Paper 428-2013

Cox Proportional Hazard Model Evaluation in One Shot

Polina Kukhareva, Collaborative Studies Coordinating Center, UNC, Chapel Hill, NC, USA

ABSTRACT

Cox proportional hazard models are often used to analyze survival data in clinical research. This article describes a macro that makes producing the correct diagnostics for Cox proportional hazard models fast and easy. The macro has three advantages over performing all the diagnostics one by one. First, it makes it easy to run diagnostics for a long list of similar models. Second, it allows the specification of the variables for which diagnostics should be run. Third, it produces a comprehensive list of plots and tables necessary for evaluation of the Cox proportional hazard model assumptions as recommended in the SAS® course "Survival Analysis Using the Proportional Hazards Model." This macro can help save hours of code-writing time for a programmer who performs survival analysis.

INTRODUCTION

Cox proportional hazard models are often used to analyze survival data in clinical research. Unfortunately, producing the correct diagnostics necessary to confirm model assumptions can be time-consuming, especially when they are needed for a long list of models. This article describes a macro which makes producing correct diagnostics fast and easy.

Working as a statistical programmer for a large clinical study I often need to write and run code for Cox proportional hazard models. Many of those models would just differ one from the other by one or two predictors. It's so easy to make a mistake and forget to add or delete one predictor from a long list. So over time I developed some techniques which make producing long sets of models easier by enabling me to run the diagnostics on each model of that large set without having to set up the diagnostic-production code for each model. I achieved a further savings of time and effort by including both brief diagnostic tables and supporting graphs in the same macro. The macro doesn't include all possible diagnostics, but provides enough results for a biostatistician to locate a problem and ask for a more detailed report.

My experience comes from a randomized clinical trial but the macro is probably even more useful for a clinical study where participants are not randomized and the primary analysis more often requires controlling for lots of covariates of interest and assessment for confounding.

Objectives

Describe a macro which:

- Verifies the assumptions of the Cox proportional hazards model.
- Assesses the model for interactions and confounding between the predictor variables.

Model Assumptions

- The relationship between the continuous predictor variables and the log hazard should be linear.
- The effects of the predictor variables are the same at all values of time. In other words, the hazard ratio is constant over time.

INPUT DATA SET

The macro was developed in SAS 9.3 but works equally well in SAS 9.2. The following procedures are used in the macro: PHREG, SQL, DATASETS, SGPLOT, RANK, LIFETEST, MEANS, PRINT, and SORT.

The macro can be used to diagnose whether a particular Cox proportional hazard model is suitable for a given survival data set. It can be used with any survival data set which has at least one predictor variable.

All the pictures and tables presented in this paper are generated using a simulated data set. This data set includes response variable event (events coded as 1, censored observations coded as 0), follow-up variable time, five continuous predictor variables (age, systolic_bp, diastolic_bp, ldl, bmi) and five categorical predictors (diabetes, smoking, sex, treatment group, activity). Activity has three levels ('low', 'high', 'average') and all other categorical variables have two levels ('0', '1'). The data set is intended to represent a small data set from a cardiovascular drug study with time following an exponential distribution. Variable Linpred represents the linear predictor for the model (the sum of the predictor variables multiplied by the matching beta coefficients). Although the author did not purposely intend this, the random nature of the simulated data set resulted in no violations of proportional hazard assumptions, no interactions among the predictors, and no confounding; the user is cautioned that this is not necessarily representative of the data they may use with the macro.

```
t = 2+rand("WEIBULL", 1, lambdaT*linpred);  * time to event;
c = 2+rand("WEIBULL", 1, lambdaC);    * time to censoring;
time = min(t, c);     * which came first?;
event = (t lt c);
```

MACRO PARAMETERS

The macro has 10 parameters which are specified in Table 1.

Parameter	Description	Example
Required Parameters		
_NUMBER	The model number (user-defined) – appears in the title (useful for distinguishing between models in a large set)	1
_DATA	Name of the data set containing survival data	rq.simcox
_PREDICTORS	List of predictor variables (this list can have some extra variables which later can be excluded using _Exclude or _Stop parameter)	age sex treatment
_CLASS	The list of variables which need to be included in a CLASS statement (categorical and other variable types)	sex(ref="0") treatment(ref="0")
_OUTCOME	Name of the outcome variable (as per standard practice, events are coded as 1 and censored observations are coded as 0)	CV_event
_FU	Follow-up time	CV_time
Optional Parameters with specified default values		
_STOP=_NOT_SPECIFIED	The last variable from predictor list user wants to include in the specific model	sex
_EXCLUDE=_NOTHING	List of variables from predictor list which shouldn't be included in the specific model (allows quick customization of models using a pre-set list of predictors)	age
_CHECK=_ALL	List of variables for which user wants to produce diagnostics	sex
_RQ=cox_assumptions	List of characters to be included in the name of output RTF file. Name of RTF file is composed from _Rq and _Number parameters.	cox_assumptions
_INTERACTIONS=_ALL	List of interactions of interest	sex*age sex*treatment
_FOOTNOTE=%str(&sysdate , &systime -- produced by macro check_cox_assumptions)	Text of the footnote	%str(&sysdate, &systime -- produced by macro check_cox_assumptions)

