

# Package ‘QCA’

October 11, 2009

**Type** Package

**Title** Qualitative Comparative Analysis

**Version** 0.6-3

**Date** 2009-10-09

**Author** Adrian Dusa <dusa.adrian@unibuc.ro>

**Maintainer** Adrian Dusa <dusa.adrian@gmail.com>

**Depends** R (>= 1.9.0), lpSolve

**Suggests** XML

**Description** Performs the Quine-McCluskey algorithm for Qualitative Comparative Analysis, both for bvQCA (binary-value QCA) and for mvQCA (multi-value QCA). In the classical approach it currently handles about 15 conditions and one outcome for binary data only, but since version 0.4-5 the package has an enhanced, faster function that obtains the same exact solutions with a substantially lower memory consumption, it handles even more causal conditions and it can be used for mvQCA as well. The next versions of this algorithm will be oriented towards fuzzy-set functions. Also, in the near future the current functions will deal with missing values in the data

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2009-10-11 14:37:52

## R topics documented:

QCA-package . . . . .	2
allExpressions . . . . .	3
big10 . . . . .	4
createMatrix . . . . .	4
createString . . . . .	5
demoChart . . . . .	6
dozen12 . . . . .	7

Ebbinghaus	7
eqmcc	8
even11	10
factorize	11
findPrimes	12
findSubsets	13
getRow	14
lucky13	15
MI.15	15
mighty14	16
MV	16
Osa	16
prettyTable	17
qmcc	18
readTosmana	20
Rokkan	21
solveChart	22
Stokke	23
truthTable	23
Yamasaki	25

<b>Index</b>	<b>28</b>
--------------	-----------

---

QCA-package

*Qualitative Comparative Analysis*

---

## Description

This library implements the comparative methodology as described in Ragin (1987) and Ragin (2000). The analysis is the bridge between the qualitative and quantitative worlds, making use of the qualitative procedures in a systematic, algorithmic way (therefore increasing the "confidence" in the results, as understood by quantitative researchers). The "exact" Quine-McCluskey algorithm for crisp sets is implemented, both for bvQCA (binary-value QCA) and for mvQCA (multi-value QCA). Starting with version 0.4-5 the package has a new function called "eqmcc" (from "E"nhanced Quine-McCluskey) that finds exact solutions with a substantially lower memory consumption, and from version 0.6-0 this function also performs mvQCA. The next versions of the package will have more functions to address fsQCA (fuzzy-sets QCA) and to deal with missing values in the data. As of version 0.1-3 there is also a graphical user interface in the package QCAGUI.

## Details

Package: QCA  
 Type: Package  
 Version: 0.6-3  
 Date: 2009-10-09  
 License: GPL version 2 or newer

**Author(s)**

Author: Adrian Dusa <dusa.adrian@unibuc.ro>  
Maintainer: Adrian Dusa <dusa.adrian@gmail.com>

**References**

Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley:University of California Press

Dusa, Adrian (2007) *A mathematical solution to the boolean minimization problem*, available on <http://www.compass.org/files/WPfiles/Dusa2007.pdf>

Dusa, Adrian (2007a) *Enhancing Quine-McCluskey*, <http://www.compass.org/files/WPfiles/Dusa2007a.pdf>

---

allExpressions      *Create a matrix with all possible combinations of conditions*

---

**Description**

There are  $3^k - 1$  possible expressions (where  $k$  is the number of conditions) for a crisp-set procedure. Possible values for conditions are 1 (presence), 0 (absence) and NA (the condition being a subset of a larger one). The matrix grows exponentially so it is not recommended to run this function for a large number of causal conditions (say over 12) as the computer will quickly run out of resources.

**Usage**

```
allExpressions(no.conditions, inside=TRUE, arrange=FALSE)
```

**Arguments**

no.conditions	the number of conditions from the dataset
inside	if TRUE, returns the raw result matrix; if FALSE, print the result matrix without NAs
arrange	if TRUE the program tries hard to arrange the result matrix for visual inspection (however this takes a lot of additional time for large matrices)

**Value**

a matrix with  $3^k - 1$  rows

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**References**

Ragin, Charles C. (2000) *Fuzzy-set social science*, Chicago: The University of Chicago Press

**Examples**

```
# for 3 conditions
allExpressions(3, inside=FALSE)

# the same matrix, this time arranged better
allExpressions(3, inside=FALSE, arrange=TRUE)
```

