PDF/SOLUTIONS

JSM
Salt Lake City, Utah
July 29 to August 2, 2007

`DierckxSpline`: An R Package For Minimal Knot Splines

Sundar Dorai-Raj
Spencer Graves

July 29, 2007

Agenda

- **Splines in R**
- **FITPACK Routines**
- **Univariate Splines**
  - Smoothing splines
  - Least square splines
  - Free knot splines
- **The `DierckxSpline` package for R**
- **Examples**
- **Software Status and Extensions**

Splines in R

- **Many algorithms have been improved since Dierckx**
  - Better free knot selection algorithms
  - Applications for functional data analysis

- **Purpose of the package is to make available Dierckx FITPACK functions**
  - Univariate splines
  - Free knot splines
  - Bivariate splines

- **R lacks a comprehensive spline package**
  - **`spline`**
  - **`smooth.spline`**
  - Several packages
    - **`splines`** – Spline package for B-splines
    - **`fda`** – Functional Data Analysis
    - **`ssr`** – Spline Smoothing Regression
  - No splines package for free knots or constrained splines

Dierckx FITPACK

- **The FITPACK library is available in Fortran from NETLIB**

  http://www.netlib.org/dierckx

- **Includes**
  - Code to accompany *Curve and Surface Fitting with Splines*

    Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Oxford Science Publications, New York.

  - Examples and data
  - Currently R package interfaces with approximately half of the provided functions

- **Not to be confused with commercial FITPACK library http://www.netlib.org/fitpack**

Smoothing Splines

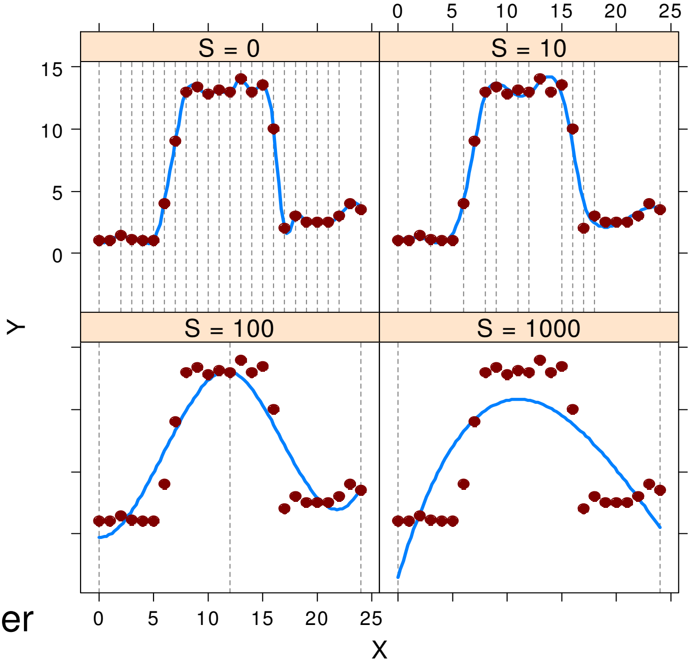
- **Given**
  - Data: (x_r, y_r), $r = 1, \ldots, m$
  - Constraints: $a \le x_r \le x_{r+1} \le b$
  - Weights: w_r
- **Goal**
  - Determine spline $s(x)$ on $[a, b]$
  - Degree: k
  - Knots: $a = \lambda_0, \lambda_1, \ldots, \lambda_g, \lambda_{g+1} = b$
- **Unconstrained minimization**

$$\text{Minimize } \tilde{\eta} := \sum_{i=1}^{g} \left(s^{(k)}(\lambda_i +) - s^{(k)}(\lambda_i -)\right)^2$$

$$\text{Subject to } \delta := \sum_{i=1}^{m} \left(w_r\left(y_r - s(x_r)\right)\right)^2 < S$$

  - where S is some user-specified smoothing parameter
  - Increase S → increase smoothing

```
ss <- list()
s <- c(0, 10, 100, 1000)
for(i in seq(s)) {
  ss[[i]] <- curfit(x, y,
    s = s[i], method = "ss")
}
```



