UNIVERSITY OF ESSEX
INSTITUTE FOR SOCIAL AND ECONOMIC RESEARCH
Professor Stephen P. Jenkins <stephenj@essex.ac.uk>

Essex Summer School course 'Survival Analysis'
and
EC968. Part II: Introduction to the analysis of spell duration data

Lesson 4. Estimation of the (integrated) hazard and survivor functions: Kaplan-Meier product-limit and lifetable methods

Contents

1 Aim

The aim of this lesson is to illustrate how to use Stata to estimate (integrated) hazard and survival functions using Kaplan-Meier product-limit and lifetable methods. Again, the data are assumed to be the simple single-spell type considered in earlier Lessons (with right censoring but not left censoring or left truncation).

2 Introduction: sts and ltable

Estimation of the Kaplan-Meier empirical hazard and survival functions is done very easily in Stata by using either the **sts** collection of commands or by using the **ltable** ('lifetable') command.

Many of the differences between **sts** and **ltable** derive from the underlying assumptions about the nature of the survival time data. With **sts**, survival times are treated as observations on a continuous variable. In the **ltable** case, the technique is based on survival data that have been grouped into intervals (or implicitly assumed to be).

Observe that the continuous time hazard rate is defined with reference to an instant of time (and not a probability), whereas the discrete time hazard rate is a probability, the definition of which refers to an interval of time, by construction. One might try and estimate the continuous time hazard from the slope of the integrated hazard function (a step function): a discrete approximation based on the 'hazard contribution' – the change in the integrated hazard over the interval of time between the dates between two successive observed failure times. One might then estimate the hazard rate from the ratio of the hazard contribution to the length of the time interval. However this estimator is known to have relatively poor properties and this has led analysts to focus instead on estimates of a smoothed hazard rate.

The **sts** commands consist of **sts graph**, **sts list**, **sts generate**, and **sts test**. (See **help sts**.) The first two of these are likely to be of most use to you: they provide a graph and listing of the survivor function (estimated by the Kaplan-Meier product-limit method), the integrated hazard function (estimated by the Nelson-Aalen method) plus associated statistics such as standard errors. (Note that Stata refers to the integrated hazard as the 'cumulative hazard'.) The command **sts** used by itself is a synonym for **sts graph** (and understands all the standard options from **graph**). Estimates of smoothed hazard rates can be derived using **sts graph** and its **hazard** option. This uses kernel smoothing, and there are options to choose kernels and bandwidths other than the default ones.

The **sts** commands can only be used with data which has already been **stset** (see Lesson 3), but this has substantial pay-offs. Once the data have been correctly **stset**, then estimates can be derived very straightforwardly – regardless of whether the data are organised by person or by person-month. (**stset** takes care of any potential complications on that score. See Lesson 3.)

The **ltable** command displays and graphs estimates of survivor functions estimated using lifetable methods. It can be applied to data organised in either person or person-month form. Estimates of the empirical hazard can be shown in a table, but cannot be graphed. (This reflects the issues raised earlier about estimation of continuous hazard rates.) By default, **ltable** uses the so-called actuarial adjustment for the number of subjects at risk (this may vary within intervals if survival times are grouped). To get the unadjusted estimates, corresponding to the Kaplan-Meier assumption, which are also the ones produced by **sts list**, you need to use the **noadjust** option to **ltable**.

I will now work through an illustrative example based on the Cancer data. There are exercises at the end that ask you to work through the same material but with other course data sets.

3 Illustration using the Cancer data set: (i) sts

First we first **stset** the data (as in Lesson 3):

```
use cancer
su
de
stset studytim , failure(died)
ge id = _n
lab var id "subject identifier"
```

We also recode the drug variable (for use later)

```
. recode drug 1=0 2/3=1
(drug: 48 changes made)
. lab var drug "receives drug?"
. lab def drug 0 "placebo" 1 "drug"
. lab val drug drug
```

Now we simply:

```
. sts list

         failure _d:  died
   analysis time _t:  studytim
```

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	0	0.8333	0.0538	0.6943	0.9129
6	40	2	1	0.7917	0.0586	0.6474	0.8820
7	37	1	0	0.7703	0.0608	0.6236	0.8656
8	36	3	1	0.7061	0.0661	0.5546	0.8143
9	32	0	1	0.7061	0.0661	0.5546	0.8143
10	31	1	1	0.6833	0.0678	0.5302	0.7957
11	29	2	1	0.6362	0.0708	0.4807	0.7564
12	26	2	0	0.5872	0.0733	0.4304	0.7145
13	24	1	0	0.5628	0.0742	0.4060	0.6931
15	23	1	1	0.5383	0.0749	0.3821	0.6712
16	21	1	0	0.5127	0.0756	0.3570	0.6483
17	20	1	1	0.4870	0.0761	0.3326	0.6249
19	18	0	2	0.4870	0.0761	0.3326	0.6249
20	16	0	1	0.4870	0.0761	0.3326	0.6249
22	15	2	0	0.4221	0.0786	0.2680	0.5684
23	13	2	0	0.3572	0.0788	0.2087	0.5083
24	11	1	0	0.3247	0.0780	0.1809	0.4771
25	10	1	1	0.2922	0.0767	0.1543	0.4449
28	8	1	1	0.2557	0.0753	0.1247	0.4093
32	6	0	2	0.2557	0.0753	0.1247	0.4093
33	4	1	0	0.1918	0.0791	0.0676	0.3634
34	3	0	1	0.1918	0.0791	0.0676	0.3634
35	2	0	1	0.1918	0.0791	0.0676	0.3634
39	1	0	1	0.1918	0.0791	0.0676	0.3634

The second column (beg. total) is the total number of subjects at risk of failure (death) at the time shown in the first column. The third column (Fail) shows the number dying at each time. The fourth column (Net lost) gives the number of subjects censored (and thence no longer entering the risk set). The estimates of the survivor function together with estimates of their statistical significance are shown in the remaining columns.

The table shows that 48% of the sample remained alive after t = 17 (with a 95 percent. confidence interval of 0.33, 0.62). You could change the confidence interval shown if you wished using the level option. (This applies to all Stata's estimation commands: see **help level**.)

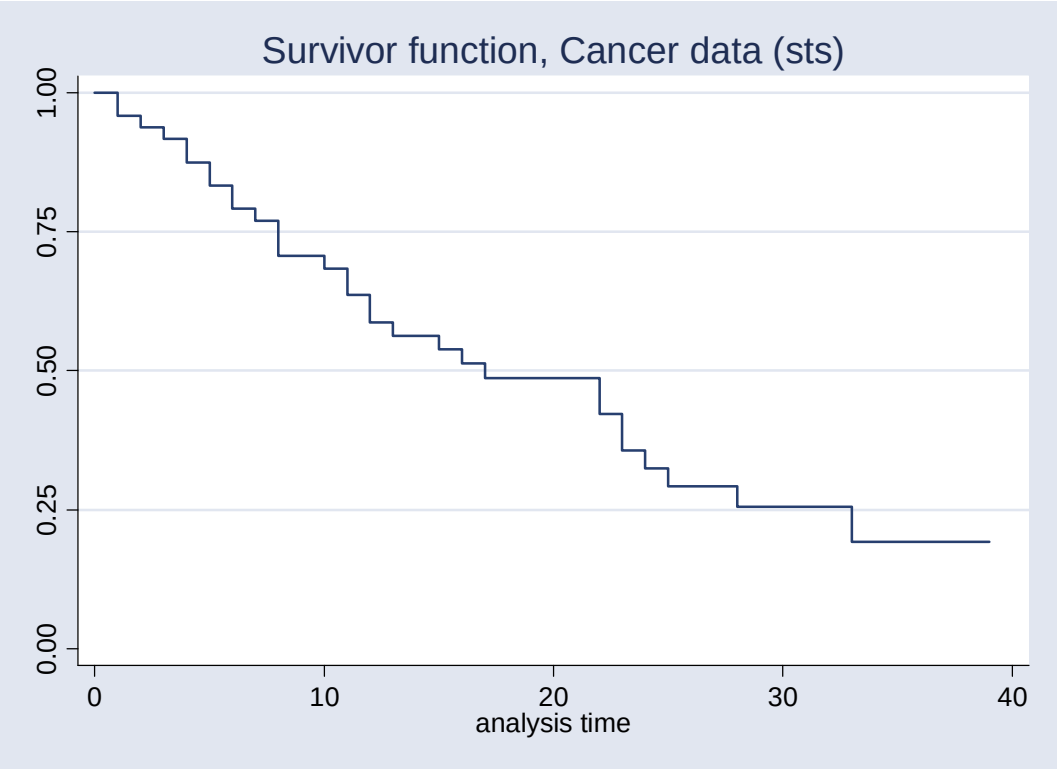
Our estimate of the median duration is t between 16 and 17. At the end of the (fictional) drug trial which provided the data, only about one fifth of the sample remained alive (four-fifths

