

# Package ‘discord’

May 8, 2026

**Type** Package

**Title** Functions for Discordant Kinship Modeling

**Version** 1.3

**Description** Functions for discordant kinship modeling (and other sibling-based quasi-experimental designs). Contains data restructuring functions and functions for generating biometrically informed data for kin pairs. See [Garrison and Rodgers, 2016 <[doi:10.1016/j.intell.2016.08.008](https://doi.org/10.1016/j.intell.2016.08.008)>], [Sims, Trattner, and Garrison, 2024 <[doi:10.3389/fpsyg.2024.1430978](https://doi.org/10.3389/fpsyg.2024.1430978)>] for empirical examples, and [Garrison and colleagues for theoretical work <[doi:10.1101/2025.08.25.25334395](https://doi.org/10.1101/2025.08.25.25334395)>].

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

**License** GPL-3

**LazyData** TRUE

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**Depends** R (>= 3.50)

**Imports** stats

**Suggests** NlsyLinks, ggpedigree, BGmisc, DT, broom, dplyr, grid, gridExtra, ggExtra, ggplot2, janitor, kableExtra, knitr, magrittr, rmarkdown, scales, sessioninfo, stargazer, snakecase, testthat, tidyverse, tidyr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** S. Mason Garrison [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0002-4804-6003>>),  
Jonathan Trattner [aut] (ORCID:  
<<https://orcid.org/0000-0002-1097-7603>>),  
Yoo Ri Hwang [aut],  
Cermet Ream [ctb]

**Maintainer** S. Mason Garrison <[garrissm@wfu.edu](mailto:garrissm@wfu.edu)>

**Repository** CRAN

**Date/Publication** 2026-01-18 22:20:02 UTC

## Contents

check_discord_errors . . . . .	2
check_sibling_order . . . . .	3
data_flu_ses . . . . .	3
data_sample . . . . .	4
discord_between_model . . . . .	4
discord_cond . . . . .	5
discord_data . . . . .	6
discord_regression . . . . .	8
kinsim . . . . .	9
make_mean_diffs . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

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. The 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(..., fast = FALSE)
```

**Arguments**

...                    Additional arguments to be passed to the function.  
fast                   Logical. If TRUE, uses a faster method for data processing.

**Value**

A one-row data frame with a new column order indicating which familial member (1, 2, or neither) has more of the outcome.

---

data\_flu\_ses             *Flu Vaccination and SES Data*

---

**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_between_model	<i>Perform a Between-Family Linear Regression within the Discordant Kinship Framework</i>
-----------------------	---

---

**Description**

Perform a Between-Family Linear Regression within the Discordant Kinship Framework

**Usage**

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

**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. Can be NULL if no predictors are desired.
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. Default is c("_s1", "_s2").
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".
fast	Logical. If TRUE, uses a faster method for data processing.

**Value**

Resulting 'lm' object from performing the between-family regression.

**Examples**

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

---

 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)
```

**Description**

Restructure Data to Determine Kinship Differences

**Usage**

```
discord_data(
  data,
  outcome,
  predictors,
  id = NULL,
  sex = "sex",
  race = "race",
  pair_identifiers = c("_s1", "_s2"),
  demographics = "both",
  coding_method = "none",
  fast = TRUE,
  ...
)
```

**Arguments**

<code>data</code>	The data set with kinship pairs
<code>outcome</code>	A character string containing the outcome variable of interest.
<code>predictors</code>	A character vector containing the column names for predicting the outcome. Can be NULL if no predictors are desired.
<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>pair_identifiers</code>	A character vector of length two that contains the variable identifier for each kinship pair. Default is <code>c("_s1", "_s2")</code> .
<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>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".
<code>fast</code>	Logical. If TRUE, uses a faster method for data processing.
<code>...</code>	Additional arguments to be passed to the function.