Table 1. Macro parameters

ADVANTAGES OF THE MACRO

The macro described in this paper has three advantages over performing all the diagnostics one by one as they are described in the course notes. First, it makes it easy to run diagnostics for a long list of similar models. Second, it allows the specification of variables for which diagnostics should be run within each model specified. Third, it produces a comprehensive list of plots and tables necessary for evaluation of the Cox proportional hazard model. Also, each graph comes with a little description section inserted using ODS text.

1. HOW DO I CALL THE MACRO FOR A LONG LIST OF SIMILAR MODELS?

Sometimes a programmer needs to check the proportional hazards assumption for many models with a similar list of predictors. Imagine that you need to run three models as specified in Table 2 (below; “+” indicates that a variable is to be included in the model). You could do this in a very tedious, manual, time-consuming way by running all diagnostics for each model and specifying the list of predictors every time. The macro allows the saving of that time and effort by letting the user customize the existing standard list of predictor variables using _STOP and _EXCLUDE macro parameters. _STOP is the last variable from the predictor list which needs to be included in the model. _EXCLUDE is the list of variables which need to be excluded from the model. With this macro you can save a list of all 10 predictors as a macro variable and then only change _STOP and _EXCLUDE parameters in the macro. For example to run diagnostics for model 1 you only need to specify _STOP=activity and _EXCLUDE= `systolic_bp diastolic_bp ldl diabetes smoking sex.`

Variable	Model 1	Model 2	Model 3
age	+	+	+
systolic_bp		+	+
diastolic_bp		+	+
ldl		+	+
bmi	+	+	+
diabetes		+	+
smoking		+	+
sex		+	
treatment	+	+	+
activity	+	+	+

Table 2. Specification of the models.

Below is an example of how a user can call the macro for these three models.

```
%let predictors=age systolic_bp diastolic_bp ldl bmi diabetes smoking sex treatment
activity;
%let class = diabetes(ref="0") smoking(ref="0") sex(ref="0") treatment(ref="0")
activity(ref="average")

%check_Cox_assumptions_all (_NUMBER=1, _DATA=simCox, _PREDICTORS=&predictors,
_class=&class, _OUTCOME=CV_event, _FU=CV_time, _STOP= activity,
_EXCLUDE= systolic_bp diastolic_bp ldl diabetes smoking sex)

%check_Cox_assumptions_all (_NUMBER =2, _DATA =simCox, _ PREDICTORS=&predictors,
_class=&class, _OUTCOME=CV_event, _FU=CV_time, _STOP=activity, _EXCLUDE=NOTHING)

%check_Cox_assumptions_all (_NUMBER =3, _DATA =simCox, _ PREDICTORS=&predictors,
_class=&class, _OUTCOME=CV_event, _FU=CV_time, _STOP=activity, _EXCLUDE=sex)
```

How does the macro exclude/include variables from the predictor list?

The macro excludes variables from the pre-specified list of predictors using two different methods. The first of these is a simple data step loop by starting at the first of the variables in the predictor list and ending when it encounters the variable specified by _STOP. At the same time, the macro checks each variable in the predictor list against the variable(s) in _EXCLUDE – any matches are excluded. After the data step we use PROC SQL to put the list of predictors in a macro variable called PREDICTORS_USED. The default value for _EXCLUDE is _NOTHING. If _EXCLUDE is not specified all the variables from the list of predictors (macro variable _PREDICTORS) up until the variable listed in _STOP will be included in the PREDICTORS_USED macro variable. If the macro parameter _STOP is not specified then the macro assigns it the value of the last variable in the _PREDICTORS parameter.

```
data predictors_&_NUMBER;
   %let cnt=0;
   format predictor $30.;
%do  %while (%qscan(%UPCASE(&_PREDICTORS),&cnt, %str())^=%str(%UPCASE(&_stop)));
   %if (%eval(&cnt) = %eval(%sysfunc(countw(&_PREDICTORS)))) %then %do;
         %put ERROR: CHECK SPECIFICATION OF PARAMETERS; %return;
   %end;
   %let var=%qscan(%UPCASE(&_PREDICTORS),%eval(&cnt+1),%str( ));
   %if  ((&var in &exclude)=0) %then %do;
         order=&cnt;
         predictor="&var";
         output;
   %end;
   %let cnt=%eval(&cnt+1);
%end;
run;
proc sql;
   select distinct predictor into :predictors_used separated by ' '
   from predictors_&_NUMBER
   order by order;
quit;
```

If parameters are not specified (e.g., _CHECK), they will assume their default values if they have one (_CHECK's default value is _ALL). The default values can be found in the first column of the Table 1.

