**Documentation for SAS survival (and related) macros**

**Table of contents**

[Introduction 2](#_Toc87194933)

[%rel\_surv life table survival methods 3](#_Toc87194934)

[Appendix to %rel\_surv: variables in the collapsed output file 7](#_Toc87194935)

[Appendix to %rel\_surv: variables in the age standardised output file 8](#_Toc87194936)

[%stpm2 Fit flexible parametric survival models 9](#_Toc87194937)

[%rcsgen Generate restricted cubic splines 16](#_Toc87194938)

[%stset prepare data file for regression survival analysis 20](#_Toc87194939)

[%predict post fit estimation (after fitting with %stpm2) 22](#_Toc87194940)

[%stpm2CM estimation of crude mortality (using fit from %stpm2) 26](#_Toc87194941)

[%stpm2CIF estimation of cumulative incidence functions (after fitting with %stpm2) 29](#_Toc87194942)

[%mrate compute mortality rate in cancer cohort 31](#_Toc87194943)

[%range create a temporary time variable 32](#_Toc87194944)

[%lexis split survival time based on supplied intervals 33](#_Toc87194945)

[%smooth create smoothed hazard functions from observed cumulative hazard data 36](#_Toc87194946)

[%hazard\_late smoothed hazard function from late entry data 37](#_Toc87194947)

## Introduction

This is the documentation for each of the SAS© macro programs in the accompanying suite. These macros are used in one or more of the exercises in the exercise folder, but are generally useful in the analysis of cancer registry survival data. Most of the macros provided here are based on the models developed by Dr. Paul Lambert (Stata command stpm2 and its prediction functions), Dr. Paul Dickman (an earlier sas version of %rel\_surv) and others acknowledged in the respective macros.

## %rel\_surv life table survival methods

**Syntax:**

%rel\_surv([non-positional parameters]);

**Comments**

The %rel\_surv macro program was developed with the help of funding from the Canadian Partnership Against Cancer (CPAC). It is based on a much earlier sas program from Paul Dickman, with additions by several others, including Larry Ellison of Statistics Canada and Enzo Coviello of the Cancer Registry, Barletta, Italy. The intention was to make a sas-callable stand-alone facility for computing relative survival using life table methods while providing the most recent estimators (Pohar-Perme net survival, and crude probabilities of death). It was a requirement of CPAC that both ICSS and Canadian (cancer site-specific) weights be available for age standardization. Since that original development, a facility for appending an arbitrary weight set has been added, as well as the ability to specify survival in terms of time from diagnosis to entry and to death/censoring). The estimation of survival probabilities uses the hazard transformation methodology exclusively.

**Notes on using dates or duration to specify time at risk.**  
*Using dates*  
Cancer registry analysts will usually have access to full date of diagnosis and death/censoring. If a design requiring late entry (period/hybrid or conditional on previous survival time) is in use, the late entry date (or start of the period window) would be specified as a date. All dates are expected to be numeric, rather than string variables formatted to appear as dates. In this instance, the parameters of origin, exit (and entry, if required) would point to numeric variables holding sas dates. The yydx parameter (year of diagnosis) should NOT be specified.

*Using duration (time in days, months, etc.)*  
For reasons of patient confidentiality, or for simplicity, it is also possible to specify time at risk using durations, for example total time survived as days from diagnosis to the death/censoring event. In this case, the yydx parameter MUST be specified, otherwise the linkage to population life tables will not function correctly (or at all). One symptom will be the message that ‘… patients did not link to life tables…’. Aside from the yydx parameter, only the exit parameter is required. This parameter points to the numeric variable holding total survival time for the subject. If a period or hybrid design is in use, or patients must first survive a minimum of time (conditional survival), then the time of entry into risk (the period window or time of late entry) would be specified with the entry parameter. Again, if time is specified in days, this would be the initial days survival, or the time between the date of diagnosis and start date of the period window.

Note that especially for the computation of weights used in the Pohar-Perme (‘*list =pohar’*) estimator, all patients are followed from the date of diagnosis. For this reason, intervals described in the ‘*intervals =* ‘ directive must start at 0.

Parameter / value pairs may be stated in any order. Those not explicitly stated may have a default value pre-assigned. These default values may be easily changed in the macro to meet local needs. The table below presents each parameter and its current default. Note that some directives can take values directly (e.g., scale =, maxage =), while others (such as the origin, entry, and exit) that define the risk periods must point to variables in the infile dataset.

**Non-positional parameters**

infile name of sas data file to be analysed. (required, no default)

patientID variable holding unique patient identifier (required, no default)

age name of variable on *infile* containing age at diagnosis in completed years (required, default = age)

sex variable containing sex value (required, default = sex)

origin variable holding date of diagnosis (required if dates are used, no default)

exit variable holding date (or time) of death or censoring, (required, no default)

entry variable holding start date of left truncation for period analysis (required for period analysis, no default)

yydx variable holding year of diagnosis (required if survival duration is used, no default)

scale scale factor to convert duration to years (required, default = 365.24)

censor censor indicator variable, with value(s) (required, no default)

intervals life table intervals to be used in analysis (required, no default)

popmort general population life table for this analysis. (required, default = life\_table)

survprob survival probability variable on popmort (required, default = prob)

mergeby life table variables for merging to popmort (required, default = \_age sex \_year)

maxage maximum age in popmort (required, default = 99)

strat stratification variable(s)

strat\_fmt formats for grouping stratification variable(s)

crude control printing of crude survival estimates

age*\_*adj control printing of age adjusted estimates

list control contents of printed reports

crude\_estimatesname of file holding crude survival results (default = grouped)

std\_estimates name of file holding results of age-standardisation (default = std\_final)

**Description of parameters**

infile Definition of input dataset, which may also specify a permanent sas library name, and may contain a ‘where’ clause  
e.g. infile = cpac.lung (where = (1992 <= year(diag) <= 1996))

patientID Patient ID must be unique. A test is carried out, and an error message generated (and the program stopped), if duplicated patient identifiers are detected.

age name of variable on infile containing age at diagnosis in completed years

sex variable containing sex value (coding must agree with life table sex variable)

origin variable containing date of diagnosis (only if dates are used). If survival duration (for example, in days survived) is specified in the exit variable, then this variable should **not** be supplied.

exit date of death or censoring, if dates are used to define survival. Otherwise, this variable holds the duration of survival

entry start date of left truncation for period analysis (or conditional survival). If survival duration is specified in the exit variable then this is the time (on the same scale) of the start of the period window. If not specified, defaults to the value in origin.

yydx year of diagnosis (required if survival duration is supplied in exit variable). DO NOT USE this parameter if dates (of diagnosis, entry, death/censoring) have been used to specify events and time at risk.

scale scale factor to convert duration in days to duration in years. For example, *scale* = 12 for survival provided in months

censor censor indicator variable, with value(s) indicating censored observations  
 eg. *censor = c(1 2)* describes censored observations have variable **c** set to either 1 or 2

intervals life table intervals to be used in analysis, specified using sas ‘do’ syntax. Intervals must start at 0 (EVEN for an analysis of conditional survival). If the interval list contains a comma, then mask (that is, hide the comma from the macro processor by using the %str() function as in the following examples:  
  
Intervals = %str(0 to .5 by 1/12, .5 to 2 by .25, 3 to 5 by 1)  
will provide monthly intervals to 6 months, then half-year intervals to 2 years, followed by annual intervals to 5 years  
  
