



HOKKAIDO UNIVERSITY

Title	Rを利用したCase 1 Best-Worst Scalingの実施手順
Author(s)	合崎, 英男
Citation	北海道大学農経論叢, 71, 59-71
Issue Date	2017-03-31
Doc URL	https://hdl.handle.net/2115/65921
Type	departmental bulletin paper
File Information	59-71.pdf



R を利用した Case 1 Best-Worst Scaling の実施手順

合 崎 英 男

Implementing Case 1 Best-Worst Scaling in R

Hideo AIZAKI

Summary

The application of Case 1 Best-Worst Scaling (BWS) to empirical studies consists of the following steps: (1) establishing the objective of applying BWS, (2) writing a scenario where respondents evaluate items, (3) creating choice sets, (4) preparing a questionnaire survey, (5) implementing the survey, (6) preparing the data set for analysis of the responses, and (7) measuring the relative importance of items. This paper provides Japanese R beginners with an explanation of how the support.BWS and other packages in R are used for implementing this series of steps via an illustration of consumers' valuation of fruits.

1. はじめに

Case 1 あるいは Object Case Best-Worst Scaling (以下, BWS) – Maximum Difference (MaxDiff) Scaling と呼ばれる – は, 複数の項目に対する人々の選好 (相対評価) を測定するための質問調査法である。評価対象とする項目リストを定めた上で, そこから複数の部分集合 (異なる項目の組み合わせ) を作成する。各部分集合が 1 つの設定で提示する選択肢集合となる。回答者は, 各部分集合から最も重要な (ベスト) 項目と最も重要ではない (ワースト) 項目を選択するよう求められる。回答結果を分析することで, 各項目の相対評価を算出する。1980年代後半に Jordan J. Louviere によって開発された手法であり, Finn and Louviere (1992) が初出の適用論文である (Flynn 2010)。2000年代に入って理論的基礎の確立と適用事例の拡大が図られ, 近年には関連する 2 つの手法 (Case 2 and Case 3 Best-Worst Scaling) とともに体系的な書籍 (Louviere et al. 2015) が刊行された。医療経済学をはじめとして幅広い学術分野で活用されており, 農業経営・経済学に関連する適用事例も数多くある。たとえば, 食品価値の消費者評価 (Lusk and Briggeman 2009), 牛肉生産における食料自給率向上の消費者評価 (澤田ら2010), 酪農

経営者による酪農政策の評価 (Wolf and Tonsor 2013), 農業経営者による温暖化緩和技術の評価 (Dumbrell et al. 2016), 農業用水の多目的利用に対する地域住民の評価 (西原ら2016), 健康増進の観点からみた農作業の市民評価 (Aizaki et al. 2016) などがある。

適用事例の広がりやの要因としては, 離散選択実験と比べて多数の項目の相対評価を効率的に行えること, 評定型評価の弱点を緩和できることなどの方法論上の優位性ととともに, 調査設計と回答データ分析が離散選択実験より容易なことも指摘できよう。後者の特徴の背景には, 項目の相対評価には必ずしも計量経済分析を必要としないことや利用可能なソフトウェア (e.g., Kuhfeld 2010 ; Sawtooth Software 2016) の存在もある。

BWS の実行を主目的とした R (R Core Team 2016) 用パッケージとしては, 2013年5月に The Comprehensive R Archive Network (CRAN) で公開された support.BWS パッケージ (Aizaki 2015) がある。選択肢集合の作成と離散選択モデルの推定以外の主な作業を R で行えるようにする関数と例題データセットを含んでおり, 実証研究での活用実績 (Aizaki et al. 2016 ; 西原ら2016) もある。ただし, support.BWS パッケージの使用方法に関する一般に入手可能な資料は英文 (Aizaki et al.

2014)に限られていたため、日本語環境での利便性は低いものであった。そのような状況を改善する目的で、本稿は執筆された。

以下、BWSの適用手順を概観した上で、BWSの選択肢集合の設計から項目の相対評価の計算までの作業をRで行う手順を解説する。ただし、紙幅に限りがあることから、手法の詳細については既存のBWS関連文献(e.g., Auger et al. 2007; Louviere et al. 2013; Louviere et al. 2015)を、Rでの実行方法の詳しい情報は support.BWS パッケージ(バージョン0.1-3)のヘルプや Aizaki et al.(2014)の第4章を参照されたい。本稿の読者は、Rの基本操作とダミー変数を利用した線形回帰分析に関する知識は有していることを前提としている。Rや回帰分析の基礎については多くの資料が存在するので、必要に応じて自身に適したものを参照されたい。

2. BWSの適用手順

本稿では、BWSの適用手順を7つのステップにわけて考える。

ステップ1と2は、それぞれBWSの適用目的の決定と評価シナリオの作成である。評価シナリオの作成には、評価対象項目の決定のみならず、調査票に記載するBWSの設問文や提示項目を回答者が評価するときの基準の検討などの諸作業も含まれる。これらの作業の詳細については、既存のBWS関連文献や表明選好法のマニュアル類を参照されたい。ここでは2点のみ指摘したい。1つは回答者による項目の評価基準についてである。「重要」「好き」「買いたい」といったよく利用されている基準に加え、「信頼できる」「実用的である」「望ましい」などさまざまな基準を使うことができる。重要なことは、回答者からみて適切な基準となっていることである。もう1つ指摘しておきたいことは、評価対象項目の決定についてである。評価対象項目の数は、後述する選択肢集合の設計方法にも影響を受けるため、研究目的からみて必要な項目を全て設定できるとは限らない。したがって、ステップ2の一環として評価対象項目を決定した後に、それを変更しなければならない事態に遭遇することもあり得る。

