STAT 597: Functional Data Analysis
Week 01 – Day 01 – Lecture 01
Course Overview and Introduction

Instructor: Matthew Reimherr

Outline

- Examples of functional data
- Course structure
- First concepts in FDA and some basic computation

What is Functional Data?

- *Data which take values in an infinite dimensional (function) space*, Ferraty and Vieu, 2006.
- *Data consisting of functions defined over some domain*, Hováth and Kokoszka, 2012
- *Multivariate data generalized from finite to infinite dimensions*, Jin-Ting Zhang, 2013.
- *Analysis of data varying over a continuum*, Some dude on Wikipedia

Our Perspective

Statistical analysis of samples of functions over some domain which may or may not be completely observed.

- FDA is more than multivariate anlaysis as the data may not be observed on a common grid, and the data/parameters are strongly tied to the domain (i.e. smoothness).
- FDA is more than nonparametric smoothing as we have multiple curves each of which is unit/subject specific. Furthermore these curves are often (pratically) completely observed.

Example: Geomagnetic Storms

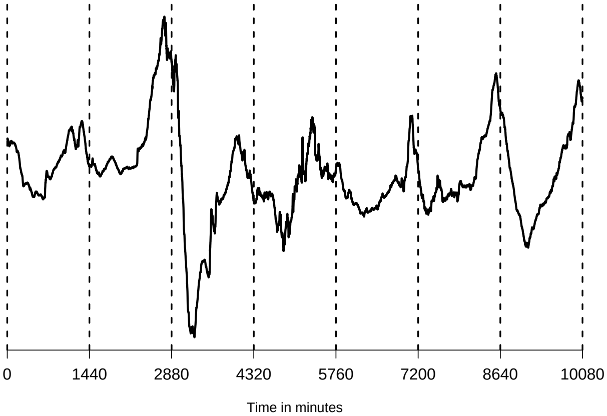


Figure: The horizontal component of the magnetic field measured at Honolulu magnetic observatory from 1/1/2001 00:00 UT to 1/7/2001 24:00 UT. The vertical dashed lines separate 24h days. Each daily curve is viewed as a single functional observation.

Example: Microsoft

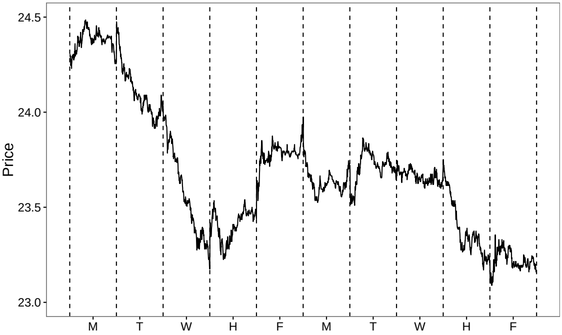


Figure: Microsoft stock prices in one-minute resolution, May 1-5, 8-12, 2006. The closing price on day n is not the same as the opening price on day $n+1$. The displayed data can be viewed as a sample of 10 functional observations.

Berkeley Growth Study

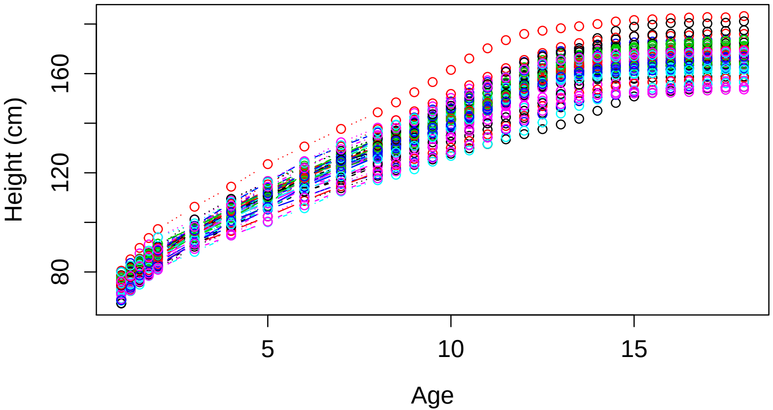


Figure: Heights of 54 girls in Berkeley Growth Study. Sampling times are the same across girls, but not equally spaced.

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?

Answers coming soon!

Course Structure

- Tuesday + Thursday
- 9:05-10:20am
- Thomas 110
- No book, notes will be provided
- Homework + Class project only
- No listed prereqs, but you should have a good grasp of graduate level stat
- Webpage is through Canvas

Course Structure - Homework

- Every other week
- Due on Fridays
- No late homework
- Mix of computational, methodological, and theoretical problems
- 50% of grade
- Homework MUST be written in Sweave, Knitr, or R markdown.

Course Structure - Project

Individual Class Project: Can be a novel data application, discussion of a methodology not covered in class, or the development of an R package for some functional data tool (either from class or not). You are graded on three categories.