Intervals = 1 to 10 by 1  
will cause the program to **halt** because the directive does not include an interval with a left boundary of 0

popmort general population life table for this analysis. May be a fully-specified sas library and dataset.

survprob variable on the popmort file that supplies the survival probability for the combination of covariates specified

mergeby life table variable names must be in correct order (age/sex/year) followed by any other stratifier, (e.g., SES, province).

maxage when matching to popmort, subjects will have their attained age constrained to be no more than this value. Set this parameter to the upper age limit in the popmort file, to avoid an error message (and program halt) when subjects age past that limit and cannot be matched.

strat stratification variable(s). separate life table analyses will be reported for each stratum  
 e.g. *strat* = sex stage

strat\_fmt formats for grouping stratification variables as a string of variable/format pairs. Use this option to group the stratification variables, if the required grouping has not already been done. (Formats already assigned to a stratification variable will be used to label output in printed reports)  
e.g. strat\_fmt = sex $sex. Stage $stage\_grp.

crude used to select printed output of crude estimates. Set to 1 to see all survival estimates. This parameter defaults to 0 when age standardization has been requested (ie. suppresses all printed output of crude estimates). Set to an expression that evaluates to 0 (false) or non-zero (true) that could be evaluated in a ‘where’ clause  
  
crude = right = 2 restricts printing to the intervals that end in 2

crude = right in(1 3 5) restricts printing to intervals ending in 1, 3 or 5  
  
crude = %str(mod(right,1) = 0) selects intervals with integer endpoints (note use of %str() function to mask the comma in the mod() function

age\_adj used to restrict printed output in an age-standardisation run (not required, default = 1)   
  
age\_adj = right = 2 restricts printing to the intervals that end in 2  
  
age\_adj = %str(mod(right,1) = 0) selects intervals with integer endpoints (note use of %str () function to mask the comma in the mod () function

list variables to output in the crude estimates step. allowed values are:  
ederer (Ederer 2, the default)  
pohar  
cuminc (cumulative probabilities of death)  
   
age-standardisation is only available for ederer and pohar. Stratifiers, interval label and numbers of cases are always listed. This list can also be used to specify an arbitrary list of variables that are present on the crude\_estimates dataset (see appendix below for the list of these variables)

crude\_estimates name of file with resulting crude estimates (default = grouped)

std\_estimates name of file holding results of age-standardisation (default = std\_final)

**Options for age standardization: either method (A) or (B) may be used**.

1. **Built-in standardization** (both parameters must be specified)

stnd weighting to be used for age standardization. Use ICSS () or CDN ()  
ICSS takes values in (1,2,3, 4)  
CDN takes values in the variable ccr\_fine in the Canadian weight dataset

weight\_lib name of sas library containing weights datasets for standardization (default = work) Must contain the two weighting datasets below. These dataset names are hard coded in the macro  
international weights icss.sas7bdat  
Canadian weights cdn.sas7bdat

1. **User-supplied weights** (both parameters must be specified)

Weightvar name of variable in infile that holds the user-supplied weights

Standstrata name of variable that defines strata to be combined in an age-standardisation step

Appendix to %rel\_surv: variables in the collapsed output file  
the collapsed output file created by *%rel\_surv*, specified by the ‘*crude\_estimates =* ’ (default is ‘*grouped*’) contains the following variables, any of which may be specified for reporting in the ‘*list =* ‘ directive.

|  |  |
| --- | --- |
| **Content of variable** | **variable name** |
| Alive at start | l |
| cumulative (weighted) expected survival | cp\_star\_w |
| *cumulative (weighted) observed survival* | *cp\_p* |
| Cumulative incidence death due to cancer | cgc |
| Cumulative incidence death due to other causes | cgo |
| cumulative net survival (PPE) | cr\_p |
| Cumulative expected survival | cp\_star |
| Cumulative observed survival | cp |
| Cumulative relative survival | cr |
| Deaths | d |
| Effective number at risk | l\_prime |
| Empirical excess hazard | excess |
| Expected deaths (approximate) | d\_star\_group |
| Expected number of deaths | d\_star |
| Follow-up interval | fu |
| Interval length (potential not actual) | length |
| Interval-specific expected survival | p\_star |
| Interval-specific net survival (PPE) | ns\_w |
| Interval-specific observed survival | p |
| Interval-specific relative survival | r |
| Life table interval | interval |
| ln(person-time at risk) | ln\_y |
| Lower 95% CI excess hazard | excess\_lci |
| Lower 95% CI interval net survival | lo\_rw |
| Lower 95% CI cgc | lo\_cgc |
| Lower 95% CI cgo | lo\_cgo |
| Lower 95% CI CP | lo\_cp |
| Lower 95% CI CR | lo\_cr |
| Lower 95% CI P | lo\_p |
| Lower 95% CI PPE | lo\_cr\_p |
| Lower 95% CI R | lo\_r |
| Number surviving the interval | ns |
| Person-time (years) at risk | y |
| Person-time (years) at risk during the interval (PPE) | y\_p |
| Standard error of CP | se\_cp |
| Standard error of CR | se\_cr |
| Standard error of P | se\_p |
| Standard error of R | se\_r |
| Suggested Interpretation | se\_flag |
| Suggested Interpretation (PPE) | se\_flag\_ppe |
| Upper 95% CI excess hazard | excess\_uci |
| Upper 95% CI interval net survival | hi\_rw |
| Upper 95% CI cgc | hi\_cgc |
| Upper 95% CI cgo | hi\_cgo |
| Upper 95% CI CP | hi\_cp |
| Upper 95% CI CR | hi\_cr |
| Upper 95% CI P | hi\_p |
| Upper 95% CI PPE | hi\_cr\_p |
| Upper 95% CI R | hi\_r |
| weighted expected deaths | d\_s\_weigh |
| weighted number at risk | y\_weigh |
| weighted number of deaths | d\_weigh |
| Withdrawals | w |

Appendix to %rel\_surv: variables in the age standardised output file  
the output file created by *%rel\_surv*, containing the age-standardised results specified by the ‘*std\_estimates =* ’ (default is ‘*std\_final*’) contains the following variables.

|  |  |
| --- | --- |
| **content of variable** | **variable name** |
| 95% CI Rel Survival | ci\_rel |
| Age Standardised Observed Survival | as\_obs |
| Age Standardised Rel Survival | as\_rel |
| left end of interval | left |
| lower 95% CI Rel Surv | lo\_rel |
| right end of interval | right |
| SE Obs Survival | se\_cp\_ageadj |
| SE Rel Survival | se\_cr\_ageadj |
| suggested interpretation | se\_flag |
| upper 95% CI Rel Surv | hi\_rel |

## %stpm2 Fit flexible parametric survival models

**Syntax**

%stpm2([varlist] [, other parameters]);

**Positional parameters** (these must appear before any non-positional parameters)

[varlist] a list of numeric variables to include in the model (may be blank). All variables to be included must be explicitly stated. Implied lists are not supported. For example, an implied list such as ‘age1 – age3’ must be expanded to the equivalent explicit list ‘age1 age2 age3’.

