Extending the Use of PROC PHREG in Survival Analysis

Christopher F. Ake, VA Healthcare System, San Diego, CA
Arthur L. Carpenter, Data Explorations, Carlsbad, CA

ABSTRACT

Proc PHREG is a powerful SAS® tool for conducting proportional hazards regression. Its utility, however, can be greatly extended by auxiliary SAS code. We describe our adaptation of a group of existing public domain SAS survival analysis macros, as well as our development of additional control, management, display, and other macros, to accommodate a project with requirements that included:

- large data sets to be analyzed in counting process format and containing time-varying covariates
- missing data requiring multiple imputation procedures
- varying combinations of covariates, outcome events, censoring mechanisms, and origin definitions resulting in several hundred different models

We also describe how to provide in this analysis process for:

- Exploratory Data Analysis (EDA)
- assessment of model assumptions
- consolidation of multiple analyses
- final output HTML displays packaged for easy access

Tools extending the capabilities of PHREG are already available-- join us to learn more about them.

Keywords: PROC PHREG, counting process format, survival analysis, proportional hazards model

INTRODUCTION

In the three decades since its introduction, the proportional hazards model has been established as the first choice of many persons wanting to perform regression analysis of censored survival data. PHREG has emerged as a powerful SAS procedure to conduct such analyses by itself. Its capabilities can be greatly extended, however, by use of a variety of public domain macros as well as customization techniques.

These macros have been made possible in part by theoretical advances that have provided a rigorous foundation for the counting process framework that underlies proportional hazards regression. For everyday practitioners these advances have resulted in the availability of a variety of residuals that can be used to assess functional form of covariates, proportional hazards assumptions, and influence of individual observations, in somewhat parallel fashion to their use in linear regression.

These developments in counting process theory have also facilitated the development of a new method of providing data as input to proportional hazards regression. Using input data created with this method, which has come to be known as data in counting process style or counting process format, PHREG can handle, among other things, time-dependent covariates and left-truncated phenomena.

Since the availability of counting process format is relatively recent, it is often relatively less discussed than alternatives such as the use of programming statements in the PROC PHREG step itself, for example, to define time-varying covariates. Paul Allison's well-known *Survival Analysis Using the SAS System*, for instance, gives examples of the use of such programming statements (pp. 138-154) but does not discuss counting process format at all. Because it appears still less well known to many SAS users we have chosen to make counting process format the centerpiece of our discussion in this paper.

PHREG's ability to handle a greater variety of data, in turn, confers additional value on customizing its use to allow user-defined control of the entire survival analysis process it can be contained in. So we briefly also discuss macros which can provide a range of options for creation of control data sets, for running exploratory data analyses, for creation of datasets with specified outcomes and/or covariates, for transformation of input data into counting process format, for running various diagnostics, for handling missing data, and for categorizing, displaying, and providing access to output.

COUNTING PROCESS FORMAT

Typical survival analysis data often takes the form of one record per subject. Each of these records is a vector of the sort (T,I,…) where T has the value of time since the origin, and is either the time of an event of the kind being studied, in the case when the indicator variable I takes the value 1, say, or otherwise is a censoring time, in which case I will have the value 0, say.

Data in counting process format, on the other hand, may often contain more than one record per subject; for any subject with multiple records in the dataset, each such record represents one interval for that subject. Each such record is of the form (T1,T2,I,…), where T1 represents the time at which the interval started, T2 the time at which the interval ended, and I, as before, is an indicator variable showing the status of the interval. The indicator I could take one value to represent at event occurring at time T2, another to indicate censoring at T2, and possibly other values as well to represent such occurrences as a competing event. The actual time interval represented by this record can be represented as (T1,T2], i.e., open on the left, closed on the right, so that the time instant T2 itself is included in the time interval but T1 is not. Thus an event or change in status occurring at T2 would belong to this interval but one occurring at T1 would not –it would belong to the preceding time interval.

Counting process format can easily accommodate a number of special features in one's data, including multiple events of the same type, multiple events of a different type, time-dependent covariates, and discontinuous intervals of risk, as well as any combination of these features, as we now illustrate.