2. HOW DO I SPECIFY VARIABLES FOR WHICH DIAGNOSTICS SHOULD BE RUN?

This macro also allows the specification of variables within each model for which diagnostics should be run using the _CHECK parameter. For example, if you only want diagnostics run for the variable age, specify _CHECK=age. If you want to run diagnostics for all variables in the predictor list, you can either omit specifying the _CHECK parameter, or specify _ALL (the default for this parameter).

```
%check_cox_assumptions_all (_NUMBER=1, _DATA=rq.simcox,_PREDICTORS=&predictors,
_CLASS=&class, _OUTCOME=CV_event, _FU=CV_time, _CHECK=age,_STOP=activity,
_EXCLUDE=systolic_bp diastolic_bp ldl diabetes smoking sex)
```

The sub-macro %check_Cox_assumptions will produce diagnostics for the Cox assumptions for as many variables as are specified in the _CHECK list.

```
%if &_check =_ALL %then %do;
    %do all_count=1 %to %sysfunc(countw(&predictors_used));
       %let CHECK_VAR=%qscan(&predictors_used, &all_count,%str( ));
       %check_cox_assumptions (NUMBER=%UNQUOTE(%UPCASE(&_NUMBER)), Data=&_DATA,
          predictors=%UPCASE(&_PREDICTORS), class=%UNQUOTE(&_CLASS), OUTCOME=
          &_OUTCOME, FU=&_FU, STOP=%UPCASE(&_STOP), EXCLUDE=%UPCASE(&_EXCLUDE),
          CHECK=%UNQUOTE(%UPCASE(&CHECK_VAR)))
    %end;
%end;
%else %do;
    %do  all_count=1 %to %sysfunc(countw(&_CHECK));
       %let CHECK_VAR=%qscan(&_CHECK, &all_count,%str( ));
       %check_cox_assumptions (NUMBER=%UNQUOTE(%UPCASE(&_NUMBER)), Data=&_DATA,
          predictors=%UPCASE(&_PREDICTORS), class=%UNQUOTE(&_CLASS), OUTCOME=
          &_OUTCOME, FU=&_FU,STOP=%UPCASE(&_STOP), EXCLUDE=%UPCASE(&_EXCLUDE),
          CHECK=%UNQUOTE(%UPCASE(&CHECK_VAR)))
    %end;
%end;
```

How does the macro choose which diagnostics to run for a specific variable?

Different diagnostics need to be run for continuous and categorical variables. The macro runs different diagnostics for those variables mentioned in the _CLASS parameter (categorical predictors) from those produced for continuous predictors. Variables which are specified in a _CLASS parameter cannot be used directly for two reasons. First, the user can specify reference categories in the class parameter. Second, not all variables specified in class parameter need to be included in the model. If a variable is in the class parameter but not in the model statement it will influence the number of observations used (reducing it by the number of observations for which data on that categorical variable are missing); but it wouldn't change the model structure.

So I used ODS OUTPUT to produce the ClassLevelInfo data set and determine which variables should be treated as categorical.

```
ods output ClassLevelInfo=ClassLevelInfo_&Number;
    proc phreg data=&data;
           where &FU>0;
            class &class;
           model &FU*&outcome (0)= &predictors_used /ties=exact rl;
    run;
ods output close;

proc sql;
    select distinct class into :class_variables separated by ' '
    from ClassLevelInfo_&Number;
quit;

%if &CHECK in &class_variables %then %do;
```

3. WHAT PLOTS AND TABLES ARE INCLUDED IN THE OUTPUT RTF FILE?

The macro produces a list of plots and tables necessary for evaluation of the Cox model as well as the possibility of the presence of confounders or interaction. In developing this macro, I used as a guide the steps outlined in the SAS® course “Survival Analysis Using the Proportional Hazards Model” notes developed by Mike Patetta.

