology,
Ludwig-Maximilians-University Munich,
Leopoldstr. 13,
80802 Munich, Germany
e-mail: felix.schoenbrodt@psy.lmu.de

M. D. Back
Johannes Gutenberg University Mainz,
Mainz, Germany

S. C. Schmukle
University of Leipzig,
Leipzig, Germany

approach for SRAs are round-robin designs in which each person in a group judges/interacts with every other person in that group (Kenny, 1990, 1994). The conceptual foundations for handling round-robin data were laid about 20 years ago (Kenny, 1994; Kenny & La Voie, 1984; Malloy & Kenny, 1986). In the 1990s, Kenny provided the FORTRAN program SOREMO, which was the first applicable solution for calculating SRAs, and fostered a variety of research articles and programs (see <http://davidakenny.net/doc/srmbiblio.pdf> for a bibliography of published articles using the SRM).

There is a need for an up-to-date software solution that improves usability and flexibility and provides new developments in statistical computing. Despite the growing popularity of the model, many psychologists still refrain from employing this approach, a hesitancy that may be attributed also to the lack of user-friendly software solutions that allow for convenient handling of the data (Back & Kenny, 2010). Therefore, we decided to implement SRAs in an open-source package for the free *R Environment for Statistical Computing* (R Development Core Team, 2008). The package is called TripleR (Schmukle, Schönbrodt & Back, 2011) and provides all standard functions for calculating round-robin analyses in R. Furthermore, TripleR extends existing software solutions by implementing new methods for significance testing and handling of missing values.

Social relations analyses

The most comprehensive approach for SRAs are data in round-robin format. In round-robin designs, participants from a group interact with or judge every other member of this group.

Variance components

The SRM assumes that dyadic phenomena (e.g., social behaviors, interpersonal perceptions) are composed of three independent components. In the case of A's interpersonal perception "I like B very much," for instance, these components would be (1) A's general tendency to like other people (perceiver effect; i.e., Is A a "liker"?), (2) B's general tendency to be liked by others (target effect; i.e., Is B likable?), and (3) A's unique perception of B *beyond these two general tendencies* (relationship effect). Whereas these components are present in every dyadic phenomenon, a unique feature of the SRM is the ability to separate these effects statistically. On the basis of the dissociated effects, it is possible to compute variance components that indicate the extent to which each source of variance (e.g., perceiver variance, target variance, and relationship variance) contributes to the overall variance. This means that one can, for example,

answer the following questions: How much variance of an interpersonal perception can be attributed to the perceiver? How much can be attributed to the target? And how much can be attributed to the unique perception of a specific other person? In the case of liking, about 10%–20% of the variance can be attributed to perceivers, about the same amount to targets, and about 30%–40% to the unique perception (Kenny, 1994). These percentages vary depending on the level of acquaintance (e.g., first encounters vs. long-time friends).

Any type of measurement contains some amount of error variance. If only one indicator of a construct is measured, SRA cannot disentangle relationship variance from error variance (i.e., the estimate for relationship variance also contains all error variance). If multiple indicators for a latent construct are assessed, these two sources of variance can be separated, resulting in a total of four variance components summing up to the overall variance—for example, perceiver, target, relationship, and error variance for interpersonal perceptions.

Within-construct correlations

Beyond variance partitioning, two correlations can be calculated within one single construct. These correlations cannot be calculated in other designs. For our example of liking judgments, one can analyze (1) the perceiver–target correlation (also called *generalized reciprocity*)—Does a bias in the perception of others correlate with the way perceivers are seen by others? A positive generalized reciprocity in our liking example would mean that "likers" are generally liked by others—and (2) the relationship correlation (also called *dyadic reciprocity*): The two members' relationship effects of each dyad are correlated. A positive dyadic reciprocity would mean that if A uniquely likes B, B also uniquely likes A.

For liking, one typically finds mixed results for generalized reciprocity with low positive correlations, on average. For dyadic reciprocity, however, one finds robust positive correlations ranging from .26 for short-term acquaintances to .61 for long-term acquaintances (Kenny, 1994).

Between-construct correlations: Bivariate SRAs

SRAs are also defined for the bivariate relations of two round-robin variables (e.g., variable 1 = liking; variable 2 = metaperception of being liked—i.e., assuming that others like oneself or not). In this case, a variance decomposition of each variable is performed, and six additional correlations between these variance components are computed: perceiver_{V1}–perceiver_{V2} correlation (*perceiver-assumed reciprocity*; Do likers assume they are liked more?), target_{V1}–target_{V2} correlation (*generalized assumed reciprocity*; Are people who are liked assumed to like others more?), perceiver_{V1}–target_{V2} correlation (*perceiver meta-*

accuracy; Do people know who is a liker?), target_{V1}–perceiver_{V2} correlation (*generalized meta-accuracy*; Do people know how much they are liked by others?), intrapersonal relationship correlation (*dyadic assumed reciprocity*; Do people assume they are uniquely liked by those they uniquely like?), and interpersonal relationship correlation (*dyadic meta-accuracy*; Do people know who uniquely likes them?). In contrast to the within-construct correlations described above, these correlations are between two different constructs.

To extend the example given above, one can also combine a behavioral variable (e.g., variable 1 = smiling) and a perceptual variable (e.g., variable 2 = liking). In this case, bivariate SRAs would result in an actor_{V1}–perceiver_{V2} correlation (Do smilers like others more?), a partner_{V1}–target_{V2} correlation (Are people who are liked smiled at more?), an actor_{V1}–target_{V2}–correlation (Are smilers more liked by others?), a partner_{V1}–perceiver_{V2} correlation (Are likers smiled at more?), an intrapersonal relationship correlation (Do people uniquely like those they uniquely smile at?), and an interpersonal relationship correlation (Do people uniquely like those who uniquely smile at them?). By combining any two perceptual and/or behavioral variables, bivariate SRAs provide “a dizzying array of possible correlations that can give rise to novel and interesting results” (Back & Kenny, 2010, p. 864; for details on these bivariate covariances, see Back & Kenny, 2010; Kenny, 1994).

SRM effects

Variance components and covariances are estimated on the group level. Furthermore, SRAs can also provide estimates of the perceiver and target effects for each individual, as well as two relationship effects for each dyad. These effects can then be saved and used in subsequent analyses. For example, they can be correlated with external variables such as personality scales or demographic variables. One could, for example, analyze whether more extraverted (or younger) people generally smile more (the correlation between extraversion or age with the actor effect of smiling) or tend to be smiled at more often (the correlation between extraversion or age with the partner effect of smiling). One could also predict unique smiling (the relationship effect of smiling) between dyad members by assessing their similarity with respect to age or extraversion.

Method

Statistical models

SRAs can be done with three different approaches (Kenny et al., 2006): the method-of-moments approach (Kenny, 1994), the multilevel modeling (MLM) approach (Snijders

& Kenny, 1999), or structural equation models (SEMs; Olsen & Kenny, 2006).¹ Each approach has several advantages and drawbacks.

