ETC5410: Nonparametric smoothing methods

July 2008

MONASH University

Rob J Hyndman

http://www.robjhyndman.com/

Outline

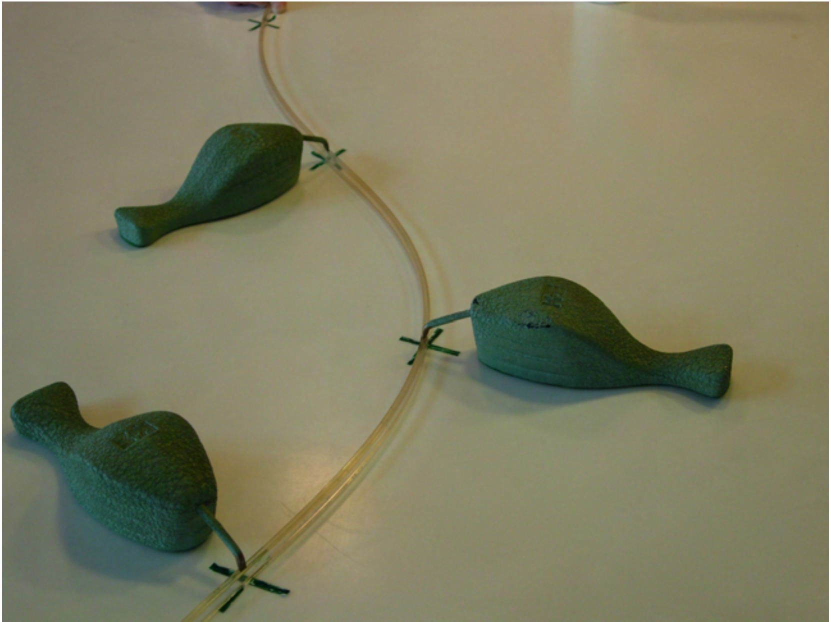
1. Density estimation
2. Kernel regression
3. **Splines**
4. Additive models
5. Functional data analysis

ETC5410: Nonparametric smoothing methods

3. Splines

Outline

Interpolating splines



Interpolating splines

Observations: (x_i, y_i) for $i = 1, \ldots, n$.

Interpolating splines

Observations: (x_i, y_i) for $i = 1, \ldots, n$.

A spline is a continuous function $f(x)$ interpolating all points and consisting of polynomials between each consecutive pair of 'knots' x_i and x_{i+1}.

Interpolating splines

Observations: (x_i, y_i) for $i = 1, \dots, n$.

A spline is a continuous function $f(x)$ interpolating all points and consisting of polynomials between each consecutive pair of 'knots' x_i and x_{i+1}.

- Parameters constrained so that $f(x)$ is continuous.

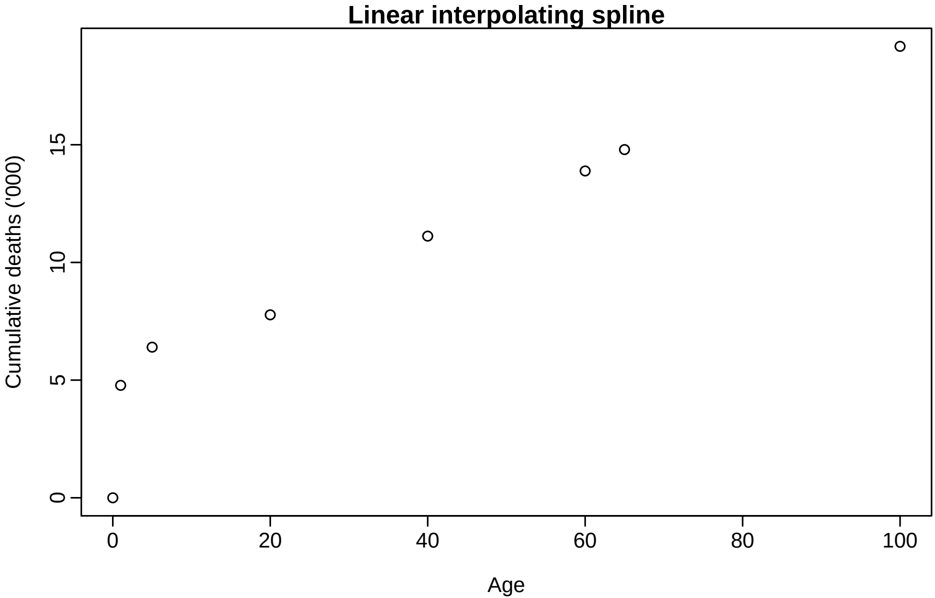
Interpolating splines

Observations: (x_i, y_i) for $i = 1, \dots, n$.

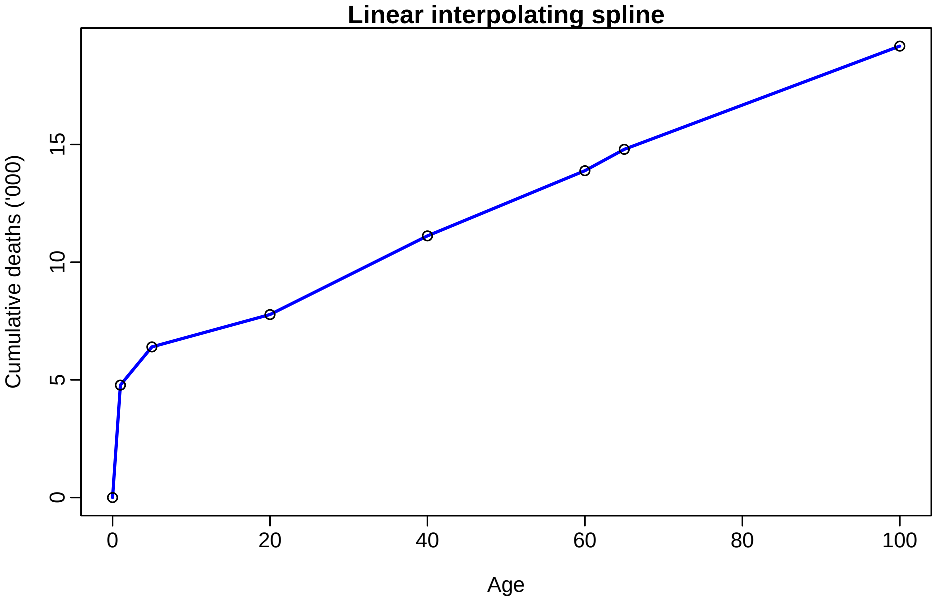
A spline is a continuous function $f(x)$ interpolating all points and consisting of polynomials between each consecutive pair of 'knots' x_i and x_{i+1}.

- Parameters constrained so that $f(x)$ is continuous.
- Further constraints imposed to give continuous derivatives.

