SAS code for the estimation and between-group comparison of cumulative incidence functions in competing risks survival analysis

Simon Bond

Cambridge Clinical Trial Unit, Cambridge University Hospitals NHS Trust, UK

Competing risks extends survival data by also observing a cause of failure. Once a subject fails, it is impossible for him to subsequently fail from any other cause. A well-established tool for summarising competing risks data is the cumulative incidence estimate, which estimates the probability of a subject failing from a specific cause of interest before a given time. Comparisons of the cumulative incidence estimates between groups of subjects can be made using Gray's test. However, there is no commonly available SAS code to perform such analyses, which helps to perpetuate the mistaken use of Kaplan–Meier estimates and log-rank tests in the analysis of competing risks data. This paper presents SAS code to provide cumulative incidence estimates and Gray's tests.

Keywords: Survival analysis, Competing risks, Cumulative incidence, Gray's test, SAS

Introduction

The purpose of this paper is to make available SAS code for calculating the cumulative incidence function, and performing Gray's test,[1] in the context of competing risks survival analysis. Both pieces of statistics methodology have been available for decades as the essential tools to present and compare groups of subjects in data that record a continuous time-to-event variable and a categorical type-of-event variable. However, partly due to the lack of software, the practice of incorrectly presenting Kaplan–Meier curves,[2] and the potentially misleading comparison made by using the log-rank test[3] in isolation, has been perpetuated due to easy availability of software.

This paper will be structured by presenting the assumptions and notation in competing risks data. A section will provide a brief overview of the theoretical comparison between the two sets of tools (cumulative incidence/Gray's test and Kaplan–Meier/log-rank test). It is not an attempt to provide a thorough derivation of the statistical theory, rather it just provides an overview; for further clarification and exposition of classical competing risks theory, we refer to the two text books.[4,5] The subsequent parts of this paper are the section giving the SAS code and a short illustration of its use.

Correspondence to: Simon Bond, Cambridge University Hospitals NHS Foundation Trust, Cambridge Clinical Trials Unit, Level 3-Box 111, Addenbrookes Hospital, Hills Road, Cambridge CB2 0QQ, UK. Email: simon.bond@addenbrookes.nhs.uk

Assumption and Notation

Each subject will experience a unique event taken from a finite set of possible events. This is defined as random variable X, and X can take values from $\{1, 2, \ldots, k\}$. The time from baseline at which this event occurred, is defined as random variable T, and it can take any positive numeric value. Once a subject has experienced their event, also referred to as 'failed', no further events are possible. A classic example of this is mortality, and the event is cause of death; typically in a clinical trial context, there will be a cause of interest (the disease that is being studied) versus all other causes of failure. Imposed on the underlying data structure, there will be a censoring mechanism, whereby a subject can only be under observation up to an independently distributed random time, the censoring time, defined as C. Hence, the data actually observed is (T^*, X^*), which is defined as the minimum of the censoring time C and failure time T

$$T^* = \min(C, T)$$

and we augment the sample space of variable X with a value of 0 for censoring.

$$X^* = 0 \text{ if } C < T,$$

X, otherwise.

The key assumption here is that the censoring is *independent* of the event-type and event-time. So it can be assumed that a censored subject would still observe an event in the future – it is just that we cannot observe the exact time in the future, but do observe a lower

Published by Maney
DOI 10.1179/175709311X13141797023422

bound equal to the censoring time. A classic example is when a clinical trial is closed and all subjects under follow-up are censored at a specific calendar time, so the duration between their baseline (i.e. randomisation) and censoring is independent of the failure time and thus, the censoring mechanism meets the independence assumption. In contrast, when a subject experiences an event from a cause not of primary interest, then this cannot be treated as censoring – the observation tells us that the subject will *never in the future* experience the event of interest.

To understand why the cumulative incidence function is the natural quantity to estimate it helps to consider a case with no censoring. At any point in time, we can count the number of subjects who have failed from each specific cause. When this is converted into a proportion of the total number of subjects, then this is the empirical estimate of the cumulative incidence function. When phrased in terms of the variables T and X, it is an estimate of $\Pr(T<t$ and $X=i)$, for each different cause i. When we sum these across all the different causes, we obtain the overall incidence function $\Pr(T<t)$, which is the complement of the overall survival function $\Pr(T>t)=1-\Pr(T<t)$.

Like (single-cause) survival, when censoring is introduced, we cannot simply use counts and proportions of subjects to provide estimates, since the denominator is varying over time. Rather, we can provide estimates of probabilities conditional on a subject still being at risk at a point in time. A key function is the cause-specific hazard function for each $i=1, 2, \ldots, k$

$$h_i(t)=\mathrm{Lim}_{dt\to 0}\Pr(t<T<t+dt \text{ and } X=i|t<T)/dt$$

which is the risk per unit time of a subject failing from a specific cause, conditional on the subject not failing yet.

We define additionally the cumulative cause-specific hazard function

$$H_i(t)=\int_0^t h_i(u)\mathrm{d}t$$

It can be shown that the overall survival function is

$$\Pr(t<T)=S(t)=-\exp\left[\sum_{j=1}^{k} H_j(t)\right]$$

where it can be seen that the summation of cumulative cause-specific hazards is the overall cumulative hazard due to mutually exclusive events and independent censoring.

