

Package ‘dyadMLM’

July 30, 2026

Title Tools for Dyadic Multilevel Models

Version 0.1.0

Description Provides tools for dyadic multilevel modeling with linear and generalized linear mixed-effects models. It validates and prepares long-format cross-sectional and intensive longitudinal data, including ecological momentary assessment designs, for distinguishable and exchangeable dyads. It also supports datasets containing multiple observed dyad compositions. It constructs composition-aware, model-ready variables for Actor-Partner Interdependence Models (APIMs), Dyadic Score Models (DSMs), and Dyad-Individual Models (DIMs). Prepared data can be used with model engines such as 'glmmTMB' and 'brms' for Gaussian and non-Gaussian outcomes, including counts, proportions, and skewed continuous responses. Post-estimation tools compare compatible fitted models and back-transform exchangeable sum-and-difference random-effect covariance structures into member-level quantities. The APIM and DSM specifications and their relationships follow Iida et al. (2018) [doi:10.1177/0265407517725407](https://doi.org/10.1177/0265407517725407); the multilevel sum-and-difference random-effects implementation for exchangeable dyads adapts del Rosario and West (2025) [doi:10.1177/25152459251351286](https://doi.org/10.1177/25152459251351286).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 8.0.0

Suggests brms, glmmTMB, htmltools, knitr, lavaan, marginalesffects, rmarkdown, spelling, testthat (>= 3.1.7)

Config/testthat/edition 3

Imports dplyr (>= 1.1.0), pillar, rlang (>= 0.4.0), tidysselect, tibble (>= 3.0.0)

VignetteBuilder knitr, rmarkdown

Depends R (>= 4.1)

LazyData true

URL <https://pascal-kueng.github.io/dyadMLM/>,
<https://github.com/Pascal-Kueng/dyadMLM>

BugReports <https://github.com/Pascal-Kueng/dyadMLM/issues>

Language en-US

NeedsCompilation no

Author Pascal Küng [aut, cre, cph] (ORCID:
[<https://orcid.org/0000-0001-7346-9414>](https://orcid.org/0000-0001-7346-9414))

Maintainer Pascal Küng <kueng.pascal@gmail.com>

Repository CRAN

Date/Publication 2026-07-30 17:20:23 UTC

Contents

compare_nested_glmmTMB_models	2
dyads_cross	4
dyads_ild	5
dyads_nbinom_cross	6
dyads_nbinom_ild	7
prepare_dyad_data	8
recover_exchangeable_covariance	11
Index	16

compare_nested_glmmTMB_models
<i>Compare nested glmmTMB models fitted to equivalent data</i>

Description

Performs a likelihood-ratio test for two nested glmmTMB models. The models may use ordinary data frames or [dyadMLM_data](#) objects, and their calls do not need to refer to the same R object. Models may be supplied in either order. The model with fewer estimated parameters is shown first in the result.

Usage

compare_nested_glmmTMB_models(model1, model2)

Arguments

model1, model2 Two fitted glmmTMB models to compare.

Details

Both model calls must use named data-frame objects that remain available when the models are compared. The checks assume these objects have not been modified since fitting. All ordinary data columns must be identical, including their types and attributes. For `dyadMLM_data`, generated `.dy_` columns may differ, but the original columns must be identical. Ordinary and prepared data may be compared with each other. Dyad metadata are checked when both models use `dyadMLM_data`. The function also checks fitted rows, outcomes, weights and offsets, model family and link, maximum-likelihood estimation, and model convergence. Each model must use the same untransformed response column.

These checks establish that the models use equivalent observations. They cannot establish that one model is mathematically nested within the other. The caller remains responsible for supplying genuinely nested models. The usual chi-squared reference distribution may also be inappropriate when tested variance parameters are on the boundary.

Value

An anova-style data frame containing model degrees of freedom, information criteria, log-likelihoods, the likelihood-ratio statistic, and its chi-squared p-value. When printed, a short conclusion interprets the test at the 5% significance level.

