[Survival analysis in Stata 6, slide 1]

`st` refers to a suite or programs to perform survival analysis:

`stset`	Declare data to be survival-time data
`stdes`	Describe survival-time data
`stsum`	Summarize survival-time data
`stvary`	Report which variables vary over time
`stfill`	Fill in by carrying forward values of covariates
`stgen`	Generate variables reflecting entire histories
`sts`	Generate, graph, list, and test the survivor and cumulative hazard functions
`stir`	Report incidence-rate comparison
`strate`	Tabulate failure rate
`stmh`	Calculate rate ratios using Mantel-Haenszel method
`stmc`	Calculate rate ratios using Mantel-Cox method
`stcox`	Estimate Cox proportional hazards model
`stphtest`	Test of Cox proportional hazards assumption
`stphplot`	Graphical assessment of the Cox prop. hazards assumption
`stcoxkm`	Graphical assessment of the Cox prop. hazards assumption
`streg`	Estimate parametric survival models (exponential, weibull, gompertz, lognormal, loglogistic, gamma)
`stcurv`	Plot fitted survival functions
`stsplit`	Split time-span records
`stjoin`	Join time-span records
`stbase`	Form baseline dataset
`sttocc`	Convert survival-time data to case-control data
`sttoct`	Convert survival-time data to count-time data
`cttost`	Convert count-time data to survival-time data
`snapspan`	Convert snapshot data to time-span data
`st_is`	Survival analysis subroutines for programmers

[Survival analysis in Stata 6, slide 2]

st datasets

Observations in st datasets record spans of time (_t0, _t] and contain an *event* variable _d that indicates censoring or failure (_d==0 or _d==1) that occurs at time _t.

_t0	_t	x	_d
0	5	0	1
0	8	0	0
0	7	1	1
0	9	1	1
2	6	0	0

The values of all other variables (for instance, x) are assumed to be constant over the interval (_t0, _t].

There can be multiple observations per subject:

id	_t0	_t	x	_d
1	0	3	0	0
1	3	5	0	1
2	0	3	1	0
2	3	8	0	0
3	0	3	1	0
3	4	8	0	0
3	8	9	1	1
4	0	9	1	1
5	2	4	1	0
5	4	6	0	0

In the above, the first two records record the same information as the first observation in the first dataset; the splitting does not matter.

In the second pair of observations, the ultimate censoring time is the same as in the second observation of the first dataset, but the person changes x values at _t==3.

In the triple of observations for id==3, the subject changes x values and there is observational gap during $[3, 4)$.

There is a single observation for id==4, just as in the first dataset.

In the pair of observations for id==5, the subject changes x values at _t==4. Just as in the first dataset, the subject is first observed at _t==2.

Declaring st datasets — Example 1

failtime	x	failed
5	0	1
8	0	0
7	1	1
9	1	1

```
. stset failtime, failure(failed)
```

failtime	x	failed	_t0	_t	_d
5	0	1	0	5	1
8	0	0	0	8	0
7	1	1	0	7	1
9	1	1	0	9	1

Declaring st datasets — Example 2

enttime	failtime	x	failed
0	5	0	1
0	8	0	0
0	7	1	1
0	9	1	1
2	6	0	0

```
. stset failtime, failure(failed) enter(time enttime)
```

enttime	failtime	x	failed	_t0	_t	_d
0	5	0	1	0	5	1
0	8	0	0	0	8	0
0	7	1	1	0	7	1
0	9	1	1	0	9	1
2	6	0	0	2	6	0

Declaring st datasets — Example 3

patient	time	x	died
1	3	0	0
1	5	0	1
2	3	1	0
2	8	0	0
3	3	1	0
3	8	0	0
3	9	1	1
4	9	1	1

```
. stset time, failure(died) id(patient)
```

patient	time	x	died	_t0	_t	_d
1	3	0	0	0	3	0
1	5	0	1	3	5	1
2	3	1	0	0	3	0
2	8	0	0	3	8	0
3	3	1	0	0	3	0
3	8	0	0	3	8	0
3	9	1	1	8	9	1
4	9	1	1	0	9	0

