PharmaSUG 2025 - Paper ST-343

Implementation of the Inverse Probability of Censoring Weighting (IPCW) Model in Oncology Trials

Mei Huang, Shibani Harite, Linsong (Athena) Zhang, Haijun Ma, Exelixis Inc.

ABSTRACT

Crossover is a common feature in oncology clinical trial designs that allows patients in the control arm to switch to the experimental arm, often upon disease progression. Crossover can bias the treatment effect estimate for long-term endpoints, such as overall survival. The inverse probability of censoring weighting (IPCW) model is a widely used technique to estimate the treatment effect on overall survival as if no crossover had occurred. Although the implementation of the IPCW model in SAS has been previously discussed (Mosier, 2023), applying the IPCW model remains challenging in practice and requires careful consideration.

This paper provides a step-by-step guide to implementing the IPCW model, explaining the method in layman's terms and highlighting technical issues and solutions. For instance, there is limited literature on preparing multiple time-varying covariates in a counting process format, which can be cumbersome. This paper introduces macros that can be readily used to generate counting process data with multiple time-varying covariates. And we modified the macro presented by Mosier (2023) to handle switching or use of new anti-cancer therapies in both arms. Additionally, the paper examines the processes and statistical challenges associated with implementing the IPCW model to offer guidance for practitioners.

INTRODUCTION OF IPCW METHOD, ASSUMPTIONS AND LIMITATIONS

Crossover is a clinical trial design feature in oncology where patients in the control arm of a study are allowed to switch to the experimental arm, often upon disease progression. This design is often used in trials where the control arm provides sub-standard treatment (e.g., placebo) and there is an ethical imperative to offer participants a potentially effective treatment option. In many cases, trial enrollment would not be possible without offering crossover. Even when crossover is not an explicit feature of the trial design, participants often receive additional post-progression anti-cancer therapies.

In oncology clinical trials, characterizing the long-term overall survival (OS) benefit or lack of detrimental effect for an experimental treatment vs control is a critical question. However, crossover can bias the estimated treatment differences for OS, especially when in practice the experimental treatment is not used for next-line of treatment. Various statistical methods have been proposed to adjust for the effect of crossover, with IPCW being a common approach (Robins and Finkelstein, 2000). IPCW aims to estimate the treatment effect as if no crossover had occurred by comparing the treatment arm with a pseudo-control arm that would have been observed without crossover.

IPCW method censors patients at the point of crossover, and upweights remaining patients in a manner designed to reflect the original population (defined based on a collection of covariates). The assumption is that the reweighted control arm resembles the original population in terms of performance.

IPCW method relies on several assumptions, which unfortunately may be hard to verify:

- All baseline or post-baseline characteristics that predict both treatment switching and outcome are available (i.e., "no unmeasured confounders").
- No confounding factor perfectly predicts switching.

IPCW could produce less bias than ITT population analysis in some scenarios, however, IPCW is not appropriate for studies with small sample sizes or when the percentage of crossover is either very low or very high.

STEPS TO FIT THE IPCW MODEL AND COMMON APPROACHES TO HANDLE SOME TECHNICAL DETAILS

Fitting an IPCW model includes the following steps. Without loss of generality, we assume a one-way crossover, i.e. that only control group patients are allowed to cross over to treatment group:

1. Identify important baseline and time-dependent covariates that predict both switching and outcome. The identification of covariates should be preplanned. Literature search and clinical judgment is necessary to identify the potential list of covariates. If variable selection is warranted, specify criteria for variable selection.

2. Prepare data where counting process type of dataset is created with time-varying covariates.

3. Fit a switching model to control group patients. Common choices of switching model include logistic regression with switching as the dependent variable, Cox-regression model with time to switch as the dependent variable, etc. Model selection might be needed at this step.

4. Calculate time-dependent weights for control group patients as the inverse probabilities of remaining unswitched. Stabilized weights are often used. To calculate the stabilized weight for a patient, the probability estimated by the switching model conditional on baseline and time-dependent covariates is the denominator of the weight and the probability estimated by the model with baseline covariates only (i.e., excluding time-dependent variables) is the numerator of the weight.

5. Treatment group patients receive a weight of 1.

6. Fit an outcome model comparing treatment group vs. the pseudo-control group created with IPCW. Any prognostic baseline covariates used in the switching model for the numerator of the weights should be included in the outcome model.

