Package ‘QCApro’

January 11, 2018

Version 1.1-2

Date 2018-01-10

Title Advanced Functionality for Performing and Evaluating Qualitative Comparative Analysis

URL http://www.alrik-thiem.net/;
https://www.researchgate.net/profile/Alrik_Thiem

Depends R (>= 3.0.0)

Suggests testthat (<= 2.0.0)

Imports lpSolve, utils

Description Provides advanced functionality for performing configurational comparative research with Qualitative Comparative Analysis (QCA), including crisp-set, multi-value, and fuzzy-set QCA. It also offers advanced tools for sensitivity diagnostics and methodological evaluations of QCA.

License GPL-3

NeedsCompilation yes

Author Alrik Thiem [aut, cre, cph],
Michael Baumgartner [ctb],
Adrian Dusa [ctb],
Reto Spoehel [ctb]

Maintainer Alrik Thiem <alrik.thiem@unilu.ch>

Repository CRAN

Date/Publication 2018-01-11 00:11:33 UTC

R topics documented:

`QCApro-package` *Advanced Functionality for Performing and Evaluating Qualitative Comparative Analysis*

Description

QCApro is a successor package to the **QCA** package, with **QCA** 1.1-4 as its original basis (Dusa and Thiem 2014; Thiem and Dusa 2012; 2013a; 2013b; 2013c). Just like its predecessor, **QCApro** implements the method of *Qualitative Comparative Analysis* (QCA)—a family of techniques for analyzing configurational data in accordance with the INUS theory of causation (Mackie 1965; 1974), but it has fixed various technical and methodological problems of the **QCA** package and includes many new features and enhancements for applying QCA.

Moreover, **QCApro** is currently the only QCA software that provides many purpose-built functions for testing methodological properties of QCA and QCA-related procedures. For example, the effects of changing discretionary parameters such as the inclusion cut-off on the degree of ambiguity affecting a QCA solution can be analyzed (Baumgartner and Thiem 2017a), the consequences of increasing limited empirical diversity on the probability of QCA not committing causal fallacies can be computed (Baumgartner and Thiem 2017b), and the relation between correlational and implicational independence can be examined (Thiem and Baumgartner 2016).

Three variants can currently be processed by **QCApro**: *crisp-set QCA* (csQCA; Ragin 1987), *multi-value QCA* (mvQCA; Cronqvist and Berg-Schlosser 2009; Thiem 2013; 2014) and *fuzzy-set QCA* (fsQCA; Ragin 2000; 2008). A subvariant of csQCA called *temporal QCA* (tQCA) is also available (Caren and Panofsky 2005; Ragin and Strand 2008).

Several datasets from various areas are integrated in **QCApro** so as to facilitate familiarization with the package's functionality. Currently covered are business, management and organization (`d.stakeholder`), education (`d.education`), environmental sciences (`d.biodiversity`), evaluation (`d.transport`), legal studies (`d.napoleon`), political science (`d.jobsecurity`, `d.partybans`, `d.represent`), public health (`d.health`, `d.tumorscreen`), urban affairs (`d.urban`), and sociology (`d.homeless`, `d.socialsecurity`). For more details, see the datasets' documentation files.

As an additional resource, **QCApro** includes a comprehensive glossary for Configurational Comparative Methods. The glossary is directly accessible via the link 'User guides, package vignettes and other documentation' in the package's help index or the 'doc' folder of the package's installation folder.

If you make use of the **QCApro** package in your work, please acknowledge it in the interest of good scientific practice and transparency. The package citation displays on loading the package or by using the command `citation(package = "QCApro")` after loading. The aforesaid command also provides a suitable BibTeX entry. To browse the latest news about the **QCApro** package (bug fixes, enhancements, etc.), enter `news(package = "QCApro")`.

Happy QCAing!

Details

Package:	QCApro
Type:	Package
Version:	1.1-2
Date:	2018-01-10
License:	GPL-3

Author(s)

Author:
Alrik Thiem
Department of Political Science
University of Lucerne, Switzerland
E-Mail
Personal Website
ResearchGate Website

Maintainer:
Alrik Thiem
Department of Political Science
University of Lucerne, Switzerland
E-Mail

Personal Website
ResearchGate Website

References

Baumgartner, Michael, and Alrik Thiem. 2017a. “Model Ambiguities in Configurational Comparative Research.” *Sociological Methods & Research* **46** (4):954-87. DOI: 10.1177/0049124115610351.

Baumgartner, Michael, and Alrik Thiem. 2017b. “Often Trusted but Never (Properly) Tested: Evaluating Qualitative Comparative Analysis.” *Sociological Methods & Research*. Advance online publication. DOI: 10.1177/0049124117701487.

Caren, Neal, and Aaron Panofsky. 2005. “TQCA: A Technique for Adding Temporality to Qualitative Comparative Analysis.” *Sociological Methods & Research* **34** (2):147-72. DOI: 10.1177/0049124105277197.

Cronqvist, Lasse, and Dirk Berg-Schlosser. 2009. “Multi-Value QCA (mvQCA).” In *Configurational Comparative Methods: Qualitative Comparative Analysis (QCA) and Related Techniques*, eds. B. Rihoux and C. C. Ragin. London: Sage Publications, pp. 69-86.

Dusa, Adrian, and Alrik Thiem. 2014. *QCA: A Package for Qualitative Comparative Analysis*. R Package Version 1.1-4. URL: `http://www.alrik-thiem.net/software/`.

Mackie, John L. 1965. “Causes and Conditions.” *American Philosophical Quarterly* **2** (4):245-64. URL: http://www.jstor.org/stable/20009173.

Mackie, John L. 1974. *The Cement of the Universe: A Study of Causation*. Oxford: Oxford University Press.

Ragin, Charles C. 1987. *The Comparative Method: Moving beyond Qualitative and Quantitative Strategies*. Berkeley: University of California Press.

Ragin, Charles C. 2000. *Fuzzy-Set Social Science*. Chicago: University of Chicago Press.

Ragin, Charles C. 2008. *Redesigning Social Inquiry: Fuzzy Sets and Beyond*. Chicago: University of Chicago Press.

Ragin, Charles C., and Sarah Ilene Strand. 2008. “Using Qualitative Comparative Analysis to Study Causal Order: Comment on Caren and Panofsky (2005).” *Sociological Methods & Research* **36** (4):431-41. DOI: 10.1177/0049124107313903.

Thiem, Alrik. 2013. “Clearly Crisp, and Not Fuzzy: A Reassessment of the (Putative) Pitfalls of Multi-Value QCA.” *Field Methods* **25** (2):197-207. DOI: 10.1177/1525822x13478135.

Thiem, Alrik. 2014. “Parameters of Fit and Intermediate Solutions in Multi-Value Qualitative Comparative Analysis.” *Quality & Quantity* **49** (2):657-74. DOI: 10.1007/s11135-014-0015-x.

Thiem, Alrik, and Adrian Dusa. 2012. “Introducing the QCA Package: A Market Analysis and Software Review.” *Qualitative & Multi-Method Research* **10** (2):45-9. Link.

Thiem, Alrik, and Adrian Dusa. 2013a. “Boolean Minimization in Social Science Research: A Review of Current Software for Qualitative Comparative Analysis (QCA).” *Social Science Computer Review* **31** (4):505-21. DOI: 10.1177/0894439313478999.

Thiem, Alrik, and Adrian Dusa. 2013b. “QCA: A Package for Qualitative Comparative Analysis.” *The R Journal* **5** (1):87-97. Link.

Thiem, Alrik, and Adrian Dusa. 2013c. *Qualitative Comparative Analysis with R: A User’s Guide*. New York: Springer. Link.

Thiem, Alrik, and Michael Baumgartner. 2016. “Modeling Causal Irrelevance in Evaluations of Configurational Comparative Methods.” *Sociological Methodology* 46 (1):345-57. DOI: 10.1177/0081175016654736.

`ambiguity` *Analyze the Degree of Ambiguity across Research Design Variations*

Description

This evaluation function computes the degree of ambiguity across variations of a reference research design. It has initially been programmed for Baumgartner and Thiem (2017).

Usage

```
ambiguity(data, outcome = c(""), neg.out = c(FALSE), exo.facs = c(""),
          tuples = c(), incl.cut1 = c(1), incl.cut0 = c(1), sol.type = c("ps"),
          row.dom = c(FALSE), min.dis = c(FALSE))
```

Arguments

`data`	A set of configurational data as processable by the `eQMC` function.
`outcome`	A character vector of outcomes.
`neg.out`	A logical vector specifying whether to negate outcomes.
`exo.facs`	A character vector with the names of the exogenous factors.
`tuples`	A numeric vector of tuples of exogenous factors to be created from `exof.cols`.
`incl.cut1`	The minimum sufficiency inclusion score for an output function value of "1".
`incl.cut0`	The maximum sufficiency inclusion score for an output function value of "0".
`sol.type`	A character vector specifying the solution types to be generated.
`row.dom`	A logical vector imposing row dominance as a constraint on the solution to eliminate dominated inessential prime implicants.
`min.dis`	A logical vector imposing minimal disjunctivity as a constraint on the solution to eliminate models with more prime implicants than the model(s) with the fewest prime implicants.

Details

This evaluation function computes the degree of ambiguity across variations of a reference design by recording the number of models for each design solution. It has initially been programmed for Baumgartner and Thiem (2015).

The argument `data` requires a set of configurational data as processable by the `eQMC` function.

The argument `outcome` is a character vector, specifying the outcome(s) to be analyzed, either in curly-bracket notation (e.g., `O{value}`) if the outcome is from a multivalent (or a bivalent) factor, or in upper-case notation if the outcome is from a bivalent factor (e.g., `O` as a short-cut for `O{1}`). Outcomes from multivalent crisp-set factors always require curly-bracket notation. Outcomes can be single levels of factors not simultaneously passed to `exo.facs`. At least one outcome has to be specified.

The argument `neg.out` requires a logical vector of length one or two, whose values, which must not be duplicated, specify whether to negate the outcomes determined by `outcome`. If an element in `outcome` is a level from a multivalent factor, `neg.out = TRUE` makes the disjunction of all remaining levels the outcome. Possible values for `neg.out` include `FALSE`, `TRUE`, `FALSE, TRUE` and `TRUE, FALSE`.

The argument `exo.facs` is a character vector with the names of the exogenous factors. If omitted, all factors in `data` are used except that/those of the outcome/s given in `outcome`. and `tuples` specifies a numeric vector of tuples of exogenous factors to be created from `exo.facs`.

Minterms with an inclusion score of at least `incl.cut1` are coded positive (`OUT = "1"`), minterms with an inclusion score below `incl.cut1` but with at least `incl.cut0` are coded as a contradiction (`OUT = "C"`), and minterms with an inclusion score below `incl.cut0` are coded negative (`OUT = "0"`). If `inc .cut0` is not explicitly changed, it is set equal to `incl.cut1`.

The argument `sol.type` requires a character vector specifying the solution types to be generated. For example, `c("ps", "cs")` means parsimonious and conservative solution type.

The argument `row.dom` requires a logical vector, and controls whether the principle of row dominance is imposed as a constraint on the solution. An inessential prime implicant P dominates another Q if all configurations covered by Q are also covered by P, but they are not interchangeable (cf. McCluskey 1956, 1425; McCluskey 1965, 164-152). If row dominance is operative, models that contain dominated prime implicants will not be returned.

The argument `min.dis` requires a logical vector, and controls whether the principle of minimal disjunctivity is imposed as a constraint on the solution (McCluskey 1965, 12 -126). If minimal disjunctivity is operative, models that contain more than the number of prime implicants of the model(s) with the fewest prime implicants will not be returned.

Value

A list with the following two main components:

`tuples` A list of all tuples of exogenous factors of the respective size taken from all factors given in `exo.facs`.

`n.models` A list of matrices giving the number of models in each solution for each design. The coding of labels has the following structure: `0.1234`, where `0` is the outcome, `1` specifies the value at the respective index of the argument `neg.out`, `2` the value at the respective common index of the arguments `incl.cut1` and `incl.cut0`, `3` the value at the respective index of the argument `sol.type`, and `4` the value at the respective common index of the arguments `row.dom` and `min.dis`.

Contributors (alphabetical)

Thiem, Alrik : development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Baumgartner, Michael, and Alrik Thiem. 2017. "Model Ambiguities in Configurational Comparative Research." *Sociological Methods & Research* **46** (4):954-87. DOI: 10.1177/0049124115610351.

McCluskey, Edward J. 1956. "Minimization of Boolean Functions." *Bell Systems Technical Journal* **35** (6):1417-44. DOI: 10.1002/j.1538-7305.1956.tb03835.x.

McCluskey, Edward J. 1965. *Introduction to the Theory of Switching Circuits*. Princeton: Princeton University Press.

See Also

eQMC

Examples

```
## Not run:
# load dataset
data(d.tumorscreen)

# designs: outcomes HPF and LPF; all 3 to 5-tuples of exogenous factors
designs <- ambiguity(d.tumorscreen, outcome = c("HPF", "LPF"),
  neg.out = c(FALSE, TRUE), tuples = 3:5)

# share of solutions with ambiguities
mapply(function (x) round(colSums((x > 1)) / nrow(x), 2), designs$n.models)

## End(Not run)
```

---

calibrate *Calibrate Raw Data into Configurational Data*

Description

This function generates configurational data from raw data (base variables) and some specified threshold(s). The calibration of bivalent fuzzy-set factors is possible for positive and negative end-point and mid-point concepts, using the method of transformational assignment.

Usage

```
calibrate(x, type = "crisp", thresholds = NA, include = TRUE,
          logistic = FALSE, idm = 0.95, ecdf = FALSE, p = 1, q = 1)
```

Arguments

x	An interval or ratio-scaled base variable.
type	The calibration type, either "crisp" or "fuzzy".
thresholds	A vector of thresholds.

include	Logical, include threshold(s) (type = "crisp" only).
logistic	Calibrate to fuzzy-set variable using the logistic function.
idm	The set inclusion degree of membership for the logistic function.
ecdf	Calibrate to fuzzy-set variable using the empirical cumulative distribution function of the base variable.
p	Parameter: if $p > 1$ concentration, if $0 < p < 1$ dilation below crossover.
q	Parameter: if $q > 1$ dilation, if $0 < q < 1$ concentration above crossover.

Details

Calibration is the process by which configurational data is produced, that is, by which set membership scores are assigned to cases. With interval and ratio-scaled base variables, calibration can be based on transformational assignments using (piecewise-defined) membership functions.

