stbase — Form baseline dataset

Description

stbase without the at() option converts multiple-record st data to st data with every variable set to its value at baseline, defined as the earliest time at which each subject was observed. stbase without at() does nothing to single-record st data.

stbase, at() converts single- or multiple-record st data to a cross-sectional dataset (not st data), recording the number of failures at the specified time. All variables are given their values at baseline—the earliest time at which each subject was observed. In this form, single-failure data could be analyzed by logistic regression and multiple-failure data by Poisson regression, for instance.

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

Quick start

Set all variables to their values at the earliest time the subject was observed using stset data

stbase

Create a dataset with one observation per subject, recording number of failures at time 10, with all variables set to the value at the earliest time the subject was observed

stbase, at(10)

Menu

Statistics > Survival analysis > Setup and utilities > Form baseline dataset
Syntax

```
stbase [if] [in] [ , options ]
```

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>at(#)</td>
<td>convert single/multiple-record st data to cross-sectional dataset at time #</td>
</tr>
<tr>
<td>gap(newvar)</td>
<td>name of variable containing gap time; default is gap or gaptime</td>
</tr>
<tr>
<td>replace</td>
<td>overwrite current data in memory</td>
</tr>
<tr>
<td>noshow</td>
<td>do not show st setting information</td>
</tr>
<tr>
<td>nopreserve</td>
<td>programmer’s option; see Options below</td>
</tr>
</tbody>
</table>

You must `stset` your data before using `stbase`; see [ST] `stset`. `fweight`, `iweight`, and `pweight` may be specified using `stset`; see [ST] `stset`. `nopreserve` does not appear in the dialog box.

Options

<table>
<thead>
<tr>
<th>Main</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>at(#)</td>
<td>changes what <code>stbase</code> does. Without the at() option, <code>stbase</code> produces another related st dataset. With the at() option, <code>stbase</code> produces a related cross-sectional dataset.</td>
</tr>
<tr>
<td>gap(newvar)</td>
<td>is allowed only with at(); it specifies the name of a new variable to be added to the data containing the amount of time the subject was not at risk after entering and before # as specified in at(). If gap() is not specified, the new variable will be named gap or gaptime, depending on which name does not already exist in the data.</td>
</tr>
<tr>
<td>replace</td>
<td>specifies that it is okay to change the data in memory, even though the dataset has not been saved to disk in its current form.</td>
</tr>
<tr>
<td>noshow</td>
<td>prevents <code>stbase</code> from showing the key st variables. This option is rarely used because most people type <code>stset</code>, <code>show</code> or <code>stset</code>, <code>noshow</code> to set once and for all whether they want to see these variables mentioned at the top of the output of every st command; see [ST] <code>stset</code>.</td>
</tr>
</tbody>
</table>

The following option is available with `stbase` but is not shown in the dialog box:

| nopreserve | is for use by programmers using `stbase` as a subroutine. It specifies that `stbase` not preserve the original dataset so that it can be restored should an error be detected or should the user press Break. Programmers would specify this option if, in their program, they had already preserved the original data. |

Remarks and examples

Remarks are presented under the following headings:

- `stbase without the at() option`
- `stbase with the at() option`
- Single-failure st data where all subjects enter at time 0
- Single-failure st data where some subjects enter after time 0
- Single-failure st data with gaps and perhaps delayed entry
- Multiple-failure st data
stbase without the at() option

Once you type `stbase`, you may not `streset` your data, even though the data are st. `streset` will refuse to run because the data have changed, and if the original rules were reapplied, they might produce different, incorrect results. The st commands use four key variables:

- `_t0` the time at which the record came under observation
- `_t` the time at which the record left observation
- `_d` 1 if the record left under failure, 0 otherwise
- `_st` whether the observation is to be used (contains 1 or 0)

These variables are adjusted by `stbase`. The `_t0` and `_t` variables, in particular, are derived from your variables according to options you specified at the time you `stset` the data, which might include an `origin()` rule, an `entry()` rule, and the like. Once intervening observations are eliminated, those rules will not necessarily produce the same results that they did previously.

To illustrate how `stbase` works, consider multiple-record, time-varying st data, on which you have performed some analysis. You now wish to compare your results with a simpler, non-time-varying analysis. For instance, suppose that variables `x1` and `x2` measure blood pressure and weight, respectively, and that readings were taken at various times. Perhaps you fit the model

```
use https://www.stata-press.com/data/r16/mfail
.stset t, id(id) failure(d) exit(time .) noshow
     id: id
failure event: d != 0 & d < .
obs. time interval: (t[_n-1], t]
exit on or before: time .
```

```
1,734  total observations
  0   exclusions
1,734  observations remaining, representing
  926  subjects
  808  failures in multiple-failure-per-subject data
435,855 total analysis time at risk and under observation
     at risk from t =  0
     earliest observed entry t =  0
     last observed exit t = 960
```
with these data. You now wish to fit that same model but this time use the values of \( x_1 \) and \( x_2 \) at baseline. You do this by typing

```
. stbase, replace
```

