Program 16.3 SAS code for performing cost analysis

libname local "C:\Documents and Settings\hongwei\My Documents\Home\SASbook\Example";

/\* Read survival data \*/

use local.surv1;

read all var {id delta surv};

/\*Subject ID, death indicator, and survival time;\*/

close local.surv1;

/\* Read cost data \*/

use local.cost1;

read all var {cid start stop cost};

/\* Subject ID, cost starting date, stop date, cost incurred \*/

show names;

close local.cost1;

/\* Define global variables \*/

n=nrow(id); /\* number of subjects \*/

nobs=nrow(cid); /\* total number of observations for cost data\*/

L=**1461**; /\* time limit \*/

r=**0.03**; /\* annual discount rate \*/

/\* Truncate survival time to L, name new variables tsurv and tdelta \*/

**run** TrunSurv(delta, surv, tdelta, tsurv);

/\* Make the largest observation a failure \*/

do i= **1** to n;

if (tsurv[i] = L) then tdelta[i]=**1**;

end;

/\* Calculate the percentage of data that is censored \*/

percens=CalCensor(tdelta);

print , "Percent of censoring =" percens;

/\* Calculate the Kaplan Meier estimator for K(t)=Pr(C>t), name it kc \*/

**run** KmCal(tsurv,tdelta,kc);

/\* Calculate the Kaplan Meier estimator for S(t)=Pr(C>t), name it s \*/

censor=j(n,**1**,**0**);

do i= **1** to n;

censor[i]=**1**-tdelta[i];

end;

**run** KmCal(tsurv,censor,s);

/\* Calculate the total cost for each subject, which is needed for the simple weighted estimator \*/

**run** CalTCost(cid, start,stop, cost, id, tsurv, tcost);

/\* Calculate the mean cost using the simple weighted estimator \*/

mean\_sw=CalOurMean(tdelta, kc, tcost);

print , "Simple weighted estimator for mean cost =" mean\_sw;

/\* Calculate the standard error of the simple weighted estimator \*/

var\_sw=CalOurVar(tsurv, tdelta, s, kc, tcost, mean\_sw);

se\_sw=sqrt(var\_sw);

print , "Standard error estimate for the simple weighted estimator =" se\_sw;

/\* Calculate the mean discounted survival time and its standard error\*/

dsurv=j(n,**1**,**0**);

do i= **1** to n;

dsurv[i] = **365.25**/r \* (**1.0**-exp(-r\*(double)tsurv[i]/**365.25**));

end;

mean\_T = CalOurMean(tdelta, kc, dsurv);

var\_T = CalOurVar(tsurv, tdelta, s, kc, dsurv, mean\_T);

se\_T = sqrt(var\_T);

print , "Mean survival time =" mean\_T;

print , "Standard error for the mean survival time =" se\_T;

/\* Calculate cumulative cost at each censored time, which is needed for the improved estimator \*/

**run** CalCulCost(cid, start, stop, cost, id, tsurv, tdelta, culcost);

/\* Calculate improved estimator for mean cost and its standard error\*/

**run** CalMeanAdd(tsurv, tdelta, kc, s, tcost, culcost, meanadd, varsub);

mean\_imp = mean\_sw+meanadd;

se\_imp = sqrt(var\_sw-varsub);

print , "Imporved estimator for mean cost =" mean\_imp;

print , "Standard error of the improved estimator =" se\_imp;

/\* Calculate the covariance between mean survival time and simple weighted cost estimator \*/

cov\_sw = CalOurCov(tdelta, tsurv, s, kc, tcost, mean\_sw, dsurv, mean\_T);

print , "Covariance between mean survival time and the simple weighted cost estimator =" cov\_sw;

/\* Calculate the covariance between mean survival time and improved cost estimator \*/

covsub=CalCovSub(tdelta, tsurv, s, kc, culcost, dsurv);

cov\_imp=cov\_sw-covsub;

print , "Covariance between mean survival time and the improved cost estimator =" cov\_imp;

/\* Subroutine to truncate the survival time to L \*/

start TrunSurv(delta, surv, tdelta, tsurv) global (L,n);

tsurv=surv;

tdelta=delta;

do i= **1** to n;

if surv[i]>L then do;

tsurv[i]=L;

tdelta[i]=**1**;

end;

end;

finish TrunSurv;

/\* Subroutine to calculate the percentage of data that is censored \*/

start CalCensor(tdelta) global (L,n);

cens=**1**-tdelta;

percens = sum(cens)/n;

return(percens);

finish CalCensor;

/\* Subroutine to calculate the Kaplan Meier estimator for K(t)=Pr(C>t) \*/

start KmCal(surv, delta,kc);

create InputDataSet var {surv delta};

append;

close InputDataSet;

submit;

