Package ‘TripleR’

October 12, 2022

Encoding UTF-8

Type Package

Title Social Relation Model (SRM) Analyses for Single or Multiple Groups

Version 1.5.4

Date 2022-04-25

Author F. D. Schönbrodt, M. D. Back, S. C. Schmukle

Maintainer Felix Schönbrodt <felix.schoenbrodt@psy.lmu.de>

Depends ggplot2 (>= 0.9.3)

Imports reshape2, plyr

Suggests lme4, R.rsp

VignetteBuilder R.rsp

Description Social Relation Model (SRM) analyses for single or multiple round-robin groups are performed. These analyses are either based on one manifest variable, one latent construct measured by two manifest variables, two manifest variables and their bivariate relations, or two latent constructs each measured by two manifest variables. Within-group t-tests for variance components and covariances are provided for single groups. For multiple groups two types of significance tests are provided: between-groups t-tests (as in SOREMO) and enhanced standard errors based on Lashley and Bond (1997) <DOI:10.1037/1082-989X.2.3.278>. Handling for missing values is provided.

License GPL (>= 2)

LazyLoad yes

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-26 11:40:08 UTC

R topics documented:

TripleR-package 2
getEffects 3
long2matrix 4
matrix2long 6
parCor 7
plot.RRuni 8
plot_missings 9
RR 10
RR.style 16
RR.summary 17
selfCor 18

Index **19**

`TripleR-package` *Triple R: A Package for Round-Robin Analyses Using R*

Description

Social Relation Model (SRM) analyses for single or multiple round-robin groups are performed. These analyses are either based on one manifest variable, one latent construct measured by two manifest variables, two manifest variables and their bivariate relations, or bivariate relations between two latent constructs each measured by two manifest variables.

Details

For further information of usage see `RR` or the built in pdf document 'How to use TripleR'. You can browse all included vignettes by opening the index of the package documentation (scroll down to the very end of this page and click on 'Index'; than click on 'Overview of user guides and package vignettes').

If you use this package in your research, please cite it as: Schönbrodt, F. D., Back, M. D., & Schmukle, S. C. (2012). TripleR: An R package for social relations analyses based on round-robin designs. *Behavior Research Methods, 44*, 455–470. doi:10.3758/s13428-011-0150-4

Author(s)

Felix D. Schönbrodt, Mitja D. Back, & Stefan C. Schmukle

References

Further information on SRM and its application:

- 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. doi:10.1002/per.790

- Bond, C. F., Jr., & Lashley, B. R. (1996). Round-robin analysis of social interaction: Exact and estimated standard errors. *Psychometrika, 61*, 303-311. doi:10.1007/BF02294341
- Bonito, J. A., & Kenny, D. A. (2010). The measurement of reliability of social relations components from round-robin designs. *Personal Relationships, 17*, 235 - 251. doi:10.1111/j.1475-6811.2010.01274.x
- Kenny, D. A. (1994). *Interpersonal perception: A social relations analysis*. New York: Guilford Press.
- Kenny, D. A., & La Voie, L. J. (1984). The social relations model. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (Vol. 18, pp. 142-182). San Diego, CA: Academic Press.
- Kwan, V. S. Y., 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. doi:10.1037/0033-295X.111.1.94
- Lashley, B. R., & Bond, C. F., Jr. (1997). Significance testing for round robin data. *Psychological Methods*, *2*, 278-291. doi:10.1037/1082-989X.2.3.278
- Lashley, B. R. & Kenny, D. A. (1998). Power estimation in social relations analyses. *Psychological Methods, 3*, 328-338. doi:10.1037/1082-989X.3.3.328

`getEffects` *Calculates round robin effects for multiple variables*

Description

This is a helper function which facilitates the calculation of round robin effects if many variables are assessed. Only univariate analyses are possible at the moment.

Usage

```
getEffects(formule, data, varlist, by=NA, na.rm=TRUE,
minVar=localOptions$minVar, gm=FALSE, ...)
```

Arguments

`formule` The right hand side of the formula, specifying the actor, partner and group variable.

`data` The data frame.

`varlist` A vector with the column names (the column numbers are not possible!) of the variables which should be inserted at the left hand side of the formula.

`by` By which variables should the results be merged? If by is NA (the default), a intelligent default handling is performed. It is strongly recommended to keep the defaults.

`na.rm` This argument is passed to function `RR`.

