Package ‘r2mlm’

Type Package
Title R-Squared Measures for Multilevel Models
Version 0.2.0
Maintainer Mairead Shaw <mairead.shaw@mail.mcgill.ca>
Description Generates both total- and level-specific R-squared measures from Rights and Sterba’s (2019) <doi:10.1037/met0000184> framework of R-squared measures for multilevel models with random intercepts and/or slopes, which is based on a completely full decomposition of variance. Additionally generates graphical representations of these R-squared measures to allow visualizing and interpreting all measures in the framework together as an integrated set. This framework subsumes 10 previously-developed R-squared measures for multilevel models as special cases of 5 measures from the framework, and it also includes several newly-developed measures. Measures in the framework can be used to compute R-squared differences when comparing multilevel models (following procedures in Rights & Sterba (2020) <doi:10.1080/00273171.2019.1660605>).
License GPL-3
Encoding UTF-8
LazyData true
Imports broomExtra (>= 4.1.0), dplyr (>= 0.8.5), magrittr (>= 1.5), rlang (>= 0.4.6), stringr (>= 1.4.0), tidyselect (>= 1.0.0)
Depends lme4 (>= 1.1.23), nlme (>= 3.1.14), R (>= 3.2.0)
RoxygenNote 7.1.0
Suggests testthat, Matrix
URL https://github.com/mkshaw/r2mlm
BugReports https://github.com/mkshaw/r2mlm/issues
NeedsCompilation no
Author Mairead Shaw [aut, cre], Jason Rights [aut], Sonya Sterba [aut], Jessica Flake [aut]
**r2mlm**

Compute R-squared values for multilevel models, automatically inputting parameter estimates.

**Description**

`r2mlm` reads in a multilevel model (MLM) object generated using `lmer` or `nlme`, and outputs all relevant R-squared measures from the Rights and Sterba (2019) framework of multilevel model R-squared measures, which can be visualized together as a set using the outputted bar chart decompositions of outcome variance. That is, when predictors are cluster-mean-centered, all R-squared measures from Rights & Sterba (2019) Table 1 and decompositions from Rights & Sterba (2019) Figure 1 are outputted. When predictors are not cluster-mean-centered, the total R-squared measures from Rights & Sterba (2019) Table 5, as well as bar chart decompositions are outputted. Any number of level-1 and/or level-2 predictors is supported. Any of the level-1 predictors can have random slopes.

**Usage**

```r
r2mlm(model)
```

**Arguments**

- `model`: A model generated using `lmer` or `nlme`. Note that models using `lmer` must specify random effects at the end of the model, like so: `outcome ~ 1 + fixed_effects + (random_effects | cluster_variable)`. Anything else (e.g., `outcome ~ 1 + (random_effects | cluster_variable) + fixed_effects`) will not work.

**Details**

`r2mlm` first determines whether a given model was generated using `lmer` or `nlme`, then passes the model to helper functions that pull the raw data and parameter estimates from the model, and pass that information to `r2mlm_manual`.
Value

If the input is a valid model, then the output will be a list and associated graphical representation of R-squared decompositions. If the model is not valid, it will return an error prompting the user to input a valid model.

See Also


Other r2mlm single model functions: r2mlm_manual()

Examples

# Using lme4 for your model

# The "bobyqa" optimizer is required for this particular model to converge

model_lme4 <- lmer(satisfaction ~ 1 + salary_c + control_c + salary_m + control_m + s_t_ratio + (1 + salary_c + control_c) | schoolID, data = teachsat, REML = TRUE, control = lmerControl(optimizer = "bobyqa"))

r2mlm(model_lme4)

# Using nlme for your model

model_nlme <- lme(satisfaction ~ 1 + salary_c + control_c + salary_m + control_m + s_t_ratio, 
random = ~ 1 + salary_c + control_c | schoolID, 
data = teachsat, 
method = "REML", 
control = lmeControl(opt = "optim"))

r2mlm(model_nlme)

r2mlm_comp  
Compute R-squared differences between two multilevel models, automatically inputting parameter estimates.

