Introduction to Weighted Mixed Models With WeMix

Developed by Paul Bailey, Claire Kelley, and Trang Nguyen [*†]

August 10, 2018

Introduction

The `WeMix` package fits a **We**ighted **Mix**ed model, also known as a multilevel, mixed, or hierarchical linear model (HLM). The weights could be inverse selection probabilities, such as those developed for an education survey where schools are sampled probabilistically, and then students inside of those schools are sampled probabilistically. Although mixed-effects models are already available in R, `WeMix` is unique in implementing methods for mixed models using weights at multiple levels.

The package `lme4` fits these models when there are no weights or weights only for first-level units (Bates, Maechler, Bolker, & Walker, 2015) and is recommended for those situations—by default, `WeMix` uses the `lme4` results as a starting point. `WeMix` adds the ability to fit models with weights at every level of the model, similar to GLLAMM (Rabe-Hesketh, Skrondal, & Pickles, 2002; Rabe-Hesketh, Skrondal, & Pickles, 2005; Rabe-Hesketh & Skrondal, 2006) and maximizes the same likelihood function as GLLAMM. Because the model applies weights at every level, the units must be nested in "levels", so `WeMix` only fits models where the units have a nested structure.

Installing and Loading `WeMix`

Inside R, run the following command to install `WeMix`:

```
install.packages("WeMix")
```

Once the package is successfully installed, `WeMix` can be loaded with the following command:

```
library(WeMix)
```

Specifying a Mixed-Effects Model

To illustrate the functionality of WeMix, we will use an example based on publicly available data from the Programme for International Student Assessment (PISA) 2012 data from the United States (OECD, 2013). PISA is a repeated multinational assessment of skills and attitudes of 15-year-olds, with students (the unit of observation) probabilistically sampled within schools (the second-level unit) that are also probabilistically sampled within the country. In the United States, there were 3,136 student observations in 157 schools. We provide examples of a model with a random intercept and a model with both a random slope and intercept.

The first model can be specified as the math assessment predicted by a few variables chosen from the PISA survey:

- Dependent variable: `pv1math`, a math assessment score
- Independent variables:
  - `escs`: a continuous Socio-Economic Status index

[*]This publication was prepared for NCES (National Center for Education Statistics) under Contract No. ED-IES-12-D-0002 with American Institutes for Research. Mention of trade names, commercial products, or organizations does not imply endorsement by the U.S. government.

[†]The authors would like to thank Mike Cohen and Dan Sherman for reviewing this document, and Yuqi Liao and Bitnara Park for their help with use of mixed models in other software packages.

  - `sc14q02` school questionnaire item: "Is your school's capacity to provide instruction hindered by any of the following... A lack of qualified mathematics teachers." (levels: "A lot"; "To some extent"; "Very little"; "Not at all", the reference gorup)
  - `st04q02` student gender (levels: "Male" and "Female", the reference group)
  - `st29q03` student questionnaire item: "I look forward to math lessons" (levels: "Strongly agree"; "Agree"; "Disagree"; "Strongly disagree", the reference group)
- School level random effect: school intercept
- Student level weights: `pwt1`
- School level weights: `pwt2`

In the second model, a second random effect is added at the school level for the `escs` variable, but there is no covariance between the two random effects.

Where estimation is done by quadrature, 27 quadrature points are used for the one random effect model and 13 for the two random effects model.

An appendix describes the transformation used to prepare the data for analysis.

The WeMix call for this model, using a `data.frame` containing the PISA 2012 data for the United States and named "data", would be:

```
# model with one random effect
mix(pv1math ~ st29q03 + sc14q02 +st04q01+escs+ (1|schoolid), data=data,
    weights=c("pwt1", "pwt2"), nQuad=27, verbose=FALSE, fast=TRUE,run=TRUE)

# model with two random effects assuming zero correlation between the two
mix(pv1math ~ st29q03 + sc14q02 +st04q01+escs+ (1|schoolid)+ (0+escs|schoolid), data=data,
    weights=c("pwt1", "pwt2"), nQuad=13, verbose=FALSE, fast=TRUE,run=TRUE)
```

Note that the syntax for the model formula is the same syntax one would use to specify a model using `lme4`. Thus, in the random slope and intercept model, the slope and intercept are in separate terms in order to constrain their covariance to be zero.

Comparison to Alternate Software

For readers familiar with specification of other software, this section shows results in comparison with those from Stata, SAS, and HLM. When possible, the code specifies a random slope model and then a random slope and intercept model with the covariance of slope and intercept fixed to be zero. The models are fitted by maximum likelihood estimation and example code in Stata GLLAMM, Stata MIXED, SAS GLIMMIX, and HLM are shown in the appendix.

Table 1 shows the results of the model with a single random effect where `WeMix`, GLLAMM, Stata MIXED, and SAS show agreement on the likelihood, variance estimates of random effects, and fixed effects. HLM normalizes the weights (for both students and schools) and so produces somewhat different results. Although the estimates are very similar (as reported here to the fith digit), there are some differences in the standard errors of the random effects. Most notably, Stata MIXED calculates a lower standard error for the random intercept than other methods, and HLM does not calculate standard errors for random effects. All the programs differ somewhat in the standard errors estimated for the fixed effects, and `WeMix` most closely matches the results from GLLAMM.