For type = "crisp", one threshold produces a factor with two levels: 0 and 1. More thresholds produce factors with multiple levels. For example, two thresholds produce three levels: 0, 1 and 2.

For type = "fuzzy", this function can generate bivalent fuzzy-set variables by linear, s-shaped, inverted s-shaped and logistic transformation for end-point concepts. It can generate bivalent fuzzy-set variables by trapezoidal, triangular and bell-shaped transformation for mid-point concepts (Bojadziev and Bojadziev 2007; Clark *et al.* 2008; Thiem 2014; Thiem and Dusa 2013).

For calibrating bivalent fuzzy-set variables based on end-point concepts, thresholds should be specified as a numeric vector c(thEX, thCR, thIN), where thEX is the threshold for full exclusion, thCR the threshold for the crossover, and thIN the threshold for full inclusion.

If thEX < thCR < thIN, then the membership function is increasing from thEX to thIN. If thIN < thCR < thEX, then the membership function is decreasing from thIN to thEX.

For calibrating bivalent fuzzy-set variables based on mid-point concepts, thresholds should be specified as a numeric vector c(thEX1, thCR1, thIN1, thIN2, thCR2, thEX2), where thEX1 is the first (left) threshold for full exclusion, thCR1 the first (left) threshold for the crossover, thIN1 the first (left) threshold for full inclusion, thIN2 the second (right) threshold for full inclusion, thCR2 the second (right) threshold for crossover, and thEX2 the second (right) threshold for full exclusion.

If thEX1 < thCR1 < thIN1 ≤ thIN2 < thCR2 < thEX2, then the membership function is first increasing from thEX1 to thIN1, then flat between thIN1 and thIN2, and finally decreasing from thIN2 to thEX2. In contrast, if thIN1 < thCR1 < thEX1 ≤ thEX2 < thCR2 < thIN2, then the membership function is first decreasing from thIN1 to thEX1, then flat between thEX1 and thEX2, and finally increasing from thEX2 to thIN2.

The parameters p and q control the degree of concentration and dilation. They should be left at their default values unless good reasons for changing them exist.

If logistic = TRUE, the argument idm specifies the inclusion degree of membership.

If ecdf = TRUE, calibration is based on the empirical cumulative distribution function of x.

Value

A numeric vector of set membership scores between 0 and 1 for bivalent crisp-set factors and bivalent fuzzy-set variables, or a numeric vector of levels for multivalent crisp-set factors (beginning with 0 at increments of 1).

Contributors

Dusa, Adrian	: programming
Thiem, Alrik	: development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Bojadziev, George, and Maria Bojadziev. 2007. *Fuzzy Logic for Business, Finance, and Management*. 2nd ed. Hackensack, NJ: World Scientific. Link.

Clark, Terry D., Jennifer M. Larson, John N. Mordeson, Joshua D. Potter, and Mark J. Wierman. 2008. *Applying Fuzzy Mathematics to Formal Models in Comparative Politics*. Berlin: Springer. Link.

Thiem, Alrik. 2014. "Membership Function Sensitivity of Descriptive Statistics in Fuzzy-Set Relations." *International Journal of Social Research Methodology* **17** (6):625-42. DOI: 10.1080/13645579.2013.806118.

Thiem, Alrik, and Adrian Dusa. 2013. *Qualitative Comparative Analysis with R: A User's Guide*. New York: Springer. Link.

Examples

```
# base variable; random draw from standard normal distribution
set.seed(30)
x <- rnorm(30)

# calibration thresholds
th <- quantile(x, seq(from = 0.05, to = 0.95, length = 6))

# calibration of bivalent crisp-set factor
calibrate(x, thresholds = th[3])

# calibration of trivalent crisp-set factor
calibrate(x, thresholds = c(th[2], th[4]))

# fuzzy-set calibration
# 1. positive end-point concept, linear
# 2. positive and corresponding negative end-point concept, logistic
# 3. positive end-point concept, ECDF
# 4. negative end-point concept, s-shaped (quadratic)
# 5. negative end-point concept, inverted s-shaped (root)
# 6. positive mid-point concept, triangular
# 7. positive mid-point concept, trapezoidal
# 8. negative mid-point concept, bell-shaped

yl <- "Set Membership"
xl <- "Base Variable Value"
```

```
par(mfrow = c(2,4), cex.main = 1)

plot(x, calibrate(x, type = "fuzzy", thresholds = c(th[1], (th[3]+th[4])/2,
   th[6])), xlab = xl, ylab = yl,
   main = "1. positive end-point concept,\nlinear")

plot(x, calibrate(x, type = "fuzzy", thresholds = c(th[1], (th[3]+th[4])/2,
  th[6]), logistic = TRUE, idm = 0.99), xlab = xl, ylab = yl,
  main = "2. positive and corresponding negative\nend-point concept, logistic")
  points(x, calibrate(x, type = "fuzzy", thresholds = c(th[6], (th[3]+th[4])/2,
    th[1]), logistic = TRUE, idm = 0.99))

plot(x, calibrate(x, type = "fuzzy", thresholds = c(th[1], (th[3]+th[4])/2,
  th[6]), ecdf = TRUE), xlab = xl, ylab = yl,
  main = "3. positive end-point concept,\nECDF")

plot(x, calibrate(x, type = "fuzzy", thresholds = c(th[6], (th[3]+th[4])/2,
  th[1]), p = 2, q = 2), xlab = xl, ylab = yl,
  main = "4. negative end-point concept,\ns-shaped (quadratic)")

plot(x, calibrate(x, type = "fuzzy", thresholds = c(th[6], (th[3]+th[4])/2,
  th[1]), p = 0.5, q = 0.5), xlab = xl, ylab = yl,
  main = "5. negative end-point concept,\ninverted s-shaped (root)")

plot(x, calibrate(x, type = "fuzzy", thresholds = th[c(1,2,3,3,4,5)]),
  xlab = xl, ylab = yl, main = "6. positive mid-point concept,\ntriangular")

plot(x, calibrate(x, type = "fuzzy", thresholds = th[c(1,2,3,4,5,6)]),
  xlab = xl, ylab = yl, main = "7. positive mid-point concept,\ntrapezoidal")

plot(x, calibrate(x, type = "fuzzy", thresholds = th[c(3,2,1,5,4,3)],
  p = 3, q = 3), xlab = xl, ylab = yl,
  main = "8. negative mid-point concept,\nbell-shaped")
```

`d.biodiversity` *Linkages and Collective Action in Biodiversity Conservation*

Description

This dataset is from Basurto (2013), who analyzes the determinants of the emergence and endurance of autonomy among local institutions for biodiversity conservation in Costa Rica using fsQCA.

Usage

```
data(d.biodiversity)
```

Format

This data frame contains 30 rows (cases) and the following 9 columns (factors):

[, 1]	**AU**	endogenous factor:	local autonomy ("1" always, "0" never)
[, 2]	**EM**	exogenous factor:	local communal involvement through direct employment ("1" 100 percent, "0" 0 percent)
[, 3]	**SP**	exogenous factor:	local direct spending ("1" always, "0" never)
[, 4]	**CO**	exogenous factor:	co-management with local or regional stakeholders ("1" present, "0" absent)
[, 5]	**CI**	exogenous factor:	degree of influence of national civil service policies ("1" 100 percent civil service employees, "0" 0 percent)
[, 6]	**PO**	exogenous factor:	national participation in policy-making ("1" perceived programme influence, "0" no perceived influence)
[, 7]	**RE**	exogenous factor:	research-oriented partnerships ("1" many, "0" few)
[, 8]	**CN**	exogenous factor:	conservation-oriented partnerships ("1" many, "0" few)
[, 9]	**DE**	exogenous factor:	direct support by development organizations ("1" always, "0" never)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Basurto, Xavier. 2013. "Linking Multi-Level Governance to Local Common-Pool Resource Theory using Fuzzy-Set Qualitative Comparative Analysis: Insights from Twenty Years of Biodiversity Conservation in Costa Rica." *Global Environmental Change* **23** (3):573-87. DOI: 10.1016/j.gloenvcha.2013.02.011.

`d.education` *Impact of New Public Management Instruments on PhD Education*

Description

This dataset is from Schneider and Sadowski (2010), who analyze the determinants of PhD placement success for 14 economics departments using csQCA.

Usage

```
data(d.education)
```

Format

This data frame contains 14 rows (cases) and the following 7 columns (factors):

[, 1]	**NPM1**	exogenous factor	:	local competition	("1" used, "0" not used)
[, 2]	**NPM2**	exogenous factor	:	national competition	("1" used, "0" not used)
[, 3]	**NPM3**	exogenous factor	:	transparency	("1" used, "0" not used)
[, 4]	**NPM4**	exogenous factor	:	university regulations	("1" used, "0" not used)
[, 5]	**NPM5**	exogenous factor	:	target agreements	("1" used, "0" not used)
[, 6]	**NPM6**	exogenous factor	:	state regulations	("1" used, "0" not used)
[, 7]	**O**	endogenous factor	:	placement success	("1" yes, "0" no)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Schneider, Peter, and Dieter Sadowski. 2010. "The Impact of New Public Management Instruments on PhD Education." *Higher Education* **59** (5):543-65. DOI: 10.1007/s10734-009-9264-3.

`d.graduate` *Graduate Student Union Recognition at Research Universities*

Description

This dataset is originally from Caren and Panofsky (2005), who analyze the determinants of unionization attempts by graduate student workers at research universities using tQCA. Their study has been replicated and corrected by Ragin and Strand (2008).

Usage

```
data(d.graduate)
```

Format

This data frame contains 17 rows (cases) and the following 6 columns (factors):

[, 1]	**P**	exogenous factor	:	public university	("1" yes, "0" no)
[, 2]	**E**	exogenous factor	:	support of elite allies	("1" yes, "0" no)
[, 3]	**A**	exogenous factor	:	national union affiliation	("1" yes, "0" no)

[, 4]	**S**	exogenous factor	:	a strike or strike threat	("1" yes, "0" no)
[, 5]	**EBA**	exogenous factor	:	**E** present before **A**	("1" yes, "0" no, "-" don't care)
[, 6]	**REC**	endogenous factor	:	union recognition	("1" yes, "0" no)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Caren, Neal, and Aaron Panofsky. 2005. "TQCA: A Technique for Adding Temporality to Qualitative Comparative Analysis." *Sociological Methods & Research* **34** (2):147-72. DOI: 10.1177/0049124105277197.

Ragin, Charles C., and Sarah Ilene Strand. 2008. "Using Qualitative Comparative Analysis to Study Causal Order: Comment on Caren and Panofsky (2005)." *Sociological Methods & Research* **36** (4):431-41. DOI: 10.1177/0049124107313903.

---

d.health — *Trends in Narrowing Health Inequalities in England*

Description

This dataset is from Blackman, Wistow and Byrne (2011), who analyze the determinants of varying progress with tackling health inequalities with respect to cancers and cardiovascular disease among a group of 27 local authority areas in England.

Usage

```
data(d.health)
```

Format

This data frame contains 27 rows (cases) and the following 18 columns (factors):

[, 1]	**CAN**	endogenous factor:	area gap for deaths before age 75 from cancers ("1" narrowing, "0" not narrowing)
[, 2]	**BC**	exogenous factor:	assessments of commissioning ("1" basic, "0" not basic)
[, 3]	**SP**	exogenous factor:	assessments of strategic partnership working ("1" less than good, "0" at least good)
[, 4]	**PH**	exogenous factor:	assessments of public health workforce planning

			("1" less than good, "0" at least good)
[, 5]	**PR**	exogenous factor:	frequency of progress reviews ("1" less frequent, "0" more frequent)
[, 6]	**CH**	exogenous factor:	working culture of individual commitment and champions ("1" yes, "0" no)
[, 7]	**AS**	exogenous factor:	organisational culture ("1" aspirational, "0" comfortable or complacent)
[, 8]	**LI**	exogenous factor:	index of multiple deprivation ("1" lower, "0" higher)
[, 9]	**HS**	exogenous factor:	spend per head on cancer programmes ("1" higher, "0" lower)
[, 10]	**LC**	exogenous factor:	crime rate ("1" lower, "0" higher)
[, 11]	**TS**	exogenous factor:	primary care trust performance rating ("1" higher, "0" lower)
[, 12]	**CVD**	endogenous factor:	area gap for deaths before age 75 from cardiovascular disease ("1" narrowing, "0" not narrowing)
[, 13]	**SC**	exogenous factor:	smoking cessation services ("1" better than basic, "0" basic)
[, 14]	**PC**	exogenous factor:	primary care services ("1" better than basic, "0" basic)
[, 15]	**MP**	exogenous factor:	a few major programmes ("1" yes, "0" no)
[, 16]	**GL**	exogenous factor:	leadership ("1" good or excellent, "0" less than good)
[, 17]	**BA**	exogenous factor:	budget allocation relative to target ("1" higher, "0" lower)
[, 18]	**IM**	exogenous factor:	internal migration ("1" lower, "0" higher)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Blackman, Tim, Jonathan Wistow, and David Byrne. 2011. “A Qualitative Comparative Analysis of Factors Associated with Trends in Narrowing Health Inequalities in England.” *Social Science & Medicine* **72** (12):1965-74. DOI: 10.1016/j.socscimed.2011.04.003.

d.homeless *Outcomes of Homeless Social Movement Organizations*

Description

This dataset is from Cress and Snow (2000), who analyze the determinants of the outcomes attained by homeless social movement organizations using csQCA.

Usage

```
data(d.homeless)
```

Format

This data frame contains 15 rows (cases) and the following 10 columns (factors):

[, 1]	**VI**	exogenous factor	:	viability	("1" present, "0" absent)
[, 2]	**DT**	exogenous factor	:	disruptive tactics	("1" present, "0" absent)
[, 3]	**SA**	exogenous factor	:	sympathetic allies	("1" present, "0" absent)
[, 4]	**CS**	exogenous factor	:	city support	("1" present, "0" absent)
[, 5]	**DF**	exogenous factor	:	diagnostic frame	("1" present, "0" absent)
[, 6]	**PF**	exogenous factor	:	prognostic frame	("1" present, "0" absent)
[, 7]	**REP**	endogenous factor	:	representation	("1" present, "0" absent)
[, 8]	**RES**	endogenous factor	:	resources	("1" present, "0" absent)
[, 9]	**RIG**	endogenous factor	:	rights	("1" present, "0" absent)
[, 10]	**REL**	endogenous factor	:	relief	("1" present, "0" absent)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Cress, Daniel M., and David A. Snow. 2000. “The Outcomes of Homeless Mobilization: The Influence of Organization, Disruption, Political Mediation, and Framing.” *American Journal of Sociology* **105** (4):1063-104. Link.

d.jobsecurity *Job Security Regulations in Western Democracies*

Description

This dataset is from Emmenegger (2011), who analyzes the determinants of high job security regulations in Western democracies using fsQCA.