Description

r2mlm_comp reads in two multilevel models (MLMs) (generated using lmer or nlme) under comparison (designated Model A and Model B), and outputs all R-squared measures in the Rights and Sterba (2019) framework for both models, as well as R-squared differences between the two models. Definitions of these R-squared difference measures are provided in Rights & Sterba (2020) Table 1; importantly, to detect the impact of a specific kind of term (e.g., the kind of term added to Model A to form Model B), a particular target single-source R-squared difference measure from this framework is used. For instructions on how to identify which target single-source R-squared
difference measure to interpret to detect the impact of which kind of term that distinguishes Model A from B, see Rights and Sterba (2020) Table 2. Additionally, this function produces side-by-side graphical comparisons of the R-squared measures for Model A vs. Model B that can be used to visualize changes in each measure across models. This function assumes all level-1 predictors are cluster-mean-centered, for reasons described in Rights & Sterba (2020). Any number of level-1 and/or level-2 predictors is supported and any of the level-1 predictors can have random slopes. This function can be used with either the hierarchical or the simultaneous model-building approach described in Rights and Sterba (2020). This function can be used with either nested or non-nested model comparisons (in which R-squared estimates for Model A are subtracted from those for Model B).

Usage

```r
r2mlm_comp(modelA, modelB, data = NULL)
```

Arguments

- `modelA, modelB` Models generated using `lmer` or `nlme`. Note that models using `lmer` must specify random effects at the end of the model, like so: outcome ~ 1 + fixed_effects + (random_effects | cluster_variable). Anything else (e.g., outcome ~ 1 + (random_effects | cluster_variable) + fixed_effects) will not work.
- `data` Optional argument, only needed if models are not hierarchical. Dataset with rows denoting observations and columns denoting variables.

Details

Assumes that both models are fit with lmer or both models are fit with nlme.

Value

If the inputs are valid models, then the output will be a list and associated graphical representation of R-squared decompositions. If the models are not valid, the function will return an error prompting the user to input valid models.

See Also


Other r2mlm model comparison functions: `r2mlm_comp_manual()`

Examples

```r
# Using lme4 for your model
# The "bobyqa" optimizer is required for these particular models to converge

# Model A, no "salary" components included
\begin{verbatim}
modelA_lme4 <- lmer(satisfaction ~ 1 + control_c + control_m + s_t_ratio + (1 + control_c | schoolID), data = teachsat, REML = TRUE, control = lmerControl(optimizer = "bobyqa"))

# Model B, full model with "salary" components included
modelB_lme4 <- lmer(satisfaction ~ 1 + salary_c + control_c + salary_m + control_m + s_t_ratio + (1 + salary_c + control_c | schoolID), data = teachsat, REML = TRUE, control = lmerControl(optimizer = "bobyqa"))

# Compare models
r2mlm_comp(modelA_lme4, modelB_lme4)

# Compare models, optional data argument specified
r2mlm_comp(modelA_lme4, modelB_lme4, teachsat)

# Using nlme for your model

# Model A, no "salary" components included
modelA_nlme <- lme(satisfaction ~ 1 + control_c + control_m + s_t_ratio, random = ~1 + control_c | schoolID, data = teachsat, method = "REML", control = lmeControl(opt = "optim"))

# Model B, full model with "salary" components included
modelB_nlme <- lme(satisfaction ~ 1 + salary_c + control_c + salary_m + control_m + s_t_ratio, random = ~1 + salary_c + control_c | schoolID, data = teachsat, method = "REML", control = lmeControl(opt = "optim"))

# Compare models
r2mlm_comp(modelA_nlme, modelB_nlme)

# Compare models, optional data argument specified
r2mlm_comp(modelA_nlme, modelB_nlme, teachsat)
\end{verbatim}

\textbf{Description}

\texttt{r2mlm\_comp\_manual} reads in raw data and multilevel model (MLM) parameter estimates from two separate models under comparison (designated Model A and Model B), and outputs all R-squared differences.
measures in the Rights and Sterba (2019) framework for both models, as well as R-squared differences between the two models. Definitions of these R-squared difference measures are provided in Rights & Sterba (2020) Table 1; importantly, to detect the impact of a specific kind of term (e.g., the kind of term added to Model A to form Model B), a particular target single-source R-squared difference measure from this framework is used. For instructions on how to identify which target single-source R-squared difference measure to interpret to detect the impact of which kind of term that distinguishes Model A from B, see Rights and Sterba (2020) Table 2. Additionally, this function produces side-by-side graphical comparisons of the R-squared measures for Model A vs. Model B that can be used to visualize changes in each measure across models. This function assumes all level-1 predictors are cluster-mean-centered for reasons described in Rights & Sterba (2020). Any number of level-1 and/or level-2 predictors is supported and any of the level-1 predictors can have random slopes. This function can be used with either the hierarchical or the simultaneous model-building approach described in Rights and Sterba (2020). This function can also be used with either nested or non-nested model comparisons (in which R-squared estimates for Model A are subtracted from those for Model B).

Usage

