生存分析

生存関数やハザード関数（Nelson-Alesn 推定ハザード関数など）の変数を作成するには、`sts` コマンドを使用します。生存関数を作成するために、Kaplan-Meier 推定を行ったり、Cox 回帰を用いて調整した推定を行うこともできます。

この例題集でできること

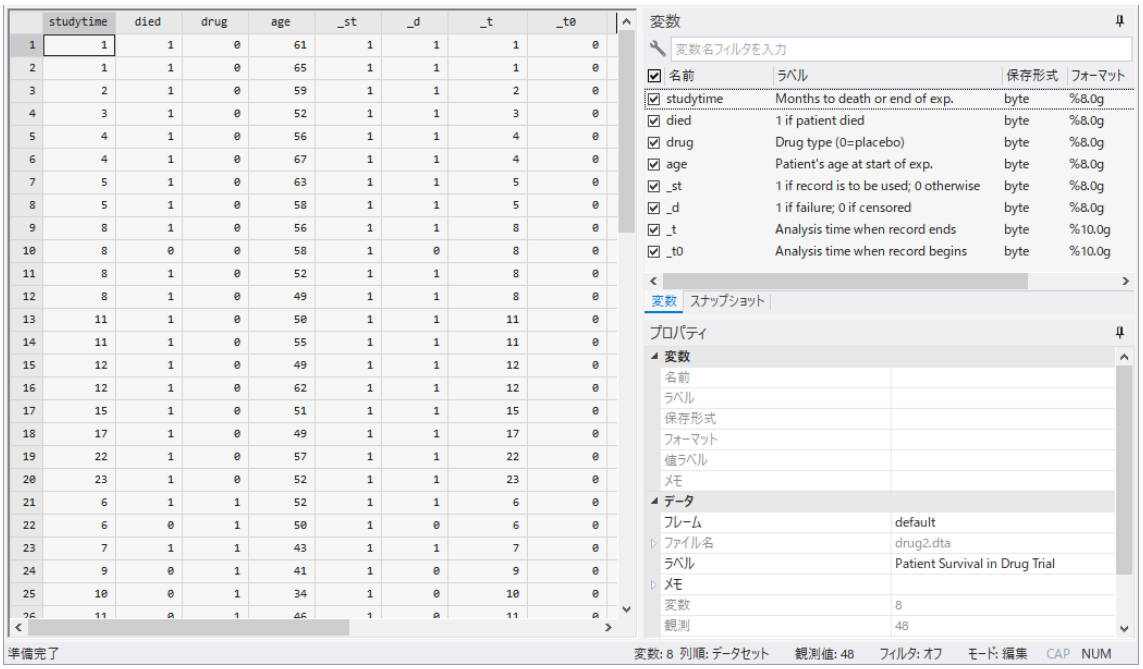
- Kaplan-Meier 法による生存関数グラフの作成 ― 生存確率（イベント発生確率）の時間変化を調べます
- Nelson-Aalen 累積ハザード関数のグラフ作成 ― 死亡確率の時間変化を調べます
- ハザード関数の推定 ― Nelson-Aalen 累積ハザード関数を平滑化します
- Cox 比例ハザードモデルのグラフ作成 ― 説明変数がハザード関数に影響を及ぼしているか否かを調べます

Kaplan-Meier 法による生存関数グラフの作成

- 最初に、基本となるグラフを作成します。下記のコマンドで例題用のサンプルデータ「drug2」を入手します。

```
.use https://www.stata-press.com/data/r16/drug2
```

