stgenreg: A Stata package for general parametric survival analysis

Stata UK Meeting
13th September 2012

Michael J. Crowther[1*] and Paul C. Lambert[1,2]

[1]Department of Health Sciences
University of Leicester, UK.

[2]Department of Medical Epidemiology and Biostatistics
Karolinska Institutet, Sweden.

[*]michael.crowther@le.ac.uk

Background

- Most popular survival model is the Cox (Cox, 1972)
- Parametric survival models are used extensively
- More flexible parametric models are becoming popular (Royston and Lambert, 2011)
- Advantages in terms of prediction, extrapolation, quantification

Background

Standard parametric model estimated using maximum likelihood:

$$
\begin{aligned}
l_i &= \log \left\{ f(t_i)^{d_i} \left(\frac{S(t_i)}{S(t_{0i})} \right)^{1-d_i} \right\} \\
&= d_i \log\{f(t_i)\} + (1-d_i)\log\{S(t_i)\} \\
&\quad - (1-d_i)\log\{S(t_{0i})\} \qquad (1)
\end{aligned}
$$

Using Equation (1) we can directly maximise the log-likelihood if using known probability density and survival functions.

Background

Alternatively, using $f(t) = h(t)S(t)$ we can write

$$\begin{aligned} l_i &= \log\left\{h(t_i)^{d_i}\frac{S(t_i)}{S(t_{0i})}\right\} \\ &= d_i \log\{h(t_i)\} + \log\{S(t_i)\} - \log\{S(t_{0i})\} \end{aligned} \tag{2}$$

which becomes

$$l_i = d_i \log\{h(t_i)\} - \int_{t_{0i}}^{t_i} h(u)\mathrm{d}u \tag{3}$$

So, we only need a hazard function...

$$l_i = d_i \log\{h(t_i)\} - \int_{t_{0i}}^{t_i} h(u)\mathrm{d}u \tag{4}$$

For example a Weibull model:

$$\begin{aligned} l_i &= d_i \log\{\lambda\gamma t_i^{\gamma-1}\} - \int_{t_{0i}}^{t_i} \lambda\gamma u^{\gamma-1}\mathrm{d}u \\ &= d_i \log\{\lambda\gamma t_i^{\gamma-1}\} - \lambda t_i^{\gamma} + \lambda t_{0i}^{\gamma} \end{aligned}$$

But what if we can't evaluate the integral in Equation (4) analytically?

Numerical Integration

Gaussian quadrature allows us to evaluate an analytically intractible integral through a weighted sum of a function evaluated at a set of pre-defined points, known as nodes (Stoer and Burlirsch, 2002). We have

$$\int_{-1}^{1} h(x)\mathrm{d}x = \int_{-1}^{1} W(x)g(x)\mathrm{d}x \approx \sum_{i=1}^{k} w_i g(x_i) \qquad (5)$$

Numerical Integration

The integral over $[t_{0i}, t_i]$ in equation (3) must be changed to an integral over $[-1, 1]$ using the following rule

$$\int_{t_{0i}}^{t_i} h(x)\mathrm{d}x = \frac{t_i - t_{0i}}{2}\int_{-1}^{1} h\left(\frac{t_i - t_{0i}}{2}x + \frac{t_{0i}+t_i}{2}\right)\mathrm{d}x$$

$$\approx \frac{t_i - t_{0i}}{2}\sum_{i=1}^{k} w_i h\left(\frac{t_i - t_{0i}}{2}x_i + \frac{t_{0i}+t_i}{2}\right) \quad (6)$$

Really useful property of this is that delayed entry is accounted for.

