The Stata Journal (2007)
7, Number 1, pp. 45–70

Multivariable modeling with cubic regression splines: A principled approach

Patrick Royston
UK Medical Research Council
London, UK
patrick.royston@ctu.mrc.ac.uk

Willi Sauerbrei
University Medical Center
Freiburg, Germany

Abstract. Spline functions provide a useful and flexible basis for modeling relationships with continuous predictors. However, to limit instability and provide sensible regression models in the multivariable setting, a principled approach to model selection and function estimation is important. Here the multivariable fractional polynomials approach to model building is transferred to regression splines. The essential features are specifying a maximum acceptable complexity for each continuous function and applying a closed-test approach to each continuous predictor to simplify the model where possible. Important adjuncts are an initial choice of scale for continuous predictors (linear or logarithmic), which often helps one to generate realistic, parsimonious final models; a goodness-of-fit test for a parametric function of a predictor; and a preliminary predictor transformation to improve robustness.

Keywords: st0120, mvrs, uvrs, splinegen, multivariable analysis, continuous predictor, regression spline, model building, goodness of fit, choice of scale

1 Introduction

Building reliable multivariable regression models is a major concern in many application areas, including the health sciences and econometrics. When one or more of the predictors is continuous, the question arises of how to represent the relationship with the outcome meaningfully in accordance with substantive background knowledge. For example, one would want the fitted curve for a variable thought to be monotonically related to the outcome to be monotonic. Furthermore, removing apparently uninfluential variables from a final model is often desirable. Although procedures to select variables create difficulties, sometimes inducing selection and omission bias of regression coefficients, selection of variables is often required (Miller 1990, Sauerbrei 1999). Furthermore, when several continuous variables are present, checking the linearity assumption is essential.

The Stata program `mfp` provides a framework for carrying out both tasks simultaneously: selecting (fractional polynomial) functions of continuous variables, and if desired, removing apparently uninfluential predictors from the model (Royston and Sauerbrei 2005). The key element of `mfp` is a structured approach to hypothesis testing, and within this, a systematic search for the best-fitting fractional polynomial (FP) function for each predictor, within a class of such functions of prespecified complexity. Remov-

ing uninfluential binary variables is done by conventional significance testing (backward elimination) with a predetermined nominal p-value, α. For continuous variables, using a closed-test approach (Marcus, Peritz, and Gabriel 1976) maintains the type I error probability for removing a given predictor at approximately α, irrespective of the fact that several tests for simplifying the chosen function may be carried out. Although the `mfp` approach does not (indeed, cannot) solve the problems of selection and omission bias, Royston and Sauerbrei (2003) showed by bootstrap resampling that it may nevertheless find stable multivariable models.

In brief, for those unfamiliar with FP models: an FP function of degree $m \geq 1$ is an extension of a conventional polynomial. If we write the latter for an argument x as

$$\mathrm{FP}m\,(x) = \beta_0 + \beta_1 x^{p_1} + \cdots + \beta_m x^{p_m} \tag{1}$$

where $p_1 = 1$, $p_2 = 2$, ..., $p_m = m$, then an FP function is obtained by generalizing the indices or powers $p_1, \ldots, p_m$ to certain fractional and nonpositive values such that each p_j for $j = 1, \ldots, m$ belongs to the set $S = \{-2, -1, -0.5, 0, 0.5, 1, 2, 3\}$ rather than to the set of integers $\{1, \ldots, m\}$. By convention $x^0 = \ln(x)$ here. Also, a special definition is applied for equal powers, according to a mathematical-limit argument. For example, a degree-2 FP (or FP2) function with equal powers (p, p) is defined as $\beta_0 + \beta_1 x^p + \beta_2 x^p \ln x$, and similar rules apply to the definition of higher-order functions. In practical data analysis, we find that FP2 functions are usually flexible enough.

For given m, the best-fitting powers are selected by maximizing the likelihood of model (1) over all combinations of powers in S. When conditioned on the powers, model (1) is linear in the transformed x's; maximizing the likelihood, therefore, is simply a matter of enumerating the models generated by all possible combinations of powers, fitting each in the conventional manner, and evaluating its likelihood.

Univariable model selection for FP functions follows a closed-test approach (Marcus, Peritz, and Gabriel 1976). First, the most complex acceptable FP function is chosen. In most applications, this will be an FP2 function. Starting, then, with the best-fitting FP2 model, the overall significance of x is tested by comparing the best FP2 model with the model from which x has been omitted. A likelihood-ratio χ^2 test on 4 degrees of freedom (df) is used. The p-value from this test is the evidence of whether x should enter the model. If the test is not significant, the procedure stops, x being declared nonsignificant. Otherwise, possible nonlinearity of the effect of x is tested by comparing the FP2 model with a straight line. If the test is not significant, the linear model is accepted; otherwise, the final test is performed, that of FP2 versus the best-fitting FP1. A significant result here indicates the need for a more complex (FP2) function; with a nonsignificant outcome, the simpler FP1 function is chosen.

We aim to provide for regression spline (RS) modeling a framework similar to `mfp` for FP modeling. Splines are good general-purpose smoothers. For example, they can model complex curve shapes beyond the scope of FP functions. For several types of application, splines provide a sensible approach. However, in multivariable modeling with simultaneous selection of variables and functional forms, no single approach is universally accepted. Difficulties occur even when modeling one predictor (Rosenberg et al.

2003). Here we adapt the closed-test approach that works well with fractional polynomials in `mfp` to selecting spline functions. We will describe and illustrate the resulting Stata ado-file `mvrs` (multivariable regression splines).

Our general philosophy of model building is based on experience in real applications, in simulation studies, and on investigations of model stability by bootstrap resampling (Sauerbrei and Schumacher 1992, Royston and Sauerbrei 2003). Guided by principles such as the need for interpretability, reproducibility, and transportability, we prefer a simple model unless the data indicate the need for greater complexity. When modeling the effect of a continuous predictor, the simplest available function is the line, and that is our default option in most modeling situations. We would use a more complex model only if enough evidence of nonlinearity existed within the data and/or (a different situation) if prior knowledge dictated such a model. By contrast, local (nonparametric) regression modeling typically starts and ends with a complex model. A distinctive feature of modeling with `mfp` is the availability of a rigorous procedure for selecting variables and functions. Here we replace FP functions in `mfp` with RS functions, resulting in a new procedure called `mvrs` that is similar in spirit and character to `mfp`. The `mvrs` procedure combines selection of simplified spline functions with backward elimination of weak predictors and is applicable without detailed expert knowledge.

A second, useful application of splines is in checking the goodness of fit of a function modeled by some other approach, such as a linear or FP function. Since the aim is different, no function selection is needed. An RS function of predefined complexity is estimated in addition to the function to be checked, and the improvement in fit (representing lack of fit of the original function) is assessed (Royston 2000b).

We introduce two other ado-files: `uvrs` and `splinegen`. `uvrs` (univariate regression splines) is called by `mvrs` but is also useful in its own right, just as `fracpoly` complements `mfp`. Similarly, `splinegen` is analogous to `fracgen` and creates RS basis functions for a given variable. We first briefly introduce the three programs and then turn to methodological matters, describing restricted cubic splines and the method of spline model selection. Then we consider the choice of scale for a predictor, a preliminary transformation of a continuous predictor to improve robustness, and the application of RS functions in assessing goodness of fit of a postulated function. Two different examples follow. The first, with one predictor, is the well-known but challenging motorcycle dataset. The second involves multivariable prognostic modeling in breast cancer. We give details of the three programs near the end of the article.

2 Description

The main program, `mvrs`, selects the RS model that best predicts the outcome variable from one or more independent variables, at least one of which should be continuous. `uvrs` selects the RS model that best predicts the outcome variable from one continuous predictor, adjusting linearly for other predictors, if specified. The programs use basis functions for regression splines, based on knots whose positions are determined automatically according to equally spaced centiles of the distribution of the continuous

predictor. User-specified knot positions may also be introduced. `mvrs` and `uvrs` work with different types of regression model, e.g., `regress`, `logit`, and `stcox`. `splinegen` creates new variables comprising the required basis functions.

