Correspondence Analysis and Multiple Correspondence Analysis

Shane T. Mueller shanem@mtu.edu

2019-04-23

Correspondence Analysis and Multiple Correspondence Analysis

Correspondance Analysis (CA) and Multiple Correspondence Analysis (MCA) are described as analogs to PCA for categorical variables. For example, it might be useful as an alternative to looking at cross-tabs for categorical variables. It can also be used in concert with inter-rater reliability to establish coding schemes that correspond with one another, or with k-means or other clustering to establish how different solutions correspond.

Resources

- Packages: `ca`
- corresp() in `MASS`
- https://www.utdallas.edu/~herve/Abdi-MCA2007-pretty.pdf
- http://gastonsanchez.com/visually-enforced/how-to/2012/10/13/MCA-in-R/
- Nenadic, O. and Greenacre, M. (2007). Correspondence analysis in R, with two- and three-dimensional graphics: The ca package. Journal of Statistical Software, 20 (3), http://www.jstatsoft.org/v20/i03/

Correspondence Analysis (CA) is available with the corresp() in MASS and the ca package, which provides better visualizations.

Background and Example

For CA, The basic problem we have is trying to understand whether two categorical variables have some sort of relationship. The problem is that, unless they are binary, correlation won't work. One could imagine some sort of matching algorithm, to find how categories best align, and then finding the sum of the diagonals. Another problem is that we may not have the same number of levels of groups–can we come up with some sort of correspondence there? This is similar to the problem we face when looking at clustering solutions and trying to determine how well a solution corresponds to some particular category we used outside of the model. The clusters don't have labels that will match the secondary variable, but we'd still like to measure how well they did.

Let's try this with the iris data, and k-means clustering

```
library(MASS)
set.seed(5220)
iris2 <- scale(iris[, 1:4])
model <- kmeans(iris[, 1:4], centers = 3)

table(iris$Species, model$cluster)
```

```
            1  2  3
  setosa     50  0  0
```

```
  versicolor  0  2 48
  virginica   0 36 14
```

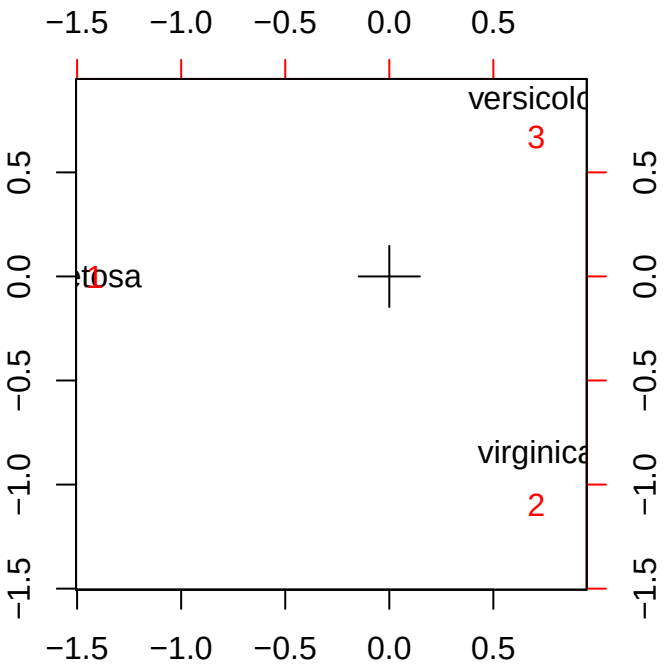
MASS library corresp function

Can we measure how good the correspondence is between our clusters and the true species?

```
library(MASS)
camodel <- corresp(iris$Species, model$cluster, nf = 2)

print(camodel)
```

```
First canonical correlation(s): 1.0000000 0.7004728

 x scores:
                [,1]          [,2]
setosa     -1.4142136  3.276967e-16
versicolor  0.7071068  1.224745e+00
virginica   0.7071068 -1.224745e+00

 y scores:
       [,1]          [,2]
1 -1.4142136  2.287696e-16
2  0.7071068 -1.564407e+00
3  0.7071068  9.588299e-01
```

```
plot(camodel)
```



