PharmaSUG 2014 - Paper SP10

Our Survival Confidence Intervals are not the Same!

David Franklin, Quintiles Inc., Cambridge, MA

ABSTRACT

"Our Survival Confidence Intervals are not the same!" This was a message I received from a client who was doing some checking of my statistics. After some research it was found that our results were "correct" insofar as the programming was correct but the internal calculations by the software was different. This paper looks into the calculation of the Kaplan-Meier estimate and compares the different methods of how the quartiles are calculated and the formulas for the confidence intervals -- they are not only different among different software but also among different versions of SAS®. Along the way we shall see some of the options that affect the calculations and see how the defaults have changed in recent releases of SAS. There will also be a macro that calculates some options that SAS does not do in older versions.

INTRODUCTION

"Our Survival Confidence Intervals are not the same! What is your program doing? All we are doing is running the program again with a couple of new adverse events that do not effect survival." That was what I was greeted with by a statistician one early morning. I knew the program was fine and the change in data should not affect the output, so started my journey into how SAS calculates the confidence intervals, from a programmers perspective.

THE KAPLAN-MEIER SURVIVAL ESTIMATE

The Kaplan–Meier estimator, also known as the product limit estimator, is one of the best ways that can be used to measure proportion of a sample over time, from a starting point until an event. In clinical trials it is usually used to measure the effect on a drug, in engineering it can be used to measure time until an event occurs.

In the example we shall use in this paper, we are going to use the Kaplan-Meier for a time to event analysis, namely the time it took ten groups of boy scouts to light a fire and burn through a length of string 18 inches above the ground given a time limit of two minutes. The one main reason for using this analysis was to censor events where the group either have up or went beyond the two minute limit. The results recorded were

45*, 75, 77, 84, 87, 88, 115, 117, 120*, 120*,

measured in seconds. The recorded times with a '*' were censored observations.

The Life Table is shown below -- as is usual with the Kaplan-Meier, the Survival Estimate was calculated using the formula

$$S(t) = \prod_{t_i<t} \frac{(n_i - d_i)}{n_i}$$

The Standard Error could be calculated using

$$SE(t) = S(t)\sqrt{\frac{(1-S(t))}{d_i}}$$

but the more usual calculation is done using Greenwood's formula

$$SE(t) = S(t)\sqrt{\sum_{i=1}^{t}\left[\frac{d_i}{n_i(n_i-d_i)}\right]}$$

The simple calculation for the Confidence Intervals is to assume that the sample follows a Normal Distribution and is approximately

S(t) +/- 1.96 SE(t)

This is known as the "Linear" or "Plain" method.

The following Life Table demonstrates the calculations needed to do the Sample Estimate, Standard Error and Confidence Intervals:

ID#	Time to Event	Number at Risk (n_i)	Observe d Events (d_i)	$(n_i-d_i)/n_i$	Survival Proportion S(*t*)	$d_j/(n_j*(n_j-n_j)$ (e)	SUM(e)	SE(*t*)	LCL	UCL
	0	10	0							
1	45	10								
2	75	9	1	0.8889	0.8889	0.0139	0.0139	0.1048	0.6836	1.0942
3	77	8	1	0.8750	0.7778	0.0179	0.0317	0.1386	0.5062	1.0494
4	84	7	1	0.8571	0.6667	0.0238	0.0556	0.1571	0.3587	0.9747
5	87	6	1	0.8333	0.5556	0.0333	0.0889	0.1656	0.2309	0.8802
6	88	5	1	0.8000	0.4444	0.0500	0.1389	0.1656	0.1198	0.7691
7	115	4	1	0.7500	0.3333	0.0833	0.2222	0.1571	0.0253	0.6413
8	117	3	1	0.6667	0.2222	0.1667	0.3889	0.1386	-0.0494	0.4938
9&10	120*	2	0							

The calculations above are not difficult by hand, but the issue is, as mentioned above, the Confidence Intervals -- for the purpose of this paper the 95% CI will be calculated.

Which yields a confidence interval of [0.6836,1.0942] -- using this approach, it is not uncommon for answers to be greater than 1 or less than zero, and to add insult to injury SAS "corrects" these "anomalies" by anything <0 setting to '0', and anything that is >1 setting to '1'.

