Duration Analysis In Stata

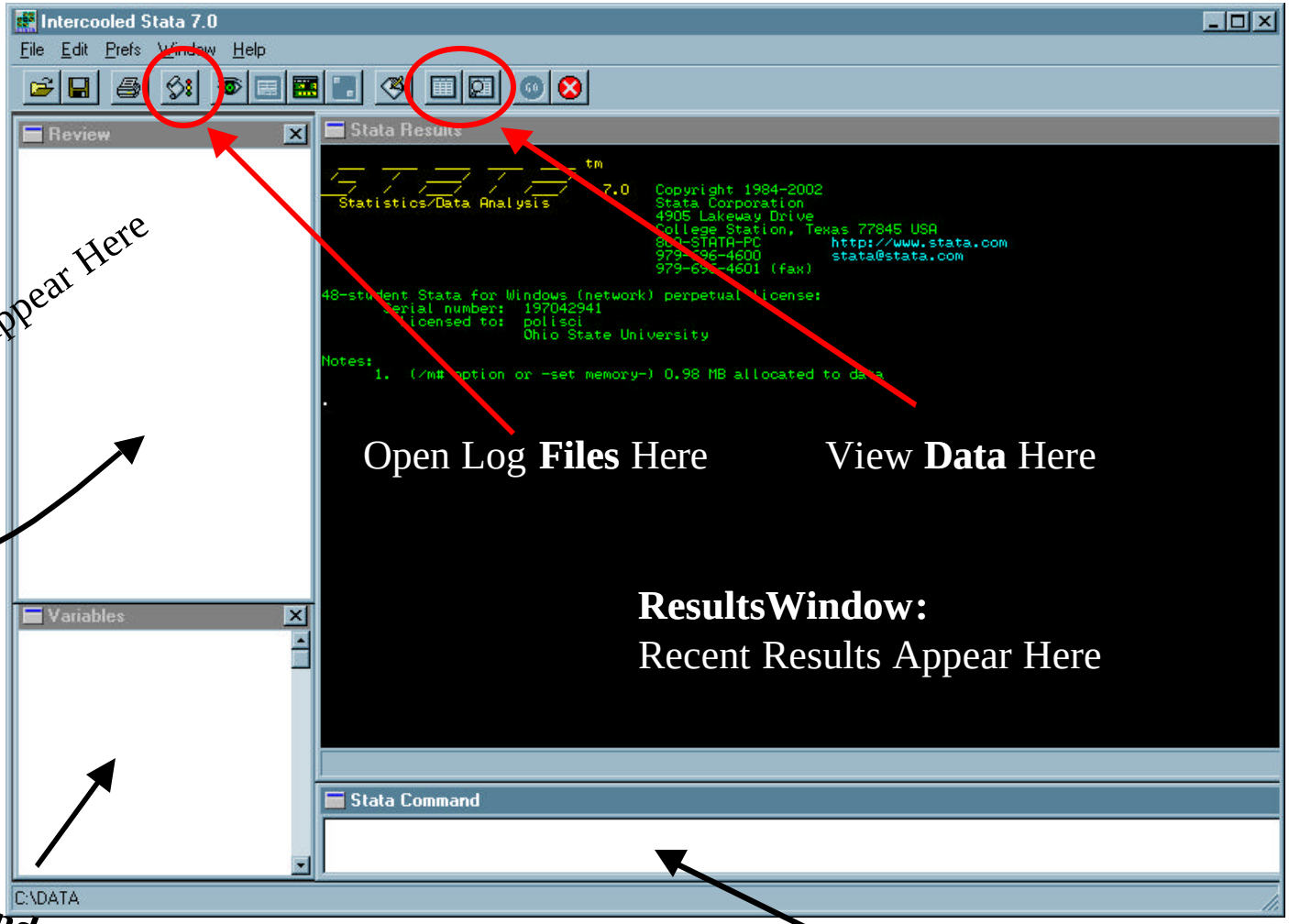
Kevin Sweeney
Assistant Director, Political Research Lab

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

When You Open Stata…



Review Window:
Past Commands Appear Here

Open Log **Files** Here

View **Data** Here

ResultsWindow:
Recent Results Appear Here

Variable Window:
All Variables in Dataset Appear Here

Command Window:
Issued *commands* Appear Here

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*

failtime	x
1	3
5	2
9	4
20	9
22	10

lasttime	x	failed
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 lasttime, failure(failed)

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

ST Setting Your Data, Important Options

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

tset lasttime, failure(failed) id(name)

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

name	lasttime	x	event
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

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…

name	begin	lasttime	x	event
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…

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)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable 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
-------------------------------------------------------------------------------
```

Canned Hip Fracture Data

Begin

End

Fail

Some Xs

```
. 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
```

Notice: Some subjects have multiple observations to incorporate the time varying covariate calcium

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…

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           1           2           3
(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          10
time at risk                714      14.875           1        11.5          39
failures                     31    .6458333           0           1           1
-------------------------------------------------------------------------------
```

