Package 'QCAcluster'

October 26, 2021

Type Package

Title Tools for the Analysis of Clustered Data in QCA

Version 0.1.0

Depends R (>= 2.10)

Description Clustered set-relational data in Qualitative Comparative Analysis (QCA) can have a hierarchical structure, a panel structure or repeated cross sections. 'QCAcluster' allows QCA researchers to supplement the analysis of pooled the data with a disaggregated perspective focusing on selected partitions of the data. The pooled data can be partitioned along the dimensions of the clustered data (individual cross sections or time series) to perform partition-specific truth table minimizations. Empirical researchers can further calculate the weight that each partition has on the parameters of the pooled solution and the diversity of the cases under analysis within and across partitions (see <https://ingorohlfing.github.io/QCAcluster/>).

License GPL-3

Encoding UTF-8

LazyData true

Imports data.table (>= 1.12.8), plyr (>= 1.8.5), QCA (>= 3.7), testit (>= 0.11), purrr (>= 0.3.3), UpSetR (>= 1.4.0), magrittr, stringi (>= 1.7.4), rlist(>= 0.4.6.1)

RoxygenNote 7.1.2

URL https://github.com/ingorohlfing/QCAcluster

BugReports https://github.com/ingorohlfing/QCAcluster/issues

Suggests rmarkdown, knitr

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Ingo Rohlfing [aut, cre] (0000-0001-8715-4771), Ayjeren Bekmuratovna [aut], Jan Schwalbach [aut] (0000-0002-6990-8098)

Maintainer Ingo Rohlfing <i.rohlfing@uni-koeln.de>

Repository CRAN

Date/Publication 2021-10-26 15:00:01 UTC

R topics documented:

`Grauvogel2014` *Original data used by Grauvogel/von Soest (2014)*

Description

A dataset containing the calibrated set values for the article: Grauvogel, Julia and Christian von Soest (2014): Claims to Legitimacy Count: Why Sanctions Fail to Instigate Democratisation in Authoritarian Regimes. European Journal of Political Research 53 (4): 635-653.

Usage

```
Grauvogel2014
```

Format

A data frame with 120 rows and 10 variables:

Code Sender-target ID

Sender Country or institution imposing sanctions

Target Country that is target of sanctions

Timeframe Considered years for each country case

Persistence Degree of regime persistence after the intervention

Comprehensiveness Scope of the imposed sanctions - comprehensive vs. targeted sanctions

Linkage Economic and social, respectively communicative and geographic ties

Vulnerability Military and economic vulnerability of the state to outside pressure

Repression Degree of repression by the state

Claims Variety and strength of claims to legitimacy

Source

Grauvogel (2014) <doi:10.1111/1475-6765.12065>

`partition_div` *Diversity of cases belonging to the same partition of the pooled data*

Description

`partition_div` calculates the diversity of cases that belong to the same partition of the clustered data (a time series; a cross section; etc.). Diversity is measured by the number of truth table rows that the cases of a partition cover. `partition_div` calculates the partition diversity for all truth table rows and for the subsets of consistent and inconsistent rows.

Usage

```
partition_div(dataset, units, time, cond, out, n_cut, incl_cut)
```

Arguments

`dataset`	Calibrated pooled dataset that is partitioned and minimized for deriving the pooled solution.
`units`	Units defining the within-dimension of data (time series)
`time`	Periods defining the between-dimension of data (cross sections)
`cond`	Conditions used for the pooled analysis
`out`	Outcome used for the pooled analysis
`n_cut`	Frequency cut-off for designating truth table rows as observed in the pooled data
`incl_cut`	Inclusion cut-off for designating truth table rows as consistent in the pooled data

Value

A dataframe presenting the diversity of cases belonging to the same partition with the following columns:

- `type`: The type of the partition. `pooled` are rows with information on the pooled data; `between` is for cross-section partitions; `within` is for time-series partitions.
- `partition`: Specific dimension of the partition at hand. For between-dimension, the unit identifiers are included here (argument `units`). For the within-dimension, the time identifier are listed (argument `time`). The entry is `-` for the pooled data without partitions.
- `diversity`: Count of all truth table rows with at least one member belonging to a partition.
- `diversity_1`: Count of consistent truth table rows with at least one member belonging to a partition.
- `diversity_0`: Count of inconsistent truth table rows with at least one member belonging to a partition.