Multiple events of the same type

As an example, assume that a subject (with time-independent indicator covariates drugA=1, sex=1 and race=1) has an event on months 100 and 185 and has now been followed to month 250. This subject would be coded as three observations or "lines" of data whose intervals are (0,100], (100,185], (185,250] with corresponding exit status codes of 1, 1, and 0. The data file for this subject is below.

Subj	Entry	Exit	Status	DrugA	Sex	Race
1	0	100	1	1	1	1
1	100	185	1	1	1	1
1	185	250	0	1	1	1

Note that time-independent (static) indicators repeat for the three lines of code for that one subject, whereas the exit status variable changes across lines. The exit status values reflect the one type of event (1), as well as the end of observation (0). For analyses of time to the first event, we ignore the last two lines of the data in the analysis. This can be done by taking the first status=1 for each subject, and deleting the observations after observing status=1 for that subject. *n.b.* the values used for the STATUS variable are arbitrary and the effect that the value is to have is indicated to PHREG through the MODEL statement.

In the following example of SAS code that uses the above data for the PHREG procedure, Status(0) indicates to SAS that an event of interest has *not* occurred at that exit time, and that the subject is still at risk for the event(s) of interest at that time. SAS assumes that the other exit status values provided in the data set are the event(s) of interest.

```
proc phreg;
model (Entry, Exit) * Status(0) =
              DrugA Sex Race;
run;
```

Multiple events of a different type

Assume instead that the subject (still with time-independent indicator covariates drugA=1, sex=1 and race=1) has a Type 1 event on month 100 and Type 2 event on month 185. We still observed the subject in (0, 250]. This subject would still be coded as three observations or "lines" of data whose intervals are (0,100], (100,185], (185,250] with corresponding exit status codes of 1,2, and 0:

Subj	Entry	Exit	Status	DrugA	Sex	Race
1	0	100	1	1	1	1
1	100	185	2	1	1	1
1	185	250	0	1	1	1

The time-independent (static) indicators still repeat for the three lines of code for that one subject, whereas the exit status variable changes across lines. The exit status values reflect the types of events (1=Type1, 2=Type2), as well as the end of observation (0). For analyses of time to the first event of a particular type, we ignore the information after an event of that type occurs.

In the first PHREG step below, the Status(0,1) indicates to SAS that 0 and 1 are the *censoring* (as defined above) values, and the other possible values are the events of interest. For the second PHREG step, the Status(0) indicates to SAS that 0 is the only censoring value, and the other values (at least the values of 1 and 2) are the events of interest.

```
proc phreg;
model (Entry, Exit) * Status(0,1) =
              DrugA Sex Race;
run;

proc phreg;
model (Entry, Exit) * Status(0) =
              DrugA Sex Race;
run;
```

Time-dependent covariates

Now assume that the subject (again with time-independent indicator covariates sex=1 and race=1) did not have an event throughout the observation period (0,250], but was exposed to drugA during periods (0,100] and (185,250], but not during (100,185]. This subject would be coded as three observations or "lines" of data whose intervals are (0,100], (100,185], (185,250] with corresponding exit status codes of 50, 50, and 0:

Subj	Entry	Exit	Status	DrugA	Sex	Race
1	0	100	50	1	1	1
1	100	185	50	0	1	1
1	185	250	0	1	1	1

For the first two lines of code, the exit status variable is now coded as 50 (instead of 0) to reflect a change in a time-dependent variable (DrugA) as opposed to a change in the outcome variable. We code the status variable in the last line as 0 because the reason for the exit time is the end of the observation period, not really a change in DrugA status.

In the following PHREG step, the Status(0, 50) indicates to SAS that 0 and 50 are the censoring values, and the other values are the events of interest.

```
proc phreg;
model (Entry, Exit) * Status(0,50)=
              DrugA Sex Race;
run;
```

Discontinuous intervals of risk

As an example, assume that a subject (with time-independent indicator covariates drugA=1, sex=1 and race=1) was observed during (0,100] and (185,250], had an event at 250 (the end of the study), but was not observed during (100,185]. This subject would be coded as *three* observations or "lines" of data whose intervals are (0,100], (100,185], and (185,250] with corresponding exit status codes of 0, 99, and 1, as follows.

