Computer Appendix：コンピュータを用いた生存時間解析

本Appendixでは，本文で述べた生存時間解析を実行するコンピュータプログラム例を提供します．Appendixでは現在利用可能なコンピュータパッケージすべてを紹介する訳ではなく，最も広く使用されている4つのパッケージの類似点と相違点を紹介します．紹介するソフトウェアパッケージはStata（バージョン10.0），SAS（バージョン9.2），SPSS（PASW 18），Rです．これらのパッケージに関する詳細は説明しないので，詳細につきましては，各プログラムのヘルプ機能を参照してください．

データセット

Appendixで紹介するプログラム文や出力は，薬物常用者データセットを用いた逐次的な生存時間解析によるものです．Appendixでは，それ以外にも再発イベントの解析のために「膀胱がん」データセットも使用します．「薬物常用者」と「膀胱がん」データは，私たちのウェブサイトからダウンロード可能です（**http://web1.sph.emory.edu/dkleinb/surv3.htm**）．このウェブサイトから，本書で例や練習問題に取り上げた他のデータセットも入手できます．このウェブサイトでは，データを次の5つの形式で提供しています．（1）Stataデータセット（拡張子**.dta**），（2）SASデータセット（拡張子**.sas7bdat**），（3）SPSSデータセット（拡張子**.sav**），（4）Rデータセット（拡張子**.rda**），（5）テキストデータセット（拡張子**.dat**）です．

薬物常用者データセット（addicts.dat）

1991年Caplehornらによるオーストラリア試験で，ヘロイン常用者にメタドン治療を実施する2施設を，患者のメタドン治療継続時間により比較するものです．患者の生存時間を，患者が施設から離脱するか，打ち切りまでの時間（日）と定めました．2施設には患者への院内方針に違いがありました．変数の定義は以下の通りです．

ID – 患者ID
SURVT – 患者がCLINICから離脱するか，または打ち切りまでの時間（日）．
STATUS – 患者がCLINICから離脱（code = 1）か，打ち切り（code = 0）かを示す．
CLINIC – 患者がメタドン治療を受けたCLINIC（code = 1，2）を示す．
PRISON – 患者に服役歴がある（code = 1），ない（code = 0）を示す．
DOSE – 患者の最大メタドン用量（mg/日）に関する連続変数．

膀胱がんデータセット（bladder.dat）

膀胱がんデータセットには，86名の経尿道的切除術後の膀胱がん患者に繰り返し起こる再発を追跡した再発イベントアウトカムの情報が含まれています（Byar and Green，1980）．興味ある曝露は，チオテパ薬物治療の効果です．調整変数は最初の腫瘍数と腫瘍の大きさです．CPアプローチのデータレイアウトになっています．変数の定義は以下の通りです．

ID –	患者ID（同一患者に複数のオブザベーションが存在する場合がある）.
EVENT –	患者に腫瘍が発生（code = 1）したか，しなかった（code = 0）かを示す.
INTERVAL –	患者内の時間区間の順番（code = 1：当該対象者の1番目の時間区間，code = 2：当該対象者の2番目の時間区間など）
START –	各区間の開始時間（月）
STOP –	各区間におけるイベント時間（月）または打ち切り時間（月）
TX –	治療（code = 1：チオテパ治療，code = 0：プラセボ）
NUM –	最初の腫瘍数
SIZE –	最初の腫瘍の大きさ（cm）

ソフトウェア

ここからは，本書で紹介した各種の生存時間解析を実行するために必要なプログラムや出力の詳細な説明を始めます．下記の4つのソフトウェアパッケージごとにセクションを設けます．

A. Stata
B. SAS
C. SPSS
D. Rソフトウェア

各セクションは完結型になっていますので，読者は関心のある統計パッケージを選んで読むことができます．

A. Stata

Stataによる解析を行うためには，適切なコマンドをStata Command windowかStata Do-file Editor windowに指定します．生存時間解析に用いる主なコマンドを以下に挙げます．Stataでは大文字と小文字の区別があり，コマンドは小文字を使用してください．

stset – メモリ内のデータが生存データであることを宣言します．「time-to-event」変数，「status」変数，他の関連する生存変数を定義するのに使用します．stで始まるその他のStataコマンドも，ここで定義された変数を利用します．

sts list – Kaplan-Meier（KM）またはCox調整生存推定値をResults windowに出力する．デフォルトはKM生存推定値です．

sts graph – Kaplan-Meier（KM）の生存推定値のプロットを作成する．このコマンドは，Cox調整生存推定値のプロットにも使用できます．

sts generate – 作業データセット中にKaplan-MeierまたはCox調整生存推定値を格納する変数を作成します．

sts test – 生存関数の層間の同等性に関する検定を実行します．

stphplot – 比例ハザード（PH）性を確認するための，対数時間vs. 対数－対数生存プロットを生成します．Kaplan-Meier対数－対数生存プロットまたはCox調整対数-対数生存プロットを指定できます．

stcoxkm – KM生存プロットとCox調整生存プロットを同じグラフ上に作成します．

stcox – Cox比例ハザードモデル，層化Coxモデル，拡張Coxモデル（例えば時間依存性共変量を含む）を実行します．

stphtest – Schoenfeld残差に基づく比例ハザード性に関する検定を実行します．このコマンドを使用するためには，stcoxコマンドとschoenfeld()オプションを用いて事前にCoxモデルを実行しておく必要があります．

streg – パラメトリック生存モデルを実行します．

Stataを開くと，4つのwindowが現れます．それらwindowは，Stata Command，Stata Results，Review，Variablesとラベルされています．解析用の作業データセットを選択するためには，File→Openをクリックします．データセットを選択すると，変数の名前がVariables windowに表示されます．コマンドはStata Command windowに入力します．リターンキーを押すと，コマンドによって作成された出力がResults windowに表示されます．Review windowには，Stataセッションにおけるすべてのコマンド実行履歴が保存されます．Review windowのコマンドは，ユーザーが望むように保存，コピー，編集できます．Review windowのコマンドをダブルクリックしてコマンドを実行することも可能です．Stataツールバーのlogボタンをクリックすることにより，コマンドをファイルに保存することもできます．

コマンドを実行する別の方法には，コマンドをDo-file Editorに入力あるいは貼り付けるやり方もあります．Do-file Editor windowを開くには，Window→Do-file Editorをクリックするか，Stataツールバーの Do-file Editorボタンをクリックします．Tools→Doをクリックすると，コマンドがDo-file Editorから実行されます．コマンドをDo-file Editorから実行する利点は，Stata Command windowから実行する場合は，コマンドを一度に1つずつ入力し実行しなければなりませんが，Do-file Editorでは複数のコマンドが一度に実行可能です．

Do-file Editorは，SASのプログラムエディタと同様の機能を持っています．実際，Do-file Editor windowに**#delim**を入力すると，デフォルトの改行コード(CR)ではなくセミコロンが，Stataのステートメントの終わりを示す区切り文字になります(SASのように)．

Stataによる生存時間解析では，以下を取り上げます．

1. 生存関数(未調整)の推定と層間での比較
2. グラフを用いた比例ハザード性の検討
3. Cox比例ハザード(PH)モデルの実行
4. 層化Coxモデルの実行
5. 統計的検定による比例ハザード(PH)仮定の評価
6. Cox調整生存曲線の作成
7. 拡張Coxモデルの実行
8. パラメトリックモデルの実行
9. frailtyモデルの実行
10. 再発イベントのモデル構築

まずFile→Openをクリックし，Stata薬物常用者データセット“**addicts.dta**”を選択することから始めます．それが完了すると，「**use “addicts.dta”, clear**」コマンドがReview windowとResults windowに表示されます．これは，薬物常用者データセットがStataのメモリ内で利用可能になったことを示します．

生存時間解析を行うためには，time-to-event変数とstatus変数を指定する必要があります．生存時間解析コマンドごとに変数を指定する代わりに，Stataでは**stset**コマンド用いて一度のプログラミングで指定することができます．**st**で始まるすべての生存解析用コマンドは，このデータセットがメモリ上で利用可能状態にある限り，**stset**によって定義された生存時間変数を利用します．薬物常用者データの生存時間変数を定義するコードは以下の通りです．

```
stset survt, failure(status==1) id(id)
```

stsetの後ろにはtime-to-event変数名がきます．Stataコマンドのオプションはカンマの後に続けます．最初のオプションは，イベント(またはfailure)の有無を示す変数を指定し，打ち切りではなくイベントの値を指定します．このオプションがない場合は，Stataではすべてのオブザベーションにイベントがある(つまり，打ち切りがない)となります．2つの等号“==”は両辺が等しいという条件式に用いられ，1つの等号“=”は右辺の値を左辺に割り当てるときに用います．次のオプションはIDという名前の変数をid変数に指定するものです．これは薬物常用者データセットでは不要です．なぜならば．1患者1オブザベーションなので，1人の患者に複数のオブザベーションが存在するようなクラスタが存在しないからです．

しかしながら，1対象に対して複数のオブザベーションと複数のイベントがある場合は(クラスタ)，Stataはクラスタデータに対して適切なロバスト分散推定値を与えます．

stsetコマンドは，4つの新しい変数をデータセットに追加します．Stataでの定義変数は以下となります．

_t – time-to-event変数
_d – status変数(1 = イベント，0 = 打ち切り)
_t0 – 時間変数の始まりの時間．デフォルトでは，すべてのオブザベーションが時間0から始まります．
_st – 解析に用いるオブザベーションを指定します．デフォルトでは，すべてのオブザベーションが解析に用いられます(1がコードされている)．

最初の10オブザベーションを出力画面に表示するには，以下のコマンドを用います．

```
list in 1/10
```

stdesコマンドは生存時間の記述統計量を与えます(出力を以下に示す)．

```
stdes
```

```
      failure _d:  status == 1
analysis time _t:  survt
             id:  id

                           |---------per subject----------|
Category             total       mean   min   median    max
------------------------------------------------------------------------
no. of subjects        238
no. of records         238          1     1        1      1

(first) entry time                  0     0        0      0
(final) exit time             402.5714     2    367.5   1076

subjects with gap        0
time on gap if gap       0          .     .        .      .
time at risk         95812   402.5714     2    367.5   1076

failures               150   .6302521     0        1      1
------------------------------------------------------------------------
```

コマンド**strate**と**stir**は，指定した変数のカテゴリ間で発生率を比較するためのものです．**strate**コマンドがCLINIC別の発生率を示し，**stir**コマンドは発生率の比と差を示します．

以下のコマンドを1つずつ入力してみてください(出力省略).

```
strate clinic
stir clinic
```

これから記述する生存時間解析では, 前述したように**stset**コマンドを薬物常用者データセットに対して実行済みであると前置きします.

1. 生存関数(未調整)の推定と層間の比較

Kaplan-Meier生存推定値を得るには, **sts list**コマンドを用います. コードと出力は以下の通りです.

```
sts list
```

```
      failure _d:  status == 1
analysis time _t:  survt
              id:  id

      Beg.          Net    Survivor    Std.
Time  Total  Fail  Lost    Function   Error  [95% Conf. Int.]
-------------------------------------------------------------------------
   2    238     0     2      1.0000       .          .        .
   7    236     1     0      0.9958  0.0042     0.9703   0.9994
  13    235     1     0      0.9915  0.0060     0.9665   0.9979
  17    234     1     0      0.9873  0.0073     0.9611   0.9959
  19    233     1     0      0.9831  0.0084     0.9555   0.9936
  26    232     1     0      0.9788  0.0094     0.9499   0.9911
  28    231     0     2      0.9788  0.0094     0.9499   0.9911
  29    229     1     0      0.9745  0.0103     0.9442   0.9885
  30    228     1     0      0.9703  0.0111     0.9386   0.9857
  33    227     1     0      0.9660  0.0118     0.9331   0.9828
     .                            .
     .                            .
     .                            .
 905      8     0     1      0.1362  0.0364     0.0748   0.2159
 932      7     0     2      0.1362  0.0364     0.0748   0.2159
 944      5     0     1      0.1362  0.0364     0.0748   0.2159
 969      4     0     1      0.1362  0.0364     0.0748   0.2159
1021      3     0     1      0.1362  0.0364     0.0748   0.2159
1052      2     0     1      0.1362  0.0364     0.0748   0.2159
1076      1     0     1      0.1362  0.0364     0.0748   0.2159
```

時間ごとにCLINICごとの生存推定値を横に並べて比較したい場合は, **by**()と**compare**()オプションを使用します. コードと出力は以下の通りです.

sts list, by(clinic) compare at(0 20 to 1080)

```
        failure _d:  status == 1
  analysis time _t:  survt
                id:  id

                        Survivor Function
clinic                      1           2
-------------------------------------------
time          0        1.0000      1.0000
             20        0.9815      0.9865
             40        0.9502      0.9595
             60        0.9189      0.9459
             80        0.9000      0.9320
            100        0.8746      0.9320
            120        0.8681      0.9179
            140        0.8422      0.9038
            160        0.8093      0.8753
            180        0.7690      0.8466
            200        0.7420      0.8323
            220        0.6942      0.8179
                  .
                  .
                  .
            840        0.0725      0.5745
            860        0.0543      0.5745
            880        0.0543      0.5171
            900        0.0181      0.5171
            920             .      0.5171
            940             .      0.5171
            960             .      0.5171
            980             .      0.5171
           1000             .      0.5171
           1020             .      0.5171
           1040             .      0.5171
           1060             .      0.5171
           1080             .           .
```

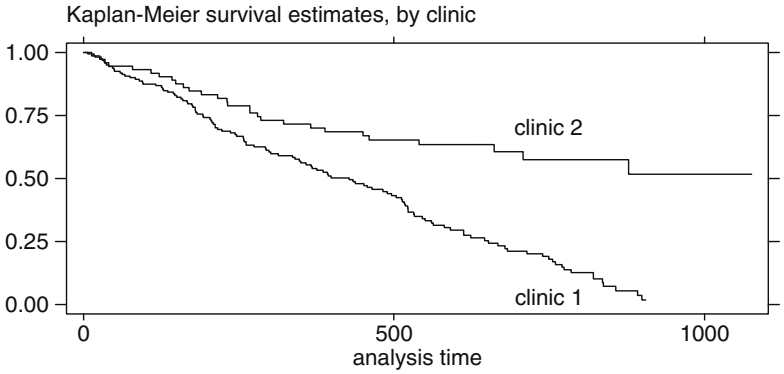
CLINIC = 2の生存率がCLINIC = 1よりも高くなっています．**compare()**オプションを使用して，その他の生存時間について調べることもできます．

Kaplan-Meier生存関数(時間に対する)をグラフ化するには，以下のコードを使用します．

sts graph

CLINIC別のKaplan-Meier生存関数のグラフを与えるコードと出力は，以下の通りです．

```
sts graph, by(clinic)
```



failureオプションは，生存関数ではなくfailure関数（累積リスク）をグラフ化します（1から0ではなく，0から1）．コードは以下の通りです（出力は省略）．

```
sts graph, by(clinic) failure
```

変数CLINICについてログランク検定を実行するコード（および出力）は以下の通りです．

```
sts test clinic
```

```
      failure _d:  status == 1
analysis time _t:  survt
             id:  id

Log-rank test for equality of survivor functions

           Events    Events
clinic   observed  expected
---------------------------------
1             122     90.91
2              28     59.09
---------------------------------
Total         150    150.00

        chi2(1) =     27.89
        Pr>chi2 =    0.0000
```

Wilcoxon，Tarone-Ware，Peto，Flemington-Harringtonの各種検定を指定することもできます．これらの検定はログランク検定の変形であり，オブザベーションの重みが異なります．Wilcoxon検定では，j番目のfailure時間をn_i（at risk数）で重み付けします．Tarone-Ware検定では，j番目のfailure時間を$\sqrt{n_j}$で重み付けします．Peto検定はj番目のfailure時間を，すべてのグループを併合して計算した生存推定値 $\hat{s}(t_j)$で重み付けします．この生存推定$\hat{s}(t_j)$はKaplan-Meier生存推定値に似た値ですが，完全に等しくはありません．Flemington-Harington検定のj番目のfailure時間の重みは，全群によるKaplan-Meier生存推定値 $\hat{s}(t)$と2つの引数からなる$\hat{s}(t_{j-1})^p[1-\hat{s}(t_{j-1})]^q$の形を取ります．コードは以下の通りです（出力は省略）．

```
sts test clinic, wilcoxon
sts test clinic, tware
sts test clinic, peto
sts test clinic, fh(1,3)
```

sts testコマンドのデフォルトの検定はログランク検定です．どの重みの検定統計量を使うかの選択（例えば，ログランクあるいはWilcoxon）は，どの検定の検出力が最も高いと考えられるか，つまり，帰無仮説が棄却されやすいかによります．ただし，事後的に望ましいp値を釣り上げるのではなく，どの統計的検定を使うのかを事前に決めておかなければなりません．

CLINICに関する層化ログランク検定（PRISONによる層別）は，strataオプションで可能です．層化アプローチを用いる場合，層内のグループごとにイベントの実測値から期待値を引いたものを全failure時間を通して合計し，さらにすべての層の合計をとります．コードは以下の通りです（出力は省略）．

```
sts test clinic, strata(prison)
```

sts generateコマンドを用いれば，作業用データセットにKaplan-Meier生存推定値を含む新しい変数を作成することができます．以下のコードで，CLINIC別のKM生存推定値を含む，SKM（変数名はユーザーが決めます）と呼ぶ新しい変数を定義します．

```
sts generate skm=s, by(clinic)
```

ltableコマンドは生命表を生成します．生命表はKaplan-Meierの代替アプローチであり，個人レベルのデータがない場合には特に役に立ちます．以下のコードおよび出力は，**interval()**オプションにより指定した時点（日）のCLINIC別の生命表生存推定値を与えます．

ltable survt status, by(clinic) interval(60 150 200 280 365 730 1095)

Interval	Beg. Total	Deaths	Lost	Survival	Std. Error	[95% Conf.	Int.]
clinic = 1							
0	.	163	13	4	0.9193	0.0215	0.8650	0.9523
60	150	146	14	6	0.8293	0.0300	0.7609	0.8796
150	200	126	13	3	0.7427	0.0352	0.6661	0.8043
200	280	110	17	2	0.6268	0.0393	0.5446	0.6984
280	365	91	10	6	0.5556	0.0408	0.4720	0.6313
365	730	75	41	15	0.2181	0.0367	0.1509	0.2934
730	1095	19	14	5	0.0330	0.0200	0.0080	0.0902
clinic = 2							
0	.	75	4	2	0.9459	0.0263	0.8624	0.9794
60	150	69	5	3	0.8759	0.0388	0.7749	0.9334
150	200	61	3	0	0.8328	0.0441	0.7242	0.9015
200	280	58	5	1	0.7604	0.0508	0.6429	0.8438
280	365	52	3	2	0.7157	0.0540	0.5943	0.8065
365	730	47	7	23	0.5745	0.0645	0.4385	0.6890
730	1095	17	1	16	0.5107	0.0831	0.3395	0.6584

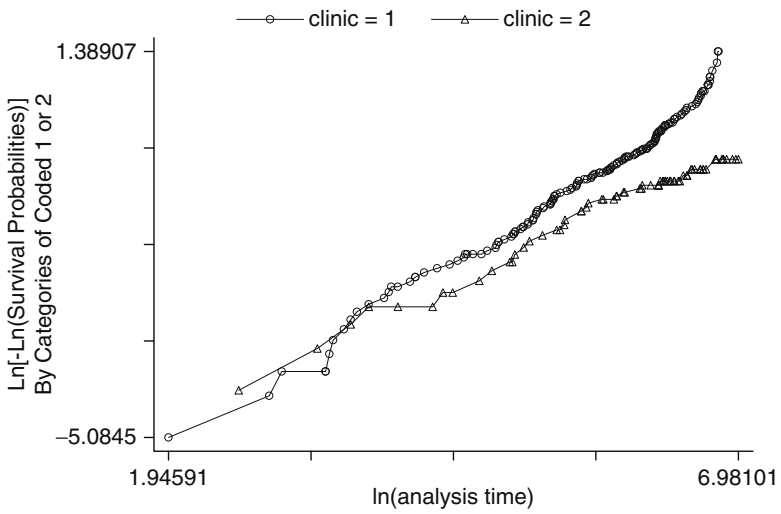
2. グラフを用いた比例ハザード性の検討

変数CLINICの比例ハザード性を検討するグラフを用いた3つのアプローチを紹介します.

1）対数(－対数)Kaplan-Meier生存推定値(CLINICで層別)と時間(または時間の(－対数)をプロット
2）対数-対数Cox調整生存推定値(CLINICで層別)と時間をプロット
3）Kaplan-Meier生存推定値とCox調整生存推定値を同じグラフにプロット

これら3つのアプローチは主観的なところもありますが，うまくいけば有益な情報が得られます．最初の2つのアプローチは, CLINICの各水準間で対数(－対数)生存曲線が平行であるかを確認します．3番目のアプローチは，Cox調整生存曲線(CLINICを層別ではなく予測変数とする)がKM曲線に近いかどうかを調べるものです．言い換えれば，比例ハザードモデルから(Coxから)得られた予測値がKMを用いた実測値に近いかということです.

最初の2つのアプローチは**stphplot**コマンドを使用し，3番目のアプローチは**stcoxkm**コマンドを使用します. 対数(－対数)Kaplan-Meier生存プロットのコードおよび出力は以下の通りです.

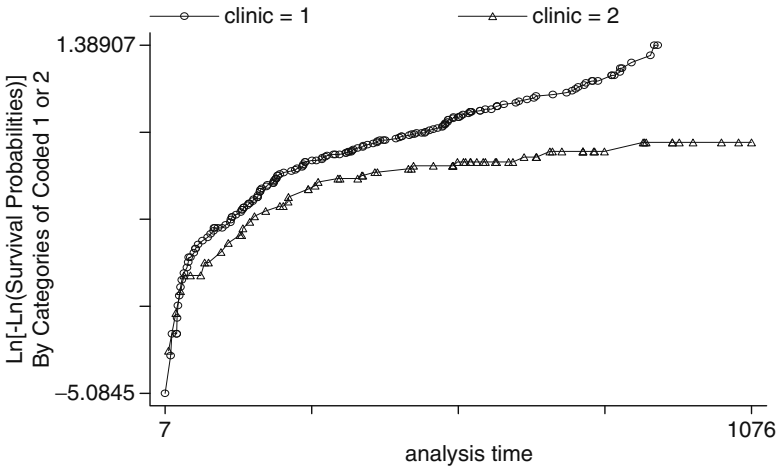


CLINIC = 1に関しては，グラフの左側で曲線が跳ね上がるところがありますが，これはこれら時点付近のイベント数が少ないことに起因します．また，時間の後半部分（グラフの右側）ではプロットが乖離しているようにみえます．上記のコードにある**nonegative**オプションは，デフォルトの－log(－log)曲線ではなくlog(－log)曲線を要求します．どちらの曲線にするのかはユーザーの好みになります．このオプションを使用しないと，グラフは右肩上がりではなく右肩下がりになります．

Stataは（SASも同様ですが），デフォルトでは，横軸に生存時間ではなくlog（生存時間）をとります．曲線の平行性を確認するということに関しては，横軸にlog（生存時間）をとろうが生存時間をとろうが問題にはなりません．しかしながら，横軸にlog（生存時間）を取った場合，対数（－対数）生存曲線が直線になれば，time-to-event変数がWeibull分布に従うことが示唆されます．その直線の傾きが1になるとき，生存時間変数（SURVT）は指数分布（Weibull分布の特殊な場合）に従うことが示唆されます．このような場合は，パラメトリック生存モデルが使用可能となります．

対数生存時間ではなく，生存時間を横軸にした対数（－対数）生存曲線のグラフの方が，視覚的により多くの情報を得られる可能性があります．**nolntime**オプションを使用すれば，生存時間を横軸にとることができます．コードと出力は以下の通りです．

stphplot, by(clinic) nonegative nolntime

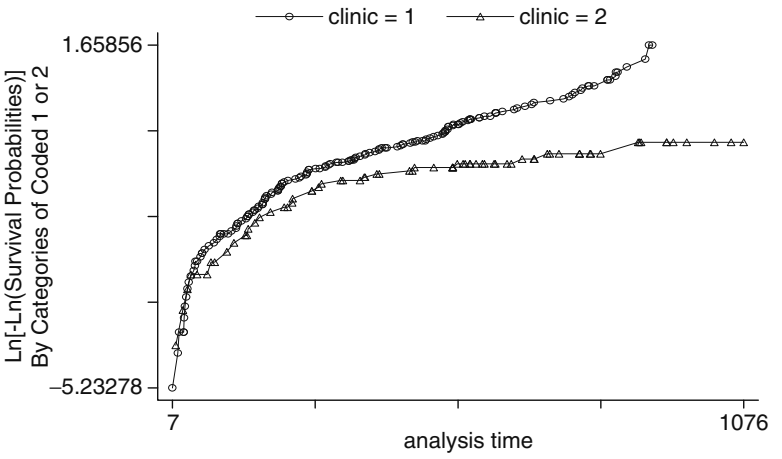


このグラフは，2本の曲線が時間の経過に伴って離れることを示唆しています．

stphplotコマンドを使用すれば，対数(－対数)Cox調整生存推定値を得ることができます．コードは以下の通りです．

stphplot, strata(clinic) adjust(prison dose) nonegative nolntime

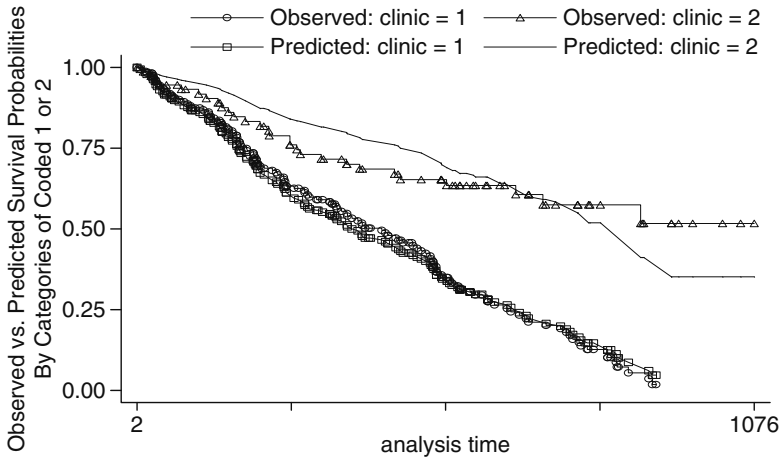
対数(－対数)曲線は，変数CLINICに関する層化Coxモデルを使用し，PRISONとDOSEで調整したものです．調整曲線の作成には，PRISONとDOSEの平均値が用いられます．出力は以下の通りです．



Cox調整曲線はKM曲線にかなり似ています．

stcoxkm コマンドは，同じグラフ上にプロットしたKaplan-Meier生存推定値とCox調整生存推定値の比較に使用します．コードと出力は以下の通りです．

stcoxkm, by(clinic)



KM曲線と調整生存曲線は，CLINIC = 1では非常に似通っていますが，CLINIC = 2ではそれほどでもありません．このグラフを用いたアプローチから，比例ハザード性からの弱い乖離があることを示唆しています．予測値はCLINICで調整したCoxモデル値であり，比例ハザード性を仮定しています．CLINICで調整しているのに，CLINIC別の予測生存曲線は平行ではありません．Cox調整値が平行となるのは，生存曲線ではなく対数(－対数)生存曲線の方です．

これと同じグラフを用いた解析をPRISONやDOSEについても行うことができます．ただしDOSEは連続変数であるため，カテゴリ化する必要があります．

3. Cox比例ハザードモデルの実行

Cox比例ハザードモデルに関する重要な仮定は，すべての共変量パターン間でハザードが比例するということです．最初に取り上げるモデルでは，3つの共変量PRISON，DOSE，CLINICが含まれます．このモデルでは，これら3つの共変量による共変量パターンすべてについて同じ基準ハザードを仮定しています．言い換えれば，それぞれの共変量に関して比例ハザード性を仮定しています(おそらく正しくありません)．コードと出力は以下の通りです．

stcox prison clinic dose, nohr

```
        failure _d:  status == 1
  analysis time _t:  survt
                id:  id

Iteration 0:   log likelihood =  -705.6619
Iteration 1:   log likelihood = -674.54907
Iteration 2:   log likelihood =   -673.407
Iteration 3:   log likelihood = -673.40242
Iteration 4:   log likelihood = -673.40242
Refining estimates:
Iteration 0:   log likelihood = -673.40242

Cox regression -- Breslow method for ties

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   64.52
Log likelihood   =  -673.40242          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
     _t
     _d      Coef.  Std. Err.      z   p>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
prison   .3265108   .1672211   1.95  0.051  -.0012366   .6542581
clinic  -1.00887   .2148709  -4.70  0.000  -1.430009  -.5877304
  dose  -.0353962   .0063795  -5.55  0.000  -.0478997  -.0228926
------------------------------------------------------------------------------
```

この出力から，5回の反復により対数尤度が-673.40242で収束したことがわかります．この反復過程はStataモデル出力の冒頭に表示されますが，これ以降の出力例では削除します．最後の表には，各共変量に対する回帰係数，その標準誤差，Wald検定統計量(z)，p値，95%信頼区間が示されています．

stcoxコマンドの**nohr**オプションは，デフォルトの指数化係数(ハザード比)ではなく回帰係数を要求します．指数化係数を求めたい場合は**nohr**オプションを削除します．コードと出力は以下の通りです．

stcox prison clinic dose

```
Cox regression -- Breslow method for ties

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   64.52
Log likelihood   =  -673.40242          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
     _t
     _d  Haz. Ratio  Std. Err.      z   p>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
prison    1.386123    .231789   1.95  0.051    .9987642  1.923715
clinic    .3646309   .0783486  -4.70  0.000    .2393068  .5555868
  dose     .965223   .0061576  -5.55  0.000    .9532294  .9773675
```

この表には，ハザード比とその標準誤差，ハザード比の信頼区間が示されています．**stcox**コマンドを使用する場合は，time-to-event変数やstatus変数を指定する必要がないことに注意してください．**stcox**コマンドは，**stset**コマンドで指定した情報を利用します．以前に実行した**stset**コマンドに依存しないで，**cox**コマンドからCoxモデルを実行することもできます．コードは以下の通りです．

cox survt prison clinic dose, dead（status）

このcoxコマンドには，変数SURVTの指定があります．dead（）オプションは，イベントと打ち切りを区別する変数STATUSを指定するために使用します．dead（）オプションで指定する変数は，イベントの場合は0以外，打ち切りの場合は0をとる必要があります．coxコマンドの出力は以下の通りです．

```
Cox regression -- Breslow method for ties
Entry time 0                            Number of obs  =     238
                                        LR chi2(3)     =   64.52
                                        Prob > chi2    =  0.0000
Log likelihood = -673.40242             Pseudo R2      =  0.0457

------------------------------------------------------------------------------
 survt
status      Coef.  Std. Err.      z   p>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
prison   .3265108   .1672211   1.95  0.051  -.0012366   .6542581
clinic  -1.00887   .2148709  -4.70  0.000  -1.430009  -.5877304
  dose  -.0353962   .0063795  -5.55  0.000  -.0478997  -.0228926
------------------------------------------------------------------------------
```

この出力結果は，デフォルトで回帰係数が与えられる点を除けば，**stcox**コマンドの出力と同じです．

coxコマンドの**hr**オプションが指数化係数を提供します．

デフォルトでは（この出力では），同順位の処理（複数のイベントが同時に起こった場合）はBreslow法です．よりexactな方法を使用したい場合は，**stcox**コマンドまたは**cox**コマンドの，**exactp**オプション（exactな部分尤度に対応）または**exactm**オプション（exactな周辺尤度に対応）を使用できます．exact methodは集約的な計算であり，大抵の場合exactを指定してもパラメータ推定値はあまり変わりません．しかしながら，同時に多くのイベントが起こる場合は，exact methodが適切です．コードと出力は以下の通りです．

stcox prison clinic dose, nohr exactm

```
Cox regression -- exact marginal likelihood

No. of subjects  =        238           Number of obs  =     238
No. of failures  =        150
Time at risk     =      95812
                                        LR chi2(3)     =   64.56
Log likelihood   =  -666.3274           Prob > chi2    =  0.0000

------------------------------------------------------------------------------
     _t
     _d      Coef.  Std. Err.      z    p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
prison    .326581   .1672306    1.95  0.051  -.0011849   .6543469
clinic  -1.009906   .2148906   -4.70  0.000  -1.431084  -.5887285
  dose  -.0353694   .0063789   -5.54  0.000  -.0478718  -.0228669
```

同順位の処理にEfron法を使用することもできます．R統計パッケージのデフォルトはEfron法です．コードは以下の通りです（出力は省略）．

stcox prison clinic dose, nohr efron

CLINICとの交互作用項である2変数を含むCoxモデルを考えます．**generate**コマンドは新しい変数を定義できます．変数CLIN_PRとCLIN_DOの定義は，それぞれ積項CLINIC × PRISONとCLINIC × DOSEです．コードは以下の通りです．

```
generate clin_pr=clinic*prison
generate clin_do=clinic*dose
```

describeまたは**list**と入力して，これらの新しい変数が作業用データセットに存在することを確認してください．

以下のコードは，2つの交互作用項を含むCoxモデルを実行します．

stcox prison clinic dose clin_pr clin_do, nohr

```
Cox regression -- Breslow method for ties

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(5)     =   68.12
Log likelihood   =  -671.59969          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
     _t
     _d      Coef.  Std. Err.      z   P>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
 prison   1.191998   .5413685   2.20  0.028   .1309348   2.253061
 clinic   .1746985    .893116   0.20  0.845  -1.575777   1.925174
   dose  -.0193175     .01935  -1.00  0.318  -.0572428   .0186079
clin_pr  -.7379931   .4314868  -1.71  0.087  -1.583692   .1077055
clin_do  -.0138608   .0143275  -0.97  0.333  -.0419422   .0142206
```

lrtestコマンドは尤度比検定を実行します．例えば，2つの交互作用項CLIN_PRとCLIN_DOについて主効果モデルとの尤度比検定を実行する場合は，以下のコマンドを入力することにより，フルモデルの－2対数尤度統計量をコンピュータのメモリに保存できます．

lrtest, saving(0)

次に，以下の入力で縮小モデル（交互作用項を含まない）を実行します（出力は省略）．

stcox prison clinic dose

縮小モデルの実行後，以下のコマンドにより，フルモデル（交互作用項あり）と縮小モデルを比較する尤度比検定の結果が得られます．

Lrtest

出力結果は以下の通りです.

```
Cox: likelihood-ratio test          chi2(2)     =     3.61
                                    Prob > chi2 =   0.1648
```

*p*値0.1648は，α = 0.05水準では有意ではありません.

4. 層化Coxモデルの実行

比例ハザード仮定を変数CLINICは満たさないが，変数PRISONおよびDOSEは満たす場合, 層化Cox解析を適用できます．**stcox**コマンドで層化Coxモデルを実行できます．以下のコードは，CLINICについて層化したCoxモデルを実行します(出力も表示).

stcox prison dose, strata(clinic)

```
Stratified Cox regr. -- Breslow method for ties

No. of subjects  =       238            Number of obs  =     238
No. of failures  =       150
Time at risk     =     95812
                                        LR chi2(2)     =   33.94
Log likelihood   =  -597.714            Prob > chi2    =  0.0000

------------------------------------------------------------------------------
     _t
     _d  Haz. Ratio  Std. Err.      z   P>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
prison   1.475192    .2491827   2.30  0.021    1.059418  2.054138
  dose    .9654655   .0062418  -5.44  0.000     .953309   .977777
------------------------------------------------------------------------------
                                                Stratified by clinic
```

strata()オプションでは，最大5つの層化変数を使用できます.

