

Package ‘ramchoice’

September 4, 2026

Type Package

Title Revealed Preference and Attention Analysis in Random Limited Attention Models

Version 3.0.0

Description Implements identification, estimation, inference, and specification procedures for random limited-attention models, including the Random Attention Model of Cattaneo, Ma, Masatlioglu, and Suleymanov (2020) <[doi:10.1086/706861](https://doi.org/10.1086/706861)> and the Attention Overload Model of Cattaneo, Cheung, Ma, and Masatlioglu (2026) <[doi:10.48550/arXiv.2110.10650](https://doi.org/10.48550/arXiv.2110.10650)>. The methods use standard choice data to partially identify preferences and attention and provide simulation-based procedures for statistical inference.

Imports lpSolve, MASS

Suggests testthat (>= 3.0.0)

Depends R (>= 3.1.0)

License GPL-2

URL <https://github.com/mdcattaneo/ramchoice>,
<https://arxiv.org/abs/2110.10650>

BugReports <https://github.com/mdcattaneo/ramchoice/issues>

Encoding UTF-8

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Matias D. Cattaneo [aut, cre],
Paul Cheung [aut],
Xinwei Ma [aut],
Yusufcan Masatlioglu [aut],
Elchin Suleymanov [aut]

Maintainer Matias D. Cattaneo <matias.d.cattaneo@gmail.com>

Repository CRAN

Date/Publication 2026-09-04 06:40:10 UTC

Contents

ramchoice-package	2
aomIdentify	3
aomModel	4
aomTest	5
genMat	7
hlaoEvent	9
hlaoModel	9
hlaoNoPITest	11
hlaoRankings	13
hlaoTest	14
logitAtte	16
logitSimu	17
ramdata	18
ramTest	18
rAtte	20
revealAtte	22
revealPref	24
revealPrefModel	26
sumData	28
Index	30

ramchoice-package	<i>ramchoice: Revealed Preference and Attention Analysis in Random Limited Attention Models</i>
-------------------	---

Description

Preferences and attention are important for understanding decision making, conducting welfare analysis, and providing robust policy recommendations. Decision makers may not pay full attention to all available alternatives, however, which can invalidate standard revealed preference analysis.

This package implements identification, estimation, inference, and specification procedures for the Random Attention Model of Cattaneo, Ma, Masatlioglu, and Suleymanov (2020; [doi:10.1086/706861](https://doi.org/10.1086/706861)) and the Attention Overload Model of Cattaneo, Cheung, Ma, and Masatlioglu (2026).

The principal RAM and homogeneous-AOM interfaces are `revealPref`, `ramTest`, `revealAtte`, `revealPrefModel`, `aomModel`, `aomTest`, and `aomIdentify`. The heterogeneous list-based AOM interfaces are `hlaoModel`, `hlaoTest`, `hlaoNoPITest`, `hlaoEvent`, and `hlaoRankings`. Data preparation and simulation utilities include `sumData`, `genMat`, `logitAtte`, and `logitSimu`. The legacy `rAtte` interface and simulated `ramdata` dataset are retained for compatibility and illustration.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hycheung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

See Also

Useful links:

- <https://github.com/mdcattaneo/ramchoice>
- <https://arxiv.org/abs/2110.10650>
- Report bugs at <https://github.com/mdcattaneo/ramchoice/issues>

aomIdentify

Population Identification for Homogeneous AOM

Description

‘aomIdentify’ implements the mixed-integer characterization of homogeneous Attention Overload. Binary variables encode pairwise comparisons, while totality, transitivity, and the observed ‘succ’-Regularity inequalities characterize the sharp set of compatible strict preferences. The routine tests model feasibility and all pairwise revealed-preference conclusions without enumerating the factorial collection of rankings. A status of 2 is treated as a solver-certified infeasibility result; any other nonzero status is reported as a solver error rather than as model incompatibility.

Usage

```
aomIdentify(menu, prob, tolerance = sqrt(.Machine$double.eps), pairwise = TRUE)
```

Arguments

menu	Numeric zero-one matrix with one row per observed menu.
prob	Numeric matrix of population choice probabilities with the same dimensions as ‘menu’.
tolerance	Nonnegative numerical tolerance added to the population inequalities.
pairwise	Logical; if ‘TRUE’, determine whether each direction of every pairwise comparison occurs in a compatible preference.

Value

An object of class ‘ramchoiceAOMIdentification’. It contains model ‘compatible’, one feasible ‘preference’ when the model is nonempty, pairwise possibility and revelation results, solver diagnostics, and the mixed-integer system used in the calculation.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```
menu <- prob <- matrix(c(
  1, 1, 1,
  1, 1, 0,
  1, 0, 1,
  0, 1, 1
), ncol = 3, byrow = TRUE)
for (i in seq_len(nrow(prob))) {
  prob[i, menu[i, ] == 1] <- logitAtte(sum(menu[i, ]), 2)$choiceProb
}
aomIdentify(menu, prob)
```

aomModel

*Population Analysis for the Homogeneous Attention Overload Model***Description**

‘aomModel’ evaluates the population choice-probability inequalities implied by a collection of candidate preference orderings under the homogeneous Attention Overload Model (AOM). It provides a model-specific interface to the AOM restrictions implemented by [revealPrefModel()].

Usage

