Journal of Statistical Software

May 2007, Volume 20, Issue 3. *http://www.jstatsoft.org/*

Correspondence Analysis in R, with Two- and Three-dimensional Graphics: The ca Package

Oleg Nenadić
Georg-August-Universität Göttingen

Michael Greenacre
Universitat Pompeu Fabra Barcelona

Abstract

We describe an implementation of simple, multiple and joint correspondence analysis in R. The resulting package comprises two parts, one for simple correspondence analysis and one for multiple and joint correspondence analysis. Within each part, functions for computation, summaries and visualization in two and three dimensions are provided, including options to display supplementary points and perform subset analyses. Special emphasis has been put on the visualization functions that offer features such as different scaling options for biplots and three-dimensional maps using the **rgl** package. Graphical options include shading and sizing plot symbols for the points according to their contributions to the map and masses respectively.

Keywords: biplot, correspondence analysis, graphics, joint correspondence analysis, multiple correspondence analysis, singular value decomposition.

1. Introduction

The geometric interpretation of correspondence analysis originated in the research and teaching of Jean-Paul Benzécri in France – the classic reference is Benzécri (1973). Interest in correspondence analysis increased in the late 1980s and 1990s, and simple and multiple correspondence analysis were introduced into most of the mainstream statistical software packages. In R (R Development Core Team 2007) the functions `corresp()` and `mca()` (from the **MASS** package, Venables and Ripley 2002) provide a facility for the computation of CA and MCA. However, the implementation in these functions is kept to a minimum. For example, the simple CA function does not offer the inclusion of supplementary rows or columns. In the case of MCA, the implementation offers neither supplementary points, nor more recent developments such as adjustment of eigenvalues for improved fit and corresponding adjustments of contributions, joint correspondence analysis (JCA) and subset analysis. Apart from the **XL-**

STAT software (Addinsoft 2007), none of the major programs offers these recent developments either.

This paper illustrates the implementation of CA and MCA in the R package **ca** (as of version 0.1-9, October 2006). The package comprises functions for simple, multiple and joint CA with support for subset analyses and the inclusion of supplementary variables. Furthermore, it offers functions for the graphical display of the results in two and three dimensions.

The package comprises the following components:

- Simple CA:
  - Computation: `ca()`
  - Printing and Summaries: `print()` and `summary()` methods for `ca` objects
  - Plotting: `plot.ca()` and `plot3d.ca()`
- MCA and JCA:
  - Computation: `mjca()`
  - Printing and Summaries: `print()` and `summary()` methods for `ca` objects
  - Plotting: `plot.mjca()` and `plot3d.mjca()`
- Datasets:
  - `smoke`, `author` and `wg93`

The package contains further functions, such as `iterate.mjca()` for the updating of the Burt matrix in JCA. The remaining sections describe the functions for the various forms of CA along with selected examples. Since the visualization functions are very similar for both cases, they are jointly covered in one section.

2. Simple correspondence analysis

As in principal component analysis, the idea in CA is to reduce the dimensionality of a data matrix and visualize it in a subspace of low-dimensionality, commonly two- or three-dimensional. The data of interest in simple CA are usually a two-way contingency table or any other table of nonnegative ratio-scale data for which relative values are of primary interest. The CA solution was shown by (Greenacre 1984, Chapter 2 and Appendix) to be neatly encapsulated in the singular-value decomposition (SVD) of a suitably transformed matrix. To summarize the theory, first divide the $I \times J$ data matrix, denoted by $\mathbf{N}$, by its grand total n to obtain the so-called correspondence matrix $\mathbf{P} = \mathbf{N}/n$. Let the row and column marginal totals of $\mathbf{P}$ be the vectors $\mathbf{r}$ and $\mathbf{c}$ respectively, that is the vectors of row and column masses, and $\mathbf{D}_r$ and $\mathbf{D}_c$ be the diagonal matrices of these matrices. The computational algorithm to obtain coordinates of the row and column profiles with respect to principal axes, using the SVD, is as follows:

1. Calculate the matrix of standarized residuals: $\mathbf{S} = \mathbf{D}_r^{-\frac{1}{2}}(\mathbf{P} - \mathbf{r}\mathbf{c}^\top)\mathbf{D}_c^{-\frac{1}{2}}$
2. Calculate the SVD: $\mathbf{S} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\top$ where $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$

