

資 料

質的比較分析 (QCA) のソフトの使用方法—fs/QCAと R の QCA・SetMethods パッケージ(2)

森 大 輔

前回の森(2017)で取り上げたQCAの分析では、各変数は0か1のみだった。QCAでは各変数は集合を表し、1が集合への帰属、0が非帰属を表す。しかし、0と1の2つのみに分けるのではなく、もっと微妙な差異も表現したい場合があるかもしれない。そのような場合に、通常の集合(クリスプ集合 crisp set)を拡張したファジィ集合(fuzzy set)を利用する⁽¹⁾。今回は、このファジィ集合を用いたQCA(ファジィ集合QCA⁽²⁾)の、fs/QCAと、RのQCAパッケージ(およびSetMethodsパッケージ、vennパッケージ)での行い方を主に扱う⁽³⁾。

1. QCA専用ソフトウェアfs/QCA

1.1. fs/QCAでのファジィ集合QCA

ファジィ集合では、0と1だけでなく、その間の小数も値として認める。この値を帰属度(メンバーシップ度 membership score)と呼ぶ。そして、帰属度が0のとき集合に完全非帰属、0より大きく0.5未満のとき集合に非帰属寄り、0.5より大きく1未満のとき集合に帰属寄り、1のとき集合に完全帰属の状態と考える。ここでは、ファジィ集合QCAのfs/QCAでの分析方法を説明する。ただしファジィ集合QCAも、実はクリスプ集合QCAと大部分は手順が共通している⁽⁴⁾。

1.1.1. ファジィ集合帰属度への変換(手動での割り振り)

1) データを用意する。ここでは、弥永(2009:205)にある、株主総会や取締役

役会の決議を欠いた行為が有効とされるか否かという効力の問題について、ファクターで考えるという内容の下表 1 を、例として用いる⁽⁵⁾。

表 1：株主総会や取締役会の決議を欠いた行為の効力（弥永 2009：205 による）

	①	②	③	④	⑤	効力
総会決議						
資本減少	×	×	×	×	×	無効
剰余金配当	×	×	×	×	×	無効
合併	×	○	×	×	×	無効
事業譲渡	×	○	△	○	×	相対的無効
新株有利発行	×	○	×	○	○	有効
取締役会決議						
株主総会招集	○	×	○	△	△	無効
利益相反取引	○	△	○	△	○	相対的無効
多額の借財	○	○	○	△	○	相対的無効
社債発行	○	○	○	○	○	有効
競業	○	○	○	○	○	有効
新株発行	○	○	○	○	○	有効

注：①株主にとっての重要性は低い、②内部的事項にとどまらない、③第三者は瑕疵の存否を知りにくい、④無効とした場合にそれが第三者に連鎖的に影響を及ぼす、⑤無効としない場合に株主・会社の不利益の回復は比較的容易である

2) ファジィ集合の帰属度への変換（これをキャリブレーション calibration と呼ぶ）を行う。変換には様々な方法がある⁽⁶⁾が、簡単なのは、帰属度を等間隔で 4 つに分けた 4 値ファジィ集合帰属度（0, 0.33, 0.67, 1）を用い、各事例がどの帰属度に最も近いかを考えるものである⁽⁷⁾。

このとき注意点は、0 と 1 のちょうど真ん中である 0.5 が入らないような分け方にするということである。0.5 は、帰属か非帰属か全くわからない情報なしの状態を意味するからである。よって今の例では、△や相対的無効を 0.5 とすることはできないので、帰属寄りの 0.67 か非帰属寄りの 0.33 のどちらにするか考える必要がある⁽⁸⁾。このときには、法学における議論など、分析者の有する事例に関する知識を用いて考えることになる⁽⁹⁾（石田・齋藤監訳 2016：142）。

質的比較分析 (QCA) のソフトの使用方法

ここでは、相対的無効は、当事者間では当然に無効になるし悪意の第三者に対しても無効になるので、無効というところに分析の力点があると考え非帰属寄りの0.33とする⁽¹⁰⁾。また、△は、すべて0.67としておく⁽¹¹⁾。

3) データすべてを2)の方法によって変換する。今の例では、この結果、表2ができあがる。以下では、これをファイルCorporate.csvに保存して分析に使用するものとする。

表2：表1をファジィ集合の帰属度へ変換したもの

Case	Unimportant	Noninternal	Ignorant	Impact	Recovery	Effect
Shihon	0	0	0	0	0	0
Yojo	0	0	0	0	0	0
Gappei	0	1	0	0	0	0
Jigyō	0	1	0.67	1	0	0.33
Yuri	0	1	0	1	1	1
Kabunushi	1	0	1	0.67	0.67	0
Rieki	1	0.67	1	0.67	1	0.33
Tagaku	1	1	1	0.67	1	0.33
Shasai	1	1	1	1	1	1
Kyogyō	1	1	1	1	1	1
Shinkabu	1	1	1	1	1	1

注：Case：事例の名前、UnimportantからRecoveryは表1の①～⑤に当たる、Effect：行為が有効である

以下の分析については、ファジィ集合であってもクリスプ集合の場合と基本的には変わらず、森（2017）の1.2や1.3と同様に行う⁽¹²⁾。

1.1.2 必要条件の分析

以下の1)～6)は森（2017：247）の1.2.1と同様である。

- 1) メニューのAnalyzeをクリックする。
- 2) 出てくるメニューからNecessary Conditionsを選択する。
- 3) Dialogというウィンドウが現れる。Outcomeの下の特ダウから、「結果」

資 料

として使用する集合である Effect を選ぶ。

4) Add Condition の下のプルダウンから、原因条件とその否定すべて (Unimportant、~Unimportant、Noninternal、~Noninternal、Ignorant、~Ignorant、Impact、~Impact、Recovery、~Recovery) を選ぶ。

5) OK を押す。

6) 「原因条件」として使用する変数とその否定すべてについて、必要条件の整合度と被覆度が表示される。整合度や被覆度の解釈もクリスプ集合の場合と変わらない。整合度は、その条件が必要条件である度合いを表し、被覆度は、その条件が取るに足らない必要条件である度合いを表す。整合度の閾値を0.9とすると、今の例では Noninternal (整合度 1、被覆度0.650587)、Impact (整合度 1、被覆度0.711840)、Recovery (整合度0.933868、被覆度0.698651) の3つが必要条件の可能性があると判断される。また、被覆度がある程度大きいので、これらは取るに足らない必要条件でないこともわかる。

以下の7)以降は、ファジィ集合の分析特有のものである。ファジィ集合の場合、XYプロットと呼ばれる図で必要条件関係を視覚的に表現する機能⁽¹³⁾が、fs/QCAには備わっている。ここでは、Recoveryが結果Effectの必要条件であるという関係をXYプロットで図示してみる(他の必要条件のNoninternalやImpactも以下と同様に図示できる)。

7) メニューの Graph をクリックする。

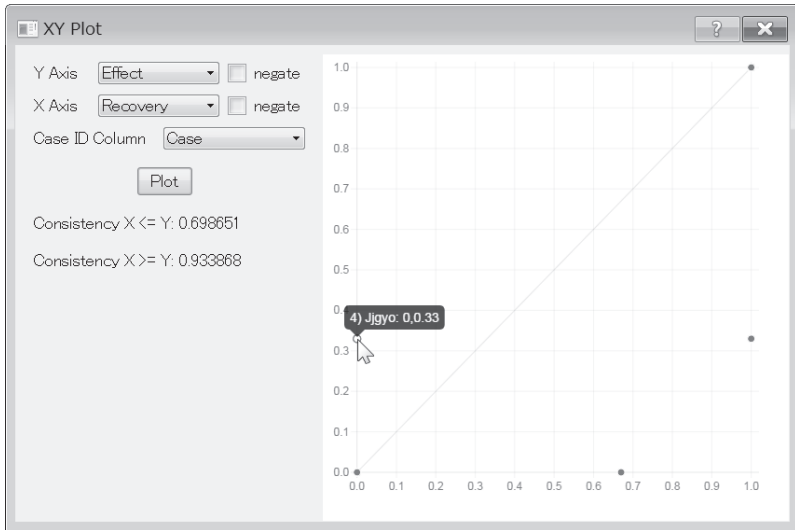
8) 出てくるメニューから XY Plot を選択する。

9) XY Plot というタイトルのウィンドウが現れる。Y Axis と X Axis という表示があるが、それぞれ Y 軸と X 軸を意味する。XY プロットは、Y 軸に「結果」の集合をとり、X 軸に「原因条件」の集合をとる。今の場合、Y Axis の横のプルダウンで Effect を選択し、X Axis の横のプルダウンで Recovery を選択することになる。

10) Case ID Column の横のプルダウンでは、事例の番号や名前を記している変数を選択する。今の例では、Case に事例の名前を記しているのでこれを選択

する。

- 11) Plot というボタンを押すと、右側の欄に XY プロットが表示される。



- 12) 図中の点は、各事例を表している。この点の上にポインタを当てると、その点の表す事例と、X軸の値（原因条件の集合への帰属度）、Y軸の値（結果の集合への帰属度）が表示される。例えば、図のY軸上の点の所にポインタを当てると、4) Jigyo: 0, 0.33と表示され、表2の行4のJigyoという事例がここに位置していることがわかる。

- 13) X軸に指定した集合がY軸に指定した集合の必要条件である場合、図中の点は原点からの45度線（対角線）よりも右下にある⁽¹⁴⁾。完全な必要条件（必要条件の整合度が1のとき）であれば、図中の点はすべて対角線よりも右下にあるが、必要条件の整合度が1よりも小さい場合は、図中の点の一部が対角線よりも左上にある⁽¹⁵⁾。対角線よりも左上にある点が表す事例は、X軸に指定した集合がY軸に指定した集合の必要条件であるという主張とは矛盾する事例ということになる⁽¹⁶⁾。今の例では、12)でも述べたJigyoが対角線の左上にある矛盾した事例だとわかる。

資 料

- 14) Plot ボタンの下に、必要条件の整合度と被覆度の値が表示される。
Consistency $X \leq Y$ が必要条件の被覆度を表し、Consistency $X \geq Y$ が必要条件の整合度を表している⁽¹⁷⁾。今の場合、整合度が0.933868で被覆度が0.698651となる。
- 15) この図の上で右クリックして Copy Image を選択すると、図をコピーすることができる。そして描画ソフトに貼り付けることで、図を編集したり画像形式で保存したりできる⁽¹⁸⁾。