Nonparametric Analysis: Kaplan-Meier

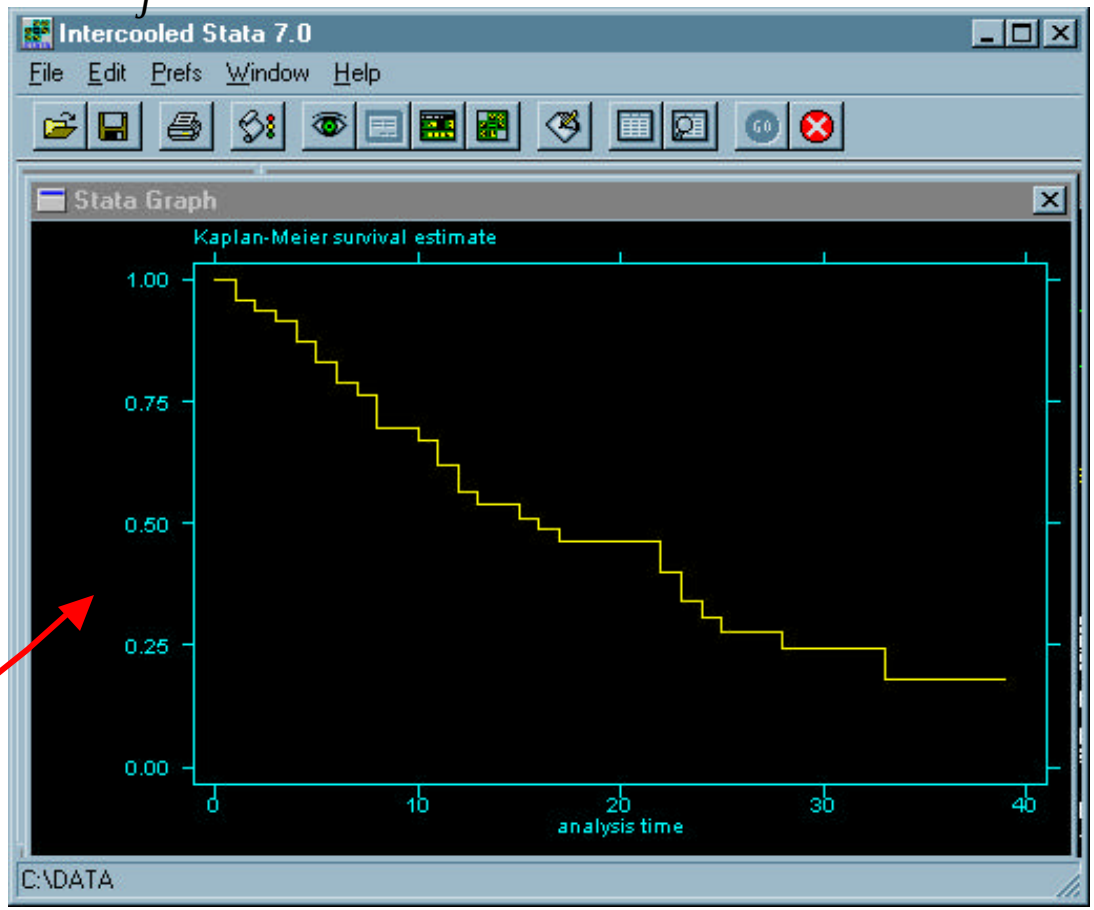
$$\hat{S}(t) = \prod_{j|t_j \le t} \left(\frac{n_j - d_j}{n_j}\right)$$

-the probability of survival past time *t*, or the probability of failing after time *t*.

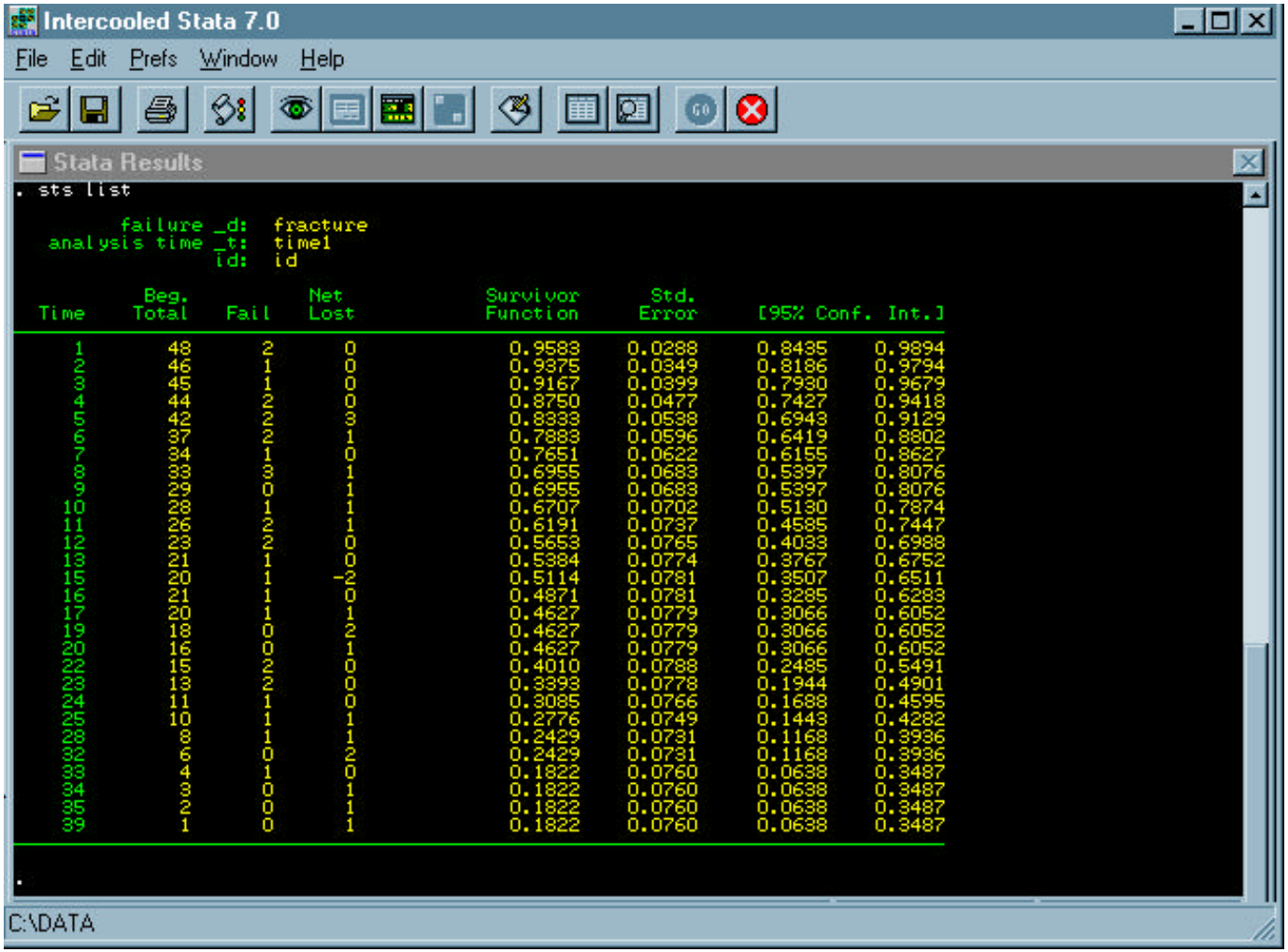
Where n_j is the number of individuals at risk at t_j and d_j is the number of failures at t_j.

