Online Appendix A.

Online Appendix A provides the derivation for the denominator expression of manuscript Equations (8), (9), and (10)--i.e., the model-implied total variance of y_{ij} using the general mixture model in Equation (7). Slight modifications of this derivation yield the denominator expression in Equations (3), (4), and (5), i.e., the model-implied total variance of y_i using the simpler special case mixture model in Equation (1). Specifically, the below derivation also applies to the Equations (3), (4), and (5) expressions after the following replacements are made: replace all *ij* subscripts with *i*, replace all *kh* superscripts with *k*, replace all $\bullet\bullet$ superscripts with $\bullet$, and remove $d_j=h$ from all operations.

As defined in the manuscript: *i* = individual, *j* = cluster, c_{ij} = level-1 latent classification variable with classes $k = 1\ldots K$, d_j = level-2 latent classification variable with classes $h = 1\ldots H$, $\mathbf{x}'_{ij}$ = a vector of 1 and exogenous predictors, $\boldsymbol{\gamma}^{kh}$ = a vector of regression coefficients (intercepts and slopes) specific to class-combination *kh*. We denote the total model-implied variance of y_{ij} as $\text{var}_{ij}(y_{ij})$. In taking the variance across *ij*, as in $\text{var}_{ij}(.)$, we are equivalently taking the variance across all *kh*, (i.e., $\text{var}_{kh}(.)$) because each individual *i* within cluster *j* is a member of a class-combination *kh*. We will use the *ij* subscript throughout the derivation for simplicity. Relatedly, let $E_{ij}(.)$ denote the expectation across *i* and *j*.

To begin, using an extended application of the law of total variance (see, e.g., Bowsher & Swain, 2012's Equation (13)) $\text{var}_{ij}(y_{ij})$ can be expressed as Equation (A.1):

$$\begin{aligned}\text{var}_{ij}(y_{ij}) &= E_{ij}[\text{var}_{ij}(y_{ij} \mid \mathbf{x}'_{ij},(c_{ij}=k,d_j=h))]+\\ &E_{ij}[\text{var}_{ij}(E_{ij}[y_{ij} \mid \mathbf{x}'_{ij},(c_{ij}=k,d_j=h)] \mid \mathbf{x}'_{ij})]+\text{var}_{ij}(E_{ij}[y_{ij} \mid \mathbf{x}'_{ij}])\end{aligned} \quad \text{(A.1)}$$

First we will show how Equation (A.1) can be written as a function of quantities in the data model of manuscript Equation (7), yielding Equation (A.2):

$$\text{var}(y_{ij}) = E_{ij}[\mathbf{x}'_{ij}E_{ij}[(\boldsymbol{\gamma}^{kh}-\boldsymbol{\gamma}^{\bullet\bullet})(\boldsymbol{\gamma}^{kh}-\boldsymbol{\gamma}^{\bullet\bullet})']\mathbf{x}_{ij}]+E_{ij}[(\mathbf{x}'_{ij}\boldsymbol{\gamma}^{\bullet\bullet})^2]-E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{\bullet\bullet}]^2+\theta^{\bullet\bullet} \quad \text{(A.2)}$$

Subsequently we will show how Equation (A.2) can be re-expressed as the denominator of Equations (8)-(10), i.e., Equation (A.3).

$$\text{var}(y_{ij}) = \mathbf{s}'\boldsymbol{\psi}+2\mathbf{p}'\boldsymbol{\kappa}+\boldsymbol{\gamma}^{\bullet\bullet\prime}\boldsymbol{\Phi}\boldsymbol{\gamma}^{\bullet\bullet}+\theta^{\bullet\bullet}\,. \quad \text{(A.3)}$$

Steps involved in re-expressing Equation (A.1) as Equation (A.2) are as follows:

The first term, $E_{ij}[\text{var}_{ij}(y_{ij} \mid \mathbf{x}'_{ij},(c_{ij}=k,d_j=h))]$, is simply the expected value of the residual variance.

$$\begin{aligned}&E_{ij}[\text{var}_{ij}(y_{ij} \mid \mathbf{x}'_{ij},(c_{ij}=k,d_j=h))]\\ &=E_{ij}[\theta^{kh}]\\ &=\theta^{\bullet\bullet}\end{aligned}$$

The second term, $E_{ij}[\text{var}_{ij}(E_{ij}[y_{ij} \mid \mathbf{x}'_{ij},(c_{ij}=k,d_j=h)] \mid \mathbf{x}'_{ij})]$, simplifies as follows:

$$E_{ij}[\text{var}_{ij}(E_{ij}[y_{ij} \mid \mathbf{x}'_{ij}, (c_{ij} = k, d_j = h)] \mid \mathbf{x}'_{ij})]$$
$$= E_{ij}[\text{var}_{ij}(\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} \mid \mathbf{x}'_{ij})]$$
$$= E_{ij}[\mathbf{x}'_{ij}\ \text{var}_{ij}(\boldsymbol{\gamma}^{kh})\mathbf{x}_{ij}]$$
$$= E_{ij}[\mathbf{x}'_{ij}E_{ij}[(\boldsymbol{\gamma}^{kh} - \ddot{\boldsymbol{\gamma}})(\boldsymbol{\gamma}^{kh} - \ddot{\boldsymbol{\gamma}})']\mathbf{x}_{ij}]$$

The third term, $\text{var}_{ij}(E_{ij}[y_{ij} \mid \mathbf{x}'_{ij}])$, simplifies as follows:

$$\text{var}_{ij}(E_{ij}[y_{ij} \mid \mathbf{x}'_{ij}])$$
$$= \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} + \varepsilon_{ij} \mid \mathbf{x}'_{ij}])$$
$$= \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} \mid \mathbf{x}'_{ij}] + \cancel{E_{ij}[\varepsilon_{ij} \mid \mathbf{x}'_{ij}]})$$
$$= \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} \mid \mathbf{x}'_{ij}])$$
$$= \text{var}_{ij}(\mathbf{x}'_{ij}E_{ij}[\boldsymbol{\gamma}^{kh}])$$
$$= \text{var}_{ij}(\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}})$$
$$= E_{ij}[(\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}})^2] - E_{ij}[\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}}]^2$$

Note that, above, the exogenous $\mathbf{x}'_{ij}$ is a constant with respect to the conditional expectation but not with respect to the variance. Thus, the total variance of y_{ij} can be expressed as Equation (A.2), i.e.:

$$\text{var}(y_{ij}) = E_{ij}[\mathbf{x}'_{ij}E_{ij}[(\boldsymbol{\gamma}^{kh} - \ddot{\boldsymbol{\gamma}})(\boldsymbol{\gamma}^{kh} - \ddot{\boldsymbol{\gamma}})']\mathbf{x}_{ij}] + E_{ij}[(\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}})^2] - E_{ij}[\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}}]^2 + \ddot{\theta}$$

In the remainder of Online Appendix A we show how this Equation (A.2) is equal to the denominator of Equations (8)-(10) (i.e., $\mathbf{s}'\boldsymbol{\psi} + 2\mathbf{p}'\boldsymbol{\kappa} + \ddot{\boldsymbol{\gamma}}'\boldsymbol{\Phi}\ddot{\boldsymbol{\gamma}} + \ddot{\theta}$ from Equation (A.3) above).

