# Package 'QCA3'

January 2, 2012

**Type** Package

**Title** Yet another package for Qualitative Comparative Analysis

**Version** 0.0-4

**Date** 2011-09-20

**Author** Ronggui HUANG

**Maintainer** Ronggui HUANG<ronggui.huang@gmail.com>

**Imports** lpSolve

**Suggests** venneuler, QCA

**Description** A set of functions for Qulitative Comparative Analysis (QCA). It can be used for various types of QCA (csQCA, mvQCA fsQCA and crip set TQCA and time-serious QCA). It has methods for simplifying assumption, contradictory simplifying assumption.

**License** BSD

**LazyLoad** yes

**LazyData** yes

**URL** http://asrr.r-forge.r-project.org/

**Repository** CRAN

**Date/Publication** 2011-09-20 08:33:19

## R topics documented:

---

`CarenPanofsky` *Hypothetical data set about TQCA.*

---

## Description

This is a hypothetical data set about TQCA in Caren and Panksky (2005).

## Usage

```
data(CarenPanofsky)
```

## Format

A data frame with 17 observations on the following 6 variables.

`public` a numeric vector

`elite` a numeric vector

`affiliate` a numeric vector

`strike` a numeric vector

`e_before_a` a numeric vector

`recognition` a numeric vector

## Details

For condition e_before_a, -9 denotes don't care.

## References

Caren, Neal and Aaron Panofsky 2005. "TQCA: A Technique for Adding Temporality to Qualitative Comparative Analysis." Sociological Methods Research 34 (2) : 147-172.

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) : 431-441.

## Examples

```
data(CarenPanofsky)
tqca.tt <- cs_truthTable(CarenPanofsky,'recognition',names(CarenPanofsky)[1:5])
reduce(tqca.tt) ## result in Ragin and Strad (2008: 438)
```

---

CoD *The consolidation of Democracy.*

---

## Description

Data used in "The consolidation of democracy: comparing Europe and Latin America", Schneider (2009).

## Usage

```
data(CoD)
```

## Format

A data frame with 32 observations on the following 20 variables.

`country` a factor with levels `Albania` `Argentina` `Belarus` `Bolivia` `Brazil` `Bulgaria` `Chile` `Czech_R` `Ecuador` `Estonia` `Georgia` `Greece` `Guatemala` `Honduras` `Hungary` `Latvia` `Lithuania` `Mexico` `Mongolia` `Nicaragua` `Paraguay` `Peru` `Poland` `Portugal` `Romania` `Russia` `Slovakia` `Slovenia` `Spain` `Turkey` `Ukraine` `Uruguay`

`label` a factor with levels `AL` `AR` `BE` `BO` `BR` `BU` `CH` `CR` `EC` `EST` `GE` `GR` `GUA` `HO` `HU` `LAT` `LIT` `MO` `MX` `NI` `PA` `PE` `PL` `PO` `RO` `RU` `SK` `SL` `SP` `TU` `UK` `UR`

`CODPERCENTAGE` a factor with levels `14.58` `26.14` `26.39` `31.48` `33.58` `39.46` `43.45` `48.96` `49.34` `54.17` `54.66` `62.5` `65.42` `65.91` `66.67` `68.33` `69.17` `70.14` `70.45` `71.25` `78.55` `78.85` `79.92` `82.29` `83.58` `missing`

`cod` a numeric vector

`GDP` a numeric vector

`econdev` a numeric vector

`EDUCATION` a numeric vector

`eduhi` a numeric vector

`ETHNOLING` a numeric vector

`ethlihom` a numeric vector

`DIST` a numeric vector

`close` a numeric vector

`DEMYEARS` a numeric vector

`demex` a numeric vector

`COMYEARS` a numeric vector

`nocom` a numeric vector

`parlia` a numeric vector

`PARTYNUMBER` a numeric vector

`efpahi` a numeric vector

`decent` a numeric vector

## Source

This data set is kindly provided by Prof Schneider.

## References

Schneider, 2009, "The consolidation of democracy: comparing Europe and Latin America". Routledge.

## Examples

```
data(CoD)
# Examples from Chapter 7
### table 6.2 in p77
round(apply(CoD[,c("econdev","eduhi","ethlihom","close","demex","nocom","parlia","efpahi","decent")],2,FUN=fun
round(apply(CoD[,c("econdev","eduhi","ethlihom","close","demex","nocom","parlia","efpahi","decent")],2,FUN=fun
round(apply(CoD[,c("econdev","eduhi","ethlihom","close","demex","nocom","parlia","efpahi","decent")],2,FUN=fun
round(apply(CoD[,c("econdev","eduhi","ethlihom","close","demex","nocom","parlia","efpahi","decent")],2,FUN=fun
round(apply(CoD[,c("econdev","demex")],2,FUN=function(x) coverage(1-x,1-CoD$cod,'great')),3)
round(consistency(apply(1-CoD[,c("econdev","demex")],1,min),1-CoD$cod,'great'),3)
round(coverage(apply(1-CoD[,c("econdev","demex")],1,min),1-CoD$cod,'great'),3)

## QCA of remote conditions
reduce(CoD,'cod',c('econdev','ethlihom','close','demex','nocom'),'positive','include',case='label',pre='fs_tru

## consitency of remote conditions as sufficent condition (table 6.4 in p82)
round(apply(CoD[,c("econdev","ethlihom","nocom")],2,FUN=function(x) consistency(x,CoD$cod,'less')),3)
round(apply(CoD[,c("econdev","ethlihom","nocom")],2,FUN=function(x) coverage(x,CoD$cod,'less')),3)
round(consistency(apply(CoD[,c("econdev","ethlihom","nocom")],1,max),CoD$cod),3)
round(coverage(apply(CoD[,c("econdev","ethlihom","nocom")],1,max),CoD$cod),3)

## table 6.6 in page 86
reduce(CoD,'cod',c('parlia','efpahi','decent'),'positive','include',case='label',pre='fs_truthTable')
reduce(CoD,'cod',c('econdev','ethlihom','nocom','parlia','efpahi','decent'),'positive','include',case='label',
## table 6.7 in pp87-88
fs_truthTable(CoD,'cod',c('econdev','ethlihom','nocom','parlia','efpahi','decent'),case='label')
```

`coincid` *Produces a coincidence matrix*

## Description

Produce a coincidence matrix from a crip set or fuzzy set data.

## Usage

```
coincid(x, standardized = FALSE, use = c("complete", "pairwise"))

## S3 method for class 'coincid'
print(x, digits = 3, ...)
```

## Arguments

| | |
|---|---|
| `x` | For coincide, x is a dataframe of crip set or fuzzy set data, which ranges from 0 to 1. |
| `standardized` | When TRUE, produce the standardized coincidence scores. |
| `use` | The method of handling missing data. "complete" means listwise deletion and "pairwise" means pairwise deletion. |
| `digits` | a non-null value for 'digits' specifies the minimum number of significant digits to be printed in values. |
| `...` | arguments passed to default method of print. |

## Details

Coincidence measures the amount of overlap or coincidence between two sets or configurations. The formula is: `sum(pmin(x,y))/sum(pmax(x,y))`. When standardized is TRUE, the formular is: `sum(pmin(x,y))/min(sum(x),sum(y))`.

## Value

A coincidence matrix.

## Author(s)

Ronggui HUANG

## References

Ragin, Charles C. 2006. "Set Relations in Social Research: Evaluating Their Consistency and Coverage." Political Analysis 14 (3) : 291-310

## See Also

`suffnec`

constrReduce *Impose constraints on a QCA solution*

## Description

To impose constraints on a QCA solution and returns a new QCA solution.