Type *sts graph*

And you get the simple Kaplan-Meier graph

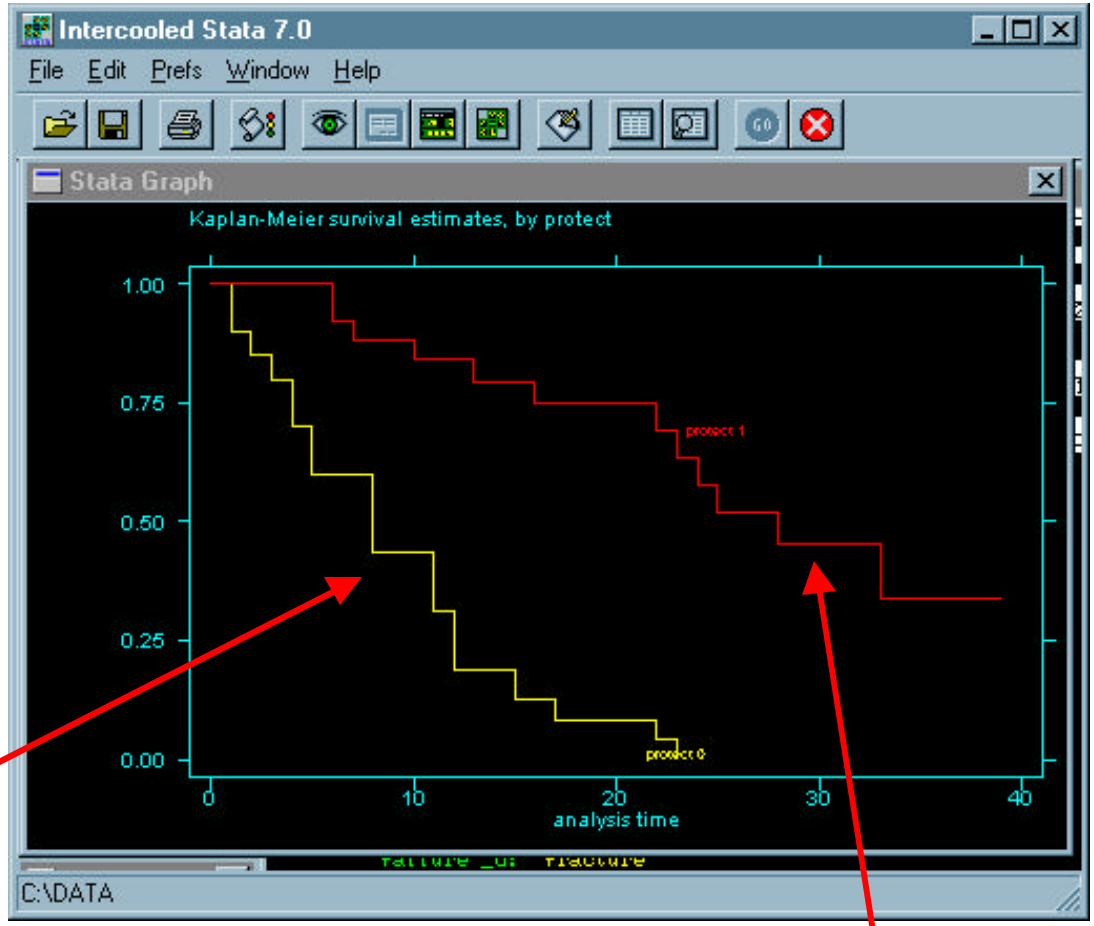


STS List


Intercooled Stata 7.0
File Edit Prefs Window Help
Stata Results
. sts list
failure _d: fracture
analysis time _t: time1
id: id
Time | Beg. Total | Fail | Net Lost | Survivor Function | Std. Error | [95% Conf. | Int.]
1 | 48 | 2 | 0 | 0.9583 | 0.0288 | 0.8435 | 0.9894
2 | 46 | 1 | 0 | 0.9375 | 0.0349 | 0.8186 | 0.9794
3 | 45 | 1 | 0 | 0.9167 | 0.0399 | 0.7930 | 0.9679
4 | 44 | 2 | 0 | 0.8750 | 0.0477 | 0.7427 | 0.9418
5 | 42 | 2 | 3 | 0.8333 | 0.0538 | 0.6943 | 0.9129
6 | 37 | 2 | 1 | 0.7883 | 0.0596 | 0.6419 | 0.8802
7 | 34 | 1 | 0 | 0.7651 | 0.0622 | 0.6155 | 0.8627
8 | 33 | 3 | 1 | 0.6955 | 0.0683 | 0.5397 | 0.8076
9 | 29 | 0 | 1 | 0.6955 | 0.0683 | 0.5397 | 0.8076
10 | 28 | 1 | 1 | 0.6707 | 0.0702 | 0.5130 | 0.7874
11 | 26 | 2 | 1 | 0.6191 | 0.0737 | 0.4585 | 0.7447
12 | 23 | 2 | 0 | 0.5653 | 0.0765 | 0.4033 | 0.6988
13 | 21 | 1 | 0 | 0.5384 | 0.0774 | 0.3767 | 0.6752
15 | 20 | 1 | -2 | 0.5114 | 0.0781 | 0.3507 | 0.6511
16 | 21 | 1 | 0 | 0.4871 | 0.0781 | 0.3285 | 0.6283
17 | 20 | 1 | 1 | 0.4627 | 0.0779 | 0.3066 | 0.6052
19 | 18 | 0 | 2 | 0.4627 | 0.0779 | 0.3066 | 0.6052
20 | 16 | 0 | 1 | 0.4627 | 0.0779 | 0.3066 | 0.6052
22 | 15 | 2 | 0 | 0.4010 | 0.0788 | 0.2485 | 0.5491
23 | 13 | 2 | 0 | 0.3393 | 0.0778 | 0.1944 | 0.4901
24 | 11 | 1 | 0 | 0.3085 | 0.0766 | 0.1688 | 0.4595
25 | 10 | 1 | 1 | 0.2776 | 0.0749 | 0.1443 | 0.4282
28 | 8 | 1 | 1 | 0.2429 | 0.0731 | 0.1168 | 0.3936
32 | 6 | 0 | 2 | 0.2429 | 0.0731 | 0.1168 | 0.3936
33 | 4 | 1 | 0 | 0.1822 | 0.0760 | 0.0638 | 0.3487
34 | 3 | 0 | 1 | 0.1822 | 0.0760 | 0.0638 | 0.3487
35 | 2 | 0 | 1 | 0.1822 | 0.0760 | 0.0638 | 0.3487
39 | 1 | 0 | 1 | 0.1822 | 0.0760 | 0.0638 | 0.3487
C:\DATA

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.

