Title

stata.com

stsum — Summarize survival-time data

Description Quick start Menu Syntax
Options Remarks and examples Stored results Methods and formulas
Also see

Description

`stsum` presents summary statistics: time at risk; incidence rate; number of subjects; and the 25th, 50th, and 75th percentiles of survival time.

`stsum` can be used with single- or multiple-record or single- or multiple-failure st data.

Quick start

Time at risk, incidence rate, number of subjects, and quartiles of survival time for `stset` data

```
stsum
```

As above, but only report statistics for observations with `v1` $= 1$

```
stsum if v1==1
```

Report separate summary statistics for each level of `v1`

```
stsum, by(v1)
```

Menu

Statistics > Survival analysis > Summary statistics, tests, and tables > Summarize survival-time data

Syntax

```
stsum [if] [in] [, by(varlist) noshow]
```

You must `stset` your data before using `stsum`; see [ST] **stset**.

`by` is allowed; see [D] **by**.

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see [ST] **stset**.

Options

Main

`by(`*varlist*`)` requests separate summaries for each group along with an overall total. Observations are in the same group if they have equal values of the variables in *varlist*. *varlist* may contain any number of string or numeric variables.

`noshow` prevents `stsum` from showing the key st variables. This option is seldom used because most people type `stset, show` or `stset, noshow` to set whether they want to see these variables mentioned at the top of the output of every st command; see [ST] **stset**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Single-failure data
Multiple-failure data
Video example

Single-failure data

Here is an example of `stsum` with single-record survival data:

```
. use http://www.stata-press.com/data/r14/page2
. stset, noshow
. stsum
```

	time at risk	incidence rate	no. of subjects	Survival time 25%	50%	75%
total	9118	.0039482	40	198	232	261

```
. stsum, by(group)
```

group	time at risk	incidence rate	no. of subjects	Survival time 25%	50%	75%
1	4095	.0041514	19	190	216	234
2	5023	.0037826	21	232	233	280
total	9118	.0039482	40	198	232	261

`stsum` works equally well with multiple-record survival data. Here is a summary of the multiple-record Stanford heart transplant data introduced in [ST] **stset**:

```
. use http://www.stata-press.com/data/r14/stan3
(Heart transplant data)
. stsum
        failure _d:  died
  analysis time _t:  t1
                id:  id
```

	time at risk	incidence rate	no. of subjects	Survival time 25%	50%	75%
total	31938.1	.0023483	103	36	100	979

`stsum` with the `by()` option may produce results with multiple-record data that, at first, you may think are in error.

```
. stsum, by(posttran) noshow
```

posttran	time at risk	incidence rate	no. of subjects	Survival time 25%	50%	75%
0	5936	.0050539	103	36	149	340
1	26002.1	.0017306	69	39	96	979
total	31938.1	.0023483	103	36	100	979

For the time at risk, $5{,}936 + 26{,}002.1 = 31{,}938.1$, but, for the number of subjects, $103 + 69 \neq 103$. The `posttran` variable is not constant for the subjects in this dataset:

```
. stset, noshow
. stvary posttran
```

	subjects for whom the variable is				
variable	constant	varying	never missing	always missing	sometimes missing
posttran	34	69	103	0	0

In this dataset, subjects have one or two records. All subjects were eligible for heart transplantation. They have one record if they die or are lost because of censoring before transplantation, and they have two records if the operation was performed. Then the first record records their survival up to transplantation and the second records their subsequent survival. `posttran` is 0 in the first record and 1 in the second.

Thus all 103 subjects have records with `posttran` $= 0$, and when `stsum` reported results for this group, it summarized the pretransplantation survival. The incidence of death was 0.005, and median survival time was 149 days.

The `posttran` $= 1$ line of `stsum`'s output summarizes the posttransplantation survival: 69 patients underwent transplantation, incidence of death was 0.002, and median survival time was 96 days. For these data, this is not 96 more days, but 96 days in total. That is, the clock was not reset at transplantation. Thus, without attributing cause, we can describe the differences between the groups as an increased hazard of death at early times followed by a decreased hazard later.

Multiple-failure data

If you simply type `stsum` with multiple-failure data, the reported survival time is the survival time to the first failure, assuming that the hazard function is not indexed by number of failures.

Here we have some multiple-failure data:

```
. use http://www.stata-press.com/data/r14/mfail2
. st
-> stset t, id(id) failure(d) time0(t0) exit(time .) noshow
                id:  id
     failure event:  d != 0 & d < .
obs. time interval:  (t0, t]
 exit on or before:  time .
. stsum
```

	incidence	no. of	Survival time		
	time at risk	rate	subjects	25%	50%	75%
total	435444	.0018556	926	201	420	703

To understand this output, let's also obtain output for each failure separately:

```
. stgen nf = nfailures()
. stsum, by(nf)
```

	incidence	no. of	Survival time		
nf	time at risk	rate	subjects	25%	50%	75%
0	263746	.0020057	926	196	399	604
1	121890	.0018131	529	252	503	816
2	38807	.0014946	221	415	687	.
3	11001	0	58	.	.	.
total	435444	.0018556	926	201	420	703

The `stgen` command added, for each subject, a variable containing the number of previous failures. For a subject, up to and including the first failure, `nf` is 0. Then `nf` is 1 up to and including the second failure, and then it is 2, and so on; see [ST] **stgen**.