1.1.3 十分条件の分析

以下の 1) ～13) は森 (2017: 245) の1.2.2と同様である。

- 1) メニューの Analyze をクリックする。
- 2) 出てくるメニューから Truth Table Algorithm を選択する。
- 3) Select Variables ウィンドウが現れる。左の variables の欄から、「結果」として使用する集合 (Effect) を選び、Set というボタンを押す。すると、右の outcome の欄に、その集合が移る。
- 4) 左の variables の欄から、「原因条件」として使用する集合 (Unimportant、Noninternal、Ignorant、Impact、Recovery) を選び、Add というボタンを押す。すると、右の causal condition の欄に、その集合が移る。
- 5) Show solution cases in output のボックスにチェックを入れて、その横のプルダウンから、事例の番号や名前を記している変数 (Case) を選択する。
- 6) OK を押す。
- 7) クリスポ集の場合と同様に、真理表が現れる。
- 8) cases という列の cases ボタンを押すと、真理表の各行に属する事例の番号や名前が見られる。
- 9) 結果 Effect の集合の列が空欄なので、raw consist. (整合度) の値が 1 である行 (今回は行 5) には 1 を、raw consist. の値が 0 である行 (今回は行 2、3) には 0 を入力し、raw consist. が空欄である行は空欄のままにする。また、raw consist. の値が小数である行 (今回は行 1、4、6) については、森 (2017:

235) で述べられているように以下の2つの方法がある。①原因条件を追加したり、既存の原因条件の値を見直したりする。②raw consist. の値がある閾値(例えば0.8) 以上の場合には結果の欄に1を入力し、ある閾値より下の場合には結果の欄に0を入力する。ファジィ集合の場合は①の方法で整合度が完全に小数でなくなることはほとんどないので、②の方法を用いる。各行について手で結果の欄に1か0を入力してもよい⁽¹⁹⁾ し、次のようにソフトに行わせることもできる。

②-1) Edit Truth Table ウィンドウのメニューのEditをクリックし、出てくるメニューの中からDelete and codeを選択する。

②-2) Dialog というタイトルのウィンドウが出る。and set Counsel to 1 for rows with consist>という欄で、raw consist. の閾値を設定する。デフォルトは0.8であるが、他の値を設定することもできる。今はこの0.8を閾値として使用することにする。

②-3) OKを押すと、raw consist. の値が0.8以上の行には結果の欄に1、小さい行には0が自動で入り、raw consist. が空欄の行は削除される。

10) 上の②-3) の方法などによりraw consist. が空欄の行が削除されたら、真理表の右下にあるStandard Analysesというボタンをクリックする。このときに主項の選択という画面が現れることもある(今の例では現れない)。その対処については森(2017:234)を参照されたい。

11) Intermediate solution ウィンドウで、各原因条件の存在が結果の発生に寄与すると予想されるのか(Presentを選択)、それとも各原因条件の欠如が結果の発生に寄与すると予想されるのか(Absentを選択)、特にそうした予想はできないか(Present or Absentを選択)、という方向性予想を指定する。Unimportant、Noninternal、Ignorant、Impact、Recoveryのいずれも、存在がEffect(行為が有効である)という結果の発生に寄与すると予想されるので、いずれもPresentを選択する。

12) OKを押す。

13) 出力の欄に、上から順に複雑解、簡潔解、中間解という3つの解が表示され

る（表示の詳しい読み方は森2017：240を参照）。今の例の場合、複雑解は Noninternal*Impact*Recovery*(~Unimportant*~Ignorant+Unimportant*Ignorant) → Effect⁽²⁰⁾、簡潔解は Noninternal*Recovery→Effect、中間解は Noninternal*Impact*Recovery→Effect である⁽²¹⁾。

それぞれについて、解整合度や解被覆度や解の項に属する事例も表示される。ここでは中間解のみ記しておく。中間解の解整合度は0.872659、解被覆度は0.933868である。中間解の項に属する事例については、fs/QCAの出力は Cases with greater than 0.5 membership in term Noninternal*Impact*Recovery: Yuri (1,1)、Shasai (1,1)、Kyogyo (1,1)、Shinkabu (1,1)、Rieki (0.67,0.33)、Tagaku (0.67,0.33) となる。つまりこれらは、Noninternal*Impact*Recoveryへの帰属度が0.5より大きい事例である。事例の番号の横に付いているカッコの中の数字は、左側がその事例の解の項への帰属度、右側がその事例の結果への帰属度を表している。つまり、RiekiとTagakuは、結果への帰属度が0.5よりも小さく、Noninternal*Impact*Recoveryが十分条件であるということと矛盾する事例だと言える。

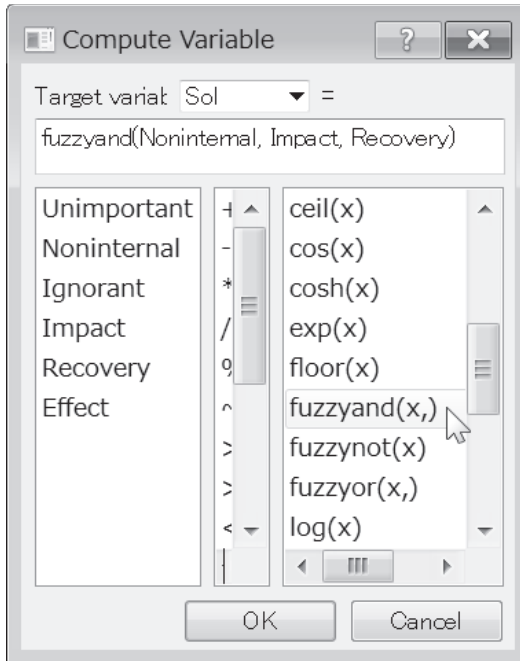
以下の14)以降は、ファジィ集合の分析特有のものである。1.1.2の必要条件の場合と同様、XYプロットにより、十分条件関係を視覚的に表現することができる。ここでは、中間解の十分条件関係を表示することにする。そのためにはまず、中間解の項 Noninternal*Impact*Recovery の集合を表す変数を作る必要がある。以下の14)～19)でそれを行っている。

- 14) メニューの中の Variables をクリックする。
- 15) 出てくるメニューから Compute を選択する。
- 16) Compute Variables ウィンドウが現れる。一番上の Target Variable の欄に新しい変数の名前をキーボードで入力する。ここでは Sol という名前をつけることにする⁽²²⁾。
- 17) 一番右の欄の中にある fuzzyand (x,) をクリックする⁽²³⁾。すると上の横長の

欄に `fuzzyand(x,)` が入力される。

18) `fuzzyand(x,)` を、`fuzzyand(Noninternal, Impact, Recovery)` と書き換える。

`Noninternal*Impact*Recovery` は `Noninternal` と `Impact` と `Recovery` が論理積でつながれているためである。



19) OK を押すと、元のウィンドウの左側のデータの一番右の列に、新しい Sol の列が付け加わる。

以下の20) 以降でXY プロットを表示する。

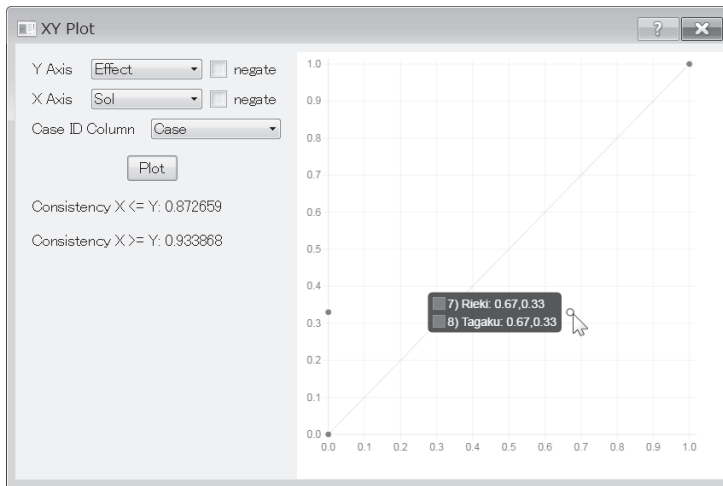
20) メニューの Graph をクリックする。

21) 出てくるメニューから XY Plot を選択する。

22) XY Plot というタイトルのウィンドウが現れる。Y Axis で「結果」である Effect を選択し、X Axis で Sol を選択する。

23) Case ID Column で、Case を選択する。

24) Plot というボタンを押すと、右側の欄にXYプロットが表示される。



25) 図中の点は、各事例を表している。この点の上にポインタを当てると、その点の表す事例と、X軸の値（原因条件の集合への帰属度）、Y軸の値（結果の集合への帰属度）が表示される。

26) X軸に指定した集合がY軸に指定した集合の十分条件である場合、図中の点は原点からの45度線（対角線）よりも左上にある⁽²⁴⁾。完全な十分条件（十分条件の整合度が1のとき）であれば、図中の点はすべて対角線よりも左上にある⁽²⁵⁾が、十分条件の整合度が1よりも小さい場合は、図中の点の一部が対角線よりも右下にある⁽²⁶⁾。対角線よりも右下にある点を表す事例は、X軸に指定した集合がY軸に指定した集合の十分条件であるという主張とは矛盾する事例ということになる。今の例でも、対角線の右下に点があり、ポインタを当てると表2の行7のRiekiと行8のTagakuであることがわかる⁽²⁷⁾。

27) Plot ボタンの下に、必要条件の整合度と被覆度の値が表示される。Consistency $X \leq Y$ が十分条件の整合度を表し、Consistency $X \geq Y$ が十分条件の被覆度を表している⁽²⁸⁾。今の場合、整合度が0.872659で被覆度が

0.933868となる。

28) この図の上で右クリックして Copy Image を選択すると、図をコピーすることができるのは1.1.2の15) と同様である。

1.1.4 結果の不存在の分析

これまでの1.1.2や1.1.3は結果の存在についての分析であった。結果の不存在についても必要条件や十分条件を明らかにしたい場合、別個に分析を行う必要がある⁽²⁹⁾が、その手順は1.1.2や1.1.3と基本的に同様であり、また森(2017:238)でも説明をした。以下、注意する点のみ記しておく。

必要条件の分析では、1.1.2の3)でEffectの代わりに~Effectを選ぶ。9)でY AxisでEffectを選択した後、横のnegateのボックスにチェックを入れる(これでY軸は~Effectとなる)。

十分条件の分析では、1.1.3の3)でSetの代わりにSet Negatedというボタンを押す。9)は全く同様にraw consist.が1なら1、0なら0、空欄なら空欄、小数なら閾値以上を1で閾値より下を0とする。11)の方向性予想は、Presentの代わりにいずれもAbsentを選択する(結果の存在に寄与すると予想されているので、結果の不存在には寄与しないからである)。24)ではY AxisでEffectを選択した後、横のnegateのボックスにチェックを入れる。

1.2. 補足：その他の補足的・発展的な事項

前回の森(2017)においてクリスプ集合のQCA、そして今回の1.1でファジィ集合のQCAの方法を説明した。以下では、それらに関する補足的な、あるいは発展的な事項について説明する。1.2.1はファジィ集合に特有の事柄だが、1.2.2はファジィ集合に限らずクリスプ集合でも問題になる事柄である。

