Survival Analysis Using SAS® Macros, ODS and SAS/GRAPH

Natalia Udaltsova
Kaiser Permanente
64 PUBLICATIONS 2,820 CITATIONS
SEE PROFILE

Irina Tolstykh
University of California, San Francisco
52 PUBLICATIONS 1,023 CITATIONS
SEE PROFILE

Arthur Klatsky
Kaiser Permanente
209 PUBLICATIONS 10,918 CITATIONS
SEE PROFILE

Survival Analysis Using SAS® Macros, ODS and SAS/GRAPH®

Natalia V Udaltsova, Division of Research, Kaiser Permanente, Oakland, CA
Irina V Tolstykh, Division of Research, Kaiser Permanente, Oakland, CA
Arthur L Klatsky, M.D., Division of Research, Kaiser Permanente, Oakland, CA

ABSTRACT

Survival analysis is a popular method in medical research. Preliminary analysis and visualization of survival curves are important to make adequate assumptions about the statistical model.

This paper shows various approaches that produce Kaplan–Meier curves from PROC LIFETEST or from the baseline statement in PROC PHREG. We developed macro code to plot survival curves with confidence intervals for selected points by strata. The number of subgroups in strata and corresponding SAS graph options are calculated and assigned by design. The paper also presents macro for adjusted survival curves.

Powerful tool that performs multivariate analysis is a PROC PHREG. We analyzed data with time dependent variables and repeated measurements. Two types of data structures were considered: “horizontal” with a unique observation per ID and multiple time-points, and “vertical” with multiple observations per ID. The paper demonstrates macros with nested Cox proportional hazard models for both types of these data structure.

INTRODUCTION

Prior to 1970, the estimation of survivor functions was the predominant method of survival analysis. Nowadays, the workhorse of survival analysis is the Cox regression method. Nevertheless, survival curves are still useful for preliminary examination of the data and for evaluating the fit of regression models (Allison, 1995).

PART 1. KAPLAN-MEIER SURVIVOR FUNCTIONS

We will be considering the Kaplan-Meier method of estimates of survivor functions produced by PROC LIFETEST. The Kaplan-Meier method is most suitable for smaller data sets with precisely measured event times.

In biomedicine, the Kaplan-Meier estimator is the most widely used method for estimating survivor functions. Also known as the “product-limit estimator”, this method is, in fact, the nonparametric maximum likelihood estimator.

Here is the example of how to get the Kaplan-Meier estimator using PROC LIFETEST with some test data of clinical nature:

```
proc lifetest data=test method=km plots=(s) cs=none;
 time years*deaths(0);
 strata AgeDX;
 run;
```

“METHOD=KM” in PROC LIFETEST statement requests Kaplan-Meier estimator, though you can omit it, because it’s the default.

“PLOTS=(S)” requests a plot of estimated survivor function, and “CS=NONE” specifies that the symbol value for the censored observations is to be omitted. (The default is CS=CIRCLE).

The “YEARS” variable is the time of the event or censoring; the “deaths” variable contains information on whether or not the observation was censored; and the number (or numbers) in parentheses are values of the “deaths” variable that correspond to the censored observations.

A statement “STRATA **AgeDX**” requests that your data to be divided into various strata (the variable **AgeDX** is categorical) and a separate survivor function to be estimated for each stratum.

These statements produce graph shown at Figure 1.

The investigator was not satisfied with this graph and requested to make these curves with 95% confidence limits as vertical bars at certain points (each year) to see if survivor functions are significantly different by strata. It was done by performing additional calculations using the results produced by PROC LIFETEST. The authors developed set of Macros to generate plots for any number of strata.

