cttost — Convert count-time data to survival-time data

Description

`cttost` 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] **ctset** for a definition and examples of count data.

Quick start

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

```
cttost
```

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

```
cttost, wvar(mywvar)
```

Menu

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

Syntax

`cttost` [, *options*]

options	Description
`t0(`*t0var*`)`	name of entry-time variable
`wvar(`*wvar*`)`	name of frequency-weighted variable
`clear`	overwrite current data in memory
`nopreserve`	do not save the original data; programmer's command

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,

```
. use https://www.stata-press.com/data/r19/cttost
. ct
Count-time data settings
           Time: time
       Failures: ndead
    Number lost: ncens
 Number entered: All enter at time 0
Group variables: agecat treat
. list in 1/5
```

	agecat	treat	time	ndead	ncens
1.	2	1	464	4	0
2.	3	0	268	3	1
3.	2	0	638	2	0
4.	1	0	803	1	4
5.	1	0	431	2	0

and to convert it, we type cttost:

```
. cttost
Count-time data settings
           Time: time
       Failures: ndead
    Number lost: ncens
 Number entered: All enter at time 0
Group variables: agecat treat
Converting count-time data to survival-time data ...
Survival-time data settings
         Failure event: ndead!=0 & ndead<.
Observed time interval: (0, time]
     Exit on or before: failure
               Weight: [fweight=w]
--------------------------------------------------------------------------------
         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]
Equality of survivor functions
Log-rank test
       |  Observed       Expected
treat  |    events         events
-------+--------------------------
    0  |        22          17.05
    1  |        17          21.95
-------+--------------------------
Total  |        39          39.00
                chi2(1) =    2.73
                Pr>chi2 =  0.0986
```

Also see

[ST] **ct** — Count-time data

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