## Usage

```
constrReduce(object, exclude = NULL, include = NULL,necessary = NULL)

excludeCSA(object, csa)
```

## Arguments

| | |
|---|---|
| object | An object of class "QCA". |
| exclude | A data frame, each row represent one configuration. |
| include | A data frame, each row represent one configuration. |
| necessary | A list, specifying the necessary conditions. |
| csa | an object returned by CSA. |

## Details

constrReduce allows you to include a set of configurations to the a solution, or exclude a set of configurations from a solution, then return a new solution.

excludeCSA conducts Boolean minimization including other remainders except those in condictory simplifying assumptions.

The difference between constrReduce and excludeCSA mainly lies in how to deal with other remainders when imposing constraints on a QCA solution. constrReduce does not include further remainders, while excludeCSA does.

Sometime, you may encounter contraditory simplifying assumptions. In that case, you may want to exclude the CSAs to attain a more reasonable solution. In this case, excludeCSA is the most suitable way to go, which can make QCA easier. However, it does not guarantee a final solution, in particular when there are multiple solutions to both positive and negative outcome.

Sometimes, you may attain a solution without including all the remainders, latter you may want to include a small number of remainders in order to get a intermediate solution. You may include raminders in Boolean minimization, but some of the simplifying assumptions (see SA) are not feasible. You might want to exclude these unjustified simplifying assumptions without search for further simplifying assumptions. If you attain a solution without including remainders, you may want to include easy counterfactuals (justifiable remainders) to attain an intermediate solution. In either case, constrReduce is the most suitable way to go. See the example section on Ragin (2008: chapter 9) on this usage.

Usually, the exclude and include argument should repsent a set of configurations. In these case, there should be no '-9' in the data frame. However, it is NOT wrong to include -9 in the data frame. When there is -9, it means a multiple configurations because -9 denotes "dont' care".

## Value

For constrReduce, it is an object of class "QCA". It is esentailly a list of10 components. See reduce for more details. The only difference is the call componenet.

## Author(s)

Rongggui HUANG

## References

Ragin, Charles C. 2008. "Redesigning social inquiry: fuzzy sets and beyond." Chapter 9. Chicago: University of Chicago Press.

## See Also

reduce and CSA

## Examples

```
example(HuangGui2009)
newSol <- excludeCSA(ans2[2],CSA(ans1,ans2[2]))
## ans2 has 3 solutions, only the 3rd has not CSA.
identical(newSol$solutions, ans2[3]$solutions)
## they are the same.

if (require(QCA)){
##Use easy counterfactuals to get an intermediate solution (Ragin 2008:chapter 9)
data(Stokke,package="QCA")
comp <- reduce(Stokke,"Y",c("A","C","S","I","R"),explain="positive")
pars <- reduce(Stokke,"Y",c("A","C","S","I","R"),explain="positive",remaind="include")
sa <- SA(pars)
## determins easy counterfactuals
idx <- c(12,14,## A*S*R must be kept
         14, ## A*C*S*i must be kept
         3,5,8,10,12,14 ## A*i must be kept
         )
idx <- unique(idx) ## index of easy counterfactuals
easy <- sa$solution[[1]] [idx,]
difficult <- sa$solution[[1]] [-idx,]
constrReduce(comp,include=easy)
constrReduce(pars,exclude=difficult) ## the last two are equivalent.

## another way is to manually construct the easy counterfactuals
easy2 <- rbind(
c(1,-9,1,-9,1), # A*S*R
c(1,1,1,0,-9), # A*C*S*i
c(1,-9,-9,0,-9) # A*i
)
easy2 <- as.data.frame(easy2)
constrReduce(comp,include=easy2)
## end of Ragin (2009:chapter 9) example
```

```
## example 1
data(Yamasaki,package="QCA")
ans0 <- reduce(Yamasaki,"AGENDA",names(Yamasaki)[1:5],"negative","include") ## 5 solutions
ans1 <- reduce(Yamasaki,"AGENDA",names(Yamasaki)[1:5],"positive","include")
(csa2 <- CSA(ans0[2],ans1)) ## get and show CSAs
(ans02 <- constrReduce(ans0[2],exclude=csa2$solutions[[1]]))
## impose constraint
CSA(ans02,ans1) ## no CSA now

ans02b <- excludeCSA(ans0[2],csa2)
CSA(ans02b,ans1) ## no CSA
## note that ans02b is more parsimonious than ans02
## since ans02b includes further reminders.

## example 2
data(Osa,package="QCA") ## QCA package is required to run this example
conditions <- c("DYNA","ACCES","INFLU","ELITE","SOCIAL")
Osa$OUT2 <- Osa$OUT ## OUT is reserved word in QCA3.
a <- reduce(Osa, outcome = "OUT2", conditions = conditions, explain = "negative", remainders = "ex", contradictions
b <- reduce(Osa, outcome = "OUT2", conditions = conditions, explain = "negative", remainders = "include", contradic

# there are two solutions in b, let's focus on the first only.
b1 <- b[1]
sa <- SA(b1) ## simplifying assumptions
constrReduce(a,inc=sa$solutions[[1]]) ## == b1
constrReduce(b1,exc=sa$solutions[[1]]) ## ==a1
}
```

---

coverage *Coverage and consistency.*

---

## Description

Coverage and consistency.

## Usage

```
coverage(x, ...)

## Default S3 method:
coverage(x, y, alternative = c("less", "greater"),...)

## S3 method for class 'QCA'
coverage(x,traditional=TRUE,...)

consistency(x, y, alternative = c("less", "greater"))
```

## Arguments

| | |
|---|---|
| x | a numeric vector of fuzzy set score for default method of coverage and consistency. A QCA object for QCA method of coverage. |
| y | a numeric vector of fuzzy set score. |
| alternative | specify the relationship between x and y. |
| traditional | logical, use traditional symbol to present the solution when it is TRUE. Otherwise, use Tosmana-style symbol. |
| ... | not used currently. |

## Details

Given two fuzzy set score variables x and y, coverage returns the coverage of x by y, given their relationship; and consistency returns the consistency of x by y.

When alternative is "less", it returns coverage(x<=y). This is coverage for sufficent test. When "greater", it returns coverage(x>=y), it is coverage for necessary condition.

This value can be used to evaluate the importance or relevance of x as a necessary condition for Y. When coverage(x>=y) is small, the constraining effect of x on y is negligible. In other words, small value of coverage(x>=y) indicates en empirically trivial necessary condition (Ragin, 2008: 61).

## Value

For default method of coverage and consistency, the value is a length-1 numeric vector.

For QCA method of coverage, it is a list of data frame. Each data frame represent coverage of one solution. It contains three columns, namely implicant, raw and unique (coverage in terms of number of case).

## Author(s)

Ronggui HUANG

## References

Ragin, C. 2006. Set Relations in Social Research: Evaluating Their Consistency and Coverage Political Analysis,14(3):291-310.

Ragin, C. 2008, Redesigning social inquiry: fuzzy set and beyond. Chicago: The University of Chicago Press.

## Examples

```
## evaluate the congerage of necessary conditions for "no democratic consolidation"
## see help(CoD) for more information.
## coverage for necessary test for not-CoD (Schneider, 2009:77)

coverage(1-CoD$econdev,1-CoD$cod,'greater')
# not socioeconomically developed
coverage(1-CoD$demex,1-CoD$cod,'greater')
# no previous democratic experience
```

```
coverage(pmin(1-CoD$demex,1-CoD$econdev),1-CoD$cod,'greater')
# not socioeconomic developed society without prior democratic experience

## coverage method for QCA object
tb2<-reduce(policyChange~soc+pol+ind+are+foc,Yamasaki2009,cases="country")
coverage(tb2)
```

---

`CressSnow` *The Outcomes of Homeless Mobilization*

---

## Description

The Influence of organization, disruption, political mediation, and and framing on the outcome of homeless soical movement organizations.

