Session 8: Time dependent covariates

We will input the Stanford heart transplant data set (stanford.dat).

The SAS code is as follows:

```
options ls=132;

data stanford;
     infile 'stanford.dat' missover;
     input patid birthm birthd birthy randm randd randy transm transd
           transy lastm lastd lasty
           dead priorsurg missnum hla_a2 miss misscore reject;
           birthy=birthy+1900;
           birthdt=mdy(birthm,birthd,birthy);
           randdt= mdy(randm, randd, randy );
           transdt=mdy(transm,transd,transy);
           lastdt= mdy(lastm, lastd, lasty );
           age=((randdt-birthdt)/365.25-48);
           wait=1+(transdt-randdt);
           if patid=38 then wait=wait-.1;
           if transdt=. then do;
              rx=0;
              start=0;
              stop=1+lastdt-randdt;
              status=dead;
              output;
           end;

           else do;
              rx=0;
              start=0;
              stop=wait;
              status=0;
              output;

              rx=1;
              start=stop;
              stop=1+lastdt-randdt;
              status=dead;
              output;
           end;
           format birthdt randdt transdt lastdt date9.;
           drop birthd birthm birthy transd transm transy randd randm
                randy lastm lastd lasty;
run;
```

A partial printout of the raw data is

Obs	patid	birthm	birthd	birthy	randm	randd	randy	transm	transd	transy	lastm	lastd	lasty	dead	prisourg	missnum	hla_a2	miss	missscore	rejects
1	1	1	10	37	11	15	67	.	.	.	1	3	68	1	0	.	.	.	.	.
2	2	3	2	16	1	2	68	.	.	.	1	7	68	1	0	.	.	.	.	.
3	3	9	19	13	1	6	68	1	6	68	1	21	68	1	0	2	0	1.11	0	.
4	4	12	23	27	3	28	68	5	2	68	5	5	68	1	0	3	0	1.66	0	.
5	5	7	28	47	5	10	68	.	.	.	5	27	68	1	0	.	.	.	.	.
6	6	11	8	13	6	13	68	.	.	.	6	15	68	1	0	.	.	.	.	.
7	7	8	29	17	7	12	68	8	31	68	5	17	70	1	0	4	0	1.32	1	.
8	8	3	27	23	8	1	68	.	.	.	9	9	68	1	0	.	.	.	.	.
9	9	6	11	21	8	9	68	.	.	.	11	1	68	1	0	.	.	.	.	.
10	10	2	9	26	8	11	68	8	22	68	10	7	68	1	0	2	0	0.61	1	.
11	11	8	22	20	8	15	68	9	9	68	1	14	69	1	0	1	0	0.36	0	.
12	12	7	9	15	9	17	68	.	.	.	9	24	68	1	0	.	.	.	.	.
13	13	2	22	14	9	19	68	10	5	68	12	8	68	1	0	3	0	1.89	1	.
14	14	9	16	14	9	20	68	10	26	68	7	7	72	1	0	1	0	0.87	1	.
15	15	12	4	14	9	27	68	.	.	.	9	27	68	1	1	.	.	.	.	.
16	16	5	16	19	10	26	68	11	22	68	8	29	69	1	0	2	0	1.12	1	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

We analyze this code in sections. The first section simply inputs the data.

```
data stanford;
     infile 'stanfordch.dat' missover;
     input patid birthm birthd birthy randm randd randy transm transd
           transy lastm lastd lasty
           dead priorsurg missnum hla_a2 miss misscore reject;
```

The variables are a patient identification number, the birth day, month and year of the person, the day, month and year of randomization (i.e., study entry) the day, month and year of transplantation and the day month and year last seen at the clinic (dead or alive). There is an indicator about survival status (variable `dead`) which is 0 if the person is alive at the date last seen or 1 if the person is dead, as well as a binary indicator for prior surgery (1=Prior surgery, 0=No prior surgery), plus immune marker data (`hla_a2`), plus mismatch data (`miss`, `misscore` and `reject`).

Note also that we used the option missover to prevent SAS from searching for data in subsequent lines when data are not present in the line of input. This is *absolutely necessary* in order to read correctly some of the data lines that have missing data (although in this version of the data this is not necessary since all missing values are clearly marked with a period '.').

First we need to turn the day/month/year data into dates. We accomplish this as follows:

```
           birthy=birthy+1900;
```

turning the year of birth (which is a two-digit number) into a year after 1900. Then we use the macro `mdy(month, day, year)` to turn the three components of the date into a single date.

```
           birthdt=mdy(birthm,birthd,birthy);
           randdt= mdy(randm, randd, randy );
           transdt=mdy(transm,transd,transy);
           lastdt= mdy(lastm, lastd, lasty );
```

We also create the variables for the wait time until a heart was found and the variable for age as a function of the randomization (study entry) date minus the birth date. To change it in years after the age of 48 (as in the original analysis) we divide by 365.25 (to take into account leap years) and subtract 48.

```
           age=((randdt-birthdt)/365.25-48);
           wait=1+(transdt-randdt);
```

Patient #38, received a transplant upon entry into the study, so wait time for that patient is 0. To overcome this problem, we subtract a small value (say 0.1) to its wait time (so the patient received a transplant not exactly at time zero).

```
           if patid=38 then wait=wait-.1;
```

Now we need to figure out who got a transplant. If a person did not receive a heart then their transplant date would be missing. So one way to identify individuals without a transplant is to look for individuals with missing transplant date.

