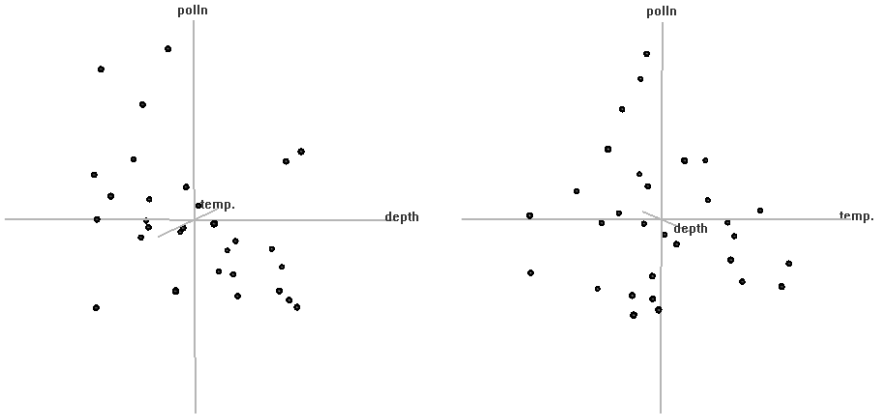
Correspondence Analysis & Related Methods

Michael Greenacre

SESSION 7:

PRINCIPAL COMPONENT ANALYSIS (continued)

Two views of three-dimensional plot



If there exists any (approximately linear) relationships between the variables, we can capitalize on that in the dimension reduction process

Generalized SVD (repeat)

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

$$\text{RSS} = \sum_{i=1}^{n}\sum_{j=1}^{p} r_i c_j (x_{ij} - x_{ij}^*)^2$$

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

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

$$\mathbf{X}^* = etc...$$

Generalized principal component analysis (repeat)

We take the case of points defined in the rows of $\mathbf{X}$; that is, n rows of dimensionality p. First we need to center $\mathbf{X}$ w.r.t. column means:

$$\mathbf{Y} = (\mathbf{I} - (1/n)\mathbf{1}\mathbf{1}^\mathsf{T})\mathbf{X}$$

Suppose the distance between (centered) rows is defined by a weighted Euclidean distance with weights $1/m_j$, and that each row has a mass of r_i.

$$\sum_{i=1}^{n} r_i \sum_{j=1}^{p} \frac{(y_{ij} - y_{ij}^*)^2}{m_j} \qquad \text{RSS} = \sum_{i=1}^{n}\sum_{j=1}^{p} (r_i / m_j)(y_{ij} - y_{ij}^*)^2$$

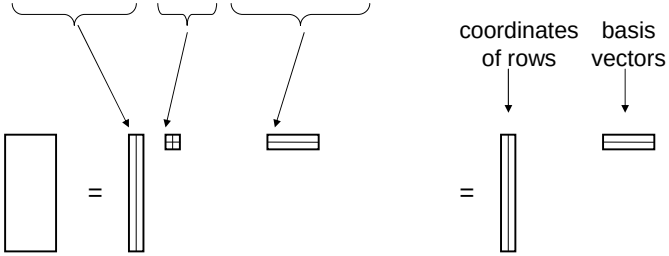
$$\mathbf{D}_r^{1/2}\mathbf{Y}\mathbf{D}_m^{-1/2} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} \quad \text{where } \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I},\ \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

$$\mathbf{Y} = \underbrace{\mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{D}_\alpha}\ \underbrace{(\mathbf{D}_m^{1/2}\mathbf{V})^\mathsf{T}} \qquad \mathbf{Y}^* = etc...$$

principal coordinates of rows basis vectors (principal axes)

Coordinates; Basis vectors (principal axes)

$$\mathbf{Y}^* = \mathbf{D}_r^{-1/2}\mathbf{U}_{[2]}\mathbf{D}_{\alpha[2]}(\mathbf{D}_m^{1/2}\mathbf{V}_{[2]})^{\mathsf{T}}$$ Is best rank 2 approximation

