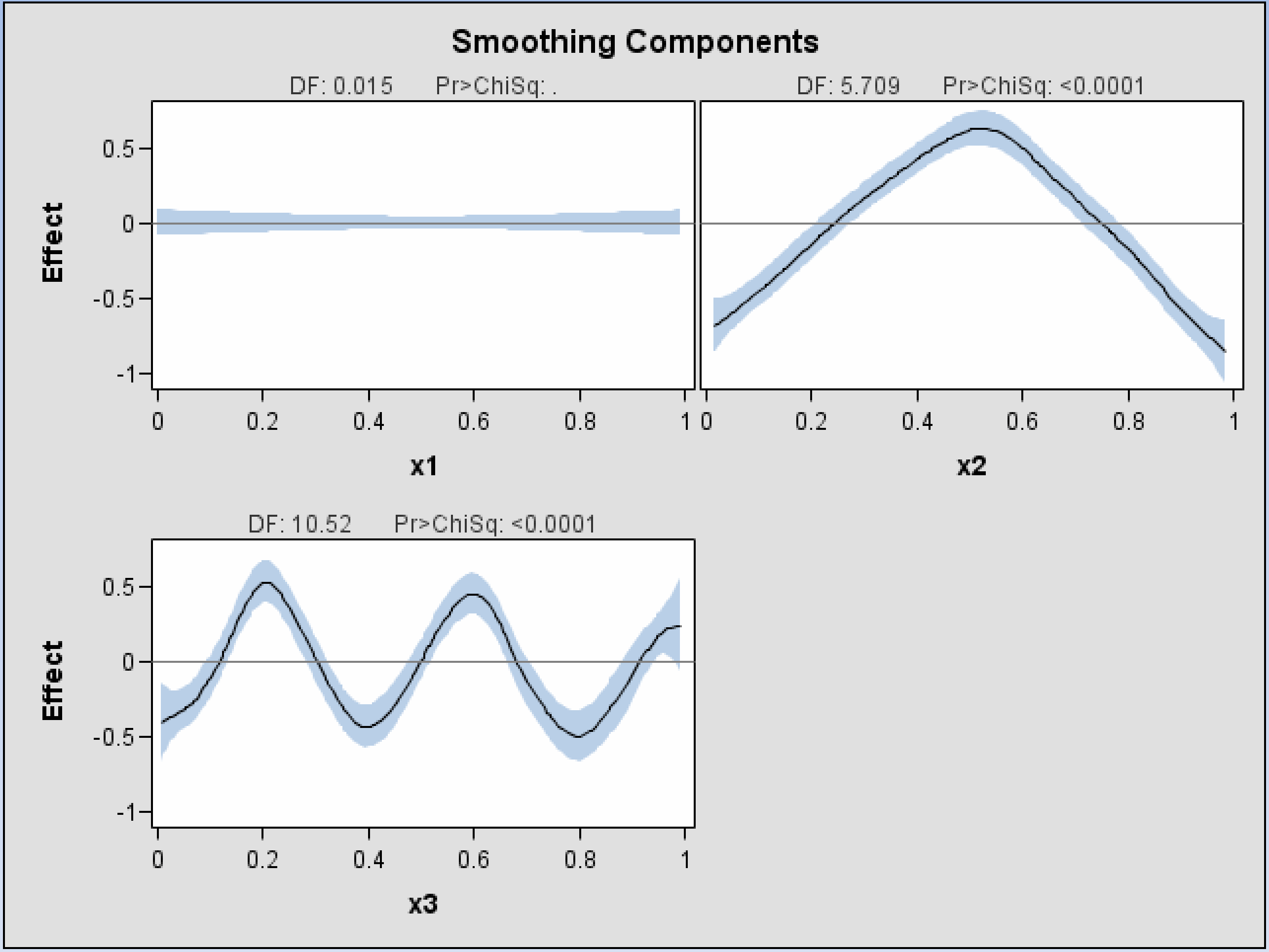
Modelling Splines and Generalized Additive Models with SAS



Peter Ott, Forest Analysis and Inventory
Branch, BC Ministry of FLNRO

Modelling Splines and GAMs with SAS

Part 1. Splines for Interpolating and Smoothing

- Intro and Motivation
- Splines and Basis Functions
- Penalized/Smoothing Splines
- Thin Plate Splines
- Connection with Mixed Models

Part 2. Generalized Additive Models (GAMs)

- Intro and Preliminaries
- Effective Degrees of Freedom (EDF) and Hypothesis Testing
- Partial Residuals
- Two examples

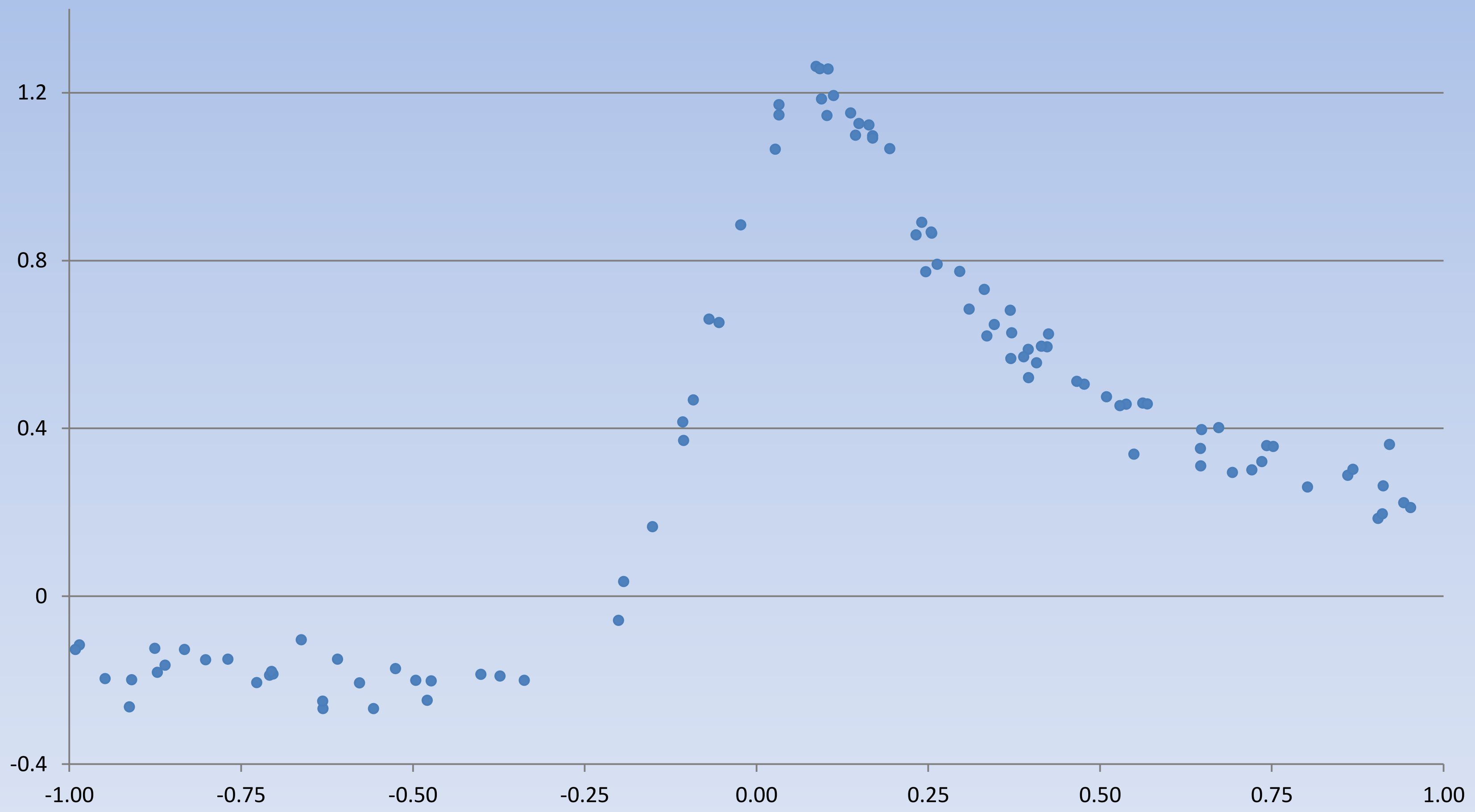
Table 1. Comparison of Features and Capabilties of SAS/STAT 9.1.3 Procedures for Penalized/Smoothing Spline Fitting