If crossover is bidirectional, i.e., both treatment and control group patients can crossover to the other group, then instead of assigning a weight of 1, the weights of treatment group patients are calculated using the same approach as described in step 3 and 4.

DESCRIPTION OF THE LOGIC HOW COUNTING PROCESS STYLE DATASET CAN BE CREATED WHEN THERE ARE MULTIPLE TIME-VARYING COVARIATES

In clinical trials, parameters may be measured at multiple timepoints post baseline. In the counting process approach, a new time interval must be created whenever a covariate changes. The final counting process data contains time intervals per patient in which all the covariates remain constant. Note that the time intervals (t1, t2] for PROC PHREG are semiclosed with t1 exclusive and t2 inclusive, so t1 of the first interval of each patient should be 0 and t1 of the subsequent intervals should be one day before covariates change. For IPCW analysis, data after cross-over time is removed for cross over patients, i.e., t2 of the last interval of these patients is the cross-over time. And in the last interval, these crossover patients have event for cross-over and are censored for the outcome analysis. Missing data should be checked carefully during data preparation. Missing data imputation method such as last observation carry forward may be considered.

In order to prepare counting process style dataset using the macros that we are presenting, the input dataset needs to be prepared in long format with one record per each visit. In Table 1, dummy data from three patients are used to illustrate the raw data. table 2 displays the final counting process data.

Table 1 Collected Raw Data

Patient	Crossover Time (days)	Overall survival (days)	Death	Baseline Ecog	Baseline Hgb (g/dL)	Time Point (days)	Ecog	Hgb (g/dL)
1	90	120	No	0	8	40	1	7
1	90	120	No	0	8	80	2	7
2		160	No	0	10	40	0	10
2		160	No	0	10	80	0	10
2		160	No	0	10	120	0	10.5

3		180	Yes	0	9	40	0	9
3		180	Yes	0	9	80	1	8.5
3		180	Yes	0	9	120	1	8
3		180	Yes	0	9	160	2	7.8

Table 2 Final Counting Process Data

Patient	T1 (days)	T2 (days)	Ecog	Hgb (g/dL)	Censoing for Crossover	Censoing for Overall Survival
1	0	39	0	8	1	1
1	39	79	1	7	1	1
1	79	90	2	7	0	1
2	0	119	0	10	1	1
2	119	160	0	10.5	1	1
3	0	79	0	9	1	1
3	79	119	1	8.5	1	1
3	119	159	1	8	1	1
3	159	180	2	7.8	1	0

INTRODUCTION OF THE MACROS

We are presenting two sets of macros to prepare time-varying covariates in counting process style. The first set includes two macros TDDATA and COMB, The TDDATA macro generates a dataset which carries the value of a time-dependent covariate on each day from day 0 to end of duration for the time to event endpoint of interest. To deal with missing values, the last observation carried forward method was used to impute values after available measurements. The COMB macro merges time-dependent covariates generated from the TDDATA macro, time to cross-over data and time to event endpoint data and generates the data in counting process style which can be readily used for IPCW analysis. This set of macros is conceptually simple. It can handle time-dependent covariates collected at different timepoints. However, it creates a lengthy interim dataset for each time-dependent covariate. It will take more time and memory space to run the macros especially when many patients are enrolled and the duration of time to event endpoint is long.

The second set of macros includes the macros LOCF and PROCESS with LOCF imputing missing values and PROCESS creating counting process data. They don't create lengthy interim datasets. They are readily used for the scenario that all time-dependent covariates are collected at the same visits. When the covariates are collected at different timepoints, a straightforward modification is needed, for example, applying LOCF to handle missingness created by merging datasets.

The IPCW macro was presented before (Mosier, 2023). But it can only handle cross-over from control group to treatment group and includes both baseline covariates and time-varying covariates in the outcome model. We modified the macro to accommodate the scenario of two-way crossover. Per NICE technical support document 24 (Gorrod et al. 2024), any prognostic baseline covariates used in the switching model for the numerator of the weights should be included in the outcome model. Thus the modified IPCW macro only includes baseline covariates in the outcome model.

The two sets of data preparation macros and modified IPCW macro are included in the appendix.

