Package 'stratEst'

January 25, 2019

Type Package

Title Strategy Estimation

Version 0.1.2

Author Fabian Dvorak

Maintainer Fabian Dvorak <fabian.dvorak@uni.kn>

Date 2019-01-25

Description Variants of the strategy frequency estimation method by Dal Bo & Frechette (2011) <doi:10.1257/aer.101.1.411>, including the adaptation to estimate choice parameters of behavior strategies by Breitmoser (2015) <doi:10.1257/aer.20130675>, and the extension in the spirit of latent-class regression by Dvorak & Fehrler (2018) <doi:10.2139/ssrn.2986445>.

Copyright See the file COPYRIGHTS for copyright, authorship and license details

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp (>= 0.12.18)

RoxygenNote 6.1.1

Suggests testthat

Depends R (>= 2.10)

URL http://github.com/fdvorak/stratEst

BugReports http://github.com/fdvorak/stratEst/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-25 17:20:03 UTC

R topics documented:

ALLC *ALLC*

Description

Matrix representation of the prisoner's dilemma strategy which always plays C

Usage

```
data(ALLC)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(ALLC,ALLD,TFT,GRIM,PTFT)
```

ALLD *ALLD*

Description

Matrix representation of the prisoner's dilemma strategy which always plays D.

Usage

```
data(ALLD)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(ALLD,ALLC,TFT,GRIM,PTFT)
```

DC *DC*

Description

Matrix representation of the prisoner's dilemma strategy which starts with D, then alternates between C and D.

Usage

```
data(DC)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(DC,ALLD,ALLC,TFT,GRIM)
```

DF2011 *Data of Dal Bo and Frechette (2011)*

Description

A dataset with observations from the repeated prisoner's dilemma experiment of Dal Bo and Frechette (2011).

Usage

```
data(DF2011)
```

Format

A data frame with 7358 rows and 6 variables:

treatment A treatment identifier of the experiment.

id Variable which identifies a participant.

supergame The supergame number.

period The period of the supergame.

cooperation A dummy variable which is one if the participant cooperated in the current period.

other_cooperation A dummy variable which is one if the partner in the current match cooperated in the current period.

Source

https://www.aeaweb.org/articles?id=10.1257/aer.101.1.411

References

Dal Bo, P. and G. R. Frechette (2011): The evolution of cooperation in infinitely repeated games: Experimental evidence, *American Economic Review*, 101, 411-429.

DF2011LCR *Data of Dal Bo and Frechette (2011)*

Description

A dataset to set up to perform latent class regression on the repeated prisoner's dilemma experiment by Dal Bo and Frechette.

Usage

```
data(DF2011LCR)
```

Format

A data frame with 37042 rows and 8 variables:

supergame The supergame number.

period Period of the supergame.

coop Dummy which is one if the participant cooperated in the current round.

date Date of the session.

r The stage game parameter of treatment.

delta Discount factor of the treatment.

group Group id of two matched participants.

id Variable which identifies a unique participant-supergame combination.

Source

https://www.aeaweb.org/articles?id=10.1257/aer.101.1.411

References

Dal Bo, P. and G. R. Frechette (2011): The evolution of cooperation in infinitely repeated games: Experimental evidence, *American Economic Review*, 101, 411-429.

DGRIM2 *DGRIM2*

Description

Matrix representation of the prisoner's dilemma strategy which plays D in the first round, then play GRIM2.

Usage

```
data(DGRIM2)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(DGRIM2,GRIM,ALLD,ALLC,TFT)
```

DGRIM3 *DGRIM3*

Description

Matrix representation of the prisoner's dilemma strategy which plays D in the first round, then play GRIM3.

Usage

```
data(DGRIM3)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(DGRIM3,GRIM,ALLD,ALLC,TFT)
```

DTF2T *DTF2T*

Description

Matrix representation of the prisoner's dilemma strategy which plays D in the first round, then play TF2T.

