Computer Appendix: Survival Analysis on the Computer

D.G. Kleinbaum and M.Klein, *Survival Analysis: A Self-Learning Text, Third Edition*,
Statistics for Biology and Health, DOI 10.1007/978-1-4419-6646-9,

In this appendix, we provide examples of computer programs for carrying out the survival analyses described in this text. This appendix does not give an exhaustive survey of all computer packages currently available, but rather is intended to describe the similarities and differences among four of the most widely used packages. The software packages that we describe are Stata (version 10.0), SAS (version 9.2), SPSS (PASW 18), and R. A complete description of these packages is beyond the scope of this appendix. Readers are referred to the built-in help functions for each program for further information.

Datasets

Most of the computer syntax and output presented in this appendix are obtained from running step-by-step survival analyses on the "addicts" dataset. The other dataset that is utilized in this appendix is the "bladder cancer" dataset for analyses of recurrent events. The "addicts" and "bladder cancer" data are described below and can be downloaded from our website at **http://www.sph.emory.edu/dkleinb/surv3.htm**. On this website, we also provide many of the other datasets that have been used in the examples and exercises throughout this text. The data on our website are provided in five forms (1) as Stata datasets (with a **.dta** extension), (2) as SAS datasets (with a **.sas7bdat** extension), (3) as SPSS datasets (with a **.sav** extension), (4) as R datasets (with an **.rda** extension), and (5) as text datasets (with a **.dat** extension).

Addicts Dataset (addicts.dat)

In a 1991 Australian study by Caplehorn et al., two methadone treatment clinics for heroin addicts were compared to assess patient time remaining under methadone treatment. A patient's survival time was determined as the time (in days) until the person dropped out of the clinic or was censored. The two clinics differed according to its live-in policies for patients. The variables are defined as follows:

ID – Patient ID
SURVT – The time (in days) until the patient dropped out of the clinic or was censored
STATUS – Indicates whether the patient dropped out of the clinic (coded 1) or was censored (coded 0)
CLINIC – Indicates which methadone treatment clinic the patient attended (coded 1 or 2)
PRISON – Indicates whether the patient had a prison record (coded 1) or not (coded 0)
DOSE – A continuous variable for the patient's maximum methadone dose (mg/day)

Bladder Cancer Dataset (bladder.dat)

The bladder cancer dataset contains recurrent event outcome information for eighty-six cancer patients followed for the recurrence of bladder cancer tumor after transurethral surgical excision (Byar and Green 1980). The exposure of interest is the effect of the drug treatment of thiotepa. Control variables are the initial number and initial size of tumors. The data layout is suitable for a counting processes approach. The variables are defined as follows:

ID – Patient ID (may have multiple observations for the same subject)
EVENT – Indicates whether the patient had a tumor (coded 1) or not (coded 0)
INTERVAL – A counting number representing the order of the time interval for a given subject (coded 1 for the subject's first time interval, coded 2 for a subject's second time interval, etc.)
START – The starting time (in months) for each interval
STOP – The time of event (in months) or censorship for each interval
TX – Treatment status (coded 1 for treatment with thiotepa and 0 for the placebo)
NUM – The initial number of tumors
SIZE – The initial size (in centimeters) of the tumor

Software

What follows is a detailed explanation of the code and output necessary to perform the type of survival analyses described in this text. The rest of this appendix is divided into four broad sections, one for each of the following software packages:

A. Stata
B. SAS
C. SPSS
D. R Software

Each of these sections is self-contained, allowing the reader to focus on the particular statistical package of his or her interest.

A. Stata

Analyses using Stata are obtained by typing the appropriate statistical commands in the Stata Command window or in the Stata Do-file Editor window. The key commands used to perform the survival analyses are listed below. These commands are case sensitive and lower-case letters should be used.

stset – Declares data in memory to be survival data. Used to define the "time-to-event" variable, the "status" variable, and other relevant survival variables. Other Stata Commands beginning with **st** utilize these defined variables.

sts list – Produces Kaplan-Meier (KM) or Cox-adjusted survival estimates in the output window. The default is KM survival estimates.

sts graph – Produces plots of Kaplan-Meier (KM) survival estimates. This command can also be used to produce Cox-adjusted survival plots.

sts generate – Creates a variable in the working dataset that contains Kaplan-Meier or Cox adjusted survival estimates.

sts test – Used to perform statistical tests for the equality of survival functions across strata.

stphplot – Produces plots of log-log survival against the log of time for the assessment of the proportional hazards (PH) assumption. The user can request KM log-log survival plots or Cox adjusted log-log survival plots.

stcoxkm – Produces KM survival plots and Cox adjusted survival plots on the same graph.

stcox – Used to run a Cox proportional hazard model, a stratified Cox model, or an extended Cox model (i.e., containing time varying covariates).

stphtest – Performs statistical tests on the PH assumption based on Schoenfeld residuals. Use of this command requires that a Cox model be previously run with the command **stcox** and the **schoenfeld()** option.

streg – Used to run parametric survival models.

Four windows will appear when Stata is opened. These windows are labeled Stata Command, Stata Results, Review, and Variables. The user can click on File → Open to select a working dataset for analysis. Once a dataset is selected, the names of its variables appear in the Variables window. Commands are entered in the Stata Command window. The output generated by commands appears in the Results window after the return key is pressed. The Review window preserves a history of all the commands executed during the Stata session. The commands in the Review window can be saved, copied, or edited as the user desires. Command can also be run from the Review window by double-clicking on the command. Commands can also be saved in a file by clicking on the log button on the Stata tool bar.

Alternatively, commands can be typed, or pasted into the Do-file Editor. The Do-file Editor window is activated by clicking on Window → Do-file Editor or by simply clicking on the Do-file Editor button on the Stata tool bar. Commands are executed from the Do-file Editor by clicking on Tools → Do. The advantage of running commands from the Do-file Editor is that commands need not be entered

and executed one at a time as they do from the Stata Command window. The Do-file Editor serves a similar function as the program editor in SAS. In fact, by typing **#delim** in the Do-file Editor window, the semicolon becomes the delimiter for completing Stata statements (as in SAS) rather than the default carriage return.

The survival analyses demonstrated in Stata are as follows:

1. Estimating survival functions (unadjusted) and comparing them across strata
2. Assessing the PH assumption using graphical approaches
3. Running a Cox PH model
4. Running a stratified Cox model
5. Assessing the PH assumption with a statistical test
6. Obtaining Cox adjusted survival curves
7. Running an extended Cox model
8. Running parametric models
9. Running frailty models
10. Modeling recurrent events

The first step is to activate the addicts dataset by clicking on File → Open and selecting the Stata dataset, **addicts.dta**. Once this is accomplished, you will see the command **use "addicts.dta", clear** in the Review window and Results window. This indicates that the addicts dataset is activated in Stata's memory.

To perform survival analyses, you must indicate which variable is the "time-to-event" variable and which variable is the "status" variable. Rather than program this in every survival analysis command, Stata provides a way to program it once with the **stset** command. All survival commands beginning with **st** utilize the survival variables defined by **stset** as long as the dataset remains in active memory. The code to define the survival variables for the addicts data is as follows:

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

Following the word **stset** comes the name of the "time-to-event" variable. Options for Stata Commands follow a comma. The first option used is to define the variable and value that indicates an event (or failure) rather than a censorship. Without this option, Stata assumes that all observations had an event (i.e., no censorships). Notice that two equal signs are used to express equality. A single equal sign is used to designate assignment. The next option defines the id variable as the variable, ID. This is unnecessary with the addicts dataset since each observation

represents a different patient (cluster). However if there were multiple observations and multiple events for a single subject (cluster), Stata can provide robust variance estimates appropriate for clustered data.

The **stset** command will add four new variables to the dataset. Stata interprets these variables as follows:

_t – The "time-to-event" variable
_d – The "status variable" (coded 1 for an event and 0 for a censorship)
_t0 – The beginning "time variable." All observations start at time 0 by default
_st – Indicates which variables are used in the analysis. All observations are used (coded 1) by default

To see the first 10 observations printed in the output window, enter the command:

list in 1/10

The command **stdes** provides descriptive information (output below) of survival time.

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
-------------------------------------------------------------------------
```

The commands **strate** and **stir** can be used to obtain incident rate comparisons for different categories of specified variables. The **strate** command lists the incident rates by CLINIC while the **stir** command gives rate ratios and rate

differences. Type the following commands one at a time (output omitted):

strate clinic
stir clinic

For the survival analyses that follow, it is assumed that the command **stset** has been run for the addicts dataset, as demonstrated on the previous page.

1. ESTIMATING SURVIVAL FUNCTIONS (UNADJUSTED) AND COMPARING THEM ACROSS STRATA

To obtain Kaplan-Meier survival estimates use the command **sts list**. The code and output follow:

sts list

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

Time	Beg. Total	Fail	Net Lost	Survivor Function	Std. 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

If we wish to stratify by CLINIC and compare the survival estimates side-to-side for specified time points, we use the **by()** and **compare()** option. The code and output follow:

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             .          .
```

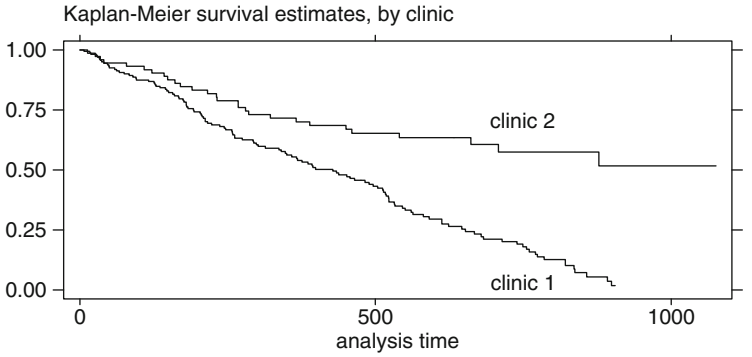
Notice that the survival rate for CLINIC=2 is higher than CLINIC=1. Other survival times could have been requested using the **compare()** option.

To graph the Kaplan-Meier survival function (against time), use the code:

sts graph

The code and output that provide a graph of the Kaplan-Meier survival function stratified by CLINIC follow:

sts graph, by(clinic)



The **failure** option graphs the failure function (the cumulative risk) rather than the survival (zero to one rather than one to zero). The code follows (output omitted):

sts graph, by(clinic) failure

The code to run the log rank test on the variable CLINIC (and output) follows:

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
```

The Wilcoxon, Tarone-Ware, Peto, and Flemington-Harrington tests can also be requested. These tests are variations of the log rank test that weight each observation differently. The Wilcoxon test weights the j^{th} failure time by n_i (the number still at risk). The Tarone-Ware test weights the j^{th} failure time by $\sqrt{n_j}$. The Peto test weights the j^{th} failure time by the survival estimate, $\tilde{s}(t_j)$ calculated over all groups combined. This survival estimate, $\tilde{s}(t_j)$, is similar but not exactly equal to the Kaplan-Meier survival estimate. The Flemington-Harington test uses the Kaplan-Meier survival estimate, $\hat{s}(t)$, over all groups to calculate its weights for the j^{th} failure time, $\hat{s}(t_{j-1})^p[1-\hat{s}(t_{j-1})]^q$, so it takes two arguments (p and q). The code follows (output omitted):

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

Notice that the default test for the **sts test** command is the log rank test. The choice of which weighting of the test statistic to use (e.g., log rank or Wilcoxon) depends on which test is believed to provide the greatest statistical power, which in turn depends on how it is believed the null hypothesis is violated. However, one should make an a priori decision on which statistical test to use rather than fish for a desired p-value.

A stratified log rank test for CLINIC (stratified by PRISON) can be run with the strata option. With the stratified approach, the observed minus expected number of events are summed over all failure times for each group within each stratum and then summed over all strata. The code follows (output omitted):

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

The **sts generate** command can be used to create a new variable in the working dataset containing the KM survival estimates. The following code defines a new variable called SKM (the variable name is the user's choice) that contains KM survival estimates stratified by CLINIC:

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

The **ltable** command produces life tables. Life tables are an alternative approach to Kaplan-Meier that are particularly useful if you do not have individual-level data. The code and output that follows provide life table survival estimates, stratified by CLINIC, at the time points (in days) specified by the **interval()** option:

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. ASSESSING THE PH ASSUMPTION USING GRAPHICAL APPROACHES

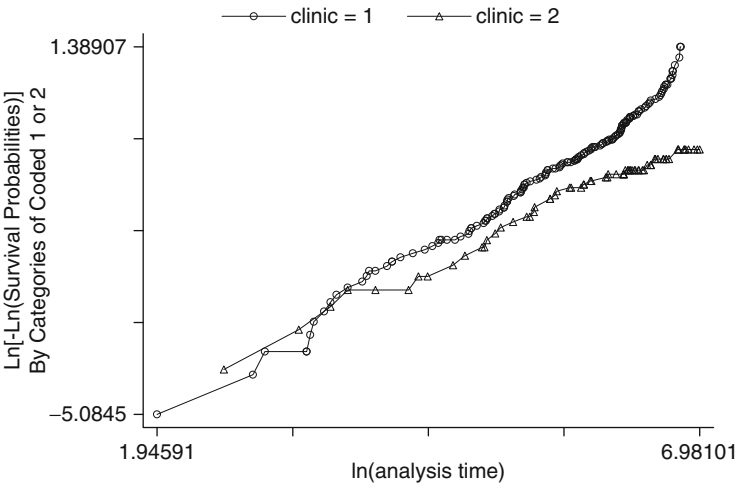
Three graphical approaches for the assessment of the PH assumption for the variable CLINIC are demonstrated:

1) Log-log Kaplan-Meier survival estimates (stratified by CLINIC) plotted against time (or against the log of time)
2) Log-log Cox adjusted survival estimates (stratified by CLINIC) plotted against time
3) Kaplan-Meier survival estimates and Cox adjusted survival estimates plotted on the same graph.

All three approaches are somewhat subjective yet hopefully informative. The first two approaches are based on whether the log log survival curves are parallel for different levels of CLINIC. The third approach is to determine if the Cox adjusted survival curve (not stratified) is close to the KM curve. In other words, are predicted values from the PH model (from Cox) close to the "observed" values using KM?

The first two approaches use the **stphplot** command while the third approach uses the **stcoxkm** command. The code and output for the log-log Kaplan-Meier survival plots follow:

stphplot, by(clinic) nonegative

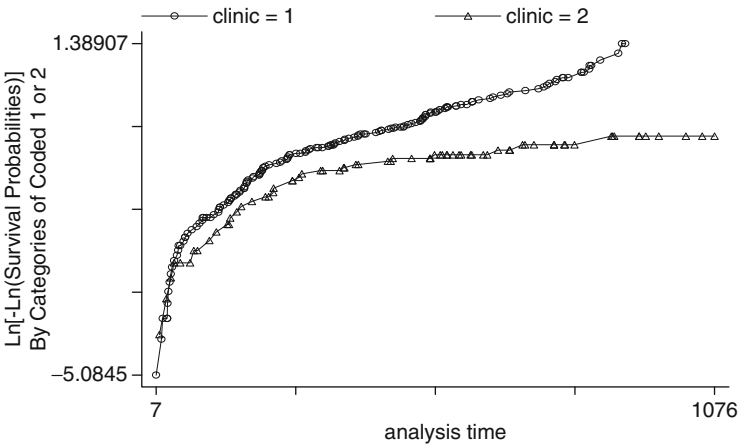


The left side of the graph seems jumpy for CLINIC=1 but it only represents a few events. It also looks like there is some separation between the plots at the later times (right side). The **nonegative** option in the code requests log(-log) curves rather than the default -log(-log) curves. The choice is arbitrary. Without the option, the curves would go downward rather than upward (left-to-right).

Stata (as well as SAS) plot log(survival time) rather than survival time on the horizontal axis by default. As far as checking the parallel assumption, it does not matter if log (survival time) or survival time is on the horizontal axis. However, if the log log survival curves look like straight lines with log(survival time) on the horizontal axis, then there is evidence that the "time-to-event" variable follows a Weibull distribution. If the slope of the line equals one, then there is evidence that the survival time variable (SURVT) follows an exponential distribution – a special case of the Weibull distribution. For these situations, a parametric survival model can be used.

It may be visually more informative to graph the log log survival curves against survival time (rather than log survival time). The **nolntime** option can be used to put survival time on the horizontal axis. The code and output follows:

stphplot, by(clinic) nonegative nolntime

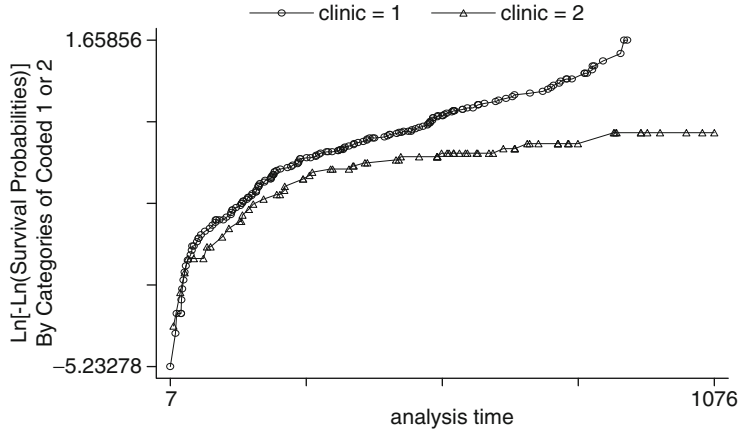


The graph suggests that the curves begin to diverge over time.

The **stphplot** command can also be used to obtain log-log Cox adjusted survival estimates. The code follows:

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

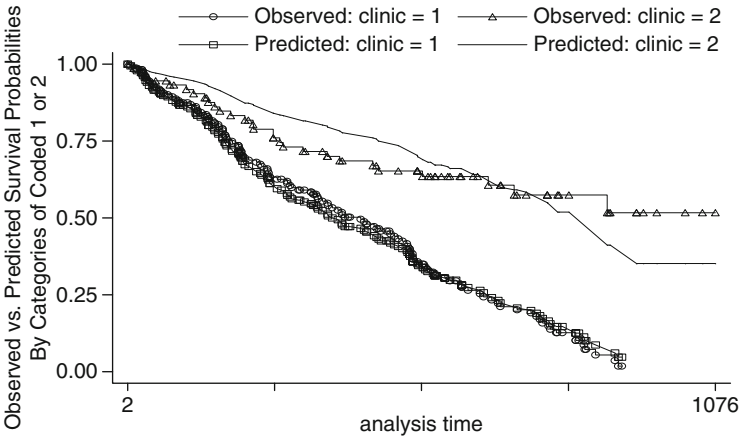
The log-log curves are adjusted for PRISON and DOSE using a stratified COX model on the variable CLINIC. The mean values of PRISON and DOSE are used for the adjustment. The output follows:



The Cox adjusted curves look very similar to the KM curves.

The **stcoxkm** command is used to compare Kaplan-Meier survival estimates and Cox adjusted survival estimates plotted on the same graph. The code and output follow:

stcoxkm, by(clinic)



The KM and adjusted survival curves are very close together for CLINIC=1 and less so for CLINIC=2. These graphical approaches suggest that there is some violation with the PH assumption. The predicted values are Cox adjusted for CLINIC, and therefore assume the PH assumption. Notice that the predicted survival curves are not parallel by CLINIC even though we are adjusting for CLINIC. It is the log-log survival curves, rather than the survival curves, that are forced to be parallel by Cox adjustment.

The same graphical analyses can be performed with PRISON and DOSE. However, DOSE would have to be categorized since it is a continuous variable.

3. RUNNING A COX PH MODEL

For a Cox PH model, the key assumption is that the hazard is proportional across different patterns of covariates. The first model that is demonstrated contains all three covariates: PRISON, DOSE, and CLINIC. In this model, we are assuming the same baseline hazard for all possible patterns of these covariates. In other words, we are accepting the PH assumption for each covariate (perhaps incorrectly). The code and output follow:

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
------------------------------------------------------------------------------
```

The output indicates that it took five iterations for the log likelihood to converge at −673.40242. The iteration history typically appears at the top of Stata model output; however, the iteration history will subsequently be omitted. The final table lists the regression coefficients, their standard errors, a Wald test statistic (z) for each covariate, with corresponding p-value, and 95% confidence interval.

The **nohr** option in the **stcox** command requests the regression coefficients rather than the default exponentiated coefficients (hazard ratios). If you want the exponentiated coefficients, omit the **nohr** option. The code and output follow:

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
------------------------------------------------------------------------------
```

This table contains the hazard ratios, its standard errors, and corresponding confidence intervals. Notice that you do not need to supply the "time-to event" variable or the status variable when using the **stcox** command. The **stcox** command uses the information supplied from the **stset** command. A Cox model can also be run using the **cox** command, which does not rely on the **stset** command having previously been run. The code follows:

cox survt prison clinic dose, dead(status)

Notice that with the **cox** command, we have to list the variable SURVT. The **dead()** option is used to indicate that the variable STATUS distinguishes events from censorship. The variable used with the **dead()** option needs to be coded nonzero for events and zero for censorships. The output from the **cox** command follows:

```
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

------------------------------------------------------------------------------
       _t
  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
------------------------------------------------------------------------------
```

The output is identical to that obtained from the **stcox** command except that the regression coefficients are

given by default. The **hr** option for the **cox** command supplies the exponentiated coefficients.

Notice with the output that the default method of handling ties (i.e., when multiple events happen at the same time) is the Breslow method. If you wish to use more exact methods, you can use the **exactp** option (for the exact partial likelihood) or the **exactm** option (for the exact marginal likelihood) in the **stcox** or **cox** command. The exact methods are computationally more intensive and typically have a slight impact on the parameter estimates. However, if there are a lot of events that occur at the same time, then exact methods are preferred. The code and output follow:

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
```

Alternatively, you could use Efron method of handling ties. This is the method that the R statistical package uses as its default. The code follows (output omitted):

stcox prison clinic dose, nohr efron

Suppose you are interested in running a Cox model with two interaction terms with PRISON. The **generate** command can be used to define new variables. The variables CLIN_PR and CLIN_DO are product terms that are defined from CLINIC $\times$ PRISON and CLINIC $\times$ DOSE. The code follows:

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

Type **describe** or **list** to see that the new variables are in the working dataset.

The following code runs the Cox model with the two interaction terms:

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
------------------------------------------------------------------------------
```

The **lrtest** command can be used to perform likelihood ratio tests. For example, to perform a likelihood ratio test on the two interaction terms, CLIN_PR and CLIN_DO, in the preceding model, we can save the –2 log likelihood statistic of the full model in the computer's memory by typing the following command:

lrtest, saving(0)

Now, the reduced model (without the interaction terms) can be run (output omitted) by typing:

stcox prison clinic dose

After the reduced model is run, the following command provides the results of the likelihood ratio test comparing the full model (with the interaction terms) to the reduced model:

lrtest

The resulting output follows:

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

The p-value of 0.1648 is not significant at the alpha = 0.05 level.

4. RUNNING A STRATIFIED COX MODEL

If the proportional hazard assumption is not met for the variable CLINIC, but is met for the variables PRISON and DOSE, then a stratified Cox analysis can be performed. The **stcox** command can be used to run a stratified Cox model. The following code (with output) runs a Cox model stratified on CLINIC:

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
```

The **strata()** option allows up to five stratified variables.

A stratified Cox model can be run including the two interaction terms. Recall that the **generate** command created these variables in the previous section. This model allows for the effect of PRISON and DOSE to differ for different values of CLINIC. The code and output follow:

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
```

Suppose we wish to estimate the hazard ratio for PRISON=1 vs. PRISON=0 for CLINIC=2. This hazard ratio can be estimated by exponentiating the coefficient for prison plus 2 times the coefficient for the clinic-prison interaction term. This expression is obtained by substituting the appropriate values into the hazard in both the numerator (for PRISON=1) and denominator (for PRISON=0) (see below):

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

The **lincom** command can be used to exponentiate linear combinations of parameters. Run this command directly after running the model to estimate the HR for PRISON where CLINIC=2. The code and output follow:

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
------------------------------------------------------------------------------
```

Models can also be run on a subsetted portion of the data using the **if** statement. The following code (with output) runs a Cox model on the data where CLINIC=2:

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
------------------------------------------------------------------------------
```

The hazard ratio estimates for PRISON=1 vs. PRISON=0 (for CLINIC=2) are exactly the same using the stratified Cox approach with product terms and the subsetted data approach (0.9210324).

5. ASSESSING THE PH ASSUMPTION WITH A STATISTICAL TEST

The **stphtest** command can be used to perform a statistical test. A statistical test gives objective criteria for assessing the PH assumption compared to using the graphical approach. This does not mean that this statistical test is better than the graphical approach. It is just more objective. In fact, the graphical approach is generally more informative for descriptively characterizing the form of a PH violation.

The command **stphtest** outputs a PH global test for all the covariates simultaneously and can also be used to obtain a test for each covariate separately with the detail option. To run these tests, you must obtain Schoenfeld residuals for the global test and scaled Schoenfeld residuals for separate tests with each covariate. The idea behind the PH test is that if the PH assumption is satisfied, then the residuals should not be correlated with survival time (or ranked survival time). On the other hand, if the residuals tend to be positive for subjects who become events at a relatively early time and negative for subjects who become events at a relatively late time (or vice versa), then there is evidence that the hazard ratio is not constant over time (i.e., PH assumption is violated).

Before the **stphtest** can be implemented, the **stcox** command needs to be run to obtain the Schoenfeld residuals (with the **schoenfeld()** option) and the scaled Schoenfeld residuals (with the **scaledsch()** option). The names of newly defined variables are in the parentheses: **schoen*** creates SCHOEN1, SCHOEN2, and SCHOEN3 while **scaled*** creates SCALED1, SCALED2, and SCALED3. These variables contain the residuals for PRISON, DOSE, and CLINIC, respectively (the order that the variables were entered in the model). The user is free to type any variable name in the parentheses. The Schoenfeld residuals are used for the global test while the scaled Schoenfeld residuals are used for the testing of the PH assumption for individual variables:

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

Once the residuals are defined, the **stphtest** command can be run. The code and output follow:

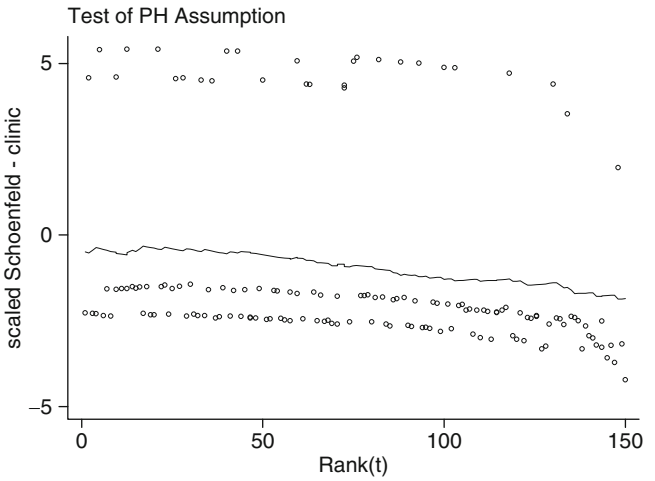
stphtest, rank detail

