

Package ‘SetMethods’

August 14, 2016

Type Package

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

Version 2.0

Date 2016-07-26

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 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 grew to include functions for performing set-theoretic multi-method research, QCA for clustered data, theory evaluation, and Enhanced Standard Analysis. Additionally it includes data to replicate the examples in the book and in the online appendix.

Depends QCA, QCAGUI, lattice, betareg, R (>= 3.3.0)

License GPL-2

NeedsCompilation no

Date/Publication 2016-08-14 15:26:57

Repository CRAN

R topics documented:

| | |
|--------------------|----|
| SetMethods-package | 2 |
| cases.nec.iir | 4 |
| cases.suf.dcn | 4 |
| cases.suf.dcv | 6 |
| cases.suf.iir | 7 |
| cases.suf.typ | 9 |
| cases.suf.typ.most | 11 |

| | |
|------------------------------------|----|
| cases.suf.typ.unique | 12 |
| cases.theory.evaluation | 14 |
| cluster.diagnostics | 15 |
| cluster.eqmcc | 16 |
| CronBerg | 17 |
| DTmv | 18 |
| emme | 19 |
| esa | 20 |
| FakeCS | 21 |
| FRSC | 22 |
| indirectCalibration | 23 |
| KA | 25 |
| LipsetCS | 26 |
| LipsetFS | 27 |
| LipsetRaw | 28 |
| matches.suf.dcvir | 29 |
| matches.suf.typdcn | 30 |
| matches.suf.typiir | 32 |
| matches.suf.tytyp | 34 |
| Pennings | 35 |
| pimdata | 36 |
| pimplot | 38 |
| print.clusterdiagnostics | 39 |
| print.clustereqmcc | 40 |
| QCAfit | 41 |
| Samford | 42 |
| Schneider | 43 |
| SchneiderLong | 44 |
| Selbst | 45 |
| Selbst.disappear | 46 |
| theory.evaluation | 47 |
| theory.fit | 48 |
| VisCS | 49 |
| VisFS | 50 |
| xy.plot | 51 |
| xy.plot.lat | 54 |

Index**57**

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, and Enhanced Standard Analysis. Additionally it includes data to replicate the examples in the book and in the online appendix.

Details

Package: SetMethods
Type: Package
Version: 2.0
Date: 2016-08-09
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. 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

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. COMPASS Working Paper 2013-76

cases.nec.iir

List individually irrelevant cases.

Description

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

Usage

```
cases.nec.iir(results, outcome, sol = 1)
```

Arguments

| | |
|---------|--|
| results | An object of class "qca". |
| outcome | A character string with the name of the outcome. |
| sol | A numeric vector where the first number indicates the number of the solution according to the order in the "qca" object. |

Author(s)

Juraj Medzihorsky

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

Rohlfing, I., Schneider C. Q. 2013. Combining QCA With Process Tracing in Analyses of Necessity. *Political Research Quarterly* 66(1): 220-35

See Also

[eqmcc](#)

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, intermed=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? |
| intermed | Logical. Should the intermediate solution 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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, intermed=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? |
| intermed | Logical. Should the intermediate solution 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[eqmcc](#)**Examples**

```

# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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, intermed=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? |
| intermed | Logical. Should the intermediate solution 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)
```



```

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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, intermed=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? |
| intermed | Logical. Should the intermediate solution 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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.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, intermed = 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? |
| intermed | Logical. Should the intermediate solution 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
```

```

conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
incl.cut1 = .9,
include = "?",
details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
incl.cut1 = .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, intermed = 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? |
| intermed | Logical. Should the intermediate solution 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

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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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_eval)
```

Arguments

`theory_eval` A data frame with the intersections between theory and empirics. Usually the output of the `theory.evaluation` function should be inputted.

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

[eqmcc](#)

Examples

```

# Import your data. For example:

data(Schneider)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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.evaluation(theory = t, empirics = sol_yi, outcome = "EXPORT",
                       intermed = TRUE, sol = 2)

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

cases.theory.evaluation(INT)

```

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. COMPASS Working Paper 2013-76

See Also

[eqmcc](#)

Examples

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

data(SchneiderLong)

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

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

cluster.eqmcc

Diagnostic tool for clustered data

Description

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

Usage

```
cluster.eqmcc(results, data, outcome, unit_id, cluster_id, intermed=FALSE, sol = 1)
```

Arguments

| | |
|------------|--|
| results | An object of class "qca". |
| data | A data frame. |
| outcome | A character string with the name of the outcome. |
| 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). |

intermed Logical. Use the intermediate solution?
 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

[eqmcc](#)

Examples

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

data(SchneiderLong)

# Get the intermediate solution:

sol_yi <- eqmcc(SchneiderLong, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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.eqmcc(results = sol_yi, data = SchneiderLong, outcome = "EXPORT", unit_id = "COUNTRY",
              cluster_id = "YEAR", intermed = TRUE, sol = 2)
```

CronBerg

Berg-Schlosser and Cronqvist (2005)

Description

The CronBerg data frame has 18 rows and 5 variables

Usage

```
data(CronBerg)
```

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 demonstrate 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(CronBerg)
```

DTmv

Fake data for mvQCA

Description

mvQCA data frame has 25 rows and 4 sets.

