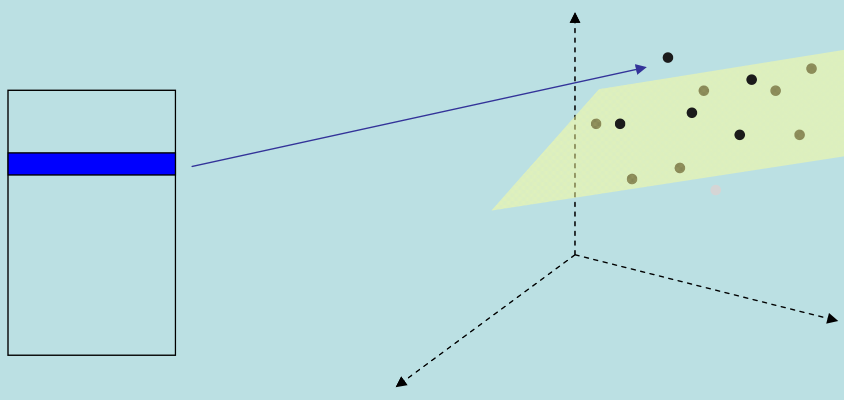
(updated version of…)

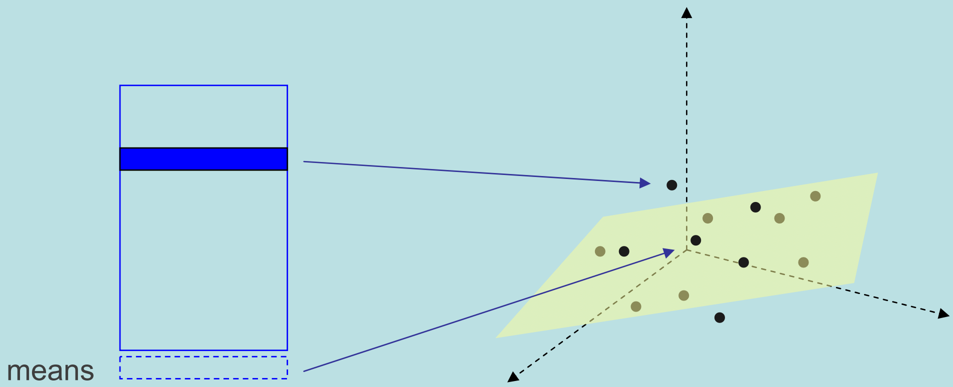
Singular value decomposition

Generalized principal component analysis

Generalized PCA

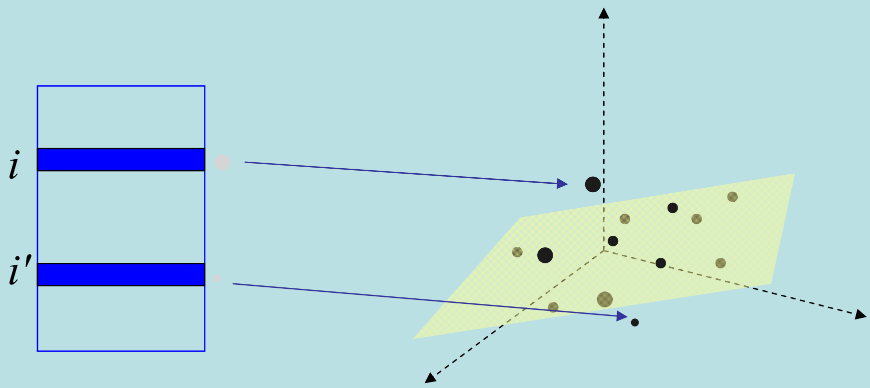


Generalized PCA



- data centred

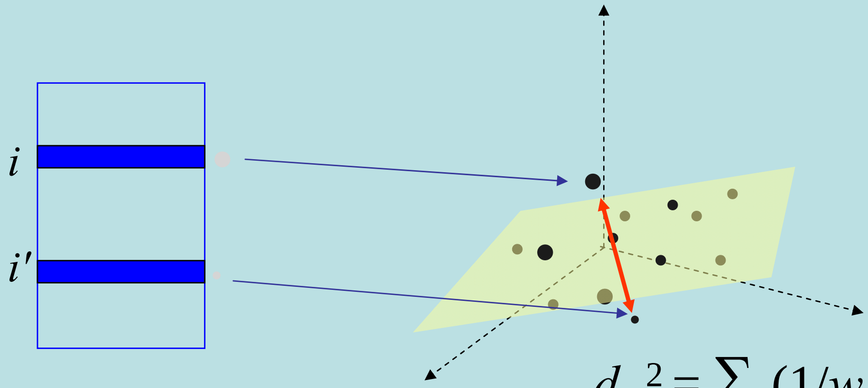
Generalized PCA



- data centred
- points weighted (row masses)

– in case of frequency data, points are weighted by their row masses, that is the relative frequencies of each row (i.e. proportional to sample sizes)