ステップ3と4は、それぞれ全選択肢集合の設

計と調査票の作成である。1つの選択肢集合(choice set)は、回答者にとって利用可能な選択肢の集まりであり、BWSでは3つあるいはそれ以上の項目(item)から構成されている。BWSの設問では、回答者は1つの選択肢集合から、指定された評価基準からみて自身にとってベストな項目とワーストな項目を1つずつ選ぶよう求められる。したがって、1つの選択肢集合がBWSの1つの設問に対応する。全選択肢集合(choice sets)は、個々の選択肢集合をまとめたものである。たとえば、全選択肢集合が9つの選択肢集合を含むとすれば、BWSの全設問数は9問であることを意味する。

BWSの全選択肢集合を設計する方法は大きく2つにわけられる。1つは2水準直交主効果計画(Orthogonal Main-Effect Design)を利用する方法(Finn and Louviere 1992)、もう1つはつり合い型不完備計画(Balanced Incomplete Block Design: BIBD)を利用する方法(Auger et al. 2007)である。本稿では、多くの事例で活用されているBIBDを利用した方法を取り上げる。

BIBDは、一般に実験を設計する際に処置(treatment)をブロック(block)に割り付けるために利用する実験計画の一種である(BIBDを含めた実験計画法の文献については合崎(2015)を参照)。BIBDの特徴は、「処置数」「ブロックの大きさ」「ブロック数」「各処置の出現数」および「各処置ペアが同一ブロックで比較される回数」で捉えることができる。BWSの用語で言い換えれば、「処置」は「項目」、「ブロック」は「設問」に対応する。たとえば、Apple, Orange, Grapes, Banana, Peach, Melon, および Pear の7つの果物(項目)を評価するBWS設問を考える。この場合、項目数は7であることから、7つの異なる処置を取り扱えるBIBDが必要となる。その条件を満たすものの1つとして、7つの処置のうち異なる4つの処置を1つのブロックに配置し、ブロック数が7となるBIBDがある(図1)。ここで、Apple から Pear までの順に1から7の項目番号を付与するとしよう。すると、処置番号1から7を用いて記述されているBIBDは、処置番号と項目番号を対応させることで、項目名で記述されたBIBDに変換することができる。各ブロック

(1) つり合い型不完備計画の作成 (選択)

ブロック 番号	列1	列2	列3	列4
1	1	4	6	7
2	2	3	4	6
3	2	4	5	7
4	1	2	5	6
5	1	3	4	5
6	3	5	6	7
7	1	2	3	7

項目 番号	項目名
1	Apple
2	Orange
3	Grapes
4	Banana
5	Peach
6	Melon
7	Pear

(2) 処置番号を
項目名で置換

ブロック 番号	列1	列2	列3	列4
1	Apple	Banana	Melon	Pear
2	Orange	Grapes	Banana	Melon
3	Orange	Banana	Peach	Pear
4	Apple	Orange	Peach	Melon
5	Apple	Grapes	Banana	Peach
6	Grapes	Peach	Melon	Pear
7	Apple	Orange	Grapes	Pear

(3) ブロック＝選択肢集合として BWS 設問を作成

ブロック 1 から作成した BWS 設問

最も好き	果物	最も好き ではない
[]	Apple	[]
[]	Banana	[]
[]	Melon	[]
[]	Pear	[]

<途中の BWS 設問は省略>

ブロック 7 から作成した BWS 設問

最も好き	果物	最も好き ではない
[]	Apple	[]
[]	Orange	[]
[]	Grapes	[]
[]	Pear	[]

図 1 つり合い型不完備計画から Case 1 Best-Worst Scaling 設問の作成

が BWS 設問に対応するため、項目名で変換した BIBD の各行から全 7 問の BWS 設問を作成することができる。たとえば、BIBD の 1 行目（ブロック 1）に対応した BWS 設問は Apple, Banana, Melon, Pear の 4 つを選択肢とする一方、7 行目（ブロック 7）からは Apple, Orange, Grapes, Pear を選択肢とする BWS 設問が作成できる。なお、図 1 に示した BIBD は、後述する R 関数で作成したものである。

ステップ 5 は、BWS 調査の実施である。調査の実施方法は、一般的な質問調査や表明選好法に関する書籍が参考になろう。

ステップ 6 は、データセットの作成である。データセットは、回答結果と全選択肢集合を結合することで作成する。この際に注意すべきことは、分析方法ならびに分析に利用するソフトウェアに応じてデータセットの構造を検討しなければならないことである。support.BWS パッケージを前提としたデータセット構造については、次節で紹介する。

ステップ 7 は、項目の相対評価値の計算である。BWS の回答結果から各項目の相対評価を求める方法には、計数法とモデリング法がある。

計数法は、回答者 n が BWS の全設問を通じて項目 i をベストと評価した回数 (B_{in}) とワーストと評価した回数 (W_{in}) を数え上げ、その結果から算出した得点によって各項目の相対評価を行う方法である。得点には非集計（個人）レベルと集計（全体）レベルのものの 2 種類がある（Finn and Louviere 1992 ; Cohen 2009 ; Mueller and Runge 2009）。

非集計レベル得点には、項目毎のベスト評価回数とワースト評価回数の差として定義される非集計 BW 得点とその標準化得点

$$BW_{in} = B_{in} - W_{in}$$

$$\text{std. } BW_{in} = BW_{in} / r$$

がある。ただし、 r は全ての BWS 設問を通じて任意の項目が出現した回数である（BIBD では各項目の出現回数は等しい）。

集計レベル得点は、非集計レベル得点を回答者全体で足し合わせたものである。つまり、 N 人の回答者が全設問にわたって項目 i をベストと評価した回数 $B_i (= \sum_n B_{in})$ とワーストと評価した回数