2つの交互作用項を含む層化Coxモデルも実行可能です．前のセクションで**generate**コマンドを用いてこれらの変数を作成したことを思い出してください．このモデルでは, CLINIC値が異なればPRISONおよびDOSEの効果が異なることが許容されます．コードと出力は以下の通りです.

stcox prison dose clin_pr clin_do, strata(clinic) nohr

```
Stratified Cox regr. -- Breslow method for ties

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(4)     =   35.81
Log likelihood   =  -596.77891          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
     _t
     _d      Coef.  Std. Err.      z    P>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
 prison   1.087282   .5386163    2.02  0.044   .0316135   2.142951
   dose  -.0348039   .0197969   -1.76  0.079  -.0736051   .0039973
clin_pr   -.584771   .4281291   -1.37  0.172  -1.423889   .2543465
clin_do  -.0010622    .014569   -0.07  0.942  -.0296169   .0274925
-------------------------------------------------------------------------------
                                                Stratified by clinic
```

CLINIC = 2に関してPRISON = 1 vs. PRISON = 0に対応するハザード比を推定したいとします．このハザード比を推定するには，(prisonの係数 + 2 × clinicとprisonの交互作用項係数)を指数化します．この式は，ハザード比の式に分子(PRISON = 1)と分母(PRISON = 0)の値を代入することにより得られます(下記参照)．

$$HR = \frac{h_0(t)\exp[1\beta_1 + \beta_2 DOSE + (2)(1)\beta_3 + \beta_4 CLIN\_DO]}{h_0(t)\exp[0\beta_1 + \beta_2 DOSE + (2)(0)\beta_3 + \beta_4 CLIN\_DO]}$$
$$= \exp(\beta_1 + 2\beta_3).$$

lincomコマンドは，パラメータの線形結合の指数化値を求めることができます．モデル実行後，CLINIC = 2に関するPRISONのハザード比を推定するために，このコマンドを直接実行します．コードと出力は以下の通りです．

lincom prison+2*clin_pr, hr

```
 ( 1)  prison + 2.0 clin_pr = 0.0
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .9210324   .3539571    -0.21   0.831     .4336648    1.956121
------------------------------------------------------------------------------
```

ifステートメントを使用すれば，特定の抽出データを対象にモデルを実行することができます．以下のコード（出力も）は，CLINIC = 2のデータについてCoxモデルを実行します．

stcox prison dose if clinic==2

```
Cox regression -- Breslow method for ties

No. of subjects  =          75          Number of obs  =       75
No. of failures  =          28
Time at risk     =       36254
                                         LR chi2(2)     =     9.70
Log likelihood   =  -104.37135          Prob > chi2    =   0.0078
------------------------------------------------------------------------------
     _t
     _d  Haz. Ratio  Std. Err.      z   p>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
prison    .9210324   .3539571  -0.21  0.831  .4336648    1.956121
  dose    .9637452   .0118962  -2.99  0.003  .9407088    .9873457
------------------------------------------------------------------------------
```

CLINIC = 2におけるPRISON = 1 vs. PRISON = 0のハザード比推定値は，交互作用項を含む層化Coxのアプローチと抽出データのアプローチとでまったく同じ値となります（0.9210324）．

5. 統計的検定による比例ハザード仮定の評価

stphtestコマンドは比例ハザード仮定に関する統計的検定を実行します．比例ハザード仮定の評価に関しては，統計的検定はグラフを用いた方法よりも客観的な基準を与えます．しかしそれは，統計的検定がグラフを用いた方法よりも良いということではありません．単に客観性が高いというだけです．実際，グラフを用いた方法の方が一般的に，比例ハザード性からの乖離に関する具体的な特徴の情報を多く得られます．

stphtestコマンドは，すべての共変量を同時に評価する比例ハザード仮定の包括的検定を出力しますが，詳細なオプションを指定すれば共変量別に検定することもできます．これらの検定を実行するためには，包括的検定用のSchoenfeld残差とそれぞれの共変量を個別に検定するためのscaled Schoenfeld残差を求める必要があります．比例ハザード検定は，「もし比例ハザード仮定が成り立つなら，残差と生存時間（または生存時間順位）との間には相関関係は存在しない」という考えに基づいています．逆に，早期にイベントが起きた対象の残差が正となり，後期にイベントが起きた対象の残差が負になる（逆も成り立つ）傾向がある場合は，ハザード比が時間の経過とともに変化する（つまり比例ハザード仮定が成り立たない）ことを示唆することになります．

stphtestを実行する前に，**stcox**コマンドを実行してSchoenfeld残差（**schoenfeld**()オプションを使用）とscaled Schoenfeld残差（**scaledsch**()オプションを使用）を求めておく必要があります. 新たに定義する変数名を()内に指定します．()内に**schoen***と指定すると，SCHOEN1，SCHOEN2，SCHOEN3の変数が作成され，**scaled***の指定はSCALED1，SCALED2，SCALED3の変数が作成されます．これらの変数はそれぞれ，PRISON，DOSE, CLINICの残差に対応します（モデルに指定した変数の順）．この変数名は固定ではなく，ユーザーが任意に指定できます．Schoenfeld残差は包括的検定に用いられ, scaled Schoenfeld残差は, 個々の変数の比例ハザード仮定の検定に用いられます.

stcox prison dose clinic, schoenfeld（schoen*） scaledsch（scaled*）

残差を定義した後は**stphtest**コマンドを実行できます．コードと出力は以下の通りです.

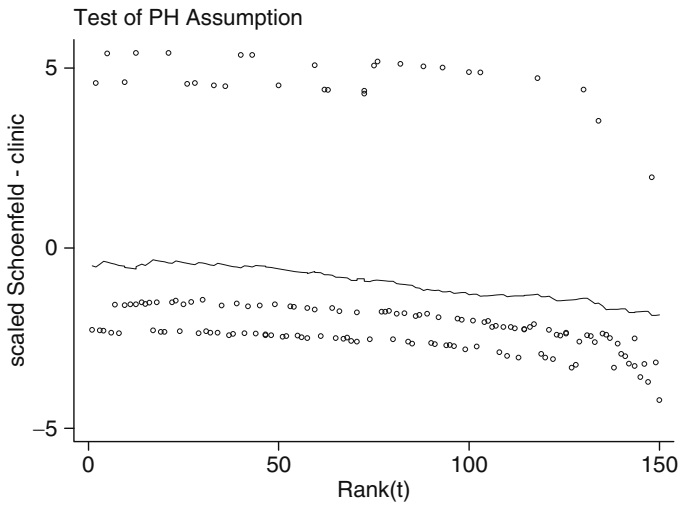
stphtest, rank detail

```
Test of proportional hazards assumption
Time: Rank(t)
------------------------------------------------------------
                    rho   chi2  df  Prob>chi2
------------------------------------------------------------
prison        -0.04645   0.32   1     0.5689
dose           0.08975   1.08   1     0.2996
clinic        -0.24927  10.44   1     0.0012
------------------------------------------------------------
global test             12.36   3     0.0062
------------------------------------------------------------
```

検定結果からは，CLINIC に関しては，p値が0.0012なので，比例ハザード仮定が成立しないことを示唆します．PRISONとDOSEに関しては，比例ハザード仮定の不成立は示唆しません.

stphtestコマンドの**plot**()オプションで, CLINICのscaled Schoenfeld残差と生存時間順位のプロットを作成できます．比例ハザード仮定が成立するならば，scaled Schoenfeld残差は生存時間と独立となるので，プロットの中心曲線は水平になるはずです．コードとグラフは以下の通りです.

stphtest, rank plot(clinic)



この中心曲線は少し右肩下がりになっています(水平ではなく).

6. Cox調整生存曲線の作成

調整生存曲線を得るには**sts graph**コマンドを用います.調整生存曲線は共変量パターンに依存します.例えば,PRISON = 1,CLINIC = 1,DOSE = 40である被験者の調整生存推定値は,PRISON = 0,CLINIC = 2,DOSE = 70である被験者とは異なります.**sts graph**コマンドは調整基準生存曲線を作成します.以下のコードは,PRISON = 0,CLINIC = 0,DOSE = 0の調整生存時間プロットを生成します(出力省略).

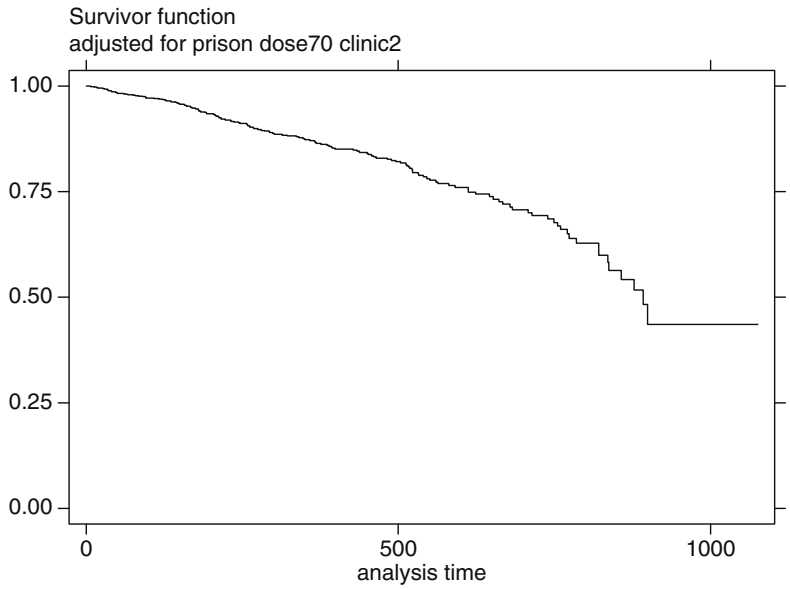
sts graph, adjustfor(prison dose clinic)

基準曲線よりも,意味のある共変量パターン(CLINIC = 0は存在しない不適切な値)での調整プロットに興味があると思います.PRISON = 0, CLINIC = 2, DOSE = 70の調整生存曲線のグラフを求めることにします.**generate**コマンドで,**sts graph**コマンドに用いる新しい変数を作成します.

generate clinic2=clinic-2
generate dose70=dose-70

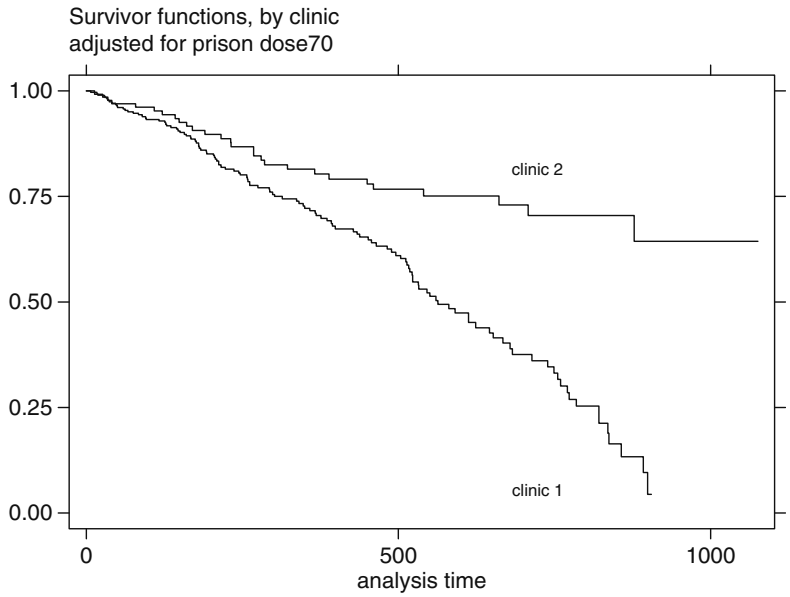
これらの変数(PRISON,CLINIC2,DOSE70)がそれぞれ0の値を取るとき,求める共変量パターンに対応します.以下のコードは求める結果を与えます.

sts graph, adjustfor(prison dose70 clinic2)



調整層化Cox生存曲線は**strata()**オプションを用います．以下のコードは，CLINICで層別した2つの生存曲線，すなわち(CLINIC = 1，PRISON = 0，DOSE = 70)と(CLINIC = 2，PRISON = 0，DOSE = 70)を作成します．

sts graph, strata(clinic) adjustfor(prison dose70)



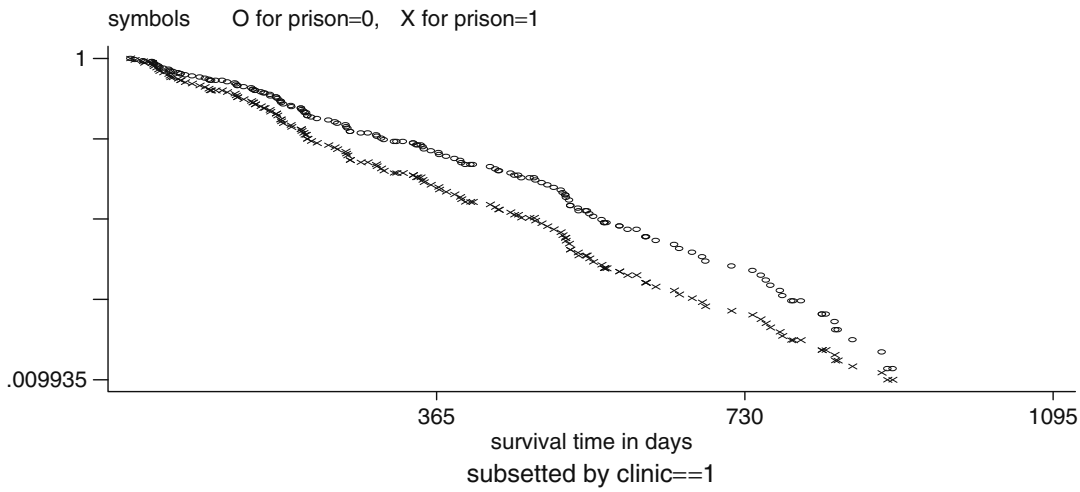
これらの調整曲線は，CLINICの生存に対する効果が大きいことを示唆しています．

CLINICで層別したときのPRISON = 1とPRISON = 0の調整生存プロットを比較したいとします. この設定では**sts graph**コマンドを直接使うことはできません．なぜならば，PRISONの両方のレベル（PRISON = 1とPRISON = 0）を同時に基準レベルとして定義することができないからです（**sts graph**は基準生存関数のみをプロットします）．しかし，**sts generate**コマンドを2回実行することで2つの生存推定値が得られます．つまりPRISON = 0を基準レベルに定義して1回実行し，PRISON = 1を基準レベルに定義してもう1回実行します．以下のコードは，求める調整生存推定値の変数を作成します.

generate prison1=prison-1
sts generate scox0=s, strata（clinic） adjustfor（prison dose70）
sts generate scox1=s, strata（clinic） adjustfor（prison1 dose70）

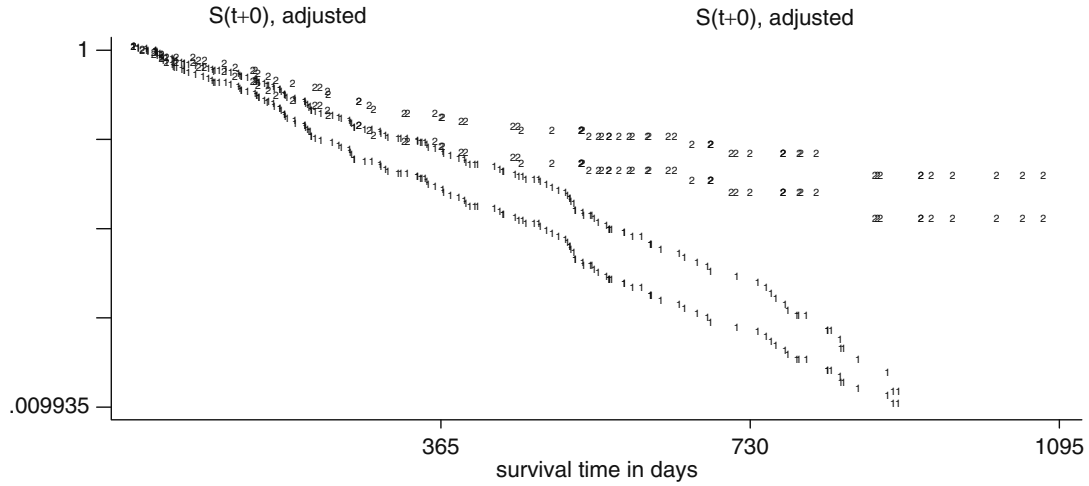
変数SCOX1にはPRISON = 1の，変数SCOX0にはPRISON = 0の，DOSEで調整しCLINICで層別した生存推定値です．**graph**コマンドは，これらの推定値をプロットします．Stataのバージョンが8.0以降（例えばStata 8.0など）の場合は，**graph**コマンドではなく**graph7**コマンドを使用します．コードと出力は以下の通りです.

Graph7 scox0 scox1 survt, twoway symbol（[clinic] [clinic]） xlabel（365,730,1095）



CLINIC = 1のデータを抽出し，PRISON = 1とPRISON = 0に関するグラフを作成することもできます. **twoway**オプションは2次元散布図を要求します．**symbol**オプションは記号を，**xlabel**オプションはX軸表示値を，**title**オプションはタイトルをそれぞれ要求します.

graph7 scox0 scox1 survt if clinic==1, twoway symbol（ox） xlabel（365,730,1095） t1（“ symbols O for prison=0, X for prison=1”） title（“subsetted by clinic==1”）



7. 拡張Coxモデルの実行

比例ハザード性が成立しない場合，考えられる1つの戦略としては，層化Coxモデルの実行があります．これとは別の戦略として，時間依存性共変量を用いたCoxモデル（拡張Coxモデル）の実行があります．拡張Coxモデルを実行するうえで課題となるのは，モデルに含める生存時間関数を適切に選択することです．

DOSEに時間の対数を掛けた時間依存性変数を考えます．この積項は，DOSE の任意の2水準を比較したハザード比が時間に対して単調増加（または減少）する場合に適している可能性があります．**stcox** コマンドの **tvc**() オプションを使用して，DOSEに時間の関数を掛けた時間依存性変数を作成できます．時間関数の指定は，時間を表す変数 **_t** を用いて **texp** オプションで記述します．時間依存性共変量DOSE × ln（_t）を含むモデルのコードおよび出力は以下の通りです．

stcox prison clinic dose, tvc(dose) texp(ln(_t)) nohr

```
Cox regression -- Breslow method for ties

No. of subjects  =          238          Number of obs  =     238
No. of failures  =          150
Time at risk     =        95812
                                          LR chi2(4)     =   66.29
Log likelihood   =  -672.51694           Prob > chi2    =  0.0000
------------------------------------------------------------------------------
        _t
        _d      Coef.  Std. Err.      z   p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
rh
    prison   .3404817   .1674672   2.03  0.042    .012252   .6687113
    clinic  -1.018682    .215385  -4.73  0.000  -1.440829  -.5965352
      dose  -.0824307   .0359866  -2.29  0.022  -.1529631  -.0118982
-------------------------------------------------------------------------------
t
      dose   .0085751   .0064554   1.33  0.184  -.0040772   .0212274
-------------------------------------------------------------------------------
note: second equation contains variables that continuously vary
      with respect to time; variables interact with current
      values of ln(_t).
```

時間依存性共変量DOSE × ln(_t)のパラメータ推定値は0.0085751です．しかし，Wald検定のp値は0.184で，統計的に有意ではありません．

Heavisideの階段関数を使用することもできます．以下のコードは，時間依存性変数の値は時間が365日以上である場合はCLINICに等しく，それ以外は0となるモデルを実行します．

stcox prison dose clinic, tvc(clinic) texp(_t>=365) nohr

Stataは式(_t> = 365)を，生存時間が365日以上の場合は値1をとり，それ以外は0をとると認識します．出力は以下の通りです．

```
Cox regression -- Breslow method for ties

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(4)     =   74.17
Log likelihood   =  -668.57443          Prob > chi2    =  0.0000
------------------------------------------------------------------------------
       _t
       _d      Coef.  Std. Err.      z   p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
rh
   prison    .377704   .1684024   2.24  0.025   .0476414    .7077666
     dose  -.0355116   .0064354  -5.52  0.000  -.0481247   -.0228985
   clinic  -.4595628   .2552911  -1.80  0.072   -.959924    .0407985
-------------------------------------------------------------------------------
t
   clinic  -1.368665   .4613948  -2.97  0.003  -2.272982    -.464348
-------------------------------------------------------------------------------
note: second equation contains variables that continuously vary
      with respect to time; variables interact with current
      values of _t>=365.
```

残念ながら，**texp** オプションは **stcox** コマンド中では1個しか指定できません．そのため，Heavisideの階段関数を2つ必要とするようなモデルの実行はこの方法では難しくなります．そこで，このような場合は，作業用データセットにオブザベーションを追加する **stsplit** コマンドを使用します．以下のコードはV1と呼ばれる変数を作成し，新しいオブザベーションをデータセットに追加します．

stsplit v1, at(365)

上記のstsplitコマンドを実行すると，365日を超える被験者は，1つではなく2つのオブザベーションになります．例えば，1番目の被験者（ID = 1）は428日目にイベントがあります．1番目のオブザベーションは0〜365日の間にイベントがないことを示し，2番目のオブザベーションは428日にイベントがあることを示します．新たに定義された変数 **v1** は，生存時間が365日以上であるオブザベーションに対しては値365をとりますが，生存時間が365日未満であるオブザベーションに対しては値0をとります．以下のコードは，要求された変数について最初の10個のオブザベーションを表示します（出力は以下の通り）．

list id _t0 _t _d clinic v1 in 1/10

```
      id    _t0     _t    _d   clinic     v1
 1.    1      0    365     0        1      0
 2.    1    365    428     1        1    365
 3.    2      0    275     1        1      0
 4.    3      0    262     1        1      0
 5.    4      0    183     1        1      0
 6.    5      0    259     1        1      0
 7.    6      0    365     0        1      0
 8.    6    365    714     1        1    365
 9.    7      0    365     0        1      0
10.    7    365    438     1        1    365
```

データがこの形式をとる場合，以下のコードを使用して2つのHeavisideの階段関数を定義することができます．

```
generate hv2=clinic*(v1/365)
generate hv1=clinic*(1-(v1/365))
```

以下のコードと出力は，オブザベーション（**159/167行**）を表示します．オブザベーション番号の出力は抑制されています（**noobs**オプション）．

list id _t0 _t clinic v1 hv1 hv2 in 159/167, noobs

```
 id    _t0    _t   clinic    v1   hv1   hv2
100     0   365        1     0     1     0
100   365   749        1   365     0     1
101     0   150        1     0     1     0
102     0   365        1     0     1     0
102   365   465        1   365     0     1
103     0   365        2     0     2     0
103   365   708        2   365     0     2
104     0   365        2     0     2     0
104   365   713        2   365     0     2
```

2つの階段関数を定義した分割データを用いて，これら関数を含む時間依存性モデルを以下のコードで実行できます（出力は以下の通り）．

stcox prison dose hv1 hv2, nohr

```
No. of subjects  =         238          Number of obs  =     360
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(4)     =   74.17
Log likelihood   =  -668.57443          Prob > chi2    =  0.0000
------------------------------------------------------------------------------
     _t
     _d      Coef.  Std. Err.      z    p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
prison    .377704   .1684024    2.24  0.025   .0476414   .7077666
  dose  -.0355116   .0064354   -5.52  0.000  -.0481247  -.0228985
   hv1  -.4595628   .2552911   -1.80  0.072   -.959924   .0407985
   hv2  -1.828228    .385946   -4.74  0.000  -2.584668  -1.071788
```

stsplitコマンドは複雑ですが，時間依存性解析に対応するデータを編集する強力なアプローチです．

データを前の形式に戻すには，分割時に新たに作成した変数を削除した後に**stjoin**コマンドを使用します．

```
drop v1 hv1 hv2
stjoin
```

ありとあらゆるfailure時間でデータを分割することは可能ですが，大量のメモリを必要とします．ゆえに，モデルの時間依存性共変量が1つの場合は，**stcox**コマンドの**tvc**および**texp**オプションを用いるのが，拡張Coxモデルを実行する最も単純な方法です．

生存時間変数（結果変数）と，時間依存性変数の定義に使用される変数（Stataでは**_t**）を混同しないように注意してください．生存時間変数は時間独立な変数です．個々のイベント（または打ち切り）の時間は変化しません．一方，時間依存性変数の定義は，値が時間とともに変化することです．

8. パラメトリックモデルの実行

Cox比例ハザードモデルは，生存時間解析において最も広く用いられるモデルです．普及している主な理由は，生存時間変数の分布を指定する必要がないことです．しかし，生存時間が特定の分布に従うと考えられる場合は，その情報を生存データのパラメトリックモデルの構築に利用できます．

パラメトリックモデルの多くは加速(AFT)モデルです．比例ハザードモデルの主要な仮定がハザード比は時間を通して一定であるのに対して，加速モデルの主要な仮定は，共変量のレベル間で生存時間が一定の割合で加速(または減速)するというものです．

生存データのパラメトリックモデルの分布の中で最も一般的なものはWeibull分布です．Weibull分布には，加速時間仮定が成り立てば比例ハザード仮定も成り立つという好ましい性質があります．指数分布はWeibull分布の特殊な例です．指数分布の主な性質は，ハザードは時間に対して一定(ハザード比だけでなく)というものです．Weibullおよび指数モデルは，比例ハザードモデル(デフォルト)または加速モデルとして実行できます．

Weibull仮定の妥当性をグラフを用いて確認する方法として，対数生存時間を横軸としたKaplan-Meier対数(－対数)生存曲線を吟味するというものがあります．これは**sts graph**コマンドを使用します(Appendixのセクション2参照)．プロットが直線であれば，生存時間の分布がWeibull分布に従うことを示します．直線の傾きが1ならば，生存時間が指数分布に従うことを示唆します．

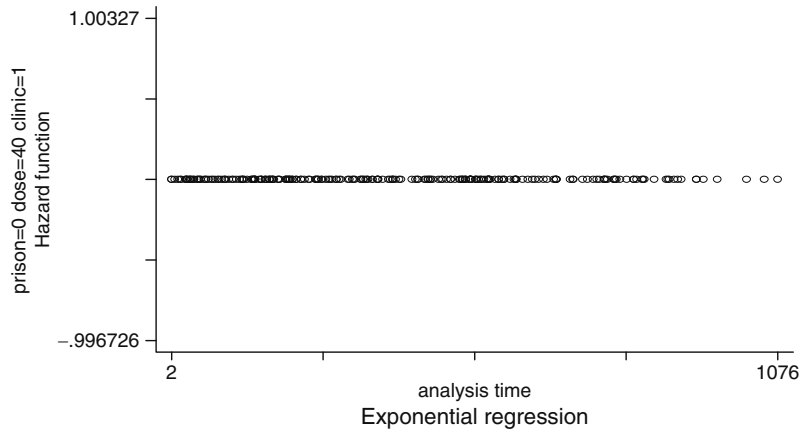
パラメトリックモデルの実行には**streg**コマンドを使用します．薬物常用者データセットから得た対数(－対数)生存曲線は直線ではありませんが，説明のためにこのデータを使用します．まず，指数分布を用いたパラメトリックモデルを示します．コードと出力は以下の通りです．

streg prison dose clinic, dist(exponential) nohr

```
Exponential regression -- log relative-hazard form

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   49.91
Log likelihood   =  -270.47929          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------------------------------------------------------------------
prison    .2526491   .1648862     1.53   0.125    -.070522    .5758201
  dose   -.0289167   .0061445    -4.71   0.000   -.0409596   -.0168738
clinic   -.8805819    .210626    -4.18   0.000   -1.293401   -.4677625
 _cons   -3.684341   .4307163    -8.55   0.000   -4.528529   -2.840152
------------------------------------------------------------------------------
```

分布は**dist()**オプションで指定します．**streg**コマンドの後に**stcurv**コマンドを用いれば，生存，ハザード，累積ハザードの適応曲線を得ることができます．以下のコードは，PRISON = 0，DOSE = 40，CLINIC = 1における推定ハザード関数を求めるものです．

stcurv, hazard at（prison=0 dose=40 clinic=1）



このグラフは，生存時間が指数分布に従う場合はハザードが時間に対して一定となることを示しています．

次に，**streg**コマンドを用いてWeibull分布を実行します．

streg prison dose clinic, dist（weibull）nohr

```
Weibull regression -- log relative-hazard form

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   60.89
Log likelihood   =  -260.98467          Prob > chi2    =  0.0000
------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------------------------------------------------------------------
prison    .3144143   .1659462     1.89   0.058    -.0108342    .6396628
  dose   -.0334675    .006255    -5.35   0.000    -.0457272   -.0212079
clinic   -.9715245   .2122826    -4.58   0.000    -1.387591   -.5554582
 _cons   -5.624436   .6588041    -8.54   0.000    -6.915668   -4.333203
-------------------------------------------------------------------------------
 /ln_p    .3149526   .0675583     4.66   0.000     .1825408    .4473644
-------------------------------------------------------------------------------
     p    1.370194    .092568                      1.200263    1.564184
   1/p    .7298235   .0493056                      .6393109    .8331507
------------------------------------------------------------------------------
```

Weibull分布の出力には，指数分布にはないパラメータpがあることに注意してください．Weibull分布のハザード関数はλpt^{p-1}です．p = 1ならば，Weibull分布は指数分布になります（$h(t) = \lambda$）．Weibull分布に関しては，デフォルトではハザード比パラメータが与えられます．加速モデルでパラメータ化したい場合は，**time**オプションを使用します．

Weibull加速モデルのコードおよび出力は以下の通りです．

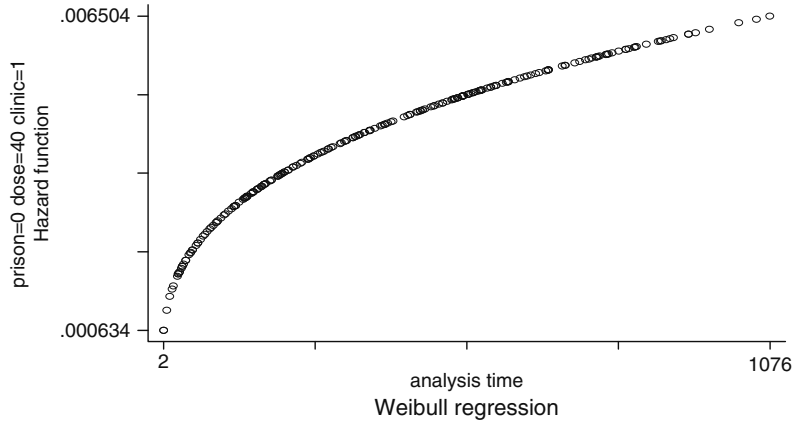
streg prison dose clinic, dist（weibull）time

```
Weibull regression -- accelerated failure-time form

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   60.89
Log likelihood   =  -260.98467          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------------------------------------------------------------------
prison  -.2294669   .1207889    -1.90   0.057    -.4662088    .0072749
  dose   .0244254   .0045898     5.32   0.000     .0154295    .0334213
clinic   .7090414   .1572246     4.51   0.000     .4008867    1.017196
 _cons   4.104845   .3280583    12.51   0.000     3.461863    4.747828
-------------------------------------------------------------------------------
 /ln_p   .3149526   .0675583     4.66   0.000     .1825408    .4473644
-------------------------------------------------------------------------------
       p   1.370194    .092568                   1.200263    1.564184
     1/p   .7298235   .0493056                   .6393109    .8331507
------------------------------------------------------------------------------
```

ハザード比パラメータβ_jと加速モデルパラメータα_jとの関係は，$\beta_j = -\alpha_j p$です．例えば，Weibull比例ハザードモデルと加速モデルのPRISONの係数推定値は，0.3144 =（− 0.2295）（1.37）という関係になります．

stcurvを**streg**コマンドの後に再び用いて，生存，ハザード，累積ハザードの適応曲線を得ることができます．以下のコードにより，PRISON = 0，DOSE = 40，CLINIC = 1における推定ハザード関数を求めます．

stcurv, hazard at（prison=0 dose=40 clinic=1）



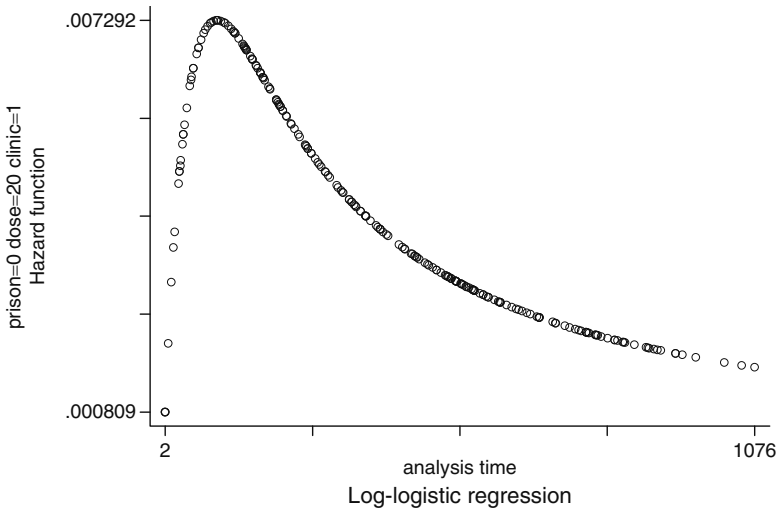
ハザードのプロットは単調増加しています．Weibull分布では，ハザードが増加した後に減少に転じるようなことはできません．次の例に示すように，対数ロジスティック分布はそうではありません．対数ロジスティックモデルは比例ハザードモデルではありません．そのため，**streg**コマンドのデフォルトモデルは加速モデルです．コードと出力は以下の通りです．

streg prison dose clinic, dist(loglogistic)

```
Log-logistic regression -- accelerated failure-time form

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   52.18
Log likelihood   =  -270.42329          Prob > chi2    =  0.0000

------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z    P>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
      prison  -.2912719   .1439646    -2.02  0.043  -.5734373   -.0091065
        dose   .0316133   .0055192     5.73  0.000   .0207959    .0424307
      clinic   .5805977   .1715695     3.38  0.001   .2443276    .9168677
       _cons   3.563268   .3894467     9.15  0.000   2.799967     4.32657
-------------------------------------------------------------------------------
     /ln_gam  -.5331424   .0686297    -7.77  0.000  -.6676541   -.3986306
-------------------------------------------------------------------------------
       gamma   .5867583    .040269                   .5129104    .6712386
```

Stataでは対数ロジスティックモデルの形状パラメータγが出力されます．PRISON = 0，DOSE = 40，CLINIC = 1におけるハザード関数のグラフを作成するためのコードは以下の通りです．

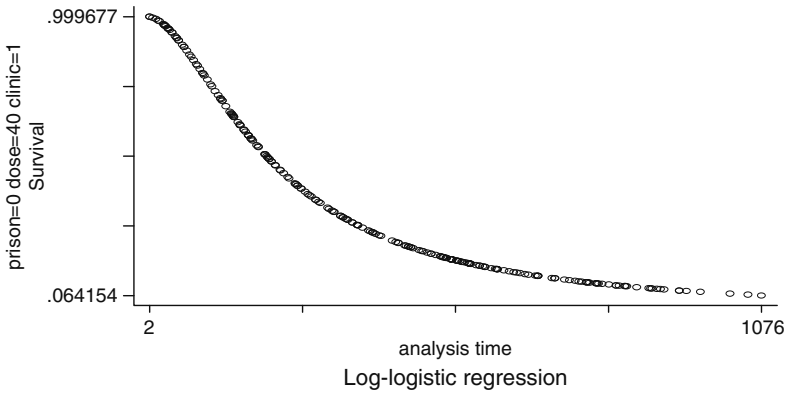
stcurv, hazard at（prison=0 dose=40 clinic=1）



ハザード関数は，（Weibullハザード関数とは対照的に）最初に増加してから減少します.