The cumulative incidence function is

$$\Pr(T<t \text{ and } X=\mathrm{i})=$$

$$\int_0^t h_i(u)S(u)\mathrm{d}u=\int_0^t h_i(u)\exp\left[-\sum_{j=1}^{k} H_j(u)\right]\mathrm{d}u$$

and for comparison purposes, the complement of the incorrectly calculated Kaplan–Meier curve discussed later is

$$1-\mathrm{KM_i}(t)=\int_0^t h_i(u)\exp[-H_i(u)]\mathrm{d}u=1-\exp[-H_i(t)]$$

which can easily be shown to be greater or equal to the cumulative incidence function.

Comparison of Methods

The Nelson–Aalen estimate[6] is commonly used to estimate the cumulative cause-specific hazard function. It defines a risk set at each time point to be the number of subjects who are uncensored and have not failed (from any cause) yet. Each time a subject fails from a specific cause, the estimate is incremented by 1 divided by the size of the risk set. The definition of the risk set copes naturally with censorings.

Standard software can be used to produce this estimator of the cumulative cause-specific hazard function for each cause. Such software is configured for single-cause survival analysis that uses a status variable to indicate if an observation is a failure or a censoring. So if the data are manipulated to set all causes other than the one of interest to be censorings, then one can see, from re-reading the steps in the previous paragraph, that the resulting estimates of the cumulative hazard function will be correct. Note that this is merely a computational trick, since the artificial censoring is not valid (see comment in the section on 'Assumptions and notation').

Frequently though, these estimates are subsequently used to produce Kaplan–Meier estimates. Indeed, it is a challenge to find software that will produce the Nelsen–Aalen estimates in isolation without producing Kaplan–Meier estimates. Nonetheless, this is a mistake. The Kaplan–Meier curve is an estimate of $\exp[-H_i(t)]$.

This has no clear-cut simple interpretation as the probability of an event. The best explanation propounded is that this would estimate the proportion of subjects surviving if all other causes of failure could be removed, and that the act of removing such causes of failure would not have an impact on the underlying mechanisms that drove the cause of interest. One would need strong external evidence to justify such assumptions: it cannot be assessed with the data.

An alternative perspective on this issue is to consider that a censored observation is formally interpreted as a subject who will still fail from the cause of interest, but at some point in time after the censoring. In reality, any subject who fails from the non-primary cause will never fail from the primary cause in the future.

If we look at the definition of the cumulative incidence function that uses an integral

$$\int_0^t h_i(u)S(u)\mathrm{d}u$$

the effect of this incorrect censoring is to make the

```
*******************************************;
*
* This macro produces estimates and variances for the
* Crude Incidence function of failure time data
* in the prescence of competing risks.
*
*-------------------------------------------;
*
* INPUTS
*
*  in: a data set containing the failure times and event codes
*  time: a variable giving the failure times;
*  status: a variable describing the event codes
*
*  OPTIONAL INPUTS
*
*  event: default value of 1, gives the event code for the cause of interest
*  cens: default value of 0, gives the event code for censored data.
*  out: default value of out, gives the data set that the result are writen to.
*-------------------------------------------;
*
* OUTPUT
* a data set containing the variables:
* time: the ordered failure times excluding censorings
* rs: the number of subjects at risk at any point in time
* nf: the number of failures observed before any point in time
* x: times values to be used in plotting functions
* f: the estimated value
* v: the variance estimate
*
*  The estimate is right-continuous, left-limited function. So for each failure time two rows
* of data are given with identical times points (x) the left-hand limit and the right-hand value
* of the estimate and its variance are given. Hence to plot the estimate as a step function one
* just uses all of  x and f/v. There is an initial  value at the origin, and the last value is
* is extended to the maximum follow-up time, even if no event of interest was observed then.
*
* DEPENDENCIES
*
* The software  uses DATA, PROC SQL, PROC SORT, PROC FREQ, PROC TRANSPOSE from SAS.
*
*-------------------------------------------------------------------------------;
* Author:
* Simon Bond
*
* translated the fortran/R code from the cuminc function in the cmprsk R package produced by RJ Gray
* http://biowww.dfci.harvard.edu/~gray
*
*-------------------------------------------------------------------------------;
*
* This software is distributed under the GNU GENERAL PUBLIC LICENSE Version 3
*
* It comes with no warranty whatsoever.
*
*******************************************************************************;


%macro cuminc( in=, time=, status=, out=out, event=1, cens=0);

proc sort data=&in out=&out;
by &time &status;
run;

data &out; set &out;
select(&status);
   when( &event) &status.2=1;
   when( &cens)  &status.2=0;
   otherwise    &status.2=2;
end;
rename &status.2=&status;
drop &status;
run;

proc sql noprint;
select count(*) into :n from &out;
quit;

proc freq data=&out noprint;
table &time * &status/ out=&out;
run;

proc transpose data=&out out=&out prefix=nd;
by &time;
var count;
id &status;
run;
```