なお、以下では、表3のデータを用いる。これは、森(2012)で用いたデータを簡略化したもので、1930年に行われた国家責任法の法典化会議で条約案に賛成票を投じた国と反対票を投じた国の状況を表している。以下では、これをファイルState.csvに保存して分析に使用するものとする。

表 3：1930年頃の各国の状況（森2012：78による）

	code	intl	steel	ind1	ind2	uk	latin	eastern
オーストラリア	AUL	1	394	0.67	0.56	1	0	0
オーストリア	AUS	1	552	0.67	0.63	0	0	0
ベルギー	BEL	1	3678	1	1	0	0	0
カナダ	CAN	1	1078	0.67	0.83	1	0	0
エストニア	EST	1	0	0	0.03	0	0	0
フィンランド	FIN	1	25	0	0.04	0	0	0
フランス	FRN	1	9121	1	1	0	0	0
ドイツ	GMV	1	14390	1	1	0	0	0
ギリシア	GRC	1	0	0	0.03	0	0	0
インド	IND	1	0	0	0.03	1	0	0
アイルランド	IRE	1	0	0	0.03	1	0	0
イタリア	ITA	1	1840	1	0.96	0	0	0
日本	JPN	1	1953	1	0.96	0	0	0
ノルウェー	NOR	1	17	0	0.04	0	0	0
オランダ	NTH	1	0	0	0.03	0	0	0
南アフリカ	SAF	1	37	0	0.05	1	0	0
スペイン	SPN	1	798	0.67	0.74	0	0	0
スウェーデン	SWD	1	575	0.67	0.65	0	0	0
スイス	SWZ	1	0	0	0.03	0	0	0
イギリス	UKG	1	7758	1	1	0	0	0
アメリカ	USA	1	49157	1	1	0	0	0
ブラジル	BRA	0	17	0	0.04	0	1	0
チリ	CHL	0	0	0	0.03	0	1	0
中国	CHN	0	25	0	0.04	0	0	0
コロンビア	COL	0	0	0	0.03	0	1	0
チェコスロバキア	CZE	0	1803	1	0.95	0	0	1
ハンガリー	HUN	0	433	0.67	0.58	0	0	1
ペルシア	IRN	0	0	0	0.03	0	0	0
メキシコ	MEX	0	86	0.33	0.09	0	1	0
ニカラグア	NIC	0	0	0	0.03	0	1	0
ポーランド	POL	0	1217	0.67	0.86	0	0	1
ポルトガル	POR	0	0	0	0.03	0	0	0
ルーマニア	ROM	0	143	0.33	0.16	0	0	1
エルサルバドル	SAL	0	0	0	0.03	0	1	0
トルコ	TUR	0	0	0	0.03	0	0	0
ウルグアイ	URU	0	0	0	0.03	0	1	0
ユーゴスラビア	YUG	0	81	0.33	0.08	0	0	1

注：Code：国の略号、intl：賛成票を投じた、steel：鉄鋼生産量（1926～1930年平均）、ind1：工業国である（4 値への手動での割り振り）、ind2：工業国である（直接法）、uk：英連邦の国である、latin：ラテンアメリカの国である、eastern：東欧の国である

1.2.1. ファジィ集合帰属度への変換の別の方法（「直接法」）

鉄鋼生産量（表3ではsteel）という量的な変数を、「工業国である」という集合への帰属度に変換することを考える。森（2012：74）では、1.1.1で説明したような4値ファジィ集合帰属度への手動での割り振りをした（表3ではind1）。その際、以下のような基準で割り振った。まず0.33と0.67の境目は、AULとROMの間とする。例えば、戦間期の経済について記述した国際連合の出版物では、AULが開発・半開発国に分類されているのに対し、ROMは低開発国に分類されている（United Nations 1949）。また、境目の決定にあたっては分布でのギャップの大きさも参考になる（Schneider & Wagemann 2012:35参照）。実際、この場合もAULとROMの生産量の差は大きい。0.67と1の境目はCZEとPOLの間、0.33と0の境目はYUGとSAFの間にする。前者はPOLがUnited Nations（1949）で低開発に分類されているのに対し、CZEは評価が高く開発国に分類されていることからである。後者は、両者の間の差が近辺のデータでは比較的大きいことによる。

このような手動による方法以外に、関数（代表的なものはロジスティック関数⁽³⁰⁾）を利用してファジィ集合帰属度に変換する方法がある。この方法は直接法（direct method）と呼ばれている（Ragin 2008:89; Schneider & Wagemann 2012:35）。直接法には、帰属度が4値や6値よりもきめ細くなるという利点がある。fs/QCAでは、ロジスティック関数による直接法を使用することができる。例として、表3のsteelにこれを使用して「工業国である」という集合ind2を作成してみる。以下のように行う。

- 1) メニューの中のVariablesをクリックする。
- 2) 出てくるメニューからComputeを選択する。
- 3) Compute Variables ウィンドウが現れる。Target Variableの欄に新しい変数の名前ind2を入力する⁽³¹⁾。
- 4) 一番右の欄の中にあるcalibrate (x,n1,n2,n3) をクリックする。すると上の横長の欄にcalibrate (x,n1,n2,n3) が入力される。

7) calibrate (x,n1,n2,n3) の中の x を変換する変数名、n1 を完全帰属（帰属度 1）とそうでないものの境目の値⁽³²⁾、n2 を帰属寄りと非帰属寄りの境目の値（帰属度 0.5 に相当する値）、n3 を完全非帰属（帰属度 0）とそうでないものの境目の値⁽³³⁾ に置き換える。すなわち、関数を使用する場合にも、完全に自動で行うわけではなく、n1 と n2 と n3 の値はやはり分析者が有する事例に関する知識を用いて考えることになる。

今の場合、x は steel である⁽³⁴⁾。n1 と n3 については、手動での割り振りの場合と同様に考え、n1 は CZE まだが完全帰属だと考えて CZE の値である 1803 とし、n3 は SAF まだが完全非帰属だと考えて SAF の値である 37 とする。n2 については、手動の場合は帰属度 0.5 に相当する場所を考える必要はなく、この点は手動の場合との違いである（Schneider & Wagemann 2012:38）。ここでは帰属寄りの下限 AUL と非帰属寄りの上限 ROM の中間の値 269 としておく。以上から、calibrate (steel, 1803, 269, 37) と入力することになる。

8) OK を押すと、元のウィンドウの左側のデータの一番右の列に、新しい ind2 の列が付け加わる。

なお、手動による割り振り ind1 と直接法 ind2 のどちらが適切かという問題があるかもしれない。ind2 の方が元の値の細かな違いまで反映しているのでよいように思えるが、そうとも限らない。例えば今回の例のデータについて言えば、元の値が時代の古いものでありどこまで正確かという問題があるし、仮に正確だったとしても、鉄鋼生産量の値の大小を「工業国である」という集合への帰属度の大小に完全に対応させるのが果たして「工業国である」という集合の意味として適切なのかという問題もある（Verkuilen 2005:470 も参照）。

しかし、ind1 と ind2 の値を比較してみるとわかるように、2 つにそれほど大きな差はない。一番大きな差で YUG の 0.25 ほどであり、非帰属寄りだったものが帰属寄りになったり⁽³⁵⁾、その逆になったりといった大きな変化はない。一般に、帰属度への変換は、帰属寄りと非帰属寄りの区別が変わってしまうようなものでなければ、変換の方法の違いなどが結果にそれほど影響を与えることはな

い⁽³⁶⁾ (Schneider & Wagemann 2012:37-38)。実際に、表 3 について反対票を投じた国 (~intl) の必要条件の分析をしても、ind1 と ind2 のどちらも必要条件にならず整合度や被覆度が多少変わる程度である⁽³⁷⁾。~intl の十分条件の分析でも、解の項は変わらず、被覆度が多少変わる程度である⁽³⁸⁾。

1.2.2. 論理的残余の扱いについて発展的な事項

十分条件の分析の過程では真理表を作り、真理表の行のうち事例が存在していない行（論理的残余）については、もし事例が存在していたらその行の結果はどうなるかという反事実的仮定（反実仮想、counterfactual）を置くことが行われている。その結果導出されるのが簡潔解と中間解である。森（2017：238）などで述べられているように、簡潔解ではとにかく解の項の形が単純になるように反事実的仮定を置き、中間解ではそこから方向性予想に基づいてもっともらしい反事実的仮定（容易な反事実的仮定）のみを残しもっともらしくない反事実的仮定（困難な反事実的仮定）のみを残すということを行っていた。

しかし、もっともらしくない反事実的仮定は、このような方向性予想に基づくもの以外にも存在するということが、最近認識されてきている (Schneider & Wagemann 2012:197-219)。例えば、以下のようなものが議論されている。

表 4：反対票を投じる国 (~intl) についての真理表

ind1	uk	latin	eastern	number	~intl	raw consist.
0	0	0	0	10	0	0.363967
1	0	0	0	10	0	0
0	0	1	0	7	1	1
0	1	0	0	3	0	0
1	0	0	1	3	1	1
1	1	0	0	2	0	0
0	0	0	1	2	1	1
1	0	1	0	0		
0	1	1	0	0	◆	
1	1	1	0	0	◆	
0	1	0	1	0	◆	
1	1	0	1	0	◆	
0	0	1	1	0	○	
1	0	1	1	0	○	
0	1	1	1	0	○◆	
1	1	1	1	0	○◆	

1.2.2.1 不可能残余についての反事実的仮定

表3について、反対票を投じた国（~intl）の十分条件の分析をすることを考える。原因条件はind1、uk、latin、easternの4つを使用する⁽³⁹⁾。真理表を作るところまでは1.1.3の8）までと同様である。真理表は表4のようになる。

しかし、真理表の論理的残余の中に、latinとeasternの両方が1のものがあるが（表4では~intlの列に○と記載されている行13から行16）、このような事例は物事の性質上存在しえない。なぜなら、これら両方が1ということはラテンアメリカと東欧の両方に分類される国ということになるが、地理的にそのような国はありえないからである。このように、事例が存在していないのが物事の性質上ありえないからである論理的残余を不可能残余（impossible remainder）と呼ぶ（Schneider & Wagemann 2012:155）。このような不可能残余について、結果~intlが存在しているという反事実的仮定を置くのはおかしいと考えられる（Schneider & Wagemann 2012:206）。

簡潔解や中間解において、そのままではこのような不可能残余が含まれてしまう可能性がある。不可能残余が簡潔解や中間解に含まないようにするには、不可能残余の行の結果の欄に0を手動で入力しておけばよい。ここで0を入力するのはこの行の結果が0（今の例では賛成票を投じた）と仮定しているわけではなく、fs/QCAの動作として、結果の欄に0が入力されていると、その行は簡潔解や中間解に含まれなくなるからである。