We describe the syntax and options for the three programs in *Syntax and options*.

3 Natural cubic spline functions

We will give the expression for a cubic regression spline that is restricted to be linear beyond two boundary knots $k_{\min}$ and $k_{\max}$ (not necessarily placed at the extremes of the argument, x). An unrestricted cubic regression spline with $m+2$ knots at $k_{\min} < k_1 < \ldots < k_m < k_{\max}$ takes the form

$$\begin{aligned} s(x) &= \beta_{00} + \beta_{10}x + \beta_{20}x^2 + \beta_{30}x^3 \\ &\quad + \sum_{j=1}^{m} \beta_j (x-k_j)_+^3 + \beta_{k_{\min}} (x-k_{\min})_+^3 + \beta_{k_{\max}} (x-k_{\max})_+^3 \end{aligned}$$

where the *plus function* $(x-k)_+$ is defined as $x-k$ if $x \geq k$ and 0 otherwise. If $x < k_{\min}$, the linearity constraint implies that all quadratic and cubic terms must vanish, so that $\beta_{20} = \beta_{30} = 0$. If $x > k_{\max}$, the linearity constraint may be handled through the fact that the second and third derivatives of $s(x)$ must both be zero. After some algebra (see Royston and Parmar 2002, app. 2), the formula for the restricted cubic spline emerges as

$$\begin{aligned} s(x) &= \beta_{00} + \beta_{10}x + \sum_{j=1}^{m} \beta_j \left\{ (x-k_j)_+^3 - \lambda_j (x-k_{\min})_+^3 - (1-\lambda_j)(x-k_{\max})_+^3 \right\} \\ &= \gamma_0 + \gamma_1 x + \gamma_2 v_1(x) + \cdots + \gamma_{m+1} v_m(x) \end{aligned}$$

where $\gamma_0 = \beta_{00}$, $\gamma_1 = \beta_{10}$ and for $j = 1, \ldots, m$,

$$\begin{aligned} \gamma_{j+1} &= \beta_j \\ v_j(x) &= (x-k_j)_+^3 - \lambda_j (x-k_{\min})_+^3 - (1-\lambda_j)(x-k_{\max})_+^3 \end{aligned}$$

Thus the m knots give rise to a model with $m+1$ basis functions and hence $m+1$ df. In the Stata implementation (`mvrs`), the basis functions $x, v_1(x), \ldots, v_m(x)$ are orthogonalized to have mean zero and standard deviation 1 and to be uncorrelated. This configuration ensures that fitting complex spline functions with large m encounters no numerical instabilities. Untransformed, the basis functions are highly correlated.

4 Selecting an RS function for one predictor

To determine an RS model for a given continuous predictor x, the following approach is used. The algorithm, which is implemented in the `uvrs` program, embodies a closed-test procedure (Marcus, Peritz, and Gabriel 1976), a sequence of tests designed to maintain

the overall type I error probability at a prespecified nominal level, α, such as 0.05. The `alpha(#)` option of `uvrs` controls the value of α. The quantity α is the key determinant of the complexity (in dimension and therefore in shape) of a selected function.

Initially, the most complex permitted RS model is chosen, which is determined by the df assigned to the RS function. As explained above, it has $m+1$ df, where m is the maximum number of knots to be considered and $m = 0$ means the linear function. By default, `uvrs` takes $m = 3$ and df $= 4$ (see *Choice of defaults* below). You may select different df through the `df()` option.

Let us call the most complex model M_m and the linear function, M_0. First, model M_m is compared with the null model (omitting x), using a χ^2 test with $m+1$ df. If the test is not significant at the α level, the procedure stops and x is eliminated.

Otherwise, the algorithm next compares the fit of M_m with that of M_0, on m df. If the deviance difference is not significant at the α level, M_0 is chosen by default and the algorithm stops.

Now consider the m possible RS models using just one of the m available knots. The best fitting of these models, say, $\widetilde{M}_1$, is found and compared with M_m. If M_m does not fit significantly better than $\widetilde{M}_1$ at the α level, there is no evidence that the more complex model is needed, so model $\widetilde{M}_1$ is accepted and the algorithm stops. Otherwise, $\widetilde{M}_1$ is augmented with each of the remaining $m-1$ knots in turn; the best-fitting model, $\widetilde{M}_2$, is found; and $\widetilde{M}_2$ is compared with M_m. The procedure continues in this fashion until either a test is nonsignificant and the procedure stops or all the tests are significant, in which case model M_m is the final choice.

If x is to be forced into the model, the first comparison, between M_m and the null model, is omitted.

All tests are based on χ^2 statistics from deviance ($-2 \times$ log likelihood) differences.

4.1 Choice of defaults

The default type of spline in `uvrs` is cubic, invoked by `degree(3)`. We believe this to be by far the most commonly used spline. The default `df(4)` is a compromise between complexity and power. If, say, `df(20)` were the default, complex functions could be modeled. But with significance testing for model simplification, the first test, of M_{20} versus M_0, would be on 20 df and could have low power if the underlying function was simple (low dimensional).

5 Choice of scale for a predictor

When working with cubic smoothing splines (Green and Silverman 1994) in generalized additive models (Hastie and Tibshirani 1990), Royston (2000a) discussed whether fitting models for a continuous predictor x on the original or a logarithmic scale is better. He showed that using a preliminary logarithmic transformation in a spline model often

yielded a more accurate and more parsimonious fit. He suggested a simple way to decide a priori which scale was preferable. This approach amounts to fitting the model $\beta_0+\beta_1 x+\beta_2 \ln x$ to each predictor and using a log scale if $z_2 > z_1$ and the untransformed scale x otherwise, where $z_j = \left|\widehat{\beta}_j \Big/ \text{SE}\left(\widehat{\beta}_j\right)\right|$ for $j = 1, 2$ is the absolute t statistic for each regression coefficient. The log scale may be chosen instead if z_2 is convincingly larger than z_1. The choice between the linear and log scales may affect the type I error probability of the closed-test procedure for selecting an RS function.

The same considerations apply to models based on regression splines. The preliminary test of scale is easy to carry out and results in a choice of x or $\ln x$ for each predictor. Making this decision univariately for each continuous x is preferable. The alternative is to fit a full model including x and $\ln x$ terms for every continuous predictor, but since this model is likely to be large and unstable, we do not recommend the approach.

The main effect on selecting a final model with `mvrs` is that if $\ln x$ is used, the default function becomes logarithmic rather than linear. Spline functions must then compete with logarithmic functions. Suppose that the true curve shape is nonlinear and resembles approximately a log function. Using $\ln x$ may then mean that a log function is selected by default in preference to a spline function that is more complex and perhaps more accurate but only weakly supported by the data. On the other hand, when the true function is logarithmic, using the linear function of x as the default may result in an unsatisfactory spline approximation. An appropriate choice of scale may influence the selected model and may give more parsimonious curve shapes for the continuous predictors than always working on the linear scale. We will illustrate this issue in the breast cancer data (see example 2).

6 Robustification by preliminary predictor transformation

As discussed by Royston and Sauerbrei (2007), a few influential extreme predictor values can severely affect FP functions. For example, an FP2 model may be selected in which one of the two terms is driven entirely by one or a few observations. Such a type of overfitting is not desirable.

Royston and Sauerbrei (2007) proposed dealing with this possibility by applying a preliminary transformation $g(x)$ to x. The transformation is defined as

$$g\left\{\frac{x-\overline{x}}{s}\right\} = \left[\ln\left\{\frac{\Phi\left(\frac{x-\overline{x}}{s}\right)+\varepsilon}{1-\Phi\left(\frac{x-\overline{x}}{s}\right)+\varepsilon}\right\}+\varepsilon^*\right]\Big/\left(2\varepsilon^*\right)$$

where $\overline{x}$ and s are, respectively, the sample mean and standard deviation of observed values of x, $\varepsilon = 0.01$, $\varepsilon^* = \ln\{(1+\varepsilon)/\varepsilon\} = 4.615$, and $\Phi(\cdot)$ is the normal cumulative distribution function (`normal()` in Stata). The parameter value $\varepsilon = 0.01$ was chosen to linearize $g(\cdot)$ as much as possible in the broad central region of the predictor. As a result, $g(\cdot)$ is close to linear over most of the range of x and tapers smoothly to 0 or 1 for $x = \pm\infty$. Figure 1 shows the shape of $g(\cdot)$ for $\varepsilon = 0.01$.

