```
%MACRO MSLT(DATA=);
%DO MA=1 %TO &NC;
  %LET COV&MA=%SCAN(&COV,&MA,' ');

  PROC SORT DATA=&DATA;
    BY &&COV&MA;
  RUN;

  DATA TEMP&MA(KEEP=ID &&COV&MA);
    SET &DATA;
    BY &&COV&MA;
    IF FIRST.&&COV&MA;
  RUN;

  DATA _NULL_;
    SET TEMP&MA END=FINAL;
    N+1;
    IF FINAL THEN CALL SYMPUT("LC&MA", N);  *** LC: LEVEL OF COVARIATES 1,2,... ***;
  RUN;
%END;
```

This section converts the list of covariate names into macro variables to be used later.

```
PROC SORT DATA=&DATA;
  BY ID AGE;
RUN;

DATA SEM;
  SET &DATA;
  BY ID AGE;
  IF FIRST.ID THEN DO;
    PREVAGE=.;
    PREVST=.;
    PREVWGT=.;
  END;
  OUTPUT;
  PREVAGE=AGE;
  PREVST=&VAR;
  PREVWGT=&WGT;
  RETAIN PREVAGE PREVST PREVWGT;
RUN;

DATA SEM1(DROP=AGE &VAR &WGT J PREVAGE PREVST PREVWGT
          RENAME=(NAGE=AGE HS=&VAR NWGT=&WGT));
  SET SEM;
  BY ID;
  IF FIRST.ID THEN DO;
    HS=&VAR;
    NAGE=AGE;
    NWGT=&WGT;
    OUTPUT;
  END;
  ELSE DO;
    IF PREVAGE+1<AGE THEN DO;
      DO J=PREVAGE+1 TO AGE;
        IF J NE AGE THEN DO;
          HS=PREVST;
          NWGT=PREVWGT+ROUND((&WGT-PREVWGT)/(AGE-PREVAGE));
        END;
        ELSE DO;
          HS=&VAR;
          NWGT=&WGT;
        END;
        NAGE=J;
        PREVWGT=NWGT;
        OUTPUT;
      END;
    END;
    ELSE IF PREVAGE+1=AGE THEN DO;
```

This section converts the input data set, which lists one interview observation per line of record, into the interval or person-year format.

```
      HS=&VAR;
      NAGE=AGE;
      NWGT=&WGT;
      OUTPUT;
    END;
  END;
RUN;

DATA PRSNYR(RENAME=(&VAR=ENDST));
  SET SEM1;
  BY ID;

  AGE=LAG(AGE);
  BEGST=LAG(&VAR);
  &WGT=LAG(&WGT);
  IF FIRST.ID THEN DO;
    AGE=0;
    BEGST=0;
    DELETE;
  END;
RUN;
```

```
DATA PS2;
  %DO MA=1 %TO &NC;
    DO &&COV&MA=1 TO &&LC&MA;
  %END;
      DO AGE=&BEG TO &END;
        CONTROL=1;
        OUTPUT;
      END;
  %DO MA=1 %TO &NC;
    END;
  %END;
RUN;

DATA PSBACK;
  SET PS2 &DATA(WHERE=(AGE>=&BEG));
  IF AGE>100 THEN AGE=100;
  IF &VAR=&NS THEN DELETE;
  IF CONTROL=. THEN CONTROL=0;
RUN;

PROC SORT;
  BY &COV AGE;
RUN;

PROC LOGISTIC DATA=PSBACK DESCENDING NOPRINT;
  CLASS &COV;
  MODEL &VAR=AGE &COV AGE*AGE / L=GLOGIT;
  WEIGHT &WGT;
  OUTPUT OUT=PREV PREDPROBS=I;
RUN;

DATA PREV2;
  SET PREV(WHERE=(CONTROL=1));
  ARRAY P{&NS};
  ARRAY IP_{&NS};
  DO I=1 TO &NS-1;
    P{I}=ROUND(IP_{I},0.0001);
    IF IP_{I}=. THEN P{I}=0;
  END;
  KEEP &COV AGE P1-P%EVAL(&NS-1);
RUN;

PROC SORT;
  BY &COV AGE;
RUN;
```

This part estimates the prevalence of health status at each ages 65-95. Smoothed prevalence estimates are necessary for life table estimation. Users need to modify the logistic regression to select the best fit model.

