Discrete-time survival analysis with Stata

Isabel Canette

Principal Mathematician and Statistician

StataCorp LP

2016 Stata Users Group Meeting
Barcelona, October 20, 2016

Introduction

Survival analysis studies the time until an event happens. It's applied to a large array of disciplines like social sciences, natural sciences, engineering, medicine.

Discrete-data survival analysis refers to the case where data can only take values over a discrete grid, e.g. 1,2,3....

In some cases, discrete data are "truly discrete"; the event can only happen at discrete values of time (e.g., length of time that a party remains is the government; change can only happen at the end of one term [1]).

In many cases, discrete data are the result of interval-censoring. Events might happen in a continuous range of time, but they can only be observed at discrete moments (e.g., "silent" heart-attacks can be observed when patient visits the doctor), or are recorded on discrete units (length of stay in a hospital is recorded in days).

[1] Allison,P. Discrete-Time Methods for the Analysis of Event Histories; Sociological Methodology, Vol. 13, (1982), pp. 61-98

Outline:

- Brief review of main concepts in survival analysis
- Methods to deal with interval-censored and discrete data
    - Method 1: using continuous methods for interval-censored data
    - Method 2: using commands written specifically for interval-censored data
    - Method 3: Estimate the discrete hazard
    - Using Method 3 for interval-censored data
    - Some extension to method 3

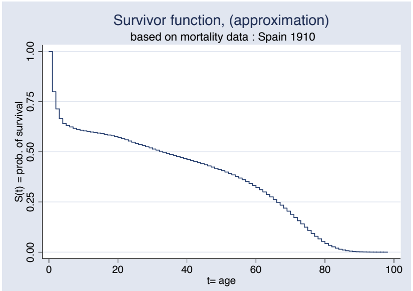
Specific challenges of survival analysis

Some specific challenges of survival analysis:

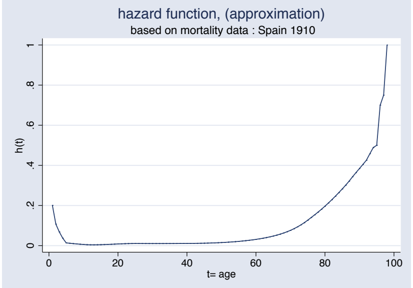
- Usually, the observed data can't be modeled by a Gaussian distribution; therefore, other distributions need to be used (e.g., in Stata, the `streg` command implements several distribution suited for survival data)
- Data are often right-censored (and sometimes left-truncated)
- Functions of interest are mainly the survivor function and the hazard function (not so much the density and the distribution)

The survivor and the hazard functions

In survival analysis, we are intersted in the survivor and the hazard function:

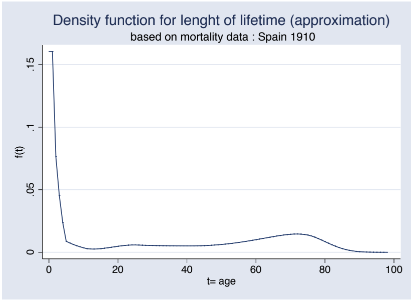


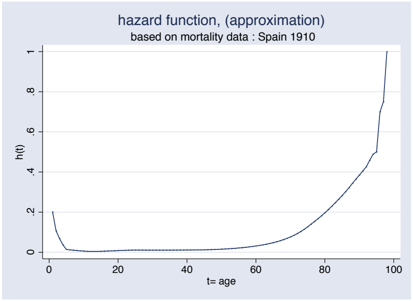
$S(t) = P(T > t) = 1 - F(t)$
e.g. what's the probability of surviving 20 years?