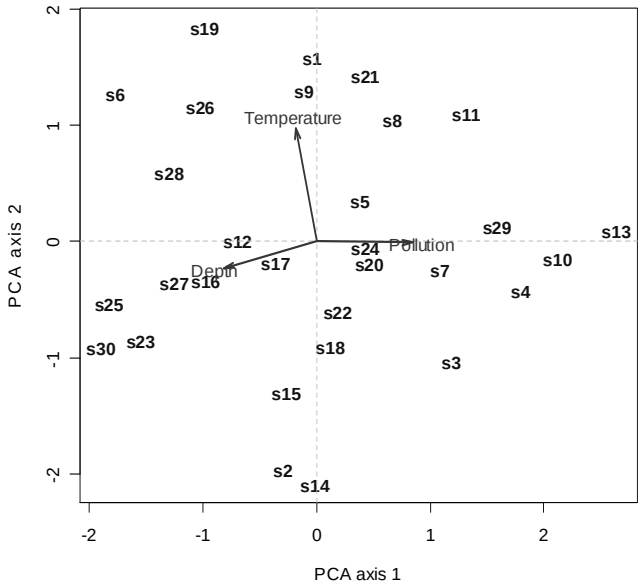


Principal axes orthonormal: $(\mathbf{D}_m^{1/2}\mathbf{V})^{\mathsf{T}}\mathbf{D}_m^{-1}(\mathbf{D}_m^{1/2}\mathbf{V}) = \mathbf{I}$

R code (assuming data in X(n×p))

```
Y     <- sweep(X, 2, apply(X, 2, mean))

# s2 contains the column variances BUT divided by n not by n-1
s2    <- ((n-1)/n)*apply(X, 2, var)

# Dsm1 is the inverse std devns in a diagonal matrix

Dsm1  <- diag(1/sqrt(s2))

S     <- sqrt(1/n) * as.matrix(Y) %*% Dsm1

S.svd <- svd(S)

FF    <- sqrt(n) * S.svd$u %*% diag(S.svd$d)


plot(FF[,1], FF[,2], type="n", xlab="PCA axis 1", ylab="PCA axis 2")
text(FF[,1], FF[,2], labels=rownames(X), col="blue", font=2)

abline(h=0, lty=2, col="gray")
abline(v=0, lty=2, col="gray")

loads <- cor(cbind(S,FF))[1:p,(p+1):2*p]

for(j in 1:p) {
  arrows(0,0,loads[j,1], loads[j,2], length=0.1, angle=15, col="red", lwd=2)
  text(1.1*loads[j,1],1.1*loads[j,2], labels=colnames(X)[j], col="red")
  }
```

"Best" view of three-dimensional plot



Explained variance of all 3 variables:

Axis 1: 1.405 (46.8%)

Axis 2: 1.002 (33.4%)

In 2-d: 2.407 (80.2%)

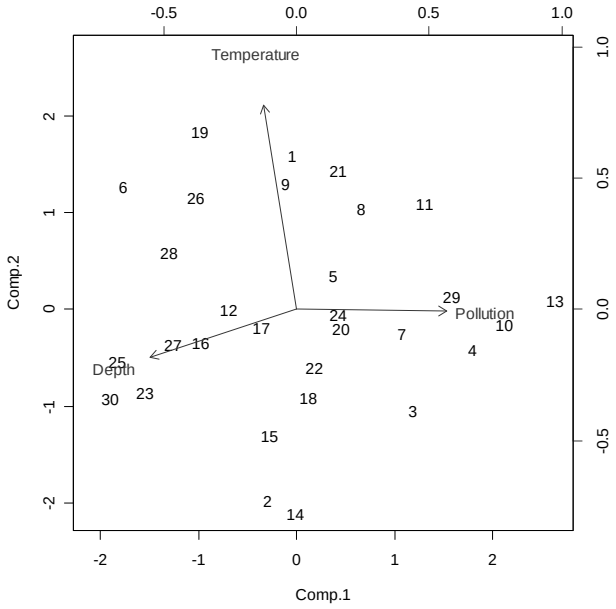
Explained variance of each variable in two dimensions:

Polln : 0.704 (70.4%)

Depth: 0.719 (71.9%)

Temp.: 0.984 (98.4%)

"Best" view of three-dimensional plot



Explained variance of all 3 variables:

Axis 1: 1.405 (46.8%)

Axis 2: 1.002 (33.4%)

In 2-d: 2.407 (80.2%)

Explained variance of each variable in two dimensions:

Polln : 0.704 (70.4%)

Depth: 0.719 (71.9%)

Temp.: 0.984 (98.4%)

Using R functions `biplot` & `princomp`

R functions **princomp** and **prcomp** for principal component analysis

```
# in function princomp,
# the option cor=T implies standardization of variables (columns)

biplot(princomp(X, cor=T), scale=0)

# in function prcomp,
# the option scale=T implies standardization of variables (columns)

biplot(prcomp(X, scale=T), scale=0)

# in both the above, the biplot function has option scale=alpha,where
#   alpha=1 if singular values go with variables (column principal coordinates)
#   alpha=0 if singular values go with cases (row principal coordinates)

names(princomp(X, cor=T))
[1] "sdev"     "loadings" "center"   "scale"     "n.obs"     "scores"    "call"
```

singular values (square these to get eigenvalues/ parts of SS explained) — eigen-vectors — variable means — (using n) variable std devns — n — row p.c.'s (using n) — function call

(using n-1) — row p.c.'s (using n-1)

```
names(prcomp(X, scale=T))
[1] "sdev"     "rotation" "center"   "scale"     "x"
```

Correlations ("loadings") of each variable with the principal axes

Correlations	Axis1	Axis2	Axis3
Pollution	0.839	-0.008	-0.544
Depth	-0.817	-0.227	-0.530
Temperature	-0.184	0.975	-0.128

- The sum of squares of each row = 1, the variance of the variable.
- The sum of squares of each column = variance on axis.
- Because the axes are uncorrelated, the correlations are identical to the regression coefficients of each variable on the axes.

Eigenvectors

Eigenvectors	Axis1	Axis2	Axis3
Pollution	0.708	-0.008	-0.706
Depth	-0.689	-0.227	-0.688
Temperature	-0.155	0.974	-0.166

- The eigenvectors (columns) give the coefficients for transforming the original three variables into their principal components, e.g. first component:

  $0.708 \times \text{Pollution} - 0.689 \times \text{Depth} - 0.155 \times \text{Temperature}$

- Scale of component depends on scale of the variables: if variables have variance 1, component has variance equal to corresponding eigenvalue (definition of variance has to be same in each case); if variables have sum of squares equal to 1, then sum of squares of the component is the eigenvalue.

Example: Economic indicators

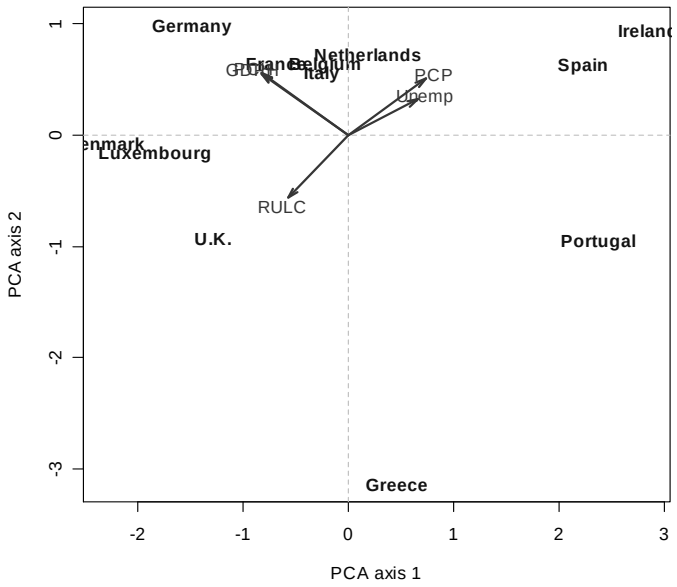
	Unemp	*GDPH*	*PCH*	*PCP*	*RULC*
Belgium	8.8	102	104.9	3.3	89.7
Denmark	7.6	134.4	117.1	1	92.4
Germany	5.4	128.1	126	3	90
Greece	8.5	37.7	40.5	2	105.6
Spain	16.5	67.1	68.7	4	86.2
France	9.1	112.4	110.1	2.8	89.7
Ireland	16.2	64	60.1	4.5	81.9
Italy	10.6	105.8	106	3.8	97.4
Luxembourg	1.7	119.5	110.7	2.8	95.9
Netherlands	9.6	99.6	96.7	3.3	86.6
Portugal	5.2	32.6	34.8	3.5	78.3
U.K.	6.5	95.3	99.7	2.1	98.9

