

# Survival Analysis in Stata

Alex Jakubow  
University of Virginia School of Law  
Legal Data Lab

01 March 2017



# Outline

- Motivations and theoretical issues
  - Why a separate set of models for time/survival?
  - Functions and rates
  - Truncation/Censoring
- Data preparation for survival analysis
  - Wrangling for desired format
  - Stata commands: `stset`, `stdescribe`, `stvary`
- Models
  - Non-parametric (e.g., Kaplan-Meier)
  - Semi-parametric (e.g., Cox)
  - Parametric (e.g., Weibull)

# Why Survive?

Interest in the time to the occurrence of some event

- Medicine/Epidemiology: Time until death (i.e., how long do subjects *survive*?)
- Political Science: Time until democracy is overthrown (i.e., how long do democratic regimes *survive*?)
- Law: Time until Supreme Court justices retire/die (i.e., how long do SC justices *survive*?)
- Economics: Time until recession (i.e., how long does an economic boom *survive*?)

# First Cut: OLS

Model the time it takes for an event to occur with 1 independent variable ( $x$ ):

$$\text{time}_j = \beta_0 + \beta_1 x_j + \epsilon_j, \quad \epsilon_j \sim N(0, \sigma^2)$$

## First Cut: OLS

Model the time it takes for an event to occur with 1 independent variable ( $x$ ):

$$\text{time}_j = \beta_0 + \beta_1 x_j + \epsilon_j, \quad \epsilon_j \sim N(0, \sigma^2)$$

Normal, symmetrically-distributed residuals are problematic when modeling time

# First Cut: OLS

Model the time it takes for an event to occur with 1 independent variable ( $x$ ):

$$\text{time}_j = \beta_0 + \beta_1 x_j + \epsilon_j, \quad \epsilon_j \sim N(0, \sigma^2)$$

Normal, symmetrically-distributed residuals are problematic when modeling time

- Time to failure is always positive
- Censoring
- Odd distributions
  - Bimodal: Time after surgery
  - $N$ -modal: Parliamentary elections

## Solution 1: Parametric Modeling

Use a more reasonable set of distributional assumptions for  $\epsilon_j$

## Solution 1: Parametric Modeling

Use a more reasonable set of distributional assumptions for  $\epsilon_j$

Several native options in Stata:

- Exponential: `streg ..., dist(exponential)`
- Gompertz: `streg ..., dist(gompertz)`
- Log-logistic: `streg ..., dist(llogistic)`
- Weibull: `streg ..., dist(weibull)`
- Log-normal: `streg ..., dist(lnormal)`
- Generalized gamma: `streg ..., dist(ggamma)`



## Solution 2: Semi-Parametric Modeling

No distributional assumptions, at all!

## Solution 2: Semi-Parametric Modeling

No distributional assumptions, at all!

But, how?

## Solution 2: Semi-Parametric Modeling

No distributional assumptions, at all!

But, how?

- Let time order the distribution of observations
- Reduce to series of binary-outcome analyses



## Solution 2: Semi-Parametric Modeling

Basic intuition behind the Cox regression (`stcox`)

- Fits a series of  $j$  conditional logistic models
  - Condition on `outcome == 1` for only one observation within each separate analysis
- Constrains regression coefficients to be the same across models

## Solution 2: Semi-Parametric Modeling

Basic intuition behind the Cox regression (`stcox`)

- Fits a series of  $j$  conditional logistic models
  - Condition on `outcome == 1` for only one observation within each separate analysis
- Constrains regression coefficients to be the same across models

Implications:

- Analysis proceeds without making any assumptions about the distribution of failure times (*non-parametric*)
- Analysis assumes that covariate values systematically influence the probability that a failure occurs (*parametric*)

## Solution 2: Semi-Parametric Modeling

Basic intuition behind the Cox regression (`stcox`)

- Fits a series of  $j$  conditional logistic models
  - Condition on `outcome == 1` for only one observation within each separate analysis
- Constrains regression coefficients to be the same across models

Implications:

- Analysis proceeds without making any assumptions about the distribution of failure times (*non-parametric*)
- Analysis assumes that covariate values systematically influence the probability that a failure occurs (*parametric*)
- non-parametric time + parametric covariates =  
**semi-parametric model**

## Solution 3: Non-Parametric Modeling

Let the data "speak for itself"

- Useful when no covariates exist or when covariates are qualitative in nature
- Can estimate probability of survival (or failure) past a certain time
- Can compare survival experiences across values of a categorical variable (e.g., by gender)

Methods:

- Kaplan-Meier method
- Nelson-Aalen method
- Log-rank test

## Solutions Overview

	<b>(Full-) Parametric</b>	<b>Semi- Parametric</b>	<b>Non- Parametric</b>
Failure time distribution assumption?	Yes	No	No
Covariate effects assumption?	Yes	Yes	No
Use information b/w failure times?	Yes	No	No