$h(t) = \frac{f(t)}{S(t)}$
(interpreted as "instant risk")

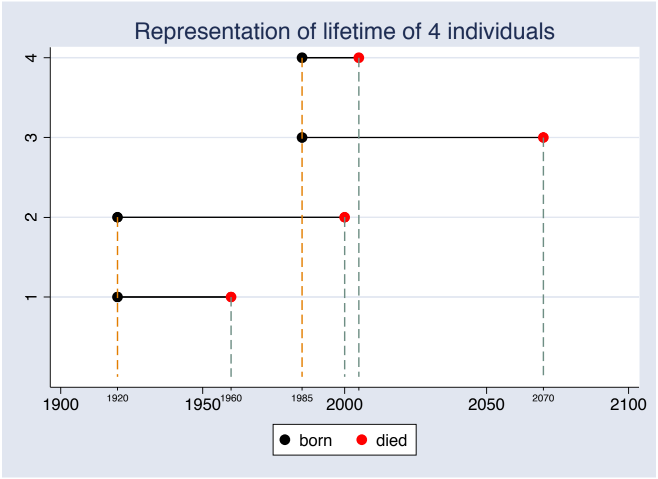
Density versus hazard



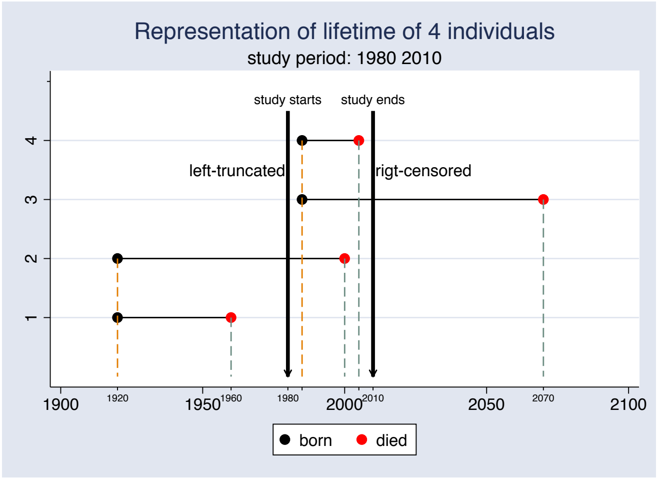


Right censoring, left truncation

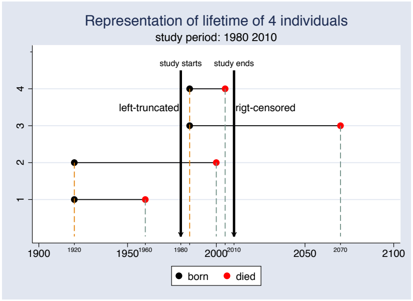
Assume we want to study the lifespan in a certain population; events would happen as follows:



However, we can only run a study for a certain amount of time. Many studies come from interviewing/following-up a sample of individuals (who are alive sometime during the study)
Let's assume that our study went from 1980 to 2010:



Our data would looks like follows:



```
. list id born study_starts enter last_time_obs died, abb(18)
```

```
     ┌──────────────────────────────────────────────────────────────────┐
     │ id   born   study_starts   enter   last_time_observed   died │
     ├──────────────────────────────────────────────────────────────────┤
  1. │  4   1985           1980    1985                 2005      1 │
  2. │  3   1985           1980    1985                 2010      0 │
  3. │  2   1920           1980    1980                 2000      1 │
     └──────────────────────────────────────────────────────────────────┘
```

We use stset to tell Stata about this information:

```
. stset last_time_obs, failure(died) origin(born) enter(enter)

     failure event:  died != 0 & died < .
obs. time interval:  (origin, last_time_observed]
 enter on or after:  time enter
 exit on or before:  failure
    t for analysis:  (time-origin)
            origin:  time born
------------------------------------------------------------------------------
         3  total observations
         0  exclusions
------------------------------------------------------------------------------
         3  observations remaining, representing
         2  failures in single-record/single-failure data
        65  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =        80
```

stset creates the “underscore” variables:

```
. list born enter last died _t0 _t _d _st

     ┌────────────────────────────────────────────────────────────┐
     │ born   enter   last_t~d   died   _t0   _t   _d   _st │
     ├────────────────────────────────────────────────────────────┤
  1. │ 1985    1985       2005      1     0   20    1     1 │
  2. │ 1985    1985       2010      0     0   25    0     1 │
  3. │ 1920    1980       2000      1    60   80    1     1 │
     └────────────────────────────────────────────────────────────┘
```