Figure 1 SAS code for %cuminc

```
data &out; set &out end=eof;
retain rs &n;
retain fk  1;
retain nf v1 v2 v3 x f v 0;
if( nd1=.) then nd1=0;
if( nd2=.) then nd2=0;
if( nd0=.) then nd0=0;
nd=nd1+nd2;
if(_n_=1) then do;
   output;
end;

if( nd>0) then do;
   fkn=fk*(rs-nd)/rs;
   if( nd1>0) then do;
      x=&time;
      output;
      f=f+fk*nd1/rs;
   end;
   if( nd2>0 AND fkn>0) then do;
      t5=1;
      if( nd2>1) then t5=1-(nd2-1)/(rs-1);
      t6=fk*fk*t5*nd2/(rs*rs);
      t3=1/fkn;
      t4=f/fkn;
      v1=v1+t4*t4*t6;
      v2=v2+t3*t4*t6;
      v3=v3+t3*t3*t6;
   end;
   if (nd1>0) then do;
      t5=1;
      if( nd1>1) then t5=1-(nd1-1)/(rs-1);
      t6=fk*fk*t5*nd1/(rs*rs);
      t3=0;
      if( fkn>0) then t3=1/fkn;
      t4=1+t3*f;
      v1=v1+t4*t4*t6;
      v2=v2+t3*t4*t6;
      v3=v3+t3*t3*t6;
      t2=f;

      v=v1+t2*t2*v3-2*t2*v2;
   end;
      fk=fkn;
      nf=nf+nd1;
      output;
end;
rs=rs-nd0-nd1-nd2;
if( eof) then do;
   x=&time;
   output;
end;
keep &time rs nf x f v;
run;
%mend;
```

Figure 1 Continued

equivalent of the estimated overall survival function $S(u)$ biased upwards, since it ignores the failures from other causes. Thus, the corresponding complement of the Kaplan–Meier curve is always larger than the correctly calculated cumulative incidence curve.

If the log-rank test is performed on the artificially censored data, then it is a valid test of the hypothesis that the cause-specific hazard, for the event of interest, is constant across groups. This is plausibly a test of interest, and it achieves the laudable aim of ignoring the rates of failure from causes not of interest. Nonetheless, it is still difficult to interpret the act of *isolating* the cause-specific hazard, as described above.

Given the simple interpretation of the cumulative incidence function, we want a test of the different hypothesis that the cumulative incidence function is equal across groups. A standard family of such test is known as Gray's test.[1] It is not correct to infer that if the cumulative incidence functions are equal across groups, the cause-specific hazard functions must be equal or the survival functions must be equal. The cumulative incidence function depends not only on the cause-specific hazards, but also on the probability that a subject is still at risk, the survival function, which in turn depends on *all* the other causes of failure.

However, it is true that both hypotheses considered above contain the more specific hypothesis that the entire data generating mechanism is identical across groups, and so both sets of tests (one for each cause) would be relevant.

The log-rank test, of the equality of cause-specific hazards, and Gray's test, of the equality of cumulative incidence functions, can both be derived in a similar manner. For each group (and possibly stratum), a statistic is formed by taking the sum of time-varying weight functions at every time a subject fails from the cause of interest. The log-rank test uses a weight function that is inversely proportional to the number of subjects who have not failed from *any* cause or been censored at a specific time; Gray's test uses a weight function that is inversely proportional to the number of subjects who have not failed from *the cause of interest* or been censored at a specific time. This summation is repeated but pooled across groups and the difference between the two statistics is obtained. It can be shown that the expectation of the weighted sums only depends on the causes-specific hazards, and the cumulative incidence functions for the log-rank and Gray's test, respectively, hence giving the difference between the grouped and pooled weighted sums an expectation of

```
*******************************************************************************;
*
* This Macro performs a hypothesis test comparing cumulative incidence
* functions between groups for survival data in the presence of competing
* risks [Gray RJ, A class of k-sample tests for comparing the cumulative
* incidence of a competing risk, Annals of Statistics, 1988, V16(3) pp 1141-1154.]
*
*------------------------------------------------------------------------------;
*
* ESSENTIAL INPUTS
* in: data set containing time, status, group and strata (optional)
* time: variable defining the failure times
* status: variable containing codes for events, by default the event of interest is
*       given the code 1, censoring is 0, anything else is any other value
* group: variable containing unique group codes
*
* OPTIONAL INPUTS
*
* strata: variable containg unique stratification codes, if left blank the test is
*       is not stratified
* out: specifies the data set to write the results to, default value = out
* event: changes the default code for the event of interest
* cens: changes the default code for censoring
* rho: a parameter that changes the weighting function: large +ve values give weight to
*     early differences, large -ve give weight to late differences. Also, considering
*     alternative hypotheses of constant Odds Ratio, Hazard Ratio, Cumulative Risk Ratio,
*     these correspond to 1,0,-1 for rho to give optimal power under certain conditions.
*     Default value = 0.
* print: change this from the default value and the results will not be printed.
*
* DEPENDENCIES
*
* The software needs the %relabel macro, which is used to re-code the groups/strata
* into sequential integers.
* It uses DATA, PROC SQL, PROC SORT, PROC IML, PROC FREQ, PROC TRANSPOSE, PROC DATASETS
* PROC PRINT from SAS.
*
*------------------------------------------------------------------------------;
* Author:
* Simon Bond
*
* translated the fortran/R code from the cuminc function in the cmprsk R package produced by RJ Gray
* http://biowww.dfci.harvard.edu/~gray
*
*------------------------------------------------------------------------------;
*
* This software is distributed under the GNU GENERAL PUBLIC LICENSE Version 3
*
* It comes with no warranty whatsoever.
*
*******************************************************************************;




%macro Gray(in=, time=, status=,group=,strata=, out=out, event=1, cens=0, rho=0, print=T);

data temp;
set &in;
%if &strata= %then %do; strata=1; %end;
run;

%if &strata= %then %let strata=strata;

%relabel(in=temp, var=&group, varout=&group.2, merge=T);
%relabel(in=temp, var=&strata, varout=&strata.2, merge=T);

data temp; set temp;
select( &status);
  when(&cens) &status.2=0;
  when(&event) &status.2=1;
  otherwise &status.2=2;
end;

rename &status.2=&status;
rename &group.2=&group;
rename &strata.2=&strata;
drop &status &group &strata ;
run;


proc sort data=temp out=temp;
by &strata &time &status &group;
run;
```

