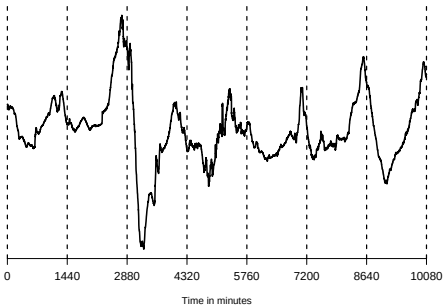
Lecture 1. Introduction

Functional Data Analysis

Jun Song

Department of Statistics
Korea University

Example: Geomagnetic Storms

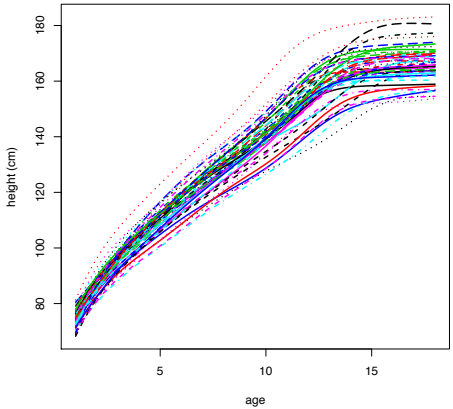


The horizontal component of the magnetic field measured at Honolulu magnetic observatory from 1/1/2001 00:00 UT to 1/7/2001 24:00 UT. The vertical dashed lines separate 24h days. Each daily curve is viewed as a single functional observation.

Example: Microsoft

Microsoft stock prices in one-minute resolution, May 1-5, 8-12, 2006. The closing price on day n is not the same as the opening price on day $n + 1$. The displayed data can be viewed as a sample of 10 functional observations.

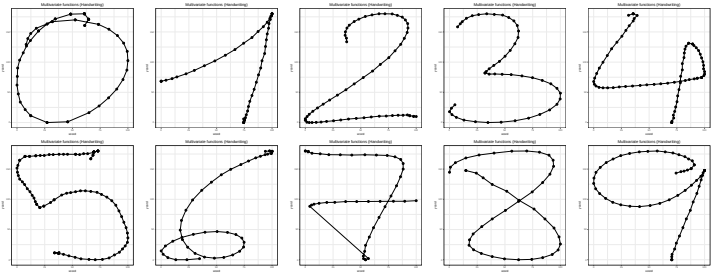
Example: Berkeley Growth Study



Heights of 54 girls in Berkeley Growth Study. Sampling times are the same across girls, but not equally spaced.

Functional Data - On-line Handwriting data

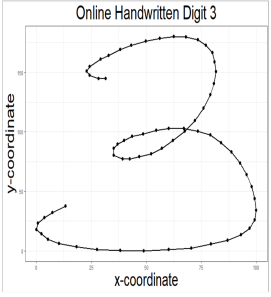
UJI Pen Characters (version 2) data set

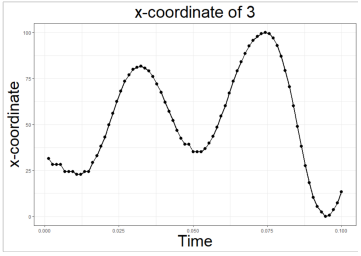


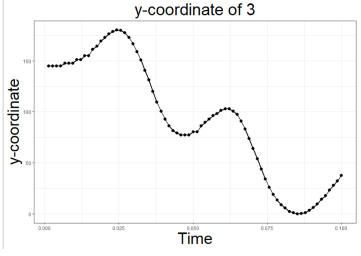
- Handwriting is collected by a touch-pad device, not a scanned image.
- The device records $x - y$ coordinates along with the time.
- Character recognition & personal handwriting style recognition.