$W_i (= \sum_n W_{in})$ に基づいて計算された集計 BW 得点とその標準化得点

$$BW_i = B_i - W_i$$

$$\text{std. } BW_i = BW_i / Nr$$

である。さらに、 B_i と W_i の比のルートに基づいた得点とその標準化得点

$$\text{sqrt. } BW_i = \sqrt{B_i / W_i}$$

$$\text{std. sqrt. } BW_i = \text{sqrt. } BW_i / \max.\text{sqrt. } BW$$

もある。ただし、 $\max.\text{sqrt. } BW$ は、全項目の中で最大の $\text{sqrt. } BW$ 値である。各項目の $\text{std. sqrt. } BW_i$ 得点を比較することで、相対的な重要度がわかりやすく捉えられる。たとえば、 $\text{std. sqrt. } BW_i$ の値が項目 A で 0.1、項目 B で 0.2 であるとすれば、項目 B は項目 A の 2 倍 ($= 0.2/0.1$) 重要であることがわかる。なお、非集計および集計レベル得点には、これら以外のバリエーションもある。詳細は BWS 関連文献を参照されたい。

モデリング法は、離散選択モデルで回答を分析し、得られた係数推定値から各項目の相対評価を行う方法である（離散選択モデルの文献については合崎（2015）を参照）。分析に際して、回答者が選択肢を評価するときに用いているルール（意思決定ルール）を仮定する必要がある。ここでは Max Diff モデルを仮定する。

1 つの選択肢集合（設問）に m 個の項目が含まれているとき、 m 個の項目から、項目 i がベスト、項目 j がワーストであると評価される組み合わせの数は $m \times (m-1)$ である。Max Diff モデルでは、回答者は各項目から効用 v を得ると仮定し、回答者が項目 i をベスト、項目 j をワーストに評価するのは、項目 i と j の効用差 ($v_i - v_j$) が、全ての組み合わせの効用差の中で最大であるためと仮定する。これらに加えて、誤差項に関する仮定を設けることで、項目 i をベスト、項目 j をワーストに評価する確率 $\text{Pr}(i, j)$ は、次の条件付きロジット・モデルとして定式化できる。

$$\text{Pr}(i, j) = \exp(v_i - v_j) / \sum_{k, k \neq i} \exp(v_k - v_i)$$

任意の 1 つの項目の係数をゼロに基準化する必要があるため、各項目の係数推定値は基準化項目と比べた効用差を表すこととなる。

さらに、係数推定値から、次に示す項目 i の Share of Preference (SP_i) が計算できる（Cohen 2003 ; Lusk and Briggeman 2009）。

$$SP_i = \exp(v_i) / \sum_i \exp(v_i)$$

この値は、計数法による *std.sqrr.BW* 得点に対応するものである。

3. R での実行例

1) 使用するパッケージの概要

表 1 は、BWS の適用ステップ毎に整理した利用可能な R 関数を示す。「全選択肢集合の設計」では、BIBD を作成するために *crossdes* パッケージ (Sailer 2013) に含まれている関数 *find.BIB()* と関数 *isGYD()* を利用する。「相対評価値の計算」のうち、モデリング法で利用する条件付きロジット・モデル (conditional logit model) の推定には、*gmnl* パッケージ (Sarrias et al. 2015) に含まれている関数 *gmnl()* を用いる。関数 *gmnl()* 向けの分析用データセットの作成には、*mlogit* パッケージ (Croissant 2013) の関数 *mlogit.data()* を用いる。これら以外の R 関数は *support.BWS* パッケージに含まれているものを利用する。なお、関数 *gmnl()* は確率係数ロジット・モデルなどの発展的な離散選択モデルも推定できる。詳しくは *gmnl* パッケージに含まれている *vignette* を参照されたい。

表 1 Case 1 Best-Worst Scaling の適用手順と対応する R 関数

手順	R 関数
1. 適用目的の決定	—
2. 評価シナリオの作成	—
3. 全選択肢集合の設計	<i>find.BIB()</i> <i>isGYD()</i>
4. 調査票の作成	<i>bws.questionnaire()</i>
5. 調査の実施	—
6. 分析用データセットの作成	<i>bws.dataset()</i>
7. 相対評価値の計算	<i>bws.count()</i> <i>mlogit.data()</i> <i>gmnl()</i>

2) 例題について

本稿では、BWS を適用する際の各種作業を R で実行する方法を解説するために、*support.BWS* パッケージに含まれているデータセット *fruit* を用いる。このデータセットは、7 種類の果物に対する消費者評価を課題とした適用事例を想定して、人工的に生成したものである。

図 1 の下段に、調査票で提示する BWS 設問イメージを示す。7 種類の果物から抽出された 4 種類の果物が各設問で提示される。評価基準は「好き」であり、回答者は「最も好きな果物」と「最

も好きではない果物」を選択する。提示される果物の組み合わせは、設問によって異なる。詳細は後述するが、本例では図 1 に示す形式の BWS 設問が全 7 問作成され、100 名の回答者がそれぞれ全 7 問に回答すると仮定する。

本例は、Aizaki et al. (2014) の Section 4.4.2 で示されている例を修正したものである。コードの一部は共通しているため、Aizaki et al. (2014) の Web サイト (<http://www.agr.hokudai.ac.jp/spmur/>) から該当する例コードを入手すれば、以下のコード入力作業は簡略化できよう。

以下、R を起動して本例で使用する 4 つのパッケージ (*crossdes*, *support.BWS*, *mlogit*, *gmnl*) は読み込み済みであると仮定する。これらのパッケージは R 本体には含まれていないので、利用者自身で CRAN から R にインストールする必要がある。

3) 全選択肢集合の設計

適用目的の決定と評価シナリオの作成は完了しているので、BIBD を利用した全選択肢集合の設計から開始する。

BIBD を生成するために利用する関数 *find.BIB()* は複数の引数を持つが、ここではそのうち 3 つの引数、項目数を指定する *trt*、1 問あたり項目数を指定する *k*、および全設問数を指定する *b* を利用する。ただし、これらの引数で指定された条件を満たす BIBD が存在するとは限らないこと、また存在するとしても同関数が対応する BIBD の生成に成功するとは限らないことから、生成された計画が BIBD であるか関数 *isGYD()* で確認する必要がある。

