The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*, *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$ 98	1-year subscription	$138
2-year subscription	$165	2-year subscription	$245
3-year subscription	$225	3-year subscription	$345
1-year student subscription	$ 75	1-year student subscription	$ 99
1-year university library subscription	$125	1-year university library subscription	$165
2-year university library subscription	$215	2-year university library subscription	$295
3-year university library subscription	$315	3-year university library subscription	$435
1-year institutional subscription	$245	1-year institutional subscription	$285
2-year institutional subscription	$445	2-year institutional subscription	$525
3-year institutional subscription	$645	3-year institutional subscription	$765
Electronic only		**Electronic only**	
1-year subscription	$ 75	1-year subscription	$ 75
2-year subscription	$125	2-year subscription	$125
3-year subscription	$165	3-year subscription	$165
1-year student subscription	$ 45	1-year student subscription	$ 45

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2013)
13, Number 1, pp. 165–184

Joint modeling of longitudinal and survival data

Michael J. Crowther
Department of Health Sciences
University of Leicester
Leicester, UK
michael.crowther@le.ac.uk

Keith R. Abrams
Department of Health Sciences
University of Leicester
Leicester, UK

Paul C. Lambert
Department of Health Sciences
University of Leicester
Leicester, UK
and
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
Stockholm, Sweden

Abstract. The joint modeling of longitudinal and survival data has received remarkable attention in the methodological literature over the past decade; however, the availability of software to implement the methods lags behind. The most common form of joint model assumes that the association between the survival and the longitudinal processes is underlined by shared random effects. As a result, computationally intensive numerical integration techniques such as adaptive Gauss–Hermite quadrature are required to evaluate the likelihood. We describe a new user-written command, `stjm`, that allows the user to jointly model a continuous longitudinal response and the time to an event of interest. We assume a linear mixed-effects model for the longitudinal submodel, allowing flexibility through the use of fixed or random fractional polynomials of time. Four choices are available for the survival submodel: the exponential, Weibull or Gompertz proportional hazard models, and the flexible parametric model (`stpm2`). Flexible parametric models are fit on the log cumulative-hazard scale, which has direct computational benefits because it avoids the use of numerical integration to evaluate the cumulative hazard. We describe the features of `stjm` through application to a dataset investigating the effect of serum bilirubin level on time to death from any cause in 312 patients with primary biliary cirrhosis.

Keywords: st0289, stjm, stjmgraph, stjm postestimation, joint modeling, mixed effects, survival analysis, longitudinal data, adaptive Gauss–Hermite quadrature

1 Introduction

A joint model of longitudinal and time-to-event data can effectively assess the impact that a longitudinal covariate, measured with error, has on the time to an event of interest, providing a framework to assess the predictive ability of a biomarker on survival. Wulfsohn and Tsiatis (1997) and Henderson, Diggle, and Dobson (2000) have

shown that by undertaking a joint model that evaluates both the longitudinal and the survival data simultaneously, we can reduce biases and improve precision over simpler approaches. Such approaches include the separate modeling of each form of data by using standard tools such as `xtmixed` and `streg` or a two-stage approach whereby fitted values, including empirical Bayes estimates of the longitudinal model, are used as a time-varying covariate in a survival model. Conversely, joint models can also be viewed from the perspective of adjusting for informative drop-out in a longitudinal study (for example, if one finds when modeling quality of life over time in patients with cancer that patients with lower quality of life are more likely to die, resulting in nonignorable drop-out, as described in Billingham and Abrams [2002]).

The most widely used form of joint model assumes that the longitudinal and survival processes are underpinned by shared random effects. This results in a joint likelihood that cannot be evaluated analytically. Consequently, computationally demanding numerical integration techniques such as adaptive Gauss–Hermite quadrature (see Pinheiro and Bates [1995]) must be used to evaluate both the cumulative hazard and the overall joint likelihood.

The implementation of joint modeling in Stata is somewhat limited. The extensive `gllamm` suite (see Rabe-Hesketh, Skrondal, and Pickles [2002]) can fit shared parameter models but can assume only a piecewise exponential form for the survival submodel. The newly implemented `jmre1` command (see Pantazis and Touloumi [2010]) approaches analyses from the point of view of adjusting for informative drop-out in a longitudinal study, assuming the longitudinal and survival components are multivariate normal.

We present the `stjm` command, which allows the user to jointly model a continuous longitudinal response and the time to an event of interest. We assume a linear mixed-effects model for the longitudinal submodel, allowing flexibility through the use of fixed or random fractional polynomials of time. Four choices are available for the survival submodel, including the exponential, Weibull (Guo and Carlin 2004), and Gompertz proportional hazards models. We believe this is the first implementation of the Gompertz survival model within a joint modeling context. Furthermore, we implement the joint model of Crowther, Abrams, and Lambert (2012), which incorporates the flexible parametric survival model, `stpm2` (see Royston and Parmar [2002] and Lambert and Royston [2009]). Flexible parametric survival models are fit on the log cumulative-hazard scale, which has direct computational benefits because it avoids the need for numerical integration to evaluate the cumulative hazard. The models are fit by using maximum likelihood, with both simple and adaptive Gauss–Hermite quadrature available.

We illustrate the command by using a dataset of 312 patients with primary biliary cirrhosis (see Murtaugh et al. [1994] for further details). Of the 312, 158 were randomized to receive D-penicillamine, and 154 assigned a placebo. Serum bilirubin was measured repeatedly at intermittent time points. We investigate the effect of treatment after adjusting for the relationship between serum bilirubin levels and time to death. There may be other areas of application; however, in this article, we concentrate on the biostatistical aspect.