Usage

```
data(DTF2T)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(DTF2T,GRIM,ALLD,ALLC,TFT)
```

DTF3T *DTF3T*

Description

Matrix representation of the prisoner's dilemma strategy which plays D in the first round, then play TF3T.

Usage

```
data(DTF3T)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(DTF3T,GRIM,ALLD,ALLC,TFT)
```

DTFT *DTFT*

Description

Matrix representation of the prisoner's dilemma strategy which plays D in the first round, then play TFT.

Usage

```
data(DTFT)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(DTFT,GRIM,ALLD,ALLC,TFT)
```

FC *FC*

Description

Matrix representation of the prisoner's dilemma strategy which plays C in the first round, then D forever.

Usage

```
data(FC)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(FC,ALLD,ALLC,TFT,GRIM)
```

GRIM *GRIM*

Description

Matrix representation of the prisoner's dilemma strategy which plays C until either player plays D, then it plays D forever.

Usage

```
data(GRIM)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(GRIM,ALLD,ALLC,TFT,PTFT)
```

GRIM2 *GRIM2*

Description

Matrix representation of the prisoner's dilemma strategy which plays C until 2 consecutive rounds occur in which either player played D, then play D forever.

Usage

```
data(GRIM2)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(GRIM2,GRIM,ALLD,ALLC,TFT)
```

GRIM3 *GRIM3*

Description

Matrix representation of the prisoner's dilemma strategy which plays C until 3 consecutive rounds occur in which either player played D, then play D forever.

Usage

```
data(GRIM3)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(GRIM3,GRIM,ALLD,ALLC,TFT)
```

M1BF *M1BF*

Description

Matrix representation of the prisoner's dilemma strategy which plays if both players played C, and D if both players played D. If the own action was C and the other player played D, play C with some probability. If the own action was D and the other player played C, play C with some (potentially different) probability.

Usage

```
data(M1BF)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(M1BF,GRIM,ALLD,ALLC,TFT)
```

PT2FT *PT2FT*

Description

Matrix representation of the prisoner's dilemma strategy which plays C if both players played C in the last 2 rounds, both players played D in the last 2 rounds, or both players played D 2 rounds ago and C last round. Otherwise play D.

Usage

```
data(PT2FT)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(PT2FT,GRIM,ALLD,ALLC,TFT)
```

PTFT *PTFT*

Description

Matrix representation of the prisoner's dilemma strategy which plays C if both players chose the same move last round, otherwise it plays D.

Usage

```
data(PTFT)
```

Format

An object of class `matrix` with 2 rows and 6 columns.

Details

#' @format A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(PTFT,GRIM,ALLD,ALLC,TFT)
```

RAND *RAND*

Description

Matrix representation of the prisoner's dilemma strategy which always randomizes between C and D.

Usage

```
data(RAND)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(RAND,GRIM,ALLD,ALLC,TFT)
```

SGRIM *SGRIM*

Description

Matrix representation of the semi grim strategy (Breitmoser 2015). The strategy plays C if both players played C, and D if both players played D. If one player played D and the other C, play C with some probability.

Usage

```
data(SGRIM)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

References

Breitmoser, Y. (2015): Cooperation, but no reciprocity: Individual strategies in the repeated prisoner's dilemma, *American Economic Review*, 105, 2882-2910.

Examples

```
strategies <- rbind(SGRIM,GRIM,ALLD,ALLC,TFT)
```

stratEst *Estimation function for strategy estimation*

Description

Performs variants of the strategy estimation method.