Functional Data - On-line Handwriting data

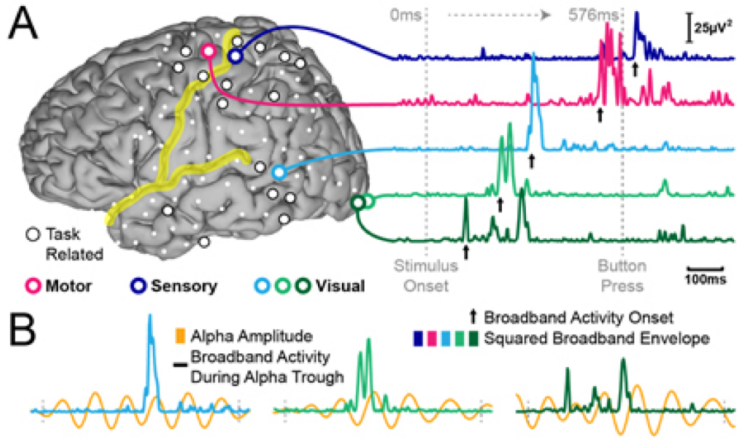
UJI Pen Characters (version 2) data set







Functional Data - fMRI



- Very noisy multivariate functional data: Do we need all?

Figure: Health Research, Inc. · Riverview Center, Menands, NY

What are the Challenges?

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

Statistical Questions/Goals

- How do we handle infinitite dimensional data? parameters?
- How do we define models which incorporate the domain?
- How do we build models which relate functional data to other variables?

Approaches to FDA

- Basis-expansion approach (Converting the data to functions)
  - Develop a method based on an object in a function space of infinite dimension
  - Data $\rightarrow$ Functional object (approximation) $\rightarrow$ Apply the method $\rightarrow$ Inference
- Direct approach
  - The method is developed directly from the discretely observed data and assumes that the dataset is realization of a random function from a function space of infinite dimension.
  - Data $\rightarrow$ Apply the method $\rightarrow$ Inference
- We will see the differences near the end of the semester.

Data Structure

We typically recieve data in the following form

$$X_n(t_{jn}) \quad n = 1, \ldots, N \quad j = 1, \ldots, J_n \quad t_{jn} \in \mathcal{T}.$$

- $\mathcal{T}$ is some closed interval of $\mathbb{R}$, usually rescaled to be $[0, 1]$.
- $X_n(t)$ is a subject/unit specific random function.
- t_{jn} is the jth observed point from unit n.
- N is the sample size.
- J_n is the number of observed points for curve n.

Converting to Function

FDA typically begins by converting the raw data into "functional objects" in R. We do this using a basis function expansion:

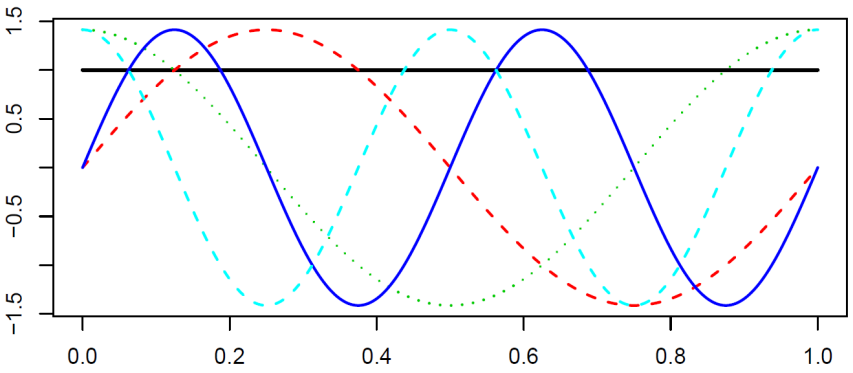
$$X_n(t) \approx \sum_{m=1}^{M} c_{nm} B_m(t).$$

The B_m are some prespecified set of basis functions. The intuition is that X_n is a smooth function, and thus can be well reprsented using some linear combination of "shapes". Also makes functions comparable if they aren't observed on a common grid.