対応する対数ロジスティック分布の生存曲線も，**stcurve**コマンドを使用して求めることができます.

stcurvsurvival at（prison=0 dose=40 clinic=1）



対数ロジスティックモデルで加速時間仮定が成り立つならば，生存関数について比例オッズ仮定が成り立ちます（比例ハザード仮定は成り立ちません）. 比例オッズ仮定は，対数生存オッズ（Kaplan-Meier推定値を用いる）と，生存時間の対数をプロットすることにより評価できます．それぞれの共変量パターンに関してこのプロットが直線になれば，対数ロジスティック分布が当てはまることになります. 直線でさらに平行であれば，比例オッズと加速時間仮定も成り立ちます．以下のコードは，CLINIC別の，時間の対数に対する対数生存オッズの推定値をプロットします（出力は省略）.

```
sts generate skm=s, by(clinic)
generate logodds=ln(skm/(1-skm))
generate logt=ln(survt)
graph7 logodds logt, twoway symbol([clinic] [clinic])
```

比例オッズ仮定を別の面から考えてみます．それは，ロジスティック回帰によるオッズ比推定値がフォローアップ期間に依存するかということです．例えば，試験のフォローアップが3年から5年に延長したとしても，比例オッズ仮定のもとでは，2つの共変量パターンを比較するオッズ比は変わらないはずです．比例オッズ仮定が真でなければ，オッズ比はフォローアップの長さによって変わってきます．

対数ロジスティックもWeibullモデルも，通常は定数と仮定する追加の形状パラメータを含みます．定数の仮定は，これらモデルの比例ハザード仮定や加速時間仮定が成り立つために必要です．Stataでは，**streg**コマンドに**ancillary**オプションを使用することで，この形状パラメータを予測変数の関数としてモデル化できます（第7章の「その他のパラメトリックモデル」を参照）．以下のコードは，形状パラメータγをCLINICの関数とし，λをPRISONとDOSEの関数とした対数ロジスティックモデルを実行します．

```
streg prison dose, dist(loglogistic) ancillary(clinic)
```

出力は以下の通りです．

```
Log-logistic regression -- accelerated failure-time form

No. of subjects  =          238                 Number of obs   =       238
No. of failures  =          150
Time at risk     =        95812
                                                 LR chi2(2)      =     38.87
Log likelihood   =   -272.65273                 Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z    P>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
_t
      prison  -.3275695   .1405119    -2.33  0.020  -.6029677  -.0521713
        dose   .0328517   .0054275     6.05  0.000    .022214   .0434893
       _cons   4.183173   .3311064    12.63  0.000   3.534216    4.83213
-------------------------------------------------------------------------------
ln_gam
      clinic   .4558089   .1734819     2.63  0.009   .1157906   .7958273
       _cons  -1.094496   .2212143    -4.95  0.000  -1.528068  -.6609238
------------------------------------------------------------------------------
```

「ln_gam」(γの対数)欄に，CLINICのパラメータ推定値と，切片(_cons)の推定値があります．このモデルでは，γの推定値はCLINIC = 1，CLINIC = 2の値に基づいています．このタイプのモデルは，予測変数の解釈がとても難しく，それが汎用的に用いられない理由です．しかしながら，PRISON，DOSE，CLINICのすべての値の組み合わせを，対数ロジスティックハザード関数および生存関数のパラメータ推定式に代入すれば，ハザード関数と生存関数を推定できます．

stregが使用できる分布としてはその他に，一般化ガンマ分布，対数正規分布，ゴンペルツ分布があります．

9. frailtyモデルの実行

frailtyモデルには，個人レベルでのハザードの違いを説明するための追加のランダム成分(それ以外ではこのモデルでその違いを説明することができない)が含まれています．frailty αは，何らかの分布に従うと仮定するハザードへの相乗的な効果です．frailtyで条件付けられたハザード関数は，$h(t|\alpha) = \alpha[h(t)]$と表すことができます．

Stataでは，frailtyの分布が2つ用意されており，平均1，分散θのガンマ分布と逆ガウス分布です．分散(θ)はモデルで推定するパラメータです．θ = 0ならばfrailtyは存在しません．

1番目の例では，PRISON，DOSE，CLINICを予測因子としたWeibull比例ハザードモデルを実行しています．frailty成分にガンマ分布を仮定しています．このセクションで紹介したモデルの実行にはStata 8.0を使用しました．コードは以下の通りです．

```
streg dose prison clinic, dist(weibull) frailty(gamma) nohr
```

frailty()オプションはfrailtyモデルの実行を要求します．出力は以下の通りです．

```
Weibull regression -- log relative-hazard form
                      Gamma frailty

No. of subjects  =          238          Number of obs  =     238
No. of failures  =          150
Time at risk     =        95812
                                          LR chi2(4)     =   60.89
Log likelihood   =  -260.98467           Prob > chi2    =  0.0000
------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z    P>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
        dose  -.0334635   .0062553  -5.35  0.000  -.0457237  -.0212034
      prison   .3143786    .165953   1.89  0.058  -.0108833   .6396405
      clinic  -.9714998   .2122909  -4.58  0.000  -1.387582  -.5554173
       _cons  -5.624342   .6588994  -8.54  0.000  -6.915761  -4.332923
-------------------------------------------------------------------------------
       /ln_p   .3149036   .0675772   4.66  0.000   .1824548   .4473525
  /ln_the  -15.37947   722.4246  -0.02  0.983  -1431.306   1400.547
-------------------------------------------------------------------------------
         p   1.370127   .0925893                  1.20016   1.564166
       1/p   .7298592   .0493218                 .6393185   .8332223
     theta   2.09e-07   .0001512                        0          .
------------------------------------------------------------------------------
Likelihood ratio test of theta = 0: chibar2(01) = 0.00
  Prob>=chibar2 = 1.000
```

前のセクションで実行したモデルと比較すると，1つのパラメータ(θ)が追加されています．θの推定値は2.09 × 10 − 7すなわち0.000000209であり，ほぼ0です．θを含めることに対する尤度比検定が出力の最後に示されており，カイ2乗値0.00とp値1.000となっています．このモデルに関してはfrailtyの効果はなく，モデルに含める必要はありません．

次のモデルは，CLINICが含まれない点を除けば前のモデルと同じです．CLINICなどの重要な共変量がモデルに含まれていない方がfrailty成分の果たす役割が大きくなると考えることもできます．コードと出力は以下の通りです．

streg dose prison, dist(weibull) frailty(gamma) nohr

```
Weibull regression -- log relative-hazard form
                      Gamma frailty

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   36.00
Log likelihood   =  -273.42782          Prob > chi2    =  0.0000
------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z   p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
        dose  -.0358231    .010734  -3.34  0.001  -.0568614  -.0147849
      prison   .2234556   .2141028   1.04  0.297  -.1961783   .6430894
       _cons  -6.457393   .6558594  -9.85  0.000  -7.742854  -5.171932
-------------------------------------------------------------------------------
      /ln_p   .2922832   .1217597   2.40  0.016   .0536385   .5309278
    /ln_the  -2.849726   5.880123  -0.48  0.628  -14.37456   8.675104
-------------------------------------------------------------------------------
           p   1.339482    .163095                 1.055103   1.700509
         1/p   .7465571   .0909006                 .5880591   .9477747
       theta   .0578602    .340225                 5.72e-07    5855.31
------------------------------------------------------------------------------
Likelihood ratio test of theta = 0: chibar2(01) = 0.03
  Prob>=chibar2 = 0.432
```

frailtyの分散(θ)の推定値は0.0578602です．この推定値は前の例のようにほぼ0という訳ではありませんが，θの尤度比検定のp値は0.432と有意ではありません．つまり，frailtyを追加しても，CLINICをモデルから除外した分の説明にはなっていません．

次に，frailtyにガンマ分布ではなく逆ガウス分布を使用した以外は前と同じモデルを実行します．コードと出力は以下の通りです．

streg dose prison, dist(weibull) frailty(invgaussian) nohr

```
Weibull regression -- log relative-hazard form
                      Inverse-Gaussian frailty

No. of subjects  =         238          Number of obs  =     238
No. of failures  =         150
Time at risk     =       95812
                                        LR chi2(3)     =   35.99
Log likelihood   =  -273.43201          Prob > chi2    =  0.0000
------------------------------------------------------------------------------
          _t      Coef.  Std. Err.      z   p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
        dose  -.0353908   .0096247  -3.68  0.000  -.0542549  -.0165268
      prison   .2166456   .1988761   1.09  0.276  -.1731445   .6064356
       _cons  -6.448779   .6494397  -9.93  0.000  -7.721658  -5.175901
-------------------------------------------------------------------------------
       /ln_p   .2875567   .1122988   2.56  0.010   .0674551   .5076583
     /ln_the  -3.137696   7.347349  -0.43  0.669  -17.53824   11.26284
-------------------------------------------------------------------------------
           p   1.333166   .1497129                 1.069782   1.661396
         1/p   .7500941   .0842347                 .6019034   .9347697
       theta   .0433827   .3187475                 2.42e-08   77873.78
-------------------------------------------------------------------------------
Likelihood ratio test of theta = 0: chibar2(01) = 0.02
  Prob>=chibar2 = 0.443
```

θの尤度比検定のp値は0.443です（出力の一番下）．この例の結果から，frailty成分に逆ガウスまたはガンマ分布のどちらを仮定しようとほとんど違いはないことがわかります．

再発イベントデータに共有frailtyを適応した例については，次のセクションで紹介します．

10. 再発イベントのモデル構築

再発イベントのモデル構築について，Appendixの冒頭で紹介した「膀胱がん」データセット“**bladder.dta**”を用いて説明します．再発イベントは，複数のイベントを経験した被験者に対しては，複数のオブザベーションを持つデータで示されます．「膀胱がん」データセットのデータレイアウトは，オブザベーションごとに時間区間を定義するCPアプローチに適しています（第8章を参照）．以下のコードは，4被験者の情報からなる12～20番目のオブザベーションを出力します．コードと出力は以下の通りです．

```
     id  event  interval  start  stop  tx  num  size
12.  10      1         1      0    12   0    1     1
13.  10      1         2     12    16   0    1     1
14.  10      0         3     16    18   0    1     1
15.  11      0         1      0    23   0    3     3
16.  12      1         1      0    10   0    1     3
17.  12      1         2     10    15   0    1     3
18.  12      0         3     15    23   0    1     3
19.  13      1         1      0     3   0    1     1
20.  13      1         2      3    16   0    1     1
```

ID = 10は3つのオブザベーション，ID = 11は1つのオブザベーション，ID = 12は3つのオブザベーション，ID = 13は2つのオブザベーションがあります．変数STARTおよびSTOPは，そのオブザベーションで指定するリスクの時間区間を表しています．変数EVENTは，イベントが発生（code = 1）したかどうかを示します．最初の3つのオブザベーションは，ID = 10の被験者に12ヵ月にイベントがあり，16ヵ月に別のイベントがあり，18ヵ月に打ち切りとなったことを示しています．

Stataのsurvivalコマンドを使用する前に，**stset**コマンドを使用してキー生存変数を定義する必要があります．コードは以下の通りです．

stset stop, failure（event==1）id（id） time0（start） exit（time.）

薬物常用者データセットについて**stset**コマンドをすでに紹介していますが，ここではより多くの**stset**コマンドのオプションが必要となります．**id**()オプションはsubject変数（クラスター変数）を定義し，**time0**()オプションは時間区間の始めを示す変数を定義し，**exit（time.）**オプションは，被験者のフォローアップ時間の長さに特に制限を設けないこと（例えば，最初のイベントの後にリスクセットから除外しないなど）を定義します．**stset**コマンドにより，Stataは自動的に変数**_t0**，**_t**，**_d**を作成し，時間区間およびイベントステータスを表す生存変数として認識します．実際には，**time0**()オプションはこの**stset**コマンドから省くことも可能で，id()オプションを使用すれば，Stataはデフォルトで正しいCP形式の開始時間変数**_t0**を作成します（**id**()オプションを使用しなければ，デフォルトでは**_t0**は0となります）．以下のコード（と出力）で，新たに作成した変数とともに12～20番目のオブザベーションを表示します．

list id _t0 _t _d tx in 12/20

```
        id      _t0      _t      _d      tx
12.     10       0      12       1       0
13.     10      12      16       1       0
14.     10      16      18       0       0
15.     11       0      23       0       0
16.     12       0      10       1       0
17.     12      10      15       1       0
18.     12      15      23       0       0
19.     13       0       3       1       0
20.     13       3      16       1       0
```

準備が完了したので，**stcox**コマンドを用いてCPアプローチによる再発イベントCoxモデルを実行できます．予測因子は，治療（TX），最初の腫瘍の数（NUM），最初の腫瘍サイズ（SIZE）です．**robust**オプションは係数推定値のロバスト標準誤差を要求します．指数化係数が必要な場合は**nohr**オプションを外してください．コードと出力は以下の通りです．

stcox tx num size, nohr robust

```
Cox regression -- Breslow method for ties

No. of subjects  =          85        Number of obs  =     190
No. of failures  =         112
Time at risk     =        2711
                                      Wald chi2(3)   =   11.25
Log likelihood   =  -460.07958        Prob > chi2    =  0.0105

               (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
  _t               Robust
  _d      Coef.  Std. Err.      z    p>|z|  [95% Conf. Interval]
-------------------------------------------------------------------------------
  tx  -.4070966   .2432658  -1.67  0.094  -.8838889    .0696956
 num   .1606478   .0572305   2.81  0.005   .0484781    .2728174
size  -.0400877   .0726459  -0.55  0.581   -.182471    .1022957
------------------------------------------------------------------------------
```

これらのパラメータ推定値の解釈については第8章で述べています．

この形式のデータを使用し，INTERVALを層化変数とした層化Coxモデルを実行することもできます．層化変数は，被験者が1番目，2番目，3番目，4番目のイベントに対してat riskであるかを示します．これは第8章で紹介した層化CPアプローチであり，再発イベントの発生順序を区別したい場合に使用します．コードと出力は以下の通りです．

stcox tx num size, nohr robust strata(interval)

```
stratified Cox regr. -- Breslow method for ties

No. of subjects  =          85        Number of obs  =     190
No. of failures  =         112
Time at risk     =        2711
                                      Wald chi2(3)   =    7.11
Log likelihood   =  -319.85912        Prob > chi2    =  0.0685

               (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
  _t               Robust
  _d      Coef.  Std. Err.      z    P>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
  tx  -.3342955   .1982339  -1.69  0.092  -.7228268   .0542359
 num   .1156526   .0502089   2.30  0.021    .017245   .2140603
size  -.0080508   .0604807  -0.13  0.894  -.1265908   .1104892
------------------------------------------------------------------------------
                                                Stratified by interval
```

治療効果が1番目，2番目，3番目，4番目のイベントで違っているかを調べるために，治療変数(TX)と層化変数との交互作用項を作成することもできます(このデータセットでは，被験者には最大4つのイベントがあります)．

別の層化アプローチとして，Gap timeアプローチと呼ばれるものがあります．これは層化CPアプローチを少し変えたものです．その違いは再発イベントの時間区間の定義方法にあります．最初のイベントのat risk時間区間には違いはありません．しかしながら，Gap timeアプローチでは，2番目のイベント以降はat riskの開始時間が0にリセットされます．以下のコードは，Gap time再発イベントモデルの実行に適したデータを作成します．

```
generate stop2 =_t - _t0
stset stop2, failure(event==1)exit(time .)
```

generate コマンドは，オブザベーションごとの時間区間の長さを表す新しい変数STOP2を定義します．**stset** コマンドは，STOP2を結果変数（**_t**）として用います．Stataはデフォルトで，変数**_t0**を0に設定します．以下のコード（と出力）は，選択された変数について12～20番目のオブザベーションを表示します．

```
list id _t0 _t _d tx in 12/20
```

```
        id     _t0     _t     _d     tx
12.     10       0     12      1      0
13.     10       0      4      1      0
14.     10       0      2      0      0
15.     11       0     23      0      0
16.     12       0     10      1      0
17.     12       0      5      1      0
18.     12       0      8      0      0
19.     13       0      3      1      0
20.     13       0     13      1      0
```

Gap timeアプローチでは，**stset** コマンドで**id()**オプションを使用しませんでした．これは，複数のオブザベーションが同一被験者に対応することをStataが認識していないことを意味します．しかしながら，**stcox** コマンドに直接**cluster()**オプションを使用することにより，IDごとに（被験者ごとに）クラスター化した解析が行われます．以下のコードは，**cluster()**および**robust**オプションを用いて，Gap timeアプローチによる層化Coxモデルを実行します．コードと出力は以下の通りです．

```
stcox tx num size, nohr robust strata(interval)cluster(id)
```

```
No. of subjects  =         190        Number of obs  =     190
No. of failures  =         112
Time at risk     =        2711
                                      Wald chi2(3)   =   11.99
Log likelihood   =  -363.16022        Prob > chi2    =  0.0074
              (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
  _t              Robust
  _d      Coef.   Std. Err.      z    p>|z|  [95% Conf. Interval]
------------------------------------------------------------------------------
  tx  -.2695213   .2093108  -1.29  0.198  -.6797628   .1407203
 num   .1535334   .0491803   3.12  0.002   .0571418   .2499249
size   .0068402   .0625862   0.11  0.913  -.1158265    .129507
------------------------------------------------------------------------------
                                          Stratified by interval
```

Gap timeアプローチを用いた解析結果は，層化CPアプローチの結果とわずかに異なります．

次に，共有frailtyモデルを再発イベントデータに適用する方法について説明します．frailtyは，被験者内相関につながる可能性のある観測されない被験者固有の因子による変動を説明するために，再発イベント解析に含めるものです．

モデルを実行する前に，すでに説明した**stset**コマンドを再実行し，データをCPアプローチに適した形式に戻します．コードは以下の通りです．

```
stset stop, failure(event==1) id(id) time0(start) exit(time .)
```

次に**streg**コマンドで，ガンマ分布の共有frailty成分をもつパラメトリックWeibullモデルを実行します．このセクションで紹介した他のモデルとの比較性を保つために，同じ3つの予測因子を使用します．コードは以下の通りです．

```
streg tx numsize, dist(weibull) frailty(gamma) shared(id) nohr
```

dist()オプションはパラメトリックモデルの分布を指定します．**frailty**()オプションはfrailtyの分布を指定し，**shared**()オプションはクラスター変数がIDであることを定義します．このモデルでは，同一被験者のオブザベーションが同じfrailtyを共有します．出力は以下の通りです．

```
Weibull regression -- log relative-hazard form
                     Gamma shared frailty

No. of subjects  =          85           Number of obs   =     190
No. of failures  =         112
Time at risk     =        2711
                                         LR chi2(3)      =    8.04
Log likelihood   =  -184.73658           Prob > chi2     =  0.0453

------------------------------------------------------------------------------
          _t      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------------------------------------------------------------------
          tx   -.4583219   .2677275    -1.71   0.087    -.9830582    .0664143
         num    .1847305   .0724134     2.55   0.011     .0428028    .3266581
        size   -.0314314   .0911134    -0.34   0.730    -.2100104    .1471476
       _cons   -2.952397   .4174276    -7.07   0.000     -3.77054   -2.134254
-------------------------------------------------------------------------------
      /ln_p   -.1193215   .0898301    -1.33   0.184    -.2953852    .0567421
    /ln_the   -.7252604   .5163027    -1.40   0.160    -1.737195    .2866742
-------------------------------------------------------------------------------
           p    .8875224   .0797262                      .7442449    1.058383
         1/p    1.126732   .1012144                      .9448377    1.343644
       theta    .4841985    .249993                      .1760134     1.33199
------------------------------------------------------------------------------
Likelihood ratio test of theta=0: chibar2(01) = 7.34
  Prob>=chibar2 = 0.003
```

このモデルの出力については第8章で論議しています.

被験者1名につき複数のオブザベーションを持つCPデータレイアウトは，再発イベントデータへの適応に必要なだけではなく，従来の1被験者1イベント型の生存時間解析にも使用することができます．4つのオブザベーションを持つ被験者に関して，4番目のオブザベーションの時間区間でイベントが発生する前の最初の3つのオブザベーションは打ち切りと考えることもできます．このデータレイアウトは，時間によって値が変わる可能性のある時間依存性の曝露を表すのに特に適しています(セクション7の**stsplit**コマンドを参照).

B. Stata

SASでは，適切なSASプロシジャをSASデータセットに用いて解析を行います．生存時間解析を実行する主要なSASプロシジャは以下の通りです.

PROC LIFETEST – このプロシジャはKaplan-Meier生存推定値やプロットを得るのに用います. 生命表法の推定値やプロットも出力できます．このプロシジャは，1つの変数で層別したときの，ログランク検定およびWilcoxon検定の統計量を出力します．生存推定値を含んだ新しいSASデータセットを作成できます.

PROC PHREG – このプロシジャは，Cox比例ハザードモデル，層化Coxモデル，時間依存性共変量をもつ拡張Coxモデルに用います．調整生存推定値を含むSASデータセットを作成することもできます．さらに，PROC GPLOTを用いて調整生存推定値をプロットすることもできます.

PROC LIFEREG – このプロシジャは，パラメトリック加速(AFT)モデルに用います.

薬物常用者データセットの解析事例でこれらのプロシジャの説明をします．薬物常用者データセットは，1991年のCaplehornらのオーストラリア試験から得られたもので, 238名のヘロイン常用者の情報が含まれます．この試験は2つのメタドン治療施設を比較するもので，患者のメタドン治療維持期間を評価指標としています．この2つの施設は，患者への院内方針に違いがありました．患者の生存時間は，施設の治療から脱落するか，または打ち切りかの時間(日)と定めました．変数はAppendixの冒頭で定義した通りです.

本書では，SASのプログラミングコードは読みやすいようにすべて大文字で記載しています．しかし，SASでは大文字と小文字の区別はありません. 文字間のスペースの数はプログラムに影響を及ぼしません（スペースは単語の区切りに使われます）. SASの各プログラミングステートメントはセミコロンで終わります.

薬物常用者データセットは，**addicts.sas7bdat**という名前の永久SASデータセットで保存されています．このSASデータセットの保存場所のパスを示すにはLIBNAMEステートメントが必要です．ここに紹介する例では，このファイルがCドライブにあると想定しています. LIBNAMEステートメントはパスに参照名を付けます．参照名をREFとします．コードは以下の通りです.

```
LIBNAME REF 'C:\';
```

参照名の定義は自由です．ファイルの保存場所のパスは引用符で囲みます. このコードの一般的な形式は以下の通りです（アルファベットと英数字だけが使用可能です．名前の先頭はアルファベットだけが使用できます）.

```
LIBNAME 好きな参照名 'ファイルの保管されているパス名';
```

PROC CONTENTS，PROC PRINT，PROC UNIVARIATE，PROC FREQ，PROC MEANSを用いてデータを表示したり要約することができます．SASコードは，一括で実行することも，プロシジャを1つずつ選択して個別に実行することもできます．コードを実行するには，編集ウィンドウからツールバーの実行ボタンをクリックします．これらのプロシジャを使用するためのコードは以下の通りです（出力は省略）.

```
PROC CONTENTS DATA=REF.ADDICTS;RUN;
PROC PRINT DATA=REF.ADDICTS;RUN;
PROC UNIVARIATE DATA=REF.ADDICTS;VAR SURVT;RUN;
PROC FREQ DATA=REF.ADDICTS;TABLES CLINIC PRISON;RUN;
PROC MEANS DATA=REF.ADDICTS;VAR SURVT;CLAS CLINIC;RUN;
```

それぞれのSASステートメントの終わりはセミコロンで区切ります．それぞれのプロシジャを1つずつ実行する場合は，プロシジャの終わりにRUNステートメントを入れます. 複数のプロシジャをまとめて実行する場合は，最後のプロシジャの終わりにRUNステートメントがあれば十分です．SASは，2レベルのファイル名でデータセットを認識します．1つはLIBNAMEステートメントで指定した参照名，もう1つは拡張子なしのファイル名です．この例ではSASファイル名をREF.ADDICTSとしています．あるいは，一時的なSASデータセットを作成してこれらのプロシジャに用いることもできます.

SAS実行に関わりのないテキストを，コメントとして書くことができます．

/* コメントの前には「/*」を付け，コメントの後ろには「*/」を付けます．*/
* コメントの前に「*」を付け，コメントの後ろに「;」を付ける方法もあります．

SASに関しては以下の生存時間解析を紹介します．

1. PROC LIFETESTによるKaplan-Meierおよび生命表生存推定値（およびプロット）の作成
2. PROC PHREGによるCox比例ハザードモデルの実行
3. 層化Coxモデルの実行
4. 統計的検定による比例ハザード仮定の評価
5. Cox調整生存曲線の作成
6. 拡張Coxモデル（時間依存性共変量を含む）の実行
7. PROC LIFEREGによるパラメトリックモデルの実行
8. 再発イベントのモデル構築

1．PROC LIFETESTによるKaplan-Meierおよび生命表生存推定値（およびプロット）の作成

PROC LIFETESTにMETHOD = KMオプションを使用して，Kaplan-Meier生存推定値を作成します．PLOTS = (S)オプションは推定生存関数をプロットします．TIMEステートメントは，time-to-event変数（SURVT）と打ち切りの値（STATUS = 0）を定義します．コードは以下の通りです（出力は省略）．

```
PROC LIFETEST DATA=REF.ADDICTS METHOD=KM PLOTS=(S);
TIME SURVT*STATUS(0);
RUN;
```

PROC LIFETESTにSTRATAステートメントを用いれば，生存推定値をグループ間で比較できます（例えばstrata clinic）．PLOTS = (S, LLS)オプションは，生存曲線と対数（－対数）曲線を生成します．比例ハザード仮定が成り立つならば，対数（－対数）の生存曲線は平行になります．STRATAステートメントは，ログランク検定およびWilcoxon検定の統計量も表示します．コードは以下の通りです．

```
PROC LIFETEST DATA=REF.ADDICTS METHOD=KM PLOTS=(S,LLS);
TIME SURVT*STATUS(0);
STRATA CLINIC;
RUN;
```

PROC LIFETEST は以下の出力（編集編）を作成します.

LIFETEST プロシジャ

層 1: CLINIC = 1

積極限法による生存推定

SURVT	生存率	死亡率	生存率の標準誤差	死亡数	生存数
0	1	0	0	0	163
2*	.	.	.	0	162
7	0.9938	0.00617	0.00615	1	161
17	0.9877	0.0123	0.00868	2	160
.	.	.	.	.	.
.	.	.	.	.	.
899	0.0181	0.9819	0.1720	122	1
905*	.	.	.	122	0

層 2: CLINIC = 2

積極限法による生存推定

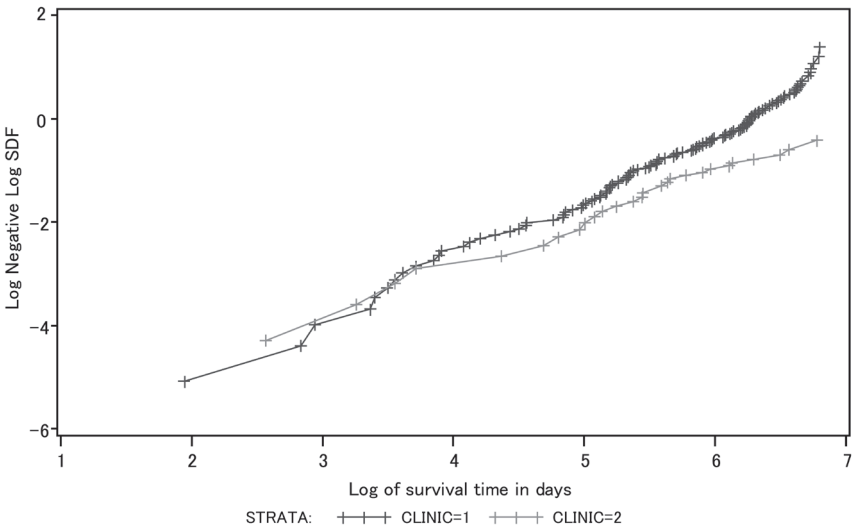
SURVT	生存率	死亡率	生存率の標準誤差	死亡数	生存数
0	1	0	0	0	75
2*	.	.	.	0	74
13	0.9865	0.0135	0.0134	1	73
26	0.9730	0.0270	0.0189	2	72
.	.	.	.	.	.
.	.	.	.	.	.

層に対しての同等性の検定

検定	カイ 2 乗	自由度	Pr > Chi-Square
ログランク	27.8927	1	<.0001
Wilcoxon	11.6268	1	0.0007
-2Log(LR)	26.0236	1	<.0001

ログランク検定とWilcoxon検定の結果はいずれも，非常に有意な結果でした．Wilcoxon検定はログランク検定の変法であり，j番目のfailure時間の（観測スコア－期待スコア）を，n_j（j番目のfailure時間のat risk数）で重み付けします．

PROC LIFETESTによる対数（－対数）プロットは以下の通りです．



ASは（StataやRも同様ですが）デフォルトでは，対数（－対数）曲線の横軸に生存時間ではなくlog（生存時間）をプロットします．平行性の確認に関しては，横軸にlog（生存時間）あるいは生存時間，どちらをとっても問題ではありません．しかしながら，log（生存時間）を横軸にした場合，対数（－対数）の生存曲線が直線のようになれば，time-to-event変数がWeibull分布に従うことを示唆することになります．もし，線の傾きが1となる場合は，生存時間変数がWeibull分布の特殊例である指数分布に従うことを示します．Weibull分布に従うと見なせる場合は，パラメトリック生存モデルが使えます．

生存推定値を含んだデータセットを作成することにより，さらに広がりのある変数値のプロット方法が可能となります．PROC LIFETESTステートメントのOUTSURV = オプションを使用すれば，KM生存推定値を含むSASデータを作成することができます．OUTSURV = DOGオプションは，変数名SURVIVALに生存推定値を格納した，dog（名前はユーザーが自由に指定できます）という名前のデータセットを作成します．コードは以下の通りです．

```
PROC LIFETEST DATA=REF.ADDICTS METHOD=KM OUTSURV=DOG;
TIME SURVT*STATUS(0);
STRATA CLINIC;
RUN;
```

データセットdogには生存推定値は含まれますが，生存推定値のlog(－(log))は含まれません．以下のコードにより，データセットdogから，対数(－対数)生存推定値を格納する新しい変数LLSを含むデータセットCATが作成されます．

```
DATA CAT;
SET DOG;
LLS=LOG(-LOG(SURVIVAL));
RUN;
```

SASでは，LOG関数は底10の対数ではなく自然対数を返します．

PROC PRINTは，データをアウトプットウィンドウに出力します．

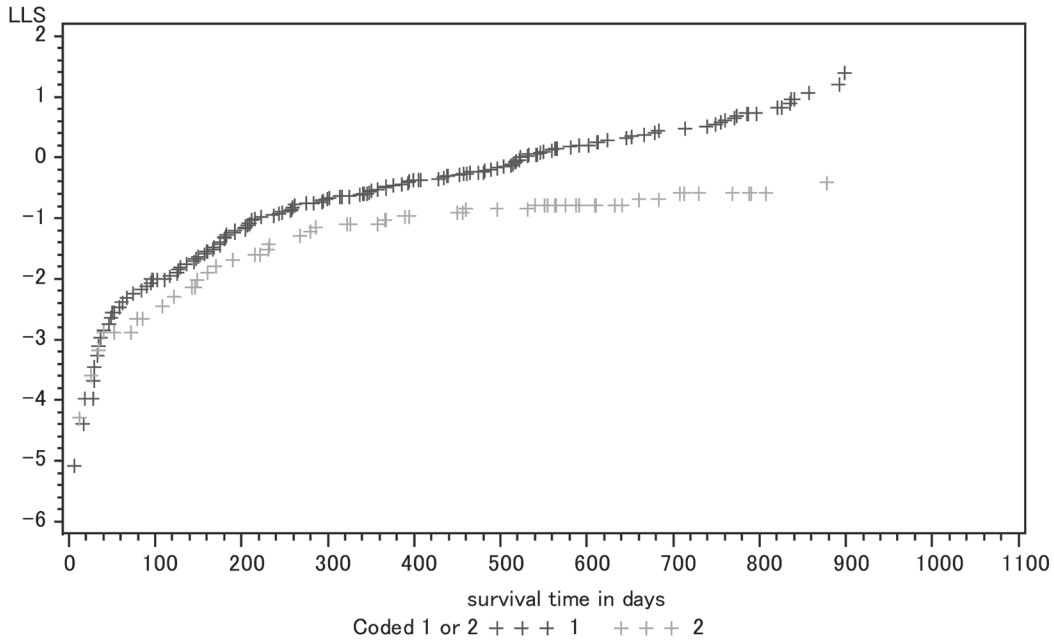
```
PROC PRINT DATA=CAT; RUN;
```

PROC PRINTにより出力された最初の10オブザベーションは以下の通りです．

```
Obs  CLINIC  SURVT  _CENSOR_  SURVIVAL     LLS

 1      1       0        .      1.00000     .
 2      1       2        1      1.00000     .
 3      1       7        0      0.99383   -5.08450
 4      1      17        0      0.98765   -4.38824
 5      1      19        0      0.98148   -3.97965
 6      1      28        1      0.98148   -3.97965
 7      1      28        1      0.98148   -3.97965
 8      1      29        0      0.97523   -3.68561
 9      1      30        0      0.96898   -3.45736
10      1      33        0      0.96273   -3.27056
```

PLOT LLS*SURVT = CLINICステートメントは，変数LLS(対数(－対数)生存変数)を縦軸に，SURVTを横軸に，CLINICで層別することを指定します．SYMBOLオプションは，CLINICの水準ごとに色を指定します．CLINIC別の対数(－対数)曲線をプロットするためのコードと出力は以下の通りです．

```
SYMBOL COLOR=BLUE;
SYMBOL2 COLOR=RED;

PROC GPLOT DATA=CAT;
PLOT LLS*SURVT=CLINIC;
RUN;
```



このプロットはデフォルトのlog(生存時間)ではなく，生存時間(日)を用いています．対数(－対数)生存時間プロットは，CLINICに関して最初の365日間は平行で，その後乖離していくようにみえます．拡張CoxモデルでCLINICを時間依存性変数としてモデル構築する際に，この情報は役に立ちます．

生命表法を用いて生存推定値を得ることもできます．この方法が役に立つのは，個人レベルの生存時間情報がない代わりに，特定の期間におけるグループの生存時間情報がある場合です．ユーザーはINTERVALS＝オプションを使用してこの時間区間を指定します．コードは以下の通りです(出力は省略)．

```
PROC LIFETEST DATA=REF.ADDICTS
  METHOD=LT
  INTERVALS= 50 100 150 200 TO 1000 BY 100
  PLOTS=(S);
TIME SURVT*STATUS(0);
RUN;
```

