This document provides supplemental materials for our article entitled "An Introduction to Multiple Imputation of Multilevel Missing Data Using R". Supplement A contains syntax files for the imputation and analysis of the two empirical examples using the statistical software M*plus*. Supplement B contains the computer code for the results reported in this article using the statistical software R. Supplement C provides a discussion, syntax files, and additional results from the analysis in Example 1 using manifest and latent group means.

Supplement A: M*plus* Syntaxes for the Empirical Examples

In this section, we provide syntax files for reproducing the steps needed to perform the imputation and analysis with the statistical software M*plus* (Muthén & Muthén, 2012). For each example, we provide separate syntax files for the imputation and analysis phases.

Example 1

Given below is the syntax for the imputation phase in Example 1, using the unrestricted variance-covariance model (H1; see Asparouhov & Muthén, 2010).

```
TITLE:
Example 1: two-level regression model (imputation phase);

DATA:
file = IGLU2001_mplusData.dat;

VARIABLE:
names = ID MathAchiev ReadAchiev CognAbility SES MathDis ReadDis SchClimate;
cluster = ID;
missing = all (-99);

ANALYSIS:
type = basic twolevel;
fbiterations = 50000;
chains = 1;
bseed = 1234;

DATA IMPUTATION:
impute = MathAchiev ReadAchiev CognAbility SES MathDis ReadDis SchClimate;
ndatasets = 100;
save = mplusExample1_*.dat;
```

```
thin = 5000;

! NOTE:
! absence of 'model' command triggers joint model (H1) imputation
```

Given below is the syntax for the analysis model in Example 1 (using manifest group means).

```
TITLE:
Example 1: two-level regression model (analysis phase);

DATA:
file = mplusExample1_list.dat;
type = imputation;

VARIABLE:
names = MathAchiev ReadAchiev CognAbility SES MathDis ReadDis SchClimate ID;
usevariables = MathAchiev SES MathDis classSES classMathDis;
within = MathDis SES;
between = classMathDis classSES;
cluster = ID;

DEFINE:
! calculate group means
classMathDis = CLUSTER_MEAN (MathDis);
classSES = CLUSTER_MEAN (SES);
! group mean centering of student-level variables
center MathDis SES (GROUPMEAN);

ANALYSIS:
type = twolevel;

MODEL:

%within%
MathAchiev ON MathDis SES;

%between%
MathAchiev ON classMathDis classSES;
```

Example 2

Given below is the syntax for the imputation phase in Example 2, using a model-based imputation procedure (H0; see Asparouhov & Muthén, 2010).

```
TITLE:
Example 2: random slope model using IGLU data (imputation phase);

DATA:
file = IGLU2001_mplusData.dat;

VARIABLE:
names = ID MathAchiev ReadAchiev CognAbility SES MathDis ReadDis SchClimate;
usevariables = CognAbility MathAchiev ReadAchiev SES MathDis ReadDis SchClimate
classCognAbility;
within = CognAbility;
between = classCognAbility;
cluster = ID;
missing = all (-99);

DEFINE:
! calculate group means
classCognAbility = CLUSTER_MEAN (CognAbility);
! group mean centering of student-level variables
center CognAbility (GROUPMEAN);

ANALYSIS:
type = twolevel random;
estimator = bayes;
fbiterations = 100000;
chains = 1;
bseed = 1234;

DATA IMPUTATION:
impute = MathAchiev ReadAchiev CognAbility SES MathDis ReadDis SchClimate;
ndatasets = 100;
save = mplusExample2_*.dat;
thin = 20000;

MODEL:

%within%
! random slopes on cognitive ability
b1 | MathAchiev ON CognAbility;
b2 | SES ON CognAbility;
b3 | ReadAchiev ON CognAbility;
b4 | MathDis ON CognAbility;
b5 | ReadDis ON CognAbility;
b6 | SchClimate ON CognAbility;

%between%
! fixed slope on class mean of cognitive ability
MathAchiev ON classCognAbility;
SES ON classCognAbility;
ReadAchiev ON classCognAbility;
```

```
MathDis ON classCognAbility;
ReadDis ON classCognAbility;
SchClimate ON classCognAbility;

! covariances among random intercepts
MathAchiev - SchClimate with MathAchiev - SchClimate;
! covariances among random slopes
b1 - b6 with b1 - b6;
! covariances of random intercepts with random slopes
b1 - b6 with MathAchiev - SchClimate;
```