The model can also be specified like this:

```
tmp <- data.frame(spec = iris$Species, cl = model$cluster)
camodel2 <- corresp(~spec + cl, data = tmp, nf = 2)
camodel2
```

```
First canonical correlation(s): 1.0000000 0.7004728
```

```
 spec scores:
                [,1]          [,2]
setosa     -1.4142136  3.276967e-16
versicolor  0.7071068  1.224745e+00
virginica   0.7071068 -1.224745e+00

 cl scores:
       [,1]          [,2]
1 -1.4142136  2.287696e-16
2  0.7071068 -1.564407e+00
3  0.7071068  9.588299e-01
```

ca library ca function

A similar result can be obtained by doing ca on the cross-table.

```
library(ca)
cmodel3 <- ca::ca(table(iris$Species, model$cluster), nd = 4)
cmodel3
```

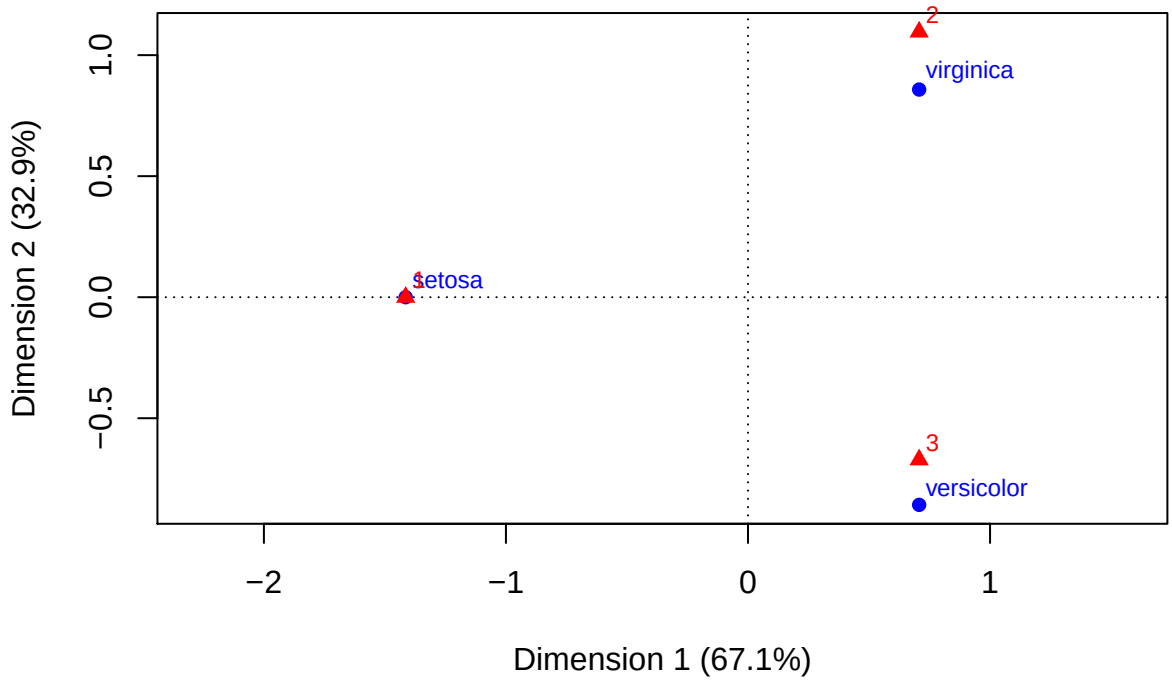
```
 Principal inertias (eigenvalues):
           1      2
Value      1      0.490662
Percentage 67.08% 32.92%


 Rows:
           setosa versicolor virginica
Mass     0.333333   0.333333  0.333333
ChiDist  1.414214   1.111752  1.111752
Inertia  0.666667   0.411998  0.411998
Dim. 1  -1.414214   0.707107  0.707107
Dim. 2   0.000000  -1.224745  1.224745


 Columns:
               1        2         3
Mass     0.333333 0.253333  0.413333
ChiDist  1.414214 1.304159  0.975240
Inertia  0.666667 0.430877  0.393118
Dim. 1  -1.414214 0.707107  0.707107
Dim. 2   0.000000 1.564407 -0.958830
```

```
summary(cmodel3)
```

```
Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      10000000  67.1  67.1  *****************
 2      0.490662  32.9 100.0  ********
        -------- -----
 Total: 1.490662 100.0
```

```
Rows:
    name   mass  qlt  inr     k=1  cor ctr    k=2 cor ctr
1 | sets |  333 1000  447 | -1414 1000 667 |    0   0   0 |
2 | vrsc |  333 1000  276 |   707  405 167 | -858 595 500 |
3 | vrgn |  333 1000  276 |   707  405 167 |  858 595 500 |

Columns:
    name   mass  qlt  inr     k=1  cor ctr    k=2 cor ctr
1 |    1 |  333 1000  447 | -1414 1000 667 |    0   0   0 |
2 |    2 |  253 1000  289 |   707  294 127 | 1096 706 620 |
3 |    3 |  413 1000  264 |   707  526 207 | -672 474 380 |
```

```
plot(cmodel3)
```