- diversity_per: Ratio of the value for diversity and the total number of truth table rows from pooled data (diversity value for pooled data).
- diversity_per_1: Ratio of the value for diversity_1 and the total number of consistent truth table rows from pooled data (diversity_1 value for pooled data).
- diversity_per_0: Ratio of the value for diversity_0 and the total number of inconsistent truth table rows from pooled data (diversity_0 value for pooled data).

Examples

```
data(Schwarz2016)
Schwarz_diversity <- partition_div(Schwarz2016,
units = "country", time = "year",
cond = c("poltrans", "ecotrans", "reform", "conflict", "attention"),
out = "enlarge", 1, 0.8)
```

---

partition_min *Generation of conservative or parsimonious solution for individual partitions*

Description

partition_min decomposes clustered data into individual partitions. For panel data, for example, these can be cross sections, time series or both. The function derives an individual solution for each partition and the pooled data to assess the robustness of the solutions in a comparative perspective.

Usage

```
partition_min(
  dataset,
  units,
  time,
  cond,
  out,
  n_cut,
  incl_cut,
  solution,
  BE_cons,
  WI_cons,
  BE_ncut,
  WI_ncut
)
```

Arguments

`dataset`	Calibrated pooled dataset that is partitioned and minimized for deriving the pooled solution.
`units`	Units defining the within-dimension of data (time series). If no units are specified, the data is assumed to lack a dimension and be hierarchical.
`time`	Periods defining the between-dimension of data (cross sections). This should be specified because it does not make sense to partition a time series into individual data points.
`cond`	Conditions used for minimization
`out`	Outcome used for minimization
`n_cut`	Frequency cut-off for designating truth table rows as observed as opposed to designating them as remainders for the *pooled* data.
`incl_cut`	Inclusion (a.k.a. consistency) cut-off for designating truth table rows as consistent for the *pooled* data.
`solution`	A character specifying the type of solution that should be derived. `C` produces the conservative (or complex) solution, `P` for the parsimonious solution. See `partition_min_inter` for a separate function for the intermediate solution.
`BE_cons`	Inclusion thresholds for creating an individual truth table for each cross section. They must be specified as a numeric vector. Its length should be equal the number of cross sections. The order of thresholds corresponds to the order of the cross sections in the data defined by the cross-section ID in the dataset (such as years in ascending order).
`WI_cons`	Inclusion thresholds for creating an individual truth table for each time series. They must be specified as a numeric vector. Its length should be equal the number of time series. The order of thresholds corresponds to the order of the of the time-series (unit) ID in the dataset (such as countries in alphabetical order).
`BE_ncut`	For *cross sections*, the minimum number of members needed for declaring a truth table row as relevant as opposed to designating it as a remainder. Must be specified as a numeric vector. Its length should be equal the number of cross sections. The order of thresholds corresponds to the order of the cross sections in the data defined by the cross-section ID in the dataset (such as years in ascending order).
`WI_ncut`	For *time series*, the minimum number of members needed for declaring a truth table row as relevant as opposed to designating it as a remainder. Must be specified as a numeric vector. Its length should be equal the number of time series. The order of thresholds corresponds to the order of the of the time-series (unit) ID in the dataset (such as countries in alphabetical order).

Value

A dataframe summarizing the partition-specific and pooled solutions with the following columns:

- `type`: The type of the partition. `pooled` are rows with information on the pooled data; `between` is for cross-section partitions; `within` is for time-series partitions.

- partition: Specific dimension of the partition at hand. For between-dimension, the unit identifiers are included here (argument units). For the within-dimension, the time identifier are listed (argument time). The entry is - for the pooled data without partitions.
- solution: The solution derived for the partition or the pooled data. Absence of a condition is denoted by the ~ sign.
- model: Running ID for models. In the presence of model ambiguity, each model has its own row with its individual solution and parameters. The rest of the information in the row is duplicated, for example by having two rows for the within-partition 1996. The column model highlights the presence of model ambiguity by numbering all models belonging to the same solution. For example, if three consecutive rows are numbered 1, 2 and 3, then these rows belong to the same solution and represent model ambiguity. If a 1 in a row is followed by another 1, then there is no model ambiguity.
- consistency: The consistency score (a.k.a. inclusion score) for the partition of the data or the pooled data.
- coverage: The coverage score for the partition of the data or the pooled data.