Variables _t0, _t, _d, _st are used for further estimations by st commands

For example, `streg` fits several parametric distributions. (Right-)censoring is handled as in `intreg` and `tobit`; and (left-)truncation is handled as in `truncreg`, using the specified distribution instead of the normal.
The syntax looks like follows:

```
.   streg [covariates], distribution(dist_name)
```

Notice that we don't include a dependent variable (this information is taken from underscore variables)

The Nurses' Health Study (NHS) [2] is a prospective study of 121,700 female nurses from 11. U.S. states. Participant were enrolled in 1976, and followed-up for 30 years.
Let's assume we have data for a similar study; we want to study time to death in a population, for individuals who are already 30 years old (and we follow-up during 30 years).

[2] http://www.nurseshealthstudy.org/

Bao et al. [3] used data from the NHS to study the association of nut consumption to mortality. We'll use this concept to create a very simplified dataset and model as an example, where we only have a `nuts` dummy covariate, that indicates nut consumption over a certain threshold.

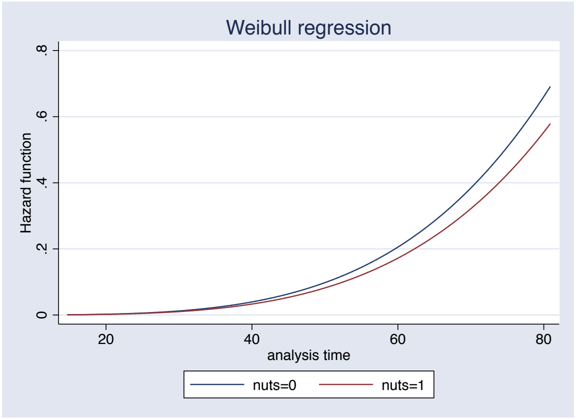
[3]Ying Bao, Jiali Han, Frank B. Hu, Edward L. Giovannucci, Meir J. Stampfer, Walter C. Willett, and Charles S. Fuchs. Association of Nut Consumption with Total and Cause-Specific Mortality N Engl J Med 2013; 369:2001-2011

We fit a Weibull model to our fictitious dataset: (after stset):

```
. streg i.nuts, di(weibull) nolog nohr

         failure _d:  1 (meaning all fail)
   analysis time _t:  t

Weibull regression -- log relative-hazard form

No. of subjects =         1,200                  Number of obs    =      1,200
No. of failures =         1,200
Time at risk    =   56495.17541
                                                 LR chi2(1)       =       9.43
Log likelihood  =     60.966853                  Prob > chi2      =     0.0021

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      1.nuts |  -.1777361   .0578383    -3.07   0.002    -.2910972    -.064375
       _cons |  -19.83802    .455061   -43.59   0.000    -20.72993   -18.94612
-------------+----------------------------------------------------------------
       /ln_p |   1.621853     .02235    72.57   0.000     1.578047    1.665658
-------------+----------------------------------------------------------------
           p |    5.06246   .1131462                      4.845485    5.289152
         1/p |   .1975324   .0044149                      .1890662    .2063777
------------------------------------------------------------------------------
```

The Weibull model implies the proportional-hazards assumption:
$h_{nuts=1}(t) = constant \times h_{nuts=0}(t)$ (and $constant = \exp(b_{1.nuts})$)
We can plot the predicted hazard curves with `stcurve`

```
. stcurve, hazard at1(nuts=0) at2(nuts=1)
```



