Package ‘braQCA’

November 28, 2024

Title Bootstrapped Robustness Assessment for Qualitative Comparative Analysis

Version 1.4.11.27

Description Test the robustness of a user's Qualitative Comparative Analysis solutions to randomness, using the bootstrapped assessment: baQCA(). This package also includes a function that provides recommendations for improving solutions to reach typical significance levels: brQCA(). Data included come from McVeigh et al. (2014) <doi:10.1177/0003122414534065>.

Depends R (>= 3.2.3)

Encoding UTF-8

Imports QCA, bootstrap

License GPL-3

LazyData true

RoxygenNote 7.2.2

NeedsCompilation no

Author C. Ben Gibson [aut] (<https://orcid.org/0000-0003-2561-9574>), Burrel Vann Jr [aut, cre] (<https://orcid.org/0000-0003-3066-5815>)

Maintainer Burrel Vann Jr <bvannjr@sdsu.edu>

Repository CRAN

Date/Publication 2024-11-27 23:20:06 UTC

Contents

baQCA *Boostrapped Assessment*

Description

This function performs the the Bootstrapped Assessment for QCA (baQCA) on a given QCA model object.

Usage

```
baQCA(
  mod,
  sim = 2000,
  all = TRUE,
  include = c(""),
  row.dom = FALSE,
  omit = c(),
  dir.exp = c()
)
```

Arguments

mod	name of the QCA model object – the minimization of the truth table.
sim	the number of simulations the baQCA function should run. Default set to sim=2000.
all	logical, whether or not causal conditions AND outcome should be resampled (with replacement). Default set to all=TRUE.
include	[from QCA package] "A vector of additional output function values to be included in the minimization." Default set to include=c("").
row.dom	[from QCA package] "Logical, impose row dominance as constraint on solution to eliminate dominated inessential prime implicants." Default set to FALSE.
omit	[from QCA package] "A vector of configuration index values or matrix of configurations to be omitted from minimization." Default set to omit=c().
dir.exp	[from QCA package] "A vector of directional expectations for deriving intermediate solutions." Default set to dir.exp=c().

Value

This function returns a value which is the probability of a random QCA result (e.g. a result from random data) given the parameters set by the researcher in the model (configurational n threshold, consistency score threshold, etc), and a confidence interval around this value. This value is interpreted similarly to a p-value."

Examples

```
qca.data <- rallies[,8:13]
rownames(qca.data)<-rownames(rallies)
truth<-QCA::truthTable(qca.data,outcome="P",sort.by="incl",incl.cut1=0.85,n.cut=1,show.cases=TRUE)
mod1 <- QCA::minimize(truth,details=TRUE,show.cases=TRUE)

summary(baQCA(mod1,sim=1))
```

brQCA *Bootstrapped Recommendation*

Description

Provides recommendations for consistency score and configurational n thresholds to attain a desired level of confidence in a QCA algorithm application.

Usage

```
brQCA(
  qca.data,
  outcome = "OUT",
  type = "crisp",
  inclcut = "",
  ncut = 2,
  neg.out = FALSE,
  sim = 10,
  verbose = TRUE
)
```

Arguments

`qca.data`	the QCA data frame.
`outcome`	the outcome variable in the QCA data frame of causal conditions; `"OUT"` is the outcome variable for an application of QCA.
`type`	of QCA application, `"crisp"` or `"fuzzy"` sets. Default set to `type = "crisp"`.
`inclcut`	range of consistency scores for inclusion. If not specified, this defaults to `seq(from = 0.5, to = 1, by = 0.01)`.
`ncut`	configurational n levels to simulate. Can be altered to give options for the range of minimum to maximum `ncut` value that the truth table yields, by naming the the truth table object (e.g. `truth`) and calling the minimum and maximum number of cases, using `ncut=min(truth$tt$n):max(truth$tt$n)` identified by the truth table. Default set to `ncut=2`.
`neg.out`	[from QCA package] “Logical, use negation of outcome (ignored if data is a truth table object).” Default set to `neg.out=FALSE`.

`sim`	number of simulations to run for each combination of parameters. The final number of simulations is `length(inclcut)*length(ncut)*sim*2`. Default set to `sim=10`.
`verbose`	prints the system time used to run the simulation and the percent complete. Default set to `verbose=TRUE`.