2 Joint modeling of longitudinal and survival data

Consider a clinical trial where we observe a continuous longitudinal biomarker, measured intermittently and with error, and the time to an event of interest. Baseline covariates are also recorded. Let S_i be the survival time of the ith patient, where $i = 1, \ldots, n$, and $T_i = \min(S_i, C_i)$ the observed survival time, with C_i the censoring time. Define an event indicator d_i, which takes the value of 1 if $S_i \leq C_i$ and 0 otherwise. Let $y_{ij} = \{y_i(t_{ij}), j = 1, \ldots, m_i\}$ denote the longitudinal response measurements of the continuous biomarker for the ith patient taken at times t_{ij}. Furthermore, we define shared random effects, $\boldsymbol{b}_i$, which underpin the survival and longitudinal processes. Each submodel can be dependent on a set of baseline covariates, U_i, which can potentially differ between submodels. We impose the common assumptions that both censoring and time of measurements are noninformative.

2.1 Longitudinal submodel

We specify for the longitudinal submodel a linear mixed-effects model where time can be modeled by using a combination of fixed or random fractional polynomials. This should provide a highly flexible framework to capture a variety of longitudinal trajectories (see Royston and Altman [1994]). Therefore, we observe

$$\begin{aligned} y_i(t_{ij}) &= W_i(t_{ij}) + e_{ij}, \qquad e_{ij} \sim \mathrm{N}(0, \sigma_e^2) \\ W_i(t_{ij}) &= x_i'(t_{ij})\beta + z_i'(t_{ij})b_i + u_i\delta \end{aligned} \tag{1}$$

with design matrices X_i and Z_i for the fixed (β) and random (b_i) effects, respectively, consisting of fractional polynomial time variables. Furthermore, we also have a vector of covariates (possibly time dependent), $u_i \in U_i$, and corresponding regression coefficients, δ. We assume that the measurement error, e_{ij}, is independent from the random effects and that $\mathrm{cov}(e_{ij}, e_{ik}) = 0$ (where $j \neq k$). $W_i(t_{ij})$ now represents the "true" underlying biomarker trajectory.

2.2 Survival submodel

Exponential, Weibull, and Gompertz

Standard parametric distributions have been implemented for the survival submodel. We define the proportional hazards submodel

$$h(t|\boldsymbol{b}_i, v_i) = h_0(t) \exp\{\alpha W_i(t_{ij}) + v_i\phi\}$$

where $h_0(t)$ is the baseline hazard function (see [ST] **streg** for more details), α denotes the association parameter, and ϕ is a set of regression coefficients associated with a set of covariates (again possibly time dependent), $v_i \in U_i$. In this formulation, we assume the association is based on the current value of the longitudinal response. In other words, the value of the biomarker, as estimated by the longitudinal submodel, is included in the survival linear predictor as a time-varying covariate.

If covariates are included in both submodels, then we can obtain overall effects on survival through combining the direct effect on the longitudinal marker, multiplied by the association parameter, plus the direct effect on survival. This concept is explained further in the example below and in Ibrahim, Chu, and Chen (2010).

Flexible parametric model

We define the proportional cumulative hazards time-to-event submodel

$$\log\{H(t|\boldsymbol{b}_i, v_i)\} = \log\{H_0(t)\} + \alpha W_i(t_{ij}) + v_i\phi \tag{2}$$

where $H_0(t)$ is the cumulative baseline hazard function. The remaining parameters are as defined in "Exponential, Weibull, and Gompertz".

The spline basis for this specification is derived from the log cumulative-hazard function of a Weibull proportional hazards model. The linear relationship between the baseline log cumulative-hazard and log time is extended by using restricted cubic splines, which impose the restriction that the fitted function be linear before the first knot and after the final knot. Further details can be found in Durrleman and Simon (1989), Royston and Parmar (2002), and Lambert and Royston (2009). We can therefore write a restricted cubic spline function of $\log(t)$, with knots $\mathbf{k}_0$, as $s\{\log(t)|\gamma, \mathbf{k}_0\}$. This is now substituted for the log cumulative baseline hazard in (2).

$$\log\{H(t|\boldsymbol{b}_i, u_{bs,i})\} = \eta_i = s\{\log(t)|\gamma, \mathbf{k}_0\} + \alpha W_i(t_{ij}) + v_i\phi$$

Transforming to the hazard and survival scales, we obtain

$$h(t|\boldsymbol{b}_i, v_i) = \left[\frac{1}{t}\frac{\mathrm{d}s\{\log(t)|\gamma, \mathbf{k}_0\}}{\mathrm{d}\log(t)} + \alpha\frac{\mathrm{d}W(t)}{\mathrm{d}t}\right]\exp(\eta_i), \qquad S(t|\boldsymbol{b}_i, v_i) = \exp\{-\exp(\eta_i)\}$$

Again this formulation is specific to the current value parameterization. We discuss the various forms of association in section 2.4.