General parametric survival modelling framework

$$l_i = d_i \log\{h(t_i)\} - \int_{t_{0i}}^{t_i} h(u) \mathrm{d}u$$

- Using quadrature we now have a general framework to estimate a survival model using almost *any* user-defined hazard function
- Default is Gauss-Legendre, with weight function $= 1$

Syntax

`stgenreg [if] [in] [, options]`

- `loghazard(string)`

  e.g. `loghazard([xb])`

- `hazard(string)`

  e.g. `hazard(exp([xb]))`

An equation name specified in square brackets in `loghazard()`/`hazard()` then becomes an option through a second level of parsing

- `xb(string)`

  e.g. `xb(trt gender)`

This is simply an exponential survival model `xb(string)` is actually `xb(`*comp1* `| ... |` *compn*`)`

Component	Description
`varlist [, nocons]`	the user may specify a standard variable list within a component section, with an optional `nocons` option
`g(#t)`	where `g()` is any user defined function of `#t` written in Mata code, e.g. `#t:^2`
`#rcs(`*options*`)`	creates restricted cubic splines of either log time or time. Options include `df(int)`, the number of degrees of freedom, `noorthog` which turns off the default orthogonalisation, `time`, which creates splines using time rather than log time, the default, and `offset(varname)` to include an offset when calculating the splines. See `rcsgen` for more details.

`xb(string)` is actually `xb(`*comp1* `| ... |` *compn*`)`

Component	Description
`#fp(numlist [,`*options*`])`	creates fractional polynomials of time with powers defined in `numlist`. If 0 is specified, log time is generated. The only current option is `offset()` which is consistent with that described in `#rcs()` above.
`varname:*f(#t)`	to include time-dependent effects, where `f(#t)` is one of `#rcs()`, `#fp()` or `g()`.

Further options

- `bhazard(varname)` - invokes relative survival models, defining the expected hazard rate at the time of event
- `jacobi` - invokes Gauss-Jacobi quadrature to evaluate the cumulative hazard
- `eform` - exponentiate coefficients of the first `ml` equation
- `showcomponent` - displays each parsed component (useful for syntax checking)

Implementation (briefly)

```
. pr define stgenreg_d0
  (output omitted)
    qui gen double `logh´ = .
    mata: logh = $mataloghazard1
    mata: st_store(.,"`logh´",touse,logh)

    if "$bhazvar"=="" {
        local lnht `logh´ + ln(_t)  //standard model
    }
    else {
        local lnht ln($bhazvar + exp(`logh´))   //rel surv model
    }

    qui gen double `ch´ = .
    mata: cumhaz("`ch´",touse,knewnodes1,kweights1,
                nnodes1 `pnames´ `pcoefnames´ $arraynames)
    qui mlsum `lnf´ = _d*(`lnht´) - `ch´
. end
```

Implementation (briefly)

```
. mata:
: void cumhaz(string scalar chvar,
>             string scalar touse,
>             numeric matrix knewnodes1,
>             numeric matrix kweights1,
>             real scalar nnodes1
>             $matasyntax
>             $coefficientmats
>             $arraysyntax)
> {
>     st_view(cumhaz=.,.,chvar,touse)
>     cumhazard = J(rows(knewnodes1),1,0)
>
>     for(j=1;j<=nnodes1;j++) {
>         cumhazard = cumhazard :+ kweights1[,j]:*($mataloghazard21)
>     }
>     cumhaz[,]=cumhazard
> }
: end
```

Example dataset

- Dataset comprising of 9721 women aged under 50 and diagnosed with breast cancer in England and Wales between 1986 and 1990
- Event of interest is death from any cause, with follow-up restricted to 5 years.
- Deprivation was categorised into 5 levels; however, we have restricted the analyses to comparing the most affluent and most deprived groups, for illustrative purposes. We therefore only consider a binary covariate, `dep5`, with 0 for the most affluent and 1 for the most deprived group

Example I: Proof of concept

We can compare a standard Weibull model using `streg`, to the equivalent model using `stgenreg`:

```
.  streg dep5, dist(w) nohr

.  stgenreg, loghazard([ln_lambda] :+ [ln_gamma] :+ ///
> (exp([ln_gamma]) :- 1) :* log(#t)) ln_lambda(dep5)
```

We can further compare how well the numerical integration performs with a varying number of quadrature nodes