**proc** **lifetest** data=InputDataSet noprint outsurv=OutputData;

time surv\*delta(**1**);

**run**;

**data** Out;

set OutputData;

tpdelta=**1**-CENSOR\_;

tpsurv=surv;

**run**;

endsubmit;

use Out;

read all var {tpsurv tpdelta survival};

**run** ChangeKmSurv(tpsurv, tpdelta, survival, surv, delta, kc);

close Out;

finish KmCal;

/\* Subroutine to carry forward the survival function estimate at the last failure time \*/

start ChangeKmSurv(tpsurv, tpdelta, survival, surv, delta, kc) global (L,n);

minkc=**1000**;

nn=nrow(tpsurv);

kc=j(n,**1**,**0**);

do j= **1** to nn;

if (survival[j]>=**0** & survival[j]<minkc) then do;

minkc=survival[j];

maxtime=tpsurv[j];

end;

end;

do i= **1** to n;

if (surv[i]>maxtime) then kc[i]=minkc;

else do;

do j = **1** to nn;

if (surv[i]=tpsurv[j]) then kc[i]=survival[j];

end;

end;

end;

finish ChangeKmSurv;

/\* Subroutine to calculate the total cost \*/

/\* This subroutine takes less time to run compared to the routine calculating cumulative cost \*/

start CalTCost(cid, start, stop, cost, id, tsurv, tcost) global (n, nobs);

tcost=j(n,**1**,**0**);

do i= **1** to n;

do k=**1** to nobs;

if (cid [k] = id[i] & start [k] <= tsurv[i]) then do;

if (stop[k] > tsurv[i]) then

tcost[i]=tcost[i]+cost[k]\*(tsurv[i]-start[k]+**1.0**)/(stop[k]-start[k]+**1.0**);

else tcost[i] = tcost[i] + cost[k];

end;

end;

end;

finish CalTCost;

/\* Subroutine to calculate the simple weighted estimator for the mean cost \*/

start CalOurMean(tdelta, kc, tcost) global (n);

mymean=**0.**;

do i= **1** to n;

if (tdelta[i]=**1**) then mymean = mymean + tcost[i]/kc[i];

end;

mymean = mymean/n;

return(mymean);

finish CalOurMean;

/\* Subroutine to calculate the variance of the simple weighted estimator \*/

start CalOurVar(tsurv, tdelta, s, kc, tcost, mymean)global (n);

temp1 = **0.**; /\* part 1 of equation (2) \*/

temp2 = **0.**; /\* part 2 of equation (2) \*/

do i= **1** to n;

if (tdelta[i]=**1**) then temp1 =temp1 + (tcost[i]-mymean)\*\***2**/kc[i];

end;

temp1 =temp1/n;

do j= **1** to n;

e=**0.**;

f=**0.**;

if (tdelta[j]=**0**) then do;

do i= **1** to n;

if(tdelta[i]=**1** & tsurv[i]>=tsurv[j]) then do;

e =e + tcost[i]/kc[i];

f = f + (tcost[i])\*\***2**/kc[i];

end;

end;

e = e / (s[j]\*n);

f = f /(s[j]\*n);

temp2 = temp2 + (f-e\*e)/(kc[j]\*kc[j]);

end;

end;

temp2=temp2/n;

myvar = temp1+temp2;

myvar = myvar/n;

return(myvar);

finish CalOurVar;

/\* Subroutine to calculate the cumulative cost \*/

/\* This routine is needed for calculating the improved estimator \*/

start CalCulCost(cid, start, stop, cost, id, tsurv, tdelta, culcost) global (n, nobs);

culcost=j(n,n,**0**);

do i= **1** to n;

do j= **1** to n;

if (tsurv[i]>=tsurv[j] & tdelta[j]=**0**) then do;

do k = **1** to nobs;

if (cid [k] = id[i] & start [k] <= tsurv[j]) then do;

if (stop[k] > tsurv[j]) then

culcost[i,j]=culcost[i,j]+cost[k]\*(tsurv[j]-start[k]+**1.0**)/(stop[k]-start[k]+**1.0**);

else culcost[i,j] = culcost[i,j] + cost[k];

end;

end;

end;

end;

end;

finish CalCulCost;

/\* Subroutine to calculate the additional terms for the improved estimator and its variance \*/

start CalMeanAdd(tsurv, tdelta, kc, s, tcost, culcost, meanadd, varsub) global (n);