Usage

```
data(d.jobsecurity)
```

Format

This data frame contains 19 rows (cases) and the following 7 columns (factors):

[, 1]	**S**	exogenous factor	:	level of statism	("1" high, "0" not high)
[, 2]	**C**	exogenous factor	:	level of non-market coordination	("1" high, "0" not high)
[, 3]	**L**	exogenous factor	:	level of labour movement strength	("1" high, "0" not high)
[, 4]	**R**	exogenous factor	:	level of Catholicism	("1" high, "0" not high)
[, 5]	**P**	exogenous factor	:	level of religious party strength	("1" high, "0" not high)
[, 6]	**V**	exogenous factor	:	institutional veto points	("1" many, "0" not many)
[, 7]	**JSR**	endogenous factor	:	level of job security regulations	("1" high, "0" not high)

Contributors

Thiem, Alrik: collection, documentation

Note

The row names are the official International Organization for Standardization (ISO) country code elements as specified in ISO 3166-1-alpha-2.

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Emmenegger, Patrick. 2011. "Job Security Regulations in Western Democracies: A Fuzzy Set Analysis." *European Journal of Political Research* **50** (3):336-64. DOI: 10.1111/j.1475-6765.2010.01933.x.

d.napoleon *Reception of the Code Napoleon in Germany*

Description

This dataset is from Arvind and Stirton (2010), who analyze the reception of the Code Napoleon in Germany using fsQCA.

Usage

```
data(d.napoleon)
```

Format

This data frame contains 14 rows (cases) and the following 8 columns (factors):

[, 1]	**D**	exogenous factor	:	legal system	("1" heterogenous, "0" homogenous)
[, 2]	**C**	exogenous factor	:	territory	("1" ruled by France, "0" ruled by enemy)
[, 3]	**I**	exogenous factor	:	state institutions	("1" strong, "0" none)
[, 4]	**F**	exogenous factor	:	economy	("1" seigneural-feudal, "0" proto-industrial)
[, 5]	**L**	exogenous factor	:	ideology of state ruler	("1" liberal, "0" conservative)
[, 6]	**N**	exogenous factor	:	nativist tendencies	("1" yes, "0" no)
[, 7]	**A**	exogenous factor	:	sentiments towards France	("1" very negative, "0" very positive)
[, 8]	**O**	endogenous factor	:	adoption of Code Napoleon	("1" yes, "0" no)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Arvind, Thiruvallore T., and Lindsay Stirton. 2010. "Explaining the Reception of the Code Napoleon in Germany: A Fuzzy-Set Qualitative Comparative Analysis." *Legal Studies* **30** (1):1-29. DOI: 10.1111/j.1748-121X.2009.00150.x.

`d.partybans` *Party Bans Sub-Saharan Africa*

Description

This dataset is from Hartmann and Kemmerzell (2010), who analyze the determinants of the introduction of party ban provisions and their actual implementation in sub-Saharan Africa using mvQCA.

Usage

```
data(d.partybans)
```

Format

This data frame contains 48 rows (cases) and the following 7 columns (factors):

[, 1]	**C**	exogenous factor:	colonial tradition ("2" British, "1" French, "0" other)
[, 2]	**F**	exogenous factor:	former regime type competition ("2" no, "1" limited, "0" multi-party)
[, 3]	**T**	exogenous factor:	mode of transition ("2" managed, "1" pacted, "0" democracy before 1990)
[, 4]	**R**	exogenous factor:	regime type ("2" authoritarian, "1" liberalizing, "0" democratic)
[, 5]	**V**	exogenous factor:	ethnic violence ("1" yes, "0" no)
[, 6]	**PB**	endogenous factor:	party ban provisions introduced ("1" yes, "0" no)
[, 7]	**PBI**	endogenous factor:	party bans implemented ("1" yes, "0" no)

Contributors

Thiem, Alrik: collection, documentation

Note

The row names are the official International Organization for Standardization (ISO) country code elements as specified in ISO 3166-1-alpha-2.

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Hartmann, Christof, and Joerg Kemmerzell. 2010. "Understanding Variations in Party Bans in Africa." *Democratization* **17** (4):642-65. DOI: 10.1080/13510347.2010.491189.

`d.represent` *Women's Representation in Western Democratic Parliaments*

Description

This dataset is from Krook (2010), who analyzes the determinants of high women's representation in Western-democratic parliaments using csQCA.

Usage

```
data(d.represent)
```

Format

This data frame contains 22 rows (cases) and the following 6 columns (factors):

[, 1]	**ES**	exogenous factor:	PR electoral system ("1" yes, "0" no)
[, 2]	**QU**	exogenous factor:	quota for women ("1" yes, "0" no)
[, 3]	**WS**	exogenous factor:	social-democratic welfare system ("1" yes, "0" no)
[, 4]	**WM**	exogenous factor:	autonomous women's movement ("1" yes, "0" no)
[, 5]	**LP**	exogenous factor:	seats held by left-libertarian parties ("1" at least 7 percent, "0" less than 7 percent)
[, 6]	**WNP**	endogenous factor:	women in single/lower house of parliament ("1" at least 30 percent, "0" less than 30 percent)

Contributors

Thiem, Alrik: collection, documentation

Note

The row names are the official International Organization for Standardization (ISO) country code elements as specified in ISO 3166-1-alpha-2.

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Krook, Mona Lena. 2010. "Women's Representation in Parliament: A Qualitative Comparative Analysis." *Political Studies* **58** (5):886-908. DOI: 10.1111/j.1467-9248.2010.00833.x.

`d.socialsecurity` *The Emergence of the Social Security State*

Description

This dataset is from Hicks, Misra and Ng (1995), who analyze the emergence of social security programs in 15 industrializing countries during the period 1880-1930 using csQCA.

Usage

```
data(d.socialsecurity)
```

Format

This data frame contains 30 rows (cases) and the following 6 columns (factors):

[, 1]	**LG**	exogenous factor	:	liberal government	("1" present, "0" absent)
[, 2]	**CG**	exogenous factor	:	Catholic government	("1" present, "0" absent)
[, 3]	**PS**	exogenous factor	:	patriarchal state	("1" present, "0" absent)
[, 4]	**UD**	exogenous factor	:	unitary democracy	("1" present, "0" absent)
[, 5]	**WM**	exogenous factor	:	working-class mobilization	("1" present, "0" absent)
[, 6]	**CO**	endogenous factor	:	consolidation	("1" present, "0" absent)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Hicks, Alexander, Joya Misra, and Tang N. Ng. 1995. "The Programmatic Emergence of the Social Security State." *American Sociological Review* **60** (3):329-49. Link.

d.stakeholder *Decoupling in Response to Stakeholder Pressures*

Description

This dataset is from Crilly, Zollo and Hansen (2012), who analyze the determinants of firms' responses to institutional pressures using fsQCA.

Usage

```
data(d.stakeholder)
```

Format

This data frame contains 17 rows (cases) and the following 5 columns (factors):

[, 1]	**PA**	exogenous factor	:	potential for asymmetry	("1" high, "0" low)
[, 2]	**SC**	exogenous factor	:	stakeholder consensus	("1" high, "0" low)
[, 3]	**OI**	exogenous factor	:	organizational interest	("1" high, "0" low)
[, 4]	**MC**	exogenous factor	:	managerial consensus	("1" high, "0" low)
[, 5]	**SA**	endogenous factor	:	substantive action	("1" high, "0" low)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Crilly, Donal, Maurizio Zollo, and Morten T. Hansen. 2012. "Faking It or Muddling Through? Understanding Decoupling in Response to Stakeholder Pressures." *Academy of Management Journal* **55** (6):1429-48. DOI: 10.5465/amj.2010.0697.

d.transport *Determinants of High Transport Project Acceptance*

Description

This dataset is from Sager and Andereggen (2012), who analyze the determinants of high transport project acceptance in Switzerland using mvQCA.

Usage

```
data(d.transport)
```

Format

This data frame contains 21 rows (cases) and the following 10 columns (factors):

[, 1]	**FED**	exogenous factor:	federal level ("2" federal, "1" cantonal, "0" municipal)
[, 2]	**FIN**	exogenous factor:	financial situation ("1" positive, "0" negative)
[, 3]	**URB**	exogenous factor:	sociostructural project location ("1" urban, "0" rural)
[, 4]	**GER**	exogenous factor:	cultural project location ("1" German-speaking, "0" French-speaking)
[, 5]	**HIS**	exogenous factor:	prior history ("1" yes, "0" no)
[, 6]	**COO**	exogenous factor:	planning coordination ("1" strong, "0" not strong)
[, 7]	**PRO**	exogenous factor:	administrative professionalization ("1" high, "0" not high)
[, 8]	**DIS**	exogenous factor:	administration's discretion ("1" broad, "0" not broad)
[, 9]	**EXP**	exogenous factor:	influence of external experts ("1" great, "0" not great)
[, 10]	**ACC**	endogenous factor:	project acceptance ("1" high, "0" not high)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Sager, Fritz, and Celine Andereggen. 2012. "Dealing With Complex Causality in Realist Synthesis: The Promise of Qualitative Comparative Analysis." *American Journal of Evaluation* **33** (1):60-78. DOI: 10.1177/1098214011411574.

`d.tumorscreen` *Comparing Universal Lynch Syndrome Tumor-Screening Programs*

Description

This dataset is from Cragun *et al.* (2014), who analyze the association between different universal tumor screening procedures and certain levels of patient follow-through with germ-line testing for Lynch Syndrome after a screen-positive result using csQCA.

Usage

```
data(d.tumorscreen)
```

Format

This data frame contains 15 rows (cases) and the following 8 columns (factors):

[, 1]	**HPF**	endogenous factor	:	high patient follow-through	("1" yes, "0" no)
[, 2]	**LPF**	endogenous factor	:	low patient follow-through	("1" yes, "0" no)
[, 3]	**CA**	exogenous factor	:	challenges to adoption at least as high as facilitators	("1" yes, "0" no)
[, 4]	**AR**	exogenous factor	:	automatic reflex test of screen-positive tumors	("1" yes, "0" no)
[, 5]	**RR**	exogenous factor	:	genetic counselor receives positive screen results	("1" yes, "0" no)
[, 6]	**DR**	exogenous factor	:	genetic counselor discloses screening result to patient	("1" yes, "0" no)
[, 7]	**DC**	exogenous factor	:	difficulty in contacting patients	("1" yes, "0" no)
[, 8]	**PR**	exogenous factor	:	need for physician referral is a barrier	("1" yes, "0" no)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Cragun, Deborah, Rita D. DeBate, Susan T. Vadaparampil, Julie Baldwin, Heather Hampel, and Tuya Pal. 2014. "Comparing Universal Lynch Syndrome Tumor-Screening Programs to Evaluate Associations between Implementation Strategies and Patient Follow-Through." *Genetics in Medicine* **16** (10):773-82. DOI: 10.1038/gim.2014.31.

`d.urban` *Explaining Urban Regimes*

Description

This dataset is from Kilburn (2004), who analyzes the influence of city context on urban regimes across 14 cities in the United States using csQCA.

Usage

```
data(d.urban)
```

Format

This data frame contains 14 rows (cases) and the following 6 columns (factors):

[, 1]	**MLC**	exogenous factor	:	mobility of local capital	("1" high, "0" not high)
[, 2]	**FRB**	exogenous factor	:	fiscal resource base	("1" large, "0" not large)
[, 3]	**CP**	exogenous factor	:	civic participation	("1" high, "0" not high)
[, 4]	**WSR**	exogenous factor	:	ward-style representation	("1" high, "0" not high)
[, 5]	**CS**	exogenous factor	:	city size	("1" large, "0" not large)
[, 6]	**RT**	endogenous factor	:	regime type	("1" progressive, "0" developmental/caretaker)

Contributors

Thiem, Alrik: collection, documentation

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

Source

Kilburn, H. Whitt. 2004. “Explaining U.S. Urban Regimes.” *Urban Affairs Review* **39** (5):633-51. DOI: 10.1177/1078087403262861.

`DeMorgan` *Negate Boolean Expressions using De Morgan’s Laws*

Description

This function negates simple or complex Boolean expressions using the two De Morgan Laws.

Usage

```
DeMorgan(expression, and.split = "", use.tilde = FALSE)

is.DeMorgan(x)
```

Arguments

`expression`	A string representing a Boolean expression or a solution object of class 'qca'.
`and.split`	The AND-operator (if any).
`use.tilde`	Logical, use '~' for negation with bivalent variables.
`x`	An object of class 'DeMorgan'.

Details

The two De Morgan laws posit that the negation of a disjunction is the conjunction of its separate negations, and the negation of a conjunction is the disjunction of its separate negations (Hohn 1966, p.80).

The argument `expression` can be any complex string representing a Boolean expression of disjunctions and conjunctions, or a solution object of class 'qca' (objects returned by the '`eQMC`' function).

Value

A list of solutions with their negations as components if `expression` is an object of class 'qca', or simply a list with the following components if `expression` is a string:

`initial`	The initial expression.
`negated`	The negation of the initial expression.

Contributors

Dusa, Adrian	: development, programming, testing
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Hohn, Franz E. 1966. *Applied Boolean Algebra: An Elementary Introduction*. 2nd ed. New York: Macmillan.

Ragin, Charles C. 1987. *The Comparative Method: Moving beyond Qualitative and Quantitative Strategies*. Berkeley: University of California Press.

See Also

eQMC

Examples

```
# example from Ragin (1987, p.99)
DeMorgan("AC + B~C")

# with different AND-operators
DeMorgan("A*C + B*~C", and.split = "*")
DeMorgan("A&C + B&~C", and.split = "&")

# use solution object of class 'qca' returned by 'eQMC' function,
# even with multiple models
data(d.represent)
KRO.ps <- eQMC(d.represent, outcome = "WNP")
DeMorgan(KRO.ps)
```

---

eQMC *Minimization with Enhanced Quine-McCluskey Algorithm*

Description

This function performs the minimization. Although it is called '*e*QMC', the implemented algorithm is different from the classical Quine-McCluskey (QMC) algorithm. Instead of QMC's approach of using positive minterms and remainders to perform minimization, *e*QMC uses positive and negative minterms, but no remainders. See Dusa and Thiem (2015) and Thiem (2015) for more details.

