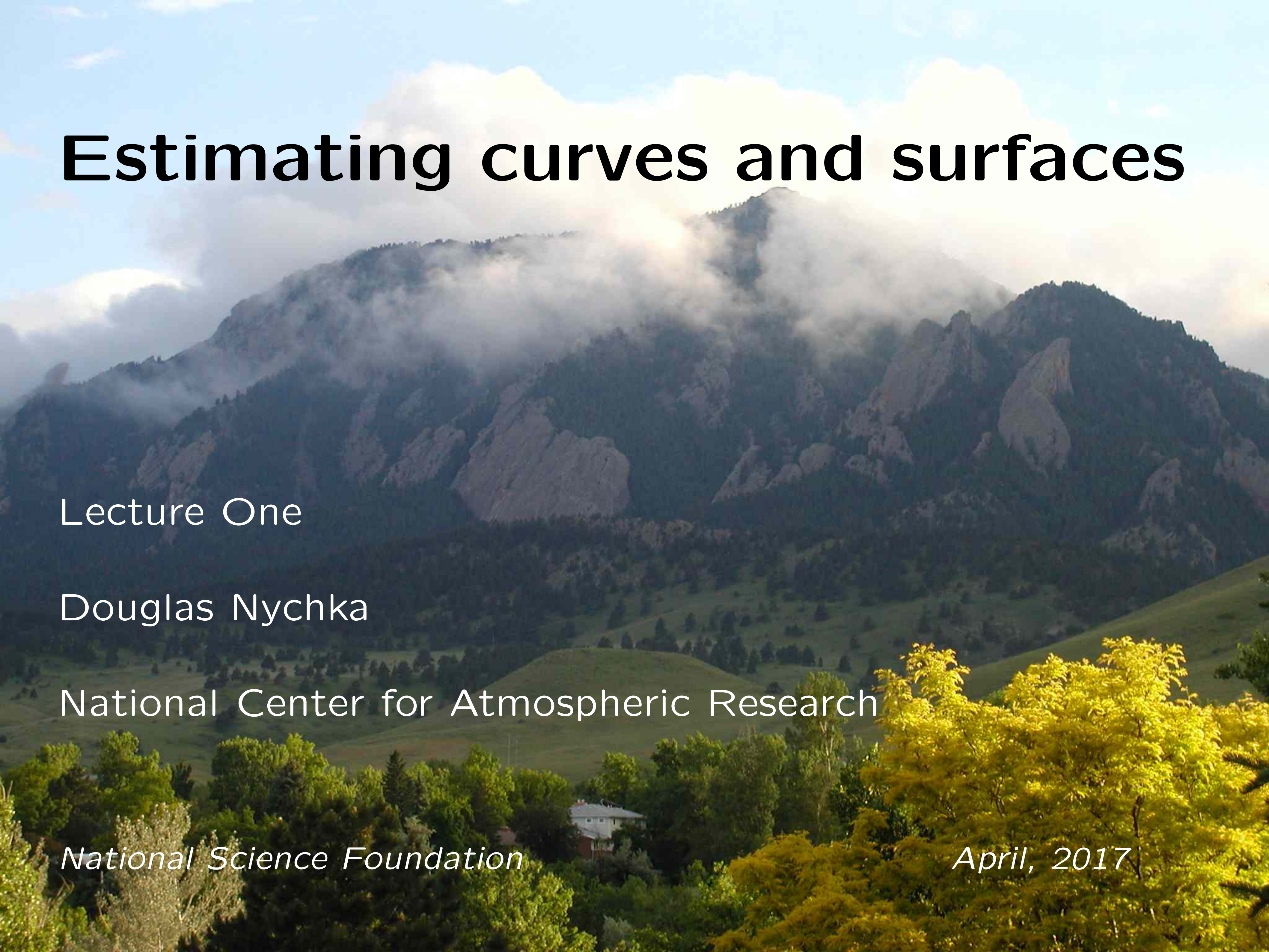


Estimating curves and surfaces

Lecture One

Douglas Nychka

National Center for Atmospheric Research

National Science Foundation *April, 2017*

These lectures

Four parts:

1. From regression to function fitting

2. Covariance models and methods for spatial data

3. Computation for large spatial data sets.

4. Inverse problems

Big ideas:

- Separate what you observe from what you want to see.
- Create a model for the unknown surface or curve
- Quantify uncertainty using a statistical model.

Outline for Lecture 1

- Basis functions
- Least squares smoothers adding a penalty
- Cross-Validation
- The Bayesian connection.

Estimating a curve or surface.

An additive statistical model:

Given n pairs of observations (x_i, y_i), $i = 1, \ldots, n$

$$y_i = g(x_i) + \epsilon_i$$

ϵ_i's are random errors and g is an unknown, smooth function.

Goals

- Estimate g based
on the observations

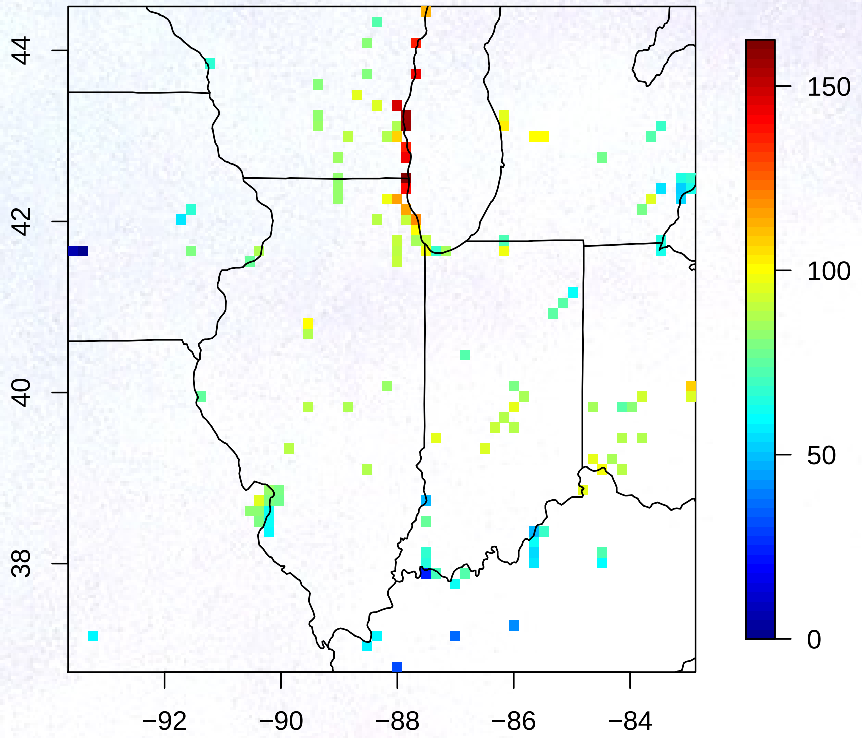
- Quantify the uncertainty in the estimate.

Some data examples

Air quality

Predict surface ozone where it is not monitored.