LOG-LOG

An alternative was proposed by Kalbfleisch and Prentice back in 2002 that gets around the problem of having confidence intervals of >1 or <0, know commonly known as the LOG-LOG transformation. SAS adopted this method back in 9.2 and indeed changed the default method of how the Confidence Interval was calculated. The transformation is done by adopting:

$$v(t)=\log(-\log(S(t))$$

$$SE(t)=\sqrt{\frac{\sum_{i=1}^{t}[\frac{d_i}{n_i(n_i-d_i)}]}{[\sum_{i=1}^{t}\log(\frac{n_i-d_i}{n_i})]^2}}$$

The 95% CI is also calculated a little differently,

$$\lfloor S(t) \rfloor^{e^{+/-1.96SE}}$$

Using the Life Table previously we get

ID#	Time to Event	Number at Risk (n_i)	Observed Events (d_i)	Survival Proportion S(*t*)	SUM(e)	$\log(n_j-d_i)/n_i)$ (l)	SUM(l)	SE*	LCL	UCL
	0	10	0							
1	45	10	0							
2	75	9	1	0.8889	0.0317	-0.1178	-0.1178	1.0006	0.4330	0.9836
3	77	8	1	0.7778	0.0556	-0.1335	-0.2513	0.7090	0.3647	0.9393
4	84	7	1	0.6667	0.0889	-0.1542	-0.4055	0.5813	0.2817	0.8783
5	87	6	1	0.5556	0.1389	-0.1823	-0.5878	0.5072	0.2042	0.8045
6	88	5	1	0.4444	0.2222	-0.2231	-0.8109	0.4596	0.1359	0.7193
7	115	4	1	0.3333	0.3889	-0.2877	-1.0986	0.4291	0.0783	0.6226
8	117	3	1	0.2222		-0.4055	-1.5041	0.4146	0.0337	0.5131
9&10	120*									

* LOG(SE)

As can be clearly seen the results for the lower and upper confidence bands are very different.

SO WHAT IS AVAILABLE IN SAS

Prior to SAS version 9.1, the only Confidence Interval was the Linear or Plain method.

Then in SAS Version 9.1x the SURVIVAL statement could be used to specify a arcsine-square root (ASINSQRT), log-log (LOGLOG), linear (LINEAR), logarithmic (LOG) or logit (LOGIT) transformation using the CONFTYPE=option. The default was then set to LOGLOG with little fanfare.

In SAS version 9.2, the options in the SURVIVAL statement were moved to options in the PROC LIFETEST call with the default remaining the LOGLOG.

Using the original data introduced at the beginning of the paper, the following table shows the 95% Confidence Intervals using the five options available:

				LINEAR		LOGLOG		LOG		LOGIT		ASINSQRT	
Time	Censor	Survival Estimate	SE	LCL	UCL	LCL	UCL	LCL	UCL	LCL	UCL	LCL	UCL
0	.	1.0000	.	.	.	.	.	.	.	.	.	.	.
45	1	1.0000	.	.	.	.	.	.	.	.	.	.	.
75	0	0.8889	0.1048	0.6836	1.0942	0.4330	0.9836	0.7056	1.1199	0.1112	0.9846	0.6178	0.9998
77	0	0.7778	0.1386	0.5062	1.0494	0.3648	0.9393	0.5485	1.1029	0.1720	0.9440	0.4679	0.9733
84	0	0.6667	0.1571	0.3587	0.9746	0.2817	0.8783	0.4200	1.0581	0.2001	0.8889	0.3458	0.9189
87	0	0.5556	0.1656	0.2309	0.8802	0.2042	0.8045	0.3097	0.9966	0.2117	0.8232	0.2421	0.8461
88	0	0.4444	0.1656	0.1198	0.7691	0.1359	0.7193	0.2141	0.9227	0.2117	0.7487	0.1539	0.7579
115	0	0.3333	0.1571	0.0254	0.6413	0.0783	0.6226	0.1323	0.8397	0.2001	0.6666	0.0811	0.6542
117	0	0.2222	0.1386	-0.0494	0.4938	0.0337	0.5131	0.0655	0.7544	0.1720	0.5790	0.0267	0.5321
120	1	0.2222	.	.	.	.	.	.	.	.	.	.	.
120	1	0.2222	.	.	.	.	.	.	.	.	.	.	.

A macro calculating the Survival Estimate, Standard Error and CI intervals without PROC LIFETEST is in Appendix A, the output of which is given below:

					LINEAR		LOGLOG		LOG		LOGIT		ASINSQRT	
t	# at Beg.	# of Events	S(t)	SE(t)	LCL	UCL	LCL	UCL	LCL	UCL	LCL	UCL	LCL	UCL
75	9	1	0.8889	0.1048	0.6836	1.0942	0.4330	0.9836	0.7056	1.1199	0.1112	0.9846	0.6178	0.9998
77	8	1	0.7778	0.1386	0.5062	1.0494	0.3648	0.9393	0.5485	1.1029	0.1720	0.9440	0.4679	0.9733
84	7	1	0.6667	0.1571	0.3587	0.9746	0.2817	0.8783	0.4200	1.0581	0.2001	0.8889	0.3458	0.9189
87	6	1	0.5556	0.1656	0.2309	0.8802	0.2042	0.8045	0.3097	0.9966	0.2117	0.8232	0.2421	0.8461
88	5	1	0.4444	0.1656	0.1198	0.7691	0.1359	0.7193	0.2141	0.9227	0.2117	0.7487	0.1539	0.7579
115	4	1	0.3333	0.1571	0.0254	0.6413	0.0783	0.6226	0.1323	0.8397	0.2001	0.6666	0.0811	0.6542
117	3	1	0.2222	0.1386	-0.0494	0.4938	0.0337	0.5131	0.0655	0.7544	0.1720	0.5790	0.0267	0.5321

CONCLUSION

The 95% CI calculations within the Kaplan–Meier have changed over time causing the results to be more accurate than in previous versions. However, due to the changing algorithms set as the default, it is important to be aware that software is subject to change and therefore must be aware of the changes, and second it is good practice to always write option settings to important procedure calls.