その後は、1.1.3の9）以下と同様である。すなわち、真理表の他の行の結果の欄をそれぞれ1か0か空欄か決定した後、真理表の空欄の行を削除して⁽⁴⁰⁾、Standard Analysesのボタンを押せば結果が得られる。

実際この例の場合、不可能残余について処置をしなかった場合と処置をした場合とで簡潔解や中間解が異なるため、処置をすることに意味がある。例えば中間解について比べてみると、処置をしなかった場合は~uk*eastern + ~ind1*~uk*latin → ~intlとなるのに対し、処置をした場合は~uk*~latin*eastern + ~ind1*~uk*latin*~eastern → ~intlとなり⁽⁴¹⁾、解の項が異なることがわかる⁽⁴²⁾。

1.2.2.2. 必要条件と矛盾する反事実的仮定

表 3 について、1.1.2 と同様に必要条件の分析をする。すると、 $\sim uk$ が整合度 1 で、これのみが閾値 0.9 を超えるので、 $\sim uk$ が $\sim intl$ の必要条件、つまり $\sim uk \leftarrow \sim intl$ だと判断される。このとき、論理式の性質から対偶 $uk \rightarrow intl$ も成り立っていないからではない。そのため、十分条件の分析の際の真理表の論理的残余で uk が 1 のもの（表 4 では $\sim intl$ の列に◆と記載されている行 9 から 12、行 15、行 16）は、 $intl$ が 1 でなければならない。今の場合は、結果は $\sim intl$ を考えているので、これらの行は $\sim intl$ が 0 でなければならない。

簡潔解や中間解において、そのままでは必要条件と矛盾する反事実的仮定が置かれてしまう可能性がある（Schneider & Wagemann 2012:201）。言い換えれば、そのままではこれらの行が解に含まれてしまう（すなわちこれらの行の $\sim intl$ が 1 になってしまう）可能性がある⁽⁴³⁾。このような、必要条件と矛盾する反事実的仮定を防ぐには、真理表を作成した段階でこれらの行の $\sim intl$ に 0 を手動で入力しておけばよい。

その後は、1.1.3 の 9）以下と同様である。すなわち、真理表の他の行の結果の欄をそれぞれ 1 か 0 か空欄か決定した後、真理表の空欄の行を削除して、Standard Analyses のボタンを押せば結果が得られる。

実際この例の場合、必要条件と矛盾する反事実的仮定を防ぐ処置をしなかった場合と処置をした場合とで簡潔解が異なる。必要条件と矛盾する反事実的仮定を防ぐ処置をしなかった場合は簡潔解は $latin + eastern \rightarrow \sim intl$ となる⁽⁴⁴⁾のに対し、処置をした場合は簡潔解は $\sim uk * latin + \sim uk * eastern \rightarrow \sim intl$ となる⁽⁴⁵⁾。

さらに、1.2.2.1 の不可能残余についての処置と、この必要条件と矛盾する反事実的仮定を防ぐ処置の両方を行うと、中間解は $\sim uk * \sim latin * eastern + \sim ind1 * \sim uk * latin * \sim eastern \rightarrow \sim intl$ となる⁽⁴⁶⁾。

このように、論理的残余の扱いについては、様々な注意を要することが認識されてきている⁽⁴⁷⁾。

2. RのQCAパッケージ

ここでは、RのQCAパッケージを使用して、fs/QCAで行ったのと同様のファジィ集合QCAの行い方を説明する。

2.1. RのQCAパッケージでのファジィ集合QCA

2.1.1 ファジィ集合帰属度への変換（手動での割り振り）

まずファジィ集合のデータを用意する必要があるが、その部分は1.1.1と全く同様なので、そちらを参照されたい。以下では1.1.1と同じデータファイルCorporate.csvを用いる。

RのQCAパッケージの起動の仕方は森（2017：231）の2.1.2と同様である。すなわち、Rのメニューで「パッケージ」をクリックし、「パッケージの読み込み」をクリックする。すると「1つを選択してください」というウィンドウが出るので、その中からQCAを選んでOKを押す。

そして、使用するデータファイルCorporate.csvの存在するフォルダに移動しておく⁽⁴⁸⁾。その後、森（2017：231）の2.1.3と同様に下のようにRに入力する。

```
> Corp <- read.csv("Corporate.csv", row.names="Case")
```

これによりデータファイルが読み込まれ、それがCorpという変数に代入される。また、事例の名前が記されている変数Caseが各行の名前に設定される。

以下の分析については、Rについてもfs/QCAと同様に、ファジィ集合であってもクリスプ集合の場合と基本的には変わらず、森（2017）の2.2や2.3と同様に行う。

2.1.2 必要条件の分析

1) 森（2017：230）の2.2.1と同様に、必要条件の分析をするためには、次のように入力する。

```
> superSubset(Corp, outcome="Effect", conditions=c("Unimportant", "Noninternal",  
"Ignorant", "Impact", "Recovery"), incl.cut=0.9, use.tilde=TRUE)
```

incl.cutを0.9としているので、整合度が0.9以上の条件や論理和、論理積が出力

される。

出力の解釈もクリスプ集合の場合と変わらない。出力のうち incl の列は必要条件の整合度、cov.r の列は必要条件の被覆度を表す。今の例では Noninternal（整合度 1、被覆度 0.651）、Impact（整合度 1、被覆度 0.712）、Recovery（整合度 0.934、被覆度 0.699）の 3 つが必要条件の可能性があると判断される⁽⁴⁹⁾。また、被覆度がある程度大きいので、これらは取るに足らない必要条件でないこともわかる。

以下の 2) 以降は、ファジィ集合の分析特有のものである。fs/QCA の場合と同様に、XY プロットで必要条件関係を視覚的に表現する。ここでは、Recovery が結果 Effect の必要条件であるという関係を XY プロットで図示してみる（他の必要条件も以下と同様に図示できる）。

2) XYプロットを表示するためには、次のように入力する。

```
> XYplot("Recovery", "Effect", Corp, relation="nec")
```

最初に X 軸にとる原因条件、次に Y 軸にとる結果の集合の名前を引用符の中に書き、データファイルを代入した変数を三番目に書いている。その後の relation のところでは、作成する XY プロットが必要条件関係のものなのか（その場合 relation="nec"）十分条件関係のものなのか（その場合 relation="suf"）を指定する。

X 軸に指定した集合が Y 軸に指定した集合の必要条件である場合、図中の点は原点からの 45 度線（対角線）よりも右下にある。

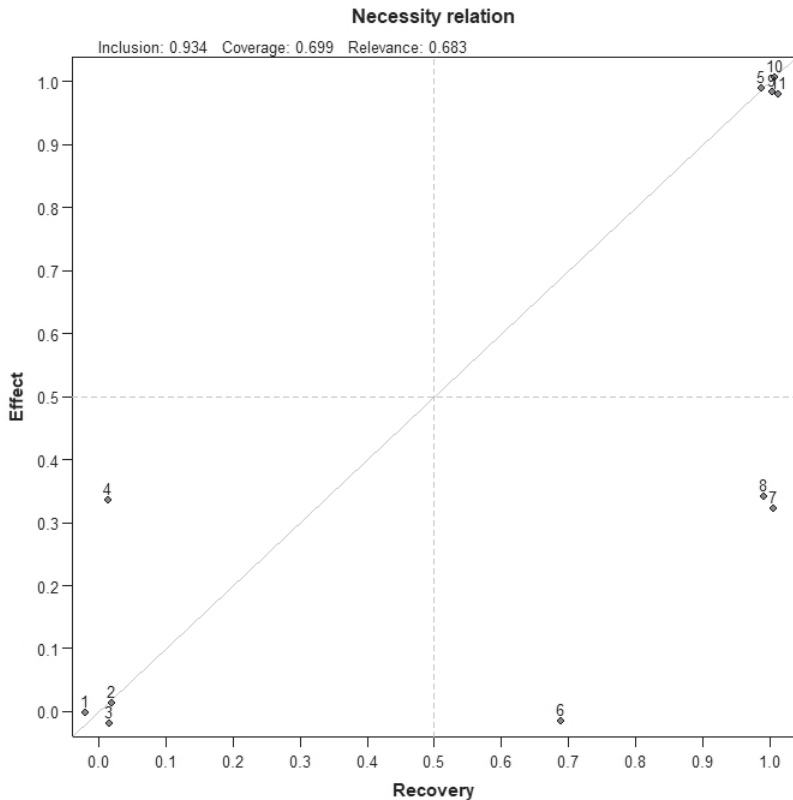
3) 上の 2) の指定だけでは、XY プロット中の各点がどの事例なのかわからない。事例がわかるようにするには、次のように入力すればよい⁽⁵⁰⁾。

```
> XYplot("Recovery", "Effect", Corp, relation="nec", labels=seq(nrow(Corp)),  
jitter=TRUE, amount=0.02)
```

labels のところに事例を表すものを入れる。事例の名前の変数 Case を使ってもよいが、名前を使うと XY プロット中に文字が重なって見づらくなりやすいので、

資 料

ここでは代わりに元のデータファイルの行番号を使っている⁽⁵¹⁾。seq (nrow (データファイルを代入した変数)) で、元のデータファイルの行番号が使える。その後の jitter は、XY プロット中の点について、何個も同じものが重なっているとき、少し点をずらして書くことにより点が重なっていることをわかりやすくする機能である。最後の amount は jitter での点のずらしの大きさを指定するもので、数字を大きくするほど点のずらしが大きくなる。



上の図が表示されるXYプロットである。上に inclusion:0.934、Coverage:0.699 という記載があるが、これは必要条件の整合度0.934で被覆度0.699という意味である⁽⁵²⁾。XYプロット中の点の上に付いている番号は、その点に対応する事例の、

質的比較分析 (QCA) のソフトの使用方法

元のデータファイルの行番号である。このXYプロットにおいて、例えば左下の1と2と3は少しずつ位置が異なっているように見えるが、これは jitter によって点をずらしているだけで、実際には完全に重なっている⁽⁵³⁾。

4) グラフの上にあるメニューの中で一番左の「ファイル」をクリックして現れるメニューにより、グラフを様々なファイル形式で保存したり（「別名で保存」をクリックしてファイル形式を選択する）、コピーしたりできる（「クリップボードにコピー」をクリックしてファイル形式を選択する）。

2.1.3 十分条件の分析

以下の1)から5)については、森（2017：229, 221）の2.2.2や2.3と同様なので、入力や出力の詳細についてはそちらを参照されたい。

1) まず真理表を表示するには、次のように入力する。

```
> CorpTable <- truthTable (Corp, outcome="Effect", conditions=c ("Unimportant",
"Noninternal", "Ignorant", "Impact", "Recovery"), complete=TRUE, show.case
=TRUE, sort.by=c ("incl"), incl.cut=0.8)
> CorpTable
```

