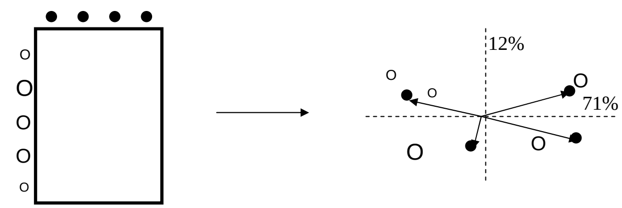
Correspondence Analysis & Related Methods

Michael Greenacre

SESSION 13: **Diagnostics, contributions in weighted PCA and Correspondence Analysis**

Inertia contributions in weighted PCA

- PCA is a method of data visualization which represents the true positions of points in a map which comes closest to all the points, closest in sense of weighted least-squares.

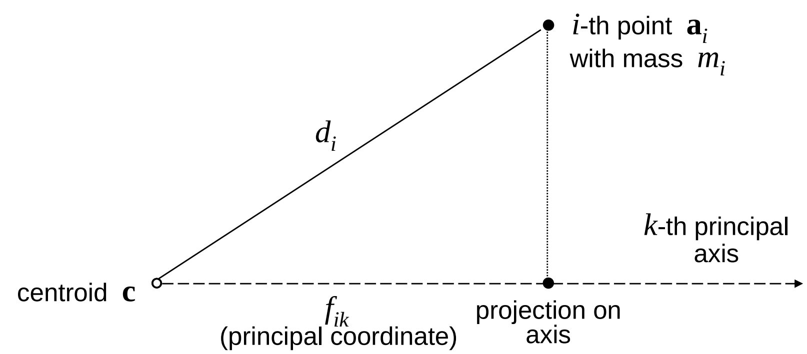


- The inertia (weighted variance) explained in the map applies to all the points: if we say 83% of the inertia is explained in the map, 71% on the first dimension and 12% on the second, this is a figure calculated for all row (or column) points together.

Inertia contributions in weighted PCA

- This type of "inertia-explained-by-axes" calculation can be made for individual points.
- These more detailed results are aids to interpretation in the form of numerical diagnostics, called contributions.
- Especially when there is not a high percentage of inertia explained by the map, these contributions will help us to identify points which are represented inaccurately.
- The inertias and their percentages tell us how much of the variance in the table is explained by the principal axes. The contributions do the same, but for each point individually, and help us to see:
  (a) which points are being explained better than others;
  (b) which points are contributing to the solution more than others.

Geometry of inertia contributions



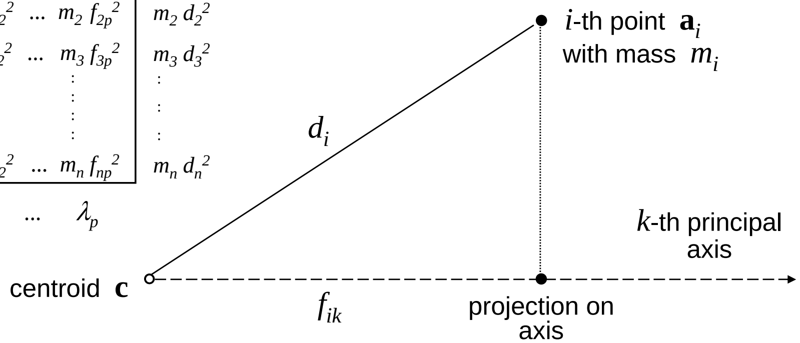
Total inertia of the cloud of points $= \Sigma_i\, m_i\, d_i^2 = \Sigma_i\, m_i\, \Sigma_k\, f_{ik}^2 = \Sigma_k\, \lambda_k$

Inertia of i-th point $= m_i\, d_i^2 = m_i\, \Sigma_k\, f_{ik}^2$

Inertia contribution of i-th point to k-th axis $= m_i\, f_{ik}^2$

Decomposition of inertia

$$
\begin{array}{c|cccc|c}
 & \multicolumn{4}{c|}{\textbf{Axes}} & \\
 & 1 & 2 & \dots & p & \\
\hline
\mathbf{1} & m_1 f_{11}^2 & m_1 f_{12}^2 & \dots & m_1 f_{1p}^2 & m_1 d_1^2 \\
\mathbf{2} & m_2 f_{21}^2 & m_2 f_{22}^2 & \dots & m_2 f_{2p}^2 & m_2 d_2^2 \\
\mathbf{3} & m_3 f_{31}^2 & m_3 f_{32}^2 & \dots & m_3 f_{3p}^2 & m_3 d_3^2 \\
 & \vdots & \vdots & & \vdots & \vdots \\
\boldsymbol{n} & m_n f_{n1}^2 & m_n f_{n2}^2 & \dots & m_n f_{np}^2 & m_n d_n^2 \\
\hline
 & \lambda_1 & \lambda_2 & \dots & \lambda_p &
\end{array}
$$



