Principal Component Analysis

for autocorrelated data:

a geostatistical perspective

Hans WACKERNAGEL

Technical Report N–22/98/G
August 1998

Centre de Géostatistique — Ecole des Mines de Paris
35 rue Saint Honoré, F–77305 Fontainebleau, FRANCE
http://cg.ensmp.fr

Foreword

This document is based on talks about Principal Component Analysis given at the *Isatis User's Meeting 1997* at Fontainebleau (Chapter 1) and at the *7th International Meeting on Statistical Climatology* at Whistler (Chapter 2). Chapter 3 follows a text written by Michel GRZEBYK and myself for the proceedings of the *6th International Meeting on Statistical Climatology* at Galway.

Various applications of Principal Component Analysis are discussed on examples and possible geostatistical extensions are listed. An appendix provides a programming example with XLispStat.

Contents

Chapter 1

Introduction

Principal Component Analysis (PCA) can be used for:

1. data compression,
2. multivariate outlier detection,
3. deciphering a correlation matrix,
4. identifying underlying factors,
5. detecting intrinsic correlation.

An application will be given and discussed for each topic.

PCA has been transposed from a *multi-variate* into a *multi-station* space-time context by climatologists, where the technique received the name of Empirical Orthogonal Functions (EOF) analysis. The multi-station implementation is obviously also very interesting for users of Isatis. For example, petroleum and mining engineers might use EOFs for a *multi-borehole* or a *multi-trace* analysis.

Principal Component Analysis is well defined for the case of independent samples as they are found in psychometric or sociologic studies. The standard PCA model can be applied to autocorrelated data in the framework of the *intrinsic correlation model*. It is thus of paramount importance to check whether autocorrelated data (in space or time) are intrinsically correlated. If this is not the case the blind application of the standard PCA model may generate misleading results.

In Chapter 2 we present standard PCA and the intrinsic correlation model together with several examples from case studies. In Chapter 3 we explain EOFs and discuss an illustrative analysis of a set of ammonium time series measured along the river Elbe.

Chapter 4 is dedicated to more sophisticated geostatistical models that extend PCA to the case of non-intrinsically correlated data.

Chapter 2

Principal Component Analysis

Let us denote by $Z_i, i = 1, \ldots, N$ a set of variables to be analysed, e.g. petrophysical properties, seismic attributes, soil pollution elements, geochemical elements, morphological parameters, to mention but a few.

The Z_i are usually correlated and the aim of Principal Component Analysis will be to examine the system using uncorrelated factors Y_p.

The variances of the variables Z_i are denoted σ_{ii} and their sum is called the *total variance*. We look for *uncorrelated* new variables Y_p, $p = 1, \ldots, N$, which *partition optimally* (in the least squares sense) the total variance.

Principal Component Analysis:

- transforms linearly the *correlated* variables Z_i into *uncorrelated* principal components Y_p.
- ordering the PCs by decreasing variance, each PC extracts a *maximal share* of the total variance.

2.1 Data compression

This is an example from geophysical exploration following [5]. Let $\mathbf{Z}$ be an $n \times N$ matrix of n seismic profiles and N samples.

The PCA transformation is:

$$\boxed{\mathbf{Y} = \mathbf{Z}\,\mathbf{Q}}$$

where $\mathbf{Q}$ is an $N \times N$ orthogonal matrix of coefficients.

The principal components matrix $\mathbf{Y}$ is of size $n \times N$ (like $\mathbf{Z}$). The variance of each principal component $\mathbf{y}_p$ (a column of the matrix $\mathbf{Y}$) is given by the corresponding eigenvalue λ_p. We assume that the eigenvalues have been ordered by decreasing variance.

The idea for compressing the data is to retain only the principal components having the largest variances. Keeping $M < N$ principal components, which

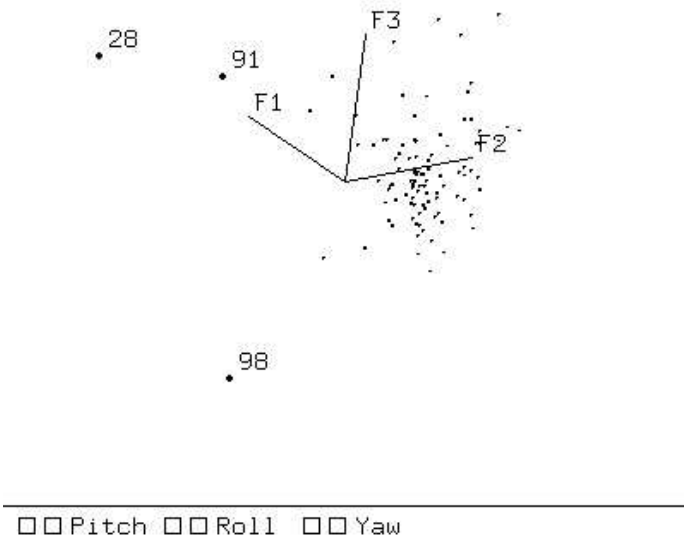

Figure 2.1: Spin plot.

explain a substantial amount of the total variance in an $n \times M$ matrix $\widetilde{\mathbf{Y}}$, we have the approximation

$$\mathbf{Z} \cong \widetilde{\mathbf{Y}}\,\widetilde{\mathbf{Q}}^{\top}$$

where $\widetilde{\mathbf{Q}}$ is an $N \times M$ matrix of eigenvectors.

This approximation is interesting for data compression if M is *much smaller* than N: we can then save substantial disk space by storing the two matrices $\widetilde{\mathbf{Y}}$, $\widetilde{\mathbf{Q}}$ instead of the original data $\mathbf{Z}$. The $n \times N$ numbers of the matrix $\mathbf{Z}$ are then expressed with:

$$n \times M + M \times N \quad \text{numbers.}$$

Numerical example: following HAGEN [5], having originally $n = 200$ good quality seismic traces in an $N = 50$ sample window, if the $M = 4$ first principal components express 85% of the total variance, the original data base of 200×50 =10,000 samples can be reduced to:

$$200 \times 4 + 4 \times 50 \;=\; 1,000 \quad \text{samples.}$$

This new data base requires only one tenth of the storage space, preserving –as it seems– an accurate description of the main geological patterns important for reservoir characterization.

Al	.69												
V	.97	.69											
P	.89	.53	.87										
Cr	.94	.72	.95	.82									
Cu	.77	.67	.72	.71	.73								
Nb	.72	.43	.81	.71	.73	.50							
As	.87	.60	.87	.84	.83	.69	.76						
Mo	.79	.67	.81	.74	.78	.65	.78	.85					
Si	-.97	-.75	-.94	-.87	-.91	-.76	-.73	-.86	-.80				
Ti	-.93	-.59	-.90	-.83	-.85	-.66	-.69	-.82	-.72	.94			
Ce	-.76	-.44	-.73	-.67	-.73	-.50	-.57	-.64	-.54	.77	.81		
Zr	-.89	-.73	-.86	-.78	-.82	-.70	-.65	-.80	-.74	.94	.91	.70	
Y	-.92	-.68	-.89	-.80	-.86	-.68	-.68	-.80	-.73	.96	.96	.84	.93
	Fe	Al	V	P	Cr	Cu	Nb	As	Mo	Si	Ti	Ce	Zr

Table 2.1: Correlation matrix of the Mali geochemical variables

2.2 Multivariate outliers

The first three Principal Components:

- usually concentrate a significant share of the total variance,
- however, in the presence of outliers or multimodality they do not provide an ideal representation of the data.

Soon as the sample cloud is not of ellipsoidal shape it is interesting to rotate it in a three dimensional PCA space using a *spinning plot*. This may help in identifying multivariate outliers or subpopulations by finding new projection planes which allow to split the data cloud into subclouds.

Example: 102 soil pollution samples were analyzed for 7 elements: Pb, Cd, Cr, Cu, Ni, Zn, Mo. The first three PCs concentrate 82% of the total variance and thus represent a very interesting subspace to have a look at the data. Rotating the sample cloud in the coordinate system given by these three PCs we easily find a projection plane showing three samples outside the main cloud, see Figure 2.1. The rotation is performed interactively on the computer screen by pressing in turn the three buttons "pitch", "roll", "yaw".

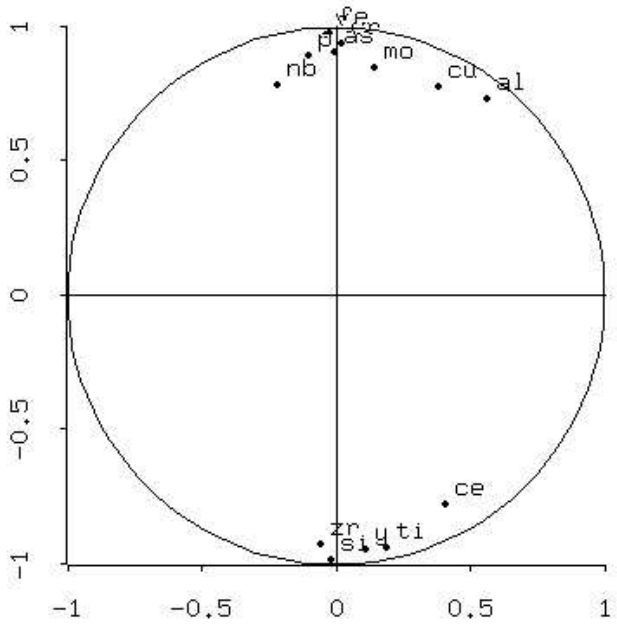

Figure 2.2: Circle of correlations for the first two principal components of the Mali geochemical variables. PC1 (ordinate) against PC2 (abscissa).

2.3 Deciphering a correlation matrix

Let $\mathbf{V}$ be the $N \times N$ correlation matrix of $\mathbf{Z}$. The *eigenvalues* $\lambda_1, \ldots, \lambda_p$ of $\mathbf{V}$, in decreasing order, are the variances of the principal components $\mathbf{y}_1, \ldots, \mathbf{y}_p$. The matrix $\mathbf{Q}$ in the PCA transformation $\mathbf{Y} = \mathbf{Z}\,\mathbf{Q}$ is the matrix of *eigenvectors* of $\mathbf{V}$.