Subj	Entry	Exit	Status	DrugA	Sex	Race
1	0	100	0	1	1	1
1	100	185	99	99	1	1
1	185	250	1	1	1	1

It is reasonable to impute values for Sex and Race during (100,185] for that subject. Imputing a value of 1 for DrugA may be questionable, however, so a value of 99 (for missing) may be more appropriate in some cases. The event status value of 0 in the first risk interval denotes the end of an observation period (not the overall observation period) without an event or a change in covariate status. The event status value of 99 in the second risk interval denotes that

this is a missing interval. We would usually delete this interval before analysis (hence, the reason for not being concerned about the value of DrugA during the unobserved interval), although it might make sense in some projects to have a "place-holder" for this interval for future reference.

As another example, assume that a subject (with time-independent indicator covariates drugA=1, sex=1 and race=1) was observed during (100,250], had an event at 230, but was not observed at all during (0,100]. This subject would be coded as *three* observations or "lines" of data whose intervals are (0,100], (100,230] and (230,250] with corresponding status codes of 90, 1 and 0:

Subj	Entry	Exit	Status	DrugA	Sex	Race
1	0	100	90	99	1	1
1	100	230	1	1	1	1
1	230	250	0	1	1	1

Here the event status value of 90 in the first risk interval denotes that this subject entered observation after time=0 (ie. a staggered entry); that is, the value reflects a different type of *missingness* for that interval. We would usually delete this interval before analysis (hence, the reason for not being concerned about the value of DrugA during the unobserved interval), but again it might make sense in some projects to have a "place-holder" for this interval for future reference.

Combinations of the above types

Quite often outcome and covariate values vary over time for many subjects; hence, a combination of the above set of possibilities will occur in many situations.

Assume, for example, that a subject (with time-independent indicator covariates sex=1 and race=1) had the following covariate/outcome history for the overall observed time period (0,250]. First, the person entered in the database at times 50 and 180, and exited out of the study at times 120 and 240. From the start, the person took DrugA, but only until time 100. The person started to take the drug again at time 200 until last seen. This person had an event at time 230.

After sorting through the various times and types of changes in the subject's outcome and covariate history, and determining the appropriate event status values at each end point, the subject's entire history is coded as follows:

Subj	Entry	Exit	Status	DrugA	Sex	Race
1	0	50	90	99	1	1
1	50	100	50	1	1	1
1	100	120	50	0	1	1
1	120	180	99	99	1	1
1	180	200	50	0	1	1
1	200	230	1	1	1	1
1	230	240	0	1	1	1
1	240	250	99	99	1	1

DATASET CREATION

Programming statements in PHREG versus counting process format

To use data with time-varying covariates as input to PHREG you have the option of using programming statements in the PHREG proc step itself, versus putting your data into counting process format prior to calling PHREG.

Each method provides a powerful set of data handling tools--but each also has potential pitfalls.

Using programming statements in the PHREG proc step allows one to use a wide variety of DATA step statements and functions, which can be used in PHREG the same way they are used in a DATA step. The parallel with the DATA step, however, can be misleading in one way. In the DATA step, SAS is acting on one record at a time. But the programming statements in PHREG are better thought of as applying to one event time at a time. (SAS is constructing the partial likelihood one term at a time, and each term applies to one event time.)

Suppose, for example, we have a dataset in the form of one record per patient, with each record containing a small number of time-independent covariates as well as the cumulative amount of a drug of interest used in each month of a two-year observation period stored in the variables DrugB1 through DrugB24. Suppose that survtime is our survival time variable and status, as before, is our censoring variable. To create a proportional hazards model with cumulative amount of DrugB used as one of the covariates in the form of a variable cumB, which is thus a time-varying covariate, one could use the following PHREG step:

```
proc phreg;
model survtime*status(0) = cumB …;
array cumB (*) drugB1-drugB24;
do i = 1 to 24;
if survtime=i then cumB=DrugB[i]; end; run;
```

Suppose the first event occurred in month 5. Then for the value i=5 SAS will begin constructing the partial likelihood by taking, for each subject in the risk set as of that month, their value of DrugB5 as the value of the model covariate cumB. (Here the array name cumB can be used in the model statement even though it is not even defined until the following statement).