## Usage

```
data(CressSnow)
```

## Format

A data frame with 14 observations on the following 11 variables.

`Caseid` The name of Social movement organizations.

`Viable` SMO visibility.

`Disrupt` Disruptive tactics.

`Allies` Sympathetic Allies.

`City` City Support.

`Diag` Diagnostic framming.

`Prog` Prognostic framming.

`Representation` Outcome of representation.

`Resources` Outcome of resources.

`Rights` Outcome of rights.

`Relief` Outcome of relief.

## References

Cress, Daniel M. and David A. Snow. 2000. "The Outcomes of Homeless Mobilization: The Influence of Organization, Disruption, Political Mediation, and and Framing." American Journal of Sociology 105 (4) : 1063-1104.

## Examples

```
cond <- c("Viable", "Disrupt","Allies","City","Diag","Prog")
## see table 5 in p1083
reduce(CressSnow,"Representation",cond,exp="positive",contr="positive",case="Caseid")
reduce(CressSnow,"Resources",cond,exp="positive",contr="positive",case="Caseid")
reduce(CressSnow,"Rights",cond,exp="positive",case="Caseid")
reduce(CressSnow,"Relief",cond,exp="positive",contr="positive",case="Caseid")
## define impact
CressSnow$Impact <- as.numeric(with(CressSnow,Representation+Rights+Relief)>=2)
reduce(CressSnow,"Impact",cond,exp="positive",case="Caseid")
```

---

`cs_truthTable` *Construct a truthTable for csQCA*

---

## Description

Construct a truthTable for csQCA. Both deterministic and probabilistic methods of determining configurations of positive, negative and contraditory outcome are implemented.

This function can be used for crip set TQCA as well. See `CarenPanofsky` for example. It needs manual construction of indicator conditions of temporal order.

## Usage

```
cs_truthTable(mydata, outcome, conditions,method = c("deterministic",
    "probabilistic"), weight = NULL, show.cases = TRUE, cases = NULL,
    cutoff1 = 1, cutoff0 = 1, benchmark = 0.65, conf.level = 0.95,
    missing = c("missing", "dontcare", "positive", "negative"))
```

## Arguments

| | |
|---|---|
| `mydata` | data frame of the raw data. |
| `outcome` | character, the name of the outcome variable in mydata. |
| `conditions` | character vector, the name of the conditions from mydata. |
| `method` | character, specifying the method of determining the outcome of a configuration. |
| `weight` | character, name of a variable specifying the weights. |
| `show.cases` | logical, when TRUE the result shows case names. |
| `cases` | character, variable specifying the case names. When it is NUll, then use row names of mydata as case names. |
| `cutoff1` | length one numeric vector. |
| `cutoff0` | length one numeric vector. |
| `benchmark` | Benchmark for statistical test. Must equal or greater than 0.5. |
| `conf.level` | confident level of statistical test. |
| `missing` | method to handle missing data. |

## Details

The value of all the conditions should start from 0. In cript set QCA, it is always be 0 or 1. Value -9 in conditions means "don't care" (though "don't care" in outcome is denoted by "-9").

Symbols of the outcome. '1' is postive configuration, '0' is negative configuration, 'C' is contraditory configuration, "?" is unobserved configuration and '-9' is don't care configuration. When show.case is TRUE and a configuration is 'C', then the name of case with negative outcome is highlighted by [].

'cutoff1' and 'cutoff0' are only meaningful for'deterministic' method. They represent cutting point of positive case and negative case. When a configuration has positive case only and the number of cases is equal or greater than the cutting point, then it is regared as positive outcome, otherwise as don't care outcome. Similarly, When a configuration has negative case only and the number of cases is equal or greater than the cutting point, then it is regared as negative outcome, otherwise as don't care outcome. If a configuration has both positive case and nagetive case, the number of each type of cases will be compared with the corresponding cutting point. If only one type of case have enough case (number of cases greater than cutting point), that configuration is regarded as that type. If both types have enough case, it is contraditory configuration. If neither type has enough case, it is don't care configuration.

The caculation of cutting point: if it is equal or greater than 1, the cutting point is the value of cutoff1 and cutoff0. If it is between 0 and 1, then the cutting point is the cutoff1/cutoff0 multiplied by the total number of case.

'benchmark' and 'conf.level' are only meaningful for 'probabilistic' method. When the method is 'probabilistic', a statistical test will conducted to test if the proportion of case for a configuration is greater then a benchmark. If the proportion of cases with outcome '1' is greater than benchmark, then the it is a configuratin with outcome '1'. If the proportion of case with outcome '0' is greater than benchmark, then the configuration with outcome of '0'. If neither proportion fits the criterion, then it is don't care configuration. There is no contraditory congfiguration in this method, as it is designed to handle with contraditory configurations.

rownames of a truthTable is grouping index (not important for end-users).

## Value

An object of class "truthTable" and "cs_trutbTable". A list with 5 components:

| | |
|---|---|
| `truthTable` | a data frame presenting a truthTable. |
| `outcome` | The name of outcome variable. length-1 character. |
| `conditions` | conditions. A character vector. |
| `nlevels` | integer vector specifying number of levels of each condition. |
| `call` | the matched call. |

## Author(s)

Ronggui HUANG

## References

Ragin, Charles. 2000. Fuzzy-Set Social Science. Pp109-116. University Of Chicago Press.

## See Also

fs_truthTable, mv_truthTable and reduce

## Examples

```
## truthTable for csQCA
cs_truthTable(Lipset_cs,"SURVIVAL", c("GNPCAP", "URBANIZA", "LITERACY",
  "INDLAB", "GOVSTAB"),case="CASEID")
```

---

# directCalibration *Calibration in fuzzy set QCA.*

---

## Description

Direct method of calibration in fuzzy set QCA.

## Usage

```
directCalibration(x, fullin, fullout, crossover, infz = 0.953, outfz = 0.047, details = FALSE)
```

## Arguments

| | |
|---|---|
| x | an interval variable to be converted to fuzzy set score. |
| fullin | scalar indicating full membership. |
| fullout | scalar indicating the full non-membership. |
| crossover | scalar indicating cross over point. |
| infz | fuzzy set score of full membership, corresponding to fullin. |
| outfz | fuzzy set score of full non-membership, corresponding to fullout. |
| details | logical. |

## Details

Usually, the sum of infz and outfz should be 1. In Ragin (2008), infz is 0.953 and outfz is 0.047, which lead to the log odd of membership being (proximitely) 3.0 and -3.0.

## Value

When details is TRUE, it the result is a data frame. Otherwise, the value is a numeric vector of fuzzy set score.

## Author(s)

Ronggui HUANG

## References

Ragin, Charles C. 2008. "Redesigning social inquiry: fuzzy sets and beyond." Chapter 5. Chicago: University of Chicago Press.

## Examples

```
directCalibration(Lipset_fs$Developed,800,300,550)
```

---

`fsnot` *The not opertor for a fuzzy set score.*

---

## Description

The not opertor for a fuzzy set score.

## Usage

```
fsnot(x)
```

## Arguments

| | |
|---|---|
| x | a variable presenting fuzzy set score |

## Details

The not operator is define as 1 - x.

## Value

a numeric vector of the new fuzzy set score.

## Author(s)

Ronggui HUANG

## Examples

```
x = c(0.1,0.5,0.6,0.8)
fsnot(x)

## can be used with fsplot
fsplot(cod~fsnot(econdev),data=CoD)
```

`fsplot` *Fuzzy set plot.*

## Description

Fuzzy set plot facilitating assessment of a causal recipe.