Conclusion

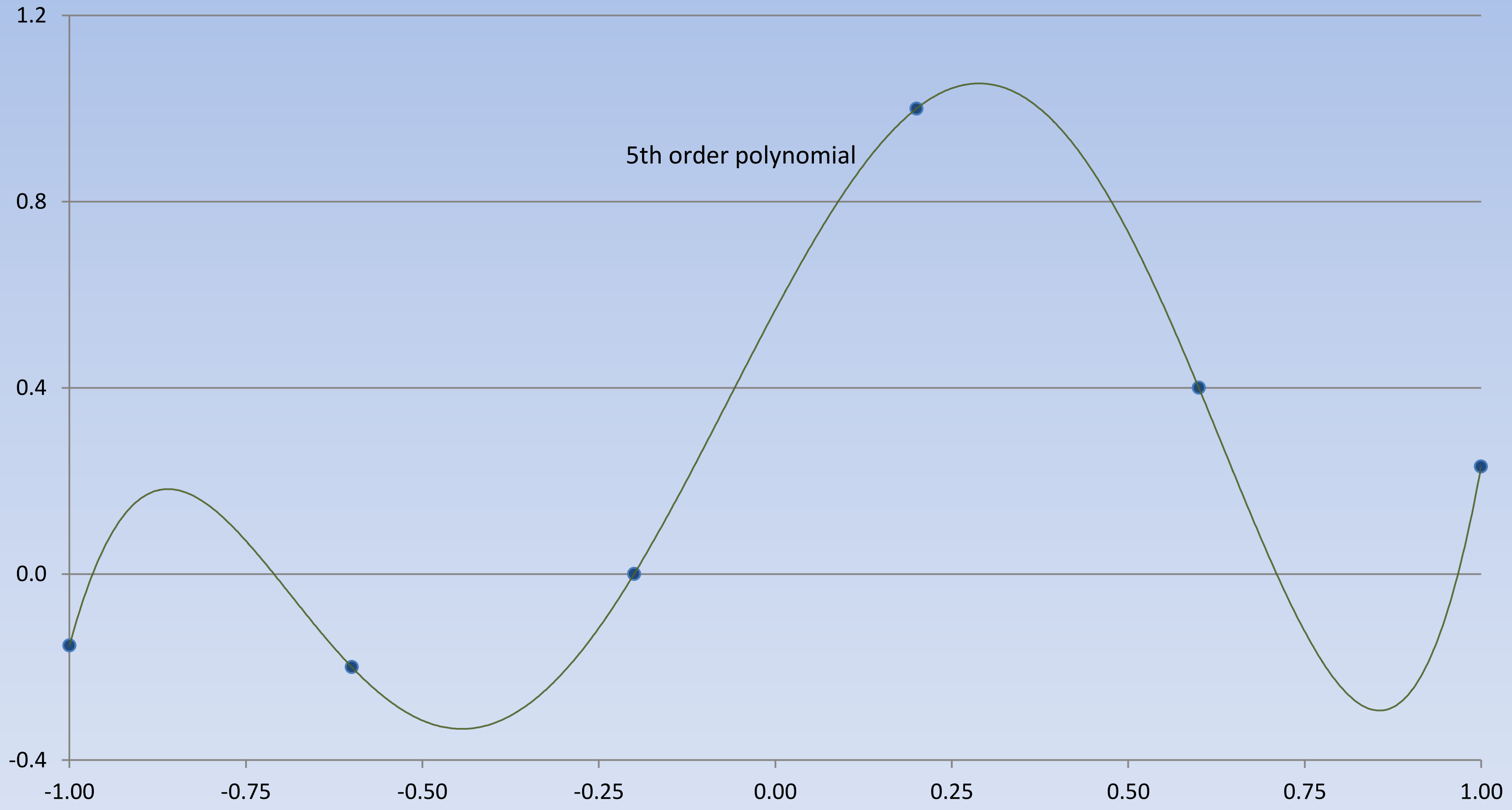
Intro – Part 1

- Part 1 deals with the situation where there is one dependent variable (y) and at least one independent variable (x)
- Relationship appears to be nonlinear
- Parametric form of relationship is not intuitively obvious or required
- Goal is to find function that can either interpolate points or fit a smooth curve or surface through them
- Higher order (≥ 4) polynomials are flawed due to undesired oscillation of interpolated fit between data points – Runge's phenomenon (Carl David Tolmé Runge)

“The dataset”



Runge’s Phenomenon



Splines

Splines – piecewise polynomials

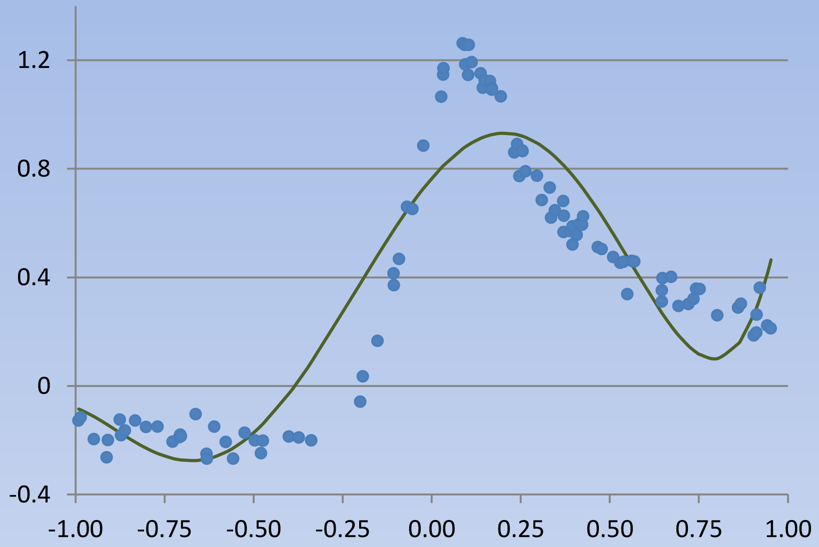
- Want to represent nonlinear *f* (or at least something very close to it) as a linear combination of *basis functions*:

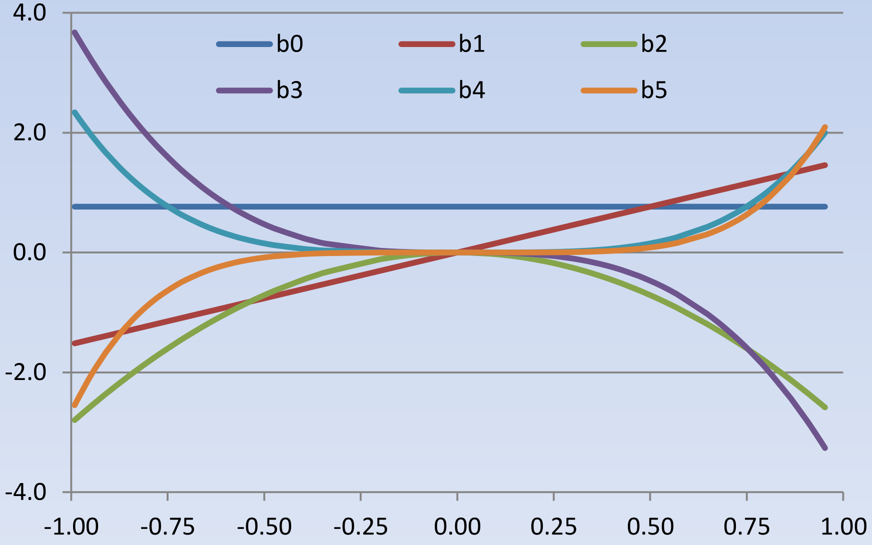
$$f(x) = \sum_{j=1}^{J} \beta_j \cdot b_j(x)$$

- Seen them before with multiple regression – the polynomial basis:

$$f(x_i) = \beta_0 \cdot 1 + \beta_1 \cdot x_i + \beta_2 \cdot x_i^2 + \beta_3 \cdot x_i^3 + \beta_4 \cdot x_i^4 + \beta_5 \cdot x_i^5$$

Polynomial basis





Splines

- Spline consists of sections of a polynomial joined together at pre-specified knots

- Value of y is usually equal at knot so curve is continuous, and sometimes first, second and higher order derivatives (WRT x) may also be equal

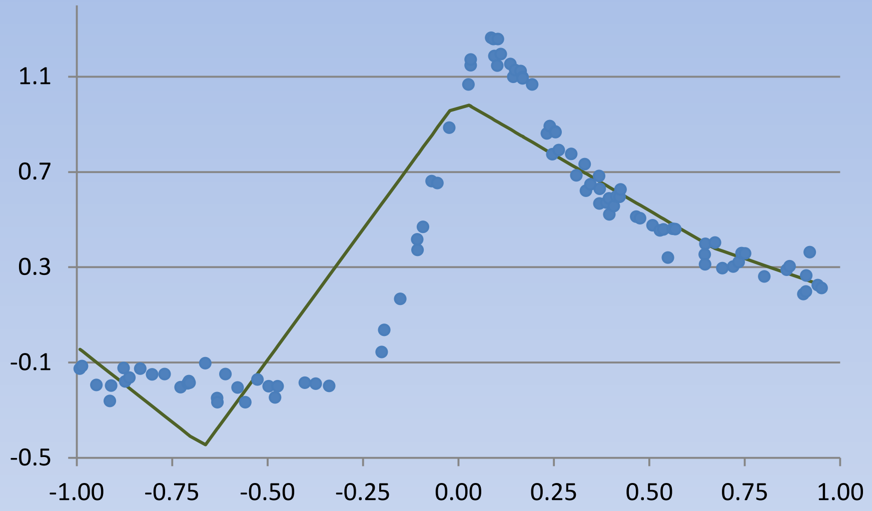
Truncated Power Function (TPF) basis example #1: *Linear* Spline Basis

Say we set-up 3 knots... κ_1, κ_2 and κ_3 then

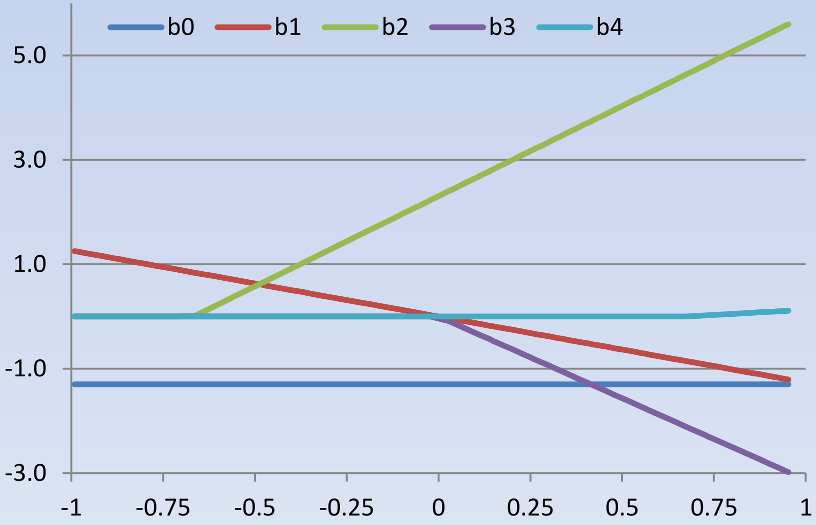
$$f(x) = \beta_0 \cdot 1 + \beta_1 \cdot x + \beta_2 \cdot (x - \kappa_1)_+ + \beta_3 \cdot (x - \kappa_2)_+ + \beta_4 \cdot (x - \kappa_3)_+$$

where $(x - \kappa_k)_+ = \begin{cases} 0 & x \le \kappa_k \\ x - \kappa_k & x > \kappa_k \end{cases}$

TPF basis example #1: *Linear Spline Basis*



```
proc transreg data=whatever;
 model identity(y)=pspline(x / degree=1
knots=-0.66667, 0, 0.66667);
 output out=pred predicted
coefficients;
title 'Linear tpf spline with 3 knots';
run;
```



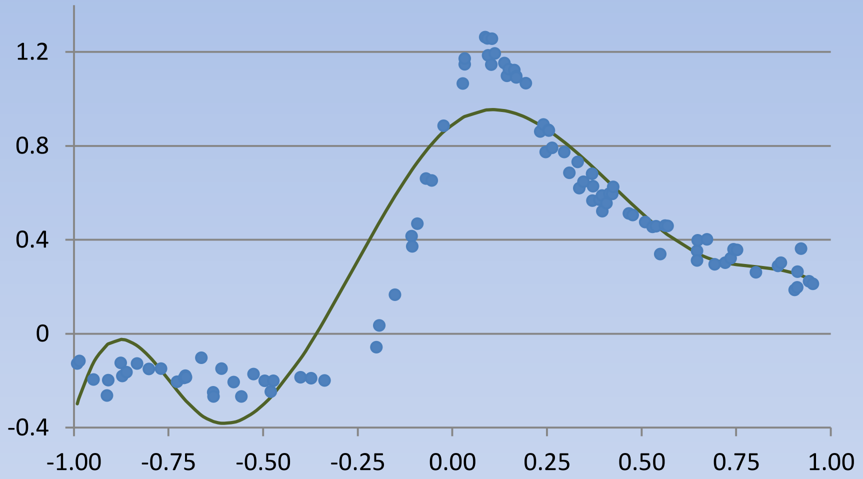
TPF basis example #2: *Cubic* Spline Basis

$$f(x) = \beta_0 \cdot 1 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3 + \\ \beta_4 \cdot (x - \kappa_1)_+^3 + \beta_5 \cdot (x - \kappa_2)_+^3 + \beta_6 \cdot (x - \kappa_3)_+^3$$

