Package ‘discord’

May 2, 2017

Type Package

Title Functions for Discordant Kinship Modeling

Version 0.1

Date 2017-05-02

Author S. Mason Garrison; Cermet Ream

Maintainer S. Mason Garrison <s.mason.garrison@gmail.com>

Description Functions for discordant kinship modeling (and other sibling-based quasi-experimental designs). Currently, the package contains data restructuring functions; functions for generating genetically- and environmentally-informed data for kin pairs.

URL https://github.com/smasongarrison/discord

License GPL-3

LazyData TRUE

RoxygenNote 6.0.1.9000

Imports stats, dplyr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-05-02 16:09:47 UTC

R topics documented:

discord-package 2
discord_data 3
discord_regression 4
kinsim1 5
kinsim_multi 6

Index **8**

discord-package *Functions for Discordant Kinship Modeling.*

Description

Utilities and functions for discordant kinship modeling (and other sibling-based quasi-experimental designs). Also includes functions for generating genetically- and environmentally informed multivariate data for kin pairs.

Note

The release version is available through CRAN by running `install.packages('discord')`. The most recent development version is available through GitHub by running `devtools::install_github (repo = 'smasongarrison/discord')` (make sure devtools is already installed). If you're having trouble with the package, please install the development version. If this doesn't solve your problem, please create a new issue, or email Mason.

Author(s)

S. Mason Garrison (Vanderbilt Univerisity)

Maintainer: S. Mason Garrison <s.mason.garrison@gmail.com>

References

EJP

Examples

```
library(discord) #Load the package into the current R session.

## Not run:
# Install/update discord with the release version from CRAN.
install.packages('discord')

# Install/update discord with the development version from GitHub
#install.packages('devtools') #Uncomment if `devtools` isn't installed already.
devtools::install_github('smasongarrison/discord')

## End(Not run)
```

`discord_data` *Restructure Data*

Description

Restructure wide form data into analyzable data, sorted by outcome.

Usage

```
discord_data(doubleentered = F, outcome = NULL, predictors = NULL,
  sep = "", scale = T, df = NULL, id = NULL, full = T)
```

Arguments

`doubleentered` Describes whether data are double entered. Default is FALSE.

`outcome` Name of outcome variable

`predictors` Names of predictors. Default is to use all variables in `df` that are not the outcome.

`sep` The character in `df` that separates root outcome and predictors from mean and diff labelscharacter string to separate the names of the `predictors` and `outcomes` from kin identifier (1 or 2). Not `NA_character_`.

`scale` If TRUE, rescale all variables at the individual level to have a mean of 0 and a SD of 1.

`df` Dataframe with all variables in it.

`id` id variable (optional).

`full` If TRUE, returns kin1 and kin2 scores in addition to diff and mean scores. If FALSE, only returns diff and mean scores.

Value

Returns `data.frame` with the following variables:

`id` id

`outcome_1` outcome for kin1; kin1 is always greater than kin2, except when tied. Then kin1 is randomly selected from the pair

`outcome_2` outcome for kin2

`outcome_diff` difference between outcome of kin1 and kin2

`outcome_mean` mean outcome for kin1 and kin2

`predictor_i_1` predictor variable i for kin1

`predictor_i_2` predictor variable i for kin2

`predictor_i_diff` difference between predictor i of kin1 and kin2

`predictor_i_mean` mean predictor i for kin1 and kin2

discord_regression *Discord Regression*

Description

Run discord analysis on discord data.

Usage

```
discord_regression(df, outcome, predictors, sep = "_", mean = "mean",
  diff = "diff", doubleentered = F, id = NULL, format_data = F,
  scale = TRUE, ...)
```

Arguments

`df` Dataframe with all variables in it.

`outcome` Name of outcome variable.

`predictors` list of predictors to be used (names will be expanded, expanded versions should exist in `df` unless format_data=T).

`sep` The character in `df` that separates root outcome and predictors from mean and diff labels

`mean` The string in `df` that designates the mean variable derived from outcome and predictor variables

`diff` The string in `df` that designates the diff variable derived from outcome and predictor variables

`doubleentered` Describes whether data are double entered. Default is FALSE. Only used if `format_data`=T.