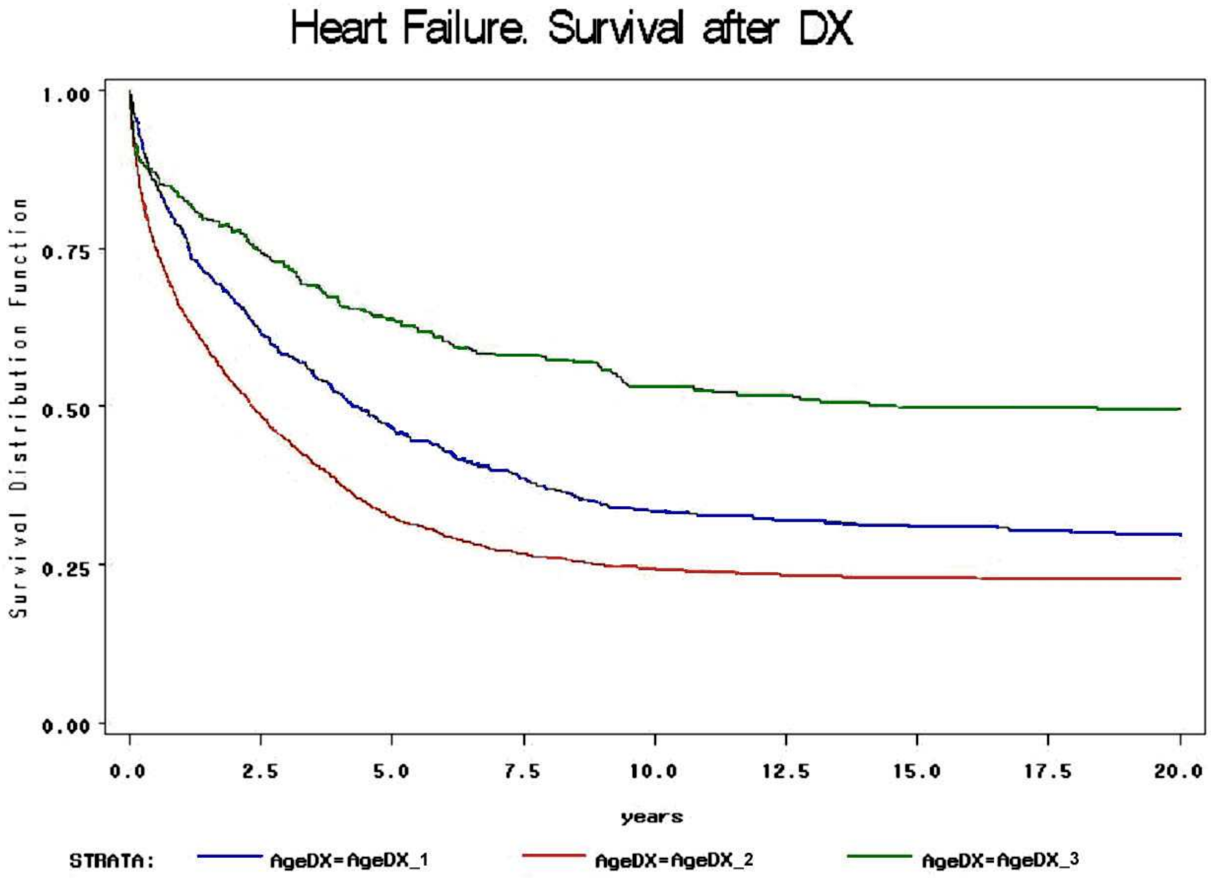

Figure 1.

1) Macro ***%SURV*** calls PROC LIFETEST, and also runs macro ***%CLIM*** to calculate confidence limits, as well as macro ***%GOPT*** to assign colors to strata lines and bars, and makes a plot using PROC GPLOT. Macro variable "***&in***" is the name of input dataset, "***&var***" is strata variable. Plot is embedded into RTF file created by ODS. The name of RTF file includes both macro variables: "***&in.&var..rtf***". The file is created in the directory "***&folder***" (the directory must pre-exist).

```
%macro SURV(in,var);
data sum; set _null_; run;
ods select none;
ods output CensoredSummary=sum;
proc LIFETEST data=&in method=km outs=tmp0 ;
   time years*deaths(0);
   strata &var  ;
  run;
ods select all;
    ods rtf file="&folder.\&in.&var..rtf" ;
%CLIM(tmp0,tmpl);
%let collist=%str(red blue green magenta cyan black);
%GOPT(&var);
proc GPLOT data=tmp1 ;
plot survival*years=&var / vaxis=axis1 haxis=axis2  frame ;
plot2 cl*years=Conf_Limits / vaxis=axis1 haxis=axis2  frame ;
format years 3.0 survival 3.1; quit;
    ods rtf close;
%mend;
```

