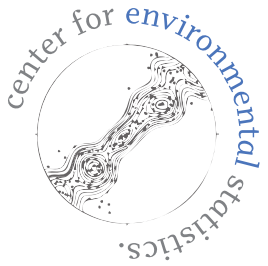


PRINCIPAL COMPONENT ANALYSIS OF A FINITE NUMBER OF CURVES

JAN DE LEEUW

ABSTRACT. We introduce and discuss Principal Component Analysis (FPCA) of curves, using only relatively simple matrix algebra and optimization results. Different loss functions, and various ways to impose constraints on the solution, are also discussed. The techniques are applied first to some theoretical examples involving smooth curves, and subsequently to a data set with one year of hourly ozone measurements in Lebec, Kern County, California. R code for all functions, tables, and figures is included for reproducibility.

Date: September 7, 2008.

2000 *Mathematics Subject Classification.* 62H25.

Key words and phrases. Multivariate Analysis, Principal Component Analysis, Karhunen-Loève Expansion.

CONTENTS

List of Tables

List of Figures

1. Introduction

Principal component analysis (PCA) is one of the most frequently used multivariate data analysis techniques. It has been extended in many different ways to deal with data structures more general than the usual rectangular table of numbers. *Functional PCA* or FPCA, which is the PCA of a finite set of *curves* or *functions*, has been discussed extensively over the last ten years, and several variations have been proposed. Excellent summaries are in Ramsay and Silverman [2002, 2005]; Ferraty and Vieu [2006].

Since curves are generally elements of infinite-dimensional linear spaces the mathematics and statistics of FPCA can easily become quite complicated. Some representative recent papers with much more mathematical detail are [Silverman, 1996; Ocana et al., 1999; Hall and Hosseini-Nasab, 2006; Hall et al., 2006]. The formulas and computations can be kept quite simple, however, by concentrating on the fact that in practice there is only a finite number of curves. As a consequence we can work in finite-dimensional space, use basic matrix algebra, and rapidly get into computational methods.

1.1. **Inner Product of Curves.**

Consider the situation in which we have *curves* f_i $(i = 1, \cdots, n)$ at a finite number of *sites*. Curves are elements of a *real inner product space* $\mathcal{F}$, i.e. a real linear space in which a real-valued inner product $\langle f, g\rangle$ is defined for each pair of elements. For reference, an inner product $\langle \bullet, \bullet\rangle$ is a function on $\mathcal{F} \otimes \mathcal{F}$ with the properties

(1) For all $f \in \mathcal{F}$ the function $\langle f, g\rangle$ is linear in g.
(2) For all $g \in \mathcal{F}$ the function $\langle f, g\rangle$ is linear in f.
(3) For all $f, g \in \mathcal{F}$ $\langle f, g\rangle = \langle g, f\rangle$.
(4) $\langle f, f\rangle \geq 0$ for all $f \in \mathcal{F}$.

If $\langle f, g\rangle = 0$ we say that f and g are *orthogonal*.

Observe that the inner product is a purely algebraic concept. We do not assume any topology, and thus no separability or completeness, and we do not assume any probability structure. We use the words "curves" for the elements of $\mathcal{F}$, because one of the most common interpretations is that the data are n functions of, say, time. Since we are dealing with a number of different curves, we will say they are defined at different "sites", although for some examples "occasions" or "replications" or "individuals" may be more appropriate.

Of course our developments also cover the finite-dimensional case in which the "curves" are just vectors of m numbers, with the inner product $\langle f, g\rangle = f'Wg$ for some positive semi-definite W. An equally important special case, however, is the $\mathcal{L}_2$ case of real-valued functions on an interval T, often interpreted as *time*. We assume a non-negative weight function ω on T, and we consider curves f for which $\int_T \omega(t)f^2(t)dt < \infty$, with inner product

$$\langle f, g\rangle = \int_T \omega(t)f(t)g(t)dt.$$

Both both these special cases the inner product space is actually separable and complete, i.e. it is a separable Hilbert space. But do observe that "curves" are not necessarily real valued functions on the real line, they can be functions from any linear space to any other linear space, as long as the target space has an inner product defined.

1.2. **Reproducing Kernel Hilbert Spaces.**

1.3. **Handling Real Data.** What we have discussed so far is a relatively straightforward generalization of PCA in which the variables can be curves, i.e. take on a possibly infinite number of values. Of course there is no such thing in real data analysis as an observed curve. Data are always discrete and finite, because it is impossible

to collect an infinite number of observations and to measure with infinite precision. So why is FPCA needed at all ?

In general, using the approach of Ramsay and Silverman [2002], FPCA has two phases. In the first phase we produce curves from the (finite and discrete) data. We use a finite number of basis functions to smooth the data. In the second phase we do the PCA on the curves, which amounts to doing a PCA on the coefficients of the basis functions. The first phase, which obviously determines the outcome of the second phase, can be thought of as preprocessing or data handling. It also extends to dealing with missing data, and with transforming existing curves in various way.

It may sound somewhat counter-intuitive to blow up finite dimensional data to infinite-dimensional curves first, and then do a PCA to make them basically finite-dimensional again. But in constructing the curves that are the input for the PCA we take into account notions of smoothness and dependency which will generally make points on the curve depend on neighboring data points, or even on all data points. To use an expression from another area of statistics, in our smoothing of data into curves we are "borrowing strength" by using information from adjacent points (as well as various mathematical modeling notions). Thus it clearly is wrong to think of the first phase as “merely” preprocessing or data handling, it is an essential and critical part of the actual data analysis. In FPCA what the French data analysts call “codage” is of even more importance as in other areas of statistics. In the first partr of this paper, however, there are no data and we assume that curves are given in analytical form.

2. Distances between Curves

2.1. **Distances from Inner Products.** With each pair f_i and f_k we associate the *inner product*

$$c_{ik} \triangleq \langle f_i, f_k \rangle. \tag{1a}$$

The c_{ik} can be collected in an $n \times n$ positive semi-definite matrix C. The matrix C is often called the *Gram matrix* of the curves.

The inner product defines a *pseudo-norm* in the usual way, by

$$\|f_i\| \triangleq \sqrt{\langle f_i, f_i \rangle}, \tag{1b}$$

and the norm defines a *pseudo-distance* by

$$\delta_{ik} \triangleq \|f_i - f_k\|. \tag{1c}$$

We will call δ_{ik} the *curve-distance* between sites. They will be collected in the *curve distance matrix* Δ.

If the pseudo-norm is actually a norm, i.e. if $\|f\| = 0$ if and only if $f = 0$, then of course the pseudo-distance is a distance. Even with proper norms, however, the matrix C can still be singular if the f_i are linearly dependent.

Expanding the squared distance gives

$$\delta_{ik}^2 = \langle f_i - f_k, f_i - f_k \rangle = \|f_i\|^2 + \|f_k\|^2 - 2\langle f_i, f_k \rangle = c_{ii} + c_{kk} - 2c_{ik}. \tag{2a}$$

If we use the unit vectors e_i (i.e. vectors of which only element i is non-zero and equal to one) then

$$\delta_{ik}^2 = (e_i - e_j)'C(e_i - e_j). \tag{2b}$$

Obviously the curve-distances do not change if we replace the curves f_i by curves in deviation from the mean curve, i.e. by

$$\tilde{f}_i \triangleq f_i - f_\bullet$$

with

$$f_\bullet \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i.$$

In fact any translation of the curves will lead to the same distances. The matrix $\tilde{C}$ with elements $\langle \tilde{f}_i, \tilde{f}_k \rangle$ is doubly-centered, i.e. its rows and columns add up to zero. Let u be a vector with all elements equal to one, and let $J = I - \frac{uu'}{u'u}$ be the *centering operator*. The number of elements of u and J will be clear from the context. Then

$$\tilde{C} = JCJ, \tag{3a}$$

and, more surprisingly perhaps,

$$\tilde{C} = -\frac{1}{2}J\Delta^{(2)}J, \tag{3b}$$

where $\Delta^{(2)}$ is the matrix with the squared curve-distances. Thus we see that the doubly-centered Gram matrix can be computed by doubly-centering the Gram matrix, but also by doubly-centering the squared curve distances. In classical multidimensional scaling (MDS) the transformation (3b) is usually attributed to Torgerson [1958].

2.2. **Approximation from Below.** In this section we introduce FPCA by extending the approach to PCA discussed by De Leeuw and Meulman [1985, 1986]. This treats PCA as a special case of metric MDS..

Suppose W is a weight matrix of order n. We suppose that W is non-negative, symmetric, and hollow (i.e. has a zero diagonal). Also define

$$V \triangleq \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}A_{ij},$$

where

$$A_{ij} \triangleq (e_i - e_k)(e_i - e_k)'.$$

The matrices A_{ij} are doubly centered, with elements (i,i) and (j,j) equal to +1, and elements (i,j) and (j,i) equal to -1. All other

elements are equal to zero. Thus V is also doubly centered, with non-positive off-diagonal elements. In addition V is singular and positive semi-definite with $\textbf{rank}(V) \leq n-1$. If all w_{ij} are positive then V actually has rank equal to $n-1$, with only the vector u in the null-space.

The problem we want to solve is to find an $n \times n$ matrix X such that $X'VX$ is diagonal and

$$d_{ik}^2(X) \overset{\Delta}{=} (e_i - e_k)'XX'(e_i - e_k) = \delta_{ik}^2. \tag{4}$$

Thus the curve-distances δ_{ik} between f_i and f_k must be equal to the Euclidean distances $d_{ik}(X)$ between the rows x_i and x_k.

The problem of finding X such that $C = XX'$ and $X'VX$ is diagonal is solved in Appendix B. The minimal solution X, i.e. the solution with minimum rank, is $X = K\Lambda L'$. Here K and Λ come from the spectral decomposition $C = K\Lambda^2K'$, in other words K and Λ^2 come from the symmetric eigenvalue problem $CK = K\Lambda^2$. The orthonormal matrix L is defined by the spectral decomposition $\Lambda K'VK\Lambda = L'\Omega^2L$, i.e. from solving the symmetric eigenvalue problem $K'VKL = L\Omega^2$. Alternatively, we can define X and Ω^2 directly by solving the symmetric eigenvalue problem

$$V^{1/2}CV^{1/2}\tilde{X} = \tilde{X}\Omega^2$$

with $\tilde{X}'\tilde{X} = I$, and then defining $X = V^{1/2}\tilde{X}$. Observe that if all weights are equal then $L = I$ and $\Omega^2 = \Lambda^2$.

Suppose $\omega_1^2 \geq \cdots \geq \omega_r^2$, and define X_p to be the first p columns of X, those associated with the p largest eigenvalues. Then

$$d_{ik}^2(X_1) \leq d_{ik}^2(X_2) \leq \cdots \leq d_{ik}^2(X_r) = \delta_{ik}^2. \tag{5}$$

Moreover

$$\sum_{i=1}^n\sum_{k=1}^n w_{ij}d_{ik}^2(X_p) = \textbf{tr}\ X_p'VX_p = \sum_{s=1}^p \omega_s^2,$$

and thus

$$\sum_{i=1}^{n}\sum_{k=1}^{n} w_{ij}(\delta_{ik}^2 - d_{ik}^2(X_p)) = \sum_{s=p+1}^{r} \omega_s^2.$$

We see that we approximate the squared curve-distances δ_{ik}^2 *from below* by squared Euclidean distances $d_{ik}^2(X_p)$, and that using additional components of the spectral decomposition will always improve the fit. Approximation is always from below because we project the curves on the hyperplane spanned by the first p principal components, and the distance between the projections is always smaller than the distance between the original points.

3. Principal Curves

3.1. **Decomposition.** Assume we have a minimal X satisfying $\textbf{rank}(X) = \textbf{rank}(C) = r$ as well as $XX' = C$ and $X'VX = \Omega^2$. Set $Y \triangleq X(X'X)^{-1}$. If $X = K\Lambda L'$ is defined by spectral decompositions, as in the previous section, then $Y = K\Lambda^{-1}L'$. Observe that $CY = X$ and $Y'CY = Y'X = I$.

The *principal curves* are

$$\psi_s \triangleq \sum_{i=1}^{r} y_{is} f_i.$$

Principal curves are orthogonal, since

$$\langle \psi_s, \psi_t \rangle = \{Y'CY\}_{st} = \begin{cases} 1 & \text{if } s = t, \\ 0 & \text{if } s \neq t. \end{cases}$$

Principal loadings are defined as the inner products of the original curves and the principal curves. Thus

$$\langle f_i, \psi_s \rangle = \{CY\}_{is} = x_{is}.$$

Using the loadings and principal curves we can define *curve decompositions*

$$\hat{f}_i \triangleq \sum_{s=1}^{r} x_{is} \psi_s, \tag{6}$$

and we see that

$$\langle \hat{f}_i, \hat{f}_k \rangle = \sum_{s=1}^{r} \sum_{t=1}^{r} x_{is} x_{kt} \langle \psi_s, \psi_t \rangle = \sum_{s=1}^{r} x_{is} x_{ks} = c_{ik}.$$

Moreover $\|f_i - \hat{f}_i\|^2 = 0$, because $\|f_i\| = \|\hat{f}_i\| = c_{ii}$, and also

$$\langle f_i, \hat{f}_i \rangle = \sum_{s=1}^{r} x_{is} \langle f_i, \psi_s \rangle = \sum_{s=1}^{r} x_{is}^2 = c_{ii}.$$

Thus if the inner product actually defines a norm (and not a pseudo-norm) then $f_i = \hat{f}_i$ and (6) gives us the singular value decomposition (or the *Karhunen-Loève expansion*) of our set of curves. If fact,

we have shown that

$$f_i = \sum_{s=1}^{r} x_{is}\psi_s. \tag{7}$$

Observe this expansion is always defined, but the one we actually compute depends both on the inner product and on the weight matrix V. A different choice of weights and inner products will lead to a different decomposition.

3.2. **Truncation.** In practice we will not use all eigenvalues, but we will truncate the expansion. Suppose $X_p \stackrel{\Delta}{=} K_p\Lambda_p L'$, where K_p and Λ_p corrresponds with the p largest eigenvalues. Again L is chosen by diagonalizing $\Lambda_p K_p' V K_p \Lambda_p$. Let $Y_p \stackrel{\Delta}{=} K_p\Lambda_p^{-1}L'$ and $C_p \stackrel{\Delta}{=} X_p X_p'$, with elements c_{ik}^p. Then $CY_p = X_p$ and thus $Y_p' C_p Y_p = I$, as before. The p principal curves computed with Y_p are just the first p "dominant" principal curves of the original r principal curves. Define the *curve approximations* by

$$\hat{f}_i^p = \sum_{s=1}^{p} x_{is}\psi_s.$$

Then

$$\|f_i - \hat{f}_i^p\|^2 = c_{ii} - c_{ii}^p, \tag{8a}$$

and consequently

$$\sum_{i=1}^{n} \|f_i - \hat{f}_i^p\|^2 = \mathbf{tr}\ (C - C_p) = \sum_{s=p+1}^{r} \lambda_s^2. \tag{8b}$$

Again we see from (8a) that the approximation is from below, and that approximations with different dimensionalities are nested.

3.3. **Rank Approximation.** There is another, more direct. way to arrive at principal curves. It does not use the notion of curve-distance, it just finds some basis functions which are optimal in the least squares sense. We use the notation and results from Appendix A. Thus we first compute the Gram-Schmidt basis $\{\breve{f}_t\}_{t=1}^r$ for the $\{f_i\}_{i=1}^n$. Here r is the *rank* of the f_i.

Theorem 3.1 (Rank Approximation)**.** *Suppose we want to minimize*

$$\sigma(X,\Psi)\stackrel{\Delta}{=}\sum_{i=1}^n\|f_i-\sum_{s=1}^p x_{is}\psi_s\|^2$$

over all $n\times p$ matrices X and over all curves ψ_s such that $\langle\psi_s,\psi_t\rangle=\delta^{st}$. The solution is given by the principal curves and principal loadings associated with the p largest eigenvalues of C.

Proof. We write $f_i=\sum_{t=1}^r z_{it}\breve{f}_t$, where Z is $n\times r$, and $\psi_s=\sum_{t=1}^r y_{ts}\breve{f}_t+\eta_s$, where Y is $r\times p$, and $\langle\breve{f}_t,\eta_s\rangle=0$ for all s and t. Then

$$\begin{aligned}\sigma(X,\Psi)&=\sum_{i=1}^n\|\sum_{t=1}^r(z_{it}-\sum_{s=1}^p x_{is}y_{ts})\breve{f}_t\|^2+\sum_{i=1}^n\|\sum_{t=1}^r x_{is}\eta_s\|^2=\\&=\sum_{i=1}^n\sum_{t=1}^r(z_{it}-\sum_{s=1}^p x_{is}y_{ts})^2+\sum_{i=1}^n\|\sum_{t=1}^r x_{is}\eta_s\|^2.\end{aligned}$$

This is easy to minimize this over the η_s, because we can just set them equal to zero. The loss function must still be minimized over all X and over all Y such that $Y'Y=I$. As Appendix C shows, this means we must choose $Y=L_pT$, where L_p are the eigenvectors of $Z'Z$ corresponding with the p largest eigenvalues Λ_p^2, and T is an arbitrary rotation matrix of order p. This implies that the optimal X is $X=ZY=ZL_pT=K_p\Lambda_pT$, where K_p are the eigenvectors of $C=ZZ'$ corresponding with the p largest eigenvalues Λ_p^2. The minimum value of the loss function is $\sum_{s=p+1}^r\lambda_s^2$. □

We can identify T, as before, by requiring that $X'VX=I$ for some weight matrix V. Or, as suggested by Ramsay and Silverman [2005], we can apply a rotation method such as *varimax*. The default will usually be to simply take $T=I$, in which case we have $X'X=\Lambda_p^2$.

3.3.1. *Choice of Basis.* If one does not want to use the Gram-Schmidt basis $\{\breve{f}_t\}_{t=1}^r$ for the f_i, but one prefers to use the $\{f_i\}_{i=1}^n$ themselves as a basis (or at least as a spanning set), then the problem

of minimizing

$$\sigma(X,Y) \stackrel{\Delta}{=} \mathbf{tr}\ (Z - XY')(Z - XY')' \tag{9a}$$

over X and Y such that $Y'Y = I$ becomes the problem of minimizing

$$\sigma(X,Y) \stackrel{\Delta}{=} \mathbf{tr}\ (I - XY')C(I - XY')' \tag{9b}$$

over X and Y such that $Y'CY = I$.

Observe that at the optimum we must have $X = CY$ and thus $X'Y = I$. It follows that XY' is the orthogonal projector $\Pi_p \stackrel{\Delta}{=} K_p' K_p'$ of rank p. The approximation problem in this section can also be formulated quite simply as the problem of finding an orthogonal projector Π of rank p such that $\rho(\Pi) \stackrel{\Delta}{=} \mathbf{tr}\ C\Pi$ is maximized. No matter which particular basis one chooses, one either has to calculate the dominant eigenvalues and corresponding eigenvectors of C or the dominant singular values and corresponding singular vectors of Z.

3.3.2. *Using Site-Weights.* A relatively straightforward generalization is weighted rank approximation, where we minimize

$$\sigma_\Gamma(X,\Psi) \stackrel{\Delta}{=} \sum_{i=1}^{n}\sum_{k=1}^{n} \gamma_{ik}\langle f_i - \sum_{s=1}^{p} x_{is}\psi_s, f_k - \sum_{s=1}^{p} x_{ks}\psi_s\rangle,$$

where $\Gamma = \{\gamma_{ik}\}$ is a positive semi-definite matrix of site-weights. We call this a straightforward generalization because we can use the spectral decomposition $\Gamma = Q\Xi^2 Q'$ to reduce the weighted problem to the unweighted problem. Just define

$$\ddot{f}_i = \sum_{k=1}^{n} \xi_k q_{ik} f_k,$$

$$\ddot{x}_{is} = \sum_{k=1}^{n} \xi_k q_{ik} x_{ks},$$

then

$$\sigma_\Gamma(X,\Psi) = \sum_{i=1}^{n} \|\ddot{f}_i - \sum_{s=1}^{p} \ddot{x}_{is}\psi_s\|^2,$$

which is an unweighted problem.