3. Principal coordinates of rows: $\mathbf{F} = \mathbf{D}_r^{-\frac{1}{2}}\mathbf{U}\mathbf{D}_\alpha$
4. Principal coordinates of columns: $\mathbf{G} = \mathbf{D}_c^{-\frac{1}{2}}\mathbf{V}\mathbf{D}_\alpha$
5. Standard coordinates of rows: $\mathbf{X} = \mathbf{D}_r^{-\frac{1}{2}}\mathbf{U}$
6. Standard coordinates of columns: $\mathbf{Y} = \mathbf{D}_c^{-\frac{1}{2}}\mathbf{V}$

The total variance of the data matrix is measured by the inertia (see, e.g., Greenacre 2007, Chapter 4), which ressembles a chi-square statistic but is calculated on relative observed and expected frequencies:

7. Inertia $= \phi^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(p_{ij} - r_i c_j)^2}{r_i c_j}$

The rows of the coordinate matrices in (3)-(6) above refer to the rows or columns, as the case may be, of the original table, while the columns of these matrices refer to the principal axes, or dimensions, of the solution. Notice that the row and column principal coordinates are scaled in such a way that $\mathbf{F}\mathbf{D}_r\mathbf{F}^\top = \mathbf{G}\mathbf{D}_c\mathbf{G}^\top = \mathbf{D}_\alpha^2$, i.e. the weighted sum-of-squares of the coordinates on the k-th dimension (i.e., their inertia in the direction of this dimension) is equal to the *principal inertia* (or eigenvalue) α_k^2, the square of the k-th singular value, whereas the standard coordinates have weighted sum-of-squares equal to 1: $\mathbf{X}\mathbf{D}_r\mathbf{X}^\top = \mathbf{Y}\mathbf{D}_c\mathbf{Y}^\top = \mathbf{I}$.

The implementation of the algorithm follows Blasius and Greenacre (1994). The function `ca()` computes a simple CA, for example

```
R> data("smoke")
R> ca(smoke)
```

performs a simple CA on the provided `smoke` dataset (Greenacre 1984), this dataset has become a test example for CA in various software packages, and is also discussed in Greenacre (1993, 2007). This dataset contains frequencies of smoking habits (none, light, medium and heavy) for staff groups (senior managers, junior managers, senior employees, junior employees and secretaries) in a fictional company.

The output of `ca()` is controlled by the printing method for CA, i.e. the following output is given by default:

```
Principal inertias (eigenvalues):
           1        2        3
Value      0.074759 0.010017 0.000414
Percentage 87.76%   11.76%   0.49%

 Rows:
              SM        JM        SE       JE        SC
Mass     0.056995  0.093264  0.264249 0.455959  0.129534
ChiDist  0.216559  0.356921  0.380779 0.240025  0.216169
Inertia  0.002673  0.011881  0.038314 0.026269  0.006053
Dim. 1  -0.240539  0.947105 -1.391973 0.851989 -0.735456
```

```
Dim. 2  -1.935708 -2.430958 -0.106508 0.576944  0.788435
 Columns:
             none    light   medium     heavy
Mass     0.316062 0.233161 0.321244  0.129534
ChiDist  0.394490 0.173996 0.198127  0.355109
Inertia  0.049186 0.007059 0.012610  0.016335
Dim. 1  -1.438471 0.363746 0.718017  1.074445
Dim. 2  -0.304659 1.409433 0.073528 -1.975960
```

The output contains the eigenvalues and percentages of explained inertia for all possible dimensions. Additionally, values for the rows and columns (masses, chi-squared distances of points to their average, inertias and standard coordinates are given – again, see Greenacre (2007) for more details about these concepts). However, these values are restricted to two dimensions where applicable, e.g. for the standard coordinates. A list of all available entries that are returned by `ca()` is obtained with `names()`:

```
R> names(ca(smoke))
```

The output of ca is structured as a list-object, for example, the row standard coordinates are obtained with

```
R> ca(smoke)$rowcoord
```

