

R と対応分析

1. 対応分析とは

対応分析 (correspondence analysis) は、フランスのベンゼクリ (Benzécri) によって 1960 年代に提唱され、1970 年代から普及し始めたカテゴリカルデータの解析方法で、コレスポンデンス分析とも呼ばれている。

類似の方法としては、1950 年代に林知己夫氏によって提案された数量化Ⅲ類、1980 年代に西里静彦氏によって提案された双対尺度法 (dual scaling) などがある。それぞれの方法が提案された背景は異なるが、基本的なアプローチおよびアルゴリズムの中核は同じである。

データ形式によっては、それぞれの手法の解析結果は変換によって一致させることも可能である。一時的には、数量化Ⅲ類と対応分析は異なるデータ分析方法と見なされたが、既に数理的には同等であることが証明されている。

数量化Ⅲ類および対応分析の基本的考え方は、分割表において、行の項目と列の項目の相関が最大になるように、行と列の双方を並び替えることである。問題解決のアプローチは、主成分分析、因子分析とほぼ同じである。そこで、対応分析を制約つき主成分分析 [1]、正準相関分析 [2]、質的因子分析として見なす研究者もいる。

対応分析は、データの構造を再現する面では、古典的主成分分析より効果が劣るが、特徴別に分類する面では、古典的主成分分析より良い結果を示すケースが多い。

対応分析の大まかなアルゴリズムを説明するため、表 1 のような度数に関する分割表がある

とする。

表 1 データ行列 $F_{r \times c}$

	x_1	x_2	...	x_j	...	x_c	合計
個体 1	f_{11}	f_{12}	...	f_{1j}	...	f_{1c}	$f_{1\bullet}$
個体 2	f_{21}	f_{22}	...	f_{2j}	...	f_{2c}	$f_{2\bullet}$
⋮	⋮	⋮	...	⋮	...	⋮	⋮
個体 i	f_{i1}	f_{i2}	⋮	f_{ij}	⋮	f_{ic}	$f_{i\bullet}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
個体 r	f_{r1}	f_{r2}	⋮	f_{rj}	⋮	f_{rc}	$f_{r\bullet}$
合計	$f_{\bullet 1}$	$f_{\bullet 2}$	⋮	$f_{\bullet j}$	⋮	$f_{\bullet c}$	n

データ行列 $F_{r \times c}$ の各要素を総度数 n で割ったデータ ($P_{r \times c}$) を表 2 に示す。

$$p_{ij} = \frac{f_{ij}}{n}, \quad i = 1, 2, \dots, r \quad j = 1, 2, \dots, c$$

表 2 データ行列 $P_{r \times c}$

	x_1	x_2	...	x_j	...	x_c	合計
個体 1	p_{11}	p_{12}	...	p_{1j}	...	p_{1c}	$p_{1\bullet}$
個体 2	p_{21}	p_{22}	...	p_{2j}	...	p_{2c}	$p_{2\bullet}$
⋮	⋮	⋮	...	⋮	...	⋮	⋮
個体 i	p_{i1}	p_{i2}	⋮	p_{ij}	⋮	p_{ic}	$p_{i\bullet}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
個体 r	p_{r1}	p_{r2}	⋮	p_{rj}	⋮	p_{rc}	$p_{r\bullet}$
合計	$p_{\bullet 1}$	$p_{\bullet 2}$	⋮	$p_{\bullet j}$	⋮	$p_{\bullet c}$	1

このような分割表の独立性の問題では、通常カイ 2 乗統計量を用いる。カイ 2 乗統計量の各セルの値は次の式で定義されている。

$$\chi_{ij}^2 = \frac{(f_{ij} - f_{i\bullet}f_{\bullet j}/n)^2}{f_{i\bullet}f_{\bullet j}/n}$$

$$\chi_{ij} = \sqrt{n} \frac{f_{ij} - f_{i\bullet}f_{\bullet j}/n}{\sqrt{f_{i\bullet}f_{\bullet j}}} = \sqrt{n} \frac{p_{ij} - p_{i\bullet}p_{\bullet j}}{\sqrt{p_{i\bullet}p_{\bullet j}}}$$