The main advantages of MLM and SEM, in contrast to the method of moments, are that missing values can be handled without imputation and that constraints can be placed upon certain (co)variances. Furthermore, fixed effects such as age or gender can be directly included in the model (see Kenny & Livi, 2009). A major drawback of these methods, however, is that, technically, they are rather complicated to set up (see Kenny, 2007; Kenny & Kashy, 2010; Kenny & Livi, 2009). They usually involve the creation of a large number of dummy variables or paths and, therefore, are tedious and error prone, and the calculation is often very time consuming. Furthermore, these analyses are not possible in many statistical programs. For example, for modeling the actor–partner covariance in MLM, it is required that the software allows the placing of constraints on the variance-covariance matrix. This ability is documented for MLwiN and SAS,² but currently is not possible using SPSS and HLM (Kenny & Livi, 2009). Additionally, for some statistical programs such as SPSS, the dyadic covariance has to be assumed to be positive, which can lead to incorrect results (Kenny, 2007). Finally, the SEM and MLM methods show even more complexities when bivariate or latent analyses are to be performed. Even if these multivariate analyses could be handled with some of the available statistical programs (see Kenny, 2007, for an example), little or no documentation has been presented so far about the necessary steps to set up the model properly.

TripleR, like the software SOREMO, implements the method of moments. The main advantage of this method is that formulas for estimating all possible SRM variances and covariances, including bivariate and latent analyses, have been developed (Kenny, 1994). A drawback is that the estimation method can produce out-of-range estimates (e.g., negative variances, correlations <1 or >1). These out-of-range estimates usually are set to their respective boundaries.

SOREMO (along with the Windows-based program WinSoReMo) allows for multiple analytic variants and gives all necessary outputs of an SRA in one run. It has, however, some drawbacks. First, it is restricted regarding

¹ An alternative for SRAs not discussed here is Bayesian statistics (Gill & Swartz, 2007). Very recently, Lüdtke, Robitzsch, Kenny and Trautwein (2011) introduced a particularly interesting and flexible Bayesian SRM approach. Future analyses and applications will be necessary to fully evaluate the utility of these promising Bayesian approaches.

² David Kenny provides a SAS macro that automatically performs an SRA on a data set and provides text output of the results (<http://davidakenny.net/dtt/srm.htm>).

Table 1 Feature comparison of different software solutions/calculation approaches for round-robin social relations analyses

Feature	Structural Equation Modeling ^a	Multilevel Modeling ^a	SOREMO ^b	TripleR 1.1
Variance decomposition	✓	✓	✓	✓
Modeling of relationship covariance	✓	✓	✓	✓
Modeling of actor–partner covariance	✓	(✓) ^c	✓	✓
Significance tests for multiple round-robin groups	✓	✓	✓	✓
Significance tests for a single round-robin group	No	✓	Only jackknife as significance test (not recommended)	✓
Estimation of actor, partner, and relationship effects	Not documented	Not documented	✓	✓
Reliabilities of actor, partner, and relationship effects	Not available	Not available	✓	✓
Maximum number of groups	Unlimited	Unlimited	999	Unlimited
Maximum number of participants within each group	Unlimited	Unlimited	25	Unlimited
Bivariate analyses	Not documented	Not documented ^d	✓	✓
Latent analyses	Not documented	Not documented	✓	✓
Missing values	✓	✓	no	✓

Note. "Not documented" means that, to the authors' knowledge, there is no resource that documents how to conduct the appropriate statistical analysis in this cell. It does not mean that the analysis is impossible with this statistical approach in general or that nobody has done this analysis before. "Not available" means that, to the authors' knowledge, these statistics are not directly provided by the software. This means that one could calculate these indices by collecting different outputs (e.g., variances) and putting them externally into a formula.

^a Documentation on how to use SEM and MLM software for SRM analyses can be found in Kenny and Livi (2009) and Kenny and Kashy (2010).

^b SOREMO can be obtained from D. A. Kenny's Web site (<http://davidakenny.net/srm/srmp.htm>).

^c Only possible for MLM with dummy variables (it is required that the software allows the user to place constraints on the variance-covariance matrix; this ability is documented for MLwiN and SAS; currently not possible with SPSS and HLM; Kenny & Kashy, 2010, and Kenny & Livi, 2009).

^d Used in Kenny (2007).

the number of participants per group ($n \leq 25$). Second, there is no way to handle missing data. Third, if there is only one group, SOREMO applies the jackknife method of significance testing, which is "extremely conservative and should not be used" (Kenny et al., 2006, p. 213). And finally, SOREMO is very demanding concerning data preparation and data formatting. Data have to be cleaned and rearranged into a nonstandard file format. Moreover, for each set of analyses a complex setup file has to be programmed, including an input format record written in FORTRAN.

In contrast to SOREMO, TripleR can handle missing values (see the Handling of Missing Values section below) and an unlimited number of participants within each group and can perform within-group t tests, which are recommended if there is only one round-robin group (Kenny et al., 2006; Lashley & Bond, 1997) (see the Tests of Statistical Significance section below). Most important, TripleR uses standard data sets that can be flexibly rearranged and one function with an intuitive brief syntax for all necessary SRAs. A systematic comparison between these two software programs and other approaches (SEM, MLM) is provided in Table 1.

Handling of missing values

TripleR handles missing values by implementing the following three steps. First, participants who have too few data points are removed both as actors and as partners. Completely missing rows occur if participants do not rate anybody, for example, because they were missing during data collection; missing columns can occur if participants cannot rate an unknown person. With a parameter (*minData*), this step can be adjusted to be more or less restrictive: *minData* defines the minimum number of data points outside the diagonal that have to be present in each row or column. For example, the definition can be that at least two measurements (*minData* = 2) should be present in each row or column.

Second, missing values outside the diagonal are imputed as the average of the corresponding row and column means.³ On the basis of these imputed matrices, actor,

³ We also tried several alternative procedures for imputation, such as iterative imputation procedures, which take care of the changing row and column means after each imputation. However, none of these more complicated procedures yielded appreciably better results than did the final procedure described above.

partner, and relationship effects are computed. Subsequently, relationship effects that were missing in the original data set are set to missing values again.

Third, in the case of multiple variables (i.e., in latent and bivariate analyses), participants who were excluded from one of the variables are excluded from all other variables to ensure a consistent data set.

The imputation procedure assumes a relationship effect of zero for the missing cells (they are imputed as consisting only of the actor and partner effects). Hence, with an increasing proportion of missing values, the relationship variance will be underestimated, and the actor and partner variances will be overestimated. To test the impact of missing values both on the estimation of variances and on the actor and partner effects, we ran several simulations. We took complete round-robin matrices with different numbers of participants and imposed an increasing number of missing values (missing completely at random). We then calculated the SRM with the procedure for missing values

as described above and compared the resulting values with the known true values from the complete matrices. We tested seven different group sizes ($n = 4, 5, 6, 8, 10, 15,$ and 20). For each group size, 20 different data sets were generated. Within each of these data sets, an increasing amount of randomly selected missing values (3%, 5%, 10%, 15%, and 20%) was imposed, and the whole procedure was repeated 10 times with different configurations of missing values; thus, we calculated a total of 7,000 SRAs.