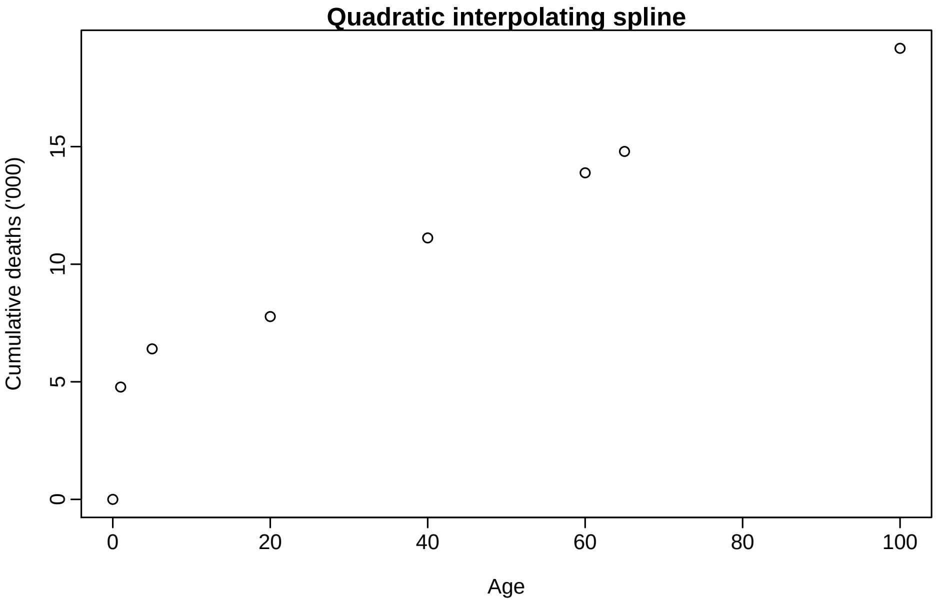
Interpolating splines



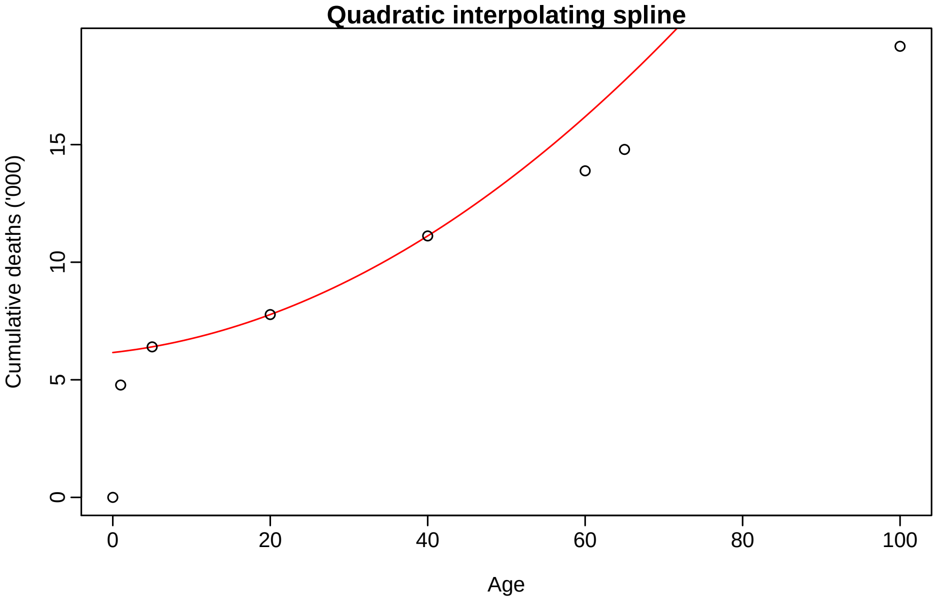
Interpolating splines



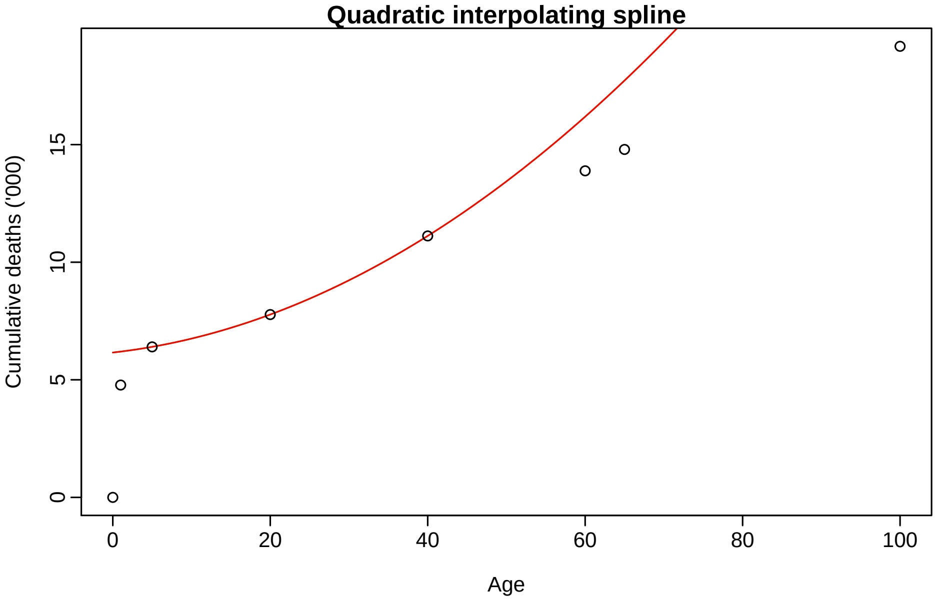
Interpolating splines



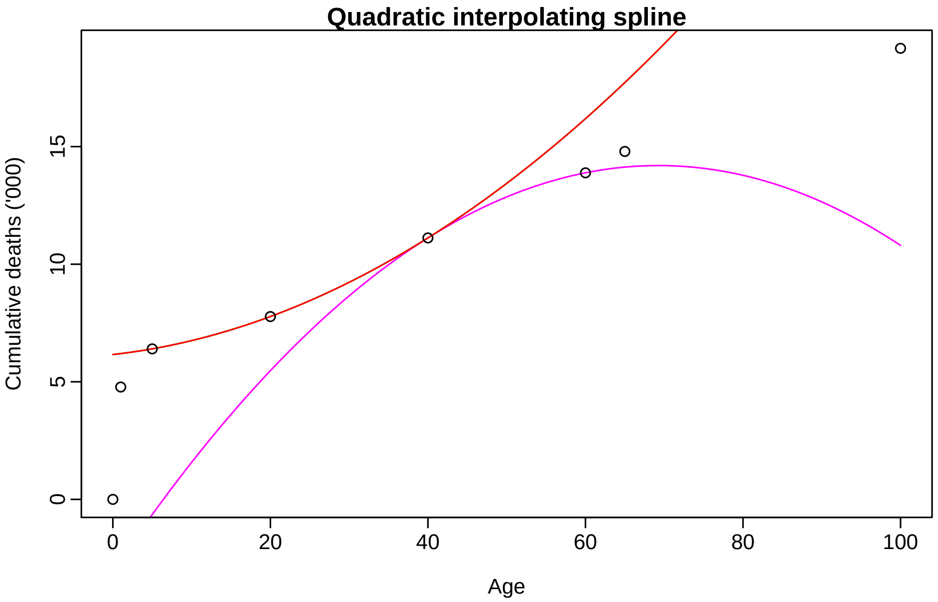
Interpolating splines



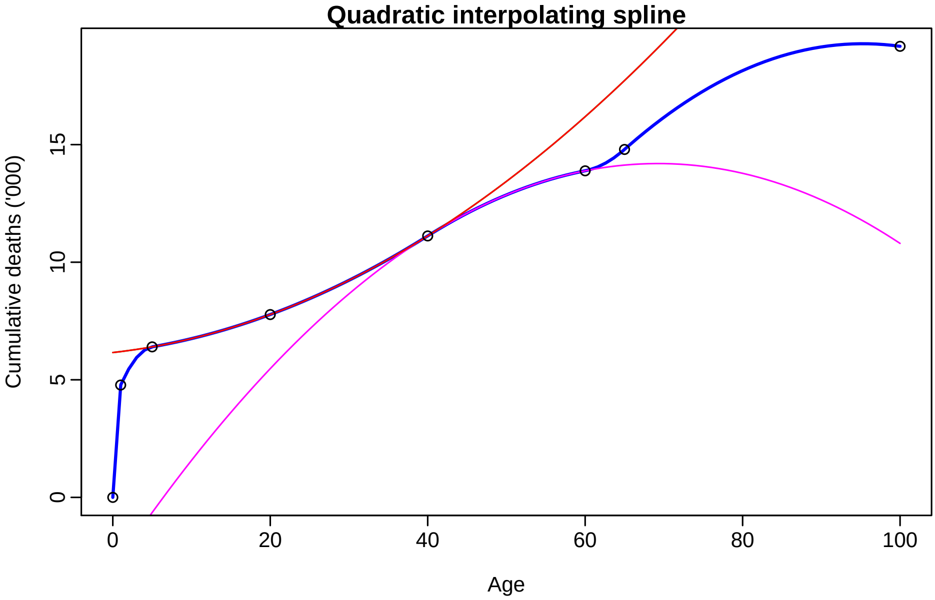
Interpolating splines



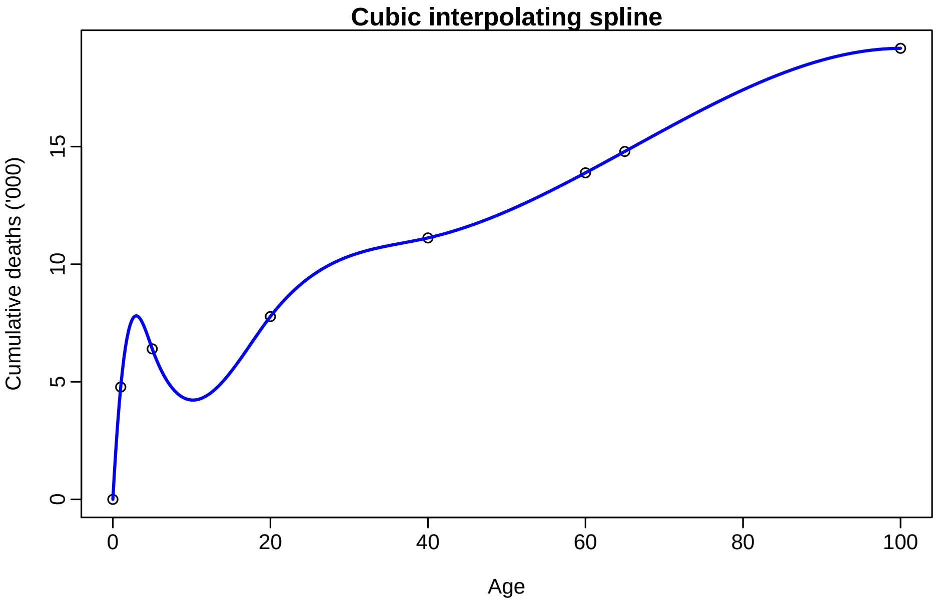
Interpolating splines



Interpolating splines



Interpolating splines



Interpolating splines

- To define a cubic spline uniquely, it is necessary to specify two additional constraints.

Interpolating splines

- To define a cubic spline uniquely, it is necessary to specify two additional constraints.
- Usually, this is done by specifying the form of the function at the two extremes, x_1 and x_n.

Interpolating splines

- To define a cubic spline uniquely, it is necessary to specify two additional constraints.
- Usually, this is done by specifying the form of the function at the two extremes, x_1 and x_n.
- For example, a "natural" spline is obtained by requiring $f''(x_1) = f''(x_n) = 0$, thus making the curve linear at the extremes.

Interpolating splines

- To define a cubic spline uniquely, it is necessary to specify two additional constraints.
- Usually, this is done by specifying the form of the function at the two extremes, x_1 and x_n.
- For example, a "natural" spline is obtained by requiring $f''(x_1) = f''(x_n) = 0$, thus making the curve linear at the extremes.

Interpolating splines

- To define a cubic spline uniquely, it is necessary to specify two additional constraints.
- Usually, this is done by specifying the form of the function at the two extremes, x_1 and x_n.
- For example, a "natural" spline is obtained by requiring $f''(x_1) = f''(x_n) = 0$, thus making the curve linear at the extremes.

Implementation in R

```
plot(x,y)
lines(spline(x,y))
```

Monotonic interpolation

‘Hyman filter’ ensures spline is monotonic by constraining derivatives while (possibly) sacrificing smoothness.

Monotonic interpolation

‘Hyman filter’ ensures spline is monotonic by constraining derivatives while (possibly) sacrificing smoothness.

Second derivative may no longer be continuous at all knots.

Monotonic interpolation

'Hyman filter' ensures spline is monotonic by constraining derivatives while (possibly) sacrificing smoothness.

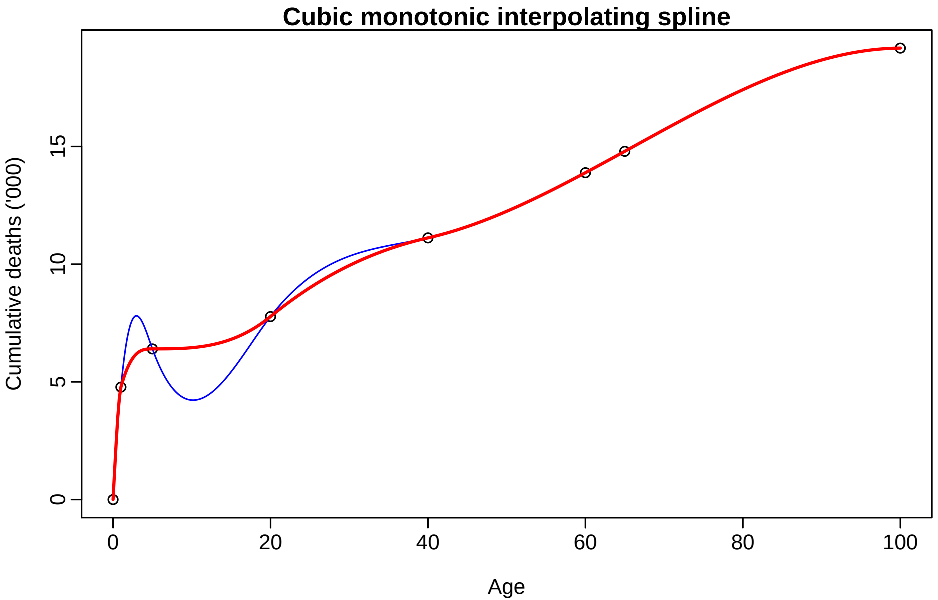
Second derivative may no longer be continuous at all knots.

Implementation in R

```
require(demography)
plot(x,y)
lines(cm.spline(x,y), col=4)
```

Reference: Smith, Hyndman and Wood (JPR, 2001)

Monotonic interpolation



Outline

Spline smoothing

The average squared prediction error is

$$\sum [y_j - r(x_j)]^2.$$

Reduces to zero if r interpolates the data.

Spline smoothing

The average squared prediction error is

$$\sum [y_j - r(x_j)]^2.$$

Reduces to zero if r interpolates the data.

Penalized sum of squares

$$S_\lambda(r) = \sum_{j=1}^{n} [y_j - r(x_j)]^2 + \lambda \int (r''(x))^2 dx$$

Spline smoothing

The average squared prediction error is

$$\sum [y_j - r(x_j)]^2.$$

Reduces to zero if r interpolates the data.

Penalized sum of squares

$$S_\lambda(r) = \sum_{j=1}^{n} [y_j - r(x_j)]^2 + \lambda \int (r''(x))^2 dx$$

- λ denotes a smoothing parameter.

Spline smoothing

The average squared prediction error is

$$\sum[y_j - r(x_j)]^2.$$

Reduces to zero if r interpolates the data.
Penalized sum of squares

$$S_\lambda(r) = \sum_{j=1}^n [y_j - r(x_j)]^2 + \lambda \int (r''(x))^2 dx$$

- λ denotes a smoothing parameter.
- First term measures closeness to the data.

Spline smoothing

The average squared prediction error is

$$\sum [y_j - r(x_j)]^2.$$

Reduces to zero if r interpolates the data.
Penalized sum of squares

$$S_\lambda(r) = \sum_{j=1}^{n} [y_j - r(x_j)]^2 + \lambda \int (r''(x))^2 dx$$

- λ denotes a smoothing parameter.
- First term measures closeness to the data.
- Second term penalizes curvature in the function.

Cubic smoothing splines

A *cubic smoothing spline* is the function $\hat{r}_\lambda(x)$ which minimizes $S_\lambda(r)$ over the class of all twice differentiable functions on the range of $\{x_j\}$.

Cubic smoothing splines

A *cubic smoothing spline* is the function $\hat{r}_\lambda(x)$ which minimizes $S_\lambda(r)$ over the class of all twice differentiable functions on the range of $\{x_j\}$.

- It consists of piecewise cubic polynomials, with the pieces separated by the x_j values.

Cubic smoothing splines

A *cubic smoothing spline* is the function $\hat{r}_\lambda(x)$ which minimizes $S_\lambda(r)$ over the class of all twice differentiable functions on the range of $\{x_j\}$.

- It consists of piecewise cubic polynomials, with the pieces separated by the x_j values.
- At the design points, x_j, $\hat{r}_\lambda(x)$ and its first two derivatives are continuous. The third derivative may be discontinuous.

Cubic smoothing splines

A *cubic smoothing spline* is the function $\hat{r}_\lambda(x)$ which minimizes $S_\lambda(r)$ over the class of all twice differentiable functions on the range of $\{x_j\}$.

- It consists of piecewise cubic polynomials, with the pieces separated by the x_j values.
- At the design points, x_j, $\hat{r}_\lambda(x)$ and its first two derivatives are continuous. The third derivative may be discontinuous.
- At the minimum and maximum x_j values, the second derivative of $\hat{r}_\lambda(x)$ is zero. Hence, the smoothing spline is linear beyond the extreme data points.

Cubic smoothing splines

- Large values of λ produce smoother curves while smaller values produce rougher curves.

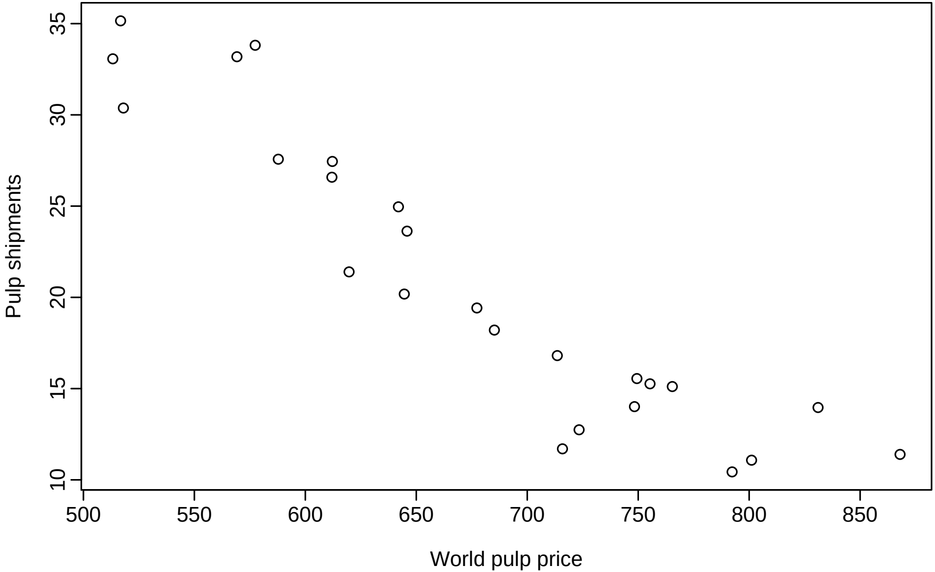
Cubic smoothing splines

- Large values of λ produce smoother curves while smaller values produce rougher curves.
- At $\lambda \to \infty$, the penalty term dominates $S_\lambda(r)$, forcing $r''(x) = 0$ for all x. So the solution is the least squares straight line.