For each continuous predictor the ***check_Cox_assumptions_all*** macro provides four graphs and two tables:

- A plot of martingale residuals plotted against the variable being checked from a model that excludes that variable to check the assumption of linearity in the log hazard
- A plot of parameter estimates for dummy variables designating specific intervals of the continuous variable and fitted instead of that continuous variable to check the assumption of linearity in the log hazard
- Log-negative-log survival curves using the same dummy variables to check proportional hazard assumption
- The Schoenfeld residual plot to check proportional hazard assumption
- Tables to check for confounding and interaction with other variables

For each categorical predictor the ***check_Cox_assumptions_all*** macro provides at least two graphs and two tables:

- Log-negative-log survival curves and Schoenfeld residuals plot to check proportional hazard assumption
- Tables to check for confounding and interaction with other variables

Examples of each output are presented on the next several pages in the order shown below:

- Figure 1. Plot to assess linearity assumption of continuous predictor AGE using Martingale Residuals
- Figure 2. Plot to assess linearity assumption of continuous predictor AGE using Parameter Estimates for Dummy variables
- Figure 3. Plot to Assess the proportional hazard assumption of continuous predictor AGE using log-negative-log survival curves for dummy variables
- Figure 4. Plot to assess the proportional hazard assumption of continuous predictor AGE using Schoenfeld Residuals
- Figure 5. Plot to Assess the proportional hazard assumption of categorical predictor ACTIVITY using log-negative-log survival curves
- Figure 6. Plot to assess the proportional hazard assumption of categorical predictor ACTIVITY using Schoenfeld ResidualsTable

- Table 3. Results of Type 3 test for interactions
- Table 4. Results of Type 3 test for interactions
- Table 5. Check for confounding

Graphs

Martingale residuals are used to assess the assumption of linearity in the log hazard for a given variable by plotting the residuals from a model which excludes that variable (y axis) against the variable (x axis). A LOESS-smoothed line is fit to the plot – the shape of the smoothed line provides an estimate of the functional form of the predictor variable in the model. If the smoothed line is fairly linear, then the chosen scale is correctly linear in the log hazard. If the smoothed line departs substantially from a linear trend, then its form can be used to decide how to scale more correctly the predictor variable in the model.

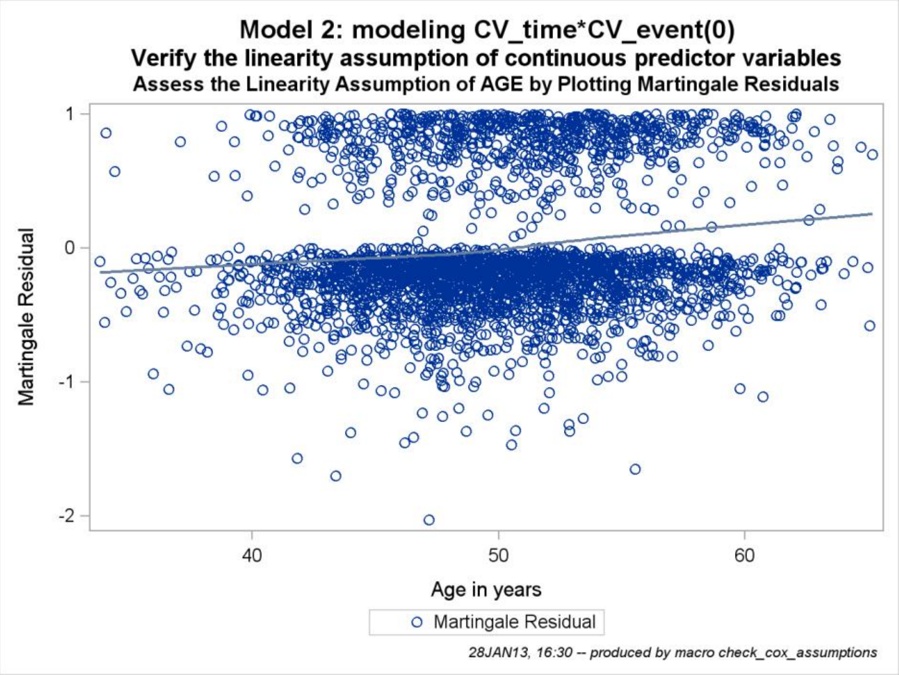

Figure 1. Plot to assess linearity assumption of continuous predictor AGE using Martingale Residuals

One method to check for linearity is to replace the predictor variable of interest with several dummy variables. Then fit the model with the new dummy variables and plot the parameter estimates of the dummy variables, with a point at zero representing the reference group (Figure 2). If the smoothed line is fairly linear, then the chosen scale is correctly linear in the log hazard. If the smoothed line departs substantially from a linear trend, then its form can be used to decide how to scale more correctly the predictor variable in the model.