対応分析では、データ $F_{r \times c}$ あるいは $P_{r \times c}$ を次の式で変換したデータの固有値問題に帰する。

$$z_{ij} = \frac{f_{ij} - f_{i \cdot} f_{\cdot j} / n}{\sqrt{f_{i \cdot} f_{\cdot j}}} = \frac{p_{ij} - p_{i \cdot} p_{\cdot j}}{\sqrt{p_{i \cdot} p_{\cdot j}}}$$

表 3 データ行列 $Z_{r \times c}$

	x_1	x_2	...	x_j	...	x_c
個体 1	z_{11}	z_{12}	...	z_{1j}	...	z_{1c}
個体 2	z_{21}	z_{22}	...	z_{2j}	...	z_{2c}
...
個体 i	z_{i1}	z_{i2}	...	z_{ij}	...	z_{ic}
...
個体 r	z_{r1}	z_{r2}	...	z_{rj}	...	z_{rc}

対応分析では、分割表の列の効果は $Q = Z'Z$ 、行の効果は $Q' = ZZ'$ の固有ベクトル (U, V) をそれぞれ $\frac{1}{\sqrt{p_{\cdot j}}}$ と $\frac{1}{\sqrt{p_{i \cdot}}}$ で基準化した値 ($D_c^{-1/2}U$ 、

$D_r^{-1/2}V$) を用いる。この D_c は $p_{\cdot j}$ を要素とした対角行列、 D_r は $p_{i \cdot}$ を要素とした対角行列である。

$Q = Z'Z$ の固有値が $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$ 、 $k = \text{rank}(Q) \leq \min(r, c)$ である場合、次の式が成り立つ。

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(f_{ij} - f_{i \cdot} f_{\cdot j} / n)^2}{f_{i \cdot} f_{\cdot j} / n} = n \sum_{i=1}^k \lambda_i$$

2. パッケージ MASS の対応分析

2 元分割表のデータを対象とする対応分析をシンプルコレスポネンシ (Simple Correspondence) 分析、多項目の反応パターン表のデータを対象とした対応分析をマルチプルコレスポネンシ (Multiple Correspondence) 分析と呼ぶが、通常前者を略して対応分析、後者

を多重対応分析と呼ぶ。

R には、幾つかのパッケージに対応分析に関連する関数が含まれているが、まずパッケージ MASS 中の関数 **corresp**、**mca** を説明することにする。

2.1 対応分析

パッケージ MASS 中の対応分析の関数 **corresp** の書き方を次に示す。

```
corresp(x, nf, ...)
```

引数 x はデータマトリックスあるいはデータフレームである。 nf は、求める軸の数 (主成分、因子とも呼ぶ) を指定する引数である。デフォルトは 1 になっているので nf を省略した場合は、1 軸の結果のみが返される。

関数 **corresp** を用いるためにはパッケージ MASS を読み込む (`library(MASS)`) 手続きが必要である。

データを用いて関数 **corresp** の使用方法を説明することにする。データの入力の手間を省くため、パッケージ MASS 中のデータセット **caith** を用いることにする。

データ **caith** はイギリスに住んでいる人々の目の色 (blue, light, medium, dark) と髪の色に (fair, red, medium, dark, black) 関して 5387 人を対象として行った調査結果である。

データセット **caith** は 4 行 5 列の 2 元分割表である。行が目の色、列が髪の色になっている。

```
> caith
      fair red medium dark black
blue  326  38   241  110    3
light  688 116   584  188    4
medium 343  84   909  412   26
dark   98  48   403  681   85
```

関数 **corresp** を用いたデータ **caith** の対応分析のコマンドおよびその出力結果を次に示す。

```

> (caith.ca<-corresp(caith,nf=4))
First canonical correlation(s): 4.463684e-01 1.734554e-
Row scores:
      [,1]      [,2]      [,3] [,4]
blue -0.89679252  0.9536227  2.1884132  1
light -0.98731818  0.5100045 -1.0837859  1
medium 0.07530627 -1.4124778  0.1894089  1
dark  1.57434710  0.7720361 -0.1482208  1
Column scores:
      [,1]      [,2]      [,3] [,4]
fair -1.21871379  1.0022432  0.4271282 -0.8692696
red -0.52257500  0.2783364 -4.0268545 -1.3400421
medium -0.09414671 -1.2009094  0.1103959 -0.8453208
dark  1.31888488  0.5992920  0.3450676 -1.2251588
black  2.45176017  1.6513565 -1.5736976  1.1609621