[Survival analysis in Stata 6, slide 5]

Declaring st datasets — Example 4

patient	date	x	code
1	14may1998	4	22
1	23may1998	4	15
1	31may1998	2	30
1	03jun1998	2	33
1	09jun1998	2	23
1	19jun1998	3	12
2	16oct1998	3	18
2	25oct1998	3	29
2	02nov1998	3	20
2	15nov1998	4	19
2	18nov1998	4	29
3	23dec1998	2	11
3	29dec1998	2	24
3	11jan1999	3	15
3	18jan1999	3	25
3	30jan1999	3	16
3	02feb1999	2	12

```
. stset date, fail(code=23) exit(code=23,16) id(patient) origin(code=15)
```

patient	date	x	code	_t0	_t	_d	_st
1	14may1998	4	22	.	.	.	0
1	23may1998	4	15	.	.	.	0
1	31may1998	2	30	0	8	0	1
1	03jun1998	2	33	8	11	0	1
1	09jun1998	2	23	11	17	1	1
1	19jun1998	3	12	.	.	.	0
2	16oct1998	3	18	.	.	.	0
2	25oct1998	3	29	.	.	.	0
2	02nov1998	3	20	.	.	.	0
2	15nov1998	4	19	.	.	.	0
2	18nov1998	4	29	.	.	.	0
3	23dec1998	2	11	.	.	.	0
3	29dec1998	2	24	.	.	.	0
3	11jan1999	3	15	.	.	.	0
3	18jan1999	3	25	0	7	0	1
3	30jan1999	3	16	7	19	0	1
3	02feb1999	2	12	.	.	.	0

[Survival analysis in Stata 6, slide 6]

Declaring st datasets — Jargon

Time

How time is recorded in your data. This could be calendar time, time from onset of risk, or whatever.

Time units

The units of *time*.

Analysis time (t)

Time from onset of risk.

$$t = \frac{\textit{time} - \textit{origin}}{\textit{scale}}$$

Default value:	$t = \textit{time}$
Option to specify:	`origin()` and `scale()`

origin

Time of onset of risk; the *time* corresponding to $t = 0$.

Default value:	$\textit{origin} = 0$
Option to specify:	`origin()`

origin may be specified as a *time* constant, e.g., 5 or 01jan1999.

origin may be specified as a *time* variable, e.g., `borndate` or `expodate`.

origin may be specified indirectly as the (earliest) *time* corresponding to some event, e.g., `code==16`.

origin may be specified as the latest time of any combination of the above.

Analysis time units (explanation of **scale**)

Time units divided by *scale*.

Default value:	$\textit{scale} = 1$
Option to specify:	`scale()`

scale may be specified as a constant (e.g., 365.25) or as a subject-specific variable.

[Survival analysis in Stata 6, slide 7]

Declaring st datasets — Substantive definition of analysis time

Analysis time t is time from onset of risk.

One implication is,

> Consider two subjects identical in terms of their characteristics. When their *analysis times* are the same, you expect their risk of the failure event occurring to be the same.

Exponential

The hazard is constant with respect to "time".

Any two subjects identical in terms of their characteristics have equal risks at all "times" and so any definition of *analysis time* will do.

Still, Stata requires you choose a definition such that $t \geq 0$ for all subjects because Stata ignores observations for which $t < 0$.

All other parametric

The hazard function is not constant over time—it has a shape. In these cases, you are attributing an effect due to "time".

Not only does it matter that two subjects of have same value for *analysis times* when their risks are the same, the definition of $t = 0$ matters, too.

For example: in a Weibull model, add 500 to all analysis times (changing the definition of 0 to, in effect, -500). Reestimate and you will get a different model that makes different predictions.

