

Duration Analysis In Stata

Kevin Sweeney
Assistant Director, Political Research Lab

Based On: *An Introduction to Survival Analysis Using Stata*

We Will Cover:

1. Overview – Stata and “Shape” of Survival Data
2. ST-Setting and Describing Your Data
3. Nonparametric Analysis: Kaplan-Meier
4. Parametric Models (Exponential, Weibull...), and post-estimation
5. The Cox Proportional Hazards Model, and post-estimation

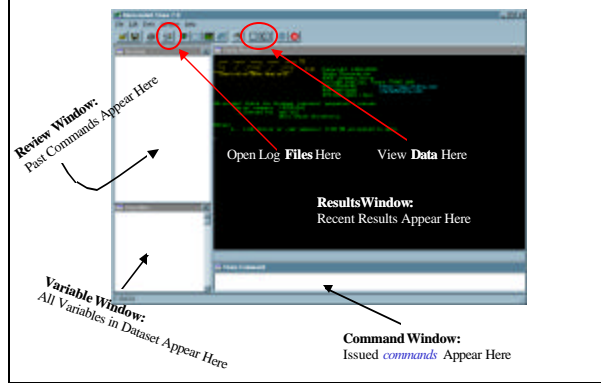
Me, the lab, vet of this class.

Today I will be talking about duration/survival/hazard analysis in stata. The presentation will be based on a recent book from the stata folks on this subject. In practice there is much more in the book than I am going to cover in this hour, so it is worth picking up (or at least having your department pick up). Ohio State folks should know we have a copy of this book in the lab.

I've selected subjects to cover from all over the book, seeking to cover a broad swath of material rather than cover anything in depth. I'm going to focus on command syntax, and not interpretation (that is what you are taking the class for). I'm also going to discuss subjects from all over your syllabus, so some of what I am talking about may not make sense yet – but it will be useful by the end of the course.

I'll begin with a brief overview of the program for those of you who have never used it before. I will then quickly transition to the particular shape of survival data. I'm sure you talked about this in your first class, but I will go over it again under the context of ST setting your data in stata. You have to tell stata you have duration data before you can analyze it. We will then go through the three main areas of duration analysis, and how to do them in stata: Nonparametric estimation, Parametric models, and the Cox PH model (Semi-parametric). When I discuss those last two points I will also talk about some of the various post estimation commands that are included in stata.

When You Open Stata...



If you've never opened stata before, this is what it looks like. 4 window.

Review. Finite, useful.

Variables. All, and Labels.

Command. Issue here, when I issue below they will be in blue italics. Hit enter, then this command is executed, and goes to the review window.

Results. Most recently obtained results here. Very finite.

So you will want to open a log to record all of your results, that can be done here.

The one window you cannot see, but may want to is your data window. These two buttons take you to those (left is editor, right is browser).

ST Setting Your Data

The basic syntax is `stset time_of_failure_or_censoring_variable, failure(one_if_failure_variable)`

So, if we had data that looked like this:

We'd type `stset failtime`

<u>failtime</u>	<u>x</u>
1	3
5	2
9	4
20	9
22	10

<u>lastime</u>	<u>x</u>	<u>failed</u>
1	3	1
5	2	1
9	4	1
20	9	1
22	10	0

If we had data that looked like this, we'd type...

`stset lastime, failure(failed)`

_t0 & _t – record time span
_d – records outcome
_st – records whether the observation is relevant.

Before doing any survival analysis in Stata, you must tell the program that you have survival data. Stata requires this for three reasons. First, you tell stata about the shape of the data at the beginning and you do not have to tell it again. Second, stata checks to see if the claims you make about your data are true. For instance, is time really time, and are observations all at risk until failure. Third, and most importantly, is to allow you to make complicated rules for your survival data revolving around risk and failure. For instance, maybe there can be repeated events (an observation can fail more than once) or competing risks (an observation can fail in more than one way). Some examples might be useful.

First the simplest case. One variable recording time and an independent variable. “Stset” is the command, and failtime is the name of the variable that tells us how much time until that observation failed.