Table 2 shows the results of the model with two random effects. Here the results are similar.`WeMix`, GLLAMM, Stata MIXED, and SAS show agreement on the likelihood, variance term estimates of random effects, and fixed effects. HLMs fit a different model and so get a different result. Similar to the one random effect model, Stata MIXED reports a lower standard error for the random intercept. It is, however, noteworthy that Stata MIXED reports a higher standard error for the random slope than the other methods. In terms of the standard errors of the fixed effects, there are differences between the estimates of all the programs, and `WeMix` again most closely matches GLLAMM.

Table 1: Results for Random Intercept Model					
	WeMix	Stata: GLLAMM	Stata: MIXED	SAS	HLM[1]
Run Time	00:31	02:00	00:30	00:03	00:10
Log-Likelihood	-2578035.4	-2578035.4	-2578035.4	-2578035.5	-17965.4
Random Effects					
Var Random Intercept	1106.6	1106.6	1106.6	1106.6	1020.0
Var Residual	5109.5	5109.5	5109.5	5109.5	5061.1
Standard Error (SE) of Random Effects					
SE Random Intercept	289.88	289.88	280.80	288.95	Not Calculated
SE Residual	202.53	202.53	201.88	201.88	Not Calculated
Fixed Effects					
(Intercept)	494.73	494.73	494.73	494.73	494.47
st29q03: Agree	-24.115	-24.115	-24.115	-24.115	-24.119
st29q03: Strongly disagree	-54.678	-54.678	-54.678	-54.678	-54.788
st29q03: Disagree	-26.040	-26.040	-26.040	-26.040	-26.109
sc14q02: A lot	-24.507	-24.507	-24.507	-24.507	-24.504
sc14q02: To some extent	-10.787	-10.787	-10.787	10.787	-10.362
sc14q02: Very little	-18.035	-18.035	-18.035	-18.035	-19.505
st04q01: Male	12.127	12.127	12.127	12.127	12.164
escs	28.729	28.729	28.729	28.729	28.332
Standard Error of Fixed Effects					
SE (Intercept)	7.1817	7.1818	7.0885	7.1586	8.6259
SE st29q03: Agree	7.2196	7.2196	7.2182	7.1966	7.1401
SE st29q03: Strongly disagree	12.234	12.2335	12.261	12.1945	12.108
SE st29q03: Disagree	6.3425	6.3425	6.3558	6.3223	6.2607
SE sc14q02: A lot	8.1732	8.1732	8.1871	8.1471	7.2026
SE sc14q02: To some extent	13.646	13.646	13.528	13.603	13.587
SE sc14q02: Very little	14.020	14.020	14.116	13.975	15.285
SE st04q01: Male	3.7203	3.7203	3.7305	3.7085	3.7008
SE escs	2.6071	2.6071	2.5621	2.5988	2.5249

[1] HLM requires that the weights are normalized before they are fit so results do not match exactly.
Also, HLM cannot set slope and intercept covariance to 0 (Raudenbush, Bryk, Cheong, Congdon, & Toit, 2016).

Table 2: Results for Random Intercept and Slope Model					
	WeMix	Stata: GLLAMM	Stata: MIXED	SAS	HLM[1]
Run Time	3:42	20:00	00:30	00:10	00:10
Log-Likelihood	-25777729.6	-25777729.6	-25777729.6	-2577729.5	-17964.0
Random Effects					
Var of Intercept	1075.1	1075.1	1075.1	1075.11	984.87
Var of Slope of escs	94.036	94.034	94.035	94.036	56.277
Var of Residual	5048.6	5048.6	5048.6	5048.6	5010.3
Standard Error (SE) of Random Effects					
SE Random Intercept	271.05	271.06	265.38	270.19	Not Calculated
SE Slope of escs	80.129	80.130	82.31	79.873	Not Calculated
SE Residual	185.63	185.63	184.57	185.04	Not Calculated
Fixed Effects					
(Intercept)	494.22	494.22	494.22	494.22	494.03
st29q03: Agree	-23.993	-23.993	-23.993	-23.993	-24.012
st29q03: Strongly disagree	-54.330	-54.330	-54.330	-54.330	-54.471
st29q03: Disagree	-25.813	-25.813	-25.813	-25.813	-27.782
sc14q02: A lot	-28.232	-28.232	-28.232	-28.232	-27.782
sc14q02: To some extent	-11.680	-11.680	-11.680	-11.680	-11.048
sc14q02: Very little	-18.076	-18.076	-18.075	-18.076	-19.513
st04q01: Male	12.230	12.230	12.230	12.230	12.252
escs	29.096	29.096	29.096	29.096	28.625
Standard Error of Fixed Effects					
SE (Intercept)	7.1473	7.1473	7.0273	7.1243	7.1488
SE st29q03: Agree	7.2000	7.2000	7.1979	7.1770	7.1235
SE st29q03: Strongly disagree	12.324	12.434	12.340	12.395	12.179
SE st29q03: Disagree	6.3242	6.3242	6.2991	6.3040	7.3716
SE sc14q02: A lot	7.8182	7.8181	7.2185	7.7931	7.3716
SE sc14q02: To some extent	13.622	13.621	13.407	13.5782	13.497
SE sc14q02: Very little	13.728	13.728	13.835	13.6838	15.008
SE st04q01: Male	3.7252	3.7252	3.7381	3.7134	3.7084
SE escs	2.6056	2.6056	2.6763	2.5973	2.6105

[1] HLM requires that the weights are normalized before they are fit, so results do not match exactly.
Also, HLM cannot set slope and intercept covariance to 0 (Raudenbush, Bryk, Cheong, Congdon, & Toit, 2016).

Mathematical Specification

This section describes the mathematical methodology behind the estimation of Weighted Mixed models in `WeMix`.

