Package: MRStdCRT (via r-universe)

October 30, 2026

Type Package

Title Model-Robust Standardization in Cluster-Randomized Trials

Version 0.1.3

Description Implements model-robust standardization for cluster-randomized trials (CRTs). Provides functions that standardize user-specified regression models to estimate marginal treatment effects. The targets include the cluster-average and individual-average treatment effects, with utilities for variance estimation and example simulation datasets. Methods are described in Li, Tong, Fang, Cheng, Kahan, and Wang (2025) <doi:10.1002/sim.70270>. Inverse-cluster-size weighted outcome models are available for generalized estimating equations and linear or generalized linear mixed models. The outcome models can use common fitting weights for both estimands or inverse-cluster-size weights for the cluster-average effect and no weights for the individual-average effect. Inference retains the original standardization framework and paired cluster jackknife. Related cluster-size saturated models are described in the arXiv preprint by Lee, Harhay, and Li (2026) <doi:10.48550/arXiv.2608.28943>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports dplyr (>= 1.0.0), geepack (>= 1.3-2), lme4 (>= 1.1-25), nlme (>= 3.1-150), magrittr (>= 2.0.0), rlang (>= 1.0.0), stats, WeMix (>= 4.0.3)

VignetteBuilder knitr

URL https://github.com/deckardt98/MRStdCRT

BugReports https://github.com/deckardt98/MRStdCRT/issues

Suggests knitr, rmarkdown

NeedsCompilation no

Author Jiaqi Tong [aut], Changjun Li [aut, cre], Xi Fang [aut], Chao Cheng [aut], Kenneth M. Lee [aut], Bingkai Wang [aut], Fan Li [aut]

Maintainer Changjun Li <changjun.li@yale.edu>

Config/roxygen2/version 8.1.1

Repository https://cran.r-universe.dev

Date/Publication 2026-10-30 19:31:41 UTC

RemoteUrl https://github.com/cran/MRStdCRT

RemoteRef HEAD

RemoteSha 54265c8780b8fb9cc88b4c561583b2df826029cd

Contents

data_sim_binary 2
data_sim_continuous 3
MRStdCRT_fit 4
ppact 8
summary.MRS_obj 9

Index **11**

`data_sim_binary` *Example Dataset: Simulated CRT (binary outcome)*

Description

A simulated dataset for demonstrating **MRStdCRT** with a binary outcome. Treatment is assigned at the cluster level and is constant within cluster.

Usage

```
data(data_sim_binary)
```

Format

A data frame with the following variables (10 columns):

A Cluster-level treatment assignment (0/1), constant within cluster.

H1 Cluster-level covariate 1.

H2 Cluster-level covariate 2.

N Cluster size recorded on each row (repeats within cluster).

X1 Individual-level covariate 1 (numeric).

X2 Individual-level covariate 2 (numeric or binary coded 0/1).

Y Observed binary outcome (0/1).

Y0 Potential outcome under control (0/1).

Y1 Potential outcome under treatment (0/1).

cluster_id Cluster identifier (integer or factor), constant within cluster.

Source

Simulated data included with the package for examples.

Examples

```
data(data_sim_binary)
head(data_sim_binary)
with(data_sim_binary, table(A, Y))
```

---

data_sim_continuous *Example Dataset: Simulated CRT (continuous outcome)*

Description

A simulated dataset for demonstrating **MRStdCRT** with a continuous outcome. Treatment is assigned at the cluster level and is constant within cluster.

Usage

```
data(data_sim_continuous)
```

Format

A data frame with the following variables (10 columns):

A Cluster-level treatment assignment (0/1), constant within cluster.

H1 Cluster-level covariate 1.

H2 Cluster-level covariate 2.

N Cluster size recorded on each row (repeats within cluster).

X1 Individual-level covariate 1 (numeric).

X2 Individual-level covariate 2 (numeric or binary coded 0/1).

Y Observed continuous outcome.

Y0 Potential outcome under control (continuous).

Y1 Potential outcome under treatment (continuous).

cluster_id Cluster identifier (integer or factor), constant within cluster.

Source

Simulated data included with the package for examples.

Examples

```
data(data_sim_continuous)
head(data_sim_continuous)
table(data_sim_continuous$cluster_id)
```

`MRStdCRT_fit` *Model-robust Standardization Estimators for the Cluster Randomized Trials*

Description

This function performs cluster randomized trials (CRT) analysis using model-robust standardization estimators to estimate the cluster-average and individual-average treatment effect. It handles different outcome mean models (GLM, LMM, GEE, GLMM) and supports continuous, binary, and count outcomes with options for different correlation structures and scales (risk difference, risk ratio and odds ratio).

