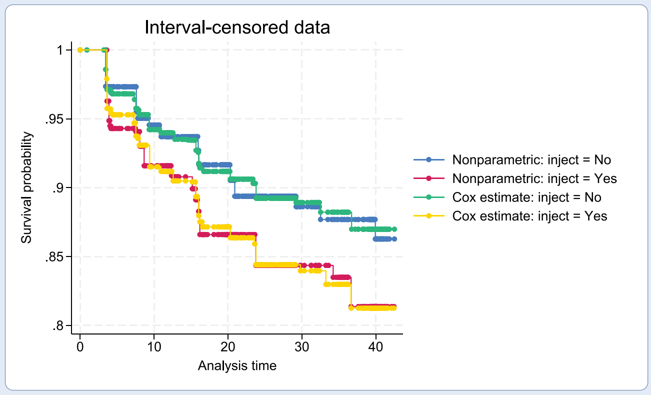
# Interval-censorded Cox model

- Genuine semiparametric modeling
- Left-censoring, right-censoring, interval-censoring
- Current-status and general interval-censored data
- Single- or multiple-record data
- Multiple events New
- Stratified estimation
- Time-varying covariates
- Two estimators for baseline hazard
- Robust and cluster–robust standard errors
- Graphs of survivor, cumulative hazard, and hazard functions
- Residual diagnostics
- Graphical checks of proportional-hazards assumption
- Graphical checks of goodness of fit
- Powerful test for a common covariate effect across all events New




Do you know the exact failure times or event times?

You can fit the Cox proportional hazards model in Stata even if you don't.

## Fit the model

The Cox proportional hazards model is widely used with right-censored event-time data because it does not require parameterization of the baseline hazard function and, under the proportional-hazards assumption, the hazard ratios are constant over time.

If we know the exact failure times, we can fit a Cox proportional hazards model using the **stcox** command. For instance, we can type

```
. stcox age_mean i.inject
```

to study the effect of mean age and injection status on failure times.

It is just as easy to fit a Cox proportional hazards model with interval-censored data, where we know only that the failure occurred sometime between two time points. With single-record-per-subject data, we specify the variables containing the upper and lower endpoints for the failure time in **stintcox**'s **interval()** option.

```
. stintcox age_mean i.inject, interval(ltime rtime)
```

Viewer - view stintcox1.smcl

view stintcox1.smcl

Dialog Also see Jump to

```
. stintcox age_mean i.inject, interval(ltime rtime)
note: using adaptive step size to compute derivatives.

Performing EM optimization (showing every 100 iterations):
Iteration 0:   Log likelihood = -1086.2564
Iteration 100: Log likelihood = -601.62673
Iteration 200: Log likelihood = -601.54523
Iteration 299: Log likelihood = -601.53336

Computing standard errors: ................ done

Interval-censored Cox regression                    Number of obs    =  1,124
Baseline hazard: Reduced intervals                       Uncensored  =      0
                                                      Left-censored  =     41
Event-time interval:                                 Right-censored  =    991
  Lower endpoint: ltime                              Interval-cens.  =     92
  Upper endpoint: rtime
                                                    Wald chi2(2)     =  11.18
Log likelihood = -601.53336                         Prob > chi2      = 0.0037

-------------------------------------------------------------------------------
             |               OPG
             |  Haz. ratio  std. err.      z    P>|z|    [95% conf. interval]
-------------+-----------------------------------------------------------------
    age_mean |   .9657816   .0124711    -2.70   0.007     .9416454    .9905365
             |
      inject |
         Yes |   1.590116   .2847623     2.59   0.010      1.11942     2.25873
-------------------------------------------------------------------------------
Note: Standard error estimates may be more variable for small datasets and
      datasets with low proportions of interval-censored observations.
```

CAP NUM INS

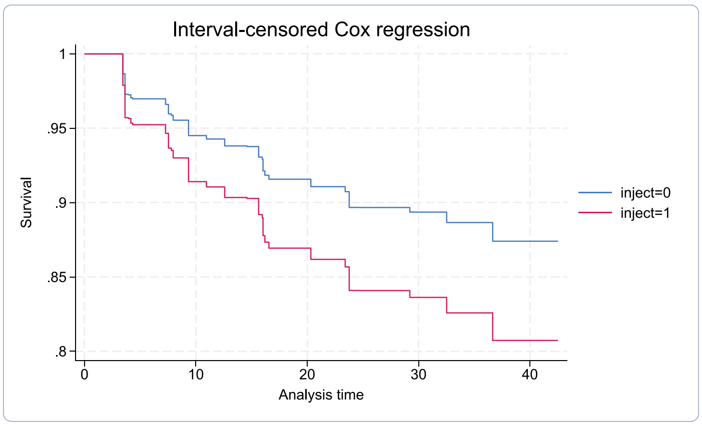
And in the presence of multiple events, we can use the new **stmgintcox** command to account for possible correlations between event times across the different events by additionally specifying the subject **id** and **event** variables:

```
. stmgintcox age_mean i.inject, id(id) event(event) interval(ltime rtime)
```