Usage

```
eQMC(data, outcome = c(""), neg.out = FALSE, exo.facs = c(""),
     relation = "suf", n.cut = 1, incl.cut1 = 1, incl.cut0 = 1,
     minimize = c("1"), sol.type = "ps", row.dom = FALSE,
     min.dis = FALSE, omit = c(), dir.exp = c(), details = FALSE,
     show.cases = FALSE, inf.test = c(""), use.tilde = FALSE,
     use.letters = FALSE, ...)

is.qca(x)
```

Arguments

`data`	A truth table object or a set of configurational data (of class 'matrix' or 'data.frame').
`outcome`	A character vector of outcomes.
`neg.out`	Logical, use negation of `outcome` (ignored if `data` is a truth table object).
`exo.facs`	A character vector with the names of the exogenous factors.
`relation`	The required relation of a model antecendent to the `outcome`; either `"suf"` (only sufficiency required) or `"sufnec"` (both sufficiency and necessity required).

`n.cut`	The minimum number of cases with set membership score above 0.5 for an output function value of "0", "1" or "C"; an integer between 1 and the maximum number of cases for all non-remainder minterms.
`incl.cut1`	The minimum sufficiency inclusion score for an output function value of "1".
`incl.cut0`	The maximum sufficiency inclusion score for an output function value of "0".
`minimize`	A vector of output function values for which a solution is sought.
`sol.type`	A character scalar specifying the QCA solution type that should be applied; either "ps" (parsimonious solution), "ps+" (parsimonious solution including both positive and contradiction minterms), "cs" (conservative solution) or "cs+" (conservative solution including both positive and contradiction minterms). Note that only "ps" and "ps+" generate correct solutions.
`row.dom`	Logical, impose row dominance as a constraint on the solution to eliminate dominated inessential prime implicants. For causal data analysis, this argument must be set to `FALSE`.
`min.dis`	Logical, impose minimal disjunctivity as a constraint on the solution to eliminate models with more prime implicants than the model(s) with the fewest prime implicants. For causal data analysis, this argument must be set to `FALSE`.
`omit`	A vector of minterm index values or a matrix of minterms to be omitted from minimization.
`dir.exp`	A vector of directional expectations for deriving intermediate solutions; can only be used in conjunction with `sol.type = "ps"` or `sol.type = "ps+"`. Note that neither conservative nor intermediate solutions produce correct solutions. This argument is only retained for purposes of method evaluation.
`details`	Logical, present solution details (inclusion, raw coverage and unique coverage scores).
`show.cases`	Logical, also print case names as part of a solution's details; `details` must be set to `TRUE` (do not use this option with many cases and/or long case names).
`inf.test`	A vector of length two specifying the inference-statistical test to be performed (currently only `"binom"`) and the critical significance level.
`use.tilde`	Logical, use tilde operator ("~") for negation with bivalent (crisp-set and fuzzy-set) factors.
`use.letters`	Logical, use single letters (in alphabetical order) instead of original variable names.
`...`	Other arguments.
`x`	An object of class 'qca'.

Details

The argument `data` can be a truth table object (an object of class 'tt' returned by the `truthTable` function) or a suitable data set. Suitable data sets have the following structure: values of 0 and 1 for bivalent crisp-set factors, values between 0 and 1 for bivalent fuzzy-set factors, and values beginning with 0 at increments of 1 for multivalent crisp-set factors. The placeholders "-" and "dc" indicate "don't cares" in auxiliary factors that specify temporal order between other substantive factors in tQCA. These values lead to the exclusion of the auxiliary factor from the computation of parameters of fit.

The argument `outcome` specifies the outcome to be analyzed, either in curly-bracket notation (e.g., `O{value}`) if the outcome is from a multivalent (or a bivalent) factor, or in upper-case notation if the outcome is from a bivalent factor (e.g., `O` as a short-cut for `O{1}`). Outcomes from multivalent crisp-set factors always require curly-bracket notation. Outcomes can be single levels of factors not simultaneously passed to `exo.facs`, or levels from any subset of the factors specified in `exo.facs` if `data` is not a truth table object. At least one outcome has to be specified.

If multiple outcomes are specified, their factors must also be specified in `exo.facs`. In this case, solution details will not be printed by default (see the example on mimicking Coincidence Analysis below).

The logical argument `neg.out` controls whether `outcome` is to be analyzed or its negation. If `outcome` is a level from a multivalent factor, `neg.out = TRUE` makes the disjunction of all remaining levels the outcome.

The argument `exo.facs` specifies the exogenous factors. If omitted, all factors in `data` are used except that of the `outcome`. With multiple outcomes, all factors in `data` are used. Please note that computation times may increase significantly beyond 17 exogenous factors, and that the computation of a solution may not be possible at all depending on end-user machine constraints.

The argument `relation` specifies the relation between the antecedent of a model and the outcome. It accepts either the value `"suf"` or `"sufnec"`. If `relation = "suf"` (default), only sufficiency is used as a criterion in identifying a model. If `relation = "sufnec"`, models must be sufficient and necessary for the outcome to be identified. The argument `incl.cut1` then acts as the cut-off for the sufficiency inclusion of a minterm as well as the necessity inclusion of the final model(s).

Minterms that contain fewer than `n.cut` cases with membership scores above 0.5 are coded as remainders (`OUT = "?"`). If the number of such cases is at least `n.cut`, minterms with an inclusion score of at least `incl.cut1` are coded positive (`OUT = "1"`), minterms with an inclusion score below `incl.cut1` but with at least `incl.cut0` are coded as a contradiction (`OUT = "C"`), and minterms with an inclusion score below `incl.cut0` are coded negative (`OUT = "0"`). If `incl.cut0` is not explicitly changed, it is set equal to `incl.cut1`.

The argument `minimize` specifies a vector of suitable values of the output function for which a solution is sought. Vectors of such values are `"1"` (default; positive minterms), `"C"` (contradictions), `"0"` (negative minterms), `c("1", "C")` and `c("0", "C")`, but not `c("1", "0")` and `c("1", "0", "C")`. Note that for `"0"`, `"C"` and `c("0", "C")`, the respective minterms will be processed but no solution details will be printed. Also note that `minimize = "0"` is not the same as using `neg.out = TRUE`.

The argument `sol.type` specifies the QCA solution type that should be generated. It accepts either `"ps"` (default, parsimonious solution), `"ps+"` (parsimonious solution including both positive minterms and contradictions), `"cs"` (conservative solution) or `"cs+"` (conservative solution including both positive minterms and contradictions). As only the parsimonious search strategy generates methodologically correct solutions (Baumgartner and Thiem 2017a), `sol.type` should not normally be changed to generate conservative or intermediate solutions.

The logical argument `row.dom` controls whether the principle of row dominance is imposed as a constraint on the solution. An inessential prime implicant P dominates another Q if all configurations covered by Q are also covered by P, but they are not interchangeable (cf. McCluskey 1956, 1425; McCluskey 1965, 164-152). If row dominance is operative, models that contain dominated prime implicants will not be returned. For purposes of causal data analysis, `row.dom` must be set to `FALSE`.

The logical argument `min.dis` controls whether the principle of minimal disjunctivity is imposed as a constraint on the solution (McCluskey 1965, 123-126). If minimal disjunctivity is operative, models that contain more than the number of prime implicants of the model(s) with the fewest prime implicants will not be returned. *For purposes of causal data analysis, both* `row.dom` *and* `min.dis` *must be set to* `FALSE` (Baumgartner and Thiem 2017b; Thiem 2014b).

The argument `omit` can be used to omit minterms from the minimization process *ex ante*. It accepts a vector of row numbers from the truth table or a matrix of minterms of the same order as passed to the `truthTable` function (if the argument `data` is a truth table object) or as specified in the argument `exo.facs`.

Neither the conservative nor the intermediate search strategy of QCA produce correct solutions (Baumgartner and Thiem 2017a). The `dir.exp` argument is retained only for purposes of method evaluation in relation to intermediate solutions. It specifies directional expectations for separating easy from difficult counterfactuals in simplifying assumptions. For bivalent crisp and fuzzy-set factors, expectations should be specified as a vector of the same length and the same order of condition variables as provided in `exo.facs`. For bivalent factors, a value of either "0" or "1" indicates that the corresponding factor is expected to contribute to a positive output function value, while a dash, "-", indicates that one or the other level of the corresponding factor does so. For multivalent factors, multiple levels have to be enclosed by double quotes and separated by a semicolon (see mvQCA example using Hartmann and Kemmerzell (2010) below). In some situations, directional expectations in mvQCA generate easy counterfactuals that do not contribute to parsimony (Thiem 2014a).

If `details = TRUE`, parameters of fit (inclusion, raw coverage, and unique coverage) will be printed for each solution and its respective prime implicants. Essential prime implicants are listed first in the solution output and in the top part of the parameters-of-fit table. Inessential prime implicants are listed in brackets in the solution output and in the middle part of the parameters-of-fit table, together with their unique coverage scores under each individual model. Inclusion and coverage scores for each model are provided in the bottom part of the parameters-of-fit table.

The logical argument `show.cases` controls whether case names are displayed next to their corresponding prime implicants (do not use with many cases and/or long case names!). In the parameters-of-fit table, semicolons separate cases from different minterms, whereas commas separate cases from the same minterm.

The argument `inf.test` provides functionality for basing output function value codings on inference-statistical tests. Currently, only an exact binomial test (`"binom"`) is available, which requires the data to contain only bivalent or multivalent crisp-set factors. The argument requires a vector of length two, comprising the test and a critical significance level. If the empirical inclusion score of a minterm is not significantly lower than `incl.cut1`, it will be coded positive (`OUT = "1"`). If it is significantly lower than `incl.cut1` yet still significantly higher than `incl.cut0`, it will be coded as a contradiction (`OUT = "C"`). If it is not significantly higher than `incl.cut0`, it will be coded negative (`OUT = "0"`).

The argument `use.tilde` should only be used for bivalent factors. If the exogenous factors are already named with single letters, the argument `use.letters` will have no effect when set to `TRUE`. Otherwise, upper-case letters will replace original factor names in alphabetical order.

Value

An object of class 'qca' for single outcomes and 'mqca' for multiple outcomes. Objects of class 'qca' are lists with the following ten main components:

`tt`	The truth table object.
`excluded`	The line numbers of the negative minterms.
`initials`	The positive (non-remainder) minterms.
`PIs`	The prime implicants.
`PIchart`	The list of prime implicant charts.
`solution`	The list of solutions.
`essential`	The list of essential prime implicants.
`pims`	The list of model prime implicant set membership scores.
`SA`	The list of simplifying assumptions that would have been used by Quine-McCluskey minimization.
`i.sol`	A list of components specific to intermediate solution(s), including the prime implicant chart, model prime implicant membership scores, (non-simplifying) easy counterfactuals and difficult counterfactuals.

Contributors

Dusa, Adrian	: development, programming
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Baumgartner, Michael. 2009. "Inferring Causal Complexity." *Sociological Methods & Research* **38** (1):71-101. DOI: 10.1177/0049124109339369.

Baumgartner, Michael, and Alrik Thiem. 2017a. "Often Trusted but Never (Properly) Tested: Evaluating Qualitative Comparative Analysis." *Sociological Methods & Research*. Advance online publication. DOI: 10.1177/0049124117701487.

Baumgartner, Michael, and Alrik Thiem. 2017b. "Model Ambiguities in Configurational Comparative Research." *Sociological Methods & Research* **46** (4):954-87. DOI: 10.1177/0049124115610351.

Dusa, Adrian, and Alrik Thiem. 2015. "Enhancing the Minimization of Boolean and Multi-value Output Functions with eQMC." *Journal of Mathematical Sociology* **39** (2):92-108. DOI: 10.1080/0022250X.2014.897949.

Emmenegger, Patrick. 2011. "Job Security Regulations in Western Democracies: A Fuzzy Set Analysis." *European Journal of Political Research* **50** (3):336-64. DOI: 10.1111/j.1475-6765.2010.01933.x.

Hartmann, Christof, and Joerg Kemmerzell. 2010. "Understanding Variations in Party Bans in Africa." *Democratization* **17** (4):642-65. DOI: 10.1080/13510347.2010.491189.

McCluskey, Edward J. 1956. "Minimization of Boolean Functions." *Bell Systems Technical Journal* **35** (6):1417-44. DOI: 10.1002/j.1538-7305.1956.tb03835.x.

McCluskey, Edward J. 1965. *Introduction to the Theory of Switching Circuits*. Princeton: Princeton University Press.

Krook, Mona Lena. 2010. "Women's Representation in Parliament: A Qualitative Comparative Analysis." *Political Studies* **58** (5):886-908. DOI: 10.1111/j.1467-9248.2010.00833.x.

Ragin, Charles C. 2008. *Redesigning Social Inquiry: Fuzzy Sets and Beyond*. Chicago: University of Chicago Press. Link.

Schneider, Carsten Q., and Claudius Wagemann. 2012. *Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis (QCA)*. Cambridge: Cambridge University Press. Link.

Thiem, Alrik. 2014a. "Parameters of Fit and Intermediate Solutions in Multi-Value Qualitative Comparative Analysis." *Quality & Quantity* **49** (2):657-74. DOI: 10.1007/s11135-014-0015-x.

Thiem, Alrik. 2014b. "Navigating the Complexities of Qualitative Comparative Analysis: Case Numbers, Necessity Relations, and Model Ambiguities." *Evaluation Review* **38** (6):487-513. DOI: 10.1177/0193841x14550863.

Thiem, Alrik. 2015. "Using Qualitative Comparative Analysis for Identifying Causal Chains in Configurational Data: A Methodological Commentary on Baumgartner and Epple (2014)." *Sociological Methods & Research* **44** (4):723-36. DOI: 10.1177/0049124115589032.