## Usage

```
fsplot(formula, data, main = "fuzzy set plot", xlab = NULL, ylab = NULL,...)
```

## Arguments

| | |
|---|---|
| `formula` | a formula like out ~ condition1 + condition2 + conditions3, with out being the outcome and condition1, condition2 and conditions3 are conditions in a causal recipe. |
| `data` | a data frame. |
| `main` | a title of the plot. |
| `xlab` | a title for the x axis |
| `ylab` | a title for the y axis |
| `...` | other arguments passed to `plot`. |

## Details

This function is useful for exploratory analysis. If a recipe is a causal condition of the outcome, then the fuzzy set score should be consistently less or equal than that of outcome. That is, if the recipe does not occur, the outcome may occur; however, when the recipe occurs, the outcome MUST occurs. It suggests that if a recipe can be interpreted as causal condition, the points scatter above the diagonal by and large. The set-theoretic consistency measures this consitency. High consistency score endors the causal interpretation.

In addition, high consistency permits interpretation of coverage score. This score indicates the proportion of membership in outcome accounted for by the recipe. It should noted that it is harzardous to interpret the coverage measure when consistency score is low.

## Author(s)

Ronggui HUANG

## References

Ragin. C. 2008. "Configurational Thinking." Chapter 6 of "Redesigning Social Inquiry." The University of Chicago Press.

## See Also

See Also as `coverage`, `consistency`

## Examples

```
fsplot(cod~econdev,data=CoD)
fsplot(cod~eduhi,data=CoD) ## low consistency, cautious when interpret coverage
fsplot(cod~econdev+eduhi+ethlihom+close,data=CoD)
```

---

`fs_truthTable` *Construction of truthTable from fuzzy set score*

---

## Description

Constructing a truthTable from fuzzy set score.

## Usage

```
fs_truthTable(mydata, outcome, conditions, ncases_cutoff = 1,
              consistency_cutoff = 0.8,
              show.cases =TRUE,quiet =FALSE, cases = NULL, ...)
```

## Arguments

| | |
|---|---|
| `mydata` | A fuzzy set score dataset. All the scores must range from 0 to 1. |
| `outcome` | character, the name of the outcome variable in the dataset. |
| `conditions` | character vetor, the name of the conditions from the dataset. |
| `ncases_cutoff` | When number of case is less than cutoff, it will be regarded as dontcare configuration. |
| `consistency_cutoff` | Cutoff point of consistenty score, cases with consistency score greater than cutoff point are regarded as OUT=1. |
| `show.cases` | show the rownames from the original dataset for each combination of conditions. |
| `quiet` | Not used currently. |
| `cases` | character, variable of case names. If it is NULL and show.cases is TRUE, name of cases are derived from row names of dataset. |
| `...` | Not used currently. |

## Details

There are serveral pillars which make it possible to construct a crip truthTable summarizing the raw data. There is a correspondance between vector space corners and truthTable rows. Thus, it is possible to get the number of cases with strong membership in each corner (usually greater then 0.5), and the consistency of the empirical evidence for each corner. By specifying the frequency thresholds for fuzzy-set assessments (the `ncases_cutoff` argument), and assessing the consistency of fuzzy-set subset relations (the `consistency_cutoff` argument), we can finally construct a truthTable.

There is a sort method for the returned object.

### Value

An object of class "truthTable" and "fs_truthTable".

### Author(s)

Ronggui HUANG

### References

Ragin. Charles. 2009. Qualitative Comparative Analyais Using Fuzzy Sets (fsQCA). In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage. This chapter can be downloaded from http://www.u.arizona.edu/~cragin/fsQCA/software.shtml.

### See Also

reduce, cs_truthTable and fs_truthTable

### Examples

```
fs_truthTable(Lipset_fs,"Survived.FZ",c("Developed.FZ","Urban.FZ","Literate.FZ","Industrial.FZ",
"Stable.FZ"),cases="Country",consistency_cutoff=0.7)
```

---

## GiugniYamasaki2009 *The Policy Impact of Social Movements*

---

### Description

This data is used to study the policy impact of social movements.

### Usage

```
data(GiugniYamasaki2009)
```

### Format

A data frame with 30 observations on the following 6 variables.

`Case` a character vector

`Protest` a numeric vector

`Opinion` a numeric vector

`Allies` a numeric vector

`Parties` a numeric vector

`Alliances` a numeric vector

`PolicyChange` a numeric vector

## Details

This data was collected from a previous study on the policy impact of antinuclear,ecology, and peace movements in three countries. Giugni and Sakura replicated the analysis with QCA.

The example section also demonstrates the advantages of using R to do QCA, because it has strong capacity of data manipulation.

## References

Giugni, Marco and Sakura Yamasaki. 2009. "The Policy Impact of Social Movements: A Replication Through Qualitative Comparative Analysis." Mobilization: An International Quarterly 14 (4) : 467 - 484.

## Examples

```
## ecology movement
ecology <- GiugniYamasaki2009[grep("-E",GiugniYamasaki2009$Case),]
ecology$opinion <- as.numeric(ecology$Opinion>63)
ecology$alliances <- as.numeric(ecology$Alliances>127)
cs_truthTable(ecology,"PolicyChange",c('Protest','opinion','alliances'),case="Case")
reduce(PolicyChange~Protest+opinion+alliances,ecology,case="Case")

## anti-nuclear movement
nuclear <- GiugniYamasaki2009[grep("-N",GiugniYamasaki2009$Case),]
nuclear$opinion <- as.numeric(nuclear$Opinion>51)
nuclear$alliances <- as.numeric(nuclear$Alliances>127)
reduce(PolicyChange~Protest+opinion+alliances,nuclear,case="Case")

## Peace movement
peace <- GiugniYamasaki2009[grep("-P",GiugniYamasaki2009$Case),]
peace$opinion <- as.numeric(peace$Opinion>40)
peace$alliances <- as.numeric(peace$Alliances>127)
reduce(PolicyChange~Protest+opinion+alliances,peace,case="Case")
```

---

# Hino2009 *Example data of Time-series QCA.*

---

## Description

Example data in Hino (2009), demonstrating the use of time-serious QCA.

## Usage

```
data(Hino2009)
```

### Format

A data frame with 30 observations on the following 5 variables.

`ID` a factor with levels `A80` `A90` `B80` `B90` `D80` `D90` `FI80` `FI90` `FR80` `FR90` `G80` `G90` `IC80` `IC90` `IR80` `IR90` `IT80` `IT90` `L80` `L90` `NR80` `NR90` `NT80` `NT90` `SWE80` `SWE90` `SWI80` `SWI90` `UK80` `UK90`

`VOTE` a numeric vector

`FOREIGN` a numeric vector

`UNEMP` a numeric vector

`DIVERT` a numeric vector

### References

Hino, Airo 2009. "Time-Series QCA: Studying Temporal Change through Boolean Analysis." Sociological Theory and Methods 24 (2) : 247-265.

### Examples

```
data(Hino2009)
Hino2009$COUNTRY <- gsub("[0-9]*","",Hino2009$ID)
Hino2009$TIME <- gsub("[A-Z]*","",Hino2009$ID)
t2 <- QCA3:::timeDiff(Hino2009,Hino2009$COUNTRY,Hino2009$TIME) ## table 2
t3 <- QCA3:::tsData_tdiff(Hino2009,Hino2009$COUNTRY,Hino2009$TIME)
t3$PRES80s <- 1
t3$PRES80s[match(c("FI","G","SWE","IC","L"),t3$COUNTRY)] <- 0
t3$CONV <- 1 - t3$DIVERT
## manually add PRES80s
rownames(t3) <- t3$ID ## used as case ID.
tt <- cs_truthTable(t3,"VOTE",c("FOREIGN","UNEMP","CONV","PRES80s"))
reduce(tt,expl="positive") ## formula (5)
reduce(tt,expl="positive",remainder="include")## formula (6)
reduce(tt,expl="negative") ## ## formula (7)
reduce(tt,expl="negative",remainder="include") ## ## formula (8)
```

---

## HIVChange *data about HIV prevalence in Sub-Saharan Africa*

---

### Description

Data set from Cronqvist and Berg-Schlosser(2006), examining the HIV prevalence in Sub-Saharan Africa.