4. Some Theoretical Examples

4.1. **27 Normal Densities.** We use 27 different versions of the normal density

$$\phi_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\}$$

defined on $(-\infty, +\infty)$. There is no special reason to choose densities for our first example, we merely use them because they are familiar and lead to fairly easy calculations.

The first nine curves have standard deviation 0.5, the next nine curves have standard deviation 1, and the last nine curves have standard deviation 2. All three groups of nine curves have means $\mu = -4(1)4$.

Consider the inner product

$$\langle f, g\rangle_L = \int_{-\infty}^{+\infty} f(x)g(x)dx.$$

Result 4.1.

$$\langle \phi_{\mu_1,\sigma_1}, \phi_{\mu_2,\sigma_2}\rangle_L = \frac{1}{\overline{\sigma}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{\mu_1-\mu_2}{\overline{\sigma}}\right)^2\right\},$$

where

$$\overline{\sigma}^2 \stackrel{\Delta}{=} \sigma_1^2 + \sigma_2^2.$$

Proof. Take $f = \phi_{\mu_1,\sigma_1}$ and $h = \phi_{-\mu_2,\sigma_2}$ .The convolution of f and h is

$$(f \star h)(z) = \int_{-\infty}^{+\infty} f(x)h(z-x)dx,$$

and we know that the convolution is the density of the sum of two independent random variables with densities f and h, i.e. it is $\phi_{\mu_1-\mu_2,\overline{\sigma}}$. Taking $z = 0$ gives

$$\int_{-\infty}^{+\infty} f(x)h(-x)dx = \frac{1}{\overline{\sigma}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{\mu_1-\mu_2}{\overline{\sigma}}\right)^2\right\},$$

but of course $h(-x) = g(x)$. □

We now do the functiional PCA, with unit site weights. All the analysis in this section and the next are done on the doubly centered C, i.e. on curves in deviations from the average curve. As Table 1 shows, the first two eigenvalues account for only 53% of the total inertia, which means the two principal curves do not approximate the normal densities very well. We also see that

Insert Table 1 about here

Figure 1(a) plots the two columns of the loadings X, which give the inner products of the observed curves f_i and the two principal curves ψ_1 and ψ_2. Points corresponding with curves of the same standard deviation have been connected by colored lines, which creates three nested "horseshoes" in the plot. Points corresponding with curves having the same mean are also connected, thus connected a warped grid. In all two-dimensional loadings plots attention must be paid to the scale of the two axes. Remember that $X'X = \Lambda^2 I$, which often means the vertical axis is much less "important". One can make a case for using the same scaling or range for the two axes [Gower and Hand, 1996], but we have chosen to use the default scaling of the plot function in R.

Another way of looking at the loadings is in Figure 2, where the loadings of the first component are plotted against the means and the loadings of the second component against the standard deviations.

Insert Figure 2 about here

Insert Figure 1(a) about here

The quality of the approximation of the squared curve-distances by the squared Euclidean distances is illustrated in Figure 1(b). We call such plots *Benzécri plots*, because they have been used mostly in Correspondence Analysis. As we have shown, approximation is

from below, so all points are below the 45-degree line. The approximation is obviously not very good, mostly because some of the larger curve-distances are seriously underestimated .

Insert Figure 1(b) about here

Finally Figure 3 gives the average curve $f_\bullet$ and the two principal curves ψ_1 and ψ_2.

Insert Figure 3 about here

Interpreting principal curves is perhaps best done by using a tool suggested by Ramsay and Silverman [2005]. We dynamically plot the curves $f_\bullet + \alpha\psi_1 + \beta\psi_2$ by letting α and β vary in a circle around the origin. This can be done by some type of Grand Tour, but it is probably better to use a two-dimensional slider, or to capture mouse clicks in the plane to define α and β, and then produce the corresponding curve. For the time being we show 25 perturbation of the average curve in the direction of the first and second principal curve in Figure 4.

Insert Figure 4 about here

4.2. **Gauss-Hermite.** We use the same example to illustrate numerical integration. In Gauss-Hermite $K-$point integration for this example we use the approximation

$$\int_{-\infty}^{+\infty} \phi_{\mu_1}(x)\phi_{\mu_2}(x)dx \approx \sum_{k=1}^{K} w(x_k)\exp(x_k^2)\phi_{\mu_1,\sigma_1}(x_k)\phi_{\mu_2,\sigma_2}(x_k),$$

where the x_k are the quadrature points and the $w(x_k)$ are the corresponding weights. Quadrature points and weights have been extensively tabulated, and they can be easily computed for any K. Although we usually think of the Gauss-Hermite quadrature sums as an approximation to the integral, we can also think of them as alternative inner products.

In Table 2 we give first 11 eigenvalues for $K = 1, \cdots, 20$. The remaining are all zero, or very close to zero. We see they converge quite rapidly to the values for the continuous inner product, in the last row (which were already given in Table 1).

Insert Table 2 about here

4.3. **A Truncated Inner Product.** Now consider the inner product

$$\langle \phi_{\mu_1,\sigma_1}, \phi_{\mu_2,\sigma_2} \rangle_p = \int_0^\infty \phi_{\mu_1,\sigma_1}(x)\phi_{\mu_2,\sigma_2}(x)dx$$

If both means are positive and large this will be approximately equal to the previous inner product. But if one of the means is negative and large then the inner product will be small, and if both of them are negative and large it will be very small. By large we mean, of course, large relative to the standard deviation.

We start with a simple computational result for the product of two normal densities. This could also have been used to give an alternative proof of Result 4.1.

Result 4.2.

$$\phi_{\mu_1,\sigma_1}(x)\phi_{\mu_2,\sigma_2}(x) = \phi_{\overline{\mu},\underline{\sigma}}(x)\phi_{\mu_1-\mu_2,\overline{\sigma}}(0),$$

where

$$\begin{aligned}\overline{\mu} &= \frac{\sigma_1^2\mu_2 + \sigma_2^2\mu_1}{\sigma_1^2 + \sigma_2^2},\\ \overline{\sigma} &= \sqrt{\sigma_1^2 + \sigma_2^2},\\ \underline{\sigma} &= \frac{\sigma_1\sigma_2}{\overline{\sigma}}\end{aligned}$$

It follows that the inner product is

$$\langle \phi_{\mu_1,\sigma_1}, \phi_{\mu_2,\sigma_2} \rangle_p = \phi_{\mu_1-\mu_2,\overline{\sigma}}(0)(1 - \Phi(-\frac{\overline{\mu}}{\underline{\sigma}})),$$

where Φ is the standard normal distribution function.

Eigenvalues for this inner product of our 27 normal densities are in Table 3.

Insert Table 3 about here

The plot of the loadings is in Figure 7(a). We see that, indeed, points corresponding to negative means tend to be close together, no matter what the standard deviation is. Thus the three horseshoes are "warped" to start at a common origin.

Insert Figure 7(a) about here

The plot of squared curve-distances and squared Euclidean distances is in Figure 7(b).

Insert Figure 7(b) about here

Finally, the principal curves corresponding with the first two eigenvalues are in Figure 8.

Insert Figure 8 about here

4.4. **Nine Rationals.** Consider the functions, defined on $[0, +\infty)$,

$$f_a(x) = \frac{a}{x + a}$$

where $a > 0$ is a parameter. Curves for $a = 1(1)9$ are drawn in Figure 9.

Insert Figure 9 about here

We use the usual $\mathcal{L}_2$ inner product. In this case it is

$$\langle f_a, f_b \rangle_L = \int_0^{+\infty} f_a(x) f_b(x) dx = \\ = ab \int_0^{+\infty} \frac{1}{x + a} \frac{1}{x + b} dx = \frac{ab}{a - b} \log \frac{a}{b}.$$

Loadings and Benzécri plot are in Figure 10. The eigenvalues of C are in Table 4. Since 99% of the inertia is captured by the first dimension, we obviously have a very good representation using only that single dimension.

Insert Table 4 about here

Insert Figure 10 about here

Finally Figure 11 gives the average curve $f_\bullet$ and the two principal curves ψ_1 and ψ_2.

Insert Figure 11 about here

5. Approximation *sans* Inner Product

In Section 2 we computed C from the inner products, doubly-centered using (3a), and then computed Δ from C. We subsequently approximated Δ from below, using the spectral decomposition of $\tilde{C}$.

If we start with Δ, computed in some way or another, then we can use (3b) to compute a double-centered C, and use its spectral decomposition to compute X and the principal curves. The major problem with this is that matrices C constructed in the way do not need to be positive semi-definite. In fact positive semi-definiteness of C is guaranteed if and only if Δ is a squared distance based on an inner product. Nevertheless, as in classical metric MDS [Torgerson, 1958], we can do the computations as long as the negative eigenvalues of C are small. Observe that at long as our approximation uses only the positive eigenvalues of C we can still compute the principal curves and the theory of Section 3 remains correct.

5.1. **KL Distance between Nine Normals.** The (asymmetric) Kullback-Leibler divergence between f and g is

$$\delta^2_{KL}(f|g) = \int_{-\infty}^{+\infty} f(x) \log \frac{f(x)}{g(x)} dx.$$

To get rid of the asymmetry we also define

$$\begin{aligned}\delta^2_{KL}(f,g) &= \delta^2_{KL}(f|g) + \delta^2_{KL}(g|f) = \\ &= \int_{-\infty}^{+\infty} (f(x) - g(x)) \log \frac{f(x)}{g(x)} dx.\end{aligned}$$

Although this divergence measure is both non-negative and symmetric, it generally does not satisfy the triangle inequality and does not derive from an inner product.

If $f = \phi_{\mu_1,\sigma_1}$ and $g = \phi_{\mu 2,\sigma_2}$, then

$$\delta^2_{KL}(f|g) = \frac{1}{2}\left\{\log \frac{\sigma_2^2}{\sigma_1^2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{\sigma_2^2} - 1\right\},$$

and thus

$$\delta_{KL}^2(f,g)=\frac{1}{2}\left\{\frac{\sigma_1^2+(\mu_1-\mu_2)^2}{\sigma_2^2}+\frac{\sigma_2^2+(\mu_1-\mu_2)^2}{\sigma_1^2}-2\right\}.$$

Observe that in the special case $\sigma_1=\sigma_2$ we have

$$\delta_{KL}^2(f,g)=\frac{(\mu_1-\mu_2)^2}{\sigma^2},$$

which is of course derived from the inner product $\langle f,g\rangle=\mu_1\mu_2/\sigma^2$. If we have shifted normals with the same variance then our KL analysis will just yield a single dimension, equal to the centered μ_i/σ.

Four our 27 shifted and scaled normals the doubly-centered matrix of squared curve-distances has two large positive and two smaller negative eigenvalues. The remaining eigenvalues are practically zero.

Insert Table 5 about here

The plot is in Figure 5(a), and it shows nice separation of the three horseshoes.

Insert Figure 5(a) about here

The plot of squared curve-distances and squared Euclidean distances is in Figure 5(b). The most remarkable characteristic is that all distances are *over-estimated*, and we seem to approximate curve-distances from above. This has a simple explanation. We use the two positive eigenvalues, so only negative eigenvalues are left. Thus additional dimensions will actually subtract from the two-dimensional distances until we arrive at the curve-distance.

Insert Figure 5(b) about here

Finally, the principal curves corresponding with the positive eigenvalues are in Figure 6.

Insert Figure 6 about here

5.2. **Sup-norm for Rationals.** For the nine simple rational functions we can define the distance

$$\delta(f_a, f_b) = \max_{x \geq 0} |f_a(x) - f_b(x)|.$$

After some computation we find the maximum is attained at $x = \sqrt{ab}$ and it is equal to

$$\delta(f_a, f_b) = \left| \frac{\sqrt{a} - \sqrt{b}}{\sqrt{a} + \sqrt{b}} \right|.$$

In this case, although the curve-distances are not derived from an inner product, it turns out that the matrix C derived from them is positive semi-definite anyway. This follows quite simply from

$$\delta^2(f_a, f_b) = \left(\frac{\sqrt{a}}{\sqrt{a} + \sqrt{b}} - \frac{\sqrt{b}}{\sqrt{a} + \sqrt{b}} \right)^2,$$

which gives

$$-\frac{1}{2}(\delta^2(f_a, f_b) - \delta^2(f_a, f_0) - \delta^2(f_a, f_0)) = \frac{\sqrt{ab}}{(\sqrt{a} + \sqrt{b})^2},$$

which is the Hadamard product [Styan, 1973] of two positive-semidefinite matrices, and is consequently positive semi-definite.

Loadings and Benzécri plot are in Figure 12. The eigenvalues of C are in Table 6. The solution is very similar to the $\mathcal{L}_2$ solution for the rationals, with again a very good one-dimensional fit.

Insert Table 6 about here

Insert Figure 12 about here

Finally Figure 13 gives the average curve $f_\bullet$ and the two principal curves ψ_1 and ψ_2.

Insert Figure 13 about here

5.3. **Two-sided Approximation.** If we start with curve-distances which are not necessarily derived from an inner product, and thus with a matrix C that may not be positive semi-definite, then there is no clear reason any more for approximating squared distances from below. As De Leeuw and Meulman [1985, 1986] argue we might as well minimize

$$\sigma(X) = \sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij}(\delta_{ij} - d_{ij}(X))^2 \tag{10}$$

over X, and which optimizes approximation from both sides. Some curve-distances will be under-estimated, some will be over-estimated. At the minimum we have

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}(X)(\delta_{ij} - d_{ij}(X)) = 0,$$

so there are both positive and negative residuals.

The loss function (10), commonly called *stress*, can be minimized by the usual MDS methods [Borg and Groenen, 2005]. This gives us X. We then bring X in the form $X = K\Lambda L'$ to ensure that $C = XX'$ and $X'VX = I$. Principal curves can still be calculated using the calculations in Subsection 3.1, although they will not have the direct least squares properties of Subsection 3.3.

As an example, suppose we use the Kullback-Leibler distance between the twenty seven normals differing in location and scale. Minimizing stress takes does not change the configuration much, but the Benzécri plot in Figure 14(b) is quite different. It shows the usual least squares characteristics: small distances are over-estimated and large distances are under-estimated. The principal curves are in Figure 15.

Insert Figure 14 about here

Insert Figure 15 about here

Although we will not actually makes these calculations, it would be simple to go to non-metric MDS from here. This will take us even further from the direct least squares approximation of the curves, but which will improve the approximation of the curve distances (or the transformed curve distances). The interesting part is that we can continue to plot the "principal curves" corresponding with the MDS solutions.

6. Constraints

In this section we study, again, the problem of minimizing

$$\sigma(X,\Psi)\overset{\Delta}{=}\sum_{i=1}^n\|f_i-\sum_{s=1}^p x_{is}\psi_s\|^2, \tag{11}$$

but now with various constraints on the loadings in X and/or the curves in Ψ.

6.1. **Constraints on the Loadings.**

6.1.1. *To Center or Not To Center.* Suppose we require the first column of X to be equal to $+1$. Thus

$$\sigma(X,\psi_1,\psi_2,\cdots.\psi_p)=\sum_{i=1}^n\|f_i-\psi_1-\sum_{s=2}^p x_{is}\psi_s\|^2.$$

The conditional minimum [1] over ψ_1 is attained at

$$\psi_1=f_\bullet-\sum_{s=1}^p x_{\bullet s}\psi_s,$$

and is equal to

$$\sigma(X,\star,\psi_2,\cdots.\psi_p)=\sum_{i=1}^n\|\tilde f_i-\sum_{s=2}^p\tilde x_{is}\psi_s\|^2,$$

where both f_i and x_{is} have been replaced by their deviations from the mean over sites. Thus a PCA of centered curves is equivalent to an uncentered PCA with loadings on the first component restricted to be $+1$. And this, in its turn, is equivalent to an eigenvalue analysis of the doubly-centered $\tilde C=JCJ$ of Equation 3a.

In some cases we may actually prefer the stronger centering constraints in which we require both $x_{i1}=1$ for all i and $\langle f_\bullet,\psi_s\rangle=0$ for $s=2,\cdots,p$. Thus the principal curves must be orthogonal to the average curve, as well as being orthogonal to each other. Making the usual transformations this means we have to maximize

[1] The *conditional minimum* of $f(x,y)$ over y is $g(x)=\min_y f(x,y)$.

tr $X'\tilde{C}Y$ over X and Y such that $Y'\tilde{C}Y = I$ as well as $Y'h = 0$. Here $\tilde{C}$ is the Gram matrix of the $\{\tilde{f}_i\}_{i=1}^n$ and h has elements $h_i = \langle f_\bullet, \tilde{f}_i\rangle$.

6.1.2. *Simple Linear Constraints.* The centering result, using conditional minima, can be generalized to more complicated linear constraints on the loadings. Consider the case in which $X = \begin{bmatrix} X_1 & X_2 & X_3 \end{bmatrix}$ where X_1 is known, $X_2 = GA$ with the $n \times q$ matrix G known, satisfying $G'X_1 = 0$, and X_3 is free.

with the $n \times q$ matrix G completely known. The basic loss function, using the least squares norm on the space of $n \times r$ matrices, becomes

$$\|Z - XY'\|^2 = \|Z - X_1Y_1 - GAY_2' - X_3Y_3\|^2,$$

which means that we must maximize

$$\textbf{tr}\ Y'Z'G(G'G)^+G'ZY$$

over $Y'Y = I$, a simple eigenvalue problem.

6.1.3. *More General Constraints.* In the case of more general constraints on X, say $X \in \mathcal{X}$, we can write the loss function in the form

$$\|Z - XY'\|^2 = \|Z - \hat{X}Y'\|^2 + \|X - \hat{X}\|^2,$$

where $\hat{X} = ZY$. This suggest a simple alternating least squares (ALS) algorithm. We alternate minimization over $X \in \mathcal{X}$ for given Y and minimization of Y with $Y'Y = I$ for for given X.

The first subproblem has the solution $X = P_{\mathcal{X}}(\hat{X})$, the *projection* of $\hat{X}$ on $\mathcal{X}$, i.e. the minimizer over $X \in \mathcal{X}$ of $\|X - \hat{X}\|^2$.

The second subproblem is a so-called *Orthogonal Procrustus Problem*. If $Z'X$ has singular value decomposition $Z'X = K\Lambda L'$, then the optimal Y for given X is $Y = KL'$.

6.2. **Constraints on the Curves.** Now consider the problem, again, of minimizing

$$\sigma(X,\Psi)\stackrel{\Delta}{=}\sum_{i=1}^n\|f_i-\sum_{s=1}^p x_{is}\psi_s\|^2,$$

but now some of the curves, say the first $q<p$ ones, are supposed to be known. Without loss of generality we can assume they are orthogonal to each other, and we require the remaining unknown $p-q$ curves to be orthogonal to the first q known ones. Rewrite the loss function as

$$\sigma(X,\Psi)=\sum_{i=1}^n\|f_i-\sum_{s=1}^q x_{is}\psi_s-\sum_{s=q+1}^p x_{is}\psi_s\|^2.$$

Now clearly the solution for x_{is} with $s\leq q$ is $\hat{x}_{is}=\langle f_i,\psi_s\rangle$. Define

$$\tilde{f}_i=f_i-\sum_{s=1}^q\langle f_i,\psi_s\rangle\psi_s,$$

which means we still have to minimize

$$\sigma(X,\Psi)=\sum_{i=1}^n\|\tilde{f}_i-\sum_{s=q+1}^p x_{is}\psi_s\|^2,$$

which is of the familiar form. This can be used, for example, to remove the mean level of each curve, to remove a linear or polynomial trend, or to remove periodic components representing seasonality. The PCA is then done on the residuals, and these residuals are orthogonal to each of the curves we have removed. Of course if one removes too much, there will not be enough left for the PCA to be interesting.

A different type of constraint on the curves would be that $\sum_{s=1}^q y_{ts}\psi_s=\eta_t$, where the η_t are known curves.

$$\psi_s=\sum y_{st}\eta_t$$

7. Transformations and Missing Values

7.1. **Missing Values.** There are various ways to deal with missing values, i.e. cases where the curves are not defined on the same interval. We could handle missing data in the first phase of FPCA, where we are constructing the curves to be entered into the PCA. But even then we have choices. We can use imputations to make the discrete data complete, or we estimate the curve by using the available data only.

