Package 'mmirestriktor'

July 17, 2024

Title Informative Hypothesis Testing Web Applications

Version 0.3.1

Description Offering enhanced statistical power compared to traditional
hypothesis testing methods, informative hypothesis testing allows researchers
to explicitly model their expectations regarding the relationships among
parameters. An important software tool for this framework is 'restriktor'.
The 'mmirestriktor' package provides 'shiny' web applications to implement
some of the basic functionality of 'restriktor'. The mmirestriktor() function
launches a 'shiny' application for fitting and analyzing models with
constraints. The FbarCards() function launches a card game application which
can help build intuition about informative hypothesis testing. The
iht_interpreter() helps interpret informative hypothesis testing results based
on guidelines in Vanbrabant and Rosseel (2020) <doi:10.4324/9780429273872-14>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

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

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

Imports DT, MASS, mmcards, pool, restriktor, RPostgres, 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 2024-07-17 16:00:02 UTC

Contents

---

deal_cards_to_grid   *Deal Cards to Grid*

Description

This function deals n^2 cards from a specified or default deck to form an n x n grid. The remaining deck is also returned alongside the grid.

Usage

```
deal_cards_to_grid(deck = mmcards::shuffle_deck(), n)
```

Arguments

deck   A dataframe representing a deck of cards, with each row being a a card. The
                parameter is designed to take mmcards::shuffle_deck() or mmcards::i_deck() as
                input.

n   A single integer representing the number of rows and columns in the grid (i.e.,
                the grid will be n x n). This parameter is required and does not have a default
                value.

Value

A list containing two elements:

- `cards_matrix`: an n x n matrix where each element is a list representing a card.
- `updated_deck`: a list representing the remaining deck after n^2 cards have been dealt.

Examples

```
# Dealing cards to a 2x2 grid using the default shuffled deck
deal_cards_to_grid(n = 2)
```

---

d_eq_spaced   *Calculate Equally Spaced Differences*

Description

This function calculates the equally spaced differences between means in a one-way ANOVA setup. It is based on the formula presented in Vanbrabant et al. (2015) for calculating differences between means, d.

Usage

```
d_eq_spaced(k, f)
```

Arguments

k   An integer representing the number of groups (k = 3, ..., 8).

f   A numeric value representing the effect size (f = 0.10, 0.15, 0.20, 0.25, 0.30, 0.40). Typical values represent small (0.10), medium (0.25), and large (0.40) effects.

Value

A numeric value representing the equally spaced difference, d, based on the number of groups (k) and the effect size (f).

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
d_eq_spaced(4, 0.25) # For k = 4 and f = 0.25
```

FbarCards   *FbarCards Shiny App*

Description

Launches a 'shiny' app for the FbarCards game. In this game, a grid of cards is displayed and the objective is to reorder the cards in each row such that, when the rows are stacked, the columns of cards are in increasing order from left to right. Players can swap the positions of two cards in the same row before finalizing their choices and scoring the game. The game utilizes Informative Hypothesis Testing (IHT) to score the final grid of cards.

Usage

```
FbarCards()
```

Value

This function launches a shiny app and does not return a value.

Examples

```
if (interactive()) {
  FbarCards()
}
```

---

generate_datasets   *Generate Datasets for ANOVA Simulation*

Description

This function generates a specified number of datasets for use in ANOVA simulations.Each dataset is generated based on a specified number of groups, effect size, and sample size per group. The data generation follows the model: $y_i = mu_1 x_{i1} + ... + mu_k x_{ik} + e_i$, as described in Vanbrabant, Van De Schoot, and Rosseel (2015).

Usage

```
generate_datasets(S, k, f, n)
```

Arguments

S	Integer, the number of datasets to generate.
k	Integer, the number of groups (k = 3, ..., 8).
f	Numeric, the effect size (f = 0.10, 0.15, 0.20, 0.25, 0.30, 0.40).
n	Integer, the sample size per group.

Value

A list of data frames, each representing a dataset. Each data frame contains two columns: 'x', indicating group membership, and 'y', representing the dependent variable generated according to the model.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
generate_datasets(S = 2, k = 4, f = 0.25, n = 30)
```

---

generate_datasets_reg   *Generate Multiple Datasets for Regression Simulation*

Description

This function generates a specified number of datasets for regression analysis simulations. Each dataset is generated using the sim_reg function, based on given parameters like sample size, number of predictors, effect size, and correlation coefficient.

