A. Univariate Random-Intercepts Model in Proc MIXED

Software programs for estimating parameters of mixed-effects model such as SAS Proc Mixed and HLM, are designed to accept data in a univariate format where each row in the data matrix is an observation for a single individual as in Equation 15. Programs vary in the manner in which fixed- and random-effects parts of the model are actually specified in the script. A clear understanding of how the mixed-effects model matrices (**X**, **Z**, **R** and **Z**) are constructed internally by the software is critical in being able to fit complex models such as a multivariate mixed-effects model. For purely illustrative purposes we use a contrived dataset with a total of 9 observations (3 individuals nested within 3 clusters).

Individual_id	Cluster_id	Y
1	1	9.03
2	1	9.54
3	1	10.80
4	2	1.22
5	2	3.12
6	2	2.18
7	3	15.75
8	3	13.69
9	3	14.13

For fitting a random-intercepts model in Proc Mixed, the data must be in a univariate format with outcome for each individual in a separate row. Data must include a variable indicating cluster membership for each individual observation. This ***cluster_id*** variable is flagged as a classification variable by including the variable in the class statement :

```
CLASS cluster_id;
```

The fixed- and random-effects parts of the mixed-model are specified in the `MODEL` and `RANDOM` statements, respectively. The fixed-effects (β) are specified by including the list of fixed predictors (**X** matrix) for the outcome in the `MODEL` statement. For the unconditional random-intercepts model, there are no predictors and the **X** matrix simply contains a vector of "1" for all individuals. Most programs including Proc Mixed include a column of "1" as the first column in the **X** matrix by default. As a result, the `MODEL` statement need not include explicit predictors

```
MODEL Y = /s;
```

The resulting **X** matrix looks like:

$$\mathbf{X} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

The random-effects are specified by a) identifying the set of variables whose effects are random; and b) identifying the clustering unit or the variable which indicates the cluster membership (i.e., cluster_id). For the random-intercepts model, the random predictor is the intercept; and the clustering unit is the classification variable identified in the class statement. The random intercept is specified in the `RANDOM` statement as

"`RANDOM int/subject = cluster_id;`".

With this specification, the software internally constructs the random-effects design matrix (**Z**) with as many columns as the number of clusters defined by the variable cluster_id. The q^{th} column of **Z** matrix has 1 for all observations that belong to the q^{th} cluster:

$$\mathbf{Z} = \begin{bmatrix} 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & 1 & \\ & & 1 \\ & & 1 \\ & & 1 \end{bmatrix}.$$

The option "`subject = cluster_id`" in the `RANDOM` statement also tells the software that the random-effect of each cluster_id is independent and therefore the software internally constructs a diagonal **G** matrix with equal between variances:

$$\mathbf{G} = \begin{bmatrix} \sigma_b^2 & & \\ & \sigma_b^2 & \\ & & \sigma_b^2 \end{bmatrix}.$$

As in ordinary multiple regression, in the absence of any explicit specification of the residual covariance matrix (**R**) it is assumed that, conditional on random-effects, individual residuals are independently and identically distributed. Hence, by default the software constructs a diagonal R matrix with equal within variances:

$$R=\begin{bmatrix}
\sigma_w^2 &&&&&&&& \\
& \sigma_w^2 &&&&&&& \\
&& \sigma_w^2 &&&&&& \\
&&& \sigma_w^2 &&&&& \\
&&&& \sigma_w^2 &&&& \\
&&&&& \sigma_w^2 &&& \\
&&&&&& \sigma_w^2 && \\
&&&&&&& \sigma_w^2 & \\
&&&&&&&& \sigma_w^2
\end{bmatrix}.$$

The complete SAS script for fitting the random-intercepts model is as follows.

```
      /*
      SAS script for fitting univariate random-intercepts model.
      The data consists of 9 rows of y_ij along with a classification variable
indicating cluster membership. There are three subjects per cluster and the
data are sorted by clusters:
      */
      DATA uni;
      INPUT individual_ID cluster_ID Y;
      CARDS;
      1 1 9.03
      2 1 9.54
      3 1 10.80
      4 2 1.22
      5 2 3.12
      6 2 2.18
      7 3 15.75
      8 3 13.69
      9 3 14.13
      ;
      TITLE "random-intercepts model";
      PROC MIXED DATA=uni COVTEST METHOD = ml;
      CLASS cluster_ID;
      MODEL y = /SOLUTION;
      RANDOM int/SUBJECT = cluster_ID G;
      RUN;
```

B. Univariate Random-Intercepts Model in Mx: I

The following Mx script for fitting the random-intercepts model is neither practical nor is it computationally efficient for any non-trivial model or dataset. The sole purpose of this Mx script is to make the structure of various mixed-effects model explicit and also to demonstrate that the likelihood of the model is defined for the complete data vector for the whole sample. The matrices used in this Mx script correspond directly to those defined above for the mixed-effects model.