Now suppose that in addition to this data, each subject record also contains an additional variable which records the month when a condition of interest increased in severity, called incmonth. Suppose we wanted to add to our covariates a binary indicator of whether a patient's severity level had increased or not, to be called inc. This new binary indicator will be a time-dependent covariate equal to 0 before the month of increase for the subject and 1 for that month and afterward.

We could use the following code:

```
proc phreg;
model survtime*status(0) = cumB inc …;
array cumB (*) drugB1-drugB24;
do i = 1 to 24;
if survtime=i then cumB=DrugB[i]; end;
if incmonth ge survtime then inc=1;
else inc=0;
run;
```

Here is where a divergence from the DATA step processing occurs. When SAS handles the second IF statement, it compares for each event time and each subject in the risk set for that event time, that subject's value of incmonth (a time-independent value) with the value of survtime for that event time.

Using counting process format may seem simpler than using these kinds of programming statements from an analytic point of

view, i.e., the analysis steps themselves including the PROC PHREG step are straightforward because all the required data construction has taken place. BUT the transformation of the input data into counting process format can necessitate quite complicated programming.

As Terry Therneau and Patricia Grambsch note in *Modeling Survival Data: Extending the Cox Model* (p. 76), the flexibility of counting process format is its greatest strength but also its weakness, because it is possible for the user who sets up the data to mess up the data. They speak from first-hand experience, including a real example from their institution of exactly such a mess-up that got as far as a final draft manuscript before being noticed.

As if that were not enough, we ourselves can add to their example: we only caught a nontrivial mistake in our results, deriving from a single incorrect line in our main dataset creation macro, which put our input data into counting process format, after we had sent our manuscript off to a major medical journal and it had been accepted for publication.

The moral of this story, which we cannot emphasize too strongly, is:

CHECK ALL DATA TRANSFORMED INTO COUNTING PROCESS FORMAT THOROUGHLY TO INSURE THAT EACH POSSIBLE KIND OF INTERVAL HAS BEEN CREATED CORRECTLY!

Thus of the four steps Therneau and Grambsch list (p. 77) for effective use of counting process format:

- thinking through the problem
- creating the dataset in counting process format
- checking the dataset
- fitting the model followed by considering the results

we strongly emphasize the necessity of the third step, and suggest that, in a case of any complexity, you print out sample records from the dataset--not just once, but as often as required during writing or modifying the dataset creation program--to insure that every possible configuration of (start, stop] intervals is being generated correctly.

We also note incidentally that perhaps due to its relative newness the SAS documentation through at least Version 8.2 for PHREG says that when using the BASELINE statement with data in counting process format no output dataset will be created that contains survivor function estimates, but when one runs PHREG with such data the output is indeed created (thanks to Terry Therneau who pointed this out in e-mail communication).

Data Creation Macro

In our modeling we began with data in the form of one record per subject per month, with 36,766 subjects over a period of 102 months, thus having a total of 3,750,132 records. To transform this data into counting process format our %ENDPOINTS macro received the data sorted by subject ID and month and began generating (start, stop] intervals.

The structure of our data creation macro is to start a new interval each time it encounters the first record of a new subject and continue through additional month records for that subject until it encounters either:

a) an outcome event for that subject;
b) a change in drug exposure status for that subject;
c) a change in activity status for that subject,

in which case it outputs an interval for that subject.
To check b) and c) the macro creates additional variables for each subject for exposure status and activity status which are lagged by one month to allow detecting a change in status by comparing the present month with the lagged value from the preceding month.

The program checks to make sure no intervals are created for a subject after a listed month of death. Since we examined a variety of different drug exposures, including exposure to each of three categories of antiretrovirals, or to any of them, as well as the corresponding cumulative exposures, with quadratic or even cubic terms in some cases as well as linear terms, the program is set to check the lagged versus current values corresponding to whichever exposure definition the current model is using. The macro has options to allow the use of a variety of possible censoring rules for defining a subject's activity status, depending on exactly when a subject is considered under observation. If desired it can write a subject's intervals of inactivity as well as of activity to the output dataset. It also can break intervals at designated calendar points such as December 31, 1997, say, in case we are interested in comparing survival experience in certain year groups.

