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

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

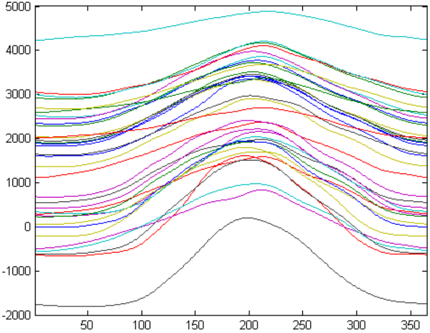
Three different scenarios

- Functional covariate, scalar response (scalar-on-function)
- Scalar covariate, functional response (function-on-scalar)
- Functional covariate, functional response (function-on-function)

We will deal with each in turn.

4.1.1 Models for Scalar Responses

Temperature curves shifted by total annual precipitation:



We want to relate annual precipitation to the *shape* of the temperature profile (see the book).

A First Idea

We observe $y_i, x_i(t)$

Choose $t_1, \ldots, t_k$

Then we set

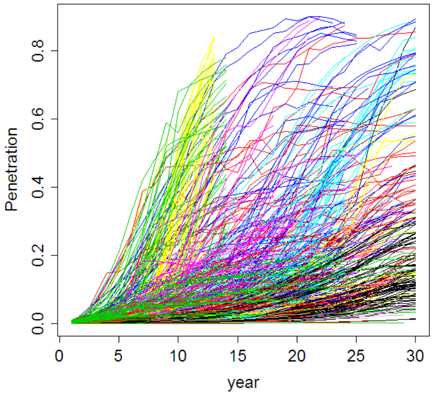
$$\begin{aligned} y_i &= \alpha + \sum \beta_j x_i(t_j) + \epsilon_i \\ &= \alpha + \mathrm{x}_i \beta + \epsilon \end{aligned}$$

And do linear regression.

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

Market Penetration Data

Yearly penetration (%) of consumer products in various countries.

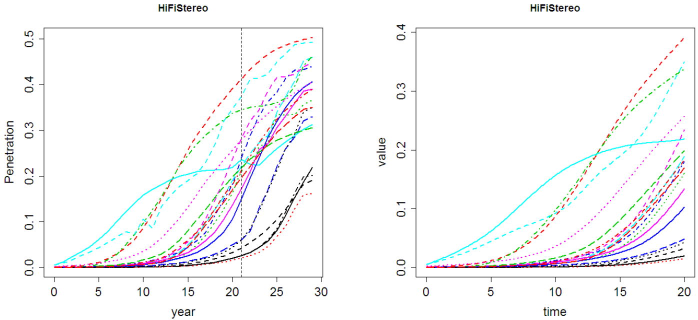


Interest in

- Forecasting
- Time to “slowdown”
- Differences between countries/products

HiFi Stereos

Introduced into 30 Countries in 1977



Attempt to forecast 2007 penetration from 1977-1997 data.

Evaluated at 6 Time Points

```
> btimes = seq(0,20,len=6)
> HiFivals = eval.fd(btimes,HiFifd)
> valmod = lm(y~t(HiFivals))
> summary(valmod)
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.20727    0.02389   8.677 9.09e-07 ***
t(HiFivals)1 24.66494   46.88618   0.526 0.607705
t(HiFivals)2 -6.29167   15.28751  -0.412 0.687369
t(HiFivals)3 -2.06591    7.78753  -0.265 0.794951
t(HiFivals)4  5.20007    3.68784   1.410 0.181998
t(HiFivals)5 -4.50342    1.80183  -2.499 0.026620 *
t(HiFivals)6  2.33180    0.52431   4.447 0.000658 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05029 on 13 degrees of freedom
Multiple R-squared: 0.8341,     Adjusted R-squared: 0.7575
F-statistic: 10.89 on 6 and 13 DF,  p-value: 0.0001987
```

In the Limit

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

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

becomes

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

Minimize squared error:

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

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.

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

I can always make the $\epsilon_i = 0$

Smoothing

We want to insist that $\beta(t)$ is smooth.

Fit by penalized squared error

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

Very much like smoothing.

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

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

Choosing A Basis

Smoothing problem

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

has a unique minimizer over all functions for which $\text{PENSSE}_\lambda(\beta) < \infty$.