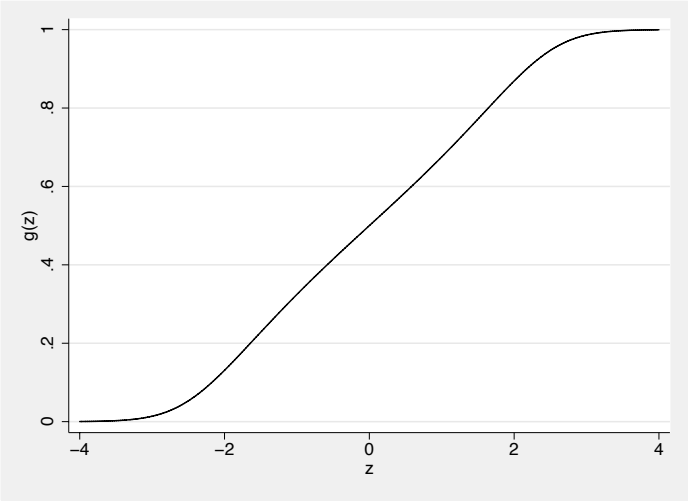

Figure 1: Robustification function $g(z)$ with $\varepsilon = 0.01$

For use with FP functions (which are sensitive to values of x near zero), a second, linear transformation is applied to $g(x)$ to shift the origin to a suitable positive value. Since RS functions are invariant to a change of origin, this step is not required here, but nevertheless $g(x)$ should help to cope with influential points in regression splines.

7 MVRS algorithm

The MVRS algorithm and `mvrs` program for selecting an RS model from a given set of predictors are motivated largely by Stata's `mfp` command for building multivariable FP models. `mfp` combines backward elimination, allowing reinclusion of omitted variables, with the selection of the functional form for continuous predictors. Each predictor is considered in turn, with the functions and inclusion/exclusion status of all other variables temporarily fixed. Variables are considered in decreasing order of statistical significance in a full linear model. The algorithm cycles over each predictor repeatedly in the same order, changing the model according to the results of the tests of individual variables—see *Selecting an RS function for one predictor*. The process stops when there is no further change in the variables included in the model and in the spline functions (knots) chosen for each continuous variable.

8 Applying splines to assess goodness of fit

Royston (2000b) described the use of cubic smoothing splines (Green and Silverman 1994) to assess how well FP functions of a given predictor, x, fit the data. We adapt the method for use with regression splines and will illustrate it for the breast cancer

data later. The idea is to fit a complex spline function for x and to test whether it improves the fit significantly compared with a parametric function of interest, such as an FP or even an RS function, fitted separately. If the smoothing spline function is not significant once the parametric function is included in the model, the conclusion (or at least the impression) is that the parametric function is adequate. Aside from questions of statistical significance, the estimated smoothing spline function may indicate where (if at all) in the range of x the fit of the parametric function is deficient.

Here we slightly modify Royston's (2000b) approach by replacing smoothing splines with regression splines. If the function to be checked for goodness of fit is also an RS function selected by `uvrs` or `mvrs`, our approach is just a test of the significance of several additional knots, together with a plot of discrepancy (see below). Such a check of the function is most relevant in a multivariable setting where the functional form for each continuous predictor should be relatively simple. For each such function, a check for serious mismodeling of some part of the function is desirable. In a univariable situation with a more complex selected RS function, the assessment is less relevant.

The procedure is as follows. First, the (maximum) number of df, ν, for the most complex RS model is chosen. Per Royston (2000b), we take $\nu = 15$. Second, starting with 15 df, a simplified RS model is found by applying `uvrs` with `df(15)` and using a rather liberal p-value, such as 0.20, to allow a still relatively complex RS function to be selected if appropriate. If other variables are considered in a multivariable model, these should be included in an adjustment model; if functions were previously selected for them, these same functions should be used without modification. Third, the parametric model is fitted and the index for x is calculated. By the index for x, we mean the partial predictor $\widehat{f}(x) = \beta_0 + \sum \widehat{\beta}_j f_j(x)$. (The more common meaning of index with a multivariable model is the prognostic index or estimated linear predictor, $x\widehat{\beta}$.) Finally, a model is fitted using the appropriate number of knots for x, adjusting for other variables as just mentioned, with the index for x offset from the linear predictor. (The index for x could be included instead as a predictor. This decision is largely a matter of taste; the results are likely to be similar.) The goodness of fit of the parametric function is assessed by testing the regression coefficients for all the spline terms jointly against zero. A significant result indicates lack of fit.

To get a more detailed view of the function, the model just described may be fitted with an RS function with ν df without simplification. With the index for x offset, the fitted RS function estimates the discrepancy between the data and the index for x. A plot of the discrepancy against x, with a pointwise confidence interval, may indicate possible hot spots of lack of fit.

A straightforward alternative to using or simplifying a complex function is to fix ν at a much lower value such as 5 df (Harrell 2001), test the spline terms for statistical significance as just described (but without removing any terms from the model), and plot the resulting fit. This approach may have the advantage of being less data driven than the simplification method while retaining considerable flexibility.

In a model for a continuous outcome (i.e., standard multiple regression), the complex spline function would be fitted to the partial residuals from the parametric function, that is, to $y - \widehat{f}(x)$.

9 Example 1: A difficult smoothing challenge

We use as a first example the famous motorcycle data (see Härdle 1990). For one predictor x, careful selection of the RS model (i.e., the knot positions) is important. The data are accelerometer readings taken through time from an experiment on the efficacy of motorcycle crash helmets. The x coordinate is the time in milliseconds after a simulated impact, and the outcome variable y is the acceleration (in g) of the head of the test dummy. Figure 2 shows the raw data.

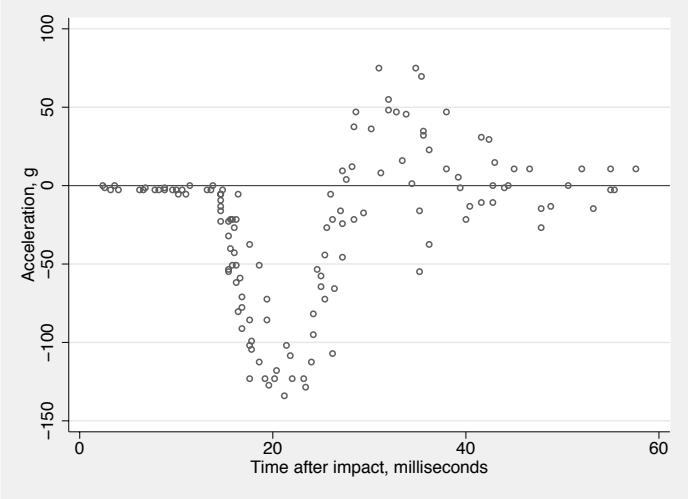

Figure 2: Motorcycle accelerometer data

No polynomial function can hope to reproduce such a pattern. To find a good spline fit, the df are increased from 1 (linear) to 15 in steps of 1, and we compute the Akaike information criterion (AIC) for each fit. The AIC is computed as the deviance (minus twice the maximized log likelihood) plus twice the df. Figure 3 shows how the AIC changes as a function of the df.

