

# Fitting Cox Proportional-Hazards Model for Interval-Censored Event-Time Data

Xiao Yang

Principal Statistician and Software Developer  
StataCorp LLC

2021 Northern European Stata Conference

## Outline

- What is interval-censored event-time data?
- Semiparametric Cox proportional hazards model for interval-censored event-time data
- Highlights of `stintcox` command
- Postestimation features of `stintcox` command
- Graphical assessment for proportional-hazards assumption
- Conclusion

# What is interval-censored event-time data?

- The event of interest is not always observed exactly, but is known only to occur within some time interval. For example, cancer recurrence, time of COVID infection.
- Interval-censored event-time data arise in many areas, including medical, epidemiological, economic, financial, and sociological studies.
- Ignoring interval-censoring may lead to biased estimates.
- There are four types of censoring: left-censoring, right-censoring, interval-censoring, and no censoring.

# Types of censoring

Event time  $T_i$  is not always exactly observed.  $(L_i, R_i]$  denotes the interval in which  $T_i$  is observed.

No censoring

$$L_i = R_i = T_i$$

Right-censoring

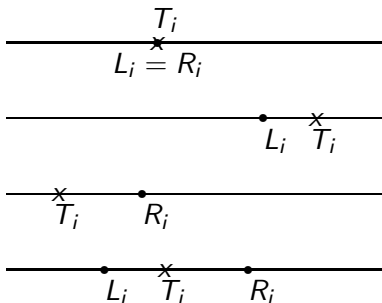
$$(L_i, R_i = +\infty)$$

Left-censoring

$$(L_i = 0, R_i]$$

Interval-censoring

$$(L_i, R_i]$$



## Types of interval-censored datasets

- Case I interval-censored data (**current status data**): occurs when subjects are observed only once, and we only know whether the event of interest occurred before the observed time. The observation on each subject is either left- or right-censored.
- Case II (**general**) interval-censored data: occurs when there are potentially two or more examination times for each study subject. The interval that brackets the event time of interest, the event-time interval, is recorded for each subject. The observation on each subject is one of left-, right-, or interval-censored.

# Methods for analyzing interval-censored data

- Simple imputation methods
- Nonparametric maximum-likelihood estimation
- Parametric regression models – `stintreg`
- **Semiparametric Cox proportional hazards model** – `stintcox`
- Bayesian analysis

# What is Cox proportional hazards model?

- The Cox proportional hazards model was first introduced by Cox in 1972 and was used routinely to analyze uncensored and right-censored event-time data.

$$h(t; \mathbf{x}) = h_0(t) \exp(\mathbf{x}'\beta)$$

- It does not require parameterization of the baseline hazard function.
- Also, under the proportional-hazard assumption, the hazard ratios are constant over time.

$$\frac{h(t; \mathbf{x}_i)}{h(t; \mathbf{x}_j)} = \frac{h_0(t) \exp(\mathbf{x}_i'\beta)}{h_0(t) \exp(\mathbf{x}_j'\beta)} = \exp(\mathbf{x}_i - \mathbf{x}_j)'\beta$$

## Cox model's challenge for interval-censored data

- Cox model is challenging for interval-censored event-time data because none of the event times are observed exactly. In particular, the traditional partial-likelihood approach is not applicable.
- Several authors have proposed spline methods to fit the Cox model to interval-censored data and those methods have their limitations.
- The direct maximum-likelihood optimization using the Newton-Raphson algorithm is highly unstable.
- Zeng, Mao, and Lin (2016) developed a genuine EM algorithm for efficient nonparametric maximum-likelihood estimation (NPMLE) method to fit the Cox model for interval-censored data.



# A genuine model for stintcox

- Suppose that the observed data consist of  $(t_{li}, t_{ui}, \mathbf{x}_i)$  for  $i = 1, \dots, n$ , where  $t_{li}$  and  $t_{ui}$  define the observed time interval and  $\mathbf{x}_i$  records covariate values for a subject  $i$ .
- Under the NPMLE approach, the baseline cumulative hazard function  $H_0$  is regarded as a step function with nonnegative jumps  $h_1, \dots, h_m$  at  $t_1, \dots, t_m$ , respectively, where  $t_1 < \dots < t_m$  are the distinct time points for all  $t_{li} > 0$  and  $t_{ui} < \infty$  for  $i = 1, \dots, n$ .
- The observed-data likelihood function is

$$\prod_{i=1}^n \exp \left\{ - \sum_{t_k \leq t_{li}} h_k \exp(\mathbf{x}_i \boldsymbol{\beta}) \right\} \left[ 1 - \exp \left\{ - \sum_{t_{li} < t_k \leq t_{ui}} h_k \exp(\mathbf{x}_i \boldsymbol{\beta}) \right\} \right]^{I(t_{ui} < \infty)} \quad (1)$$

