The R package mixmeta
An extended mixed-effects framework for meta-analysis

Antonio Gasparrini\textsuperscript{1,2} and Francesco Sera\textsuperscript{1}

\textsuperscript{1}Department of Public Health Environments and Society, LSHTM, London, UK
\textsuperscript{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

Standard

Multivariate

Multilevel

Dose-response

Longitudinal
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

\[ y_i = X_i \beta + Z_i b_i + \epsilon_i , \quad i = 1, \ldots, m \]

\[ b_i \sim N(0, \Psi_i) , \quad \epsilon_i \sim N(0, S_i) , \Sigma_i = \Psi_i + S_i \]

Likelihood

\[ \ell (\beta, \xi \mid y) = -\frac{1}{2} n \log(2\pi) - \frac{1}{2} \sum_{i=1}^{m} \log |\Sigma_i| - \frac{1}{2} \sum_{i=1}^{m} (y_i - X_i \beta)^T \Sigma_i^{-1} (y_i - X_i \beta) \]

Best linear unbiased predictions

\[ \hat{y}_{bi} = X_0 \hat{\beta} + Z_i \hat{\Psi}_i Z_i^T \hat{\Sigma}_i^{-1} (\hat{y}_i - X_i \hat{\beta}) \]

\[ V (\hat{y}_{bi}) = X_0 V (\hat{\beta}) X_0^T + Z_i \hat{\Psi}_i Z_i^T - Z_i \hat{\Psi}_i Z_i^T \hat{\Sigma}_i^{-1} Z_i \hat{\Psi}_i Z_i^T \]

Cochran Q and I^2

\[ Q = \sum_{i=1}^{m} (y_i - X_i \hat{\beta})^T S_i^{-1} (y_i - X_i \hat{\beta}) \]

\[ I^2 = \max [(Q - n + p) / Q , \ 0] \]
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**

- `mixmeta()`: general regression handler
  - `mixmeta.fit()`: defines grouping, creates block-diagonal lists
  - `mixmeta.reml()`: wrapper for REML estimator
    - `reml.rigls() + rigls.iter()`: (R)IGLS iterative algorithm
    - `reml.newton() + reml.loglik()`: Quasi-Newton iterative algorithm
Methods & functions

Summary methods
print(), summary()

Terms-related methods
terms(), model.frame(), model.matrix()

Regression methods
fitted(), residuals(), predict(), coef()/vco(), 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.

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

This represents a classic example of multilevel structure.
Multilevel meta-analysis

Usage

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

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Levels</th>
<th>Est. (SE)</th>
<th>Random</th>
<th></th>
<th></th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>District</td>
<td>Study</td>
<td>AIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>Study</td>
<td>0.128 (0.043)</td>
<td>-</td>
<td>0.087</td>
<td>37.292</td>
<td></td>
</tr>
<tr>
<td>Repeated</td>
<td>District</td>
<td>0.196 (0.086)</td>
<td>0.075</td>
<td>-</td>
<td>69.432</td>
<td></td>
</tr>
<tr>
<td>Two-level</td>
<td>Study in district</td>
<td>0.184 (0.080)</td>
<td>0.058</td>
<td>0.033</td>
<td>22.790</td>
<td></td>
</tr>
</tbody>
</table>
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.

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

This represents a classic example of longitudinal structure.
Comparison of two models:

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

**Usage**

```r
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-analysis – II

Results

Survival OR

Model 1 (Indicators) Model 2 (continuous)
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-analysis

Usage

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

Results
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
Development version in GitHub

https://github.com/gasparrini/mixmeta

Article in submission


Contacts

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