Given below is the syntax for the analysis model in Example 2.

```
TITLE:
Example 2: random slope model using IGLU data (analysis phase);

DATA:
file = mplusExample2_list.dat;
type = imputation;

VARIABLE:
names = MathAchiev ReadAchiev SES MathDis ReadDis SchClimate CognAbility
classCognAbility ID;
usevariables = MathAchiev SES CognAbility classCognAbility classSES;
within = SES CognAbility;
between = classCognAbility classSES;
cluster = ID;
missing = all (-99);

DEFINE:
! calculate group means (SES only)
! classCognAbility is already present
classSES = CLUSTER_MEAN (SES);
! group mean centering of student-level variables
center CognAbility SES (GROUPMEAN);

ANALYSIS:
type = twolevel random;

MODEL:

%within%
b1 | MathAchiev ON CognAbility;
MathAchiev ON SES;

%between%
MathAchiev ON classCognAbility classSES;
MathAchiev with b1;
```

Supplement B: Computer Code for the Empirical Examples

In this section, we provide the computer code that was used for the imputation and analysis of the two examples using the statistical software R (R Core Team, 2015). For each block of code, we provide the statistical output as given by R. As in the main article, we provide the results for multilevel MI, single-level MI, and listwise deletion (LD). Note that the empirical data set cannot be included in this document due to privacy regulations. As an alternative, the `mitml` package contains an artificial data set (`studentratings`), which was simulated such that it would mimic the data used in the main article. Readers may repeat the computations below with these artificial data. However, the results will deviate from the original results presented here and in the main article.

Multilevel MI

For multilevel MI, the pan package (Schafer & Zhao, 2014) is used in conjunction with `mitml` (Grund, Robitzsch, & Lüdtke, 2016).

```
# EXAMPLE 1

# specification of imputation model
fml <- MathAchiev + MathDis + SES + ReadAchiev + CognAbility + ReadDis +
       SchClimate ~ 1 + (1|ID)

# running the imputation
imp <- panImpute(dat, formula=fml, n.burn=50000, n.iter=5000, m=100, seed=1234)

summary(imp)
plot(imp)

# extracting the imputed data sets
impList <- mitmlComplete(imp, print="all")

# group mean centering
impList <- within( impList, { MathDis.CLS <- clusterMeans(MathDis,ID)
                              SES.CLS <- clusterMeans(SES,ID) } )
impList <- within( impList, { MathDis.STU <- MathDis - MathDis.CLS
                              SES.STU <- SES - SES.CLS } )

# null models for estimating ICCs
```

```
fit.MA <- with( impList, lmer(MathAchiev ~ 1 + (1|ID)) )
testEstimates(fit.MA, var.comp=TRUE)

fit.SES <- with( impList, lmer(SES ~ 1 + (1|ID)) )
testEstimates(fit.SES, var.comp=TRUE)

fit.DPM <- with( impList, lmer(MathDis ~ 1 + (1|ID)) )
testEstimates(fit.DPM, var.comp=TRUE)

# fitting the model of interest
fit <- with( impList, lmer(MathAchiev ~ 1 + SES.STU + SES.CLS + MathDis.STU +
                      MathDis.CLS + (1|ID)) )

# final parameter estimates and inferences
testEstimates(fit, var.comp=TRUE)

# testing for contextual effects
testConstraints(fit, "SES.CLS - SES.STU")
testConstraints(fit, "MathDis.CLS - MathDis.STU")

# EXAMPLE 2

# specification of imputation model
fml <- MathAchiev + SES + ReadAchiev + MathDis + ReadDis + SchClimate ~ 1 +
       CognAbility.STU + CognAbility.CLS + (1+CognAbility.STU|ID)

# running the imputation
imp <- panImpute(dat, formula=fml, n.burn=100000, n.iter=20000, m=100, seed=1234)

summary(imp)
plot(imp)

# extracting the imputed data sets
impList <- mitmlComplete(imp, print="all")

# group mean centering
impList <- within( impList, { CognAbility.CLS <- clusterMeans(CognAbility,ID)
                              SES.CLS <- clusterMeans(SES,ID) } )
impList <- within( impList, { CognAbility.STU <- CognAbility - CognAbility.CLS
                              SES.STU <- SES - SES.CLS } )

# fitting the model of interest
fit <- with( impList, { lmer(MathAchiev ~ 1 + CognAbility.STU + CognAbility.CLS +
                        SES.STU + SES.CLS + (1+CognAbility.STU|ID), REML=FALSE) } )

# final parameter estimates and inferences
testEstimates(fit, var.comp=TRUE)

# testing for slope variance
```

```
fit.null <- with( impList, lmer(MathAchiev~ 1 + CognAbility.STU + CognAbility.CLS +
                            SES.STU + SES.CLS + (1|ID), REML=FALSE) )

testModels(fit, fit.null, method="D3")
```

