SURVIVAL ANALYSIS FINAL
(Breast Cosmetic Deterioration Study)

Description of the Problem

The data in our study is both right-censored and interval-censored. The only information we have for interval-censored data is that each individual's event time falls in an interval, but their exact event time is unknown. Until now, we have not dealt with interval-censored data. Our problem is to:

(i) Non-parametrically estimate a survival function for interval-censored data given in section 1.18 of text (Breast Cosmetic Deterioration Study data set).

(ii) Test for a difference between the two survival functions for the two groups.

Description of the Dataset

This retrospective study of cosmetic deterioration of breast cancer contains 95 observations. There are two treatment groups of patients: 46 radiation only patients and 48 radiation plus chemotherapy patients. The event of interest was the time (in months) to first appearance of moderate or severe breast retraction. The data consists of the interval-censored observations (n=58) in which breast deterioration occurred and right-censored observations (n=37) of the last time the patient was seen without yet having breast deterioration occurred at the last visit.

Method

I. How I handled the censored data

The interval-censored data appears in the dataset as $(L_i,R_i]$ i=1,2,…,n, while the right-censored data appears at $>=L_i$ (i.e, $[L_i,\infty)$) for i=1,2,…,n. Notice that the interval-censored data does not include its left-hand endpoint. To put the data on the same scale, I wrote the right-censored data as (L_{i-1}, ∞) for i=1,2,…,n.

The maximum R_i in the interval-censored data is R_i=60. For the purpose of implementing code to distribute the probability mass, I arbitrarily used R_i=100 instead of infinity for the right-censored data. This does not impact how probability mass will be distributed for the right-censored data, but simply gives the program some endpoint > 60 months to use, as SAS cannot handle infinity.

II. Description of statistical method used in problem (i)

An estimate of the survival function is found by a modification to Turnbull's Algorithm (used in cases of doubly-censored data). The algorithm puts the interval censored data into a familiar form of right-censored data structure. The algorithm repeats itself until it reaches some specified convergence criteria by taking the maximum difference of the initial survival estimate with an updated survival estimate.

In this context, we let time points be $0 = \tau_0 < \tau_1 < \ldots < \tau_m$, which includes all points of Li and Ri, for i = 1,2,…,n. For the i^{th} observation we define an indicator of whether the event occurring in interval $(L_i, R_i]$ could have occurred at τ_j, as follows:

α_{ij} = { 1 if $(\tau_{j-1}, \tau_j]$ interval is contained in $(L_i,R_i]$ interval; 0 otherwise }.

Next, an initial estimate is made for $S(\tau_j)$ by equally distributing the probability mass of the the i^{th} individual to each value of τ can possibly take in the interval $(L_i,R_i]$. The initial estimate made for $S(\tau_j)$ is found by equally distributing the probability mass of 1/46 for radiation only patients, since there are 46 patients receiving radiation only. Similarly, the initial estimate made for $S(\tau_j)$ is found by equally distributing the probability mass of 1/48 for radiation plus chemotherapy, since there are 48 patients receiving this treatment combination. Further explanation of the initial estimates for $S(\tau_j)$ are given below. See Figure1 and Figure2 in *results* section for these initial estimates.

Initially the probability of an event occurring at time τ_j is computed by taking the difference between the survival estimates at time τ_{j-1} and time τ_j
[i.e., pj = $S(\tau_{j-1})$ - $S(\tau_j)$, for j = 1,…,m]. After the first time through this iteration, this probability is calculated by the updated $S(\tau_j)$.

The next step is to estimate the number of events which occurred at time τ_i. This is done by first dividing the probability of the event occurring at time τ_j by the total probability assigned to possible event times in the interval $(L_i,R_i]$. This same calculation is done for all observations i=1,…,n. The estimated number of events occurring is at τ_i is the sum of all these calculations. A justification for this step is given below.

Now, the estimated number at risk (Y_j) at time τ_j is calculated by summing up the estimated number of events occurring at times $\tau_j,\ldots, \tau_m$. Finally, we can find an updated survival estimate for times $\tau_0, \tau_1,\ldots, \tau_m$ by using the Product-Limit estimator.

