# Functional Data Analysis

## A Short Course

Giles Hooker

11/10/2017

# Table of Contents

# Some References

Three references for this course (all Springer)

- Ramsay & Silverman, 2005, "Functional Data Analysis"
- Ramsay & Silverman, 2002, "Applied Functional Data Analysis"
- Ramsay, Hooker & Graves, 2009, "Functional Data Analysis in R and Matlab"

Relevant Sofware:

`fda` package in `R`

# Some More References

Other monographs:

- Kokoszka & Reimherr, 2017, "Introduction to Functional Data Analysis"
- Horvath & Kokoszka, 2012, "Inference for Functional Data with Applications"
- Ferraty & Vieux, 2002, "Nonparametric Functional Data Analysis"
- Bosq, 2002, "Linear Processes on Function Spaces"

Other `R` packages

- `fda.usc`: similar to `fda`, with more emphasis on testing and regression.
- `fda.pace`: uses principal components representations in many standard models; based around kernel smoothing.
- `refund`: includes PCA analysis plus some variational Bayes methods, plus some mixed models.

# Assumptions and Expectations

Presentation philosophy:

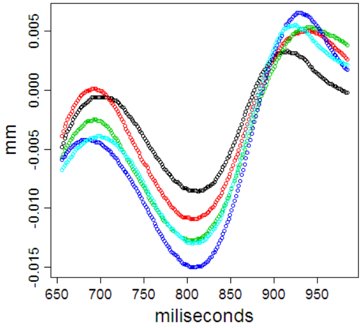
- Geared towards practical/applied use (and extension) of FDA
- Computational tools/methods: "How can we get this done?"
- Focus on particular methods `fda` library in `R`; alternative approaches will be mentioned.
- Some pointers to theory and asymptotics.

Assumed background and interest:

- Applied statistics, including some multivariate analysis.
- Familiarity with `R`
- Smoothing methods/non-parametric statistics covered briefly.
- Assumed interest in using FDA and/or extending FDA methods.

# What is Functional Data?

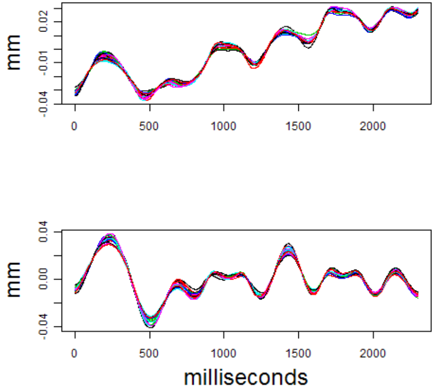
What are the most obvious features of these data?




- quantity
- frequency (resolution)
- similarity
- smoothness

# What Is Functional Data?

**Example:** 20 replications, 1401 observations within replications, 2 dimensions

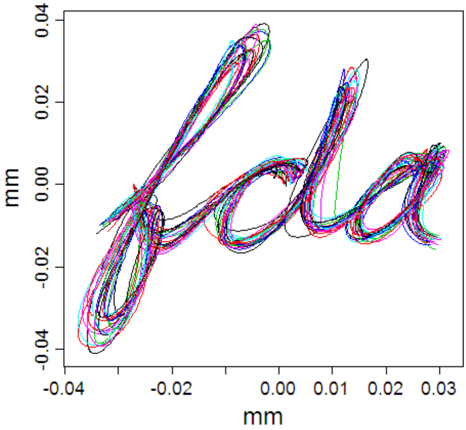



Immediate characteristics:

- High-frequency measurements
- Smooth, but complex, processes
- Repeated observations
- Multiple dimensions
- Let's plot 'y' against 'x'

# Handwriting Data

Measures of position of nib of a pen writing “fda”. 20 replications, measurements taken at 200 hertz.



# What Is Functional Data?

*Functional data is multivariate data with an ordering on the dimensions.* (Müller, (2006))

Key assumption is *smoothness*:

$$y_{ij} = x_i(t_{ij}) + \epsilon_{ij}$$

with $t$ in a continuum (usually time), and $x_i(t)$ smooth

Functional data = the functions $x_i(t)$.

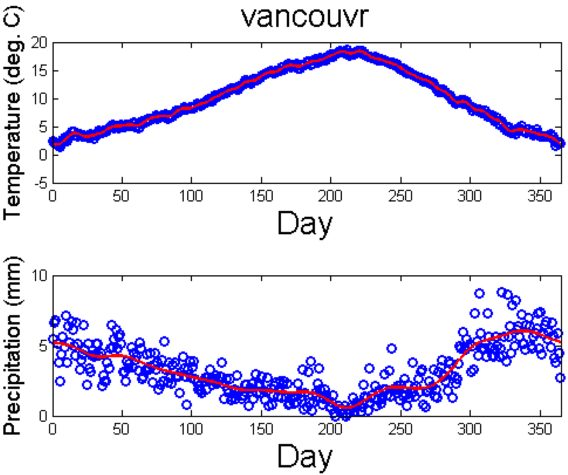
Highest quality data from monitoring equipment

- Optical tracking equipment (eg handwriting data, but also for physiology, motor control,...)
- Electrical measurements (EKG, EEG and others)
- Spectral measurements (astronomy, materials sciences)

But, noisier and less frequent data can also be used.

# Weather In Vancouver

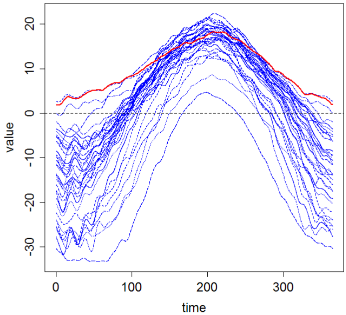
Measure of climate: daily precipitation and temperature in Vancouver, BC averaged over 40 years.



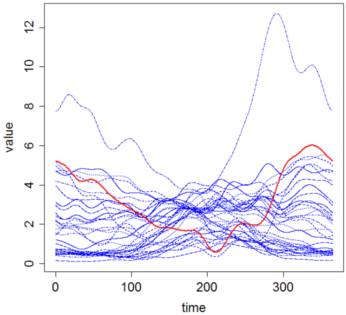
# Canadian Weather Data

Average daily temperature and precipitation records in 35 weather stations across Canada (classical and much over-used)

Temperature



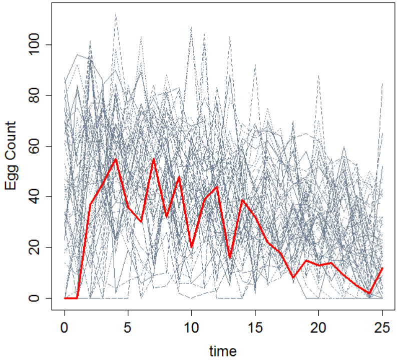

Precipitation




Interest is in variation in and relationships between smooth, underlying processes.

# Medfly Data

Records of number of eggs laid by Mediterranean Fruit Fly (Ceratitis capitata) in each of 25 days (courtesy of H.-G. Müller).




- Total of 50 flies
- Assume eggcount measurements relate to smooth process governing fertility
- Also record total lifespan of each fly.
- Would like to understand how fecundity at each part of lifetime influences lifespan.

# What Are We Interested In?

- Representations of distribution of functions
  - mean
  - variation
  - covariation
- Relationships of functional data to
  - covariates
  - responses
  - other functions
- Relationships between derivatives of functions.
- Timing of events in functions.

# What Are The Challenges?

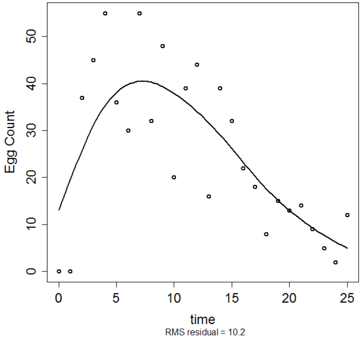
- Estimation of functional data from noisy, discrete observations.
- Numerical representation of infinite-dimensional objects
- Representation of variation in infinite dimensions.
- Description of statistical relationships between infinite dimensional objects.
- $n < p = \infty \Rightarrow$ regularization and smoothness.
- Measures of variation and confidence in estimates.

# Representing Functional Data

# From Discrete to Functional Data

Represent data recorded at discrete times as a continuous function in order to

## Medfly record 1




- Allow evaluation of record at any time point (especially if observation times are not the same across records).
- Evaluate rates of change.
- Reduce noise.
- Allow registration onto a common time-scale.

# From Discrete to Functional Data

Two problems/two methods

1. Representing non-parametric continuous-time functions.
   - Basis-expansion methods:

     $$x(t) = \sum_{i=1}^{K} \phi_i(t) c_i$$

     for pre-defined $\phi_i(t)$ and coefficients $c_i$.
   - Several basis systems available: focus on Fourier and B-splines
2. Reducing noise in measurements
   - Smoothing penalties:

     $$c = \text{argmin} \sum_{i=1}^{n} (y_i - x(t_i))^2 + \lambda \int [Lx(t)]^2 \, dt$$

   - $Lx(t)$ measures "roughness" of $x$
   - $\lambda$ a "smoothing parameter" that trades-off fit to the $y_i$ and roughness; must be chosen.

# 1. Basis Expansions

# Basis Expansions

Consider only one record

$$y_i = x(t_i) + \epsilon_i$$

represent $x(t)$ as

$$x(t) = \sum_{j=1}^{K} c_j \phi_j(t) = \Phi(t)\mathbf{c}$$

We say $\Phi(t)$ is a *basis system* for $x$.

Terms for curvature in linear regression

$$y_i = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 + \beta_3 t_i^3 + \cdots + \epsilon_i$$

implies

$$x(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \cdots$$

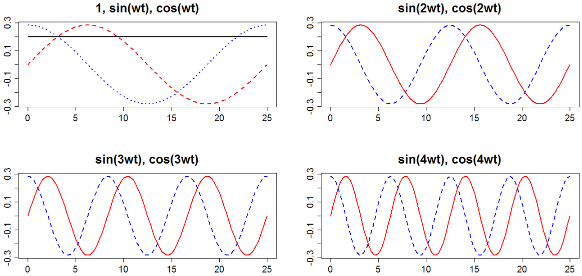
Polynomials are unstable; Fourier bases and B-splines will be more useful.

# The Fourier Basis

- basis functions are sine and cosine functions of increasing frequency:

$$1, sin(\omega t), cos(\omega t), sin(2\omega t), cos(2\omega t), \ldots$$
$$sin(m\omega t), cos(m\omega t), \ldots$$

- constant $\omega = 2\pi/P$ defines the period $P$ of oscillation of the first sine/cosine pair.



# Advantages of Fourier Bases

- Only alternative to polynomials until the middle of the 20th century
- Excellent computational properties, especially if the observations are equally spaced.
- Natural for describing periodic data, such as the annual weather cycle

**BUT** representations are periodic; this can be a problem if the data are not.

Fourier basis is first choice in many fields, eg signal processing.

# B-spline Bases

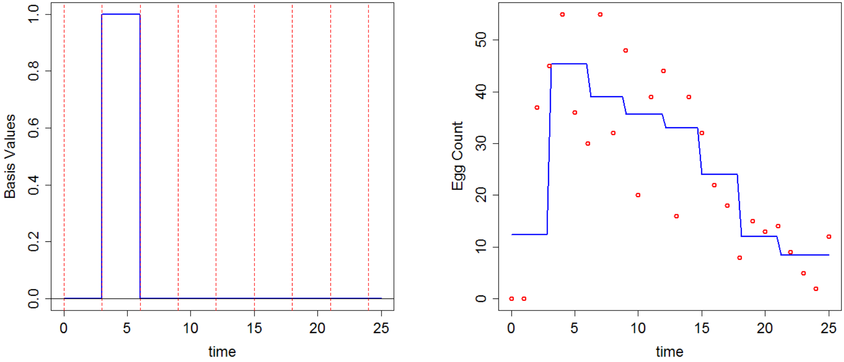
- Splines are polynomial segments joined end-to-end.
- Segments are constrained to be smooth at the joins.
- The points at which the segments join are called *knots*.
- System defined by
  - The order $m$ (order = degree+1) of the polynomial
  - the location of the knots.
- **Bsplines** are a particularly useful means of incorporating the constraints.

See de Boor, 2001, “A Practical Guide to Splines”, Springer.

# Splines

Medfly data with knots every 3 days.

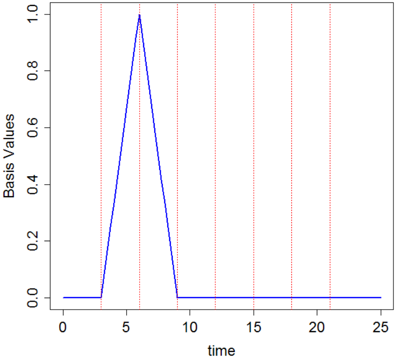
Splines of order 1: piecewise constant, discontinuous.

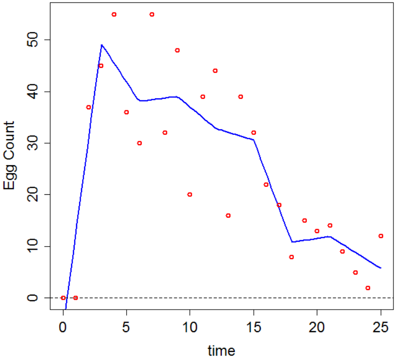


# Splines

Medfly data with knots every 3 days.

Splines of order 2: piecewise linear, continuous

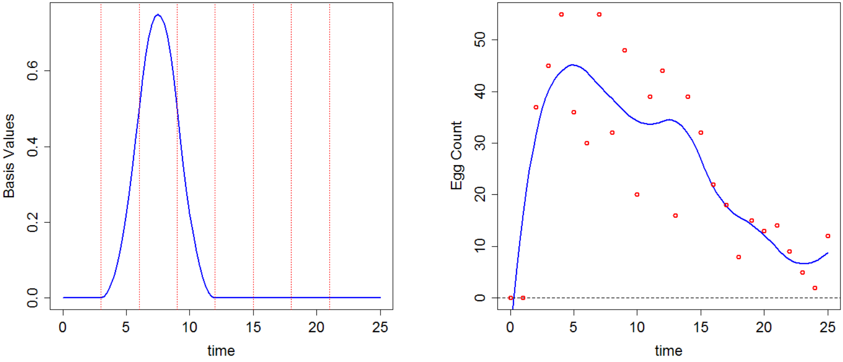


# Splines

Medfly data with knots every 3 days.