Usage

```
MRStdCRT_fit(
  formula,
  data,
  cluster,
  trt,
  trtprob = rep(0.5, nrow(data)),
  method,
  family = gaussian(link = "identity"),
  corstr,
  scale,
  alpha = 0.05,
  model_weights = c("none", "inverse_cluster_size", "estimand_specific"),
  inference_scale = c("original", "log")
)
```

Arguments

`formula`	A formula with a single untransformed response column and an intercept. Supported terms are untransformed covariate main effects and two-way interactions between treatment and one covariate; each interacting covariate must also appear as a main effect. The treatment main effect is added automatically. For example, y ~ x + A:x and y ~ A * x are supported when trt = "A". Transformations, interactions between covariates, higher-order interactions, dot notation, no-intercept formulas, and term subtraction are not supported. Compute transformed covariates as separate data columns before specifying the formula.
`data`	A data frame where categorical variables should already be converted to dummy variables. Cluster, treatment, outcome and covariates must occupy distinct columns.

`cluster` A string representing the column name of the cluster ID in the data frame. Cluster IDs must not be missing.

`trt` A string representing the column name of the treatment assignment per cluster (0=control, 1=treatment). Values must be numeric or logical 0/1, without missing values, and constant within each cluster.

`trtprob` A numeric vector of probabilities of assignment to treatment (A = 1), with one value per data row or per cluster. Values must be finite, strictly between 0 and 1, and constant within each cluster. Row-level values must follow the order of data. Named cluster-level values are matched by cluster ID, and their names must uniquely and exactly match all cluster IDs; unnamed values follow the order of first appearance of clusters. If NULL, the common probability is estimated as the proportion of clusters assigned to treatment and re-estimated in each jackknife sample. The default is rep(0.5, nrow(data)).

`method` A string specifying the outcome mean model. Possible values are:

- 'GLM': generalized linear model on cluster-level means (binary/continuous outcome).
- 'LMM': a Gaussian identity working model on individual-level observations; numeric binary and count outcomes can also be analyzed with this working model.
- 'GEE': marginal models fitted by generalized estimating equations.
- 'GLMM': generalized linear mixed model; Gaussian identity models are fitted by maximum likelihood as linear mixed models.

`family` A family object, defaulting to `gaussian(link = "identity")`. Common choices are Gaussian identity, `binomial(link = "logit")` for binary outcomes, and `poisson(link = "log")` for counts. LMM uses a Gaussian identity working model. GLMM supports these three family/link combinations. Unweighted GLM and GEE pass the family to their fitting routines; `gaussian(link = "logit")` is also an unweighted cluster-summary GLM working-model option for binary outcomes. See Details for the supported combinations under weighted fitting.

`corstr` A string specifying the GEE working correlation structure. Required for GEE; it can be omitted for other methods. Weighted GEE supports "independence" and "exchangeable".

`scale` A string specifying the risk measure of interest. Can be 'RD' (risk difference), 'RR' (relative risk), or 'OR' (odds ratio).

`alpha` A numeric value for the type-I error rate. Default is 0.05.

`model_weights` Outcome-model fitting weights. The default, `"none"`, retains the existing unweighted procedure. `"inverse_cluster_size"` fits one weighted GEE, LMM, or GLMM working model and uses its predictions for both the cluster-average and individual-average MRS estimators. `"estimand_specific"` uses the weighted model for cATE and an unweighted model for iATE, with paired cluster-jackknife inference. Both nondefault options support GEE, LMM and GLMM; cluster-summary GLM supports only `"none"`. See Details.

`inference_scale`

Inference scale. The default, `"original"`, retains confidence intervals and p-values on the original effect scale. For RR and OR, `"log"` uses a jackknife of the

log-transformed ratios and exponentiates the confidence limits. Point estimates, reported original-scale standard errors, and the paired ICS test are unchanged. The log-scale standard errors are returned separately as `log_se`. The log option is not available for RD. See Details.

Details

With `model_weights = "inverse_cluster_size"`, one weighted outcome model supplies predictions for both estimands. With `"estimand_specific"`, two models of the specified method and family are fitted to the same data and formula: the inverse-cluster-size weighted model supplies cATE, and the unweighted model supplies iATE. Cluster sizes are computed from observed membership. Outcome values and `trtprob` are unchanged, and cluster-size terms are not added to the formula.

Weighted GEE uses `geepack::geeglm()` with individual fitting weights $1/N_i$. Gaussian identity, binomial logit, and Poisson log are supported with independence or exchangeable working correlation.