### Usage

```
data(HIVChange)
```

## Format

A data frame with 12 observations on the following 7 variables.

`LIT00` Dichotomized literacy rate of 2000. 1 if above 50%.

`GENDEREQ` Dichotomized gender equality index. 1 is above 40 index point.

`MORTALITY` Trichotomized cummulated HIV morality rate up to 1997, with thresholds of 2% and 4%.

`AGRARGDP` Dichotomized variable of share of agarian production of GDP. 1 is above 25%.

`HIVChange` The outcome. "1" if the prevalence rate increases. Otherwise it is "0". "C" is indicates contraditory configurations.

`Country` Names of cases

`NCase` Number of cases

## Details

`LIT00` measures socio-economic factors. `GENDEREQ` measures the situation of women. `MORTALITY` measures the awareness of HIV threat. `AGRARGDP` measures the impact of migration.

## References

Cronqvist. L. and Berg-Schlosser, D. 2006. Determining The Conditions Of Hiv/Aids Prevalence In Sub-Saharan Africa: Employing New Tools Of Macro-Qualitative Analysis. In Innovative Comparative Methods For Policy Analysis: Beyond The Quantitative-Qualitative Divide. Benoit Rihoux and Heike Grimm (Eds).Springer.

## Examples

```
## manually construct a truthTable for mvQCA
conditions <- c("LIT00", "GENDEREQ", "MORTALITY", "AGRARGDP")
HIVChange$OUT <- HIVChange$HIVChange
HTT <- HIVChange[,c(conditions,"OUT","NCase","Country")]
names(HTT) <- c(conditions,"OUT","NCase","Cases")
## optional
mvTT <- list(truthTable=HTT,nlevels=c(2,2,3,2),conditions=conditions)
## only some components are rquired
class(mvTT) <- "truthTable"

## Example in p161: not exactly the same solution.
##    This one is correct too (tell me if you don't think so)
reduce(mvTT,expl="pos",remainder="include",contr="negative")
reduce(mvTT,expl="neg",remainder="include",contr="positive")

## Example in p163
reduce(mvTT,expl="pos",remainder="include",contr="negative")
reduce(mvTT,expl="neg",remainder="include",contr="negative")
## C.A.R is positive, all other three are nagetive
```

HuangGui2009 *data about on-line forum and homeowners' collective resistance*

## Description

Dataset used in Huang and Gui's (2009) study of Internet and homeowners' collective resistance.

## Usage

```
data(HuangGui2009)
```

## Format

A data frame with 15 observations on the following 6 variables.

`network` stock of social ties.

`big` scale of neighborhood.

`grievance` Existence of grevance.

`ha` Existence of Homeowners' Association.

`BBS` Existence of homeowners' onl-line forum.

`outcome` outcome, existence of collective resistance.

`code` id of neighborhood.

## Source

HUANG Ronggui, GUI Yong, 2009, Internet and homeowners' collective resistance: a comparative qualitative analysis in Shanghai". Sociological Research.(5):29-56.(In Chinese)

## Examples

```
data(HuangGui2009)
reduce(HuangGui2009,"outcome",c("network","big","grievance","ha","BBS"),"pos",case="code")
ans1 <- reduce(HuangGui2009,"outcome",c("network","big","grievance","ha","BBS"),"pos",case="code",rem="include"
SA(ans1)
reduce(HuangGui2009,"outcome",c("network","big","grievance","ha","BBS"),"neg",case="code")
ans2 <- reduce(HuangGui2009,"outcome",c("network","big","grievance","ha","BBS"),"neg",case="code",rem="incl")
CSA(ans1,ans2[3])
SA(ans2[3])
```

KatzHauMahoney *The "great reversal" data set.*

## Description

Data set from Kate, Hau and Mahoney (2005) to explain the "great reversal" in Spanish America.

## Usage

```
data(KatzHauMahoney)
```

## Format

A data frame with 14 observations on the following 12 variables.

`Country` a factor with levels `Argentina` `Bolivia` `Chile` `Columbia` `CostaRica` `Ecuador` `ElSalvador` `Guatemala` `Honduras` `Mexico` `Nicaragua` `Paraguay` `Peru` `Uruguay` `Venezuela`

`EDF` Economically Developed Country. The fuzzy-set outcome.

`EUDF` Economically Underdeveloped Country. The fuzzy-set outcome

`GDP` GDP in 1990. Statistical outcome

`SDF` Socially Developed Country. The fuzzy-set outcome

`SUDF` Socially Underdeveloped Country. The fuzze-set outcome

`SDI90` Socially Development Index in 1990. Statistical outcome

`DIPF` Dense Indigenous Population, fuzzy-set score

`LIEF` Labor-Intensive Estates, fuzzy-set score

`MTEF` Mineral/Tropical Exports, fuzzy-set score

`SLF` Strong Liberals, fuzzy-set score

`SCF` Strong Conservatives, fuzzy-set score

## Details

This study tries to explain the "great reversal",in Spanish America. The great reversal refers to the fact that "from 1750 to 1900, the most marginal colonial territories often became the region's wealthiest countries, whereas the most central colonial territories often became the region's poorest countries."

## Source

Katz, A.; vom Hau, M. & Mahoney, J. Explaining the Great Reversal in Spanish America: Fuzzy-Set Analysis Versus Regression Analysis Sociological Methods and Research, 2005, 33 (4) , 539-573

## Examples

```
data(KatzHauMahoney)
```

Lipsets *Breakdown/survival of democracy in inter-war Europe*

## Description

This is the fuzzy set version of Lipset data on Breakdown/survival of democracy in inter-war Europe.

## Format

A data frame with 18 observations on the following 13 variables.

`Country` Name of country

`Survived` If a country survives the economic and political upheavals of this period.

`Survived.FZ` Fuzzy set score of Survived

`Developed` The degree of development.

`Developed.FZ` Fuzzy set score of Developed.

`Urban` The degree of urbanization.

`Urban.FZ` Fuzzy set score of Urban.

`Literate` The degree of literate a county is.

`Literate.FZ` Fuzze set score of Literate.

`Industrial` The degree of industrialization.

`Industrial.FZ` Fuzze set score of Industrial

`Unstable` The degree of political instablity.

`Stable.FZ` Fuzzy set score of political stability.

## Details

The data set is from Ragin(2009:95), the details about the dataset is described in page 93.

## Source

Ragin. Charles. 2009. Qualitative Comparative Analyais Using Fuzzy Sets (fsQCA). In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage.

## Examples

```
conditions <- c("Developed.FZ","Urban.FZ","Literate.FZ","Industrial.FZ", "Stable.FZ")
reduce(Lipset_fs,"Survived.FZ",conditions,explain="positive",remaind="exclude",prepro="fs",consistency=0.7)
## Formula 1 in page 112
reduce(Lipset_fs,"Survived.FZ",conditions,explain="positive",remaind="include",prepro="fs",consistency=0.7)
## Formula 2 in page 114
reduce(Lipset_fs,"Survived.FZ",conditions,explain="negative",remaind="exclude",prepro="fs",consistency=0.7)
## Formula 5 in page 115
reduce(Lipset_fs,"Survived.FZ",conditions,explain="negative",remaind="include",prepro="fs",consistency=0.7)
## Formula 6 in page 117
```

`mv_truthTable` *Construct a truthTable for mvQCA*

## Description

Construct a truthTable for mvQCA. Both deterministic and probabilistic methods of determining configurations of positive, negative and contraditory outcome are implemented.