# Important Functions in Survival Modeling

## The Survivor Function

- Defined as the probability of surviving beyond some time,  $t$
- Equals one at  $t = 0$ , decreases towards 0 as  $t \rightarrow \infty$

## The Survival Function

$$S(t) = 1 - F(t) = \Pr(T > t)$$

$F(t)$  = cumulative distribution function of random variable,  $T$   
 $T$  = time to a failure event

# Important Functions in Survival Modeling

## The Hazard Function

- a.k.a., conditional failure rate, intensity function. . .
- Defined as the rate at which risk accumulates (e.g., instantaneous rate of failure)
- Varies from 0 (no risk) to  $\infty$  (failure is certain)

### The Hazard Function

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{\Pr(t + \Delta t > T > t | T > t)}{\Delta t}$$

$t$  = failure event

$T$  = time to the failure event

$\Delta t$  = width of time interval

# Important Functions in Survival Modeling

## The Cumulative Hazard Function

- Defined as the total amount of risk that has been accumulated up to time  $t$

### The Cumulative Hazard Function

$$H(t) = \int_0^t h(u) du$$

$H(t)$  = integral of the hazard rate between 0 and time  $t$

# Censoring

**Failure event occurs when the subject is not under observation (or in our dataset)**

- Right-Censoring
- Interval-Censoring
- Left-Censoring

# Censoring

**Failure event occurs when the subject is not under observation (or in our dataset)**

- Right-Censoring
  - Failure never occurs during period of observation
  - Still alive at end of clinical trial; withdrew early
- Interval-Censoring
- Left-Censoring

# Censoring

**Failure event occurs when the subject is not under observation (or in our dataset)**

- Right-Censoring
  - Failure never occurs during period of observation
  - Still alive at end of clinical trial; withdrew early
- Interval-Censoring
  - Failure occurs during period of observation, but exact time of failure is unknown
  - Subject healthy at  $t = 3$ , but sick at  $t = 5$
- Left-Censoring

# Censoring

## **Failure event occurs when the subject is not under observation (or in our dataset)**

- Right-Censoring
  - Failure never occurs during period of observation
  - Still alive at end of clinical trial; withdrew early
- Interval-Censoring
  - Failure occurs during period of observation, but exact time of failure is unknown
  - Subject healthy at  $t = 3$ , but sick at  $t = 5$
- Left-Censoring
  - Failure event occurred before period of observation
  - Individual already employed during first interview for time-to-employment study

# Truncation

**Period over which subject was not observed but is, a posteriori, known not to have failed**

- Left-truncation (delayed entry)
- Interval-truncation (gaps)
- Right-truncation



# Truncation

**Period over which subject was not observed but is, a posteriori, known not to have failed**

- Left-truncation (delayed entry)
  - Period of ignorance extends from on or before the onset of risk to sometime after the onset of risk
  - Did not observe subject until  $t = 3$
- Interval-truncation (gaps)
- Right-truncation

# Truncation

**Period over which subject was not observed but is, a posteriori, known not to have failed**

- Left-truncation (delayed entry)
  - Period of ignorance extends from on or before the onset of risk to sometime after the onset of risk
  - Did not observe subject until  $t = 3$
- Interval-truncation (gaps)
  - 1+ periods of ignorance after the onset of risk
  - Subject observed regularly between  $t = 0$  and  $t = 4$ , but not again until  $t = 8$ .
- Right-truncation

# Truncation

## **Period over which subject was not observed but is, a posteriori, known not to have failed**

- Left-truncation (delayed entry)
  - Period of ignorance extends from on or before the onset of risk to sometime after the onset of risk
  - Did not observe subject until  $t = 3$
- Interval-truncation (gaps)
  - 1+ periods of ignorance after the onset of risk
  - Subject observed regularly between  $t = 0$  and  $t = 4$ , but not again until  $t = 8$ .
- Right-truncation
  - Form of selection bias where observations only include subjects who experienced an event prior to some point in time
  - Subjects selected from a cancer register only includes individuals who had already developed cancer

# Censoring and Truncation

Some parting thoughts. . .

- Censoring and truncation pose inferential issues, but can be generally accommodated in survival analysis
- Easiest to handle with parametric models
- Single vs. Multiple failures

# The Simple Format Revisited

Toy dataset with one  
observation per subject:

time	x
1	3
5	2
9	4
20	9
22	-4

- time: time of failure
- x: independent variable

# The Simple Format Revisited

Toy dataset with one  
observation per subject:

time	x
1	3
5	2
9	4
20	9
22	-4

This approach is entirely  
inadequate for several reasons:

- time: time of failure
- x: independent variable

# The Simple Format Revisited

Toy dataset with one  
observation per subject:

time	x
1	3
5	2
9	4
20	9
22	-4

- time: time of failure
- x: independent variable

This approach is entirely  
inadequate for several reasons:

- Censoring?
- Truncation?
- Repeated failures?