Weighted LMM and GLMM use `WeMix::mix()` with one cluster random intercept, conditional individual weights of 1, and cluster weights of $1/N_i$. The criterion weights each integrated cluster log-likelihood contribution by $1/N_i$; it does not weight individual conditional likelihood contributions before integration. LMM uses a Gaussian identity working model, including when numeric outcomes are binary or counts. GLMM supports Gaussian identity, binomial logit, and Poisson log. The Gaussian GLMM and LMM use the same weighted linear-model backend. Additional WeMix centering is disabled. Full-data and jackknife fits use `max_iteration = 100` and `nQuad = 13`. Gaussian fits use the WeMix linear-model interface (`family = NULL`) and maximum likelihood. The default unweighted LMM retains its original REML procedure. Nonlinear WeMix models with a random-intercept variance near zero may fail numerically. Unverifiable fits stop; the random intercept is not removed, and no positive variance floor is imposed.

Predictions exclude fitted random effects. Gaussian predictions use the fixed-effects linear predictor. Logit GLMM predictions retain the approximation $\mathrm{expit}\{X\widehat{\beta}/\sqrt{1+3\widehat{\tau}^2/\pi^2}\}$, and Poisson GLMM predictions use $\exp\{X\widehat{\beta}+\widehat{\tau}^2/2\}$, where $\widehat{\tau}^2$ is the fitted random-intercept variance. GEE uses response-scale marginal predictions.

Both MRS estimands retain the residual correction. The cluster-average estimator gives equal weight to clusters; the individual-average estimator weights clusters by their sizes. The `"none"` and `"inverse_cluster_size"` options use one fitted model for both estimands. Every delete-one-cluster sample refits the selected outcome model or models. Jackknife variance is centered at the mean of the deletion estimates, with $m-1$ degrees of freedom. By default, inference is on the original effect scale. The ICS test uses paired cATE-minus-iATE deletion estimates and always remains on the original effect scale.

With `model_weights = "estimand_specific"`, both models are refitted after deleting the same cluster. The paired deletion contrast is $D_{-i}=\widehat{\theta}^{w}_{C,-i}-\widehat{\theta}^{u}_{I,-i}$, where w and u denote weighted and unweighted fitting. Its jackknife variance is $(m-1)/m\sum_i(D_{-i}-\overline{D}_{(-)})^2$, retaining the covariance between the estimates. The procedure requires two model fits per full or deleted sample. Each component retains its existing fitting conventions, including unweighted LMM REML and weighted LMM maximum likelihood. The unweighted component also retains the existing nonparametric fallback and warnings when its outcome-model fit fails. Invalid selected contrasts stop the analysis with the component and full-data or deletion context. No deletion is discarded. This option does not guarantee greater efficiency than common fitting weights.

With `inference_scale = "log"`, let $z_{-i} = \log(\widehat{\theta}_{-i})$, where $\widehat{\theta}_{-i}$ is the RR or OR after deleting cluster i and refitting the selected model. The log-scale standard error is $\{(m-1)/m \sum_i (z_{-i} - \overline{z}_{(-)})^2\}^{1/2}$. Confidence limits are $\exp\{\log(\widehat{\theta}) \pm t_{m-1,1-\alpha/2}\widehat{\mathrm{SE}}_{\log}\}$. The two-sided p-value tests a ratio of 1 using $\log(\widehat{\theta})/\widehat{\mathrm{SE}}_{\log}$ and a t_{m-1} distribution. This directly jackknifes the log ratios; it does not divide the original-scale standard error by the estimate.

Log inference requires valid arm-specific MRS means for each reported estimand in the full data and every deletion. For `"estimand_specific"`, these are the weighted cATE means and unweighted iATE means. OR requires means strictly between 0 and 1. RR requires finite, positive means; for binary outcomes (all full-data values are 0/1), they must also be at most 1. This outcome classification is retained in all deletion samples. Ratios and log-scale jackknife standard errors must be finite and positive, and `alpha` must be strictly between 0 and 1. Invalid estimates and confidence limits outside the finite, positive numerical range stop the analysis. Values are not truncated and deletions are not omitted. Positive confidence limits do not guarantee improved small-sample coverage.

Weighted fitting requires complete, finite outcome and covariate data. Weighted binomial outcomes must be 0/1, and Poisson outcomes must be nonnegative integers. Failed weighted fits or invalid predictions stop the analysis and identify the full-data fit or the deleted cluster. Weighted fitting is an additional working-model option; improved efficiency or equality with g-computation is not guaranteed. See the package vignette for examples and a specific GEE comparison.

Value

A list with the following components:

- `estimate`: A table of point estimates, original-scale standard errors, confidence limits, and p-values. With log inference, only confidence limits and p-values change; the standard-error column remains on the original effect scale.
- `m`: Number of clusters.
- `N`: A vector of observed cluster sizes.
- `family`: The family used for the model.
- `model`: The method used for the outcome mean model.
- `model_weights`: The selected outcome-model fitting option.
- `inference_scale`: The selected inference option.
- `log_se`: A named numeric vector of log-scale jackknife standard errors for cATE and iATE when log inference is selected; otherwise `NULL`.
- `ics`: A named numeric vector containing `statistic`, the original-scale paired ICS t statistic, and `p.value`, its two-sided p-value.
- `alpha`: The specified type-I error rate.
- `scale`: The requested effect measure.