The results for this simulation can be seen in Figs. 1 and 2. Figure 1 shows the deviations from the true values for each of the three standardized variance components. Variance estimates are more biased in smaller groups and with more missing values. However, even with a great proportion of missing values, there is only a small systematic bias in the estimation of standardized variance components ($< 5\%$).

Figure 2 shows the correlations of the true actor and partner effects with the respective effects from the reduced

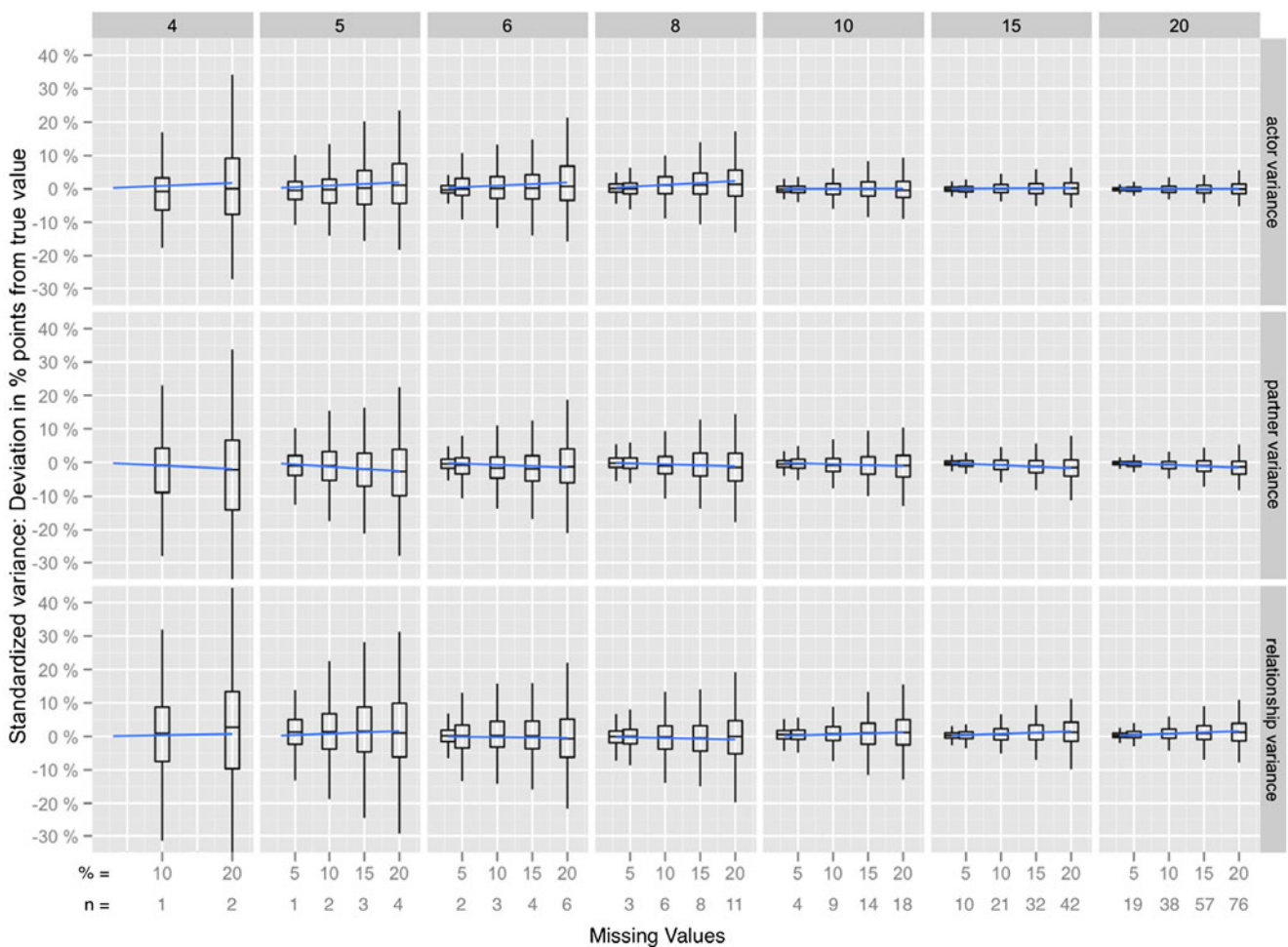


Fig. 1 Simulation results: Deviations from the true values of variance components. The horizontal axis denotes absolute and relative numbers of missing values. Distributions of deviations are represented by boxplots and a regression line