If the maximum difference between the initial survival estimates and the updated survival estimates is less than or equal to our convergence criterion, then the process is stopped. Otherwise, this process is continued by letting the next iteration's initial survival estimates be those of the updated survival estimates. The final survival estimates are plotted in Figure3 of the *results* section.

III. Justification of Step2 on page 144

As mentioned earlier, this step tries to estimate the number of events which at occurred at time τ_i.

This step can be expressed as follows:
Let's say that at event time τ_i there are *k* intervals $(L_i,R_i]$ that contain event time τ_i, denoted $(L_i,R_i]_1,\ldots, (L_i,R_i]_k$. Further, let's denote the total probability assigned to possible event times in the interval $(L_i,R_i]_l$ for l=1,…,k as $Pr\{(L_i,R_i]_l\}$. Then the number of events which occurred at time τ_i can be expressed as follows:

$$di = \frac{pi}{Pr\{(L_i,R_i]_1\}} + \frac{pi}{Pr\{(L_i,R_i]_2\}} + \ldots + \frac{pi}{Pr\{(L_i,R_i]_k\}}.$$

As you can see each term is a probability divided by another probability equal to or larger than it. Each term can take a maximum value of 1. Thus, di <= k, where k is the

number of terms (defined earlier). If the only event time in the intervals $(L_i,R_i]_1,\ldots,(L_i,R_i]_k$ is τ_i, then di = k. That is, the number of events at time τ_i is equal to the number of intervals containing this event time τ_i. Otherwise, each term will represent the proportion of that event time τ_i to all the events in that particular interval (i.e., the proportion/frequency of event time τ_i to all events within each observation). Then di would be the sum of all those terms, which would in turn provide an estimate to the frequency of that event overall. Therefore, di is a reasonable estimate to the number of events which occurred at time τ_i.

IV. Implementation Details to problem (i)

a. Choice of starting values for $S(\tau_j)$

Assume $S(\tau_0) = 1$. We can calculate the probability mass at each time point τ_j, j = 1,…,m. Then we can calculate $S(\tau_j) = S(\tau_{j-1}) - pm(\tau_j)$, where $pm(\tau_j)$ is the total probability mass given at that time point j.

Perhaps, a better explanation of this is done by example:
Consider the radiation only treatment, and time τ_1 =4 months. There are three intervals that contain this time point given below in the table. The probability mass is distributed equally for each time point as 1/46 (for reasons given above). Then each interval has probability mass of 1, so it too distributes its probability mass as seen in the following table.

Prob mass distribution at τ_1 =4 months
Intervals including τ_1 =4 \ Times

	0	4	5	6	7	8	.
(0,5]		*0.50/46*	*0.50/46*				
(0,7]		*0.25/46*	*0.25/46*	*0.25/46*	*0.25/46*		
(0,8]		*0.20/46*	*0.20/46*	*0.20/46*	*0.20/46*	*0.20/46*	
Total Prob Mass		*0.0207*					

⇨ $pm(\tau_1) = 0.0207$
⇨ $S(\tau_1) = S(\tau_0) - pm(\tau_1) = 1 - 0.0207 = 0.9793$

This process can be repeated for at time intervals τ_j, j=1,..,m, to get our starting values for $S(\tau_j)$. The starting values can be seen in Figure1 and Figure2 in the *results* section.

b. *Choice of convergence criterion = 0.0000001*

This was chosen as the convergence criterion because we needed a measure sufficiently small enough so that our survival estimates are unbiased (or as unbiased as we feel safe enough about). Having a smaller convergence criterion beyond our choice would make a trivial difference in our estimates of the survival curves.

V. Description of statistical method used in problem (ii)

To test if there is a difference between the two treatments, we need to perform a two-sample test. In the solution to problem (i), the survival estimates are given in the form of right-censored data structure. This data structure will also be used to solve problem (ii). After some research, I decided to use the two-sample Kolmogorov-Smirnov test to test for a difference in these survival functions.