Each eigenvector $\mathbf{q}_p$ shrinks the matrix $\mathbf{V}$ to a single number λ_p:

$$\mathbf{V}\,\mathbf{q}_p \;=\; \lambda_p\,\mathbf{q}_p$$

The correlations between the original variables $\mathbf{z}_i$ and the principal components $\mathbf{y}_p$ can be computed from the eigenanalysis:

$$\boxed{\mathrm{corr}(\mathbf{z}_i,\,\mathbf{y}_p) \;=\; \sqrt{\lambda_p}\,q_{ip} \;=\; r_{ip}}$$

where q_{ip} is the element number i of a given eigenvector $\mathbf{q}_p$.

Example: 1054 soil samples from lateritic terrain in Mali analysed for 14 elements: Fe, Al, V, P, Cr, Cu, Nb, As, Mo, Si, Ti, Ce, Zr, Y; they are described in ROQUIN ET AL. [10].[1]

[1] This spatially autocorrelated data turned out to be *intrinsically correlated* [8], a concept that will be introduced at the end of this chapter.

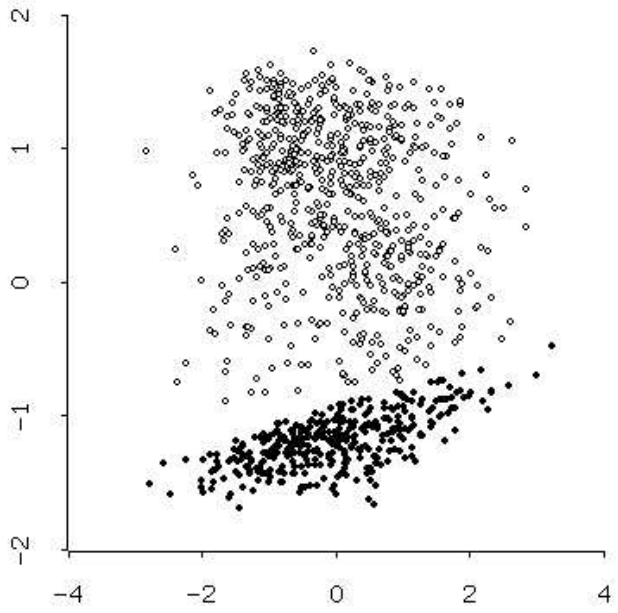

Figure 2.3: Sample cloud for the first two principal components of the Mali geochemical variables. PC1 (ordinate) against PC2 (abscissa). Duricrusts: white (Fe-Al), Flats: black (SiO_2).

The Table 2.1 shows the very simple structure of the correlation matrix: an opposition between the duricrust variables (Fe, Al, V, P, Cr, Cu, Nb, As, Mo) and the variables of the flats (Si, Ti, Ce, Zr, Y); the variables are positively correlated within each group and negatively correlated between groups.

The Figure 2.2 is called the *circle of correlations*. It displays the correlations r_{ip} between the original variables $\mathbf{z}_i$ and a pair of principal components (factors). The coordinates of the variables on Figure 2.2 are obtained using the values of correlations with the first (ordinate) and the second (abscissa) principal component. The first principal component can be termed a "duricrust factor" as it displays in an obvious way the opposition between the variables characteristic of the duricrust variables (Fe,...) and the flats (Si,...).

The Figure 2.3 plots the sample cloud in the coordinate system provided by the first (ordinate) and the second (abscissa) principal components. Two subclouds can be seen: white coloured dots represent the samples from the duricrusts and black dots represent the samples from the flats.

The Figure 2.4shows the geographical map of sample locations. The white coloured dots are samples classified as "duricrust" while the black dots are

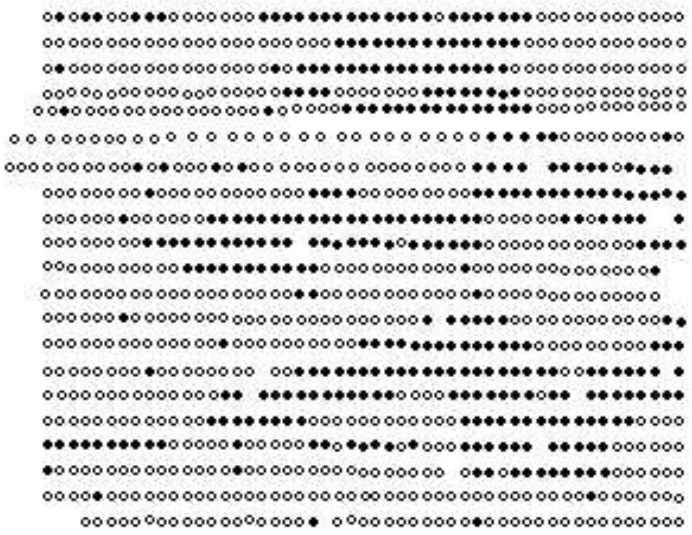

Figure 2.4: Geographical map of sample locations in a 4×5 km^2 area. Duricrusts: white (Fe-Al), Flats: black (SiO_2) .

viewed as from "flats". Actually this map matches well the geological map displayed in ROQUIN ET AL. (p152).

2.4 Identifying underlying factors

In a multigaussian context, the *uncorrelated* principal components can be viewed as *independent* factors. External evidence may confirm the interpretation of patterns in the correlation between variables and factors. A first example was already provided with the "duricrust factor" of the Mali data. Now let us see a second example from human biology.

Example: MORRISON [9] (pp282–284) examines bone length data for 276 leghorn fowl. The measured bones are humerus, tibia, ulna and femur.

The following table exhibits the correlations between the four bone length measurements on 256 individuals:

Humerus	1			
Tibia	.875	1		
Ulna	.940	.877	1	
Femur	.878	.924	.886	1
	Humerus	Tibia	Ulna	Femur

The correlation coefficients are all positive, range between .87 and .94, and seem quite alike. No distinctive pattern is to be seen in these numbers.

The following table shows the correlations between the four bone length measurements and the first two principal components:

	PC1	PC2
Humerus	.96	-.22
Ulna	.96	-.20
Tibia	.96	.22
Femur	.96	.20
Component variance	3.69	.17
Percentage	92 %	4%

The first principal component has an eigenvalue of 3.69 which corresponds to 92% of the total variance, while the second principal component accounts for only 4% of the total variance, so that the third and fourth components share the remaining 4%.

PC1 is positively correlated with all four variables, with an equal strength of .96. This strong positive correlation of all variables with the first principal component is usually explained in morphometry by the fact that all bone lengths are proportional to the size of the individuals and so PC1 is termed a "size factor".

PC2 is negatively correlated with Humerus and Ulna, while it is positively correlated with Tibia and Femur. The first two are bones of the wings of the leghorn fowl, while the other two bones belong to their legs. This means that some individuals may have long legs as compared to their wing size, and vice-versa. This is why PC2 is usually called a "shape factor".

Finally we redisplay the correlation matrix using the knowledge gained from the PCA:

Humerus	1			
Ulna	.940	1		
Tibia	.875	.877	1	
Femur	.878	.886	.924	1
	Humerus	Ulna	Tibia	Femur

By grouping wing and leg bones its structure becomes better visible: the between-group correlations are weaker than the within-group correlations, i.e. the bones within a wing or a leg are better correlated (.94, .92) than between legs and wings (.87 to .89).

Principal Component Analysis is thus a useful tool to rearrange the sequence of the variables so that patterns of correlation may become visible directly on the correlation matrix.

2.5 Detecting intrinsic correlation

With multivariate data *autocorrelated* in space or time principal components can be used to check whether the data comply with the intrinsic correlation model.

In the intrinsic correlation model all direct and cross variograms $\gamma_{ij}(\mathbf{h})$ of a set of variables are *proportional* to a basic variogram $\gamma(\mathbf{h})$:

$$\gamma_{ij}(\mathbf{h}) = b_{ij}\,\gamma(\mathbf{h}) \qquad \text{for} \quad i,j = 1,\ldots,N$$

where $\mathbf{h}$ is a vector linking pairs of points in geographical space or time and b_{ij} are proportionality coefficients.

In matrix notation this model for the matrix $\boldsymbol{\Gamma}(\mathbf{h})$ of direct and cross variograms $\gamma_{ij}(\mathbf{h})$ is written:

$$\boxed{\boldsymbol{\Gamma}(\mathbf{h}) = \mathbf{B}\,\gamma(\mathbf{h})}$$

where $\mathbf{B}$ is a variance-covariance matrix.

A coregionalization is intrinsically correlated when the *codispersion coefficients*:

$$\mathrm{cc}_{ij}(\mathbf{h}) \;=\; \frac{\gamma_{ij}(\mathbf{h})}{\sqrt{\gamma_{ii}(\mathbf{h})\,\gamma_{jj}(\mathbf{h})}}$$

are constant for any value of $\mathbf{h}$, i.e. do not depend on spatial scale.

With the *intrinsic correlation* model:

$$\mathrm{cc}_{ij}(\mathbf{h}) = \frac{b_{ij} \quad \gamma(\mathbf{h})}{\sqrt{b_{ii}\,b_{jj}} \quad \gamma(\mathbf{h})} = r_{ij}$$

i.e. the correlation r_{ij} between variables is not a function of $\mathbf{h}$.

Intrinsic correlation can be checked using PCA in the following way:

1. Compute principal components for the variable set.
2. Compute the cross-variograms between the first few principal components concentrating most of the total variance.

In the case of *intrinsic correlation*, the cross-variograms between PCs are all zero.

However, if the cross-variograms between PCs are not nil, the principal components are correlated at particular spatial or time scales. In this situation they are only globally orthogonal, but not at any scale, and it is advisable to abandon the intrinsic correlation model in favour of a more sophisticated coregionalization model.