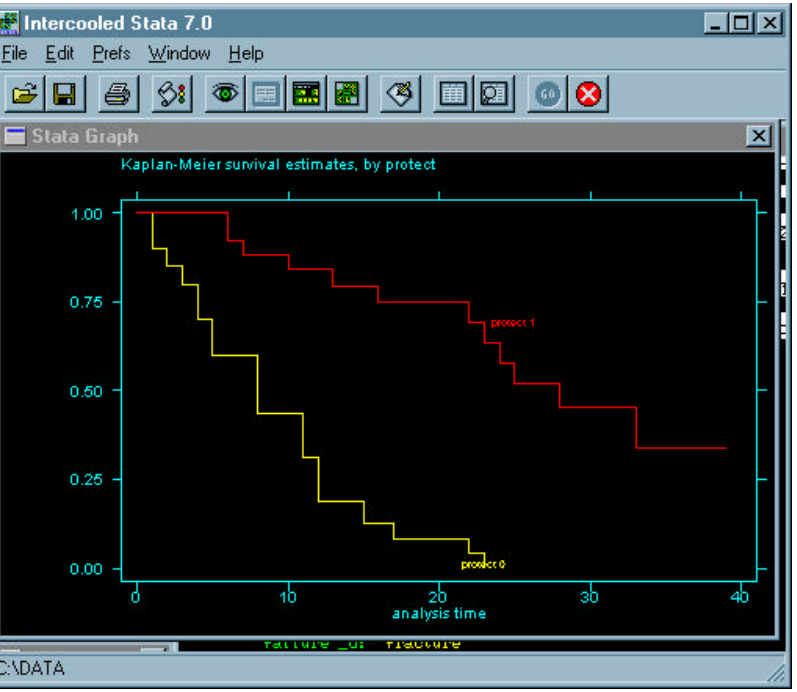


Use the *by* command to plot multiple survival curves…

Type *sts graph, by(protect)*

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

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

        |   Events          Events
protect |  observed        expected
--------+-------------------------
0       |        19            7.14
1       |        12           23.86
--------+-------------------------
Total   |        31           31.00
          chi2(1) =       29.17
          Pr>chi2 =      0.0000
```

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