The Kolmogorov-Smirnov (K-S) test is asymptotically nonparametric (distribution-free), and generally has good power properties. They hypothesis for this test are:

Ho: $S_{\text{radiation only}} = S_{\text{radiation plus chemotherapy}}$
H1: $S_{\text{radiation only}}$ not equal to $S_{\text{radiation plus chemotherapy}}$

The asymptotic distribution of the test statistic from the K-S test under Ho is generally unknown because it depends on the underlying distribution of the data. Another explanation to why K-S test is not applied to the survival estimates directly is because the K-S test is not consistent when there are point masses in the two survival curves being compared. A bootstrap sample of the survival estimates is taken to overcome this downfall. The bootstrap can provide robust estimate of the asymptotic variance and standard errors, which is necessary for the K-S test to be useful.

First, two bootstrap samples are created based on of the survival estimates given in problem (i), one sample for each of the treatments. Then the Kolmogorov-Smirnov test can be applied to these samples.

The bootstrap sample is with replacement from our survival estimates separately for each treatment. The probability of an event occurring at time τ_j is now computed again by taking the difference between of $S(\tau_j)$ and $S(\tau_{j+1})$, for $j=0,\ldots,m-1$. A sample of 95 observations is taken using this probability in determining the bootstrap sample. A sample of 95 observations is taken since we began with 95 observations.

RESULTS:

INITIAL VALUES TO SURVIVAL FUNCTIONS:

ti	*S(ti)*
0	1
4	0.97935
5	0.95507
6	0.9337
7	0.90507
8	0.87443
10	0.84814
11	0.82909
12	0.80714
14	0.78882
15	0.77386
16	0.76614
17	0.76044
18	0.7449
19	0.72864
21	0.7102
23	0.69041
25	0.66771
26	0.6513
27	0.63706
31	0.61739
32	0.59574
33	0.57191
34	0.54567
35	0.52487
36	0.50002
37	0.45525
39	0.4013
40	0.34192
44	0.28471
45	0.22931
48	1.94E-16
60	0

Figure 1: Starting S(ti) values of radiation only treatment group

ti	*S(ti)*
0	1
4	0.9886
5	0.96018
8	0.92157
9	0.90689
10	0.89902
11	0.88589
12	0.85688
13	0.82807
14	0.80854
15	0.77812
16	0.74479
17	0.70032
18	0.66127
19	0.6159
20	0.57275
21	0.53839
22	0.5056
23	0.47311
24	0.44401
25	0.41423
26	0.38737
27	0.36277
30	0.34022
31	0.31222
32	0.28604
33	0.26426
34	0.23257
35	0.20258
36	0.16328
39	0.12738
40	0.1026
44	0.083372
48	0.043732
60	0

Figure 2: Starting S(ti) values of radiation + chemotherapy treatment group

OUTPUT TO PROBLEM (i):

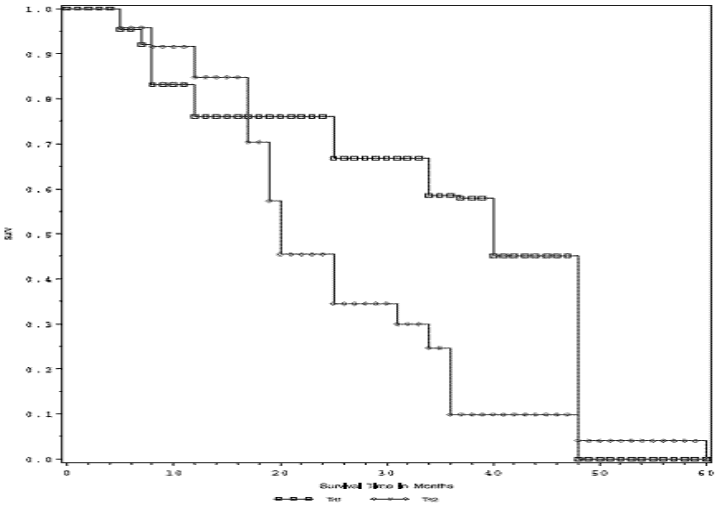

Figure3: Plot of the two survival curves by time τ_j for j=0,…,m.

