Correspondence Analysis & Related Methods

Michael Greenacre

SESSION 11: **CORRESPONDENCE ANALYSIS: theory implemented in R**

Correspondence analysis (repeat)

- Table of nonnegative data **N**
- Divide **N** by its grand total n to obtain the so-called *correspondence matrix* $\mathbf{P} = (1/n)\,\mathbf{N}$
- Let the row and column marginal totals of **P** be the vectors **r** and **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 masses

$$\mathbf{S} = \mathbf{D}_r^{-1/2}(\mathbf{P} - \mathbf{r}\mathbf{c}^\mathsf{T})\mathbf{D}_c^{-1/2} = \mathbf{D}_r^{1/2}(\mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1} - \mathbf{1}\mathbf{1}^\mathsf{T})\mathbf{D}_c^{1/2}$$

$$\frac{p_{ij} - r_i c_j}{\sqrt{r_i c_j}} \qquad \sqrt{r_i}\left(\frac{p_{ij}}{r_i c_j} - 1\right)\sqrt{c_j}$$

$$\mathbf{S} = \mathbf{U}\,\mathbf{D}_\alpha\,\mathbf{V}^\mathsf{T}$$

Principal coordinates $\mathbf{F} = \mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{D}_\alpha$, $\mathbf{G} = \mathbf{D}_c^{-1/2}\mathbf{V}\mathbf{D}_\alpha$

Standard coordinates $\mathbf{\Phi} = \mathbf{D}_r^{-1/2}\mathbf{U}$, $\mathbf{\Gamma} = \mathbf{D}_c^{-1/2}\mathbf{V}$

Parts of inertia α_k^2 $k = 1,2,\ldots$

R *implementation*

```
# read in data into data-frame data_set
# the next 14 commands are all you need to compute CA results
data.P    <- data_set/sum(data_set)
data.r    <- apply(data.P,1,sum)
data.c    <- apply(data.P,2,sum)
data.Dr   <- diag(data.r)
data.Dc   <- diag(data.c)
data.Drmh <- diag(1/sqrt(data.r))
data.Dcmh <- diag(1/sqrt(data.c))
data.P    <- as.matrix(data.P)
data.S    <- data.Drmh %*% (data.P-data.r%o%data.c) %*% data.Dcmh
data.svd  <- svd(data.S)
data.rsc  <- data.Drmh%*%data.svd$u
data.csc  <- data.Dcmh%*%data.svd$v
data.rpc  <- data.rsc%*%diag(data.svd$d)
data.cpc  <- data.csc%*%diag(data.svd$d)
# the symmetric map
plot(data.rpc[,1],data.rpc[,2],type="n",pty="s")
text(data.rpc[,1],data.rpc[,2],label=rownames(data))
# now do it in one shot using ca package (first install from CRAN)
library(ca)
plot(ca(data_set))
```

"Salud" data

(from Encuesta Nacional de Salud, España)

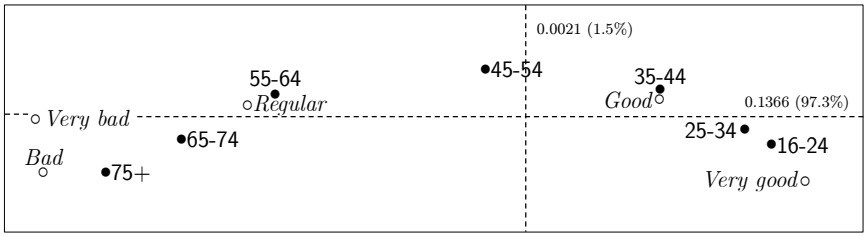
Table 1: *Crosstabulation of age groups by perceived health status*

AGE GROUP	*Very Good*	*Good*	*Regular*	*Bad*	*Very Bad*	*SUM*
16-24	243	789	167	18	6	1223
25-34	220	809	164	35	6	1234
35-44	147	658	181	41	8	1035
45-54	90	469	236	50	16	861
55-64	53	414	306	106	30	909
65-74	44	267	284	98	20	713
75+	20	136	157	66	17	396
SUM	817	3542	1495	414	103	6371

