Package 'mmiCATs'

February 21, 2024

Title Cluster Adjusted t Statistic Applications

Version 0.1.1

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

RoxygenNote 7.3.1

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

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

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

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Mackson Ncube [aut, cre], mightymetrika, LLC [cph, fnd]

Maintainer Mackson Ncube <macksonncube.stats@gmail.com>

Repository CRAN

Date/Publication 2024-02-21 08:00:02 UTC

R topics documented:

`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)
```

---

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 + (1|", 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
)
```

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	Name of the variable of interest (for power calculations) as a string.
grp	Name of the grouping variable as a string.
mod	Formula for the mixed-effects model.
catsmod	Formula for the CATs model.
r_slope	Name of the random slope variable as a string.
r_int	Name of the random intercept as a string.
n_time	Integer specifying the number of time points per group.
mean_i	Mean for the random intercept.
var_i	Variance for the random intercept.
mean_s	Mean for the random slope.
var_s	Variance for the random slope.
cov_is	Covariance between the random intercept and slope.
mean_r	Mean for the residual error.
var_r	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.

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

```
pwr_func_lmer(reps = 2)
```

Index