

Package ‘roundRobinR’

May 9, 2026

Title Manipulate and Analyze Round Robin Dyadic Data

Version 1.0.1

Description Provides utilities for processing and analyzing dyadic data collected using a round-robin design, in which each person in a group rates or interacts with every other person on at least one variable. Data manipulation functions prepare datasets for dyadic data analysis by creating the actor and partner dummy variables required by the social relations model (SRM). Analysis functions implement the SRM using multilevel modeling via a custom 'nlme' covariance class ('pdSRM'), following the approach of Snijders and Kenny (1999) <[doi:10.1111/j.1475-6811.1999.tb00204.x](https://doi.org/10.1111/j.1475-6811.1999.tb00204.x)> and Knight and Humphrey (2019) <[doi:10.1037/0000115-019](https://doi.org/10.1037/0000115-019)>. The package estimates group, actor, partner, and relationship variance components along with generalized and dyadic reciprocity correlations, and supports both null and fixed-effects models.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports data.table (>= 1.14.0), nlme (>= 3.1-150), stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://github.com/andrewpknight/roundRobinR>

BugReports <https://github.com/andrewpknight/roundRobinR/issues>

NeedsCompilation no

Author Andrew Knight [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-6700-1903>>)

Maintainer Andrew Knight <knightap@wustl.edu>

Repository CRAN

Date/Publication 2026-04-21 18:52:23 UTC

Contents

coef.pdSRM	2
corMatrix.pdSRM	3
createDummies	4
pdConstruct.pdSRM	5
pdMatrix.pdSRM	6
pdSRM	7
sampleDyadData	8
srmPseudoRSq	9
srmRun	10
srmVarPct	11
summary.pdSRM	12
Index	14

coef.pdSRM	<i>Extract Coefficients from a pdSRM Object</i>
------------	---

Description

Internal method called by `coef.pdMat` to extract the three underlying SRM parameters: actor standard deviation, partner standard deviation, and actor-partner correlation.

Usage

```
## S3 method for class 'pdSRM'
coef(object, unconstrained = TRUE, ...)
```

Arguments

object	an object inheriting from pdSRM
unconstrained	logical; if TRUE (default) the unconstrained parameterization is returned; if FALSE the three named SRM parameters are returned
...	additional arguments (currently unused)

Value

a named numeric vector with elements `std. dev-a`, `std. dev-p`, and `corr.`

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
o <- nlme::lme(
  liking ~ 1,
```

```

random = list(groupId = nlme::pdBlocked(list(
  nlme::pdIdent(~1),
  pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
))),
correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
data = d,
na.action = stats::na.omit
)

```

corMatrix.pdSRM

Extract Correlation Matrix from a pdSRM Object

Description

Internal method called by `corMatrix` to reconstruct the correlation matrix from the compressed SRM parameters.

Usage

```

## S3 method for class 'pdSRM'
corMatrix(object, ...)

```

Arguments

object	an object inheriting from pdSRM
...	additional arguments (currently unused)

Value

the correlation matrix corresponding to the positive-definite matrix represented by object, with a "stdDev" attribute giving the standard deviations

Examples

```

d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
o <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)

```

 createDummies

Create Dummy Variables for the Social Relations Model

Description

Generates the actor and partner dummy variables required to fit the Social Relations Model (SRM) using multilevel modeling, following the approach of Snijders and Kenny (1999). The function produces N actor dummies and N partner dummies, where N is the maximum group size in the dataset.

Usage

```
createDummies(
  group.id,
  act.id,
  part.id,
  d,
  include.self = FALSE,
  merge.original = FALSE
)
```

Arguments

<code>group.id</code>	string; name of the group identifier variable
<code>act.id</code>	string; name of the actor identifier variable
<code>part.id</code>	string; name of the partner identifier variable
<code>d</code>	a <code>data.frame</code> structured in directed dyadic long-form, with one row per ordered (actor, partner) pair
<code>include.self</code>	logical; if TRUE self-ratings (actor == partner) are retained. Default is FALSE
<code>merge.original</code>	logical; if TRUE the generated identifiers and dummy variables are merged back onto the original dataset and returned together. Default is FALSE

Value

a `data.frame` containing:

`pdSRM_act_id` integer internal actor identifier
`pdSRM_part_id` integer internal partner identifier
`pdSRM_act_num` integer actor slot number within group
`pdSRM_part_num` integer partner slot number within group
`pdSRM_dyad_id` integer undirected dyad identifier
`a1, a2, ...` actor dummy variables
`p1, p2, ...` partner dummy variables

If `merge.original = TRUE`, all original variables are appended.

References

Snijders, T. A. B., & Kenny, D. A. (1999). The social relations model for family data: A multilevel approach. *Personal Relationships*, 6, 471–486. doi:10.1111/j.14756811.1999.tb00204.x

Examples