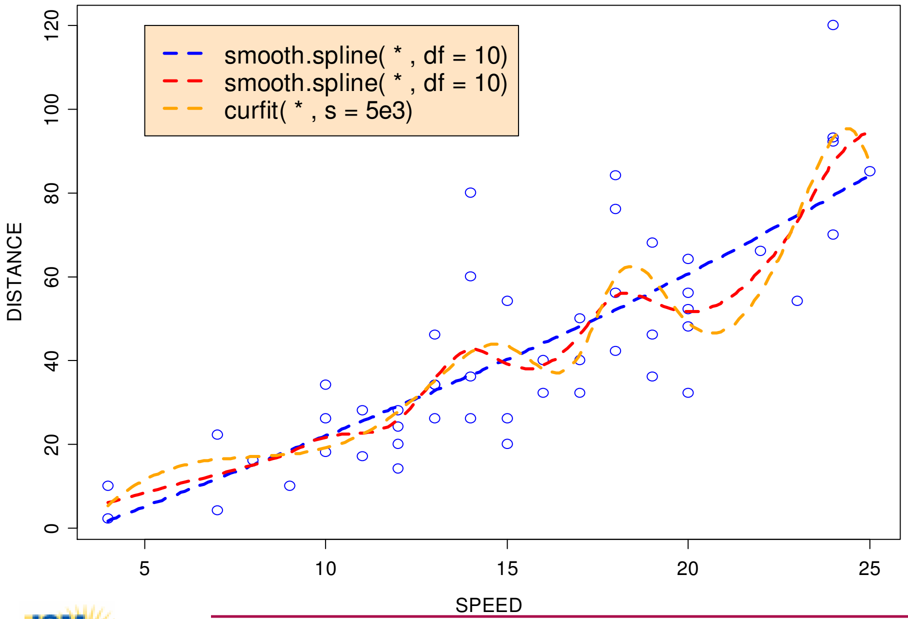
Vertical lines are knot placements

Comparison To `smooth.spline`

- R has `smooth.spline` which is a competing function for smoothing splines

```
## example from ?smooth.spline
## This example has duplicate points, so avoid cv = TRUE
cars.spl.0 <- smooth.spline(cars$speed, cars$dist)
cars.spl.1 <- smooth.spline(cars$speed, cars$dist, df = 10)
cars.spl.2 <- curfit(cars$speed, cars$dist, s = 5e3)
```

data(cars) & smoothing splines



`smooth.spline` uses cross validation or equivalent degrees of freedom to determine the amount of smoothing

`curfit` constrains the model deviance

Least Squares Splines With Fixed Knots

- **Fixed knots**
  - $a = \lambda_0, \lambda_1, \ldots, \lambda_g, \lambda_{g+1} = b$

- **Minimize**

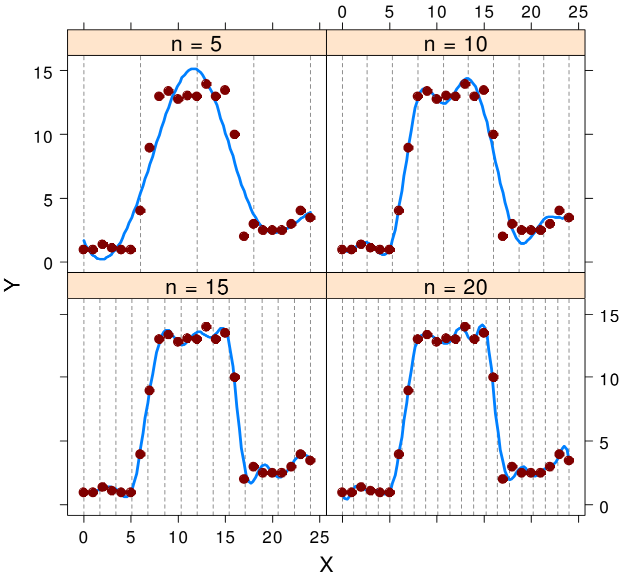
$$\delta = \sum_{r=1}^{m} \left(w_r\left(y_r - s(x_r)\right)\right)^2$$

$$= \sum_{r=1}^{m} \left(w_r y_r - \sum_{i=-k}^{g} c_i w_r N_{i,k+1}(x_r) \right)^2 ,$$

where $N_{i,k+1}$ are B-splines of degree k and c_i are the B-spline coefficients of $s(x)$

- **Knots are user-determined**
  - There is no known R equivalent
  - R function **`spline`** places a knot at each observation

```
n <- c(5, 10, 15, 25)
ls <- list()
for(i in seq(n)) {
  kn <- seq(0, 24, len = n[i])
  ls[[i]] <- curfit(x, y,
    method = "ls", knots = kn)
}
```