The *constant* (*exp*(*b*)) is called “hazards ratio”, and it’s displayed by default by `streg, di(weibull)`

```
. streg i.nuts, di(weibull) nolog nohead

         failure _d:  1 (meaning all fail)
   analysis time _t:  t

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      1.nuts |   .8371633   .0484201    -3.07   0.002      .747443    .9376533
       _cons |   2.42e-09   1.10e-09   -43.59   0.000     9.93e-10    5.91e-09
-------------+----------------------------------------------------------------
       /ln_p |   1.621853     .02235    72.57   0.000     1.578047    1.665658
-------------+----------------------------------------------------------------
           p |    5.06246   .1131462                      4.845485    5.289152
         1/p |   .1975324   .0044149                      .1890662    .2063777
------------------------------------------------------------------------------
```

The hazard of dying at any given moment for somebody in group nuts=1 is equal to .84 times the hazard of dying for somebody in the group nuts = 0.

The Cox model makes the PH assumption without using any parametric form for the hazard (i.e., the hazard can have any shape).

```
. stcox i.nuts,  nolog nohead

         failure _d:  1 (meaning all fail)
   analysis time _t:  t

Cox regression -- no ties

No. of subjects =        1,200                  Number of obs    =      1,200
No. of failures =        1,200
Time at risk    =  56495.17541
                                                LR chi2(1)       =       9.85
Log likelihood  =   -7307.6324                  Prob > chi2      =     0.0017

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      1.nuts |   .8335557   .0483259    -3.14   0.002     .7440218    .933864
------------------------------------------------------------------------------
```

Interval-censored data

Let's assume that we have a discrete version of the previous dataset. We only have information from every year (or 2 years, or 5 years).

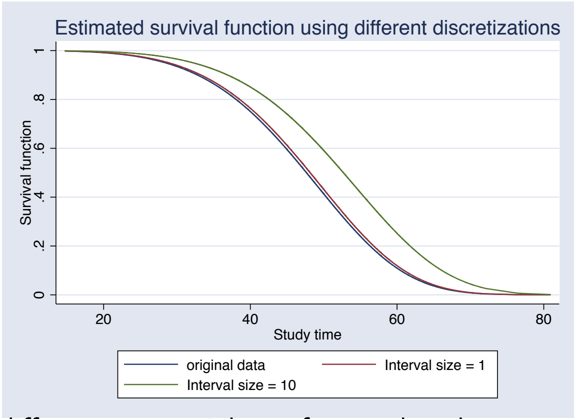
```
. use nuts_steps, clear

. list t t_1 t_5 in 1/10

     +---------------------------+
     |        t    t_1     t_5   |
     |---------------------------|
  1. | 58.50206     59      60   |
  2. | 58.85555     59      60   |
  3. | 48.10802     49      50   |
  4. | 45.56936     46      50   |
  5. | 41.07059     42      45   |
     |---------------------------|
  6. | 65.36206     66      70   |
  7. | 69.26743     70      70   |
  8. |  48.6137     49      50   |
  9. | 32.39676     33      35   |
 10. | 57.54965     58      60   |
     +---------------------------+
```

Method 1: treat the data as continuous

This is what we do most of the time, when we analyze "continuous" data (there is always some level of discretization)

```
. streg i.nuts, di(weibull) nolog

         failure _d:  1 (meaning all fail)
   analysis time _t:  t_one

Weibull regression -- log relative-hazard form

No. of subjects =        1,200                  Number of obs    =      1,200
No. of failures =        1,200
Time at risk    =        57085
                                                LR chi2(1)       =       9.89
Log likelihood  =    74.725844                  Prob > chi2      =     0.0017

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      1.nuts |   .8335685   .0482189    -3.15   0.002     .7442217    .9336416
       _cons |   1.85e-09   8.55e-10   -43.63   0.000     7.52e-10    4.58e-09
-------------+----------------------------------------------------------------
       /ln_p |   1.632823   .0223395    73.09   0.000     1.589039    1.676608
-------------+----------------------------------------------------------------
           p |   5.118306   .1143406                      4.899038    5.347388
         1/p |   .1953771   .0043646                      .1870072    .2041217
------------------------------------------------------------------------------
```

The following graph shows the predicted survival function obtained by using `streg` with the original data, and then with discretizations with grid size = 1 and 10. (predictions from a Weibull model without covariates)



For small differences, we might prefer to take advantage of the flexibility (and features availables) for this this approach. For larger differences, we might want to look for other approaches.