- No explicit representation is available.
- However, using a rich enough basis will approximate it well.
- Usually, "rich enough" isn't too rich.
- If $x_i(t)$ are represented by a basis, using the same basis often works well.

Calculation

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

so

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

and

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

Then

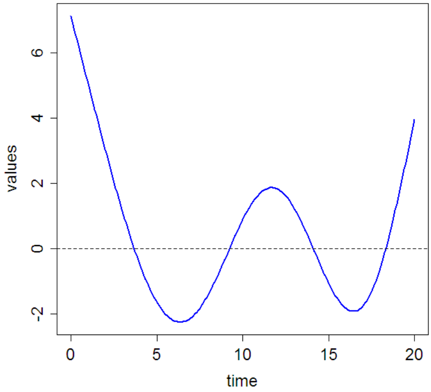
$$\hat{\mathrm{y}} = \int \hat{\beta}(t)x(t)dt = Z\begin{bmatrix}\hat{\alpha} \\ \hat{\mathrm{c}}\end{bmatrix} = S\mathrm{y}$$

HiFi Penetration

```
> cbasis = create.constant.basis(c(0,20))
> cfd = fd(matrix(1,1,20),cbasis)

> xfdlist = list(cfd,HiFifd)

> betalist = list( fdPar(cbasis,0,0), fdPar(bbasis,2,exp(-8)) )
> HiFi.reg = fRegress(y,xfdlist,betalist)

> names(HiFi.reg)
[1] "yfdPar"      "xfdlist"     "betalist"    "betaestlist"
[5] "yhatfdobj"   "Cmatinv"     "wt"          "df"

> HiFi.reg$df
[1] 6.175674

> alphahat = HiFi.reg$betaestlist[[1]]$fd$coefs
           [,1]
[1,] 0.01041727
```

HiFi Penetration

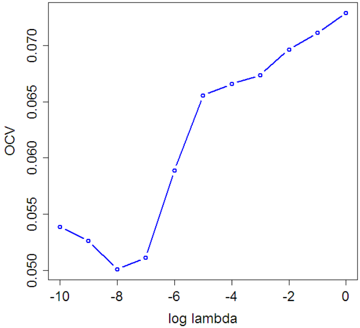
```
plot(HiFi.reg$betaestlist[[2]]$fd)
```



Choosing Smoothing Parameters

Cross-Validation:

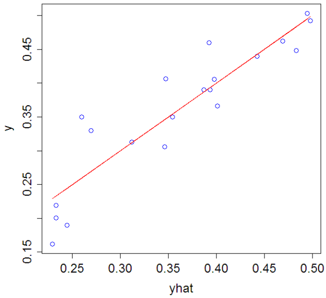
$$\mathrm{OCV}(\lambda) = \sum \left(\frac{y_i - \hat{y}_i}{1 - S_{ii}} \right)^2$$




```
> lambdas = exp(-10:0)
> ocv = rep(0,length(lambdas))

> for(i in 1:16){
>   betalist=list(fdPar(cbasis,0,0),
>       fdPar(bbasis,2,lambdas[i]) )
>   ocv[i]=fRegress.CV(y,xfdlist,betalist)
> }

> plot(-10:0,ocv)
```

Assessing Fit

```
> plot(HiFi.reg$yhatfdobj,y)
> lines(HiFi.reg$yhatfdobj,HiFi.reg$yhatfdobj)
```




```
> sigmae =  sum((y - HiFi.reg$yhatfdobj)^2)/(20-HiFi.reg$df)
> sqrt(sigmae)
[1] 0.0489183
> Rsq = 1 - sigmae/var(y)
[1] 0.7705858
```

Confidence Intervals

Following from smoothing methods we have that

$$\text{Var}\begin{bmatrix} \hat{\alpha} \\ \hat{\text{c}} \end{bmatrix} = \sigma_e^2 \left(Z^T Z + \lambda R\right)^{-1} Z^T Z \left(Z^T Z + \lambda R\right)^{-1}$$