In addition, we will split the time of pre-transplant and post transplant. The information assigned to each individual pre and post-transplant will be as follows:

a. For individuals that did not receive a transplant

i. Start of time is zero. Stop of time is the day last seen minus the study entry date plus one (i.e., the first day on study is day 1 not day 0).
ii. The survival status is equivalent to the status determined by the variable `dead`.

The SAS code to accomplish this is as follows:

```
if transdt=. then do;
   rx=0;
   start=0;
   stop=1+lastdt-randdt;
   status=dead;
   output;
end;
```

Notice the command `output` in the previous code segment.

```
   output;
```

This command outputs a line for that individual in the new data set. By including this line of code, SAS will not output by default but only where an output command exists.

b. For individuals that received a transplant

i. Before transplant

1. Start time is zero, stop (end of the interval) is the duration of the waiting time (contained in variable `wait`).
2. Transplant status during this interval is zero
3. Survival status (`dead`) is 0 (i.e., alive, otherwise the subject would not have received a transplant). Note that this is the source of the possible bias. The persons that received a transplant are alive *by definition* for some time until the transplant.

```
else do;
   rx=0;
   start=0;
   stop=wait;
   status=0;
   output;
```

ii. After transplant

1. Start time is equal to the stop time in the previous section. Alternatively, we could have written

```
start=wait
```

2. Stop time is the difference between randomization and entry into the study plus one day.
3. The survival status is whatever is determined by the variable `dead`.

```
            rx=1;
            start=stop;
            stop=1+lastdt-randdt;
            status=dead;
            output;
        end;
```

The final step is to assign formats and drop unnecessary variables.

```
        format birthdt randdt transdt lastdt date9.;
        drop birthd birthm birthy transd transm transy randd randm
             randy lastm lastd lasty;
```

The new data is as follows:

```
proc print data=stanford;run;
```

```
             p
             r          m
             i m        i           b                   t
             o i h      s r         i         r         r         l                       s
         p   r s l      s e         r         a         a         a                s      t
         a d s s a   m  c j         t         n         n         s           w    t    s a
     O   t e u n _   i  o e         h         d         s         t     a     a    a    t t
     b   i a r u a   s  r c         d         d         d         d     g     i r  r    o u
     s   d d g m 2   s  e t         t         t         t         t     e     t x  t    p s

      1  1 1 0 . .  .   . . 10JAN1937 15NOV1967         . 03JAN1968 -17.1554  . 0  0   50 1
      2  2 1 0 . .  .   . . 02MAR1916 02JAN1968         . 07JAN1968   3.8357  . 0  0    6 1
      3  3 1 0 2 0 1.11 0 . 19SEP1913 06JAN1968 06JAN1968 21JAN1968   6.2971  1 0  0    1 0
      4  3 1 0 2 0 1.11 0 . 19SEP1913 06JAN1968 06JAN1968 21JAN1968   6.2971  1 1  1   16 1
      5  4 1 0 3 0 1.66 0 . 23DEC1927 28MAR1968 02MAY1968 05MAY1968  -7.7372 36 0  0   36 0
      6  4 1 0 3 0 1.66 0 . 23DEC1927 28MAR1968 02MAY1968 05MAY1968  -7.7372 36 1 36   39 1
      7  5 1 0 . .  .   . . 28JUL1947 10MAY1968         . 27MAY1968 -27.2142  . 0  0   18 1
      8  6 1 0 . .  .   . . 18NOV1913 13JUN1968         . 15JUN1968   6.5681  . 0  0    3 1
      9  7 1 0 4 0 1.32 1 . 29AUG1917 12JUL1968 31AUG1968 17MAY1970   2.8693 51 0  0   51 0
     10  7 1 0 4 0 1.32 1 . 29AUG1917 12JUL1968 31AUG1968 17MAY1970   2.8693 51 1 51  675 1
     11  8 1 0 . .  .   . . 27MAR1923 01AUG1968         . 09SEP1968  -2.6502  . 0  0   40 1
     12  9 1 0 . .  .   . . 11JUN1921 09AUG1968         . 01NOV1968  -0.8378  . 0  0   85 1
     13 10 1 0 2 0 0.61 1 . 09FEB1926 11AUG1968 22AUG1968 07OCT1968  -5.4976 12 0  0   12 0
     14 10 1 0 2 0 0.61 1 . 09FEB1926 11AUG1968 22AUG1968 07OCT1968  -5.4976 12 1 12   58 1
     15 11 1 0 1 0 0.36 0 . 22AUG1920 15AUG1968 09SEP1968 14JAN1969  -0.0192 26 0  0   26 0
     16 11 1 0 1 0 0.36 0 . 22AUG1920 15AUG1968 09SEP1968 14JAN1969  -0.0192 26 1 26  153 1
     17 12 1 0 . .  .   . . 09JUL1915 17SEP1968         . 24SEP1968   5.1937  . 0  0    8 1
      .  . . . . .  .   . .         .         .         .         .    .      . .  .    . .
      .  . . . . .  .   . .         .         .         .         .    .      . .  .    . .
      .  . . . . .  .   . .         .         .         .         .    .      . .  .    . .
```