`id` id variable (optional). Only used if format_data=T.

`format_data` If TRUE, runs `discord_data` function.

`scale` If TRUE, rescale all variables at the individual level to have a mean of 0 and a SD of 1. Only used if format_data=T.

`...` further arguments passed to or from other methods

Value

Returns `lm` of the formula: outcome ~ predictors

kinsim1 *Simulate Biometrically informed Univariate Data*

Description

Generate paired univariate data, given ACE parameters.

Usage

```
kinsim1(r = c(1, 0.5), npg = 100, npergroup = rep(npg, length(r)),
  mu = 0, ace = c(1, 1, 1), r_vector = NULL, ...)
```

Arguments

`r`	Levels of relatedness; default is MZ and DZ twins c(1,.5)
`npg`	Sample size per group; default is 100.
`npergroup`	List of sample sizes by group; default repeats `npg` for all groups.
`mu`	Mean for generated variable; default is 0.
`ace`	Vector of variance components, ordered by c(a, c, e); default is c(1,1,1).
`r_vector`	Alternative, give vector of relatedness cofficients for entire sample.
`...`	further arguments passed to or from other methods

Value

Returns `data.frame` with the following:

`id`	id
`A1`	genetic component for kin1
`A2`	genetic component for kin2
`C1`	shared-environmental component for kin1
`C2`	shared-environmental component for kin2
`E1`	non-shared-environmental component for kin1
`E2`	non-shared-environmental component for kin2
`y1`	generated variable for kin1 with mean of `mu`
`y2`	generated variable for kin2 with mean of `mu`
`r`	level of relatedness for the kin pair

kinsim_multi *Simulate Biometrically informed Multivariate Data*

Description

Generate paired multivariate data, given ACE parameters.

Usage

```
kinsim_multi(r_all = c(1, 0.5), npg_all = 500,
  npergroup_all = rep(npg_all, length(r_all)), mu_all = 0, variables = 2,
  mu_list = rep(mu_all, variables), r_vector = NULL, ace_all = c(1, 1, 1),
  ace_list = matrix(rep(ace_all, variables), byrow = TRUE, nrow = variables),
  cov_a = 1, cov_c = 1, cov_e = 1, model = "Correlated", ...)
```

Arguments

`r_all`	Levels of relatedness; default is MZ and DZ twins c(1,.5).
`npg_all`	Sample size per group; default is 500.
`npergroup_all`	Vector of sample sizes by group; default repeats `npg_all` for all groups
`mu_all`	Mean for each generated variable; default is 0.
`variables`	Number of variables to generate; default is 2. Currently, limited to max of two variables.
`mu_list`	List of means by variable; default repeats `mu_all` for all variables
`r_vector`	Alternative, give vector of r coefficients for entire sample.
`ace_all`	Vector of variance components for each generated variable; default is c(1,1,1).
`ace_list`	Matrix of ACE variance components by variable, where each row is its own variable; default is to repeat `ace_all` for each variable.
`cov_a`	Shared variance for additive genetics (a); default is 1
`cov_c`	Shared variance for shared-environment (c); default is 1
`cov_e`	shared variance for non-shared-environment (e); default is 1
`model`	Modeling type. Default is correlated factors model "Correlated"; alternative specification will be a "Cholesky" model, where variable 1 accounts for variance in variable 2.
`...`	further arguments passed to or from other methods

Value

Returns `data.frame` with the following:

`Ai_1`	genetic component for variable i for kin1
`Ai_2`	genetic component for variable i for kin2
`Ci_1`	shared-environmental component for variable i for kin1

`Ci_2`	shared-environmental component for variable i for kin2
`Ei_1`	non-shared-environmental component for variable i for kin1
`Ei_2`	non-shared-environmental component for variable i for kin2
`yi_1`	generated variable i for kin1
`yi_2`	generated variable i for kin2
`r`	level of relatedness for the kin pair
`id`	id

Index

discord (discord-package), 2
discord-package, 2
discord_data, 3
discord_regression, 4

kinsim1, 5
kinsim_multi, 6