We first show that $E_{ij}[(\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}})^2] - E_{ij}[\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}}]^2 = \ddot{\boldsymbol{\gamma}}'\boldsymbol{\Phi}\ddot{\boldsymbol{\gamma}}$, as follows

$$E_{ij}[(\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}})^2] - E_{ij}[\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}}]^2$$
$$= \text{var}_{ij}(\mathbf{x}'_{ij}\ddot{\boldsymbol{\gamma}})$$
$$= \text{var}_{ij}(\ddot{\boldsymbol{\gamma}}'\mathbf{x}_{ij})$$
$$= \ddot{\boldsymbol{\gamma}}'\ \text{var}_{ij}(\mathbf{x}_{ij})\ddot{\boldsymbol{\gamma}}$$
$$= \ddot{\boldsymbol{\gamma}}'\boldsymbol{\Phi}\ddot{\boldsymbol{\gamma}}$$

We next show that $E_{ij}[\mathbf{x}'_{ij}E_{ij}[(\boldsymbol{\gamma}^{kh} - \ddot{\boldsymbol{\gamma}})(\boldsymbol{\gamma}^{kh} - \ddot{\boldsymbol{\gamma}})']\mathbf{x}_{ij}] = \mathbf{s}'\boldsymbol{\psi} + 2\mathbf{p}'\boldsymbol{\kappa}$ as follows

$$E_{ij}[\mathbf{x}'_{ij}E_{ij}[(\boldsymbol{\gamma}^{kh}-\boldsymbol{\gamma}^{\cdot\cdot})(\boldsymbol{\gamma}^{kh}-\boldsymbol{\gamma}^{\cdot\cdot})']\mathbf{x}_{ij}]$$
$$=E_{ij}[\mathbf{x}'_{ij}\ \mathrm{var}_{ij}(\boldsymbol{\gamma}^{kh})\mathbf{x}_{ij}]$$
$$=E_{ij}[\mathbf{x}'_{ij}\underbrace{(\mathbf{V}+\mathbf{C})}_{\substack{\mathbf{V}\text{ = diag matrix of }\boldsymbol{\gamma}^{kh}\text{ variances}\\ \mathbf{C}\text{ = cov matrix of }\boldsymbol{\gamma}^{kh}\text{; 0s on diag}}}\mathbf{x}_{ij}]$$
$$=E_{ij}[\mathbf{x}'_{ij}\mathbf{V}\mathbf{x}_{ij}+\mathbf{x}'_{ij}\mathbf{C}\mathbf{x}_{ij}]$$
$$=\underbrace{E_{ij}[\mathbf{x}'_{ij}\mathbf{V}\mathbf{x}_{ij}]}_{\text{see below: \#1}}+\underbrace{E_{ij}[\mathbf{x}'_{ij}\mathbf{C}\mathbf{x}_{ij}]}_{\text{see below: \#2}}$$

#1

$$E_{ij}[\mathbf{x}'_{ij}\mathbf{V}\mathbf{x}_{ij}]$$
$$=E_{ij}[tr(\mathbf{x}'_{ij}\mathbf{V}\mathbf{x}_{ij})]$$
$$=E_{ij}[tr(\mathbf{x}_{ij}\mathbf{x}'_{ij}\mathbf{V})]$$
$$=E_{ij}[tr((\mathbf{x}_{ij}\mathbf{x}'_{ij})_D\mathbf{V}+\cancel{(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}\mathbf{V}})]$$

Where $\mathbf{x}_{ij}\mathbf{x}'_{ij}$ is equal to its diagonal elements, $(\mathbf{x}_{ij}\mathbf{x}'_{ij})_D$, plus its off diagonal elements, $(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}$. Note that the term $(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}\mathbf{V}$ cancels because $\mathbf{V}$ is diagonal.

$$=E_{ij}[tr((\mathbf{x}_{ij}\mathbf{x}'_{ij})_D\mathbf{V})]$$
$$=tr(E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_D]\mathbf{V})$$
$$=\sum_t[E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_D]]_{tt}[\mathbf{V}]_{tt}$$
$$=\sum_t[\mathbf{s}']_t[\boldsymbol{\psi}]_t$$
$$=\mathbf{s}'\boldsymbol{\psi}$$

That is, because both $E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_D]$ and $\mathbf{V}$ are diagonal, the sum of products of diagonal elements is the same as the inner product of two vectors. Thus, $\mathbf{s}'$ is a row vector of means of squared elements of $\mathbf{x}_{ij}$ and $\boldsymbol{\psi}$ is a column vector containing diagonal elements of $\mathbf{V}$.

#2

$$E_{ij}[\mathbf{x}'_{ij}\mathbf{C}\mathbf{x}_{ij}]$$
$$=E_{ij}[tr(\mathbf{x}'_{ij}\mathbf{C}\mathbf{x}_{ij})]$$
$$=E_{ij}[tr(\mathbf{x}_{ij}\mathbf{x}'_{ij}\mathbf{C})]$$
$$=E_{ij}[tr(\cancel{(\mathbf{x}_{ij}\mathbf{x}'_{ij})_D\mathbf{C}}+(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}\mathbf{C})]$$

Note that $(\mathbf{x}_{ij}\mathbf{x}'_{ij})_D\mathbf{C}$ cancels because $\mathbf{C}$ has 0's on diagonal.

$$= E_{ij}[tr((\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}\mathbf{C})]$$
$$= tr(E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}\mathbf{C}])$$
$$= \sum_{t,u}[E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}]]_{tu,t\neq u}[\mathbf{C}]_{ut,t\neq u}$$

Because the diagonals of $\mathbf{C}$ are 0, it follows that only off-diagonal elements of $\mathbf{x}_{ij}\mathbf{x}'_{ij}$ and $\mathbf{C}$ need to be considered (i.e., $u \neq t$).

$$= \sum_{t,u}[E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}]]_{tu,t\neq u}[\mathbf{C}]_{tu,t\neq u}$$

Because **C** is symmetric, it follows that $[\mathbf{C}]_{ut} = [\mathbf{C}]_{tu}$.

$$= 2\sum_{t,u}[E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}]]_{tu,u<t}[\mathbf{C}]_{tu,u<t}$$

Because $\mathbf{x}_{ij}\mathbf{x}'_{ij}$ and $\mathbf{C}$ are both symmetric, only the lower-triangular elements of $\mathbf{x}_{ij}\mathbf{x}'_{ij}$ and $\mathbf{C}$ need to be considered (*u*<*t*), twice.

$$= 2\sum_{g}[\mathbf{p}']_g[\boldsymbol{\kappa}]_g$$
$$= 2\mathbf{p}'\boldsymbol{\kappa}$$

That is, we are renaming the typical lower-triangular elements of matrix $E_{ij}[(\mathbf{x}_{ij}\mathbf{x}'_{ij})_{OD}]$ and of matrix $\mathbf{C}$ to be typical elements of vector $\mathbf{p}'$ and of vector $\boldsymbol{\kappa}$, respectively. Thus, $\mathbf{p}'$ is a row vector of means for the pairwise products of all nonredundant elements of $\mathbf{x}_{ij}$ and $\boldsymbol{\kappa}$ is a column vector of lower-triangular elements of $\mathbf{C}$.