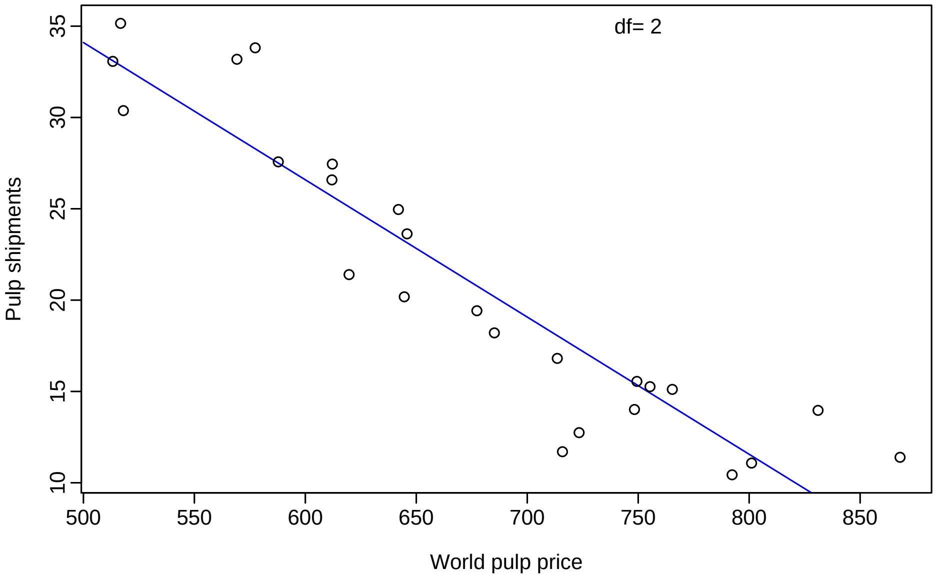
Cubic smoothing splines

- Large values of λ produce smoother curves while smaller values produce rougher curves.
- At $\lambda \to \infty$, the penalty term dominates $S_\lambda(r)$, forcing $r''(x) = 0$ for all x. So the solution is the least squares straight line.
- As $\lambda \to 0$, the penalty term becomes negligible and the solution tends to an interpolating function which is twice differentiable.

