Package 'SetMethods'

December 1, 2017

Type Package

Title Functions for Set-Theoretic Multi-Method Research and Advanced QCA

Version 2.3

Date 2017-11-29

Author Juraj Medzihorsky [aut],
Ioana-Elena Oana [aut, cre],
Mario Quaranta [aut],
Carsten Q. Schneider [aut]

Maintainer Ioana-Elena Oana <oana_ioana-elena@phd.ceu.edu>

Description Functions for performing set-theoretic multi-method research, QCA for clustered data, theory evaluation, Enhanced Standard Analysis, indirect calibration, radar visualisations. Additionally it includes data to replicate the examples in the book by C. Q. Schneider and C. Wagemann ``Set Theoretic Methods for the Social Sciences", Cambridge University Press and in the online appendix.

Depends QCA, lattice, betareg, fmsb, methods, R (>= 3.3.0)

License GPL-2

NeedsCompilation no

Date/Publication 2017-12-01 18:37:25 UTC

Repository CRAN

R topics documented:

SetMethods-package *Functions for Set-Theoretic Multi-Method Research and Advanced QCA*

Description

This initiated as a package companion to the book by C. Q. Schneider and C. Wagemann "Set-Theoretic Methods for the Social Sciences", Cambridge University Press. It has now grown to include functions for performing set-theoretic multi-method research (Schneider and Rohlfing 2013), QCA for clustered data (Garcia-Castro and Arino 2013), theory evaluation, Enhanced Standard Analysis, QCA Radar Charts, indirect calibration etc.. Additionally it includes data to replicate the examples in the book and in the online appendix.

Details

Package:	SetMethods
Type:	Package
Version:	2.3
Date:	2017-11-29
License:	GPL-2

The package contains functions to perform set-theoretic multi-method research, theory evaluation, QCA for clustered data, Enhanced Standard Analyses, indirect calibration, calculate parameters of fit and produce XY plots and QCA Radar Charts. Furthermore, it contains all the data used in "Set-Theoretic Methods for the Social Sciences" and more.

Author(s)

Juraj Medzihorsky [aut], Ioana-Elena Oana [aut, cre], Carsten Q. Schneider [aut], Mario Quaranta [aut]

Maintainer: Ioana-Elena Oana <oana_ioana-elena@phd.ceu.edu>

References

Haesebrouck, T. (2015) Pitfalls in QCA's consistency measure. Journal of Comparative Politics 2:65-80.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012). How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods Research 42(4): 559-597

Garcia-Castro, A., Arino, M. A.. 2013. A General Approach to Panel Data Set-Theoretic Research. COMPASSS Working Paper 2013-76

BCMV *Berg-Schlosser and Cronqvist (2005)*

Description

The BCMV data frame has 18 rows and 5 variables

Usage

```
data(BCMV)
```

Format

A data frame with 18 observations on the following 5 variables.

GNP a numeric vector. Condition, Gross National Product/Capita (ca. 1930). 0 if below 500$, 1 if between 550 and 850$, 2 above 850$.

URB a numeric vector. Condition, urbanization (population in towns with 20000 and more inhabitants); 0 if below 50 per cent; 1 if above.

LIT a numeric vector. Condition, literacy: 0 if below 75 per cent; 1 if above.

INDUS a numeric vector. Condition, Industrial Labour Force (incl. mining); 0 if below 30 per cent of active population; 1 if above.

DEMOC a numeric vector. Condition, stability of a democracy: 0 if not stable; 1 if stable.

Details

The data are used by Berg-Schlosser and Cronqvist (2005) to demostrate mvQCA. The original data are from Lipset (1963). Data are multi-value.

References

Berg-Schlosser, D. and Cronqvist, L. (2005) "Macro-Quantitative vs. Macro-Qualitative Methods in the Social Sciences - An Example from Empirical Democratic Theory", Historical Social Research 30, pp. 154-175.

Lipset, Seymour M. (1963) Political Man. The Social Bases of Politics. Doubleday: New York.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(BCMV)
```

`cases.suf.dcn` *List cases deviant with regards to consistency for sufficiency.*

Description

A function that extracts cases deviant with regards to consistency for sufficiency from an object of class "qca".

Usage

```
cases.suf.dcn(results, outcome, neg.out=FALSE, sol = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

Schneider, C. Q., Rohlfing I. 2016. Case Studies Nested in Fuzzy-Set QCA on Sufficiency: Formalizing Case Selection and Causal Inference. Sociological Methods & Research 45(3): 526-68

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:
```

```
sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Return deviant cases consistency for sufficiency for the parsimonious solution:

cases.suf.dcn(results = sol_yp, outcome = "EXPORT")

# Return deviant cases consistency for sufficiency for the intermediate solution:

cases.suf.dcn(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Return deviant cases consistency for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.dcn(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.suf.dcv` *List cases deviant with regards to coverage for sufficiency.*

Description

A function that extracts cases deviant with regards to coverage for sufficiency from an object of class "qca".

Usage

```
cases.suf.dcv(results, outcome, neg.out=FALSE, sol = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Return deviant cases coverage for sufficiency for the parsimonious solution:

cases.suf.dcv(results = sol_yp, outcome = "EXPORT")
```

```
# Return deviant cases coverage for sufficiency for the intermediate solution:

cases.suf.dcv(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Return deviant cases coverage for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.dcv(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.suf.iir` *List individually irrelevant cases for sufficiency.*

Description

A function that extracts individually irrelevant cases for sufficiency from an object of class "qca".

Usage

```
cases.suf.iir(results, outcome, neg.out=FALSE, sol = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

Schneider, C. Q., Rohlfing I. 2016. Case Studies Nested in Fuzzy-Set QCA on Sufficiency: Formalizing Case Selection and Causal Inference. Sociological Methods & Research 45(3): 526-68

Rohlfing, I., Schneider C. Q. 2016. A Unifying Framework for Causal Analysis in Set-Theoretic Multi-Method Research. Sociological Methods & Research: 1-27. DOI: 10.1177/0049124115626170

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))



# Return individually irrelevant cases for sufficiency for the parsimonious solution:

cases.suf.iir(results = sol_yp, outcome = "EXPORT")

# Return individually irrelevant cases cases for sufficiency for the intermediate solution:

cases.suf.iir(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
```

```
                details = TRUE, show.cases = TRUE)

# Return individually irrelevant cases for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.iir(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.suf.typ` *List cases typical with regards to sufficiency.*

Description

A function that extracts cases typical with regards to sufficiency from an object of class "qca".

Usage

```
cases.suf.typ(results, outcome, neg.out=FALSE, sol = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Return typical cases for sufficiency for the parsimonious solution:

cases.suf.typ(results = sol_yp, outcome = "EXPORT")

# Return typical cases for sufficiency for the intermediate solution:

cases.suf.typ(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Return typical cases for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.typ(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.suf.typ.fct` *List cases typical with regards to sufficiency for each focal conjunct.*

Description

A function that extracts cases typical with regards to sufficiency from an object of class "qca".

Usage

```
cases.suf.typ.fct(results, outcome, term = 1, neg.out=FALSE, sol = 1)
```

Arguments

results	An object of class "qca".
outcome	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the minimize() result from the sufficiency analysis of the negated outcome in the argument results. Changing the name in the argument outcome or using a tilde is not necessary.
neg.out	Logical. Should the negated outcome be used?
term	A numeric vector where the first number indicates the number of the term according to the order in the "qca" object.
sol	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.

Author(s)

Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:
```

```
sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Return typical cases for sufficiency for the parsimonious solution:

cases.suf.typ.fct(results = sol_yp, outcome = "EXPORT")

# Return typical cases for sufficiency for the intermediate solution:

cases.suf.typ.fct(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Return typical cases for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.typ.fct(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.suf.typ.most` *Lists most typical cases with regards to sufficiency.*

Description

A function that extracts most typical cases with regards to sufficiency from an object of class "qca".

Usage

```
cases.suf.typ.most(results, outcome, neg.out = FALSE, sol = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?

`sol` A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Return most typical cases for sufficiency for the parsimonious solution:

cases.suf.typ.most(results = sol_yp, outcome = "EXPORT")

# Return most typical cases for sufficiency for the intermediate solution:

cases.suf.typ.most(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
```

```
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)


# Return most typical cases for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.typ.most(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.suf.typ.unique` *Lists uniquely typical cases with regards to sufficiency.*

Description

A function that extracts uniquely typical cases with regards to sufficiency from an object of class "qca".