---

big10	<i>Random data with 10 conditions and 1 outcome</i>
-------	---

---

**Description**

The data was randomly generated using `set.seed(10)`

**Usage**

```
data(big10)
```

---

createMatrix	<i>Create a base matrix for the truth table</i>
--------------	---

---

**Description**

The truth table consists from all combinations of presence/absence of conditions (coded binary 1/0). There are  $2^k$  such combinations and this function should be among the fastest to create this matrix, using ideas inspired from `expand.grid`

**Usage**

```
createMatrix(noflevels)
```

**Arguments**

`noflevels` a vector containing the number of levels for each variable in the dataset

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**References**

Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press

**See Also**

[truthTable](#)

**Examples**

```
# create a binary matrix based on 3 conditions
createMatrix(rep(2, 3))

# create a matrix based on 3 conditions where the second has three levels
createMatrix(c(2, 3, 2))

# the matrix has 2*3*2 = 12 rows
```

---

createString	<i>Create a vector of strings containing combinations of presence/absence of conditions</i>
--------------	---

---

**Description**

The function takes the values from the dataset (0 or 1) and writes the corresponding lower/upper column names.

**Usage**

```
createString(mydata, use.letters=TRUE, uplow=TRUE)
```

**Arguments**

mydata	a matrix or a dataframe
use.letters	logical, use letters instead of column names
uplow	logical, use upper/lower notation if possible, for binary data

**Value**

A vector of strings, with length equal to the number of rows in **mydata**

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**See Also**

[createChart](#)

**Examples**

```
test.dataset <- data.frame(First=c(1,1), Second=c(0,1), Third=c(0,0))
createString(test.dataset) # "Abc" "ABc"

# using the column names instead of letters
createString(test.dataset, use.letters=FALSE)
```

---

demoChart

*Create the prime implicants chart*

---

**Description**

This function creates a chart having the prime implicants on the rows and the observed combinations of conditions on the columns. It is useful to determine visually which prime implicant (if any) is essential. The chart is subsequently processed algorithmically to further reduce the redundant prime implicants. This function is for demonstration purposes only. The internal function `createChart()` is faster but its arguments are more complex.

**Usage**

```
demoChart(rows, columns, splitmethod="")
```

**Arguments**

<code>rows</code>	a vector of strings, containing the prime implicants
<code>columns</code>	a vector of strings, containing all combinations of conditions from the original data
<code>splitmethod</code>	string, to declare the separator of the input strings

**Value**

a logical matrix showing which conditions from the (minimized) prime implicants are found in which columns

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**References**

Quine, W.V. (1952) *The Problem of Simplifying Truth Functions*, The American Mathematical Monthly, Vol. 59, No. 8. (Oct., 1952), pp. 521-531.

Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press

**See Also**

[prettyTable](#)

**Examples**

```
chart <- demoChart(c("A", "B", "c"), c("ABC", "Abc", "AbC", "aBc"))
prettyTable(chart)

# Quine's example, page 528
rows <- c("AB", "BC", "Ac", "aC", "abd", "bcd")
cols <- c("ABCD", "ABCd", "ABcD", "ABcd", "AbcD", "Abcd",
         "aBCD", "aBCd", "abCD", "abCd", "abcd")

chart <- demoChart(rows, cols)
prettyTable(chart)
```

**Description**

The data was randomly generated using `set.seed(12)`

**Usage**

```
data(dozen12)
```

**Description**

The **Ebbinghaus** data frame has 13 rows and 10 columns.

During the early post-war period, Western trade union movements grew in membership and achieved an institutionalized role in industrial relations and politics. However, during the last decades, many trade unions have seen their membership decline as they came increasingly under pressures due to the social, economic and political changes. This article reviews the main structural, cyclical and institutional factors explaining union growth and decline. Concentrating on Western Europe, the empirical analysis compares cross-national union density data for 13 countries over the first period (1950-75) and for 16 countries over the second, "crisis" period (1975-95)

**Usage**

data (Ebbinghaus)

**Format**

The dataset contains the following columns:

OUT	union decline (outcome variable)
GHENT	country with "Ghent-system" (union led unemployment insurance)
WORKACCESS	workplace access for unions
NEOCORP	neo-corporatist institutionalization of unions
CLOSHOP	"closed-shop" tradition
SOCDEM	government participation of Social-Democratic / Socialist party
INDUS	share of industry in dependent unemployment
PUBLIC	share of public sector development employment
UNEMPL	average unemployment rate
INFLA	average inflation rate

**Source**

<http://www.compass.org>

**References**

Ebbinghaus, Bernhard and Visser, Jelle 2000 *Trade Unions in Western Europe since 1945*. In Flora, P. and Rothenbacher, Kraus F. (eds.) *The Societies of Europe Series*, London: Macmillan Reference / Palgrave, March 2000 / New York: Grove's Dictionaries / Palgrave, July 2000 [xxii, 808 pp. and CD-ROM]

**Description**

This function is designed to improve the speed and memory problems that `qmcc()` currently has and it has very high chances to replace the main `qmcc()` in the future; all tests have been successful, the solutions are exactly the same down to the last literal. It has about the same options as `qmcc()`, minus the "details" and "diffmatrix" arguments which are no longer required. It is called "eqmcc" because it doesn't follow the classic minimization algorithm but a more direct and rapid enhancement. As of version 0.6-0, `eqmcc()` accepts multi-valued data.