```r
r2mlm_comp_manual(
  data, 
  within_covs_modA, 
  between_covs_modA, 
  random_covs_modA, 
  gamma_w_modA, 
  gamma_b_modA, 
  Tau_modA, 
  sigma2_modA, 
  within_covs_modB, 
  between_covs_modB, 
  random_covs_modB, 
  gamma_w_modB, 
  gamma_b_modB, 
  Tau_modB, 
  sigma2_modB 
)
```

Arguments

data Dataset with rows denoting observations and columns denoting variables.

within_covs_modA, within_covs_modB
List of numbers corresponding to the columns in the dataset of the level-1 predictors used in the MLM (if none used, set to NULL).

between_covs_modA, between_covs_modB
List of numbers corresponding to the columns in the dataset of the level-2 predictors used in the MLM (if none used, set to NULL).

random_covs_modA, random_covs_modB
List of numbers corresponding to the columns in the dataset of the level-1 predictors that have random slopes in the MLM (if no random slopes, set to NULL).
gamma_w_modA, gamma_w_modB
Vector of fixed slope estimates for all level-1 predictors, to be entered in the order of the predictors listed by within_covs (if none, set to NULL).

gamma_b_modA, gamma_b_modB
Vector of fixed intercept estimate (if applicable; see has_intercept below) and fixed slope estimates for all level-2 predictors, to be entered intercept first (if applicable) followed by level-2 slopes in the order listed by between_covs (if none, set to NULL).

Tau_modA, Tau_modB
Random effect covariance matrix; note that the first row/column denotes the intercept variance and covariances (if intercept is fixed, set all to 0) and each subsequent row/column denotes a given random slope's variance and covariances (to be entered in the order listed by random_covs).

sigma2_modA, sigma2_modB
Level-1 residual variance.

Value
If the inputs are valid models, then the output will be a list and associated graphical representation of R-squared decompositions.

See Also

Other r2mlm model comparison functions: r2mlm_comp()

Examples

```r
# Model A: no "salary" components included
modelA <- lmer(satisfaction ~ 1 + control_c + control_m + s_t_ratio + (1 + control_c | schoolID), data = teachsat, REML = TRUE, control = lmerControl(optimizer = "bobyqa"))

# Model B: full model with "salary" components included
modelB <- lmer(satisfaction ~ 1 + salary_c + control_c + salary_m + control_m + s_t_ratio + (1 + salary_c + control_c | schoolID), data = teachsat, REML = TRUE, control = lmerControl(optimizer = "bobyqa"))

r2mlm_comp_manual(data = teachsat, within_covs_modA = c(4), between_covs_modA = c(6, 8), random_covs_modA = c(4), gamma_w_modA = c(2.68263), gamma_b_modA = c(19.6868596, 3.61309, -0.42385), Tau_modA = matrix(c(26.882, -0.298, -0.298, 3.536), 2, 2), sigma2_modA = 53.522, within_covs_modB = c(5, 4), between_covs_modB = c(7, 6, 8), random_covs_modB = c(5, 4), gamma_w_modB = c(1.55160, 2.69277), gamma_b_modB = c(19.68596, 1.45138, 3.68330, -0.37230),
```

\[ \text{Tau\_modB} = \text{matrix}(c(18.548, -0.676, -0.396, -0.676, 1.065, -0.143, -0.396, -0.143, 3.612), 3, 3), \text{sigma2\_modB} = 39.821) \]

---

## Description

`r2mlm_manual` takes as input raw data and parameter estimates from a multilevel model, and outputs all relevant R-squared measures from the Rights and Sterba (2019) framework of R-squared measures for multilevel models, which can be visualized together as a set using the outputted bar chart decompositions of outcome variance. That is, when predictors are cluster-mean-centered, all R-squared measures from Rights & Sterba (2019) Table 1 and decompositions from Rights & Sterba (2019) Figure 1 are outputted. When predictors are not cluster-mean-centered, the total R-squareds from Rights & Sterba (2019) Table 5, as well as bar chart decompositions are outputted. Any number of level-1 and/or level-2 predictors is supported. Any of the level-1 predictors can have random slopes.

## Usage

