Chapter 7
Exploring Variation: Functional Principal and Canonical Components Analysis

Now we look at how observations vary from one replication or sampled value to the next. There is, of course, also variation *within* observations, but we focused on that type of variation when considering data smoothing in Chapter 5.

Principal components analysis, or PCA, is often the first method that we turn to after descriptive statistics and plots. We want to see what primary modes of variation are in the data, and how many of them seem to be substantial. As in multivariate statistics, *eigenvalues* of the bivariate *variance-covariance function* $v(s,t)$ are indicators of the importance of these principal components, and plotting eigenvalues is a method for determining how many principal components are required to produce a reasonable summary of the data.

In functional PCA, there is an *eigenfunction* associated with each eigenvalue, rather than an eigenvector. These eigenfunctions describe major variational components. Applying a rotation to them often results in a more interpretable picture of the dominant modes of variation in the functional data, without changing the total amount of common variation.

We take some time over PCA partly because this may be the most common functional data analysis and because the tasks that we face in PCA and our approaches to them will also be found in more model-oriented tools such as functional regression analysis. For example, we will see that each eigenfunction can be constrained to be smooth by the use of roughness penalties, just as in the data smoothing process. Should we use rough functions to capture every last bit of interesting variation in the data and then force the eigenfunctions to be smooth, or should we carefully smooth the data first before doing PCA?

A companion problem is the analysis of the *covariation* between two different functional variables based on samples taken from the same set of cases or individuals. For example, what types of variation over weather stations do temperature and log precipitation share? How do knee and hip angles covary over the gait cycle? Canonical correlation analysis (CCA) is the method of choice here. We will see many similarities between PCA and CCA.

J.O. Ramsay et al., *Functional Data Analysis with R and MATLAB*, Use R,
DOI: 10.1007/978-0-387-98185-7_7,

7.1 An Overview of Functional PCA

In multivariate statistics, variation is usually summarized by either the covariance matrix or the correlation matrix. Because the variables in a multivariate observation can vary a great deal in location and scale due to relatively arbitrary choices of origin and unit of measurement, and because location/scale variation tends to be uninteresting, multivariate analyses are usually based on the correlation matrix. But when an observation is functional, values $x_i(s)$ and $x_i(t)$ have the same origin and scale. Consequently, either the estimated *covariance function*

$$v(s,t) = (N-1)^{-1} \sum_i [x_i(s) - \bar{x}(s)][x_i(t) - \bar{x}(t)],$$

or the *cross-product function*

$$c(s,t) = N^{-1} \sum_i x_i(s) x_i(t),$$

will tend to be more useful than the *correlation function*

$$r(s,t) = \frac{v(s,t)}{\sqrt{[v(s,s)v(t,t)]}}.$$

Principal components analysis may be defined in many ways, but its motivation is perhaps clearer if we define PCA as the search for a probe ξ, of the kind that we defined in Chapter 6, that reveals the most important type of variation in the data. That is, we ask, "For what weight function ξ would the probe scores

$$\rho_\xi(x_i) = \int \xi(t) x_i(t) \mathrm{d}t$$

have the largest possible variation?" In order for the question to make sense, we have to impose a size restriction on ξ, and it is mathematically natural to require that $\int \xi^2(t)\mathrm{d}t = 1$.

Of course, the mean curve by definition is a mode of variation that tends to be shared by most curves, and we already know how to estimate this. Consequently, we usually remove the mean first and then probe the functional residuals $x_i - \bar{x}$. Later, when we look at various types of functional regression, we may also want to first remove other known sources of variation that are explainable by multivariate and/or functional covariates.

The probe score variance $\mathtt{Var}[\int \xi(t)(x_i(t) - \bar{x}(t))^2 \mathrm{d}t]$ associated with a probe weight ξ is the value of

$$\mu = \max_\xi \{\sum_i \rho_\xi^2(x_i)\} \text{ subject to } \int \xi^2(t)\mathrm{d}t = 1. \tag{7.1}$$

In standard terminology, μ and ξ are referred to as the largest *eigenvalue* and *eigenfunction*, respectively, of the estimated variance-covariance function v. An alternative to the slightly intimidating term "eigenfunction" is *harmonic*.