2. PROC PHREGによるCox比例ハザードモデルの実行

PROC PHREGはCox比例ハザードモデルに用います．コードは以下の通りです．

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)= PRISON DOSE CLINIC;
RUN;
```

MODELステートメントのコードSURVT*STATUS(0)は，time-to-event変数(SURVT)と打ち切りの値(STATUS = 0)を指定します．このモデルには3つの予測変数PRISON，DOSE，CLINICが含まれています．PROC PHREGのMODELステートメントのオプションRLは，ハザード比推定値に95%信頼区間を与えます．これらの各予測変数は比例ハザード仮定が成り立つと仮定しています(成り立たないかも知れませんが)．PROC PHREGによる出力は以下の通りです．

PHREG プロシジャ
モデルの情報

データセット	REF.ADDICTS	
従属変数	SURVT	survival time in days
打ち切り変数	STATUS	status (0=censored, 1=endpoint)
打ち切り値の数	0	
タイデータの処理	BRESLOW	

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
238	150	88	36.97

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	1411.324	1346.805
AIC	1411.324	1352.805
SBC	1411.324	1361.837

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比	95%ハザード比信頼限界	
PRISON	1	0.32650	0.16722	3.8123	0.0509	1.386	0.999	1.924
DOSE	1	-0.03540	0.00638	30.7844	<.0001	0.965	0.953	0.977
CLINIC	1	-1.00876	0.21486	22.0419	<.0001	0.365	0.239	0.556

上記の表は，回帰係数のパラメータ推定値，その標準誤差，各予測因子に対するWaldカイ2乗検定統計量，対応するp値を示します．“Hazard Ratio”列には，回帰係数推定値を指数化した，各予測因子の1単位変化あたりのハザード比推定値が示されます．最後の2列は，このハザード比の95%信頼限界を示しています．

実行モデルでは，デフォルトでMODELステートメントにTIES= BRESLOWオプションが使われていますが，その代わりに，TIES=EXACTオプションを使用することができます．TIES=EXACTオプションは，同時に発生したイベントを処理するための集約的な計算方法です．

多くのイベントが同時に発生する場合は，TIES = EXACTオプションが望ましいです．それ以外の場合は，EXACTオプションもデフォルトもほとんど違いはありません．SASで利用できる同順位処理アプローチには

TIES = EFRONオプションもあります．TIES = EFRONオプションは，Rではデフォルトに設定されています．

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)= PRISON DOSE CLINIC/TIES=EXACT RL;
RUN;
```

出力は以下の通りです．

PHREG プロシジャ
モデルの情報

データセット	REF.ADDICTS	
従属変数	SURVT	survival time in days
打ち切り変数	STATUS	status (0=censored, 1=endpoint)
打ち切り値の数	0	
タイデータの処理	EXACT	

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比	95％ハザード比信頼限界	
PRISON	1	0.32657	0.16723	3.8135	0.0508	1.386	0.999	1.924
DOSE	1	-0.03537	0.00638	30.7432	<.0001	0.965	0.953	0.977
CLINIC	1	-1.00980	0.21488	22.0832	<.0001	0.364	0.239	0.555

パラメータ推定値とその標準誤差は，TIES = EXACTオプションを使用しなかった以前のモデルとわずかに違うだけです．同順位処理方法は，Model Informationに出力されます．

PRISONとCLINIC，PRISONとDOSEとの交互作用を評価したいとします．2つの交互作用項を含む新しい一時SASデータセット（addicts2）を作成すれば，これらの項を含むモデルを実行できます．CLINICと PRISONの積項（CLIN_PR）とCLINICと DOSEの積項（CLIN_DO）を，以下のデータステップで定義します．

```
DATA ADDICTS2;
SET REF.ADDICTS;
CLIN_PR=CLINIC*PRISON;
CLIN_DO=CLINIC*DOSE;
RUN;
```

次に，これらの交互作用項(CLIN_PRとCLIN_DO)をモデルに加えます．CONTRASTステートメントを使用すれば，これらの交互作用項を一般化Wald検定で同時に検定できます．“CONTRAST”という単語の後ろには，ユーザーの指定する出力名を引用符で囲んだものを記載します．次にそれぞれの検定する共変量(積項)とその後ろに1を付けたものを，カンマで区切って列挙します(下記コード参照)．

```
PROC PHREG DATA=ADDICTS2;
MODEL SURVT*STATUS(0)= PRISON DOSE CLINIC CLIN_PR CLIN_DO;
CONTRAST "TEST INTERACTION" CLIN_PR 1, CLIN_DO 1;
RUN;
```

PROC PHREGの出力以下の通りです．

PHREG プロシジャ
モデルの情報

データセット	WORK.ADDICTS2	
従属変数	SURVT	survival time in days
打ち切り変数	STATUS	status (0=censored, 1=endpoint)
打ち切り値の数	0	
タイデータの処理	BRESLOW	

モデルの適合度統計量

基準	共変量なし	共変量あり
-2 LOG L	1411.324	1343.199
AIC	1411.324	1353.199
SBC	1411.324	1368.253

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
PRISON	1	1.19200	0.54137	4.848	0.0277	3.294
DOSE	1	-0.01932	0.01935	0.9967	0.3181	0.981
CLINIC	1	0.17469	0.89312	0.0383	0.8449	1.191
CLIN_PR	1	-0.73799	0.43149	2.9253	0.0872	0.478
CLIN_DO	1	-0.01386	0.01433	0.9359	0.3333	0.986

Contrast Test Results

Contrast	DF	Wald Chi-Square	Pr > ChiSq
TEST INTERACTION	2	3.5803	0.1669

積項がモデルに含まれる場合，それに関連するハザード比推定値(最右列)はほとんど意味がありません．例えば，PRISONの係数推定値をexp(1.19200) = 3.284と指数化すれば，DOSE = 0，CLINIC = 0でのPRISON = 1 vs. PRISON = 0のハザード比推定値が得られます．しかし，このハザード比には意味がありません．なぜならば，CLINICは0ではなく1または2であり，DOSEは常に0よりも大きい(すべての対象者がメタドン治療を受けている)からです．次のセクション(層化Coxモデル)で，交互作用項を含むモデルに関して，CONTRASTステートメントを用いて意味のあるハザード比推定値を求める方法を説明します．CONTRASTステートメントを用いると，上述の一般化Wald検定に加えてパラメータ推定値の線形結合が得られます．

2つの積項のWaldカイ2乗p値は，CLIN_PRが0.0872，CLIN_DOが0.3333です．両積項を同時に検定する一般化Waldカイ2乗p値は0.1669です．これとは別に，－2対数尤度統計量に関して，縮小モデル(積項を含まず)からフルモデル(2つの積項を含む)を引くことにより，尤度比検定で両積項を同時に検定することができます．－2対数尤度統計量は，出力の"Model Fit Statistics"の"With Covariates"列に示されています．－2log尤度統計量は，フルモデルでは1,343.199，縮小モデルでは1,346.805です．2積項を同時に検定するため，この検定の自由度は2です．

SASでは，PROBCHI関数を用いるとカイ2乗検定のp値が得られます．コードは以下の通りです．

```
DATA TEST;
REDUCED = 1346.805;
FULL = 1343.199;
DF = 2;
P_VALUE = 1 - PROBCHI(REDUCED-FULL,DF);
RUN;

PROC PRINT DATA=TEST;RUN;
```

(1 – PROBCHI関数)とすることで，カイ2乗確率密度関数の右裾部分の面積の確率が得られます．PROC PRINTによる出力結果は以下の通りです．

Obs	REDUCED	FULL	DF	P_VALUE
1	1346.81	1343.2	2	0.1648

両積項に対する尤度比検定のp値は0.16480であり，一般化Wald検定から得られたp値(0.1669)と似た結果となります．2交互作用項を同時に検定するため，これらの検定はどちらも自由度は2です．

3. 層化Coxモデルの実行

変数CLINICは比例ハザード仮定を満たさないが，変数PRISONとDOSEは，CLINICのそれぞれの水準内では比例ハザード仮定を満たすと想定します．変数CLINICに関する層化Coxモデルを，PROC PHREGとSTRATA CLINICステートメントを用いて実行します．コードは以下の通りです．

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)= PRISON DOSE/RL;
STRATA CLINIC;
RUN;
```

パラメータ推定値の出力は以下の通りです．

PHREG プロシジャ
最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比	95%ハザード比信頼限界	
PRISON	1	0.38877	0.16892	5.2974	0.0214	1.475	1.059	2.054
DOSE	1	-0.03514	0.00647	29.5471	<.0001	0.965	0.953	0.978

CLINICは層化変数であるため，CLINICのパラメータ推定値は出力されません．PRISON = 1 vs. PRISON = 0のハザード比は1.475と推定されます．PRISONとCLINICとの交互作用項はモデルに含まないため，このハザード比はCLINICに依存しないと仮定しています．

CLINICで層別したCoxモデルで，PRISONとCLINICとの交互作用，DOSEとCLINICとの交互作用を評価します．新しいSASデータセット（addicts2）で交互作用項を定義し，それらの項を含むモデルを実行します．

```
DATA ADDICTS2;
SET REF.ADDICTS;
CLIN_PR=CLINIC*PRISON;
CLIN_DO=CLINIC*DOSE;
RUN;
```

この交互作用モデルでは，DOSEを調整したCLINIC = 1における PRISON = 1 vs. PRISON = 0のハザード比は$\exp(\beta_1 + \beta_3)$であり，DOSEを調整したCLINIC = 2における PRISON = 1 vs. PRISON = 0のハザード比は$\exp(\beta_1 + 2\beta_3)$です．後者の計算は，分子のハザードの式に(PRISON = 1, CLINIC = 2)と分母のハザードの式に(PRISON = 0, CLINIC = 2)の値を代入することにより求められます(下記参照)．

$$HR = \frac{h_0(t)\exp[1\beta_1 + \beta_2 DOSE + (2)(1)\beta_3 + \beta_4 CLIN\_DO]}{h_0(t)\exp[0\beta_1 + \beta_2 DOSE + (2)(0)\beta_3 + \beta_4 CLIN\_DO]} = \exp(\beta_1 + 2\beta_3).$$

パラメータ推定値の線形結合の推定値を求めるときは，PROC PHREGで，CONTRASTステートメントとESTIMATE = オプションを指定します．また，CONTRASTステートメントを使用して，前述したように一般化Wald検定で2交互作用項を同時に検定することもできます．

以下のコードは，2交互作用項を含む層化Coxモデル(STRATA CLINIC)を実行します．3つのCONTRASTステートメントを使用しています．1つ目はCLINIC = 1における PRISONのハザード比$\exp(\beta_1 + \beta_3)$を推定するため，2つ目はCLINIC = 2における PRISONのハザード比$\exp(1 + 2\beta_3)$を推定するため，3つ目は，自由度2の一般化Wald検定で2交互作用項を同時に検定するためのものです．最初の2つのCONTRASTステートメントで使用するESTIMATE = EXPオプションは，パラメータ推定値の指数化値を要求します．2番目のCONTRASTステートメント内のコード PRISON 1 CLIN_PR 2/ESTIMATE = EXP;は$\exp(\beta_1 + 2\beta_3)$の推定値を要求します．β_1はPRISONに対応し，β3はモデルの3番目の変数CLIN_PRに対応します．コードは以下の通りです．

```
PROC PHREG DATA=ADDICTS2;
MODEL SURVT*STATUS(0)= PRISON DOSE  CLIN_PR CLIN_DO;
STRATA CLINIC;
CONTRAST 'HR FOR PRISON AMONG CLINIC=1' PRISON 1 CLIN_PR 1/ESTIMATE=EXP;
CONTRAST 'HR FOR PRISON AMONG CLINIC=2' PRISON 1 CLIN_PR 2/ESTIMATE=EXP;
CONTRAST 'TEST INTERACTION' CLIN_PR 1, CLIN_DO 1;
RUN;
```

CLINICで層別しているときは，変数CLINICはMODELステートメントに入れません．しかしながら，CLINICをSTRATAステートメントに指定しているにもかかわらず，交互作用項CLIN_PRおよびCLIN_DOはMODELステートメントに入れます．出力は以下の通りです．

PHREG プロシジャ
最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
PRISON	1	1.08716	0.53861	4.0741	0.0435	2.966
DOSE	1	-0.03482	0.01980	3.0930	0.0786	0.966
CLIN_PR	1	-0.58467	0.42813	1.8650	0.1721	0.557
CLIN_DO	1	-0.00105	0.01457	0.0052	0.9427	0.999

行ごとの対比推定と検定の結果

対比	タイプ	推定量	標準誤差	信頼限界	
HR FOR PRISON AMONG CLINIC=1	EXP	1.6528	0.3119	1.1419	2.3925
HR FOR PRISON AMONG CLINIC=2	EXP	0.9211	0.3540	0.4337	1.9563

対比検定の結果

対比	自由度	カイ2乗	Wald Pr > ChiSq
HR FOR PRISON AMONG CLINIC=1	1	7.0918	0.0077
HR FOR PRISON AMONG CLINIC=2	1	0.0457	0.8307
TEST INTERACTION	2	1.8650	0.3936

ハザード比（PRISON = 1 vs. PRISON = 0）は，CLINIC = 1では1.6528と推定され，CLINIC = 2では0.9211と推定されます．両交互作用項を同時に検定する一般化Wald検定（自由度2：1は $\beta_3 = 0$，1は $\beta_4 = 0$）の p 値は0.3936です．

CLINICとその他の共変量に交互作用が存在するときに取り得る別のアプローチは，CLINIC = 1とCLINIC = 2の部分集団でそれぞれ別々に解析することです．コードと出力は以下の通りです．

SASプロシジャ内のWHEREステートメントは，解析用のオブザベー

```
PROC PHREG DATA=ADDICTS2;
MODEL SURVT*STATUS(0)=PRISON DOSE;
WHERE CLINIC=1;
TITLE 'COX MODEL RUN ONLY ON DATA WHERE CLINIC=1';
RUN;
```

ションを抽出するものです．TITLEステートメントをこのプロシジャに加えることもできます．CLINIC = 1のオブザベーションだけの解析の出力結果（パラメータ推定値を含む）は以下の通りです．

COX MODEL RUN ONLY ON DATA WHERE CLINIC=1
最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
PRISON	1	0.50249	0.18869	7.0918	0.0077	1.653
DOSE	1	-0.03586	0.00774	21.4761	<.0001	0.965

同様に，CLINIC = 2のオブザベーションだけの解析に関するコードと出力（パラメータ推定値を含む）は以下の通りです．

```
PROC PHREG DATA=ADDICTS2;
MODEL SURVT*STATUS(0)=PRISON DOSE;
WHERE CLINIC=2;
TITLE 'COX MODEL RUN ONLY ON DATA WHERE CLINIC=2';
RUN;
```

COX MODEL RUN ONLY ON DATA WHERE CLINIC=2
最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
PRISON	1	-0.08226	0.38430	0.0458	0.8305	0.921
DOSE	1	-0.03693	0.01234	8.9500	0.0028	0.964

CLINIC = 2におけるDOSEを調整したPRISON = 1 vs. PRISON = 0のハザード比推定値は0.921です．この結果は，以前に実行した，CLINICに関するすべての積項を含む層化Coxモデルの結果と一致しています．

4. 統計的検定による比例ハザード仮定の評価

これから説明するSASプログラムは，薬物常用者データセットを用いて，特定の共変数に関する比例ハザード仮定の統計的検定を実行する方法を示しています（Harrel and Lee，1986）．これは，特定の共変数に関するSchoenfeld残差とfailure時間順位との相関に基づいています．比例ハザード仮定が成り立つならば，この相関はほぼ0となるはずです．この相関検定の*p*値は，PROC CORR（またはPROC REG）で求められます．特定のモデルに対するSchoenfeld残差は，PROC PHREGを用いてSASデータセットに保存できます．PROC RANKを用いて，イベントのfailure時間を順位化してSASデータセットに保存できます．帰無仮説は，比例ハザード仮定が成立するというものです．

まず，CLINIC，PRISON，DOSEを含むモデルを実行します．OUTPUTステートメントでSASデータセットを作成します．OUT=オプションは出力データセット名を定義し，RESSCH =ステートメントの後ろにユーザー定義の変数名を入れます．これら変数にSchoenfeld残差が格納されます．変数名の順序は，MODELステートメントで指定した独立変数の順序に対応します．実際の変数名は任意です．ここでは，SASデータセットの名前をRESIDとし，CLINIC，PRISON，DOSEに関するSchoenfeld残差を含む変数の名前をそれぞれRCLINIC，RPRISON，RDOSEとしました．

コードは以下の通りです．

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=CLINIC PRISON DOSE;
OUTPUT OUT=RESID RESSCH=RCLINIC RPRISON RDOSE;
RUN;

PROC PRINT DATA=RESID;RUN;
```

PROC PRINTによる出力の最初の10個のオブザベーションは以下の通りです．右側の3列はSchoenfeld残差の変数です．

OBS	SURVT	STATUS	CLINIC	PRISON	DOSE	RCLINIC	RPRISON	RDOSE
1	428	1	1	0	50	-0.18715	-0.40641	-8.2100
2	275	1	1	1	55	-0.15841	0.55485	-2.6277
3	262	1	1	0	55	-0.16453	-0.45197	-2.5635
4	183	1	1	0	30	-0.14577	-0.48727	-26.0823
5	259	1	1	1	65	-0.16306	0.54313	7.3701
6	714	1	1	0	55	-0.25853	-0.50074	-8.5347
7	438	1	1	1	65	-0.19292	0.58106	6.6072
8	796	0	1	1	60	.	.	.
9	892	1	1	0	50	-0.34478	-0.22372	-15.9088
10	393	1	1	1	65	-0.17712	0.57376	6.6886

次に，打ち切りオブザベーションを削除したSASデータセット（つまり，failureオブザベーションのみを含む）を作成します．

```
DATA EVENTS;
SET RESID;
IF STATUS=1;
RUN;
```

PROC RANKを用いて，failure時間の順序変数を含むデータセットを作成します．OUT＝オプションを用いて出力データセットの名前を指定します．順位化を行う変数はSURVTです．RANKSステートメントの後に，failure時間の順位を格納する変数名を指定します．変数名は任意です．ここでは，この変数名をTIMERANKとしました．コードは以下の通りです．

```
PROC RANK DATA=EVENTS OUT=RANKED TIES=MEAN;
VAR SURVT;
RANKS TIMERANK;
RUN;

PROC PRINT DATA=RANKED;RUN;
```

PROC CORRは，failure時間順位変数（この例ではTIMERANK）とCLINIC，PRISON，DOSEに関するSchoenfeld残差の変数（この例ではそれぞれRCLINIC，RPRISON，RDOSE）との間の相関を求めるのに用います．NOSIMPLEオプションは要約統計量の出力を抑制します．特定の共変数に関して比例ハザード仮定が成り立つのであれば，相関はほぼ0のはずです．PROC CORRによる相関が0かどうかの検定のp値は，比例ハザード仮定の検定のp値になります．コードは以下の通りです．

```
PROC CORR DATA=RANKED NOSIMPLE;
VAR RCLINIC RPRISON RDOSE;
WITH TIMERANK;
RUN;
```

PROC CORRの出力は以下の通りです．

CORR プロシジャ

Pearsonの相関係数，N = 150
H0: Rho=0 に対するProb > |r|

	RCLINIC	RPRISON	RDOSE
TIMERANK	-0.26153	-0.07970	0.07733
変数SURVTの順位	0.0012	0.3323	0.3469

上記に相関係数とその下にp値を示します．CLINIC，PRISON，DOSEのp値はそれぞれ0.0012，0.3323，0.3469であり，CLINICに関しては比例ハザード仮定が成立せず，一方，PRISONとDOSEに関しては成立とみなせることが示唆されます．

同じp値は，PROC REGを用いて予測因子ごとに（一度に1つ）線形回帰を実行し，回帰係数に対するp値として求めることができます．以下のコードで，CLINICに対するp値を含む出力を行います．

```
PROC REG DATA=RANKED;
MODEL TIMERANK=RCLINIC;
RUN;
```

PROC REGによる出力は以下の通りです.

REG プロシジャ
パラメータ推定値

変数	自由度	パラメータ推定値	標準誤差	t値	Pr > |t|
Intercept	1	75.49955	3.43535	21.98	<.0001
RCLINIC	1	-28.38848	8.61194	-3.30	0.0012

右端の列に示されるCLINICのp値(0.0012)は，PROC CORRを用いたp値と一致しています.

5. Cox調整生存曲線の作成

PROC PHREGのBASELINEステートメントを用いて，特定の共変量パターンに対するCox調整生存推定値を格納する出力データセットを作成します．事前に興味ある特定の共変量パターンをSASデータセットに作成し，それを後に，PROC PHREGのBASELINEステートメントのCOVARIATES =オプションの入力データセットとして使用します．それぞれの共変量パターンごとに異なる生存曲線が得られます(共変量の効果が0ではないとき)．比例ハザード仮定を評価するための調整log(－log)生存時間プロットを得ることもできます．これについて例を3つ挙げて説明します.

例1 – PRISON，DOSE，CLINICを用いた比例ハザードモデルを実行し，PRISON = 0，DOSE = 70，CLINIC = 2における調整生存曲線を求めます.

例2 – 層化Coxモデル(CLINICで層別)を実行し，PRISONとDOSEの平均値を用いて，CLINIC = 1およびCLINIC = 2に関する2本の調整生存曲線を求めます. 対数(－対数)曲線を用いて, PRISONとDOSEで調整したときのCLINICの比例ハザード仮定を評価します.

例3 – 層化Coxモデル(CLINICで層別)を実行し, PRISON = 0, DOSE = 70, およびPRISON = 1，DOSE = 70における調整生存曲線を求めます．ここではCLINIC = 1に関して2つ，CLINIC = 2に関して2つの合計4つの生存曲線が得られます.

基本的には，3つのステップがあります．

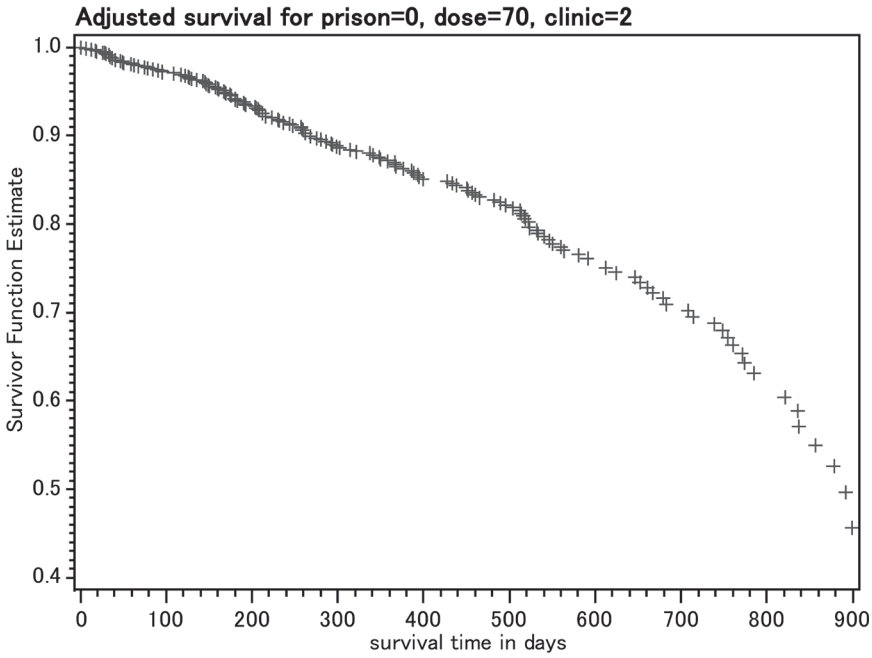
1） 調整生存曲線に用いた変数の共変量パターン（値）を含む入力データセットを作成します．
2） ステップ1）で作成した入力データセットをBASELINEステートメントで指定してPROC PHREGでCoxモデルを実行し，調整生存推定値を含むデータセットを出力します．
3） ステップ2）で作成した出力データセットの調整生存推定値をプロットします．

例1では，PRISON = 0，DOSE = 70，CLINIC = 2の，1オブザベーションからなる入力データセット（IN1）を作成します．次に，モデルを実行して，調整生存推定値を格納する変数（S1）を含む出力データセット（OUT1）を作成します．最後に，PROC GPLOTを使用して調整生存曲線をプロットします．コードは以下の通りです．

```
DATA IN1;
INPUT PRISON DOSE CLINIC;
CARDS;
0 70 2
;
RUN;
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON DOSE CLINIC;
BASELINE COVARIATES=IN1 OUT=OUT1 SURVIVAL=S1/NOMEAN;
RUN;

PROC GPLOT DATA=OUT1;
PLOT S1*SURVT;
TITLE 'Adjusted survival for prison=0, dose=70, clinic=2';
RUN;
```

PROC PHREGにおけるBASELINEステートメントでは，入力データセット，出力データセット，調整生存推定値を含む変数名を指定します．NOMEANオプションは，PRISON，DOSE，CLINICの平均値を用いた生存推定値の出力を抑制します．次の例（例2）ではNOMEANオプションを使用しません．

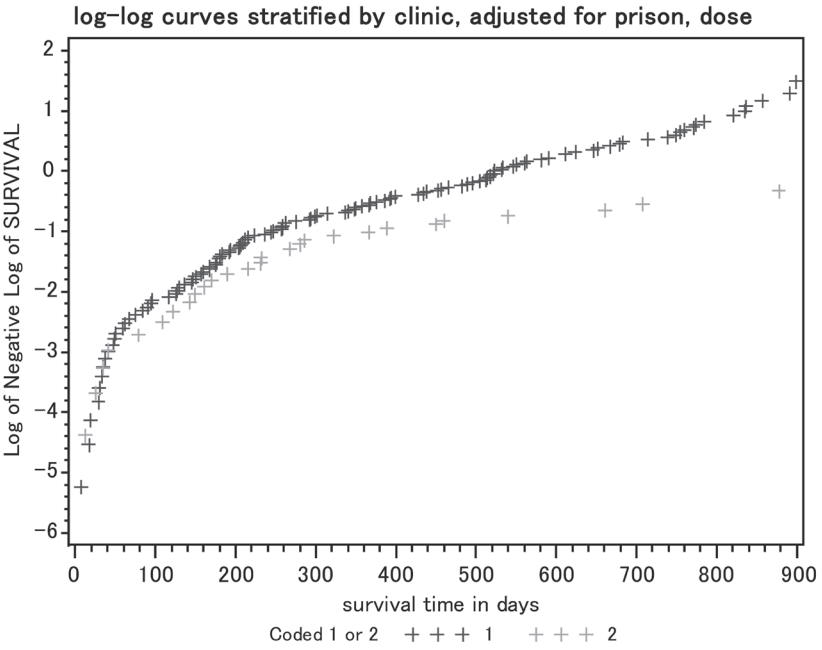
PROC GPLOTの出力は以下の通りです.



例2では，PRISONおよびDOSEの平均値を用い，CLINICで層別したCoxモデルから調整生存推定値を含むデータセット(OUT2)を作成し，出力します．BASELINEステートメントでNOMEANオプションを使用しない場合，PRISONとDOSEの平均値がデフォルトで使用されるため，入力データセットを指定する必要はありません．コードは以下の通りです.

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0) = PRISON DOSE;
STRATA CLINIC;
BASELINE OUT=OUT2 SURVIVAL=S2 LOGLOGS=LS2;
RUN;

PROC GPLOT DATA=OUT2;
PLOT S2*SURVT=CLINIC;
TITLE 'Adjusted survival stratified by clinic';
RUN;

PROC GPLOT DATA=OUT2;
PLOT LS2*SURVT=CLINIC;
TITLE 'log-log curves stratified by clinic, adjusted for
  prison, dose';
RUN;
```

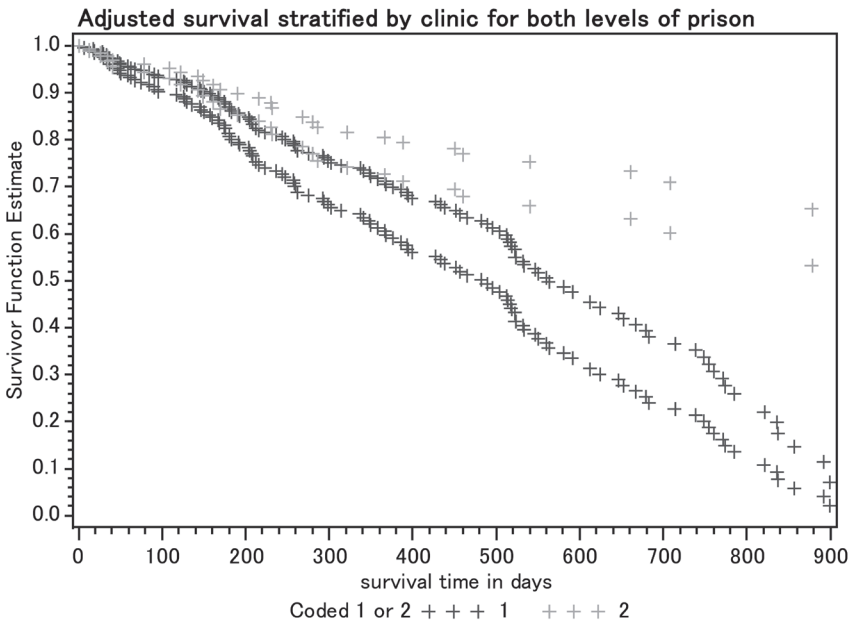
2番目のPROC GPLOTのコードPLOT LS2*SURVT = CLINICは，LS2を縦軸に，SURVTを横軸にとり，CLINIC別のプロットを1図表内に作成します．変数LS2はPROC PHREGのBASELINEステートメントで作成されたものであり，調整対数(－対数)生存推定値を格納しています．CLINICで層別し，PRISONとDOSEで調整した対数－対数生存曲線のPROC GPLOTによる出力は以下の通りです．



調整対数(－対数)プロットはすでに紹介した未調整対数(－対数)Kaplan-Meierプロットとよく似ており，365日まではプロットはほぼ平行であるがその後は乖離していくので，1年以後は比例ハザード性が成立しないことを示唆しています．

例3では，層化Cox(CLINICによる層別)を実行し，DOSE = 70におけるPRISON = 1とPRISON = 0に関する調整曲線を求めます．入力データセット(IN3)では，DOSE = 70とそれぞれのPRISON水準に対応する2行のオブザベーションが作成されます．出力データセット(OUT3)は，BASELINEステートメントで作成され，4本の曲線(CLINIC 2 × PRISON 2)に対応する生存推定値の変数(S3)を含みます．コードは以下の通りです．

```
DATA IN3;
INPUT PRISON DOSE;
CARDS;
1 70
0 70
;
RUN;

PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)= PRISON DOSE;
STRATA CLINIC;
BASELINE COVARIATES=IN3 OUT=OUT4 SURVIVAL=S3/NOMEAN;
RUN;

PROC GPLOT DATA=OUT4;
PLOT S3*SURVT=CLINIC;
TITLE 'Adjusted survival stratified by clinic for both levels
  of prison';
RUN;
```



上記のグラフでは，CLINIC を層別変数に用いているので，CLINICに関しては比例ハザード性を仮定していません．しかしPRISONについては，CLINICの各層内(CLINIC = 1とCLINIC = 2)で比例ハザード性を仮定しています．

6. 拡張Coxモデルの実行

時間依存性変数を含むモデルは, PROC PHREGを用いて実行します. 時間依存性変数は，PROC PHREGプロシジャ内のプログラミングステートメントで作成します．時々，ユーザーは間違ってデータステップで時間依存性変数を定義してしまうことがあります．データステップで用いる時間変数(SURVT)は時間独立であり，PROC PHREGステートメントで時間変数 (SURVT)を用いて定義する時間依存性変数とは異なりますので，データステップでの定義は誤った結果につながります．詳細については，第6章の拡張Cox尤度に関する記述を参照してください.

CLINICで層別したKM対数(－対数)曲線とCox調整対数(－対数)曲線のプロットについて曲線の平行性を確認することで，変数CLINICに関する比例ハザード性を評価しました．変数PRISONとDOSEについても同様の解析を行うことが可能です．ただし，DOSEに関しては，プロットして層間を比較するためには，連続変数をカテゴリ化する必要があります.

DOSEの効果に関するハザード比が時間とともに単調増加(または減少)すると予想される場合は，DOSEと何らかの時間関数を用いた時間依存性連続量積項を追加することもできます．以下に定義するモデルは，DOSEと時間(SURVT)の自然対数の積として定義される時間依存性変数(LOGTDOSE)を含みます．特定の変数について比例ハザード仮定が成り立たないということは，何らかの意味でその変数と時間との間に交互作用があることを示しています．変数LOGTDOSEがデータステップではなくPHREGプロシジャ内で定義されていることに注意してください．コードは以下の通りです.

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON CLINIC DOSE LOGTDOSE;
LOGTDOSE=DOSE*LOG(SURVT);
RUN;
```

PROC PHREGによる出力は以下の通りです.

PHREG プロシジャ
最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
PRISON	1	0.34047	0.16747	4.1333	0.0420	1.406
CLINIC	1	-1.01857	0.21538	22.3655	<.0001	0.361
DOSE	1	-0.08243	0.03599	5.2468	0.0220	0.921
LOGTDOSE	1	0.00858	0.00646	1.7646	0.1841	1.009

時間依存性変数LOGTDOSEのWald検定のp値は0.1841です．p値が有意でないことは，必ずしも比例ハザード仮定がDOSEに関して成立していることを意味している訳ではありません. 別の定義の時間依存性変数(例えばDOSE × (TIME – 100))が有意になる可能性もあるからです．また，試験例数は帰無仮説を棄却する検出力に大きく影響します(この場合の帰無仮説は比例ハザード仮定).

次に, CLINICに関する時間依存性変数を考えます．次の2つのモデルでは，CLINICに関して365日前後で異なるハザード比を推定するHeaviside階段関数を使用しています．1番目のモデルでは，モデル中に2つのHeaviside関数(HV1とHV2)を使用しますが, CLINICは用いません．2番目のモデルはHeaviside関数は1つのみ(HV)ですが，CLINICが含まれます．これら2つのモデルから得られるCLINICに関するハザード比推定値は同じですが，コードは異なります．2つのHeaviside関数を持つモデルのコードおよび出力は以下の通りです.