Assuming independent

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

Estimate

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

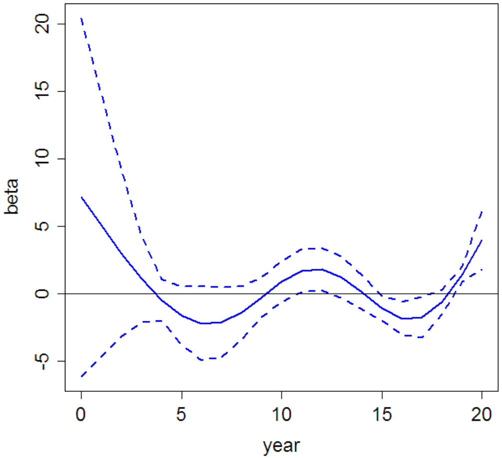
And confidence intervals are

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

Market Penetration Data

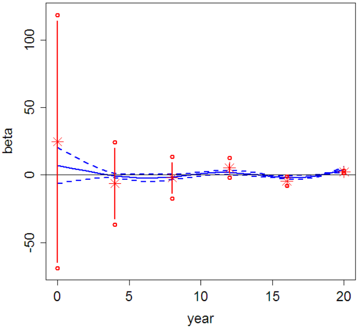
```
> HiFi.reg.std = fRegress.stderr(HiFi.reg,NULL,
        sigmae*diag(rep(1,20)))
> names(HiFi.reg.std)
[1] "betastderrlist" "bvar"           "c2bMap"


> alphahaterr = HiFi.reg.std$betastderrlist[[1]]$coefs
> c(alphahat-2*alphahaterr,alphahat+2*alphahaterr)

[1] 0.00815757 0.01267696

> betaest = HiFi.reg$betaestlist[[2]]$fd
> plot(betaest)
> lines(betaest+2*HiFi.reg.std$betastderrlist[[2]])
> lines(betaest-2*HiFi.reg.std$betastderrlist[[2]])
> abline(h=0)
```

Market Penetration Data



In Comparison

```
points(btimes,valmod$coef[2:7])
for(i in 1:6){
ci = c(valmod$coef[i+1]) + c(-2,2)*valmodstd[i+1]
lines(rep(btimes[i],2),ci)
}
```



R^2 comparisons:

0.77 (fRegress)

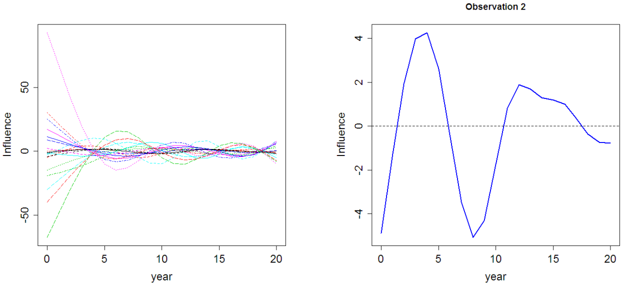
vs

0.75 (point-wise)

Assessing Influence

$$\beta(t) = \left[0 \ \Phi(t)^T\right] \left(Z^T Z + \lambda R\right)^{-1} Z^T \mathrm{y}$$

```
phivals = eval.basis(0:20,bbasis)
matplot(0:20,phivals%*%HiFi.reg.std$c2bMap[2:24,])
```



Multivariate and Mixed Functional Linear Regression

What if there are scalar covariates z and multiple functional covariates $x_1(t), \ldots, x_K(t)$?

$$y_i = \alpha + \mathbf{z}_i\gamma + \sum_{j=1}^{k} \int \beta_j(t) x_{ij}(t) dt + \epsilon_i$$

Then the penalized sum of squares is

$$\sum_{i=1}^{n} \left(y_i - \alpha - \mathbf{z}_i\gamma + \sum_{j=1}^{k} \int \beta_j(t) x_{ij}(t) dt \right)^2 + \sum_{j=1}^{K} \lambda_j \int \left[L_j \beta_j(t) \right]^2 dt$$

Multivariate and Mixed Functional Linear Regression Calculations

Take

$$\beta_j(t) = \Phi_j(t)\mathrm{c}_j$$

and

$$\zeta = \left[\alpha \ \gamma^T \ \mathrm{c}_1 \ \cdots \ \mathrm{c}_K\right]^T$$

Set

$$\mathrm{x}_{ij} = \int x_{ij}(t)\Phi(t)dt$$

and

$$Z_i = [1 \ \mathrm{z}_i \ \mathrm{x}_{i1} \ \cdots \mathrm{x}_{iK}]$$

Set

$$R_j = \int L_j\Phi_j(t)L_j\Phi)j(t)^T dt$$

$$R = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 R_1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_k R_K \end{bmatrix}$$

Then

$$\hat{\zeta} = \left(Z^T Z + R\right)^{-1} Z^T \mathrm{y}$$

Multivariate Functional Linear Regression

$$\hat{\zeta} = \left(Z^T Z + R\right)^{-1} Z^T \mathrm{y}$$

Extract $\hat{\alpha}$, $\hat{\gamma}$, $\hat{c_1}, \ldots, \hat{c_K}$ from $\hat{\zeta}$.