As in multivariate PCA, a nonincreasing sequence of eigenvalues $\mu_1 \geq \mu_2 \geq \ldots \mu_k$ can be constructed stepwise by requiring each new eigenfunction, computed in step ℓ, to be orthogonal to those computed on previous steps,

$$\int \xi_j(t)\xi_\ell(t)\mathrm{d}t = 0,\ j = 1,\ldots,\ell-1 \ \text{ and } \ \int \xi_\ell^2(t)\mathrm{d}t = 1. \tag{7.2}$$

In multivariate settings the entire suite of eigenvalue/eigenvector pairs would be computed by the *eigenanalysis* of the covariance matrix $\mathbf{V}$, solving the matrix eigenequation $\mathbf{V}\xi_j = \mu_j\xi_j$. The approach is essentially the same for functional data; that is, we calculate eigenfunctions ξ_j of the bivariate covariance function $v(s,t)$ as solutions of the functional eigenequation

$$\int v(s,t)\xi_j(t)\mathrm{d}t = \mu_j\xi_j(s). \tag{7.3}$$

We see here as well as elsewhere that going from multivariate to functional data analysis is often only a matter of replacing summation over integer indices by integration over continuous indices such as t. Although the computation details are not at all the same, this is thankfully hidden by the notation and dealt with in the `fda` package.

However, there is an important difference between multivariate and functional PCA caused by the fact that, whereas in multivariate data the number of variables p is usually less than the number of observations N, for functional data the number of observed function values n is usually greater than N. This implies that the maximum number of nonzero eigenvalues in the functional context is $\min\{N-1, K, n\}$, and in most applications will be $N-1$.

Suppose, then, that our software can present us with, say, $N-1$ positive eigenvalue/eigenfunction pairs (μ_j, ξ_j). What do we do next? For each choice of ℓ, $1 \leq \ell \leq N-1$, the ℓ leading eigenfunctions or harmonics define a basis system that can be used to approximate the sample functions x_i. These basis functions are orthogonal to each other and are normalized in the sense that $\int \xi_\ell^2 = 1$. They are therefore referred to as an *orthonormal* basis. They are also the most efficient basis possible of size ℓ in the sense that the total error sum of squares

$$\mathtt{PCASSE} = \sum_i^N \int [x_i(t) - \bar{x}(t) - \mathbf{c}_i'\xi(t)]^2\mathrm{d}t \tag{7.4}$$

is the minimum achievable with only ℓ basis functions. Of course, other ℓ-dimensional systems certainly exist that will do as well, and we will consider some shortly, but none will do better. In the physical sciences, these optimal basis functions ξ_j are often referred to as *empirical orthogonal functions*.

It turns out that there is a simple relationship between the optimal total squared error and the eigenvalues that are discarded, namely that

$$\texttt{PCASSE} = \sum_{j=\ell+1}^{N-1} \mu_j.$$

It is usual, therefore, to base a decision on the number ℓ of harmonics to use on a visual inspection of a plot of the eigenvalues μ_j against their indices j, a display that is often referred to in the social science literature as a *scree plot*. Although there are a number of proposals for automatic data-based rules for deciding the value of ℓ, many nonstatistical considerations can also affect this choice.

The coefficient vectors $\mathbf{c}_i, i = 1,\ldots,N$ contain the coefficients c_{ij} that define the optimal fit to each function x_i, and are referred to as *principal component scores*. They are given by the following:

$$c_{ij} = \rho_{\xi_j}(x_i - \bar{x}) = \int \xi_j(t)[x_i(t) - \bar{x}(t)]\mathrm{d}t. \tag{7.5}$$

As we will show below, they can be quite helpful in interpreting the nature of the variation identified by the PCA. It is also common practice to treat these scores as "data" to be subjected to a more conventional multivariate analysis.

We suggested that the eigenfunction basis was optimal but not unique. In fact, for any nonsingular square matrix $\mathbf{L}$ of order ℓ, the system $\phi = \mathbf{T}\xi$ is also optimal and spans exactly the same functional subspace as that spanned by the eigenfunctions. Moreover, if $\mathbf{T}' = \mathbf{T}^{-1}$, such matrices being often referred to as *rotation matrices*, the new system ϕ is also orthonormal. There is, in short, no mystical significance to the eigenfunctions that PCA generates, a simple fact that is often overlooked in textbooks on multivariate statistics. Well, okay, perhaps $\ell = 1$ is an exception. In fact, it tends to happen that only the leading eigenfunction has an obvious meaningful interpretation in terms of processes known to generate the data.

But for $\ell > 1$, there is nothing to prevent us from searching among the infinite number of alternative systems $\phi = \mathbf{T}\xi$ to find one where all of the orthonormal basis functions ϕ_j are seen to have some substantive interpretation. In the social sciences, where this practice is routine, a number of criteria for optimizing the chances of interpretability have been devised for choosing a rotation matrix $\mathbf{T}$, and we will demonstrate the usefulness of the popular *VARIMAX* criterion in our examples.

Readers are referred at this point to standard texts on multivariate data analysis or to the more specialized treatment in Jolliffe (2002) for further information on principal components analysis. Most of the material in these sources applies to this functional context.