Usage

```
stratEst(data, strategies, shares, covariates, cluster,
  response = "mixed", r.responses = "no", r.trembles = "global",
  select = "no", min.strategies = 1, crit = "bic", se = "yes",
  outer.runs = 10, outer.tol = 0, outer.max = 1000,
  inner.runs = 100, inner.tol = 0, inner.max = 10, lcr.runs = 1000,
  lcr.tol = 0, lcr.max = 1000, bs.samples = 1000,
  print.messages = TRUE)
```

Arguments

`data`	Mandatory input object which contains the data for the estimation in the long format. Each row in `data` represents one observation of one individual. The object `data` must be a data frame object with variables in columns. Three columns are mandatory: A column named `id` which identifies the observations of the same individual across the rows of the data frame. A column named `input` which indicates the type of information observed by the individual before giving a response. A column named `output` which contains the behavioral response of the individual after observing the input. If an individual plays the same game for more than one period with the same partner, `data` must contain a variable `period` which identifies the period within the game. If an individual plays the same game more than once with different partners, `data` must contain a variable `game` (or `supergame`) which identifies data from different games. For data from prisoner's dilemma experiments, two more data formats are possible. Instead of using the variables `input` and `output`, the data frame may also contain the variables `cooperation` and `other_cooperation`, or alternatively, the variables `cooperation` and `group`. The variable `cooperation` should be a dummy which indicates if the participant cooperated in the current period. The variable `other_cooperation` should be a dummy which indicates if the other player cooperated in the current period. The variable `group` should be an identifier variable with a unique value for each unique match of two individuals.
`strategies`	Mandatory input object. Can be either a positive integer or a matrix. If an integer is used, the estimation function will generate the respective number of memory-one strategies with as many states as there are unique input values in `data`. A matrix can be used to supply a set of customized strategies. In the matrix, each row corresponds to one state of a strategy, starting with the start state of an automaton. The first column enumerates the states of each strategy in ascending order. A value of one in the first column indicates the begin of a new strategy with its start state. The columns after the first column contain the collection of multinomial response vectors. The number of columns for the multinomial response vectors must correspond to the number of unique non-zero outputs in data. Without a reference output - which is labeled with a zero in the output column of data - the columns specify the complete multinomial response distribution for each unique value in the output column. In this case, the response probabilities in each row must sum to one. With a reference output, the response probability for the response labeled with zero is omitted and the response probabilities in each row must sum to a value smaller or equal to one. The remaining columns of the strategies matrix define the deterministic state

transitions. The number of columns must equal the number of unique non-zero inputs in the data. The numbers in the first column indicate the next state of the automaton if the input is one. The numbers in the second column indicate the next state if the input is two and so on.

`shares` A column vector of strategy shares. The number of elements must correspond to the number of strategies defined in the strategies matrix. Elements which are NA are estimated from the data. If the object is not supplied, a share is estimated for every strategy defined in the strategies matrix.

`covariates` A matrix where each row corresponds to same row in data. Hence, the covariate matrix must have as many rows as the data matrix. Observations which have the same ID in data must also have the same vector of covariates. Missing value are not allowed. If covariates are supplied, a latent class regression model is estimated.

`cluster` A column vector which indicates the assignment of each row in data to cluster units for block-bootstrapped standard errors. Note that estimates will nevertheless be biased due to the non-linearity of the model.

`response` String which can be set to `"pure"` or `"mixed"`. If set to `"pure"` all response probabilities estimated from the data are pure responses. If set to `"mixed"` all response probabilities estimated from the data are mixed responses. The default is `"mixed"`.

`r.responses` A string which can be set to `"no"`, `"strategies"`, `"states"` or `"global"`. If set to `"strategies"`, the estimation function estimates strategies with one strategy specific vector of responses in every state of the strategy. If set to `"states"`, one state specific vector of responses is estimated for each state. If set to `"global"`, a single vector of responses is estimated which applies in every state of each strategy. Default is `"no"`.

`r.trembles` String which can be set to `"no"`, `"strategies"`, `"states"` or `"global"`. If set to `"strategies"`, the estimation unction estimates strategies with one strategy specific tremble probability. If set to `"states"`, one state specific tremble probability is estimated for each state. If set to `"global"`, a single tremble is estimated which applies in every state of each strategy. Default is `"global"`.