2.3 Joint likelihood

Constructing the full likelihood for the joint model, we obtain

$$\prod_{i=1}^{n}\left(\int_{-\infty}^{\infty}\left[\prod_{j=1}^{m_i} f\left\{y_i(t_{ij})|b_i,\theta\right\}\right] f(b_i|\theta)f(T_i,d_i|b_i,\theta)\ \mathrm{d}b_i\right) \tag{3}$$

where

$$f\left\{y_i(t_{ij})|b_i,\theta\right\} = (2\pi\sigma_e^2)^{-1/2}\exp\left\{-\frac{y_i(t_{ij})-W_i(t_{ij})}{2\sigma_e^2}\right\}$$

and

$$f(b_i|\theta) = (2\pi|V|)^{-1/2}\exp\left(-\frac{b_i'V^{-1}b_i}{2}\right)$$

The survival likelihood component under an exponential, Weibull, or Gompertz submodel can be expressed as

$$f(T_i,d_i|b_i,\theta) = \left[h_0(T_i)\exp\left\{\alpha W_i(t)+\phi v_i\right\}\right]^{d_i}\exp\left[-\int_0^{T_i} h_0(u)\exp\left\{\alpha W_i(u)+\phi v_i\right\}du\right]$$

Under the flexible parametric modeling approach, the survival likelihood component is written as

$$f(T_i,d_i|b_i,\theta) = \left(\left[\frac{1}{T_i}\frac{\mathrm{d}s\{\log(T_i)|\gamma,\mathbf{k}_0\}}{\mathrm{d}\log(T_i)}+\alpha\frac{\mathrm{d}W(T_i)}{\mathrm{d}T_i}\right]\exp(\eta_i)\right)^{d_i}\exp\left\{-\exp(\eta_i)\right\}$$

Evaluating (3) is a computationally demanding task, the details of which are discussed in section 2.5.

2.4 Association structure

There are a variety of ways to link the longitudinal and survival components by using the trajectory function defined in (1). The most commonly used form, called the current value parameterization (described above), includes the trajectory function as a time-dependent covariate in the linear predictor of the survival submodel. As in (2), we assess the strength of the association through α.

Alternatively, we may be interested in the effect that the slope or rate of change of the biomarker has on survival. This can be achieved by including $\alpha W_i'(t_{ij})$ in the linear predictor of the survival submodel.

Finally, we could link the component models through a time-independent association structure, $\alpha(\beta_k + b_{ik})$, linking the subject-specific deviation from the mean of the kth random effect. A special case of this links the subject-specific random intercept and its effect on survival.

The value of α is simply the log hazard-ratio for a one-unit increase in the longitudinal component included in the survival submodel. Note that if α is estimated to

be 0, that is, no association is present, then the joint model reduces to the two standard separate models. Any combination of the three association structures can be used in the same model: for example, in some settings, both the subject-specific baseline and the current value may be predictive of survival. Choice of association structure should be guided by the clinical question under investigation.

2.5 Maximization

Using Stata's default Newton–Raphson method (see Gould, Pitblado, and Poi [2010]), `stjm` uses a `d0` evaluator program to maximize the likelihood. The joint likelihood in (3) contains an analytically intractable integral where we wish to integrate out the random effects. This can be achieved by using numerical techniques such as simple Gauss–Hermite quadrature (Pinheiro and Bates 1995). Essentially, we can approximate the integral by a weighted summation of the function evaluated at a set of m points, where the m points are the roots of a mth degree Hermite polynomial. Increasing m increases the accuracy of the approximation; however, computation time also increases. Extension to multivariate integrals (random effects) follows naturally; however, computation time will grow exponentially. For example, a model with only a random intercept evaluated with 5-point quadrature evaluates the likelihood at 5 specified points. If this is extended to a random intercept and slope model, with 5-point quadrature for each random effect, then the likelihood is evaluated at $5 \times 5 = 25$ points.

In addition to the full joint likelihood, under an exponential, Weibull, or Gompertz survival submodel, we must use Gauss–Kronrod quadrature to calculate the cumulative hazard. This can be done by using 7- or 15-point quadrature in `stjm`. This is not required when using a flexible parametric survival submodel, because we model on the log cumulative-hazard scale, providing computational benefits.

Crowther, Abrams, and Lambert (2012) note that the use of simple Gauss–Hermite quadrature in the joint model setting can drastically underestimate the standard errors of the parameters in the longitudinal submodel unless a sufficiently high number of quadrature nodes are used. This substantially increases computation time, which is exponentiated with the addition of more random effects. A more complex but accurate extension is to use adaptive Gauss–Hermite quadrature. The implementation of this in Stata in the mixed-model context has been described in Rabe-Hesketh, Skrondal, and Pickles (2002). At the beginning of each full Newton–Raphson iteration, we can center and scale the quadrature node locations for each individual panel, positioning the node matrix in the most appropriate area. This is achieved by using the empirical Bayes estimates and associated standard errors of the random effects for each panel. The use of adaptive quadrature means that a much-reduced number of nodes are required for each random-effects dimension, resulting in substantial computational benefits and much greater accuracy in the estimation.

We caution the user that these models are complex, and sometimes the default estimation algorithm may lead to a model that does not converge. As in all random-effects models, one should be cautious about overmodeling, particularly the number of

random-effects parameters. The majority of previous work on joint models has only considered up to two random effects, that is, intercept and slope (Wulfsohn and Tsiatis 1997). `stjm` can have up to five random effects; however, with a limited data size, it is not feasible to have too complex a model.

2.6 Delayed entry and time-varying covariates

`stjm` has been developed to be entirely consistent with the setup of multiple-record `st` data. We can therefore use `_t0` to denote the measurement times defined as t_{ij} in section 2. This allows both for delayed entry models, which, for example, let age be used as the time scale, and for inclusion of further time-varying covariates within both submodels, assuming they vary at the time of measurements, that is, that they are allowed to change at times `_t0` but are constant within intervals [`_t0`, `_t`).