```
aomModel(
  menu,
  prob,
  pref_list = NULL,
  tolerance = sqrt(.Machine$double.eps),
  attBinary = 1
)
```

Arguments

menu	Numeric matrix of zeros and ones. Each row identifies an observed menu.
prob	Numeric matrix of choice probabilities with the same dimensions as ‘menu’.

pref_list	Numeric matrix whose rows are candidate strict preference orderings. The default is '1, 2, ...'.
tolerance	Nonnegative numerical tolerance used when classifying a population inequality as violated.
attBinary	Numeric value between one half and one. Values below one impose the attentive-at-binaries restriction used by the legacy API.

Value

An object of class 'ramchoiceAOMModel'. Its 'results' component has one row per candidate preference, including compatibility, inequality counts, and violation magnitudes. The object also contains 'preferences', candidate-specific 'inequalities', the classification 'tolerance', and the complete legacy [revealPrefModel()] result.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```
menu <- prob <- matrix(c(
  1, 1, 1,
  1, 1, 0,
  1, 0, 1,
  0, 1, 1
), ncol = 3, byrow = TRUE)
for (i in seq_len(nrow(prob))) {
  prob[i, menu[i, ] == 1] <- logitAtte(sum(menu[i, ]), 2)$choiceProb
}
aomModel(menu, prob, pref_list = rbind(1:3, 3:1))
```

Description

'aomTest' tests candidate preference orderings under the homogeneous AOM and returns a tidy inference table. Row-i.i.d. calculations delegate to [revealPref()] with AOM restrictions only. When 'cluster' is supplied, the function uses cluster-level influence vectors and multiplier critical values while preserving the legacy result for backward-compatible auditing.

Usage

```
aomTest(
  menu,
  choice,
  pref_list = NULL,
  method = "GMS",
  alpha = 0.05,
  nCritSimu = 2000,
  BARatio2MS = 0.1,
  BARatio2UB = 0.1,
  MNRatioGMS = NULL,
  attBinary = 1,
  cluster = NULL
)
```

Arguments

menu	Numeric matrix of zeros and ones containing observed menus.
choice	Numeric matrix of zeros and ones containing observed choices.
pref_list	Numeric matrix whose rows are candidate strict preference orderings. The default is '1, 2, ...'.
method	Critical-value method: "GMS", "PI", "LF", "2MS", "2UB", or "ALL".
alpha	One or more nominal test levels chosen from '0.10', '0.05', and '0.01'.
nCritSimu	Number of Gaussian or cluster-multiplier simulations used for critical values.
BARatio2MS	Beta-to-alpha ratio for two-step moment selection.
BARatio2UB	Beta-to-alpha ratio for the two-step upper-bound method.
MNRatioGMS	Generalized moment-selection tuning parameter. 'NULL' uses '1/log(N)', where 'N' is the total sample size under row-i.i.d. sampling and the number of clusters under clustered sampling.
attBinary	Numeric value between one half and one. Values below one impose the attentive-at-binaries restriction used by the legacy API.
cluster	Optional vector identifying independent sampling clusters. When supplied, covariance estimation and Gaussian critical values use cluster-level influence vectors and multiplier draws.

Value

An object of class 'ramchoiceAOMTest'. Its 'results' component has one row per preference, method, and nominal level. The object also contains 'preferences', candidate-specific 'inequalities', menu-level 'summary' estimates, 'constraints', inference 'options', elapsed computation time, and the complete legacy [revealPref()] result.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```
set.seed(42)
simulated <- lapply(4:2, function(size) {
  logitSimu(n = 10, uSize = 4, mSize = size, a = 2)
})
menu <- do.call(rbind, lapply(simulated, `[`, "menu"))
choice <- do.call(rbind, lapply(simulated, `[`, "choice"))
aomTest(
  menu,
  choice,
  pref_list = rbind(1:4, 4:1),
  nCritSimu = 100
)
```

genMat	<i>Generate Constraint Matrices</i>
--------	-------------------------------------

Description

genMat generates constraint matrices for a range of preference orderings according to (i) the monotonic attention assumption proposed by Cattaneo, Ma, Masatlioglu, and Suleymanov (2020), (ii) the attention overload assumption proposed by Cattaneo, Cheung, Ma, and Masatlioglu (2021), and (iii) the attentive-at-binaries restriction.

This function is embedded in [revealPref](#).

Usage

```
genMat(
  sumMenu,
  sumMsize,
  pref_list = NULL,
  RAM = TRUE,
  AOM = TRUE,
  limDataCorr = TRUE,
  attBinary = 1
)
```

Arguments

sumMenu	Numeric matrix, summary of choice problems, returned by sumData .
sumMsize	Numeric matrix, summary of choice problem sizes, returned by sumData .
pref_list	Numeric matrix, each row corresponds to one preference. For example, c(2, 3, 1) means 2 is preferred to 3 and to 1. When set to NULL, the default, c(1, 2, 3, ...), will be used.

RAM	Boolean, whether the restrictions implied by the random attention model of Cattaneo, Ma, Masatlioglu, and Suleymanov (2020) should be incorporated, that is, their monotonic attention assumption (default is TRUE).
AOM	Boolean, whether the restrictions implied by the attention overload model of Cattaneo, Cheung, Ma, and Masatlioglu (2021) should be incorporated, that is, their attention overload assumption (default is TRUE).
limDataCorr	Boolean, whether assuming limited data (default is TRUE). When set to FALSE, will assume all choice problems are observed. This option only applies when RAM is set to TRUE.
attBinary	Numeric, between 1/2 and 1 (default is 1), whether additional restrictions (on the attention rule) should be imposed for binary choice problems (i.e., attentive at binaries).

Value

R	Matrices of constraints, stacked vertically.
ConstN	The number of constraints for each preference, used to extract from R individual matrices of constraints.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hycheung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). **Attention Overload**. Working paper.

Examples

```
# Load data
data(ramdata)