Examples

```
if (requireNamespace("glmmTMB", quietly = TRUE)) {
  restricted_data <- prepare_dyad_data(
    dyads_cross,
    dyad = coupleID,
    member = personID,
    role = gender,
    # All three observed compositions in `dyads_cross` are detected and retained
    # by default. This example focuses on `female-male` dyads, so we restrict the
    # analysis here.
    keep_compositions = "female-male"
  )
  full_data <- restricted_data

  restricted_model <- glmmTMB::glmmTMB(
    closeness ~ 1 + us(1 | coupleID),
    data = restricted_data
  )
  full_model <- glmmTMB::glmmTMB(
    closeness ~ gender + us(1 | coupleID),
    data = full_data
  )

  compare_nested_glmmTMB_models(restricted_model, full_model)
}
```

dyads_cross

*Example Gaussian cross-sectional dyadic data***Description**

A simulated long-format dataset containing distinguishable female-male dyads and exchangeable female-female and male-male dyads. Each dyad has two members and each member has one row. `closeness` and `provided_support` are the member's averages across the 14 observations in `dyads_ild`.

Usage

```
dyads_cross
```

Format

A data frame with 720 rows and 6 variables:

personID Unique person identifier.

coupleID Dyad identifier.

gender Gender role, with levels female and male.

dyad_composition Observed dyad composition, with levels female_x_male, female_x_female, and male_x_male.

closeness Mean simulated Gaussian closeness score across 14 days.

provided_support Mean simulated provided-support score across 14 days.

Details

Prepare with `dyad = coupleID`, `member = personID`, and `role = gender`. These data contain three compositions. Use `keep_compositions` in `prepare_dyad_data()` when an analysis should retain only selected compositions; omit it when all supplied compositions should remain.

Source

Simulated for dyadMLM; design adapted from <https://github.com/Pascal-Kueng/05DyadicDataAnalysis>. See Küng, P. M. (2026). *Distinguishable and Exchangeable Dyads: Bayesian Multilevel Modelling* (v2.0.9). Zenodo. doi:10.5281/zenodo.20720321.

dyads_ild*Example Gaussian intensive longitudinal dyadic data*

Description

A simulated long-format dataset containing distinguishable female-male dyads and exchangeable female-female and male-male dyads. Each dyad has two members observed on 14 diary days.

Usage

dyads_ild

Format

A data frame with 10,080 rows and 7 variables:

personID Unique person identifier.

coupleID Dyad identifier.

diaryday Measurement day, from 0 through 13.

gender Gender role, with levels female and male.

dyad_composition Observed dyad composition, with levels female_x_male, female_x_female, and male_x_male.

closeness Simulated Gaussian closeness outcome.

provided_support Simulated provided-support score.

Details

Prepare with `dyad = coupleID`, `member = personID`, `role = gender`, and `time = diaryday`. These data contain three compositions. Use `keep_compositions` in `prepare_dyad_data()` when an analysis should retain only selected compositions; omit it when all supplied compositions should remain.

Source

Simulated for dyadMLM; design adapted from <https://github.com/Pascal-Kueng/05DyadicDataAnalysis>. See Küng, P. M. (2026). *Distinguishable and Exchangeable Dyads: Bayesian Multilevel Modelling* (v2.0.9). Zenodo. doi:10.5281/zenodo.20720321.

dyads_nbinom_cross *Example negative-binomial cross-sectional dyadic data*

Description

A simulated long-format dataset containing the same dyads, members, and dyad compositions as [dyads_cross](#). Each member has one row. `stress` is the member's average across the 14 observations in [dyads_nbinom_ild](#). The count outcome is a separate negative-binomial draw using the NB2 variance function; it is not an average or sum of the daily counts.

Usage

```
dyads_nbinom_cross
```

Format

A data frame with 720 rows and 6 variables:

personID Unique person identifier.

coupleID Dyad identifier.

gender Gender role, with levels female and male.

dyad_composition Observed dyad composition, with levels female_x_male, female_x_female, and male_x_male.

conflict_count Simulated number of conflictual interactions reported by a member.

stress Mean simulated stress score across 14 days.

Details

Prepare with `dyad = coupleID`, `member = personID`, and `role = gender`. These data contain three compositions. Use `keep_compositions` in [prepare_dyad_data\(\)](#) when an analysis should retain only selected compositions; omit it when all supplied compositions should remain.

Source

Simulated for dyadMLM; design adapted from <https://github.com/Pascal-Kueng/05DyadicDataAnalysis>. See Küng, P. M. (2026). *Distinguishable and Exchangeable Dyads: Bayesian Multilevel Modelling* (v2.0.9). Zenodo. doi:10.5281/zenodo.20720321.

dyads_nbinom_ild*Example negative-binomial intensive longitudinal dyadic data*

Description

A simulated long-format dataset containing the same dyads, members, diary days, and dyad compositions as [dyads_ild](#). The outcome is a count generated from a negative-binomial distribution using the NB2 variance function.