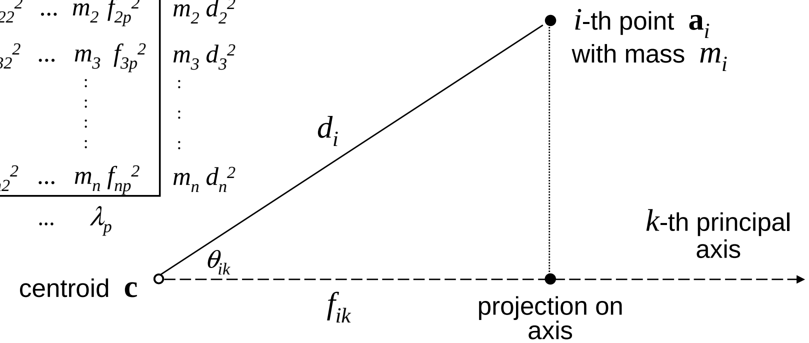
Total inertia of the cloud of points $= \sum_i m_i d_i^2 = \sum_i m_i \sum_k f_{ik}^2 = \sum_k \lambda_k$

Inertia of i-th point $= m_i d_i^2 = m_i \sum_k f_{ik}^2$

Inertia contribution of i-th point to k-th axis $= m_i f_{ik}^2$

Inertia contributions

$$
\begin{array}{c|cccc|c}
 & \multicolumn{4}{c|}{\textbf{Axes}} & \\
 & 1 & 2 & \dots & p & \\
\hline
\mathbf{1} & m_1 f_{11}^2 & m_1 f_{12}^2 & \dots & m_1 f_{1p}^2 & m_1 d_1^2 \\
\mathbf{2} & m_2 f_{21}^2 & m_2 f_{22}^2 & \dots & m_2 f_{2p}^2 & m_2 d_2^2 \\
\mathbf{3} & m_3 f_{31}^2 & m_3 f_{32}^2 & \dots & m_3 f_{3p}^2 & m_3 d_3^2 \\
 & \vdots & \vdots & & \vdots & \vdots \\
\boldsymbol{n} & m_n f_{n1}^2 & m_n f_{n2}^2 & \dots & m_n f_{np}^2 & m_n d_n^2 \\
\hline
 & \lambda_1 & \lambda_2 & \dots & \lambda_p &
\end{array}
$$



$m_i f_{ik}^2 / \lambda_k$: amount of inertia of axis k explained by point i (*contribution*, **CTR**)

$m_i f_{ik}^2 / m_i d_i^2$: amount of inertia of point i explained by axis k (*squared correlation*, **COR**)

$m_i f_{ik}^2 / m_i d_i^2 = f_{ik}^2 / d_i^2$, i.e. the square of $f_{ik} / d_i = cos(\theta_{ik})$, where θ_{ik} is the angle point-axis

Inertia contributions for CA of "author"

col	name	mass	qlt	inr	k=1	cor	ctr	k=2	cor	ctr
1	a	80	162	10	2	1	0	-19	161	8
2	b	16	365	18	86	338	15	-24	27	2
3	c	23	831	60	185	691	102	-83	140	43
4	d	46	920	89	-169	788	170	-69	132	59
5	e	127	357	34	8	12	1	-42	345	60
6	f	19	529	28	112	456	32	-45	72	10
7	g	20	344	26	-89	325	21	21	19	2
8	h	65	735	83	-131	721	146	-18	14	6
9	i	70	465	28	23	74	5	54	392	55
10	j	1	28	7	40	9	0	56	18	1
11	k	9	724	43	-241	661	70	75	64	14
12	l	43	555	33	89	548	44	-10	7	1
13	m	26	436	35	62	153	13	85	284	50
14	n	69	166	21	-18	54	3	-25	112	12
15	o	77	205	32	-9	12	1	39	193	31
16	p	15	515	51	141	317	39	-112	198	51
17	q	1	416	12	357	376	11	-116	40	2
18	r	52	374	35	52	215	18	-45	159	28
19	s	61	413	49	75	374	45	25	40	10
20	t	93	90	13	-9	30	1	12	59	4
21	u	30	283	23	14	14	1	62	268	31
22	v	10	550	37	200	548	50	11	2	0
23	w	26	888	75	-219	883	161	-17	6	2
24	x	1	418	22	292	237	13	256	182	21
25	y	22	899	106	0	0	0	286	899	485
26	z	1	576	30	596	511	37	-213	65	10

<u>Summary:</u> Contributions to inertia

- Each principal inertia can decomposed into parts due to each point, either row points or column points. These contributions explain how each principal axis has been constructed (hence the influence of each point in defining the dimension).
- The inertia of a point is similarly decomposed over all the axes, thanks to using Euclidean-type distance and Pythagoras' theorem. Each component on an axis can be expressed relative to the point inertia and this is the same as the squared cosine (i.e., squared correlation) between the point and the axis. These values can be added over axes and tell you how well the point is represented in the solution space.