Usage

```
generate_datasets_reg(S = 20000, n, p, f2, rho, beta = 0.1)
```

Arguments

S	The number of datasets to generate, default is 20000.
n	The number of observations in each dataset.
p	The number of predictors in the regression model for each dataset.
f2	The effect size for each dataset, defined as ($f^2 = R^2 / (1 - R^2)$).
rho	The correlation coefficient between predictors in each dataset.
beta	The regression coefficients for the predictors in each dataset, either as a single value or a vector of length (p).

Details

The function uses sim_reg to simulate individual datasets, which are then combined into a list. Each dataset is a data frame with named columns for the response variable and predictors.

Value

A list of data frames, each representing a simulated dataset for regression analysis. Each data frame contains columns for the response variable 'y' and predictors 'x1', 'x2', ..., 'xp'.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
datasets <- generate_datasets_reg(S = 2, n = 50, p = 3, f2 = 0.10, rho = 0.5)
```

---

iht_interpreter   *Interpret Results of Informative Hypothesis Test*

Description

This function provides a human-readable explanation of the results of an informative hypothesis test. It interprets the p-values of both Type A and Type B tests and provides an overall conclusion.

Usage

```
iht_interpreter(iht_res, alpha = 0.05)
```

Arguments

iht_res	A 'conTest' object containing the results of an informative hypothesis test.
alpha	The significance level for interpreting the p-values. Default is 0.05.

Value

A character string providing a detailed interpretation of the hypothesis test results.

References

Vanbrabant, L., & Rosseel, Y. (2020). An Introduction to Restriktor: Evaluating informative hypotheses for linear models. In R. van de Schoot & M. Miocevic (Eds.), Small Sample Size Solutions: A Guide for Applied Researchers and Practitioners (1st ed., pp. 157 -172). Routledge.

Examples

```
model <- mmir_model(mpg ~ -1 + hp + wt, data = mtcars, engine = "lm",
                    standardize = TRUE)
iht_res <- restriktor::iht(model, constraints = 'hp < wt')
iht_interpreter(iht_res) |> cat()
```

| `lr_pow` | *Calculate Power for Linear Regression Simulations* |

Description

This function computes the power of hypothesis tests in a linear regression setting, considering constraints on the regression coefficients. It processes a list of data frames, each representing a different dataset, and calculates the power based on specified constraints.

Usage

```
lr_pow(df_list, constr = 0, standardize = TRUE, alpha = 0.05)
```

Arguments

`df_list`	A list of data frames, each representing a dataset for regression analysis. Each data frame should contain the response variable 'y' and the predictor variables 'x1', 'x2', ..., 'xp'.
`constr`	The number of inequality constraints imposed on the regression coefficients. It must be a non-negative integer less than or equal to the number of predictors (p). A value of 0 implies no constraints or equality constraints.
`standardize`	A logical value indicating whether the predictor variables should be standardized before fitting the model. Default is TRUE.
`alpha`	The significance level used in hypothesis testing, default is 0.05.

Details

The function validates the 'constr' parameter, optionally standardizes the predictor variables, constructs the necessary constraints, and calculates power by fitting a linear model to each dataset. It uses the 'iht' function from the 'restriktor' package to apply the constraints and evaluate the hypothesis tests.

Value

A numeric value representing the calculated power, defined as the proportion of datasets meeting the hypothesis test criteria as defined by the constraints and significance level.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
generate_datasets_reg(S = 4, n = 30, p = 3, f2 = 0.20, rho = 0.5) |> lr_pow()
```

---

mmirestriktor   *Mighty Metrika Interface to Restriktor Shiny App*

Description

This function launches a Shiny app which allows users to fit and analyze models with restrictions using the mmir_model(), restriktor::iht(), and restriktor::restriktor() functions. The app provides a user interface to upload a CSV file, specify a model formula, and define constraints for informative hypothesis testing.