How do you know if the approximation is good enough?

You can generate artificial data for certain parameters and compare the estimates (`help statistical functions`)

You can perform a simulation to study coverage (`help simulate`)

Method 2: use a command specific for interval-censored data

These commands would use interval-censored data to estimate the underlying continuous survival function.

Currently, this can be done by the J. Griffin's (user-written) command `intcens`[4]

Also, you can fit a lognormal distribution by transforming the dependent variable and using `intreg`, for interval-censored Gaussian data.

This approach can be used for interval-censored data in general, i.e., intervals can be different for each individual, and there can be right-censoring.

[4] Griffin, J. (2005) 'INTCENS': module to perform interval-censored survival analysis. package intcens from http://fmwww.bc.edu/RePEc/bocode/i

Method 3: Estimate the discrete hazard and distribution function

This approach is appropriate for "truly discrete" data, but it can be used by interval-censored data under certain conditions, and it must be interpreted accordingly. Let's start by assuming that we have

"truly discrete" data; e.g., we have a machine that produces washers, and we count how many washers it produces before it breaks.

In a discrete setting, for $i = 1, \dots,$ the survivor function is defined as

$$S_t = S(t) = P(T > t) = P(T \geq t - 1)$$

and the hazard function is defined as

$$h_t = h(t) = P(T = t | T \geq t) = P(T = t | t > t - 1)$$

It can be proved that

$$S_t = \prod_{s=1}^{t} (1 - h_s)$$

therefore, if we have an estimate $\hat{h}_t$ for h_t, we will also have

$$\hat{S}_t = \prod_{s=1}^{t} (1 - \hat{h}_s)$$

An intuitive way to estimate the hazard would be:

$$\hat{h}_t = \frac{\text{\# of individuals who failed at time t}}{\text{\# of individuals who have survived time t-1}} \tag{1}$$

For example, if we have the following small dataset:

```
input
id   time  failure
1     1      1
2     2      0
3     2      1
end
```

we can compute the empirical hazard as in the following table:

time	# indiv. survived $t-1$	# indiv. failed at t	hazard
1	3	1	1/3
2	2	1	1/2

Estimations are simpler if we take advantage of `stsplit`.
We start by `stset`-ting our data as if continuous.

```
input
id    time   failure
1       1       1
2       2       0
3       2       1
end
```

. stset time, failure(failure) id(id)
(output omitted)

```
. list id  time _t0 _t _st _d, sepby(id)

     +----------------------------------+
     | id   time   _t0   _t   _st   _d  |
     |----------------------------------|
  1. |  1      1     0    1     1    1  |
     |----------------------------------|
  2. |  2      2     0    2     1    0  |
     |----------------------------------|
  3. |  3      2     0    2     1    1  |
     +----------------------------------+
```

Then, we split the data at every integer number.

```
. stsplit x, every(1)
(2 observations (episodes) created)

. list id  time _t0 _t _st _d, sepby(id)

     +-------------------------------+
     | id   time   _t0   _t   _st   _d |
     |-------------------------------|
  1. |  1      1     0    1     1    1 |
     |-------------------------------|
  2. |  2      1     0    1     1    0 |
  3. |  2      2     1    2     1    0 |
     |-------------------------------|
  4. |  3      1     0    1     1    0 |
  5. |  3      2     1    2     1    1 |
     +-------------------------------+
```