**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",
  fast = TRUE
)
```

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

**Arguments**

<code>data</code>	The data set with kinship pairs
<code>outcome</code>	A character string containing the outcome variable of interest.
<code>predictors</code>	A character vector containing the column names for predicting the outcome. Can be NULL if no predictors are desired.
<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>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>pair_identifiers</code>	A character vector of length two that contains the variable identifier for each kinship pair. Default is <code>c("_s1", "_s2")</code> .
<code>data_processed</code>	Logical operator if data are already preprocessed by <code>discord_data</code> , default is FALSE
<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".
<code>fast</code>	Logical. If TRUE, uses a faster method for data processing.

**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**

Generates paired multivariate data for kinship pairs based on specified ACE (Additive genetic, Common environment, unique Environment) parameters with covariance structure.

**Usage**

```
kinsim(
  r_all = c(1, 0.5),
  c_all = 1,
  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,
  c_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,
  id = NULL,
  ...
)
```

**Arguments**

<code>r_all</code>	Numeric vector. Levels of genetic relatedness for each group; default is <code>c(1, 0.5)</code> representing MZ and DZ twins respectively.
<code>c_all</code>	Numeric. Default shared variance for common environment; default is 1.
<code>npg_all</code>	Integer. Default sample size per group; default is 500.
<code>npergroup_all</code>	Numeric vector. Sample sizes by group; default repeats <code>npg_all</code> for all groups in <code>r_all</code> .
<code>mu_all</code>	Numeric. Default mean value for all generated variables; default is 0.
<code>variables</code>	Integer. Number of variables to generate; default is 2. Currently limited to a maximum of two variables.
<code>mu_list</code>	Numeric vector. Means for each variable; default repeats <code>mu_all</code> for all variables.
<code>r_vector</code>	Numeric vector. Alternative specification providing genetic relatedness coefficients for the entire sample; default is <code>NULL</code> .
<code>c_vector</code>	Numeric vector. Alternative specification providing shared-environmental relatedness
<code>ace_all</code>	Numeric vector. Default variance components in order <code>c(a, c, e)</code> for all variables; default is <code>c(1, 1, 1)</code> .
<code>ace_list</code>	Matrix. ACE variance components by variable, where each row represents a variable and columns are a, c, e components; default repeats <code>ace_all</code> for each variable.
<code>cov_a</code>	Numeric. Shared variance for additive genetics between variables; default is 0.
<code>cov_c</code>	Numeric. Shared variance for shared-environment between variables; default is 0.

cov_e	Numeric. Shared variance for non-shared-environment between variables; default is 0.
id	Numeric vector. Optional unique identifiers for each kinship pair;
...	Additional arguments passed to other methods.

### Details

This function extends the univariate ACE model to multivariate data, allowing simulation of correlated phenotypes across kinship pairs with different levels of genetic relatedness. It supports simulation of up to two phenotypic variables with specified genetic and environmental covariance structures.

### Value

A data frame with the following columns:

**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** Unique identifier for each kinship pair

### Examples

```
# Generate basic multivariate twin data with default parameters
twin_data <- kinsim()

# Generate data with genetic correlation between variables
correlated_data <- kinsim(cov_a = 0.5)

# Generate data for different relatedness groups with custom parameters
family_data <- kinsim(
  r_all = c(1, 0.5, 0.25), # MZ twins, DZ twins, and half-siblings
  npergroup_all = c(100, 100, 150), # Sample sizes per group
  ace_list = matrix(
    c(
      1.5, 0.5, 1.0, # Variable 1 ACE components
      0.8, 1.2, 1.0
    ), # Variable 2 ACE components
    nrow = 2, byrow = TRUE
  ),
  cov_a = 0.3, # Genetic covariance
  cov_c = 0.2 # Shared environment covariance
)
```

---

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(..., fast = FALSE)
```

**Arguments**

<code>...</code>	Additional arguments to be passed to the function.
<code>fast</code>	Logical. If TRUE, uses a faster method for data processing.

# Index

## \* datasets

data\_flu\_ses, 3

data\_sample, 4

check\_discord\_errors, 2

check\_sibling\_order, 3

data\_flu\_ses, 3

data\_sample, 4

discord\_between\_model, 4

discord\_cond, 5

discord\_data, 6

discord\_regression, 8

discord\_within\_model  
(discord\_regression), 8

kinsim, 9

make\_mean\_diffs, 12