Package 'RGremlinsConjoint'

January 20, 2025

Type Package

Title Estimate the ``Gremlins in the Data" Model for Conjoint Studies

Version 0.9.1

Description The tools and utilities to estimate the model described in ``Gremlin's in the Data: Identifying the Information Content of Research Subjects" (Howell et al. (2021) <doi:10.1177/0022243720965930>) using conjoint analysis data such as that collected in Sawtooth Software's 'Lighthouse' or 'Discover' products. Additional utilities are included for formatting the input data.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat

RoxygenNote 7.2.3

Depends R (>= 2.10)

Imports bayesm

VignetteBuilder knitr

URL https://github.com/statuser/RGremlinsConjoint

BugReports https://github.com/statuser/RGremlinsConjoint/issues

NeedsCompilation no

Author John Howell [aut, cre, cph],
Peter Ebbes [aut],
John Leichty [aut]

Maintainer John Howell <john_howell@byu.edu>

Repository CRAN

Date/Publication 2023-09-08 21:40:02 UTC

Contents

`cbc.df` *Simulated data for the "Gremlins in the Data Model"*

Description

A dataset containing simulated choices from a CBC study where some of the respondents are information poor or 'Gremlins'. The data is simulated data and does not reflect actual preferences.

Usage

```
cbc.df
```

Format

A data frame with 32000 rows and 10 variables:

resp.id A respondent identifier

ques The question or task number

alt The choice alternative

choice An indicator that takes on a value of 1 if the alternative was chosen. (Default is 0.)

brandFord A dummy coded variable indicating the brand is Ford

brandGM A dummy coded variable indicating the brand is GM

brandDodge A dummy coded variable indicating the brand is Dodge

enghyb A dummy coded variable indicating the engine is a hybrid

engelec A dummy coded variable indicating the engine is electric

price A continuous variable for the relative price of the individual offerings.

code_sawtooth_design *Convert a Sawtooth Software generated design file to a dummy coded design file*

Description

Take a design file such as those generated by the Sawtooth Software 'Lighthouse Studio' and convert it into a dummy coded design file. The last level in the attribute is considered the reference level and will be dropped.

Usage

```
code_sawtooth_design(
  sawtooth_design,
  columns_to_code = c(4:ncol(sawtooth_design)),
  include_none_option = FALSE
)
```

Arguments

`sawtooth_design`
: A matrix that contains the Sawtooth design. Can be loaded with read.csv.

`columns_to_code`
: (Optional, Default = all columns) A vector listing the numeric index of the columns to code. Note: The first column is column 4 due to the control variables

`include_none_option`
: (Optional, Default = FALSE) A boolean value indicating whether to expand the task to include a none option

Details

This function is written to not require converting columns to be factors. All variables should be numeric indexes for the levels of the attributes. If you would like to manually code a attribute of the design, for example if you have a price variable, you must manually code that attribute and then can call the function with the optional columns to code parameter.

Value

A matrix object chat contains the dummy coded design file. The last attribute is considered the reference level

See Also

/urlhttp://www.sawtoothsoftware.com/help/lighthouse-studio/manual/index.html?hid_web_cbc_exporting.html

Documentation for the Sawtooth Software Design file format can be found at

Examples

```
## Not run:
 # Read in the Sawtooth Formatted data
 design <- read.csv("Design.csv")
 prices = c(0.79, 1.29, 1.79, 2.29, 2.79)
 design$price <- prices[design$price]
 codedDesign <- codeSawtoothDesignFile(design, c(4:9))

## End(Not run)
```

`convert_to_bayesm` *Convert 'RGremlinsConjoint' formatted Data to 'bayesm' format*

Description

Converts a data file and a coded design file from the format expected by the package to a format appropriate for estimation in 'bayesm' `rhierMnlRwMixture`

Usage

```
convert_to_bayesm(data, design)
```

Arguments

`data`	The data.frame or matrix that contains the respondents answers
`design`	The data.frame or matrix that contains the coded design

Value

lgtdata The list data structure for use with 'bayesm'

See Also

`code_sawtooth_design`

Examples

```
## Not run:
 data <- read.csv("data.csv")
 design <- read.csv("design.csv")
 design <- code_sawtooth_design(design)
 convert_to_bayesm(data, design)

## End(Not run)
```