3 The stjm command

3.1 Syntax

`stjm` *depvar* [*indepvars*] [*if*] [*in*], `panel(`*varname*`)` `survmodel(`*survsubmodel*`)`
[`ffp(`*numlist*`)` `rfp(`*numlist*`)` `timeinteraction(`*varlist*`)` `covariance(`*vartype*`)`
`survcov(`*varlist*`)` `df(#)` `knots(`*numlist*`)` `noorthog` `nocurrent`
`derivassociation` `intassociation` `association(`*numlist*`)`
`assoccovariates(`*varlist*`)` `gh(#)` `gk(#)` `adaptit(#)` `noshowadapt` `atol(#)`
`nonadapt` `fulldata` `nullassoc` *maximize_options* `showinitial` `variance`
`showcons` `keepcons` `level(#)`]

You must `stset` the data into enter and exit times before using `stjm`; see [ST] **stset**. *depvar* is the longitudinal response, and *indepvars* are covariates in the longitudinal submodel. `stjm` uses `_t0` as measurement times and each patient's final row of `_t` as the survival time.

3.2 Options

Required

`panel(`*varname*`)` contains the panel identification variable. Each panel should be identified by a unique integer. `panel()` is required.

`survmodel(`*survsubmodel*`)` specifies the survival submodel to be fit. `survmodel()` is required. *survsubmodel* can be one of the following:

`survmodel(fpm)` fits a flexible parametric survival submodel. This is a highly flexible, fully parametric alternative to the Cox model, modeled on the log cumulative-hazard scale by using restricted cubic splines. For more details, see `stpm2`.

`survmodel(exponential)` fits an exponential survival submodel.

`survmodel(weibull)` fits a Weibull survival submodel.

`survmodel(gompertz)` fits a Gompertz survival submodel.

Longitudinal submodel

`ffp(`*numlist*`)` specifies power transformations of the time variable to be included in the longitudinal submodel as fixed effects. `_t0` is used as the time of measurements. Values must be in {-5, -4, -3, -2, -1, -0.5, 0, 0.5, 1, 2, 3, 4, 5}.

`rfp(`*numlist*`)` specifies power transformations of the time variable to be included in the longitudinal submodel as fixed and random effects. `_t0` is used as the time of measurements. Values must be in {-5, -4, -3, -2, -1, -0.5, 0, 0.5, 1, 2, 3, 4, 5}.

`timeinteraction(`*varlist*`)` specifies covariates to interact with the fixed fractional polynomials of measurement time.

`covariance(`*vartype*`)` specifies the variance–covariance structure of the random effects. *vartype* can be one of the following:

`covariance(independent)` specifies a distinct variance for each random effect, with all covariances equal to 0.

`covariance(exchangeable)` specifies equal variances for all random effects and one common pairwise covariance.

`covariance(identity)` specifies equal variances for all random effects, with all covariances equal to 0.

`covariance(unstructured)` specifies that all variances and covariances are distinctly estimated. This is the default.

Survival submodel

`survcov(`*varlist*`)` specifies covariates to be included in the survival submodel.

`df(`#`)` specifies the degrees of freedom for the restricted cubic spline function used for the baseline cumulative hazard under a flexible parametric survival submodel. # must be between 1 and 10, but usually, a value between 1 and 5 is sufficient.

`knots(`*numlist*`)` specifies knot locations for the baseline distribution function under a flexible parametric survival submodel, as opposed to the default locations set by `df()`. Note that the locations of the knots are placed on the standard time scale. However, the scale used by the restricted cubic spline function is always log time. Default knot positions are determined by the `df()` option.

`noorthog` suppresses orthogonal transformation of spline variables under a flexible parametric survival submodel.

Association

`nocurrent` specifies that the association between the survival and the longitudinal submodels is not based on the current value. The default association is based on the current value of the longitudinal response. If `nocurrent` is invoked, at least one of `intassociation`, `association()`, and `derivassociation` must be specified.

`derivassociation` specifies that the association between the survival and the longitudinal submodels is based on the first derivative of the longitudinal submodel.

`intassociation` specifies that the association between the survival and the longitudinal submodels is based on the random intercept of the longitudinal submodel.

`association(`*numlist*`)` specifies that the association between the survival and the longitudinal submodels is based on a random coefficient of time fractional polynomials specified in `rfp()`.

`assoccovariates(`*varlist*`)` specifies covariates to be included in the linear predictor of the association parameters. Under the default current value association, this corresponds to interacting the longitudinal submodel with covariates.

Maximization

`gh(#)` specifies the number of quadrature points for the simple or adaptive Gauss–Hermite quadrature used to evaluate the joint likelihood. Minimum number of quadrature points is two. The default is `gh(5)` or `gh(15)` under adaptive or simple quadrature, respectively.

`gk(#)` specifies the number of quadrature points for the Gauss–Kronrod quadrature used to evaluate the cumulative hazard under an exponential, Weibull, or Gompertz survival submodel. Two choices are available, either 7 or 15. The default is `gk(15)`.

`adaptit(#)` defines the number of iterations of adaptive Gauss–Hermite quadrature to use in the maximization process. The default is `adaptit(5)`. Adaptive quadrature is implemented at the beginning of each full Newton–Raphson iteration.

`noshowadapt` suppresses the display of the log-likelihood values under the subiterations used to assess convergence of the adaptive quadrature implemented at the beginning of each full Newton–Raphson iteration.

`atol(#)` specifies tolerance for the log likelihood under adaptive quadrature subiterations. The default is `atol(1.0E-05)`.