2) Macro ***%CLIM*** calculates confidence limits at discrete points and combines them with curves data.

```
%macro CLIM(tmp,out);
*** to find ~integer(year) points for 95% CL ***;
proc sort data=&tmp; by &var years; run;
data CI; set &tmp(where=(_censor_=0 ));
 by &var years; retain ucl lcl .  ;
 Conf_Limits=&var; k=floor(years);
 dif=abs(years-k); if dif<=0.3 then do;
 lcl=sdf_lcl; ucl=sdf_ucl; output; end; run;
proc sort data=CI; by stratum k dif; run;
data CI1; set CI; by stratum k  dif; if first.k then output; run;
proc sort data=CI1; by &var years; run;
proc sort data=&in; by &var years; run;
data &out; set &in CI1; by &var years; where _censor_=0 ;
retain cl .; Conf_Limits=&var; if ucl=. and lcl=. then do; cl=.; output; end;
if ucl>. or lcl>. then do;
     cl=sdf_ucl;  years=k; output;
     cl=sdf_lcl;  years=k; output; end; run;
proc sort data=&out; by &var years; run;
%mend;
```

3) Macro ***%GOPT*** defines list of strata, assigns colors to curves and bars in graphical options.

```
%macro GOPT(var);
*** to create a list of strata subgroup names ***;
proc sql noprint;
  select distinct &var into :list separated by " "
  from &in where &var ne ' ' order by &var;
quit;
*** to count number of strata ***;
%LET NUM=1;
%LET WORD=%QSCAN(&list,&NUM,%STR( ));
%DO %WHILE(&WORD NE);
  %LET NUM=%EVAL(&NUM+1);
  %LET WORD=%QSCAN(&list,&NUM,%STR( ));
%END;
%LET NUM=%EVAL(&NUM-1);
goptions RESET = goptions;
goptions reset=global ;
*** to assign colors accordingly strata names ***;
%DO COUNT=1 %to &NUM;
  %LET WORD=%QSCAN(&collist,&COUNT,%STR( ));
  %let color=%unquote(&word);
  %let K=%eval(&NUM+&count);
    symbol&count color=&color interpol=join value=point line=&count width=2;
    symbol&K color=&color interpol=hilot width=1;
%END;
 axis1 label=(f='helvetica' h=4pct a=90 "Kaplan-Meier Estimates")
      offset=(3) value=(h=3pct) order=(0. to 1. by 0.1) minor=none;
 axis2 label=(justify=center f='helvetica' h=4pct "Years of follow-up after DX" )
      offset=(3) value=(h=3pct) order=(0 to 10 by 1) minor=none;
%mend;
```

The plot produced with these macros is presented at the Figure 2.