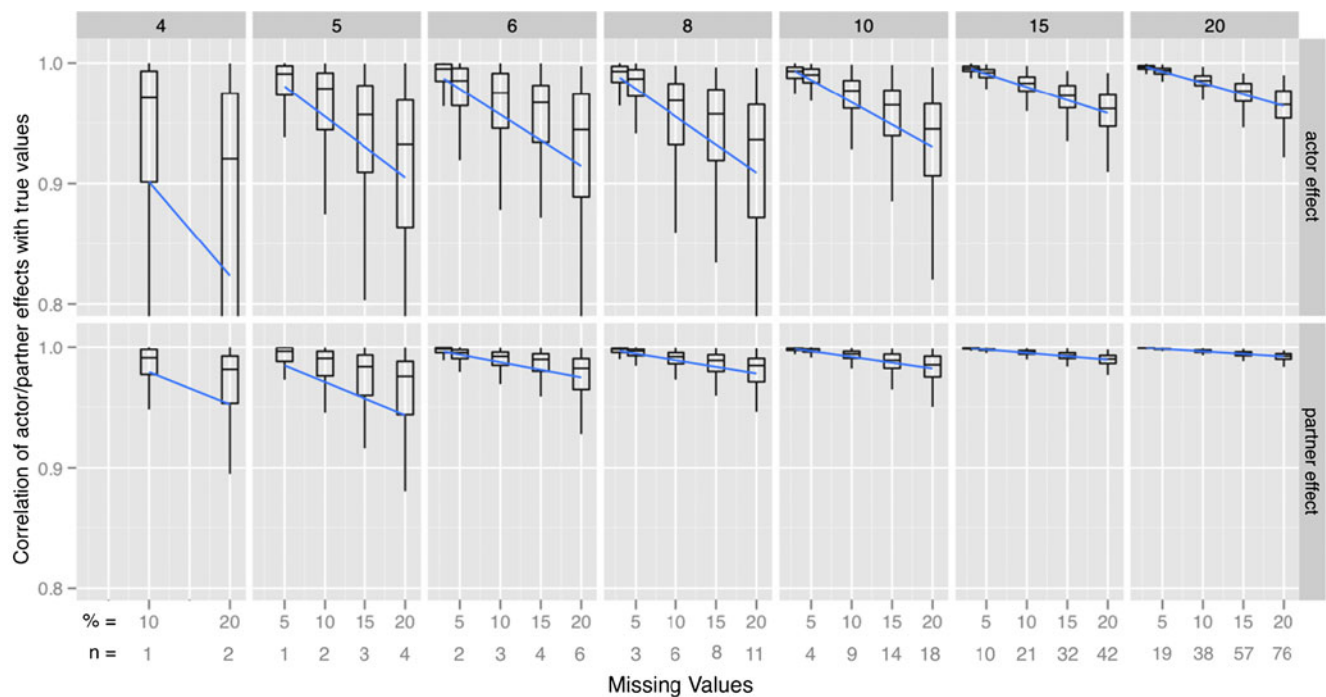


Fig. 2 Simulation results: Correlations of actor and partner effects with their true values. The horizontal axis denotes absolute and relative numbers of missing values. Distributions of correlations are represented by boxplots and a regression line

data set. These computations seem to be relatively robust, since correlations are well above .90 if not too many missing values are present.

An inspection of these results leads us to the conclusion that relatively unbiased results can be expected if the number of missing values does not exceed 1 in groups with $n = 4$, ≤ 2 missing values in groups of 5, ≤ 4 missing values in groups of 6, ≤ 6 missing values in groups of 7, ≤ 8 missing values in groups of 8, and less than 10 missing values in groups of 10. For large groups with $n > 10$, even 20% and more missing values can be present.

Tests of statistical significance

Two types of tests for significance can be applied to variance components of SRAs: (1) a between-groups t test in the case of multiple round-robin groups and (2) a within-group t test for the analysis of a large single round-robin group. In the first case, variance components are computed within each single group. Overall variance components are then calculated as the weighted average across groups (weighted by group size - 1) and tested against zero with a weighted one-sample t test. In the second case of a single round-robin group, TripleR provides the t test introduced by Bond and Lashley (1996) and Lashley and Bond (1997), which allows for the calculation of a p value within a single round-robin group. By default, TripleR uses the between-

groups t test as soon as three or more groups are present. However, common sense is needed to judge whether a t test for very few groups is sensible. In the case of very few large groups, we strongly recommend judging p values on the basis of the within-group t test of each single group, since this approach is more robust and has much more power than a between-groups t test with very few degrees of freedom.

The TripleR package

In the remainder of the article, we explain how to use the package and how to interpret the output in different kinds of analyses.⁴ The very first steps of setting up the R environment can be found in the Appendix. Since the

⁴ The TripleR package contains a built-in PDF file with a more detailed step-by-step tutorial on how to use the package. The tutorial can be accessed via R's help system. Typing "?TripleR" (without the quotes) after the package has been loaded opens a help page about the package. This help page contains a link to the tutorial. The tutorial can also be downloaded from <http://www.rforge.net/TripleR/files/TripleR-vignette/TripleR.pdf>.

Furthermore, there is a mailing list where current updates of TripleR are announced and problems and bugs can be posted: <http://lists.rforge.net/cgi-bin/mailman/listinfo/tripler-info>. The official Web site for TripleR can be found at <http://www.persoc.net/Toolbox/TripleR>.

examples we deal with apply interpersonal perception measures, analyses and results will be described in terms of perceivers and targets, instead of actors and partners. For a convenient output, all labels are set to the “perception style” after having loaded the TripleR package. The respective commands to achieve this are as follows (lines that are preceded by “#” are ignored by R):

```
# load the TripleR package for the active session
# and set labels in outputs to "perception style"
library(TripleR)
RR.style("perception")
```

See the built-in PDF tutorial or type in “?RR.style” (without the quotes) for more details concerning how to format the TripleR output. If no style settings are specified, all labels in the output will be printed in line with the default terminology (i.e., actors and partners).

The data format

For TripleR, the round-robin data have to be organized in long format. In this format, each observation is presented in one row (in contrast to the wide format, in which each row refers to 1 participant, and multiple observations are presented in multiple columns; see also Kenny et al., 2006). At least three columns are needed: the perceiver ID, the target ID, and one column for each assessed variable. If multiple groups are assessed, an additional column is needed to store the group ID. Perceiver and actor IDs have to be unique across groups. In the demo data sets of the package and in the examples below, the ID columns are called “perceiver.id,” “target.id,” and “group.id.” Note, however, that any other name can be assigned to these columns.

As an example, a round-robin data set for a single round-robin group and four measured variables (two indicators of liking, *liking_a* and *liking_b*; two indicators of metaperceived liking, *metaliking_a* and *metaliking_b*) looks like

perceiver.id	target.id	liking_a	liking_b	metaliking_a	metaliking_b
1	1	NA	NA	NA	NA
2	1	4	5	3	2
3	1	4	4	4	4
4	1	3	3	3	3
5	1	5	5	3	3
6	1	3	4	4	3
7	1	5	4	3	3
8	1	4	3	3	3
9	1	3	4	3	3
10	1	3	3	2	2
11	1	3	3	3	3
...					

The data indicate, for example, that perceiver “3” rates target “1” with a 4 on a liking scale (*liking_a*). For liking, there are no self-ratings; therefore, rows with the

same perceiver and target ID, which would normally contain the self-rating, are set to *NA* (i.e., not available/missing value).

If additional (non-round-robin) variables are assessed for each person, we recommend storing them in a separate data set. After export of the round-robin target and perceiver effects (see below), these two data sets can easily be combined for subsequent analyses.⁵

Raw data can be loaded into R in several formats—for example, as comma-separated files (csv; see *?read.csv*), Excel files (see *?xlsx* in the *xlsx* package; Dragulescu, 2010), or SPSS files (see *?read.spss* in the *foreign* package).

Possible analyses

TripleR can perform four kinds of analyses: (1) univariate manifest analyses (i.e., one measured variable), (2) univariate latent analyses, where two manifest variables are indicators for one latent construct (with the assumption of uncorrelated errors), (3) bivariate manifest analyses (i.e., two measured variables that are correlated within the SRM), and (4) bivariate latent analyses, where two latent constructs are measured by two manifest variables each.

All of these analyses can be done in a single round-robin group or with multiple groups. TripleR also provides reliability estimates for the perceiver and target effects in all analyses and the reliability of relationship effects in latent analyses (Bonito & Kenny, 2010).

Subsequent analyses and exporting the results

Usually one does not want to know only about the variance components and the within-SRM correlations. Often, one wants to correlate the actor and partner effects with the self-ratings, with non-round-robin personality questionnaires or with demographic variables. To do this, the actor/partner effects can be extracted from the results object and combined with other data (e.g., self-ratings) in another data set.

