Estimating and modelling relative survival using SAS

Paul W. Dickman
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet, Stockholm, Sweden
paul.dickman@ki.se

June 2009

Downloading the SAS files

- Download `all_sas_files.zip` from `http://www.pauldickman.com/survival/sas/`
- If the files are copied to `c:\coursetemp\sas\` you won't have to change any library references.

Central exercises

1. 'Hand calculation' of life table and Kaplan-Meier estimates
5. Estimating expected survival (Ederer I and II)
7. Poisson regression, cause-specific mortality
9. Cox regression, cause-specific mortality
14. Life table estimates of relative survival
18. Period analysis
20. Modelling relative survival

SAS PROC LIFETEST

- The LIFETEST procedure can compute nonparametric estimates of the survivor function (using either the actuarial or Kaplan-Meier method) and test the equality of survival distributions across strata (e.g. using the log-rank test).
- The following code produces Kaplan-Meier estimates (and a corresponding plot) of all-cause survival for the sample of 35 patients with colon carcinoma.

```
proc lifetest data=survival.colon_sample plots=(s) nocensplot;
time surv_mm*status(0,4);
run;
```

- Note that we must specify the variable containing survival time, `surv_mm`, the status variable, `status`, and the codes in the status variable which define censored observations (0=alive and 4=lost to follow-up).
- The `plots=(s)` option requests plots of the survivor function.

- The `nocensplot` option suppresses the plot showing censoring times.
- The following code produces the corresponding actuarial estimates. The `width=12` option specifies annual intervals for the life table.

```
proc lifetest data=survival.colon_sample plots=(s)
     nocensplot method=act width=12;
time surv_mm*status(0,4);
run;
```

- We could also use `surv_yy` as the outcome and no width statement.
- Stratified estimates can be made using the `STRATA` statement.

- The following code estimates the survivor function separately for males and females and calculates several tests for differences in survival between the 2 groups (including the log-rank test).
- A subsetting `WHERE` statement is used to restrict the analysis to patients diagnosed during 1985 and later (note that 'ge' is SAS notation for 'greater than or equal to ($\geq$)').

```
proc lifetest data=survival.colon_sample plots=(s) nocensplot;
time surv_mm*status(0,4);
where yydx ge 85;
strata sex;
run;
```

- The complete SAS code for these analyses is in the file `example_lifetest.sas`.

Time splitting, tabulating rates, and Poisson regression for cause-specific mortality

- Code available in `Q7.SAS`.
- The lexis macro was written by Bendix Carstensen and available from his web site (`http://www.biostat.ku.dk/~bxc/Lexis/`) along with example and help files.
- Syntax similar to the Stata command `stsplit`.
- Code not shown for creating two extra variables; `entry` (zero for everyone) and `dead_cancer` (indicator for dead due to cancer).


```
%lexis (
data=melanoma,
out=melanoma_split,
breaks = %str( 0 to 10 by 1 ),
origin = 0,
entry = entry,
exit = surv_mm,
fail = dead_cancer,
scale = 12,
right = right,
risk = y,
lrisk = ln_y,
nint = fu
)
;
```


Tabulating mortality rates

```
proc summary data=melanoma_split nway;
var y dead_cancer;
class fu;
output out=rates(drop=_type_ _freq_) sum=y dead_cancer;
run;

data rates;
set rates;
_rate=1000*(dead_cancer/y);
ci_low=_rate/exp(1.96*sqrt(1/dead_cancer));
ci_high=_rate*exp(1.96*sqrt(1/dead_cancer));
run;

proc print;
title1 'Rates per 1000 person-years';
var fu dead_cancer y _rate ci_low ci_high;
run;
```

Output (compare with solutions for 7e)

FU	DEAD_ CANCER	Y	_RATE	CI_LOW	CI_HIGH
1	71	5257.00	13.5058	10.7029	17.0428
2	228	4857.92	46.9337	41.2203	53.4390
3	202	4235.46	47.6926	41.5489	54.7447
4	138	3711.58	37.1809	31.4673	43.9319
5	100	3265.58	30.6224	25.1720	37.2530
6	80	2864.67	27.9265	22.4309	34.7684
7	56	2524.79	22.1800	17.0692	28.8211
8	35	2190.25	15.9799	11.4734	22.2565
9	34	1886.38	18.0240	12.8786	25.2251
10	16	1583.04	10.1071	6.1919	16.4980

Poisson regression

```
ods output parameterestimates=parmest /* parameter estimates */
        type3=type3estimates;      /* Type III estimates */

proc genmod data=melanoma_split order=formatted;
title1 'Poisson regression model for cause-specific mortality';
class fu sex age yydx;
model dead_cancer = fu sex yydx age /
                  error=poisson offset=ln_y type3;
format fu fu. age age. yydx yydx.;
run;

ods output close;
```

- We save the parameter estimates to a data file and then exponentiate to get the rate ratios and CIs.


```
data parmest;
set parmest;
if df gt 0 then do;
rr=exp(estimate);
low_rr=exp(estimate-1.96*stderr);
hi_rr=exp(estimate+1.96*stderr);
end;
run;

proc print data=parmest label noobs;
title2 'Estimates for beta and relative risks (rr=exp(beta))';
    id parameter; by parameter notsorted;
    var level1 estimate stderr rr low_rr hi_rr;
    format estimate stderr rr low_rr hi_rr 6.3;
run;
```

- Results are identical to those obtained using Stata.

Cox proportional hazards model in SAS (PROC TPHREG)

- The TPHREG procedure extends the functionality of PROC PHREG by the inclusion of a CLASS statement.

  Code available in `Q9.SAS`.

```
proc tphreg data=survival.colon(where=(stage=1));
class agegrp;
model surv_mm*status(0,2,4) = sex agegrp year8594 / risklimits;
run;
```

- Note the similarity with PROC LIFETEST in the method of defining the outcome variable and censoring indicator.
- The `risklimits` option requests confidence limits for the estimated hazard ratios (i.e. $\exp(\hat{\beta})$).

An algorithm for estimating relative survival in SAS

- Relative survival is estimated using actuarial methods — we estimate the RSR at discrete points in the follow-up, generally by taking the product of interval-specific estimates over sub-intervals of the follow-up.
- The crux of the approach is to split the observation time for each individual into multiple observations, one for each subinterval of follow-up time.
  - Split using the `lexis` macro in SAS (Bendix Carstensen).
  - Ensure each record has the correct value for attained age, calendar period, and an index for follow-up interval.
  - Merge in the expected survival proportions
  - For each observation, create indicator variables for dead and censored.
  - Create life tables for each desired combination of covariates by collapsing over relevant records.
- The sub intervals can be of any length and intervals of differing lengths can be used within the same life table.

Requirements

- Individual patient data with information on survival time and vital status.
- General-population mortality rates (presented as probabilities of surviving one year).
- Note that we must have *annual probabilities of surviving*.
- It is possible to stratify by further variables such as region, social class, or race.

SEX	_YEAR	_AGE	PROB
1	1951	0	0.96429
1	1951	1	0.99639
1	1951	2	0.99783
1	1951	3	0.99842
1	1951	4	0.99882
1	1951	5	0.99893
1	1951	6	0.99913
1	1951	7	0.99905
1	1951	8	0.99920
1	1951	9	0.99931
1	1951	10	0.99940
1	1951	11	0.99939
1	1951	12	0.99920
1	1951	13	0.99925
1	1951	14	0.99914

- Since this file will be merged with the patient data file by sex, period, and age, the patient data file must contain variables with identical names (`SEX _YEAR _AGE`).
- Standard life tables (so-called cohort life tables) are generally constructed according to age at diagnosis and calendar period at diagnosis (in addition to other classification variables such as site, sex, stage, etc.).
- When estimating relative survival, we also have to keep track of attained age and attained period in order to merge in the expected probabilities of death.
- I have used the standard that those variables which are updated are prefixed with an underscore.

Sample SAS code (15 annual intervals, Ederer II method)

- I'll demonstrate the approach based on the following three individuals.
- The variable `SURV_MM` represents survival time in months and the variable `D` is the event indicator.

ID	SEX	AGE	YYDX	SURV_MM	D
2	Male	80	80	8.5	1
99	Female	77	79	31.5	1
4999	Male	80	92	46.5	0

- The code to split the data into multiple records per patient is as follows

```
%split (data=colon, out=colon,
origin = 0, exit = surv_mm,
event = d, scale = 1/12,
cuts = %str( 0 to 15 by 1 )
);
```

- The cutpoints must be specified in years (since the program assumes time at risk is given in years).
- Since survival time is specified in months, we specify `scale = 1/12`.
- Other possibilities for specifying the interval lengths are
  `cuts = %str( 0,0.5,1,2,3,4,5,10,20,30 )`
  or
  `cuts = %str( 0 to 10 by 0.5 )`.

- If we do not have a variable containing the calculated survival time, but have SAS date variables representing the date of diagnosis and date of exit then we can specify the following

```
%split (data=colon, out=colon,
entry = dx_date, exit = end_date,
event = d, scale = 1/365.25,
cuts = %str( 0 to 15 by 1 )
);
```


- After splitting the data we obtain the following

ID	SEX	AGE	YYDX	D	W	LEFT	FU	Y	LENGTH
2	Male	80	80	1	0	0	1	0.70833	1
99	Female	77	79	0	0	0	1	1.00000	1
99	Female	77	79	0	0	1	2	1.00000	1
99	Female	77	79	1	0	2	3	0.62500	1
4999	Male	80	92	0	0	0	1	1.00000	1
4999	Male	80	92	0	0	1	2	1.00000	1
4999	Male	80	92	0	0	2	3	1.00000	1
4999	Male	80	92	0	1	3	4	0.87500	1

- `LEFT` is the lower cutpoint of the interval (not necessarily an integer).
- `FU` is the index for the interval (`FU`=1,2,3,4,...)
- `Y` is the time at risk during the interval (in years).

- LENGTH is the length of the interval.
- D is the death indicator.
- W is the censoring indicator.
- Note that the variables AGE and YYDX represent the age and year at diagnosis, not the attained age and year during the interval.

- We now want to create variables for attained age and calendar year which are 'updated'.
- These are the variables by which we will merge in the expected probabilities of death, so they must have the same names and same format as the variables indexing the POPMORT file (sex, _period, and _age in this example).

```
data &individ;
set &individ;
_age=floor(age+left);
_year=floor(yydx+left);
run;
```


- This results in the following:

ID	SEX	AGE	_AGE	YYDX	_YEAR	D	W	FU	Y	LENGTH
2	Male	80	80	80	1980	1	0	1	0.70833	1
99	Female	77	77	79	1979	0	0	1	1.00000	1
99	Female	77	78	79	1980	0	0	2	1.00000	1
99	Female	77	79	79	1981	1	0	3	0.62500	1
4999	Male	80	80	92	1992	0	0	1	1.00000	1
4999	Male	80	81	92	1993	0	0	2	1.00000	1
4999	Male	80	82	92	1994	0	0	3	1.00000	1
4999	Male	80	83	92	1995	0	1	4	0.87500	1

- The next step is to merge in the expected survival proportions. Both the data file and the file of expected survival proportions are indexed by sex, _period, and _age.

```
data &individ;
length d w fu 4 y ln_y length 5;
merge &individ(in=a) &popmort(in=b);
by sex _year _age;
if a;
/* Need to adjust for interval lengths other than 1 year */
p_star=prob**length;
/* Expected number of deaths */
d_star=-log(p_star)*(y/length);
run;
```

- If the probability of surviving one year from a given date is p then the probability of surviving k years is p^k (provided k is not much larger than 1).
- k may be less than 1. For example, if $k = 0.5$ then we have $p^{0.5} = \sqrt{p}$.
- Here you can see why we need the population mortality data to be specified in the form of annual probabilities of death and we need the intervals to be specified in units of years.

- After merging in the probabilities of death (PROB) we have the following

ID	SEX	AGE	_AGE	YYDX	_YEAR	D	W	FU	Y	LENGTH	P_STAR
2	Male	80	80	80	1980	1	0	1	0.70833	1	0.88573
99	Female	77	77	79	1979	0	0	1	1.00000	1	0.94384
99	Female	77	78	79	1980	0	0	2	1.00000	1	0.93809
99	Female	77	79	79	1981	1	0	3	0.62500	1	0.93755
4999	Male	80	80	92	1992	0	0	1	1.00000	1	0.90338
4999	Male	80	81	92	1993	0	0	2	1.00000	1	0.89360
4999	Male	80	82	92	1994	0	0	3	1.00000	1	0.88628
4999	Male	80	83	92	1995	0	1	4	0.87500	1	0.87186

- The next step is to collapse the data as the first step in constructing life table estimates of survival.

```
proc summary data=&individ nway;
var d w p_star y d_star;
id range length;
class &vars fu; /* Follow-up must be the last variable in this list *
output out=&grouped(drop=_type_ rename=(_freq_=l))
               sum(d w y d_star)=d w y d_star mean(p_star)=p_star;
format &formats ;
run;
```

- We are collapsing the data so that we have one observation for each life table interval for each combination of sex, calendar period at diagnosis, and age category at diagnosis.
- Over all of the observations in each class, we sum the number of deaths (and other quantities) and take the average of the expected survival probabilities (giving the Ederer II interval-specific expected survival).

- If we collapse the observations over calendar period at time of diagnosis then the usual life table estimates are obtained.
- If we collapse over 'updated' calendar period then we obtain 'period' estimates (i.e. Brenner method).

- The previous step produced 'interval-specific' estimates. Now we calculate the cumulative estimates by multiplying the interval-specific estimates.

```
/* Construct the life table estimates */
data grouped;
retain cp cp_star cr 1;
set grouped;
if fu=1 then do;
  cp=1; cp_star=1; cr=1;
  end;
n_prime=n-w/2;  /* Effective number at risk */
p=1-d/n_prime;  /* Interval-specific observed survival */
r=p/p_star;     /* Interval-specific relative survival */
cp=cp*p;        /* Cumulative observed survival */
cp_star=cp_star*p_star; /* Cumulative expected survival */
cr=cp/cp_star;  /* Cumulative relative survival */
ns=n_prime-d;   /* Number of survivors */
run;
```


Sample life table output

Skin melanoma diagnosed in Finland 1975-1994 (follow-up to 1995)
Life table estimates of patient survival
Localised 1985-94, males, aged 75+ at diagnosis

I	N	D	W	Effective number at risk	Interval-specific observed survival	Interval-specific expected survival	Interval-specific relative survival	Cumulative relative survival
1	200	24	0	200.0	0.88000	0.88740	0.99166	0.99166
2	176	38	21	165.5	0.77039	0.88223	0.87324	0.86596
3	117	20	13	110.5	0.81900	0.87558	0.93539	0.81001
4	84	11	15	76.5	0.85621	0.86933	0.98491	0.79778
5	58	9	8	54.0	0.83333	0.86593	0.96236	0.76775
6	41	14	9	36.5	0.61644	0.84517	0.72936	0.55997
7	18	3	3	16.5	0.81818	0.82787	0.98830	0.55342
8	12	2	3	10.5	0.80952	0.80681	1.00336	0.55528
9	7	2	2	6.0	0.66667	0.80774	0.82535	0.45830
10	3	0	2	2.0	1.00000	0.81035	1.23403	0.56556

Poisson regression for excess mortality

```
proc genmod data=&grouped(where=(fu le 5)) order=formatted;
fwdlink link = log(_MEAN_-d_star);
invlink ilink= exp(_XBETA_)+d_star;
class fu sex age yydx;
model d = fu sex yydx age  / error=poisson offset=ln_y type3;
format fu fu. age age. yydx yydx.;
run;
```


Estève *et al.* full likelihood approach

```
proc nlp data=nlp_data cov=2 vardef=n;
max loglike;
parms int fu_2-fu_5 female year2 age2-age4;
theta = int+fu_2*fu2+fu_3*fu3+fu_4*fu4+fu_5*fu5+year2*year8594
        +age2*age_gr2+age3*age_gr3+age4*age_gr4+female*sex2;
loglike = d*log(-log(p_star)+exp(theta))-exp(theta)*y;
run;
```


Hakulinen–Tenkanen approach

```
proc genmod data=&grouped(where=(fu le 5)) order=formatted;
fwdlink link = log(-log(_mean_/p_star));
invlink ilink = exp(-exp(_xbeta_))*p_star;
class fu sex age yydx;
model ns/l_prime = fu sex yydx age / error=bin type3;
format fu fu. age age. yydx yydx.;
run;
```

Files available on the website

- The code in `survival.sas` produces life table estimates of relative survival stratified by sex, age, and calendar period of diagnosis. In addition, two output data sets are created (one containing grouped data and one containing individual patient data) which are used as input data sets for modelling.
- The code in `models.sas` estimates the excess mortality model using several different approaches.
- The code in `survival_period.sas` estimates survival using a period approach.
- `README.PDF` contains further details.