QCA Sufficiency Inclusion Score & Minimum Frequency Threshold Simulation with Random Variables

Description

Returns QCA results for a range of minimum frequency thresholds across an arbitrarily large set of sufficiency inclusion scores when a random variable is repeatedly added to the dataset

Usage

```
QCA.random(n, type, data, outcome, conditions=NULL,
    min.incl.cut, max.incl.cut, min.n.cut,
    max.n.cut, reps, plot, plot.legend, ...)
```

Arguments

`n`	number of times to simulate random variable inclusion
`type`	type of random variabe to be included; either "binary", which samples from 0 and 1, or "uniform", which draws from a uniform distribution bound by 0 and 1
`data`	an object of class 'data.frame'
`outcome`	a character string or column index indicating the outcome variable
`conditions`	optional character vector or vector of column indices indicating explanatory variables
`min.incl.cut`	numeric lower bound for sampling of sufficiency inclusion scores
`max.incl.cut`	numeric upper bound for sampling of sufficiency inclusion scores
`min.n.cut`	numeric lower bound for minimum frequency thresholds
`max.n.cut`	numeric upper bound for minimum frequency thresholds
`reps`	number of sufficiency inclusion score pairs to be sampled
`plot`	if TRUE, plot solutions
`plot.legend`	"solutions" indicates plot legend should contain actual unique solutions; "ids" indicates plot should contain numeric identifiers for unique solutions; "none" indicates plot should not contain a legend
`...`	optional arguments passed to `eqmcc()`

Value

`plot`	plot of QCA results for given sufficiency inclusion score pairs and minimum frequency thresholds
`results`	data frame containing combinations of sufficiency inclusion scores and QCA solutions

`legend` list containing unique solutions (config) and their numeric identifiers (config.id)

Author(s)

Chris Krogslund, http://ckro.gs; ckrogslund@berkeley.edu

Examples

```
protest.data<-read.csv(file="http://philhoward.org/wp-content/
    uploads/2012/11/International-Studies-Review-Replication-Data.csv")
protest.data<-protest.data[,!colnames(protest.data)

QCA.random(n=3, type="uniform", data=protest.data, outcome="success",
    min.incl.cut=0, max.incl.cut=1, min.n.cut=1,
    max.n.cut=4, reps=100, plot=TRUE, plot.legend="ids")

QCA.random(n=5, type="uniform", data=protest.data, outcome="success",
    conditions=c("mobile", "fuel"), min.incl.cut=0,
    max.incl.cut=1, min.n.cut=1, max.n.cut=4, reps=100,
    plot=TRUE, plot.legend="ids")
```