had died). We can see this directly from a graph of the survivor function derived from **sts graph**:

```
. sts graph, title("Survivor function, Cancer data (sts)") saving(surv1, replace)

        failure _d:  died
  analysis time _t:  studytim
```

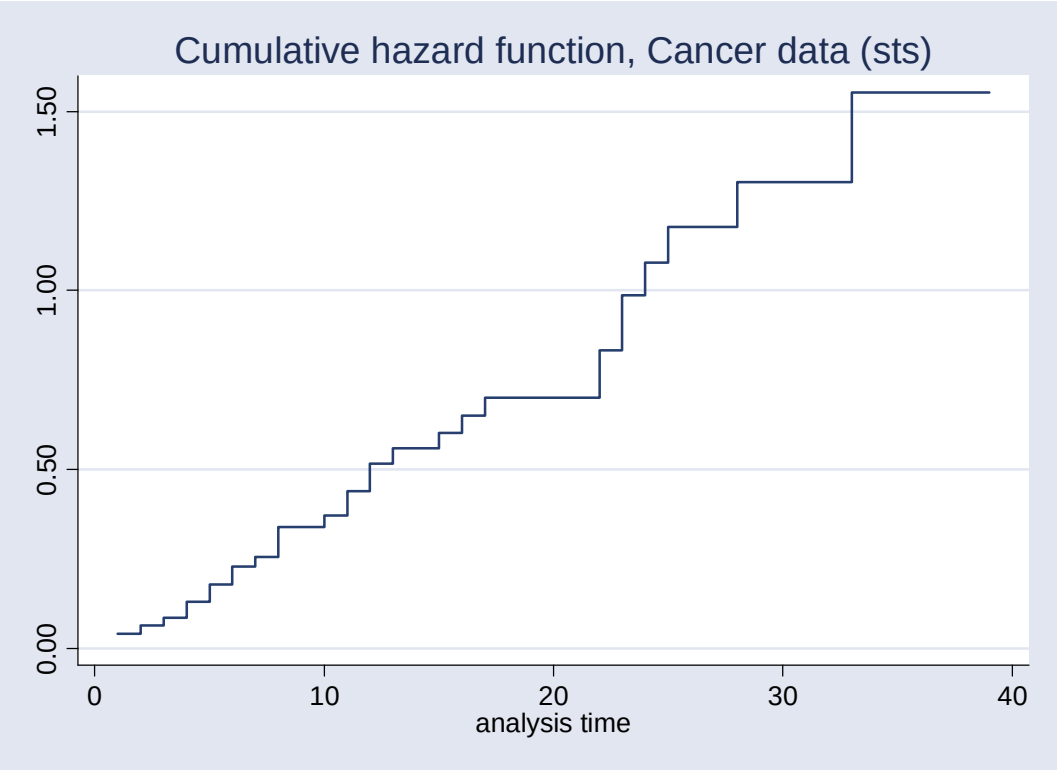
Figure 4.1 Survivor function derived from the Cancer data using sts



If you wanted instead an estimate of the integrated hazard function, add the **na** option to the command above. ('na' stands for Nelson-Aalen, the progenitors of the method used to derive the estimates of the integrated hazard. The estimates could in principle also be derived via the survivor function but the literature has shown that the Nelson-Aalen estimator of the integrated hazard is to be preferred.)

```
. sts graph, title("Cumulative hazard function, Cancer data (sts)") ///
>         na saving(integh1, replace).
```

Figure 4.2 Cumulative hazard function derived from the Cancer data using sts



Using **sts gen**, we can retrieve the estimates shown in the graphs above, together with the estimate of the hazard contribution (defined earlier):

```
. sts gen s = s
. lab var s "KM survivor function, from -sts gen-"
. sts gen cumh = na
. lab var cumh "NA cumulative hazard function, from -sts gen-"
. sts gen deltach = h
. lab var deltach "Hazard contribution, from -sts gen-"
```

We can of course also graph these variables against survival time too, and get pictures that look like Figures 4.1 and Figure 4.2. Here follow some examples.

```
twoway line s _t, sort connect(J) title("Survivor function, Cancer data (sts gen)") ///
       saving(surv2, replace)

twoway line cumh _t, sort connect(J) ///
       title("Cumulative hazard function, Cancer data (sts gen)")  saving(chaz1, replace)

twoway line deltach _t, sort connect(J) ///
       title("Hazard contribution, Cancer data (sts gen)")  saving(deltach1, replace)
```

The graphs could have been made more ‘pretty’ using **graph twoway** options for labelling and scaling axes, and so on. Note the use of the **connect(J)** option: this ensures that points are connected in steps, rather than a straight line between points.