Online Appendix B

Online Appendix B provides the derivation for the numerator of manuscript Equation (9), i.e., $\mathbf{v}'\boldsymbol{\psi} + 2\mathbf{r}'\boldsymbol{\kappa} + \boldsymbol{\gamma}^{\bullet\bullet\prime}\boldsymbol{\Phi}\boldsymbol{\gamma}^{\bullet\bullet}$. Slight modifications of this derivation yield the numerator of manuscript Equation (4), as a special case. Specifically, the below derivation also applies to Equation (4) after the following replacements are made: replace all *ij* subscripts with *i*, replace all *kh* superscripts with *k*, replace all $\bullet\bullet$ superscripts with $\bullet$, and remove $d_j = h$ from all operations.

Please see Online Appendix A for definitions of these terms: *i*, *j*, c_{ij}, *k*, d_j, *h*, $\mathbf{x}'_{ij}$, $\boldsymbol{\gamma}^{kh}$, $\mathbf{s}'\boldsymbol{\psi}$, $2\mathbf{p}'\boldsymbol{\kappa}$, $\boldsymbol{\gamma}^{\bullet\bullet\prime}\boldsymbol{\Phi}\boldsymbol{\gamma}^{\bullet\bullet}$, and $\theta^{\bullet\bullet}$. Note that $\text{var}_{ij}(.)$ indicates taking the variance across *ij*, or equivalently across all *kh*, (i.e., $\text{var}_{kh}(.)$) because each individual *i* within cluster *j* is a member of a class-combination *kh*. We will use the *ij* subscript throughout the derivation for simplicity. Relatedly, let $E_{ij}(.)$ denote the expectation across *i* and *j*.

The numerator of $R_T^{2(fv)}$ (from Equation (9)), is here newly denoted $\text{var}_{ij}(\hat{y}_T^{fv})$. As stated in the manuscript, it involves subtracting the variance of the model-implied class-combination means of y_{ij} from the explained portion of variance from $R_T^{2(fvm)}$.[1]

$$\text{var}_{ij}(\hat{y}_T^{fv}) = \mathbf{s}'\boldsymbol{\psi} + 2\mathbf{p}'\boldsymbol{\kappa} + \boldsymbol{\gamma}^{\bullet\bullet\prime}\boldsymbol{\Phi}\boldsymbol{\gamma}^{\bullet\bullet} - \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} + \varepsilon_{ij} \mid (c_{ij} = k, d_j = h)]) \tag{B.1}$$

First we will show that Equation (B.1) can be expressed as Equation (B.2):

$$\text{var}_{ij}(\hat{y}_T^{fv}) = \mathbf{s}'\boldsymbol{\psi} + 2\mathbf{p}'\boldsymbol{\kappa} + \boldsymbol{\gamma}^{\bullet\bullet\prime}\boldsymbol{\Phi}\boldsymbol{\gamma}^{\bullet\bullet} - (\mathbf{a}'\boldsymbol{\psi} + 2\mathbf{q}'\boldsymbol{\kappa}) \tag{B.2}$$

Finally, we will show that Equation (B.2) can be re-expressed as Equation (B.3), which is the numerator of manuscript Equation (9).

$$\text{var}_{ij}(\hat{y}_T^{fv}) = \mathbf{v}'\boldsymbol{\psi} + 2\mathbf{r}'\boldsymbol{\kappa} + \boldsymbol{\gamma}^{\bullet\bullet\prime}\boldsymbol{\Phi}\boldsymbol{\gamma}^{\bullet\bullet} \tag{B.3}$$

To begin, we show how $\text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} + \varepsilon_{ij} \mid (c_{ij} = k, d_j = h)])$ from Equation (B.1) = $\mathbf{a}'\boldsymbol{\psi} + 2\mathbf{q}'\boldsymbol{\kappa}$ from Equation (B.2). The former can be simplified as follows:

$$\begin{aligned}
&\text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} + \varepsilon_{ij} \mid (c_{ij} = k, d_j = h)]) \\
&= \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} \mid (c_{ij} = k, d_j = h)] + \cancel{E_{ij}[\varepsilon_{ij} \mid (c_{ij} = k, d_j = h)]}) \\
&= \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}\boldsymbol{\gamma}^{kh} \mid (c_{ij} = k, d_j = h)]) \\
&= \text{var}_{ij}(E_{ij}[\mathbf{x}'_{ij}]\boldsymbol{\gamma}^{kh})
\end{aligned}$$

---

[1] This is analytically derived but is akin to literally fitting a null model with the same number of classes and with intercepts fixed to model-implied means from the full model, and subtracting its explained variance from the explained portion of variance from $R_T^{2(fvm)}$ (i.e., Equation (8)).

Note that, above, $\boldsymbol{\gamma}^{kh}$ is a constant with respect to the conditional expectation but not with respect to the variance. Now designate $\mathbf{m}$ as a vector of means of the exogenous $\mathbf{x}_{ij}$, i.e., $\mathbf{m} = E_{ij}[\mathbf{x}'_{ij}]$.

$$= \mathbf{m}' \operatorname{var}_{ij}(\boldsymbol{\gamma}^{kh})\mathbf{m}$$

$$= \mathbf{m}' \underbrace{(\mathbf{V}+\mathbf{C})}_{\substack{\mathbf{V} \text{ = diag matrix of } \boldsymbol{\gamma}^{kh} \text{ variances} \\ \mathbf{C} \text{ = cov matrix of } \boldsymbol{\gamma}^{kh}\text{; 0s on diag}}} \mathbf{m}$$

$$= \underbrace{\mathbf{m}'\mathbf{V}\mathbf{m}}_{\text{see below: \#1}} + \underbrace{\mathbf{m}'\mathbf{C}\mathbf{m}}_{\text{see below: \#2}}$$

#1

$$\mathbf{m}'\mathbf{V}\mathbf{m}$$
$$= tr(\mathbf{m}'\mathbf{V}\mathbf{m})$$
$$= tr(\mathbf{m}\mathbf{m}'\mathbf{V})$$
$$= tr((\mathbf{m}\mathbf{m}')_D\mathbf{V} + \cancel{(\mathbf{m}\mathbf{m}')_{OD}\mathbf{V}})$$

Where $\mathbf{m}\mathbf{m}'$ is equal to its diagonal elements, $(\mathbf{m}\mathbf{m}')_D$, plus its off diagonal elements, $(\mathbf{m}\mathbf{m}')_{OD}$. (Note that the term $(\mathbf{m}\mathbf{m}')_{OD}\mathbf{V}$ cancels because $\mathbf{V}$ is diagonal).

$$= tr((\mathbf{m}\mathbf{m}')_D\mathbf{V})$$
$$= \sum_t [(\mathbf{m}\mathbf{m}')_D]_{tt}[\mathbf{V}]_{tt}$$
$$= \sum_t [\mathbf{a}']_t[\boldsymbol{\psi}]_t$$
$$= \mathbf{a}'\boldsymbol{\psi}$$

That is, because both $(\mathbf{m}\mathbf{m}')_D$ and $\mathbf{V}$ are diagonal, the sum of products of diagonal elements is the same as the inner product of two vectors. Thus, $\mathbf{a}'$ is a row vector of squared means of $\mathbf{x}_{ij}$ and $\boldsymbol{\psi}$ is a column vector containing diagonal elements of $\mathbf{V}$.