`select` String which can be set to `"no"`, `"strategies"`, `"responses"`, `"trembles"`, `"both"`, and `"all"`. If set to `"strategies"`, `"responses"`, `"trembles"`, the number of strategies, responses, trembles respectively are selected based on the selection criterion specified in option `"crit"`. If set to `"both"`, the number of responses and trembles are selected. If set to `"all"`, the number of strategies, responses, and trembles are selected. Note that the selection of responses and trembles occurs within the scope of the restriction set to these parameters (E.g. if `r.responses` is set to `"strategies"`, `select = "responses"` will select responses within each strategy). Default is `"no"`.

`min.strategies` Integer which specifies the minimum number of strategies for strategy selection. The selection procedure stops if the minimum is reached.

`crit` String which can be set to `"bic"`, `"aic"` or `"icl"`. If set to `"bic"`, model selection based on the Bayesian Information criterion is performed. If set to `"aic"`, the Akaike Information criterion is used. If set to `"icl"` the Integrated Classification Likelihood criterion is used. Default is `"bic"`.

`se`	String which can be set to `"no"`, `"yes"` or `"bs"`. If set to `"no"`, standard errors are not reported. If set to `"yes"`, analytic standard errors are reported. If set to `"bs"`, bootstrapped standard errors are reported for responses and trembles. Default is `"yes"`.
`outer.runs`	Positive integer which stets the number of outer runs of the solver. Default is 10.
`outer.tol`	Positive number which stets the tolerance of the continuation condition of the outer runs. The iterative algorithm stops if the relative decrease of the log-likelihood is smaller than `outer.tol`. Default is 0.
`outer.max`	Positive integer which stets the maximum number of iterations of the outer runs of the solver. The iterative algorithm stops if it did not converge after `"outer.max"` iterations. Default is 1000.
`inner.runs`	Positive integer which stets the number of inner runs of the solver. Default is 100.
`inner.tol`	Positive number which stets the tolerance of the continuation condition of the inner EM runs. The iterative algorithm stops if the relative decrease of the log-likelihood is smaller than `inner.tol`. Default is 0.
`inner.max`	Positive integer which stets the maximum number of iterations of the inner EM runs. The iterative algorithm stops if it did not converge after `inner.max` iterations. Default is 100.
`lcr.runs`	Positive integer which stets the number of estimation runs for latent class regression. Default is 100.
`lcr.tol`	Positive number which stets the tolerance of the continuation condition of the Latent Class Regression runs. The iterative algorithm stops if the relative decrease of the log-likelihood is smaller than `lcr.tol`. Default is 0.
`lcr.max`	Positive integer which stets the maximum number of iterations of the Latent Class Regression EM runs. The iterative algorithm stops if it did not converge after `lcr.max` iterations. Default is 1000.
`bs.samples`	Positive integer which sets the number of bootstrap samples drawn with replacement.
`print.messages`	Logical, if `TRUE` messages are printed which illustrate the status of the estimation process.

Details

The estimation function `stratEst()` returns maximum-likelihood estimates for the population shares and response parameters of a set of candidate strategies given some data from an economic experiment. Candidate strategies can be supplied by the user in the form of deterministic finite-state automata. The number and the complexity of strategies can be restricted by the user or selected based on information criteria. stratEst also features latent class regression to assess the influence of covariates on strategy choice.

Value

The function returns a list with the following elements.

`shares` Column vector which contains the estimates of population shares for the strategies. The first element corresponds to the first strategy defined in the strategy matrix, the second element to corresponds to the second strategy and to on. Can be used as input object of the estimation function.

`strategies` Matrix which contains the strategies of the model. Can be used as input object of the of the estimation function.

`responses` Column vector which contains the response probabilities of the strategies. The value -1 indicates that the corresponding response could not be estimated since data does not contain observations the model assigns to the corresponding state.