# Generate summary statistics
summaryStats <- sumData(ramdata$menu, ramdata$choice)

# Generate constraint matrices
constraints <- genMat(summaryStats$sumMenu, summaryStats$sumMsize)
constraints$ConstN
constraints$R[1:10, 1:10]
```

hlaoEvent	<i>Structured H-LAO Preference Event</i>
-----------	--

Description

‘hlaoEvent‘ describes the event that one alternative is strictly preferred to every alternative in a supplied comparison set. Unlike an indicator over enumerated rankings, this representation can be priced directly by the H-LAO column-generation algorithm.

Usage

```
hlaoEvent(alternative, preferred_to, name = NULL)
```

Arguments

- alternative Integer identifying the focal alternative.
- preferred_to Distinct integers identifying alternatives that the focal alternative must be preferred to.
- name Optional event label.

Value

An object of class ‘ramchoiceHLAOEvent‘.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). **Attention Overload**. Working paper.

Examples

```
hlaoEvent(4, 1, name = "4 above 1")
hlaoEvent(1, c(2, 3, 4), name = "1 top ranked")
```

hlaoModel	<i>Population Analysis for Heterogeneous List-Based Attention Overload</i>
-----------	--

Description

‘hlaoModel’ recovers list-based reach probabilities and prefix masses on a suffix-closed menu domain. It evaluates recovered-attention restrictions, constructs sharp independent, dependence-robust, or path-independence-robust preference polytopes, and computes sharp bounds for supplied preference events. The first two modes require a suffix-closed domain because they use Sequential Path Independence to recover attention. The “noPI” mode treats prefix masses as latent and is available on any observed-menu domain. With full menu data and positive terminal reach, the SPI modes also recover the full-attention choice rule and report Block–Marschak diagnostics. Optional agreement targets measure whether observed and full-attention choices agree. Under benchmark independence, structured events support status-checked column generation with mixed-integer pricing over linear orders. Returned diagnostics report solver statuses, tolerance, reduced costs, primal and dual residuals, an optimality-gap bound, and whether the numerical certificate checks succeeded.

Usage

```
hlaoModel(
  menu,
  prob,
  outside_prob = NULL,
  list_order = NULL,
  events = NULL,
  dependence = c("independent", "robust", "noPI", "both", "all"),
  tolerance = sqrt(.Machine$double.eps),
  agreement = FALSE,
  algorithm = c("auto", "enumerate", "column_generation"),
  max_rankings = 5000L,
  max_iterations = 1000L
)
```

Arguments

menu	Numeric matrix of zeros and ones with one row per distinct menu.
prob	Numeric matrix of inside choice probabilities with the same dimensions as ‘menu’.
outside_prob	Optional vector of outside-option probabilities. When omitted, it is computed as one minus the row sum of ‘prob’.
list_order	Permutation giving the observed presentation order. The default is the column order of ‘menu’.
events	Optional zero-one event indicators over the rows returned by [hlaoRankings()], or one or more structured [hlaoEvent()] objects.
dependence	Which population polytope to construct: “independent”, “robust”, “noPI”, “both”, or “all”. For backward compatibility, “both” continues to request the independent and dependence-robust SPI polytopes; “all” adds the no-SPI polytope.
tolerance	Nonnegative numerical tolerance for model diagnostics.
agreement	‘FALSE’, ‘TRUE’, or observed-menu indices. ‘TRUE’ computes full-attention agreement bounds for every observed menu.

algorithm	Computational method: "auto", "enumerate", or "column_generation". Column generation currently applies to the benchmark independent model and structured events.
max_rankings	Maximum number of ranking columns to enumerate.
max_iterations	Maximum number of master and pricing iterations under column generation.

Value

An object of class 'ramchoiceHLAOModel' containing recovered 'attention', attention 'diagnostics', population 'pairwise' shares, compatibility by dependence mode, event 'bounds', full-attention 'agreement', ranking columns used by the selected algorithm, computation diagnostics, and, when available, 'full_attention' and 'block_marschak' results.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```
menu <- rbind(c(1, 0), c(0, 1), c(1, 1))
prob <- rbind(c(.8, 0), c(0, .75), c(.56, .24))
rankings <- hlaoRankings(1:2)
event <- rankings[, 1] == 2
hlaoModel(menu, prob, events = list(`2 above 1` = event))
hlaoModel(
  menu, prob,
  events = hlaoEvent(2, 1, name = "2 above 1"),
  agreement = TRUE
)
```

 hlaoNoPITest

Path-Independence-Robust Inference for H-LAO

Description

'hlaoNoPITest' projects a simultaneous confidence region for primitive menu-choice probabilities through the sharp H-LAO model that retains prefix consideration, attention overload, and a stable marginal preference distribution but does not impose Sequential Path Independence. Prefix masses and menu-specific preference-stopping couplings are latent variables. Every reported event endpoint is obtained by linear programming, and the observed menu domain need not be suffix closed.

Usage

```
hlaoNoPITest(
  menu,
  choice,
  outside = NULL,
  list_order = NULL,
  events = NULL,
  alpha = 0.05,
  band_method = c("hoeffding", "gaussian"),
  n_band_draws = 2000L,
  boundary_count = 5L,
  tolerance = sqrt(.Machine$double.eps),
  max_rankings = 5000L,
  cluster = NULL
)
```

Arguments

menu	Zero-one matrix of menus, with one row per observation.
choice	Zero-one matrix of inside choices. An all-zero row denotes the outside option unless ‘outside’ is supplied.
outside	Optional zero-one indicator for outside choices.
list_order	Permutation giving the observed presentation order.
events	Optional zero-one event indicators over [hlaoRankings()].
alpha	Nominal error probability for the common simultaneous region.
band_method	Probability-band method, either “hoeffding” or “gaussian”.
n_band_draws	Number of Gaussian or cluster-multiplier draws used by the covariance-aware probability band.
boundary_count	Minimum number of successes and failures required for a cell to use the Gaussian band. Under clustered sampling this counts clusters with successes and failures.
tolerance	Nonnegative numerical tolerance.
max_rankings	Maximum ranking count used for event projection.
cluster	Optional vector identifying independent sampling clusters. When supplied, covariance estimation and Gaussian calibration use cluster-level influence vectors and multiplier draws.