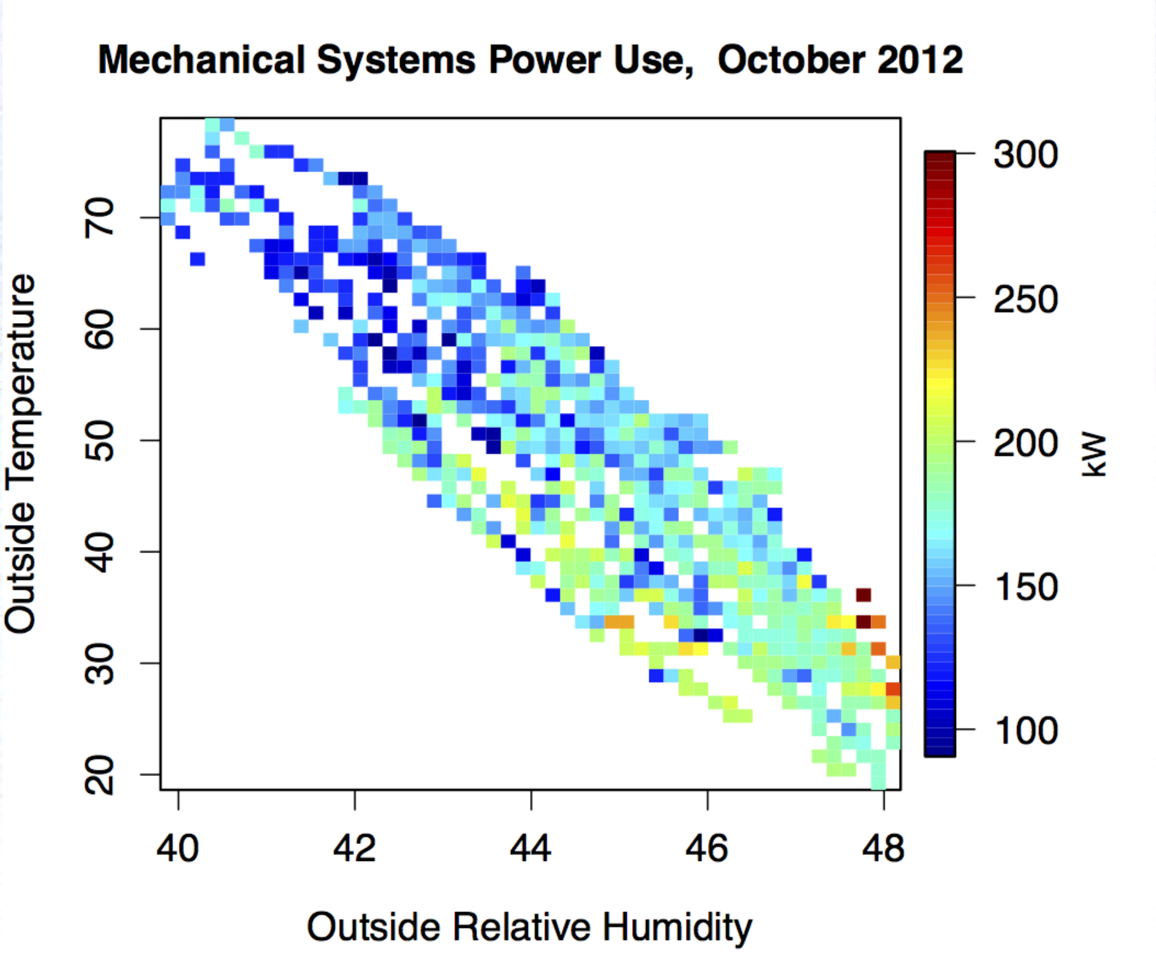
Ambient daily ozone in PPB June 16, 1987, US Midwestern Region.



Supercomputer operations

Predict power needed for cooling.

Power for mechanical systems based on temperature and humidity.

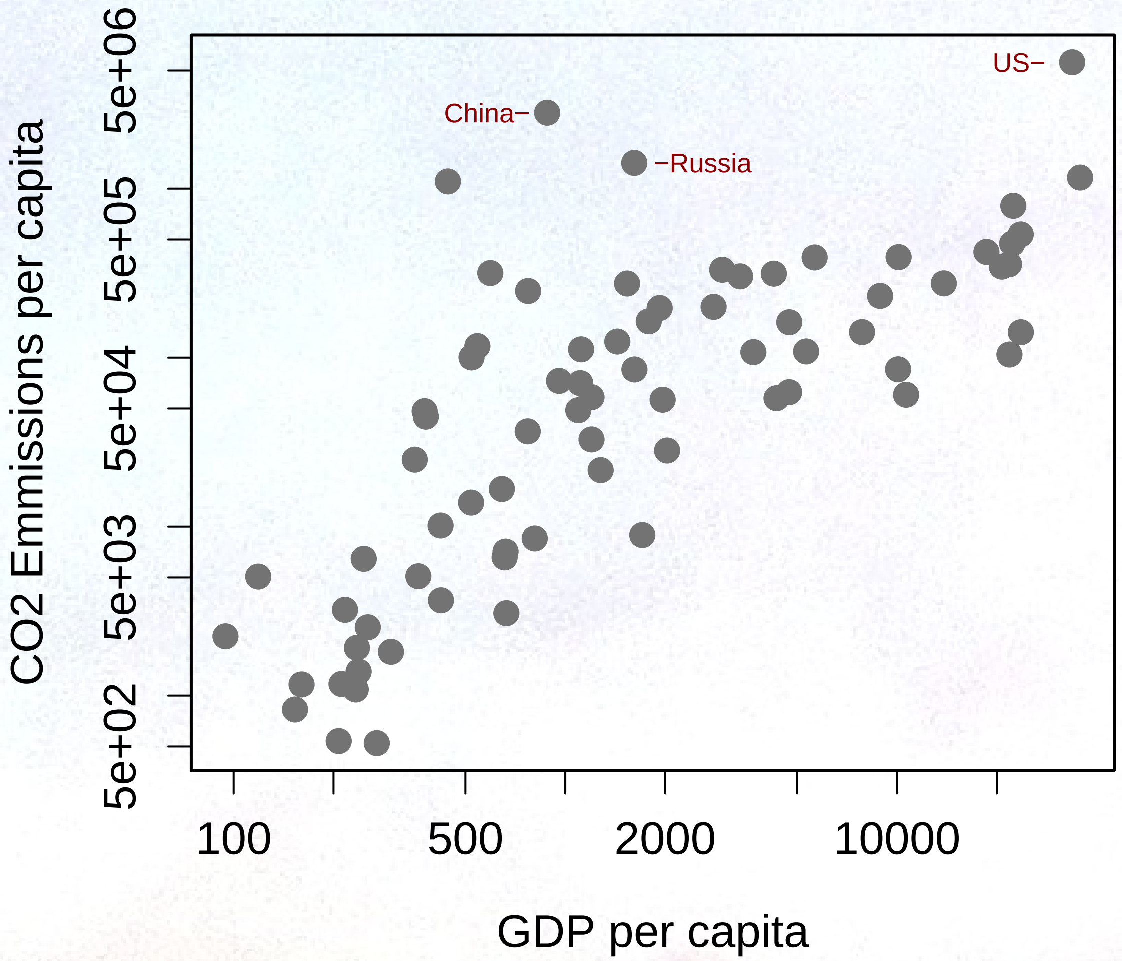


NCAR/Wy Supercomputing Center, *Yellowstone* – 72K cores, 15 Pb

Human CO2 emissions

Relationship of economics and CO_2 emissions.

CO_2 per capita by country for 1999 (Source: *World Bank*)



Basis functions and least squares.