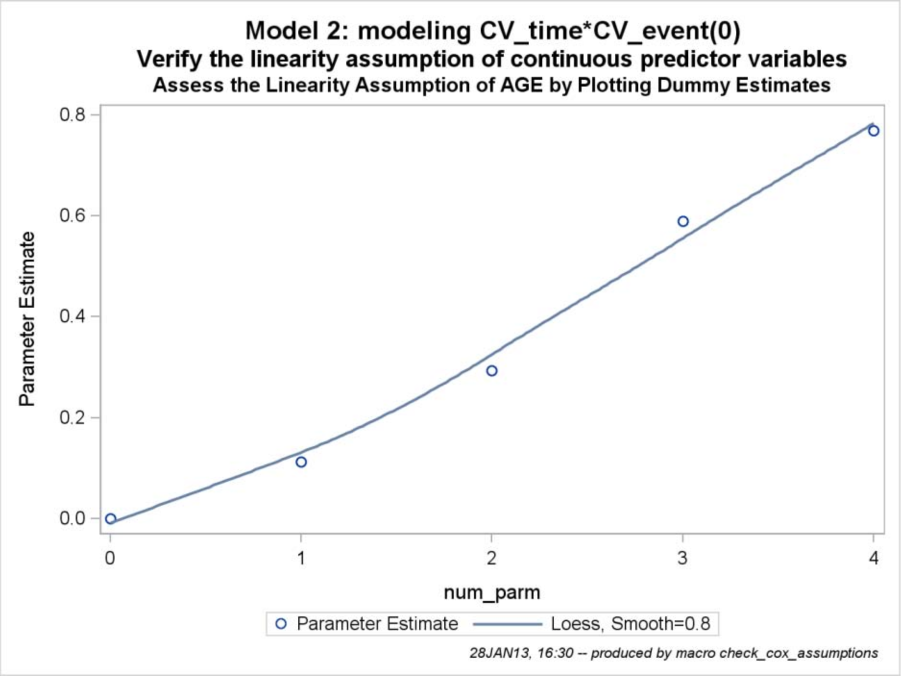

Figure 2. Plot to assess linearity assumption of continuous predictor AGE using Parameter Estimates for Dummy variables

One standard method of checking the proportional hazards assumption is to plot the log-negative-log of the Kaplan-Meier estimates of the survival function versus the log of time (Figure 3). If the plot has reasonably parallel lines there is not a significant problem with the assumption for the model.

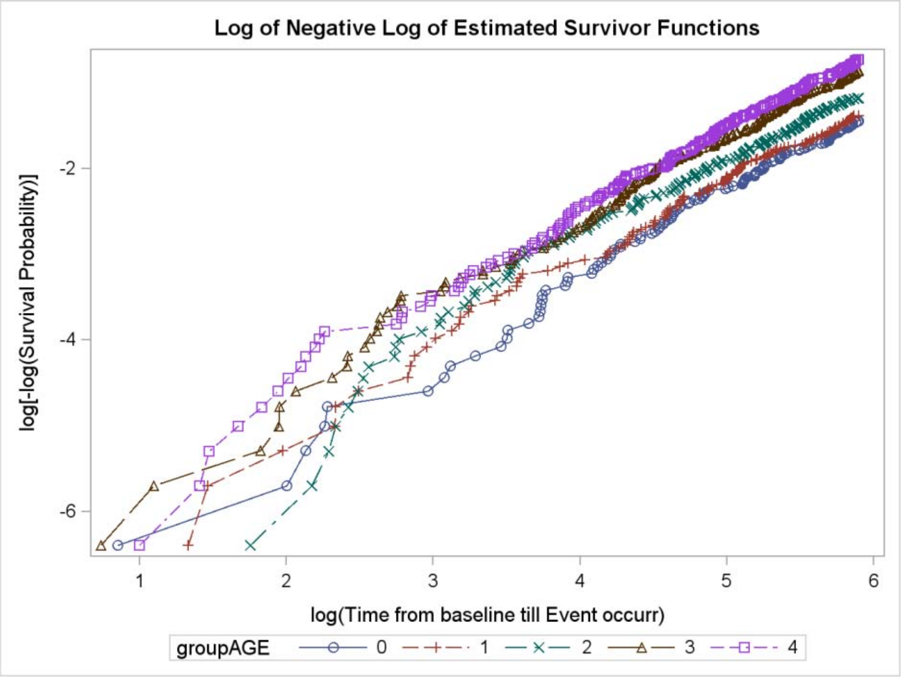

Figure 3. Plot to Assess the proportional hazard assumption of continuous predictor AGE using log-negative-log survival curves for dummy variables

Since Schoenfeld residuals are based on the effects of the predictor variables that are assumed to be independent of time, a plot of these residuals versus time can be used to visually assess whether the effect of the predictor variable changes over the period of follow-up (Figure 4). A smoothed line fit to the plot of the residuals should have an intercept and a slope of approximately 0 (it should be close to horizontal).