#2

$$\mathbf{m}'\mathbf{C}\mathbf{m}$$
$$= tr(\mathbf{m}'\mathbf{C}\mathbf{m})$$
$$= tr(\mathbf{m}\mathbf{m}'\mathbf{C})$$
$$= tr(\cancel{(\mathbf{m}\mathbf{m}')_D\mathbf{C}} + (\mathbf{m}\mathbf{m}')_{OD}\mathbf{C})$$

Note that $(\mathbf{m}\mathbf{m}')_D\mathbf{C}$ cancels because $\mathbf{C}$ has 0's on diagonal.

$$= tr((\mathbf{m}\mathbf{m}')_{OD}\mathbf{C})$$
$$= \sum_{t,u} [(\mathbf{m}\mathbf{m}')_{OD}]_{tu,t\neq u}[\mathbf{C}]_{ut,t\neq u}$$

Because the diagonals of $\mathbf{C}$ are 0, it follows that only off-diagonal elements of $\mathbf{mm}'$ and $\mathbf{C}$ need to be considered (i.e., $u \neq t$).

$$= \sum_{t,u} [(\mathbf{mm}')_{OD}]_{tu,t\neq u} [\mathbf{C}]_{tu,t\neq u}$$

Because $\mathbf{C}$ is symmetric, it follows that $[\mathbf{C}]_{ut} = [\mathbf{C}]_{tu}$.

$$= 2\sum_{t,u} [(\mathbf{mm}')_{OD}]_{tu,u<t} [\mathbf{C}]_{tu,u<t}$$

Because $\mathbf{mm}'$ and $\mathbf{C}$ are both symmetric, only the lower-triangular elements of $\mathbf{mm}'$ and $\mathbf{C}$ need to be considered (*u*<*t*), twice.

$$= 2\sum_{g} [\mathbf{q}']_{g} [\boldsymbol{\kappa}]_{g}$$

$$= 2\mathbf{q}'\boldsymbol{\kappa}$$

That is, we are renaming the typical lower-triangular elements of matrix $(\mathbf{mm}')_{OD}$ and of matrix $\mathbf{C}$ to be typical elements of vector $\mathbf{q}'$ and of vector $\boldsymbol{\kappa}$, respectively. Thus, $\mathbf{q}'$ is a row vector of the pairwise products of means of all nonredundant elements of $\mathbf{x}_{ij}$ and $\boldsymbol{\kappa}$ is a column vector of lower-triangular elements of $\mathbf{C}$.

Next we show how Equation (B.2) can be re-expressed as Equation (B.3), which is the numerator of manuscript Equation (9). Rearranging terms yields:

$$\text{var}_{ij}(\hat{y}_T^{fv}) = \mathbf{s}'\boldsymbol{\psi} + 2\mathbf{p}'\boldsymbol{\kappa} + \ddot{\boldsymbol{\gamma}}'\boldsymbol{\Phi}\ddot{\boldsymbol{\gamma}} - (\mathbf{a}'\boldsymbol{\psi} + 2\mathbf{q}'\boldsymbol{\kappa})$$

$$= (\mathbf{s}' - \mathbf{a}')\boldsymbol{\psi} + 2(\mathbf{p}' - \mathbf{q}')\boldsymbol{\kappa} + \ddot{\boldsymbol{\gamma}}'\boldsymbol{\Phi}\ddot{\boldsymbol{\gamma}}$$

Because, as mentioned above,

$\mathbf{s}$ = vector of means (across *i* and *j*) of squared elements of $\mathbf{x}_{ij}$

$\mathbf{a}$ = vector of squared means (across *i* and *j*) of $\mathbf{x}_{ij}$

$\mathbf{p}$ = vector of means (across *i* and *j*) for the pairwise products of all nonredundant elements of $\mathbf{x}_{ij}$

$\mathbf{q}$ = vector of pairwise products of means (across *i* and *j*) of all nonredundant elements of $\mathbf{x}_{ij}$

we can substitute

$\mathbf{v}' = (\mathbf{s}' - \mathbf{a}')$ which are variances of $\mathbf{x}_{ij}$ across all observations (across *i* and *j*)

$\mathbf{r}' = (\mathbf{p}' - \mathbf{q}')$ which are covariances of $\mathbf{x}_{ij}$ across all observations (across *i* and *j*)

thus yielding the numerator of manuscript Equation (9):

$$\text{var}_{ij}(\hat{y}_T^{fv}) = \mathbf{v}'\boldsymbol{\psi} + 2\mathbf{r}'\boldsymbol{\kappa} + \ddot{\boldsymbol{\gamma}}'\boldsymbol{\Phi}\ddot{\boldsymbol{\gamma}} .$$

Online Appendix C

Empirical example 1 results: Parameter estimates and standard errors from the best fitting *K*=3 single-level regression mixture model predicting colleges' average professor salary

	k=1 (25%)	*k*=2 (60%)	*k*=3 (15%)
Intercept	347.976 (6.076)	397.956 (4.635)	489.652 (17.193)
admit	85.499 (36.609)	-203.998 (26.957)	-357.115 (65.167)
perPhD	1.154 (0.636)	3.299 (0.297)	3.331 (0.946)
sfratio	3.059 (1.110)	-3.247 (1.125)	0.030 (2.806)

Notes: Metric is in 100's of dollars earned annually. Standard errors and class percentages are in parentheses. admit= applications received divided by applicants accepted. perPhD= percentage of faculty with a Ph.D. sfratio= student-to-faculty ratio. Residual variances were constrained equal across class for parsimony $\hat{\theta}^k = \hat{\theta}^{\bullet} =$ 1559.880 (143.507).

Online Appendix D

Empirical example 2 results: Parameter estimates and standard errors from the best-fitting *K*=3 *H*=2 multilevel regression mixture model predicting teacher job satisfaction

	h=1 (52%)			*h*=2 (48%)		
	k =1 (11%)	*k* =2 (28%)	*k* =3 (13%)	*k* =1 (22%)	*k* =2 (6%)	*k* =3 (20%)
Intercept	3.934	3.296	2.407	3.974	2.554	3.492
	(0.060)	(0.063)	(0.056)	(0.058)	(0.119)	(0.138)
lead	0.132	0.343	0.348	0.128	0.392	0.495
	(0.047)	(0.021)	(0.032)	(0.044)	(0.096)	(0.142)
control	0.078	0.245	0.122	0.070	0.269	0.039
	(0.048)	(0.043)	(0.033)	(0.044)	(0.073)	(0.115)
delinquency	0.212	3.296	2.407	-0.199	-0.336	-0.362
	(0.117)	(0.063)	(0.056)	(0.085)	(0.197)	(0.168)

Notes: Standard errors and class proportions are in parentheses. lead = teacher-reported quality of principal's leadership. control = teacher-reported amount of control over curriculum and policy. delinquency = school-level delinquency. Residual variances were constrained equal across class for parsimony $\hat{\theta}^{kh} = \hat{\theta}^{\cdot\cdot}$ = 0.145 (0.011).

Online Appendix E

***regMixR2* R function Description:**

This function reads in regression mixture parameter estimates and outputs all relevant R^2 measures and decompositions. That is, when Equation (1) or (2) is fit, R^2's in Table 1a and decompositions in Table 3 are outputted. When Equation (7) is fit, R^2's in Table 1b and decompositions in Table 4 and 5 are outputted. Additionally, barcharts for total R^2 (and decompositions) and level-2 R^2 (and decompositions) are produced, as in Figures 4, 7, and 8.