```
d_out <- createDummies(
  group.id = "groupId",
  act.id   = "actId",
  part.id  = "partId",
  d        = sampleDyadData
)
head(d_out)
```

pdConstruct.pdSRM *Construct pdSRM Object*

Description

Internal method called by `pdConstruct` to initialize a pdSRM object. Enforces the SRM constraint of equal actor variances, equal partner variances, and a single actor-partner covariance.

Usage

```
## S3 method for class 'pdSRM'
pdConstruct(
  object,
  value = numeric(0),
  form = stats::formula(object),
  nam = nlme::Names(object),
  data = sys.frame(sys.parent()),
  ...
)
```

Arguments

<code>object</code>	an object inheriting from pdSRM
<code>value</code>	an optional initialization value
<code>form</code>	an optional one-sided linear formula
<code>nam</code>	an optional vector of character strings
<code>data</code>	an optional data frame in which to evaluate the variables
<code>...</code>	optional arguments passed to other methods

Value

a pdSRM object representing the SRM positive-definite matrix

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
o <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)
```

pdMatrix.pdSRM

*Extract Matrix or Square-Root Factor from a pdSRM Object***Description**

Internal method called by `pdMatrix` to reconstruct the full covariance matrix from the three stored SRM parameters (actor SD, partner SD, actor-partner correlation).

Usage

```
## S3 method for class 'pdSRM'
pdMatrix(object, factor = FALSE)
```

Arguments

object	an object inheriting from pdSRM
factor	logical; if TRUE the upper Cholesky factor is returned, otherwise the full positive-definite matrix

Value

if factor is FALSE, the positive-definite matrix represented by object; if TRUE, an upper triangular Cholesky factor with a logDet attribute

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
```

```

)
o <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)

```

pdSRM

Construct the pdSRM object

Description

Creates the positive-definite matrix class used to specify the actor-partner covariance structure of the Social Relations Model within [lme](#). This class enforces the SRM constraint that all actors share a single variance, all partners share a single variance, and a single actor-partner covariance (generalized reciprocity) is estimated.

Usage

```

pdSRM(
  value = numeric(0),
  form = NULL,
  nam = NULL,
  data = sys.frame(sys.parent())
)

```

Arguments

value	an optional initialization value, inherited from pdMat
form	an optional one-sided linear formula specifying the row/column names for the matrix
nam	an optional vector of character strings specifying the row/column names for the matrix
data	inherited from the surrounding nlme call

Value

a [pdMat](#) object representing a positive-definite matrix conforming to the SRM covariance structure

References

- Knights, A. P., & Humphrey, S. E. (2019). Dyadic data analysis. In S. E. Humphrey & J. M. LeBreton (Eds.), *The Handbook for Multilevel Theory, Measurement, and Analysis* (pp. 423–447). American Psychological Association. doi:10.1037/0000115019
- Snijders, T. A. B., & Kenny, D. A. (1999). The social relations model for family data: A multilevel approach. *Personal Relationships*, 6, 471–486. doi:10.1111/j.14756811.1999.tb00204.x

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
o <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)
```

sampleDyadData

Sample Round Robin Dataset

Description

A simulated directed dyadic dataset in long format, suitable for demonstrating the functions in this package. The dataset contains observations from a round-robin design in which each person in a group rated every other person. Ratings were collected at two time points.

Usage

```
sampleDyadData
```

Format

A data frame with 1548 rows and 16 variables:

- groupId** integer; group identification variable
- actId** integer; actor (rater) identification variable
- partId** integer; partner (ratee) identification variable
- timeId** integer; time point identifier (1 or 2)
- contact** integer; directed dyad-level measure of contact frequency

liking integer; directed dyad-level measure of liking
actEx numeric; actor-level measure of extraversion
actAg numeric; actor-level measure of agreeableness
actMale integer; actor-level binary gender indicator (1 = male)
actAge integer; actor-level age in years
partEx numeric; partner-level measure of extraversion
partAg numeric; partner-level measure of agreeableness
partMale integer; partner-level binary gender indicator (1 = male)
partAge integer; partner-level age in years
groupCohesion numeric; group-level measure of cohesion
groupEfficacy numeric; group-level measure of efficacy

Source

Simulated data created for package demonstration purposes.

srmPseudoRSq

Calculate Pseudo R-Squared Values for the Social Relations Model

Description

Computes pseudo R-squared values for each SRM variance component by comparing a null model (intercept only) with a predictor model (with fixed effects). The pseudo R-squared for each component is $(\text{null} - \text{predicted}) / \text{null}$, reflecting the proportion of each variance component explained by the fixed effects.