## Usage

```
mv_truthTable(mydata, outcome, conditions, method = c("deterministic",
    "probabilistic"), weight = NULL, show.cases = TRUE, cases = NULL,
    cutoff1 = 1, cutoff0 = 1, benchmark = 0.65, conf.level = 0.95,
    missing = c("missing", "dontcare", "positive", "negative"))
```

## Arguments

| | |
|---|---|
| `mydata` | data frame of the raw data. |
| `outcome` | character, the name of the outcome variable in mydata. |
| `conditions` | character vector, the name of the conditions from mydata. |
| `method` | character, specifying the method of determining the outcome of a configuration. |
| `weight` | character, name of a variable specifying the weights. |
| `show.cases` | logical, when TRUE the result shows case names. |
| `cases` | character, variable specifying the case names. When it is NUll, then use row names of mydata as case names. |
| `cutoff1` | length one numeric vector. |
| `cutoff0` | length one numeric vector. |
| `benchmark` | Benchmark for statistical test. Must equal or greater than 0.5. |
| `conf.level` | confident level of statistical test. |
| `missing` | method to handle missing data. |

## Details

The value of all the conditions should start from 0. For example, for mvQCA with 3 levels,it should be 0,1 or 2. Value -9 in conditions means "don't care" (though "don't care" in outcome is denoted by "-9").

Refer to `cs_truthTable` for for more details.

## Value

An object of class "truthTable" and "mv_truthTable".

### Author(s)

Ronggui HUANG

### References

Ragin, Charles. 2000. Fuzzy-Set Social Science. Pp109-116. University Of Chicago Press.

### See Also

fs_truthTable and reduce

### Examples

```
## truthTable for mvQCA.
mv_truthTable(Lipset_mv,"SURVIVAL", c("GNPCAP", "URBANIZA", "LITERACY",
              "INDLAB"),case="CASEID")
```

---

## QCA.methods *Methods for "QCA" an object*

---

### Description

Various methods for object from reduce

### Usage

```
## S3 method for class 'QCA'
plot(x,...)

## S3 method for class 'QCA'
print(x, traditional = TRUE, show.truthTable = TRUE, ...)

## S3 method for class 'QCA'
summary(object, traditional = TRUE, show.case = TRUE, ...)

## S3 method for class 'QCA'
object[which]

## S3 method for class 'QCA'
update(object,...,evaluate = TRUE)
```

### Arguments

| | |
|---|---|
| x | an object of class 'QCA', which is usually returned from reduce. |
| traditional | logical, use traditional symbol when it is TRUE. Otherwise, use Tosmana-style symbol. |

`show.truthTable`
: logical, show truthTable when it is TRUE. Of course, it has effect only when the 'keepTruthTable' argument of `reduce` is set to TRUE.

`object`
: an object of class 'QCA', which is usually returned from `reduce`.

`show.case`
: logical, show case names when it is TRUE.

`which`
: numeric vector, indices specifying elements to extract. Extraction of a solution or (prime implicant) is essentially a extraction on a list. you can refer to `[` for more details.

`...`
: For `print.QCA` and `summary.QCA`, currently not used. For `update`, additional arguments to the call, or arguments with changed values. Use 'name=NULL' to remove the argument 'name'.

`evaluate`
: when TRUE, return the evaluated result which is an object of QCA class. Otherwise, it returns the call.

## Details

The traditional way uses upper-case letters representing 1 and and lower-case letters reprensenting 0. The Tosmana-style uses `condition{value}` to represent the prime implicants.

## Value

print method does not return any value.

summary method returns an object of class "summary.QCA".

The index method returns an object of class "QCA".

update method returns a new "QCA" object if evaluate is TRUE, the call if FALSE.

## Author(s)

Ronggui HUANG

## See Also

`reduce` and `constrReduce`

---

# `QCA.Notation` *Notations used in QCA3 package*

---

## Description

This page describes some notations used in QCA3 package

## Details

-9: don't case case.

*: boolean operator of and.

+: boolean operator of or.

cond{v}: condition has a value of v.

A/B: condition A happend before B.

NA: missing data.

---

reduce *Boolean miniziation for csQCA, mvQCA and fsQCA*

---

## Description

This is the core funtion for QCA (Qualitative Comparative Analysis). Given the outcome and conditions, it returns an object of class 'QCA', which contains all the possible configurations leading to the outcome. It can handle various kinds of QCA., namely csQCA, mvQCA, fsQCA and csTQCA.

## Usage

```
reduce(x,...)

## default method is an alias of truthTable method.

## S3 method for class 'truthTable'
reduce(x, explain = c("positive", "negative"),
       remainders = c("exclude","include"),
       contradictions = c("remainders","positive","negative"),
       dontcare = c("remainders", "positive", "negative"),
       keepTruthTable = TRUE,...)

## S3 method for class 'data.frame'
reduce(x, outcome, conditions,
        explain = c("positive", "negative"),
        remainders = c("exclude", "include"),
        contradictions = c("remainders", "positive", "negative"),
        dontcare = c("remainders", "positive", "negative"),
        preprocess = c("cs_truthTable", "fs_truthTable",
        "mv_truthTable"),
       keepTruthTable = TRUE, ...)

## S3 method for class 'formula'
reduce(x, data, explain = c("positive", "negative"),
      remainders = c("exclude", "include"),
      contradictions = c("remainders", "positive", "negative"),
      dontcare = c("remainders", "positive", "negative"),
```

```
preprocess = c("cs_truthTable", "fs_truthTable", "mv_truthTable"),
keepTruthTable = TRUE, ...)
```

## Arguments

| | |
|---|---|
| x | a R object, it could be a truthTable, data frame or a formula |
| outcome | a character string to specify the outcome |
| data | a data frame, which is not optional. |
| conditions | a character vector to specify the conditions |
| explain | a character string specifying the cases to be explained. Must one of "positive" or "negative". |
| remainders | a character string specifying how to deal with remainders. Must one of "exclude" or "include". |
| contradictions | a character string specifying how to deal with contraditory configurations. Must one of "remainders","positive" or "negative" |
| dontcare | a character string specifying how to deal with dontcare cases. Must one of "remainders", "positive", "negative" |
| preprocess | a character string specifying the function for preprocessing data, which turns raw data to a truthTable. Must one of `cs_truthTable`, `fs_truthTable` or `mv_truthTable`. |
| keepTruthTable | logical, when TRUE the returned object keeps the truthTable |
| ... | other arguments passed to a function. |

## Details

Outcome is the variable to be explained by the conditions. Conditions is explanatory variables that may affect the outcome. It is not "independent variable" in statistical sense. Configuration is a combination of conditions relevant to a given outcome. Remainders are configurations that lack empirical instances. Conraditory configuration is a configuration whose outcome value is positive[1] for some cases and negative[0] for other cases.

It is good practices to attain the solutions for both positive outcome with and without remainders, and negative outcome with and without remainders. If a common necessary condition appears in both solutions for positive and negative outcome (without remainders), then such necessary condition is a trivial necessary condition(Caramani, 2009:62). It is not necessary to include trivial necessary condition in the final solutions.

It is good practices to generate and examine a truthTable, then use truthTable method of reduce to do the boolean minimization.

Since version 0.0-3, reduce uses enhanced internal function ereduce1 (which uses enhanced internal function esubset). It has been tested and yields the same result (see tests directory for details).

## Value

An object of class "QCA". It is essentailly a list of 10 components.

| | |
|---|---|
| solutions | a list of data.frame, each data frame represents one solution. |

`commonSolutions`
: A list of lenth nrow(solutionsIDX). For each row of solutionsIDX, if the primeImplicants index are the same for all solutions, then the index is return. Otherwise, it is NULL.

`solutionsIDX`
: a matrix. Each column represents one solution. The number of the matrix is row index of primeImplicants.

`primeImplicants`
: A matrix of prime implicants.

`truthTable`
: a truthTable if keepTruthTable is TRUE, otherwise NULL.