DISCLAIMER

The contents of this paper are the work of the author and do not necessarily represent the opinions, recommendations,or practices of Quintiles Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

David Franklin
Quintiles Inc.
david.franklin@Quintiles.com

APPENDIX: MACRO CALCULATING STANDARD ESTIMATES AND CIs

```
%macro survci(dsn=_LAST_
             ,dso=CIRSLT
             ,time=time
             ,censor=censor /*0=Censored, 1=Event*/
             ,alpha=0.05
             );

   *Compute total subjects;
   data _null_;
      if 0 then set &dsn nobs=nobs;
      call symput('num',trim(left(put(nobs,8.))));
      stop;
   run;

   *Number of subjects at each timepoint;
   proc summary data=&dsn nway;
      class &time;
      var &censor;
      output out=_ci0 n=_sout sum=_sact;
   run;

   *Calcuations;
   data _ci1;
      retain _prestart &num survival . _linearB 0;
      set _ci0;
      _alpha = probit(1-&alpha/2);
      _entered_period=_prestart;
      _event_period=_sact;
      _censor_period=_sout-_sact;
      if _event_period>0 then do;
         _survesta=(_entered_period-_event_period)/_entered_period;
         if missing(survival) then survival=_survesta;
          else survival=survival*_survesta;

         *Greenwood, Standard Error;
         _LinearA=_event_period/(_entered_period*(_entered_period-_event_period));
         _LinearB=_LinearB+_LinearA;
         stderr=sqrt(survival*survival*_LinearB);

         *Linear;
         lcl_linear=survival-_alpha*stderr;
         ucl_linear=survival+_alpha*stderr;

         *LOG-LOG;
         band_loglog = _alpha * sqrt(stderr*stderr / ((survival*log(survival))**2));
         band_loglog = _alpha * abs(stderr / (survival*log(survival)));
         lcl_loglog = survival ** (exp(band_loglog));
         ucl_loglog = survival ** (exp(-band_loglog));

         *LOG;
         band_log = _alpha* sqrt(stderr*stderr / (survival**2));
         lcl_log = survival*exp(-band_log);
         ucl_log = survival*exp(band_log);

         *LOTIT;
         lcl_logit = survival /
          (survival + (survival)*exp(_alpha*stderr / (survival * (1-survival))));
         ucl_logit = survival /
          (survival + (1-survival)* exp(-_alpha*stderr / (survival * (1-survival))));

         *ASINSQRT;
         lcl_asinsqrt = (sin(max(0, arsin(sqrt(survival))-
```

```
          _alpha*stderr/(2*sqrt(survival*(1-survival))))))**2;
         ucl_asinsqrt = (sin(min(constant('pi')/2, arsin(sqrt(survival)) +
          _alpha*stderr/(2*sqrt(survival*(1-survival))))))**2;

         output;

      end;
      _prestart=_prestart-_sout;
   run;

   data &dso;
      retain &time _entered_period _event_period _censor_period survival stderr
             lcl_linear ucl_linear lcl_loglog ucl_loglog lcl_log ucl_log
             lcl_logit ucl_logit lcl_asinsqrt ucl_asinsqrt;
       set _ci1;
       drop _prestart _linearB _TYPE_ _FREQ_ _sout _sact _alpha _survesta
            _LinearA band_loglog band_log;
       label &time='Time'
             _entered_period='Entered Period'
             _event_period='Events During Period'
             _censor_period='Censored During Period'
             survival='Survival Estimate'
             stderr='Standard Error'
             lcl_linear='LINEAR, LCL'
             ucl_linear='LINEAR, UCL'
             lcl_loglog='LOGLOG, LCL'
             ucl_loglog='LOGLOG, UCL'
             lcl_log='LOG, LCL'
             ucl_log='LOG, UCL'
             lcl_logit='LOGIT, LCL'
             ucl_logit='LOGIT, UCL'
             lcl_asinsqrt='ASINSQRT, LCL'
             ucl_asinsqrt='ASINSQRT, UCL';
      format survival stderr lcl_linear ucl_linear lcl_loglog ucl_loglog
             lcl_log ucl_log lcl_logit ucl_logit lcl_asinsqrt ucl_asinsqrt 7.4;
   run;

   proc delete data=work._ci0 work._ci1;
   run;

%mend survci;
```