`nonadapt` uses nonadaptive Gauss–Hermite quadrature to evaluate the joint likelihood. This will generally require a much higher number of nodes, `gh()`, to ensure accurate estimates and standard errors, resulting in much greater computation time.

`fulldata` forces `stjm` to use all rows of data in the survival component of the likelihood. By default, `stjm` assesses whether all covariates specified in `survcov()` are constant within panels; if they are, `stjm` only needs to use the first row of `_t0` and the final row of `_t` in the maximization process, providing considerable advantages in speed.

`nullassoc` sets the initial value for association parameters to 0. Use of the default initial values may in rare situations cause `stjm` to display initial values not feasible. Using this option solves this; however, convergence time is generally longer.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `hessian`, `shownrtolerance`, `tolerance(#)`, `ltolerance(#)`, `gtolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **maximize**. These options are seldom used, but the `difficult` option may be useful if there are convergence problems.

Reporting

`showinitial` displays the output from the `xtmixed` and `stpm2` or `streg` models fit to obtain initial values.

`variance` shows random-effects parameter estimates as variances–covariances.

`showcons` displays the constraints used by `stpm2` and `stjm` for the derivatives of the spline function. This option is only valid under a flexible parametric survival submodel.

`keepcons` prevents the constraints imposed by `stjm` on the derivatives of the spline function when fitting delayed entry models from being dropped. By default, the constraints are dropped. This option is only valid under a flexible parametric survival submodel.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals (CIs). The default is `level(95)` or as set by `set level`.

4 The stjm postestimation command

4.1 Syntax for obtaining best linear unbiased predictions (BLUPs) of random effects or the standard errors of BLUPs

`predict` {*stub*`*` | *newvarlist*}`,` {`reffects` | `reses`}

4.2 Syntax for obtaining other predictions

```
predict newvar [if] [in] [, longitudinal residuals rstandard hazard
    survival cumhazard martingale deviance reffects reses xb fitted m(#)
    at(varname # [varname # ...]) ci timevar(varname) meastime
    survtime zeros]
```

4.3 Options

Longitudinal submodel

`longitudinal` predicts the fitted values for the longitudinal submodel. If `xb` is specified (the default), then only contributions from the fixed portion of the model are included. If `fitted` is specified, then estimates of the random effects are also included.

`residuals` calculates residuals for the longitudinal submodel, equal to the responses minus fitted values. By default, the fitted values take into account the random effects.

`rstandard` calculates standardized residuals, equal to the residuals multiplied by the inverse square root of the estimated error covariance matrix.

Survival submodel

`hazard` calculates the predicted hazard. Default prediction, `xb`, is the average of the fixed portion of the model plus `m()` random draws from the estimated variance–covariance matrix of the random-effects distribution. If `fitted` is specified, then individual specific estimates of the random effects are included with the fixed portion of the model.

`survival` calculates each observation's predicted survival probability. Default prediction, `xb`, is the average of the fixed portion of the model plus `m()` random draws from the estimated variance–covariance matrix of the random-effects distribution. If `fitted` is specified, then individual specific estimates of the random effects are included with the fixed portion of the model.

`cumhazard` calculates the predicted cumulative hazard. Default prediction, `xb`, is the average of the fixed portion of the model plus `m()` random draws from the estimated variance–covariance matrix of the random-effects distribution. If `fitted` is specified, then individual specific estimates of the random effects are included with the fixed portion of the model.

`martingale` calculates martingale-like residuals. Default includes contributions from random effects.

`deviance` calculates the deviance residuals.

Random effects

`reffects` calculates BLUPs of the random effects. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, ..., *stubq* for you.

`reses` calculates the standard errors of the BLUPs of the random effects. You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub*`*` and let Stata name the variables *stub*`1`, ..., *stubq* for you.

Subsidiary

`xb` specifies predictions based on the fixed portion of the model when a `longitudinal` option is specified. When the prediction option is `hazard`, `cumhazard`, or `survival`, the predictions are based on the average of the fixed portion plus `m()` draws from the estimated random-effects variance–covariance matrix.

`fitted` specifies the linear predictor of the fixed portion plus contributions based on predicted random effects.

`m(#)` specifies, when `xb` is chosen, the number of draws from the estimated random-effects variance–covariance matrix in survival submodel predictions.

`at(`*varname* # [*varname* # ...]`)` requests that the covariates specified by the listed *varname*s be set to the listed # values. For example, `at(x1 1 x3 50)` would evaluate predictions at `x1` = 1 and `x3` = 50. This is a useful way to obtain out-of-sample predictions. Note that if `at()` is used together with `zeros`, all covariates not listed in `at()` are set to 0. If `at()` is used without `zeros`, then all covariates not listed in `at()` are set to their sample values. See also `zeros`.

`ci` calculates a CI for the requested statistic and stores the confidence limits in *newvar*`_lci` and *newvar*`_uci`.

`timevar(`*varname*`)` defines the variable used as time in the predictions. This is useful for large datasets where for plotting purposes, predictions are only needed for, say, 200 observations. Note that you should take some caution when using this option because predictions may be made at whatever covariate values are in the first 200 rows of data. This can be avoided by using the `at()` option or the `zeros` option to define the covariate patterns for which you require the predictions.

`meastime` evaluates predictions at measurement times, that is, `_t0`. Default for longitudinal submodel predictions.

`survtime` evaluates predictions at survival times, that is, `_t`. Default for survival submodel predictions.