`estimateGremlinsModel` *Estimate Gremlin's Model - Hierarchical MNL*

Description

The function estimates the model described in "Gremlin's in the Data: Identifying the Information Content of Research Subjects" (Howell et al. (2021) <doi:10.1177/0022243720965930>) using a hierarchical multinomial logit model

Usage

```
estimateGremlinsModel(
  data,
  design,
  Priors = NULL,
  R = NULL,
  keepEvery = 1,
  verbose = TRUE,
  num_lambda_segments = 2,
  constraints = NULL,
  startingValues = NULL,
  previous_iterations = 0,
  Atchade_slope_tuning = 0.1,
  Atchade_lambda_tuning = 10
)
```

Arguments

`data`	A matrix containing the raw data. The first column the respondent identifier, followed by the design number, the remaining columns indicate the choices for the tasks that coincide to the design file.
`design`	A matrix representing the coded (dummy of effects) design file. The design file should be formatted as a matrix with number of versions X number of tasks X number of alternatives rows and number of parameters + 3 columns. The first column contains the version number, the second columns contains the task number, the third column contains the alternative, and the remaining columns contain the coded design. A generic Sawtooth Software design file can be converted to this format using the `code_sawtooth_design` function.
`Priors`	A data structure that contains the priors for to the model. Can be null indicating the use of default priors or must contain a full prior specification.
`R`	The number of repetitions in the chain
`keepEvery`	saves every keepEvery-th draw for output
`verbose`	Print intermediate results to the screen (default = TRUE)
`num_lambda_segments`	(Default = 2) The number of segments for the scale factor

`constraints` (Optional) a vector of length n-param specifying the constraints to impose on the parameters or NULL. a 1 indicates the parameter is constrained to be positive a -1 constrains to be negative, and a 0 indicates no constraint.

`startingValues` (Optional) starting values to use for the MCMC algorithm. This is a list of containing: slope = a nRespondent by nParamter matrix of slopes for the respondent slopeBar = a nParameter vector of the slopeBar parameter slopeCov = a nParameter by nParameter matrix containing the variance covariance matrix for the slopeBar parameter lambda = a nSegment vector containing the starting values for the lambda parameters. The first element in the vector should be 1. segMembership = a nRespondent vector containing the segment membership for each respondent. phi_lambda = a nParameter vector containing the base probabilities that an individual belongs to each segment. Should sum to 1.

`previous_iterations` The number of previous iterations run. This parameter is used for the Atchade adaptive MCMC step size algorithm. This is used since the Atchade update does not happen for less than 1000 iterations. (Default = 0)

`Atchade_slope_tuning` The tuning factor to use for Atchade step for the slopes parameter. Larger values decrease the acceptance rate. (Default = 0.01)

`Atchade_lambda_tuning` The tuning factor to use for the Atchade step for the lambda parameter. Larger values decrease the acceptance rate. (Default = 10)

Value

A data structure containing the draws from the complete MCMC chain

See Also

code_sawtooth_design

Examples

```
truck_design_file <- system.file("extdata", "simTruckDesign.csv", package = "RGremlinsConjoint")
truck_data_file <- system.file("extdata", "simTruckData.csv", package = "RGremlinsConjoint")

truckDesign <- read.csv(truck_design_file)
truckData <- read.csv(truck_data_file)
outputSimData_burn <- estimateGremlinsModel(truckData,
                                            truckDesign,
                                            R = 10,
                                            keepEvery = 1,
                                            num_lambda_segments = 2)
```

`gremlins` *'RGremlinsConjoint': A package for estimating the "Gremlins in the Data" model*

Description

The tools and utilities to estimate the model described in "Gremlin's in the Data: Identifying the Information Content of Research Subjects" (Howell et al. (2021) <doi:10.1177/0022243720965930>) using conjoint analysis data such as that collected in Sawtooth Software's 'Lighthouse' or 'Discover' products. Additional utilities are included for formatting the input data.

`gremlinsEnv` *Set global options for the gremlins models. These options are not expected to be modified by the user but are extracted from the functions to simplify the coding.*

Description

Set global options for the gremlins models. These options are not expected to be modified by the user but are extracted from the functions to simplify the coding.

Usage

```
gremlinsEnv
```

Format

An object of class `environment` of length 4.

Index