Second, a more complicated case where not every observation failed, note the final observation is censored. Here lastime is the variable that describes the amount of time until the observation failed or was right censored, and a dummy variable indicates whether the observations failed.

Every time you stset your data, stata adds four variables to the dataset. _t0 and _t record the time span the observation lived, with t units for each observation. _d records the outcome at the end of that span, 1 if the span ends in failure, 0 otherwise. Finally, _st records whether the observation is relevant, that is, can be included in the study. If the observation existed before the specified origin date of the study, for instance, _st would be zero. What is the origin of the study? Well, there are lots of options to the stset command that may never come up, but it is quite powerful.

ST Setting Your Data, Important Options

If you have more than one record per subject you must tell Stata what the id variable is...

stset lasttime, failure(failed) id(name)

<u>name</u>	<u>lasttime</u>	<u>x</u>	<u>failed</u>
Bob	1	3	1
Bob	5	2	1
Jim	9	4	1
Jim	20	9	1
Jim	22	10	0

<u>name</u>	<u>lasttime</u>	<u>x</u>	<u>event</u>
Bob	1	3	7
Bob	5	2	9
Jim	9	4	6
Jim	20	9	7
Jim	22	10	9

You can also tell Stata a certain kind of event is a failure, whereas others are not...

stset lasttime, failure(event==9) id(name)

There are many important options with ST set. A few of them are vital.

First, if you have multiple observations per subject (time varying covariates) you must tell stata which variable is the ID variable. Otherwise it will treat each observation as distinct.

You can also tell stata a certain kind of event was a failure, your failure variable does not have to be a dummy. This is important if you have data that ends in a number of ways, but only one of them is an actual failure.

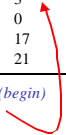
The == sign is a stata expression meaning “equal to,” “less than or equal to” is written <=, and “greater than or equal to” is written >=.

ST Setting Your Data, One More Option

Finally (well, not really), you can tell Stata when your observations begin, if you don't Stata will do it for you...

<u>name</u>	<u>begin</u>	<u>lasttime</u>	<u>x</u>	<u>event</u>
Bob	0	1	3	7
Bob	3	5	2	9
Jim	0	9	4	6
Jim	17	20	9	7
Jim	21	22	10	9

stset lasttime, failure(event==9) id(name) time0(begin)



Now, on to a real live example...

Finally, it is possible (or even likely) that you will know when your observation of each subject begins and ends. If this is the case you should tell Stata when you `stset` the data. You do this with the `time0` option.

There are lots of other things you can tell Stata when you `stset` the data. These options will come after the comma in the `stset` command. You should look them up when you get a chance – some of them will be necessary depending on what type of data you have. But for the example I am going to use today, these commands are sufficient.

Canned Hip Fracture Data

use <http://www.stata-press.com/data/cgg/hip2>

Type: *describe* to see what you have

```
Contains data from http://www.stata-press.com/data/cgg/hip2.dta
obs:      106                hip fracture study
vars:     12                30 Jan 2002 17:58
size:     2,332 (96.7% of memory free)
```

variable name	storage type	display format	value label
id	byte	%4.0g	patient id
time0	byte	%5.0g	begin of span
time1	byte	%5.0g	end of span
fracture	byte	%8.0g	fracture event
protect	byte	%8.0g	wears device
age	byte	%4.0g	age at enrollment
calcium	float	%8.0g	blood calcium level

You can play along at home with this data, it was canned by the authors of the manual and placed on the stata website. You can get it by typing this command.

The data is described in the book, let me read the paragraph description to you, it will be good to know as we move through the examples. Read 80-1.

A good idea at the outset is to issue a describe command. This will give you the basic details of each variable. This is particularly a good idea for us since we've never seen this data before and know nothing about hip fractures. The most useful pieces of information here are the variable names and labels. We have 4 variables we will use to ST set the data – id time0, time1, and fracture. And, we have three variables that appear to be covariates that we could use in analysis of the data.