A CASE STUDY

We used data SHIdat in R package ipcwswitch to show the process of using the macros presented in this paper to prepare data and perform IPCW analysis. Data SHIdat was collected from a randomized clinical trial to compare molecularly targeted therapy based on tumor molecular profiling (MTA) versus conventional therapy (CT) for advanced cancer (Graffeo etc, 2019). Overall, 100 patients were

randomized to the MTA arm and 97 patients were randomized to the CT arm. And 68 CT patients switched to the MTA arm and 25 MTA patients switched to the CT arm. We used overall survival as the analysis endpoint and included the following covariates in the analysis:

Baseline time-fixed covariates:

Age at randomization

Gender

Number of previous lines of treatment

The dichotomized Royal Marsden Hospital score (0 or 1 vs. 2 or 3)

The altered molecular pathway (distinguishing 3 pathways, namely hormone receptors pathway, PI3K/AKT/mTOR pathway, and RAF/MEK pathway)

Time-varying covariates:

The Eastern Cooperative Oncology Group (ECOG) performance status

The presence of concomitant treatments

The need of platelet transfusions

To use the first set of macros to generate counting process data, we prepare the data of each time-varying covariate in the long format as shown in table 1 and call the TDDATA macro to generate an interim dataset for each covariate with the measurement from day 0 to end of duration of overall survival for each patient.

%***tddata***(data=ps, subject=id, ady=ady,VAR=ps,base=psbase,TYPE=num, length=, tte=os);

%***tddata***(data=tran, subject=id, ady=ady,VAR=tran,base=tranbase,TYPE=num, length=, tte=os);

%***tddata***(data=ttc, subject=id, ady=ady,VAR=ttc,base=ttcbase,TYPE=num, length=, tte=os);

And then use the macro COMB to combine the interim datasets generated from the macro TDDATA and overall survival data and time to crossover data, and generate the counting process data.

```
%comb(n=3,tddata=%str(ps2 tran2 ttc2), tdvar=%str(ps tran ttc), length=20,
subject=id,ttedata=data1,tte=os,ttecnsr=oscnsr,ttcross=ttcross,
crosscnsr=crosscnsr );
```

Then merge the dataset created by %comb with fixed baseline covariate data to create “final” dataset. Run the macro IPCW to perform the analysis. The hazard ratio of overall survival for MTA vs CT is 1.185 with 95% CI (0.746, 1.883).

```
%IPCW(data=final,direction=dual,subject=id,trt=arm,ref=CT,
tstart=start,tend=end,xo_cnsr=crosscnsr,dth_cssr=oscnsr,bl_ct=%str(lnum
agerand),bl_cl=%str(sex rmh pathway),tv_ct=ps,tv_cl=%str(tran ttc));
```

To use the second set of macros, we prepare the data of time-dependent covariates in the long format as shown in table 1. Then we use the LOCF macro to carry forward the measurement of previous visit to the visit with missing value.

```
%locf(dat=psr,varc=ps,ret=psr);

%locf(dat=tranr,varc=tran,ret=tranr);

%locf(dat=ttcr,varc=ttc,ret=ttcr);
```

And the next step is to use the PROCESS macro to create counting process data.

```
%process(dsi=All,var=psr tranr ttcr, dso=final);
```

CONCLUSION

This paper presents two sets of macros to prepare data of multiple time-varying covariates in counting process style. It also presents a modified IPCW macro to accommodate bidirectional crossover, i.e., both treatment and control group patients can crossover to the other group.

REFERENCES

Mosier B. 2023. “Survival Methods for Crossover in Oncology Trials” PharmaSUG proceedings, Paper AP-191. San Francisco, CA.

Gorrod HB, Latimer NR, Abrams KR, NICE DSU Technical Support Document 24: Adjusting Survival Time Estimates In The Presence Of Treatment Switching: An Update To TSD 16, April 2024.

Graffeo N, Latouche A, Le Tourneau C, Chevret S. “IPWSWITCH: An R package for inverse probability of censoring weighting with an appicaiton to switches in clinical trials. Computers in Biology and Medicine. 111 (2019)