## A genuine model for stintcox (cont.)

- Let  $W_{ik}$  ( $i = 1, \dots, n; k = 1, \dots, m$ ) be independent latent Poisson random variables with means  $h_k \exp(\mathbf{x}_i \beta)$ . Define  $A_i = \sum_{t_k \leq t_{li}} W_{ik}$  and  $B_i = I(t_{ui} < \infty) \sum_{t_{li} < t_k \leq t_{ui}} W_{ik}$ . The likelihood for the observed data  $(t_{li}, t_{ui}, \mathbf{x}_i, A_i = 0, B_i > 0)$  is

$$\prod_{i=1}^n \prod_{t_k \leq t_{li}} \Pr(W_{ik} = 0) \left\{ 1 - \Pr\left( \sum_{t_{li} < t_k \leq t_{ui}} W_{ik} = 0 \right) \right\}^{I(t_{ui} < \infty)} \quad (2)$$

- (1) and (2) are exactly equal. The maximization of a weighted sum of Poisson log-likelihood functions is strictly concave and has a closed-form solution for  $h_k$ 's.

## A genuine model for stintcox (cont.)

- We maximize (2) through an EM algorithm treating  $W_{ik}$  as missing data.
  - In the E-step, we evaluate the posterior means of  $W_{ik}$ .
  - In the M-step, we update  $\beta$  and  $h_k$  for  $k = 1, \dots, m$ .
- This method allows a completely arbitrary baseline hazard function, and the results are consistent, asymptotically normal, and asymptotically efficient.

# stintcox highlights

stintcox fits semiparametric Cox proportional hazards models to interval-censored event-time data, which may contain right-censored, left-censored, or interval-censored observations.

- Fits current-status and general interval-censored data.
- Provides four methods for standard-error computation.
- Provides standard-error computation on replay.
- Provides options to control the tradeoff between the execution speed and accuracy of the results.
- Supports two ways to choose the time intervals to be estimated for baseline hazard contributions.
- Supports stratification.

# Basic Syntax

```
stintcox [ indepvars ], interval( $t_l$   $t_u$ )
```

- `st` setting the data is not necessary and will be ignored.
- Option `interval()` is required and is used to specify two time variables that contain the endpoints of the event-time interval.
- *indepvars* is optional. You can fit a Cox model without any covariates.

# Motivating example

## Modified Bangkok IDU Preparatory Study

- 1124 subjects were initially negative for HIV-1 virus.
- They were followed and tested for HIV approximately every four months.
- The event of interest was time to HIV-1 seropositivity.
- The exact time of HIV infection was not observed, but it was known to fall in intervals between blood tests with time variables `ltime` and `rtime`.
- We want to identify the factors that influence HIV infection. The covariates that we are interested in are centered age variable (`age_mean`), and history of drug injection before recruitment (`inject`).

# Motivating example

```
. list in 701/710
```

	ltime	rtime	age_mean	inject
701.	41.049179	.	-1.4617438	Yes
702.	20.09836	.	3.5382562	No
703.	40.918034	.	5.5382562	No
704.	11.934426	16.065575	4.5382562	No
705.	32.327869	.	-10.461744	Yes
706.	40.360657	.	-5.4617438	No
707.	39.901638	.	-9.4617438	No
708.	24.065575	.	7.5382562	Yes
709.	28.163935	32.52459	-7.4617438	No
710.	0	16.196722	3.5382562	Yes

# First example

```
. 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
      (output omitted)
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
                                           Right-censored  =   991
                                           Interval-cens.  =    92
                                           Wald chi2(2)    =   11.18
                                           Prob > chi2     =   0.0037

Log likelihood = -601.53336
```

	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



## Types of standard-error estimation in stintcox

- stintcox estimates VCE for regression coefficients using the profile log-likelihood, which is obtained by maximizing the likelihood by holding the regression coefficients fixed.