Legend:
Trt1 (darker line denoted with boxes)
= radiation only

Trt2 (lighter color line denoted with diamonds)
= radiation plus chemotherapy

The survival curves cross in the beginning and once more in the end. The survival curves appear to be quite different between the two treatment groups, especially after 17 months.

Final Probabilities:

RADIATION ONLY:	
Interval	*Survival Prob.*
[0,5)	1
[5,7)	0.95365
[7,8)	0.92029
[8,12)	0.83162
[12,25)	0.76087
[25,34)	0.66833
[34,37)	0.58534
[37,40)	0.57953
[40,48)	0.45172
>=48	1

RADIATION + CHEMO:	
Interval	*Survival Prob.*
[0,5)	1
[5,8)	0.9576
[8,12)	0.9152
[12,17)	0.84762
[17,19)	0.70354
[19,20)	0.57358
[20,25)	0.4543
[25,31)	0.34493
[31,34)	0.2996
[34,36)	0.24643
[36,48)	0.09904
[48,60)	0.04082
>=60	0

OUTPUT TO PROBLEM (ii):

Two-sample Kolmogorov-Smirnov test

data: s1 and s2
D = 0.5158, p-value = 2.113e-11
alternative hypothesis: two.sided

Warning message:
cannot compute correct p-values with ties in: ks.test(s1, s2, alternative = "two.sided")

Interpretation:

The p-value is significantly of the K-S test is very small. Certainly, it's much smaller than a 0.05 level of significance.

The K-S test detects a significant difference between the two survival curves, which agrees with our previous observations of the plot in Figure3.

Computer Program:

NOTE ABOUT ORGANIZATION OF APPENDIX:

I did most of the project in SAS.

To answer the first question of interest, I did the work all in SAS. Radiation only treatment group and radiation + chemotherapy treatment group were run through my SAS program separately. I will include only one program for radiation only treatment group. Suffice it to say that the code for radiation + chemotherapy treatment group is identical to that of radiation only treatment group except for the uses the information for it's own treatment group. Then the output from each of these two programs was read into a third program, which I have also attached, where the data is combined to output the plot in Figure3.

To answer the third question of interest, I did part of the work in SAS and part in R. I need to use R to specify the probability used in the bootstrap sample. Then the Kolmogorov-Smirnov test was also performed in R.