Rejected

Parametric Models - Exponential

Type: *streg age protect, dist(exp) nohr*

The command for all Parametric Models

The covariates in this model.

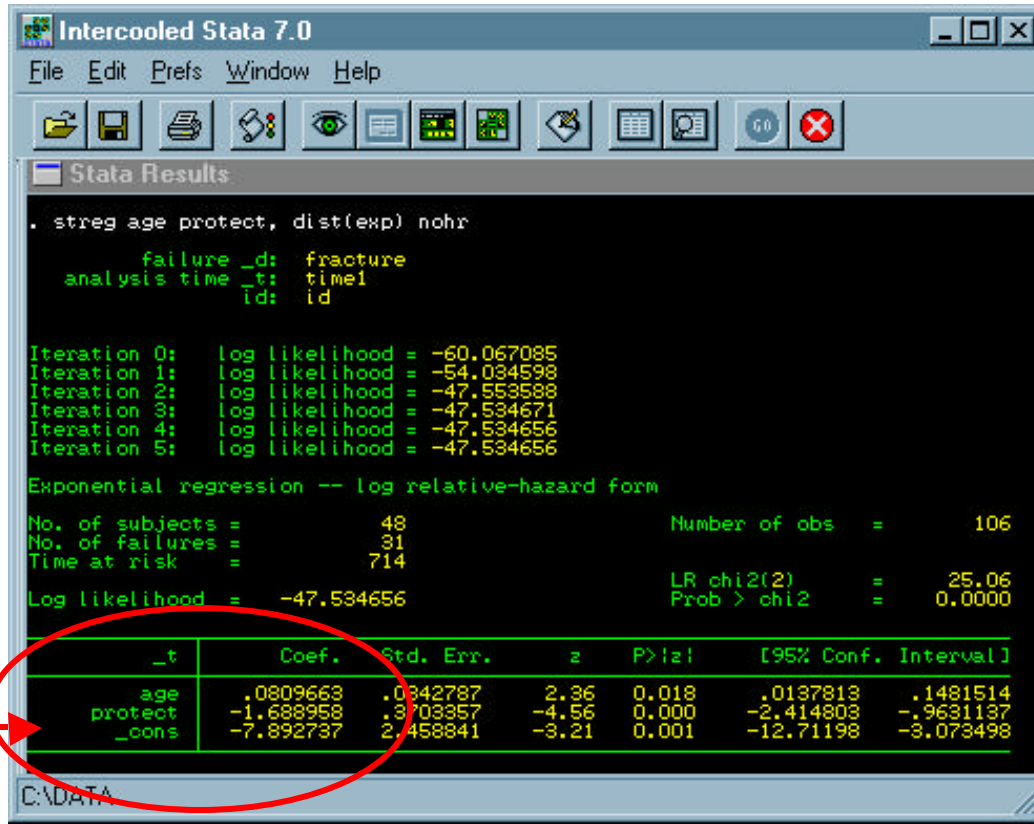
Specify the parameterization of the baseline hazard.

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

$$h(t) = \exp(b_0 + b_k' X_i)$$

$$h_0(t) = \exp(-7.89) = .00037$$

$$h(t \mid x_j) = .00037 \exp(-1.69 protect_j + .08 age_j)$$



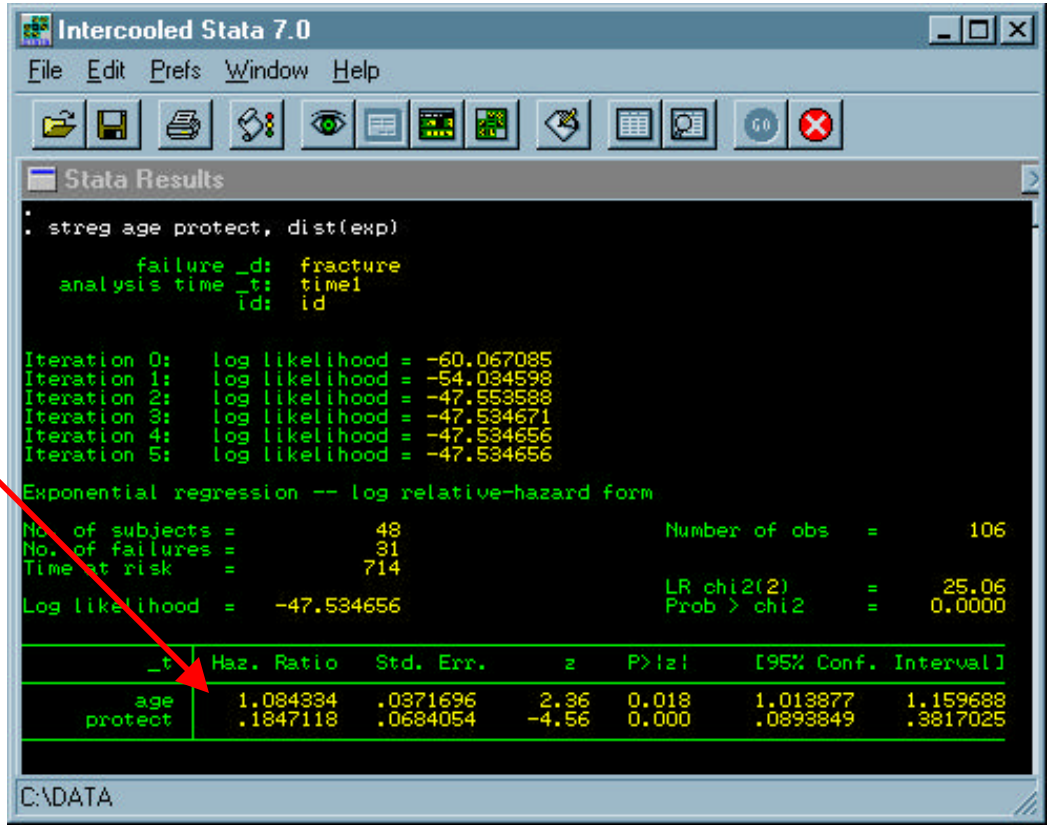
Parametric Models - Exponential

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.



Parametric Models - Weibull