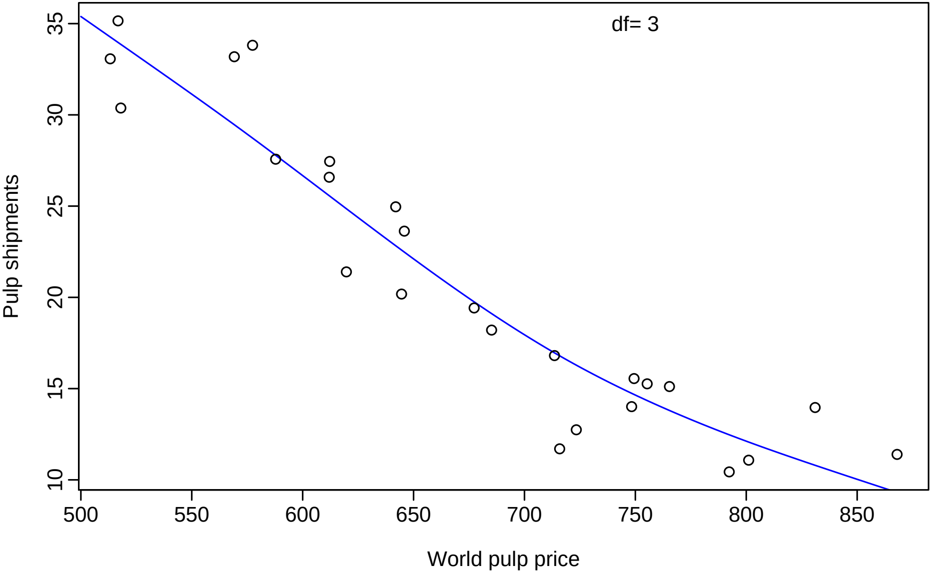
Cubic smoothing splines



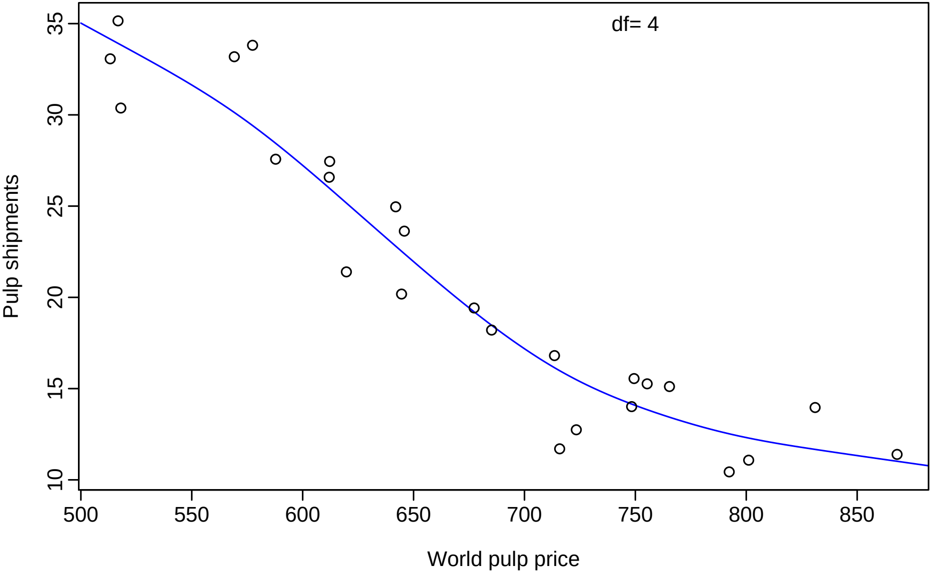
Cubic smoothing splines



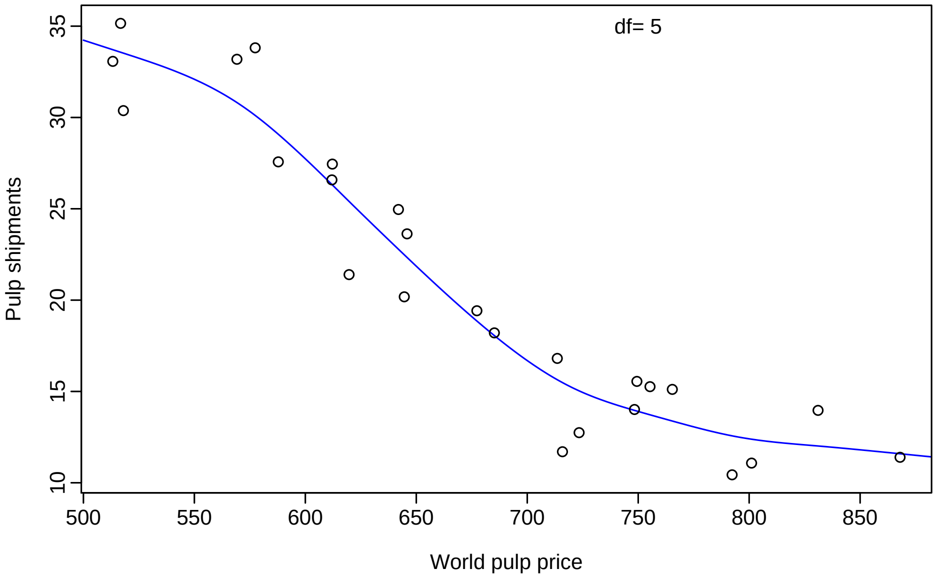
Cubic smoothing splines



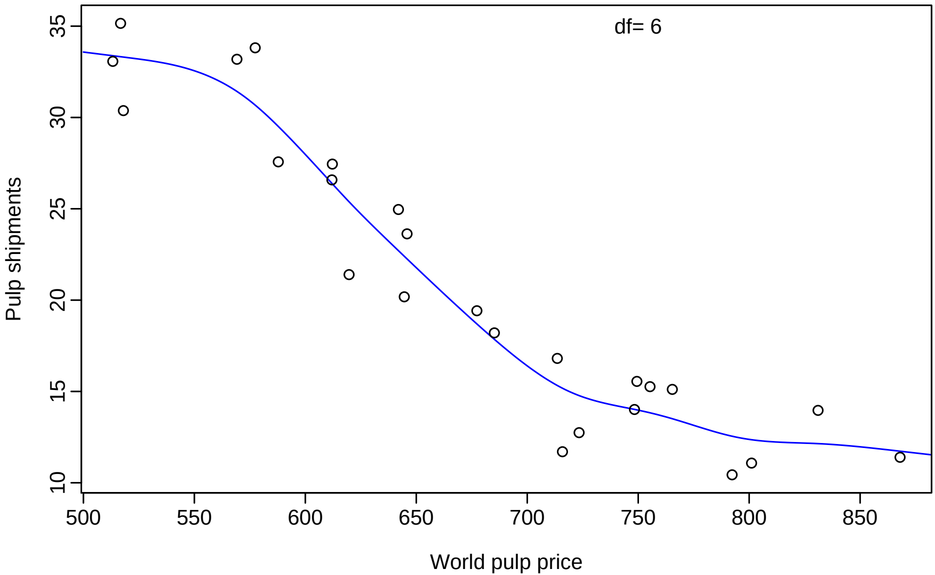
Cubic smoothing splines



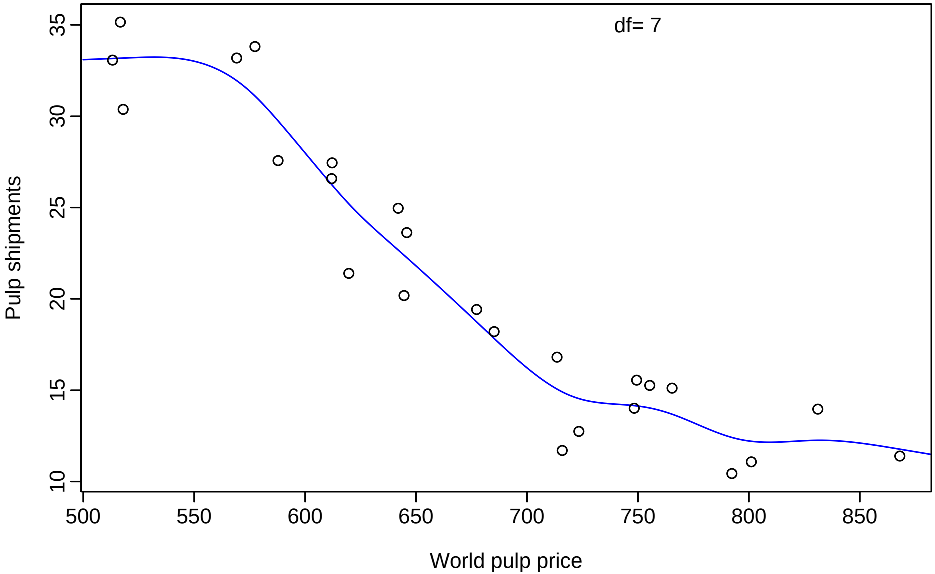
Cubic smoothing splines



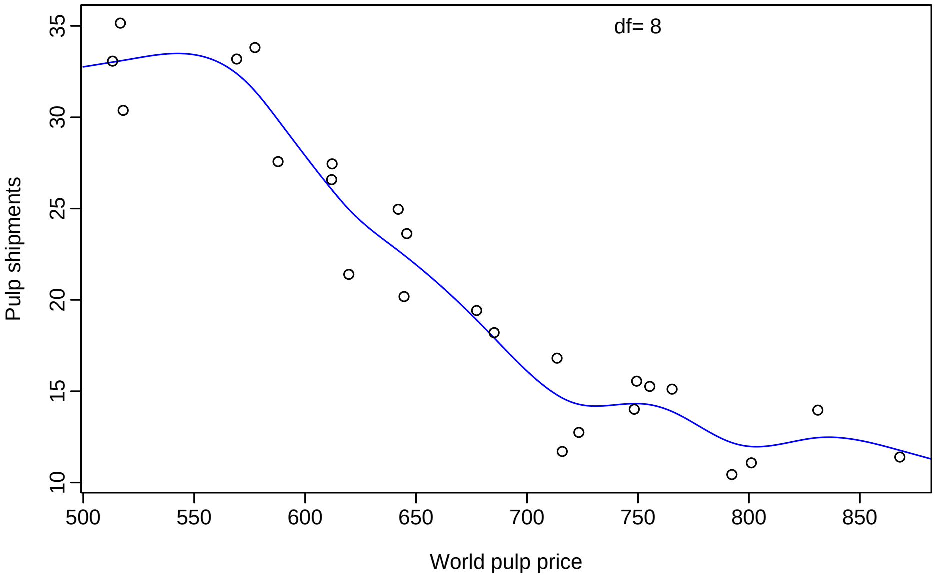
Cubic smoothing splines



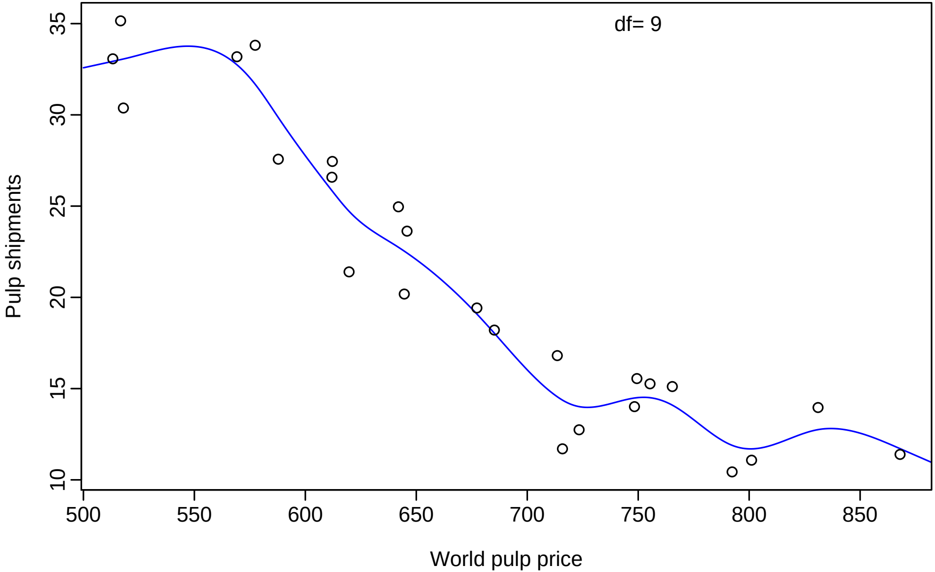
Cubic smoothing splines



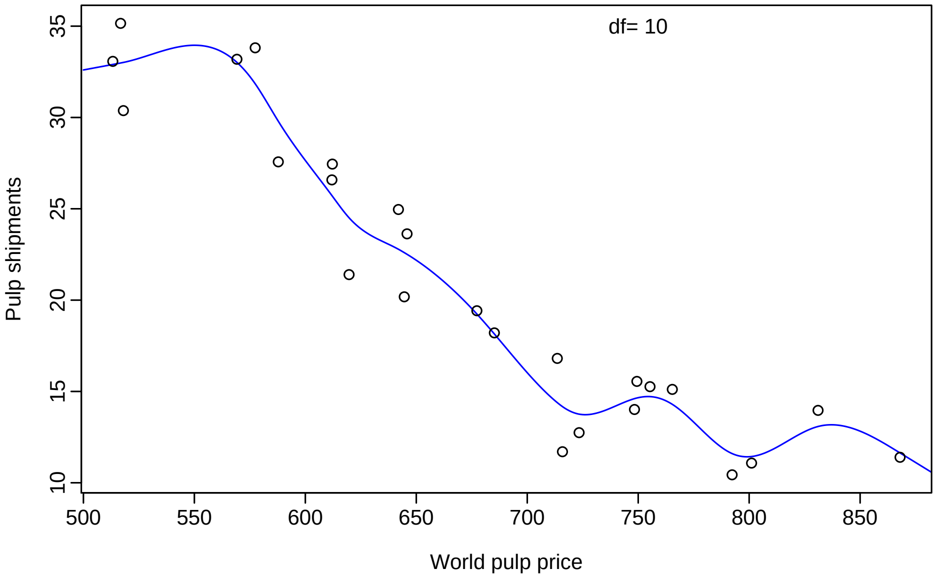
Cubic smoothing splines



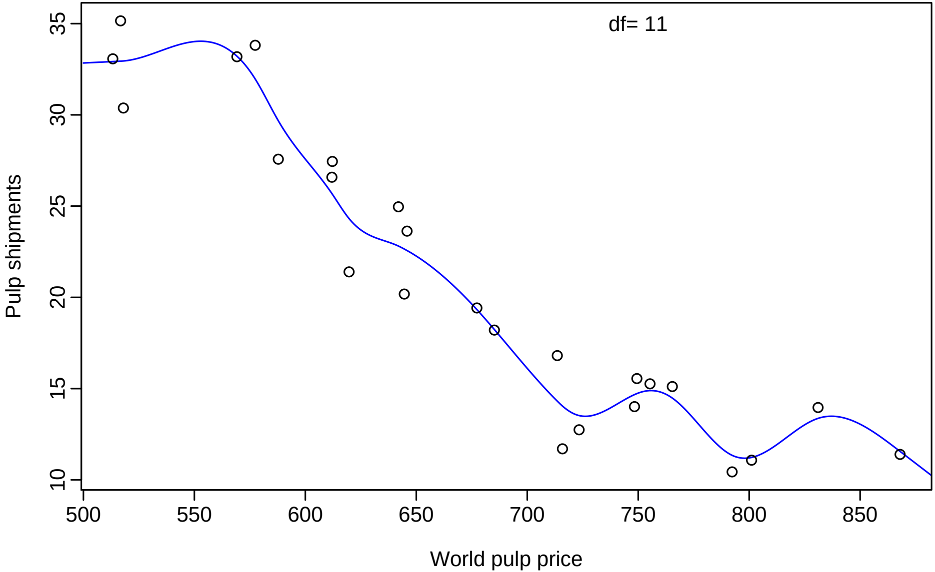
Cubic smoothing splines



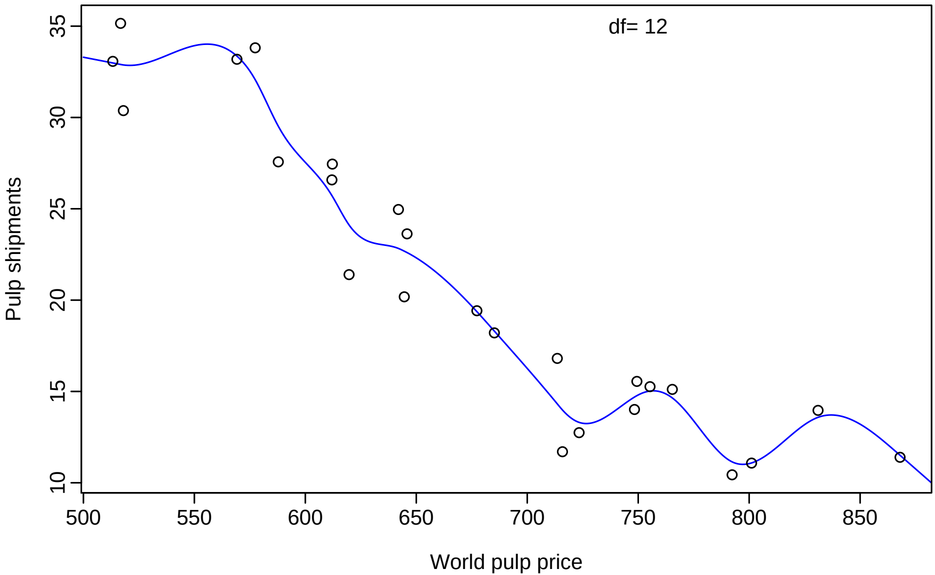
Cubic smoothing splines



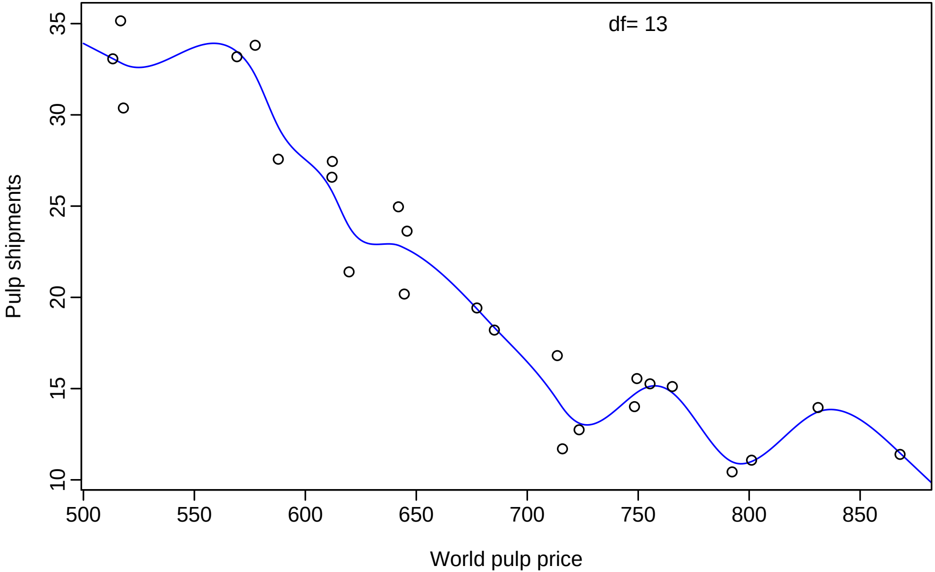
Cubic smoothing splines



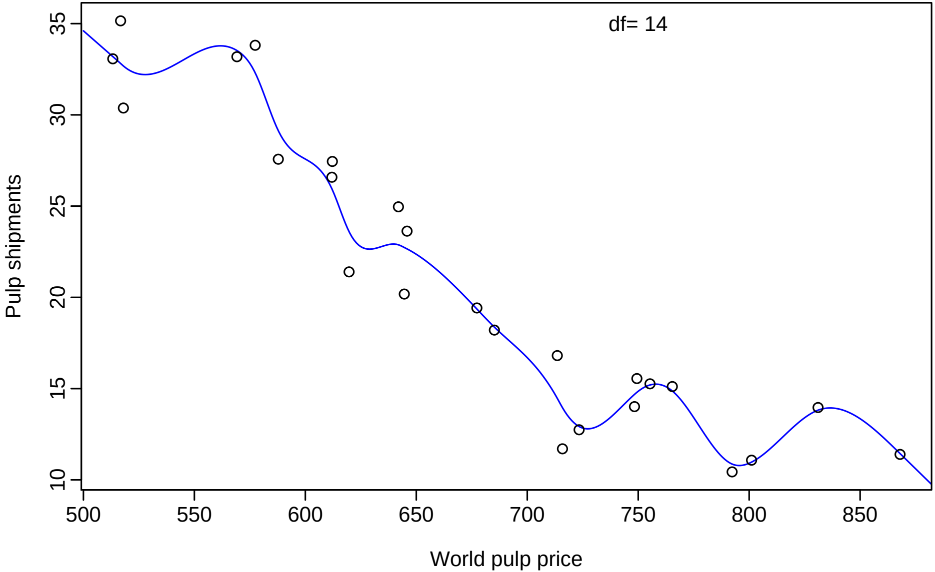
Cubic smoothing splines



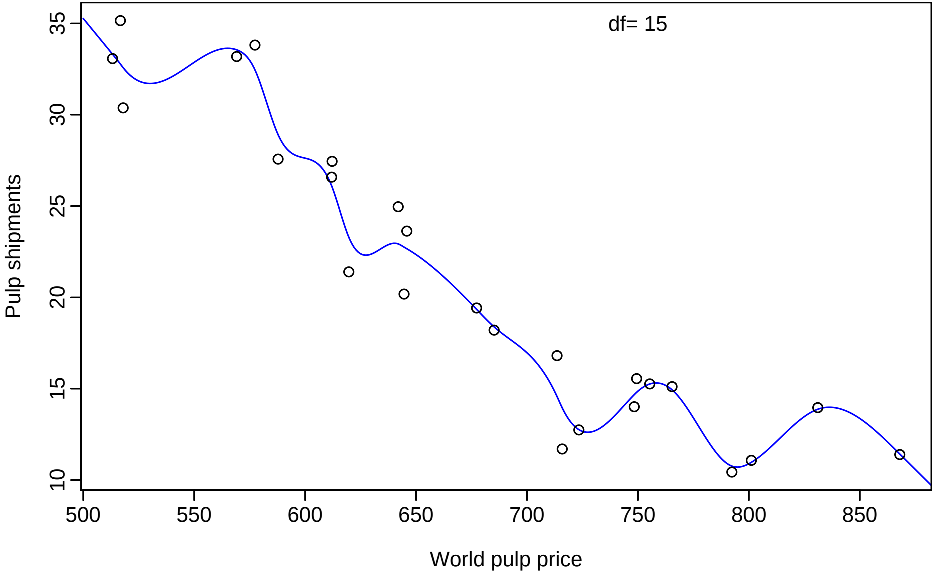
Cubic smoothing splines



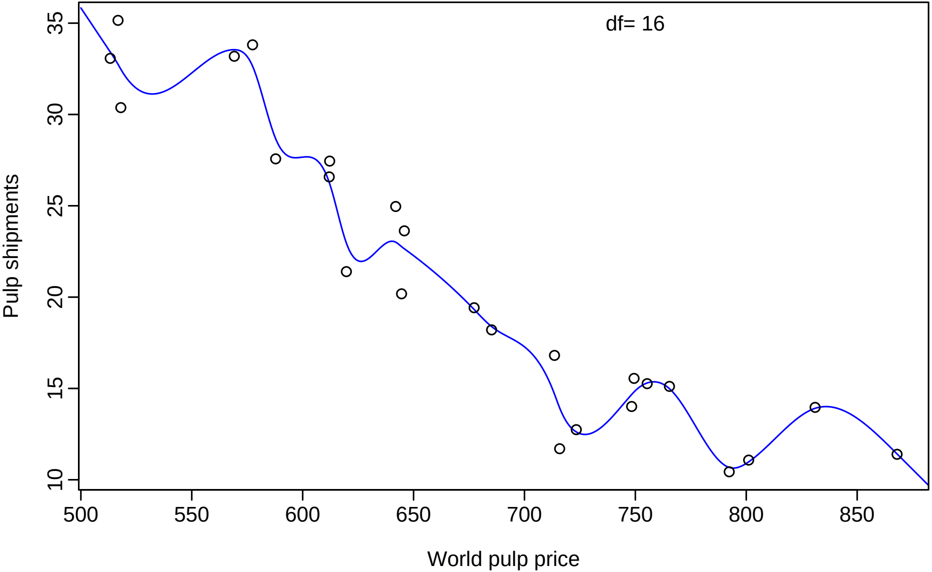
Cubic smoothing splines



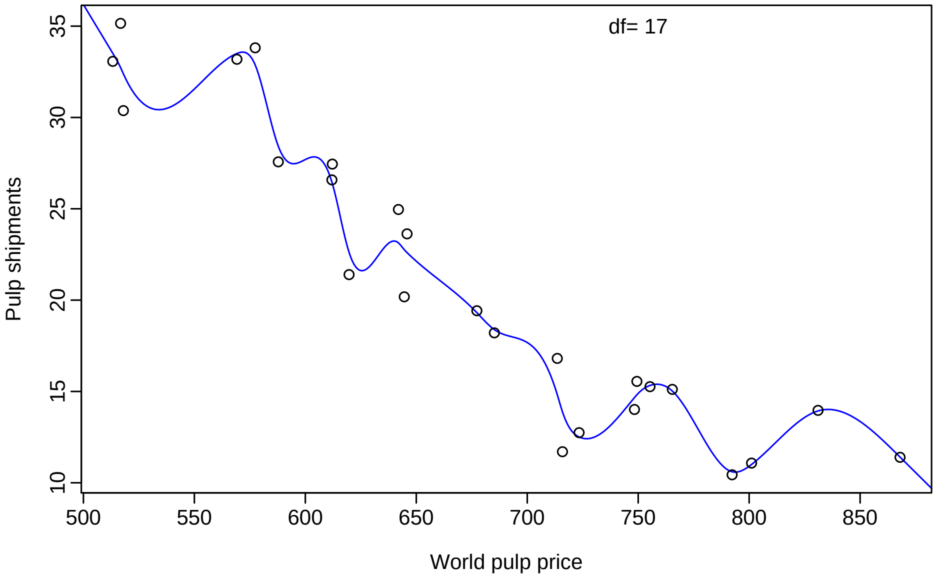
Cubic smoothing splines



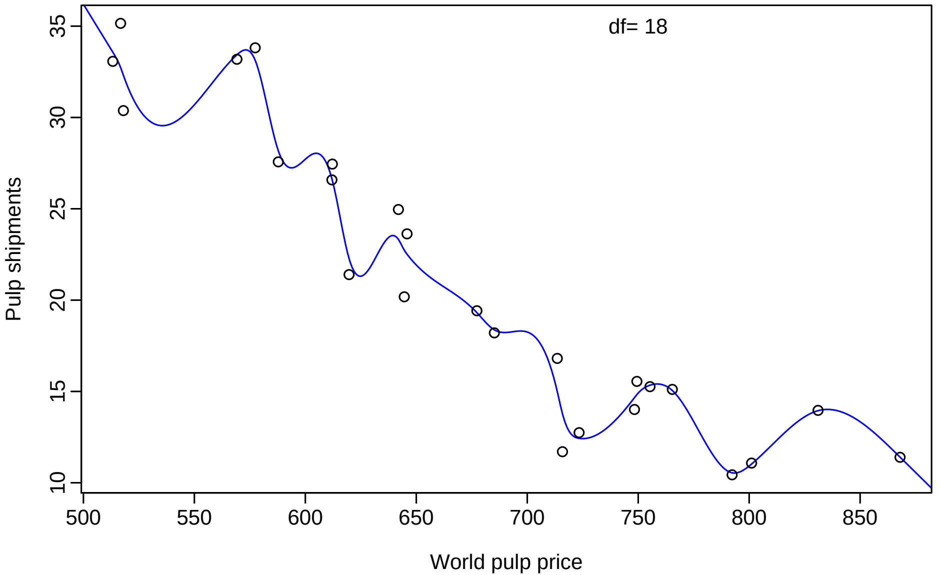
Cubic smoothing splines



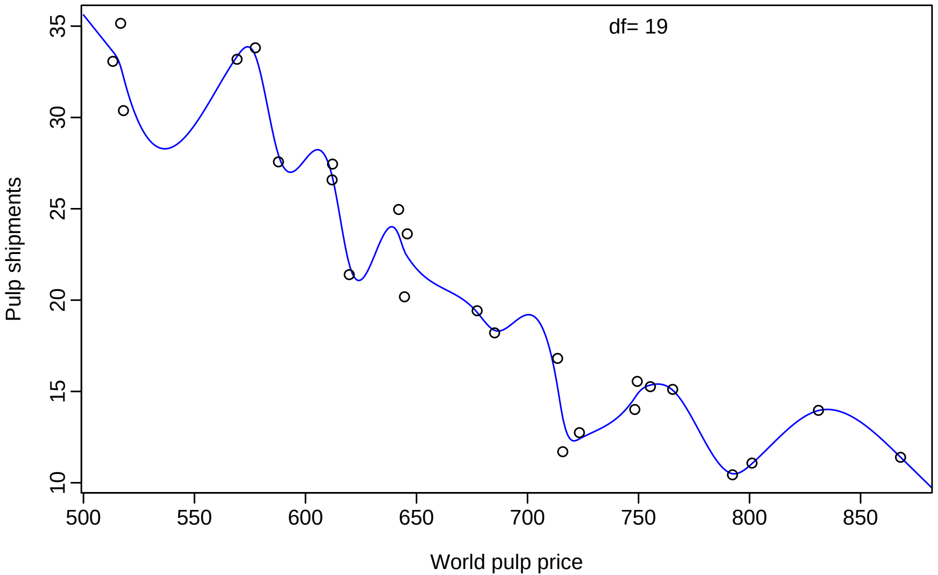
Cubic smoothing splines



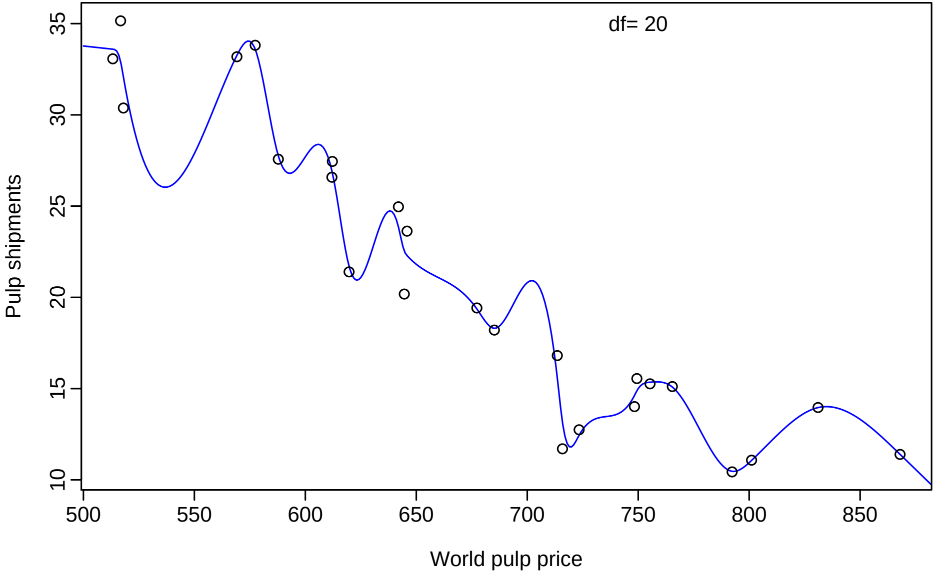
Cubic smoothing splines



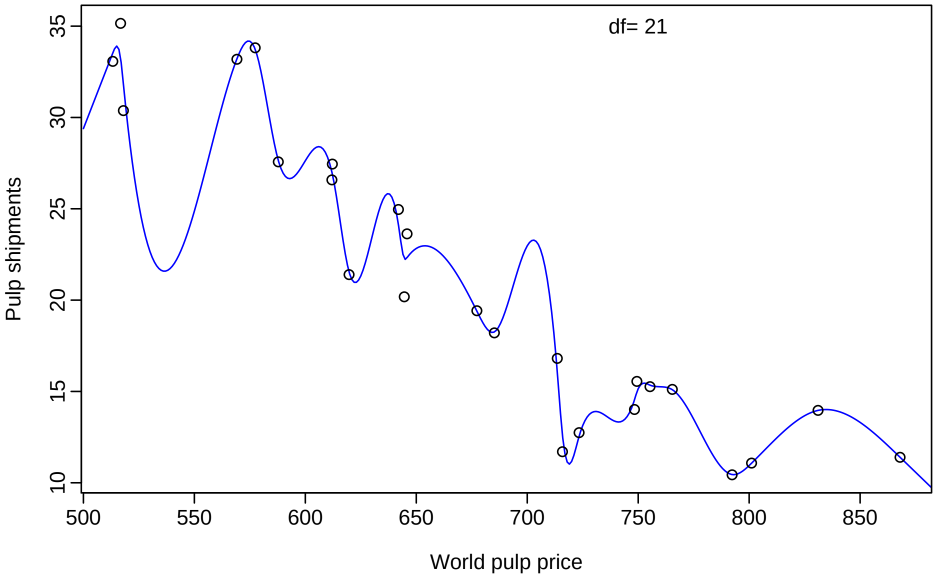
Cubic smoothing splines



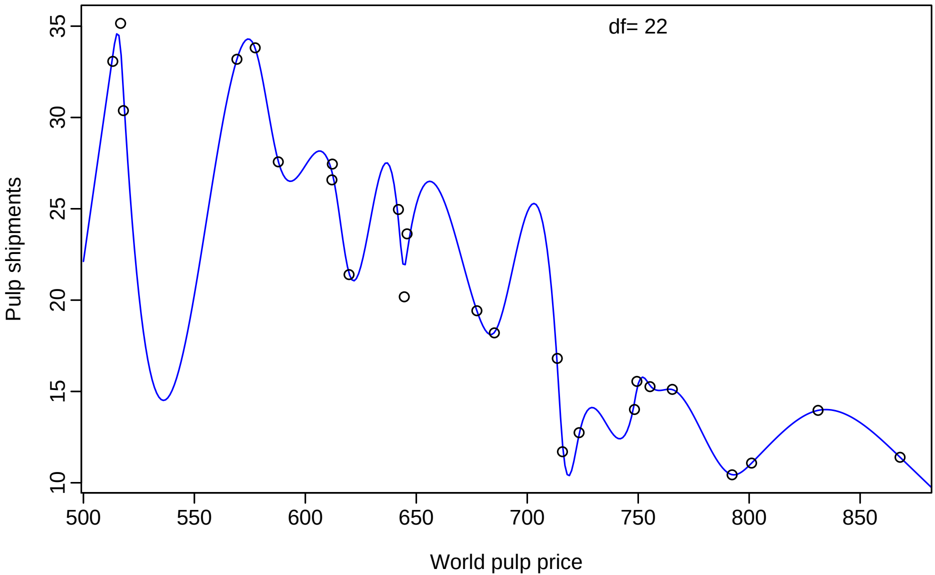
Cubic smoothing splines



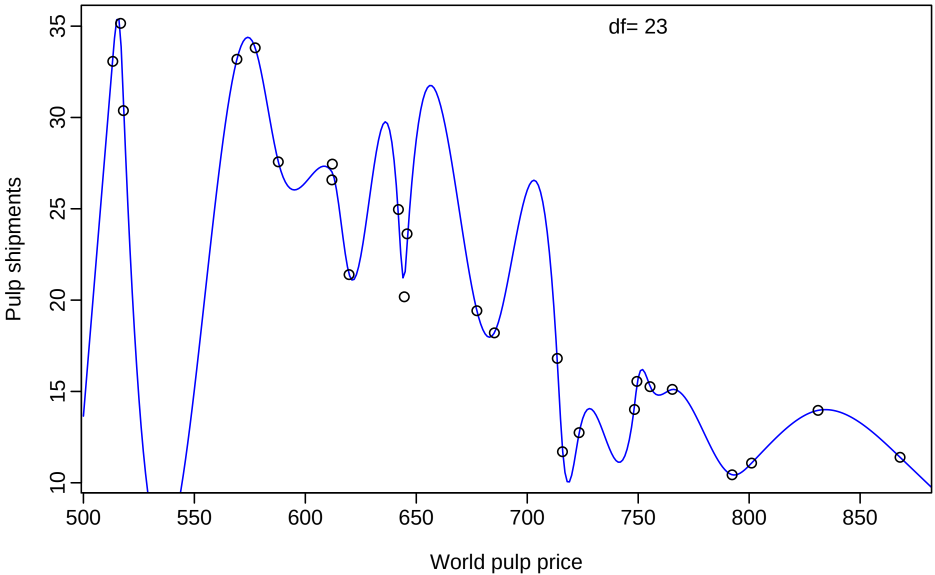
Cubic smoothing splines



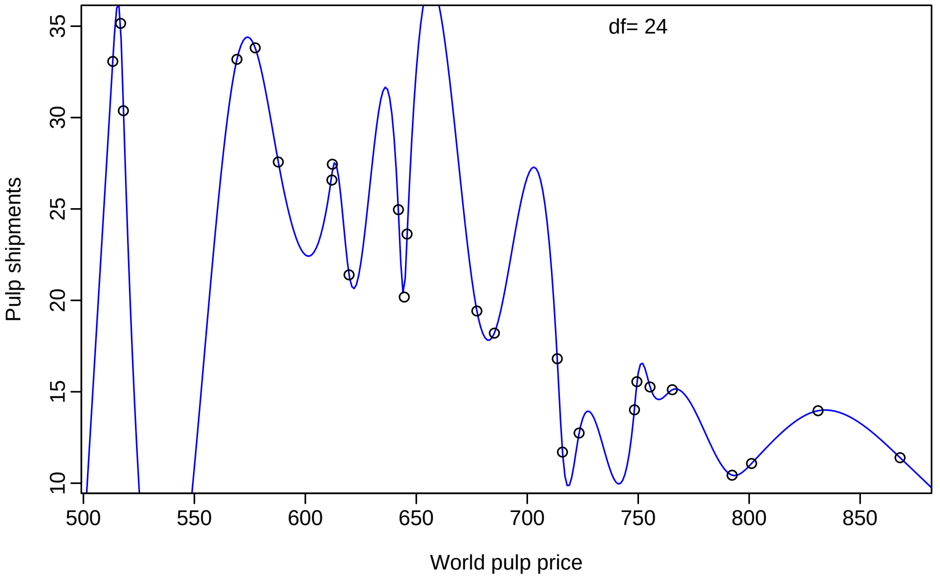
Cubic smoothing splines



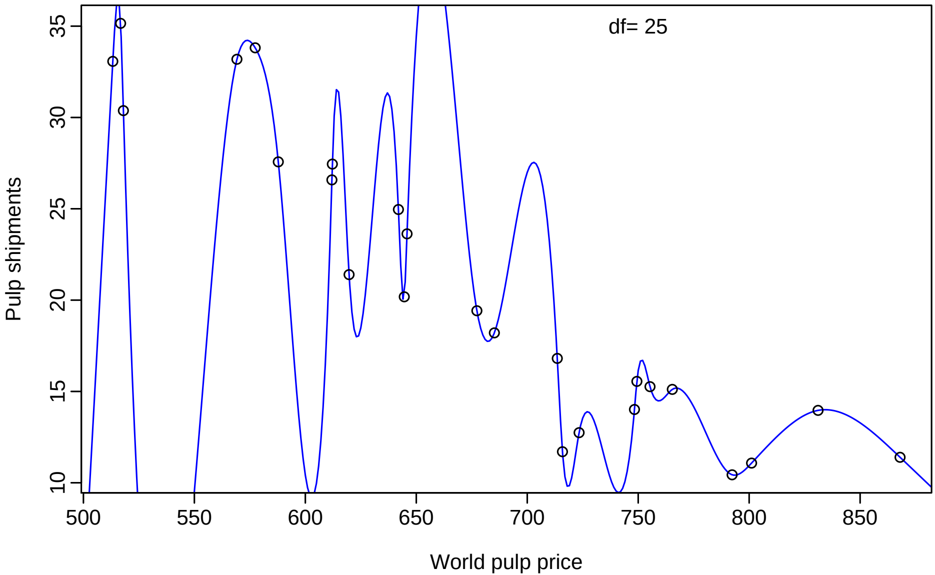
Cubic smoothing splines



Cubic smoothing splines



Cubic smoothing splines



Cubic smoothing splines

Implementation in R

```
plot(price, shipments)
lines(smooth.spline(price, shipments))
lines(smooth.spline(price, shipments,
      df=5), col=2)
```

Cubic smoothing splines

Implementation in R

```
plot(price, shipments)
lines(smooth.spline(price, shipments))
lines(smooth.spline(price, shipments,
    df=5), col=2)
```

- Fits a cubic smoothing spline to the data.

Cubic smoothing splines

Implementation in R

```
plot(price, shipments)
lines(smooth.spline(price, shipments))
lines(smooth.spline(price, shipments,
     df=5), col=2)
```

- Fits a cubic smoothing spline to the data.
- The value of λ is selected automatically using cross-validation.

Cubic smoothing splines

Implementation in R

```
plot(price, shipments)
lines(smooth.spline(price, shipments))
lines(smooth.spline(price, shipments,
     df=5), col=2)
```

- Fits a cubic smoothing spline to the data.
- The value of λ is selected automatically using cross-validation.