(*Continued on next page*)

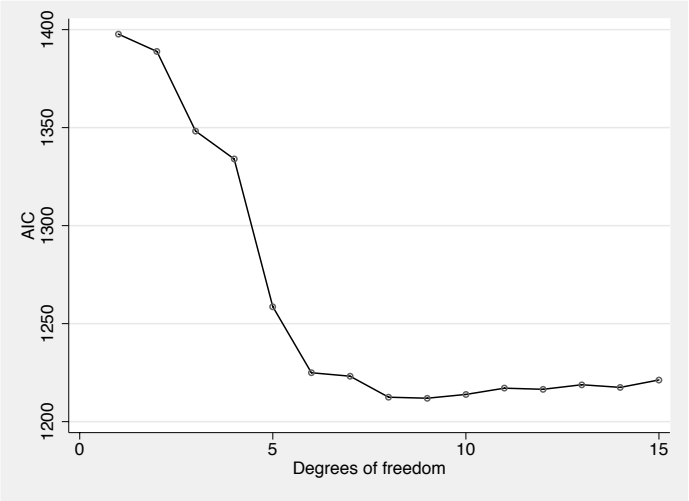

Figure 3: Motorcycle data: relationship between the AIC and the df of the spline smoother

The fit that minimizes the AIC is with 8 df. Figure 4 shows this fit, with a 95% pointwise confidence interval.

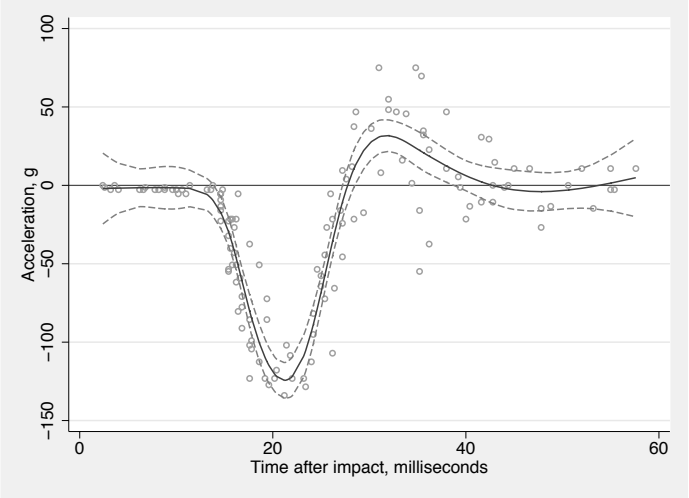

Figure 4: Fit of a regression spline with 8 df to the motorcycle data. Dashed lines, 95% pointwise confidence interval.

The fit appears excellent. Figure 5 shows the smoothed residuals from this fit.

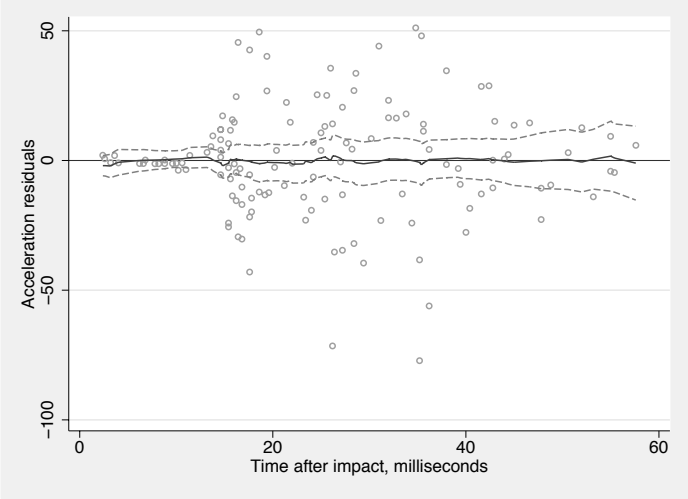

Figure 5: Smoothed residuals from the 8-df spline fit to the motorcycle data

The fit is formidably good. There is no sign of any departure of the smoothed residuals from the line $y = 0$.

Regarding preliminary transformation of x: the z values for x and $\ln x$ are 5.5 and 4.4, slightly favoring x. There are no outlying values of x, so the $g(x)$ transformation offers little scope for improving the fit.

10 Example 2: Prognostic model for breast cancer data

We use the data for 686 lymph node–positive breast cancer patients analyzed by Sauerbrei and Royston (1999). The data are provided in `brcancer.dta`. The full dataset is available at http://www.blackwellpublishers.co.uk/rss/. The outcome of interest is the recurrence-free survival time, that is, the duration in years from entry into the study (typically, the time of diagnosis of primary breast cancer) until either death or disease recurrence, whichever occurred first. There were 299 events for this outcome and the median follow-up time was about 5 years. The data in `brcancer.dta` have already been `stset`. Sauerbrei and Royston (1999) published a prognostic model that accounted for patient's age (`x1`), number of positive lymph nodes (`x5`, a predictor that is positively and strongly associated with a poor prognosis), tumor grade 2/3 (`x4a`), tumor progesterone receptor status (`x6`), and whether the patient received hormonal treatment (`hormon`). Sauerbrei and Royston (1999) forced `hormon`, a treatment variable, into the model; here we will regard it as a normal predictor. Other potential prognostic factors in the dataset include menopausal status (`x2`), tumor size (`x3`), grade 3 (`x4b`), and estrogen receptor status (`x7`).

We will build a model selecting variables and RS functions at the $\alpha = 0.05$ significance level. The chosen value of α is input using the key option `select()`; default settings are taken for all other options. We show the `mvrs` command and its output below.

```
. use brcancer
(German breast cancer data)
. mvrs stcox x1 x2 x3 x4a x4b x5 x6 x7 hormon, select(0.05)
Deviance for model with all terms untransformed =  3471.637, 686 observations
```

Variable	Final df	Deviance	Dev.diff cf. null	P	Final knot positions
x5	2	3442.467	61.143	0.000	3
x6	3	3434.121	29.855	0.000	7 132
hormon	1	3434.121	9.544	0.002	linear
x4a	1	3434.121	4.380	0.036	linear
x3	0	3435.181	1.060	0.180	
x2	0	3435.871	0.690	0.406	
x4b	0	3435.979	0.109	0.742	
x1	3	3415.328	20.743	0.000	46 53
x7	0	3416.546	1.217	0.135	
End of Cycle 1: deviance =		3416.546			
x5	2	3416.546	73.125	0.000	3
x6	3	3416.546	29.658	0.000	7 132
hormon	1	3416.546	12.122	0.000	linear
x4a	0	3420.320	3.774	0.052	
x3	0	3420.320	2.191	0.117	
x2	0	3420.320	0.332	0.565	
x4b	0	3420.320	0.067	0.796	
x1	3	3420.320	22.442	0.000	46 53
x7	0	3420.320	1.231	0.148	
End of Cycle 2: deviance =		3420.320			
x5	2	3420.320	75.424	0.000	3
x6	3	3420.320	36.964	0.000	7 132
hormon	1	3420.320	12.232	0.000	linear
x4a	0	3420.320	3.774	0.052	
x3	0	3420.320	2.191	0.117	
x2	0	3420.320	0.332	0.565	
x4b	0	3420.320	0.067	0.796	
x1	3	3420.320	22.442	0.000	46 53
x7	0	3420.320	1.231	0.148	

```
Regression spline fitting algorithm converged after 3 cycles.
Transformations of covariates:
```

```
Final multivariable spline model for _t
```

Variable	Initial df	Initial Select	Initial Alpha	Final Status	Final df	Final Knot positions
x1	4	0.0500	0.0500	in	3	[lin] 46 53
x2	1	0.0500	0.0500	out	0	
x3	4	0.0500	0.0500	out	0	
x4a	1	0.0500	0.0500	out	0	
x4b	1	0.0500	0.0500	out	0	
x5	4	0.0500	0.0500	in	2	[lin] 3
x6	4	0.0500	0.0500	in	3	[lin] 7 132
x7	4	0.0500	0.0500	out	0	
hormon	1	0.0500	0.0500	in	1	Linear

```
Cox regression -- Breslow method for ties
Entry time _t0                                    Number of obs   =        686
                                                  LR chi2(9)      =     156.03
                                                  Prob > chi2     =     0.0000
Log likelihood =   -1710.16                       Pseudo R2       =     0.0436
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
x1_0	.0176848	.0568584	0.31	0.756	-.0937556	.1291252
x1_1	-.1812154	.0556255	-3.26	0.001	-.2902394	-.0721913
x1_2	.2091295	.0608275	3.44	0.001	.0899098	.3283492
x5_0	.4201975	.0530724	7.92	0.000	.3161774	.5242175
x5_1	.2578767	.0596713	4.32	0.000	.1409232	.3748303
x6_0	-.3850654	.1960612	-1.96	0.050	-.7693384	-.0007925
x6_1	-.0593699	.1952547	-0.30	0.761	-.4420621	.3233222
x6_2	.1979548	.0706654	2.80	0.005	.0594532	.3364564
hormon	-.4394067	.1282634	-3.43	0.001	-.6907985	-.188015

