

Extended multivariate generalised linear and non-linear mixed effects models

Stata UK Meeting
Cass Business School
7th September 2017

Michael J. Crowther

*Biostatistics Research Group,
Department of Health Sciences,
University of Leicester, UK,
michael.crowther@le.ac.uk
@Crowther.MJ*

Funding: MRC (MR/P015433/1)

Outline

- Motivation for this work
- Extended multivariate generalised linear and non-linear mixed effects models
- `megenreg`
- Methods development using `megenreg`
- Future directions

Motivation

- More data \rightarrow more questions
need for appropriate statistical modelling techniques,
and implementations

Motivation

- More data → more questions
need for appropriate statistical modelling techniques,
and implementations
- Growth in access to EHR
biomarkers < patients < GP practice area <
geographical regions...

Motivation

- More data → more questions
need for appropriate statistical modelling techniques,
and implementations
- Growth in access to EHR
biomarkers < patients < GP practice area <
geographical regions...
- More challenges
time-dependent effects, non-linear covariate effects

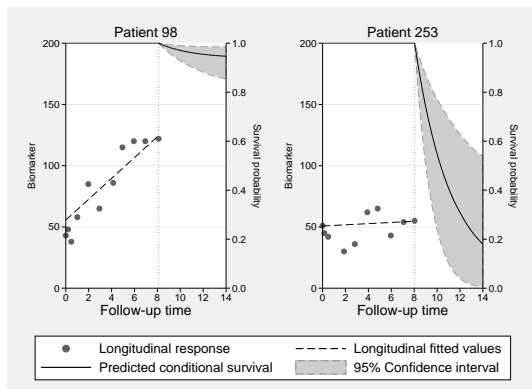
Motivation

- More data → more questions
need for appropriate statistical modelling techniques,
and implementations
- Growth in access to EHR
biomarkers < patients < GP practice area <
geographical regions...
- More challenges
time-dependent effects, non-linear covariate effects

**We need modelling frameworks that can accommodate
a lot of different things**

Motivation

Joint longitudinal-survival models



Linking via - current value, gradient, AUC, random effects...

Motivation

Joint longitudinal-survival models - extensions

- Competing risks [1]
- Different types of outcomes [2]
- Multiple continuous outcomes [3]
- Delayed entry [4]
- Recurrent events and a terminal event [5]
- Prediction [6]
- Many others...

Motivation

Joint longitudinal-survival models - software

- `stjm` in Stata [7]
- `gsem` in Stata, see Yulia's talk from last year
- `frailtypack` in R [8]
- `joineR` in R [9]
- `JM` and `JMBayes` in R [10, 11]
- Many others...

Motivation

(My) Methods development - software

- stjmc - joint longitudinal-survival models
- stmixed - multilevel survival models
- stgenreg - general parametric survival models
- ...

Motivation

(My) Methods development - software

- stjmc - joint longitudinal-survival models
- stmixed - multilevel survival models
- stgenreg - general parametric survival models
- ...

Each new project brings a new code base to maintain...could I make my life easier?

The goal

A general framework for the analysis of data of all types

- Multiple outcomes of varying types
- Measurement schedule can vary across outcomes
- Any number of levels and random effects
- Sharing and linking random effects between outcomes
- Sharing functions of the expected value of other outcomes
- A reliable estimation engine
- Easily extendable by the user
- ...

The goal

A general framework for the analysis of data of all types

- Multiple outcomes of varying types
- Measurement schedule can vary across outcomes
- Any number of levels and random effects
- Sharing and linking random effects between outcomes
- Sharing functions of the expected value of other outcomes
- A reliable estimation engine
- Easily extendable by the user
- ...

I think I made my life more difficult!