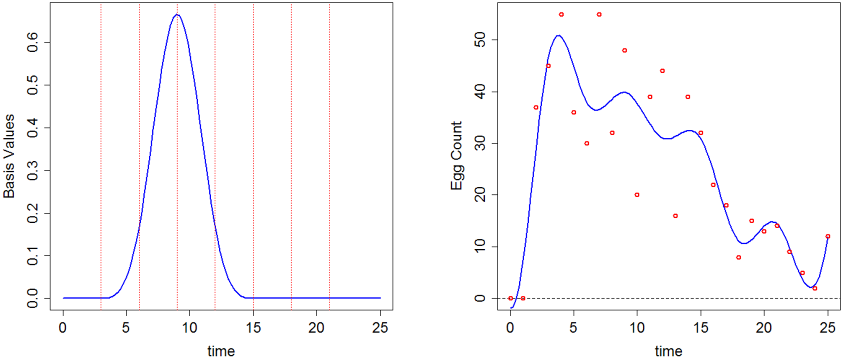
Splines of order 3: piecewise quadratic, continuous derivatives



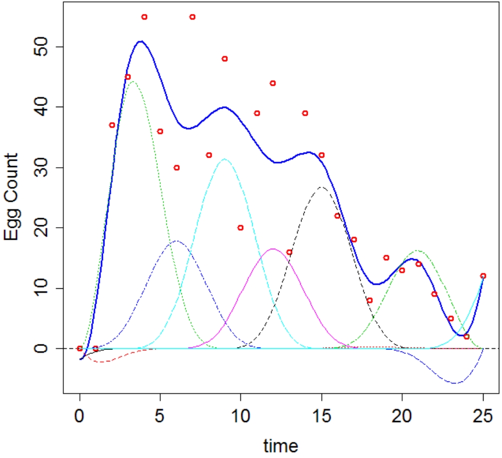
# Splines

Medfly data with knots every 3 days.

Splines of order 4: piecewise cubic, continuous 2nd derivatives



# An illustration of basis expansions for B-splines




Sum of scaled basis functions results in fit.

# Properties of B-splines

- Number of basis functions:

$$order + number\ interior\ knots$$

- Order $m$ splines: derivatives up to $m-2$ are continuous.
- Support on $m$ adjacent intervals – highly sparse design matrix.

Advice

- Flexibility comes from knots; derivatives from order.
- Theoretical justification (later) for knots at observation times.
- Frequently, fewer knots will do just as well (approximation properties can be formalized).

# Other Bases in fda Library

- **Constant** $\phi(t) = 1$, the simplest of all.
- **Monomial** $1, x, x^2, x^3, \ldots,$ ..., mostly for legacy reasons.
- **Power** $t^{\lambda_1}, t^{\lambda_2}, t^{\lambda_3}, \ldots,$ powers are distinct but not necessarily integers or positive.
- **Exponential** $e^{\lambda_1 t}, e^{\lambda_2 t}, e^{\lambda_3 t}, \ldots$

Other possible bases to represent $x(t)$:

- **Wavelets** especially for sharp, local features (not in `fda`)
- **Empirical** functional Principal Components (special topics)

# 2. Smoothing Penalties

# Ordinary Least-Squares Estimates

Assume we have observations for a single curve

$$y_i = x(t_i) + \epsilon$$

and we want to estimate

$$x(t) \approx \sum_{j=1}^{K} c_j \phi_j(t)$$

Minimize the sum of squared errors:

$$SSE = \sum_{i=1}^{n} (y_i - x(t_i))^2 = \sum_{i=1}^{n} (y_i - \Phi(t_i)\mathbf{c})^2$$

This is just linear regression!

# Linear Regression on Basis Functions

- If the $N$ by $K$ matrix $\mathbf{\Phi}$ contains the values $\phi_j(t_k)$, and $\mathbf{y}$ is the vector $(y_1, \ldots, y_N)$, we can write

$$SSE(\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi}\mathbf{c})^T(\mathbf{y} - \mathbf{\Phi}\mathbf{c})$$

- The error sum of squares is minimized by the *ordinary least squares estimate*

$$\hat{\mathbf{c}} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{y}$$

- Then we have the estimate

$$\hat{x}(t) = \Phi(t)\hat{\mathbf{c}} = \Phi(t)\left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{y}$$

# Smoothing Penalties

- Problem: how to choose a basis? Large affect on results.
- Finesse this by specifying a very rich basis, but then imposing smoothness.
- In particular, add a penalty to the least-squares criterion:

$$\text{PENSSE} = \sum_{i=1}^{n} (y_i - x(t_i))^2 + \lambda J[x]$$

- $J[x]$ measures "roughness" of $x$.
- $\lambda$ represents a continuous tuning parameter (to be chosen):
  - $\lambda \uparrow \infty$: roughness increasingly penalized ,$x(t)$ becomes smooth.
  - $\lambda \downarrow 0$: penalty reduces, $x(t)$ fits data better.

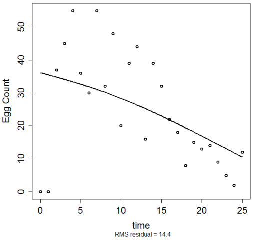
# What do we mean by smoothness?

Some things are fairly clearly smooth:

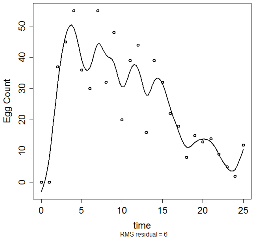
- constants
- straight lines

What we really want to do is eliminate small “wiggles” in the data while retaining the right shape

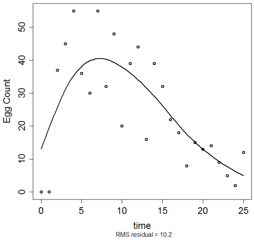
Too smooth




Too rough




Just right



# The $D$ Operator

We use the notation that for a function $x(t)$,

$$Dx(t) = \frac{d}{dt}x(t)$$

We can also define further derivatives in terms of powers of $D$:

$$D^2x(t) = \frac{d^2}{dt^2}x(t), \ldots, D^kx(t) = \frac{d^k}{dt^k}x(t), \ldots$$

- $Dx(t)$ is the instantaneous *slope* of $x(t)$; $D^2x(t)$ is its *curvature*.
- We measure the size of the curvature for all of $x$ by

$$J_2[x] = \int \left[D^2x(t)\right]^2 dt$$

# The Smoothing Spline Theorem

Consider the “usual” penalized squared error:

$$PENSSE_\lambda(x) = \sum (y_i - x(t_i))^2 + \lambda \int \left[D^2 x(t)\right]^2 dt$$

- The function $x(t)$ that minimizes $PENSSE_\lambda(x)$ is
  - a spline function of order 4 (piecewise cubic)
  - with a knot at each sample point $t_i$

Cubic B-splines are exact; other systems will approximate solution as close as desired.

# Calculating the Penalized Fit

When $x(t) = \Phi(t)\mathbf{c}$, we have that

$$\int \left[D^2 x(t)\right]^2 dt = \int \mathbf{c}^T \left[D^2\Phi(t)\right] \left[D^2\Phi(t)\right]^T \mathbf{c} dt = \mathbf{c}^T R_2 \mathbf{c}$$

$[R_2]_{jk} = \int [D^2\phi_j(t)][D^2\phi_k(t)]dt$ is the *penalty matrix*.

The penalized least squares estimate for $\mathbf{c}$ is n

$$\hat{\mathbf{c}} = \left[\mathbf{\Phi}^T\mathbf{\Phi} + \lambda R_2\right]^{-1} \mathbf{\Phi}^T\mathbf{y}$$

This is still a linear smoother:

$$\hat{\mathbf{y}} = \mathbf{\Phi}\left[\mathbf{\Phi}^T\mathbf{\Phi} + \lambda R_2\right]^{-1} \mathbf{\Phi}^T\mathbf{y} = S(\lambda)\mathbf{y}$$

# More General Smoothing Penalties

- $D^2x(t)$ is only one way to measure the roughness of $x$.
- If we were interested in $D^2x(t)$, we might penalize $D^4x(t)$.
- What about the weather data? We know temperature is periodic, and not very different from a sinusoid.
- The *Harmonic acceleration* of $x$ is

$$Lx = \omega^2 Dx + D^3 x$$

  and $L\cos(\omega t) = 0 = L\sin(\omega t)$.
- We can measure departures from a sinusoid by

$$J_L[x] = \int [Lx(t)]^2 \, dt$$

# A Very General Notion

We can be even more general and allow roughness penalties to use any *linear differential operator*

$$Lx(t) = \sum_{k=1}^{m} \alpha_k(t) D^k x(t)$$

Then $x$ is "smooth" if $Lx(t) = 0$.

We will see later on that we can even ask the data to tell us what should be smooth.

However, we will rarely need to use anything so sophisticated.

# Linear Smooths and Degrees of Freedom

- In least squares fitting, the degrees of freedom used to smooth the data is exactly $K$, the number of basis functions
- In penalized smoothing, we can have $K > n$.
- The smoothing penalty reduces the flexibility of the smooth
- The degrees of freedom are controlled by $\lambda$. A natural measure turns out to be

$$df(\lambda) = \text{trace}\left[S(\lambda)\right], \; S(\lambda) = \mathbf{\Phi}\left[\mathbf{\Phi}^T\mathbf{\Phi} + \lambda R_L\right]^{-1}\mathbf{\Phi}^T$$

- Medfly data fit with 25 basis functions, $\lambda = e^4$ resulting in $df = 4.37$.

# Choosing Smoothing Parameters: Cross Validation

There are a number of data-driven methods for choosing smoothing parameters.

- Ordinary Cross Validation: leave one point out and see how well you can predict it:

$$\mathrm{OCV}(\lambda) = \frac{1}{n}\sum\left(y_i - x_\lambda^{-i}(t_i)\right)^2 = \frac{1}{n}\sum\frac{(y_i - x_\lambda(t_i))^2}{(1 - S(\lambda)_{ii})^2}$$

- Generalized Cross Validation tends to smooth more:

$$\mathrm{GCV}(\lambda) = \frac{\sum(y_i - x_\lambda(t_i))^2}{\left[\mathrm{trace}(\mathbb{I} - S(\lambda))\right]^2}$$

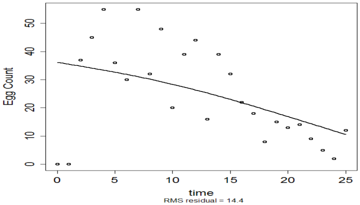
  will be used here.

- Other possibilities: AIC, BIC,...

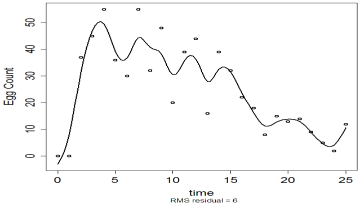
# Generalized Cross Validation

Use a grid search, best to do this for $\log(\lambda)$

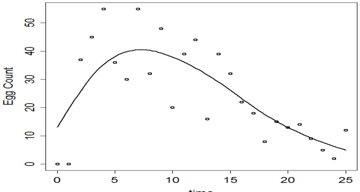
Smooth



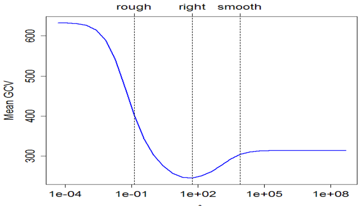

Rough




Right




GCV



# Alternatives: Smoothing and Mixed Models

Connection between the smoothing criterion for $\mathbf{c}$:

$$\text{PENSSE}(\lambda) = \sum_{i=1}^{n} (y_i - \mathbf{c}^T \Phi(t_i))^2 + \lambda \mathbf{c}^T R \mathbf{c}$$

and negative log likelihood if $\mathbf{c} \sim N(0, \tau^2 R^{-1})$:

$$\log L(\mathbf{c}|\mathbf{y}) = \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mathbf{c}^T \Phi(t_i))^2 + \frac{1}{2\tau^2} \mathbf{c}^T R \mathbf{c}$$

(note that $R$ is singular – must use generalized inverse).

Suggests using ReML estimates for $\sigma^2$ and $\tau^2$ in place of $\lambda$.

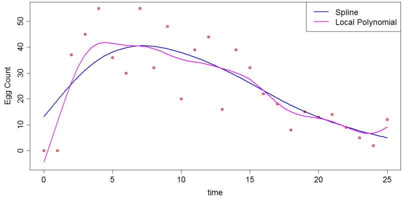
This can be carried further in FDA; see Ruppert, Wand and Carroll, 2003, "Semiparametric Regression"

# Alternatives: Local Polynomial Regression

- Alternative to basis expansions.
- Perform polynomial regression, but only near point of interest

$$(\hat{\beta}_0(t), \hat{\beta}_1(t)) = \underset{\beta_0,\beta_1}{\text{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1(t - t_i))^2 K\left(\frac{t - t_i}{\lambda}\right)$$

  Weights $(y_i, t_i)$ by distance from $t$
- Estimate $\hat{x}(t) = \hat{\beta}_0(t)$, $\widehat{Dx}(t) = \hat{\beta}_1(t)$.
- $\lambda$ is bandwidth: how far away can $(y_i, t_i)$ have influence?



# Summary

1. Basis Expansions

$$x_i(t) = \Phi(t)\mathbf{c}_i$$

   - Good basis systems approximate any (sufficiently smooth) function arbitrarily well.
   - Fourier bases useful for periodic data.
   - B-splines make efficient, flexible generic choice.

2. Smoothing Penalties used to penalize roughness of result
   - $Lx(t) = 0$ defines what is "smooth".
   - Commonly $Lx = D^2x \Rightarrow$ straight lines are smooth.
   - Alternative: $Lx = D^3x + \omega Dx \Rightarrow$ sinusoids are smooth.
   - Departures from smoothness traded off against fit to data.
   - GCV used to decide on trade off; other possibilities available.

These tools will be used throughout the rest of FDA.

Once estimated, we will treat smooths as fixed, observed data (but see extensions at end).