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON DOSE HV1 HV2;
IF SURVT < 365 THEN HV1 = CLINIC; ELSE HV1 = 0;
IF SURVT >= 365 THEN HV2 = CLINIC; ELSE HV2 = 0;
CONTRAST 'TEST EQUALITY OF HEAVISIDES' HV1 1 HV2 -1;
RUN;
```

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
PRISON	1	0.37770	0.16840	5.0304	0.0249	1.459
DOSE	1	-0.03551	0.00644	30.4503	<.0001	0.965
HV1	1	-0.45956	0.25529	3.2405	0.0718	0.632
HV2	1	-1.82823	0.38595	22.4392	<.0001	0.161

対比検定の結果

対比	自由度	Wald カイ2乗	Pr > ChiSq
TEST EQUALITY OF HEAVISIDES	1	8.7993	0.0030

HV1とHV2に関するパラメータ推定値から直接，365日前後のCLINIC＝1に対するCLINIC＝2のハザード比推定値を求めることができます．100日でのCLINICのハザード比推定値はexp（－0.45956）＝0.632であり，400日でのCLINICのハザード比推定値はexp（－1.82823）＝0.161です．CONTRASTステートメントは，2つのHeaviside変数の係数値が等しい（$\beta_3 = \beta_4$または$\beta_3 - \beta_4 = 0$）かのWald検定を行います．2つのHeaviside変数の係数値が等しいならば，CLINICのハザード比は時間によらないことになります．つまり，この検定は，比例ハザード性の乖離の1パターンを検定するものだとみなすことができます．この検定のp値は0.0030と非常に有意であり，CLINICに関しては比例ハザード仮定が成立しないことを示唆しています．

前述と同等な，ヘビサイド関数が1つのモデルに関するコードおよび出力は以下の通りです．

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=CLINIC PRISON DOSE HV;
IF SURVT >= 365 THEN HV = CLINIC; ELSE HV = 0;
CONTRAST 'HR FOR CLINIC <365 days' CLINIC 1/ESTIMATE=EXP;
CONTRAST 'HR FOR CLINIC >=365 days' CLINIC 1 HV 1/ESTIMATE=EXP;
RUN;
```

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	カイ2乗	Pr > ChiSq	ハザード比
CLINIC	1	-0.45956	0.25529	3.2405	0.0718	0.632
PRISON	1	0.37770	0.16840	5.0304	0.0249	1.459
DOSE	1	-0.03551	0.00644	30.4503	<.0001	0.965
HV	1	-1.36866	0.46139	8.7993	0.0030	0.254

対比	タイプ	推定量	標準誤差	信頼限界	
HR FOR CLINIC <365 days	EXP	0.6316	0.1612	0.3829	1.0416
HR FOR CLINIC >=365 days	EXP	0.1607	0.0620	0.0754	0.3424

変数CLINICがモデルに含まれており，時間依存性Heaviside関数の係数HVは365日まではハザード比推定に寄与しないことに注意してください．CONTRASTステートメント内のESTIMATE＝EXPオプションを用いて計算した，100日でのCLINICのハザード比推定値はexp（－0.45956）＝0.6316であり，400日でのCLINICのハザード比推定値はexp（（－0.45956）＋（－1.36866））＝0.1607です．これらの結果は，2つのHeaviside関数を持つモデルの推定値と一致しています．変数HVについてのWald検定のp値0.003は統計学的に有意であり，CLINICに関する比例ハザード仮定が成立しないことを示唆しています．これは2つのHeaviside関数を持つモデルでのCONTRASTステートメントによるものと同じ検定です．

CLINIC = 2 vs. CLINIC = 1のハザード比が最初の1年間は一定であるがその後は単調増加（または減少）すると想定します．以下のコードは，366日以後にCLINICのハザード比に寄与する時間依存性共変量CLINTIME（コードで定義）を含むモデルを定義します（出力は省略）．

```
PROC PHREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=CLINIC PRISON DOSE CLINTIME;
IF SURVT < 365 THEN CLINTIME=0;
ELSE IF SURVT >= 365 THEN CLINTIME = CLINIC*(SURVT-365);
RUN;
```

SASは，様々なデータ形式による時間依存性共変量のモデル化に対応しているという点で柔軟性があります．この点を説明するために，第6章で挙げた例について考えます．下記のデータ（D1）には，49ヵ月にイベントを経験した（MONTHS = 49, STATUS = 1）Janeのオブザベーションが1つ含まれています．Janeへの薬剤用量は，フォローアップ開始時は60 mg（DOSE1 = 60, TIME1 = 0）でしたが，フォローアップ12ヵ月で120 mg（DOSE2 = 120, TIME2 = 12）に，フォローアップ30ヵ月には150 mg（DOSE3 = 120, TIME3 = 30）に変更しました．

(Data D1) DOSE changes at 3 time points for Jane

ID	MONTHS	STATUS	DOSE1	TIME1	DOSE2	TIME2	DOSE3	TIME3
Jane	49	1	60	0	120	12	150	30

用量が複数の時点で測定されたならば，用量を時間依存性共変量として取り扱いたいという要望も出てきます．Janeのオブザベーションは多くの個人の代表例だと想定します．以下のコードは，上記の形式のデータ形式に対応した拡張Coxモデルを実行します．

```
PROC PHREG DATA=D1;
MODEL MONTHS*STATUS(0)=T_DOSE;
IF MONTHS<=TIME2 THEN T_DOSE=DOSE1;
ELSE IF MONTHS<=TIME3 THEN T_DOSE=DOSE2;
ELSE T_DOSE=DOSE3;
RUN;
```

時間依存性変数T_DOSEはMODELステートメントの下で定義しており，対応する時点ごとにDOSE1，DOSE2，DOSE3を採用しています．

これとは別の方法は，Janeの3つの用量のリスク期間に対応する3個のオブザベーションを持つCP形式のデータに変換するやり方もあります．

以下のコードはデータ（D1）をCP形式（D2）に変換します．

```
DATA D2;
SET D1;
START=TIME1;STOP=TIME2;EVENT=0;DOSE=DOSE1;OUTPUT;
START=TIME2;STOP=TIME3;EVENT=0;DOSE=DOSE2;OUTPUT;
START=TIME3;STOP=MONTHS;EVENT=1;DOSE=DOSE3;OUTPUT;
DROP MONTHS DOSE1 DOSE2 DOSE3 TIME1 TIME2 TIME3 STATUS;
RUN;
```

データ（D2）は，例えばJaneについては3オブザベーションとなるように変換したものであり，その結果，DOSEは時間依存性変数に適応している．最初の区間（START = 0，STOP = 12）ではJaneの用量は60 mg，2番目の区間（12～30月）では120 mg，3番目の区間（30～49月）では150 mgです．このデータは，Janeは49月にイベントがあったこと（STOP = 49，STATUS = 1）を示しています．Janeの3オブザベーションを次にに示します．

```
PROC PRINT DATA=D2;RUN;
```

ID	START	STOP	EVENT	DOSE
JANE	0	12	0	60
JANE	12	30	0	120
JANE	30	49	1	150

CP形式のデータでモデルを実行するためのコードは以下の通りです.

```
PROC PHREG DATA=D2;
MODEL (START,STOP)*EVENT(0)=DOSE;
RUN;
```

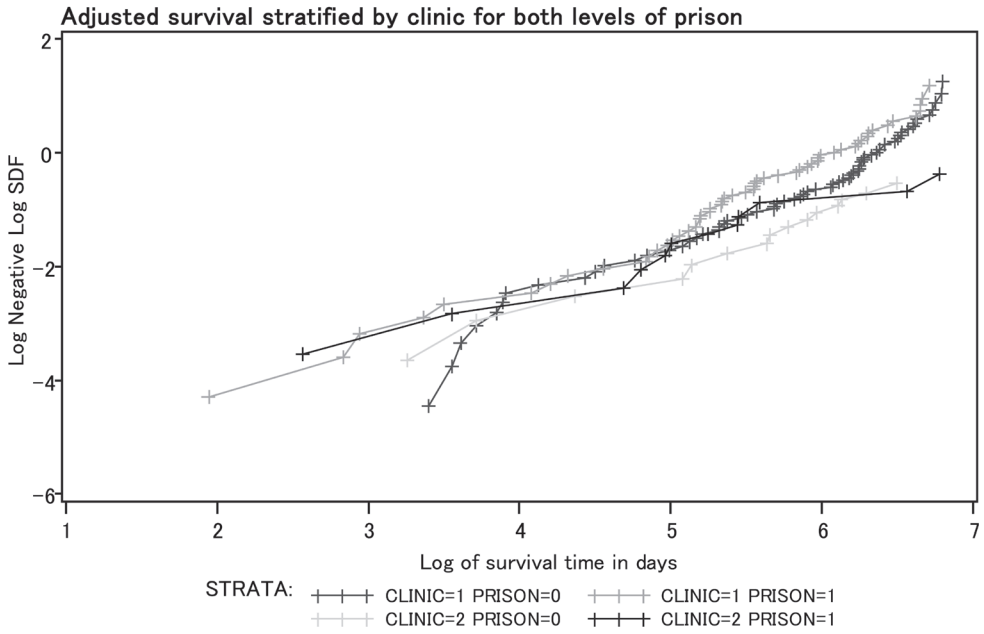
7. PROC LIFEREGによるパラメトリックモデルの実行

PROC LIFEREGは, 比例ハザードモデルではなくパラメトリック加速モデルを実行します. 比例ハザードモデルの重要な仮定が「ハザード比は時間によらず一定」であるのに対して，加速モデルの主要な仮定は，「共変量のレベル間で，生存時間は定数倍に加速（または減速）する」です.

生存データのパラメトリックモデルに最も一般的に用いられる分布はWeibull分布です．Weibull分布のハザード関数はλpt^{p-1}です．もしp = 1ならば，Weibull分布は指数分布でもあります．加速時間仮定が成り立てば比例ハザード仮定も成り立つという点において，Weibull分布には好ましい性質があります．指数分布はWeibull分布の特殊な場合です．指数分布の重要な性質は，「ハザードは時間によらず一定（$h(t) = \lambda$）」です．SASでは，Weibullモデルと指数モデルは加速時間モデルで実行されます.

Weibull分布には，対数（－対数）生存関数は対数時間と直線関係にあるという性質があります．PROC LIFETESTでは, 対数時間に対するKaplan-Meier対数（－対数）曲線をプロットできます．曲線がほぼ直線（かつ平行）ならば，Weibull仮定に適合しています．さらに，これらの直線の傾きが1であれば，指数分布にも適合しています．以下のコードは，CLINIC別，PRISON別に層化された対数（－対数）曲線を作成します．これら曲線を用いて，これら変数に関するWeibull仮定の妥当性を調べることができます.

```
PROC LIFETEST DATA=REF.ADDICTS METHOD=KM PLOTS=(LLS);
TIME SURVT*STATUS(0);
STRATA CLINIC PRISON;
RUN;
```



対数(－対数)曲線は直線には見えませんが，説明上，Weibull仮定に適合しているとして話を進めます．まず，PROC LIFEREGを用いて指数モデルを実行します．このモデルでは，Weibull形状パラメータ(p)は1に固定され，ハザードは一定になっています．

```
PROC LIFEREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON DOSE CLINIC/DIST=EXPONENTIAL;
RUN;
```

MODELステートメントのDIST = EXPONENTIALオプションは指数分布を指定します．PROC LIFEREGによるパラメータ推定値の出力は以下の通りです．

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	3.6843	0.4307	2.8402	4.5285	73.17	<.0001
PRISON	1	-0.2526	0.1649	-0.5758	0.0705	2.35	0.1255
DOSE	1	0.0289	0.0061	0.0169	0.0410	22.15	<.0001
CLINIC	1	0.8806	0.2106	0.4678	1.2934	17.48	<.0001
Scale	0	1.0000	0.0000	1.0000	1.0000		
Weibull Shape	0	1.0000	0.0000	1.0000	1.0000		

この指数モデルではハザードは一定と仮定しています．これは，出力にWeibull形状パラメータ値(1.0000)で示されます．この出力から，特定の共変量パターンを持つ被験者のハザード推定値を計算できます．例えば，PRISON = 0，DOSE = 50，CLINIC = 2の被験者のハザードD推定値は$\exp\{-(3.6843 + 50 \times 0.0289) + 2 \times 0.8806\} = 0.001$です．SASは加速時間の形式で指数モデルのパラメータ推定値を与えます．係数推定値に－1を掛けると，比例ハザード形式のパラメータ推定値になります(第7章を参照)．

次に，PROC LIFEREGを用いてWeibull加速モデルを実行します．

```
PROC LIFEREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON DOSE CLINIC/DIST=WEIBULL;
RUN;
```

MODELステートメントのDIST = WEIBULLオプションはWeibull分布を指定します．パラメータ推定値の出力は以下の通りです．

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.1048	0.3281	3.4619	4.7478	156.56	<.0001
PRISON	1	-0.2295	0.1208	-0.4662	0.0073	3.61	0.0575
DOSE	1	0.0244	0.0046	0.0154	0.0334	28.32	<.0001
CLINIC	1	0.7090	0.1572	0.4009	1.0172	20.34	<.0001
Scale	1	0.7298	0.0493	0.6393	0.8332		
Weibull Shape	1	1.3702	0.0926	1.2003	1.5642		

Weibull形状パラメータは1.3702と推定されます．SASは，Scaleパラメータと呼ぶWeibull形状パラメータの逆数を出力し，その値は0.7298です．CLINIC = 2とCLINIC = 1を比較する加速係数は$\exp(0.7090) = 2.03$と推定されます．つまり，メディアン生存時間推定値(ヘロインから離れている時間)は，CLINIC = 1に比べCLINIC = 2は2倍の長さです．

ハザード比パラメータをWeibull加速モデルから得るには，Weibull形状パラメータに加速時間パラメータのマイナス値を掛けます(第7章を参照)．例えば，他の共変量で調整したCLINIC = 2 vs. CLINIC = 1のハザード比推定値は，$\exp(1.3702(-0.7090)) = 0.38$となります．

次に, PROC LIFEREGを用いて対数ロジスティック加速モデルを実行します.

```
PROC LIFEREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON DOSE CLINIC/DIST=LLOGISTIC;
RUN;
```

対数ロジスティックパラメータ推定値の出力は以下の通りです.

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr >ChiSq
Intercept	1	3.5633	0.3894	2.8000	4.3266	83.71	<.0001
PRISON	1	-0.2913	0.1440	-0.5734	-0.0091	4.09	0.0431
DOSE	1	0.0316	0.0055	0.0208	0.0424	32.81	<.0001
CLINIC	1	0.5806	0.1716	0.2443	0.9169	11.45	0.0007
Scale	1	0.5868	0.0403	0.5129	0.6712		

この出力から，CLINIC = 1に対するCLINIC = 2の加速係数はexp（0.5806）= 1.79と推定されます．対数ロジスティックモデルに加速時間仮定が成立するならば, 生存関数に関して比例オッズ仮定も成立します（比例ハザード仮定は成立しない）．比例オッズ仮定は，対数生存オッズ（KM推定値を使用）と対数生存時間のプロットから評価できます. 共変量パターンに関してプロットが直線であれば，対数ロジスティック分布は適合しています．直線かつ平行であれば，比例オッズ仮定に加えて加速時間仮定も成り立ちます.

PROC LIFETESTを用いて，KM生存推定値を含むSASデータセットを作成します（Appendixのセクション1を参照）．このデータセットを基に，対数生存オッズ推定値と対数生存時間が導出します．さらにPROC GPLOTを使用して，対数生存オッズと対数生存時間をプロットします.

比例オッズ仮定を別の面，「ロジスティック回帰で推定するオッズ比はフォローアップの長さに影響されない」, から考えてみます．例えば，もし比例オッズ仮定が成立するならば，たとえフォローアップ期間を3年から5年に延長したとしても, 2つの共変量パターンを比較するオッズ比は変わらないはずです. 比例オッズ仮定が成立しないならば, オッズ比はフォローアップの長さによって違う値となります.

加速モデルは，生存時間に対しては乗法モデルですが，対数時間に対しては加法モデルです．前例では，メディアン生存時間はCLINIC = 1に対してCLINIC = 2は1.79倍長いと推定されました．前例では，生存時間は対数ロジスティック分布に従う，それはつまり，対数生存時間はロジスティック分布に従うと仮定しました．

SASでは加法failure時間モデルも実行可能です（第7章の「その他のパラメトリックモデル」を参照）．PROC LIFEREGのMODELステートメントのNOLOGオプションは，デフォルトの対数リンク関数を抑制します．つまり，log（時間）ではなく時間が回帰パラメータの線形関数としてモデル化されます．以下のコードは，時間がロジスティック（対数ロジスティックではなく）分布に従う加法failure時間モデルを指定します．

```
PROC LIFEREG DATA=REF.ADDICTS;
MODEL SURVT*STATUS(0)=PRISON DOSE CLINIC/DIST=LLOGISTIC NOLOG;
RUN;
```

DIST = LLOGISTICオプションは生存時間が対数ロジスティック分布に従うように指定しているようにみえますが，NOLOGオプションにより，実際は生存時間がロジスティック分布に従うように指定されています．（StataでのNOLOGオプションは，**streg**コマンドに関するものではなく，単に反復記録の出力を抑制するものです）．加法failure時間モデルの出力は以下の通りです．

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-358.482	114.0161	-581.949	-135.014	9.89	0.0017
PRISON	1	-89.7816	42.9645	-173.990	-5.5727	4.37	0.0366
DOSE	1	10.3893	1.6244	7.2055	13.5731	40.91	<.0001
CLINIC	1	214.2525	53.1204	110.1385	318.3665	16.27	<.0001
Scale	1	172.4039	11.3817	151.4792	196.2191		

CLINICのパラメータ推定値は214.2525です．この推定値の解釈は，CLINIC = 2のメディアン生存時間（または$S(t)$の任意の固定値までの時間）がCLINIC = 1よりも214日長いと推定されるというものです．言い換えれば，CLINIC = 1のメディアン生存時間推定値に214日を足すと，CLINIC = 2のメディアン生存時間推定値が得られます．前述の加速モデルでのCLINIC = 1のメディアン生存時間推定値に1.79を掛けるとCLINIC = 2のメディアン生存時間推定値が得られるのとは対照的です．加法モデルは生存時間がずれるとみなすのに対して，加速モデルは生存時間の縮尺が変わるとみなすことができます．

生存時間がロジスティック分布に従い，加法failure時間仮定が成り立つならば，比例オッズ仮定も成立します．ロジスティック仮定は，対数生存オッズ（KM推定値を使用）と時間（対数ロジスティック仮定の評価の対数時間とは違い）をプロットすることにより評価できます．それぞれの共変量パターンに関するプロットが直線であれば，ロジスティック分布が適合しています．直線かつ平行であれば，比例オッズ仮定と加法failure時間仮定が成立します．

PROC LIFEREGでサポートする他の分布には，一般化ガンマ（DIST = GAMMA）分布と対数正規（DIST = LNORMAL）分布があります．MODELステートメントでNOLOGオプションをDIST = LNORMALオプションとともに指定した場合，生存時間が正規分布に従うと仮定されます．

8. 再発イベントのモデル構築

再発イベントのモデル構築については，Appendixの冒頭で紹介した「膀胱がん」データセット（**bladder.sas7bdat**）を用いて説明します．再発イベントは，イベントを複数経験した被験者に対応する複数のオブザベーションを持つデータとして特徴づけられます．「膀胱がん」データセットのデータレイアウトは，区間ごとにオブザベーションを持つCPアプローチに適しています（第8章を参照）．以下のコードは，被験者4名の情報からなる12～20番目のオブザベーションを出力します．コードは以下の通りです．

```
PROC PRINT DATA=REF.BLADDER (FIRSTOBS= 12 OBS=20);
RUN;
```

出力は以下の通りです.

OBS	ID	EVENT	INTERVAL	START	STOP	TX	NUM	SIZE
12	10	1	1	0	12	0	1	1
13	10	1	2	12	16	0	1	1
14	10	0	3	16	18	0	1	1
15	11	0	1	0	23	0	3	3
16	12	1	1	0	10	0	1	3
17	12	1	2	10	15	0	1	3
18	12	0	3	15	23	0	1	3
19	13	1	1	0	3	0	1	1
20	13	1	2	3	16	0	1	1

ID = 10に関しては3オブザベーション，ID = 11に関しては1オブザベーション，ID = 12に関しては3オブザベーション，ID = 13に関しては2オブザベーションがあります．変数STARTおよびSTOPは，オブザベーションに対応するリスク期間の時間を指定します．変数EVENTは，イベント(code = 1)が発生したかどうかを示します．最初の3オブザベーションは，ID = 10の被験者に12ヵ月にイベントがあり，16ヵ月にまた別のイベントがあり，18ヵ月に打ち切りとなったことを示しています．

PROC PHREGはCPデータレイアウトの生存データにも用いることができます．以下のコードは，治療(TX)，最初の腫瘍数(NUM)，および最初の腫瘍サイズ(SIZE)の3つの予測因子を含むモデルを実行します．

```
PROC PHREG DATA=REF.BLADDER COVS(AGGREGATE);
MODEL (START,STOP)*EVENT(0)=TX NUM SIZE;
ID ID;
RUN;
```

MODELステートメントのコード(START,STOP)*EVENT(0)は，各オブザベーションの時間区間を変数STARTとSTOPで定義し，EVENT = 0が打ち切りオブザベーションであることを示します．IDステートメントは，対象を定義する変数がIDであることを示します．PROC PHREGステートメントのCOVS(AGGREGATE)オプションは，パラメータ推定値のロバスト標準誤差を要求します．PROC PHREGによる出力は以下の通りです．

PHREG プロシジャ

モデルの情報

データセット	REF.BLADDER
従属変数	START
従属変数	STOP
打ち切り変数	EVENT
打ち切り値の数	0
タイデータの処理	BRESLOW

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	標準誤差比	カイ2乗	Pr > ChiSq	ハザード比
TX	1	-0.40710	0.24183	1.209	2.8338	0.0923	0.666
NUM	1	0.16065	0.05689	1.185	7.9735	0.0047	1.174
SIZE	1	-0.04009	0.07222	1.028	0.3081	0.5788	0.961

係数推定値はロバスト標準誤差とともに出力されます．“StdErr Ratio”列には，非ロバスト標準誤差に対するロバスト標準誤差の比が示されます．例えば，TX係数の標準誤差0.24183は，ロバスト標準誤差を要求しなかった場合（つまりCOVS（AGGREGATE）オプションを省いた場合）の標準誤差の1.209倍です．ロバスト標準誤差の推定値は，StataやRとはわずかに異なります．

この形式のデータを使用し，変数INTERVALを層化変数とした層化Coxモデルを実行することもできます．層化変数は，被験者が1番目，2番目，3番目，4番目のイベントに関してat riskであるかを示します．これは8章で層化CPアプローチと紹介したものであり，再発イベントの発生順序を区別したい場合に使用します．層化Cox用のコードは以下の通りです．

```
PROC PHREG DATA=REF. BLADDER COVS(AGGREGATE) ;
MODEL (START,STOP)*EVENT(0)=TX NUM SIZE;
ID ID;
STRATA INTERVAL;
RUN;
```

前のモデルに追加するコードは，変数INTERVALが層化変数であることを示すSTRATAステートメントだけです．パラメータ推定値を含む出力は以下の通りです．

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	標準誤差比	カイ2乗	Pr > ChiSq	ハザード比
TX	1	-0.33430	0.19706	0.912	2.8777	0.0898	0.716
NUM	1	0.11565	0.04991	0.930	5.3690	0.0205	1.123
SIZE	1	-0.00805	0.06012	0.827	0.0179	0.8935	0.992

1番目，2番目，3番目，4番目のイベントに対する治療の効果の違いを確認するために，治療変数（TX）と層化変数との交互作用項を作成することもできます．

Gap Timeと呼ばれるもう1つの層化アプローチは，層化CPアプローチを少し変えたものです．その違いは再発イベントの時間区間の定義方法にあります．被験者の最初のイベントに対するat riskの時間区間には違いはありません．しかしながら次のイベントからは，Gap Timeアプローチでは，at riskの開始時間が0にリセットされます．以下のコードは，Gap Timeアプローチに適したデータを作成します．

```
DATA BLADDER2;
SET REF.BLADDER;
START2=0;
STOP2=STOP-START;
RUN;
```

新しいデータセット（BLADDER2）は，データをREF.BLADDERからコピーし，2つの新しい時間区間変数START2およびSTOP2を作成します．START2は常に0に設定され，STOP2は区間の長さ（STOP-START）です．以下のコードは，新たに作成したこれらの変数を使用して，Gap TimeモデルをPROC PHREGで実行します．

```
PROC PHREG DATA=BLADDER2 COVS(AGGREGATE);
MODEL (START2,STOP2)*EVENT(0)=TX NUM SIZE;
ID ID;
STRATA INTERVAL;
RUN;
```

出力は以下の通りです.

最尤推定量の分析

パラメータ	自由度	パラメータ推定値	標準誤差	標準誤差比	カイ2乗	Pr > ChiSq	ハザード比
TX	1	-0.26952	0.20808	1.002	1.6778	0.1952	0.764
NUM	1	0.15353	0.04889	0.938	9.8620	0.0017	1.166
SIZE	1	0.00684	0.06222	0.889	0.0121	0.9125	1.007

Gap Timeアプローチによる結果は，層化CPアプローチの結果とほんのわずか異なります.

被験者1名につき複数のオブザベーションを持つCPデータレイアウトは，再発イベントデータのみならず，被験者のイベントが1つの従来型の生存時間解析にも使用することができます．4オブザベーシの被験者は, 最初の3つのオブザベーションは打ち切りで, 4番目のオブザベーションの区間でイベントを経験すると考えることもできます．このデータレイアウトは，時間依存性曝露(つまり，時間区間により値が異なる曝露)を表すのに特に適しています.

C. SPSS

SPSSの解析は，SPSSデータセットを用いて適切なSPSSプロシジャにより行います．大半のユーザーは，一連のメニューとダイアログボックスをマウスでクリックすることでプロシジャを選択します．この手順で生成されるコード，あるいはコマンドシンタックスは表示や編集が可能です.

薬物常用者データセットの解析例を用いて，これらのプロシジャについて説明します．薬物常用者データセットは，1991年のCaplehornらによるオーストラリア試験のもので，238名のヘロイン常用者の情報が含まれます．この試験は，被験者のメタドン治療期間を2つのメタドン治療施設で比較するものです. 2施設には被験者への院内方針に違いがありました. 被験者の生存時間は，被験者が施設での治療から脱落するかまたは観察が打ち切られるまでの期間(日)と定められました．変数はAppendixの冒頭に定義した通りです.

SPSSを起動後，データセット**addicts.sav**を開きます．データが画面に表示されます．これで作業用データセットになりました．結果変数（SURVT）の要約統計量を求めるには，ドロップダウンメニューから分析→記述統計→記述統計をクリックすると，解析変数を指定するダイアログボックスが表示されます．変数リストからSURVTを選択して変数ボックスに入れます．[OK]をクリックすると出力が表示されます．あるいは，[OK]の代わりに[貼り付け]をクリックすれば，対応するSPSSシンタックスを得ることもできます．このシンタックスは実行（[実行]ボタンをクリック），編集，または別のセッション用に保存できます．作成されるシンタックスは以下の通りです（出力は省略）．

```
DESCRIPTIVES
  VARIABLES=survt
  /STATISTICS=MEAN STDDEV MIN MAX.
```

SPSSの一部の解析は，ポイント&クリック方式では実行できず，シンタックスの実行のみで可能なものがあります（例えば，時間依存性共変量が2つある拡張Coxモデル）．ポイント&クリック方式で実行するたびに，対応するシンタックスが示されます．

より詳細なCLINICごとの生存時間の要約統計量を求めるためには，ドロップダウンメニューから，[解析]→[記述統計]→[探索的]をクリックします．変数リストからSURVTを選択して[従属変数]ボックスに移動してから，CLINICを選択して[因子]ボックスに入れます．[OK]をクリックすれば出力が表示されます．[貼り付け]を（[OK]の代わりに）クリックすると以下のシンタックスが作成されます（出力は省略）．

```
EXAMINE
  VARIABLES=survt BY clinic
  /PLOT BOXPLOT STEMLEAF
  /COMPARE GROUP
  /STATISTICS DESCRIPTIVES
  /CINTERVAL 95
  /MISSING LISTWISE
  /NOTOTAL.
```

生存時間解析をSPSSで行うには，[分析]→[生存分析]を選択します．4選択のボックス，つまり[生命表]，[Kaplan-Meier]，[Cox回帰]，[時間依存のCox回帰]が示されます．SPSSの主な生存時間解析用プロシジャはKMとCOXREGです．

SPSSでは以下の生存時間解析を行います.

1. 生存関数(未調整)の推定および層間での比較
2. Kaplan-Meier対数(－対数)生存曲線を用いた比例ハザード性の検討
3. Cox比例ハザードモデルの実行
4. 層化Coxモデルの実行とCox調整対数(－対数)曲線の作成
5. 統計的検定による比例ハザード仮定の評価
6. 拡張Coxモデルの実行

SPSS(バージョンPASW 18)には,パラメトリック生存モデル,frailtyモデル,再発イベント用のCPデータレイアウトに対応したモデルを実行するコマンドはありません.

1. 生存関数(未調整)の推定および層間での比較

Kaplan-Meier生存推定値を得るには,[分析]→[生存分析]→[Kaplan-Meier]を選択します.変数リストからSURVTを選択して[生存変数]ボックスに移動し,変数STATUSを選択して[状態変数]ボックスに移動します.[状態変数]ボックスに表示されているステータス(?)の意味は,イベントを示す値を入力する必要があるということです.変数STATUSはイベント=1,打ち切り=0なので,[事象の定義]ボタンをクリックし,値1を入れます.[続行]→[OK]をクリックすれば出力が得られます.[OK]ではなく[貼り付け]をクリックすると以下のシンタックスを得ます(出力省略).

```
KM
    survt /STATUS=status(1)
    /PRINT TABLE MEAN.
```

このKM推定の出力は非常に長いです.出力を編集したいときには,出力内を右クリックし,[内容編集]を選択します.編集出力を別ウィンドウで開くかどうかで,[ビューア]または[別ウィンドウ]のどちらかを選択します.

CLINIC別のKM生存推定値とプロット，ログランク検定やその他の検定統計量を得るには，[分析]→[生存分析]→[Kaplan-Meier]を選択し，SURVTをtime-to-event変数として，またSTAUSをstatus変数として前例のように選択します．CLINICを[因子]ボックスに選択し，[因子の比較]ボタンをクリックします．3つの検定統計量がCLINIC間の生存関数の違いを検定するために準備されています．比較のために3つとも（ログランク，Breslow，Tarone-Ware）選択し，[続行]をクリックします．[オプション]ボタンをクリックしてプロットを要求します．4種類のプロットが作成可能です（残念ながら，その中に対数-対数生存時間プロットはありません）．[続行]→[OK]をクリックすればCLINIC別のKMプロットが得られます．

シンタックスは以下の通りです．

```
KM
  survt BY clinic /STATUS=status(1)
  /PRINT TABLE MEAN
  /PLOT SURVIVAL
  /TEST LOGRANK BRESLOW TARONE
  /COMPARE OVERALL POOLED.
```

イベントまたは打ち切りの最初の5時点についての，CLINIC = 1およびCLINIC = 2のKM推定値，ならびにログランク検定，Breslow検定，Tarone-Ware検定の出力は以下の通りです．

```
Survival Analysis for SURVT    Survival time (days)
```

CLINIC = 1

Survival Table

clinic		Time	Status	Cumulative Proportion Surviving at the Time		N of Cumulative Events	N of Remaining Cases
				Estimate	Std. Error		
1.00	1	7.000	endpoint	.994	.006	1	161
	2	17.000	endpoint	.988	.009	2	160
	3	19.000	endpoint	.981	.011	3	159
	4	28.000	censored	.	.	3	158
	5	28.000	censored	.	.	3	157
	.						
	.						
	.						

```
Factor CLINIC = 2.00
```

2.00	1	13.000	endpoint	.986	.013	1	73
	2	26.000	endpoint	.973	.019	2	72
	3	35.000	endpoint	.959	.023	3	71
	4	41.000	endpoint	.946	.026	4	70
	5	53.000	censored	.	.	4	69

```
      .
      .
Test Statistics for Equality of Survival Distributions for CLINIC

                Statistic        df       Significance
```

SPSSのBreslow検定統計量とは，Stata（SASも）でいうWilcoxon検定統計量のことです．

生命表推定は，[分析]→[生存分析]→[生命表]を選択します．time-to-event変数とstatus変数は前述のKM推定と同様に定義しますが，生命表の場合は[時間間隔]ボックスが表示されます．このボックスで，生命表解析に用いる時間区間を定義します．例えば，0～1,000/100と指定すると，同じ長さの時間区間が10個定義されます．生命表プロットも，上記のKMプロットの場合と同様に作成します．

2. Kaplan-Meier対数(−対数)生存曲線を用いた比例ハザード性の検討

SPSSでは，KMコマンドをポイント&クリックで直接的に未調整KM対数(−対数)曲線を作成することはできません．SPSSは，層化Coxモデルを実行することにより，調整対数(−対数)曲線を作成します(層化Coxのセクションで後述)．SPSSで共変量をモデルに含まない層化Coxを実行すれば，作成された対数(−対数)曲線は未調整KM対数(−対数)曲線になります．しかしこのセクションでは，作業用データセットに新しい変数を定義して未調整対数(−対数)KMプロットを作成する方法について説明します．

まず，KM生存推定値を含む変数を作成します．次に，生存推定値の対数(－対数)をとった別の新しい変数を作成します．最後に，対数(－対数)生存推定値と生存時間をプロットして，CLINIC = 1の曲線とCLINIC = 2の曲線が平行であるかどうか確認します．これらの各ステップを実行するには，ポイント&クリックするか，コードを直接入力します．

生存推定値を含む変数を作成するには，[分析]→[生存分析]→[Kaplan-Meier]を選択し，SURVTをtime-to-event変数として，STAUSをstatus変数として，CLINICを因子変数として前述のように選択します．そして次に，[保存]ボタンをクリックします．[Kaplan-Meier の新変数の保存]ダイアログボックスが表示されます．[累積生存確率]をクリックし，[続行]，そして[貼り付け]をクリックします．以下のコードが作成されます．

```
KM
  survt BY clinic /STATUS=status(1)
  /PRINT TABLE MEAN
  /SAVE SURVIVAL.
```

このコードを実行することにより，KM推定値を含む新しい変数**SUR_1**が作成されます．**SUR_1**をlog(－log)変換した新しい変数**lls**を作成するには，以下のコードを実行します．

```
COMPUTE lls = LN(-LN (SUR_1)).
  EXECUTE.
```

上記のコードは，[変換]→[変数を計算]を選択し，ダイアログボックスで新しい変数を定義しても作成されます．**lls**の生存時間に対するプロットを作成するには，以下のコードを実行します．

```
GRAPH
  /SCATTERPLOT(BIVAR)=survt WITH lls BY clinic
  /MISSING=LISTWISE.
```

コードの最後の部分は別の方法でも実行できます．[グラフ]→[レガシーダイアログ]→[散布図/ドット]を選択し，[散布図/ドット]ダイアログボックスで[単純な散布]，[定義]の順にクリックします．LLSをy軸に，SURVTをX軸に，[マーカー設定]ボックスでCLINICを選択します．[貼り付け]をクリックしてコードを作成するか，[OK]をクリックしてプログラムを実行します．LLSとlog(SURVT)のプロットも同様に作成できます．曲線が平行であれば，CLINICについての比例ハザード仮定を支持するものとなります．

3. Cox比例ハザードモデルの実行

Cox比例ハザードモデルを実行するには，[分析]→[生存分析]→[Cox回帰]を選択します．変数リストからSURVTを[時間]ボックスに選択し，STATUSを[状態変数]ボックスに選択します．[状態変数]ボックスがステータス(?)となっているのは，イベント値を入力する必要があるという意味です．変数STATUSはイベント＝1，打ち切り＝0なので，[事象の定義]ボタンをクリックし，値1を入れます．[続行]をクリックして，変数リストからPRISON，DOSE，CLINICを[共変量]ボックスに選択します．[作図]クリックするか，いくつかのオプション(例えばexp(β)の95%信頼区間など)を指定するために[オプション]をクリックします．[OK]をクリックすれば出力が，[貼り付け]をクリックすればコードが表示されます．コードは以下の通りです．

```
COXREG
  survt /STATUS=status(1)
  /METHOD=ENTER prison dose clinic
  /CRITERIA=PIN(.05) POUT(.10) ITERATE(20).
```

Coxモデルでは，3つすべての共変量について比例ハザード仮定が成立すると仮定しています(出力は以下の通り)．

Omnibus Tests of Model Coefficients[a,b]

−2Log Likelihood	Overall (score)			Change From Previous Step			Change From Previous Block		
	Chi-square	df	Sig.	Chi-square	df	Sig.	Chi-square	df	Sig.
1346.805	56.273	3	.000	64.519	3	.000	64.519	3	.000

[a] Beginning Block Number 0, initial Log Likelihood function: −2 Log likelihood: 1411.324
[b] Beginning Block Number 1. Method = Enter

Variables in the Equation

	B	SE	Wald	df	Sig.	Exp(B)
PRISON	.327	.167	3.813	1	.051	1.386
DOSE	−.035	.006	30.785	1	.000	.965
CLINIC	−1.009	.215	22.045	1	.000	.365