Figure 2 SAS code for %gray

```
proc sql noprint;
select count(*), max(&group), max(&strata) into :no, :ng, :nst from temp;
quit;

%let ng=%trim(&ng);
%let ng1=%trim(%eval(&ng-1));
%let ng2=%trim(%eval(&ng*&ng1/2));
%let vdim=%eval( &ng*(&ng-1)/2);
%let vdim=%trim(&vdim);

%do str = 1 %to &nst;
  data databystrata; set temp;
    where &strata=&str;
  run;

  proc sql noprint;
  select count(*) into :rs_1-:rs_&ng from databystrata group by &group;
  quit;


  proc freq data=databystrata noprint;
  table &time*&status*&group/out=unique;
  run;
  data unique; set unique;
  id="d_"||compress(put(&status,2.))||"_"||compress(put(&group,2.));
  run;

  proc transpose data=unique out=unique ;
  by &time;
  var count;
  id id;
  run;



  data unique; set unique;

  retain s_1-s_&ng1 0;
  retain v_1-v_&ng2 0;
  retain f1m_1-f1m_&ng 0;
  retain f1_1-f1_&ng 0;
  retain fm 0;
  retain f 0;
  retain skmm_1-skmm_&ng 1;
  retain skm_1-skm_&ng 1;
  retain v3_1 - v3_&ng 0;

  %do i= 1 %to &ng;
    retain rs_&i &&rs_&i;
    %do j =1 %to &ng1;
      retain v2_&j._&i 0;
    %end;
    %do j =1 %to &ng;
      retain c_&i._&j 0;
    %end;
  %end;



  %do st=0 %to 2;
    %do gp= 1 %to &ng;
      if( d_&st._&gp =.) then d_&st._&gp=0;
    %end;
  %end;
  nd1=sum(of d_1_1 - d_1_&ng);
  nd2=sum(of d_2_1 - d_2_&ng);

  if( nd1>0 OR nd2>0) then do;
    tr=0;
    tq=0;
    %do i=1 %to &ng;
      if( rs_&i >0) then do;
        td=d_1_&i + d_2_&i;
        skm_&i=skmm_&i *(rs_&i-td)/rs_&i;
        f1_&i=f1m_&i+(skmm_&i *d_1_&i )/ rs_&i;
        tr=tr+rs_&i/skmm_&i;
        tq=tq+rs_&i *( 1- f1m_&i)/skmm_&i;
      end;
    %end;
    f=fm+nd1/tr;
    fb=(1-fm)**&rho;

    %do i=1 %to &ng;
      %do j=&i %to &ng;
        a_&i._&j=0;
      %end;
```

Figure 2 Continued

```
    if( rs_&i>0) then do;
      t1=rs_&i/skmm_&i;
      a_&i._&i=fb*t1*(1-t1/tr);
      c_&i._&i=c_&i._&i+ a_&i._&i * nd1/(tr*(1-fm));
      %let k=%eval(&i+1);
      %if( &k <= &ng) %then %do;
        %do j= &k %to &ng;
          if(  rs_&j>0) then do;
            a_&i._&j=-fb*t1*rs_&j./(skmm_&j.*tr);
            c_&i._&j= c_&i._&j + a_&i._&j*nd1/(tr*(1-fm));
          end;
        %end;
      %end;
    end;
  %end;
  %do i= 2 %to &ng;
    %let k= %eval(&i-1);
    %do j= 1 %to &k;
      a_&i._&j=a_&j._&i;
      c_&i._&j=c_&j._&i;
    %end;
  %end;
  %do i=1 %to &ng1;
    if( rs_&i.>0) then do;
      s_&i=s_&i+fb*(d_1_&i - nd1 * rs_&i*(1-f1m_&i)/(skmm_&i*tq));
    end;
  %end;

  if( nd1>0) then do;
    %do k= 1 %to &ng;
      if( rs_&k>0) then do;
        t4=1;
        if( skm_&k >0) then t4 = 1 - (1-f)/skm_&k;
        t5=1;
        if( nd1>1) then t5=1- (nd1-1)/(tr*skmm_&k-1);
        t3=t5*skmm_&k*nd1/(tr*rs_&k);
        v3_&k=v3_&k+t4*t4*t3;
        %do i = 1 %to &ng1;
          t1=a_&i._&k - t4*c_&i._&k;
          v2_&i._&k=v2_&i._&k + t1*t4*t3;
          %do j= 1 %to &i;
            %let l=%trim(%eval( &i*(&i-1)/2+&j ));
            t2=a_&j._&k - t4 * c_&j._&k;
            v_&l=v_&l+t1*t2*t3;
          %end;
        %end;
      end;
    %end;
  end;
  if( nd2>0) then do;
    %do k=1 %to &ng;
      if( skm_&k>0 AND d_2_&k >0) then do;
        t4=(1-f)/skm_&k;
        t5=1;
        if( d_2_&k > 1) then t5=1-(d_2_&k -1.0)/(rs_&k-1.0) ;
        t3=t5*((skmm_&k **2)* d_2_&k)/(rs_&k **2);
        v3_&k=v3_&k +t4*t4*t3;
        %do i = 1 %to &ng1;
          t1=t4*c_&i._&k;
          v2_&i._&k= v2_&i._&k - t1*t4*t3;
          %do j=1 %to &i;
            %let l=%trim(%eval( &i*(&i-1)/2+&j ));
            t2=t4*c_&j._&k;
            v_&l=v_&l + t1*t2*t3;
          %end;
        %end;
      end;
    %end;
  end;
end;
fm=f;
%do i= 1 %to &ng;
  rs_&i=rs_&i - d_0_&i - d_1_&i - d_2_&i ;
  f1m_&i=f1_&i;
  skmm_&i=skm_&i;
%end;
run;
%let l=0;


data unique; set unique end=last;

%do i=1 %to &ng1;
  %do j=1 %to &i;
    %let l=%trim(%eval(&l+1));
    %do k=1 %to &ng;
      v_&l = v_&l + c_&i._&k * c_&j._&k * v3_&k;
      v_&l = v_&l + c_&i._&k * v2_&j._&k;
      v_&l = v_&l + c_&j._&k * v2_&i._&k;
```