Type of VCE	Order of deriv.	Stepsize
vce(opg[,stepsize(adaptive)])	first-order	adaptive
vce(opg, stepsize(fixed [#]))	first-order	fixed
vce(oim[,stepsize(adaptive)])	second-order	adaptive
vce(oim, stepsize(fixed [#]))	second-order	fixed

## Standard-error estimation example

- For small dataset or dataset with low proportions of interval-censored observations, the standard-error estimates may be more variable between different VCE methods. In that case, you may want to compare several VCE methods.
- `stintcox` provides `vce()` on replay so you can compare different VCE methods without rerunning the estimation command.

# Standard-error estimation example

```
. stintcox, vce(oim)
note: using adaptive step size to compute derivatives.
Computing standard errors: ..... done
Interval-censored Cox regression          Number of obs    = 1,124
Baseline hazard: Reduced intervals        Uncensored       =    0
                                           Left-censored    =   41
                                           Right-censored   =  991
                                           Interval-cens.   =   92
                                           Wald chi2(2)     = 11.18
                                           Prob > chi2      = 0.0037

Log likelihood = -601.53336
```

	Haz. ratio	OIM std. err.	z	P> z	[95% conf. interval]	
age_mean	.9657816	.0121666	-2.76	0.006	.9422274	.9899245
inject Yes	1.590116	.3285746	2.24	0.025	1.060572	2.384061

Note: Standard-error estimates may be more variable for small datasets and datasets with low proportions of interval-censored observations.

## favorspeed vs. favoraccuracy

- `stintcox` may become time consuming for large datasets.
- Options `favorspeed` and `favoraccuracy` control the tradeoff between the execution speed and accuracy of the results.
- `stintcox` uses less stringent convergence criteria when `favorspeed` is specified.

# favorspeed example

```
. stintcox age_mean i.inject, interval(ltime rtime) favorspeed
note: using fixed step size with a multiplier of 5 to compute derivatives.
note: using EM and VCE tolerances of 0.0001.
note: option noemhsgtolerance assumed.
Performing EM optimization (showing every 100 iterations):
Iteration 0:    log likelihood = -1086.2564
Iteration 31:  log likelihood = -602.62237
Computing standard errors: ..... done

Interval-censored Cox regression                               Number of obs    = 1,124
Baseline hazard: Reduced intervals                          Uncensored       =    0
                                                            Left-censored    =   41
                                                            Right-censored   =  991
                                                            Interval-cens.   =   92
                                                            Wald chi2(2)     =  11.19
                                                            Prob > chi2      =  0.0037

Log likelihood = -602.62237
```

	OPG					
	Haz. ratio	std. err.	z	P> z	[95% conf. interval]	
age_mean	.965774	.012463	-2.70	0.007	.9416534	.9905125
inject						
Yes	1.591654	.2848271	2.60	0.009	1.120794	2.260329

## reduced vs. full

- Option `reduced`, the default, specifies that the baseline hazard function be estimated using a reduced (innermost) set of time intervals. The innermost time intervals were originally used by Turnbull (1976) to estimate the survivor function for nonparametric estimation.
- Option `full` specifies that the baseline hazard function be estimated using all observed time intervals. This is the approach used by Zeng, Mao, and Lin (2016) and Zeng, Gao, and Lin (2017).
- Option `full` is more time consuming, but it may provide more accurate results.
- When the dataset is right-censored dataset, `full` is assumed.

## reduced vs. full example

```
. stintcox age_mean i.inject, interval(ltime rtime) full
note: using adaptive step size to compute derivatives.
Performing EM optimization (showing every 100 iterations):
Iteration 0:   log likelihood = -951.11659
              (output omitted)
Iteration 733: log likelihood = -601.56204
Computing standard errors: ..... done
Interval-censored Cox regression          Number of obs   = 1,124
Baseline hazard: All intervals           Uncensored     =    0
                                          Left-censored  =   41
                                          Right-censored =   991
                                          Interval-cens. =    92
                                          Wald chi2(2)   =  11.18
                                          Prob > chi2    =  0.0037

Log likelihood = -601.56204
```

	OPG					
	Haz. ratio	std. err.	z	P> z	[95% conf. interval]	
age_mean	.9657924	.0124751	-2.69	0.007	.9416485	.9905553
inject						
Yes	1.590554	.2849228	2.59	0.010	1.119616	2.259581

# Postestimation overview

`stintcox` provides several postestimation features after estimation:

- Predictions of hazard ratios, linear predictions, and standard errors
- Predictions of baseline survivor, baseline cumulative hazard, and baseline hazard contribution functions
- Prediction of martingale-like residuals
- Plots for survivor, hazard, and cumulative hazard function