- The argument `df` can be supplied and then λ is chosen to give approximately `df` degrees of freedom.

Hodrick-Prescott filter

- The Hodrick-Prescott filter is a special case of a cubic smoothing spline.

Hodrick-Prescott filter

- The Hodrick-Prescott filter is a special case of a cubic smoothing spline.
- HP recommended $\lambda = 1600$ for quarterly data. There is no theoretical justification for this.

Hodrick-Prescott filter

- The Hodrick-Prescott filter is a special case of a cubic smoothing spline.
- HP recommended $\lambda = 1600$ for quarterly data. There is no theoretical justification for this.
- Better to use proper bandwidth selection tools.

Cross-validation again

Recall: Find smoothing parameter which minimises

$$\text{CV}(h) = \frac{1}{n}\sum_{j=1}^{n}[\hat{r}_j(x_j) - y_j]^2$$

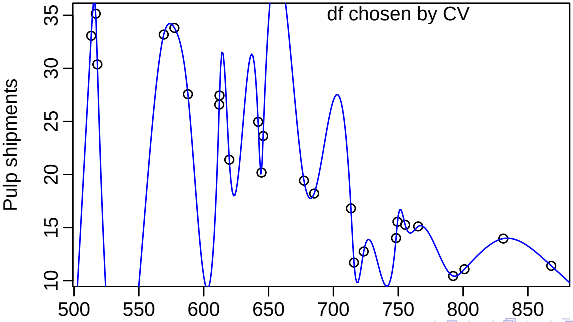
where $\hat{r}_j(x_j)$ uses all data *except* (x_j, y_j).

Cross-validation again

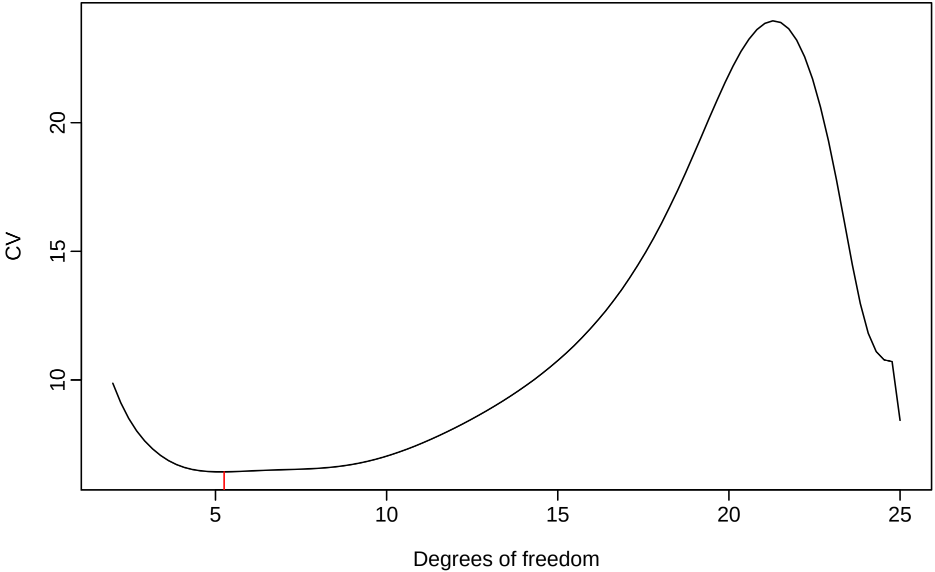
Recall: Find smoothing parameter which minimises

$$\mathrm{CV}(h) = \frac{1}{n}\sum_{j=1}^{n}[\hat{r}_j(x_j) - y_j]^2$$

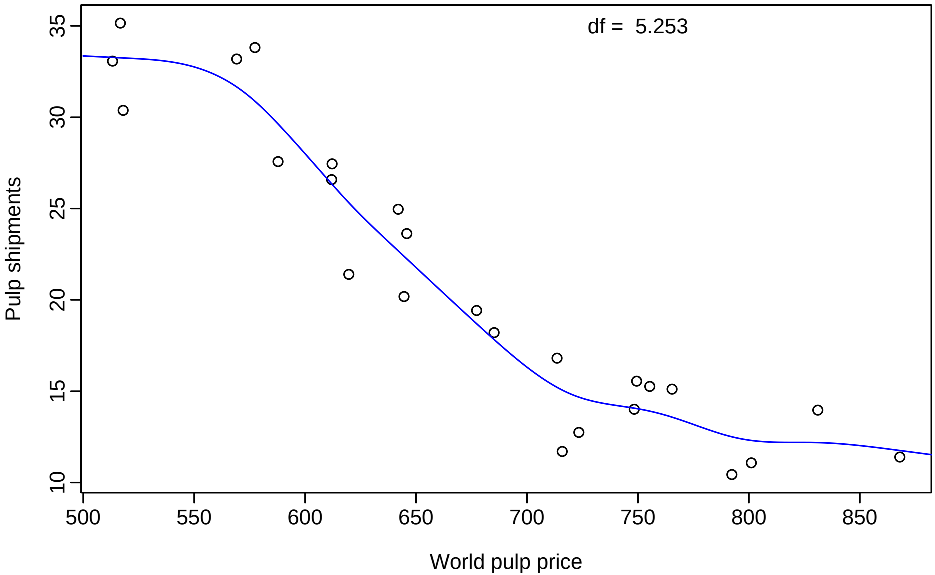
where $\hat{r}_j(x_j)$ uses all data *except* (x_j, y_j).



Cross-validation again



Cross-validation again



General cubic splines

Let $\kappa_1 < \kappa_2 < \cdots < \kappa_k$ be knots in interval (a, b).

General cubic splines

Let $\kappa_1 < \kappa_2 < \cdots < \kappa_k$ be knots in interval (a, b).

Let $h_1(x) = 1$, $h_2(x) = x$, $h_3(x) = x^2$, $h_4(x) = x^3$,
$h_j(x) = (x - \kappa_{j-4})_+^3$ for $j = 5, \dots, k + 4$.

General cubic splines

Let $\kappa_1 < \kappa_2 < \cdots < \kappa_k$ be knots in interval (a, b).

Let $h_1(x) = 1$, $h_2(x) = x$, $h_3(x) = x^2$, $h_4(x) = x^3$,
$h_j(x) = (x - \kappa_{j-4})_+^3$ for $j = 5, \ldots, k+4$.

$\{h_1, \ldots, h_{k+4}\}$ form a basis for the set of cubic splines at these knots, called the **truncated power basis**.

General cubic splines

Let $\kappa_1 < \kappa_2 < \cdots < \kappa_k$ be knots in interval (a, b).

Let $h_1(x) = 1$, $h_2(x) = x$, $h_3(x) = x^2$, $h_4(x) = x^3$,
$h_j(x) = (x - \kappa_{j-4})_+^3$ for $j = 5, \ldots, k+4$.

$\{h_1, \ldots, h_{k+4}\}$ form a basis for the set of cubic splines at these knots, called the **truncated power basis**.

Any cubic spline $r(x)$ with these knots can be written as

$$\boxed{r(x) = \sum_{j=1}^{k+4} \beta_j h_j(x).}$$

General cubic splines

Let $\kappa_1 < \kappa_2 < \cdots < \kappa_k$ be knots in interval (a, b).

Let $h_1(x) = 1$, $h_2(x) = x$, $h_3(x) = x^2$, $h_4(x) = x^3$,
$h_j(x) = (x - \kappa_{j-4})^3_+$ for $j = 5, \ldots, k+4$.

$\{h_1, \ldots, h_{k+4}\}$ form a basis for the set of cubic splines at these knots, called the **truncated power basis**.