図 2 は、本例の条件に沿った BIBD を作成するためのコードとその実行結果 (出力) を示す。1 行目は乱数の種を指定している。これは、2 行目で用いる関数 *find.BIB()* が乱数を利用していることから、本稿の結果の再現性を担保するために設定しているに過ぎない。2 行目は、本例の条件に応じて引数を指定した関数 *find.BIB()* により計画を生成し、その結果をオブジェクト *sets.fruit* に保存するよう指示している。3 行目で *sets.fruit* に保存した計画をコンソールに表示するよう指示し、その結果が 4 ~ 11 行目に示されている。この計画が BIBD であるか確認するために関数 *isGYD()* を

利用したコードが12行目である。14行目に示されている結果から、得られた計画は BIBD であると判断できる。

なお、BIBD は実験計画法のテキストなどを参照し、行列形式として R に入力したものを、以降の作業で利用することもできる。

4) 設問形式への変換

関数 `bws.questionnaire()` を利用して、上記で生成した BIBD を調査票で提示する設問形式に変換する。同関数の引数は3つある。全選択肢集合を指定する `choice.sets`, 全選択肢集合を作成するために使用した実験計画法の種類を指定する `design.type`, そして BWS 設問で表示する項目名

(文字列ベクトル) を指定する `item.names` である。

図3は、同関数を利用してBIBDを設問形式に変換するコードとその出力を示す。1行目は BWS 設問で表示する項目名 (Apple から Pear の7種類の果物の名前) を文字列ベクトルとしてオブジェクト `items.fruit` に保存している。2～4行目は、本例の条件に沿って引数を指定した関数 `bws.questionnaire()` を実行している。引数 `design.type` は、利用する全選択肢集合の作成方法に応じて対応する番号を指定する必要がある、本例では BIBD であることから2を割り当てている。同関数の適用結果が、7行目以降に表示されている。この出力結果をワープロソフトの編集画面にコピー

行	コードと出力
1	<code>> set.seed(123)</code>
2	<code>> sets.fruit <- find.BIB(trt = 7, k = 4, b = 7)</code>
3	<code>> sets.fruit</code>
4	<code> [, 1] [, 2] [, 3] [, 4]</code>
5	<code>[1,] 1 4 6 7</code>
6	<code>[2,] 2 3 4 6</code>
7	<code>[3,] 2 4 5 7</code>
8	<code>[4,] 1 2 5 6</code>
9	<code>[5,] 1 3 4 5</code>
10	<code>[6,] 3 5 6 7</code>
11	<code>[7,] 1 2 3 7</code>
12	<code>> isGYD(sets.fruit)</code>
13	
14	<code>[1] The design is a balanced incomplete block design w.r.t. rows.</code>

図2 つり合い型不完備計画の生成

行	コードと出力
1	<code>items.fruit <- c("Apple", "Orange", "Grapes", "Banana", "Peach", "Melon", "Pear")</code>
2	<code>bws.questionnaire(choice.sets = sets.fruit,</code>
3	<code> design.type = 2,</code>
4	<code> item.names = items.fruit)</code>
5	
6	
7	<code>Q1</code>
8	<code>Best Items Worst</code>
9	<code>[] Apple []</code>
10	<code>[] Banana []</code>
11	<code>[] Melon []</code>
12	<code>[] Pear []</code>
13	
14	<code># <省略></code>
15	
16	<code>Q7</code>
17	<code>Best Items Worst</code>
18	<code>[] Apple []</code>
19	<code>[] Orange []</code>
20	<code>[] Grapes []</code>
21	<code>[] Pear []</code>

図3 全選択肢集合の設問形式への変換

ー&ペーストして加工すれば、BWS の設問を効率的に作成できよう。

5) 回答データセットの作成

調査の実施部分は省略し、BWS の分析用データセットを準備する作業に進む。調査後に取り組む最初の作業は、表計算ソフトなどを使って回答結果を整理して回答データセットを作成し、R に読み込む作業である。回答データセットは、各行が回答者 1 名に対応する形式とする。回答者の識別番号を表す変数を 1 列目に、2 列目以降に BWS 設問への回答結果を保存する回答変数を設定する。各回答変数には、それぞれの設問でいずれの項目がベストとワーストに評価されたかを表す数値を

保存する。回答者識別変数と回答変数の名前は任意に設定できるが、回答変数は設問毎に「ベスト」の次に「ワースト」が続くように作成しなければならない。たとえば、BWS 設問が 7 つある場合、回答変数は B1, W1, B2, W2, ..., B7, W7 とする。ここで、Bi と Wi は、それぞれ i 番目の設問でベストとワーストに選ばれた項目を表す回答変数である。

回答変数に保存する数値には 2 つの形式がある。1 つは行番号形式であり、ベストとワーストに選ばれた項目が表示されていた行番号を回答変数に保存する。たとえば、3 つの項目が設問で表示され、上から 2 つめの項目がベストに選ばれたとき

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ID	B1	W1	B2	W2	B3	W3	B4	W4	B5	W5	B6	W6	B7	W7
2	1	1	2	1	4	2	3	1	4	4	1	2	3	2	4
3	2	1	3	4	3	4	3	3	4	1	4	4	3	1	2
4	3	2	3	1	4	2	3	2	4	3	2	4	1	2	3
5	4	1	2	2	4	3	4	1	4	1	3	2	3	1	3
6	5	4	2	1	3	1	2	2	3	1	3	4	2	2	1
7	6	1	2	1	2	3	2	3	4	4	2	4	1	2	3