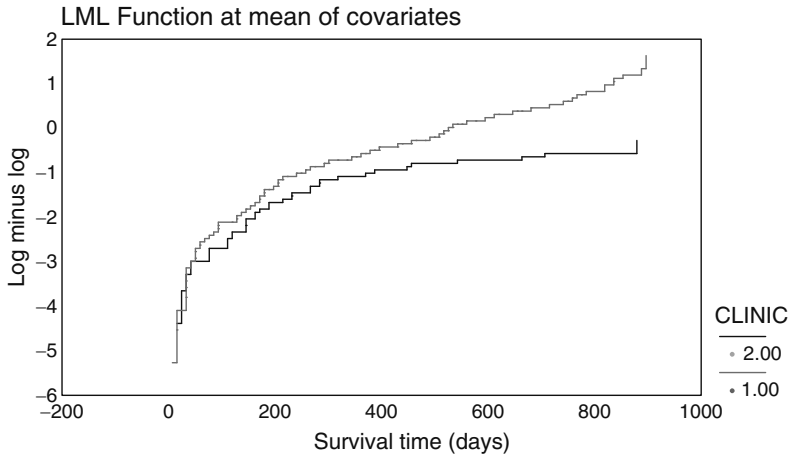
4. 層化Coxモデルの実行とCox調整対数(–対数)曲線の作成

層化Coxモデルを実行するには，[分析]→[生存分析]→[Cox回帰]を選択します．変数リストからSURVTを[時間]ボックスに選択し，STATUSを[状態変数]ボックスに選択し，イベント値を1と定義します．変数PRISONとDOSEを[共変量]ボックスに選択し，変数CLINICを[ストラータ]ボックスに選択します．CLINICで層化したCoxモデルになります．[作図]ボタンをクリックし，作図の種類で[ログマイナスログ]をチェックし，[続行]をクリックします．[OK]をクリックすれば出力が，[貼り付け]をクリックすればコードが表示されます．コードは以下の通りです．

```
COXREG
    survt /STATUS=status(1)
    /STRATA=clinic
    /METHOD=ENTER prison dose
    /PLOT LML
    /CRITERIA=PIN(.05) POUT(.10) ITERATE(20).
```

パラメータ推定値を含む出力と調整対数(－対数)プロットは以下の通りです．

	B	SE	Wald	df	Sig.	Exp(B)
PRISON	.389	.169	5.298	1	.021	1.475
DOSE	−.035	.006	29.552	1	.000	.965



PRISONとDOSEのパラメータ推定値はあるのにCLINICがないのは，CLINICが層化変数だからです．Cox調整対数(－対数)プロットはPRISONとDOSEの平均値を用いています．プロットはCLINICに関する比例ハザード性の評価に用います．

調整対数(－対数)プロットにDOSEの平均値を用いる代わりに，DOSE = 70を用いる場合は以下のようにします．[作図]ボタンをクリックし，作図の種類で[ログマイナスログ]をチェックするところまでは前述と同じです．ここから，[Cox回帰分析　作図]ウィンドウで[DOSE(平均)]をクリックします．[値の変更]の下の[値]をクリックします．値を70と入力し，[変更]ボタンをクリックします．これで，ウィンドウ内の変数は[DOSE(平均)]ではなく[DOSE(70)]となります．[続行]，さらに[OK]をクリックすればグラフが作図されます．

5.　統計的検定による比例ハザード仮定の評価

SPSSでは，Schoenfeld残差を用いた比例ハザード仮定に関する統計的検定を簡単に行うことはできません．ただし，以下の手順でプログラミングすることは可能です．

1. Cox比例ハザードモデルを実行して，すべての共変量についてのSchoenfeld残差を求めて，新しい変数として作業用データセットに保存する．
2. 打ち切りオブザベーションを削除する．
3. 生存時間の順位変数を作成する．例えば，4番目にイベントを経験した被験者の変数値は4となります．
4. Schoenfeld残差と生存順位との相関解析を実行する．
5. 生存順位と特定の共変量のSchoenfeld残差との相関を検定すると，p値は，そのまま比例ハザード仮定の検定p値になります．帰無仮説は，「比例ハザード仮定が成立する」です．

まず，CLINIC，PRISON，DOSEを含むCox比例ハザードモデルを実行します．このモデルの実行前に[保存]ボタンをクリックします．[Cox回帰：モデル変数を保存]ダイアログボックスが表示されます．[偏残差]をチェックし[続行]をクリックします．これは，3つの新しい変数**PR1_1**，**PR2_1**，**PR3_1**を作業用データセットに作成するものです．これらの変数はそれぞれ，CLINIC，PRISON，DOSEの偏残差(Schoenfeld残差)です．[OK]をクリックしてモデルを実行(または[貼り付け]をクリックしてコードを表示)します．

次に，打ち切りオブザベーションをすべて削除します(つまり，STATUS = 1のオブザベーションのみを残します)．これを行うには，[データ]→[ケースの選択]を選択します．[If 条件が満たされるケース]にチェックマークを付け，[If]をクリックします．ダイアログボックスにstatus = 1と入力し，[続行]をクリックします．[選択されなかったケース]ボックスで[削除]にチェックし[OK]をクリックします．イベントのあるオブザベーションのみがデータセットに残ります(薬物常用者データでの解析を続けたい場合は，打ち切りオブザベーションを含む薬物常用者データセットに戻ることが必要です)．

生存時間の順位変数を以下の手順で作成します．[変換]→[ケースのランク付け]を選択します．SURVTを選択して[変数]ボックスに入れます．[ケースのランク付け:タイプ]をクリックし，[順位]にチェックマークを付けて，[続行]をクリックします．[同順位]をクリックし，[平均]をチェックし[続行]をクリックします．[OK]をクリックすると，生存時間の順位変数(Rsurvt)が作成されます．

最後に，生存時間順位とSchoenfeld残差との相関(p値も)を解析します．[分析]→[相関]→[二変量]を選択します．生存時間順位変数と3つの偏残差変数を変数ボックスに移動します．[Pearson](Pearson相関係数)をチェックし，両側検定に対応する[両側]をチェックしたら，[OK]をクリックすると出力が表示されます．この手順で生成されるコードは以下の通りです．

```
COXREG
    survt /STATUS=status(1)
    /METHOD=ENTER clinic prison dose
    /SAVE= PRESID
    /CRITERIA=PIN(.05) POUT(.10) ITERATE(20) .

FILTER OFF.
    USE ALL.
    SELECT IF(status=1).
    EXECUTE

RANK
    VARIABLES=survt (A) /RANK /PRINT=YES
    /TIES=MEAN.

CORRELATIONS
    /VARIABLES=Rsurvt PR1_1 PR2_1 PR3_1
    /PRINT=TWOTAIL NOSIG
    /MISSING=PAIRWISE.
```

相関係数を含む出力は以下の通りです.

Correlations

		RANK of SURVT	Partial residual for CLINIC	Partial residual for PRISON	Partial residual for DOSE
RANK of SURVT	Pearson Correlation	1	-.262**	-.080	.077
	Sig. (2-tailed)	.	.001	.332	.347
	N	150	150	150	150
Partial residual for CLINIC	Pearson Correlation	–.262**	1	.010	.023
	Sig. (2-tailed)	.001	.	.904	.776
	N	150	150	150	150
Partial residual for PRISON	Pearson Correlation	–.080	.010	1	.171*
	Sig. (2-tailed)	.332	.904	.	.037
	N	150	150	150	150
Partial residual for DOSE	Pearson Correlation	.077	.023	.171*	1
	Sig. (2-tailed)	.347	.776	.037	.
	N	150	150	150	150

相関のp値は比例ハザード検定のp値です. この出力の「RANK of SURVT」行見出しの「Sig. (2-tailed)」行を見てください. CLINIC (p値 = 0.001)に関する帰無仮説は棄却されますが, PRISON (p値 = 0.332)と DOSE (p値 = 0.347)は棄却されません.

6. 拡張Coxモデルの実行

1つの時間依存性共変量を含む拡張Coxモデルだけが, ポイント&クリック操作により実行できます. DOSEに生存時間の対数を掛けた間依存性共変量を考えます. DOSEの任意の2レベルを比較するハザード比が時間に対して単調増加(あるいは減少)する場合は, この積項が適切となり得ます. [分析]→[生存分析]→[時間依存のCox回帰]を選択します. [時間依存の共変量の計算]ダイアログボックスが表示されます. このダイアログボックスで時間依存性共変量(T_COV_)を定義します. 変数T_が変数リストに含まれています. これは時間によって変動する生存を説明する変数です(一方, SURVTは個人の固定したイベント時間です). T_COV_を対数(T_)× DOSEと定義するには, 式LN(T_)*DOSEをダイアログボックスに入力し, [モデル]ボタンをクリックします. 次に, 共変量PRISON, CLINIC, DOSE, T_COV_を含むCoxモデルを実行します. 生成されるコードは以下です.

```
TIME PROGRAM.
  COMPUTE T_COV_ = LN(T_) * dose.

COXREG
  survt /STATUS=status(1)
  /METHOD=ENTER prison clinic dose T_COV_
  /CRITERIA=PIN(.05) POUT(.10) ITERATE(20).
```

パラメータ推定値を含む出力は以下の通りです.

Variables in the Equation

	B	SE	Wald	df	Sig.	Exp(B)
PRISON	.340	.167	4.134	1	.042	1.406
CLINIC	−1.019	.215	22.369	1	.000	.361
DOSE	−.082	.036	5.247	1	.022	.921
T_COV_	.009	.006	1.765	1	.184	1.009

変数T_COV_はモデルに含まれる時間依存性変数を表します. この例ではDOSEに対数生存時間を掛けたものです.

CLINICに関するHeavisideの階段関数も同様に簡単に作成できます. 365日以上ではCLINIC値に等しく，365日未満では0となる時間依存性変数を定義することを考えます. [分析]→[生存分析]→[時間依存のCox回帰]を選択します. T_COVを(T_ × 365) × clinicと定義します. [モデル]ボタンをクリックした後，PRISON，DOSE，CLINIC，T_COV_を含むCoxモデルを実行します. 生成されるコードは以下です.

```
TIME PROGRAM.
  COMPUTE T_COV_ = (T_ > = 365)* clinic.

COXREG
  survt /STATUS=status(1)
  /METHOD=ENTER prison clinic dose T_COV_
  /CRITERIA=PIN(.05) POUT(.10) ITERATE(20).
```

SPSSにおける表記(T_ × 365)は，生存時間が365日以上であれば値1，その他の場合は値0を返す関数です.

出力は以下の通りです.

Variables in the Equation

	B	SE	Wald	df	Sig.	Exp(B)
PRISON	.378	.168	5.030	1	.025	1.459
CLINIC	−.460	.255	3.241	1	.072	.632
DOSE	−.036	.006	30.450	1	.000	.965
T_COV_	−1.369	.461	8.799	1	.003	.254

変数CLINICがこのモデルに含まれ，時間依存性Heavisideの階段関数T_COV_が365日まではハザード比推定に寄与しないことを考慮すれば，100日時点のCLINICのハザード比推定値は，exp（− 0.460） = 0.632となり，400日時点のCLINICのハザード比推定値はexp((− 0.460)+(−1.369))=0.161となります.

CLINIC に関する2つのHeaviside の階段関数を定義してCLINICをモデルに含めないという方法もあります．これは前述の1つの階段関数モデルと本質的には同じものです．ただし, CLINICに関する2つのハザード比推定値（365日未満と365日以上における）を計算することに関しては，2つの階段関数を用いる方が簡単です．残念ながら，SPSSのポイント＆クリック操作では，時間依存性変数は1つ（T_COV_）しか扱えません．しかしながら，1つのHeavisideの階段関数におけるコードに若干の手を加えれば，2つのHeavisideの階段関数に対応するコードが作成できます．以下のコードは，2つの階段関数（HV1とHV2）を作成し，PRISON，DOSE，HV1，HV2を含むモデルを実行します.

```
TIME PROGRAM.
COMPUTE hv1= (T_ < 365)* clinic.
COMPUTE hv2= (T_ >= 365)* clinic.

COXREG
    survt /STATUS=status(1)
    /METHOD=ENTER prison dose hv1 hv2
    /CRITERIA=PIN(.05) POUT(.10) ITERATE(20).
```

出力は以下の通りです.

Variables in the Equation

	B	SE	Wald	df	Sig.	Exp(B)
PRISON	.378	.168	5.030	1	.025	1.459
DOSE	−.036	.006	30.450	1	.000	.965
HV1	−.460	.255	3.241	1	.072	.632
HV2	−1.828	.386	22.439	1	.000	.161

HV1とHV2に関するパラメータ推定値から, 365日未満と365日以上におけるCLINIC = 2 vs. CLINIC = 1のハザード比推定値を直接求めることができます. 100日時点のCLINICのハザード比推定値はexp(− 0.460) = 0.632であり，400日時点のCLINICのハザード比推定値はexpexp(− 1.828) = 0.161です．これらの結果は，前述の1Heaviside の階段関数モデルの結果と一致しています.

D. Rソフトウェア

Rは，Comprehensive R Archive Network（CRAN）のWebサイト（http：//www.r-project.org/）から無料でダウンロードできるソフトウェアです. Rによる解析は，Rのデータ（Rのオブジェクトとして保存される）に関数を適用して実行します．Rの関数はパッケージに含まれます．パッケージがロードされているときだけ, そのコンテンツが使用可能です. 基本パッケージはRをダウンロードするときにインストールされます．基本パッケージ以外のパッケージは個別にインストールする必要があります.

Rを起動すると，プロンプト（>）が表示されます．1+1と入力してEnterキーを押します．次の行に答え2が返されます．あるいは，スクリプトにコマンドを入力することでも実行できます．[File]→[New script]をクリックすると，新しいスクリプトウィンドウが開きます．ここにコマンドを入力して，一連のコードを選択し，[Edit]→[Run line or selection]をクリックすれば，選択した部分を一度に実行できます．スクリプトウィンドウ内でのプログラミングは，一度に一行ずつコードを処理するのではなくブロック単位で処理をする点で，SASのプログラムエディタやStataのDo-file Editorと類似の機能です.

どのパッケージがインストールされているかを確認するには，library()と入力し，Enterキーを押します．生存時間解析の実行に必要な関数の多くは，survivalパッケージに含まれ（基本パッケージには含まれない），これをインストールする必要があります.

Survivalパッケージをインストールするには，[Packages]→[Install package(s)]をクリックします．[CRAN mirror]ウィンドウに表示される国別サイトから1つ(例えばJapan(Tokyo))をクリックします．表れた[Packages]ウィンドウを下方にスクロールし，リストからsurvivalを選択し，[OK]をクリックします．これでsurvivalパッケージ(とその多くの生存関数)がインストールされました．**library(survival)**と入力しEnterキーを押すと，survivalパッケージが使用可能な状態になります．確認のため，kidneyと入力しEnterキーを押してみてください．Kidneyという名前のデータセット(survivalパッケージの一部)が画面に出力されます．Survivalパッケージは，一度インストールしてしまえばセッションのたびに再インストールする必要はありません．ただし，このパッケージに含まれている生存関数を実行するためには，セッションのたびに**library(survival)**を入力する必要があります．

Rでの生存時間解析について述べる前に，Rでのデータ保存方法をいくつか簡単に説明しておきます．特に，データ保存の4クラス，すなわちベクトル(**vectors**)，行列(**matrices**)，**dataframe**，**list**について説明します．以下のコードを入力してEnterキーを押すと，5つの要素を持つ数値ベクトルが作成されます．

```
c(1,7,12,6,3)
```

c関数は，その引数をベクトル形式の組み合わせにします．このベクトルを**x1**という名前(識別名)のオブジェクトとして保存できます．

```
x1=c(1,7,12,6,3)
```

x1と入力してEnterキーを押すと，ベクトル**x1**が出力表示されます．コードと出力は以下の通りです．

```
x1
1 7 12 6 3
```

ベクトルx1の要素は，**x1**の後ろの角括弧**[ ]**に位置を指定することにより特定できます．例えば，**x1[2]**は**x1**の2番目の要素を特定します．コード**x1[1:3]**は**x1**の最初の3つの要素を特定し，コード**x1[x1>6]**は6よりも大きい**x1**の要素のを特定します．これら3つの例のコードおよび出力は以下の通りです．

```
x1[2]
7
x1[1:3]
1 7 12
x1[x1>6]
7 12
```

x1[1:3]に用いられた演算子「:」は，1ずつ増加する整数連続値を作成します．次に，4つのベクトル**x2**，**x3**，**x4**，**x5**を作成します．

```
x2=2*(1:5)
x3=2*x1 + x2
x4=x1>6
x5=c("blue","green","red","green","purple")
```

コード**x2=2*(1:5)**は，**x2**という名のベクトル2，4，6，8，10を作成します．ベクトル**x3**は**x1**と**x2**の算術演算（2 × **x1** + **x2**）の結果です．ベクトル**x1**，**x2**，**x3**はいずれも数値ベクトルです．**mode**関数を**x1**に適用する（**mode(x1)**と入力してEnterキーを押す）と，“numeric”という単語が出力されます．ベクトル**x4**は論理ベクトルです．コード**mode(x4)**を実行すると，**mode**関数は“logical”という単語を返します．論理ベクトルの要素は“TRUE”または“FALSE”です．コード**x4**を入力すると出力は以下になります．

```
x4
FALSE TRUE TRUE FALSE FALSE
```

x4の2番目と3番目の要素がTRUEであるのは，**x1**の2番目と3番目の要素が6よりも大きいからです．ベクトル**x5**は文字ベクトルです．Rでは大文字と小文字が区別されるため，ベクトル名**x5**とベクトル名**X5**は同じではありません．

cbind関数を適用することにより，ベクトル**x1**，**x2**，**x3**を行列の列とする数値行列yを作成できます．

```
y=cbind(x1,x2,x3)
```

コード**class(y)**を入力すると，“matrix”という単語が出力されます．コード**mode(y)**を入力すると“numeric”という単語が返されますが，これはyが数値行列だからです．同じ行列の中に数値ベクトルと文字ベクトルを混在させることはできません．

yと入力してEnterキーを押すと，以下の行列が出力されます．

```
y
     x1 x2 x3
[1,]  1  2  4
[2,]  7  4 18
[3,] 12  6 30
[4,]  6  8 20
[5,]  3 10 16
```

dataframeは数値，文字，論理変数を混在させることが可能なので，Rでは行列よりも汎用的なデータ保存クラスです．Rのdataframeは，様々な種類の変数を保存できるという点で，StataやSAS，SPSSのデータセットと似ています．**data.frame**関数を使用して，以下のようにベクトルや行列を結合させることができます．

z=data.frame(x1,x2,x3,x4,x5)，あるいは，**z=data.frame(y,x4,x5)**

zと入力してEnterキーを押すと，以下のdataframeが出力されます．

```
z
   x1 x2 x3    x4     x5
1   1  2  4 FALSE   blue
2   7  4 18  TRUE  green
3  12  6 30  TRUE    red
4   6  8 20 FALSE  green
5   3 10 16 FALSE purple
```

角括弧[]を用いると，dataframeまたは行列の特定の行や列にアクセスできます．コード**z[2,5]**を入力すると，行列zから2行，5列が出力されます(この例の対応する要素は“green”)．5列目の最初の3行(オブザベーション)にアクセスするには，コード**z[1:3,5]**，または**z[c(1,2,3),5]**と入力します．5列目全体にアクセスするには，**z[,5]**と入力します．あるいは，5列目(変数)は**x5**という名前なので，コード**z$x5**を入力することにより5列目全体にアクセスできます．この例で，**$**はzという名のdataframeにある**x5**という名の変数を示します．

Listは，ベクトルや行列，dataframeよりもさらに汎用的なデータ保存形式であり，これらのデータオブジェクトのいずれもlistに含めることができます．以下のコードはwという名のlistを作成します．**w**の1番目の要素は長さ(ベクトルの要素の個数)2の文字ベクトル，2番目の要素はベクトル**x1**，3番目の要素は行列**y**，4番目の要素はdataframe **z**です．

w=list(c("hello","good-bye"),x1,y,z)

二重角括弧[[ ]]を用いれば，listの特定の要素にアクセスできます．List **w**のdataframe zにアクセスするには，**z**は**w**の4番目の要素なので，コード**w[[4]]**と入力します．**w**の4番目の要素の第1行，3列にアクセスするには，以下のコードを入力します．

w[[4]] [1,3]

wの4番目の要素の1行，3列は，値4です．

Rの生存関数

一度survivalパッケージをインストールすれば，Appendixで述べる生存時間解析の実行に必要な生存関数が利用可能になります．生存関数にアクセスするには，作業開始時にコード**library(survival)**を入力してください．主な生存関数は以下の通りです．

Surv – 「time-to-event」および「status」結果変数を定義するのに使用します．この関数で作成した生存オブジェクトは，Rの他の生存関数の結果変数に用いることができます．

survfit – KMまたはCox調整生存推定値，または事前に実行したパラメトリックモデルからの生存推定値を作成します．

Survdiff – 層間の生存関数の同等性に関する統計的検定の実行に用います．

coxph – Cox比例ハザードモデル，層化Coxモデル，拡張Coxモデルの実行に用います．

cox.zph – Schoenfeld残差に基づく比例ハザード仮定に関する統計的検定を実行します．

survSplit – CP形式の新しいデータセットを作成します．レコードごとに開始時間，終了時間，イベント状態を含みます．1つのオブザベーションを，特定の時間ごとにそれぞれの生存データを付加した複数のオブザベーションに分割します．

survreg – パラメトリック生存モデルの実行に用います．

summary関数や**plot**関数のようなRの一般関数も，生存推定や作図のために生存関数とともに用います．

これらの関数についてのRのマニュアル（オンラインヘルプ）を参照するには，?後に関数名を続けて（スペースを入れずに）入力，実行します．例えば，**coxph**関数についてRのマニュアルを参照するには，コード**?coxph**を入力，実行します．

Rでは以下の生存時間解析を説明します．

1. 生存関数（未調整）の推定および層間での比較
2. グラフを用いた方法による比例ハザード性の評価
3. Cox比例ハザードモデルの実行
4. 層化Coxモデルの実行
5. 統計的検定による比例ハザード仮定の評価
6. Cox調整生存曲線の作成

7. 拡張Coxモデルの実行
8. パラメトリックモデルの実行
9. frailtyモデルの実行
10. 再発イベントのモデル構築

薬物常用者データセットを例にとって説明します．ファイルとして保存されているR dataframeにアクセスするには**load**関数を用います．薬物常用者データセットがCドライブにC:\craddicts.rdaとして保存されているとします．以下のコードで薬物常用者データを読み込みます．

```
load("c:/craddicts.rda")
```

薬物常用者データセットを出力するには，以下のコードを入力します．

```
addicts
```

最初の5オブザベーションを出力するには，以下のコードを入力します．

```
addicts[1:5,]
```

カンマの後ろに何も入力しなかったので，6つの変数(列)すべてが出力されます．コード**addicts[1:5,1:6]**と入力しても同じ結果が得られます．出力は以下の通りです．

```
  id clinic status survt prison dose
1  1      1      1   428      0   50
2  2      1      1   275      1   55
3  3      1      1   262      0   55
4  4      1      1   183      0   30
5  5      1      1   259      1   65
```

薬物常用者データセットにおけるtime-to-event変数はSURVTであり，被験者がイベント経験か打ち切りかを示す変数はSTATUSです．**Surv**関数は，これら2つの結果変数をリンクさせたRの生存オブジェクトを作成します(コードは以下の通り)．

```
Surv(addicts$survt,addicts$status==1)
```

1番目の引数(**addicts$survt**)はtime-to-event変数で，$記号の前に記載されたaddicts dataframeから引用します．2番目の引数(**addicts$status==1**)は，status変数が1ときイベント(打ち切りではなく)と指定します．2つの等号は等しい条件を表すのに用い，1つの等号はRでの割り当てに用います．**Surv**関数の出力の一部を以下に示します．

```
[1]   428  275  262  183  259  714  438  796+ 892  393  161+ 836
[13] 523  612  212  399  771  514  512  624   209  341  299  826+
[25] 262  566+ 368  302  602+ 652  293  564+ 394  755  591  787+
```

上記の出力は，薬物常用者データ（238名）の最初の36被験者の生存時間を示しています．時間の後ろのプラス（+）記号は打ち切り（イベントではなく）を示しています．

Rでの生存時間解析においては，**Surv**関数によって作成するこの生存オブジェクトを応答変数に用いることが多いです．次は，Rでの生存時間解析について具体的なテーマごとに説明します．

1．生存関数（未調整）の推定および層間での比較

Kaplan-Meier生存推定値は，Rでは3つの関数を用いて求めます．**Surv**関数（上述の通り）をsurvfit関数内で用い，指定したsurvfit関数を**summary**関数内で用います．コードは以下の通りです．

summary(survfit(Surv(addicts$survt,addicts$status==1)~1))

このコードの仕組みがわかりやすいように，各関数を分解して考えます．コード**Y=Surv(addicts$survt,addicts$status==1)**は解析で応答変数として使われる**Y**という名の生存オブジェクトを作成します．次に，コード**Y~1**について考えます．このシンタックスをformulaといいます．Rの多くの関数において，特に統計モデルを指定する関数においてformulaを引数に用います．**Y~1**は切片のみのモデルを要求します．言い換えれば，応答変数に何ら条件付けを行わないということです．このセクションの後で変数CLINICで層別するときには，formula **Y~〜addicts$clinic**を用います．Formulaを**survfit**関数の引数に用います（以下のように）．

kmfit1=survfit(Y~1)

survfit関数により**kmfit1**と名付けたオブジェクトが作成されます．コード**kmfit1**を入力してEnterキーを押すと，以下の出力が得られます．

```
>kmfit1
records  n.max  n.start  events  median  0.95LCL  0.95UCL
    238    238      238     150     504      399      560
```

出力は，レコード数，時間0におけるat risk数，イベント数，メディアン生存時間推定値と95%信頼区間の要約統計量となります．次に**summary**関数を用いて，すべてのイベント時間に関するKaplan-Meier生存推定値を求めます．コード**summary(kmfit1)**は，前述のコード**summary(survfit(Surv(addicts$survt,addicts$status==1)～1))**と同じ結果になります．出力は以下の通りです．

```
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    7    236       1    0.996 0.00423        0.9875        1.000
   13    235       1    0.992 0.00597        0.9799        1.000
   17    234       1    0.987 0.00729        0.9731        1.000
   19    233       1    0.983 0.00840        0.9667        1.000
   26    232       1    0.979 0.00937        0.9606        0.997
   29    229       1    0.975 0.01026        0.9546        0.995
   30    228       1    0.970 0.01107        0.9488        0.992
      .
      .
      .
  821     20       2    0.225 0.03675        0.1635        0.310
  836     17       1    0.212 0.03690        0.1506        0.298
  837     16       1    0.199 0.03689        0.1380        0.286
  857     14       1    0.184 0.03688        0.1246        0.273
  878     13       1    0.170 0.03667        0.1116        0.260
  892     10       1    0.153 0.03675        0.0958        0.245
  899      9       1    0.136 0.03639        0.0807        0.230
```

summary関数に**times=**オプションを使用して，特定の生存時間（例えば365日）に対する生存推定値を得ることもできます．コードと出力は以下の通りです．

summary(kmfit1,times=365)

```
time n.risk  n.event  survival  std.err   lower 95% CI upper 95% CI
 365    122       87    0.606   0.0331          0.545          0.675
```

変数CLINICで層別し，特定の時間のKaplan-Meier生存推定値を比較したい場合は，まず，**kmfit2**という名前のオブジェクト（名前は任意）を**survfit**関数から作成します．

kmfit2=survfit(Y~addicts$clinic)

CLINICの水準ごとに，特定の時間（100日ごと）における生存推定値を得るためには，以下のコードを入力します．

summary(kmfit2,times=c(0,100,200,300,400,500,600,700,800,900,1000))

出力は以下の通りです．

```
                addicts$clinic=1
time n.risk n.event survival std.err lower 95% CI upper 95% CI
   0    163       0   1.0000  0.0000      1.00000        1.000
 100    137      20   0.8746  0.0262      0.82467        0.928
 200    110      20   0.7420  0.0353      0.67601        0.814
 300     87      20   0.6046  0.0399      0.53120        0.688
 400     68      14   0.5025  0.0415      0.42741        0.591
 500     53       9   0.4319  0.0418      0.35719        0.522
 600     30      16   0.2951  0.0403      0.22570        0.386
 700     20       8   0.2113  0.0383      0.14818        0.301
 800     10       8   0.1268  0.0326      0.07660        0.210
 900      1       7   0.0181  0.0172      0.00283        0.116

                addicts$clinic=2
time n.risk n.event survival std.err lower 95% CI upper 95% CI
   0     75       0    1.000  0.0000        1.000        1.000
 100     66       5    0.932  0.0294        0.876        0.991
 200     58       7    0.832  0.0442        0.750        0.924
 300     50       7    0.730  0.0530        0.633        0.842
 400     43       3    0.685  0.0558        0.584        0.804
 500     39       2    0.653  0.0577        0.549        0.776
 600     27       1    0.634  0.0590        0.528        0.761
 700     19       1    0.606  0.0625        0.495        0.742
 800     11       1    0.575  0.0669        0.457        0.722
 900      7       1    0.517  0.0812        0.380        0.703
1000      3       0    0.517  0.0812        0.380        0.703
```

生存推定値が100日ごとに示されています．CLINIC = 1では，生存時間を1000まで要求したにもかかわらず900までしかありません．これは，1000日にはat riskの被験者はいないからです．生存時間ベクトルを要求するsummary関数の2番目の引数は，summary(kmfit2,times=100*(0:10))と書くこともできます．このシンタックスを使用した場合でも出力は同じです．

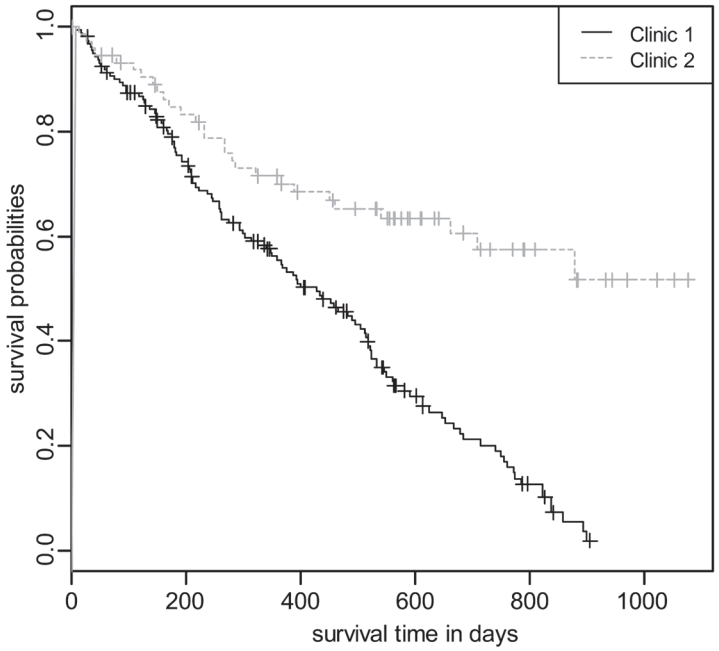
KM生存時間プロットは，plot関数によって求めます．

plot(kmfit2)

plot関数の作図用のオプションはたくさんあります．CLINIC = 1とCLINIC = 2を作図上で区分するためのコードは，線種(**lty=**)，色(**col=**)などがあります．x軸とy軸のラベルに関しては，xlab=とylab=があります．もし，コード**col( )**を実行すると，600を超える色のリストが返されます．その中の色をcol=オプションで指定することができます．**legend**関数は凡例を追加するのに使用します．1番目の引数“**topright**”は，凡例をグラフの右上に配置します．コードと出力は以下の通りです．

plot(kmfit2, lty = c(“solid”, “dashed”), col=c(“black”,“grey”), xlab=“survival time in days”,ylab=“survival probabilities”)

legend(“topright”, c(“Clinic 1”,“Clinic 2”), lty=c(“solid”,“dashed”), col=c(“black”,“grey”))



このプロットでは，CLINIC = 2の被験者の生存率はCLINIC = 1よりも高くなっています．

survdiff関数で変数CLINICに関するログランク検定を実行します（コードは以下の通り）．

survdiff（Surv（survt,status）~clinic, data=addicts）

survdiff関数の2番目の引数**data=addicts**は，addictsデータセットの変数を使うことを示しています．あるいは以下のコード指定の方法もあります．

survdiff（Surv（addicts$survt,addicts$status）～addicts$clinic）

3つ目の方法として，**attach**関数を用いて，以後指定する変数名はaddictsデータセットのものであることを設定することもできます（変数が指定されたときRはaddictsデータセット内を検索する）．**detach**関数を用いると，引数内でのデータセット名の指定は不要になります．

attach（addicts）
survdiff（Surv（survt,status）~clinic）

出力は以下の通りです.

```
            N Observed Expected (O-E)^2/E (O-E)^2/V
clinic=1 163      122     90.9      10.6      27.9
clinic=2  75       28     59.1      16.4      27.9

 Chisq= 27.9  on 1 degrees of freedom, p= 1.28e-07
```

ログランク統計量は，p値 = 0.000000128（1.28e-07）と非常に有意です.

survdiff関数の**rho**=オプションを用いて，様々なログランク検定の変法が行えます．j番目のfailure時間の検定統計量への寄与に関する重みは$s(t_j)^{rho}$となります．ここで$s(t_j)$は時間t_jにおけるKM生存推定値です．rho = 0ならば$s(t_j)^0 = 1$となるので，各failure時間に関する重みは等しくなり，ログランク検定となります．rho = 1ならば$s(t_j)^1 = s(t_j)$となるため，各failure時間の重みはそのfailure時間におけるKM生存推定値となります．この検定はGehan-Wilcoxon流のPeto・Peto検定となります．rho = 1の場合のコードおよび出力は以下の通りです.

survdiff(Surv(survt,status)～clinic,data=addicts,rho=1)

```
            N Observed Expected (O-E)^2/E (O-E)^2/V
clinic=1 163     77.3     61.4      4.08      15.8
clinic=2  75     19.9     35.7      7.03      15.8

 Chisq= 15.8  on 1 degrees of freedom, p= 7.18e-05
```

rho = 1における検定結果は$\chi^2 = 15.8$，p値 = 0.0000718で，ログランク検定の結果といくぶん異なりますが，生存に関するCLINICの効果が高い有意性を示すことは変わりません.