Usage

```
cases.suf.typ.unique(results, outcome, neg.out = FALSE, sol = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Return uniquely typical cases for sufficiency for the parsimonious solution:

cases.suf.typ.unique(results = sol_yp, outcome = "EXPORT")

# Return uniquely typical cases for sufficiency for the intermediate solution:

cases.suf.typ.unique(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)


# Return uniquely typical cases for sufficiency for the second parsimonious solution
# for the absence of the outcome:

cases.suf.typ.unique(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`cases.theory.evaluation`

Names of cases in the intersections between theory and the empirical solution.

Description

Function that returns names of cases in the intersections between theory and the empirical solution.

Usage

```
cases.theory.evaluation(theory_data)
```

Arguments

theory_data	A data frame with the intersections between theory and empirics. Usually the output of the theory.data function should be inputed.

Author(s)

Ioana-Elena Oana and Juraj Medzihorsky

References

Ragin, C. C. 1987. The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies. Berkeley: University of California Press, pp. 118-121.

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3.

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Specify the theory. Let's assume the theory says that the
# absence of EMP and the presence of MA is sufficient for EXPORT:

t<-"~EMP*MA"

# Get membership of cases in the intersections between theory and
# the second intermediate solution and place it in an object:

INT <- theory.data(theory = t, empirics = sol_yi, outcome = "EXPORT", sol = 2)
```

```
# Get the names of cases in the intersections between theory and the empirical solution
# using the output of the theory.data function:

cases.theory.evaluation(INT)
```

`cluster` *Diagnostic tool for clustered data*

Description

Function returns pooled, within, and between consistencies for the relationship between two sets, for an object of class "qca", and for a Boolean expression.

Usage

```
cluster(data=NULL, results, outcome, unit_id, cluster_id, sol = 1, necessity = FALSE)
```

Arguments

`data`	A data frame in the long format containing both a colum with the unit names and a column with the cluster names. Column names should be in capital letters.
`results`	An object of class "qca". For performing cluster diagnostics of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome. The argument results can also be a vector, a character string, or a boolean expression of the form e.g. "A*~B + ~B*C".
`outcome`	A character string with the name of the outcome in capital letters. When performing cluster diagnostics of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. When performing cluster diagnostics for boolean expressions or vectors the negated outcome can be used by inserting a tilde in the outcome name in the argument `outcome`. The outcome can also be a vector.
`unit_id`	A character string with the name of the vector containing the units (e.g. countries).
`cluster_id`	A character string with the name of the vector containing the clustering units (e.g. years).
`sol`	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.
`necessity`	Logical. Perform the diagnostic for the relationship of necessity?

Author(s)

Ioana-Elena Oana

References

Garcia-Castro, Roberto, and Miguel A. Arino. 2016. "A General Approach to Panel Data Set-Theoretic Research."" Journal of Advances in Management Sciences & Information Systems 2: 6376.

See Also

minimize

Examples

```
# Import your clustered data in the long format.
# For example:

data(SCHLF)

# Get the intermediate solution:

sol_yi <- minimize(SCHLF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Get pooled, within, and between consistencies for the second intermediate solution:

cluster(SCHLF, sol_yi, "EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR", sol = 2)

# or:

cluster(SCHLF, sol_yi, "EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR", sol = "c1p1i2")

# Get pooled, within, and between consistencies for EMP as necessary for EXPORT:

cluster(SCHLF, results="EMP", outcome="EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR", necessity=TRUE)
# or:

cluster(results=SCHLF$EMP, outcome=SCHLF$EXPORT, unit_id = SCHLF$COUNTRY,
              cluster_id = SCHLF$YEAR, necessity=TRUE)

# Get pooled, within, and between consistencies for ~EMP as necessary for EXPORT:

cluster(SCHLF, results="~EMP", outcome="EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR", necessity=TRUE)
```

```
# or:

cluster(results=1-SCHLF$EMP, outcome=SCHLF$EXPORT, unit_id = SCHLF$COUNTRY,
              cluster_id = SCHLF$YEAR, necessity=TRUE)

# Get pooled, within, and between consistencies for EMP*~MA*STOCK as sufficient for EXPORT:

cluster(SCHLF, "EMP*~MA*STOCK", "EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR")

# Get pooled, within, and between consistencies for EMP*MA + ~STOCK as sufficient for ~EXPORT:

cluster(SCHLF, "EMP*MA + ~STOCK", "~EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR")
```

`cluster.diagnostics` *Diagnostic tool for clustered data.*

Description

Function returns pooled, within, and between consistencies and coverages for the relationship between two sets.

Usage

```
cluster.diagnostics(x, y, unit, cluster, necessity= FALSE)
```

Arguments

`x`	A vector containing the condition.
`y`	A vector containing the outcome.
`unit`	A vector containing the names of the units (i.e. countries).
`cluster`	A vector containing the names of the clustering unit (i.e. years).
`necessity`	Logical. Perform the diagnostic for the relationship of necessity?

Author(s)

Ioana-Elena Oana and Juraj Medzihorsky

References

Garcia-Castro, A., Arino, M. A.. 2013. A General Approach to Panel Data Set-Theoretic Research. COMPASSS Working Paper 2013-76

See Also

`minimize`

Examples

```
# Import your clustered data in the long format.
# For example:

data(SCHLF)

# Get pooled, within, and between consistencies and coverages
# for STOCK as necessary for EXPORT:


cluster.diagnostics(SCHLF$STOCK, SCHLF$EXPORT, unit = SCHLF$COUNTRY,
                    cluster = SCHLF$YEAR, necessity= TRUE)
```

`cluster.minimize` *Diagnostic tool for clustered data*

Description

Function returns pooled, within, and between consistencies for an object of class "qca".

Usage

```
cluster.minimize(results, data, outcome, unit_id, cluster_id, sol = 1)
```

Arguments

`results`	An object of class "qca". For performing cluster diagnostics of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome.
`data`	A data frame with column names in capital letters.
`outcome`	A character string with the name of the outcome in capital letters. When performing cluster diagnostics of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`unit_id`	A character string with the name of the vector containing the units (i.e. countries).
`cluster_id`	A character string with the name of the vector containing the clustering units (i.e. years).
`sol`	A numeric vector where the first number indicates the number of the solution according to the order in the "qca" object.

Author(s)

Ioana-Elena Oana and Juraj Medzihorsky

References

Garcia-Castro, A., Arino, M. A.. 2013. A General Approach to Panel Data Set-Theoretic Research. COMPASSS Working Paper 2013-76

See Also

minimize

Examples

```
# Import your clustered data in the long format.
# For example:

data(SCHLF)

# Get the intermediate solution:

sol_yi <- minimize(SCHLF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Get pooled, within, and between consistencies for the second intermediate solution:

cluster.minimize(results = sol_yi, data = SCHLF, outcome = "EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR", sol = 2)
```

EMMF *EMMFnegger (2011)*

Description

The EMMF data frame has 19 rows and 8 sets

Usage

```
data(EMMF)
```

Format

A data frame with 19 observations on the following 8 sets.

country a factor with levels Australia Austria Belgium Canada Denmark Finland France Germany Ireland Italy Netherlands NewZealand Norway Portugal Spain Sweden Switzerland UK USA

s a numeric vector. Condition, state-society relationships.

`c` a numeric vector. Condition, non-market coordination.

`l` a numeric vector. Condition, strength of the labour movement.

`r` a numeric vector. Condition, religious denomination.

`p` a numeric vector. Condition, strenght of religious parties.

`v` a numeric vector. Condition, institutional veto points.

`jsr` a numeric vector. Outcome, job-secturity regulations.

Details

Data are used by Emmenegger (2011) to analyze job-security regulations in Western democracies. The data are fuzzy-sets.