Usage

```
dyads_nbinom_ild
```

Format

A data frame with 10,080 rows and 7 variables:

personID Unique person identifier.

coupleID Dyad identifier.

diaryday Measurement day, from 0 through 13.

gender Gender role, with levels female and male.

dyad_composition Observed dyad composition, with levels female_x_male, female_x_female, and male_x_male.

conflict_count Simulated number of conflictual interactions reported by a member on that day.

stress Simulated stress score.

Details

Prepare with `dyad = coupleID`, `member = personID`, `role = gender`, and `time = diaryday`. These data contain three compositions. Use `keep_compositions` in [prepare_dyad_data\(\)](#) when an analysis should retain only selected compositions; omit it when all supplied compositions should remain.

Source

Simulated for dyadMLM; design adapted from <https://github.com/Pascal-Kueng/05DyadicDataAnalysis>. See Küng, P. M. (2026). *Distinguishable and Exchangeable Dyads: Bayesian Multilevel Modelling* (v2.0.9). Zenodo. doi:10.5281/zenodo.20720321.

prepare_dyad_data	<i>Prepare dyadic data for multilevel models</i>
-------------------	--

Description

Validates dyadic data, records the structural variables, and adds metadata and model-ready columns for dyadic multilevel model parameterizations.

Usage

```
prepare_dyad_data(
  data,
  dyad,
  member,
  role = NULL,
  time = NULL,
  predictors = NULL,
  lag1_predictors = NULL,
  model_types = "apim",
  dsm_role_order = NULL,
  temporal_decomposition = c("auto", "2l", "none"),
  set_exchangeable_compositions = NULL,
  keep_compositions = NULL,
  pool_compositions = NULL,
  incomplete_dyads = c("error", "drop"),
  missing_role = c("error", "drop"),
  seed = NULL
)
```

Arguments

data	A data frame or tibble. Data must be in long format. For cross-sectional dyadic data, each observed member of each dyad has one row. For intensive longitudinal dyadic data, each observed member of each dyad has one row per observed time point.
dyad	Column identifying the dyad.
member	Column identifying a person or the member within dyad.
role	Optional column identifying a stable member role, such as gender. Non-missing values must be consistent within each dyad x member and must not contain <code>_x_</code> . In repeated-measures data, an observed role is propagated to missing rows for the same member within a dyad. <code>missing_role</code> controls dyads in which a member has no non-missing role on any row. If no role is supplied, all dyads are treated as the same type of exchangeable dyads.
time	Optional column identifying time or measurement order of repeated measures.
predictors	Optional variables to use for temporal predictor decomposition and model-ready predictor construction.

lag1_predictors	Optional subset of predictors for which lag-1 model-ready columns should be created. Requires time to be a finite, integer-valued numeric measurement index. Lagging respects the dyad and member structure, matches observations at exactly time - 1, and does not bridge missing occasions. Only raw and within-person predictors are lagged. Stable between-person versions are not.
model_types	Model-ready column families to construct. Can contain one or more of "apim", "dim", and "dsm". "apim" creates actor and partner predictors. "dim" creates dyad-mean and within-dyad member-deviation predictors. "dsm" creates dyadic-score model predictor columns. "none" skips model-specific predictor construction after validation, composition inference, and optional temporal predictor decomposition, and must be used alone. "dim" and "dsm" must be requested in separate calls.
dsm_role_order	For model_types = "dsm", a character vector giving the two distinguishable roles in the order used for directional differences. For example, c("female", "male") defines predictor differences as female minus male and assigns the DSM role contrast +0.5 to female partners and -0.5 to male partners. Required when DSM columns are requested and must be NULL otherwise.
temporal_decomposition	Temporal decomposition strategy for predictors. "none" leaves predictors undecomposed before model-specific columns are constructed. "2l" indicates a two-level temporal predictor decomposition into within-person and between-person components. "auto" resolves to "2l" when both time and predictors are supplied, and to "none" otherwise. "2l" retains raw model-ready predictors in addition to their within-person and between-person components. For longitudinal DIM and DSM construction, raw and within-person dyadic scores are computed within each dyad occasion, while between-person scores are computed within dyads. Raw DIM and DSM dyad means are grand-mean centered. Do not include the raw, within-person, and between-person versions of the same contemporaneous predictor in one model because they are linearly dependent.
set_exchangeable_compositions	Optionally specify dyad compositions to treat as exchangeable, when their roles would otherwise imply distinguishability. Requires role. Compositions that are already exchangeable should not be listed. Each composition must be supplied as one string, using <code>_x_</code> , <code>-</code> , <code>_</code> , or whitespace () between the two role labels, for example "female_x_male", "female-male", "female_male", or "female male", in arbitrary order. To set multiple compositions, use a character vector of such strings.
keep_compositions	Optional observed dyad compositions to keep before exchangeability overrides and pooling. Requires role. Composition references use the same format as set_exchangeable_compositions. NULL keeps all observed compositions.
pool_compositions	Optionally pool exchangeable dyad compositions into a shared final composition label. Must be a named list where each name is the final composition label and each value is a character vector of composition references, for example <code>list(same_sex_couples = c("female-female", "male-male"))</code> . Only ex-