- サンプルデータには、下記のデータが入力されています。

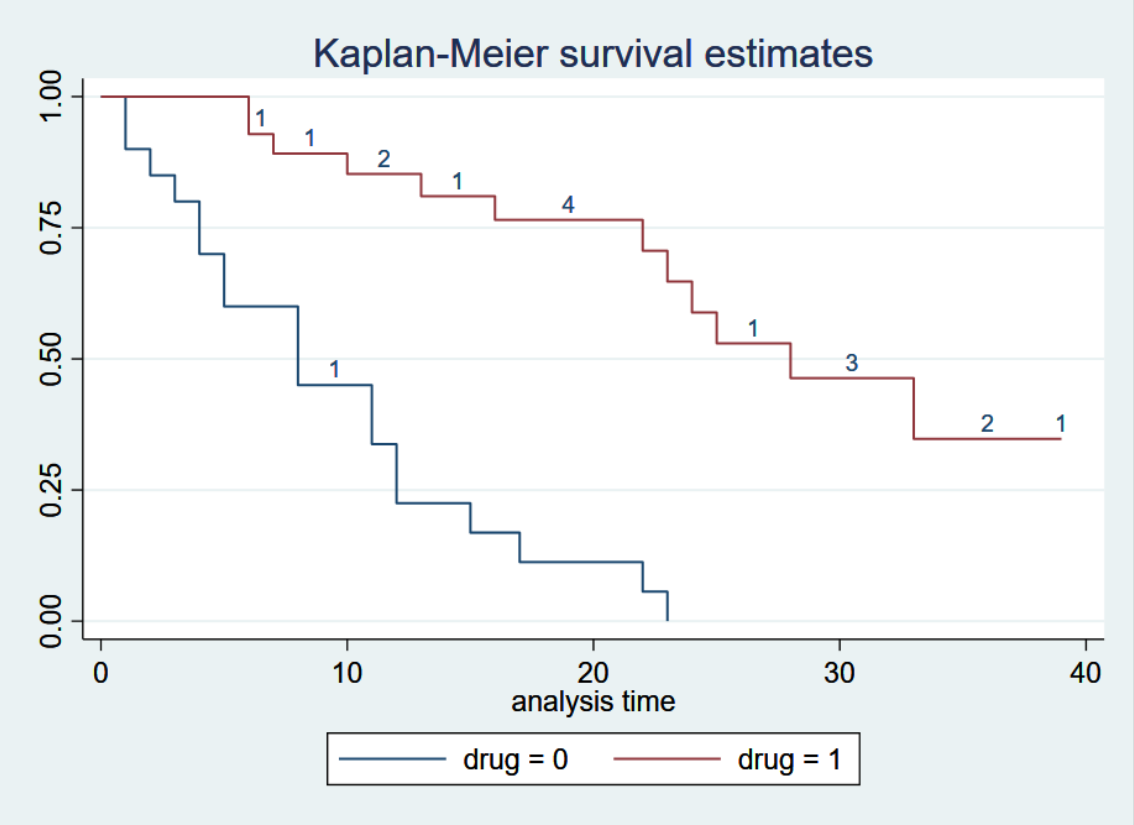


studytime	died	drug	age	_st	_d	_t	_t0
1	1	0	61	1	1	1	0
1	1	0	65	1	1	1	0
2	1	0	59	1	1	2	0
3	1	0	52	1	1	3	0
4	1	0	56	1	1	4	0
4	1	0	67	1	1	4	0
5	1	0	63	1	1	5	0
5	1	0	58	1	1	5	0
8	1	0	56	1	1	8	0
8	0	0	58	1	0	8	0
8	1	0	52	1	1	8	0
8	1	0	49	1	1	8	0
11	1	0	50	1	1	11	0
11	1	0	55	1	1	11	0
12	1	0	49	1	1	12	0
12	1	0	62	1	1	12	0
15	1	0	51	1	1	15	0
17	1	0	49	1	1	17	0
22	1	0	57	1	1	22	0
23	1	0	52	1	1	23	0
6	1	1	52	1	1	6	0
6	0	1	50	1	0	6	0
7	1	1	43	1	1	7	0
9	0	1	41	1	0	9	0
10	0	1	34	1	0	10	0

sutdytime	死亡または打ち切りまでの月数
died	1=死亡、0=打ち切り
drug	1=服薬あり、0=服薬なし
age	観察開始時の患者の年齢

- 投薬の有無（drug）の違いによる生存率の時間変化をグラフにします。

```
.sts graph, by(drug) lost
```



このデータは、すべての患者のデータが観察開始時間（t=0）から始まっています。t=0 以降に観察を始める患者のデータを追加すること（遅延組入）はしていません。そこで、次は t=0 以降に新規患者のデータを追加していく場合のグラフを作成します。

- 遅延組入のあるサンプルデータ「drug2b」を入手します。

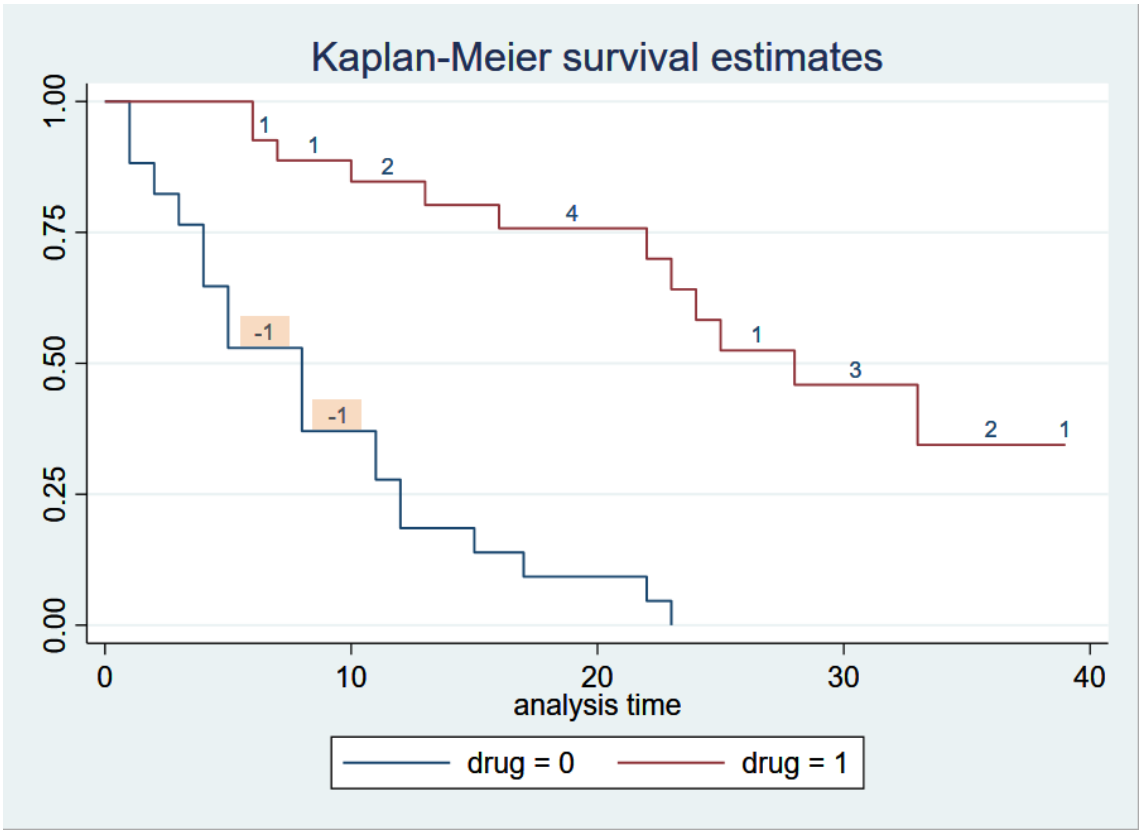
```
.use https://www.stata-press.com/data/r16/drug2b
```

	studytime	died	drug	age	t0	_st	_d	_t	_t0
10	8	0	0	58	0	1	0	8	0
11	8	1	0	52	0	1	1	8	0
12	8	1	0	49	0	1	1	8	0
13	11	1	0	50	0	1	1	11	0
14	11	1	0	55	0	1	1	11	0
15	12	1	0	49	0	1	1	12	0
16	12	1	0	62	5	1	1	12	5
17	15	1	0	51	8	1	1	15	8
18	17	1	0	49	8	1	1	17	8
19	22	1	0	57	0	1	1	22	0
20	23	1	0	52	0	1	1	23	0