We can see that using any of these methods, we have mapped the categories into a space of principal components–using SVD. Furthermore it places both the 'rows' and 'columns' into that space, so we can see how closely aligned the groups are. We chose to use 2 dimensions in each case for easy visualization. What if the correspondence is not as good?

```
set.seed(1001)
iris2 <- scale(iris[, 1:4])
model <- kmeans(iris[, 1:4], centers = 5)

table(iris$Species, model$cluster)
```

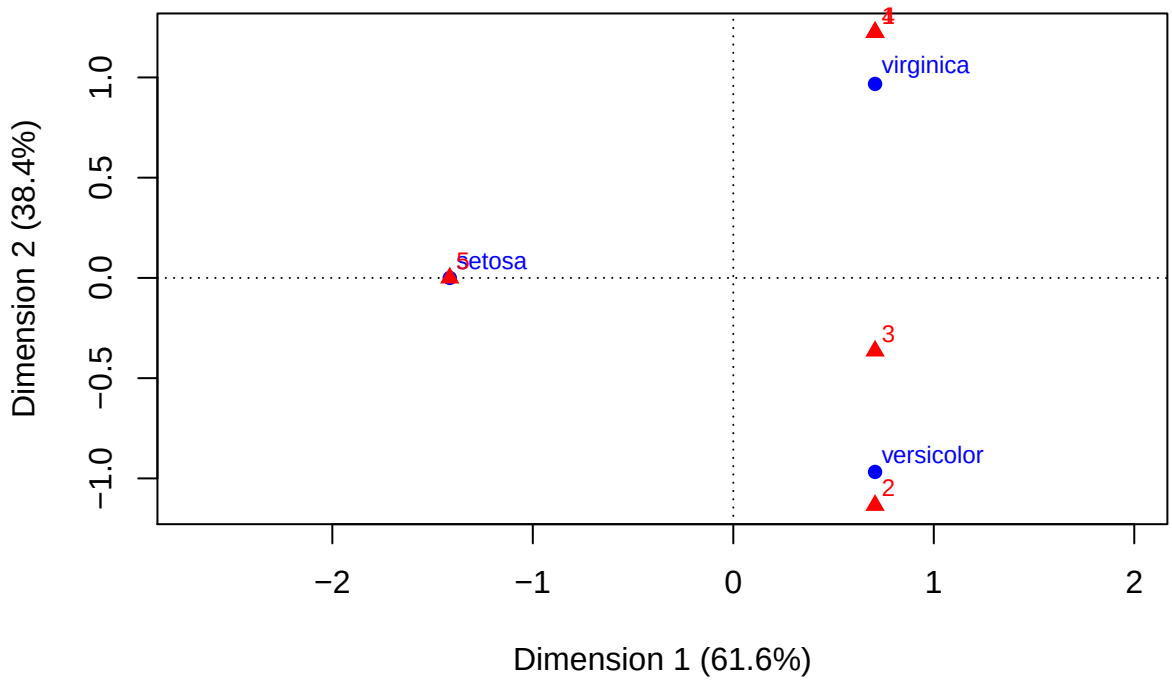
```
            1  2  3  4  5
  setosa    0  0  0  0 50
  versicolor  0 26 24  0  0
  virginica  12  1 13 24  0
```

```
cmodel5 <- ca(table(iris$Species, model$cluster), nd = 2)
```

```
cmodel5
```

```
 Principal inertias (eigenvalues):
           1      2
Value      1      0.624184
Percentage 61.57% 38.43%


 Rows:
           setosa versicolor virginica
Mass     0.333333   0.333333  0.333333
ChiDist  1.414214   1.198447  1.198447
Inertia  0.666667   0.478759  0.478759
Dim. 1  -1.414214   0.707107  0.707107
Dim. 2   0.000000  -1.224745  1.224745


 Columns:
               1         2         3        4         5
Mass    0.080000  0.180000  0.246667 0.160000  0.333333
ChiDist 1.414214  1.336416  0.795348 1.414214  1.414214
Inertia 0.160000  0.321481  0.156036 0.320000  0.666667
Dim. 1  0.707107  0.707107  0.707107 0.707107 -1.414214
Dim. 2  1.550205 -1.435375 -0.460872 1.550205  0.000000
```

```
plot(cmodel5)
```



This should match our intuition for where the two sets of labels belong.