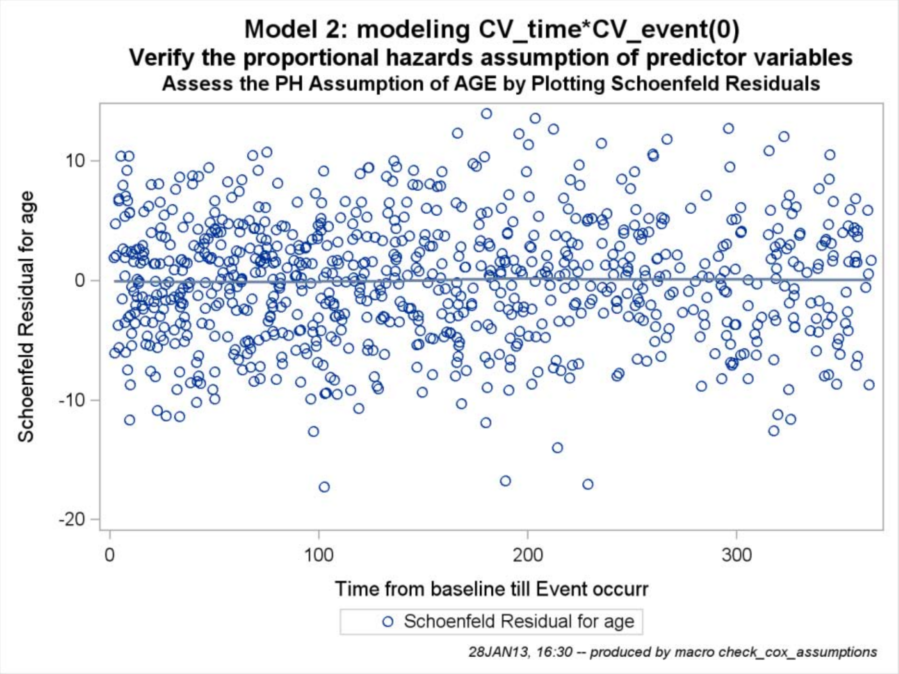

Figure 4. Plot to assess the proportional hazard assumption of continuous predictor AGE using Schoenfeld Residuals

Another method for assessing the proportional hazards assumption is to plot the log-negative-log of the Kaplan-Meier estimates of the survival function versus the log of time (Figure 5). If the plot has parallel lines, the hazard is reasonably constant over time.

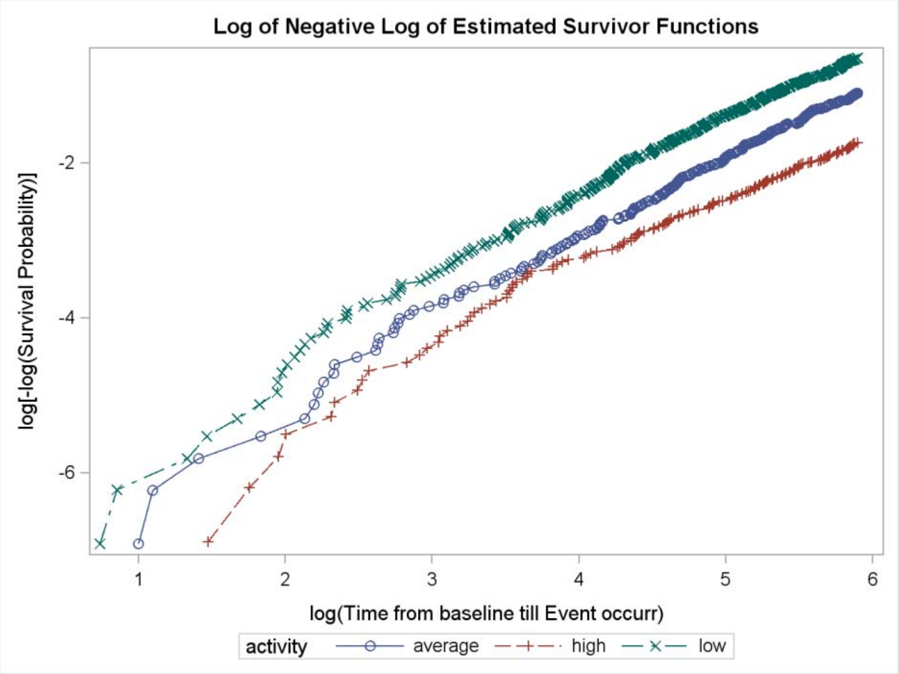

Figure 5. Plot to Assess the proportional hazard assumption of categorical predictor ACTIVITY using log-negative-log survival curves