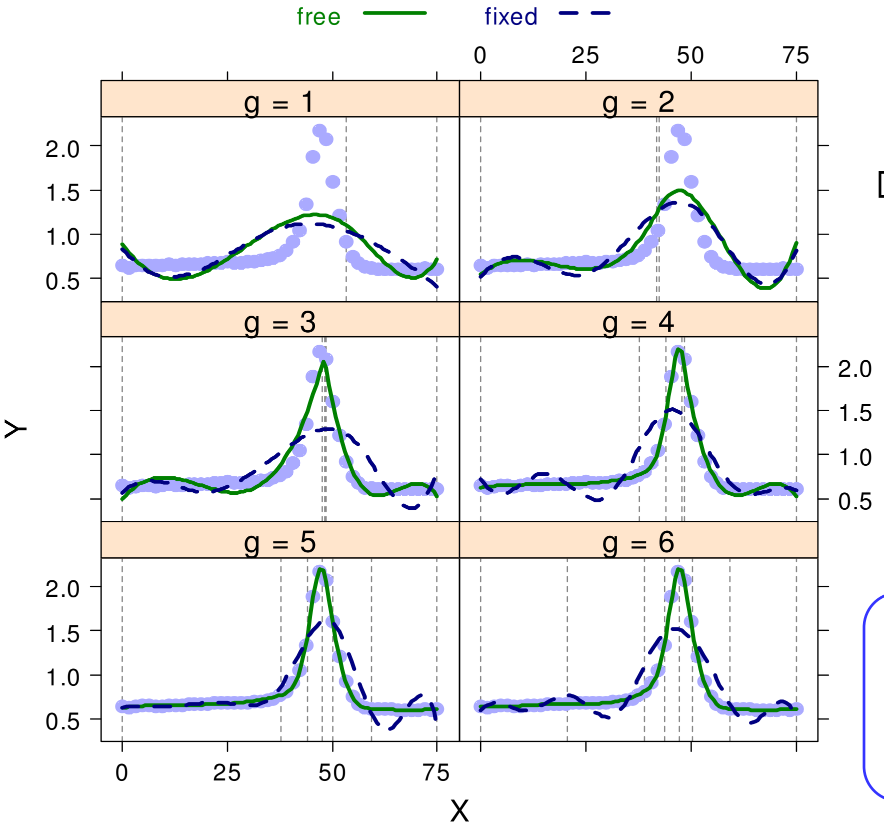


Vertical lines are knot placements

Least Squares Splines with Variable Knots

- **Titanium data (de Boors and Rice, 1968)**
  - Dierckx (1993) for optimizing number of knots
  - We use **`optim`** to minimize the residual sums of squares

```
data(titanium)
r <- curfit.free.knot(titanium$x2,
   titanium$y, g = 10, eps = 5e-4)
```



g	sigma	T
1	8.29E-02	5.95
2	4.81E-02	5.41
3	1.10E-02	5.01
4	1.56E-03	3.58
5	1.88E-04	-0.84
6	1.33E-04	-2.21

Optimal solution uses 5 interior knots

knots (5)
37.58
43.96
47.42
50.15
59.35

Plots include both fixed equally-spaced knots (dashed) and free knots (solid)

Selecting An Appropriate Number Of Knots

Algorithm described by Dierckx (1993)

- Supply starting value of $\lambda_1^0 =$ $(a + b)/2$ for the first knot, where $a = \min(x)$ and $b = \max(x)$
- Determine λ by minimizing a penalized RSS with user-defined ε and $g = \texttt{length}(\boldsymbol{\lambda})$

$$\xi(\boldsymbol{\lambda}) = RSS(\boldsymbol{\lambda}) + \varepsilon \frac{(b-a)RSS(\boldsymbol{\lambda}^0)}{(g+1)^2} \sum\nolimits_{j=0}^{g} (\lambda_{j+1} - \lambda_j)^{-1}$$