Apply same R code to the above matrix (see R script, week 4), starting with

X <- EU

(above data stored in data.frame **EU)**

PCA/biplot


Germany
Ireland
Netherlands
Spain
Italy
PCP
Unemp
Luxembourg
RULC
U.K.
Portugal
Greece
PCA axis 2
PCA axis 1
1
0
-1
-2
-3
-2
-1
0
1
2
3

Correspondence Analysis & Related Methods

Michael Greenacre

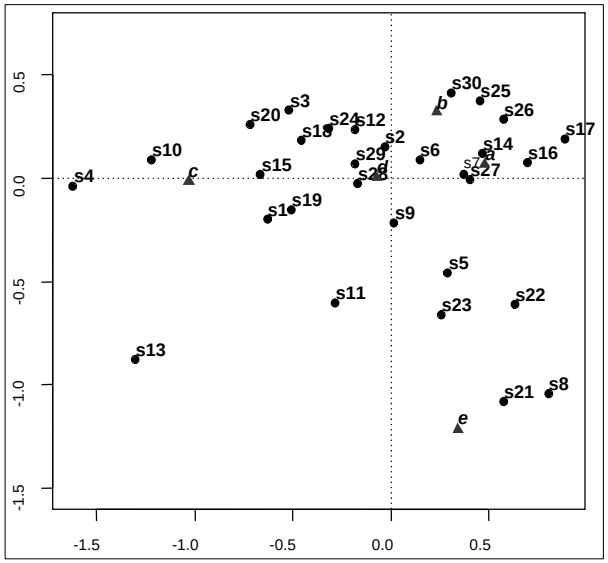
SESSION 8:

CORRESPONDENCE ANALYSIS (introduction)

From PCA to CA

- Principal component analysis (PCA) applies to variables on a continuous measurement scale (actually, an interval scale; if data are strongly "ratio" then a logarithmic transformation should be performed)
- PCA uses the Euclidean distance (usually standardized because of the issue of scale & variance for continuous variables; but NOT standardized if data have been log-transformed), and usually each sample receives the same weight.
- Correspondence analysis (CA) is the natural analogue of PCA for count data.
- CA uses the chi-square distance between the row profiles and weights each row proportionally to its marginal frequency.
- Otherwise, the process of dimension reduction, projection onto planes to conserve maximum variance, correlating variables with axes, is the same.
- An unusual and very particular property of CA: it applies equally and symmetrically to the column profiles. That is, if you transpose the matrix you get the same result (this is not true for PCA).

CA of species counts



Explained inertia:

Axis 1: 0.288 (53.0%)

Axis 2: 0.120 (22.2%)

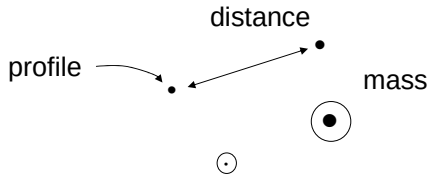
In 2-d: 0.408 (75.2%)

Total inertia:

0.5436

Using R function ca from **ca** package (Nenadić & Greenacre, *JSS*, 2006)