See Also

pof, truthTable

Examples

```
# csQCA using Krook (2010)
#-------------------------
data(d.represent)
head(d.represent)

# solution with details and case names
KRO <- eQMC(d.represent, outcome = "WNP", details = TRUE, show.cases = TRUE)
KRO

# check PI chart
KRO$PIchart

# solution with truth table object
KRO.tt <- truthTable(d.represent, outcome = "WNP")
KRO <- eQMC(KRO.tt)
KRO

# simplifying assumptions (SAs) that would have been used with Quine-McCluskey
# optimization
KRO$SA
```

```
# fsQCA using Emmenegger (2011)
#-------------------------------
data(d.jobsecurity)
head(d.jobsecurity)

# solution with details
EMM <- eQMC(d.jobsecurity, outcome = "JSR", incl.cut1 = 0.9, details = TRUE)
EMM

# are the model prime implicants also sufficient for the negation of the outcome?
pof(EMM$pims, outcome = "JSR", d.jobsecurity, neg.out = TRUE, relation = "suf")

# are the negations of the model prime implicants also sufficient for the outcome?
pof(1 - EMM$pims, outcome = "JSR", d.jobsecurity, relation = "suf")

# plot all three prime implicants of the solution
PIsc <- EMM$pims
par(mfrow = c(1, 3))
for(i in 1:3){
 plot(PIsc[, i], d.jobsecurity$JSR, pch = 19, ylab = "JSR",
  xlab = names(PIsc)[i], xlim = c(0, 1), ylim = c(0, 1),
  main = paste("Prime Implicant", print(i)))
 mtext(paste(
  "Inclusion = ", round(EMM$IC$overall$incl.cov$incl[i], 3),
  "; Coverage = ", round(EMM$IC$overall$incl.cov$cov.r[i], 3)),
  cex = 0.7, line = 0.4)
 abline(h = 0.5, lty = 2, col = gray(0.5))
 abline(v = 0.5, lty = 2, col = gray(0.5))
 abline(0, 1)
}

# mvQCA using Hartmann and Kemmerzell (2010)
#------------------------------------------
data(d.partybans)
head(d.partybans)

# specify exogenous factors beforehand
exo.facs <- c("C", "F", "T", "V")

# parsimonious solution with contradictions included
HK.sol <- eQMC(d.partybans, outcome = "PB{1}", exo.facs = exo.facs,
  incl.cut0 = 0.4, sol.type = "ps+", details = TRUE)
HK.sol

# which are the two countries in T{2} but not PB{1}?
rownames(d.partybans[d.partybans$T == 2 & d.partybans$PB != 1, ])

# QCA with multiple outcomes from multivalent variables
#-----------------------------------------------------
d.mmv <- data.frame(A = c(2,0,0,1,1,1,2,2), B = c(2,2,2,2,1,1,0,0),
                    C = c(0,1,0,0,0,2,1,0), D = c(2,1,2,2,3,1,3,0),
                    E = c(3,2,3,3,0,1,3,2),
```

```
  row.names = letters[1:8])
head(d.mmv)

mmv.s <- eQMC(d.mmv, outcome = c("D{2}", "E{3}"))
mmv.s

# use quotes with curly-bracket notation to access solution component
print(mmv.s$"E{3}", details = TRUE, show.cases = TRUE)

# negation of outcome from multivalent factor is disjunction of all other
# levels; high under-determination (18 models)
mmv.s <- eQMC(d.mmv, outcome = "E{3}", neg.out = TRUE)
mmv.s

# causal chains with QCA (Thiem 2015); data from Baumgartner (2009)
#-------------------------------------------------------------------------------
d.Bau <- data.frame(
  U = c(1,1,1,1,0,0,0,0), D = c(1,1,0,0,1,1,0,0),
  L = c(1,1,1,1,1,1,0,0), G = c(1,0,1,0,1,0,1,0),
  E = c(1,1,1,1,1,1,1,0),
  row.names = letters[1:8])
head(d.Bau)

# with multiple outcomes, no solution details are printed;
# "causal-chain structure": (D + U <=> L) * (G + L <=> E)
# "common-cause structure": (D + U <=> L) * (G + D + U <=> E)
Bau.cna <- eQMC(d.Bau, outcome = names(d.Bau), relation = "sufnec")
Bau.cna

# get the truth table, solution details and case names for outcome "E"
print(Bau.cna$E, details = TRUE, show.cases = TRUE)

# examples relating to QCA method evaluation
#---------------------------------------------
#
# is the conservative solution (QCA-CS) really "conservative"?
#-------------------------------------------------------------
# Ragin (2008, 173): "The complex [conservative] solution [...] does not
# permit any counterfactual cases and thus no simplifying assumptions
# regarding combinations of conditions that do not exist in the data.";
# the conservative solution is "[c]onservative because [...] the
# researcher [...] is exclusively guided by the empirical information
# at hand" (Schneider and Wagemann 2012, 162)
#
# in fact, QCA-CS makes extremely strong assumptions on ALL remainders;
# QCA-CS assumes every remainder exists at least 'freq.cut' times,
# and occurs with the negation of the outcome more than
# 'freq.cut' * (1 - 'incl.cut1') times

# create a test data-set 'CS' with 32 cases and randomly assign values
# on the endogenous factor 'Z'
CS <- data.frame(mintermMatrix(rep(2,5)))
CS$Z <- sample(0:1, 2^5, replace = TRUE)
```

```
# randomly draw 20 cases to create a limitedly diverse data-set 'CS.LD'
# and turn all 12 remainder minterms into observations that occur with
# 'Z = 0' in original data-set 'CS'
CS.LD <- CS[sample(1:2^5, 20), ]
change <- as.numeric(setdiff(rownames(CS), rownames(CS.LD)))
CS$Z[change] <- 0

# create the (conservative) solutions for 'CS' and 'CS.LD'
CS.sol <- eQMC(CS, outcome = "Z")
CS.LD.sol <- eQMC(CS.LD, outcome = "Z", sol.type = "cs")

# test whether the two solutions are identical
identical(unlist(CS.LD.sol$solution), unlist(CS.sol$solution))

# both solutions are identical, for two datasets that do not allow the same
# causal inferences to be made; this indicates that QCA-CS draws causal inferences
# beyond what the data warrants; the lower the diversity index (ratio of non-remainder
# minterms to all minterms), the stronger the assumptions QCA-CS makes
```

`factorize` *Factorize Configurational Expressions*

Description

This function finds all possibilities for factorizing a configurational expression.

Usage

```
factorize(expression, and.split = "", sort.factorizing = FALSE,
          sort.factorized = FALSE)
```

Arguments

`expression` A string representing a configurational expression or a QCA solution object of class "qca" generated by `eQMC()`.

`and.split` The AND-operator (if any).

`sort.factorizing` Logical, sort results beginning with largest number of factorizing elements.

`sort.factorized` Logical, sort results beginning with largest number of factorized elements.

Details

In Boolean algebra, the "*"-operator is distributive over the "+"- operator such that for any three literals a, b and c, the following law holds: $(a * b) + (a * c) = a * (b + c)$ (Hohn 1966, pp.78-80; South 1974, p.12). The '`factorize`' function finds all possible a for any configurational expression. Factorized versions of the initial expression(s) can be sorted in decreasing order by the number of factorizing literals or in decreasing order by the number of factorized literals.

Value

A list with the following components:

`initial`	The initial expression.
`factored`	The factorizations of the initial expression.

Contributors

Dusa, Adrian	: development, programming, testing
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Hohn, Franz E. 1966. *Applied Boolean Algebra: An Elementary Introduction*. 2nd ed. New York: Macmillan.

South, G. F. 1974. *Boolean Algebra and Its Uses*. New York: Van Nostrand Reinhold.

See Also

`eQMC`

Examples

```
# factorize a disjunction of two two-way conjunctions;
# if single letters are used, argument "and.split" is not needed
factorize("AB + AC")

# "and.split" is needed in these cases
factorize("one*TWO*four + one*THREE + THREE*four", and.split = "*")
factorize("~ONE*TWO*~FOUR + ~ONE*THREE + THREE*~FOUR", and.split = "*")
factorize("one&TWO&four + one&THREE + THREE&four", and.split = "&")

# factorize solution objects directly
data(d.represent)
KRO.sol <- eQMC(d.represent, outcome = "WNP")
factorize(KRO.sol)
```

findTh *Find Calibration Thresholds*

Description

This function finds calibration thresholds for splitting base variables into the desired number of groups using cluster analysis.

Usage

```
findTh(x, groups = 2, hclustm = "complete", distm = "euclidean")
```

Arguments

`x`	An interval or ratio-scaled base variable.
`groups`	A vector of integers with the desired number of groups.
`hclustm`	The agglomeration (clustering) method to be used.
`distm`	The distance measure to be used.

Details

For more details about argument `groups`, see `?cutree`. For more details about argument `hclustm`, see `?hclust`. For more details about argument `distm`, see `?dist`.

Value

A numeric vector of suggested threshold(s) for dividing base variables into the desired number of groups.

Contributors

Dusa, Adrian	: programming
Thiem, Alrik	: development, documentation, testing

Note

Default values from the `hclust` method and the `dist` method are used for both the distance measure `distm` and the clustering method `hclustm`.

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

See Also

cutree, hclust, dist

Examples

```
# 15 random values between 1 and 100
x <- sample(1:100, size = 15)

# split into two groups for csQCA
findTh(x)

# split into three groups for mvQCA
findTh(x, groups = 3)
```

---

implicantMatrix *Create Implicant Matrices*

Description

This function creates implicant matrices. An implicant matrix consists of all truth table minterms and their subsets, including the empty set.

Usage

```
implicantMatrix(noflevels, raw = FALSE, arrange = FALSE)
```

Arguments

`noflevels`	The number of levels for each exogenous factor.
`raw`	Logical, return implicant matrix with indicator for elimination.
`arrange`	Logical, arrange for easier visual inspection.

Details

An implicant matrix consists of all minterms and their subsets, including the empty set (Dusa 2007, 2010; Thiem and Dusa 2015). The number of implicants q is given by $q = \prod_{j=1}^{k}(p_j + 1)$, where p_j is the number of levels for factor j and k is the total number of exogenous factors.

If `raw = TRUE`, the indicator for elimination (`-1`) is used.

Value

A matrix with $\prod_{j=1}^{k}(p_j + 1)$ rows and k columns.

Contributors

Dusa, Adrian	: programming
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Dusa, Adrian. 2007. *Enhancing Quine-McCluskey*. COMPASSS: Working Paper 2007-49. URL: http://www.compasss.org/wpseries/Dusa2007b.pdf.

Dusa, Adrian. 2010. "A Mathematical Approach to the Boolean Minimization Problem." *Quality & Quantity* 44 (1):99-113. DOI: 10.1007/s11135-008-9183-x.

Dusa, Adrian, and Alrik Thiem. 2015. "Enhancing the Minimization of Boolean and Multi-value Output Functions with *e*QMC." *Journal of Mathematical Sociology* **39** (2):92-108. DOI: 10.1080/0022250X.2014.897949.

See Also

mintermMatrix

Examples

```
# three exogenous factors with two levels each;
# first row is empty set
implicantMatrix(noflevels = rep(2, 3))

# two exogenous factors with three levels each
implicantMatrix(noflevels = rep(3, 2))

# arranged differently
implicantMatrix(noflevels = rep(3, 2), arrange = TRUE)

# with internal indicator for eliminated values
implicantMatrix(noflevels = rep(3, 2), raw = TRUE)
```

---

implicIndep — *Test for Implicational Independence between two Factors*

Description

This evaluation function tests for the implicational independence between two factors. It has been programmed for Thiem and Baumgartner (2016).

Usage

```
implicIndep(expression, n.samples = 1, size.sample = 100, corr = "0")
```

Arguments

`expression`	A string representing the Boolean function to be evaluated.
`n.samples`	The number of datasets to be sampled.
`size.sample`	The size of each data sample.
`corr`	The direction of correlation between the endogenous factor and the implicationally independent factor.

Details

Randomly sample `n.samples` different data-sets with uniform probability mass function (any other discrete function would do as well; proficient users may adjust this at the relevant places in the function); and run QCA for each dataset; check whether the irrelevant factor is eliminated and get the correlations between the irrelevant factor and the outcome factor.

The correlation can be controlled with `corr`: `"0"` means no correlation; `"+"` positive correlation, and `"-"` negative correlation. The larger the sample size, the larger the positive / negative correlation. The argument `expression` may represent any Boolean function in disjunctive normal form as shown below, including proper causal structures such as `"(X1*X2 + X3*X4 <=> Y)*(Z1 + z1)"` or non-causal structures such as `"(x1*X2 + X1*x2 + X1*X3 + X2*X3 <=> Y)*(Z1 + z1)"` (contains redundant prime implicants) or `"(X3*x2 + X2 <=> Y)*(Z1 + z1)"` (contains redundant conjuncts).

If `expression` is no causal structure, an additional note will be issued together with the test output for whether the irrelevant factor has been eliminated.

You can use the following possibilities for `expression`: `"(X1X2 + X3X4 <=> Y)(Z1 + z1)"` if a factor has one letter and a one-digit number, `"(AB + CD <=> E)(F + f)"` if a factor has one letter, `"(X1*X2 + X3*X4 <=> Y)*(Z1 + z1)"` or `"(A*B + C*D <=> E)*(F + f)"` (curly-bracket notation is not supported). Empty spaces and the type of the biconditional operator (<->/<=>) have no effect.

Value

A list with the following five components:

`tt`	The evaluated truth tables.
`dat.list`	The test datasets generated on the basis of the evaluaetd truth tables.
`sol.list`	The corresponding solutions.
`cor.list`	The correlations between the endogenous factor and the irrelevant factor.
`test`	The result for whether the irrelevant factor has been eliminated in all tests.

Contributors

Baumgartner, Michael	: testing
Thiem, Alrik	: development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Thiem, Alrik, and Baumgartner, Michael. 2016. "Modeling Causal Irrelevance in Evaluations of Configurational Comparative Methods." *Sociological Methodology* 46 (1):345-57. DOI: 10.1177/0081175016654736.

See Also

eQMC

Examples

```
## Not run:
# simulation with 10 sample datasets
simulation <- vector(mode = "list", length = 3)
n.samples <- 10

# directions of correlation and number of cases in each sample
corr <- c("-", "0", "+")
nofc <- c(40, 160, 640)
simulation <- lapply(nofc, function (x) {lapply(corr, function (y) {
  implicIndep("(X1*X2 + X3*X4 <=> Y)*(Z1 + z1)", n.samples, x, y)})})

# has Z1 been eliminated in all data experiments of a block of tests?
series.test <- matrix(sapply(1:length(corr), function (x) {sapply(1:length(nofc),
  function (y) {simulation[[x]][[y]]$test == "Z1 has been eliminated."})}),
  ncol = length(corr), dimnames = list(as.character(nofc), corr))
series.test

## End(Not run)
```

`limitedDiversity` *Analyze QCA Solution Behaviour under Limited Empirical Diversity*

Description

This evaluation function computes all solutions and unique models that result when all *n*-tuples of minterms are systematically eliminated from a truth table. It has initially been programmed for Baumgartner and Thiem (2017) to test the correctness of QCA's three search strategies (conservative/complex, intermediate, parsimonious).