A different alternative is to handle the missing data in the PCA, i.e. in the second phase. This can be done with optimal scaling (OS), using the alternating least squares (ALS) ideas of nonlinear PCA [De Leeuw, 2006a]. We minimize

$$\sigma(X,\Psi,F)=\sum_{i=1}^n\|f_i-\sum_{s=1}^p x_{is}\psi_s\|^2 \tag{12}$$

over Ψ and X, as usual, but in addition over $F=\{f_1,\cdots,f_n\}$. But the f_i are constrained to be equal to the observed curves in all points where the observed curves are defined.

The ALS algorithm alternates two steps in each iteration. We start with curves that have been made complete in some arbitrary way. Then, in the first step, we do the PCA, which leads to approximations $\hat{f}_i$. We then make new curves as the second step. The new curves are equal to the observed value of f_i if that is available and to the value of $\hat{f}_i$ otherwise. These new curves are then fed into PCA again, and so on, until convergence. It is not necessary to do a complete PCA in the first step, just improving the PCA approximation to the current set of curves is sufficient.

In the special case of the $\mathcal{L}_2$ norm, and observations missing outside an interval, the problem can be formulated as minimization

$$\sigma(X,\Psi)=\sum_{i=1}^n\int_{\alpha_i}^{\beta_i}(f_i(t)-\sum_{s=1}^p x_{is}\psi_s(t))^2,$$

or, equivalently, using a weight function w_i that is equal to one on (α_i, β_i) and equal to zero elsewhere,

$$\sigma(X, \Psi) = \sum_{i=1}^{n} \int_{-\infty}^{+\infty} w_i(t)(f_i(t) - \sum_{s=1}^{p} x_{is}\psi_s(t))^2.$$

7.2. **Projection.** A more general problem is minimization of (12) over X, Ψ and over all $f_i \in \mathcal{F}_i$, where $\mathcal{F}_i$ is a convex set. We suppose our inner-product space is complete, and our convex set is closed. Appendix D shows the that projection on $\mathcal{F}$ exists and is uniquely defined, so we can actually carry out the same ALS algorithm as the one we used for missing data. In fact the missing data case is the special case in which $\mathcal{F}_i$ is the set of all curves that agree with the observed curve in all points where it has been observed (and is arbitrary everywhere else).

If $\mathcal{F}_i$ is a closed convex cone we have to solve a different problem. Minimizing (12) is too easy, because we can always set $f_i = 0$ and find a perfect, but trivial, solution. In that case, in the non-metric MDS tradition, we require $f_i \in \mathcal{F} \cap \mathcal{S}$, where $\mathcal{S}$ is the set of all curves with unit norm, i.e. $\|f_i\|^2 = 1$. Or, alternatively, we require that $\|\hat{f}_i\|^2 = 1$, where $\hat{f}_i = \sum_{s=1}^{p} x_{is}\psi_s$. Thus $\|\hat{f}_i\|^2 = \sum_{s=1}^{p} x_{is}^2$, which defines a simple restriction on the loadings.

It does not matter if we normalize f or $\hat{f}$ in the case of convex cones. This follows from a simple general result. Suppose f and g are arbitrary curves. Define $\tilde{f} \triangleq \frac{\|g\|}{\|f\|} f$ and $\tilde{g} \triangleq \frac{\|f\|}{\|g\|} g$. Then $\|f - g\|^2 = \|\tilde{f} - \tilde{g}\|^2$.

7.3. **Transforming the Domain.** $\overline{f} = g \circ f$Example

$$\langle f_1, f_2 \rangle = \int_{-\infty}^{+\infty} g_1(f_1 x))g_2(f_2(x))dx$$

where g_1 and g_2 are restricted to be monotone increasing.

$\overline{f} = f \circ g$ Example:

$$\langle f_1, f_2 \rangle = \int_{-\infty}^{+\infty} f_(\alpha_1 + \beta_1 x)f_2(\alpha_2 + \beta_2 x)dx,$$

7.4. **Use of Majorization.** We know that the minimum value of the loss function (12) is equal to the sum of the $n-p$ smallest eigenvalues of C, with $c_{ik} = \langle f_i, f_k \rangle$. Now suppose C is not fixed but depends on transformations of the curves (or, to put it differently, suppose the curves depend on a number of parameters). Ther are many examples: curves can depend a a smoothing parameter, or on knot placement, or on monotone tranformations. We can apply the theory in De Leeuw [1988, 1990].

Theorem 7.1. *$\ell_{n-p}(C)$, the sum of the $n-p$ smallest eigenvalues, is a concave function of C.*

Proof. $\ell_{n-p}(C) = \min_K \mathbf{tr}\, K'CK$, where K varies over the $n \times (n-p)$ orthonormal matrices. Thus ℓ_{n-p} is the pointwise minimum of a family of linear functions, and it is thus concave. □

If $\tilde{K}$ is any set of eigenvectors corresponding with the $n-p$ smallest eigenvalues of $\tilde{C}$ then for any C we have $\ell_{n-p}(C) \leq \mathbf{tr}\, \tilde{K}'C\tilde{K}$. It follows that if we can find a $\check{C}$ such that

$$\mathbf{tr}\, \tilde{K}'\check{C}\tilde{K} < \mathbf{tr}\, \tilde{K}'\tilde{C}\tilde{K} = \ell_{n-p}(\tilde{C})$$

then $\ell_{n-p}(\check{C}) < \ell_{n-p}(\tilde{C})$.

7.5. **Singular Spectrum Analysis of Time Series.**

7.6. **Application to Count Data.** FPCA techniques seem not directly applicable to count data, because counts typically take place in temporal intervals or spatial regions. The counting itself already takes the continuity out of the process, and leads to essentially discrete data. Thus counts are more naturally modeled as integrals of intensity curves over intervals or regions. But we can of course use a PCA-type specification of this intensity curve.

If we assume a non-homogeneous Poisson process in time, for example, then the counts n_{ij} at site i in interval j are independent

Poisson's with expected values

$$\lambda_{ij} = \alpha_i \int_{\xi_j}^{\xi_{j+1}} \beta(t) \exp\{\sum_{s=1}^{p} x_{is}\psi_s(t)\}dt.$$

A simple-minded approximation using the Mean Value Theorem gives

$$\lambda_{ij} \approx \alpha_i\beta_j \exp\{\sum_{s=1}^{p} x_{is}y_{js}\},$$

which is the Goodman-Haberman RC model. This can be fitted by minimizing the Poission Deviance

$$\mathcal{D}(\alpha, \beta, X, Y) \stackrel{\Delta}{=} \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_{ij} - n_{ij}\log\lambda_{ij},$$

for instance by the iterated singular value decomposition methods of De Leeuw [2006b].

A further first-order power series approximation, valid for small interaction terms, gives

$$\lambda_{ij} \approx \alpha_i\beta_j\{1 + \sum_{s=1}^{p} x_{is}y_{js}\},$$

which is the model used in Correspondence Analysis. The usual loss function for Correspondence analysis is

$$S \stackrel{\Delta}{=} (\alpha, \beta, X, Y) \stackrel{\Delta}{=} \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{(n_{ij} - \lambda_{ij})^2}{n_{I\bullet}n_{\bullet j}},$$

which we can minimize by computing the singular value decomposition of the matrix of Pearson residuals.

8. The Lebec Data

8.1. **Data.** The California Air Resources Board (CARB) collected data on hourly ozone and PM-2.5 levels (in parts per million) over approximately one year (February 2006-February 2007) in the township of Lebec, Kern County, California. The data (in the form of R data frames) can be obtained from

http://idisk.mac.com/jdeleeuw-Public/curves/lebec.

The directory

http://idisk.mac.com/jdeleeuw-Public/curves/.

actually has LaTeX code and all the figures and tables for this paper. To download a file using your webserver or ftp client (supporting WebDAV), just add the file name to the URL.

For ozone we have measurements on 372 consecutive days, and thus a data matrix of order 372×24. The matrix has 8415 nonmising element, out of a possible 8928, i.e. about 5.7% of the measurements are missing. There are nine days for which we have no measurements at all. The total sum of squares of the non-missing elements is 13.499. Data for ozone are plotted in Figure 16.

Insert Figure 16 about here

It should be emphasized that the "codage" we have chosen is, to some extent, arbitrary. Instead of a 372×24 matrix we could have interpreted the data as a 1×8928 matrix, with a single long time series, or we could have included the PM-2.5 to make a 2×8928 multiple time series. These alternative codings would have suggested alternative techniques, which are not necessarily of the FPCA type. Our data format suggests we think of days as replications, and we ignore the dependence between days. This is most clearly illustrated by the fact that we would get identical results of we were to permute the 372 rows of the matrix.

8.2. **Discrete PCA.** The first analysis we try is a simple PCA on the data matrix. This means using the inner product

$$\langle f_i, f_k \rangle = \sum_{t=0}^{T} f_{it} f_{kt}$$

with $T = 23$. If we want to put this into the $\mathcal{L}_2$ framework we can think of the curves as step functions, constant on each one-hour interval. For our software we need an SVD routine that can handle missing data. The R-code is given in Appendix H.2. We use three components, leading to a residual sum of squares of 0.4139.

Alternatively, we can also remove the mean curve first and then extract two components. As we saw in Section 6.1 this amounts to a restricted form of PCA with three components, with all loadings on the first component equal to one. In Table 7 we show the 24 eigenvalues of C, where C is the Gram matrix of the imputed data. The first column has the SVD of the imputed data, the second column has the SVD of the matrix of deviations from the mean curve. As we see the average explains about 78.5% of the total inertia, while the dominant principal curve in the uncentered analysis explains about 92.5%. In the uncentered analysis the additional principal curves have small contributions, which decrease slowly. In the centered analysis the second principal curve contributes 14.5%, which is still quite substantial. On the basis of this we could decide that we prefer the centered analysis, with the average and one principal curve, to the uncentered analysis with three principal curves.

If we look more carefully at Table 7, and we compare Figures **??** and **??**, then it looks as if the first principal curve of the uncentered analysis is split into the average curve and the first principal curve in the centered analysis. The second and the third principal curves in the centered analysis are very similar to the second and the third principal curves in the uncentered analysis. So it looks as if the centered analysis gives more detail, and a somewhat easier interpretation (since one of the components is the average).

Insert Figure ?? about here

Insert Figure ?? about here

As a final alternative in this section we can use Correspondence Analysis. See Section 7.6 for a possible justification.

8.3. **Removing Main Effects.**

$$z_{ij} \approx \mu + \alpha_i + f_j + \sum_{s=1}^{p} x_{is}y_{js},$$

$$z_{ij} \approx \mu + \alpha_{\text{month}(i)} + \gamma_{\text{weekday}(i)} + f_j + \sum_{s=1}^{p} x_{is}y_{js},$$

$$z_{ij} \approx \mu + f_j + \sum_{s=1}^{p} (x_{\text{month}(i)s} + x_{\text{weekday}(i)s})y_{js},$$

8.4. **Using a Smoother.** One simple way to smooth our data is to rely on the `lowess()` function in `R`. The theory for `lowess()` is given in Cleveland [1979, 1981]. Code for smoothing a matrix with curves (and missing data) is in Appendix H.3. The results of the smoothing are plotted in Figure 22. The total sum of squares of the non-missing elements is 12.177

Insert Figure 22 about here

Again, the first principal curve of the uncentered analysis is split into the average curve and the first principal curve of the centered analysis. The second and the third principal curves in the centered analysis are again very similar to the second and the third principal curves in the uncentered analysis.

8.5. **Using a Fourier Basis.** The basic loss function we will use is, not surprisingly,

$$\sigma() = \sum_{i=1}^{n} \int_0^T (f_i(t) - \sum_{s=1}^{p} x_{is}\psi_s(t))^2 dt$$

where time-points zero and T both correspond to midnight.

We smooth the curves by using the Fourier coefficients. This will make them of the form

$$f_i(t) = \mu_i + \sum_{u=1}^{v} \left\{ \alpha_{iu} \sin(2\pi \frac{ut}{T}) + \beta_{iu} \cos(2\pi \frac{ut}{T}) \right\}.$$

As a consequence

$$c_{ik} = \mu_i \mu_k + \sum_{u=1}^{v} \{\alpha_{iu}\alpha_{ku} + \beta_{iu}\beta_{ku}\},$$

and thus $\textbf{rank}(C) \leq 2v + 1$.

In first Lebec example we will fit the Fourier series for $v = 2$ by simple least squares, iteratively imputing the missing data on the way. We start the iterations by replacing all missing data by zeroes, with sum of squares of the residuals equal to 13.50. The ALS iterations for imputation then decrease the residual sum of squares to 0.3416.

In Figure 26 we see the observed and fitted data. Note that the fitted curves are continuous, they are not just drawn in the 24 data points. They are, of course, also smoothed. They do not really get rid of the major outliers, which correspond with curves have a relatively large number of missing data. This is basically because each curve is fit separately.

Insert Figure 25 about here

Figure 26 gives the scatterplot of the non-missing observed and fitted values. Clearly the fit is good, and there seem to be no systematic deviations from the line of perfect fit.

Insert Figure 26 about here

We now do the PCA on the fitted curves. As we have seen, this amounts to doing an SVD on the least squares Fourier coefficients.

The eigenvalues for this analysis are in Table 8. The dominant principal curve fits about 95% of the inertia (of the smoothed curves, obviously, not of the raw data).

Insert Table 8 about here

The first three principal curves are in Figure 27 and the loadings on the first principal curve are plotted in Figure 28.

Insert Figure 27 about here

Insert Figure 28 about here

The "sites" in this study are really "days", and days are characterized by being a particular weekday in a particular month. It may be a good idea to take this information into account. We can account for months and days in different ways. It can be done in the first phase, in which we use least squares to smooth the curves by

$$f_{it} = \mu_i + \gamma_{d(i)} + \delta_{m(i)} + \sum_{u=1}^{v} \left\{ \alpha_{iu} \sin(2\pi \frac{ut}{T}) + \beta_{iu} \cos(2\pi \frac{ut}{T}) \right\},$$

where $m(i)$ is the month of day i and $d(i)$ is the weekday. Or, alternatively, we can restrict the component loadings in the PCA, for example by

$$x_{is} = \mu_s + \beta_{d(i)s} + \gamma_{m(i)s}.$$

References

I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, Second edition, 2005.

W.S. Cleveland. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

W.S. Cleveland. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *The American Statistician*, 35:54, 1981.

J. De Leeuw. Multivariate Analysis with Optimal Scaling. In S. Das Gupta and J. Sethuraman, editors, *Progress in Multivariate Analysis*, Calcutta, India, 1990. Indian Statistical Institute.

J. De Leeuw. Nonlinear Principal Component Analysis and Related Techniques. In M. Greenacre and J. Blasius, editors, *Multiple Correspondence Analysis and Related Methods*. Chapman and Hall, 2006a.

J. De Leeuw. Principal Component Analysis of Binary Data by Iterated Singular Value Decomposition. *Computational Statistics and Data Analysis*, 50(1):21–39, 2006b.

J. De Leeuw. Multivariate Analysis with Linearizable Regressions. *Psychometrika*, 53:437–454, 1988.

J. De Leeuw. Fixed-rank Approximation with Singular Weight Matrices. *Computational Statistics Quarterly*, 1:3–12, 1984.

J. De Leeuw and J.J. Meulman. Principal Component Analysis and Restricted Multidimensional Scaling. In W. Gaul and M. Schader, editors, *Classification as a Tool of Research*, pages 83–96, Amsterdam, London, New York, Tokyo, 1986. North-Holland.

J. De Leeuw and J.J. Meulman. Alternative Approximations in Principal Component Analysis. Research Report RR-85-13, Department of Data Theory FSW/RUL, 1985.

F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis*. Springer Series in Statistics. Springer Verlag, Berlin, Heidelberg, New York, 2006.

J.C. Gower and D.J. Hand. *Biplots*. Number 54 in Monographs on Statistics and Applied Probability. Chapman and Hall, 1996.

P. Halmos. *Introduction to Hilbert Space*. Chelsea Publishing Company, New York, New York, second edition, 1957.

F.A. Ocana, A.M Aguilera, and M.J. Valderrame. Functional Principal Components Analysis by Choice of Norm. *Journal of Multivariate Analysis*, 71:262–276, 1999.

J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer Verlag, Berlin, Heidelberg, New York, Second edition, 2005.

J.O. Ramsay and B.W. Silverman. *Applied Functional Data Analysis: Methods and Case Studies*. Springer Series in Statistics. Springer Verlag, Berlin, Heidelberg, New York, 2002.

W. Rudin. *Real and Complex Analysis*. McGrawHill, international student edition, 1970.

L. Schwartz. *Analyse Hilbertienne*. Hermann, Paris, France, 1979.

G.P. Styan. Hadamard Products and Multivariate Statistical Analysis. *Linear Algebra and its Applications*, 6:217–240, 1973.

W.S Torgerson. *Theory and Methods of Scaling*. Wiley, New York, 1958.

H. Zha. The Product–Product Singular Value Decomposition of Matrix Triplets. *BIT*, 31:711–726, 1991.

APPENDIX A. GRAM-SCHMIDT FOR CURVES

Suppose we have n curves f_i. Apply the following Gram-Schmidt (or GS) algorithm:

Start: $i = 1, r = 0$.
Test: if $i > n$ then go to **Stop**.
Ortho: $\tilde{f}_i = f_i - \sum_{k=1}^{r} \langle f_i, \breve{f}_k \rangle \breve{f}_k$.
Skip: If $\|\tilde{f}_i\| = 0$ then $i \leftarrow i + 1$ and go to **Test**.
Norm: $\breve{f}_r = \tilde{f}_r / \|\tilde{f}_r\|$.
Count: $r \leftarrow r + 1, i \leftarrow i + 1$ and go to **Test**.
Stop: Return $(\breve{f}_1, \cdots, \breve{f}_r)$.

We define the *rank* of the n curves f_i as the number of $\breve{f}_i$ generated by the GS algorithm.

Theorem A.1 (Gram-Schmidt)**.** *Suppose f_i are n curves and $\breve{f}_1, \cdots, \breve{f}_r$ are the curves produced by the GS algorithm. Then for all $i \neq k$ we have $\langle \breve{f}_i, \breve{f}_k \rangle = 0$ and for all i we have $\|\breve{f}_i\| = 1$.*

Proof. Use induction on i. □

Suppose $F \triangleq \mathcal{L}(f_1, \cdots, f_n)$ is the subspace spanned by the f_i, or, equivalently, the subspace spanned by the $\breve{f}_s$. We define the *projection* of any curve g on F as

$$\mathcal{P}_F(g) \triangleq \sum_{s=1}^{r} \langle g, \breve{f}_s \rangle \breve{f}_s$$

and we define the *anti-projection* or *residual* by $\mathcal{R}_F(g) \triangleq g - \mathcal{P}_F(g)$.

Theorem A.2 (Least Squares)**.** *Let $F = \mathcal{L}(f_1, \cdots, f_n)$. For all $f \in F$ we have $\mathcal{P}_F(f) = f$, and thus $\mathcal{R}_F(f) = 0$. More generally, for any curve g decomposed as $g = \mathcal{P}_F(g) + \mathcal{R}_F(g)$ we have $\langle f, \mathcal{R}_F(g) \rangle = 0$ for all $f \in F$.*

Proof. □

The theorems in this Appendix can be used to solve various regression problems for curves. We only discuss two simple ones, but there are many possible univariate and multivariate generalizations.

First, consider the situation in which y_i are n numbers and we look for a curve g such that

$$\sigma(g) = \sum_{i=1}^{n} (\langle f_i, g \rangle - y_i)^2$$

is minimized. The solution is $g = \sum_{s=1}^{r} \beta_s \breve{f}_s$, where $\beta_s = (H'H)^{-1}H'y$ and H has elements $h_{is} \stackrel{\Delta}{=} \langle f_i, \breve{f}_s \rangle$.