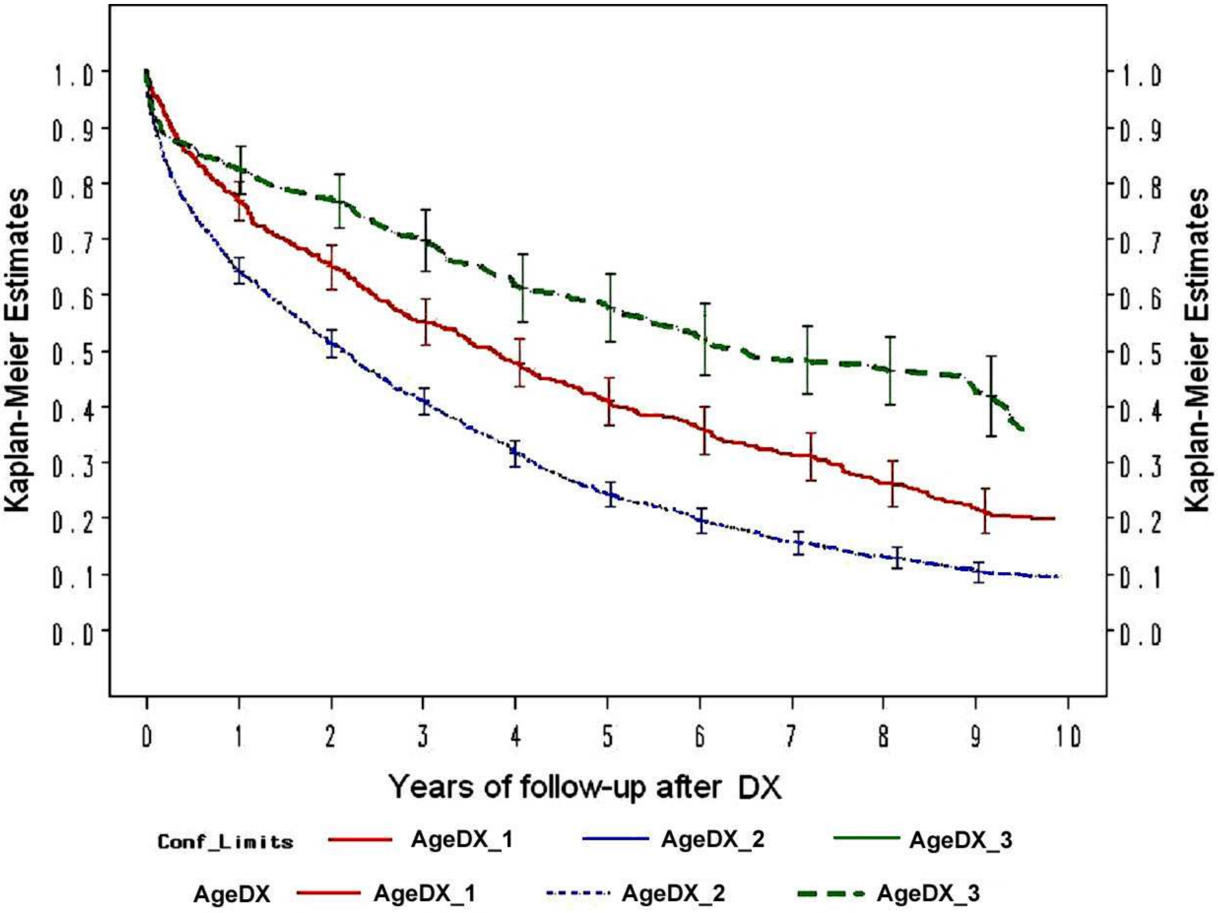

Figure 2.

The most popular and powerful method of survival analysis is Cox regression model implemented in PROC PHREG. Using PROC PHREG we can produce the same survivor function estimates. The only problem we met that PROC PHREG has the lack of built-in graphics. So we used PROC PHREG to get survivor function estimates and PROC GPLOT to plot the curves (see Figure 3. You can compare it with Figure 1 and see that they represent the same curves).

Here is the simple code to do this. "***&covars***" is a list of covariates. If the list is used with no option "COVARIATES=" in "BASELINE" statement, it does not effect calculation of survivor function estimates, which are not adjusted by default.

```
proc PHREG data=test;
  model years*deaths(0) = &covars;
  baseline out=tmp0 survival=survival;
  strata &desc;
run;
  axis1 label=(f='helvetica' h=4pct a=90 "Kaplan-Meier Estimates")
    offset=(3) value=(h=3pct) order=(0. to 1. by 0.1) minor=none;
  axis2 label=(justify=center f='helvetica' h=4pct "Years of follow-up after DX" )
    offset=(3) value=(h=3pct) order=(0 to 20 by 2) minor=none;
proc GPLOT data=tmp0 ;
  plot survival*years=&desc / vaxis=axis1 haxis=axis2  frame ;
  format years 3.0 survival 3.1;
run;
```