Examples

```
# loading data from Thiem (EPSR, 2011; see data documentation)
data(Thiem2011)

# running function for parsimonious solution
Thiem_pars <- partition_min(
  dataset = Thiem2011,
  units = "country", time = "year",
  cond = c("fedismfs", "homogtyfs", "powdifffs", "comptvnsfs", "pubsupfs", "ecodpcefs"),
  out = "memberfs",
  n_cut = 1, incl_cut = 0.8,
  solution = "P",
  BE_cons = c(0.9, 0.8, 0.7, 0.8, 0.6, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8),
  WI_cons = c(0.5, 0.8, 0.7, 0.8, 0.6, rep(0.8, 10)))
```

---

partition_min_inter *Generation of intermediate solutions for individual partitions of clustered set-relational data*

Description

partition_min_inter decomposes clustered data into individual partitions such as cross-sections and time-series for panel data. It derives an individual intermediate solution for each partition and the pooled data to assess the robustness of the solutions.

Usage

```
partition_min_inter(
  dataset,
  units,
  time,
  cond,
  out,
  n_cut,
  incl_cut,
  intermediate,
  BE_cons,
  WI_cons,
  BE_ncut,
  WI_ncut
)
```

Arguments

dataset	Calibrated pooled dataset for partitioning and minimization
units	Units defining the within-dimension of data (time series)
time	Periods defining the between-dimension of data (cross sections)
cond	Conditions used for the pooled analysis
out	Outcome used for the pooled analysis
n_cut	Frequency cut-off for designating truth table rows as observed
incl_cut	Inclusion cut-off for designating truth table rows as consistent
intermediate	A vector of directional expectations to derive intermediate solutions
BE_cons	Inclusion (or consistency) thresholds for cross sections. Must be specified as a numeric vector with length equaling the number of cross sections. Numbers correspond to the order of the cross section ID in the data (such as years in ascending order).
WI_cons	Inclusion (or consistency) thresholds for time series. Must be specified as a numeric vector with length equaling the number of time series. Numbers correspond to the order of the time series (unit) ID in the data (such as countries in alphabetical order).
BE_ncut	For *cross sections*, the minimum number of members needed for declaring a truth table row as relevant as opposed to designating it as a remainder. Must be specified as a numeric vector. Its length should be equal the number of cross sections. The order of thresholds corresponds to the order of the cross sections in the data defined by the cross-section ID in the dataset (such as years in ascending order).
WI_ncut	For *time series*, the minimum number of members needed for declaring a truth table row as relevant as opposed to designating it as a remainder. Must be specified as a numeric vector. Its length should be equal the number of time series. The order of thresholds corresponds to the order of the of the time-series (unit) ID in the dataset (such as countries in alphabetical order).

Value

A dataframe summarizing the partition-specific and pooled solutions with the following columns:

- `type`: The type of the partition. `pooled` are rows with information on the pooled data; `between` is for cross-section partitions; `within` is for time-series partitions.
- `partition`: Specific dimension of the partition at hand. For between-dimension, the unit identifiers are included here (argument `units`). For the within-dimension, the time identifier are listed (argument `time`). The entry is `-` for the pooled data without partitions.
- `solution`: The solution derived for the partition or the pooled data. Absence of a condition is denoted by the ~ sign.
- `model`: Running ID for models. In the presence of model ambiguity, each model has its own row with its individual solution and parameters. The rest of the information in the row is duplicated, for example by having two rows for the within-partition 1996. The column `model` highlights the presence of model ambiguity by numbering all models belonging to the same solution. For example, if three consecutive rows are numbered 1, 2 and 3, then these rows belong to the same solution and represent model ambiguity. If a 1 in a row is followed by another 1, then there is no model ambiguity.
- `consistency`: The consistency score (a.k.a. inclusion score) for the partition of the data or the pooled data.
- `coverage`: The coverage score for the partition of the data or the pooled data.