The first line of the output, corresponding to `nf` $= 0$, states that among those who had experienced no failures yet, the incidence rate for (first) failure is 0.0020. The distribution of the time to the first failure is as shown.

Similarly, the second line, corresponding to `nf` $= 1$, is for those who have already experienced one failure. The incidence rate for (second) failures is 0.0018, and the distribution of time of (second) failures is as shown.

When we simply typed `stsum`, we obtained the same information shown as the total line of the more detailed output. The total incidence rate is easy to interpret, but what is the "total" survival-time distribution? It is an estimate of the distribution of the time to the first failure assuming that the hazard function $h(t)$ is the same across failures—that the second failure is no different from the first failure. This is an odd definition of "same" because the clock, t, is not reset in $h(t)$. What is the hazard of a failure—any failure—at time t? The answer is $h(t)$.

Another definition of "same" would have it that the hazard of a failure is given by $h(\tau)$, where τ is the time since last failure—that the process repeats. These definitions are different unless $h()$ is a constant function of t (τ).

So let's examine these multiple-failure data under the process-replication idea. The key variables in these st data are `id`, `t0`, `t`, and `d`:

```
. st
-> stset t, id(id) failure(d) time0(t0) exit(time .) noshow
                id:  id
     failure event:  d != 0 & d < .
obs. time interval:  (t0, t]
 exit on or before:  time .
```

Our goal is, for each subject, to reset `t0` and `t` to 0 after every failure event. We are going to have to trick Stata, or at least trick `stset`, because it will not let us set data where the same subject has multiple records summarizing the overlapping periods. So, the trick is to create a new `id` variable that is different for every ID−`nf` combination (remember, `nf` is the variable we previously created that records the number of prior failures). Then all the "new" subjects can have their clocks start at time 0:

```
. egen newid = group(id nf)
. sort newid t
. by newid: replace t = t - t0[1]
(808 real changes made)
. by newid: generate newt0 = t0 - t0[1]
. stset t, failure(d) id(newid) time0(newt0)
                id:  newid
     failure event:  d != 0 & d < .
obs. time interval:  (newt0, t]
 exit on or before:  failure
------------------------------------------------------------------------------
      1734  total observations
         0  exclusions
------------------------------------------------------------------------------
      1734  observations remaining, representing
      1734  subjects
       808  failures in single-failure-per-subject data
    435444  total analysis time at risk and under observation
                                        at risk from t =         0
                              earliest observed entry t =         0
                                   last observed exit t =       797
```

`stset` no longer thinks that we have multiple-failure data. Whereas with `id`, subjects had multiple failures, `newid` gives a unique identity to each ID−`nf` combination. Each "new" subject has at most one failure.

```
. stsum, by(nf)
        failure _d:  d
  analysis time _t:  t
                id:  newid
```

nf	time at risk	incidence rate	no. of subjects	Survival time 25%	50%	75%
0	263746	.0020057	926	196	399	604
1	121890	.0018131	529	194	384	580
2	38807	.0014946	221	210	444	562
3	11001	0	58	.	.	.
total	435444	.0018556	1734	201	404	602

Compare this table with the one we previously obtained. The incidence rates are the same, but the survival times differ because now we measure the times from one failure to the next. Previously we

measured the time from a fixed point. The time between events in these data appears to be independent of event number.

❑ Technical note

The method shown for converting multiple-failure data to replicated-process single-event failure data is completely general. The generic outline of the conversion process is

```
. stgen nf = nfailures()
. egen newid = group(id nf)
. sort newid t
. by newid: replace t = t - t0[1]
. by newid: generate newt0 = t0 - t0[1]
. stset t, failure(d) id(newid) t0(newt0)
```

where *id*, *t*, *t0*, and *d* are the names of your key survival-time variables.

Once you have done this to your data, you need exercise only one caution. If, in fitting models with `stcox`, `streg`, etc., you wish to obtain robust estimates of variance, you should include the `vce(cluster` *id*`)` option.

When you specify the `vce(robust)` option, `stcox`, `streg`, etc., assume that you mean `vce(cluster` *stset_id_variable*`)`, which, here, will be `vce(cluster newid)`. The data, however, are really more clustered than that. Two "subjects" with different `newid` values may, in fact, be the same real subject. `vce(cluster` *id*`)` is what is appropriate.

❑

Video example

How to describe and summarize survival data

Stored results

`stsum` stores the following in `r()`:

Scalars

`r(p25)`	25th percentile	`r(risk)`	time at risk
`r(p50)`	50th percentile	`r(ir)`	incidence rate
`r(p75)`	75th percentile	`r(N_sub)`	number of subjects

Methods and formulas

The 25th, 50th, and 75th percentiles of survival times are obtained from $S(t)$, the Kaplan–Meier product-limit estimate of the survivor function. The 25th percentile, for instance, is obtained as the minimum value of t such that $S(t) \leq 0.75$.

Also see

[ST] **stci** — Confidence intervals for means and percentiles of survival time

[ST] **stdescribe** — Describe survival-time data

[ST] **stgen** — Generate variables reflecting entire histories

[ST] **stir** — Report incidence-rate comparison

[ST] **stptime** — Calculate person-time, incidence rates, and SMR

[ST] **sts** — Generate, graph, list, and test the survivor and cumulative hazard functions

[ST] **stset** — Declare data to be survival-time data

[ST] **stvary** — Report variables that vary over time