Figure 4.3 Survivor function derived from the Cancer data using -sts gen-

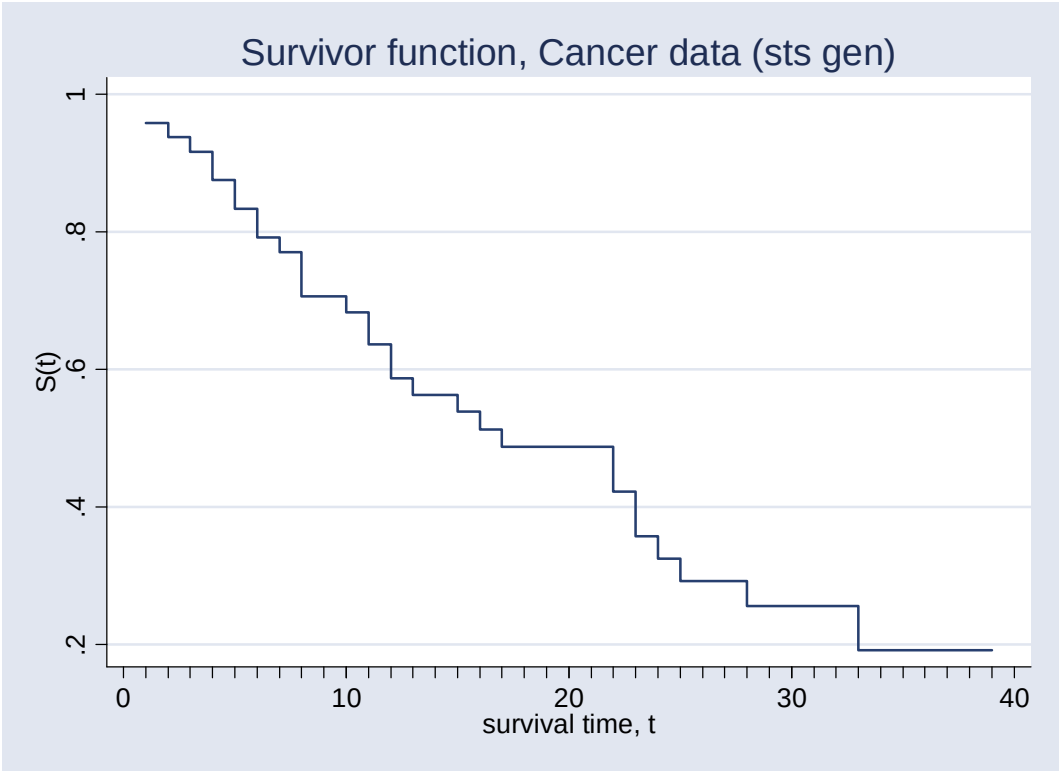
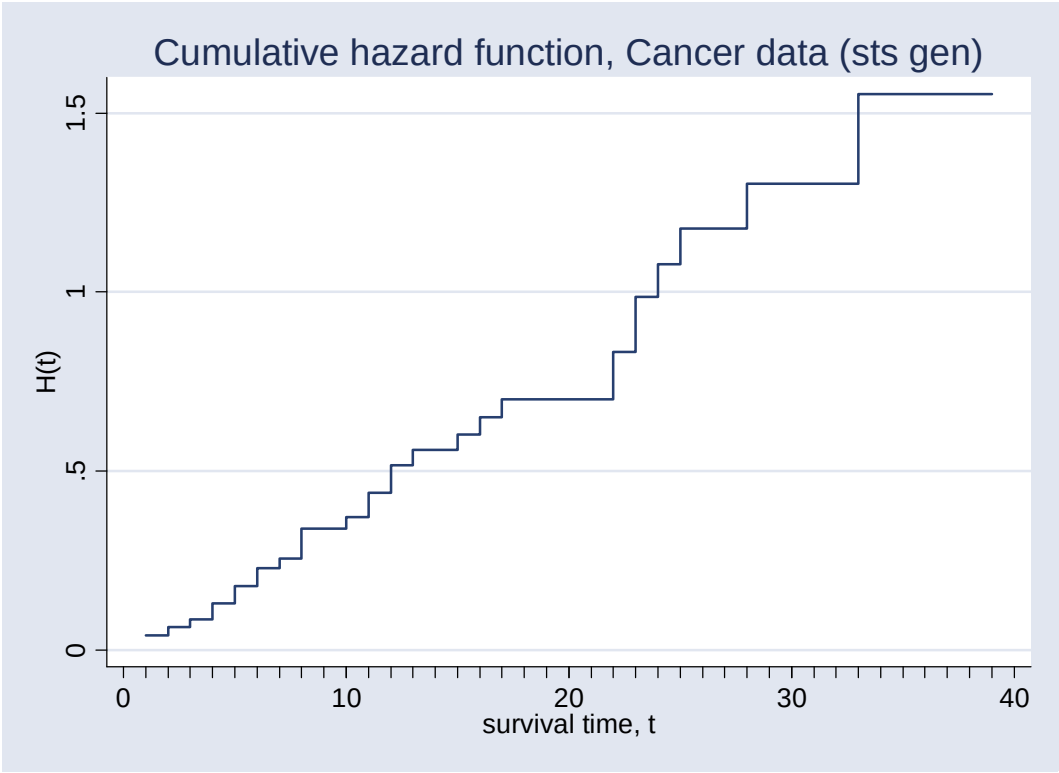


Figure 4.4 Cumulative hazard function derived from the Cancer data using sts

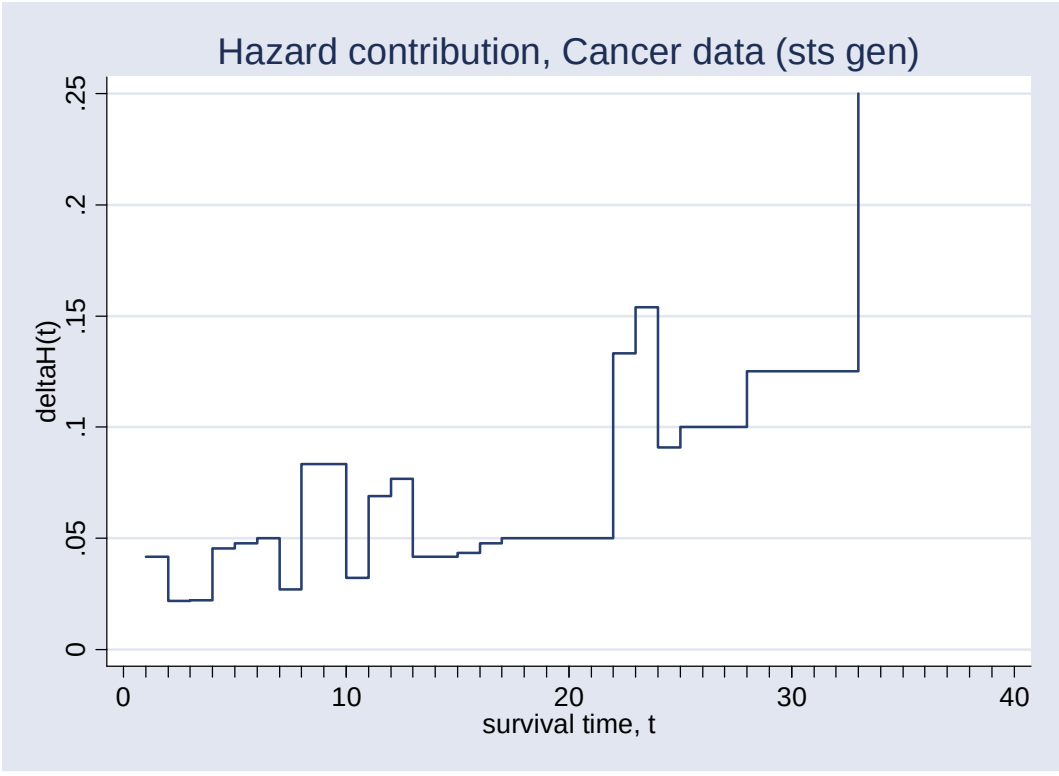


Note the reference to '_t' in the specification of the commands. I knew that **stset** created this variable to represent survival time (it is a synonym for studytim in this case): see the discussion in the previous Lesson of the '_' variables created by **stset**.

Eyeballing the (slope of the) cumulative hazard function shown above suggests that the hazard rate might be relatively constant, decline and then increase (remember that the continuous time hazard rate is the slope of the continuous time integrated hazard rate). But this is indeed just 'eyeballing', and made harder to tell given the staircase nature of the function arising from real data. We shall have another look using the estimate of the smoothed hazard below.

Estimates of the (continuous time) hazard rate might be derived by dividing each value of the hazard contribution by the length of the time interval between the current failure time and the previous failure time – this is an attempt to look at the slope of the cumulative hazard function. But this usually generates poor estimates. (Things would be easier if failures times could be guaranteed to occur at equal-spaced intervals of short length!) Here's what the hazard contributions look like (remember, from earlier, that these are the changes in the cumulative hazard over the intervals of time between successive failure times observed in the data):