References

Emmenegger, P. (2011) "Job-security regulations in Western democracies", European Journal of Political Research 50, pp. 336-364.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(EMMF)
```

---

esa *Function that performs the Enhanced Standard Analysis.*

Description

Function that performs the Enhanced Standard Analysis.

Usage

```
esa(oldtt, nec_cond, imposs_LR, contrad_rows)
```

Arguments

`oldtt`	A truthTable object.
`nec_cond`	A vector of character strings containing the necessary conditions. Conditions should be capitalized and negated conditions should be inserted with a "~". Unions of conditions are performed with a "+". Using this argument, logical remainder rows that contradict the statement of necessity will not be used in the analysis (i.e. OUT will be set to 0 in the truth table).

`imposs_LR` A vector of character strings containing the impossible logical remainders. Conditions should be capitalized and negated conditions should be inserted with a "~". Intersections of conditions are performed with a "*". Using this argument, logical remainder rows containing the particular intersection specified will not be used in the analysis (i.e. OUT will be set to 0 in the truth table).

`contrad_rows` A vector containing the names of the rows that are contradictory. Using this argument, all rows with the names specified (both logical remainders and rows containing empirical information) will not be used in the analysis (i.e. OUT will be set to 0 in the truth table).

Value

It returns a new truth table in which all truth table rows are set to outcome value 0 that would otherwise present untenable assumptions.

Author(s)

Ioana-Elena Oana

References

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 8.

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the truth table for the presence of the outcome:

TT_y <- truthTable(SCHF, outcome = "EXPORT",
                   conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                   incl.cut = .9,
                   complete = TRUE,
                   PRI = TRUE,
                   sort.by = c("out", "incl", "n"))

# Exclude condition STOCK + MA and condition EMP as necessary for EXPORT
# Exclude all remainder rows containing the combination BARGAIN*~OCCUP
# Exclude the rows "19", "14", "46", "51" as contradictory:

newtt <- esa(oldtt = TT_y, nec_cond = c("STOCK+MA", "EMP"),
          imposs_LR = "BARGAIN*~OCCUP", contrad_rows = c("19", "14", "46", "51"))

# The truth table newly created can afterwards be used in further analyses
```

FakeCS *Fake crisp-set data*

Description

The FakeCS data frame has 30 rows and 5 sets

Usage

```
data(FakeCS)
```

Format

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

`y` a numeric vector. Outcome with 2 categories (crisp-set).

`j` a numeric vector. Condition with 2 categories (crisp-set).

`z` a numeric vector. Condition with 2 categories (crisp-set).

`w` a numeric vector. Condition with 2 categories (crisp-set).

`k` a numeric vector. Condition with 2 categories (crisp-set).

Details

The data frame has only exercise purpuses to let the user learn how to perform crisp-set QCA in R.

References

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Examples

```
data(FakeCS)
```

FakeMV *Fake data for mvQCA*

Description

mvQCA data frame has 25 rows and 4 sets.

Usage

```
data(FakeMV)
```

Format

A data frame with 25 observations on the following 4 sets.

Y a numeric vector. Outcome with 2 categories (crisp).

A a numeric vector. Condition with 2 categories (crisp).

B a numeric vector. Condition with 3 categories (multi-value).

C a numeric vector. Condition with 3 categories (multi-value).

Details

The data frame has only exercise purpuses to let the user learn how to perform mvQCA in R.

References

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Examples

```
data(FakeMV)
```

FSR *Freitag and Schlicht (2009)*

Description

The FSR data frame has 16 rows and 8 sets

Usage

```
data(FSR)
```

Format

A data frame with 16 observations on the following 8 sets.

`integrated_comp_schools` a numeric vector. Condition, percentage of Pupils Enrolled in Integrated Comprehensive Schools,

`coop_comp_schools` a numeric vector. Condition, percentage of Pupils Enrolled in Cooperative Comprehensive Schools.

`full_day_schools` a numeric vector. Condition, percentage of Pupils Enrolled in All-Day Schools

`child_care` a numeric vector. Condition, ratio of Number of Child Care Facilities to Total Population between 0 and 6 Years (percent).

`pre_schools` a numeric vector. Condition, ratio of pupils Enrolled in Pre-School to Total 6-Year-Old Population (per cent)

`early_tracking` a numeric vector. Condition, onset of Tracking, Legal Regulation.

`outcome` a numeric vector. Outcome, high Degree of Social Inequality Cases in Education.

`indep_hauptschule` a numeric vector. Condition, autonomy of the Hauptschule.

Details

Data are used by Freitag and Schlicht (2009) to analyze social inequality in education. The data are raw scores.

References

Freitag, M, and Schlicht, R. (2009) "Educational Federalism in Germany: Foundations of Social Inequalities in Education", Governance 22(1), pp. 47-72.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(FSR)
```

`indirectCalibration` *Function performing the indirect calibration*

Description

indirectCalibration is a function for the indirect calibration procedure as described by Ragin (2008). It uses a binomial or a beta regression for tranforming raw scores into calibrated scores. In our opinion, using a fractional polynomial may not be appropriate to this case. In fact, we do not deal with proportions. This function requires the package `betareg`.

Usage

```
indirectCalibration(x, x_cal, binom = TRUE)
```

Arguments

`x`	vector of raw scores.
`x_cal`	vector of theoretically calibrated scores.
`binom`	logical. If indirect calibration has to be performed using binomial regression or beta regression. The default is `TRUE`, which means that binomial regression is used.

Value

It returns a vector of indirectly calibrated values.

Author(s)

Mario Quaranta

References

Ragin, C. C. (2008) Redesigning Social Inquiry: Fuzzy Sets and Beyond, The Chicago University Press: Chicago and London.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
# Generate fake data
set.seed(4)
x <- runif(20, 0, 1)
```

```
# Find quantiles
quant <- quantile(x, c(.2, .4, .5, .6, .8))

# Theoretical calibration
x_cal <- NA
x_cal[x <= quant[1]] <- 0
x_cal[x > quant[1] & x <= quant[2]] <- .2
x_cal[x > quant[2] & x <= quant[3]] <- .4
x_cal[x > quant[3] & x <= quant[4]] <- .6
x_cal[x > quant[4] & x <= quant[5]] <- .8
x_cal[x > quant[5]] <- 1
x_cal

# Indirect calibration (binomial)
a <- indirectCalibration(x, x_cal, binom = TRUE)

# Indirect calibration (beta regression)
b <- indirectCalibration(x, x_cal, binom = FALSE)

# Correlation
cor(a, b)

# Plot
plot(x, a); points(x, b, col = "red")
```

KAF *Koenig-Archibugi (2004)*

Description

The KAF data frame has 13 rows and 5 sets

Usage

```
data(KAF)
```

Format

A data frame with 13 observations on the following 5 sets.

`supranat` a numeric vector. Government support for supranational CFSP.

`identmass` a numeric vector. European identity of the general public.

`conform` a numeric vector. Policy conformity.

`region` a numeric vector. Regional governance.

`capab` a numeric vector. Material capabilities.

Details

Data are used by Koenig-Archibugi (2004) to analyze government preferences for istitutional change in EU foreign and security policy. Data are fuzzy-sets.

References

Koenig-Archibugi, M. (2004) "Explaining Government Preferences for Istitutional Change in EU Foreign and Security Policy", International Organization 58, pp.137-174.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(KAF)
```

LIPC *Lipset (1959), crisp-set*

Description

The LIPC data frame has 18 rows and 6 sets

Usage

```
data(LIPC)
```

Format

A data frame with 18 observations on the following 6 sets.

`DEVELOPED` a numeric vector. Condition, economically developed country.

`URBAN` a numeric vector. Condition, urbanized countries.

`LITERATE` a numeric vector. Condition, countries with high literacy rate.

`INDUSTRIAL` a numeric vector. Condition, Industrialized countries.

`GOVSTAB` a numeric vector. Condition, politically stable countries.

`SURVIVED` a numeric vector. Outcome, survival of democracy during the inter-war period.

Details

Data used by Ragin (2009) to illustrates the variants of QCA. Originally by Lipset (1959). Data are crisp-sets.

References

Lipset, S. M. (1959) "Some Social Requisites of Democracy: Economic Development and Political Legitimacy", American Political Science Review 53, pp. 69-105.

Ragin, C. C. (2009) "Qualitative Comparative Analysis. Using Fuzzy Sets (fsQCA)." In Rihoux, B., and Ragin, C. C. (eds.) Configurational Comparative Methods. Qualitative Comparative Analysis (QCA) and Related Techniques. Thousand Oaks, CA and London: Sage, pp. 87-121.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann.

Examples

```
data(LIPC)
```

LIPF *Lipset (1959), fuzzy-set*

Description

The LIPF data frame has 18 rows and 6 sets

Usage

```
data(LIPF)
```

Format

A data frame with 18 observations on the following 6 sets.

`Survived` a numeric vector. Outcome, survival of democracy during the inter-war period.

`DEVELOPED` a numeric vector. Condition, economically developed countries.

`URBAN` a numeric vector. Condition, urbanized countries.

`LITERATE` a numeric vector. Condition, countries with high literacy rate.

`INDUSTRIAL` a numeric vector. Condition, industrialized countries.

`STABLE` a numeric vector. Condition, politically stable countries.

Details

Data used by Ragin (2009) to illustrates the variants of QCA. Originally by Lipset (1959). Data are fuzzy-sets.

References

Lipset, S. M. (1959) "Some Social Requisites of Democracy: Economic Development and Political Legitimacy", American Political Science Review 53, pp. 69-105.

Ragin, C. C. (2009) "Qualitative Comparative Analysis. Using Fuzzy Sets (fsQCA)." In Rihoux, B., and Ragin, C. C. (eds.) Configurational Comparative Methods. Qualitative Comparative Analysis (QCA) and Related Techniques. Thousand Oaks, CA and London: Sage, pp. 87-121.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann.

Examples