Usage

```
limitedDiversity(truth.tab, outcome = "", exo.facs = c(""), sol.type = "ps",
                 dir.exp = c(), n.drop = 1, c.minterms = FALSE)
```

Arguments

`truth.tab`	A truth table (either in plain format or a truth table object of class "tt" generated by the `truthTable` function).
`outcome`	A character vector with the name of the outcome.
`exo.facs`	A character vector with the names of the exogenous factors.
`sol.type`	A character scalar specifying the QCA solution type that should be applied; either "ps" (parsimonious solution), "ps+" (parsimonious solution including both positive and contradiction minterms), "cs" (conservative solution) or "cs+" (conservative solution including both positive and contradiction minterms).
`dir.exp`	A vector of directional expectations for deriving intermediate solutions; can only be used in conjunction with `sol.type = "ps"` or `sol.type = "ps+"`.
`n.drop`	The number of minterms to be dropped from the truth table.
`c.minterms`	Logical, should contradictions be treated as positive minterms.

Details

This function computes all solutions and unique models that result when all n-tuples of observed minterms are systematically dropped from a truth table. It has been programmed for Baumgartner and Thiem (2017) to test the correctness of QCA's three search strategies (conservative/complex, intermediate, parsimonious) in conjunction with the `submodels` function.

The argument `truth.tab` specifies the truth table from which minterms are to be dropped. The truth table can either be in plain format or be a truth table object of class "tt" generated by the `truthTable` function. If it is a truth table object, the arguments `outcome` and `exo.facs` need not be specified. The main difference between a truth table in plain format (as also used by Coincidence Analysis, for example (Baumgartner 2009)), is that each minterm includes only cases that have identical values on the exogenous factors and the endogenous factor. A QCA truth table object, in contrast, consists of minterms that include both cases with the outcome being analyzed as well as cases with the negation of this outcome. The ratio between these cases is used as the basis for the output function value. Thus, dropping minterms from plain truth tables will drop all cases that are identical with respect to all factors in the factor frame, whereas dropping minterms from QCA truth table objects will drop all cases that are identical with respect to all exogenous factors in the factor frame.

The argument `n.drop` specifies the size of the tuples of minterms to be dropped for generating limited empirical diversity. For example, if the truth table has 16 observed minterms, `n.drop = 2` creates 120 2-tuples, `n.drop = 3` creates 560 3-tuples, and so on.

The argument `c.minterms` specifies whether contradictions should be treated as positive minterms (`TRUE`) or negative minterms (`FALSE`).

Value

A list with the following three components:

`model.shares`	All unique models for all n-tuples of dropped minterms and their occurrence shares.
`solutions`	The solutions for all n-tuples of eliminated minterms.
`tt`	The truth table.

Contributors

Dusa, Adrian	: programming
Thiem, Alrik	: development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Baumgartner, Michael. 2009. "Inferring Causal Complexity." *Sociological Methods & Research* **38** (1):71-101. DOI: 10.1177/0049124109339369.

Baumgartner, Michael, and Alrik Thiem. 2017. "Often Trusted but Never (Properly) Tested: Evaluating Qualitative Comparative Analysis." *Sociological Methods & Research*. Advance online publication. DOI: 10.1177/0049124117701487.

See Also

eQMC, submodels, truthTable

Examples

```
## Not run:
# number (n) of minterms (mt) and levels (lv) for each factor (exogenous
# and endogenous)
n.mt <- 2^5
n.lv <- rep(2, 5)

# expand to unevaluated truth table and assign case/factor labels
tt.unev <- data.frame(mintermMatrix(n.lv))
dimnames(tt.unev) <- list(1:n.mt, c(LETTERS[1:4], "Z"))

# cull rows from tt.unev that are compatible with aB + Bc + D <=> Z
# to produce evaluated truth table tt.ev
tt.ev <- tt.unev[pmax(tt.unev$D, pmin(1 - tt.unev$A, tt.unev$B),
                 pmin(tt.unev$B, 1 - tt.unev$C)) == tt.unev$Z, ]

# conservative solutions for all 1-tuples (16)
limitedDiversity(tt.ev, outcome = "Z", sol.type = "cs")$model.shares

# using a truth table object of class 'tt' created by eQMC function
#-------------------------------------------------------------------
data(d.represent)
tt <- truthTable(d.represent, outcome = "WNP")

# with objects of class 'tt', exogenous factors and the outcome need not be
# specified again
limitedDiversity(tt)
```

```
# proof that the conservative/complex solution type of QCA is incorrect,
# (see Baumgartner and Thiem (2017) for more details)
#----------------------------------------------------------------------

# 1. build truth table on the basis of reference model aB + Bc + D
tt <- data.frame(mintermMatrix(rep(2, 5)))
dimnames(tt) <- list(as.character(1:32), c(LETTERS[1:4], "OUT"))
tt <- tt[pmax(pmin(1 - tt$A, tt$B), pmin(tt$B, 1 - tt$C), tt$D) == tt$OUT, ]

# 2. generate all conservative/complex solutions for all 16 + 120 scenarios
# of one/two dropped minterm/s
sollist.cs <- vector("list", 2)
sollist.cs <- lapply(1:2, function (x) {
  limitedDiversity(tt, outcome = "OUT", sol.type = "cs", n.drop = x)
  }
)

# 3. compute in how many scenarios a correctness-preserving submodel of
# the reference model was part of the solution (43.75% for one dropped
# minterm and  16.67% for two dropped minterms)
cs.correct <- numeric(2)
cs.correct <- sapply(1:2, function (x) {round((sum(unlist(lapply(
  sollist.cs[[x]][[2]], function (y) {any(
    submodels("aB + Bc + D")$submodels %in% y)}
  ))) / choose(16, x))*100, 2)}
)
cs.correct

## End(Not run)
```

---

mintermMatrix *Create Minterm Matrices*

Description

This function creates minterm and implicant matrices. It is mainly used for internal and demonstration purposes.

Usage

```
mintermMatrix(noflevels, logical = FALSE)
```

Arguments

noflevels	The number of levels for each exogenous factor.
logical	Logical, return the matrix in logical values (only bivalent data).

Details

Minterm matrices contain all unique and complete conjunctions that can be formed from all levels of k factors (Dusa and Thiem 2015). The total number of minterms d is given by $d = \prod_{j=1}^{k} p_j$, where p_j is the number of levels for exogenous factor j and k is the total number of exogenous factors. A minterm matrix is an essential part of a truth table.

Contributors

Dusa, Adrian	: development, programming
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Dusa, Adrian, and Alrik Thiem. 2015. "Enhancing the Minimization of Boolean and Multi-value Output Functions with *e*QMC." *Journal of Mathematical Sociology* **39** (2):92-108. DOI: 10.1080/0022250X.2014.897949.

See Also

`implicantMatrix`, `truthTable`

Examples

```
# a minterm matrix with three bivalent exogenous factors
noflevels <- rep(2, 3)
mintermMatrix(noflevels)

# with logical values
mintermMatrix(noflevels, logical = TRUE)
```

`pof` *Compute Set-Relational Parameters of Fit*

Description

This function computes inclusion (consistency) and coverage scores.

Usage

```
pof(setms, outcome, data, neg.out = FALSE, relation = "suf",
    inf.test = "", incl.cut1 = 0.75, incl.cut0 = 0.5, ...)

is.pof(x)
```

Arguments

`setms`	A data frame of set membership scores, or a matrix of implicants, or a vector of implicant matrix line numbers.
`outcome`	The name of the outcome.
`data`	The working data set.
`neg.out`	Logical, use negation of `outcome`.
`relation`	The set relation to `outcome`, either `"nec"` or `"suf"`.
`inf.test`	The inference-statistical test to be performed (currently only `"binom"` for bivalent and multivalent crisp-set variables).
`incl.cut1`	The upper inclusion cut-off against which the empirical inclusion score is tested if `inf.test = "binom"`.
`incl.cut0`	The lower inclusion cut-off against which the empirical inclusion score is tested if `inf.test = "binom"`.
`...`	Other arguments (not used in this function).
`x`	An object of class "pof".

Details

The argument `setms` specifies a data frame of *set* membership scores, where *set* refers to any kind of set, including simple sets, combinations returned by the `superSubset` function (`coms`), prime implicants returned by the `eQMC` function (`pims`), or any other compound set.

The function also accepts a matrix of implicants with the level representation of created by the `mintermMatrix` function, or even a corresponding vector of implicant matrix line numbers.

The argument `outcome` specifies the outcome to be analyzed, either in curly-bracket notation (e.g., `O{value}`) if the outcome is from a multivalent (or a bivalent) factor, or in upper-case notation if the outcome is from a bivalent factor (e.g., `O` as a short-cut for `O{1}`). Outcomes from multivalent crisp-set factors always require curly-bracket notation. Outcomes must be single levels of factors not simultaneously passed to `exo.facs`.

The logical argument `neg.out` controls whether `outcome` is to be analyzed or its negation. If `outcome` is a level from a multivalent factor, `neg.out = TRUE` causes the disjunction of all remaining levels to become the outcome to be analyzed.

The argument `inf.test` provides functionality for adjudicating between rival hypotheses on the basis of inference-statistical tests. Currently, only an exact binomial test (`"binom"`) is available, which requires the data to contain only bivalent or multivalent crisp-set factors. Two one-tailed tests are performed. The null hypothesis with respect to `incl.cut1` is that the empirical inclusion score of each element in `setms` is not lower than the upper critical inclusion cut-off provided in `incl.cut1`. The null hypothesis with respect to `incl.cut0` is that the empirical inclusion score of each element in `setms` is not higher than the lower critical inclusion cut-off provided in `incl.cut0`.

Contributors

Dusa, Adrian	: programming
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Emmenegger, Patrick. 2011. “Job Security Regulations in Western Democracies: A Fuzzy Set Analysis.” *European Journal of Political Research* **50** (3):336-64. DOI: 10.1111/j.1475-6765.2010.01933.x.

Hartmann, Christof, and Joerg Kemmerzell. 2010. “Understanding Variations in Party Bans in Africa.” *Democratization* **17** (4):642-65. DOI: 10.1080/13510347.2010.491189.

Krook, Mona Lena. 2010. “Women’s Representation in Parliament: A Qualitative Comparative Analysis.” *Political Studies* **58** (5):886-908. DOI: 10.1111/j.1467-9248.2010.00833.x.

See Also

eQMC

Examples

```
# csQCA using Krook (2010)
#-------------------------
data(d.represent)
head(d.represent)

# solution with details
KRO <- eQMC(d.represent, outcome = "WNP", incl.cut1 = 0.9,
            details = TRUE)
KRO

# exact binomial tests of sufficiency inclusion
pof(KRO$pims, outcome = "WNP", d.represent, inf.test = c("binom", 0.1),
    incl.cut1 = 0.75, incl.cut0 = 0.5)

# fsQCA using Emmenegger (2011)
#------------------------------
data(d.jobsecurity)
head(d.jobsecurity)

# solution with details
EMM.sol <- eQMC(d.jobsecurity, outcome = "JSR", incl.cut1 = 0.9,
                details = TRUE)
EMM.sol
```

```
# are the model prime implicants also sufficient for the negation
# of the outcome?
pof(EMM.sol$pims, outcome = "JSR", d.jobsecurity, neg.out = TRUE)

# are the negations of the model prime implicants also sufficient
# for the outcome?
pof(1 - EMM.sol$pims, outcome = "JSR", d.jobsecurity)

# parameters of fit for matrix of implicants;
# "-1" is the placeholder for an eliminated variable;
# e.g.: R*p*V and S*c*L*P*v
#      "S"  "C"  "L"  "R"  "P"  "V"
#     [,1] [,2] [,3] [,4] [,5] [,6]
#[1,]   -1   -1   -1    1    0    1
#[2,]    1    0    1   -1    1    0
mat <- matrix(c(-1,-1,-1, 1, 0, 1,
                 1, 0, 1,-1, 1, 0), nrow = 2, byrow = TRUE)

pof(mat, outcome = "JSR", d.jobsecurity)

# or even vectors of line numbers from the implicant matrix
pof(c(43, 57), "JSR", d.jobsecurity)


# mv-data from Hartmann and Kemmerzell (2010)
#--------------------------------------------
data(d.partybans)
head(d.partybans)

# parameters of fit for several mv-expressions
expr <- c("C{1}", "F{2}", "T{2}", "T{1}*V{0}")
dat <- data.frame(ifelse(d.partybans$C == 1, 1, 0),
                  ifelse(d.partybans$F == 2, 1, 0),
                  ifelse(d.partybans$T == 2, 1, 0),
                  ifelse(d.partybans$T == 1 & d.partybans$V == 0, 1, 0))
colnames(dat) <- expr

pof(dat, outcome = "PB{1}", d.partybans)

# miscellaneous
#--------------
# parameters of fit for a data frame
x <- data.frame(A = c(1,1,1,0,1), B = c(1,1,1,0,1),
                C = c(0,1,0,0,1), D = c(0,0,1,0,1),
                O = c(1,1,1,0,1))

pof(x[, -5], outcome = "O", x)

# for a single column from that data frame
pof(x$A, x$O)

# for multiple columns from that data frame
pof(x[, 1:2], outcome = "O", x)
```

randomDGS *Build a Random Data-Generating Structure*

Description

This evaluation function can be used to randomly build data-generating structures. It has initially been programmed for Baumgartner and Thiem (2017) to test the correctness of QCA's three search strategies (conservative/complex, intermediate, parsimonious).

Usage

```
randomDGS(n.DGS = 1, exo.facs = c(""), seed.1 = NULL, seed.2 = NULL,
          prob = 0.5, diversity = 1, delete.trivial = FALSE)
```

Arguments

`n.DGS`	The number of random data-generating structures to be built.
`exo.facs`	A character vector with the names of the exogenous factors.
`seed.1`	The seed for the random generation of output function values.
`seed.2`	The seed for the random selection of a DGS in cases of structural ambiguities.
`prob`	The probability of assigning a positive output function value to a minterm.
`diversity`	The diversity index value.
`delete.trivial`	Logical, delete "TRUE" and "FALSE" from set of structures.

Details

The argument `n.DGS` specifies the number of random data-generating structures to be built.

The argument `exo.facs` is a character vector with the names of the exogenous factors.

The argument `seed.1` sets the seed for the random generation of output function values, whereas `seed.2` sets the seed for the random selection of a DGS in cases of structural ambiguities.

The argument `prob` is the probability of assigning a positive output function value to a minterm. The argument `diversity` specifies the diversity index value. It must be a number between 0 and 1.

The argument `delete.trivial` is logical, and specifies whether "TRUE" and "FALSE" should be deleted from the set of structures.

Value

A list with the following two components:

`DGS`	A vector of the data-generating structure(s).
`tt`	The corresponding truth table(s).