Value

An object of class ‘ramchoiceHLAONoPITest’ containing event ‘intervals’, simultaneous probability ‘bands’, the LP ‘projection’, options, and elapsed time.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```

menu <- rbind(
  matrix(rep(c(1, 0), 20), ncol = 2, byrow = TRUE),
  matrix(rep(c(1, 1), 20), ncol = 2, byrow = TRUE)
)
choice <- matrix(0, nrow = nrow(menu), ncol = 2)
choice[1:15, 1] <- 1
choice[21:30, 1] <- 1
choice[31:36, 2] <- 1
rankings <- hlaoRankings(1:2)
hlaoNoPITest(
  menu, choice,
  events = list(`2 above 1` = rankings[, 1] == 2)
)

```

hlaoRankings

*Enumerate Strict Preference Rankings***Description**

‘hlaoRankings’ returns all strict rankings of a supplied alternative set in the deterministic ordering used by the H-LAO linear-programming routines.

Usage

```
hlaoRankings(alternatives)
```

Arguments

alternatives Vector of distinct alternative labels.

Value

A matrix with one strict ranking per row, from best to worst.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```
hlaoRankings(1:3)
```

Description

‘hlaoTest’ forms simultaneous bands for all observed menu–outcome probabilities. The default Hoeffding method is finite-sample valid. The correlated-Gaussian method uses the estimated block-multinomial covariance and retains exact binomial bands for sparse or degenerate cells. When ‘cluster’ is supplied, the Gaussian component instead uses cluster-level influence vectors and multiplier draws, and the fallback is a cluster-Hoeffding band. When both types of cells are present, each component receives half of the common error budget. The function inverts the undivided binary-menu moments to obtain simultaneous pairwise preference-share intervals that remain valid at zero reach. It also reports a Bonferroni-calibrated studentized inversion of the same moments. The studentized set is obtained by exact quadratic inversion and may therefore contain more than one component; exactly degenerate moments return $[0,1]$. For supplied general preference events, the function also computes the dependence-robust outer projection intervals described in the Supplemental Appendix.

Usage

```
hlaoTest(
  menu,
  choice,
  outside = NULL,
  list_order = NULL,
  events = NULL,
  alpha = 0.05,
  band_method = c("hoeffding", "gaussian"),
  diagnostic_method = c("outer", "delta"),
  n_band_draws = 2000L,
  boundary_count = 5L,
  tolerance = sqrt(.Machine$double.eps),
  max_rankings = 5000L,
  cluster = NULL
)
```

Arguments

menu	Zero-one matrix of menus, with one row per observation.
choice	Zero-one matrix of inside choices. An all-zero row denotes the outside option unless ‘outside’ is supplied.
outside	Optional zero-one indicator for outside choices.
list_order	Permutation giving the observed presentation order.
events	Optional zero-one event indicators over [hlaoRankings()].
alpha	Nominal error probability for the common simultaneous region.

<code>band_method</code>	Probability-band method, either <code>"hoeffding"</code> or <code>"gaussian"</code> .
<code>diagnostic_method</code>	Specification-diagnostic method. <code>"outer"</code> uses the simultaneous probability region and is the finite-sample default. <code>"delta"</code> uses a direct delta-Gaussian approximation on a complete menu domain with positive terminal reach.
<code>n_band_draws</code>	Number of Gaussian or cluster-multiplier draws used for the simultaneous band and direct-diagnostic critical values.
<code>boundary_count</code>	Minimum number of successes and failures required for a cell to use the Gaussian band. Under clustered sampling this counts clusters with successes and failures. Other cells retain the applicable simultaneous fallback band.
<code>tolerance</code>	Nonnegative numerical tolerance for zero-reach conventions.
<code>max_rankings</code>	Maximum ranking count used for general-event projection.
<code>cluster</code>	Optional vector identifying independent sampling clusters. When supplied, covariance estimation and Gaussian calibration use cluster-level influence vectors and multiplier draws.

Value

An object of class `'ramchoiceHLAOTest'` containing aggregated choice `'summary'`, plug-in `'attention'` and `'full_attention'` estimates, simultaneous probability `'bands'`, weak-reach `'pairwise'` intervals, studentized `'pairwise_studentized'` sets and their `'pairwise_studentized_components'`, optional general-event `'event_intervals'`, simultaneous specification diagnostics, projection dimensions, options, and elapsed time.

References

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). [Attention Overload](#). Working paper.

Examples

```
menu <- rbind(
  matrix(rep(c(1, 0), 10), ncol = 2, byrow = TRUE),
  matrix(rep(c(0, 1), 10), ncol = 2, byrow = TRUE),
  matrix(rep(c(1, 1), 10), ncol = 2, byrow = TRUE)
)
choice <- matrix(0, nrow = nrow(menu), ncol = 2)
choice[1:8, 1] <- 1
choice[11:17, 2] <- 1
choice[21:26, 1] <- 1
choice[27:28, 2] <- 1
hlaoTest(menu, choice)
```

logitAtte

*Compute Choice Probabilities and Attention Frequencies for the Logit Attention Rule***Description**

logitAtte computes choice probabilities and attention frequencies for the logit attention rule considered by Brady and Rehbeck (2016). To be specific, for a choice problem S and its subset T , the attention that T attracts is assumed to be proportional to its size: $|T|^a$, where a is a parameter that one can specify. It will be assumed that the first alternative is the most preferred, and that the last alternative is the least preferred.