```

ここでは軸数の引数 `nf` を 4 にしている。関数 `corresp` では、累積寄与率を返さないのので累積寄与率を計算するためには、`nf=min(行数、列数)` にしたほうがよい。もしデータ行列のランクが `min(行数、列数)` より小さいときには、`nf` を下げればよい。

返される結果は、正準相関 (canonical correlation)、行の得点 (Row scores)、列の得点 (Column scores) の順になっている。返される正準相関は大きい順に並べられ、その 2 乗が固有値に等しい。対応分析では、計算された軸の行・列に対応する値をそれぞれ行の得点、列の得点と呼ぶ。

対応分析でも主成分分析や因子分析と同じく各軸の寄与率や累積寄与率について考察を行うが、関数 `corresp` は寄与率を返さないのので、正準相関を用いて算出する必要がある。

正準相関は `$cor` に記録されている。データ `caith` の固有値および寄与率は次のコマンドで求めることができる。

```

>固有値<- caith.ca$cor^2
> round(固有値,3)
[1] 0.199 0.030 0.001 0.000
> round(100*固有値 /sum(固有値),2)
[1] 86.56 13.07 0.37 0.00

```

第 2 固有値までの累積寄与率は 99.63% で、非常に高い。このような場合は第 1, 2 固有値に対応する得点のみを分析すればよい。

対応分析では、主成分分析や因子分析と同じく、寄与率が高い首位の固有値に対応する行・列の得点の値の大小とそれらの相対関係について分析する。そのため、行の得点と列の得点を同じの画面に配置した散布図 (biplot) がよく用いられている。

関数 `corresp` の結果は、関数 `plot` で第 1, 2 軸の得点の散布図 (biplot) を作成することができる。次にデータセット `caith` の第 1, 2 軸の得点の散布図を作成するコマンドおよびその結果を図 1 に示す。

```
> biplot(caith.ca)
```

図 1 `corresp(caith, nf=2)` の biplot

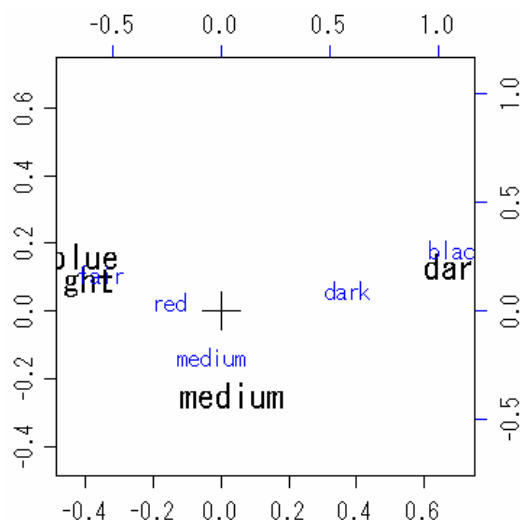


図 1 から分かるように、目の色が `dark` の人は髪色が `black` の人が多く、髪色が `fair` の人は目の色が `blue` か `light` の人が多いことが読み取られる。

関数 `biplot` に引数 `type="row"` (あるいは `type="col"`) を用いることで、行 (あるいは列) を基準とした散布図を作成することもできる。

関数 `corresp` で計算された行の得点は `$rscore`、列の得点は `$cscore` にマトリックス形式で記録されている。

2.2 多重対応分析

(1) 関数 `corresp` を用いる場合

説明の便利のため、3つの項目 A、B、C について 10 人のアンケート回答の結果を考えよう。項目 A には 2 つの選択肢(A1、A2)、項目 B には 3 つの選択肢(B1、B2、B3)、項目 C には 3 つの選択肢(C1、C2、C3)があるとす。

例えば、10 人の回答結果は表 4 のようになったとする。表 4(a)では、回答の番号を文字列で記入し、表 4(b)では回答番号を数字に置き換えて記入している。また、表 4 のデータは、表 5 のように項目を選択している場合は「1」、選択していない場合は「0」で記入することも可能である。

表 4 10 人のアンケート回答の結果 (架空)

		項目					項目		
		A	B	C			A	B	C
1	A1	B1	C2	1	1	1	2		
2	A1	B2	C3	2	1	2	3		
3	A1	B3	C1	3	1	3	1		
4	A1	B1	C3	4	1	1	3		
5	A1	B1	C2	5	1	1	2		
6	A2	B2	C1	6	2	2	1		
7	A2	B2	C3	7	2	2	3		
8	A2	B1	C1	8	2	1	1		
9	A2	B3	C1	9	2	3	1		
10	A2	B2	C2	10	2	2	2		