Fourier Basis

Very commmon in mathematics, works best with periodic data.

```
library(fda)
fourier_five <- create.fourier.basis(c(0,1), nbasis=5)
plot(fourier_five,lwd=2)
```

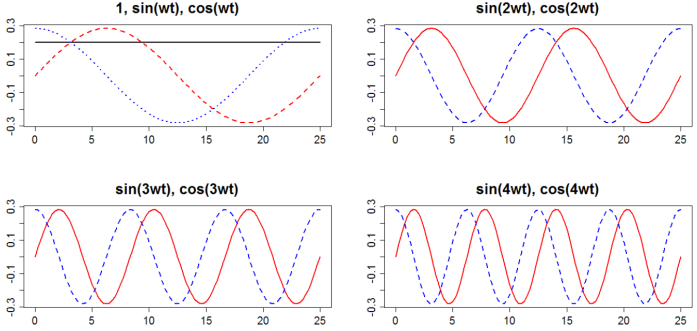


Fourier Basis Definition

- To ensure that the Fourier basis is orthonormal (more on this later), we use the set of functions

$$\left\{1, \sqrt{2}\cos(wt), \sqrt{2}\sin(wt), \sqrt{2}\sin(2 \times wt), \sqrt{2}\cos(2 \times wt), \ldots\right\}.$$

- Constant $w = 2\pi/P$ defines the period P defines the period P of oscillations of the first sine/cosine pair

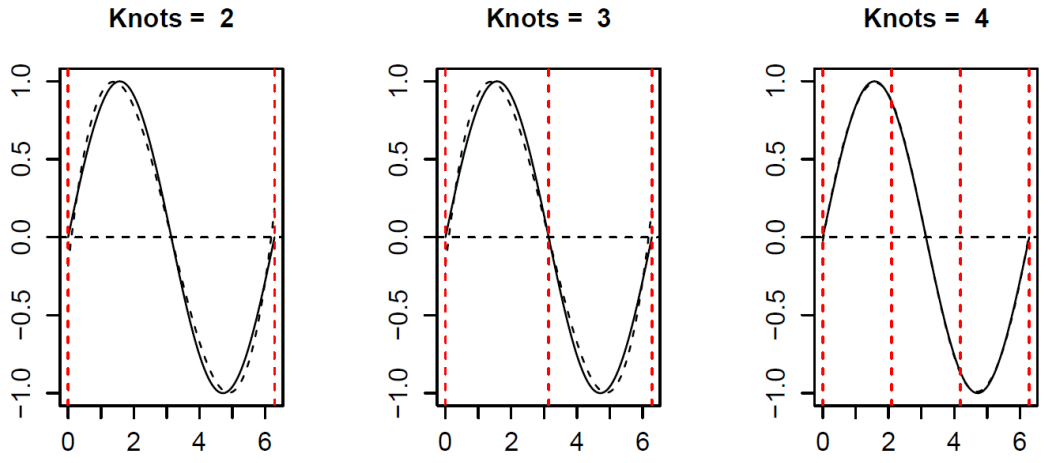


Advantages of Fourier Bases

- Excellent computational properties, especially if the observations are equally spaced.
- Natural for describing periodic data, such as the annual weather cycle and signal processing.
- Representations are periodic; this can be a problem if the data are not.