`minVar` Set the minVar parameter for all analyses. See `RR` for details on this parameter.

`gm` Should effects returned as group centered (`gm=FALSE`, or with group means included (`gm=TRUE`)?)

`...` Additional parameters passed to RR (e.g., selfenhance)

Value

A data frame with all effects is returned

Author(s)

Felix Schönbrodt

See Also

RR

Examples

```
data(likingLong)
res <- getEffects(~perceiver.id*target.id,
  data=likingLong,
  varlist=c("liking_a", "liking_b", "metaliking_a", "metaliking_b"))
str(res)

# effects including group means:
res.gm <- getEffects(~perceiver.id*target.id,
  data=likingLong, gm=TRUE,
  varlist=c("liking_a", "liking_b", "metaliking_a", "metaliking_b"))
str(res.gm)


# Multipe groups
#----------------------
data("multiLikingLong")

res.g <- getEffects(~perceiver.id*target.id|group.id,
  data=multiLikingLong,
  varlist=c("liking_a", "liking_b", "metaliking_a", "metaliking_b"))
str(res.g)

# effects including group means:
res.g.gm <- getEffects(~perceiver.id*target.id|group.id,
  data=multiLikingLong, gm=TRUE,
  varlist=c("liking_a", "liking_b", "metaliking_a", "metaliking_b"))
str(res.g.gm)
```

long2matrix *Convert long format to a quadratic matrix*

Description

This function converts data from the long format to a quadratic round robin matrix. It is ensured that the matrix in fact is quadratic (i.e. rows or columns are added if necessary).

Usage

```
long2matrix(formule, data, minData=1, verbose=TRUE, reduce=TRUE,
skip3=TRUE, g.id=NULL, exclude.ids=NULL, ...)
```

Arguments

`formule`	A formula specifying the variable names. Example usage: `DV ~ perceiver.id*target.id | group.id` (group.id only necessary in multi group data sets)
`data`	A data frame in long format)
`verbose`	Should additional information be printed?
`minData`	Sets the minimum of data points which have to be present in each row and column
`reduce`	Should persons that only are actors or only partners be removed? Actor/partner effects and variance components can only be calculated, if every person is both an actor and a partner. For displaying reasons (e.g. for data inspection), however, it can be necessary to check, which persons are missing.
`skip3`	Should groups with 3 or fewer participants be skipped? SRAs need groups with a minimum of 4 participants
`g.id`	For internal use only
`exclude.ids`	A list of participant ids, which should be excluded
`...`	Further undocumented or internal arguments passed to the function

Value

A list of quadratic matrix. Each list entry is one group, as specified by `group.id`. If only one group is present, this is stored in the first list entry.

See Also

`matrix2long`

Examples

```
#load a data set in long style
data("multiGroup")

str(multiGroup)
qm <- long2matrix(ex~perceiver.id*target.id|group.id, multiGroup)
qm[[2]]

# we see some warnings that some persons are only actors or only partners.
# Let's check the data without removing them:
qm2 <- long2matrix(ex~perceiver.id*target.id|group.id, multiGroup, reduce=FALSE)
qm2[[2]]
```

matrix2long *Convert a quadratic matrix to long format*

Description

This function converts data from a quadratic round robin matrix into long format.

Usage

```
matrix2long(M, new.ids=TRUE, var.id=”value”)
```

Arguments

`M` A matrix with actors in rows and partners in columns)

`new.ids` Should new ids for actors and partners be defined? (If new.ids=FALSE the row and column names are taken. In that case, you have to make sure, that rows and columns have the same set of names.)

`var.id` The name of the column with the measured variable in the returned data frame

Value

A data frame in long format

See Also

long2matrix

Examples

```
#The example data are taken from the ”Mainz Freshman Study” and consist
# of ratings of liking as well as ratings of the metaperception of
# liking at zero-acquaintance using a Round-Robin group of 54 participants
# (Back, Schmukle, & Egloff, in pres)

# load a data set in matrix style
data(”liking_a”)

str(liking_a)
long <- matrix2long(liking_a)

str(long)
```

parCor *partial correlation*

Description

Performs partial correlations between x and y, controlled for z.

Usage

```
parCor(x,y,z)
```

Arguments

x	First variable
y	Second variable
z	Control variable. This variable is coerced into a factor; in the TripleR context z usually is the group id.