Generalized PCA

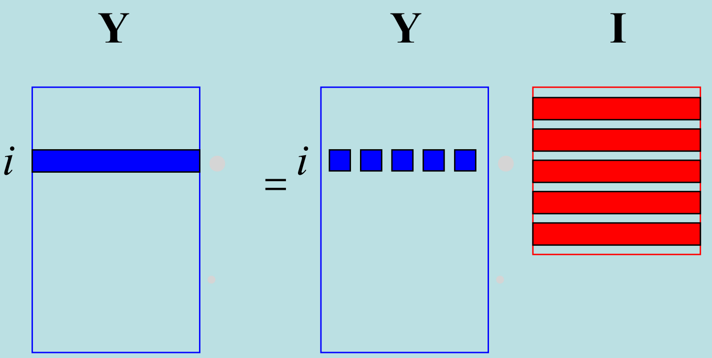


$$d_{ii'}^2 = \Sigma_j\,(1/w_{ij})(y_{ij} - y_{i'j})^2$$

- data centred
- points weighted (row masses)
- metric weighted (column weights)

e.g. $1/w_j = 1/\sigma_j^2$ the inverse of the variance (PCA)
$1/w_j = 1/c_j$ the inverse of the mean (CA)

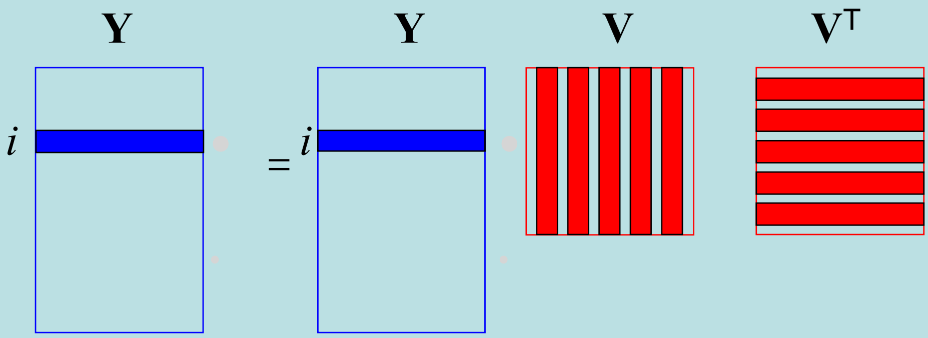
Basic vector geometry



	C1	C2	C3
E1	.36	.50	.14
E2	.21	.55	.24
E3	.22	.33	.45
E4	.12	.40	.49
E5	.12	.27	.62

= .21 [1 0 0]
+.55 [1 0 0]
+.24 [1 0 0]

choose different set of orthornormal axes in columns of $\mathbf{V}$: $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{V}\mathbf{V}^\mathsf{T} = \mathbf{I}$

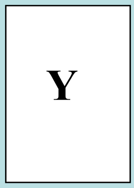


$\mathbf{YV}$ (scalar products of old coordinates with basis vectors) are new coordinates with respect to new axes in $\mathbf{V}$

Dimensional Transmogrifier = Singular Value Decomposition (SVD)

- The SVD is a matrix decomposition of the (possibly centred and possibly normalized) $\mathbf{Y}$:

  $\mathbf{Y} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T}$ where $\mathbf{U}$ and $\mathbf{V}$ are matrices with orthonormal columns: $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$ and $\mathbf{D}_\alpha$ is a diagonal matrix of singular values $\alpha_1 \geq \alpha_2 \geq \cdots > 0$

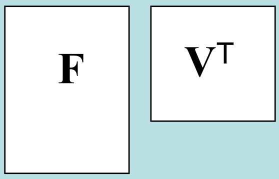


Principal axes of columns: $\mathbf{V}$

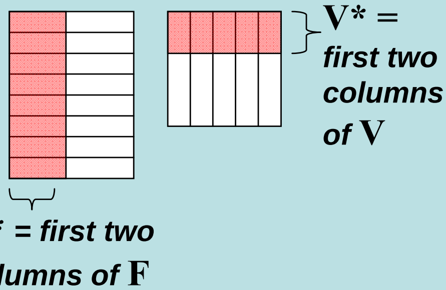
Principal coordinates of rows: $\mathbf{F} = \mathbf{U}\mathbf{D}_\alpha$

So:

$\mathbf{Y} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} = \mathbf{F}\mathbf{V}^\mathsf{T} =$



=



$\mathbf{Y} \approx \mathbf{Y}^* = \mathbf{F}^*\mathbf{V}^{*\mathsf{T}}$

$\sum_i\sum_j(y_{ij} - y_{ij}^*)^2$ minimized

Generalized SVD

We often want to associate weights on the rows and columns, so that the fit is by weighted least-squares, not ordinary least squares, that is we want to minimize (over lower rank matrices $\mathbf{Y}^*$)

$$\sum_i\sum_j r_i c_j (y_{ij} - y_{ij}^*)^2$$

$$\mathbf{D}_r^{1/2}\mathbf{Y}\mathbf{D}_c^{1/2} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} \quad \text{where} \quad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}, \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