- Midterm Report (20%)
- Final Report/Product (30%)
- Short In-Class Presentation

Course Topics

- First steps
- Mathematical framework
- Statistical inference
- Functional regression
- Generalized functional regression
- Sparse functional data
- Selected further topics (time series/spatial)

Chapter 1
First Steps in FDA

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:

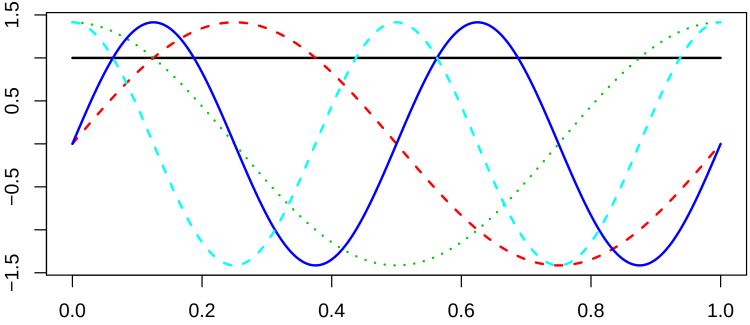
$$X_n(t) \approx \sum_{m=1}^{M} c_{nm} B_m(t).$$

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.
Number of basis functions is always odd.

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



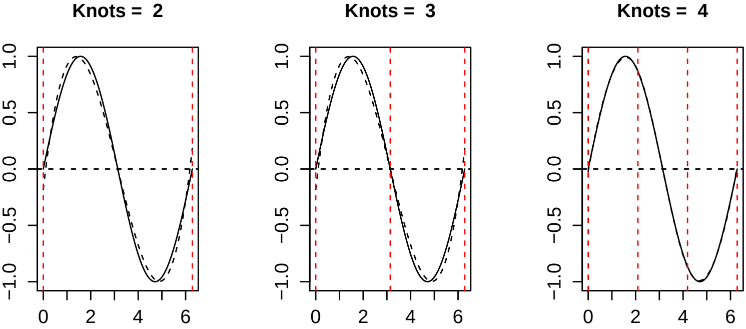
Fourier Basis Definition

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

$$\left\{1, \sqrt{2}\cos(2\pi t), \sqrt{2}\sin(2\pi t), \sqrt{2}\cos(2 \times 2\pi t), \dots \right\}.$$

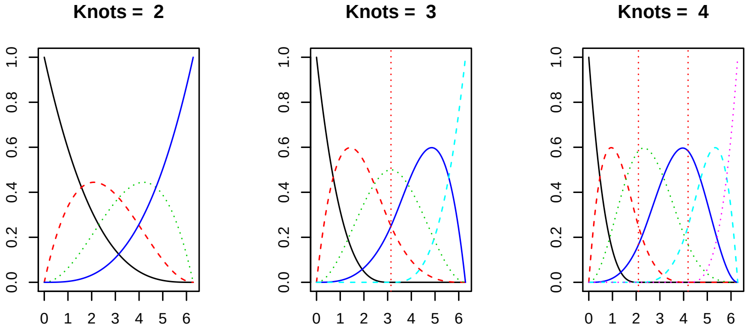
Splines

Splines are piece-wise polynomials that have been “joined” smoothly at “knots.” Each piece is a 3rd order 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.



Bsplines Definition

Fourier bases are orthonormal while bsplines are not. Fourier are also much easier to write down, while bsplines are defined via a recursion. Let $t_0 = 0 < t_1 < \cdots < t_m = 1$ denote the $m+1$ knots. Then the first order bsplines are given by

$$B_{i+1}^{1}(t) = \begin{cases} 1 & t_i \leq t < t_{i+1} \\ 0 & o.w. \end{cases} \qquad i = 0, 1, \ldots.$$

The jth order bsplines are given by

$$B_{i+1}^{j}(t) = \frac{t - t_i}{t_{i+j} - t_i} B_i^{j-1}(t) + \frac{t_{i+j} - t}{t_{i+j} - t_i} B_{i+1}^{j-1}(t) \qquad i = 0, 1, \ldots.$$

If there are $m+1$ knots, one obtains m first order bsplines, $m+1$ second order, and, in general $m+j-1$ bsplines for order j (note $j-1$ is often called the degree of the bspline). Put another way

$$\#bsplines = order + \#knots - 2.$$

Bsplines Example

There is some noational convention that we have skipped, so lets work out an example. Suppose that there are only two knots $t_0 = 0$ and $t_1 = 1$. The there is only one first order bspline which is just 1 over the entire domain:

$$B_1^1(t) \equiv 1.$$

For the second order bspline, the first function is a linear combination of $B_0^1(t)$ and $B_1^1(t)$, since there is no $B_0^1(t)$ it will be treated as zero. Also, there are no knots past t_1 so any t_2, t_3 etc will be treated as 1. This means

