Hour 2: Survival data structure and non-parametric data exploration

1. Survival data structure

The survival data looks similarly as usual cross-sectional data, with some special features. For the first, we will have some variables to characterize the duration: start, stop, transition/censoring.

Stata does not know the nature of survival data; we will have to tell Stata the data we are working on is survival data. We do that by

Single record per individual:
stset timevar if in, options;
streset …;
stset clear;

multiple records per individual:
stset timevar if in, id() failure() options;

timevar is the time variable showing the actual duration of spell, provided that the spell starts at time 0. if there is explicit start time and stop time, the timevar should be the stop time, and we should in options provide the start time: origin(startvar)

2. Practices (in interactive mode)

Exercise 2-1: single risk Weibull baseline

The Weibull distribution of baseline is
$b(t) = \alpha t^{\alpha-1}$,
the hazard rate is
$h(t) = b(t)\exp(X'\beta) = \alpha t^{\alpha-1}\exp(X'\beta)$

1. open the data file continous_weibull.dta
2. describe the variables
3. do a summarizing statistics
4. use stset timevar , failure(d);
5. generate a dummy variable yy= 1 if y is non negative, yy=0 if y is negative
6. summarize again, this time by dummy variable yy. you need to sort the data by yy first: sort yy; by yy: summarize;
7. use sts to do non-parametric Kaplan-Meier inference
   sts; draw Kaplan-Meier survival function

sts, hazard; draw smoothed hazard
sts, list; display the non parametric survival function for each period
sts graph, options; to display graphs

8. save graphs
9. use ltable command to do life table analysis
   ltable timevar deadvar: ltable t d;
   display hazard: ltable t d, hazard
   display graph: ltable t d, graph
10. compare ltable with sts list results

Exercise 2-2: single risk constant/exponential baseline

The constant baseline/exponential distribution of baseline is
$b(t) = b$,
the hazard rate is
$h(t) = b\exp(X'\beta)$
The data is simulated with $b = 1$,

1. open data file constant_hazard.dta.
2. describe the variables
3. do a summarizing statistics
4. generate again a dummy yy=y >=0;
5. summarize again, this time by dummy variable yy. you need to sort the data by yy
   first: sort yy; by yy: summarize;
6. use stset timevar , failure(d);
7. use sts to do non-parametric Kaplan-Meier inference
   sts; draw Kaplan-Meier survival function
   sts, hazard; draw smoothed hazard (why we have decline empirical hazard?)
   sts list; display the non parametric survival function for each period
   sts graph, options; to display graphs
   sts, by(yy) draw separate graphs by yy
8. save graphs
9. use ltable command to do life table analysis
   ltable timevar deadvar: ltable t d;
   display hazard: ltable t d, hazard
   display graph: ltable t d, graph
10. compare ltable with sts list results

Exercise 2-3: time-varying covariates and episode splitting

One interesting feature with survival data is that it is capable of coping time-varying covariates, meaning the variables can change values during the duration, such as age, calendar month, business cycle conditions etc.

Suppose the spell is recorded originally as one individual per observation. But with time-varying covariates, we will have to split the duration in to subspells at each point where the time-varying covariates change values. It is typical that e.g. age changes at each birth month, or education attainment increases at end of spring semester etc.

In Stata, there are two ways to split spells. We will look at the one that can be done by stsplit command first. However, stsplit can only split spells at pre-specified time points, i.e.
stsplit newvar, at(numlist) or every(#).

If even the time points for splitting are varying with no determinant pattern, some creative programming would be needed. We will leave this to users. Now let's reopen continous_weibull.dta.

```
des; sum;
```

we see that there is another variable x2 which is the second value of say x, at from certain time and beyond. By sum t , detail; we can see that the median duration time is approximately 1. So lets split duration at point 1 and let x have value x2 from t=1 and onwards.

```
stset t, failure(d);
stsplit splitid, at(1) ;
```

Note we get an error message saying: stsplit requires id() option at previous stset. This is easy to understand, since by splitting spell, we get for the same individual multiple records. So to preserve the correct spell length, we need to specify id(id) at stset. It is always a good practice to have id() option at stset, no matter what we do later.

```
stset t, failure(d) id(id);
stsplit splitid, at(1) ;
sort id;
list id t d x x2 in 1/30;
```

we see that splitid indicates the order of subspells, starting from 0. for observations where spell has been splitted, the transition indicator is missing for the subspell before split point. We need to do: define the first subspell to be censored; change the x value to x2 for the second subspell; re stset data.

```
replace d=0 if d==.; /* or we could based on splitid, replace d=0 if splitid==0; */
replace x=x2 if splitid==1; /* change x value for subspell 1*/
stset t, failure(d) id(id);
```

Compare the new stset results with the previous one. E.g. sum t; tab d; etc.

Exercise 2-4: write a do file to do all above, with log. (exercise2.do)