To visualize our computation more easily, we sort by time:

```
. sort  _t id

. list _t0  _t id _st _d, sepby(_t)

     +--------------------------+
     | _t0   _t   id   _st   _d |
     |--------------------------|
  1. |   0    1    1     1    1 |
  2. |   0    1    2     1    0 |
  3. |   0    1    3     1    0 |
     |--------------------------|
  4. |   1    2    2     1    0 |
  5. |   1    2    3     1    1 |
     +--------------------------+
```

Then:

- for every value of time t (_t), we have as many valid observations as individuals survived $t-1$ (_st $=1$);
- from those, we need to compute the proportion that failed at t (_d=1) (e.g. using `proportion`, `tabulate`, `ratio`, etc)

```
. proportion  _d if _st==1, over(_t)

Proportion estimation              Number of obs   =          5

      _prop_1: _d = 0
      _prop_2: _d = 1

            1: _t = 1
            2: _t = 2

--------------------------------------------------------------
        Over | Proportion   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
_prop_1      |
           1 |   .6666667   .3333333      .0301335    .9922924
           2 |         .5         .5      .0038613    .9961387
-------------+------------------------------------------------
_prop_2      |
           1 |   .3333333   .3333333      .0077076    .9698665
           2 |         .5         .5      .0038613    .9961387
--------------------------------------------------------------

. . display _b[_prop_2:1]
.33333333

. . display _b[_prop_2:2]
.5
```

Applying method 3 to interval-censored data

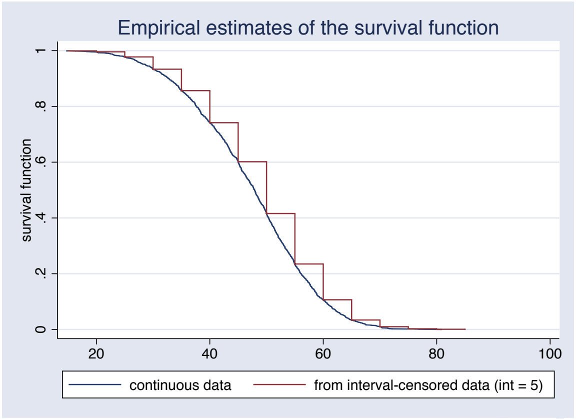
For interval-censored data, if the censoring intervals are the same for all observations, the observed data is discrete. What happens when we apply Method 3 to this kind of interval-censored data? The survivor underlying survival function will be correctly estimated for the limits of the intervals.
Let's assume that the interval length is 1; we'll have, for example:

time	observed value
0.3	1
2.5	2
47.8	48
t	int(t) +1

Therefore, the survival function $\tilde{S}(t)$ based on the discrete version of the data, will be, for every integer value

$$\tilde{S}(k) = P(int(t) + 1) > k = P(t > k) = S(k)$$

Therefore, $\tilde{S}(k) = S(k)$ for every integer k. (it's OK to use Third approach for interval-censored data, provided that results are interpreted in the right units)



To include covariates, we can fit a binary model for each group, eventually constraining the parameter to be the same; this is equivalent (from the log-likelihood point of view) to fit just one binary model, for example:

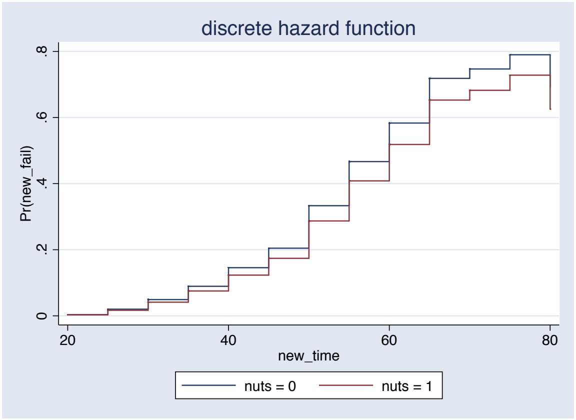
```
. use nuts_steps, clear

. gen id = _n

. gen fail = 1

. stset t_5, id(id) failure(fail)

                id:  id
     failure event:  fail != 0 & fail < .
obs. time interval:  (t_5[_n-1], t_5]
 exit on or before:  failure

------------------------------------------------------------------------------
       1200  total observations
          0  exclusions
------------------------------------------------------------------------------
       1200  observations remaining, representing
       1200  subjects
       1200  failures in single-failure-per-subject data
      59465  total analysis time at risk and under observation
                                                 at risk from t =         0
                                      earliest observed entry t =         0
                                           last observed exit t =        85
```

```
. stsplit x, every(5)
(10,693 observations (episodes) created)

.
. gen new_fail = _d

. gen new_time = _t
. cloglog new_fail nut i.new_time, nolog noomitted  noemptycells vsquish
```
(output omitted)

```
------------------------------------------------------------------------------
    new_fail |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        nuts |   -.180724   .0587115    -3.08   0.002    -.2957965   -.0656515
    new_time |
         15  |  -7.165722   1.241575    -5.77   0.000    -9.599165    -4.73228
         20  |  -5.777303   .8896678    -6.49   0.000     -7.52102   -4.033586
         25  |  -4.061683   .7661359    -5.30   0.000    -5.563282   -2.560084
         30  |  -3.149577   .7485864    -4.21   0.000     -4.61678   -1.682375
         35  |  -2.531663   .7432287    -3.41   0.001    -3.988364   -1.074961
         40  |  -2.011277   .7407928    -2.72   0.007    -3.463204   -.5593497
         45  |  -1.636863    .739931    -2.21   0.027    -3.087101   -.1866247
         50  |  -1.065383   .7389674    -1.44   0.149    -2.513732    .3829669
         55  |    -.62668   .7391177    -0.85   0.397    -2.075324    .8219641
         60  |  -.2956231   .7405705    -0.40   0.690    -1.747115    1.155868
         65  |   .0743624   .7446043     0.10   0.920    -1.385035     1.53376
         70  |   .1550258   .7621283     0.20   0.839    -1.338718     1.64877
         75  |   .2822172   .8197373     0.34   0.731    -1.324438    1.888873
       _cons |   .1623849   .7361614     0.22   0.825    -1.280465    1.605235
------------------------------------------------------------------------------
```

To predict the hazard, you can use `predict,pr` with the binary model

```
. predict hazard, pr

. keep if new_time>=20 & new_time <=80
(3,601 observations deleted)

. twoway line hazard new_time if nuts == 0 , sort connect(J)  || ///
>  line hazard new_time if nuts == 1  , sort c(J)  ///
>   legend(order( 1 "nuts = 0" 2 "nuts = 1")) ///
>   title("discrete hazard function")
```



Notes:

- Under the PH assumption for the underlying distribution, the `cloglog` model estimates the log-hazard [5]
- This method naturally accounts for left-truncation, right-censoring, and time-varying covariates.
- For not-truncated data, you can fit random-effects/multilevel models by using `melogit`, `mecloglog`, `meprobit`

[5]D. W. Hosmer, S.Lemeshow, and S. May. 2008. Applied Survival Analysis: Regression Modeling of Time to Event Data, 2nd Edition Wiley.

Models can be more flexible; for example, we'll estimate time-specific coefficients using the `promotion` dataset [6] [7]
(the sample consists of 200 male biochemists who received Ph.D.'s in the late 1950s or early 1960s)
The model if from Bauldry and Bollen. [8]
covariates:
`ungrad`: a measure of the selectivity of the undergraduate institution the individuals attended
`phdmed`: whether the individual earned his Ph.D. from a medical school.
`phdpres`: prestige of the Ph.D. granting institution.
`art1, .... art10`: cumulative count of the number of articles published by each individual for each year.