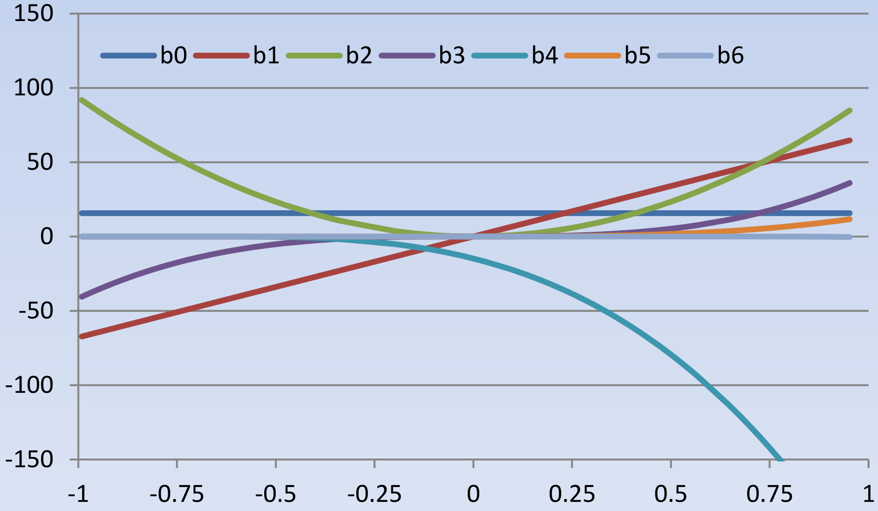
In general, the truncated power function (TPF) spline basis functions of degree q are:

$$1, x, x^2, \ldots, x^q, (x - \kappa_1)_+^q, (x - \kappa_2)_+^q, \ldots, (x - \kappa_K)_+^q$$

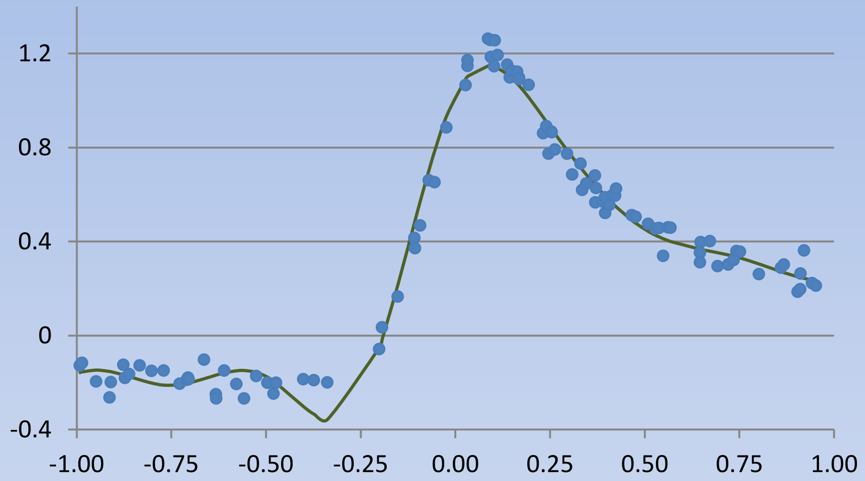
TPF basis example #2: *Cubic* Spline Basis



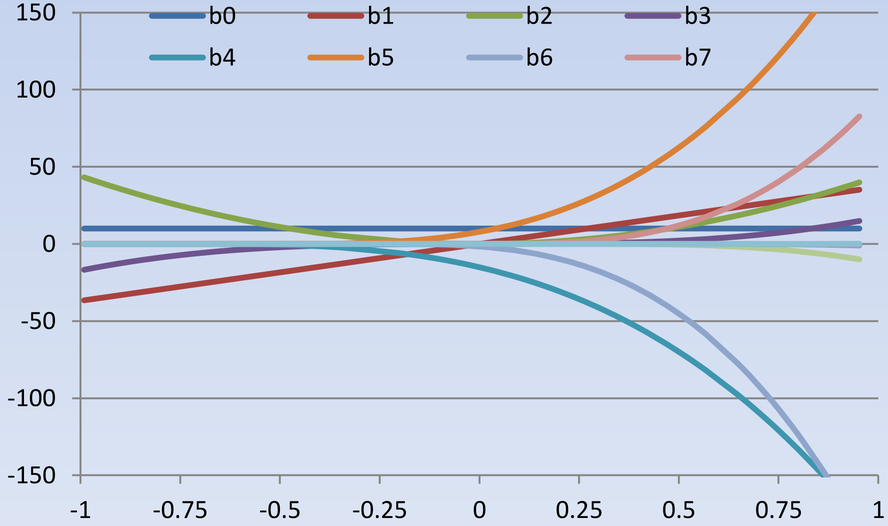
```
proc transreg data=whatever;
 model identity(y)=pspline(x / degree=3
knots=-0.66667, 0, 0.66667);
 output out=pred predicted
coefficients;
 title 'Cubic tpf spline with 3 knots';
run;
```



TPF basis example #2: *Cubic* Spline Basis



```
proc transreg data=whatever;
 model identity(y)=pspline(x / degree=3
knots= -0.75, -0.5, -0.25, 0, 0.25, 0.5,
0.75);
 output out=pred predicted coefficients;
 title 'Cubic tpf spline with 7 knots';
run;
```



General TPF basis

Can make these by hand, or use proc transreg or the effect statement in SAS 9.3*:

```
proc transreg data=whatever design;
 model pspline(x / degree=3 knots= -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75);
 id y;
 output out=tpfsplineout;
 title 'getting cubic TPF basis functions';
run;

data tpfsplineout;
 set whatever;
 array b{7} b1-b7;
 array knot{7} (-0.75 -0.50 -0.25 0 0.25 0.5 0.75);
 do i=1 to 7;
    if x <= knot[i] then b[i]=0;
    else b[i]=(x-knot[i])**3;
 end;
 title 'getting cubic TPF basis functions';
run;

effect spl_x=spline(x / knots= -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75);
```

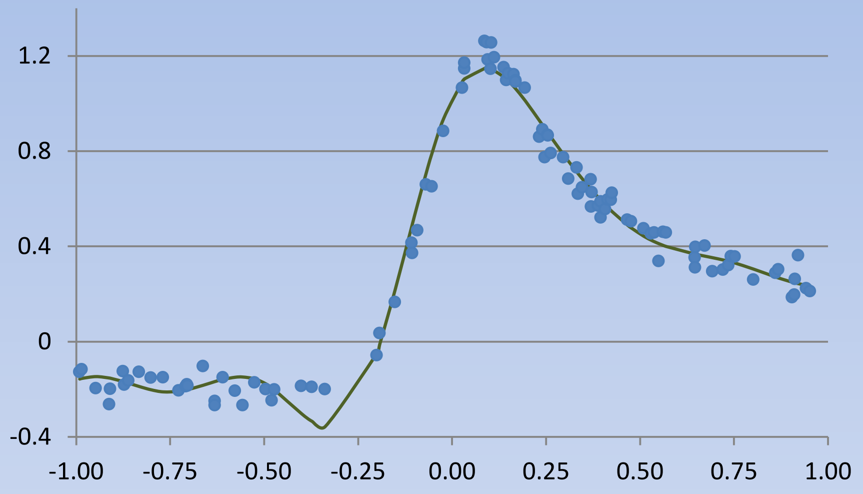
*The effect statement is available in these SAS 9.3 procedures: hpmixed, glimmix, glmselect, logistic, orthoreg, phreg, pls, quantreg, robustreg, surveylogistic, surveyreg