## The Desired Format

A better way to record data...

id	t0	t1	outcome	x
1	0	1	1	3
2	0	5	1	2
3	0	9	1	4
4	0	20	1	9
5	0	22	1	-4

- id = subject identifier
- t0 = start time
- t1 = end time
- outcome = failure?
- x = independent variable



# The Desired Format

## Right-Censoring Example

id	t0	t1	outcome	x
1	0	1	1	3
2	0	5	1	2
3	0	9	1	4
4	0	20	1	9
5	0	22	0	-4

- Subject 5 did not fail yet

# The Desired Format

## Left-Truncation Example

id	t0	t1	outcome	x
1	0	1	1	3
2	0	5	1	2
3	3	9	1	4
4	0	20	1	9
5	0	22	0	-4

- Subject 3 entered late ( $t = 3$ )

# The Desired Format

## Interval-Truncation Example

id	t0	t1	outcome	x
1	0	1	1	3
2	0	5	1	2
3	3	9	1	4
4	0	9	1	9
4	11	20	1	9
5	0	22	0	-4

- Subject 4 was unobserved between  $t = 9$  &  $t = 11$

# The Desired Format

## A Complicated Subject

id	t0	t1	outcome	x
1	0	1	1	3
...	...	...	...	...
5	2	4	0	-4
5	6	8	0	-4
5	10	15	0	-4
5	17	21	0	-4

- What's up with Subject 5?

## The Desired Format

Stata will make the correct statistical calculations if (and only if) datasets follow this format

id	t0	t1	outcome	x
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

# Dealing with "Real-World" Data

Data do not always arrive in a format that Stata will use correctly, so we need to pre-process them before estimating any models

Real-world considerations:

- Transaction or "snapshot" data
- Calendar time vs. analysis time
- Enduring (characteristic) vs. Instantaneous (event) variables

# Dealing with “Real-World” Data

Data do not always arrive in a format that Stata will use correctly, so we need to pre-process them before estimating any models

Real-world considerations:

- Transaction or “snapshot” data
- Calendar time vs. analysis time
- Enduring (characteristic) vs. Instantaneous (event) variables
  - Enduring: characteristics we treat as valid for longer than an instant (e.g., gender, age in years, BP at time of hospitalization, etc.))
  - Instantaneous: events that occurred at particular instant in time (e.g., heart attack, going bankrupt, outbreak of a civil war, etc.)

## Transforming "Long-Form" Snapshot Data

id	date	rectype	x	xbin
2	20jan2000	enrolled (1)	2	0
2	22jan2000	checkup (2)		1
2	24jan2000	checkup (2)		0
2	25jan2000	failed (9)		

- id: subject ID
- date: entry date
- rectype: record or event type (instantaneous)
- x: continuous independent variable (enduring)
- xbin: binary independent variable (enduring)



# Transforming "Long-Form" Snapshot Data

## Objectives:

- Collapse each pair of records into one time-span record (2 rows to 1 row)
- Enduring variables take values of first observation in each pair
- Instantaneous variables take values of second observation in each pair

We want the values of (enduring) variables over the entire time span and the values of events (instantaneous) at the *end* of the period

## Transforming "Long-Form" Snapshot Data

From this...

id	date	rectype	x	xbin
2	20jan2000	enrolled (1)	2	0
2	22jan2000	checkup (2)		1

To this...

id	date0	date1	rectype	x	xbin
2	20jan2000	22jan2000	?	?	?

# Transforming "Long-Form" Snapshot Data

From this...

id	date	rectype	x	xbin
2	20jan2000	enrolled (1)	2	0
2	22jan2000	checkup (2)		1

To this...

id	date0	date1	rectype	x	xbin
2	20jan2000	22jan2000	checkup (2)	?	?

## Transforming "Long-Form" Snapshot Data

From this...

id	date	rectype	x	xbin
2	20jan2000	enrolled (1)	2	0
2	22jan2000	checkup (2)		1

To this...

id	date0	date1	rectype	x	xbin
2	20jan2000	22jan2000	checkup (2)	2	0

## Transforming "Long-Form" Snapshot Data

Use the `snapspan` command in `stata`...

```
. snapspan idvar time_var instantaneous_vars,  
generate(new_begin_date)
```

## Transforming "Long-Form" Snapshot Data

Use the `snapspan` command in `stata`...

```
. snapspan idvar time_var instantaneous_vars,  
generate(new_begin_date)
```

*Let's try this in Stata...*

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used



# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used
- Executes diagnostic checks on the data

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used
- Executes diagnostic checks on the data
- Facilitates the creation of complicated rules for:

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used
- Executes diagnostic checks on the data
- Facilitates the creation of complicated rules for:
  - excluding vs. including observations

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used
- Executes diagnostic checks on the data
- Facilitates the creation of complicated rules for:
  - excluding vs. including observations
  - defining onset of risk

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used
- Executes diagnostic checks on the data
- Facilitates the creation of complicated rules for:
  - excluding vs. including observations
  - defining onset of risk
  - defining "failure"