PROGRAM FOR RADIATION ONLY TREATMENT GROUP (code for radiation + chemotherapy is nearly identical so is not included):

```
/* FINAL PROJECT */
options ls=80 ps=60 nodate nocenter nonumber;

data cancer;
 infile "data.txt" dlm='09'x missover dsd;
 input Li Ri trt;
  if trt=1;
run;

* FIND UNIQUE TIME POINTS IN BOTH Li & Ri *;
proc sort nodupkey data=cancer out=distinctLi;
 by Li;
proc sort nodupkey data=cancer out=distinctRi;
 by Ri;

* Merge to get one distinct list of time points *;
data distinctTi(keep=Ti) ;
 set distinctLi(rename=(Li=Ti)) distinctRi(rename=(Ri=Ti));
run;

proc sort nodupkey data=distinctTi; by Ti;

data distinctTi;
 set distinctTi;
     if Ti=. then delete;
     if Ti=60 then delete; * delete max=60 Ti value *;
run;

proc transpose data = distinctTi out=tposetime (rename=(Col1-Col38 = t1-t38));
run;
```

```
%global list;
data tpose;
 set tposetime(drop = _NAME_);
 array t{*} t1-t38;
 length list $150;
 list = t2;
 do i = 3 to dim(t);
    list = trim(left(list))||' '||trim(left(t{i}));
 end;
 call symput('list',trim(left(list)));
run;

%put list = &list;

proc means data=cancer noprint n;
by trt;
output out=count;
run;

%global n_trt1;
data count;
 set count;
    if STAT_ = 'N' then do;
     call symput(trim(left('n_trt1')), _FREQ_);
    end;
run;

%put n_trt1 = "&n_trt1";

data cancer_trt1 cancer_trt1_cp;
 set cancer ;
run;

*-----------------------------------------------------------------------;
%macro converges(data=,converg_criteria=,first=);
proc printto log='final1_2.log' new;
run;

%*-----------------------------------------------------*;
%macro probmass(dataset=,looper=);
%global Li Ri;
data cancer_trt1 nonempty;
 set &dataset;
 if _N_=1 then output cancer_trt1;
 else output nonempty;
run;

data cancer_trt1;
 set cancer_trt1;
 if Ri = . then Ri = 100;
 call symput('Li',left(trim(Li)));
 call symput('Ri',left(trim(Ri)));
run;

%put Li = &Li;
%put Ri = &Ri;


data distinctTi1;
```

```
 set distinctTi;
    if Ti<=&Li then delete;
    if Ti>&Ri then delete;
run;

%global num;
data distinctTi1;
 set distinctTi1 end=final;
     if final then call symput('num',left(trim(put(_N_,8.))));
run;


proc transpose data = distinctTi1 out=tposeTi(drop=_NAME_);

data cancer_trt1;
 merge cancer_trt1 tposeTi;
run;

data cancer_trt1;
 set cancer_trt1;
 format pm1 - pm&num 6.3;
 array time{*} COL1 - COL&num;
 array pm{*} pm1-pm&num;
   do i = 1 to %eval(&num);
    pm{i} = 1/%eval(&num);
   end;
rename COL1-COL&num = time1-time&num;
run;

%*------------------------------------------------------------*;
data _null_;
 %put looper = &looper;
 if &looper = '0' or &looper = 0 then do;
    put "loop = 0";
    call execute('data cancer_trt1_final; set cancer_trt1; run;');
 end; else do;
    put "loop ^= 1";
    call execute('data cancer_trt1_final; set cancer_trt1_final cancer_trt1;
run;');
 end;
run;


%*------------------------------------------------------------*;
%global CONTINUE;
Data _null_;
name = "work.nonempty";
if exist(name) then do;
   dsid=open(name);

 if attrn(dsid,'anobs') then do;

   if attrn(dsid,'any') = 1 then do;
       call symput('CONTINUE',left(trim('Y')));
   end; else do;
       call symput('CONTINUE',left(trim('N')));
   end;
 end;
 else do;
```

```
       call symput('CONTINUE',left(trim('N')));
 end;
end;
else do;
       call symput('CONTINUE',left(trim('N')));
end;
Run;
%PUT conTINUE = &CONTINUE;
%mend probmass;

%*********************************************************;

%macro run_probmass;
  %probmass(dataset=&data, looper='0');

  %do %while ( (&CONTINUE='Y' or &CONTINUE = Y) );
     %probmass(dataset=nonempty, looper='1');
  %end;
%mend run_probmass;
%run_probmass;

%*********************************************************;


data cancer_trt1_final;
 set cancer_trt1_final;
 array time{*} time1-time26;
 array pm{*} pm1-pm26;
 *new arrays*;
 array ftime{*} ftime1-ftime38 (&list);
 array tpm{*} tpm1-tpm38 (38*0);

 do j = 1 to dim(ftime);
  do i = 1 to dim(time);
    if time{i}=ftime{j} then do;
        tpm{j} + pm{i};
    end;
  end;
 end;
run;

data cancer_trt1_lastobs (drop= i time1-time26 pm1-pm26);
 set cancer_trt1_final end=last;
     if last  ;
run;

%global droplst cnt;
data cancer_trt1_lastobs(drop = Li Ri i);
 set cancer_trt1_lastobs;
  array tpm{*} tpm1-tpm38;
  array ftime{*} ftime1-ftime38;
  length droplist $300;

   do i = 1 to dim(tpm);
     tpm{i}=tpm{i}*(1/%eval(&n_trt1));
   end;

   count_discard = 0;
   do i = 1 to dim(tpm);
```

```
     if i = 1 then droplist = " ";
     if ftime{i} =. then do;
        count_discard + 1;
        droplist = trim(droplist)||' '||trim(vname(ftime{i}))||'