7.2 PCA with Function `pca.fd`

Principal component analysis is implemented in the functions *`pca.fd`* and `pca_fd` in R and Matlab, respectively. The call in R is

```
pca.fd(fdobj, nharm = 2, harmfdPar=fdPar(fdobj),
```

```
          centerfns = TRUE)
```

The first argument is a functional data object containing the functional data to be analyzed, and the second specifies the number ℓ of principal components to be retained. The third argument is a functional parameter object that provides the information necessary to smooth the eigenfunctions if necessary; we will postpone this topic to Section 7.3. Finally, although most principal components analyses are applied to data with the mean function subtracted from each function, the final argument permits this to be suppressed.

Function `pca.fd` in R returns an object with the class name `pca.fd`, so that it is effectively a constructor function. Here are the named components for this class:

`harmonics` A functional data object for the ℓ harmonics or eigenfunctions ξ_j.
`values` The complete set of eigenvalues μ_j.
`scores` The matrix of scores c_{ij} on the principal components or harmonics.
`varprop` A vector giving the proportion $\mu_j / \sum \mu_j$ of variance explained by each eigenfunction.
`meanfd` A functional data object giving the mean function $\bar{x}$.

7.2.1 PCA of the Log Precipitation Data

Here is the command to do a PCA using only two principal components for the log precipitation data and to display the eigenvalues.

```
logprec.pcalist = pca.fd(logprecfd, 2)
print(logprec.pcalist$values)
```

We observe that these two harmonics account for 96% of the variation around the mean log precipitation curve; the first four eigenvalues are 39.5, 3.9, 1.0 and 0.4, respectively.

The two principal components are plotted by the command

```
plot.pca.fd(logprec.pcalist)
```

Figure 7.1 shows the two principal component functions by displaying the mean curve along +'s and −'s indicating the consequences of adding and subtracting a small amount of each principal component. We do this because a principal component represents *variation* around the mean, and therefore is naturally plotted as such. We see that the first harmonic, accounting for 88% of the variation, represents a relative constant vertical shift in the mean, and that the second shows essentially a contrast between winter and summer precipitation levels.

It is in fact usual for unrotated functional principal components to display the same sequence of variation no matter what is being analyzed. The first will be a constant shift, the second a linear contrast between the first and second half with a single crossing of zero, the third a quadratic pattern, and so on. That is, we tend to see the sequence of orthogonal polynomials. However, for periodic data, where only periodic harmonics are possible, the linear contrast is suppressed.

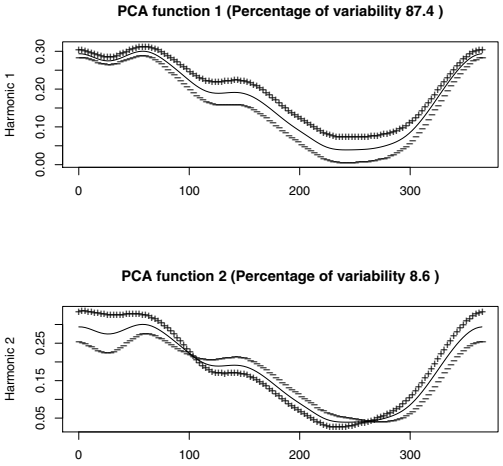

Fig. 7.1 The two principal component functions or harmonics are shown as perturbations of the mean, which is the solid line. The +'s show what happens when a small amount of a principal component is added to the mean, and the −'s show the effect of subtracting the component.

The fact that unrotated functional principal components are so predictable emphasizes the need for looking for a rotation of them that can reveal more meaningful components of variation. The VARIMAX rotation algorithm is often used for this purpose. The following command applies this rotation and then plots the result:

```
logprec.rotpcalist = varmx.pca.fd(logprec.pcalist)
plot.pca.fd(logprec.rotpcalist)
```

The results are plotted in Figure 7.2. The first component portrays variation that is strongest in midwinter and the second captures primarily summer variation.

It can be profitable to plot the principal component scores for pairs of harmonics to see how curves cluster and otherwise distribute themselves within the K-dimensional subspace spanned by the eigenfunctions. Figure 7.3 reveals some fascinating structure. Most of the stations are contained within two clusters: the upper right with the Atlantic and central Canada stations and the lower left with the prairie and mid-Arctic stations. The outliers are the three west coast stations and Resolute in the high Arctic. Often, functional data analyses will turn into a multivariate data analysis at this point by using the component scores as "data matrices" in more conventional analyses.