# The Many Purposes of `stset`

`stset` command aids survival analysis in three important ways:

- Informs Stata about the structure of survival data being used
- Executes diagnostic checks on the data
- Facilitates the creation of complicated rules for:
  - excluding vs. including observations
  - defining onset of risk
  - defining "failure"
  - defining analysis time

# Syntax Basics for `stset`

## Most Basic Usage

```
. stset time_of_failure_var
```

# Syntax Basics for `stset`

## Most Basic Usage

```
. stset time_of_failure_var
```

<code>failtime</code>	<code>x</code>
1	3
5	2
9	4
20	9
22	-4



# Syntax Basics for `stset`

## Most Basic Usage

```
. stset time_of_failure_var
```

<i>failtime</i>	<i>x</i>
1	3
5	2
9	4
20	9
22	-4

```
. stset failtime
```

- *failtime* contains the times at which each obs failed.

# Syntax Basics for stset

## Slightly Less Basic Usage

```
. stset time_of_failure_or_censoring_var,  
failure(one_if_failure_var)
```

# Syntax Basics for stset

## Slightly Less Basic Usage

```
. stset time_of_failure_or_censoring_var,  
failure(one_if_failure_var)
```

lasttime	x	failed
1	3	1
5	2	1
9	4	1
20	9	1
22	-4	0

# Syntax Basics for stset

## Slightly Less Basic Usage

```
. stset time_of_failure_or_censoring_var,  
failure(one_if_failure_var)
```

lasttime	x	failed
1	3	1
5	2	1
9	4	1
20	9	1
22	-4	0

```
. stset lasttime,  
failure(failed)
```

- if failed = 1, lasttime denotes failure time
- if failed = 0, lasttime denotes censoring time

# Setting Analysis Time

Analysis time is always 0 at the onset of risk, so we need to define

$$t = \text{someFunction}(\text{time/date variables in dataset})$$

Two options:

- Manually-define  $t$  and then `stset` on it
- Use `stset`'s `origin` and (possibly) `scale` options so `stset` calculates it automatically for us

# Setting Analysis Time

An important conceptual distinction between *time* and *t*

- *time*: time as recorded in data
- *t*: analysis time

$$t = \frac{\text{time} - \text{origin}}{\text{scale}}$$

## Setting Analysis Time

An important conceptual distinction between *time* and *t*

- *time*: time as recorded in data
- *t*: analysis time

$$t = \frac{\text{time} - \text{origin}}{\text{scale}}$$

Stata Defaults to  $t = 0$

- `origin(0)`
- `scale(1)`
- onset of risk otherwise assumed to be  $t = 0$  in Stata time, or, January 1, 1960...

# Setting Analysis Time

## Examples of using `origin()`



# Setting Analysis Time

## Examples of using `origin()`

```
origin(time td(15feb1999))
```

# Setting Analysis Time

## Examples of using `origin()`

```
origin(time td(15feb1999))
```

- onset of risk occurs on February 15, 1999 (or 14290 as an integer)

# Setting Analysis Time

## Examples of using `origin()`

```
origin(time td(15feb1999))
```

- onset of risk occurs on February 15, 1999 (or 14290 as an integer)

```
origin(time diagdate)
```

# Setting Analysis Time

## Examples of using `origin()`

```
origin(time td(15feb1999))
```

- onset of risk occurs on February 15, 1999 (or 14290 as an integer)

```
origin(time diagdate)
```

- onset of risk uses date of diagnosis
- if subjects have multiple records, Stata will select earliest (i.e., smallest) nonmissing value of `diagdate`

## Setting Analysis Time

### Examples of using `origin()`

```
origin(time td(15feb1999))
```

- onset of risk occurs on February 15, 1999 (or 14290 as an integer)

```
origin(time diagdate)
```

- onset of risk uses date of diagnosis
- if subjects have multiple records, Stata will select earliest (i.e., smallest) nonmissing value of `diagdate`

```
origin(time min(diagdate, d2date))
```

# Setting Analysis Time

## Examples of using `origin()`

```
origin(time td(15feb1999))
```

- onset of risk occurs on February 15, 1999 (or 14290 as an integer)

```
origin(time diagdate)
```

- onset of risk uses date of diagnosis
- if subjects have multiple records, Stata will select earliest (i.e., smallest) nonmissing value of `diagdate`

```
origin(time min(diagdate, d2date))
```

- Stata selects earliest instance of date of diagnosis (`diagdate`) or retrospective judgment from healthcare professional (`d2date`)