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

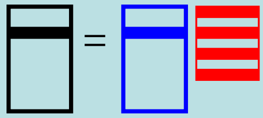
$$\mathbf{Y}^* = etc\ldots$$

$\sum_i\sum_j r_i c_j (y_{ij} - y_{ij}^*)^2$ minimized

minimum (for rank 2 solution) $= \alpha_3^2 + \alpha_4^2 + \cdots$

Generalized principal component analysis

- Suppose we want to represent the (centred) rows of a matrix $\mathbf{Y}$, weighted by (positive) elements (masses) down the diagonal of matrix $\mathbf{D}_m$, where distances between rows are computed in the (weighted) metric defined by matrix $\mathbf{D}_w^{-1}$.
- Total inertia $= \sum_i \sum_j m_i (1/w_j) y_{ij}^2$ =trace$(\mathbf{D}_m \mathbf{Y} \mathbf{D}_w^{-1} \mathbf{Y}^\mathsf{T})$
- $\mathbf{S} = \mathbf{D}_m^{½}\, \mathbf{Y}\, \mathbf{D}_w^{-½} = \mathbf{U}\, \mathbf{D}_\alpha \mathbf{V}^\mathsf{T}$ where $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$
- $\mathbf{Y} = \mathbf{D}_m^{-½}\, \mathbf{U}\, \mathbf{D}_\alpha \mathbf{V}^\mathsf{T} \mathbf{D}_w^{½} = (\mathbf{D}_m^{-½}\, \mathbf{U}\, \mathbf{D}_\alpha)(\mathbf{D}_w^{½}\mathbf{V})^\mathsf{T}$



- Principal coordinates of rows: $\mathbf{F} = \mathbf{D}_m^{-½}\, \mathbf{U}\, \mathbf{D}_\alpha$
- Principal axes of the rows: $\mathbf{D}_w^{½}\, \mathbf{V}$
- Standard coordinates of columns (points metric, axes): $\mathbf{G} = \mathbf{I}(\mathbf{D}_w^{-1})\mathbf{D}_w^{½}\mathbf{V} = \mathbf{D}_w^{-½}\, \mathbf{V}$
- Variances (inertias) explained: $\lambda_1 = \alpha_1^2,\ \lambda_2 = \alpha_2^2, \ldots$

Correspondence analysis

- Table of nonnegative data $\mathbf{N}$
- Divide $\mathbf{N}$ 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 $\mathbf{P}$ 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 masses

Generalized PCA of the rows:

- $\mathbf{Y}$ is the centred matrix of row profiles
- row masses in $\mathbf{D}_m$ are the relative frequencies of the rows $\mathbf{D}_r$
- metric in $\mathbf{D}_w^{-1}$ consists of inverses of the relative frequencies of the columns $\mathbf{D}_c^{-1}$
- Total inertia = χ^2/n

Generalized PCA of the columns:

- $\mathbf{Y}$ is the centred matrix of column profiles
- column masses in $\mathbf{D}_m$ are the relative frequencies of the rows $\mathbf{D}_c$
- metric in $\mathbf{D}_w^{-1}$ consists of inverses of the relative frequencies of the columns $\mathbf{D}_r^{-1}$
- Total inertia = χ^2/n

Both problems lead to the SVD of the same matrix!

Correspondence analysis

to be completed in class

row problem | column problem

⋮

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

$$\frac{p_{ij} - r_i c_j}{\sqrt{r_i c_j}}$$

or equivalently

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

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

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

Asymmetric map of rows: $\mathbf{F}$ and $\mathbf{\Gamma}$ **... of columns: $\mathbf{G}$ and $\mathbf{\Phi}$**