It may be revealing to apply PCA to some order of derivative rather than to the curves themselves, because underlying processes may reveal their effects at the change level rather than at the level of what we measure. This is certainly true of growth curve data, where hormonal processes and other growth activators change

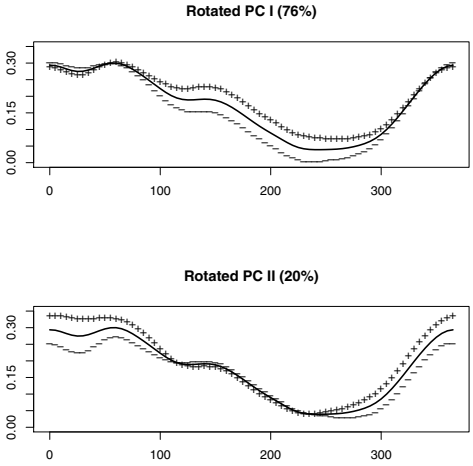

Fig. 7.2 The two rotated principal component functions are shown as perturbations of the mean, which is the solid line. The top panel contains the strongest component, with variation primarily in the midwinter. The bottom panel shows primarily summer variation.

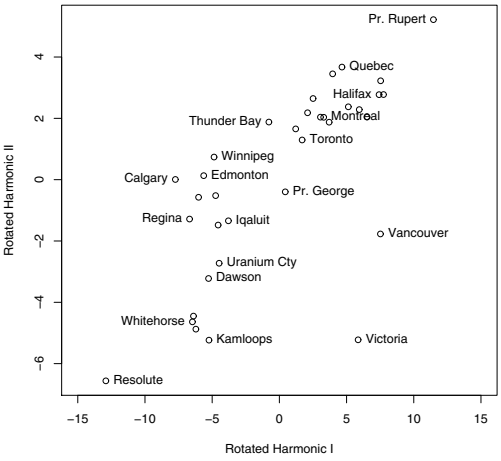

Fig. 7.3 The scores for the two rotated principal component functions are shown as circles. Selected stations are labeled in order to identify the two central clusters and the outlying stations.

the rate of change of height and can be especially evident at the level of the acceleration curves that we plotted in Section 1.1.

7.2.2 PCA of Log Precipitation Residuals

We can now return to exploring the residuals from the smooths of the log precipitation curves in Chapter 5. First, we set up function versions of the residuals and plot them:

```
logprecres.fd = smooth.basis(day.5, logprecres,
                             fdParobj)$fd
plot(logprecres.fd, lwd=2, col=1, lty=1, cex=1.2,
     xlim=c(0,365), ylim=c(-0.07, 0.07),
     xlab="Day", ylab="Residual (log 10 mm)")
```

These are shown in Figure 7.4. There we see that, while most of these residual functions show fairly chaotic variation, three stations have large oscillations in summer and autumn. The result of estimating a single principal component is shown in Figure 7.5, where we see the mean residual along with the effect of adding and subtracting this first component. The mean residual itself shows the oscillation that we have noted. The principal component accounts for about 49% of the residual variance about this mean. It defines variation around the mean oscillation located in these months. Three stations have much larger scores on this component: They are Kamloops, Victoria and Vancouver, all in southern British Columbia. It seems that rainfall events come in cycles in this part of Canada at this time of the year, and there is interesting structure to be uncovered in these residuals.

7.3 More Functional PCA Features

In multivariate PCA, we control the level of fit to the data by selecting the number of principal components. In functional PCA, we can also modulate fit by controlling the roughness of the estimated eigenfunctions. We do this by modifying the definition of orthogonality. If, for example, we want to penalize excessive curvature in principal components, we can use this generalized form of orthogonality:

$$\int \xi_j(t)\xi_k(t)dt + \lambda \int D^2\xi_j(t)D^2\xi_k(t)dt = 0, \tag{7.6}$$

where λ controls the relative emphasis on orthogonality of second derivatives in much the same way as it does in roughness–controlled smoothing. This gives us a powerful new form of leverage in defining a decomposition of variation.

Roughness-penalized PCA also relates to a fundamental aspect of variation in function spaces. Functions can be large in two distinct ways: first and most obvi-

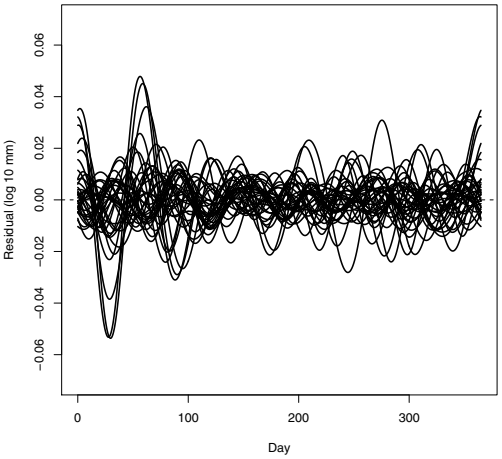

Fig. 7.4 The smoothed residual functions for the log precipitation data.

