

Package: mmiCATs (via r-universe)

October 22, 2026

Title Cluster Adjusted t Statistic Applications

Version 0.2.0.9000

Description Simulation results detailed in Esarey and Menger (2019) [doi:10.1017/psrm.2017.42](https://doi.org/10.1017/psrm.2017.42) demonstrate that cluster adjusted t statistics (CATs) are an effective method for correcting standard errors in scenarios with a small number of clusters. The 'mmiCATs' package offers a suite of tools for working with CATs. The mmiCATs() function initiates a 'shiny' web application, facilitating the analysis of data utilizing CATs, as implemented in the cluster.im.glm() function from the 'clusterSEs' package. Additionally, the pwr_func_lmer() function is designed to simplify the process of conducting simulations to compare mixed effects models with CATs models. For educational purposes, the CloseCATs() function launches a 'shiny' application card game, aimed at enhancing users' understanding of the conditions under which CATs should be preferred over random intercept models.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL <https://github.com/mightymetrika/mmiCATs>

BugReports <https://github.com/mightymetrika/mmiCATs/issues>

Imports broom, broom.mixed, clubSandwich, clusterSEs, DT, ggplot2, lme4, lmerTest, MASS, mmcards, pool, robust, robustbase, RPostgres, shiny, shinythemes

Suggests pbkrtest, robustlmm (>= 3.5.0-2), testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libpq-dev zlib1g-dev

Repository <https://mightymetrika.r-universe.dev>

Date/Publication 2026-09-03 02:25:47 UTC

RemoteUrl <https://github.com/mightymetrika/mmicats>
RemoteRef HEAD
RemoteSha 964589ac7d1bf8e237fae10b1e92c318b1db6954

Contents

CloseCATs	2
cluster_data_explore	3
cluster_im_glmRob	4
cluster_im_lmRob	5
cluster_model_diagnostics	7
KenRCATs	8
kenward_roger	9
mmiCATs	10
prepare_study12_freeze	11
prepare_study3_empirical	12
pwr_func_lmer	12
pwr_func_study1	14
pwr_func_study2	17
record_study12_registration	19
run_study1_definitive	20
run_study2_definitive	20
run_study3_empirical	21
Index	23

CloseCATs	<i>CloseCATs Shiny Application</i>
-----------	------------------------------------

Description

This function creates and runs a Shiny application for the CloseCATs game. The application provides a user interface for setting up the game, dealing cards, swapping cards, and scoring the game based on statistical computations. The game involves dealing cards to players and the computer, allowing the player to swap cards in their column, and scoring the game based on the mispecification distance calculated from the processed hands.

Usage

CloseCATs()

Details

The UI allows players to input various statistical parameters and preferences for the game setup. It also provides interactive elements for dealing cards, swapping cards within a column, and scoring the game based on the calculated mispecification distance.

The main components of the Shiny application include:

- A sidebar for inputting game parameters and controls for dealing and scoring.
- A main panel for displaying game cards, swap options, and results.
- Reactive elements that update based on user interaction and game state.

Value

A Shiny app object which can be run to start the application.

Examples

```
# To run the CloseCATs Shiny application:
if(interactive()){
  CloseCATs()
}
```

cluster_data_explore *Explore Clustered Data Before Model Fitting*

Description

Produces structured pre-model summaries and plots for a clustered linear analysis with one continuous focal predictor. The function is descriptive: it does not select or recommend an estimator.

Usage

```
cluster_data_explore(formula, cluster, data)
```

Arguments

formula	A two-sided formula of the form outcome ~ predictor. The initial implementation supports one untransformed numeric predictor.
cluster	A one-sided formula such as ~ school, or a character string naming the clustering variable.
data	A data frame.

Value

A list containing overall, cluster_summary, missingness, cluster_missingness, analysis_data, and plots. plots is a named list of ggplot2 objects.