Optional arguments for the `ca()` function include an option for setting the dimensionality of the solution (`nd`), options for marking selected rows and/or columns as supplementary ones (`suprow` and `supcol`, respectively) and options for setting subset rows and/or columns (`subsetrow` and `subsetcol`, respectively) for subset CA (Greenacre and Pardo 2006a,b). The subset option restricts the analysis to the selected subset(s) while maintaining the original margins of the table. As an extension to the printing method, a summary method is also provided. This gives a more detailed output in the classic style of Tabet (1973).

```
R> summary(ca(smoke))
```

returns the summary of the CA:

```
Principal inertias (eigenvalues):
 dim    value    %     cum%  scree plot
 1      0.074759 87.8  87.8  ************************
 2      0.010017 11.8  99.6  ***
 3      0.000414 0.5   100.1
        --------
 Total: 0.085190 100.1

Rows:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 |   SM |   57  893   31 |  -66  92   3 | -194 800 214 |
2 |   JM |   93  991  139 |  259 526  84 | -243 465 551 |
3 |   SE |  264 1000  450 | -381 999 512 |  -11   1   3 |
```

```
4 |   JE |  456 1000  308 |  233 942 331 |   58  58 152 |
5 |   SC |  130  999   71 | -201 865  70 |   79 133  81 |

Columns:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 |  non |  316 1000  577 | -393 994 654 |  -30   6  29 |
2 |  lgh |  233  984   83 |   99 327  31 |  141 657 463 |
3 |  mdm |  321  983  148 |  196 982 166 |    7   1   2 |
4 |  hvy |  130  995  192 |  294 684 150 | -198 310 506 |
```

Again, eigenvalues and relative percentages of explained inertia are given for all available dimensions. Additionally, cumulated percentages and a scree plot are shown. The items given in `Rows` and `Columns` include the principal coordinates for the first two dimensions ($k = 1$ and $k = 2$). Squared correlations (`cor`) and contributions (`ctr`) for the points are displayed next to the coordinates. Notice that the quantities in these tables are multiplied by 1000 (e.g., the coordinates and masses) which for the `cor` and `ctr` quantities means they are expressed in permills. The total quality (`qlt`) is given with respect to the dimensionality of the solution, i.e. in this case it is the sum of the squared correlations over the two included dimensions. In the case of supplementary variables, an asterisk is appended to the supplementary variable names in the output. For example, the summary of a CA on the `smoke` data where the "none" category (i.e. the first column) is treated as a supplementary one is given by:

```
R> summary(ca(smoke, supcol = 1))
```

In the corresponding section of the output the following is given:

```
Columns:
...
      name   mass  qlt  inr    k=1 cor  ctr    k=2 cor  ctr
1 | (*)non |  462   55 <NA> |  292  39 <NA> | -187  16 <NA> |
...
```

Supplementary variables have no impact on the computation. They are projected onto the solution space afterwards. Thus, contributions are not applicable for this case. Squared correlations (`cor`) as a measure of how well a point is represented by the axes are still meaningful for the case of supplementary variables and thus are included in the output.

3. Multiple and joint correspondence analysis

Multiple and Joint Correspondence Analysis (MCA and JCA, respectively) are extensions of simple CA of a single cross-tabulation to more than two categorical variables. More details about these methods can be found in Greenacre (2007), Chapters 18 and 19 respectively, while the computation of MCA and JCA is described in detail in the appendix of Nenadić and Greenacre (2006). Essentially, four approaches for the computation are considered.

The classic approach to MCA is to perform a simple CA on the indicator matrix, i.e. by performing a SVD on the matrix of standardized residuals, as shown previously, calculated on the indicator matrix. The indicator matrix $\mathbf{Z}$ is the cases$\times$variables matrix with columns

being dummy variables (with values only 0 or 1), a dummy variable for each category of the set of categorical variables. This approach yields principal inertias and principal coordinates equal to the eigenvalues and scale values in homogeneity analysis in the **Gifi** system (Michailidis and de Leeuw 1998).

An almost equivalent and more preferable approach from a CA point of view is given by performing an eigenvalue-eigenvector decomposition based on the Burt matrix, which is equal to the cross-product of the indicator matrix, $\mathbf{Z}^\top\mathbf{Z}$: i.e., the matrix which concatenates all two-way cross-tabulations between pairs of variables. Due to the structure of the Burt matrix, with submatrices on the main diagonal that are cross-tabulations of each variable with itself, the solution overestimates the total inertia.