Figure 6 shows a plot to assess the proportional hazard assumption of categorical predictor ACTIVITY using Schoenfeld Residuals. The objective is to generate plots to assess whether the hazard ratio for each other level of class variable vs the reference level is constant over time. This means that for a class variable with 3 levels, two plots are needed. Unfortunately, PROC PHREG produces a separate column for each continuous variable and for each level of a categorical variable and assigns an often-non-descriptive label to each column. This made it tricky and difficult to automatically produce graphs for the Schoenfeld residuals. To compensate for PHREG's problematic output, I had to produce a list of names which would correspond to predictors but wouldn't be the same names since those names are already used in the data set. This resulted in the display of a different number of levels, generally the number of levels without the reference category, e.g. for variables which have three levels there should be two names in the list.

```
/* data step to delete reference categories*/

       data ClassLevelInfo_&NUMBER;
       length class $30 value$30;
       set ClassLevelInfo_&NUMBER;
              class=UPCASE(CLASS);
              if ^missing(class) then call symput ('new',class);
              else class=symget('new');
              if sum(of X:)=0 then delete;
       run;
       proc sql;
              create table schoenfeld
              as select predictor, order, value,
              compress(catx('',predictor,'_',left(value))," -$%^()&#@","c") as
              schoenfeld
              from  Predictors_&NUMBER left join
              ClassLevelInfo_&NUMBER on left(class)=left(predictor)
              order by order, schoenfeld;
       quit;
       proc sql;
              select schoenfeld into :schoenfeld separated by ' '
              from schoenfeld;
       quit;

%put schoenfeld = &schoenfeld;

/*proportional hazards assumption of categorical and continuous predictor variables */

proc phreg data=&data noprint;
       where &FU>0;
       class &class;
       model &FU*&outcome (0)=  &predictors_used /ties=exact rl;
       output out=ressch_&CHECK ressch=&schoenfeld;
run;
ods rtf select all;
/* Producing as many plots as categories in check variable */
       %let cntsch=0;
       %do  i=1 %to %sysfunc(countw(&schoenfeld));
        %let category=%qscan(&schoenfeld, &i,%str( ));
%if %upcase(%substr(&category,1, %sysfunc(length(&CHECK))))=%upcase(&CHECK) %then %do;
       proc sgplot data=ressch_&CHECK;
              scatter y=&category x=&fu/ name="Schoenfeld_&CHECK";
              loess y=&category x=&fu/SMOOTH=.8 DEGREE=1 INTERPOLATION= CUBIC ;
              keylegend "Schoenfeld_&CHECK"/ title=' ' Position=bottom;
       run;
```

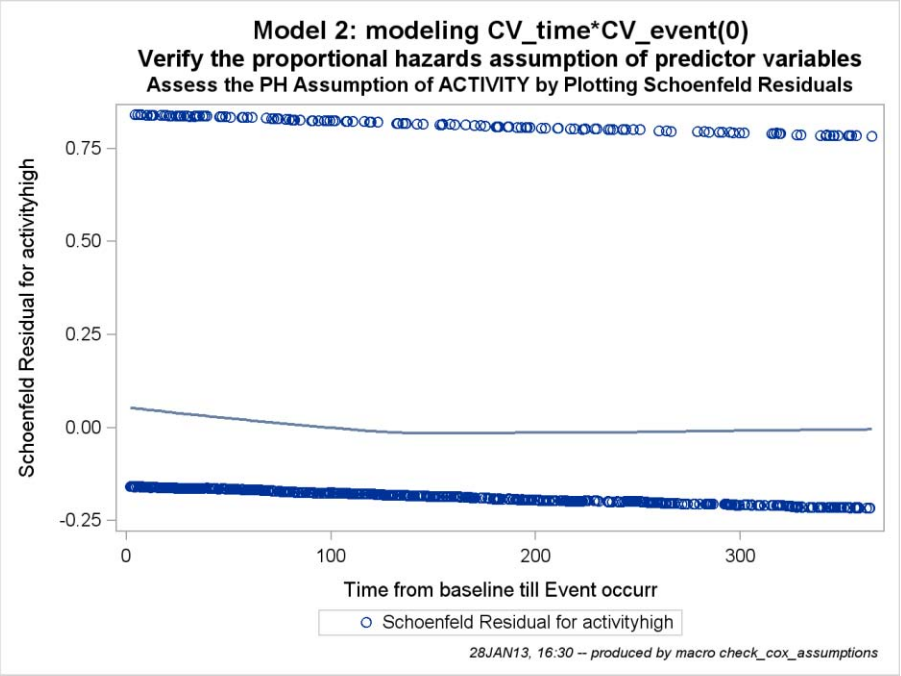

Figure 6. Plot to assess the proportional hazard assumption of categorical predictor ACTIVITY using Schoenfeld Residuals

Tables

The macro produces two tables for each variable being checked.