Contributors

Thiem, Alrik : development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Baumgartner, Michael, and Alrik Thiem. 2017. "Often Trusted but Never (Properly) Tested: Evaluating Qualitative Comparative Analysis." *Sociological Methods & Research*. Advance online publication. DOI: 10.1177/0049124117701487.

See Also

`submodels`

Examples

```
# randomly generate three data-generating structures on the basis of four
# exogenous factors
str <- randomDGS(n.DGS = 3, exo.facs = LETTERS[1:4], seed.1 = 1375, seed.2 = 3917)
str$DGS

# all correctness-preserving submodels of DGS 2, bd + abC, can then be found with the
# 'submodels' function
submodels(str$DGS[2])$submodels
```

`retention` *Compute Retention Probabilities of QCA Baseline Solutions*

Description

This evaluation function computes retention probabilities of QCA baseline solutions. It has been programmed for Thiem, Spoehel, and Dusa (2016).

Usage

```
retention(data, outcome = "", exo.facs = c(""), type = "corruption",
          assump = "DPA", n.cut = 1, incl.cut = 1, p.pert = 0.5,
          n.pert = 1)
```

Arguments

`data`	A dataset of bivalent crisp-set factors.
`outcome`	The name of the outcome.
`exo.facs`	A character vector with the names of the exogenous factors.
`type`	Induce errors on the endogenous factor or delete cases.
`assump`	Assume dependent or independent perturbations.
`n.cut`	The minimum number of cases for a minterm not to be considered as a remainder.
`incl.cut`	The minimum sufficiency inclusion score for an output function value of "1".
`p.pert`	The probability of perturbation under independence.
`n.pert`	The number of perturbations under dependence.

Details

This function computes exact retention probabilities of QCA baseline solutions for saturated truth tables and truth tables with a two-difference restriction (every remainder differs on at least two positions from every positive minterm).

The argument `data` requires a suitable dataset. Suitable datasets have the following structure: values of "0" and "1" for bivalent crisp-set factors.

The argument `exo.facs` specifies the exogenous factors. If omitted, all factors in `data` are used except that of the `outcome`.

The argument `type` specifies whether errors are to be induced in the endogenous factor ("1" is recoded to "0"; "0" is recoded to "1") of cases or whether entire cases are to be deleted from the data.

The argument `assump` specifies whether the perturbations detailed in `type` occur independently of each other or whether they are dependent on each other. Note that the assumption of dependence increases the consumption of computational resources significantly.

Minterms that contain fewer than `n.cut` cases with membership scores above 0.5 are coded as remainders (`OUT = "?"`). If the number of such cases is at least `n.cut`, minterms with an inclusion score of at least `incl.cut` are coded positive (`OUT = "1"`), and minterms with an inclusion score below `incl.cut` are coded negative (`OUT = "0"`). The possibility to specify contradictions using a second inclusion cut-off as in the `truthTable` function does not exist.

The argument `p.pert` specifies the probability of perturbation for `type = "independent"`. For example, if `p.pert = 1`, each case is guaranteed to have measurement error on the endogenous factor.

The argument `n.pert` specifies the number of perturbations for `type = "dependent"`. This must be an integer between zero (no case suffers from measurement error in the endogenous factor or no case gets deleted) and the total number of cases in `data` (all cases suffer from measurement error in the endogenous factor or all cases get deleted.)

Contributors

Dusa, Adrian	: programming, testing
Spoehel, Reto	: development
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Hug, Simon. 2013. "Qualitative Comparative Analysis: How Inductive Use and Measurement Error lead to Problematic Inference." *Political Analysis* **21** (2):252-65. DOI: 10.1093/pan/mps061.

Thiem, Alrik, Reto Spoehel, and Adrian Dusa. 2016. "Enhancing Sensitivity Diagnostics for Qualitative Comparative Analysis: A Combinatorial Approach." *Political Analysis* **24** (1):104-20. DOI: 10.1093/pan/mpv028.

See Also

`truthTable`

Examples

```
# replicate results from Hug (2013) for 2 deleted cases
#-------------------------------------------------------
dat <- data.frame(matrix(c(
   rep(1,25),rep(0,20),rep(c(0,0,1,0,0),3),
   0,0,0,1,0,0,1,0,0,0,0,rep(1,7),0,1),
   nrow = 16, byrow = TRUE, dimnames = list(c(
    "AT","DK","FI","NO","SE","AU","CA","FR",
    "US","DE","NL","CH","JP","NZ","IE","BE"),
   c("P", "U", "C", "S", "W"))
))

retention(dat, outcome = "W", type = "deletion", assump = "dependent", n.pert = 2)
```

---

submodels — *Compute All Correctness-Preserving Submodels of a QCA Reference Model*

Description

This evaluation function computes all correctness-preserving submodels of a QCA reference model. It has initially been programmed for Baumgartner and Thiem (2015) to test the correctness of QCA's three search strategies (conservative/complex, intermediate, parsimonious).

Usage

```
submodels(expression, noflevels = c(), test = TRUE)
```

Arguments

`expression`	A string representing a csQCA or an fsQCA model, or a csQCA or fsQCA solution object of class 'qca' (created by the `eQMC` function).
`noflevels`	A numeric vector specifying the number of levels for each factor (experimental, can be ignored).
`test`	Logical, test whether `expression` is a causal structure.

Details

This function has initially been programmed for Baumgartner and Thiem (2015) to test the correctness of QCA's three solution types (conservative/complex, intermediate, parsimonious). It computes all submodels of a csQCA or an fsQCA reference model that do not violate the criterion of correctness (mvQCA models are not yet supported). The following expression structures can be used: `"A*B + C*D <=> Y"` or `"AB + CD <=> Y"`. Empty spaces and the type of conditional operator (<->/<=>/->/=>) are irrelevant, but only single letters are allowed for exogenous factors. The full model need not be provided; the antecedent also suffices (e.g., `"AB + CD"`).

Objects of class 'qca', which are returned by the `eQMC` function, are also accepted, provided that all exogenous factors have a single-letter label (set the argument `use.letters` to `TRUE` in the function call to `eQMC` if original factor labels are not single letters).

The argument `noflevels` expects a numeric vector of the number of factor levels with a `names` attribute. Currently, this argument is experimental and can be ignored.

The argument `test` specifies whether `expression` should be pre-tested for its causal interpretability before forming submodels. The value to this argument does not affect whether basic tests for likely typos in expressions such as `"abb <-> C"` or `"abB <-> C"` are performed. If `expression` is an object of class 'qca', `test` will be set to `FALSE` because QCA models generated by the `eQMC` function at default argument settings are always causally interpretable.

Note that for highly complex models containing many conjuncts within many disjuncts, computing times tend to increase considerably.

Value

A list with the following four main components:

`model`	The reference model.
`noflevels`	The number of levels for each factor in the factor frame of the model.
`outcome`	The outcome specified as part of the expression or a pseudo outcome if only an antecedent was specified.
`submodels`	A character vector of all correctness-preserving submodels.

Contributors (alphabetical)

Baumgartner, Michael	: development, testing
Thiem, Alrik	: development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Baumgartner, Michael, and Alrik Thiem. 2015. *Often Trusted but Never (Properly) Tested: Evaluating Qualitative Comparative Analysis*. Paper presented at the 12th Conference of the European Sociological Association, 25-28 August, Czech Technical University, Prague (Czech Republic). Link.

See Also

eQMC, limitedDiversity

Examples

```
## Not run:
# provide a) a full model as an equivalence and inspect its submodels
models1 <- submodels("a*B + B*c + D <-> Z")
models1$submodels

# ... b) a full model with a negated outcome
# submodels
models2 <- submodels("AcD + BCD + abcd <=> e")
length(models2$submodels)

# ... c) or only an antecedent
models3 <- submodels("aB + Bc + D")
models3$submodels

# directly provide an object of class 'qca' generated by the 'eQMC' function,
# even when the solution comprises multiple models; specify
# 'use.letters = TRUE' when the original exogenous factors have multi-letter
# labels; for example:
data(d.represent)
sol1 <- eQMC(d.represent, outcome = "WNP", neg.out = TRUE, use.letters = TRUE)
sol1
# M1: ae + cde + (bdE) <=> wnp
# M2: ae + cde + (bcd) <=> wnp
# M3: ae + cde + (Abc) <=> wnp
# M1 has 138 submodels, M2 has 129, and M3 has 139 submodels
models4 <- submodels(sol1)
sapply(models4, "[")

# when original labels of exogenous factors already consist of single
# letters only, 'use.letters = TRUE' need not be specified
data(d.napoleon)
sol2 <- eQMC(d.napoleon, outcome = "O")
sol2
```

```
models5 <- submodels(sol2)
sapply(models5, "[")

# prior testing is recommended because non-causal models can sometimes only
# be identified computationally
submodels("aB + Ac + Ad + bc + bd + CD")

# can a + AbC => Y be an acceptable QCA solution as Schneider and Wagemann
# (2012, p. 108) argue? No, because in Boolean algebra, it holds that
# F + fG = (F + f) * (F + G) = 1*(F + G) = F + G by the laws of distribution,
# complementarity, and identity
submodels("a + AbC => Y", test = TRUE)

# proof that the conservative/complex solution type of QCA is incorrect,
# using model 3 from above (see Baumgartner and Thiem (2015) for more details)

# 1. build saturated truth table on the basis of model 3: aB + Bc + D
tt <- data.frame(mintermMatrix(rep(2, 5)))
dimnames(tt) <- list(as.character(1:32), c(LETTERS[1:4], "OUT"))
tt <- tt[pmax(pmin(1 - tt$A, tt$B), pmin(tt$B, 1 - tt$C), tt$D) == tt$OUT, ]

# 2. use function 'limitedDiversity' to generate all conservative/complex
# solutions for all 16 + 120 scenarios of one/two dropped minterm/s
sollist.cs <- vector("list", 2)
sollist.cs <- lapply(1:2, function (x) {
  limitedDiversity(tt, outcome = "OUT", sol.type = "cs", n.drop = x)
  }
)

# 3. compute in how many scenarios a correctness-preserving submodel of
# model 3 was part of the solution (43.75% for one dropped minterm and
# 16.67% for two dropped minterms)
cs.correct <- numeric(2)
cs.correct <- sapply(1:2, function (x) {round((sum(unlist(lapply(
  sollist.cs[[x]][[2]], function (y) {any(models3$submodels %in% y)}
  ))) / choose(16, x))*100, 2)}
)
cs.correct

## End(Not run)
```

---

superSubset *Find Superset and Subset Relations*

Description

This helper function finds all combinations of conditions among all possible combinations that optimize the fulfilment of the specified criteria for a superset (necessity) or subset (sufficiency) relation to the outcome.

Usage

```
superSubset(data, outcome = "", neg.out = FALSE, exo.facs = c(""),
            relation = "nec", incl.cut = 1, cov.cut = 0,
            use.tilde = FALSE, use.letters = FALSE, ...)
```

Arguments

`data`	A dataset of bivalent or multivalent crisp-set factor or bivalent fuzzy-set variables.
`outcome`	The name of the outcome.
`neg.out`	Logical, use negation of `outcome`.
`exo.facs`	A character vector with the names of the exogenous factors.
`relation`	The relation to `outcome`, either `"nec"`, `"suf"`, `"necsuf"` or `"sufnec"`.
`incl.cut`	The minimal inclusion score of the relation.
`cov.cut`	The minimal coverage score of the relation.
`use.tilde`	Logical, use "~" for negation with bivalent variables.
`use.letters`	Logical, use simple letters instead of original factor names.
`...`	Other arguments for backward compatibility.

Details

This helper function to the `testTESA` function returns a list of those of the $\prod_{j=1}^{k}(p_j + 1) - 1$ potential value combinations, where p_j is the number of values for exogenous variable j and k is the number of exogenous variables, that define minimal condition sets for the specified inclusion (consistency) and coverage score cut-offs with respect to an outcome.

If `relation = "nec"` (default), the function finds (combinations of) conditions that are supersets of (necessary for) the outcome. It starts with an initiation set, which is comprised of all $\sum_{j=1}^{k} p_j$ simple condition sets. This set is expanded by incrementally forming set-theoretic intersections of a higher order as long as `incl.cut` and `cov.cut` are still met (the former always takes precedence over the latter). If suitable conjunctions exist, they will be returned, together with all their lower-order conjuncts.

If none of the simple conditions or their negations in the initiation set passes `incl.cut`, disjunctions instead of conjunctions are formed until `incl.cut` and `cov.cut` will have been met. Only the disjunctions thus found will be returned.

If `relation = "suf"`, the function finds (combinations of) conditions that are subsets of (sufficient for) the outcome. The initiation set is comprised of all $\prod_{j=1}^{k} p_j$ intersections of order k. This set is reduced by incrementally forming intersections of a lower order as long as `incl.cut` and `cov.cut` are still met. Only the intersections of the lowest order will be printed. For more details, see Thiem and Dusa (2013). For `relation = "necsuf"` and `relation = "sufnec"`, `incl.cut` will be applied to each relation and `cov.cut` has no effect.

The argument `outcome` specifies the outcome. Outcomes from multivalent variables require curly-bracket notation (`X{value}`).

The logical argument `neg.out` controls whether `outcome` is to be used or its negation. If `outcome` is from a multivalent crisp-set factor, `neg.out = TRUE` has the effect that the disjunctions of all remaining values becomes the new outcome.

The argument exo.facs specifies the names of the exogenous factors. If omitted, all factors in data are used except the factor of which outcome is a level.

The argument use.tilde only applies to bivalent factors. If factors are already named with single letters, the argument use.letters has no effect.

Value

A list with the following two main components:

incl.cov	A data frame with the parameters of fit.
coms	A data frame with the combination membership scores.

Contributors

Dusa, Adrian	: development, programming
Thiem, Alrik	: development, documentation, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Ragin, Charles C. 2009. "Qualitative Comparative Analysis Using Fuzzy Sets (fsQCA)." In *Configurational Comparative Methods: Qualitative Comparative Analysis (QCA) and Related Techniques*, ed. B. Rihoux and C. C. Ragin. London: Sage Publications, pp. 87-121.

Schneider, Carsten Q., and Claudius Wagemann. 2013. "Doing Justice to Logical Remainders in QCA: Moving Beyond the Standard Analysis." *Political Research Quarterly* **66** (1):211-20. DOI: 10.1177/1065912912468269.