Symmetric map: $\mathbf{F}$ and $\mathbf{G}$

R implementation of CA

```
# 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.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(c(data.rpc[,1],data.cpc[,1]),c(data.rpc[,2],data.cpc[,2]),
 +          type="n", xlab="dim1", ylab="dim2")
text(data.rpc[,1],data.rpc[,2],label=rownames(data_set),col="blue",
 +          font=4)
text(data.cpc[,1],data.cpc[,2],label=colnames(data_set), col="red",
 +          font=2)
# 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

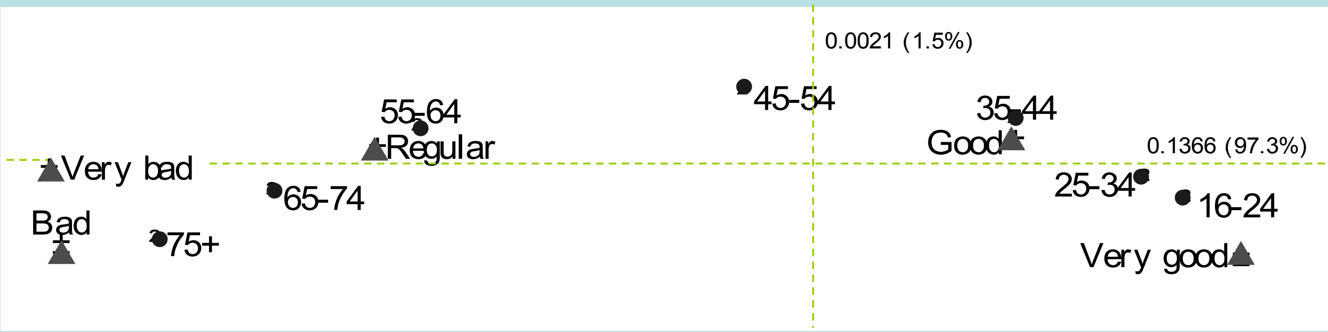
AGE GROUP	Very Good	Good	Regular	Bad	Very Bad	SUM
16-24	243	789	167	18	6	1223
25-34	220	809	164	35	6	1234
35-44	147	658	181	41	8	1035
45-54	90	469	236	50	16	861
55-64	53	414	306	106	30	909
65-74	44	267	284	98	20	713
75+	20	136	157	66	17	396
SUM	817	3542	1495	414	103	6371

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

CA of perceived health status



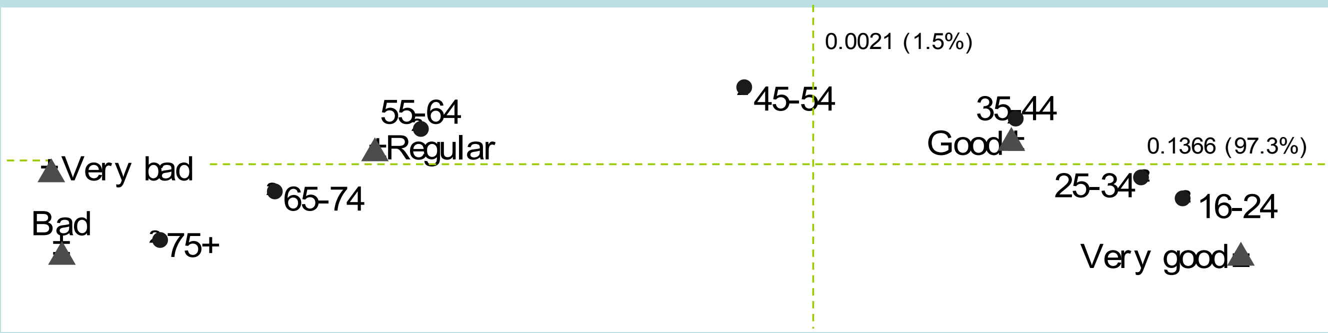
Numerical scale from CA solution (principal coordinates)

```
                 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

```
# 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
```