The simplest version of this model has two levels and is of the form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon} \ , \tag{1}$$

where $\boldsymbol{y}$ is the vector of outcomes, $\boldsymbol{X}$ is s a matrix of covariates associated with regressors that are assumed to be fixed, $\boldsymbol{\beta}$ is the vector of fixed-effect regression coefficients, $\boldsymbol{Z}$ is a matrix of covariates associated with regressors that are assumed to be random, and $\boldsymbol{u}$ is the vector of random effects. The meaning of $\boldsymbol{u}$ being random is simply that a level is shared within a group and across groups $\boldsymbol{u} \sim MVN(\vec{\mathbf{0}}, \boldsymbol{\Sigma})$, where $MVN(\cdot, \cdot)$ is the multivariate-normal distribution, $\vec{\mathbf{0}}$ is the mean vector of all zeros, and $\boldsymbol{\Sigma}$ is the covariance matrix of the MVN.

Hierarchical Linear Models Notation

The models `WeMix` fits can also be called hierarchical linear model (HLMS; Radenbush & Bryk, 2002) where the first level is of the form:

$$y_{ij} = \beta_{0j} + X\beta_{1j} + \epsilon_{ij} \ , \tag{2}$$

So, the second level could then be, for a random slope and intercept model:

$$\beta_{0j} = \gamma_{00} + \delta_{0j} \tag{3}$$

$$\beta_{1j} = \gamma_{01} + \delta_{1j} \ , \tag{4}$$

where δ_{0j} and δ_{1j} are the error terms for the intercept and slope, respectively, and have variances of τ_{00} and τ_{11}, respectively, and δ_{0j} and δ_{1j} have covariance τ_{01}.

This is just one example; many other models can also be fit in `WeMix`. WeMix can fit models that are stated as an HLM or as a mixed model. For notational convenience for the rest of this document, we will use the non-HLM mixed model notation.

Multiple Levels

When there are more than two levels, eq. 1 can be rewritten as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \sum_{l=2}^{L} \boldsymbol{Z}^{(l)}\boldsymbol{u}^{(l)} + \boldsymbol{\epsilon} \tag{5}$$

where a superscript (l) is added to $\boldsymbol{Z}$ and $\boldsymbol{u}$ to indicate that they are at the lth level. Note: In the summation above, l starts at $l = 2$ because there cannot be random effects at the lowest level of observation ($l = 1$).

Note that `WeMix 2.0` does not support three or more levels. Three-level model fitting will be added to the future `WeMix 3.0`.

Model Fitting

The central concern in `WeMix` is properly incorporating sampling weights into the mixed model. Because each individual or group may have an unequal probability of selection into the sample, the estimate of the distribution of the MVN must include those weights to correctly estimate the parameters of the distribution. Also, except for trivial cases, the nested nature of the multilevel model creates a likelihood function that is not analytically calculable.

Table 1: Notation

$\boldsymbol{\theta}$	vector of all of the fit model parameters fit, including, $\boldsymbol{\beta}$ and the nonduplicated elements of the covariate matrices $\boldsymbol{\Sigma}^{(l)}$. Because they are integrated out, $\boldsymbol{u}$ is included in $\boldsymbol{\theta}$.
$\boldsymbol{\Sigma}^{(l)}$	covariance matrix for level l
$\boldsymbol{y}$	response vector
$\boldsymbol{X}$	covariate matrix for fixed effects
$\boldsymbol{\beta}$	coefficients of the fixed effects
$\boldsymbol{Z}^{(l)}$	covariate matrix for random effects at level l
$\boldsymbol{Z}$	covariate matrix for all random effects
$\boldsymbol{u}^{(l)}$	vector of the random effects at level l
$\boldsymbol{U}^{(l)}$	vector of the random effects at all levels l and higher, $(u^l, ..., u^L)'$
$\boldsymbol{w}^{(l)}$	vector of the weights at level l
$\boldsymbol{W}$	matrix of the weights at all levels
k_l	number of random effects at level l
L	number of levels in the model
i	subscript denoting the individual
j	subscript denoting group
j'	subscript denoting a group within j
$\mathcal{L}^{(l)}$	likelihood function at level l
$\ell^{(l)}$	log-likelihood function at level l
$\boldsymbol{\epsilon}$	regression residuals, net of fixed and random effects

We consider the likelihood as a summation at each level, starting with the lowest level. The likelihood is conditional on the random effects at each higher level and is scaled by the weights at the lowest level.

In the case of the normal distribution (the only link function currently implemented in WeMix), the likelihood ($\mathcal{L}^{(1)}$) at the individual unit level is given by the equations below. Note that here the subscript i is used to indicate that this is the the likelihood of the i^{th} individual.

$$\mathcal{L}_i^{(1)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{U}^{(l)}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}) = \frac{1}{\Sigma^{(1)}} \phi\left(\frac{\hat{e_i}}{\Sigma^{(1)}}\right) \tag{6}$$

Where $\phi(\cdot)$ is the standard normal density function and $\Sigma^{(1)}$ is the residual variance (a scalar). The residuals vector $\hat{\boldsymbol{e}}$ represents the residuals $\hat{e}_i$ for each individual, which is calculated as:

$$\hat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \sum_{l=2}^{L} \boldsymbol{Z}^{(l)} \hat{\boldsymbol{u}}^{(l)} \ , \tag{7}$$