`zeros` sets all covariates to 0 (baseline prediction). For example, `predict s0, survival zeros` calculates the baseline survival function. See also `at()`.

5 The stjmgraph command

A subsidiary command, `stjmgraph`, is available. This creates a longitudinal trajectory plot whereby the time scale is adjusted by taking away each patient's event or censoring time. This form of graph can be useful to display joint longitudinal and survival data, giving an indication of any association between the two processes. A separate plot is created for patients who were censored and for patients who experienced the event of interest. They are then combined by using `graph combine`.

5.1 Syntax

`stjmgraph` *depvar* [*if*] [*in*] , `panel(`*varname*`)` [`censgraphopts(`*string*`)` `eventgraphopts(`*string*`)` `combineopts(`*string*`)` `draw` `lowess`]

The dataset must be `stset`, as described for `stjm`.

5.2 Options

`panel(`*varname*`)` defines the panel identification variable. `panel()` is required.

`censgraphopts(`*string*`)` pass options to the `twoway graph` of censored observations; see [G-3] ***twoway_options***.

`eventgraphopts(`*string*`)` pass options to the `twoway graph` of observations who experienced the event of interest; see [G-3] ***twoway_options***.

`combineopts(`*string*`)` pass options to the final `graph combine`; see [G-2] **graph combine**.

`draw` displays the intermediate twoway plots used to create the final graph.

`lowess` overlays a lowess smoother to each graph to aid interpretation.

6 Example

We illustrate `stjm` through application to a dataset of 312 patients with primary biliary cirrhosis (see Murtaugh et al. [1994]). Of the 312, 158 were randomized to receive D-penicillamine, and 154 assigned a placebo. Serum bilirubin was measured repeatedly at intermittent time points. We investigate the effect of treatment after adjusting for the relationship between serum bilirubin levels and time to death. Because of right skewness, in all analyses, we work with log(serum bilirubin).

The dataset must be correctly `stset` for use with `stjm` through the use of start and stop times. This allows `stjm` to use `_t0` as the measurement times and the final row of `_t` as the survival times. We illustrate the data structure below:

```
. use fullpbc
. stset stop, enter(start) f(event=1) id(id)
                id:  id
     failure event:  event == 1
obs. time interval:  (stop[_n-1], stop]
 enter on or after:  time start
 exit on or before:  failure
------------------------------------------------------------------------------
     1945  total obs.
        0  exclusions
------------------------------------------------------------------------------
     1945  obs. remaining, representing
      312  subjects
      140  failures in single failure-per-subject data
 2000.307  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =  14.30566
. list id logb drug _t0 _t _d if id==3 | id==5, noobs sepby(id)
```

id	logb	drug	_t0	_t	_d
3	.3364722	D-penicil	0	.48187494	0
3	.0953102	D-penicil	.48187494	.99660498	0
3	.4054651	D-penicil	.99660498	2.0342789	0
3	.5877866	D-penicil	2.0342789	2.7707808	1
5	1.223776	placebo	0	.54484725	0
5	.6418539	placebo	.54484725	1.070529	0
5	.9162908	placebo	1.070529	2.1054649	0
5	1.740466	placebo	2.1054649	3.0062425	0
5	1.648659	placebo	3.0062425	3.9836819	0
5	2.944439	placebo	3.9836819	4.1205783	0

Here we have two patients with four and six measurements of log(serum bilirubin), respectively. The data have been `stset`, allowing `_t0` to be used to denote the time that measurements were taken and the final row (for each patient) of `_t` to denote the survival time. We can explore the joint data by using `stjmgraph`. We use the `lowess` option to aid interpretation.

```
. stjmgraph logb, panel(id) lowess
```

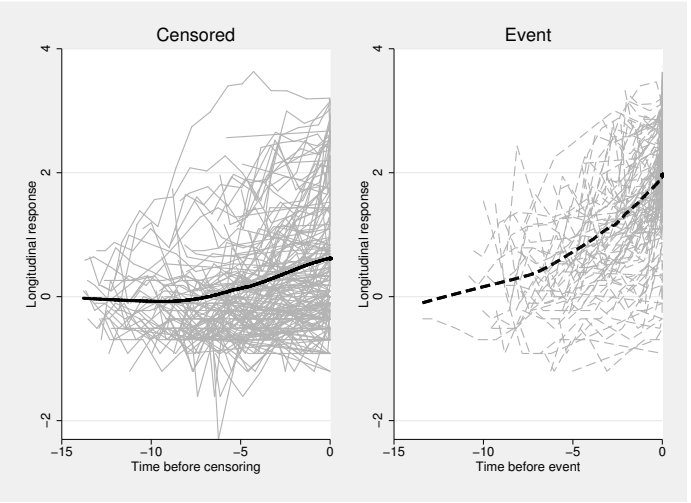

Figure 1. Longitudinal profiles of log(serum bilirubin) for patients who were censored or who died. Time scale is adjusted by taking away each patient's survival time.

Figure 1 displays all patients' longitudinal trajectories against time, across died and censoring status, with the time scale adjusted by subtracting each patient's survival or censoring time. We could restrict the plotted sample by using the `if` or `in` qualifier. There appears to be a generally increasing trend that is much sharper in patients who died than in those who were censored. This is indicative of a positive association between longitudinal response and time to death, whereby a higher level of the biomarker appears to be associated with time to death. We now investigate this formally by using `stjm`.