- For $j = 0, 1, 2, \ldots, g$, determine the region between knots with the largest RSS

$$RSS_j = \frac{1}{m - m_j} \sum_{i=q_j+1}^{q_j+m_j} (w_i(y_i - s_g(x_i)))^2,$$

where

$$\lambda_j \le x_{q_j+1} < x_{q_j+2} < \cdots < x_{q_j+m_j} \le \lambda_{j+1}$$

- Add a new knot at the midpoint of λ_j and λ_{j+1} where RSS_j is maximized

Stopping criteria

$$T_g = \frac{\sqrt{m-1} \sum\nolimits_{i=2}^{m} r_i r_{i-1}}{\sum\nolimits_{i=1}^{m} r_i^2}$$

- Number of optimal knots g is determined by the first $T_g < 0$

The `DierckxSpline` Package

Package Functions

- Includes interfaces for computing univariate splines
- FORTRAN for modeling bivariate

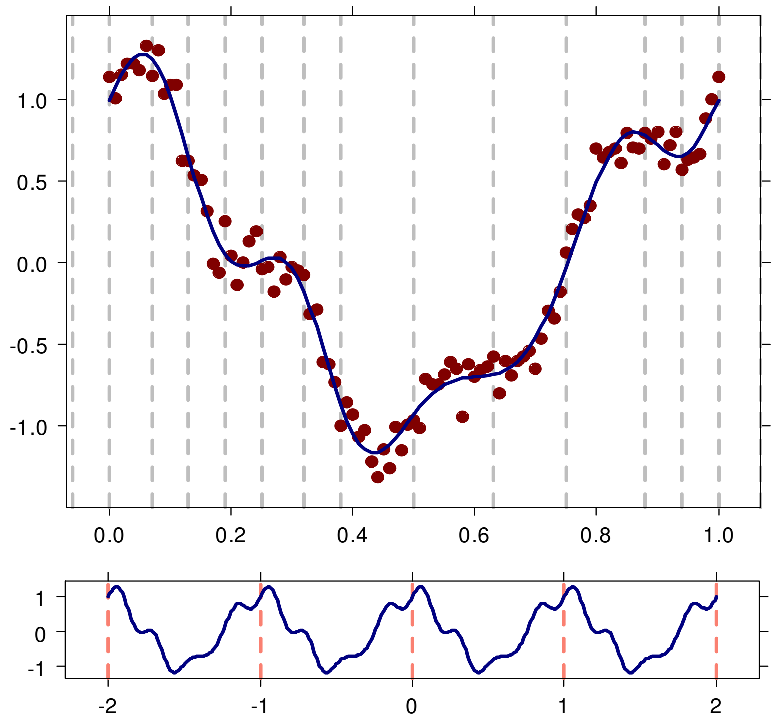
Examples and data included

- Data
  - De Boor and Rice (1968) **`titanium`**
  - Dierckx (1980) volumetric **`moisture`** content
  - Soudan and Dierckx (1979) **`knee`** flexion-extension during walking
  - Additional data extracted from FITPACK
- **`demo(DierckxSpline)`**
  - Includes examples with data discussed
- **`vignette(DierckxSpline)`**
  - Provides more details on the spline fitting and algorithms
  - Includes relevant sections from DierckxSpline (1993)

Example #1 – Smoothing With Periodic Splines

- **Quintic periodic smoothing spline**
  - Penalty: 90
  - Periodic: s(a) == s(b)

```
## periodic
set.seed(42)
n <- 100
r <- 1:n
x <- 0.01 * (r - 1)
e <- rnorm(n, 0, 0.1)
s2 <- var(e)
w <- rep(1/s2, n + 1)
y <- cos(2*pi*x) + 0.25*sin(8*pi*x) + e
x <- c(x, 1)
y <- c(y, y[1])
kn <- seq(0.01, 0.99, length = 12)
f1 <- percur(x, y, w, method = "ss",
             s = 90, k = 5)
```