**Non-positional parameters** (may appear in any order after any positional parameter)  
parameter Description

Bhazard invokes relative survival models. The variable name passed holds the expected mortality rate (hazard) at the time of death

Bknots boundary knots for baseline hazard function (2 boundary knots)

Bknotstvc boundary knots for time-dependent effects

Constheta constrain value of theta when using Aranda-Ordaz family of link functions

df degrees of freedom for baseline hazard function

dftvc degrees of freedom for each time-dependent effect

Inittheta initial value of theta (default = 1, implying a log cumulative odds scale)

knots knot locations for baseline hazard

knotstvc knot locations for time-dependent effects

knscale scale for user-defined knots (default scale is time)

scale specifies the scale on which the survival model is to be fitted

tvc list of time varying effects

weights the variable holding individual weights for a weighted analysis

**Options** options are specified as a space-delimited string passed with the options parameter consisting of one or more options below

debug<x> turn on log output useful for solving problems when things go wrong

cure fit a cure model

eform exponentiate coefficients

lininit obtain initial values by first fitting a linear function of ln(time)

noorthog do not use orthogonal transformation of spline variables

noinit suppress constant term

rcsbaseoff do not include baseline spline variables in fitted model

**Description**

%stpm2 fits flexible parametric survival models (Royston-Parmar models). %stpm2 can be used with single- or multiple-record or single- or multiple-failure data. Survival models can be fitted on the log cumulative hazard scale, the log cumulative odds scale, the standard normal deviate (probit) scale, or on a scale defined by the value of theta using the Aranda-Ordaz family of link functions.

%stpm2 can fit relative survival models by use of the bhazard option

The first parameter to %stpm2 is a list of variables to be used in the modelling. The list may be blank, in which case the only parameters to be estimated are the constant term and the parameters of the baseline splines.

%stset is recommended to be used to prepare your data for use by %stpm2

**Description of parameters**

bhazard = varname is used when fitting relative survival (excess hazard) models. varname gives the expected mortality rate at the time of death/censoring. The scale parameter must be *hazard*. %stpm2 gives an error message when there are missing values of varname, since this usually indicates that an error has occurred when merging the expected mortality rates. (exercises q140, q230, q231, q232, q242, q251, q261, q284)

Bknots = knotslist knotslist is a two-element list of numbers specifying the boundary knots. By default, these are located at the minimum and maximum of the uncensored survival times. They are specified on the scale defined by knscale.

Bknotstvc = knotslist knotslist gives the boundary knots for any time-dependent effects. By default, these are the same as for the bknots option. They are specified on the scale defined by knscale. For example,  
  
Bknotstvc = x1 0.01 10 x2 0.01 8 specifies the boundary knots for the tvc spline for variable x1 to be 0.01 and 10, those for x2 are 0.01 and 8

Constheta = # constrains the value of theta, that is, it is treated as a known constant.

df = # specifies the degrees of freedom for the restricted cubic spline function used for the baseline function. # must be between 1 and 10, but usually a value between 1 and 5 is often sufficient. The knots option is not applicable if the df option is specified. A model with one knot (df = 1) is equivalent to fitting a Weibull survival model. Knots are placed at the following centiles of the distribution of the uncensored log survival times:

|  |  |  |
| --- | --- | --- |
| df | knots | Centile positions |
| 1 | *0 (no knots)* |  |
| 2 | 1 | 50 |
| 3 | 2 | 33 67 |
| 4 | 3 | 25 50 75 |
| 5 | 4 | 20 40 60 80 |
| 6 | 5 | 17 33 50 67 83 |
| 7 | 6 | 14 29 43 57 71 86 |
| 8 | 7 | 12.5 25 37.5 50 62.5 75 87.5 |
| 9 | 8 | 11.1 22.2 33.3 44.4 55.6 66.7 77.8 88.9 |
| 10 | 9 | 10 20 30 40 50 60 70 80 90 |

Note that these are interior knots, as there are also boundary knots placed at the minimum and maximum of the distribution of uncensored survival times. When the cure option is used df must be between 3 and 11, in which case the default location of the knots are as follows.

|  |  |  |
| --- | --- | --- |
| df | knots | Centile positions |
| 3 | 2 | 50 95 |
| 4 | 3 | 33 67 95 |
| 5 | 4 | 25 50 75 95 |
| 6 | 5 | 20 40 60 80 95 |
| 7 | 6 | 17 33 50 67 83 95 |
| 8 | 7 | 14 29 43 57 71 86 95 |
| 9 | 8 | 12.5 25 37.5 50 62.5 75 87.5 95 |
| 10 | 9 | 11.1 22.2 33.3 44.4 55.6 66.7 77.8 88.9 95 |
| 11 | 10 | 10 20 30 40 50 60 70 80 90 95 |

Dftvc = df\_list gives the degrees of freedom for time-dependent effects in tvc\_list. The potential degrees of freedom are listed under the df option. With 1 degree of freedom a linear effect of log time is fitted. If there is more than one time-dependent effect and different degrees of freedom are requested for each time-dependent effect then the following syntax applies:  
  
Dftvc = x1:3 x2:2 1  
  
This will use 3 degrees of freedom for x1, 2 degrees of freedom for x2 and 1 degree of freedom for all remaining time-dependent effects.

Inittheta = # gives an initial value for theta in the Aranda-Ordaz family of link functions.

Knots = # [# ...] specifies knot locations for the baseline distribution function, as opposed to the default locations set by df. Note that the locations of the knots are placed on the scale defined by knscale. However, the scale used by the restricted cubic spline function is always log time.

Knotstvc = knotslist knotslist defines as the locations of the interior knots for time-dependent effects. If different knots are required for different time-dependent effects the option is specified, for example, as follows:  
  
Knotstvc = x1 1 2 3 x2 1.5 3.5

Knscale = scale sets the scale on which user-defined knots are specified. Knscale = time denotes the original time scale, knscale = log the log time scale and knscale =centile specifies that the knots are taken to be centile positions in the distribution of the uncensored log survival times. The default is knscale = time. (see example Q131)

Scale = scalename specifies on which scale the survival model is to be fitted.  
 scale = hazard fits a model on the log cumulative hazard scale, i.e. the scale of ln(-ln S(t)). If no time-dependent effects are specified, the resulting model has proportional hazards. This is the default scale.  
  
Scale = odds fits a model on the log cumulative odds scale, i.e. ln((1 - S(t))/S(t)). If no time-dependent effects are specified then this is a gives a proportional odds model.  
  
Scale = normal fits a model on the normal equivalent deviate scale (i.e. a probit link for the survival function, invnorm(1 - S(t))).  
  
Scale = theta fits a model on a scale defined by the value of theta for the Aranda-Ordaz family of link functions, i.e. ln((S(t)^(-theta) - 1)/theta). Note that theta = 1 corresponds to a proportional odds model and theta = 0 to a proportional cumulative hazards model. (see Q133, Q210)

Tvc = varlist gives the name of the variables that are time-dependent. Time-dependent effects are fitted using restricted cubic splines. The degrees of freedom are specified using the dftvc option.

**Options**

cure is used when fitting cure models. It forces the cumulative (excess) hazard to be constant after the last knot. When the df option is used together with the cure option the internal knots are placed evenly according to centiles of the distribution of the uncensored log survival times except one that is placed at the 95th centile. Cure models can only be used when modelling on the log cumulative hazard scale and require the use of the *bhazard* variable as the model is only defined for an excess hazards analysis. (see Q261)

Debug<x> debug0 (default) turns off notes, verbose mode, macro print and logic listing, deletes temporary files   
debug1 - turn on log notes and warnings  
debug2 – include debug1, print log notes  
debug3 – include debug2, print macro print and logic listing, retain temporary files

Noinit specifies a model with no constant term

Noorthog suppresses orthogonal transformation of spline variables.

rcsbaseoff drops baseline spline variables from the model. With this option you will generally want to specify your baseline separately in two or more strata. See example below and example q140

eform reports the exponentiated coefficents. For models on the log cumulative hazard scale scale(hazard) this gives hazard ratios if the covariate is not-time dependent. Similarly, for models on the log cumulative odds scale this option will give odds ratios for non time-dependent effects.

lininit This obtains initial values by fitting only the first spline basis function (i.e., a linear function of log survival time). This option is seldom needed, but may be useful if default methods of obtaining initial values for the likelihood optimization routine (proc nlmixed) has not converged.