Missing data

Our programmatic work to accommodate missing data through use of multiple imputation via PROC MI and PROC MIANALYZE was described in Ake and Carpenter (2002).

MAYO MACROS

A series of public domain SAS macros have been written at the Mayo Clinic in Rochester, Minnesota. These macros allow the user to take advantage of theoretical advances by working with PHREG in ways not directly anticipated by the developers of the procedure. These include the use of martingale residuals in a variety of diagnostics. The macros were developed under the leadership of Therneau and Grambsch, and are maintained by the clinic staff, they have been thoroughly tested and used there, but the Mayo Clinic does not warrant their use in any way. They are discussed and their use illustrated in Therneau and Grambsch's book *Modeling Survival Data: Extending the Cox Model,* and are available at www.mayo.edu/hsr/people/therneau/book/book.html.

These macros include the %SURV program, which can calculate Kaplan-Meier survival curves together with standard errors, confidence limits, and median survival times, with numerous options to print and plot results. It allows a number of options for how to calculate confidence intervals (Greenwood, log and logit transformations). Upon request %SURV can also compute k-sample logrank statistics with plotting, in which case it calls a separate %SURVLRK macro. The %SURVTD variant can perform the same operations as %SURV for data that is left-truncated or may have time-varying covariates.

The %PHLEV macro generates robust variance estimates for a proportional hazards model by using one kind of residuals the model outputs to compute an approximate jackknife variance

estimate. The %PHLEV macro invokes PHREG; it does not, as written, accommodate many of the possible options for PHREG, but they can be added in.

Their %SCHOEN macro uses another kind of residuals that can be output by PHREG, the scaled Schoenfeld residuals, to produce plots and tests of proportional hazards assumptions;. The %SCHOEN macro plots the scaled Schoenfeld residual corresponding to a given covariate from each event time against time or a function of time; if the proportional hazards assumption for the variable in question holds then a line fit to the plot should have zero slope. To aid in detecting the possible form of departure from proportional hazards, a smooth curve with confidence bands is added to the plot using a spline fit. The macro does not, as provided, permit data in counting process format to be used, and thus one of our adaptations was to expand this macro to accommodate our use of data in counting process format

CONTROL MACROS

The number of parameters required for many of the Mayo macros can be quite staggering. This is a direct result of the number of options and supporting statements used by PHREG, as well as, the other SAS/STAT® procedures which can be used during an analysis. Moreover, since many of the macro calls can be nested, the control and specification of the parameters can be problematic. In our modeling, we wrote a series of macros to control the processing flow and to specify many of the parameters for the calls to the Mayo macros. The overall control of the process was through the use of a SAS data table as the initial specification for each model. The construction of the control data table and how it was used to generate macro variables and macro parameters is discussed in specific terms in Ake and Carpenter (2002) and in more general terms in Carpenter and Smith (2000).

ANALYSIS MACROS

In the course of our analysis work, we found it convenient to create several separate smaller macros that make use diagnostic properties of model residuals.

One of these macros investigates the functional form of covariates. The proportional hazards model by itself assumes that the log of the hazard is linear in each of the covariates, so if this is questionable then one of the methods you can use is a so-called “simple” approach developed by Therneau and others (Therneau et al., p. 87), which works well when data are uncorrelated. In this method one plots the martingale residuals from a null model, i.e., a model whose beta vector is the null vector, against each covariate in question. If a smoothed plot is then superimposed on the points plotted it will, if certain conditions are met, actually show the correct functional form of the covariate. Our macro thus calls PHREG to run a null model and then merges the PHREG output with the original input data for the model to produce a data set containing both the covariates and the residuals. Then the following code plots the martingale residuals against each covariate listed by the macro variable &FFVARS.

```
title1 "Functional form assessment - simple
method" ;

 proc gplot data=temp1 ;
  %do i=1 %to &nffs;
    %let xvble=%scan(&ffvars,&i);
    title2 "Plot of martingale residuals" ;
    plot resmart*&xvble / haxis=axis1
                          vaxis=axis2 ;

    symbol i=sm60s v=J font=special ;
    label resmart='Residual' ;
    /* resmart is variable name given in
       phreg macro to the saved martingale
       residuals */
    axis2 order=-2 to 2 by 0.5 major=none;
    axis1 minor=none ;
  %end ;
 run ;
 quit ;
```

Another macro plots the dfbeta residuals which are used to examine influence or how important each observation is to the fit of the model. These can be called as an option from within our overall analysis macro for any model in which one is interested.