表 5 10 人のアンケート回答の (0, 1) データ

	項目 A		項目 B			項目 C		
	A-1	A-2	B-1	B-2	B-3	C-1	C-2	C-3
1	1	0	1	0	0	0	1	0
2	1	0	0	1	0	0	0	1
3	1	0	0	0	1	1	0	0
4	1	0	1	0	0	0	0	1
5	1	0	1	0	0	0	1	0
6	0	1	0	1	0	1	0	0
7	0	1	0	1	0	0	0	1
8	0	1	1	0	0	1	0	0
9	0	1	0	0	1	1	0	0
10	0	1	0	1	0	0	1	0

このような表 4 および表 5 のデータ形式の対応分析を多重対応分析と呼ぶ。表 4(b) および表 5 は数量化Ⅲ類で扱っているデータ形式である。関数 `corresp` を用いて、多重対応分析を行うためには、データを表 5 のように「0」「1」形式にすることが必要である。

関数 `corresp` を用いた多重対応分析を説明するため、まず表 5 のデータセットを作成することにする。

```
>hyou5<-matrix(c(
1,1,1,1,1,0,0,0,0,0, 0,0,0,0,0,1,1,1,1,1,
1,0,0,1,1,0,0,1,0,0, 0,1,0,0,0,1,1,0,0,1,
0,0,1,0,0,0,0,0,1,0, 0,0,1,0,0,1,0,1,1,0,
1,0,0,0,1,0,0,0,0,1, 0,1,0,1,0,0,1,0,0,0),10,8)
>colnames(hyou5)<-c("項目 1.A1","項目 1.A2",
"項目 2.B1","項目 2.B2","項目 2.B3","項目 3.C1",
"項目 3.C2","項目 3.C3")
```

作成したデータ `hyou5` の対応分析のコマンドを次に示す。

```
> (hyou5.ca<-corresp(hyou5,nf=3))
First canonical correlation(s): 0.7740774

Row scores:
      [,1]      [,2]      [,3]
[1,] -1.32217906 -0.907832620 -0.6800204
[2,] -0.55366806 1.163884128 1.3600408
[3,] 0.91728550 -1.381687982 1.1900357
[4,] -1.09566362 -0.008541258 1.3600408
[5,] -1.32217906 -0.907832620 -0.6800204
[6,] 1.04051631 0.729755881 -0.6800204
[7,] 0.19765704 1.733249149 0.3400102
[8,] 0.49851874 -0.442669504 -0.6800204
[9,] 1.66858861 -0.812302961 0.1700051
[10,] -0.02885840 0.833957787 -1.7000510

Column scores:
      [,1]      [,2]      [,3]
項目1-1 -0.8723733 -0.5905856 8.833724e-01
項目1-2 0.8723733 0.5905856 -8.833724e-01
項目2-1 -1.0468924 -0.8195340 -2.944575e-01
項目2-2 0.2117517 1.6127109 -2.944575e-01
項目2-3 1.6702814 -1.5863538 1.177830e+00
項目3-1 1.3321954 -0.6893878 -2.414710e-16
項目3-2 -1.1511410 -0.4732166 -1.766745e+00
項目3-3 -0.6251195 1.3924004 1.766745e+00
```