Any number of level-1 and/or level-2 classes is supported (including the special case of K=1, a multilevel mixture with classes only at level-2, or H=1, a single-level mixture). When fitting a multilevel mixture with classes at levels 1 and 2, if the number of level-1 classes differs across h (as described in manuscript Footnote 15), a researcher can constrain all parameters equal across class-combinations within any h (e.g., if there are K=3 level-1 classes, constraining two of these equal within h effectively yields K_h =2). Parameter estimates should still be entered as described below, which would involve entering all constrained parameters as though they were belonging to separate classes.

***regMixR2* R function Input:**

data – Data set with rows denoting observations and columns denoting variables.

covariate.cols – List of numbers corresponding to the columns in the data set that represent the predictors used in the regression model

H – Number of level-2 classes.

K – Number of level-1 classes.

intslopes – Vector of coefficient estimates for all class-combination-specific (or class-specific) intercepts and all class-combination-specific slopes, to be entered in the following order: (1) all intercepts going in order of increasing k (level-1 class) then increasing h (level-2 class) (e.g., k=1,h=1; k=2,h=1; k=1,h=2; k=2,h=2); (2) all slopes for each class-combination (classes increasing as in (1)), e.g., xslope1_k1h1, xslope2_k1h1, xslope1_k2h1, xslope2_k2h1, etc.) If coefficients are constrained equal across certain classes, then the same estimates would be entered for those classes.

resvar – Vector of class-combination-specific (or class-specific) residual variance estimates (entered in the order of classes described in *intslopes* above)

mcwi – Vector of level-1 class multinomial intercept estimates, entered in order of increasing k, with 0 entered for K

mcws – Vector of multinomial slopes of k on h *estimates* (entered in the order of classes described in *intslopes* above, with 0 entered for every $k=K$ and $h=H$)

mcbi – Vector of level-2 class multinomial intercept estimates, entered in order of increasing h, with 0 entered for H