# Exploratory Data Analysis

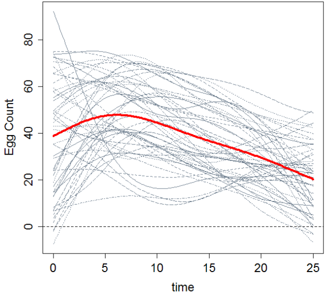
# Mean and Variance

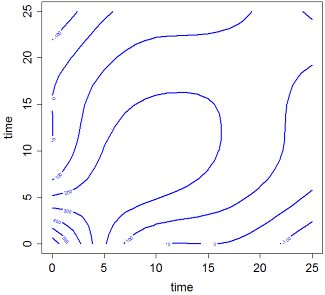
Summary statistics:

- mean $\bar{x}(t) = \frac{1}{n}\sum x_i(t)$
- covariance

  $\sigma(s,t) = \text{cov}(x(s), x(t)) = \frac{1}{n}\sum(x_i(s) - \bar{x}(s))(x_i(t) - \bar{x}(t))$

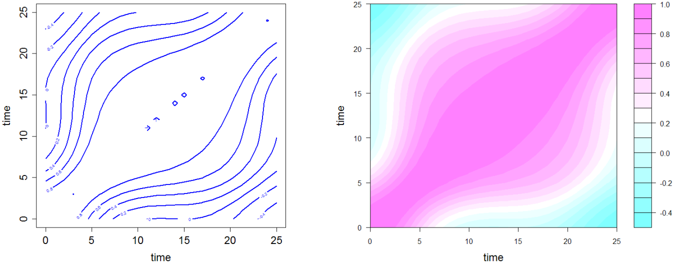
Medfly Data:



# Correlation

$$\rho(s,t) = \frac{\sigma(s,t)}{\sqrt{\sigma(s,s)}\sqrt{\sigma(t,t)}}$$




From multivariate to functional data: turn subscripts $j, k$ into arguments $s, t$.

# Functional PCA

- Instead of covariance matrix $\Sigma$, we have a surface $\sigma(s, t)$.
- Would like a low-dimensional summary/interpretation.
- Multivariate PCA, use Eigen-decomposition:

$$\Sigma = U^T DU = \sum_{j=1}^{p} d_j u_j u_j^T$$

and $u_i^T u_j = I(i = j)$.

- For functions: use Karhunen-Loève decomposition:

$$\sigma(s, t) = \sum_{j=1}^{\infty} d_j \xi_j(s) \xi_j(t)$$

for $\int \xi_i(t)\xi_j(t)dt = I(i = j)$

# PCA and Karhunen-Loève

$$\sigma(s,t) = \sum_{i=1}^{\infty} d_i \xi_i(s) \xi_i(t)$$

- $\xi_i(t)$ sequentially maximize $\text{Var}_i\left[\int \xi_i(t) x_j(t) dt\right]$.
- $d_i = \text{Var}_i\left[\int \xi_i(t) x_j(t) dt\right]$
- $d_i / \sum d_i$ is proportion of variance explained
- Principal component scores are

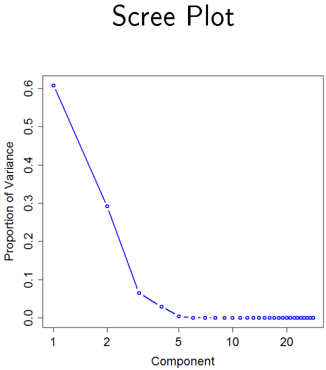
$$f_{ij} = \int \xi_j(t)[x_i(t) - \bar{x}(t)]dt$$

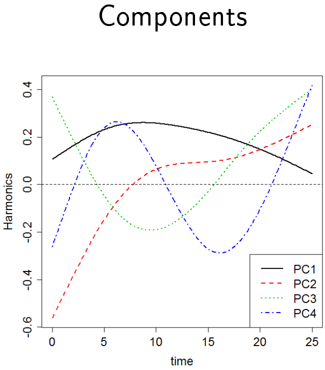
- Reconstruction of $x_i(t)$:

$$x_i(t) = \bar{x}(t) + \sum_{j=1}^{\infty} f_{ij} \xi_j(t)$$

# functional Principal Components Analysis

## fPCA of Medfly data

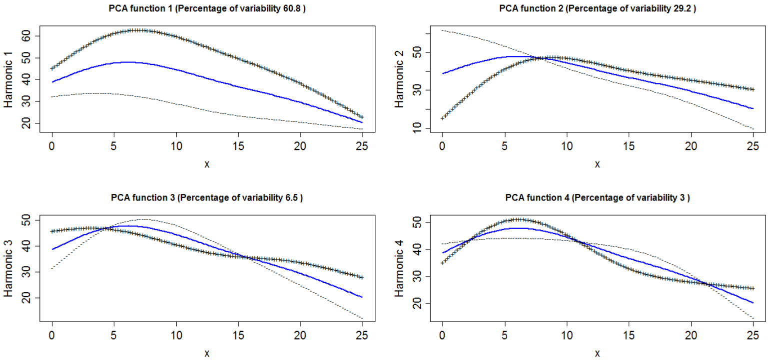



Usual multivariate methods: choose # components based on percent variance explained, screeplot, or information criterion.

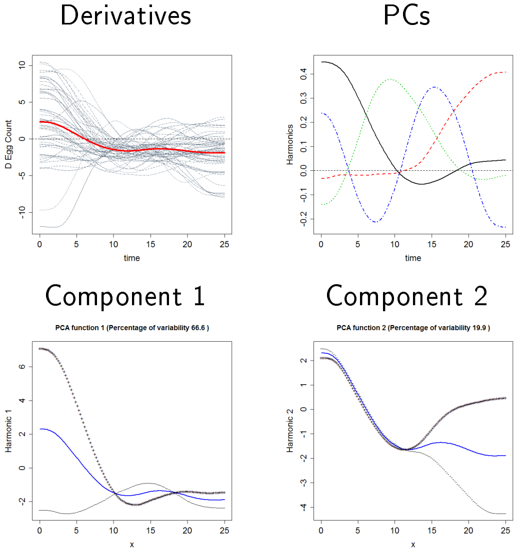
# functional Principal Components Analysis

Interpretation often aided by plotting $\bar{x}(t) \pm 2\sqrt{d_i}\xi_i(t)$




PC1 = overall fecundity
PC2 = beginning versus end
PC3 = middle versus ends

# Derivatives

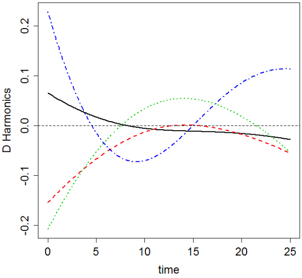



- Often useful to examine a rate of change.
- Examine first derivative of medfly data.
- Variation divides into fast or slow either early or late.

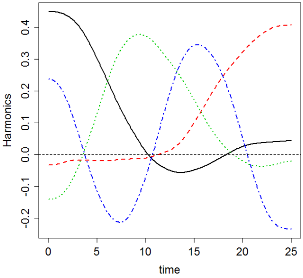
# Derivatives and Principal Components

Note that the derivatives of Principal Components are *not* the same as the Principal Components of Derivatives.

D[PCA(x)]




PCA(D[x])



# The fda Package

# fda Objects

The fda package provides utilities based on basis expansions and smoothing penalties.

fda works by defining objects that can be manipulated with pre-defined functions.

In particular

- **basis objects** define basis systems that can be used
- **fd objects** store functional data objects
- **Lfd objects** define smoothing penalties
- **fdPar objects** collect fd, Lfd plus a smoothing parameter
- **bifd objects** store functions of two-dimensions

Each of these are lists with prescribed elements.

# Basis Objects

Define basis systems of various types. They have elements

- `rangeval` Range of values for which basis is defined.
- `nbasis` Number of basis functions.

Specific basis systems require other arguments.

Basis objects created by `create....basis` functions. eg

```
fbasis = create.fourier.basis(c(0,365),21)
```

creates a fourier basis on [0 365] with 21 basis functions.

# Bspline Basis Objects

Bspline bases also require

- **norder** Order of the splines.
- **breaks** Knots (or break-points) for the splines.

```
nbasis = 17
norder = 6
months = cumsum(c(0,31,28,31,30,31,30,31,31,30,31,30,31))
bbasis = create.bspline.basis(c(0,365),nbasis,norder,months)
```

Creates a B-spline basis of order 6 on the year ([0 365]) with knots at the months.

Note that

$$\texttt{nbasis} = \texttt{length(knots)} + \texttt{norder} - 2$$

nbasis is fragile in case of conflict.

# Manipulating Basis Objects

Some functions that work with bases:

```
plot(bbasis)
```

plots bbasis.

```
eval.basis(0:365,fbasis)
```

evaluates fbasis at times 0:365.

```
inprod(bbasis,fbasis)
```

produces the inner product matrix $J_{ij} = \int \phi_i(t)\psi_j(t)dt$.

Additional arguments allow use of $L\Phi$ for linear differential operators $L$.

# Functional Data (fd) Objects

Stores functional data: a list with elements

- `coefs` array of coefficients
- `basis` basis object
- `fdnames` defines dimension names

```
fdobj = fd(coefs,bbasis)
```

creates a functional data object with coefficients `coefs` and basis `bbasis`

`coefs` has three dimensions corresponding to

1. index of the basis function
2. replicate
3. dimension

# Functional Arithmetic

fd objects can be manipulated arithmetically

`fdobj1+fdobj2`, `fdobj1^k`, `fdobj1*fdobj2`

are defined pointwise.

fd objects can also be subset

`fdobj[3,2]`

gives the 2nd dimension of the 3rd observation

Additionally

`eval.fd(0:365,fdobj)` returns an array of values of fdobj on 0:365.

`deriv.fd(fdobj,nderiv)` gives the nderiv-th derivative of fdobj.

`plot(fdobj)` plots fdobj

eval.fd and plot can also take argument nderiv.

# Lfd Objects

Define smoothing penalties

$$Lx = D^m x - \sum_{j=0}^{m-1} \alpha_j(t) D^j x$$

and require the $\alpha_j$ to be given as a list of `fd` objects.

Two common shortcuts:

`int2Lfd(k)` creates an `Lfd` object $Lx = D^k x$

`vec2Lfd(a)` for vector `a` of length `m` creates an `Lfd` object $Lx = D^m x - \sum_{j=1}^{m} a_j D^{j-1} x$.

In particular

```
vec2Lfd(c(0,-2*pi/365,0))
```

creates a Harmonic acceleration penalty $Lx = D^3 x + \frac{2\pi}{365} Dx$.

# fdPar Objects

This is a utility for imposing smoothing. It collects

- `fdobj` an fd (or a basis) object.
- `Lfdobj` a Lfd object.
- `lambda` a smoothing parameter.

# bifd Objects

Represents functions of two dimensions $s$ and $t$ as

$$x(s,t) = \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} \phi_i(s)\psi_j(t)c_{ij}$$

requires

- `coefs` for the matrix of $c_{ij}$.
- `sbasis` basis object defining the $\phi_i(s)$.
- `tbasis` basis object defining the $\psi_j(t)$.

Can also be evaluated (but not plotted).

`bifdPar` objects store `bifd` plus `Lfd` objects and $\lambda$ for each of $s$ and $t$.

# Smoothing Functions

Main smoothing function is `smooth.basis`

```
data(daily)
argvals = (1:365)-0.5
fdParobj = fdPar(fbasis,int2Lfd(2),1e-2)
tempSmooth = smooth.basis(argvals,tempav,fdParobj)
```

smooths the Canadian temperature data with a second derivative penalty, $\lambda = 0.01$. Along with an `fd` object it returns

| | |
|---|---|
| `df` | equivalent degrees of freedom |
| `SSE` | total sum of squared errors |
| `gcv` | vector giving GCV for each smooth |

Typically, $\lambda$ is chosen to minimize average `gcv`.

Note: numerous other smoothing functions, `Data2fd` just returns the `fd` and can avoid the `fdPar` object, `data2fd` is depricated.

## Functional Statistics

Basic utilities:

- `mean.fd` mean `fd` object
- `var.fd` Variance or covariance (`bifd` object)
- `cor.fd` Correlation (given as a matrix)
- `sd.fd` Standard deviation (root diagonal of `var.fd`)

In addition, fPCA obtained through

```
temppca=pca.fd(tempfd$fd,nharm=4,fdParobj)
```

(Smoothing not strictly necessary). `pca.fd` output:

- `harmonics` `fd` objects giving eigen-functions
- `values` eigen values
- `scores` PCA scores
- `varprop` Proportion of variance explained

Diagnostics plots given by `plot(temppca)`

# Functional Linear Models

# Statistical Models

So far we have focussed on *exploratory data analysis*

- Smoothing
- Functional covariance
- Functional PCA

Now we wish to examine predictive relationships $\rightarrow$ generalization of linear models.

$$y_i = \alpha + \sum \beta_j x_{ij} + \epsilon_i$$

# Functional Linear Regression

$$y_i = \alpha + \mathbf{x}_i\beta + \epsilon_i$$

Three different scenarios for $y_i$ $\mathbf{x}_i$

- Functional covariate, scalar response
- Scalar covariate, functional response
- Functional covariate, functional response

We will deal with each in turn.

# Scalar Response Models

# Scalar Response Models

We observe $y_i, x_i(t)$, and want to model dependence of $y$ on $x$.

Option 1: choose $t_1, \ldots, t_k$ and set

$$\begin{aligned} y_i &= \alpha + \sum \beta_j x_i(t_j) + \epsilon_i \\ &= \alpha + \mathbf{x}_i \beta + \epsilon \end{aligned}$$

But how many $t_1, \ldots, t_k$ and which ones?

See McKeague and Sen, 2010, "Fractals with Point Impact in Functional Linear Regression", *Biometrika*; Ferratty, Hall and Vieu, 2010, "Most Predictive Design points for Functional Data Predictors", *Biometrika*.

# In the Limit

If we let $t_1, \ldots$ get increasingly dense

$$y_i = \alpha + \sum \beta_j x_i(t_j) + \epsilon_i = \alpha + \mathbf{x}_i \beta + \epsilon_i$$

becomes

$$y_i = \alpha + \int \beta(t) x_i(t) dt + \epsilon_i$$

General trick: functional data model = multivariate model with sums replaced by integrals.

