Package ‘holi’

July 15, 2024

Title Higher Order Likelihood Inference Web Applications
Version 0.1.0

Description Higher order likelihood inference is a promising approach for analyzing small sample size data. The 'holi' package provides web applications for higher order likelihood inference. It currently supports linear, logistic, and Poisson generalized linear models through the rstar_glm() function, based on Pierce and Bellio (2017) <doi:10.1111/insr.12232> and 'likelihoodAsy'. The package offers two main features: LA_rstar(), which launches an interactive 'shiny' application allowing users to fit models with rstar_glm() through their web browser, and sim_rstar_glm_pgsq1(), which streamlines the process of launching a web-based 'shiny' simulation application that saves results to a user-created 'PostgreSQL' database.

License MIT + file LICENSE
Encoding UTF-8

RoxygenNote 7.3.1

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

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

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports DT, ggplot2, likelihoodAsy, MASS, pool, RPostgres, shiny, shinythemes, sn

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2024-07-15 07:40:02 UTC
Contents

LA_rstar ................................................................. 2
rstar_glm .............................................................. 3
run_sim_rstar_glm .................................................... 5
sim_rstar_glm .......................................................... 7
sim_rstar_glm_pgsql ................................................... 8

Index 10

LA_rstar  Launch Shiny App for likelihoodAsy rstar Analysis

Description

This function launches a Shiny application that facilitates the setup and execution of likelihoodAsy rstar analysis. The app allows users to upload a dataset, specify a model and parameters of interest, and perform the analysis with the option to compute confidence intervals for r* statistics.

Usage

LA_rstar()

Value

A Shiny app object that can be run locally.

References


Examples

if (interactive()) {
    LA_rstar()
}
rstar_glm

Compute r* Statistics for Generalized Linear Models

Description

The rstar_glm function computes r* statistics for hypothesis testing on coefficients of interest in generalized linear models (GLMs). It supports logistic, linear, and Poisson regression models. For logistic models, the outcome must be binary.

Usage

rstar_glm(
  .formula,
  .data,
  .model = c("logistic", "linear", "poisson"),
  .psidesc = "Coefficient of Interest",
  .psival = 0,
  .fpsi = 2,
  .rstar.ci = FALSE,
  ...
)

## S3 method for class 'logistic'

rstar_glm(
  .formula,
  .data,
  .model = c("logistic", "linear", "poisson"),
  .psidesc = "Coefficient of Interest",
  .psival = 0,
  .fpsi = 2,
  .rstar.ci = FALSE,
  ...
)

## S3 method for class 'linear'

rstar_glm(
  .formula,
  .data,
  .model = c("logistic", "linear", "poisson"),
  .psidesc = "Coefficient of Interest",
  .psival = 0,
  .fpsi = 2,
  .rstar.ci = FALSE,
  ...
)

## S3 method for class 'poisson'

rstar_glm

rstar_glm
rstar_glm(.formula, .data, .model = c("logistic", "linear", "poisson"), .psidesc = "Coefficient of Interest", .psival = 0, .fpsi = 2, .rstar.ci = FALSE, ...)

## Default S3 method:
rstar_glm(.formula, .data, .model = c("logistic", "linear", "poisson"), .psidesc = "Coefficient of Interest", .psival = 0, .fpsi = 2, .rstar.ci = FALSE, ...)

Arguments

- `.formula` A formula specifying the model.
- `.data` A data frame containing the variables in the model.
- `.model` The type of GLM model: "logistic", "linear", or "poisson".
- `.psidesc` A description of the parameter of interest.
- `.psival` The value of the parameter of interest under the null hypothesis.
- `.fpsi` The index of the parameter of interest.
- `.rstar.ci` Logical; if TRUE, compute confidence intervals for r*.

... Additional arguments passed to the likelihoodAsy functions.

Value

A list with the object returned from likelihoodAsy::rstar (rs), the object returned from likelihoodAsy::rstar.ci (rs_ci), and the object returned from stats::glm (fit_glm).

References


Examples

```r
# Logistic model
rstar_glm(law ~ DriversKilled + VanKilled + drivers + kms,
    .data = Seatbelts,
    .model = "logistic") |> suppressWarnings()

# Poisson model
rstar_glm(count ~ spray,
    .data = InsectSprays,
    .model = "poisson") |> suppressWarnings()

# Linear model
rstar_glm(mpg ~ wt + hp,
    .data = mtcars,
    .model = "linear") |> suppressWarnings()
```

Description

This function runs multiple iterations of simulation for the sim_rstar_glm function and summarizes the results, including rejection rates, bias, empirical standard error, mean squared error, and root mean squared error.