Second, look for n numbers β_i such that

$$\sigma(\beta) = \| \sum_{i=1}^{n} \beta_i f_i - g \|^2$$

is minimized. At the solution, by definition, we must have $\sum_{i=1}^{n} \beta_i f_i = \mathcal{P}_F(g)$. This works out to $\beta = H(H'H)^{-1}y$, where $h_{is} = \langle f_i, \breve{f}_s \rangle$ as before, and $y_s \stackrel{\Delta}{=} \langle g, \breve{f}_s \rangle$.

Appendix B. Factoring Positive Semi-definite Matrices

Theorem B.1. *Suppose A is a positive semi-definite $n \times n$ matrix of rank r with eigen-decomposition*

$$A = \begin{bmatrix} K & | & K_\perp \\ {\scriptstyle n\times r} & & {\scriptstyle n\times(n-r)} \end{bmatrix} \begin{bmatrix} \Lambda^2 & \emptyset \\ {\scriptstyle r\times r} & {\scriptstyle r\times(n-r)} \\ \emptyset & \emptyset \\ {\scriptstyle (n-r)\times r} & {\scriptstyle (n-r)\times(n-r)} \end{bmatrix} \begin{bmatrix} K' \\ {\scriptstyle r\times n} \\ K'_\perp \\ {\scriptstyle (n-r)\times n} \end{bmatrix}.$$

Then the $n \times p$ matrix X is a solution of the quadratic matrix equation $A = XX'$ if and only if $X = K\Lambda L'$, where L is any $p \times r$ matrix such that $L'L = I$.

Proof. Write X in the form

$$X = \begin{bmatrix} K & | & K_\perp \\ {\scriptstyle n\times r} & & {\scriptstyle n\times(n-r)} \end{bmatrix} \begin{bmatrix} S \\ {\scriptstyle r\times p} \\ T \\ {\scriptstyle (n-r)\times p} \end{bmatrix}$$

It follows that we must have

$$\begin{bmatrix} \Lambda^2 & \emptyset \\ {\scriptstyle r\times r} & {\scriptstyle r\times(n-r)} \\ \emptyset & \emptyset \\ {\scriptstyle (n-r)\times r} & {\scriptstyle (n-r)\times(n-r)} \end{bmatrix} = \begin{bmatrix} SS' & ST' \\ {\scriptstyle r\times r} & {\scriptstyle r\times(n-r)} \\ TS' & TT' \\ {\scriptstyle (n-r)\times r} & {\scriptstyle (n-r)\times(n-r)} \end{bmatrix}.$$

This means $T = \emptyset$ and $SS' = \Lambda^2$, which implies the statement in the theorem. □

It follows that X can only be a solution if $p \geq r$ and that the rank of any solution X is equal to r. This makes it possible to call X a *minimal solution* of $A = XX'$ if $\mathbf{rank}(X) = \mathbf{rank}(A)$. Theorem B.1 then implies that $X = K\Lambda L$, for some $L'L = LL' = I$.

Corollary B.2. *Suppose A and V are positive semi-definite $n \times n$ matrices. Then there is a minimal solution X of $A = XX'$ such that $\Omega^2 \triangleq X'VX$ is diagonal.*

Proof. We know from Theorem B.1 that $X = K\Lambda L'$ for some L such that $L'L = LL' = I$. Thus $X'VX = L\Lambda K'VK\Lambda L'$ and thus L must be chosen as a complete set of eigenvectors of $H \triangleq \Lambda K'VK\Lambda$. This makes Ω^2 equal to the eigenvalues of H. □

Appendix C. Singular Value Decomposition

The Gram-Schmidt process is one way to compute the QR-decomposition of an $n \times m$ matrix X of rank r. It gives us an $n \times r$ matrix K with $K'K = I$ and an $r \times r$ non-singular upper-triangular matrix S such that $X = KS$. The Singular Value Decomposition (SVD) gives us a different, but closely related, decomposition $X = K\Lambda L'$, where K is a before, but now Λ is diagonal and $L'L = I$. For the proof we need two basic lemmas.

Lemma C.1. *If* $\mathbf{rank}(X) = r$ *then both* $X'X$ *and* XX' *are positive semi-definite of rank* r*.*

Proof. If $Xa = 0$ then $X'Xa = 0$. Conversely, if $X'Xa = 0$ then $a'X'Xa = 0$ and thus $Xa = 0$. Same reasoning for XX'. □

Lemma C.2. *$X'X$ and XX' have the same non-zero eigenvalues.*

Proof. If $X'Xa = \lambda a$ with $\lambda > 0$ then $XX'(Xa) = \lambda(Xa)$. Thus either $Xa = 0$ or λ is an eigenvalue of XX' corresponding to eigenvector Xa. And, similarly, if $XX'b = \mu b$ with $\mu > 0$ then either $X'b = 0$ or μ is an eigenvalue of $X'X$ corresponding with eigenvector $X'b$. □

Now we can state and prove the theorem.

Theorem C.3. *Suppose X is an $n \times m$ matrix of rank r. Then X can be written as*

$$\underset{n\times m}{X} = \underset{n\times r}{K}\,\underset{r\times r}{\Lambda}\,\underset{r\times m}{L'}.$$

where $K'K = L'L = I$ and Λ is a diagonal matrix with positive diagonal elements.

Proof. We use Lemma C.1, and take L equal to the orthonormal eigenvectors corresponding with the r non-zero eigenvalues Λ^2 of $X'X$. Define $K = XL\Lambda^{-1}$. Then $K'K = I$ and $K\Lambda L' = XLL'$. But

$LL' = I - L_\perp L_\perp'$, where $L_\perp$ is an orthonormal basis for the null.space of X. Thus $XLL' = X(I - L_\perp L_\perp') = X$. □

Remark . The decomposition can obviously also be written as

$$X = \begin{bmatrix} K \\ {\scriptstyle n\times r} \end{bmatrix} \mid \begin{bmatrix} K_\perp \\ {\scriptstyle n\times(n-r)} \end{bmatrix} \begin{bmatrix} \Lambda & \emptyset \\ {\scriptstyle r\times r} & {\scriptstyle r\times(m-r)} \\ \emptyset & \emptyset \\ {\scriptstyle (n-r)\times r} & {\scriptstyle (n-r)\times(m-r)} \end{bmatrix} \begin{bmatrix} L' \\ {\scriptstyle r\times m} \\ L_\perp' \\ {\scriptstyle (m-r)\times m} \end{bmatrix},$$

where both $\begin{bmatrix} K & \mid & K_\perp \end{bmatrix}$ and $\begin{bmatrix} L & \mid & L_\perp \end{bmatrix}$ are square and orthonormal.

C.1. **Rank Approximation.** Suppose X is an $n\times m$ matrix of rank r with singular value decomposition $X = K\Lambda L'$. The *p-truncated* SVD is $X_p = K_p\Lambda_p L_p'$, with the left singular vectors K_p, the right singular vectors L_p, and the p singular values Λ_p, all corresponding with the p largest singular values. Of course the p-truncated SVD is not necessarily unique, because we can have $\lambda_p = \lambda_{p+1}$. Also $X_p = X$ if and only if $p \geq r$.

Theorem C.4. *If X is an $n \times m$ matrix of rank r the minimum of*

$$\sigma(Y) = \mathbf{tr}\ (X - Y)'(X - Y)$$

over all $n\times m$ matrices Y of rank p is attained at a p-truncated SVD of X and

$$\min_{\mathbf{rank}(Y)=p} \mathbf{tr}\ (X - Y)'(X - Y) = \sum_{s=p+1}^{r} \lambda_s^2.$$

Generalizations to minimization of $\|A(X - Y)B\|^2$ are discussed, for example, in De Leeuw [1984]; Zha [1991].

C.2. **Orthogonal Procrustus.**

Theorem C.5. *If X is an $n \times m$ matrix of rank r the minimum of*

$$\sigma(Y) = \mathbf{tr}\ (X - Y)'(X - Y)$$

over all $n \times m$ orthonormal matrices Y is attained at the Procrustus transformation of X and

$$\min_{Y'Y=I} \mathbf{tr}\ (X-Y)'(X-Y) = \sum_{s=1}^{p} \lambda_s.$$

Appendix D. Projection Theorems

We do not prove the results in this section. For proofs we refer to the Hilbert space literature, for instance Schwartz [1979]; Halmos [1957] or Rudin [1970, Chapter 4].

Definition D.1. A set $\mathcal{X}$ in a linear space $\mathcal{H}$ is

- *convex* if $x, y \in \mathcal{X}$ and $0 \leq \lambda \leq 1$ implies $\lambda x + (1-\lambda)y \in \mathcal{X}$.
- a *cone* if $x \in \mathcal{X}$ and $\lambda \geq 0$ implies $\lambda x \in \mathcal{X}$.
- *affine* if $x, y \in \mathcal{X}$ implies $\lambda x + (1-\lambda)y \in \mathcal{X}$ for all λ.
- a *convex cone* if $\mathcal{X}$ is convex and is a cone.
- a *subspace* is $\mathcal{X}$ is affine and is a cone.

We now study the problem of finding the closest point to a point y in the set $\mathcal{X} \subseteq \mathcal{H}$, where $\mathcal{H}$ is an inner product space with the associated norm. Such a point, if it exists, is the *projection* of y on $\mathcal{X}$, and we write it as $P_{\mathcal{X}}(y)$. If the projection does not exist, then $P_{\mathcal{X}}(y)$ is the empty set. If the projection exists, but is not unique, then $P_{\mathcal{X}}(y)$ is a non-singleton convex subset of $\mathcal{X}$.

Theorem D.1. *If $\mathcal{X}$ is a closed convex set in a Hilbert space $\mathcal{H}$ then $P_{\mathcal{X}}(y)$ exists and is unique.*

For the next theorem we define the *polar cone* $\mathcal{X}^\circ$ of a cone $\mathcal{X}$ as the set of all y such that $\langle y, x\rangle \leq 0$ for all $x \in \mathcal{X}$. We define the *orthogonal complement* $\mathcal{X}_\perp$ of a subspace $\mathcal{X}$ as the set of all y such that $\langle y, x\rangle = 0$ for all $x \in \mathcal{X}$.

Theorem D.2. *Suppose $z \in P_{\mathcal{X}}(y)$.*

- *If $\mathcal{X}$ is convex then $\langle y - z, x - z\rangle \geq 0$ for all $x \in \mathcal{X}$.*
- *If $\mathcal{X}$ is affine then $\langle y - z, x - z\rangle = 0$ for all $x \in \mathcal{X}$.*
- *If $\mathcal{X}$ is a cone then $\langle y - z, z\rangle = 0$ and $\|y\|^2 = \|z\|^2 + \|y - z\|^2$.*
- *If $\mathcal{X}$ is a convex cone then $y - z = P_{\mathcal{X}^\circ}(y)$.*
- *If $\mathcal{X}$ is a subspace then $y - z = P_{\mathcal{X}_\perp}(y)$.*

Appendix E. Imputation by Alternating Least Squares

Consider the problem of minimizing

$$\sigma(X,Y)=\mathbf{tr}(X-Y)'(X-Y) \tag{13}$$

over $X\in\mathcal{X}$ and $Y\in\mathcal{Y}$, where both $\mathcal{X}$ and $\mathcal{Y}$ are subsets $\mathbb{R}^{n\times m}$.

The problem is a *missing data problem* if $\mathcal{X}$ is defined by a set of cells $\mathcal{K}$ and the constraints $x_{ij}=z_{ij}$ for all $(i,j)\in\mathcal{K}$. Thus some elements of X are fixed and known, the others are unknown, and must be *imputed*.

Alternating least squares algorithms can conveniently be employed if it is relatively easy to minimize (13) over Y when X is fixed at some known value. The algorithm iterates by cycling over two steps. Suppose we start with some $X^{(0)}\in\mathcal{X}$, i.e. some matrix in which we have imputed the missing values in some way or another. Set $s=0$. Then

$$Y^{(s)}=\mathop{\mathbf{argmin}}_{Y\in\mathcal{Y}}\|X^{(s)}-Y\|^2, \tag{14a}$$

$$X^{(s+1)}=\mathop{\mathbf{argmin}}_{X\in\mathcal{X}}\|X-Y^{(s)}\|^2. \tag{14b}$$

Using P_S for least squares projection on S we can write

$$Y^{(s)}=P_{\mathcal{Y}}(X^{(s)})=P_{\mathcal{Y}}(P_{\mathcal{X}}(Y^{(s-1)})), \tag{15a}$$

$$X^{(s+1)}=P_{\mathcal{X}}(Y^{(s)})=P_{\mathcal{X}}(P_{\mathcal{Y}}(X^{(s)})). \tag{15b}$$

In the missing data case ALS is attractive because projection on $\mathcal{X}$ is very simple. For $(i,j)\in\mathcal{K}$ we always set $x_{ij}^{(s)}=z_{ij}$ and for $(i,j)\notin\mathcal{K}$ we set $x_{ij}^{(s)}=y_{ij}^{(s-1)}$. This means, of course, that $P_{\mathcal{X}}$ is a very simple linear projector, which requires virtually no computation.

Clearly the iterative process produces a non-increasing sequences of loss function values, because

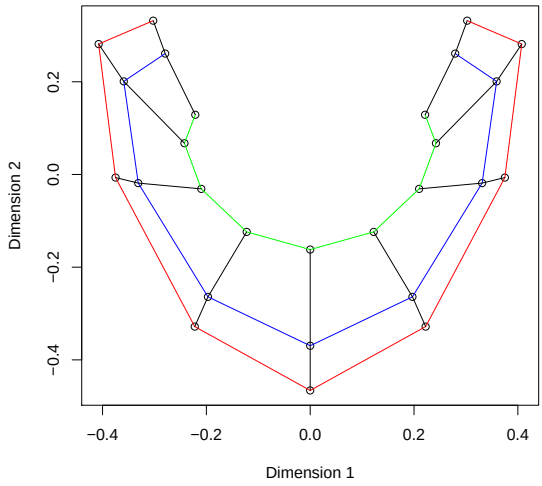
$$\sigma(X^{(s+1)},Y^{(s)})\leq\sigma(X^{(s)},Y^{(s)})\leq\sigma(X^{(s)},Y^{(s-1)}),$$

where the inequalities will usually be strict (if they are not, we just stop, and we are done). Since loss function values are bounded below by zero, they converge. Convergence will generally be linear, with convergence speed equal to the spectral radius

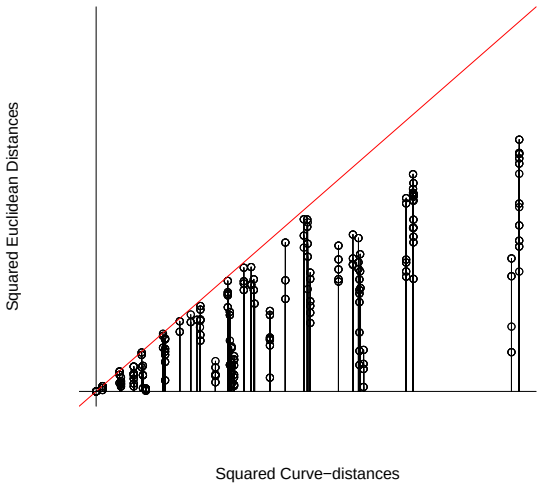
$$\rho \triangleq \|\frac{\partial(P_X \circ P_Y)}{\partial X}\| = \|\frac{\partial(P_Y \circ P_X)}{\partial Y}\|$$

at the solution.

APPENDIX F. FIGURES

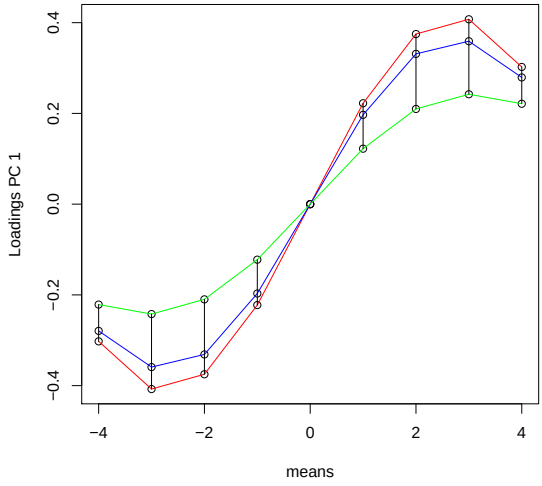


(a) Loadings for 27 Normals

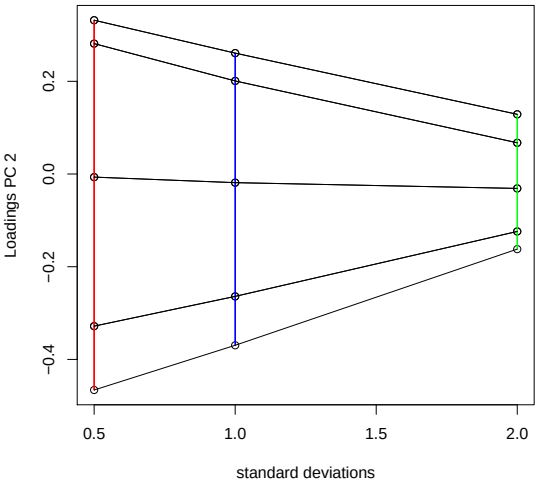


(b) Benzécri Plot for 27 Normals

FIGURE 1. Results for 27 Normals (L2)

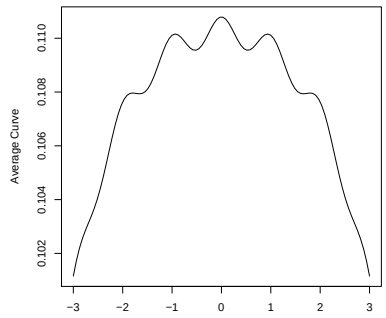


(a) First Principal Curve Against Mean

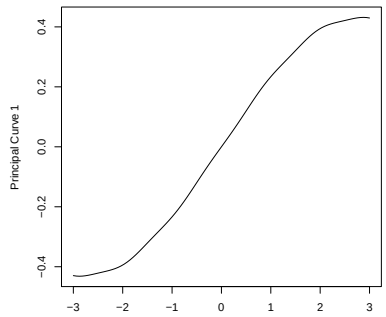


(b) Second Principal Curve Against Standard Deviation

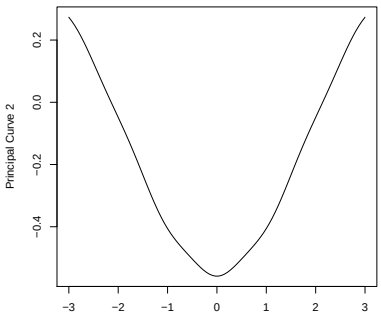
FIGURE 2. Loadings for 27 Normals



(a) Average Curve

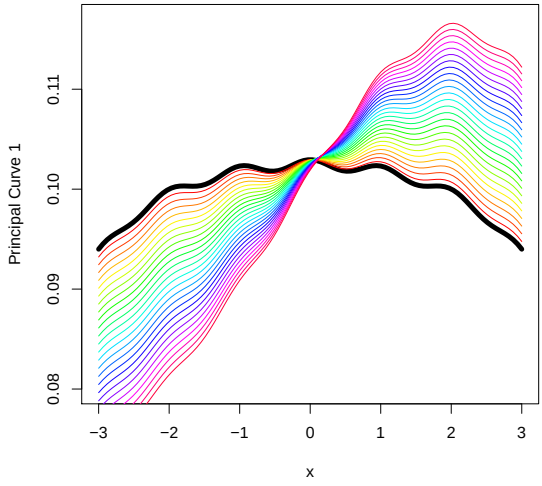


(b) First Principal Curve

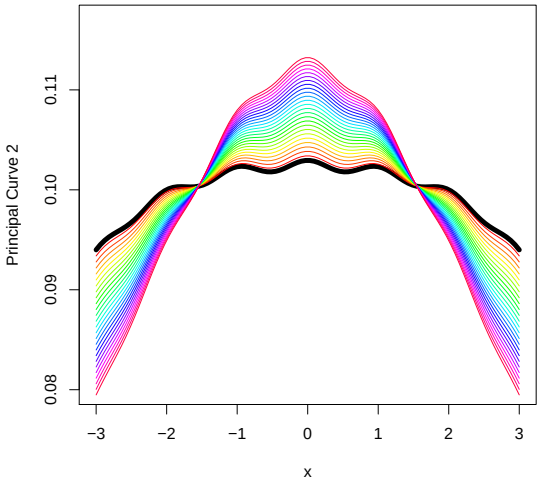


(c) Second Principal Curve

FIGURE 3. Principal Curves for 27 Normals (L2)