# Setting Analysis Time

## Examples of using `scale()`

# Setting Analysis Time

## Examples of using `scale()`

```
scale(30)
```



# Setting Analysis Time

## Examples of using `scale()`

`scale(30)`

- convert days to months

# Setting Analysis Time

## Examples of using `scale()`

`scale(30)`

- convert days to months

`scale(365.25)`

# Setting Analysis Time

## Examples of using `scale()`

`scale(30)`

- convert days to months

`scale(365.25)`

- convert days to years

# Setting Analysis Time

## Examples of using `scale()`

`scale(30)`

- convert days to months

`scale(365.25)`

- convert days to years

`scale(1/12)` or `scale(0.08333)`

# Setting Analysis Time

## Examples of using `scale()`

`scale(30)`

- convert days to months

`scale(365.25)`

- convert days to years

`scale(1/12)` or `scale(0.08333)`

- convert years to months

## New Variables with `stset`

Executing `stset` command automatically adds 4 new variables to dataset:

- `_t0`: start time for each record
- `_t`: end time for each record
- `_d`: records outcome at the end of time span (1 if time span ends in failure; 0 if otherwise)
- `_st`: indicates whether observation is relevant in current analysis (1 if obs is to be used; 0 if obs is to be ignored)

## New Variables with `stset`

Executing `stset` command automatically adds 4 new variables to dataset:

- `_t0`: start time for each record
- `_t`: end time for each record
- `_d`: records outcome at the end of time span (1 if time span ends in failure; 0 if otherwise)
- `_st`: indicates whether observation is relevant in current analysis (1 if obs is to be used; 0 if obs is to be ignored)

*Let's try this in Stata...*

## Other Specifications with `stset`

### `failure()`

- Identifies variable that marks failures
- `failure(failed)` → all non-zero values of `failure` indicate failure
- `failure(failed==9)` → only 9s treated as failure
- `failure(event==9/11)` → failures when `event` equals 9, 10, or 11
- NOTE: missing values treated as non-failures (0s), not missing



## Other Specifications with `stset`

### `exit()`

- By default, subjects exit when:
  - Subject runs out of data (censored)
  - Subject fails for the first time (as denoted by `failure()`)

What if we wish to have a subject "exit" the dataset based on third set of criteria?

- e.g., drop cancer patients from analysis if heart attack
- `exit(heartatk == 1)`

What if we want to allow repeated failures (e.g., strokes)?

- `exit(time .)` → each subject should exit only when he/she is out of data, regardless of the number of failures, if any

## Other Specifications with `stset`

### `enter()`

- By default, subjects enter:
  - at analysis time  $t = 0$  (specified in `origin()`), OR
  - at subjects available earliest data record

### Specify alternative entry criteria

- `enter(event == 2)`
- Subjects enter when event is 2 or  $t = 0$ , whichever is later

### Specify alternative entry criteria w.r.t. a time variable

- `enter(time intvdate)`
- Subjects enter after earliest time given by `intvdate` (interview date)

## Other Specifications with `stset`

### `id()`

- Stata assumes 1 record per subject

Specify multiple per subject records using `id()`

- `id(idVar)`
- Multiple entires on the basis of `idVar`

## Other Specifications with `stset`

### `time0()`

- By default, `stata` calculates the beginning of each time span automatically
- No time gaps are assumed

Specify a begin-of-span variable to account for gaps and/or

- `time0(begin)`
- Each time-span entry begins with `begin`

# Summarizing stset

## Minimum Requirement

- Time at which failure occurred
- stset *time\_of\_failure\_var*

# Summarizing `stset`

## Other 'Details'

- `failure()`
- Options that affect the definition of analysis time:
  - `origin()`
  - `scale()`
- Options that affect when the subject is under observation:
  - `enter()`
  - `exit()`
- Options that provide info about data structure:
  - `id()`
  - `time0()`

# Stata Examples

## Example 1: Hip Fracture Data

- Objective: Quantify benefit of a new inflatable device on hip fractures
- Sample: 48 women aged 60+ w/ no prior history of hip trauma
  - 28 randomly-assigned to wear the device; 20 in the control group
- Covariates: Blood calcium levels drawn every 5 months
- Dependent Variable: Time to hip fracture (or censoring)
- Other: Women who were hospitalized (for any reason) should no longer be considered at-risk for fracture

# Stata Examples

## Example 2: Reye's Syndrome

- Objective: Does an experimental treatment protocol improve the odds of survival among 150 children diagnosed with Reyes Syndrome?
- Sample: 150 patients
- Covariates: Treatment protocol (experimental vs. standard)
- Dependent Variable: Time to death (or censoring)



# Using stset in Stata

*Let's try this in Stata...*

# A Suggested Pre-Analysis Workflow

What do you do now that you've `stset` your data?