Usage

```
mmirestriktor()
```

Details

The app has the following functionalities:

- Upload a CSV file to be used as the dataset for modeling.
- View the variables available in the uploaded dataset.
- Input a formula to define the model to be fit.
- Choose a model fitting engine from "lm", "glm", and "rlm".
- Specify extra arguments for the model fitting function.
- View the terms available for defining constraints after fitting the model.
- Define constraints for hypothesis testing.
- Set a significance level (alpha) for hypothesis testing.
- Choose the type of analysis to perform: Informative Hypothesis Test and/or Restricted Means.
- View the results and interpretation of the hypothesis tests after running the analysis.

Value

This function does not return a value; it launches a Shiny app in the user's default web browser.

Examples

```
if (interactive()){
  mmirestriktor()
}
```

---

mmir_model   *Fit Restriktor Supported Model*

Description

mmir_model() is a function for the 'mmirestriktor' package that fits a model to data using one of the specified engines ('lm', 'glm', or 'rlm'). It also provides an option to standardize numeric variables.

Usage

```
mmir_model(formula, data, engine = "lm", standardize = TRUE, ...)
```

Arguments

formula	An object of class 'formula' (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	A data frame containing the variables in the model.
engine	A character string indicating which engine to use for model fitting. Can be one of 'lm', 'glm', or 'rlm'. Default is 'lm'.
standardize	Logical. If TRUE, numeric predictor variables in 'data' are standardized before fitting the model. Default is TRUE.
...	Additional arguments to be passed to the model fitting function (lm, glm, or rlm).

Details

The mmir_model() function serves as a utility function for fitting models in the 'mmirestriktor' package. It supports different modeling engines and allows for variable standardization.

Value

An object representing the fitted model, of class 'lm', 'glm', or 'rlm' depending on the engine used.

See Also

[lm](#), [glm](#), [rlm](#)

Examples

```
mod <- mmir_model(mpg ~ hp + wt, data = mtcars, engine = "lm")
summary(mod)
```

---

mui *Calculate Group Means for One-Way ANOVA*

Description

This function calculates the means of different groups in a one-way ANOVA setting. It uses the equally spaced difference calculated by 'd_eq_spaced' and follows the approach described in Vanbrabant et al. (2015).

Usage

```
mui(k, f)
```

Arguments

k   An integer representing the number of groups (k = 3, ..., 8).

f   A numeric value representing the effect size (f = 0.10, 0.15, 0.20, 0.25, 0.30, 0.40).Typical values represent small (0.10), medium (0.25), and large (0.40) effects.

Value

A numeric vector containing the means of the groups. Each element of the vector corresponds to a group mean.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
mui(4, 0.25) # For k = 4 and f = 0.25
```

---

pj_pow *Power Calculation for ANOVA Simulation*

Description

This function calculates the power for hypothesis tests in a constrained statistical inference setting, particularly in the context of ANOVA and regression as discussed in Vanbrabant et al. (2015). It is designed to work with a list of data frames, where each data frame represents a different dataset. The function accommodates both equality and inequality constraints.

Usage

```
pj_pow(df_list, constr = 0, alpha = 0.05)
```

Arguments

df_list   A list of data frames, each representing a dataset. Designed to use results generated from the generate_datasets() function.

constr   An integer indicating the number of inequality constraints. A value of 0 indicates that all constraints are equality constraints. The value must be a non-negative integer less than the number of groups.

alpha   The significance level used in the hypothesis testing, with a default value of 0.05. It should be a numeric value between 0 and 1.

Details

The function first checks the validity of the 'constr' parameter and then constructs the constraint string based on the number of constraints. It runs the model for each dataset in the df_list using the mmir_model function and applies the constraints using the restriktor::iht function. The power is calculated based on the proportion of datasets that meet the hypothesis test criteria defined by the constraints and the significance level.

Value

The function returns the calculated power as a numeric value, representing the proportion of p-values smaller than the predefined significance level alpha.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
generate_datasets(S = 2, k = 4, f = 0.25, n = 30) |> pj_pow(constr=1)
```

---

replext   *Launch Replext Simulation Shiny Application*

Description

This function creates and launches a Shiny web application for running simulations related to constrained statistical inference in ANOVA and regression settings. The application allows users to set various parameters for `replext_t1_c1` and `replext_t2_c1` functions and view the resulting simulation data. The simulation is based on Vanbrabant et al. (2015).