```
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
------------------------------------------------------------
```

The tests suggest that the PH assumption is violated for CLINIC with the p-value at 0.0012. The tests do not suggest violation of the PH assumption for PRISON or DOSE.

The **plot()** option of the **stphtest** command can be used to produce a plot of the scaled Schoenfeld residuals for CLINIC against survival time ranking. If the PH assumption is met, the fitted curve should look horizontal since the scaled Schoenfeld residuals would be independent of survival time. The code and graph follow:

stphtest, rank plot(clinic)



The fitted curve slopes slightly downward (not horizontal).

6. OBTAINING COX ADJUSTED SURVIVAL CURVES

Adjusted survival curves can be obtained with the **sts graph** command. Adjusted survival curves depend on the pattern of covariates. For example, the adjusted survival estimates for a subject with PRISON=1, CLINIC=1, and DOSE=40 are generally different than for a subject with PRISON=0, CLINIC=2, and DOSE=70. The **sts graph** command produces adjusted baseline survival curves. The following code produces an adjusted survival plot with PRISON=0, CLINIC=0, and DOSE=0 (output omitted):

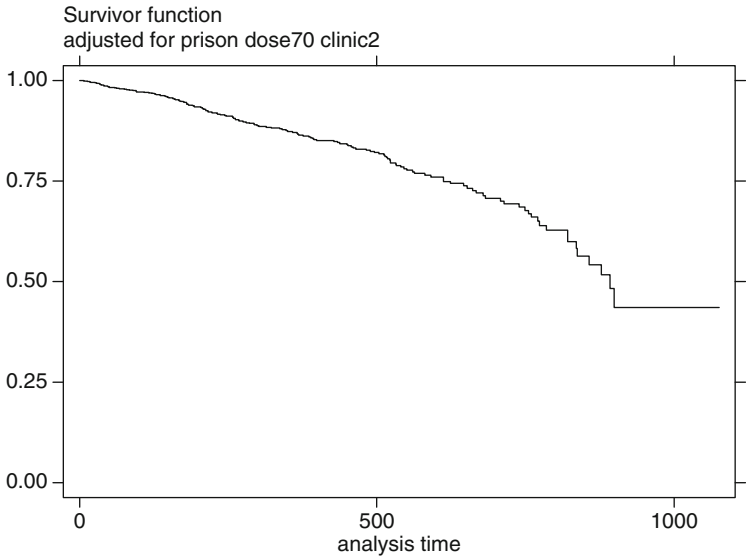
sts graph, adjustfor(prison dose clinic)

It is probably of more interest to create adjusted plots for reasonable patterns of covariates (CLINIC=0 is not even a valid value). Suppose we are interested in graphing the adjusted survival curve for PRISON=0, CLINIC=2, and DOSE=70. We can create new variables with the **generate** command that can be used with the **sts graph** command:

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

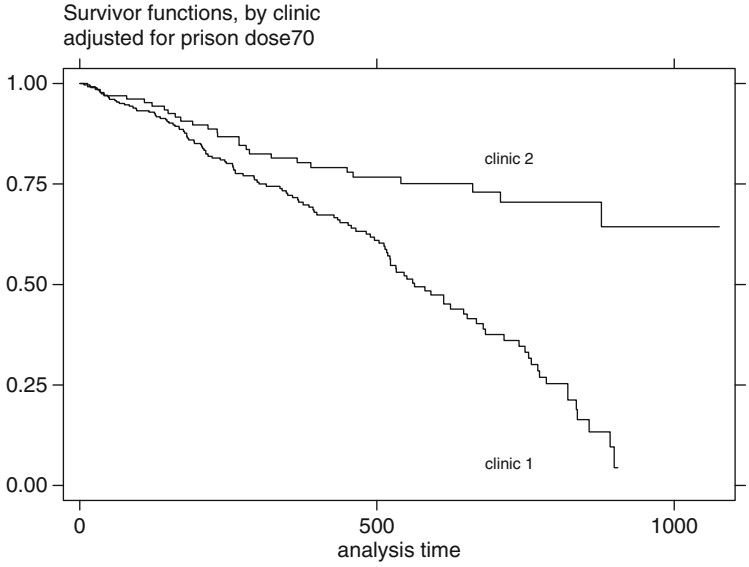
These variables (PRISON, CLINIC2, and DOSE70) produce the desired pattern of covariate when each is set to zero. The following code produces the desired results:

sts graph, adjustfor(prison dose70 clinic2)



Adjusted stratified Cox survival curves can be obtained with the **strata()** option. The following code creates two survival curves stratified by clinic (CLINIC=1, PRISON=0, and DOSE=70) and (CLINIC=2, PRISON=0, and DOSE=70):

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



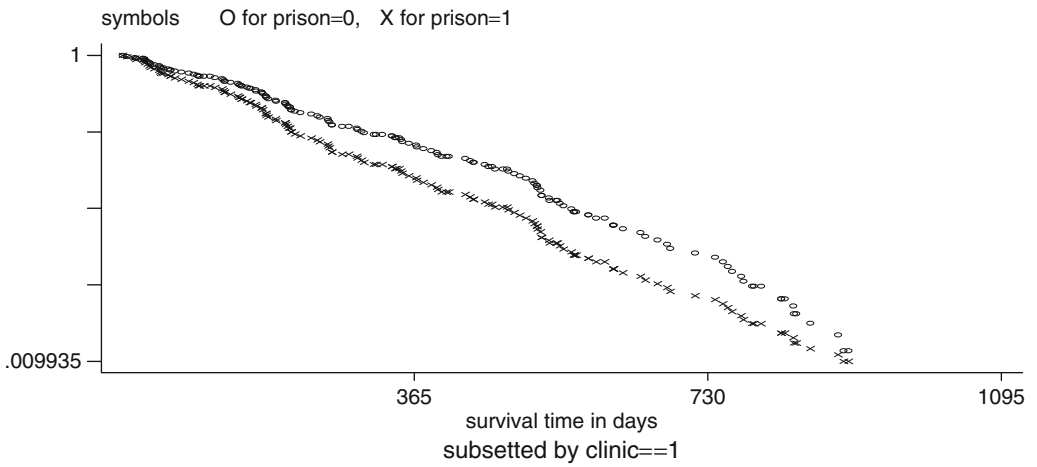
The adjusted curves suggest that there is a strong effect from CLINIC on survival.

Suppose the interest is in comparing adjusted survival plots of PRISON=1 to PRISON=0 stratified by CLINIC. In this setting, the **sts graph** command cannot be used directly since we cannot simultaneously define both levels of prison (PRISON=1 and PRISON=0) as the baseline level (recall **sts graph** plots only the baseline survival function). However, survival estimates can be obtained using the **sts generate** command twice, once where PRISON=0 is defined as baseline and once where PRISON=1 is defined as baseline. The following code creates variables containing the desired adjusted survival estimates:

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

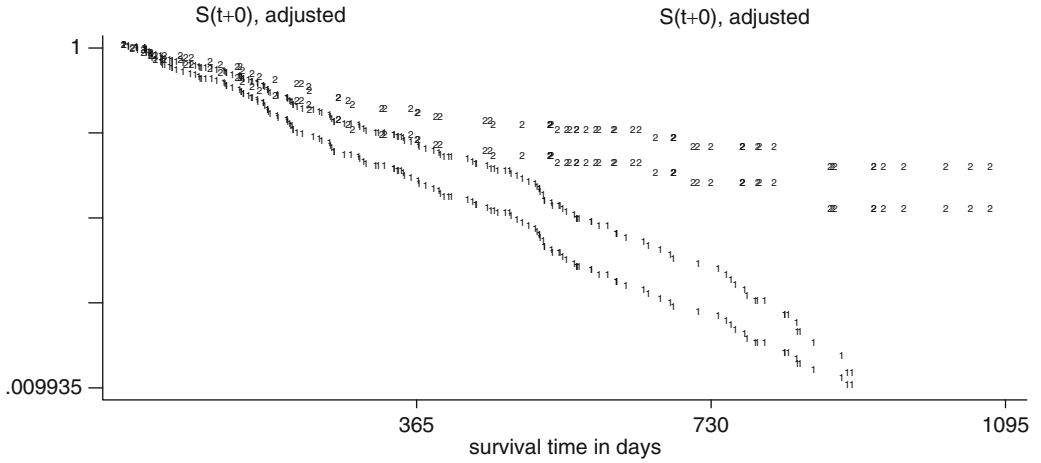
The variables SCOX1 and SCOX0 contain the survival estimates for PRISON=1 and PRISON=0, respectively, adjusting for dose and stratifying by clinic. The **graph** command is used to plot these estimates. If you are using a higher version of Stata than Stata 7.0 (e.g., Stata 8.0), then you should replace the **graph** command with the **graph7** command. The code and output follow:

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



We can also graph PRISON=1 and PRISON=0 subsetting the data where CLINIC=1. The option **twoway** requests a two-way scatter plot. The options **symbol**, **xlabel**, and **title** request the symbols, axis labels, and title, respectively:

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. RUNNING AN EXTENDED COX MODEL

If the PH assumption is not satisfied, a possible strategy is to run a stratified Cox model. Another strategy is to run a Cox model with time-varying covariates (an extended Cox model). The challenge of running an extended Cox model is to choose the appropriate function of survival time to include in the model.

Suppose we want to include a time dependent covariate DOSE times the log of time. This product term could be appropriate if the hazard ratio comparing any two levels of DOSE monotonically increases (or decreases) over time. The **tvc** option( ) of the **stcox** command can be used to declare DOSE a time varying covariate that will be multiplied by a function of time. The specification of that function of time is stated in the **texp** option with the variable **_t** representing time. The code and output for a model containing the time varying covariate, DOSE x ln(_t), follow:

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).
```

The parameter estimate for the time-dependent covariate, DOSE x ln(_t), is 0.0085751; however, it is not statistically significant with a Wald test p-value of 0.184.

A heaviside function can also be used. The following code runs a model with a time-dependent variable equal to CLINIC if time is greater than or equal to 365 days and 0 otherwise.

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

Stata recognizes the expression (_t>=365) as taking the value 1 if survival time is ≥365 days and 0 otherwise. The output follows:

```
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.
```

Unfortunately, the **texp** option can only be used once in the **stcox** command. This makes it more difficult to run the equivalent model with two heaviside functions. However, it can be accomplished using the **stsplit** command, which adds extra observations to the working dataset. The following code creates a variable called V1 and adds new observations to the dataset:

stsplit v1, at(365)

After the above **stsplit** command is executed, any subject followed more than 365 days is represented by two observations rather than one. For example, the first subject (ID=1) had an event on the 428th day; the first observation for that subject shows no event between 0 and 365 days while the second observation shows an event on the 428th day. The newly defined variable **v1** has the value 365 for observations with survival time exceeding or equal to 365 and 0 otherwise. The following code lists the first ten observations for the requested variables (output follows):

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
```

With the data in this form, two heaviside functions can actually be defined in the data using the following code:

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

The following code and output list a sample of the observations (**in 159/167**) with the observation number suppressed (the **noobs** option):

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
```

With the two heaviside functions defined in the split data, a time dependent model using these functions can be run with the following code (the output follows):

stcox prison dose hv1 hv2, nohr

stcox prison clinic 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
```

The **stsplit** command is complicated but it offers a powerful approach for manipulating the data to accommodate time varying analyses.

If you wish to return the data to its previous form, drop the variables that were created from the split and then use the **stjoin** command:

drop v1 hv1 hv2
stjoin

It is possible to split the data at every single failure time, but this uses a large amount of memory. However, if there is only one time varying covariate in the model, the simplest way to run an extended Cox model is by using the **tvc** and **texp** options with the **stcox** command.

One should not confuse an individual's survival time variable (the outcome variable) with the variable used to define the time dependent variable (**_t** in Stata). The individual's survival time variable is a time independent variable. The time of the individual's event (or censorship) does not change. A time-dependent variable, on the other hand, is defined so that it can change its values over time.

8. RUNNING PARAMETRIC MODELS

The Cox PH model is the most widely used model in survival analysis. A key reason why it is so popular is that the distribution of the survival time variable need not be specified. However, if it is believed that survival time follows a particular distribution, then that information can be utilized in a parametric modeling of survival data.

Many parametric models are accelerated failure time (AFT) models. Whereas the key assumption of a PH model is that hazard ratios are constant over time, the key assumption for an AFT model is that survival time accelerates (or decelerates) by a constant factor when comparing different levels of covariates.

The most common distribution for parametric modeling of survival data is the Weibull distribution. The Weibull distribution has the desirable property that if the AFT assumption holds, then the PH assumption also holds. The exponential distribution is a special case of the Weibull distribution. The key property for the exponential distribution is that the hazard is constant over time (not just the hazard ratio). The Weibull and exponential model can be run as a PH model (the default) or an AFT model.

A graphical method for checking the validity of a Weibull assumption is to examine Kaplan-Meier log-log survival curves against log survival time. This is accomplished with the **sts graph** command (see Section 2 of this appendix). If the plots are straight lines, then there is evidence that the distribution of survival times follows a Weibull distribution. If the slope of the line equals one, then the evidence suggests that survival time follows an exponential distribution.

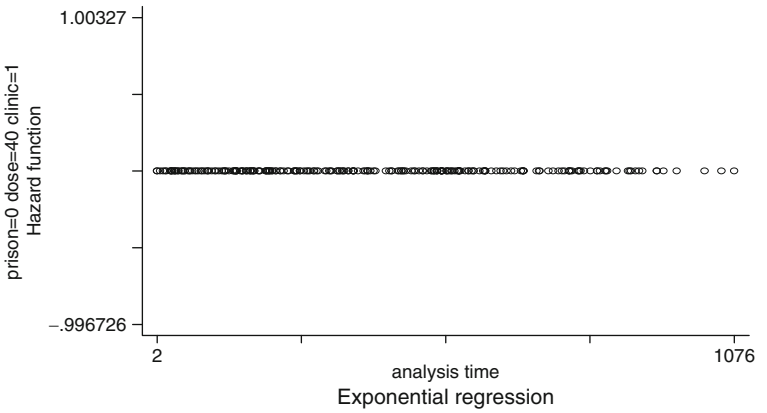
The **streg** command is used to run parametric models. Even though the log log survival curves obtained using the addicts dataset are not straight lines, the data will be used for illustration. First, a parametric model using the exponential distribution will be demonstrated. The code and output follow:

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
------------------------------------------------------------------------------
```

The distribution is specified with the **dist()** option. The **stcurv** command can be used following the **streg** command to obtain fitted survival, hazard, or cumulative hazard curves. The following code obtains the estimated hazard function for PRISON=0, DOSE=40, and CLINIC=1:

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



The graph illustrates the fact that the hazard is constant over time if survival time follows an exponential distribution.

Next, a Weibull distribution is run using the **streg** command:

```
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
------------------------------------------------------------------------------
```

Notice that the Weibull output has a parameter p that the exponential distribution does not have. The hazard function for a Weibull distribution is λpt^{p-1}. If p = 1, then the Weibull distribution is also an exponential distribution (h (t) = λ). Hazard ratio parameters are given by default for the Weibull distribution. If you want the parameterization for an AFT model, then use the **time** option.

The code and output for a Weibull AFT model follow:

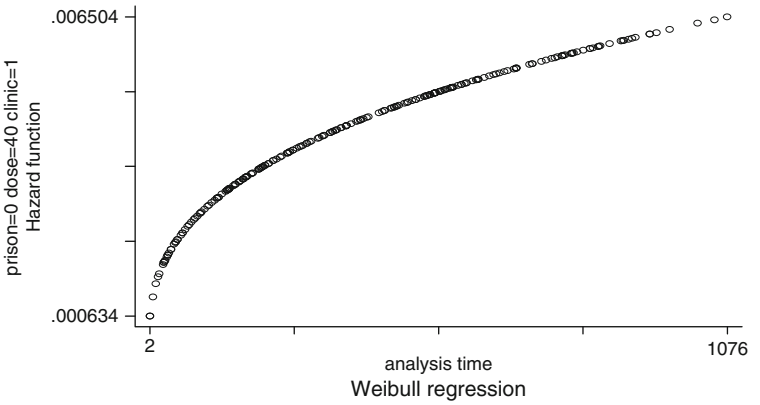
streg prison dose clinic, dist(weibull) time

```
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
------------------------------------------------------------------------------
```

The relationship between the hazard ratio parameter β_j and the AFT parameter α_j is $\beta_j = -\alpha_j p$. For example, using the coefficient estimates for PRISON in the Weibull PH and AFT models yields the relationship 0.3144 = (−0.2295)(1.37).

The **stcurv** can again be used following the **streg** command to obtain fitted survival, hazard, or cumulative hazard curves. The following code obtains the estimated hazard function for PRISON=0, DOSE=40, and CLINIC=1:

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



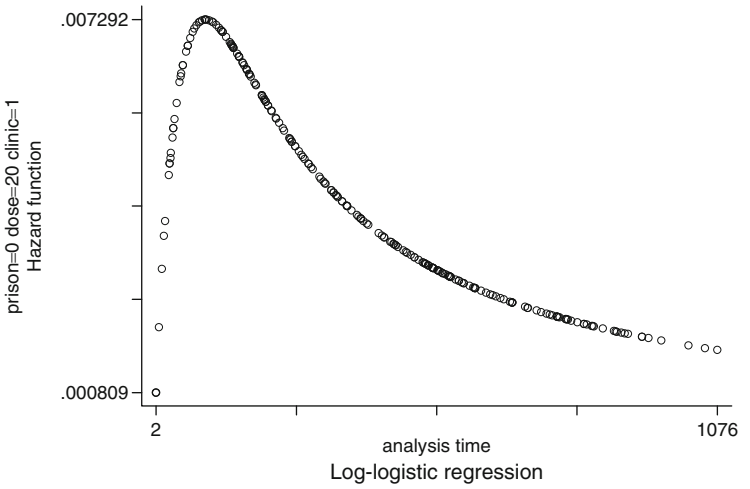
The plot of the hazard is monotonically increasing. With a Weibull distribution, the hazard is constrained such that it cannot increase and then decrease. This is not the case with the log logistic distribution as demonstrated in the next example. The log logistic model is not a PH model, so the default model for the **streg** command is an AFT model. The code and output follow:

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
------------------------------------------------------------------------------
```

Note that Stata calls the shape parameter gamma for a log-logistic model. The code to produce the graph of the hazard function for PRISON=0, DOSE=40, and CLINIC=1 follows:

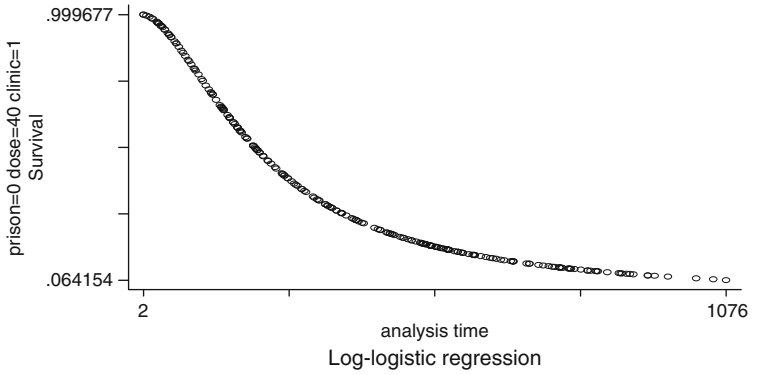
stcurv, hazard at (prison=0 dose=40 clinic=1)



The hazard function (in contrast to the Weibull hazard function) first increases and then decreases.

The corresponding survival curve for the log logistic distribution can also be obtained with the **stcurve** command:

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



If the AFT assumption holds for a log logistic model, then the proportional odds assumption holds for the survival function (although the PH assumption would not hold). The proportional odds assumption can be evaluated by plotting of the log odds of survival (using KM estimates) against the log of survival time. If the plots are straight lines for each pattern of covariates, then the log-logistic distribution is reasonable. If the straight lines are also parallel, then the proportional odds and AFT assumptions also hold. The following code will plot the estimated log odds of survival against the log of time by CLINIC (output omitted):

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

Another context for thinking about the proportional odds assumption is that the odds ratio estimated by a logistic regression does not depend on the length of the follow-up. For example, if a follow-up study was extended from 3 to 5 years, then the underlying odds ratio comparing two patterns of covariates would not change. If the proportional odds assumption is not true, then the odds ratio is specific to the length of follow-up.

Both the log-logistic and Weibull models contain an extra shape parameter that is typically assumed constant. This assumption is necessary for the PH or AFT assumption to hold for these models. Stata provides a way of modeling the shape parameter as a function of predictor variables by use of the **ancillary** option in the **streg** command (see Chapter 7 under the heading "Other Parametric Models"). The following code runs a log-logistic model in which the shape parameter gamma is modeled as a function of CLINIC while λ is modeled as a function of PRISON and DOSE:

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

The output follows:

```
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
------------------------------------------------------------------------------
```

Notice there is a parameter estimate for CLINIC as well as an intercept (_cons) under the heading ln_gam (the log of gamma). With this model, the estimate for gamma depends on whether CLINIC=1 or CLINIC=2. There is no easy interpretation for the predictor variables in this type of model, which is why it is not commonly used. However, for any specified value of PRISON, DOSE, and CLINIC, the hazard and survival functions can be estimated by substituting the parameter estimates into the expressions for the log-logistic hazard and survival functions.

Other distributions supported by **streg** are the generalized gamma, the lognormal, and the Gompertz distributions.

9. RUNNING FRAILTY MODELS

Frailty models contain an extra random component designed to account for individual-level differences in the hazard otherwise unaccounted for by the model. The frailty, α, is a multiplicative effect on the hazard assumed to follow some distribution. The hazard function conditional on the frailty can be expressed as $h(t|\alpha) = \alpha[h(t)]$.

Stata offers two choices for the distribution of the frailty: the gamma and the inverse-Gaussian, both of mean 1 and variance theta. The variance (theta) is a parameter estimated by the model. If theta = 0, then there is no frailty.

For the first example, a Weibull PH model is run with PRISON, DOSE, and CLINIC as predictors. A gamma distribution is assumed for the frailty component. The models in this section were run using Stata 8.0. The code follows:

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

The **frailty()** option requests that a frailty model be run. The output follows:

```
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
```

Notice that there is one additional parameter (theta) compared to the model run in the previous section. The estimate for theta is 2.09 times 10^{-7} or 0.000000209 which is essentially zero. A likelihood ratio test for the inclusion of theta is provided at the bottom of the output and yields a chi-square value of 0.00 and a p-value of 1.000. The frailty has no effect on the model and need not be included.

The next model will be the same as the previous except that CLINIC will not be included. One might expect a frailty component to play a larger role if an important covariate, such as CLINIC, is not included in the model. The code and output follow:

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
```

The variance (theta) of the frailty is estimated at 0.0578602. Although this estimate is not exactly zero as in the previous example, the p-value for the likelihood ratio test for theta is nonsignificant at 0.432. So the addition of frailty did not account for CLINIC being omitted from the model.

Next, the same model is run except that the inverse-Gaussian distribution is used for the frailty rather than the gamma distribution. The code and output follow:

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
```

The p-value for the likelihood ratio test for theta is 0.443 (at the bottom of the output). The results in this example are very similar whether assuming the inverse-Gaussian or the gamma distribution for the frailty component.

An example of shared frailty applied to recurrent event data is shown in the next section.

10. MODELING RECURRENT EVENTS

The modeling of recurrent events is illustrated with the bladder cancer dataset (**bladder.dta**) described at the start of this appendix. Recurrent events are represented in the data with multiple observations for subjects having multiple events. The data layout for the bladder cancer dataset is suitable for a counting process approach with time intervals defined for each observation (see Chapter 8). The following code prints the 12th–20th observation, which contains information for four subjects. The code and output follow:

list in 12/20

	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

There are three observations for ID=10, one observation for ID=11, three observations for ID=12, and two observations for ID=13. The variables START and STOP represent the time interval for the risk period specific to that observation. The variable EVENT indicates whether an event (coded 1) occurred. The first three observations indicate that the subject with ID=10 had an event at 12 months, another event at 16 months, and was censored at 18 months.

Before using Stata's survival commands, the **stset** command must be used to define the key survival variables. The code follows:

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

We have previously used the **stset** command on the "addicts" dataset, but more options from **stset** are included here. The **id()** option defines the subject variable (i.e., the cluster variable), the **time0()** option defines the variable that begins the time interval, and the **exit(time .)** option tells Stata that there is no imposed limit on the length of follow-up time for a given subject (e.g., subjects are not out of the risk set after their first event). With the **stset** command, Stata creates the variables **_t0**, **_t**, and **_d**, which Stata automatically recognizes as survival variables representing the time interval and event status. Actually, the **time0()** option could have been omitted from this **stset** command and by default Stata would have created the starting time variable, **_t0**, in the correct counting process format as long as the **id()** option was used (otherwise **_t0** would default to zero). The following code (and output) lists the 12^{th}–20^{th} observation with the newly created variables:

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
```

A Cox model with recurrent events using the counting process approach can now be run with the **stcox** command. The predictors are treatment status (TX), initial number of tumors (NUM), and the initial size of tumors (SIZE). The **robust** option requests robust standard errors for the coefficient estimates. Omit the **nohr** option if you want the exponentiated coefficients. The code and output follow:

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
------------------------------------------------------------------------------
```

The interpretation of these parameter estimates is discussed in Chapter 8

A stratified Cox model can also be run using the data in this format with the variable INTERVAL as the stratified variable. The stratified variable indicates whether subjects were at risk for their 1st, 2nd, 3rd, or 4th event. This approach is called a Stratified CP approach in Chap. 8 and is used if the investigator wants to distinguish the order in which recurrent events occur. The code and output follow:

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
```

Interaction terms between the treatment variable (TX) and the stratified variable could be created to examine whether the effect of treatment differed for the 1st, 2nd, 3rd, or 4th event. (Note that in this dataset, subjects have a maximum of 4 events).

Another stratified approach (called Gap Time) is a slight variation of the Stratified CP approach. The difference is in the way the time intervals for the recurrent events are defined. There is no difference in the time intervals when subjects are at risk for their first event. However, with the Gap Time approach, the starting time at risk gets reset to zero for each subsequent event. The following code creates data suitable for running a Gap Time recurrent event model.

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

The **generate** command defines a new variable called STOP2 representing the length of the time interval for each observation. The **stset** command is used with STOP2 as the outcome variable (**_t**). By default, Stata sets the variable **_t0** to zero. The following code (and output) lists the 12th through 20th observations for selected variables.

```
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
```

Notice that the **id()** option was not used with the **stset** command for the Gap Time approach. This means that Stata does not know that multiple observations correspond to the same subject. However, the **cluster()** option can be used directly in the **stcox** command to request that the analysis be clustered by ID (i.e., by subject). The following code runs a stratified Cox model using the Gap Time approach with the **cluster()** and **robust** options. The code and output follow:

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

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

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
```

The results using the Gap Time approach vary slightly from that obtained using the Stratified CP approach.

Next, we demonstrate how a shared frailty model can be applied to recurrent event data. Frailty is included in recurrent event analyses to account for variability due to unobserved subject-specific factors that may lead to within-subject correlation.

Before running the model, we rerun the **stset** command shown earlier in this section to get the data back to the form suitable for a counting process approach. The code follows:

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

Next a parametric Weibull model is run with a gamma-distributed shared frailty component using the **streg** command. We use the same three predictors for comparability with the other models presented in this section. The code follows:

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

The **dist()** option requests the distribution for the parametric model. The **frailty()** option requests the distribution for the frailty and the **shared()** option defines the cluster variable, ID. For this model, observations from the same subject share the same frailty. The output follows:

```
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
```

The model output is discussed in Chapter 8.

The counting process data layout with multiple observations per subject need not only apply to recurrent event data, but can also be used for a more conventional survival analyses in which each subject is limited to one event. A subject with four observations may be censored for the first three observations before getting the event in the time interval represented by the fourth observation. This data layout is particularly suitable for representing time-varying exposures, which may change values over different intervals of time (see the **stsplit** command in Section 7 of this appendix).

B. SAS

Analyses are carried out in SAS by using the appropriate SAS procedure on a SAS dataset. The key SAS procedures for performing survival analyses are:

PROC LIFETEST – This procedure is used to obtain Kaplan-Meier survival estimates and plots. It can also be used to output life table estimates and plots. It will generate output for the log rank and Wilcoxon test statistics if stratifying by a covariate. A new SAS dataset containing survival estimates can be requested.

PROC PHREG – This procedure is used to run the Cox proportional hazards model, a stratified Cox model, and an extended Cox model with time-varying covariates. It can also be used to create a SAS dataset containing adjusted survival estimates. These adjusted survival estimates can then be plotted using PROC GPLOT.

PROC LIFEREG – This procedure is used to run parametric accelerated failure time AFT models.

Analyses on the "addicts" dataset will be used to illustrate these procedures. The "addicts" dataset was obtained from a 1991 Australian study by Caplehorn et al. and contains information on 238 heroin addicts. The study compared two methadone treatment clinics to assess patient time remaining under methadone treatment. The two clinics differed according to its live-in policies for patients. A patient's survival time was determined as the time (in days) until the person dropped out of the clinic or was censored. The variables are defined at the start of this appendix.

All of the SAS programming code will be written in capital letters for readability. However, SAS is *not* case sensitive. If a program is written with lower-case letters, SAS reads them as upper case. The number of spaces between words (if more than one) has no effect on the program. Each SAS programming statement ends with a semicolon.

The addicts dataset is stored as a permanent SAS dataset called **addicts.sas7bdat**. A LIBNAME statement is needed to indicate the path to the location of the SAS dataset. In our examples, we assume the file is located on the C drive. The LIBNAME statement includes a reference name as well as the path. We call the reference name REF. The code is as follows:

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

The user is free to define his/her own reference name. The path to the location of the file is given between the quotation marks. The general form of the code is:

```
LIBNAME Your reference name 'Your path to file location';
```

PROC CONTENTS, PROC PRINT, PROC UNIVARIATE, PROC FREQ, and PROC MEANS can be used to list or describe the data. SAS code can be run in one batch or highlighted and submitted one procedure at a time. Code can be submitted by clicking on the submit button on the toolbar in the Editor window. The code for using these procedures follows (output omitted):

```
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;
```

Notice that each SAS statement ends with a semicolon. If each procedure is submitted one at a time, then each procedure must end with a RUN statement. Otherwise one RUN statement at the end of the last procedure is sufficient. With the LIBNAME statement, SAS recognizes a two-level file name: the reference name and the file name without an extension. For our example, the SAS file name is REF.ADDISTS. Alternatively, a temporary SAS dataset could be created and used for these procedures.

Text that you do not wish SAS to process can be written as a comment:

```
/* A comment begins with a forward slash followed by a
   star and ends with a star followed by a forward slash. */
* A comment can also be created by beginning with a star
  and ending with a semicolon;
```

The survival analyses demonstrated in SAS are as follows:

1. Demonstrating PROC LIFETEST to obtain Kaplan-Meier and life table survival estimates (and plots).
2. Running a Cox PH model with PROC PHREG.
3. Running a stratified Cox model.
4. Assessing the PH assumption with a statistical test.
5. Obtaining Cox adjusted survival curves.
6. Running an extended Cox model (i.e., containing time varying covariates).
7. Running parametric models with PROC LIFEREG.
8. Modeling recurrent events

1. DEMONSTRATING PROC LIFETEST TO OBTAIN KM AND LIFE TABLE SURVIVAL ESTIMATES (AND PLOTS)

PROC LIFETEST produces Kaplan-Meier survival estimates with the METHOD=KM option. The PLOTS=(S) option plots the estimated survival function. The TIME statement defines the time-to-event variable (SURVT) and the value for censorship (STATUS=0). The code follows (output omitted):

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

Use a STRATA statement in PROC LIFETEST to compare survival estimates for different groups (e.g., strata clinic). The PLOTS=(S, LLS) option produces log-log curves as well as survival curves. If the PH assumption is met, the log-log survival curves will be parallel. The STRATA statement also provides the log rank test and Wilcoxon test statistics. The code follows:

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

PROC LIFETEST yields the following edited output:

The LIFETEST Procedure (stratified)

Stratum 1: CLINIC = 1

Product-Limit Survival Estimates

SURVT	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.00	1.0000	0	0	0	163
2.00*	.	.	.	0	162
7.00	0.9938	0.00617	0.00615	1	161
17.00	0.9877	0.0123	0.00868	2	160
.	.	.	.	.	.
.	.	.	.	.	.
836.00	0.0869	0.9131	0.0295	118	6
837.00	0.0725	0.9275	0.0279	119	5

Stratum 2: CLINIC = 2

Product-Limit Survival Estimates

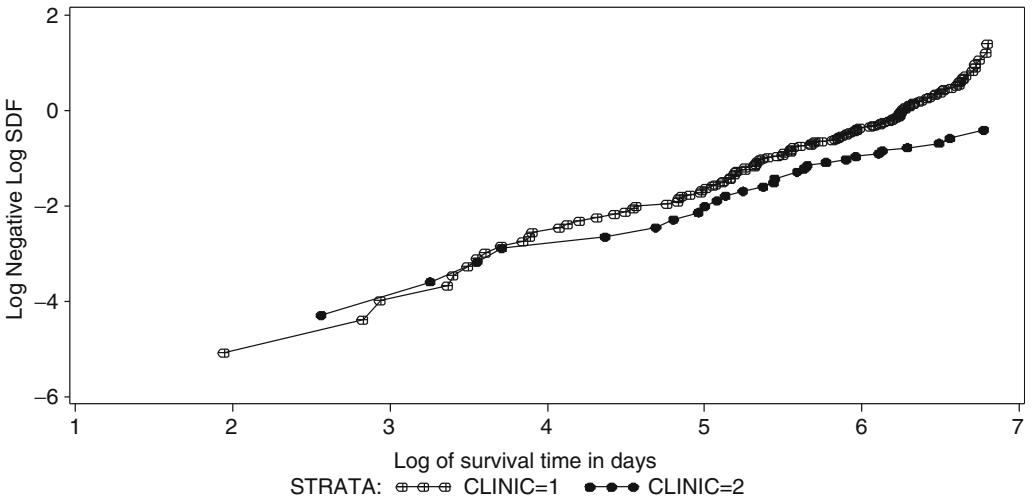
SURVT	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.00	1.0000	0	0	0	75
2.00*	.	.	.	0	74
13.00	0.9865	0.0135	0.0134	1	73
26.00	0.9730	0.0270	0.0189	2	72
.	.	.	.	.	.
.	.	.	.	.	.

Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	27.8927	1	<.0001
Wilcoxon	11.6268	1	0.0007
-2Log(LR)	26.0236	1	<.0001

Both the log rank and Wilcoxon test yield highly significant chi-square test statistics. The Wilcoxon test is a variation of the log rank test weighting the observed minus expected score of the j^{th} failure time by n_j (the number still at risk at the j^{th} failure time).

The requested log-log plots from PROC LIFETEST follow:



SAS (as well as Stata and R) plots log(survival time) rather than survival time on the horizontal axis by default for log-log curves. As far as checking the parallel assumption, it does not matter if log(survival time) or survival time is on the horizontal axis. However, if the log-log survival curves look like straight lines with log(survival time) on the horizontal axis, then there is evidence that the "time-to-event" variable follows a Weibull distribution. If the slope of the line equals one, then there is evidence that the survival time variable follows an exponential distribution – a special case of the Weibull distribution. For these situations, a parametric survival model can be used.

You can gain more control over how variables are plotted, by creating a dataset that contains the survival estimates. Use the OUTSURV= option in the PROC LIFETEST statement to create a SAS data containing the KM survival estimates. The option OUTSURV=DOG creates a dataset called dog (make up your own name) containing the survival estimates in a variable called SURVIVAL. The code follows:

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

Data dog contains the survival estimates but not the log(-(log)) of the survival estimates. Data cat is created in the following code from data dog (using the statement SET DOG) and defines a new log-log variable called LLS.

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

In SAS, the LOG function returns the natural log, not the log base 10.

PROC PRINT prints the data in the output window.

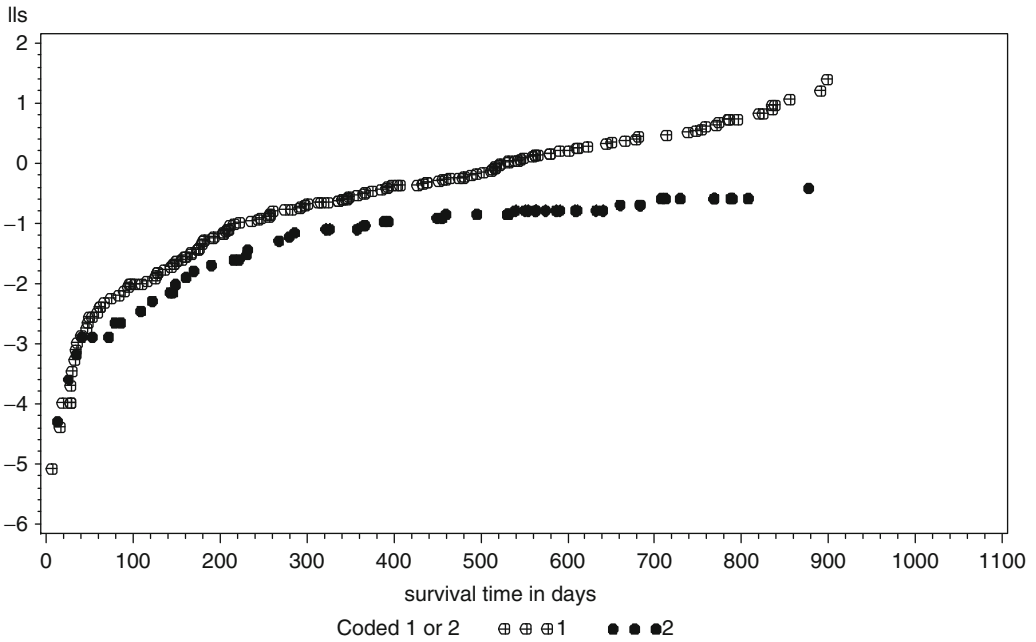
```
PROC PRINT DATA=CAT; RUN;
```

The first 10 observations from PROC PRINT are listed below:

Obs	CLINIC	SURVT	_CENSOR_	SURVIVAL	LLS
1	1	0	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

The PLOT LLS*SURVT=CLINIC statement puts the variable LLS (the log-log survival variables) on the vertical axis and SURVT on the horizontal axis, stratified by CLINIC. The SYMBOL option can be used to choose plotting colors for each level of clinic. The code and output for plotting the log log curves by CLINIC follow:

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

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



The plot has survival time (in days) rather than the default log(survival time). The log-log survival plots look parallel for CLINIC the first 365 days but then seem to diverge. This information can be utilized when developing an approach for modeling CLINIC with a time dependent variable in an extended Cox model.

You can also obtain survival estimates using life tables. This method is useful if you do not have individual level survival information but rather have group survival information for specified time intervals. The user determines the time intervals using the INTERVALS= option. The code follows (output omitted):

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

2. RUNNING A COX PROPORTIONAL HAZARD MODEL WITH PROC PHREG

PROC PHREG is used to request a Cox proportional hazards model. The code follows:

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

The code SURVT*STATUS(0), in the MODEL statement specifies the time-to-event variable (SURVT) and the value for censorship (STATUS=0). Three predictors are included in the model: PRISON, DOSE, and CLINIC. The option RL in the MODEL statement of PROC PHREG provides 95% confidence intervals for the hazard ratio estimates. The PH assumption is assumed to follow for each of these predictors (perhaps incorrectly). The output produced by PROC PHREG follows:

The PHREG Procedure
Model Information

Data Set	REF.ADDICTS	
Dependent Variable	SURVT	survival time in days
Censoring Variable	STATUS	status (0=censored, 1=endpoint)
Censoring Value(s)	0	
Ties Handling	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

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	95% Hazard Ratio Confidence Limits	
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

The table above lists the parameter estimates for the regression coefficients, their standard errors, a Wald chi-square test statistic for each predictor, and corresponding p-value. The column labeled HAZARD RATIO gives the estimated hazard ratio per one-unit change in each predictor by exponentiating the estimated regression coefficients. The final two columns give the 95% confidence limits for this hazard ratio.

You can use the TIES=EXACT option in the model statement rather than run the default TIES=BRESLOW option that was used in the previous model. The TIES=EXACT option is a computationally intensive method to handle events that occur at the same time. If many events

occur simultaneously in the data, then the TIES=EXACT option is preferred. Otherwise, the difference between this option and the default is slight. The TIES=EFRON option is another tie-handling approach that SAS offers. The TIES=EFRON is the default method used in R.

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

The output follows:

```
                    The PHREG Procedure

                     Model Information

Data Set                  REF.ADDICTS
Dependent Variable        SURVT         survival time in days
Censoring Variable        STATUS        status (0=censored, 1=endpoint)
Censoring Value(s)        0
Ties Handling             EXACT

   Analysis of Maximum Likelihood Estimates

                Parameter  Standard                          Hazard   95% Hazard Ratio
Variable DF      Estimate     Error  Chi-Square  Pr > ChiSq   Ratio  Confidence  Limits

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
```

The parameter estimates and their standard errors vary only slightly from the previous model without the TIES=EXACT option. Notice that the type of ties-handling approach is listed in the table called MODEL INFORMATION in the output.

Suppose we wish to assess interaction between PRISON and CLINIC and between PRISON and DOSE. We can define two interaction terms in a new temporary SAS dataset (called addicts2) and then run a model containing those terms. Product terms for CLINIC times PRISON (called CLIN_PR) and CLINIC time DOSE (called CLIN_DO) are defined in the following data step:

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

The interaction terms (called CLIN_PR and CLIN_DO) are then added to the model. The CONTRAST statement can be used to test the two interaction terms simultaneously with a generalized Wald test. After the word CONTRAST is a user-supplied label in quotes (i.e., the user's option what to put in quotes). Then the tested covariates (the product terms) are listed followed by a 1 and separated by a comma (see code below):

```
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;
```

The PROC PHREG output follows:

The PHREG Procedure

Model Information

Data Set	WORK.ADDICTS2	
Dependent Variable	SURVT	survival time in days
Censoring Variable	STATUS	status (0=censored, 1=endpoint)
Censoring Value(s)	0	
Ties Handling	BRESLOW	

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	1411.324	1343.199
AIC	1411.324	1353.199
SBC	1411.324	1368.253

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
PRISON	1	1.19200	0.54137	4.8480	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

The estimates of the hazard ratios (left column) may be deceptive when product terms are in the model. For example, by exponentiating the estimated coefficient for PRISON at exp (1.19200) = 3.284, we obtain the estimated hazard ratio for PRISON=1 versus PRISON=0, where DOSE=0 and CLINIC=0. This is a meaningless hazard ratio since CLINIC is coded 1 or 2 and DOSE is always greater than zero (all patients are on methadone). In the next section (on stratified Cox models), we illustrate how a CONTRAST statement can be used to obtain more meaningful hazard ratio estimates for models with interaction terms. The CONTRAST statement can be used to obtain a linear combination of parameter estimates in addition to the generalized Wald test shown above.

The Wald chi-square p-values for the two product terms are 0.0872 for CLIN_PR and 0.3333 for CLIN_DO. The generalized Wald chi-square p-values for testing both product terms simultaneously is 0.1669. Alternatively, a likelihood ratio test can simultaneously test both product terms by subtracting the –2 log-likelihood statistic for the full model (with the two product terms) from the reduced model (without the product terms). The –2 log likelihood statistic can be found on the output in the table called MODEL FIT STATISTICS and under the column called WITH COVARIATES. The –2 log likelihood statistic is 1,343.199 for the full model and 1,346.805 for the reduced model. The test is a two degree of freedom test since 2 product terms are simultaneously tested.

The PROBCHI function in SAS can be used to obtain p-values for chi-square tests. The code follows:

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

PROC PRINT DATA=TEST;RUN;
```

Note that you must write 1 minus the PROBCHI function to obtain the area under the right side of the chi-square probability density function. The output from the PROC PRINT follows:

Obs	REDUCED	FULL	DF	P_VALUE
1	1346.81	1343.20	2	0.16480

The p-value for the likelihood ratio test for both product terms is 0.16480, a similar result to the p-value that was obtained from the generalized Wald test (0.1669). Both of these tests are two degree of freedom tests since the two interaction terms are simultaneously tested.

3. RUNNING A STRATIFIED COX MODEL

Suppose we believe that the variable CLINIC violates the proportional hazards assumption but the variables PRISON and DOSE follow the PH assumption within each level of CLINIC. A stratified Cox model on the variable CLINIC can be run with PROC PHREG using the STRATA CLINIC statement. The code follows:

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

The output of the parameter estimates follows:

The PHREG Procedure
Analysis of Maximum Likelihood Estimates

		Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	95% Hazard Ratio Confidence Limits	
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

Notice there is no parameter estimate for CLINIC since CLINIC is the stratified variable. The hazard ratio for PRISON=1 vs. PRISON=0 is estimated at 1.475. This hazard ratio is assumed not to depend on CLINIC since an interaction term between PRISON and CLINIC was not included in the model.

Suppose we wish to assess interaction between PRISON and CLINIC as well as DOSE and CLINIC in a Cox model stratified by CLINIC. We can define interaction terms in a new SAS dataset (called addicts2) and then run a model containing these terms.

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

Note with the interaction model that the hazard ratio for PRISON=1 versus PRISON=0 for CLINIC=1 controlling for

DOSE is $\exp(\beta_1 + \beta_3)$, and the hazard ratio for PRISON=1 versus PRISON=0 for CLINIC=2 controlling for DOSE is $\exp(\beta_1 + 2\beta_3)$. This latter calculation is obtained by substituting the appropriate values into the hazard in both the numerator (for PRISON=1) and denominator (for PRISON=0) (see below):

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

A CONTRAST statement with the ESTIMTES= option can be used with PROC PHREG when we wish to obtain estimates of a linear combination of parameter estimates. We can also use the CONTRAST statement to test the two interaction terms simultaneously with a generalized Wald test as we illustrated in the previous section.

The code below runs a stratified Cox model (STRATA CLINIC) including two interaction terms in the model. Three CONTRAST statements are used: the first to estimate the hazard ratio for PRISON among those with CLINIC=1, $\exp(\beta_1 + \beta_3)$; the second to estimate the hazard ratio for PRISON among those with CLINIC=2, $\exp(1 + 2\beta_3)$; and the third to test the two interaction terms with a two degree of freedom generalized Wald test. The ESTIMATE=EXP option in the first two CONTRAST statements requests that the parameter estimates be exponentiated. The code in the second CONTRAST statement PRISON **1** CLIN_PR **2**/ESTIMATE=EXP; requests the estimate for $\exp(\beta_1 + 2\beta_3)$. The β_1 corresponds to PRISON and the beta3 (β)corresponds to the third variable in the model, CLIN_PR. The code follows:

```
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;
```

Notice that when we stratify by CLINIC, we do not put the variable CLINIC in the model statement. However, the interaction terms CLIN_PR and CLIN_DO are put in the model statement while CLINIC is put in the strata statement. The output follows:

The PHREG Procedure
Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
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

Contrast Rows Estimation and Testing Results

Contrast	Type	Estimate	Standard Error	Confidence Limits	
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

Contrast Test Results

Contrast	DF	Wald Chi-Square	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

The hazard ratio (PRISON=1 vs PRISON=0) is estimated at 1.6528 among CLINIC=1 and 0.9211 among CLINIC=2. The generalized Wald test for testing both interaction terms simultaneously (a 2 df test: 1 $\beta_3 = 0$, 1 $\beta_4 = 0$) yields a p-value of 0.3936.

An alternative approach allowing for interaction with CLINIC and the other covariates is obtained by running two models: one subsetting on the observations where CLINIC=1 and the other subsetting on the observations where CLINIC=2. The code and output follow:

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

A WHERE statement in a SAS procedure subsets the number of observations for analyses. A TITLE statement can also be added to the procedure. The output containing the parameter estimates subsetting on the observations where CLINIC=1 follows:

COX MODEL RUN ONLY ON DATA WHERE CLINIC=1
Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
PRISON	1	0.50249	0.18869	7.0918	0.0077	1.653
DOSE	1	-0.03586	0.00774	21.4761	<.0001	0.965

Similarly, the code and output containing the parameter estimates subsetting on the observations where CLINIC=2:

```
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
Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Variable Label
PRISON	1	-0.08226	0.38430	0.0458	0.8305	0.921	0=none, 1=prison record
DOSE	1	-0.03693	0.01234	8.9500	0.0028	0.964	methadone dose (mg/day)

The estimated hazard ratio for PRISON=1 versus PRISON=0 is 0.921 among CLINIC=2 controlling for DOSE. This result is consistent with the stratified Cox model previously run in which all the product terms with CLINIC were included in the model.

4. ASSESSING THE PH ASSUMPTION WITH A STATISTICAL TEST

The following SAS program makes use of the addicts dataset to demonstrate how a statistical test of the PH assumption is performed for a given covariate (Harrel and Lee 1986). This is accomplished by finding the correlation between the Schoenfeld residuals for a particular covariate and the ranking of individual failure times. If the PH assumption is met, then the correlation should be near zero. The p-value for testing this correlation can be obtained from PROC CORR (or PROC REG). The Schoenfeld residuals for a given model can be saved in a SAS dataset using PROC PHREG. The ranking of events by failure time can be saved in a SAS dataset using PROC RANKED. The null hypothesis is that the PH assumption is not violated.

First, we run a model containing CLINIC, PRISON, and DOSE. The output statement creates a SAS dataset, the OUT= option defines an output dataset, and the RESSCH= statement is followed by user-defined variable names, so that the output dataset contains the Schoenfeld residuals. The order of the names corresponds to the order of the independent variables in the model statement. The actual variable names are arbitrary. The name we chose for the dataset is RESID and the names we chose for the variables containing the Schoenfeld residuals for CLINIC, PRISON, and DOSE are RCLINIC, RPRISON, and RDOSE. The code follows:

The code follows:

```
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;
```

The first 10 observations of the PROC PRINT are printed below. The three columns on the right are the variables containing the Schoenfeld residuals.

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

Next, create a SAS dataset that deletes censored observations (i.e., only contains observations that fail).

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

Use PROC RANK to create a dataset containing a variable that ranks the order of failure times. The user supplies the name of the output dataset using the OUT= option. The variable to be ranked is SURVT. The RANKS statement precedes a user-defined variable name for the rankings of failure times. The user-defined names are arbitrary. The name we chose for this variable is TIMERANK. The code follows:

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

PROC PRINT DATA=RANKED;RUN;
```

PROC CORR is used to get the correlations between the ranked failure time variable (called TIMERANK in this example) and the variables containing the Schoenfeld residuals of CLINIC, PRISON, and DOSE (called RCLINIC, RPRISON, and RDOSE, respectively, in this example). The NOSIMPLE option suppresses the printing of summary statistics. If the PH assumption is met for a particular covariate, then the correlation should be near zero. The p-value obtained from PROC CORR which tests whether this correlation is zero is the same p-value we use for testing the PH assumption. The code follows:

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

The PROC CORR output follows:

```
                    The CORR Procedure

      Pearson Correlation Coefficients, N = 150
              Prob > |r| under H0: Rho=0

                              RCLINIC    RPRISON     RDOSE

TIMERANK                    -0.26153   -0.07970   0.07733
Rank for Variable SURVT       0.0012     0.3323    0.3469
```

The sample correlations with their corresponding p-values printed underneath are shown above. The p-values for CLINIC, PRISON, and DOSE are 0.0012, 0.3323, and 0.3469, respectively, suggesting that the PH assumption is violated for CLINIC, but reasonable for PRISON and DOSE.

The same p-values can be obtained by running linear regressions with each predictor (one at a time) using PROC REG and examining the p-values for the regression coefficients. The code below will produce output containing the p-value for CLINIC:

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

The output produced by PROC REG follows:

The REG Procedure

Parameter Estimates

Variable	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	75.49955	3.43535	21.98	<.0001
RCLINIC	-28.38848	8.61194	-3.30	0.0012

The p-value for CLINIC (0.0012) is printed in the column on the right and matches the p-value that was obtained using PROC CORR.

5. OBTAINING COX ADJUSTED SURVIVAL CURVES

We use the BASELINE statement in PROC PHREG to create an output dataset containing Cox adjusted survival estimates for a specified pattern of covariates. The particular pattern of covariates of interest must first be created in a SAS dataset that is subsequently used as the input dataset for the COVARIATES= option in the BASELINE statement of PROC PHREG. Each pattern of covariates yields a different survival curve (assuming nonzero effects). Adjusted log(-log) survival plots can also be obtained for assessing the PH assumption. This will be illustrated with three examples:

Ex1 – Run a PH model using PRISON, DOSE, and CLINIC and obtain adjusted survival curves where PRISON=0, DOSE=70, and CLINIC=2.

Ex2 – Run a stratified Cox model (by CLINIC). Obtain two adjusted survival curves using the mean value of PRISON and DOSE for CLINIC=1 and CLINIC=2. Use the log log curves to assess the PH assumption for CLINIC adjusted for PRISON and DOSE.

Ex3 – Run a stratified Cox model (by CLINIC) and obtain adjusted survival curves for PRISON=0, DOSE=70 and for PRISON=1, DOSE=70. This yields four survival curves in all, two for CLINIC=1 and two for CLINIC=2.

Basically, there are three steps:

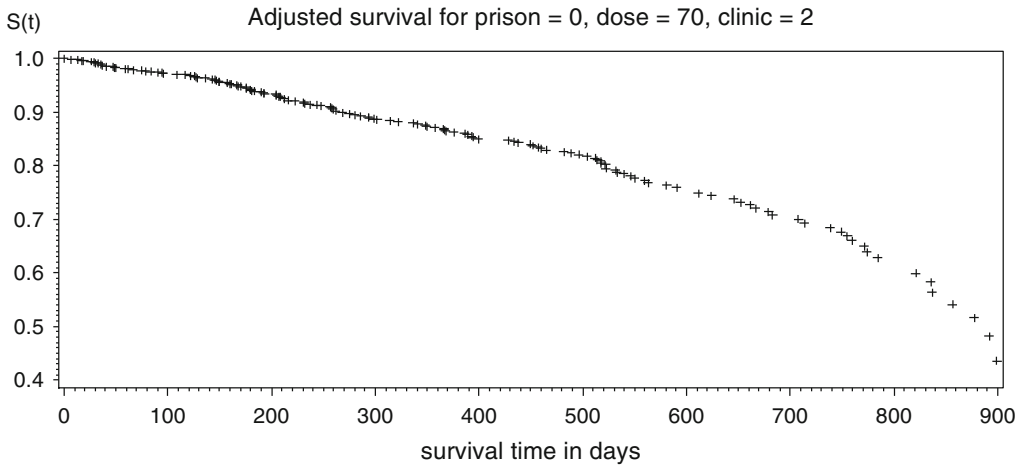
1) Create the input dataset containing the pattern (values) of covariates used for the adjusted survival curves.
2) Run a Cox model with PROC PHREG using the BASELINE statement to input the dataset from step (1) and output a dataset containing the adjusted survival estimates.
3) Plot the adjusted survival estimates from the output dataset created in step (2).

For Ex1, we create an input dataset (called IN1) with one observation where PRISON=0, DOSE=70, and CLINIC=2. We then run a model and create an output dataset (called OUT1) containing a variable with the adjusted survival estimates (called S1). Finally, the adjusted survival curve is plotted using PROC GPLOT. The code follows:

```
DATA IN1;
INPUT PRISON DOSE CLINIC;
CARDS;
0 70 2
;

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;
```

The BASELINE statement in PROC PHREG specifies the input dataset, the output dataset, and the name of the variable containing the adjusted survival estimates. The NOMEAN option suppresses the survival estimates using the mean values of PRISON, DOSE, and CLINIC. The next example (Ex2) will not use the NOMEAN option.

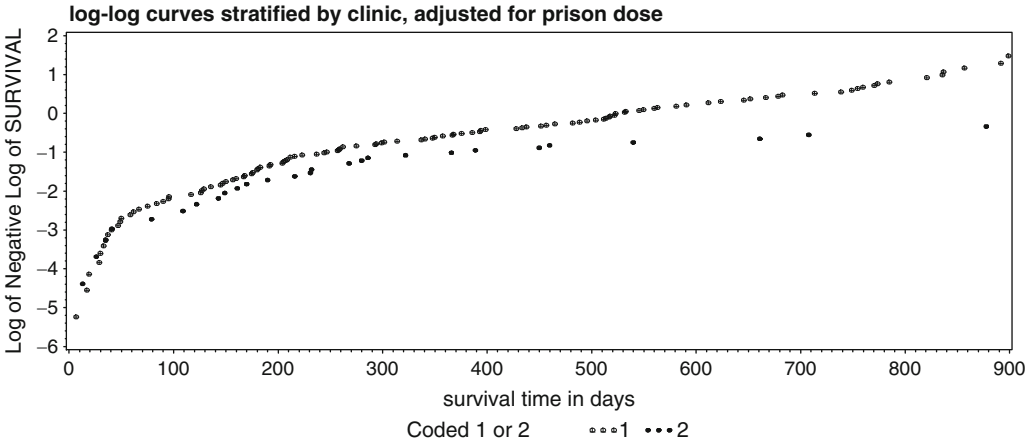
The output for PROC GPLOT follows:



For Ex2, we wish to create and output dataset (called OUT2) that contains the adjusted survival estimates from a Cox model stratified by CLINIC using the mean values of PRISON and DOSE. An input dataset need not be specified since by default the mean values of PRISON and DOSE will be used if the NOMEAN option is not used in the BASELINE statement. The code follows:

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

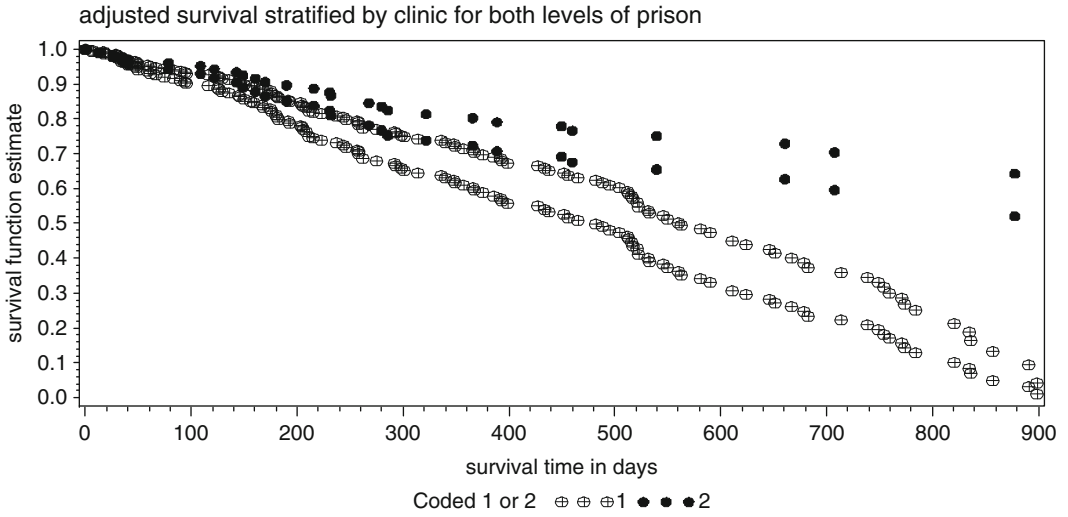
The code, PLOT LS2*SURVT=CLINIC, in the 2nd PROC GPLOT will plot LS2 on the vertical axis, SURVT on the horizontal axis, stratified by CLINIC on the same graph. The variable LS2 was created in the BASELINE statement of PROC PHREG and contains the adjusted log-log survival estimates. The PROC GPLOT output for the log-log survival curves stratified by CLINIC adjusted for PRISON and DOSE follows:



The adjusted log-log plots look similar to the unadjusted log-log Kaplan-Meier plots shown earlier, in that the plots look reasonably parallel before 365 days but then diverge, suggesting that the PH assumption is violated after 1 year.

For Ex3, a stratified Cox (by CLINIC) is run and adjusted curves are obtained for PRISON=1 and PRISON=0 holding DOSE=70. An input dataset (called IN3) is created with two observations for both levels of PRISON with DOSE=70. An output dataset (called OUT3) is created with the BASELINE statement that contains a variable (called S3) of survival estimates for all four curves (two for each stratum of CLINIC). The code follows:

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

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=OUT3;
PLOT S3*SURVT=CLINIC;
TITLE adjusted survival stratified by clinic for both levels
  of prison;
RUN;
```



For the graph above, the PH assumption is not assumed for CLINIC since that is the stratified variable. However, the PH assumption is assumed for PRISON within each stratum of CLINIC (i.e., CLINIC=1 and CLINIC=2).

6. RUNNING AN EXTENDED COX MODEL

Models containing time-dependent variables are run using PROC PHREG. Time dependent variables are created with programming statements within the PROC PHREG procedure. Sometimes, users incorrectly define time-dependent variables in the data step. This leads to wrong estimates because the time variable used in the data step (SURVT) is actually time-independent and therefore different than the time variable (also called SURVT) used to define time-dependent variables in the PROC PHREG statement. See the discussion on the extended Cox likelihood in Chapter 6 for further clarification of this issue.

We have evaluated the PH assumption for the variable CLINIC by plotting KM log-log curves and Cox-adjusted log log curves stratified by CLINIC and checking whether the curves were parallel. We could do similar analyses with the variables PRISON and DOSE although with DOSE we would need to categorize the continuous variable before comparing plots for different strata of DOSE.

If it is expected that the hazard ratio for the effect of DOSE increases (or decreases) monotonically with time, we could add a continuous time-varying product term with DOSE and some function of time. The model defined below contains a time-varying variable (LOGTDOSE) defined as the product of DOSE and the natural log of time (SURVT). In some sense, a violation of the PH assumption for a particular variable means that there is an interaction between that variable and time. Note that the variable LOGTDOSE is defined within the PHREG procedure and not in the data step. The code follows:

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

The output produced by PROC PHREG follows:

The PHREG Procedure
Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
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

The Wald test for the time-dependent variable LOGTDOSE yields a p-value of 0.1841. A nonsignificant p-value does not necessarily mean that the PH assumption is reasonable for DOSE. Perhaps, a different defined time-dependent variable would have been significant (e.g., DOSE $\times$ (TIME – 100)). Also, the sample-size of the study is a key determinant of the power to reject the null, which in this case means rejection of the PH assumption.

Next, we consider time-dependent variables for CLINIC. The next two models use heaviside functions that allow a different hazard ratio to be estimated for CLINIC before and after 365 days. The first model uses two heaviside functions in the model (HV1 and HV2) but not CLINIC. The second model uses one heaviside function (HV) but also includes CLINIC in the model. These two models yield the same hazard ratio estimates for CLINIC but are coded differently. The code and output for the model with two heaviside functions follows:

```
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;
```

Analysis of Maximum Likelihood Estimates

Parameter	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
PRISON	0.37770	0.16840	5.0304	0.0249	1.459
DOSE	-0.03551	0.00644	30.4503	<.0001	0.965
HV1	-0.45956	0.25529	3.2405	0.0718	0.632
HV2	-1.82823	0.38595	22.4392	<.0001	0.161

Contrast Test Results

Contrast	DF	Wald Chi-Square	Pr > ChiSq
TEST EQUALITY OF HEAVISIDES	1	8.7993	0.0030

The parameter estimates for HV1 and HV2 can be used directly to obtain the estimated hazard ratio for CLINIC=2 vs CLINIC=1 before and after 365 days. The estimated hazard ratio for CLINIC at 100 days is exp(–0.45956) = 0.632 and the estimated hazard ratio for CLINIC at 400 days is exp(–1.82823) = 0.161. The CONTRAST statement provides a Wald test on the equality of two heaviside coefficients ($\beta_3 = \beta_4$ or $\beta_3 - \beta_4 = 0$). If the two heaviside coefficients were equal, then the hazard ratios for CLINIC would not depend on time. So the test could be viewed as a test of one form of PH violation. The p-value for the test is highly significant at 0.0030, suggesting that the PH assumption is violated for CLINIC.

The code and output for an equivalent model with one heaviside function are shown below:

```
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;
```

Analysis of Maximum Likelihood Estimates

Parameter	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
CLINIC	-0.45956	0.25529	3.2405	0.0718	0.632
PRISON	0.37770	0.16840	5.0304	0.0249	1.459
DOSE	-0.03551	0.00644	30.4503	<.0001	0.965
HV	-1.36866	0.46139	8.7993	0.0030	0.254

Contrast Rows Estimation and Testing Results

Contrast	Type	Estimate	Standard Error	Confidence Limits	
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

Notice that the variable CLINIC is included in this model and the coefficient for the time-dependent heaviside function, HV, does not contribute to the estimated hazard ratio until day 365. The estimated hazard ratio for CLINIC at 100 days is exp(–0.45956) = 0.6316 while the estimated hazard ratio for CLINIC at 400 days is exp((–0.45956) + (–1.36866)) = 0.1607 as calculated using the ESTIMATE=EXP option in the CONTRAST statement. These results are consistent with the estimates obtained from the model with two heaviside functions. A Wald test for the variable HV shows a statistically significant *p*-value of 0.003 suggesting a violation of the PH assumption for CLINIC. This is the same test as that obtained with the CONTRAST statement using the model with two heaviside functions.

Suppose it is believed that the hazard ratio for CLINIC=2 versus CLINIC=1 is constant over the first year but then monotonically increases (or decreases) after the first year. The following code defines a model allowing for a time-varying covariate called CLINTIME (defined in the code) which contributes to the hazard ratio for CLINIC after 365 days (output omitted):

```
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 is flexible in the way it can accommodate the modeling of time-varying covariates from different data formats. To illustrate this point, consider an example that was discussed in Chapter 6. The data below (Data D1) contain one observation for Jane who had an event at 49 months (MONTHS=49 and STATUS=1). Her dose of medication at the beginning of follow-up was 60 mg (DOSE1=60 and TIME1=0). At the 12th month of follow-up, her dose was changed to 120 mg (DOSE2=120 and TIME2=12). At the 30th month of follow-up, her dose was changed to 150 mg (DOSE3=120 and TIME3=30).