[6] Long, J. S., Allison, P. D., and MCGinnis, R. 1979 "Entrance into the academic career." American Sociological Review 44:816-830.

[7] Rabe-Hesketh,S and Skrondal,A Multilevel and Longitudinal Modeling Using Stata, Third Edition Stata Press, 2012

[8] Bauldry, S. and Bollen, K. Estimating Discrete-Time Survival Models as Structural Equation Models 2009 Anual Meeting, Population Association of America http://paa2009.princeton.edu/abstracts/90513.

```
. use promotion, clear

. stset dur, fail(event) id(id)

                id:  id
     failure event:  event != 0 & event < .
obs. time interval:  (dur[_n-1], dur]
 exit on or before:  failure

------------------------------------------------------------------------------
      301  total observations
        0  exclusions
------------------------------------------------------------------------------
      301  observations remaining, representing
      301  subjects
      217  failures in single-failure-per-subject data
     1741  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =        10
```

```
. stsplit x, every(1)
(1,440 observations (episodes) created)

.
.
. *** data comes in wide form; an art`i´ variable per year
. gen art = .
(1,741 missing values generated)

. forvalues i = 1(1)10{
  2.     qui   replace art = art`i´ if _t ==`i´
  3. }
```

We could fit

```
.logit _d i._t undgrad phdmed phdpres art
```

this would estimate a fixed parameter for a time-varying covariate; but we estimate time-specific parameters for the art variable; we fit:

```
.logit _d i._t undgrad phdmed phdpres i._t#c.art
```

```
          _d |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          _t |
           2 |  -1.388905   2.091031    -0.66   0.507    -5.487251    2.709441
           3 |   1.249771   1.438497     0.87   0.385    -1.569632    4.069174
           4 |   2.627005   1.408863     1.86   0.062    -.1343165    5.388327
           5 |   3.270551   1.405798     2.33   0.020     .5152387    6.025864
           6 |   3.403807   1.415779     2.40   0.016     .6289317    6.178682
           7 |   3.639125    1.42877     2.55   0.011     .8387874    6.439462
           8 |   2.853392   1.494764     1.91   0.056    -.0762922    5.783077
           9 |   3.312974   1.501252     2.21   0.027     .3705744    6.255373
          10 |   3.189993   1.613474     1.98   0.048     .0276423    6.352344
     undgrad |   .1557172   .0621884     2.50   0.012     .0338301    .2776043
      phdmed |  -.2408355    .171712    -1.40   0.161    -.5773848    .0957138
     phdprest |   -.025635   .0896171    -0.29   0.775    -.2012813    .1500113
    _t#c.art |
           1 |  -.2645596   .4702403    -0.56   0.574    -1.186214    .6570945
           2 |    .102138   .1524217     0.67   0.503    -.1966031    .4008792
           3 |   .1165234   .0347489     3.35   0.001     .0484169      .18463
           4 |   .0825747    .028345     2.91   0.004     .0270196    .1381298
           5 |   .0670765   .0253901     2.64   0.008     .0173127    .1168402
           6 |   .0775953   .0273048     2.84   0.004     .0240789    .1311116
           7 |   .0600786    .029941     2.01   0.045     .0013953    .1187619
           8 |   .0976589   .0413054     2.36   0.018     .0167018     .178616
           9 |   .0109346   .0422948     0.26   0.796    -.0719617    .0938309
          10 |   .0032171   .0712058     0.05   0.964    -.1363437    .1427778
       _cons |  -5.527617   1.429006    -3.87   0.000    -8.328418   -2.726817
------------------------------------------------------------------------------
```

The more information (i.e. number of obs) we have per group, the more time-specific parameters we can experiment with.

For relatively few groups, we can represent this kind of model with one equation per group (`gsem`), eventually setting constraints for parameters that are constant across groups.

This allows us to extend the model to new situations, including additional equations, and latent variables, taking advantage of structural equation models; example: joint longitudinal and discrete-survival models.