The conditional likelihood at each higher level is then recursively defined, for the jth unit, at level l ($\mathcal{L}_j^{(l)}$; Rabe-Hesketh,Skrondal & Pickles, 2002, eq. 3):

$$\mathcal{L}_j^{(l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{U}^{(l+1)}) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} g^{(l)}(\boldsymbol{u}^{(l)}) \prod_{j'} \left[\mathcal{L}_{j'}^{(l-1)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{U}^{(l)})\right]^{\boldsymbol{w}_{j'}^{(l-1)}} du_1^{(l)} ... du_{k_l}^{(l)} \tag{8}$$

where the subscript j' that the product is indexed over indicates that the likelihood $\mathcal{L}_{j'}^{(l-1)}(\cdot)$ is for the units of level $l-1$ nested in unit j. Additionally, $g(\boldsymbol{u}^{(l)})$ is the empirical Bayes "prior" probability density of the random effects (at level l) having a value of $u^{(l)}$, so $g(\cdot)$ is a multivariate normal distribution parameterized by a mean of 0 and variance Σ. The integrals are over the elements of $\boldsymbol{u} = (u_1, ..., u_{k_l})$, where k_l is the number of random effects at level l. It is important to note that each $\mathcal{L}^{(l)}$ is independent for each j'. This allows us to integrate out the values of u for all groups simultaneously, which leaves us with a k_l dimensional integral and is essential to making this problem computationally feasible. At the highest level, the result is not conditional on any $\boldsymbol{u}$ values, but is otherwise the same.

For ease of computation and in order to avoid problems with accurate storage of extremely small numbers we first take the log of the function before maximizing. The log-likelihood function is (Rabe-Hesketh 2006, eq. 1):

$$\ell_j^{(l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{U}^{(l+1)}) = ln\left\{\int ... \int g^{(l-1)}(u^{(l-1)}) \cdot \exp\left[\sum_{j'} \boldsymbol{w}_{j'}^{(l-1)} \ell_{j'}^{(l-l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|U^{(l)})\right] du_1^{(l)} ... du_{k_l}^{(l)}\right\} \tag{9}$$

where $\ell_j^{(l)}(\cdot)$ is the log-likelihood function for the jth unit at level l.

Unfortunately, there is no closed-form expression for the integral in (10), and so we must evaluate the likelihood numerically. This is possible in with adaptive Gauss-Hermite quadrature, which is a modification of Gauss-Hermite quadrature.

First, to evaluate the integral at level l, we must evaluate the integral over k_l random effects variables that have a covariance matrix $\boldsymbol{\Sigma}^{(l)}$. To avoid dependence, we use a change of variables to create independent and standard normal distributed vectors $v^{(l)}$. Using the Cholesky decomposition $\boldsymbol{\Sigma}^{(l)} = \boldsymbol{C}^{(l)}\left(\boldsymbol{C}^{(l)}\right)^T$, the value of $u^{(l)}$ can then be calculated as $u^{(l)} = \boldsymbol{C}^{(l)}\boldsymbol{v}^{(l)}$.

Adaptive Gauss-Hermite Quadrature

Using the transformation of variables described above,

$$\ell_j^{(l)}(\cdot) = \int ... \int g^{(l)}(u^{(l)}) \cdot \exp\left[\sum_{j'} \boldsymbol{w}_{j'}^{(l-1)} \ell_{j'}^{(l-l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|U^{(l)})\right] du^{(l)} \tag{10}$$

$$= \int \phi(v_{k_l}^{(l)}) ... \int \phi(v_1^{(l)}) \cdot \exp\left[\sum_{j'} \boldsymbol{w}_{j'}^{(l-1)} \ell_{j'}^{(l-l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|U^{(l)})\right] du^{(l)} \ , \tag{11}$$

where $u^{(l)}$ is now a function of the v values and ϕ is the standard normal density.

Quadrature replaces integration with a summation over a finite set of points (known as quadrature points) and weights so that the sum approximates the integral—we annotate the quadrature points with a tilde so that, for example, u becomes $\tilde{u}$.

These equations come from Rabe-Hesketh et al. (2002, p.5 eq. 4) and follow from eq. 11.

$$\ell_j^{(l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{U}^{(l+1)})) = \sum_{r_1=1}^{R} p_{r_1}^{(l)} ... \sum_{r_{k_l}=1}^{R} p_{r_{k_l}}^{(l)} \cdot \exp\left[\sum_{j'} \boldsymbol{w}_{j'}^{(l-1)} \ell_{j'}^{(l-l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\tilde{\boldsymbol{u}}^{(l)}, \boldsymbol{U}^{(l+1)})\right] \ , \tag{12}$$

where R is the number of quadrature points, $\boldsymbol{p}$ are the quadrature weights, $\tilde{\boldsymbol{u}}^{(l)} = C^{(l)}\tilde{\boldsymbol{v}}^{(l)}$ are the quadrature point locations for each of the random effect vectors. This results in a grid with R quadrature points per element in u (and v), for a total of R^{k_l} quadrature points, and the summation is over every point in that grid. The quadrature points and weights come from the `statsmod` implementation of gaussian quadrature (Smyth 1998).