```
data(LIPF)
```

LIPR *Lipset (1959), raw data*

Description

The LIPR data frame has 18 rows and 6 variables

Usage

```
data(LIPR)
```

Format

A data frame with 18 observations on the following 6 variables.

SURVIVED a numeric vector. Outcome, survival of democracy during the inter-war period.

DEVELOPED a numeric vector. Condition, level of economic development.

URBAN a numeric vector. Condition, level of urbanization.

LITERATE a numeric vector. Condition, level of literacy.

INDUSTRIAL a numeric vector. Condition, level of industrialization.

UNSTABLE a numeric vector. Condition, politically stable countries.

Details

Data used by Ragin (2009) to illustrates the variants of QCA. Originally by Lipset (1959). Data are raw-scores.

References

Lipset, S. M. (1959) "Some Social Requisites of Democracy: Economic Development and Political Legitimacy", American Political Science Review 53, pp. 69-105.

Ragin, C. C. (2009) "Qualitative Comparative Analysis. Using Fuzzy Sets (fsQCA)." In Rihoux, B., and Ragin, C. C. (eds.) Configurational Comparative Methods. Qualitative Comparative Analysis (QCA) and Related Techniques. Thousand Oaks, CA and London: Sage, pp. 87-121.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann.

Examples

```
data(LIPR)
```

`matches.suf.dcviir` *Match deviant coverage cases and individually irrelevant cases with regards to sufficiency.*

Description

A function that matches deviant coverage cases and individually irrelevant cases with regards to sufficiency.

Usage

```
matches.suf.dcviir(results, outcome, neg.out=FALSE, sol=1, max_pairs = 5)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.
`max_pairs`	Maximum number of pairs to extract.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods Research 42(4): 559-597

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Match deviant coverage cases and individually irrelevant cases for the parsimonious solution:

matches.suf.dcviir(results = sol_yp, outcome = "EXPORT")

# Match deviant coverage cases and individually irrelevant cases for the parsimonious solution
# and return only the best 3 pairs:

matches.suf.dcviir(results = sol_yp, outcome = "EXPORT", max_pairs=3)

# Match deviant coverage cases and individually irrelevant cases for the intermediate solution:

matches.suf.dcviir(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)
```

```
# Match deviant coverage cases and individually irrelevant cases for
# the second parsimonious solution for the absence of the outcome:

matches.suf.dcviir(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`matches.suf.typdcn` *Match typical cases and deviant consistency cases with regards to sufficiency.*

Description

A function that matches typical cases and deviant consistency cases with regards to sufficiency.

Usage

```
matches.suf.typdcn(results, outcome, neg.out=FALSE, sol=1, max_pairs = 5)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.
`max_pairs`	Maximum number of pairs to extract.

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods Research 42(4): 559-597

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Match typical cases and deviant consistency cases for the parsimonious solution:

matches.suf.typdcn(results = sol_yp, outcome = "EXPORT")

# Match typical cases and deviant consistency cases for the parsimonious solution
# and return only the best 3 pairs:

matches.suf.typdcn(results = sol_yp, outcome = "EXPORT", max_pairs=3)

# Match typical cases and deviant consistency cases for the intermediate solution:

matches.suf.typdcn(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Match typical cases and deviant consistency cases for the second
# parsimonious solution for the absence of the outcome:

matches.suf.typdcn(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, sol = 2)
```

`matches.suf.typiir` *Match typical cases and individually irrelevant cases for each conjunct in a sufficient term.*

Description

A function that matches typical cases and individually irrelevant cases with regards to sufficiency for each INUS condition.

Usage

```
matches.suf.typiir(results, outcome, term = 1, neg.out=FALSE,
                   sol=1, max_pairs = 5)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`term`	A numeric vector where the first number indicates the number of the term according to the order in the "qca" object.
`neg.out`	Logical. Should the negated outcome be used?
`sol`	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.
`max_pairs`	Maximum number of pairs to extract.

Author(s)

Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods Research 42(4): 559-597

Schneider, C. Q., Rohlfing I. 2016. Case Studies Nested in Fuzzy-Set QCA on Sufficiency: Formalizing Case Selection and Causal Inference. Sociological Methods & Research 45(3): 526-68

Rohlfing, I., Schneider C. Q. 2016. A Unifying Framework for Causal Analysis in Set-Theoretic Multi-Method Research. Sociological Methods & Research: 1-27. DOI: 10.1177/0049124115626170

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)


# Match typical cases and individually irrelevant cases for each conjunct
# for the first term (default) of the parsimonious solution and return only the best 3 pairs:

matches.suf.typiir(results = sol_yp, outcome = "EXPORT", max_pairs=2)

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Match typical cases and individually irrelevant cases for each conjunct for
# the first term of the second parsimonious solution for the absence of the outcome:

matches.suf.typiir(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, term = 1, sol = 2)

# since it has only one condition, it cannot find pairs based on the INUS principle!
```

`matches.suf.typtyp` *Match typical cases for each conjunct in a sufficient term..*

Description

A function that matches typical cases with regards to sufficiency for each INUS condition.

Usage

```
matches.suf.typtyp(results, outcome, term = 1, neg.out=FALSE,
                   sol=1, max_pairs = 5)
```

Arguments

results	An object of class "qca".
outcome	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the minimize() result from the sufficiency analysis of the negated outcome in the argument results. Changing the name in the argument outcome or using a tilde is not necessary.
term	A numeric vector where the first number indicates the number of the term according to the order in the "qca" object.
neg.out	Logical. Should the negated outcome be used?
sol	A numeric vector where the first number indicates the number of the solution in case of model ambiguity according to the order in the "qca" object.
max_pairs	Maximum number of pairs to extract.

Author(s)

Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods Research 42(4): 559-597

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))
```

```
# Match typical cases for each conjunct for the second term of the parsimonious solution:

matches.suf.typtyp(results = sol_yp, outcome = "EXPORT", term = 2)

# Match typical cases for each conjunct for the first term (default) of
# the parsimonious solution and return only the best 3 pairs:

matches.suf.typtyp(results = sol_yp, outcome = "EXPORT", max_pairs=3)

# Match typical cases for each conjunct for the first term (default)
# of the intermediate solution:

matches.suf.typtyp(results = sol_yi, outcome = "EXPORT")

# Get the parsimonious solution for the absence of the outcome:

sol_nyp <- minimize(SCHF, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Match typical cases for each conjunct of the second term
# of the second parsimonious solution for the absence of the outcome:

matches.suf.typtyp(results = sol_nyp, outcome = "EXPORT", neg.out = TRUE, term = 2, sol = 2)
```

`mmr` *Function for performing set-theoretic multi-method research.*

Description

A function that selects best available cases for single case studies and best pairs of matching cases for comparative case studies.

Usage

```
mmr(results, outcome, neg.out = FALSE, sol = 1, match = NULL,
      cases = NULL, max_pairs = 5, term = 1)
```

Arguments

`results`	An object of class "qca".
`outcome`	A character string with the name of the outcome.
`neg.out`	Logical. Should the negated outcome be used?

sol	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.
match	Logical. Should comparative MMR be used?
cases	A numerical vector indicating the type of cases to be returned. For single case studies: 1 = typical cases, 2 = typical cases for each focal conjunct in a sufficient term, 3 = deviant consistency, 4 = deviant coverage, 5 = individually irrelevant, 6 = all of the above; For comparative case studies: 1 = Typical-Typical for each focal conjuct in a sufficient term, 2 = Typical-IIR for each focal conjuct in a sufficient term, 3 = Typical-Dev.Cons., 4 = Dev.Cov.-IIR; 5 = all of the above;
max_pairs	Maximum number of pairs to extract.
term	A numeric vector where the first number indicates the number of the term according to the order in the "qca" object.

Author(s)

Ioana-Elena Oana

References

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods and Research 42(4): 559-97

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:

sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
```

```
                details = TRUE, show.cases = TRUE)

# Get typical cases for each focal conjunct in the third term of the parsimonious solution:

mmr(results = sol_yp, outcome = "EXPORT", match=FALSE, cases=2, term = 3)

# Get matching typical-typical cases for the second term of the parsimonious solution:

mmr(results = sol_yp, outcome = "EXPORT", match=TRUE, cases=1, term = 2)

# Get matching typical-DCN cases:

mmr(results = sol_yp, outcome = "EXPORT", match=TRUE, cases=3)
```

PENF *Pennings (2003)*

Description

The PENF data frame has 45 rows and 5 sets

Usage

```
data(PENF)
```

Format

A data frame with 45 observations on the following 5 sets.

