An Introduction to Survival Analysis Using Stata

Third Edition

MARIO CLEVES
Department of Pediatrics
University of Arkansas Medical Sciences

WILLIAM GOULD
StataCorp

ROBERTO G. GUTIERREZ
StataCorp

YULIA V. MARCHENKO
StataCorp

A Stata Press Publication
StataCorp LP
College Station, Texas

 First edition 2002
Revised edition 2004
Second edition 2008
Third edition 2010

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2_ε
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-59718-074-2
ISBN-13: 978-1-59718-074-0

Contents

(Pages omitted)

Preface to the Third Edition

This third edition updates the second edition to reflect the additions to the software made in Stata 11, which was released in July 2009. The updates include syntax and output changes. The two most notable differences here are Stata's new treatment of factor (categorical) variables and Stata's new syntax for obtaining predictions and other diagnostics after `stcox`.

As of Stata 11, the `xi:` prefix for specifying categorical variables and interactions has been deprecated. Whereas in previous versions of Stata, you might have typed

```
. xi: stcox i.drug*i.race
```

to obtain main effects on `drug` and `race` and their interaction, in Stata 11 you type

```
. stcox i.drug##i.race
```

Furthermore, when you used `xi:`, Stata created indicator variables in your data that identified the levels of your categorical variables and interactions. As of Stata 11, the calculations are performed intrinsically without generating any additional variables in your data.

Previous to Stata 11, if you wanted residuals or other diagnostic measures for Cox regression, you had to specify them when you fit your model. For example, to obtain Schoenfeld residuals you might have typed

```
. stcox age protect, schoenfeld(sch*)
```

to generate variables `sch1` and `sch2` containing the Schoenfeld residuals for `age` and `protect`, respectively. This has been changed in Stata 11 to be more consistent with Stata's other estimation commands. The new syntax is

```
. stcox age protect
. predict sch*, schoenfeld
```

Chapter 4 has been updated to describe the subtle difference between right-censoring and right-truncation, while previous editions had treated these concepts as synonymous.

Chapter 9 includes an added section on Cox regression that handles missing data with multiple imputation. Stata 11's new `mi` suite of commands for imputing missing data and fitting Cox regression on multiply imputed data are described. `mi` is discussed in the context of `stcox`, but what is covered there applies to `streg` and `stcrreg` (which also is new to Stata 11), as well.

Chapter 11 includes added discussion of three new diagnostic measures after Cox regression. These measures are supported in Stata 11: DFBETA measures of influence, LMAX values, and likelihood displacement values. In previous editions, DFBETAs were discussed, but they required manual calculation.

Chapter 17 is new and describes methods for dealing with competing risks, where competing failure events impede one's ability to observe the failure event of interest. Discussion focuses around the estimation of cause-specific hazards and of cumulative incidence functions. The new `stcrreg` command for fitting competing-risks regression models is introduced.

College Station, Texas
July 2010

Mario A. Cleves
William W. Gould
Roberto G. Gutierrez
Yulia V. Marchenko

Preface to the Second Edition

This second edition updates the revised edition (revised to support Stata 8) to reflect Stata 9, which was released in April 2005, and Stata 10, which was released in June 2007. The updates include the syntax and output changes that took place in both versions. For example, as of Stata 9 the `estat phtest` command replaces the old `stphtest` command for computing tests and graphs for examining the validity of the proportional-hazards assumption. As of Stata 10, all `st` commands (as well as other Stata commands) accept option `vce(`*vcetype*`)`. The old `robust` and `cluster(`*varname*`)` options are replaced with `vce(robust)` and `vce(cluster` *varname*`)`. Most output changes are cosmetic. There are slight differences in the results from `streg, distribution(gamma)`, which has been improved to increase speed and accuracy.

Chapter 8 includes a new section on nonparametric estimation of median and mean survival times. Other additions are examples of producing Kaplan–Meier curves with at-risk tables and a short discussion of the use of boundary kernels for hazard function estimation.

Stata's facility to handle complex survey designs with survival models is described in chapter 9 in application to the Cox model, and what is described there may also be used with parametric survival models.

Chapter 10 is expanded to include more model-building strategies. The use of fractional polynomials in modeling the log relative-hazard is demonstrated in chapter 10. Chapter 11 includes a description of how fractional polynomials can be used in determining functional relationships, and it also includes an example of using concordance measures to evaluate the predictive accuracy of a Cox model.