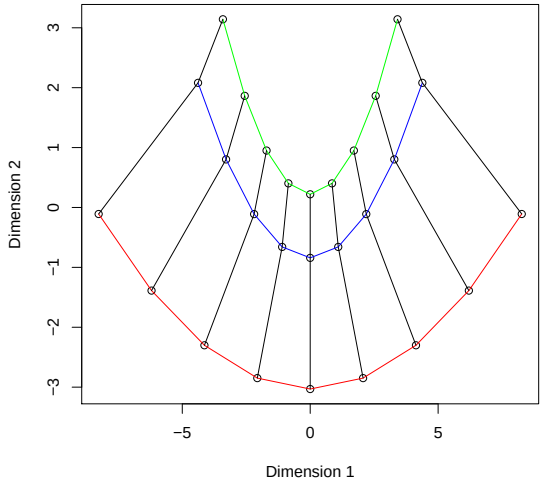


(a) First Principal Curve Perturbations

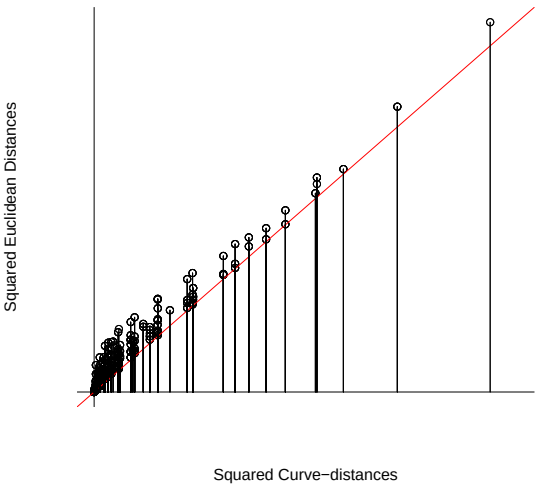


(b) Second Principal Curve Perturbations

FIGURE 4. Principal Curves for 27 Normals (Average Perturbations)

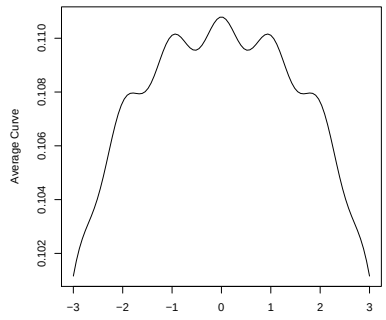


(a) Loadings for 27 Normals

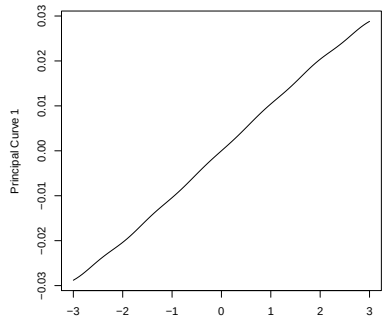


(b) Benzécri Plot for 27 Normals

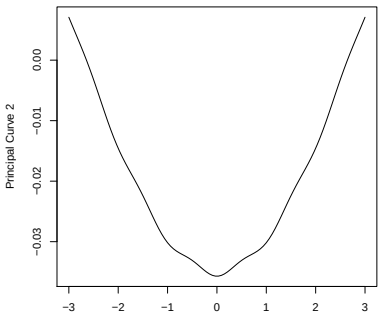
FIGURE 5. Results for 27 Normals (KL)



(a) Average Curve

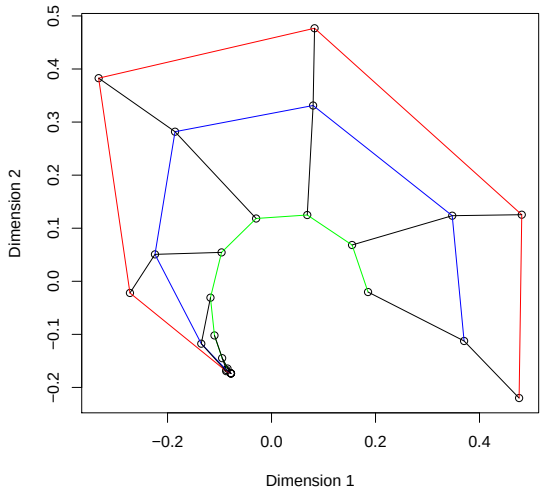


(b) First Principal Curve

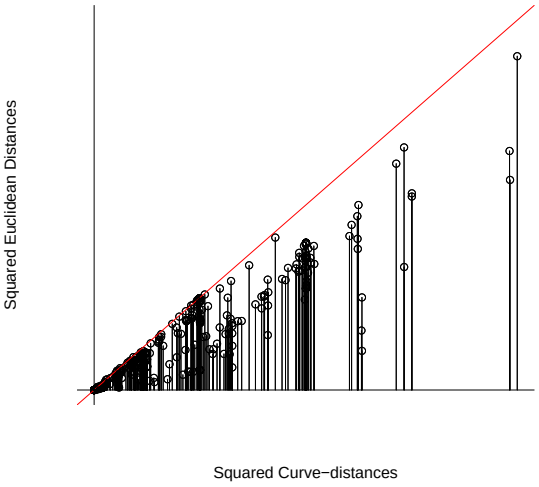


(c) Second Principal Curve

FIGURE 6. Principal Curves for 27 Normals (KL)

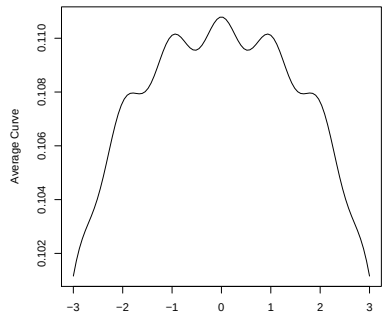


(a) Loadings for 27 Normals

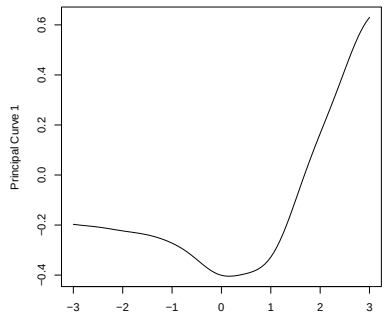


(b) Benzécri Plot for 27 Normals

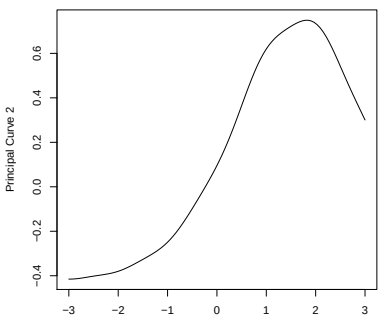
FIGURE 7. Results for 27 Normals (TR)



(a) Average Curve



(b) First Principal Curve



(c) Second Principal Curve

FIGURE 8. Principal Curves for 27 Normals (TR)

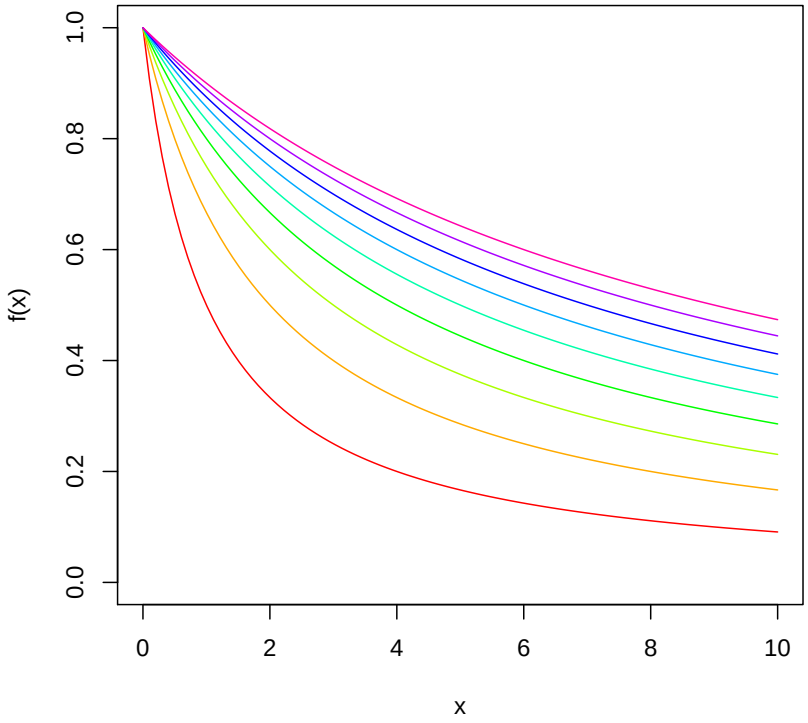
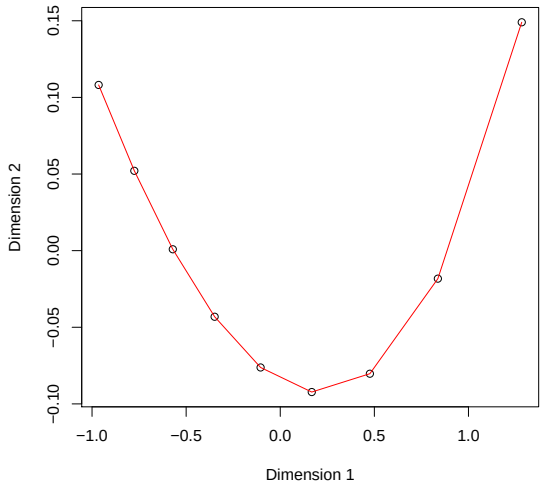
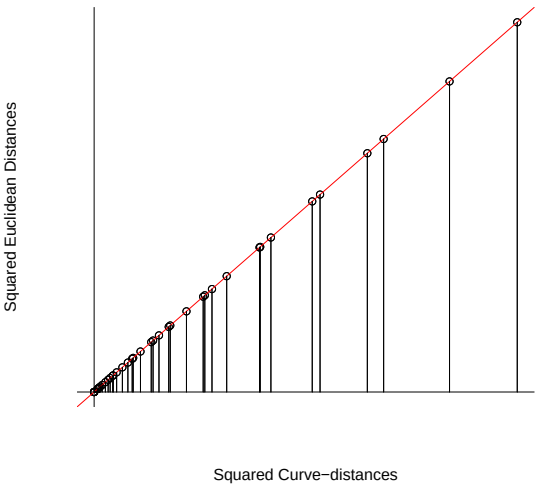


FIGURE 9. Nine Simple rationals

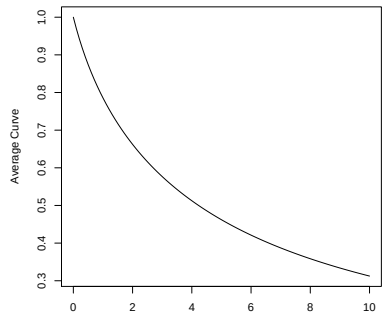


(a) Loadings for 9 Rationals

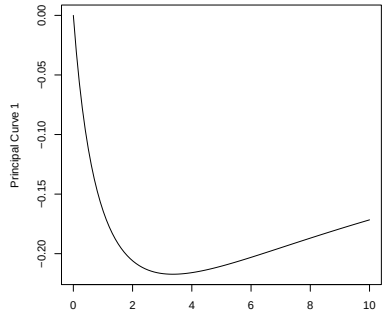


(b) Benzécri Plot for 9 Rationals

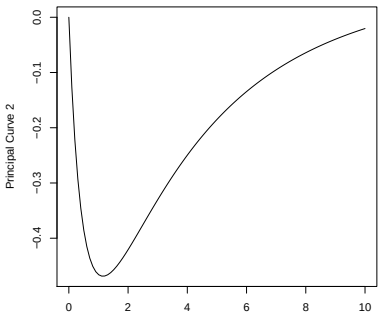
FIGURE 10. Results for 9 Rationals (LS)



(a) Average Curve

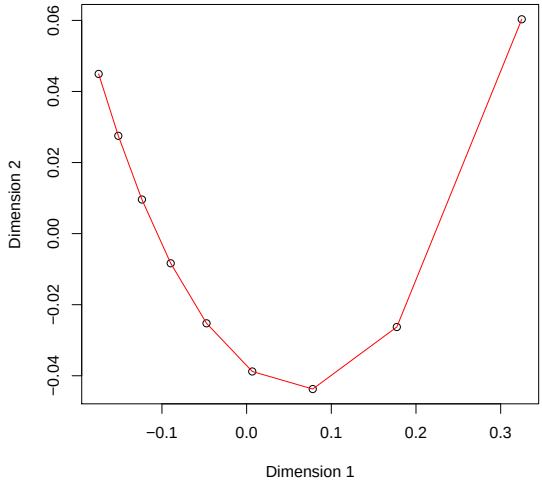


(b) First Principal Curve

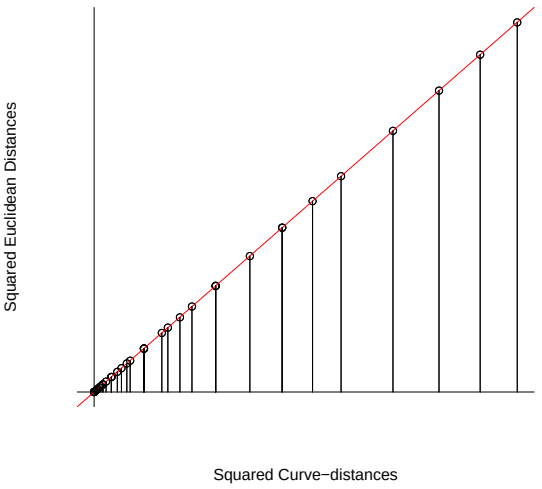


(c) Second Principal Curve

FIGURE 11. Principal Curves for 9 Rationals (LS)

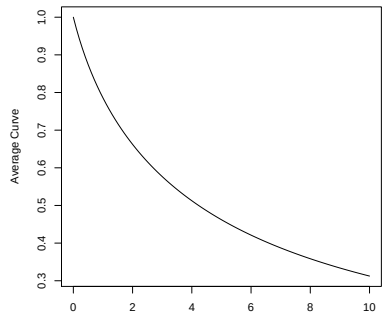


(a) Loadings for 9 Rationals

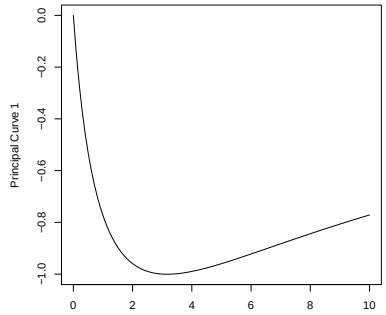


(b) Benzécri Plot for 9 Rationals

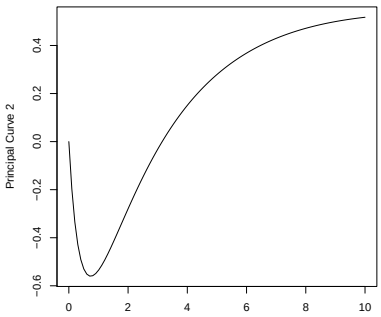
FIGURE 12. Results for 9 Rationals (SP)



(a) Average Curve

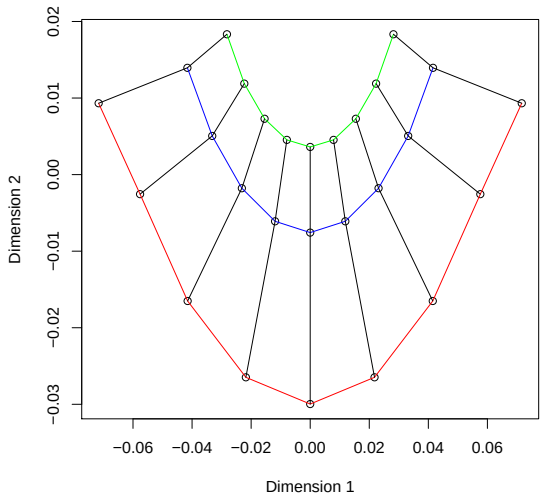


(b) First Principal Curve

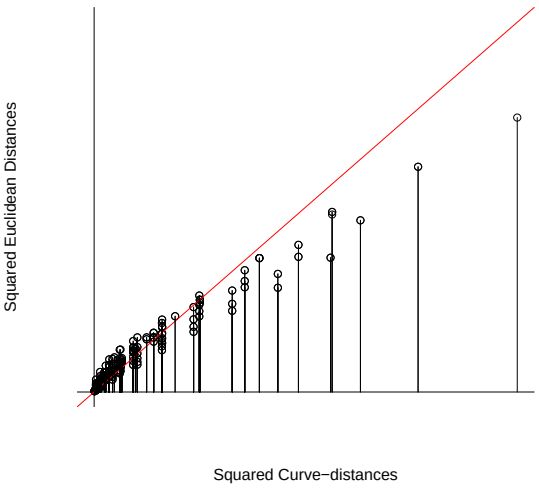


(c) Second Principal Curve

FIGURE 13. Principal Curves for 9 Rationals (SP)

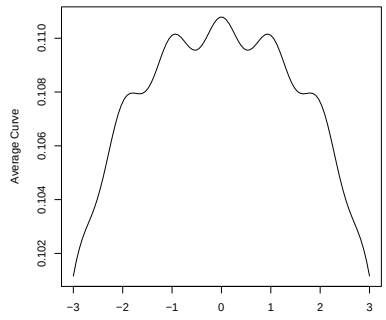


(a) Loadings for 27 Normals

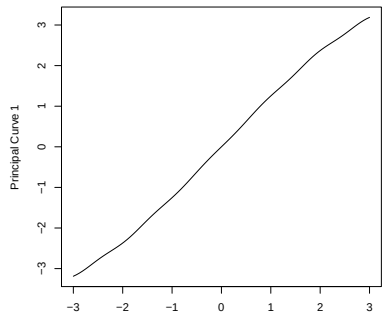


(b) Benzécri Plot for 27 Normals

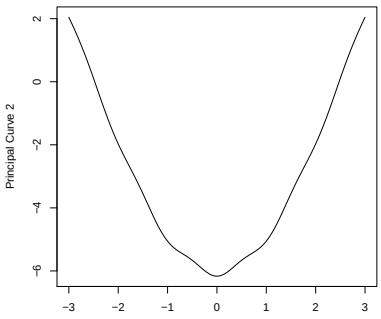
FIGURE 14. Results for 27 Normals Two Sided