changeable compositions can be pooled. Each pool must contain at least two distinct observed compositions after composition references are resolved.

incomplete_dyads	How to handle dyads with fewer than two unique members across all rows in data. "error" stops with an error and "drop" removes the entire dyad. A dyad with more than two unique members is invalid and always causes an error, regardless of this setting.
missing_role	How to handle dyads in which at least one member has no non-missing role value on any row. A consistent non-missing role observed for a member is propagated to that member's other rows before this policy is applied. "error" stops with an error and "drop" removes the entire dyad. Conflicting non-missing roles always cause an error. Ignored when no role column is supplied.
seed	Optional seed for random .dy_member_contrast_* sign assignment in exchangeable dyads. If NULL, the current R session's RNG state is used.

Details

Data must be in long format. Cross-sectional dyadic data may contain at most one row per member within dyad. Intensive longitudinal dyadic data may contain at most one row per member and observed measurement occasion within dyad. Measured variables may contain missing values. Structural completeness is assessed across all rows. `incomplete_dyads` controls dyads with fewer than two members; dyads with more than two members always cause an error. When role is supplied, stable member roles are resolved across repeated rows before `missing_role` is applied.

Dyad composition labels are canonical: role labels are sorted alphabetically before being combined, so labels do not depend on row or member order.

Value

The original data as a tibble with class `dyadMLM_data`, `.dy_composition` and `.dy_composition_role` factor columns, `.dy_is_*` numeric indicator columns, composition-specific numeric `.dy_member_contrast_*` columns coded -1 and 1 for the two members of matching exchangeable dyads and 0 otherwise, and a `dyadMLM` attribute containing structural metadata, `dyad_compositions`, and predictor metadata such as `temporal_decompositions`, `lag1_predictors`, `apim_predictors`, and `dim_predictors`, as well as `dsm_predictors` and `dsm_role_order` when applicable.

Examples

```
data <- data.frame(
  dyad_id = c(1, 1, 2, 2, 3, 3),
  person_id = c(1, 2, 3, 4, 5, 6),
  role = c("female", "male", "female", "female", "male", "male"),
  x = c(4, 7, 5, 6, 3, 8)
)

prepared <- prepare_dyad_data(
  data,
  dyad = dyad_id,
  member = person_id,
  role = role,
```

```

    predictors = x,
    model_types = "apim"
  )

  print(prepared)

  pooled <- prepare_dyad_data(
    data,
    dyad = dyad_id,
    member = person_id,
    role = role,
    predictors = x,
    model_types = "apim",
    set_exchangeable_compositions = "female-male",
    pool_compositions = list(
      romantic_couples = c("female-female", "male-male", "female-male")
    )
  )

  print(pooled)

  ild_data <- data.frame(
    dyad_id = rep(c(1, 2), each = 4),
    person_id = rep(c(1, 2), times = 4),
    time = rep(c(1, 1, 2, 2), times = 2),
    x = c(4, 7, 5, 8, 3, 6, 4, 7)
  )

  ild_prepared <- prepare_dyad_data(
    ild_data,
    dyad = dyad_id,
    member = person_id,
    time = time,
    predictors = x,
    lag1_predictors = x,
    model_types = "apim",
    seed = 123
  )

  print(ild_prepared)

```

recover_exchangeable_covariance

Recover member-level residual covariance from exchangeable random-effect blocks

Description

Back-transforms covariance matrices from paired shared and member-difference random-effect blocks to the covariance structure of two exchangeable members. The result is on the fitted ran-

dom effects' linear-predictor scale. In non-Gaussian models, it therefore describes a Gaussian latent covariance, not response-scale residual covariance. For the model specification, derivation, and interpretation, see the [exchangeable APIM vignette](#).