Chapter 16 is new and introduces power analysis for survival data. It describes Stata's ability to estimate sample size, power, and effect size for the following survival methods: a two-sample comparison of survivor functions and a test of the effect of a covariate from a Cox model. This chapter also demonstrates ways of obtaining tabular and graphical output of results.

College Station, Texas
March 2008

Mario A. Cleves
William W. Gould
Roberto G. Gutierrez
Yulia V. Marchenko

(Pages omitted)

8 Nonparametric analysis

The previous two chapters served as a tutorial on `stset`. Once you `stset` your data, you can use any `st` survival command, and the nice thing is that you do not have to continually restate the definitions of analysis time, failure, and rules for inclusion.

As previously discussed in chapter 1, the analysis of survival data can take one of three forms—nonparametric, semiparametric, and parametric—all depending on what we are willing to assume about the form of the survivor function and about how the survival experience is affected by covariates.

Nonparametric analysis follows the philosophy of letting the dataset speak for itself and making no assumption about the functional form of the survivor function (and thus no assumption about, for example, the hazard, cumulative hazard). The effects of covariates are not modeled, either—the comparison of the survival experience is done at a qualitative level across the values of the covariates.

Most of Stata's nonparametric survival analysis is performed via the `sts` command, which calculates estimates, saves estimates as data, draws graphs, and performs tests, among other things; see [ST] **sts**.

8.1 Inadequacies of standard univariate methods

Before we proceed, however, we must discuss briefly the reasons that the typical preliminary data analysis tools do not translate well into the survival analysis paradigm. For example, the most basic of analyses would be one that analyzed the mean time to failure or the median time to failure. Let us use the hip-fracture dataset, which we `stset` at the end of chapter 7:

```
. use http://www.stata-press.com/data/cggm3/hip2
(hip fracture study)
. list id _t0 _t fracture protect age calcium if 20<=id & id<=22, sepby(id)

     +--------------------------------------------------------+
     | id   _t0   _t   fracture   protect   age   calcium |
     |--------------------------------------------------------|
 32. | 20     0    5          0         0    67     11.19 |
 33. | 20     5   15          0         0    67     10.68 |
 34. | 20    15   23          1         0    67     10.46 |
     |--------------------------------------------------------|
 35. | 21     0    5          0         1    82      8.97 |
 36. | 21     5    6          1         1    82      7.25 |
     |--------------------------------------------------------|
 37. | 22     0    5          0         1    80      7.98 |
 38. | 22     5    6          0         1    80      9.65 |
     +--------------------------------------------------------+
```

Putting aside for now the possible effects of the covariates, if we were interested in estimating the population mean time to failure, we might be tempted to use the standard tools such as

```
. ci _t
    Variable |      Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
          _t |      106     11.5283    .8237498        9.894958    13.16165
```

We might quickly realize that this is not what we want because there are multiple records for each individual. We could just consider those values of `_t` corresponding to the last record for each individual,

```
. sort id _t
. by id: gen last = _n==_N
. ci _t if last
    Variable |      Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
          _t |       48        15.5    1.480368        12.52188    18.47812
```

and we now have a mean based on 48 observations (one for each subject). This will not serve, however, because `_t` does not always correspond to failure time—some times in our data are censored, meaning that the failure time in these cases is known only to be greater than `_t`. As such, the estimate of the mean is biased downward.

Dropping the censored observations and redoing the analysis will not help. Consider an extreme case of a dataset with just one censored observation and assume the observation is censored at time 0.1, long before the first failure. For all you know, had that subject not been censored, the failure might have occurred long after the last failure in the data and thus had a large effect on the mean. Wherever the censored observation is located in the data, we can repeat that argument, and so, in the presence of censoring, obtaining estimates of the mean survival time calculated in the standard way is simply not possible.

Estimates of the median survival time are similarly not possible to obtain using standard nonsurvival tools. The standard way of calculating the median is to order the observations and to report the middle one as the median. In the presence of censoring, that ordering is impossible to ascertain. (The modern way of calculating the median is to turn to the calculation of survival probabilities and find the point at which the survival probability is 0.5. See section 8.5.)

Thus even the most simple analysis—never mind the more complicated regression models—will break down when applied to survival data. Also there are even more issues related to survival data—truncation, for example—that would only further complicate the estimation.