(a) Average Curve



(b) First Principal Curve



(c) Second Principal Curve

FIGURE 15. Principal Curves 27 Normals Two Sided

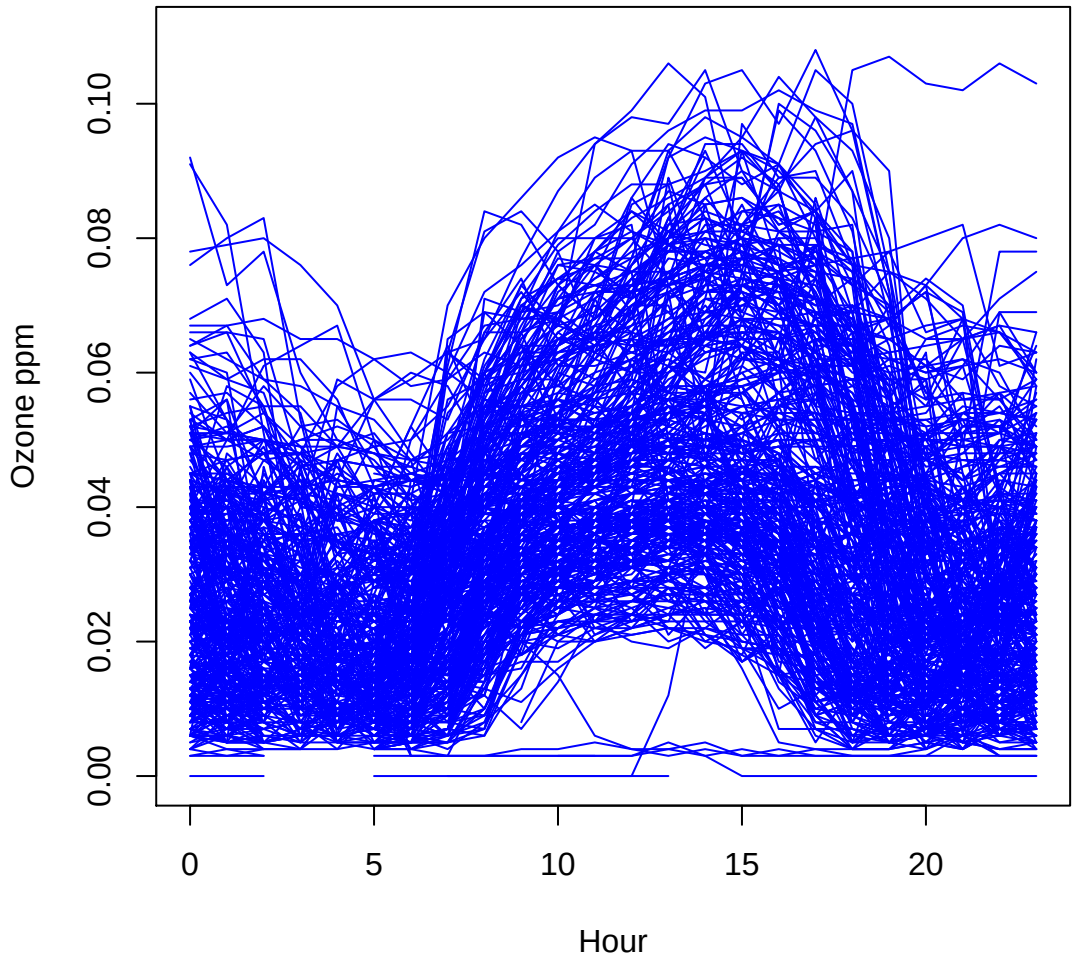
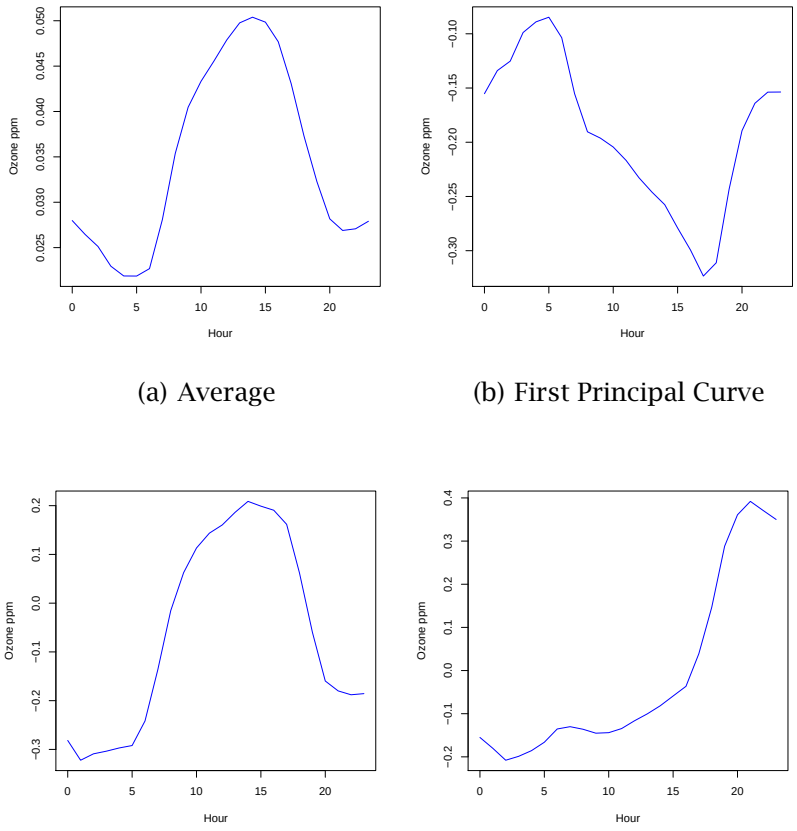


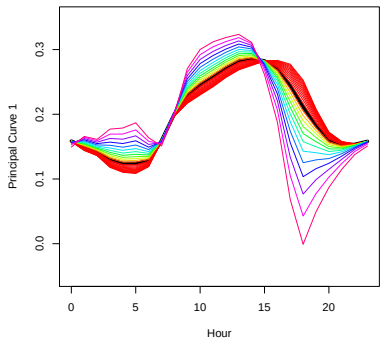
FIGURE 16. Lebec Data



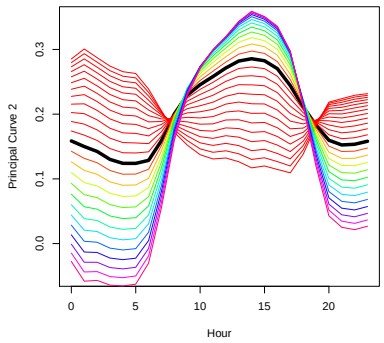
(a) Average (b) First Principal Curve

(c) Second Principal Curve (d) Third Principal Curve

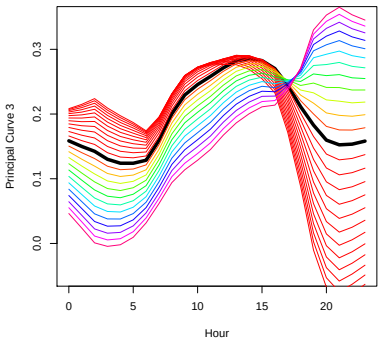
FIGURE 17. Principal Curves for Lebec



(a) First Principal Curve



(b) Second Principal Curve



(c) Third Principal Curve

FIGURE 18. Principal Curves for Lebec as Perturbations

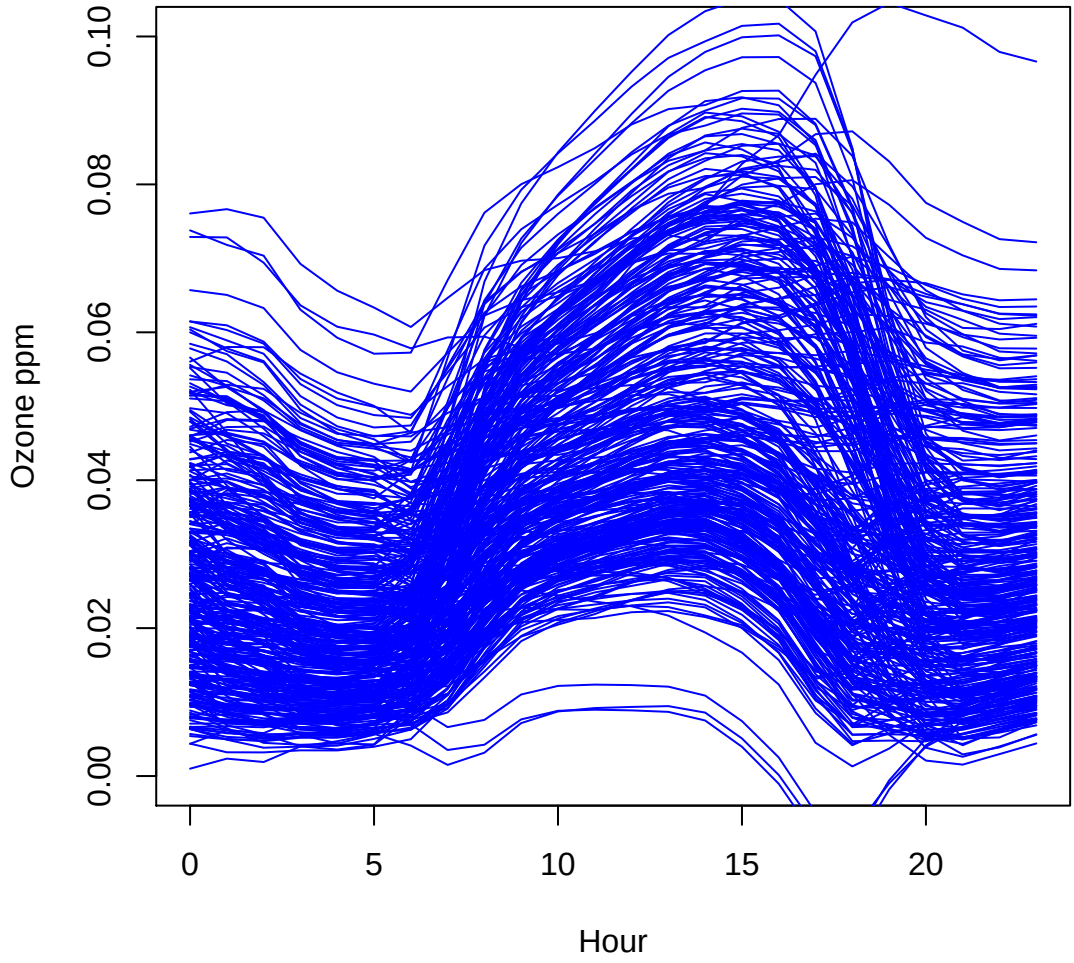
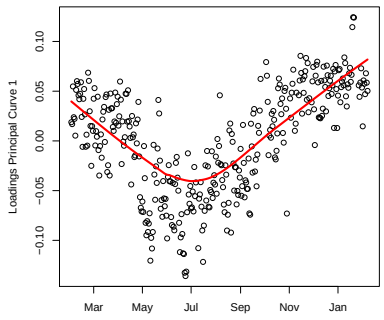
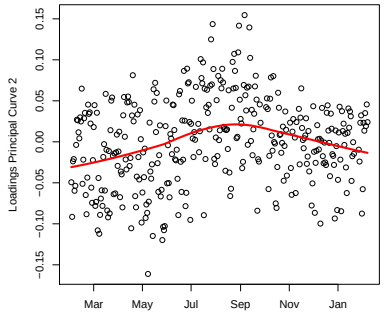


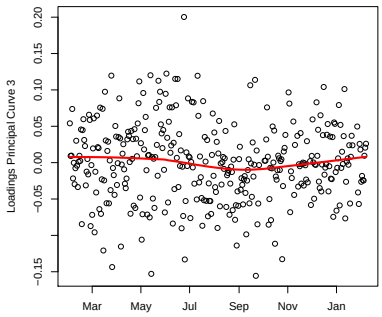
FIGURE 19. Lebec SVD Fitted



(a) First Principal Curve

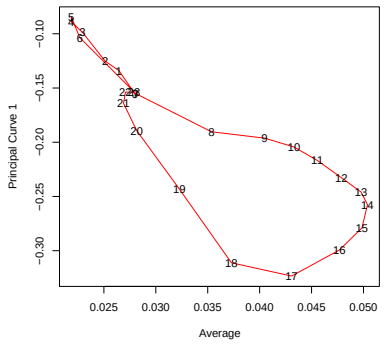


(b) Second Principal Curve

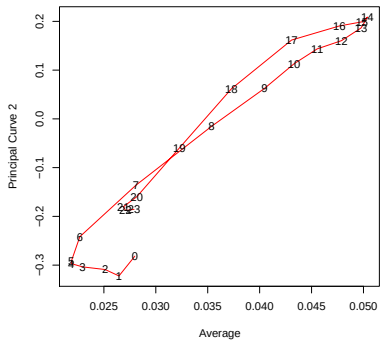


(c) Third Principal Curve

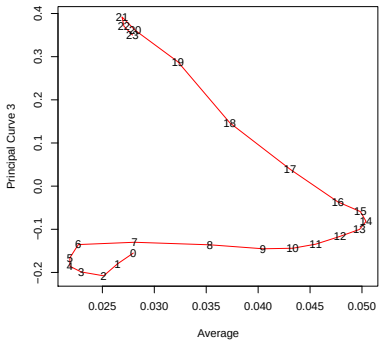
FIGURE 20. Loadings Principal Curves Lebec Against Time



(a) First Principal Curve



(b) Second Principal Curve



(c) Third Principal Curve

FIGURE 21. Principal Curves Lebec Against Average Curve

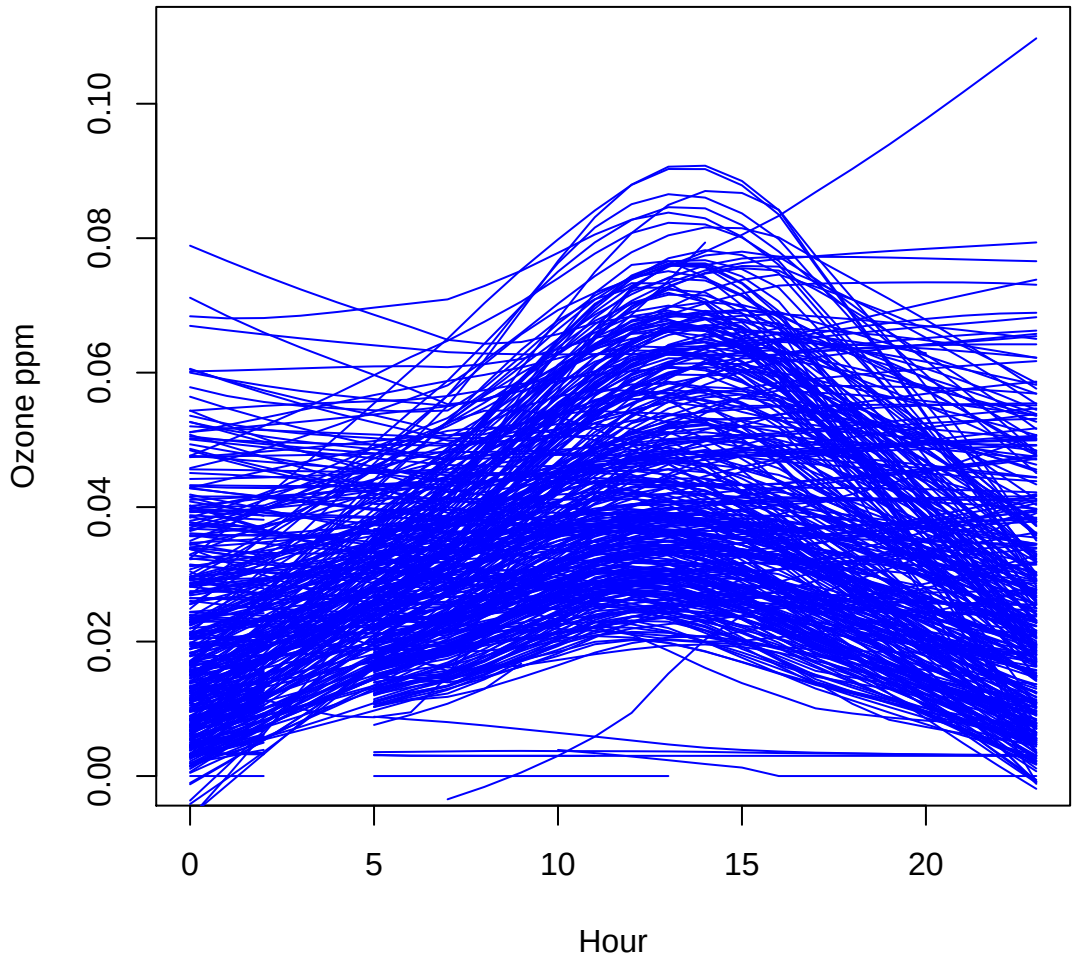
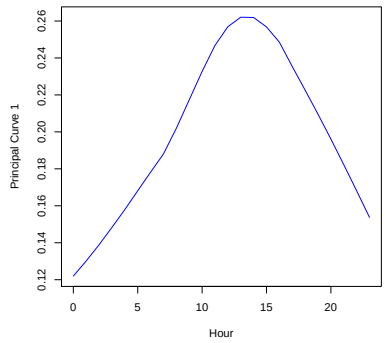
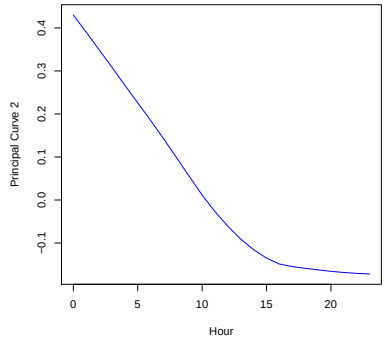


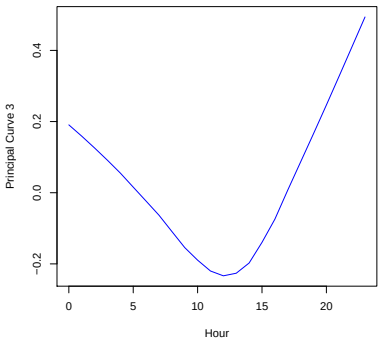
FIGURE 22. Lebec Lowess



(a) First Principal Curve

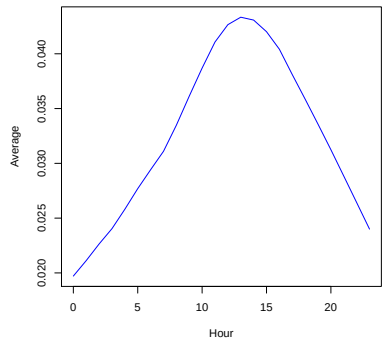


(b) Second Principal Curve

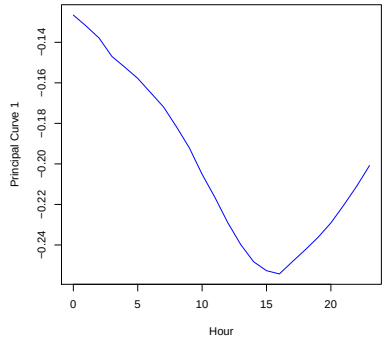


(c) Third Principal Curve

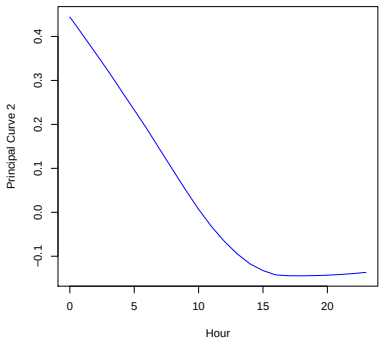
FIGURE 23. Principal Curves for Lebec Lowess Uncentered