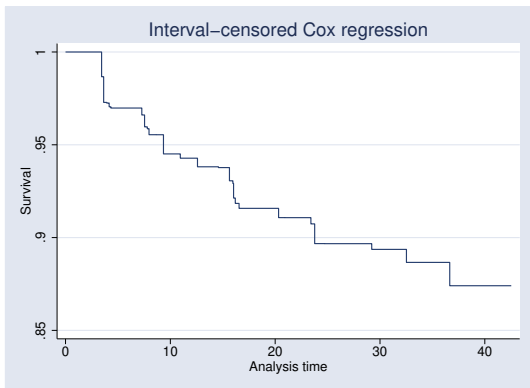
# Predict baseline survival functions

```
. stintcox age_mean i.inject, interval(ltime rtime)
  (output omitted)
. predict bs_l bs_u, basesurv
. list bs_l bs_u ltime rtime age_mean inject in 701/710
```

	bs_l	bs_u	ltime	rtime	age_mean	inject
701.	.8740674	0	41.049179	.	-1.4617438	Yes
702.	.9157519	0	20.09836	.	3.5382562	No
703.	.8740674	0	40.918034	.	5.5382562	No
704.	.9427818	.9213125	11.934426	16.065575	4.5382562	No
705.	.8936399	0	32.327869	.	-10.461744	Yes
706.	.8740674	0	40.360657	.	-5.4617438	No
707.	.8740674	0	39.901638	.	-9.4617438	No
708.	.896766	0	24.065575	.	7.5382562	Yes
709.	.8967278	.8866288	28.163935	32.52459	-7.4617438	No
710.	1	.9184227	0	16.196722	3.5382562	Yes

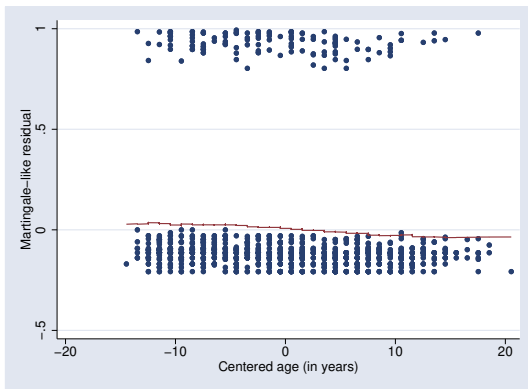
# Graph baseline survival functions

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



# Assess functional form of a covariate

```
. stintcox i.inject, interval(ltime rtime)
  (output omitted)
. predict mg, mgale
. lowess mg age_mean, mean noweight title("") note("") m(o)
```



## Graphical check for proportional-hazards assumption

- `stintpplot` plots "log-log" survival plots for each level of a nominal or ordinal covariate. The proportional-hazard assumption is satisfied when the curves are parallel.
- `stintcoxnplot` plots Turnbull's nonparametric and Cox predicted survival curves for each level of a categorical covariate. The closer the nonparametric estimates are to the Cox estimates, the less likely it is that the proportional-hazards assumption has been violated.
- You don't need to run `stintcox` before using those commands. `stintcox` has been called within those two commands.

## stintphplot basic syntax

```
stintphplot, interval( $t_l$   $t_u$ ) by()
```

- Computes nonparametric estimates of the survivor function for each level of `by()` variable.

```
stintphplot, interval( $t_l$   $t_u$ ) by() adjustfor()
```

- Fits a separate Cox model, which contains all covariates from the `adjustfor()` option, for each level of `by()` variable.

```
stintphplot, interval( $t_l$   $t_u$ ) strata() adjustfor()
```

- Fits one stratified Cox model with all covariates from the `adjustfor()` option, then plots the estimated survivor function for each level of `strata()` variable.

## stintcoxnp basic syntax

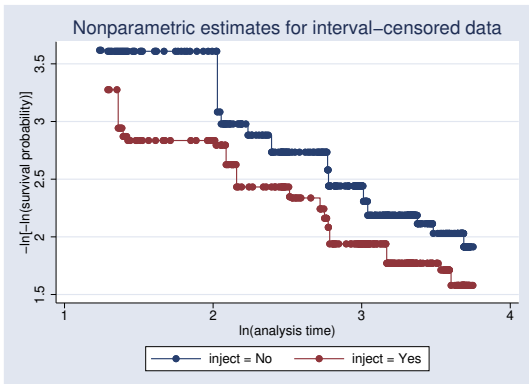
```
stintcoxnp, interval( $t_l$   $t_u$ ) by() [separate]
```

- The nonparametric and Cox predicted survivor functions are plotted for each level of by() variable.
- Option `separate` produces separate plots of nonparametric and Cox predicted survivor functions for each level of by() variable.