R provides many capabilities for subsequent analyses of the results of an SRA. If users prefer other software solutions, however, a data export can easily be done in several file formats.⁶ See the examples below for illustrations regarding how to export the results.

⁵ There is an extended example included in the TripleR tutorial (see Footnote 4), which describes the necessary steps to combine round-robin results and external variables, such as additional personality questionnaires.

⁶ Export of standard file formats such as .csv (see *?write.csv*) or .tab (see *?write.table*) is built into R's base system. Additional formats can be exported with the *foreign* package or with the *xlsx* package (Dragulescu, 2010).

Examples

The examples below are based on a subset of data collected in a study on first impressions (Back, Schmukle & Egloff, 2008; 2010; 2011). Since this data set is included in the TripleR package, all examples can be reproduced by the reader. By typing “?RR” (without the quotation marks), a help page about the main function "RR" is opened; it contains all the examples described below.

Univariate analyses

The simplest analysis is the partitioning of variances for a single variable in a single round-robin group. The data set

"likingLong" contains a single group of 54 members who rated how much they liked each other. This rating is stored in the variable *liking_a*. The analysis is done by a function called *RR*. As parameters, one has to specify the perceiver ID, the target ID, and the dependent variable. These variables are defined in a "formula syntax" that takes the form *DV~perceiver.id * target.id*. Furthermore, the user must specify the data set to which the formula should be applied.

Variance components

In the present example, the command and its output would appear as:

```
# load the TripleR package
library(TripleR)

# load the built-in data set
data(likingLong)

# set labels in outputs to "perception style"
RR.style("perception")

# Do a univariate SRA
RR(liking_a ~ perceiver.id * target.id, data=likingLong)

[1] "Round-Robin object ('RR'), calculated by TripleR"
[1] "Univariate analysis of one round robin variable"

[1] "Univariate analyses for: liking_a"
      estimate standardized   se t.value p.value
perceiver variance      0.172      0.194 0.035  4.914  0.000
target variance         0.105      0.119 0.022  4.727  0.000
relationship variance   0.609      0.687 0.017 36.827  0.000
error variance          NA         NA    NA    NA    NA
perceiver-target covariance 0.014      0.105 0.020  0.703  0.618
relationship covariance  0.080      0.131 0.017  4.809  0.000
[1] "Perceiver effect reliability: .937"
[1] "Target effect reliability: .901"
```

The output shows the variance partitioning and covariances of the SRA. The column *estimate* shows the unstandardized variance estimates; *standardized* shows these estimates normalized to 100% (i.e., perceiver, target, relationship, and error variance are summed to 100%). The next three columns show the standard error of the (co)variance estimate, the *t* value, and the corresponding *p* value. Since this analysis was for one single group, the within-groups *t* test was applied. There are significant interindividual differences for how much

people generally like others (i.e., interpersonal leniency; perceiver variance), as well as for how much people are generally liked (i.e., popularity; target variance). Most of the variance (68.7%) of liking, however, can be attributed to unique liking (relationship variance; this quantity also contains an unknown amount of error variance [see the Additional Analyses section below for a latent analysis that can separate error from relationship variance]). Furthermore, one can see a low but significant relationship correlation (dyadic reciprocity; $r = .131$), showing

that unique relationship effects within each dyad are reciprocated (if A uniquely likes B, B also uniquely likes A). These liking ratings were done at zero acquaintance; at longer acquaintance, higher dyadic reciprocities can be expected.

In data sets where self-ratings are provided (in the diagonal of the round-robin matrix), the output also prints correlations between self-ratings and perceiver and target effects. These correlations already are controlled for group membership but are not disattenuated for perceiver/target

effect reliability (note that SOREMO prints the disattenuated correlations).

Perceiver and target effects

Effects are also calculated and returned by the function (they are not printed in the standard output, but they are in the results object). To retrieve the effects, one has to assign the result of the function to a new variable. This variable then stores additional information:

```
# Assign the result to the new variable 'R1'
R1 <- RR(liking_a ~ perceiver.id * target.id, data=likingLong)

# 'head' shows the first few rows of the effects
head(R1$effects)
  id liking_a.p liking_a.t
1  1 -0.4768519  0.263888889
2 10 -0.3671652  0.077279202
3 11 -0.4063390  0.001068376
4 12  0.1520655 -0.403490028
5 13  0.6627493 -0.337250712
6 14  0.4141738  0.488247863
```

Effects can be retrieved with the \$ operator. In the present example, the participant with ID 1 has a negative perceiver effect (*liking_a.p*), meaning that he or she does not like others in general as much as the average perceiver of this group (the rating is about 0.5 scale points lower than the group average). By contrast, he or she has a positive target effect (*liking_a.t*), meaning that others rated him or her as more likable than the average target. In data sets where self-ratings are present, this data structure contains an additional column with the self-ratings.

All effects and self-ratings in this output are group mean centered. As was suggested by Kenny and colleagues (2006), correlations between these effects and external variables should be computed as partial correlations controlled for group membership.

Relationship effects

Since relationship effects are dyadic, they are provided in long format, with two columns specifying the perceiver and the target ID and one additional column specifying the unique dyad. Turning back to the “liking” data set, the first 10 relationship effects for the variable *liking_a* of our single group data set “likingLong” would look like

```
head(R1$effectsRel, 10)
  actor.id partner.id dyad relationship
1         1         10 1_01  0.21962855
2        10         1 1_01 -0.07666774
3         1         11 1_02  0.29583938
4        11         1 1_02 -0.03749395
5         1         12 1_03 -0.29960221
6        12         1 1_03 -0.59589851
7         1         13 1_04 -0.36584153
8        13         1 1_04 -1.10658227
9         1         14 1_05 -0.19134011
10       14         1 1_05  1.14199323
```

Latent analyses and multiple groups

In the analysis above, error variance could not be separated from relationship variance. To allow this separation, one has to provide two indicators for a latent construct. This is done in the formula interface by providing two variables separated by a slash:

```
liking_a/liking_b ~ perceiver.id*target.id.
```

If multiple groups have been assessed, a group ID has to be provided. This is done by a pipe symbol at the end of the formula:

```
liking_a/liking_b ~ perceiver.id*target.id|group.id.
```

The package also contains a data set with five round-robin groups of 10 persons each, in which a latent analysis for liking can be performed:

```

# load a multigroup data set
data(multiLikingLong)

# set labels in outputs to "perception style"
RR.style("perception")

# Do a latent analysis
RR(liking_a/liking_b ~ perceiver.id * target.id | group.id,
data=multiLikingLong)

[1] "Round-Robin object ('RR'), calculated by TripleR"
[1] "Latent construct analysis of one construct measured by two round robin
variables in multiple groups"

[1] "Univariate analyses for: liking_a/liking_b"
[1] "Group descriptives: n = 5 ; average group size = 10 ; range: 10 -
10"

```

	estimate	standardized	se	t.value	p.value
perceiver variance	0.158	0.154	0.045	3.478	0.013
target variance	0.064	0.062	0.026	2.492	0.034
relationship variance	0.615	0.597	0.119	5.170	0.003
error variance	0.192	0.187	NA	NA	NA
perceiver-target covariance	0.039	0.386	0.020	1.975	0.120
relationship covariance	0.155	0.252	0.070	2.213	0.091

```

[1] "Perceiver effect reliability: .625"
[1] "Target effect reliability: .440"
[1] "Relationship effect reliability: .887"

```

As can be seen in the output, an estimate for error variance is now provided, as well as reliability coefficients for the relationship effect and some group statistics. The reported error variance is the sum of the unstable perceiver variance, unstable target variance, and unstable relationship variance.