R *implementation of CA (repeat)*

```
# 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))
```

Computation of contributions

```
# compute matrix of contributions for rows and inertias
data.rcon <- data.rpc^2 * data.r
apply(data.rcon, 1, sum)
# compute contributions and squared correlations
data.rctr <- t( t(data.rcon) / apply(data.rcon, 2, sum) )
data.rcor <- data.rcon / apply(data.rcon, 1, sum)
# compute qualities in 2-d solution
apply(data.rcor[,1:2], 1, sum)

# compute matrix of contributions for columns and inertias
data.ccon <- data.cpc^2 * data.c
apply(data.ccon, 1, sum)
# compute contributions and squared correlations
data.cctr <- t( t(data.ccon) / apply(data.ccon, 2, sum) )
data.ccor <- data.ccon / apply(data.ccon, 1, sum)
# compute qualities in 2-d solution
apply(data.ccor[,1:2], 1, sum)
```

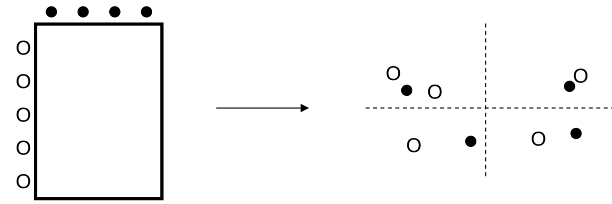
Correspondence Analysis & Related Methods

Michael Greenacre

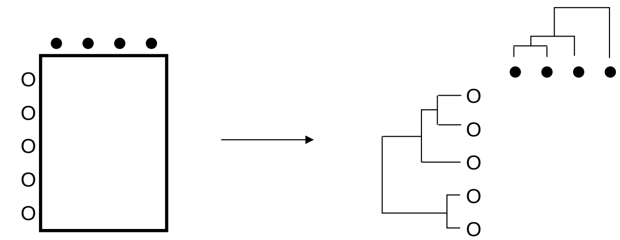
SESSION 14: **1. CORRESPONDENCE ANALYSIS & CLUSTER ANALYSIS**

2. CORRESPONDENCE ANALYSIS & BIPLOT

- Correspondence analysis (CA) is a method of data visualization that reveals continuous structures (the dimensions)



- But in our search for structure in the table we can also consider clustering the rows and columns, to reveal discrete structures (the clusters, or classes):



A simple example

- 988 students, males and females classified each according to their parents having been or not to university, cross-tabulated with their choice of studies at high school

	NS	*MA*	*LS*	*PS*
F_no	94	43	197	61
F_uni	28	17	103	37
M_no	65	19	51	132
M_uni	17	9	30	85

Inertia = 0.1848   Chi-square = 182.6

Which two rows can we merge so that inertia (or chi-square) is reduced the *least*?

F_no and F_uni: reduces inertia by 0.0070

- **F_no**: female, parents no university **F_uni**: female, parents university
  M_no: male, parents no university **M_uni**: male, parents university
- ***NS***: non-science ***MA***: mathematics ***LS***: life sciences ***PS***: physical sciences

A simple example

- 988 students, males and females classified each according to their parents having been or not to university, cross-tabulated with their choice of studies at high school

	NS	*MA*	*LS*	*PS*
{F_no, F_uni}	122	60	300	98
M_no	65	19	51	132
M_uni	17	9	30	85

Inertia = 0.1778

Which two rows can we merge so that inertia is reduced the *least*?

M_no and M_uni: reduces inertia by 0.0104

- **F_no**: female, parents no university **F_uni**: female, parents university
  M_no: male, parents no university **M_uni**: male, parents university
- ***NS***: non-science ***MA***: mathematics ***LS***: life sciences ***PS***: physical sciences

A simple example

- 988 students, males and females classified each according to their parents having been or not to university, cross-tabulated with their choice of studies at high school

	NS	MA	LS	PS
{F_no, F_uni}	122	60	300	98
{M_no, M_uni}	83	28	81	217

Inertia = 0.1674

Which two rows can we merge so that inertia (or chi-square) is reduced the *least*?