Any cubic spline $r(x)$ with these knots can be written as

$$r(x) = \sum_{j=1}^{k+4} \beta_j h_j(x).$$

A cubic smoothing spline is obtained by setting $\kappa_j = x_j$, $j = 1, \ldots, n$.

Matrix form

Let $B_{ij} = h_j(x_i)$ be the basis matrix. Then for smoothing splines we need to minimize

$$(Y - B\boldsymbol{\beta})'(Y - B\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\Omega\boldsymbol{\beta}$$

where

$$\Omega_{jk} = \int h_j''(x)h_k''(x)dx.$$

Matrix form

Let $B_{ij} = h_j(x_i)$ be the basis matrix. Then for smoothing splines we need to minimize

$$(Y - B\beta)'(Y - B\beta) + \lambda\beta'\Omega\beta$$

where

$$\Omega_{jk} = \int h_j''(x) h_k''(x) dx.$$

This gives $\hat{\beta} = (B'B + \lambda\Omega)^{-1}B'\mathbf{Y}.$

Inference

Any cubic smoothing spline can be written as a linear smoother with smoothing matrix $\mathbf{S} = B(B'B + \lambda\Omega)^{-1}B'$.

Inference

Any cubic smoothing spline can be written as a linear smoother with smoothing matrix $\mathbf{S} = B(B'B + \lambda\Omega)^{-1}B'$.

$\hat{\mathbf{r}} = \mathbf{SY}$ where $\hat{\mathbf{r}} = [\hat{r}(x_1), \dots, \hat{r}(x_n)]$¿

Inference

Any cubic smoothing spline can be written as a linear smoother with smoothing matrix $\mathbf{S} = B(B'B + \lambda\Omega)^{-1}B'$.

$\hat{\mathbf{r}} = \mathbf{SY}$ where $\hat{\mathbf{r}} = [\hat{r}(x_1), \dots, \hat{r}(x_n)]¿$

- The effect of $\lambda\Omega$ is to shrink the regression coefficients to give a smoother fit.

Inference

Any cubic smoothing spline can be written as a linear smoother with smoothing matrix $\mathbf{S} = B(B'B + \lambda\Omega)^{-1}B'$.

$\hat{\mathbf{r}} = \mathbf{SY}$ where $\hat{\mathbf{r}} = [\hat{r}(x_1), \dots, \hat{r}(x_n)]¿$

- The effect of $\lambda\Omega$ is to shrink the regression coefficients to give a smoother fit.
- Because this is in the form of a linear smoother, the inference derived for local polynomial smoothing can be applied here too.

Inference

Any cubic smoothing spline can be written as a linear smoother with smoothing matrix $\mathbf{S} = B(B'B + \lambda\Omega)^{-1}B'$.

$\hat{\mathbf{r}} = \mathbf{S}\mathbf{Y}$ where $\hat{\mathbf{r}} = [\hat{r}(x_1), \dots, \hat{r}(x_n)]¿$

- The effect of $\lambda\Omega$ is to shrink the regression coefficients to give a smoother fit.
- Because this is in the form of a linear smoother, the inference derived for local polynomial smoothing can be applied here too.
- $\text{df} = \text{trace}(\mathbf{S})$

Outline

Regression splines

- Rather than have knots at each data point ($\kappa_i = x_i$), use fewer knots.