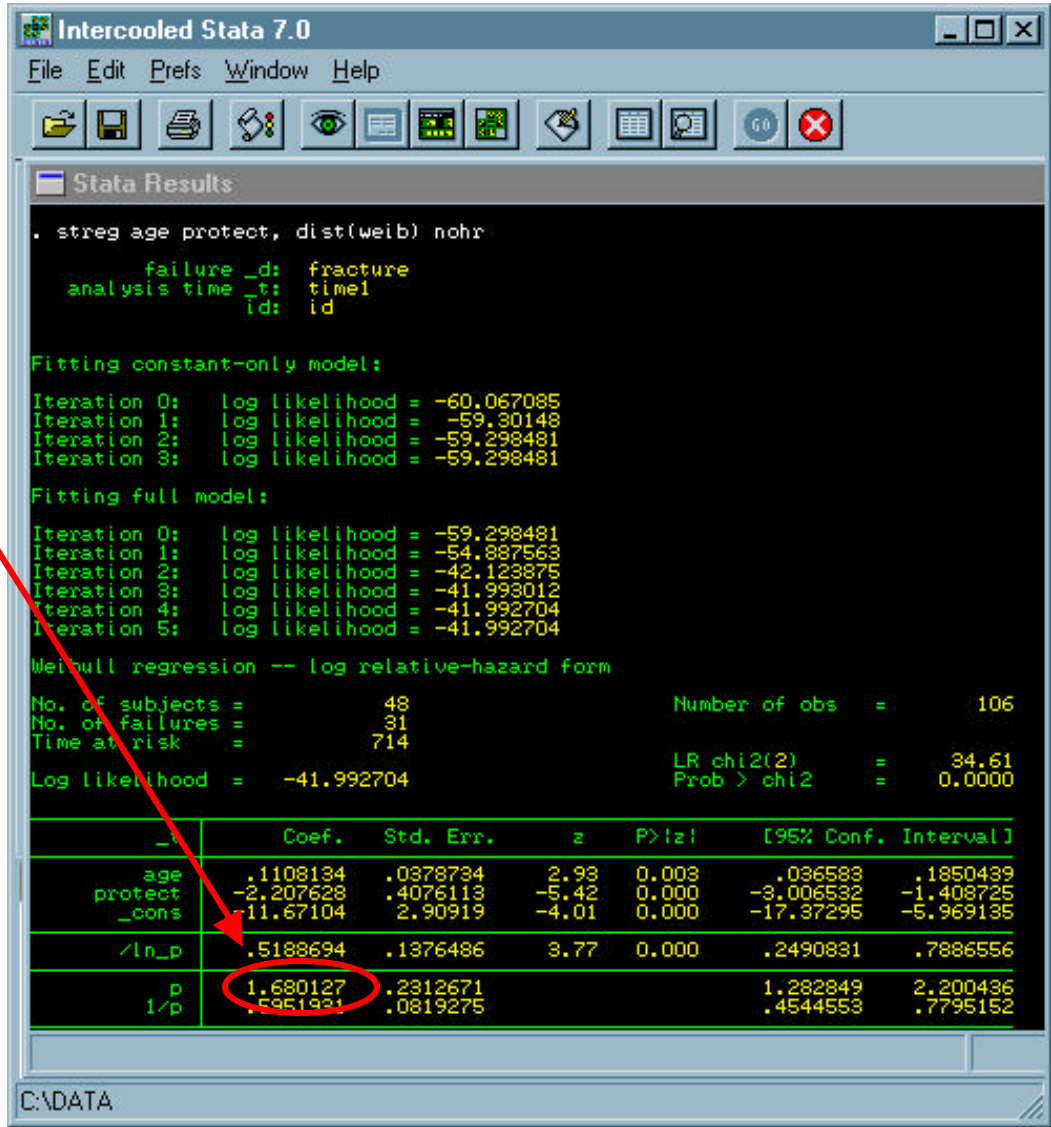
Type: *streg age protect, dist(weib) nohr*

Here, Stata estimates the shape of the hazard function with *p*.

p>1 indicates the hazard is monotonically increasing.

p<1 indicates the hazard is monotonically decreasing.

Gompertz, lognormal, loglogistic, gamma



Parametric Models, postestimation

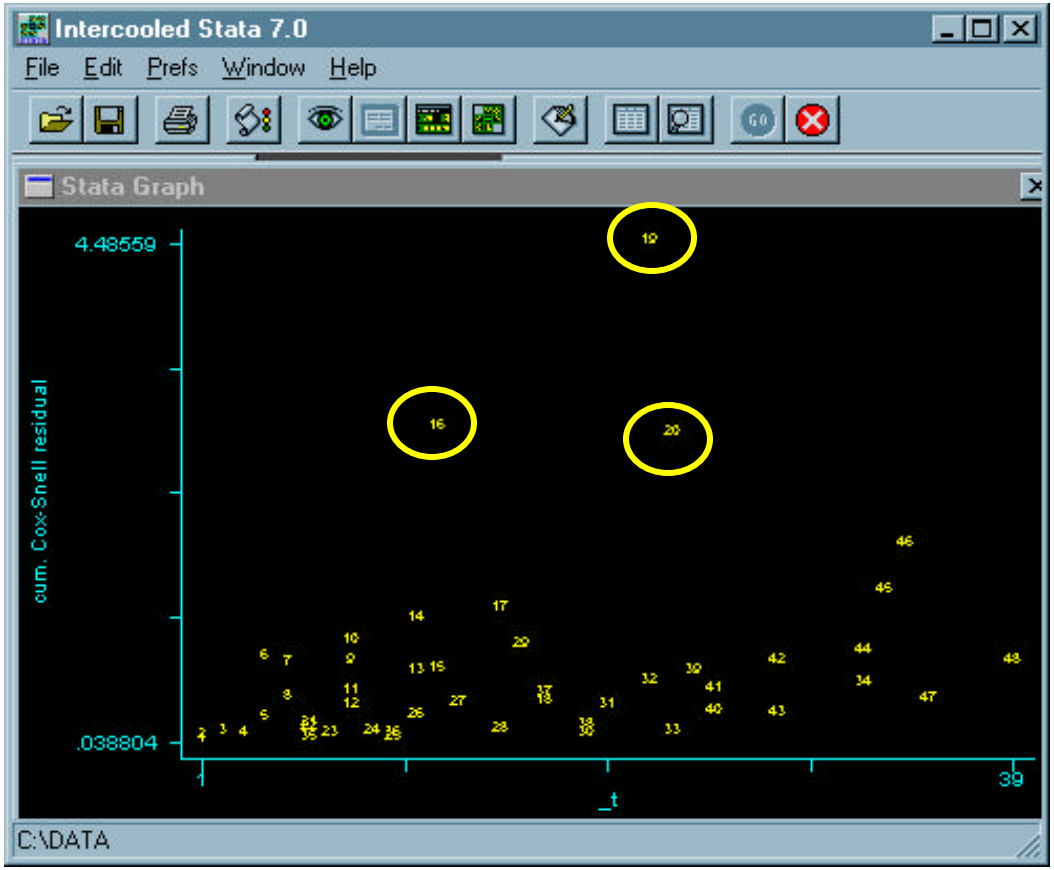
median time	**predicted median survival time; the default**
median lntime	**predicted median ln(survival time)**
mean time	**predicted mean survival time**
mean lntime	**predicted mean ln(survival time)**
hazard	**predicted hazard**