Examples

```
dat <- data.frame(
  y = rnorm(60),
  x = rnorm(60),
  id = rep(1:6, each = 10)
)
out <- cluster_data_explore(y ~ x, ~ id, dat)
out$overall
out$cluster_summary
```

cluster_im_glmRob

Cluster-Adjusted Confidence Intervals And p-Values Robust GLMs

Description

Performs cluster-adjusted inference on a robust generalized linear model object, using robust generalized linear regression within each cluster. This function is tailored for models where observations are clustered, and standard errors need adjustment for clustering. The function applies a robust generalized linear regression model to each cluster using the specified family and method, and then aggregates the results.

Usage

```
cluster_im_glmRob(
  robmod,
  dat,
  cluster,
  ci.level = 0.95,
  drop = TRUE,
  return.vcv = FALSE,
  engine = "robust",
  ...
)
```

Arguments

robmod	A robust generalized linear model object created using <code>robust::glmRob()</code> or <code>robustbase::glmrob()</code> . It must contain elements 'formula', 'family', and 'method'.
dat	A data frame containing the data used in the model.
cluster	A formula indicating the clustering variable in <code>dat</code> .
ci.level	Confidence level for the confidence intervals, default is 0.95.
drop	Logical; if TRUE, drops clusters where the model does not converge.
return.vcv	Logical; if TRUE, the variance-covariance matrix of the cluster-averaged coefficients will be returned.

engine	Set the engine to "robust" to use <code>robust::glmRob()</code> or "robustbase" to use <code>robustbase::glmrob()</code> . Default is "robust".
...	Additional arguments to be passed to <code>robust::glmRob()</code> or <code>robustbase::glmrob()</code> .

Value

An invisible list containing the following elements:

p.values A matrix of p-values for each independent variable.

ci A matrix with the lower and upper bounds of the confidence intervals for each independent variable.

vcv.hat The variance-covariance matrix of the cluster-averaged coefficients, returned if `return.vcv` is TRUE.

beta.bar The cluster-averaged coefficients, returned if `return.vcv` is TRUE.

Examples

```
iris_bin <- iris
# Create a binary variable for MPG (e.g., MPG > 20)
iris_bin$high_Sepal.Length = as.factor(ifelse(iris_bin$Sepal.Length > 5.8, 1, 0))

robout <- robustbase::glmrob(formula = high_Sepal.Length ~ Petal.Length + Petal.Width,
                             family = binomial,
                             data = iris_bin)
cluster_im_glmRob(robout, dat = iris_bin, ~Species, return.vcv = TRUE,
                  engine = "robustbase")
```

cluster_im_lmRob	<i>Cluster-Adjusted Confidence Intervals And p-Values Robust Linear Models</i>
------------------	--

Description

Performs cluster-adjusted inference on a robust linear model object, using robust linear regression within each cluster. This function is designed to handle models where observations are clustered, and standard errors need to be adjusted to account for this clustering. The function applies a robust linear regression model to each cluster and then aggregates the results.

Usage

```
cluster_im_lmRob(
  robmod,
  formula,
  dat,
  cluster,
  ci.level = 0.95,
```

```

    drop = TRUE,
    return.vcv = FALSE,
    engine = "robust",
    ...
  )

```

Arguments

<code>robmod</code>	A robust linear model object created using <code>robust::lmRob()</code> or <code>robustbase::lmrob()</code> .
<code>formula</code>	A formula or a string that can be coerced to a formula.
<code>dat</code>	A data frame containing the data used in the model.
<code>cluster</code>	A formula indicating the clustering variable in <code>dat</code> .
<code>ci.level</code>	Confidence level for the confidence intervals, default is 0.95.
<code>drop</code>	Logical; if TRUE, drops clusters where the model does not converge.
<code>return.vcv</code>	Logical; if TRUE, the variance-covariance matrix of the cluster-averaged coefficients will be returned.
<code>engine</code>	Set the engine to "robust" to use <code>robust::lmRob()</code> or "robustbase" to use <code>robustbase::lmrob()</code> . Default is "robust".
<code>...</code>	Additional arguments to be passed to the <code>robust::lmRob()</code> or the <code>robustbase::lmrob()</code> function.