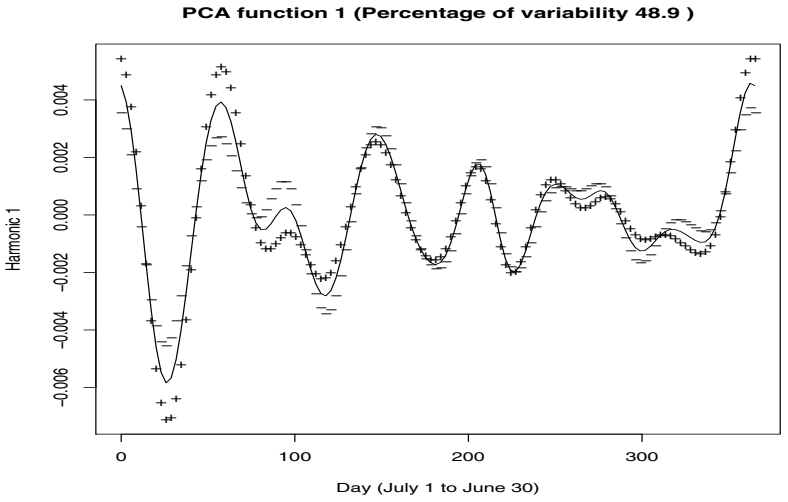

Fig. 7.5 The first principal component for the log precipitation residual functions, shown by adding (+) and subtracting (-) the component from the mean function (solid line).

ously in terms of their amplitude, and second in terms of their complexity or amount of high-frequency variation. This second feature is closely related to how rapidly a Fourier series expansion of a function converges, and is therefore simply another aspect of how PCA itself works. This second type of size of principal components is what λ controls. Ramsay and Silverman (2005) show how λ in PCA can be data-defined via cross-validation.

7.4 PCA of Joint X-Y Variation in Handwriting

Of course, functions themselves may be multivariate. When we apply PCA to the data shown in Section 1.2 on the writing of the script "fda," we have to do a simultaneous PCA of the X and Y coordinates. The corresponding eigenfunctions will also be multivariate, but each eigenfunction is still associated with a single eigenvalue μ_j. This means that multivariate PCA is not the same thing as separate PCA's applied to each coordinate in turn. The multivariate PCA problem, therefore, blends together the aspects of multivariate and functional data analyses.

At the level of code, however, multivariate PCA is achieved seamlessly by function `pca.fd`. These R commands define a small but sufficient number of basis functions for representing the "fda" handwriting data as a bivariate functional data object, smooth the data, and install appropriate labels for the dimensions.

```
fdarange = c(0, 2300)
fdabasis = create.bspline.basis(fdarange, 105, 6)
fdatime  = seq(0, 2300, len=1401)
fdafd =
  smooth.basis(fdatime, handwrit, fdabasis)$fd
fdafd$fdnames[[1]] = "Milliseconds"
fdafd$fdnames[[2]] = "Replications"
fdafd$fdnames[[3]] = list("X", "Y")
```

These R commands carry out the PCA of the bivariate functional data object `fdafd` using three harmonics, plot the unrotated eigenfunctions, perform a VARIMAX rotation of these eigenfunctions, and replot the results.

```
nharm = 3
fdapcaList = pca.fd(fdafd, nharm)
plot.pca.fd(fdapcaList)
fdarotpcaList = varmx.pca.fd(fdapcaList)
plot.pca.fd(fdarotpcaList)
```

How did we settle on three for the number of harmonics? We have found that the logarithm of eigenvalues tend to decrease linearly after an initial few that are large. The following commands plot the log eigenvalues up to $j = 12$ with the least-squares linear trend in the eigenvalue with indices 4 to 12.

```
fdaeig = fdapcaList$values
```

```
neig   = 12
x      = matrix(1,neig-nharm,2)
x[,2]  = (nharm+1):neig
y      = log10(fdaeig[(nharm+1):neig])
c      = lsfit(x,y,int=FALSE)$coef
par(mfrow=c(1,1),cex=1.2)
plot(1:neig, log10(fdaeig[1:neig]), "b",
     xlab="Eigenvalue Number",
     ylab="Log10 Eigenvalue")
lines(1:neig, c[1]+ c[2]*(1:neig), lty=2)
```

The result is Figure 7.6. The first three log eigenvalues seem well above the linear trend in the next nine, suggesting that the leading three harmonics are important. Together they account for 62% of the variation in the scripts.

