Supplementary material I:
Manual for `%JointFrailty`-SAS-Macro

(Associated article: Computational issues in fitting joint frailty models for recurrent events and an associated terminal event, *Computer Methods and Programs in Biomedicine*)

Contents

1 Description

The `%JointFrailty`-SAS-Macro fits a parametric joint frailty model for the analysis of recurrent events with an associated terminal event using the NLMIXED procedure. The joint frailty model is given by

$$\lambda_1(t|X_1, Z) = Z\lambda_{10}(t)\exp(\beta_1' X_1)$$

$$\lambda_2(t|X_2, Z) = Z^{\gamma}\lambda_{20}(t)\exp(\beta_2' X_2)$$

where Z is a gamma or lognormal frailty with $E(Z) = 1$ and $Var(Z) = \theta$. Covariate vectors and the corresponding regression coefficient vectors are denoted by $X_1 = (X_{11}, \ldots, \beta_{1r})'$, $X_2 = (X_{21}, \ldots, X_{2s})'$ and $\beta_1 = (\beta_{11}, \ldots, \beta_{1r})'$, $\beta_2 = (\beta_{21}, \ldots, \beta_{2s})'$. Covariates can be endpoint-specific and do not have to coincide for recurrent and terminal events.

2 Arguments

```
%macro JointFrailty(input, output, sampleidvar, subjectidvar, timevar,
                    eventindicatorvar, linpredrec, linpredterm, frailtydist,
                    methodgamma, hazards, startval, optimstartval, quad,
                    quadpoints, path);
   ...
%mend JointFrailty;
```

`input`

Name of the dataset containing the data to be analysed. The dataset must have the following long-format-structure:

subjectid	time	eventindicator	covariate1	covariate2	...
1	0.174	1	1	1.57	...
1	1	0	1	1.57	...
2	1	0	0	0.43	...
3	0.376	1	1	2.31	...
3	0.863	1	1	2.31	...
3	0.942	2	1	2.31	...
⋮	⋮	⋮	⋮	⋮	...

The `input`-dataset must contain the following variables:

- `subjectid` is the subject-specific identification number (ID). The dataset has to be ordered by the subject-specific ID.
- `time` is the event-time. Within a single subject-ID, the dataset has to be ordered by the event times.
- `eventindicator` specifies the type of event. The following coding has to be applied: 0 for censoring, 1 for recurrent event, 2 for terminal event. In the example above, subject 1 has a recurrent event at time 0.174 and is censored at time 1. Subject 2 is censored at time 1 without having a recurrent event before. Subject 3 has two recurrent events at times 0.376 and 0.863 before having its terminal event at time 0.942. Importantly, the last line of each subject must contain either 0 or 2 as `eventindicator`, because the follow-up may only end due to censoring or due to the terminal event.
- `covariate1`, `covariate2` etc. specify subject-specific covariates which have to be either continuous or binary. For binary variables, a 0/1 coding has to be applied. Categorical covariates have to be split up into binary dummy-variables in advance.

The macro also allows to analyse multiple datasets simultaneously (e.g. in case of a simulation). Then the `input`-dataset has to contain the stacked sub-datasets and the sub-dataset-number has to be specified by the additional variable `sampleid`:

sampleid	subjectid	time	eventindicator	covariate1	covariate2	...
1	1	0.174	1	1	1.57	...
1	1	1	0	1	1.57	...
1	2	1	0	0	0.43	...
1	3	0.376	1	1	2.31	...
1	3	0.863	1	1	2.31	...
1	3	0.942	2	1	2.31	...
⋮	⋮	⋮	⋮	⋮	...	
2	1	0.573	1	0	1.62	...
2	1	0.872	2	0	1.62	...
⋮	⋮	⋮	⋮	⋮	...	

- `sampleid` is the sub-dataset-number in case of analysing multiple stacked datasets. In that case the `input`-dataset has to be ordered first by the sample-ID, second by the subject-ID and third by the event-time. The existence of a `sampleid`-variable is optional and may be ommitted if only a single dataset is to be analysed.

The variable names of the `input`-dataset can be specified by other arguments (e.g. `subjectidvar`, ...) and thus do not have to correspond to the above ones which are just exemplary.

`output`

Prefix for all output-dataset-names. The macro produces 3 datasets containing the results of the analysis (see Output-section 3). These datasets have the suffix `_est` (parameter estimates), `_conv` (covergence status), `_time` (processing time). As an example, in case of specifying `output = Test`, the output-datasets have the names `Test_est`, `Test_conv` and `Test_time`.

`sampleidvar`

Name of the variable in the `input`-dataset containing the sample-ID in case of analysing multiple stacked sub-datasets (see also documentation for `input`). If only one single dataset is analysed and no sample-ID-variable exists, please specify `sampleidvar = none`.

`subjectidvar`

Name of the variable in the `input`-dataset containing the subject-ID (see also documentation for `input`).

`timevar`

Name of the variable in the `input`-dataset containing the event times (see also documentation for `input`).

`eventindicatorvar`

Name of the variable in the `input`-dataset containing the eventindicator (see also documentation for `input`).

`linpredrec`

Linear predictor $\beta_1' X_1$ for the recurrent events. As an example, in case of the covariates with names `treatment` and `sex`, you have to specify
`linpredrec = beta11*treatment + beta12*sex`.
Please number the parameters in ascending order, i.e. `beta11`, `beta12` etc. The covariates have to be either continuous or binary. For binary variables, a 0/1 coding has to be applied. Categorical covariates have to be split up into binary dummy-variables in advance (see also documentation for `input`). The covariates specified for recurrent events may differ from those for the terminal event. Please consider, that `beta11`, `beta12` etc. are model parameters that require starting values which have to be specified (see also documentation for `startval`).

`linpredterm`

Linear predictor $\beta_2' X_2$ for the terminal event. As an example, in case of the covariates with names `treatment` and `sex`, you have to specify
`linpredterm = beta21*treatment + beta22*sex`.
Please number the parameters in ascending order, i.e. `beta21`, `beta22` etc. The covariates have to be either continuous or binary. For binary variables, a 0/1 coding has to be applied. Categorical covariates have to be split up into binary dummy-variables in advance (see also documentation for `input`). The covariates specified for recurrent events may differ from those for the terminal event. Please consider, that `beta21`, `beta22` etc. are model parameters that require starting values which have to be specified (see also documentation for `startval`).

`frailtydist`

Frailty-distribution; you have to specify either `frailtydist = lognormal` or `frailtydist = gamma`.

`methodgamma`

Method how to deal with a non-normal random effect in the NLMIXED procedure.

- If `frailtydist = lognormal`, the random effect is normally distributed and you have to specify `methodgamma = none`.
- If `frailtydist = gamma`, the random effect is not normally distributed and you can choose between the probability integral transformation (PIT) and the Likelihood reformulation (LR) method to enable estimation with the NLMIXED procedure . So you either have to specify `methodgamma = pit` or `methodgamma = lr`.

`hazards`

Parametric specification for the baseline-hazards $\lambda_{10}(t)$ and $\lambda_{20}(t)$. Possible options are:

- `hazards = piecewise`
  Baseline-hazards $\lambda_{10}(t)$ and $\lambda_{20}(t)$ are piecewise constant, each with 10 pieces. The sizes of the pieces are determined by the empirical 0.1-, 0.2-,..., 1-Quantiles $qr_1, \ldots, qr_{10}$ for recurrent events and $qd_1, \ldots, qd_{10}$ for terminal events. Thus, the baseline-hazards are given by

$$\lambda_{10}(t) = \begin{cases} \texttt{r01}, & \text{if } 0 \leq t \leq qr_1 \\ \texttt{r02}, & \text{if } qr_1 < t \leq qr_2 \\ \texttt{r03}, & \text{if } qr_2 < t \leq qr_3 \\ \vdots & \\ \texttt{r10}, & \text{if } qr_9 < t \leq qr_{10} \end{cases}$$

$$\lambda_{20}(t) = \begin{cases} \texttt{h01}, & \text{if } 0 \leq t \leq qd_1 \\ \texttt{h02}, & \text{if } qd_1 < t \leq qd_2 \\ \texttt{h03}, & \text{if } qd_2 < t \leq qd_3 \\ \vdots & \\ \texttt{h10}, & \text{if } qd_9 < t \leq qd_{10} \end{cases}$$

  Please consider, that `r01`,...,`r10` (all ≥ 0) and `h01`,...,`h10` (all ≥ 0) are model parameters that require starting values which have to be specified (see also documentation for `startval`).

- `hazards = constant`
  Baseline-hazards are constant with $\lambda_{10}(t) = \texttt{r01}$ and $\lambda_{20}(t) = \texttt{h01}$. Please consider, that `r01` and `h01` (both ≥ 0) are model parameters that require starting values which have to be specified (see also documentation for `startval`).

- `hazards = weibull`
  Baseline-hazards originate from Weibull-distributions, i.e. $\lambda_{10}(t) = \lambda_1 \nu_1 t^{\nu_1 - 1}$ and $\lambda_{20}(t) = \lambda_2 \nu_2 t^{\nu_2 - 1}$. Here $\lambda_1 \geq 0$, $\lambda_2 \geq 0$ and $\nu_1 \geq 0$, $\nu_2 \geq 0$ are the scale- and the shape parameters. They are designated as `scalerec`, `scaleterm` and `shaperec`, `shapeterm` in the macro. Please consider, that `scalerec`, `scaleterm`, `shaperec` and `shapeterm` are model parameters that require starting values which have to be specified (see also documentation for `startval`).

`startval`

Name of the dataset that contains the starting values for the likelihood-maximization-algorithm. The dataset has to contain the two variables `parameter` and `estimate`. The variable `parameter` is a character variable that contains the parameter-names and `estimate` is a numeric variable that contains the starting values for the respective parameters.
Regarding the choice of starting values, please note parameter bounds. Let's consider as an example a joint frailty model with constant hazards: Then the parameters `theta` (frailty variance), `r01` and `h01` (baseline hazards) are ≥ 0. Please do not specify the parameter bounds (i.e.

0) as starting values for these parameters because this may cause numerical problems. A possible `startval`-dataset for a joint frailty model with constant hazards and only one parameter in each linear predictor is:

parameter	estimate
beta11	-0.46
beta21	-0.32
gamma	1.50
theta	2.10
r01	0.43
h01	0.12

A possible `startval`-dataset for a joint frailty model with Weibull-hazards and only one parameter in each linear predictor is:

parameter	estimate
beta11	-0.46
beta21	-0.32
gamma	1.50
theta	2.10
scalerec	0.43
shaperec	1
scaleterm	0.52
shapeterm	2

In the NLMIXED procedure, it is also possible to run the optimization algorithm without specification of starting values. Then SAS automatically is using 1 as starting value for each parameter. However, within this macro you always have to specify a `startval`-dataset. If you have no ideas about good starting values, just use 1 as starting value for each parameter.

`optimstartval`

This argument determines if the starting values that were determined in `startval` should be optimized. Possible options are `optimstartval = true` and `optimstartval = false`. If `optimstartval = true`, first a simplified model without random effect is fitted in PROC NLMIXED, i.e.

$$\lambda_1(t|X_1, Z) = \lambda_{10}(t)\exp(\beta_1' X_1)$$

$$\lambda_2(t|X_2, Z) = \lambda_{20}(t)\exp(\beta_2' X_2)\ .$$

This simplified model does not contain the random effect Z (along with its exponent γ), but is otherwise equal (regarding covariates, hazards etc.) to the joint frailty model that is to be fit. The fit of the simplified model delivers estimates and therefore new starting values for all joint-frailty-parameters, except for the parameters θ (`theta`) and γ (`gamma`).
As an example, let's consider fitting a joint frailty model with constant hazards. We specify `optimstartval = true` along with the following `startval`-dataset:

parameter	estimate
beta11	1
beta21	1
gamma	1
theta	1
r01	1
h01	1

Then first the above mentioned simplified model is fit using 1 as starting value for each parameter. Afterwards, the estimates for `beta11`, `beta21`, `r01` and `h01` will (automatically) be used as starting values for fitting the joint frailty model. The specified starting values for `gamma` and `theta` will not be updated this way. So the (updated/optimized) starting values that are used for the joint frailty model fit could for example be:

parameter	estimate
beta11	-0.524
beta21	-0.876
gamma	1
theta	1
r01	0.345
h01	0.121

`quad`

This argument specifies which numerical quadrature-procedure is used for approximating the integrals (with respect to the random effect distribution) in the marginal likelihood. The two possible options are `quad = ad` (Adaptive Gaussian Quadrature) and `quad = noad` (Non-adaptive Gaussian Quadrature). For a detailed reference, we refer to the SAS manual of PROC NLMIXED (keyword *Integral Approximations*).

`quadpoints`

This argument specifies the number of quadrature-points used for the numerical approximation of the integrals in the marginal likelihood. Please either specify an integer value (e.g. `quadpoints = 45`) or specify `quadpoints = auto`. In the latter case, PROC NLMIXED will automatically choose the number of quadrature points. For a detailed reference, we refer to the SAS manual of PROC NLMIXED (keyword *Integral Approximations*).

`path`

This argument specifies the path where to store the output-datasets in csv-format (example: `path = C:\documents\results`). If no output-datasets should be stored in csv-format, please specify `path = none`.

3 Output

Besides the NLMIXED-summary-output in html-format, the macro produces 3 datasets in your SAS-library containing the results of the analysis:

- A dataset for parameter estimates (suffix `_est`)
- A dataset for convergence status (suffix `_conv`)
- A dataset for needed processing time (suffix `_time`)

These datasets may also be stored in csv-format, if the argument `path` is correctly specified. The argument `output` determines the prefix of the dataset-names (see documentation for `path`). Let's illustrate that by an example: We use the macro for a fitting a joint frailty model with constant hazards, one covariate and specify `output = Test`. The output-datasets are given by:

`Test_est`

In case of an `input`-dataset without `sampleid`:

Parameter	Estimate	StandardError	DF	tValue	Probt	...
r01	0.5649	0.03067	1999	18.42	<.0001	...
h01	0.1779	0.02072	1999	8.59	<.0001	...
beta11	-0.1705	0.07819	1999	-2.18	0.0293	...
beta21	-0.3603	0.1309	1999	-2.75	0.0059	...
gamma	1.5135	0.2202	1999	6.87	<.0001	...
theta	0.6756	0.08229	1999	8.21	<.0001	...

In case of an `input`-dataset with 2 stacked sub-datasets identified by `sampleid`:

sampleid	Parameter	Estimate	StandardError	DF	tValue	Probt	...
1	r01	0.5649	0.03067	1999	18.42	<.0001	...
1	h01	0.1779	0.02072	1999	8.59	<.0001	...
1	beta11	-0.1705	0.07819	1999	-2.18	0.0293	...
1	beta21	-0.3603	0.1309	1999	-2.75	0.0059	...
1	gamma	1.5135	0.2202	1999	6.87	<.0001	...
1	theta	0.6756	0.08229	1999	8.21	<.0001	...
2	r01	0.6298	0.03614	1999	17.43	<.0001	...
2	h01	0.2061	0.02291	1999	9.00	<.0001	...
2	beta11	-0.2529	0.08018	1999	-3.15	0.0016	...
2	beta21	-0.4924	0.1309	1999	-3.76	0.0002	...
2	gamma	1.4143	0.1745	1999	8.10	<.0001	...
2	theta	0.8753	0.08930	1999	9.80	<.0001	...

For a detailed overview on the variables in the output-dataset `Test_est` we refer to the SAS manual of PROC NLMIXED (keyword *ODS table ParameterEstimates*).

`Test_conv`

In case of an `input`-dataset without `sampleid`:

Reason	Status
NOTE: GCONV convergence criterion satisfied.	0

In case of an `input`-dataset with 2 stacked sub-datasets identified by `sampleid`:

sampleid	Reason	Status
1	NOTE: GCONV convergence criterion satisfied.	0
2	NOTE: GCONV convergence criterion satisfied.	0

If `Status` $= 0$ the algorithm converged. In case of `Status` $\neq 0$ the algorithm did not converge. For a detailed overview on the variables in the output-dataset `Test_conv` we refer to the SAS manual of PROC NLMIXED (keyword *ODS table ConvergenceStatus*).

`Test_time`

duration_in_min
0.123453

This is the processing time (in min) needed for the macro-call, regardless whether the `input`-dataset is divided into several sub-datasets or not.

4 Examples

Let's consider the following dataset with name `example_input`:

subjectid	time	eventindicator	x1	x2	...
1	0.174	1	1	1.57	...
1	1	0	1	1.57	...
2	1	0	0	0.43	...
3	0.376	1	1	2.31	...
3	0.863	1	1	2.31	...
3	0.942	2	1	2.31	...
⋮	⋮	⋮	⋮	⋮	...

Our output-dataset-names should have the prefix `example_output` and be stored in the folder `C:\documents\results`.

Example 1

Fit a joint frailty model with constant hazards and gamma-distributed frailty (using the likelihood reformulation method). The only covariate to be included into the model is `x1`. Choose 1 as starting value for each parameter and do not optimize starting values. Adaptive gaussian quadrature with 30 quadrature-points should be chosen for numerical integral-approximations.

```
data initpar;
  length parameter $ 20;
  input parameter $ estimate;
  datalines;
  r01 1
  h01 1
  beta11 1
  beta21 1
  gamma 1
  theta 1
  ;
run;

%JointFrailty(input = example_input,
              output = example_output,
              sampleidvar = none,
              subjectidvar = subjectid,
              timevar = time,
              eventindicatorvar = eventindicator,
              linpredrec = beta11*x1,
              linpredterm = beta21*x1,
              frailtydist = gamma,
              methodgamma = lr,
              hazards = constant,
              startval = initpar,
              optimstartval = false,
              quad = ad,
              quadpoints = 30,
              path = C:\documents\results);
```

Example 2

Fit a joint frailty model with Weibull hazards and lognormal-distributed frailty. The covariates to be included into the model are x1 and x2. Choose 1 as starting value for each parameter and optimize starting values. Non-adaptive gaussian quadrature with 15 quadrature-points should be chosen for numerical integral-approximations.

```
data initpar;
  length parameter $ 20;
  input parameter $ estimate;
  datalines;
  scalerec 1
  shaperec 1
  scaleterm 1
  shapeterm 1
  beta11 1
  beta12 1
  beta21 1
  beta22 1
  gamma 1
  theta 1
  ;
run;

%JointFrailty(input = example_input,
              output = example_output,
              sampleidvar = none,
              subjectidvar = subjectid,
              timevar = time,
              eventindicatorvar = eventindicator,
              linpredrec = beta11*x1+beta12*x2,
              linpredterm = beta21*x1+beta22*x2,
              frailtydist = lognormal,
              methodgamma = none,
              hazards = weibull,
              startval = initpar,
              optimstartval = true,
              quad = noad,
              quadpoints = 15,
              path = C:\documents\results);
```

Example 3

Fit a joint frailty model with piecewise constant hazards and lognormal-distributed frailty. The only covariate to be included into the model is `x1`. Choose 1 as starting value for each parameter and optimize starting values. Non-adaptive gaussian quadrature with 15 quadrature-points should be chosen for numerical integral-approximations.

```
data initpar;
  length parameter $ 20;
  input parameter $ estimate @@;
  datalines;
  r01 1 r02 1 r03 1 r04 1 r05 1 r06 1 r07 1 r08 1 r09 1 r10 1
  h01 1 h02 1 h03 1 h04 1 h05 1 h06 1 h07 1 h08 1 h09 1 h10 1
  beta11 1
  beta21 1
  gamma 1
  theta 1
  ;
run;

%JointFrailty(input = example_input,
              output = example_output,
              sampleidvar = none,
              subjectidvar = subjectid,
              timevar = time,
              eventindicatorvar = eventindicator,
              linpredrec = beta11*x1,
              linpredterm = beta21*x1,
              frailtydist = lognormal,
              methodgamma = none,
              hazards = piecewise,
              startval = initpar,
              optimstartval = true,
              quad = noad,
              quadpoints = 15,
              path = C:\documents\results);
```