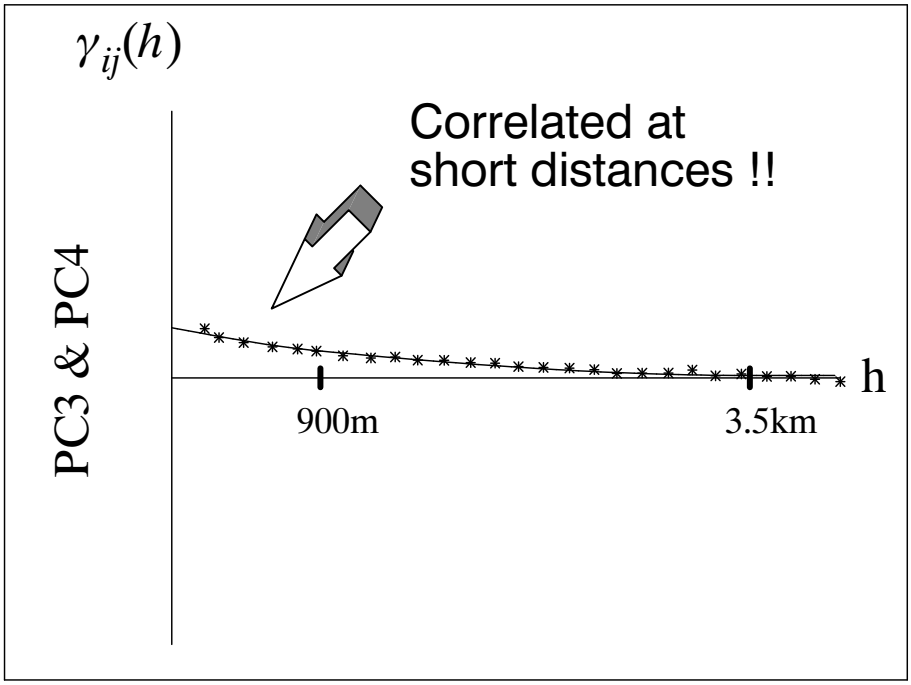

Figure 2.5: Cross-variogram between PCs.

Example: The Figure 2.5 shows an example of a pair of principal components whose cross-variogram only vanishes at large scale. For this data the intrinsic correlation model clearly is not adequate.

A more detailed discussion of this example, explaining in particular how the ordinate of Figure 2.5 has been scaled, is provided in [17].

Chapter 3

Empirical Orthogonal Functions

Empirical Orthogonal Functions (EOFs) are an application of PCA to time series and have been widely used in meteorology and climatology since the 1950s [13, 14, 6]. As multiple or multivariate time series usually are autocorrelated we show that EOFs can only make sense if the data are in accordance with the intrinsic correlation model.

3.1 Definition of EOFs

EOFs are an application of *Principal Component Analysis* to multiple or multivariate time series. We define:

$$\mathbf{z}(t) \;=\; \Big(Z_1(t), \ldots, Z_i(t), \ldots, Z_N(t)\Big)$$

a vector of time dependent second-order stationary random functions $Z_i(t)$ where the index i refers either to different stations (multiple time series) or to different variables at one station (multivariate time series).

The variance of the vector $\mathbf{z}(t)$ is defined as:

$$\mathrm{var}\Big(\mathbf{z}(t)\Big) \;=\; \Big[\,\sigma_{ij}\,\Big] = \mathbf{V},$$

where $\mathbf{V}$ is a variance-covariance matrix.

The correlated $Z_i(t)$ are projected onto uncorrelated *Empirical Orthogonal Functions* $Y_p(t)$ using an orthonormal $N \times N$ matrix $\mathbf{Q}$,

$$\mathbf{z}(t) \;=\; \mathbf{Q}\,\mathbf{y}(t)$$

where $\mathbf{y}(t) = \Big(Y_1(t), \ldots, Y_p(t), \ldots, Y_N(t)\Big)$.

Conversely, we have the equation

$$\mathbf{y}(t) \;=\; \mathbf{Q}^\top\,\mathbf{z}(t)$$

with

$$\mathrm{E}\Big[\,\mathbf{y}(t)\,\Big] \;=\; 0 \quad \text{and} \quad \mathrm{var}\Big(\mathbf{y}(t)\Big) \;=\; \mathbf{\Lambda}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues of $\mathbf{V}$ and $\mathbf{Q}$ is the corresponding matrix of eigenvectors with

$$\mathbf{Q}^{\top}\,\mathbf{Q} \;=\; \mathbf{I}$$

3.2 EOFs and intrinsic correlation

The problem with EOF analysis is to know whether the matrix function $\mathbf{C}_Y(\tau)$, i.e. the matrix of cross-covariance functions between EOFs,

$$\mathrm{cov}\Big(\mathbf{y}(t),\mathbf{y}(t+\tau)\Big) \;=\; \Big[\,C_{pq}(\tau)\,\Big] \;=\; \mathbf{C}_Y(\tau),$$

has a *diagonal* structure?

The intrinsic correlation model for $\mathbf{z}$ is:

$$\mathrm{cov}\Big(\mathbf{z}(t),\mathbf{z}(t+\tau)\Big) \;=\; \mathbf{C}_Z(\tau) \;=\; \mathbf{V}\,\rho(\tau)$$

where $\rho(\tau)$ is an autocorrelation function.

The intrinsic correlation model for the EOFs is then

$$\boxed{\mathbf{C}_Y(\tau) \;=\; \mathbf{\Lambda}\,\rho(\tau)}$$

so that the EOFs are orthogonal whatsoever the time-scale τ.

Principal Component Analysis was originally designed for the iid (independent identically distributed random variables) model, which is a particular case of the intrinsic correlation model. The intrinsic correlation model allows to extend PCA to the case of autocorrelated (i.e. non-independent) data. The question is now: do the data follow the intrinsic correlation model? How can we check this?

3.3 Checking for intrinsic correlation

The first test that can be applied is to check graphically whether the sample codispersion functions

$$\mathrm{cc}^{\star}_{ij}(\tau) \;=\; \frac{\gamma^{\star}_{ij}(\tau)}{\sqrt{\gamma^{\star}_{ii}(\tau)\,\gamma^{\star}_{jj}(\tau)}}$$

are constant and equal to the correlation coefficient $r^{\star}_{ij}$.

When N is large, there is a large number of codispersion functions to plot and so the following second test is more straightforward. It consists in checking whether the cross-covariance functions between sample EOFs, $C^{\star}_{pq}(\tau)$, are zero for any lag τ when $p \neq q$.

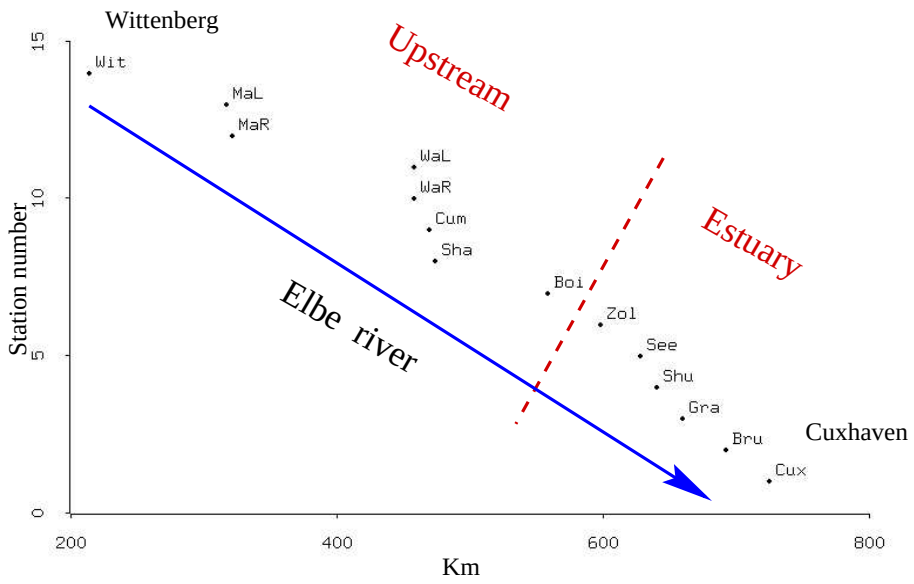

Figure 3.1: Position of the stations along the river Elbe.

3.4 Cross-covariance function and variogram

We recall briefly the relations between the cross-covariance function and the cross-variogram [17]. The *cross-covariance function* can be split into an *even* and an *odd* term:

$$C_{ij}(\tau) = \underbrace{\frac{1}{2}\Big(C_{ij}(\tau)+C_{ij}(-\tau)\Big)}_{\text{even term}} + \underbrace{\frac{1}{2}\Big(C_{ij}(\tau)-C_{ij}(-\tau)\Big)}_{\text{odd term}}$$

The *cross-variogram*

$$\gamma_{ij}(\tau) = \frac{1}{2}\,\mathrm{E}\Big[\Big(Z_i(t+\tau)-Z_i(t)\Big)\cdot\Big(Z_j(t+\tau)-Z_j(t)\Big)\Big]$$

is an *even* function. In the case of second-order stationarity it corresponds to the *even* part of the cross-covariance function

$$\gamma_{ij}(\tau) = \sigma_{ij} - \frac{1}{2}\Big(C_{ij}(\tau)+C_{ij}(-\tau)\Big)$$

	Wit	MaL	MaR	WaL	WaR	Cum	Sha	Boi	Zol	See	Shu	Gra	Bru	Cux
Wit	1													
MaL	.90	1												
MaR	.95	.96	1											
WaL	.91	.91	.91	1										
WaR	.84	.85	.89	.88	1									
Cum	.91	.90	.92	.93	.88	1								
Sha	.92	.88	.89	.93	.80	.88	1							
Boi	.90	.86	.86	.89	.73	.84	.93	1						
Zol	.92	.87	.91	.93	.84	.92	.93	.91	1					
See	.93	.89	.91	.95	.83	.92	.94	.90	.95	1				
Shu	.87	.83	.87	.94	.83	.88	.88	.85	.92	.96	1			
Gra	.87	.79	.85	.89	.78	.87	.88	.86	.91	.93	.95	1		
Bru	.81	.78	.81	.85	.75	.83	.85	.83	.86	.88	.90	.93	1	
Cux	.57	.60	.59	.61	.59	.57	.62	.63	.63	.66	.65	.60	.76	1

Table 3.1: Correlation matrix between Elbe stations for ammonium.