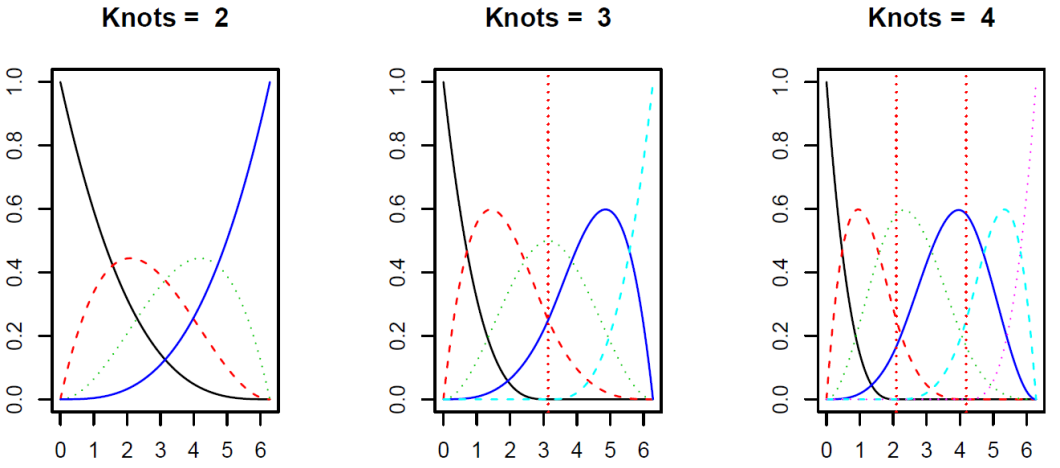
Splines

Splines are piece-wise polynomials that have been "joined" **smoothly** at "knots." Each piece is a degree 3 polynomial (cubic spline). At each knot, the spline is $3 - 1 = 2$ times differentiable.



Bsplines

Splines are actually quite easy to fit due to *Basis Splines* or *bsplines*. Fitting a spline is equivalent to a basis expansion using a particular set of basis functions.



B-spline Bases

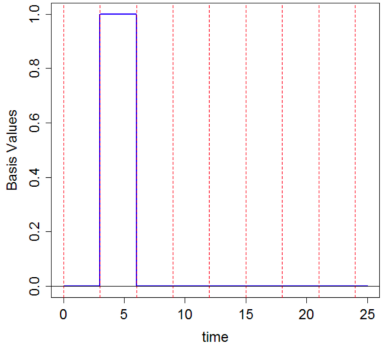
- Splines are polynomial segments joined end-to-end.
- Segments are constrained to be smooth at the joins.
- The points at which the segments join are called **knots**.
- System defined by
  - The order m (order = degree $+1$) of the polynomial
  - the location of knots
- Bsplines are a particularly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

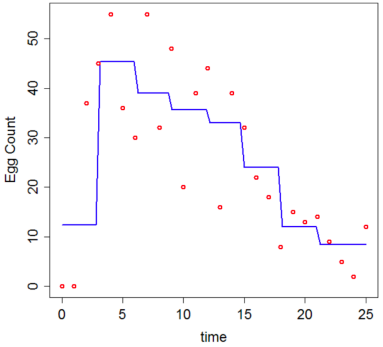
See de Boor, 2001, "A Practical Guide to Splines"

Splines: Example

Medfly data with knots every 3 days.

Splines of order 1: piecewise constant, discontinuous.

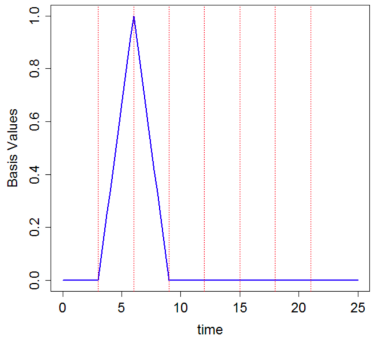


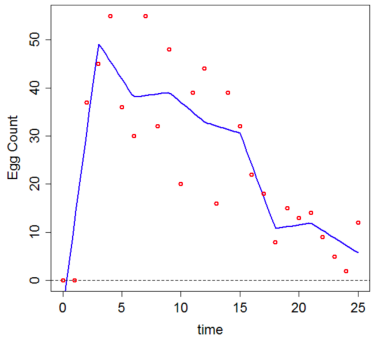


Splines: Example

Medfly data with knots every 3 days.