Usage

```
srmPseudoRSq(null.model, predict.model)
```

Arguments

null.model an lme object fitted with `pdSRM` and no fixed-effect predictors (intercept only)
predict.model an lme object fitted with `pdSRM` and one or more fixed-effect predictors; must use the same dataset and random effects structure as `null.model`

Value

a `data.frame` with three columns and four rows (Group, Actor, Partner, Dyad):

null variance component from the null model
predict variance component from the predictor model
pseudoR2 pseudo R-squared: $(\text{null} - \text{predict}) / \text{null}$

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
null_mod <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)
pred_mod <- nlme::lme(
  liking ~ actEx + partEx + contact,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)
srmPseudoRSq(null.model = null_mod, predict.model = pred_mod)
```

srmRun

Run the Social Relations Model Using Multilevel Modeling

Description

A wrapper function that fits the Social Relations Model (SRM) on a directed dyadic dataset using restricted maximum likelihood via `lme`. The function creates the necessary actor and partner dummy variables, constructs the SRM covariance structure using `pdSRM`, and returns both the raw `lme` output and a formatted variance decomposition table.

Usage

```
srmRun(dv, groupId, actId, partId, feVars = NULL, data)
```

Arguments

<code>dv</code>	string; name of the directed dyadic criterion (outcome) variable
<code>groupId</code>	string; name of the group identifier variable
<code>actId</code>	string; name of the actor identifier variable

partId	string; name of the partner identifier variable
feVars	character vector of fixed-effect predictor variable names, or NULL (default) for an intercept-only null model
data	a data.frame at the directed dyad level

Value

a named list with two elements:

lme.output the full lme model object

srm.output a data.frame from `srmVarPct` giving variances, percentages, and reciprocity correlations

References

Knight, A. P., & Humphrey, S. E. (2019). Dyadic data analysis. In S. E. Humphrey & J. M. LeBreton (Eds.), *The Handbook for Multilevel Theory, Measurement, and Analysis* (pp. 423–447). American Psychological Association. doi:10.1037/0000115019

Snijders, T. A. B., & Kenny, D. A. (1999). The social relations model for family data: A multilevel approach. *Personal Relationships*, 6, 471–486. doi:10.1111/j.14756811.1999.tb00204.x

Examples

```
o <- srmRun(
  dv      = "liking",
  groupId = "groupId",
  actId   = "actId",
  partId  = "partId",
  feVars  = c("actEx", "partEx", "contact"),
  data    = sampleDyadData[sampleDyadData$timeId == 1, ]
)
o$srm.output
```

srmVarPct

Extract Variance Decomposition from a Fitted SRM

Description

Extracts and formats the variance components and reciprocity correlations from an `lme` object fitted with the `pdSRM` covariance structure. Returns group, actor, partner, and dyadic (relationship) variances as both raw values and percentages of total variance, along with generalized reciprocity (actor-partner correlation) and dyadic reciprocity.

Usage

```
srmVarPct(object)
```

Arguments

object an lme model object fitted with [pdSRM](#)

Value

a data.frame with two columns and six rows:

variances.and.covariances Group, Actor, Partner, and Dyad variances; Generalized Reciprocity covariance; Dyadic Reciprocity covariance

percents.and.correlations variance percentages for the four components; Generalized Reciprocity correlation; Dyadic Reciprocity correlation

References

Kenny, D. A., Kashy, D. A., & Cook, W. L. (2006). *Dyadic Data Analysis*. Guilford Press.

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
o <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)
srmVarPct(o)
```

summary.pdSRM

Summarize a pdSRM Object

Description

Internal method that produces a summary.pdMat representation of a pdSRM object, used by [lme](#) when printing model output.

Usage

```
## S3 method for class 'pdSRM'
summary(object, structName = "Social Relations Model", ...)
```

Arguments

object	an object inheriting from pdSRM
structName	a character string describing the covariance structure; defaults to "Social Relations Model"
...	optional arguments passed to other methods

Value

an object of class `summary.pdMat` with additional attributes `structName` and `noCorrelation`

Examples

```
d <- createDummies(
  group.id = "groupId", act.id = "actId", part.id = "partId",
  d = sampleDyadData[sampleDyadData$timeId == 1, ],
  merge.original = TRUE
)
o <- nlme::lme(
  liking ~ 1,
  random = list(groupId = nlme::pdBlocked(list(
    nlme::pdIdent(~1),
    pdSRM(~ -1 + a1 + a2 + a3 + a4 + p1 + p2 + p3 + p4)
  ))),
  correlation = nlme::corCompSymm(form = ~1 | groupId / pdSRM_dyad_id),
  data = d,
  na.action = stats::na.omit
)
```

Index

* datasets

sampleDyadData, 8

coef.pdMat, 2

coef.pdSRM, 2

corMatrix, 3

corMatrix.pdSRM, 3

createDummies, 4

lme, 7, 10–12

pdConstruct, 5

pdConstruct.pdSRM, 5

pdMat, 7

pdMatrix, 6

pdMatrix.pdSRM, 6

pdSRM, 7, 9–12

sampleDyadData, 8

srmPseudoRSq, 9

srmRun, 10

srmVarPct, 11, 11

summary.pdSRM, 12