真理表の行の整合度の閾値はデフォルトでは1になっているので、ここでは incl.cut=0.8でfs/QCAと同じく閾値を0.8に設定している（森2017：221を参照）。

2) 複雑解を表示するには、次のようにする。

```
> CorpCS <- eqmcc (Corp, outcome="Effect", conditions=c ("Unimportant",
"Noninternal", "Ignorant", "Impact", "Recovery"), details=TRUE, show.case
=TRUE, use.tilde=TRUE, incl.cut=0.8)
> CorpCS
```

Noninternal*Impact*Recovery*(~Unimportant*~Ignorant+Unimportant*Ignorant) ⇔ Effectが複雑解⁽⁵⁴⁾となる⁽⁵⁵⁾（解整合度0.873、解被覆度0.934）。

3) 簡潔解を表示するには、次のようにする。

```
> CorpPS <- eqmcc (Corp, outcome="Effect", conditions=c ("Unimportant",
"Noninternal", "Ignorant", "Impact", "Recovery"), include="?", details=TRUE,
show.case=TRUE, use.tilde=TRUE, incl.cut=0.8)
> CorpPS
```

これは、複雑解の eqmcc のカッコ内に include="?" を加えたものである。この結果、Noninternal*Recovery ⇔ Effect が簡潔解となる（解整合度0.822、解被覆度0.934）。

4) 中間解を表示するには、次のようにする。今回の場合、方向性予想は、5つの原因条件すべて存在が結果の発生に寄与すると設定している。

```
> CorpIS <- eqmcc (Corp, outcome="Effect", conditions=c ("Unimportant",
"Noninternal", "Ignorant", "Impact", "Recovery"), include="?", dir.exp="1,1,1,1,1",
details=TRUE, show.case=TRUE, use.tilde=TRUE, incl.cut=0.8)
> CorpIS
```

これは、簡潔解の eqmcc のカッコ内に方向性予想 dir.exp="1,1,1,1,1" を加えたものである。この結果、Noninternal*Impact*Recovery ⇔ Effect が中間解となる（解整合度0.873、解被覆度0.934）。

5) 簡潔解の単純化の仮定を確認するには、次のようにする。

```
> CorpPSSA
```

6) 中間解の容易な反事実的仮定と困難な反事実的仮定を確認するには、次のようにする。

```
> CorpIS$.sol
```

以下の7)以降は、ファジィ集合の分析特有のものである。2.1.2の必要条件の場合と同様、XYプロットにより、十分条件関係を視覚的に表現することができる。ここでは、中間解の十分条件関係を表示することにする。そのためにはまず

7) で、中間解の項 `Noninternal*Impact*Recovery` の集合を表す変数を作る。そして、8) と 9) で XY プロットを表示する。

7) 中間解の論理式を表す変数 `Sol` を作る。

```
> Sol <- fuzzyand (Corp$Noninternal, Corp$Impact, Corp$Recovery)
```

中間解の項 `Noninternal*Impact*Recovery` は `Noninternal` と `Impact` と `Recovery` が論理積でつながれているため `fuzzyand` を使用する⁽⁵⁶⁾。変数 `Corp` (データファイルが代入されている) の中にある変数であることを示すために、`Noninternal`、`Impact`、`Recovery` の前にそれぞれ `Corp$` を付けなければならない。

8) XY プロットを表示するためには、次のように入力する。

```
> XYplot (Sol, "Effect", Corp, relation = "suf")
```

最初に X 軸にとる変数として 7) で作成した変数を書き (この変数には引用符は付けないことに注意)、次に Y 軸にとる結果の集合の名前を引用符の中に書き、データファイルを代入した変数を三番目に書いている。その後の `relation` のところでは、十分条件の分析であるので、`relation = "suf"` と入力する。

X 軸に指定した集合が Y 軸に指定した集合の十分条件である場合、図中の点は対角線よりも左上にある。

9) XY プロット中の各点の事例がわかるようにするには、次のように入力すればよい。

```
> XYplot (Sol, "Effect", Corp, relation = "suf", clabels=seq (nrow (Corp)),  
jitter=TRUE, amount=0.02)
```

2.1.4. 結果の不存在の分析

結果の不存在の分析をするには、森 (2017: 224) と同様で、今回新たに使用した XY プロットに関するコマンドも含めてすべてのコマンド内の “Effect” の部分を “~Effect” に書き換えるだけでよい。ただし、2.1.3 の 4) の中間解の方向性

予想だけは `dir.exp="0,0,0,0,0"` にする必要がある（結果の存在に寄与すると予想されているので、結果の不存在には寄与しないからである）。

2.2 補足：その他の補足的・発展的な事項

`fs/QCA` の 1.2 と同じく補足的な事項について説明する。2.2.1 は 1.2.1 と同じくファジィ集合に特有の事柄だが、2.2.2 は 1.2.2 と同じくファジィ集合に限らずクリスプ集合でも問題になる事柄である。また 2.2.3 では、クリスプ集合 `QCA` での視覚的表現であるベン図の表示の仕方を説明する。

2.2.1 と 2.2.2 では、1.2 と同じ表 3 のデータファイル `State.csv` を用いる。次のコマンドでそのファイルを読み込み、`State` という変数に代入しておく（国名が記されている変数 `code` が各行の名前に設定される）。

```
> State <- read.csv("State.csv", row.names="code")
```

2.2.1 ファジィ集合帰属度への変換の別の方法（「直接法」）

関数（代表的なものはロジスティック関数）を利用してファジィ集合帰属度に変換する方法である直接法の、`R` の `QCA` パッケージでの行い方を説明する。ここでは 1.2.1 と同様に、鉄鋼生産量（表 3 では `steel`）という量的な変数を、「工業国である」という集合 `ind3` への帰属度に直接法で変換する事を考える。次のように入力する。

```
> ind3 <- calibrate(State$steel, thresholds = "i=1803, c=269, e=37")
```

このコマンドで `fs/QCA` の直接法と同じファジィ集合帰属度が出力される。実際、`ind3` を出力すると、`fs/QCA` の直接法で変換したファジィ集合帰属度である表 3 の `ind2` と同じ値となる⁽⁵⁷⁾。

`calibrate` のカッコの中は、最初にファジィ集合帰属度に変換する変数を書くが、今の場合、変数 `State`（データファイルが代入されている）の中にある変数であることを示すために、`steel` の前に `State$` を付けなければならない。`thresholds` の後の引用符の中は、`i` の後に完全帰属とそうでないものの境目の値⁽⁵⁸⁾、`c` の後に帰属寄りと非帰属寄りの境目の値（帰属度 0.5 に相当する値）、`e` の後に完全非帰

属とそうでないものの境目の値を書く。また、calibrate のカッコの中に logistic = FALSE と書くと、変換にロジスティック関数ではなく、線形の関数が使用される⁽⁵⁹⁾。

2.2.2 論理的残余の扱いについて発展的な事項

まず2.1.3の1)と同様にして、次のように真理表を作成する。

```
> StateTable <- truthTable (State, outcome = "~intl", conditions = c ("ind1", "uk", "latin", "eastern"),
complete = TRUE, show.case = TRUE, sort.by = c ("incl"), incl.cut = 0.8)

> StateTable
```

すると、下の表5のような真理表が作成できる⁽⁶⁰⁾。ただし、真理表の整合度の値 (incl) の大きさ順に元の真理表をソートしているので、一番左の行番号は1から順には並んでいない (森2017:228)。また、○や◆はわかりやすいように筆者が書き直したもので、元の出力では論理的残余を表す?となっている。○は不可能残余の行 (1.2.2.1参照)、◆は必要条件と矛盾する反事実的仮定が置かれる可能性のある行 (1.2.2.2参照) である。

表5：反対票を投じる国 (~intl) についての真理表 (Rで作成)

	IND1	UK	LATIN	EASTERN	OUT	n	incl
2	0	0	0	1	1	2	1.000
3	0	0	1	0	1	7	1.000
10	1	0	0	1	1	3	1.000
1	0	0	0	0	0	10	0.364
5	0	1	0	0	0	3	0.000
9	1	0	0	0	0	10	0.000
13	1	1	0	0	0	2	0.000
4	0	0	1	1	○	0	—
6	0	1	0	1	◆	0	—
7	0	1	1	0	◆	0	—
8	0	1	1	1	○◆	0	—
11	1	0	1	0	?	0	—
12	1	0	1	1	○	0	—
14	1	1	0	1	◆	0	—
15	1	1	1	0	◆	0	—
16	1	1	1	1	○◆	0	—

2.2.2.1 不可能残余についての反事実的仮定

1.2.2.1で見たように、論理的残余のうち、latinとeasternが両方1のものは、ラテンアメリカかつ東欧ということになるが、このような国は現実にはありえない。このような論理的残余の行を不可能残余と言う。しかし、簡潔解や中間解では不可能残余も含まれてしまう可能性があるため、それを防ぐ処置が必要である。

解に不可能残余が含まれないようにするには、次のように入力する。ここでは1.2.2.1と同様に中間解を求めることにする⁽⁶¹⁾。

```
> StateIS1 <- eqmcc (State, outcome=~intl", conditions=c ("ind1", "uk", "latin",
"eastern"), include="?", dir.exp="0,0,1,1", details=TRUE, show.case=TRUE, use.
tilde=TRUE, incl.cut=0.8, omit=c (4,8,12,16))
> StateIS1
```

これは通常の中間解のコマンドに、omitという部分を追加したものになっている。このomitは、真理表のうち指定した番号の行を含めずに解を求める、というものである。不可能残余の行は、表5で○の付いている、行番号4、8、12、16である。行番号を1つだけ指定するときはomit=4のように数字をそのまま書けばよいが、複数の行番号を指定するときはomit=c (4,8,12,6) のようにc () でくくる必要がある⁽⁶²⁾。

このように入力することで、不可能残余を除いた中間解として $\sim uk * \sim latin * eastern + \sim ind1 * \sim uk * latin * eastern \rightarrow \sim intl$ が導かれる。それに対して、このような処置をせずに普通に中間解を求めた場合は $\sim uk * eastern + \sim ind1 * \sim uk * latin \rightarrow \sim intl$ となり、論理式が異なることがわかる。これらは1.2.2.1と同じ結果になっている。

2.2.2.2 必要条件と矛盾する反事実的仮定

State.csvについて2.1.2と同様のコマンドで必要条件の分析をすると、 $\sim uk$ が整合度1で、これのみが閾値0.9を超えるので、 $\sim uk$ が $\sim intl$ の必要条件、つまり $\sim uk \leftarrow \sim intl$ だと判断される。このとき、1.2.2.2で述べたように対偶 $uk \rightarrow intl$ も成り立つ必要があるため、真理表の論理的残余で uk が1のものは、 $intl$ が1 ($\sim intl$ の場合は0) でなければならない。簡潔解や中間解において、そのままではこれらの