`explained`
: A data frame, representing the configuration of conditions for explained cases. Note it is not on basis of case but basis of configuration.

`idExclude`
: integer vector. id of observed configurations that are excluded from minimization. The meaning of id is equivalent to the line number of a configuration discussed in Dusa (2007).

`nlevels`
: a integer vector, the number of levels of each condition.

`PIChart`
: a prime implicants charts, constructed according to primeImplicants and explained. It is a logic matrix with dimension of nrow(primeImplicants)x ncol(explained). It is TRUE if the corresponding primeImplicant covers the corresponding explained.

`call`
: the matched call.

## Note

With a 2.4 GHz and 4.0GB PC, it takes about about 0.5 minute for 12 conditions, 2 minutes for 13 conditions, 4.5 minutes for 14 conditions, and 9 minutes for 15 conditions. It may take a long time to get the solution when there are more conditions. You may use `eqmcc` if speed becomes an issue for `reduce`. The disparity is due to the fact that `eqmcc` eliminates redundant PIs before solving the PIChart (Thanks Adrian for pointting it out), but `reduce` does not. `reduce` is a bit greedy in terms of memory usage, for 15 conditions, it uses proximately 500 to 600 Mb memory in typical QCA study. I emphasis "typical" because the exact scenario also depends on the number of observed configurations.

## Author(s)

Ronggui HUANG

## References

Caramani, Daniele. 2009. "Introduction to the comparative method with Boolean algebra." Sage.

Cronqvist, Lasse and Berg-Schlosser, Dirk. 2009. Multi-Value QCA (mvQCA). In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage.

Ragin. Charles. 2009. Qualitative Comparative Analyais Using Fuzzy Sets (fsQCA). In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage.

Rihoux, Benoit and De Meur, Gisele. 2009. Crip-Set Qualitative Comparative Analysis (csQCA). In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage.

Dusa, Adrian. 2007. Enhancing Quine-McCluskey, http://www.compasss.org/files/WPfiles/Dusa2007a.pdf

Ragin, Charles. 2000. Fuzzy-Set Social Science. University Of Chicago Press.

Ragin, Charles. 1987. The Comparative Method. Moving beyond qualitative and quantitative strategies. University of California Press.

## See Also

factorize, SA, CSA, constrReduce

## Examples

```
if (require(QCA)){
data(Osa,package="QCA")
## QCA package is required to run this example
Osa$OUT2 <- Osa$OUT ## OUT is reserved word in QCA3
## the same as examples of QCA:::qmcc
conditions <- c("DYNA","ACCES","INFLU","ELITE","SOCIAL")
reduce(Osa,"OUT2",conditions,explain="positive",remaind="exclude")
reduce(Osa,"OUT2",conditions,explain="positive",contradictions="positive",remaind="include")
ans <-
 reduce(Osa,"OUT2",conditions,explain="positive",contradictions="negative",remaind="include")
simplifyingAssumption(ans) ## or SA(ans)
reduce(Osa,"OUT2",conditions,explain="negative",contradictions="negative",remaind="include")

## Results of Osa and Corduneanu-Huci (2003)
reduce(Osa,"OUT2",conditions,explain="pos",contrad="neg",remaind="exclude")
# table 1 in page 617
reduce(Osa,"OUT2",conditions,explain="neg",contrad="pos",remaind="exclude")
# table 2 of page 621
reduce(Osa,"OUT2",conditions,explain="positive",contradictions="pos",remaind="incl")
# maximum reduction in page 623
reduce(Osa,"OUT2",conditions[1:4],explain="pos",contradictions="neg",remaind="excl")
# Appendix 2 in page 629
}

## csQCA, mvQCA and fsQCA examples from "Configuraional comparative Methods"
## csQCA
conditions <- c("GNPCAP", "URBANIZA", "LITERACY", "INDLAB", "GOVSTAB")
reduce(Lipset_cs,"SURVIVAL",conditions,explain="positive",remainder="exclude",case="CASEID")
## or use formula
reduce(SURVIVAL~GNPCAP+URBANIZA+LITERACY+INDLAB+GOVSTAB,Lipset_cs,
       explain="positive",remainder="exclude",case="CASEID")
## Formula 1 in Rihoux and De Meur(2009:57)
reduce(Lipset_cs,"SURVIVAL",conditions,explain="negative",remainder="exclude",case="CASEID")
## Formula 3 in Rihoux and De Meur(2009:59)
ans1 <-
 reduce(Lipset_cs,"SURVIVAL",conditions,explain="positive",remainder="include",case="CASEID")
```

```
print(ans1) ## Formula 4 in Rihoux and De Meur(2009:60)
SA(ans1) ## 5 simplifying assumptions in p61
ans0 <-
 reduce(Lipset_cs,"SURVIVAL",conditions,explain="negative",remainder="include",case="CASEID")
print(ans0) ## Formula 5 in Rihoux and De Meur(2009:61)
SA(ans0) ## 18 simplifying assumptions

## mvQCA
conditions <- c("GNPCAP", "URBANIZA", "LITERACY", "INDLAB")
reduce(Lipset_mv,"SURVIVAL",conditions,explain="positive",remainder="exclude",case="CASEID",prep="mv_truthTabl
## formula 1 Cronqvist and Berg-Schlosser(2009:80)
ans1 <-
 reduce(Lipset_mv,"SURVIVAL",conditions,explain="positive",remainder="include",case="CASEID",prep="mv_truthTab
print(ans1) ## formula 2 in Cronqvist and Berg-Schlosser(2009:81)
SA(ans1) ## 9 SAs (see end note 7)
reduce(Lipset_mv,"SURVIVAL",conditions,explain="negative",remainder="exclude",case="CASEID",prep="mv_truthTabl
## formula 3 in Cronqvist and Berg-Schlosser(2009:81)
ans0 <-
 reduce(Lipset_mv,"SURVIVAL",conditions,explain="negative",remainder="include",contrad="positive",case="CASEID
print(ans0) ## formula 4 in Cronqvist and Berg-Schlosser(2009:81)
SA(ans0) ## 7 SAs (see end note 9)

## fsQCA
conditions <- c("Developed.FZ","Urban.FZ","Literate.FZ","Industrial.FZ", "Stable.FZ")
reduce(Lipset_fs,"Survived.FZ",conditions,explain="positive",remaind="exclude",prepro="fs",consistency=0.7)
## Formula 1 in Ragin (2009:112)
reduce(Lipset_fs,"Survived.FZ",conditions,explain="positive",remaind="include",prepro="fs",consistency=0.7)
## Formula 2 in Ragin (2009:114)
reduce(Lipset_fs,"Survived.FZ",conditions,explain="negative",remaind="exclude",prepro="fs",consistency=0.7)
## Formula 5 in Ragin (2009:115)
reduce(Lipset_fs,"Survived.FZ",conditions,explain="negative",remaind="include",prepro="fs",consistency=0.7)
## Formula 6 in Ragin (2009:117)
```

---

`simplifyingAssumption` *Simplyfing assumptions and contraditory simplifying assumptions*

---

## Description

Return the simplifying assumptions and contradictory simplifying assumptions.

## Usage

```
simplifyingAssumption(object, ...)

SA(object, ...) ## alias of simplifyingAssumption

CSA(object1, object0)

findNoCSA(x, y, noCSA = TRUE)
```

## Arguments

| | |
|---|---|
| object | An object of class "QCA", which is return from reduce |
| ... | Not used currently |
| object1 | An object of class "QCA" with one solution. |
| object0 | An object of class "QCA" with one solution. |
| x | object of class 'QCA' explaining positive outcome. |
| y | object of class 'QCA' explaining negative outome. |
| noCSA | logical, show index of solutions without CSA only when TRUE. |

## Details

Simplyfying assumption is assumption made on the outcome value of a logical remainder, so it can be included in the minimization procedure. Thus, it is meaning to use SA and CSA when the object is return by a call to reduce with remainder arugment set to "include".