CLINICに関する層化ログランク検定（PRISONで層別）は，モデルformulaに+ **strata(prison)**項を加えることで実行できます．この層化アプローチでは，（イベント観測数－期待イベント数）を各群内の各層内ですべてのfailure時間で積算し，各層の積算値を合計します．コードと出力は以下の通りです.

survdiff(Surv(survt,status) ~ clinic + strata(prison),data=addicts)

```
            N Observed Expected (O-E)^2/E (O-E)^2/V
clinic=1 163      122     91.7      10.0      26.9
clinic=2  75       28     58.3      15.8      26.9

 Chisq= 26.9  on 1 degrees of freedom, p= 2.1e-07
```

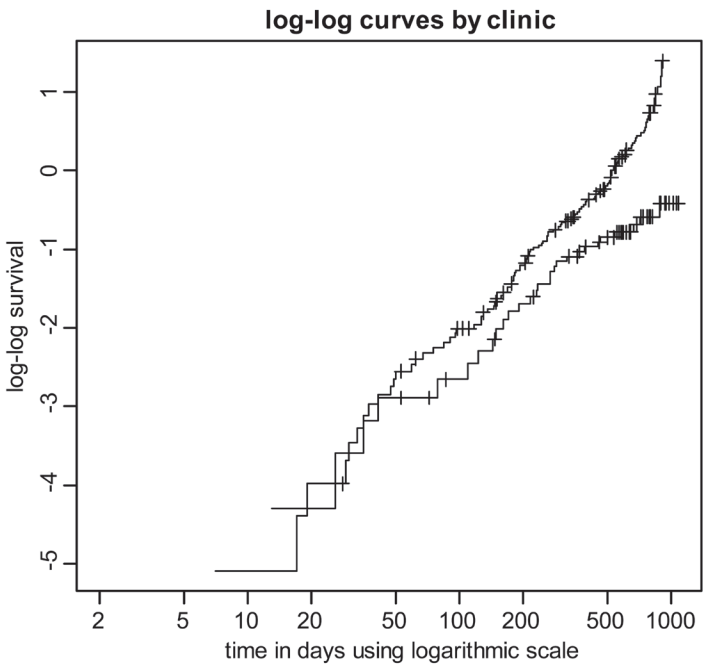
survdiff関数のformulaには＋**strata**(**prison**)項が含まれますが，この検定結果は，PRISONで層別しないログランク検定の結果とよく似ています．

2. グラフを用いた方法による比例ハザード性の評価

CLINICに関する比例ハザード性は，対数(－対数)Kaplan-Meier生存推定値と時間(または時間の対数)をプロットし，曲線が平行と見なせるかで評価します．前のセクションでは，**survfit**を用いて**kmfit2**という名の生存オブジェクトを作成しました．また，コード**plot**(**survfit2**)を用いて生存推定値と時間をプロットしました．**fun="cloglog"**オプションを**plot**関数に加えると，対数(－対数)生存時間と対数時間がプロットされます．コードは以下の通りです．

plot(kmfit2,fun="cloglog",xlab="time in days using logarithmic scale",ylab="log-log survival", main="log-log curves by clinic")

xlab=と**ylab=**はそれぞれx軸とy軸のラベルを指定し，**main=**オプションはタイトルを指定します．**fun="cloglog"**はcloglog関数の指定です．出力は以下の通りです．



対数(－対数)生存曲線が平行ではないため，このプロットからは比例ハザード性が成立しないことが示唆されます．**fun**=オプション(**fun**はfunction(関数))は時間を対数尺度でプロットします. 対数(－対数)生存推定値と時間(対数尺度ではない)のプロットを直接的に出力するオプションはありません．しかし，プログラムにより解析結果を保存し，加工して，作図する方法があります．この作業を行うためには，まず，summary関数を用いて生存推定値をオブジェクトとして(**kmfit3**という名前で)保存します．

```
kmfit3=summary(kmfit2)
```

コード**names**(**kmfit3**)を実行すると，オブジェクト**kmfit3**の列名が出力されます．この列で興味があるのは，各被験者の生存時間，KM生存推定値，CLINICの水準(1 or 2)に該当する列です．**names**関数により，これらの列はそれぞれ**time**，**surv**，**strata**という名前であることがわかります．**kmfit3$time**，**kmfit3$surv**，**kmfit3$strata**などのコードを実行すれば，これら列の内容を確認することができます．data.frame関数を用いて，これら3つの列を変数に持つdataframe(**kmfit4**という名前の)を作成します．

```
kmfit4=data.frame(kmfit3$strata,kmfit3$time,kmfit3$surv) names
(kmfit4)=c("clinic","time","survival")
```

kmfit4に**names**関数を適用し(上記載)，デフォルト変数名を変更します．以下に**kmfit4**の最初の5オブザベーションを出力します．

```
kmfit4[1:5, ]
```

	clinic	time	survival
1	addicts$clinic=1	7	0.99382716
2	addicts$clinic=1	17	0.98765432
3	addicts$clinic=1	19	0.98148148
4	addicts$clinic=1	29	0.97523001
5	addicts$clinic=1	30	0.96897853

dataframe **kmfit4**をCLINIC = 1とCLINIC = 2に対応する2つのdataframe(**clinic1**と**clinic2**という名前)に分割します．コードは以下です．

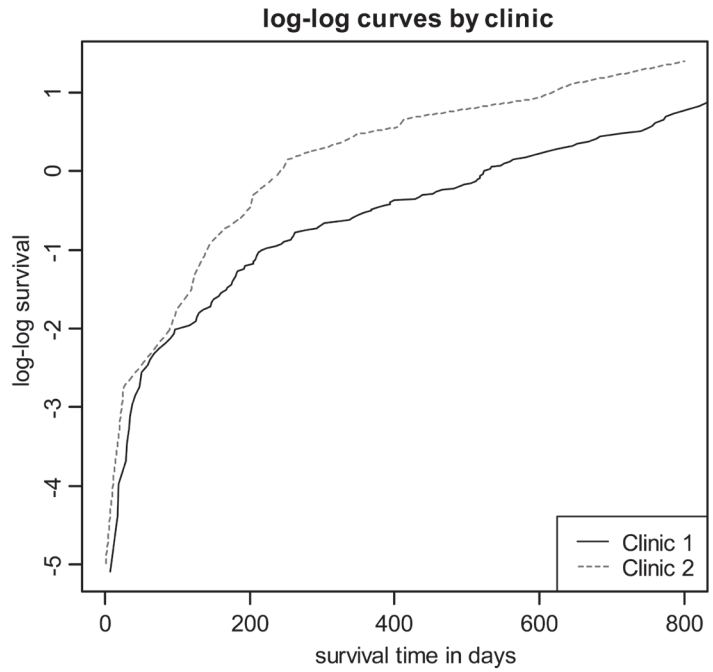
```
clinic1=kmfit4[kmfit4$clinic=="addicts$clinic=1",]
clinic2=kmfit4[kmfit4$clinic=="addicts$clinic=2",]
```

dataframe **clinic1**と**clinic2**にはそれぞれ，CLINIC = 1とCLINIC = 2の被験者の生存時間，生存推定値が含まれます．

これを用いて，**plot**関数により対数(－対数)の生存曲線と時間のプロットを行います(時間は対数尺度ではない)．コードは以下の通りです．

```
plot(clinic1$time,log(-log(clinic1$survival)),xlab="survival time in days",ylab="log-log survival",xlim=c(0,800),col="black",type='l',lty="solid",main="log-log curves by clinic")
par(new=T)
plot(clinic2$time,log(-log(clinic2$survival)),axes=F,xlab="survival time in days",ylab="log-log survival",col="grey50",type='l',lty= "dashed")
legend("bottomright", c("Clinic 1", "Clinic 2"), lty = c("solid", "dashed"),col=c("black","grey50"))
par(new=F)
```

1番目のプロットでは，dataframe **clinic1** を用いて，時間(**clinic1$time**)をx軸に，生存(**clinc1$survival**)の対数(－対数)をy軸にプロットします．コード**par**(**new=T**)は，1番目のプロットが2番目のプロット作成時に消去されない指定です(つまり2つのプロットを重ねる)．**par**関数は，グラフパラメータの設定または照会に用います．2番目の**plot**関数は最初のものと似ていますが，プロットするデータはdataframe **clinic2**のものです．**legend**関数で凡例が追加され，最後に**par**(**new=F**)でグラフパラメータ **new**をデフォルト値のfalseに戻します(次のプロットの作成時に既存のプロットを消去する)．下記のグラフが出力されます．



このプロットは，CLINICに関する比例ハザード性が成立しないことを示唆します．

3. Cox比例ハザードモデルの実行

coxph関数を用いてCox比例ハザードモデルを実行します．まず，**Surv**関数を用いて応答変数を指定し，次に**coxph**関数を用いて変数CLINIC，PRISON，DOSEを含むCox比例ハザードモデルを実行します．コードと**coxph**による出力は以下の通りです．

```
Y=Surv(addicts$survt,addicts$status==1)
coxph(Y~prison + dose + clinic,data=addicts)
```

```
          coef exp(coef) se(coef)      z       p
prison  0.3266     1.386  0.16722  1.95 5.1e-02
dose   -0.0354     0.965  0.00638 -5.54 2.9e-08
clinic -1.0099     0.364  0.21489 -4.70 2.6e-06

Likelihood ratio test=64.6  on 3 df, p=6.23e-14  n= 238, number of events= 150
```

この出力には，回帰係数，指数化係数（ハザード比推定値），標準誤差，z検定，係数に対応するp値が含まれます．95%信頼区間を含む追加出力を得るには，**coxph**関数に**summary**関数を適用します（コードと出力は以下の通り）．

```
summary(coxph(Y~ prison + dose + clinic,data=addicts))
```

```
              coef exp(coef)  se(coef)       z Pr(>|z|)
prison   0.326555  1.386184  0.167225   1.953   0.0508 .
dose    -0.035369  0.965249  0.006379 -5.545 2.94e-08 ***
clinic  -1.009896  0.364257  0.214889 -4.700 2.61e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1

        exp(coef) exp(-coef) lower .95 upper .95
prison     1.3862     0.7214    0.9988    1.9238
dose       0.9652     1.0360    0.9533    0.9774
clinic     0.3643     2.7453    0.2391    0.5550

Rsquare= 0.238   (max possible= 0.997 )
Likelihood ratio test= 64.56  on 3 df,   p=6.228e-14
Wald test            = 54.12  on 3 df,   p=1.056e-11
Score (logrank) test = 56.32  on 3 df,   p=3.598e-12
```

出力の2番目の表から，CLINIC = 2 vs CLINIC = 1に関するハザード比推定値は「exp(coef)」列にある0.3643で，その95%信頼区間（0.2391，0.5550）です．「exp(-coef)」列は，CLINIC = 1 vs CLINIC = 2に関するハザード比推定値で，値2.7453は0.3643の逆数となっています．

複数の被験者のイベントが同時点で観察されるデータの場合，Cox尤度の同順位の処理にはいくつかのオプションが準備されています．Rの**coxph**関数では3つの方法，1）Efron法（デフォルト），2）Breslow法，3）exact法，が提供されています．一般的に，これらの方法の違いは推定にほとんど影響を及ぼしませんが，ソフトウェアパッケージによってデフォルトの設定は異なっています．Rのデフォルトは Efron 法ですが, Stata, SAS, SPSS では Breslow 法がデフォルトです. **coxph** 関数の **method**= オプションを用いて，同順位の処理方法を指定します（コードは以下の通り，出力は省略）．

coxph（Y~ prison + dose + clinic,data=addicts, method="efron"）
coxph（Y~ prison + dose + clinic,data=addicts, method="breslow"）
coxph（Y~ prison + dose + clinic,data=addicts, method="exact"）

次に，PRISONに関する2つの交互作用（積）項をモデルに含め，尤度比検定を用いて交互作用項の同時有意性を検定します．以下のコードはcoxph関数からの情報を持つ2つのオブジェクト（mod1とmod2という名の）を作成します．1つは非交互作用モデル（mod1 – 縮小モデル）であり，もう1つは交互作用モデル（mod2 – フルモデル）です．

mod1=coxph（Y ~ prison + dose + clinic,data=addicts）
mod2=coxph（Y ~ prison + dose + clinic + clinic*prison + clinic*dose, data=addicts）

交互作用項を確認するためにコードmod2を入力します（コードと出力は以下の通り）．

mod2

```
                coef exp(coef) se(coef)      z     p
prison         1.1924     3.295   0.5414  2.202 0.028
dose          -0.0192     0.981   0.0194 -0.990 0.320
clinic         0.1796     1.197   0.8933  0.210 0.840
prison:clinic -0.7383     0.478   0.4315 -1.711 0.087
dose:clinic   -0.0140     0.986   0.0143 -0.974 0.330

Likelihood ratio test=68.2  on 5 df, p=2.45e-13  n= 238, number of events= 150
```

これからは少し複雑になりますが，Rの解析出力にアクセスし処理する方法を説明します．

オブジェクト**mod1**と**mod2**には，利用したい情報が含まれています．コード**names(mod2)**と入力して，**mod2**の要素の名前を確認します(コードと出力は以下の通り)．

```
names(mod2)
 [1] "coefficients"      "var"           "loglik"
 [4] "score"             "iter"          "linear.predictors"
 [7] "residuals"         "means"         "concordance"
[10] "method"            "n"             "nevent"
[13] "terms"             "assign"        "wald.test"
[16] "y"                 "formula"       "call"
```

mod2の3番目の要素は“loglik”です．この名前で保存されているデータにアクセスするにはコード**mod2$loglik**と入力するか，loglikがこのlistの3番目の要素であるため，**mod2[[3]]**と入力します(コードと出力は以下の通り)．

```
mod2$loglik
[1] -705.5393 -671.4500
```

mod2$loglikの2番目の要素は－671.5997であり，2つの交互作用項を含むモデルの対数尤度です．1番目の要素 －704.6619は，説明変数を含まないモデルの対数尤度です(今は興味がありません)．

次に，2つの交互作用項に関する尤度比検定を実行したいと思います．検定統計量を計算するには，縮小モデル(交互作用項なし)の対数尤度からフルモデル(交互作用項あり)の対数尤度を引いたものに，－2を掛ける必要があります．以下のコードにより，その計算結果が得られます．

```
(-2)*(mod1$loglik[2]-mod2$loglik[2])
```

得られた結果3.605457が，尤度比検定統計量です．帰無仮説の基で，この検定統計量は自由度2のχ^2分布に従います．**pchisq**関数を用いてこの検定のp値を求めます．コード**1-pchisq(3.605457,2)**は，自由度2のχ^2検定のp値を返します．まとめると，以下のコードで尤度比検定のp値が得られます(出力も添付)．

```
LRT=(-2)*(mod1$loglik[2]-mod2$loglik[2])
Pvalue = 1 - pchisq(LRT, 2)
Pvalue
0.1648485
```

有意水準0.05において，p値0.168485は有意ではありません．

Rの強力な機能の1つに，ユーザーが独自の関数を定義できることがあります．この機能について説明するために，2つのCoxモデル（フルモデルと縮小モデル）から尤度比検定を実行する独自の関数を定義してみます．以下のコードは**lrt.surv**という関数を作成します．この関数は，3つの引数，すなわち（1）フルモデルの名前，（2）縮小モデルの名前，（3）検定の自由度，を持ちます．この関数は尤度比検定のp値を返します．

新しい関数の定義はR関数**function**を用います．定義する関数の3つの引数を**mod.full**，**mod.reduced**，**df**とすることにします．引数の記載部分の後ろに，関数の計算部分のコードを中括弧{}で囲みます．R関数**return**は，定義した関数から返すR出力の内容を示すものです（この例では，尤度比検定のp値）．コードは以下の通りです．

```
lrt.surv=function(mod.full,mod.reduced,df) {
lrts=(-2)*(mod.full$loglik[2]- mod.reduced$loglik[2])
pvalue=1-pchisq(lrts,df)
return(pvalue)
}
```

一度このコードを実行すれば，R実行中はいつでも関数**lrt.surv**を用いて，2つのCoxモデルを比較する尤度比検定のp値を得ることができます．この新しい関数を用いて，前述と同様，オブジェクト**mod1**と**mod2**に関する尤度比検定を実行してみます．コードと出力は以下の通りです．

```
lrt.surv(mod1, mod2, 2)
[1]0.1648485
```

p値は前述のものと同じです．関数**lrt.surv**を定義することで汎用性が高まり，この例以外にも，2つのCoxモデル（フルモデルと縮小モデル）を比較する尤度比検定のp値を簡単に求めることができます．

4. 層化Coxモデルの実行

変数CLINICに関しては比例ハザード仮定は成立しないが，PRISONとDOSEに関しては成立するならば，CLINICを層化変数に用いた層化Coxモデルが可能となります．**coxph**関数のモデルformulaに**strata()**オプションを加えます．まず**Surv**関数を用いて応答変数**Y**を定義し，次に**coxph**関数を用いて層化Coxモデルを実行します（コードと出力は以下の通り）．

```
Y=Surv(addicts$survt,addicts$status==1)
coxph(Y~ prison + dose + strata(clinic),data=addicts)
```

```
          coef exp(coef) se(coef)      z       p
prison  0.3896     1.476  0.16893  2.31 2.1e-02
dose   -0.0351     0.965  0.00646 -5.43 5.6e-08

Likelihood ratio test=33.9  on 2 df, p=4.32e-08  n= 238, number of events= 150
```

「:」演算子を用いて積項（**clinic:prison**と**clinic:dose**）をモデルformula入れることにより，CLINICに関する交互作用項を直接モデル化することができます（コードと出力は以下の通り）．

```
coxph(Y~ prison + dose + clinic:prison + clinic:dose +
strata(clinic),data=addicts)
```

```
                  coef exp(coef) se(coef)       z     p
prison         1.08584     2.962   0.5386  2.0159 0.044
dose          -0.03464     0.966   0.0198 -1.7495 0.080
prison:clinic -0.58299     0.558   0.4281 -1.3617 0.170
dose:clinic   -0.00116     0.999   0.0146 -0.0799 0.940

Likelihood ratio test=35.8  on 4 df, p=3.22e-07  n= 238, number of events= 150
```

CLINIC = 2におけるPRISON = 1 vs. PRISON = 0のハザード比を推定するには，「prisonの係数」+ 2 ×「CLINIC* PRISON交互作用項の係数」を指数化します．この計算式は，ハザード比の式の分子にPRISON = 1，分母にPRISON = 0を代入することにより得られます（以下参照）．

$$HR = \frac{h_0(t)\exp[1\beta_1 + \beta_2 DOSE + (2)(1)\beta_3 + \beta_4 CLINIC \times DOSE]}{h_0(t)\exp[10 + \beta_2 DOSE + (2)(0)\beta_3 + \beta_4 CLINIC \times DOSE]} = \exp(\beta_1 + 2\beta_2).$$

ハザード比$\exp(\beta_1 + 2\beta_2)$は，パラメータの線形結合を指数化したものです．残念ながらRにはパラメータ推定値の線形結合計算するための，Stataの**lincom**コマンドやSASの**estimate**ステートメントに該当するオプションはありません．しかしながら，どのような統計ソフトでもこの状況に対応できる方法があります．それは，求める推定がパラメータの線形結合ではなくなるように興味ある変数を再コード化するというものです．

この例で知りたいのはCLINIC = 2におけるPRISON = 1 vs. PRISON = 0のハザード比です．そこで，先のCLINIC = 2の計算式において$2\beta_2 = 0$となるように新しい変数CLINIC2を定義します．

```
addicts$clinic2=addicts$clinic-2
summary(coxph(Y~ prison+dose+clinic2:prison+
clinic2:dose+strata(clinic2),data=addicts))
```

コードの1行目は新しい変数CLINIC2を定義しています．CLINIC2をCLINICの代わりに層化Coxモデルで使用します．知りたいのはCLINIC2 = 0（CLINIC = 2）のPRISON = 1 vs. PRISON = 0のハザード比となります．CLINIC2 = 0ならば積項は整理され，ハザード比は$\exp(\beta_1)$となります．

コードの2行目は，**coxph**関数に**summary**関数を適用しています．このようにsummary関数を用いると，ハザード比の95%信頼区間を含む出力が得られます．出力は以下の通りです．

```
 n= 238, number of events= 150

                    coef exp(coef)  se(coef)      z Pr(>|z|)
prison          -0.080143  0.922985  0.384305 -0.209  0.83481
dose            -0.036964  0.963711  0.012346 -2.994  0.00275 **
prison:clinic2 -0.582989  0.558227  0.428135 -1.362  0.17329
dose:clinic2   -0.001164  0.998837  0.014570 -0.080  0.93632
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

               exp(coef) exp(-coef) lower .95 upper .95
prison            0.9230      1.083    0.4346    1.9603
dose              0.9637      1.038    0.9407    0.9873
prison:clinic2    0.5582      1.791    0.2412    1.2919
dose:clinic2      0.9988      1.001    0.9707    1.0278

Concordance= 0.649  (se = 0.034)
Rsquare= 0.14   (max possible= 0.994 )
Likelihood ratio test= 35.77  on 4 df,   p=3.222e-07
Wald test            = 34.09  on 4 df,   p=7.138e-07
Score (logrank) test = 34.97  on 4 df,   p=4.706e-07
```

exp(β_1)の推定値は，2番目の表の“exp(coef)”列にprison = 0.9230とあります．信頼区間の上限値および下限値はそれぞれ0.4346と1.9603です．もし変数CLINICを再コード化しなければ，分散－共分散行列(**vcov**関数で出力可能)を用いてハザード比の95%信頼区間を計算しなければならず，面倒なことになります．

5. 統計的検定による比例ハザード仮定の評価

cox.zph関数は，比例ハザード仮定に関する統計的検定を行うためのものです．この統計的検定は，Schoenfeld残差と生存時間(または生存時間順位)との相関の有無を検定するものです．相関が0であれば，比例ハザード仮定(帰無仮説)を支持します．まず，**Surv**関数を用いて反応変数**Y**を定義し，次にcoxph関数を用いて，変数PRISON，DOSE，CLINICを説明変数としたCox比例ハザードモデルを実行します．

```
Y=Surv(addicts$survt,addicts$status==1)
mod1=coxph(Y~prison + dose + clinic, data=addicts)
```

coxph関数によりオブジェクトmod1が作成されます．このオブジェクトがcox.zph関数の1番目の引数となります．比例ハザード仮定の検定を実行するコードは以下の通りです．

```
cox.zph(mod1,transform=rank)
```

2番目の引数は，実際の生存時間(デフォルト)ではなく生存時間順位をSchoenfeld残差との相関の検定に用いるように指定するものです．出力は以下の通りです．

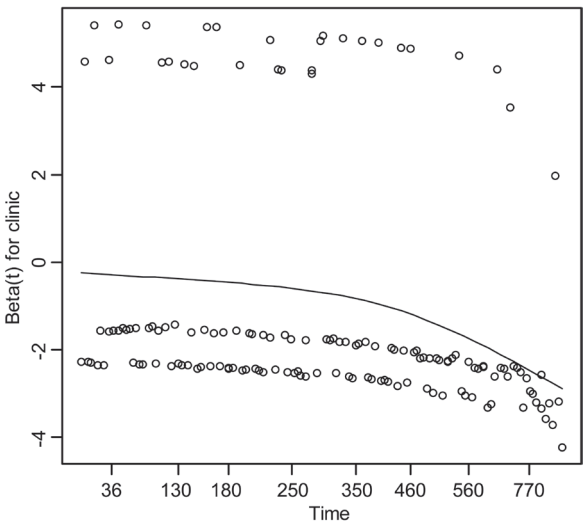
```
            rho  chisq       p
prison -0.0462  0.322 0.57068
dose    0.0905  1.096 0.29521
clinic -0.2498 10.495 0.00120
GLOBAL      NA 12.425 0.00606
```

出力からは，変数CLINICのSchoenfeld残差(3行目)と生存時間順位との相関は－0.2498でp値は0.00120であることがわかります．この有意なp値は，変数CLINICについては比例ハザード仮定が成立しないこと示しています．PRISONとDOSEのp値は有意ではなく，PRISONとDOSEに関する比例ハザード仮定を棄却するには十分な証拠がないことを示唆しています．

包括的検定（4行目 GLOBAL）はモデル全体の比例ハザード仮定を検定するものであり（3つの予測変数すべてを同時に），p = 0.00606で有意です．この包括的検定は，モデル全体としては比例ハザード仮定が成立していないことを示しています．

Schoenfeld残差とfailure時間のプロットは，**cox.zph**関数で作成したオブジェクトを最初の引数とした**plot**関数にて行います．引数**var=clinic**は，変数CLINICの残差を指定するものです．引数**se=F**は，回帰曲線の信頼区間を表示しない指定です．コードと出力は以下の通りです．

plot（cox.zph（mod1,transform=rank）,se=F,var= ‘clinic’ ）



比例ハザード仮定が成立するならばSchoenfeld残差は生存時間と独立になるので，回帰曲線は水平になるはずです．しかし，この回帰曲線は右肩下がりになっています．

6. Cox調整生存曲線の作成

Cox調整生存推定値とプロットを得るには，**survfit**関数で作成したオブジェクトに**summary**関数または**plot**関数を適用します．まず，**coxph**関数でCoxモデルを実行します．

```
Y=Surv（addicts$survt,addicts$status==1）
mod1=coxph（Y~ prison + dose + clinic, data=addicts）
```

一般的に，調整生存曲線は共変量パターンに依存します．そこで，PRISON = 0，DOSE = 70，CLINIC = 2のパターンで生存曲線をプロットします．まず，data.frame関数を用いて1オブザベーションのデータセット（またはdataframe）を作成します．

コードと出力は以下の通りです.

pattern1=data.frame(prison=0,dose=70,clinic=2)

pattern1

```
  prison dose clinic
1      0   70      2
```

1オブザベーションdataframeを**pattern1**と名付けました．Cox調整生存推定値を得るには，以下のように**survfit**関数と**summary**関数を用います.

summary(survfit(mod1,newdata=pattern1))

survfit関数の1番目の引数は，**coxph**関数で作成したオブジェクト**mod1**です．2番目の引数は，先程の共変量パターン(**pattern1**)を格納したdataframeです.

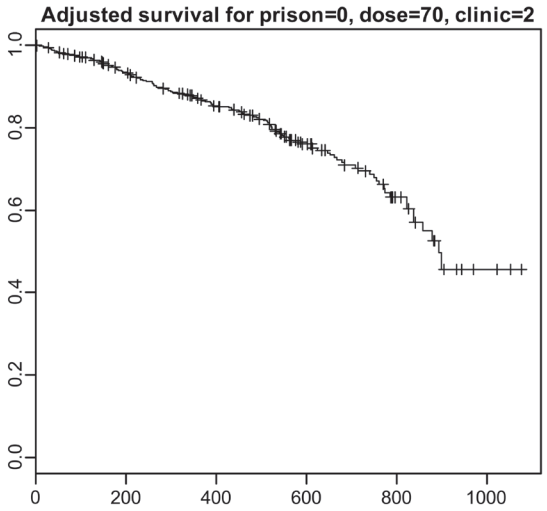
出力は以下の通りです.

```
time n.risk n.event survival std.err lower 95% CI upper 95% CI
   7    236       1    0.999 0.00105        0.997        1.000
  13    235       1    0.998 0.00154        0.995        1.000
  17    234       1    0.997 0.00193        0.993        1.000
  19    233       1    0.996 0.00229        0.991        1.000
  26    232       1    0.995 0.00263        0.990        1.000
  29    229       1    0.994 0.00296        0.988        1.000
  30    228       1    0.993 0.00328        0.986        0.999
  33    227       1    0.992 0.00359        0.985        0.999
  35    226       2    0.989 0.00419        0.981        0.998
      .
      .
      .
 857     14       1    0.549 0.07953        0.414        0.730
 878     13       1    0.526 0.08204        0.387        0.714
 892     10       1    0.496 0.08580        0.354        0.697
 899      9       1    0.456 0.09090        0.309        0.674
```

この共変量パターンでCox調整生存曲線を得るには，上記でsummary関数を適用したのと同じやり方でplot関数を用います．コードは以下の通りです.

plot(survfit(mod1,newdata=pattern1),conf.int=F,main="Adjusted survival for prison=0, dose=70, clinic=2")

conf.int=Fオプションは信頼区間の表示を抑制します．**conf.int=T**オプション（デフォルト）を用いれば95%信頼区間を表示します．**main=**オプションで図のタイトルを指定します．出力は以下の通りです．



層化Cox調整生存曲線を作成するためには，まず層化Coxモデル（CLINICで層別）を実行します．

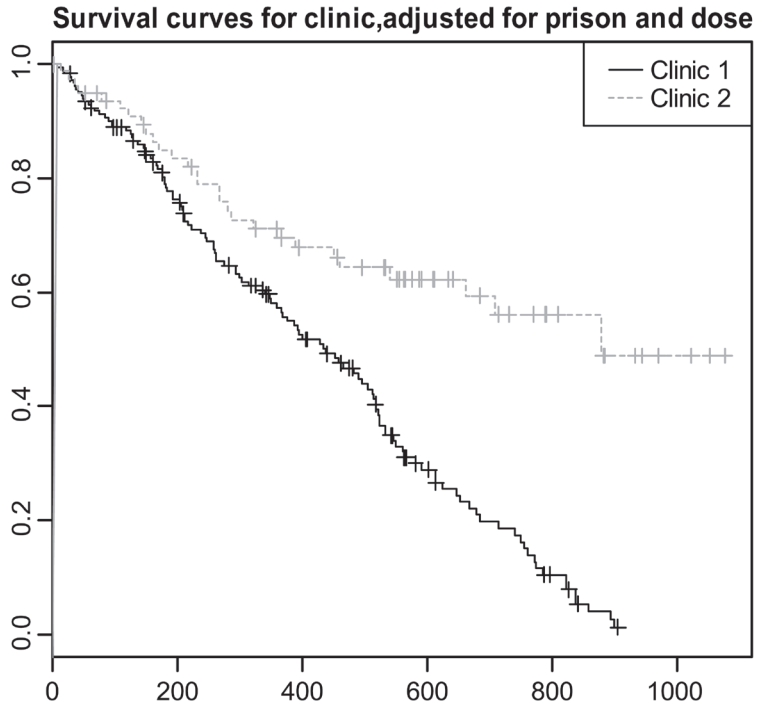
mod3=coxph（Y~ prison + dose + strata（clinic）,data=addicts）

PRISONとDOSEで調整した調整層化Cox曲線を作成するために，PRISONの平均値0.46，DOSEの平均値60.4を持つ1オブザベーションのdataframeを作成します．

pattern2=data.frame（prison=.46,dose=60.40）

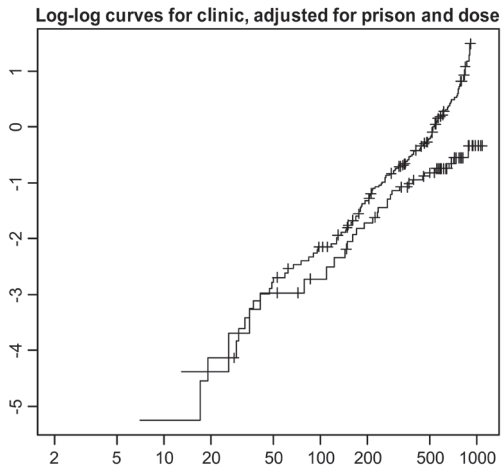
そして先の例で示したように，**survfit**関数と**plot**関数を使います．コードと出力は以下の通りです．

plot（survfit（mod3,newdata=pattern2）, conf.int=F, lty = c（“solid”, “dashed”）, col=c（“black”,“grey”）, main=“Survival curves for clinic, adjusted for prison and dose”）
legend（“topright”, c（“Clinic 1”,“Clinic 2”）, lty=c（“solid”,“dashed”）, col=c（“black”,“grey”））



plot関数に**fun**=オプションを用いると対数(－対数)生存曲線がプロットされます．コードと出力は以下の通りです．

plot(survfit(mod3,newdata=pattern2),fun="cloglog", main= "Log-log curves for clinic, adjusted for prison and dose")

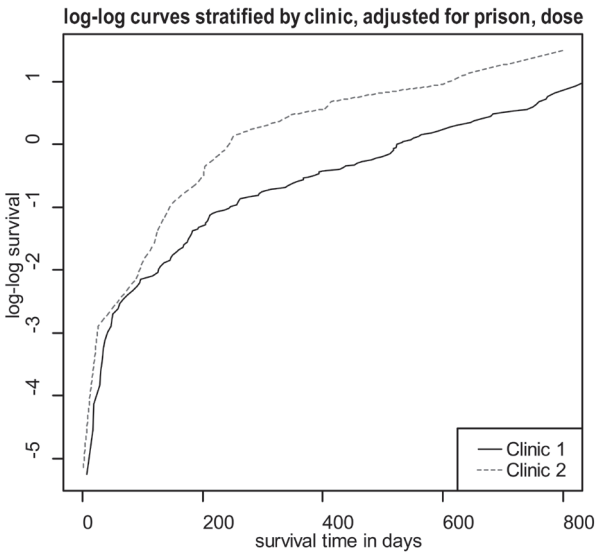


fun="cloglog"オプションは時間を対数尺度でプロットします．時間(対数尺度ではない)に対する対数(－対数)プロットを作成するオプションはありません．この作成方法についてはセクション2のKM対数(－対数)曲線ですでに示しています．まず，調整生存推定値をオブジェクトsum.mod3に保存します(以下の通り)．

```
sum.mod3=summary(survfit(mod3,newdata=pattern2))
```

次に，時間（対数尺度ではない）に対する対数（－対数）プロットを作成するために，セクション2で示したコードを用い，オブジェクト**kmfit3**の代わりに上記で作成したオブジェクト**sum.mod3**に置き換えます．コードとプロットは以下の通りです．

```
sum.mod4=data.frame(sum.mod3$strata,sum.mod3$time,sum.
mod3$surv)
colnames(sum.mod4)=c("clinic","time","survival")
clinic1=sum.mod4[sum.mod4$clinic=="clinic=1", ]
clinic2=sum.mod4[sum.mod4$clinic=="clinic=2", ]

plot(clinic1$time,log(-log(clinic1$survival)),xlab="survival
time in days",ylab="log-log survival",xlim=c(0,800),col=
"black",type= 'l' ,lty="solid", main="log-log curves stratified by
clinic, adjusted for prison, dose")

par(new=T)

plot(clinic2$time,log(-log(clinic2$survival)),axes=F,xlab=
"survival time in days",ylab="log-log survival",col="grey50",
type= 'l' ,lty="dashed")

legend("bottomright", c("Clinic 1", "Clinic 2"), lty = c("solid",
"dashed"),col=c("black","grey50"))

par(new=F)
```



7. 拡張Coxモデルの実行

Stata，SAS，SPSSとは異なり，Rで拡張Coxモデルを実行するためには，解析データセットがCP（開始，終了）形式になっていなければなりません．残念ながらaddictsデータセットはこの形式になっていないため，時間依存性共変量を含めるためにはCP形式に変換する必要があります．CP形式への変換には**survsplit**関数を用います．**survsplit**関数は同一被験者に複数のオブザベーションを持つデータセットを作成し，共変量値の被験者内変化を可能とします．時間の区切り方はユーザーが指定します．

時間依存性共変量をモデル化するときの最も一般的な時間の区切り方は，データ内のすべてのイベント時間を時間区切点としたベクトル形式です．addictsデータセットの変数SURVTには，各被験者のイベントまでの時間または打ち切りまでの時間が含まれています．以下の**survSplit**関数を用いたコードにより，addictsデータからCP形式の新たな解析データセット（**addicts.cp**）を作成します．

addicts.cp=survsplit(addicts,cut=addicts$survt[addicts$status==1], end="survt", event="status",start="start",id="id")

survSplit関数の1番目の引数には，CP形式に変換するdataframe（addicts）を指定します．**cut=addicts$survt[addicts$status==1]**オプションは，変数STATUSが1のときの時間（SURVT）で区切るように指定しています（イベント時間は対象となるが打ち切り時間は無視）．**event="status"**オプションで，被験者のイベント状態（イベントor打ち切り）を示す変数にSTATUSを指定します．**start="start"**オプションは新しい変数STARTを作成します．データをCP（開始，終了）形式にするためには，各オブザベーションの開始時間を定義したこの新しい変数が必要となります．**end="survt"**オプションでSURVTを終了時間変数（time-to-event変数）に指定します．**id="id"**オプションは，ID変数値が各被験者を識別する変数であることを示します．**survSplit**関数は，被験者がそれぞれの時点でat riskであるかを示す複数のオブザベーションを作成します．238オブザベーションのadditcsデータセットから作成した**addicts.cp**データセットのオブザベーション数は18,708となります．（**nrow**関数を用いてコード**nrow(addicts.cp)**とすればオブザベーション数が分かります）．

変数DOSEに関しては比例ハザード仮定が成立しないので，DOSEと時間（SURVT）の自然対数との積を時間依存性共変量に用いると想定します．

もしデータセットが，下に示すようなイベント時間ごとに区切られたCP形式になっているならば，この時間依存性共変量は簡単に定義できます．

addicts.cp$logtdose=addicts.cp$dose*log(addicts.cp$survt)