regMixR2 R function Syntax:

```
##R code
regMixR2 <-
  function(data,covariate.cols,H,K,intslopes,resvar,mcwi,mcws,mcbi){

 ##compute phi
   covariates <- cbind(1,data[,covariate.cols])
    phi <- cov(covariates)

    ##compute s
     s <- matrix(NA,1+length(covariate.cols),1)
    for(i in seq(1+length(covariate.cols))) {
      s[i] <- mean(covariates[,i]^2)
    }

    ##compute p
   p <- matrix(NA,1+length(covariate.cols),1+length(covariate.cols))
    for(i in seq(1+length(covariate.cols))){
      for(j in seq(1+length(covariate.cols))){
        p[i,j] <- mean(covariates[,i]*covariates[,j])
      }
    }
    p <- as.vector(p[lower.tri(p)==TRUE])

    ##compute v
    v <- matrix(NA,1+length(covariate.cols),1)
    for(i in seq(1+length(covariate.cols))) {
      v[i] <- var(covariates[,i])
    }

    ##compute r
    r <- matrix(NA,1+length(covariate.cols),1+length(covariate.cols))
    for(i in seq(1+length(covariate.cols))){
      for(j in seq(1+length(covariate.cols))){
        r[i,j] <- cov(covariates[,i],covariates[,j])
      }
    }
    r <- as.vector(r[lower.tri(r)==TRUE])

    ##read in intercepts and slopes
     int_slopes <- aperm(array(data=intslopes,dim=c(K,H,1+length(covariate.cols))),perm=c(2,1,3))

    ##read in residual variances
     residual_var <- matrix(data=resvar,H,K,byrow=TRUE)

    ##read in multinomials

    #L1 class intercepts
     mcw <- matrix(data=mcwi,1,K)
    mcw[1,K] <- 0

    #L2 class on L1 class slopes
    ms <- matrix(data=mcws,H,K,byrow=TRUE)
    ms[H,1:K] <- 0
    ms[1:H,K] <- 0

    #L2 class intercepts
    mcb <- matrix(data=mcbi,H,1)
    mcb[H,1] <- 0

    ##compute probabilities of class membership

    #denominator for p_cbcw for each cbcw
    den_cbcw <- matrix(NA,H,K)
    for (i in 1:H)
    {
      for (j in 1:K)
      {
```

```
      den_cbcw[i,j] <- sum(exp(mcw[1,1:K]+ms[i,1:K]))
     }
   }

   #prob of cw given cb
   prob_cwgivencb <- matrix(NA,H,K)
    for (i in 1:H)
   {
     for (j in 1:K)
     {
      prob_cwgivencb[i,j] <- exp(mcw[1,j]+ms[i,j]) / den_cbcw[i,j]
     }
   }

   #marginal prob of cb
  prob_cb <- matrix(NA,H,1)
   for (i in 1:H)
   {
     prob_cb[i,1] <- exp(mcb[i,1])/ sum(exp(mcb[1:H,1]))
   }

   #class combination prob
   prob_cbcw <- matrix(NA,H,K)
    for (i in 1:H)
   {
     for (j in 1:K)
     {
      prob_cbcw[i,j] <- prob_cb[i,1]*prob_cwgivencb[i,j]
     }
   }

   ##compute marginal L2 class parameters
   margL2params <- array(NA,c(H,1,1+length(covariate.cols)))
    for (i in 1:H)
   {
     for (j in 1:c(1+length(covariate.cols)))
     {
      margL2params[i,1,j] <- sum(prob_cwgivencb[i,1:K]*int_slopes[i,1:K,j])
     }
   }

   ##compute fixed effects
    fixedeffects <- matrix(NA,1+length(covariate.cols),1)
     for (i in 1:c(1+length(covariate.cols)))
   {
     fixedeffects[i,1] <- sum(prob_cb*margL2params[,,i])
   }

   ##compute implied tau matrix
  impliedtau <- matrix(NA,1+length(covariate.cols),1+length(covariate.cols))
  for (i in 1:c(1+length(covariate.cols)))
   {
     for (j in 1:c(1+length(covariate.cols)))
     {
      impliedtau[i,j] <- sum(prob_cb*margL2params[,1,i]*margL2params[,1,j])-
sum(prob_cb*margL2params[,1,i])*sum(prob_cb*margL2params[,1,j])
     }
   }

   ##implied sigma equation
  sigmamatrix <- matrix(NA,1+length(covariate.cols),1+length(covariate.cols))
   for (i in 1:c(1+length(covariate.cols)))
   {
     for (j in 1:c(1+length(covariate.cols)))
     {
      sigmamatrix[i,j] <- sum(prob_cbcw*((int_slopes[,,i]-margL2params[,,i])*
                                  (int_slopes[,,j]-margL2params[,,j])))
     }
   }
   sigmamatrix_psi <- matrix(diag(sigmamatrix),c(1+length(covariate.cols)),1)
```

```
  sigmamatrix_kappa <- as.vector(sigmamatrix[lower.tri(sigmamatrix)==TRUE])
  sigmamatrix_kappa <- matrix(sigmamatrix_kappa,length(sigmamatrix_kappa),1)

  #compute coefficient variance matrix for total
  coefvariance <- matrix(NA,1+length(covariate.cols),1+length(covariate.cols))
   for (i in 1:c(1+length(covariate.cols)))
  {
    for (j in 1:c(1+length(covariate.cols)))
    {
      coefvariance[i,j] <- sum(prob_cbcw[1:H,1:K]*int_slopes[1:H,1:K,i]*int_slopes[1:H,1:K,j])-
sum(prob_cbcw[1:H,1:K]*int_slopes[1:H,1:K,i])*sum(prob_cbcw[1:H,1:K]*int_slopes[1:H,1:K,j])
    }
  }

  #compute coefficient variance matrix for L2R2
  coefvariance_H <- list()
   int_slopes <- aperm(array(data=intslopes,dim=c(K,H,1+length(covariate.cols))),perm=c(2,1,3))
  coefvariance_H <- aperm(array(NA,dim=c(1+length(covariate.cols),H,1+length(covariate.cols))),perm=c(2,1,3))
   for(q in 1:H){
    for (i in 1:c(1+length(covariate.cols)))
    {
      for (j in 1:c(1+length(covariate.cols)))
      {
        coefvariance_H[q,i,j] <- sum(prob_cwgivencb[q,1:K]*int_slopes[q,1:K,i]*int_slopes[q,1:K,j])-
sum(prob_cwgivencb[q,1:K]*int_slopes[q,1:K,i])*sum(prob_cwgivencb[q,1:K]*int_slopes[q,1:K,j])
      }
    }
  }
  margresvar <- sum(prob_cbcw*residual_var)
  psi <- matrix(diag(coefvariance),c(1+length(covariate.cols)),1)
  kappa <- as.vector(coefvariance[lower.tri(coefvariance)==TRUE])
  kappa <- matrix(kappa,length(kappa),1)
  varypred <- t(s)%*%psi+2*t(p)%*%kappa+t(fixedeffects)%*%phi%*%fixedeffects
  totalR2_con <- 1 - (margresvar/(varypred+margresvar))
  totalR2_con_cb <- 1 - ((margresvar+t(s)%*%sigmamatrix_psi+2*t(p)%*%sigmamatrix_kappa)/(varypred+margresvar))
  totalR2_marg <- 1 - ((t(s)%*%psi+2*t(p)%*%kappa+margresvar)/(varypred+margresvar))
  totalvar <- margresvar + varypred
  totalR2_con_H <- matrix(NA,H,1)
  totalR2_con_H_RInull <- matrix(NA,H,1)
  totalR2_marg_H <- matrix(NA,H,1)
  contribution_fixed_H <- matrix(NA,H,1)
  contribution_cw_sep_H <- matrix(NA,H,1)
  contribution_cw_slope_H <- matrix(NA,H,1)

  ##random intercept null
   totalR2_con_RInull <- (t(v)%*%psi+2*t(r)%*%kappa+t(fixedeffects)%*%phi%*%fixedeffects)/(varypred+margresvar)
  totalR2_con_cb_RInull <- (t(v)%*%(psi-sigmamatrix_psi)+2*t(r)%*%(kappa-
sigmamatrix_kappa)+t(fixedeffects)%*%phi%*%fixedeffects)/(varypred+margresvar)

  ##contributions
   contribution_fixed <- totalR2_marg
  contribution_cb_sep <- totalR2_con_cb - totalR2_con_cb_RInull
  contribution_cb_slope <- totalR2_con_cb_RInull - totalR2_marg
  contribution_cw_sep <- totalR2_con - totalR2_con_cb - totalR2_con_RInull + totalR2_con_cb_RInull
  contribution_cw_slope <- totalR2_con_RInull - totalR2_con_cb_RInull
    if(contribution_fixed<0) contribution_fixed <- 0
  if(contribution_cb_sep<0) contribution_cb_sep <- 0
  if(contribution_cb_slope<0) contribution_cb_slope <- 0
  if(contribution_cw_sep<0) contribution_cw_sep <- 0
  if(contribution_cw_slope<0) contribution_cw_slope <- 0

  ##L2R2
      for(i in 1:H){
      margresvar_H <- sum(sum(prob_cwgivencb[i,1:K]*residual_var[i,1:K]))
      psi_H <- matrix(diag(coefvariance_H[i,,]),c(1+length(covariate.cols)),1)
      kappa_H <- as.vector(coefvariance_H[i,,][lower.tri(coefvariance_H[i,,])==TRUE])
    kappa_H <- matrix(kappa_H,length(kappa_H),1)
    varypred_H <- t(s)%*%psi_H+2*t(p)%*%kappa_H+t(margL2params[i,,])%*%phi%*%margL2params[i,,]
    totalR2_con_H[i,] <- 1 - (margresvar_H/(varypred_H+margresvar_H))
    totalR2_marg_H[i,] <- 1 - ((t(s)%*%psi_H+2*t(p)%*%kappa_H+margresvar_H)/(varypred_H+margresvar_H))
```

```
      totalR2_con_H_RInull[i,] <-
(t(v)%*%psi_H+2*t(r)%*%kappa_H+t(margL2params[i,,])%*%phi%*%margL2params[i,,])/(varypred_H+margresvar_H)
      contribution_fixed_H[i,] <- totalR2_marg_H[i,]
      contribution_cw_sep_H[i,] <- totalR2_con_H[i,] - totalR2_con_H_RInull[i,]
      contribution_cw_slope_H[i,] <- totalR2_con_H_RInull[i,] - totalR2_marg_H[i,]
      if(contribution_fixed_H[i,]<0) contribution_fixed_H[i] <- 0
      if(contribution_cw_sep_H[i,]<0) contribution_cw_sep_H[i,] <- 0
      if(contribution_cw_slope_H[i,]<0) contribution_cw_slope_H[i,] <- 0
    }

    ##LI1L2R2

    totalR2_KH <- matrix(NA,H,K)
    for(i in 1:H){
      for(j in 1:K){
        varypred_KH <- t(int_slopes[i,j,])%*%phi%*%int_slopes[i,j,]
        totalR2_KH[i,j] <- 1 - (residual_var[i,j]/(varypred_KH+residual_var[i,j]))

      }
    }

    ##output tables

    totalR2table <- matrix(c(totalR2_con,totalR2_con_RInull,totalR2_marg),1,3)
    colnames(totalR2table) <- c("R2_fvm","  R2_fv","  R2_f")
    rownames(totalR2table) <- c("")

    L2R2table <- matrix(c(totalR2_con_H,totalR2_con_H_RInull,totalR2_marg_H),H,3,byrow=FALSE)
    colnames(L2R2table) <- c("R2_fvm","  R2_fv","  R2_f")
    rownames(L2R2table) <- c(paste("h =",1:H))

    L1L2R2table <- matrix(totalR2_KH,H,K)
    colnames(L1L2R2table) <- c(paste("k =",1:K))
    rownames(L1L2R2table) <- c(paste("h =",1:H))
    if (H==1) rownames(L1L2R2table) <- ""
    if (K==1) colnames(L1L2R2table) <- ""

    contributionstable_FInull_fixed <- matrix(c("contribution via marginal slopes",contribution_fixed),2,1)
    rownames(contributionstable_FInull_fixed) <- c("","")
    colnames(contributionstable_FInull_fixed) <- ""
    contributionstable_FInull_fixed <- noquote(contributionstable_FInull_fixed)

    contributionstable_FInull_cbsep <- matrix(c("contribution via variation in L2 class means",contribution_cb_sep),2,1)
    rownames(contributionstable_FInull_cbsep) <- c("","")
    colnames(contributionstable_FInull_cbsep) <- ""
    contributionstable_FInull_cbsep <- noquote(contributionstable_FInull_cbsep)

    contributionstable_FInull_cbslope <- matrix(c("contribution via variation in L2 class slopes",contribution_cb_slope),2,1)
    rownames(contributionstable_FInull_cbslope) <- c("","")
    colnames(contributionstable_FInull_cbslope) <- ""
    contributionstable_FInull_cbslope <- noquote(contributionstable_FInull_cbslope)

    if (H==1){
      contributionstable_FInull_cwsep <- matrix(c("contribution via variation in L1 class means",contribution_cw_sep),2,1)
    } else {
      contributionstable_FInull_cwsep <- matrix(c("contribution via variation in L1 class means within L2 class",contribution_cw_sep),2,1)
    }
    rownames(contributionstable_FInull_cwsep) <- c("","")
    colnames(contributionstable_FInull_cwsep) <- ""
    contributionstable_FInull_cwsep <- noquote(contributionstable_FInull_cwsep)

    if (H==1){