/\* First calculate Ubar[j] and risk set y[j] at censoring places \*/

Ubar=j(n,**1**,**0**);

y=j(n,**1**,**0**);

do j= **1** to n;

if (tdelta[j]=**0**) then do;

do i= **1** to n;

if (tsurv[i]>=tsurv[j]) then do;

Ubar[j]= Ubar[j]+ culcost[i,j];

y[j] = y[j]+**1**;

end;

end;

Ubar[j]= Ubar[j]/y[j];

end;

end;

/\* Next calculate the additional terms for the improved estimator and its variance \*/

part1=**0.**; /\* Additional term for the improved estimator \*/

part2=**0.**; /\* Second term in the variance formula for the improved estimator, equation (4) \*/

part3=**0.**; /\* Third term in the variance formula for the improved estimator, equation (4) \*/

do j= **1** to n;

if (tdelta[j]=**0**) then do;

part1 = part1+ (tcost[j]-Ubar[j])/kc[j];

gu=**0.**;

par2temp=**0.**;

par3temp=**0.**;

do i= **1** to n;

if(tdelta[i]=**1** & tsurv[i]>=tsurv[j]) then

gu = gu + tcost[i]/kc[i];

end;

gu = gu/( s[j]\*n);

do i= **1** to n;

if(tdelta[i]=**1** & tsurv[i]>=tsurv[j]) then

par2temp = par2temp+ (tcost[i]-gu)\*(culcost[i,j]-Ubar[j])/kc[i];

end;

part2 = part2 + par2temp/(y[j]\*kc[j]);

do i= **1** to n;

if(tsurv[i]>=tsurv[j])then

par3temp = par3temp +(culcost[i,j]-Ubar[j])\*\***2**;

end;

part3 = part3+ par3temp/(y[j]\*kc[j]\*kc[j]);

end;

end;

part1 = part1/n;

meanadd=part1;

varsub=(**2.0**\*part2-part3)/(n\*n);

finish CalMeanAdd;

/\* Subroutine to calculate the covariance between mean survival time and simple weighted cost estimator \*/

start CalOurCov(tdelta, tsurv, s, kc, tcost, mymean, dsurv, tmean) global (n);

temp1 = **0.**;

temp2 = **0.**;

do i= **1** to n;

if (tdelta[i]=**1**) then temp1 = temp1 + tcost[i]\*dsurv[i]/kc[i];

end;

temp1 = temp1/n;

temp1 = temp1 - mymean \* tmean;

do j= **1** to n;

gtc=**0.**;

gt=**0.**;

gc=**0.**;

if (tdelta[j]=**0**) then do;

do i= **1** to n;

if(tdelta[i]=**1** & tsurv[i]>=tsurv[j]) then do;

gtc = gtc + tcost[i]\*dsurv[i]/kc[i];

gt = gt + dsurv[i]/kc[i];

gc = gc + tcost[i]/kc[i];

end;

end;

gtc = gtc / (s[j]\*n);

gt = gt/(s[j]\*n);

gc = gc/(s[j]\*n);

temp2 = temp2 +(gtc-gt\*gc)/(kc[j]\*kc[j]);

end;

end;

temp2 =temp2/n;

mycov = temp1+temp2;

mycov = mycov/n;

return(mycov);

finish CalOurCov;

/\* Subroutine to calculate the additional term for covariance between mean survival time and improved cost estimator \*/

start CalCovSub(tdelta, tsurv, s, kc, culcost, dsurv) global (n);

/\* First calculate Ubar[j] and risk set y[j] at censoring places \*/

Ubar=j(n,**1**,**0**);

y=j(n,**1**,**0**);

do j= **1** to n;

if (tdelta[j]=**0**) then do;

do i= **1** to n;

if (tsurv[i]>=tsurv[j]) then do;

Ubar[j] = Ubar[j] + culcost[i,j];

y[j] = y[j] + **1**;

end;

end;

Ubar[j] =Ubar[j]/y[j];

end;

end;

/\* Next calculate the additional term for the covariance using improved cost estimator \*/

part2=**0.**;

do j= **1** to n;

if (tdelta[j]=**0**) then do;

par2temp=**0.**;

gt = **0.**;

do i= **1** to n;

if(tdelta[i]=**1** & tsurv[i]>=tsurv[j]) then

gt = gt + dsurv[i]/kc[i];

end;

gt = gt/(s[j]\*n);

do i= **1** to n;

if(tdelta[i]=**1** & tsurv[i]>=tsurv[j]) then

par2temp = par2temp+(culcost[i,j]-Ubar[j])\*(dsurv[i]-gt)/kc[i];

end;

part2 = part2 + par2temp/(y[j]\*kc[j]);

end;

end;

covsub=part2/(n\*n);

return(covsub);

finish CalCovSub;