CA of perceived health status
– optimal scale



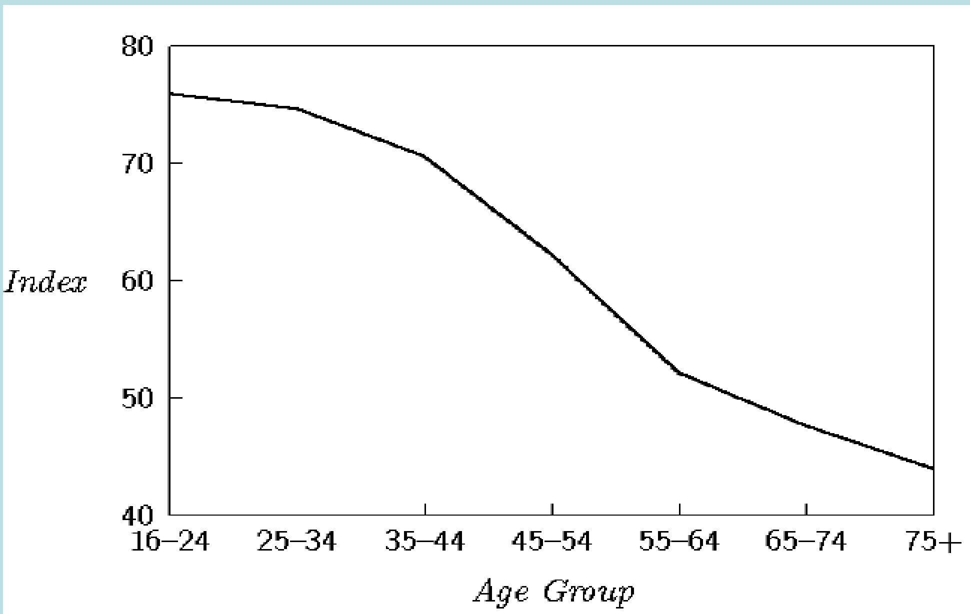
Redefining the optimal scale

	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

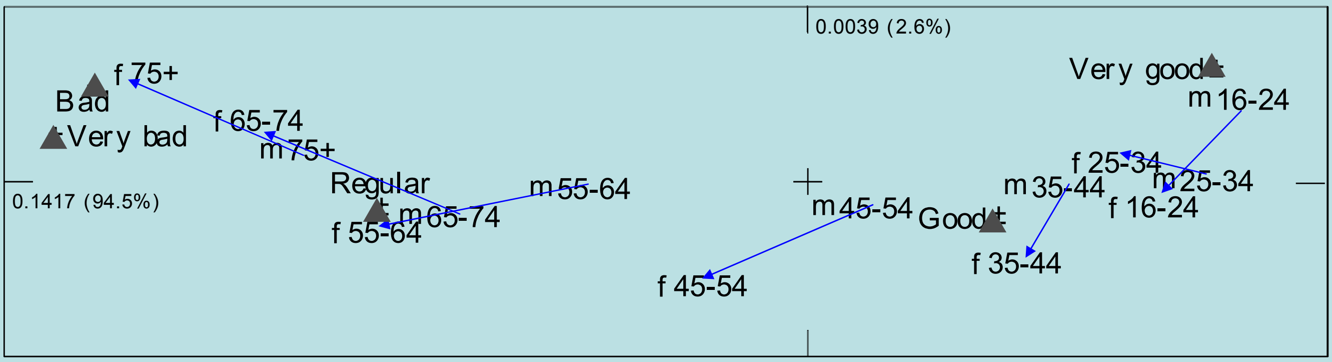


Perceived health status: male – female

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



Data set "author"

Data set from R in the **ca** package*, obtained by issuing the following commands:

```
library(ca)
data(author)
```

The data form a 12 × 26 matrix with the rows representing 12 texts:

The Three Daughters… (Buck)
East Wind:West Wind (Buck)
Lost World (Clarke)
Profiles of the Future (Clarke)
The Drifters (Michener)
Asia (Michener)
Farewell to Arms (Hemingway)
Islands (Hemingway)
Sound and Fury, ch. 6 (Faulkner)
Sound and Fury, chap. 7 (Faulkner)
Bride of Pendorric, ch. 2 (Holt)
Bride of Pendorric, ch. 3 (Holt)

The data are the counts of the letters *a* to *z* in a sample of text (8000-10000 letters) from each of the books.

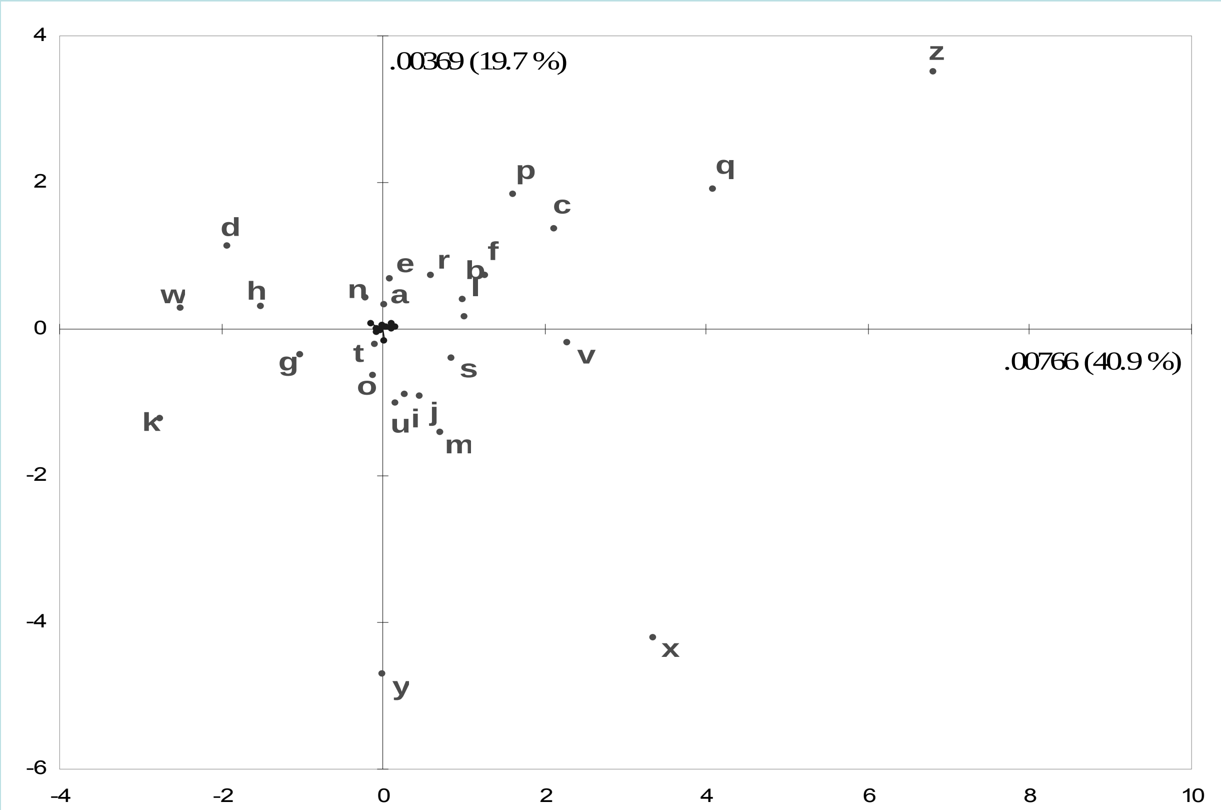
* Nenadić & Greenacre (*Journal of Statistical Software*, 2007)