図 4 表計算ソフトを利用した調査結果の整理イメージ

行	コードと出力
1	> data(fruit)
2	> head(fruit)
3	ID B1 W1 B2 W2 B3 W3 B4 W4 B5 W5 B6 W6 B7 W7
4	1 1 1 2 1 4 2 3 1 4 4 1 2 3 2 4
5	2 2 1 3 4 3 4 3 3 4 1 4 4 3 1 2
6	3 3 2 3 1 4 2 3 2 4 3 2 4 1 2 3
7	4 4 1 2 2 4 3 4 1 4 1 3 2 3 1 3
8	5 5 4 2 1 3 1 2 2 3 1 3 4 2 2 1
9	6 6 1 2 1 2 3 2 3 4 4 2 4 1 2 3
10	> data.fruit <- bws.dataset(respondent.dataset = fruit,
11	response.type = 1,
12	choice.sets = sets.fruit,
13	design.type = 2,
14	item.names = items.fruit)
15	> head(data.fruit)
16	Q PAIR ID RES.B RES.W BEST WORST Apple Orange Grapes Banana Peach Melon Pear
17	1 1 1 1 1 4 1 4 1 0 0 -1 0 0 0
18	2 1 2 1 1 4 1 6 1 0 0 0 0 -1 0
19	3 1 3 1 1 4 1 7 1 0 0 0 0 0 -1
20	4 1 4 1 1 4 4 1 -1 0 0 1 0 0 0
21	5 1 5 1 1 4 4 6 0 0 0 1 0 -1 0
22	6 1 6 1 1 4 4 7 0 0 0 1 0 0 -1
23	RES STR
24	1 TRUE 101
25	2 FALSE 101
26	3 FALSE 101
27	4 FALSE 101
28	5 FALSE 101
29	6 FALSE 101

図 5 本例で使用する例題データセットの読み込みと分析用データセットの作成

には、行番号は2となり、その設問に対応するベスト回答変数には2の値を保存する。もう1つの項目番号形式では、ベストとワーストに選ばれた項目の項目番号を回答変数に保存する。たとえば、全7項目を評価対象としている場合、各項目には1から7のいずれかの整数値が項目番号として割り当てられている。そして、ある設問で項目番号3に該当する項目がベストに選択されたとすれば、その設問に対するベスト回答変数には3の値を保存する。

図4は、本例をイメージして表計算ソフトに回答結果を入力したものを示す（最初の6名分のみ）。1行目の変数名、2行目以降が回答者毎の回答結果となる。回答変数は、行番号形式で入力されている。本例の回答者は100名であるので、101行目までデータが入力されていることになる。列は、回答者識別変数を表す変数ID（1列目）、BWSの各設問でベストとワーストに選択された項目を表す変数B1~W7（2~15列目）で構成されている。たとえば、2行目からは、回答者1（ID=1）はBWSの1問目で1行目に表示された項目（B1=1）をベストに、2行目に表示された項目（W1=2）をワーストに選択した一方、7問目では2行目の表示項目をベスト（B7=2）に、4行目の表示項目をワースト（W7=4）に選択したことが読み取れる。

本例の回答結果は、すでにデータセット `fruit` として `support.BWS` パッケージに準備されているが（図5の1~9行）、読者が自分の課題に取り組むときは、上記のように表計算ソフトなどでデータセットを準備し、Rに読み込む必要がある。なお、欠損値があると、以降の作業に支障があるため、回答結果に欠損値のある回答者は回答データセットから除外する必要がある。

6) 分析用データセットの作成

続いて、関数 `bws.dataset()` を利用して、回答データセットと全選択肢集合を1つに統合して分析用データセットを作成する。同関数は6つの引数を持つが、ここでは5つの引数を利用する。回答データセットを指定する `respondent.dataset`、回答変数のデータ形式を指定する `response.type`、全選択肢集合を指定する `choice.sets`、全選択肢集合を作成するために使用した実験計画法を指定す

る `design.type`、そして BWS 設問で表示する項目名（文字列ベクトル）を指定する `item.names` である。最後の3つの引数は、関数 `bws.questionnaire()` における同名の引数と同じものである。

図5の10~14行目は、本例の条件に従って関数 `bws.dataset()` を実行し、その結果をオブジェクト `data.fruit` に保存するコードを示す。回答データセットにおける回答変数は行番号形式で作成されているため、引数 `response.type` には行番号形式であることを示す1を割り当てている（項目番号形式のときは2）。作成された分析用データセットの一部を16行目以降に示す。分析用データセットは、1行がベスト項目とワースト項目の可能な組み合わせの1つに対応している。本例では、BWSの設問1つで4つの項目が表示されるため、1問あたりのベスト項目とワースト項目の可能な組み合わせは $4 \times 3 = 12$ 通りとなる。つまり、12行で回答者1名の1問分のデータセットを構成する。本例では100名の回答者それぞれが全7問のBWS設問に回答していることから、分析用データセットは8,400行（12行×7問×100名）となる。

各変数の定義は、次の通りである。変数QはBWS設問の番号を示す。変数PAIRは、設問毎に作成されるベスト項目とワースト項目の可能な組み合わせを表す通し番号である。変数IDは、回答者番号である。変数RES.Bと変数RES.Wは、それぞれ回答者によってベスト項目あるいはワースト項目と評価されたものの項目番号である。変数BESTとWORSTは、それぞれ設問毎に作成されるベスト項目とワースト項目の可能な組み合わせにおいて、ベストあるいはワーストとして取り扱われる項目の番号である。AppleからPearは、各項目の状態変数である。変数名は適用事例によって異なるが（引数 `item.names` で指定された変数名に基づいて設定されるため）、当該変数に対応する項目がベストとして処理されるときは1、ワーストとして処理されるときは-1、そのほかのときには0の値をとる。離散選択モデルで分析するときの説明変数に該当する変数群である。変数RESは、BWS設問への回答である。対応する行のベスト項目とワースト項目の組み合わせが回答者によって選択されているときはTRUE、そのほかのときにはFALSEの値をとる。離散選択モ