The goal

**Extended multivariate generalised linear and non-linear
mixed effects models**

`megenreg`

The goal

Extended multivariate generalised linear and non-linear mixed effects models

`megenreg`

- Much of what `megenreg` can do, can be done (better) with `gsem`

The goal

Extended multivariate generalised linear and non-linear mixed effects models

`megenreg`

- Much of what `megenreg` can do, can be done (better) with `gsem`
- Much of what `megenreg` can do, cannot be done with `gsem`

A general level likelihood

Straight from the Stata manual...for a one-level model with n response variables:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{b}, \boldsymbol{\beta}) = \prod_{i=1}^n p_i(y_i|\mathbf{x}, \mathbf{b}, \boldsymbol{\beta})$$

For a two-level model:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{b}, \boldsymbol{\beta}) = \prod_{i=1}^n \prod_{j=1}^t p_i(y_{ij}|\mathbf{x}, \mathbf{b}, \boldsymbol{\beta})$$

A general level likelihood

The log likelihood is obtained by integrating out the unobserved random effects

$$ll(\boldsymbol{\beta}) = \log \int_{\mathcal{R}^r} p(\mathbf{y}|\mathbf{x}, \mathbf{b}, \boldsymbol{\beta}) \phi(\mathbf{b}|\Sigma_{\mathbf{b}}) d\mathbf{b}$$

we assume $\phi(\cdot)$ is the multivariate normal density for \mathbf{b} , with mean vector $\mathbf{0}$ and variance-covariance matrix $\Sigma_{\mathbf{b}}$. We have $\Sigma_{\mathbf{b}}$ becoming block diagonal with further levels, with a block for each level

A general level likelihood

Alternatively, exploiting conditional independence amongst level $l - 1$ units, given the random effects at higher levels,

$$ll(\boldsymbol{\beta}) = \log \int \phi(\mathbf{b}^{(L)} | \boldsymbol{\Sigma}^{(L)}) \prod p^{(L-1)}(\mathbf{y} | \mathbf{x}, \mathbf{b}^L, \boldsymbol{\beta}) d\mathbf{b}^{(L)}$$

where, for $l = 2, \dots, L$

$$p^{(l)}(\mathbf{y} | \mathbf{x}, \mathbf{B}^{l+1}, \boldsymbol{\beta}) = \int \phi(\mathbf{b}^{(l)} | \boldsymbol{\Sigma}^{(l)}) \prod p^{(l-1)}(\mathbf{y} | \mathbf{x}, \mathbf{B}^l, \boldsymbol{\beta}) d\mathbf{b}^{(l)}$$

Estimation challenges

- At each level, we need to integrate out our normally distributed random effects
- Generally this is done using Gauss-Hermite numerical quadrature

```
intmethod(mvaghermite | ghermite)
```

- Issue with GH quadrature is it doesn't scale up well:
 - 7-point quadrature; for 1 random effect we evaluate our function at 7-points
 - 7-point quadrature; for 6 random effects, we evaluate it at $7^6 = 117,649$ points

Estimation challenges - alternatives

- An alternative is Monte Carlo integration
- Also known for its use in maximum simulated likelihood - see the special issue in the Stata Journal Vol 6 No 2
- This is a rather brute force approach, but it's usefulness is in it's simplicity

$$L(\boldsymbol{\theta}) = \int f(\mathbf{y}|\boldsymbol{\theta}, \mathbf{b})\phi(\mathbf{b})\partial\mathbf{b} = \frac{1}{m} \sum_{u=1}^m f(y|\theta, \mathbf{b}_u)$$

The important thing to note is m doesn't have to change when extra random effects are added.

Estimation challenges - alternatives

Monte Carlo integration can be improved by:

- antithetic sampling [12]
- Halton sequences [13]
- an adaptive procedure just like adaptive GH quadrature, resulting in an importance sampling approximation

Extensions - level-specific random effect distributions

$$ll(\boldsymbol{\theta}) = \log \int \phi_L(\mathbf{b}^{(L)} | \boldsymbol{\Sigma}^{(L)}) \prod p^{(L-1)}(\mathbf{y} | \mathbf{x}, \mathbf{b}^L, \boldsymbol{\beta}) d\mathbf{b}^{(L)}$$

where, for $l = 2, \dots, L$

$$p^{(l)}(\mathbf{y} | \mathbf{x}, \mathbf{B}^{l+1}, \boldsymbol{\beta}) = \int \phi_l(\mathbf{b}^{(l)} | \boldsymbol{\Sigma}^{(l)}) \prod p^{(l-1)}(\mathbf{y} | \mathbf{x}, \mathbf{B}^l, \boldsymbol{\beta}) d\mathbf{b}^{(l)}$$

Extensions - level-specific random effect distributions and integration techniques

- This formulation now allows us to specify different distributions at each level
- Assess robustness using the t -distribution
- Issue of which integration techniques to apply at each level
 - e.g. one random effect at level 1, many at level 2, then use AGHQ at level 3, and MCI at level 2

```
intmethod(mvaghermite mcarlo)  
redistribution(normal t) df(3)
```


Standard linear predictor

The standard linear predictor for a general level model can be written as follows,

$$\eta = \mathbf{X}\boldsymbol{\beta} + \sum_{l=2}^L \mathbf{X}^l \mathbf{b}^l$$

where subscripts are omitted. We have \mathbf{X} our vector of covariates, which could vary at any level, with associated fixed effect coefficient vector $\boldsymbol{\beta}$, and \mathbf{X}^l the vector of covariates with random effects \mathbf{b}^l at level l .

Extended linear predictor

$$\eta_i = g_i(E[y_i | \mathbf{X}, \mathbf{b}]) = \sum_{r=1}^{R_i} \prod_{s=1}^{S_{ir}} \psi_{irs}$$

where $g_i(\cdot)$ is the link function for the i th outcome. To maintain generality, $\psi_{irs}(t)$ can take many forms, including,

$$\psi_{irs}(t) = X$$

$$\psi_{irs}(t) = \beta$$

$$\psi_{irs}(t) = b$$

$$\psi_{irs}(t) = q(t)$$

$$\psi_{irs}(t) = d_{rs}(E[y_j]), \quad \text{where } j = 1, \dots, k, j \neq i$$

Title

megenreg — Extended multivariate generalised linear and non-linear mixed effects models

Syntax

```
megenreg models [if] [in] [, options]
```

where *models* are the model specifications; see **megenreg models**.

<i>options</i>	Description
<i>model_description_options</i>	fully define, along with <i>models</i> , the model to be fit
<i>estimation_options</i>	method used to obtain estimation results
<i>reporting_options</i>	reporting of estimation results

fweights, **iwweights**, and **pweights** are allowed; see **weight**.

Also see **megenreg postestimation** for features available after estimation.

megenreg in Stata

- Everything I've talked about will be available in the megenreg package in Stata
- It is a simplified/modified version of Stata's official gsem
- megenreg will have many extensions, such as
 - Alternative models, such as spline based survival models
 - Extending sharing between outcomes, motivated by joint modelling
 - User-defined likelihood functions
 - Other things...

Distributional choices

- Gaussian, Poisson, binomial, beta, negative binomial
- exponential, Weibull, Gompertz, log-normal, log-logistic, gamma, Royston-Parmar
- Non-linear outcome models
- User-defined hazard functions
- More to add...

1. A general level parametric survival model

The Royston-Parmar survival model uses restricted cubic splines of log time, on the log cumulative hazard scale, i.e.,

$$\log H(y_i) = s(\log(y_i)|\beta_{\mathbf{k}}) + \eta_i$$

```
. list patient time infect age female in 1/4, noobs
```

patient	time	infect	age	female
1	8	1	28	0
1	16	1	28	0
2	13	0	48	1
2	23	1	48	1

```
. megenreg (time age female M1[patient], ///  
> family(rp, failure(infect) scale(h) df(3)))
```

1. A general level parametric survival model

Relax the normally dist. random effects assumption;

```
. megenreg (time age female M1[patient], family(rp, failure(infect) scale(h) df(3)))  
>           , redistribution(t) df(3)
```

1. A general level parametric survival model

Relax the normally dist. random effects assumption;

```
. megenreg (time age female M1[patient], family(rp, failure(infect) scale(h) df(3)))  
>           , redistribution(t) df(3)
```

Higher levels of clustering;

```
. megenreg (time trt M1[trial] M2[trial>patient], ...)
```


1. A general level parametric survival model

Relax the normally dist. random effects assumption;

```
. megenreg (time age female M1[patient], family(rp, failure(infect) scale(h) df(3)))
>           , redistribution(t) df(3)
```

Higher levels of clustering;

```
. megenreg (time trt M1[trial] M2[trial>patient], ...)
```

Random coefficients;

```
. megenreg (time trt M1[trial] trt#M1[trial] M2[trial>patient], ... )
```

1. A general level parametric survival model

Relax the normally dist. random effects assumption;

```
. megenreg (time age female M1[patient], family(rp, failure(infect) scale(h) df(3)))
>           , redistribution(t) df(3)
```

Higher levels of clustering;

```
. megenreg (time trt M1[trial] M2[trial>patient], ...)
```

Random coefficients;

```
. megenreg (time trt M1[trial] trt#M1[trial] M2[trial>patient], ... )
```

Time-dependent effects;

```
. megenreg (stime trt trt#{log(&t)} M1[id1] M2[id1>id2], ... timevar(stime))
```

1. A general level parametric survival model

Relax the normally dist. random effects assumption;

```
. megenreg (time age female M1[patient], family(rp, failure(infect) scale(h) df(3)))
>           , redistribution(t) df(3)
```

Higher levels of clustering;

```
. megenreg (time trt M1[trial] M2[trial>patient], ...)
```

Random coefficients;

```
. megenreg (time trt M1[trial] trt#M1[trial] M2[trial>patient], ... )
```

Time-dependent effects;

```
. megenreg (stime trt trt#{log(&t)} M1[id1] M2[id1>id2], ... timevar(stime))
```

Non-linear covariate effects

```
. gen age2 = age^2
. megenreg (stime trt trt#{log(&t)} age age2 M1[id1] M2[id1>id2], ... )
```

2. A general level relative survival model

Relative survival models are used widely, particularly in population based cancer epidemiology [14]. They model the excess mortality in a population with a particular disease, compared to a reference population.

$$h(y) = h^*(y) + \lambda(y)$$

where $h^*(y)$ is the expected mortality in the reference population. Any of the previous models can be turned into a relative survival model;

```
. megenreg (stime trt trt#log(&t) M1[id1] M2[id1>id2], ///  
>          family(rp, failure(died) df(3) scale(h) bhazard(bhaz))
```

3. General level joint frailty survival models

- An area of intense research in recent years is in the field of joint frailty survival models, for the analysis of joint recurrent event and terminal event data
- Here I focus on the two most popular approaches, proposed by Liu et al. (2004) [15] and Mazroui et al. (2012) [16]
- In both, we have a survival model for the recurrent event process, and a survival model for the terminal event process, linked through shared random effects

3. General level joint frailty survival models

$$h_{ij}(y) = h_0(y) \exp(\mathbf{X}_{1ij}\boldsymbol{\beta}_1 + b_i)$$

$$\lambda_i(y) = \lambda_0(y) \exp(\mathbf{X}_{1i}\boldsymbol{\beta}_2 + \alpha b_i)$$

where $h_{ij}(y)$ is the hazard function for the j th event of the i th patient, $\lambda_i(y)$ is the hazard function for the terminal event, and $b_i \sim N(0, \sigma^2)$. We can fit such a model with `megenreg`, adjusting for treatment in each outcome model,

```
. megenreg (rectime trt M1[id1] , family(rp, failure(recevent)) scale(h) df(5))
>          (stime trt M1[id1]@alpha , family(rp, failure(died)) scale(h) df(3))
```

3. General level joint frailty survival models

$$h_{ij}(y) = h_0(y) \exp(\mathbf{X}_{1ij}\boldsymbol{\beta}_1 + b_{1i} + b_{2i})$$

$$\lambda_i(y) = \lambda_0(y) \exp(\mathbf{X}_{1i}\boldsymbol{\beta}_2 + b_{2i})$$

where $b_{1i} \sim N(0, \sigma_1^2)$ and $b_{2i} \sim N(0, \sigma_2^2)$. We give an example of how to fit this model with `megenreg`, this time illustrating how to use different distributions for the recurrent event and terminal event processes,

```
. megenreg (rectime trt M1[id1] M2[id1] , family(weibull, failure(recevent))) ///
>          (stime trt M2[id1] , family(rp, failure(died) scale(h) df(3)))
```

4. Generalised multivariate joint models

Multiple longitudinal biomarkers

$$Y_1 \sim Weib(\lambda, \gamma), \quad Y_2 \sim N(\mu_2, \sigma_2^2), \quad Y_3 \sim N(\mu_3, \sigma_3^2)$$

The linear predictor of the survival outcome can be written as follows,

$$\eta_1(t) = \mathbf{X}\boldsymbol{\beta}_0 + E[y_2(t)|\eta_2(t)]\beta_1 + E[y_3(t)|\eta_3(t)]\beta_2 + \\ E[y_2(t)|\eta_2(t)] \times E[y_3(t)|\eta_3(t)]\beta_3$$

```
. megenreg (stime trt EV[logb]@beta1 EV[logp]@beta2 EV[logb]#EV[logp]@beta3 ,
>
>                                     family(weibull, failure(died)))
>     (logb {&t}@l1 {&t}#M2[id] M1[id] , family(gaussian) timevar(time))
>     (logp {&t}@l2 {&t}#M4[id] M3[id] , family(gaussian) timevar(time))
>     , covariance(unstructured)
```


4. Generalised multivariate joint models

Competing risks

```
. list id logb logp time trt stime diedpbc diedother if id==3, noobs
```

id	logb	logp	time	trt	stime	diedpbc	diedother
3	.3364722	2.484907	0	D-penicil	2.77078	1	0
3	.0953102	2.484907	.481875	D-penicil	.	.	.
3	.4054651	2.484907	.996605	D-penicil	.	.	.
3	.5877866	2.587764	2.03428	D-penicil	.	.	.

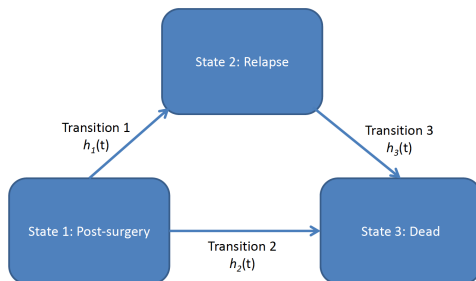
```
. megenreg (stime trt EV[logb]@a1 EV[logp]@a2 , family(weibull, failure(diedpbc)))
> (stime trt EV[logb]@a3 EV[logp]@a4 , family(gompertz, failure(diedother)))
> (logb {&t}@l1 {&t}#M2[id] M1[id] , family(gaussian) timevar(time))
> (logp {&t}@l2 {&t}#M4[id] M3[id] , family(gaussian) timevar(time))
```

4. Generalised multivariate joint models

Joint frailty - The extensive frailtypack in R has recently been extended to fit a joint model of a continuous biomarker, a recurrent event process, and a terminal event [5, 8]. We can use `megenreg`,

```
. megenreg (canctime trt EV[logb]@a1 EV[logp]@a2 M5[id]           , ... )
>          (stime    trt EV[logb]@a4 EV[logp]@a5 M5[id]@alpha , ... )
>          (logb    {&t}@l1 {&t}#M2[id] M1[id]                 , ... )
>          (logp    {&t}@l2 {&t}#M4[id] M3[id]                 , ... )
```

4. Generalised multivariate joint models



```

. megenreg (canctime trt EV[logb]@a1 EV[logp]@a2 , family(weibull, failure(canc))) ///
> (stimenocanc trt EV[logb]@a4 EV[logp]@a5 , ///
> family(gompertz, failure(diednocanc) ltrunc(canctime)) ///
> (stimecanc trt EV[logb]@a4 EV[logp]@a5 , family(gompertz, failure(diedcanc))) ///
> (logb {&t} @l1 {&t} #M2[id] M1[id] , family(gaussian) timevar(time)) ///
> (logp {&t} @l2 {&t} #M4[id] M3[id] , family(gaussian) timevar(time))
  
```

5. A user-defined model - utility functions

A Gaussian response model

$$y \sim N(\eta, \sigma^2)$$

```
real matrix gauss_logl(transmorphic gml)
{
  y          = gml_util_depvar(gml)          //dep. var.
  linpred    = gml_util_xzb(gml)            //lin. pred.
  sdre       = exp(gml_util_xb(gml,1))      //anc. param.
  return(lnnormalden(y,linpred,sdre))      //logl
}

. megenreg (logb time time#M2[id] M1[id], family(user, loglf(gauss_logl)) np(1))
```

6. A NLME example with multiple linear predictors

Consider Murawska et al. (2012), they developed a Bayesian NL joint model, with Gaussian response variable, and multiple non-linear predictors each with fixed effects and a random intercept. The overall non-linear predictor is defined as,

$$f(t) = \beta_{1i} + \beta_{2i} \exp^{-\beta_{3i}t}$$

where each linear predictor was constrained to be positive,

$$\beta_{1i} = \exp(X_1\beta_1 + b_{1i})$$

$$\beta_{2i} = \exp(X_2\beta_2 + b_{2i})$$

$$\beta_{3i} = \exp(X_3\beta_3 + b_{3i})$$

and for the survival outcome

$$\lambda(t) = \lambda_0(t) \exp(\alpha_1 b_{1i} + \alpha_2 b_{2i} + \alpha_3 b_{3i})$$

6. A NLME example with multiple linear predictors

We can fit this, and extend it, easily with `megenreg`

```
. megenreg (resp age female M1[id], family(user, loglf(nlme_logl))    ///  
>                                     np(1) timevar(time))  
>                                     (age female M2[id], family(null))  
>                                     (age female M3[id], family(null))  
>                                     (stime age female EV[resp]@alpha1 EV[2]@alpha2 EV[3]@alpha3, ///  
>                                     family(weibull, failure(died))),  
>                                     covariance(unstructured))  
  
real matrix nlme_logl(transmorphic gml, real matrix t)  
{  
  y           = gml_util_depvar(gml)           //dep.var.  
  linpred1   = exp(gml_util_xzb(gml))         //main lin. pred.  
  linpred2   = exp(gml_util_xzb_mod(gml,2))   //extra lin. preds  
  linpred3   = exp(gml_util_xzb_mod(gml,3))  
  sdre       = exp(gml_util_xb(gml,1))        //anc. param  
  linpred    = linpred1 :+ linpred2:*exp(-linpred3:*t) //nonlin. pred  
  return(lnnormalden(y,linpred,sdre))         //logl  
}
```

7. Mixed effects for the level 1 variance function

A recent paper by Goldstein et al. (2017) [17] proposed a two-level model with complex level 1 variation, of the form,

$$\begin{aligned}y_{ij} &= \mathbf{X}_{1ij}\boldsymbol{\beta}_1 + \mathbf{Z}_{1ij}\mathbf{b}_{1j} + \epsilon_{ij} \\ \epsilon_{ij} &\sim N(0, \sigma_e^2) \\ \log(\sigma_e^2) &= \mathbf{X}_{2ij}\boldsymbol{\beta}_2 + \mathbf{Z}_{2ij}\mathbf{b}_{2j} \\ \begin{pmatrix} \mathbf{b}_{1j} \\ \mathbf{b}_{2j} \end{pmatrix} &\sim N \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{b_1} & \\ \boldsymbol{\Sigma}_{b_1 b_2} & \boldsymbol{\Sigma}_{b_2} \end{pmatrix} \right]\end{aligned}$$

7. Mixed effects for the level 1 variance function

We can fit this, and extend it, easily with `megenreg`

```
real matrix lev1_logl(transmorphic gml, real matrix t)
{
  y          = gml_util_depvar(gml)           //response
  linpred1  = gml_util_xzb(gml)             //lin. pred.
  varresid  = exp(gml_util_xzb_mod(gml,2))   //lev1 lin. pred
  return(lnnormalden(y,linpred,sqrt(varresid))) //logl
}

. megenreg (resp female age age#M2[id] M1[id], family(user, loglf(lev1_logl)))
           (age female M3[id], family(null))
           covariance(unstructured)
```


Summary

- I've presented a very general, and hopefully usable, implementation which can fit a lot of different and new models

Summary

- I've presented a very general, and hopefully usable, implementation which can fit a lot of different and new models
- Through the complex linear predictor, we allow seamless development of novel models, and crucially, a way of making them immediately available to researchers through an accessible implementation
 - Realised it can fit multivariate network IPD meta-analysis models this week

Summary

- I've presented a very general, and hopefully usable, implementation which can fit a lot of different and new models
- Through the complex linear predictor, we allow seamless development of novel models, and crucially, a way of making them immediately available to researchers through an accessible implementation
 - Realised it can fit multivariate network IPD meta-analysis models this week
- I've incorporated level-specific random effect distributions, and integration techniques

Stuff I didn't show

- `family(user, hazard(funcname)
cumhazard(funcname))`
- `fp()` and `rsc()` as elements
- `dEV[]`, `d2EV[]`, `iEV[]` as elements
- Shell files - just like `gsem`

Future directions

- Dynamic risk prediction, predictions will be a key focus of the `megenreg` engine

Future directions

- Dynamic risk prediction, predictions will be a key focus of the `megenreg` engine
- location-scale models, multi-membership models...

Future directions

- Dynamic risk prediction, predictions will be a key focus of the `megenreg` engine
- location-scale models, multi-membership models...
- It's so general, and hence it can be slow(er)

Future directions

- Dynamic risk prediction, predictions will be a key focus of the `megenreg` engine
- location-scale models, multi-membership models...
- It's so general, and hence it can be slow(er)
- score and Hessian - analytic derivatives will provide substantial speed gains, so far I've implemented hybrid analytic and numeric derivatives.

Future directions

- Dynamic risk prediction, predictions will be a key focus of the `megenreg` engine
- location-scale models, multi-membership models...
- It's so general, and hence it can be slow(er)
- score and Hessian - analytic derivatives will provide substantial speed gains, so far I've implemented hybrid analytic and numeric derivatives.
- I should've released it by now...

Future directions

- Dynamic risk prediction, predictions will be a key focus of the `megenreg` engine
- location-scale models, multi-membership models...
- It's so general, and hence it can be slow(er)
- score and Hessian - analytic derivatives will provide substantial speed gains, so far I've implemented hybrid analytic and numeric derivatives.
- I should've released it by now...
- Crowther MJ. *Extended multivariate generalised linear and non-linear mixed effects models*. (Under review).
- Updates and tutorials here:
www.mjcrowther.co.uk/software/megenreg

References I

- [1] Li N, Elashoff RM, Li G. Robust joint modeling of longitudinal measurements and competing risks failure time data. *Biom J* Feb 2009; **51**(1):19–30, doi:10.1002/bimj.200810491. URL <http://dx.doi.org/10.1002/bimj.200810491>.
- [2] Rizopoulos D, Verbeke G, Lesaffre E, Vanrenterghem Y. A two-part joint model for the analysis of survival and longitudinal binary data with excess zeros. *Biometrics* 2008; **64**(2):pp. 611–619. URL <http://www.jstor.org/stable/25502097>.
- [3] Lin H, McCulloch CE, Mayne ST. Maximum likelihood estimation in the joint analysis of time-to-event and multiple longitudinal variables. *Stat Med* Aug 2002; **21**(16):2369–2382, doi:10.1002/sim.1179. URL <http://dx.doi.org/10.1002/sim.1179>.
- [4] Crowther MJ, Andersson TML, Lambert PC, Abrams KR, Humphreys K. Joint modelling of longitudinal and survival data: incorporating delayed entry and an assessment of model misspecification. *Statistics in medicine* 2016; **35**(7):1193–1209.

References II

- [5] Król A, Ferrer L, Pignon JP, Proust-Lima C, Ducreux M, Bouché O, Michiels S, Rondeau V. Joint model for left-censored longitudinal data, recurrent events and terminal event: Predictive abilities of tumor burden for cancer evolution with application to the ffcd 2000–05 trial. *Biometrics* 2016; **72**(3):907–916.
- [6] Barrett J, Su L. Dynamic predictions using flexible joint models of longitudinal and time-to-event data. *Statistics in Medicine* 2017; :n/a–n/adoi:10.1002/sim.7209. URL <http://dx.doi.org/10.1002/sim.7209>, sim.7209.
- [7] Crowther MJ, Abrams KR, Lambert PC, *et al.*. Joint modeling of longitudinal and survival data. *Stata J* 2013; **13**(1):165–184.
- [8] Król A, Mauguen A, Mazroui Y, Laurent A, Michiels S, Rondeau V. Tutorial in joint modeling and prediction: a statistical software for correlated longitudinal outcomes, recurrent events and a terminal event. *arXiv preprint arXiv:1701.03675* 2017; .
- [9] Philipson P, Sousa I, Diggle P, Williamson P, Kolamunnage-Dona R, Henderson R. *joineR - Joint Modelling of Repeated Measurements and Time-to-Event Data* 2012. URL <http://cran.r-project.org/web/packages/joineR/index.html>.

References III

- [10] Rizopoulos D. JM: An R Package for the Joint Modelling of Longitudinal and Time-to-Event Data. *J Stat Softw* 7 2010; **35**(9):1–33. URL <http://www.jstatsoft.org/v35/i09>.
- [11] Rizopoulos D. Jmbayes: joint modeling of longitudinal and time-to-event data under a bayesian approach 2015.
- [12] Henderson R, Diggle P, Dobson A. Joint modelling of longitudinal measurements and event time data. *Biostatistics* 2000; **1**(4):465–480.
- [13] Drukker DM, Gates R, et al.. Generating halton sequences using mata. *Stata Journal* 2006; **6**(2):214–228.
- [14] Dickman PW, Sloggett A, Hills M, Hakulinen T. Regression models for relative survival. *Stat Med* 2004; **23**(1):51–64, doi:10.1002/sim.1597. URL <http://dx.doi.org/10.1002/sim.1597>.
- [15] Liu L, Wolfe RA, Huang X. Shared frailty models for recurrent events and a terminal event. *Biometrics* 2004; **60**(3):747–756.
- [16] Mazroui Y, Mathoulin-Pelissier S, Soubeyran P, Rondeau V. General joint frailty model for recurrent event data with a dependent terminal event: application to follicular lymphoma data. *Statistics in medicine* 2012; **31**(11-12):1162–1176.

References IV

- [17] Goldstein H, Leckie G, Charlton C, Tilling K, Browne WJ. Multilevel growth curve models that incorporate a random coefficient model for the level 1 variance function. *Statistical methods in medical research* Jan 2017; :962280217706728doi:10.1177/0962280217706728.