1 つ目のサンプルデータと違って、t=5 や t=8 から観察を開始した患者のデータがあります。

- 先程と同様にグラフを作成します。

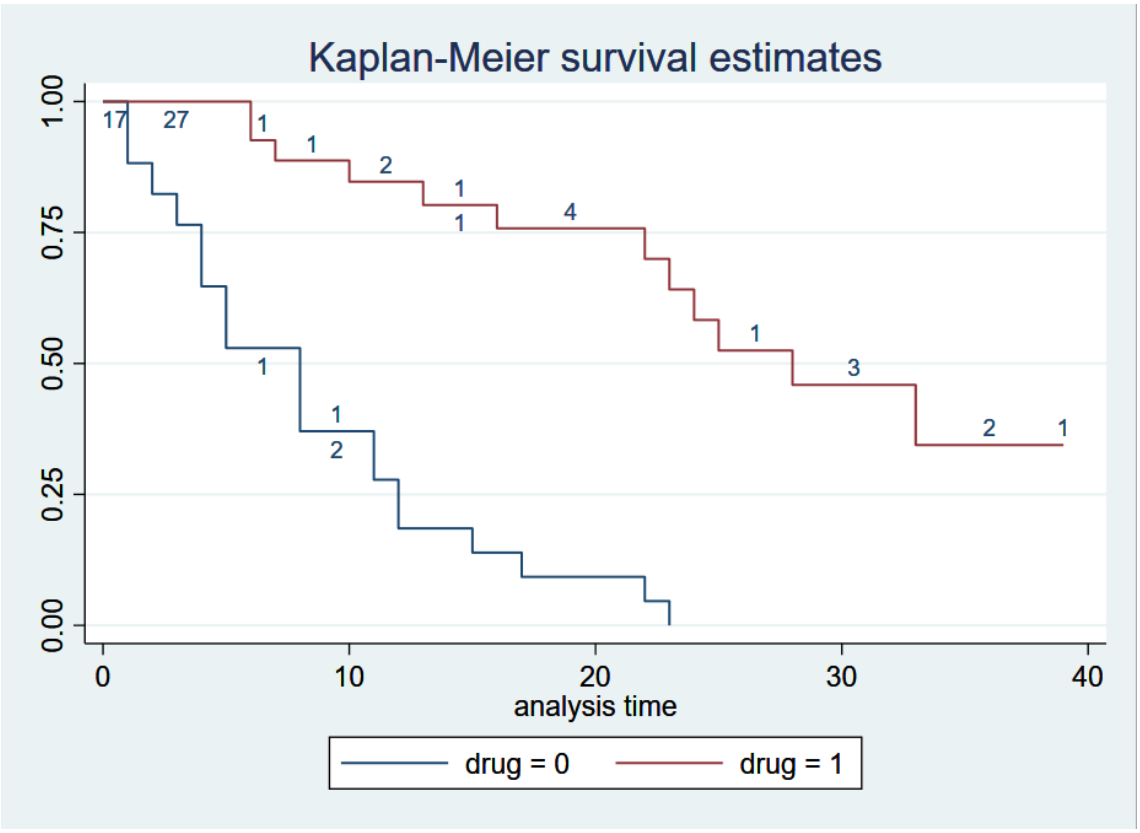
```
.sts graph, by(drug) lost
```



lost オプションは打ち切り人数を表示するオプションなので、遅延組入の人数は負の値として表示されます。この「-1」は、例えば「遅延組入 1 人」や「遅延組入 2 人、打ち切り 1 人」のような状態を表します。

- enter オプションを使用すると、遅延組入と打ち切りを分けて表示できます。

```
.sts graph, by(drug) lost enter
```



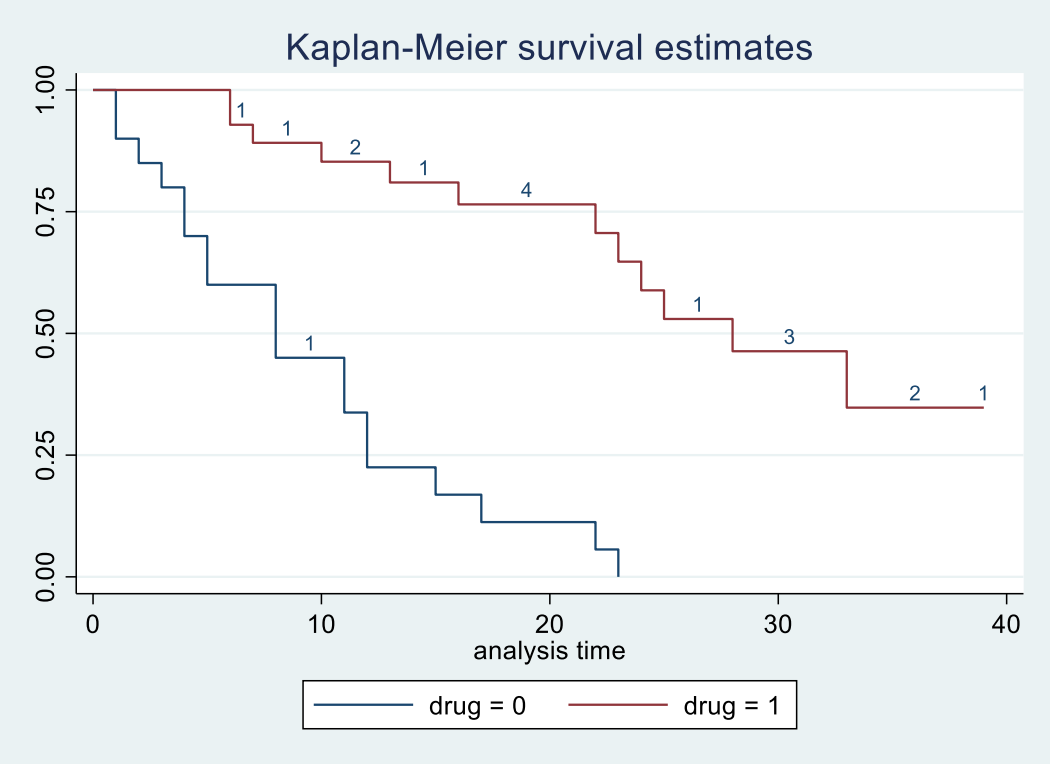
- 遅延組入と打ち切りを分けて表示する方法は、常に最善とは限りません。次のサンプルデータ「drug2c」は、最初の「drug2」と同じ内容のデータです。しかし、記述方法が異なり、時間経過によって変化する共変量（投薬量や体重など）を併記できるように、1 人の患者に対して複数行のデータがあります。遅延組入はありません。

	studytime	died	drug	age	id	_st	_d	_t	_t0
1	1	1	0	61	1	1	1	1	0
2	1	1	0	65	2	1	1	1	0
3	1	0	0	59	3	1	0	1	0
4	2	1	0	59	3	1	1	2	1
5	1.5	0	0	52	4	1	0	1.5	0
6	3	1	0	52	4	1	1	3	1.5
7	2	0	0	56	5	1	0	2	0
8	4	1	0	56	5	1	1	4	2