Most parametric functions can be thought of as accumulating something and that accumulation begins at $t = 0$. Generators start accumulating heat when they are switched on. Smokers start accumulating bodily damage when they start smoking. You are assuming that those accumulations are zero at $t = 0$. How the accumulation process works is what determines the choice of parameterization.

Cox

The hazard varies with time. Two subjects with equal values of analysis time face equal risk, so how you set analysis time matters. The definition of $t = 0$, however, is irrelevant because Cox does not force a parametric relationship between hazards at different times.

[Survival analysis in Stata 6, slide 8]

Declaring st datasets — More jargon

Entry time

The **time** at which the subject first came under observation.
Default value: *time* corresponding to $t = 0$
Option to specify: `entry()`

Entry time may be specified as a *time* constant, e.g., 5 or 01jan1999.

Entry time may be specified as a *time* variable, e.g., `intvdate` or `diagdate`.

Entry time may be specified indirectly as the (earliest) *time* corresponding to some event, e.g., `code==23`.

Entry time may be specified as the latest time of any combination of the above.

Exit time

The **time** at which the subject was last under observation.
Default value: *time* corresponding to failure or, if no failure,
time subject last in data
Option to specify: `exit()`

Exit time may be specified as a *time* constant, e.g., 5 or 01jan1999.

Exit time may be specified as a *time* variable, e.g., `lastdate` or `dieddate`.

Exit time may be specified indirectly as the (earliest) *time* corresponding to some event, e.g., `code==23`. Doing this, you can omit the failure event and so keep the subject at risk for repeated failures.

Exit time may be specified as the earliest date of any combination of the above.

Time0 (constructing gaps)

Remember that *time* in an observation records the end of the time span covered by the record. *Time0* records the beginning of the time span.
Default value: *time* corresponding to $t = 0$ on first record and
time of previous record for subsequent record.
Option to specify: `time0()`

Time0 may be specified as a variable.

st datasets are odd

patient	date0	date	x1	x2	code
1	12may1998	14may1998	4		22
1	14may1998	23may1998	4	2	15
1	23may1998	31may1998	2	2	30
1	31may1998	03jun1998	2	2	33
1	03jun1998	09jun1998	2	2	23
1	03jun1998	19jun1998	3	2	12
2	13oct1998	16oct1998	3		22
2	16oct1998	25oct1998	3	6	29
2	25oct1998	02nov1998	3	6	20
2	02nov1998	15nov1998	4	6	19
2	15nov1998	18nov1998	4	6	29

Much more reasonable is

patient	date	x1	x2	code	*Explanation*
1	12may1998	4	.	69	*admitted; x1 measured*
1	14may1998	.	2	22	*22 happens; x2 measured*
1	23may1998	2	.	15	*15 happens; x1 remeasured*
1	31may1998	.	.	30	*30 happens*
1	03jun1998	.	.	33	*33 happens*
1	09jun1998	3	.	23	*23 happens, x1 remeasured*
1	19jun1998	.	.	12	*12 happens*
2	13oct1998	3	.	69	
2	16oct1998	.	6	22	
2	25oct1998	.	.	29	
2	02nov1998	4	.	20	
2	15nov1998	.	.	19	
2	18nov1998	.	.	29	

This is called **snapshot dataset**.

Each record records an instant in time.

Our problem: to convert this dataset to st data.

[Survival analysis in Stata 6, slide 10]

Converting snapshot data to st data

Type

```
    . snapspan patient date code, generate(date0)
```

You now have

patient	date0	date	x1	x2	code
1	.	12may1998	.	.	69
1	12may1998	14may1998	4	.	22
1	14may1998	23may1998	.	2	15
1	23may1998	31may1998	2	.	30
1	31may1998	03jun1998	.	.	33
1	03jun1998	09jun1998	.	.	23
1	09jun1998	19jun1998	3	.	12
2	.	13oct1998	.	.	69
2	13oct1998	16oct1998	3	.	22
2	16oct1998	25oct1998	.	6	29
2	25oct1998	02nov1998	.	.	20
2	02nov1998	15nov1998	4	.	19
2	15nov1998	18nov1998	.	.	29