`K` a numeric vector. Outcome, constitutional control.

`C` a numeric vector. Condition, consensus democracy.

`P` a numeric vector. Condition, presidentialism.

`N` a numeric vector. Condition, new democracy.

`R` a numeric vector. Condition, rigid constrtitution.

Details

Data used by Pennings (2009) to explain constitutional control. Data are fuzzy-sets.

References

Pennings, P. (2009) "Beyond Dichotomous Explanations: Explaining Constitutional control of the Executive with Fuzzy-sets", European Journal of Political Research 42, pp. 541-567.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann.

Examples

```
data(PENF)
```

`pimdata` *Function to extract prime implicants table from object of class "qca"*

Description

A function that displays each case's set membership scores in each sufficient term, the solution formula, and the outcome from an object of class "qca".

Usage

```
pimdata(results, outcome, sol = 1)
```

Arguments

`results`	An object of class "qca". For performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome.
`outcome`	A character string with the name of the outcome in capital letters. When performing pimdata of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`sol`	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.

Value

A table with set memberships.

`solution_formula`	The solution formula.
`out`	Membership in the outcome.

Author(s)

Ioana-Elena Oana and Juraj Medzihorsky

See Also

`minimize` `pimplot`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Get the prime implicants table for the parsimonious solution:

pimdata(results = sol_yp, outcome = "EXPORT")

# Get the prime implicants table for the first intermediate solution:

pimdata(results = sol_yi, outcome = "EXPORT", sol = 1)
```

`pimplot` *Prime implicants, truth table rows, and necessity plots.*

Description

A function that displays XY plots for each sufficient term and the solution formula plotted against the outcome from an object of class "qca" (obtained by using the `minimize` function in package `QCA`). The function can also plot truth table rows against the outcome. Additionally, the function can plot results obtained from necessity analyses using an object of class "sS" (obtained by using the `superSubset` function in package `QCA`).

Usage

```
pimplot(data = NULL, results, outcome, neg.out=FALSE, incl.tt=NULL, ttrows= c(),
        necessity=FALSE, sol=1, case_labels=TRUE, all_labels=FALSE,
        lab_color=rgb(0,0,0,0.5), lab_jitter=FALSE)
```

Arguments

data	For analyses of sufficiency, providing a dataframe is not necessary. For analyses of necessity on objects of class "sS, you need to provide a dataframe with the name of the outcome and of the conditions in capital letters.
results	An object of class "qca" when necessity is FALSE. An object of class "sS" when necessity is TRUE. For performing pimplots of the sufficient solution for the negated outcome one must only use the minimize() result from the sufficiency analysis of the negated outcome together with neg.out set to TRUE.
outcome	A character string with the name of the outcome in capital letters. When performing pimplots of the sufficient solution for the negated outcome one must only use the minimize() result from the sufficiency analysis of the negated outcome in the argument results together with neg.out set to TRUE. Changing the name in the argument outcome or using a tilde is not necessary.
neg.out	Logical. Should the negated outcome be used?
incl.tt	A numerical vector of length 1 specifying the row consistency threshold above which it should plot truth table rows. By default it is NULL and the function will produce plots using "qca" or "sS" objects. If a numerical value is specifyied, then it automatically only plots truth table rows above that consistency value. N.B. This argument cannot be used simultaneously with the ttrows argument.
ttrows	A vector of character strings specifying the names of the truth table rows to be printed. By default this vector is empty and the function will produce plots using "qca" or "sS" objects. If a value is specifyied, then it automatically only plots those particular truth table rows. N.B. This argument cannot be used simultaneously with the incl.tt argument.
necessity	logical. It indicates if the output should be for the results of sufficiency or necessity analyses. By default, FALSE, the function works with an object of class "qca" obtained from the minimize function in package QCA. When it set to TRUE the function returns plots for an object of class "sS" obtained from the superSubset function in package QCA.
sol	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.
case_labels	Logical. Print case labels for relevant cases?
all_labels	Logical. Print ALL case labels?
lab_color	Color of the labels.
lab_jitter	Logical. Add jitter to label positions?

Value

XY plots.

Author(s)

Ioana-Elena Oana and Juraj Medzihorsky

References

Haesebrouck, T. (2015) Pitfalls in QCA's consistency measure. Journal of Comparative Politics 2:65-80.

Schneider, C. Q., Rohlfing, I. 2013. Combining QCA and Process Tracing in Set-Theoretic Multi-Method Research. Sociological Methods Research 42(4): 559-597

See Also

minimize pimdata

Examples

```
# Import your data. For example:

data(SCHF)

# Get the parsimonious solution:


sol_yp <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Plot the prime implicants of the parsimonious solution:

pimplot(data = SCHF, results = sol_yp, outcome = "EXPORT")

# Plot the prime implicants of the first intermediate solution showing all case labels:

pimplot(data = SCHF, results = sol_yi, outcome = "EXPORT", sol = 1, all_labels = TRUE)

# Plot all truth table rows with a consistency higher than 0.9:

pimplot(data=SCHF, results = sol_yi, incl.tt=0.9, outcome = "EXPORT", sol = 1)

# Plot truth table rows "60" and "61":

pimplot(data=SCHF, results = sol_yi, ttrows =c("60","61"),
        outcome = "EXPORT", sol = 1)

# For plotting results of necessity analyses using superSubset,
```

```
# the first stept is to obtain an "sS" object:

SUPSUB <- superSubset(SCHF, outcome="EXPORT",
                      conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                      relation = "necessity", incl.cut = 0.8)
SUPSUB

# This can be imputed as result and necessity should be set to \code{TRUE}:

pimplot(data = SCHF, results = SUPSUB, outcome = "EXPORT", necessity = TRUE)
```

`print.casestheoryeval` *Print a casestheoryeval Object*

Description

Information regarding the names and numbers of cases in the intersections between theory and empirics.

Usage

```
  ## S3 method for class 'casestheoryeval'
print(x, ...)
```

Arguments

`x`	an object inheriting from class `cases.theory.evaluation`, representing the name and numbers of cases in the various intersections between theory and empirics.
`...`	optional arguments passed to `print.default`; see the documentation on that method function

Author(s)

Ioana-Elena Oana

References

Ragin, C. C. 1987. The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies. Berkeley: University of California Press, pp. 118-121.

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3.

See Also

`minimize`

print.clusterdiagnostics *Print a clusterdiagnostics Object*

Description

Information regarding pooled, within, and between consistencies for for the relationship between two sets in the case of clustered data is printed.

Usage

```
## S3 method for class 'clusterdiagnostics'
print(x, ...)
```

Arguments

x	an object inheriting from class cluster.diagnostics, representing pooled, within, and between consistencies and coverages for the relationship between two sets in the case of clustered data
...	optional arguments passed to print.default; see the documentation on that method function

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Garcia-Castro, A., Arino, M. A.. 2013. A General Approach to Panel Data Set-Theoretic Research. COMPASSS Working Paper 2013-76

See Also

minimize

print.clusterminimize *Print a clusterminimize Object*

Description

Information regarding pooled, within, and between consistencies for an object of class "qca" is printed.

Usage

```
## S3 method for class 'clusterminimize'
print(x, ...)
```

Arguments

x	an object inheriting from class `cluster.minimize`, representing pooled, within, and between consistencies and coverages for an object inheriting from class 'qca' in the case of clustered data
...	optional arguments passed to `print.default`; see the documentation on that method function

Author(s)

Juraj Medzihorsky and Ioana-Elena Oana

References

Garcia-Castro, A., Arino, M. A.. 2013. A General Approach to Panel Data Set-Theoretic Research. COMPASSS Working Paper 2013-76

See Also

`minimize`

`print.matchessuf` *Print a matchessuf Object*

Description

Information regarding matching of cases following the INUS principle.

Usage

```
  ## S3 method for class 'matchessuf'
print(x, ...)
```

Arguments

x	an object inheriting from class `matches.suf.typtyp` and `matches.suf.typiir`, representing matching of cases following the INUS principle.
...	optional arguments passed to `print.default`; see the documentation on that method function

Author(s)

Ioana-Elena Oana

References

Ragin, C. C. 1987. The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies. Berkeley: University of California Press, pp. 118-121.

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3.

See Also

minimize

`print.theoryeval` *Print a theoryeval Object*

Description

Information regarding the intersections between theory and empirics.

Usage