Figure 4.5 Hazard contribution function derived from the Cancer data using sts

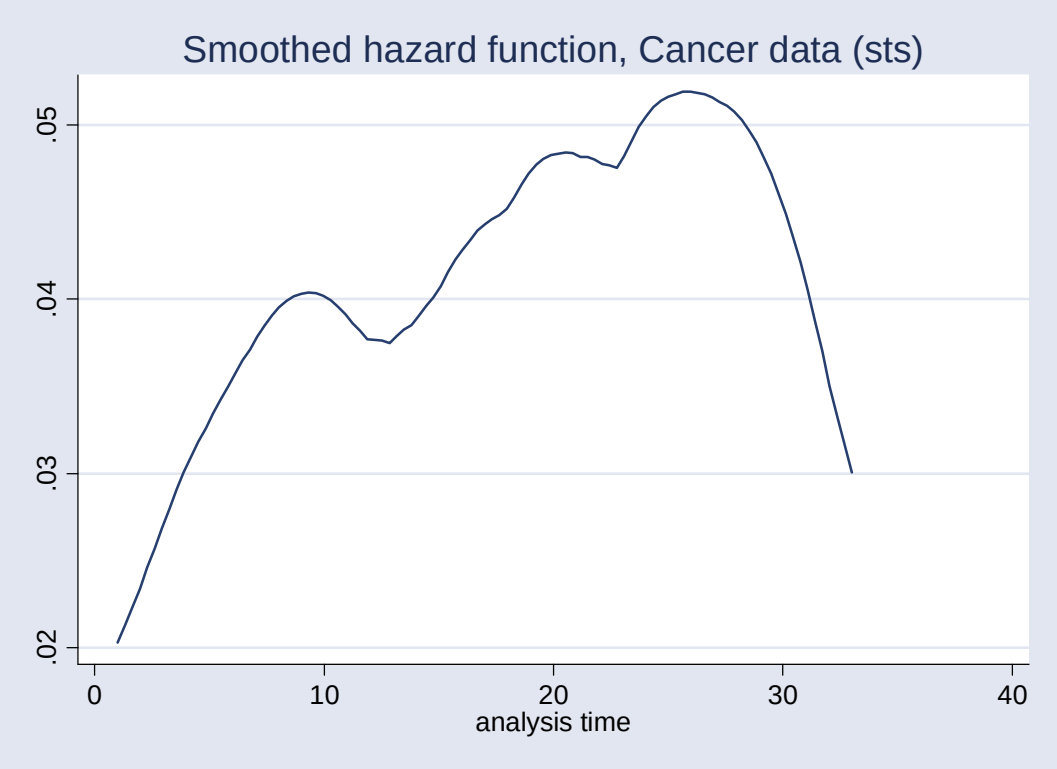


One can get an estimate of the smoothed hazard with the command that follows below. The estimate of the hazard uses a kernel-based smoothing of the hazard contributions defined earlier (the change in the cumulative hazard between successive failures). With kernel

smoothing, the smoothed value at a given time is based on a weighted average of the values in the neighbourhood of that point. The choice of kernel function determines how much weight is given to observations that are close to the one being considered relative to ones that are far away (most kernel functions give greater weights to closer points), and the 'bandwidth' determines how wide the window of observation is when considering which observations are used for the averaging. The larger the bandwidth, the greater the smooth. (Choice of the bandwidth is typically much more important than choice of the kernel function.) I start with the default bandwidth and function.

```
* smoothed hazard, default bandwidth (and default kernel)
sts graph, hazard title("Smoothed hazard function, Cancer data (sts)") ///
       saving(smoothhaz1, replace)
```

Figure 4.6 Smoothed hazard derived from the Cancer data using sts



This illustrates how eyeballing the cumulative hazard function can be potentially misleading about the shape of the (smoothed) hazard! Of course, perceptions also depend on the amount of smoothing. Try the following commands and see how the picture differs, noting that the width option specifies the bandwidths:

```
sts graph, hazard width(2)

sts graph, hazard width(10)
```

Note that one can look directly at the estimates of the various functions by **list**ing the data. What we want is one estimate presented for each value of *t*, but we know that there are estimates for each subject. So, if we simply listed all the data, we would get more than we really want to see: there would be repeated observations on the variables of interest for each subject. The trick is to generate a variable that will select one observation on *t* for each group of observations with the same *t* value. We use the **egen tag()** function to create such a

variable (called *tagt*), and then observations with values of *tagt* = 1. We sort the data by _t before listing them.

```
. egen tagt = tag(_t)
. sort _t
. list _t deltach cumh s if tagt == 1
```

	_t	deltach	cumh	s
1.	1	.04166667	.04166667	.95833333
3.	2	.02173913	.0634058	.9375
4.	3	.02222222	.08562802	.91666667
5.	4	.04545455	.13108256	.875
7.	5	.04761905	.17870161	.83333333
9.	6	.05	.22870161	.79166667
12.	7	.02702703	.25572864	.77027027
13.	8	.08333333	.33906197	.70608108
17.	9	.	.33906197	.70608108
18.	10	.03225806	.37132004	.68330427
20.	11	.06896552	.44028555	.63617984
23.	12	.07692308	.51720863	.58724293
25.	13	.04166667	.5588753	.56277447
26.	15	.04347826	.60235356	.53830602
28.	16	.04761905	.64997261	.5126724
29.	17	.05	.69997261	.48703878
31.	19	.	.69997261	.48703878
33.	20	.	.69997261	.48703878
34.	22	.13333333	.83330594	.42210027
36.	23	.15384615	.98715209	.35716177
38.	24	.09090909	1.0780612	.32469252
39.	25	.1	1.1780612	.29222327
41.	28	.125	1.3030612	.25569536
43.	32	.	1.3030612	.25569536
45.	33	.25	1.5530612	.19177152
46.	34	.	1.5530612	.19177152
47.	35	.	1.5530612	.19177152
48.	39	.	1.5530612	.19177152

Check how these values correspond with what was graphed.

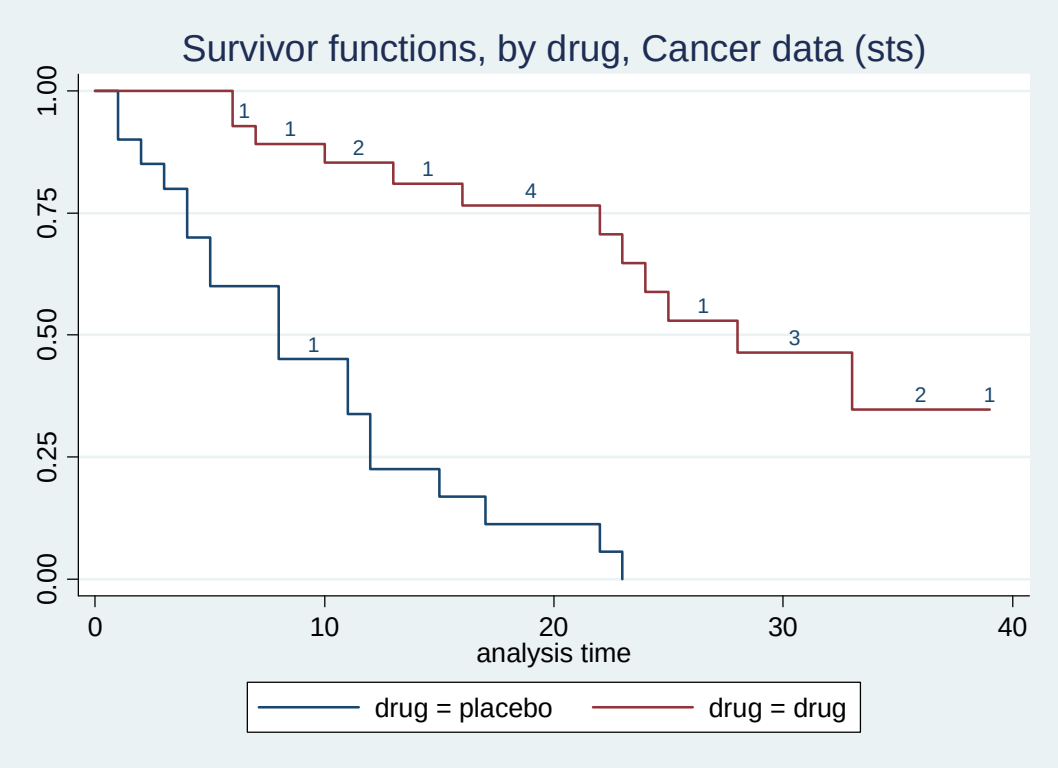
Stratification

One might suppose that survival times will vary according to whether the subjects received a drug (drug = 2 or 3) or a placebo (drug = 1). One can use **sts** with the **by(byvar)** option to derive separate estimates for sample subgroups stratified by a variable such as type of drug received:

```
. sts graph, title("Survivor functions, by drug, Cancer data (sts)") ///
>         by(drug) lost saving(surv5, replace)
```

See Figure 4.7 for the resulting graph. Note the **lost** option in this case: this adds the number of censored subjects onto the graph at the relevant survival times.