'||trim(vname(tpm{i}));
     end;
   end;

  call symput('droplst',trim(droplist));
  call symput('cnt',left(trim(dim(tpm)-count_discard)));
  drop droplist j count_discard;
run;

data cancer_trt1_final;
 merge cancer_trt1_final(keep=Li Ri trt)
cancer_trt1_lastobs(drop=&droplst);
 by trt;
run;


%*********************************;
%macro surv_tpm_first;
data cancer_trt1_final;
 set cancer_trt1_final;
 array s{*} s1-s&cnt;
 array tpm{*} tpm1-tpm&cnt;

 do i = 1 to dim(s);
  if i=1 then s{i}=1-tpm{i};
  else s{i}=s{i-1}-tpm{i};
 end;
 drop i;
run;
%put WENT THROUGHT SURV_TPM_FIRST MACRO;
%mend surv_tpm_first;
%*********************************;

%************************************************;
%* We already have an initial estimate of survival *;
%* so based on this survival estimate we can       *;
%* calculate the total  prob mass at each time pt. *;
%*-------------------------------------------------*;
%macro surv_tpm_notfirst;
data cancer_trt1_final;
 set cancer_trt1_final;
 array s{*} s1-s&cnt;
 array news{*} news1-news&cnt;
 array tpm{*} tpm1-tpm&cnt;

 do i = 1 to dim(s);
   s{i}=news{i};
 end;
 drop news1-news&cnt;

 do i = 1 to dim(s);
   if i=1 then tpm{i} = 1-s{i};
   else tpm{i}=s{i-1}-s{i};
 end;
 drop i;
```

```
run;
%put WENT THROUGH SURV_TPM_NOTFIRST MACRO;
%mend surv_tpm_notfirst;
%*************************************************;

%*************************;
%macro choose_surv_tpm;
%if &first='Yes' %then %do;
  %surv_tpm_first;
%end;  %else %do;
  %surv_tpm_notfirst;
%end;
%mend choose_surv_tpm;
%*************************;

%choose_surv_tpm;

%*---------------------------;
data cancer_trt1_final;
 set cancer_trt1_final;
 array ftime{*} ftime1-ftime&cnt;
 array ind{*} ind1-ind&cnt;
 do i = 1 to dim(ftime);
  if ftime{i}<=Ri and ftime{i}>Li then ind{i}=1;
  else ind{i} = 0;
 end;
run;

data cancer_trt1_final;
 set cancer_trt1_final;
 array ind{*} ind1-ind&cnt;
 array tpm{*} tpm1-tpm&cnt;
 tpm_sum=0;
 do i = 1 to dim(tpm);
  if ind{i}=1 then do;
     tpm_sum + tpm{i};
  end;
 end;
run;

data cancer_trt1_final;
 set cancer_trt1_final;
 array ind{*} ind1-ind&cnt;
 array tpm{*} tpm1-tpm&cnt;
 array d{*} d1-d&cnt (%eval(&cnt)*0);

 do i = 1 to dim(ind);
  if ind{i}=1 then do;
     d{i} + (tpm{i}/tpm_sum);
  end;
 end;
 drop i;
run;

data cancer_trt1_di(drop=  Li Ri ind1-ind&cnt);
 set cancer_trt1_final end=final;
  if final then output;
run;
```

```
data cancer_trt1_di;
 set cancer_trt1_di;
 array d{*} d1-d&cnt;
 array Y{*} Y1-Y&cnt;

 do i = 1 to dim(d);
   if i = 1 then Y{i} = &n_trt1;
   else Y{i} = Y{i-1} - d{i-1};
 end;
 drop i;
run;

data cancer_trt1_di;
 set cancer_trt1_di;
 array d{*} d1-d&cnt;
 array Y{*} Y1-Y&cnt;
 array diff{*} diff1-diff&cnt;

 do i = 1 to dim(d);
   diff{i}=1-(d{i}/Y{i});
 end;
 drop i;
run;

data cancer_trt1_di;
 set cancer_trt1_di;
 array diff{*} diff1-diff&cnt;
 array news{*} news1-news&cnt;

  do i = 1 to dim(diff);
   if i=1 then news{i}=diff{i};
   else news{i}=diff{i}*news{i-1};
  end;
 drop i;
run;

data cancer_trt1_di;
 set cancer_trt1_di;
 array s{*} s1-s&cnt;
 array news{*} news1-news&cnt;
 array change{*} change1-change&cnt;

 do i = 1 to dim(s);
   change{i}=abs(s{i}-news{i});
 end;
 drop i;
run;

data cancer_trt1_di;
 set cancer_trt1_di;
 array change{*} change1 - change&cnt;

 do i = 1 to dim(change);
   if i = 1 then max=change{i};
   else do;
     if change{i}>max then max=change{i};
   end;
 end;
 drop i;
```

```
run;

%global REPEAT;
data cancer_trt1_di;
 set cancer_trt1_di;
   if max<=&converg_criteria then do;
     call symput('REPEAT','No');
     put "MAX CONVERGES TO BE " max;
   end; else do;
     call symput('REPEAT','Yes');
     put "MAX DOES NOT CONVERGE: " max;
   end;
run;
%put REPEAT INDICATOR = &REPEAT;

data cancer_trt1_cp;
 merge cancer_trt1_cp(keep=Li Ri trt) cancer_trt1_di (keep = news1-news&cnt trt);
 by trt;
run;

%mend converges;
*--------------------------------------------*;

%macro loop_converges;

  %converges(data=cancer_trt1_cp,converg_criteria=0.000001,first='Yes');

  %let repeat_cnt = 1;
  %put repeat_cnt = &repeat_cnt;

  %do %until (&REPEAT='No' or &REPEAT=No);
     %converges(data=cancer_trt1_cp,converg_criteria=0.000001,first='No');
     %let repeat_cnt = %eval(%eval(&repeat_cnt) + 1);
     %put repeat_cnt_2 = &repeat_cnt;
  %end;

%mend loop_converges;
*--------------------------------------------*;
%global repeat_cnt;
%loop_converges;
%put NUMBER OF ITERATIONS = &repeat_cnt;

proc contents data=cancer_trt1_di;
proc export data = cancer_trt1_di outfile='cancer_trt1.txt' DBMS=TAB
replace;
```