Optimised model and node comparison

Variable	streg	stgenreg15	stgenreg30	stgenreg50	stgenr~100
#1					
dep5	.2698715	.26983514	.26986326	.26986899	.26987095
	.0392017	.03920178	.03920173	.03920172	.03920171
_cons	-2.8252423	-2.8232443	-2.8248136	-2.8251059	-2.8252139
	.03694985	.03718485	.03701515	.03697471	.03695639
#2					
_cons	.04673335	.04542627	.04645138	.04664313	.04671442
	.01792781	.01812554	.01798227	.01794843	.0179332
Statistics					
ll	-8808.0854	-8808.3461	-8808.149	-8808.1075	-8808.0906

legend: b/se

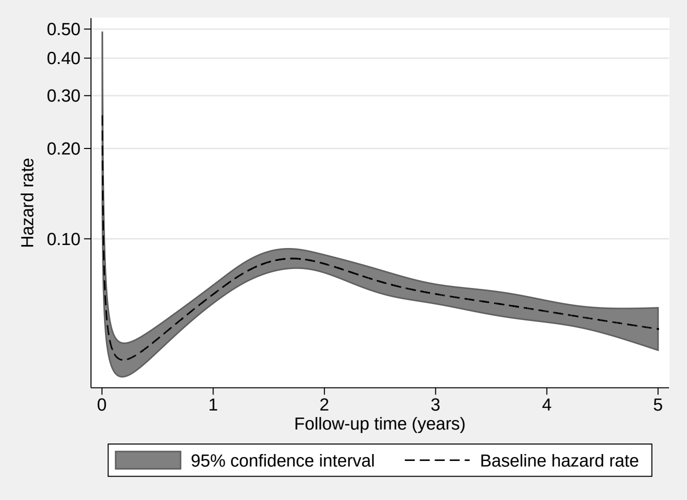
Example II: Models unavailable in Stata

Splines for the log baseline hazard function

```
. stgenreg, loghazard([xb]) xb(dep5 | #rcs(df(5))) nolog
Variables _eq1_cp2_rcs1 to _eq1_cp2_rcs5 were created
Log likelihood = -8750.1403                    Number of obs   =       9721
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
dep5	.2691643	.0392021	6.87	0.000	.1923297	.345999
_eq1_cp2_rcs1	-.0723057	.0275693	-2.62	0.009	-.1263404	-.0182709
_eq1_cp2_rcs2	.0638052	.0196604	3.25	0.001	.0252715	.102339
_eq1_cp2_rcs3	.1301083	.0181169	7.18	0.000	.0945999	.1656167
_eq1_cp2_rcs4	-.031646	.014479	-2.19	0.029	-.0600243	-.0032677
_eq1_cp2_rcs5	.0065428	.0134478	0.49	0.627	-.0198144	.0329
_cons	-2.916613	.0608087	-47.96	0.000	-3.035795	-2.79743

```
Quadrature method: Gauss-Legendre with 15 nodes
```




```
. predict haz1, hazard ci zeros
```

Example II: Models unavailable in Stata

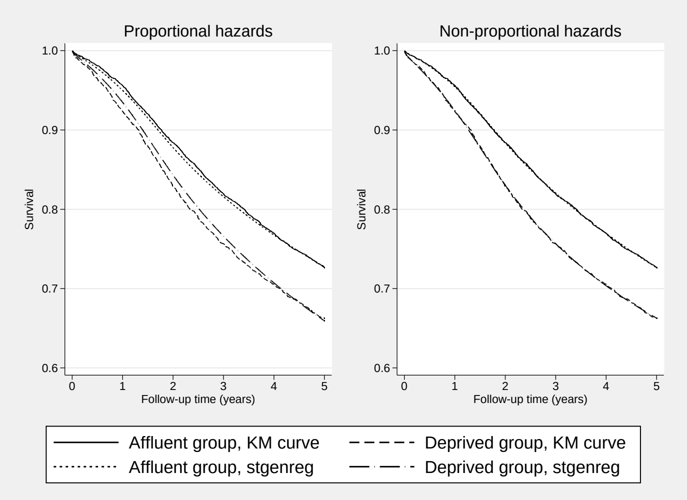
Splines for the log baseline hazard function and time-dependent effect

```
. stgenreg, loghazard([xb]) xb(dep5 | #rcs(df(5)) | dep5:*#rcs(df(3))) nodes(30)
Variables _eq1_cp2_rcs1 to _eq1_cp2_rcs5 were created
Variables _eq1_cp3_rcs1 to _eq1_cp3_rcs3 were created