デルで分析するときの被説明変数に該当する変数である。変数 STR は、回答者と設問の各組み合わせを区別するための層化変数である。離散選択モデルを推定する一部の関数で利用される。

以上の変数の定義に従えば、データセットの 1 行目（図 5 全体でみれば 17 行目および 24 行目）は、次のように読み取ることができる。回答者 1（ID = 1）が、問 1（Q = 1）において直面したベスト項目とワースト項目の可能な組み合わせのうちの 1 つである「項目 1 をベスト（BEST = 1）、項目 4 をワースト（WORST = 4）」とする組み合わせに対応する行であり、回答者 1 は当該組み合わせを選択した（RES.B = 1 かつ RES.W = 4、あるいは RES = TRUE）ことを表している。

7) 計数法による相対評価

関数 `bws.count()` を利用して、非集計および集計 BW 得点を計算する。同関数の引数は `data` のみであり、関数 `bws.dataset()` で作成した分析用データセットを割り当てる。図 6 の 1 行目は、本例に沿って引数を指定した関数 `bws.count()` を実行し、その結果をオブジェクト `count.fruit` に保存するよう指示している。2 行目でそれを表示するよう指示し、4 行目に回答者数、6 行目以降に非集計 BW 得点の要約が示されている。列 `mean-`

BW に示されている結果より、回答者全体では Apple が最も好まれており（1.98）、次いで Orange（0.98）、Grapes（-0.04）、Peach と Pear（-0.25）、Banana（-0.30）と続き、最後が Melon（-2.12）となっている。16 行目以降に表示されている集計 BW 得点のうち `std.sqrtBW` 得点からは、最も好まれている Apple と最も好まれていない Melon では、好みの違いが約 14 倍（ $=1/0.0703$ ）となっていることがわかる。

図 6 では、結果の要約等が表示されているが、関数 `bws.count()` の実際の出力は 3 要素から構成されるリストであり、より詳細な計算結果を含んでいる。1 つめの要素である非集計 BW 得点に関するリスト（`disaggregate`）は、回答者番号（ID）、回答者毎に整理された項目毎のベストないしはワースト選択回数（B と W）、回答者毎に整理された項目毎の BW 得点（BW）とその標準化値（`stdBW`）から構成されている。2 つめの要素は集計 BW 得点を含むデータフレーム（`aggregate`）であり、項目毎のベスト選択回数とワースト選択回数（B と W）、項目毎の BW 得点（BW）とその標準化得点（`stdBW`）、項目毎の B と W の比のルートに基づいた得点（`sqrtBW`）とその標準化得点（`std.sqrtBW`）から構成されている。3 つめの要

行	コードと出力
1	<code>> count.fruit <- bws.count(data = data.fruit)</code>
2	<code>> count.fruit</code>
3	
4	Number of respondents = 100
5	
6	Summary of disaggregated best-worst scores:
7	meanB meanW meanBW mean.stdBW stdev.stdBW
8	Apple 2.13 0.15 1.98 0.4950 0.2864
9	Orange 1.49 0.51 0.98 0.2450 0.3860
10	Grapes 1.08 1.12 -0.04 -0.0100 0.4923
11	Banana 0.72 1.02 -0.30 -0.0750 0.3735
12	Peach 0.67 0.92 -0.25 -0.0625 0.3264
13	Melon 0.16 2.28 -2.12 -0.5300 0.2713
14	Pear 0.75 1.00 -0.25 -0.0625 0.3085
15	
16	Aggregated best-worst scores:
17	B W BW stdBW sqrtBW std.sqrtBW
18	Apple 213 15 198 0.4950 3.7683 1.0000
19	Orange 149 51 98 0.2450 1.7093 0.4536
20	Grapes 108 112 -4 -0.0100 0.9820 0.2606
21	Banana 72 102 -30 -0.0750 0.8402 0.2230
22	Peach 67 92 -25 -0.0625 0.8534 0.2265
23	Melon 16 228 -212 -0.5300 0.2649 0.0703
24	Pear 75 100 -25 -0.0625 0.8660 0.2298

図 6 計数法による BW 得点の計算

素は BW 設問に関する基本情報で構成されているリスト (information) であり, 回答者数 (nrespondents), 項目数 (nitems), 選択肢集合に含まれる各項目の出現頻度 (fitem), および各項目の名前 (vnames) を含んでいる. これら関数 bws.count() の出力の詳細を知りたいときには, 関数 str() で出力構造を把握した上で, 各要素にアクセスされたい.

出力要素の活用例として, ここでは非集計 BW 得点に基づいた各果物の BW 得点の分布を求める (図 7). 1 行目で非集計 BW 得点のみを計算結果オブジェクトからデータフレームとして取り出し, 2 行目で水準を指定して各項目の BW 得点を因子化する. 3 行目で項目毎に関数 table() を適用して, -4 から 4 までの BW 得点毎の回答者

分布を求める. 5 行目以降に, 作成した表が示されている. 関数 bws.count() の出力 (要約) から予想できるように, 相対的に最も評価の高かった Apple では正の BW 得点の範囲に回答者が分布する一方, 相対的に最も評価の低かった Melon では数名の回答者を除いて負の BW 得点の範囲に回答者が集中している.

8) モデリング法による相対評価

最後に, モデリング法で果物の相対評価を求める. 関数 gmn1() を利用した分析コードの説明に入る前に, 本例における確定効用関数を具体的に設定する. 本例では, Pear の効用をゼロに基準化した上で, それ以外の果物の効用を当該変数の係数推定値として推定できるように, 次の通りに確定効用関数を仮定している.