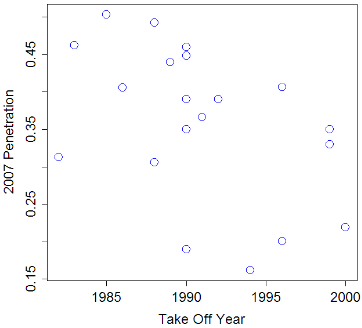
Usual statistics can be calculated:

- Cross-validation estimate
- R^2, error variance
- Co-efficient standard errors

Already used `fRegress` to include intercept in the model.

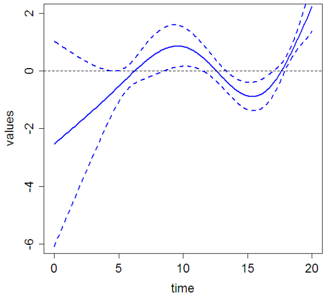
Include 'Take-Off' Time

- Measurement of when product penetration sharply increases.
- Functional of covariate function, but *nonlinear*.



Takeoff Time Effect

```
xfdlist[[3]] = fd(yr_takeoff-1977,cbasis)
betalist[[3]] = fdPar(cbasis(0,0))
HiFi.reg2 = fRegress(y,xfdlist,betalist)
```



- Residual std.err: 0.043
- Adjusted R^2: 0.8224657
- $\hat{\alpha} = -0.0005$
- Confidence interval: $[-0.00800.0070]$
- $\hat{\gamma} = 0.0005$
- Confidence interval: $[0.00020.0009]$

Summary

- Functional linear regression: move from summation to integration
- Identifiability – use a smoothing penalty (remarkable similarity to smoothing)
- Cross validation for smoothing penalty choice
- Usual diagnostic procedures
- Confidence intervals based on residual errors
- Can generalized to multiple and mixed covariates

4.1.2. Principal Components Regression

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

Usual least squares may be inappropriate if

- x is high dimensional, especially in comparison to sample size
- x is highly correlated
- main focus is on future prediction

In these cases, we would like to find some way to reduce the amount of covariate information

- Variable selection procedures (unstable)
- Principle components regression

Principal Components Regression

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

Instead, represent

$$\mathrm{x}_i = \sum_{j=1}^{p} \alpha_{ij} \xi_j$$

for ξ_j the principle components of x.

Then model

$$y_i = \beta_0' + \sum_{j=1}^{p'} \beta_j' \alpha_{ij} + \epsilon_i$$

for some $p' < p$.

Principal Components Regression

$$y_i = \beta_0' + \sum_{j=1}^{p'} \beta_j' \alpha_{ij} + \epsilon_i$$

Advantages:

- α_{ij} are uncorrelated – stability of estimates
- dimension reduction
- stable variable selection

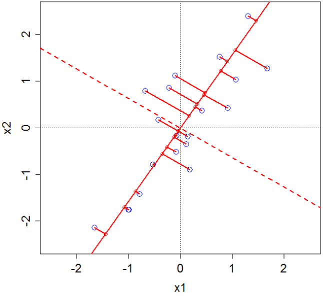
Choosing p'

- Use enough PCs to caputure 90% of variation
- Maximize adjusted R^2
- Do variable selection (ie, not necessarily leading PCs)

Principal Components Regression

$$y_i = \beta_0' + \sum_{j=1}^{p'} \beta_j' \alpha_{ij} + \epsilon_i$$

This is a bet that most variation in y is in direction of large variation in x.



Bias: variation in y due to PCs not included in the model

If $n > p$, we can consider all p Principle Components. Otherwise, we can't escape this bet.

Models for "Miss Congeniality"

Predict rating from ratings of other movies.