Already seen in fPCA scores $x^T u_i \to \int x(t) \xi_i(t) dt$.

# Identification

Problem:

- In linear regression, we must have fewer covariates than observations.
- If I have $y_i, x_i(t)$, there are *infinitely* many covariates.

$$y_i = \alpha + \int \beta(t)x_i(t)dt + \epsilon_i$$

Estimate $\beta$ by minimizing squared error:

$$\beta(t) = \text{argmin} \sum \left( y_i - \alpha - \int \beta(t)x_i(t)dt \right)^2$$

But I can always make the $\epsilon_i = 0$.

# Smoothing

Additional constraints: we want to insist that $\beta(t)$ is smooth.

Add a smoothing penalty:

$$\text{PENSSE}_{\lambda}(\beta) = \sum_{i=1}^{n} \left( y_i - \alpha - \int \beta(t) x_i(t) dt \right)^2 + \lambda \int [L\beta(t)]^2 \, dt$$

Very much like smoothing (can be made mathematically precise).

Still need to represent $\beta(t)$ – use a basis expansion:

$$\beta(t) = \sum c_i \phi_i(t).$$

# Calculation

$$y_i = \alpha + \int \beta(t) x_i(t) dt + \epsilon_i = \alpha + \left[ \int \Phi(t) x_i(t) dt \right] \mathbf{c} + \epsilon_i$$
$$= \alpha + \mathbf{x}_i \mathbf{c} + \epsilon_i$$

for $\mathbf{x}_i = \int \Phi(t) x_i(t) dt$, with $Z_i = [1, \ \mathbf{x}_i]$,

$$\mathbf{y} = Z \begin{bmatrix} \alpha \\ \mathbf{c} \end{bmatrix} + \epsilon$$

and with smoothing penalty matrix $R_L$:

$$[\hat{\alpha} \ \hat{\mathbf{c}}^T]^T = \left( Z^T Z + \lambda R_L \right)^{-1} Z^T \mathbf{y}$$

Then

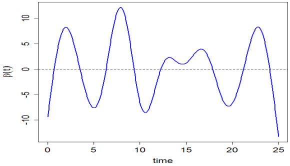
$$\hat{\mathbf{y}} = \int \hat{\beta}(t) x_i(t) dt = Z \begin{bmatrix} \hat{\alpha} \\ \hat{\mathbf{c}} \end{bmatrix} = S_\lambda \mathbf{y}$$

# Choosing Smoothing Parameters

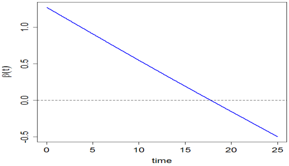
## Cross-Validation:

$$\mathrm{OCV}(\lambda) = \sum \left( \frac{y_i - \hat{y}_i}{1 - S_{ii}} \right)^2$$

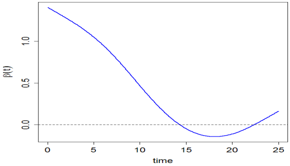
$\lambda = e^{-1}$



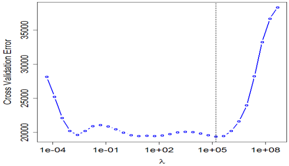

$\lambda = e^{20}$




$\lambda = e^{12}$




CV Error



# Confidence Intervals

Assuming independent

$$\epsilon_i \sim N(0, \sigma_e^2)$$

We have that

$$\text{Var}\left[\begin{array}{c} \hat{\alpha} \\ \hat{\mathbf{c}} \end{array}\right] = \left[\left(Z^T Z + \lambda R\right)^{-1} Z^T\right] \left[\sigma_e^2 \mathbb{I}\right] \left[Z \left(Z^T Z + \lambda R\right)^{-1}\right]$$

Estimate

$$\hat{\sigma}_e^2 = SSE/(n - df), \ df = \text{trace}(S_\lambda)$$

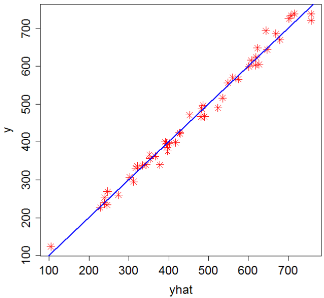
And (pointwise) confidence intervals for $\beta(t)$ are

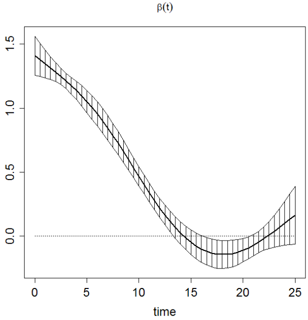
$$\Phi(t)\hat{\mathbf{c}} \pm 2\sqrt{\Phi(t)^T \text{Var}[\hat{\mathbf{c}}]\Phi(t)}$$

can also include scalar covariates.

# Confidence Intervals

$R^2 = 0.987$ $\sigma^2 = 349$, df $= 5.04$




Extension to multiple functional covariates follows same lines:

$$y_i = \beta_0 + \sum_{j=1}^{p} \int \beta_j(t) x_{ij}(t) dt + \epsilon_i$$

# Uniform Confidence Intervals

[Currently only implemented in `SCBmeanfd` for $\hat{\mu}(t)$]

- Pointwise bands give coverage *at each t*
- Uniform (simultaneous) bands are such that *the true curve will fall entirely within the band 95% of the time.*

If $\text{cov}(\hat{\beta}(t), \hat{\beta}(s)) = \sigma(s,t)$, $\text{cor}(\hat{\beta}(t), \hat{\beta}(s)) = \rho(s,t)$ use

$$\hat{\beta}(t) \pm Z_\alpha \sqrt{\sigma(t,t)}$$

where $Z_a$ satisfies

$$P_{X \sim N(0,\rho)}(\max_t X(t) < Z_\alpha) = \alpha$$

Usually obtained from

- Simulate $X_1, \ldots, X_{1000}$ from $N(0, \rho)$.
- $Z_\alpha$ from $\alpha$-quantile of $\max_t X_1(t), \ldots, \max_t X_{1000}(t)$.

# functional Principal Components Regression

# functional Principal Components Regression

Alternative: principal components regression.

$$x_i(t) = \sum d_{ij}\xi_j(t) \quad d_{ij} = \int x_i(t)\xi_j(t)dt$$

Consider the model:

$$y_i = \beta_0 + \sum \beta_j d_{ij} + \epsilon_i$$

- Reduces to a standard linear regression problem.
- Avoids the need for cross-validation (assuming number of PCs is fixed).

By far the most theoretically studied method.

# fPCA and Functional Regression Interpretation

$$y_i = \beta_0 + \sum \beta_j d_{ij} + \epsilon_i$$

Recall that $d_{ij} = \int x_i(t)\xi_j(t)dt$ so

$$y_i = \beta_0 + \sum \int \beta_j \xi_j(t) x_i(t) dt + \epsilon_i$$

and we can interpret

$$\beta(t) = \sum \beta_j \xi_j(t)$$

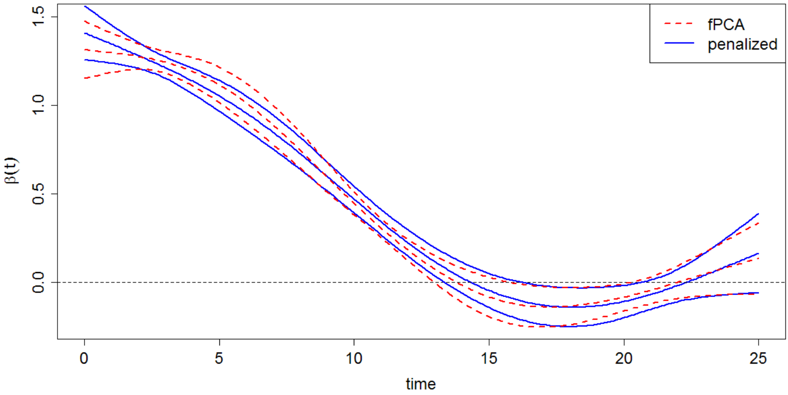
and write

$$y_i = \beta_0 + \int \beta(t) x_i(t) dt + \epsilon_i$$

Confidence intervals derive from variance of the $d_{ij}$.

# A Comparison

Medfly Data: fPCA on 4 components ($R^2 = 0.988$) vs Penalized Smooth ($R^2 = 0.987$)



# Two Fundamental Approaches

(Almost) all methods reduce to one of

1. Perform fPCA and use PC scores in a multivariate method.
2. Turn sums into integrals and add a smoothing penalty.

Applied in functional versions of

- generalized linear models
- generalized additive models
- single index models
- survival analysis
- mixture regression
- ...

Both methods also apply to functional response models.

# Functional Response Models

# Functional Response Models

Case 1: Scalar Covariates: $(y_i(t), \mathbf{x}_i)$, most general linear model is

$$y_i(t) = \beta_0(t) + \sum_{j=1}^{p} \beta_i(t) x_{ij}.$$

Conduct a linear regression at each time $t$ (also works for ANOVA effects).

But we might like to smooth; penalize *integrated squared error*

$$\text{PENSISE} = \sum_{i=1}^{n} \int (y_i(t) - \hat{y}_i(t))^2 \, dt + \sum_{j=0}^{p} \lambda_j \int [L_j \beta_j(t)]^2 \, dt$$

Usually keep $\lambda_j$, $L_j$ all the same.

# Concurrent Linear Model

Extension of scalar covariate model: response only depends on $x(t)$ at the current time

$$y_i(t) = \beta_0(t) + \beta_1(t)x_i(t) + \epsilon_i(t)$$

- $y_i(t)$, $x_i(t)$ must be measured on same time domain.
- Must be appropriate to compare observations time-point by time-point (see registration section).
- Especially useful if $y_i(t)$ is a derivative of $x_i(t)$ (see dynamics section).

# Confidence Intervals

We assume that

$$\mathrm{Var}(\epsilon_i) = \sigma(s, t)$$

then

$$\mathrm{Cov}(\beta(t), \beta(s)) = (X^T X)^{-1}\sigma(s, t).$$

Estimate $\sigma(s, t)$ from $e_i(t) = y_i(t) - \hat{y}_i(t)$.

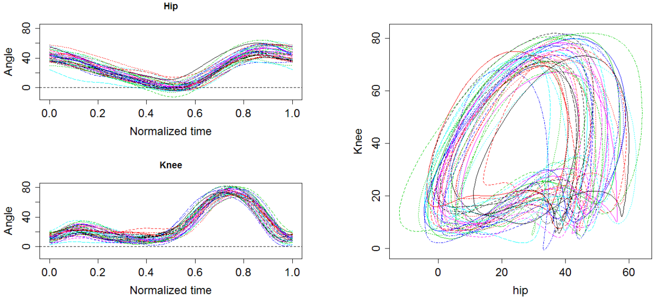
Pointwise confidence intervals ignore covariance; just use

$$\mathrm{Var}(\beta(t)) = (X^T X)^{-1}\sigma(t, t).$$

Effect of smoothing penalties (both for $y_i$ and $\beta_j$) can be incorporated.

# Gait Data

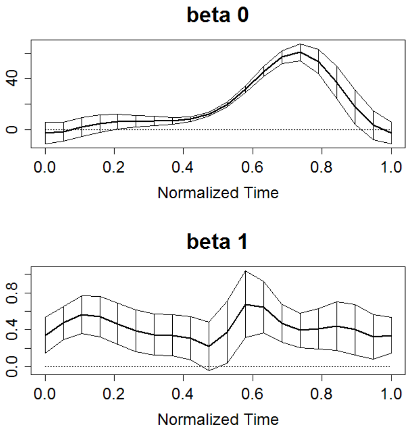
Gait data - records of the angle of hip and knee of 39 subjects taking a step.




Interest in kinetics of walking.

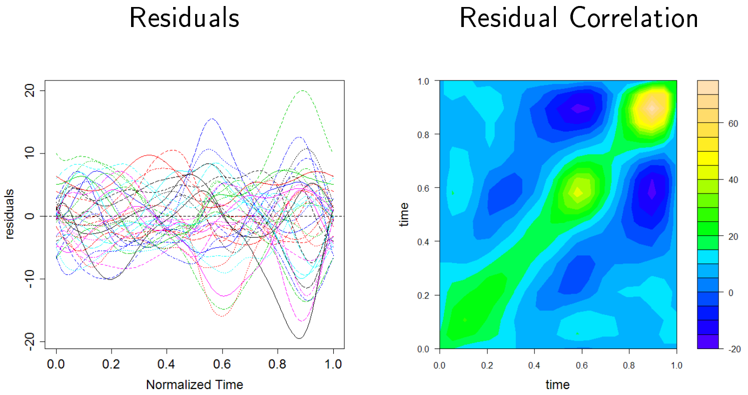
# Gait Model

$$\text{knee}(t) = \beta_0(t) + \beta_1(t)\text{hip}(t) + \epsilon(t)$$




- $\beta_0(t)$ indicates a well-defined autonomous knee cycle.
- $\beta_1(t)$ modulation of cycle with respect to hip
- More hip bend also indicates more knee bend; by a fairly constant amount throughout cycle.

# Gait Residuals: Covariance and Diagnostics




Examine residual functions for outliers, skewness etc (can be challenging).

Residual correlation may be of independent interest.

# Functional Response, Functional Covariate

General case: $y_i(t), x_i(s)$ - a functional linear regression at each time $t$:

$$y_i(t) = \beta_0(t) + \int \beta_1(s,t) x_i(s) ds + \epsilon_i(t)$$

- Same identification issues as scalar response models.
- Usually penalize $\beta_1$ in each direction separately

$$\lambda_s \int \left[L_s \beta_1(s,t)\right]^2 dsdt + \lambda_t \int \left[L_t \beta_1(s,t)\right]^2 dsdt$$

- Confidence Intervals etc. follow from same principles.

# Summary

Three models

Scalar Response Models
- Functional covariate implies a functional parameter.
- Use smoothness of $\beta_1(t)$ to obtain identifiability.
- Variance estimates come from sandwich estimators.