Canned Hip Fracture Data

```

. sort id time0
. by id: list time0-calcium
> id = 1
time0  time1  fracture  protect  age  calcium
1.    0      1          1         0  76    9.35
--> id = 2
time0  time1  fracture  protect  age  calcium
2.    0      1          1         0  80    7.8
--> id = 3
time0  time1  fracture  protect  age  calcium
3.    0      2          1         0  74    8.8
--> id = 18
time0  time1  fracture  protect  age  calcium
27.    0      5          0         0  64   11.58
28.   15     17          1         .    .   11.59
--> id = 19
time0  time1  fracture  protect  age  calcium
29.    0     10          0         0  72    9.49
30.   10     15          0         .    .    9.46
31.   15     22          1         .    .     9

```

Begin → (points to time0 of id=1)

End → (points to time1 of id=1)

Fail → (points to fracture of id=2)

Some Xs → (points to fracture of id=3)

Notice: Some subjects have multiple observations to incorporate the time varying covariate calcium { (points to the group of subjects 27-31)

After describing the data, we can take a look at it. So, we first sort the data by ID and Time0. This is because we want all of the subject to be in order (hence sorting by ID) and we want all of the observations for each subject, assuming there are more than one for some subjects, to bin in correct temporal order.

We can then list the data. Alternatively, we could go into the data editor and look at the data itself, but list is useful if you want to see only some of the variables in the dataset. I've placed here the data from 5 subjects we can see when have a beginning time variable, and an ending time variables. These could be any units of time, but here they are months. We also see we have a dummy variable indication fractures (our failures), and we have 3 Xs.

Notice that the first three subjects have only one observation each, but the next two have more than one. This is because calcium is a time varying covariate.

ST Setting the Data

Type: *stset time1, id(id) time0(time0) failure(fracture)*

And this is what you get...

```
      id: id
failure event: fracture == 0 & fracture == .
obs. time interval: (time0, time1]
exit on or before: failure
-----
      106 total obs.
         0 exclusions
-----
      106 obs. remaining, representing
         48 subjects
      31 failures in single failure-per-subject data
      714 total analysis time at risk, at risk from t = 0
              earliest observed entry t = 0
              last observed exit t = 39
```

This may not make a lot of sense, Stata has a more descriptive command...

OK, so let's go ahead and `stset` this data. We know we have four variables that we can use to tell Stata what is going on. As you remember the base of the command is `stset` and we know that the variable that marks the end of the time span is called `time1`. After the comma, we know we have subjects with multiple observations, so we have to tell Stata which variable is the `id` variable. We also know there is a variable in the data that marks the beginning of the time span, `time0`, so we should tell Stata that as well. Finally, we know that the failures we are interested in assessing are hip fractures, so we tell Stata that the failures are indicated by the variable `fracture`.

When we type that in, we get this `st set` report. It is of limited use, but if something is awry Stata will display a message that says **PROBABLE ERROR** here. When you get this message, which we did not here, you should double check to make sure everything is OK. Stata is often right when you are wrong.

stdes

Type: *stdes* and we see that...

```
failure_d: fracture
analysis time_t: time1
id: id

-----|----- per subject -----|
Category      total      mean      min      median     max
-----|-----|-----|-----|-----|
no. of subjects      48
no. of records      106      2.208333
(first) entry time      0      0      0      0
(final) exit time      15.5      1      12.5     39
subjects with gap      3
time on gap if gap      30      10      10      10
time at risk      714      14.875
failures      31      .6458333      0      1      1
```

Stdes is a command that provides a brief description of the data you have just stset.

The first thing to notice is that stata tells you about some of the key variables in the dataset (some of which it may have set itself). In this case we named the failure, analysis time and id variables so stata confirms their name.

The first two lines tell us that there are 48 subjects in the data and there are a total of 106 records, with the average number of records per subject being a little over two. Remember, this is because we have a time varying covariate in the data. These two lines confirm that stata correctly recognizes the set up of the data we have st set.

Lines 3 and 4 report that everyone entered at time 0 (there is no delayed entry) and that the average exit time was 15.5 months. This is not the average survival time because some of our subjects are censored.