General TPF basis

At this point, two things should (hopefully!) be obvious:

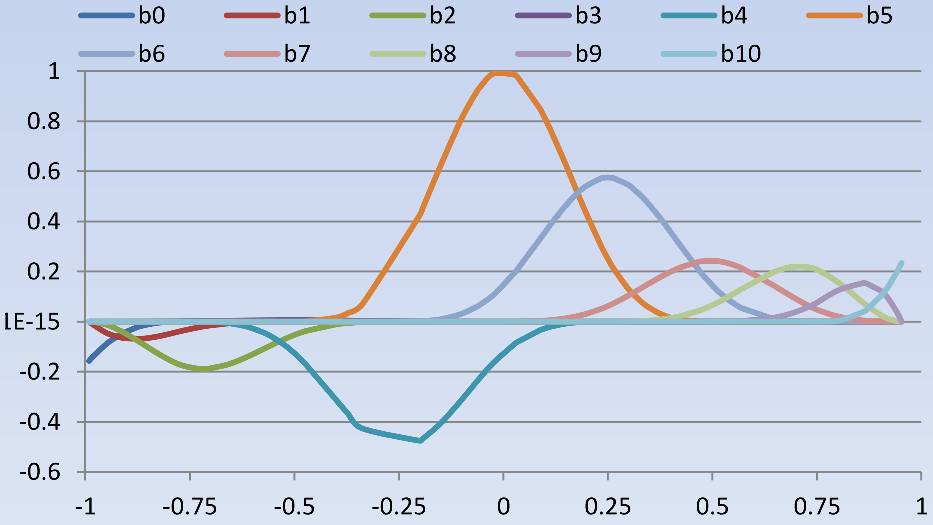
1. Could fit these in any software that fits linear models (e.g. proc glm, proc reg, proc quantreg, proc glmselect, etc.)
2. A predictive equation is available, although not very pretty

Cubic B-Spline Basis



```
proc transreg data=whatever;
 model identity(y)=bspline(x / degree=3
knots= -0.75, -0.5, -0.25, 0, 0.25, 0.5,
0.75) / details;
 output out=pred predicted coefficients;
 title 'Cubic B-spline basis regression
spline with 7 knots';
run;
```

```
proc transreg data=whatever design;
 model bspline(x / degree=3 knots= -0.75,
-0.5, -0.25, 0, 0.25, 0.5, 0.75);
 output out=bsplineout;
 title 'getting cubic b-spline basis
functions';
run;
```



Knot number and placement

- Placement where there is data, evenly spaced along x, or along quantiles of x

- Number and placement of knots may have dramatic effect on fit and smoothness. Lots of work in this area and it's hard to know the optimum number.

- In general, the placement is not as critical to model fit as the number of knots

- Splines we have looked at so far are called *regression splines* or *interpolating splines*. We will now discuss *penalized splines* or *smoothing splines* – which involve a penalty for wiggliness.

Penalized (Smoothing) Splines

Instead of estimating parameters by minimizing the sum of squares, add a penalty for wiggliness:

$$L = \sum_{i=1}^{n} \left[y_i - f\left(x_i\right) \right]^2 + \lambda \int f''\left(x\right)^2 dx$$

where $\lambda \geq 0$ is an unknown parameter that controls the wiggliness or roughness

$\lambda \to 0$ results in an unpenalized regression spline

$\lambda \to \infty$ results in a straight-line estimate of $f(x)$

How to choose λ? Want smoothing spline to capture signal but not the noise.

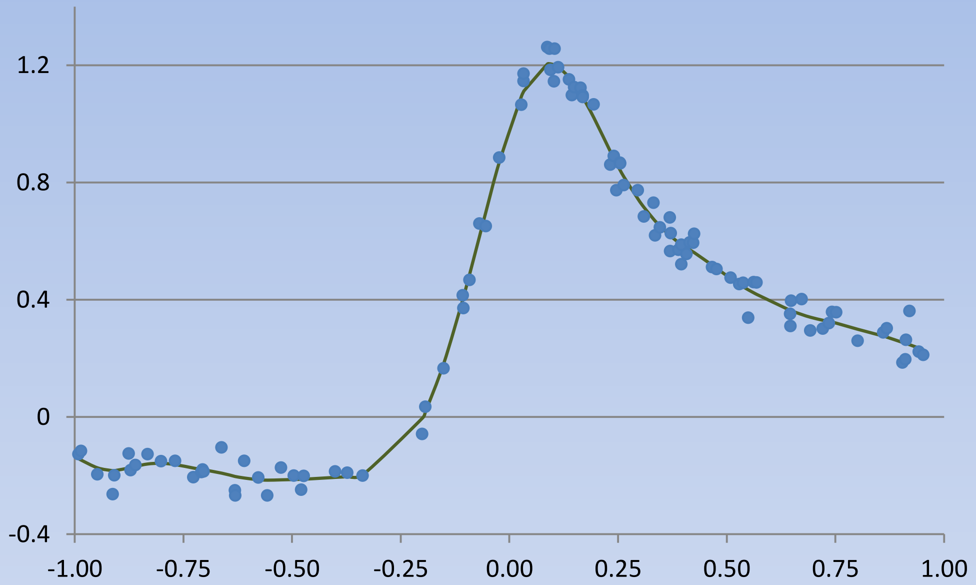
Generalized Cross Validation (GCV)

- Choose λ that yields the lowest GCV score:

$$\nu_g = \frac{n\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2}{\left[n - tr(\mathbf{H})\right]^2}$$

where $\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X} + \boldsymbol{\lambda}\mathbf{S}\right)^{-1}\mathbf{X}'$ and $\mathbf{S}$ is a $p \times p$ matrix describing the relationship between the coefficients of the basis functions in the penalty term (i.e. $\int f''(x_i)^2 dx = \boldsymbol{\beta}'\mathbf{S}\boldsymbol{\beta}$).

Generalized Cross Validation (GCV)



```
proc gam data=whatever;
 model y=spline(x) / method=gcv;
 output out=pred_gam predicted;
 title 'Penalized cubic spline';
run;
```

Thinplate Splines (Duchon 1977)

Forget about knots and basis functions for a second...