`trembles` Column vector which contains the tremble probabilities of the strategies. The value -1 indicates that the corresponding response could not be estimated since data does not contain observations the model assigns to the corresponding state.

`coefficients` Column vector which contains the Latent Class Regression coefficients for strategies.

`response.mat` Matrix which contains the estimates of the response probabilities for the columns of the strategy matrix which represent the response probabilities.

`tremble.mat` Matrix which contains the estimates of the tremble probabilities for the columns of the strategy matrix which represent the response probabilities.

`coefficient.mat`
Matrix which contains the latent class regression coefficients of strategies in columns. Note that the coefficients of the first strategy are one by definition.

`loglike` The log-Likelihood value corresponding to the reported estimates. Bigger values indicate a better fit of the model to the data.

`crit.val` The value of the selection criterion defined under `crit`. Bigger values indicate a better fit of the model.

`eval` Number of iterations of the solver. The reported number is the sum of iterations performed in the inner and the outer run which led to the reported estimates.

`tol.val` The tolerance value in the last iteration.

`entropy` Entropy of the assignments.

`assignments` Matrix which contains the posterior probability assignments of individuals to strategies. The rows of the matrix correspond to the ID sorted in ascending order beginning with the individual with the lowest ID. The columns correspond to the strategies, starting with the first strategy defined in the strategy matrix in column one.

`priors` Matrix which contains the individual prior probabilities of individuals as predicted by the covariate vectors of the individuals. The rows correspond to the ID sorted in ascending order beginning with the individual with the lowest ID. The columns correspond to the strategies, starting with the first strategy defined in the strategy matrix.

`shares.se` Column vector which contains the standard errors of the estimated shares. The elements correspond to the vector of estimates.

`responses.se` Column vector which contains the standard errors of the reported responses. The elements correspond to the vector of estimates.

`trembles.se` Column vector which contains the standard errors of the reported trembles. The elements correspond to the vector of estimates.

`coefficients.se` Column vector which contains the standard errors of the reported coefficients. The elements correspond to the vector of estimates.

`convergence` Row vector which reports the maximum value of the score vector of the shares as the first element, responses as the second element, trembles as the third element, and LCR coefficients as the forth element. Small values indicate convergence of the algorithm to a (local) maximum.

Note

The strategy estimation method was introduced by (Dal Bo & Frechette 2011) to estimate the relative frequency of a fixed set of pure strategies in the indefinitely repeated prisoner's dilemma. Breitmoser (2015) extended the method to the estimation of behavior strategies. The **stratEst** package uses the EM algorithm (Dempster, Laird & Rubin 1977) and the Newton-Raphson method to obtain maximum-likelihood estimates for the population shares and response parameters of a set of candidate strategies. The package builds on other software contributions of the R community. To increase speed the estimation procedures, the package uses integration of C++ and R achieved by the Rcpp package (Eddelbuettel & Francois 2011) and the open source linear algebra library for the C++ language RppArmadillo (Sanderson & Curtin 2016).

References

Breitmoser, Y. (2015): Cooperation, but no reciprocity: Individual strategies in the repeated prisoner's dilemma, *American Economic Review*, 105, 2882-2910.

Dal Bo, P. and G. R. Frechette (2011): The evolution of cooperation in infinitely repeated games: Experimental evidence, *American Economic Review*, 101, 411-429.

Dempster, A., N. Laird, and D. B. Rubin (1977): Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B*, 39, 1-38.

Eddelbuettel, D. and R. Francois (2011): Rcpp: Seamless R and C++ Integration, *Journal of Statistical Software*, 40, 1-18.

Sanderson, C. and R. Curtin (2016): Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software*, 1-26.