行が解に含まれてしまう可能性がある。

解において必要条件と矛盾する反事実的仮定を防ぐには、次のように入力する。ここでは1.2.2.2と同様に簡潔解を求めることにする。

```
> StatePS2 <- eqmcc (State, outcome = "~intl", conditions = c ("ind1", "uk",
"latin", "eastern"), include = "?", details = TRUE, show.case = TRUE, use.tilde =
TRUE, incl.cut = 0.8, omit = c (6,7,8,14,15,16))

> StatePS2
```

必要条件と矛盾する反事実的仮定の置かれる行は表5で◆の付いている、行番号6、7、8、14、15、16である。このように入力することで、必要条件と矛盾する反事実的仮定を防ぐ処置をした簡潔解として $\sim uk * eastern + \sim uk * latin \rightarrow \sim intl$ が導かれる。それに対して、このような処置をせずに普通に簡潔解を求めた場合は $latin + eastern \rightarrow \sim intl$ となり、論理式が異なることがわかる。これらは1.2.2.2と同じ結果になっている。

さらに、2.2.2.1の不可能残余についての処置と、この必要条件と矛盾する反事実的仮定を防ぐ処置の両方を行うには次のようにする。ここでは1.2.2.2と同様に中間解を求めることにする。

```
> StateIS3 <- eqmcc (State, outcome = "~intl", conditions = c ("ind1", "uk",
"latin", "eastern"), include = "?", dir.exp = "0,0,1,1", details = TRUE, show.case =
TRUE, use.tilde = TRUE, incl.cut = 0.8, omit = c (4,6,7,8,12,14,15,16))

> StateIS3
```

これによる中間解は $\sim uk * \sim latin * eastern + \sim ind1 * \sim uk * latin * \sim eastern \rightarrow \sim intl$ となる。

2.2.2.3 SetMethodsパッケージでの方法

SetMethods パッケージを使用して、2.2.2.1や2.2.2.2と同様の分析を行うこともできる。以下の分析を行う前に、QCA パッケージのインストールと起動（森2017：231）と同様にして、SetMethods パッケージのインストールと起動を行っておく。

資 料

1) 分析のために真理表を代入した変数が必要であるので、まず2.2.2と同様に入力して、真理表を代入した変数 StateTable を作る。

2) 2.2.2.1と同様の不可能残余についての処置をした中間解を求めるには、次のように入力する。

```
> StateTableA <- esa (StateTable, imposs_LR="LATIN*EASTERN")
> StateISA <- eqmcc (StateTableA, outcome="~intl", conditions=c ("ind1",
"uk", "latin", "eastern"), include="?", dir.exp="0,0,1,1", details=TRUE,
show.case=TRUE, use.tilde=TRUE, incl.cut=0.8)
> StateISA
```

1つ目の入力促進記号>のところで、不可能残余の行には結果の欄に0が入力されている真理表が作成され、それが StateTableA という変数に代入されている⁽⁶³⁾。imposs_LRのところには、不可能残余となる条件の論理式を書く（ただし、ここでは集合の名前をすべて大文字で書かないと機能しないので注意する）。今の場合、ラテンアメリカかつ東欧（latin*eastern）という国が現実にはありえず不可能残余であるため、LATIN*EASTERNと書く。

2つ目の入力促進記号>のところで、中間解を求めてそれを StateISA という変数に代入している。解を求めるコマンド eqmcc は、ファイルを代入した変数だけでなく真理表を代入した変数でも同じように機能する（森2017：227も参照）。そのため、StateTableA という、1つ目の入力促進記号>のところで作った、真理表を代入した変数をここで使用している。

これにより、不可能残余についての処置をした中間解として、2.2.2.1と同様の $\sim uk * \sim latin * eastern + \sim ind1 * \sim uk * latin * \sim eastern \rightarrow \sim intl$ が導かれる。

3) 必要条件と矛盾する反事実的仮定を防ぐ処置をした簡潔解を求めるには、次のように入力する。

```
> StateTableB <- esa (StateTable, nec_cond = "~UK")
> StatePSB <- eqmcc (StateTableB, outcome = "~intl", conditions = c ("ind1", "uk",
"latin", "eastern"), include = "?", details = TRUE, show.case = TRUE, use.tilde =
TRUE, incl.cut = 0.8)
> StatePSB
```

1 つ目の入力促進記号>のところで、必要条件と矛盾する反事実的仮定が置かれる可能性のある論理的残余の行には結果の欄に 0 が入力されている真理表が作成され、それが StateTableB という変数に代入されている。nec_cond のところには、必要条件となっている集合の名前を書く（ただし、ここでは集合の名前をすべて大文字で書かないと機能しないので注意する）。今の場合 ~uk が必要条件なので、~UK と書く。

2 つ目の入力促進記号>のところで、中間解を求めている。解を求めるコマンド eqmcc はファイルを代入した変数だけでなく真理表を代入した変数でも同じように機能する。そのため、StateTableB という、1 つ目の入力促進記号>のところで作った、真理表を代入した変数をここで使用している。

これにより、必要条件と矛盾する反事実的仮定を防ぐ処置をした簡潔解として、2.2.2.2 と同様の ~uk*eastern + ~uk*latin → ~intl が導かれる⁽⁶⁴⁾。

4) さらに、不可能残余についての処置と、必要条件と矛盾する反事実的仮定を防ぐ処置の両方を行った中間解を求めるには、次のように入力する。

```
> StateTableC <- esa (StateTable, nec_cond = "~UK", imposs_LR = "LATIN*EASTERN")
> StateISC <- eqmcc (StateTableC, outcome = "~intl", conditions = c ("ind1", "uk", "latin",
"eastern"), include = "?", dir.exp = "0,0,1,1", details = TRUE, show.case = TRUE, use.tilde
= TRUE, incl.cut = 0.8)
> StateISC
```

これにより、2.2.2.2 と同様の中間解である ~uk*~latin*eastern + ind1*~uk*latin*~

eastern → ~intlが導かれる。

2.2.3 ベン図

ファジィ集合ではXYプロットを使用して視覚的な表現をしたが、クリスプ集合（通常の集合）で視覚的な表現をしたい場合にはベン図を用いることが多い。しかし、fs/QCAにはベン図を描く機能は備わっていない。それに対して、Rではvennパッケージを使用することでベン図を描くことができる⁽⁶⁵⁾。以下の分析を行う前に、QCAパッケージのインストールと起動（森2017：231）と同様にして、vennパッケージのインストールと起動を行っておく。

ここでは、クリスプ集合の分析の例として、前回の森（2017）で使用した、下の表6のデータを使用する。これは、米国連邦最高裁判所の「弁護人の援助を受ける権利」に関する裁判例のデータ（森2017：248）である。ファイルを読み込み、それを変数に代入するため次のように入力しておく。

```
> Betts<-read.csv("Betts.csv", row.names="Case")
```

表6：米国の「弁護人の援助を受ける権利」に関する裁判例のデータの一部（森2016より）

Case	Counsel	Capital	Youth	Illiteracy	Between
1	1	1	1	1	1
2	0	1	0	0	0
3	1	0	0	1	1
4	0	0	0	0	1

注：Case：事例番号、Counsel：弁護人を付すべきと裁判所が判断した、Capital：死刑事件である、Youth：被告人が若年である、Illiteracy：被告人に文字が読めない等の事情がある、Between：罪状認否手続から公判開始までの間弁護人が付いていなかった

- 1）ベン図を作るには、その前にまず真理表を代入した変数が必要であるので、以下のようにそれを作成する（森2017：229と同じコマンドである）。

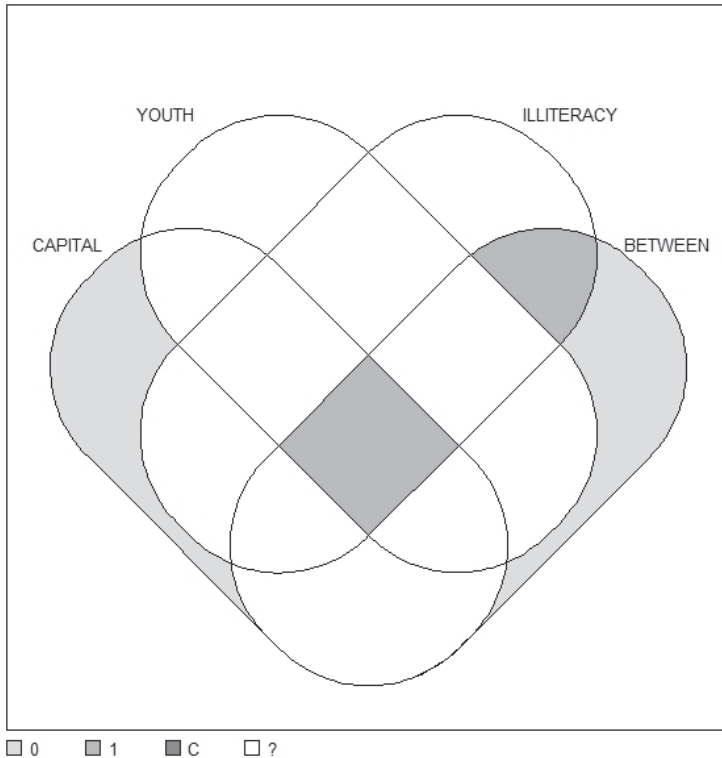
```
> BettsTable<-truthTable(Betts, outcome="Counsel", conditions=c("Capital",
"Youth", "Illiteracy", "Between"), complete=TRUE, show.case=TRUE, sort.
by="incl")
```

2) 次に次のように入力すれば、ベン図が作成できる。

```
> venn (BettsTable)
```

これは venn の後のカッコの中に 1) で作成した真理表の変数が入っている。

これにより、下のようなベン図が表示される。



ベン図の各領域は、ベン図の下の凡例にあるように、真理表の各行の結果の欄に対応して色分けされている。すなわち、凡例の 0 や 1 は真理表の当該行の結果が 0 や 1 であることを表し、C は真理表の当該行の結果に矛盾がある（つまり 0 の事例もあれば 1 の事例もある）ことを示し、? は真理表の当該行が論理的残余である（つまり該当する事例がない）ことを示している。ベン図を見ると、今回の例の場合、例えば Capital の集合の内側で他の集合の外側

(Capital*~Youth*~Illiteracy*~Between) は結果 Counsel が 0 であることがわかる。
また、0、1、? の領域があり、C の領域は存在していないことがわかる。