Say we just want a function to minimize:

$$L = \sum_{i=1}^{n} \left[y_i - g(x_i) \right]^2 + \lambda \int g''(x)^2 dx \quad \text{or} \quad L = \sum_{i=1}^{n} \left[y_i - g(\mathbf{x_i}) \right]^2 + \lambda \cdot J(g)$$

For example, if *d*=2:

$$J(g) = \iint \left[\left(\frac{\partial^2 g}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 g}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 g}{\partial x_2^2} \right)^2 \right] dx_1 dx_2$$

Thinplate Splines

Solution to this problem is a function that relies on *radial basis functions* and function passes through data without knots.

d=1: $$f(x) = \alpha_0 + \alpha_1 x + \frac{1}{12}\sum_{i=1}^{n} \beta_i \cdot \left|x - x_i\right|^3$$

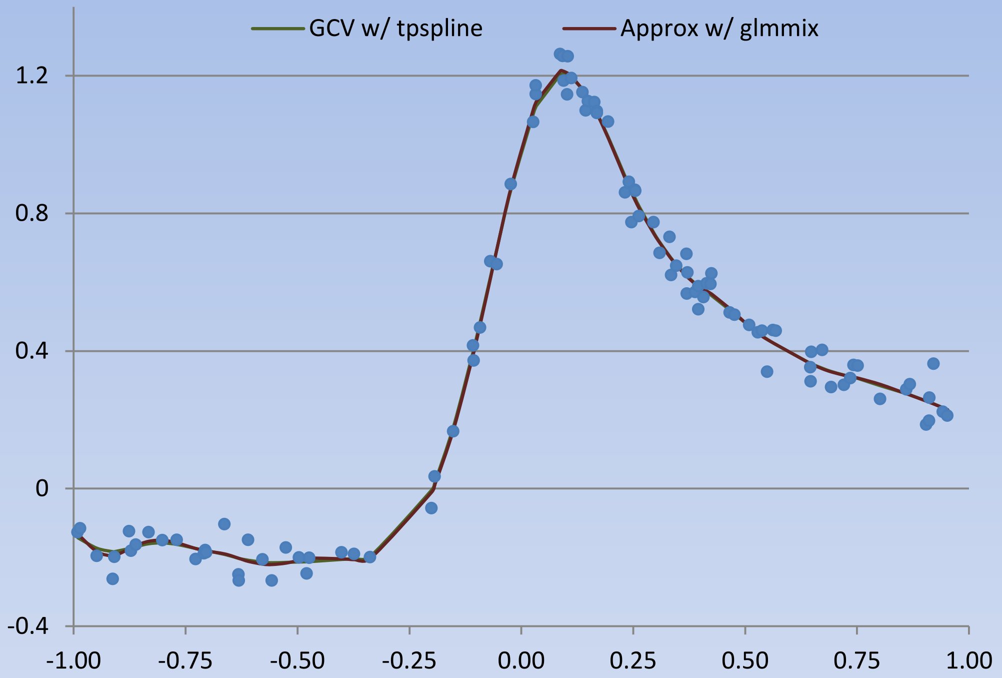
d=2: $$f(x_1, x_2) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \frac{1}{8\pi}\sum_{i=1}^{n} \beta_i \cdot z_i^{\,2} \log(z_i)$$

where $z_i = \sqrt{(x_1 - x_{1i})^2 + (x_2 - x_{2i})^2}$

Thinplate Splines

- Still need to estimate λ , usually via GCV
- Original derivation was for $d \geq 2$
- Works great but computational cost is high, especially with more than one predictor. Number of free parameters is equal to the number of unique predictor combinations.
- For this reason often see knot-based approximation, which looks very similar to smoothing spline with radius basis functions
- Only one smoothing penalty (λ), so wiggliness must be isotropic, and scale of predictors must be chosen carefully

Thinplate splines



```
proc tpspline data=whatever;
 model y = (x);
 output out=pred1 pred std coef;
 title 'Thin plate spline';
run;

proc glimmix data=whatever;
 model y = x / s dist=normal
link=identity;
 random x / type=rsmooth;
 output out=pred_smooth
pred=predicted;
title 'Approx thin plate spline';
run;
```

Connection with Mixed Models

- Smoothing splines can be cast as mixed models, where the basis functions that do not depend on the knots are the fixed effects, and the ones that are functions of the knots are the random effects

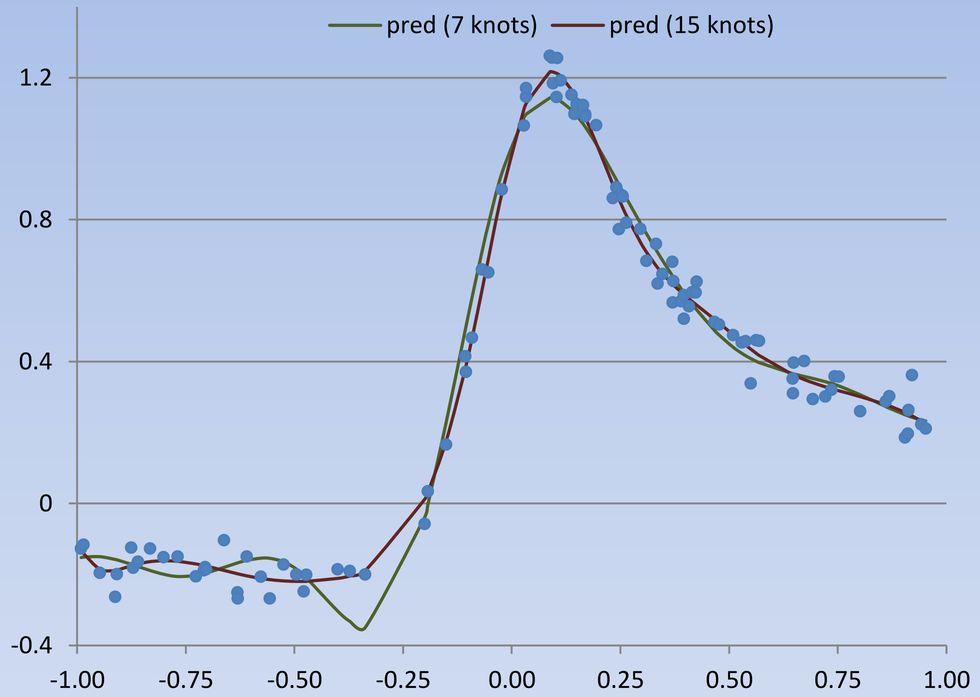
- TPF (cubic) with *d*=1: $\overbrace{1, x, x^2, x^3}^{\text{fixed}}, \overbrace{(x-\kappa_1)_+^3, (x-\kappa_2)_+^3, \ldots, (x-\kappa_K)_+^3}^{\text{random}}$

- Knot based radial with *d*=1: $\overbrace{1, x}^{\text{fixed}}, \overbrace{|x-\kappa_1|^3, |x-\kappa_2|^3, \ldots, |x-\kappa_K|^3}^{\text{random}}$

Connection with Mixed Models

- The resulting variance component associated with the ‘knot effects’ is equal to $\sigma_u^2 = \sigma_e^2 / \lambda^{2q}$ (recall that q is the degree of the basis), so we can estimate the β_j ’s and λ simultaneously
- Can add other effects, error structures, any other embellishments do-able with mixed models

Connection with Mixed Models



```
proc transreg data=whatever design;
 model pspline(x / degree=3 nknots=15);
*lots of knots;
 id y;
 output out=tpfsplineout;
 title 'getting cubic TPF basis
functions';
run;

proc mixed data=tpfsplineout;
 model y=x_1 x_2 x_3 / s
outp=predicted;
 random x_4-x_18 / type=toep(1) s;
 title 'Penalized cubic spline';
run;
```

Intro – Part 2 (GAMs)

A GAM is a generalized linear model that involves a sum of smooth functions (e.g. penalized splines) as its linear predictor:

$$y_i \sim \text{an exponential family member}$$

$$\mu_i = E\left(y_i \mid x_{1i}, x_{2i}, \dots, x_{pi}\right)$$

$$g(\mu_i) = \eta_i = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \cdots + f_p\left(x_{pi}\right)$$