References

Li F, Tong J, Fang X, Cheng C, Kahan BC, Wang B (2025). Model-Robust Standardization in Cluster-Randomized Trials. Statistics in Medicine, 44, e70270. doi:10.1002/sim.70270.

Lee KM, Harhay MO, Li F (2026). Saturation in G: simple & robust causal inference in cluster randomized trials with informative cluster sizes. arXiv preprint. doi:10.48550/arXiv.2608.28943.

Bailey P, Webb B, Kelley C, Nguyen T, Huo H (2023). WeMix: Weighted Mixed-Effects Models Using Multilevel Pseudo Maximum Likelihood Estimation. R package version 4.0.3. doi:10.32614/CRAN.package.WeMix.

Examples

```
utils::data("ppact", package = "MRStdCRT")

fit <- MRStdCRT_fit(
  formula = PEGS ~ AGE + FEMALE + comorbid + Dep_OR_Anx + pain_count + PEGS_bl +
    BL_benzo_flag + BL_avg_daily + satisfied_primary + n,
  data     = ppact,
  cluster  = "CLUST",
  trt      = "INTERVENTION",
  trtprob  = NULL,
  method   = "GEE",
  corstr   = "independence",
  scale    = "RR"
)
summary(fit)

weighted_fit <- MRStdCRT_fit(
  PEGS ~ AGE + PEGS_bl, data = ppact,
  cluster = "CLUST", trt = "INTERVENTION",
  method = "GEE", family = gaussian(), corstr = "independence",
  scale = "RD", model_weights = "inverse_cluster_size"
)
weighted_fit$estimate

estimand_specific_fit <- MRStdCRT_fit(
  PEGS ~ AGE + PEGS_bl, data = ppact,
  cluster = "CLUST", trt = "INTERVENTION",
  method = "GEE", family = gaussian(), corstr = "independence",
  scale = "RD", model_weights = "estimand_specific"
)
estimand_specific_fit$estimate
estimand_specific_fit$ics
```

---

ppact *Example Dataset: PPACT*

Description

The Pain Program of Active Coping and Training(PPACT) is a large-scale, mixed methods, cluster-randomized trial (CRT) to compare the effectiveness of an integrated, interdisciplinary program versus usual care in treating patients with chronic pain on long-term opioid treatment (CP-LOT). The primary outcome is the impact of pain (assessed using the PEGS)

Usage

```
ppact
```

Format

A data frame with primary outcome, cluster-level, individual level covariates:

SID Study ID

CLUST Cluster

INTERVENTION Study arm

AGE Patient age at randomization

FEMALE Participant gender

comorbid Diagnosis of 2 or more of the chronic medical conditions in 6 month prior to randomization

Dep_OR_Anx Anxiety and/or depression diagnosis in 6 months prior to randomization

pain_count Number of different pain types from which participants have diagnoses in 12 months prior to randomization

BL_benzo_flag Benzodiazepine dispensed in 6 months prior to randomization

BL_avg_daily Average morphine miligram equivalents dose per day in 6 month prior to randomization

PEGS_bl PEGS score at baseline

satisfied_primary Satisfaction with primary care services in prior 3 months

PEGS PEGS score

n cluster size

Source

ClinicalTrials.gov: NCT02113592, The manuscript of the study's main outcomes is published in the Annals of Internal Medicine (https://doi.org/10.7326/M21-1436).

`summary.MRS_obj` *Summarize a* `MRS_obj` *Fit*

Description

Print a concise summary of a model-robust standardization CRT fit, including the c-ATE and i-ATE estimates with SEs and CIs.

Usage

```
## S3 method for class 'MRS_obj'
summary(object, ...)
```

Arguments

object	An object of class MRS_obj, as returned by MRStdCRT_fit().
...	Additional arguments (currently ignored).

Value

Invisibly returns the original MRS_obj object, after printing:

- Fitting method and family, with the weighting option and the estimand-specific assignments when selected,
- Number of clusters; per-cluster sizes are available in object$N,
- A four-column table (Estimate, SE, CI, p-value) with rownames c-ATE and i-ATE. The confidence level is $100(1-\alpha)\%$, using the alpha stored in the fit. SE is always on the original effect scale. With log inference, CI and p-value use the log scale, with confidence limits transformed back to the ratio scale; the log-scale SE is available separately in object$log_se,
- The NICS test statistic and p-value.

Index

∗ **datasets**
data_sim_binary, 2
data_sim_continuous, 3
ppact, 8

data_sim_binary, 2
data_sim_continuous, 3

MRStdCRT_fit, 4

ppact, 8

summary.MRS_obj, 9