誌面の都合により、`biplot(hyou5.ca)` 散布図のみを図 2(第 1、2 得点の散布図)に示す。

図 2 corresp(hyou5)の biplot

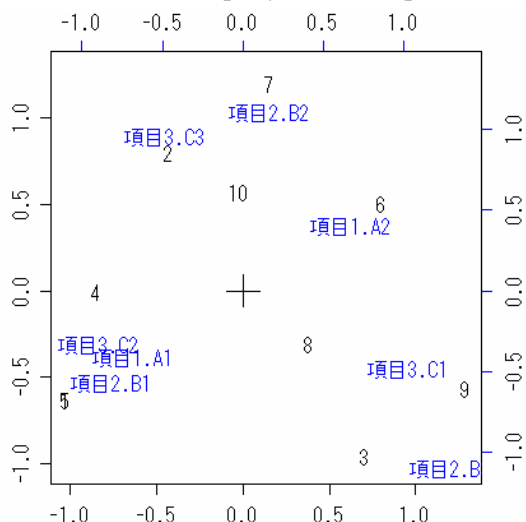


表 5 のようなデータセットを作成するのは若干面倒である。特に質問項目の中に選択肢が多い場合は、列の数が膨大になり、入力する際に間違いやすい。よって、表 4 のように集計するのが多い。そこで群馬大学社会情報学部青木繁氏は、表 4(b) のようなデータを表 5 のような「0」「1」形式に変換する関数 `make.dummy` を作成し、次のサイトで公開している。また、そのページでは R 用の数量化Ⅲ類のプログラムも公開している。ホームページから数量化Ⅲ類と関数 `corresp` の結果が一致を確認することができる。

<http://aoki2.si.gunma-u.ac.jp/R/qt3.html>

(2) 関数 `mca` を用いる場合

パッケージ MASS には、多重対応分析のための関数 `mca` がある。関数 `mca` の書き方を次に示す。

```
mca(df, nf,)
```

引数 `df` はデータフレーム、`nf` は軸の数で、デフォルトは 2 になっている。関数 `mca` は表 4(a) の形式を対象としている。

R 上でデータフレーム形式のデータセットの作成は幾つかの方法がある。そのひとつは、ま

ず空のデータフレームを作成し、データ作成エディタで各セルの値を入力する方法である。

表 4(a) のデータを `hyou4` という名前のデータフレームの作成過程を次に示す。

```
>hyou4<-data.frame()
```

```
>fix(hyou4)
```

コマンド `fix(hyou4)` を実行すると空のデータシートが開かれる。各セルにデータの入力が終わったらシートを閉じることで入力したデータが `hyou4` に格納される。

関数 `mca` によるデータセット `hyou4` の多重対応分析のコマンドおよびその結果を次に示す。

```
> (hyou4.mca <- mca(hyou4))
```

```
Call:
mca(df = hyou4)

Multiple correspondence analysis of 10 cases of 3 factors
Correlations 0.774 0.692 cumulative % explained 38.70 73.28
```

コマンド `summary(hyou4.mca)` で `hyou4.mca` に格納されている結果のリストが確認できる。正準相関は `$d`、行の得点は `$rs`、列の得点は `$cs` に記録されている。

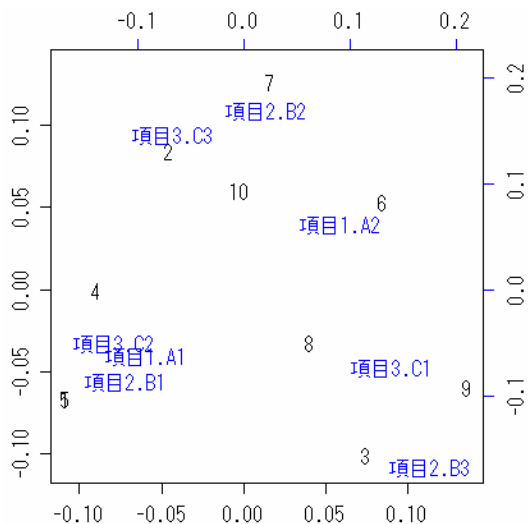
```
> hyou4.mca$rs
```

```
      1      2
1 -0.107883095 -0.0861736894
2 -0.045176338  0.0848377829
3  0.074844205 -0.1007124726
4 -0.089400586 -0.0006225889
5 -0.107883095 -0.0861736894
6  0.084900845  0.0531933287
7  0.016127811  0.1263399093
8  0.040676597 -0.0322670431
9  0.136148354 -0.0592103463
10 -0.002354699  0.0607888088
```

関数 `corresp` の計算結果と関数 `mca` の結果の各々数値は異なるが、散布図の結果は基本的には同じであることが分かる。関数 `mca` の得点の biplot を図 3 に示す。

```
> biplot(hyou4.mca$rs,hyou4.mca$cs, var.axes = FALSE)
```

図 3 `mca(hyou4)` の biplot



3. その他の対応分析パッケージ

対応分析には多くのアルゴリズムが提案されている。MASS 以外に、パッケージ `ade4` や `vegan` などに対応分析の関数がある。パッケージ `ade4` の中に同梱されている対応分析の関数を表 6 に示す。