3.5 Example: Elbe Ammonium data

The data consist of ammonium (NH_4) measurements and stems from 14 stations along the Elbe river. Sampling was performed every other week from December 1993 to December 1996. We use data that have been deseasonalized by BERTINO [1]. At each station the data have been standardized separately, i.e. at each station the samples have been rescaled to have a zero mean and a unit variance.

The locations of the 14 stations along the river Elbe are shown on Figure 3.1, starting from Wittenberg, which is at km214 from the Czech border, and ending at km725 in Cuxhaven at the North Sea. The correlation matrix between the 14 stations for ammonium has been is shown on Table 3.1. These correlations are all positive and above .5.

3.5.1 Correlation between EOFs and stations

The correlations between the first two EOFs and the 14 stations are the following:

	Wit	MaL	MaR	WaL	WaR	Cum	Sha	Boi	Zol	See	Shu	Gra	Bru	Cux
EOF1	.95	.93	.95	.97	.89	.95	.95	.93	.97	.98	.95	.94	.91	.69
EOF2	-.15	-.14	-.16	-.08	-.12	-.14	-.04	.01	-.04	.01	.05	.05	.28	.67

The corresponding circle of correlations is shown on Figure 3.2.

The first EOF can be interpreted as a "size effect", in analogy to the morphometric example of the previous chapter, while the second EOF reproduces roughly the geographical order of the stations and opposes the "upstream" and the "estuarine" sections of the Elbe.

The first two EOFs account for 91% of the total variance. The *screegraph* is shown on Figure 3.3. The screegraph is a plot of the variances of the EOFs (as

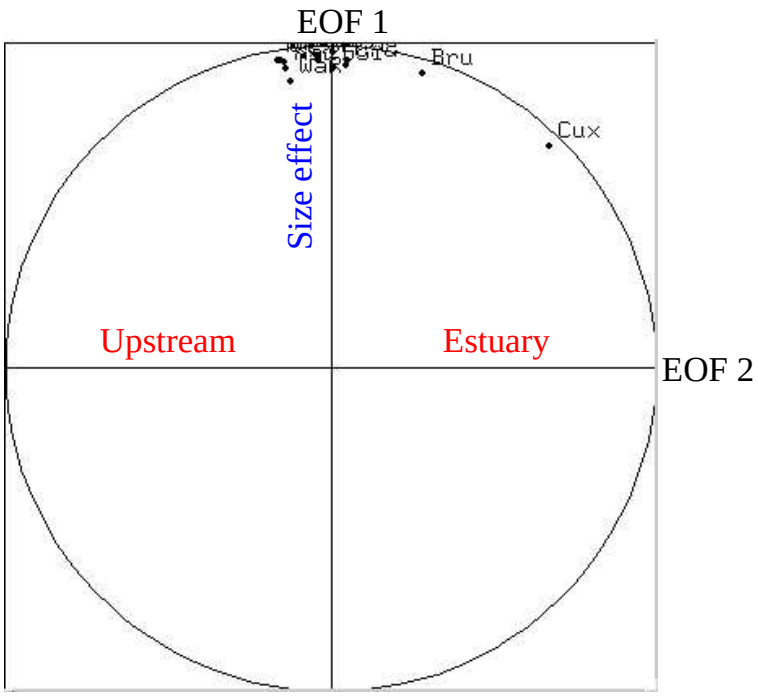

Figure 3.2: Correlation circle for the first two EOFs.

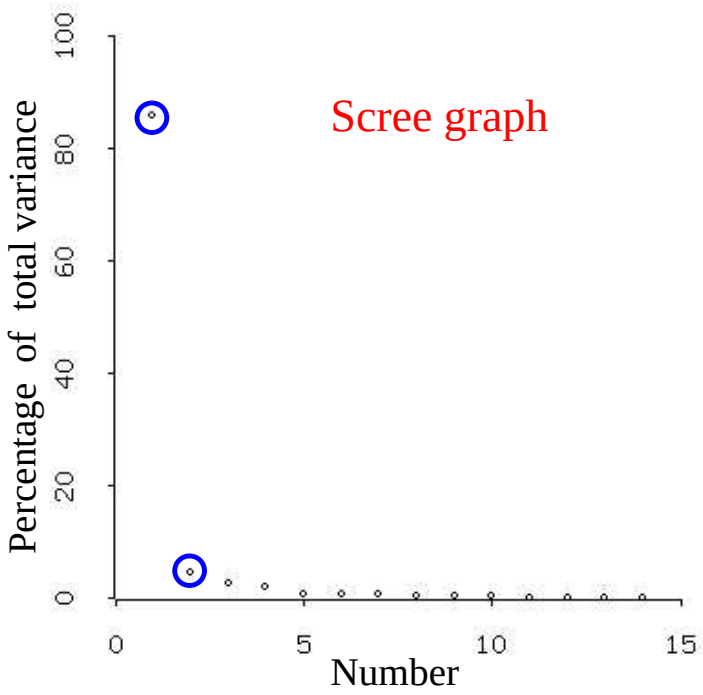

Figure 3.3: Scree graph: plot of the variances of all EOFs in decreasing order. The variances of the first two EOFs are circled.

percentages of the total variance) in decreasing order. The first EOF explains 86% of the total variance, while the second EOF amounts to 4.7%, which is already less than 1/14=7.1%, i.e. the variance contributed by each variable.

3.5.2 A traveling anomaly

Now we plot ammonium against time at the stations Wittenberg and Cuxhaven on Figure 3.4. These stations are located at both ends of the available set. Although the same plots can be generated for all intermediate stations, we shall only display –for the sake of sparing space– the first and the last pictures of the "movie" of ammonium traveling down the river Elbe.

An anomaly can be seen on Figure 3.4 at time steps 1, 2, 3. At station Wittenberg the three values decrease in time. At station Cuxhaven at time step 1, the anomaly is still low; it peaks only at time step 2, two weeks later, which is about the average water transport time from Wittenberg to Cuxhaven.

Please note on the ordinates on Figure 3.4 that the values have been standardized separately at each station and that consequently the anomaly is more outlying (in standardized units) at Wittenberg than it is at Cuxhaven. The anomaly looses some strength diluting progressively in the river as it travels along the 14 stations.

Next we plot the first two EOFs against time on Figure 3.5. The "size effect" EOF tells us that there is an anomaly that starts at time step 1 and decreases with time until time step 3; the anomaly thus has an extension of 4 weeks. The "geographical" EOF shows us that the anomaly starts at time step 1 in the upstream part of the river, reaches the estuary after a fortnight at time step 2 and is entirely in the estuary yet another fortnight later, at time step 3. It is satisfactory that the interpretation of the EOFs enables us to understand both time variation of the size of the anomaly (EOF1) and its traveling through geographical space (EOF2).

3.5.3 Checking for intrinsic correlation

To check for intrinsic correlation we compute the codispersion functions between the stations. The codispersion between the stations Wittenberg and Cuxhaven is displayed on Figure 3.6. Instead of being constant and equal to the correlation coefficient, it starts at a value of .2 and reaches the correlation of $r_{ij} = .57$ at a lag greater than 8 weeks.

Computing the cross-variogram between the first two EOFs on Figure 3.7 does not exhibit truly uncorrelated EOFs, especially at lags beyond half a year.

As the cross-variogram is an even function, it is inappropriate to express the asymetry induced in the time-series by the water transport. Thus we compute the odd term of the cross-correlation function, which is displayed on Figure 3.8.

Clearly this odd term is far from being zero, showing the inadequacy of the cross-variogram, which represents only the even term of the cross-correlation function. Notice that the graph has been scaled using the overall correlation coefficient of $r_{ij} = .57$.

The cross-correlation function between the first two EOFs is displayed on Figure 3.9. The two EOFs show a distinctive time structure, although they are by definition globally uncorrelated!

To try to eliminate the lagged correlation induced by water transport, we shift back by two weeks the Cuxhaven series against the Wittenberg series. The odd term of the cross-correlation function is displayed on Figure 3.10: it has flattened down at the origin for lags below six weeks. The two weeks shift thus explains well the short term lagged correlation. Note that the overall correlation between the two series has increased to $r_{ij} = .64$.

The codispersion function between Wittenberg and Cuxhaven on Figure 3.11 is less steep in the neighborhood of the origin after the backshift, than it was on Figure 3.6 without shifting. The two weeks backshift has thus brought us nearer to the intrinsic correlation model for this pair of stations.

BERTINO [1] has rescaled the time coordinates of all 14 stations using data on water transport times. In this new Lagrangian system of time coordinates the ammonium measurements at the 14 stations appear to be intrinsically autocorrelated.

The transformation of BERTINO turns out to be actually only weakly nonlinear. This explains to us the fact that the EOFs were interpretable in this case study: the time series are approximately intrinsically correlated up to a backshift.