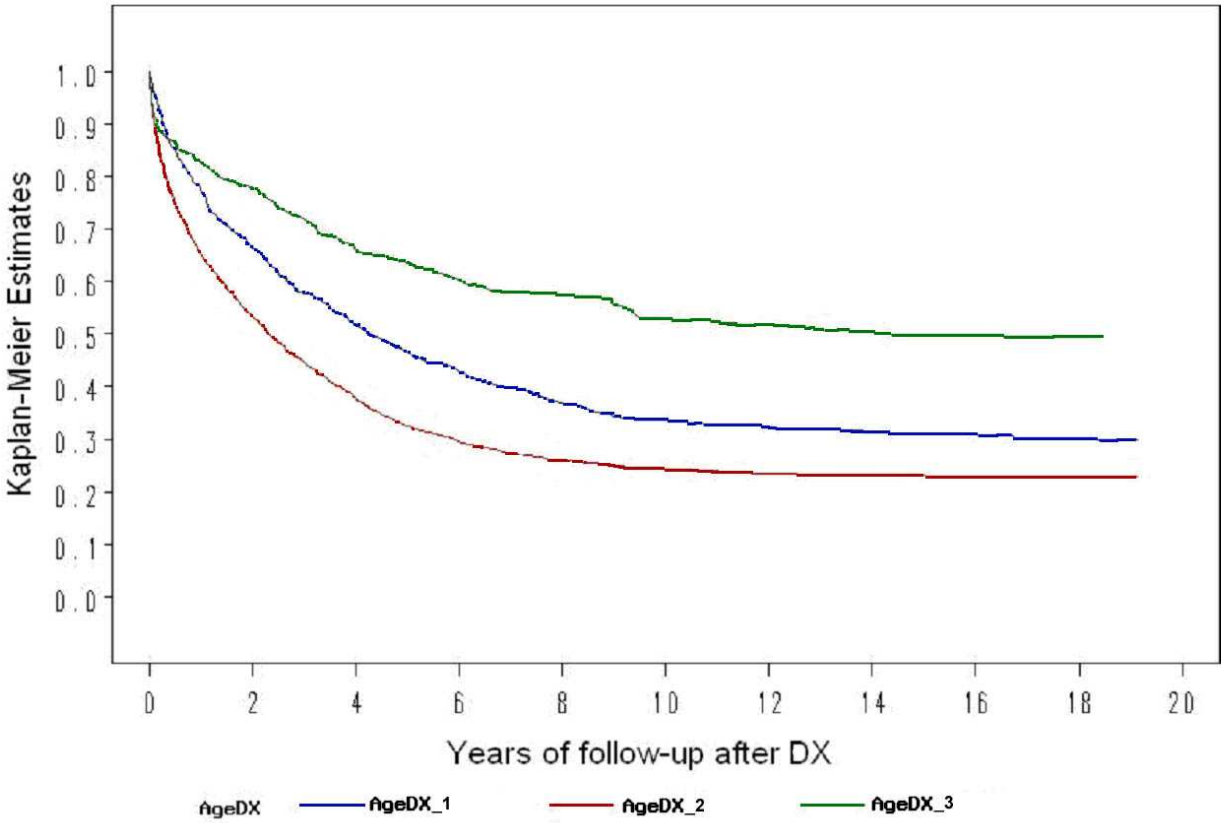

Figure 3.

The PROC PHREG may be used instead of PROC LIFETEST in presented above macro ***%SURV*** to generate plot of survival curves with 95% confidence limits.

PART 2. COX REGRESSION WITH TIME-DEPENDENT COVARIATES

Cox regression method implemented by PROC PHREG does not require that you choose some particular probability distribution to represent survival times. Cox regression makes it relatively easy to incorporate time dependent covariates, that is, covariates that may change in value over the course of the observation period (Allison, 1995).

PROC PHREG has two different ways of setting up models with time-dependent covariates (Allison, 2003).

1) Programming statements requires “horizontal” dataset structure with one record for each individual, where multiple covariate values are separate variables on data record. Time dependent covariates are defined in programming statements after the MODEL statement.

2) Counting process syntax uses “vertical” dataset structure: multiple records per individual with one record for each interval of time in which covariates are constant. Time-dependent variables here treated just like other variables. This approach is also known as “episode splitting”.

You should choose the setting and the dataset structure that fit your project requirements and type of your data. Some features/options of PROC PHREG cannot be used or have limitations when using in programming statements setting; for example, residuals or “dfbeta” variables calculation (Hosmer, 1999).

We will give the examples of macros for both types of model settings aiming to get the same kind of report.

In our actual assignment the investigator required running set of nested Cox models, sequentially adding more and more covariates to main list of explanatory variables. This set of models looked like the following:

Model 1	
On drug	(reference: off drug) – time dependent variable
Model 2	
On drug	(ref: off drug) – time dependent variable
Eligible	(ref: not eligible) – time dependent variable
Model 3	
On drug	(ref: off drug) – time dependent variable
Eligible	(ref: not eligible) – time dependent variable
Age	(numeric value) – time dependent variable
Sex	(Female, ref: male)
Model 4	
On drug	(ref: off drug) – time dependent variable
Eligible	(ref: not eligible) – time dependent variable
Age	(numeric value) – time dependent variable
Sex	(Female, ref: male)
Race	(Black; Hispanic; Asian; Other/Unknown, White as ref)
Model 5	
On drug	(ref: off drug) – time dependent variable
Eligible	(ref: not eligible) – time dependent variable
Age	(numeric value) – time dependent variable
Sex	(Female, ref: Male)
Race	(Black; Hispanic; Asian; Other/Unknown, White as ref)
Risk factors	(by list, “NO” as ref) – time dependent variables.