```
  ## S3 method for class 'theoryeval'
print(x, ...)
```

Arguments

`x`	an object inheriting from class `theory.evaluation`, representing the various intersections between theory and empirics.
`...`	optional arguments passed to `print.default`; see the documentation on that method function

Author(s)

Ioana-Elena Oana

References

Ragin, C. C. 1987. The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies. Berkeley: University of California Press, pp. 118-121.

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3.

See Also

minimize

QCAfit *Function calculating the parameters of fit*

Description

QCAfit is a function calculating parameters of fit useful in QCA and fsQCA that are consistency, coverage, PRI, Haesebrouck's consistency, RoN and PRODUCT. It works with both single and multiple conditions.

Usage

```
QCAfit(x, y, cond.lab = NULL, necessity = FALSE, neg.out = FALSE,
              product = FALSE, sol=1, ttrows= c())
```

Arguments

x	A vector containing the values of a condition, a matrix with more than one conditions, or an object of class "qca" when necessity is FALSE and when outcome is specifyied as a character string.
y	A vector containing the values of the output or a character string when y is of class "qca".
cond.lab	is a vector containing the label(s) of the condition(s).
necessity	logical. It indicates if the output should be for sufficient or necessary condition(s). By default, FALSE, the function returns a table of parameters of fit for sufficient condition(s) (Consistency, Coverage, PRI, Haesebrouck's Consistency, and optionally Product). When it set to TRUE the function returns a table of parameters of fit for necessary condition(s) (Consistency, Coverage, Relevance of Necessity).
neg.out	logical. It indicates if the parameters of fit should be computed for the positive or the negative outcome. By default, FALSE, the function returns parameters of fit for the positive outcome .
product	logical. It indicates whether the parameter of fit PRODUCT should be shown. This stands for the product between the consistency sufficiency parameter and the PRI parameter.
sol	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.
ttrows	A vector specifying the names of the truth table rows for which the function reports parameters of fit. For using this option y must be a "qca" object.

Value

It returns a matrix containing the parameters of fit for each condition.

Author(s)

Mario Quaranta and Ioana-Elena Oana

References

Haesebrouck, T. (2015) Pitfalls in QCA's consistency measure. Journal of Comparative Politics 2:65-80.

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

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

See Also

minimize

Examples

```
# Generate fake data
set.seed(1234)

a <- runif(100, 0, 1)
b <- runif(100, 0, 1)
c <- runif(100, 0, 1)
y <- runif(100, 0, 1)

# Only one condition, for sufficiency
QCAfit(a, y, cond.lab = "A")

# With three conditions, for sufficiency
QCAfit(cbind(a, b, c), y, cond.lab = c("A", "B", "C"))

# Only one condition, for necessity
QCAfit(a, y, cond.lab = "A", necessity = TRUE)

# With three conditions, for necessity
QCAfit(cbind(a, b, c), y, cond.lab = c("A", "B", "C"), necessity = TRUE)

# With three conditions and negated output, for sufficiency
QCAfit(cbind(a, b, c), y, cond.lab = c("A", "B", "C"), neg.out = TRUE)

# Load the Schneider data:

data(SCHF)

# Get parameters of fit for condition EMP as necessary for outcome EXPORT:
```

```
QCAfit(SCHF$EMP, SCHF$EXPORT, necessity = TRUE, cond.lab = "EMP")

# Get parameters of fit for condition ~EMP as necessary for outcome ~EXPORT:

QCAfit(1-SCHF$EMP, SCHF$EXPORT, neg.out=TRUE, necessity = TRUE, cond.lab = "~EMP")

# Get parameters of fit for all conditions as necessary for outcome EXPORT:

QCAfit(SCHF[,1:6], SCHF$EXPORT, necessity = TRUE, cond.lab = names(SCHF[,1:6]))

# Obtain the intermediate solution for outcome "EXPORT":

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Get parameters of fit for the second model of the intermediate solution:

QCAfit(x = sol_yi, y = "EXPORT", sol=2)

# Get parameters of fit for truth table rows 2,8, and 10:

QCAfit(x = sol_yi, y = "EXPORT", ttrows=c("2","8","10"))
```

QCAradar *Function for displaying a radar chart.*

Description

Function displays a radar chart for an object of class "qca" or for a boolean expression.

Usage

```
QCAradar(results, outcome= NULL, fit= FALSE, sol = 1)
```

Arguments

results	An object of class "qca". For performing radar charts of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome. The argument results can also be a boolean expression of the form e.g. "A*~B + ~B*C".
outcome	A character string with the name of the outcome in capital letters when results is of type 'qca'. When performing radar charts of the sufficient solution for the negated outcome one must only use the `minimize()`result from the sufficiency analysis of the negated outcome in the argument `results`. Changing the name in the argument `outcome` or using a tilde is not necessary.

`fit` Logical. Print parameters of fit when results is of type 'qca'

`sol` A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.

Author(s)

Ioana-Elena Oana

References

Maerz, F. Seraphine. 2017. "Pathways of Authoritarian Persistence." Paper presented at the CEU Annual Doctoral Conference

See Also

`minimize`

Examples

```
# Import data.
# For example:

data(SCHF)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Display radar chart for the second intermediate solution:

QCAradar(results = sol_yi, outcome = "EXPORT", fit=TRUE, sol = 2)

# Show a radar chart for the following boolean expression "A + ~B*Z*~C"

QCAradar(results = "A + ~B*Z*~C")
```

SAMF *Samford (2010)*

Description

The SAMF data frame has 61 rows and 4 sets

Usage

```
data(SAMF)
```

Format

A data frame with 61 observations on the following 4 sets.

`Y` a numeric vector. Outcome, trade liberalization.

`G` a numeric vector. Condition, lack of weak growth.

`H` a numeric vector. Condition, lack of hyper-inflation

`HorG` a numeric vector. Condition, H or G.

Details

Data are used by Samford (2010) to analyze rapid trade liberalization in Latin America. Data are fuzzy-sets.

References

Samford, S. (2010) "Averting 'Disruption and Reversal': Reassessing the Logic of Rapid Trade Reform in Latin America", Politics and Policy 38(3), pp. 373-407.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(SAMF)
```

SC *Selbst, practicing the truth table algorithm data*

Description

The SC data frame has 130 rows and 4 sets

Usage

```
data(SC)
```

Format

A data frame with 130 observations on the following 4 sets.

A a numeric vector. Condition, crisp-set.

B a numeric vector. Condition, crisp-set.

C a numeric vector. Condition, crisp-set.

Y a numeric vector. Condition, crisp-set.

Details

The authors of the data are Carsten Schnieder and Claudius Wagemann. The data are used in the on-line appendix of "Set-Theoretic Methods for the Social Sciences" to practice the truth table algorithm. Data are crisp-sets.

References

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(SC)
```

SCHF *Schneider et. al. (2010)*

Description

The SCHF data frame has 76 observations and 7 sets

Usage

```
data(SCHF)
```

Format

A data frame with 76 observations on the following 7 sets.

`EMP` a numeric vector. Condition, employment protection.

`BARGAIN` a numeric vector. Condition, collective bargaining.

`UNI` a numeric vector. Condition, university training.

`OCCUP` a numeric vector. Condition, occupational training.

`STOCK` a numeric vector. Condition, stock market size.

`MA` a numeric vector. Condition, mergers and acquisitions.

`EXPORT` a numeric vector. Outcome, export performance in high-tech industries.

Details

Data used by Schneider et. al. (2010) to explain capitalist variety and export performance in high-tech industries. Data are fuzzy-sets.

References

Schneider, M. R., Schulze-Bentrop, C., Paunescu, M. (2010) "Mapping the institutional capital of high-tech firms: A fuzzy-set analysis of capitalist variety and export performance", Journal of International Business Studies 41, pp. 246-266.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Examples

```
data(SCHF)
```

SCHLF *Schneider et. al. (2010)*

Description

The SCHLF data frame has 76 observations and 9 variables.

Usage

```
data(SCHLF)
```

Format

A data frame with 76 observations on the following 9 variables.

EMP a numeric vector. Condition, employment protection.

BARGAIN a numeric vector. Condition, collective bargaining.

UNI a numeric vector. Condition, university training.

OCCUP a numeric vector. Condition, occupational training.

STOCK a numeric vector. Condition, stock market size.

MA a numeric vector. Condition, mergers and acquisitions.

EXPORT a numeric vector. Outcome, export performance in high-tech industries.

COUNTRY a string vector. Name of the country in which the observation was made.

YEAR a numeric vector. Year in which the observation was made.

Details

Data used by Schneider et. al. (2010) to explain capitalist variety and export performance in high-tech industries. Data is saved in the long format and the first 7 variables are fuzzy-sets.

References

Schneider, M. R., Schulze-Bentrop, C., Paunescu, M. (2010) "Mapping the institutional capital of high-tech firms: A fuzzy-set analysis of capitalist variety and export performance", Journal of International Business Studies 41, pp. 246:266;

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Examples