In order to overcome this problem, two alternative approaches are considered, namely the adjustment of inertias (Greenacre 1993) and joint correspondence analysis (Greenacre 1988). The adjustment approach improves the MCA solution by rescaling the coordinates of the solution to best fit the pairwise cross-tabulations off the main diagonal of the Burt matrix. JCA is a different iterative algorithm which finds the optimal weighted least-squares fit to these off-diagonal tables.

MCA and JCA are performed with the function `mjca()`. The structure of the function is kept similar to its counterpart from simple CA. The two most striking differences are the format of the input data and the restriction to columns for the analyses. The function `mjca()` takes a response pattern matrix as input. In R terms this is a data frame with the columns containing factors. Within the function, the response pattern matrix is converted to an indicator matrix and a Burt matrix, depending on the type of analysis. The restriction to columns means that by default, only values for the columns are given in the output. Also, the specification of supplementary variables or of a subset is limited to columns. The "approach" to MCA is specified by the `lambda` option in `mjca()`:

- `lambda = "indicator"`: Analysis based on a simple CA of the indicator matrix
- `lambda = "Burt"`: Analysis based on an eigenvalue-decomposition of the Burt matrix
- `lambda = "adjusted"`: Analysis based on the Burt matrix with an adjustment of inertias
- `lambda = "JCA"`: Joint correspondence analysis

By default, `mjca()` performs an adjusted analysis, i.e. `lambda = "adjusted"`, which we believe to be the best default option, since the optimal scaling properties of MCA are conserved while raising the percentages of inertia and squared correlations to be usually very close to those one would get with a JCA. In the case of a full-blooded JCA, which involves an updating of the Burt matrix by iteratively weighted least squares, the auxiliary function `iterate.mjca()` is internally used for the updating of the Burt matrix. The updating function has two convergence criteria, namely `epsilon` and `maxit`. The option `epsilon` sets a convergence criterion by means of maximum absolute difference of the Burt matrix in an iteration step compared to the Burt matrix of the previous step. The maximum number of iterations is given by the option `maxit`. This way, the program iterates until any one of the two conditions is satisfied. Setting one option to `NA` results in that criterion being ignored – for

example, exactly 50 iterations without considering convergence are performed with `maxit=50` and `epsilon=NA`.

As with simple CA, the solution is restricted by the `nd` option to two dimensions. However, eigenvalues are given for all possible dimensions, which is equal to $(J - Q)$ for the "indicator" and "Burt" case. In the case of an adjusted analysis or a JCA, the eigenvalues are given only for those dimensions k, where the singular values from the Burt matrix λ_k (i.e., the principal inertias of the indicator matrix) satisfy the condition $\lambda_k > 1/Q$. For example, a MCA (based on an adjusted analysis) is performed with

```
R> data("wg93")
R> mjca(wg93[,1:4])
```

In this case a MCA of the first four columns of the provided dataset `wg93` (taken from the International Social Survey Programme 1993, see http://www.issp.org/) is performed. These columns contain attitudes of 871 individuals towards science and the environment (see Greenacre 2006a, Chapter 2, for more details on these data). Each category contains five possible answers (strongly agree, somewhat agree, neither agree nor disagree, somewhat disagree, strongly disagree, coded as 1 to 5). Thus, the output labels are given by appending the level names to the category names:

```
Eigenvalues:
           1        2        3        4        5        6
Value      0.076455 0.05822  0.009197 0.00567  0.001172 7e-06
Percentage 44.91%   34.2%    5.4%     3.33%    0.69%    0%

 Columns:
            A.1      A.2      A.3      A.4      A.5      B.1      B.2
Mass     0.0342   0.0924   0.0586   0.0511   0.0138   0.0204   0.0499
ChiDist  1.3434   0.6764   0.9473   1.0492   2.2149   1.8560   1.0342
Inertia  0.0102   0.0039   0.0062   0.0087   0.0114   0.0171   0.0045
Dim. 1   1.8366   0.5462  -0.4468  -1.1659  -1.9952   2.9243   0.6415
Dim. 2  -0.7275   0.2844   1.1994  -0.7368  -2.4700  -1.3701   0.6669
            B.3      B.4      B.5      C.1      C.2      C.3      C.4
Mass     0.0588   0.0807   0.0402   0.0436   0.0907   0.0565   0.0442
ChiDist  0.9333   0.7600   1.2940   1.2411   0.6881   0.9778   1.1483
Inertia  0.0046   0.0057   0.0198   0.0208   0.0042   0.0076   0.0091
Dim. 1   0.3461  -0.7141  -1.3537   2.1578   0.2468  -0.6190  -1.3489
Dim. 2   0.9639   0.2801  -2.1077  -0.9086   0.5916   1.0444  -0.6346
            C.5      D.1      D.2      D.3      D.4      D.5
Mass     0.0149   0.0172   0.0666   0.0580   0.0649   0.0433
ChiDist  2.1328   1.9159   0.8431   0.9620   0.8607   1.1366
Inertia  0.0122   0.0067   0.0020   0.0075   0.0024   0.0058
Dim. 1  -1.4676   1.2038  -0.2212  -0.3847  -0.2216   0.7078
Dim. 2  -3.0166  -1.8220   0.0069   1.1587   0.2105  -1.1518
```