- Examine output of `stset`
- Casual inspection of data (`_t0`, `_t`, `_d`, `_st`)
- Run `stdescribe` to help identify problems
- Run `stvary` if multiple-record data
- Use `stfill` and/or `streset` to fix any identified problems

# Stata Examples

Illustrate workflow using our running stata examples:

- Hip-fracture trial
- Reye's syndrome treatment trial

# Modeling Options for Survival Data

Three basic modeling solutions for survival/event history data:

- Non-parametric → No covariates, no distributional assumptions
- Semi-parametric → Covariates, but no distributional assumptions
- Parametric → Covariates and distributional assumptions

# Non-Parametric Modeling

## The Kaplan-Meier Estimator

A Non-parametric estimate of the survivor function  $S(t) \rightarrow$  prob. of survival past time  $t$

$$\hat{S}(t) = \prod_{j|t_j \leq t} \left( \frac{n_j - d_j}{n_j} \right)$$

$n_j$ : number of individuals at risk at time  $t_j$

$d_j$ : number of failures at time  $t_j$

$\prod_{j|t_j \leq t}$ : Product over all observed failure times  $\leq t$

# The Kaplan-Meier Estimator

Other properties:

- Operates only on observed failures times (censoring times are ignored)
- Stata assumes censoring happens immediately prior ( $t - \epsilon$ ) to failures if different units simultaneously fail and are censored at time  $t$

# A Simple Example

## A Simple Example

Our Dataset:

id	t	failed
1	2	1
2	4	1
3	4	1
4	5	0
5	7	1
6	8	0



## A Simple Example

Our Dataset:

id	t	failed
1	2	1
2	4	1
3	4	1
4	5	0
5	7	1
6	8	0

Alternatively:

t	# at risk	# failed	# censored
2	6	1	0
4	5	2	0
5	3	0	1
7	2	1	0
8	1	0	1

## A Simple Example

Compute component probabilities ( $p$ ) of survival at each  $t$

$t$	# at risk	# failed	# censored	$p$
2	6	1	0	5/6
4	5	2	0	3/5
5	3	0	1	1
7	2	1	0	1/2
8	1	0	1	1

## A Simple Example

The Kaplan-Meier estimate: running product of the values of  $p$

$t$	# at risk	# failed	# censored	$p$	$\hat{S}(t)$
2	6	1	0	5/6	5/6
4	5	2	0	3/5	1/2
5	3	0	1	1	1/2
7	2	1	0	1/2	1/4
8	1	0	1	1	1/4

# Non-parametric Estimation in Stata

The suite of `sts` commands handle most non-parametric functions in stata:

- `sts list` → Calculates Kaplan-Meier estimate
- `sts graph` → Graphs Kaplan-Meier estimate

*Let's try this in Stata...*

# Semi-Parametric Modeling

## Cox Proportional Hazards Model

The hazard rate for the  $j$ th subject in the data is:

$$h(t|\mathbf{x}_j) = h_0(t)\exp(\mathbf{x}_j\boldsymbol{\beta}_x)$$

$h_0(t)$ : The baseline hazard rate

$\mathbf{x}_j\boldsymbol{\beta}_x$ : Independent variables with estimated coefficients

# Semi-Parametric Modeling

## Cox Proportional Hazards Model

Key implications:

- Baseline hazard rate is constant → covariates multiplicatively shift hazard function for each subject
- Baseline hazard rate is not parametrized
  - Estimation without defining  $h_0(t)$  b/c analysis confined to only those times where failures occur

# Semi-Parametric Modeling

## Cox Proportional Hazards Model

Key implications:

- Baseline hazard rate is constant  $\rightarrow$  covariates multiplicatively shift hazard function for each subject
- Baseline hazard rate is not parametrized
  - Estimation without defining  $h_0(t)$  b/c analysis confined to only those times where failures occur
- Main advantage: No assumptions required about shape of  $h_0(t)$

# Semi-Parametric Modeling

## Cox Proportional Hazards Model

Key implications:

- Baseline hazard rate is constant  $\rightarrow$  covariates multiplicatively shift hazard function for each subject
- Baseline hazard rate is not parametrized
  - Estimation without defining  $h_0(t)$  b/c analysis confined to only those times where failures occur
- Main advantage: No assumptions required about shape of  $h_0(t)$
- Main disadvantage: Loss of efficiency  $\rightarrow$  specifying *correct* functional form for  $h_0(t)$  improves estimation of  $\beta_x$



# Cox Regressions in Stata

- . `stcox covariatesList [, nohr]`
  - Do not specify response variable
    - response is always the triple  $(t_0, t, d) \rightarrow$  time span  $(t_0, t]$  with failure/censoring indicator  $d$
  - Cox model will not report an intercept (subsumed into  $h_0(t)$ )
  - Exponentiated coefficients are reported by default
    - ratio of hazards for a 1-unit change in corresponding covariate
    - specify `, nohr` for un-exponentiated coefficients

## Cox Regessions in Stata

- `stcox covariatesList [, nohr]`
  - Do not specify response variable
    - response is always the triple  $(t_0, t, d) \rightarrow$  time span  $(t_0, t]$  with failure/censoring indicator  $d$
  - Cox model will not report an intercept (subsumed into  $h_0(t)$ )
  - Exponentiated coefficients are reported by default
    - ratio of hazards for a 1-unit change in corresponding covariate
    - specify `, nohr` for un-exponentiated coefficients