**Usage**

```
eqmcc(mydata, outcome = "", conditions = c(""), incl.rem = FALSE,
      expl.1 = FALSE, expl.0 = FALSE, expl.ctr = FALSE, expl.mo = FALSE,
      incl.1 = FALSE, incl.0 = FALSE, incl.ctr = FALSE, incl.mo = FALSE,
      quiet = FALSE, chart = FALSE, use.letters = TRUE, show.cases = FALSE,
      uplow=TRUE)
```

**Arguments**

<code>mydata</code>	a truth table (an R object with class "truthTable"), or the dataset used for minimization (either as a dataframe or as a matrix)
<code>outcome</code>	the name of the outcome variable from the dataset
<code>conditions</code>	a string vector containing the conditions' names from the dataset (if not specified, all variables but the outcome are included)
<code>incl.rem</code>	logical, include the remainders in the minimization procedure
<code>expl.1</code>	logical, explain the outcomes equal to 1
<code>expl.0</code>	logical, explain the outcomes equal to 0
<code>expl.ctr</code>	logical, explain the contradictions
<code>expl.mo</code>	logical, explain the missing outcomes (not implemented yet)
<code>incl.1</code>	logical, include the outcomes equal to 1 in the minimization procedure
<code>incl.0</code>	logical, include the outcomes equal to 0 in the minimization procedure
<code>incl.ctr</code>	logical, include the contradictions in the minimization procedure
<code>incl.mo</code>	logical, include the missing outcome in the minimization procedure (not implemented yet)
<code>quiet</code>	logical, print the solution without any other information
<code>chart</code>	logical, print the prime implicants chart
<code>use.letters</code>	logical, should letters be used instead of column names
<code>show.cases</code>	logical, show the lines corresponding to every minimized prime implicant
<code>uplow</code>	logical, use upper/lower notation if possible, for binary data

**Note**

The speed is greatly improved: for bvQCA and 15 causal conditions it takes less than 2 minutes, compared to 10 minutes in qmcc(). The memory consumption is tiny by comparison: also for 15 causal conditions, qmcc() uses about 1.5 GB of RAM, where eqmcc() uses about 50 MB.

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**References**

Dusa, Adrian (2007a) *Enhancing Quine-McCluskey* <http://www.compass.org/files/WPfiles/Dusa2007a.pdf>

**See Also**

[truthTable](#), [qmcc](#)

**Examples**

```
data(Osa)

# explaining only the presence of the outcome
eqmcc(Osa, outcome="OUT", expl.1=TRUE)

# now including the remainders and the contradictions
eqmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE)

# the same as above, but we want to see the prime implicants chart
eqmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE, chart=TRUE)

# printing the lines corresponding to each prime implicant
eqmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE, show.cases=TRUE)

# now explaining the absence of the outcome
eqmcc(Osa, outcome="OUT", expl.0=TRUE, incl.rem=TRUE, incl.ctr=TRUE, show.cases=TRUE)

# for multi-value data
data(MV)
eqmcc(MV, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, show.cases=TRUE)
```

---

even11

*Random data with 11 conditions and 1 outcome*

---

**Description**

The data was randomly generated using `set.seed(11)`

**Usage**

```
data(even11)
```

---

factorize	<i>Factorize a QCA solution</i>
-----------	---------------------------------

---

**Description**

This function finds all combinations of common factors of literals in a QCA solution

**Usage**

```
factorize(my.string, splitmethod="", sort.by.literals=FALSE, sort.by.number=FALSE)
```

**Arguments**

`my.string` a string which contains a sequence of literals  
`splitmethod` string, to declare the separator of the input string  
`sort.by.literals`  
sort results by the largest number of literals as common factor  
`sort.by.number`  
sort results by the largest number of elements that have been factorized

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**References**

Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press

**See Also**

[qmcc](#), [eqmcc](#)

**Examples**

```
factorize("AB + AC + CD")  
  
factorize("one*TWO*four + one*THREE + THREE*four", splitmethod="*")
```

---

findPrimes	<i>Find all possible prime implicants of one or several combinations of causal conditions</i>
------------	---