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

(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

If dosage is measured at multiple time points, then we would want to treat dose as a time varying covariate. We are assuming that Jane's observation is one out of many individuals Å. The following code would run an extended Cox model for data formatted as above:

```
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;
```

The time dependent variable T_DOSE is defined below the MODEL statement and is defined in terms of DOSE1, DOSE2, and DOSE3 at specified points in time.

Alternatively, the data can be transposed in a counting process format such that Jane would have three observations to accommodate her three values of dosage over her risk period.

The following code transposes the data (D1) into a counting process format (D2):

```
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=STATUS;DOSE=DOSE3;OUTPUT;
DROP MONTHS DOSE1 DOSE2 DOSE3 TIME1 TIME2 TIME3 STATUS;
RUN;
```

Now the data (D2) is transposed to contain three observations for Jane, allowing DOSE to be represented as a time-dependent variable. For the first time interval (START=0, STOP=12), Jane's dose was 60 mg. For the second time interval (12–30 months), Jane's dose was 120 mg. For the third time interval (30–49 months), Jane's dose was 150 mg. The data indicate that Jane had an event at 49 months (STOP=49 and STATUS=1). Jane's three observations are printed below:

```
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

The code to run the model with the data in counting process format is shown below:

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

Using PROC PHREG on data in the counting process format is discussed in more detail when we discuss the modeling recurrent events in SAS (Section 8).

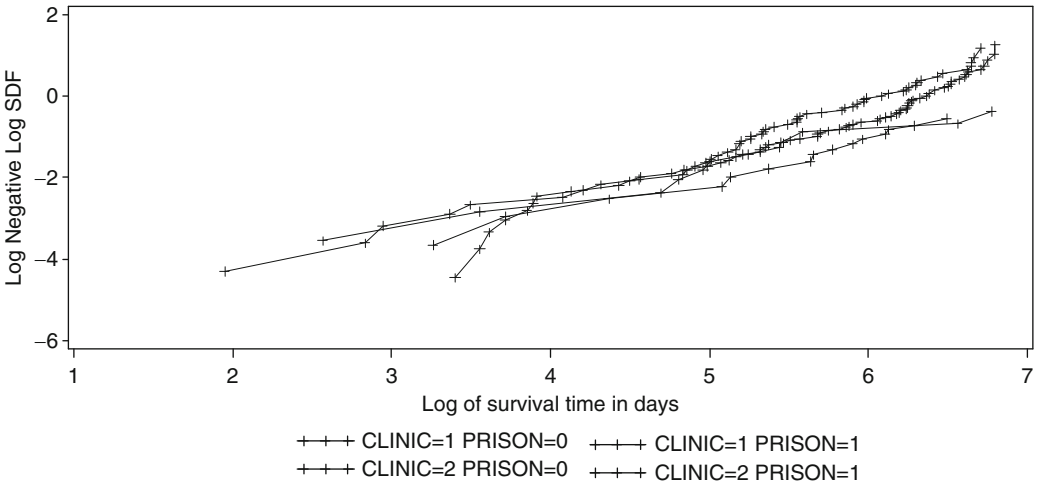
7. RUNNING PARAMETRIC MODELS WITH PROC LIFEREG

PROC LIFEREG runs parametric AFT models rather than PH models. Whereas the key assumption of a PH model is that hazard ratios are constant over time, the key assumption for an AFT model is that survival time accelerates (or decelerates) by a constant factor when comparing different levels of covariates.

The most common distribution for parametric modeling of survival data is the Weibull distribution. The hazard function for a Weibull distribution is $\lambda p t^{p-1}$. If p = 1, then the Weibull distribution is also an exponential distribution. The Weibull distribution has a desirable property, in that if the AFT assumption holds then the PH assumption also holds. The exponential distribution is a special case of the Weibull distribution. The key property for the exponential distribution is that the hazard is constant over time ($h(t) = \lambda$). In SAS, the Weibull and exponential model are run only as AFT models.

The Weibull distribution has the property that the log-log of the survival function is linear with the log of time. PROC LIFETEST can be used to plot Kaplan-Meier log-log curves against the log of time. If the curves are approximately straight lines (and parallel), then the assumption is reasonable. Furthermore, if the straight lines have a slope of 1, then the exponential distribution is appropriate. The code below produces log-log curves stratified by CLINIC and PRISON that can be used to check the validity of the Weibull assumption for those variables:

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



The log-log curves do not look straight but for illustration we shall proceed as if the Weibull assumption were appropriate. First, an exponential model will be run with PROC LIFEREG. In this model, the Weibull shape parameter (p) is forced to equal 1, which forces the hazard to be constant.

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

The DIST=EXPONENTIAL option in the MODEL statement requests the Weibull distribution. The output of parameter estimates obtained from PROC LIFEREG follows:

Analysis of 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		

The exponential model assumes a constant hazard. This is indicated in the output by the value of the Weibull shape parameter (1.0000). The output can be used to calculate the estimated hazard for any subject given a pattern of covariates. For example, a subject with PRISON=0, DOSE=50, and CLINIC=2 has an estimated hazard of $\exp\{-(3.6843 + 50(0.0289)) + 2(0.8806)\} = 0.001$. Note that SAS gives the parameter estimates for the AFT form of the exponential model. Multiply the estimated coefficients by negative one to get estimates consistent with the PH parameterization of the model (see Chapter 7).

Next, a Weibull AFT model is run with PROC LIFEREG.

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

The DIST=WEIBULL option in the MODEL statement requests the Weibull distribution. The output for the parameter estimates follows:

Analysis of 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		

The Weibull shape parameter is estimated at 1.3702. SAS calls the reciprocal of the Weibull shape parameter, the Scale parameter, estimated at 0.7298. The acceleration factor comparing CLINIC=2 to CLINIC=1 is estimated at exp(0.7090) = 2.03. So, the estimated median survival time (time off heroin) is double for patients enrolled in CLINIC=2 compared to CLINIC=1.

To obtain the hazard ratio parameters from the Weibull AFT model, multiply the Weibull shape parameter by the negative of the AFT parameter (see Chapter 7). For example, the HR estimate for CLINIC=2 vs CLINIC=1 controlling for the other covariates is exp(1.3702(-0.7090)) = 0.38.

Next, a log-logistic AFT model is run with PROC LIFEREG.

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

The output of the log-logistic parameter estimates follows:

Analysis of 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		

From this output, the acceleration factor comparing CLINIC=2 to CLINIC=1 is estimated at exp(0.5806) = 1.79. If the AFT assumption holds for a log-logistic model, then the proportional odds assumption holds for the survival function (although the PH assumption will not hold). The proportional odds assumption can be evaluated by plotting the log odds of survival (using KM estimates) against the log of survival time. If the plots are straight lines for each pattern of covariates, then the log-logistic distribution is reasonable. If the straight lines are also parallel, then the proportional odds and AFT assumptions also hold.

A SAS dataset containing the KM survival estimates can be created using PROC LIFETEST (see Section 1 of this appendix). Once this variable is created, a dataset containing variables for the estimated log odds of survival and the log of survival time can also be created. PROC GPLOT can then be used to plot the log odds of survival against survival time.

Another context for thinking about the proportional odds assumption is that the odds ratio estimated by a logistic regression does not depend on the length of the follow-up. For example, if a follow-up study was extended from 3 to 5 years, then the underlying odds ratio comparing two patterns of covariates would not change. If the proportional odds assumption is not true, then the odds ratio is specific to the length of follow-up.

An AFT model is a multiplicative model with respect to survival time or equivalently an additive model with respect to the log of time. In the previous example, the median survival time was estimated as 1.79 **times** longer for CLINIC=2 compared to CLINIC=1. In that example, survival time was assumed to follow a log-logistic distribution or equivalently the log of survival time was assumed to follow a logistic distribution.

SAS allows additive failure time models to be run (see chapter 7 under the heading "Other Parametric Models"). The NOLOG option in the MODEL statement of PROC LIFEREG suppresses the default log link function which means that time, rather than log(time), is modeled as a linear function of the regression parameters. The following code requests an additive failure time model in which time follows a logistic (not log-logistic) distribution:

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

Even though the option DIST=LLOGISTIC appears to request that survival time follows a log-logistic distribution. The NOLOG option actually means that survival time is assumed to follow a logistic distribution. (Note that the NOLOG option in Stata means – something completely different using the **streg** command – that the iteration log file not be shown in the output.) The output from the additive failure time model follows:

Analysis of 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		

The parameter estimate for CLINIC is 214.2525. The interpretation for this estimate is that the median survival time (or time to any fixed value of $S(t)$) is estimated at 214 days more for CLINIC=2 compared to CLINIC=1. In other words, you add 214 days to the estimated median survival time for CLINIC=1 to get the estimated median survival time for CLINIC=2. This contrasts with the previous AFT model in which you multiply estimated median survival time for CLINIC=1 by 1.79 to get the estimated median survival time for CLINIC=2. The additive model can be viewed as a shifting of survival time while the AFT model can be viewed as a scaling of survival time.

If survival time follows a logistic distribution and the additive failure time assumption holds, then the proportional odds assumption also holds. The logistic assumption can be evaluated by plotting the log odds of survival (using KM estimates) against time (rather than against the log of time as analogously used for the evaluation of the log-logistic assumption). If the plots are straight lines for each pattern of covariates, then the logistic distribution is reasonable. If the straight lines are also parallel, then the proportional odds and additive failure time assumptions hold.

Other distributions supported by PROC LIFEREG are the generalized gamma (DIST=GAMMA) and lognormal (DIST=LNORMAL) distributions. If the NOLOG option is specified with the DIST=LNORMAL option in the model statement, then survival time is assumed to follow a normal distribution.

8. MODELING RECURRENT EVENTS

The modeling of recurrent events is illustrated with the bladder cancer dataset (**bladder.sas7bdat**) described at the start of this appendix. Recurrent events are represented in the data with multiple observations for subjects having multiple events. The data layout for the bladder cancer dataset is suitable for a counting process approach with time intervals defined for each observation (see Chapter 8). The following code prints the 12^{th}–20^{th} observation, which contains information for four subjects. The code follows:

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

The output follows:

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

There are three observations for ID=10, one observation for ID=11, three observations for ID=12, and two observations for ID=13. The variables START and STOP represent the time interval for the risk period specific to that observation. The variable EVENT indicates whether an event (coded 1) occurred. The first three observations indicate that the subject with ID=10 had an event at 12 months, another event at 16 months, and was censored at 18 months.

PROC PHREG can be used for survival data using a counting process data layout. The following code runs a model with three predictors – treatment status (TX), initial number of tumors (NUM), and the initial size of tumors (SIZE) – included in the model:

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

The code (START,STOP)*EVENT(0) in the MODEL statement indicates that the time intervals for each observation are defined by the variables START and STOP and that EVENT=0 denotes a censored observation. The ID statement defines ID as the variable representing each subject. The COVS(AGGREGATE) option in the PROC PHREG statement requests robust standard errors for the parameter estimates. The output generated by PROC PHREG follows:

The PHREG Procedure

Model Information

Data Set	WORK.BLADDER
Dependent Variable	START
Dependent Variable	STOP
Censoring Variable	EVENT
Censoring Value(s)	0
Ties Handling	BRESLOW

Analysis of Maximum Likelihood Estimates

Variable	Parameter Estimate	Standard Error	StdErr Ratio	Chi-Square	Pr > ChiSq	Hazard Ratio
TX	-0.40710	0.24183	1.209	2.8338	0.0923	0.666
NUM	0.16065	0.05689	1.185	7.9735	0.0047	1.174
SIZE	-0.04009	0.07222	1.028	0.3081	0.5788	0.961

Coefficient estimates are provided with robust standard errors. The column under the heading StdErrRatio provides the ratio of the robust to the non-robust standard errors. For example, the standard error for the coefficient for TX (0.24183) is 1.209 greater than the standard error would be if we had not requested robust standard errors (i.e., omit the COVS(AGGREGATE) option). The robust standard errors are estimated slightly different compared to the corresponding model in Stata or R.

A stratified Cox model can also be run using the data in this format with the variable INTERVAL as the stratified variable. The stratified variable indicates whether the subject was at risk for their 1st, 2nd, 3rd, or 4th event. This approach is called a Stratified CP approach in Chapter 8 and is used if the investigator wants to distinguish the order in which recurrent events occur. The code for a stratified Cox follows:

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

The only additional code from the previous model is the STRATA statement, indicating that the variable INTERVAL is the stratified variable. The output containing the parameter estimates follows:

Analysis of Maximum Likelihood Estimates

Variable	Parameter Estimate	Standard Error	StdErr Ratio	Chi-Square	Pr > ChiSq	Hazard Ratio
TX	-0.33430	0.19706	0.912	2.8777	0.0898	0.716
NUM	0.11565	0.04991	0.930	5.3690	0.0205	1.123
SIZE	-0.00805	0.06012	0.827	0.0179	0.8935	0.992

Interaction terms between the treatment variable (TX) and the stratified variable could be created to examine whether the effect of treatment differed for the 1st, 2nd, 3rd, or 4th event.

Another stratified approach (called Gap Time) is a slight variation of the stratified counting process approach. The difference is in the way the time intervals for the recurrent events are defined. There is no difference in the time intervals when subjects are at risk for their first event. However, with the Gap Time approach, the starting time at risk gets reset to zero for each subsequent event. The following code creates data suitable for using the gap-time approach:

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

The new dataset (bladder2) copies the data from re.bladder and creates two new variables for the time interval: START2, which is always set to zero and STOP2, which is the length of the time interval (i.e., STOP–START). The following code uses these newly created variables to run a Gap Time model with PROC PHREG:

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

The output follows:

Variable	Parameter Estimate	Standard Error	StdErr Ratio	Chi-Square	Pr > ChiSq	Hazard Ratio
TX	-0.26952	0.20808	1.002	1.6778	0.1952	0.764
NUM	0.15353	0.04889	0.938	9.8620	0.0017	1.166
SIZE	0.00684	0.06222	0.889	0.0121	0.9125	1.007

The results using the Gap Time approach vary slightly from that obtained using the Stratified CP approach.

The counting process data layout with multiple observations per subject need not only apply to recurrent event data, but can also be used for a more conventional survival analyses in which each subject is limited to one event. A subject with four observations may be censored for the first three observations before getting the event in the time interval represented by the fourth observation. This data layout is particularly suitable for representing time-varying exposures (i.e., exposures which change values over different intervals of time).

C. SPSS

Analyses are carried out in SPSS by using the appropriate SPSS procedure on an SPSS dataset. Most users select procedures by pointing and clicking the mouse through a series of menus and dialog boxes. The code, or command syntax, generated by these steps can be viewed and edited.

Analyses on the "addicts" dataset will be used to illustrate these procedures. The addicts dataset was obtained from a 1991 Australian study by Caplehorn et al. and contains information on 238 heroin addicts. The study compared two methadone treatment clinics to assess patient time remaining under methadone treatment. The two clinics differed according to its live-in policies for patients. A patient's survival time was determined as the time (in days) until the person dropped out of the clinic or was censored. The variables are defined at the start of this appendix.

After getting into SPSS, open the dataset **addicts.sav**. The data should appear on your screen. This is now your working dataset. To obtain a basic descriptive analysis of the outcome variable (SURVT), click on Analyze → Descriptive Statistics → Descriptive from the drop-down menus to reach the dialog box to specify the analytic variables. Select the SURVT from the list of variables and enter it into the variable box. Click on OK to view the output. Alternatively, you can click on Paste (rather than OK) to obtain the corresponding SPSS syntax. The syntax can then be submitted (by clicking the button under Run), edited, or saved for another session. The syntax created is as follows (output omitted):

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

There are some analyses that SPSS only performs by submitting syntax rather than using the point and click approach (e.g., running an extended Cox model with two time-varying covariates). Each time the point and click approach is presented, the corresponding syntax will also be presented.

To obtain more detailed descriptive statistics on survival time stratified by CLINIC, click on Analyze → Descriptive Statistics → Explore from the drop-down menus. Select SURVT from the list of variables and enter it into the Dependent List and then select CLINIC and enter it into the Factor List. Click on OK to see the output. The syntax created from clicking on Paste (rather than OK) is as follows (output omitted):

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

Survival analyses can be performed in SPSS by selecting Analyze → Survival. There are then four choices for selection: Life Tables, Kaplan-Meier, Cox Regression, and Cox w/ Time-Dep Cov. The key SPSS procedures for survival analysis are the KM and COXREG procedures.

The survival analyses demonstrated in SPSS are as follows:

1. Estimating survival functions (unadjusted) and comparing them across strata
2. Assessing the PH assumption using Kaplan-Meier log-log survival curves
3. Running a Cox PH model
4. Running a stratified Cox model and obtaining Cox adjusted log-log curves
5. Assessing the PH assumption with a statistical test
6. Running an extended Cox model

SPSS (version PASW 18) does not provide commands to run parametric survival models, frailty models, or models using a counting process data layout for recurrent events.

1. ESTIMATING SURVIVAL FUNCTIONS (UNADJUSTED) AND COMPARING THEM ACROSS STRATA

To obtain Kaplan-Meier survival estimates, select Analyze → Survival → Kaplan-Meier. Select the SURVT from the variable list and enter it into the Time box, then select the variable STATUS and enter it into the Status box. You will then see a question mark in parentheses after the status variable, indicating that the value of the event needs to be entered. Click the Define Event button and insert the value 1 in the box since the variable STATUS is coded 1 for events and 0 for censorships. Click on Continue and then OK to view the output. The syntax, obtained from clicking on Paste (rather than OK), is as follows (output omitted):

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

The stream of output of these KM estimates is quite long. If you wish to edit the output, try right clicking inside the output and then select Edit Content. You then have a choice to select In Viewer or In Separate Window. Click on one of these depending on if you want to open a separate window for your edited output.

To obtain KM survival estimates and plots by CLINIC as well as log rank (and other) test statistics, select Analyze → Survival → Kaplan-Meier and then select SURVT as the time-to-event variable and STAUS as the status variable as described above. Enter CLINIC into the Factor box and click the Compare Factor button. You have a choice of three test statistics for testing the equality of survival functions across CLINIC. Select all three (log rank, Breslow, and Tarone–Ware) for comparison and click Continue. Select the Options button to request plots. There are four choices (unfortunately, log-log survival plots are not included). Select Survival to obtain KM plots by clinic. Click Continue and then OK to view the output.

The syntax follows:

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

The output containing the KM estimates for the first five events or censorship times from CLINIC=1 and CLINIC=2 as well for the log rank, Breslow, and Tarone–Ware tests follow:

Survival Analysis for SURVT Survival time (days)

Survival Table

clinic	Time	Status	Cumulative Proportion Surviving at the Time		N of Cumulative Events	N of Remaining Cases
			Estimate	Std. Error		
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
```

Note that what SPSS calls the Breslow test statistic is equivalent to what Stata (and SAS) call the Wilcoxon test statistic.

Life Table estimates can be obtained by selecting Analyze → Survival → Life Tables. The time-to-event and status variables are defined similarly as described above for KM estimates. However with life tables, SPSS presents a Display Time Intervals box. This allows the user to define the time intervals used for the life table analysis. For example, 0 to 1,000/100 would define 10 time intervals of equal length. Life table plots can similarly be requested as described above for the KM plots.

2. ASSESSING THE PH ASSUMPTION USING KAPLAN-MEIER LOG-LOG SURVIVAL CURVES

SPSS does not provide unadjusted KM log-log curves by directly using the point and click approach with the KM command. SPSS does provide adjusted log log curves from running a stratified Cox model (described later in the stratified Cox section). A log-log curve equivalent to the unadjusted KM log-log curve can be obtained in SPSS by running a stratified Cox without including any covariates in the model. In this section, however, we illustrate how new variables can be defined in the working dataset and then used to plot unadjusted log-log KM plots.

First, a variable will be created containing the KM survival estimates. Then, another new variable will be created containing the log-log of the survival estimates. Finally, the log-log survival estimates will be plotted against survival time to see if the curves for CLINIC=1 and CLINIC=2 are parallel. Each step can be done with the point and click approach or by typing in the code directly.

A variable containing the survival estimates can be created by selecting Analyze → Survival → Kaplan-Meier and then selecting SURVT as the time-to-event variable, STAUS as the status variable, and CLINIC as the factor variable as described above. Then click the Save button. This opens a dialogue box called Kaplan-Meier Save New Variables. Check Survival and click on Continue and then on Paste. The code that is created is as follows:

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

By submitting this code, a new variable containing the KM estimates called **SUR_1** is created. To create a new variable called **lls** containing the log(-log) of **SUR_1**, submit the following code:

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

The above code could also be generated by selecting Transform → Compute Variable and defining the new variable in the dialogue box. To plot **lls** against survival time, submit the code:

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

This final piece of code could also be run by selecting Graphs → Legacy Dialogue → Scatter/Dot → and then clicking on Simple Scatter and then Define in the Scatter/Dot dialogue box. Select LLS for the Y-axis, SURVT for the X-axis, and CLINIC in the Set Marker By box. Clicking on paste creates the code or clicking OK submits the program. A plot of LLS against log(SURVT) could similarly be created. Parallel curves support the PH assumption for CLINIC.

3. RUNNING A COX PH MODEL

A Cox PH model can be run by selecting Analyze → Survival → Cox Regression. Select the SURVT from the variable list and enter it into the Time box, then select the variable STATUS and enter it into the Status box. You will then see a question mark in parentheses after the status variable, indicating that the value of the event needs to be entered. Click the Define Event button and insert the value 1 in the box since the variable STATUS is coded 1 for events and 0 for censorships. Click on Continue and select PRISON, DOSE, and CLINIC from the variable list and enter them into the Covariates box. You can click on Plots or Options to explore some of the options (e.g., 95% CI for exp(β)). Click OK to view the output or click on Paste to see the code. The code follows:

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

Note that the PH assumption is assumed to hold for all three covariates using this Cox model (the output follows).

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. RUNNING A STRATIFIED COX MODEL AND OBTAINING COX ADJUSTED LOG-LOG CURVES

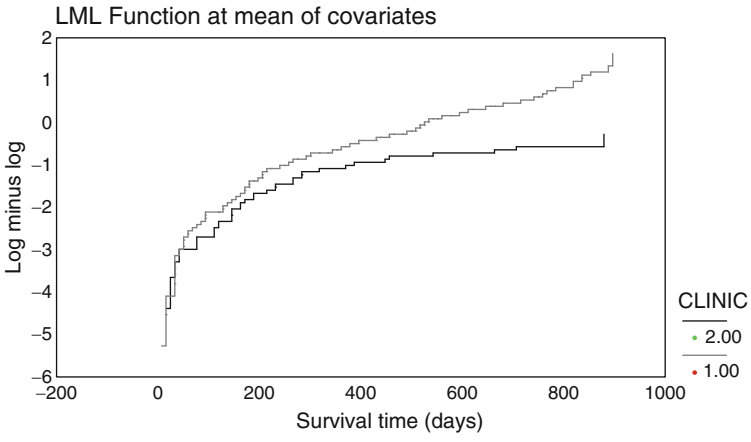
A stratified Cox model is run by selecting Analyze → Survival → Cox Regression. Select the SURVT from the variable list and enter it into the Time box. Select the variable STATUS and enter it into the Status box and then define the value of the event as 1. Put the variables PRISON and DOSE in the Covariates box and the variable CLINIC in the Strata box. The Cox model will be stratified by CLINIC. Click the Plots button and check Log minus log as the plot type and then click on Continue. Click on OK to view the output or click on Paste to see the code. The code follows:

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

The output containing the parameter estimates and the adjusted log log plots follows:

Variables in the Equation

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



Notice that there are parameter estimates for PRISON and DOSE but not CLINIC since CLINIC is the stratified variable. The Cox-adjusted log log plots are fitted using the mean values of PRISON and DOSE and are used to evaluate the PH assumption for CLINIC.

Suppose rather than using the mean value of DOSE for the adjusted log log plots, you wish to obtain adjusted plots in which DOSE=70. Run the same code as before, up to clicking on the Plots button and checking Log minus log as the plot type. Instead, click on DOSE(Mean) in the window called Covariate Values Plotted at. Underneath the heading called Change Value, click on the word Value, type in the value 70, and then click on the button called Change. Now, the variable in the window should be called DOSE(70) rather than DOSE(Mean). Click on Continue and then OK to view the output.

5. ASSESSING THE PH ASSUMPTION WITH A STATISTICAL TEST

SPSS does not easily accommodate a statistical test on the PH assumption using the Schoenfeld residuals. However, it can be programmed using several steps. The steps are as follows:

1. Run a Cox PH model to obtain the Schoenfeld residuals for all the covariates. These residuals are saved as new variables in the working dataset.
2. Delete observations that were censored.
3. Create a variable that contains the ranked order of survival time. For example, the subject who had the fourth event gets a value of 4 for this variable.
3. Run correlations on the survival rankings with the Schoenfeld residuals.
4. The p-value for testing whether the correlation between the ranked survival time and the covariate's Schoenfeld residuals is zero is the same p-value used to test the PH assumption. The null hypothesis is that the PH assumption is not violated.

First, run a Cox PH model with CLINIC, PRISON, and DOSE. Click on the Save button before submitting the model. A dialogue box appears that is called Cox Regression: Save Model Variables. Check Partial Residuals and click on Continue. This creates three new variables in the working dataset called **PR1_1**, **PR2_1**, and **PR3_1**, which are the partial residuals (Schoenfeld residuals) for CLINIC, PRISON, and DOSE, respectively. Click OK to run the model (or Paste to generate the code).

Next, delete all censored observations (i.e., only keep observations in which STATUS=1). To do this, select Data → Select Cases. Then check If condition is satisfied, and then click on If. Type status=1 in the dialogue box and click on Continue. Check Delete unselected cases in the box called Output. Click OK and only observations with events will be kept in the dataset. (Remember to go back to the addicts dataset that contains the censored observations when you continue work through the other sections that use the addicts data.)

Create the variable that contains the ranking of survival times by selecting Transform → Ranked Cases. Select the SURVT into the Variables box. Click on Rank Types, check Ranks, and click on Continue and then click on Ties, check Mean, and click Continue. Click OK and a new variable (called **Rsurvt**) will be created containing the ranked survival time.

Finally, obtain correlations (and their p-values) between the ranked survival and the Schoenfeld residuals. Select Analyze → Correlate → Bivariate. Move the ranked survival time variable as well as the three partial residual variables into the variable box. Check Pearson (for Pearson correlations) and Two-tailed for a two-tail test of significance and click OK to see the output. The code that is generated from these steps follows:

```
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.
```

The output containing the correlations follows:

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

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

The p-values for the correlations are the p-values for the PH test. In the output, examine the row labeled RANK of SURVT Sig(2-tailed). Notice that the null hypothesis is rejected for CLINIC (p = 0.001) but not for PRISON (p = 0.332) or DOSE (p = 0.347).

6. RUNNING AN EXTENDED COX MODEL

An extended Cox model with exactly one time-dependent covariate can be run using the point and click approach. Suppose we want to include a time-dependent covariate DOSE times the log of survival time. This product term could be appropriate if the hazard ratio comparing any two levels of DOSE monotonically increases (or decreases) over time. Select Analyze → Survival → Cox w/ Time-Dep Cov. This opens a dialogue called Expression for T_COV_. The user defines a time-dependent variable (called T_COV_) in this box. A variable T_ is included in the variable list. This is the variable that represents time-varying survival (as opposed to SURVT which is an individual's fixed time of event). We wish to define T_COV_ to be the log of T_ × DOSE. Enter the expression LN(T_)*dose into the dialogue box and click on the Model button. Now, run a Cox model that includes the covariates: PRISON, CLINIC, DOSE, and T_COV_. The code generated is as follows:

```
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).
```

The output containing the parameter estimates follows:

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

The variable T_COV_ represents the time-dependent variable included in the model, which in this example is DOSE times the log of survival time.

A heaviside function for CLINIC can similarly be created. We can define a time dependent variable equal to CLINIC if time is greater than or equal to 365 days and 0 otherwise. Select Analyze → Survival → Cox w/ Time-Dep Cov. Define T_COV to be (T_ × 365) × clinic. After clicking on the Model button, run a Cox model that includes PRISON, DOSE, CLINIC, and T_COV_. The code generated is as follows:

```
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).
```

Note that SPSS recognizes the expression (T_ × 365) as taking the value 1 if survival time is ≥365 days and 0 otherwise.

The output follows:

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

Notice the variable CLINIC is included in this model and the time-dependent heaviside function, T_COV_, does not contribute to the estimated hazard ratio until day 365. The estimated hazard ratio for CLINIC at 100 days is exp(-0.460) = 0.632 while the estimated hazard ratio for CLINIC at 400 days is exp((-0.460) + (-1.369)) = 0.161.

It may be of interest to define two heaviside functions (with CLINIC) and not include CLINIC in the model. This is essentially the same model as the one described above with one heaviside function. However, the coding of two heaviside functions makes it somewhat computationally more convenient for estimating the two hazard ratios for CLINIC (HR for <365 days and HR for ≥ 365 days). Unfortunately, SPSS allows just one time-dependent variable (i.e., T_COV_) using the point and click approach. However, by examining the code created for the single heaviside function, there is only a slight adjustment needed to create code for two heaviside functions. The following code creates two heaviside functions (called HV1 and HV2) and runs a model containing PRISON, DOSE, HV1, and HV2:

```
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).
```

The output follows:

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

The parameter estimates for HV1 and HV2 can be used directly to obtain the estimated hazard ratio for CLINIC=2 vs CLINIC=1 before and after 365 days. The estimated hazard ratio for CLINIC at 100 days is exp(-0.460) = 0.632 and the estimated hazard ratio for CLINIC at 400 days is expexp(-1.828) = 0.161. These results are consistent with the estimates obtained from the previous model with one heaviside function.

D. R Software

R is available for free and can be downloaded from the Comprehensive R Archive Network (CRAN) at its home site at http://www.r-project.org/. Analyses are carried out in R by applying functions on R data (stored as R objects). R functions are stored in packages. Only when a package is loaded, its contents are available. The base packages are installed when you download R. Packages that are not base packages need to be installed separately.

Once you open R, you'll see a prompt: Type 1+1 and press enter. You'll (hopefully) see the answer 2 returned at the line below. Alternatively, you can type commands in a script by clicking on File → New script. A new script window will open up. By typing commands in this window, you can submit batches of code at one time by highlighting the code and clicking on Edit → Run line or by clicking on Edit → selection. Programming in a script window serves a similar function as the program editor in SAS or the Do-file Editor in Stata, in that code can be submitted as a block rather than one line at a time.

To see which packages are installed at your site, type and enter **library( )**. To run many of the functions needed to perform survival analyses, you will need to install the survival package (not generally a base package).