Concurrent Linear Model
- $y_i(t)$ only depends on $x_i(t)$ at the current time.
- Scalar covariates = constant functions.
- Will be used in dynamics.

Functional Covariate/Functional Response
- Most general functional linear model.
- See special topics for more + examples.

# Functional Linear Models in R

# fRegress

Main function for scalar responses and concurrent model, requires

- `y` response, either vector or `fd` object.
- `xlist` list containing covariates; vectors or `fd` objects.
- `betalist` list of `fdPar` objects to define bases and smoothing penalties for each coefficient

  **Note:** scalar covariates have *constant* coefficient functions, use a constant basis.

Returns depend on `y`; always

- `betaestlist` list of `fdPar` objects with estimated $\beta$ coefficients
- `yhatfdobj` predicted values, either numeric or `fd`.

# fRegress.stderr

Produces pointwise standard errors for the $\hat{\beta}_j$.

- `model` output of `fRegress`
- `y2cmap` smoothing matrix for the response (obtained from `smooth.basis`)
- `SigmaE` Error covariance for the response.

Produces a list including `betastderrlist`, which contains `fd` objects giving the pointwise standard errors.

# Other Utilities

`fRegress.CV` provides leave-one-out cross validation

- Same arguments as `fRegress`, allows use of specific observations.
- For concurrent linear models, we cross-validate by

$$\mathrm{CV}(\lambda) = \sum_{i=1}^{n} \int \left( y_i(t) - \hat{y}_\lambda^{-i}(t) \right)^2 dt$$

  $\hat{y}_\lambda^{-i}(t)$ = prediction with smoothing parameter $\lambda$ and without $i$th observation

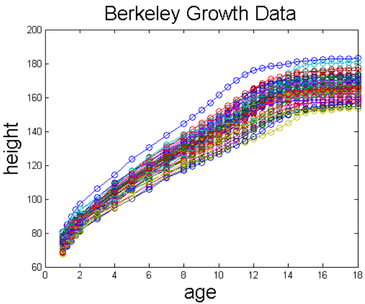
- Redundant (and slow) for scalar response models – use `OCV` in output of `fRegress` instead.

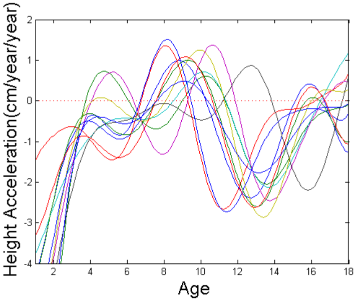
`plotbeta(betaestlist,betastderrlist)` produces graphs with confidence regions.

# Registration

# Berkeley Growth Data

- Heights of 20 girls taken from ages 0 through 18.
- Growth process easier to visualize in terms of acceleration.
- Peaks in acceleration = start of growth spurts.



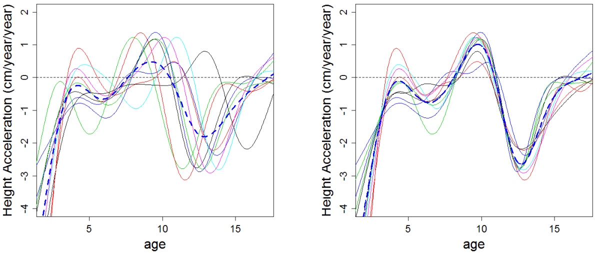
# The Registration Problem

Most analyzes only account for variation in *amplitude*.

Frequently, observed data exhibit features that vary in *time*.

Berkeley Growth Acceleration

Observed Aligned




- Mean of unregistered curves has smaller peaks than any individual curve.
- Aligning the curves reduces variation by 25%

# Defining a Warping Function

Requires a transformation of *time*.

Seek

$$s_i = w_i(t)$$

so that

$$\tilde{x}_i(t) = x_i(s_i)$$

are well aligned.

$w_i(t)$ are *time-warping* (also called *registration*) functions.

# Landmark registration

For each curve $x_i(t)$ we choose points

$$t_{i1}, \ldots, t_{iK}$$

We need a reference (usually one of the curves)

$$t_{01}, \ldots, t_{0K}$$

so these define constraints

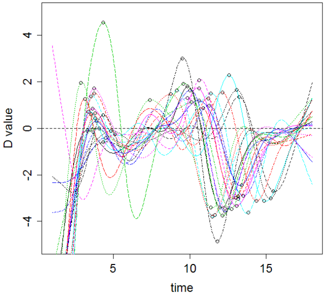
$$w_i(t_{ij}) = t_{0j}$$

Now we define a smooth function to go between these.

# Identifying Landmarks

Major landmarks of interest:

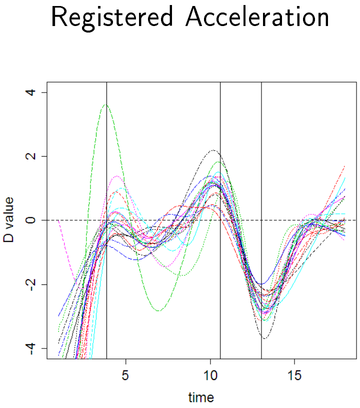
- where $x_i(t)$ crosses some value
- location of peaks or valleys
- location of inflections

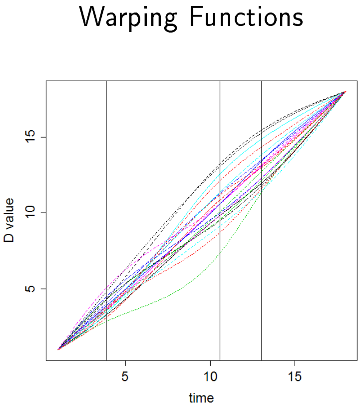



Almost all are points at which some derivative of $x_i(t)$ crosses zero.

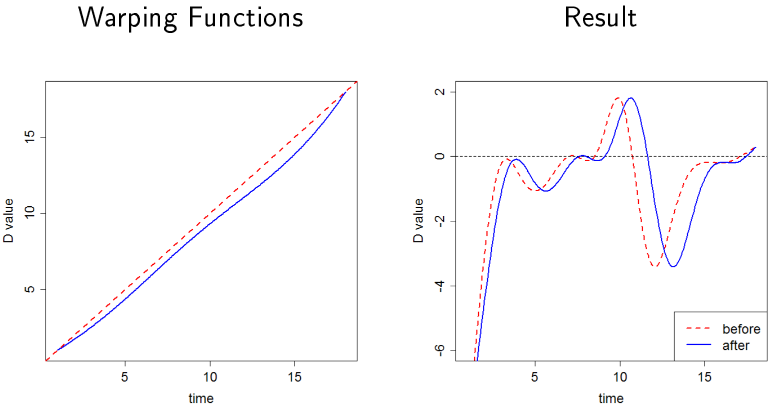
In practise, zero-crossings can be found automatically, but usually still require manual checking.

# Results of Warping



# Interpretation




Warping function below diagonal pushes registered function *later* in time.

# Constraints on Warping Functions

Let $t \in [0\ T]$, the $w_i(t)$ should follow a number of constraints:

- Initial conditions

$$w_i(0) = 0, \ w_i(T) = T$$

- landmarks

$$w_i(t_{ij}) = t_{0j}$$

- Monotonicity: if $t_1 < t_2$,

$$w_i(t_1) < w_i(t_2)$$

# Enforcing Constraints

Starting from the basis expansion

$$W_i(t) = \Phi(t)\mathbf{c}_i$$

we can transform $W_i(t)$ to enforce the following constraints:

Positive

$$E_i(t) = \exp(W_i(t))$$

Monotonic

$$I_i(t) = \int_0^t \exp(W_i(s))ds$$

Normalized

$$w_i(t) = T\frac{I_i(t)}{I_i(T)} = T\frac{\int_0^t \exp(W_i(s))ds}{\int_0^T \exp(W_i(s))ds}$$

The last of these defines a warping function.

# Computing Landmark Registration

Requires an estimate of

$$t_{0k} = \int_0^{t_{ik}} exp(\Phi(s)\mathbf{c}_i)ds$$

obtained from non-linear least squares.

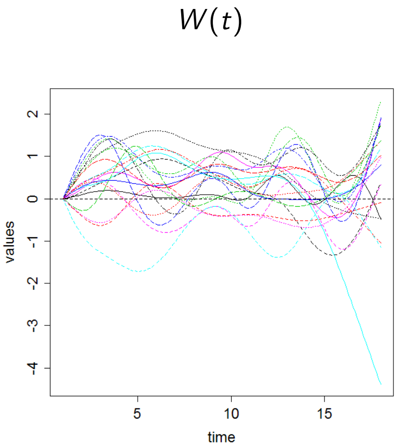
Convex optimization problem, but can be problematic.

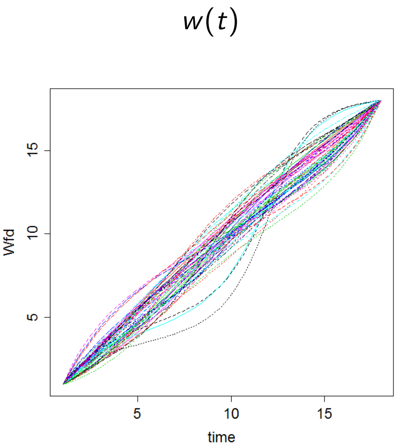
Directly estimating $\mathbf{c}_i$ to satisfy

$$t_{0k} = \Phi(t_{ik})\mathbf{c}_i$$

frequently retains monotonicity: easier, but should be checked.

# From $W(t)$ to $w(t)$




$W(0) = 0$ to obtain identifiability under normalization.

# Automatic Methods

Landmark registration requires

- clearly identifiable landmarks
- manual care in defining and finding landmarks

can we come up with something more general?

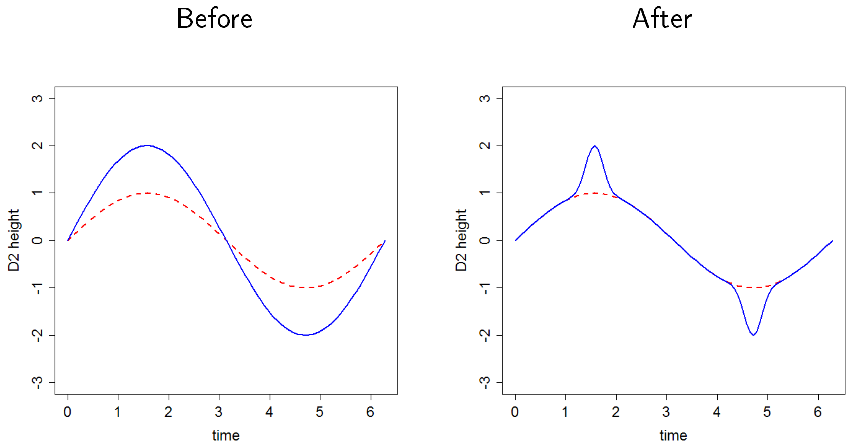
Obvious criterion is *between-curve sum of squares* for each curve

$$\text{BCSSE}[w_i] = \int (x_0(t) - x_i(w_i(t)))^2 \, dt$$

Requires a reference $x_0(t)$, works well for simple $w_i$ (eg linear transformations).

# Why Squared Error Doesn't Work for Flexible Methods

Amplitude-only variation is not ignored.



# Alternatives

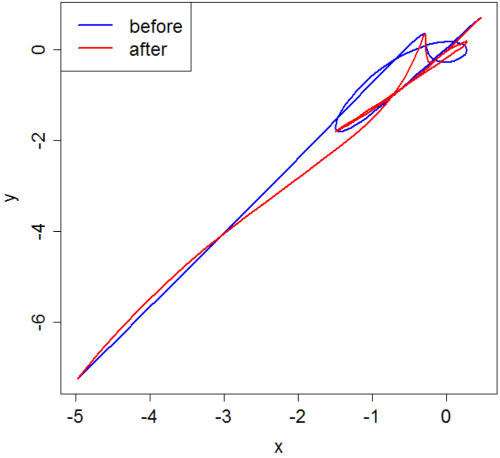
Major issue: we do not want to account for effects that are due solely to amplitude variation.

Instead want a measure of linearity between $x_i(w_i(t))$ and $x_0(t)$.

- For univariate $x_i(t)$, this is just correlation between curves.
- For multivariate $x_i(t)$, minimize smallest eigenvalue of correlation matrix.

Many other methods have been proposed.

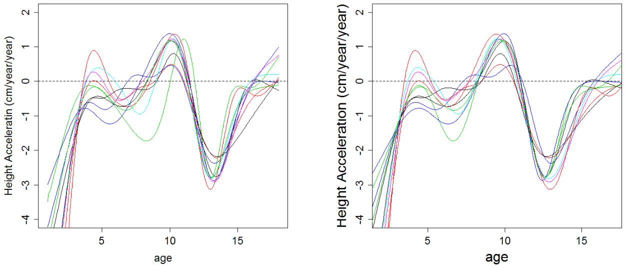
# Collinearity Before and After Registration



# Comparison Of Registration Results

First 10 subjects:

Landmark Automatic




Note: minimum-eigenvalue condition can have local minima and yield poor results.

# Alternatives

"Dark Art" of FDA. Approaches include

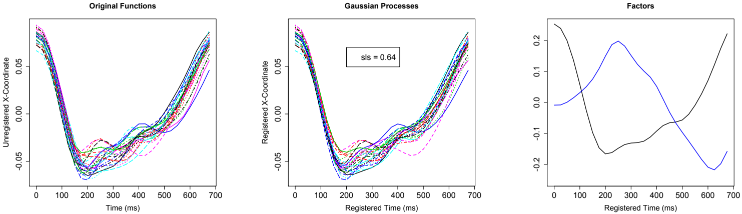
- $w_i(t) = 1/|Dx_i(t)|$; usually after scaling $x(t)$ (heuristic).
- Alternative metrics: Srivastava *et. al.* uses distances for $Dx(i)/\sqrt{|Dx_i(t)|}$.
- More complex models of variation. Earls and Hooker (2016) allow multiple directions of variation in registered functions

$$\begin{aligned} x_i(w_i(t)) &\sim GP(\mu(t) + z_0\xi_1(t) + z_2\xi_2(t), \Sigma) \\ (z_0, z_1) &\sim N(0, D) \\ w_i'(t) &\sim GP(0, G) \end{aligned}$$

estimated with Bayesian methods. See also Kneip and Ramsay 2008.