**notes:**
1. no gaps
2. there were multiple failures or reentries after failures
3. baseline data have multiple records per id(id)
4. all records have covariate values at baseline

```
. stcox x1 x2
```

**Iteration 0:** log likelihood = -7886.9779  
**Iteration 1:** log likelihood = -7863.9974  
**Iteration 2:** log likelihood = -7863.9295  
**Iteration 3:** log likelihood = -7863.9295  
**Refining estimates:**  
**Iteration 0:** log likelihood = -7863.9295

**Cox regression -- Breslow method for ties**  
No. of subjects = 926  
No. of failures = 1,337  
Time at risk = 435855  
LR chi2(2) = 46.10  
Log likelihood = -7863.9295  
Prob > chi2 = 0.0000

|   | Haz. Ratio | Std. Err. | z    | P>|z| | [95% Conf. Interval] |
|---|------------|-----------|------|------|----------------------|
| x1| 2.273456   | 0.216537  | 8.62 | 0.000| 1.886311 2.740059   |
| x2| 0.329011   | 0.0685638 | -5.33| 0.000| 0.2186883 0.4949888 |

Another way you could perform the analysis is to type

```
. generate x1_0 = x1
. generate x2_0 = x2
. stfill x1_0 x2_0, baseline
. stcox x1 x2
```

See [ST] *stfill*. The method you use makes no difference, but if there were many explanatory variables, *stbase* would be easier.
**stbase** changes the data to record the same events but changes the values of all other variables to their values at the earliest time the subject was observed.

**stbase** also simplifies the st data where possible. Say that one of your subjects has three records in the original data and ends in a failure:

```
|   |   |   | X |
```

After running **stbase**, this subject would have one record in the data:

```
|   |   |   |   | X |
```

Here are some other examples of how **stbase** would process records with gaps and multiple failure events:

```
|   |   |   |   |   |   |   | X |
```

```
|   |   |   |   |   | X |
```

```
X |   |   |   |   |   |   | X |
```

```
X |   |   |   |   |   |   | X |
```

The following example shows numerically what is shown in the diagram above.

```
. use https://www.stata-press.com/data/r16/stbasetmpl, clear
. list, sepby(id)
```

<table>
<thead>
<tr>
<th>id</th>
<th>time0</th>
<th>time</th>
<th>wgt</th>
<th>death</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>114</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>110</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>11</td>
<td>118</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>120</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>11</td>
<td>111</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>108</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>105</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>113</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>98</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>106</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>6</td>
<td>11</td>
<td>104</td>
</tr>
</tbody>
</table>
. stset time, id(id) fail(death) time0(time0) exit(time .)
  id: id
  failure event: death != 0 & death < .
  obs. time interval: (time0, time]
  exit on or before: time .

12 total observations
0 exclusions

12 observations remaining, representing
4 subjects
6 failures in multiple-failure-per-subject data
36 total analysis time at risk and under observation
  at risk from t = 0
  earliest observed entry t = 0
  last observed exit t = 11

. list, sepby(id)

<table>
<thead>
<tr>
<th>id</th>
<th>time0</th>
<th>time</th>
<th>wgt</th>
<th>death</th>
<th>_st</th>
<th>_d</th>
<th>_t</th>
<th>_t0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>114</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>11</td>
<td>118</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>120</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>11</td>
<td>111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>108</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>105</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>113</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>98</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>101</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>106</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>6</td>
<td>11</td>
<td>104</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

. stbase, replace
  failure _d: death
  analysis time _t: time
  exit on or before: time .
  id: id

notes:
1. there were gaps
2. there were multiple failures or reentries after failures
3. baseline data have multiple records per id(id)
4. all records have covariate values at baseline
### stbase with the at() option

`stbase`, `at()` produces a cross-sectional dataset recording the status of each subject at the specified time. This new dataset is not st. Four “new” variables are created:

- the first entry time for the subject,
- the time on gap,
- the time at risk, and
- the number of failures during the time at risk.

The names given to those variables depend on how your data are `stset`. Pretend that your `stset` command was

```
  . stset var1, failure(var2) time0(var3) ...
```

Then

- the first entry time will be named `var3` or `time0` or `_t0`
- the time on gap will be named `gap()` or `gap` or `gaptime`
- the time at risk will be named `var1`
- the number of (or whether) failures will be named `var2` or `failure` or `_d`

The names may vary because, for instance, if you did not specify a `var2` variable when you `stset` your data, `stbase, at()` looks around for a name.