これで，時間によって値が変化する新しい変数がデータセットに作成されました(LOGTDOSE = ln(T)*DOSE)．時間 = 35日にイベントを経験した1名の被験者(id = 106)についてデータセットを出力します．すべての変数を出力するのではなく，c関数を用いて特定の変数を指定します．

addicts.cp[addicts.cp$id==106,c('id','start','survt','status','dose','logtdose')]

```
 id   start survt status dose logtdose
106     0     7      0   40  77.83641
106     7    13      0   40 102.59797
106    13    17      0   40 113.32853
106    17    19      0   40 117.77756
106    19    26      0   40 130.32386
106    26    29      0   40 134.69183
106    29    30      0   40 136.04790
106    30    33      0   40 139.86030
106    33    35      1   40 142.21392
```

LOGTDOSEは時間依存性変数で，ハザードの増加に合わせるように時間とともに増加しています．変数SURVT列には，この被験者がイベントを経験した35日までの，additcsデータセットにおけるすべてのイベント時間が示されています．イベント発生時点はSTATUS = 1であり，イベント発生前の時点はSTATUS = 0となっています．次に，予測因子PRISON, DOSE, CLINICと時間依存性変数LOGTDOSEを含む拡張Coxモデルを実行します．

coxph(Surv(addicts.cp$start,addicts.cp$survt,addicts.cp$status) ~ prison + dose + clinic + logtdose + cluster(id),data=addicts.cp)

今度の**Surv**関数には3つの引数，開始変数(START)，終了変数(SURVT)，status変数(STATUS)があります．モデルformulaに含まれている項**cluster(ID)**は，同一被験者に複数のオブザベーション(クラスター)があり，係数推定値のロバスト標準誤差を求めることを示しています．ロバスト標準誤差は，被験者内オブザベーション間の非独立性を考慮するためのものです．このモデルの出力は以下の通りです．

```
            coef exp(coef) se(coef) robust se      z       p
prison   0.34063     1.406  0.16747   0.15972  2.13 3.3e-02
dose    -0.08262     0.921  0.03598   0.02960 -2.79 5.3e-03
clinic  -1.01988     0.361  0.21542   0.23637 -4.31 1.6e-05
logtdose 0.00862     1.009  0.00645   0.00525  1.64 1.0e-01

Likelihood ratio test=66.3  on 4 df, p=1.34e-13  n= 18708, number of events= 150
```

LOGTDOSE における Wald 検定の z 統計量 1.64（*p* 値 = 1.0e-01 または *p* 値 = 0.10）は有意ではなく，DOSE に関する比例ハザード仮定を否定する証拠はありませんでした．

次に，365 日で区切られた Heaviside の階段関数と CLINIC を用いた時間依存性変数による拡張 Cox モデルを実行します．先に作成した **addicts.cp** データセットを使用することもできますが，用いる階段関数には時間区切点が 1 つしかないので，時間区切点が 1 つしかない CP 形式のデータセットを作成する方法について説明します．前に作成した **addicts.cp** データセットのオブザベーション数が 18,708 もあったのに対して，新しいデータセット（**addicts.cp365**）のオブザベーション数は 360 です．コードは以下の通りです．

addicts.cp365=survSplit(addicts,cut=365,end="survt",
event="status",start="start",id="id")

survSplit 関数の **cut = 365** オプションは，365 日を唯一の時間区切点に指定するものです．次に，2 つの時間依存性変数（HV1 と HV2）を作成します．HV1 は，生存時間が 365 日未満の場合は CLINIC の値を，それ以外は 0 の値を取ると定義します．HV2 は，生存時間が 365 日未満の場合は 0 の値を，生存時間が 365 日以上の場合は CLINIC の値を取ると定義します（コードは以下の通り）．

addicts.cp365$hv1=addicts.cp365$clinic*(addicts.cp365$start<365)
addicts.cp365$hv2=addicts.cp365$clinic*(addicts.cp365$start>=365)

コード内の条件文（**addicts.cp365$start<365**）と（**addicts.cp365$start>=365**）は，真であれば値 1 をとり，偽であれば値 0 をとります．これに変数 CLINIC を掛けて HV1 と HV2 を定義します．

次に，データセットを変数 ID と START でソートします．これは必須の作業ではありませんが，被験者順，時間順にオブザベーションを表示すると，データの確認がしやすくなります．**order** 関数でデータセットをソートします．

addicts.cp365=addicts.cp365[order(addicts.cp365$id,addicts.cp365$start),]

次のページに変数を選んで最初の 10 オブザベーションを出力します．

addicts.cp365[1:10,c(‘id’,‘start’,‘survt’,‘status’,‘clinic’,‘hv1’,‘hv2’)]

```
   id start survt status clinic hv1 hv2
  1     0   365      0      1   1   0
  1   365   428      1      1   0   1
 10     0   365      0      1   1   0
 10   365   393      1      1   0   1
100     0   146      0      2   2   0
101     0   365      0      2   2   0
101   365   450      1      2   0   2
102     0   365      0      2   2   0
102   365   555      0      2   0   2
103     0   365      0      2   2   0
```

変数IDのソート順は1，2，3ではなく，1，10，100となっています．変数IDは数値変数ではなく文字変数であるため，数値順ではなく「アルファベット順」でソートされます．最初の被験者（ID = 1）は428日にイベントがあるので，最初の時間区間（0，365）は打ち切り（STATUS = 0）となり，2番目の区間（365，428）でイベント（STATUS = 1）となっています．この被験者はCLINIC = 1であるため，最初の区間の時間依存性変数値はHV1 = 1，HV2 = 0となり，2番目の区間ではHV1 = 0，HV2 = 1となります．

これらのHeavisideの階段関数を含む拡張Coxモデルを実行する前に，**Surv**関数を用いて応答変数のオブジェクト（**Y365**）を定義します．このオブジェクトは後で**coxph**のモデルformulaに用います．実はこのオブジェクトの定義は必ずしも必要なく，前にLOGTDOSEを含む拡張Coxモデルを実行したときにはそうしませんでした．しかし，これを用いると応答変数の記述が簡明になり，見やすいコードになります．定義するコードは以下の通りです．

Y365=Surv(addicts.cp365$start,addicts.cp365$survt, addicts.cp365$status)

次に，2つのHeavisideの階段関数を含むモデルを実行します（コードと出力は以下の通り）．

coxph(Y365 ~ prison + dose + hv1 + hv2 + cluster(id), data=addicts.cp365)

```
           coef exp(coef) se(coef) robust se     z       p
prison  0.3780     1.459  0.16841   0.16765  2.25 2.4e-02
dose   -0.0355     0.965  0.00643   0.00652 -5.44 5.3e-08
hv1    -0.4594     0.632  0.25529   0.25998 -1.77 7.7e-02
hv2    -1.8305     0.160  0.38595   0.39838 -4.59 4.3e-06

Likelihood ratio test=74.2  on 4 df, p=2.89e-15  n= 360, number of events= 150
```

ハザード比推定値（CLINIC = 2 vs. CLINIC = 1）は，365日未満では0.632であり，365日以上では0.160です（2番目の数値列「exp（coef）」）．SAS，Stata，SPSSの出力と同じ結果が欲しい場合は，ロバスト標準誤差の指定を外し，イベント同順位のCox尤度の処理を**method="breslow"**に指定してモデルを実行します．コードは以下の通りです（出力は省略）．

coxph（Y365~ prison + dose + hv1 + hv2,data=addicts. cp365, method="breslow"）

Heavisideの階段関数が1つで変数CLINICをモデルに含む，上記と同等のモデルを実行することができます（コードと出力は以下の通り）．

coxph（Y365 ~ prison + dose + clinic + hv2 + cluster（id）, data=addicts.cp365）

```
          coef exp(coef) se(coef) robust se      z       p
prison  0.3780     1.459  0.16841   0.16765  2.25 2.4e-02
dose   -0.0355     0.965  0.00643   0.00652 -5.44 5.3e-08
clinic -0.4594     0.632  0.25529   0.25998 -1.77 7.7e-02
hv2    -1.3711     0.254  0.46140   0.47054 -2.91 3.6e-03

Likelihood ratio test=74.2  on 4 df, p=2.89e-15  n= 360, number of events= 150
```

係数推定値は，このモデルと2つのHeavisideの階段関数モデルとでは異なりますが，ハザード比推定値は同じです．ハザード比推定値（CLINIC = 2 vs.0 CLINIC = 1）は，365日未満では0.632です（CLINICの係数を指数化）．365日以上でのハザード比を推定するためには，CLINICとHV2の係数推定値の合計を求め，指数化します（exp（－0.4594 + －1.3711））= 0.160）．HV2の係数推定値のp値（p = 3.6e-10 or p = 0.0036）は有意であり，CLINICに関する2つの異なる時間区間のハザード比は等しくないことが示唆されます．言い換えれば，有意なp値は，CLINICに関しては比例ハザード仮定が成立しない証拠となります．

8. パラメトリックモデルの実行

Rでは，**survreg**関数でパラメトリック加速モデル（AFT）を実行します．比例ハザード（PH）モデルの主要な仮定が「ハザード比は時間経過に関係なく一定」であるのに対して，AFTモデルの主要な仮定は，「共変量パターン間の生存時間は一定の係数比で加速する」というものです．

生存データのパラメトリックモデルに最も用いられる分布はWeibull分布です．Weibull分布のハザード関数はλpt^{p-1}です．p = 1の場合，Weibull分布は指数分布でもあります．Weibull分布には，AFT仮定が成立すればPH仮定も成立するという好ましい性質があります．指数分布はWeibull分布の特別な場合です．指数分布の重要な性質は，「ハザードは時間にかかわらず一定」というものです（$h(t) = \lambda$）．Rでは，Weibullモデルと指数モデルは加速モデルだけが実行可能です．

Weibull分布には，対数（－対数（生存関数））が対数時間と直線関係にあるという性質があります．セクション2（グラフを用いた方法による比例ハザード性の評価）では，**plot**関数の**fun="cloglog"**オプションで，変数CLINICに関してKaplan-Meier対数（－対数）生存率と時間（対数尺度で）のプロットを作成しました．このプロットの曲線でWeibull仮定が評価できます．これらの生存曲線がほぼ直線（かつ平行）であれば，CLINICに関するWeibull仮定は妥当と判断できます．さらに，これらの直線の傾きが1であれば，指数分布が当てはまります．セクション2に示したコードの抜粋を再掲します（セクション2の出力図参照）．

plot(survfit(Y~addicts$clinic), fun="cloglog",xlab="time in days using logarithmic scale",ylab="log-log survival", main="log-log curves by clinic")

セクション2の対数（－対数）曲線は直線には見えませんが，ここではWeibull仮定が成立するとして話を進めます．まず，**survreg**関数を用いて指数モデルを実行します．このモデルは，Weibull形状パラメータ（p）を1に固定，つまり，ハザードを一定としたものです．結果をオブジェクト**modpar1**に保存します．

modpar1=survreg(Surv(addicts$survt,addicts$status) ~ prison + dose + clinic,data=addicts,dist="exponential")

次に，今作成したオブジェクトにsummary関数を適用します（コードと出力は以下の通り）．

summary（modpar1）

```
               Value Std. Error          z           p
(Intercept)    5.215      0.262      19.93     2.40e-88
clinic         0.975      0.210       4.65     3.24e-06

Scale fixed at 1

Exponential distribution
Loglik(model)= -1105.9   Loglik(intercept only)= -1118.9
      Chisq= 26.02 on 1 degrees of freedom, p= 3.4e-07
Number of Newton-Raphson Iterations: 5
n= 238
```

指数モデルの主要な仮定は，「ハザードは時間にかかわらず一定」というものです．この出力のパラメータ推定値表の下に示された“Scale fixed at 1”という記述はこのことを示しています．この出力を用いれば，特定の共変量パターン間のハザード比を計算できます．Rは指数モデルのパラメータ推定値をAFT形式で出力しています．ゆえに，係数推定値に－1を掛けると，このモデルのPHパラメータ推定値が得られます（第7章を参照）．例えば，PRISON = 1 vs. PRISON = 0のハザード比推定値はexp（0.2526）= 1.29です．指数モデルにおいて，対応する加速係数はハザード比の逆数，exp（－0.2526）= 0.78です．服役歴がある人は，イベントまでの時間が0.78倍になります．

次に，survreg関数を用いてWeibull加速モデルを実行します．結果をオブジェクト**modpar2**に保存します．

modpar2=survreg（Surv（addicts$survt,addicts$status）
~ prison + dose + clinic,data=addicts,dist=“weibull”）

次に，summary関数をオブジェクト**modpar2**に適用します（コードと出力は以下の通り）．

```
summary(modpar2)
              Value Std. Error     z        p
(Intercept)  4.1048    0.32806 12.51 6.37e-36
prison      -0.2295    0.12079 -1.90 5.75e-02
dose         0.0244    0.00459  5.32 1.03e-07
clinic       0.7090    0.15722  4.51 6.49e-06
Log(scale)  -0.3150    0.06756 -4.66 3.13e-06

Scale= 0.73

Weibull distribution
Loglik(model)= -1084.5   Loglik(intercept only)= -1114.9
        Chisq= 60.89 on 3 degrees of freedom, p= 3.8e-13
Number of Newton-Raphson Iterations: 7
n= 238
```

Weibull形状パラメータは，Rでいうスケールパラメータ（0.73と推定）の逆数です．ゆえに，Weibull形状パラメータの推定値は，逆数をとり1/0.73 = 1.37となります．CLINIC = 2とCLINIC = 1を比較する加速係数はexp（0.7090）= 2.03と推定されます．ゆえに，CLINIC = 2の患者のメディアン生存時間推定値（ヘロインを使用しない期間）は，CLINIC = 1の患者の2倍となっています．

モデルからの結果と**predict**関数を用いて，特定の共変量パターンに関するイベントまでのメディアン（あるいは任意の分位点）時間を推定することができます．例えば，共変量パターンPRISON = 1，DOSE = 50，CLINIC = 1を持つ被験者の，25，50，75パーセンタイル生存時間を，前述のWeibullモデルの結果を保存したオブジェクト**modpar2**から求めます．コードは以下の通りです．

```
pattern1=data.frame(prison=1,dose=50,clinic=1)
pct=c(.25,.50,.75)
days=predict(modpar2,newdata=pattern1,type="quantile",p=pct)
cbind(pct,days)
```

このコードの1行目の記述は，興味のある共変量パターンを指定する1オブザベーションのdataframeを作成します．もし異なる共変量パターンを比較したいのなら，このdataframe（**pattern1**）に複数のオブザベーションを含めます．2行目の記述は，興味あるパーセンタイル（25，50，75）からなるベクトル（**pct**）を作成します．3行目の記述は，**predict**関数からの出力を格納するオブジェクト（**days**）を作成します．**predict**関数の1番目の引数は，Weibullモデルの結果を格納しているオブジェクト**modpar2**です．

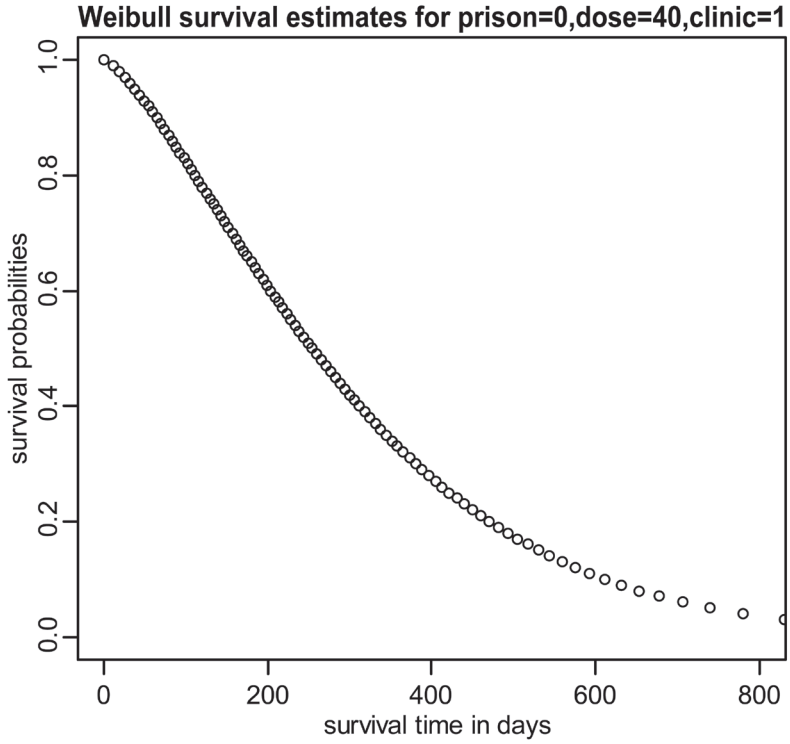
2番目の引数**newdata=pattern1**は，検討する共変量パターンの指定です．3番目の引数**type=“quantile”**は分位数の出力を指定します．4番目の引数**p=pct**は，このコードの上の行で作成した分位数ベクトルを指定します．このコードの最終行の記述は，**cbind**関数を使用してベクトル**pct**とベクトル**days**を組み合わせ，隣同士の列にします．出力は以下の通りです．

```
      pct    days
[1,]  0.25   133.8074
[2,]  0.50   254.2196
[3,]  0.75   421.6070
```

メディアン生存時間推定値は254.2196日です．同様のコードを用いて，Weibullモデルの結果から共変量パターンPRISON = 1，DOSE = 50，CLINIC = 1を持つ被験者の生存曲線をプロットします．コードは以下の通りです．

```
pct2=0:100/100
days2=predict(modpar2,newdata=pattern1,type="quantile",p=pct2)
survival=1-pct2

plot(days2,survival,xlab="survival time in days",ylab= "survival
probabilities",main="Weibull survival estimates for prison=0,
dose=40,clinic=1",xlim=c(0,800))
```

1行目の記述は，0から1まで0.01刻み（0，0.01，0.02，・・・，0.99，1）の順列パーセンタイルを格納したベクトル**pct2**を作成します．2行目の記述は，**predict**関数からの出力を格納するオブジェクト**days2**を作成します．3行目は，**pct2**のデータ順を逆にしたベクトル**survival**を作成します．最後に，**plot**関数で，ベクトル**days2**を横軸に，ベクトル**survival**を縦軸にとりプロットします．軸ラベルとタイトルは**plot**関数のオプションで指定します．出力図は次ページです．



次は，**survreg**関数を用いて対数ロジスティックAFTモデルを実行します．結果をオブジェクト**modpar3**に保存します．

modpar3=survreg(Surv(addicts$survt,addicts$status)~
prison + dose + clinic,data=addicts,dist="loglogistic")

次に，summary関数をオブジェクト**modpar3**に適用します(コードと出力は以下の通り)．

summary(modpar3)

```
              Value Std. Error     z        p
(Intercept)  3.5633     0.38945  9.15 5.72e-20
prison      -0.2913     0.14396 -2.02 4.31e-02
dose         0.0316     0.00552  5.73 1.02e-08
clinic       0.5806     0.17157  3.38 7.14e-04
Log(scale)  -0.5331     0.06863 -7.77 7.95e-15

Scale= 0.587

Log logistic distribution
Loglik(model)= -1093.9   Loglik(intercept only)= -1120
        Chisq= 52.18 on 3 degrees of freedom, p= 2.7e-11
Number of Newton-Raphson Iterations: 4
n= 238
```

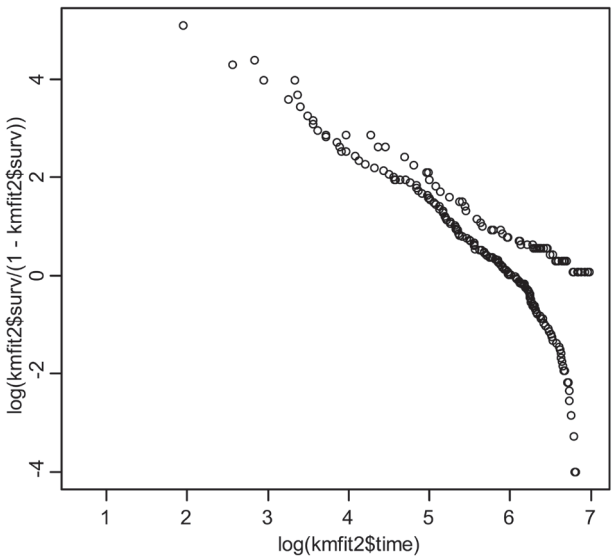
この出力から，CLINIC = 2とCLINIC = 1を比較した加速係数はexp（0.5806）= 1.79と推定されます．対数ロジスティックモデルでAFT仮定が成立するならば，生存関数に関して比例オッズ仮定が成立します（ただし，比例ハザード仮定は成り立たない）．対数生存オッズ（KM推定値を使用）と対数生存時間をプロットすることにより，比例オッズ仮定を評価できます．それぞれの共変量パターンにおけるプロットが直線的であれば，対数ロジスティック分布は適合しています．直線かつ平行であれば，比例オッズ仮定に加えてAFT仮定も成り立ちます．

セクション2でKaplan-Meier生存推定値を格納するオブジェクト**kmfit2**を作成しましたが，このオブジェクトを作成するコードを再掲します．

```
kmfit2=survfit(Surv(addicts$survt,addicts$status)～addicts$clinic)
```

ベクトル**kmfit2$time**には生存時間が，ベクトル**kmfit2$surv**にはCLINIC別のKM生存推定値が格納されています．**plot**関数を用いて対数生存オッズlog[(*S*/(1 × *S*))と対数生存時間をプロットします．コードと出力は以下の通りです．

```
plot(log(kmfit2$time),log(kmfit2$surv/(1-kmfit2$surv)))
```



これらの曲線は直線的でも平行的でもないため，CLINICに関する比例オッズ仮定は成立しないと思われます．前述では説明目的であえてこの対数ロジスティックモデルを実行しましたが，このグラフからは，対数ロジスティックモデルは適切ではないことが示唆されます．

survreg関数はこの他にも，正規（dist="gaussian"），対数正規（dist="log-normal"）の各分布に対応しています．

9. frailtyモデルの実行

frailtyモデルは，モデルでは説明できない個人レベルのハザードの違いを考慮するための追加的なランダム成分を含みます．Frailty αは，ある分布に従うと仮定したハザードへの相乗効果です．Frailtyを用いた条件付きハザード関数は，$h(t|\alpha) = \alpha[h(t)]$と表すことができます．

Rでは，frailtyの分布に，ガンマ分布，ガウス分布，t分布の3つを準備しています．frailty成分の分散(θ)が，モデルから推定するパラメータとなっています．θ = 0ならばfrailtyは存在しません．

最初に，frailtyを持たない層化Coxモデルに戻ります(セクション4参照)．層化変数はCLINICで，PRISONとDOSEは説明変数です．CLINICについてはPH仮定が成立せず，PRISONとDOSEでは成立するならば，層化Coxモデルは適切です．また，われわれの興味はPRISONあるいはDOSEのハザード比を推定することです．コードと出力は以下の通りです．

```
Y=Surv(addicts$survt,addicts$status==1)
coxph(Y~ prison + dose + strata(clinic),data=addicts)
```

```
          coef exp(coef) se(coef)      z       p
prison  0.3896     1.476  0.16893  2.31 2.1e-02
dose   -0.0351     0.965  0.00646 -5.43 5.6e-08

Likelihood ratio test=33.9  on 2 df, p=4.32e-08  n= 238, number of events= 150
```

PRISON = 1 vs. PRISON = 0のハザード比推定値はexp(0.3896) = 1.476です．次に，frailty成分をこのモデルに入れる方法について説明します．コードは以下の通りです．

```
coxph(Y~ prison + dose + strata(clinic) + frailty(id,
distribution="gamma"), data=addicts)
```

モデルformulaに，項 + **frailty(id, distribution="gamma")**が入っています．**frailty**関数の1番目の引数は変数**id**であり，測定できない不均一性(frailty)が個人レベルで存在することを示しています．2番目の引数は，ランダム成分の分布がガンマ分布であることを示しています．出力を次ページに示します．

```
                              coef    se(coef) se2     Chisq DF   p
prison                         0.3900 0.16916  0.16893  5.32 1.00 2.1e-02
dose                          -0.0352 0.00647  0.00647 29.51 1.00 5.6e-08
frailty(id, distribution                                0.34 0.32 3.1e-01

Iterations: 5 outer, 41 Newton-Raphson
     Variance of random effect= 0.00227   I-likelihood = -597.5
Degrees of freedom for terms= 1.0 1.0 0.3
Likelihood ratio test=34.6  on 2.32 df, p=5.26e-08  n= 238
```

パラメータ推定値の出力表の下に，ランダム効果の分散 = 0.00227 と示されています．表の3行目右列にfrailty成分の p 値3.1e − 01 = 0.31が示されていて，frailty成分は有意ではないことがわかります．結論として，このモデルに関してはランダム成分の分散は0と判断します（つまりfrailtyは存在しない）．PRISONとDOSEのパラメータ推定値をfrailtyを持たないモデルと比べても，違いはほとんどありません．

今度は，仮に変数CLINICが観測されていない場合を想定します．CLINICを含まないCoxモデル（frailtyがない場合）のコードと出力は以下の通りです．

coxph（Y~ prison + dose, data=addicts）

```
          coef   exp(coef) se(coef)    z     p
prison  0.1897     1.209    0.164   1.15 2.5e-01
dose   -0.0361     0.965    0.006  -6.01 1.8e-09
Likelihood ratio test=38.2  on 2 df, p=5.04e-09  n= 238, number of events= 150
```

PRISON = 1 vs. PRISON = 0のハザード比推定値はexp（0.1897）= 1.209です．CLINICを層化変数とした場合のモデルではexp（0.3896）= 1.476でした．以前のセクションで，CLINICが比例ハザード仮定を阻害する重要な予測因子であることを示しました．もしCLINICを考慮しなければ（上記のモデルのように），それが観察されない不均一性となり，frailty成分として現れる可能性があります．次のモデルではCLINICを外し，frailty成分と説明変数PRISONとDOSEはそのまま残しています．コードと出力は以下の通りです．

coxph(Y~ prison + dose + frailty(id, distribution="gamma"), data=addicts)

```
                          coef    se(coef) se2     Chisq DF   p
prison                     0.4144 0.22160  0.17590   3.5  1.0 6.1e-02
dose                      -0.0517 0.00845  0.00699  37.4  1.0 9.6e-10
frailty(id, distribution                            100.5 69.3 8.6e-03

Iterations: 6 outer, 44 Newton-Raphson
     Variance of random effect= 0.65   I-likelihood = -685.4
Degrees of freedom for terms=  0.6  0.7 69.3
Likelihood ratio test=190  on 70.7 df, p=6.17e-13  n= 238
```

frailty成分の分散は0.65と推定され，CLINICを層化変数に含む前モデルの0.00227と比べると大きくなっています．frailtyのp値は8.6e − 03 = 0.0086と非常に有意です．PRISON効果のハザード比はexp(0.4144) = 1.51です．frailty成分をCoxモデルに含むときに，Rのパラメータ推定値(95%信頼区間も)の指数化を行うために，**summary**関数を**coxph**関数に適用します．コードと出力は以下の通りです．

summary(coxph(Y~ prison + dose + frailty(id, distribution="gamma"), data=addicts))

```
                          coef       se(coef)   se2     Chisq   DF     p
prison                     0.41441  0.221604 0.17590    3.5   1.00  6.1e-02
dose                      -0.05166  0.008448 0.00699   37.4   1.00  9.6e-10
frailty(id, distribution                              100.5  69.34  8.6e-03

        exp(coef)  exp(-coef) lower .95 upper .95
prison      1.5135     0.6607     0.9803     2.3367
dose        0.9496     1.0530     0.9341     0.9655

Iterations: 6 outer, 44 Newton-Raphson
     Variance of random effect= 0.6495364   I-likelihood = -685.4
Degrees of freedom for terms=  0.6  0.7 69.3
Concordance= 0.854  (se = 0.026 )
Likelihood ratio test= 190.4  on 70.65 df,   p=6.172e-13
```

興味深いことに，このモデル(CLINICなし，frailty成分あり)で得られたPRISONのハザード比推定値(1.51)は，モデル(CLINICなし，frailty成分なし)でのハザード比推定値(1.209)よりも，モデル(CLINICあり，frailty成分なし)でのハザード比推定値(1.476)に近いことがわかります．この例では，CLINICをモデルから除いた影響をfrailty成分が代わりに説明している可能性があります．

10. 再発イベントのモデル構築

再発イベントのモデル構築については，このAppendixの冒頭に記述した膀胱がんデータセット(bladder.rda)を用いて説明します．複数のイベントを経験した被験者の再発イベントは，データ内で対応する複数のオブザベーションで表されます. 膀胱がんデータセットのデータレイアウトはCP形式(開始，終了)であり，時間区間がオブザベーションごとに設定されています(第8章を参照). ファイルとして保存されているRのdataframeにアクセスするには，**load**関数を用います．bladderデータセットがCドライブにC:\bladder.rdaとして保存されているとすると，以下のコードでbladderデータを読み込みます．

```
load("C://bladder.rda")
```

以下のコードで4被験者の情報，12〜20番目のオブザベーションを出力します．

```
bladder[12:20, ]
```

以下に出力を示します．

```
    ID  EVENT  INTERVAL  INTTIME START STOP  TX  NUM  SIZE
12  10      1         1       12     0   12   0    1     1
13  10      1         2        4    12   16   0    1     1
14  10      0         3        2    16   18   0    1     1
15  11      0         1       23     0   23   0    3     3
16  12      1         1       10     0   10   0    1     3
17  12      1         2        5    10   15   0    1     3
18  12      0         3        8    15   23   0    1     3
19  13      1         1        3     0    3   0    1     1
20  13      1         2       13     3   16   0    1     1
```

ID = 10の被験者には3オブザベーション，ID = 11には1オブザベーション，ID = 12には3オブザベーション，ID = 13には2オブザベーションあります．変数STARTとSTOPは, そのオブザベーションに該当するリスク期間を表す時間区間です．変数EVENTは，イベントの有(code = 1)，無を示します．最初の3オブザベーションは, ID = 10の被験者に12ヵ月にイベントがあり，16ヵ月にまた別のイベントがあり，18ヵ月に打ち切りとなったことを示しています．

拡張Coxモデルの実行(セクション7)でもCP形式のデータを解析しました．拡張Coxモデルでは，時間区間の変化により被験者の共変量値が変わりました．bladderデータセットでは，(開始，終了)データ形式は，被験者の複数のイベントを示す方法となっています．

Rに関する最初の説明で述べたように，Rの生存関数にアクセスするには，Rの立ち上げ時にコード**library(survival)**を実行する必要があります．

```
library(survival)
```

coxph関数を用いて，再発イベントCoxモデルを実行することができます．まず，**Surv**関数を用いて応答変数(**Y**)を定義します．

```
Y=Surv(bladder$START,bladder$STOP,bladder$EVENT==1)
```

セクション7で示したように，**Surv**関数はCP形式のデータでは3つの引数が必要となり，開始変数(START)，終了変数(STOP)，status変数(EVENT)です．コード**bladder$EVENT==1**は，イベントが1であることを示しています．Rでのイベントのデフォルト値は1であるため，ここで示したように**Surv**関数で指定する必要は実際にはありません．次に，3つの説明変数，治療(TX)，最初の腫瘍数(NUM)，最初の腫瘍サイズ(SIZE)を持つ再発イベントCoxモデルを実行します．

```
coxph(Y~ TX + NUM + SIZE + cluster(ID), data=bladder)
```

モデルformulaの項 + **cluster(ID)**は，パラメータ推定値のロバスト標準誤差を指定します．このモデルの出力は以下の通りです．

```
        coef exp(coef) se(coef) robust se      z      p
TX    -0.4116     0.663   0.1999    0.2488 -1.655 0.0980
NUM    0.1637     1.178   0.0478    0.0584  2.801 0.0051
SIZE  -0.0411     0.960   0.0703    0.0742 -0.554 0.5800

Likelihood ratio test=14.7  on 3 df, p=0.00213  n= 190, number of events= 112
```

治療変数(TX)は，チオテパ治療 = 1，プラセボ = 0とコードしています．ハザード比推定値(TX = 1 vs. TX = 0)は0.663です(p値は0.0980)．出力表には2組の標準誤差，se(coef)列とrobust se列があります．この表のp値とz-検定統計量は，ロバスト標準誤差を用いて計算したものです．**summary**関数を**coxph**関数に適用すれば，追加出力(95%信頼区間など)が得られます．

この形式のデータを用いて，変数INTERVALを層化変数とする層化Coxモデルを実行することもできます．層化変数は，被験者が1番目，2番目，3番目，4番目のイベントでそれぞれat riskであるかを示します．

この方法を層化CP再発イベントモデル（第8章を参照）と呼び，再発イベントを発生順序で区別したい場合に用います．bladderデータは，このモデルを実行するための正しい形式になっています．コードと出力は以下の通りです．

coxph（Y~ TX + NUM + SIZE + strata（INTERVAL） + cluster（ID）,data=bladder）

```
        coef exp(coef) se(coef) robust se      z    p
TX   -0.33349    0.716   0.2162    0.2048 -1.628 0.10
NUM   0.11962    1.127   0.0533    0.0514  2.328 0.02
SIZE -0.00849    0.992   0.0728    0.0616 -0.138 0.89

Likelihood ratio test=6.51  on 3 df, p=0.0893  n= 190, number of events= 112
```

前のモデルとの違いは，モデルformulaに項 + **strata（INTERVAL）**が追加されただけです．このコードはINTERVALが層化変数であることを示します．治療変数（TX）と層化変数との交互作用項を作成して，治療効果が1番目，2番目，3番目，4番目のイベントで異なるかを調べることもできます．

別の層化アプローチ（Gap Time）は層化CPアプローチをわずかに変えたものです．その違いは再発イベントの時間区間の定義方法にあります．被験者の最初のイベントのat riskに関しては，時間区間に違いはありません．しかしながら，Gap Timeアプローチでは，次のイベントからはat risk開始時間がそれぞれ0にリセットされます．Gap Timeモデルを実行するには，bladderデータセットに2つの新しい（開始，終了）変数を作成する必要があり，これらをSTART2，STOP2とします．

bladder$START2=0
bladder$STOP2=bladder$STOP – bladder$START

新たに定義した2つの変数のうち，1つ目（START2）はすべて0です．2つ目（STOP2）は，イベント間の時間（STOP–START）と定義されます．**data.frame**関数を使用して，これらの変数のいくつかを出力します．**attach**関数は，bladderデータセットの変数を**bladder$**表記なしで指定できます．（12～20番目のオブザベーションを出力するコードと出力は以下の通り）．

attach（bladder）
data.frame（ID,EVENT,START,STOP,START2,STOP2）[12:20,]

```
      id  event start stop start2 stop2
12    10      1     0   12      0    12
13    10      1    12   16      0     4
14    10      0    16   18      0     2
15    11      0     0   23      0    23
16    12      1     0   10      0    10
17    12      1    10   15      0     5
18    12      0    15   23      0     8
19    13      1     0    3      0     3
20    13      1     3   16      0    13
```

次に，**Surv**関数を用いて応答変数をリセットし，時間区間を（START，STOP）から（START2，STOP2）に変更します．

Y2=Surv（bladder$START2,bladder$STOP2,bladder$EVENT）

次に，層化CPモデルに用いたコードの応答変数**Y**を**Y2**に置き換えたものを用いて，bladderデータでのGap Timeモデルを実行します．コードと出力は以下の通りです．

coxph（Y2~ TX + NUM + SIZE + strata（INTERVAL）+ cluster（ID）,data=bladder）

```
         coef exp(coef) se(coef) robust se      z      p
TX   -0.27900     0.757   0.2073    0.2156 -1.294 0.2000
NUM   0.15805     1.171   0.0519    0.0509  3.103 0.0019
SIZE  0.00742     1.007   0.0700    0.0643  0.115 0.9100

Likelihood ratio test=9.33  on 3 df, p=0.0252  n= 190, number of events= 112
```

Gap Timeアプローチによる結果は，層化CPアプローチの結果とわずかに異なります．