Splines of order 2: piecewise linear, continuous

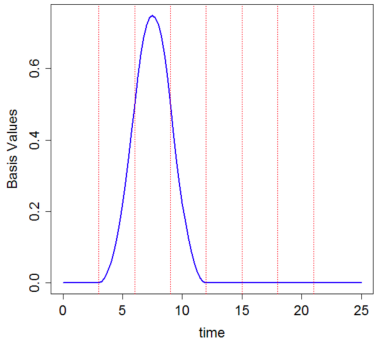


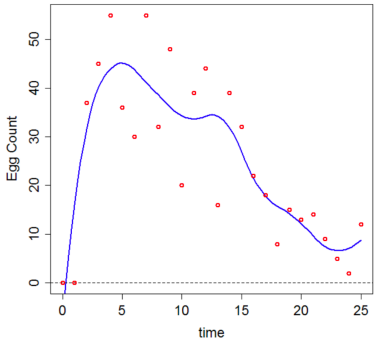


Splines: Example

Medfly data with knots every 3 days.

Splines of order 3: piecewise quadratic, continuous derivatives

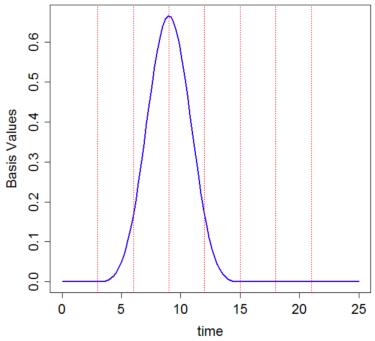


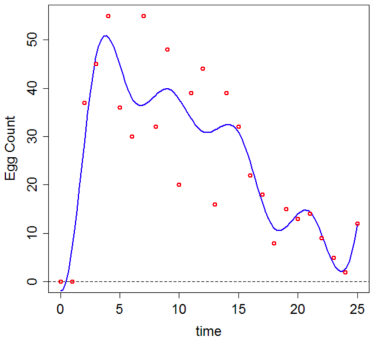


Splines: Example

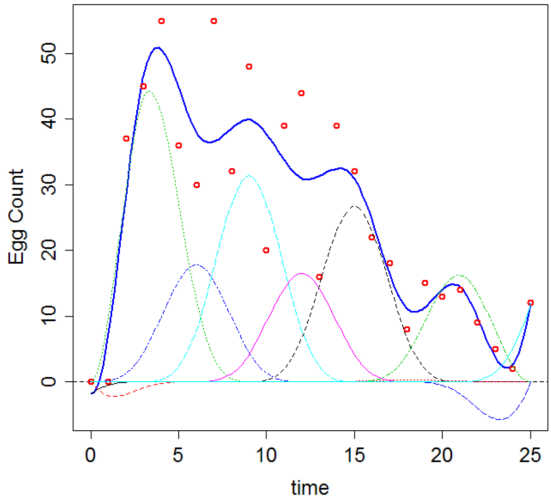
Medfly data with knots every 3 days.

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





An illustration of basis expansions for B-splines

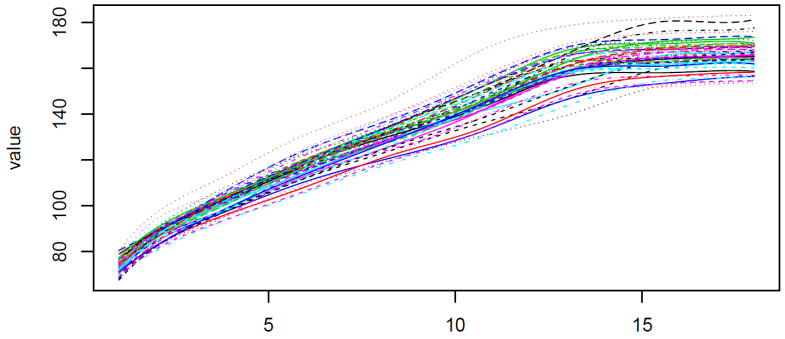


Other Bases

- There are many many other bases out there that one can use.
- However, the bsplines basis is so practically effective that it is the one most commonly used in FDA.
- For more complicated domains, such as two/three dimensional regions or manifolds, other options may turn out to be very useful.
- For example, kernel methods such as thin plate splines are very easy to generalize to complicated domains.