---

### Description

It is a general rule that all prime implicants can be found in the  $3^k$  space, understood as all possible combinations of values in base 3, each variable having three levels: 0, 1 and 2 (Dusa, 2007, 2007a). There is a finite number of prime implicants for any combination of causal conditions, equal to  $2^k - 1$ , where  $k$  is the number of causal conditions. Counting out the input combination itself, the actual number is  $2^k - 2$ . This function computes all possible unique prime implicants for a specific set of combinations (either to explain or to exclude)

### Usage

```
findPrimes(noflevels, input.combs)
```

### Arguments

noflevels	a vector containing the number of levels for each causal condition plus 1 (because all prime implicants are to be found in the higher matrix)
input.combs	a matrix with combinations of causal conditions or a vector of line numbers from the same matrix

### Value

a vector with the line numbers of all possible prime implicants in the  $3^k$  space

### Author(s)

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

### References

Dusa, Adrian (2007) *A mathematical approach to the boolean minimization problem*, available on <http://www.compass.org/files/WPfiles/Dusa2007.pdf>

Dusa, Adrian (2007a) *Enhancing Quine-McCluskey*, <http://www.compass.org/files/WPfiles/Dusa2007a.pdf>

### See Also

[getRow](#), [findSubsets](#)

**Examples**

```

# all three conditions are binary, having two levels: 0 and 1
noflevels <- c(2, 2, 2)

findPrimes(noflevels + 1, 14)      # 2 4 5 10 11 13 14
findPrimes(noflevels + 1, 17)      # 2 7 8 10 11 16 17

# both line numbers 14 and 17
findPrimes(noflevels + 1, c(14, 17)) # 2 4 5 7 8 10 11 13 14 16 17

# input.combs as a matrix
(input.combs <- getRow(noflevels + 1, c(14, 17)))

#      [,1] [,2] [,3]
#[1,]    1    1    1
#[2,]    1    2    1

findPrimes(noflevels + 1, input.combs) # 2 4 5 7 8 10 11 13 14 16 17

```

---

findSubsets

*Find all possible subsets of a given prime implicant*


---

**Description**

It is a general rule that all subsets can be found in the  $3^k$  space, understood as all possible combinations of values in base 3, each variable having three levels: 0, 1 and 2 (Dusa, 2007, 2007a). If a prime implicant can be considered a superset of an initial combination of causal conditions, the reverse is also true: the initial combination is a subset of a prime implicant. Even more, a less minimum prime implicant (with more literals) is also a subset of another shorter (more minimum) prime implicant. This function finds all possible such subsets for a given prime implicant, in the  $3^k$  space.

**Usage**

```
findSubsets(noflevels, row.no, maximum)
```

**Arguments**

noflevels	a vector containing the number of levels for each causal condition plus 1 (because all subsets are to be found in the higher matrix)
row.no	the row number where the (minimum) prime implicant is located
maximum	the maximum line number (subset) to be returned

**Value**

a vector with the line numbers of all possible subsets in the  $3^k$  space

**Author(s)**

Adrian Dusa  
 University of Bucharest, Faculty of Sociology and Social Work  
 dusa.adrian@unibuc.ro

**References**

Dusa, Adrian (2007) *A mathematical approach to the boolean minimization problem*, available on <http://www.compass.org/files/WPfiles/Dusa2007.pdf>  
 Dusa, Adrian (2007a) *Enhancing Quine-McCluskey*, <http://www.compass.org/files/WPfiles/Dusa2007a.pdf>

**See Also**

[findPrimes](#)

**Examples**

```
# all three conditions are binary, having two levels: 0 and 1
noflevels <- c(2, 2, 2)

findSubsets(noflevels + 1, 2)      # 5 8 11 14 17 20 23 26

# stopping at maximum row number 20
findSubsets(noflevels + 1, 2, 20) # 5 8 11 14 17 20
```

---

getRow

*Get a specific row from a truth table*

---

**Description**

A truth table is formed by the unique combinations of causal conditions' levels. For binary crisp sets, there are exactly  $2^k$  combinations, where  $k$  is the number of conditions. This function accepts multiple levels (not just 0 and 1) and its purpose is to transform the decimal representation of a row number into its corresponding combination of levels.