Representing a curve

Start with your favorite m basis functions $\{b_1(x), b_2(x), \ldots, b_m(x)\}$

The estimate has the form

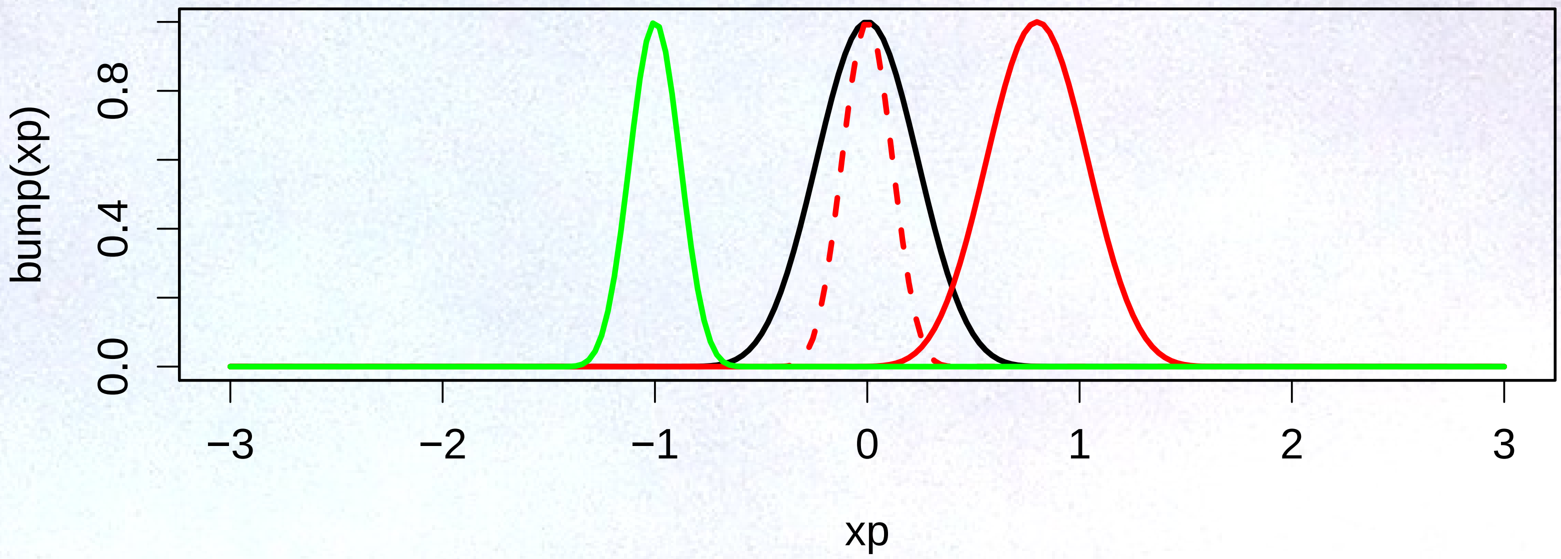
$$\hat{g}(x) = \sum_{k=1}^{m} \beta_k b_k(x)$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ are the coefficients.

The basis functions are fixed and so the problem is to just find the coefficients.

Many spatial statistics problems have this general form or can be approximated by it.

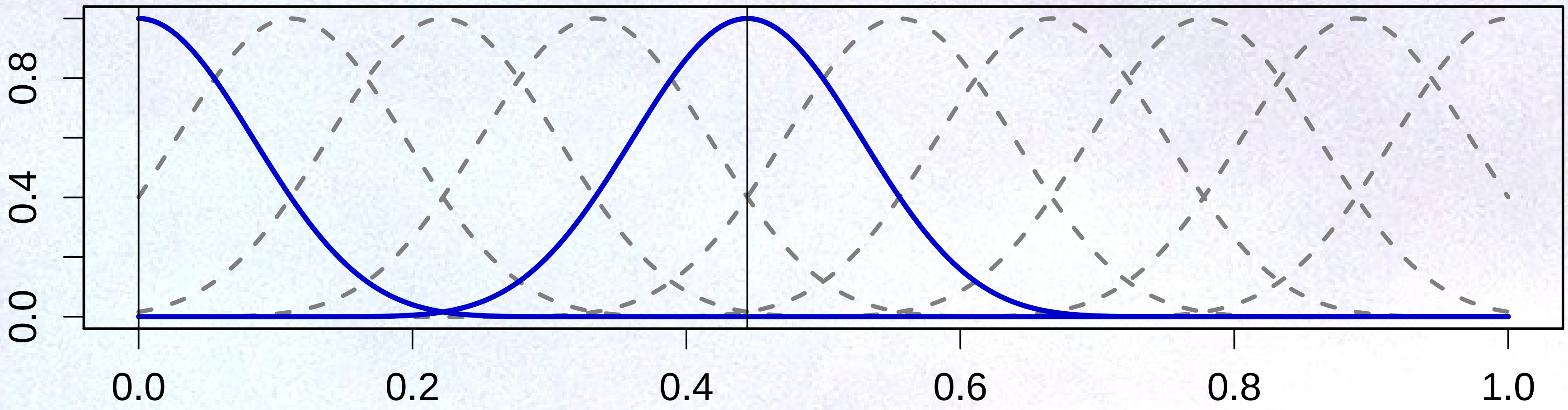
Example of basis functions



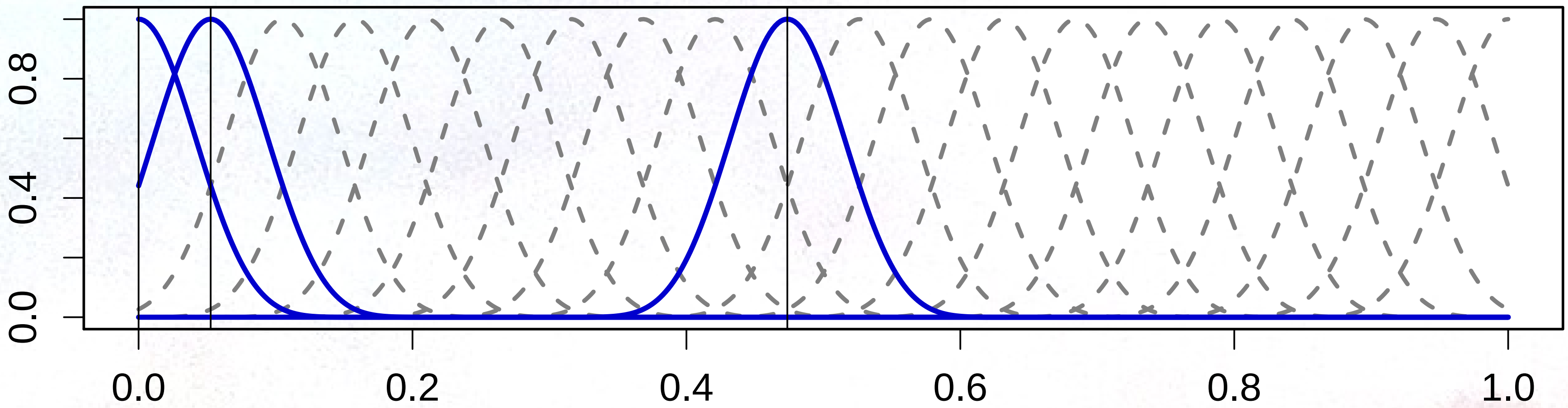
- Build a basis by translating and scaling a bump shaped curve
- Not your usual sine/cosine or polynomials!
- Bsplines not required!

Two Bases

10 Functions:



20 Functions:



Least squares.