Figure 2 Continued

```
      %end;
    %end;
  %end;
  if last;
  run;


  data stats; set
  %if &str>1 %then %do; stats %end;
    unique;
  run;
%end;

proc sql;
create table stats2 as
select
sum(s_1) as s_1
%if( &ng1>1) %then %do;
  %do i=2 %to &ng1;
    ,sum(s_&i) as s_&i
  %end;
%end;
%do i=1 %to &vdim;
    , sum(v_&i) as v_&i
%end;
from stats;
quit;



proc iml;
use stats2;
%let l=0;
Z=shape(0,&ng1,1);
Sig=shape(0, &ng1,&ng1);
%do i = 1 %to &ng1;
  read var{ s_&i} into temp;
  Z[&i]=temp;
  %do j=1 %to &i;
    %let l=%trim(%eval(&l+1));
    read var{v_&l} into temp;
    Sig[&i, &j]=temp;
    Sig[&j, &i]=Sig[&i, &j];
  %end;
%end;


Chisq=t(Z)*Inv(Sig)*Z  ;
PVal=1-probchi(Chisq,&ng1);
df=&ng1;
result=chisq||pval||df;
name={'Chisq' 'Signif' 'df'};
create &out from result[ colname=name];
append from result;
quit;

%if &print=T %then %do;
  proc print data=&out;
  run;
%end;

proc datasets nolist;
delete databystrata labels stats stats2 temp unique;
run;
quit;
%mend;
```

Figure 2 Continued

zero under the two different null hypotheses. Counting process theory,[7] which is beyond the scope of this paper, can show under mild assumptions that a version of the central-limit theorem applies whereby the covariance matrix of these difference statistics is consistently estimated, and the difference statistics follow a multivariate normal distribution with mean zero under the null. The vector of difference statistics is then scaled by the square root of covariance matrix, squared and summed over groups and strata, to arrive at the familiar chi-squared statistic and *p*-value.

Code

The author of the original article[1] developed two pieces of Fortran code that produces estimates and standard errors of the cumulative incidence curve, and performs the calculations for the hypothesis test of equal cumulative incidence curves across groups of subjects and strata. These pieces of code are currently available as part of an add-on package, or library (cmprsk)[8] to the R system. The R code is available under the GNU General Public Licence,[9] which allows anyone else to use or adapt the code as they

```
*************************************************************;
*
* This macro takes as input a  data set and a specific variable.
* The output is a dataset containing unique values of the variable,
* and a corresponding variable with values 1,2,...n.
* Optionally this output data set is merged with the original dataset
* to map the original variable to the new values.
*
*-------------------------------------------------------------;
*
*  INPUT
*
* in: the input data set
* var: the variable to be re-coded
* labels: the name of the data set defining the mapping
* varout: the name of the new variable
* merge: If given the value T, the new variable will be merged
*      into the original dataset.
*
*-------------------------------------------------------------;
*
* Author:
*
* Simon Bond
*
*************************************************************;

%macro relabel( in=, labels=labels, var=, varout=, merge=);

proc sort data=&in out=&labels nodupkey;
by &var ;
run;

data &labels; set &labels;
&varout=_n_;
keep &var &varout;
run;

%if &merge=T %then %do;
proc sort data=&in out=&in;
by &var;
run;

data &in; merge &in &labels;
by &var;
run;
%end;

%mend;
```

Figure 3 SAS code for %relabel

see fit, but with the constraint that it is passed on under the same licence. Hence, any code herein is published under the same licence.