OUTPUT MACROS

The procedural output (text and graphical) for each model can be quite voluminous. In addition since there were a large number of models in the study it became necessary to manage the output. A combination of standardized naming conventions and directory structures were used to store graphs and output (all output tables were saved as HTML files using the Output Delivery System, ODS).

Macros were written to navigate the hundreds of graphs, charts, and tables that were generated by the analyses. HTML indices with the capability of drilling down to the table or chart of interest were provided. PROC PRINT was used along with the control files to generate the lists of tables as well as a hierarchical search structure. Specific information on these techniques can be found in Carpenter and Smith (2002b).

SUMMARY

Providing input data to PROC PHREG in counting process format allows you to accommodate time-varying covariates, discontinuous intervals of risk, or multiple events of the same or different types. While providing great flexibility in defining input data records, use of counting process format should be stringently tested and monitored to insure intervals are generated correctly. In general, the capacities of PROC PHREG can be greatly extended by making use of analysis, control, and other macros, a number of which are already available in the public domain.

ACKNOWLEDGEMENTS

Dr. Jacinte Jean was responsible for the schemata we use here to discuss counting process format as well as initial development

of some of the code used in our modeling. Dr. Samuel A. Bozzette has directed the entire modeling project. Dr. Thomas A. Louis of the Johns Hopkins Bloomberg School of Public Health has served as a statistical consultant.

REFERENCES

Ake, Christopher F. and Arthur L. Carpenter, 2002, "Survival Analysis with PHREG: Using MI and MIANALYZE to Accommodate Missing Data", Proceedings of the 10th Annual Western Users of SAS Software Regional Users Group Conference (2002), pp. 102-107, Cary, NC: SAS Institute Inc.

Allison, Paul D., 1995, *Survival Analysis Using the SAS® System :A Practical Guide*. Cary, NC: SAS Institute.

Carpenter, Arthur L. and Richard O. Smith, 2000, "Clinical Data Management: Building a Dynamic Application", Proceedings of the Pharmaceutical SAS® Users Group Conference, Cary, NC: SAS Institute Inc., paper dm10, pp. 151-156. Also in the Proceedings of the 8th Annual Western Users of SAS Software Regional Users Group Conference (2000), pp. 3-8. Also in the Proceedings of the 10th Annual Western Users of SAS Software Regional Users Group Conference (2002), pp. 487-492, Cary, NC: SAS Institute Inc.

Carpenter, Arthur L. and Richard O. Smith, 2002a, "Library and File Management: Building a Dynamic Application", Proceedings of the Twenty-Seventh Annual SAS® Users Group International Conference, Cary, NC: SAS Institute Inc., paper 21.

Carpenter, Arthur L. and Richard O. Smith, 2002b, "ODS and Web Enabled Device Drivers: Displaying and Controlling Large Numbers of Graphs", Proceedings of the 10th Annual Western Users of SAS® Software Regional Users Group Conference (2002) pp. 182-189, Cary, NC: SAS Institute Inc.

Therneau, Terry M. and Patricia M. Grambsch, 2000, *Modeling Survival Data: Extending the Cox Model*. New York: Springer-Verlag.

AUTHOR CONTACT INFORMATION

Chris Ake
Health Services Research & Development
Veterans Affairs San Diego Healthcare System
3350 La Jolla Village Drive, 111N-1
San Diego, CA 92161
(858) 552-8585 x 5492
Christopher.Ake@med.va.gov

Art Carpenter
Data Explorations
2270 Camino Vida Roble, Suite L
Carlsbad, CA 92009
(760) 945-0613
art@caloxy.com

TRADEMARK INFORMATION