This function is useful for replicating the simulation results in Cattaneo, Ma, Masatlioglu, and Suleymanov (2020; doi:10.1086/706861) and Cattaneo, Cheung, Ma, and Masatlioglu (2026).

Usage

```
logitAtte(mSize = NULL, a = NULL)
```

Arguments

mSize	Positive integer, size of the choice problem.
a	Numeric, the parameter of the logit attention rule.

Value

choiceProb	The vector of choice probabilities.
atteFreq	The attention frequency.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hycheung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

- R. L. Brady and J. Rehbeck (2016). Menu-Dependent Stochastic Feasibility. *Econometrica* 84(3): 1203–1223. doi:10.3982/ECTA12694
- M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861
- M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

Examples

```
logitAtte(mSize = 5, a = 2)
```

logitSimu

Choice Data Simulation Following the Logit Attention Rule

Description

logitSimu simulates choice data according to the logit attention rule considered by Brady and Rehbeck (2016). To be specific, for a choice problem S and its subset T , the attention that T attracts is assumed to be proportional to its size: $|T|^a$, where a is a parameter that one can specify. It will be assumed that the first alternative is the most preferred, and that the last alternative is the least preferred.

This function is useful for replicating the simulation results in Cattaneo, Ma, Masatlioglu, and Suleymanov (2020; doi:10.1086/706861) and Cattaneo, Cheung, Ma, and Masatlioglu (2026).

Usage

```
logitSimu(n, uSize, mSize, a)
```

Arguments

<code>n</code>	Positive integer, the effective sample size for each choice problem.
<code>uSize</code>	Positive integer, total number of alternatives.
<code>mSize</code>	Positive integer, size of the choice problem.
<code>a</code>	Numeric, the parameter of the logit attention rule.

Value

<code>menu</code>	The choice problems.
<code>choice</code>	The simulated choices.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hycheung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

R. L. Brady and J. Rehbeck (2016). Menu-Dependent Stochastic Feasibility. *Econometrica* 84(3): 1203-1223. doi:10.3982/ECTA12694

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

Examples

```
set.seed(42)
logitSimu(n = 5, uSize = 6, mSize = 5, a = 2)
```

ramdata	<i>ramdata: Simulated Choice Data</i>
---------	---------------------------------------

Description

The file contains a standard choice data of 9,000 observations. There are five alternatives in the grand set.

See [revealPref](#) for revealed preference analysis, and [revealAtte](#) for revealed attention. [sumData](#) is a low-level function that computes summary statistics, and [genMat](#) generates constraint matrices subject to given preferences.

Format

- menu** Numeric matrix of 0s and 1s, choice problems (1 indicates an alternative in the choice problem and 0 otherwise).
- choice** Numeric matrix of 0s and 1s, choices (1 indicates an alternative being chosen).

ramTest	<i>Sample Inference for the Random Attention Model</i>
---------	--

Description

‘ramTest’ provides tidy candidate-ranking inference for the Random Attention Model of Cattaneo, Ma, Masatlioglu, and Suleymanov (2020). Row-i.i.d. calculations use [revealPref()]. When ‘cluster’ is supplied, the function retains the same RAM inequalities but estimates their joint covariance from cluster influence vectors and uses multiplier critical values.

Usage

```
ramTest(
  menu,
  choice,
  pref_list = NULL,
  method = "GMS",
  alpha = 0.05,
  nCritSimu = 2000,
  BARatio2MS = 0.1,
  BARatio2UB = 0.1,
  MNRatioGMS = NULL,
  attBinary = 1,
  limDataCorr = TRUE,
  cluster = NULL
)
```

Arguments

menu	Numeric matrix of zeros and ones containing observed menus.
choice	Numeric matrix of zeros and ones containing observed choices.
pref_list	Numeric matrix whose rows are candidate strict preference orderings. The default is '1, 2, ...'.
method	Critical-value method: "GMS", "PI", "LF", "2MS", "2UB", or "ALL".
alpha	One or more nominal test levels chosen from '0.10', '0.05', and '0.01'.
nCritSimu	Number of Gaussian or cluster-multiplier simulations used for critical values.
BARatio2MS	Beta-to-alpha ratio for two-step moment selection.
BARatio2UB	Beta-to-alpha ratio for the two-step upper-bound method.
MNRatioGMS	Generalized moment-selection tuning parameter. 'NULL' uses '1/log(N)', where 'N' is the total sample size under row-i.i.d. sampling and the number of clusters under clustered sampling.
attBinary	Numeric value between one half and one. Values below one impose the attentive-at-binaries restriction used by the legacy API.
limDataCorr	Logical indicating whether to use the limited-menu-domain correction from the legacy RAM implementation.
cluster	Optional vector identifying independent sampling clusters. When supplied, covariance estimation and Gaussian critical values use cluster-level influence vectors and multiplier draws.

Value

An object of class 'ramchoiceRAMTest' with the same tidy components as [aomTest()] and a complete legacy [revealPref()] result.

References

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:[10.1086/706861](https://doi.org/10.1086/706861)

rAtte

Revealed Preference Analysis in Random Limited Attention Models

Description

This has been replaced by [revealPref](#).