Value

A list containing the following elements:

- p.values** A matrix of p-values for each independent variable.
- ci** A matrix with the lower and upper bounds of the confidence intervals for each independent variable.
- vcv.hat** The variance-covariance matrix of the cluster-averaged coefficients, returned if `return.vcv` is TRUE.
- beta.bar** The cluster-averaged coefficients, returned if `return.vcv` is TRUE.

Examples

```

form <- Sepal.Length ~ Petal.Length + Petal.Width
mod <- robust::lmRob(formula = form, dat = iris)
cluster_lm_lmRob(robmod = mod, formula = form, dat = iris, cluster = ~Species)

```

cluster_model_diagnostics

Compare Clustered Linear-Model Diagnostics Across Methods

Description

Fits selected clustered-inference methods to the same complete-case data and returns method-level inference, cluster-specific slopes, observation-level diagnostics, optional leave-one-cluster-out influence results, and plots. The function is descriptive and does not recommend or select an estimator.

Usage

```
cluster_model_diagnostics(
  formula,
  cluster,
  data,
  methods = c("rs", "ri", "cr2", "cats", "cats_trunc", "cats_robust", "cats_robustbase",
    "robust_ri", "robust_rs"),
  alpha = 0.05,
  seed = 20261101L,
  leave_one_cluster_out = FALSE
)
```

Arguments

formula	A two-sided formula of the form outcome ~ predictor.
cluster	A one-sided formula such as ~ school, or a character string naming the clustering variable.
data	A data frame.
methods	Character vector containing any of "rs", "ri", "cr2", "cats", "cats_trunc", "cats_robust", "cats_robustbase", "robust_ri", or "robust_rs".
alpha	Significance level used for confidence intervals.
seed	Non-negative integer used to make method fitting reproducible.
leave_one_cluster_out	Logical; if TRUE, refit every requested method after omitting each cluster and return estimate changes.

Details

The initial implementation supports one untransformed numeric predictor. Internally, the requested variables are mapped to the out, x, and cluster names used by the verified Study 1/Study 2 fitting helpers so that the empirical diagnostics use the same fitting and extraction rules as the simulation studies.

Value

A list containing `comparison`, `cluster_fits`, `cluster_slope_differences`, `observation_diagnostics`, `influence`, `analysis_data`, and `plots`. The function does not retain fitted model objects.

Examples

```
dat <- data.frame(
  y = rnorm(100),
  x = rnorm(100),
  id = rep(1:10, each = 10)
)
out <- cluster_model_diagnostics(
  y ~ x,
  ~ id,
  dat,
  methods = c("cr2", "cats"),
  seed = 123
)
out$comparison
```

KenRCATs

Launch KenRCATs Shiny Application

Description

This function launches a 'shiny' application for conducting power analysis simulations using CATs (Clustered Adjusted t-statistics) and Kenward-Roger methods. The app allows users to input simulation parameters, run simulations, view results, and manage data in a PostgreSQL database.

Usage

```
KenRCATs(dbname, datatable, host, port, user, password)
```

Arguments

<code>dbname</code>	Character string specifying the name of the PostgreSQL database.
<code>datatable</code>	Character string specifying the name of the table in the database.
<code>host</code>	Character string specifying the host name or IP address of the database server.
<code>port</code>	Integer specifying the port number on which the database is running.
<code>user</code>	Character string specifying the username for database connection.
<code>password</code>	Character string specifying the password for database connection.

Details

The KenRCATs function sets up a Shiny application with the following features:

- Input fields for various simulation parameters
- Ability to run power analysis simulations
- Display of simulation results
- Option to submit results to a PostgreSQL database
- Functionality to download data from the database
- Display of relevant citations

Value

A 'shiny' app object.