Correlations of SRM effects with other variables

As another example, the package contains a demo data set called *multiGroup*, which contains both round-robin and self-ratings of extraversion for 10 round-robin groups with 19 to 24 participants each. Additionally, another data set

called *multiNarc* contains individual scale scores for a self-report questionnaire of narcissism for the same participants.

Correlations with self-ratings In data sets where self-ratings are provided (in the diagonal of the round-robin matrix), the output prints correlations between self-ratings and perceiver and target effects. In the case of multiple groups, these correlations are controlled for group membership but are not disattenuated for perceiver/target effect unreliability (for an example on how to disattenuate these correlations, see below). In the following, there is an example of such an analysis for the *multiGroup* data set:

```

data(multiGroup)
RR.style("perception")

RR1 <- RR(ex~perceiver.id*target.id|group.id, data=multiGroup, na.rm=TRUE)
RR1

[1] "Round-Robin object ('RR'), calculated by TripleR"
[1] "Univariate analysis of one round robin variable in multiple groups"
[1] "Univariate analyses for: ex"
[1] "Group descriptives: n=10 ; average group size = 21.7 ; range: 19 - 24"

```

	estimate	standardized	se	t.value	p.value
perceiver variance	0.236	0.103	0.032	7.418	0.000
target variance	0.845	0.370	0.127	6.653	0.000
relationship variance	1.204	0.527	0.045	26.463	0.000
error variance	NA	NA	NA	NA	NA
perceiver-target covariance	-0.011	-0.024	0.050	-0.217	0.833
relationship covariance	0.106	0.088	0.037	2.881	0.018

```

[1] "Perceiver effect reliability: .801"
[1] "Target effect reliability: .935"

Partial correlations with self ratings (controlled for group membership):

```

	r	t	df	p
self rating with Perceiver effect (assumed similarity)	.307	4.634	206	.000
self rating with Target effect (self-other agreement)	.609	11.012	206	.000

The partial correlations at the end of the output show that self-ratings of extraversion are correlated both with the target effect ($r = .609$; self–other agreement: people who describe themselves as extraverted are seen by others as extraverted) and with the perceiver effect for extraversion ($r = .307$; assumed similarity: people who describe themselves as extraverted see others as extraverted).

Correlations with external variables To obtain the partial correlation between the target effect of extraversion ratings and the external narcissism score, controlled for group membership, one can use the TripleR function *parCor*:

```

data(multiGroup)
data(multiNarc)

# the function 'head' shows the first few lines of a data structure:
head(multiNarc)

  id narc
90201   7
90205   6
90207   3
90209  12
90210   8
90212   6

# set labels in outputs to "perception style"
RR.style("perception")

# calculate SRA effects for extraversion ratings
RR1 <- RR(ex ~ perceiver.id * target.id | group.id, multiGroup, na.rm=TRUE)

# merge effects and external variables to one data set
dat <- merge(RR1$effects, multiNarc, by="id")

# We now have a combined data set with SRA effects and external self
ratings:

head(dat)

  id group.id      ex.p      ex.t      ex.s narc
1 90201      2 -0.59649123  0.6892231  1.1428571  7
2 90205      2  0.04511278  0.5213033  0.1428571  6
3 90207      2  0.03007519 -1.7794486 -1.8571429  3
4 90209      2 -0.16541353  2.3107769  2.1428571 12
5 90210      2 -0.02756892  1.0676692  1.1428571  8
6 90212      2  0.30075188 -0.3659148  1.1428571  6

# function parCor(x, y, z) computes partial correlation between x and y,
controlled for group membership z

d1 <- parCor(dat$ex.t, dat$narc, dat$group.id)
d1

$par.cor
[1] 0.7620022

$t.value
[1] 15.91813

$df
[1] 183

$p
[1] 0

```

Correlations that are calculated by SOREMO are, by default, disattenuated for perceiver and/or target effect unreliability. To replicate these results, correlations have

to be disattenuated by following formula: $r_{\text{disatt}} = r_{\text{raw}} * 1 / \sqrt{(Rel_{\text{perceiver/target effect}})}$.

For the example above, the disattenuated partial correlation with the narcissism score would be

```
# disattenuate for target effect unreliability
parCor2 <- dl$par.cor * (1/sqrt(attr(RR1$effects$ex.t, "reliability")))
parCor2
[1] 0.7879463
```

Self-enhancement index The *RR* function can also calculate an index of self-enhancement (Kwan, John, Kenny, Bond & Robins, 2004, Eq. 3). This index compares each participant's self-rating with his or her tendency to over- or underrate others, as well as with others' ratings of him or her. While previous indexes of self-enhancement were confounded with irrelevant components of interpersonal

perception (Kwan, John, Robins & Kuang, 2008), this index based on SRAs is an unbiased index of self-enhancement, since it takes each person's perceiver and target effect into account. To compute this index, a parameter has to be passed to the *RR* function: *index="enhance"*. Now the data frame with the effects contains an additional column with the self-enhancement index:

```
RR1 <- RR(ex~perceiver.id*target.id|group.id, data=multiGroup, na.rm=TRUE,
index="enhance")
head(RR1$effects)
```

	id	group.id	ex.p	ex.t	ex.s	ex.enhance
1	90201	2	-0.59649123	0.6892231	1.1428571	1.050125313
2	90203	2	0.73934837	-0.5939850	1.1428571	0.997493734
3	90205	2	0.04511278	0.5213033	0.1428571	-0.423558897
4	90206	2	-0.49373434	0.7443609	0.1428571	-0.107769424
5	90207	2	0.03007519	-1.7794486	-1.8571429	-0.107769424
6	90209	2	-0.16541353	2.3107769	2.1428571	-0.002506266

Additional functions

TripleR provides some additional functions, which ease data inspection, transformation, and processing of results. Summary statistics for multiple groups can be obtained by *RR.summary*; missing values can be visually inspected by *plot_missings*. The function *getEffects* calculates the actor and partner (or perceiver and target) effects for a large number of variables and returns them together with the group-centered self-ratings and, eventually, the self-enhancement indexes in a convenient table. For further information on these functions, consult the help pages and the built-in tutorial of the package (see Footnote 4). A help page for each function with descriptions and examples can be displayed by typing a question mark into the R console, directly followed by the function name (e.g., *?plot_missings*).