While Gauss-Hermite quadrature centers the quadrature points on the prior distribution, adaptive Gauss-Hermite quadrature (AGHQ) centers the quadrature points on either the likelihood surface or the posterior distribution. We use the posterior maximum (the likelihood function of which , including $g(\cdot)$), is detailed below, but this section is general to AGHQ as detailed in Lui and Pierce 1994, and Hartzel, Agresti, and Caffo, 2001; we use notation similar to Hartzel et al., 2001, p.87. The goal of adaptive quadrature is to

reduce the number of quadrature points needed for an accurate evaluation of the integral by centering the points closer to the bulk of the integral. The location of the points is scaled based on the values of the likelihood function. To do this, the mode of the likelihood is used to center the points, and the dispersion is the estimated standard deviation of the likelihood at that point (using the inverse second derivative).

$$\tilde{\boldsymbol{u}} = \hat{\boldsymbol{\omega}}_j + \sqrt{2}\hat{\boldsymbol{Q}}_j^{1/2}\tilde{\boldsymbol{v}} \tag{13}$$

where $\tilde{\boldsymbol{u}}_i^{(l)}$ are the quadrature points, $\hat{\boldsymbol{\omega}}_i$ is a vector of locations for group j, and $\hat{\boldsymbol{Q}}_j$ is the matrix that is the inverse numerical second derivative of the likelihood function evaluated at the unit normal *iid* points $\boldsymbol{v}$.

We can then use the adapted quadrature points to calculate the log likelihood as:

$$\ell_j^{(l)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{U}^{(l+1)}) = \\ \sum_{r_1=1}^{R} p_{r_1}^{(l)} ... \sum_{r_{k_l}=1}^{R} p_{r_{k_l}}^{(l)} \cdot \exp\left[\sum_{j'} \boldsymbol{w}_{j'}^{(l-1)} \ell_{j'}^{(l-1)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{XZ}, \boldsymbol{W}|\tilde{u}_{r_1} ... \tilde{u}_{r_{k_l}}, \boldsymbol{U}^{(l+1)}) + g(\tilde{\boldsymbol{u}}^{(l)}; \boldsymbol{\Sigma}^{(l)}) + \boldsymbol{v}^T\boldsymbol{v}\right] \tag{14}$$

This approximation of the likelihood can then be evaluated and minimized numerically. In this package, we minimize the function using Newton's method.

Calculation of Conditional Mode

AGHQ requires an estimated location ($\hat{\boldsymbol{\omega}}_j$) and variance $\boldsymbol{Q}_j$ for each group. These are calculated by iteratively finding the conditional mode of the random effects (to find $\hat{\boldsymbol{\omega}}_j$) and then using the inverse of the second derivative of the likelihood surface as an estimate of $\boldsymbol{Q}_j$.

The conditional modes are identified by sequentially increasing levels of l; the software first identifies the MAP at level 2, and then, using those estimates, uses AGHQ at level 2 to estimate conditional modes at level 3, and so on.

For each group, the conditional mode is identified using Newton's method on the likelihood for that unit ($\ell_j^{(l)}(\cdot)$) at a particular level of $u^{(l)}$, called $\hat{u}$, and conditional on an existing estimate of θ,

$$\ell_j^{(l)}(\hat{\boldsymbol{u}}^{(l)}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{\theta}) = ln\left[g^{(l)}(\hat{\boldsymbol{u}}^{(l)}) \sum_{j'} \boldsymbol{w}_{j'}^{(l-1)} \ell_{j'}^{(l-1)}(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\hat{\boldsymbol{u}}^{(l)})\right] \tag{15}$$

where this formulation implicitly sets all values of $\boldsymbol{U}^{(l)}$ to zero. Note that the values of $\ell_{j'}^{(l-l)}$ still integrate out the values of $\boldsymbol{u}$ for all levels below l.

Newton's method requires a first and second derivative, and these are calculated with numerical derivatives of the likelihood calculated using numerical quadrature.

Estimate of the Conditional Means

The conditional mean is estimated by simply taking the expected value of each parameter using

$$E\left(\hat{\boldsymbol{u}}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\theta}\right) = \frac{\int_{-\infty}^{\infty} \tilde{\boldsymbol{u}}^{(l-1)} \cdot \ell_j^{(l)}(\hat{\boldsymbol{u}}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{\theta}) d\hat{u}}{\int_{-\infty}^{\infty} \ell_j^{(l)}(\tilde{\boldsymbol{u}}^{(l-1)}; \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{\theta}) d\hat{u}} \tag{16}$$

Variance Estimation of Random Effects: Sandwich Estimator

The variance estimator was calculated following the method of Rabe-Hesketh & Skrondal, 2006 . Thus, the variance is expressed as:

$$var(\boldsymbol{u}) = \boldsymbol{I^{-1}JI^{-1}} \tag{17}$$

where $\boldsymbol{I}$ is the pseudo Fisher information and calculated by observing the Fisher information at the maximum likelihood estimates. Given that the likelihood is twice differentiable, we estimate the Fisher information as the second derivative (Hessian) of the likelihood evaluated at the maximum likelihood point.

$$\boldsymbol{I} = \frac{\partial^2 \mathcal{L}(\cdot)}{\partial \theta^2} \tag{18}$$

and

$\boldsymbol{J}$ is estimated as

$$\boldsymbol{J} = \sum_{L-1} \frac{n_{L-1}}{n_{L-1}-1} \sum_{j'} \frac{\partial \mathcal{L}_j^L(\cdot)}{\partial \theta} \cdot \left[\frac{\partial \mathcal{L}_j^L(\cdot)}{\partial \theta}\right]^T \tag{19}$$

where n_{L-1} represents the number of observations in the $(L-1)^{th}$ level (i.e., the second from the top level). And the subscript $L-1$ indicates that the outermost sum is over the units j' in $(L-1)^{th}$ level.

Here, the derivative of the likelihood function is calcuated numerically and evaluated at the maximum likelihood point estimates.