ID	患者 ID

- サンプルデータ「drug2c」を入手し、lost オプションを付けてグラフを作成します。

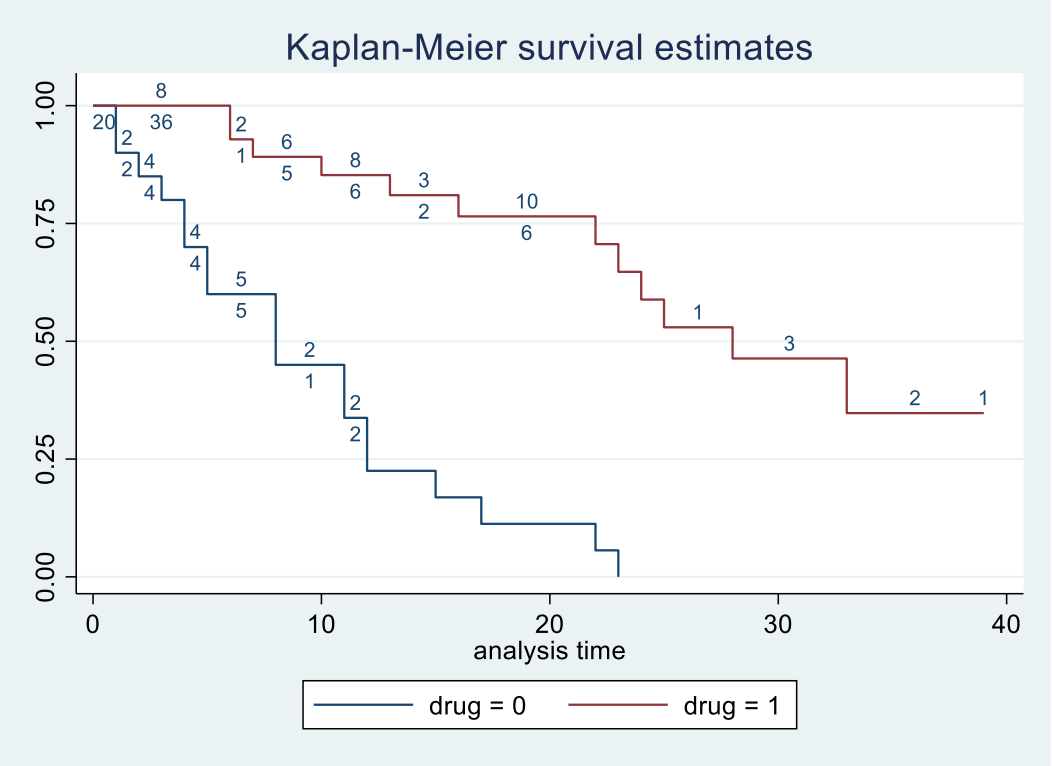
```
.use https://www.stata-press.com/data/r16/drug2c
.sts graph, by(drug) lost
```



このグラフは、「drug2」のグラフと同じです。データの内容も、形式が違うだけで理論的には同じ事象を表しています。

- しかし、enter オプションを使用すると、たくさんの数値が記入されてしまいます。

```
.sts graph, by(drug) lost enter
```

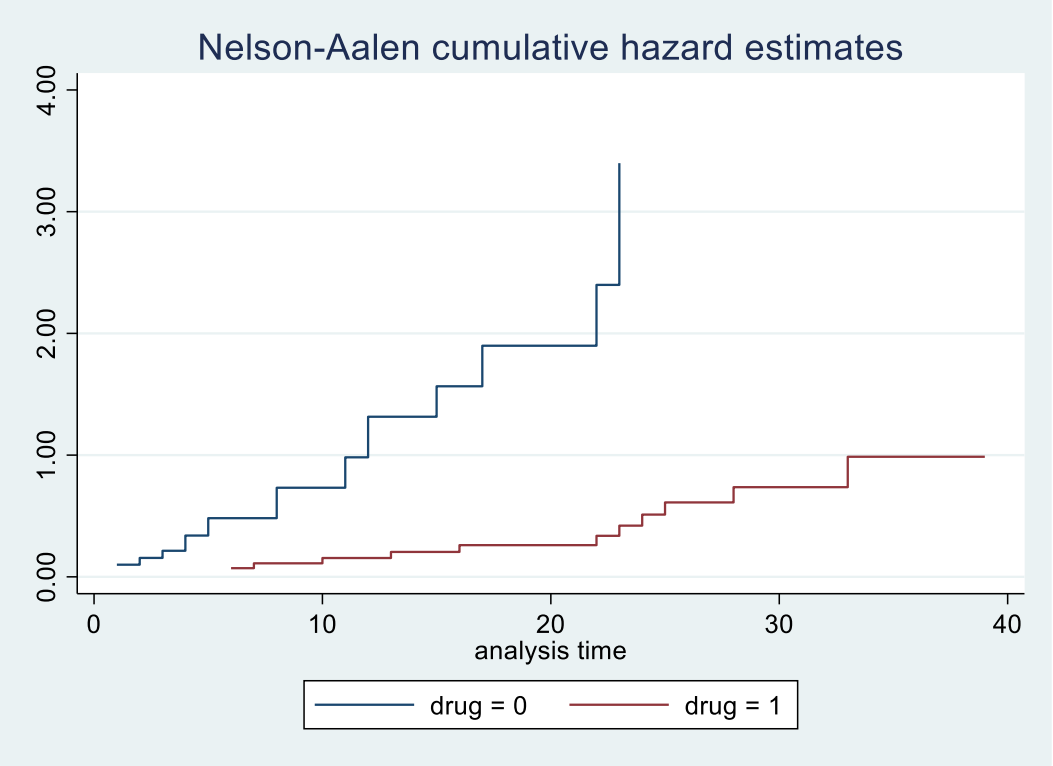


同じ患者 ID のデータが複数行あり、一行ごとに遅延組入・打ち切りとして処理されるためです。遅延組入と打ち切りが区別できない場合は lost と enter のオプションを使うと良かったのですが、このような共変量のあるデータの場合は適しません。

