

Package: discord (via r-universe)

September 13, 2024

Type Package

Title Functions for Discordant Kinship Modeling

Version 1.2.1

Date 2023-09-27

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

URL <https://github.com/R-Computing-Lab/discord>,
<https://r-computing-lab.github.io/discord/>

License GPL-3 + file LICENSE

LazyData TRUE

RoxygenNote 7.2.3

Encoding UTF-8

Depends R (>= 3.50)

Imports stats

Suggests NlsyLinks, broom, dplyr, janitor, kableExtra, knitr, magrittr, rmarkdown, scales, snakecase, testthat

VignetteBuilder knitr

Repository <https://r-computing-lab.r-universe.dev>

RemoteUrl <https://github.com/r-computing-lab/discord>

RemoteRef HEAD

RemoteSha 358d2f80a5cd84b5d658692d2f452afd7cb83e62

Contents

discord-package	2
check_discord_errors	3
check_sibling_order	3

data_flu_ses	4
data_sample	4
discord_cond	5
discord_data	6
discord_regression	7
kinsim	8
kinsim_internal	10
make_mean_diffs	11

Index 12

discord-package *Functions for Discordant Kinship Modeling.*

Description

Utilities and functions for discordant kinship modeling (and other sibling-based quasi-experimental designs). Currently, the package contains data restructuring functions and functions for generating biometrically informed 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 = 'R-Computing-Lab/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](#) (Wake Forest University) [Jonathan Trattner](#) Cermet Ream
 Maintainer: S. Mason Garrison <garrissm@wfu.edu>

References

Garrison et al. (2019)

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("R-Computing-Lab/discord")

## End(Not run)
```

check_discord_errors *Check Discord Errors*

Description

This function checks for common errors in the provided data, including the correct specification of identifiers (ID, sex, race) and their existence in the data.

Usage

```
check_discord_errors(data, id, sex, race, pair_identifiers)
```

Arguments

data	The data to perform a discord regression on.
id	A unique kinship pair identifier.
sex	A character string for the sex column name.
race	A character string for the race column name.
pair_identifiers	A character vector of length two that contains the variable identifier for each kinship pair.

Value

An error message if one of the conditions are met.

check_sibling_order *Check Sibling Order*

Description

This function determines the order of sibling pairs based on an outcome variable. It function checks which of the two kinship pairs has more of a specified outcome variable. It adds a new column named 'order' to the dataset, indicating which sibling (identified as "s1" or "s2") has more of the outcome. If the two siblings have the same amount of the outcome, it randomly assigns one as having more.

Usage

```
check_sibling_order(data, outcome, pair_identifiers, row)
```

Arguments

data	The data set with kinship pairs
outcome	A character string containing the outcome variable of interest.
pair_identifiers	A character vector of length two that contains the variable identifier for each kinship pair
row	The row number of the data frame

Value

A character string signifying which familial member (1, 2, or neither) has more of the outcome.

data_flu_ses	<i>Flu Vaccination and SES Data</i>
--------------	-------------------------------------

Description

A data frame that accompanies the regression vignette. It contains data on SES and flu vaccination.

Usage

```
data_flu_ses
```

Format

A data frame.

Kinship pairs and their relatedness, SES, and flu vaccination information.

Source

NLSY/R Lab

data_sample	<i>Sample Data from NLSY</i>
-------------	------------------------------

Description

A data frame output from the NlsyLinks package that contains data for kinship pairs' height and weight.

Usage

```
data_sample
```

Format

A data frame.

Kinship pairs and their relatedness, height, and weight information.

Source

NLSY/R Lab

discord_cond

Custom Conditions for the discord package

Description

Custom Conditions for the discord package

Usage

```
discord_cond(type, msg, class = paste0("discord-", type), call = NULL, ...)
```

Arguments

type	One of the following conditions: c("error", "warning", "message")
msg	Message
class	Default is to prefix the 'type' argument with "discord", but can be more specific to the problem at hand.
call	What triggered the condition?
...	Additional arguments that can be coerced to character or single condition object.

Value

A condition for discord.

Examples

```
## Not run:

derr <- function(x) discord_cond("error", x)
dwarn <- function(x) discord_cond("warning", x)
dmess <- function(x) discord_cond("message", x)

return_class <- function(func) {
  tryCatch(func,
    error = function(cond) class(cond),
    warning = function(cond) class(cond),
    message = function(cond) class(cond)
  )
}
```

```

return_class(derr("error-class"))
return_class(dwarn("warning-class"))
return_class(dmess("message-class"))

## End(Not run)

```

discord_data

Restructure Data to Determine Kinship Differences

Description

Restructure Data to Determine Kinship Differences

Usage

```

discord_data(
  data,
  outcome,
  predictors,
  id = NULL,
  sex = "sex",
  race = "race",
  pair_identifiers,
  demographics = "both",
  coding_method = "none"
)

```

Arguments

data	The data set with kinship pairs
outcome	A character string containing the outcome variable of interest.
predictors	A character vector containing the column names for predicting the outcome.
id	Default's to NULL. If supplied, must specify the column name corresponding to unique kinship pair identifiers.
sex	A character string for the sex column name.
race	A character string for the race column name.
pair_identifiers	A character vector of length two that contains the variable identifier for each kinship pair
demographics	Indicator variable for if the data has the sex and race demographics. If both are present (default, and recommended), value should be "both". Other options include "sex", "race", or "none".
coding_method	A character string that indicates what kind of additional coding schemes should be used. Default is none. Other options include "binary" and "multi".

Value

A data frame that contains analyzable, paired data for performing kinship regressions.

Examples