Plotting the results

Plots are provided for each kind of analysis. Plots can easily be produced by calling the *plot* function with the results object as the parameter. In the example of liking judgments in multiple groups, a plot is produced that shows the distribution and the confidence estimate of variance components in each group (see Fig. 3):

```
data(multiLikingLong)
RR.style("perception")
R2 <- RR(liking_a/liking_b ~ perceiver.id * target.id | group.id,
data=multiLikingLong)
plot(R2)
```

Bivariate analyses

SRAs are capable of estimating the bivariate covariances for two round-robin variables. In the example data set, the

researchers also asked for the metaperception of liking (i.e., "How much do you think the other person likes you?"). Bivariate analyses are defined by providing two variables on

the left-hand side of the formula, separated by a plus sign:

$liking_a + metaliking_a \sim perceiver.id * target.id.$

```
# Do a bivariate analysis
data(likingLong)
RR.style("perception")
RR(liking_a + metaliking_a ~ perceiver.id * target.id, data=likingLong)

[1] "Round-Robin object ('RR'), calculated by TripleR"
[1] "Bivariate analysis of two variables, each measured by one round robin
variable"
[1] "Univariate analyses for: liking_a"
      estimate standardized   se t.value p.value
perceiver variance      0.172      0.194 0.035   4.914  0.000
target variance         0.105      0.119 0.022   4.727  0.000
relationship variance   0.609      0.687 0.017  36.827  0.000
error variance          NA         NA    NA     NA     NA
perceiver-target covariance 0.014      0.105 0.020   0.703  0.618
relationship covariance  0.080      0.131 0.017   4.809  0.000
[1] "Perceiver effect reliability: .937"
[1] "Target effect reliability: .901"

[1] "Univariate analyses for: metaliking_a"
      estimate standardized   se t.value p.value
perceiver variance      0.140      0.233 0.028   4.953  0.000
target variance         0.027      0.044 0.007   4.005  0.000
relationship variance   0.436      0.723 0.012  36.767  0.000
error variance          NA         NA    NA     NA     NA
perceiver-target covariance 0.002      0.031 0.010   0.195  0.779
relationship covariance  0.062      0.143 0.012   5.247  0.000
[1] "Perceiver effect reliability: .944"
[1] "Target effect reliability: .764"

[1] "Bivariate analyses:"
      estimate stand.   se   t.value   p.value
perceiver-perceiver cov      0.072 0.462 0.025   2.900  0.015
target-target cov            0.049 0.920 0.011   4.310  0.000
perceiver-target cov         0.014 0.206 0.011   1.258  0.359
target-perceiver cov         0.000 0.003 0.018   0.021  0.794
intrapersonal relationship cov 0.289 0.560 0.011  25.498  0.000
interpersonal relationship cov 0.067 0.129 0.011   5.893  0.000
```

In this case, univariate analyses are provided for each of the two variables. Additionally, six covariances are estimated, and are displayed in the block "Bivariate analyses." The labels in this block refer to the variables in the order they were entered in the formula. That means, the *perceiver-target correlation* (line 3 of the "Bivariate analyses" block) refers to the correlation of perceiver effects of *liking_a* and target effects of *metaliking_a*.

Bivariate analyses can be extended also to latent variables by providing two indicators for each construct,

and they can be extended to multiple groups—for example,

$liking_a/liking_b + metaliking_a/metaliking_b$
 $\sim perceiver.id * target.id / group.id.$

Longitudinal analyses

Longitudinal analyses with two times of measurement can be handled with a bivariate SRA (see above)—for

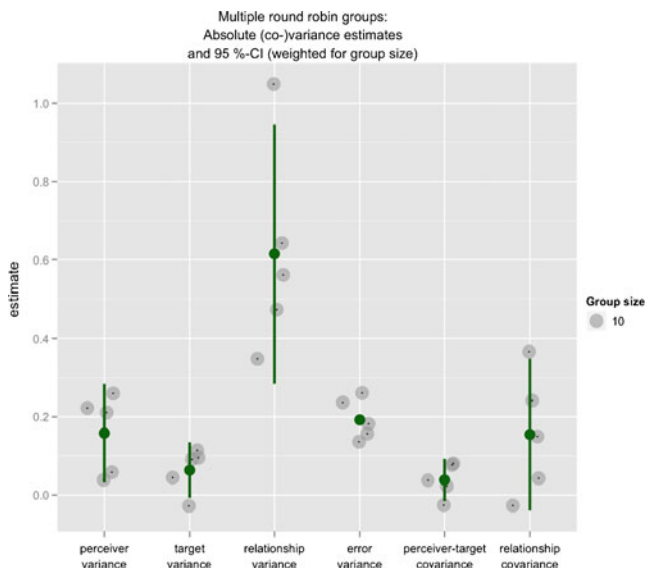


Fig. 3 Plot of (co)variances and confidence intervals for a multiple-group social relations analysis of perception data

example, $RR(\text{liking}_{t1} + \text{liking}_{t2} \sim \text{perceiver.id} * \text{target.id}, \text{data} = \text{dat})$.

If more than two longitudinal measurements are made, we suggest calculating the effects within each wave of measurement and submitting these data to a standard longitudinal analysis (for an example, see Denissen, Schönbrodt, van Zalk, Meeus & van Aken, 2011).

Limitations of the package

TripleR deals only with SRAs with indistinguishable members—for example, working teams, groups of friends, or unacquainted participants in a group study. In groups with strong roles prescribed for each member (like families), another approach based on SEMs has to be taken to estimate the variance components (see Kenny et al., 2006, Chap. 9). Another limitation is the current restriction to a maximum of two indicators for latent constructs.

Conclusion

SRAs based on round-robin designs allow for a differentiated understanding of many important social phenomena. However, these analyses require nonstan-

dard and complex statistical solutions and are, thus, still seldom applied. Building on the groundbreaking work of David Kenny's SRM and the SOREMO software, TripleR provides these solutions within a powerful yet convenient-to-use open-source software package. We hope that TripleR will be an invaluable tool for analyzing round-robin data and that it will foster the more frequent use of SRAs.

Author Note We thank Sascha Krause, Albrecht Küfner, and Kathrin Rentzsch for thoroughly testing the program and David Kenny for helpful comments on previous versions of the manuscript. Preparation of the manuscript was supported by a grant from the 2010 "Google Summer of Code" program to Felix Schönbrodt and by Grants BA 3731/1-1 and BA 3731/2-1 of the German Research Foundation (DFG) to Mitja Back.

Appendix

Installing R and the TripleR package

First, the R core system has to be installed. Installation files for R can be obtained from <http://cran.r-project.org/> and are provided for all major operating systems (Windows, Mac OS, Linux). Detailed instructions for installation can be obtained from the R Web site (<http://www.r-project.org>).