参考文献

- 石田淳・齋藤圭介監訳 (2016) 『質的比較分析 (QCA) と関連手法入門』 晃洋書房
(B. Rihoux & C. C. Ragin (2008) *Configurational Comparative Methods: Qualitative Comparative Analysis (QCA) and Related Techniques*, Sage Publications, Inc.)
- 田村正紀 (2015) 『経営事例の質的比較分析：スモールデータで因果を探る』 白桃書房.
- 舟尾暢男 (2016) 『The R Tips 第3版：データ解析環境Rの基本技・グラフィックス活用集』 オーム社.
- 森大輔訳 (2010a) 「Fuzzy - Set/Qualitative Comparative Analysis ユーザーガイド」
(<http://park18.wakwak.com/~mdai/qca/software.html>) (C. C. Ragin 2008 “User’s Guide to Fuzzy-Set/Qualitative Comparative Analysis 2.0.” Tucson, Arizona : Department of Sociology, University of Arizona. <http://www.u.arizona.edu/~cragin/fsQCA/software/fsQCAJapanese.pdf>)
- 森大輔訳 (2010b) 「Tosmana ユーザーマニュアル」 (<http://park18.wakwak.com/~mdai/qca/software/TosmanaJapanese.pdf>) (L.Cronqvist 2007 “Tosmana User Manual.” http://www.tosmana.net/tosmana_manual1_3beta.pdf)
- 森大輔 (2012) 「国家責任法の経済学的分析 (5・完)」 国家学会雑誌125 (11・12)、60-120.
- 森大輔 (2014) 「国家間の紛争解決での裁判の有効性と限界—複数事例の質的比較分析」 安川文朗・石原明子編『現代社会と紛争解決学—学際的理論と応用』 ナカニシヤ出版、70-87.
- 森大輔 (2016) 「判例研究への質的比較分析 (QCA) の応用の可能性：米国の弁護士人依頼権に関する判例の分析を例に」 熊本法学136、318-262.
- 森大輔 (2017) 「質的比較分析 (QCA) のソフトの使用法—fs/QCAとRのQCA・SetMethods パッケージ (1)」 熊本法学140、250-209.
- 弥永真生 (2009) 『法律学習マニュアル』 有斐閣.

- Ragin, C. C. (2000) *Fuzzy-Set Social Science*. University of Chicago Press.
- Ragin, C. C. (2008) *Redesigning Social Inquiry: Fuzzy Sets and Beyond*. University of Chicago Press.
- Schneider, C.Q. & C. Wagemann (2012) *Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis*. Cambridge University Press.
- Thiem, A. (2014) “Membership Function Sensitivity of Descriptive Statistics in Fuzzy-Set Relations.” *International Journal of Social Research Methodology* 17 (6), 625-642.
- Thiem, A. & A. Dusa (2012) *Qualitative Comparative Analysis with R: A User's Guide*. Springer.
- United Nations (1949) *International Capital Movements during the Inter-War Period*. United Nations (楊井克巳・中西直行訳『国際投資論』日本評論社、1970年)
- Verkuilen, J. (2005) “Assigning Membership in a Fuzzy Set Analysis.” *Sociological Methods and Research* 33, 462-496.

注

- (1) 他に 2 値でなく多値の変数を利用する多値 QCA (multi-value QCA) という方法があり、Tosmana などこれを実行できるソフトウェアもある (fs/QCA では直接は実行できない)。しかし、これは 2 値の QCA に完全に置き換えられ、2 値の QCA で十分であることが Schneider & Wagemann (2012:258) によって示されている。
- (2) ファジィ集合 QCA の邦語での解説として森訳 (2010a : 78-96)、田村 (2015)、石田・齋藤監訳 (2016 : 108-146)。
- (3) 前回の森 (2017) と同様に fs/QCA は Windows 版のバージョン 3.0、R は Windows 版のバージョン 3.4.0、QCA パッケージはバージョン 2.6、SetMethods パッケージはバージョン 2.1 を使用している。それぞれのインストールや起動の仕方、データファイルの読み込み方などは森 (2017 : 249)、マニュアル等の所在は森 (2017 : 215) を参照。また、venn パッケージはバージョン 1.2 を

使用しており、マニュアル（英語）などは

<https://CRAN.R-project.org/package=venn>にある（2017年10月1日確認）。

- (4) この点は、QCA 自体の提唱者でもあるチャールズ・レイガン（Charles C. Ragin）の功績である。詳しくは注12を参照。
- (5) 弥永（2009：202）では、この表は必ずしも確立した考え方に基づいたものではなく、1つの例を挙げたに過ぎないと断っている。本稿でも同様である。以下のファジィ集合QCAによる分析も、会社法の分析に新しい知見を付け加える意図はなく、あくまでソフトウェアの使い方の例として挙げるのみである。
- (6) 直接法と呼ばれる、ロジスティック関数などを用いた変換方法については1.2.1で説明する。
- (7) 他に6値ファジィ集合帰属度（0，0.2，0.4，0.6，0.8，1）なども用いられる。値が細くなればなるほど、微妙な差異を表現できるが、他方で、ある事例がなぜ他の値ではなくその値だと考えられるのかを説明するのが難しくなる。
- (8) これは逆に、△という記号が、帰属寄りか非帰属寄りか明らかにしておらず曖昧であることを浮き彫りにしているとも言えよう。
- (9) 変換を行う際には、なぜその値にしたか他者にわかるように理由を書いて、分析の透明性を確保することが推奨される（Schneider & Wagemann 2012:35）。ある意味で、分析過程の中で、この変換の部分が最も難しい部分である。
- (10) もちろん、ここでの議論はあくまで例示であり、深い法学的知識に基づくものではない。0.67として分析することも、その根拠を明示すれば十分にありうる。実際、本文の分析でも、後の注27でこれを0.67と変えることを考えている。
- (11) △は例えば、利益相反取引について特に第三者との取引であれば無効にする
と第三者への影響はそれなりにあると考えて0.67にするなど、本来は1つ1
つに理由が必要で、それを記すべきである。ここではあくまで例示なのでそ
れらを省略し、すべてを0.67としている。したがって、他の値（0.33など）で

質的比較分析 (QCA) のソフトの使用方法

分析することも、その根拠を明示すれば十分にありうる。

- (12) 実は、このことは非常に画期的である。Ragin (2000) において QCA にファジィ集合が導入された際は、必ずしもクリスプ集合と分析方法は共通ではなかった。しかしその後 Ragin (2008) において、クリスプ集合 QCA とファジィ集合 QCA で共通の分析方法が確立され、fs/QCA でも共通の方法で分析ができるようになった。現在では Schneider & Wagemann (2012) に見られるように、両者は完全に統合されている。ソフトウェア fs/QCA においてもこの流れが反映されている。バージョン 2.0 までの段階では、クリスプ集合とファジィ集合で別々のメニューを選択するようになっていた（ただし実際はどちらを選んでも分析できた）が、バージョン 3.0 ではそれが廃止され両者は完全に統合された。
- (13) 前回の森 (2017) で扱ったようなクリスプ集合（通常の集合）の分析で視覚的な表現をしたい場合には、ベン図を用いることが多い。しかし、fs/QCA にはベン図を描く機能は備わっていない。ベン図を描きたい場合は Tosmana か、R の `venn` パッケージを用いる。Tosmana の使い方については森沢 (2010b) 参照。R の `venn` パッケージの使い方については 2.2.3 を参照。
- (14) 右下は対角線上も含む。つまり、すべての事例で、X 軸に指定した集合への帰属度が Y 軸に指定した集合への帰属度以上になる（事例番号を i 、X 軸に指定した集合への事例番号 i の帰属度を X_i 、Y 軸に指定した集合への事例番号 i の帰属度を Y_i とすると、すべての i で $X_i \geq Y_i$ となる）。このことは、クリスプ集合で X が Y の必要条件である場合、集合 X の大きさが集合 Y 以上である（ベン図で言えば X の円の中に Y の円がすっぽりと入る）ことを思い出すと、直感的に理解しやすい。
- (15) 左上は対角線上を含まない。
- (16) 整合度が閾値（例えば 0.9）を超えれば必要条件であると判断されることは本文の 6) で説明したとおりであるが、仮にそうやって必要条件であると判断するとしても、このように矛盾する事例が具体的にどれであるか図で確認し、場合によっては事例に立ち戻って、なぜそれらの事例が矛盾するか、条件を

解釈し直したり追加したりして矛盾を解消する必要はないかなどを考えることが、質的分析の側面を持つQCAでは重要である。

- (17) 後に1.1.3で触れるように、十分条件の分析では逆に、Consistency $X \leq Y$ が整合度で Consistency $X \geq Y$ が被覆度である。これは以下のような理由からである。Consistency $X \leq Y$ は、すべての i で $X_i \leq Y_i$ という関係、つまり十分条件関係における整合度ということになる（注24参照）。Consistency $X \geq Y$ は、すべての i で $X_i \geq Y_i$ という関係、つまり必要条件関係における整合度ということになる（注14参照）。さらに実は、必要条件の整合度と十分条件の被覆度は計算式が同じであり（どちらも $\sum \min(X_i, Y_i) / \sum Y_i$ ）、十分条件の整合度と必要条件の被覆度は計算式が同じである（どちらも $\sum \min(X_i, Y_i) / \sum X_i$ ）（Schneider & Wagemann 2012:119-150参照）。そのため、必要条件関係では Consistency $X \geq Y$ が整合度で Consistency $X \leq Y$ が被覆度、十分条件関係では Consistency $X \leq Y$ が整合度で Consistency $X \geq Y$ が被覆度ということになる。
- (18) この図は背景に透過情報を含んでいるので、透過情報を扱える描画ソフトに貼り付ける必要がある。Windows に最初から備わっている Paint では透過情報を扱えないので、図全体が黒くなってしまう。透過情報が扱えるフリーの描画ソフトとしては、例えば LibreOffice の Draw（OpenOffice の Draw もほぼ同様）がある。
- (19) その際は真理表の raw consist. と書かれている所をクリックすると raw consist. の値の大きさ順にソートされるのでやりやすい。
- (20) 最初の形は $\sim \text{Unimportant} * \text{Noninternal} * \sim \text{Ignorant} * \text{Impact} * \text{Recovery} + \text{Unimportant} * \text{Noninternal} * \text{Ignorant} * \text{Impact} * \text{Recovery} \rightarrow \text{Effect}$ であるが、通常の数式と同様に分配法則が成り立つので、カッコでくくり、本文の形にできる。
- (21) どの解も、結果 Effect に帰属寄りである全事例（事例 Yuri、Shasai、Kyogyo、Shinkabu）がカバーされているので、必要十分条件と言ってもよく、 \Leftrightarrow で表すこともできる。ただし、fs/QCA では解が十分条件にとどまるか、それとも必要十分条件になるかは出力されないなので、もし区別したい場合には自分で