Nelson-Aalen 累積ハザード関数のグラフ作成

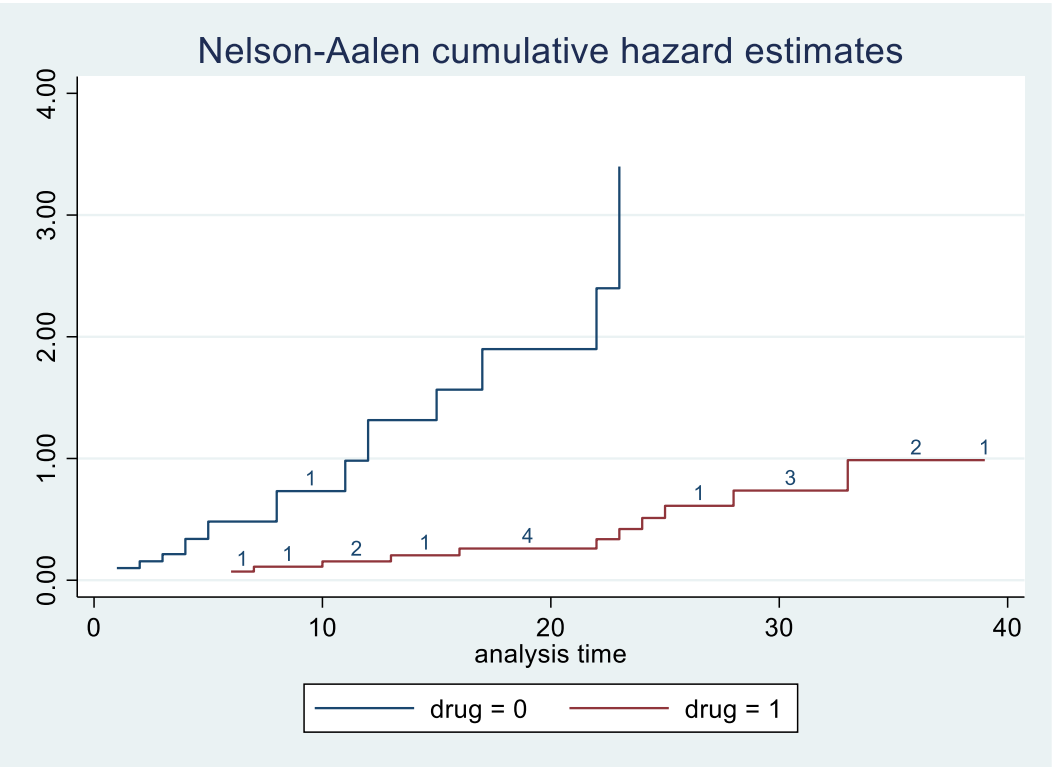
- cumhaz オプションを使用すると、Nelson-Alen 累積ハザード関数を描くことができます。最初と同じサンプル「drug2」を使用して、投薬の有無による累積ハザード関数をグラフにします。

```
.use https://www.stata-press.com/data/r16/drug2
.stset, noshow
. sts graph, cumhaz by(drug)
```



- 打ち切りの人数を表示するグラフは、`lost` オプションで作成できます。

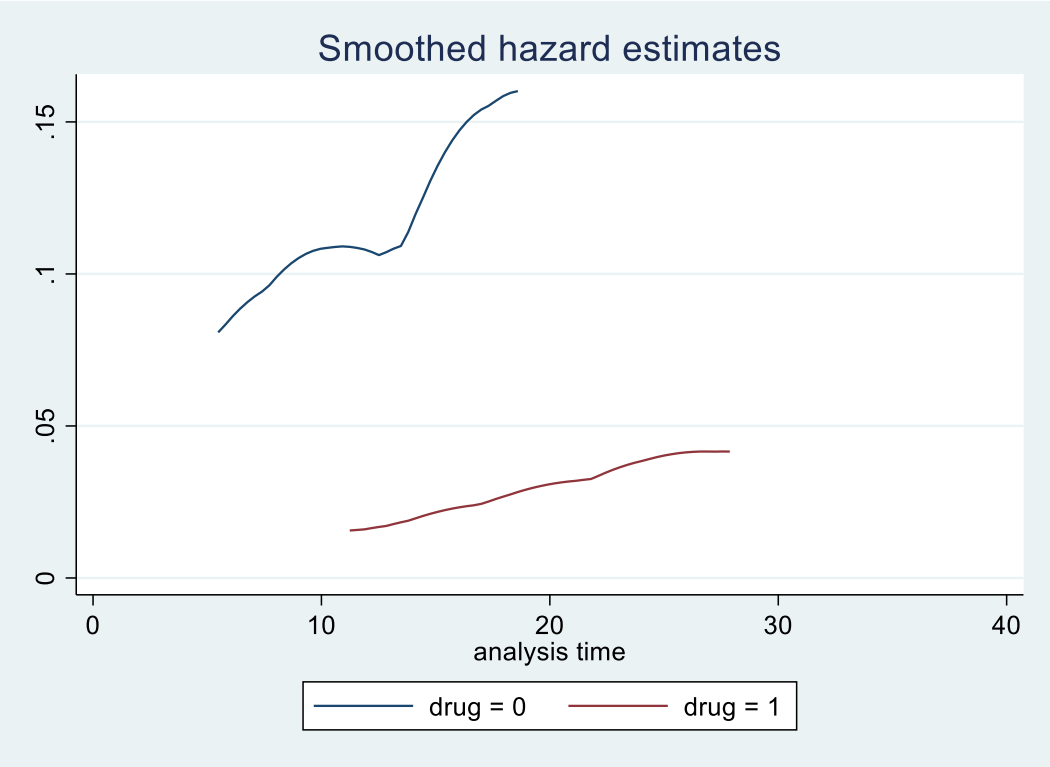
```
. sts graph, cumhaz by(drug) lost
```



ハザード関数のグラフ作成

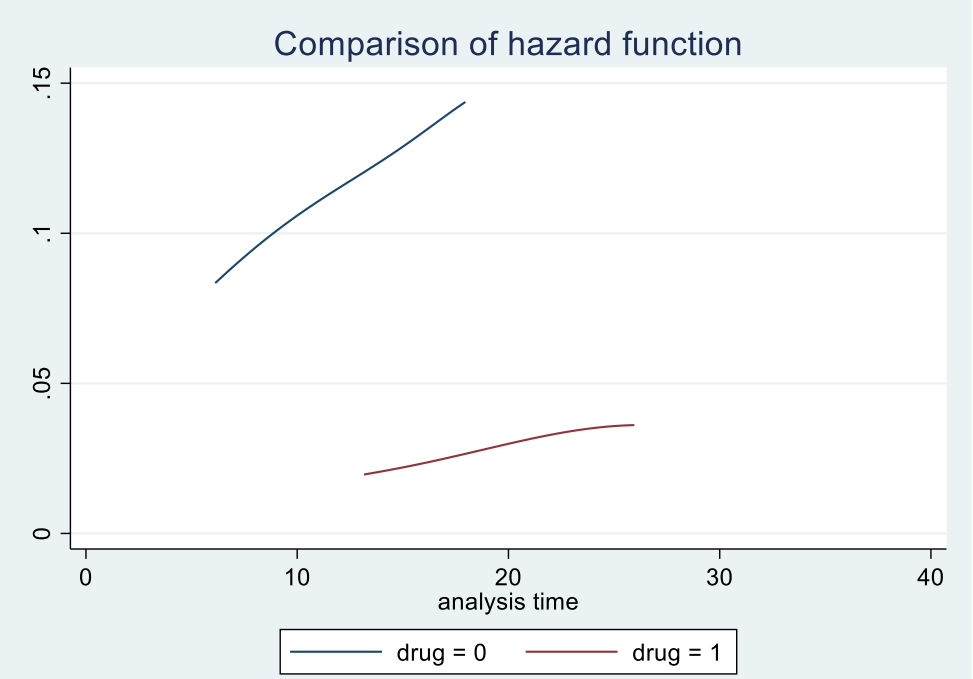
- sts コマンドは、ハザード関数の推定にも使用することができます。下記のグラフは、weighted カーネル平滑化を基にハザード関数を推定しています。カーネル関数とバンド幅の選び方によって結果が変わるため、注意が必要です。

```
.sts graph, hazard by(drug)
```