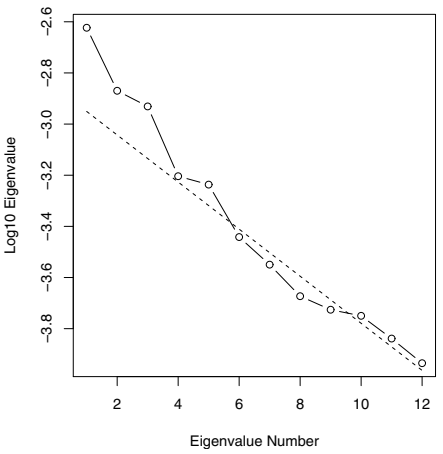

Fig. 7.6 The logarithms (base 10) of the first 12 eigenvalues in the principal components analysis of the "fda" handwriting data. The dashed line indicates the linear trend in the last nine in the sequence.

Figure 7.7 plots two of the VARIMAX–rotated eigenfunctions as perturbations of the mean script. The rotated harmonic on the left mostly captures variation in the lower loop of "f", and the harmonic on the right displays primarily variation in its upper loop. This suggests that variabilities in these two loops are independent of each other.

We can also analyze situations where there are both functional and multivariate data available, such as handwritings from many subjects along with measurements

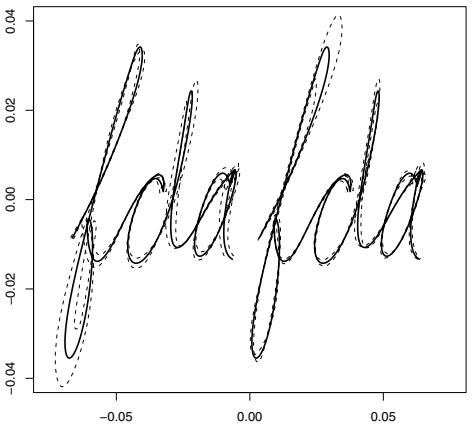

Fig. 7.7 Two of the rotated harmonics are plotted as a perturbations of the mean "fda" script, shown as a heavy solid line.

of subject characteristics such as age, ethnicity, etc. See Ramsay and Silverman (2005) for further details.

7.5 Exploring Functional Covariation with Canonical Correlation Analysis

We often want to examine the ways in which two sets of curves $(x_i, y_i), i = 1, \ldots, N$, share variation. How much variation, for example, is shared between temperature and log precipitation over the 35 Canadian weather stations? This question is related to the issue of how well one can predict one from another, which we will take up in the next chapter. Here, we consider a symmetric view on the matter that does not privilege either variable. We offer here only a quick summary of the mathematical aspects of canonical correlation analysis, and refer the reader to Ramsay and Silverman (2005) for a more detailed account.

To keep the notation tidy, we will assume that the two sets of variables have been *centered*, that is, x_i and y_i have been replaced by the residuals $x_i - \bar{x}$ and $y_i - \bar{y}$, respectively, if this was considered appropriate. That is, we assume that $\bar{x} = \bar{y} = 0$. As before, we define modes of variation for the x_i's and the y_i's in terms of the pair of *probe weight functions* ξ and η that define the integrals

$$\rho_{\xi i} = \int \xi(t) x_i(t) \mathrm{d}t \quad \text{and} \quad \rho_{\eta i} = \int \eta(t) y_i(t) \mathrm{d}t, \tag{7.7}$$

respectively. The N pairs of *probe scores* $(\rho_{\xi i}, \rho_{\eta i})$ defined in this way represent shared variation if they correlate strongly with one another.

The *canonical correlation criterion* is the squared correlation

$$R^2(\xi,\eta) = \frac{[\sum_i \rho_{\xi i}\rho_{\eta i}]2}{[\sum_i \rho_{\xi i}^2][\sum_i \rho_{\eta i}^2]} = \frac{[\sum_i (\int \xi(t)x_i(t)\mathrm{d}t)(\int \eta(t)y_i(t)\mathrm{d}t)]^2}{[\sum_i (\int \xi(t)x_i(t)\mathrm{d}t)^2][\sum_i (\int \eta(t)y_i(t)\mathrm{d}t)^2]}. \quad (7.8)$$

As in PCA, the probe weights ξ and η are then specified by finding that weight pair that optimizes the criterion $R^2(\xi,\eta)$. But, again as in PCA, we can compute a nonincreasing series of squared canonical correlations $R_1^2, R_2^2, \ldots, R_k^2$ by constraining successive canonical probe values to be orthogonal. The length k of the sequence is the smallest of the sample size N, the number of basis functions for either functional variable, or the number of basis functions used for ξ and η.

That we are now optimizing with respect to two probes at the same time makes canonical correlation analysis an exceedingly *greedy* procedure, where this term borrowed from data mining implies that CCA can capitalize on the tiniest variation in either set of functions in maximizing this ratio to the extent that, unless we exert some control over the process, it can be hard to see anything of interest in the result. It is in practice essential to enforce strong smoothness on the two weight functions ξ and η to limit this greediness. This can be done by either selecting a low-dimensional basis for each or by using an explicit roughness penalty in much the same manner as is possible for functional PCA.