Lines 5 and 6 tell us that there are 3 subjects in the data that have gaps – in this case each of those gaps are ten months long. What this likely means is that the researchers lost touch with the subjects or that the subjects did not meet the criteria for the study for some period of time. If you have gaps in your data you should know where they are and why they are there.

Line 7 reports that the subjects were at risk for a total of 714 months. This is simply the sum of the time spanned by the records (or the length of time from *_t0* to *_t*).

Finally line 8 shows that there were 31 failures (broken hips) among the subjects, since that is less than the total number of subjects we know some of them did not fail while under investigation – hence, they were censored. Line 8 also shows that the maximum number of failures per subject was one. That means that this is not repeated events data.

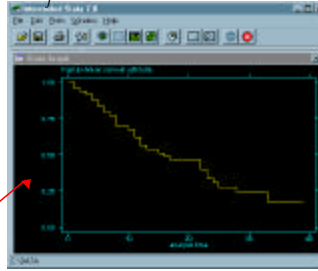
Nonparametric Analysis: Kaplan-Meier

$$\hat{S}(t) = \prod_{j: t_j \leq t} \left(\frac{n_j - d_j}{n_j} \right) \quad \begin{array}{l} \text{-the probability of survival past time } t, \text{ or} \\ \text{the probability of failing after time } t. \end{array}$$

Where n_j is the number of individuals at risk at t_j
and d_j is the number of failures at t_j .

Type *sts graph*

And you get the simple
Kaplan-Meier graph



The estimator of Kaplan and Meier (1958) is a nonparametric estimate of the survivor function $s(t)$, or the probability of survival past time t . In our dataset it is given by this equation. The stata book on survival analysis has a nice straightforward explanation of how this is calculated, and so do most textbooks, so I will not go into it here.

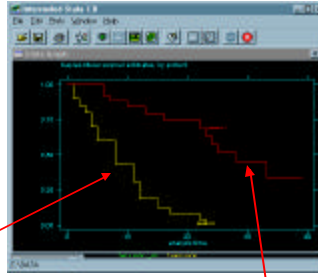
What I want to demonstrate is that this is relatively easy to get stata to do, simply type *sts graph* and stata knows to graph the Kaplan – Meier. Here it is. While you won't get published if this is where you end, your chances of getting published may increase if this is where you start. The Kaplan-Meier is a good way to see the basic shape of your survival data. *Sts* is the root command in stata to graph, list, and generate survival data. You must have your data *stset* before issuing it..

Nonparametric Analysis: Kaplan-Meier

We can make the graph a little more complicated by comparing those in the treatment group with those in the control group.

Use the *by* command to plot multiple survival curves...

Type *sts graph, by(protect)*

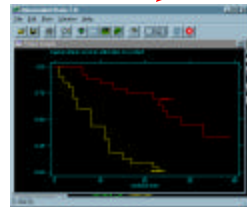


Those without the protection fail more quickly than those with it.

We can also make the graph a little more complicated by using the *by* command to plot multiple Kaplan-Meier curves. We could use this to begin to test a simple hypothesis. Remember one of the covariates in the study was this inflatable protective device which is indicated in the data with the dummy variable *protect*. Presumably, the key hypothesis for whoever is using this data is that this protective device decreases the hazard of hip fractures.

If we type *sts graph, by(protect)* we will get two survival lines, one for those with the protective device (the treatment group) and one for those who are not wearing the protective device (the control group). The folks who have the device are in the red line, and the folks without it are represented by the yellow line. Call the New England Journal of medicine, we've got a winner. The folks without protection fail more quickly.

Simple Nonparametric Tests



Here's our Kaplan-Meier graph from The last slide.

Type: *sts test protect, logrank*

```
failure_d: fracture
analysis time _t: time1
id: id
```

Log-rank test for equality of survivor functions

protect	observed	expected
---------	----------	----------

0	19	7.14
---	----	------

1	12	23.86
---	----	-------

Total	31	31.00
-------	----	-------

chi2(1) = 29.17

Pr>chi2 = 0.0000

$H_0: h_1(t) = h_2(t)$

Rejected

We can add a little statistical certainty to this finding with a number of tests that stata has canned for use with nonparametric estimators. One such test is the logrank test. Here's the Kaplan-Meier graph from the last slide, if we type `sts test protect, logrank` stata will return the logrank test. The log rank test tests the null hypothesis that that two groups have the same hazard of failure with a series of k contingency tables (where k is the number of distinct failure times). Stata returns a summary of those contingency tables.