Details

Performs partial correlations between x and y, controlled for z. The control variable is coerced into a factor; in the TripleR context z usually is the group id. Do *not* use this function with a continuous control variable - results will be wrong! Degrees of freedom for the *t* test are reduced by *g* - 1 (*g* is the number of groups).

Value

par.cor	partial correlation
df	degrees of freedom for the t test
t.value	t value
p	p value

See Also

RR, getEffects

Examples

```
data(multiGroup)
data(multiNarc)

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

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

```
# merge 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)

# 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


# disattenuate for target effect reliability
parCor2 <- d1$par.cor * (1/sqrt(attr(RR1$effects$ex.t, "reliability")))
parCor2
```

`plot.RRuni` *Plot variance components from SRAs*

Description

This function plots variance components for all RR objects.

Usage

```
## S3 method for class 'RRuni'
plot(x, ..., measure=NA, geom="bar")
## S3 method for class 'RRmulti'
plot(x, ..., measure=NA, geom="scatter", conf.level=0.95, connect=FALSE)
```

Arguments

`x`	A RR results object
`...`	Further arguments passed to the plot function
`measure`	Sets the labels to ‘behavior’ or ‘perception’; if no parameter is provided the function uses the defaults from RR.style
`geom`	Style of plot: set ‘bar’ or ‘pie’ for single groups; ‘scatter’ or ‘bar’ for multiple groups
`conf.level`	Confidence level for error bars in the scatter style
`connect`	Should dots for variance components of each group be connected? (Looks usually very cluttered ...)

Value

A ggplot2 object is returned - that means, you can subsequently adjust the scale labels, etc.

Examples

```
## Not run:
data(likingLong)
RR1 <- RR(liking_a ~ perceiver.id*target.id, data=likingLong)
plot(RR1)
plot(RR1, geom="pie")

RR2 <- RR(liking_a + metaliking_a ~ perceiver.id*target.id, data=likingLong)
plot(RR2)


data("multiLikingLong")
RR1m <- RR(liking_a ~ perceiver.id*target.id|group.id, data=multiLikingLong)
plot(RR1m)
plot(RR1m, measure="perception")
plot(RR1m, measure="perception", geom="bar")
plot(RR1m, measure="perception", connect=TRUE)

RR2m <- RR(liking_a + metaliking_a ~ perceiver.id*target.id|group.id, data=multiLikingLong)
plot(RR2m)

## End(Not run)
```

`plot_missings` *Plot missing values*

Description

This function plots missing values in the round robin matrix for visual inspection.

Usage

```
plot_missings(formule, data, show.ids=TRUE)
```

Arguments

`formule` A formula specifying the variable names. Example usage: `DV ~ perceiver.id*target.id | group.id` (group.id only necessary in multi group data sets)

`data` The data frame

`show.ids` Should the id's of the persons be printed on the axes?

Examples

```
data(multiGroup)
m2 <- multiGroup
m2$ex[m2$perceiver.id==90201] <- NA
m2$ex[m2$perceiver.id>92001] <- NA
m2$ex[sample(nrow(m2),  1000)] <- NA
plot_missings(ex~perceiver.id*target.id|group.id, data=m2)
(RR1m <- RR(ex~perceiver.id*target.id|group.id, data=m2, na.rm=TRUE))
```

RR *Triple R: Round-Robin Analyses Using R*

Description

The function RR performs Social Relation Model (SRM) analyses for single or multiple round-robin groups.

Usage