行	コードと出力
1	> BW <- data.frame(count.fruit\$disaggregate\$BW)
2	> BW <- lapply(BW, factor, levels = c(-4:4))
3	> score.tables <- lapply(BW, table)
4	> score.tables
5	\$Apple
6	
7	-4 -3 -2 -1 0 1 2 3 4
8	0 0 0 1 9 24 32 25 9
9	
10	\$Orange
11	
12	-4 -3 -2 -1 0 1 2 3 4
13	0 2 3 10 23 28 14 16 4
14	
15	\$Grapes
16	
17	-4 -3 -2 -1 0 1 2 3 4
18	2 11 15 12 18 17 15 8 2
19	
20	\$Banana
21	
22	-4 -3 -2 -1 0 1 2 3 4
23	2 4 13 26 29 15 6 5 0
24	
25	\$Peach
26	
27	-4 -3 -2 -1 0 1 2 3 4
28	0 2 16 28 24 20 9 1 0
29	
30	\$Melon
31	
32	-4 -3 -2 -1 0 1 2 3 4
33	8 32 32 21 6 1 0 0 0
34	
35	\$Pear
36	
37	-4 -3 -2 -1 0 1 2 3 4
38	1 1 15 23 31 24 4 1 0

図 7 非集計 BW 得点の作表

$$v = b_1 \text{Apple} + b_2 \text{Orange} + b_3 \text{Grapes} + b_4 \text{Banana} + b_5 \text{Peach} + b_6 \text{Melon}$$

ただし、各 b は推定すべき係数である。分析用データセットの説明で述べたように、各果物に対応する変数 (Apple から Melon) は、当該果物がベストに選ばれたとき 1、ワーストに選ばれたとき -1、それら以外のとき 0 の値を取るよう設計されている。したがって、推定された各係数が、基準とされた項目 (今回であれば Pear) を 0 としたときの相対評価を表すことになる。

図 8 の 1 ~ 5 行目は、分析用データセットを関数 `gmnl()` に適したものに変換するためのコード

を示す。関数 `gmnl()` で使用するデータセットは、データセットに関する様々な情報を属性として保持している必要がある。`mlogit` パッケージの関数 `mlogit.data()` を使い、引数に適切な条件を指定することで、データセットに必要な情報を付与できる。ここでは、引数 `data` に本例の分析用データセット `data.fruit` を割り付け、その中に含まれる変数 `RES` が選択結果を保存する変数であることをデータセットの属性として記録できるよう引数 `choice` に“RES”を指定している。オリジナルのデータが `long` 形式であることから引数 `shape` で“long”を指定している。引数 `alt.var` と引数 `id.var` には、

行	コードと出力
1	<code>> data.fruit.ml <- mlogit.data(data = data.fruit,</code>
2	<code>+</code>
3	<code>choice = "RES",</code>
4	<code>+</code>
5	<code>shape = "long",</code>
6	<code>alt.var = "PAIR",</code>
7	<code>id.var = "ID")</code>
8	<code>> fr.fruit <- RES ~ Apple + Orange + Grapes + Banana + Peach + Melon - 1</code>
9	<code>> cndl.fruit <- gmnl(formula = fr.fruit,</code>
10	<code>+</code>
11	<code>data = data.fruit.ml,</code>
12	<code>model = "mnl")</code>
13	<code>> summary(cndl.fruit)</code>
14	Model estimated on: 土 7 09 9:14:30 2016
15	Call:
16	<code>gmnl(formula = fr.fruit, data = data.fruit.ml, model = "mnl",</code>
17	<code>method = "nr")</code>
18	Frequencies of categories:
19	
20	1 2 3 4 5 6 7 8 9
21	0.132857 0.170000 0.191429 0.034286 0.122857 0.087143 0.017143 0.038571 0.075714
22	10 11 12
23	0.030000 0.040000 0.060000
24	
25	The estimation took: 0h:0m:1s
26	
27	Coefficients:
28	Estimate Std. Error z-value Pr(> z)
29	Apple 1.123771 0.105979 10.6037 < 2.2e-16 ***
30	Orange 0.610309 0.101378 6.0201 1.743e-09 ***
31	Grapes 0.099832 0.098553 1.0130 0.3111
32	Banana -0.024739 0.101508 -0.2437 0.8075
33	Peach -0.019496 0.099702 -0.1955 0.8450
34	Melon -0.962572 0.105843 -9.0944 < 2.2e-16 ***
35	---
36	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
37	
38	Optimization of log-likelihood by Newton-Raphson maximisation
39	Log Likelihood: -1516.1
40	Number of observations: 700
41	Number of iterations: 4
42	Exit of MLE: gradient close to zero>

図 8 関数 `gmnl()` に適した分析用データセットへの変換と条件付きロジット・モデルの推定

それぞれ選択肢識別変数（今回は変数 PAIR）と回答者識別変数（今回は変数 ID）を指定する．6行目は、関数 `gmnl()` で推定するモデル式を事前に設定して、オブジェクト `fr.fruit` に保存している．記号 `~`（チルダ）の左辺に選択結果を表す変数 `RES` を、右辺に説明変数である変数 `Apple` から変数 `Melon` までを設定している．右辺の末尾の「`-1`」は、モデルに定数を含めないようにするための措置である．7～9行で関数 `gmnl()` を利用して条件付きロジット・モデルを推定し、その結果をオブジェクト `cndl.fruit` に保存し、10行でその要約を表示するよう指示している．関数 `gmnl()` で条件付きロジット・モデルを推定するときには、引数 `formula` と引数 `data` にモデル式と推定に利用する変数を含んだデータセット（データフレーム）を指定する一方、引数 `model` に“`mnl`”を指定する．関数 `gmnl()` は、推定するモデルによってモデル式と引数の設定が変化するため、本稿では一般的な説明は省略する．関数 `summary()` を適用した結果のうち、28～34行に各説明変数の係数推定値と関連する検定統計量が表示されている． P 値（列 `Pr(>|z|)`）を踏まえると、係数をゼロに制約した `Pear` と比べて有意な評価差を持つ果物は `Apple`、`Orange`、および `Melon` の3つと判断できる．