Instead, survival analysis is a field of its own. Given the nature of the role that time plays in the analysis, much focus is given to the functions that characterize the distribution of the survival time: the hazard function, the cumulative hazard function, and the survivor function being the most common ways to describe the distribution. Much of survival analysis is concerned with the estimation of and inference for these functions of time.

8.2 The Kaplan–Meier estimator

8.2.1 Calculation

The estimator of Kaplan and Meier (1958) is a nonparametric estimate of the survivor function $S(t)$, which is the probability of survival past time t or, equivalently, the probability of failing after t. For a dataset with observed failure times, $t_1, \dots, t_k$, where k is the number of distinct failure times observed in the data, the Kaplan–Meier estimate [also known as the *product limit* estimate of $S(t)$] at any time t is given by

$$\widehat{S}(t) = \prod_{j|t_j \le t} \left(\frac{n_j - d_j}{n_j} \right) \tag{8.1}$$

where n_j is the number of individuals at risk at time t_j and d_j is the number of failures at time t_j. The product is over all observed failure times less than or equal to t.

How does this estimator work? Consider the hypothetical dataset of subjects given in the usual format,

```
id    t    failed
 1    2         1
 2    4         1
 3    4         1
 4    5         0
 5    7         1
 6    8         0
```

and form a table that summarizes what happens at each time in our data (whether a failure time or a censored time):

t	No. at risk	No. failed	No. censored
2	6	1	0
4	5	2	0
5	3	0	1
7	2	1	0
8	1	0	1

At $t = 2$, the earliest time in our data, all six subjects were at risk, but at that instant, only one failed (`id==1`). At the next time, $t = 4$, five subjects were at risk, but at that instant, two failed. At $t = 5$, three subjects were left, and no one failed, but one subject was censored. This left us with two subjects at $t = 7$, of which one failed. Finally, at $t = 8$, we had one subject left at risk, and this subject was censored at that time.

Now we ask the following:

- What is the probability of survival beyond $t = 2$, the earliest time in our data? Because five of the six subjects survived beyond this point, the estimate is 5/6.
- What is the probability of survival beyond $t = 4$ given survival right up to $t = 4$? Because we had five subjects at risk at $t = 4$, and two failed, we estimate this probability to be 3/5.
- What is the probability of survival beyond $t = 5$ given survival right up to $t = 5$? Because three subjects were at risk, and no one failed, the probability estimate is $3/3 = 1$.

and so on. We can now augment our table with these component probabilities (calling them p):

t	No. at risk	No. failed	No. censored	p
2	6	1	0	5/6
4	5	2	0	3/5
5	3	0	1	1
7	2	1	0	1/2
8	1	0	1	1

- The first value of p, 5/6, is the probability of survival beyond $t = 2$.
- The second value, 3/5, is the (conditional) probability of survival beyond $t = 4$ given survival up until $t = 4$, which in these data is the same as survival beyond $t = 4$ given survival beyond $t = 2$. Thus unconditionally, the probability of survival beyond $t = 4$ is $(5/6)(3/5) = 1/2$.

- The third value, 1, is the conditional probability of survival beyond $t = 5$ given survival up until $t = 5$, which in these data is the same as survival beyond $t = 5$ given survival beyond $t = 4$. Unconditionally, the probability of survival beyond $t = 5$ is thus equal to $(1/2)(1) = 1/2$.

Thus the Kaplan–Meier estimate is the running product of the values of p that we have previously calculated, and we can add it to our table.

t	No. at risk	No. failed	No. censored	p	$\widehat{S}(t)$
2	6	1	0	5/6	5/6
4	5	2	0	3/5	1/2
5	3	0	1	1	1/2
7	2	1	0	1/2	1/4
8	1	0	1	1	1/4

Because the Kaplan–Meier estimate in (8.1) operates only on observed failure times (and not at censoring times), the net effect is simply to ignore the cases where $p = 1$ in calculating our product; ignoring these changes nothing.