Log likelihood = -8747.3275                      Number of obs   =       9721
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
dep5	.0723415	.0924005	0.78	0.434	-.1087602	.2534433
_eq1_cp2_rcs1	-.0108058	.0309504	-0.35	0.727	-.0714673	.0498558
_eq1_cp2_rcs2	.0672877	.0224852	2.99	0.003	.0232177	.1113578
_eq1_cp2_rcs3	.1128672	.0207167	5.45	0.000	.0722634	.1534711
_eq1_cp2_rcs4	-.0261438	.0145455	-1.80	0.072	-.0546525	.002365
_eq1_cp2_rcs5	.0014202	.0134079	0.11	0.916	-.0248589	.0276992
_eq1_cp3_rcs1	-.1464002	.0443983	-3.30	0.001	-.2334194	-.0593811
_eq1_cp3_rcs2	.0425164	.0333753	1.27	0.203	-.022898	.1079307
_eq1_cp3_rcs3	.0135896	.0322604	0.42	0.674	-.0496396	.0768187
_cons	-2.849318	.0649361	-43.88	0.000	-2.976591	-2.722046

```
Quadrature method: Gauss-Legendre with 30 nodes
```




```
. predict s1, survival
```

Example III: Models unavailable in Stata

Generalised gamma with proportional hazards

```
. local mu [mu]
. local sigma exp([ln_sigma])
. local kappa [kappa]
. local gamma (abs(`kappa´):^(-2))
. local z (sign(`kappa´):*(log(#t):-`mu´):/(`sigma´))
. local u ((`gamma´):*exp(abs(`kappa´):*(`z´)))
. local surv1 (1:-gammap(`gamma´,`u´)):*(`kappa´:>0)
. local surv2 (1:-normal(`z´)):*(`kappa´:==0)
. local surv3 gammap(`gamma´,`u´):*(`kappa´:<0)
. local pdf1 ((`gamma´:^`gamma´):*exp(`z´:*sqrt(`gamma´):-`u´):/(`sigma´:*#t:*s
> qrt(`gamma´):*gamma(`gamma´))):*(`kappa´:!=0)
. local pdf2 (exp(-(`z´:^2):/2):/(`sigma´:*#t:*sqrt(2:*pi()))):*(`kappa´:==0)
. local haz (`pdf1´ :+ `pdf2´):/(`surv1´ :+ `surv2´ :+ `surv3´)
. stgenreg, hazard(exp([xb]):*(`haz´)) nodes(30) xb(dep5,nocons)
```

Example III: Models unavailable in Stata

Generalised gamma with proportional hazards

```
. stgenreg, hazard(exp([xb]):*(`haz´)) nodes(30) xb(dep5,nocons)
Log likelihood = -8801.2754                         Number of obs   =       9721
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
xb						
dep5	.2694578	.0391992	6.87	0.000	.1926289	.3462868
kappa						
_cons	.6752793	.0749985	9.00	0.000	.528285	.8222735
mu						
_cons	2.710497	.032793	82.65	0.000	2.646224	2.774771
ln_sigma						
_cons	.1727204	.0521935	3.31	0.001	.0704231	.2750178

```
Quadrature method: Gauss-Legendre with 30 nodes
```

stgenreg as a development tool

- stgenreg will clearly not be the most computationally efficient and numerically accurate way to implement some models
- For example, the estimation process when using restricted cubic splines to model the baseline hazard function can be improved
- The restricted component assumes a linear trend before and after the boundary knots - in which we can directly integrate the hazard function
- This improved routine will be available as strcs

Discussion

- `stgenreg` is a general framework for the parametric analysis of survival data
- It is extremely flexible though requires careful use
- Struggles when log hazard wanders off to $\pm\infty$ - but just increase nodes
- Extensions:
  - Competing risks - `stgenregcif`
  - Multi-state models
- To be released...soon

References I

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

P. Royston and P. C Lambert. *Flexible Parametric Survival Analysis Using Stata: Beyond the Cox Model*. Stata Press, 2011.

J. Stoer and R. Burlirsch. *Introduction to Numerical Analysis*. Springer, 3rd edition, 2002.