## Graph the results

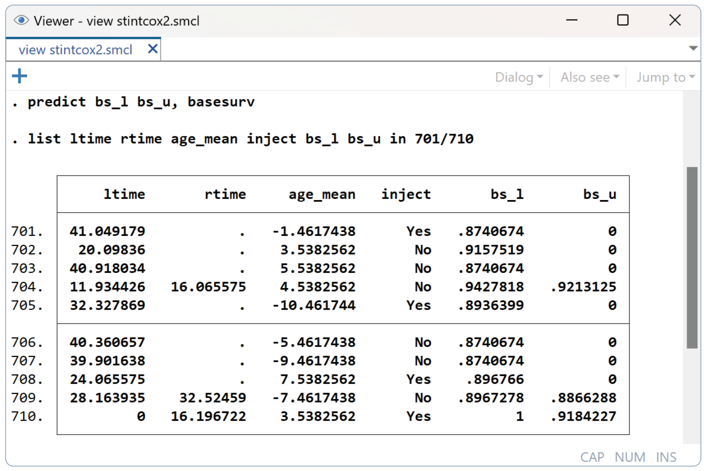
Use **stcurve** to plot the survivor, hazard, or cumulative hazard function.

```
. stcurve, survival at(inject = (0 1))
```




## Predict baseline survivor function

For each individual, we can predict the baseline survivor functions corresponding to the lower and upper endpoints of our interval.

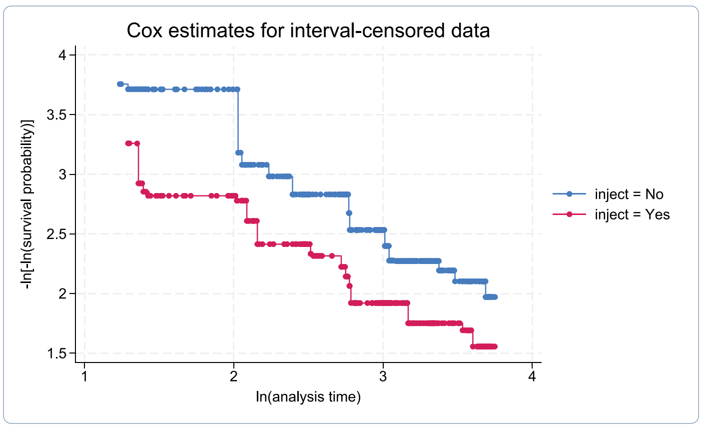


```
. predict bs_l bs_u, basesurv

. list ltime rtime age_mean inject bs_l bs_u in 701/710

       ltime       rtime    age_mean   inject       bs_l       bs_u

701.  41.049179          .  -1.4617438      Yes   .8740674          0
702.   20.09836          .   3.5382562       No   .9157519          0
703.  40.918034          .   5.5382562       No   .8740674          0
704.  11.934426  16.065575   4.5382562       No   .9427818   .9213125
705.  32.327869          .  -10.461744      Yes   .8936399          0

706.  40.360657          .  -5.4617438       No   .8740674          0
707.  39.901638          .  -9.4617438       No   .8740674          0
708.  24.065575          .   7.5382562      Yes    .896766          0
709.  28.163935   32.52459  -7.4617438       No   .8967278   .8866288
710.          0  16.196722   3.5382562      Yes          1   .9184227
```

## Check the proportional-hazards assumption

We can assess the proportional-hazards assumption graphically using the **stintphplot** command.

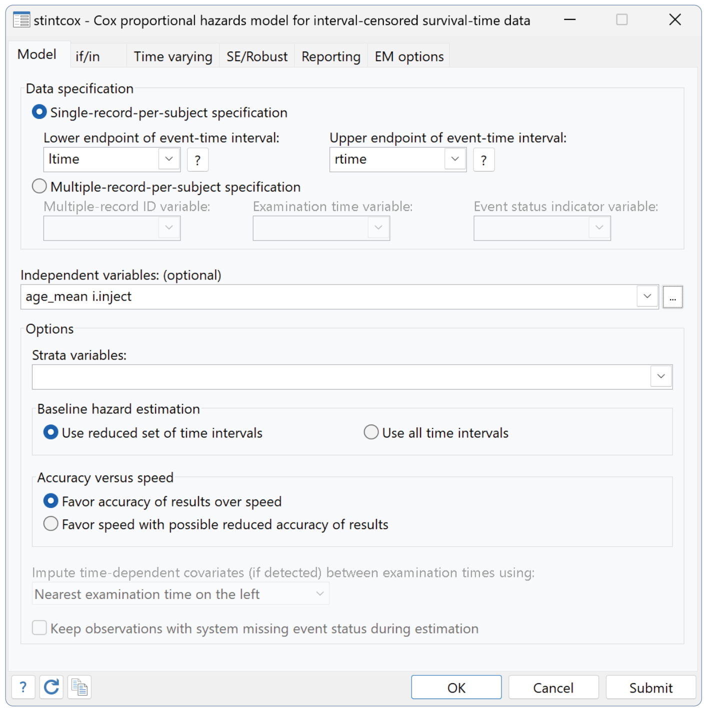
```
. stintphplot, interval(ltime rtime) by(inject)
  adjustfor(age_mean)
```




Or we can test this assumption when fitting the model. Specify the **tvc()** option to interact covariates with time, and test for coefficients of time-interacted covariates equal to zero.

```
. stintcox age_mean i.inject, interval(ltime rtime)
  tvc(age_mean i.inject)
```

## Type or point and click




stata.com/stintcox