Thiem, Alrik. 2015. *Standards of Good Practice and the Methodology of Necessary Conditions in Qualitative Comparative Analysis: A Critical View on Schneider and Wagemann's Theory-Guided/Enhanced Standard Analysis*. COMPASSS WP Series 2015-83. URL: http://www.compasss.org/wpseries/Thiem201

See Also

testTESA

Examples

```
# Schneider and Wagemann (2013, 212), using data from Ragin
# (2009, 95), only present G and L as minimally necessary conditions
#--------------------------------------------------------------------

LIP <- data.frame(
 D = c(0.81,0.99,0.58,0.16,0.58,0.98,0.89,0.04,0.07,
```

```
       0.72,0.34,0.98,0.02,0.01,0.01,0.03,0.95,0.98),
 U = c(0.12,0.89,0.98,0.07,0.03,0.03,0.79,0.09,0.16,
       0.05,0.10,1.00,0.17,0.02,0.03,0.30,0.13,0.99),
 L = c(0.99,0.98,0.98,0.98,0.99,0.99,0.99,0.13,0.88,
       0.98,0.41,0.99,0.59,0.01,0.17,0.09,0.99,0.99),
 I = c(0.73,1.00,0.90,0.01,0.08,0.81,0.96,0.36,0.07,
       0.01,0.47,0.94,0.00,0.11,0.00,0.21,0.67,1.00),
 G = c(0.43,0.98,0.91,0.91,0.58,0.95,0.31,0.43,0.13,
       0.95,0.58,0.99,0.00,0.01,0.84,0.20,0.91,0.98),
 S = c(0.05,0.95,0.89,0.12,0.77,0.95,0.05,0.06,0.42,
       0.92,0.05,0.95,0.12,0.05,0.21,0.06,0.95,0.95)
)

rownames(LIP) <- c("AT","BE","CZ","EE","FI","FR","DE","GR","HU",
                   "IE","IT","NL","PL","PT","RO","ES","SE","UK")

rownames(superSubset(LIP, outcome = "S", incl.cut = 0.9)$incl.cov)

# with mv-data from Hartmann and Kemmerzell (2010)
#-------------------------------------------------
data(d.partybans)
head(d.partybans)

HK <- superSubset(d.partybans, outcome = "PB",
  exo.facs = c("C", "F", "T", "V"), incl.cut = 0.75)
HK

# combination membership scores for all cases (only first four
# combinations and first ten lines displayed)
HK$coms[1:10, 1:4, drop = FALSE]
```

---

testTESA — *Testing Schneider and Wagemann's Theory-Guided/Enhanced Standard Analysis (T/ESA)*

Description

This evaluation function can be used to test the implications of Schneider and Wagemann's Theory-Guided/Enhanced Standard Analysis (T/ESA; Schneider and Wagemann 2013), and in particular, the procedure's first two stages, with respect to the extent of remainders that would have to be declared insufficient for the outcome. It has been programmed for Thiem (2016).

Usage

```
testTESA(data, outcome = "", neg.out = FALSE, exo.facs = c(""), n.cut = 1,
         incl.cut1 = 1, incl.cut0 = 1)
```

Arguments

`data`	A dataset of bivalent crisp-set factors or bivalent fuzzy-set factors or multivalent crisp-set factors.
`outcome`	The name of the outcome.
`neg.out`	Logical, use negation of `outcome`.
`exo.facs`	A character vector with the names of the exogenous factors.
`n.cut`	The minimum number of cases with set membership score above 0.5 for an output function value of "0", "1" or "C".
`incl.cut1`	The minimum sufficiency inclusion score for an output function value of "1".
`incl.cut0`	The maximum sufficiency inclusion score for an output function value of "0".

Details

The arguments `data`, `outcome`, `exo.facs`, `n.cut`, `incl.cut1` and `incl.cut0` are those of the `eQMC` function.

Value

A numeric vector with the percentages of remainder minterms that would have been used as simplifying assumptions by Quine-McCluskey optimization but that were declared to be insufficient for the outcome by T/ESA.

Contributors

Thiem, Alrik : development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Ragin, Charles C. 2009. “Qualitative Comparative Analysis Using Fuzzy Sets (fsQCA).” In *Configurational Comparative Methods: Qualitative Comparative Analysis (QCA) and Related Techniques*, ed. B. Rihoux and C. C. Ragin. London: Sage Publications, pp. 87-121.

Schneider, Carsten Q., and Claudius Wagemann. 2013. “Doing Justice to Logical Remainders in QCA: Moving Beyond the Standard Analysis.” *Political Research Quarterly* **66** (1):211-20. DOI: 10.1177/1065912912468269.

Thiem, Alrik. 2016. “Standards of Good Practice and the Methodology of Necessary Conditions in Qualitative Comparative Analysis.” *Political Analysis* **24** (4):478-84. DOI: 10.1093/pan/mpw024.

See Also

`eQMC`, `truthTable`, `superSubset`

Examples

```
# Schneider and Wagemann (2013, 212), using data from Ragin
# (2009, 95), only present L and S as minimally necessary conditions
#-----------------------------------------------------------------

LIP <- data.frame(
 D = c(0.81,0.99,0.58,0.16,0.58,0.98,0.89,0.04,0.07,
       0.72,0.34,0.98,0.02,0.01,0.01,0.03,0.95,0.98),
 U = c(0.12,0.89,0.98,0.07,0.03,0.03,0.79,0.09,0.16,
       0.05,0.10,1.00,0.17,0.02,0.03,0.30,0.13,0.99),
 L = c(0.99,0.98,0.98,0.98,0.99,0.99,0.99,0.13,0.88,
       0.98,0.41,0.99,0.59,0.01,0.17,0.09,0.99,0.99),
 I = c(0.73,1.00,0.90,0.01,0.08,0.81,0.96,0.36,0.07,
       0.01,0.47,0.94,0.00,0.11,0.00,0.21,0.67,1.00),
 G = c(0.43,0.98,0.91,0.91,0.58,0.95,0.31,0.43,0.13,
       0.95,0.58,0.99,0.00,0.01,0.84,0.20,0.91,0.98),
 S = c(0.05,0.95,0.89,0.12,0.77,0.95,0.05,0.06,0.42,
       0.92,0.05,0.95,0.12,0.05,0.21,0.06,0.95,0.95)
)

rownames(LIP) <- c("AT","BE","CZ","EE","FI","FR","DE","GR","HU",
                   "IE","IT","NL","PL","PT","RO","ES","SE","UK")

superSubset(LIP, outcome = "S", incl.cut = 0.9)

testTESA(LIP, outcome = "S", incl.cut1 = 0.75)
```

`truthTable` *Create a Truth Table*

Description

This function creates truth tables from configurational data.

Usage

```
truthTable(data, outcome = "", neg.out = FALSE, exo.facs = c(""),
           n.cut = 1, incl.cut1 = 1, incl.cut0 = 1, complete = FALSE,
           show.cases = FALSE, sort.by = c(""), decreasing = TRUE,
           inf.test = c(""), use.letters = FALSE, ...)

is.tt(x)
```

Arguments

`data`	A set of configurational data (of class 'matrix' or 'data.frame').
`outcome`	The name of the outcome.
`neg.out`	Logical, use the negation of `outcome`.

exo.facs	A character vector with the names of the exogenous factors.
n.cut	The minimum number of cases with set membership score above 0.5 for an output function value of "0", "1" or "C"; an integer between 1 and the maximum number of cases for all non-remainder minterms.
incl.cut1	The minimum sufficiency inclusion score for an output function value of "1".
incl.cut0	The maximum sufficiency inclusion score for an output function value of "0".
complete	Logical, print the complete truth table.
show.cases	Logical, print case names (do not use this option with many cases and/or long case names).
sort.by	Sort the truth table by inclusion scores and/or number of cases.
decreasing	Sort in decreasing or increasing order of value(s) passed to sort.by.
inf.test	A vector of length two specifying the inference-statistical test to be performed (currently only "binom") and the critical significance level.
use.letters	Logical, use single letters (in alphabetical order) instead of original variable names.
...	Other arguments.
x	An object of class 'tt'.

Details

The argument data can be a truth table object (an object of class 'tt' returned by the truthTable function) or a suitable data set. Suitable data sets have the following structure: values of 0 and 1 for bivalent crisp-set factors, values between 0 and 1 for bivalent fuzzy-set factors, and values beginning with 0 at increments of 1 for multivalent crisp-set factors. The placeholders "-" and "dc" indicate "don't cares" in auxiliary factors that specify temporal order between other substantive factors in tQCA. These values lead to the exclusion of the auxiliary factor from the computation of parameters of fit.

The argument outcome specifies the outcome to be analyzed, either in curly-bracket notation (e.g., O{value}) if the outcome is from a multivalent (or a bivalent) factor, or in upper-case notation if the outcome is from a bivalent factor (e.g., O as a short-cut for O{1}). Outcomes from multivalent crisp-set factors always require curly-bracket notation. Outcomes must be single levels of factors not simultaneously passed to exo.facs.

The logical argument neg.out controls whether outcome is to be analyzed or its negation. If outcome is a level from a multivalent factor, neg.out = TRUE causes the disjunction of all remaining levels to become the outcome to be analyzed.

The argument exo.facs specifies the exogenous factors. If omitted, all factors in data are used except that of the outcome. Please note that computation times may increase significantly beyond 17 exogenous factors, and that the computation of a solution may not be possible at all depending on end-user machine constraints.

Minterms that contain fewer than n.cut cases with membership scores above 0.5 are coded as remainders (OUT = "?"). If the number of such cases is at least n.cut, minterms with an inclusion score of at least incl.cut1 are coded positive (OUT = "1"), minterms with an inclusion score below incl.cut1 but with at least incl.cut0 are coded as a contradiction (OUT = "C"), and minterms with an inclusion score below `incl.cut0` are coded negative (`OUT = "0"`). If `incl.cut0` is not explicitly changed, it is set equal to `incl.cut1`.

The logical argument `show.cases` controls whether case names are displayed next to their corresponding minterm (do not use this option with many cases and/or long case names).

The `sort.by` argument orders all minterms by their inclusion scores (*incl*) or the number of cases with membership above 0.5 they contain (*n*) or both, in either order.

If the exogenous factors are already named with single letters, the argument `use.letters` will have no effect when set to `TRUE`. Otherwise, upper-case letters will replace original factor names in alphabetical order.

The argument `inf.test` provides functionality for basing output function value codings on inference-statistical tests. Currently, only an exact binomial test (`"binom"`) is available, which requires the data to contain only bivalent or multivalent crisp-set factors. The argument requires a vector of length two, comprising the test and a critical significance level. If the empirical inclusion score of a minterm is not significantly lower than `incl.cut1`, it will be coded positive (`OUT = "1"`). If it is significantly lower than `incl.cut1` yet still significantly higher than `incl.cut0`, it will be coded as a contradiction (`OUT = "C"`). If it is not significantly higher than `incl.cut0`, it will be coded negative (`OUT = "0"`).

Value

An object of class 'tt', which is a list with the following six main components:

`tt`	The truth table.
`indexes`	The minterm line numbers.
`noflevels`	A vector with the number of levels of the exogenous factors.
`initial.data`	The initial data.
`recoded.data`	Recoded data (if crisp, same as `initial.data`; if fuzzy, dichotomized version of `initial.data`).
`cases`	The cases with membership above 0.5 in a minterm.

Contributors

Dusa, Adrian	: development, programming
Thiem, Alrik	: development, documentation, programming, testing

Author(s)

Alrik Thiem (Personal Website; ResearchGate Website)

References

Emmenegger, Patrick. 2011. "Job Security Regulations in Western Democracies: A Fuzzy Set Analysis." *European Journal of Political Research* **50** (3):336-64. DOI: 10.1111/j.1475-6765.2010.01933.x.

Hartmann, Christof, and Joerg Kemmerzell. 2010. “Understanding Variations in Party Bans in Africa.” *Democratization* **17** (4):642-65. DOI: 10.1080/13510347.2010.491189.

Krook, Mona Lena. 2010. “Women’s Representation in Parliament: A Qualitative Comparative Analysis.” *Political Studies* **58** (5):886-908. DOI: 10.1111/j.1467-9248.2010.00833.x.

Ragin, Charles C., and Sarah Ilene Strand. 2008. “Using Qualitative Comparative Analysis to Study Causal Order: Comment on Caren and Panofsky (2005).” *Sociological Methods & Research* **36** (4):431-41. DOI: 10.1177/0049124107313903.

Examples

```
# csQCA using Krook (2010)
#-------------------------
data(d.represent)
head(d.represent)

# print truth table; if all factors except that of the outcome in
# the data should be included as exogenous factors, then these need
# not be specified separately
truthTable(d.represent, outcome = "WNP")

# print complete truth table, show cases, and first sort by
# inclusion scores, then by number of cases
truthTable(d.represent, outcome = "WNP", complete = TRUE,
  show.cases = TRUE, sort.by = c("incl", "n"))

# code minterms with a single case as remainders (note: use of
# 'n.cut' should be well justified)
KRO.tt <- truthTable(d.represent, outcome = "WNP", n.cut = 2,
  show.cases = TRUE)
KRO.tt

# print cases that were assigned to remainders based on argument 'n.cut'
KRO.tt$excluded

# fsQCA using Emmenegger (2011)
#------------------------------
data(d.jobsecurity)
head(d.jobsecurity)

# code non-remainder minterms with inclusion scores between 0.4
# and 0.8 as contradictions (note: these are not 'contradictions'
# in the logical sense of the word but minterms that can neither
# be coded as sufficient nor as insufficient for the outcome)
truthTable(d.jobsecurity, outcome = "JSR", incl.cut1 = 0.8, incl.cut0 = 0.4)

# truth table based on the negated outcome
truthTable(d.jobsecurity, outcome = "JSR", neg.out = TRUE, incl.cut1 = 0.8,
  incl.cut0 = 0.4)

# mvQCA using Hartmann and Kemmerzell (2010)
#-------------------------------------------
data(d.partybans)
```

```
head(d.partybans)

# code non-remainder minterms with inclusion scores below 1
# but above 0.4 as contradictions
HK.tt <- truthTable(d.partybans, outcome = "PB",
  exo.facs = c("C","F","T","V"), incl.cut0 = 0.4)
HK.tt

# list the number of levels for the exogenous factors
HK.tt$noflevels

# which minterms have more than 2 cases?
HK.tt$tt[which(HK.tt$tt$n > 2), ]

# code output function values in truth table based on
# exact binomial test
truthTable(d.partybans, outcome = "PB", exo.facs = c("C","F","T"),
  incl.cut1 = 0.9, incl.cut0 = 0.4, show.cases = TRUE,
  inf.test = c("binom", 0.1))

# tQCA using Ragin and Strand (2008)
#----------------------------------
data(d.graduate)
head(d.graduate)

# tQCA truth table with "don't care" values
truthTable(d.graduate, outcome = "REC")
```

Index