$X_{i,j} = b_k(\boldsymbol{x}_i)$

$(\hat{g}(x_1), \hat{g}(x_2), \ldots, \hat{g}(x_n))^T = X\boldsymbol{\beta}$ and $\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$

minimize over β:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\boldsymbol{y} - [X\boldsymbol{\beta}]_i)^2$$

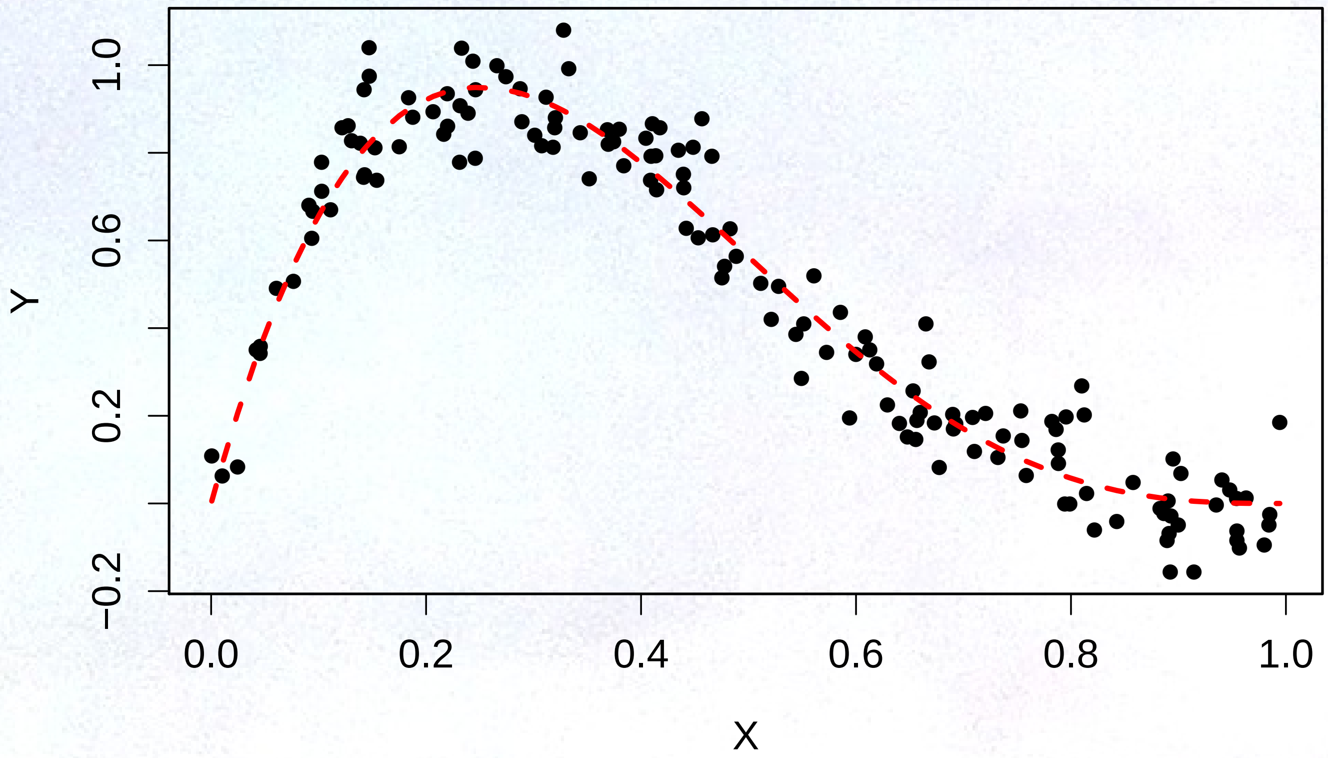
Solution:

$$\hat{\beta} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Curve/surface estimate:

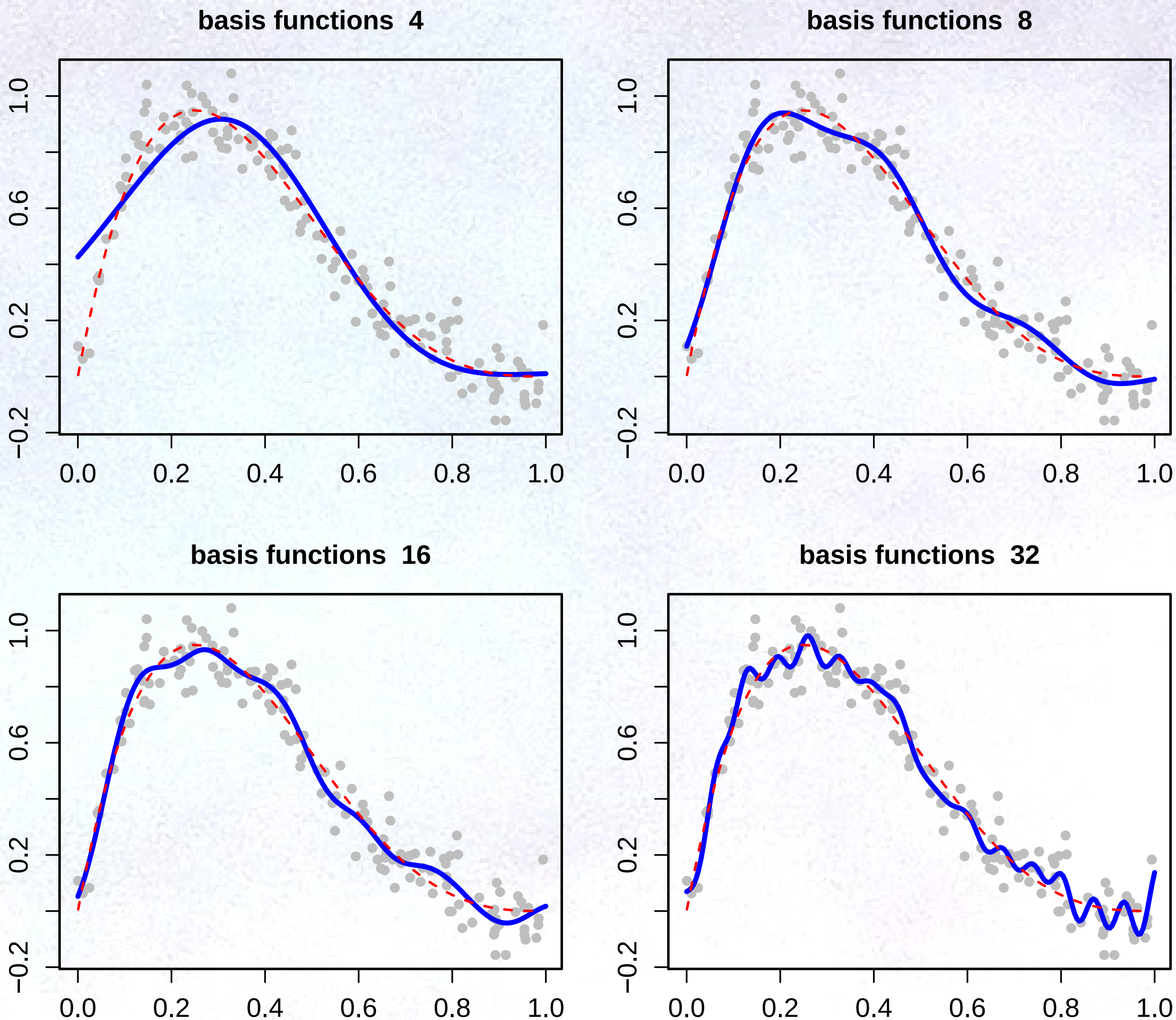
$$\hat{g}(x) = \sum_{k=1}^{m} \hat{\beta}_k b_k(x)$$

Some synthetic data: $Y_k = g(x_k) + e_k$
$g(x) = 9x(1-x)^3$



x_k are 150 unequally spaced points in $[0,1]$
$h(x) = 9x(1-x)^3$, $e_k \sim N(0, (.1)^2)$

Varying basis size



Problem: How to choose basis? How to choose *number* of basis functions? Uncertainty of estimate?