# Check PH-assumption for a model with a single covariate

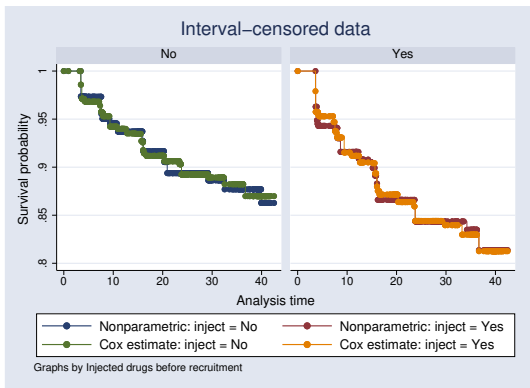
We want to check whether the PH-assumption holds for `inject`.

```
. stintplot, interval(ltime rtime) by(inject)
Computing nonparametric estimates for inject = No ...
Computing nonparametric estimates for inject = Yes ...
```



# Check PH-assumption for a model with a single covariate

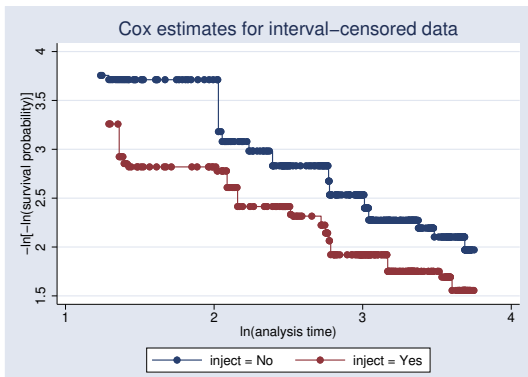
```
. stintcoxnp, interval(ltime rtime) by(inject) separate
Computing nonparametric estimates ...
Computing Cox estimates ...
```





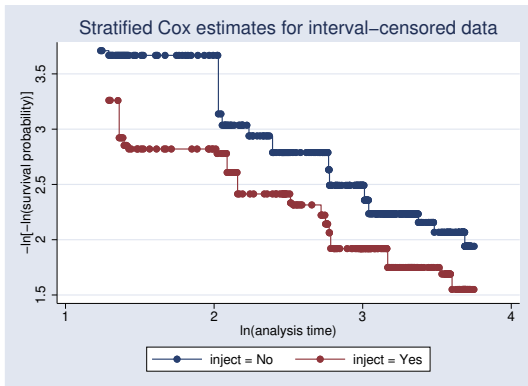
# Check PH-assumption for a model with multiple covariates

```
. stintphplot, interval(ltime rtime) by(inject) adjustfor(age_mean)
Fitting Cox model with covariates from option adjustfor()
for inject = No ...
Fitting Cox model with covariates from option adjustfor()
for inject = Yes ...
```



# Check PH-assumption for a stratified Cox model

```
. stintphplot, interval(ltime rtime) strata(inject) adjustfor(age_mean)
Fitting Cox model stratified on inject with covariates from option adjustfor()
...
```



# Conclusions

- Fit a genuine semiparametric Cox proportional-hazards model with time-independent covariates for two types of interval-censored data.
- Support different methods for standard-error computation.
- Support modeling of stratification.
- Support options to control the tradeoff between speed and accuracy.
- Support two ways to choose the time intervals to be estimated for baseline hazard function.
- Provide diagnostic measures, predictions, and much more after fitting the model.
- Provide graphical assessments for proportional-hazard assumption.

## More resources

<https://www.stata.com/manuals/ststintcox.pdf>

<https://www.stata.com/manuals/ststintcoxpostestimation.pdf>

<https://www.stata.com/manuals/ststintcoxph-assumptionplots.pdf>

## References

- [1] B. W. Turnbull. “The empirical distribution function with arbitrarily grouped censored and truncated data”. In: *Journal of the Royal Statistical Society, Series B* 38 (1976), pp. 290–295.
- [2] D. Zeng, F. Gao, and D.Y. Lin. “Maximum likelihood estimation for semiparametric regression models with multivariate interval-censored data”. In: *Biometrika* 104 (2017), pp. 505–525.
- [3] D. Zeng, L. Mao, and D.Y. Lin. “Maximum likelihood estimation for semiparametric transformation models with interval-censored data”. In: *Biometrika* 103 (2016), pp. 253–271.