```
Deviance: 3420.320.
```

The routine converged after three cycles. In the report for each cycle, `df` refers to the df of the model selected for each predictor, and `df` $= 0$ denotes a variable that was omitted from the model. Here we allowed at most 4 df for each continuous predictor (the default). For example, in the final cycle the df for `x5` and `x6` were 2 and 3, meaning that spline functions with 1 and 2 knots were selected for these variables, respectively. The knot positions are given as 3 and 7, 132. The reported `Deviance` is minus twice the log likelihood for the model selected at that stage of the procedure. `Dev. diff. cf. null` is the difference in deviance between the model including the variable in question and the null model excluding that variable. The corresponding p-value for a test of the null hypothesis is `P`. The deviance may change as variables are included and/or excluded, and different functions are chosen; this happened in cycles 1 and 2. In cycle 1, the effect of `x4a` was assessed in a model with linear effects for all variables coming after `x4a` in the ordering. At the end of cycle 1, two knots were selected for `x1`. The corresponding age function was used in all models considering the influence of `x5` to `x4b` in cycle 2. When the model was adjusted for the age function with two knots, `x4a` just failed to be significant at the 5% level. The final Cox model is summarized in the last two output tables. Variables `x1`, `x5`, `x6`, and `hormon` were selected, and `x2`, `x3`, `x4a`, `x4b`, and `x7` were dropped.

Figure 6 shows the estimated functions and pointwise 95% confidence intervals for each selected continuous predictor.

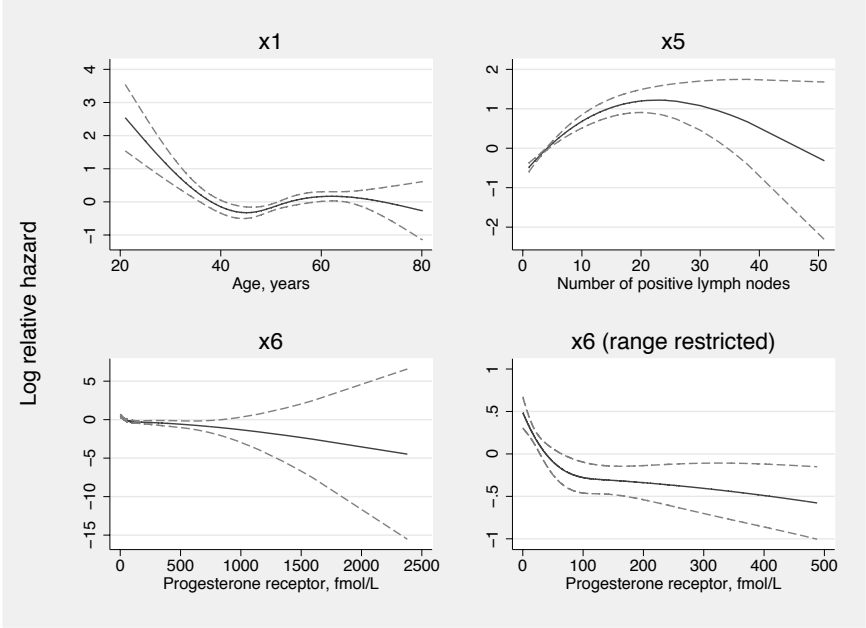

Figure 6: Estimated partial predictors (log relative hazards) for each continuous predictor in the multivariable RS model for the German breast cancer data. Dashed lines, 95% pointwise confidence intervals.

You may produce the values for such figures by using `fracpred` after `mvrs`; for example,

