

# QCA Sufficiency Inclusion Score & Minimum Frequency Threshold Simulation

## Description

Returns QCA results for a range of minimum frequency thresholds across an arbitrarily large set of sufficiency inclusion scores

## Usage

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

## Arguments

<code>data</code>	an object of class 'data.frame'
<code>outcome</code>	a character string or column index indicating the outcome variable
<code>conditions</code>	optional character vector or vector of column indices indicating explanatory variables
<code>min.incl.cut</code>	numeric lower bound for sampling of sufficiency inclusion scores
<code>max.incl.cut</code>	numeric upper bound for sampling of sufficiency inclusion scores
<code>min.n.cut</code>	numeric lower bound for minimum frequency thresholds
<code>max.n.cut</code>	numeric upper bound for minimum frequency thresholds
<code>reps</code>	number of sufficiency inclusion score pairs to be sampled
<code>plot</code>	if TRUE, plot solutions
<code>plot.legend</code>	"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
<code>...</code>	optional arguments passed to <code>eqmcc ( )</code>

## Value

<code>plot</code>	plot of QCA results for given sufficiency inclusion score pairs and minimum frequency thresholds
<code>results</code>	data frame containing combinations of sufficiency inclusion scores and QCA solutions
<code>legend</code>	list containing unique solutions (config) and their numeric identifiers (config.id)

## Author(s)

Chris Krogslund, <http://ckro.gs>; [ckrogslund@berkeley.edu](mailto: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.sim(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.sim(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")
```