# Package: mmiCATs (via r-universe)

September 3, 2024

**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> 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)

**RoxygenNote** 7.3.2

**URL** https://github.com/mightymetrika/mmiCATs

**BugReports** https://github.com/mightymetrika/mmiCATs/issues

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

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Repository** https://mightymetrika.r-universe.dev

**RemoteUrl** https://github.com/mightymetrika/mmicats

**RemoteRef** HEAD

**RemoteSha** 917d661bfcab40fc89eacf2918e9120f30a4210c

# Contents

---

CloseCATs                      *CloseCATs Shiny Application*

---

### 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_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 robust::glmRob() or robustbase::glmrob(). 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 dat. |
| 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 robust::glmRob() or "robustbase" to use robustbase::glmrob(). Default is "robust". |
| ... | Additional arguments to be passed to robust::glmRob() or robustbase::glmrob(). |

## 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 | *Cluster-Adjusted Confidence Intervals And p-Values Robust Linear Models* |
|---|---|

---

## 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

| | |
|---|---|
| robmod | A robust linear model object created using robust::lmRob() or robustbase::lmrob(). |
| formula | A formula or a string that can be coerced to a formula. |
| dat | A data frame containing the data used in the model. |

| | |
|---|---|
| `cluster` | A formula indicating the clustering variable in dat. |
| `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 robust::lmRob() or "robustbase" to use robustbase::lmrob(). Default is "robust". |
| `...` | Additional arguments to be passed to the robust::lmRob() or the robustbase::lmrob() 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_im_lmRob(robmod = mod, formula = form, dat = iris,cluster = ~Species)
```

---

| | |
|---|---|
| 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

| | |
|---|---|
| dbname | Character string specifying the name of the PostgreSQL database. |
| datatable | Character string specifying the name of the table in the database. |
| host | Character string specifying the host name or IP address of the database server. |
| port | Integer specifying the port number on which the database is running. |
| user | Character string specifying the username for database connection. |
| password | 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()
}
```

---

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)
```

# Index