Other Examples (from ca help file)

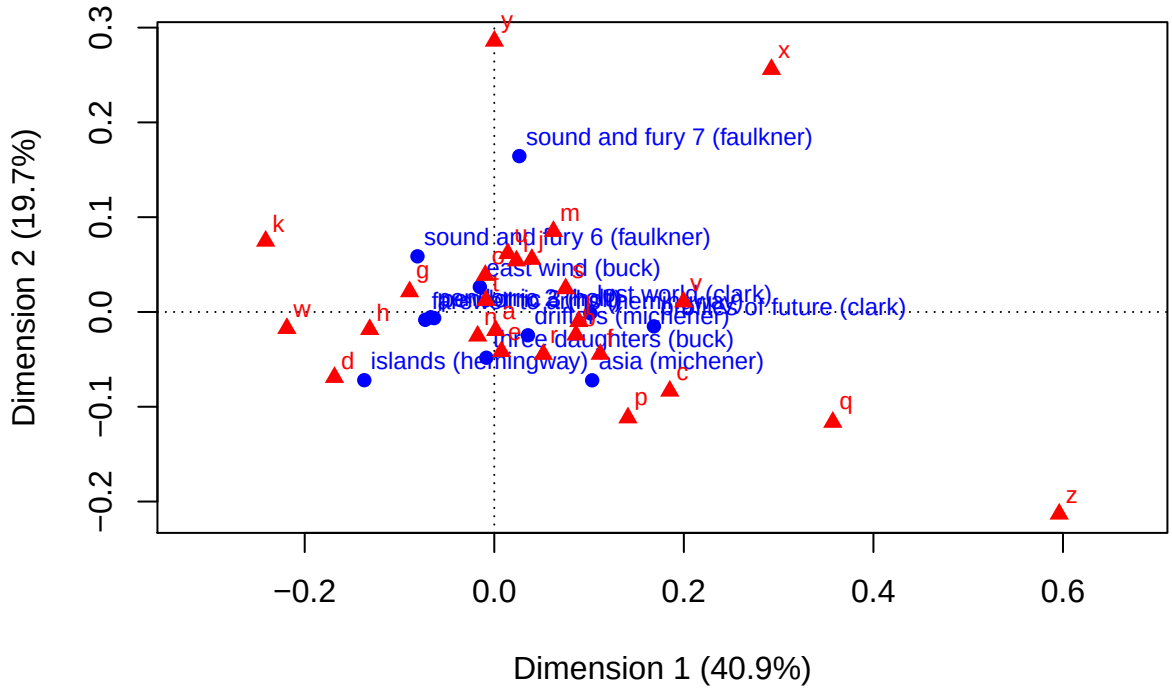
This data sets maps the distribution of letters of the alphabet to authors. These dimensions might be used to detect language, style, historic period, or something

```
data("author")
ca(author)
```

```
 Principal inertias (eigenvalues):
           1        2        3        4        5        6        7
Value      0.007664 0.003688 0.002411 0.001383 0.001002 0.000723 0.000659
Percentage 40.91%   19.69%   12.87%   7.38%    5.35%    3.86%    3.52%
           8        9        10       11
Value      0.000455 0.000374 0.000263 0.000113
Percentage 2.43%    2%       1.4%     0.6%


 Rows:
        three daughters (buck) drifters (michener) lost world (clark)
Mass                  0.085407            0.079728           0.084881
ChiDist               0.097831            0.094815           0.128432
Inertia               0.000817            0.000717           0.001400
Dim. 1               -0.095388            0.405697           1.157803
Dim. 2               -0.794999           -0.405560          -0.023114
        east wind (buck) farewell to arms (hemingway)
Mass             0.089411                     0.082215
ChiDist          0.118655                     0.122889
Inertia          0.001259                     0.001242
Dim. 1          -0.173901                    -0.831886
Dim. 2           0.434443                    -0.136485
        sound and fury 7 (faulkner) sound and fury 6 (faulkner)
Mass                        0.082310                    0.083338
ChiDist                     0.172918                    0.141937
Inertia                     0.002461                    0.001679
Dim. 1                      0.302025                   -0.925572
Dim. 2                      2.707599                    0.966944
        profiles of future (clark) islands (hemingway) pendorric 3 (holt)
Mass                       0.089722             0.082776           0.079501
ChiDist                    0.187358             0.165529           0.113174
Inertia                    0.003150             0.002268           0.001018
Dim. 1                     1.924060            -1.566481          -0.724758
Dim. 2                    -0.249310            -1.185338          -0.106349
        asia (michener) pendorric 2 (holt)
Mass           0.077827            0.082884
ChiDist        0.155115            0.101369
Inertia        0.001873            0.000852
Dim. 1         1.179548           -0.764937
Dim. 2        -1.186934           -0.091188


 Columns:
                a         b         c         d         e         f
Mass     0.079847  0.015685  0.022798  0.045967  0.127070  0.019439
```

```
ChiDist  0.048441  0.148142  0.222783  0.189938  0.070788  0.165442
                g         h         i         j         k         l         m
Mass     0.020025  0.064928 0.070092 0.000789  0.009181  0.042667 0.025500
ChiDist  0.156640  0.154745 0.086328 0.412075  0.296727  0.120397 0.159747
                n         o         p         q         r         s
Mass     0.068968  0.076572  0.015159  0.000669  0.051897 0.060660
ChiDist  0.075706  0.088101  0.250617  0.582298  0.111725 0.123217
                t         u         v         w         x         y         z
Mass     0.093010 0.029756 0.009612  0.025847 0.001160 0.021902  0.000801
ChiDist  0.050630 0.119215 0.269770  0.232868 0.600831 0.301376  0.833700
 [ reached getOption("max.print") -- omitted 3 rows ]
```

```
plot(ca(author))
```