Figure 4.7 Survivor function, stratified by drug (Cancer data), derived using sts



Subjects receiving the placebo appear to have shorter survival times. One can look at the estimates themselves, stratified by subgroup, using the command **sts list, by(drug)**. The resulting table has the same format as the earlier **sts list** table except that there is a separate panel of estimates for each subgroup. An alternative and more succinct listing can be produced with

```
. sts list, by(drug) compare

        failure _d:  died
  analysis time _t:  studytim

                Survivor Function
drug           placebo       drug
--------------------------------
time       1     0.9000     1.0000
           5     0.6000     1.0000
           9     0.4500     0.8914
          13     0.2250     0.8100
          17     0.1125     0.7650
          21     0.1125     0.7650
          25          .     0.5296
          29          .     0.4634
          33          .     0.3476
          37          .     0.3476
          41          .          .
--------------------------------
```

One can also use **sts generate** to calculate stratified estimates.

You might also like to test whether the observed subgroup differences in the survivor functions are statistically significant. Two standard tests in the literature are the Log-rank and the Wilcoxon tests. Test statistics can be derived using **sts test** as follows:

```
. sts test drug

         failure _d:  died
   analysis time _t:  studytim


Log-rank test for equality of survivor functions

        |   Events         Events
drug    |  observed       expected
--------+-------------------------
placebo |        19           7.25
drug    |        12          23.75
--------+-------------------------
Total   |        31          31.00

              chi2(1) =      28.27
              Pr>chi2 =     0.0000

. sts test drug, wilcoxon

         failure _d:  died
   analysis time _t:  studytim


Wilcoxon (Breslow) test for equality of survivor functions

        |   Events         Events        Sum of
drug    |  observed       expected        ranks
--------+--------------------------------------
placebo |        19           7.25          385
drug    |        12          23.75         -385
--------+--------------------------------------
Total   |        31          31.00            0

              chi2(1) =      22.61
              Pr>chi2 =     0.0000
```

If the chi-squared value associated with the test is sufficiently large (associated *p*-value sufficiently small), then we reject the null hypothesis of no subgroup differences in survivor functions. In this case, the probability that the observed differences occur by chance is less than 0.00 (i.e. less than 1%, a standard reference point). We would reject the null hypothesis.

Graphical checks regarding model specification

Our final exercise with data assumed to be continuous is illustration of the graphical checks that one may use to check whether the data are consistent with a proportional hazards model, with a Weibull model, or with a log-logistic model. See the Lecture Notes for details.

The check concerning the validity of the PH assumption is based on graphs of the log of non-parametric estimates of the cumulative hazard against time for different subgroups: recall that we are hoping to see the graphs move in parallel. Throughout, we shall simply compare values according to one explanatory variable *drug*, which has two values corresponding to whether the subject received the drug or the placebo.

The first step in implementation is creation of the subgroup variables referring to the log of the estimated cumulative hazard. We use **sts generate** to create a variable *cumhg* which contains values of the cumulative hazard calculated separately for each group. The '**= na**' part of the command tells Stata that we want the cumulative hazard; the **by(.)** option tells Stata that we want separate calculations for the groups defined by the byvar (two groups in this case corresponding to drug = 0 (placebo recipient) and drug = 1 (drug recipient)). We then compute the log of this variable.

```
. sts gen cumhg = na, by(drug)

. ge lch = log(cumhg)
```

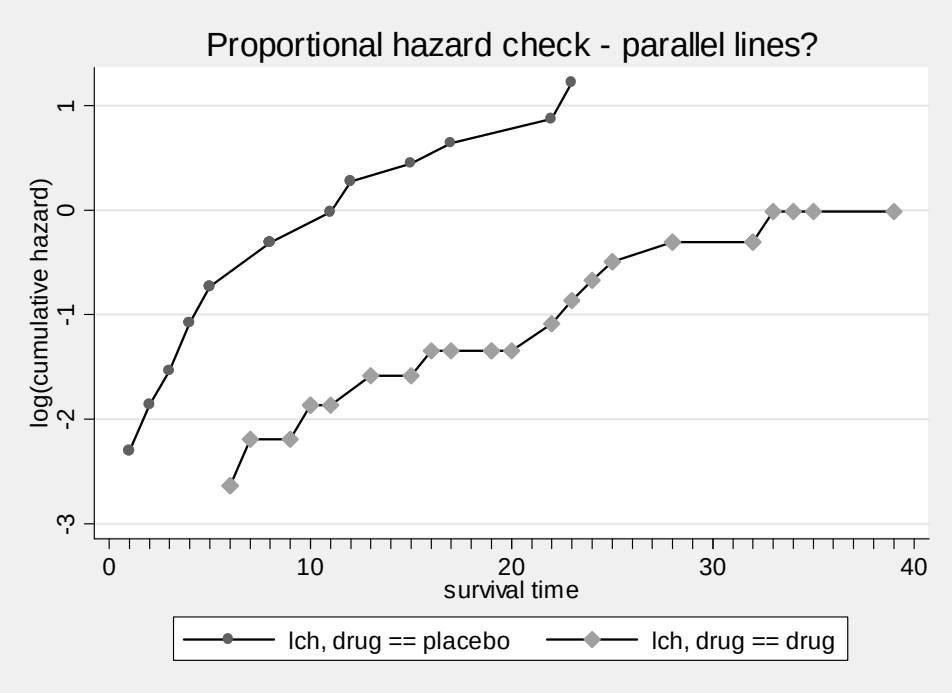
In order to graph the log cumulative variables with separate lines for each group, we need two new variables. The first is equal to lch if drug = 0 and missing otherwise, and the second is equal to lch if drug = 1 and missing otherwise. The **separate** command creates these two new variables, naming lch0 and lch1 by default (the suffix to the name corresponds to the values of drug). You could name the variables using your own variable name stub if you wished, using the **generate()** option.

```
. separate lch, by(drug)  // creates lch0 lch1
```

Now we draw the relevant graph

```
. twoway connect lch0 lch1 _t, ytitle("log(cumulative hazard)") ///
>         title("Proportional hazard check - parallel lines?") ///
>         xtitle("survival time") xtick(1(1)39) sort saving(PHtest1, replace)
```

Figure 4.8 Informal graphical check of the PH assumption



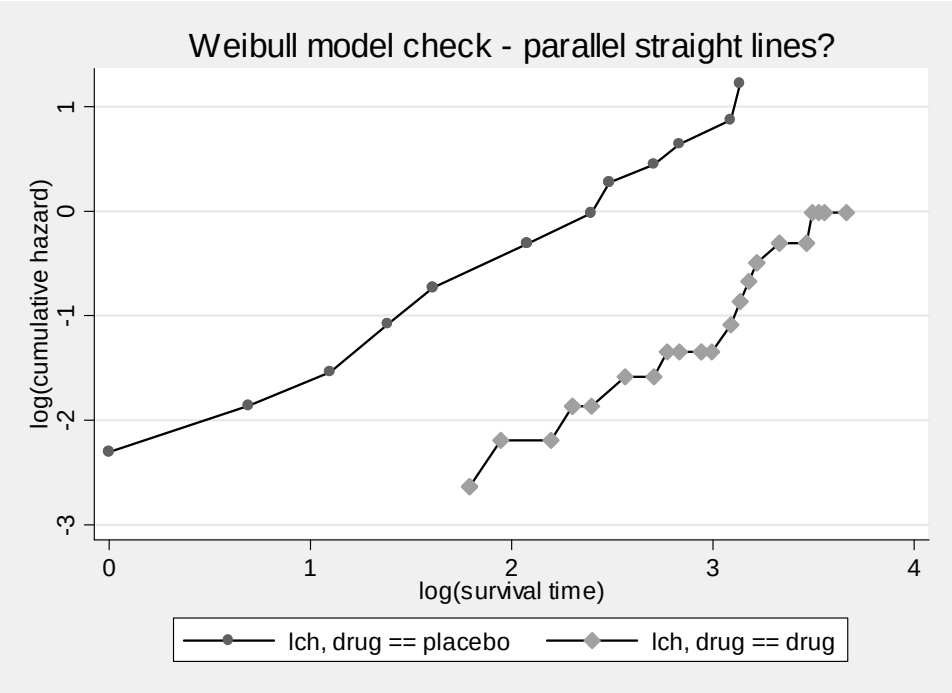
Do you think the graphs have the same slope at each survival time (are 'parallel')? To my eyes, they do not! Note the difficulty of making a firm decision on the basis of a visual inspection – this issue reminds us that these sorts of checks are relatively informal.