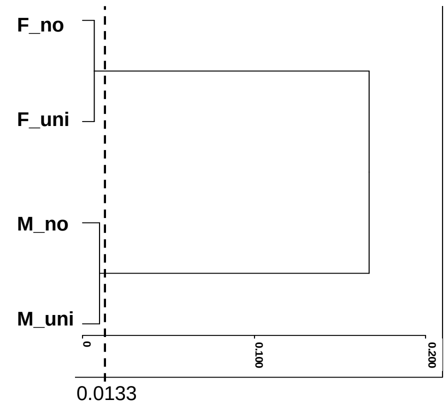
Only two rows left to merge and this reduces inertia by 0.1674

- **F_no**: female, parents no university **F_uni**: female, parents university
  M_no: male, parents no university **M_uni**: male, parents university
- ***NS***: non-science ***MA***: mathematics ***LS***: life sciences ***PS***: physical sciences

A simple example

- 988 students, males and females classified each according to their parents having been or not to university, cross-tabulated with their choice of studies

	NS	MA	LS	PS
F_no	94	43	197	61
F_uni	28	17	103	37
M_no	65	19	51	132
M_uni	17	9	30	85



From Greenacre(1993:118), the critical point for the chi-square is 13.11, that is for the inertia: 13.11/988 = **0.0133**. This gives multiple comparison test for differences between rows.

Ward clustering

- The type of clustering performed by this procedure of "minimizing the reduction of inertia at each step" is called Ward clustering (see our earlier classes on cluster analysis)
- Ward clustering is a hierarchical clustering analysis which needs:

  (a) description vectors of objects to be clustered

  (b) weights for each object

  If you prefer to have a "distance" criterion for clustering, this is it:

$$d(G_1, G_2) = \frac{r_1 r_2}{r_1 + r_2} \| prof_1 - prof_2 \|_c^2$$

Masses of clusters G_1 and G_2

Chi-square distance

Profiles of clusters G_1 and G_2

- We want to perform Ward clustering on the profiles, with weights equal to the masses.
- Since Ward clustering calculates Euclidean distances between vectors, we would need to prepare the profiles so that the Euclidean distances will be chi-squared: that is, we have to divide the profile elements by the square roots of their average (expected) values.
- But we need to weight the points: use XLSTAT or Fionn Murtagh's R code: http://astro.u-strasbg.fr/~fmurtagh/mda-sw/correspondances

Biplot

- Correspondence analysis is based on the SVD of

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

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

- We want the right hand side in the form of scalar products between the coordinate matrices

$$\mathbf{D}_r^{-1}(\mathbf{P} - \mathbf{r}\mathbf{c}^\mathsf{T})\mathbf{D}_c^{-1} = \mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{D}_\alpha(\mathbf{D}_c^{-1/2}\mathbf{V})^\mathsf{T} = \mathbf{F}\mathbf{\Gamma}^\mathsf{T}$$

In full space: $\dfrac{p_{ij} - r_i c_j}{r_i c_j} = f_{i1}\gamma_{j1} + f_{i2}\gamma_{j2} + \cdots$

In reduced space: (e.g., 2-d) $\dfrac{p_{ij} - r_i c_j}{r_i c_j} \approx f_{i1}\gamma_{j1} + f_{i2}\gamma_{j2}$

$$\left(\frac{p_{ij}}{r_i} - c_j\right) / c_j \approx f_{i1}\gamma_{j1} + f_{i2}\gamma_{j2}$$

row profile element **average profile element**

scalar product between row profile and column vertex

Biplot variations by Gabriel & Greenacre

(repeat) $$\left(\frac{p_{ij}}{r_i} - c_j\right) / c_j \approx f_{i1}\gamma_{j1} + f_{i2}\gamma_{j2}$$

row profile element **average profile element**

scalar product between row profile and column vertex

- Gabriel's modification:

$$\left(\frac{p_{ij}}{r_i} - c_j\right) \approx f_{i1} c_j \gamma_{j1} + f_{i2} c_j \gamma_{j2}$$

deviation of profile from average

vertices (standard coordinates) "shrunk by their respective masses

- Greenacre's modification (the standard biplot):

$$\left(\frac{p_{ij}}{r_i} - c_j\right) / c_j^{1/2} \approx f_{i1} c_j^{1/2} \gamma_{j1} + f_{i2} c_j^{1/2} \gamma_{j2}$$

standardized deviation of profile from average

vertices shrunk by <u>square roots</u> of their respective masses; squares of these rescaled column coordinates are exactly the (relative) contributions of the column to the respective dimension

- (Relative) column contribution:

$$c_j g_{jk}^2 / \lambda_k = c_j (\lambda_k^{1/2} \gamma_{jk})^2 / \lambda_k = c_j \gamma_{jk}^2$$