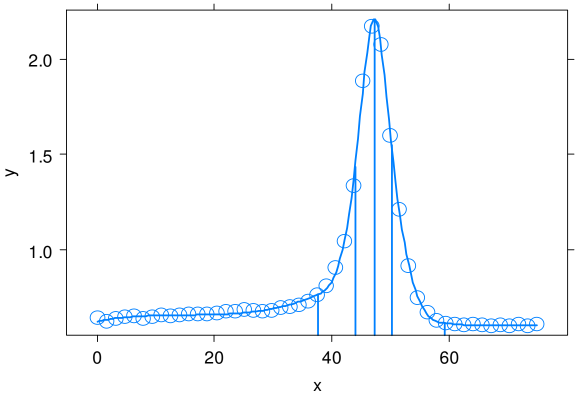


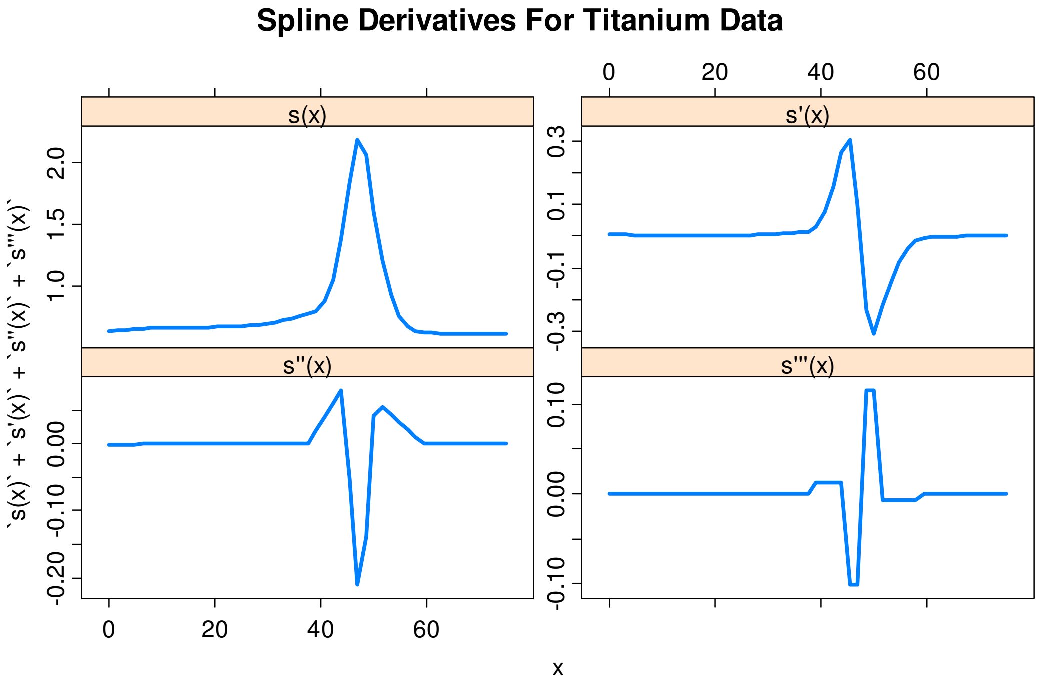
Example #2 – Differentiation With Free Knot Splines

- **Obtain analytical spline derivatives with `deriv` function**
  - Derivatives available from 0 (fitted spline value) to *k* (spline order)

```
data(titanium)
r <- curfit.free.knot(titanium$x2,
   titanium$y, g = 10, eps = 5e-4)
xyplot(r, show.knots = TRUE)

dr <- sapply(0:3, deriv,
  expr = r, at = titanium$x2)
```





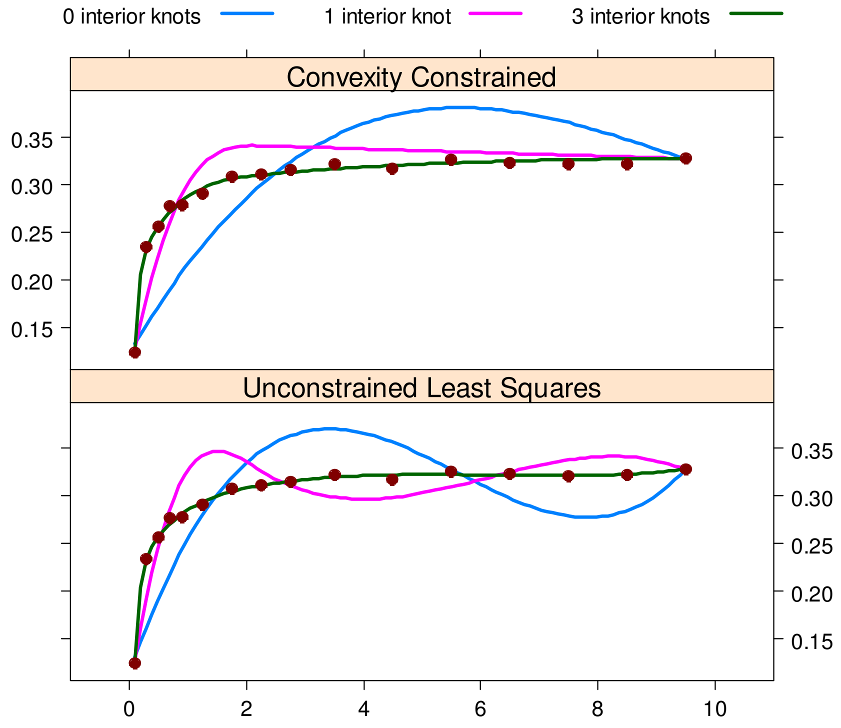
Example #3 – Splines With Convexity Constraints