Data set "author"

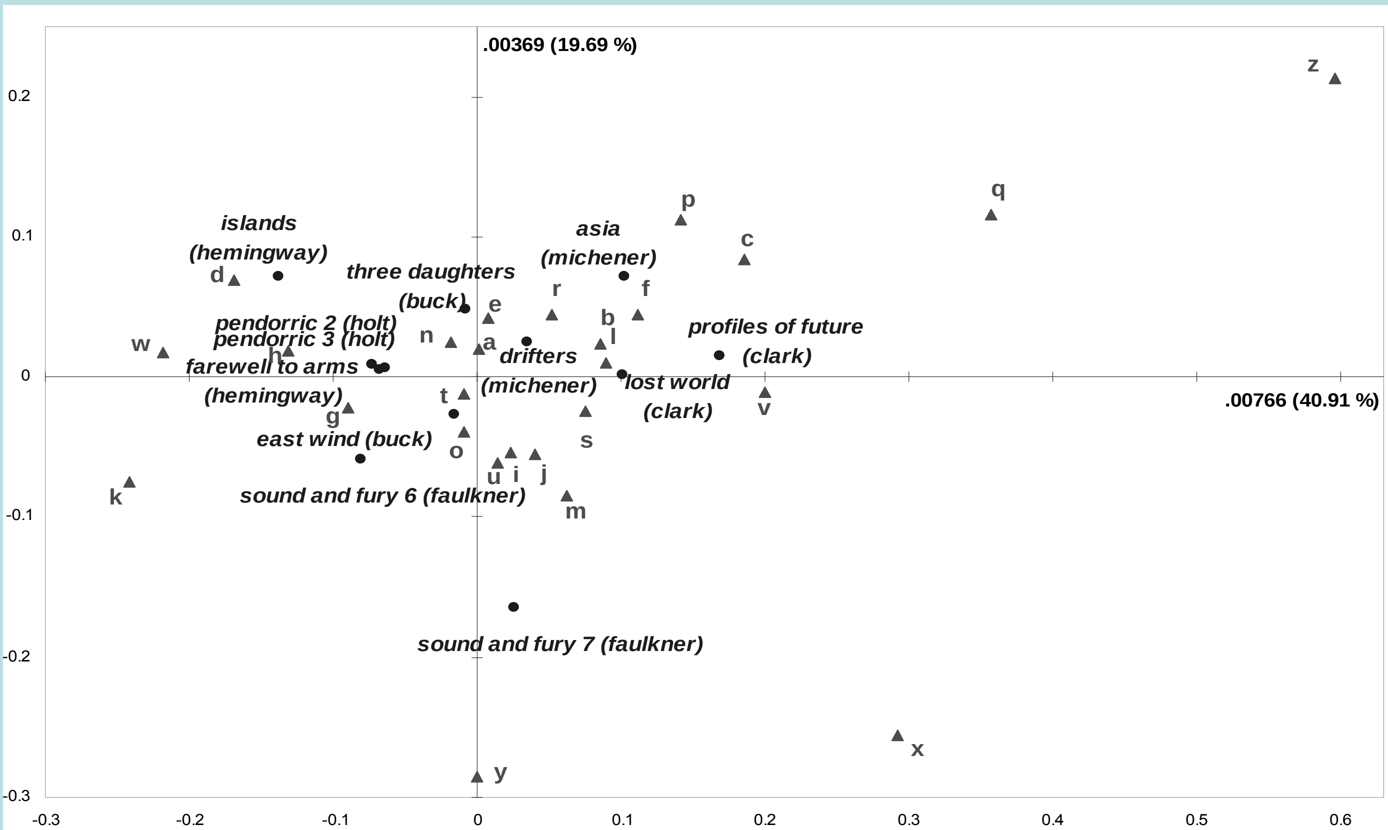
Abbrev.	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
TD-Buck	550	116	147	374	1015	131	131	493	442	2	52	302	159
EW-Buck	557	129	128	343	996	158	129	571	555	4	76	291	247
Dr-Mich	515	109	172	311	827	167	136	376	432	8	61	280	146
As-Mich	554	108	206	243	797	164	100	328	471	4	34	293	149
LW-Clark	590	112	181	265	940	137	119	419	514	6	46	335	176
PF-Clark	592	151	251	238	985	168	152	381	544	7	39	416	236
FA-Hem	589	72	129	339	866	108	159	449	472	7	59	264	158
Is-Hem	576	120	136	404	873	122	156	593	406	3	90	281	142
SF7-Faul	541	109	136	228	763	126	129	401	520	5	72	280	209
SF6-Faul	517	96	127	356	771	115	189	478	558	6	80	322	163
Pen3-Holt	557	97	145	354	909	97	121	479	431	10	94	240	154
Pen2-Holt	541	93	149	390	887	133	154	463	518	4	65	265	194