- 次のコマンドでは、カーネル平滑化とバンド幅を調整しています。titile は、グラフタイトルを変更するコマンドです。

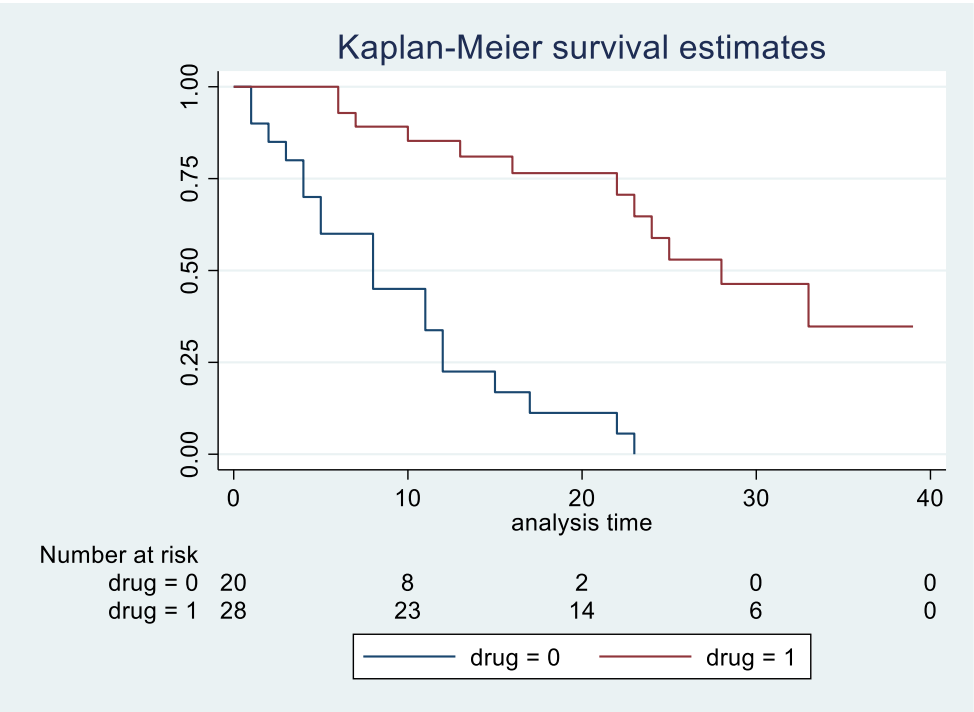
```
.sts graph, hazard by(drug) kernel(gauss) width(5 7) title(Comparison of hazard function)
```



リスクテーブルを追加する

- 以下のコマンドで、リスクテーブルをグラフの下に追加します。

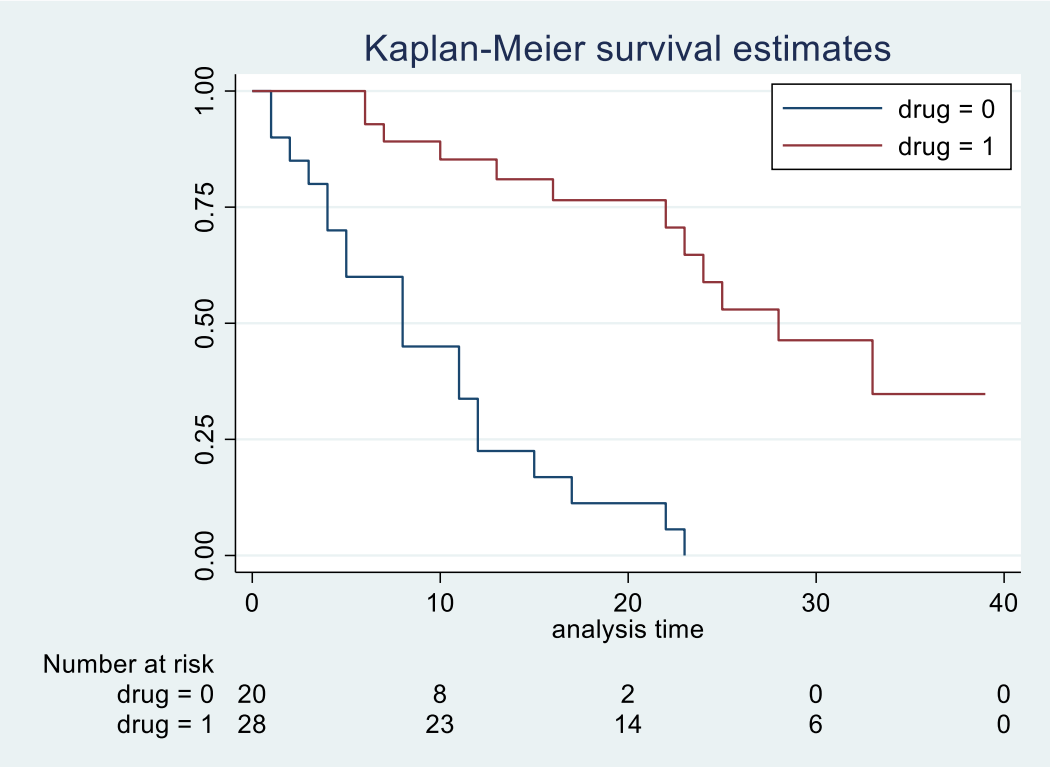
```
.sts graph, by(drug) risktable
```



書式オプションは下記の通りです。

- 凡例の位置を変更します。

```
.sts graph, by(drug) risktable legend(ring(0) position(2) rows(2))
```



