Package ‘discord’

July 15, 2021

Type Package
Title Functions for Discordant Kinship Modeling
Version 1.1.0
Date 2021-07-14
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
License GPL-3
LazyData TRUE
RoxygenNote 7.1.1.9000
Encoding UTF-8
Depends R (>= 2.10)
Imports stats
Suggests testthat
NeedsCompilation no
Author S. Mason Garrison [aut, cre, cph],
Jonathan Trattner [aut],
Cermet Ream [ctb]
Maintainer S. Mason Garrison <garrissm@wfu.edu>
Repository CRAN
Date/Publication 2021-07-15 08:50:02 UTC

R topics documented:
discord-package ......................................................... 2
check_discord_errors ............................................... 3
check_sibling_order ............................................... 3
discord_data ......................................................... 4
discord_regression ............................................... 5
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

test

Examples

```r
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 for common errors in the discord regression function

Description
Check for common errors in specifying id, sex, and race columns for discord regressions.

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 which sibling has more of the outcome

Description
This function checks a column order by comparing which familial member has more of the outcome. This is done per pair (i.e. row).

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.

discord_data

Restructure Data to Determine Kinship Differences

Usage

discord_data(
  data,
  outcome,  # A character string containing the outcome variable of interest.
  predictors,  # A character vector containing the column names for predicting the outcome.
  id = "extended_id",  # A unique kinship pair identifier.
  sex = "sex",  # A character string for the sex column name.
  race = "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 = "both"  # 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".
)

Arguments

data A data frame.
outcome A character string containing the outcome variable of interest.
predictors A character vector containing the column names for predicting the outcome.
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.
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".

Value

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

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

discord_regression

Perform a Linear Regression within the Discordant Kinship Framework

Description

Perform a Linear Regression within the Discordant Kinship Framework

Usage

discord_regression(
data,  
outcome,  
predictors,  
id = "extended_id",  
sex = "sex",  
race = "race",  
pair_identifiers = c("_s1", "_s2")
)

Arguments

data A data frame.
outcome A character string containing the outcome variable of interest.
predictors A character vector containing the column names for predicting the outcome.
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

Resulting ‘lm’ object from performing the discordant regression.
Examples

discord_regression(data = sample_data,
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),
  reliability_list = NULL,
  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

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
**kinsim_internal**

**Simulate Biometrically informed Univariate Data**

Generate paired univariate data, given ACE parameters.

### Usage

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

**Value**

Returns a 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
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

---

**sample_data**

*Sample Data from NLSY*

Description

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

Usage

sample_data

Format

A data frame.

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

Source

NLSY/R Lab
Index

* **behavior**
  discord-package, 2

* **datasets**
  sample_data, 8

  check_discord_errors, 3
  check_sibling_order, 3

  discord (discord-package), 2
  discord-package, 2
  discord_data, 4
  discord_regression, 5

  kinsim, 6
  kinsim_internal, 7

  sample_data, 8