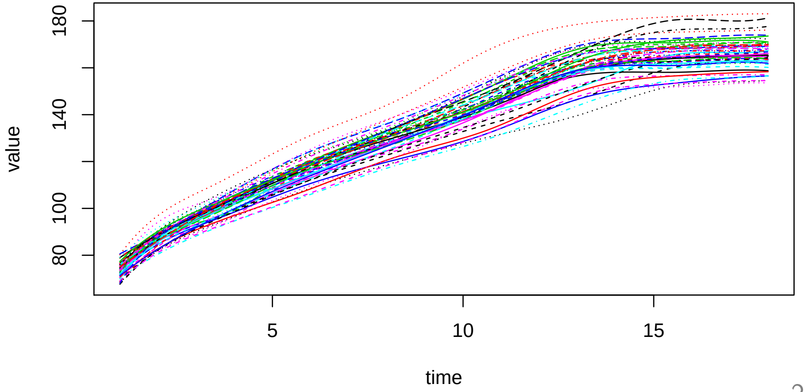
$$B_1^2(t) = 1 - t \qquad B_2^2(t) = t - 1.$$

Other Bases

There are many many other bases out there that one can use. However, the bsplines basis is so pratically 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 repititions (in our case these are different subjets), 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.

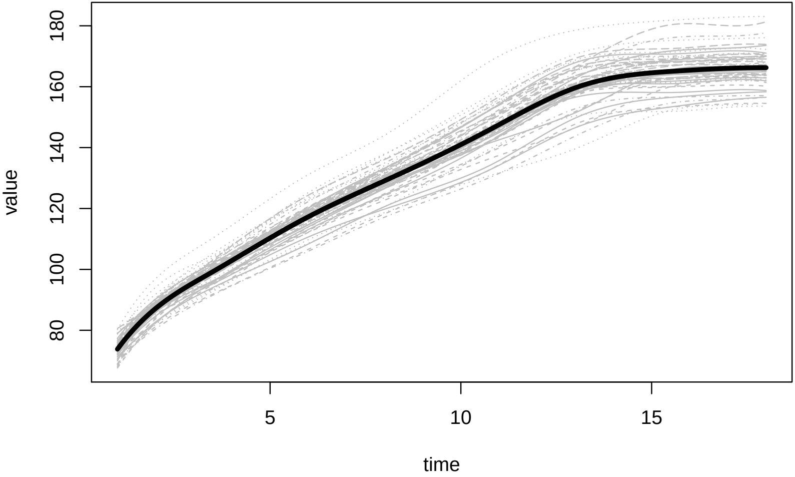
- argvals: Vector of domain/time points where functions are observed. This must be common to every function.
- y: Matrix of function values. Columns are for different functions, rows are different different arguments (opposite of whats more common in stat).
- basisojb: This is a user chosen basis. If 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
- Prinicipal Components
- More later on!

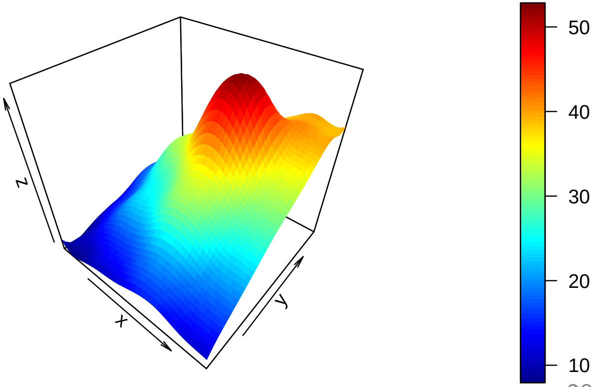
Mean Function

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



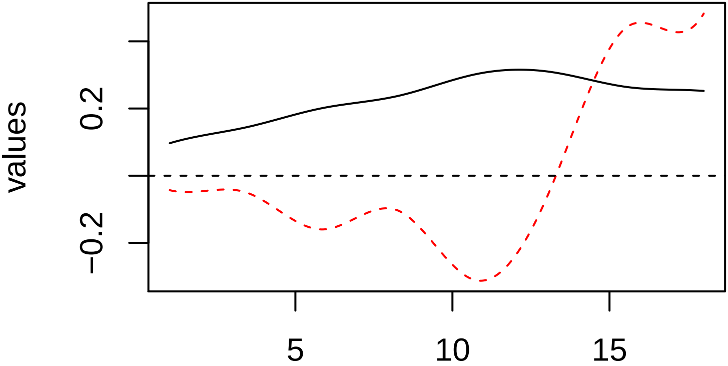
Covariance Function

```
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)
```



The first and second FPCs explain 89.3% and 6.4% of variability.