You might want to take this to a journal, but I doubt you'll have much luck. The reviewers will probably want you to assess your main hypothesis in the face of some control variables. To do that you need to specify a duration model. Stata can handle a wide range of such models. I will break them up into parametric and semiparametric models, just like they are broken up on your syllabus.

From the results here, we can see that for those not wearing the device we observed 19 hip fractures, but expected only a little over 7. For those wearing the protective device we observed 12 failures but expected almost 24 (there are twice as many observations with the protective device than without). The stata performs a chi squared test of the observed vs. expected. It is clear that we can reject the null hypothesis that both groups face the same hazard of failure.

Parametric Models - Exponential

Type: streg age protect, dist(exp) nohr

The command for all Parametric Models

The covariates in this model.

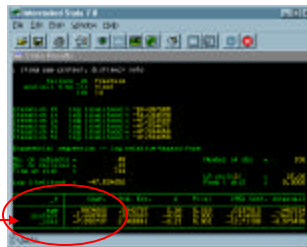
Specify the parameterization of the baseline hazard.

Tells Stata you want coefficients and not hazard ratios => must exponentiate.

$$h(t) = \exp(\mathbf{b}_0 + \mathbf{b}'_x X_i)$$

$$\hat{h}_i(t) = \exp(-7.89) = 0.00037$$

$$\hat{h}(x_i) = 0.00037 \exp(1.69 \text{protect} - 0.08 \text{age})$$



So far, with our nonparametric tests, we have established that patients with the protective device seem to fracture their hips at a different rate than patients without the protective device. We can surmise from the previous figure that fractures are much more common among non-protective device subjects, but how much more common? Are there other covariates, other than the protective device that is, that affect the hazard rate? To answer these questions we would want to specify a model. These fall into two general classes. The first class contains parametric models. Here we pick a parameterization of the baseline hazard rate (the rate at which subjects will fail if all covariates are equal to zero). If we have good theory on the shape of this hazard rate, which we often do not, this is a powerful estimation tool.

Stata can estimate a number of parametric regressions, I will consider the two most commonly used in political science. First, the exponential model parameterizes the baseline hazard rate as being constant over time. That is, subjects fail at the same rate through time and the hazard function is a flat line. We can estimate such a model with our canned hip data, adding the age of the subject as a relevant control variable. Presumably, older subject are more likely to break their hips.

We can estimate such a model with this command. Perhaps the first thing to notice, for those of you who have used stata before to estimate other types of regressions, is that you do not have to specify a dependent variable. In fact, those four variables that stata added to your dataset after you stset it have taken care of this for you. Stata knows the time span of the event and what constitutes a failure. The rest of the command contains first the independent variables you want in the model. A comma separates that section of the command from the specification of the distribution of the hazard rate, which in this case is exponential, and the no hazard ratio command – which tells stata you want coefficients and not hazard ratios. We will see what happens when we do not specify this command below.

There are a number of other options that stata includes for estimating parametric regressions. You should look these up on your own time. A couple that you will probably use in this class is time (which estimates the model as an accelerated failure time model) and strata(varname) which stratifies the sample by the variable you name.

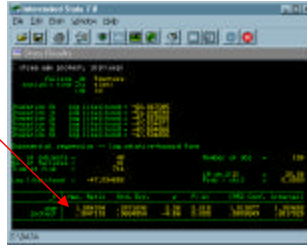
Parametric Models - Exponential

Or, you could estimate the model and get hazard ratios...

Type: `streg age protect, dist(exp)`

Remember the coefficient on age was .0809663,
 $e^{.0809663} = 1.084334$

Hazard ratios have the virtue of being relatively easy to interpret.