The graphical check of the Weibull specification is implemented similarly (recall that it is a PH model): the variable on the vertical axis is the same, but we graph it against the log of survival time on the horizontal axis. We create the log(time) variable first and then draw the graph.

```
. ge logt = log(_t)
```

```
. twoway connect lch0 lch1 logt, ytitle("log(cumulative hazard)") ///
>         xtitle("log(survival time)") ///
>         title("Weibull model check - parallel straight lines?") ///
>         sort saving(Weibtest1, replace)
(file Weibtest1.gph saved)
```

Figure 4.9 Informal graphical check of the Weibull assumption



If the Weibull model is appropriate, the two lines should be parallel straight lines – are they? To my eyes, they are not. (And I hadn't expected them to be because I was already sceptical of the PH assumption on the basis of the previous graph.)

As an exercise, you are asked to do the informal check for the log-logistic model (an AFT model), based on graphs of the log odds of survival

4 Illustration using the Cancer data set: (ii) ltable

ltable provides estimates of survival, failure, and hazard functions. Recall that now that the survival time data are assumed grouped into (equal-spaced) intervals, assumed by default to be of length one unit (but this can be changed – see below).

One important distinction, as ever, is between whether the underlying survival times are continuous but have been observed in grouped form, or whether the times are intrinsically discrete. If the former case is appropriate (it is the most common one in the social sciences), then we can derive estimates of the survivor, failure, and cumulative hazard function using Stata's lifetable command, **ltable**. We can also draw graphs of the survivor function.

Estimates of the underlying continuous time hazard rate can only be derived with assumptions about the shape of the hazard within each interval. The most common assumption in this context is that failures occur at a uniform rate within the intervals, and

thence one can derive an estimate for the midpoint of each interval (see Lecture Notes): this is idea of the 'actuarial adjustment'. **ltable** uses this adjustment by default (the **noadjust** option turns it off). Because of these various complications, versions of Stata from version 8 onwards won't allow you to graph the estimates of the hazard rate in the same way that you can graph the estimated survivor function.

Alternatively we might be interested in the *interval hazard* itself (recall that this is a probability), in which case we do not want the actuarial adjustment. We use the **noadjust** option (in which case the survivor function estimates correspond with those that can be derived using **sts)**. What if we want to draw a graph of the interval hazard? The simplest way to circumvent Stata's restrictions from version 8 onwards is to use the Stata 7 version of **ltable**, using version control as illustrated below. Use this too if survival times are intrinsically discrete.

Let us turn to the mechanics of the **ltable** command. The survivor function is the default estimate; specify the **hazard** option for hazard estimates or the **failure** option for failure function estimates (one minus the survivor function). Use the **gr(aph)** option to draw a graph in addition to providing a table of estimates. If you want a graph but no table, use the **notab** option. The graph, if requested, also shows the point-wise confidence interval. You can suppress this using the **noconf** option. The default estimates are based on the actuarial adjustment for within-interval changes in the number at risk of failure.

I show the tables of survivor function estimates first and then the table of hazard rate estimates.

```
ltable studytim died, graph title("Survivor function, Cancer data (ltable)")
```

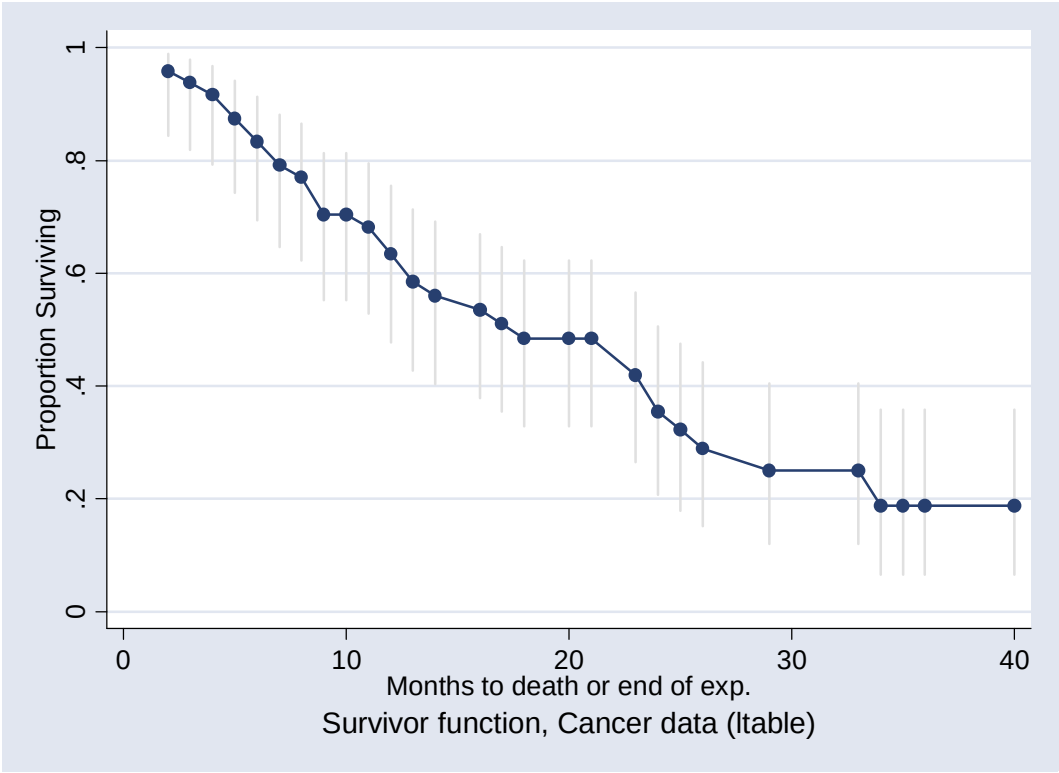
Interval		Beg. Total	Deaths	Lost	Survival	Std. Error	[95% Conf.	Int.]
1	2	48	2	0	0.9583	0.0288	0.8435	0.9894
2	3	46	1	0	0.9375	0.0349	0.8186	0.9794
3	4	45	1	0	0.9167	0.0399	0.7930	0.9679
4	5	44	2	0	0.8750	0.0477	0.7427	0.9418
5	6	42	2	0	0.8333	0.0538	0.6943	0.9129
6	7	40	2	1	0.7911	0.0588	0.6465	0.8817
7	8	37	1	0	0.7698	0.0609	0.6228	0.8653
8	9	36	3	1	0.7047	0.0664	0.5527	0.8134
9	10	32	0	1	0.7047	0.0664	0.5527	0.8134
10	11	31	1	1	0.6816	0.0681	0.5279	0.7945
11	12	29	2	1	0.6338	0.0712	0.4775	0.7547
12	13	26	2	0	0.5850	0.0736	0.4277	0.7129
13	14	24	1	0	0.5606	0.0745	0.4036	0.6914
15	16	23	1	1	0.5357	0.0752	0.3791	0.6692
16	17	21	1	0	0.5102	0.0758	0.3543	0.6463
17	18	20	1	1	0.4840	0.0763	0.3293	0.6226
19	20	18	0	2	0.4840	0.0763	0.3293	0.6226
20	21	16	0	1	0.4840	0.0763	0.3293	0.6226
22	23	15	2	0	0.4195	0.0786	0.2656	0.5660
23	24	13	2	0	0.3550	0.0787	0.2069	0.5061
24	25	11	1	0	0.3227	0.0778	0.1794	0.4749
25	26	10	1	1	0.2887	0.0767	0.1512	0.4418
28	29	8	1	1	0.2502	0.0755	0.1196	0.4050
32	33	6	0	2	0.2502	0.0755	0.1196	0.4050
33	34	4	1	0	0.1877	0.0784	0.0653	0.3585
34	35	3	0	1	0.1877	0.0784	0.0653	0.3585
35	36	2	0	1	0.1877	0.0784	0.0653	0.3585
39	40	1	0	1	0.1877	0.0784	0.0653	0.3585

