```
/* SURVIVAL ANALYSIS USING SAS */
/* Anna Johansson, MEB, 2012-06-07 */

/* Print name of file on top of output */
title1 "%upcase(%sysget(SAS_EXECFILEPATH))";
title;

/* Change current directory (where graphs will be stored)*/
x 'cd Z:\SAS\SAS Seminars\seminar_2012-06-07_survival';
x 'cd C:\Users\annjoh\Desktop\SASsem';


/* READ DATA */
libname lib "C:\Users\annjoh\Desktop\SASsem";

/*
Kaplan-Meier plots
Nelson-Aalen plots
Estimating rates
Splitting on time
Poisson regression
Cox regression
Timescales
Testing non-proportional hazards
Stratified Cox
*/

/* Which variables do we have in dataset? */
proc contents data=lib.melanoma;
run;

proc freq data=lib.melanoma;
  tables status;
run;

/* Cause-specific mortality: status = 1 (death due to ca),
                              censoring events= 0 (alive), 2 (death due to other), 4 (lost to follow up)
*/

/* Kaplan-Meier estimates of S(t) */
proc lifetest data=lib.melanoma;
  time surv_mm*status(0,2,4);
run;

/* Reduce the list of displayed interval to a selected few (0,5,10,20,30)*/
proc lifetest data=lib.melanoma intervals=5,10 to 30 by 10;
  time surv_mm*status(0,2,4);
run;

/* list at risk*/
proc lifetest data=lib.melanoma intervals=5,10 to 30 by 10 atrisk;
  time surv_mm*status(0,2,4);
```

```
run;

/* plot Kaplan-Meier estimates */
proc lifetest data=lib.melanoma plots=(s) intervals=5,10 to 30 by 10 atrisk;
  time surv_mm*status(0,2,4);
run;

/* remove plotting censored values */
proc lifetest data=lib.melanoma plots=(s) intervals=5,10 to 30 by 10 censoredsymbol=none;
  time surv_yy*status(0,2,4);
run;
proc lifetest data=lib.melanoma plots=(s) intervals=60,120 to 360 by 60 censoredsymbol=none;
  time surv_mm*status(0,2,4);
run;

/* Proc lifetest can be used to obtain:
 - Kaplan-Meier estimates (method=km  - this is default)
 - Actuarial method (method=act)
 - Fleming-Harington (method=fh)
 - Breslow estimator (method=breslow)
*/

/* Actuarial method */
proc lifetest data=lib.melanoma plots=(s) method=act intervals=5 to 30 by 5 censoredsymbol=none;
  time surv_yy*status(0,2,4);
run;



/* Kaplan-Meier estimates by group */
proc lifetest data=lib.melanoma plots=(s) intervals=60,120 to 360 by 60 censoredsymbol=none;
  time surv_mm*status(0,2,4);
  strata sex;
run;

/* pointwise confidence limits (CL) and Hall-Wellner confidence bands (CB=HW).
   The STRATA=PANEL specification requests that the survival curves be displayed
   in a panel of three plots, one for each risk group.
*/
ods graphics on;
proc lifetest data=lib.melanoma plots=survival(cl cb=hw strata=panel) intervals=60,120 to 360 by 60
censoredsymbol=none;
  time surv_mm*status(0,2,4);
  strata sex;
run;
ods graphics off;



/* Nelson-Aalen estimator */
ods output productlimitestimates=NAdata;
proc lifetest data=lib.melanoma nelson ;
```

```
  time surv_mm*status(0,2,4);
  strata sex;
run;
ods output close;

/*
       Test of Equality over Strata

                                   Pr >
Test      Chi-Square      DF    Chi-Square

Log-Rank    104.5093       1      <.0001
Wilcoxon    109.4653       1      <.0001
-2Log(LR)   136.1458       1      <.0001

*/

proc gplot data = NAdata;
  title "Nelson-Aalen";
  plot cumhaz * surv_mm = sex  ;
run;
quit;


/* only plot the cum haz for timepoints where it is not zero (i.e. where there is an event) */
proc sort data=nadata (where=(cumhaz ne .));
   by sex surv_mm;
run;
proc print data=nadata(where=(sex=1));
  title "Nelson-Aalen Cum haz , sex=1";
  var sex surv_mm cumhaz failed left;
run;
proc print data=nadata(where=(sex=2));
  title "Nelson-Aalen Cum haz , sex=2";
  var sex surv_mm cumhaz failed left;
run;



title;




/*******************************************************
 RATES
*******************************************************/

/*
1. Use PROC SUMMARY to calculate the number of events and person-time at risk in each
exposure group and save this to a SAS data set (I've used a format to de
ne the grouping);