```
RR(formula, data, na.rm=FALSE, minData=1, verbose=TRUE, g.id=NULL, index="",
exclude.ids="", varname=NA, se="LashleyBond", minVar=localOptions$minVar, ...)
```

Arguments

`formula` a formula (see details) consisting of a measure (e.g. a rating) obtained with a round-robin group

`data` A data frame in long format

`na.rm` If missing values are present, you have to set this parameter to TRUE

`minData` Sets the minimum of data points which have to be present in each row and column

`verbose` Defines if warnings and additional information should be printed on execution

`g.id` For internal use only; do not set this parameter

`index` set `index = 'enhance'` for additionally requesting an index for self enhancement (self rating - perceiver effect - target effect - group mean of self ratings; Kwan, John, Kenny, Bond, & Robins, 2004) along with the actor and partner effects.

`exclude.ids` For internal use only; do not set this parameter

`varname` The name stem of the effects variables. By default, this is the first variable passed in the formula. In case of latent constructs, however, it might be preferable to set a new name for the latent construct.

`se` This defines how significance tests are computed in the multi-group case. Either "LashleyBond" (= recommended significancetest by Lashley & Bond, 1997; default) or "SOREMO" (= between group significance test as implemented in SOREMO). In single groups, always the Lashley-Bond standard errors and significance tests are provided.

`minVar` Actor and partners effects are only calculated if the respective relative variance component is greater than minVar. Set minVar to NA if this cleaning should not be performed. For small groups, Kenny (1994) suggests a minimum relative variance of 10% for the interpretation of SRM effects. In any case, actor/ partner effects and correlations with these variables should not be interpreted if these components have negative variance estimates. minVar defaults to zero; with RR.style this default can be changed for all subsequent analyses.

`...` Further undocumented or internal arguments passed to the function

Details

Please note: detailed instructions on how to use the TripleR package are provided in the built in pdf document 'How to use TripleR'. You can open this document either by on this link: `../doc/TripleR.pdf`, or you can browse all included vignettes by opening the index of the package documentation (scroll down to the very end of this page and click on 'Index'; than click on 'Overview of user guides and package vignettes'). These help files are only for quick references.

The estimation of the parameters of the Social Relation Model (SRM) is based on formulae provided by Kenny (1994; p. 236-244). If multiple groups are provided, SRMs are computed within each single group. Variance components then are calculated as the weighted average across groups (weighted with N – 1). Please note that in case of largely varying group sizes a precision weighting based on inverse-variance weights might be preferable but is not currently implemented in TripleR. For tests of significance of a single group, Triple R computes standard errors by using formulae published by Bond and Lashley (1996) for the case of a univariate SRM analysis. The formulae for the standard errors of the bivariate SRM parameters were kindly provided to us by C.F. Bond in personal communication. If multiple groups are provided, by default the standard error is computed as the square root of the weighted mean of the squared group-specific standard errors, as recommended by Lashley and Bond (1997). For `se` = "`SOREMO`" a between-group t-test is employed to calculate the significance which was suggested by Kenny and La Voie (1984) and is used in the software SOREMO.

The Bond-Lashley standard errors operate under the assumption that the only source of variance in the SRM parameter is sampling error, i.e. that variance estimates do not differ systematically between groups. Lashley and Kenny (1998) found that this assumption was basically true in 13 different round robin studies. Thus, we generally recommend using the Bond-Lashley standard errors (`se="LashleyBond"`), which are more powerful than the Kenny-La Voie standard errors (`se="SOREMO"`), see Lashley and Bond (1997) and Lashley and Kenny (1998). However, in case that you have reasons to assume that variances systematically vary across groups, the Kenny-La Voie standard errors (`se="SOREMO"`) might be preferable as they are likely more robust in the presence of group effects (Lashley & Kenny, 1998).

The formula to specify the round robin design has following notation: `DV ~ perceiver.id * target.id | group.id` (group.id is only provided if multiple groups are present). If two variables are used to describe one latent construct, both are connected with a slash on the left hand side: `DV1a/DV1b ~ perceiver.id * target.id`. If two variables are used to describe two manifest constructs, both are connected with a + on the left hand side: `DV1+DV2 ~ perceiver.id * target.id`. A latent analysis of two constructs would be notated as following: `DV1a/DV1b + DV2a/DV2b ~ perceiver.id * target.id`.

Data sets from the Mainz Freshman Study (see Back, Schmukle, & Egloff, in press) are included (liking_a, liking_b, metaliking_a, metaliking_b, likingLong), as well as an additional data set containing missing values (multiGroup)

The handling for missing data (`na.rm=TRUE`) is performed in three steps:

- Rows and columns which have less then `minData` data points are removed from the matrix (i.e. both the 'missing' row or column and the corresponding column or row, even if they have data in them)
- For the calculation of actor and partner variances, actor-partner-covariances and the respective effects, missing values are imputed as the average of the respective row and col means. The calculation of relationship variances and covariances as well as relationship effects is also

based on the imputed data set; however, single relationship effects which were missing in the original data set are set to missing again.