- **Volumetric moisture content data (Dierckx 1980)**
  - Force convex constraints for all data points

```
## convexity contstraints
data(moisture)

f1 <- with(moisture,
  concon(x, y, w, v, s = 0.2))
f2 <- update(f1, s = 0.04)
f3 <- update(f1, s = 0.0002)

g1 <- with(moisture,
  curfit(x, y, w, method = "ls",
         knots = knots(f1)))
g2 <- update(g1, knots = knots(f2))
g3 <- update(g1, knots = knots(f3))
```



Example #4 – Profile Likelihood

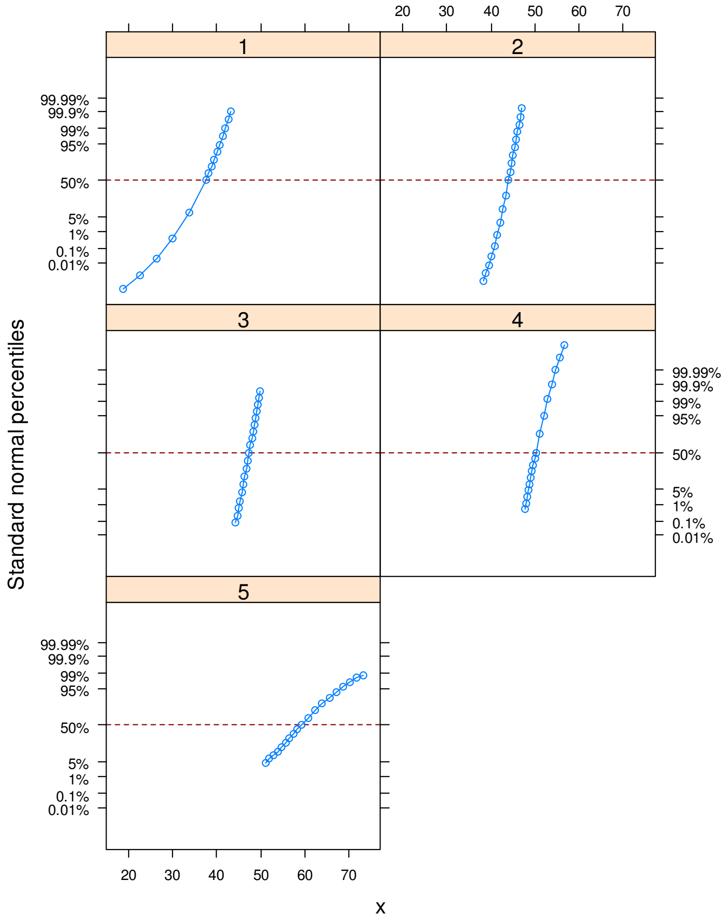
- Profiling the likelihood provides confidence intervals on knot placement

```
data(titanium)
r <- curfit.free.knot(titanium$x2,
   titanium$y, g = 10, eps = 5e-4)

pro <- confint(profile(r))
xyplot(pro)
```

	knots	2.50%	97.50%
1	37.58	32.35	41.42
2	43.98	42.00	45.80
3	47.37	45.54	49.17
4	50.19	48.26	52.36
*5	59.23	50.19	70.38

Lower bound for knot 5 is not achievable



Software Status And Extensions

- **Available for download from CRAN after JSM 2007**
  - Contact the author for bug reports and coding help
- **Bayesian Free Knots**
  - Adding priors to knot placement to stabilize free knot location
  - Allows mixture of free and fixed knots through prior information matrix
- **Create interfaces for remaining FITPACK routines**
- **Enhanced plotting for 3d splines with `lattice`**