**Usage**

```
getRow(noflevels, row.no, zerobased=FALSE)
```

**Arguments**

`noflevels` a vector containing the number of levels for each causal condition  
`row.no` a vector, the desired row number(s) from the truth table  
`zerobased` logical; if TRUE the first row number from the truth table is zero

**Value**

a matrix with the combination of levels corresponding to the desired row number(s)

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**See Also**

[createMatrix](#), [expand.grid](#)

**Examples**

```
# all three conditions are binary, having two levels: 0 and 1
noflevels <- c(2, 2, 2)
getRow(noflevels, 8) # 1 1 1
getRow(noflevels, 0, zerobased=TRUE) # 0 0 0

# the second condition has three levels: 0, 1 and 2
noflevels <- c(2, 3, 2)
getRow(noflevels, 11) # 1 2 0
```

---

lucky13

*Random data with 13 conditions and 1 outcome*

---

**Description**

The data was randomly generated using `set.seed(13)`

**Usage**

```
data(lucky13)
```

---

MI.15

*Random data with 15 conditions and 1 outcome*

---

**Description**

The data was randomly generated using `set.seed(15)` "MI" stands for Mission Impossible

**Usage**

```
data(MI.15)
```

---

`mighty14`*Random data with 14 conditions and 1 outcome*

---

**Description**

The data was randomly generated using `set.seed(14)`

**Usage**

```
data(mighty14)
```

---

`MV`*Random Multi-Value data with 4 causal conditions and 1 outcome*

---

**Description**

This data was generated for testing purposes on MV data.

**Usage**

```
data(MV)
```

---

`Osa`*Political Opportunity in Non-democracies*

---

**Description**

The **Osa** data frame has 24 rows and 6 columns.

This data is drawn from a study which analyzes twenty-four cases of occurrence/non-occurrence of mobilization in non-democratic states to determine conditions of political opportunity in high-risk authoritarian contexts. Political opportunity is sensitive to conditions created by divided elites, changes in repression, media access, influential allies, and social networks

**Usage**

```
data(Osa)
```

**Format**

The dataset contains the following columns:

DYNA	Dynamics of repression, coded 1 for evidence of increase in long-term and/or short-term state repression, 0 for evidence of decrease
ACCESS	Media access - coded 1 if the public information flow concerning a particular event of popular mobilization is controlled by the political authorities (via censorship or ban on foreign media), 0 if there was evidence of sustained relaxation of state censorship or substantive presence of foreign and/or underground media
INFLU	Influential allies - coded 1 for presence of domestic or foreign politically influential groups supporting popular mobilization; 0 for absence of such organizational support
ELITE	Division of elites, coded 1 - evidence of competing factions within the ruling elites; 0 - relatively unified ruling group
SOCIAL	Social networks - coded 1 if mobilization resulted from the activity of interconnected groups, 0 if organizational and/or individual ties were severely destroyed or impeded to emerge by the state
OUT	Social mobilization, coded 1 for major episodes of sustained collective action opposing state policies by participants drawn from nonelite or repressed segments of society in non-democratic regimes, 0 - non-mobilization

**Source**

<http://www.compass.org>

**References**

Osa, Maryjane and Corduneanu-Huci, Cristina 2003 *Running Uphill: Political Opportunity in Non-democracies*, Comparative Sociology 2(4), pp. 605-629(25)

---

```
prettyTable
```

*Print a nicely formatted matrix or truth table*

---

**Description**

R prints the content of a matrix with the values alligned to the right. This function computes the number of spaces needed to reach the middle of the column (minus one for odd number of characters) and adds a certain number of spaces to each value from the table in order to alling these values to the middle.

**Usage**

```
prettyTable(mytable)
```

**Arguments**

```
mytable      a matrix
```

**Value**

a matrix with character elements

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**See Also**

[print](#)

**Examples**

```
mymatrix <- matrix(sample(0:1, 20, replace=TRUE), nrow=5)
colnames(mymatrix) <- c("First", "Second", "Third", "Fourth")
rownames(mymatrix) <- c("One", "Two", "Three", "Four", "Five")
print(prettyTable(mymatrix))
```

---

qmcc

*Find a minimum solution in a boolean procedure, using the Quine-McCluskey algorithm*

---

**Description**

This function is the core of the QCA (Qualitative Comparative Analysis) package. The QCA is the bridge between the qualitative and quantitative worlds, and its fundamental philosophy is based on Mill's methods of inductive inquiry. Given a dataset with cases on the rows (or better said, case studies) and various conditions on the columns, this function finds which condition (or combination of conditions) is necessary and/or sufficient to trigger the outcome. As of version 0.6-0, qmcc() is explicitly working with binary data only.