```
. fracpred fx5, for(x5)
. fracpred sfx5, for(x5) stdp
. gen llx5 = fx5 - 1.96*sfx5
. gen ulx5 = fx5 + 1.96*sfx5
```

The bottom right-hand panel shows more detail of the RS function for `x6` and shows that the risk descends rapidly up to about 100 fmol/L and then is roughly constant. In general, the functions are similar to those obtained by Sauerbrei and Royston (1999) using the `mfp` command, except that here, `x4a` is omitted. The spline function for `x5` is biologically implausible, since it is unlikely that the risk of breast cancer recurrence will be reduced for patients with many positive lymph nodes. Replacing `x5` with a preliminary negative exponential transformation `x5e` $= \exp(-0.12 \times$ `x5`$)$, as suggested by Sauerbrei and Royston (1999), gives a more sensible monotonically increasing curve with no loss of goodness of fit.

10.1 Choice of scale

Table 1 gives the results of the preliminary test of scale for each continuous predictor, i.e., for `x1`, `x3`, `x5`, `x6`, and `x7`.

Table 1: Choice of scale for each continuous predictor in the breast cancer dataset

Predictor	z_1	z_2
`x1`	3.72	3.91
`x3`	0.30	1.67
`x5`	0.23	4.46
`x6`	0.95	4.46
`x7`	1.42	4.19

NOTE: A larger value of z_2 indicates preference for a logarithmic scale.

As seen in table 1, a log scale is indicated for all predictors, strongly for `x5`, `x6`, and `x7`; weakly for `x3`; and very weakly for `x1`. On applying `mvrs` to select a model from the log-transformed predictors and the binary predictors, the same factors emerge as before (i.e., `x1`, `x4a`, `x5`, `x6`) but now only `x1` requires spline transformation (two knots are indicated). The final model is `x1` (spline), `x4a`, log `x5`, log (`x6` + 1), and `hormon`. The deviance ($-2 \times$ maximized log partial likelihood) for this model is 3,424.4, compared with 3,420.3 before. However, because the new model is more parsimonious, the values of the AIC are nearly identical: 3,438.4 versus 3,438.3 before.

Figure 7 compares the fitted curves for `x5` and `x6` from the two models.

(*Continued on next page*)

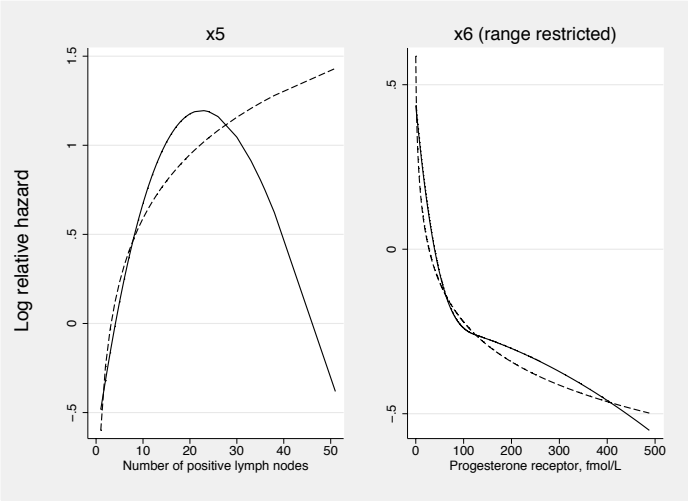

Figure 7: Fitted curves for `x5` and `x6` in the breast cancer data on the original and log-transformed scales. Solid lines, original scale; dashed lines, log scale.

The curves for `x6` are similar, whereas those for `x5` differ dramatically for `x5` > 10 nodes (12% of the sample). The log function is much more plausible than the quadratic-like spline function here and shows the benefit of paying attention to the initial choice of scale when building complex multivariable spline models. A major aim, and indeed the art, in such an exercise is to limit the complexity and hence the instability of the final model while retaining enough flexibility to represent realistic functional forms.

10.2 Preliminary robustness transformation

To illustrate the idea, and to analyze the `mvrs` procedure's sensitivity to possible influential observations, we subjected the continuous predictors `x1`, `x3`, `x5`, `x6`, and `x7` to the preliminary transformation $g(\cdot)$ before applying `mvrs` to the resulting data. Figure 8 shows the estimated functions and pointwise 95% confidence intervals for each selected continuous predictor.

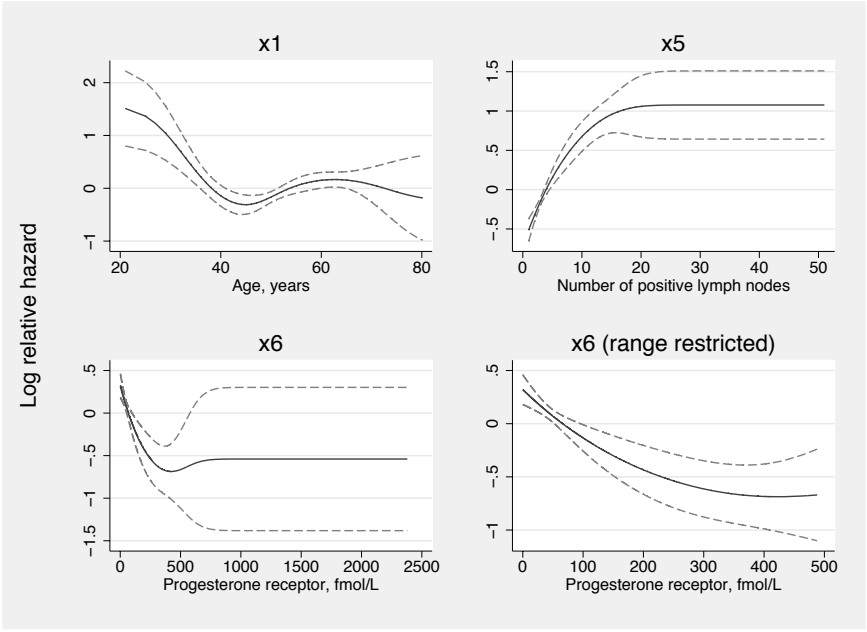

Figure 8: Estimated partial predictors (log relative hazards) and 95% pointwise confidence intervals for each continuous predictor in a multivariable RS model for the German breast cancer data. The continuous predictors were subjected to the transformation $g(x)$ before multivariable model building commenced.

The results are generally similar to those shown in figure 6. However, the function for `x5` is much improved. The function for `x6` has an implausible minimum near 500 fmol/L but is otherwise acceptable.

See *Discussion* in Royston and Sauerbrei (2007) for more comments on using the robustness transformation.

10.3 Goodness of fit

We illustrate the procedure discussed in *Application of splines to assess goodness of fit* for the predictor `x1` (patient's age) in the breast cancer data. We adjust for other variables in Sauerbrei and Royston's (1999) FP-based model III, namely, `x4a`, `x5e`, $\sqrt{\texttt{x6}+1}$, and `hormon`. The best-fitting FP for `x1` in this model has two terms with powers $(-2, -0.5)$.

The simplified RS model for `x1` derived starting with $\nu = 15$ has 5 df and four knots selected at 37, 56, 59, and 67 years. (See *Stata code for the goodness-of-fit test* for details of how this was done.) For testing goodness of fit of the FP function, the FP-based index for `x1`, that is, $\widehat{\beta}_1\texttt{x1}^{-2} + \widehat{\beta}_2\texttt{x1}^{-0.5}$, is offset in the model. The test of the

four spline basis functions and a linear function on 5 df has $X^2 = 8.5$ (Pr = 0.13). This result indicates no serious lack of fit of the FP model.

With the largest (unsimplified) spline model with 15 df, the test of fit of the FP function gives $X^2 = 24.4$ (Pr = 0.059). There is weak evidence of lack of fit. The top-left panel of figure 9 shows the discrepancy plot for the spline with $\nu = 15$.

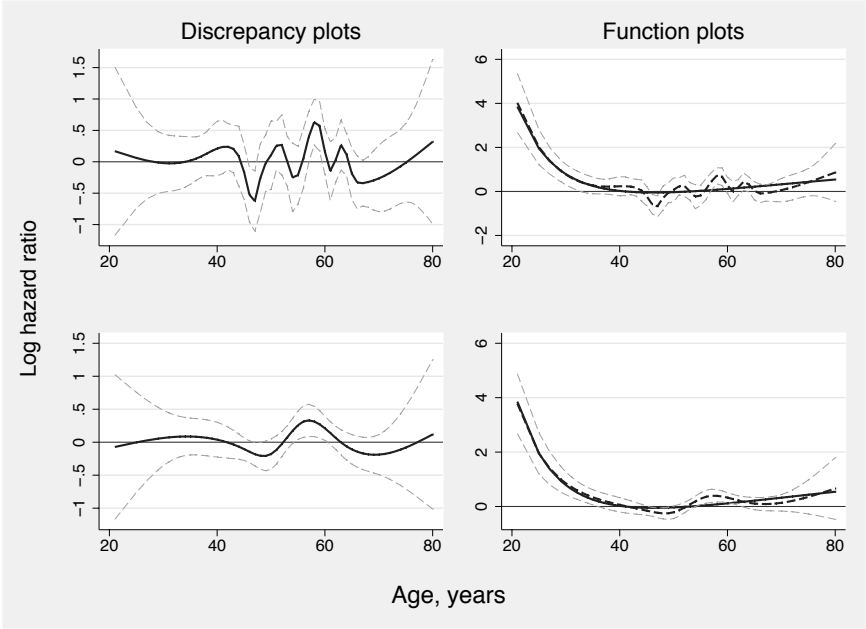

Figure 9: Goodness of fit for the FP2 function of `x1` in the breast cancer data. Top panels: results from a spline with 15 df and predetermined knot positions. Bottom panels: same as top panels, except that a spline with 5 df and predetermined knot positions were used. Left panels: discrepancy function with 95% pointwise confidence interval. Right panels: solid line, fitted FP2 function; dashed lines, spline fit plus the offset FP fit, with pointwise 95% confidence interval.

The discrepancy plot is simply a graph of the fitted spline function, accompanied by a 95% pointwise confidence interval. Since the FP function for `x1` has been offset and therefore removed from the linear predictor, the fitted spline (the discrepancy function) wiggles erratically around zero. The 95% pointwise confidence interval mostly includes zero. Importantly, there is no systematic pattern in the discrepancy function. In the corresponding function plot (figure 9, top-right panel), the FP function has been added back to the discrepancy function and to its confidence limits. The general conclusion from this analysis is that the FP2 function is an adequate fit.

Finally, we repeated the analysis with an unsimplified spline with 5 df, using predetermined knot positions of 45, 50, 56, and 63 years, again without simplification. The

graphical results are shown in the bottom two panels of figure 9. The discrepancy and function plots are less noisy than those in figure 9 and indicate that the FP function may not fit as well around 50–60 years. However, as before, the lack of fit is not statistically significant at the 5% level. The joint significance test of the spline terms has $X^2 = 7.8$ (Pr = 0.17).

Figure 9 nicely shows how unstable high-dimensional spline functions can be. Finding a convincing biological explanation of, for example, the top-right plot would be hard. The much simpler 5-df spline may be a bit easier to understand, since the perturbation in the fitted function detected by the spline occurs around the time of the menopause of the patients. The hormonal disturbances associated with that time of life could affect the prognosis. Nevertheless, the difference in estimated relative hazard between the spline and FP functions is small.

Stata code for the goodness-of-fit test

Stata code for performing the first version of the goodness-of-fit test for an FP2 function of `x1` in the breast cancer data and a spline with 15 initial df is as follows:

```
* FP transformation of x6
fracgen x6 0.5

* Get FP function for x1 with powers -2, -0.5,
* adjusting for other prognostic factors
fracpoly stcox x1 -2 -.5 x4a x5e x6_1 hormon
fracpred fpx1, for(x1)

* Store deviance of this model
local dev2 = e(fp_dev)

* Derive reduced spline model, store knots and df
uvrs stcox x1 x4a x5e x6_1 hormon, alpha(.05) df(15)
local knots `e(fp_k1)'
local dfreduced = e(fp_fdf)

* Test spline terms with FP index offset
uvrs stcox x1 x4a x5e x6_1 hormon, offset(fpx1) knots(`knots')
local dev1 = e(fp_dev)

di "Test of spline terms: chi2=" `dev2'-`dev1' " df = " `dfreduced' ///
 " P = " chi2tail(`dfreduced',`dev2'-`dev1')
```

To obtain the discrepancy function (`discrep`) and its 95% pointwise confidence interval (`ll`, `ul`), we proceed as follows:

```
uvrs stcox x1 x4a x5e x6_1 hormon, offset(fpx1) df(15)
fracpred discrep, for(x1)
fracpred sdiscrep, for(x1) stdp
generate ll = discrep - 1.96*sdiscrep
generate ul = discrep + 1.96*sdiscrep
```

11 Syntax and options

11.1 Syntax for mvrs

The ado-file `mvrs` is a regression-like command with the following syntax:

`mvrs` *regression_cmd* [*yvar*] *xvarlist* [*if*] [*in*] [*weight*] [`,` `all`
`alpha(`*alpha_list*`)` `cycles(#)` `degree(#)` `df(`*df_list*`)` `dfdefault(#)`
`knots(`*knot_list*`)` `select(`*select_list*`)` `xorder(+|-|n)` *regression_cmd_options*]

regression_cmd may be `clogit`, `glm`, `logistic`, `logit`, `ologit`, `oprobit`, `poisson`, `probit`, `qreg`, `regress`, `stcox`, `streg`, or `xtgee`. All weight types supported by *regression_cmd* are allowed.

11.2 Options for mvrs

`all` includes out-of-sample observations when generating the spline transformations of predictors. By default, the generated variables contain missing values outside the estimation sample.

`alpha(`*alpha_list*`)` sets the significance levels for testing between RS models of different complexity (numbers of knots). The rules for *alpha_list* are the same as for *df_list* in the `df()` option (see below). The default nominal p-value (significance level, selection level) is 0.05 for all variables. `alpha(0.01)` specifies that all variables have RS selection level 1%. `alpha(0.05, weight:0.1)` specifies that all variables except `weight` have RS selection level 5%; `weight` has level 10%.

`cycles(#)` is the maximum number of iteration cycles permitted. The default is `cycles(5)`.

`degree(#)` determines the type of spline transformation to be used. Valid values of # are 0, 1, and 3. The value of 0 denotes a step function, whereas 1 and 3 denote linear and cubic splines, respectively. The cubic splines are natural (sometimes called restricted); that is, the curves are restricted to be linear beyond the observed range of the predictor in question (or more precisely, beyond the boundary knots—see the `bknots()` option of `splinegen`). The default is `degree(3)`, meaning a natural cubic spline.

`df(`*df_list*`)` sets up the df for each predictor. For splines of degree 1 and 3, the df (not counting the regression constant) are equal to the number of knots plus 1. For splines of degree 0 (i.e., step functions), the df are equal to the number of knots. Specifying `df(1)` forces linear functions (no knots) for splines of degree 1 or 3 but forces dichotomization at the median for splines of degree 0. The first item in *df_list* may be either # or *varlist*`:`#. Later items must be *varlist*`:`#. Items are separated by commas and *varlist* is specified in the usual way for variables. With the first type of item, the df for all predictors are taken to be #. With the second type of item,

all members of *varlist* (which must be a subset of *xvarlist*) have # df. The default df for a predictor of type *varlist* specified in *xvarlist* but not in *df_list* are assigned according to the number of distinct (unique) values of the predictor, as shown in table 2:

Table 2: Rules for assigning df to continuous predictors

No. of distinct values	Default df
≤1	(Invalid predictor)
2–3	1
4–5	min(2, `dfdefault()`)
≥6	`dfdefault()`

`df(4)` means that all variables have 4 df. `df(2, weight displ:4)` means that `weight` and `displ` have 4 df; all other variables have 2 df. `df(weight displ:4, mpg:2)` means that `weight` and `displ` have 4 df, `mpg` has 2 df, and all other variables have the default of 1 df. `df(weight displ:4, 2)` is an invalid combination: the final 2 would override the earlier 4.

`dfdefault(#)` determines the default maximum df for a predictor. The default is `dfdefault(4)` (three knots for degree 1 or 3, four knots for degree 0).

`knots(`*knot_list*`)` sets knots for predictors individually. The syntax of *knot_list* is the same as for *df_list* in the `df()` option. By default, knots are placed at equally spaced centiles of the distribution of the predictor x in question. For example, by default three knots are placed at the 25th, 50th, and 75th centiles of any continuous x. The `knots()` option can be used to override this choice. `knots(1 3 5)` means that all variables have knots at 1, 3, and 5 (unlikely to be sensible). `knots(x5:1 3 5)` means that all variables except `x5` have default knots; `x5` has knots at 1, 3, and 5.

`select(`*select_list*`)` sets the nominal p-values (significance levels) for variable selection by backward elimination. A variable is dropped if its removal causes a nonsignificant increase in deviance. The rules for *select_list* are the same as for *df_list* in the `df()` option (see above). Using the default `select(1)` for all variables forces them all into the model. The nominal p-value for elements `(`*varlist*`)` of *xvarlist* is specified by including `(`*varlist*`)` in *select_list*. See also the `alpha()` option. The nominal p-value for elements *varlist* of *xvarlist* bound by parentheses is specified by including `(`*varlist*`)` in *select_list*.

`select(0.05)` means that all variables have nominal p-value 5%. `select(0.05, weight:1)` means that all variables except `weight` have nominal p-value 5%; `weight` is forced into the model.

`xorder(+|-|n)` determines the order of entry of the predictors into the model selection algorithm. The default is `xorder(+)`, which enters them in decreasing order

of significance in a multiple linear regression. `xorder(-)` places them in reverse significance order, whereas `xorder(n)` respects the original order in *xvarlist*.

regression_cmd_options may be any of the options appropriate to *regression_cmd*.

11.3 Syntax for uvrs

The ado-file `uvrs` is a regression-like command with the following syntax:

`uvrs` *regression_cmd* [*yvar*] *xvar covars* [*if*] [*in*] [*weight*] [`, all alpha(#) degree(#) df(#) knots(`*knot_list*`) report(`*numlist*`) trace` *regression_cmd_options*]

regression_cmd may be `clogit`, `glm`, `logistic`, `logit`, `ologit`, `oprobit`, `poisson`, `probit`, `qreg`, `regress`, `stcox`, `streg`, or `xtgee`. All weight types supported by *regression_cmd* are allowed.

`uvrs` (and hence also `mvrs`) automatically orthogonalizes the spline basis functions when `degree(3)` is used (i.e., cubic splines, the default). To do this, it applies the `orthog` option to calls to `splinegen`.

11.4 Options for uvrs

`all` includes out-of-sample observations when generating the spline transformations of *xvar*. By default, the generated variables contain missing values outside the estimation sample.

`alpha(#)` determines the nominal p-value used for testing the statistical significance of basis functions representing knots in the spline model. The default is `alpha(1)`, meaning to fit the full spline model without simplification.

`degree(#)` specifies the degree of spline. Allowed choices for # are 0 (meaning a step function), 1 (meaning a linear spline), and 3 (meaning a cubic spline). The default is `degree(3)`.

`df(#)` determines how many spline terms in *xvar* are used initially. The default is `df(4)`, which corresponds to three interior knots.

`knots(`*knot_list*`)` specifies knots in *knot_list* and overrides the default knots implied by `df()`.

`report(`*numlist*`)` reports effect sizes (differences on the spline curve) at values of *numlist*.

`trace` provides details of all models fitted and the progress of the knot selection algorithm.

regression_cmd_options are options appropriate to the regression command in use. For example, for `stcox`, *regression_cmd_options* may include `efron` or some other method for handling tied failures.

11.5 Syntax for splinegen

`splinegen` generates spline basis variables. The syntax is as follows:

`splinegen` *varname* [# [# ...]] [*if*] [*in*] [, `basis(`*stubname*`)` `bknots(##)` `degree(#)` `df(#)` `kfig(#)` `orthog`]