You need not memorize this; the names are obvious from the output produced by `stbase, at()`.
Consider the actions of \texttt{stbase}, \texttt{at()} with some particular \texttt{st} datasets. Pretend that the command given is
\begin{verbatim}
    . use https://www.stata-press.com/data/r16/stbasexmpl2, clear
    . list, sepby(id)
    \end{verbatim}
\begin{verbatim}
<table>
<thead>
<tr>
<th>id</th>
<th>time0</th>
<th>time</th>
<th>wgt</th>
<th>death</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>114</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>110</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>8</td>
<td>11</td>
<td>118</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>111</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>108</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>98</td>
</tr>
</tbody>
</table>
\end{verbatim}
\begin{verbatim}
    . stset time, id(id) fail(death) time0(time0)
    id: id
    failure event: death != 0 & death < .
    obs. time interval: (time0, time]
    exit on or before: failure
    \end{verbatim}
\begin{verbatim}
7 total observations
0 exclusions
7 observations remaining, representing
2 subjects
2 failures in single-failure-per-subject data
21 total analysis time at risk and under observation
at risk from t = 0
earliest observed entry t = 0
last observed exit t = 11
\end{verbatim}
\begin{verbatim}
    . list, sepby(id)
    \end{verbatim}
\begin{verbatim}
<table>
<thead>
<tr>
<th>id</th>
<th>time0</th>
<th>time</th>
<th>wgt</th>
<th>death</th>
<th>_st</th>
<th>_d</th>
<th>_t</th>
<th>_t0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>114</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>110</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>8</td>
<td>11</td>
<td>118</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>120</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>111</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>108</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>98</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>
\end{verbatim}
. `stbase, at(5) replace`
  failure _d: death
  analysis time _t: time
  id: id

data now cross-section at time 5

<table>
<thead>
<tr>
<th>Variable</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>subject identifier</td>
</tr>
<tr>
<td>time0</td>
<td>first entry time</td>
</tr>
<tr>
<td>gap</td>
<td>time on gap</td>
</tr>
<tr>
<td>time</td>
<td>time at risk</td>
</tr>
<tr>
<td>death</td>
<td>number of failures during interval time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>time0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>gap</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>time</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>death</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

. `list`

<table>
<thead>
<tr>
<th>id</th>
<th>wgt</th>
<th>death</th>
<th>time</th>
<th>time0</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

thus producing a cross-section at analysis time 5.

Note that the value of time specified with the `at()` option must correspond to time in the analysis scale, that is, \( t \). See [ST] `stset` for a definition of analysis time.

**Single-failure st data where all subjects enter at time 0**

The result of `stbase, at(5)` would be one record per subject. Any subject who was censored before time 5 would not appear in the data; the rest would. Those that failed after time 5 will be recorded as having been censored at time 5 (`failvar = 0`); those that failed at time 5 or earlier will have `failvar = 1`.

`timevar` will contain

for the failures:
- time of failure if failed on or before time 5 or
- 5 if the subject has not failed yet

for the censored:
- 5 if the subject has not failed yet

With such data, you could perform

- logistic regression of `failvar` on any of the characteristics or
- incidence-rate analysis, summing the failures (perhaps within strata) and the time at risk, `timevar`.

With these data, you could examine 5-year survival probabilities.
Single-failure st data where some subjects enter after time 0

The data produced by \texttt{stbase}, \texttt{at(5)} would be similar to the above, except

- persons who enter on or after time 5 would not be included in the data (because they have not entered yet) and
- the time at risk, \texttt{timevar}, would properly account for the time at which each patient entered.

\texttt{timevar} (the time at risk) will contain

\begin{itemize}
  \item for the failures:
    \begin{itemize}
      \item time of failure if failed on or before time 5 (or less because the subject may not have entered at time 0); or
      \item 5 or less if the subject has not failed yet (or less because the subject may not have entered at time 0)
    \end{itemize}
  \item for the censored:
    \begin{itemize}
      \item 5 or less if the subject has not failed yet (or less because the subject may not have entered at time 0)
    \end{itemize}
\end{itemize}

Depending on the analysis you are performing, you may have to discard those that enter late. This is easy to do because \texttt{t0} contains the first time of entry.

With these data, you could perform the following:

- Logistic regression of \texttt{failvar} on any of the characteristics, but only if you restricted the sample to \texttt{if t0 == 0} because those who entered after time 0 have a lesser risk of failing over the fixed interval.
- Incidence-rate analysis, summing the failures (perhaps within stratum) and the time at risk, \texttt{timevar}. Here you would have to do nothing differently from what you did in the previous example. The time-at-risk variable already includes the time of entry for each patient.

Single-failure st data with gaps and perhaps delayed entry

These data will be similar to the delayed-entry, no-gap data, but \texttt{gap} will contain 0 only for those observations that have no gap.

If analyzing these data, you could perform

- logistic regression, but the sample must be restricted to \texttt{if t0 == 0 & gap == 0}, or
- incidence-rate analysis, and nothing would need to be done differently; the time at risk, \texttt{timevar}, accounts for late entry and gaps.

Multiple-failure st data

The multiple-failure case parallels the single-failure case, except that \texttt{fail} will not solely contain 0 and 1; it will contain 0, 1, 2, \ldots, depending on the number of failures observed. Regardless of late entry, gaps, etc., you could perform

- Poisson regression of \texttt{fail}, the number of events, but remember to specify \texttt{exposure(timevar)}, and
- incidence-rate analysis.
Also see

[ST] `stfill` — Fill in by carrying forward values of covariates

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