**Usage**

```
qmcc(mydata, outcome = "", conditions = c(""), incl.rem = FALSE,
     expl.1 = FALSE, expl.0 = FALSE, expl.ctr = FALSE, expl.mo = FALSE,
     incl.1 = FALSE, incl.0 = FALSE, incl.ctr = FALSE, incl.mo = FALSE,
     quiet = FALSE, details = TRUE, chart = FALSE, use.letters = TRUE,
     show.cases = FALSE, diffmat=TRUE)
```

**Arguments**

<code>mydata</code>	the dataset used for minimization (dataframe or matrix)
<code>outcome</code>	the name of the outcome variable from the dataset
<code>conditions</code>	a string vector containing the conditions' names from the dataset (if not specified, all variables but the outcome are included)
<code>incl.rem</code>	include the remainders in the minimization procedure
<code>expl.1</code>	explain the outcomes equal to 1
<code>expl.0</code>	explain the outcomes equal to 0
<code>expl.ctr</code>	explain the contradictions
<code>expl.mo</code>	explain the missing outcomes (not implemented yet)
<code>incl.1</code>	include the outcomes equal to 1 in the minimization procedure
<code>incl.0</code>	include the outcomes equal to 0 in the minimization procedure
<code>incl.ctr</code>	include the contradictions in the minimization procedure
<code>incl.mo</code>	include the missing outcome in the minimization procedure (not implemented yet)
<code>quiet</code>	prints the solution without any other information
<code>details</code>	prints some relevant details (set to FALSE if quiet is TRUE)
<code>chart</code>	print the prime implicants chart
<code>use.letters</code>	should letters be used instead of column names
<code>show.cases</code>	show the lines corresponding to every minimized prime implicant
<code>diffmat</code>	generate the differences matrix

**Note**

The function finds an exact solution very quickly for relatively small number of conditions. With a 2 GHz Intel Core2Duo processor and 1 GB of RAM, the algorithm finds a solution for 10 conditions in a less than 1 second. For 11 conditions it takes about 2.5 seconds, for 12 conditions it takes 9 seconds, for 13 conditions it takes about 37 seconds and for 14 conditions it takes about 130 seconds (a little over 2 minutes). For more than 14 conditions more memory is definitely needed; choosing not to create the differences matrix (which requires a lot of RAM), the analysis can squeeze a result for 15 conditions at the expense of 10 minutes of computing time.

**Author(s)**

Adrian Dusa  
 University of Bucharest, Faculty of Sociology and Social Work  
 dusa.adrian@unibuc.ro

**References**

- Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press
- Dusa, Adrian (2007) *A mathematical approach to the boolean minimization problem*, available on <http://www.compass.org/files/WPfiles/Dusa2007.pdf>

**See Also**

[truthTable](#), [eqmcc](#)

**Examples**

```
data(Osa)

# find a solution using the Quine-McCluskey (qmcc) algorithm
# explaining only the presence of the outcome
qmcc(Osa, outcome="OUT", expl.1=TRUE)

# now including the remainders and the contradictions
qmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE)

# the same as above, but we want to see the prime implicants chart
qmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE, chart=TRUE)

# even more details
qmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE, details=TRUE)

# printing the lines corresponding to each prime implicant
qmcc(Osa, outcome="OUT", expl.1=TRUE, incl.rem=TRUE, incl.ctr=TRUE, show.cases=TRUE)

# now explaining the absence of the outcome
qmcc(Osa, outcome="OUT", expl.0=TRUE, incl.rem=TRUE, incl.ctr=TRUE, show.cases=TRUE)
```

---

readTosmana

*Read Tosmana XML data file*

---

**Description**

Tosmana saves its data by default in XML format. This function reads the XML file and coerces it to an R dataframe object.

**Usage**

```
readTosmana(filename)
```

**Arguments**

filename      a string, the path to the XML file

**Note**

This function requires the XML package

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

---

Rokkan

*Divided Working-Class Movements*

---

**Description**

The **Rokkan** data frame has 16 rows and 5 columns.

Abridged from Ragin (1987):

The data was used by Rokkan (1970) in his work on nation building in Western Europe. Rokkan used a "configurational" approach that bears many similarities to the Boolean approach presented in this work. His main substantive interest was the growth of mass democracy and the emergence of different cleavage structures in Western European polities. One outcome that interested him was the division of some working-class movements in these countries following the Russian Revolution into internationally oriented wings and some into nationally oriented wings. He considered the distribution of this outcome important because of its implication for the future of working-class mobilization (and cleavage structures in general) in Western Europe

**Usage**

`data(Rokkan)`

**Format**

The dataset contains the following columns:

- C - National church (vs. state allied to Roman Catholic church)
- R - Significant Roman Catholic population and Roman Catholic participation in mass education
- L - State protection of landed interests
- E - Early state
- S - Major split in working-class movement provoked by Russian Revolution (outcome variable) (yes/no)

**Source**

Ragin, Charles C. 1987 *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press, pp.129

**References**

Rokkan, Stein 1970 *Citizens, Elections, Parties*, New York: McKay

---

`solveChart`*Reduce the redundant prime implicants*

---

**Description**

While the minimization procedure finds a certain number of prime implicants, not all of them are necessary as some are already covered by others (therefore redundant). This function further reduces the number of prime implicants for a minimum solution.