Examples

```
if(interactive()){  
  KenRCATs(  
    dbname = "your_database_name",  
    datatable = "your_table_name",  
    host = "localhost",  
    port = 5432,  
    user = "your_username",  
    password = "your_password"  
  )  
}
```

kenward_roger

Kenward-Roger Analysis Shiny Application

Description

A Shiny application that allows users to upload a dataset, modify variable types, and fit a mixed-effects model using the Kenward-Roger approximation for small sample inference.

Usage

```
kenward_roger()
```

Details

The application provides an interactive interface for setting up and running a mixed-effects model analysis with the Kenward-Roger method for estimating the degrees of freedom in linear mixed-effects models.

Value

A Shiny application object.

References

Kenward, M. G., & Roger, J. H. (1997). Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood. *Biometrics*, 53, 983-997.

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software*, 82(13), 1-26. <doi: 10.18637/jss.v082.i13>.

Examples

```
if (interactive()) {  
  kenward_roger()  
}
```

mmiCATs

Set Up CATs Analysis in Shiny Application

Description

This function creates a Shiny application for performing CATs (Cluster-Adjusted t-statistics) analysis. It provides a user interface for uploading a CSV file, specifying the model and additional arguments, and running the analysis. The output includes variable selection, GLM (Generalized Linear Model) summary, and results of the CATs analysis.

Usage

```
mmiCATs()
```

Details

The application allows the user to upload a dataset, specify a GLM model and additional arguments, and run CATs analysis. The UI consists of various input elements like file upload, text input, numeric input, and action buttons. The server part handles the data processing, model fitting, and execution of the CATs analysis. The application outputs include the list of variables, GLM model summary, and the results from the CATs analysis.

Value

A Shiny app object which can be run to start the application.

References

Esarey J, Menger A. Practical and Effective Approaches to Dealing With Clustered Data. *Political Science Research and Methods*. 2019;7(3):541-559. doi:10.1017/psrm.2017.42

Examples

```
# To run the Shiny app
if(interactive()){
  mmiCATs()
}
```

```
prepare_study12_freeze
```

Prepare the Prospective Study 1/2 Source Freeze

Description

Creates an immutable, non-model-fitting freeze of the exact manuscript-version source state and the frozen Study 1/2 simulation specifications. The function refuses to create a new freeze unless the Git worktree is clean, the current commit matches its configured upstream, the definitive Study 1/2 output directories are absent/empty, and every prospective scientific-design check passes.

Usage

```
prepare_study12_freeze(
  project_root = NULL,
  protocol_path = NULL,
  freeze_dir = NULL
)
```

Arguments

project_root	Source checkout root containing DESCRIPTION. If NULL, located from the working directory.
protocol_path	Approved prospective Study 1/2 protocol. If NULL, uses data-raw/study12_definitive_protocol_app
freeze_dir	Destination for the immutable source-freeze bundle. If NULL, uses data-raw/study12-results/pre-res

Details

This function does not run Study 1 or Study 2 and does not perform the external registration. After this source freeze is archived prospectively, call `record_study12_registration()` with the permanent registration location.

Value

Invisibly, the saved Study 1/2 freeze record.

```
prepare_study3_empirical
```

Prepare the Frozen Study 3 Empirical Illustration

Description

Recreates the prespecified Phase 6B empirical inputs from `lme4::sleepstudy`, or verifies and returns an existing completed freeze. The function never fits comparative Study 3 methods.

Usage

```
prepare_study3_empirical(
  project_root = NULL,
  plan_path = NULL,
  freeze_dir = NULL
)
```

Arguments

<code>project_root</code>	Project root containing DESCRIPTION and the approved Study 3 plan. If NULL, the root is located from the working directory.
<code>plan_path</code>	Path to the approved Study 3 empirical-analysis plan. If NULL, uses the project data-raw plan.
<code>freeze_dir</code>	Destination for the immutable pre-results freeze. If NULL, uses data-raw/study3-results/pre-results.