Users need to test for the best model here.

This macro estimates age-specific hazard rates, given the list of covariates. The algorithm used here is based on Mark Hayward's program.

```
%MACRO MODEL_hazard;
  *** CALCULATE TRANSITION RATES ***;
  DATA SURVIVE;
    %DO MA=1 %TO &NC;
      DO &&COV&MA=1 TO &&LC&MA;
    %END;
        DO AGE=&BEG TO &END;
          OUTPUT;
        END;
    %DO MA=1 %TO &NC;
      END;
    %END;
  RUN;

  PROC SORT;
    BY &COV AGE;
  RUN;

  %DO I=1 %TO &NS-1;
    DATA DTH;
      SET PRSNYR(WHERE=(BEGST=&I));
      IF BEGST=ENDST THEN EXPOS=&LOI; ELSE EXPOS=&LOI/2;
      IF AGE>=100 THEN DELETE;
      %DO A=1 %TO &NS;
        IF ENDST=&A THEN EV&A=1; ELSE EV&A=0;
      %END;
    RUN;

    DATA COV;
      %DO MA=1 %TO &NC;
        DO &&COV&MA=1 TO &&LC&MA;
      %END;
          DO AGE=&BEG TO &END;
            CNTL=1;
            EXPOS=.;
            OUTPUT;
          END;
      %DO MA=1 %TO &NC;
        END;
      %END;
    RUN;

    %DO J=1 %TO &NS;
      %IF &I NE &J %THEN %DO;
        DATA DTH2;
          SET DTH COV;
          IF CNTL=. THEN CNTL=0;
        RUN;

        PROC LIFEREG DATA=DTH2 NOPRINT;
          CLASS &COV;
          MODEL EXPOS*EV&J(0) = &COV AGE / DIST=EXPONENTIAL;
          WEIGHT &WGT;
          OUTPUT OUT=EVTW&I&J XBETA=LP&I&J CONTROL=CNTL;
        RUN;

        PROC SORT DATA=EVTW&I&J;
          BY &COV AGE;
        RUN;

        DATA SURVIVE;
          UPDATE SURVIVE EVTW&I&J(KEEP=&COV AGE LP&I&J);
          BY &COV AGE;
          H&I&J=EXP(-LP&I&J);  *** HAZARD ESTIMATE (ALLISON P.82) ***;
          DROP LP&I&J;
        RUN;
```

User needs to find the best fit model here, including selecting the interaction terms.

```
    %END;
  %END;
%END;

DATA TPTW;
  SET SURVIVE;
  %DO I=1 %TO &NS-1;
    H&I&I=0;
    %DO J=1 %TO &NS;
      IF H&I&J=. THEN H&I&J=0;
      %IF &I NE &J %THEN %DO;
        H&I&I=H&I&I+H&I&J;
        H&I&J=-H&I&J;
      %END;
    %END;
  %END;
RUN;
%MEND;

%MODEL_hazard;

DATA BSLE;
  %DO MA=1 %TO &NC;
    &&COV&MA=.;
  %END;
  AGE=.;
  STATE=.;
  TLE=.;
  ALE=.;
  DLE=.;
RUN;

%MACRO LE_2(SEX=,RACE=);
  PROC IML;
    USE PREV2(WHERE=(SEX=&SEX & RACE=&RACE));
    READ ALL VAR {%DO U=1 %TO &NS-1; P&U %END;} INTO PREV;

    *** IF USE %MODEL_HAZARD TO GET THE TRANSITION RATES ***;
    USE TPTW(WHERE=(SEX=&SEX & RACE=&RACE));
    READ ALL VAR {AGE} INTO AGE;
    READ ALL VAR {%DO Y=1 %TO &NS-1; %DO Z=1 %TO &NS; H&Y&Z %END; %END;} INTO MRATES;
    NA=NROW(MRATES);
    CALL SYMPUT('NA',LEFT(CHAR(NA[1,1])));         *** AGEGROUP=# of Age-Intervals ***;

    STATES=&NS;   *** # of Living States Plus Absorbing State ***;
    I=I(&NS);     *** Identity Matrix ***;

    *** Reshape the Observed Exposure Rates into M(x,n) Matrix for Each Age-Interval ***;
    %DO X=1 %TO &NA;
      M&X=J(&NS,&NS,0);
      DO J=1 TO &NS-1;
        DO K=1 TO &NS;
          M&X[J,K]=MRATES[&X,(((J-1)*&NS)+K)];
        END;
      END;
    %END;

    * THE FOLLOWING CALCULATES POPULTATION- AND STATUS-BASED LE AT 65-95 USING *
    * THE RADIX POPULATION. THE ALGORITHM IS BASED ON MARK HAYWARD PROGRAM.    *
    *                                                                - 02/02/08 *;

    START SBLT;
      Lx=J(&NA,STATES,0);
      * Set the Radix Value *;
      Lx[Radix_X,Radix_J]=100000;
      nLx=J(&NA,STATES,0);
      %DO X=1 %TO &NA-1;
```