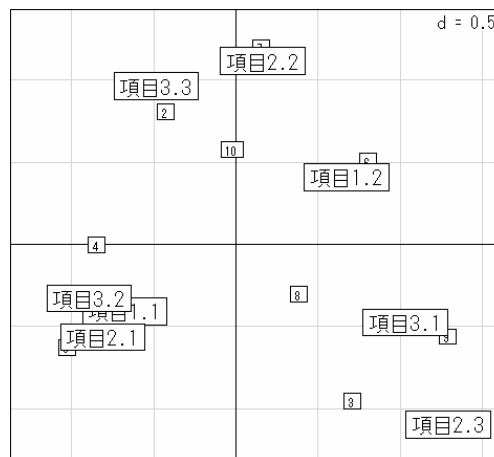
表 6 パッケージ `ade4` における対応分析の関数

関数名	名 称
<code>dudi.coa</code>	対応分析
<code>dudi.acm</code>	多重対応分析
<code>dudi.fac</code>	ファジイ対応分析
<code>dudi.nsc</code>	非対称(non symmetric)対応分析
<code>dudi.dec</code>	偏(decentered)対応分析
<code>foucart</code>	K-table の対応分析

関数 `dudi.coa` は関数 `corresp`、関数 `dudi.acm` は関数 `mca` に対応する。計算結果も基本的には同じである。ただし、関数 `dudi.coa` と関数 `dudi.acm` では正準相関の代わりに固有値を返す(`$eig` に格納されている)。得点の散布図は関数 `scatter` を用いて作成することができる。次にそのコマンド例を示す。関数 `dudi.coa` では、データフレーム形式しか扱わないことに注意が必要である。

```
>hyou5<-as.data.frame(hyou5)
> hyou5.coa<-dudi.coa(hyou5,scannf=FALSE,
nf=3)
>scatter(hyou5.coa,posieig = "none")
```

図 4 `dudi.coa(hyou5)` の第 1, 2 軸の散布図



散布図の軸は次のように指定する。

```
> scatter(hyou5.coa,xax = 1, yax = 3,posieig =
"none")#第 1 軸と 3 軸の散布図の作成
```

パッケージ `vegan` には関数 `cca`、`decorana` がある。前者 `cca` は正準対応分析(Canonical Correspondence Analysis)で、後者 `decorana` は DCA(Detrended Correspondence Analysis)である。

対応分析では、図 2~4 のように点の散布形態が馬蹄のような形になるケースが多い。この問題を馬蹄問題あるいはアーチ(arch)問題と呼ぶ。DCA は、アーチ効果を除去する対応分析の方法である。

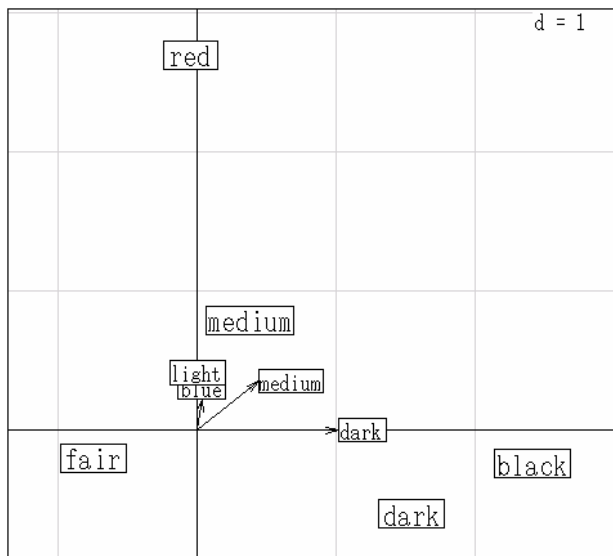
また、パッケージ `made4` には重み付きの対応分析の関数 `dudi.rwcoa`(Row weighted Correspondence Analysis)がある。

これらの詳細を説明する誌面がないので、本稿では、`decorana` とパッケージ `ade4` 中の作図関数 `s.label` および `s.arrow` による対応分析の

散布図を作成するコマンド例のみを示す。

```
>library(ade4);library(vegan);  
> caith.dca<-decorana(caith)  
> s.label(caith.dca$cproj, clab = 1.3)  
> s.arrow(caith.dca$rproj, add.pl = TRUE)
```

図 5 decorana(caith)の散布図



[1] 高根芳雄(1995):制約つき主成分分析法、朝倉書店

[2] 柳井晴夫(1994):多変量データ解析法—理論と応用—、朝倉書店