# Example

## Vertical Position of Juggling Hand



# Summary

- Registration – important tool for analyzing non-amplitude variation.
- Easiest: landmark registration, requires manual supervision.
- Continuous registration: numerically difficult alternative.
- Usually a preprocessing step; unaccounted for in inference.
- Warning: interaction with derivatives

  $$D\left[x\left(w(t)\right)\right] = D[w](t)D[x]\left(w(t)\right)$$

  Registration and $D$ do not commute; this can affect dynamics.
- R functions: `landmarkreg` and `register.fd`.

# Dynamics

# Relationships Between Derivatives

Access to derivatives of functional data allows new models.

Variant on the concurrent linear model: e.g.

$$Dy_i(t) = \beta_0(t) + \beta_1(t)y_i(t) + \beta_2(t)x_i(t) + \epsilon_i(t)$$

Higher order derivatives could also be used.

Can be estimated like concurrent linear model.

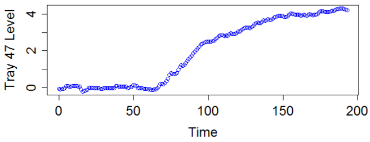
But how do we understand these systems?

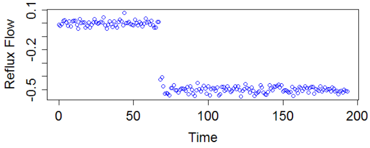
Focus: physical analogies and behavior of first and second order systems.

# First Order Systems

# Oil-Refinery Data

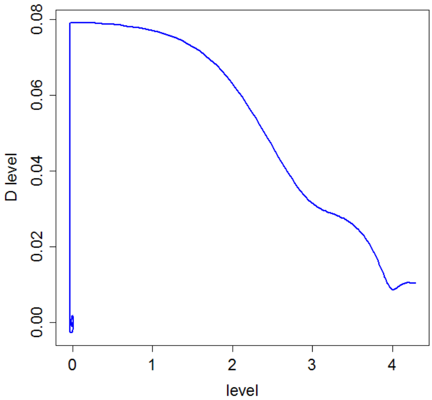
Measurement of level of oil in a refinery bucket and reflux flow out of bucket.




- Clearly, level responds to outflow.
- No linear model will capture this relationship.
- But, there is clearly something with fairly simple structure going on.

# Relationships Among Derivatives




- Initial period flat – no relationship.
- Following: negative relationship between $Dx$ and $x$.
- Suggests

$$Dx(t) = -\beta x(t) + \alpha u(t)$$

for input $u(t)$ (reflux flow).

# Mechanistic Models for Rates

Imagine a bucket with a hole in the bottom.

- Left to itself, the water will flow out the hole and the level will drop
- Adding water will increase the level in the bucket
- We want to describe the rate at which this happens

# Thinking About Models for Rates

Water in a leaky bucket.

To make things simple, assume the bucket has straight sides. Let $x(t)$ be the current volume of liquid in the bucket.

- Firstly, we need a rate for outflow without input ($u(t) = 0$).
  - The rate at which water leaves the bucket is proportional to how much pressure it is under.

$$Dx(t) = -Cp(t)$$

  - The pressure will be proportional to the weight of liquid. This in turn is proportional to volume: $p(t) = Kx(t)$. So

$$Dx(t) = -\beta x(t)$$

# Solution to First Order ODE

When the tap is turned on:

$$Dx(t) = -\beta x(t) + \alpha u(t)$$

Solutions to this equation are of the form

$$x(t) = Ce^{-\beta t} + \alpha \int_0^t e^{-(t-s)\beta} u(s) ds$$

This formula is not particularly enlightening; we would like to investigate how $x(t)$ behaves.

# Characterizing Solutions to Step-Function Inputs

In engineering, it is common to study the reaction of $x(t)$ when $u(t)$ is abruptly stepped up or down.

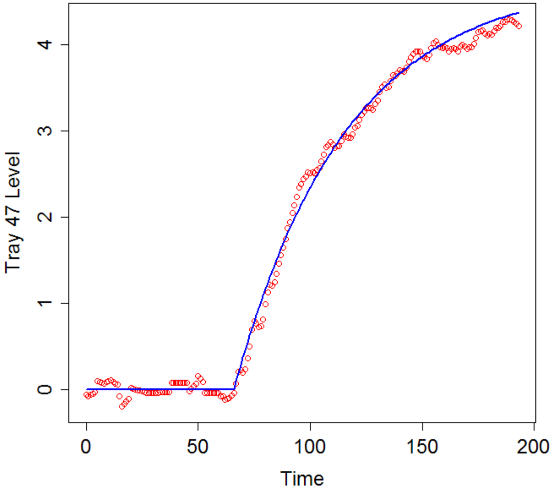
Let's start from $x(0) = 0$ $u(0) = 0$ and step $u(t)$ to 1 at time $t$

$$x(t) = \begin{cases} 0 & 0 \leq t \leq 1 \\ (\alpha/\beta)\left[1 - e^{-\beta(t-1)}\right] & t > 1 \end{cases}$$

- when $u$ is increased, $x$ tends to $\alpha/\beta$.
- Trend is exponential – gets to 98% of $\alpha/\beta$ in about $4/\beta$ time units.

# Fit to Oil Refinery Data

Set $\alpha = -0.19$, $\beta = 0.02$



# Nonconstant Coefficients

For the inhomogeneous system

$$Dx(t) = -\beta(t)x(t) + \alpha(t)u(t)$$

solution is

$$x(t) = Ce^{\int_0^t -\beta(s)ds} + e^{-\int_0^t \beta(s)ds} \int_0^t \alpha(s)u(s)e^{\int_0^s \beta(v)dv} ds$$

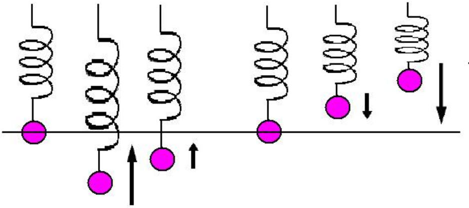
- When $\alpha(t)$ and $\beta(t)$ change slower $x(t)$ easiest to think of instantaneous behavior.
- $x(t)$ is tending towards $\alpha(t)/\beta(t)$ at an exponential rate $e^{-\beta(t)}$.

# Second Order Systems

# Second Order Systems

Physical processes often measured in terms of acceleration

We can imagine a weight at the end of a spring. For simple mechanics



$$D^2x(t) = f(t)/m$$

here the force, $f(t)$, is a sum of components

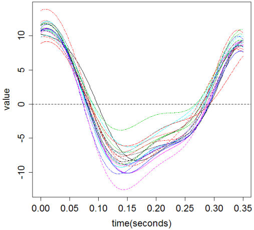
1. $-\beta_0(t)x(t)$: the force pulling the spring back to rest position.
2. $-\beta_1(t)Dx(t)$: forces due to friction in the system
3. $\alpha(t)u(t)$: external forces driving the system

Springs make good initial models for physiological processes, too.

# Lip Data

Measured position of lower lip saying the word “Bob”.

20 repetitions.



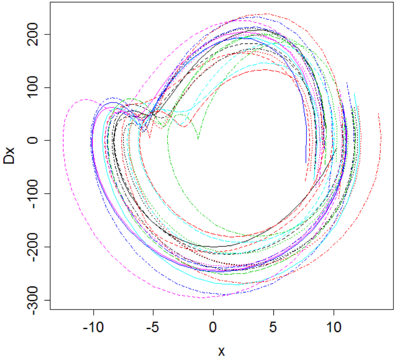

- initial rapid opening
- sharp transition to nearly linear motion
- rapid closure.

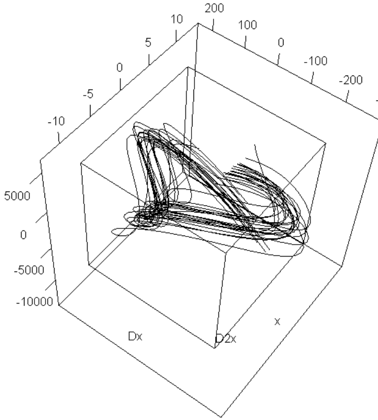
Approximate second-order model – think of lip as acting like a spring.

$$D^2x(t) = -\beta_1(t)Dx(t) - \beta_0(t)x(t) + \epsilon(t)$$

# Looking at Derivatives

Clear relationship of $D^2x$ to $Dx$ and $x$.



# The Discriminant Function

$$D^2x(t) = -\beta_1(t)Dx(t) - \beta_0(t)x(t)$$

Constant co-efficient solutions are of the form:

$$x(t) = C_1 e^{\left[-\frac{\beta_1}{2}+\sqrt{d}\right]t} + C_2 e^{\left[-\frac{\beta_1}{2}-\sqrt{d}\right]t}$$

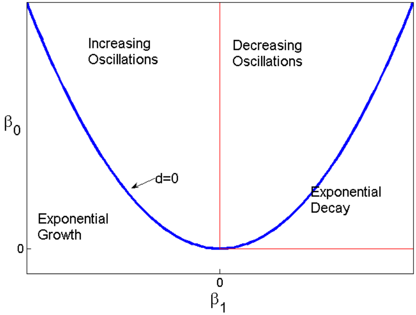
with the *discriminant* being

$$d = \left(\frac{\beta_1}{2}\right)^2 - \beta_0$$

- If $d < 0$, $e^{it} = \sin(t)$; system oscillates with growing or shrinking cycles according to the sign of $\beta_1$.
- If $d > 0$ the system is *over-damped*
  - If $\beta_1 < 0$ or $\beta_0 > 0$ the system exhibits exponential growth.
  - If $\beta_1 > 0$ and $\beta_0 < 0$ the system decays exponentially.

# Graphically

This means we can partition $(\beta_0, \beta_1)$ space into regions of different qualitative dynamics.

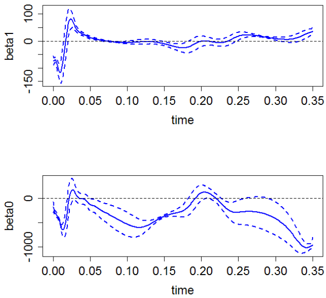



This is known as a bifurcation diagram.

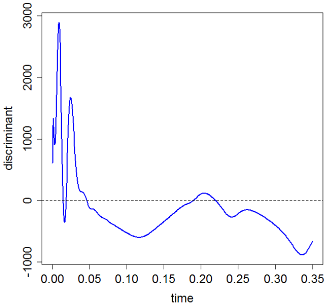
Time-varying dynamics. Like constant-coefficient dynamics at each time, if $\beta_1(t)$, $\beta_0(t)$ evolve more slowly than $x(t)$.

# Estimates From a Model

Estimated Coefficients



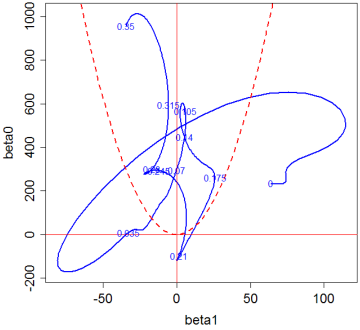

Discriminant

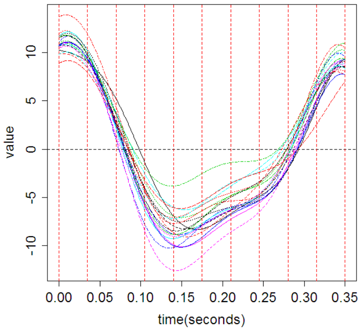



- initial impulse
- middle period of damped behavior (vowel)
- around periods of undamped behavior with period around 30-40 ms.

# On a Bifurcation Diagram

Plot $(-\beta_1(t), -\beta_0(t))$ from `pda.fd` and add the discriminant boundary.



# Principle Differential Analysis

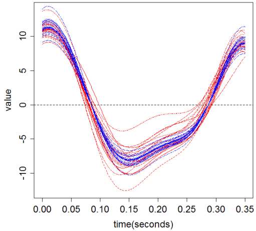
Translate autonomous dynamic model into linear differential operator:

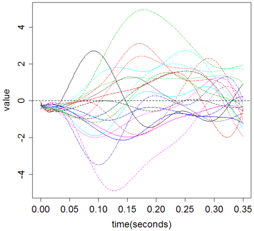
$$Lx = D^2 x + \beta_1(t) Dx(t) + \beta_0(t) x(t) = 0$$

Potential use in improving smooths (theory under development).

We can ask what is smooth? How does the data deviate from smoothness?

Solutions of $Lx(t) = 0$   Observed $Lx(t)$



# Summary

- FDA provides access to models of rates of change.
- Dynamics = models of relationships among derivatives.
- Interpretation of dynamics relies on physical intuition/analogies.
  - First order systems – derivative responds to input; most often control systems.
  - Second order systems – Newton's laws; springs and pendulums.
  - Higher-dimensional models also feasible (see special topics).
- Many problems remain:
  - Relationship to SDE models.
  - Appropriate measures of confidence.
  - Which orders of derivative to model.

# Current and Future Problems

# Correlated Functional Data

- Most models so far assume the $x_i(t)$ to be independent.
- But, increasing situations where a set of functions has its own order
  - Time series of functions.
  - Spatially correlated functions.
- We need new models and methods to deal with these processes.

# Time Series of Functions

- A functional AR(1) process

$$y_{i+1}(t) = \beta_0(t) + \int \beta_1(s,t) y_i(s) dt + \epsilon_i(t)$$

  can be fit with a functional linear model.

- Additional covariates can be incorporated, too.

- What about ARMA process etc?