In practice, all mixed-effects model software such as Proc MIXED recognize that observations are independent across clusters and hence the likelihood can be computed one cluster at time. In Proc MIXED "subject=cluster_ID" option in the random-statement indicates independence across levels of the cluster_id variable.

The DATA FILE (ran.int.mixed.txt) contains a SINGLE row of 9 observations. Note that within each cluster, the order of individuals is irrelvant.

```
9.03 9.54 10.80 1.22 3.12 2.18 15.75 13.69 14.13
```

The Mx script is similar in structure to a LISREL script with minor differences. The three main differences are: (a) Any arbitrary matrix can be defined in a matrix definition block. (b) Fairly complex matrix computations can be accomplished in a matrix algebra block. (c) Model implied mean and covariance matrices have to be explicitly specified.

```
! COMMENT: Text following an exclamation mark is a comment
! Mx allows arbitrary number of groups.
! This example needs a single group.

#Ngroups 1
TITLE: Random-intercepts model: mixed-effects model formulation

! DATA group with number of variables and observations:
DATA Ninput=9 Nobservations=1

!Input file name:
RECT FILE= ran.int.mixed.txt

! VARIABLE LABEL:
! The first index indicates the person ID
! The second index indicates the cluster ID
! Within each cluster, the order of individuals is irrelvant.
Labels
Y1_1 Y2_1 Y3_1 Y1_2 Y2_2 Y3_2 Y1_3 Y2_3 Y3_3

! ALL matrices used in the script are defined
! within the "MATRICES" block:
! Matrix definition includes the shape and
! size of the matrix along with symmetry information.
Begin Matrices;
! G: covariance matrix among cluster-effects (u0j).
! G is a symmetric matrix with 3 rows and 3 columns:
G SYMM 3 3

! R: covariance matrix among individual residuals (eij).
! R is symmetric with 9 rows and 9 columns:
R SYMM 9 9

! Note: Both G and R matrices could have been declared as "diagonal"

! Z: random-effects design matrix:
! dummy variables indicating cluster membership.
! Z is a FULL matrix with 9 rows and 3 columns:
Z FULL 9 3

! X: design matrix for fixed-effects (vector of "1").
! X is a FULL matrix with 9 rows and 1 column:
! "Unit" indicates that all elements of the 9 by 1
! column vector are fixed to 1.0.
X UNIT 9 1

! B: fixed-effects coefficients vector.
B full 1 1

End Matrices;

! Free and Fixed parameters in matrices are
! flagged just as in  LISREL. The following
! matrices look just like the corresponding
! mixed model matrices described above.
! SP or SPECIFICAION is used to indicate
! free (positive integer) or fixed elements (zero)
! in a matrix.
! An integer value such as "11" at multiple
! places constrains the parameter to be equal.
! MA or Matrix is used to provide start values.

! G: Diagonal with equal variances
! Diagonal elements of G are constrained to equality.
! This matrix could have been more easily
! specified as "DIAGONAL 3 3".

SPECIFY G !Free/Fixed values in the G matrix
11
0 11
0 0 11

MATRIX G !Start Values
1
0 1
0 0 1

! R: Diagonal with equal variances
! Diagonal elements of R are constrained to equality.
! This matrix could have been more easily
! specified as "diagonal 9 9".

SPECIFY R
12
00 12
00 00 12
00 00 00 12
00 00 00 00 12
00 00 00 00 00 12
00 00 00 00 00 00 12
00 00 00 00 00 00 00 12
00 00 00 00 00 00 00 00 12

MATRIX R  !Start values
02
00 02
00 00 02
00 00 00 02
00 00 00 00 02
00 00 00 00 00 02
00 00 00 00 00 00 02
00 00 00 00 00 00 00 02
00 00 00 00 00 00 00 00 02

! Z: is a 9 x 3  matrix of dummy variables
! indicating cluster membership for each person
! Since no element of the Z-matrix is free,
! it is not necessary to use "SPECIFICATION".
MATRIX Z    !Fixed values
1 0 0
1 0 0
1 0 0
```

```
0 1 0
0 1 0
0 1 0
0 0 1
0 0 1
0 0 1

!B: single fixed-effects parameter, i.e. mean intercept
Specify B
13
Matrix B
6

! All computations are specified in the "algebra" block:
! Model implied means and covariances expressions
! are identical to those in a CFA model
BEGIN ALGEBRA;
!V: model-implied covariance matrix for the entire sample
 V = Z*G*Z' + R;
!M: model-implied means for the entire sample
 M = X*B;
END ALGEBRA;

! Model implied mean and covariance matrices are
! supplied so that Mx can estimate model parameters.
Mean M/
Covariance V/

End GROUP !END of the Mx script
```