hr	**predicted hazard ratio**
xb	**linear prediction**
stdp	**standard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

A Post-estimation Example

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

predict cc, ccsnell

graph cc _t, s([id])

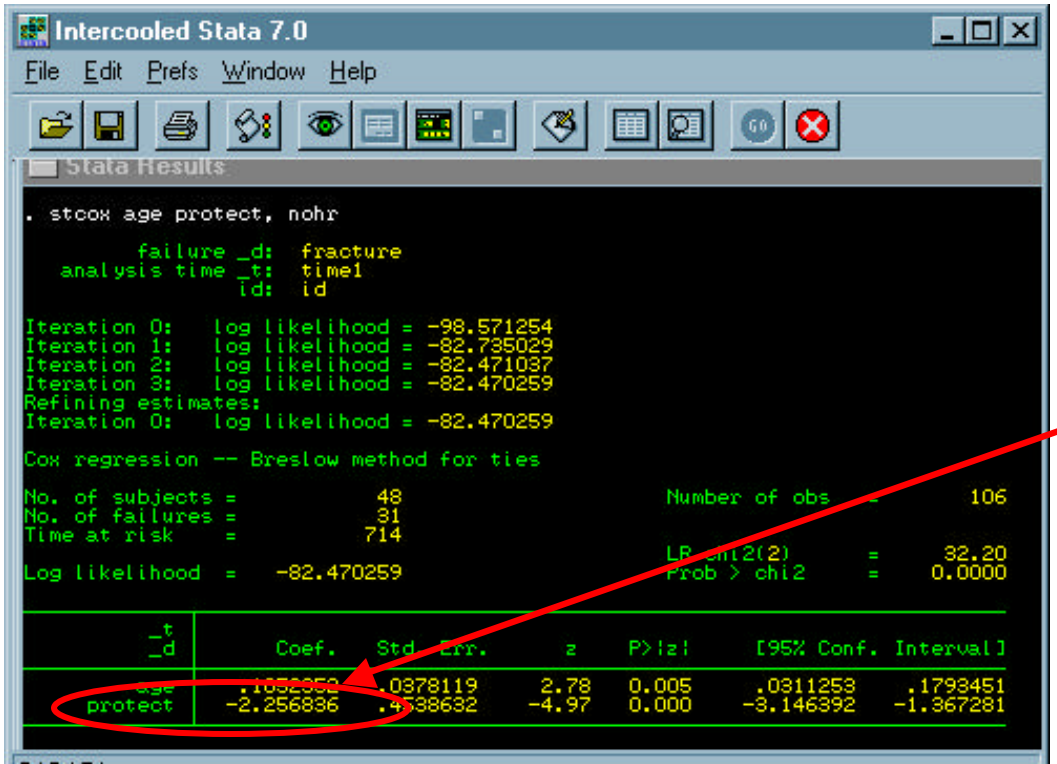


We will consider some more post-estimation commands with the Cox Model

The Semiparametric Cox Proportional Hazards Model

$$h(t \mid x) = h_0(t)\exp(b_k' x_i)$$

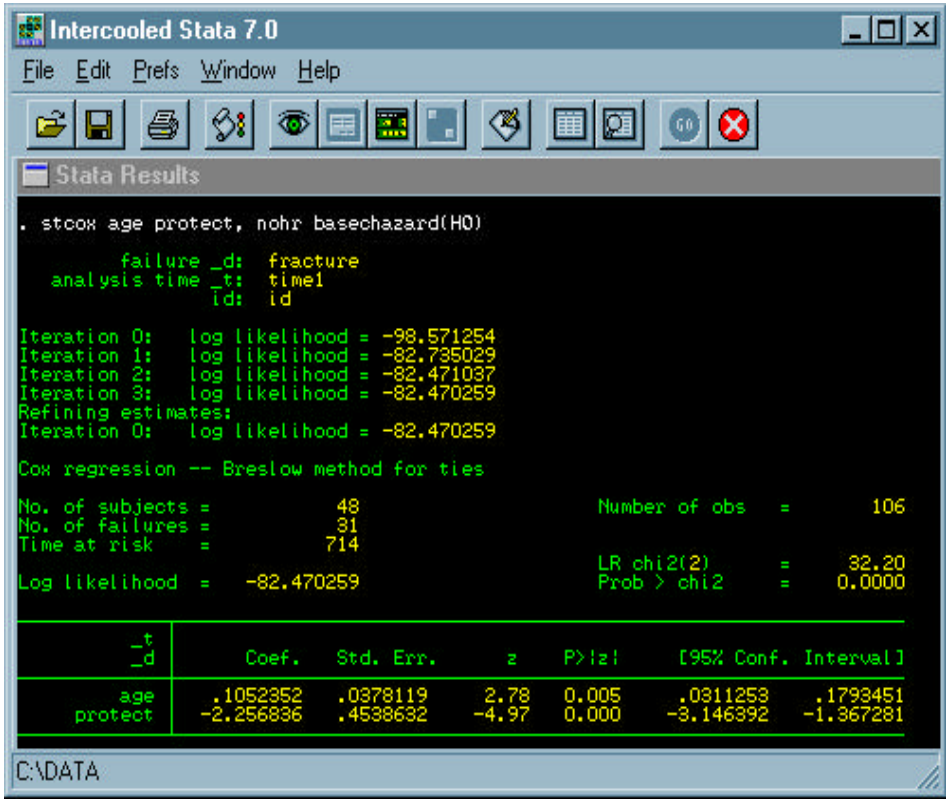
Type: *stcox age protect, nohr*

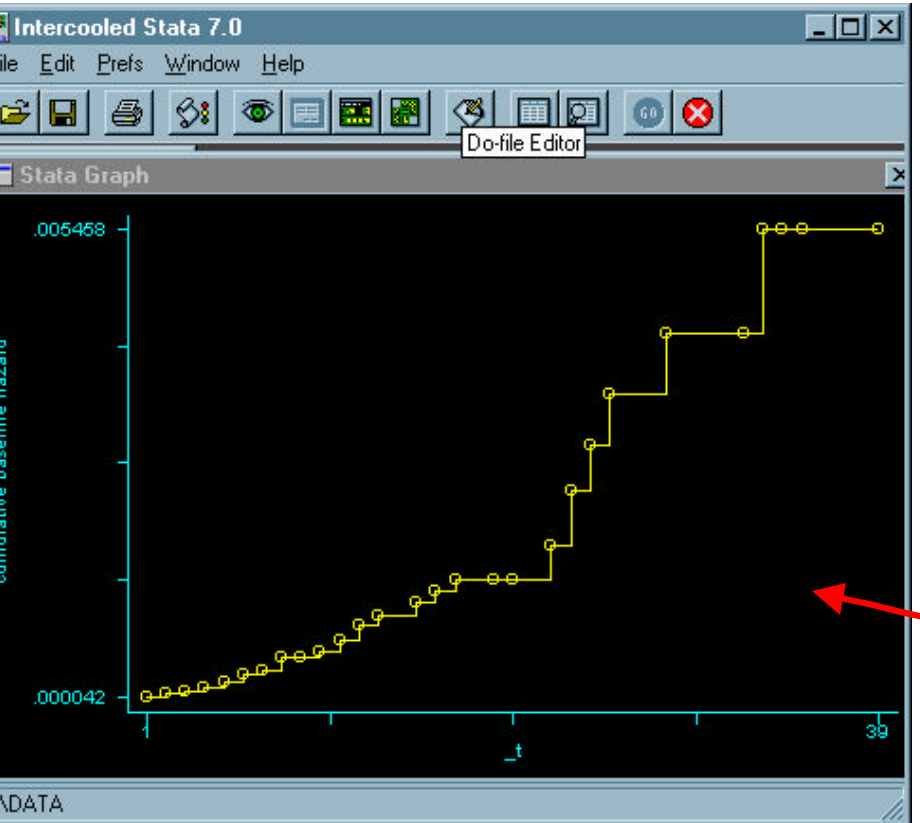


$e^{-2.257} = .105$