- リスクテーブルのラベル列を「Placebo」と「Test drug」に変更します

```
.sts graph, by(drug ) risktable(, order(1 "Placebo" 2 "Test drug"))
```

Number at risk

Placebo	20	8	2	0	0
Test drug	28	23	14	6	0

- ラベル列の文字を左揃えにします。

```
. sts graph, by(drug ) risktable(, order(1 "Placebo" 2 "Test drug")
rowtitle(, justification(left)))
```

Number at risk

Placebo	20	8	2	0	0
Test drug	28	23	14	6	0

- リスクテーブルのタイトルをとラベル列の行頭を揃えます。

```
.sts graph, by(drug ) risktable(, order(1 "Placebo" 2 "Test drug")
rowtitle(, justification(left)) title(, at(rowtitle)))
```

Number at risk					
Placebo	20	8	2	0	0
Test drug	28	23	14	6	0

Cox 比例ハザードモデル

- `stcox` コマンドを使用して、比例 Cox ハザードモデルを適用します。

打ち切りのないデータで Cox 回帰を行う

- 新型のベアリングを搭載した非常用発電機の耐久実験を行います。この実験では、保護回路を無効にして発電機が発火するまで負荷をかけて運転を行います。
- サンプルデータを入手し、データを確認します。

```
. use https://www.stata-press.com/data/r16/kva
. list
```

	failtime	load	bearings
1.	100	15	0
2.	140	15	1
3.	97	20	0
4.	122	20	1
5.	84	25	0
6.	100	25	1
7.	54	30	0
8.	52	30	1
9.	40	35	0
10.	55	35	1
11.	22	40	0
12.	30	40	1

`failtime`	故障するまでの時間（h）
`load`	負荷（kVA）
`bearing`	`1`=新型のベアリング、`0`=旧型のベアリング

12 台の発電機があり、内訳は新型ベアリング搭載機と旧型ベアリング搭載機が各 6 台ずつです。例えば、1 行目のデータは旧式ベアリング搭載機に 15 IVA の負荷をかけて

運転したところ、100 時間後に故障したことを意味します。

- Cox 比例ハザードモデルに当てはめて、発電機の故障は負荷の程度に影響を受けるのか、それともベアリングの型によるのかを調べます。ベアリングの型と負荷の大きさは、ハザード関数の形全体には影響しないものとします。

- 下記のコマンドでモデルを適用します。

```
. stset failtime
（処理結果が出力されます）
. stcox load bearings
```

```
        failure _d:  1 (meaning all fail)
  analysis time _t:  failtime

Iteration 0:   log likelihood = -20.274897
Iteration 1:   log likelihood = -10.515114
Iteration 2:   log likelihood = -8.8700259
Iteration 3:   log likelihood = -8.5915211
Iteration 4:   log likelihood = -8.5778991
Iteration 5:   log likelihood =  -8.577853
Refining estimates:
Iteration 0:   log likelihood =  -8.577853

Cox regression -- Breslow method for ties

No. of subjects =           12                 Number of obs    =          12
No. of failures =           12
Time at risk    =          896
                                               LR chi2(2)       =       23.39
Log likelihood  =    -8.577853                 Prob > chi2      =      0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        load |   1.52647   .2188172     2.95   0.003     1.152576    2.021653
    bearings |  .0636433   .0746609    -2.35   0.019     .0063855    .6343223
------------------------------------------------------------------------------
```

負荷の程度の影響を制御すると、新型ベアリングはハザード率が 6.36%と低く正常作動時間も長くなることが分かります。

- 一度 stcox を適用すると、以降は引数を省略できます。続けて、上記のモデルにハザード比ではなく係数を表示オプション nohr を使用して再表示させます。

```
. stcox, nohr
```

```
Cox regression -- Breslow method for ties

No. of subjects =           12                  Number of obs    =          12
No. of failures =           12
Time at risk    =          896
                                                LR chi2(2)       =       23.39
Log likelihood  =    -8.577853                  Prob > chi2      =      0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
load	.4229578	.1433485	2.95	0.003	.1419999	.7039157
bearings	-2.754461	1.173115	-2.35	0.019	-5.053723	-.4551981

打ち切りのあるデータで Cox 回帰を行う

- 下記のような 48 人の癌患者の服薬データがあります。21 人は薬物療法を行っていて、20 人は行っていません。患者の年齢は 47 才から 67 才までです。このデータを使って、患者が死亡するまでの月数について解析を行います。

	studytime	died	drug	age	_st	_d	_t	_t0
1	1	1	0	61	1	1	1	0
2	1	1	0	65	1	1	1	0
3	2	1	0	59	1	1	2	0
4	3	1	0	52	1	1	3	0
5	4	1	0	56	1	1	4	0
6	4	1	0	67	1	1	4	0
7	5	1	0	63	1	1	5	0
8	5	1	0	58	1	1	5	0
9	8	1	0	56	1	1	8	0
10	8	0	0	58	1	0	8	0

sutdytime	死亡または打ち切りまでの月数
died	1=死亡、0=打ち切り
drug	1=服薬あり、0=服薬なし
age	観察開始時の患者の年齢

- データを入手して stset でデータを整え、サマリーを確認します。

```
. use https://www.stata-press.com/data/r16/drugtr
. stset studytime, failure(died)
.summarize
```

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
studytime	48	15.5	10.25629	1	39
died	48	.6458333	.4833211	0	1
drug	48	.5833333	.4982238	0	1
age	48	55.875	5.659205	47	67
_st	48	1	0	1	1
_d	48	.6458333	.4833211	0	1
_t	48	15.5	10.25629	1	39
_t0	48	0	0	0	0

各変数のデータ数、平均、標準偏差、最小値、最大値が表示されます。

- Cox モデルを適用します。

```
.stcox drug age
```

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

Iteration 0:   log likelihood = -99.911448
Iteration 1:   log likelihood = -83.551879
Iteration 2:   log likelihood = -83.324009
Iteration 3:   log likelihood = -83.323546
Refining estimates:
Iteration 0:   log likelihood = -83.323546

Cox regression -- Breslow method for ties

No. of subjects =           48                 Number of obs    =          48
No. of failures =           31
Time at risk    =          744
                                               LR chi2(2)       =       33.18
Log likelihood  =   -83.323546                 Prob > chi2      =      0.0000
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
drug	.1048772	.0477017	-4.96	0.000	.0430057	.2557622
age	1.120325	.0417711	3.05	0.002	1.041375	1.20526

年齢を制御すると、投薬ありの場合のハザード率が 10.5 %と低く、生存時間が長いことが分かります。

- ハザード率は、変数が 1 単位変わる時の変化を表しています。上記の例では、年齢が 1

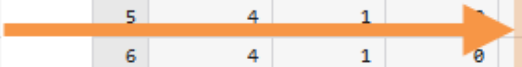
才上がるごとにハザードが 12 %増加しています。

- 5 才年齢が変わる場合のリスクを計算します。年齢を 5 才で 1 単位とするために、新しい変数を作成します。