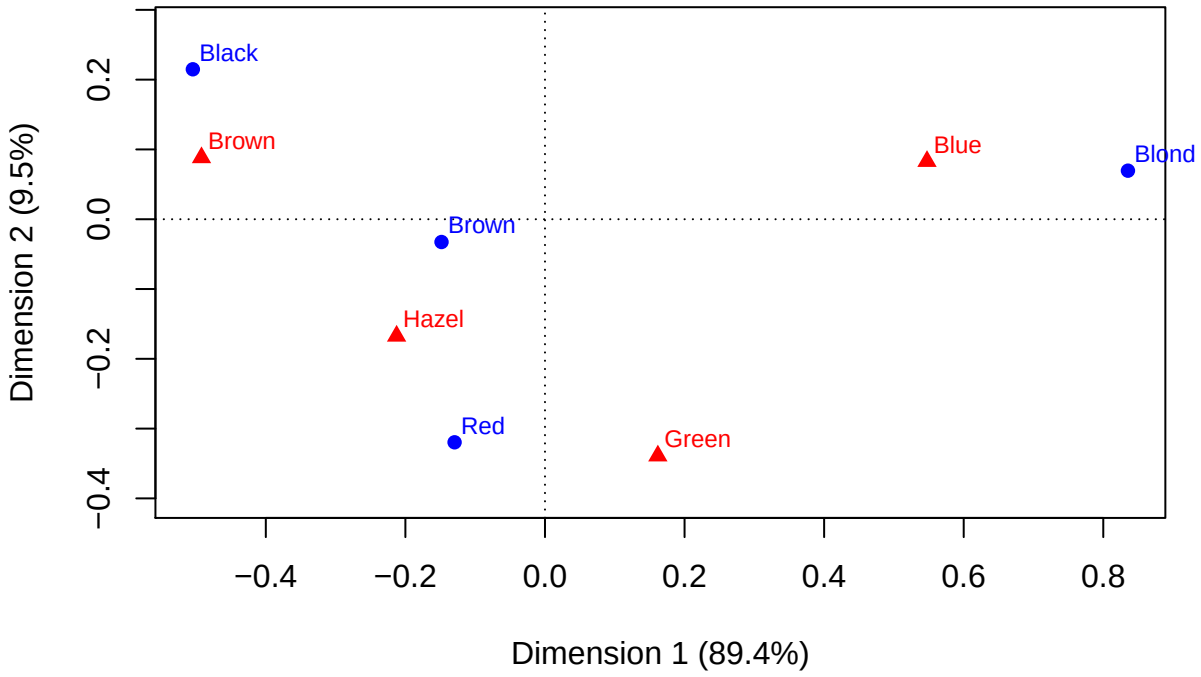



```
# table method
haireye <- margin.table(HairEyeColor, 1:2)
haireye.ca <- ca(haireye)
haireye.ca
```