Usage

```
data(DTmv)
```

Format

- A 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 purposes 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(DTmv)
```

emme

Emmenegger (2011)

Description

The emme data frame has 19 rows and 8 sets

Usage

```
data(emme)
```

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-security 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(emme)
```

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

| | |
|---------------------------|---|
| <code>oldtt</code> | A truthTable object. |
| <code>nec_cond</code> | 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 "+", while intersections are performed with a "*". |
| <code>imposs_LR</code> | A vector of character strings containing the impossible logical remainders. 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 "*". |
| <code>contrad_rows</code> | A vector containing the names of the rows that are contradictory. |

Value

It returns a new truth table in which all logical remainder 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

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

TT_y <- truthTable(Schneider, outcome = "EXPORT",
                  conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                  incl.cut1 = .9,
                  complete = TRUE,
                  PRI = TRUE,
                  sort.by = c("out", "incl", "n"))

# Exclude condition STOCK as necessary for EXPORT
# Exclude the rows "13", "14", "46", "51" as contradictory:

newtt <- esa(oldtt = TT_y, nec_cond = "STOCK", contrad_rows = c("13", "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 purposes 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)
```

FRSC

Freitag and Schlicht (2009)

Description

The FRSC data frame has 16 rows and 8 sets

Usage

```
data(FRSC)
```

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(FRSC)
```

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 transforming 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")
```

KA *Koenig-Archibugi (2004)*

Description

The KA data frame has 13 rows and 5 sets

Usage

```
data(KA)
```

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 institutional change in EU foreign and security policy. Data are fuzzy-sets.

References

Koenig-Archibugi, M. (2004) "Explaining Government Preferences for Institutional 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(KA)
```

LipsetCS

Lipset (1959), crisp-set

Description

The LipsetCS data frame has 18 rows and 6 sets

Usage

```
data(LipsetCS)
```

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(LipsetCS)
```

LipsetFS

Lipset (1959), fuzzy-set

Description

The LipsetFS data frame has 18 rows and 6 sets

Usage

```
data(LipsetFS)
```

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(LipsetFS)
```

LipsetRaw

Lipset (1959), raw data

Description

The LipsetRaw data frame has 18 rows and 6 variables

Usage

```
data(LipsetRaw)
```

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(LipsetRaw)
```

| | |
|---------------------|--|
| matches.suf.dcviiir | <i>Match deviant coverage cases and individually irrelevant cases with regards to sufficiency.</i> |
|---------------------|--|

Description

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

Usage

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

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? |
| intermed | Logical. Should the intermediate solution 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

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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
```

```

      incl.cut1 = .9,
      include = "?",
      details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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.dcviiir(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.dcviiir(results = sol_yp, outcome = "EXPORT", max_pairs=3)

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

matches.suf.dcviiir(results = sol_yi, outcome = "EXPORT", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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.dcviiir(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, intermed=FALSE, sol=1, max_pairs = 5)
```

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? |
| intermed | Logical. Should the intermediate solution 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

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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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.tydcn(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.tydcn(results = sol_yp, outcome = "EXPORT", max_pairs=3)

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

matches.suf.tydcn(results = sol_yi, outcome = "EXPORT", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .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.tydcn(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,
                  intermed=FALSE, sol=1, max_pairs = 5)

```

Arguments

| | |
|-----------|---|
| results | An object of class "qca". |
| outcome | A character string with the name of the outcome. |
| 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? |
| intermed | Logical. Should the intermediate solution 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[eqmcc](#)**Examples**

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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 <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
                conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
                incl.cut1 = .9,
                include = "?",
                details = TRUE, show.cases = TRUE)

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

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

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

Arguments

| | |
|-----------|---|
| results | An object of class "qca". |
| outcome | A character string with the name of the outcome. |
| 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? |
| intermed | Logical. Should the intermediate solution 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
```

```

conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
incl.cut1 = .9,
include = "?",
details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
incl.cut1 = .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", intermed = TRUE)

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

sol_nyp <- eqmcc(Schneider, outcome = "EXPORT", neg.out = TRUE,
conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
incl.cut1 = .9,
include = "?",
details = TRUE, show.cases = TRUE)

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

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

```

Pennings

Pennings (2003)

Description

The Pennings data frame has 45 rows and 5 sets

Usage

```
data(Pennings)
```

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 constitution.