In Stata, the Kaplan–Meier estimate is obtained using the `sts list` command, which gives a table similar to the one we constructed:

```
. clear
. input id time failed
            id       time     failed
  1. 1 2 1
  2. 2 4 1
  3. 3 4 1
  4. 4 5 0
  5. 5 7 1
  6. 6 8 0
  7. end
. stset time, fail(failed)
  (output omitted)
. sts list
        failure _d:  failed
   analysis time _t:  time
           Beg.          Net            Survivor      Std.
  Time    Total   Fail   Lost           Function     Error      [95% Conf. Int.]
-------------------------------------------------------------------------------
     2        6      1      0             0.8333    0.1521     0.2731    0.9747
     4        5      2      0             0.5000    0.2041     0.1109    0.8037
     5        3      0      1             0.5000    0.2041     0.1109    0.8037
     7        2      1      0             0.2500    0.2041     0.0123    0.6459
     8        1      0      1             0.2500    0.2041     0.0123    0.6459
-------------------------------------------------------------------------------
```

The column "Beg. Total" is what we called "No. at risk" in our table; the column "Fail" is "No. failed"; and the column "Net lost" is related to our "No. censored" column but is modified to handle delayed entry (see sec. 8.2.3).

The standard error reported for the Kaplan–Meier estimate is that given by Greenwood's (1926) formula:

$$\widehat{\text{Var}}\{\widehat{S}(t)\} = \widehat{S}^2(t) \sum_{j|t_j \leq t} \frac{d_j}{n_j(n_j - d_j)} \tag{8.2}$$

These standard errors, however, are not used for confidence intervals. Instead, the asymptotic variance of $\ln\{-\ln \widehat{S}(t)\}$,

$$\widehat{\sigma}^2(t) = \frac{\sum \frac{d_j}{n_j(n_j-d_j)}}{\left\{\sum \ln\left(\frac{n_j-d_j}{d_j}\right)\right\}^2}$$

is used, where the sums are calculated over j such that $t_j \leq t$ (Kalbfleisch and Prentice 2002, 18). The confidence bounds are then calculated as $\widehat{S}(t)$ raised to the power $\exp\{\pm z_{\alpha/2}\widehat{\sigma}(t)\}$, where $z_{\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution.

8.2.2 Censoring

When censoring occurs at some time other than an observed failure time, for a different subject the effect is simply that the censored subjects are dropped from the "No. at risk" total without processing the censored subject as having failed. However, when some subjects are censored at the same time that others fail, we need to be a bit careful about how we order the censorings and failures. When we went through the calculations of the Kaplan–Meier estimate in section 8.2.1, we did so without explaining this point, yet be assured that we were following some convention.

The Stata convention for handling a censoring that happens at the same time as a failure is to assume that the failure occurred before the censoring, and in fact, all Stata's `st` commands follow this rule. In chapter 7, we defined a time span based on the `stset` variables `_t0` and `_t` to be the interval $(t_0, t]$, which is open at the left endpoint and closed at the right endpoint. Therefore, if we apply this definition of a time span, then any record shown to be censored at the end of this span can be thought of as instead being censored at some time $t+\epsilon$ for an arbitrarily small ϵ. The subject can fail at time t, but if the subject is censored, then Stata assumes that the censoring took place just a little bit later; thus failures occur before censorings.

This is how Stata handles this issue, but there is nothing wrong with the convention that handles censorings as occurring before failures when they appear to happen concurrently. One can force Stata to look at things this way by subtracting a small number from the time variable in your data for those records that are censored, and most of the time the number may be chosen small enough as to not otherwise affect the analysis.

❑ Technical note

If you force Stata to treat censorings as occurring before failures, be sure to modify the time variable in your data and not the `_t` variable that `stset` has created. In general, manually changing the values of the `stset` variables `_t0`, `_t`, `_d`, and `_st` is dangerous because these variables have relations to your variables, and some of the data-management `st` commands exploit that relationship.

Thus instead of using a command such as

```
. replace _t = _t - 0.0001 if _d == 0
```

use

```
. replace time = time - 0.0001 if failed == 0
. stset time, failure(failed)
```

Better yet, use

```
. replace time = time - 0.0001 if failed == 0
. stset
```

because `stset` will remember the details of how you previously set your data and will apply these same settings to the modified data.

❑

8.2.3 Left-truncation (delayed entry)

Left-truncation refers to subjects who do not come under observation until after they are at risk. By the time you begin observing this subject, they have already survived for some time, and you are observing them only because they did not fail during that time.

At one level, such observations cause no problems with the Kaplan–Meier calculation. In (8.1), n_j is the number of subjects at risk (eligible to fail), and this number needs to take into account that subjects are not at risk of failing until they come under observation. When they enter, we simply increase n_j to reflect this fact.