Value

Invisibly, the Study 3 freeze record.

```
pwr_func_lmer
```

Power Analysis for Clustered Data

Description

Conducts a power analysis for clustered data using simulation. This function allows for comparing the performance of different estimation methods in terms of power, rejection rate, root mean square error (RMSE), relative RMSE, coverage probability, and average confidence interval width.

Usage

```
pwr_func_lmer(
  betas = list(int = 0, x1 = -5, x2 = 2, x3 = 10),
  dists = list(x1 = stats::rnorm, x2 = stats::rbinom, x3 = stats::rnorm),
  distpar = list(x1 = list(mean = 0, sd = 1), x2 = list(size = 1, prob = 0.4), x3 =
    list(mean = 1, sd = 2)),
  N = 25,
  reps = 1000,
  alpha = 0.05,
  var_intr = "x1",
  grp = "ID",
  mod = paste0("out ~ x1 + x2 + x3 + (x3|", grp, ")"),
  catsmod = "out ~ x1 + x2 + x3",
  r_slope = "x1",
  r_int = "int",
  n_time = 20,
  mean_i = 0,
  var_i = 1,
  mean_s = 0,
  var_s = 1,
  cov_is = 0,
  mean_r = 0,
  var_r = 1,
  cor_mat = NULL,
  corvars = NULL,
  time_index = NULL
)
```

Arguments

betas	Named list of true coefficient values for the fixed effects.
dists	Named list of functions to generate random distributions for each predictor.
distpar	Named list of parameter lists for each distribution function in dists.
N	Integer specifying the number of groups.
reps	Integer specifying the number of replications for the simulation.
alpha	Numeric value specifying the significance level for hypothesis testing.
var_intr	Character string specifying the name of the variable of interest (for power calculations).
grp	Character string specifying the name of the grouping variable.
mod	Formula for fitting mixed-effects model to simulated data.
catsmod	Formula for the CATs model.
r_slope	Character string specifying the name of the random slope variable. This is the random slope when simulating data.
r_int	Character string specifying the name of the random intercept.

n_time	Integer or vector specifying the number of time points per group. If a vector, its length must equal N.
mean_i	Numeric value specifying the mean for the random intercept.
var_i	Numeric value specifying the variance for the random intercept.
mean_s	Numeric value specifying the mean for the random slope.
var_s	Numeric value specifying the variance for the random slope.
cov_is	Numeric value specifying the covariance between the random intercept and slope.
mean_r	Numeric value specifying the mean for the residual error.
var_r	Numeric value specifying the variance for the residual error.
cor_mat	Correlation matrix for correlated predictors, if any.
corvars	List of vectors, each vector containing names of correlated variables.
time_index	Either "linear" for a linear time trend, a custom function, or a character string that evaluates to a function for generating time values. If specified, 'time' should be included in betas but not in dists.

Value

A dataframe summarizing the results of the power analysis, including average coefficient estimate, rejection rate, root mean square error, relative root mean square error, coverage probability, and average confidence interval width for each method.

Examples

```
# Basic usage with default parameters
pwr_func_lmer(reps = 2)
```

pwr_func_study1

Simulation Study of Robust Cluster-Adjusted t Statistics

Description

Runs one condition from a focused simulation study comparing a correctly specified random-intercept model, a robust random-intercept mixed model, ordinary least squares with CR2 inference, ordinary cluster-adjusted t statistics (CATs), truncated CATs, and robust CATs based on either `robust::lmRob()` or `robustbase::lmrob()`.