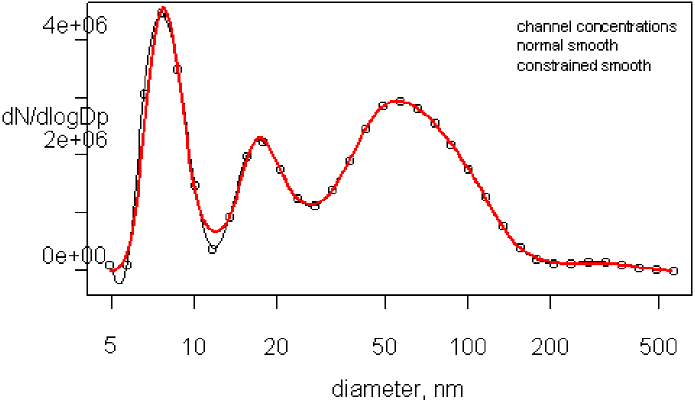
$$y_i(t) = \beta_0(t) + \sum_{j=1}^{p} \int \beta_j(s,t) y_{i-j}(s) dt + \sum_{k=1}^{q} \int \gamma_j(s,t) \epsilon_{i-k}(s) ds$$

- Are these always the best way of modeling functional time series? How do we estimate them?

# Example: Particulate Matter Distributions

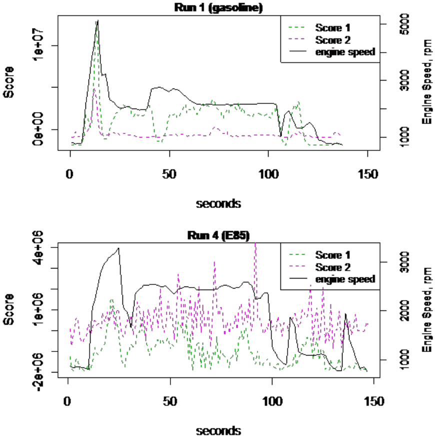
Project in Civil and Environmental Engineering at Cornell University

- Records distribution of particle sizes in car exhaust.
- 36 size bins, measured every second.



# Particulate Matter Models

First step: take an fPCA and use multivariate time series of PC scores.




Legitimate when stationary, but in presence of covariates?

# Particulate Matter Models

Possible AR models ($s$ used for “size”):

$$y_{i+1}(s) = \alpha(s) + \gamma(s)\mathbf{z}_i + \int \beta_1(u,s)y_i(u)du + \epsilon_i(s)$$

$\mathbf{z}_i$ = engine speed and other covariates

High-frequency data: should we consider smooth change over time?

$$D_t y(t,s) = \alpha(t) + \gamma(s)\mathbf{z}(t) + \int \beta_1(u,s)y_i(t,u)du + \epsilon_i(s)$$

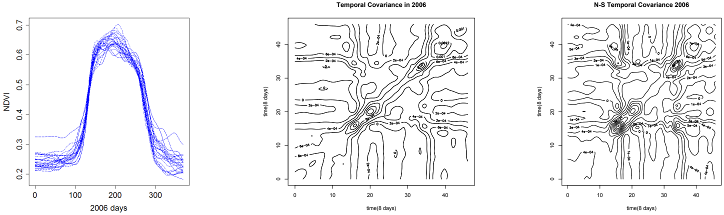
Dynamic model: how do we fit? How do we distinguish from discrete time?

# Spatial Correlation

Example: Boston University Geosciences

- $x_{ij}(t)$ gives 8-day NDVI ("greenness") values at adjacent 500-yard patches on a square.
- Interest in year-to-year variation, but also spatial correlation.

Data $x_{ij}(t)$ | $\mathrm{Var}(x_{ij}(t))$ | $\mathrm{Cov}(x_{ij}(t), x_{i(j+1)}(t))$




Required: models and methods for correlation at different spatial scales.

Liu, Ray and Hooker, 2016, "Functional Principal Components Analysis of Spatially Correlated Data", *JCGS*.

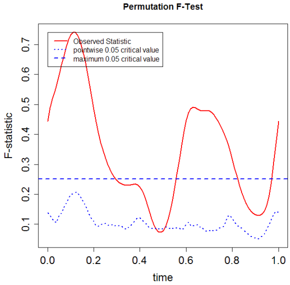
# Tests and Bootstrap

How do we test for significance of a model? Eg

$$y_i(t) = \beta_0 + \beta_1(t)x_i(t) + \epsilon_i(t)$$

Existing method: permutation tests (`Fperm.fd`)

## Permutation test for Gait model




1. Pair response with randomly permuted covariate and estimate model.
2. Calculate $F$ statistic at each point $t$.
3. Compare observed $F(t)$ statistic to permuted $F$.
4. Test based on $\max F(t)$.

# Tests and Bootstrap

Formalizing statistical properties of tests

- Some theoretical results on asymptotic normality of test statistics; esp with fPCA.
- Still requires bootstrap/permutation procedures to evaluate.
- Consistency of bootstrap for functional models unknown.
- Many possible models/methods to be considered.

# Model Selection

- Usual problem: which covariates to use?
  - Tests (see previous slide)
  - Functional information criteria.
- Also: which *parts* of a functional covariate to use?
  See James, Wang and Zhu, 2009, "Functional Linear Regression That's Interpretable", *Annals of Statistics*; Hall and Hooker, 2015, "Truncated Linear Models for Functional Data", *JRSSB*.
- Not touched: which derivative to model?
- Similarly, which derivative to register?

# Functional Random Effects

- Avoiding functional random effects a unifying theme.
- But, much of FDA can be written in terms of functional random effects.

**Eg 1: Smoothing and Functional Statistics**

$$y_{ij} = x_i(t_{ij}) + \epsilon_{ij}$$
$$x_i(t) \sim (\mu(t), \sigma(s,t))$$

Kauermann & Wegener (2010) assume the $x_i(t)$ have a Gaussian Process distribution.

Estimate $\mu(t)$, $\sigma(s,t)$ with MLE + smoothing penalty.

# Functional Random Effects

**Eg 2: Registration** re-characterized as

$$
\begin{aligned}
y_i(t) &= x_i(w_i(t)) \\
x_i(t) &\sim (\mu(t), \sigma(s,t)) \\
\log Dw_i(t) &\sim (0, \tau(s,t))
\end{aligned}
$$

- use $\log Dw_i(t)$ so that $w_i$ is monotone
  Earls and Hooker, 2017, "Adapted Variational Bayes for Functional Data Registration, Smoothing, and Prediction", *Bayesian Analysis*.
- Growth data: replace first line with acceleration?

$$
D^2 y_i(t) = x_i(w_i(t))
$$

  Model selection question!

# Functional Random Effects

**Eg 3: Accounting for Smoothing** with functional covariate

$$y_i = \beta_0 + \int \beta_1(t) x_i(t) dt + \epsilon_i$$
$$z_{ij} = x_i(t_{ij}) + \eta_{ij}$$
$$x_i(t) \sim (\mu(t), \sigma(s,t))$$

More elaborate models feasible

- Include observation process in registration.
- Linear models involving registration functions:

$$f_i = \beta_0 + \int \beta_1(t) w_i(t) dt + \zeta_i$$

- Needs numerical machinery for estimation.

Earls and Hooker, 2014, "Bayesian covariance estimation and inference in latent Gaussian process models", *Statistical Methodology*.

# Conclusions

- FDA seeing increasing popularity in application and theory.
- Much basic definitional work already carried out.
- Many problems remain open in
  - Theoretical properties of testing methods.
  - Representations of dependence between functional data.
  - Random effects in functional data.
  - Functional data and dynamics.

Still lots of room to have some fun.

# Thank You

Acknowledgements to: Jim Ramsay, Spencer Graves, Hans-Georg Müller, Oliver Gao, Darrel Sonntag, Maria Asencio, Surajit Ray, Mark Friedl, Cecilia Earls, Chong Liu, Matthew McLean, Andrew Talal, Marija Zeremkova, Luo Xiao, Peter Hall; and many others.

# Special Topics

# Smoothing and fPCA

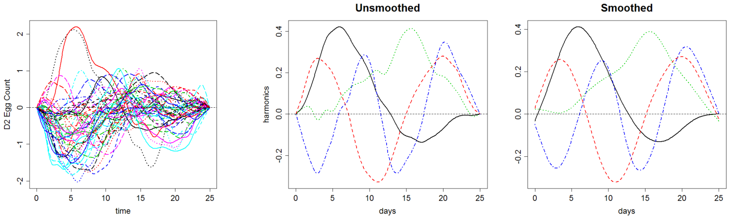
# Smoothing and fPCA

When observed functions are rough, we may want the PCA to be smooth

- reduces high-frequency variation in the $x_i(t)$
- provides better reconstruction of future $x_i(t)$

We therefore want to find a way to impose smoothness on the principal components.

## PCA of 2nd derivative of medfly data:



# Penalized PCA

Standard penalization = add a smoothing penalty to fitting criteria.

eg

$$\text{Var}\left(\int \xi_1(t)x_i(t)dt\right) + \lambda \int \left[L\xi_1(t)\right]^2 dt$$

For PCA, fitting is done sequentially – choice of smoothing for first component affects second component.

Instead, we would like a single penalty to apply to all PCs at once.

# Penalized PCA

For identifiability, we usually normalize PCs:

$$\xi_1(t) = \operatorname{argmax} \operatorname{Var} \left\{ \left[ \int x_i(t)\xi(t)dt \right] / \|\xi(t)\|_2^2 \right\}$$

To penalize, we include a derivative in the norm:

$$\|\xi(t)\|_L^2 = \int \xi(t)^2 dt + \lambda \int \left[L\xi(t)\right]^2 dt$$

Search for the $\xi$ that maximizes

$$\frac{\operatorname{Var}\left[\int \xi(t)x_i(t)dt\right]}{\int \xi(t)^2 dt + \lambda \int \left[L\xi(t)\right]^2 dt}$$

Large $\lambda$ focusses on reducing $L\xi(t)$ instead of maximizing variance.

# Choice of $\lambda$

Equivalent to leave-one-out cross validation: try to reconstruct $x_i$ from first $k$ PCs

- Estimate $\hat{\xi}_{\lambda 1}^{-i}, \ldots, \hat{\xi}_{\lambda k}^{-i}$ without $i$th observation.
- Attempt a reconstruction

$$\tilde{x}_{i\lambda}(t) = \underset{\mathbf{c}}{\text{argmin}} \int \left( x(t) - \sum_{j=1}^{k} c_j \hat{\xi}_{\lambda j}^{-i}(t) \right)^2 dt$$

- Measure

$$\text{CV}(\lambda) = \sum_{i=1}^{n} \int \left( x_i(t) - \tilde{x}_{i\lambda}(t) \right)^2 dt$$

# FDA and Sparse Data

Consider the use of smoothing for data with

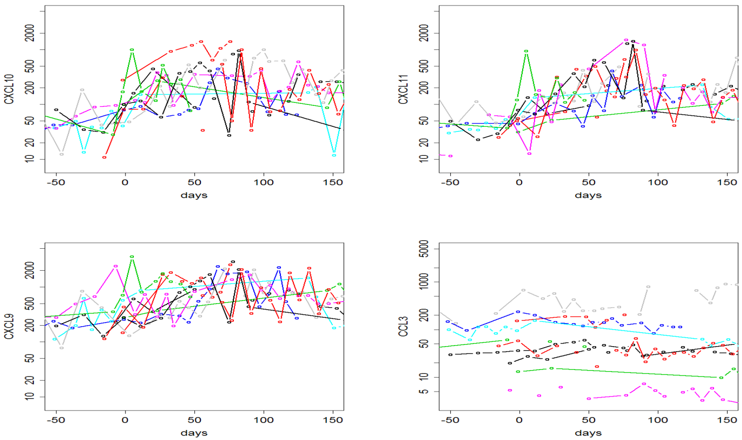
$$y_{ij} = x_i(t_{ij}) + \epsilon_i$$

with

- $t_{ij}$ sparse, unevenly distributed between records
- Assumed common mean and variance of the $x_i(t)$

# HCV Data

Measurements of chemokines (immune response) up to and post infection with Hepititis C in 10 subjects.




Sparse, noisy, high-dimensional. Aim is to understand dynamics.

# Smoothed Moment-Based Variance Estimates

(Based on Yao, Müller, Wang, 2005, JASA)

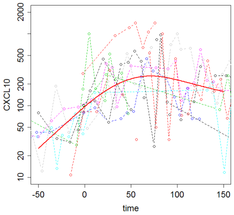
- When data are sparse for each curve, smoothing may be poor.
- But, we may over-all, have enough to estimate a covariance.
  1. Estimate a smooth $\hat{m}(t)$ from all the data pooled together
  2. For observation times $t_{ij}$, $t_{ik}$, $j \neq k$ of curve $i$ compute one-point covariance estimate

     $$Z_{ijk} = (Y_{ij} - \hat{m}(t_{ij}))\,(Y_{ik} - \hat{m}(t_{ik}))$$

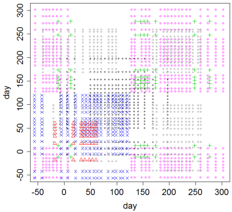
  3. Now smooth the data $(t_{ij}, t_{ik}, Z_{ijk})$ to obtain $\hat{\sigma}(s, t)$.
- PCA of $\hat{\sigma}(s, t)$ can be used to reconstruct trajectories, or in functional linear regression.

# Smoothed Moment-Based Variance Estimates

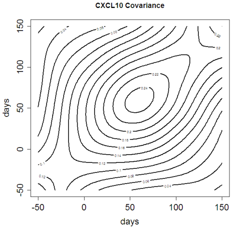
## Mean Smooth



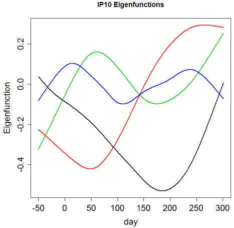

## Design



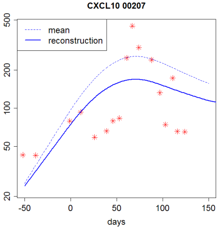

## Smoothed Covariance




## fPCA

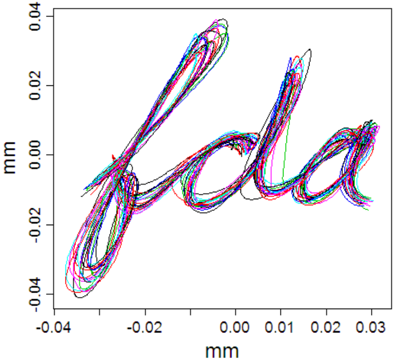



## Reconstruction




Not all subjects plotted in design.