To install the survival package, click on Packages → Install package(s). You will see a heading called CRAN mirror with a listing of many different countries under that heading. Click on one of these (e.g., USA (AZ)) and then scroll down and click on survival and then click OK. The survival package (with its many survival functions) should now be installed. Type **library(survival)** and press enter, and the survival package will be ready to use. As a check, type the word kidney and hit enter. A dataset called kidney (which is part of the survival package) should print on your screen. Once the survival package is installed, you do not have to reinstall it each session. However, you will need to type **library(survival)** each session before you run the survival functions contained in the package.

Before discussing survival analyses in R, it may be useful to give a brief overview on some of the ways data are stored in R. In particular, we describe four classes of data storage: **vectors**, **matrices**, **dataframes**, and **lists**. If you type and enter the code below, R will create a numerical vector with five elements:

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

The **c** function combines its arguments to form a vector. We can store this vector as an object under the name (identifier), **x1**:

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

Type **x1** and press enter, and you will see the vector **x1** printed as output. The code and output are shown below:

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

We can identify elements from the vector **x1** by placing brackets **[ ]** after **x1**. For example, **x1[2]** will identify the 2nd element of **x1**. The code **x1[1:3]** will identify the first three elements of **x1** and the code **x1[x1>6]** will identify the elements in **x1** greater than 6. The code and output for these three examples follow:

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

The **:** operator (used in the 2nd example) creates a sequence of integers incremented by 1. Next, we create four more vectors called **x2**, **x3**, **x4**, and **x5**:

```
x2=2*(1:5)
x3=2*x1 + x2
x4=x1>z6
x5=c("blue","green","red","green","purple")
```

The code **x2=2*(1:5)** creates the vector 2, 4, 6, 8, and 10 which we named **x2**. The vector **x3** results from arithmetic operations of **x1** and **x2** ($2 \times$ **x1** $+$ **x2**). The vectors **x1**, **x2**, and **x3** are each numeric vectors. If you apply the **mode** function on **x1** (i.e., type **mode(x1)** and press enter), it will return the word "numeric" as output. The vector **x4** is a logical vector. The **mode** function will return the word "logical" if you submit the code **mode(x4)**. The elements of a vector of mode logical are either "TRUE" or "FALSE." Enter the code **x4** (output below):

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

The 2nd and 3rd elements of **x4** are TRUE because the 2nd and 3rd elements of **x1** are greater than 6. The vector **x5** is a character vector. R is case sensitive, so naming the vector **x5** is not the same as naming it **X5**.

We can create a numeric **matrix** (called **y**) using the vectors **x1**, **x2**, and **x3** as columns of the matrix by applying the **cbind** function:

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

Enter the code **class(y)** and the word "matrix" will be returned as output. Enter the code **mode(y)** and the word "numeric" will be returned since **y** is a numeric matrix. You cannot mix numeric and character vectors in a matrix.

Type **y** and press enter, and the matrix will be printed (shown below):

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

A dataframe provides a more general class of data storage than a matrix in R because a dataframe can contain a mix of numeric, character, and logical variables. A dataframe in R is similar to a dataset in Stata, SAS, or SPSS, in that it can store different types of variables. The **data.frame** function can be used to combine vectors and matrices as follows:

z=data.frame(x1,x2,x3,x4,x5) or, equivalently, **z=data.frame(y,x4,x5)**

Type **z** and press enter, and the dataframe will be printed (shown below):

```
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
```

Brackets [] can be used to access particular rows and/or columns of a dataframe or matrix. Enter the code: **z[2,5]** and the 2nd row, 5th column will be printed from the matrix **z** (the element "green" in this example). If you want to access the first three rows (observations) of the fifth column, type the code **z[1:3,5]** or equivalently **z[c(1,2,3),5]**. If you want to access the entire 5th column, enter **z[,5]**. Alternatively, since the 5th column (or variable) is named **x5**, you can access the entire 5th column by entering the code **z$x5**. The **$** in this example points to the variable named **x5** from the dataframe named **z**.

A list offers a more general type of data storage than the vector, matrix, or dataframe, and can include any of those data objects as part of the list. The following code creates a list called **w** that contains a character vector of length 2 as its first element, the vector **x1** as its second element, the matrix **y** as its third element, and the dataframe **z** as its fourth element:

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

Double brackets **[[ ]]** can be used to access particular elements of a list. If you want to access the dataframe **z** from the list **w**, enter the code **w[[4]]** since **z** is the fourth element of **w**. If you want to access the first row third column of the fourth element of **w** from the list, enter the following code:

w[[4]] [1,3]

The 1st row 3rd column of the 4th element of w has the value 4.

Survival Functions in R

Once the survival package has been installed you will have access to the survival functions needed to perform the survival analyses in this appendix. Enter the code **library (survival)** each session to access these functions. Some of the key survival functions are listed below:

Surv – Used to define the "time-to-event" and "status" outcome variables. This function creates a survival object that can be used as the outcome variable for other survival functions in R

survfit – Produces KM or Cox-adjusted survival estimates or survival estimates from a previously fitted parametric model

survdiff – Used to perform statistical tests for the equality of survival functions across strata

coxph – Used to run a Cox PH model, a stratified Cox model, or an extended Cox model

cox.zph – Performs statistical tests on the PH assumption based on Schoenfeld residuals

survSplit – Creates a new dataset in the counting process format, with a start time, stop time, and event status for each record. Splits single observations into multiple observations given survival data and specified cut times

survreg – Used to run parametric survival models

Generic functions in R such as the **summary** function and the **plot** function are often used in conjunction with these survival functions in order to produce survival estimates and plots.

R documentation (online help) for these functions can be obtained by typing and submitting a question mark and then the name of the function as one word. For example, to access R documentation on the **coxph** function, submit the code **?coxph**.

The survival analyses demonstrated in R are as follows:

1. Estimating survival functions (unadjusted) and comparing them across strata.
2. Assessing the PH assumption using graphical approaches.
3. Running a Cox PH model.
4. Running a stratified Cox model.
5. Assessing the PH assumption with a statistical test.
6. Obtaining Cox-adjusted survival curves.

7. Running an extended Cox model.
8. Running parametric models.
9. Running frailty models.
10. Modeling recurrent events.

We use the addicts dataset for illustration. The **load** function is used to access an R dataframe that has been saved as a file. Suppose the addicts dataset has been saved on your C drive as C:\craddicts.rda. The following code will load the addicts data:

load("C:\craddicts.rda")

To print the addicts dataset, enter the code:

addicts

To print the first five observations, enter the code:

addicts[1:5,]

All 6 variables (columns) are printed because there was no entry after the comma. Equivalently, we could have entered the code **addicts[1:5,1:6]**. The output follows:

	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

The time-to-event variable in the addicts dataset is named SURVT and the variable indicating whether a subject had an event or was censored is named STATUS. The function **Surv** creates a survival object in R linking these two outcome variables (code shown below):

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

The first argument is the time-to-event variable which is accessed from the addicts dataframe with the **$** notation (**addicts$survt**). The second argument (**addicts$status==1**) indicates an event occurs (as opposed to a censorship) when the status variable equals 1. Notice that two equal signs are used to express equality. A single equal sign is used to designate assignment in R. A portion of the output from the **Surv** function is shown below:

```
[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+
```

The output above shows the survival times for the first 36 subjects in the addicts data (out of 238). A plus (+) sign after their time indicates censorship rather than event.

This survival object created by the **Surv** function is often used in R as the response variable for survival analyses. Next we demonstrate survival analyses in R by specific topics.

1. ESTIMATING SURVIVAL FUNCTIONS (UNADJUSTED) AND COMPARING THEM ACROSS STRATA

Kaplan-Meier survival estimates are obtained in R with the use of three functions. The **Surv** function (described above) is used within the **survfit** function, which is then used within the **summary** function. The code follows:

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

To better understand how this code works we'll break down each function. The code: **Y=Surv(addicts$survt, addicts$status==1)** creates a survival object called **Y** that is used as the response variable in the analysis. Now consider the code **Y∼1** This syntax is called a formula. Formulas are used as arguments in many functions in R, particularly those that specify statistical models. **Y∼1** requests an intercept only model. In other words, we are not conditioning on any other variable. Later in this section we stratify on the variable CLINIC and use the formula **Y∼ addicts$clinic**. A formula needs to be supplied as the argument of the **survfit** function (shown below):

kmfit1=survfit(Y∼1)

An object, which we named **kmfit1**, was created with the **survfit** function. Enter the code **kmfit1** and press enter (output shown below):

kmfit1

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

The output contains descriptive information on the number of records, the number at risk at time 0, the number of events, and the median estimated survival time with a 95% confidence interval. The **summary** function can then be used to get Kaplan-Meier survival estimates for all event times. The code **summary(kmfit1)** is equivalent to the code **summary(survfit(Surv(addicts$survt, addicts$status==1)∼1))** shown above. The output follows:

```
   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
```

The **summary** function can also produce survival estimates for specified survival times (e.g., at day 365) with the **times=** option. Code and output follow:

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
```

If we wish to stratify by the variable CLINIC and compare the Kaplan-Meier survival estimates at specified times, we can first create an object (called **kmfit2**, where the name is arbitrary) from the **survfit** function:

kmfit2=survfit(Y∼addicts$clinic)

To get survival estimates at specified times (every 100 days) for each level of CLINIC, enter the code:

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

The output follows:

```
                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
```

Survival estimates are supplied for each 100th day. For CLINIC=1, survival times stopped at 900 rather than 1000 as requested because no subject was at risk on day 1000. The second argument of the summary function requesting a vector of survival times could have been equivalently written: **summary(kmfit2,times=100*(0:10))**. The output would be identical if this alternative syntax had been used.

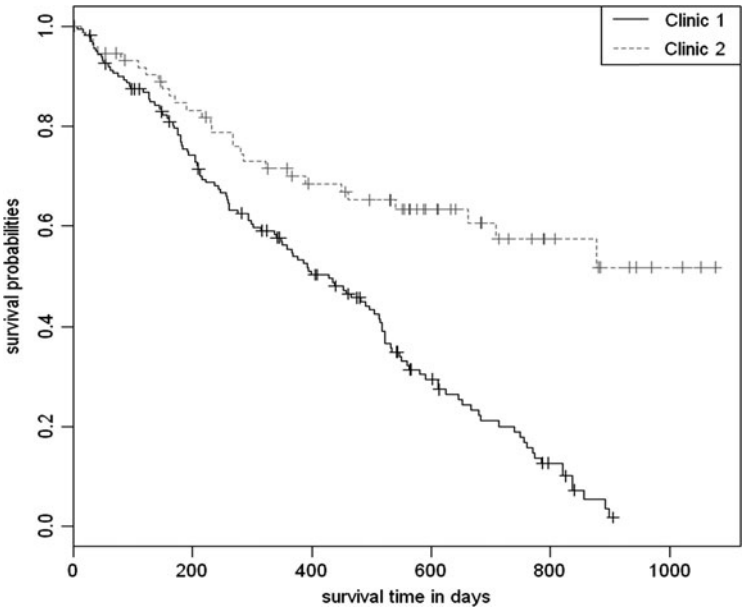
KM survival plots can be obtained using the **plot** function:

```
plot(kmfit2)
```

There are many plotting options that can be applied with the **plot** function. The code below requests different line types (**lty=**) and different colors (**col=**) for CLINIC=1 and CLINIC=2 as well as labels for the X and Y axes (**xlab=** and **ylab=**). If the code **col( )** is submitted, then R returns a list of over 600 colors that can be selected with the **col=** option. The **legend** function is used to add a legend. The first argument, "**topright**," places the legend at the top right part of the graph. The code and output follow:

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"))



The plot indicates that subjects from CLINIC=2 have a higher rate of survival than subjects from CLINIC=1.

The **survdiff** function can be used to implement a log rank test on the variable CLINIC (the code follows):

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

The second argument of the **survdiff** function, **data= addicts**, indicates that the variables come from the addicts dataset. Alternatively, you could use the code:

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

As a third alternative, the **attach** function can be used to indicate that all subsequent variable names apply to the addicts dataset (R will search the addicts dataset for variables). The **detach** function can be used to remove a dataset from the search path.

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

The output follows:

```
           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
```

The log rank statistic is highly significant with a *p*-value of 0.000000128 (i.e., 1.28e-07).

Variations of the log rank test can be obtained by using the **rho=** option as an argument in the **survdiff** function. The contribution of the j^{th} failure time to the test statistic is weighted by $s(t_j)^{rho}$, where $s(t_j)$ represents the KM survival estimates at time t_j. If rho=0, then each failure time is equally weighted since $s(t_j)^0 = 1$ and the resulting test is the log rank test. If rho=1, then the weights for each failure time are the KM survival estimate at that failure time since $s(t_j)^1 = s(t_j)$. This test is equivalent to the Peto & Peto modification of the Gehan–Wilcoxon test. The code and output with rho=1 follows:

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
```

The results of the test in which rho=1 yield a chi-square value of 15.8 with a p-value of 0.0000718. This is a somewhat different result than the log rank test but still shows a highly significant effect of CLINIC on survival.

A stratified log rank test for CLINIC (stratified by PRISON) can be run with the + **strata(prison)** term included in the model formula. With the stratified approach, the observed minus expected number of events are summed over all failure times for each group within each stratum and then summed over all strata. The code and output follow:

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
```

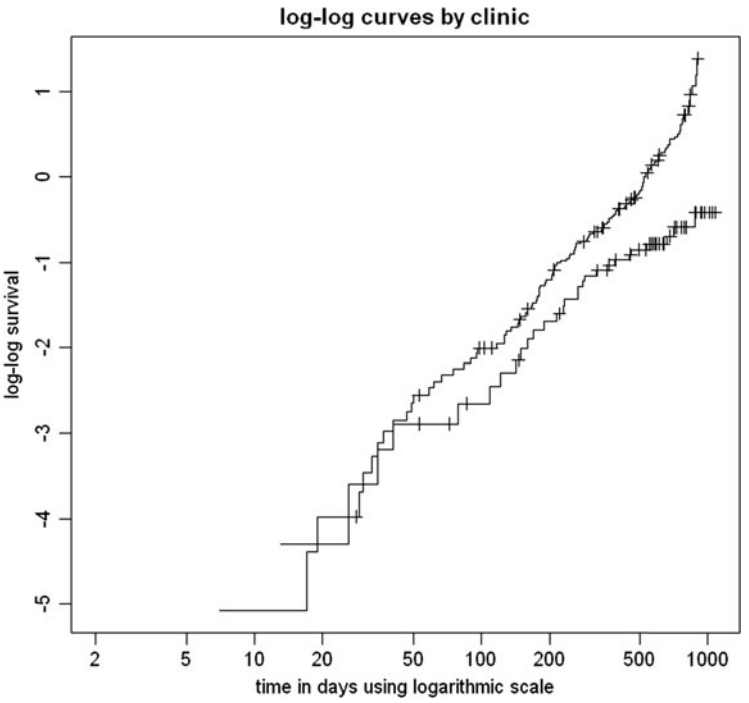
The formula includes the term **+ strata(prison)** in the **survdiff** function. The result of this test is very similar to that obtained from the log rank test without stratifying on PRISON.

2. ASSESSING THE PH ASSUMPTION USING GRAPHICAL APPROACHES

The proportional hazards assumption for CLINIC can be assessed by plotting log-log Kaplan Meier survival estimates against time (or against the log of time) and evaluating whether the curves are reasonably parallel. Recall that a survival object, called **kmfit2**, was created in the previous section with the **survfit**. The code **plot(survfit2)** was used to plot the survival estimates against time. The **fun="cloglog"** option in the **plot** function requests that log-log survival plot be plotted against time (on the log scale). The code follows:

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

The **xlab=** and **ylab=** request labels for the *x*- and *y*-axes and the **main=** option requests a title. **fun="cloglog"** requests the complimentary log log function. The output follows:



The plot suggests that the proportional hazards assumption is violated as the log-log survival curves are not parallel. The **fun=** option (**fun** denotes function) plots time on a logarithmic scale. It is not so straightforward if you want the log-log survival estimates plotted against time with time not on a logarithmic scale. However, it may be useful to program this task in order to illustrate how analytic output can be saved, manipulated, and plotted. To do that, we first save the survival estimates as an object (which we will call **kmfit3**) using the **summary** function:

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

If we submit the code **names(kmfit3)**, then the column names of the object **kmfit3** are printed. We are interested in the columns that indicate each subject's survival time, KM survival estimate, and level of clinic (1 or 2). With the **names** function, we can see that these columns are called **time**, **surv**, and **strata**. We can examine any of these three columns by submitting **kmfit3$strata**, **kmfit3$time**, or **kmfit3$surv** as code. A dataframe (called **kmfit4**) consisting of these three columns as variables can be created with the **data.frame** function:

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

The **names** function is used (above) on **kmfit4** to overwrite the default variable names. Next, we'll print the first 5 observations of **kmfit4**:

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

	clinic	time	survival
1	addicts$clinic=1	7	0.9938272
2	addicts$clinic=1	17	0.9876543
3	addicts$clinic=1	19	0.9814815
4	addicts$clinic=1	29	0.9752300
5	addicts$clinic=1	30	0.9689785

We are interested in separating out CLINIC=1 and CLINIC=2. Below, we create two dataframes (**clinic1** and **clinic2**) from **kmfit4**:

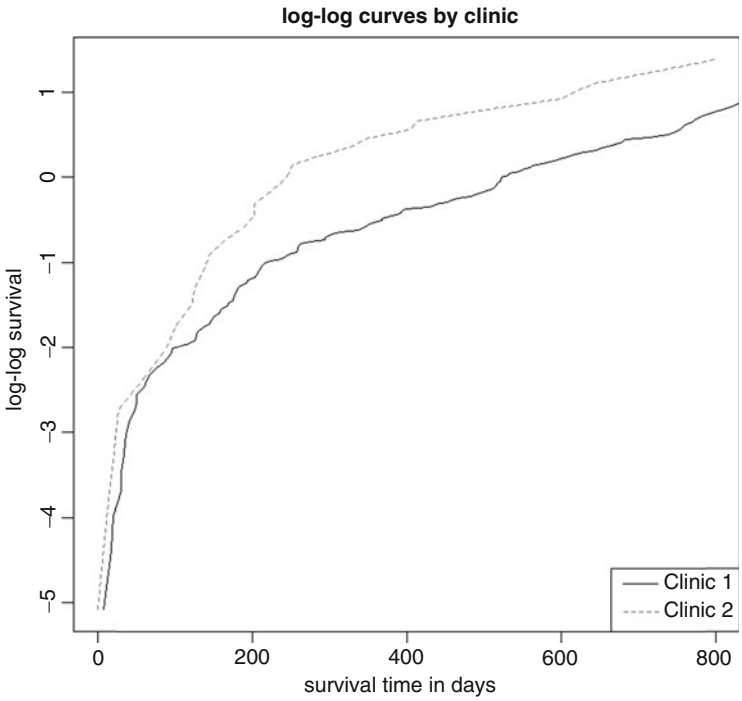
```
clinic1=kmfit4[kmfit4$clinic=="addicts$clinic=1", ]
clinic2=kmfit4[kmfit4$clinic=="addicts$clinic=2", ]
```

The dataframes **clinic1** and **clinic2** contain the survival times and survival estimates for those in CLINIC=1 and CLINIC=2, respectively. We can now use the **plot** function

to plot the log-log survival curves against time (with time not plotted on the log scale). The code follows:

```
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)
```

In the first plot, time (**clinic1$time**) is plotted on the x axis, and the log(-log) of survival (**clinc1$survival**) is plotted on the y axis using the dataframe **clinic1**. The code **par (new=T)** requests that the first plot not get erased when the second plot is requested (i.e., the two plots will be overlayed). The **par** function is used to set or query graphical parameters. The second **plot** function is similar to the first except that the data that is plotted are from the dataframe **clinic2**. A legend is added with the **legend** function and finally **par(new=F)** sets the graphical parameter **new** back to its default value of false (so that these plots will be erased when the next plot is requested). The output follows:



The plot suggests that the proportional hazards assumption is violated for CLINIC.

3. RUNNING A COX PH MODEL

The **coxph** function is used to run a Cox proportional hazards model. First, the response variable is created with the **Surv** function and then a Cox PH model containing the variables CLINIC, PRISON, and DOSE is run with the **coxph** function. The code and the **coxph** output follow:

```
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
```

The output contains the regression coefficients, the exponentiated coefficients (estimated hazard ratios), as well as the standard errors, z-tests, and corresponding p-values for the coefficients. Additional output including 95% confidence intervals can be obtained by applying the **summary** function to the **coxph** function (code and output shown below):

```
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 ' ' 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
```

The second table of the output gives the estimated hazard ratio, under the column exp(coef), for CLINIC=2 vs CLINIC=1 at 0.3643 with 95% CI (0.2391, 0.5550). Under the column exp(-coef), we see that the estimated hazard ratio for CLINIC=1 vs CLINIC=2 is 2.7453 (the reciprocal of 0.3643).

If any of the events in the data occur at the same time, there are several options for handling ties in the Cox likelihood. R offers three approaches with the **coxph** function 1) the Efron method (the default), 2) the Breslow method, and 3) the exact method. Generally, these methods have little impact on the estimates but model results obtained from different software packages may differ depending on the default tie handling method. R uses the Efron method as the default while Stata, SAS and SPSS use the Breslow method as the default. The **method=** option in the **coxph** function is used to specify the method for handling ties (code shown below, output omitted):

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")

Next we include two interaction (product) terms with PRISON and test the significance of the interaction terms simultaneously with a likelihood ratio test. The following code creates two objects (called **mod1** and **mod2**) that contain information obtained from the **coxph** function for the no interaction model (**mod1** – the reduced model) and the model with the two interaction terms (**mod2** – the full model)

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

Enter the code **mod2** to see the output with the interaction terms (code and output shown below):

mod2

```
                coef exp(coef) se(coef)      z     p
prison         1.1920     3.294   0.5414  2.202 0.028
dose          -0.0193     0.981   0.0194 -0.998 0.320
clinic         0.1747     1.191   0.8931  0.196 0.840
prison:clinic -0.7380     0.478   0.4315 -1.710 0.087
dose:clinic   -0.0139     0.986   0.0143 -0.967 0.330

Likelihood ratio test=68.1  on 5 df, p=2.52e-13  n= 238
```

The rest of this section gets a little complicated but we include it to demonstrate how analytic output in R can be accessed and then manipulated.

The objects **mod1** and **mod2** contain information that we may wish to utilize. Type the code **names(mod2)** to see the names of the elements in **mod2** (code and output shown below):

names(mod2)

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

The 3rd element of **mod2** is named "loglik." We can access the data stored under this name by entering the code **mod2$loglik** or equivalently **mod2[[3]]** since loglik is the 3rd element in the list (code and output follow):

mod2$loglik
-705.6619 -671.5997

The second element of **mod2$loglik** is −671.5997, which is the log likelihood of the two interaction terms in the model. The first element of -704.6619 is the log likelihood of a model that contains none of the predictors (not of interest right now).

Next we wish to perform a likelihood ratio test on the two interaction terms. To calculate the test statistic, we need to subtract the log-likelihood of the full model (with the interaction terms) from the reduced model (without the interaction terms) and multiply that difference by negative 2. We can obtain this by entering the following code:

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

We get the output: 3.605457, which is the likelihood ratio test statistic. Under the null, this test statistic follows a chi-square distribution with two degrees of freedom. We can use the **pchisq** function to obtain a p-value for this test. The code **1** × **pchisq(3.605457,2)** returns the p-value for a two degree of freedom chi-square test. In summary, the following code will produce a p-value for the likelihood ratio test (output follows):

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

The *p*-value of 0.168485 is not significant at the 0.05 level of significance.

One of the powerful features of R is the ability for users to define their own functions. We illustrate this feature by defining our own function that performs a likelihood ratio test from two Cox models (a full and reduced model). The following code creates a function which we call **lrt.surv**. This function requests the user to enter three arguments (1) the name of the full model, (2) the name of the reduced model, and (3) the degrees of freedom for the test. The function will return the p-value for the likelihood ratio test.

An R function called **function** is used to define a new function. The three arguments for this function we call **mod.full**, **mod.reduced**, and **df**. The code that R will use to calculate the function output is contained within brackets **{ }** after the arguments are listed. The argument in the R function called **return** informs R of the output that we wish to return from this function (in this example, the p-value for the likelihood ratio test). The code follows:

```
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)
}
```

Once this code is submitted any user can obtain a p-value from a likelihood ratio test from two Cox models by invoking the function **lrt.surv**. We invoke this new function by performing the same likelihood ratio test that we previously ran for the objects **mod1** and **mod2**. The code and output follow:

```
lrt.surv(mod1, mod2, 2)
0.1648485
```

The p-value is the same as that which we obtained earlier. The function **lrt.surv** is more general and now available to simply obtain p-values for *other* likelihood ratio tests that compare two (full and reduced) Cox models.

4. RUNNING A STRATIFIED COX MODEL

If the proportional hazards assumption is violated for the variable CLINIC but met for PRISON and DOSE, a stratified Cox model can be performed with CLINIC the stratified variable. The **coxph** function includes a **strata()** option in the model formula. First we define the response variable **Y** with the **Surv** function and then the **coxph** function is used to run a stratified Cox model (code and output shown below):

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
```

Interaction terms for CLINIC can be included directly in the model formula by including product terms using the **:** operator (**clinic:prison** and **clinic:dose**) (code and output follow):

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
```

Suppose we wish to estimate the hazard ratio for PRISON=1 vs. PRISON=0 for CLINIC=2. This hazard ratio can be estimated by exponentiating the coefficient for prison plus 2 times the coefficient for the CLINIC* PRISON interaction term. This expression is obtained by substituting the appropriate values into the hazard in both the numerator (for PRISON=1) and denominator (for PRISON=0) (see below):

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

The resulting hazard ratio, $\exp(\beta_1 + 2\beta_2)$, is an exponentiated linear combination of parameters. Unfortunately, R does not have a **lincom** command that Stata provides or an **estimate** statement that SAS provides in order to calculate a linear combination of parameter estimates. However an approach that can be used in any statistical software package for such a situation is to recode the variable(s) of interest such that the desired estimate is no longer a linear combination of parameter estimates.

In this example, we are interested in a hazard ratio PRISON=1 versus PRISON=0 for CLINIC=2. We can define a new variable CLINIC $\times$ 2 so when CLINIC=2, CLINIC $\times$ 2=0.

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

The first line of code defines a new variable CLINIC2. CLINIC2 is used in the stratified Cox model rather than CLINIC. We are interested in the hazard ratio for PRISON=1 vs PRISON=0 for CLINIC2=0. When CLINIC2=0, the product terms cancel and the hazard ratio reduces to $\exp(\beta_1)$.

The second line of code applies the **summary** function to the **coxph** function. The summary function applied in this way produces additional output including 95% confidence intervals for the hazard ratios. The output follows:

```
 n= 238

                     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

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
```

The estimate for $\exp(\beta_1)$ can be found in the second table, exp(coef) for prison = 0.9203. The lower and upper confidence limits are 0.4346 and 1.9603, respectively. If we did not recode the variable CLINIC the problem would have been more complicated in that we would have had to use variance–covariance matrix (which can be obtained with the **vcov** function) to calculate a 95% confidence interval for this hazard ratio.

5. ASSESSING THE PH ASSUMPTION WITH A STATISTICAL TEST

The **cox.zph** function is designed to perform a statistical test on the proportional hazards assumption. This statistical test is a test of correlation between the Schoenfeld residuals and survival time (or ranked survival time). A correlation of zero supports the proportional hazards assumption (the null hypothesis). First, we define the response variable **Y** with the **Surv** function and then the **coxph** function is used to run a Cox proportional hazards model with the variables PRISON, DOSE, and CLINIC:

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

The object called **mod1** is created from the **coxph** function. This object is the first argument for the **cox.zph** function. The code to run the test of the proportional hazards assumption follows:

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

The second argument requests that ranked survival times be tested against the Schoenfeld residuals rather than the actual survival times (the default). The output follows:

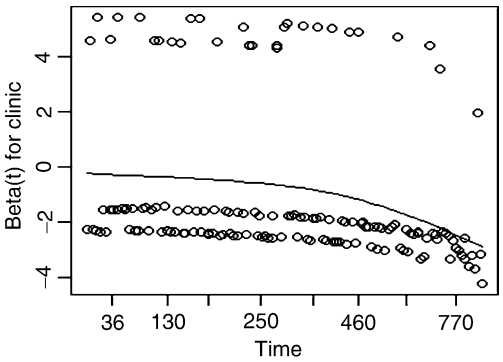
```
           rho  chisq       p
prison -0.0462  0.322 0.57068
dose    0.0905  1.096 0.29521
clinic -0.2498 10.495 0.00120
GLOBAL      NA 12.425 0.00606
```

The output shows that the correation between the Schoenfeld residuals for the variable CLINIC (3rd row) and ranked survival time is -0.2498 with a p-value of 0.00120. The significant p-value offers evidence that the proportional hazards assumption is not satisfied for the variable CLINIC. The p-values for PRISON and DOSE are not significant suggesting that there is not enough evidence to

reject the proportional hazards assumption for PRISON and DOSE. The global test (4th row) tests the proportional hazards assumption for the entire model (i.e., for all three predictor variables simultaneously) and is significant with p = 0.00606. The global test offers evidence that the proportional hazards assumption is violated for this model.

We can plot the Schoenfeld residuals against each individual's failure time with the **plot** function and an object created from the **cox.zph** function as the first argument. The argument **var=clinic**, specifies that the residuals should pertain the variable CLINIC. The argument **se=F**, suppresses the printing of confidence limits for the fitted curve. The code and output follow:

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



If the PH assumption is met then the fitted curve should look horizontal because the Schoenfeld residuals would be independent of survival time. However, the fitted curve slopes downward.

6. OBTAINING COX-ADJUSTED SURVIVAL CURVES

Cox adjusted survival estimates and plots can be obtained by applying the **summary** or **plot** function to an object created from the function **survfit**. The first step is to run the Cox model with the **coxph** function:

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

Adjusted survival curves generally depend on the pattern of covariates. Suppose we are interested in plotting the survival curve for the pattern PRISON=0, DOSE=70, and CLINIC=2. First, we need to create a dataset (or dataframe)

with the **data.frame** function with one observation. Code and output follows:

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

pattern1

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

This one observation dataframe is called **pattern1**. To obtain Cox adjusted survival estimates apply the **survfit** function within the **summary** function as shown below:

summary(survfit(mod1,newdata=pattern1))

The first argument of the **survfit** function is the object called **mod1** created with **coxph** function. The second argument supplies the dataframe containing the pattern of covariates of interest (called **pattern1**).

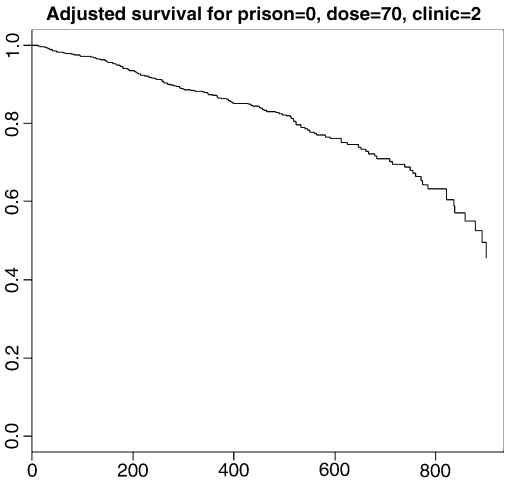
The output follows:

```
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
```

To obtain a Cox adjusted survival curve for the same pattern of covariates, apply the plot function in the same manner that the summary function was applied above. The code follows:

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

The **conf.int=F** option suppresses the plotting of the confidence limits. The option **conf.int=T** (the default) would plot the 95% confidence limits. The **main=** option requests a title for the plot. The output follows:



Stratified Cox adjusted survival curves can be obtained by first running a stratified Cox model (stratified by CLINIC):

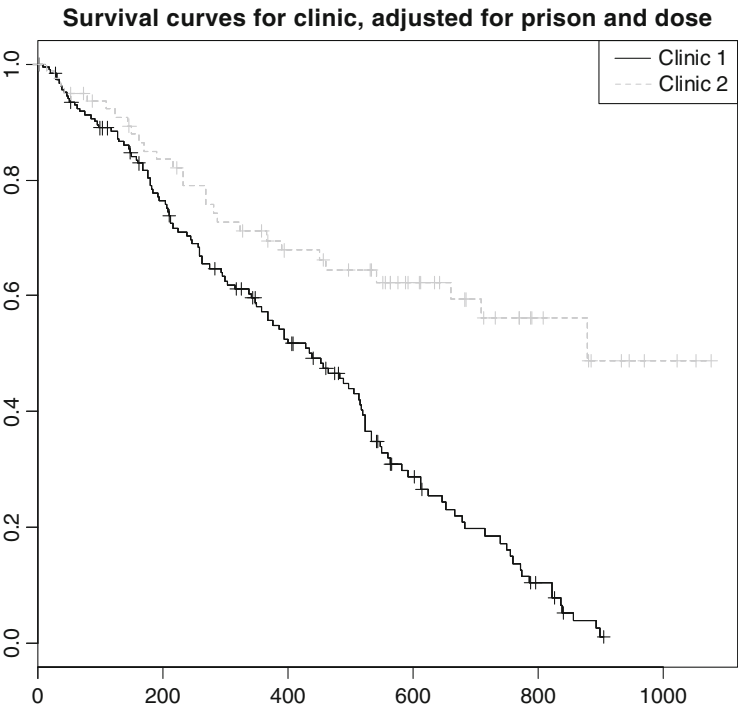
mod3=coxph(Y∼ prison + dose + strata(clinic),data=addicts)

To obtain stratified Cox adjusted curves controlling for PRISON and DOSE, we create a one observation dataframe with the mean values of 0.46 for PRISON and 60.4 for DOSE:

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

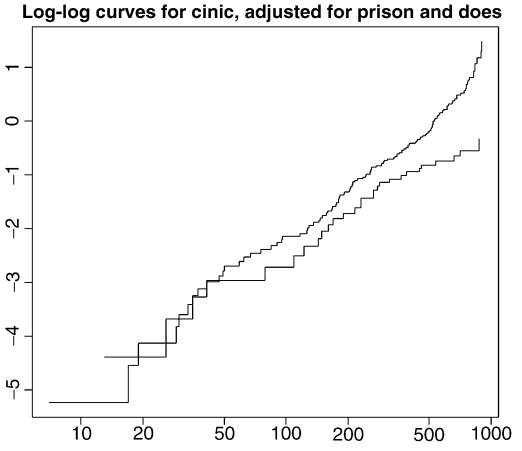
Now apply the **plot** function to the **survfit** function as shown in the last example. The code and output follow:

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”))



The **fun=** option in the plot function can be used to plot log-log survival curves. The code and output follow:

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



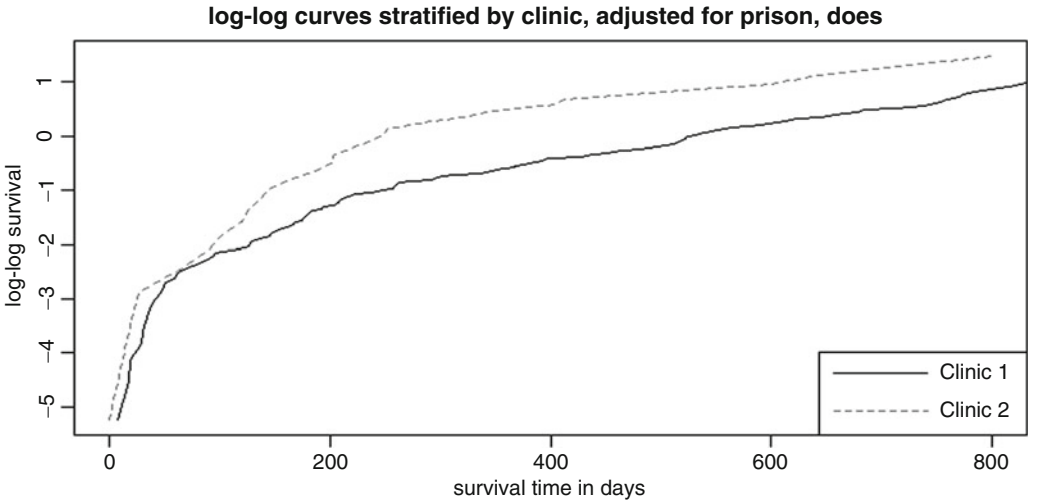
The **fun=** option plots time on a logarithmic scale. It is not so straightforward if you want the log-log plot against time with time not on a logarithmic scale. This was shown in Sect. 2 for KM log log curves. First, the adjusted survival

estimates can be saved in an object we'll call **sum.mod3** (shown below):

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

Now, if it desired to plot the log-log plot against time with time not on a logarithmic scale, similar code can be used as was shown in Section 2, except replace the object we had called **kmfit3** in Section 2 with the object created above, called **sum.mod3**. The code and plot follows:

```
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. RUNNING AN EXTENDED COX MODEL

In contrast to Stata, SAS, and SPSS, in order to run an extended Cox model in R, the analytic dataset must be in the counting process (start, stop) format. Unfortunately, the addicts dataset is not in that format, so it needs to be altered in order to include a time-varying covariate. This can be accomplished with the **survSplit** function. The **survSplit** function can create a dataset that provides multiple observations for the same subject allowing a subject's covariate to change values from observation to observation. The user supplies the time cutpoint(s).

The most general choice for time cutpoints that can accommodate the modeling of any time-varying covariate is a vector of time cutpoints that includes all event times in the data. The variable SURVT in the addicts dataset contains each individual's time-to-event or time-to-censorship. The following code creates a new analytic dataset (called **addicts.cp**) which puts the addicts data in the counting process format using the **survSplit** function:

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

The first argument of the **survSplit** function specifies the dataframe (addicts) to be manipulated into the counting process format. The **cut= addicts$survt[addicts$status==1]** option specified that the time cutpoints are indicated by the SURVT variable subsetted where the STATUS variable equals 1 (i.e., keeping the event times but omitting censorship times). The **event="status"** option specifies STATUS as the variable indicating whether the individual had an event or was censored. The **start="start"** option creates a new variable called START. This newly defined variable for the starting times for each observation is necessary for the data to be in counting process (start, stop) format. The **end="survt"** option defines SURVT as the stop variable (i.e., the time-to-event variable). The option **id="id"** indicates that ID is the variable that identifies each individual. The **survSplit** function creates multiple observations for individuals at risk at multiple time points. The dataset **addicts.cp** created above contains 18,708 observations from the 238 observations in the addicts dataset (use the **nrow** function and the code **nrow(addicts.cp)**) to return the number of observations.

Suppose the PH assumption was violated for the variable DOSE and we were interested in defining a time-varying covariate as the product of DOSE and the natural log of

time (SURVT). This variable can easily be defined if the dataset is in counting process form with time cutpoints at each event time as shown below:

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

We now have a new variable in the dataset (called LOGTDOSE=ln(DOSE)*T) that varies over time. We print the dataset for one individual (id=106) who had an event at time=35 days. Rather than print all the variables, we request a subset of them with the **c** function:

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

id	start	survt	status	dose	logtdose
106	0	7	0	40	77.8364
106	7	13	0	40	102.5980
106	13	17	0	40	113.3285
106	17	19	0	40	117.7776
106	19	26	0	40	130.3239
106	26	29	0	40	134.6918
106	29	30	0	40	136.0479
106	30	33	0	40	139.8603
106	33	35	1	40	142.2139

The variable LOGTDOSE is time dependent as its values increase with time as expected. The variable SURVT lists all the event times in the addicts dataset up to day 35 when this individual had an event. Notice STATUS=1 when the event occurred and STATUS=0 prior to the event. Next we run an extended Cox model including the predictors PRISON, DOSE, and CLINIC and the time-dependent variable LOGTDOSE:

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

The **Surv** function now takes three arguments: the start variable (called START), the stop variable (called SURVT), and the status variable (called STATUS). The term **cluster (ID)** in the model formula indicates that there are multiple observations (clusters) from the same subject and requests that robust standard errors be produced for the coefficient estimates. These robust standard errors are designed to account for the non-independence of observations from the same subject. The model output follows:

```
              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
```

The Wald test z statistic of 1.64 (p = 1.0e-01 or p=0.10) is not significant for LOGTDOSE, providing no evidence that the proportional hazards assumption is violated for DOSE.

Next we run an extended Cox model with heaviside functions for CLINIC defined about the time cutpoint of 365 days. We could use the dataset that we just created, **addicts.cp**, but since there is now only one cutpoint, we illustrate how to create a dataset in counting process format with only one cutpoint. The new dataset (called **addicts.cp365**) will have 360 observations compared to 18,708 in the dataset we previously had created called **addicts.cp**. The code follows:

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

The **cut=365** option in the **survSplit** function requests that day 365 be the only cutpoint. Next we create the two time-dependent variables (HV1 and HV2). HV1 is defined to equal the value of CLINIC if survival time is less than 365 days and 0 otherwise. HV2 is defined to equal 0 if survival time is less than 365 days and equal the value of CLINIC otherwise (code follows):

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

The conditional statements in the code **(addicts.cp365 $start<365)** and **(addicts.cp365$start>=365)**, take the values of 1 if true and 0 if false and are then multiplied by the variable CLINIC to define HV1 and HV2.

Next we'll sort the dataset by the variables ID and START. This is not a necessary step but it is easier to view and understand the data when multiple observations from the same subject are consecutive. The **order** function sorts the dataset:

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

Next we print the first 10 observations for selected variables:

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
```

Notice the sorted order of the ID variable is 1, 10, and 100 rather than 1, 2, and 3. The ID variable is a character rather than numeric variable and is sorted in "alphabetical" rather than numerical order. The first subject (ID=1) had an event at 428 days, so was censored (STATUS=0) during the first time interval (0, 365) but had an event (STATUS=1) during the second interval (365, 428). This subject has the value CLINIC=1, thus has the time-dependent values HV1=1 and HV2=0 over the first interval and HV1=0 and HV2=1 over the second interval.

Before running an extended Cox model with these heaviside functions we define an object (called **Y365**) for the response variable using the **Surv** function. This object is then used in the **coxph** model formula. It is not necessary to explicitly define this object and we did not do so for the previous extended Cox model that we ran containing LOGTDOSE, but the code is more readable with the notation for the response variable simplified. The code follows:

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

Next we run the model with two heaviside functions (code and output follow):

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
```

The estimated hazard ratio (CLINIC=2 vs. CLINIC=1) is 0.632 for days <365 and 0.160 for days ≧365 (found in the second numeric column under exp(coef)). If we wish to match the SAS, Stata, and SPSS output, we could run the model without robust standard errors and use the **method="breslow"** to handle simultaneous events (ties) in the Cox likelihood. The code follows (output omitted):

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

To run an equivalent model with one heaviside function, we need to include the CLINIC variable in the model (code and output shown below):

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
```

The coefficient estimates are different with this model compared to the model with two heaviside functions but the estimated hazard ratios are the same. The estimated hazard ratio (CLINIC=2 vs. CLINIC=1) is 0.632 for days <365 (exponentiate the coefficient for CLINIC). In order to estimate the hazard ratio for days ≥ 365, we need to sum the coefficient estimates for CLINIC and HV2 and then exponentiate (exp(-0.4594 + -1.3711)) = 0.160). The significant p-value for the estimated coefficient for HV2 of (p = 3.6e-10 or p = 0.0036) suggests that the hazard ratios for CLINIC for the two different time periods are not equal. In other words, the significant p-value provides evidence that the proportional hazard assumption is violated for CLINIC.

8. RUNNING PARAMETRIC MODELS

The **survreg** function in R runs parametric accelerated failure time (AFT) models. Whereas the key assumption of a proportional hazards (PH) model is that hazard ratios are constant over time, the key assumption for an AFT model is that survival time accelerates (or decelerates) by a constant factor when comparing different levels of covariates.

The most common distribution for parametric modeling of survival data is the Weibull distribution. The hazard function for a Weibull distribution is $\lambda p t^{p-1}$. If p = 1, then the Weibull distribution is also an exponential distribution. The Weibull distribution has the desirable property in that if the AFT assumption holds then the PH assumption also holds. The exponential distribution is a special case of the Weibull distribution. The key property for the exponential distribution is that the hazard is constant over time ((*h* (*t*) = λ). In R, the Weibull and exponential model are run only as AFT models.

The Weibull distribution has the property that the log-log of the survival function is linear with the log of time. Recall in Section 2 (assessing the PH assumption graphical approach) that the **fun="cloglog"** option in the **plot** function requested Kaplan-Meier log-log survival plot be plotted against time (on the log scale) for the variable CLINIC. The curves from this plot can be used to evaluate the Weibull assumption. If the survival curves are approximately straight lines (and parallel), then the Weibull assumption is reasonable for CLINIC. Furthermore, if the straight lines have a slope of 1, then the exponential distribution is appropriate. We repeat and condense the code that was given in Section 2 (see outputted plot in Section 2):

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

The log–log curves in Section 2 do not look straight but for illustration, we shall proceed as if the Weibull assumption were appropriate. First an exponential model is run with the **survreg** function. In this model, the Weibull shape parameter (p) is forced to equal 1, which forces the hazard to be constant. We'll save the results in an object called **modpar1**:

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

Next we apply the summary function to the object we just created (code and output shown below):

summary(modpar1)

```
               Value Std. Error     z        p
(Intercept)  3.6843    0.43072  8.55 1.19e-17
prison      -0.2526    0.16489 -1.53 1.25e-01
dose         0.0289    0.00614  4.71 2.52e-06
clinic       0.8806    0.21063  4.18 2.91e-05

Scale fixed at 1

Exponential distribution
Loglik(model)= -1094   Loglik(intercept only)= -1118.9
        Chisq= 49.91 on 3 degrees of freedom, p= 8.3e-11
Number of Newton-Raphson Iterations: 5
n= 238
```

The key assumption of an exponential model is that the hazard is constant over time. This is indicated in the output by the statement "Scale fixed at 1" listed under the tables of parameter estimates. The output can be used to estimate the hazard ratio for any subject given a pattern of covariates. Note that R outputs the parameter estimates for the AFT form of the exponential model. Multiply the estimated coefficients by one to get estimates consistent with the PH parameterization of the model (see Chapter. 7). For example, the estimated hazard ratio comparing PRISON=1 vs PRISON=0 is exp(0.2526) = 1.29. The corresponding acceleration factor for an exponential model is just the reciprocal of the hazard ratio, exp(-0.2526) = 0.78. Having a prison record accelerates the time to event by a factor of 0.78.

Next a Weibull AFT model is run with the **survreg** function. The results are saved in an object called **modpar2**:

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

Next we apply the summary function to the object **modpar2** (code and output follow):

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
n= 238
```

The Weibull shape parameter is the reciprocal of what R calls the Scale parameter (estimated at 0.73). An estimate for the Weibull shape parameter can be obtained by taking the reciprocal, 1/0.73 = 1.37. The acceleration factor comparing CLINIC=2 to CLINIC=1 is estimated at exp(0.7090) = 2.03. So, the estimated median survival time (time off heroin) is double for patients enrolled in CLINIC=2 compared to CLINIC=1.

We can use the model results and the **predict** function to estimate the median (or any other quantile) time to event for any specified pattern of covariates. For example, we can obtain the 25^{th}, 50^{th}, and 75^{th} percentile of survival time estimated from the Weibull model results that we saved in the object **modpar2** for an individual who has the covariate pattern PRISON=1, DOSE=50, and CLINIC=1. The code follows:

```
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)
```

The first statement in the code creates a dataframe of one observation specifying the pattern of covariates of interest. This dataframe (called **pattern1**) could have contained more than one observation if we were interested in comparing different patterns of covariates. The next statement creates a vector (called **pct**) which contains the percentiles of interest (25^{th}, 50^{th}, and 75^{th}). The third statement creates an object (called **days**) that contains output from the **predict** function. The first argument of the **predict** function is the object we called **modpar2** that contains the

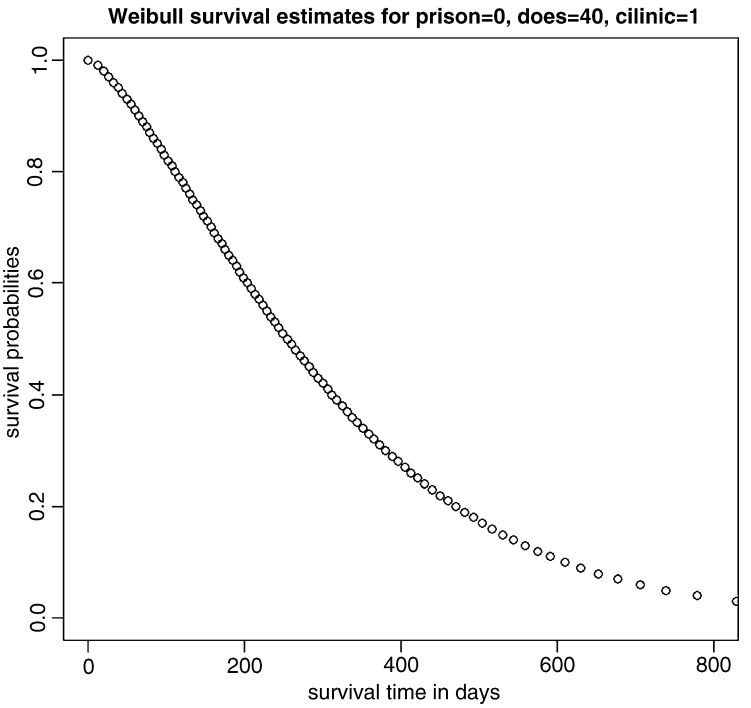
Weibull model results. The second argument, **newdata= pattern1**, inputs the pattern of covariates of interest. The third argument, **type="quantile"**, requests that quantiles be output. The fourth argument, **p=pct**, inputs the vector of quantiles that we created in the line of code above it. The last statement of code uses the **cbind** function to combine the vectors **pct** and **days** side-by-side in columns. The output follows:

```
pct     days
0.25   133.8074
0.50   254.2196
0.75   421.6070
```

The estimated median survival time is 254.2196 days. We can use similar code to plot the survival curve for an individual who has the covariate pattern PRISON=1, DOSE=50, and CLINIC=1 using the Weibull model results. The code follows:

```
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))
```

The first statement creates a vector called **pct2** that contains a sequence of percentiles between 0 and 1 incremented by 0.01,(0, 0.01, 0.02,...,0.99, 1). The second statement creates an object, called **days2**, containing output from the **predict** function. The third argument creates a vector called **survival** which reverses the order of **pct2**. Finally, the **plot** function plots the vectors **days2** on the horizontal axis and **survival** on the vertical axis. Axis labels and a title are added using **plot** function options. The output follows:



Next a log-logistic AFT model is run with the **survreg** function. The results are saved in an object called **modpar3**:

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

Next, we apply the summary function to the object **modpar3** (code and output shown below):

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
```

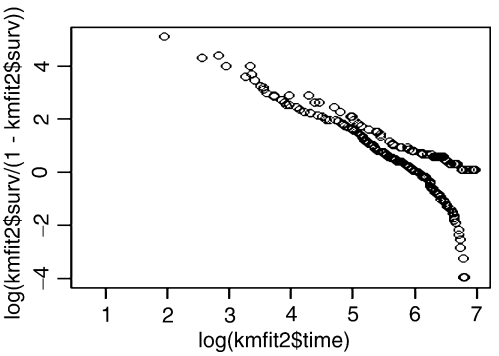
From this output, the acceleration factor comparing CLINIC=2 to CLINIC=1 is estimated as exp(0.5806) = 1.79. If the AFT assumption holds for a log logistic model, then the proportional odds assumption holds for the survival function (although the PH assumption will not hold). The proportional odds assumption can be evaluated by plotting the log odds of survival (using KM estimates) against the log of survival time. If the plots look like straight lines for each pattern of covariates then the log logistic distribution is reasonable. If the straight lines are also parallel then the proportional odds and AFT assumptions also hold.

In Section. 2, we created an object which we called **kmfit2** that contained the Kaplan-Meier survival estimates. We repeat the code to recreate that object:

```
kmfit2=survfit(Surv(addicts$survt,addicts$status)~addicts$clinic)
```

The vector **kmfit2$time** contains the survival times and the vector **kmfit2$surv** contains the KM survival estimates by CLINIC. The **plot** function can be used to plot log odds of survival, $\log[(S/(1 \times S)]$, against the log of survival time. The code and output follow:

```
plot(log(kmfit2$time),log(kmfit2$surv/(1-kmfit2$surv)))
```



The curves do not look like straight lines or parallel so the proportional odds assumption for CLINIC looks to be violated. We had run the log-logistic model earlier for illustration, even though the graph suggests that it is not the appropriate model.

Other distributions supported by the **survreg** function are the normal (dist="gaussian") and the lognormal (dist="lognormal") distributions.

9. RUNNING FRAILTY MODELS

Frailty models contain an extra random component designed to account for individual-level differences in the hazard otherwise unaccounted for by the model. The frailty, α, is a multiplicative effect on the hazard assumed to follow some distribution. The hazard function conditional on the frailty can be expressed as $h(t|\alpha) = \alpha[h(t)]$.

R offers three choices for the distribution of the frailty: the gamma, Gaussian, and t distributions. The variance (theta) of the frailty component is a parameter typically estimated by the model. If theta = 0, then there is no frailty.

First, we rerun a stratified Cox model without frailty (previously shown in Section. 4). The stratified variable is CLINIC while PRISON and DOSE are predictor variables. A stratified Cox model is appropriate if the PH assumption is violated for CLINIC and met for PRISON and DOSE and our interest is in estimating a hazard ratio for PRISON or DOSE. The code and output follow:

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
```

The estimated hazard ratio for PRISON=1 versus PRISON=0 is exp(0.3896) = 1.476. Next we illustrate how to include a frailty component in this model. The code follows:

coxph(Y~ prison + dose + strata(clinic) + frailty(id, distribution= "gamma"), data=addicts)

The term + **frailty(id, distribution="gamma")** is included in the model formula. The first argument of the **frailty** function is the variable **id** and indicates that the unmeasured heterogeneity (the frailty) is at the individual level. The second argument indicates that the distribution of the random component is the gamma distribution. The output follows:

```
                          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
```

Under the table of parameter estimates the output indicates that the variance of random effect = 0.00227. The *p*-value for the frailty component of 3.1e-01= 0.31 is provided in the third row and right column of the table and indicates that the frailty component is not significant. We conclude that the variance of the random component is zero for this model (i.e., there is no frailty). The parameter estimates for PRISON and DOSE changed minimally in this model compared to the model previously run without the frailty.

Now, suppose the variable CLINIC was unmeasured. Next we consider a Cox model (without frailty) that does not contain CLINIC. The code and output follow:

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
```

The estimated hazard ratio for PRISON=1 versus PRISON=0 is exp(0.1897) = 1.209 as compared to exp (0.3896) = 1.476 that was observed in the model that contained CLINIC as a stratified variable. In previous sections CLINIC was shown to be an important predictor that violates the proportional hazards assumption. If CLINIC was unaccounted for (as in the model above), there may be a source of unobserved heterogeneity that a frailty component might address. The next model omits CLINIC but includes a frailty component and the predictors PRISON and DOSE. The code and output follow:

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
```

The variance of the frailty component is estimated at 0.65 compared to 0.00227 for the model that we showed previously that contained CLINIC as the stratified variable. The *p*-value for the frailty is highly significant at 8.6e–3 = 0.0086. The hazard ratio for the effect of PRISON is exp (0.4144) = 1.51. The **summary** function can be applied to the **coxph** function to get R to exponentiate the parameter estimates (with 95% CI) when a frailty component is included in a Cox model. The code and output follow:

summary(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

       exp(coef) exp(-coef) lower .95 upper .95
prison      1.51      0.661     0.980     2.337
dose        0.95      1.053     0.934     0.966

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
Rsquare= 0.551   (max possible= 0.997 )
Likelihood ratio test= 190  on 70.7 df,   p=6.17e-13
Wald test            = 38.8  on 70.7 df,   p=1
```

It is interesting that the estimated hazard ratio for PRISON (1.51) obtained in this model (without CLINIC but with the frailty component) is closer to the corresponding hazard ratio obtained from the model that included CLINIC (1.476) compared to the one that did not include CLINIC (1.209). In this example, the frailty component might be accounting to some extent for the fact that CLINIC was omitted from the model.

10. MODELING RECURRENT EVENTS

The modeling of recurrent events is illustrated with the bladder cancer dataset (**bladder.rda**) described at the start of this appendix. Recurrent events are represented in the data with multiple observations for subjects having multiple events. The data layout for the bladder cancer dataset is in the counting process (start, stop) format with time intervals defined for each observation (see Chapter 8). The **load** function is used to access an R dataframe that has been saved as a file. Suppose the bladder dataset has been saved on your C drive as C:\crbladder.rda. The following code will load the bladder data:

```
load("C:\\bladder.rda")
```

The following code prints the 12^{th}–20^{th} observation, which contains information for four subjects:

```
bladder[12:20, ]
```

The output follows:

```
    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
```

There are three observations for ID=10, one observation for ID=11, three observations for ID=12, and two observations for ID=13. The variables START and STOP represent the time interval for the risk period specific to that observation. The variable EVENT indicates whether an event (coded 1) occurred. The first three observations indicate that the subject with ID=10 had an event at 12 months, another event at 16 months, and was censored at 18 months.

Recall we analyzed data in the counting process format when we ran extended Cox models (Section 7). In that section we saw how a subject's covariate can change values from time-interval to time-interval. With the bladder dataset, the (start,stop) data format provides a way to indicate that a subject experienced multiple events.

As mentioned in the beginning of our discussion of R, the code **library(survival)** must be submitted at each session before survival functions in R can be accessed.