Usage

```
pwr_func_study1(
  n_clusters = 20,
  cluster_size = 40,
  beta = 0,
  intercept = 0,
  random_intercept_sd = 1,
  residual_sd = 1,
  x_sd = 1,
  contamination = c("none", "vertical", "bad_leverage"),
  contamination_prop = 0.05,
  contamination_size = 10,
  leverage_size = 10,
  reps = 1000,
  alpha = 0.05,
  methods = c("ri", "cr2", "cats", "cats_trunc", "cats_robust", "cats_robustbase",
    "robust_ri"),
  seed = NULL,
  keep_replicates = FALSE,
  replicate_seeds = NULL
)
```

Arguments

<code>n_clusters</code>	Integer number of clusters.
<code>cluster_size</code>	Integer number of observations in each cluster.
<code>beta</code>	Numeric value of the true slope for x.
<code>intercept</code>	Numeric fixed intercept.
<code>random_intercept_sd</code>	Positive standard deviation of the random intercept.
<code>residual_sd</code>	Positive standard deviation of the residual error.
<code>x_sd</code>	Positive standard deviation of the normally distributed predictor.
<code>contamination</code>	Character string specifying the contamination condition: "none", "vertical", or "bad_leverage".
<code>contamination_prop</code>	Numeric proportion of observations contaminated within each cluster. The number contaminated is rounded to the nearest whole observation, with at least one observation selected when the proportion is greater than zero.
<code>contamination_size</code>	Positive size of the outcome contamination in residual standard deviation units.
<code>leverage_size</code>	Positive absolute size of contaminated predictor values in predictor standard deviation units.
<code>reps</code>	Integer number of simulation replications.
<code>alpha</code>	Numeric significance level.

methods	Character vector of methods to fit. Available values are "ri", "cr2", "cats", "cats_trunc", "cats_robust", "cats_robustbase", and "robust_ri".
seed	Optional non-negative integer random-number seed.
keep_replicates	Logical; if TRUE, retain the replicate-level results.
replicate_seeds	Optional integer vector of length reps giving the exact random-number seed for each replication. This is intended for deterministic sharded execution. When supplied, seed must be NULL.

Details

Data are generated from a constant-slope random-intercept model. Optional contamination is applied within every cluster after the clean data are generated. All requested methods are fit to the same data in each replication. The "ri" method uses Kenward-Roger inference and requires the suggested package `pbkrtest`. The "robust_ri" method uses robust Satterthwaite inference and requires the suggested package `robustlmm`.

Value

A list with three elements:

summary A data frame containing method-level simulation results.

replicates A data frame containing replicate-level results when `keep_replicates = TRUE`; otherwise NULL.

settings A list containing the simulation settings and replication seeds.

Rejection rates, coverage rates, failure rates, and singularity rates are reported as percentages. Their Monte Carlo standard errors are reported in percentage points. When replicate-level results are retained, robust CATs rows also include cluster-level warning, error, and dropping counts and a list column containing the cluster-specific coefficients and diagnostics.

References

Esarey, J., and Menger, A. (2019). Practical and effective approaches to dealing with clustered data. *Political Science Research and Methods*, 7(3), 541-559. doi:[10.1017/psrm.2017.42](https://doi.org/10.1017/psrm.2017.42)

Pustejovsky, J. E., and Tipton, E. (2018). Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business and Economic Statistics*, 36(4), 672-683. doi:[10.1080/07350015.2016.1247004](https://doi.org/10.1080/07350015.2016.1247004)

Examples

```
pwr_func_study1(
  n_clusters = 5,
  cluster_size = 20,
  reps = 1,
  methods = c("cr2", "cats"),
  seed = 123
)
```

pwr_func_study2	<i>Simulation Study of Robust Cluster-Adjusted t Statistics with Random Slopes</i>
-----------------	---

Description

Runs one condition from Study 2, which extends Study 1 by generating genuine cluster-to-cluster slope heterogeneity. The comparison includes a correctly specified independent random-intercept and random-slope model, the Study 1 random-intercept benchmark, robust random-intercept and independent random-slope mixed models, ordinary least squares with CR2 inference, ordinary cluster-adjusted t statistics (CATs), truncated CATs, and robust CATs based on either `robust::lmRob()` or `robustbase::lmrob()`.