The graph that appears on your screen is not saved by default, but you can save it by issuing the **graph save** command:

```
. graph save surv3, replace
(file surv3.gph saved)
```

To get the graph back anytime later, you can simply **graph use surv3.gph**. Here is what it looks like:

Figure 4.10 Survivor function (Cancer data), derived using ltable



Now contrast these estimates with those derived without the **noadjust** option:

```
. ltable studytim died, noadjust
```

Interval		Beg. Total	Deaths	Lost	Survival	Std. Error	[95% Conf.	Int.]
1	2	48	2	0	0.9583	0.0288	0.8435	0.9894
2	3	46	1	0	0.9375	0.0349	0.8186	0.9794
3	4	45	1	0	0.9167	0.0399	0.7930	0.9679
4	5	44	2	0	0.8750	0.0477	0.7427	0.9418
5	6	42	2	0	0.8333	0.0538	0.6943	0.9129
6	7	40	2	1	0.7917	0.0586	0.6474	0.8820
7	8	37	1	0	0.7703	0.0608	0.6236	0.8656
8	9	36	3	1	0.7061	0.0661	0.5546	0.8143
9	10	32	0	1	0.7061	0.0661	0.5546	0.8143
10	11	31	1	1	0.6833	0.0678	0.5302	0.7957
11	12	29	2	1	0.6362	0.0708	0.4807	0.7564
12	13	26	2	0	0.5872	0.0733	0.4304	0.7145
13	14	24	1	0	0.5628	0.0742	0.4060	0.6931
15	16	23	1	1	0.5383	0.0749	0.3821	0.6712
16	17	21	1	0	0.5127	0.0756	0.3570	0.6483
17	18	20	1	1	0.4870	0.0761	0.3326	0.6249
19	20	18	0	2	0.4870	0.0761	0.3326	0.6249
20	21	16	0	1	0.4870	0.0761	0.3326	0.6249
22	23	15	2	0	0.4221	0.0786	0.2680	0.5684
23	24	13	2	0	0.3572	0.0788	0.2087	0.5083
24	25	11	1	0	0.3247	0.0780	0.1809	0.4771
25	26	10	1	1	0.2922	0.0767	0.1543	0.4449
28	29	8	1	1	0.2557	0.0753	0.1247	0.4093
32	33	6	0	2	0.2557	0.0753	0.1247	0.4093
33	34	4	1	0	0.1918	0.0791	0.0676	0.3634
34	35	3	0	1	0.1918	0.0791	0.0676	0.3634
35	36	2	0	1	0.1918	0.0791	0.0676	0.3634
39	40	1	0	1	0.1918	0.0791	0.0676	0.3634

Survival probabilities estimated with the **noadjust** option are slightly larger at each corresponding duration interval than those with the **adjust** default, as you would expect. Compare the results too with those from **sts** shown earlier in this Lesson.

Now look at the estimates of the hazard function (these use the actuarial adjustment)

```
. ltable studytim died, hazard
```

Interval		Beg. Total	Cum. Failure	Std. Error	Hazard	Std. Error	[95% Conf.	Int.]
1	2	48	0.0417	0.0288	0.0426	0.0301	0.0000	0.1015
2	3	46	0.0625	0.0349	0.0220	0.0220	0.0000	0.0651
3	4	45	0.0833	0.0399	0.0225	0.0225	0.0000	0.0665
4	5	44	0.1250	0.0477	0.0465	0.0329	0.0000	0.1110
5	6	42	0.1667	0.0538	0.0488	0.0345	0.0000	0.1164
6	7	40	0.2089	0.0588	0.0519	0.0367	0.0000	0.1239
7	8	37	0.2302	0.0609	0.0274	0.0274	0.0000	0.0811
8	9	36	0.2953	0.0664	0.0882	0.0509	0.0000	0.1880
9	10	32	0.2953	0.0664	0.0000	.	.	.
10	11	31	0.3184	0.0681	0.0333	0.0333	0.0000	0.0987
11	12	29	0.3662	0.0712	0.0727	0.0514	0.0000	0.1735
12	13	26	0.4150	0.0736	0.0800	0.0565	0.0000	0.1908
13	14	24	0.4394	0.0745	0.0426	0.0425	0.0000	0.1259
15	16	23	0.4643	0.0752	0.0455	0.0454	0.0000	0.1345
16	17	21	0.4898	0.0758	0.0488	0.0488	0.0000	0.1444
17	18	20	0.5160	0.0763	0.0526	0.0526	0.0000	0.1558
19	20	18	0.5160	0.0763	0.0000	.	.	.
20	21	16	0.5160	0.0763	0.0000	.	.	.
22	23	15	0.5805	0.0786	0.1429	0.1008	0.0000	0.3403
23	24	13	0.6450	0.0787	0.1667	0.1174	0.0000	0.3968
24	25	11	0.6773	0.0778	0.0952	0.0951	0.0000	0.2817
25	26	10	0.7113	0.0767	0.1111	0.1109	0.0000	0.3285
28	29	8	0.7498	0.0755	0.1429	0.1425	0.0000	0.4221
32	33	6	0.7498	0.0755	0.0000	.	.	.
33	34	4	0.8123	0.0784	0.2857	0.2828	0.0000	0.8400
34	35	3	0.8123	0.0784	0.0000	.	.	.
35	36	2	0.8123	0.0784	0.0000	.	.	.
39	40	1	0.8123	0.0784	0.0000	.	.	.

Observe that for intervals in which a hazard rate cannot be calculated (when there are no failures), the **ltable** table shows a hazard equal to zero and no confidence band. Now here are the estimates derived with the **noadjust** option:

```
. ltable studytim died, hazard noadjust
```