Value

Significance levels reached (.10,.05, .01, .001) when specifying a combination of inclcut, ncut, and neg.out in a QCA model.

Examples

```
qca.data <- rallies[,8:13]

## Not run:
brQCA(qca.data,outcome="P",ncut=5,sim=1)

## End(Not run)
```

`conf.table` *Configuration Table*

Description

Internal function; calculates via logistic regression the output of the Bootstrapped Robustness Recommendation

Usage

```
conf.table(data, ncut = ncut)
```

Arguments

`data`	name of the model object; the table of solutions for an application of QCA. Default set to `data`.
`ncut`	configurational n levels for inclusion. Default set to `ncut=4`

Value

The output of the Bootstrapped Recommendation #' @export

`rallies` *McVeigh et al. (2014) Tea Party Data*

Description

This data set is an abbreviated version of the data set used by McVeigh et al. (2014). These data cover all 67 counties in Florida, and come from the American Community Survey (2005-2009).

Usage

```
rallies
```

Format

A data frame with 67 observations and 13 variables.

tprallies	number of Tea Party rallies in county, 2009-2010
reppct2008	percent of county vote for the Republican Presidential candidate (John McCain) in 2008
dempct2008	percent of county vote for the Democratic Presidential candidate (Barack Obama) in 2008
pctBA25	percent of county, aged 25 or older, with a bachelor's degree
pctunemp	percent of county that is unemployed
pctevang	percent of county that belongs to an Evangelical denomination
pctblack	percent of county that identifies as Black
P	binary. `0` if county had no Tea Party rallies, `1` if county had *at least* on Tea Party rally
R	binary. `0` if the majority of votes in the county were for the Democratic Presidential candidate (Barack Obama)
C	binary. `0` if percent of county with a bachelor's degree was below-average for Florida, `1` if percent of county wi
U	binary. `0` if percent unemployed in county was below-average for Florida, `1` if percent unemployed in county w
E	binary. `0` if percent Evangelical in county was below-average for Florida, `1` if percent Evangelical in county wa
B	binary. `0` if percent Black in county was below-average for Florida, `1` if percent Black in county was at or abov

`sim.brQCA` *Simulation Application*

Description

Internal function to calculate the Bootstrapped Recommendation.

Usage

```
sim.brQCA(
  qca.data,
  outcome = "OUT",
  conditions = c(""),
  sim = 10,
  ncut = 2,
  type = "crisp",
  inclcut = "",
  neg.out = FALSE,
  verbose = TRUE
)
```

Arguments

`qca.data` the data frame of causal conditions.

`outcome` the outcome variable (object name) in the QCA data frame of causal conditions; `"OUT"` is the outcome variable for an application of QCA. Default set to `outcome="OUT"`.

`conditions` a set of causal conditions. Default set to `conditions=c("")`

`sim` number of simulations to run. Default set to `sim=10`.

`ncut` configurational n levels for inclusion. Default set to `ncut=2`.

`type` type of QCA application, `"crisp"` or `"fuzzy"` sets. Default set to `type = "crisp"`.

`inclcut` minimum sufficiency score for inclusion. Default set to `inclcut=""`.

`neg.out` [from QCA package] “Logical, use negation of outcome (ignored if data is a truth table object).” Default set to `neg.out=FALSE`.

`verbose` prints the system time used to run the simulation and the percent complete. Default set to `verbose=TRUE`.

Value

Simulation information later passed on to conf.table.

`summary.baQCAtest` *Summarize Results of baQCA*

Description

Displays results of baQCA.

Usage

```
## S3 method for class 'baQCAtest'
summary(object, ...)
```

Arguments

`object`	Object returned by `baQCA`.
`...`	Additional parameters to pass on.

Value

Matrix of values for percent of simulations returning result from random data, along with confidence interval.

Examples

```
qca.data <- rallies[,8:13]
rownames(qca.data)<-rownames(rallies)
truth<-QCA::truthTable(qca.data,outcome="P",sort.by="incl",incl.cut1=0.85,n.cut=1,show.cases=TRUE)
mod1 <- QCA::minimize(truth,details=TRUE,show.cases=TRUE)

test <- baQCA(mod1,sim=1)
summary(test)
```

Index