```
library(survival)
```

The **coxph** function can be used to run Cox models with recurrent events. First, we'll define a response variable using the **Surv** function (called **Y**):

```
Y=Surv(bladder$start,bladder$stop,bladder$event==1)
```

As we have seen in Section 7, the **Surv** function requires three arguments with data in the counting process format: the start variable (called START), the stop variable (called STOP), and the status variable (called EVENT). The code **bladder$event==1** indicates that an event is coded 1. R recognizes the value 1 as the default coding of an event, so it was not necessary to state this explicitly in the **Surv** function as we did. Next, a recurrent-events Cox model is run with the predictors: treatment status (TX), initial number of tumors (NUM), and the initial size of tumors (SIZE):

```
coxph(Y ~ tx + num + size + cluster(id), data=bladder)
```

The term + **cluster(id)** in the model formula requests robust standard errors for the parameter estimates. The model output follows:

```
        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
```

The treatment variable (TX) is coded 1 for treatment with thiotepa and 0 for the placebo. The estimated hazard ratio (TX=1 vs. TX=0) is 0.663 (with a p-value of 0.0980). There are two sets of standard errors presented in the table under the columns labeled: se(coef) and robust se. The p-values and z-test statistics in this table are calculated using the robust standard errors. We could obtain additional model output (including 95% CIs) by applying the **summary** function to the **coxph** function.

A stratified Cox model can also be run using the data in this format with the variable INTERVAL as the stratified variable. The stratified variable indicates whether the subject was at risk for their first, second, third, or fourth event.

This approach is called a **Stratified CP** recurrent event model (see Chap. 8) and is used if the investigator wants to distinguish the order in which recurrent events occur. The bladder data is in the proper format to run this model. The code and output follow:

coxph(Y ~ tx + num + size + strata(interval) + cluster (id),data=bladder)

```
         coef exp(coef) se(coef) robust se      z    p
tx   -0.3335     0.716   0.2162    0.2048 -1.628 0.10
num   0.1196     1.127   0.0533    0.0514  2.328 0.02
size -0.0085     0.992   0.0728    0.0616 -0.138 0.89

Likelihood ratio test=6.51  on 3 df, p=0.0893  n=190
```

The only additional code from the previous model is the term + **strata(interval)** in the model formula which indicates that INTERVAL is the stratified variable. Interaction terms between the treatment variable (TX) and the stratified variable could be created to examine whether the effect of treatment differed for the 1st, 2nd, 3rd, or 4th event.

Another stratified approach (called Gap Time) is a slight variation of the Stratified CP approach. The difference is in the way the time intervals for the recurrent events are defined. There is no difference in the time intervals when subjects are at risk for their first event. However, with the Gap Time approach, the starting time at risk gets reset to zero for each subsequent event. To run a Gap Time model, we need to create two new (start, stop) variables in the bladder dataset, which we'll call START2 and STOP2:

bladder$start2=0
bladder$stop2=bladder$stop – bladder$start

The first of the two newly defined variables (START2) is always zero. The second (STOP2) is defined as the time between each event (STOP–START). To print a subset of these variables, we can use the **data.frame** function. The **attach** function allows variables in the bladder dataset to be listed without the **bladder$** prefix (code and output for printing the 12th–20th observation below).

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
```

Next we need to reset our response variable using the **Surv** function by changing our time intervals from (START, STOP) to (START2, STOP2):

Y2=Surv(bladder$start2,bladder$stop2,bladder$event)

Next we run a Gap Time model with the bladder data using similar code that was used for the Stratified CP model except we use **Y2** rather than **Y** as our response variable. The code and output follow:

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
```

The results using the Gap Time approach varies slightly from that obtained using the Stratified CP approach.

Test Answers

Chapter 1

True-False Questions:

1. T
2. T
3. T
4. F: Step function.
5. F: Ranges between 0 and 1.
6. T
7. T
8. T
9. T
10. F: Median survival time is longer for group 1 than for group 2.
11. F: Six weeks or greater.
12. F: The risk set at 7 weeks contains 15 persons.
13. F: Hazard ratio.
14. T
15. T
16. $h(t)$ gives the instantaneous potential per unit time for the event to occur given that the individual has survived up to time t;$k(t)$ is greater than or equal to 0; $h(t)$ has no upper bound.
17. Hazard functions
    - give insight about conditional failure rates;
    - help to identify specific model forms (e.g., exponential, Weibull);
    - are used to specify mathematical models for survival analysis.
18. Three goals of survival analysis are:
    - to estimate and interpret survivor and/or hazard functions;
    - to compare survivor and/or hazard functions;
    - to assess the relationship of explanatory variables to survival time.

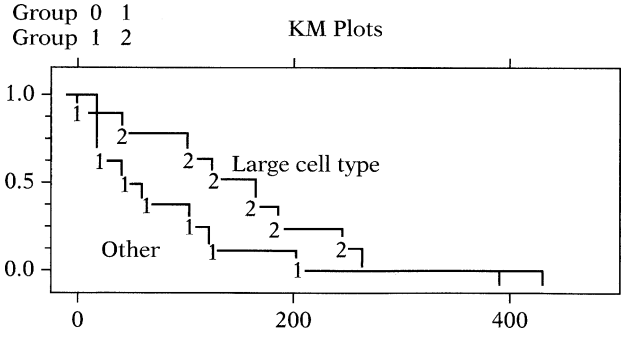
19.

	$t_{(j)}$	m_j	q_j	$R(t_{(j)})$
Group 1:	0	0	0	25 persons survive $\geq$ 0 years
	1.8	1	0	25 persons survive $\geq$ 1.8 years
	2.2	1	0	24 persons survive $\geq$ 2.2 years
	2.5	1	0	23 persons survive $\geq$ 2.5 years
	2.6	1	0	22 persons survive $\geq$ 2.6 years
	3.0	1	0	21 persons suivive $\geq$ 3.0 years
	3.5	1	0	20 persons survive $\geq$ 3.5 years
	3.8	1	0	19 persons survive $\geq$ 3.8 years
	5.3	1	0	18 persons survive $\geq$ 5.3 years
	5.4	1	0	17 persons survive $\geq$ 5.4 years
	5.7	1	0	16 persons survive $\geq$ 5.7 years
	6.6	1	0	15 persons survive $\geq$ 6.6 years
	8.2	1	0	14 persons survive $\geq$ 8.2 years
	8.7	1	0	13 persons survive $\geq$ 8.7 years
	9.2	2	0	12 persons survive $\geq$ 9.2 years
	9.8	1	0	10 persons survive $\geq$ 9.8 years
	10.0	1	0	9 persons survive $\geq$ 10.0 years
	10.2	1	0	8 persons survive $\geq$ 10.2 years
	10.7	1	0	7 persons survive $\geq$ 10.7 years
	11.0	1	0	6 persons survive $\geq$ 11.0 years
	11.1	1	0	5 persons survive $\geq$ 11.1 years
	11.7	1	3	4 persons survive $\geq$ 11.7 years

20. a. Group 1 has a better survival prognosis than group 2 because group 1 has a higher average survival time and a correspondingly lower average hazard rate than group 2.
    b. The average survival time and average hazard rates give overall descriptive statistics. The survivor curves allow one to make comparisons over time.

Chapter 2

1. a. KM plots and the log rank statistic for the cell type 1 variable in the vets.data dataset are shown below.

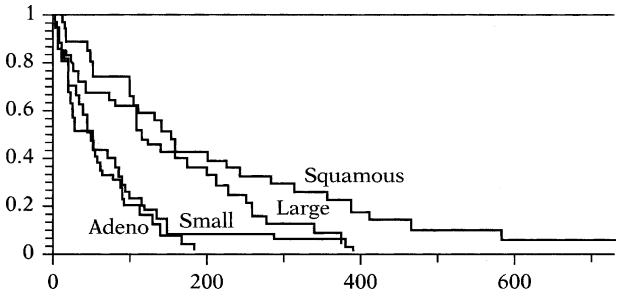


Group	Events observed	Events expected
1	102	93.45
2	26	34.55
Total	128	128.00

Log rank = chi2(1) = 3.02
p-value = Pr > chi2 = 0.0822

The KM curves indicate that persons with large cell type have a consistently better prognosis than persons with other cell types, although the two curves are essentially the same very early on and after 250 days. The log rank test is not significant at the .05 level, which gives somewhat equivocal findings.

b. KM plots and the log rank statistic for the four categories of cell type are shown below.



The KM curves suggest that persons with adeno or small cell types have a poorer survival prognosis than persons with large or squamous cell types. Moreover, there does not appear to be a meaningful difference between adeno or small cell types. Also, persons with squamous cell type seem to have, on the whole, a better prognosis than persons with large cell type.

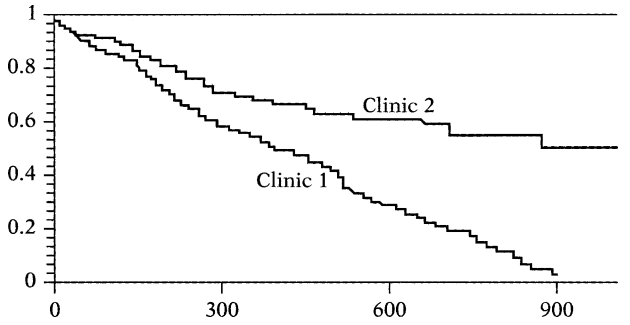
Computer results from Stata giving log rank statistics are now shown.

Group	Events observed	Events expected
1	26	34.55
2	26	15.69
3	45	30.10
4	31	47.65
Total	128	128.00

Log rank = chi2(3) = 25.40
P-value = Pr > chi2 = 0.0000

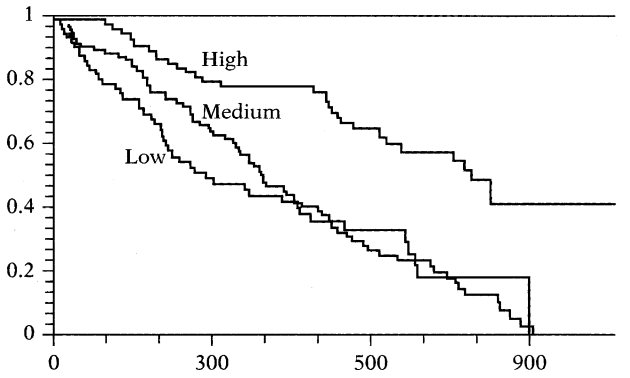
The log-rank test yields highly significant p-values, indicating that there is some overall difference between all four curves; that is, the null hypothesis that the four curves have a common survival curve is rejected,

2. a. KM plots for the two clinics are shown below. These plots indicate that patients in clinic 2 have a consistently better prognosis for remaining under treatment than do patients in clinic 1. Moreover, it appears that the difference between the two clinics is small before one year of follow-up but diverges after one year of follow up.



b. The log rank statistic (27.893) and Wilcoxon statistic (11.63) are both significant well below the .01 level, indicating that the survival curves for the two clinics are significantly different. The log rank statistic is nevertheless much larger than the Wilcoxon statistic, which makes sense because the log rank statistic emphasizes the later survival experience, where the two survival curves are far apart, whereas the Wilcoxon statistic emphasizes earlier survival experience, where the two survival curves are closer together.

c. If methadone dose is categorized into high (70+), medium (55–70) and low (<55), we obtain the KM curves shown below.



The KM curves indicate that persons with high doses have a consistently better survival prognosis (i.e.. maintenance) than persons with medium or low doses. The latter two groups are not very different from each other, although the medium dose group has a somewhat better prognosis up to the first 400 days of follow-up.
The log rank test statistic is shown below for the above categorization scheme.

Group	Events observed	Events expected
0	45	30.93
1	74	54.09
2	31	64.99
Total	150	150.00

Log rank = chi2(2) = 33.02
P-value = Pr>chi2 = 0.0000

The test statistic is highly significant, indicating that these three curves are not equivalent.

Chapter 3

1. a. $h(t,\mathbf{X}) = h_0(t)\exp[\beta_1 T1 + \beta_2 T_2 + \beta_3 PS + \beta_4 DC + \beta_5 BF + \beta_6(T1 \times PS) + \beta_7(T2 \times PS) + \beta_8(T1 \times DC) + \beta_9(T2 \times DC) + \beta_{10}(T1 \times BF) + \beta_{11}(T2 \times BF)]$

   b. **Intervention A: X*** = (1, 0, PS, DC, BF, PS, 0, DC, 0, BE 0)
   Intervention C: X = (− 1, −1, PS, DC, BF, –PS, –PS, –DC, –DC, –BF, –BF)

$$\mathrm{HR} = \frac{h(t,\mathbf{X}^*)}{h(t,\mathbf{X})} = \exp[2\beta_1 + \beta_2 + 2\beta_6 PS + \beta_7 PS + 2\beta_8 DC + \beta_9 DC + 2\beta_{10} BF + \beta_{11} BF]$$

c. H_0: $\beta_6 = \beta_7 = \beta_8 = \beta_9 = \beta_{10} = \beta_{11} = 0$ in the full model.
Likelihood ratio test statistic: $-2 \ln \hat{L}_R - (-2\mathrm{In}\hat{L}_F)$, which is approximately χ^2_6 under H_0, where R denotes the reduced model (containing no product terms) under H_0, and F denotes the full model (given in Part la above)

d. The two models being compared are:
Full model (F): $h(t,\mathbf{X}) = h_0(t)\exp[\beta_1 Tl + \beta_2 T2 + \beta_3 PS + \beta_4 DC + \beta_5 BF]$

Reduced model (R): $h(t,\mathbf{X}) = h_0(t)\exp[\beta_3 \text{PS} + \beta_4 \text{DC} + \beta_5 \text{BF}]$
H_0: $\beta_1 = \beta_2 = 0$ in the full model
Likelihood ratio test statistic: $-2\ln\hat{L}_R - (-2\ln\hat{L}_F)$, which is approximately χ^2_2 under H_0.

e.

Intervention A:
$\hat{S}(t,\mathbf{X}) = [\hat{S}_0(t)]^{\exp[\hat{\beta}_1 + (\overline{\text{PS}})\hat{\beta}_3 + (\overline{\text{DC}})\hat{\beta}_4 + (\overline{\text{BF}})\hat{\beta}_5]}$

Intervention B:
$\hat{S}(t,\mathbf{X}) = [\hat{S}_0(t)]^{\exp[\hat{\beta}_2 + (\overline{\text{PS}})\hat{\beta}_3 + (\overline{\text{DC}})\hat{\beta}_4 + (\overline{\text{BF}})\hat{\beta}_5]}$

Intervention C:
$\hat{S}(t,\mathbf{X}) = [\hat{S}_0(t)]^{\exp[-\hat{\beta}_1 - \hat{\beta}_2 + (\overline{\text{PS}})\hat{\beta}_3 + (\overline{\text{DC}})\hat{\beta}_4 + (\overline{\text{BF}})\hat{\beta}_5]}$

2. a. $h(t,\mathbf{X}) = h_0(t)\exp[\beta_1 \text{ CHR} + \beta_2 \text{ AGE} + \beta_3(\text{CHR} \times \text{AGE})]$

b. H_0: $\beta_3 = 0$
LR statistic = 264.90 − 264.70 = 0.21; χ^2 with 1 d.f. under H_0; not *significant*.
Wald statistic gives a chi-square value of .01, also not significant. Conclusions about interaction: the model should not contain an interaction term.

c. When AGE is controlled (using the gold standard model 2), the hazard ratio for the effect of CHR is exp(.8051) = 2.24, whereas when AGE is not controlled, the hazard ratio for the effect of CHR (using Model 1) is exp(.8595) = 2.36. Thus, the hazard ratios are not appreciably different, so AGE is not a confounder.
Regarding precision, the 95% confidence interval for the effect of CHR in the gold standard model (Model 2)is given by exp[.8051 ± 1.96(.3252)] = (1.183, 4.231) whereas the corresponding 95% confidence interval in the model without AGE (Model 1) is given by exp[.8595 ± 1.96(.3116)] = (1.282, 4.350). Both confidence intervals have about the same width, with the latter interval being slightly wider. Thus, controlling for AGE has little effect on the final point and interval estimates of interest.

d. If the hazard functions cross for the two levels of the CHR variable, this would mean that none of the models provided is appropriate, because each model assumes that the proportional hazards assumption is met for each predictor in the model. If hazard functions cross for CHR, however, the proportional hazards assumption cannot be satisfied for this variable.

e. for $CHR = 1 : \hat{S}(t, X) = \left[\hat{S}_0(t)\right]^{\exp\left[0.8051+0.0856(\overline{\text{AGE}})\right]}$
For $CHR = 0 : \hat{S}(t, X) = \left[\hat{S}_0(t)\right]^{\exp\left[0.0856(\overline{\text{AGE}})\right]}$

f. Using Model 1, which is the best model, there is evidence of a moderate effect of CHR on survival time, because the hazard ratio is about 2.4 with a 95% confidence interval between 1.3 and 4.4, and the Wald text for significance of this variable is significant below the .01 level.

a3. Full model (F = Model 1): $h(t,\mathbf{X})=h_0(t)\exp[\beta_1 Rx + \beta_2 \text{Sex} + \beta_3 \log \text{WBC} + \beta_4(Rx \times \text{Sex}) + \beta_5(Rx \times \log \text{WBC})]$
Reduced model (R = model 4):
$h(t,\mathbf{X}) = h_0(t)\exp[\beta_1 Rx + \beta_2 \text{Sex} + \beta_3 \text{logWBC}]$
H_0: $\beta_4 - \beta_5 = 0$
LR statistic = 144.218 − 139.030 = 5.19; χ^2 with 2 d.f. under H_0; not significant at 0.05, though signilicant at 0.10. The chunk test indicates some (though mild) evidence of interaction.

b. Using either a Wald test (p-value = .776) or a LR test, the product term $Rx \times$ log WBC is clearly not significant, and thus should be dropped from Model 1. Thus, Model 2 is preferred to Model 1.

c. Using Model 2, the hazard ratio for the effect of Rx is given by HR − $(h(t,\mathbf{X}^*))/(h(t,\mathbf{X})) = \exp[0.405 + 2.013\ \text{Sex}]$

d. Males (Sex = 0) : $\widehat{HR} = \exp[0.405] = 1.499$
Females (Sex = 1) : $\widehat{HR} = \exp[0.405 + 2.013(1)] = 11.223$

e. Model 2 is preferred to Model 3 if one decides that the coefficients for the variables Rx and $Rx \times$ Sex are meaningfully different for the two models. It appears that such corresponding coefficients (0.405 vs. 0.587 and 2.013 vs. 1.906) are different. The estimated hazard ratios tor Model 3 are 1.799 (males) and 12.098 (females), which are different, but not very different from the estimates computed in Part 3d for Model 2. If it is decided that there is a meaningful difference here, then we would conclude that log WBC is a confounder; otherwise log WBC is not a confounder. Note that the log WBC variable is significant in Model 2 ($P = .000$), but this addresses precision and not confounding. When in doubt, as in this case, the safest thing to do (for validity reasons) is to control for log WBC.

f. Model 2 appears to be best, because there is significant interaction of $Rx \times$ Sex ($P = .023$) and because log WBC is a likely confounder (from Part e).

Chapter 4

1. The $P(PH)$ values in the printout provide GOF statistics for each variable adjusted for the other variables in the model These $P(PH)$ values indicate that the clinic variable does not satisfy the PH assumption ($P << .01$), whereas the prison and dose variables satisfy the PH assumption ($P > .10$).
2. The log–log plots shown are parallel. However, the reason why they are parallel is because the clinic variable has been included in the model, and log–log curves for any variable in a PH model must always be parallel. If, instead, the clinic variable had been stratified (i.e., not included in the model), then the log–log plots comparing the two clinics adjusted for the prison and dose variables might not be parallel.
3. The log–log plots obtained when the clinic variable is stratified (i.e., using a stratified Cox PH model) are not parallel. They intersect early on in follow-up and diverge from each other later in follow-up. These plots therefore indicate that the PH assumption is not satisfied for the clinic variable.
4. Both graphs of log–log plots for the prison variable show curves that intersect and then diverge from each other and then intersect again. Thus, the plots on each graph appear to be quite nonparallel, indicating that the PH assumption is not satisfied for the prison variable. Note, however, that on each graph, the plots are quite close to each other, so that one might conclude that, allowing for random variation, the two plots are essentially coincident; with this latter point of view, one would conclude that the PH assumption is satisfied for the prison variable.
5. The conclusion of nonparallel log–log plots in Question 4 gives a different result about the PH assumption for the prison variable than determined from the GOF tests provided in Question 1. That is, the log–log plots suggest that the prison variable does not satisfy the PH assumption, whereas the GOF test suggests that the prison variable satisfies the assumption. Note, however, if the point of view is taken that the two plots are close enough to suggest coincidence, the graphical conclusion would be the same as the GOF conclusion. Although the final decision is somewhat equivocal here, we prefer to conclude that the PH assumption is satisfied for the prison variable because this is strongly indicated from the GOF test and questionably counterindicated by the log–log curves.

6. Because maximum methadone dose is a continuous variable, we must categorize this variable into two or more groups in order to graphically evaluate whether it satisfies the PH assumption. Assume that we have categorized this variable into two groups, say, low versus high. Then, **observed** survival plots can be obtained as KM curves for low and high groups separately To obtain **expected** plots, we can fit a Cox model containing the dose variable and then substitute suitably chosen values for dose into the formula for the estimated survival curve. Typically, the values substituted would be either the mean or median (maximum) dose in each group.
   After obtaining observed and expected plots for low and high dose groups, we would conclude that the PH assumption is satisfied if corresponding observed and expected plots art; not widely discrepant from each other. If a noticeable discrepancy is found for at least one pair of observed versus expected plots, we conclude that the PH assumption is not satisfied.
7. $h(t,\mathbf{X}) = h_0(t)\exp[\beta_1 \text{ clinic} + \beta_2 \text{ prison} + \beta_3 \text{ dose} + \delta_1 (\text{clinic} \times g(t)) + \delta_2 (\text{prison} \times g(t)) + \delta_3 (\text{dose} \times g(t))]$

   where $g(t)$ is some function of time. The null hypothesis is given by H_0: $\delta_1 = \delta_2 = \delta_3 = 0$. The test statistic is a likelihood ratio statistic of the form $LR = -2\ln L_R - (-2\text{In}L_F)$ where R denotes the reduced (PH) model obtained when all δs are 0, and F denotes the full model given above. Under H_0, the LR statistic is approximately chi-square with 3 d.f.
8. Drawbacks of the extended Cox model approach:

   - Not always clear how to specify $g(t)$; different choices may give different conclusions;
   - Different modeling strategies to choose from, for example, might consider $g(t)$ to be a polynomial in t and do a backward elimination to eliminate nonsignificant higher-order terms; alternatively, might consider $g(t)$ to be linear in t without evaluating higher-order terms.
     Different strategies may yield different conclusions.

9. $h(t,\mathbf{X}) = h_0(t)\exp[\beta_1 \text{ clinic} + \beta_2\text{prison} + \beta_3 \text{ dose} + \delta_1(\text{clinic} \times g(t))]$ where $g(t)$ is some function of time. The null hypothesis is given by H_0: $\delta_1 = 0$, and the test statistic is either a Wald statistic or a likelihood ratio statistic. The LR statistic would be of the form $LR = -2 \text{ In } L_R - (-2\text{In } L_F)$, where R denotes the reduced (PH) model obtained when $\delta_1 = 0$, and F denotes the full model given above. Either statistic is approximately chi-square with 1 d.f. under the null hypothesis.
10. $t > 365$ days: $HR = \exp[\beta_1 + \delta_1]$
    $t \leq 365$ days: $HR = \exp[\beta_1]$
    If δ_1 is not equal to zero, then the model does not satisfy the PH assumption for the clinic variable. Thus, a test of H_0: $\delta_1 = 0$ evaluates the PH assumption; a significant result would indicate that the PH assumption is violated. Note that if δ_1 is not equal to zero, then the model assumes that the hazard ratio is not constant over time by giving a different hazard ratio value depending on whether t is greater than 365 days or t is less than or equal to 365 days.

Chapter 5

1. By fitting a stratified Cox (SC) model that stratifies on clinic, we can compare adjusted survival curves for each clinic, adjusted for the prison and dose variables. This will allow us to visually describe the extent of clinic differences on survival over time. However, a drawback to stratifying on clinic is that it will not be possible to obtain an estimate of the hazard ratio for the effect of clinic, because clinic will not be included in the model.
2. The adjusted survival surves indicate that clinic 2 has a better survival prognosis than clinic 1 consistently over time. Moreover, it seems that the difference between the effects of clinic 2 and clinic 1 increases over lime.
3. $h_g(t, \mathbf{X}) = h_{0_g}(t)\exp[\beta_1 \text{ prison} + \beta_2 \text{ dose}], \; g = 1, 2$
   This is a no-interaction model because the regression coefficients for prison and dose are the same for each stratum.
4. Effect of prison, adjusted for clinic and dose: $\widehat{HR} = 1.475$, 95% CI: (1.059, 2.054). It appears that having a prison record gives a 1.475 increased hazard for failure than not having a prison record. The p-value is 0.021, which is significant at the 0.05 level.
5. Version 1: $h_g(t,\mathbf{X}) = h_{0_g}(t)\exp[\beta_{1g} \text{ prison} + \beta_{2g} \text{ dose}], \; g = 1, 2$
   Version 2: $h_g(t,\mathbf{X}) = h_{0_g}(t)\exp[\beta_1 \text{ prison} + \beta_2 \text{ dose} + \beta_3 (\text{clinic} \times \text{prison}) + \beta_4 (\text{clinic} \times \text{dose})], \; g = 1, 2$

6. $g = 1$ (clinic 1):
   $h_1(t,\mathbf{X}) = h_{01}(t)\exp[(0.502)\text{prison} + (-0.036)\text{ dose}]$
   $g = 2$ (clinic 2):
   $h_2(t,\mathbf{X}) = h_{02}(t)\exp[(-0.083)\text{prison} + (-0.037)\text{dose}]$
7. The adjusted survival curves stratified by clinic are virtually identical for the no-interaction and interaction models. Consequently, both graphs (no-interaction versus interaction) indicate the same conclusion that clinic 2 has consistently larger survival (i.e., retention) probabilities than clinic 1 as time increases.
8. H_0: $\beta_3 = \beta_4 = 0$ in the version 2 model (i.e., the no-interaction assumption is satisfied). $LR = -2\text{In } L_R - (-2 \text{ In } L_F)$ where R denotes the reduced (no-interaction) model and F denotes the full (interaction) model. Under the null hypothesis, LR is approximately a chi square with 2 degrees of freedom.
   Computed $LR = 1195.428 - 1193.558 = 1.87$; p-value = 0.395; thus, the null hypothesis is not rejected and we conclude that the no interaction model is preferable to the interaction model.

Chapter 6

1. For the chemo data, the –log-log KM curves intersect at around 600 days; thus the curves are not parallel, and this suggests that the treatment variable does not satisfy the PH assumption.
2. The P (PH) value for the Γx variable is 0, indicating that the PH assumption is not satisfied for the treatment variable based on this goodness-of-fit test.
3. $h(t,\mathbf{X}) = h_0(t)\exp[\beta_1(T\,x)g_1(t) + \beta_2(T\,x)g_2(t) + \beta_3(T\,x)g_3(t)]$
   where
   $$g_1(t) = \begin{cases} 1 & \text{if } 0 \le t < 250 \text{ days} \\ 0 & \text{if otherwise} \end{cases}$$
   $$g_2(t) = \begin{cases} 1 & \text{if } 25\,0 \le t < 500 \text{ days} \\ 0 & \text{if otherwise} \end{cases}$$
   $$g_3(t) = \begin{cases} 1 & \text{if } t \ge 500 \text{ days} \\ 0 & \text{if otherwise} \end{cases}$$
4. Based on the printout the hazard ratio estimates and corresponding p-values and 95% confidence intervals are given as follows for each time interval:

	Haz. Ratio	p> |z|	[95% Conf. Interval]	
$0 \leq t < 250$ days:	0.221	0.001	0.089	0.545
$250 \leq t < 500$ days:	1.629	0.278	0.675	3.934
$t \geq 500$ days:	1.441	0.411	0.604	3.440

The results show a significant effect of treatment below 250 days and a nonsignificant effect of treatment in each of the two intervals after 250 days. Because the coding for treatment was 1 = chemotherapy plus radiation versus 2 = chemotherapy alone, the results indicate that the hazard for chemotherapy plus radiation is 1/0.221 = 4.52 times the hazard for chemotherapy alone. The hazard ratio inverts to a value less than 1 (in favor of chemotherapy plus radiation after 250 days), but this result is nonsignificant. Note that for the significant effect of 1/0.221 = 4.52 below 250 days, the 95% confidence interval ranges between 1/0.545= 1.83 and 1/0.089= 11.24 when inverted, which is a very wide interval.

5. Model with two Heaviside functions:
$h(t,\mathbf{X}) = h_0(t)\exp[\beta_1(Tx)g_1(t) + \beta_2(Tx)g_2(t)]$
where

$$g_1(t) = \begin{cases} 1 & \text{if } 0 \leq t < 250 \text{ days} \\ 0 & \text{if otherwise} \end{cases}$$

$$g_2(t) = \begin{cases} 1 & \text{if } t \geq 250 \text{ days} \\ 0 & \text{if otherwise} \end{cases}$$

Model with one Heaviside function:
$h(t,\mathbf{X}) = h_0(t)\exp[\beta_1(Tx) + \beta_2(Tx)g_1(t)]$
where $g_1(t)$ is defined above.

6. The results for two time inteivals give hazard ratios that are on the opposite side of the null value (i.e., 1). Below 250 days, the use of chemotherapy plus radiation is, as in the previous analysis, 4.52 times the hazard when chemother apy is used alone. This result is significant and the same confidence interval is obtained as before. Above 250 days, the use of chemotherapy alone has 1.532 times the hazard of chemotherapy plus radiation, but this result is nonsignificant.

Chapter 7

1. F: They are multiplicative models, although additive on the log scale.
2. T
3. T
4. F: If the AFT assumption holds in a log-logistic model, the proportional odds assumption holds.
5. F: An acceleration factor greater than one suggests the exposure is beneficial to survival.
6. T
7. T
8. T
9. F: ln(T) follows an extreme value minimum distribution.
10. F: The subject is right-censored.
11. $$\gamma = \frac{\exp[\alpha_0 + \alpha_1(2) + \alpha_2 PRISON + \alpha_3 DOSE + \alpha_4 PRISDOSE]}{\exp[\alpha_0 + \alpha_1(1) + \alpha_2 PRISON + \alpha_3 DOSE + \alpha_4 PRISDOSE]}$$
$$= \exp(\alpha_1)$$
$$\hat{\gamma} = \exp(0.698) = 2.01$$
$$95\%\ \text{CI} = \exp[0.698 \pm 1.96(0.158)] = (1.47, 2.74)$$

    The point estimate for the acceleration factor (2.01) suggests that the survival time (time off heroin) is double for those enrolled in CLINIC = 2 compared to CLINIC = 1. The 95% confidence interval does not include the null value of 1.0 indicating a statistically significant preventive effect for CLINIC = 2 compared to CLINIC = 1.
12. $$\text{HR} = \frac{\exp[\beta_0 + \beta_1(2) + \beta_2 PRISON + \beta_3 DOSE + \beta_4 PRISDOSE]}{\exp[\beta_0 + \beta_1(1) + \beta_2 PRISON + \beta_3 DOSE + \beta_4 PRISDOSE]}$$
$$= \exp(\beta_1)$$
$$\hat{HR} = \exp(-0.957) = 0.38$$
$$95\%\ \text{CI} = \exp[-0.957 \pm 1.96(0.213)] = (0.25, 0.58)$$

    The point estimate of 0.38 suggests the hazard of going back on heroin is reduced by a factor of 0.38 for those enrolled in CLINIC = 2 compared to CLINIC = 1. Or from the other perspective: the estimated hazard is elevated for those in CLINIC = 1 by a factor of exp (+0 957) = 2.60.
13. $\beta_1 = -\alpha_1 p$ for CLINIC, so $\hat{\beta}_1 = -(0.698\text{X}1.370467) = -0.957$, which matches the output for the PH form of the model.

14. The product term PRISDOSE is included in the model as a potential confounder of the effect of CLINIC on survival. It is not an effect modifier because under this model the hazard ratio or acceleration factor for CLINIC does not depend on the value of PRISDOSE. The PRISDOSE term would cancel in the estimation of the hazard ratio or acceleration factor (see Questions 11 and 12). On the other hand, a product term involving CLINIC would be a potential effect modifier.
15. Using the AFT form of the model:

$$\frac{1}{\lambda^{1/p}} = \exp[\ \alpha_0 + \ \alpha_1\,\text{CLINIC} + \ \alpha_2\,\text{PRISON} + \ \alpha_3\,\text{DOSE} + \alpha_4\,\text{PRISDOSE}]$$

Median survival time for CLINIC = 2, PRISON = 1, DOSE = 50, PRISDOSE = 100:

$$t = [-\ln\ S(t)]^{1/p} \times \frac{1}{\lambda^{1/p}} = [-\ln(0.5)]^{1/p} \times \exp[\beta_0 + 2\beta_1 + \beta_2 + 50\beta_3 + 100\beta_4]$$

$\hat{t}$ (median) = 403.66 days (obtained by substituting parameter estimates from output).

16. Using the same approach as the previous question: Median survival time for CLINIC = 1, PRISON = 1, DOSE = 50, PRISDOSE = 100:
$t = [-\ln(0.5)]^{1/p} \times \exp[\beta_0 + l\beta_1 + \beta_2 + 50\beta_3 + 100\delta_4]$
$\hat{t}$ (median) = 200.85 days.
17. The ratio of the median survival times is 403.66/200.85 = 2.01. This is the estimated acceleration factor for CLINIC = 2 vs. CLINIC = 1 calculated in Question 11. Note that if we used any survival probability (i.e., any quantile of survival time), not just S(ι) -= 0.5 (the median), we would have obtained the same ratio.
18. The addition of the frailty component did not change any of the other parameter estimates nor did it change the log likelihood of −260.74854.
19. If the variance of the frailty is zero (theta = 0), then the frailty has no effect on the model. A variance of zero means the frailty (α) is constant at 1. Frailty is defined as a multiplicative random effect on the hazard $h(t|\alpha) = \alpha h(t)$. If $\alpha = 1$ then $h(t|\alpha) = h(t)$, and there is no frailty.

Chapter 8

1. a. **Survival time (say, in weeks) to the first event (stratum 1)**:

$t_{(f)}$	n_f	m_f	q_f	$R(t_{(f)})$
0	2	0	0	{B,L}
12	2	1	0	{B,L}
20	1	1	0	{L}

b. For each approach, the observation for the first event is identical.

c. **Survival rime (say, in weeks) from the first to the second event (stratum 2) using the Stratified CP approach**:

$t_{(f)}$	n_f	m_f	q_f	$R(t_{(f)})$
0	0	0	0	–
16	1	1	0	{B}
23	1	1	0	{L}

d. **Survival time (say, in weeks) from the first to the second event (stratum 2) using the Gap Time approach**:

$t_{(f)}$	n_f	m_f	q_f	$R_{(f)})$
0	2	0	0	{B,L}
3	2	1	0	{B,L}
4	1	1	0	{B}

e. **Survival time (say, in weeks) from the first to the second event using the Marginal approach**:

$t_{(f)}$	n_f	m_f	q_f	$R(t_{(f)})$
0	2	0	0	{B,L}
16	2	1	0	{B.L}
23	1	1	0	{L}

f. Correct choice is iii.
Bonnie is at risk for a second event between times 12 to 16.
Lonnie is at risk for a second event between times 20 to 23.
Neither is in the risk set for the other's second event.

g. Correct choice is ii.
Bonnie is at risk for a second event between times 0 to 4.
Lonnie is at risk for a second event between times 0 to 3.
Bonnie is in the risk set when Lonnie gets her second event.

h. Correct choice is i.
Bonnie is at risk for a second event between times 0 to 16.
Lonnie is at risk for a second event between times 0 to 23.
Lonnie is in the risk set when Bonnie gets her second event.

2. a. Cox PH Model for **CP** approach to Defibrillator Study:

$$h(t,\mathbf{X}) = h_0(t)\mathbf{exp}[\beta\ \mathbf{tx} + \gamma\ \mathbf{smoking}]$$

where **tx** = 1 if treatment A, 0 if treatment B. **smoking status** = 1 if ever smoked, 0 if never smoked.

b. Using the **CP** approach, there is no significant effect of treatment status adjusted for smoking. The estimated hazard ratio for the effect of treatment is 1.09, the corresponding P-value is 0.42 and a 95% CI for the hazard ratio is (0.88, 1.33).

c. No-interaction SC model for **Marginal** approach:

$$h_g[t,\mathbf{X}) = h_{0g}(t)\mathbf{exp}[\beta\ \mathbf{tx} + \gamma\ \mathbf{smoking}],\ g = 1, 2,3$$

Interaction SC model for **Marginal** approach:

$$h_g[t,\mathbf{X}) = h_{0g}(t)\mathbf{exp}[\beta_g\ \mathbf{tx} + \gamma_g\ \mathbf{smoking}],\ g = 1, 2, 3$$

d. $\mathbf{LR} = -2\ln L_R - (-21n\ L_F)$ is approximately χ^2 with 4 df under
$\mathbf{H_0}$:no-interaction SC model is appropriate, where R denotes the reduced (no interaction SC) model and **F** denotes the full (interaction SC) model

e. The use of a no-interaction model does not allow you to obtain stratum-specific HR estimates, even though you are assuming that strata are important.

f. The **CP** approach makes sense for these data because recurrent defibrillator (shock) events on the same subject are the same kind of event no matter when it occurred.

g. You might use the **Marginal** approach if you determined that different recurrent events on the same subject were different because they were of different order.

h. The number in the risk set (n_f) remains unchanged through day 68 because every subject who failed by this time was still at risk for a later event.

i. Subjects 3,6, 10,26, and 31 all fail for the third time at day 98 and are not followed afterwards.

j. Subjects 9, 15, and 28 fail for the second time at 79 days, whereas subject #16 is censored at 79 days.

k. Subjects 4, 14, 15, 24, and 29 were censored between days 111 and 112.

l. Subject #5 gets his first event at 45 days and his second event at 68 days, after which he drops out of the study. This subject is the first of the 36 subjects to drop out of the study, so the number in the risk set changes from 36 to 35 after 68 days.

m. None of the above.

n. The product limit formula is not applicable to the **CP** data; in particular, $P(T > t|T \geq t)$ does not equal "# failing in time interval /# in the risk set at start of interval."

o. Use the information provided in Table T.2 to complete the data layouts for plotting the following survival curves.

i. $S_1(t) = Pr(T_1 > t)$ where T_1 = time to first event from study entry

$t_{(f)}$	n_f	m_f	q_f	$S(t_p) = S(t_{f-1}) \times Pr(T_1 > t \mid T_1 \geq t)$
0	36	0	0	1.00
33	36	2	0	0.94
34	34	3	0	0.86
36	31	3	0	0.78
37	28	2	0	0.72
38	26	4	0	0.61
39	22	5	0	0.47
40	17	1	0	0.44
41	16	1	0	0.42
43	15	1	0	0.39
44	14	1	0	0.36
45	13	2	0	0.31
46	11	2	0	0.25
48	9	1	0	0.22
49	8	1	0	0.19
51	**7**	**2**	**0**	**0.19 × 5/7 = 0.14**
57	**5**	**2**	**0**	**0.14 × 3/5 = 0.08**
58	**3**	**2**	**0**	**0.08 × 1/3 = 0.03**
61	**1**	**1**	**0**	**0.03 × 0/1 = 0.00**

ii. **Gap Time** $S_{2c}(t) = Pr(T_{2c} > t)$ where T_{2c} = time to second event from first event.

$t_{(f)}$	n_f	m_f	q_f	$S_2(t_{(f)}) = S_2(t_{(f-1)}) \times Pr(T_2 > t \mid T_2 \geq t)$
0	36	0	0	1.00
5	36	1	0	0.97
9	35	1	0	0.94
18	34	2	0	0.89
20	32	1	0	0.86
21	31	2	1	0.81
23	28	1	0	0.78
24	27	1	0	0.75

(*Continued on next page*)

(*Continued*)

$t_{(f)}$	n_f	m_f	q_f	$S_2(t_{(f)}) = S_2(t_{(f-1)}) \times \Pr(T_2 > t \mid T_2 \geq t)$
25	26	1	0	0.72
26	25	2	0	0.66
27	23	2	0	0.60
28	21	1	0	0.58
29	20	1	0	0.55
30	19	1	0	0.52
31	18	3	0	0.43
32	15	1	0	0.40
33	14	5	0	0 26
35	9	1	0	0.23
39	8	2	0	0.17
40	**6**	**2**	**0**	**0.17 × 4/6 = 0.12**
41	**4**	**1**	**0**	**0.12 × 3/4 = 0.09**
42	**3**	**1**	**0**	**0.09 × 2/3 = 0.06**
46	**2**	**1**	**0**	**0.06 × 1/2 = 0.03**
47	**1**	**1**	**0**	**0.03 × 0/1 = 0.00**

iii. **Marginal** $S_{2m}(t) = \Pr(T_{2m} > t)$ where T_{2m} = time to second event from study entry.

$t_{(f)}$	n_f	m_f	q_f	$S(t_{(f)}) = S_2(t_{(f-1)}) \times \Pr(T_2 > t \mid T_2 \geq t)$
0	36	0	0	1.00
63	36	2	0	0.94
64	34	3	0	0.86
65	31	2	0	0.81
66	29	3	0	0.72
67	26	4	0	0.61
68	22	2	0	0.56
69	20	1	0	0.53
70	19	1	0	0.50
71	18	1	0	0.47
72	17	2	0	0.42
73	15	1	0	0.39
74	14	1	0	0.36
76	13	1	0	0.33
77	12	1	0	0.31
78	11	2	0	0.25
79	**9**	**3**	**1**	**0.25 × 6/9 = 0.17**
80	**5**	**2**	**0**	**0.17 × 3/5 = 0.10**
81	**3**	**2**	**0**	**0.10 × 1/3 = 0.03**
97	**1**	**1**	**0**	**0.03 × 0/1 = 0.00**

p. The survival curves corresponding to the above data layouts will differ because they are describing different survival functions. In particular, the composition of the risk set differs in all three data layouts and the ordered survival times being plotted are different as well.

Chapter 9

1. Cause-specific no interaction model for local recurrence of bladder cancer (event = 1):

$$h_1(t,\mathbf{X}) = h_{01}(t)\exp[\beta_{11}tx + \beta_{21}num + \beta_{31}size]$$

2. Censored subjects have bladder metastasis (**event** = 2) or other metastasis (**event** = 3).
3. Cause-specific no-interaction model for bladder metastasis (**event** =2):

$$h_2(t,\mathbf{X}) = h_{02}(t)\exp[\beta_{12}tx + \beta_{22} + \beta_{32}size]$$

where censored subjects have local recurrence of bladder cancer (**event** = 1) or other metastasis (**event** = 3).

4. A sensitivity analysis would consider worst-case violations of the independence assumption. For example, subjects censored from failing from events = 2 or 3 might be treated in the analysis as either all being event-free (i.e., change event status to 0 and time to 53) or all experiencing the event of interest (i.e., change event status to 1 and leave time as is).
5. a. Verify the CIC_1 calculation provided at failure time t_f=8 for persons in the treatment group (**tx** = 1):

$$\hat{\mathbf{h}}_1(8) = 1/23 = 0.0435$$
$$\hat{\mathbf{S}}(4) = \hat{\mathbf{S}}(3)\mathbf{Pr}(T > 4|T \geq 4) = 0.9630(1 - 2/26)$$
$$= 0.9630(0.9231) = 0.8889$$
$$\hat{\mathbf{i}}_1(8) = \hat{\mathbf{h}}_1(8)\hat{\mathbf{S}}(4) = 0.0435(.8889) = 0.0387$$
$$\mathbf{CIC}_1(8) = \mathbf{CIC}_1(4) + 0.0387 = 0 + 0.0387 = 0.0387$$

   b. Verify the $\mathbf{CIC}_1$ calculation provided at failure time t_f= 25 for persons in the placebo group (**tx** = 0):

$$\hat{\mathbf{h}}_1(25) = 1/6 = 0.1667$$
$$\hat{\mathbf{S}}(23) = \hat{\mathbf{S}}(21)\mathbf{Pr}(T > 23|T \geq 23) = 0.4150(1 - 1/8)$$
$$= 0.4150(0.875) = 0.3631$$
$$\hat{\mathbf{I}}_1(25) = \hat{\mathbf{h}}_1(25)\hat{\mathbf{S}}(23) = 0.1667(.3631) = 0.0605$$
$$\mathbf{CIC}_1(25) = \mathbf{CIC}_1(23) + 0.0605 = 0.2949 + 0.0605$$
$$= 0.3554$$

   c. interpret the $\mathbf{CIC}_1$ values obtained for both the treat ment and placebo groups at $t_f = 30$.
   For tx = 1, $\mathbf{CIC}_1(t_f = 30) = 0.3087$ and for tx = 0, $\mathbf{CIC}_1(t_f = 30) = 0.3554$.

Thus, for treated subjects (tx = 1), the cumulative risk (i.e., marginal probability) for local bladder cancer recurrence is about 30.1 % at 30 months when allowing for the presence of competing risks for bladder metastasis or other metastasis.

For placebo subjects (tx = 1), the cumulative risk (i.e., marginal probability) for local bladder cancer recurrence is about 35.5% at 30 months when allowing for the presence of competing risks for bladder metastasis or other metastasis.

The placebo group therefore has a 5% increased risk of failure than the treatment group by 30 months of follow-up.

d. Calculating the $\mathbf{CPC_1}$ values for both treatment and placebo groups at $t_f = 30$:

The formula relating CPC to CIC is given by $\mathbf{CPC_c} = CIC_c/(1 - CTC_{c'})$ where CIC_c = CIC for cause-specific risk event = 1 and $CIC_{c'}$ = CIC from risks for events = 2 or 3 combined

For tx = l, $\mathbf{CIC_1}(t_f = 30) = 0.3087$ and for tx = 0, $CIC_1(t_f = 30) = 0.3554$.

The calculation of $CIC_{c'}$ involves recoding the event variable to 1 for subjects with bladder metastasis or other metastasis and 0 otherwise and then computing $CIC_{c'}$. Calculation of $\mathbf{CIC_{c'}}$ involves the following calculations.

tx = 1 (Treatment A)

t_f	n_f	d_{1f}	$\hat{h}_1(t_f)$	$\hat{S}(t_{f-1})$	$\hat{I}_1(t_f)$	$CIC_{1'}(t_f)$
0	27	0	0	—	—	—
2	27	1	**.0370**	**l**	**.0370**	**.0370**
3	26	2	**.0769**	**.9630**	**.0741**	**.1111**
4	24	0	0	.8889	0	**.1111**
8	**23**	**1**	**.0435**	**.8889**	**.0387**	**.1498**
9	21	1	**.0476**	**.8116**	**.0386**	**.1884**
10	20	1	**.0500**	**.7729**	**.0386**	**.2270**
15	17	1	**.0588**	**.7343**	**.0432**	**.2702**
16	15	1	**.0667**	**.6479**	**.0432**	**.3134**
18	14	0	0	.6047	0	.3134
22	12	0	0	.6047	0	.3134
23	11	0	0	.5543	0	.3134
24	8	0	0	.5039	0	.3134
26	7	0	0	.4409	0	.3134
28	4	1	**.2500**	**.3779**	**.0945**	**.4079**
29	2	0	0	.2835	0	.4079
30	1	0	0	.2835	0	.4079

tx = 0 (Placebo)

t_f	n_f	d_{1f}	$\hat{h}_1(t_f)$	$\hat{S}(t_{f-1})$	$\hat{I}_1(t_f)$	$CIC_{\mathbf{1}'}(t_f)$
0	26	0	0	—	—	—
1	26	0	0	1	0	0
2	24	0	0	.9615	0	0
3	23	0	0	.9215	0	0
5	21	1	**.0476**	**.8413**	**.0400**	**.0400**
6	20	2	**.1000**	**.8013**	**.0801**	**.1201**
7	18	1	**.0556**	**.7212**	**.0401**	**.1602**
10	16	1	**.0625**	**.6811**	**.0426**	**.2028**
12	15	1	**.0667**	**.6385**	**.0426**	**.2454**
14	13	0	0	6835	0	.2454
16	12	1	**.0833**	**.5534**	**.0461**	**.2915**
17	10	0	0	.4612	0	.2915
18	9	0	0	.4150	0	.2915
21	8	1	**.1250**	**.4150**	**.0519**	**.3434**
23	7	0	0	.3632	0	.3434
25	6	1	**.1667**	**.3632**	**.0605**	**.4039**
29	4	0	0	.2421	0	.4039
30	2	0	0	.2421	0	.4039

From these tables, for tx = 1, $CIC_{\mathbf{1}'}((t_f) = 30) = 0.4079$, and for tx = 0, $CIC_{\mathbf{1}'}((t_f) = 30) = 0.4039$.
Thus, for tx =1, $\mathbf{CPC_1}((t_f)=30) = 0.3087/(1 - 0.4079) = \mathbf{0.5213}$, and for tx = 0, $\mathbf{CPC_1}((t_f) = 30) = 0.3554/(1 - 0.4039) = \mathbf{0.5962}$.

6. a. $HR_1(\text{tx} = 1 \text{ vs. tx} = 0) = 0.535 (= 1/1.87)$, p-value = 0.250, N.S.
   b. $HR_2(\text{tx} = 1 \text{ vs. tx} = 0) = 0.987$, p-value = .985, N.S.
   c. $HR_3(\text{tx} = 1 \text{ vs. tx} = 0) = 0.684\ (= 1/1.46)$, p-value = .575, N.S.
7. a. Hazard model formula for the LM model:

$$h_g^*(t, \mathbf{X}) = h_{0g}^*(t) \exp[\beta_1 \text{tx} + \beta_2 \text{num} + \beta_3 \text{size} + \delta_1(\text{txd2}) + \delta_2(\text{numd2}) + \delta_3(\text{sized2}) + \delta_4(\text{txd3}) + \delta_5(\text{numd3}) + \delta_6(\text{sized3})]$$

$g = 1, 2, 3$

where
d2 = 1 if bladder metastasis and 0 otherwise, and
d3 = 1 if or other metastasis and 0 otherwise

b. Hazard ratios for the effect of each of the 3 cause-specific events:

$$\begin{aligned}
HR_1(tx = 1 \text{ vs. } tx = 0) &= \exp(-0.6258) \\
&= 0.535(= 1/1.87) \\
HR_2(tx = 1 \text{ vs. } tx = 0) &= \exp(-0.6258 + .6132) \\
&= 0.987(= 1/1.01) \\
HR_3(tx = 1 \text{ vs. } tx = 0) &= \exp(-0.6258 + .2463) \\
&= 0.684(= 1/1.46)
\end{aligned}$$

c. Corresponding HRs are identical.

8. a. Hazard model formula for the LM_{alt} model:

$$\begin{aligned}
h'_g(t, \mathbf{X}) \quad &= h'_{0g}(t) \exp[\ \delta'_{11}\,\text{txd1} + \delta'_{12}\,\text{numd1} + \delta'_{13}\,\text{sized1} \\
g = 1, 2, 3 \qquad &+ \delta'_{21}\text{txd2} + \delta'_{22}\text{numd2} \\
&+ \delta'_{23}\text{sized2} + \delta'_{31}\text{txd3} \\
&+ \delta'_{32}\text{numd3} + \delta'_{33}\text{sized3}]
\end{aligned}$$

where
d1 = 1 if local bladder cancer recurrence and 0 otherwise
d2 = 1 if bladder metastasis and 0 otherwise, and
d3 = 1 if or other metastasis and 0 otherwise

b. Hazard ratios for the effect of each of the three cause-specific events:
output.

$$\begin{aligned}
HR_1(tx = 1 \text{ vs. } tx = 0) &= \exp(-0.6258) \\
&= 0.535(= 1/1.87) \\
HR_2(tx = 1 \text{ vs. } tx = 0) &= \exp(-0.0127) \\
&= 0.987(= 1/1.01) \\
HR_3(tx = 1 \text{ vs. } tx = 0) &= \exp(-0.3796) \\
&= 0.684(= 1/1.46)
\end{aligned}$$

c. Corresponding hazard ratios arc identical.

9. No interaction SC LM model:

$$\begin{aligned}
h^*_g(t, \mathbf{X}) \quad &= h^*_{0g}(t) \exp[\ \beta_1\,\text{tx} + \beta_2\,\text{num} + \beta_8\,\text{size}] \\
g = 1, 2, 3
\end{aligned}$$

Assumes $HR_1(X) = HR_2(X) = HR_3(X)$ for any X variable e.g., Rx = 0 vs. Rx = 1:
$HR_1(tx) = HR_2(tx) = HR_3(tx) = \exp[\beta_1]$

10. Carry out the following likelihood ratio test:

H_0: $\delta_{gj} = 0 \quad g = 2, 3; \quad j = 1, 2, 3$

where δ_{gj} is coefficient of $D_g X_j$ in the interaction SC LM model
$LR = 2\log L_R - (-2\text{Log}L_F)$ approx χ^2_6 under H_0
R = no-interaction SC (reduced) model
F = interaction SC (full) model

Chapter 10

1. Example: A=2, F=2, so Mt=A/2 + F =3, R=2
   α=0.05, β=0.10
   $\lambda_0 = 0.10$, $\lambda_1 = 0.05$, $\Delta = \lambda_0/\lambda_1 = 2$

   $N_{EV} = \{(1.96 + 1.282)[2(2) + 1]/[\sqrt{2}(2 - 1)]\}^2$
   $= 131.382 \approx 132$

   Using Formula 1:

   $$N = \frac{131.382}{\frac{2}{2+1}\{1 - e^{-2(0.05)(3)}\} + \frac{1}{2+1}\{1 - e^{-(0.05)3}\}}$$

   (N_{EV} ← 131.382 / 0.2192 → P_{EV})

   $= \frac{131.832}{0.2192} = 601.4 \approx 602$

   $N_1 = [2/3]601.4 = 400.93 \approx 401$ and
   $N_0 = 400.9/2 = 200.45 \approx 200$

2. Nev =131.383 = 132 from question 1.

   $$P_{EV1} = 1 - \frac{1}{(0.05)(2)}\left[e^{-(0.05)(2)} - e^{-0.05)(2+2)}\right]$$
   $= 1 - 0.8611 = 0.1389$

   $$P_{EV0} = 1 - \frac{1}{(0.10)(2)}\left[e^{-(0.10)(2)} - e^{-(0.10)(2+2)}\right]$$
   $= 1 - 0.7421 = 0.2579$

   $$N = \frac{131.382}{\frac{2}{2+1}(0.1389) + \frac{1}{2+1}(0.2579)}$$
   $= \frac{131.832}{0.1786} = 738.14 \approx 739$

   $N_1 = [2/3]738.14 = 492.09 \approx 492$ and
   $N_0 = 492.09/2 = 246.04 \approx 246$.

3. The results using Formulae 1 and 2 are somewhat different since Formula 1 yields N=602 whereas Formula 2 yields N=739. Formula 1 uses the median follow-up time M_F in the computation of p_{EVi} whereas Formula 2 computes p_{EVi} by assuming that the time X at which any subject enters the study has the uniform distribution over the accrual period.
4. $N_{LOFadj} = 739/(1 - 0.25) = 985.33 \approx 986$
5. $N_1 = [2/3]985.33 = 656.89 \approx 657$ and $N_0 = 656.89/2 = 328.44 \approx 328$
6. $N_{ITTadj} = 986/(1 - 0\ 05\ -0.10)^2 = 1364.71 \approx 1365$

7. $N_1 = [2/(2+l)]1364.71 = 909.81 \approx 910$ and
   $N_0 = 909.81/2 = 454.91 \approx 455$
8. From question 6, the required accrual rate is $r = N/A = 1365/2 = 682.5 \approx 683$ subjects per year. If this accrual rate is not feasible, but r* was considered feasible, then you can adjust your sample size by reducing the accrual period to $A^* = N/r^*$. For example, if the maximum for r is $r_{max} = 600$/yr, then the required accrual period is modified from A=2 to $A^* = 2.275$ years.

   Now suppose, we keep N_{EV} (=131.382), F=2, R=2, α=0.05, β=0.10, $\lambda_0 = 0.10$, $\lambda_1 = 0.05$, and $\Delta = \lambda_0/\lambda_1 = 2$ all constant, but increase the accrual time to A*=2.275 years. Then we would need to re-compute p_{EV1}, p_{EV0} and N to obtain p_{EV1}= 0.2677, p_{EV0}= 0.1447, and N = 579.541 (prior to adjusting for LOF and Crossovers), which is modified to N* = 1069.51 after adjusting for 25% LOF rate, 5% d_c rate and 10% d_t, rate. For this modified sample size, the modified required accrual rate is $r^* = N^*/A^* = 1069.71/2.275 = 470.11$, which is less than $r_{max} = 600$, so that the study is feasible.

   Note, however, it is also possible to obtain a feasible study if the accrual period remains at A=2, but the follow-up period increases to, say F=4, again keeping N_{EV}(=131.382), R = 2, $\alpha = 0.05$, $\beta = 0.10$, $\lambda_0$0.10, $\lambda_1 = 0.05$, and $\Delta = \lambda_0/\lambda_1 = 2$ all constant. This will require re-computing p_{EV1}, p_{EV0} and N again, followed by adjustments for LOF and Crossovers. In particular, if F is increased (to say F=4), then p_{EV1} and p_{EV0} should correspondingly increase from previously calculated values because the probability for an event occurring should increase if follow-up time is increased.

References

Andersen P.K., Borgan O., Gill R.D., and Keiding N. 1993. *Statistical Models Based on Counting Processes*. Springer Publishers, New York.

AREDS Research Group. 2003. Potential public health impact of age-related eye disease study results. *Arch Opthalmol*, 121: 1621–1624.

Arriagada R., Rutqvist L.E., Kramar A., and Johansson H. 1992. Competing risks determining event-free survival in early breast cancer. *Br. J. Cancer*, 66(5): 951–957.

Berkson J. and Gage R.P. 1952. Survival curve for cancer patients following treatment. *J. Amer. Statist. Assoc.*, 47, 501–515.

Boag J.Q. 1949. Maximum likelihood estimates of the proportion of patients cured by cancer therapy. *J. Roy. Statist. Soc.*, 11, 15–53.

Brookmeyer R. and Crowley J. 1982. A Confidence Interval for the Median Survival Time. *Biometrics* 38 (1): 29–41.

Byar D. 1980. The Veterans Administration study of chemoprophylaxis for recurrent stage I bladder tumors: Comparisons of placebo, pyridoxine, and topical thiotepa. In *Bladder Tumors and Other Topics in Urological Oncology*. Plenum Publishers, New York: 363–370.

Byar D. and Green S. 1980. The Choice of treatment for Cancer Patients based on Covariate Information. *Bulletin du Cancer* 67: 4, 477–490.

Byar D. and Corle D. 1977. Selecting optimal treatment in clinical trials using covariate information. *J Chronic Dis*, 30, 445–459.

Cantor A. 1992. Sample size calculations for the log-rank test: A Gompertz model approach. *J. Clin Epidemiol*, 45, 1131–1136.

Caplehorn J., et al. 1991. Methadone dosage and retention of patients in maintenance treatment. *Med. J. Aust.*, 154, 195–199.

Clayton D. 1994. Some Approaches to the Analysis of Recurrent Event Data. *Statistical Methods in Medical Research*. 3: 244–262.

Cox D.R. and Oakes D. 1984. *Analysis of Survival Data*. Chapman and Hall, London.

Crowley J. and Hu M. 1977. Covariance analysis of heart transplant data. *J. Amer. Stat. Assoc.*, 72, 27–36.

Dixon W.J. 1990. *BMDP Statistical Software Manual*. Berkeley, CA, University of California Press.

Fine J. and Gray R. 1999. A proportional hazards model for the subpopulation of a competing risk. *J. Amer. Stat. Assoc.*, 94, 496–509.

Freedman L.S. 1982. Tables of the number of patients required for clinical trials using the logrank test. *Statistics in Medicine*. 1, 121–129.

Freireich E.O., et al. 1963. The effect of 6-mercaptopmine on the duration of steroid induced remission in acute leukemia. *Blood*, 21, 699–716.

Gebski V. 1997. *Analysis of Censored and Correlated Data (ACCORD)*. Data Analysis and Research Technologies, Eastwood, NSW, Australia.

George S.L. and Desu M.M. 1974. Planning the size and duration of a clinical trial studying the time to some critical event. *J. Chron. Dis.* 27: 15–24.

Goldman A.I. 1984. Survivorship analysis when cure is a possibility: A Monte Carlo study. *Statist. in Med.*, 3: 153–163.

Grambsch P.M. and Therneau T.M. 1994 Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika* 81: 515–526.

Gray R.J. 1988. A class of k-sample tests for comparing the cumulative incidence of a competing risk. *Annals of Stat* 16, 1141–1154.

Gutierrez R.G. 2002. Parametric frailty and shared frailty survival models. *Stata J.* 2: 22–24.

Harrell F. and Lee K. 1986. *Proceedings of the Eleventh Annual SASW User's Group International*: 823–828.

Harris E. and Albert A. 1991. *Survivorship Analysis for Clinical Studies*. Marcel Dekker Publishers, New York.

Hosmer D.W. and Lemeshow S. 2008. *Applied Survival Analysis- 2nd Edition*. John Wiley &Sons, New York.

Kalbfleisch J.D. and Prentice R.L. 2002. *The Statistical Analysis of Failure Time Data- Second Edition*. John Wiley and Sons, New York.

Kaplan E.L. and Meier P. 1958. Nonparametric Estimation from Incomplete Observations. *J. Amer. Statist. Assoc*, 53: 457–481.

Kay R. 1986. Treatment effects in competing-risk analysis of prostate cancer data, *Biometrics* 42, 203–211.

Klein J.P. and Moeschberger M.L. 2003. *Survival Analysis- Techniques for Censored and Truncated Data, 2nd Edition*. Springer Publishers, New York.

Kleinbaum D.G. and Klein M. 2010. *Logistic Regression- A Self Learning Text-Third Edition* (Chapter 14). Springer Publishers, New York.

Kleinbaum D.G., Kupper L.L., and Morgenstern H. 1982. *Epidemiologic Research: Principles and Quantitative Methods*. John Wiley and Sons, New York.

Kleinbaum D.G., Kupper L.L., Nizam A., and Muller, K.A. 2008. *Applied Regression Analysis and Other Multivariable Methods, Fourth Edition*. Cengage Learning, Inc, Florence, KY.

Korn E.L., Graubard B.I., and Midthune D. 1997. Time-to-event analysis of longitudinal follow-up for a survey: choice of the time scale. *Am. J. Epid* 145: 72–80.

Krall J.M., Uthoff V.A., and Harley J.B. 1975. A step-up procedure for selecting variables associated with survival data. *Biometrics*, 31, 49–57.

Lachlin J.M. 1981. Introduction to Sample Size Determination and Power Analysis for Clinical Trials, *Controlled Clinical Trials* 2, 93–113.

Lee E.T. 1980. *Statistical Methods for Survival Data Analysis*. Wadsworth Publishers, Belmont, CA.

Lin D.Y. and Wei L.J. 1989. The robust inference for the Cox proportional hazards model. *J. Amer. Statist. Assoc.* 84: 1074–1078.

Lunn M. 1998. Applying k-sample tests to conditional probabilities for competing risks in a clinical trial. *Biometrics* 54, 1662–1672.

Lunn M. and McNeil D. 1995. Applying Cox regression to competing risks. *Biometrics* 51, 524–532.

Makuch R.W. and Parks W.P. 1988. Statistical methods for the analysis of HIV-1 core polypeptide antigen data in clinical studies. *AIDS Research and Human Retroviruses* 4: 305–316.

Pasternack B.S. and Gilbert H.S. 1971. Planning the duration of long-term survival time studies designed for accrual by cohorts. *J. Chronic Dis.* 24: 681–700

Pencina M.J., Larson M.G., and D'Agostino R.B. 2007. Choice of time scale and its effect on significance of predictors in longitudinal studies. *Statist. in Med.* 26: 1343–1359.

Pepe M.S. and Mori M. 1993. Kaplan-Meier, marginal or conditional probability curves in summarizing competing risks failure time data? *Statist. in Med.* 12, 737–751.

Prentice R.L. and Marek P. 1979. A qualitative discrepancy between censored data rank tests. *Biometrics*, 35(4): 861–867

Prentice R.L., Williams B.J., and Peterson A.V. 1981. On the Regression Analysis of Multivariate Failure Time Data. *Biometrika* 68 (2):373–79.

Rubinstein L., Gail M., and Santner T. 1981. Planning the duration of a comparative clinical trial with loss to follow-up and a period of continued observation. *J. Chron. Dis.* 34: 469–479.

Schoenbach V.J., Kaplan B.H., Fredman L., and Kleinbaum D.G. 1986. Social ties and mortality in Evans County, Georgia. *Amer. J. Epid.*, 123:4, 577–591.

Schoenfeld D. 1982. Partial residuals for the proportional hazards model. *Biometrika*, 69, 51–55.

Stablein D., Carter W., and Novak J. 1981. Analysis of survival data with non-proportional hazard functions. *Controlled Clinical Trials*, 2, 149–159.

Tai B.C., Machin D., White I., and Gebski V. 2001. Competing risks analysis of patients with osteosarcoma: a comparison of four different approaches. *Stat. Med.* 20(5): 661–684.

Thiebaut A.C. and Benichou J. 2004. Choice of time-scale in Cox's model analysis of epidemiologic cohort data: a simulation study. *Stat. Med.* 23: 3803–3820.

Wei L.J., Lin D.Y., and Weissfeld L. 1989. Regression analysis of multivariate incomplete failure time data by modeling marginal distributions, *J. Amer. Statist. Assoc.*, 84 (408).

Zeger S.L. and Liang C.Y. 1986. Longitudinal Data Analysis for Discrete and Continuous Outcomes. *Biometrics* 42, pp 121–130.

Index