- In the case of multiple variables (i.e., latent or bivariate analyses), participants are excluded listwise to ensure that all analyses are based on the same data set.

Value

Printed are both unstandardized and standardized SRM estimates with the corresponding standard errors and t-values for actor variance, partner variance, relationship variance, error variance, actor-partner covariance, and relationship covariance. In case of a bivariate analysis values are additionally provided for actor-actor covariance, partner-partner covariance, actor-partner covariance, partner-actor covariance, intrapersonal relationship covariance, and interpersonal relationship covariance.

Reliabilities of actor, partner, and relationship effects (the latter only in the case of latent analyses) are printed according to Bonito & Kenny (2010).

The returned values depend on the kind of analyses that is performed:

Univariate, single group:

`effects`	actor and partner effects for all participants; if self ratings are provided they also are returned as group mean centered values
`effects.gm`	actor and partner effects for all participants, group mean added
`effectsRel`	relationship effects for all participants
`varComp`	variance components

Bivariate, single group:

`univariate`	List of results of univariate of SRM analyses of both variables- specify variable in index: univariate[[1]] or univariate[[2]]. That means, each of the both `univariate` objects is a complete RR object for the univariate analyses, nested in the results objects. If you want to retrieve the effects for the first variable, you have to type `RR2$univariate[[1]]$effects`. If you want to retrieve the variance components, you have to type `RR2$univariate[[1]]$varComp`
`bivariate`	Results of bivariate SRM analyses

In the multiple group case, following values are returned:

`univariate`	The weighted average of univariate SRM results
`bivariate`	The weighted average of bivariate SRM results
`groups`	SRM results for each group
`effects`	actor and partner effects for all participants
`effectsRel`	relationship effects for all participants
`varComp.group`	a list of variance components in all single groups
`group.var`	group variance estimate

If self ratings are present in the data set, the function also prints the correlation between self ratings and actor/partner effects. In case of multiple groups, these are corrected for group membership (partial correlations). These correlations with self-ratings can also directly be computed with the function `selfCor`. Partial correlations to external (non-SRM) variables can be computed with the function `parCor`

Note

In case that a behavior was measured, the variances of an SRM analysis are labeled as actor variance, partner variance and relationship variance (default output labels). In case that a perception was measured, perceiver is printed instead of actor and target is printed instead of partner. You can set appropriate output labels by using the function `RR.style`. These settings from RR.style, however, can be overwritten for each `print` call: :

- `print(RRobject, measure1="behavior")`: prints output for a univariate analysis of behavior data.
- `print(RRobject, measure1="perception")`: prints output for a univariate analysis of perception data.
- `print(RRobject, measure1="behavior", measure2="behavior")`: prints output for a bivariate analysis of behavior data.
- `print(RRobject, measure1="perception", measure2="perception")`: prints output for a bivariate analysis of perception data.
- `print(RRobject, measure1="behavior", measure2="perception")` or `print(RRobject, measure1="perception", measure2="behavior")`: prints output for a bivariate analysis of behavior and perception data.
- `print(RRobject, measure1="perception", measure2="metaperception")`: is for the special case that a perception and a corresponding metaperception was measured. In this case, additionally the appropriate output labels for bivariate SRM analyses of other- and metaperceptions (reciprocities, assumed reciprocities, meta-accuracies; see Kenny, 1994) are presented.
- `print(RRobject, digits=6)`: Provide the number of displayed digits.

You can plot any RR object using `plot(RR)`. See `plot.RRuni` for possible parameters.

Author(s)

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

References

The main reference for the TripleR package is:

- Schönbrodt, F. D., Back, M. D., & Schmukle, S. C. (2012). TripleR: An R package for social relations analyses based on round-robin designs. *Behavior Research Methods, 44*, 455-470. doi:10.3758/s13428-011-0150-4

Please cite this paper if you use TripleR in your research.

Further information on SRM and its application:

- 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. doi:10.1002/per.790
- Bond, C. F., Jr., & Lashley, B. R. (1996). Round-robin analysis of social interaction: Exact and estimated standard errors. *Psychometrika, 61*, 303-311. doi:10.1007/BF02294341

- Bonito, J. A., & Kenny, D. A. (2010). The measurement of reliability of social relations components from round-robin designs. *Personal Relationships, 17*, 235 - 251. doi:10.1111/j.1475-6811.2010.01274.x
- Kenny, D. A. (1994). *Interpersonal perception: A social relations analysis*. New York: Guilford Press.
- Kenny, D. A., & La Voie, L. J. (1984). The social relations model. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (Vol. 18, pp. 142-182). San Diego, CA: Academic Press.
- Kwan, V. S. Y., 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. doi:10.1037/0033-295X.111.1.94
- Lashley, B. R., & Bond, C. F., Jr. (1997). Significance testing for round robin data. *Psychological Methods*, *2*, 278-291. doi:10.1037/1082-989X.2.3.278
- Lashley, B. R. & Kenny, D. A. (1998). Power estimation in social relations analyses. *Psychological Methods, 3*, 328-338. doi:10.1037/1082-989X.3.3.328

See Also

RR.style, getEffects, plot_missings, long2matrix, matrix2long, plot.RRuni, RR.summary, selfCor, parCor

Examples