**Usage**

```
solveChart(chart)
```

**Arguments**

`chart` the prime implicants chart, a matrix with TRUE/FALSE values

**Value**

a matrix containing row indices of all possible solutions

**Author(s)**

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

**References**

Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press

**See Also**

[createChart](#)

**Examples**

```
# the chart we're trying to further reduce
chart <- demoChart(c("A", "B", "c"), c("ABC", "Abc", "AbC", "aBc"))
prettyTable(chart)

solveChart(chart)

## two minimum solutions: first and second or first and third prime implicants
```

---

Stokke

*Causal mechanisms and regime effectiveness*

---

### Description

The **Stokke** data frame has 9 rows and 6 columns.

The purpose of introducing mechanisms in regime research, is partly to allow detailed examination of the various ways in which regimes may affect behaviour: this could be coined a magnifying purpose. The second objective is the methodological one of facilitating systematic comparison of cases by constituting them in ways that make them sufficiently homogeneous to permit employment of available comparative techniques. The cases presented in this data stress on "shaming" as a causal mechanism for international resource management. Shaming highlights attempts to bring about a change in problem-related behaviour not by material rewards or punishment but by exposing certain practices to third parties whose opinion matters to the target of shaming

### Usage

```
data(Stokke)
```

### Format

The dataset contains the following columns:

- A - Advice
- C - Commitment
- S - Shadow of the future
- I - Inconvenience
- R - Reverberation
- Y - Success (outcome variable)

### Source

<http://www.compass.org>

### References

Stokke, Olav Schram 2004 *Boolean Analysis, Mechanisms, and the Study of Regime Effectiveness* in Underdal, Arild and Young, Oran R. (eds.) *Methodological Challenges and Research Strategies* Dordrecht: Kluwer Academic, pp. 87-119

---

truthTable

*Truth table*

---

## Description

For any number of conditions, there is a finite number of possible combinations of presence/absence. `truthTable()` finds the observed combinations among all possible ones, prints the frequency of each observed combination and establishes the value for the outcome in this way:

- if all observed combinations agree on having the same outcome value (either 0 or 1), then the value for the outcome will be set to that value
- for any given combination, if the outcome present values of both 0 and 1 then the value for the outcome will be set to a contradiction ("C")
- for all other possible combinations, the outcome is missing and will be coded with "?"

`is.tt()` checks if an object has the class `tt` (if it is a truth table); such an object is created by `truthTable()`

`print.tt()` has an S3 method for printing objects of class `'tt'`

## Usage

```
truthTable(mydata, outcome = "", conditions = c(""), complete = FALSE,
           show.cases = FALSE, quiet = FALSE)
```

```
is.tt(x)
```

```
## S3 method for class 'tt':
print(x, funqmcc=FALSE, ...)
```

## Arguments

<code>mydata</code>	the dataset we use for minimization
<code>outcome</code>	the name of the outcome variable in the dataset
<code>conditions</code>	the name of the conditions from the dataset (if not specified, all variables but the outcome are considered conditions)
<code>quiet</code>	if TRUE, return the truth table invisibly
<code>show.cases</code>	show the rownames from the original dataset for each combination of conditions
<code>complete</code>	prints the complete truth table, including the missing combinations
<code>x</code>	an object of class <code>tt</code>
<code>funqmcc</code>	logical, if called by <code>(e)qmcc()</code> function(s) it prints only the observed combinations from the initial data
<code>...</code>	other arguments from the generic <code>print</code> (not used in this function)

## Value

An object of class `"tt"`, which is essentially a list with three components:

<code>\$tt</code> :	the truth table itself
<code>\$indexes</code> :	a vector with the base 10 representation of the

truth table observed combinations  
\$noflevels: a vector with the number of levels from all input  
variables

### Author(s)

Adrian Dusa  
University of Bucharest, Faculty of Sociology and Social Work  
dusa.adrian@unibuc.ro

### References

Ragin, Charles C. (1987) *The Comparative Method. Moving beyond qualitative and quantitative strategies*, Berkeley: University of California Press

### See Also

[qmcc](#), [eqmcc](#)

### Examples

```
data(Osa)

# print the truth table
truthTable(Osa, outcome="OUT", show.cases=TRUE)

# print the complete truth table
truthTable(Osa, outcome="OUT", complete=TRUE, show.cases=TRUE)

# save the result into an R object:
mytable <- truthTable(Osa, outcome="OUT", complete=TRUE, show.cases=TRUE, quiet=TRUE)

mytable # or print.tt(mytable)

# check the components

mytable$tt # the truth table itself

mytable$indexes # base 10 representation of input combinations

mytable$noflevels # number of levels from each causal condition
```

## Description

The **Yamasaki** data frame has 6 rows and 6 columns.