Type

```
    . stset date, id(patient) time0(date0) origin(min) failure(code==1000)
    . stfill x1, forward
    . stfill x2, forward
```

You now have

patient	date0	date	x1	x2	code
1	.	12may1998	.	.	69
1	12may1998	14may1998	4	.	22
1	14may1998	23may1998	4	2	15
1	23may1998	31may1998	2	2	30
1	31may1998	03jun1998	2	2	33
1	03jun1998	09jun1998	2	2	23
1	09jun1998	19jun1998	3	2	12
2	.	13oct1998	.	.	69
2	13oct1998	16oct1998	3	.	22

etc.

Useful data management once the dataset has been stset

`stvary`	report on constant and missing values
`stfill`	replace missing values
`stgen`	make new variables

```
. stvary
           subjects for whom the variable is
                                                  never    always   sometimes
variable |  constant    varying                 missing   missing   missing
---------+------------------------------------------------------------------
     sex |       337          0                       2         0       335
  weight |       235        100                       4         2       331
      bp |        14        320                       3         3       331

. stfill sex weight, forward
replace missing values with previously observed values
             sex:  333 real changes made
          weight:  330 real changes made

. stgen new = max(sex)

. replace sex = new
(2 real changes made)

. drop new
```

Comment: `stfill` very much needs a `backward` option. Because it does not have such an option, I used `stgen` to fill make a new variable that filled in the earlier observations.

[Survival analysis in Stata 6, slide 12]

Histories

patient	date	x	code
1	14may1998	4	22
1	23may1998	4	15
1	31may1998	2	30
1	03jun1998	2	33
1	09jun1998	2	23
1	19jun1998	3	12

```
. stset date, fail(code=23) exit(code=23,16) id(patient) origin(code=15)
```

patient	date	x	code	_t0	_t	_d	_st
1	14may1998	4	22	.	.	.	0
1	23may1998	4	15	.	.	.	0
1	31may1998	2	30	0	8	0	1
1	03jun1998	2	33	8	11	0	1
1	09jun1998	2	23	11	17	1	1
1	19jun1998	3	12	.	.	.	0

```
<------------------------------ history --------------------------------->
<- past ->|<----------------------current-------------------->|<-future->
        (15)                    (30)      (33)              (23)        (event)
          |                       |         |                 |
          |                       |         |                 |
    x = 4 |        x = 2          |  x = 2  |     x = 2       | x = 3
          |                       |         |                 |
----------+-----------------------+---------+-----------------+------->-
        23may                   31may      3jun              9jun      time
          0                       8        11                17        t
```

`streset, past`	sets past + current
`streset, future`	sets current + future
`streset, past future`	sets past + current + future
`streset`	sets current

How to stset all the data

```
. stset time, id(idvar) origin(min) exit(time .) failure(anything)
```

or

```
. stset time, time0(time0) ... (same as above) ...
```

Importantly,

`origin(min)`	obtains the past It finds a definition for analysis time t that excludes no data
`exit(time .)`	obtains the future It says subjects never exit until they run out of data
`id()`	sets the id variable, as always
`time0()`	sets the *time0* variable, if you have one
`failure()`	you should not have to specify; but you do Any definition will do

Stata ought to have the syntax

```
. stset time, id(idvar) [time0(time0)] all
```

That is on the list of things to do.