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(Pennings)
```

| | |
|---------|--|
| pimdata | <i>Function to extract prime implicants table from object of class "qca"</i> |
|---------|--|

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, intermed = FALSE, sol = 1)
```

Arguments

| | |
|----------|---|
| results | An object of class "qca". |
| outcome | A character string with the name of the outcome in capital letters. |
| intermed | Logical. Use the intermediate solution? |
| 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. |

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

[eqmcc pimplot](#)

Examples

```
# Import your data. For example:
data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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", intermed = TRUE, sol = 1)
```

pimplot

Prime implicants 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".

Usage

```
pimplot(results, outcome, intermed=FALSE, sol = 1, case_labels=TRUE,  
lab_color=rgb(0,0,0,0.5), lab_jitter=FALSE)
```

Arguments

| | |
|-------------|---|
| results | An object of class "qca". |
| outcome | A character string with the name of the outcome in capital letters. |
| intermed | Logical. Use the intermediate solution? |
| 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. |
| case_labels | Logical. Print case labels? |
| lab_color | Color of the labels. |
| lab_jitter | Logical. Add jitter to label positions? |

Value

A plot.

Author(s)

Juraj Medzihorsky

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

[eqmcc pimdata](#)

Examples

```

# Import your data. For example:

data(Schneider)

# Get the parsimonious solution:

sol_yp <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .9,
               include = "?",
               details = TRUE, show.cases = TRUE)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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(results = sol_yp, outcome = "EXPORT")

# Plot the prime implicants of the first intermediate solution:

pimplot(results = sol_yi, outcome = "EXPORT", intermed=TRUE, sol = 1)

```

```
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

[eqmcc](#)

`print.clustereqmcc` *Print a clustereqmcc Object*

Description

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

Usage

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

Arguments

`x` an object inheriting from class `cluster.eqmcc`, 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

[eqmcc](#)

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 and PRODUCT. It works with both single and multiple conditions.

Usage

```
QCAfit(x, y, cond.lab = NULL, necessity = FALSE, negation = FALSE)
```

Arguments

| | |
|-----------|--|
| x | is a vector containing the values of a condition or a matrix with more than one conditions. |
| y | is a vector containing the values of the output. |
| 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, TRUE, the function returns a table of parameters of fit for sufficient condition(s) (Consistency, Coverage, PRI and Product). When it set to FALSE the function returns a table of parameters of fit for necessary condition(s) (Consistency, Coverage, Relevance of Necessity). |
| negation | 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. |

Value

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

Author(s)

Mario Quaranta

References

- 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

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"), negation = TRUE)
```

Samford

Samford (2010)

Description

The Samford data frame has 61 rows and 4 sets

Usage

```
data(Samford)
```

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(Samford)
```

| | |
|-----------|---------------------------------|
| Schneider | <i>Schneider et. al. (2010)</i> |
|-----------|---------------------------------|

Description

The Schneider data frame has 76 observations and 7 sets

Usage

```
data(Schneider)
```

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(Schneider)
```

| | |
|---------------|---------------------------------|
| SchneiderLong | <i>Schneider et. al. (2010)</i> |
|---------------|---------------------------------|

Description

The SchneiderLong data frame has 76 observations and 9 variables.

Usage

```
data(SchneiderLong)
```

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(SchneiderLong)
```

| | |
|--------|--|
| Selbst | <i>Selbst, practicing the truth table algorithm data</i> |
|--------|--|

Description

The Selbst data frame has 130 rows and 4 sets

Usage

```
data(Selbst)
```

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(Selbst)
```

| | |
|------------------|--|
| Selbst.disappear | <i>Selbst, disappearing necessary condition data</i> |
|------------------|--|

Description

The Selbst.disappear data frame has 98 rows and 4 sets

Usage

```
data(Selbst.disappear)
```

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(Selbst.disappear)
```

| | |
|-------------------|--|
| theory.evaluation | <i>Membership of cases in the intersections between theory and the empirical solution.</i> |
|-------------------|--|

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.evaluation(theory, empirics, outcome, intermed=FALSE, 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'. |
| outcome | A character string with the name of the outcome. |
| intermed | Logical. Use the intermediate solution? |
| 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. |

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

[eqmcc](#)

Examples

```

# Import your data. For example:

data(Schneider)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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.evaluation(theory = t, empirics = sol_yi, outcome = "EXPORT", intermed = TRUE, sol = 2)

```

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_eval)
```