Extensions to the Cox Model

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



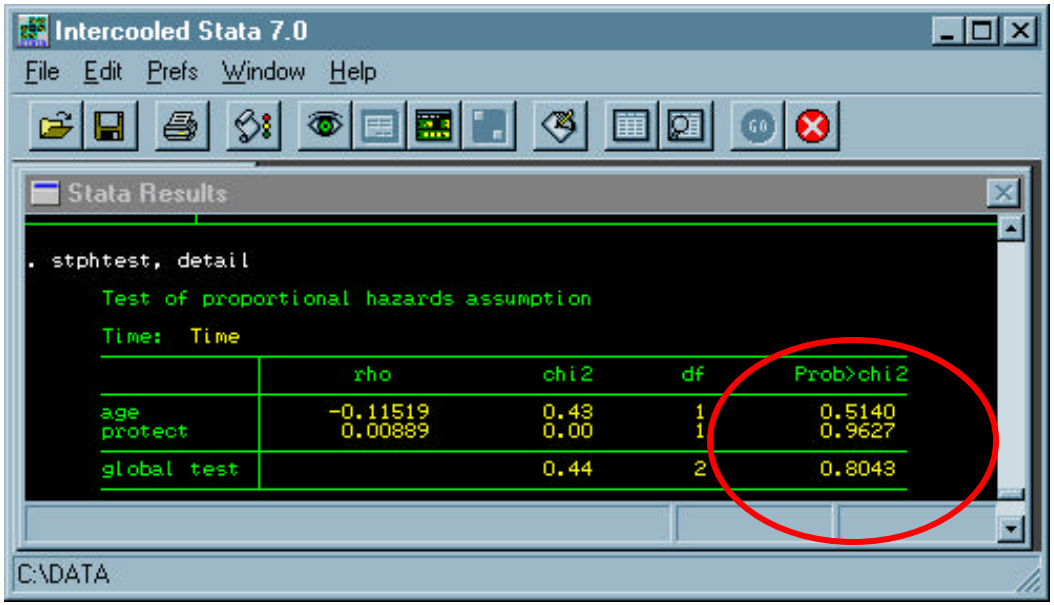


Type: *graph HO _t, c(J) sort*

Testing the Proportional Hazards Assumption: Schoenfeld Residuals

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

After estimation type: *stphtest, detail*

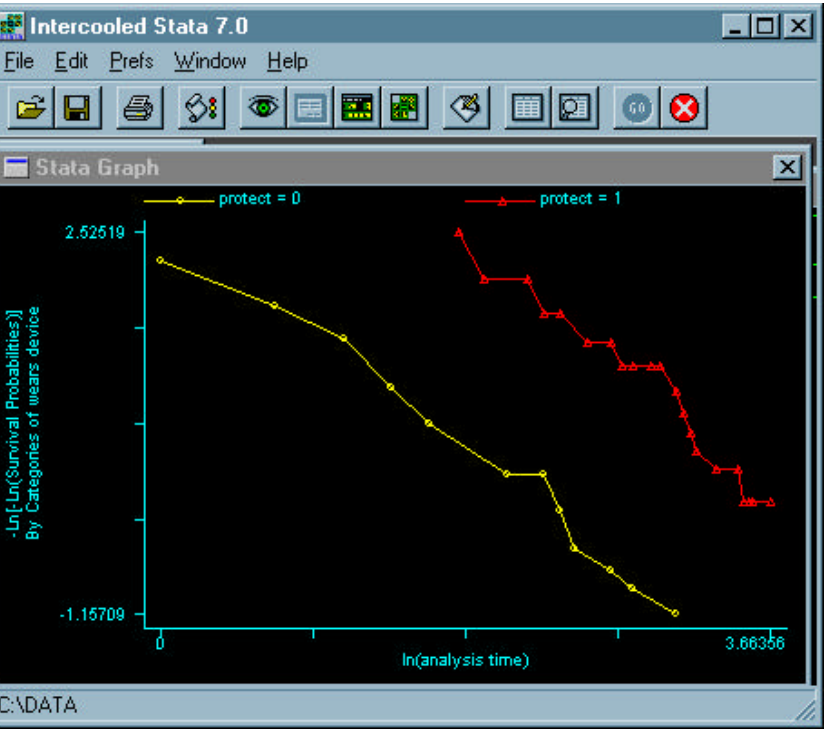


We find no evidence that the model violates the PH assumption.

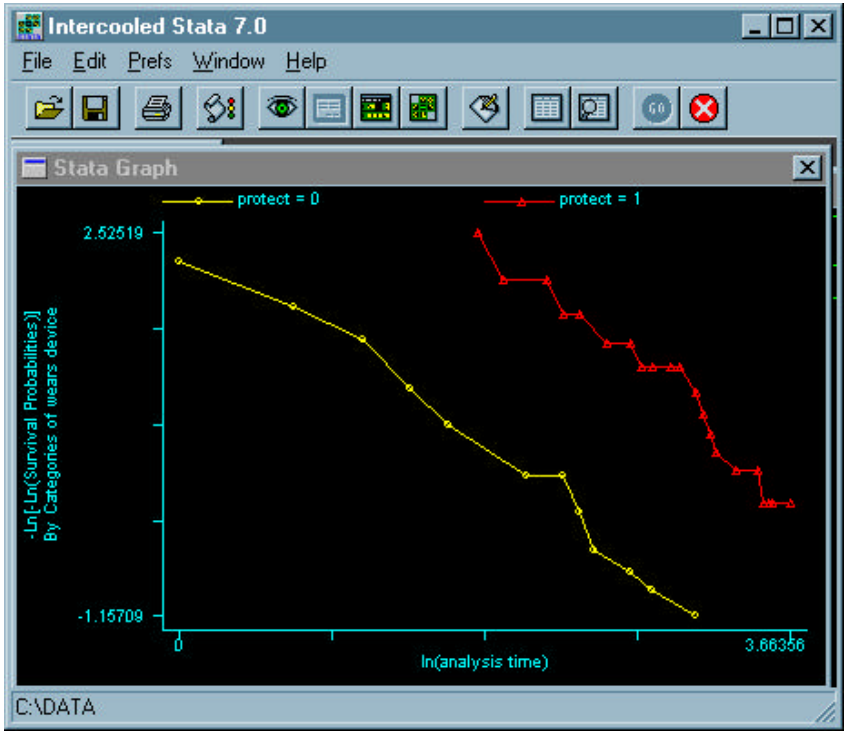
Testing the Proportional Hazards Assumption: Stata's Plots: *stphplot*

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.