Examples

```
data(Schwarz2016)
Schwarz_inter <- partition_min_inter(
  Schwarz2016,
  units = "country", time = "year",
  cond = c("poltrans", "ecotrans", "reform", "conflict", "attention"),
  out = "enlarge",
  n_cut = 1, incl_cut = 0.8,
  intermediate = c("1", "1", "1", "1", "1"))
```

Schwarz2016 *Original data used by Schwarz (2016)*

Description

A dataset containing the calibrated set values for the article: Schwarz, Oliver (2016): Two Steps Forward One Step Back: What Shapes the Process of EU Enlargement in South-Eastern Europe? Journal of European Integration 38 (7): 757-773.

Usage

```
Schwarz2016
```

Format

A data frame with 74 rows and 9 variables:

Case.ID Country-year ID

enlarge Progress in the EU accession process

poltrans Democracy status of the country

ecotrans Market economy status of the country

reform State of reform policy

conflict Mean conflict intensity in a country per year

attention EU's attention to the issue of enlargement

year Year ID

country Country ID

Source

Schwarz (2016) <doi:10.1080/07036337.2016.1203309>

`Thiem2011` *Original data used by Thiem (2011)*

Description

A dataset containing the calibrated set values for the article: Thiem, Alrik (2011): Conditions of Intergovernmental Armaments Cooperation in Western Europe, 1996-2006. European Political Science Review 3 (1): 1-33.

Usage

```
Thiem2011
```

Format

A data frame with 165 rows and 10 variables:

id Country-year ID

year Time ID

country Country ID

memberfs Monadic count of membership in formal intergovernmental agreements on armaments cooperation

fedismfs Degree to which a country's domestic constitutional setup is federalist in character

homogtyfs Bilateral interaction scores based on all UN and NATO military missions conducted between 1996 and 2006

powdifffs Score to measure a country's military power based on the CINC score

comptvnsfs Competitiveness of a country's domestic armaments industry

pubsupfs Public support for cooperation in defence

ecodpcefs Degree of economic dependence

Source

Thiem(2011) <doi:10.1017/S1755773910000251>

`upset_conditions` *Aggregation of individual conditions over partition-specific models*

Description

Models that have been derived for individual partitions are first decomposed into conditions, that is single conditions or conditions that are INUS (insufficient conditions that are necessary parts of a conjunction that is unnecessary and sufficient). The individual conditions are aggregated using UpSet plots to determine how frequent they are individually and in combination.

Usage

```
upset_conditions(df, nsets)
```

Arguments

`df`	Dataframe created with `partition_min` or `partition_min_inter`.
`nsets`	Number of sets to include in plot (default is 5).

Value

An UpSet plot produced with `upset`.

Examples

```
data(Grauvogel2014)
GS_pars <- partition_min(
 dataset = Grauvogel2014,
 units = "Sender",
 cond = c("Comprehensiveness", "Linkage", "Vulnerability",
          "Repression", "Claims"),
 out = "Persistence",
 n_cut = 1, incl_cut = 0.75,
 solution = "P",
 BE_cons = rep(0.75, 3),
 BE_ncut = rep(1, 3))
upset_conditions(GS_pars, nsets = 5)
```

upset_configurations *Aggregation of individual configurations over partition-specific models*

Description

Models that have been derived for individual partitions are first decomposed into sufficient terms, that is single sufficient conditions or configurations. The individual terms are aggregated using UpSet plots to determine how frequent they are individually and in combination.

Usage

```
upset_configurations(df, nsets)
```

Arguments

df	Dataframe created with partition_min or partition_min_inter.
nsets	Number of sets to include in plot (default is 5).

Value

An UpSet plot produced with upset.

Examples

```
data(Grauvogel2014)
GS_pars <- partition_min(
 dataset = Grauvogel2014,
 units = "Sender",
 cond = c("Comprehensiveness", "Linkage", "Vulnerability",
          "Repression", "Claims"),
 out = "Persistence",
 n_cut = 1, incl_cut = 0.75,
 solution = "P",
 BE_cons = rep(0.75, 3),
 BE_ncut = rep(1, 3))
upset_configurations(GS_pars, nsets = 4)
```

wop *Weight of partitions for pooled solution parameters for conservative or parsimonious solution*

Description

wop calculates the contribution or weight of partitions for the pooled solution parameters of consistency and coverage for the conservative or parsimonious solution.