Usage

```
replext()
```

Details

The Shiny application consists of a user interface for setting simulation parameters and a server logic to process the simulations. Users can select between different simulation settings (cell blocks), specify parameters, run the simulations, view the results in a table format, and download the results. The application also handles dynamic UI elements based on user selections and manages data downloads.

The app's UI includes:

- A sidebar for input parameters and action buttons.
- A main panel for displaying simulation results.

The server logic includes:

- Rendering parameter input UI based on selected cell block.
- Running simulations and storing results.
- Rendering and exporting the results table.

Value

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

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
# Launch the Replext Simulation Shiny application
if(interactive()){
  replext()
}
```

---

replext_pgsql   *Launch Replext Simulation Shiny Application*

Description

This function creates and launches a Shiny web application for running simulations related to constrained statistical inference in ANOVA and regression settings. The application allows users to set various parameters for replext_t1_c1 and replext_t2_c1 functions and view the resulting simulation data. The simulation is based on Vanbrabant et al. (2015). The app includes functionality to interact with a PostgreSQL database. The app includes a user interface for selecting simulation parameters and a server logic to process the simulation and handle user interactions, including saving and retrieving data from a database.

Usage

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

Arguments

dbname	The name of the PostgreSQL database to connect to.
datatable	The name of the table in the database where the simulation results will be stored and retrieved.
host	The host address of the PostgreSQL database.
port	The port number for the PostgreSQL database connection.
user	The username for accessing the PostgreSQL database.
password	The password for the specified user to access the PostgreSQL database.

Details

The Shiny application consists of a user interface for setting simulation parameters and a server logic to process the simulations and save to PostgreSQL database. Users can select between different simulation settings (cell blocks), specify parameters, run the simulations, view the results in a table format, submit results to PostgreSQL database, and download the database table. The application also handles dynamic UI elements based on user selections and manages data downloads.

Value

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

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
if (interactive()) {
  replext_pgsql(
    dbname = "your_db_name",
    datatable = "your_data_table",
    host = "localhost",
    port = 5432,
    user = "your_username",
    password = "your_password"
  )
}
```

---

replext_t1_c1   *Replext Function for ANOVA Simulations in Table 1 Cell 1*

Description

This function performs repeated simulations for ANOVA to determine minimum sample sizes for given power and effect sizes, as well as calculating Type I error rates. It is designed to replicate and extend the results for Table 1 Cell 1 in Vanbrabant et al. (2015).

Usage

```
replext_t1_c1(
  S = 20000,
  k = 3,
  fs = c(0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4),
  n_start = 6,
  constrs = c(0, 1, 2),
  alpha = 0.05,
  pow = 0.8,
  nmax = 1000
)
```

Arguments

S	The number of datasets to generate for each simulation, default is 20000.
k	The number of groups in the ANOVA design.
fs	A vector of effect sizes to consider in the simulations.
n_start	The starting sample size for the simulations.
constrs	A vector of constraint types to be used in the simulations.
alpha	The significance level used in hypothesis testing, default is 0.05.
pow	The desired power for the statistical test, default is 0.80.
nmax	The maximum sample size to consider in the simulations.

Details

The function uses a nested approach, first determining minimum sample sizes for various combinations of effect size and constraints, and then calculating Type I error rates. It leverages the 'pj_pow' function for power calculation and integrates internal function 'find_min_sample_size' for determining the smallest sample size achieving the desired power.

Value

A data frame containing the calculated Type I error rates and the minimum sample sizes required for each combination of effect size and constraint type.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
replext_t1_c1(S=5, fs = c(0.40), constrs = c(2))
```

---

replext_t2_c1   *Generate Replext Tables for Linear Regression Analysis*

Description

This function generates replext tables for linear regression, similar to those in Table 2 Cell 1 of the referenced paper. It computes minimum sample sizes for various power and effect size combinations, and calculates Type I error rates.

Usage