Abbrev.	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
TD-Buck	534	516	115	4	409	467	632	174	66	155	5	150	3
EW-Buck	479	509	92	3	413	533	632	181	68	187	10	184	4
Dr-Mich	470	561	140	4	368	387	632	195	60	156	14	137	5
As-Mich	482	532	145	8	361	402	630	196	66	149	2	80	6
LW-Clark	403	505	147	8	395	464	670	224	113	146	13	162	10
PF-Clark	526	524	107	9	418	508	655	226	89	106	15	142	20
FA-Hem	504	542	95	0	416	314	691	197	64	225	1	155	2
Is-Hem	516	488	91	3	339	349	640	194	40	250	3	104	5
SF7-Faul	471	589	84	2	324	454	672	247	71	160	11	280	1
SF6-Faul	483	617	82	8	294	358	685	225	37	216	12	171	5
Pen3-Holt	417	477	100	3	305	415	597	237	64	194	9	140	4
Pen2-Holt	484	545	70	4	299	423	644	193	66	218	2	127	2

CA asymmetric map of “author” data

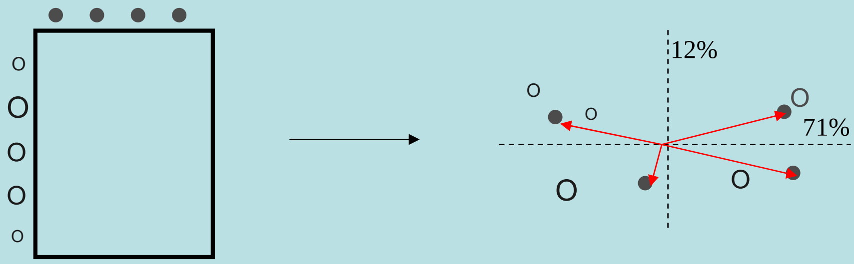


CA symmetric map of “author” data



Inertia contributions in generalized PCA

- (Generalized) 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.

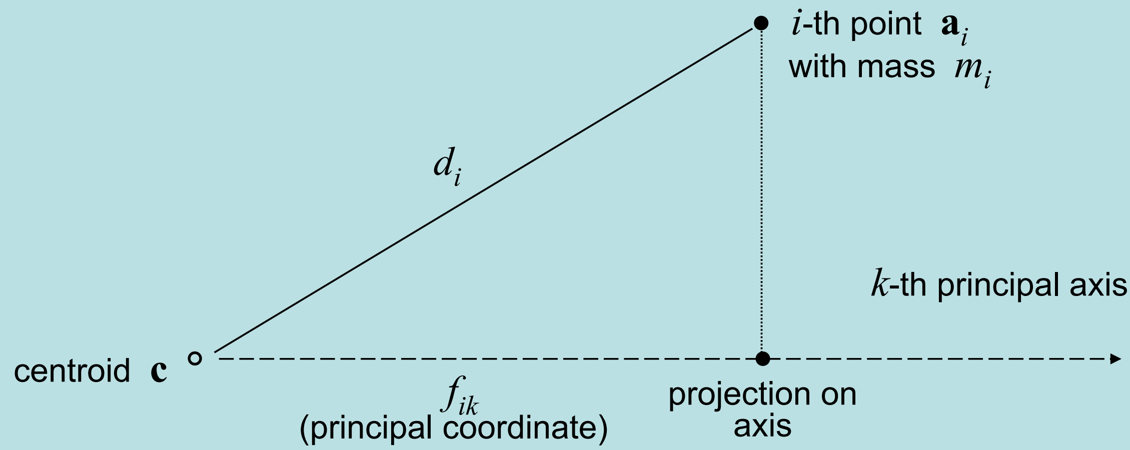


- The inertia (weighted variance) explained in the map applies to all the points: if we say 83% of the inertia is explained in the map, 71% on the first dimension and 12% on the second, this is a figure calculated for all row (or column) points together.

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 $= \Sigma_i\, m_i\, d_i^2 = \Sigma_i\, m_i\, \Sigma_k f_{ik}^2 = \Sigma_k\, \lambda_k$