**Remarks**

Let t denote time. *%stpm2* works by first calculating the survival function after fitting a Cox proportional hazards model. The procedure is illustrated for proportional hazards models, specified by option scale = hazard). S(t) is converted to an estimate of the log cumulative hazard function Z(t) by the formula

Z(t) = ln(-ln S(t))

This estimate of Z(t) is then smoothed on ln(t) using regression splines with knots placed at certain quantiles of the distribution of t. The knot positions are chosen automatically if the spline complexity is specified by the df option, or manually by way of the knots option. (Note that the knots are placed on values of ln(t), not t.) Denote the predicted values of the log cumulative hazard function by Z\_hat(t). The density function f(t) is

f(t) = -dS(t)/dt = dS/dZ\_hat dZ\_hat/dt = S(t) exp(Z\_hat) dZ\_hat(t)/dt

dZ\_hat(t)/dt is computed from the regression coefficients of the fitted spline function. The estimated survival function is calculated as

S\_hat(t) = exp(-exp Z\_hat(t)).

The hazard function is calculated as f(t)/S\_hat(t). If varlist is specified, the baseline survival function (i.e. at zero values of the covariates) is used instead of the survival function of the raw observations. With df(1) a Weibull model is fitted.

With scale = normal, smoothing is of the Normal quantile function, invnorm(1 - S(t)), instead of the log cumulative hazard function. With df = 1 a lognormal model is fitted.

With scale = odds, smoothing is of the log odds of failure function, ln((1 - S(t))/S(t)), instead of the log cumulative hazard function. With df 1 a log-logistic model is fitted.

Estimation is performed by maximum likelihood. Optimisation uses SAS proc nlmixed which uses a version of Newton-Raphson iteration.

**SAS datasets retained in the *work.* library by *%stpm2***

|  |  |
| --- | --- |
| **name** | **purpose** |
| \_converge\_ | Convergence status of model fitting procedure |
| \_cov\_ | variance - covariance matrix of estimated parameters |
| \_dim\_ | dimensions of fit (number of observations, parameters, etc) |
| \_events\_ | analytic dataset created by *%stset* (or equivalent) |
| \_fit\_ | fit statistics (AIC, BIC, -2 log likelihood) |
| \_model\_ | description of current model |
| \_parms\_ | parameter estimates and their standard errors from the current fit |
| \_t\_<vn> | matrix used for orthogonalisation of splines generated for variable <vn>, where <vn> was listed in the *tvc* parameter |
| \_t\_bh\_ | matrix used for orthogonalisation of baseline splines (if present) |

**Examples**:

Provided that an analytic dataset created by *%stset*, exists in the work library, this command will fit a survival model with all defaults and no covariates (splines and constant term only). The model will be on the hazard scale, with 3 orthogonalized baseline splines, and will report the estimates.

%stpm2;

Fit a proportional excess hazard (PEH) model with covariates sex and age (represented by spline functions of years of age), and report exponentiated estimates of the excess mortality rate ratio. Use four splines to describe the excess baseline hazard function. The general population mortality rates have first been appended in a variable ‘rate’. Age splines are computed, and then the PEH model is fit:

%rcsgen(age\_yrs, gen = age, df = 3);  
 %stpm2(sex age1 age2 age3, df = 4, bhazard = rate, options = eform);

Fit a similar model, but allow the effect of sex to vary over followup time. This is a non-proportional excess hazards model.

%stpm2(sex age1 age2 age3,   
bhazard = rate, df=3, tvc = sex, dftvc = 3);

**Acknowledgments**

*%*stpm2 is derived from the Stata user-supplied program stpm2, written by Paul Lambert, University of Leicester, UK.

The option to fit cure models in %stpm2 was derived from the corresponding code for cure models fit with the above Stata program, written by Therese Andersson, Karolinska Institutet, Stockholm, Sweden

**References**

P. C. Lambert and P. Royston. Further development of flexible parametric models for survival analysis. Stata Journal 2009;9:265-290

C. P. Nelson, P. C. Lambert, I. B. Squire and D. R. Jones. Flexible parametric models for relative survival, with application in coronary heart disease. Statistics in Medicine 2007;26:5486-5498

P. Royston and M. K. B. Parmar. Flexible proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 2002;21:2175-2197.

P. Royston, P.C. Lambert. Flexible parametric survival analysis in Stata: Beyond the Cox model StataPress, 2011

Cure model references

## %rcsgen Generate restricted cubic splines

Syntax

%rcsgen ( varname [, non-positional parameters] );

**Positional parameters**  
  
varname the name of a variable that will be used to generate the basis splines.

**Non-positional parameters Description**

bknots the location of boundary knots

df degrees of freedom for knots

dgen stubname for generated derivatives of spline variables

fw name of variable containing weights when generating knots using the df or percentile options

gen stubname for generated spline variables

if1 use extra condition on varname when generating knots using df or percentile options

knots a numeric list of knot locations

orthog othogonalise generated spline variables

percentiles location of knots using percentiles

reverse derives the spline variables in reversed order

scalar a single value to calculate the spline basis for

set name of a sas dataset holding the variable to be used for spline generation

tmatrix use supplied matrix for orthogonalization

One (and only one) of the knots, percentiles or df options should be specified. If none are specified then the only variable created is a copy of varname.

Description

%rcsgen generates basis functions for restricted cubic splines and (optionally) their derivatives. Restricted cubic spline functions assume linearity beyond the two boundary knots. It is possible to specify knots on the original scale, as default percentiles or user specified percentiles. Orthogonalization can be performed using Gram-Schmidt orthogonalization. When orthogonalising, a matrix (tmat) is returned, which can be useful for re-generating the orthogonalised spline variables for out-of-sample predictions. In addition, a local macro variable &save\_knots is created which holds the knot points (both internal and boundary knots) on the scale of the variable specified in Varname. These saved data structures are used in exercises Q130, Q211, Q232, Q232, and Q251.

**Description of parameters**

Bknots = # # the two boundary knots when using the df option. By default these are the minimum and maximum of the varname

Df = # sets the desired degrees of freedom. The number of knots is one less than the df. Knots are placed at equally spaced centiles of the distribution of varname. For example, for df = 5 knots are placed at the 20th, 40th, 60th, 80th centiles of the distribution of varname. In addition boundary knots are placed at the maximum and minimum values of varname or those specified using the bknots = option.

dgen = stub gives a stubname for the derivatives of the restricted cubic splines variables. For example, dgen = drcs will create variables drcs1, drcs2, ....

Fw = weight gives the name of the variable containing weights when generating knots using df or percentile options. (see q130)

gen = stub gives a stubname for the generated cubic splines variables. For example, gen = rcs (the default) will create variable rcs1, rcs2, ....

if1 = condition supplies a condition (to be used in a sas ‘where’ clause) when generating the knots using the df or percentile options. For example in survival (time-to-event) data when using splines for the time scale it is common to calculate the knot locations based on the distribution of uncensored event times. The default value is if = 1 (ie, select all rows in the dataset) (see ‘05 OS lifelost (period).sas’ in the template folder)

knots = list numeric list of the location of the knots. The boundary knots are included in the list. (see q140, q232, q231, q251, q131, q211, q130)

Orthog = 1 will orthogonalize the generated spline variables using Gram-Schmidt orthogonalization. This is the default action. Turn off orthogonalization with orthog = 0.

Percentiles = list list of percentiles for the location of the knots. The boundary knots are included in the list.

reverse = 1 will force the spline variables to be derived in reversed order, treating the last knot as the first and the first knot as the last. This can be used to add a constraint to a regression model for a constant effect after the last knot. This option is selected by %stpm2 when fitting cure models. The default is reverse = 0

scalar = # will calculate the spline variables for a single value (#) and store the results in in the dataset specified by the Set parameter (or \_events\_ by default). It is useful when obtaining in or out of sample predictions in large datasets and you want to predict at a certain value of varname. (see q232, q231, q251). The scale used would be the scale that is native to the variable, but log time is always used for baseline and tvc spline variables.

Set = filename specify a sas dataset containing the variable to derive splines for, and which will hold the resulting variables defined by gen = (and optionally, dgen = ). The default dataset is the standard analytic dataset used by %stpm2, namely \_events\_. (see q130, q251, q120, q231, q232)

tmatrix = filename will orthogonalize the generated spline variables using the R matrix supplied by filename. If the dataset has n rows, and df = p, and X is the n\*p matrix of untransformed spline variables, and Q is the n\*(p+1) matrix of orthogonlized variables plus a column of ones, then (in matrix notation) X=QR (see q232, q231, q251)

**Examples**:

%rcsgen is used within %stpm2, and in many of the examples (A100, A101,… Q120, Q130 ,…)

Specify where to position the knots, based on variable x. and generate the default variables rcs1 … rcs3. Use a dataset in the lib.library, rather than the standard analytic dataset (\_events\_) in the work library.

%rcsgen( x, knots = 10 30 50 70 90, set = lib.mydata);

Alternatively, you can generate the knots positions according to the distribution of varname. In this example, the df = 3 option is used to specify that 4 knots are used, at 0th 33rd 67th and 100th centiles of age\_yrs. The new variables age1-age3 will be generated.

%rcsgen (age\_yrs, gen = age, df = 3);

SAS datasets created in the *work.* library by %rcsgen

|  |  |
| --- | --- |
| **name** | **purpose** |
| tmat | Matrix used to orthogonalise the spline variables created. |

In addition, a local macro variable &save\_knots is created which holds the knot points (both internal and boundary knots) on the scale of the variable specified in Var.

**Acknowledgements**

This command is based on the user-supplied Stata .ado program rcsgen written by Paul Lambert (paul.lambert@le.ac.uk),Mark Rutherford (mjr40@le.ac.uk) and Therese Andersson (therese.m-l.andersson@ki.se)

## %stset prepare data file for regression survival analysis

**Syntax**

%stset (set, death(), time, id [, options =, enter =]);

purpose: build the standard analytic dataset (*\_events\_*) from a user-supplied dataset.

%stset will exclude from the resulting dataset any data rows where the survival time variable is missing, or less than or equal to zero. This dataset will also be sorted by the created ID variable. If late entry time is specified (with the *enter = <varname>* directive) then any records with an exit time less than that time will also be excluded.

**Positional parameters** (all are required, and must be specified before any optional parameters):

set user’s input dataset. May include a data step ‘where’ clause, as in this example:  
  
%stset(my\_data (where=(site=’Colon’)), censor(0), years\_survived, chart\_num);

death a variable and value (in brackets) that identifies which observations represent observed (ie, non-censored) events.  
   
NOTE: This is different from how all other SAS time to event procedures (lifetest, phreg) specify events, which require the user to identify **censored** events.   
**Example:**censor(0) the variable ‘censor’ takes a value 0 for observations of the event, otherwise censored  
status(1 2) the variable *status* takes the values 1 or 2 for events to be considered non-censored

time a variable holding the event time

id a variable holding the unique identifier for each observation. *%stset* will test for duplicated ID values. If any are found they will be reported as errors, and the \_*events\_* dataset will not be produced. The ID variable may be character or numeric.

**(optional) Non-positional parameters**

Entervariable holding entry time for delayed-entry data. Cases with exit times earlier than this time will be excluded from the output dataset. This directive is used for period analysis.

Options noprint suppresses printing of the default report.

SAS datasets created in the work. library by %stset

|  |  |
| --- | --- |
| **name** | **purpose** |
| *\_*events\_ | The standard dataset required by %stpm2. Will contain all the data variables in the input dataset specified, as well as the following variables: \_t\_, \_death\_, \_study\_id\_ (and \_t0\_, if the enter = directive is specified) |

If not using %stset, the user must build a corresponding standard input data file (work.\_events\_) with one record per subject identified and **sorted** by unique ID containing the following data variables:

**Variable required contents, formatting**

\_t\_ yes Time to event/censoring. Real, positive values only.

\_death\_ yes 0 = censored at time \_t\_  
 1 = dead (experienced event) at time \_t\_

\_study\_id\_ yes unique subject ID. No duplicates within file. Numeric or string

\_t0\_ no time at which subject comes into surveillance. Real, non-negative.

covariates no numeric only, for use in %stpm2

examples of useage: all records in *lung2* dataset enter into analysis of survival time (*surv*),

*censor = 0* indicates a death event, chart is the unique patient identifier

%stset(lung2, censor(0), surv, chart);

Results in the following message printed

|  |
| --- |
| Summary of events read from lung2 dataset, saved in \_events\_ dataset |

| **cases read** | **survival <= 0** | **written to file** | **Deaths** | **Censored** | **Maximum survival time** | **Minimum survival time** |
| --- | --- | --- | --- | --- | --- | --- |
| 10,418 | . | 10,418 | 8,673 | 1,745 | 10.010 | 0.003 |

For a period analysis (with cases entering on 1 Jan 2015),

%stset(lung2, censor(0), surv, chart, enter = jan2015\_entry);

Results in the following message printed

|  |
| --- |
| Summary of events read from lung2 dataset, saved in \_events\_ dataset |

| **cases read** | **survival <= 0** | **exit before entry time** | **written to file** | **Deaths** | **Censored** | **Maximum survival time** | **Minimum survival time** | **maximum entry time** | **earliest entry** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10,418 | . | 5,851 | 4,567 | 2,822 | 1,745 | 10.010 | 0.003 | 8.994 | 0.003 |

## %predict post fit estimation (after fitting with %stpm2)

**syntax**

**%predict** (var, measure, at=, hdiff1=, hdiff2=, hrdenom =, hrnum =, sdiff1=, sdiff2 =, tcond =, ifP =, per =, timevar =, cent =, uncured =, options = )

purpose: post-fit estimation of survival measures. Currently, the following measures are supported: hazard, hazard difference, hazard ratio, survival, survival difference, failure, lifelost, meansurv, cumulative hazard, xb, dxb, xbnobaseline, deviance, martingale.

The following table lists the excercises that make use of this macro

|  |  |
| --- | --- |
|  |  |
| measure | **Exercise or template where %predict is used to estimate this measure** |
| hazard | q239 q232 q231 q132 q133 q131 |
| Hazard difference | q230 q232 q132 |
| Hazard ratio | q230 q232 q231 q132 q133 |
| Survival | A102 A104 |
|  | q242 q230 q231 q232 q132 q261 q133 q131 |
| Survival difference | q230 q232 q132 |
| Failure | q251 |
| Lifelost | Q284 |
| Cure | q261 |
| meansurv | A100 A101 A102 A104 A105 q242 |
| martingale | q231 |
|  |  |

**Positional parameters**

var (required) name of variable to be generated by prediction equation for requested measure. Expands to   
<var> predicted value  
<var>\_lci lower 95% confidence interval (if the CI option is specified)  
<var>\_uci upper 95% confidence interval

measure one of the following:  
hazard, survival, failure, cure, lifelost, meansurv, cumhazard, xb, dxb, xbnobaseline, deviance, martingale. The resulting estimates will be appended to the \_events\_ dataset. This parameter is not required if requesting one of the following:  
hazard difference, hazard ratio, survival difference

**non-positional parameters**

at covariate pattern for out of sample estimation of survival measure. Any covariates in the covariate list for the most recently fitted model may appear. Covariate patterns are specified by a set of at least one <covariate>:<value> pair. The keyword *zero* may be used to hold all remaining covariates at the value zero. The form <covariate>:. (the colon is followed by a period) specifies that the current value of the covariate be used in prediction on a subject-by-subject basis. If not present, then all estimation takes place with all covariates evaluated at their observed level.

Uncured 1 used with the survival measure to predict the survival curve in ‘uncured’. Used after fitting a cure model. The default is 0.

Cent estimate the centile of survival measure. Measure must be survival. Can be combined with uncured to estimate median survival of uncured from a cure model. Specify a number from 1 to 99. The default is 0 (do not estimate a centile).

hdiff1 used to specify a covariate pattern for a difference in hazard functions. Covariates and values are specified as in the at= parameter. For example, to predict the difference in hazard between two values of sex (coded 0 for males and 1 for females), use  
hdiff1 = sex:0, hdiff2 = sex:1  
if there are other covariates in the model, use the *zero* option to set their values to the reference level zero, as in  
hdiff1 = sex:0 zero, hdiff2 = sex:1 zero

hdiff2 used to specify the covariate pattern for a comparison hazard function. If hdiff2 is not specified, then all covariates will be set to **zero** for a comparison covariate pattern.

hrnum, hrdenom these parameters define the covariate patterns for the numerator and denominator of a hazard ratio to be estimated. If hrnum is specified, and hrdenom is not specified, the default value of **zero** will be used for hrdenom.

sdiff1, Sdiff2 similarly to hazard difference or hazard ratio estimation, these parameters define the covariate patterns for the difference of a pair of survival functions.

ifp a sas *where* clause to be applied in order to restrict estimation to a subset of the analytic file

per a numeric value (1) used to scale the display of an estimated hazard function. Hazard values predicted with per=1000 can be interpreted as mortality rates / 1,000 person-years (assuming that the time scale in the original model is years of follow-up)

tcond a non-negative integer (default = 0). Use for computation of conditional life lost (years lost after tcond years initial survival), or for conditional survival or meansurv estimates of survival, conditional on an initial survival of tcond units on the scale of the time variable (years, months …) specified in %stset.

timevar an alternate time variable to be used, instead of the default (*\_t\_*). This allows post-fit estimation to be restricted to a subset of the full dataset, or over a limited (or expanded) time frame. For large datasets, the use of a *timevar* specification is recommended, rather than carrying out estimation on all observations in the dataset.

Extra options available for the lifelost measure:

stub Result of estimation of observed and expected mean survival will be stored in variables with this as a prefix (a stub). The default value is surv. For example,  
stub = mean will change the default expected mean survival variable from survexp to meanexp

diagage (required) age used in matching to life table for survival probabilities. This variable must exist on the \_events\_ dataset

diagyear (required) year of diagnosis for matching to life table. This variable must exist on the \_events\_ dataset

survprob name of variable (default is prob) in life table that holds survival probability for the matching diagage, diagyear and sex

maxage life table survival estimates for follow-up at ages greater than this value will be replaced with estimates for this age (default is 99)

maxyear life table survival estimates for follow-up at years greater than this value will be replaced with estimates for this year (default is 2050)

by if grpd option in effect, mean survival expectations will be grouped by unique combinations of the variables passed, rather than based on individual covariate values

grpd/nogrpd do not compute mean expected survival grouped by the variables specified in the by parameter (default is nogrpd)

using name of population life table file

mergeby specification of variable names on the using file to match with the values specified in diagage, sex and diagyear in the case file. These mergeby variable must appear in the order age/sex/year. Other covariates that are present in both the case file and the using file may be appended after the age/sex/year variables. Examples would be: race, state, SES. The macro will test if the mergeby variables uniquely determine rows in the using file, and notify (and halt) when they do not.

nodes a positive integer (default = 40). Used in Guassian quadrature computations

tinf a positive integer (default = 50). Expected and observed (modelled) survival will be calculated up to the minimum of diagage + tinf and maxage

**options**

ci request 95% confidence intervals (available for the following measures: hazard, hazard difference, hazard ratio, survival, survival difference, failure, lifelost, meansurv)

debug<x> where <x> is 0 (the default) 1, 2, or 3, as described for %stpm2

**Considerations when performing lifelost estimation**:   
Be careful when requesting confidence limits for lifelost estimates for large datasets, as Proc IML may not be able to assign sufficient memory. One strategy is to partition the dataset into discrete blocks of no more than 10,000 rows. The alternative strategy is to temporarily replace the default dataset (\_events\_) with one row for each unique covariate pattern for which an estimate is required.   
  
The following example could be used to prepare an analysis of time trends (by year of diagnosis) in future life years lost, for selected ages (the fitted model would contain sex, year of diagnosis and age as covariates. Age and year of diagnosis could be parameterized with splines, using %rcsgen, but raw age in years and year of diagnosis would have to be retained on the \_events\_ dataset):

\* this works best if only the cases of interest are included in the \_events\_ dataset;

\* make copy of \_events\_: you may need it later in your program;

**data** events\_save; set \_events\_; **run**;

\* select one subject for each covariate pattern of interest;

**proc** **sort** data = \_events\_   
 (where = (age\_yrs in(**50** **60** **70** **80**))) nodupkey;   
by age\_yrs yr\_diag female; **run**;

\* all %predict functions require the \_events\_ file to be sorted by \_study\_ID\_;

**proc** **sort** data = \_events\_; by \_study\_id\_; **run**;

\* future years of life lost (individual predictions);

%***predict*** (ll, lifelost, mergeby = \_age sex \_year, diagage = age\_yrs, diagyear = yr\_diag,

stub = surv, options = ci,

nodes = **40**, tinf = **80**, using = pop.medium, maxyear = **2060** );

The following variables will be appended to each record in the \_*events\_* dataset.  
**variable name content**  
survobs predicted ‘all causes’ mean survival time. Will be replaced by <stub>obs  
survexp population expected mean survival time. Will be replaced by <stub>exp  
<var> life years lost (survexp – survobs)  
\_lcl, \_ucl lower and upper 95% confidence limits for <var> and survobs (if requested)

## **%stpm2CM** estimation of crude mortality (using fit from %stpm2)

**Syntax  
  
%stpm2CM(<var> [, options]);**

**Description**

*%stpm2cm* calculates mortality probabilities (due to cancer and due to all other causes) after fitting a relative survival model with *%stpm2***.** These probabilities are known as crude probabilities of death, or in the competing risks literature they are known as cumulative incidence functions. Note that *%stpm2cm* performs prediction for an individual with a particular covariate pattern (specified with the at parameter). Specification of age, sex and calendar year of diagnosis is also required, so that the correct population values can be merged in from the population mortality file.  
  
The cumulative incidence and other functions for cancer (<var>\_d) and other causes (<var>\_o) are appended to the analytic datset (\_events\_). Names of variables are built from the value specified in positional parameter var.

Excercise q251.sas provide an example of the use of this macro

**positional parameters:**

var the stub name for variables to be estimated.

**Non-positional parameters:**

at specifies covariate pattern for prediction

mergeby specifies the variables to merge with the population mortality file.

diagage age at diagnosis

diagyear year of diagnosis

sex gives coding for sex required in population mortality file

maxage maxium age in popmort file

nobs specifies the number of observations (of time) to predict for (default 1000). Observations are evenly spread between the minimum and maximum value of follow-up time.

ci calculate confidence intervals

maxt the maximum value of follow up time

stub the stub name for calculation of new variables

tcond define an initial length of survival (default = 0).

tgen name of variable for generated follow-up time.

mergegen other merge variables required for the population mortality file

**Description of parameters**

var the stub name for variables to be estimated. The following variables are created:  
<var>\_d - crude probability of death due to disease,   
<var>\_o - crude probability of death due to other causes,   
<var>\_all - probability of death (all causes),   
<var>\_lambda - excess mortality rate,   
<var>\_St\_star - Expected survival,   
<var>\_s\_all - overall survival.

at specifies covariate pattern for prediction, in the form of <variable>:<value> pairs. All covariates in the model must be specified (zero option is not implemented). For example at = age:60 sex:1 requests estimation for a female (if sex is coded 0 for males, 1 for females) diagnosed at age 60.

mergeby specifies the variables to merge with the population mortality file. The default value is \_age sex \_year

using the name of the population mortality file (default: popmort). May be a two-level SAS name that includes a library name (eg, data.popmort)

popprob the name of the variable on the using file that holds the population death probability (default: prob)

diagage the value of age at diagnosis (defalt: 50). Note that this must be specified even if age has been modelled as a categorical covariate

diagyear the value of year at diagnosis (default: 2000)

sex the value for sex to be matched in using fie. The type of the value (character or numeric) must match that of the sex variable in the using file

maxage maxium age in using file (default: 99)

maxyear maximum year available in the using file (default: 2015)

nobs the number of observations (of time) to predict (default: 1000). Observations are evenly spread between 0 and maximum value of follow-up time (maxt).

maxt the maximum value of follow up time in years (default: 10)

tcond CIF estimation is carried out conditional on an initial survival of this number of years (default = 0)

tgen name of variable for generated follow-up time. The default name (<var>\_t) is generated from the value specified by the positional parameter *var*.

mergegen a list of <variable>:<value> pairs that are required for the merge to the using file. This is used when there are additional variables in the using file. For example, region or socio-economic group. An example of the use of this parameter to facilitate merging on region = 1 and ses = 5 would be   
mergegen = region:1 ses:5

options ci compute confidence intervals for <var>\_d and <var>\_o

debug<x> as in %stpm2 default is debug0

**Example**

Fit a relative survival model using age as categorical variable. In this example, agegrp1 (35-45) would be the reference category for age group.  
%stpm2(agegrp2 agegrp3 agegrp4, bhazard = rate, df = 5,   
tvc= agegrp2 agegrp3 agegrp4, dftvc = 3);

Predict death due to cancer (cmF\_d) other causes (cmF\_o) for reference age group. Note that sex is not specified in model, but is required to facilitate the merge to population mortality data. The diagnosis age of 40 is chosen as the mid-point of the reference age category. Prediction is for a subject diagnosed in 1985.

%stpm2cm(cmF, at = agegrp2:0 agegrp3:0 agegrp4:0,   
mergeby = \_year sex \_age, diagage = 40, diagyear = 1985, sex = 1);

Similar to above, but now estimate for patients who have already survived one year after diagnosis

%stpm2cm(cmF, at = agegrp2:0 agegrp3:0 agegrp4:0, tcond = 1,   
mergeby = \_year sex \_age, diagage = 40, diagyear = 1985, sex = 1);

**Acknowledgements:**%stpm2CM is derived from the user-supplied Stata program stpm2CM written by Paul Lambert

## **%stpm2CIF** estimation of cumulative incidence functions (after fitting with %stpm2)

**Syntax**

%stpm2cif(cause1 … cause<x>, [options);

**Purpose**

%stpm2cif can be used after %stpm2 to obtain cumulative incidence functions for up to 10 causes of death.

**Description of required dataset**  
Dataset must have a separate row for each potential outcome for each subject. (Note that each row must have a unique subject ID). An indicator covariate must be assigned to each separate cause. For example, 3 potential causes of death (cardiovascular, cancer, other) might be represented as 3 distinct variables such as cardio, canc, oth, taking values 0 or 1 depending on whether the row describes the outcome for that endpoint. The \_t\_ variable will be the same value for all rows for a single subject, and the event (\_death\_) variable will take the value 1 if the patient experienced the endpoint described by that row, otherwise 0. If the patient reached the end of follow-up without experiencing any of the endpoints, the \_death\_ variable would be 0 for all rows.  
  
Cumulative incidence and other functions will be added to the rows of the analytic file (\_events\_). Names of variables are built from the names specified in positional parameter var. Variables present will depend on options ci, contmort, conthaz, hazard. The time variable timename will always be present, as well as the cumulative incidence functions cif\_vn, with suffixes vn from the list given in the positional parameter.

Exercise q140 provides an example of the use of this macro

**Positional parameters**:

cause1...cause<n> a list of <n> names to use when saving requested results

**Non-positional parameters:**

at covariate patterns for each cause

obs specifies the number of observations (of time) to predict for

ci calculates confidence intervals for cumulative incidence function

maxt the maximum value of follow up time

timename name of created time variable

hazard predict cause-specific hazard function for each cause

contmort predict the relative contribution to total mortality

conthaz predict the relative contribution to hazard

**Description of parameters**

at covariate patterns for each cause (vn:# [vn:# ..]) requests (in the style of at = in %predict) that the covariates specified by the listed variable name(s) vn be set to # when predicting the cumulative incidence functions for each cause. It is compulsory to specify cause1 and cause2. Note that any covariates in the model not specified are set to zero.

obs specifies the number of observations (of time) to predict (default: 1000). Observations are evenly spread between 0 and maximum value of follow-up time (maxt). Note that a trapezoid rule is used for the numerical integration to estimate the CIF. If the number of observations is too small, this will not give an accurate result.

maxt the maximum value of follow up time. If not specified, the maximum non-censored event time in the *\_events\_* file is used.

timename the name of time variable generated during predictions for the cumulative incidence functions (default: new\_t) This is the time variable that is used when plotting the cumulative incidence functions and the cause-specific hazard functions.

hazard predicts cause-specific hazard function for each cause. Variables created are h\_<newvar>

contmort predicts the relative contribution to total mortality for each cause. Variables created are contmort\_<newvar>

conthaz predicts the relative contribution to hazard. Variables created are conthaz\_newvar

options ci calculates a 95% confidence interval for each cumulative incidence function and stores the limits in CIF\_<vn>\_lci and CIF\_<vn>\_uci, where names <vn> are generated from the list given in newvar

## %mrate compute mortality rate in cancer cohort

**syntax**

%mrate(in=, out =, byvar =, eventvar =, eventlist =, timevar =,

Per = , level = );

**purpose**

calculate event rates from censored survival data. Input dataset requires an event variable, specifying whether the observation is to be counted in the numerator of the event rate and person-time, specifying the contribution of this event to the denominator of the event rate. Mortality rates computed in this way represent the rate within the cancer cohort, as opposed to a rate based on a population denominator.

This macro was written to help with calculations needed in some of the examples in the Albuquerque workshop. It is not used in flexible parametric survival modelling

All parameters are non-positional

**required parameters**:

in dataset with observations

byvar classification variable

eventvar defines events to count

eventlist values of eventvar to consider as non-censored events

timevar time variable (years)

**optional parameters**:

out dataset to hold computed rates (default: \_rates\_)

options noprint do not print a report of computed rates

per divisor for person-years computation. default is 1

level level for confidence intervals. default is 95% confidence intervals.

## %range create a temporary time variable

**Syntax**

%range(var, start, end, nobs);

**Purpose**

The %predict macro will estimate the measure(s) requested for each row in the analytic dataset. For some computations (for example, the calculation of confidence intervals) this can be computationally intensive, and for large datasets, also greatly redundant. The %range macro creates a new time variable on the analytic dataset that can be used as the time variable in calculations. Nobs time points will be created, equally spaced between the start and end times.  
  
**All parameters to this macro are positional**

Var the name of the new time variable to create

Start the first time in the series

End the last time in the series

Nobs the number of time points to use. Must not be more than the number of rows in the analytic dataset

Example:

%range(temptime, 0, 5, 201);

Will create the new variable temptime, with 201 values from 0 to 5 in the analytic dataset.

Use of %range appears in the following exercises:   
A100  
A101  
A102  
A104  
A105  
q133  
q132  
q131  
q230  
q231  
q232  
q242  
q251

## %lexis split survival time based on supplied intervals

Syntax

%lexis([non-positional parameters]);

**Purpose**

Split time at risk for individual subjects into records representing the contribution of each subject to each interval defined. %lexis is used in %rel\_surv.  
  
**referenced in the following exercises**.

q121, q112m q120, q111, q130

**Non-positional parameters**

data name of dataset with original data

out name of output dataset

entry date (or time) of entry into surveillance

exit date of exit (or total survival time)

fail exit status

breaks interval cut points on the transformed scale

cens code for censoring (may be a variable)

scale factor to transform the time scale

origin date (or time) of origin on the transformed scale

risk variable receiving the risk time

lrisk variable receiving the log(risk time)

left variable receiving left endpoint of interval

other other dataset statements to be used

disc name of dataset holding discarded observations

Variables for making life-tables and other housekeeping. These will only appear in the output dataset if given here. The existence of these arguments are tested in the macro so they cannot have names that are also logical operators such as: or, and, eq, ne, le, lt, gt

right variable receiving right endpoint of interval

lint variable receiving interval length

os\_left variable receiving left endpoint of interval (original scale)

os\_right variable receiving right endpoint of interval (original scale)

os\_lint variable receiving interval length (original scale)

cint varriable receiving censoring indicator

nint variable receiving index of follow-up interval

st flag to indicate if subject is at risk in this interval

closed determine which end of the interval is closed

**Description of parameters**

data filename of data set with original data, defaults to \_last\_

out output dataset. defaults to dataset specified by data parameter.

entry variable holding the entry date (default entry = entry)

exit variable holding the exit date (default exit = exit)

fail variable holding the exit status (default fail = fail). If any of the entry, exit or fail variables are missing the subject is discarded from the computations.

breaks specification of the cut points on the transformed scale. Syntax as for a do statement. the specification breaks = failures generates breaks at each distinct failure time

cens Code for censoring (may be a variable) (default is cens = 0)

scale Factor to transform from the scale of entry and exit to the scale where breaks and risk are given (default is scale = 1)

origin origin of the transformed scale (default is origin = 0)

risk variable receiving the risk time (default risk = risk)

lrisk variable receiving the log(risk time) (default lrisk = lrisk)

left variable receiving left endpoint of interval (default left = left)

other other dataset statements to be used such as:   
 %str( format var ddmmyy10. ; )  
 %str( label risk ="P-years" ; )

disc dataset holding discarded observations (default disc = discrd)

right variable receiving right endpoint of interval

lint variable receiving interval length

os\_left variable receiving left endpoint of interval

os\_right variable receiving right endpoint of interval

os\_lint variable receiving interval length

cint variable receiving censoring indicator for the current input record

nint variable receiving index of follow-up interval

st logical variable receiving status of subject in this interval (0 subject is does not contribute to this interval, 1 subject should contribute to the interval). Defaults to \_st. For most applications, subject contributions to interval person-years at risk should only be included if \_st = 1.

closed determines which end of the interval is closed (default closed = right) life table intervals used by %rel\_surv are open on right (closed = left)

The lexis macro was originally developed by Bendix Carstensen and Paul Dickman, and updated for use in this macro suite

## %smooth create smoothed hazard functions from observed cumulative hazard data

Syntax

**%**smooth([options,);

**purpose**

**%**smooth produces graphs of smoothed hazard functions using survival curve data output from either proc lifetest or proc phreg. With lifetest, it uses the data set produced by the outsurv option in the proc statement. With proc phreg, it uses the data set produced by the baseline statement. **%**smooth employs a kernel smoothing method described by H. Ramlau-Hansen (1983), "Smoothing Counting Process Intensities by Means of Kernel Functions," The Annals of Statistics 11, 453-466. If there is more than one survival curve in the input data set, **%**smooth will produce multiple smoothed hazard curves on the same axis.  
  
Note that %smooth can also be used to compute a hazard function that corresponds to a survival measure that has been age standardized, for instance one computed by a meansurv call to %predict.

**Non-positional parameters**

data the name of the data set containing survivor function estimates. The default is the most recently created data set.

time name of the variable containing event times.

survival the name of a variable containing survivor function estimates (the default is SURVIVAL, which is the automatic name in PROC LIFETEST).

width bandwidth of smoothing function. The default is 1/5 of the range of event times.

yscale scale to be used for plotting (default: linear). a log scale is requested by yscale = log

option plot/noplot

Example of usage:

%smooth(data=my.data,time=duration,width=8,survival=s)

Author: Paul D. Allison, University of Pennsylvania

allison@ssc.upenn.edu

## %hazard\_late smoothed hazard function from late entry data

**syntax**

**%**hazard\_late ([non-positional parameters]);

Purpose

SAS life table and Kaplan-Meier methods, as provided by proc lifetest, do not allow for the analysis of period (late entry or left-truncated) survival data. Since the Kaplan-Meier survival and hazard plots may be of interest in the initial phases of an analysis, this macro is provided to accommodate this lack. Proc phreg is called within the macro to perform the late-entry analysis, and the resulting survival estimates are processed by the %smooth macro to prepare a smoothed hazard plot. This macro was written for the purposes of the Albuquerque workshop, to mirror a facility available in Stata.

**Non-positional parameters**

Data input dataset

Strat stratification variable

Entry time of entry into surveillance

Exit time to event or censoring

Fail variable holding the censoring indicator

Censor value representing a censored observation

**Description of parameters**

Data name of input dataset (required)

Strat name of a stratification variable. If this parameter is not present, then an unstratified analysis is performed

Entry time of this subject’s entry into the period of observation. (required) The time scale of this value (days, years, years of age) must be the same as the scale used for the exit variable

Exit survival time for this subject (required). Must be on the same time scale as the entry time

Fail variable holding the censoring indicator

Censor values of the fail variable that are to be considered as censored events. More than one code is allowed

Exercise q112 uses this macro