The console output for the multilevel MI procedure is given below. Where necessary, the output is truncated to promote readability.

```
# EXAMPLE 1

> summary(imp)

Call:

panImpute(data = dat, formula = fml, n.burn = 50000, n.iter = 5000,
    m = 100, seed = 1234)

Cluster variable:         ID
Target variables:         MathAchiev MathDis SES ReadAchiev CognAbility ...
Fixed effect predictors:  (Intercept)
Random effect predictors: (Intercept)

Performed 50000 burn-in iterations, and generated 100 imputed data sets,
each 5000 iterations apart.

Potential scale reduction (Rhat, imputation phase):

         Min   25%  Mean Median   75%   Max
Beta:  1.000 1.000 1.000  1.000 1.000 1.000
Psi:   1.000 1.000 1.001  1.000 1.001 1.011
Sigma: 1.000 1.000 1.000  1.000 1.000 1.001

Largest potential scale reduction:
Beta: [1,6], Psi: [1,1], Sigma: [1,1]

Missing data per variable:
    ID MathAchiev MathDis SES  ReadAchiev CognAbility ReadDis SchClimate
MD% 0  19.4       61.4    35.0 0          0           21.5    21.7

> testEstimates(fit.MA, var.comp=TRUE)

Call:

testEstimates(model = fit.MA, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.
```

```
             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)   500.207     1.974   253.395 25195.609     0.000     0.067     0.063

                        Estimate
Intercept~~Intercept|ID 1219.400
Residual~~Residual      8887.912
ICC|ID                     0.121

Unadjusted hypothesis test as appropriate in larger samples.

> testEstimates(fit.SES, var.comp=TRUE)

Call:

testEstimates(model = fit.SES, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)    44.089     0.352   125.388  4158.344     0.000     0.182     0.155

                        Estimate
Intercept~~Intercept|ID   34.995
Residual~~Residual       252.884
ICC|ID                     0.122

Unadjusted hypothesis test as appropriate in larger samples.

> testEstimates(fit.DPM, var.comp=TRUE)

Call:

testEstimates(model = fit.DPM, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)     2.361     0.020   118.550  1575.200     0.000     0.335     0.252

                        Estimate
Intercept~~Intercept|ID    0.111
Residual~~Residual         0.510
ICC|ID                     0.179

Unadjusted hypothesis test as appropriate in larger samples.

> testEstimates(fit, var.comp=TRUE)

Call:
```

```
testEstimates(model = fit, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)   502.498    19.254    26.098  1210.447     0.000     0.401     0.287
SES.STU         1.065     0.084    12.614   526.055     0.000     0.766     0.436
SES.CLS         2.150     0.267     8.065  1429.302     0.000     0.357     0.264
MathDis.STU   -20.736     2.032   -10.203   372.305     0.000     1.065     0.518
MathDis.CLS   -41.131     5.035    -8.169  1358.967     0.000     0.370     0.271

                        Estimate
Intercept~~Intercept|ID  655.957
Residual~~Residual      8318.936
ICC|ID                     0.073

Unadjusted hypothesis test as appropriate in larger samples.

> testConstraints(fit, "SES.CLS - SES.STU")

Call:

testConstraints(model = fit, constraints = "SES.CLS - SES.STU")

Hypothesis test calculated from 100 imputed data sets. The following
constraints were specified:

 SES.CLS - SES.STU

Combination method: D1

    F.value      df1      df2  p.value      RIV
     15.297        1 1292.993    0.000    0.365

Unadjusted hypothesis test as appropriate in larger samples.

> testConstraints(fit, "MathDis.CLS - MathDis.STU")

Call:

testConstraints(model = fit, constraints = "MathDis.CLS - MathDis.STU")

Hypothesis test calculated from 100 imputed data sets. The following
constraints were specified:

 MathDis.CLS - MathDis.STU

Combination method: D1
```

```
    F.value      df1      df2  p.value      RIV
     15.313        1 1538.459    0.000    0.325

Unadjusted hypothesis test as appropriate in larger samples.

# Example 2

> summary(imp)

Call:

panImpute(data = dat, formula = fml, n.burn = 1e+05, n.iter = 20000,
    m = 100, seed = 1234)

Cluster variable:         ID
Target variables:         MathAchiev SES ReadAchiev MathDis ReadDis SchClimate
Fixed effect predictors:  (Intercept) CognAbility.STU CognAbility.CLS
Random effect predictors: (Intercept) CognAbility.STU

Performed 100000 burn-in iterations, and generated 100 imputed data sets,
each 20000 iterations apart.

Potential scale reduction (Rhat, imputation phase):

         Min   25%  Mean Median   75%   Max
Beta:  1.000 1.000 1.000  1.000 1.000 1.000
Psi:   1.000 1.000 1.000  1.000 1.000 1.001
Sigma: 1.000 1.000 1.000  1.000 1.000 1.000

Largest potential scale reduction:
Beta: [2,6], Psi: [6,2], Sigma: [2,1]

Missing data per variable:
    ID MathAchiev SES  ReadAchiev MathDis ReadDis SchClimate CognAbility ...
MD% 0  19.4       35.0 0          61.4    21.5    21.7       0

> testEstimates(fit, var.comp=TRUE)

Call:

testEstimates(model = fit, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)    84.573    21.364     3.959  8488.701     0.000     0.121     0.108
CognA...STU     6.114     0.150    40.812  2897.854     0.000     0.227     0.185
CognA...CLS     7.608     0.521    14.592  3720.298     0.000     0.195     0.164
```

```
SES.STU        0.605     0.077     7.813   467.262     0.000     0.853     0.463
SES.CLS        1.081     0.261     4.134  1196.456     0.000     0.404     0.289

                                    Estimate
Intercept~~Intercept|ID              485.050
Intercept~~CognAbility.STU|ID          0.333
CognAbility.STU~~CognAbility.STU|ID    1.528
Residual~~Residual                  6452.506
ICC|ID                                 0.070

Unadjusted hypothesis test as appropriate in larger samples.

> testModels(fit, fit.null, method="D3")

Call:

testModels(model = fit, null.model = fit.null, method = "D3")

Model comparison calculated from 100 imputed data sets.
Combination method: D3

     F.value       df1       df2   p.value       RIV
       5.119         2 10386.237     0.006     0.157
```