*Let's try this in Stata...*

## Post-Estimation Options in Stata

We can estimate functions related to the baseline hazard function ( $h_0(t)$ ), including a smoothed estimate of  $h_0(t)$ , itself

Condition on estimates of estimates of  $\beta_x$ :

- $H_0(t)$ : baseline cumulative hazard function
- $S_0(t)$ : baseline survivor function
- $h_0(t)$ : baseline hazard function (rate of change in  $H_0(t)$ )

## Post-Estimation Options in Stata

We can estimate functions related to the baseline hazard function ( $h_0(t)$ ), including a smoothed estimate of  $h_0(t)$ , itself

Condition on estimates of estimates of  $\beta_x$ :

- $H_0(t)$ : baseline cumulative hazard function
- $S_0(t)$ : baseline survivor function
- $h_0(t)$ : baseline hazard function (rate of change in  $H_0(t)$ )

*Let's try this in Stata...*

## Ties and Likelihood Calculations

Cox regression calculates results in two basic steps:

- Identifies the *risk set*: subjects who are at risk of failure when at least one subject fails
- Maximizes the conditional probability of failure

# Ties and Likelihood Calculations

Cox regression calculates results in two basic steps:

- Identifies the *risk set*: subjects who are at risk of failure when at least one subject fails
- Maximizes the conditional probability of failure

Implications:

- The times at which failures occur are irrelevant; the *ordering* of failures is what matters

## Ties and Likelihood Calculations

Cox regression calculates results in two basic steps:

- Identifies the *risk set*: subjects who are at risk of failure when at least one subject fails
- Maximizes the conditional probability of failure

Implications:

- The times at which failures occur are irrelevant; the *ordering* of failures is what matters
- So, what happens when multiple failures occur at the same time (i.e., ties)?

## Ties and Likelihood Calculations

Cox regression calculates results in two basic steps:

- Identifies the *risk set*: subjects who are at risk of failure when at least one subject fails
- Maximizes the conditional probability of failure

Implications:

- The times at which failures occur are irrelevant; the *ordering* of failures is what matters
- So, what happens when multiple failures occur at the same time (i.e., ties)?
  - Exact marginal method → `stcox xvar, exactm`
  - Exact partial method → `stcox xvar, exactp`
  - Efron → `stcox xvar, efron`
  - Breslow (the default) → `stcox xvar`



# Cox Model Diagnostics: Model Specification

## Tests of the Proportional Hazards Assumption

### Re-estimation

- The link test
  - $\beta_1(\mathbf{x}\hat{\beta}_x) + \beta_2(\mathbf{x}\hat{\beta}_x)^2$ , where  $H_0 : \beta_2 = 0$
  - re-estimate using linear prediction and square of linear prediction as covariates
  - square of linear predictor should be insignificant
- Interact covariates w/ time ( $t$ )
  - $\mathbf{x}\beta_x + \beta_1(x_1 t)$ , where  $H_0 : \beta_1 = 0$
  - $\mathbf{x}\beta_x + \beta_2(x_2 t)$ , where  $H_0 : \beta_2 = 0 \dots$
  - separately fit one-model per covariate

# Cox Model Diagnostics: Model Specification

## Tests of the Proportional Hazards Assumption

### Residuals

- Test on Schoenfeld residuals
  - Analyze the distribution of residuals over analysis time
  - $H_0$ : No relationship

# Cox Model Diagnostics: Model Specification

## Tests of the Proportional Hazards Assumption

### Residuals

- Test on Schoenfeld residuals
  - Analyze the distribution of residuals over analysis time
  - $H_0$ : No relationship

# Cox Model Diagnostics: Model Specification

## Tests of the Proportional Hazards Assumption

### Residuals

- Test on Schoenfeld residuals
  - Analyze the distribution of residuals over analysis time
  - $H_0$ : No relationship

### Graphical Methods (discrete covariates only)

- Parallel lines test
  - Plot a function of the Kaplan-Meier estimate against a function of analysis time, for each level of covariate in question
  - Specifically:  $-\ln[-\ln(\hat{S}(t))]$  vs.  $\ln(t)$

# Cox Model Diagnostics: Goodness-of-Fit

## Goodness-of-Fit Using Cox-Snell Residuals

The Plot:

- Y-axis: estimates for the cumulative hazard rate  $(\hat{H}_0)$  derived from Nelson-Aalen estimation
- X-axis: Cox-Snell residuals  $CSr_j = \hat{H}_0(t_j) \exp(\mathbf{x}_j \hat{\beta}_x)$ 
  - $\hat{H}_0(t_j)$ : estimate of cumulative hazard function
  - $\mathbf{x}_j \hat{\beta}_x$ : estimate of covariate vector

The Goal:

- Straight line at 45°

# Cox Model Diagnostics: Other Useful Things

## Functional Form

- Fit null model
- Obtain martingale residuals
- Smooth plot of residuals against covariate in question
- Examine nonlinearities, make appropriate transformation

