**Description**

cctost converts count-time data to their survival-time format so that they can be analyzed with Stata. Do not confuse count-time data with counting-process data, which can also be analyzed with the st commands; see [ST] cctset for a definition and examples of count data.

**Quick start**

Convert count-time data to survival-time data using `ctset` data

cctost

As above, but name the new weight variable `mywvar` instead of using the default name

cctost, wvar(mywvar)

**Menu**

Statistics > Survival analysis > Setup and utilities > Convert count-time data to survival-time data
## Syntax

```plaintext
cttost [, options]
```

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>t0(t0var)</code></td>
<td>name of entry-time variable</td>
</tr>
<tr>
<td><code>wvar(wvar)</code></td>
<td>name of frequency-weighted variable</td>
</tr>
<tr>
<td><code>clear</code></td>
<td>overwrite current data in memory</td>
</tr>
<tr>
<td><code>nopreserve</code></td>
<td>do not save the original data; programmer’s command</td>
</tr>
</tbody>
</table>

You must `ctset` your data before using `cttost`; see `[ST] ctset`. `nopreserve` does not appear in the dialog box.

## Options

- **`t0(t0var)`**: Specifies the name of the new variable to create that records entry time. (For most `ct` data, no entry-time variable is necessary because everyone enters at time 0.)

  Even if an entry-time variable is necessary, you need not specify this option. `cttost` will, by default, choose `t0`, `time0`, or `etime` according to which name does not already exist in the data.

- **`wvar(wvar)`**: Specifies the name of the new variable to be created that records the frequency weights for the new pseudo-observations. Count-time data are actually converted to frequency-weighted `st` data, and a variable is needed to record the weights. This sounds more complicated than it is. Understand that `cttost` needs a new variable name, which will become a permanent part of the `st` data.

  If you do not specify `wvar()`, `cttost` will, by default, choose `w`, `pop`, `weight`, or `wgt` according to which name does not already exist in the data.

- **`clear`**: Specifies that it is okay to proceed with the conversion, even though the current dataset has not been saved on disk.

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

- **`nopreserve`**: Speeds the conversion by not saving the original data that can be restored should things go wrong or should you press `Break`. `nopreserve` is intended for use by programmers who use `cttost` as a subroutine. Programmers can specify this option if they have already preserved the original data. `nopreserve` does not affect the conversion.

## Remarks and examples

Converting `ct` to `st` data is easy. We have some count-time data,

```plaintext
. use https://www.stata-press.com/data/r16/cttost
c
```

```plaintext
  dataset name: https://www.stata-press.com/data/r16/cttost.dta
time: time
no. fail: ndead
no. lost: ncens
no. enter: --                                  (meaning all enter at time 0)
  by: agecat treat
```

---

### stata.com

Converting `ct` to `st` data is easy. We have some count-time data,

```plaintext
. use https://www.stata-press.com/data/r16/cttost
c
```

```plaintext
  dataset name: https://www.stata-press.com/data/r16/cttost.dta
time: time
no. fail: ndead
no. lost: ncens
no. enter: --                                  (meaning all enter at time 0)
  by: agecat treat
```
and to convert it, we type `cttost`:

```
. cttost
  failure event:  ndead != 0 & ndead < .
  obs. time interval:  (0, time]
  exit on or before:  failure
  weight:  [fweight=w]
```

<table>
<thead>
<tr>
<th>agecat</th>
<th>treat</th>
<th>time</th>
<th>ndead</th>
<th>ncens</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>464</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>268</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>638</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>803</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>431</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

33 total observations
0 exclusions
33 physical observations remaining, equal to
82 weighted observations, representing
39 failures in single-record/single-failure data
48,726 total analysis time at risk and under observation
at risk from $t = 0$
earliest observed entry $t = 0$
last observed exit $t = 1,227$

Now that it is converted, we can use any of the `st` commands:

```
. sts test treat, logrank
  failure _d:  ndead
  analysis time _t:  time
  weight:  [fweight=w]
```

**Log-rank test for equality of survivor functions**

<table>
<thead>
<tr>
<th>treat</th>
<th>Events observed</th>
<th>Events expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>22</td>
<td>17.05</td>
</tr>
<tr>
<td>1</td>
<td>17</td>
<td>21.95</td>
</tr>
<tr>
<td>Total</td>
<td>39</td>
<td>39.00</td>
</tr>
</tbody>
</table>

$\chi^2(1) = 2.73$
$Pr>\chi^2 = 0.0986$

**Also see**

[ST] ct — Count-time data
[ST] ctset — Declare data to be count-time data