THIRD SAS PROGRAM WHICH COMBINEDS BOTH TREATMENT GROUP OUTPUTS:

```
/* FINAL PROJECT:  FINAL3.SAS */
options ls=80 ps=60 nocenter nonumber nodate;
goptions device = PS;

* READ IN TRT1 DATASET WITH SUVIVAL CURVE INFORMATION *;
proc import datafile = 'cancer_trt1.txt' out = trt1  DBMS=TAB;


* READ IN TRT2 DATASET WITH SUVIVAL CURVE INFORMATION *;
proc import datafile = 'cancer_trt2.txt' out = trt2 DBMS=TAB;
```

```
data trt1;
 set trt1 (keep=ftime1-ftime31 news1-news31 trt);
   ftime0 = 0;
   ftime32=61;
   news0 = 1;
   news60 = 0;
run;

data trt2;
 set trt2 (keep=ftime1-ftime33 news1-news33 trt);
  ftime0 = 0;
  ftime34 = 61;
  news0 = 1;
  news34=0;
run;

data trt1;
 set trt1;
   array t{*} time1-time61;
   array ftime{*} ftime0-ftime32;
   array s{*} s1-s61 (1);
   array news{*} news0-news32;

 do j = 1 to 61;
  if j ^= 1 then s{j}=.;
  t{j}=j-1;
  do i = 1 to dim(news);
   if ftime{i} = j then do;
             s{j}=news{i};
             end;
  end;
   if (s{j} = . and j^=1) then s{j}=s{j-1};
 end;
 drop i j ftime0-ftime32 news0-news32;
run;


data trt2;
 set trt2;
   array t{*} time1-time61;
   array ftime{*} ftime0-ftime34;
   array s{*} s1-s61 (1);
   array news{*} news0-news34;

 do j = 1 to 61;
  if j ^= 1 then s{j}=.;
  t{j}=j-1;
  do i = 1 to dim(news);
   if ftime{i} = j then do;
             s{j}=news{i};
             end;
  end;
   if (s{j} = . and j^=1) then s{j}=s{j-1};
 end;
 drop i j ftime0-ftime34 news0-news34;
run;
```

```
data cancer;
 set trt1 trt2;
run;

* Shift survivals up by one time unit*;
data cancer;
 set cancer;
 array t{*} time1-time61;
 array s{*} s1-s61;
 array news{*} news1-news61;
 do i = 1 to dim(t)-1;
  news{i+1}=s{i};
 end;
 news1 = 1;  news61=0;
 drop i s1-s61;
 rename news1-news61 = s1-s61;
run;

proc transpose data=cancer out=cancer_time;
by trt;
var time1-time61;
run;

data cancer_time;
 set cancer_time;
 place=_N_;
 rename COL1=time;
 drop _NAME_;
run;

proc transpose data=cancer out=cancer_surv;
by trt;
var s1-s61;
run;

data cancer_surv;
 set cancer_surv;
  place=_N_;
  rename COL1=surv;
   drop _NAME_;
run;

data cancer (drop=place);
 merge cancer_time cancer_surv;
 by place;
run;

data cancer;
 set cancer;
 if trt=1 then Pattern=1;
 else Pattern=2;
run;

axis1 label=(h=1 f=swiss a=90) minor=(n=1);
axis2 label=(h=1 f=swiss 'Survival Time in Months') minor=(n=4);
legend1 label=none shape=symbol(4,.8) value=(f=swiss h=.8 'Trt1' 'Trt2');
proc gplot data = cancer;
plot surv*time=Pattern /legend=legend1 vaxis=axis1 haxis=axis2 ;
 symbol1 interpol=stepLJ h=1 v=square c=blue;
```

```
 symbol2 interpol=stepLJ h=1 v=diamond c=red;
run;
```