```r
r2mlm_manual(
  data,
  within_covs,
  between_covs,
  random_covs,
  gamma_w,
  gamma_b,
  Tau,
  sigma2,
  has_intercept = TRUE,
  clustermeancentered = TRUE
)
```

## Arguments

- **data**: Dataset with rows denoting observations and columns denoting variables.
- **within_covs**: List of numbers corresponding to the columns in the dataset of the level-1 predictors used in the MLM (if none used, set to NULL).
- **between_covs**: List of numbers corresponding to the columns in the dataset of the level-2 predictors used in the MLM (if none used, set to NULL).
- **random_covs**: List of numbers corresponding to the columns in the dataset of the level-1 predictors that have random slopes in the MLM (if no random slopes, set to NULL).
- **gamma_w**: Vector of fixed slope estimates for all level-1 predictors, to be entered in the order of the predictors listed by within_covs (if none, set to NULL).
gamma_b  Vector of fixed intercept estimate (if applicable; see has_intercept below) and fixed slope estimates for all level-2 predictors, to be entered intercept first (if applicable) followed by level-2 slopes in the order listed by between_covs (if none, set to NULL).

Tau  Random effect covariance matrix; note that the first row/column denotes the intercept variance and covariances (if intercept is fixed, set all to 0) and each subsequent row/column denotes a given random slope’s variance and covariances (to be entered in the order listed by random_covs).

sigma2  Level-1 residual variance.

has_intercept  If set to TRUE, the first element of gamma_b is assumed to be the fixed intercept estimate; if set to FALSE, the first element of gamma_b is assumed to be the first fixed level-2 predictor slope; set to TRUE by default.

clustermeancentered  If set to TRUE, all level-1 predictors (indicated by the within_covs list) are assumed to be cluster-mean-centered and function will output all decompositions; if set to FALSE, function will output only total decompositions (see Description above); set to TRUE by default.

Value  If the input is valid, then the output will be a list and associated graphical representation of R-squared decompositions. If the input is not valid, it will return an error.


Other r2mlm single model functions: r2mlm()

Examples  # The bobyqa optimizer is required for this model to converge in lme4

model <- lmer(satisfaction ~ 1 + salary_c + control_c + salary_m + control_m + s_t_ratio + (1 + salary_c + control_c | schoolID), data = teachsat, REML = TRUE, control = lmerControl(optimizer = "bobyqa"))
r2mlm_manual(data = teachsat, within_covs = c(5, 4), between_covs = c(7, 6, 8), random_covs = c(5, 4), gamma_w = c(1.55160, 2.69277), gamma_b = c(19.68596, 1.45138, 3.68630, -0.37230), Tau = matrix(c(18.548, -0.676, -0.396, -0.676, 1.064, -0.143, -0.396, -0.143, 3.612), 3, 3), sigma2 = 39.821, has_intercept = TRUE, clustermeancentered = TRUE)
Description
A simulated dataset containing information about teacher job satisfaction. Teachers clustered within schools.

Usage
teachsat

Format
A data frame with 9000 rows and 8 columns:
schoolID  school identification number
teacherID  teacher identification number
satisfaction numerical rating of teacher job satisfaction
control_c  school-mean-centered rating of teacher’s reported control over curriculum
salary_c   school-mean-centered teacher’s salary (in thousands of dollars)
control_m  grand-mean-centered school-mean rating of teacher’s reported control over curriculum
salary_m   grand-mean-centered school-mean teacher’s salary (in thousands of dollars)
s_t_ratio  grand-mean-centered student to teacher ratio for the school (total number of students/
total number of teachers)
Index

* datasets
  teachsat, 10
* r2mlm model comparison functions
  r2mlm_comp, 3
  r2mlm_comp_manual, 5
* r2mlm single model functions
  r2mlm, 2
  r2mlm_manual, 8

lmer, 2–4

nlme, 2–4

r2mlm, 2, 9
r2mlm_comp, 3, 7
r2mlm_comp_manual, 4, 5
r2mlm_manual, 2, 3, 8

teachsat, 10