Penalized least squares

Adding a constraint on parameters

$g(x) = \sum_{k=1}^{m} \beta_k b_k(x)$

minimize over β:

Sum of squares($\boldsymbol{\beta}$) + penalty on $\boldsymbol{\beta}$

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\boldsymbol{y} - [X\boldsymbol{\beta}]_i)^2 + \lambda \boldsymbol{\beta}^T Q \boldsymbol{\beta}$$

Fit to the data + penalty for complexity/smoothness

- $\lambda > 0$ a smoothing parameter
- Q a nonnegative definite matrix.

Solution to penalized problem

$$\hat{\beta} = (X^T X + \lambda Q)^{-1} X^T \boldsymbol{y}$$

Also known as ridge regression

The Hat matrix and effective degrees of freedom

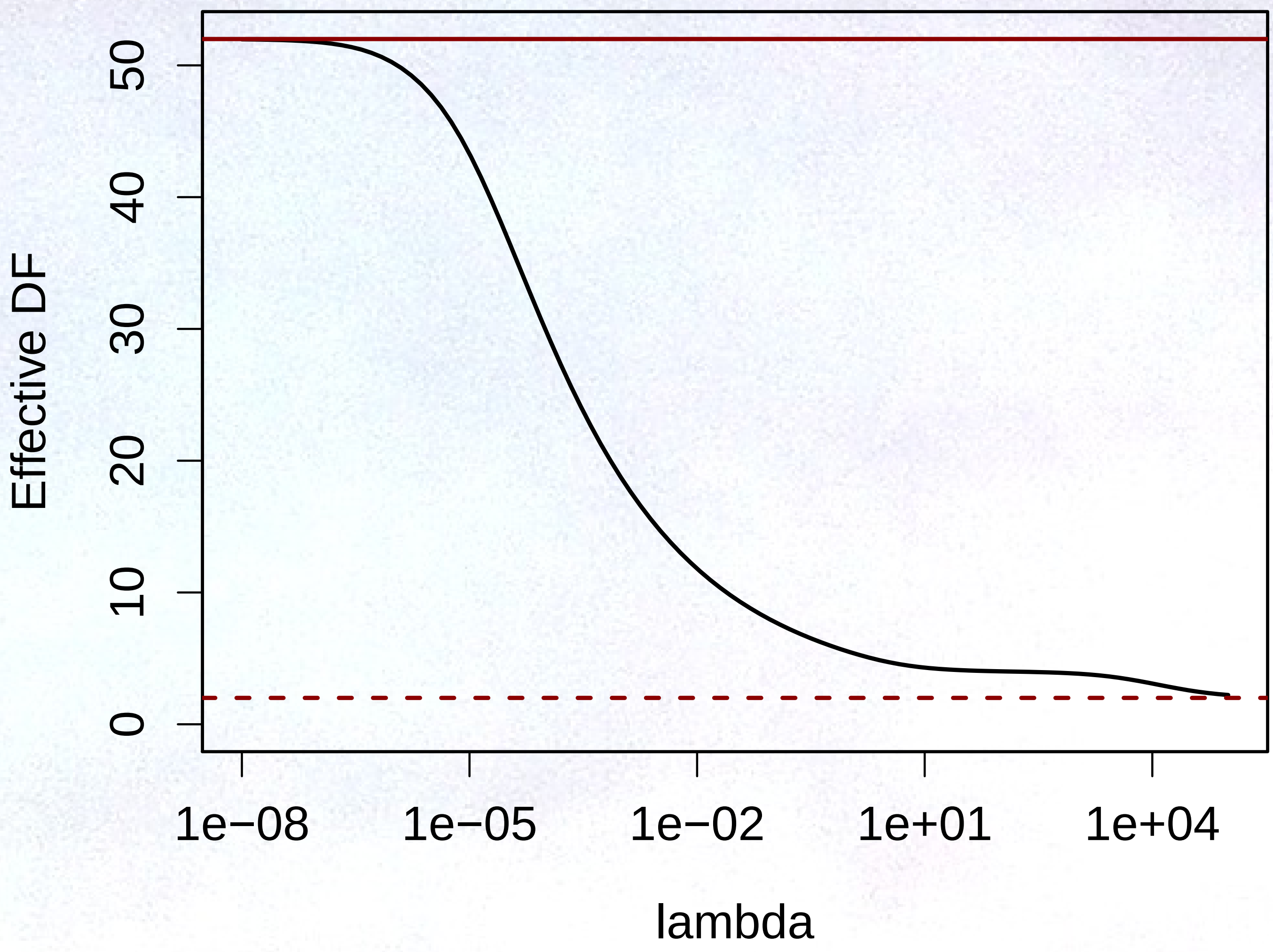
For ordinary least squares

$tr(X(X^T X)^{-1} X^T) =$ number of parameters
i.e. the column rank of X

For the penalized estimate
$\hat{g} = X(X^T X + \lambda Q)^{-1} X^T \boldsymbol{y}$

$tr(X(X^T X + \lambda Q)^{-1} X^T) =$ effective number of parameters

λ and eff. degrees of freedom



Basis: Constant, linear, and 50 bump-shaped functions.
Second derivative type penalty.

More on Q

$$\hat{\beta} = (X^TX + \lambda Q)^{-1}X^T\boldsymbol{y}$$

- Q is an identity $\boldsymbol{\beta}^T Q \boldsymbol{\beta} = \sum(\beta_l)^2$

- Choose Q so that $\boldsymbol{\beta}^T Q \boldsymbol{\beta} = \sum(\beta_l - \beta_{l-1})^2$

- Cubic smoothing spline type: $\boldsymbol{\beta}^T Q \boldsymbol{\beta} = \int (g''(x))^2 dx$

i.e. $Q_{kl} = \int b_k''(x) b_l''(x) dx$

- Spatial model: $Q^{-1}_{kl} = exp(-|x_k - x_l|/\theta)$

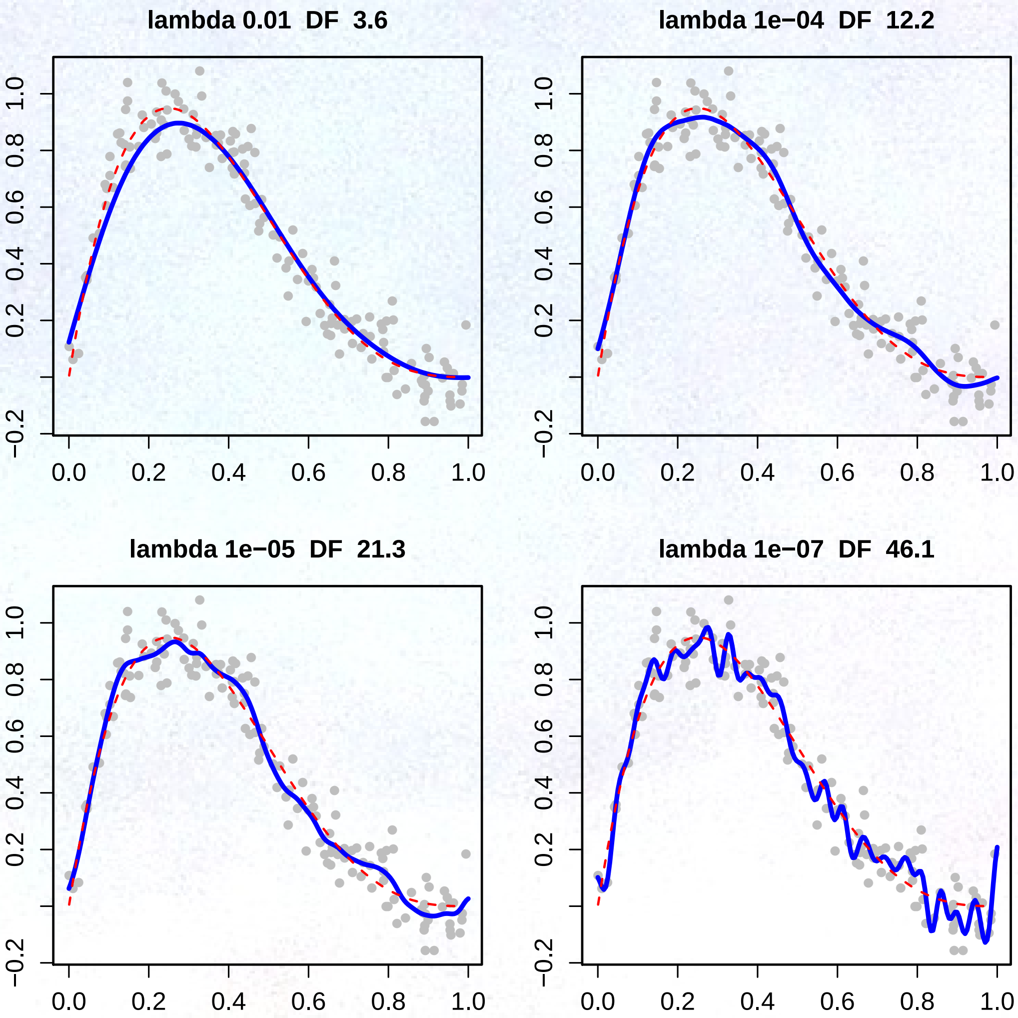
- Lasso/Wavelet:

Replace quadratic form by L_1 type criteria: $\boldsymbol{\beta}^T Q \boldsymbol{\beta} \rightarrow \sum |\beta_l|$

Varying smoothing parameter

Cubic spline-type choice with 50 basis functions

Effective number of parameters (basis functions) depends on λ.



Problem: How to choose basis? How to choose λ? Uncertainty of estimate?

Choosing the basis.

Basis selection

Reformulate the estimate in terms of the function, not the basis.

$$\min_{g} \sum_{i=1}^{n} (\boldsymbol{y}_i - g(\boldsymbol{x}_i))^2 + \lambda \int g''(x)^2 dx$$

Solution is a cubic smoothing spline.

Of course in practice we still need to represent g as

$$g(x) = \sum_{k=1}^{m} \beta_k b_k(x)$$

but this is a computational detail – not part of our model for g.

Splines

- Usually based on derivative penalites

- Basis has closed form and solves the minimization

- Cubic smoothing splines – basis is piecewise cubic functions with the join points at the observations.

Thin plate spline:

$$\min_g \sum_{i=1}^{n} (\boldsymbol{y}_i - g(\boldsymbol{x}_i))^2 + \lambda J(g)$$

$J(g)$ integral of squared partial derivatives of g.

E.g. in two dimension ($d = 2$), second order ($m = 2$)
$J(g) = \int_{\Re^2} \left[(\frac{\partial^2}{\partial^2 x_1} g)^2 + 2(\frac{\partial^2}{\partial x_1 \partial x_2} g)^2 + (\frac{\partial^2}{\partial^2 x_2} g)^2 \right] dx_1 dx_2$

Incredibly this minimization over functions has a simple and computable solution!

Estimating λ from data

Cross-validation – Wahba (1970)

- Leave each observation out. Predict it with the remaining data for a fixed lambda.

- Find λ that the minimizes mean squared error.

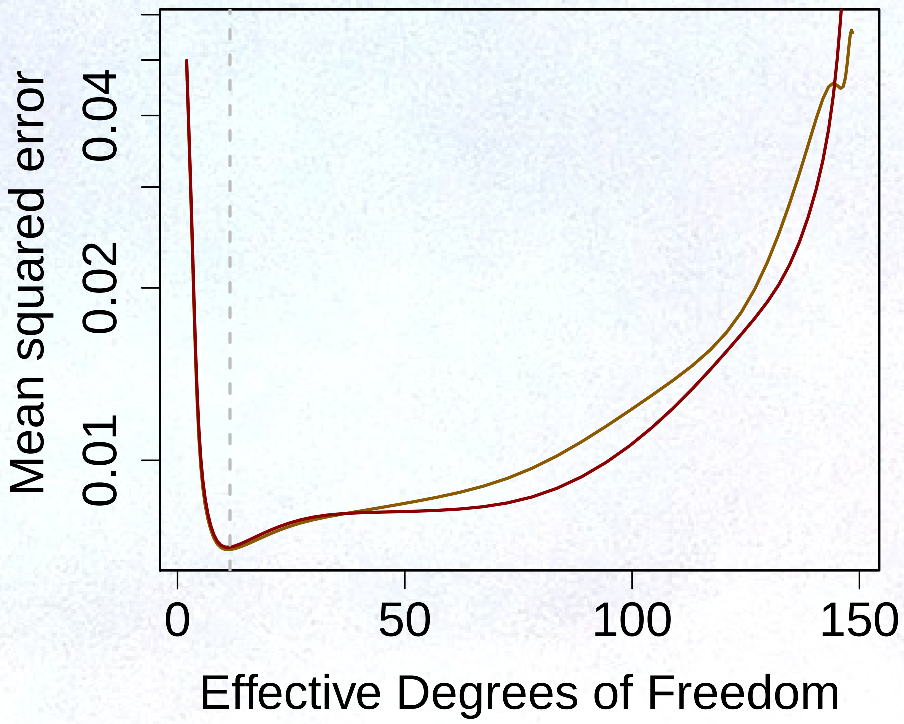
- Initially proposed for cubic smoothing splines but works for any smoother.

CV: obvious leave-one-out MSE

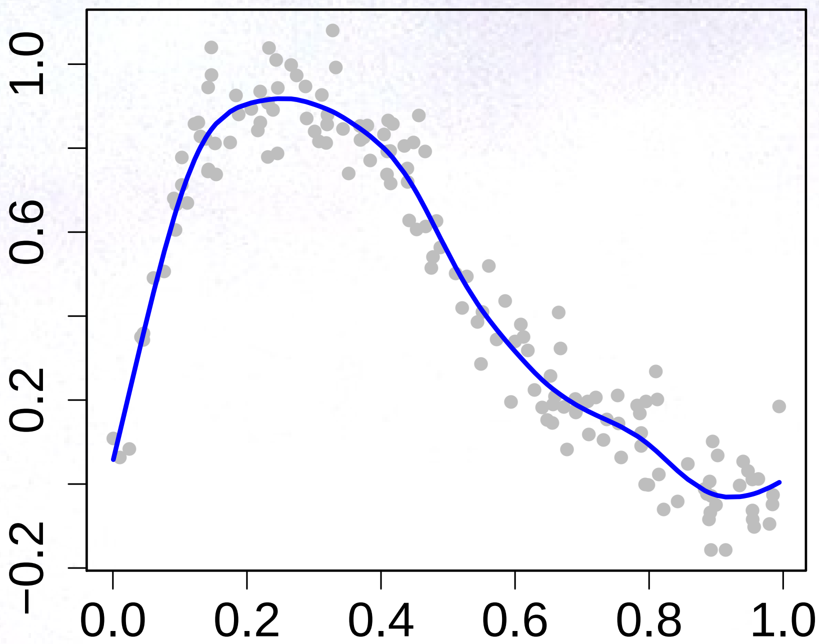
GCV: Adjust each CV squared residual for prediction error.