Single-Level MI

Single-level MI was performed using the `mice` package (van Buuren & Groothuis-Oudshoorn, 2011). For simplicity, we do not present the calculation of the ICCs in Example 1. These were calculated from separate null models, one for each variable of interest (see above).

```
# EXAMPLE 1

# running single-level imputation
imp <- mice(dat[,-1], m=100, method="norm", maxit=100, seed=1234)

# extracting the imputed data sets
impList <- lapply(1:100, function(i){data.frame(ID=dat$ID, complete(imp,i))})
class(impList) <- c("mitml.list","list")

# group mean centering
impList <- within( impList, { MathDis.CLS <- clusterMeans(MathDis,ID)
                              SES.CLS <- clusterMeans(SES,ID) } )
impList <- within( impList, { MathDis.STU <- MathDis - MathDis.CLS
                              SES.STU <- SES - SES.CLS } )
```

```
# null models for estimating ICCs
fit.MA <- with( impList, lmer(MathAchiev ~ 1 + (1|ID)) )
testEstimates(fit.MA, var.comp=TRUE)

fit.SES <- with( impList, lmer(SES ~ 1 + (1|ID)) )
testEstimates(fit.SES, var.comp=TRUE)

fit.DPM <- with( impList, lmer(MathDis ~ 1 + (1|ID)) )
testEstimates(fit.DPM, var.comp=TRUE)

# fitting the model of interest
fit <- with( impList, lmer(MathAchiev ~ 1 + SES.STU + SES.CLS + MathDis.STU +
  MathDis.CLS + (1|ID)) )

# final parameter estimates and inferences
testEstimates(fit, var.comp=TRUE)

# EXAMPLE 2

# group mean centering
impList <- within( impList, { CognAbility.CLS <- clusterMeans(CognAbility,ID)
                              SES.CLS <- clusterMeans(SES,ID) } )
impList <- within( impList, { CognAbility.STU <- CognAbility - CognAbility.CLS
                              SES.STU <- SES - SES.CLS } )

# fitting the model of interest
fit <- with( impList, { lmer(MathAchiev ~ 1 + CognAbility.STU + CognAbility.CLS +
                        SES.STU + SES.CLS + (1+CognAbility.STU|ID), REML=FALSE) } )

# final parameter estimates and inferences
testEstimates(fit, var.comp=TRUE)
```