```
discord_data(
  data = data_sample,
  outcome = "height",
  predictors = "weight",
  pair_identifiers = c("_s1", "_s2"),
  sex = NULL,
  race = NULL,
  demographics = "none"
)
```

discord_regression	<i>Perform a Linear Regression within the Discordant Kinship Framework</i>
--------------------	--

Description

Perform a Linear Regression within the Discordant Kinship Framework

Usage

```
discord_regression(
  data,
  outcome,
  predictors,
  demographics = NULL,
  id = NULL,
  sex = "sex",
  race = "race",
  pair_identifiers = c("_s1", "_s2"),
  data_processed = FALSE,
  coding_method = "none"
)
```

Arguments

data	The data set with kinship pairs
outcome	A character string containing the outcome variable of interest.
predictors	A character vector containing the column names for predicting the outcome.
demographics	Indicator variable for if the data has the sex and race demographics. If both are present (default, and recommended), value should be "both". Other options include "sex", "race", or "none".

id	Default's to NULL. If supplied, must specify the column name corresponding to unique kinship pair identifiers.
sex	A character string for the sex column name.
race	A character string for the race column name.
pair_identifiers	A character vector of length two that contains the variable identifier for each kinship pair
data_processed	Logical operator if data are already preprocessed by discord_data , default is FALSE
coding_method	A character string that indicates what kind of additional coding schemes should be used. Default is none. Other options include "binary" and "multi".

Value

Resulting 'lm' object from performing the discordant regression.

Examples

```
discord_regression(
  data = data_sample,
  outcome = "height",
  predictors = "weight",
  pair_identifiers = c("_s1", "_s2"),
  sex = NULL,
  race = NULL
)
```

kinsim

Simulate Biometrically informed Multivariate Data

Description

Generate paired multivariate data, given ACE parameters.

Usage

```
kinsim(
  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 = 0,
    cov_c = 0,
    cov_e = 0,
    ...
)

```

Arguments

<code>r_all</code>	Levels of relatedness; default is MZ and DZ twins <code>c(1,.5)</code> .
<code>npg_all</code>	Sample size per group; default is 500.
<code>npergroup_all</code>	Vector of sample sizes by group; default repeats <code>npg_all</code> for all groups
<code>mu_all</code>	Mean for each generated variable; default is 0.
<code>variables</code>	Number of variables to generate; default is 2. Currently, limited to max of two variables.
<code>mu_list</code>	List of means by variable; default repeats <code>mu_all</code> for all variables
<code>r_vector</code>	Alternative, give vector of r coefficients for entire sample.
<code>ace_all</code>	Vector of variance components for each generated variable; default is <code>c(1,1,1)</code> .
<code>ace_list</code>	Matrix of ACE variance components by variable, where each row is its own variable; default is to repeat <code>ace_all</code> for each variable.
<code>cov_a</code>	Shared variance for additive genetics (a); default is 0.
<code>cov_c</code>	Shared variance for shared-environment (c); default is 0.
<code>cov_e</code>	shared variance for non-shared-environment (e); default is 0.
<code>...</code>	Optional pass on additional inputs.

Value

Returns `data.frame` with the following:

<code>Ai_1</code>	genetic component for variable i for kin1
<code>Ai_2</code>	genetic component for variable i for kin2
<code>Ci_1</code>	shared-environmental component for variable i for kin1
<code>Ci_2</code>	shared-environmental component for variable i for kin2
<code>Ei_1</code>	non-shared-environmental component for variable i for kin1
<code>Ei_2</code>	non-shared-environmental component for variable i for kin2
<code>yi_1</code>	generated variable i for kin1
<code>yi_2</code>	generated variable i for kin2
<code>r</code>	level of relatedness for the kin pair
<code>id</code>	id

kinsim_internal *Simulate Biometrically informed Univariate Data*

Description

Generate paired univariate data, given ACE parameters.

Usage

```
kinsim_internal(
  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 coefficients for entire sample.
...	Optional pass on additional inputs.

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

make_mean_diffs	<i>Make Mean Differences</i>
-----------------	------------------------------

Description

This function calculates differences and means of a given variable for each kinship pair. The order of subtraction and the variables' names in the output dataframe depend on the order column set by `check_sibling_order()`. If the `demographics` parameter is set to "race", "sex", or "both", it also prepares demographic information accordingly, swapping the order of demographics as per the order column.

Usage

```
make_mean_diffs(
  data,
  id,
  sex,
  race,
  demographics,
  variable,
  pair_identifiers,
  row,
  coding_method = "none"
)
```

Arguments

<code>data</code>	The data set with kinship pairs
<code>id</code>	Default's to NULL. If supplied, must specify the column name corresponding to unique kinship pair identifiers.
<code>sex</code>	A character string for the sex column name.
<code>race</code>	A character string for the race column name.
<code>demographics</code>	Indicator variable for if the data has the sex and race demographics. If both are present (default, and recommended), value should be "both". Other options include "sex", "race", or "none".
<code>variable</code>	outcomes and predictors for manipulating the data
<code>pair_identifiers</code>	A character vector of length two that contains the variable identifier for each kinship pair
<code>row</code>	The row number of the data frame
<code>coding_method</code>	A character string that indicates what kind of additional coding schemes should be used. Default is none. Other options include "binary" and "multi".

Index

* **behavior**

discord-package, [2](#)

* **datasets**

data_flu_ses, [4](#)

data_sample, [4](#)

check_discord_errors, [3](#)

check_sibling_order, [3](#)

data_flu_ses, [4](#)

data_sample, [4](#)

discord (discord-package), [2](#)

discord-package, [2](#)

discord_cond, [5](#)

discord_data, [6](#)

discord_regression, [7](#)

kinsim, [8](#)

kinsim_internal, [10](#)

make_mean_diffs, [11](#)