      contributionstable_FInull_cwslope <- matrix(c("contribution via variation in L1 class slopes",contribution_cw_slope),2,1)
    } else {
      contributionstable_FInull_cwslope <- matrix(c("contribution via variation in L1 class slopes within L2 class",contribution_cw_slope),2,1)
    }
    rownames(contributionstable_FInull_cwslope) <- c("","")
    colnames(contributionstable_FInull_cwslope) <- ""
    contributionstable_FInull_cwslope <- noquote(contributionstable_FInull_cwslope)
```

```
    contributionstable_RInull_fixed <- matrix(c("contribution via marginal slopes",contribution_fixed),2,1)
    rownames(contributionstable_RInull_fixed) <- c("","")
    colnames(contributionstable_RInull_fixed) <- ""
    contributionstable_RInull_fixed <- noquote(contributionstable_RInull_fixed)

    contributionstable_RInull_cbslope <- matrix(c("contribution via variation in L2 class slopes",contribution_cb_slope),2,1)
    rownames(contributionstable_RInull_cbslope) <- c("","")
    colnames(contributionstable_RInull_cbslope) <- ""
    contributionstable_RInull_cbslope <- noquote(contributionstable_RInull_cbslope)

    if (H==1){
      contributionstable_RInull_cwslope <- matrix(c("contribution via variation in L1 class slopes",contribution_cw_slope),2,1)
    } else {
      contributionstable_RInull_cwslope <- matrix(c("contribution via variation in L1 class slopes within L2 class",contribution_cw_slope),2,1)
    }
    rownames(contributionstable_RInull_cwslope) <- c("","")
    colnames(contributionstable_RInull_cwslope) <- ""
    contributionstable_RInull_cwslope <- noquote(contributionstable_RInull_cwslope)

    contributionstable_FInull <- rbind(contributionstable_FInull_fixed,contributionstable_FInull_cbsep,contributionstable_FInull_cbslope,
                                contributionstable_FInull_cwsep,contributionstable_FInull_cwslope)
    contributionstable_RInull <- rbind(contributionstable_RInull_fixed,contributionstable_RInull_cbslope,contributionstable_RInull_cwslope)

    if (H==1){
      contributionstable_FInull <- rbind(contributionstable_FInull_fixed,contributionstable_FInull_cwsep,contributionstable_FInull_cwslope)
      contributionstable_RInull <- rbind(contributionstable_RInull_fixed,contributionstable_RInull_cwslope)
    }

    if (K==1){
      contributionstable_FInull <- rbind(contributionstable_FInull_fixed,contributionstable_FInull_cbsep,contributionstable_FInull_cbslope)
      contributionstable_RInull <- rbind(contributionstable_RInull_fixed,contributionstable_RInull_cbslope)
    }

    contributionstable_FInull <- noquote(contributionstable_FInull)
    contributionstable_RInull <- noquote(contributionstable_RInull)

    contributionstable_H_FInull_fixed <- matrix(c("",contribution_fixed_H),c(H+1),1)
    rownames(contributionstable_H_FInull_fixed) <- c("contribution via marginal slopes in L2 class h",paste("h =",1:H))
    colnames(contributionstable_H_FInull_fixed) <- ""
    contributionstable_H_FInull_fixed <- noquote(contributionstable_H_FInull_fixed)

    contributionstable_H_FInull_cwsep <- matrix(c("",contribution_cw_sep_H),c(H+1),1)
    rownames(contributionstable_H_FInull_cwsep) <- c("contribution via variation in L1 class means within L2 class h",paste("h =",1:H))
    colnames(contributionstable_H_FInull_cwsep) <- ""
    contributionstable_H_FInull_cwsep <- noquote(contributionstable_H_FInull_cwsep)

    contributionstable_H_FInull_cwslope <- matrix(c("",contribution_cw_slope_H),c(H+1),1)
    rownames(contributionstable_H_FInull_cwslope) <- c("contribution via variation in L1 class slopes within L2 class h",paste("h =",1:H))
    colnames(contributionstable_H_FInull_cwslope) <- ""
    contributionstable_H_FInull_cwslope <- noquote(contributionstable_H_FInull_cwslope)

    contributionstable_H_RInull_fixed <- matrix(c("",contribution_fixed_H),c(H+1),1)
    rownames(contributionstable_H_RInull_fixed) <- c("contribution via marginal slopes in L2 class h",paste("h =",1:H))
    colnames(contributionstable_H_RInull_fixed) <- ""
    contributionstable_H_RInull_fixed <- noquote(contributionstable_H_RInull_fixed)

    contributionstable_H_RInull_cwslope <- matrix(c("",contribution_cw_slope_H),c(H+1),1)
    rownames(contributionstable_H_RInull_cwslope) <- c("contribution via variation in L1 class slopes within L2 class h",paste("h =",1:H))
    colnames(contributionstable_H_RInull_cwslope) <- ""
    contributionstable_H_RInull_cwslope <- noquote(contributionstable_H_RInull_cwslope)

    contributionstable_H_FI <-
rbind(contributionstable_H_FInull_fixed,contributionstable_H_FInull_cwsep,contributionstable_H_FInull_cwslope)
    contributionstable_H_FI <- noquote(contributionstable_H_FI)

    contributionstable_H_RI <- rbind(contributionstable_H_RInull_fixed,contributionstable_H_FInull_cwslope)
    contributionstable_H_RI <- noquote(contributionstable_H_RI)

    ##total R2 contributions barplots
```

```
  if (H>1 && K>1){
    contributions_stacked <-
matrix(c(contribution_fixed,contribution_cb_slope,contribution_cw_slope,contribution_cb_sep,contribution_cw_sep,
                          contribution_fixed,contribution_cb_slope,contribution_cw_slope,0,0,
                          contribution_fixed,0,0,0,0),5,3)
    colnames(contributions_stacked) <- c("R2_fvm_T","R2_fv_T","R2_f_T")
    rownames(contributions_stacked) <- c("contribution via marginal slopes",
                           "contribution via variation in L2 class slopes",
                           "contribution via variation in L1 class slopes within L2 class",
                           "contribution via variation in L2 class means",
                           "contribution via variation in L1 class means within L2 class")
  }

  if (H==1){
    contributions_stacked <- matrix(c(contribution_fixed,contribution_cw_slope,contribution_cw_sep,
                          contribution_fixed,contribution_cw_slope,0,
                          contribution_fixed,0,0),3,3)
    colnames(contributions_stacked) <- c("R2_fvm","R2_fv","R2_f")
    rownames(contributions_stacked) <- c("contribution via marginal slopes",
                           "contribution via variation in class slopes",
                           "contribution via variation in class means")
  }

  if (K==1){
    contributions_stacked <- matrix(c(contribution_fixed,contribution_cb_slope,contribution_cb_sep,
                          contribution_fixed,contribution_cb_slope,0,
                          contribution_fixed,0,0),3,3)
    colnames(contributions_stacked) <- c("R2_fvm","R2_fv","R2_f")
    rownames(contributions_stacked) <- c("contribution via marginal slopes",

                           "contribution via variation in class slopes",
                           "contribution via variation in class means")
  }
  # contributions_stacked[which(contributions_stacked<.000001)] <- 0

  if (H>1 && K>1){

    old.par <- par(mar = c(0, 0, 0, 0))
    par(mar=c(7,5,3,3))

    barplot(contributions_stacked, main="Contributions to Total R2", horiz=FALSE,
        ylim=c(0,1),col=c("navyblue","darkblue","steelblue","steelblue","lightcyan2"),ylab="Proportion of variance explained",
        density=c(NA,40,NA,15,NA),angle=c(0,45,0,135,0),xlim=c(0,1),width=c(.3,.3))
    legend(-.03,-.15,legend=rownames(contributions_stacked),fill=c("navyblue","darkblue","steelblue","steelblue","lightcyan2"),
       cex=.7, pt.cex = 1,xpd=TRUE,density=c(NA,40,NA,15,NA),angle=c(0,45,0,135,0))
    #abline(h=0)
    #abline(h=1)
    par(old.par)
  }

  if (H==1 | K==1){

    old.par <- par(mar = c(0, 0, 0, 0))
    par(mar=c(7,5,3,3))

    barplot(contributions_stacked, main="Contributions to Total R2", horiz=FALSE,
        ylim=c(0,1),col=c("navyblue","darkblue","steelblue"),ylab="Proportion of variance explained",
        density=c(NA,40,NA),angle=c(0,45,0),xlim=c(0,1),width=c(.3,.3))
    legend(.1,-.17,legend=rownames(contributions_stacked),
       fill=c("navyblue","darkblue","steelblue"),density=c(NA,40,NA),angle=c(0,45,0),
       cex=.7, pt.cex = 1,xpd=TRUE)
    #,"navyblue","steelblue"

    #abline(h=0)
    #abline(h=1)
    par(old.par)

  }

  ##level-2 contribution barplots
```

```
  contributions_stacked_H <- array(NA,c(3,3,H))

  if (H>1 && K>1){
   for(i in seq(H)){
    contributions_stacked_H[,,i] <-matrix(c(contribution_fixed_H[i],contribution_cw_slope_H[i],contribution_cw_sep_H[i],
                                            contribution_fixed_H[i],contribution_cw_slope_H[i],0,
                                            contribution_fixed_H[i],0,0),3,3)
    colnames(contributions_stacked_H[,,i]) <- c("R2_fvm","R2_fv","R2_f")
    rownames(contributions_stacked_H[,,i]) <- c("contribution via marginal slopes",
                                              "contribution via variation in L1 class slopes",
                                              "contribution via variation in L1 class means")
   }
  }

  if (H>1 && K>1){
   for(i in seq(H)){

   old.par <- par(mar = c(0, 0, 0, 0))
   par(mar=c(7,5,3,3))

   x <- contributions_stacked_H[,,i]
   colnames(x) <- c("R2_fvm","R2_fv","R2_f")

   barplot(x, main=paste0("Contributions to Level-2 Class R2, h = ",i), horiz=FALSE,
        ylim=c(0,1),col=c("navyblue","steelblue","lightcyan2"),ylab="Proportion of variance explained",
       xlim=c(0,1),width=c(.3,.3))
  #  density=c(NA,40,NA,15,NA),angle=c(0,45,0,135,0),
    legend(-.03,-.15,legend=c("contribution via marginal slopes","contribution via variation in L1 class slopes",
                    "contribution via variation in L1 class means"),fill=c("navyblue","steelblue","lightcyan2"),
       cex=.7, pt.cex = 1,xpd=TRUE)
   #abline(h=0)
   #abline(h=1)
   par(old.par)
   }
  }

  Output <-
list(totalR2table,L2R2table,L1L2R2table,contributionstable_FInull,contributionstable_RInull,contributionstable_H_FI,contributionstable_H_RI)
  names(Output) <- c("Total R2", "Level-2 R2","Class-combination R2","Relative Contributions to Total R2_fvm","Relative Contributions to
Total R2_fv","Relative Contributions to Level-2 R2_fvm","Relative Contributions to Level-2 R2_fv")

  if(H==1) Output <- Output[c(1,3,4,5)]
  if(K==1) Output <- Output[c(1,3,4,5)]

  return(Output)

 }
```