Given a λ_j for each smooth component, estimate parameters by maximizing penalized log-likelihood

Effective Degrees of Freedom (EDF)

The influence or hat matrix for a GAM is:

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X} + \sum_i \lambda_i \mathbf{S}_i\right)^{-1}\mathbf{X}'$$

Similar to classical multiple regression, this $n \times n$ matrix has some interesting features:

- it is idempotent: $\mathbf{HH} = \mathbf{H}$
- it is diagonal elements are bounded: $0 \le h_{ii} \le 1$
- it is a projection matrix: $\hat{\mathbf{y}} = \mathbf{Hy}$
- $tr(\mathbf{H}) = EDF$ indicates the cost of projection

Comparing Gam Models

Inference is generally approximate and relies on:

- Models being compared are ‘nested’
- Asymptotic distributional assumptions
- The λ_j are known, i.e. must condition inference on chosen values for penalized splines

Deviance-based tests utilize ΔEDF and either χ^2 (e.g. Poisson, binomial) or *F* (e.g. normal, gamma) statistics

Mixed model tests include likelihood ratio and Wald-type *F* statistic

Partial Residuals

- How to graphically depict the relationship between the dependent variable and each independent variable, given that other independent variables are also in the model?
- Luckily, most GAMs are additive on the scale of the link function:

$$g(\mu_i) = \eta_i = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \cdots + f_p\left(x_{pi}\right)$$

Partial Residuals

- For the normal model where $g(\mu_i) = \mu_i$. The partial residual for x_1 is the leftover variation not accounted for by x_1: $$\begin{aligned} \hat{r}_{1i} &= y_i - \left(\hat{\beta}_0 + \hat{f}_2(x_{2i}) + \cdots + \hat{f}_p(x_{pi})\right) \\ &= y_i - \hat{y}_i + \hat{f}_1(x_{1i}) \\ &= \hat{e}_i + \hat{f}_1(x_{1i}) \end{aligned}$$

- For non-normal models, the partial residual for x_1 is: $$\hat{r}_{1i} = (y_i - \hat{\mu}_i)g^{'}(\hat{\mu}_i) + \hat{f}_1(x_{1i})$$

- In either case, the fitted component that explains this variation is just: $\hat{f}_1(x_{1i})$

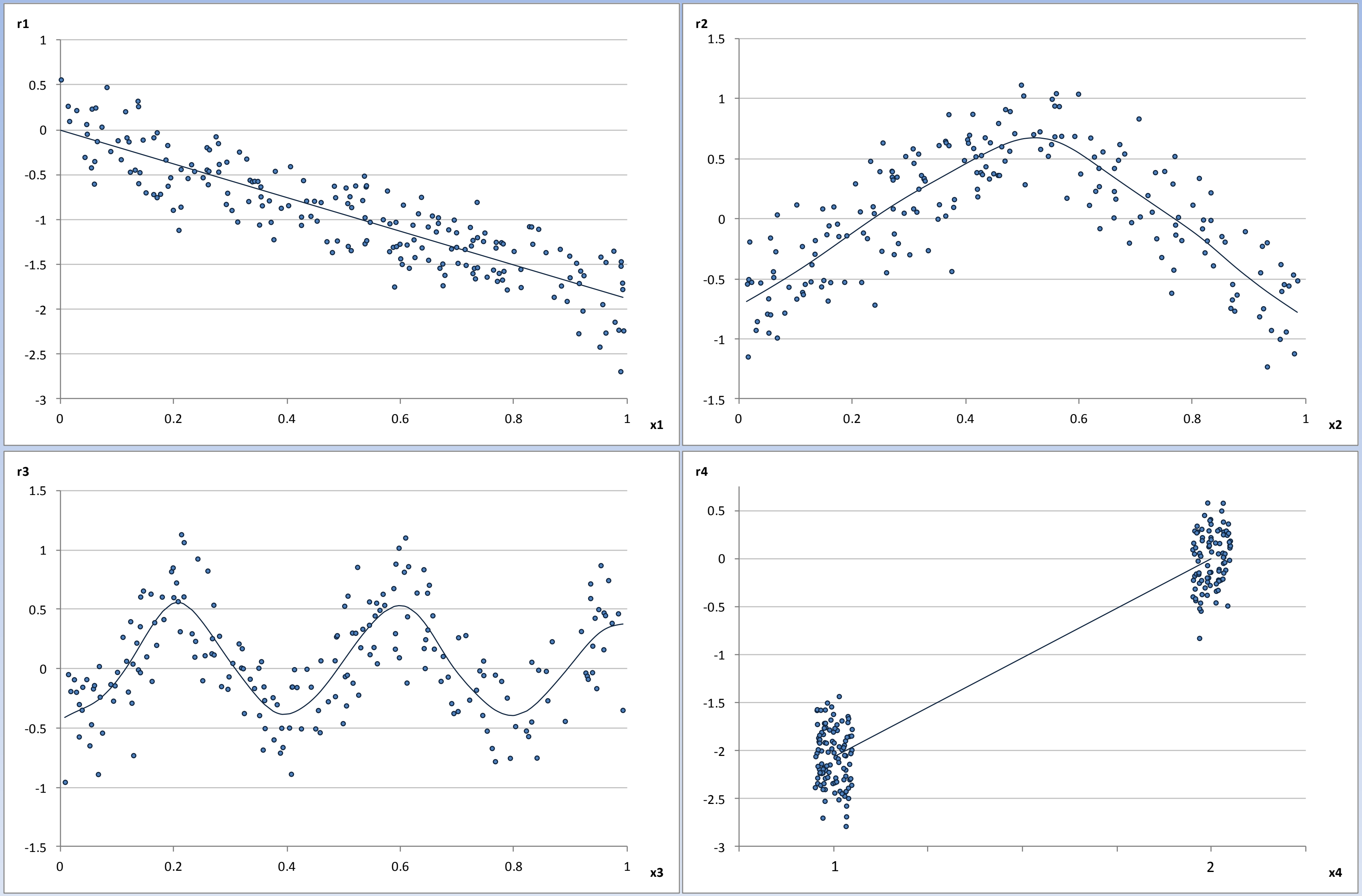
Two Examples

- Example 1. Showcase of partial residuals for normal errors and assessing the effect of different additive components (Proc Gam)
- Example 2. Testing whether two curves are different (Proc Mixed), using a variable coefficient model

Example 1. Analysis of Deviance

Variable Removed	Deviance	DF	ΔDeviance	ΔDF	$\hat{\phi}$	F	P
x4	198.95	15.09	182.33	6.15		318.76	2.09E-92
x1	68.82	17.53	52.19	3.71		151.19	1.02E-48
x2	53.41	12.45	36.78	8.79		44.97	7.05E-39
x3	37.77	9.71	21.14	11.53		19.71	1.49E-25
None (full model)	16.63	21.24			0.093		