```
replext_t2_c1(
  S = 20000,
  p = 3,
  f2s = c(0.02, 0.05, 0.08, 0.1, 0.15, 0.2, 0.25, 0.35),
  n_start = 6,
  constrs = c(0, 1, 2, 3),
  rho = 0,
  beta = 0.1,
  alpha = 0.05,
  pow = 0.8,
  standardize = TRUE,
  nmax = 1000
)
```

Arguments

S	The number of datasets to generate for each simulation, default is 20000.
p	The number of predictors in the regression model.
f2s	A vector of effect sizes to be used in the simulations.
n_start	The starting sample size for the simulations.
constrs	A vector of constraint types (number of inequality constraints) to be applied in the simulations.
rho	The correlation coefficient between predictors, default is 0.0.
beta	The regression coefficient for predictors, default is 0.1.
alpha	The significance level used in hypothesis testing, default is 0.05.
pow	The desired power for the statistical test, default is 0.80.
standardize	A logical flag to indicate whether to standardize the predictors in the datasets, default is TRUE.
nmax	The maximum sample size to consider in the simulations.

Details

The function uses a nested approach to first determine minimum sample sizes for different combinations of effect size and constraints, and then calculates Type I error rates. It leverages the `lr_pow` function for power calculation and uses `generate_datasets_reg` for dataset generation.

Value

A data frame containing Type I error rates and the minimum sample sizes required for each combination of effect size and constraint type.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
replext_t2_c1(S = 2, f2s = c(0.35), constrs = c(2))
```

---

rm_interpreter   *Interpret Results of Restricted Means Analysis*

Description

This function provides a human-readable interpretation of the results of a restricted means analysis. It compares the original (unconstrained) and reduced (restricted) R-squared values to evaluate the imposed constraints. It also returns the Generalized Order-Restricted Information Criterion (GORIC) which can be used for model comparison.

Usage

```
rm_interpreter(rm_res)
```

Arguments

rm_res   An object of class 'restriktor', typically the result of a call to restriktor.

Value

A character string with an interpretation of the analysis results, including the R-squared values, their reduction, and the Generalized Order-Restricted Information Criterion (GORIC) if available.

References

Vanbrabant, L., & Rosseel, Y. (2020). An Introduction to Restriktor: Evaluating informative hypotheses for linear models. In Small Sample Size Solutions (1st ed., p. 16). Routledge.

See Also

restriktor for generating 'restriktor' objects.

Examples

```
model <- mmir_model(mpg ~ -1 + hp + wt, data = mtcars, engine = "lm",
                    standardize = TRUE)
rm_res <- restriktor::restriktor(model, constraints = 'hp < wt')
rm_interpreter(rm_res) |> cat()
```

sim_reg	*Regression Data Simulation for Linear Models*

Description

This function simulates data for linear regression analysis, as described in the supplemental material of the referenced paper. It generates datasets with a specified number of predictors and sample size, effect size, and correlation coefficient, considering a linear model with fixed regression coefficients.

Usage

```
sim_reg(n, p, f2, rho, beta = 0.1)
```

Arguments

n	The total number of observations to generate.
p	The number of predictors (Beta) in the regression model.
f2	The effect size, calculated as ($f^2 = R^2 / (1 - R^2)$), where (R^2) is the coefficient of determination.
rho	The correlation coefficient between predictors, representing the off-diagonal elements in the covariance matrix. Should be a numeric value.
beta	The regression coefficients, either a single value replicated for each predictor or a vector of length equal to the number of predictors (p).

Details

The function validates the length of the beta vector, constructs a covariance matrix for the predictors, and calculates the variance of the error term. It then uses the multivariate normal distribution to generate predictor values and calculates the response variable based on the specified regression coefficients and effect size.

Value

A list containing two elements: 'y', the simulated response variable, and 'X', the matrix of predictors.

References

Vanbrabant, Leonard; Van De Schoot, Rens; Rosseel, Yves (2015). Constrained statistical inference: sample-size tables for ANOVA and regression. Frontiers in Psychology, 5. DOI:10.3389/fpsyg.2014.01565. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01565

Examples

```
# Example usage:
# Simulate data for a regression model with 100 observations, 3 predictors,
# an effect size of 0.10, and a correlation coefficient of 0.5
sim_reg(n = 100, p = 3, f2 = 0.10, rho = 0.5)
```

Index