Let us see how this plays out in the exploration of covariation between daily temperature and log precipitation, being careful to avoid the greediness pitfall by placing very heavy penalties on roughness of the canonical weight functions as measured by the size of their second derivatives. Here are the commands in R that function `cca.fd` to do the job:

```
ccafdPar = fdPar(daybasis, 2, 5e6)
ncon     = 3
ccalist  = cca.fd(temp.fd, logprec.fd, ncon,
                  ccafdPar, ccafdPar)
```

The third argument of `cca.fd` specifies the number of canonical weight/variable pairs that we want to examine, which, in this case, is the complete sequence. The final two arguments specify the bases for the expansion of ξ and η, respectively, as well as their roughness penalties.

The canonical weight functional data objects and the corresponding three squared canonical correlations are extracted from the list object `ccalist` produced by function `cca.fd` as follows:

```
ccawt.temp    = ccalist$ccawtfd1
ccawt.logprec = ccalist$ccawtfd2
corrs         = ccalist$ccacorr
```

The squared correlations are 0.92, 0.62 and 0.35; so that there is a dominant pair of modes of variation that correlates at a high level, and then two subsequent pairs with modest but perhaps interesting correlations.

Consider first the type of variation associated with the first canonical correlation. Figure 7.8 displays the corresponding two canonical weight functions. The temperature canonical weight function ξ_1 resembles a sinusoid with period 365/2 and having zeros in July, October, January and April. But the log precipitation counterpart η_1 is close to a sinusoid with period 365 and zeros in July and January.

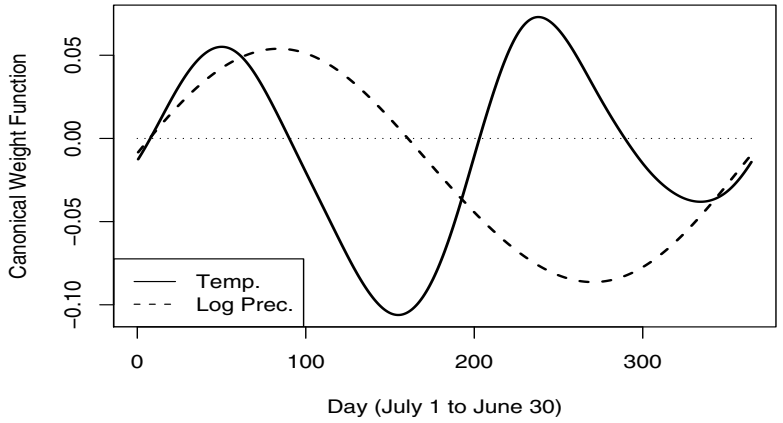

Fig. 7.8 The first pair of canonical weight functions or probes (ξ, η) correlating temperature and log precipitation for the Canadian weather data.

Regarding each weight function as contrasting corresponding variable values, the temperature curve seems primarily to contrast spring and autumn temperatures with winter temperatures; while the corresponding log precipitation contrast is between rainfall in the spring and autumn. A station will score high on both canonical variables if it is cool in winter relative to its temperatures in spring and autumn, and at the same time has more precipitation in the spring than in the fall.

The scores of each weather station on each set of canonical variables are extracted by

```
ccascr.temp    = ccalist$ccavar1
ccascr.logprec = ccalist$ccavar2
```

Figure7.9 plots the scores for the first log precipitation canonical variable scores against their temperature counterparts for selected weather stations. We see a near-perfect ordering with respect to latitude, although favoring eastern stations over western stations at the same latitudes so that Vancouver and Victoria wind up at the bottom left. Certainly Resolute's temperatures are cold in winter, and what precipitation it gets comes more in the spring than at another time, so that it earns it's place in the upper right of the plot. The marine weather stations, Prince Rupert and

St. John's, on the other hand, are actually relatively warm in the winter and get more precipitation in the fall than in the winter, and therefore anchor the lower left of the plot. Note, though, that the linear order in Figure7.9 misses Kamloops by a noticeable amount. The position of this interior British Columbia city deep in a valley, where relatively little rain or snow falls at any time of the year, causes it to be anomalous in many types of analysis.