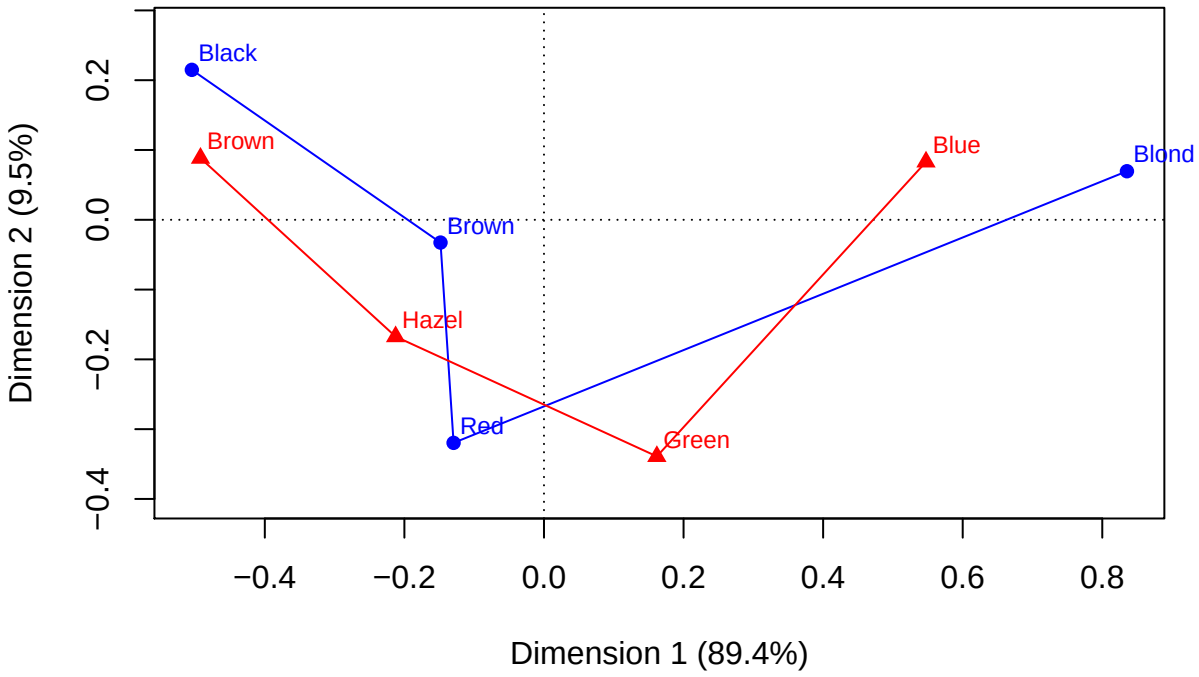
```
 Principal inertias (eigenvalues):
           1        2        3
Value      0.208773 0.022227 0.002598
Percentage 89.37%   9.52%    1.11%


 Rows:
            Black     Brown       Red    Blond
Mass     0.182432  0.483108  0.119932 0.214527
ChiDist  0.551192  0.159461  0.354770 0.838397
Inertia  0.055425  0.012284  0.015095 0.150793
Dim. 1  -1.104277 -0.324463 -0.283473 1.828229
Dim. 2   1.440917 -0.219111 -2.144015 0.466706
```

```
 Columns:
            Brown     Blue     Hazel     Green
Mass     0.371622 0.363176  0.157095  0.108108
ChiDist  0.500487 0.553684  0.288654  0.385727
Inertia  0.093086 0.111337  0.013089  0.016085
Dim. 1  -1.077128 1.198061 -0.465286  0.354011
Dim. 2   0.592420 0.556419 -1.122783 -2.274122
```