The # [# ...] are knot positions in the distribution of *varname*. If knots are specified, one cannot use the `df()` option.

11.6 Options for splinegen

`basis(`*stubname*`)` defines *stubname* as the first characters of the names of the new variables holding the basis functions. The default *stubname* is *varname*. The new variables are called *stubname*`_1`, *stubname*`_2`, ...

`bknots(# #)` define boundary knots for the spline. The spline function will be linear beyond the boundary knots. The default values of # # are the minimum and maximum values of *varname*.

`degree(#)` is the degree of spline basis functions desired. Possible values of # are 0, 1, and 3. Quadratic splines or splines higher than cubic are not supported at this time. The default is `degree(3)`, meaning cubic spline.

`df(#)` sets the desired df of the spline basis. The number of knots required is one less than the df for linear and cubic splines and equal to the df for zero-order splines (i.e., a step-function or dummy-variable basis). Knots are placed at equally spaced centiles of the distribution of *varname*; e.g., for linear or cubic splines with `df(4)`, knots are placed at the 25th, 50th, and 75th centiles of the distribution of *varname*. For degrees 1 and 3, default # is determined from the formula `int`$(n^{0.25}) - 1$, where n is the sample size; for degree 0, # is `int`$(n^{0.25})$.

`kfig(#)` determines the amount of rounding applied to the knots determined automatically from the distribution of *varname*. The default is `kfig(6)`, meaning that four significant figures are preserved. This option is seldom specified.

`orthog` creates orthogonalized basis functions. All basis functions higher than the first (linear) function are uncorrelated and have mean 0 and standard deviation 1. The linear function is also uncorrelated with the higher-basis functions.

12 Discussion

We have shown how our approach to multivariable model selection and function estimation for continuous predictors embodied in the `mfp` command may be transferred, in spirit and in practice, to multivariable modeling with regression splines. The function estimates that the new command `mvrs` produces are often similar to those from `mfp`. However, because splines are potentially more flexible than FP functions, the scope of `mvrs` is generally wider than that of `mfp`. Therefore, `mvrs` may be suitable for modeling complex functions such as may occur, for example, in mortality in city populations in response to variations in air pollution and climatic conditions over space and time.

Flexibility, although desirable, comes at a price—instability and a potential for overfitting. In medicine, dose–response functions are often simple and are well approximated by a linear or a logarithmic function. Trying to approximate a log function by a spline is likely to result in mismodeling; a clearly inappropriate function may be selected (see figure 6). In such a situation, a one-term FP function (i.e., a power or logarithmic transformation of x) may give a more sensible and satisfactory estimate than a spline. That is partly why we regard the preliminary choice of scale for a spline to be worthwhile. It certainly improves the quality of the estimated function for `x5` in the breast cancer dataset (see the dashed line in figure 7).

If complex functions are expected in a given dataset, the `mvrs` option `df()` may be changed accordingly. However, because of how the closed-test procedure operates to protect the overall type I error probability, the power to detect a complex function may be reduced. In such situations, therefore, using a compromise such as `df(8)` rather than `df(12)` or more may be sensible. When specific subject matter knowledge is available, the knots for a spline may instead be placed manually at values motivated by such knowledge. For example, it may be known that conditions changed at particular value(s) of x, leading to changes in the outcome being anticipated.

The significance level (option `alpha()`) is the second important input required for `mvrs` to select a more or less complex function and a more or less complex (i.e., large) multivariable model. As `mfp` also does, `mvrs` supports using different significance levels for the function selection (option `alpha()`) and variable selection (option `select()`) tasks. We discussed this point in more detail in modeling epidemiological data with `mfp` (Royston, Ambler, and Sauerbrei 1999).

In conclusion, we believe that `mvrs` will provide a useful tool to supplement `mfp` and Stata's other techniques for multivariable modeling. The command enables flexible modeling of continuous predictors while allowing the user some control over the excessive instability and tendency of spline functions to generate artifactual and uninterpretable features of a curve. In future work we will investigate with simulation studies of `mvrs` properties, as well as compare `mvrs` to `mfp`.

13 References

Green, P. J., and B. W. Silverman. 1994. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. London: Chapman & Hall.

Härdle, W. 1990. *Applied Nonparametric Regression*. Cambridge: Cambridge University Press.

Harrell, F. E., Jr. 2001. *Regression Modeling Strategies, With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York: Springer.

Hastie, T. J., and R. J. Tibshirani. 1990. *Generalized Additive Models*. London: Chapman & Hall.

Marcus, R., E. Peritz, and K. R. Gabriel. 1976. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika* 63: 655–660.

Miller, A. J. 1990. *Subset Selection in Regression*. New York: Chapman & Hall.

Rosenberg, P. S., H. Katki, C. A. Swanson, L. M. Brown, S. Wacholder, and R. N. Hoover. 2003. Quantifying epidemiologic risk factors using nonparametric regression: Model selection remains the greatest challenge. *Statistics in Medicine* 22: 3369–3381.

Royston, P. 2000a. Choice of scale for cubic smoothing spline models in medical applications. *Statistics in Medicine* 19: 1191–1205.

———. 2000b. A strategy for modeling the effect of a continuous covariate in medicine and epidemiology. *Statistics in Medicine* 19: 1831–1847.

Royston, P., G. Ambler, and W. Sauerbrei. 1999. The use of fractional polynomials to model continuous risk variables in epidemiology. *International Journal of Epidemiology* 28: 964–974.

Royston, P., and M. K. B. Parmar. 2002. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Statistics in Medicine* 21: 2175–2197.

Royston, P., and W. Sauerbrei. 2003. Stability of multivariable fractional polynomial models with selection of variables and transformations: A bootstrap investigation. *Statistics in Medicine* 22: 639–659.

———. 2005. Building multivariable regression models with continuous covariates in clinical epidemiology with an emphasis on fractional polynomials. *Methods of Information in Medicine* 44: 561–571.

———. 2007. Improving the robustness of fractional polynomial models by preliminary covariate transformation: A pragmatic approach. *Computational Statistics and Data Analysis*. Forthcoming.

Sauerbrei, W. 1999. The use of resampling methods to simplify regression models in medical statistics. *Applied Statistics* 48: 313–329.

Sauerbrei, W., and P. Royston. 1999. Building multivariable prognostic and diagnostic models: Transformation of the predictors using fractional polynomials. *Journal of the Royal Statistical Society, Series A* 162: 71–94.

Sauerbrei, W., and M. Schumacher. 1992. A bootstrap resampling procedure for model building: Application to the Cox regression model. *Statistics in Medicine* 11: 2093–2109.

About the authors

Patrick Royston is a medical statistician with nearly 30 years' experience, with a strong interest in biostatistical methodology and in statistical computing and algorithms. He now works in cancer clinical trials and related research issues. Currently, he is focusing on problems of model building and validation with survival data, including prognostic factor studies; on parametric modeling of survival data; on multiple imputation of missing values; and on new trial designs.

Willi Sauerbrei has worked for more than two decades as an academic biostatistician. He has extensive experience of randomized trials in cancer, with a particular concern for breast cancer. Having a long-standing interest in modeling prognosis and a Ph.D. thesis in issues in model building, he has more recently concentrated on model uncertainty, meta-analysis, treatment-covariate interactions, and time-varying effects in survival analysis.

Royston and Sauerbrei have collaborated on regression methods using continuous predictors for more than a decade. They are currently writing a book on multivariable modeling.