Regression splines

- Rather than have knots at each data point ($\kappa_i = x_i$), use fewer knots.
- Then do ordinary linear regression on the basis function.

Regression splines

- Rather than have knots at each data point ($\kappa_i = x_i$), use fewer knots.
- Then do ordinary linear regression on the basis function.
- Choice of knots can be difficult and arbitrary.

Regression splines

- Rather than have knots at each data point ($\kappa_i = x_i$), use fewer knots.
- Then do ordinary linear regression on the basis function.
- Choice of knots can be difficult and arbitrary.
- Automatic knot selection algorithms very slow.

Regression splines

- Rather than have knots at each data point ($\kappa_i = x_i$), use fewer knots.
- Then do ordinary linear regression on the basis function.
- Choice of knots can be difficult and arbitrary.
- Automatic knot selection algorithms very slow.

Regression splines

- Rather than have knots at each data point ($\kappa_i = x_i$), use fewer knots.
- Then do ordinary linear regression on the basis function.
- Choice of knots can be difficult and arbitrary.
- Automatic knot selection algorithms very slow.

$$\hat{\beta} = (B'B)^{-1}B'\mathbf{Y}.$$

Outline

Penalized spline regression

$$r(x) = \sum_{j=1}^{K} \beta_j h_j(x).$$

Idea: retain all the knots but constrain their influence by

$$\sum \beta_j^2 < C.$$

Penalized spline regression

$$\boxed{r(x) = \sum_{j=1}^{K} \beta_j h_j(x).}$$

Idea: retain all the knots but constrain their influence by

$$\sum \beta_j^2 < C.$$

Let

$$D = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times K} \\ \mathbf{0}_{K\times 2} & \mathbf{I}_{K\times K} \end{bmatrix}.$$

Then we want to

minimize $\|\mathbf{y} - \mathbf{B}\beta\|^2$ subject to $\beta'\mathbf{D}\beta \leq C$.

Penalized regression splines

A Lagrange multiplier argument shows that this is equivalent to minimizing

$$\|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|^2 + \lambda^2 \boldsymbol{\beta}' \mathbf{D} \boldsymbol{\beta}$$

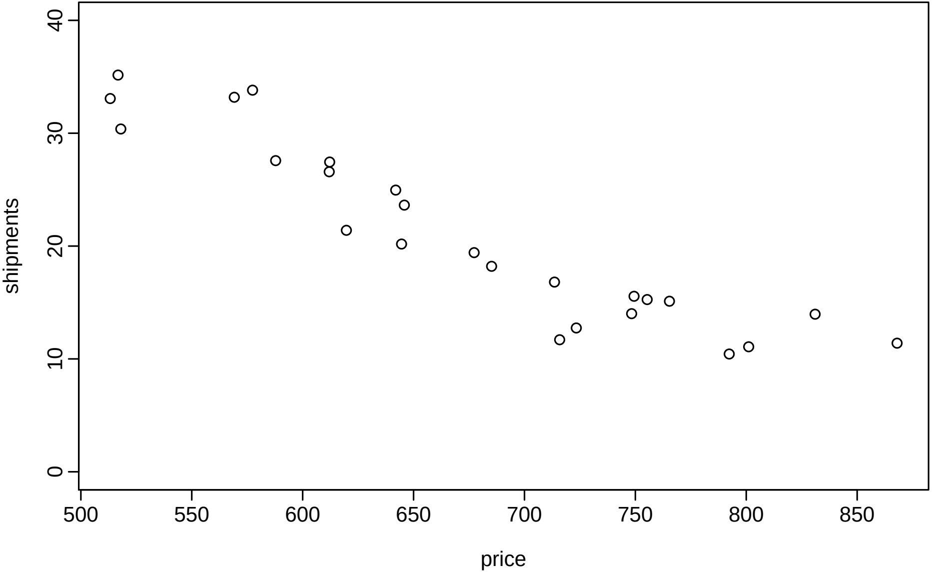
for some number $\lambda \geq 0$.

Solution: $\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}'\mathbf{X} + \lambda^2 \mathbf{D})^{-1} \mathbf{X}'\mathbf{y}$.

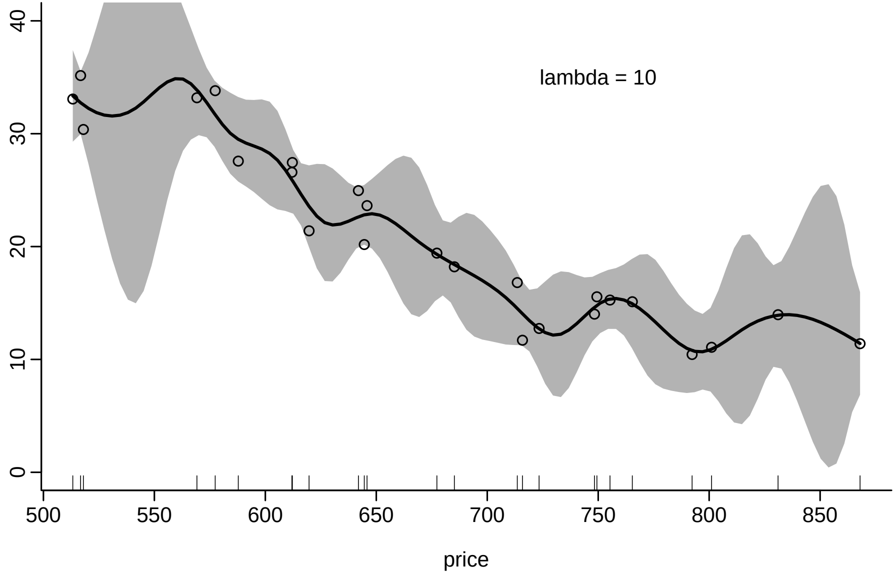
Fitted values: $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda^2 \mathbf{D})^{-1} \mathbf{X}'\mathbf{y}$.

- A type of ridge regression.