The SAS code given in Figs. 1–3 is a translation of the underlying Fortran code source code, and consists of three macros: %cuminc, %gray, and %relabel. The first, %cuminc, calculates the cumulative incidence curve and does not require the other two macros; the second performs the Gray's test and uses the %relabel macro to perform some housekeeping to map a variable with an arbitrary set of values to a set of consecutive integers starting at 1. All code is available to download at http://www.mrc-bsu.cam.ac.uk/Software/download.html#SAS.[10]

```
%let path=C:/Define/The/Path;


/* Load the Macros */
%include "&path/cuminc.sas";
%include "&path/Gray.sas";
%include "&path/relabel.sas";

/*
Import the data from http://www.uhnresearch.ca/labs/hill/datasets/Pintilie/datasets/hd6886.txt
Referred to in Chapter 1 of "Melania Pintilie: Competing Risks A Practical Perspective" figures 1.2 and 1.3
*/

proc import file="&path/hodg.txt"
out=hodgkin dbms=csv;
run;

/*
Data Manipulation to:
select the Radiotherapy Subset,
compute the first event time (2nd malignancy or death)
compute which event this is
compute the age group
*/

proc format;
value agegrp 0="<=30 years" 1=">30 years";
run;

data rt; set hodgkin;
format agegrp agegrp.;
where trtgiven="RT";
maligTime=min(survtime, maltime);
select;
        when( maltime < survtime and mcens=1) maligEvent=1;
        when( survtime <= maltime and stat=1) maligEvent=2;
        otherwise maligEvent =0;
end;
if( age<=30) then agegrp=0;
if( age>30) then  agegrp=1;
run;

/* Evaluate Gray's test for 2nd Malignancy and Death */

%gray( in=rt, time=maligTime, status=maligEvent, group=agegrp);
%gray( in=rt, time=maligTime, status=maligEvent, event=2,  group=agegrp);

/*Evaluate the analagous Log-Rank tests */

proc lifetest data=rt;
time maltime*mcens(0) ;
test agegrp;
run;

proc lifetest data=rt;
time survtime*stat(0) ;
test agegrp;
run;

/*
Evaluate the cumulative incidence curves for each age group and event type
*/

%cuminc( in=RT(where=(agegrp=0)), time=maligTime, status=maligEvent, event=1, out=CI1);
%cuminc( in=RT(where=(agegrp=1)), time=maligTime, status=maligEvent, event=1, out=CI2);
%cuminc( in=RT(where=(agegrp=0)), time=maligTime, status=maligEvent, event=2, out=CI3);
%cuminc( in=RT(where=(agegrp=1)), time=maligTime, status=maligEvent, event=2, out=CI4);

/*
Data manipulation to join the age groups together in the same data set for plotting
*/

data CIMalig; set CI1(in=a)  CI2(in=b);
format agegrp agegrp.;
if (a) then agegrp=0;
if (b) then agegrp=1;
label agegrp="Age Group";
run;

data CIDeath; set CI3(in=a)  CI4(in=b);
format agegrp agegrp.;
if (a) then agegrp=0;
if (b) then agegrp=1;
label agegrp="Age Group";
run;

/* Plotting the Cumulative Incidence Curves */
```

Figure 4 SAS code for the worked example

```
symbol1 color=black interpol=join value=none line=1;
symbol2 color=black interpol=join value=none line=2;
axis1 minor=none label=("Time (years)");
axis2 minor=none label=(angle=90 justify=center "Cumulative Incidence") order=(0 to 1 by 0.2);

Title "Second Malignancy";
proc gplot data=CIMalig;
plot f*x=agegrp/ haxis=axis1 vaxis=axis2;
run;
quit;
title "Death Without Second Malignancy";
proc gplot data=CIDeath;
plot f*x=agegrp/ haxis=axis1 vaxis=axis2;
run;
quit;
```

Figure 4 Continued

The translation is almost a literal one, which was a purposeful decision to guarantee reproducibility of the original code. The main difference is that whenever possible, the outer-most loop of the Fortran code is replaced by the line-by-line functionality of the SAS DATA step, to increase speed and readability. Inner loops, though, have been handled using macro preprocessing. The code comes with no warranty, and in particular, there is no checking for invalid inputs.

Example

The example data and analysis in Fig. 4 replicate that given in the first chapter of Pintilie's book.[5] It considers a set of 616 subjects with Hodgkin's disease who are treated with radiotherapy. The two competing risks are recurrence of second malignancy, or death without such a malignancy. The effect of age is considered by defining two groups of subjects according to their age at baseline being $\leqslant 30$ or >30. It provides a classic case where the log-rank test produces a statistically significant result comparing age groups, but the cumulative incidence curves are very similar and show little evidence of a difference.

The data are available from the website[11] and is derived from the original article.[12] The SAS code that follows illustrates how to produce the analysis.