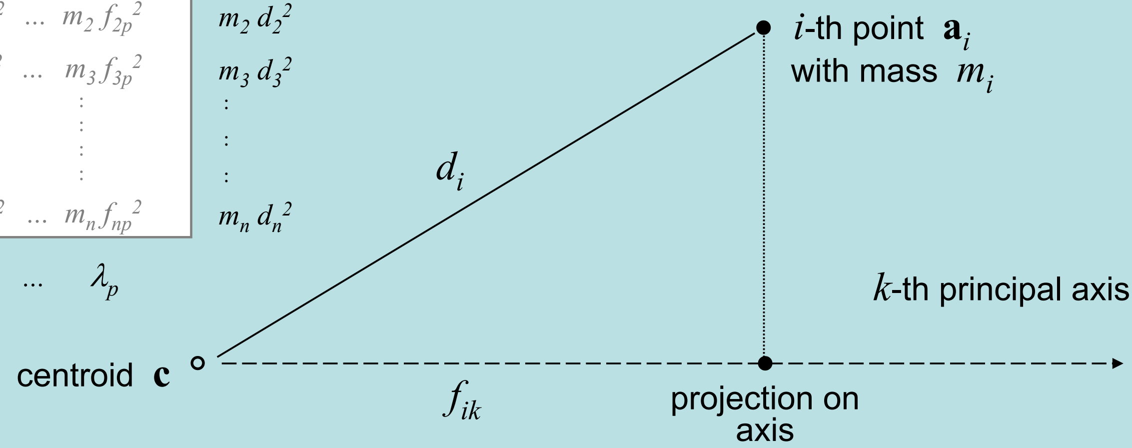
Inertia of i-th point $= m_i\, d_i^2 = m_i\, \Sigma_k f_{ik}^2$

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

Decomposition of inertia

Axes

	1	2	...	p	
1	$m_1 f_{11}^2$	$m_1 f_{12}^2$	...	$m_1 f_{1p}^2$	$m_1 d_1^2$
2	$m_2 f_{21}^2$	$m_2 f_{22}^2$	...	$m_2 f_{2p}^2$	$m_2 d_2^2$
3	$m_3 f_{31}^2$	$m_3 f_{32}^2$	...	$m_3 f_{3p}^2$	$m_3 d_3^2$
	:	:		:	:
n	$m_n f_{n1}^2$	$m_n f_{n2}^2$	...	$m_n f_{np}^2$	$m_n d_n^2$
	λ_1	λ_2	...	λ_p	

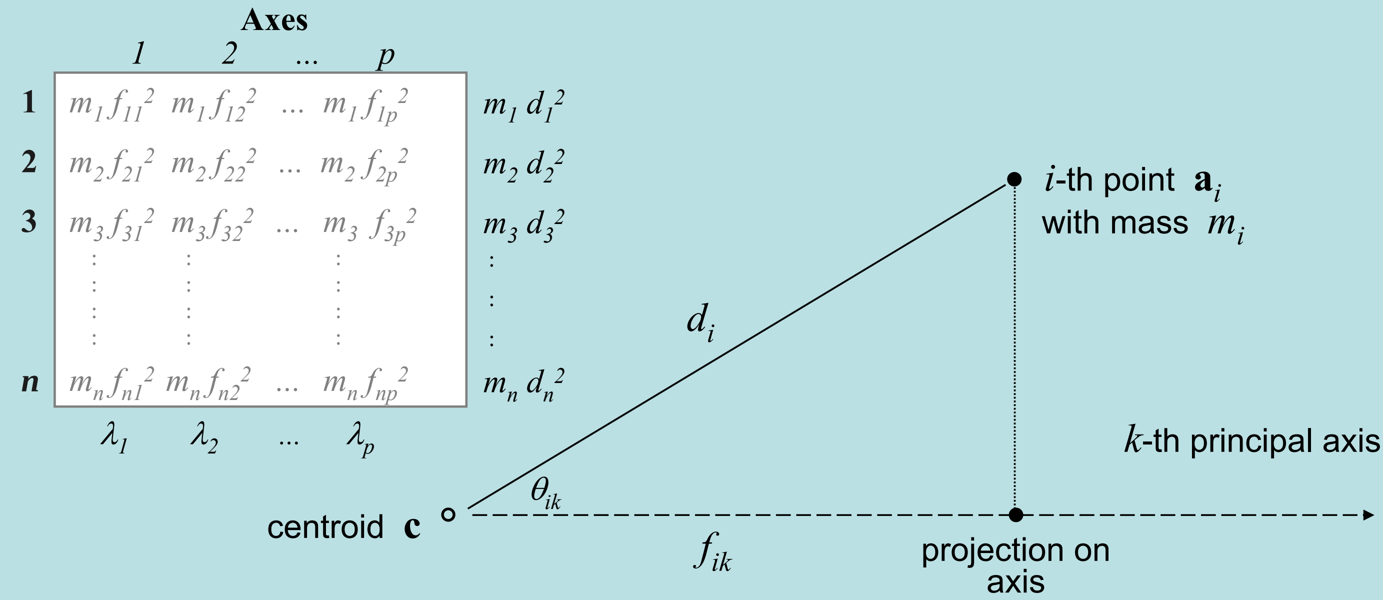


Total inertia of the cloud of points $= \Sigma_i\, m_i\, d_i^2 = \Sigma_i\, m_i\, \Sigma_k f_{ik}^2 = \Sigma_k\, \lambda_k$

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 $= m_i f_{ik}^2$

Inertia contributions



$m_i f_{ik}^2 / \lambda_k$: 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

Summary: 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.

Computation of contributions in CA

```
# 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)
```