FINAL SAS & R CODE TO TEST FOR A DIFFERENCE IN SURVIVAL CURVES:

```
data cancer1;
 set cancer;
 if trt=1;
 keep surv;
run;

proc transpose data=cancer1 out=cancer1;
var surv;

data cancer1;
 set cancer1;
  rename COL1-COL61 = SURV1-SURV61;
run;

data cancer2;
 set cancer;
  if trt=2;
  keep surv;
run;

proc transpose data = cancer2 out = cancer2;
var surv;

data cancer2;
 set cancer2;
 rename COL1-COL61 = SURV1-SURV61;
run;


* CALCULATE PROB MASS AT EACH TIME POINT *;
data cancer1;
 set cancer1;
  array s{*} SURV1-SURV61;
  array p{*} pm1-pm60;

   do i = 1 to dim(s)-1;
     p{i}=s{i}-s{i+1};
   end;
  drop i;
run;

data cancer2;
 set cancer2;
  array s{*} SURV1-SURV61;
  array p{*} pm1-pm60;

   do i = 1 to dim(s)-1;
     p{i}=s{i}-s{i+1};
   end;
  drop i;
run;
```

```
proc export data=cancer1 outfile='cancer1.txt' DBMS=TAB replace;
proc export data=cancer2 outfile='cancer2.txt' DBMS=TAB replace;
```

*** READ THESE VALUES INTO R *;**

```
> cancer1<-read.table('/home/merganser/stefanis/pubh7450/final/cancer1.txt', header=TRUE)
> cancer2<-read.table('/home/merganser/stefanis/pubh7450/final/cancer2.txt', header=TRUE)
> pm1<-cancer1[63:122]
> pm2<-cancer2[63:122]
> s1<-sample(1:60,95,replace=TRUE,prob=pm1)
> s2<-sample(1:60,95,replace=TRUE,prob=pm2)
> ks.test(s1,s2,alternative="two.sided")
```