**You stset all the data for purposes of data cleaning,
not analysis.**

Finally, survival ANALYSIS

```
. stset time, id(id) failure(died)
```

id	time	died	drug	age	_t0	_t	_d	_st
4	3	1	1	52	0	3	1	1
5	4	1	1	56	0	4	1	1
7	5	1	1	63	0	5	1	1
11	8	1	1	52	0	8	1	1
13	11	1	1	50	0	11	1	1

etc.

```
. stcox age drug
```

```
. stsplit t, at(5)
```

id	time	died	drug	age	_t0	_t	_d	_st	t
4	3	1	1	52	0	3	1	1	0
5	4	1	1	56	0	4	1	1	0
7	5	1	1	63	0	5	1	1	0
11	5	.	1	52	0	5	0	1	0
11	8	1	1	52	5	8	1	1	5
13	5	.	1	50	0	5	0	1	0
13	11	1	1	50	5	11	1	1	5

etc.

```
. gen drug5 = drug*(t==5)
. stcox age drug drug5
```

Finally, survival ANALYSIS, continued

```
. drop t drug5
. stjoin

              id  time  died  drug  age    _t0  _t  _d  _st
             ---------------------------------------------
               4     3     1     1   52      0   3   1    1
               5     4     1     1   56      0   4   1    1
               7     5     1     1   63      0   5   1    1
              11     8     1     1   52      0   8   1    1
              13    11     1     1   50      0  11   1    1
             etc.
```

```
. stsplit t, at(5(5)25)

            id  time  died  drug  age    _t0  _t  _d  _st     t
           ---------------------------------------------------
             4     3     1     1   52      0   3   1    1     0
             5     4     1     1   56      0   4   1    1     0
             7     5     1     1   63      0   5   1    1     0

            11     5     .     1   52      0   5   0    1     0
            11     8     1     1   52      5   8   1    1     5

            13     5     .     1   50      0   5   0    1     0
            13    10     .     1   50      5  10   0    1     5
            13    11     1     1   50     10  11   1    1    10

           etc.
```

```
. gen dxt = drug*t
. stcox age drug dxt
```

How Cox works

id	failtime	x	failed	_t0	_t	_d
1	5	0	1	0	5	1
2	8	0	0	0	8	0
3	7	1	1	0	7	1
4	9	1	1	0	9	1

```
        +----------------------------------------------------------+
        | Piece of information, _t=5                               |
        |     4 subjects in the risk group:, ids (1,2,3,4)         |
        |     (1) fails                                            |
_t = 5, |     Calculate likelihood (1) fails                       |   =  L1
        |          given (1, 2, 3, 4) could fail and               |
        |          given one failure occurs now                    |
        |     FYI, the risk group now contains (2,3,4)             |
        +----------------------------------------------------------+
        +----------------------------------------------------------+
        | Accounting (no information), _t = 8                      |
_t = 8, |    (2) is censored                                       |
        |    FYI, the risk group now contains (3,4)                |
        +----------------------------------------------------------+
        +----------------------------------------------------------+
        | (Conditionally) independent piece of information, _t=7 |
        |     Risk group now contains (3,4)                        |
        |     (3) fails                                            |
_t = 7, |     Calculate likelihood (3) fails                       |   =  L2
        |          given (3, 4) could fail and                     |
        |          given one failure occurs now                    |
        |     FYI, the risk group now contains (4)                 |
        +----------------------------------------------------------+
        +----------------------------------------------------------+
        | Conditionally independent piece of information, _t=9   |
        |     Risk group now contains (4)                          |
        |     (4) fails                                            |
_t = 9, |     Calculate likelihood (4) fails                       |  =    1
        |          given (4) could fail and                        |
        |          given one failure occurs now                    |
        |     FYI, the risk group now contains ()                  |
        +----------------------------------------------------------+

                         LIKELIHOOD = L1 * L2 * 1
```

[Survival analysis in Stata 6, slide 17]

How Cox works

Only the values of covariates at the failure times matter.

Whereas one way of introducing continuous time into the model is,

```
. stsplit t, at(1(1)50)
. gen dxt = drug*t
. stcox age drug dxt
```

assuming _t is integral and 50 the the maximum value of _t, another way, were `stsplit` improved, would be

```
. stsplit t, at(failures)
. gen dxt = drug*t
. stcox age drug dxt
```

Moreover, `at(failures)` ought to be the default, so you could just type

```
. stsplit t
```

`stsplit` will be improved in this way and the update published in the STB.