regMixR2 example input:

```
#NOTE:
#the estimates in the input represent hypothetical results for a H=2, K=2 class model with 2 covariates
#in practice a user would have previously obtained these input estimates by fitting their model in a mixture modeling software package
#additionally, the input consists of simulated covariate data, whereas in practice a user would read-in their actual covariate data

data <- cbind(rnorm(100,0,1),rnorm(100,0,2))

regMixR2(data=data,H=2, K=2,
    covariate.cols=c(2,1),
    intslopes=c(2,3.884,3.368,4.078,
           1,0.157,0.432,0.107,
           -0.158,-0.033,-0.248,-0.039),
    resvar=c(0.427,0.108,0.311,0.077),
    mcwi=c(0.758,0),
    mcws=c(0.478,0,0,0),
    mcbi=c(0.171,0))
```

Online Appendix F

The below example code can be used for fitting a multilevel regression mixture with two covariates (one level-1 and one level-2) and classes at level-1 and level-2 (here, H=2 and K=2) in M*plus*. See Equation (7) for details.

```
TITLE:    example
DATA:     FILE =  C:\filepath\example.dat;
VARIABLE:
NAMES ARE y x w cluster_id subject_id;
          USEVARIABLES = y x w;
          CLASSES = cb (2) cw (2);
          !cb denotes level-2 class, cw denotes level-1 class
          WITHIN =x cw;
          BETWEEN = w cb;
          CLUSTER = cluster_id;
ANALYSIS:
          TYPE = TWOLEVEL MIXTURE;
          ESTIMATOR=ML;
MODEL:
%WITHIN%
%OVERALL%
  y on x;
  [cw#1]; !multinomial intercept for level-1 class k
%cb#1.cw#1%
   y; !class-combination residual variance
   y on x; !class-combination slope of x
%cb#1.cw#2%
   y; y on x;
%cb#2.cw#1%
   y; y on x;
%cb#2.cw#2%
   y; y on x ;
%BETWEEN%
%OVERALL%
   cw#1 on cb#1; !multinomial slope of k on h
   y on w; [y];
   [cb#1]; !multinomial intercept for level-2 class h
%cb#1.cw#1%
   [y ]; !class-combination intercept
   y on w; !class-combination slope of w
   y@0;
%cb#1.cw#2%
   [y ]; y on w; y@0;
%cb#2.cw#1%
   [y ]; y on w; y@0;
%cb#2.cw#2%
   [y ]; y on w; y@0;
```