## Outliers

- Examine DFBETAs (just as with regular regression)

# Cox Model Diagnostics: Other Useful Things

## Functional Form

- Fit null model
- Obtain martingale residuals
- Smooth plot of residuals against covariate in question
- Examine nonlinearities, make appropriate transformation

## Outliers

- Examine DFBETAs (just as with regular regression)

*Let's try this in Stata...*

# Parametric Models

## Quick recap of non- and semi-parametric

- Compare subjects at times when failures happen to occur
- Emphasis on the *order* of failure times
- No assumptions about distribution of failure times

## Parametric

- More efficient → uses information over entire time interval for each record
- Distributional assumptions required, generating two general classes of models:
  - Proportional hazards models
  - Accelerated failure-time models



# Parametric Proportional Hazards Models

$$\ln(t_j) = (\mathbf{x}_j\boldsymbol{\beta}_x) + \ln(\tau_j), \text{ where } \ln(\tau_j) = \exp(-\mathbf{x}_j\boldsymbol{\beta}_x)t_j$$

- $h_0(t) = pt^{p-1}\exp(a)$ 
  - Weibull model with parameters  $a$  and  $p$
- Model constant: baseline hazard when all covariates = 0
- Proportional hazards models directly comparable to Cox regression
  - Perform this comparison to validate parameterization of baseline hazard

# Parametric Accelerated Failure Time Models

$$\ln(t_j) = (\mathbf{x}_j\beta_x) + \ln(\tau_j), \text{ where } \ln(\tau_j) = \exp(-\mathbf{x}_j\beta_x)t_j$$

- Acceleration parameter,  $p$  ( $\exp(-\mathbf{x}_j\beta_x)$ ) describes failure times
  - if  $p = 1$ , time passes at "normal" rate
  - if  $p > 1$ , time is *accelerated*; failure expected to occur sooner
  - if  $p < 1$ , time is *decelerated*; failure expected to occur later

# Parametric Accelerated Failure Time Models

$$\ln(t_j) = (\mathbf{x}_j\beta_x) + \ln(\tau_j), \text{ where } \ln(\tau_j) = \exp(-\mathbf{x}_j\beta_x)t_j$$

- Exponentiated coefficients interpreted as time ratios
  - factor by which expected time-to-failure is multiplied for a unit change in corresponding covariate
- Specify `tr` option to get exponentiated coefficients
- Some parameterizations have both PH and AFT interpretations (e.g., Weibull) → specify `time` option to get AFT

# Which Modeling Approach is "Best"?

## Parametric is preferred to Cox if:

- You have a general sense of what the baseline hazard looks like AND
- You want to leverage that knowledge to:
  - Obtain most efficient estimates of  $\beta_x$
  - Obtain an estimate of  $h_0 t$  subject to that constraint

# Which Modeling Approach is "Best"?

## Parametric is preferred to Cox if:

- You have a general sense of what the baseline hazard looks like AND
- You want to leverage that knowledge to:
  - Obtain most efficient estimates of  $\beta_x$
  - Obtain an estimate of  $h_0 t$  subject to that constraint

## Which Parametric Parameterization: PH vs. AFT?

- PH if interested in the actual risk process behind failure and how risks changes with covariates
- AFT if explicit interest in analysis time (e.g., predicting failure times)

## Selecting the Appropriate Parameterization

Distribution	Metric	Hazard shape	c
Exponential	PH, AFT	constant	1
Weibull	PH, AFT	monotone	2
Gompertz	PH	monotonic	2
Lognormal	AFT	variable	2
Loglogistic	AFT	variable	2
Gen. gamma	AFT	variable	3

- Theoretically: Think about the data generating process!
- Statistically
  - Use likelihood-ratio test or Wald test for nested models
  - Compare Akaike information criterion (AIC) for non-nested models: smaller is better!

# Parametric Modeling Diagnostics

Same general suite of tools available for parametric:

- Model specification tests
- Goodness-of-fit
- Functional form
- Outliers & curve plotting

# Recap

## A Suggested Workflow for Survival Analysis in Stata:

- Begin with a well-structured dataset
  - Identification variable
  - Time variables for start and end of each observation period
  - Indicator variable for failure(s)
  - Covariates
- Declare survival data
  - `stset`
  - Diagnostics with `stdescribe`, `stvary`



# Recap

Workflow continued. . .

- Modeling Choices
  - Non-Parametric: No distributional assumptions, No covariates
  - Semi-Parametric: No distributional assumptions, Yes covariates
  - Parametric: Yes distributional assumptions, Yes covariates
- Model Diagnostics
  - Specification, goodness-of-fit, curve plotting, outliers
  - Parameterization tests (if parametric modeling)
- Revisit and Refine Models

# References

- Mario Cleves et al. *An Introduction to Survival Analysis Using Stata*. College Station: Stata Press, 2010.