```
.replace age = age/5
```

```
variable age was byte now float
(48 real changes made)
```

	studytime	died	drug	age
1	1	1	0	61
2	1	1	0	65
3	2	1	0	59
4	3	1	0	52
5	4	1	0	56
6	4	1	0	67
7	5	1	0	63
8	5	1	0	58
9	8	1	0	56
10	8	0	0	58



	studytime	died	drug	age
1	1	1	0	12.2
2	1	1	0	13
3	2	1	0	11.8
4	3	1	0	10.4
5	4	1	0	11.2
6	4	1	0	13.4
7	5	1	0	12.6
8	5	1	0	11.6
9	8	1	0	11.2
10	8	0	0	11.6

- 再度 Cox モデルを適用します。

```
.stcox drug age, nolog
```

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

Cox regression -- Breslow method for ties

No. of subjects =           48                 Number of obs    =          48
No. of failures =           31
Time at risk    =          744
                                               LR chi2(2)       =       33.18
Log likelihood  =   -83.323544                 Prob > chi2      =      0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .1048772   .0477017    -4.96   0.000     .0430057    .2557622
         age |   1.764898   .3290196     3.05   0.002     1.224715    2.543338
------------------------------------------------------------------------------
```

年齢が 5 才上がると、ハザードは 76 %増加します。

タイ（同じ値）のあるデータを扱う

- ハザードモデルを構築するには、タイのない連続的な値を扱うのが理想的です。タイが発生した場合には、部分尤度を計算する必要があります。
- タイのあるデータに対して Cox 部分尤度を計算するために、Stata には 4 つのオプション（brslow, efron, exactm, exactp）が用意されています。

時間で変化する共変量のあるデータで Cox 回帰を行う

- スタンフォード心臓移植プログラムの心臓移植データを例にします。これには、実際に心臓移植を受けた患者と受けられなかった患者（合計 103 人）のデータが含まれています。心臓移植を待っている間に死亡した患者や待機をやめた患者もいますが、67 % の患者は移植を受けました。
- データを入手します。

```
. use https://www.stata-press.com/data/r16/stan3, clear
```

	id	year	age	died	stime	surgery	transplant	wait	posttran	t1	_st	_d
1	1	67	30	1	50	0	0	0	0	50	1	1
2	2	68	51	1	6	0	0	0	0	6	1	1
3	3	68	54	0	16	0	1	1	0	1	1	0
4	3	68	54	1	16	0	1	1	1	16	1	1
5	4	68	40	0	39	0	1	36	0	36	1	0
6	4	68	40	1	39	0	1	36	1	39	1	1
7	5	68	20	1	18	0	0	0	0	18	1	1
8	6	68	54	1	3	0	0	0	0	3	1	1
9	7	68	50	0	675	0	1	51	0	51	1	0
10	7	68	50	1	675	0	1	51	1	675	1	1

id	患者 ID
year	心臓移植プログラムに参加した年
age	患者の年齢
died	1=死亡、0=打ち切り
stime	生存時間（日）
surgey	1=外科手術あり、0=外科手術なし（CABG など）
transplant	1=心臓移植実施、0=心臓移植なし
wait	移植までの待機時間
posttran	post-transplant 1=新しい心臓を受け取った、0=受け取らなかった

- stset でデータを整え、Cox モデルを適用します。

```
. stset t1, failure(died) id(id)
```

（処理結果が出力されます）

```
. stcox age posttran surg year
```

```
        failure _d:  died
  analysis time _t:  t1
                id:  id

Iteration 0:   log likelihood = -298.31514
Iteration 1:   log likelihood =  -289.7344
Iteration 2:   log likelihood = -289.53498
Iteration 3:   log likelihood = -289.53378
Iteration 4:   log likelihood = -289.53378
Refining estimates:
Iteration 0:   log likelihood = -289.53378

Cox regression -- Breslow method for ties

No. of subjects =          103                 Number of obs    =          172
No. of failures =           75
Time at risk    =      31938.1
                                               LR chi2(4)       =        17.56
Log likelihood  =   -289.53378                 Prob > chi2      =       0.0015

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.030224   .0143201     2.14   0.032     1.002536    1.058677
    posttran |   .9787243   .3032597    -0.07   0.945     .5332291    1.796416
     surgery |   .3738278    .163204    -2.25   0.024     .1588759       .8796
        year |   .8873107    .059808    -1.77   0.076     .7775022    1.012628
------------------------------------------------------------------------------
```

高齢の患者ほどハザード率が高く、手術を受けた患者のハザード率が低いことが分かります。また、最終的に心臓移植を受けたかどうかによって、ハザード率は大きく変わりません。

継続的に時間変化する共変量のあるデータで Cox 回帰を行う

- 基本的な Cox 回帰は、次の式で表されます。
  $\mathrm{h}(t) = h_0(t)exp(\beta_1 x_1 + \cdots + \beta_k + x_k)$
- 変数が$z_i(t) = z_i g(t)$に変化したとすると、式は下記のようになります。
  $\mathrm{h}(t) = h_0(t)exp\{\beta_1 x_1 + \cdots + \beta_k + x_k + g(t)(\gamma_1 z_1 + \cdots + \gamma_m z_m)\}$
- $z_1, \cdots\cdots, z_m$が時間変化する共変量し推定に影響を及ぼす場合、回帰係数γ_1と共変量$g(t)z_i$は現在の関数です。
- 変数$z_1, \cdots\cdots, z_m$を決定するには tvc(*varlist*)オプションを使用します。$g(t)$におけるtは分析時間なので、$g(t)$は texp(*exp*)オプションで計算できます。例えば、$g(t) = \log(t)$を

計算する場合は下記のように入力します。_t は tset によって計算されています。

```
texp(log(_t))
```

- Cox 回帰はイベント発生時の部分尤度を基に計算されているので、イベント発生時に **stsplit** を使って分割し、手動で時間変化のある共変量を生成することでも計算できます。
- **stsplit** を使ってイベント発生の多い大きなデータセットを処理すると大量のメモリを消費するため、その場合は **tvc()**や **texp()**を使用します。