Usage

```
recover_exchangeable_covariance(model, block_pairings = NULL)
```

Arguments

- | | |
|----------------|---|
| model | A fitted glmmTMB or single-response brmsfit model. For glmmTMB, only conditional-model random effects are processed. |
| block_pairings | <p>NULL (default) for automatic block matching. Otherwise, supply one block-pair specification or a list of block-pair specifications. Each pair contains:</p> <ul style="list-style-type: none"> • shared_block: a single string naming the shared random-effect term copied from the fitted model formula or an equivalent selector (see Details), or NULL if the entire shared block was omitted when fitting; • difference_block: a single string naming the member-difference random-effect term or an equivalent selector, or NULL if the entire difference block was omitted; • difference_indicator: the exact name of the difference-indicator column used in difference_block. It is required when difference_block selects a block and optional when difference_block = NULL; • shared_indicator: the exact shared composition-indicator column, needed only for composition-specific blocks in mixed-dyad models. It defaults to "1", meaning that every fitted row belongs to the pair and an ordinary intercept is the shared intercept coordinate. |

Details

Automatic matching recognizes exact `.dy_member_contrast_*_arbitrary` and legacy `.dy_diff_*_arbitrary` coefficient names and first looks for the corresponding `.dy_is_*` shared block. It requires the two blocks to use the same grouping factor and the same underlying terms. Most models fitted with dyadMLM-generated columns therefore need only:

```
result <- dyadMLM::recover_exchangeable_covariance(model)
print(result)
```

Supply `block_pairings` when automatic matching is ambiguous or when a model uses custom indicators, multiple covariance levels, or deliberately omitted blocks or terms. To specify one pair with a custom difference indicator:

```
result <- dyadMLM::recover_exchangeable_covariance(
  model,
  block_pairings = list(
    shared_block = "(1 + time | coupleID)",
    difference_block = "(0 + my_diff + I(my_diff * time) | coupleID)",
    difference_indicator = "my_diff"
```

```
)
)
```

For multiple covariance levels, wrap the pairings in an outer list. For example, in a Gaussian glmmTMB model fitted with `dispformula = ~ 0`, this call recovers both a stable dyad-level covariance with an omitted difference time slope and the same-occasion partner residual covariance:

```
result <- dyadMLM::recover_exchangeable_covariance(
  model,
  block_pairings = list(
    dyad = list(
      shared_block = "(1 + diaryday | coupleID)",
      difference_block = "(0 + .dy_member_contrast_assumed_exchangeable_arbitrary | coupleID)",
      difference_indicator =
        ".dy_member_contrast_assumed_exchangeable_arbitrary"
    ),
    same_occasion = list(
      shared_block = "(1 | coupleID:diaryday)",
      difference_block = "(0 + .dy_member_contrast_assumed_exchangeable_arbitrary | coupleID:diaryday)",
      difference_indicator =
        ".dy_member_contrast_assumed_exchangeable_arbitrary"
    )
  )
)
```

At the dyad level, the fitted model includes a shared time slope but no difference time slope. Thus, the two members' time random effects are identical at this level, with correlation +1 whenever the shared slope variance is non-zero; at zero variance, the correlation is undefined. Covariances involving the diary-day slope are therefore supplied entirely by the shared block.

The random-effect terms may be copied exactly from the model formula. Equivalent backend syntax is also recognized, such as `(1 + time | group)` and `us(1 + time | group)`, or `(0 + x || group)` and `diag(0 + x | group)`. A custom difference-indicator name is supplied literally. Difference slopes may be written in either interaction order or as a simple product inside `I()`.

When a difference block is supplied and the fitted model frame retains the indicator columns, `difference_indicator` must assign -1 and +1 to the two arbitrary member positions consistently within each dyad. For composition-specific blocks, it must be zero where `shared_indicator` is zero.