The R installation provides the core system and basic packages for standard statistical analyses such as multiple regression, ANOVAs, or factor analyses. There are, however, numerous additional packages with new functions, such as TripleR. To install TripleR, one has to launch the R console (which was installed in step 1) and to type `install.packages("TripleR")` into the R console. R will automatically load the necessary files and install the package to your system. TripleR uses some other packages (reshape, plyr, and ggplot2), which will be automatically installed on your system as well. Please note that the installation of some packages—for example, ggplot2—may take several minutes, during which the system is unresponsive or seems to be crashed.

After installation, TripleR is loaded into the current R session by typing `library(TripleR)`. Typing `?TripleR` opens the main help file for TripleR, in which a link to a step-by-step tutorial can be found, among other information. Typing `?RR` opens the help file for the main function RR.

A demo script

```
## lines beginning with '#' are comments

# if not already done:
# install the TripleR package (only has to be done once)
# install.packages("TripleR")

# load the TripleR package for the active session
# and set labels in outputs to "perception style"
library(TripleR)
RR.style("perception")

# open the help page
?RR

# load a built-in data set
data(likingLong)

# Do a univariate SRA
RR(liking_a ~ perceiver.id * target.id, data=likingLong)

# Assign the result to the new variable 'R1'
R1 <- RR(liking_a ~ perceiver.id * target.id, data=likingLong)

# show the first few rows of the effects
head(R1$effects)

# save these effects to a file
write.csv(R1$effects, "effects.csv")

# plot the results
plot(R1)

# load a multigroup data set
data(multiGroup)

# show a summary of groups in this data set
RR.summary(ex ~ perceiver.id * target.id | group.id, data=multiGroup)

# inspect the distribution of missing values in variable liking_a
plot_missings(ex ~ perceiver.id * target.id | group.id, data=multiGroup)

# Do a bivariate analysis
RR(ex+ne ~ perceiver.id * target.id | group.id, data=multiGroup,
na.rm=TRUE)
```

References

- Back, M. D., Baumert, A., Denissen, J. J. A., Hartung, F.-M., Penke, L., Schmukle, S. C., ... et al. (2011a). PERSOC: A unified framework for understanding the dynamic interplay of personality and social relationships. *European Journal of Personality*, *25*, 90–107.
- Back, M. D., & Kenny, D. A. (2010). The social relations model: How to understand dyadic processes. *Social and Personality Psychology Compass*, *4*, 855–870.
- Back, M. D., Schmukle, S. C., & Egloff, B. (2008). Becoming friends by chance. *Psychological Science*, *19*, 439–440.
- Back, M. D., Schmukle, S. C., & Egloff, B. (2010). Why are narcissists so charming at first sight? Decoding the narcissism–popularity link at zero acquaintance. *Journal of Personality and Social Psychology*, *98*, 132–145.
- Back, M. D., Schmukle, S. C., & Egloff, B. (2011). A closer look at first sight: Social relations lens model analyses of personality and interpersonal attraction at zero acquaintance. *European Journal of Personality*, *25*, 225–238.

- Bond, C. F., & Lashley, B. R. (1996). Round-robin analysis of social interaction: Exact and estimated standard errors. *Psychometrika*, *61*, 303–311.
- Bonito, J. A., & Kenny, D. A. (2010). The measurement of reliability of social relations components from round-robin designs. *Personal Relationships*, *17*, 235–251.
- Denissen, J. J. A., Schönbrodt, F. D., van Zalk, M., Meeus, W. H. J., & van Aken, M. A. G. (2011). Antecedents and consequences of peer-rated intelligence. *European Journal of Personality*, *25*, 108–119. doi:10.1002/per.799
- Dragulescu, A. A. (2010). *xlsx: Read, write, format Excel 2007 (xlsx) files. R package version 0.2.1*. Retrieved from <http://CRAN.R-project.org/package=xlsx>
- Gill, P. S., & Swartz, T. B. (2007). Bayesian analysis of dyadic data. *American Journal of Mathematical and Management Sciences*, *27*, 73–92.
- Kenny, D. A. (1990). Design issues in dyadic research. In C. Hendrick & M. S. Clark (Eds.), *Review of personality and social psychology* (Research methods in personality and social psychology, Vol. 11, pp. 164–184). Newbury Park, CA: Sage.
- Kenny, D. A. (1994). *Interpersonal perceptions: A social relations analysis*. New York: Guilford.
- Kenny, D. A. (2007). *Estimation of the SRM using specialized software*. Retrieved from davidakenny.net/doc/srmssoftware.doc.
- Kenny, D. A., & Kashy, D. A. (2010). Dyadic data analysis using multilevel modeling. In J. Hox & J. K. Roberts (Eds.), *The handbook of multilevel analysis* (pp. 335–370). London: Taylor Francis.
- Kenny, D. A., Kashy, D. A., & Cook, W. L. (2006). *Dyadic data analysis*. New York: Guilford.
- Kenny, D. A., & La Voie, L. (1984). The social relations model. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (pp. 142–182). Orlando, FL: Academic Press.
- Kenny, D. A., & Livi, S. (2009). A componential analysis of leadership using the social relations model. In F. J. Yammarino & F. Dansereau (Eds.), *Multi-Level issues in organizational behavior and leadership* (pp. 147–191). Bingley, U.K.: Emerald.
- Kwan, V. S., John, O. P., Kenny, D. A., Bond, M. H., & Robins, R. W. (2004). Reconceptualizing individual differences in self-enhancement bias: An interpersonal approach. *Psychological Review*, *111*, 94–110.
- Kwan, V. S., John, O. P., Robins, R. W., & Kuang, L. L. (2008). Conceptualizing and assessing self-enhancement bias: A componential approach. *Journal of Personality and Social Psychology*, *94*, 1062–1077.
- Lashley, B. R., & Bond, C. F. (1997). Significance testing for round robin data. *Psychological Methods*, *2*, 278–291.
- Lüdtke, O., Robitzsch, A., Kenny, D. A., & Trautwein, U. (2011). *A general and flexible approach to estimating the social relations model using Bayesian methods*. Manuscript submitted for publication.
- Malloy, T. E., & Kenny, D. A. (1986). The social relations model: An integrative method for personality research. *Journal of Personality*, *54*, 199–225.
- Olsen, J. A., & Kenny, D. A. (2006). Structural equation modeling with interchangeable dyads. *Psychological Methods*, *11*, 127–141.
- R Development Core Team. (2008). *R: A language and environment for statistical computing*. Vienna, Austria: Basic Books.
- Schmukle, S. C., Schönbrodt, F. D., & Back, M. D. (2011). *TripleR: A package for round robin analyses using R (version 1.1)*. Retrieved from <http://www.persoc.net>
- Snijders, T. A. B., & Kenny, D. A. (1999). The social relations model for family data: A multilevel approach. *Personal Relationships*, *6*, 471–486.