If we did not specify the `nohr` command, we would get hazard ratios rather than coefficients. There is a direct relationship, of course, between the estimated coefficients and the hazard ratios.

Hazard ratios have the virtue of being relatively easy to interpret. Those greater than one, like age, increase the hazard of failure. Those less than one decrease the hazard. In this case each year in age increases the hazard of breaking your hip by a about 8 ½ percent – something to look forward to as we get older. My advice would be to get one of these protective devices because people who wear them are over 80% less likely to break their hip.

Parametric Models, postestimation

median time	predicted median survival time; the default
median lntime	predicted median ln(survival time)
mean time	predicted mean survival time
mean lntime	predicted mean ln(survival time)
hazard	predicted hazard
hr	predicted hazard ratio
xb	linear prediction
stdp	standard error of the linear prediction
surv	predicted $S(\text{depvar})$ or $S(\text{depvar} t0)$
csnell	(partial) Cox-Snell residuals
mgale	(partial) martingale-like residuals
deviance	deviance residuals
csurv	predicted $S(\text{depvar} \text{earliest } t0 \text{ for subject})$
ccsnell	cumulative Cox-Snell residuals
cmgale	cumulative martingale-like residuals

After estimating a parametric model there are a number of post-estimation commands you can issue. Those of you who have used stata to do other types of models will be familiar with the basic syntax it is predict varname, command. Some of these commands are not very useful, to be honest. Others are, you can use, for instance the mgale post-estimation command to generate martingale residuals, which are used to test the proportional hazards assumption in these parametric models. You should check out these commands on your own time, but I will give one example.

A Post-estimation Example

After our most recent regression (the Weibull) we could Type:

```
predict cc, ccsnell  
graph cc _t, s(fid)
```



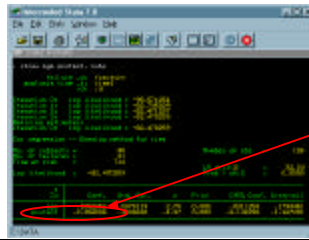
We will consider some more post-estimation commands with the Cox Model

We could look for outliers in the data with the cumulative cox-snell residuals. This example shows you the command structure I alluded to above. When you issue the predict command stata will generate a new variable in your dataset named cc. We can then issue a graph command to graph the residual against time, specifying that we want stata to list the id numbers rather than dots or circles. Actually, I picked this one because it looked cool, but we can see that there are three subjects that are clearly outliers and may warrant further investigation.

The Semiparametric Cox Proportional Hazards Model

$$h(t | x) = h_0(t) \exp(\mathbf{b}'_k x_i)$$

Type: *stcox age protect, nohr*



$e^{2.257} = .105$

The Cox Proportional hazard model allows the researcher to leave the baseline hazard unparameterized. That is, we need not make an assumption about the shape of the hazard over time. This is nice, because in the social sciences we often do not have enough theory to make a strong case that the baseline hazard is constant or increasing or decreasing.

Here the hazard rate takes this form. Where $h_0(t)$ is the baseline hazard function – which is left unspecified, and the x_i 's are the covariate values and the beta are their coefficients.

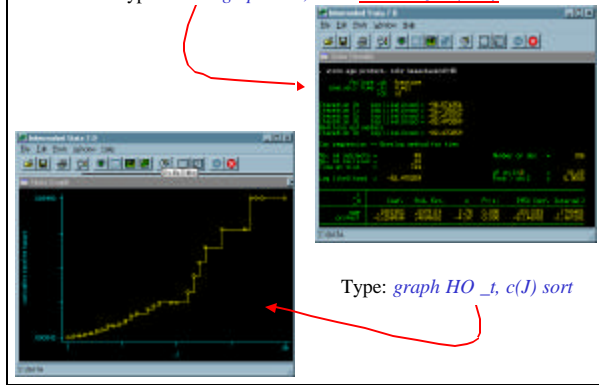
We can estimate the same model we have been estimating for our canned hip fracture data using the `stcox` command. And when we do we get the following results. Since we specified the no hazard ratio option, we must exponentiate these coefficients in order to meaningfully interpret them – or we could specify the model without the no hazard ratio command. When we exponentiate the coefficient on `protect` we get a value of .105, which basically means that a person who wears the protective device has a hazard of breaking their hip that is only 10% of the hazard for those who do not wear the device.