```
# The example data are taken from the "Mainz Freshman Study" and consist
# of ratings of liking as well as ratings of the metaperception of
# liking at zero-acquaintance using a Round-Robin group of 54 participants

#-----------------------------------------------------------
# ----  Single group   --------------------------------------
#-----------------------------------------------------------

# Load data frame in long format - it contains 4 variables:
#liking ratings indicator a (liking_a, "How likeable do you find this person?")
#liking ratings indicator b (liking_b, "Would you like to get to know this person?")
#metaliking ratings indicator a (metaliking_a, "How likeable does this person find you?")
#metaliking ratings indicator b (metaliking_b, "Would this person like to get to know you?")
data("likingLong")

#manifest univariate SRM analysis
RR1 <- RR(liking_a ~ perceiver.id*target.id, data=likingLong)

## Not run:
#manifest bivariate SRM analysis
RR2 <- RR(liking_a + metaliking_a ~ perceiver.id*target.id, data=likingLong)

#latent (construct-level) univariate SRM analysis
RR3 <- RR(liking_a / liking_b ~ perceiver.id*target.id, data=likingLong)

#latent (construct-level) univariate SRM analysis, define new variable name for the latent construct
RR3b <- RR(liking_a / liking_b ~ perceiver.id*target.id, data=likingLong, varname="liking")
```

```
# compare:
head(RR3$effects)
head(RR3b$effects)

#latent (construct-level) bivariate SRM analysis
RR4 <- RR(liking_a/liking_b + metaliking_a/metaliking_b ~ perceiver.id*target.id, data=likingLong)


# prints output of the manifest univariate analysis
# in terms of actor and partner variance (default output labels)
print(RR1, measure1=”behavior”)

# prints output of the manifest univariate analysis
# in terms of perceiver and target variance (appropriate for perception data)
print(RR1, measure1=”perception”)

# prints output of the manifest bivariate SRM analysis appropriate
# for perception-metaperception data
print(RR2, measure1=”perception”, measure2=”metaperception”)

#prints output of the latent univariate SRM analysis
# appropriate for perception data
print(RR3, measure1=”perception”)

#prints output of the latent bivariate SRM analysis
# appropriate for perception-perception data
# Note: you can use abbreviations of the strings ”behavior”, ”perception”, and ”metaperception”
print(RR4, measure1=”p”, measure2=”p”)

## End(Not run)


#-------------------------------------------------------------
# ----  Multiple groups ---------------------------------------
#-------------------------------------------------------------

# data(”multiLikingLong”) is a variant of the liking data set (see above) with multiple groups
data(”multiLikingLong”)

# set RR.style to ”perception” (affects subsequent printing of objects)
RR.style(”perception”)

#manifest univariate SRM analysis
RR1m <- RR(liking_a ~ perceiver.id*target.id|group.id, data=multiLikingLong)

## Not run:
#manifest bivariate SRM analysis
RR2m <- RR(liking_a + metaliking_a ~ perceiver.id*target.id|group.id, data=multiLikingLong)

#latent (construct-level) univariate SRM analysis
RR3m <- RR(liking_a / liking_b ~ perceiver.id*target.id|group.id, data=multiLikingLong)

#latent (construct-level) bivariate SRM analysis
```

```
RR4m <- RR(liking_a/liking_b + metaliking_a/metaliking_b ~ perceiver.id*target.id|group.id,
  data=multiLikingLong)

# prints output of the manifest univariate analysis
# in terms of actor and partner variance (default output labels)
print(RR1m, measure1="behavior")

# prints output of the manifest univariate analysis
# in terms of perceiver and target variance (appropriate for perception data)
print(RR1m, measure1="perception")


#-------------------------------------------------------------
# ----  Multiple groups with missing values ------------------------------------
#-------------------------------------------------------------

# a multi group data set with two variables:
# ex = extraversion ratings, and ne = neurotizism ratings
data("multiGroup")