For example, if you have the following data (subject 6 enters at $t_0 = 4$ and is censored at $t = 7$),

```
id    t0    t1   failed
 1     0     2        1
 2     0     4        1
 3     0     4        1
 4     0     5        0
 5     0     7        1
 6     4     7        0
 7     0     8        0
```

then the risk-group table is

t	No. at risk	No. failed	No. censored	No. added
2	6	1	0	0
4	5	2	0	1
5	4	0	1	0
7	3	1	1	0
8	1	0	1	0

and now it is just a matter of making the Kaplan–Meier calculations based on how many are in the "No. at risk" and "No. failed" columns. We will let Stata do the work:

```
. clear
. input id time0 time1 failed
            id      time0      time1     failed
  1.  1      0     2        1
  2.  2      0     4        1
  3.  3      0     4        1
  4.  4      0     5        0
  5.  5      0     7        1
  6.  6      4     7        0
  7.  7      0     8        0
  8.  end
. stset time1, fail(failed) time0(time0)
  (output omitted)
. sts list
       failure _d:  failed
 analysis time _t:  time1
           Beg.          Net            Survivor      Std.
  Time    Total   Fail   Lost           Function     Error     [95% Conf. Int.]
-------------------------------------------------------------------------------
     2        6      1      0             0.8333    0.1521     0.2731    0.9747
     4        5      2     -1             0.5000    0.2041     0.1109    0.8037
     5        4      0      1             0.5000    0.2041     0.1109    0.8037
     7        3      1      1             0.3333    0.1925     0.0461    0.6756
     8        1      0      1             0.3333    0.1925     0.0461    0.6756
-------------------------------------------------------------------------------
```

Notice how Stata listed the delayed entry at $t = 4$: "Net Lost" is -1. To conserve columns, rather than listing censorings and entries separately, Stata combines them into one column containing censorings-minus-entries and labels that column as "Net Lost".

There is a level at which delayed entries cause considerable problems. In these entries' presence, the Kaplan–Meier procedure for calculating the survivor curve can yield absurd results. This happens when some late arrivals enter the study after everyone before them has failed.

Consider the following output from `sts list` for such a dataset:

```
. sts list

        failure _d:  failed
  analysis time _t:  time1

           Beg.          Net            Survivor      Std.
  Time    Total   Fail   Lost           Function     Error     [95% Conf. Int.]
-------------------------------------------------------------------------------
     2        6      1      0             0.8333    0.1521     0.2731    0.9747
     4        5      2     -1             0.5000    0.2041     0.1109    0.8037
     5        4      0      1             0.5000    0.2041     0.1109    0.8037
     7        3      1      1             0.3333    0.1925     0.0461    0.6756
     8        1      1      0             0.0000         .          .         .
     9        0      0     -3             0.0000         .          .         .
    10        3      1      0             0.0000         .          .         .
    11        2      1      1             0.0000         .          .         .
-------------------------------------------------------------------------------
```

We constructed these data to include three more subjects to enter at $t = 9$, after everyone who was previously at risk had failed. At $t = 8$, $\widehat{S}(t)$ has reached zero, never to return. Why does this happen? Note the product form of (8.1). Once a product term of zero (which occurs at $t = 8$) has been introduced, the product is zero, and further multiplication by anything nonzero is pointless. This is a shortcoming of the Kaplan–Meier method, and in section 8.3 we show that there is an alternative.

❑ **Technical note**

There is one other issue about the Kaplan–Meier estimator regarding delayed entry. When the earliest entry into the study occurs after $t = 0$, one may still calculate the Kaplan–Meier estimation, but the interpretation changes. Rather than estimating $S(t)$, you are now estimating $S(t|t_{\min})$, the probability of surviving past time t given survival to time $t_{\min}$, where $t_{\min}$ is the earliest entry time.

❑

8.2.4 Interval-truncation (gaps)

Interval-truncation is really no different from censoring followed by delayed entry. The subject disappears from the risk groups for a while and then reenters. The only issue is making sure that our "No. at risk" calculations reflect this fact, but Stata is up to that.

As with delayed entry, if a subject with a gap reenters after a final failure—meaning that a prior Kaplan–Meier estimate of $S(t)$ is zero—then all subsequent estimates of $S(t)$ will also be zero regardless of future activity.