Usage

```
rAtte(
  menu,
  choice,
  pref_list = NULL,
  method = "GMS",
  nCritSimu = 2000,
  BARatio2MS = 0.1,
  BARatio2UB = 0.1,
  MNRatioGMS = NULL,
  RAM = TRUE,
  AOM = TRUE,
  limDataCorr = TRUE,
  attBinary = 1
)
```

Arguments

menu	Numeric matrix of 0s and 1s, the collection of choice problems.
choice	Numeric matrix of 0s and 1s, the collection of choices.
pref_list	Numeric matrix, each row corresponds to one preference. For example, c(2, 3, 1) means 2 is preferred to 3 and to 1. When set to NULL, the default, c(1, 2, 3, ...), will be used.
method	String, the method for constructing critical values. Default is GMS (generalized moment selection). Other available options are LF (least favorable model), PI (plug-in method), 2MS (two-step moment selection), 2UB (two-step moment upper bound), or ALL (report all critical values).
nCritSimu	Integer, number of simulations used to construct the critical value. Default is 2000.
BARatio2MS	Numeric, beta-to-alpha ratio for two-step moment selection method. Default is 0.1.
BARatio2UB	Numeric, beta-to-alpha ratio for two-step moment upper bound method. Default is 0.1.

MNRatioGMS	Numeric, tuning parameter. Default is $1/\log(N)$, where N is the total sample size; the GMS recentering uses its square root, $1/\sqrt{\log(N)}$.
RAM	Boolean, whether the restrictions implied by the RAM of Cattaneo et al. (2020; doi:10.1086/706861) should be incorporated, that is, their monotonic attention assumption (default is TRUE).
AOM	Boolean, whether the restrictions implied by the AOM of Cattaneo et al. (2026) should be incorporated, that is, their attention overload assumption (default is TRUE).
limDataCorr	Boolean, whether assuming limited data (default is TRUE). When set to FALSE, will assume all choice problems are observed. This option only applies when RAM is set to TRUE.
attBinary	Numeric, between $1/2$ and 1 (default is 1), whether additional restrictions (on the attention rule) should be imposed for binary choice problems (i.e., attentive at binaries).

Value

sumStats	Summary statistics, generated by sumData .
constraints	Matrices of constraints, generated by genMat .
Tstat	Test statistic.
critVal	Critical values.
pVal	P-values (only available for GMS, LF and PI).
method	Method for constructing critical value.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hycheung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

- M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. [doi:10.1086/706861](https://doi.org/10.1086/706861)
- M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

 revealAtte

Revealed Attention Analysis in Random Limited Attention Models

Description

Given a random sample of choice problems and choices, `revealAtte` returns the upper and lower bounds on the attention frequency following the construction of [Cattaneo, Cheung, Ma, and Masatlioglu \(2026\)](#).

`sumData` is a low-level function that generates summary statistics. For revealed preference analysis, see `revealPref`.

Usage

```
revealAtte(
  menu,
  choice,
  alternative = NULL,
  S = NULL,
  lower = TRUE,
  upper = TRUE,
  pref = NULL,
  nCritSimu = 2000,
  level = 0.95
)
```

Arguments

<code>menu</code>	Numeric matrix of 0s and 1s, the collection of choice problems.
<code>choice</code>	Numeric matrix of 0s and 1s, the collection of choices.
<code>alternative</code>	Numeric vector, the alternatives for which to compute bounds on the attention frequency. For example, <code>c(1, 2, 4)</code> means the first, second, and fourth alternatives.
<code>S</code>	Numeric matrix of 0s and 1s, the collection of choice problems to compute bounds on the attention frequency.
<code>lower</code>	Boolean, whether lower bounds should be computed (default is TRUE).
<code>upper</code>	Boolean, whether upper bounds should be computed (default is TRUE).
<code>pref</code>	Numeric vector, corresponding to the preference. For example, <code>c(2, 3, 1)</code> means 2 is preferred to 3 and to 1. When set to NULL, the default, <code>c(1, 2, 3, ...)</code> , will be used. This option only applies to the upper bounds (i.e., when <code>upper</code> is set to TRUE).
<code>nCritSimu</code>	Integer, number of simulations used to construct the critical value. Default is 2000.
<code>level</code>	Numeric, the significance level (default is 0.95).

Value

sumStats	Summary statistics, generated by sumData .
lowerBound	Matrix containing the lower bounds.
upperBound	Matrix containing the upper bounds.
critVal	The simulated critical value.
opt	Options used in the function call.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hycheung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861

M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

Examples

```
# Load data
data(ramdata)

# Set seed, to replicate simulated critical values
set.seed(42)

# preference
pref <- matrix(c(1, 2, 3, 4, 5), ncol=5, byrow=TRUE)
# list of choice problems
S <- matrix(c(1, 1, 0, 0, 0,
              1, 1, 1, 0, 0,
              1, 1, 1, 0, 1,
              1, 1, 1, 1, 1), ncol=5, byrow=TRUE)
result <- revealAtte(menu = ramdata$menu, choice = ramdata$choice,
                    alternative = c(1,2), S = S,
                    lower = TRUE, upper = TRUE,
                    pref = pref)
summary(result)
```

Description

Given a random sample of choice problems and choices, `revealPref` returns test statistics, critical values and p-values against a collection of preferences. Five methods for choosing critical values are available: (i) GMS: generalized moment selection (plug-in (estimated) moment conditions with shrinkage); (ii) PI: critical values based on plug-in estimated moment conditions (this is not uniformly valid); (iii) LF: critical values based on the least favorable model (plug-in 0 for the moment conditions); (iv) 2MS: two-step moment selection; and (v) 2UB: refined moment selection (plug-in upper bound of moment inequalities).