Notice that the percentages of inertia do not add up to 100% in the adjusted analysis. By entering

```
R> summary(mjca(wg93[,1:4], lambda = "Burt"))
```

a summary of a MCA based on the Burt matrix is given in the same style as for simple CA:

```
Principal inertias (eigenvalues):
 dim    value    %    cum% scree plot
 1      0.209196 18.6 18.6 ************************
 2      0.185732 16.5 35   **********************
 3      0.103636 9.2  44.2 ***********
 4      0.093926 8.3  52.5 **********
 5      0.075997 6.7  59.3 ********
 6      0.063468 5.6  64.9 ******
 7      0.058835 5.2  70.1 ******
 8      0.055202 4.9  75   *****
 9      0.050836 4.5  79.5 *****
 10     0.048677 4.3  83.8 ****
 11     0.044032 3.9  87.7 ****
 12     0.038868 3.4  91.2 ***
 13     0.031642 2.8  94   **
 14     0.028599 2.5  96.5 **
 15     0.023354 2.1  98.6 *
 16     0.015687 1.4  100
        --------
 Total: 1.127687 100

Columns:
     name   mass  qlt  inr    k=1 cor ctr     k=2 cor ctr
1  |  A.1 |   34  445   55 |  840 391 115 |  -314  54  18 |
2  |  A.2 |   92  169   38 |  250 136  28 |   123  33   7 |
3  |  A.3 |   59  344   47 | -204  47  12 |   517 298  84 |
4  |  A.4 |   51  350   50 | -533 258  69 |  -318  92  28 |
5  |  A.5 |   14  401   60 | -913 170  55 | -1064 231  84 |
6  |  B.1 |   20  621   62 | 1338 519 174 |  -590 101  38 |
7  |  B.2 |   50  158   47 |  293  80  21 |   287  77  22 |
8  |  B.3 |   59  227   45 |  158  29   7 |   415 198  55 |
9  |  B.4 |   81  210   41 | -327 185  41 |   121  25   6 |
10 |  B.5 |   40  722   60 | -619 229  74 |  -908 493 179 |
11 |  C.1 |   44  732   60 |  987 632 203 |  -392 100  36 |
12 |  C.2 |   91  164   38 |  113  27   6 |   255 137  32 |
13 |  C.3 |   57  296   48 | -283  84  22 |   450 212  62 |
14 |  C.4 |   44  345   52 | -617 289  80 |  -274  57  18 |
15 |  C.5 |   15  471   60 | -671  99  32 | -1300 372 136 |
16 |  D.1 |   17  251   56 |  551  83  25 |  -785 168  57 |
17 |  D.2 |   67   14   42 | -101  14   3 |     3   0   0 |
18 |  D.3 |   58  303   48 | -176  33   9 |   499 269  78 |
19 |  D.4 |   65   25   43 | -101  14   3 |    91  11   3 |
20 |  D.5 |   43  272   50 |  324  81  22 |  -496 191  57 |
```