Constructing Functional Units

```
library(fda)
times = growth$age; GHeight = growth$hgtf
my_basis<-create.bspline.basis(c(1,18),nbasis=10)
GHeight.F<-Data2fd(times,GHeight,my_basis)
plot(GHeight.F)
```



What is a Function Object in R?

```
class(GHeight.F)

## [1] "fd"

names(GHeight.F)

## [1] "coefs"   "basis"   "fdnames"
```

- *coefs*: These are the c_{nm}, i.e. the coefficients of the basis expansion.
- *basis*: This is the basis you are using, in this case, the bsplines.
- *fdnames*: This is a list of three string vectors indicating the labels the domain (in our case time), the repetitions (in our case these are different subjects), and the value of the output (in our case heights).

Data2fd

The primary function for constructing functional objects is `Data2fd`. A second function, `smooth.fd` will be discussed later.

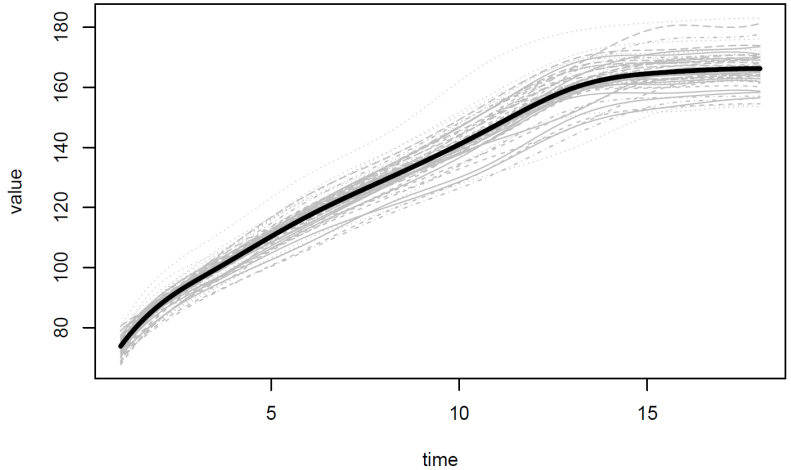
- `argvals`: Vector of domain/time points where functions are observed. This must be common to every function.
- `y`: Matrix of function values. Columns are for different functions, rows are different different arguments (opposite of whats more common in stat).
- `basisobj`: This is a user chosen basis. If you don't provide it, then R will use bsplines as the default.
- There are additional options to control the smoothness of the curves, more on this later.

What can we do with this object?

- Mean function
- Covariance function
- Principal Components
- More later on!

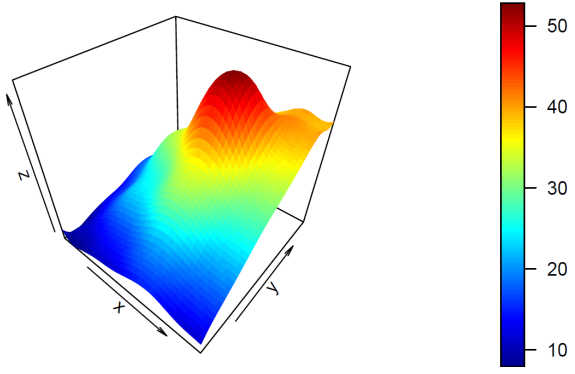
Mean Function

```
plot(GHeight.F,col="grey")
plot(mean(GHeight.F),lwd=4,add=TRUE)
```



Covariance Function

```
library(plot3D)
GHeight_var<-var.fd(GHeight.F)
pts<-seq(from=1,to=18,length=50)
GHeight_mat = eval.bifd(pts,pts,GHeight_var)
persp3D(pts,pts,GHeight_mat)
```



Functional Principal Components

```
GHeight_pca<-pca.fd(GHeight.F, nharm=2)
plot(GHeight_pca$harmonics)
```