Finding smoothing parameter

Cross-validation functions



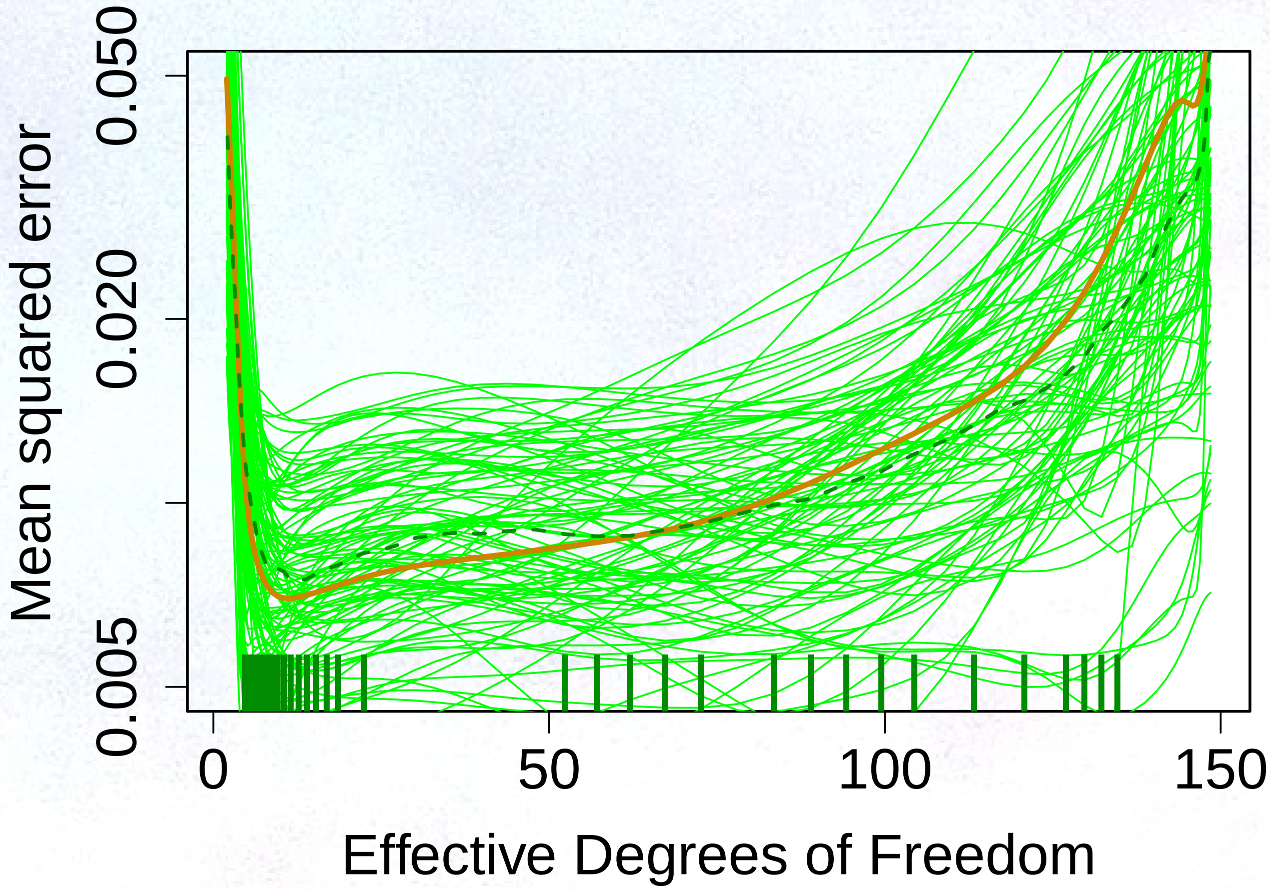
Cubic spline at minimum



Machine learning and λ

Divide data up into 90/10 training/validation and do out-sample-CV

Based on 100 random divisions ...



90/10 CV curves, minima |, median - - - , GCV —

In R

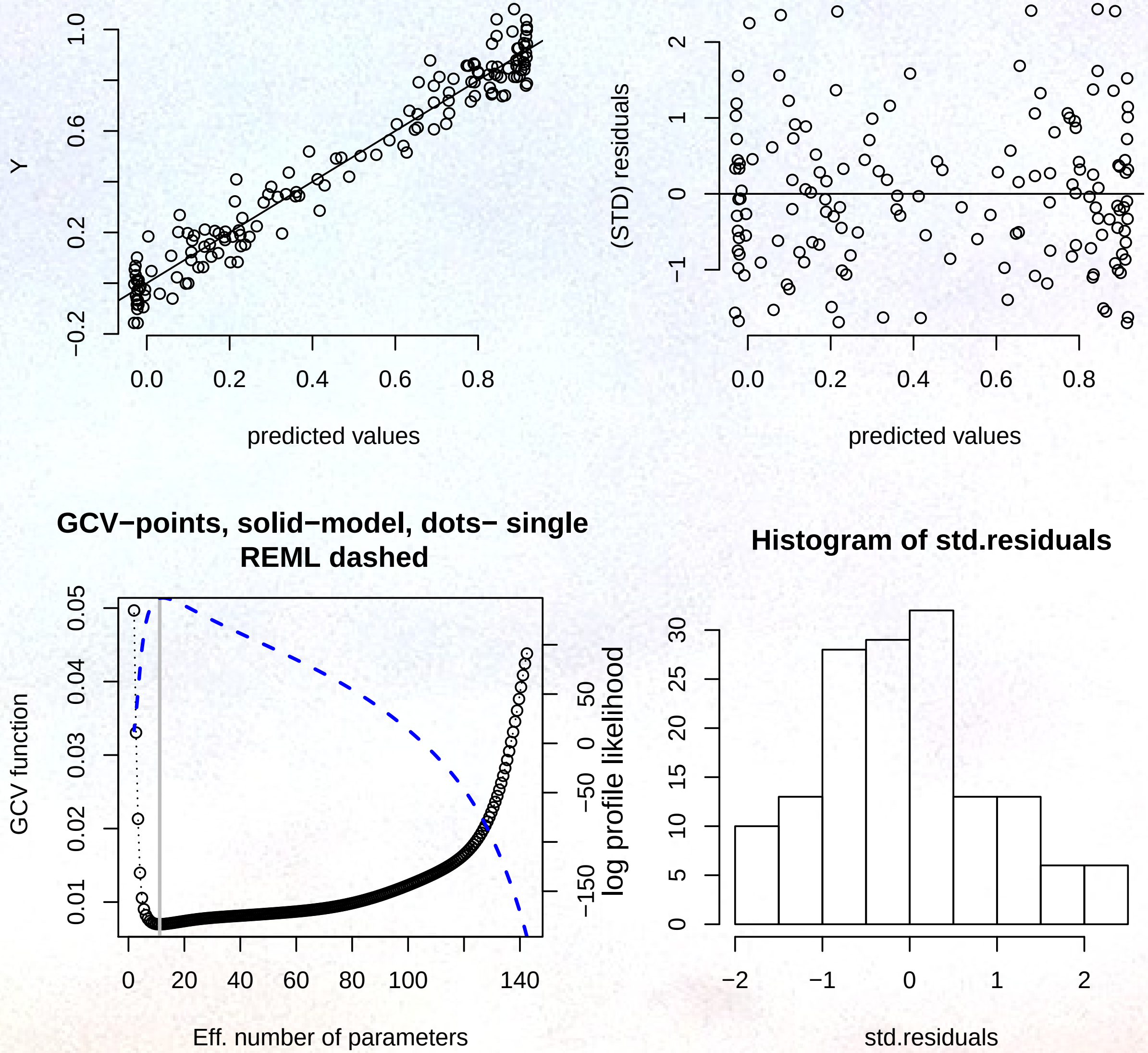
Tps -Thin plate spline function

```
# x and y are the synthetic example

library( fields)
obj<- Tps( x,y)
```