8.2.5 Relationship to the empirical distribution function

The cumulative distribution function is defined as $F(t) = 1 - S(t)$, and in fact, by specifying the `failure` option, you can ask `sts list` to list the estimate of $F(t)$, which is obtained as 1 minus the Kaplan–Meier estimate:

```
. clear
. input id time0 time1 failed
             id      time0      time1     failed
  1.  1      0     2        1
  2.  2      0     4        1
  3.  3      0     4        1
  4.  4      0     5        0
  5.  5      0     7        1
  6.  6      4     7        0
  7.  7      0     8        0
  8. end
. stset time1, fail(failed) time0(time0)
  (output omitted)
. sts list, failure
         failure _d:  failed
   analysis time _t:  time1
           Beg.          Net            Failure      Std.
  Time    Total   Fail   Lost           Function     Error     [95% Conf. Int.]
-------------------------------------------------------------------------------
     2        6      1      0             0.1667    0.1521     0.0253    0.7269
     4        5      2     -1             0.5000    0.2041     0.1963    0.8891
     5        4      0      1             0.5000    0.2041     0.1963    0.8891
     7        3      1      1             0.6667    0.1925     0.3244    0.9539
     8        1      0      1             0.6667    0.1925     0.3244    0.9539
-------------------------------------------------------------------------------
```

For standard nonsurvival datasets, the *empirical distribution function* (edf) is defined to be

$$\widehat{F}_{\mathrm{edf}}(t) = \sum_{j|t_j \le t} n^{-1}$$

where we have $j = 1, \dots, n$ observations. That is, $\widehat{F}_{\mathrm{edf}}(t)$ is a step function that increases by $1/n$ at each observation in the data. Of course, $\widehat{F}_{\mathrm{edf}}(t)$ has no mechanism to account for censoring, truncation, and gaps, but when none of these exist, it can be shown that

$$\widehat{S}(t) = 1 - \widehat{F}_{\mathrm{edf}}(t)$$

where $\widehat{S}(t)$ is the Kaplan–Meier estimate. To demonstrate, consider the following simple dataset, which has no censoring or truncation:

```
. clear
. input t
             t
  1. 1
  2. 4
  3. 4
  4. 5
  5. end
. stset t
  (output omitted)
```

```
. sts list, failure
        failure _d:  1 (meaning all fail)
  analysis time _t:  t
           Beg.          Net            Failure      Std.
  Time    Total   Fail   Lost           Function     Error     [95% Conf. Int.]
-------------------------------------------------------------------------------
     1        4      1      0             0.2500    0.2165     0.0395    0.8721
     4        3      2      0             0.7500    0.2165     0.3347    0.9911
     5        1      1      0             1.0000         .          .         .
-------------------------------------------------------------------------------
```

This reproduces $\widehat{F}_{\text{edf}}(t)$, which is a nice property of the Kaplan–Meier estimator. Despite its sophistication in dealing with the complexities caused by censoring and truncation, it reduces to the standard methodology when these complexities do not exist.

8.2.6 Other uses of sts list

The `sts list` command lists the Kaplan–Meier survivor function. Let us use our hip-fracture dataset (the version we already `stset`):

```
. use http://www.stata-press.com/data/cggm3/hip2, clear
(hip fracture study)
. sts list
        failure _d:  fracture
  analysis time _t:  time1
                id:  id
           Beg.          Net            Survivor     Std.
  Time    Total   Fail   Lost           Function     Error     [95% Conf. Int.]
-------------------------------------------------------------------------------
     1       48      2      0             0.9583    0.0288     0.8435    0.9894
     2       46      1      0             0.9375    0.0349     0.8186    0.9794
     3       45      1      0             0.9167    0.0399     0.7930    0.9679
     4       44      2      0             0.8750    0.0477     0.7427    0.9418
  (output omitted)
    13       21      1      0             0.5384    0.0774     0.3767    0.6752
    15       20      1     -2             0.5114    0.0781     0.3507    0.6511
    16       21      1      0             0.4871    0.0781     0.3285    0.6283
  (output omitted)
    35        2      0      1             0.1822    0.0760     0.0638    0.3487
    39        1      0      1             0.1822    0.0760     0.0638    0.3487
-------------------------------------------------------------------------------
```

`sts list` can also produce less-detailed output. For instance, we can ask to see five equally spaced survival times in our data by specifying the `at()` option:

(*Continued on next page*)