In Qualitative Comparative Analysis (QCA) of social data, the generation of parsimonious explanatory equations is enhanced by the inclusion of "logical configurations". Even if this procedure proves to be very useful, it also raises various methodological issues. Among them, the tricky problem of "contradictory simplifying assumptions" has remained largely unexplored. Yet the careful control of this obstacle is crucial for any QCA to be successful, not only because contradictory assumptions are inducing wrong conclusions, but also because their resolution can generate most interesting results. Hence, our contribution aims at enlightening this difficulty, as well as designing an efficient way to overcome it. In this perspective, we start from data collected for a comparative research on "the political feasibility of an unconditional basic income" in six OECD countries (1980-2002). After having briefly stated the core elements of the research question, six operational variables are defined (section 1). On this basis, we conduct a Boolean analysis and comment the various 'feasibility scenarios' generated by the QCA 3.0 software (section 2). Starting from these first results, we identify contradictory simplifying assumptions used by the software, and discuss possible solutions to this problem. New results are then generated (section 3). In the conclusion, we shortly discuss the general implications of this methodological problem

## Usage

```
data(Yamasaki)
```

## Format

The dataset contains the following columns:

POSTMAT	the level of postmodern values in a society takes the value "1" in the Dutch and Finnish cases, and "0" in the others
NONGHENT	the absence of a Ghent system is coded "1" and "0" when the Ghent system exists in the country
MOVEMENT	in the presence of a social movement advocating Basic Income, the variable will be coded "1"
UNITARY	based on the Lijphart index (1.0 to 5.0, from unitary to federal and decentralised states), the variable is coded "0" for cases in the three following categories: semi-federal states (3.0), federal and centralized states (4.0), and federal and decentralized states (5.0), and "1" otherwise.
SOCIAL	based on Esping-Andersen's categorisation of welfare regimes, the variable is coded "1" for non-liberal countries and "0" otherwise
AGENDA	the outcome variable, coded "1" when Basic Income has been considered as a serious alternative at the governmental level

## Source

<http://www.compass.org>

## References

Vanderborght, Yannick and Yamasaki, Sakura 2003 *The Problem of Contradictory Simplifying Assumptions in Qualitative Comparative Analysis (QCA)*, Paper presented at the ECPR General on-

ference, 18-21 Sept., Marburg, Germany

Vanderborght, Yannick and Yamasaki, Sakura 2004 *Des cas logiques...contradictaires? Un piège de l'AQQC dejoué à travers l'étude de la faisabilité politique de l'Allocation Universelle*, Revue Internationale de Politique Comparée, Vol.11, pp.51-66

# Index

## \*Topic **cluster**

QCA-package, 2

## \*Topic **datasets**

big10, 4

dozen12, 7

Ebbinghaus, 7

even11, 10

lucky13, 15

MI.15, 15

mighty14, 15

MV, 16

Osa, 16

Rokkan, 20

Stokke, 22

Yamasaki, 25

## \*Topic **misc**

allExpressions, 3

createMatrix, 4

createString, 5

demoChart, 6

eqmcc, 8

factorize, 10

findPrimes, 11

findSubsets, 13

getRow, 14

prettyTable, 17

qmcc, 18

readTosmana, 20

solveChart, 21

truthTable, 23

allExpressions, 3

big10, 4

createChart, 5, 22

createMatrix, 4, 14

createString, 5

demoChart, 6

dozen12, 7

Ebbinghaus, 7

eqmcc, 8, 11, 19, 24

even11, 10

expand.grid, 14

factorize, 10

findPrimes, 11, 13

findSubsets, 12, 13

getRow, 12, 14

is.tt (*truthTable*), 23

lucky13, 15

MI.15, 15

mighty14, 15

MV, 16

Osa, 16

prettyTable, 7, 17

print, 17

print.tt (*truthTable*), 23

QCA (*QCA-package*), 2

QCA-package, 2

qmcc, 9, 11, 18, 24

readTosmana, 20

Rokkan, 20

solveChart, 21

Stokke, 22

truthTable, 4, 9, 19, 23

Yamasaki, 25