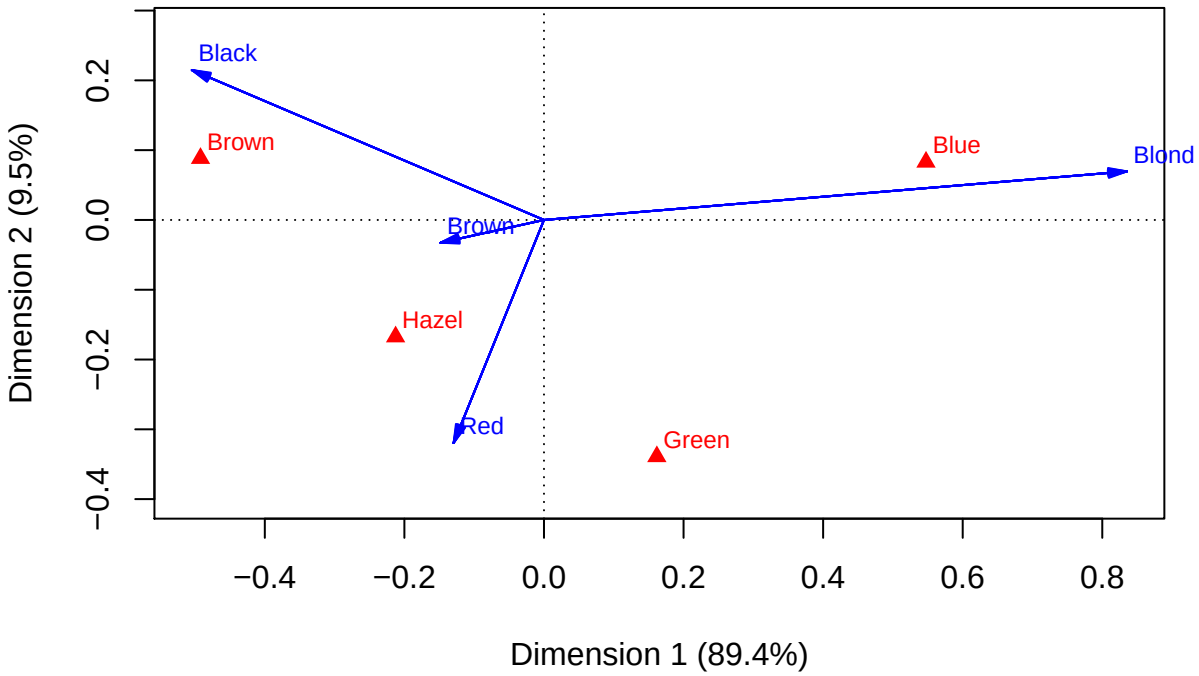
```r
plot(haireye.ca)
```




```r
# some plot options
plot(haireye.ca, lines = TRUE)
```




```
plot(haireye.ca, arrows = c(TRUE, FALSE))
```