Three basic geometrical concepts

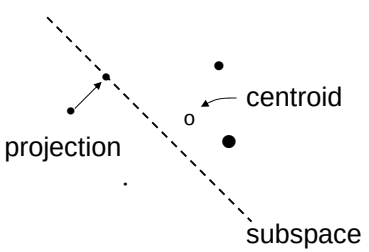


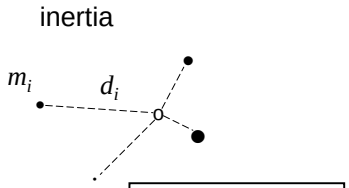
profile – *the coordinates (position) of the point*

mass – *the weight given to the point*

distance – *the measure of proximity between points*

Four derived geometrical concepts





inertia = $\sum_i m_i d_i^2$

centroid – *the weighted average position*

subspace – *space of reduced dimensionality within the space*

projection – *the closest point in the subspace*

inertia – *the weighted sum-of-squared distances to centroid*

Row and column profiles

- A profile is a set of relative frequencies, that is a set of frequencies expressed relative to their total (often in percentage form).
- Each row or each column of a table of frequencies defines a different profile.
- It is <u>these profiles</u> which CA visualises as points in a map.

original data

	C1	C2	C3	
E1	5	7	2	14
E2	18	46	20	84
E3	19	29	39	87
E4	12	40	49	101
E5	3	7	16	26
	57	129	126	312

row profiles

	C1	C2	C3	
E1	.36	.50	.14	1
E2	.21	.55	.24	1
E3	.22	.33	.45	1
E4	.12	.40	.49	1
E5	.12	.27	.62	1

column profiles

	C1	C2	C3
E1	.09	.05	.02
E2	.32	.37	.16
E3	.33	.22	.31
E4	.21	.31	.39
E5	.05	.05	.13
	1	1	1

original data

SITE	a	b	c	d	e	sum
s1	0	2	9	14	2	27
s2	26	4	13	11	0	54
s3	0	10	9	8	0	27
s4	0	0	15	3	0	18
s5	13	5	3	10	7	38
s6	31	21	13	16	5	86
s7	9	6	0	11	2	28
s8	2	0	0	0	1	3
s9	17	7	10	14	6	54
s10	0	5	26	9	0	40
⋮	⋮	⋮	⋮	⋮	⋮	⋮
s29	11	0	7	8	0	26
s30	24	37	5	18	1	85
sum	404	262	252	327	89	1334

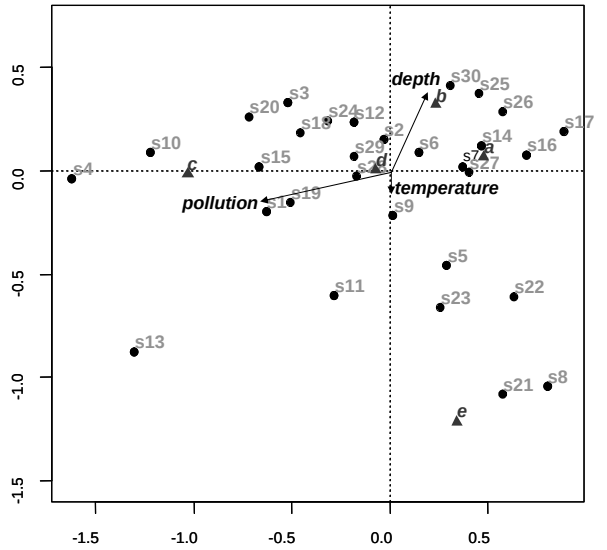
row profiles

SITE	a	b	c	d	e
s1	0.000	0.074	0.333	0.519	0.074
s2	0.481	0.074	0.241	0.204	0.000
s3	0.000	0.370	0.333	0.296	0.000
s4	0.000	0.000	0.833	0.167	0.000
s5	0.342	0.132	0.079	0.263	0.184
s6	0.360	0.244	0.151	0.186	0.058
s7	0.321	0.214	0.000	0.393	0.071
s8	0.667	0.000	0.000	0.000	0.333
s9	0.315	0.130	0.185	0.259	0.111
s10	0.000	0.125	0.650	0.225	0.000
⋮			⋮		
s29	0.423	0.000	0.269	0.308	0.000
s30	0.282	0.435	0.059	0.212	0.012
ave.	0.303	0.196	0.189	0.245	0.067

column profiles

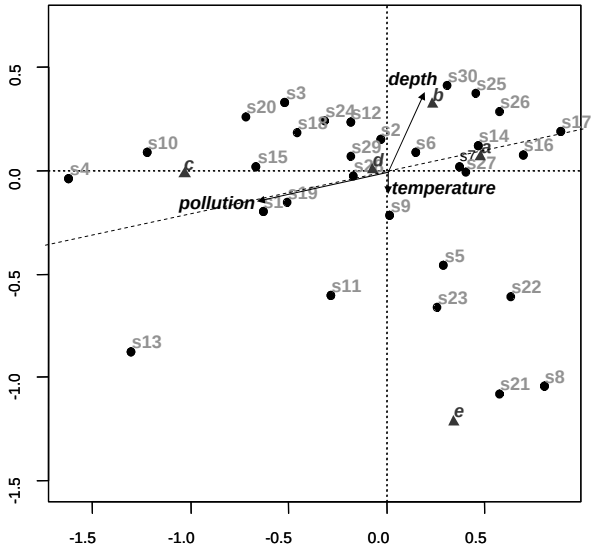
SITE	a	b	c	d	e	ave.
s1	0.000	0.008	0.036	0.043	0.022	0.020
s2	0.064	0.015	0.052	0.034	0.000	0.040
s3	0.000	0.038	0.036	0.024	0.000	0.020
s4	0.000	0.000	0.060	0.009	0.000	0.013
s5	0.032	0.019	0.012	0.031	0.079	0.028
s6	0.077	0.080	0.052	0.049	0.056	0.064
s7	0.022	0.023	0.000	0.034	0.022	0.021
s8	0.005	0.000	0.000	0.000	0.011	0.002
s9	0.042	0.027	0.040	0.043	0.067	0.040
s10	0.000	0.019	0.103	0.028	0.000	0.030
⋮			⋮			⋮
s29	0.027	0.000	0.028	0.024	0.000	0.019
s30	0.059	0.141	0.020	0.055	0.011	0.064

Adding explanatory variables to the map



Do a (weighted) regression of the variable on the two dimensions, then use the regression coefficients as coordinates

Adding explanatory variables to the map



Variance explained of supplementary variables:

Polln: 69.5%
Depth: 30.4%
Temp: 2.1%

Explanatory variables not specifically "optimized" in the display (this is called "***indirect gradient analysis***")

The hidden secrets of the species

Species modelled to respond linearly to the gradients:

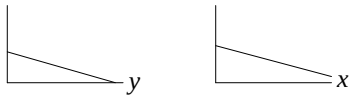
*Species **a**:*

*Species **b**:*

*Species **c**:*

*Species **d**:*

*Species **e**:*



y - pollution

x - depth

Data were generated conforming to these relationships, leading to a table of 30 samples by 5 species, with a lot of "noise" added; there is no relationship to temperature.

CA versus PCA

- **PCA analyzes interval data (typically continuous measurements such as temperature, depth, salinity, concentrations)**
- **CA analyzes ratio data (typically counts, abundances, morphometric data)**
- **Since ratio data can be made interval by taking logarithmic transformation, PCA on log-transformed ratio data is an alternative to CA, BUT not a good idea when there are lots of zeros**
- **CA analyzes relative values in the rows (or columns), i.e. the profiles – PCA analyzes the absolute values**
- **PCA standardizes (if necessary) using the square root of variance (std devn)**
- **CA standardizes using the square root of the mean – chi^2 distance**
- **PCA usually does not weight the points**
- **CA always weights the points**
- **Both methods reduce dimensionality in the same way – by minimizing the (weighted) sum of squared distances from the points to the map, i.e. maximizing the (weighted) variance of the points in the map**
- **Both methods have the same biplot options for the map**