```
data(SCHLF)
```

SDC *Selbst, disappearing necessary condition data*

Description

The SDC data frame has 98 rows and 4 sets

Usage

```
data(SDC)
```

Format

A data frame with 98 observations on the following 4 sets.

A a numeric vector. Condition, crisp-set.

B a numeric vector. Condition, crisp-set.

C a numeric vector. Condition, crisp-set.

Y a numeric vector. Outcome, crisp-set.

Details

The authors of the data are Carsten Schnieder and Claudius Wagemann. The data are used in the on-line appendix of "Set-Theoretic Methods for the Social Sciences" to show the disappearance of a necessary condition. Data are crisp-sets.

References

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(SDC)
```

`theory.data` *Membership of cases in the intersections between theory and the empirical solution.*

Description

Function that returns membership of cases in the intersections between theory and the empirical solution in the form of a data frame.

Usage

```
theory.data(theory, empirics, outcome, sol = 1)
```

Arguments

`theory`	A character string specifying the theory. Conditions should be capitalized and negated conditions should be inserted with a "~". Unions of conditions are performed with a "+", while intersections are performed with a "*".
`empirics`	An object of class 'qca'. When performing analyses for the negated outcome, just use the results from the `minimize()` function for the negation of the outcome.
`outcome`	A character string with the name of the outcome. When performing analyses of the sufficient solution for the negated outcome one must only use the `minimize()` result from the sufficiency analysis of the negated outcome in the argument `empirics`. Changing the name in the argument `outcome` or using a tilde is not necessary.
`sol`	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.

Value

It returns a data frame containing the intersections between theory and empirical findings.

Author(s)

Ioana-Elena Oana and Juraj Medzihorsky

References

Ragin, C. C. 1987. The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies. Berkeley: University of California Press, pp. 118-121

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Specify the theory. Let's assume the theory says that the
# absence of EMP and the presence of MA is sufficient for EXPORT:

t<-"~EMP*MA"

# Get membership of cases in the intersections between theory and
# the second intermediate solution:

theory.data(theory = t, empirics = sol_yi, outcome = "EXPORT", sol = 2)
```

---

theory.evaluation *Performs theory evaluation.*

Description

Function that returns membership of cases in the intersections between theory and the empirical solution in the form of a data frame, the names of cases in the intersections between theory and the empirical solution, and the parameters of fit for these intersections.

Usage

```
theory.evaluation(theory, empirics, outcome, sol = 1, print.data=FALSE)
```

Arguments

theory	A character string specifying the theory. Conditions should be capitalized and negated conditions should be inserted with a "~". Unions of conditions are performed with a "+", while intersections are performed with a "*".

empirics	An object of class 'qca'. When performing analyses for the negated outcome, just use the results from the minimize() function for the negation of the outcome.
outcome	A character string with the name of the outcome. When performing analyses of the sufficient solution for the negated outcome one must only use the minimize() result from the sufficiency analysis of the negated outcome in the argument empirics. Changing the name in the argument outcome or using a tilde is not necessary.
sol	A vector where the first number indicates the number of the conservative or parsimonious solution according to the order in the "qca" object. For more complicated structures of model ambiguity, the intermediate solution can also be specified by using a character string of the form "c1p3i2" where c = conservative solution, p = parsimonious solution and i = intermediate solution.
print.data	Logical. Print also the membership of cases in all the intersections between theory and empirics?

Author(s)

Ioana-Elena Oana

References

Ragin, C. C. 1987. The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies. Berkeley: University of California Press, pp. 118-121

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3

See Also

minimize

Examples

```
# Import your data. For example:

data(SCHF)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))


# Specify the theory. Let's assume the theory says that the
# absence of EMP and the presence of MA is sufficient for EXPORT:

t<-"~EMP*MA"
```

```
# Perform theory evaluation:

theory.evaluation(theory = t, empirics = sol_yi, outcome = "EXPORT", sol = 2, print.data=TRUE)

# Get only the names of the cases and the parameters of fit:

theory.evaluation(theory = t, empirics = sol_yi, outcome = "EXPORT", sol = 2, print.data=FALSE)

# Get only the case names:

TH <- theory.evaluation(theory = t, empirics = sol_yi, outcome = "EXPORT",
                        sol = 2, print.data=FALSE)
TH$cases

# Or only the parameters of fit:

TH$fit
```

`theory.fit` *Parameters of fit for the intersections between theory and the empirical solution.*

Description

Function that returns parameters of fit for the intersections between theory and the empirical solution.

Usage

```
theory.fit(theory_data)
```

Arguments

`theory_data`	A data frame with the intersections between theory and empirics. Usually the output of the theory.data function should be inputed.

Author(s)

Ioana-Elena Oana

References

Schneider, C. Q., Wagemann, C. 2012. Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis. Cambridge: Cambridge University Press, chapter 11.3

See Also

`minimize`

Examples

```
# Import your data. For example:

data(SCHF)

# Get the intermediate solution:

sol_yi <- minimize(SCHF, outcome = "EXPORT",
                conditions = c("EMP","BARGAIN","UNI","OCCUP","STOCK", "MA"),
                incl.cut = .9,
                include = "?",
                details = TRUE, show.cases = TRUE, dir.exp = c(0,0,0,0,0,0))

# Specify the theory. Let's assume the theory says that the
# absence of EMP and the presence of MA is sufficient for EXPORT:

t<-"~EMP*MA"

# Get membership of cases in the intersections between theory and
# the second intermediate solution and place it in an object:

INT <- theory.data(theory = t, empirics = sol_yi, outcome = "EXPORT", sol = 2)

# Get the parameters of fit for the intersections between theory and the empirical solution
# using the output of the theory.data function:

theory.fit(INT)
```

VISC *Vis (2009), crisp set data*

Description

The VISC data frame has 25 rows and 4 sets

Usage

```
data(VISC)
```

Format

A data frame with 25 observations on the following 4 sets.

- P a numeric vector. Condition, weak political positions, with parties in government expecting losses at the next election.
- S a numeric vector. Condition, deteriorating economic situation.
- R a numeric vector. Condition, government dominated by parties from the right of the political spectrum.
- U a numeric vector. Outcome, unpopular reform.

Details

Data are used by Vis (2009) to analyze the pursuit of unpopular reforms by governments. Data are crisp-sets.

References

Vis, B. (2009) "Government and Unpopular Social Policy Reforms: Biting the Bullet or Steering Clear?", European Journal of Political Research 48, pp. 31-57.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(VISC)
```

VISF *Vis (2009), fuzzy set data*

Description

The VISF data frame has 25 rows and 4 sets

Usage

```
data(VISF)
```

Format

A data frame with 25 observations on the following 4 sets.

- p a numeric vector. Condition, weak political positions, with parties in government expecting losses at the next election.
- s a numeric vector. Condition, deteriorating economic situation.
- r a numeric vector. Condition, government dominated by parties from the right of the political spectrum.
- u a numeric vector. Outcome, unpopular reform.

Details

Data are used by Vis (2009) to analyze the pursuit of unpopular reforms by governments. Data are fuzzy-sets.