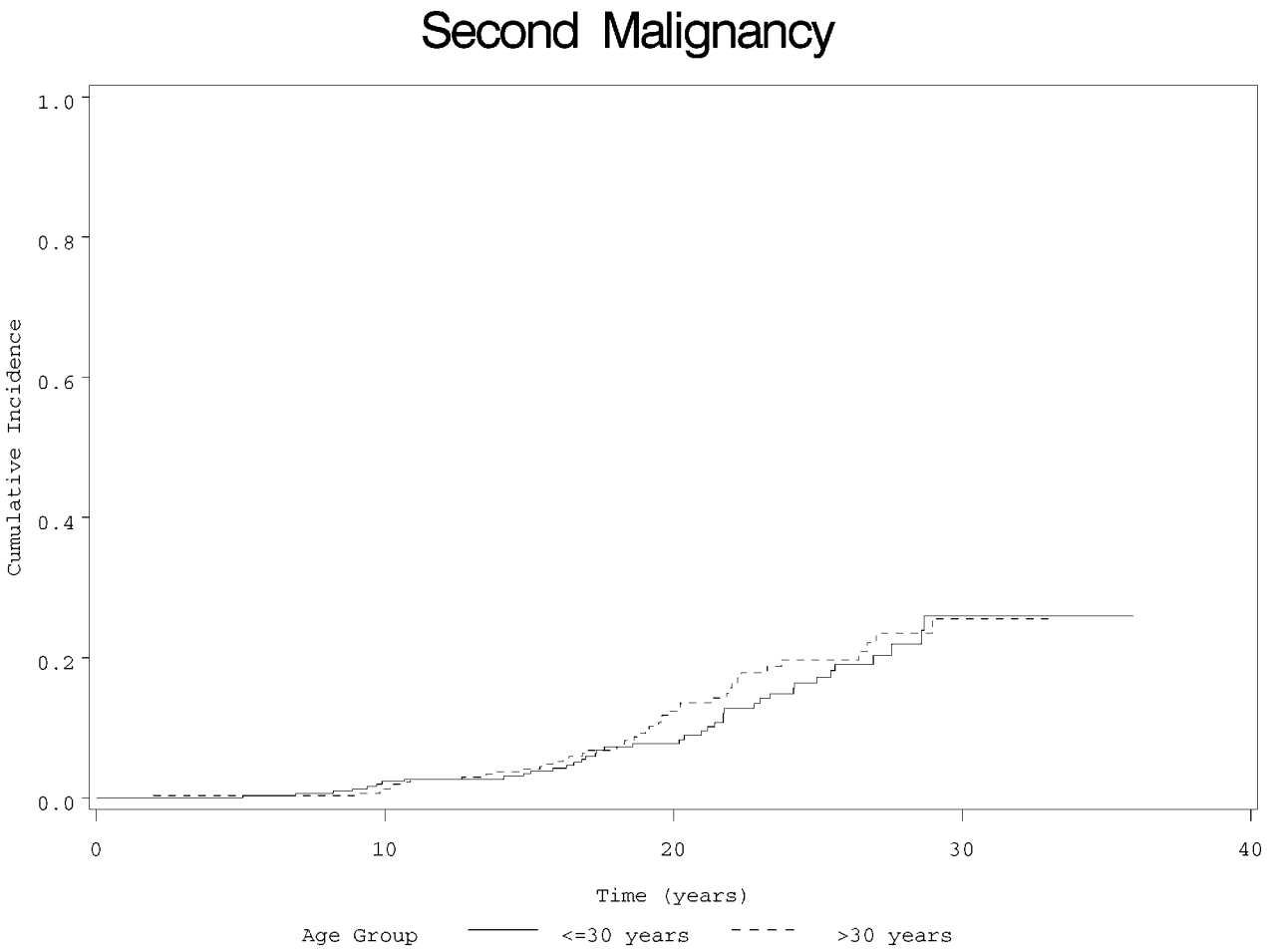

Figure 5 Cumulative incidence curves for second malignancy by age group

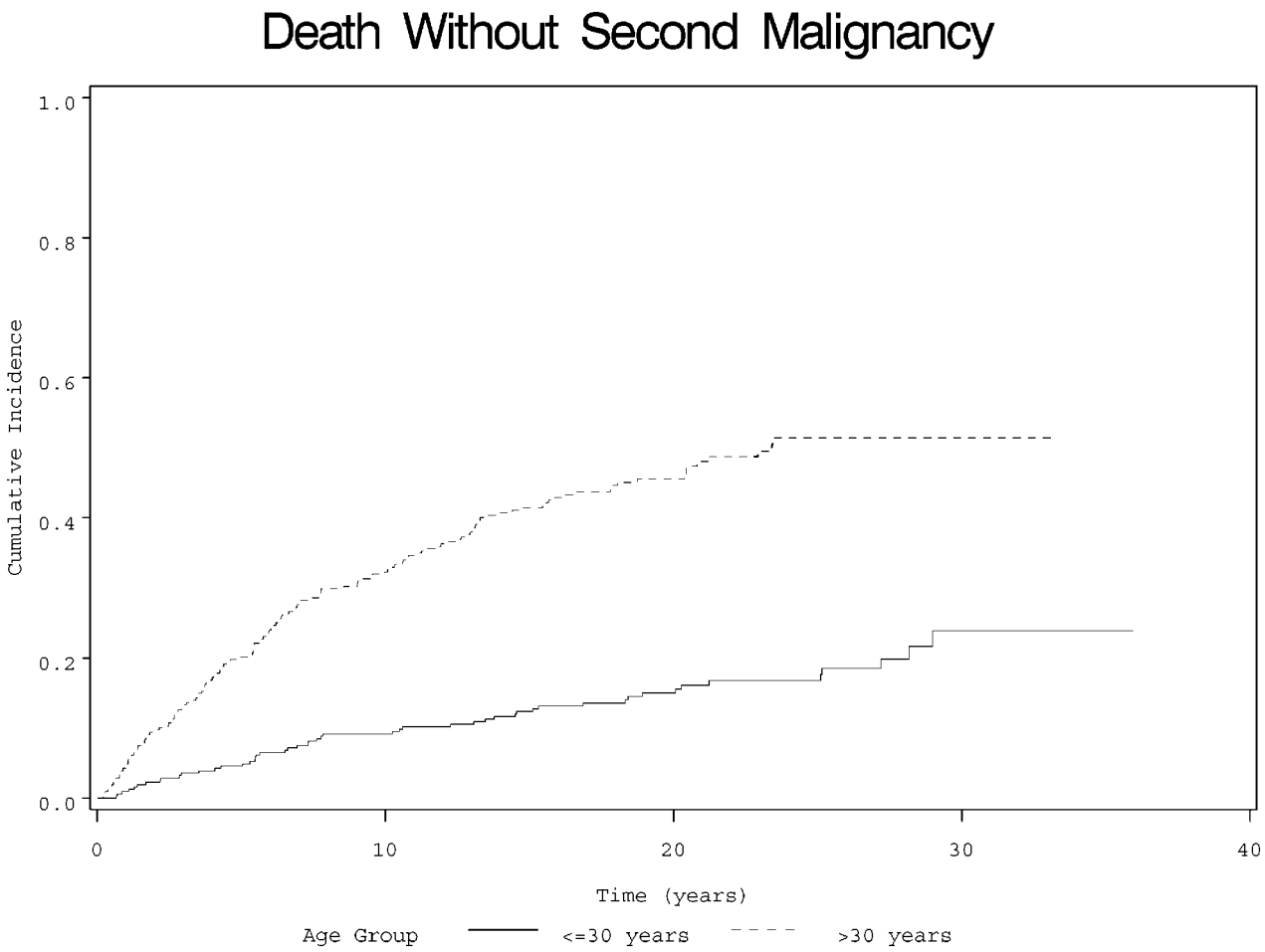

Figure 6 Cumulative incidence curves for death without second malignancy by age group

The output from the two calls of the %gray macro is given below.

Obs	Chisq	Signif	df
1	0.41476	0.51956	1

for the time to second malignancy, and

Obs	Chisq	Signif	df
1	67.9160	2.2204×10^{-16}	1

for death.

The chi-squared statistics and significance levels for the corresponding log-rank tests are: 9.4010 and 0.0022 for the time to second malignancy, and 86.4871 and 1.407E-20 for death.

These duplicate the values printed in the book (up to the rounding in the book) and exactly duplicate the equivalent analysis performed using the R system.

The two graphs of the cumulative incidence curves produced are given in Figs. 5 and 6.

Further testing of different analyses and datasets has shown 100% reproducibility of the same analyses performed using R system.

Conclusion

A brief overview of the motivation and theory behind this methodology has been provided here. Further expansion and explanation is provided by two textbooks[4,5] and the counting process theory underlying the asymptotic use of the central limit theory is given Andersen *et al.*[7]

The original contribution made in this article is the translation of the Fortran code used to perform Gray's test[1] into SAS code. As such, it is hoped that this will encourage the practice of routinely complementing the analysis of competing-risks survival data with the standard log-rank tests[3] with the equally informative Gray's test, and replacing the incorrect use of the Kaplan–Meier curves[2] with the cumulative incidence curves.

Acknowledgements

I would like to acknowledge the help of Adrian Mander (MRC Biostatistics Unit Hub for Trials Methodology Research) for general help in preparing this manuscript.