Splitting on other time-varying covariates

id	time	died	drug	age	_t0	_t	_d	_st
4	3	1	1	52	0	3	1	1
5	4	1	1	56	0	4	1	1
7	5	1	1	63	0	5	1	1
11	8	1	1	52	0	8	1	1
13	11	1	1	50	0	11	1	1
etc.								

```
. stcox age drug
```

```
. gen yrborn = -age
. stsplit age5 = yrborn, at(50(5)65)
```

id	time	died	drug	age	_t0	_t	_d	_st	yrborn	age5
4	3	1	1	52	0	3	1	1	-52	50
5	4	1	1	56	0	4	1	1	-56	55
7	2	.	1	63	0	2	0	1	-63	60
7	5	1	1	63	2	5	1	1	-63	65
11	3	.	1	52	0	3	0	1	-52	50
11	8	1	1	52	3	8	1	1	-52	55
13	5	.	1	50	0	5	0	1	-50	50
13	10	.	1	50	5	10	0	1	-50	55
13	11	1	1	50	10	11	1	1	-50	60
etc.										

```
. stcox age5 drug
```

How Cox works . . . continuous age

Only the values of covariates at the failure times matter.

Whereas one way of introducing continuous age into the model is,

```
. stsplit c_age=borndate, at(50(1)65)
. stcox c_age drug
```

assuming _t is integral and that 50 and 65 are the minimum and maximum ages over the period, another way, were `stsplit` improved, would be

```
. stsplit c_age=borndate, at(failures)
. stcox c_age drug
```

Moreover, `at(failures)` ought to be the default, so you could just type

```
. stsplit c_age=borndate
```

`stsplit` will be improved in this way and the update published in the STB.

Multiple splits

id	time	died	drug	age	_t0	_t	_d	_st
4	3	1	1	52	0	3	1	1
5	4	1	1	56	0	4	1	1
7	5	1	1	63	0	5	1	1
11	8	1	1	52	0	8	1	1
13	11	1	1	50	0	11	1	1

etc.

```
. stsplit t, at(5(5)25)
. gen yrborn = -age
. stsplit age5 = yrborn, at(50(5)65)
```

id	time	died	drug	age	_t0	_t	_d	_st	t	age5
4	3	1	1	52	0	3	1	1	0	50
5	4	1	1	56	0	4	1	1	0	55
7	2	.	1	63	0	2	0	1	0	60
7	5	1	1	63	2	5	1	1	0	65
11	3	.	1	52	0	3	0	1	0	50
11	5	.	1	52	3	5	0	1	0	55
11	8	1	1	52	5	8	1	1	5	55
13	5	.	1	50	0	5	0	1	0	50
13	10	.	1	50	5	10	0	1	5	55
13	11	1	1	50	10	11	1	1	10	60

etc.

```
. gen dxt = drug*t
. stcox age5 drug dxt
```

Multiple splits after stsplit is improved

After improvement, typing

```
    . stsplit t
```

will split the data at every failure time.

Thus, to now obtain continuous age (continuous as far as the Cox model is concerned),

```
    . gen c_age = age + t
```

assuming age measures age at analysis time $t = 0$.

[Survival analysis in Stata 6, slide 22]

Parametric models

If you do not suspect that the hazard goes up and then down, or down and then up in odd ways, I urge you to consider parametric models. Stata will fit a variety of shapes of smooth hazards.

```
. streg indepvars, dist(weibull)
```

Stata reports all parametric models---where possible---in the hazard metric. In such cases Results are directly comparable to those obtained from the Cox proportional hazards model:

```
. stcox indepvars
```

Criticism: Stata does not estimate stratified parametric models.

In stratified models, the baseline hazard is allowed do differ.

In parametric models, there are parameters that control this shape. It is p and the intercept in the case of Weibull.

Including dummy variables for strata still restricts the shape parameter p to be the same.

Fixing this is on our list of things to do.