Arguments

theory_eval A data frame with the intersections between theory and empirics. Usually the output of the theory.evaluation 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

[eqmcc](#)

Examples

```
# Import your data. For example:

data(Schneider)

# Get the intermediate solution:

sol_yi <- eqmcc(Schneider, outcome = "EXPORT",
               conditions = c("EMP", "BARGAIN", "UNI", "OCCUP", "STOCK", "MA"),
               incl.cut1 = .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.evaluation(theory = t, empirics = sol_yi, outcome = "EXPORT",
                        intermed = TRUE, sol = 2)

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

theory.fit(INT)
```

VisCS

Vis (2009), crisp set data

Description

The VisCS data frame has 25 rows and 4 sets

Usage

```
data(VisCS)
```

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(VisCS)
```

VisFS

Vis (2009), fuzzy set data

Description

The VisFS data frame has 25 rows and 4 sets

Usage

```
data(VisFS)
```

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(VisFS)
```

```
xy.plot
```

Function producing enhanced XY plots

Description

xy.plot produces XY plots and provides values for consistency, coverage, RoN, and 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 $c(\text{bottom}, \text{left}, \text{top}, \text{right})$ 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 $\text{mgp}[1]$ affects title whereas $\text{mgp}[2:3]$ affect axis. The default is $c(2.2, 0.8, 0)$. See ?par. |
| cex.fit | character expansion for the parameters of fit. The default is 0.6. See ?pch or ?text. |
| cex.axis | character expansion for the x-axis and y-axis. The default is 0.7. See ?pch or ?text. |
| cex.main | character expansion for the overall title of the plot. The default 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 outside the plotting area just below the main title, or "corner", which places the parameters of fit in the corner of the plotting area. The default is $\text{pos.fit} = \text{"top"}$. |
| 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.

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 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)
```

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, 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 default 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 area. The default is pos.fit = "top". |
| strip.cex | character expansion for the parameters of fit when pos.fit = "top". The default 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

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
- 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")
```


Index

*Topic **QCAGUI**

- cases.nec.iir, 4
- cases.suf.dcn, 4
- cases.suf.dcv, 6
- cases.suf.iir, 7
- cases.suf.typ, 9
- cases.suf.typ.most, 11
- cases.suf.typ.unique, 12
- cases.theory.evaluation, 14
- cluster.diagnostics, 15
- cluster.eqmcc, 16
- esa, 20
- matches.suf.dcvir, 29
- matches.suf.typdcn, 30
- matches.suf.typiir, 32
- matches.suf.typtyp, 34
- pimdata, 36
- pimplot, 38
- print.clusterdiagnostics, 39
- print.clustereqmcc, 40
- theory.evaluation, 47
- theory.fit, 48

*Topic **datasets**

- CronBerg, 17
- DTmv, 18
- emme, 19
- FakeCS, 21
- FRSC, 22
- KA, 25
- LipsetCS, 26
- LipsetFS, 27
- LipsetRaw, 28
- Pennings, 35
- Samford, 42
- Schneider, 43
- SchneiderLong, 44
- Selbst, 45
- Selbst.disappear, 46
- VisCS, 49

VisFS, 50

- cases.nec.iir, 4
- cases.suf.dcn, 4
- cases.suf.dcv, 6
- cases.suf.iir, 7
- cases.suf.typ, 9
- cases.suf.typ.most, 11
- cases.suf.typ.unique, 12
- cases.theory.evaluation, 14
- cluster.diagnostics, 15
- cluster.eqmcc, 16
- CronBerg, 17

- DTmv, 18

- emme, 19
- eqmcc, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 21, 29, 31, 33, 34, 37, 38, 40, 47, 49
- esa, 20

- FakeCS, 21
- FRSC, 22

- indirectCalibration, 23

- KA, 25

- LipsetCS, 26
- LipsetFS, 27
- LipsetRaw, 28

- matches.suf.dcvir, 29
- matches.suf.typdcn, 30
- matches.suf.typiir, 32
- matches.suf.typtyp, 34

- Pennings, 35
- pimdata, 36, 38
- pimplot, 37, 38
- print.clusterdiagnostics, 39

`print.clustereqmcc`, [40](#)

`QCAfit`, [41](#)

`Samford`, [42](#)

`Schneider`, [43](#)

`SchneiderLong`, [44](#)

`Selbst`, [45](#)

`Selbst.disappear`, [46](#)

`SetMethods (SetMethods-package)`, [2](#)

`SetMethods-package`, [2](#)

`theory.evaluation`, [47](#)

`theory.fit`, [48](#)

`VisCS`, [49](#)

`VisFS`, [50](#)

`xy.plot`, [51](#)

`xy.plot.lat`, [54](#)