(a) First Principal Curve



(b) Second Principal Curve



(c) Third Principal Curve

FIGURE 24. Principal Curves for Lebec Lowess Centered

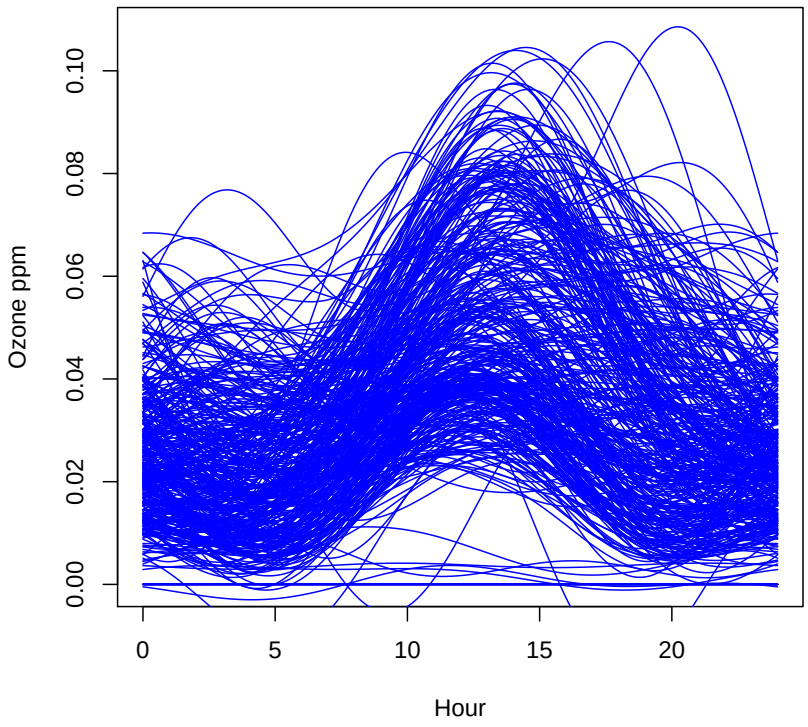


FIGURE 25. Lebec Fourier

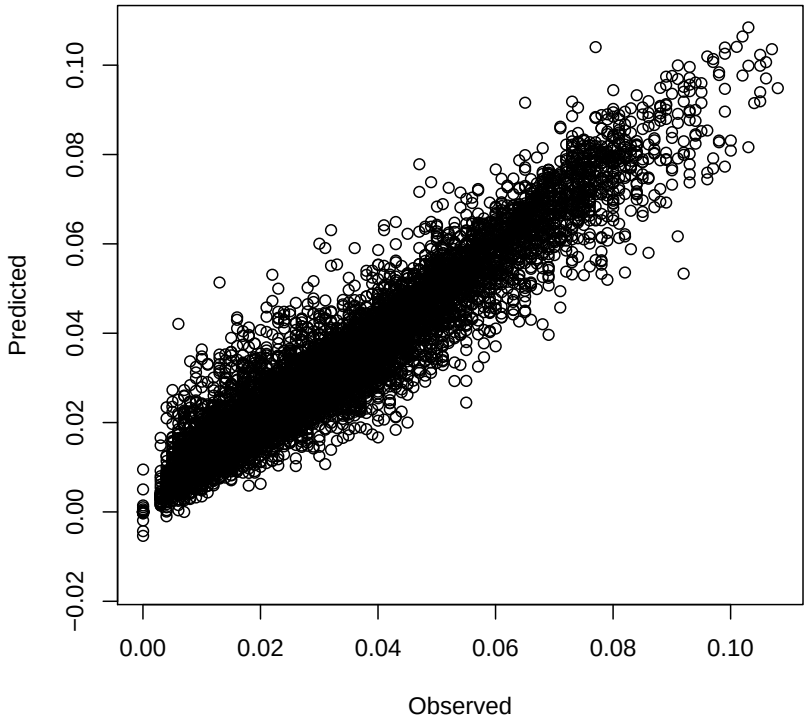
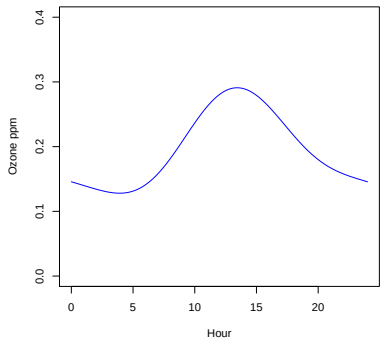
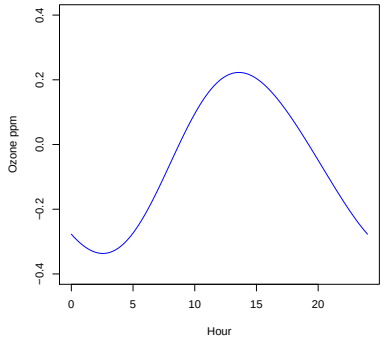


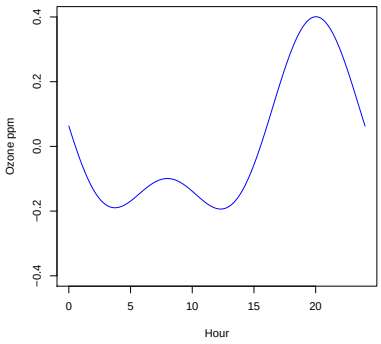
FIGURE 26. Scatterplot Lebec Fit



(a) First Principal Curve



(b) Second Principal Curve



(c) Third Principal Curve

FIGURE 27. Principal Curves for Lebec (LS)

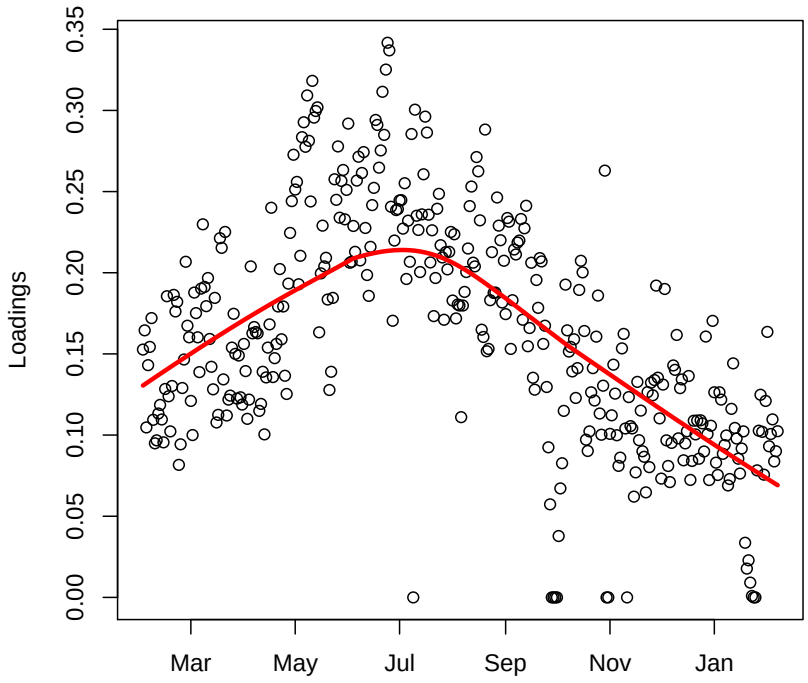


FIGURE 28. Scatterplot Lebec Loadings, First PC

APPENDIX G. TABLES

TABLE 1. Eigenvalues for 27 Normals (L2)

	Eigenvalue	Percentage	Cumulative
1	1.94	0.31	0.31
2	1.41	0.22	0.53
3	0.94	0.15	0.68
4	0.68	0.11	0.78
5	0.47	0.07	0.86
6	0.34	0.05	0.91
7	0.24	0.04	0.95
8	0.19	0.03	0.98
9	0.07	0.01	0.99
10	0.05	0.01	1.00
11	0.01	0.00	1.00
12	0.00	0.00	1.00
13	0.00	0.00	1.00
14	0.00	0.00	1.00
15	0.00	0.00	1.00
16	0.00	0.00	1.00
17	0.00	0.00	1.00
18	0.00	0.00	1.00
19	0.00	0.00	1.00
20	0.00	0.00	1.00
21	0.00	0.00	1.00
22	0.00	0.00	1.00
23	0.00	0.00	1.00
24	0.00	0.00	1.00
25	0.00	0.00	1.00
26	0.00	0.00	1.00
27	0.00	0.00	1.00

TABLE 2. Eigenvalues for 27 Normals (GH)

	1	2	3	4	5	6	7	8	9	10	11
1	1.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	1.11	1.00	0.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	1.20	0.88	0.46	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	1.47	1.02	0.58	0.56	0.36	0.00	0.00	0.00	0.00	0.00	0.00
6	1.48	1.01	0.46	0.43	0.19	0.01	0.00	0.00	0.00	0.00	0.00
7	1.58	1.11	0.59	0.43	0.26	0.17	0.13	0.00	0.00	0.00	0.00
8	1.72	1.21	0.75	0.52	0.34	0.27	0.21	0.00	0.00	0.00	0.00
9	1.76	1.25	0.77	0.53	0.31	0.22	0.16	0.00	0.00	0.00	0.00
10	1.79	1.28	0.79	0.55	0.33	0.23	0.11	0.04	0.01	0.00	0.00
11	1.84	1.33	0.85	0.61	0.39	0.27	0.18	0.17	0.03	0.00	0.00
12	1.90	1.39	0.93	0.68	0.48	0.35	0.25	0.19	0.04	0.00	0.00
13	1.92	1.41	0.95	0.70	0.49	0.35	0.25	0.19	0.03	0.00	0.00
14	1.92	1.40	0.94	0.68	0.47	0.33	0.24	0.19	0.03	0.00	0.00
15	1.93	1.40	0.93	0.67	0.46	0.33	0.24	0.19	0.03	0.01	0.00
16	1.93	1.40	0.94	0.68	0.47	0.34	0.24	0.19	0.04	0.03	0.00
17	1.93	1.40	0.94	0.68	0.47	0.34	0.24	0.19	0.05	0.04	0.00
18	1.94	1.41	0.94	0.68	0.47	0.34	0.24	0.19	0.06	0.04	0.00
19	1.94	1.40	0.94	0.68	0.47	0.34	0.24	0.19	0.06	0.04	0.00
20	1.94	1.40	0.94	0.68	0.47	0.34	0.24	0.19	0.06	0.04	0.00
∞	1.94	1.40	0.94	0.68	0.47	0.34	0.24	0.19	0.07	0.05	0.01

TABLE 3. Eigenvalues for 27 Normals (TR)

	Eigenvalue	Percentage	Cumulative
1	1.18	0.37	0.37
2	0.98	0.31	0.68
3	0.51	0.16	0.84
4	0.29	0.09	0.93
5	0.16	0.05	0.98
6	0.05	0.02	1.00
7	0.00	0.00	1.00
8	0.00	0.00	1.00
9	0.00	0.00	1.00
10	0.00	0.00	1.00
11	0.00	0.00	1.00
12	0.00	0.00	1.00
13	0.00	0.00	1.00
14	0.00	0.00	1.00
15	0.00	0.00	1.00
16	0.00	0.00	1.00
17	0.00	0.00	1.00
18	0.00	0.00	1.00
19	0.00	0.00	1.00
20	0.00	0.00	1.00
21	0.00	0.00	1.00
22	0.00	0.00	1.00
23	0.00	0.00	1.00
24	0.00	0.00	1.00
25	0.00	0.00	1.00
26	0.00	0.00	1.00
27	0.00	0.00	1.00

TABLE 4. Eigenvalues for Nine Rationals (LS)

	Eigenvalue	Percentage	Cumulative
1	4.59	0.99	0.99
2	0.06	0.01	1.00
3	0.00	0.00	1.00
4	0.00	0.00	1.00
5	0.00	0.00	1.00
6	0.00	0.00	1.00
7	0.00	0.00	1.00
8	0.00	0.00	1.00
9	0.00	0.00	1.00

TABLE 5. Eigenvalues for 27 Normals (KL)

	Eigenvalue	Percentage	Cumulative
1	372.14	0.66	0.66
2	80.31	0.14	0.80
3	0.00	0.00	0.80
4	0.00	0.00	0.80
5	0.00	0.00	0.80
6	0.00	0.00	0.80
7	0.00	0.00	0.80
8	0.00	0.00	0.80
9	0.00	0.00	0.80
10	0.00	0.00	0.80
11	0.00	0.00	0.80
12	0.00	0.00	0.80
13	0.00	0.00	0.80
14	0.00	0.00	0.80
15	0.00	0.00	0.80
16	0.00	0.00	0.80
17	0.00	0.00	0.80
18	0.00	0.00	0.80
19	0.00	0.00	0.80
20	0.00	0.00	0.80
21	0.00	0.00	0.80
22	0.00	0.00	0.80
23	0.00	0.00	0.80
24	0.00	0.00	0.80
25	0.00	0.00	0.80
26	−52.46	0.09	0.90
27	−57.14	0.10	1.00

TABLE 6. Eigenvalues for Nine Rationals (SP)

	Eigenvalue	Percentage	Cumulative
1	0.22	0.95	0.95
2	0.01	0.05	1.00
3	0.00	0.00	1.00
4	0.00	0.00	1.00
5	0.00	0.00	1.00
6	0.00	0.00	1.00
7	0.00	0.00	1.00
8	0.00	0.00	1.00
9	0.00	0.00	1.00

TABLE 7. Eigenvalues for Lebec

	1	2	3	4
1	12.61	1.98	11.74	1.87
2	0.35	0.34	0.32	0.32
3	0.26	0.23	0.23	0.20
4	0.10	0.08	0.02	0.02
5	0.07	0.07	0.00	0.00
6	0.04	0.04	0.00	0.00
7	0.04	0.04	0.00	0.00
8	0.03	0.03	0.00	0.00
9	0.02	0.02	0.00	0.00
10	0.02	0.02	0.00	0.00
11	0.01	0.01	0.00	0.00
12	0.01	0.01	0.00	0.00
13	0.01	0.01	0.00	0.00
14	0.01	0.01	0.00	0.00
15	0.01	0.01	0.00	0.00
16	0.01	0.01	0.00	0.00
17	0.01	0.01	0.00	0.00
18	0.01	0.01	0.00	0.00
19	0.01	0.01	0.00	0.00
20	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00
mean	–	10.71	–	9.93
total	13.64	13.65	12.31	12.34

Column 1: SVD
Column 2: SVD Centered
Column 3: SVD Lowess
Column 4: SVD Lowess Centered

TABLE 8. Eigenvalues for Lebec-Fourier

	Eigenvalue	Percentage	Cumulative
1	12.625	0.946	0.946
2	0.337	0.025	0.971
3	0.248	0.019	0.990
4	0.072	0.005	0.995
5	0.068	0.005	1.000

APPENDIX H. CODE

H.1. FPCA.

```
source("./util.R")

FPCA<-function(x,ndim,defunc,deprin,wghts=FALSE,npts=100) {
res<-defunc(x); cp<-res$cp; de<-res$de; n<-nrow(x)
ea<-eigen(cp); ev<-ea$values
xv<-(ea$vectors[,1:ndim])*outer(rep(1,n),sqrt(ev[1:ndim]))
yv<-xv/outer(rep(1,n),colSums(xv^2))
di<-as.matrix(dist(xv))^2
pc<-matrix(0,npts,ndim)
for (i in 1:ndim) pc[,i]<-deprin(yv[,i],x,npts=npts)
result<-list(cp=cp,de=de,di=di,xv=xv,yv=yv,ev=ev,pc=pc)
class(result) <- "fpca"
return(result)
}

plot.fpca<-function(xfpca,plot.type,deprin=linC,intv=c(-3,3),npts=100) {
        xv<-xfpca$xv; yv<-xfpca$yv; de<-xfpca$de; di<-xfpca$di; pc<-xfpca$pc
        ndim<-ncol(pc); n<-nrow(x); asr<-seq(intv[1],intv[2],length=npts)
        if (plot.type == "coord")
                plot(xv,xlab="Dimension 1",ylab="Dimension 2")
        if (plot.type == "benz") {
                ma<-max(c(de,di))
                plot(matrix(c(0,ma,0,ma),2,2),type="n",xlab="Squared Curve-
                    distances",ylab="Squared Euclidean Distances",axes=FALSE)
                points(de,di)
                abline(h=0)
                abline(v=0)
                abline(0,1,col="RED")
                for (i in 1:length(de))
                        lines(rbind(c(de[i],0),c(de[i],di[i])))
                }
        if (plot.type == "pc")
                for (i in 1:ndim) {
                        plot(asr,deprin(yv[,i],x),type="l",xlab="",ylab=
                            paste("Principal Curve",i))
                        }
        if (plot.type == "ave")
                plot(asr,deprin(rep(1/n,n),x),xlab="",ylab="Average Curve",
                    type="l")
```

```
}

LSNorm<-function(x) {
n<-nrow(x); ip<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n) {
        sg<-sqrt((x[i,2]^2)+(x[j,2]^2))
        ip[i,j]<-dnorm(x[i,1]-x[j,1],mean=0,sd=sg)
        }
de<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n)
        de[i,j]<-ip[i,i]+ip[j,j]-2*ip[i,j]
cp<-doublyCenter(ip)
return(list(cp=cp,de=de))
}

TRNorm<-function(x) {
n<-nrow(x); ip<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n) {
        sg<-sqrt((x[i,2]^2)+(x[j,2]^2))
        sh<-x[i,2]*x[j,2]/sg
        mh<-((x[i,2]^2)*x[j,1]+(x[j,2]^2)*x[i,1])/((x[i,2]^2)+(x[j,2]^2))
        ip[i,j]<-dnorm(x[i,1]-x[j,1],mean=0,sd=sg)*(1-pnorm(-mh/sh))
        }
de<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n)
        de[i,j]<-ip[i,i]+ip[j,j]-2*ip[i,j]
cp<-doublyCenter(ip)
return(list(cp=cp,de=de))
}

GHNorm<-function(x) {
n<-nrow(x); ip<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n) {
        xx<-GH[,1]; ww<-GH[,2]
        d1<-dnorm(xx,mean=x[i,1],sd=x[i,2])
        d2<-dnorm(xx,mean=x[j,1],sd=x[j,2])
        ip[i,j]<-sum(ww*exp(xx^2)*d1*d2)
        }
de<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n)
        de[i,j]<-ip[i,i]+ip[j,j]-2*ip[i,j]
cp<-doublyCenter(ip)
return(list(cp=cp,de=de))
```

```
}

KLNorm<-function(x) {
n<-nrow(x); de<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n) {
        dm<-(x[i,1]-x[j,1])^2; s1<-x[i,2]^2; s2<-x[j,2]^2
        de[i,j]<-(((s1+dm)/s2)+((s2+dm)/s1)-2)/2
        }
cp<--0.5*doublyCenter(de)
return(list(cp=cp,de=de))
}

LSRatio<-function(x) {
n<-nrow(x); ip<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n)
        ip[i,j]<-(x[i]*x[j]/(x[i]-x[j]))*log(x[i]/x[j])
for (i in 1:n) ip[i,i]<-x[i]
de<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n)
        de[i,j]<-ip[i,i]+ip[j,j]-2*ip[i,j]
cp<-doublyCenter(ip)
return(list(cp=cp,de=de))
}

SPRatio<-function(x) {
n<-nrow(x); de<-matrix(0,n,n)
for (i in 1:n) for (j in 1:n)
        de[i,j]<-abs((sqrt(x[i])-sqrt(x[j]))/(sqrt(x[i])+sqrt(x[j])))
de<-de^2
cp<--0.5*doublyCenter(de)
return(list(cp=cp,de=de))
}

linC<-function(b,x,intv=c(-3,3),npts=100) {
p<-seq(intv[1],intv[2],length=npts); n<-nrow(x)
q<-rep(0,npts)
for (i in 1:length(b))
        q<-q+b[i]*dnorm(p,mean=x[i,1],sd=x[i,2])
return(q)
}

linR<-function(b,x,intv=c(0,10),npts=100) {
p<-seq(intv[1],intv[2],length=npts); n<-nrow(x)
```

```
q<-rep(0,npts)
for (i in 1:length(b))
        q<-q+b[i]*x[i]/(x[i]+p)
return(q)
}
```