4. Visualizing results from CA, MCA and JCA

The graphical representation of results from CA and MCA is commonly done with so-called symmetric maps. In that case, the row and column coordinates on each axis are scaled to have inertias equal to the principal inertia along that axis: these are the principal row and column coordinates. Depending on the situation, other types of display are appropriate. This can be set with the scaling option `map` in the plotting functions for CA and MCA. Table 1 gives a brief overview over the available options and their meanings.

option	*description*
`"symmetric"`	Rows and columns in principal coordinates (default)
`"rowprincipal"`	Rows in principal and columns in standard coordinates
`"colprincipal"`	Rows in standard and columns in principal coordinates
`"symbiplot"`	Row and column coordinates are scaled to have variances equal to the singular values
`"rowgab"`	Rows in principal coordinates and columns in standard coordinates times mass
`"colgab"`	Columns in principal coordinates and rows in standard coordinates times mass (according to a proposal by Gabriel and Odoroff 1990)
`"rowgreen"`	Rows in principal coordinates and columns in standard coordinates times the square root of the mass
`"colgreen"`	Columns in principal coordinates and rows in standard coordinates times the square root of the mass (according to a proposal by Greenacre 2006b)

Table 1: Scaling options in `plot.ca` and `plot.mjca`

By default, a symmetric map is plotted. For example, a symmetric map of a CA of the `smoke` data (with the "none" smoking category set as a supplementary one) is created with the plot method for CA:

```
R> plot(ca(smoke, supcol = 1))
```

The result is shown in Figure 4. The symmetric map is not a true biplot but all other options are.

By default, supplementary variables are added to the plot with a different symbol. The symbols can be defined with the `pch` option in `plot.ca()`. This option takes four values in the following order: plotting point character or symbol for i.) active rows, ii.) supplementary rows, iii.) active columns and iv.) supplementary columns. As a general rule, options that contain entries for rows and for columns contain the entries for the rows first and than the entries for the columns. (For example, the colour of the symbols is specified with the `col` option, by default it is `col = c("#000000", "#FF0000")`, i.e. black for the rows and red for the columns.) The option `what` controls the content of the plot. It can be set to `"all"`, `"active"`, `"passive"` or `"none"` for the rows and for the columns. For example, a plot of only the active (i.e. not supplementary) points is created by using `what=c("active", "active")`. Apart from the scaling option with `map`, the plotting methods offer options for the inclusion

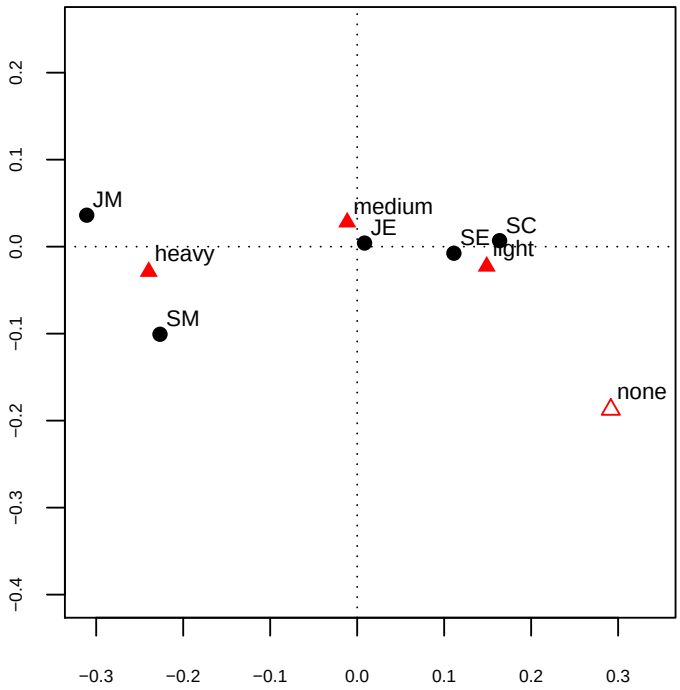

Figure 1: Symmetric map of the `smoke` dataset ("none" supplementary)

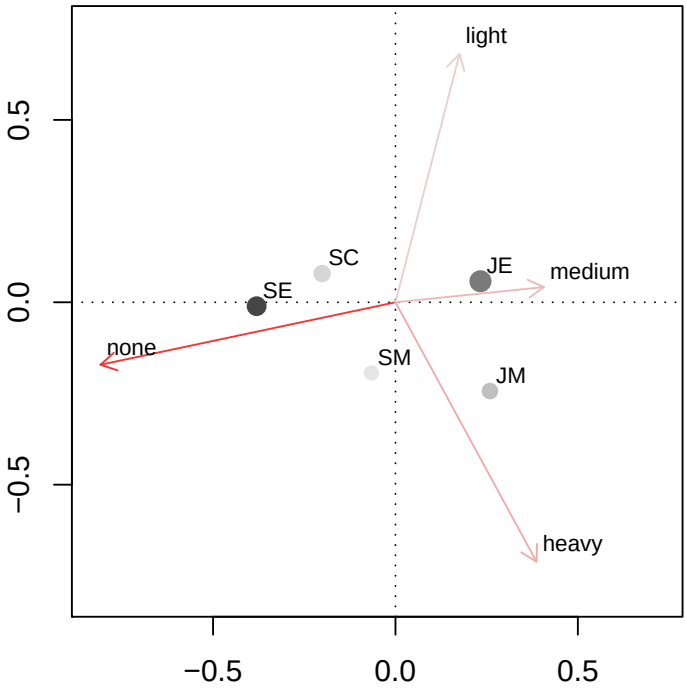

Figure 2: Standard CA biplot illustrating some graphical options of `plot.ca()`

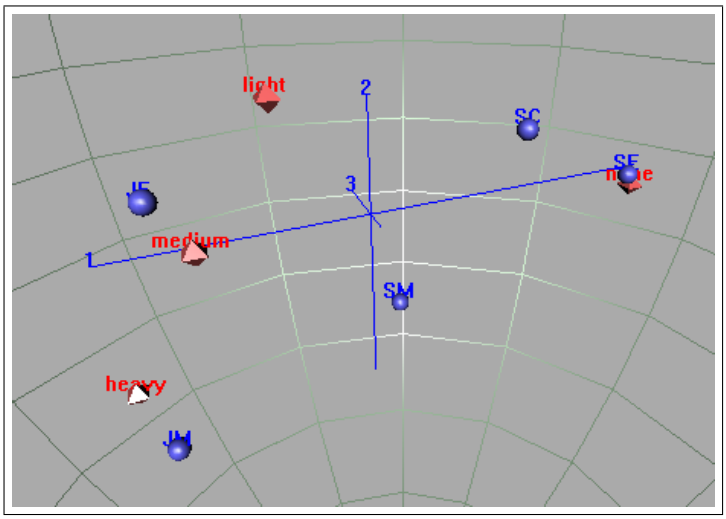

Figure 3: Three-dimensional display of a simple CA

of certain values into the plot. The option `mass` (again with the first value for the rows and the second one for the columns) sets if the mass should be indicated by the size of the point. Similarly, relative or absolute contributions can be indicated by the colour intensity in the plot by using the `contrib` option. Figure 4 shows the resulting plot of

```
R> plot(ca(smoke), mass = TRUE, contrib = "absolute",
+       map = "rowgreen", arrows = c(FALSE, TRUE))
```

Greenacre (2006b) justifies the biplot options implemented here as `"rowgreen"` and `"colgreen"` calling them "standard biplot" because they give displays which function well for low and high inertia examples. The option `dim` controls which dimensions to plot. The default value is `dim = c(1, 2)`, i.e. the first two dimensions are plotted. A plot of e.g. the second and third dimensions is obtained by setting `dim = c(2, 3)`. Another possibility for adding the third dimension to the plot is given with the functions `plot3d.ca()` and `plot3d.mjca()`. These two functions rely on the **rgl** package (Adler, Nenadić, and Zucchini 2003; Adler and Murdoch 2006) for a three-dimensional display in R. Their structure is kept similar to their counterparts for two dimensions, for example:

```
R> plot3d.ca(ca(smoke, nd=3))
```

creates a three-dimensional display of the CA. The resulting display is shown in Figure 4: This type of display offers the advantage that one can zoom and navigate using the mouse. This way, maps of two dimensions are revealed by navigating to the appropriate viewpoint, e.g. to $(0°, 0°)$ in azimuthal coordinates for the first two dimensions or $(-90°, 0°)$ for the second and third dimension.

5. Summary

We have presented the R package **ca** for simple, multiple and joint correspondence analysis. This package contains all the features of present commercially available software packages as

well as various new features that are not available elsewhere. Amongst these new features are the inclusion of adjustments in MCA, the corresponding adjustments in the percentages of inertia and squared correlations, the facility for subset analysis, joint correspondence analysis and provision of fully integrated three-dimensional graphics.

Acknowledgments

The authors thank the Fundación BBVA in Madrid and its director, Prof. Rafael Pardo, for support in this research.

References

Addinsoft (2007). **XLSTAT** – *Statistical Software for MS* **Excel**. URL http://www.xlstat.com/.

Adler D, Murdoch D (2006). **rgl***: 3D Visualization Device System (OpenGL)*. R package version 0.68, URL http://rgl.neoscientists.org/.

Adler D, Nenadić O, Zucchini W (2003). "**rgl**: A R-library for 3D Visualization with OpenGL." In "Proceedings of the 35th Symposium of the Interface: Computing Science and Statistics," Salt Lake City.

Benzécri JP (1973). *L'analyse des données. Tome 1: La taxinomie. Tome 2: L'analyse des correspondances.* Dunod, Paris.

Blasius J, Greenacre MJ (1994). "Computation of Correspondence Analysis." In MJ Greenacre, J Blasius (eds.), "Correspondence Analysis in the Social Sciences," pp. 53–75. Academic Press, London.

Gabriel KR, Odoroff C (1990). "Biplots in Biomedical Research." *Statistics in Medicine*, **9**, 469–485.

Greenacre MJ (1984). *Theory and Applications of Correspondence Analysis*. Academic Press, London.

Greenacre MJ (1988). "Correspondence Analysis of Multivariate Categorical Data by Weighted Least Squares." *Biometrika*, **75**, 457–467.

Greenacre MJ (1993). *Correspondence Analysis in Practice*. Academic Press, London.

Greenacre MJ (2006a). "From Simple to Multiple Correspondence Analysis." In MJ Greenacre, J Blasius (eds.), "Multiple Correspondence Analysis and Related Methods," pp. 41–76. Chapman & Hall/CRC, Boca Raton.

Greenacre MJ (2006b). "Tying up the Loose Ends in Simple Correspondence Analysis." *Economics Working Papers 940*, Department of Economics and Business, Universitat Pompeu Fabra. URL http://www.econ.upf.es/eng/research/onepaper.php?id=940.

Greenacre MJ (2007). *Correspondence Analysis in Practice*. Chapman & Hall/CRC, Boca Raton, second edition.

Greenacre MJ, Pardo R (2006a). "Muliple Correspondence Analysis of Subsets of Response Categories." In MJ Greenacre, J Blasius (eds.), "Multiple Correspondence Analysis and Related Methods," pp. 197–217. Chapman & Hall/CRC, Boca Raton.

Greenacre MJ, Pardo R (2006b). "Subset Correspondence Analysis: Visualizing Relationships Among a Selected Set of Response Categories from a Questionnaire Survey." *Sociological Methods and Research*, **35**, 193–218.

Michailidis G, de Leeuw J (1998). "The **Gifi** System for Descriptive Multivariate Analysis." *Statistics in Medicine*, **13**, 307–336.

Nenadić O, Greenacre MJ (2006). "Computation of Multiple Correspondence Analysis, with Code in R." In MJ Greenacre, J Blasius (eds.), "Multiple Correspondence Analysis and Related Methods," pp. 523–551. Chapman & Hall/CRC, Boca Raton.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Tabet N (1973). *Programme d'analyse des correspondances. Part of doctoral thesis, 3e cycle*. Ph.D. thesis, Université de Paris VI.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer-Verlag, New York, fourth edition. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Affiliation:

Oleg Nenadić
Institut für Statistik und Ökonometrie
Georg-August-Universität Göttingen
Platz der Göttinger Sieben 5
37073 Göttingen, Germany
E-mail: onenadi@uni-goettingen.de

Michael Greenacre
Departament d'Economia i Empresa
Universitat Pompeu Fabra
Ramon Trias Fargas, 25-27
08005 Barcelona, Spain
E-mail: michael@upf.es

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 20, Issue 3 *Submitted:* 2006-10-01
May 2007 *Accepted:* 2007-02-22