Interval		Beg. Total	Cum. Failure	Std. Error	Hazard	Std. Error	[95% Conf.	Int.]
1	2	48	0.0417	0.0288	0.0417	0.0295	0.0050	0.1161
2	3	46	0.0625	0.0349	0.0217	0.0217	0.0006	0.0802
3	4	45	0.0833	0.0399	0.0222	0.0222	0.0006	0.0820
4	5	44	0.1250	0.0477	0.0455	0.0321	0.0055	0.1266
5	6	42	0.1667	0.0538	0.0476	0.0337	0.0058	0.1327
6	7	40	0.2083	0.0586	0.0500	0.0354	0.0061	0.1393
7	8	37	0.2297	0.0608	0.0270	0.0270	0.0007	0.0997
8	9	36	0.2939	0.0661	0.0833	0.0481	0.0172	0.2007
9	10	32	0.2939	0.0661	0.0000	.	.	.
10	11	31	0.3167	0.0678	0.0323	0.0323	0.0008	0.1190
11	12	29	0.3638	0.0708	0.0690	0.0488	0.0084	0.1921
12	13	26	0.4128	0.0733	0.0769	0.0544	0.0093	0.2143
13	14	24	0.4372	0.0742	0.0417	0.0417	0.0011	0.1537
15	16	23	0.4617	0.0749	0.0435	0.0435	0.0011	0.1604
16	17	21	0.4873	0.0756	0.0476	0.0476	0.0012	0.1757
17	18	20	0.5130	0.0761	0.0500	0.0500	0.0013	0.1844
19	20	18	0.5130	0.0761	0.0000	.	.	.
20	21	16	0.5130	0.0761	0.0000	.	.	.
22	23	15	0.5779	0.0786	0.1333	0.0943	0.0161	0.3714
23	24	13	0.6428	0.0788	0.1538	0.1088	0.0186	0.4286
24	25	11	0.6753	0.0780	0.0909	0.0909	0.0023	0.3354
25	26	10	0.7078	0.0767	0.1000	0.1000	0.0025	0.3689
28	29	8	0.7443	0.0753	0.1250	0.1250	0.0032	0.4611
32	33	6	0.7443	0.0753	0.0000	.	.	.
33	34	4	0.8082	0.0791	0.2500	0.2500	0.0063	0.9222
34	35	3	0.8082	0.0791	0.0000	.	.	.
35	36	2	0.8082	0.0791	0.0000	.	.	.
39	40	1	0.8082	0.0791	0.0000	.	.	.

Hazard rates estimated with the **noadjust** option are slightly smaller at each corresponding duration interval than those with the **adjust** default, as you would expect.

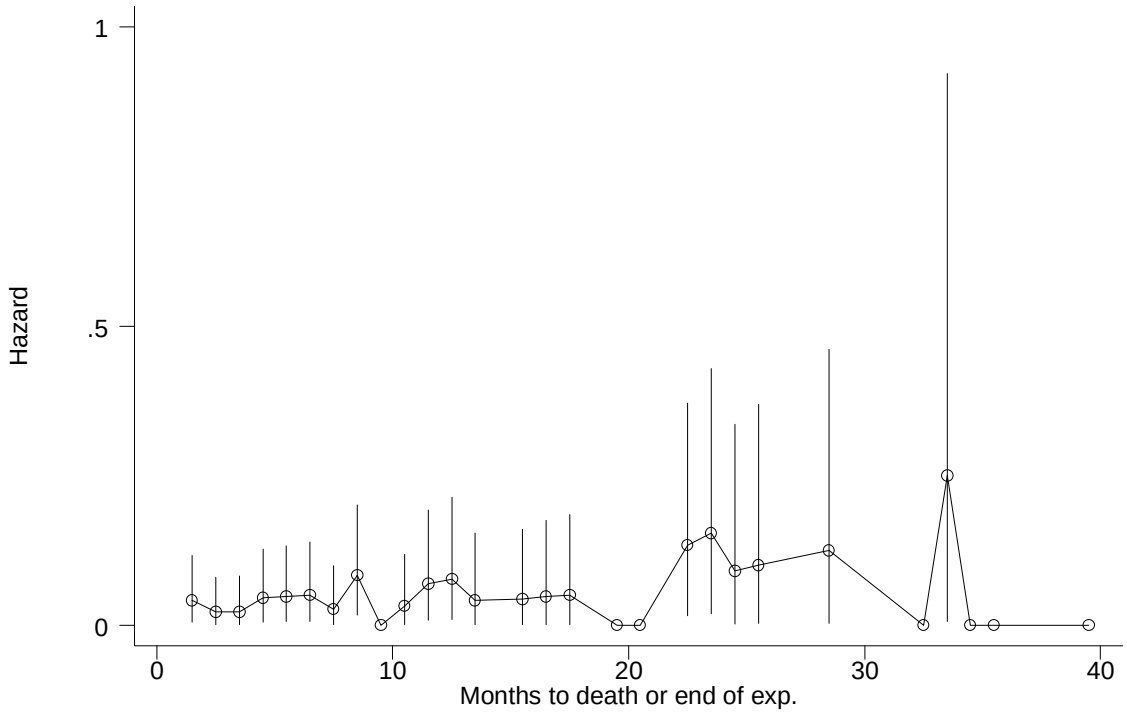
What if one wants a graph of the interval hazard rates (the numbers shown in the table above). We call the Stata version 7 **ltable** command as follows:

```
. version 7: ltable studytim died, haz graph saving(haz3a-noa, replace) noadjust
```

Note the prefix for version control. Note too that in Stata 7, the **saving()** option to **ltable** saved the graph. (In Stata 8 and later versions, the **saving()** option saves the table of estimates to a data file. See ex4_1.do for an example of how to draw the graph using the data from this saved file.

The command results in the following graph. The vertical lines show the 95% confidence intervals for the estimates. Where the hazard rate could not be estimated, because there were no events within the interval (with an estimate of '.' shown in the table), the graph has shown the estimate of the hazard as zero. It is up to you, the analyst, to decide how to interpret the zeros.

Figure 4.11 Interval hazard function (Cancer data), derived using ltable, noadjust



ltable can also produce estimates stratified by subgroup and test the equality of subgroup survivor functions. For stratified estimates, use the **by(groupvar)** option; add **test** for the Log-rank and Wilcoxon tests.

ltable can also produce estimates based on time intervals specified by the user – the default interval has a length of one unit (as in our examples). Typically one would group survival times in this way either because there is a large range of survival times (yielding more detail than required) or it is only by grouping that there are sufficient events per interval to derive the hazard. Use the **intervals(interval)** option.

5 Estimation using data in person-month form rather than person form

In the last chapter, we described how we might do episode-splitting prior to estimation of continuous time models, and almost certainly will in order to estimate discrete time models. **sts** and **ltable** can derive estimates regardless of whether the data have been episode split or not.

We use a simple modification of the commands illustrated earlier. Assuming the data have been appropriately **stset**, then the **sts** commands can be used unchanged. The **ltable** commands simply require the addition of **tvid(idvar)** with the options. In the Cancer data, the idvar is ‘id’, as created earlier.

6 Exercise 4.1

(1) Replicate the tasks above using the marriage data (duration.dta) and the strike data (kennan.dta). What do the survivor functions look like in each of these data sets? If you have access to a Limdep manual (William Greene, *Limdep Version 7.0*, Econometric Software Inc., 1997), you may like to compare your estimates with those shown in the survival analysis chapter. They should be the same! What is the median duration in each case? Do survivor functions rates differ between the sexes in the Marriage data? (Stratification variables are not available in the Strike data.)

(2) Using the cancer data, and assuming survival times are continuous, undertake the graphical check for the log-logistic model. Recall from the Lecture Notes that, if this model is appropriate, then the log odds of survival are a linear function of time.: $\log[S(t,X)/(1 - S(t,X)] = \beta^{*\prime}X - \varphi.\log(t)$. First create the log odds variable, having first derived non-parametric estimates of the survivor function (as in the Lesson), and the log time variable (as in the Lesson). Then draw the graph.

(3) You could also (i) compare **ltable** estimates derived without the **noadjust** option with those derived with it, and (ii) convert your data into expanded (person-month) form and compare estimates with those derived from the data prior to reorganisation.

(4) Estimate the smoothed hazard function using the Marriage data, and see how the picture changes as you vary the width option.

(5) Compare the outputs from the following commands on the cancer data:

```
ltable studytim died, hazard noadjust

stset studytim, f(died)
sts gen hazc = h
egen first = tag(_t)
sort _t
list _t hazc if first == 1
```

Compare the estimates of the hazard with the estimates of the hazard contribution: what do you notice? Now repeat the exercise using the command `ltable studytim died, hazard` (i.e. without the adjust option). How do the results compare now? Can you provide an explanation?