APPENDIX

```
*************************************************************************
****************************************;
* This program will generate a dataset which carries the value of a time-
dependent covariate on each day from day 0 to end
 of follow-up for the time to event endpoint of interest. The final dataset
generated is called &data.2, in the dataset generated,
 the time is in variable day and the covariate value is in variable &var.
* By Mei Huang
*************************************************************************
****************************************;

*************************************************************************
****************************************;
* Macro variables:
*
* Data:     the dataset containing time-dependent covariate and time to event
endpoint;
* Subject:  subject ID;
* ady:      study day of post-baseline covariate assessments. calculated as
           covariate assessment date - randomizaton date/start date of
treatment +1;
* var:      the variable containing post-baseline covariate values;
* base:     the variable containing baseline covariate values;
* type:     type of the covariate, the value could be num or char, num means
numeric and char means character;
* length:   if the covariate is character, specify the lenght of it. leave it
blank if it's numeric;
* tte:      the variable for the duration of time to event endpoint, for
example, overall survival;
*************************************************************************
****************************************;
```

```
%macro tddata(data,subject,ady,var,base,type, length, tte);
proc sort data=&data;
 by &subject &ady;
run;
*calculate number of changes in the covariate per patient;
data &data;
 set &data;
 by &subject &ady;
 retain newval ;
 if first.&subject then do; nchg=.; tchg=.;if &var^=&base then do; nchg+1;
tchg=&ady; end; newval=&var; end;
 %if &type=char %then %do;
if &var^='' and &var^=newval then do;
   tchg=&ady;
   newval=&var;
   nchg+1;
 end;
%end;
%if &type=num %then %do;
if &var^=. and &var^=newval then do;
   tchg=&ady;
   newval=&var;
   nchg+1;
 end;
%end;

run;

proc sql;
 select max(nchg) into:maxnchg
 from &data;
quit;

%let maxnchg=%sysevalf(&maxnchg);

proc sql;
 create table nchg&var
 as
 select &subject,max(nchg) as tnchg
 from &data
 group by &subject;
quit;

data &data.1;
 merge &data nchg&var;
 by &subject;
run;

data &data.1;
 set &data.1;
 if tnchg=. then tnchg=0;
run;
*create variables to carry the new value and time of change for each change;
data &data.1;
 set &data.1;
 by &subject;
```

```
  array a(&maxnchg) &var.1-&var.&maxnchg;
 array b(&maxnchg) chgdt1-chgdt&maxnchg;
 retain &var.1-&var.&maxnchg chgdt1-chgdt&maxnchg;
%if &type=num %then %do;
if first.&subject  then do j=1 to &maxnchg;
     a(j)=.; b(j)=.;
end;
 %end;
 %if &type=char %then %do;
 length &var.1-&var.&maxnchg $&length;
 if first.&subject  then do j=1 to &maxnchg;
     a(j)=''; b(j)=.;
 end;
%end;

 do i=1 to &maxnchg;
    if i=nchg and tchg^=. then do;
    a(i)=newval;
      b(i)=&ady;
    end;
  end;
 drop i j;
 if last.&subject;
run;

*create the dataset containing the value of covariate on each day from day 0
to end of duration of the time to event endpoint;
data &data.2;
 set &data.1;
 by &subject;
  array a(&maxnchg) &var.1-&var.&maxnchg;
 array b(&maxnchg) chgdt1-chgdt&maxnchg;
  %if &type=char %then %do;
    length &var $&length;
    if &var.1='' and &base^='' then do i=0 to &tte by 1;
     day=i;
     &var=&base;
    output;
    end;
   else if &var.1^='' then do j=1 to tnchg by 1;
    if j<&maxnchg then k=j+1;
      if j=1 then do;
       do l=0 to b(j)-1 by 1;
           day=l;
           &var=&base;
          output;
          end;
      end;
   %end;

   %if &type=num %then %do;
    if &var.1=. and &base^=. then do i=0 to &tte by 1;
     day=i;
     &var=&base;
    output;
    end;
   else if &var.1^=. then do j=1 to tnchg by 1;
```

```
    if j<&maxnchg then k=j+1;
      if j=1 then do;
       do l=0 to b(j)-1 by 1;
           day=l;
           &var=&base;
          output;
          end;
      end;
   %end;

      if j<&maxnchg then do;
        if b(k)=. then do m=b(j) to &tte by 1;