The report after regression analysis has to have the following structure and is to be printed as a table in Word compatible document (RTF file):

title	Variable	Hazard Ratio	Lower 95% CL	Upper 95% CL	P-value
Outcome (name)					
1. On drug (Not Adjusted)					
	ondrug				
2. #1 + eligib.score					
	ondrug				
	eligib				
3. #2 + age & sex					
	ondrug				
	eligib				
	age				
	male				
4. #3 + race/ethnicity					
	ondrug				
	eligib				
	age				
	male				
	black				
	hispanic				
	asian				
	other				
5. #4 +Risk Factors					
	ondrug				
	eligib				
	age				
	male				
	black				
	hispanic				
	asian				
	other				
	STRt				
	HTNt				
	CHFt				
	DBMt				
	CADt				

Here we present macros to run nested Cox models for each type of dataset structure and get the report table.

1) Macro ***%LINE*** is common for both types of data settings. It appends a report line with “***&title***” text to the report.

```
%macro LINE(title);
data line;
length title $25. ;
title=&title; run;
data report; set report line; run;
%mend;
```

2) Macros for Programming Statements model setting (“horizontal” data structure).

2.1) Macro ***%COXH*** prepares working dataset, calls macros for nested Cox models, and prints the report into RTF file. Variable “***&event***” is the date of the event. “***&work***” is working dataset with “horizontal” structure, one record per subject. “***&n***” is number of time-points for each of time-dependent variables.

```
%macro COXH(event,evtext,work,n);
%line(&evtext);
data work1; set &work;
   if stop>=&event>start then do; outcome=1;
   time=&event-start; end;
else do; outcome=0; time=stop-start; end; run;
%let racelist=black hispanic asian other;
%let RFXlist=STRt HTNt CHFt DBMt CADt;
%let text='1. Not Adjusted ';
%cox1(&n, noelig=*, noage=*, noRFX=*, elig=, age=, sex=, racelist=, RFXlist=);
%let text='2. #1 + eligib.score';
%cox1(&n, noelig=, noage=*, noRFX=*, elig=eligib, age=, sex=, racelist=, RFXlist=);
%let text='3. #2 + age & sex';
%cox1(&n, noelig=, noage=, noRFX=*, elig=eligib, age=age, sex=female, racelist=,
RFXlist=);
%let text='4. #3 + race/ethnicity’;
%cox1(&n, noelig=, noage=, noRFX=*, elig=eligib, age=age, sex=female,
racelist=&racelist, RFXlist=);
%let text='5. #4 + Risk Factors';
%cox1(&n, noelig=, noage=, noRFX=, elig=eligib, age=age, sex=female,
racelist=&racelist, FRXlist=&RFXlist);
ods rtf file="&folder.\&work..rtf" ;
 proc print data=report noobs label=split(*);
   title "Cox Regression Model with data in programming statements setting"
   var title variable hr lcl ucl p;
   label hr='Hazard Ratio' lcl='Lower 95% CL' ucl='Upper 95% CL' p='P-value';
 run;
ods rtf close;
%mend cox;
```

2.2) Macro ***%COX1*** is for any of nested Cox models: it runs Cox model and appends result lines to the “report” dataset.

```
%macro COX1(n, noelig=*, noage=*, noRFX=*, elig=, age=, sex=, racelist=, FRXlist=);
%line(&text);
ods listing close;
ODS output ParameterEstimates=res;
proc phreg data=work1;
  model time*outcome(0)= ondrug &elig &age &sex &racelist &FRXlist / ties=exact
risklimits alpha=0.05;
    time1=0;
    %let n1=%eval(&n+1);
        array t{*} time1-time&n1;
&noelig array el{*} elig1-elig&n;
        array dr{*} drug1-drug&n;
  do j=1 to dim(dr);
```

```
 if (t{j}<=time<t{j+1} or (t{j}=time and t{j}>. and t{j+1}=.))
     then do;
        if dr{j}=1 then ondrug=1; else ondrug=0;
&noelig eligib=el{j};
&noage  age=int((start+t{j}-birth_dt)/365.25);
    end;   end;
&noRFX array date{*} date_STR date_HTN date_CHF date_DBM date_CAD;
&noRFX array RFX{*} STRt HTNt CHFt DBMt CADt;
&noRFX    do k=1 to dim(RFX)
&noRFX  if (.<(date{k}-start)<=time) then RFX{k}=1; else RFX{k}=0;
&noRFX  end;
run;
ods listing;
data results; set res;
rename HazardRatio=HR ProbChiSq=P HRlowerCL=LCL HRUpperCL=UCL;
keep Variable HazardRatio ProbChiSq HRlowerCL HRUpperCL;
data report; set report results; run;
%mend;
```