Hierarchical Generalized Linear Models

If data comes in nested strucutre and the assumptions of linearity and normality cannot be met even after applying transformations, Hierarchical Generalized Linear Models (HGLM) offer a possible solution. There are three components of a HGLM model: a sampling model, a link function, and a structural model (Raudenbush & Bryk, 2002). Currently `WeMix 2.0` supports two models: Gaussian (or normal) sampling model with identity link function (the default), and Binomial sampling model with logit link function (for binary outcomes.) Different models can be specified using `family` argument in `mix` function. Note that the first model (Gaussian with identity link) is the default so there is no need to specify `family` argument.

Binomial With Logit Link Function

This model is used when the data has binary outcomes (i.e. the number of successes in m trials). Here is an example code using **sleepstudy** dataset in `lme4` package:

```
library(lme4)
library(WeMix)
ss1 <- sleepstudy
doubles <- c(308, 309, 310) # subject with double obs

# Create weights
ss1$W1 <- ifelse(ss1$Subject %in% doubles, 2, 1)
ss1$W2 <- 1

# Create binary outcome variable called "over300"
ss1$over300 <- ifelse(sleepstudy$Reaction<300,0,1)
```

```
# Run mixed model with random intercept and fixed slope
bi_1 <- mix(over300~ Days + (1|Subject),data=ss1,
            family=binomial(link="logit"),verbose=FALSE,
            weights=c("W1", "W2"),nQuad=13)
```

Table 3 shows the output comparison between `WeMix` and STATA `GLLAMM`.

Table 3: Results for Binomial Model with Logit Link		
	WeMix	Stata: GLLAMM
Log-Likelihood	-93.75179	93.751679
Random Effects		
Var Random Intercept	4.127208	4.1335015
Standard Error (SE) of Random Effects		
SE Random Intercept	2.966179	2.9815704
Fixed Effects		
(Intercept)	-3.34411	-3.344987
Days	0.59271	.5928508
Standard Error of Fixed Effects		
SE (Intercept)	0.92390	.9250737
SE Days	0.12429	.1244216

Weight Adjustments

`WeMix` assumes that the weights are already scaled. However, for informational purposes, we describe here two common methods of scaling.

If there were no total nonresponses at any level, the ideal weights would simply be the inverse selection probabilities (Pfeffermann, Skinner, Holmes, Goldstein, & Rasbash, 1998). But, because the samples are adjusted based on demographics, alternative weighting schemes are encouraged in the literature. Carle (2009) and Rabe-Hesketh et al. (2002) recommend that sample weights be scaled; they cannot simply be the raw inverse probability of selection because this fails to adequately prevent bias when cluster sizes differ. The notation is consistent with Carle, 2009.

Method A:

$$w_{ij}^{*} = w_{ij}\left(\frac{n_j}{\sum_i w_{ij}}\right) , \tag{20}$$

where w_{ij} are the full sample weights, i indexes the individuals, j indexes the groups, and n_j represents the number of observations in group j.

Method B:

$$w_{ij}^{*} = w_{ij}\left(\frac{\sum_i w_{ij}}{\sum_i w_{ij}^2}\right) \tag{21}$$

These w^{*} are then used to scale the likelihood function.

Centering

`WeMix 2.0` allows users to incorporate grand- or group-mean centering when fitting mixed-effects models. In the group-mean centered model, the predictors are centered around the group-level mean. Compared to Equation (2), the model can be expressed as follows:

$$y_{ij} = \beta_{0j} + \beta_{1j}(X_{ij} - \overline{X_{.j}}) + \epsilon_{ij} , \tag{22}$$

and the intercept can be interpreted as the unadjusted mean for group j:

$$\beta_{0j} = \mu_{Y_j} \tag{23}$$

In the grand-mean centered model, the predictors are centered around the overall mean so the model can be written as follows:

$$y_{ij} = \beta_{0j} + \beta_{1j}(X_{ij} - \overline{X_{..}}) + \epsilon_{ij} \ , \tag{24}$$

and the intercept can be interpreted as the adjusted mean for group j:

$$\beta_{0j} = \mu_{Y_j} - \beta_{1j}(X_{ij} - \overline{X_{..}}) \tag{25}$$

There are two main advantages of centering the predictors. First of all, centering makes estimates of level-1 coefficients (β_{0j}) and other effects easier to interpret because it decomposes the relationship between predictors and outcome into within-group and between-group components. Second, it removes high correlations between the random intercept and slopes, as well as high correlations between first- and second-level predictors and cross-level interactions (Kreft & de Leeuw, 1998).

The following example shows how to implement group or grand mean centering.

```
library(lme4)  #to use example sleep study data

#create dummy weights
sleepstudy$weight1L1 <- 1
sleepstudy$weight1L2 <- 1

# Group mean centering of the variable Days within group Subject
group_center <- mix(Reaction ~ Days + (1|Subject), data=sleepstudy,
                    center_group=list("Subject"= ~Days),
                    weights=c("weight1L1", "weight1L2"), nQuad=13,
                    verbose=FALSE, fast=TRUE,run=TRUE)

# Grand mean centering of the variable Days
grand_center <- mix(Reaction ~ Days + (1|Subject), data=sleepstudy,
                    center_grand=~Days,weights=c("weight1L1", "weight1L2"),
                    nQuad=13, verbose=FALSE, fast=TRUE,run=TRUE)
```

Appendix: Alternative Software Specifications

For reference, these sections show the specification of the models in Stata's GLLAMM, Stata's MIXED, SAS PROC GLIMMIX, and HLM.