The console output for the single-level MI procedure is given below. Where necessary, the output is truncated to promote readability.

```
# EXAMPLE 1

> testEstimates(fit.MA, var.comp=TRUE)

Call:

testEstimates(model = fit.MA, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)   500.272     1.912   261.671 35390.938     0.000     0.056     0.053
```

```
                        Estimate
Intercept~~Intercept|ID 1122.913
Residual~~Residual      9012.998
ICC|ID                     0.111

Unadjusted hypothesis test as appropriate in larger samples.

> testEstimates(fit.SES, var.comp=TRUE)

Call:

testEstimates(model = fit.SES, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)    44.274     0.301   147.322  3902.052     0.000     0.189     0.160

                        Estimate
Intercept~~Intercept|ID   20.822
Residual~~Residual       266.885
ICC|ID                     0.072

Unadjusted hypothesis test as appropriate in larger samples.

> testEstimates(fit.DPM, var.comp=TRUE)

Call:

testEstimates(model = fit.DPM, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)     2.365     0.016   151.629  2758.632     0.000     0.234     0.190

                        Estimate
Intercept~~Intercept|ID    0.062
Residual~~Residual         0.555
ICC|ID                     0.100

Unadjusted hypothesis test as appropriate in larger samples.

> testEstimates(fit, var.comp=TRUE)

Call:

testEstimates(model = fit, var.comp = TRUE)
```

```
Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)   502.268    22.326    22.497  1875.805     0.000     0.298     0.231
SES.STU         1.054     0.080    13.156   621.597     0.000     0.664     0.401
SES.CLS         2.474     0.303     8.169  1779.882     0.000     0.309     0.237
MathDis.STU   -21.609     1.816   -11.899   515.557     0.000     0.780     0.440
MathDis.CLS   -47.165     5.764    -8.183  2856.929     0.000     0.229     0.187

                        Estimate
Intercept~~Intercept|ID  592.132
Residual~~Residual      8387.913
ICC|ID                     0.066

Unadjusted hypothesis test as appropriate in larger samples.

# EXAMPLE 2

> testEstimates(fit, var.comp=TRUE)

Call:

testEstimates(model = fit, var.comp = TRUE)

Final parameter estimates and inferences obtained from 100 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV       FMI
(Intercept)    79.514    20.562     3.867  7799.684     0.000     0.127     0.113
CognA...STU     6.138     0.154    39.916  1847.682     0.000     0.301     0.232
CognA...CLS     7.549     0.501    15.082  3893.277     0.000     0.190     0.160
SES.STU         0.599     0.075     8.007   505.005     0.000     0.795     0.445
SES.CLS         1.254     0.291     4.314  1298.812     0.000     0.381     0.277

                                    Estimate
Intercept~~Intercept|ID              417.802
Intercept~~CognAbility.STU|ID          1.470
CognAbility.STU~~CognAbility.STU|ID    1.381
Residual~~Residual                  6547.366
ICC|ID                                 0.060
```

Listwise Deletion (LD)