           day=m;
           &var=a(j);
          output;
        end;
      else do n=b(j) to b(k)-1 by 1;
           day=n;
             &var=a(j);
             output;
         end;
      end;

    if j=&maxnchg then do o=b(j) to &tte by 1;
           day=o;
           &var=a(j);
          output;
       end;
  end;
drop i j k l m n o;
run;

%mend tddata;
*************************************************************************
****************************************;
* This program will combine the datasets generated from the TDDATA macro for
time-dependent covariates, the dataset with
  time to event endpoint data and time to cross-over data, and generates a
dataset in counting process style.
* By Mei Huang
*************************************************************************
****************************************;

*************************************************************************
****************************************;
* Macro variable:
* n:        specify the number of time-dependent covariates, for example, 3
* tddata:   the datasets of time-dependent covariates, seperate the dataset
names by white space, for example, %str(ps2 tran2 ttc2)
* tdvar:    the variables for time-dependent covariates, separate the
variable names by white space
* length:   the macro concatenates time-dependent covariates on the same day
in a character variable, specify the max possible
            length of the concatenated variable
* subject:   subject ID
* ttedata:   the dataset containing time to event endpoint data (during and
censoring variable) and time from randomization/start of
```

```
            treatment to crossover
* tte:      the variable for duration of time to event endpoint
* ttecnsr:  time to event censoring variable (1: censoring, 0: event)
* ttcross:  the variable of time from randomization/start of treatment to
crossover
* crosscnsr: the macro will generate a censoing variable for cross over (1:
censoring, 0: crossover). specify the name for the variable.
************************************************************************
*****************************************;

%macro comb(n, tddata, tdvar, length, subject,ttedata,tte,ttecnsr,
ttcross,crosscnsr );
%do i=1 %to &n;
 %let j=%eval(&i-1);
 %put &j;
  %let tddata&i=%qscan(&tddata,&i);
  %let tdvar&i=%qscan(&tdvar,&i);
  %if &i=1 %then %do;
   %let merge&i=%str(&&tddata&i (keep=&subject day &&tdvar&i));
   %let comvar&i=%str(trim(left(&&tdvar&i)));
  %end;
  %else %do;
   %let merge&i=%str(&&merge&j &&tddata&i (keep=&subject day &&tdvar&i));
   %let comvar&i=%str(&&comvar&j||'|'||trim(left(&&tdvar&i)));
 %end;

%end;
%put &&merge&n &&comvar&n;

data timedep;
 merge &&merge&n;
 by &subject day;
run;

data timedep;
 set timedep;
 by &subject day;
 length comb $&length;
 comb=&&comvar&n;
run;

data timedep1;
 set timedep;
 by &subject;
 retain comb1;
 if first.&subject then do; start=0; comb1=comb; end;
 if comb1^=comb then do; start=day-1; comb1=comb; end;
 if last.&subject then end=day;
run;

data timedep1;
 set timedep1;
 if start^=. or end^=.;
run;
data timedep1;
 set timedep1;
  by &subject;
```

```
  if start>0 and end ^=. then do; output; end=start; start=.; output; end;
  else if start>0 and end=. then do; end=start; output; end;
  else if start=0 and first.&subject=0 and end=. then do; start=1; end=1;
output; end;
 else output;
run;

data start;
 set timedep1;
 if start^=.;
 drop end;
run;

proc sort data=start;
 by &subject start;
run;

data start;
 set start;
 by &subject;
 if first.&subject then seq=1;
  seq+1;
run;

data end;
 set timedep1;
 if end^=.;
 drop start;
run;

proc sort data=end;
 by &subject end;
run;

data end;
 set end;
 by &subject;
 if first.&subject then seq=1;
  seq+1;
run;

data  timedep2 (drop=seq);
 merge start (keep=&subject seq start &tdvar) end (keep=&subject seq end);
 by &subject seq;
run;

data  timedep2 ;
 merge timedep2 &ttedata (in=a keep=&subject &tte &ttecnsr &ttcross);
 by &subject;
 if a;
run;

data timedep2;
 set timedep2;
 by &subject;
 if &ttcross>. then &ttecnsr=1;
  else do;
```

```
   if last.&subject then &ttecnsr=&ttecnsr;
   else &ttecnsr=1;
  end;