3) Macros for Counting Process Syntax (“vertical” dataset structure)

3.1) Macro ***%COXV*** calls macros for nested Cox models and prints the report into RTF file.

```
%macro COXV(evtext,work);
%line(&evtext);
ods listing close;
%let model="1. Not Adjusted ";
%cox2(&model,&work,ondrug);
%let model="2. #1 + eligib.score ";
%cox2(&model,&work,ondrug eligib);
%let model="3. #2 + age & sex";
%cox2(&model,&work, ondrug eligib age female );
%let model="4. #3 + ethnicity";
%let ethnlist= black hispanic asian other;
%cox2(&model,&work, ondrug eligib age female &ethnlist);
%let model="5. #4 +Risk Factors ";
%let RFXlist= STRt HTNt CHFt DBMt CADt;
%cox2(&model,&work, ondrug eligib age female &ethnlist &RFXlist);
ODS listing;
ods rtf file="&folder.\&work..rtf" ;
 proc print data=report noobs label=split(*);
   title "Cox Regression Model with data in counting process syntax setting"
   var title variable hr lcl ucl p;
   label hr='Hazard Ratio' lcl='Lower 95% CL' ucl='Upper 95% CL' p='P-value';
 run;
ods rtf close;
%mend cox;
```

3.2) Macro ***%COX2*** executes any of nested models depending on the list of covariates ***&list.***

```
%macro COX2(model,work,list);
%line(&model);
ODS output ParameterEstimates=res ;
proc phreg data=&work covs(aggregate) multipass;
  model time*outcome(0)= &list /  ties=exact risklimits alpha=0.05;
  id study_id; run;
data results; set res;
  rename HazardRatio=HR ProbChiSq=P HRlowerCL=LCL HRUpperCL=UCL;
  keep Variable HazardRatio ProbChiSq HRlowerCL HRUpperCL ;
data report; set report results; run;
%mend;
```

CONCLUSION

It’s not surprising that Cox regression has become the overwhelmingly favored method for doing regression analysis of survival data. It makes no assumptions about the shape of the distribution of survival times; it allows for time-dependent covariates; it is appropriate for both discrete-time and continuous-time data (Allison, 1995). The lack of built-in graphic features in PROC PHREG can be easily compensated by using PROC GPLOT after calculation with PROC PHREG.

At the first part of this presentation the authors demonstrated macros with tips and tricks working with SAS graphics. At the second part the authors used such essential feature of PROC PHREG as its extremely general ability to handle time-dependent covariates, and presented macros to implement nested Cox models in two different model settings.

REFERENCES

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

Allison, Paul D., “Survival Analysis, Course Notes”, 2003.

Hosmer DW, Lemeshow S, “Applied Survival Analysis”, New York, John Wiley & Sons, 1999.

SAS Institute Inc. (1999), SAS/STAT® User’s Guide, Version 8, Cary, NC: SAS

ACKNOWLEDGMENTS

Authors want to thank Drs. Alan Go, Carlos Iribarren, Mark Eisner, and Gary Friedman for continuous supervision of projects for which presented methods and macros have been developed.

CONTACT INFORMATION

Natalia V. Udaltsova
Division of Research, Kaiser Permanente
2000 Broadway,
Oakland, CA 94612
(510) 891-3738
E-mail: Natalia.Udaltsova@kp.org

Irina V. Tolstykh
Division of Research, Kaiser Permanente
2000 Broadway,
Oakland, CA 94612
(510) 891-3535
E-mail: Irina.V.Tolstykh@kp.org

Arthur L. Klatsky, MD,
Division of Research, Kaiser Permanente
2000 Broadway,
Oakland, CA 94612
(510) 891-3241