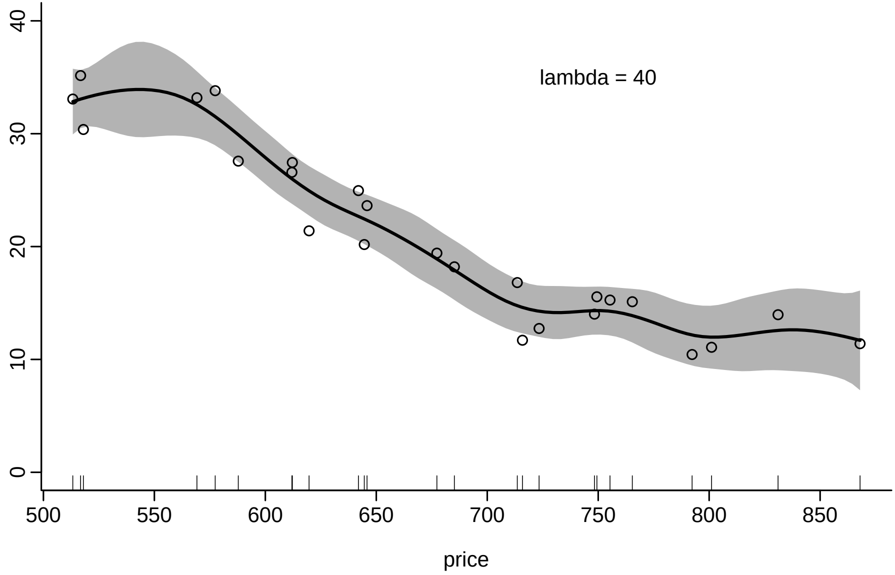
Penalized regression splines



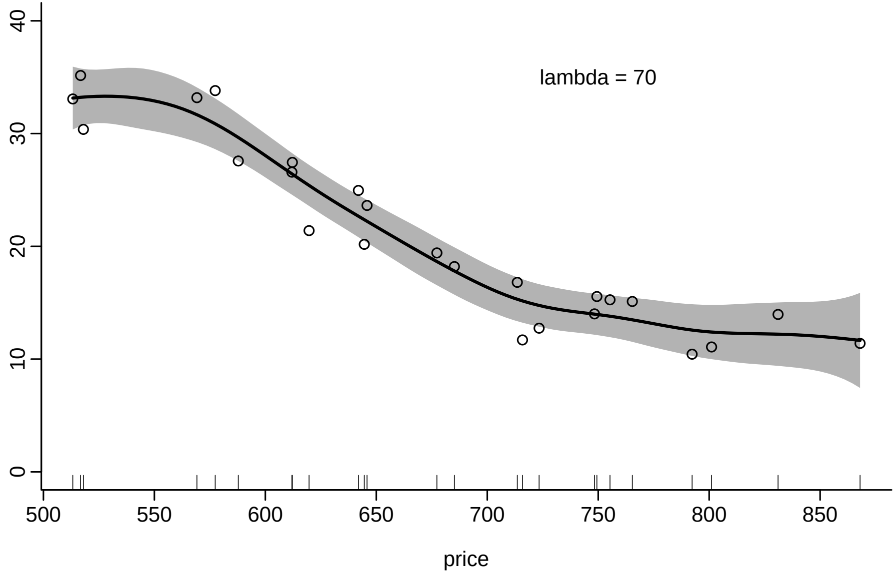
Penalized regression splines



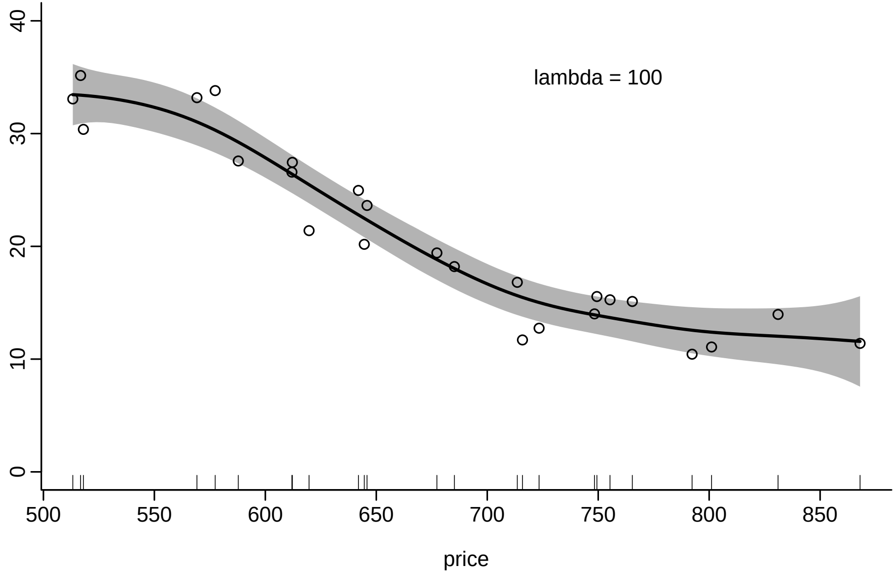
Penalized regression splines



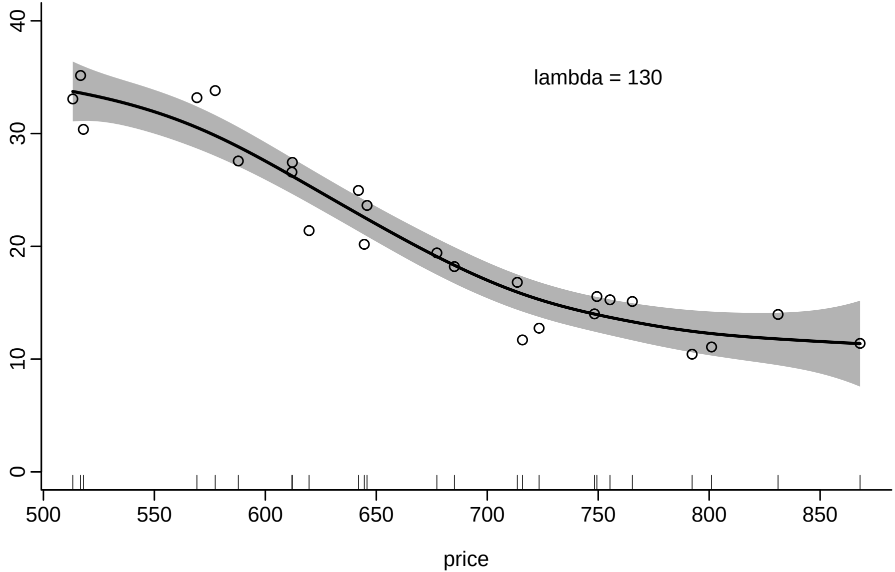
Penalized regression splines



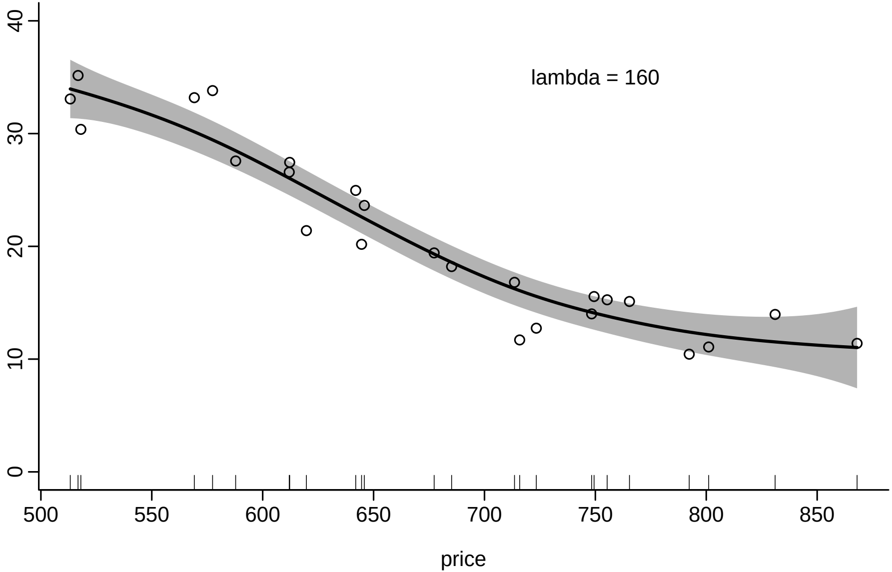
Penalized regression splines



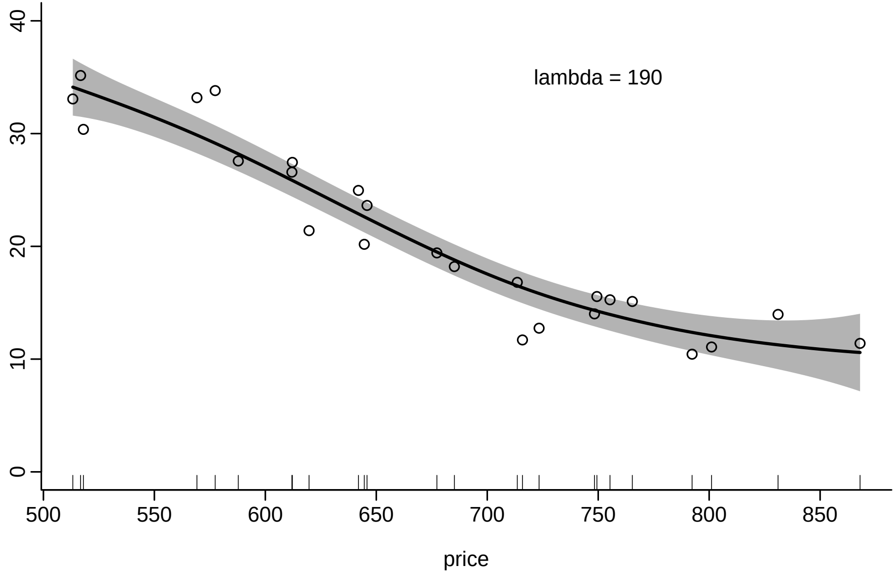
Penalized regression splines



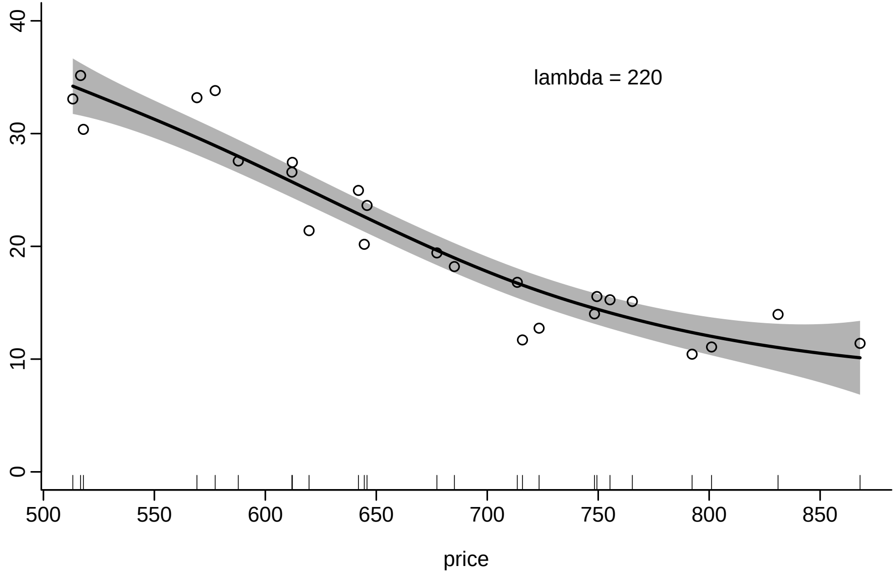
Penalized regression splines



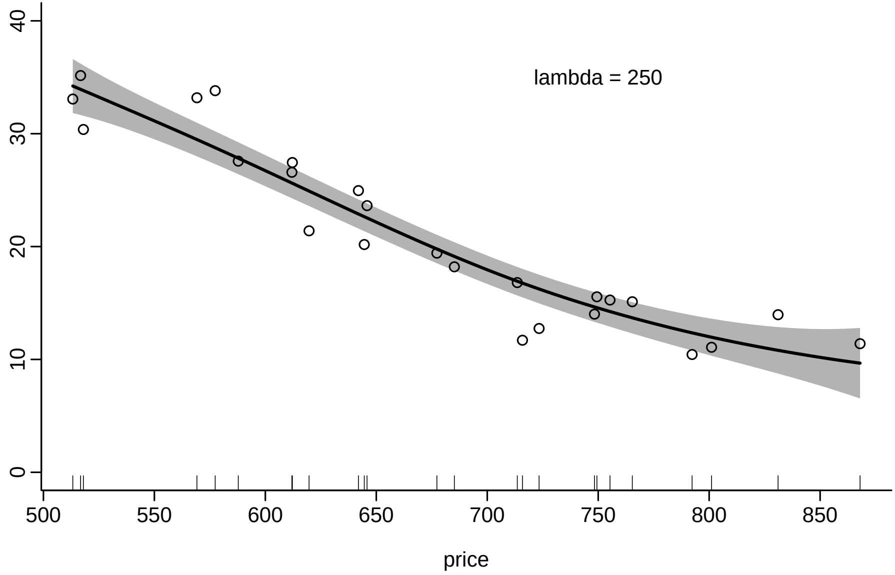
Penalized regression splines



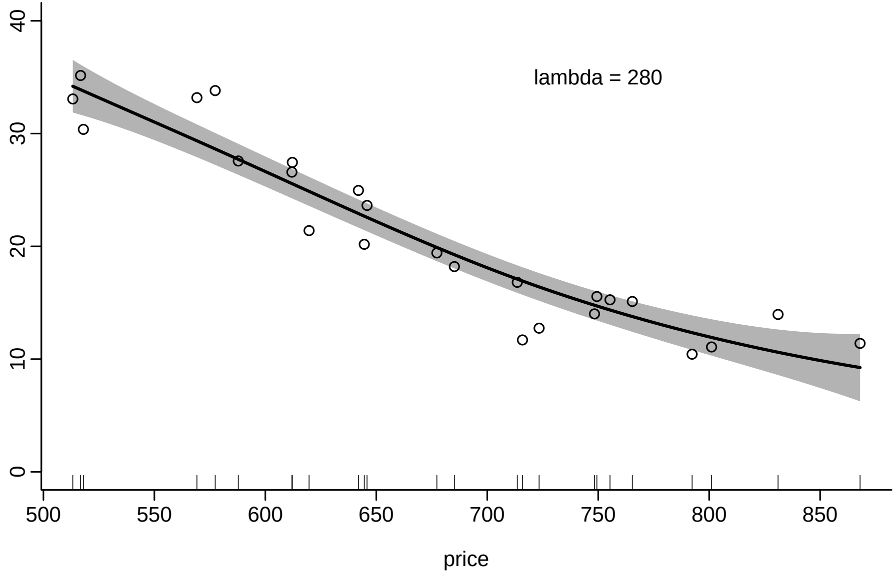
Penalized regression splines



Penalized regression splines



Penalized regression splines



Penalized regression splines

Implementation in R

```
require(SemiPar)
fit <- spm(shipments ~ f(price))
plot(fit)
points(price,shipments)
```

Penalized regression splines

Implementation in R

```
require(SemiPar)
fit <- spm(shipments ~ f(price))
plot(fit)
points(price,shipments)
```

For lots of examples and an introduction to the theory:

http://www.uow.edu.au/~mwand/SPmanu.pdf

Outline

Spline bases

Truncated power basis of degree p

$1,\ x,\ \dots,\ x^p,\ (x-\kappa_1)_+^p,\ \dots,\ (x-\kappa_K)_+^p$

Spline bases

Truncated power basis of degree p

$1, x, \ldots, x^p, (x-\kappa_1)_+^p, \ldots, (x-\kappa_K)_+^p$

- $p-1$ continuous derivatives

Spline bases

Truncated power basis of degree p

$1, x, \ldots, x^p, (x-\kappa_1)_+^p, \ldots, (x-\kappa_K)_+^p$

- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.

Spline bases

Truncated power basis of degree p

$1, x, \ldots, x^p, (x-\kappa_1)_+^p, \ldots, (x-\kappa_K)_+^p$

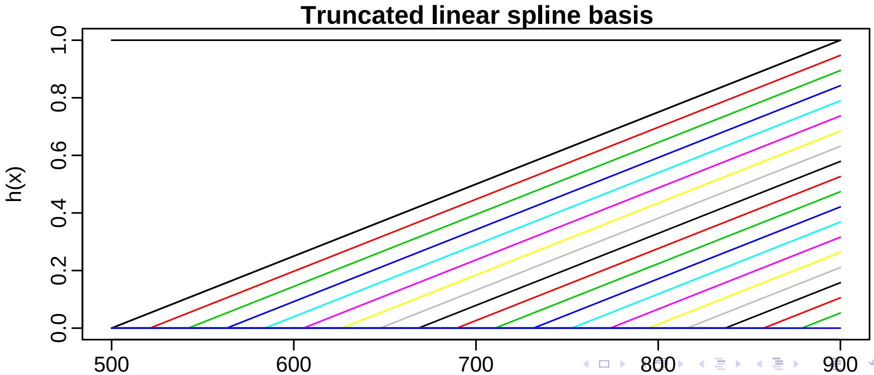
- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.

Spline bases

Truncated power basis of degree p

$$1, x, \ldots, x^p, (x-\kappa_1)_+^p, \ldots, (x-\kappa_K)_+^p$$

- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.

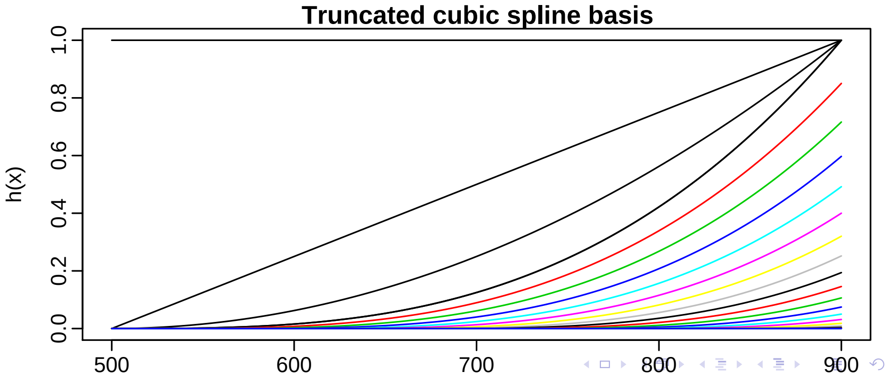


Spline bases

Truncated power basis of degree p

$1, x, \ldots, x^p, (x-\kappa_1)_+^p, \ldots, (x-\kappa_K)_+^p$

- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.



Spline bases

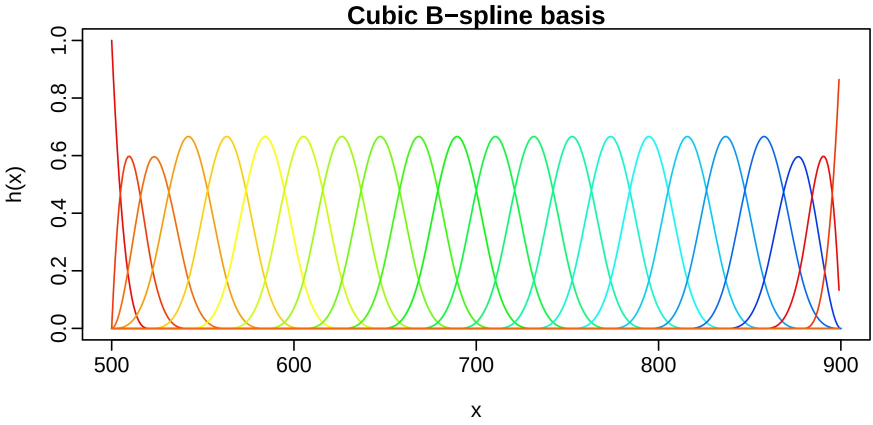
B-splines

- *Equivalent* to truncated power bases but with more stable properties.

Spline bases

B-splines

- *Equivalent* to truncated power bases but with more stable properties.



Spline bases

Radial basis functions of degree p

$$1,\ x,\ \dots,\ x^p,\ |x-\kappa_1|^p,\ \dots,\ |x-\kappa_K|^p$$

Spline bases

Radial basis functions of degree p

$1, x, \ldots, x^p, |x-\kappa_1|^p, \ldots, |x-\kappa_K|^p$

- $p-1$ continuous derivatives

Spline bases

Radial basis functions of degree p

$1, x, \dots, x^p, |x-\kappa_1|^p, \dots, |x-\kappa_K|^p$

- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.

Spline bases

Radial basis functions of degree p

$1, x, \ldots, x^p, |x-\kappa_1|^p, \ldots, |x-\kappa_K|^p$

- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.

Spline bases

Radial basis functions of degree p

$1, x, \ldots, x^p, |x-\kappa_1|^p, \ldots, |x-\kappa_K|^p$

- $p-1$ continuous derivatives
- In penalized regression splines, none of the polynomial coefficients is penalized.