Now we are ready to perform the Cox regression. The simplest model involving the effect of transplant is

```
proc phreg data=stanford;
     model (start stop)*status(0)=rx;
     title 'Analysis of the Stanford heart transplant data';
run;
```

Notice the new definition of the model

```
     model (start stop)*status(0)=rx;
```

in terms of start and stop times (i.e., intervals of the form (t_{j-1}, t_j) instead of a single time).

```
            Analysis of the Stanford heart transplant data

                         The PHREG Procedure

                          Model Information

              Data Set                 WORK.STANFORD
              Dependent Variable       start
              Dependent Variable       stop
              Censoring Variable       status
              Censoring Value(s)       0
              Ties Handling            BRESLOW


          Summary of the Number of Event and Censored Values

                                                 Percent
               Total       Event    Censored    Censored

                 172          75          97       56.40


                          Convergence Status

            Convergence criterion (GCONV=1E-8) satisfied.


                       Model Fit Statistics

                                 Without           With
                  Criterion     Covariates     Covariates

                  -2 LOG L         596.651        596.475
                  AIC              596.651        598.475
                  SBC              596.651        600.793
```

```
                          The PHREG Procedure

                Testing Global Null Hypothesis: BETA=0

        Test                  Chi-Square       DF     Pr > ChiSq

        Likelihood Ratio          0.1757        1         0.6751
        Score                     0.1743        1         0.6763
        Wald                      0.1742        1         0.6764


                Analysis of Maximum Likelihood Estimates

                Parameter    Standard                               Hazard
 Variable  DF    Estimate       Error  Chi-Square  Pr > ChiSq      Ratio

 rx         1     0.12567     0.30108      0.1742      0.6764      1.134
```

The interpretation of the model is that the overall impact of heart transplant on survival was not significant (if anything the hazard of death seems higher by 13.4% among patients having received transplants).

Naïve analysis of the Stanford heart transplant data

It is educational to try to perform the naïve analysis of the Stanford transplant data. For this analysis we are only interested in the final time from study entry to dead or last date seen alive and whether the person received a transplant or not.

We will input the previous data set and only keep the last line for each patient. This is accomplished as follows:

```
data naive;
     set stanford;
     by patid;
     if last.patid;
     keep patid stop rx status;
run;
```

A critical statement in the above data step is

```
     by patid;
```

which acknowledges that the data are sorted by patient id (make sure this is the case or you'll get an error message). This also has the effect of creating two temporary SAS variables that keep track whether each observation is the first in the particular patient ID (this is called `first.patid` and is 1 if the observation is the first one within the specific patient ID and zero any other time) and `last.patid`, which is one and zero in the opposite order as `first.patid`.

The other critical statement is

```
     if last.patid;
```

This is an SQL (structured query language – that is database language – statement that tells SAS to only keep those observations for which `last.patid` is one (i.e., the last observation from each patient.

We keep only the minimum amount of variables with the remaining statements. A printout of the data is as follows:

```
proc print data=naive;
     title 'Data for naive analysis of Stanford transplant data';
run;
```

Data for naive analysis of Stanford transplant data

Obs	patid	rx	stop	status
1	1	0	50	1
2	2	0	6	1
3	3	1	16	1
4	4	1	39	1
5	5	0	18	1
6	6	0	3	1
7	7	1	675	1
8	8	0	40	1
9	9	0	85	1
10	10	1	58	1
11	11	1	153	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
99	99	0	21	1
100	100	1	39	0
101	101	0	31	0
102	102	0	11	0
103	103	0	6	1

Compare this with the printout of the complete data set above. Note now that the variable stop is our time until death or censoring. Note also that there are now 103 lines of data (as many as the subjects) versus 172 in the previous analysis.

The analysis will be concluded by evoking the following command.

```
proc phreg data=naive;
    model stop*status(0)=rx;
    title 'Naive analysis of the Stanford heart transplant data';
run;
```

The output from the naïve analysis is as follows:

Naive analysis of the Stanford heart transplant data

The PHREG Procedure

Model Information

Data Set	WORK.NAIVE
Dependent Variable	stop
Censoring Variable	status
Censoring Value(s)	0
Ties Handling	BRESLOW

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
103	75	28	27.18

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	596.649	570.924
AIC	596.649	572.924
SBC	596.649	575.242

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	25.7251	1	<.0001
Score	33.0137	1	<.0001
Wald	29.1873	1	<.0001

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
rx	1	-1.31832	0.24402	29.1873	<.0001	0.268

This shows the potentially dramatically erroneous analysis that can result from not adjusting properly for the bias inherent in this type of problem.