Stata: GLLAMM

In Stata prior to version 14, weighted mixed-effects models could be estimated only with GLLAMM (Rabe-Hesketh, Skrondal, & Pickles, 2004). The work of GLLAMM authors Rabe-Hesketh, Skrondal, and Pickles provided the methods that we used in our implementation of weighted mixed models in `WeMix`.

```
import delimited "PISA2012_USA.csv"

generate intercept = 1
eq intercept: intercept
eq slope: escs
```

```
tabulate st29q03, generate (st29q03d)
tabulate sc14q02, generate (sc14q02d)
tabulate st04q01, generate (st04q01d)

//Random intercept model
gllamm pv1math st29q03d1 st29q03d2 st29q03d4 sc14q02d1 sc14q02d3 sc14q02d4 st04q01d2
escs, i(schoolid) pweight(pwt) l(identity) f(gau) nip(27) nrf(1) eqs(intercept)
adapt nocorrel


//Random slope and intercept model
gllamm pv1math st29q03d1 st29q03d2 st29q03d4 sc14q02d1 sc14q02d3 sc14q02d4 st04q01d2
escs, i(schoolid) pweight(pwt) l(identity) f(gau) nip(13) nrf(2) eqs(intercept slope)
adapt nocorrel
```

Stata: MIXED

In Stata Version 14, the MIXED command includes the ability to fit models with survey weights (StataCorp, 2015).

```
import delimited "PISA2012_USA.csv"
tabulate st29q03, generate (st29q03d)
tabulate sc14q02, generate (sc14q02d)
tabulate st04q01, generate (st04q01d)

//Random intercept model
mixed pv1math st29q03d1 st29q03d2 st29q03d4 sc14q02d1 sc14q02d3 sc14q02d4 st04q01d2
escs [pw = pwt1] || schoolid: , pweight (pwt2)

//Random slope and intercept model
mixed pv1math st29q03d1 st29q03d2 st29q03d4 sc14q02d1 sc14q02d3 sc14q02d4 st04
q01d2 escs [pw = pwt1] || schoolid: escs, pweight (pwt2)
```

SAS

Model specification in SAS uses the GLIMMIX procedure. It is notable here that when fit with the default optimization parameters, the model converged to a likelihood lower than the maximum likelihood estimate found by other software. Decreasing the convergence parameter GCONV to E-10 was necessary to find the same maximum likelihood as other software.

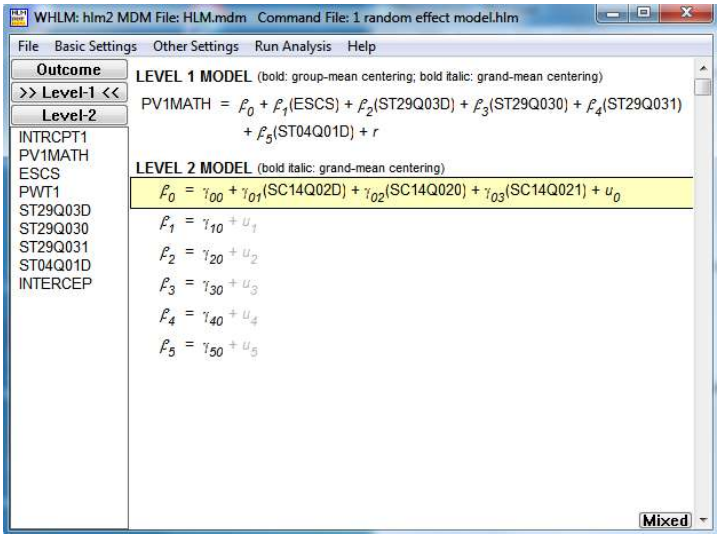
```
proc import datafile="PISA2012_USA.csv"
     out=pisa_data
     dbms=csv
     replace;
run;

proc glimmix data=pisa_data method=quadrature(qpoints=27) empirical=classical NOREML;
  nloptions GCONV=1E-10 technique=TRUREG;
    class  sc14q02(ref='Not at all') st04q01(ref='Female') st29q03(ref='Strongly agree');
    model pv1math = escs sc14q02 st04q01 st29q03/  obsweight=pwt1 solution;
    random INT/subject=schoolid weight=pwt2;
run;
```

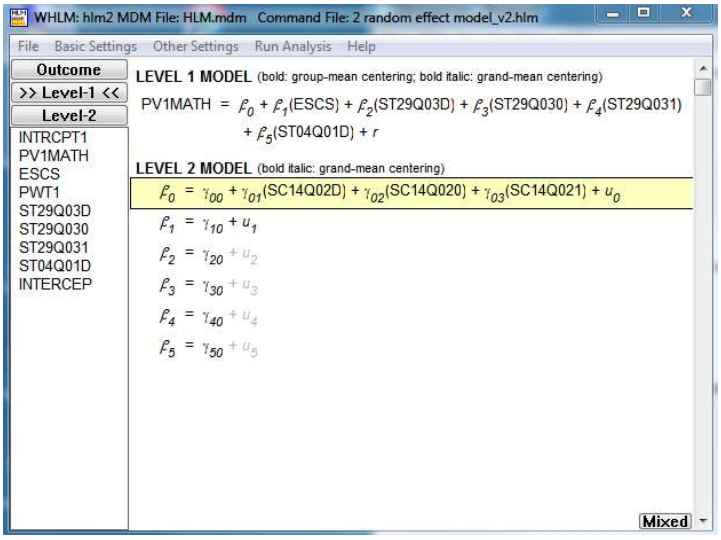
```
proc glimmix data=pisa_data method=quadrature(qpoints=13) empirical=classical NOREML;
  nloptions GCONV=1E-10 technique=TRUREG;
  class  sc14q02(ref='Not at all') st04q01(ref='Female') st29q03(ref='Strongly agree');
  model pv1math = escs sc14q02 st04q01 st29q03/ obsweight=pwt1 solution;
  random intercept escs/subject=schoolid   weight=pwt2;
run;
```