Listwise deletion (LD) was implemented with base R. For simplicity, we do not present the calculation of the ICCs in Example 1 (see above).

```
# EXAMPLE 1

# omit cases with missing data
datLD <- dat[,c("ID","MathAchiev","MathDis","SES")]
datLD <- na.omit(datLD)

# group mean centering
datLD <- within( datLD, { MathDis.CLS <- clusterMeans(MathDis,ID)
                          SES.CLS <- clusterMeans(SES,ID) } )
datLD <- within( datLD, { MathDis.STU <- MathDis - MathDis.CLS
                          SES.STU <- SES - SES.CLS } )

# null models for estimating ICCs
summary( lmer(MathAchiev ~ 1 + (1|ID), data=datLD) )
summary( lmer(SES ~ 1 + (1|ID), data=datLD) )
summary( lmer(MathDis ~ 1 + (1|ID), data=datLD) )

# fitting the model of interest
fit <- lmer(MathAchiev ~ 1 + SES.STU + SES.CLS + MathDis.STU + MathDis.CLS +
       (1|ID), data=datLD)

summary(fit)

# EXAMPLE 2

# omit cases with missing data
datLD <- dat[,c("ID","MathAchiev","SES","CognAbility")]
datLD <- na.omit(datLD)

# group mean centering
datLD <- within(datLD, { CognAbility.CLS <- clusterMeans(CognAbility,ID)
                         SES.CLS <- clusterMeans(SES,ID) } )
datLD <- within(datLD, { CognAbility.STU <- CognAbility - CognAbility.CLS
                         SES.STU <- SES - SES.CLS } )

# fitting the model of interest
fit <- lmer(MathAchiev ~ 1 + CognAbility.STU + CognAbility.CLS +
       SES.STU + SES.CLS + (1+CognAbility.STU|ID), data=datLD, REML=FALSE)

summary(fit)
```

The console output for LD is given below. Where necessary, the output is truncated to promote readability.

```
# Example 1

> summary( lmer(MathAchiev ~ 1 + (1|ID), data=datLD) )
```

```
...

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept) 1137     33.7
 Residual             8787     93.7
Number of obs: 2192, groups:  ID, 381
...

> summary( lmer(SES ~ 1 + (1|ID), data=datLD) )
...

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept)  37.9     6.16
 Residual             244.2    15.63
Number of obs: 2192, groups:  ID, 381
...

> summary( lmer(MathDis ~ 1 + (1|ID), data=datLD) )
...

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept) 0.101    0.318
 Residual             0.500    0.707
Number of obs: 2192, groups:  ID, 381
...

> summary(fit)
...

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept)  732     27.1
 Residual             8299     91.1
Number of obs: 2192, groups:  ID, 381

Fixed effects:
            Estimate Std. Error t value
(Intercept)  505.911     20.063   25.22
SES.STU        0.849      0.138    6.16
SES.CLS        1.752      0.275    6.36
MathDis.STU  -21.874      3.054   -7.16
MathDis.CLS  -31.552      5.689   -5.55
...

# Example 2
```

```
> summary(fit)

Linear mixed-effects model fit by maximum likelihood
...

Random effects:
 Groups   Name            Variance Std.Dev. Corr
 ID       (Intercept)      540.92  23.26
          CognAbility.STU    1.57   1.25    0.19
 Residual                 6287.60  79.29
Number of obs: 4646, groups:  ID, 383

Fixed effects:
                Estimate Std. Error t value
(Intercept)      96.7919    25.9564     3.7
CognAbility.STU   6.2503     0.1849    33.8
CognAbility.CLS   7.7142     0.5835    13.2
SES.STU           0.5778     0.0786     7.4
SES.CLS           0.7487     0.2534     3.0
...
```

Supplement C: Latent versus Manifest Group Means

One crucial aspect of the analysis model in Example 1 is that predictors are decomposed into separate within- and between-group portions. When imputing multilevel missing data, the `pan` model assumes a latent decomposition of all variables that have missing data. By contrast, the analysis model from the main article in Example 1 used manifest group means for the two predictor variables. It has been argued that choosing latent or manifest group means has little importance for the imputation model and that both approaches can be expected to yield similar results in subsequent analyses (Carpenter & Kenward, 2013; Lüdtke, Robitzsch, & Grund, in press; Mistler, 2015). For the analysis model, however, the two approaches tend to produce different results (see Lüdtke et al., 2008).