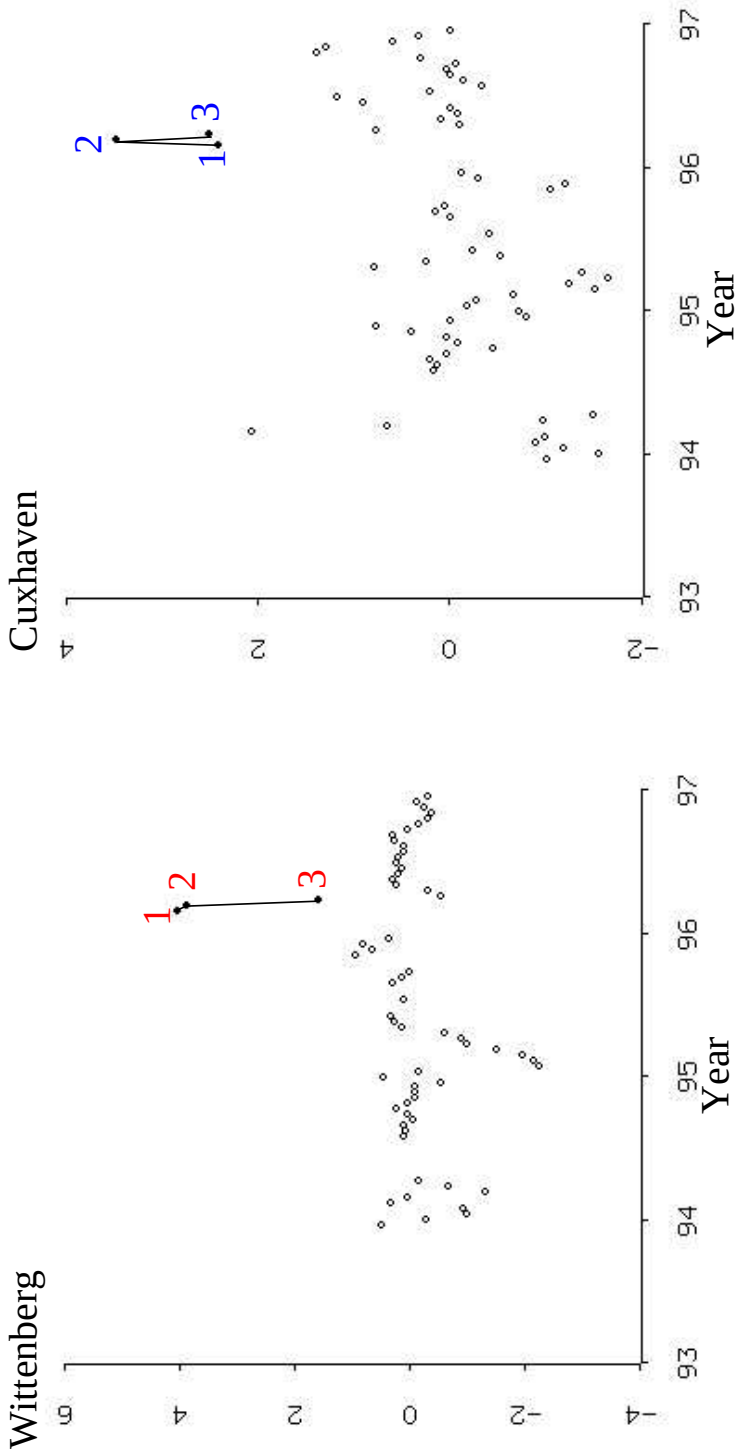

Figure 3.4: Plot of ammonium against time at the stations Wittenberg and Cuxhaven.

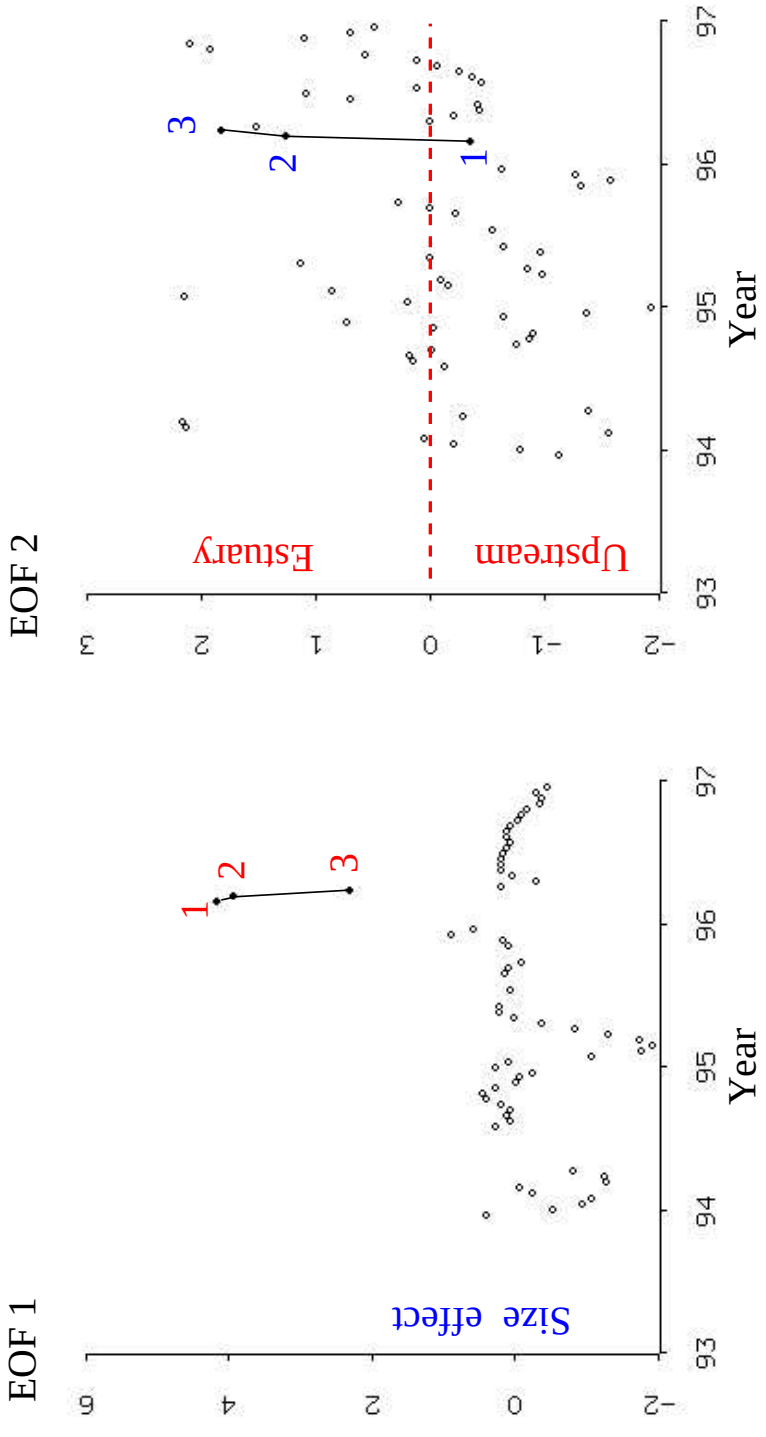

Figure 3.5: Plot of the first two EOFs against time.

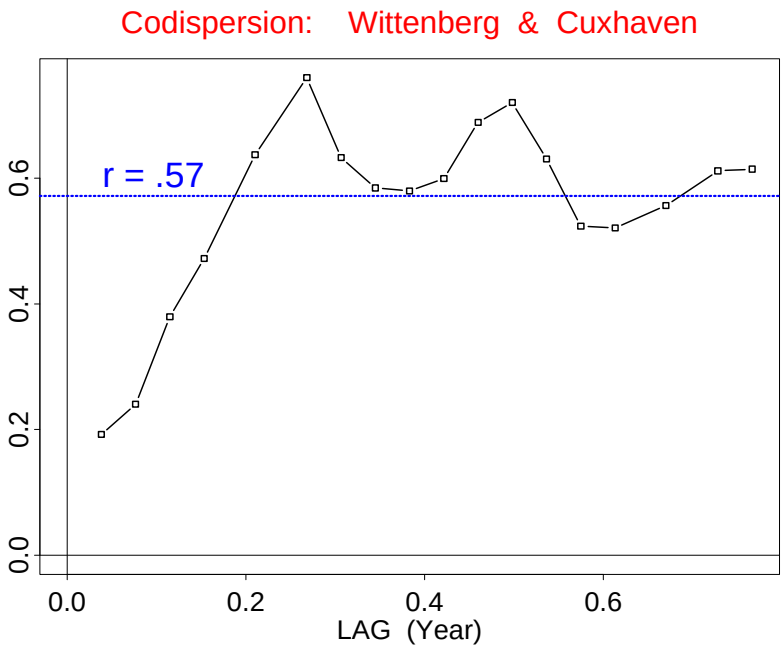

Figure 3.6: Codispersion function between Wittenberg and Cuxhaven stations.

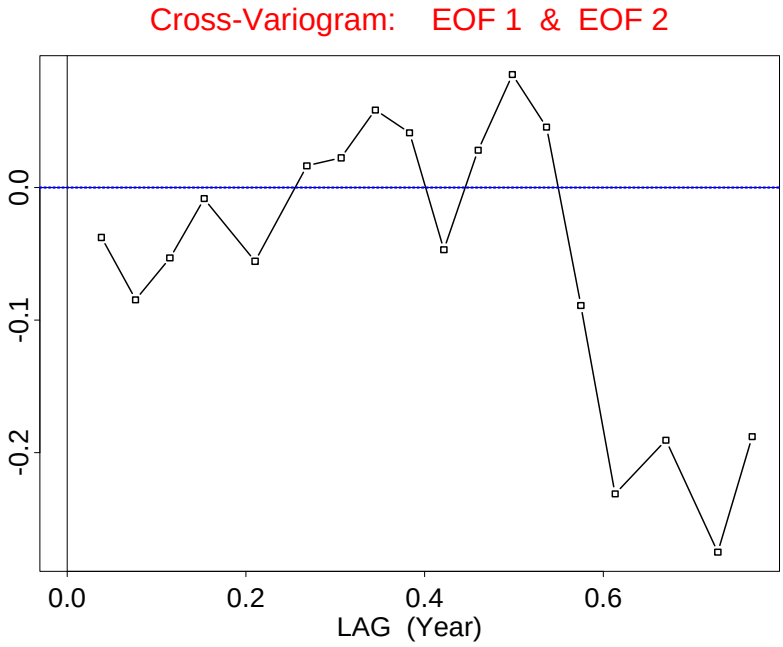

Figure 3.7: Cross-variogram between the first two EOFs.

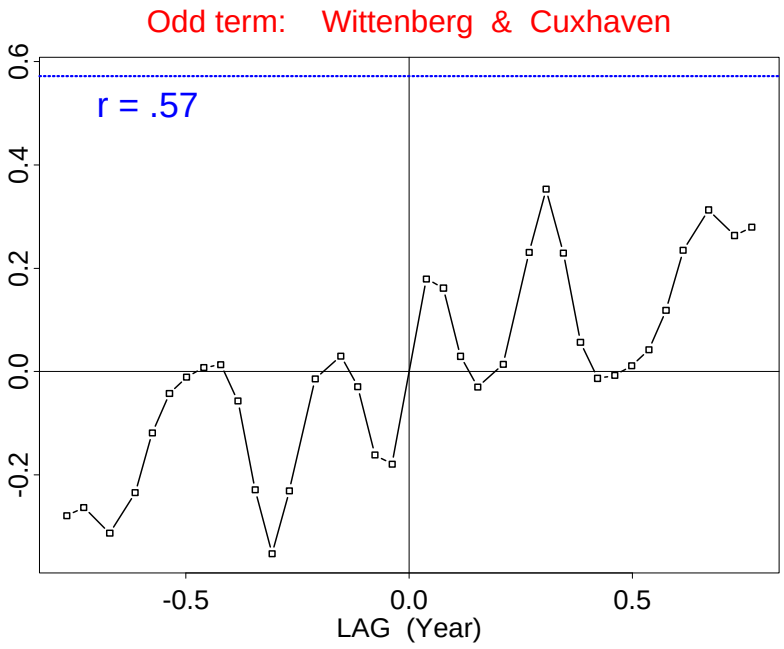

Figure 3.8: Odd term of the cross-correlation function between Wittenberg and Cuxhaven stations.