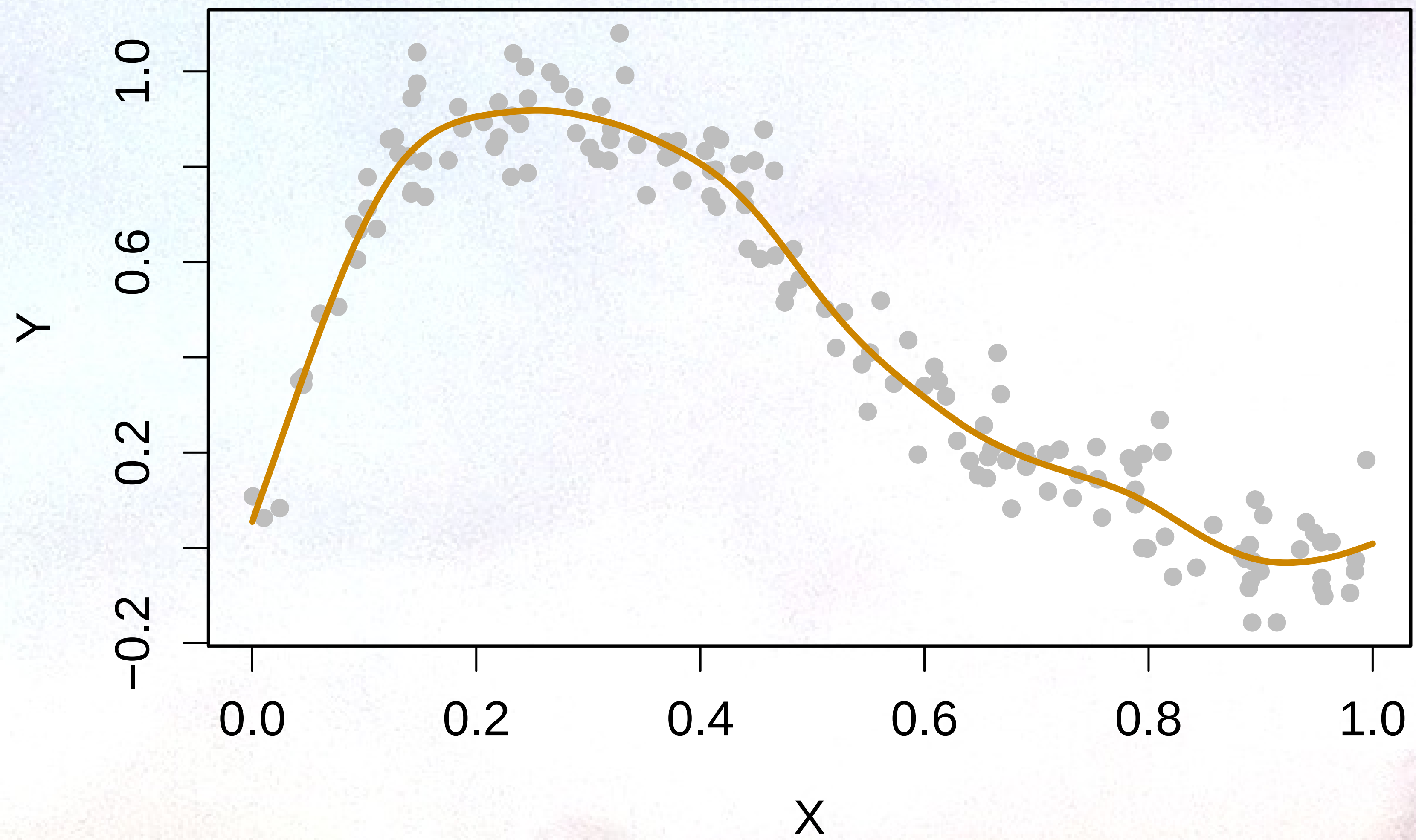
In R

```
> obj
Call:
Tps(x = x, Y = y)

 Number of Observations:                      150
 Number of parameters in the null space 2
 Parameters for fixed spatial drift     2
 Model degrees of freedom:                    11.2
 Residual degrees of freedom:                 138.8
 GCV estimate for sigma:                      0.08037
 MLE for sigma:                               0.08091
 MLE for rho:                                 31.1
 lambda                                       0.00021
 User supplied rho                            NA
 User supplied sigma^2                        NA
Summary of estimates:
                 lambda      trA         GCV       shat -lnLike Prof converge
GCV        0.0002105075 11.17840 0.006979633 0.08037097    -147.2260        1
REML       0.0001397338 12.25141 0.006998224 0.08016631    -147.5925        8
>
```

```
set.panel( 2,2)
plot( obj)
```



```
xgrid<- seq( 0,1, length.out=200)
fhat<- predict( obj, xgrid)
```



A statistical model.

Reverse engineering our estimator

A statistical refactoring

$$\min_{\boldsymbol{\beta}} \quad \Sigma_{i=1}^{n}(\boldsymbol{y} - [X\boldsymbol{\beta}]_i)^2 + \lambda \boldsymbol{\beta}^T Q \boldsymbol{\beta} \quad ,$$

Divide up $\lambda = \sigma^2/\rho$ and switch to "max"

$$\max_{\boldsymbol{\beta}} \; -\Sigma_{i=1}^{n} \frac{(\boldsymbol{y} - [\boldsymbol{X}\boldsymbol{\beta}]_i)^2}{2\sigma^2} - \frac{\boldsymbol{\beta}^T \boldsymbol{Q} \boldsymbol{\beta}}{2\rho},$$

exponentiate

$$\max_{\boldsymbol{\beta}} \quad e^{-\Sigma_{i=1}^{n} \frac{(\boldsymbol{y} - [\boldsymbol{X}\boldsymbol{\beta}]_i)^2}{2\sigma^2}} \quad e^{-\frac{\boldsymbol{\beta}^T \boldsymbol{Q} \boldsymbol{\beta}}{2\rho}}$$

More engineering

Normalize to densities and $Q = K^{-1}$

$$\max_{\boldsymbol{\beta}} \quad \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{2\sigma^2}} \quad e^{-\frac{\boldsymbol{\beta}^T(\boldsymbol{\rho K})^{-1}\boldsymbol{\beta}}{2}} |\rho K|^{-1/2}$$

Keep the Bayesians happy ...

$$\max_{\boldsymbol{\beta}} \quad \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{2\sigma^2}} \quad e^{-\frac{\boldsymbol{\beta}^T(\boldsymbol{\rho K})^{-1}\boldsymbol{\beta}}{2}} |\rho K|^{-1/2} \quad p(\rho, \sigma, K)$$

Gelfand Notation

$$[\boldsymbol{y}|\boldsymbol{\beta}, \sigma, \rho, K] \qquad [\boldsymbol{\beta}|\rho, \sigma, K] \qquad [\rho, \sigma, K]$$

What have we built?

$[\boldsymbol{y}|\boldsymbol{\beta}, \sigma, \rho, K] \quad [\boldsymbol{\beta}|\rho, \sigma, K] \quad [\rho, \sigma, K]$

- $\boldsymbol{y}$ *given* the function and parameters is Gaussian.
- Coefficients, $\boldsymbol{\beta}$, given the parameters is Gaussian $(0, \rho K)$
- Fixing parameters $\boldsymbol{y}$ is Gaussian $(0, \sigma^2 I + \rho \boldsymbol{K})$

Penalized least squares estimator $\equiv$ a Gaussian statistical model for the data and coefficients.

Minimizing the penalized LS $\equiv$ posterior maximum or a conditional expectation from Gaussian.

Thank you!