  if start>=&ttcross>. then delete;
  if start<&ttcross<=end then do; &crosscnsr=0; end=&ttcross; end;
  else &crosscnsr=1;
run;

%mend comb;


*************************************************************************
****************************************;
* The second example code to generate counting process data .
* By Shibani Harite
*************************************************************************
**List the fixed variables to be included in the data preparation for IPCW;

%let varlist=arm agerand sex lnum rmh pathway randdt ddeath ddn debttco
status;

proc sort data=shidata out=tran;
by id &varlist;
run;

***1. Transpose the data to rows if the baseline and post baseline values are
collected as columns;

%macro transpose(dat=,varc=,ret=);
proc transpose data=tran out=&dat(rename=(col1=&varc));
by id &varlist;
var &varc.1-&varc.21;
run;

data &dat(drop=_name_);
 set &dat;
 day=(tranwrd(_name_,"&varc",''))/1;
run;

proc sort data=&dat ;
by id day &varlist;
run;

%mend transpose;
%transpose(dat=tran4,varc=vsdt);
%transpose(dat=tran1,varc=ps);
%transpose(dat=tran2,varc=tran);
%transpose(dat=tran3,varc=ttc);

***2. Include baseline records along with the post baseline records;
proc sort data=shidata  out=base(keep=id &varlist psbase ttcbase tranbase);
by id;
run;

data base(drop=psbase ttcbase tranbase);
 set base;
 by id;
```

```
 if first.id;
 ps=psbase;
 ttc=ttcbase;
 tran=left(put(tranbase,1.));
run;

data shidata0;
 merge tran1 tran2 tran3 tran4 ;
by id day &varlist;
run;

data shidata1;
 set shidata0 base;
 if day=. then
  do;
   day=0;
   vsdt=put(randdt,date9.);
 end;
 run;

 proc sort;
 by id day &varlist;
run;

***3.last observation carried forward method for missing values in time-
dependent covariates;
*************************************************************************
****************************************;
* Macro variables:
*
* dat:     the dataset containing longitudinal measurements to derive time-
dependent covariates;
* varc:   variable names for which imputation will be done
* ret:    names of variables with imputed values  ;

%macro locf(dat=,varc=,ret=);
data &dat;
 set shidata1;

 if notdigit(&varc) then do;
        /* Convert numeric variable to character */
        &varc.c = left(put(input(&varc, best.), $12.));
    end;
    else do;
        /* Keep character variable as is */
        &varc.c = left(&varc);
    end;

 run;

data &dat(keep=id day &ret);
 set &dat;
 length &ret $20.;
 retain &ret ' ' ;
by id day;
```

```
 if not missing(&varc.c) then &ret = &varc.c;
    else  &varc.c = &ret;
 run;

%mend locf;
%locf(dat=psr,varc=ps,ret=psr);
%locf(dat=tranr,varc=tran,ret=tranr);
%locf(dat=ttcr,varc=ttc,ret=ttcr);


***4.Collapse intervals where time-dependent variables remain unchanged;
*************************************************************************
****************************************;
* Macro variables:
*
* dsi:    input dataset. The dataset dsi should include cv: time to
crossover, os: time to os, cnsr: censoring indicator for OS, cnsr=1 if
censored and 0 if event
* var:   time varying variable to be processed into the counting process
format
* dso:    output dataset name  ;

%macro process(dsi=,var=,dso=);
proc sort data=&dsi out=_dsi;
  by pt cv os cnsr ady ;
run;

data _dsi;
  set _dsi;
  obs=_n_;
run;

proc sort data=_dsi;
  by pt cv os cnsr &var obs  ;
run;

data _dsi;
  set _dsi;
  by pt cv os cnsr &var obs  ;
  pobs=lag(obs);
  if first.pt then pobs=.;

  if first.%scan(&var,-1,%str( )) or (pobs+1 ne obs);
run;

proc sort data=_dsi;
  by pt cv os cnsr descending ady;
run;

data _dsi1;
  set _dsi;
  by pt cv os cnsr descending ady;

  ***specify the intervals when the covariates change;
  pady = lag(ady);
  if first.cnsr then pady=.;
```

```
  if first.cnsr then end=max(cv,os,ady);
  else end=pady-1;

  if ady > 0 then ady=ady-1;
  if end >= ady;

  drop pady;
run;


proc sort data=_dsi1;
  by pt cv os cnsr ady;
run;

data &dso;
  set _dsi1;
  by pt cv os cnsr ady;
  pady=lag(end);
  if first.pt then pady=.;

  if first.pt then start=0;
  else start=pady;

  end=min(cv,os,end);
  if end<start then delete;


 if start < os <= end then oscnsr=1-status;