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6
Independence Day	-0.249	0.266	-0.103		-0.106	-0.113
Patriot	-0.259		0.308	0.484	-0.186	0.272
Day After Tomorrow	-0.317	0.135	0.119	-0.658	-0.343	0.201
Pirates Caribbean	-0.149		-0.242		-0.338	0.553
Pretty Woman	-0.232	-0.427	-0.303			-0.363
Forrest Gump	-0.113			0.286	-0.299	-0.170
The Green Mile	-0.162			0.303	-0.277	
Con Air	-0.301	0.318	-0.223		0.441	0.105
Twister	-0.302	0.138	-0.165	-0.192	-0.265	-0.489
Sweet Home Alabama	-0.295	-0.599	-0.181	-0.172	0.234	0.350
Pearl Harbor	-0.366	-0.131	0.743		0.258	-0.113
Armageddon	-0.333	0.215			0.195	
The Rock	-0.244	0.307	-0.221	0.192	0.295	
What Women Want	-0.286	-0.297	-0.111	0.183	-0.202	-0.101

```
Coefficients:
                       Estimate Std. Error  t value Pr(>|t|)
(Intercept)           3.418e-17  8.105e-03 4.22e-15 1.000000
netpca2$scoresComp.1  -2.484e-01  4.413e-03  -56.277  < 2e-16 **
netpca2$scoresComp.2  -1.914e-01  8.580e-03  -22.310  < 2e-16 **
netpca2$scoresComp.3  -1.169e-01  9.183e-03  -12.734  < 2e-16 **
netpca2$scoresComp.4  -5.635e-02  9.314e-03   -6.050 1.50e-09 **
netpca2$scoresComp.5   2.117e-02  1.017e-02    2.081 0.037426 *
netpca2$scoresComp.6   2.050e-02  1.042e-02    1.967 0.049198 *
netpca2$scoresComp.7  -2.494e-02  1.062e-02   -2.348 0.018906 *
netpca2$scoresComp.8  -6.175e-02  1.102e-02   -5.601 2.19e-08 **
netpca2$scoresComp.9   5.444e-03  1.131e-02    0.481 0.630175
netpca2$scoresComp.10 -2.448e-02  1.154e-02   -2.121 0.033928 *
netpca2$scoresComp.11  2.787e-03  1.176e-02    0.237 0.812658
netpca2$scoresComp.12  2.826e-02  1.285e-02    2.200 0.027825 *
netpca2$scoresComp.13  5.054e-02  1.315e-02    3.844 0.000122 **
netpca2$scoresComp.14 -6.644e-02  1.336e-02   -4.972 6.74e-07 **
```

Functional PCR

For functional data analysis, we can also employ principal components regression.

$$x_i(t) = \sum f_{ij}\xi_j(t)$$

And we can model

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

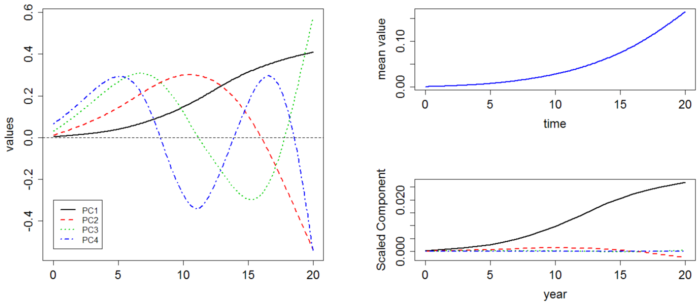
or

$$\mathrm{y} = F\beta + \epsilon$$

- reduces to a standard linear regression problem on orthogonal covariates
- avoids the need for cross-validation (assuming number of PCs is fixed)

By far the most theoretically studied method.

Principal Components of Market Data




```
> HiFi.pca$varprop
 [1] 9.275017e-01 6.184101e-02 9.802453e-03
 [3] 5.535989e-04 2.877357e-04
```

Principal Components Regression of Market Data

```
> pcamod = lm(y~HiFi.pca$scores[,1:5])
> summary(pcamod)
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              0.35925    0.01110  32.365 1.46e-14 ***
HiFi.pca$scores[, 1:5]1  0.25340    0.04328   5.854 4.19e-05 ***
HiFi.pca$scores[, 1:5]2 -0.74783    0.16763  -4.461 0.000538 ***
HiFi.pca$scores[, 1:5]3  0.93695    0.42103   2.225 0.043003 *
HiFi.pca$scores[, 1:5]4 -4.20080    1.77167  -2.371 0.032624 *
HiFi.pca$scores[, 1:5]5  3.17800    2.45745   1.293 0.216870
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.04964 on 14 degrees of freedom
Multiple R-squared: 0.8259,     Adjusted R-squared: 0.7638
F-statistic: 13.28 on 5 and 14 DF,  p-value: 6.733e-05
```