質的比較分析 (QCA) のソフトの使用方法

- 判断する必要がある。RのQCAパッケージでは区別して出力される。
- (22) なお、プルダウンをクリックすると既存の変数も選択できるが、既存の変数を選択するとその変数が上書きされることになる。また、変数の名前をつける際には、2～8文字の半角英数文字を使う必要がある。これよりも長い名前をつけようとするソフトが異常終了することがある。また、変数の名前に空白、ダッシュ、句読点を含んではならない。
- (23) もしも解の項に論理和が入っている場合には fuzzyor を使う。例えば解の項が $A+B$ の場合、fuzzyor (A,B) となる。 $\sim C$ の場合は fuzzynot を使って fuzzynot (C) とする。さらに $A*\sim C+B*D$ の場合、fuzzyor (fuzzyand (A, fuzzynot(C)), fuzzyand (B,D)) となる。
- (24) 左上は対角線上も含む。つまり、すべての事例で、Y軸に指定した集合への帰属度がX軸に指定した集合への帰属度以上になる（事例番号を i 、X軸に指定した集合への事例番号 i の帰属度を X_i 、Y軸に指定した集合への事例番号 i の帰属度を Y_i とすると、すべての i で $X_i \leq Y_i$ となる）。このことは、クリスプ集合でXがYの十分条件である場合、集合Yの大きさが集合X以上である（ベン図で言えばYの円の中にXの円がすっぽりと入る）ことを思い出すと、直感的に理解しやすい。
- (25) 完全な必要十分条件の場合には、事例はすべて右上がりの対角線上にある。
- (26) 右下は対角線上を含まない。
- (27) この図を見ると、例えば Rieki や Tagaku の Effect への帰属度を0.33から0.67に変更すれば、これらの事例の点は対角線より左上に移動し、十分条件関係に矛盾する事例はなくなることがわかる。それに合わせて Jigyo も Effect への帰属度を0.33から0.67に変更してもよい。ただし、帰属度を変える際には、そうすれば矛盾がなくなるからというだけでなく、そうする確固たる理由が必要であり、それを文中で明らかにすることが推奨される。
- (28) 1.1.2で見たように、必要条件の分析では逆に、Consistency $X \leq Y$ が被覆度で Consistency $X \geq Y$ が整合度である。
- (29) ド・モルガンの法則で導かれる式と結果の不存在の解の式が必ずしも一致し

- ないことについては、森（2017：213）の注30を参照。
- (30) なぜロジスティック関数を使用するかについて、それほど確固たる理論的根拠があるわけではない。実際、2.2.1で見るようにRではロジスティック関数以外の使用も可能である。また、様々な関数での変換を論じたものとして Thiem（2014）参照。
- (31) 注22と同様の点にここでも注意する。
- (32) 正確には帰属度0.95に相当する値となっている。
- (33) 正確には帰属度0.05に相当する値となっている。
- (34) キーボードから入力してもよいし一番左の欄から変数 steel を選択してもよい。
- (35) 例えば、帰属度0.33だったものが帰属度0.67に変わってしまうような、0.5をまたぐ変化を指している。
- (36) ただし、絶対に大きな影響がないわけではないので、もし心配な場合には、帰属度への変換の方法を実際にいろいろ変えて、結果が変わるか試してみるという頑健性（robustness）の分析を行ってみるとよい。頑健性の分析については Schneider & Wagemann（2012:284-295）参照。
- (37) ind1 の整合度は0.208125で被覆度は0.243421、~ind1 の整合度は0.791875で被覆度は0.543310、ind2 の整合度は0.190000で被覆度は0.222222、~ind2 の整合度は0.810000で被覆度は0.555746となる。
- (38) 例えば中間解は、ind1 を使うと ~uk*eastern+~ind1*~uk*latin → ~intl で解整合度 1 で解被覆度0.729375であるのに対し、ind2 を使うと ~uk*eastern+~ind2*~uk*latin → ~intl で解整合度 1 で解被覆度0.7325と、解被覆度が多少変わる程度の違いしかない。
- (39) この4つの原因条件を考えているのは、以下の理由からである。すなわち、この会議で反対票を投じた国は、ラテンアメリカや東欧の、あまり発展していない国が多かったと言われていたこと、ただし英連邦の国は特に当時まだ英国の影響が強く英国に追従して賛成票を投じていたと考えられることである。
- (40) この場合、真理表の行を並び替えないと下の方に結果の欄に 0 が入った不可

質的比較分析 (QCA) のソフトの使用方法

能残余の行があるので、Delete current row to last rowで最後の行まで一括して削除するわけにはいかない。そのため、①Editの一番上のDelete current rowを使って一行ずつ削除するか、②真理表の一番上の~intlと書かれた部分をクリックすると真理表の行が~intlの値の順番でソートされるので、空欄の行が下に来るようにソートした上でDelete current row to last rowで一括して削除することになる。

- (41) 方向性予想は、ind1とukは欠如が結果に寄与し、latinとeasternは存在が結果に寄与するとする。このように予想した理由は、注39のような状況のためである。
- (42) どちらの場合も、簡潔解の解整合度は1で解被覆度は0.75、中間解の解整合度は1で解被覆度は0.729375となっている。
- (43) 森（2014：80）でも分析の際にこの点を考慮している。
- (44) この式 $\text{latin} + \text{eastern} \rightarrow \sim \text{intl}$ には、必要条件である $\sim \text{uk}$ がないということから、必要条件と矛盾する反事実的仮定が含まれていることがわかる。一般に必要な条件となっている条件は、 $\sim \text{uk} * \text{latin} + \sim \text{uk} * \text{eastern} \rightarrow \sim \text{intl}$ のように、十分条件の式の左辺のすべての項に現れる。このことは、必要条件は結果を生じるのに必ずなければならない条件であるということを考えれば理解できる。
Schneider & Wagemann（2012:221-227）参照。
- (45) どちらの場合も、簡潔解の解整合度は1で解被覆度は0.75となっている。また、中間解はどちらの場合も $\sim \text{uk} * \sim \text{latin} * \text{eastern} + \sim \text{ind1} * \sim \text{uk} * \text{latin} * \sim \text{eastern} \rightarrow \sim \text{intl}$ で変わらない。ただし、これは今回の場合たまたま変わらなかっただけで、中間解も両方で異なる可能性がある。Schneider & Wagemann（2012:203）参照。
- (46) 簡潔解は $\sim \text{uk} * \text{latin} * \sim \text{eastern} + \sim \text{uk} * \sim \text{latin} * \text{eastern} \rightarrow \sim \text{intl}$ となる。また、中間解の解整合度は1で解被覆度は0.729375、簡潔解の解整合度は1で解被覆度は0.75となる。
- (47) 他にも例えば、結果の存在（今の例ではintl）と不存在（今の例では~intl）を両方分析する場合、結果の存在の分析で解に含めた論理的残余は、結果の不

存在の分析で解に含めないようにする必要がある（そうでなければ矛盾する反事実的仮定を置いていることになってしまう）といった注意点などがある。

Schneider & Wagemann (2012:203) 参照。この場合、fs/QCA ではどの論理的残余が解に含まれるか出力しないので、R などを使って確認する必要がある。

- (48) メニューの「ファイル」をクリックし、「ディレクトリの変更」を選択、フォルダの選択画面で Corporate.csv の存在するフォルダを選択する。
- (49) fs/QCA と違い、あらかじめ設定した必要条件の整合度の閾値（今の例の場合は 0.9）を超えるものは、論理和や論理積を含めてすべて出力されるが、ここではそれらは除いている。
- (50) この他に、R のグラフ描画で一般的に使える様々なグラフィックスパラメーターがこの場合も使えるため、かなり自由に図の見た目を変えることができる。例えば、XY プロット中の点の色や形を変えることもできるし、軸の名前を変えたりすることも可能である。その方法については、舟尾 (2016) などの R 全般についての解説書を参照。
- (51) それでも事例の名前を使いたい場合は、`clabels=rownames(Corp)` とすればよい。`rownames()` は行の名前を出力するコマンドだが、今の場合ファイルの読み込みのところで `Corp<-read.csv("Corporate.csv", row.names="Case")` としており行の名前を Case の変数で設定しているので、`rownames()` で事例の名前を指定することになる。
- (52) その横にある Relevance:0.683 は、Relevance of Necessity の値を表す。これは、Schneider & Wagemann (2012:236) において必要条件の被覆度の代わりに提案された指標であり、2.1.2 の 1) の出力では、RoN という列に現れる。通常は被覆度を見れば十分である。森 (2017:210) の注 47 を参照。
- (53) 本当に位置が異なっている場合もあるので、jitter を指定していない XY プロットも出力して、jitter でずれているだけなのか否かを確認しておいた方がよい。
- (54) 注 20 と同様、カッコでくくっている。
- (55) 十分条件にとどまらず必要十分条件と言えるので左辺と右辺が ⇔ で結ばれて出力される。必要十分条件になる理由は注 21 参照。

質的比較分析 (QCA) のソフトの使用方法

- (56) もしも解の項に論理和が入っている場合には fuzzyor を使う。例えば解の項が $A+B$ の場合、fuzzyor (A,B) となる。これらは注23の fs/QCA と同じコマンドである。しかし $\sim C$ の場合は異なり、R では 1-C とする。さらに $A*\sim C+B*D$ の場合、fuzzyor (fuzzyand (A,1-C), fuzzyand (B,D)) となる。
- (57) 出力するには、ind3 と入力してエンターキーを押せばよい。表 3 の ind2 は小数第二位までの表示だがこれに合わせるためには、round (ind3, 2) と入力してエンターキーを押せばよい。
- (58) デフォルトでは帰属度0.95に相当する値となっており、これは fs/QCA と同じ値である。calibrate (State\$steel, thresholds =“i=1803, c=269, e=37”, idm = 0.99) のように idm を付け加えれば、この値を変更することができる (idm = 0.99で0.95から0.99に変更される)。
- (59) 他にも変換の関数について様々な設定が可能である。詳しくはThiem & Dusa (2012:54-62) を参照。
- (60) 出力ではinclの右側にPRIやcaseが表示されるが、ここでは省略している。
- (61) 方向性予想は注41と同様、ind1とukは欠如が結果に寄与し、latinとeasternは存在が結果に寄与するとする。そのため、dir.exp=“0,0,1,1”と設定している。
- (62) c () は、Rで数値の組 (ベクトル) を表すときに使用する。
- (63) 真理表の不可能残余の行の結果の欄に正しく 0 が入力されているか否かを確認するには、StateTableAと入力してエンターキーを押せばよい。
- (64) 実はSetMethodsパッケージのバージョン2.1ではバグのため、正しい結果とならない。開発者のIoana-Elena Oanaに問い合わせをしたところ、2017年12月に修正予定であるという返答を得た。
- (65) vennパッケージの場合、ベン図を描くことができるのは、原因条件が7個までである (Tosmanaの場合は5個まで)。もっとも、条件を多くしすぎるとQCA自体行うことが難しくなるので、このことはあまり問題とならないと思われる。QCAでは、40個までくらいの中規模の事例数の場合は、原因条件の最大数は7個程度までが望ましいとされている。石田・齋藤監訳 (2016: 42) 参照。