We model the longitudinal process by using a linear trajectory model with random intercept and slope, adjusting for treatment group. We model the survival process by using a Weibull proportional hazards survival submodel and adjusting for treatment group. We use the default current value association and the default `unstructured` form for the random-effects variance–covariance matrix.

```
. stjm logb trt, panel(id) survm(weibull) rfp(1) survcov(trt)
-> gen double _time_1 = X^(1)
(where X = _t0)
Obtaining initial values:
Fitting full model:
-> Conducting adaptive Gauss-Hermite quadrature
-- Iteration 0:   Adapted log likelihood = -1920.5096
-- Iteration 1:   Adapted log likelihood = -1923.2378
-- Iteration 2:   Adapted log likelihood = -1923.2206
-- Iteration 3:   Adapted log likelihood = -1923.2214
  (output omitted)
Joint model estimates                           Number of obs     =      1945
Panel variable: id                              Number of panels  =       312
                                                Number of failures =      140
Log-likelihood = -1918.5172
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Longitudinal						
_time_1	.1848437	.0132919	13.91	0.000	.1587921	.2108953
trt	-.1313587	.1120029	-1.17	0.241	-.3508803	.0881629
_cons	.5591394	.0812295	6.88	0.000	.3999324	.7183463
Survival						
assoc:value						
_cons	1.240947	.0931014	13.33	0.000	1.058471	1.423422
ln_lambda						
trt	.0389711	.1790989	0.22	0.828	-.3120563	.3899985
_cons	-4.408948	.2738691	-16.10	0.000	-4.945722	-3.872175
ln_gamma						
_cons	.0189773	.0827617	0.23	0.819	-.1432327	.1811874

Random effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
id: Unstructured				
sd(_time_1)	.1805185	.0123477	.1578695	.2064167
sd(_cons)	1.00034	.0425768	.9202769	1.087369
corr(_time_1,_cons)	.4247242	.0727761	.2723586	.5563106
sd(Residual)	.3471654	.0066731	.3343297	.3604939

```
 Longitudinal submodel: Linear mixed effects model
     Survival submodel: Weibull proportional hazards model
    Integration method: Adaptive Gauss-Hermite quadrature using 5 nodes
     Cumulative hazard: Gauss-Kronrod quadrature using 15 nodes
```

We observe a nonstatistically significant direct treatment effect on log (serum bilirubin) of -0.131 (95% CI: $[-0.351, 0.088]$). A nonstatistically significant direct treatment effect on survival is observed of 0.039 (95% CI: $[-0.312, 0.390]$). However, a highly positive statistically significant association can be seen of 1.241 (95% CI: $[1.058, 1.423]$), indicating that a higher value of log (serum bilirubin) increases the risk of death. This corresponds to a hazard ratio for a one-unit increase in the value of the time-dependent biomarker of 3.459 (95% CI: $[2.881, 4.150]$). This is consistent with figure 1. Because we

have adjusted for treatment in both submodels, we can calculate an overall treatment effect on survival. For example, we have $\alpha = 1.241$, $\delta = -0.131$, and $\phi = 0.039$. The overall log hazard-ratio for the effect of treatment is therefore $\alpha\delta + \phi$. This can be calculated as follows:

```
. nlcom [alpha_1][_cons]*[Longitudinal][trt] + [ln_lambda][trt]
       _nl_1:  [alpha_1][_cons]*[Longitudinal][trt] + [ln_lambda][trt]
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_nl_1	-.124038	.2293071	-0.54	0.589	-.5734717	.3253957

This shows a nonstatistically significant log hazard-ratio due to treatment of -0.124 (95% CI: $[-0.573, 0.325]$). Standard predictions can be obtained following an `stjm` fit. Fitted values and standardized residuals can be plotted against each other to evaluate model fit.

```
. predict longfitvals, fitted longitudinal
. predict stresids, rstandard
. scatter stresids longfitvals, yline(0) ytitle("Standardized residuals")
> xtitle("Fitted values") title("Fitted values vs. residuals")
```

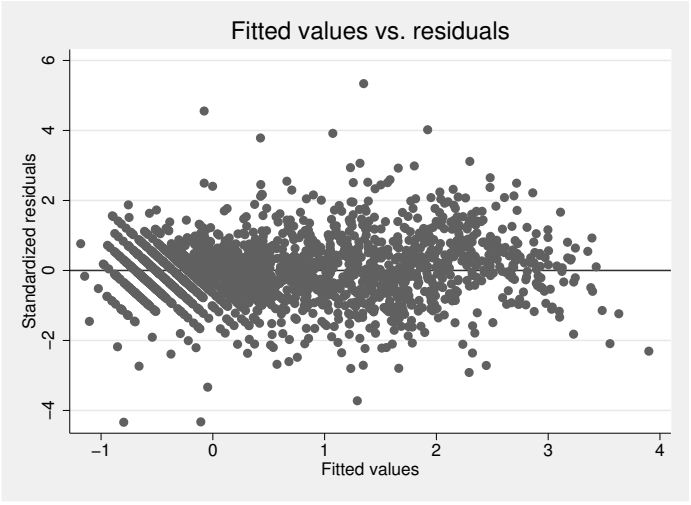

Figure 2. Fitted values versus standardized residuals to assess model fit

Note that the longitudinal residuals described in this article must be interpreted with caution because of the inherent missing-data process underpinning the longitudinal process. A form of multiple-imputed residuals has been proposed by Rizopoulos, Verbeke, and Molenberghs (2010).