 else oscnsr=status;

 if cv ^=. then
  do;
    if cv = end then crosscnsr=0;
       else crosscnsr=1;
  end;
 else if cv=. then crosscnsr=1;

run;
%mend;

%process(dsi=All,var=psr tranr ttcr,dso=final);

*************************************************************************
****************************************;
* Modified IPCW macro to perform IPCW analysis. It can handle two-way
crossover, i.e. of either treatment group.
*
*************************************************************************

****************************************;
* Macro variable:
* data:       dataset with time-dependent covariates, fixed baseline
covariates and time to event data in counting process style
* direction:  crossover direction. “direction = single” means only crossover
from control to treatment is allowed, “direction = dual” means crossover to
either arm is allowed
```

```
* subjid:    the variable for subject ID
* trt:       variable for treatment group
* ref:       the reference treatment group in outcome analysis, usually use
control group as reference group.
* tstart:   variable for start time of the time intervals
* tend:     variable for end time of the time intervals
* xo_cnsr:  time to crossover censoring variable (1: censoring, 0: crossover)
* bl_ct:    variables for baseline continuous covariates
* bl_cl:    variables for baseline categorical covariates
* tv_ct:    variables for time-dependent continuous covariates
* tv_cl:    variables for time-dependent categorical covariates
*************************************************************************
****************************************;

%macro IPCW(data,direction,subjid,trt,ref,
tstart,tend,xo_cnsr,dth_cnsr,bl_ct,bl_cl,tv_ct,tv_cl);

%if &direction=single %then %do;
data weight_est;
 set &data;
 where &trt = &ref;
run;
%end;

%if &direction=dual %then %do;
data weight_est;
 set &data;
run;
%end;
proc sort data=weight_est; by &trt &subjid &tstart; run;

proc phreg data=weight_est outest=survmod covs(aggregate) covm;
  by &trt; class &bl_cl;
 model (&tstart,&tend)*&xo_cnsr(1)= &bl_ct &bl_cl/rl; id &subjid;
output out= probs_BL survival = w_BL;
run;

proc phreg data=weight_est outest=survmod covs(aggregate) covm;
* covs option and id statement adjust for repeated measures;
by &trt; class &bl_cl &tv_cl;
model (&tstart,&tend)*&xo_cnsr(1)= &bl_ct &bl_cl &tv_ct &tv_cl /rl;
id &subjid;
output out= probs_TV survival = w_TV;
run;

data probs_BL; set probs_BL; keep &trt &subjid &trt w_BL &tstart &tend; run;
data probs_TV; set probs_TV; keep &trt &subjid &trt w_TV &tstart &tend; run;
proc sort data=probs_TV;
 by &trt &subjid &tstart &tend;
run;
proc sort data=probs_BL;
 by &trt &subjid &tstart &tend;
run;

data weights;
merge probs_TV probs_BL;
by &trt &subjid &tstart &tend;
```

```
sw_t = w_BL/w_TV;
run;
* Truncation Percentiles; %let pctl_l = 1; %let pctl_u = 99;
proc univariate data=weights noprint; var sw_t;
output out=trunc pctlpts = &pctl_l &pctl_u pctlpre = p_;
by &trt;
run;
/*data trunc; set trunc; &trt = &ref1; output; run;*/
data weights; merge weights trunc; by &trt; run;
data weights;
set weights; if sw_t < p_&pctl_l then sw_t = p_&pctl_l; else if sw_t >
p_&pctl_u then sw_t = p_&pctl_u; run;

proc sort data=&data;
 by &trt &subjid &tstart &tend;
run;
proc sort data=weights;
 by &trt &subjid &tstart &tend;
run;

%if &direction=single %then %do;
data surv_ph;
merge &data weights;
by &trt &subjid &tstart &tend;
if &trt ne &ref then do;
sw_t = 1; w_TV = 1; end;
log_w_TV = log(w_TV);
log_sw_t = log(sw_t);
run;
%end;

%if &direction=dual %then %do;
data surv_ph;
merge &data weights;
by &trt &subjid &tstart &tend;
log_w_TV = log(w_TV);
log_sw_t = log(sw_t);
run;
%end;

proc sort data=surv_ph; by &trt; run;


proc phreg data=surv_ph outest=survmod covs(aggregate) covm; * covs option
and id statement adjust for repeated measures;
class &bl_cl &trt(ref="&ref");

model (&tstart,&tend)*&dth_cnsr(1)= &trt &bl_ct &bl_cl  /rl;

id &subjid;
weight sw_t/norm;

run;
%mend IPCW;
```

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Mei Huang
mhuang@exelixis.com

Shibani Harite
sharite@exelixis.com

Athena Zhang
atzhang@exelixis.com

Haijun Ma
hma@exelixis.com