A contraditory simplifying assumption (CSA) occurs when the same logical remainder is used both in the minization of the positive outcome configurations and in the minization of the negative outcome configuration. The CSA should be solved. An overly heavy presence of CSAs is one indicator of problem in the selection of conditions.

If can object of class "QCA" have multiple solutions, you can use [ to extract one of solution, then pass it to CSA. see example section for an example. For object1 and object0, one is the solution for explaination of positive case and the other is the solution for explaination of negative case.

## Value

For SA and CSA, the value is an object of class c("SA","QCA"). It is a list of 8 components.

| | |
|---|---|
| solutions | a list of data frame.For SA, each data frame consists one set of simplifying assumptions of each solution. For CSA, it presents a set of contradictory simplifying assumptions. Each row of a data frame is one implicant. |
| primeImplicants | the same as that of object. |
| explained | the same as that of object for SA and object1 for CSA. |
| idExclude | the same as that of object for SA and object1 for CSA. |
| nlevels | the same as that of object for SA and object1 for CSA. |
| PIChart | the same as that of object for SA and object1 for CSA. |
| call | the same as that of object for SA and object1 for CSA. |
| SAIDs | a list of integer vector, corresponding grouping id of solutions component. |

## Author(s)

Ronggui HUANG

## References

Yamasaki and Rihoux. 2009. A commented review of applications. In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage.

## See Also

reduce and constrReduce

## Examples

```
## Not run:
data(Yamasaki,package="QCA")
cond <- names(Yamasaki)[1:5]
ans0 <- reduce(Yamasaki,"AGENDA",cond,"negative","include") ## 5 solutions
ans1 <- reduce(Yamasaki,"AGENDA",cond,"positive","include") ## 1 solutions
SA(ans0)
SA(ans1)
CSA(ans0[1],ans1) ## no CSA, please note the subset operation
CSA(ans0[2],ans1) ## ans0[2]-ans0[5] have CSA

findNoCSA(ans1,ans0,FALSE)

## End(Not run)
```

---

# suffnec *Produces a sufficiency and necessity matrix*

---

## Description

creates a matrix of the sufficiency and necessity scores for a crisp set or fuzzy set data frame.

## Usage

```
suffnec(x, use = c("complete", "pairwise"))

## S3 method for class 'suffnec'
print(x, digits = 3, ...)
```

## Arguments

| | |
|---|---|
| x | For coincide, x is a dataframe of crip set or fuzzy set data, which ranges from 0 to 1. |
| use | The method of handling missing data. "complete" means listwise deletion and "pairwise" means pairwise deletion. |
| digits | a non-null value for 'digits' specifies the minimum number of significant digits to be printed in values. |
| ... | arguments passed to default method of print. |

## Details

In the terminology of set theory, if X is the sufficient condition of Y, then the score of X should be consistently less or equal than that of Y. Similarly, if X is the necessary condition of Y, then the sore of X should be consistently greater or equal than that of Y. The necessary score measures such consitency. The formulas can be found in Ragin(2006:297).

For crip set, sufficient score measures proportation of Y=1 given that X=1. The neccessary score measures the proportation of X=1 given that Y=1.

## Value

A list of two matrixs of consistency score.

| | |
|---|---|
| `suff` | Sufficiency Scores Matrix,measuring the consistency score of 'X is sufficient condition of Y'. |
| `nec` | Necessity Scores Matrix, measuring the consistency score of 'X is necessary condition of Y'. |

## Author(s)

Ronggui HUANG

## References

Ragin, Charles C. 2006. "Set Relations in Social Research:Evaluating Their Consistency and Coverage." Political Analysis 14 (3) : 291-310

## See Also

`coincid`

## Examples

```
suffnec(CoD[,c('cod',"econdev","ethlihom","nocom")])
```

---

`thresholdssetter` *Recode continuous variable into binary/multi-value variable*

---

## Description

If the conditions is continuous variable, it has to be recoded into binary variable in order to conduct csQCA, or multi-value variable in order to conduct mvQCA. `thresholdsster` helps to recode the variable using cluster analysis (the default method is 'average linkage').

## Usage

```
thresholdssetter(x, nthreshold = 1, value = TRUE, method ="average",
                 thresholds=NULL,dismethod="euclidean",print.table=TRUE)
```

## Arguments

| | |
|---|---|
| x | A continouse variable. |
| nthreshold | The number of thresholds. |
| value | logical, returns the recoded variable when TRUE, otherwise returns the thresholods only. |
| method | method of cluster analysis. See hclust for more details. |
| thresholds | numeric vector of the threshold values. If it is not NULL, then recode the variable according to thresholds rather than cluster analysis. The threshold values are included in the smaller group. |
| dismethod | method argument for dist. |
| print.table | logical, if TRUE, print the table of the recoded variable. |

## Details

In order to dichotomize the data, 1 threshold is needed; to trichotomize the data, 2 thresholds are needed; and so on. Yet, it should not be a large number, otherwise, limited diversity is a problem. It is suggested to be 1 to 3.

Whenever possible, use theoretically meaningful thresholds. Use statistical means only when necessary. The thresholds should make theoretical sense, thus you should examine the threshold values before going on. Good thresholod values should NOT create very differently sized subgroups. Mean and median should be avoided as well.

## Value

A vector of the thresholds when value is FALSE. The recoded variable when value is TRUE.

## Author(s)

Ronggui HUANG

## References

Cronqvist, Lasse and Berg-Schlosser, Dirk. 2009. Multi-Value QCA (mvQCA). In Configuraional comparative Methods: qualitative comparative analysis (QCA) and related techniques. ed by Benoit RiHoux and Charles Ragin. Sage.

Cronqvist, L. 2007. Tosmana user manual. http://www.tosmana.net/tosmana_manual1_3beta.pdf

## Examples

```
cuttingPoint <- thresholdssetter(Lipset$GNPCAP,2,value=FALSE)
thresholdssetter(Lipset$GNPCAP,thresholds=cuttingPoint)
thresholdssetter(Lipset$GNPCAP,2)## the same as the previous one
thresholdssetter(Lipset$GNPCAP,2,print=FALSE) ## print recoded variable rather than a table
```

Yamasaki2009 *Movement Impact on Nuclear Energy Policy.*

## Description

This is a data set about Movement Impact on Nuclear Energy Policy (Yamasaki 2009).

## Usage

```
data(Yamasaki2009)
```

## Format

A data frame with 10 observations on the following 8 variables.

`country` a factor with levels `Belgium` `Finland` `Germanty` `Italy` `Netherlands` `Spain` `Sweden` `Switzerland` `UK`

`period` a factor with levels `1975-1980` `1979-1984` `1982-1987` `1984-1989` `1985-1990` `1989-1994` `1993-1998` `1997-2002` `1998-2003`

`soc` social movement organisation.

`pol` political allies.

`ind` state-industry relationship.

`are` arena shift.

`foc` focusing event.

`policyChange` Major policy change.

## Source

Manually input by HUANG Ronggui according the truthTable in Yamasaki (2009: 490).

## References

Yamasaki, Sakura 2009. "A Boolean Analysis of Movement Impact on Nuclear Energy Policy." Mobilization: An International Quarterly 14 (4) : 485 - 504.

## Examples

```
data(Yamasaki2009)
reduce(policyChange~soc+pol+ind+are+foc,Yamasaki2009,cases="country")
## or construct truthTable first
cstt <- cs_truthTable(Yamasaki2009, "policyChange",c("soc","pol","ind","are","foc"), cases="country")
tb2 <- reduce(cstt) ## result in table 2
summary(tb2) ## coverage of each configuration
```

# Index