HLM

HLM is another software package for estimated mixed-effects models (Raudenbush, Bryk, Cheong, Congdon, & Toit, 2016). It is important to note that HLM has two differences from the methods specified in other softwares. HLM normalizes all weights (which other programs do not) and also does not allow the correlation between slope and random effect to be fixed at 0. Using the "Diagonalize Tau" option reduces covariance, but does not fix it at 0 (Raudenbush et al., 2016). In addition, HLM is entirely graphical user interface (GUI) based. Specification of the HLM model for comparison here was done through the interface. The random intercept model was specified as:



And the random slope and intercept model was specified as:



Note: The specifications for the random intercept model and the random slope model are extremely similar; in the second image for β_1, the u_1 is highlighted. This is how users in HLM add random effects.

Appendix: Example Data Preparation

Data are read in using the `EdSurvey` package to access the PISA data efficiently.

```
library(EdSurvey)

#read in data
cntl <- readPISA([path], countries = "USA")
om <- getAttributes(cntl, "omittedLevels")
data <- getData(cntl, c("schoolid","pv1math","st29q03","sc14q02","st04q01","escs","w_fschwt","w_fstuwt"

#prepare weights
data$sqw <- data$w_fstuwt^2
sumsqw <- aggregate(sqw ~ schoolid, data = data, sum)
sumw <- aggregate(w_fstuwt ~ schoolid, data = data, sum)
data$sumsqw <- sapply(data$schoolid, function(s) sumsqw$sqw[sumsqw$schoolid == s])
data$sumw <- sapply(data$schoolid, function(s) sumw$w_fstuwt[sumw$schoolid == s])
data$pwt1 <- data$w_fstuwt * (data$sumw / data$sumsqw)
data$pwt2 <- data$w_fschwt

# Remove NA and omitted Levels
om <- c("Invalid","N/A","Missing","Miss",NA,"(Missing)")
for (i in 1:ncol(tempData)) {
  tempData <- tempData[!tempData[,i] %in% om,]
}

#relevel factors for model
data$st29q03 <- relevel(data$st29q03,ref="Strongly agree")
data$sc14q02 <- relevel(data$sc14q02,ref="Not at all")
```

Citations

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. Journal of Statistical Software, 67(1), 1-48.

Carle, A. C. (2009). Fitting multilevel models in complex survey data with design weights: Recommendations. BMC Medical Research Methodology, 9(1).

Hartzel, J., Agresti, A., & Caffo, B. (2001). Multinomial logit random effects models. Statistical Modelling: An International Journal, 1(2), 81-102.

Kreft, I. G., & de Leeuw, J. (1998). Introducing Statistical Methods: Introducing multilevel modeling. London, : SAGE Publications Ltd.

Liu, Q., & Pierce, D. A. (1994). A note on Gauss-Hermite quadrature. Biometrika, 81(3), 624.

OECD. (2013), PISA 2012 assessment and analytical framework: Mathematics, reading, science, problem solving and financial literacy. Paris, France. OECD Publishing. Retrieved from http://www.oecd.org/pisa/pisaproducts/PISA%202012%20framework%20e-book_final.pdf

Pfeffermann, D., Skinner, C., Holmes, D., Goldstein, H., & Rasbash, J. (1998). Weighting for unequal selection probabilities in multilevel models. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 60(1), 23-40.

Rabe-Hesketh, S., & Skrondal, A. (2006). Multilevel modelling of complex survey data. Journal of the Royal Statistical Society. Series A (Statistics in Society), 169(4), 805-827.

Rabe-Heskteh, S., Skrondal, A., & Pickles, A. (2004). GLLAMM manual (Working Paper No. 160). Berkeley, CA: University of California, Berkeley, Division of Biostatistics.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2005). Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. Journal of Econometrics, 128(2), 301-323.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2002). Reliable estimation of generalized linear mixed models using adaptive quadrature. Stata Journal, 2, 1-21.

Raudenbush, S. W., & Bryk, A. S. (2002). Hierarchical linear models: Applications and data analysis methods (2nd ed.). Thousand Oaks, CA: SAGE Publications.

Raudenbush, S. W., Bryk, A., Cheong, Y. F., & Congdon, R. (2005). HLM 6: Hierarchical linear and nonlinear modeling. Lincolnwood, IL: Scientific Software International.

Raudenbush, S. W., Bryk, A. S., Cheong, Y. F., Congdon, R. T., & Toit, M. (2016). HLM7 hierarchical linear and nonlinear modeling user manual: User guide for Scientific Software International's (S.S.I.) Program. Lincolnwood, IL: Scientific Software International.

SAS Institute, Inc. (2013). SAS/ACCESS 9.4 interface to ADABAS: Reference. Cary, NC: Author.

Scientific Software International. (2016). Design weighting in the hierarchical context. Retrieved from www.ssicentral.com/hlm/example6-2.html

Smyth, G. K. (1998). Numerical integration. In P. Armitage & T. Colton (Eds.), Encyclopedia of Biostatistics (pp. 3088-3095). London, UK: Wiley.

StataCorp. (2015). Stata Statistical Software: Release 14. College Station, TX: Author.