Table 2: *Row percentages calculated from Table 1*

AGE GROUP	*Very Good*	*Good*	*Regular*	*Bad*	*Very Bad*	*SUM*
16-24	19.9	64.5	13.7	1.5	0.5	100.0
25-34	17.8	65.6	13.3	2.8	0.5	100.0
35-44	14.2	63.6	17.5	4.0	0.8	100.0
45-54	10.5	54.5	27.4	5.8	1.9	100.0
55-64	5.8	45.5	33.7	11.7	3.3	100.0
65-74	6.2	37.4	39.8	13.7	2.8	100.0
75+	5.1	34.3	39.6	16.7	4.3	100.0
AVERAGE	12.8	55.6	23.5	6.5	1.6	100.0

Symmetric CA map



Numerical scale from CA solution (principal coordinates)

	very bad	bad	regular	good	very good
original scale:	-0.767	-0.755	-0.439	0.198	0.423

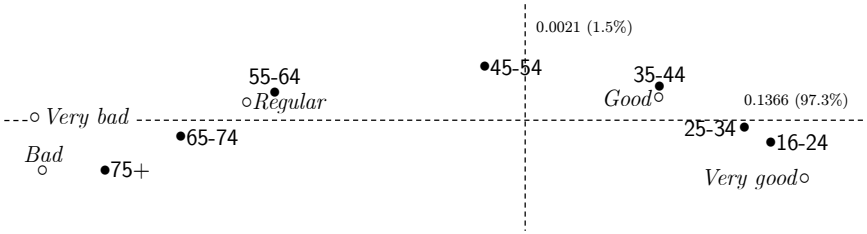
Any linear transformation still retains optimality of results.

So to convert to 0 to 100 scale (for example):

- first add `0.767` ***to all values so the scale runs from*** `0` ***to*** `0.423+0.767= 1.190`

- multiply by `100/1.190` ***so the scale runs from*** `0` ***to*** `100`

CA of perceived health status - optimal scale



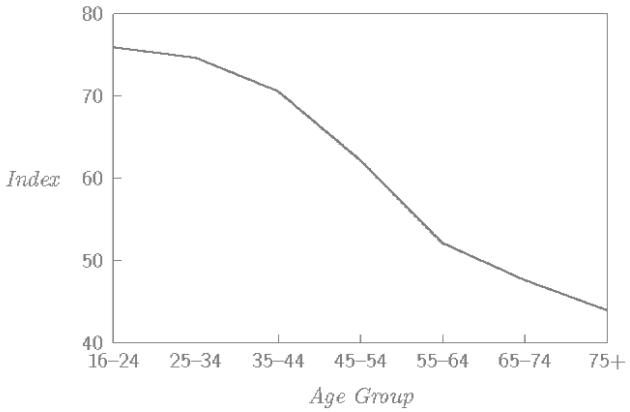
Redefining the optimal scale

	very bad	bad	regular	good	very good
original scale:	-0.767	-0.755	-0.439	0.198	0.423
new scale:	0.0	1.0	27.6	81.1	100.00

Calculating averages for the age groups

16-24	25-34	35-44	45-54	55-64	65-74	75+
75.97	74.69	70.63	62.25	52.17	47.67	44.01

CA of perceived health status - optimal scale



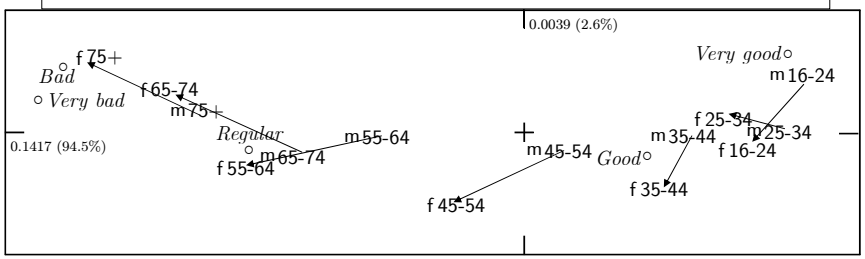

```
# rescaling the optimal scale
data.range <- max(data.csc[,1])-min(data.csc[,1])
data.scale <- (data.csc[,1]-min(data.csc[,1]))*100/data.range
```