#manifest univariate SRM analysis, data set with missings
RR1miss <- RR(ex~perceiver.id*target.id|group.id, data=multiGroup, na.rm=TRUE)

#manifest univariate SRM analysis, data set with missings,
# minimum 10 data points are requested for each participant
RR1miss <- RR(ex~perceiver.id*target.id|group.id, data=multiGroup, na.rm=TRUE, minData=10)

## End(Not run)
```

RR.style *Set labeling styles for RR analyses*

Description

This function sets labels for the printing of RR-objects and for plots. All subsequent calls of RR will be produced in this style (until another style is set). That means, usually you only have to define this style once at the start of your project.

Usage

```
RR.style(style="behavior", suffixes=NA, minVar=NA)
```

Arguments

style	a string defining the labeling style - either ‘behavior’ or ‘perception’.
suffixes	Which suffixes should be append to the actor and partner effects, and to the self ratings? Default is .a, .p, and .s, for actor/ partner/ self in the case of behavior, or .p, .t, and .s, for perceiver/ target/ self in the case of perceptions. If no

suffixes are provided in the parameters, these defaults are taken depending on the `style` parameter

`minVar` Set the minVar parameter for all subsequent analyses. See `RR` for details on this parameter.

Value

Printing options and naming conventions are set for all subsequent analyses. If you specify other styles in a print.RR call, this setting temporarily overwrites the settings from RR.style (without changing them).

Author(s)

Felix D. Schönbrodt

Examples

```
data("likingLong")

RR.style("behavior")
RR(liking_a ~ perceiver.id*target.id, data=likingLong)

RR.style("p") # a "p" is enough for "perception"
RR(liking_a ~ perceiver.id*target.id, data=likingLong)
```

RR.summary *Print group descriptives*

Description

Shows descriptives for multiple groups (how many groups, which groups are excluded, group sizes)

Usage

```
RR.summary(formule, data)
```

Arguments

`formule` A formula specifying the variable names. Example usage: `DV ~ perceiver.id*target.id | group.id` (group.id only necessary in multi group data sets)

`data` A data frame in long format)

Value

Printed output

Examples

```
data("multiGroup")
RR.summary(ex~perceiver.id*target.id|group.id, data=multiGroup)
```

selfCor *partial correlation*

Description

Performs partial correlations between x and y, controlled for z.

Usage

```
selfCor(x, digits=3, measure=NA)
```

Arguments

`x`	An RR object
`digits`	Digits to which values are rounded in the output
`measure`	Either "behavior" or "percetion". Only affects labeling of output; if NA, the current setting of RR.style is taken as default.

Details

Prints correlations between self ratings (if present in the round robin matrices) and SRA effects. In case of multiple groups, partial correlations are printed (controlled for group membership). The output of selfCor is also printed in the standard RR-ouput.

Value

A data frame with correlation coefficients and p values.

See Also

RR, getEffects

Examples

```
data(multiGroup)

RR.style("p")
# a single group
RR1 <- RR(ex~perceiver.id*target.id, data=multiGroup[multiGroup$group.id=="2", ], na.rm=TRUE)
selfCor(RR1)

# multiple groups
RR2 <- RR(ex~perceiver.id*target.id|group.id, data=multiGroup, na.rm=TRUE)
selfCor(RR2)
```

Index

∗ **htest**
  parCor, 7
  RR, 10
  selfCor, 18
∗ **package**
  TripleR-package, 2
∗ **univar**
  RR, 10

getEffects, 3, *7*, *14*, *18*

liking_a *(*RR*)*, 10
liking_b *(*RR*)*, 10
likingLong *(*RR*)*, 10
long2matrix, 4, *6*, *14*

matrix2long, *5*, 6, *14*
metaliking_a *(*RR*)*, 10
metaliking_b *(*RR*)*, 10
multiGroup *(*RR*)*, 10
multiLikingLong *(*RR*)*, 10
multiNarc *(*RR*)*, 10

parCor, 7, *12*, *14*
plot.RRbi *(*plot.RRuni*)*, 8
plot.RRmulti *(*plot.RRuni*)*, 8
plot.RRuni, 8, *13, 14*
plot_missings, 9, *14*

RR, *2–4*, *7*, 10, *17, 18*
RR.style, *13, 14*, 16
RR.summary, *14*, 17

selfCor, *12*, *14*, 18

TripleR *(*TripleR-package*)*, 2
TripleR-package, 2