Example 1. Partial Residuals and fitted components for GAM



Example 2. GAMM fit of variable coefficient model

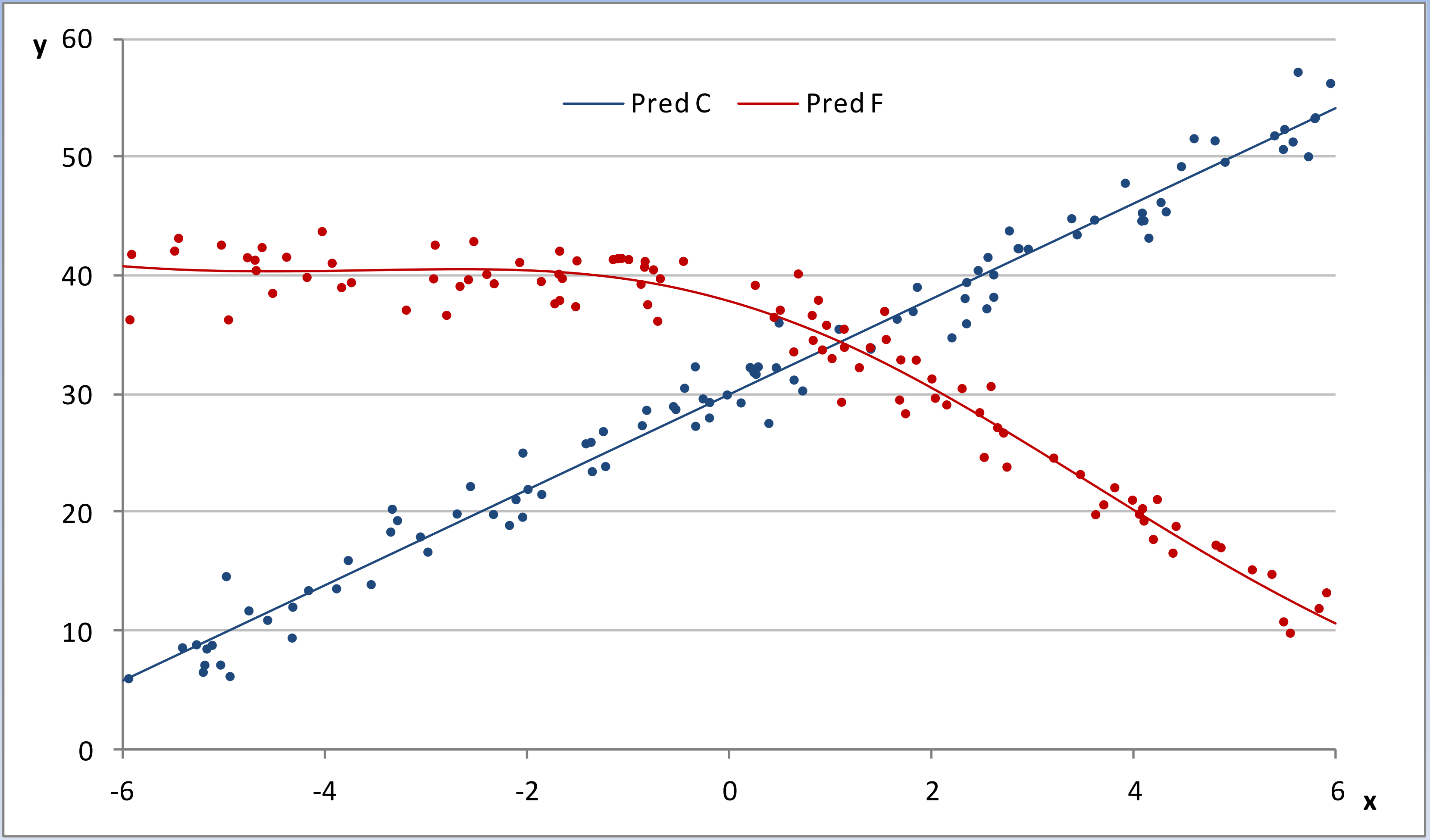


Table 1. Comparison of Features and Capabilities of SAS/STAT 9.1.3 Procedures for Penalized/Smoothing Spline Fitting

	SAS/STAT Procedure				
Capability	**GAM**	**TPSPLINE**	**TRANSREG**	**GLIMMIX**	**MIXED**
Additive smoothers allowed?	✓	✗	✓	✓	✓
Unpenalized spline do-able?	model / df= ;	model / df= ; model / lognl0 = -10;	pspline, bspline, smooth(/ sm=0), sspline(/ sm=0)	Put basis functions in model statement (instead of: random / type=rsmooth)	Put basis functions in models statement instead of random statement
Provides EDF?	✓	✓	✗*	*= n-df(res)*	✗
Variable coefficient model do-able?	Manually create new sets of x	✗	model y = class(group / zero=none) | smooth(x / sm=50 after);**	Random x / type=rsmooth group=groop ;	Manually code basis functions
GCV do-able?	model / method = gcv;	✓	✗***	NA	NA
Default λ	df=4	GCV	sm = 0 (for smooth and sspline)	Estimated via mixed model theory	Estimated via mixed model theory
How to output smooth predictions	Score dataset and statement	Score dataset and statement	Pad input data with additional x and missing y	Pad input data with additional x and missing y	Pad input data with additional x, their corresponding basis functions, and missing y
Built-in basis functions	TPF (cubic), radial (bivariate only)	radial	TPF (cubic), B-spline	Radial (knot based approx), B-spline*	None****
Knot Control	✗	NA	bspline or pspline(x / knots=low to high by incr). No knot control for smooth & sspline***	random x / type=rsmooth knotmethod=data(knotty) ;	Only via manually coded basis functions
Partial residuals obtainable?	✓	Only for special case of one univariate smoother in model	✓	Only for special case of one smoother in model	Only via manually coded basis functions

* Feature available in vers. 9.2 and 9.3

**For vers. 9.2 and 9.3, the syntax has changed slightly to: model y = class(group / zero=none) * smooth(x / sm=50);

***Since vers. 9.2 pbspline is available, knot number and placement is customizable, and λ can be determined based on either CV, GCV, AIC, AICc, or SBC

****In vers. 9.3, although the Effect statement cannot be used in proc mixed, it will work in proc hpmixed

Conclusion

- Splines are powerful tools for interpolation and curve fitting when no parametric form is needed
- TPF and radial basis functions are simple enough that getting predictive equation and other features (e.g. derivatives, peaks, etc.) is do-able
- Inference of nonlinear predictor variables is possible using GAMs (and GAMMs) built from smoothing splines
- GAMMs (proc mixed & glimmix) offer tremendous modelling flexibility
- Partial residuals are recommended to assess additive model adequacy and fit
- Newer SAS/STAT releases are adding features for fitting and testing smoothing splines, GAMs, and GAMMs. Hopefully this continues!

References

Cai, Weijie. 2008. Fitting Generalized Additive Models with the GAM Procedure in SAS 9.2. SAS Global Forum Paper 378-2008. 14 pp. SAS Institute Inc., Cary, NC. *http://www2.sas.com/proceedings/forum2008/378-2008.pdf*

Cohen, Robert C. 2009. Applications of the GLMSELECT Procedure for Megamodel Selection. SAS Global Forum Paper 259-2009. 255 pp. SAS Institute Inc., Cary, NC. *http://support.sas.com/resources/papers/proceedings09/259-2009.pdf*

Ngo, L and M.P. Wand. 2004. Smoothing with mixed model software . Journal of Statistical Software, 9 (1). *http://www.stat.ucl.ac.be/ISpersonnel/lambert/NgoWand2004JrStatSoft.pdf*

Simon Wood's webpage: *http://www.maths.bath.ac.uk/~sw283/*