Usage

```
pwr_func_study2(
  n_clusters = 20,
  cluster_size = 40,
  beta = 0,
  intercept = 0,
  random_intercept_sd = 1,
  random_slope_sd = 0.05,
  residual_sd = 1,
  x_sd = 1,
  contamination = c("none", "vertical"),
  contamination_prop = 0.05,
  contamination_size = 6,
  reps = 1000,
  alpha = 0.05,
  methods = c("rs", "ri", "cr2", "cats", "cats_trunc", "cats_robust", "cats_robustbase",
    "robust_ri", "robust_rs"),
  seed = NULL,
  keep_replicates = FALSE,
  replicate_seeds = NULL
)
```

Arguments

<code>n_clusters</code>	Integer number of clusters.
<code>cluster_size</code>	Integer number of observations in each cluster.
<code>beta</code>	Numeric value of the population mean slope for x .
<code>intercept</code>	Numeric fixed intercept.
<code>random_intercept_sd</code>	Positive standard deviation of the random intercept.
<code>random_slope_sd</code>	Positive standard deviation of the random slope.

<code>residual_sd</code>	Positive standard deviation of the residual error.
<code>x_sd</code>	Positive standard deviation of the normally distributed predictor.
<code>contamination</code>	Character string specifying the contamination condition: "none" or "vertical".
<code>contamination_prop</code>	Numeric proportion of observations contaminated within each cluster. The number contaminated is rounded to the nearest whole observation, with at least one observation selected when the proportion is greater than zero.
<code>contamination_size</code>	Positive size of the outcome contamination in residual standard deviation units.
<code>reps</code>	Integer number of simulation replications.
<code>alpha</code>	Numeric significance level.
<code>methods</code>	Character vector of methods to fit. Available values are "rs", "ri", "cr2", "cats", "cats_trunc", "cats_robust", "cats_robustbase", "robust_ri", and "robust_rs".
<code>seed</code>	Optional non-negative integer random-number seed.
<code>keep_replicates</code>	Logical; if TRUE, retain the replicate-level results.
<code>replicate_seeds</code>	Optional integer vector of length <code>reps</code> giving the exact random-number seed for each replication. This is intended for deterministic sharded execution. When supplied, <code>seed</code> must be NULL.

Details

Data are generated from a random-intercept and random-slope model with zero intercept-slope covariance. Optional symmetric vertical contamination is applied within every cluster after the clean data are generated. All requested methods are fit to the same data in each replication. The "rs" and "ri" methods use Kenward-Roger inference and require the suggested package `pbkrtest`. The "robust_ri" and "robust_rs" methods use robust Satterthwaite inference and require the suggested package `robustlmm`.

Value

A list with three elements:

summary A data frame containing method-level simulation results.

replicates A data frame containing replicate-level results when `keep_replicates = TRUE`; otherwise NULL.

settings A list containing the simulation settings and replication seeds.

Rejection rates, coverage rates, failure rates, and singularity rates are reported as percentages. Their Monte Carlo standard errors are reported in percentage points. The primary estimand is the super-population mean slope β . The realized mean of the sampled cluster slopes is retained at the replication level as a diagnostic but is not used as the simulation truth.

References

- Esarey, J., and Menger, A. (2019). Practical and effective approaches to dealing with clustered data. *Political Science Research and Methods*, 7(3), 541-559. doi:[10.1017/psrm.2017.42](https://doi.org/10.1017/psrm.2017.42)
- Pustejovsky, J. E., and Tipton, E. (2018). Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business and Economic Statistics*, 36(4), 672-683. doi:[10.1080/07350015.2016.1247004](https://doi.org/10.1080/07350015.2016.1247004)

Examples

```
pwr_func_study2(
  n_clusters = 6,
  cluster_size = 20,
  random_slope_sd = 0.10,
  reps = 1,
  methods = c("rs", "ri"),
  seed = 123
)
```

```
record_study12_registration
```

Record the Completed External Study 1/2 Registration

Description

Records the permanent location of the externally completed prospective registration after `prepare_study12_freeze()` has created and verified the source-freeze bundle. This function does not contact an external service and must be called only after the user has completed the registration/archive.