Multiple Correspondence Analysis

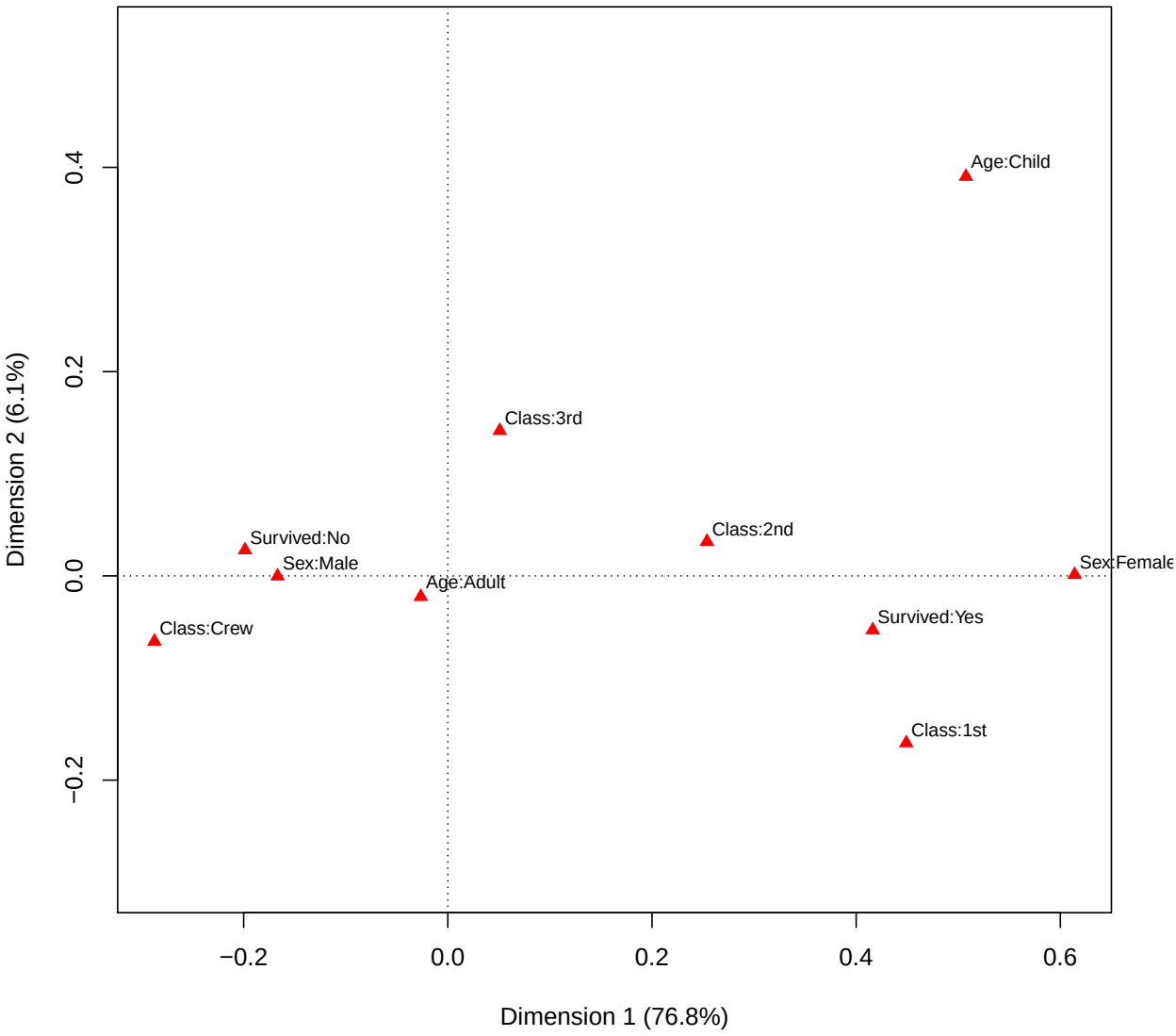
We have looked at a pair of variables, often as a contingency table, using CA. But what if you had a large number of measures that were all categorical–like a multiple-choice test. We could maybe adapt CA to tell us things more like factor analysis does for large-scale tests. In general, MCA will take the NxMxO contingency table and use SVD to map each all the dimensions into a single space, showing you where the levels of each dimension fall. Let's look at the Titanic survivor data set. Was it true that it was 'women and children first?'

Who survived? Who didn't?

```
titanicmodel <- mjca(Titanic)
print(titanicmodel)
```

```
 Eigenvalues:
           1        2        3
Value      0.067655 0.005386 0
Percentage 76.78%   6.11%    0%


 Columns:
        Class:1st Class:2nd Class:3rd Class:Crew Sex:Female  Sex:Male
Mass     0.036915  0.032372  0.080191   0.100522   0.053385  0.196615
ChiDist  1.277781  1.316778  0.749611   0.667164   1.126763  0.305938
Inertia  0.060272  0.056129  0.045061   0.044743   0.067777  0.018403
Dim. 1   1.726678  0.976191  0.195759  -1.104622   2.360505 -0.640923
Dim. 2  -2.229588  0.457212  1.937417  -0.874018   0.016164 -0.004389
        Age:Adult Age:Child Survived:No Survived:Yes
Mass     0.237619  0.012381    0.169241     0.080759
ChiDist  0.118369  2.271810    0.394352     0.826420
Inertia  0.003329  0.063898    0.026319     0.055156
Dim. 1  -0.101670  1.951309   -0.763670     1.600378
Dim. 2  -0.277601  5.327911    0.344441    -0.721825
```

```
plot(titanicmodel)
```


Age:Child
Class:3rd
Class:2nd
Survived:No
Sex:Male
Sex:Female
Age:Adult
Class:Crew
Survived:Yes
Class:1st
0.4
0.2
0.0
−0.2
Dimension 2 (6.1%)
−0.2
0.0
0.2
0.4
0.6
Dimension 1 (76.8%)