`sumData` is a low-level function that generates summary statistics, and `genMat` can be used to construct the constraint matrices. The simulated dataset `ramdata` is also provided for illustration. For revealed attention analysis, see `revealAtte`.

Usage

```
revealPref(
  menu,
  choice,
  pref_list = NULL,
  method = "GMS",
  nCritSimu = 2000,
  BARatio2MS = 0.1,
  BARatio2UB = 0.1,
  MNRatioGMS = NULL,
  RAM = TRUE,
  AOM = TRUE,
  limDataCorr = TRUE,
  attBinary = 1
)
```

Arguments

<code>menu</code>	Numeric matrix of 0s and 1s, the collection of choice problems.
<code>choice</code>	Numeric matrix of 0s and 1s, the collection of choices.
<code>pref_list</code>	Numeric matrix, each row corresponds to one preference. For example, <code>c(2, 3, 1)</code> means 2 is preferred to 3 and to 1. When set to <code>NULL</code> , the default, <code>c(1, 2, 3, ...)</code> , will be used.
<code>method</code>	String, the method for constructing critical values. Default is GMS (generalized moment selection). Other available options are LF (least favorable model), PI (plug-in method), 2MS (two-step moment selection), 2UB (two-step moment upper bound), or ALL (report all critical values).
<code>nCritSimu</code>	Integer, number of simulations used to construct the critical value. Default is 2000.

BARatio2MS	Numeric, beta-to-alpha ratio for two-step moment selection method. Default is 0.1.
BARatio2UB	Numeric, beta-to-alpha ratio for two-step moment upper bound method. Default is 0.1.
MNRatioGMS	Numeric, tuning parameter. Default is $1/\log(N)$, where N is the total sample size; the GMS recentering uses its square root, $1/\sqrt{\log(N)}$.
RAM	Boolean, whether the restrictions implied by the RAM of Cattaneo et al. (2020; doi:10.1086/706861) should be incorporated, that is, their monotonic attention assumption (default is TRUE).
AOM	Boolean, whether the restrictions implied by the AOM of Cattaneo et al. (2026) should be incorporated, that is, their attention overload assumption (default is TRUE).
limDataCorr	Boolean, whether assuming limited data (default is TRUE). When set to FALSE, will assume all choice problems are observed. This option only applies when RAM is set to TRUE.
attBinary	Numeric, between 1/2 and 1 (default is 1), whether additional restrictions (on the attention rule) should be imposed for binary choice problems (i.e., attentive at binaries).

Value

sumStats	Summary statistics, generated by sumData .
constraints	Matrices of constraints, generated by genMat .
Tstat	Test statistic.
critVal	Critical values.
pVal	P-values (only available for GMS, LF and PI).
method	Method for constructing critical value.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.
 Paul Cheung, University of Maryland. <hycheung@umd.edu>
 Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>
 Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>
 Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. [doi:10.1086/706861](https://doi.org/10.1086/706861)
 M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

Examples

```
# Load data
data(ramdata)

# Set seed, to replicate simulated critical values
set.seed(42)

# list of preferences
pref_list <- matrix(c(1, 2, 3, 4, 5,
                      2, 1, 3, 4, 5,
                      2, 3, 4, 5, 1,
                      5, 4, 3, 2, 1), ncol=5, byrow=TRUE)

# revealed preference using only RAM restrictions
result1 <- revealPref(menu = ramdata$menu, choice = ramdata$choice, method = "GMS",
  pref_list = pref_list, RAM = TRUE, AOM = FALSE)
summary(result1)

# revealed preference using only AOM restrictions
result2 <- revealPref(menu = ramdata$menu, choice = ramdata$choice, method = "GMS",
  pref_list = pref_list, RAM = FALSE, AOM = TRUE)
summary(result2)

# revealed preference using both RAM and AOM restrictions
result3 <- revealPref(menu = ramdata$menu, choice = ramdata$choice, method = "GMS",
  pref_list = pref_list, RAM = TRUE, AOM = TRUE)
summary(result3)

# revealed preference employing additional restrictions for binary choice problems
result4 <- revealPref(menu = ramdata$menu, choice = ramdata$choice, method = "GMS",
  pref_list = pref_list, RAM = TRUE, AOM = TRUE, attBinary = 2/3)
summary(result4)
```

 revealPrefModel

Model Falsification with Random Limited Attention

Description

Given a collection of choice problems and corresponding choice probabilities, `revealPrefModel` determines if they are compatible with the Random Attention Model (RAM) of Cattaneo, Ma, Masatlioglu, and Suleymanov (2020; [doi:10.1086/706861](https://doi.org/10.1086/706861)) and/or the Attention Overload Model (AOM) of Cattaneo, Cheung, Ma, and Masatlioglu (2026).

See `revealPref` for revealed preference analysis with empirical choice data.

Usage

```
revealPrefModel(
  menu,
```

```

    prob,
    pref_list = NULL,
    RAM = TRUE,
    AOM = TRUE,
    limDataCorr = TRUE,
    attBinary = 1
)

```

Arguments

menu	Numeric matrix of 0s and 1s, the collection of choice problems.
prob	Numeric matrix, the collection of choice probabilities
pref_list	Numeric matrix, each row corresponds to one preference. For example, <code>c(2, 3, 1)</code> means 2 is preferred to 3 and to 1. When set to <code>NULL</code> , the default, <code>c(1, 2, 3, ...)</code> , will be used.
RAM	Boolean, whether the restrictions implied by the RAM of Cattaneo et al. (2020; doi:10.1086/706861) should be incorporated, that is, their monotonic attention assumption (default is <code>TRUE</code>).
AOM	Boolean, whether the restrictions implied by the AOM of Cattaneo et al. (2026) should be incorporated, that is, their attention overload assumption (default is <code>TRUE</code>).
limDataCorr	Boolean, whether assuming limited data (default is <code>TRUE</code>). When set to <code>FALSE</code> , will assume all choice problems are observed. This option only applies when RAM is set to <code>TRUE</code> .
attBinary	Numeric, between 1/2 and 1 (default is 1), whether additional restrictions (on the attention rule) should be imposed for binary choice problems (i.e., attentive at binaries).