2. In DATA step, calculate the rate (events/person-time) and the corresponding CI;
3. Use PROC PRINT to print the results.
*/

proc freq data=lib.melanoma;
  tables age*agegrp / norow nocol nopercent missing;
run;

proc format;
  value agegrp
          0='0= 0-44'
                                                              1='1= 45-59'
          2='2= 60-74'
          3='3= 75+';
run;

proc freq data=lib.melanoma;
  tables agegrp / norow nocol nopercent missing;
  format agegrp agegrp.;
run;


data f;
  set lib.melanoma;

  if status=1 then dead=1;
  else dead=0;

  surv_dd= exit-dx;
  surv_yy= surv_dd/365.24;
run;

proc summary data=f nway;
  var dead surv_yy;
  class agegrp;
  output out=rates(drop=_type_ _freq_) sum=dead surv_yy;
  format agegrp agegrp.;
run;
data rates;
  set rates;
  _rate=1000*(dead/surv_yy);
  ci_low=_rate/exp(1.96*sqrt(1/dead));
  ci_high=_rate*exp(1.96*sqrt(1/dead));
run;
proc print noobs;
  title 'Table of cases, person-years, and rates per 1000 person-years';
  var agegrp dead surv_yy _rate ci_low ci_high;
run;
```

```
proc gplot data=rates;
  title "Rates per 1000 pyrs";
  plot _rate * agegrp;
run;



title;


/*******************************************
 Poisson regression
*******************************************/

data g;
  set lib.melanoma;

  if status=1 then dead=1;
  else dead=0;


  surv_dd= exit-dx;
  surv_yy= surv_dd/365.24;
  logtime=log(surv_yy);
run;

proc genmod data=g;
  title1 'Poisson regression model to estimate the effect of age';
  class agegrp;
  model dead = agegrp / error=poisson link=log offset=logtime type3;
  make 'ParameterEstimates' out=parmest;
  format agegrp agegrp.;
run;


/* Transform estimates to exp(b)*/

data parmest;
  set parmest;
  hr=exp(estimate);
  low_hr=exp(estimate-1.96*stderr);
  hi_hr=exp(estimate+1.96*stderr);
run;
proc print data=parmest label noobs;
  title2 'Estimated rate ratios and 95% CIs';
  var parameter level1 estimate stderr hr low_hr hi_hr;
  format estimate stderr hr low_hr hi_hr 7.4;
run;

/*
proc print data=g(obs=10);
  var id surv_dd surv_yy logtime;
```

```
run;
*/

/* Change reference level*/
proc genmod data=g;
  title1 'Poisson regression model to estimate the effect of age';
  class agegrp(ref='0= 0-44' param=ref) stage(ref='0' param=ref) ;
  model dead = agegrp stage  / error=poisson link=log offset=logtime type3;
  make 'ParameterEstimates' out=parmest;
  format agegrp agegrp.;
run;

/* Transform estimates to exp(b)*/

data parmest;
  set parmest;
  hr=exp(estimate);
  low_hr=exp(estimate-1.96*stderr);
  hi_hr=exp(estimate+1.96*stderr);
run;
proc print data=parmest label noobs;
  title2 'Estimated rate ratios and 95% CIs';
  var parameter level1 estimate stderr hr low_hr hi_hr;
  format estimate stderr hr low_hr hi_hr 7.4;
run;


title;



/*********************************
  Modelling timescales
*********************************/
/* Download lexis macro from:
    http://www.bendixcarstensen.com/Lexis/Lexis.sas

   Other macros for time-splitting can be found in this paper:
    http://www.epi-perspectives.com/content/5/1/7
*/


/* Splitting on time using %lexis macro */

%include "C:\Users\annjoh\Desktop\SASsem\lexis.sas";


/* splitting on time-since-entry*/
%lexis(
origin = dx,
entry = dx,
exit = exit,
fail = dead,
scale = 365.25,
data = g,
out = g_split,
breaks = %str(0,2,5 to 25 by 5),
other = %str(rename risk=t lrisk=logt),
left = fuband
) ;

/* before splitting */
proc print data=g(obs=10);
  var id dx exit surv_yy ;
run;

/* after splitting*/
proc print data=g_split(obs=20);
  var id dx exit surv_yy fuband t logt logtime;
run;


/* rates */
proc contents data=g_split;
run;

proc summary data=g_split nway;
  var dead t;
  class fuband;
  output out=rates(drop=_type_ _freq_) sum=dead t;
run;
data rates;
  set rates;
  _rate=1000*(dead/t);
  ci_low=_rate/exp(1.96*sqrt(1/dead));
  ci_high=_rate*exp(1.96*sqrt(1/dead));
run;
proc print noobs;
  title 'Table of cases, person-years, and rates per 1000 person-years';
  var fuband dead t _rate ci_low ci_high;
run;


/* plot rate by fuband */
proc gplot data=rates;
  title "Rates per ? pyrs";
  plot _rate * fuband;
run;


/* Poisson regression */

/* Change reference level*/
proc genmod data=g_split;
```

```
  title1 'Poisson regression model to estimate the effect of age';
  class agegrp(ref='0= 0-44' param=ref) stage(ref='0' param=ref) fuband(ref='0' param=ref) ;
  model dead = agegrp stage fuband / error=poisson link=log offset=logt type3;
  make 'ParameterEstimates' out=parmest;
  format agegrp agegrp.;
run;

/* Transform estimates to exp(b)*/

data parmest;
  set parmest;
  hr=exp(estimate);
  low_hr=exp(estimate-1.96*stderr);
  hi_hr=exp(estimate+1.96*stderr);
run;
proc print data=parmest label noobs;
  title2 'Estimated rate ratios and 95% CIs';
  var parameter level1 estimate stderr hr low_hr hi_hr;
  format estimate stderr hr low_hr hi_hr 7.4;
run;


title;



/* splitting on attained age*/
%lexis(
origin = birthdate,
entry = dx,
exit = exit,
fail = dead,
scale = 365.25,
data = g,
out = g_split,
breaks = %str(0 to 100 by 5),
other = %str(rename risk=t lrisk=logt),
left = ageband
) ;


/* splitting on calendar period */
%lexis(
origin = mdy(01,01,1900),
entry = dx,
exit = exit,
fail = dead,
scale = 365.25,
data = g,
out = g_split,
breaks = %str(75 to 95 by 5),
other = %str(rename risk=t lrisk=logt),
left = periodband
) ;





/*****************************************************
  Cox regression - automatically adjusts for timescale
*****************************************************/

/* Timescale: time-since-entry */
proc phreg data=lib.melanoma;
  class agegrp(ref='0= 0-44') stage(ref='0');
  model surv_mm*status(0,2,4) = agegrp stage / risklimits;
  format agegrp agegrp.;
run;

proc phreg data=g;
  class agegrp(ref='0= 0-44') stage(ref='0') ;
  model surv_dd*status(0,2,4) = agegrp stage / risklimits;
  format agegrp agegrp.;
run;

proc phreg data=g;
  class agegrp(ref='0= 0-44') stage(ref='0');
  model surv_dd*dead(0) = agegrp stage / risklimits;
  format agegrp agegrp.;
run;


/* Timescale: time-since-entry */
proc phreg data=g;
  class agegrp(ref='0= 0-44') stage(ref='0') ;
  model surv_dd*status(0,2,4) = agegrp stage / risklimits;
  format agegrp agegrp.;
run;

/* Timescale: attained age */
proc phreg data=g;
  class agegrp(ref='0= 0-44') stage(ref='0') ;
  model exitage*status(0,2,4) = agegrp stage / entrytime=dxage risklimits;
  format agegrp agegrp.;
run;

/* Timescale: calendar year */
proc phreg data=g;
  class agegrp(ref='0= 0-44') stage(ref='0') ;
  model exit*status(0,2,4) = agegrp stage / entrytime=dx risklimits;
  format agegrp agegrp.;
run;
```

```
/*****************************************************
  Non-proportional hazards
*****************************************************/

/* If the predictor satisfy the proportional hazard assumption
   then the graph of the survival function versus the survival
   time should results in a graph with parallel curves,
    similarly the graph of the log(-log(survival)) versus
   log of survival time graph should result in parallel lines
   if the predictor is proportional.
*/

/* Visual inspection of paralellism log(-log(survival))*/
proc lifetest data=lib.melanoma plot=(s, lls) noprint;
  time surv_mm*status(0,2,4);
  strata agegrp;
run;


/* from: http://sas-and-r.blogspot.se/2010/06/example-742-testing-proportionality.html*/

/* Visual inspection of Schönfeld residuals */

proc phreg data=g;
  class agegrp(ref='0= 0-44') stage(ref='0');
  model surv_dd*dead(0) = agegrp stage / risklimits;
  output out= propcheck ressch = schresage1 schresage2 schresage3 ;  /* save schönfeld residuals to
file propcheck*/
  format agegrp agegrp.;
run;

/*
Being in the intervention group would appear to increase the probability of being linked to primary
care.
But do the model assumptions hold? The output statement above makes a new data set that
contains the Schoenfeld residuals.
(Schoenfeld D. Residuals for the proportional hazards regresssion model. Biometrika, 1982,
69(1):239-241.)
One assessment of proportional hazards is based on these residuals, which ought to show no
association with
time if proportionality holds. We can plot them against the time to linkage using proc sgplot (section
5.1.1)
and adding a loess curve to assess the relationship.
*/


proc sgplot data=propcheck;
```

```
  loess x = surv_dd y = schresage1 / clm;
run;
proc sgplot data=propcheck;
  loess x = surv_dd y = schresage2 / clm;
run;
proc sgplot data=propcheck;
  loess x = surv_dd y = schresage3 / clm;
run;


/*
From the resulting plot, shown above, there is an indication of a possible problem.
One way to assess this is to include a time-varying covariate, an interaction between
the suspect predictor(s) and the event time. This can be done within proc phreg if you
split the timescale and create interactions with time.
If the interaction terms are significant, the null hypothesis of proportionality has been rejected.
*/

/* Modelling a time-dependent effect in Cox (interaction with time) */

/* splitting on time-since-entry*/
%lexis(
origin = dx,
entry = dx,
exit = exit,
fail = dead,
scale = 365.25,
data = g,
out = g_split,
breaks = %str(0,2,5 to 25 by 5),
other = %str(rename risk=t lrisk=logt),
left = fuband
) ;


ods output ParameterEstimates=parmestcox;
proc phreg data=g_split;
  class agegrp(ref='0= 0-44') stage(ref='0') fuband(ref='0') ;
  model surv_dd*dead(0) = agegrp stage agegrp*fuband/ risklimits;
  format agegrp agegrp.;
run;
ods output close;

/* Transform estimates to exp(b)*/

data parmestcox;
  set parmestcox;
  hr=exp(estimate);
  low_hr=exp(estimate-1.96*stderr);
  hi_hr=exp(estimate+1.96*stderr);
run;
proc print data=parmestcox label noobs;
```

```
  title2 'Estimated rate ratios and 95% CIs';
  var parameter classval0 classval1 estimate stderr hr low_hr hi_hr;
  format estimate stderr hr low_hr hi_hr 7.4;
run;


/*
You can construct an LR test statistic by making separate runs of PROC PHREG to fit the appropriate
models
and then taking the positive difference in the –2logL "with covariates" statistics given by PROC
PHREG in
the Fit Statistics table. The LR test statistic is easily computed as LRT = abs(L1–L2), where abs refers
to absolute value, and L1 and L2 refer to the –2logL values from PROC PHREG for the two models
being compared.
By taking the absolute value of the difference, it doesn't matter which model corresponds to L1 and
L2.
The LRT statistic has an approximate chi-square distribution with degrees of freedom equal to the
difference
in the number of parameters between the full and reduced models. After producing the LRT statistic
you can
use the PROBCHI function in the DATA step to compute a p-value for the test. Here is an example:
*/

/* Compare the two models:
    with main effect of age vs. with time-dep effect of age
*/
   data lrt_pval;
       L1= 31047.399; /* -2LogL with covariates from model 1 */
                                          L2= 34783.296; /* -2LogL with covariates from model 2 */
                                          df1=20;  /* Df model 1*/
                                          df2=6;   /* Df model 2*/
       LRT = abs(L1- L2);
       df  = abs(df1-df2);
       p_value = 1 - probchi(LRT,df);
                                          format p_value 6.4;
       run;

   proc print data=lrt_pval;
       title1 "LR test statistic and p-value";
       run;
title;



/*************************************************
 Stratified Cox
*************************************************/


proc phreg data=lib.melanoma;
  class agegrp(ref='0= 0-44') stage(ref='0');
  model surv_mm*status(0,2,4) = stage / risklimits;
  strata agegrp;
  format agegrp agegrp.;
run;


/* Stratified Cox to adjust for matching in cohort studies or to adjust for familial effects,
   but that is a totally different story....
*/




/***********************************
  END-OF-FILE
***********************************/
```