Usage

```
record_study12_registration(
  registration_location,
  project_root = NULL,
  freeze_dir = NULL
)
```

Arguments

<code>registration_location</code>	Permanent URL, DOI, or other stable identifier for the prospective registration containing the frozen protocol/bundle.
<code>project_root</code>	Source checkout root. If NULL, located automatically.
<code>freeze_dir</code>	Completed Study 1/2 source-freeze directory. If NULL, uses the project default.

Value

Invisibly, the saved registration record.

run_study1_definitive *Run the Frozen Definitive Study 1 Simulation*

Description

Runs the manuscript-version Study 1 simulation using the frozen design, deterministic replication seeds, and checkpoint/resume infrastructure. The scientific design is not configurable through this function.

Usage

```
run_study1_definitive(
  project_root = NULL,
  output_dir = NULL,
  condition_ids_to_run = NULL,
  overwrite_completed = FALSE
)
```

Arguments

project_root	Source checkout root containing DESCRIPTION. If NULL, the root is located from the working directory.
output_dir	Destination for definitive Study 1 artifacts. If NULL, uses data-raw/study1-results/definitive-stud under project_root.
condition_ids_to_run	Optional character vector of frozen condition IDs to execute. NULL means all 18 conditions. This controls scheduling only; it does not change any condition.
overwrite_completed	Logical. If FALSE, completed compatible condition checkpoints are reused.

Value

Invisibly, the combined Study 1 result object.

run_study2_definitive *Run the Frozen Definitive Study 2 Simulation*

Description

Runs the manuscript-version Study 2 simulation using the frozen design, deterministic replication seeds, and checkpoint/resume infrastructure. The scientific design is not configurable through this function.

Usage

```
run_study2_definitive(
  project_root = NULL,
  output_dir = NULL,
  condition_ids_to_run = NULL,
  overwrite_completed = FALSE
)
```

Arguments

project_root	Source checkout root containing DESCRIPTION. If NULL, the root is located from the working directory.
output_dir	Destination for definitive Study 2 artifacts. If NULL, uses data-raw/study2-results/definitive-stud under project_root.
condition_ids_to_run	Optional character vector of frozen condition IDs to execute. NULL means all 24 conditions. This controls scheduling only; it does not change any condition.
overwrite_completed	Logical. If FALSE, completed compatible condition checkpoints are reused.

Value

Invisibly, the combined Study 2 result object.

run_study3_empirical *Run the Study 3 Empirical Illustration*

Description

Runs the prespecified nine-method observed/perturbed Study 3 comparison using an existing immutable empirical freeze. The contamination map is never regenerated by this function.

Usage

```
run_study3_empirical(
  freeze_dir = NULL,
  output_dir = NULL,
  project_root = NULL,
  verify_original_sources = FALSE
)
```

Arguments

<code>freeze_dir</code>	Completed Study 3 pre-results freeze directory. If NULL, uses the project default.
<code>output_dir</code>	Study 3 result directory. Use a new directory for a post-results amended rerun.
<code>project_root</code>	Project root. If NULL, located automatically.
<code>verify_original_sources</code>	Whether current project source files must exactly match the source checksums recorded in the Phase 6B freeze. This is FALSE by default because documented post-results amendments necessarily change current source code while leaving frozen artifacts immutable.

Value

Invisibly, the saved Study 3 result record when available.

Index

CloseCATs, [2](#)
cluster_data_explore, [3](#)
cluster_im_glmRob, [4](#)
cluster_im_lmRob, [5](#)
cluster_model_diagnostics, [7](#)

KenRCATs, [8](#)
kenward_roger, [9](#)

mmiCATs, [10](#)

prepare_study12_freeze, [11](#)
prepare_study3_empirical, [12](#)
pwr_func_lmer, [12](#)
pwr_func_study1, [14](#)
pwr_func_study2, [17](#)

record_study12_registration, [19](#)
run_study1_definitive, [20](#)
run_study2_definitive, [20](#)
run_study3_empirical, [21](#)