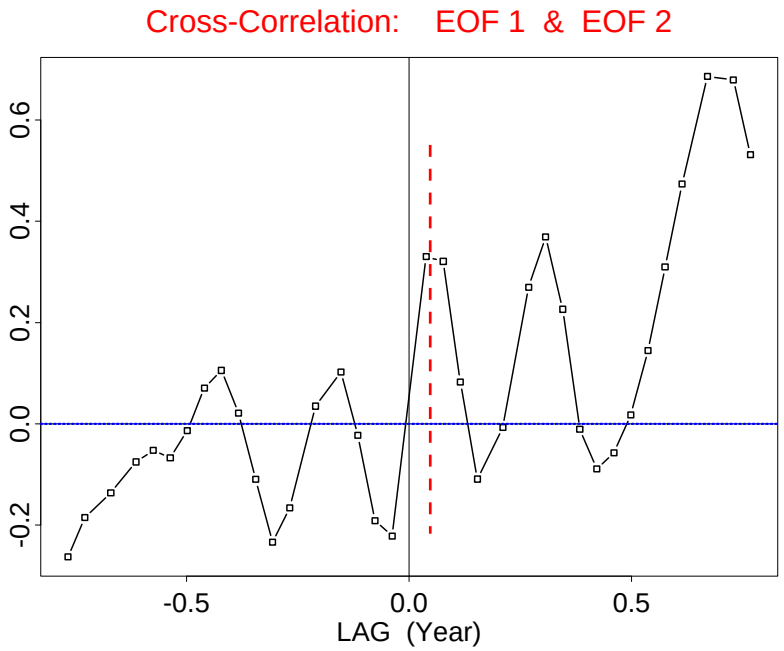

Figure 3.9: Cross-correlation function between the first two EOFs.

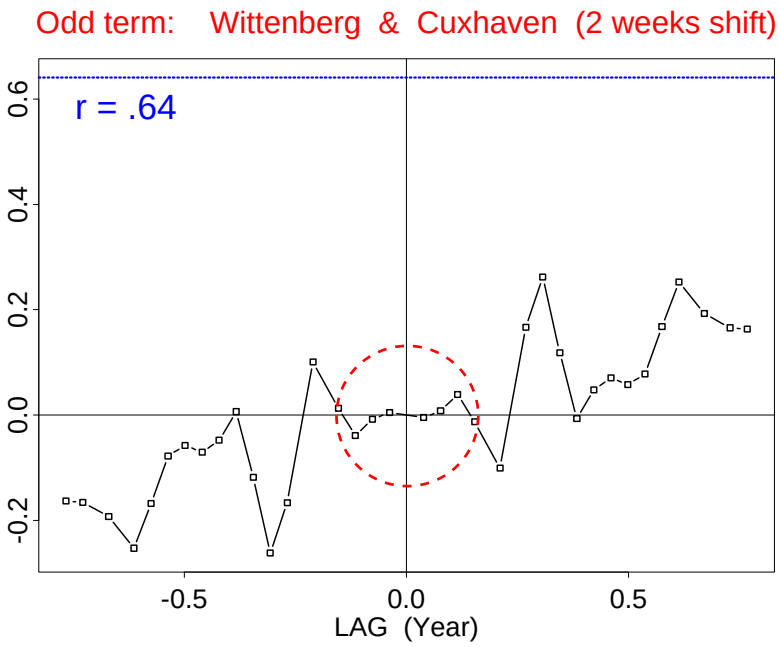

Figure 3.10: Shifting Cuxhaven two weeks back against Wittenberg: resulting odd term of the cross-correlation function.

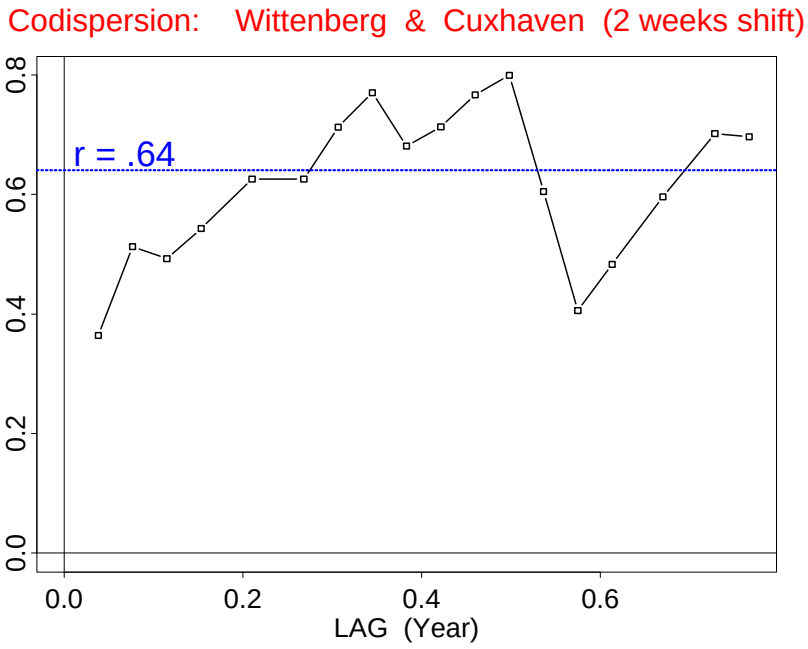

Figure 3.11: Shifting Cuxhaven two weeks back against Wittenberg: resulting codispersion function.

Chapter 4

Extensions

The question which has been left open is: what to do when the data do not comply with the intrinsic correlation model? In this chapter we shall concentrate on geostatistical alternatives.

In geostatistics, methods for characterizing the spatial or temporal variation at different scales of a multivariate system have attracted much attention during the last two decades. Applications have been particularly numerous in soil science [2]. Most work was based on the linear model of coregionalization (LMC) which is suitable for a system that can be described adequately by direct and cross variograms.

More recently, work done on complex kriging (for estimating vector variables in 2D geographical space) has inspired the bilinear model of coregionalization (BMC) which is more general in the sense that it allows the description of a system for which the cross-covariance functions are not necessarily even.

4.1 Cross-covariance functions

Cross-covariance functions are useful in describing the cross-correlations in a set of variables which can be:

- different types of measurements located in space or time,
- measurements of one quantity in a spatial region at different times,
- measurements of one quantity along time at different sites of a spatial region.

The cross-covariance functions are not necessarily even as there can be various kinds of delay or shift effects at different charateristic space or time scales between the variables. At each charateristic scale of index u we shall however assume that the space or time correlation is governed by one correlation function $\rho_u(\mathbf{h})$.

We denote $C_{ij}(\mathbf{h})$ the cross-covariance function between two jointly second-order stationary random functions $Z_i(\mathbf{x})$ and $Z_j(\mathbf{x})$, where $\mathbf{x}$ is the vector of the coordinates of a point in space or time, $\mathbf{h}$ is a vector linking a pair of points in space or time, Z is a real or complex random variable, i and j are indices of a set of N random functions. The matrix $\mathbf{C}_{ij}(\mathbf{h})$ of direct and cross covariance functions for a given set of random functions is charaterized by Cramér's generalization of the Bochner-Khintchine theorem.

4.2 Intrinsic correlation model

The simplest model for real random functions is the following

$$\mathbf{C}(\mathbf{h}) \;=\; \mathbf{V}\,\rho(\mathbf{h})$$

where $\mathbf{V}$ is the matrix of variances and covariances σ_{ij} and $\rho(\mathbf{h})$ is a direct correlation function.

It is called the *intrinsic correlation* model because the correlation between two random functions

$$\frac{\sigma_{ij}\,\rho(\mathbf{h})}{\sqrt{\sigma_{ii}\,\rho(\mathbf{h})\,\sigma_{jj}\,\rho(\mathbf{h})}} \;=\; \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\,\sigma_{jj}}} \;=\; r_{ij}$$

does not depend upon spatial scale.

The linear model associated to the intrinsic correlation model is written

$$\boxed{Z_i(\mathbf{x}) \;=\; \sum_{p=1}^{N} a_{pi}\,Y_p(\mathbf{x})}$$

where $Y_p(\mathbf{x})$ are N uncorrelated random functions whose direct covariance functions $\rho(\mathbf{h})$ do not depend on the index p and a_{pi} are transformation coefficients.

From a known intrinsic correlation model, one possible method to specify the transformation coefficients is based on the eigenvalue decomposition of the variance-covariance matrix $\mathbf{V}$ and the factors Y_p can then be interpreted as principal components.

The intrinsic correlation model is an important reference case when the variables are all sampled at the same locations, because their estimation is simplified [16, 17].

4.3 Linear model of coregionalization

A more sophisticated model for a set of real random functions is the multivariate nested covariance function model

$$\mathbf{C}(\mathbf{h}) \;=\; \sum_{u=0}^{S} \mathbf{B}_u\,\rho_u(\mathbf{h})$$

where u is an index for a set of $S+1$ characteristic spatial or temporal scales and the *coregionalization matrices* $\mathbf{B}_u$ are variance-covariance matrices describing multivariate correlation at these charateristic scales of the phenonenon.

The associated random function model is the *linear model of coregionalization* (LMC)

$$Z_i(\mathbf{x}) = \sum_{u=0}^{S} \sum_{p=1}^{N} a_{piu}\, Y_{pu}(\mathbf{x})$$

where a set of N uncorrelated factors is defined at each of the $S+1$ charateristic scales. A possibility to specify the LMC from a known multivariate nested covariance function model is by performing a principal component analysis based on the eigenvalue decomposition of the coregionalization matrices which yields the transformation coefficients a_{piu}.