References

Vis, B. (2009) "Government and Unpopular Social Policy Reforms: Biting the Bullet or Steering Clear?", European Journal of Political Research 48, pp. 31-57.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
data(VISF)
```

`xy.plot` *Function producing enhanced XY plots*

Description

xy.plot produces XY plots and provides values for consistency, Haesebrouck's consistency, coverage, RoN, PRI. Several graphic parameters can be decided by the user.

Usage

```
xy.plot(x, y,
        ylim = c(-0.05, 1.05), xlim = c(-0.05, 1.05),
        pch = 19, col = "black", main = "XY plot",
        ylab = "Outcome", xlab = "Condition",
        mar = c(4, 4, 4, 1), mgp = c(2.2, 0.8, 0),
        cex.fit = 0.6, cex.axis = 0.7, cex.main = 1,
        necessity = FALSE, show.hv = TRUE, show.fit = TRUE,
        pos.fit = "top", case.lab = TRUE, labs = NULL,
        cex.lab = 0.8, offset.x = 0, offset.y = 0,
        pos = 4, srt = 0,
        ident = FALSE)
```

Arguments

`x`	vector containing the condition.
`y`	vector containing the outcome.
`ylim`	limits of y-axis. The default is `c(-0.05, 1.05)`.
`xlim`	limits of x-axis. The default is `c(-0.05, 1.05)`.
`pch`	plotting "character". The default is `19`. See ?pch.
`col`	color for the plotting "character". The default is `"black"`. See ?par.
`main`	an overall title for the plot. The default is `"XY plot"`. See ?title.

ylab	a title for the y-axis. The default is "Outcome". See ?title.
xlab	a title for the x-axis. The default is "Condition". See ?title.
mar	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. The default is c(4, 4, 4, 1). See ?par.
mgp	The margin line (in mex units) for the axis title, axis labels and axis line. Note that mgp[1] affects title whereas mgp[2:3] affect axis. The default is c(2.2, 0.8, 0). See ?par.
cex.fit	character expansion for the parameters of fit. The defaut is 0.6. See ?pch or ?text.
cex.axis	character expansion for the x-axis and y-axis. The defaut is 0.7. See ?pch or ?text.
cex.main	character expansion for the overall title of the plot. The defaut is 1. See ?pch or ?text.
necessity	logical. Indicates if the parameters of fit are calculated for a sufficient or necessary condition. The default is FALSE, therefore it calculates the parameters of fit for sufficiency. To get the parameters of fit for necessary conditions set necessity as TRUE.
show.hv	logical. Indicates if horizontal and vertical lines at 0.5 have to be shown. The default is TRUE.
show.fit	logical. Indicates if parameters of fit have to be shown. The default is TRUE.
pos.fit	character. Indicates the position of the parameters of fit. The positions are "top", which places the parameters of fit outsite the plotting are just below the main title, or "corner", which places the parameters of fit in the corner of the plotting are. The default is pos.fit = "top".
case.lab	logical. Indicates if cases have to be labeled. The default is TRUE.
labs	the vector of case labels. The default is NULL.
cex.lab	character expansion for case labels. The default is 0.8.
offset.x	is a numerical value that sets the offset for case labels position on the x-axis. The default is 0.
offset.y	is a numerical value that sets the offset for case labels position on the y-axis. The default is 0.
pos	a position specifier for the case labels. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the specified coordinates. The default is 4. See ?text.
srt	indicates the rotation of the case labels in degrees. The default is 0. See ?par.
ident	logical. Indicates if identify() has to be used to label the cases. When set to TRUE case.lab has to be FALSE and labels have to be provided by the user. The defaults is FALSE. See ?identify.

Value

It returns an enhanced XY plot.

Author(s)

Mario Quaranta and Ioana-Elena Oana.

References

Haesebrouck, T. (2015) Pitfalls in QCA's consistency measure. Journal of Comparative Politics 2:65-80.

Ragin, C. C. (2008) Redesigning Social Inquiry: Fuzzy Sets and Beyond. The Chicago University Press: Chicago and London.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge Univeristy Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Examples

```
# Generate fake data
set.seed(123)
x <- runif(40, 0, 1)
y <- runif(40, 0, 1)

# Default
xy.plot(x, y)

# With labels
xy.plot(x, y, case.lab = TRUE, labs = 1:40)

# With labels and bigger measures of fit
xy.plot(x, y, case.lab = TRUE, labs = 1:40, cex.fit = 1)

# With labels and bigger title
xy.plot(x, y, case.lab = TRUE, labs = 1:40, cex.main = 1.5)

# Generate fake data the have perfect sufficiency
set.seed(123)
x <- runif(50, 0, 1)
y <- runif(50, 0, 1)

for(i in 1:length(y)) {
    while(x[i] > y[i]) {
        y[i] <- runif(1, 0, 1)
        x[i] <- runif(1, 0, 1)
    }
}

# Default
xy.plot(x, y)

# Load the Schneider data:
```

```
data(SCHF)

# Plot of condition EMP as necessary for outcome EXPORT with case labels
# and names for the plot and axes:

xy.plot(SCHF$EMP, SCHF$EXPORT, necessity = TRUE, labs = rownames(SCHF),
        main = "EMP as necessary for EXPORT", ylab = "EXPORT", xlab = "EMP")
```

`xy.plot.lat` *Function producing enhanced XY plots with Lattice*

Description

`xy.plot.lat` produces XY plots using the `lattice` package and provides values for consistency, Haesebrouck's consistency, coverage, RoN, and PRI. Several graphic parameters can be decided by the user.

Usage

```
xy.plot.lat(x, y,
            ylim = c(-0.05, 1.05), xlim = c(-0.05, 1.05),
            main = "", pch = 19, col = "black", cex.fit = 1,
            ylab = "Outcome", xlab = "Condition",
            pos.fit = "top", strip.cex = 0.8,
            necessity = FALSE, show.fit = TRUE, case.lab = FALSE,
            lab.pos = 4, labs = NULL,
            show.hv = TRUE)
```

Arguments

`x`	vector containing the condition.
`y`	vector containing the outcome.
`ylim`	limits of y-axis. The default is `c(-0.05, 1.05)`.
`xlim`	limits of x-axis. The default is `c(-0.05, 1.05)`.
`main`	an overall title for the plot. The default is `""`. See ?xyplot.
`pch`	plotting "character". The default is `19`. See ?pch or ?xyplot.
`col`	color for the plotting "character". The default is `"black"`. See ?par or ?xyplot.
`cex.fit`	character expansion for the parameters of fit. The defaut is `0.6`. See ?pch or ?text.
`ylab`	a title for the y-axis. The default is `"Outcome"`. See ?title or ?xyplot.
`xlab`	a title for the x-axis. The default is `"Condition"`. See ?title or ?xyplot.
`pos.fit`	character. Indicates the position of the parameters of fit. The positions are `"top"`, which places the parameters of fit in a strip on top of the plotting area, or `"corner"`, which places the parameters of fit in the corner of the plotting are. The default is `pos.fit = "top"`.

strip.cex	character expansion for the parameters of fit when pos.fit = "top". The defaut is 0.8. See ?pch or ?text.
necessity	logical. Indicates if the parameters of fit are calculated for a sufficient or necessary condition. The default is FALSE, therefore it calculates the parameters of fit for sufficiency. To get the parameters of fit for necessary conditions set necessity as TRUE.
show.fit	logical. Indicates if parameters of fit have to be shown. The default is TRUE.
case.lab	logical. Indicates if cases have to be labeled. The default is TRUE.
lab.pos	a position specifier for the case labels. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the specified coordinates. The default is 4. See ?text.
labs	the vector of case labels. The default is NULL.
show.hv	logical. Indicates if horizontal and vertical lines at 0.5 have to be shown. The default is TRUE.

Value

It returns an enhanced XY plot using the lattice package.

Author(s)

Mario Quaranta and Ioana-Elena Oana

References

Haesebrouck, T. (2015) Pitfalls in QCA's consistency measure. Journal of Comparative Politics 2:65-80.

Ragin, C. C. (2008) Redesigning Social Inquiry: Fuzzy Sets and Beyond, The Chicago University Press: Chicago and London.

Schneider, C. Q., Wagemann, C. (2012) Set-Theoretic Methods for the Social Sciences, Cambridge University Press: Cambridge.

Schneider, C. Q., Wagemann, C., Quaranta, M. (2012) How To... Use Software for Set-Theoretic Analysis. Online Appendix to "Set-Theoretic Methods for the Social Sciences". Available at www.cambridge.org/schneider-wagemann

Sarkar, D. (2008) Lattice: Multivariate Data Visualization with R, Springer: Berlin.

Examples

```
# Generate fake data to have perfect necessity
set.seed(123)
x <- runif(60, 0, 1)
y <- runif(60, 0, 1)

for(i in 1:length(y)) {
    while(x[i] < y[i]) {
        y[i] <- runif(1, 0, 1)
        x[i] <- runif(1, 0, 1)
```

```
    }
}

# Default with blue dots and pch = 1
xy.plot.lat(x, y, pch = 1, col = "blue")

# Parameters of fit in the corners with blue dots and pch = 1
xy.plot.lat(x, y, pch = 1, col = "blue", pos.fit = "corner")

# Load the Schneider data:

data(SCHF)

# Plot of condition EMP as necessary for outcome EXPORT with case labels
# and names for the plot and axes:

xy.plot.lat(SCHF$EMP, SCHF$EXPORT, necessity = TRUE, case.lab = TRUE,
            labs = rownames(SCHF), main = "EMP as necessary for EXPORT",
            ylab = "EXPORT", xlab = "EMP")
```

Index