Functional Regression Interpretation

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

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

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

So we can interpret

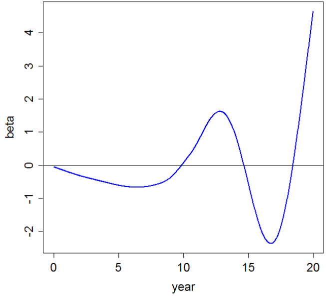
$$\beta(t) = \sum \beta_j \xi_j(t)$$

and use the model

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

HiFi Data

```
pcabeta = pcamod$coef[2]*HiFi.pca$harmonics[1]+
    pcamod$coef[3]*HiFi.pca$harmonics[2]+
    pcamod$coef[4]*HiFi.pca$harmonics[3]+
    pcamod$coef[5]*HiFi.pca$harmonics[4]+
    pcamod$coef[6]*HiFi.pca$harmonics[5]

plot(pcabeta)
```



Confidence Intervals

$$\hat{\beta}(t) = \sum \hat{\beta}_j \xi_j(t)$$

so

$$\mathrm{Var}\left[\hat{\beta}(t)\right] = [\xi_1(t) \quad \cdots \quad \xi_k(t)]\,\mathrm{Var}\left[\hat{\beta}\right] \begin{bmatrix} \xi_1(t) \\ \vdots \\ \xi_k(t) \end{bmatrix}$$

and recall that

$$\mathrm{Var}\left[\beta\right] = \sigma^2 \left(F^T F\right)^{-1}$$

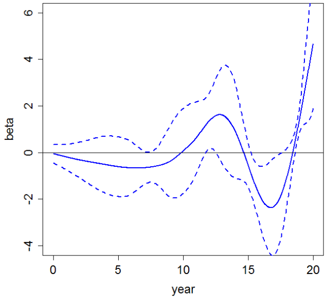
When the PCs are not smoothed, $F^T F$ is diagonal and

$$\mathrm{Var}\left[\hat{\beta}(t)\right] = \sum \mathrm{Var}\left[\hat{\beta}_j\right] \xi_j^2(t)$$

When smoothed PCs are used, $F^T F$ may no longer be orthogonal.

Confidence Intervals for HiFis

```
pcacoefvar = summary(pcamod)$coefficients[,2]^2

pcabetavar = pcacoefvar[2]*HiFi.pca$harmonics[1]^2+
    pcacoefvar[3]*HiFi.pca$harmonics[2]^2+
    pcacoefvar[4]*HiFi.pca$harmonics[3]^2+
    pcacoefvar[5]*HiFi.pca$harmonics[4]^2+
    pcacoefvar[6]*HiFi.pca$harmonics[5]^2
```

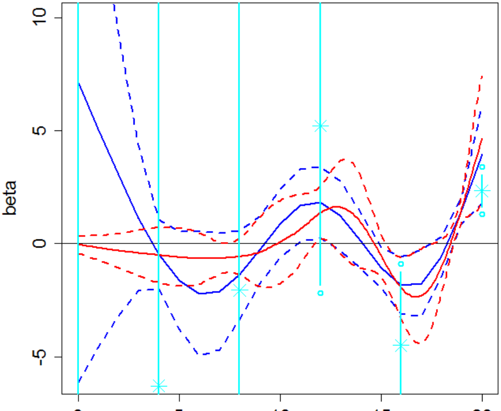



```
lines(pcabeta+2*sqrt(pcabetavar))
lines(pcabeta-2*sqrt(pcabetavar))
```

Comparisons

Adjusted R^2:

Pointwise	PCA	Smooth
0.7575	0.7368	0.7705



Summary

- Principal components regression = dimension reduction technique
- functional Principal components regression works exactly the same way
- re-interpretation as a basis expansion for $\beta(t)$
- standard errors for $\beta(t)$ calculated from linear regression covariance
- Selecting number of PCs = choice of basis
- Alternative is smoothing; performance depends on how well truth matches your assumptions.