Perceived health status: male - female

Table 3: *Age group and sex interactively crosstabulated with health status*

AGE GROUP	*Very Good*	*Good*	*Regular*	*Bad*	*Very Bad*	*SUM*
MALES						
16-24	145	402	84	5	3	639
25-34	112	414	74	13	2	615
35-44	80	331	82	24	4	521
45-54	54	231	102	22	6	415
55-64	30	219	119	53	12	433
65-74	18	125	110	35	4	292
75+	9	67	65	25	8	174
FEMALES						
16-24	98	387	83	13	3	584
25-34	108	395	90	22	4	619
35-44	67	327	99	17	4	514
45-54	36	238	134	28	10	446
55-64	23	195	187	53	18	476
65-74	26	142	174	63	16	421
75+	11	69	92	41	9	222
SUM	817	3542	1495	414	103	6371

Perceived health status: male - female



Correspondence Analysis & Related Methods

Michael Greenacre

SESSION 12: **CORRESPONDENCE ANALYSIS: three examples, using R package ca**

R *package* **ca**

The **ca** package (downloadable from CRAN) consists of two main functions **`ca`** (simple CA) and **`mjca`** (multiple and joint CA), as well as two- and three-dimensional plotting routines (the package **rgl** needs to be installed as well for the 3-d graphics)

```
ca(ca)

The function ca computes a simple correspondence analysis based on the singular value
decomposition.

The options suprow and supcol allow supplementary (passive) rows and columns to be
specified. Using the options subsetrow and/or subsetcol result in a subset CA being
performed.

Usage

ca(obj, nd = NA, suprow = NA, supcol = NA, subsetrow = NA, subsetcol = NA, swisign = NA)

Arguments

obj       A two-way table of non-negative data, usually frequencies.
nd        Number of dimensions to be included in the output; if NA the maximum possible
          dimensions are included.
suprow    Indices of supplementary rows.
supcol    Indices of supplementary columns.
subsetrow Row indices of subset.
subsetcol Column indices of subset.
swisign   A vector of 1's and -1's indicating for which dimension to switch the
          coordinates.
```

Value of **ca** *object*

```
Value

sv          Singular values
nd          Dimenson of the solution
rownames    Row names
rowmass     Row masses
rowdist     Row chi-square distances to centroid
rowinertia  Row inertias
rowcoord    Row standard coordinates
rowsup      Indices of row supplementary points
colnames    Column names
colmass     Column masses
coldist     Column chi-square distances to centroid
colinertia  Column inertias
colcoord    Column standard coordinates
colsup      Indices of column supplementary points
```

Options for plotting used so far

```
plot(x, dim = c(1,2), map = "symmetric", what = c("all", "all"), mass = c(FALSE, FALSE),
     contrib = c("none", "none"), col = c("#000000", "#FF0000"), pch = c(16, 1, 17, 24),
     labels = c(2, 2), arrows = c(FALSE, FALSE), ...)

Arguments

x         Simple correspondence analysis object returned by ca
dim       Numerical vector of length 2 indicating the dimensions to plot on horizontal and
          vertical axes respectively; default is 1st dimension horizontal and 2nd vertical
map       Character string specifying the map type. Allowed options include
                    "symmetric" (default)
                    "rowprincipal"
                    "colprincipal"
                    "symbiplot"
                    "rowgab"
                    "colgab"
                    "rowgreen"
                    "colgreen"
what      Vector of two character strings specifying the contents of the plot. First entry
          sets the rows and the second entry the columns. Allowed values are
                    "all" (all available points, default)
                    "active" (only active points are displayed)
                    "passive" (only supplementary points are displayed)
                    "none" (no points are displayed)
mass      Vector of two logicals specifying if the mass should be represented by the area
          of the point symbols (first entry for rows, second one for columns)
labels    Vector of length two specifying if the plot should contain symbols only (0), labels
          only (1) or both symbols and labels (2). Setting labels to 2 results in the symbols
          being plotted at the coordinates and the labels with an offset.
arrows    Vector of two logicals specifying if the plot should contain points (FALSE, the
          default) or arrows (TRUE). First value sets the rows and the second value sets the
          columns.
...       Further arguments passed to plot and points.
```