References

1 Gray RJ. A class of *k*-sample tests for comparing the cumulative incidence of competing risks. Ann Stat 1988;**16**:1141–54.
2 Kaplan EL, Meier P. Non-parametric estimation from incomplete observations. J Am Stat Assoc 1958;**53**:457–81, 562–63.
3 Peto R, Peto J. Asymptotically efficient rank invariant test procedures (with discussion). J R Stat Soc Ser A 1972;**135**:185–206.
4 Crowder MJ. Classical competing risks. Boca Raton (FL): Chapman & Hall; 2001.
5 Pintilie M. Competing risk: a practical perspective. Chichester: Wiley; 2006.
6 Aalen OO. Statistical inference for a family of counting processes [PhD thesis]. Berkeley (CA): University of California; 1975.
7 Andersen PK, Borgan O, Gill RD, Keiding N. Statistical models based on counting processes. New York: Springer-Verlag; 1993.
8 Gray RJ. Subdistribution analysis of competing risks [document on the Internet]. Comprehensive R Archive Network, version 2.2-1 [published 2010 Jan 4; cited 2011 Jan 5]. Available from: http://cran.r-project.org/web/packages/cmprsk/.
9 Free Software Foundation. GNU general public licence, version 3 [document on the Internet]. Boston (MA): Free Software Foundation [published 2007 Jun 29; cited 2011 Jan 5]. Available from: http://www.gnu.org/licenses/gpl.html.
10 Medical Research Council Biostatistics Unit. Software Repository [cited 2011 Jan 25]. Available from: http://www.mrc-bsu.cam.ac.uk/Software/download.html#SAS.
11 Pintilie M. Hodgkin's disease dataset [document on the Internet]. Toronto: The Tumour Microenvironment Group [cited 2011 Jan 5]. Available from: http://www.uhnres.utoronto.ca/labs/hill/datasets/Pintilie/datasets/hd6886.txt
12 Petersen P, Tsang R, Gospodarowicz M, Pintilie M, Wells W, Hodgson D, *et al*. Stage I and II Hodgkin's disease: long term outcome and second cancer risk. Radiother Oncol 2004;**72**:S23.