For custom difference indicators supplied through `block_pairings`, the function checks whether both positions occur within each supported fitted grouping unit. It rejects coding when no group contains both positions, and warns when only some groups are one-sided, which can result from fitted-row filtering. It also warns when stable member assignments cannot be verified across repeated rows without a member identifier.

Value

An `exchangeable_rescov` object: a named list with one element per matched block pairing. Each element contains `shared_term` and `difference_term` (each a fitted term string or `NULL`), plus

the member-level variance-covariance matrix in `varcov` and its standard-deviation/correlation representation in `sdcor`, with standard deviations on the diagonal and correlations off the diagonal. Covariance dimension labels `member_1` and `member_2` denote arbitrary exchangeable positions, not substantive roles or encodings. Caller-supplied outer names are preserved; unnamed pairings receive stable names `pair_1`, `pair_2`, and so on in resolved order. For `glmmTMB`, `varcov` and `sdcor` are matrices. For `brms`, they are posterior-draw by coefficient by coefficient arrays.

What omitted blocks and terms mean

`recover_exchangeable_covariance()` only describes constraints that were already imposed when the model was fitted. It does not remove a block, set a variance to zero, or otherwise constrain the supplied model. Describe only the structure that was actually fitted.

If a term occurs in only one selected block, the function represents the missing coordinate as a structural zero:

- A term present only in the shared block has no difference component, so the two members have identical random effects for that term.
- A term present only in the difference block has no shared component, so the two members have equal-magnitude, opposite-sign random effects for that term.

Setting `difference_block = NULL` or `shared_block = NULL` applies the corresponding rule to the entire omitted block. This is valid only when that block is truly absent from the fitted model. Do not use `NULL` merely to ignore an existing block; the resulting back-transformation would be incorrect.

See the constrained-block example in the [exchangeable APIM vignette](#).

Backend note

For `glmmTMB`, random effects in non-conditional components are ignored with a warning. See the vignette's [extension to exchangeable random slopes](#) section for the manual calculation.

In `brms`, cross-sectional and same-occasion partner dependence can be represented directly with `unstr(time = member_position, gr = residual_group)`. With Gaussian outcomes, `sigma ~ 1` supplies the common residual scale. Non-Gaussian families have no `sigma` parameter here; `unstr()` instead estimates a common latent residual scale and correlation on the linear-predictor scale. Here, `member_position` identifies the same two arbitrary positions within every group, and `residual_group` identifies dyads in cross-sectional data or dyad-occasions in longitudinal data. This direct specification applies when one covariance structure is sufficient. Separate composition-specific `unstr()` structures for mixed dyad types are not currently supported in a standard single-response `brms` model. For Gaussian mixed-dyad residual covariance, use `glmmTMB`. Shared/difference blocks remain relevant for higher-level random effects and can represent latent link-scale covariance in non-Gaussian models.

See Also

The [exchangeable APIM vignette](#) for the model specification, covariance derivation, and constrained-block example. Run `vignette("apim", package = "dyadMLM")` to open the installed version.

Examples

```
if (requireNamespace("glmmTMB", quietly = TRUE)) {  
  example_data <- prepare_dyad_data(  
    dyads_cross,  
    dyad = coupleID,  
    member = personID,  
    role = gender,  
    model_types = "none",  
    # All three observed compositions in `dyads_cross` are detected and retained  
    # by default. This example focuses on `female-female` dyads, so we restrict  
    # the analysis here.  
    keep_compositions = "female-female",  
    seed = 123  
  )  
  
  model <- glmmTMB::glmmTMB(  
    closeness ~ 1 +  
      us(1 | coupleID) +  
      us(0 + .dy_member_contrast_female_x_female_arbitrary | coupleID),  
    dispformula = ~ 0,  
    data = example_data  
  )  
  
  recover_exchangeable_covariance(model)  
}
```

Index

* **datasets**

dyads_cross, [4](#)

dyads_ild, [5](#)

dyads_nbinom_cross, [6](#)

dyads_nbinom_ild, [7](#)

compare_nested_glmmTMB_models, [2](#)

dyadMLM_data, [2](#)

dyads_cross, [4](#), [6](#)

dyads_ild, [4](#), [5](#), [7](#)

dyads_nbinom_cross, [6](#)

dyads_nbinom_ild, [6](#), [7](#)

prepare_dyad_data, [8](#)

prepare_dyad_data(), [4–7](#)

recover_exchangeable_covariance, [11](#)