H.2. **Imputation.**

```
imputeMat<-function(mat,fitme,eps=1e-6,niter=100,verbose=TRUE,pars=NULL) {
        n<-nrow(mat); m<-ncol(mat); oloss<-Inf; iter<-1
        fitted<-matrix(0,n,m); imputed<-mat
        repeat {
        for (i in 1:n) {
                ind<-which(is.na(mat[i,]))
                imputed[i,ind]<-fitted[i,ind]
                }
        nloss<-sum((imputed-fitted)^2)
        motor<-fitme(imputed,pars); fitted<-motor$fitted; extra<-motor$extra
        if (verbose)
                cat("Iteration:␣",formatC(iter,digits=6,width=6),
                        "␣␣␣Loss:␣␣␣",formatC(oloss,digits=6,width=12,format=
                            "f"),
                        "␣==>",formatC(nloss,digits=6,width=12,format="f"),"\
                            n")
        if (((oloss-nloss) < eps) || (iter == niter)) break();
        oloss<-nloss; iter<-iter+1
        }
return(list(iter=iter,loss=nloss,fitted=fitted,extra=extra))
}

fitSVDRaw<-function(mat,pars) {
        sv<-svd(mat,nu=pars,nv=pars)
        return(list(itted=tcrossprod(sv$u,(sv$v)%*%diag(sv$d[1:pars])),extra=
            sv))
}

fitSVDCenter<-function(mat,pars) {
        n<-nrow(mat); av<-as.vector(apply(mat,2,mean)); mav<-outer(rep(1,n),
            av)
        sv<-svd(mat-mav,nu=pars,nv=pars)
        return(list(fitted=mav+tcrossprod(sv$u,(sv$v)%*%diag(sv$d[1:pars])),
            extra=list(av,sv)))
}
```

```
31
32 fitSVDCentor<-function(mat,pars) {
33         n<-nrow(mat); av<-apply(mat,2,mean); mav<-outer(rep(1,n),av)
34         sv<-svd(mat-mav,nu=pars,nv=pars)
35         u<-cbind(1,sv$u); v<-cbind(av,(sv$v)%*%diag(sv$d[1:pars]))
36         s<-chol(crossprod(v)); ss<-s/diag(s)
37         v<-v%*%solve(ss); u<-u%*%t(ss)
38         return(list(fitted=tcrossprod(u,v),extra=list(u,v)))
39 }
40
41 fitSVDDouble<-function(mat,pars) {
42         n<-nrow(mat); av<-apply(mat,2,mean); mav<-outer(rep(1,n),av)
43         sv<-svd(mat-mav,nu=pars,nv=pars)
44         yy<-cbind(av,sv$v); xx<-cbind(1,(sv$u)%*%diag(sv$d[1:pars]))
45         cc<-crossprod(yy); ch<-chol(cc); ch<-ch/diag(ch)
46         yy<-yy%*%solve(ch); xx<-tcrossprod(xx,ch)
47         return(list(fitted=tcrossprod(xx,yy),extra=list(av,sv)))
48 }
49
50 fitOrth<-function(mat,pars) {
51 fcoef<-as.matrix(mat)%*%pars
52 return(list(fitted=tcrossprod(fcoef,pars),extra=fcoef))
53 }
```

H.3. **Miscellaneous Utilities.**

```
1  doublyCenter<-function(x) {
2          return(x+mean(x)-outer(apply(x,1,mean),apply(x,2,mean),"+"))
3  }
4
5  lowessMe<-function(mat,x=1:ncol(mat)) {
6  n<-nrow(mat); mlow<-mat
7  for (i in 1:n)
8          {
9          ind<-which(!is.na(mat[i,]))
10         if (length(ind)==0) next()
11         lwa<-lowess(x[ind],mat[i,ind])
12         mlow[i,ind]<-lwa$y
13         }
14 return(mlow)
15 }
```

H.4. **Code for Figures.**

H.4.1. *Code for Figure 1.*

```
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
LSN<-FPCA(x,ndim=2,defunc=LSNorm,deprin=linC)
pdf("LSNorm27coord.pdf")
plot(LSN,plot.type="coord")
xv<-LSN$xv
lines(xv[1:9,],col="RED")
lines(xv[10:18,],col="BLUE")
lines(xv[19:27,],col="GREEN")
for (i in 1:9) lines(xv[i+c(0,9,18),])
dev.off()
pdf("LSNorm27benz.pdf")
plot(LSN,plot.type="benz")
dev.off()
```

H.4.2. *Code for Figure 2.*

```
pdf("LSNormLoadParpc1.pdf")
plot(x[,1],xv[,1],xlab="means",ylab="Loadings␣PC␣1")
lines(x[1:9,1],xv[1:9,1],col="RED")
lines(x[10:18,1],xv[10:18,1],col="BLUE")
lines(x[19:27,1],xv[19:27,1],col="GREEN")
for (i in 1:9) lines(x[i+c(0,9,18),1],xv[i+c(0,9,18),1])
dev.off()
pdf("LSNormLoadParpc2.pdf")
plot(x[,2],xv[,2],xlab="standard␣deviations",ylab="Loadings␣PC␣2")
lines(x[1:9,2],xv[1:9,2],col="RED")
lines(x[10:18,2],xv[10:18,2],col="BLUE")
lines(x[19:27,2],xv[19:27,2],col="GREEN")
for (i in 1:9) lines(x[i+c(0,9,18),2],xv[i+c(0,9,18),2])
dev.off()
```

H.4.3. *Code for Figure 3.*

```
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
LSN<-FPCA(x,ndim=2,defunc=LSNorm,deprin=linC)
yv<-LSN$yv; pc<-LSN$pc; n<-27
pdf("LSNorm27ave.pdf")
```

```
plot(seq(-3,3,length=100),linC(rep(1/n,n),x),xlab="",ylab="Average Curve",
    type="l")
dev.off()
pdf("LSNorm27pc1.pdf")
plot(seq(-3,3,length=100),linC(yv[,1],x),type="l",xlab="",ylab="Principal
    Curve 1")
dev.off()
pdf("LSNorm27pc2.pdf")
plot(seq(-3,3,length=100),linC(yv[,2],x),type="l",xlab="",ylab="Principal
    Curve 2")
dev.off()
```

H.4.4. *Code for Figure 4.*

```
pdf("LSNormPerturbpc1.pdf")
p<-seq(-3,3,length=100); u<-rep(1/27,27); cc<-rainbow(25)
q<-linC(u,x)
plot(p,q/sqrt(sum(q^2)),type="l",lwd=5,ylim=c(.080,.117),xlab="x",ylab="
    Principal Curve 1")
for (i in 1:25) {
q<-linC(u+i*.001*xv[,1],x)
lines(p,q/sqrt(sum(q^2)),col=cc[i])
}
dev.off()
pdf("LSNormPerturbpc2.pdf")
q<-linC(u,x)
plot(p,q/sqrt(sum(q^2)),type="l",lwd=5,ylim=c(.080,.117),xlab="x",ylab="
    Principal Curve 2")
for (i in 1:25) {
q<-linC(u-i*.001*xv[,2],x)
lines(p,q/sqrt(sum(q^2)),col=cc[i])
}
dev.off()
```

H.4.5. *Code for Figure 5.*

```
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
KLN<-FPCA(x,ndim=2,defunc=KLNorm,deprin=linC)
pdf("KLNorm27coord.pdf")
plot(KLN,plot.type="coord")
xv<-KLN$xv
lines(xv[1:9,],col="RED")
```

```
lines(xv[10:18,],col="BLUE")
lines(xv[19:27,],col="GREEN")
for (i in 1:9) lines(xv[i+c(0,9,18),])
dev.off()
pdf("KLNorm27benz.pdf")
plot(KLN,plot.type="benz")
dev.off()
```

H.4.6. *Code for Figure 6.*

```
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
KLN<-FPCA(x,ndim=2,defunc=KLNorm,deprin=linC)
yv<-KLN$yv; pc<-KLN$pc; n<-27
pdf("KLNorm27ave.pdf")
plot(seq(-3,3,length=100),linC(rep(1/n,n),x),xlab="",ylab="Average Curve",
    type="l")
dev.off()
pdf("KLNorm27pc1.pdf")
plot(seq(-3,3,length=100),linC(yv[,1],x),type="l",xlab="",ylab="Principal
    Curve 1")
dev.off()
pdf("KLNorm27pc2.pdf")
plot(seq(-3,3,length=100),linC(yv[,2],x),type="l",xlab="",ylab="Principal
    Curve 2")
dev.off()
```

H.4.7. *Code for Figure 7.*

```
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
TRN<-FPCA(x,ndim=2,defunc=TRNorm,deprin=linC)
pdf("TRNorm27coord.pdf")
plot(TRN,plot.type="coord")
xv<-TRN$xv
lines(xv[1:9,],col="RED")
lines(xv[10:18,],col="BLUE")
lines(xv[19:27,],col="GREEN")
for (i in 1:9) lines(xv[i+c(0,9,18),])
dev.off()
pdf("TRNorm27benz.pdf")
plot(TRN,plot.type="benz")
dev.off()
```

H.4.8. *Code for Figure 8.*

```
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
TRN<-FPCA(x,ndim=2,defunc=TRNorm,deprin=linC)
yv<-TRN$yv; pc<-TRN$pc; n<-27
pdf("TRNorm27ave.pdf")
plot(seq(-3,3,length=100),linC(rep(1/n,n),x),xlab="",ylab="Average Curve",
    type="l")
dev.off()
pdf("TRNorm27pc1.pdf")
plot(seq(-3,3,length=100),linC(yv[,1],x),type="l",xlab="",ylab="Principal
    Curve 1")
dev.off()
pdf("TRNorm27pc2.pdf")
plot(seq(-3,3,length=100),linC(yv[,2],x),type="l",xlab="",ylab="Principal
    Curve 2")
dev.off()
```

H.4.9. *Code for Figure 9.*

```
cr<-rainbow(9)
pdf("rationals.pdf")
plot(0:10,seq(0,1,length=11),type="n",xlab="x",ylab="f(x)")
x<-seq(0,10,length=100)
for (a in 1:9) lines(x,a/(x+a),col=cr[a])
dev.off()
```

H.4.10. *Code for Figure 10.*

```
x<-matrix(0,9,1)
x[,1]<-1:9
LSR<-FPCA(x,ndim=2,defunc=LSRatio,deprin=linR)
pdf("LSRatiocoord.pdf")
plot(LSR,plot.type="coord")
xv<-LSR$xv
lines(xv[1:9,],col="RED")
dev.off()
pdf("LSRatiobenz.pdf")
plot(LSR,plot.type="benz")
dev.off()
```

H.4.11. *Code for Figure 11.*

```
x<-matrix(0,9,1)
x[,1]<-1:9
LSR<-FPCA(x,ndim=2,defunc=LSRatio,deprin=linR)
yv<-LSR$yv; pc<-LSR$pc; n<-9
pdf("LSRatioave.pdf")
plot(seq(0,10,length=100),linR(rep(1/n,n),x),xlab="",ylab="Average Curve",
    type="l")
dev.off()
pdf("LSRatiopc1.pdf")
plot(seq(0,10,length=100),linR(yv[,1],x),type="l",xlab="",ylab="Principal
    Curve 1")
dev.off()
pdf("LSRatiopc2.pdf")
plot(seq(0,10,length=100),linR(yv[,2],x),type="l",xlab="",ylab="Principal
    Curve 2")
dev.off()
```

H.4.12. *Code for Figure 12.*

```
x<-matrix(0,9,1)
x[,1]<-1:9
SPR<-FPCA(x,ndim=2,defunc=SPRatio,deprin=linR)
pdf("SPRatiocoord.pdf")
plot(SPR,plot.type="coord")
xv<-SPR$xv
lines(xv[1:9,],col="RED")
dev.off()
pdf("SPRatiobenz.pdf")
plot(SPR,plot.type="benz")
dev.off()
```

H.4.13. *Code for Figure 13.*

```
x<-matrix(0,9,1)
x[,1]<-1:9
SPR<-FPCA(x,ndim=2,defunc=SPRatio,deprin=linR)
yv<-SPR$yv; pc<-SPR$pc; n<-9
pdf("SPRatioave.pdf")
plot(seq(0,10,length=100),linR(rep(1/n,n),x),xlab="",ylab="Average Curve",
    type="l")
dev.off()
```

```
pdf("SPRatiopc1.pdf")
plot(seq(0,10,length=100),linR(yv[,1],x),type="l",xlab="",ylab="Principal Curve 1")
dev.off()
pdf("SPRatiopc2.pdf")
plot(seq(0,10,length=100),linR(yv[,2],x),type="l",xlab="",ylab="Principal Curve 2")
dev.off()
```

H.4.14. *Code for Figure 14.*

```
source("smacofPlain.R")
source("fpca.R")
x<-matrix(0,27,2)
x[,1]<- -4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
KLN<-FPCA(x,ndim=2,defunc=KLNorm,deprin=linC)
diss<-sqrt(as.dist(KLN$de))
KLMDS<-smacofSym(diss)
xv<-KLMDS$x; de<-(as.vector(KLMDS$dhat))^2; di<-(as.vector(KLMDS$d))^2
pdf("KLMDS27coord.pdf")
plot(xv,xlab="Dimension 1",ylab="Dimension 2")
lines(xv[1:9,],col="RED")
lines(xv[10:18,],col="BLUE")
lines(xv[19:27,],col="GREEN")
for (i in 1:9) lines(xv[i+c(0,9,18),])
dev.off()
pdf("KLMDS27benz.pdf")
mm<-max(c(de,di))
plot(matrix(c(0,mm,0,mm),2,2),type="n",xlab="Squared Curve-distances",ylab="Squared Euclidean Distances",axes=FALSE)
points(de,di)
abline(h=0)
abline(v=0)
abline(0,1,col="RED")
for (i in 1:length(de))
        lines(rbind(c(de[i],0),c(de[i],di[i])))
dev.off()
```

H.4.15. *Code for Figure 15.*

```
source("smacofPlain.R")
source("fpca.R")
x<-matrix(0,27,2)
```

```
x[,1]<-−4:4; x[,2]<-c(rep(.5,9),rep(1,9),rep(2,9))
KLN<-FPCA(x,ndim=2,defunc=KLNorm,deprin=linC)
diss<-sqrt(as.dist(KLN$de))
KLMDS<-smacofSym(diss)
xv<-KLMDS$x; yv<-xv%*%solve(crossprod(xv))
pc<-KLN$pc; n<-27
pdf("KLMDS27ave.pdf")
plot(seq(−3,3,length=100),linC(rep(1/n,n),x),xlab="",ylab="Average Curve",
    type="l")
dev.off()
pdf("KLMDS27pc1.pdf")
plot(seq(−3,3,length=100),linC(yv[,1],x),type="l",xlab="",ylab="Principal
    Curve 1")
dev.off()
pdf("KLMDS27pc2.pdf")
plot(seq(−3,3,length=100),linC(yv[,2],x),type="l",xlab="",ylab="Principal
    Curve 2")
dev.off()
```

H.4.16. *Code for Figure 16.*

```
pdf("lebec_data.pdf")
plot(0:23,seq(0,max(leboz,na.rm=TRUE),length=24),type="n",xlab="Hour",ylab="
    Ozone ppm")
for (i in 1:372) lines(0:23,leboz[i,],col="BLUE")
dev.off()
```

H.4.17. *Code for Figure 17.*

```
indx<-which(apply(leboz,1,function(x) length(which(is.na(x))))<5)
lebsvd<-imputeMat(leboz[indx,],fitSVDCenter,pars=3,niter=500)
pdf("lebecSVDave.pdf")
plot(0:23,lebsvd$extra[[1]],type="l",col="BLUE",xlab="Hour",ylab="Ozone ppm")
dev.off()
pdf("lebecSVDpc1.pdf")
plot(0:23,lebsvd$extra[[2]]$v[,1],type="l",col="BLUE",xlab="Hour",ylab="Ozone
    ppm")
dev.off()
pdf("lebecSVDpc2.pdf")
plot(0:23,lebsvd$extra[[2]]$v[,2],type="l",col="BLUE",xlab="Hour",ylab="Ozone
    ppm")
dev.off()
pdf("lebecSVDpc3.pdf")
```

```
13 plot(0:23,lebsvd$extra[[2]]$v[,3],type="l",col="BLUE",xlab="Hour",ylab="Ozone
       ppm")
14 dev.off()
```

H.4.18. *Code for Figure 18.*

```
1  indx<-which(apply(leboz,1,function(x) length(which(is.na(x))))<5)
2  lebsvd<-imputeMat(leboz[indx,],fitSVDCenter,pars=3,niter=500)
3  cc<-rainbow(25)
4  pdf("lebecSVDPerturbpc1.pdf")
5  p<-lebsvd$extra[[1]]
6  q<-lebsvd$extra[[2]]$v[,1]
7  plot(0:23,p/sqrt(sum(p^2)),type="l",lwd=5,ylim=c(-.05,.35),xlab="Hour",ylab="
       Principal Curve 1")
8  for (i in seq(-24,24,by=2)) {
9  r<-p+.005*i*q
10 lines(0:23,r/sqrt(sum(r^2)),col=cc[i])
11 }
12 dev.off()
13 pdf("lebecSVDPerturbpc2.pdf")
14 p<-lebsvd$extra[[1]]
15 q<-lebsvd$extra[[2]]$v[,2]
16 plot(0:23,p/sqrt(sum(p^2)),type="l",lwd=5,ylim=c(-.05,.35),xlab="Hour",ylab="
       Principal Curve 2")
17 for (i in seq(-24,24,by=2)) {
18 r<-p+.005*i*q
19 lines(0:23,r/sqrt(sum(r^2)),col=cc[i])
20 }
21 dev.off()
22 pdf("lebecSVDPerturbpc3.pdf")
23 p<-lebsvd$extra[[1]]
24 q<-lebsvd$extra[[2]]$v[,3]
25 plot(0:23,p/sqrt(sum(p^2)),type="l",lwd=5,ylim=c(-.05,.35),xlab="Hour",ylab="
       Principal Curve 3")
26 for (i in seq(-24,24,by=2)) {
27 r<-p+.005*i*q
28 lines(0:23,r/sqrt(sum(r^2)),col=cc[i])
29 }
30 dev.off()
```

H.4.19. *Code for Figure 19.*

```
indx<-which(apply(leboz,1,function(x) length(which(is.na(x))))<5)
lebsvd<-imputeMat(leboz[indx,],fitSVDCenter,pars=3,niter=500)
pdf("lebec_fitted.pdf")
plot(0:23,seq(0,.10,length=24),type="n",xlab="Hour",ylab="Ozone_ppm")
for (i in 1:length(indx)) lines(0:23,lebsvd$fitted[i,],col="BLUE")
dev.off()
```

H.4.20. *Code for Figure 20.*

```
indx<-which(apply(leboz,1,function(x) length(which(is.na(x))))<5)
lebsvd<-imputeMat(leboz[indx,],fitSVDCenter,pars=3,niter=500)
pdf("lebecSVDloads1.pdf")
plot(as.Date(rownames(leboz[indx,]),"%m-%d-%y"),lebsvd$extra[[2]]$u[,1],ylab=
    "Loadings_Principal_Curve_1")
lines(lowess(as.Date(rownames(leboz[indx,]),"%m-%d-%y"),lebsvd$extra[[2]]$u
    [,1]),col="RED",lwd=3)
dev.off()
pdf("lebecSVDloads2.pdf")
plot(as.Date(rownames(leboz[indx,]),"%m-%d-%y"),lebsvd$extra[[2]]$u[,2],ylab=
    "Loadings_Principal_Curve_2")
lines(lowess(as.Date(rownames(leboz[indx,]),"%m-%d-%y"),lebsvd$extra[[2]]$u
    [,2]),col="RED",lwd=3)
dev.off()
pdf("lebecSVDloads3.pdf")
plot(as.Date(rownames(leboz[indx,]),"%m-%d-%y"),lebsvd$extra[[2]]$u[,3],ylab=
    "Loadings_Principal_Curve_3")
lines(lowess(as.Date(rownames(leboz[indx,]),"%m-%d-%y"),lebsvd$extra[[2]]$u
    [,3]),col="RED",lwd=3)
dev.off()
```

H.4.21. *Code for Figure 21.*

```
indx<-which(apply(leboz,1,function(x) length(which(is.na(x))))<5)
lebsvd<-imputeMat(leboz[indx,],fitSVDCenter,pars=3,niter=500)
pdf("lebecSVDphase1.pdf")
plot(lebsvd$extra[[1]],lebsvd$extra[[2]]$v[,1],type="l",xlab="Average",ylab="
    Principal_Curve_1",col="RED")
text(lebsvd$extra[[1]],lebsvd$extra[[2]]$v[,1],as.character(0:23))
dev.off()
pdf("lebecSVDphase2.pdf")
plot(lebsvd$extra[[1]],lebsvd$extra[[2]]$v[,2],type="l",xlab="Average",ylab="
    Principal_Curve_2",col="RED")
text(lebsvd$extra[[1]],lebsvd$extra[[2]]$v[,2],as.character(0:23))
```

```
10 dev.off()
11 pdf("lebecSVDphase3.pdf")
12 plot(lebsvd$extra[[1]],lebsvd$extra[[2]]$v[,3],type="l",xlab="Average",ylab="
       Principal Curve 3",col="RED")
13 text(lebsvd$extra[[1]],lebsvd$extra[[2]]$v[,3],as.character(0:23))
14 dev.off()
```

H.5. **Code for Tables.**

Department of Statistics, University of California, Los Angeles, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu