The R package mixmeta

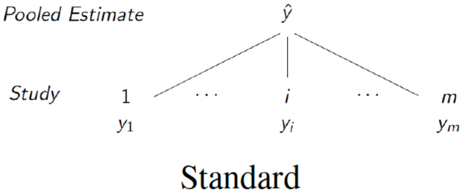
An extended mixed-effects framework for meta-analysis

Antonio Gasparrini[1,2] and Francesco Sera[1]

[1]Department of Public Health Environments and Society, LSHTM, London, UK
[2]Centre for Statistical Methodology, LSHTM, London, UK

useR!2019 Conference
Toulouse, France, 12 July 2019

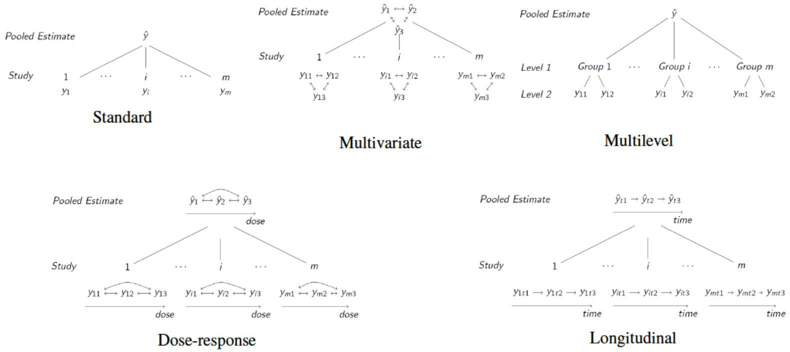
Standard meta-analytical framework



Assumptions:

- Single outcome y
- Single estimate (effect size) y_i from each study i
- Set of $i = 1, \ldots, m$ independent studies

Extended meta-analytical framework



The R package `mixmeta`

The package `mixmeta` consists of a collection of functions to perform various meta-analytical models in R through a unified mixed-effects framework, including standard univariate fixed and random-effects meta-analysis and meta-regression, and non-standard extensions such as multivariate, multilevel, longitudinal, and dose-response models.

Key features:

- Flexibility
- Simple syntax
- Computational efficiency

Mixed-effects framework

Model

$$\mathbf{y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \epsilon_i \ , \ i = 1, \ldots, m$$
$$\mathbf{b}_i \sim \mathrm{N}\left(\mathbf{0}, \mathbf{\Psi}_i\right) \ , \epsilon_i \sim \mathrm{N}\left(\mathbf{0}, \mathbf{S}_i\right) \ , \mathbf{\Sigma}_i = \mathbf{\Psi}_i + \mathbf{S}_i$$

Likelihood

$$\ell\left(\beta, \xi \mid \mathbf{y}\right) = -\tfrac{1}{2} n \log(2\pi) - \tfrac{1}{2}\sum_{i=1}^{m} \log\left|\mathbf{\Sigma}_i\right| - \tfrac{1}{2}\sum_{i=1}^{m}\left(\mathbf{y}_i - \mathbf{X}_i\beta\right)^{\mathrm{T}} \mathbf{\Sigma}_i^{-1}\left(\mathbf{y}_i - \mathbf{X}_i\beta\right)$$

Best linear unbiased predictions

$$\begin{aligned}\hat{\mathbf{y}}_{b_i} =& \mathbf{X}_0\hat{\beta} + \mathbf{Z}_i\hat{\mathbf{\Psi}}_i\mathbf{Z}_i^{\mathrm{T}}\hat{\mathbf{\Sigma}}_i^{-1}\left(\hat{\mathbf{y}}_i - \mathbf{X}_i\hat{\beta}\right)\\ V\left(\hat{\mathbf{y}}_{b_i}\right) =& \mathbf{X}_0 V(\hat{\beta})\mathbf{X}_0^{\mathrm{T}} + \mathbf{Z}_i\hat{\mathbf{\Psi}}_i\mathbf{Z}_i^{\mathrm{T}} - \mathbf{Z}_i\hat{\mathbf{\Psi}}_i\mathbf{Z}_i^{\mathrm{T}}\hat{\mathbf{\Sigma}}_i^{-1}\mathbf{Z}_i\hat{\mathbf{\Psi}}_i\mathbf{Z}_i^{\mathrm{T}}\end{aligned}$$

Cochran Q and $\mathbf{I}^2$

$$Q = \sum_{i=1}^{m}\left(\mathbf{y}_i - \mathbf{X}_i\hat{\beta}\right)^{\mathrm{T}} \mathbf{S}_i^{-1}\left(\mathbf{y}_i - \mathbf{X}_i\hat{\beta}\right)$$
$$I^2 = \max\left[(Q - n + p)/Q \ , \ 0\right]$$

LONDON SCHOOL of HYGIENE &TROPICAL MEDICINE

Estimation

Efficient and reliable computational strategies:

- Use of a **profile (restricted) likelihood** dependent only on the random-effects variance parameters
- Starting values obtained with few iterations of **(restricted) iterative generalized least square (IGLS)** algorithms
- Final minimization of the profile log-likelihood (ML or REML) using an iterative **Newton-Raphson-type** method via `optim()`
- Use of **QR decomposition** at the outer grouping level + ensuring positive-definiteness of (co)variance matrices through **Cholesky/eigen decomposition**
- Algorithms exploit the **block-diagonal structure** of the grouped data with the definition of lists

Syntax

Usage

```
mixmeta(formula, S, data, random, method="reml", bscov="unstr",
    offset, subset, contrasts=NULL, na.action, model=TRUE,
    control=list())
```

Call

```
mixmeta(cbind(y1 + y2) ~ x1 + x2, S, data, method="reml",
  random = list(~ z1 | g1, ~ 1 | g2))
```

Key aspects:

- Limited number of **arguments**
- Compact symbolic expressions through **formulae**
- **Common features** of regression functions

Structure

$\boxed{\texttt{mixmeta()}}$: general regression handler

$\searrow$

$\boxed{\texttt{mixmeta.fit()}}$: defines grouping, creates block-diagonal lists

$\searrow$

$\boxed{\texttt{mixmeta.reml()}}$: wrapper for REML estimator

$\searrow$

$\boxed{\texttt{reml.rigls()} + \texttt{rigls.iter()}}$: (R)IGLS iterative algorithm

$\searrow$

$\boxed{\texttt{reml.newton()} + \texttt{reml.loglik()}}$: Quasi-Newton iterative algorithm

LONDON SCHOOL of HYGIENE &TROPICAL MEDICINE

Methods & functions

Summary methods

`print()`, `summary()`

Terms-related methods

`terms()`, `model.frame()`, `model.matrix()`

Regression methods

`fitted()`, `residuals()`, `predict()`, `coef()`/`vco()v`, `logLik()`, `AIC()`/`BIC()`

Meta-analytical and simulation functions

`qtest()`, `blup()`, `mixmetaSim()`

Example I: the school dataset

Meta-analysis of **56 studies** that evaluate the effect of a modified school calendar on standardized reading achievement in 11 separate **districts**, with at least three studies in each district

district	study	effect	var	year
11	1	-0.18	0.12	1976
11	2	-0.22	0.12	1976
11	3	0.23	0.14	1976
11	4	-0.30	0.14	1976
12	5	0.13	0.01	1989
12	6	-0.26	0.01	1989
...	...	...	...	...
644	54	0.61	0.08	1995
644	55	0.04	0.07	1994
644	56	-0.05	0.07	1994

This represents a classic example of **multilevel structure**

Multilevel meta-anaysis

Usage

```
model <- mixmeta(effect, var, random= ~ 1 | district / study,
    data=school, method="ml")
```

Results

Model	Levels	Est. (SE)	Random		
			District	Study	AIC
Standard	Study	0.128 (0.043)	-	0.087	37.292
Repeated	District	0.196 (0.086)	0.075	-	69.432
Two-level	Study in district	0.184 (0.080)	0.058	0.033	22.790

LONDON SCHOOL of HYGIENE &TROPICAL MEDICINE

Example II: RCTs on gliomas treatment

Meta-analysis using data on **17 randomized controlled trials** comparing treatments of malignant gliomas. Each study measured the survival OR at **6, 12, 18, and 24 months** since the start of the treatment.

study	time	ntreat	ncontr	streat	scontr	logOR	logORvar
1	6.00	19	22	16	20	-0.63	0.95
1	12.00	19	22	11	12	0.14	0.40
1	18.00	19	22	4	8	-0.76	0.51
1	24.00	19	22	4	3	0.52	0.70
...	...	...	...	...	...	...	...
17	6.00	74	75	*NA*	*NA*	*NA*	*NA*
17	12.00	74	75	42	40	0.14	0.11
17	18.00	74	75	*NA*	*NA*	*NA*	*NA*
17	24.00	74	75	23	30	-0.39	0.12

This represents a classic example of **longitudinal structure**

Longitudinal meta-anaysis – I

Comparison of two models:

- The **traditional approach**, casting the problem as a multivariate model
- A more **flexible approach**, treating time as continuous

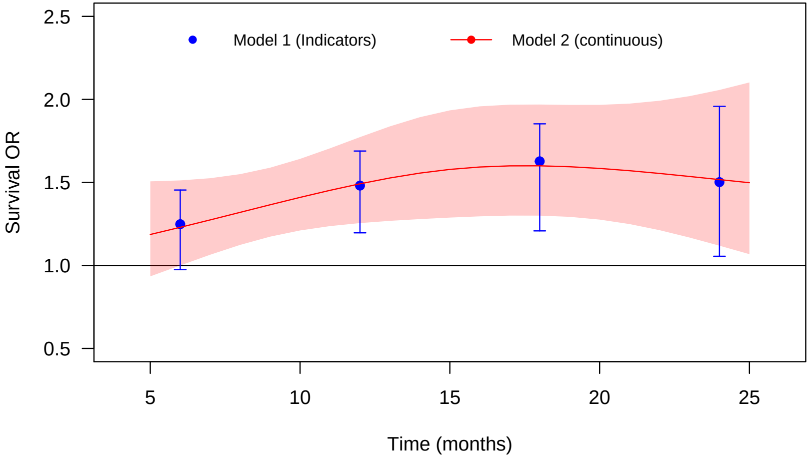
Usage

```
model1 <- mixmeta(logOR ~ 0 + factor(time), S=logORvar, data=gliomas,
    random= ~ 0 + factor(time) | study, bscov="diag")
pred1 <- exp(predict(model1, data.frame(time=unique(times)), ci=TRUE))

model2 <- mixmeta(logOR ~ ns(time, 2), S=logORvar, data=gliomas,
     random= ~ time | study, bscov="diag")
pred2 <- exp(predict(model2, data.frame(time=5:25), ci=TRUE))
```

Longitudinal meta-anaysis – II

Results

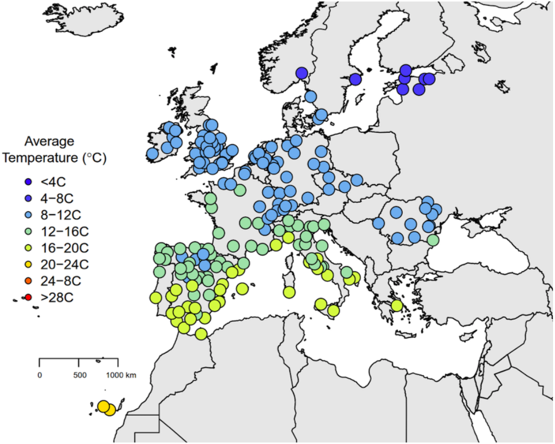


Example III: temperature and mortality

Daily time series data on temperature and mortality from **163 cities** in **17 countries** in Europe (1980-2016)

First-stage city-specific regression estimate multiple non-linear exposure-response relationships with splines

Second-stage meta-analysis to pool the spline coefficients with a series of meta-regressors

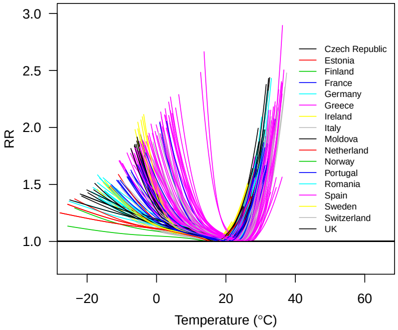


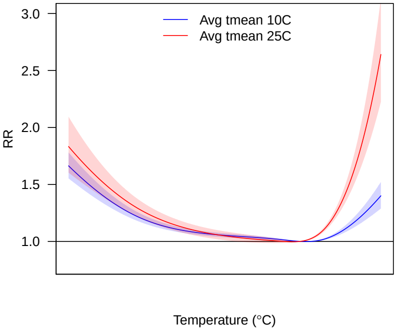
Multivariate multilevel meta-anaysis

Usage

```
model <- mixmeta(splcoef ~ kgclzone1 + gdp + avgtmean + rangetmean,
    splvcov, cities, random=~1 | country / city)
```

Results





Comments

The package `mixmeta` provides a unified and flexible framework for meta-analysis for standard and non-standard meta-analytical models

Comparison with other implementations:

- Function `rma.mv()` in package `metafor`
- Function `lme()` in package `nlme`

Current developments:

- Vectorized version of (R)IGLS algorithms
- Implementation of additional (co)variance structures (spatial, temporal)
- Development of inferential methods

Links & references

Development version in GitHub

https://github.com/gasparrini/mixmeta

Article in submission

Sera F, Gasparrini A. (2019). An extended mixed-effects framework for meta-analysis. In submission.

Contacts

Email: antonio.gasparrini@lshtm.ac.uk
Twitter: @AGasparrini75