図9は、`Share of Preference` を計算するコードである．得られた係数推定値ベクトルには、係数をゼロに制約した `Pear` が含まれていないことから、1行目で係数推定値ベクトルをオブジェクト `b` にコピーした上で、オブジェクト `items.fruit` における `Pear` の位置（7番目）に対応するように、`Pear` の係数（0）とその名前（“`Pear`”）をオブジェクト `b` に挿入している（2～3行目）．4行目で各項目の `Share of Preference` を計算してオブジェクト `SP` に保存し、その結果（6～7行）を表示するよう指示（5行）している．`Share of Pref-`

`erence` でみると、最も好まれている `Apple` と最も好まれていない `Melon` では、評価の違いが約8倍（ $=0.329/0.041$ ）となっている．

4. おわりに

本稿では、`R` を利用して基本的な `BWS` の適用手順を例示した．実証研究においては、回答者の個人・世帯特性と `BW` 得点との関連の分析や `BW` 得点による回答集団の分類なども試みられている．応用性の高い手法であることから、`BWS` を活用するためには、分野を問わずに既存文献を参照して、アイディアを膨らませることが重要になるだろう．

付 記

本稿は、各地の教育・研究機関で実施してきた講義やゼミの経験が反映されている．これまでに協力、参加いただいた方々に感謝申し上げる．本稿は、`JSPS` 科研費25450341の助成による成果の一部である．

文 献

- 合崎英男 (2015) 「`R` パッケージ `support.CEs` と `survival` を利用した離散選択実験の実施手順」『農経論叢』70, 1-16.
- Aizaki, H. (2015) *support.BWS: Basic Functions for Supporting an Implementation of Best-Worst Scaling*, `R` Package Version 0.1-3. URL <https://CRAN.R-project.org/package=support.BWS>.
- Aizaki, H., Nakatani, T., and Sato, K. (2014) *Stated Preference Methods Using R*. CRC Press.
- Aizaki, H., Onimaru, T., Katayama, C., and Ishida, K. (2016) Non-farmers' Preference for Assisting with Farm Tasks as a Method of Health Promotion. *Japan Agricultural Research Quarterly*, 50, 135-142.
- Auger, P., Devinney, T. M., and Louviere, J. J. (2007)

行	コードと出力
1	<code>> b <- c(coef(cndl.fruit))</code>
2	<code>> b[7] <- 0</code>
3	<code>> names(b)[7] <- "Pear"</code>
4	<code>> SP <- exp(b)/sum(exp(b))</code>
5	<code>> print(SP, digits = 2)</code>
6	Apple Orange Grapes Banana Peach Melon Pear
7	0.329 0.197 0.118 0.104 0.105 0.041 0.107

図9 Share of Preference の計算

- Using Best-Worst Scaling Methodology to Investigate Consumer Ethical Beliefs across Countries. *Journal of Business Ethics*, 70, 299-326.
- Cohen, E. (2009) Applying Best-Worst Scaling to Wine Marketing. *International Journal of Wine Business Research*, 21, 8-23.
- Cohen S. H. (2003) Maximum Difference Scaling: Improved Measures of Importance and Preference for Segmentation. *Sawtooth Software Research Paper Series*, 1-17. <http://www.sawtoothsoftware.com/support/technical-papers>.
- Croissant, Y. (2013) *mlogit: Multinomial Logit Model*, R Package Version 0.2-4. <https://CRAN.R-project.org/package=mlogit>.
- Dumbrell, N. P., Kragt, M. E., and Gibson, F. L. (2016) What Carbon Farming Activities Are Farmers Likely to Adopt? A Best-Worst Scaling Survey. *Land Use Policy*, 54, 29-37.
- Finn A. and Louviere, J. J. (1992) Determining the Appropriate Response to Evidence of Public Concern: The Case of Food Safety. *Journal of Public Policy & Marketing*, 11(2), 12-25.
- Flynn, T. N. (2010) Valuing Citizen and Patient Preferences in Health: Recent Developments in Three Types of Best-Worst Scaling. *Expert Review of Pharmacoeconomics & Outcomes Research*, 10, 259-267.
- Kuheld, W. F. (2010) *Marketing Research Methods in SAS*. SAS Institute. http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html.
- Louviere, J., Lings, I., Islam, T., Gudergan, S., and Flynn, T. (2013) An Introduction to the Application of (Case 1) Best-Worst Scaling in Marketing Research. *International Journal of Research in Marketing*, 30, 292-303.
- Louviere, J. J., Flynn, T. N., and Marley, A. A. J. (2015) *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.
- Lusk, J. L. and Briggeman, B. C. (2009) Food Values. *American Journal of Agricultural Economics*, 91, 184-196.
- Mueller S. and Rungie, C. (2009) Is There More Information in Best-Worst Choice Data?: Using the Attitude Heterogeneity Structure to Identify Consumer Segments. *International Journal of Wine Business Research*, 21, 24-40.
- 西原是良・中嶋康博・木村匡臣・飯田俊彰 (2016) 「農業用水の多面的機能に対するサービス科学的考察」『農業経済研究』87(4), 371-376.
- R Core Team (2016) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Sailer, M. O. (2013) *crossdes: Construction of Cross-over Designs*, R Package Version 1.1-1. <https://CRAN.R-project.org/package=crossdes>.
- Sarrias, M., Daziano, R., and Croissant, Y. (2015) *gmnl: Multinomial Logit Models with Random Parameters*, R package version 1.1-1. <https://CRAN.R-project.org/package=gmnl>.
- 澤田学・合崎英男・佐藤和夫 (2010) 「牛肉生産における飼料自給率向上の利点に関する消費者評価」『帯広畜産大学学術研究報告』31, 18-24.
- Wolf, C. A. and Tonsor, G. (2013) Dairy Farmer Policy Preferences. *Journal of Agricultural and Resource Economics*, 38, 220-234.
- Sawtooth Software (2016) *MaxDiff*. <http://www.sawtoothsoftware.com/products/maxdiff-software>