Examples

```
## Fictitious data from a helping game
## Participant 62 plays reciprocal strategy.
## Participant 87 plays alternating strategy.
id <- c(62,62,62,62,87,87,87,87)
game <- c(4,4,4,4,4,4,4,4)
period <- c(1,2,3,4,1,2,3,4)
input <- c(0,1,2,3,0,1,3,2)
output <- c(2,2,1,2,2,1,2,1)
data <- as.data.frame(cbind(id,game,period,input,output))
strategies <- matrix(c(1,2,3,1,2,0.5,0,1,0.1,NA,0.5,1,0,0.9,NA,2,2,2,2,1,
3,3,3,2,1,2,2,2,2,1,3,3,3,2,1),5,7)
```

```
model <- stratEst(data,strategies)

## Replication of Dal Bo and Frechette (2011), Table 7 on page 424
## Results for the first treatment with delta = 1/2 and R = 32 (column 1 of Table 7)
data <- DF2011[DF2011$treatment == 1,]
strategies <- rbind(ALLD,ALLC,GRIM,TFT,WSLS,T2)
stratEst(data,strategies)

## Latent class regression with data from Dal Bo and Frechette (2011)
## For the two treatments with R = 32, introduce a dummy which is one if delta = 3/4
data <- DF2011[DF2011$treatment == 1 | DF2011$treatment == 4,]
strats <- rbind(ALLD,TFT)
covar <- as.matrix(as.numeric(data$treatment == 4 ))
stratEst(data,strats,covariates = covar,lcr.runs = 500)
```

T2 *T2*

Description

Matrix representation of the prisoner's dilemma strategy which plays C until either player plays D, then it plays D twice and returns to C (regardless of all actions during the punishment rounds).

Usage

```
data(T2)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(T2,GRIM,ALLD,ALLC,TFT)
```

T2F2T *T2F2T*

Description

Matrix representation of the prisoner's dilemma strategy which plays C unless partner played 2 consecutive Ds in the last 3 rounds (2 rounds of punishment if partner plays D twice in a row).

Usage

```
data(T2F2T)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(T2F2T,GRIM,ALLD,ALLC,TFT)
```

T2FT *T2FT*

Description

Matrix representation of the prisoner's dilemma strategy which plays C unless partner played D in either of the last 2 rounds (2 rounds of punishment if partner plays D).

Usage

```
data(T2FT)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(T2FT,GRIM,ALLD,ALLC,TFT)
```

TF2T *TF2T*

Description

Matrix representation of the prisoner's dilemma strategy which plays C unless partner played D in both of the last 2 rounds.

Usage

```
data(TF2T)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(TF2T,GRIM,ALLD,ALLC,TFT)
```

TF3T *TF3T*

Description

Matrix representation of the prisoner's dilemma strategy which plays C unless partner played D in all of the last 3 rounds.

Usage

```
data(TF3T)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(TF3T,GRIM,ALLD,ALLC,TFT)
```

TFT *TFT*

Description

Matrix representation of the prisoner's dilemma strategy which plays C unless partner played D in the last round.

Usage

```
data(TFT)
```

Format

A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(TFT,GRIM,ALLD,ALLC,PTFT)
```

WSLS *WSLS*

Description

Matrix representation of the prisoner's dilemma strategy which plays C if both players chose the same move last round, otherwise it plays D.

Usage

```
data(WSLS)
```

Format

An object of class `matrix` with 2 rows and 6 columns.

Details

#' @format A matrix with 1 row and 6 columns:

Rows Each row corresponds to one states of the automaton.

Column 1 Enumerates the states of the automaton.

Column 2 Probability to play C given the current state of the automaton.

Column 3 State transition if the history of play in the last round was CC (input is 1).

Column 4 State transition if the history of play in the last round was CD (input is 2).

Column 5 State transition if the history of play in the last round was DC (input is 3).

Column 6 State transition if the history of play in the last round was DD (input is 4).

Examples

```
strategies <- rbind(PTFT,GRIM,ALLD,ALLC,TFT)
```

Index