This macro estimates HE at 65-95 using the 100,000 radix population, and the prevalence and transition rates estimates. The algorithm used here is also based on Mark Hayward's program. Highlighted areas indicate SAS codes that require user's modification if necessary.

```
      IF &X>=RADIX_X THEN DO;
        Lx[&X+1,]=Lx[&X,]*(I-M&X/2)*INV(I+M&X/2);
        nLx[&X,]=(Lx[&X,]+Lx[&X+1,])/2;
      END;
    %END;
    L_H=Lx[&NA,1] %DO J=2 %TO &NS-1; || Lx[&NA,&J] %END;;
    *nLx_H = L_H*INVM_H;
    nLx_H=L_H*INV(M&NA[1:&NS-1,1:&NS-1]);
    DO K=1 TO STATES-1;
      nLx[&NA,K]=nLx_H[K];
    END;
    QnLx=nLx[,1] %DO J=2 %TO &NS-1; || nLx[,&J] %END;;
    QTx=J(&NA,STATES-1,0);
    QTx[&NA,]=QnLx[&NA,];
    DO X=2 TO &NA-Radix_X+1;
      DO J=1 TO STATES-1;
        QTx[&NA+1-X,J]= QTx[&NA+1-X+1,J]+QnLx[&NA+1-X,J];
      END;
    END;
    QnLx=QnLx[,+]||QnLx;
    QTx=QTx[,+]||QTx;
    QEx=QTx[Radix_X,]/100000;
  FINISH SBLT;

  %DO Radix_X=1 %TO &NA;
    QR=J(1,STATES,0);
    %DO Radix_J=1 %TO &NS-1;
      Radix_X=&Radix_x;
      Radix_J=&Radix_J;

      RUN SBLT;

      * Retain the Information for &Radix_X. Age Interval *;
      QE = QE//(AGE[Radix_X,]||Radix_J ||QEx); *** QE HOLDS STATUS-BASED LE FOR ALL AGES ***;
      QW=J(1,&NS,0);
      QW=QEx#PREV[Radix_X,Radix_J];            *** QW WEIGHTS STATUS-BASED LE BY PREVALENCE ***;
      QR=QR+QW;                          *** QR SUMS OVER STATUS AT AGE X TO GET POP-BASED LE ***;

      FREE Lx QnLx QTx QEx;
    %END;
    QQ=QQ//QR;                           *** QQ HOLDS POP-BSED LE FOR ALL AGES ***;
  %END;
  QS=AGE||J(&NA,1,0)||QQ;        *** QS ADDS AGE & STATUS CODE (0) FOR THE QQ MATRIX ***;

  QA=QS//QE;                             *** QA HOLDS POP- & STATUS-BASED LE FOR ALL AGES ***;
  CREATE PSLE FROM QA;
  APPEND FROM QA;
  CLOSE PSLE;
QUIT;

DATA PSLE2;
  SEX=&SEX;
  RACE=&RACE;
  SET PSLE(RENAME=(COL1=AGE COL2=STATE COL3=TLE COL4=ALE COL5=DLE));
RUN;

DATA BSLE;
  SET BSLE PSLE2;
  IF AGE=. THEN DELETE;
RUN;

PROC SORT DATA=BSLE;
  BY SEX RACE AGE STATE;
RUN;
%MEND;

%DO SEX=1 %TO 2;
```

```
    %DO RACE=1 %TO 2;
      %LE_2(SEX=&SEX,RACE=&RACE);
    %END;
  %END;
%MEND;
```