Usage

```
wop(dataset, units, time, cond, out, n_cut, incl_cut, solution, amb_selector)
```

Arguments

dataset	Calibrated pooled dataset for partitioning and minimization of pooled solution.
units	Units that define the within-dimension of data (time series).
time	Periods that define the between-dimension of data (cross sections).
cond	Conditions used for the pooled analysis.
out	Outcome used for the pooled analysis.
n_cut	Frequency cut-off for designating truth table rows as observed in the pooled analysis.
incl_cut	Inclusion cut-off for designating truth table rows as consistent in the pooled analysis.
solution	A character specifying the type of solution that should be derived. C produces the conservative (or complex) solution, P the parsimonious solution. See wop_inter for deriving intermediate solution.
amb_selector	Numerical value for selecting a single model in the presence of model ambiguity. Models are numbered according to their order produced by minimize by the QCA package.

Value

A dataframe with information about the weight of the partitions with the following columns:

- type: The type of the partition. between stands for cross-sections; within stands for time series. pooled stands information about the pooled data.
- partition: Type of partition. For between-dimension, the unit identifiers are listed (argument units). For the within-dimension, the time identifiers are listed (argument time). The entry is - for the pooled data.
- denom_cons: Denominator of the consistency formula. It is the sum over the cases' membership in the solution.
- num_cons: Numerator of the consistency formula. It is the sum over the minimum of the cases' membership in the solution and the outcome.
- denom_cov: Denominator of the coverage formula. It is the sum over the cases' membership in the outcome.
- num_cov: Numerator of the coverage formula. It is the sum over the minimum of the cases' membership in the solution and the outcome. (identical to num_cons)

Examples

```
data(Thiem2011)
wop_pars <- wop(
  dataset = Thiem2011,
  units = "country", time = "year",
```

```
  cond = c("fedismfs", "homogtyfs", "powdifffs", "comptvnsfs", "pubsupfs", "ecodpcefs"),
  out = "memberfs",
  n_cut = 6, incl_cut = 0.8,
  solution = "P",
  amb_selector = 1)
wop_pars
```

`wop_inter` *Calculation of weight of partitions in pooled solution parameters for intermediate solution*

Description

`wop_inter` calculates the weight of partitions in the pooled solution parameters (consistency, coverage) for the intermediate solution.

Usage

```
wop_inter(
  dataset,
  units,
  time,
  cond,
  out,
  n_cut,
  incl_cut,
  intermediate,
  amb_selector
)
```

Arguments

`dataset`	Calibrated pooled dataset for partitioning and minimization
`units`	Units defining the within-dimension of data (time series)
`time`	Periods defining the between-dimension of data (cross sections)
`cond`	Conditions used for the pooled analysis
`out`	Outcome used for the pooled analysis
`n_cut`	Frequency cut-off for designating truth table rows as observed
`incl_cut`	Inclusion cut-off for designating truth table rows as consistent
`intermediate`	A vector of directional expectations to derive the intermediate solutions
`amb_selector`	Numerical value for selecting a single model in the presence of model ambiguity. Models are numbered according to their order produced by `minimize` by the `QCA` package.

Value

A dataframe with information about the weight of the partitions for pooled consistency and coverage scores and the following columns:

- `type`: The type of the partition. `between` stands for cross-sections; `within` stands for time series. `pooled` stands information about the pooled data.
- `partition`: Type of partition. For between-dimension, the unit identifiers are listed (argument `units`). For the within-dimension, the time identifiers are listed (argument `time`). The entry is - for the pooled data.
- `denom_cons`: Denominator of the consistency formula. It is the sum over the cases' membership in the solution.
- `num_cons`: Numerator of the consistency formula. It is the sum over the minimum of the cases' membership in the solution and the outcome.
- `denom_cov`: Denominator of the coverage formula. It is the sum over the cases' membership in the outcome.
- `num_cov`: Numerator of the coverage formula. It is the sum over the minimum of the cases' membership in the solution and the outcome. (identical to `num_cons`)

Examples

```
data(Schwarz2016)
```

Index