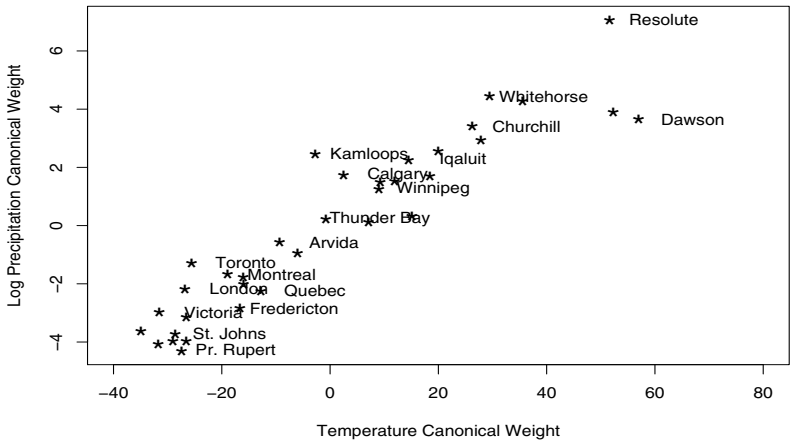

Fig. 7.9 The scores for the first pair of canonical variables plotted against each other, with labels for selected weather stations.

7.6 Details for the `pca.fd` and `cca.fd` Functions

7.6.1 The `pca.fd` Function

We give here the arguments of the constructor function `pca.fd` that carries out a functional principal components analysis and constructs an object of the `pca.fd` class. The complete calling sequence is

```
pca.fd(fdobj, nharm = 2, harmfdPar=fdPar(fdobj),
       centerfns = TRUE)
```

The arguments are as follows:

`fdobj` A functional data object.
`nharm` The number of harmonics or principal components to compute.

`harmfdPar` A functional parameter object that defines the harmonic or principal component functions to be estimated.
`centerfns` A logical value: if TRUE, subtract the mean function from each function before computing principal components.

Function `pca.fd` returns an argument of the `pca.fd` class, which is a named list with the following components:

`harmonics` A functional data object for the harmonics or eigenfunctions.
`values` The complete set of eigenvalues.
`scores` A matrix of scores on the principal components or harmonics.
`varprop` A vector giving the proportion of variance explained by each eigenfunction.
`meanfd` A functional data object giving the mean function.

7.6.2 The `cca.fd` Function

The calling sequence for `cca.fd` is

```
cca.fd(fdobj1, fdobj2=fdobj1, ncan = 2,
       ccafdParobj1=fdPar(basisobj1, 2, 1e-10),
       ccafdParobj2=ccafdParobj1, centerfns=TRUE)
```

The arguments are as follows:

`fdobj1` A functional data object.
`fdobj2` A functional data object. By default this is `fdobj1`, in which case the first argument must be a bivariate functional data object.
`ncan` The number of canonical variables and weight functions to be computed. The default is 2.
`ccafdParobj1` A functional parameter object defining the first set of canonical weight functions. The object may contain specifications for a roughness penalty. The default is defined using the same basis as that used for `fdobj1` with a slight penalty on its second derivative.
`ccafdParobj2` A functional parameter object defining the second set of canonical weight functions. The object may contain specifications for a roughness penalty. The default is `ccafdParobj1`.
`centerfns` If `TRUE`, the functions are centered prior to analysis. This is the default.

7.7 Some Things to Try

1. **Medfly Data**: The medfly data have been a popular dataset for functional data analysis and are included in the fda package. The medfly data consist of records

of the number of eggs laid by 50 fruit flies on each of 31 days, along with each individual's total lifespan.

a. Smooth the data for the number of eggs, choosing the smoothing parameter by generalized cross-validation (GCV). Plot the smooths.
b. Conduct a principal components analysis using these smooths. Are the components interpretable? How many do you need to retain to recover 90% of the variation. If you believe that smoothing the PCA will help, do so.
c. Try a linear regression of lifespan on the principal component scores from your analysis. What is the R^2 for this model? Does `lm` find that the model is significant? Reconstruct and plot the coefficient function for this model along with confidence intervals. How does it compare to the model obtained through functional linear regression?

2. Apply principal components analysis to the functional data object `Wfd` returned by the monotone smoothing function `smooth.monotone` applied to the growth data. These functions are the logs of the first derivatives of the growth curves. What is the impact of the variation in the age of the purbertal growth spurt on these components?

7.8 More to Read

Functional principal components analysis predates the emergence of functional data analysis, especially in fields in engineering and sciences that work with functional data routinely, such as climatology. Principal components are often referred to in these fields as *empirical basis functions*, a phrase that is exactly the right thing since functional principal components are both orthogonal and can also serve well as a customized low-dimensional basis system for representing the actual functions.

There are many currently active and unexplored areas of research into functional PCA. James et al. (2000) consider situations where curves are observed in fragments, so that the interval of observation varies from record to record. James and Sugar (2003) look at the same data situation in the context of cluster analysis, another multivariate exploratory tool that is now associated with a large functional literature. Readers with a background in psychometrics will wonder about a functional version of factor analysis, whether exploratory or confirmatory; and functional versions of structural equation models are well down the road, but no doubt perfectly feasible.