4.4 PCA on the basis of an LMC

The standard PCA model is only adequate for data that can be viewed as a realization of the intrinsic correlation model. The LMC covers a more general class of phenomena that can be described with a nested multivariate variogram. In this model we have different correlation structures at different charateristic spatial or time scales. A coregionalized PCA is performed separately for each of these scales.

The eigenanalysis is performed on each coregionalization matrix $\mathbf{B}_u$ of the nested multivariate variogram. The correlation coefficients r_{ipu} between the original variables and the principal components at a given scale u can be used to construct correlation circles to understand the correlation structure at that spatial or time scale.

Estimates $Y^{\star}_{pu}(\mathbf{x})$ of principal components are obtained by cokriging and can be plotted as geographical maps.

Coregionalized PCA has been successfully applied in numerous case studies (see [17, 2] for references). Coregionalized EOFs have first been applied in [12].

4.5 Complex LMC

The real linear model of coregionalization has the limitation that it can only serve to model a set of covariance functions in which the cross-covariance functions are even. It is necessary to introduce a complex linear model of coregionalization and to take its real part, the bilinear model of coregionalization, when the cross-covariance functions in a real covariance function matrix are not even.

In geostatistics, the estimation of two dimensional vector variables as complex variables was studied by LAJAUNIE AND BÉJAOUI [7], who examined ways of

modeling a complex covariance function. This work inspired the formulation of the bilinear model of coregionalization [3, 17].

The complex analogue to the intrinsic correlation model is

$$\mathbf{C}(\mathbf{h}) \;=\; \mathbf{B}\,\rho(\mathbf{h}) \;=\; \mathbf{E}\,\chi(\mathbf{h}) - \mathbf{F}\,\kappa(\mathbf{h}) + \mathrm{i}\left(\mathbf{E}\,\kappa(\mathbf{h}) + \mathbf{F}\,\chi(\mathbf{h})\right)$$

where $\rho(\mathbf{h}) = \chi(\mathbf{h}) + \mathrm{i}\,\kappa(\mathbf{h})$ is a scalar complex covariance function and $\mathbf{B}$ is a hermitian positive semi-definite matrix with $\mathbf{B} = \mathbf{E} + \mathrm{i}\,\mathbf{F}$. The matrix $\mathbf{E}$ is a symmetric positive semi-definite matrix while $\mathbf{F}$ is antisymmetric.

Naturally we can consider a nested complex multivariate covariance function model of the type $\mathbf{C}(\mathbf{h}) = \sum_u \mathbf{B}_u\,\rho_u(\mathbf{h})$ with a corresponding complex LMC.

4.6 Bilinear model of coregionalization

The real LMC is in particular not adequate for multivariate time series analysis where delay effects or phase shifts are common and cannot be included in a model with even cross covariances. A model for real random functions with non even real cross covariance functions can be derived from the complex LMC by taking its real part. We obtain the *bilinear model of coregionalization* (BMC) which is composed of the linear combination of two sets of factors $U_{pu}(\mathbf{x})$ and $V_{pu}(\mathbf{x})$ with two sets of transformation coefficients c_{pui} and d_{pui},

$$\boxed{Z_i(\mathbf{x}) \;=\; \sum_{u=0}^{S}\sum_{p=1}^{N}\left(c_{pui}\,U_{pu}(\mathbf{x}) - d_{pui}\,V_{pu}(\mathbf{x})\right)}$$

In the case of only one spatial scale (the nested case is analog) we can drop the index u and have the BMC

$$Z_i(\mathbf{x}) \;=\; \sum_{p=1}^{N}\left(c_{pi}\,U_p(\mathbf{x}) - d_{pi}\,V_p(\mathbf{x})\right)$$

with

$$\sum_{p=1}^{N}\left(c_{pi}\,c_{pi} + d_{pi}\,d_{pi}\right) \;=\; e_{ij} \quad \text{and} \quad \sum_{p=1}^{N}\left(c_{pi}\,d_{pi} - c_{pi}\,d_{pi}\right) \;=\; f_{ij}$$

where e_{ij} and f_{ij} are respectively the elements of matrices $\mathbf{E}$ and $\mathbf{F}$ such that $\mathbf{B} = \mathbf{E} + \mathrm{i}\,\mathbf{F}$.

Restraining the covariance functions for $U(\mathbf{x})$ and $V(\mathbf{x})$ to be of the form

$$C_{UU}(\mathbf{h}) = C_{VV}(\mathbf{h}) \;=\; \frac{1}{2}\,\chi(\mathbf{h}) \quad \text{and} \quad C_{UV}(-\mathbf{h}) = -C_{UV}(\mathbf{h}) \;=\; \frac{1}{2}\,\kappa(\mathbf{h})$$

where $\chi(\mathbf{h})+\mathrm{i}\,\kappa(\mathbf{h})$ is a complex covariance function with an odd imaginary part, the cross-covariance function between two real variables is

$$C_{ij}(\mathbf{h}) \;=\; \sum_{p=1}^{N} \left(\frac{c_p^i\, c_p^j + d_p^i\, d_p^j}{2}\, \chi(\mathbf{h}) \;-\; \frac{c_p^j\, d_p^i - c_p^i\, d_p^j}{2}\, \kappa(\mathbf{h}) \right)$$

The multivariate covariance function model associated to the BMC is thus

$$\mathbf{C}(\mathbf{h}) \;=\; \frac{1}{2} \Big(\mathbf{E}\, \chi(\mathbf{h}) - \mathbf{F}\, \kappa(\mathbf{h}) \Big)$$

and is real. See GRZEBYK [3] for details on fitting algorithms and an example of the fit of a covariance model based on a BMC to data from three remote sensing channels of a Landsat satellite.

Chapter 5

Conclusion

Principal Component Analysis (PCA) is used for:

- data compression
- multivariate outlier detection,
- deciphering a correlation matrix,
- identifying underlying factors,
- detecting intrinsic correlation.

In applications that do not fit the intrinsic correlation model, PCA is a powerful tool for defining linear coregionalization models (LMC, BMC, complex LMC) to:

- describe, analyze and interpret the correlation structure of a multivariate spatial or temporal system,
- estimate or simulate components of such a system.

Thus both standard and coregionalized PCA are basic and essential tools for geostatisticians dealing with multidimensional data sets.

References

[1] Bertino L (1998) *Short-term prediction of Elbe nutrient concentrations.* Report S-369, Centre de Géostatistique, Ecole des Mines de Paris, Fontainebleau, 70p.

[2] Goovaerts P (1997) *Geostatistics for Natural Resources Evaluation.* Oxford University Press, Oxford, 487p.

[3] Grzebyk M (1993) *Ajustement d'une Corégionalisation Stationnaire.* Doctoral thesis, Ecole des Mines, Paris, 154p.

[4] Grzebyk M & Wackernagel H (1994) *Multivariate analysis and spatial/temporal scales: real and complex models.* Proceedings of XVIIth International Biometrics Conference, Volume 1, 19–33, Hamilton, Ontario.

[5] Hagen D (1982) Geoexploration, 20, 93–111.

[6] Jolliffe IT (1986) *Principal Component Analysis.* Springer-Verlag, New York, 271p.

[7] Lajaunie C & Béjaoui R (1991) *Sur le krigeage des fonctions complexes.* Publication N-23/91/G, Centre de Géostatistique, Ecole des Mines de Paris, Fontainebleau, 24p.

[8] Lindner S & Wackernagel H (1993) Statistische Definition eines Lateritpanzer-Index für SPOT/Landsat-Bilder durch Redundanzanalyse mit bodengeochemischen Daten. In: Peschel G (ed) Beiträge zur Mathematischen Geologie und Geoinformatik, Bd 5, 69–73, Sven-von-Loga Verlag, Köln.

[9] Morrison DF (1978) *Multivariate Statistical Methods.* Second Edition, McGraw-Hill International, Auckland, 415p.

[10] Roquin C, Dandjinou T, Freyssinet P & Pion JC (1989) The correlation between geochemical data and SPOT satellite imagery of lateritic terrain in Southern Mali. Journal of Geochemical Exploration, 32, 149–168.

[11] ROQUIN C, FREYSSINET P, ZEEGERS H & TARDY Y (1990) Element distribution patterns in laterites of southern Mali: consequence for geochemical prospecting and mineral exploration. Applied Geochemistry, 5, 303–315.

[12] ROUHANI S & WACKERNAGEL H (1990) Multivariate geostatistical approach to space-time data analysis. Water Resources Research, 26, 585–591.

[13] STORCH H VON & NAVARRA A, editors (1995) *Analysis of Climate Variablility*. Springer Verlag, Berlin, 334p.

[14] STORCH H VON & ZWIERS (1998) *Statistical Analysis in Climate Research*. Cambridge University Press, 528p.

[15] VENARD C (1998) *Multivariate and spatial analysis of chemical elements of the river Elbe*. Report S-370, Centre de Géostatistique, Ecole des Mines de Paris, Fontainebleau, 52p.

[16] WACKERNAGEL H (1994) Cokriging versus kriging in regionalized multivariate data analysis. Geoderma, 62, 83–92.

[17] WACKERNAGEL H (1998) *Multivariate Geostatistics: an Introduction with Applications*. 2nd edition, Springer Verlag, Berlin, 291p.

Appendix A

XLispStat programming example

The XLispStat language was developped by Luke TIERNEY and can be obtained by ftp from the School of Statistics of the University of Minnesota,

http://www.stat.umn.edu

in Macintosh, Unix and Windows versions.

XLispStat is based on the Lisp programming language with its "polish notation" (first the operator, then the arguments) and many parentheses.

In this appendix we present the functions used to analyze the Elbe data.

Standardisation

The function std calculates the mean and standard deviation of a vector of data x and returns the standardised data.

```
(defun std (x)
  (let* ( ( m (mean x))
  ( s (standard-deviation x))
  ( ma (make-array (length x) :initial-element m))
  ( sa (make-array (length x) :initial-element s))
  )
    (/ (- x ma) sa))
  )
```

Reading a data set

The function readelbe reads the Elbe ammonium data in an ASCII file called AmmoElbe.xls. Each station data is standardized using the function std. The position of the measurement stations along the Elbe is stored in the vector Km. The names of the stations are stored in the vector Names. The function mapkm

is executed to plot the position of the stations along the Elbe (see below). The inter-station variance-covariance matrix is computed and stored in the matrix Covmat. The standardized station data vectors are stored in the matrix Datmat.

```
(defun readelbe ()
  (def ammo (read-data-columns
       "AmmoElbe.xls" 15))
  (def Date (select ammo 0))
  (def Wit (std (select ammo 1)))
  (def MaL (std (select ammo 2)))
  (def MaR (std (select ammo 3)))
  (def WaL (std (select ammo 4)))
  (def WaR (std (select ammo 5)))
  (def Cum (std (select ammo 6)))
  (def Sha (std (select ammo 7)))
  (def Boi (std (select ammo 8)))
  (def Zol (std (select ammo 9)))
  (def See (std (select ammo 10)))
  (def Shu (std (select ammo 11)))
  (def Gra (std (select ammo 12)))
  (def Bru (std (select ammo 13)))
  (def Cux (std (select ammo 14)))
  (def Km (list  214.1 318 322 459 459 470 474.5 559 598.7
              628.8 641 660.5 693 725.2))
  (def Names  (list "Wit" "MaL" "MaR" "WaL" "WaR" "Cum" "Sha" "Boi"
                "Zol" "See" "Shu" "Gra" "Bru" "Cux"))
  (mapkm)
  (def Covmat (covariance-matrix Wit MaL MaR WaL WaR Cum Sha Boi
                              Zol See Shu Gra Bru Cux))
  (def Datmat (bind-rows Wit MaL MaR WaL WaR Cum Sha Boi
                      Zol See Shu Gra Bru Cux))
  )
```

Plotting the position of the stations

The function mapkm plots the station numbers (in reverse order) against the locations on the river. The plot-object mapk gets the status "linked" (to the other windows). Dragging the pointer over the window, the names of the stations will appear.

```
(defun mapkm ()
 (def mapk (plot-points Km (reverse (+ (iseq (length Km)) 1)) :point-labels Names :title "Km from border") )
```

```
(send mapk :linked t)
(send mapk :showing-labels t)
)
```

Example: Figure 3.1, p15.

Plotting the correlation circle

The function pcav plots the location of the stations within the correlation cicle in the plane spanned by two EOFs.

The eigenvalues and eigenvectors of the correlation matrix are computed. The EOF indices are decreased by one because in Lisp indices start from zero. The correlations between the EOFs and the stations are obtained by multiplying each eigenvector with the square root of the corresponding eigenvalue.

It should be noted that eigenvectors are defined up to the sign: here, for display reasons, we preferred to invert the sign ((- 0 eigvector)). As sign does not matter, a software should give the user the possibility to invert the sign of each eigenvector if desired.

We first plot the positions (vv1, vv2) of the stations within the unit circle. Then we draw the circle and the axes. The window is set to the status "linked", so that the position of selected stations in the EOF plane can be compared to their corresponding position in geographical space (window mapkm).

Dragging the pointer over the window, the names of the stations will appear.

```
(defun pcav (covmat varnam factor1 factor2 titulum)
  (let* ( (eigval (eigenvalues covmat))
  (eigvec (eigenvectors covmat))
        (ff1 (- factor1 1))
        (ff2 (- factor2 1))
        (vv1 (* (sqrt (select eigval ff1)) (- 0 (select eigvec ff1))))
        (vv2 (* (sqrt (select eigval ff2)) (- 0 (select eigvec ff2))))
  (ax '(0 0))
  (ay '(-1 1))
  (cc (rseq -3.14159 3.14159 100))
  (ccs (sin cc))
  (ccc (cos cc))
  )
    (setf pcavplot (plot-points vv1 vv2 :point-labels varnam))
    (send pcavplot :add-lines ccc ccs)
    (send pcavplot :add-lines ax ay)
    (send pcavplot :add-lines ay ax)
    (send pcavplot :adjust-to-data)
    (send pcavplot :x-axis nil nil 0)
```

```
  (send pcavplot :y-axis nil nil 0 :redraw)
  (send pcavplot :title titulum)
  (send pcavplot :showing-labels t)
  (send pcavplot :linked t)
  )
 )
```

Examples: Figure 3.2, p17; Figure 2.2, p6.

The function pcav1 calls the function pcav twice to draw the correlation circles corresponding to the pairs (1, 2) and (2, 3) of EOFs.

Usually I prefer to draw the first EOF vertically and the second EOF horizontally because it frequently happens that on the first EOF all correlations with the stations have the same sign (e.g. when correlations are all positive between stations) and only the second EOF has discriminatory power (correlations of stations with the second EOF have a different signs). The same is true in the context of PCA for morphometric variables in biology, when the first PC represents a *size effect* and the second PC represents a *shape effect*.

Example: Table on p9.

The user of a software may thus be interested in choosing which axis he calls the abscissa and which one he calls the ordinate. My suggestion is that the first EOF/PC should be, by default, on the ordinate and not on the abscissa.

```
(defun pcav1 ()
 (pcav Covmat Names  2 1 "PCA: F1=vertical  &  F2=horizontal")
 (pcav Covmat Names  3 2 "PCA: F2=vertical  &  F3=horizontal")
 )
```

Plotting the position of the samples in 2D EOF space

The function pcas plots the samples in 2D EOF space.

The eigenvalues and eigenvectors of the variance-covariance matrix are computed.

The transformation coefficients of the station data into EOFs are computed by multiplying the eigenvectors by the inverse of the square root of the eigenvalues.

The EOFs are obtained by multiplying the station data with the transformation coefficients.

The position of the samples in the plane spanned by a pair of EOFs is plotted and the corresponding window is set to the status "linked".

```
(defun pcas (zall covmat factor1 factor2 titulum)
 (let* ( (eigval (eigenvalues covmat))
```

```
 (eigvec (eigenvectors covmat))
        (ff1 (- factor1 1))
        (ff2 (- factor2 1))
        (vv1 (* (/ 1 (sqrt (select eigval ff1))) (- 0 (select eigvec ff1))))
        (vv2 (* (/ 1 (sqrt (select eigval ff2))) (- 0 (select eigvec ff2))))
        (yy1 (matmult vv1 zall))
        (yy2 (matmult vv2 zall))
 )
   (setf pcasplot (plot-points yy1 yy2))
   (send pcasplot :title titulum)
   (send pcasplot :linked t)
   )
 )
```

Example: Figure 2.3, p7.

Spinning the samples in 3D EOF space

The EOFs have the property that they concentrate a significant part of the total variance in a low-dimensional space. It is thus of interest to be able to examine the sample cloud in 3D EOF space.

The function pcaspin allows to rotate the sample cloud in 3D EOF space and to generate a projection of the sample cloud on any arbitrary plane passing through the origin within this space. The eigenvalues and eigenvectors of the variance-covariance matrix are computed. The transformation coefficients of the station data into EOFs are computed by multiplying the eigenvectors by the inverse of the square root of the eigenvalues. The EOFs are obtained by multiplying the station data with the transformation coefficients. The position of the samples in the plane spanned by a pair of EOFs is plotted, the third EOF being at right angle to the screen in starting position. The corresponding window is set to the status "linked" to allow comparison with scatterplots between each EOF and the time axis (spmf).

Three buttons (because for 3D rotation we need the three Euler angles) give the user the opportunity to spin the sample cloud and thus examine it from all possible angles. This is useful for multivariate outlier detection or for identifying subpopulations.

```
(defun pcaspin (zall covmat factor1 factor2 factor3 titulum lf1 lf2 lf3)
  (let* ( (eigval (eigenvalues covmat))
  (eigvec (eigenvectors covmat))
        (ff1 (- factor1 1))
        (ff2 (- factor2 1))
        (ff3 (- factor3 1))
```

```
        (vv1 (* (/ 1 (sqrt (select eigval ff1))) (- 0 (select eigvec ff1))))
        (vv2 (* (/ 1 (sqrt (select eigval ff2))) (- 0 (select eigvec ff2))))
        (vv3 (* (/ 1 (sqrt (select eigval ff3))) (- 0 (select eigvec ff3))))
  )
  (def Y1 (matmult vv1 zall))
  (def Y2 (matmult vv2 zall))
  (def Y3 (matmult vv3 zall))
)
  (setf pcaspin (spin-plot (list Y2 Y1  Y3)
   :variable-labels (list lf2 lf1  lf3) ))
  (send pcaspin :title titulum)
  (send pcaspin :linked t)
  (setf spmf
(scatterplot-matrix (list  Date Y1 Y2 Y3)
    :variable-labels
    (list "Time" lf1 lf2 lf3 ))
)
  (send spmf :linked t)
  )
```

Example: Figure 2.1, p4.

The function pcas1 draws 2D projections of the sample cloud for the EOF pairs (1, 2) and (2, 3). It generates the spin plot window for the EOF triple (1, 2, 3).

```
(defun pcas1 ()
  (pcas Datmat Covmat 2 1 "PCA: F1=vertical  &  F2=horizontal" )
  (pcas Datmat Covmat 3 2 "PCA: F2=vertical  &  F3=horizontal" )
  (pcaspin Datmat Covmat 1 2 3 "PCA: spin F1, F2, F3"  "F1" "F2" "F3")
  )
```

Screegraph

The function pcaelbe generates the whole EOF analysis by calling pcav1 and pcas1. It also generates a screegraph.

The eigenvalues of the variance-covariance matrix are computed. The XLisp-Stat function eigenvalues actually provides them in decreasing order. The screegraph is a plot of the eigenvalue against its number. Usually the eigenvalues are displayed as a percentage of the total variance (which is equal to the sum of the eigenvalues).

```
(defun pcaelbe ()
  (pcav1)
```

```
(pcas1)
(def Eval (eigenvalues Covmat))
(plot-points (+ (iseq (length Eval)) 1) (* (/ Eval (length Eval)) 100)
   :xlab "Number"  :title "Screegraph")
)
```

Example: Figure 3.3, p17.