Value

constraints	Matrices of constraints, generated by genMat . R: a matrix containing all constraints. ConstN: number of constraints for each preference.
inequalities	The moment inequalities. Positive numbers indicate that the RAM/AOM restrictions are rejected by the given choice probabilities. R: a vector containing all moment inequalities. ConstN: number of constraints for each preference.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.

Paul Cheung, University of Maryland. <hychung@umd.edu>

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>

Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>

Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

- M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861
- M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). *Attention Overload*. Working paper.

Examples

```
# Logit attention with parameter 2
# True preference: 1 2 3 4 5 6
menu <- prob <- matrix(c(1, 1, 1, 1, 1, 1,
                        0, 1, 1, 1, 1, 1,
                        1, 0, 1, 1, 1, 1,
                        1, 1, 0, 1, 1, 1,
                        1, 1, 1, 0, 1, 1,
                        1, 1, 1, 1, 0, 1,
                        1, 1, 1, 1, 1, 0), ncol=6, byrow=TRUE)
for (i in 1:nrow(prob)) prob[i, menu[i, ]==1] <- logitAtte(sum(menu[i, ]), 2)$choiceProb

# List of preferences to be tested
pref_list <- matrix(c(1, 2, 3, 4, 5, 6,
                    2, 3, 4, 5, 6, 1), ncol=6, byrow=TRUE)

# RAM only
result1 <- revealPrefModel(menu = menu, prob = prob, pref_list = pref_list, RAM = TRUE, AOM = FALSE)
summary(result1)

# AOM only
result2 <- revealPrefModel(menu = menu, prob = prob, pref_list = pref_list, RAM = FALSE, AOM = TRUE)
summary(result2)

# Both RAM and AOM
result3 <- revealPrefModel(menu = menu, prob = prob, pref_list = pref_list, RAM = TRUE, AOM = TRUE)
summary(result3)
```

sumData

Generate Summary Statistics

Description

sumData generates summary statistics. Given a collection of choice problems and corresponding choices, sumData calculates the number of occurrences of each choice problem, as well as the empirical choice probabilities.

This function is embedded in [revealPref](#).

Usage

```
sumData(menu, choice)
```

Arguments

menu	Numeric matrix of 0s and 1s, the collection of choice problems.
choice	Numeric matrix of 0s and 1s, the collection of choices.

Value

sumMenu	Summary of choice problems, with repetitions removed.
sumProb	Estimated choice probabilities as sample averages for different choice problems.
sumN	Effective sample size for each choice problem.
sumMsize	Size of each choice problem.
sumProbVec	Estimated choice probabilities as sample averages, collapsed into a column vector.
Sigma	Estimated variance-covariance matrix for the choice rule, scaled by relative sample sizes.

Author(s)

Matias D. Cattaneo (maintainer), Princeton University. <matias.d.cattaneo@gmail.com>.
 Paul Cheung, University of Maryland. <hycheung@umd.edu>
 Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>
 Yusufcan Masatlioglu, University of Maryland. <yusufcan@umd.edu>
 Elchin Suleymanov, Purdue University. <esuleyma@purdue.edu>

References

M. D. Cattaneo, X. Ma, Y. Masatlioglu, and E. Suleymanov (2020). A Random Attention Model. *Journal of Political Economy* 128(7): 2796–2836. doi:10.1086/706861
 M. D. Cattaneo, P. H. Y. Cheung, X. Ma, and Y. Masatlioglu (2026). **Attention Overload**. Working paper.

Examples

```
# Load data
data(ramdata)

# Generate summary statistics
summaryStats <- sumData(ramdata$menu, ramdata$choice)
nrow(summaryStats$sumMenu)
min(summaryStats$sumN)

summaryStats$sumMenu[1, ]
summaryStats$sumProb[1, ]
summaryStats$sumN[1]
```

Index

aomIdentify, [2](#), [3](#)
aomModel, [2](#), [4](#)
aomTest, [2](#), [5](#)

choice (ramdata), [18](#)

genMat, [2](#), [7](#), [18](#), [21](#), [24](#), [25](#), [27](#)

hlaoEvent, [2](#), [9](#)
hlaoModel, [2](#), [9](#)
hlaoNoPITest, [2](#), [11](#)
hlaoRankings, [2](#), [13](#)
hlaoTest, [2](#), [14](#)

logitAtte, [2](#), [16](#)
logitSimu, [2](#), [17](#)

menu (ramdata), [18](#)

ramchoice (ramchoice-package), [2](#)
ramchoice-package, [2](#)
ramdata, [2](#), [18](#), [24](#)
ramTest, [2](#), [18](#)
rAtte, [2](#), [20](#)
revealAtte, [2](#), [18](#), [22](#), [24](#)
revealPref, [2](#), [7](#), [18](#), [20](#), [22](#), [24](#), [26](#), [28](#)
revealPrefModel, [2](#), [26](#)

sumData, [2](#), [7](#), [18](#), [21–25](#), [28](#)