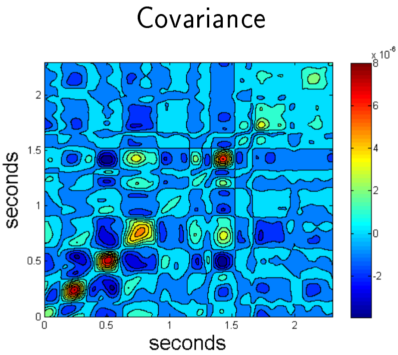
# Exploratory Analysis of Handwriting Data

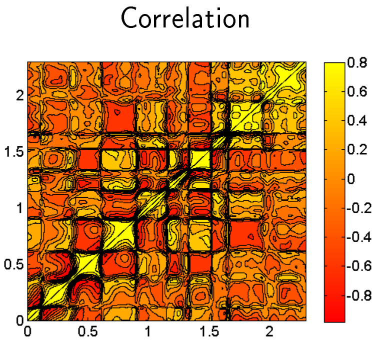


# Covariance and Correlation

Correlation often brings out sharper timing features.

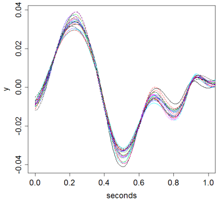
Handwriting $y$-direction:



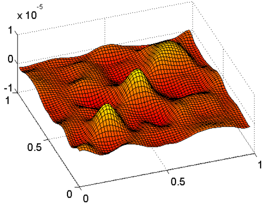



# Correlation

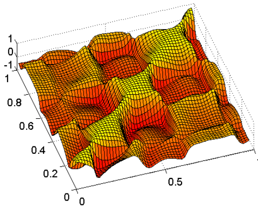
A closer look at the handwriting data




Covariance



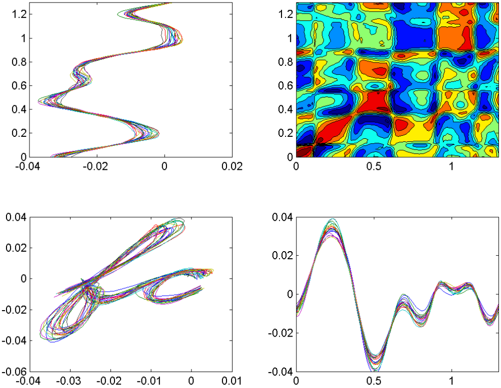

Correlation




Clear timing points are associated with loops in letters.

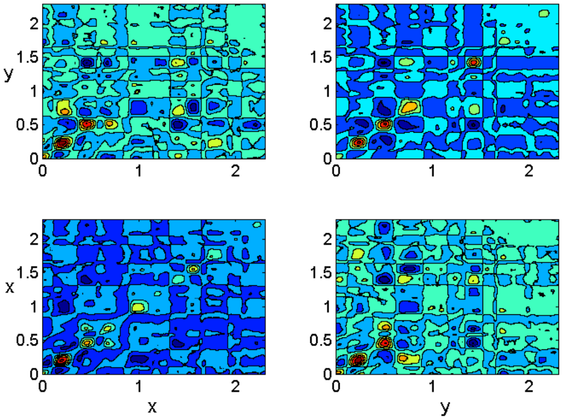
# Cross Covariance

$$\sigma_{xy}(s,t) = \frac{1}{n}\sum(x_i(s) - \bar{x}(s))(y_i(t) - \bar{y}(t))$$



# Cross Covariance

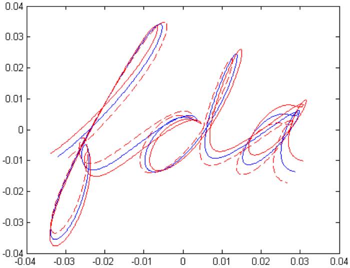
For fPCA, the distribution includes variance within and between dimensions



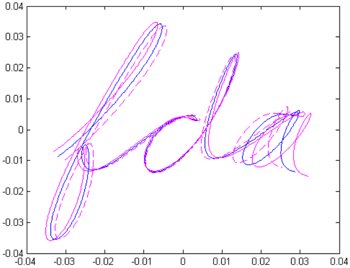
# Principal Components Analysis

Obtain the *joint* fPCA for both directions.

PC1




PC2




PC1 = diagonal spread, PC2 = horizontal spread

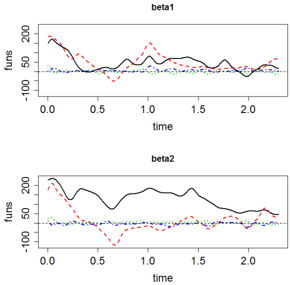
# Principal Differential Analysis

Second order model:

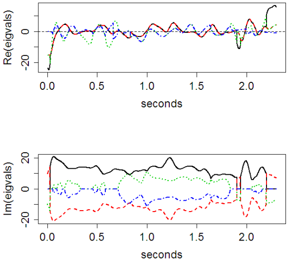
$$D^2\mathbf{x}(t) = \beta_2(t)D\mathbf{x}(t) + \beta_1(t)\mathbf{x}(t) + \epsilon(t)$$

Coefficients largely uninterpretable (may be of interest elsewhere)

Coefficient Functions




Eigenvalues




Stability analysis ⇒ almost entirely cyclic; one cycle at 1/3 second, another modulates it.

# Functional Response, Functional Covariate Models

# Functional Response, Functional Covariate

General case: $y_i(t), x_i(s)$ not necessarily on the same domain.
Multivariate model

$$Y = B_0 + XB + E$$

Generalizes to

$$y_i(t) = \beta_0(t) + \int \beta_1(s,t)x_i(s)ds + \epsilon_i(t)$$

Fitting criterion is *Sum of Integrated Squared Errors*

$$\text{SISE} = \sum \int (y_i(t) - \hat{y}_i(t))^2 \, dt$$

Same identification issues as scalar response models.

# Identification of Functional Response Model

- Need to add on a smoothing penalty for identification.
- Usually penalize $\beta_1$ in each direction separately

$$J[\beta_1, \lambda_s, \lambda_t] = \lambda_s \int [L_s \beta_1(s,t)]^2 \, dsdt + \lambda_t \int [L_t \beta_1(s,t)]^2 \, dsdt$$

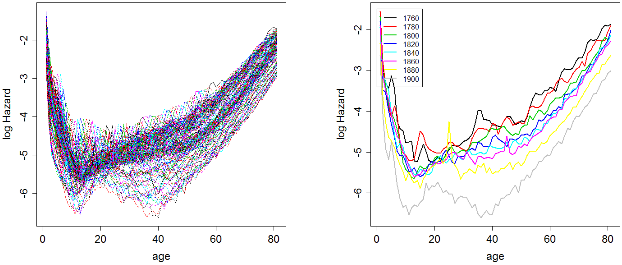
- Now minimize

$$\text{PENSISE} = \sum \int (y_i(t) - \hat{y}_i(t))^2 \, dt + J[\beta_1, \lambda_s, \lambda_t]$$

- Confidence Intervals etc follow from usual principles.
- Choice of $\lambda$'s from leave-one-curve-out cross validation.

# Swedish Mortality Data

- log hazard rates calculated from tables of mortality at ages 0 through 80 for Swedish women.
- Data available for birth years 1757 through 1900.
- Interest in looking at mortality trends.

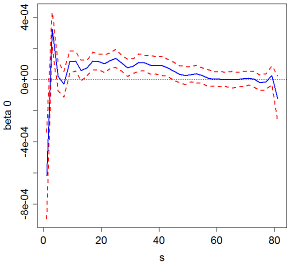



Clear over-all reduction in mortality; but effects common to adjacent cohorts?

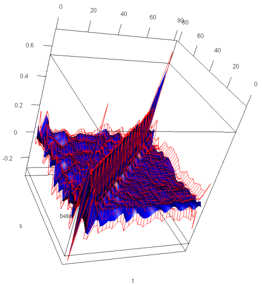
# Swedish Mortality Data

Fit a *functional auto-regressive model*:

$$y_{i+1}(t) = \beta_0(t) + \int \beta_1(s,t) y_i(s) ds + \epsilon_i(t)$$




$\beta_0$




$\beta_1(s, t)$

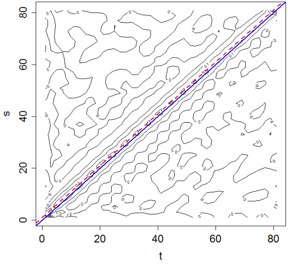
# Swedish Mortality Data

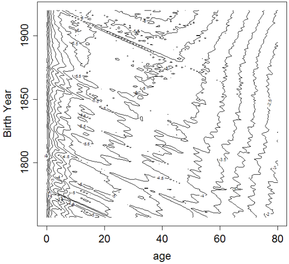
Central ridge in $\beta_1(s,t)$ one year off diagonal:

$$\int \beta_1(s,t)y_i(s)ds \approx y_i(t+1)$$

what affects one cohort, affects the next when one year younger!

$\beta_1(s,t)$ Original Data




1918 flu pandemic obvious as diagonal band.

# linmod

Produces complete functional covariate/functional response model for a single covariate.

- `yfdobj` `fd` object for response
- `xfdobj` `fd` object for covariate
- `betaList` smoothing and basis definitions for parameters
    1. `fdPar` object for $\beta_0$
    2. `bifdPar` object for $\beta_1$

Returns `beta0estfd`, `beta1estbifd` and `yhatfdobj`.

Full plotting/standard error features not yet implemented.

# Multidimensional Principal Differential Analysis

# Higher-Order and Multidimensional Systems

For dynamic analysis, second order system

$$D^2x(t) = \beta_1(t)Dx(t) + \beta_0(t)x(t)$$

reduces to multidimensional system

$$\begin{pmatrix} Dy(t) \\ Dx(t) \end{pmatrix} = \begin{pmatrix} \beta_1(t) & \beta_0(t) \\ 1 & 0 \end{pmatrix} \begin{pmatrix} y(t) \\ x(t) \end{pmatrix}$$

with $y(t) = Dx(t)$.

Can be carried on to higher-order multidimensional systems.

Still fit with original concurrent linear model (Query: is this a good idea?)

But we need to know how to analyze multidimensional systems.

# Higher-Order and Multidimensional Systems

Analysis of multidimensional systems

$$D\mathbf{x}(t) = A\mathbf{x}(t)$$

has solutions of the form

$$x_j(t) = \sum c_{ij} e^{d_i t}$$

for $d_i$ the eigenvalues of $A$.

$d_i = d_i^{Re} + i d_i^{Im}$ can be complex. Recall

$$e^{d_i t} = e^{d_i^{Re} t} \sin(d_i^{Im} t)$$

Interpretation:

- Positive real parts = exponential growth
- Negative real parts = exponential decay
- Imaginary parts = cyclic with period $2\pi / d_i^{Im}$.

Can interpret *instantaneous* qualitative behavior.

# 2nd Order Analysis of Gait Data

2nd order system to approximate cyclic motion (eg of a pendulum)

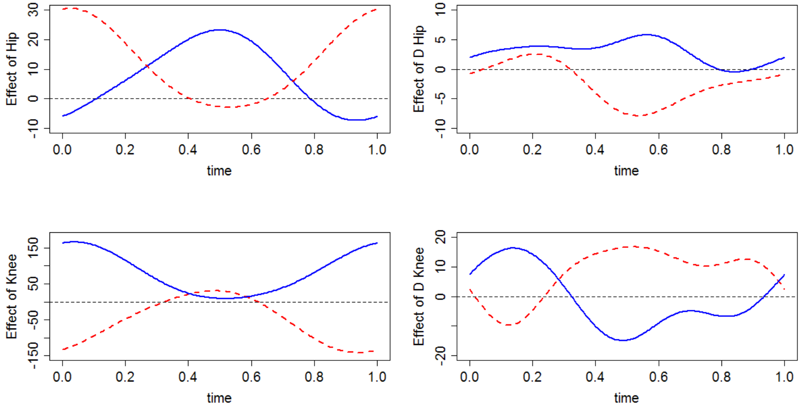
We now have a two-dimensional system

- $x$ corresponds to Hip
- $y$ corresponds to Knee

$$D^2x(t) = -\beta_{x1}(t)Dx(t) - \beta_{x0}(t)x(t) + \alpha_{x0}(t)y(t) + \alpha_{x1}(t)Dy(t)$$
$$D^2y(t) = -\beta_{y1}(t)Dy(t) - \beta_{y0}(t)y(t) + \alpha_{y0}(t)x(t) + \alpha_{y1}(t)Dx(t)$$

which we fit by the squared discrepancy from equality.

# Estimates of Coefficient Functions




Blue = influence on $D2$ Hip, Red = influence on $D2$ Knee.

Surprise = strong effect of knee angle on hip.

# Examining Stability

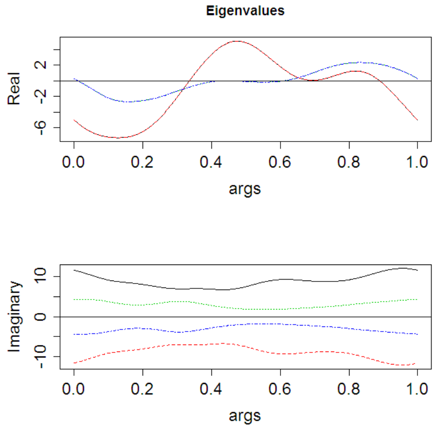
Recall that the stability of the system depends on the eigenvalues of

$$\begin{pmatrix} D^2x(t) \\ D^2y(t) \\ Dx(t) \\ Dy(t) \end{pmatrix} = \begin{pmatrix} -\beta_{x1}(t) & \alpha_{x1}(t) & -\beta_{x0}(t) & \alpha_{x0}(t) \\ \alpha_{y1}(t) & -\beta_{y1}(t) & \alpha_{y0}(t) & -\beta_{y0}(t) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} Dx(t) \\ Dy(t) \\ x(t) \\ y(t) \end{pmatrix}$$

Negative signs because we are measuring the $\beta(t)$ relative to the Lfd instead of the differential equation.

Now we can take the eigen-decomposition at each point.

# Stability Analysis




- Two magnitudes of imaginary parts – two stable cycle periods at 0.8 and 1.5 seconds.
- Mostly dissipative (negative real parts) except
- Time 0.5 = push off
- Time 0.8 = bend in knee.
- Considerably more detailed analysis possible.