Table 3 displays the results of a partial likelihood ratio test that compares the model with only the main effects to the model with the main effects and all two-way interactions. P-values less than 0.05 are marked in the ***flag*** column. Table 3 shows interaction between check variable and all other variables in the model.

parameter	Label	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	flag
age*treatment	Treatment 1 * Age in years	-0.01256	0.01385	0.8226	0.3644	
age*activity	Physical Activity alevel high	-0.02063	0.02005	1.0584	0.3036	
age*activity	Physical Activity alevel low	-0.02540	0.01564	2.6371	0.1044	

Table 3. Results of Type 3 test for interactions without specifying _INTERACTIONS parameter

If _INTERACTIONS parameter is specified then table of interactions shows up only once at the end of the output and only requested interactions are estimated. For example, calling the following model %check_cox_assumptions_all (_Number=4, _Data=rq.simcox, _predictors=&predictors, _class=&class, _OUTCOME=CV_event, _FU=CV_time, _CHECK=AGE, _INTERACTIONS=age*Systolic_bp systolic_bp*diastolic_bp) will lead to producing Table 4.

Parameter	Label	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	flag
age*systolic_bp	Age in years * Systolic Blood Pressure in mmHg	0.0009	0.0007	1.7002	0.1923	
systolic_*diast	Systolic Blood Pressure in mmHg * Diastolic Bl	-0.0002	0.0004	0.2599	0.6102	

Table 4. Results of Type 3 test for interactions with specified _INTERACTIONS parameter

The second table in the macro output (Table 5) displays estimated change between the parameter estimates and standard errors for the model without the variable of interest (crude) and for the model with the variable of interest (adjusted). Variables for which parameter estimates vary by more than 20% are marked in the flag column. Since small changes in the very small parameter estimates can cause a huge percentage difference flag is only up if one of parameter estimates is greater than 0.01.

Parameter	Label	Crude Estimate (Error)	Crude Hazard Ratio (95% CI)	Adjusted Estimate (Error)	Adjusted Hazard Ratio (95% CI)	change	flag
activity	Physical Activity alevel high	-0.63 (0.10)	0.53 (0.43- 0.65)	-0.64 (0.10)	0.53 (0.43- 0.64)	(1%)	
activity	Physical Activity alevel low	0.46 (0.08)	1.59 (1.36- 1.86)	0.46 (0.08)	1.58 (1.35- 1.85)	1%	
treatment	Treatment 1	0.24 (0.07)	1.27 (1.11- 1.46)	0.25 (0.07)	1.28 (1.12- 1.47)	(4%)	

Table 5. Check for confounding

CONCLUSION

This macro can help to save hours of work for a programmer performing survival analysis. Also this macro can save time for a biostatistician writing a request. It is so much easier to say ‘Run the PH assumptions macro’ than to write out everything that’s needed. This time savings can be multiplied with several clinical studies. It can minimize errors which can be made specifying models. Also this macro can be helpful if slight changes need to be made for already existing jobs.

REFERENCES

Survival Analysis Using the Proportional Hazards Model Course Notes developed by Mike Patetta

ACKNOWLEDGMENTS

Thanks to my colleagues at Collaborative Studies Coordinating Center: Kathy Roggenkamp, Myra Carpenter, Stephen Campbell and Kwanhye Jung

This work was supported by the Folic Acid for Vascular Outcome Reduction in Transplantation (FAVORIT) trial cooperative agreement U01 DK61700 from the National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health. The content is solely the responsibility of the author and does not necessarily represent the official views or endorsement of the National Institutes of Health

RECOMMENDED READING

Survival Analysis Using SAS® A Practical Guide

Carpenter’s Complete Guide to the SAS® Macro Language

CONTACT INFORMATION

Your comments and questions are valued and encouraged. For a copy of the macro contact the author at:

Name: Polina Kukhareva
Enterprise: Collaborative Studies Coordinating Center, University of North Carolina at Chapel Hill
Address: 137 E. Franklin Street, Suite 203, CB# 8030
City, State ZIP: Chapel Hill, NC 27514-4145
Work Phone: (919) 962-6971
Fax: (919) 962-3265
E-mail: p.kukhareva@alumni.unc.edu
Web: www.linkedin.com/pub/polina-kukhareva/29/42a/7a1