We can also compare predicted values of the survival function with the Kaplan–Meier estimate.

```
. predict survfit, xb survival
```

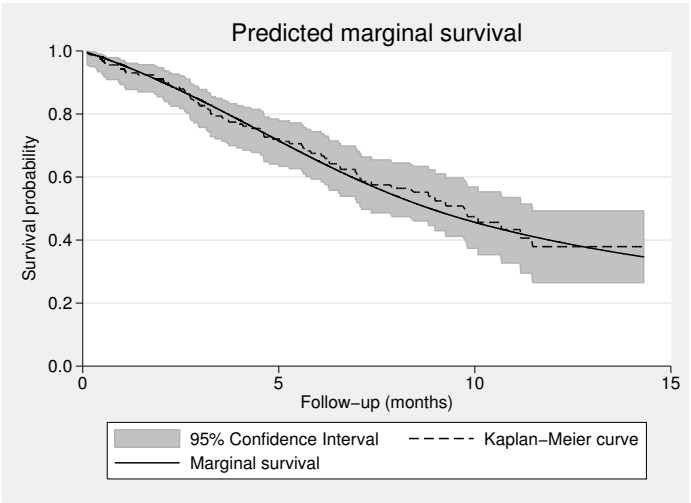

Figure 3. Predicted survival function for patients in the treatment group

One of the benefits of fitting joint models within a shared parameter framework is the ability to tailor predictions at the individual level. The set of `fitted` predictions described above is not exhaustive and does not include conditional survival predictions, whereby we wish to predict a patient's survival conditional on a set of observed longitudinal measurements. A Monte Carlo scheme has been proposed by Rizopoulos (2011) to fully account for variability in parameter estimates and in empirical Bayes estimates of the random effects. This proposal is currently being implemented in Stata.

7 Discussion

The new `stjm` command implements shared parameter joint modeling of longitudinal and survival data within Stata. It provides a highly flexible framework for both the longitudinal submodel through the use of fractional polynomials and the survival submodel through the four choices of submodel. Through the implementation of adaptive Gauss–Hermite quadrature, accurate estimates of effect can be obtained by using a much-reduced number of quadrature nodes, resulting in substantial computational benefits.

The software is being constantly updated and improved, and we aim to write further articles for the *Stata Journal* to include the extension to competing risks, the inclusion of a cure proportion, and the allowance of categorical longitudinal responses.

8 Acknowledgments

We thank an associate editor for constructive comments that greatly improved the article. Part of this work was conducted when Michael Crowther undertook an internship at StataCorp. He would like to thank all the people at StataCorp for their hospitality, in particular, Yulia Marchenko, Jeff Pitblado, Alan Riley, and Vince Wiggins.

Michael Crowther was funded by a National Institute for Health Research Methods Fellowship (RP-PG-0407-10314).

9 References

Billingham, L. J., and K. R. Abrams. 2002. Simultaneous analysis of quality of life and survival data. *Statistical Methods in Medical Research* 11: 25–48.

Crowther, M. J., K. R. Abrams, and P. C. Lambert. 2012. Flexible parametric joint modelling of longitudinal and survival data. *Statistics in Medicine* 31: 4456–4471.

Durrleman, S., and R. Simon. 1989. Flexible regression models with cubic splines. *Statistics in Medicine* 8: 551–561.

Gould, W., J. Pitblado, and B. Poi. 2010. *Maximum Likelihood Estimation with Stata*. 4th ed. College Station, TX: Stata Press.

Guo, X., and B. P. Carlin. 2004. Separate and joint modeling of longitudinal and event time data using standard computer packages. *American Statistician* 58: 16–24.

Henderson, R., P. Diggle, and A. Dobson. 2000. Joint modelling of longitudinal measurements and event time data. *Biostatistics* 1: 465–480.

Ibrahim, J. G., H. Chu, and L. M. Chen. 2010. Basic concepts and methods for joint models of longitudinal and survival data. *Journal of Clinical Oncology* 28: 2796–2801.

Lambert, P. C., and P. Royston. 2009. Further development of flexible parametric models for survival analysis. *Stata Journal* 9: 265–290.

Murtaugh, P. A., E. R. Dickson, G. M. V. Dam, M. Malinchoc, P. M. Grambsch, A. L. Langworthy, and C. H. Gips. 1994. Primary biliary cirrhosis: Prediction of short-term survival based on repeated patient visits. *Hepatology* 20: 126–134.

Pantazis, N., and G. Touloumi. 2010. Analyzing longitudinal data in the presence of informative drop-out: The jmre1 command. *Stata Journal* 10: 226–251.

Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4: 12–35.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal* 2: 1–21.

Rizopoulos, D. 2011. Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data. *Biometrics* 67: 819–829.

Rizopoulos, D., G. Verbeke, and G. Molenberghs. 2010. Multiple-imputation-based residuals and diagnostic plots for joint models of longitudinal and survival outcomes. *Biometrics* 66: 20–29.

Royston, P., and D. G. Altman. 1994. Regression using fractional polynomials of continuous covariates: Parsimonious parametric modelling (with discussion). *Journal of the Royal Statistical Society, Series C* 43: 429–467.

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.

Wulfsohn, M. S., and A. A. Tsiatis. 1997. A joint model for survival and longitudinal data measured with error. *Biometrics* 53: 330–339.

About the authors

Michael Crowther is a research associate in medical statistics. His main interest is the development and application of joint models for longitudinal and survival data.

Keith Abrams is a professor of medical statistics who maintains an active research interest in the joint modeling of longitudinal and survival data.

Paul Lambert is a reader in medical statistics. His main interest is in the development and application of methods in population-based cancer research.