Usage

```r
run_sim_rstar_glm(
    n_sims,
    alpha_level = 0.05,
    n_main,
    n_covariates,
    true_coef_main,
    n_control = NULL,
    true_coef_control = NULL,
    treatment_effect = NULL,
    model = c("logistic", "linear", "poisson"),
    skewness_main = NULL,
    skewness_control = NULL,
    Sigma_main = NULL,
    Sigma_control = NULL,
    ...
)
```
Arguments

- n_sims: Number of simulations to run.
- alpha_level: Significance level for hypothesis tests.
- n_main: Number of observations in the main group.
- n_covariates: Number of covariates.
- true_coef_main: True coefficients for the main group.
- n_control: Number of observations in the control group.
- true_coef_control: True coefficients for the control group.
- treatment_effect: Treatment effect size.
- model: Type of model: "logistic", "linear", or "poisson".
- skewness_main: Skewness for the main group covariates.
- skewness_control: Skewness for the control group covariates.
- Sigma_main: Covariance matrix for the main group covariates.
- Sigma_control: Covariance matrix for the control group covariates.
- ...: Additional arguments passed to sim_rstar_glm.

Value

A list with the results of each simulation and a summary of the results.

References


Examples

```r
sim_summary <- run_sim_rstar_glm(
  n_sims = 2, alpha_level = 0.05,
  n_main = 100, n_covariates = 2, true_coef_main = c(0.5, -0.3),
  n_control = 100, true_coef_control = c(0.2, -0.1),
  treatment_effect = 1, model = "linear"
) |> suppressWarnings()
```
sim_rstar_glm

Simulate Data and Fit GLM and r* Models

Description

This function generates simulated data for main and control groups, fits a generalized linear model (GLM) and an r* model, and returns the results.

Usage

sim_rstar_glm(
  n_main, n_covariates, true_coef_main, n_control = NULL, true_coef_control = NULL, treatment_effect = NULL, model = c("logistic", "linear", "poisson"), skewness_main = NULL, skewness_control = NULL, Sigma_main = NULL, Sigma_control = NULL, ...
)

Arguments

n_main Number of observations in the main group.
n_covariates Number of covariates.
true_coef_main True coefficients for the main group.
n_control Number of observations in the control group.
true_coef_control True coefficients for the control group.
treatment_effect Treatment effect size.
model Type of model: "logistic", "linear", or "poisson".
skewness_main Skewness for the main group covariates.
skewness_control Skewness for the control group covariates.
Sigma_main Covariance matrix for the main group covariates.
Sigma_control Covariance matrix for the control group covariates.
... Additional arguments passed to rstar_glm.
**Value**

A list with fitted GLM and r* models, and the simulated data.

**References**


**Examples**

```r
sim_result <- sim_rstar_glm(
  n_main = 100, n_covariates = 2, true_coef_main = c(0.5, -0.3),
  n_control = 100, true_coef_control = c(0.2, -0.1),
  treatment_effect = 0.5, model = "logistic"
) |> suppressWarnings()
```

**Description**

This function launches a Shiny application for setting up and running simulations based on the `rstar_glm` function. The app allows users to input parameters for the simulation, run the simulation, view results, and save results to a PostgreSQL database.

**Usage**

```r
sim_rstar_glm_pgs ql(dbname, datatable, host, port, user, password)
```

**Arguments**

- `dbname` The name of the PostgreSQL database.
- `datatable` The name of the table in the PostgreSQL database to save the results.
- `host` The host of the PostgreSQL database.
- `port` The port of the PostgreSQL database.
- `user` The username for accessing the PostgreSQL database.
- `password` The password for accessing the PostgreSQL database.

**Value**

A Shiny app object that can be run locally.
References


Examples

```r
if (interactive()) {
  sim_rstar_glm_pgsq1(
    dbname = "mydb",
    datatable = "simulation_results",
    host = "localhost",
    port = 5432,
    user = "myuser",
    password = "mypassword"
  )
}
```
Index

LA_rstar, 2
rstar_glm, 3
run_sim_rstar_glm, 5
sim_rstar_glm, 7
sim_rstar_glm_psql, 8