Note also that the Cox model does not have a constant term, it is absorbed into the baseline hazard.

You are going to discuss the cox model in depth, so I will not elaborate on it here, but I will talk about some extensions to that model that you may find useful.

Extensions to the Cox Model

Type: `stcox age protect, nohr basechazard(HO)`



Type: `graph HO _t, c(J) sort`

While the Cox model does not parameterize the baseline hazard function, we can use the model to estimate the baseline cumulative hazard based on its results. I note this, first, because it can be done, and second because when we estimated the weibull model there was some evidence that the baseline hazard rate was increasing over time.

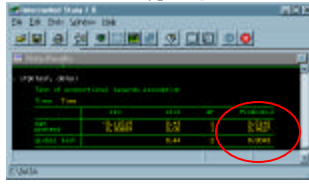
We add the `basechazard` command to the model, and specify a new variable name - HO. Stata will add a variable to your dataset of that name with the model's estimate of the baseline hazard for each observation. We can then graph this over time, and indeed it does look like the hazard of breaking your hip does increase over time.

It is important to note that although this is something you would do after estimation, you must tell stata that you want to do it when you estimate the model.

Testing the Proportional Hazards Assumption: Schoenfeld Residuals

Type: *stcox age protect, schoenfeld(sch*) scaledsch(sca*)*

After estimation type: *stphtest, detail*



We find no evidence that the model violates the PH assumption.

The term proportional hazards refers to the effect of any covariate having a proportional and constant effect that is invariant to when in the process the values of the covariate changes. As Jan will explain in the coming weeks, all duration models make a proportional hazards assumption. This should be tested, just like one should test for heteroskedasticity in an ols regression. There are a number of ways to do this in stata after estimating a cox 'proportional hazards' model. One is a test based on Schoenfeld residuals. Here you must specify the appropriate variables in the original model so that stata records them. While this is a postestimation test, there is stuff you have to do before estimation to carry it out. The command `schoenfeld` saves residuals that you can use for a global model test, the command `scaledsch` saves residuals you need for a variable by variable test. You can name these variables, which will be added to your data set, anything you want, but you must name them. Each will actually generate two variables, the * is stata programming code that lets stata name the variables for you. They will be 1 and 2, 1 and 2.

After estimation we can type `stphtest, detail` which will test whether the model violated the PH assumption both globally and with respect to each covariate. There is no evidence that the model we specified violates the PH assumption. (160-2)

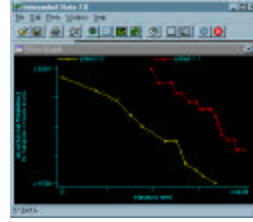
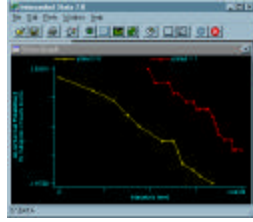
Testing the Proportional Hazards Assumption:

Stata's Plots: *stphplot*

stphplot estimates $-\ln[-\ln\{S(t)\}]$ vs. $\ln(t)$ for each level of the specified covariate.

Type: *stphplot, by(protect)*

Type: *stphplot, by(protect) adjust(age)*



Parallel lines means model has not violated PH assumption.

There are two graphical methods by which to test the proportional hazards assumption that are canned in stata. One is *stphplot*. This is a postestimation command for the Cox model. This plots a function of the estimate of the Kaplan-Meier estimate of the survivor function against the log of time, it is intended for use with discrete covariates, so it is perfect to test our key protect variable. First, type *stphplot, by(protect)* – if the lines are parallel we have not violated the PH assumption. Looks pretty good. While we cannot directly test age in this manner, because it is not discrete, we can test whether the effect of protect is constant conditional on age with the *adjust* command. Again, the lines look parallel, so we are good.