For this reason, we re-estimated the analysis model in Example 1 in an alternative specification with latent group means for the predictor variables. The same imputation procedure was used as in the main article, and the data sets were analyzed using M*plus*. The `mitml` package was used to save the imputed data sets in a format that is suitable for the analysis in M*plus* (`write.mitmlMplus`). Here we provide the M*plus* syntax file for specifying the analysis model and a brief summary of its results in comparison with the manifest analysis models. The syntax for the latent analysis model is as follows.

```
TITLE:
Example 1: two-level regression model using latent group means
for imputation using pan;

DATA:
file = mplusExample1_list.dat;
type = imputation;

VARIABLE:
names = ID MathAchiev MathDis SES ReadAchiev CognAbility ReadDis SchClimate;
usevariables = MathAchiev SES MathDis;
cluster = ID;

ANALYSIS:
type = twolevel;

MODEL:

%within%
MathAchiev ON MathDis SES;

%between%
MathAchiev ON MathDis SES;
```

In Table 1, we present the results for the latent and the manifest analysis models. The imputations were generated using `pan` and are identical to those that were used in the main article in Example 1. The analyses were carried out in M*plus* in both cases. The syntax for the manifest model is given in Supplement A.

As shown in Table 1, the results for the manifest model were almost identical to those presented in the article, which were obtained using `lme4` for model fitting (Bates, Maechler, Bolker, & Walker, 2014). The difference between the latent and manifest specifications of group means was most visible for the group-level effects, which are being corrected for the unreliability of the group means at the cost of larger standard errors in the latent model (Asparouhov & Muthén, 2006; Lüdtke et al., 2008). The intercept and the student-level

Table 1
Results from Latent and Manifest Analysis Models for the Multilevel MI Procedure in Example 1

	Manifest Covariate		Latent Covariate	
	Estimate	SE	Estimate	SE
Intercept	500.281	1.615	500.277	1.614
SES_{ij}	1.065	0.086	1.063	0.086
$\overline{SES}_j$	2.150	0.265	2.626	0.390
DPM_{ij}	−20.736	2.041	−20.769	2.044
$\overline{DPM}_j$	−41.112	5.234	−44.368	6.849
Var(υ_{0j})	648.099		606.942	
Var(ϵ_{ij})	8304.221		8302.114	

Note. Estimates were significant at $p < .001$; SE = standard error; SES = socio-economic status; DPM = disciplinary problems in math class; υ_{0j} = random intercepts; ϵ_{ij} = residuals at Level 1.

effect did not differ much between the two models. The difference observed for the group-level effects was consistent with earlier studies, which found that the manifest approach can produce different estimates for group-level effects than the latent approach (e.g., Lüdtke et al., 2008).

References

Asparouhov, T., & Muthén, B. O. (2006). *Constructing covariates in multilevel regression* (Mplus Web Notes No. 11). Retrieved from http://www.statmodel.com/

Asparouhov, T., & Muthén, B. O. (2010). *Multiple imputation with Mplus* (Technical Appendix). Retrieved from http://statmodel.com/

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2014). lme4: Linear mixed-effects models using Eigen and S4 (Version 1.1-7) [Computer software]. Retrieved from http://CRAN.R-project.org/package=lme4

Carpenter, J. R., & Kenward, M. G. (2013). *Multiple imputation and its application*. Hoboken, NJ: Wiley.

Grund, S., Robitzsch, A., & Lüdtke, O. (2016). mitml: Tools for multiple imputation in multilevel modeling (Version 0.3-2) [Computer software]. Retrieved from http://CRAN.R-project.org/package=mitml

Lüdtke, O., Marsh, H. W., Robitzsch, A., Trautwein, U., Asparouhov, T., & Muthén, B. O. (2008). The multilevel latent covariate model: A new, more reliable approach to group-level effects in contextual studies. *Psychological Methods*, *13*, 203–229. doi: 10.1037/a0012869

Lüdtke, O., Robitzsch, A., & Grund, S. (in press). Multiple imputation of missing data in multilevel designs: A comparison of different strategies. *Psychological Methods*.

Mistler, S. A. (2015). *Multilevel multiple imputation: An examination of competing methods* (Doctoral dissertation, Arizona State University). Retrieved from http://repository.asu.edu/

Muthén, L. K., & Muthén, B. O. (2012). *Mplus user's guide* (7th ed.). Los Angeles, LA: Muthén & Muthén.

R Core Team. (2015). R: A language and environment for statistical computing (Version 3.2.1) [Computer software]. Retrieved from http://www.R-project.org/

Schafer, J. L., & Zhao, J. H. (2014). pan: Multiple imputation for multivariate panel or clustered data (Version 0.9) [Computer software]. Retrieved from http://CRAN.R-project.org/package=pan

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3), 1–67. Retrieved from http://www.jstatsoft.org/