C. Univariate Random-Intercepts Model in Mx: II

The next Mx script shows how the previous random-intercepts model may be fitted as an ordinary restricted CFA model in Mx. Because the observations are independent across clusters, the likelihood can be computed separately for each cluster. Note that in the presence of a large number of individuals within a cluster and with multivariate data, cluster level likelihood may still result in very large covariance matrix. In practice, conditional independence of individuals given the cluster level random-effects is exploited for computing the likelihood efficiently. The current script however is *not* computationally efficient. The sole purpose of this Mx script is to illustrate how the mixed-effects model explicit can be reformulated as a CFA model.

The `DATA FILE` (ran.int.cfa.txt) contains a three rows of observations with scores for the three individuals in each row:

```
9.03 9.54 10.80
1.22 3.12 2.18
15.75 13.69 14.13
```

Note that within each cluster, the order of individuals is irrelvant.

```
#Ngroups 1
TITLE: Random-intercepts model, CFA (parallel-test) formulation

DATA Ninput=3 Nobservations=3
!3 clusters with a maximum of 3 observations per cluster

RECT FILE=ran.int.cfa.txt
!raw-data file is ran.int.cfa.txt

Labels
Y1 Y2 Y3
! Model matrices for the restricted CFA are simply cluster-level sub-matrices
! of the full X, Z, G, and R matrices. These are described in the paper.

Begin Matrices;
!Gj: Between-cluster variance: var(u0j).
Gj SYMM 1 1

!Rj: within-cluster residual variance: var(eij).
Rj SYMM 3 3

!Zj: random-effects design matrix (vector of 1s).
Z FULL 3 1

!Xj: fixed-effects  design matrix (vector of 1s).
! Note Xj is identical Zj
X FULL 3 1

!B:  vector of fixed-effects coefficient
B FULL 1 1

End Matrices;

!Gj matrix specification: a single free element
SPECIFY G
11
!Gj matrix start-values: most values will work
MATRIX G
1.1

!Rj matrix specification: diagonal with equal variances
SPECIFY R
12
00 12
00 00 12
!Rj matrix start-values: most start values will work
MATRIX R
02
00 02
00 00 02

!Zj vector of "ones". number of rows = nij
MATRIX Z
1
1
1
```

```
!Xj vector of "ones". number of rows = $n_{ij}$
MATRIX X
1
1
1

!B single fixed-effects parameter (grand-mean or intercept, $\beta_{00}$)
Specify B
13

!B start-value of the grand-mean
Matrix B
6

BEGIN ALGEBRA;

!Vj: computing model implied covariance matrix (for cluster j)
V = Z*G*Z' + R;

!Mj: computing model implied means (for cluster j)
M = X*B;

END ALGEBRA;

Mean M/         ! Mj is the matrix of model-implied means
Covariance V/   ! Vj is the matrix of model-implied covariance

End
```

D. Bivariate Random-Intercepts Model in Proc Mixed

The SAS dataset for the example was created as follows:

```
DATA bivar;
INPUT cluster_ID individual_id variable_id $ Y;
CARDS;
1 1 P 11.1
1 1 Q 20.6
1 2 P 13.3
1 2 Q 16.0
1 3 P 14.3
1 3 Q 15.1
2 4 P 18.6
2 4 Q 8.9
2 5 P 19.2
2 5 Q 7.3
2 6 P 21.5
2 6 Q 3.6
3 7 P 10.4
3 7 Q 2.7
3 8 P 11.1
3 8 Q 3.2
3 9 P 12.8
3 9 Q 5.0
;
run;
data bivar;
set bivar;
IF variable_ID eq "P" then Dp = 1; else Dp = 0;
IF variable_ID eq "Q" then Dq = 1; else Dq = 0;
run;
```

There are four variables in the dataset. The first three variables are ID variables and are self-explanatory: (a) cluster_ID; (b) individual_ID and (c) variable_ID. There are a total of 9 individuals nested within 3 clusters. Each individual has two rows of data, representing scores on two separate variables *P* and *Q*. The character variable representing the variable label or variable_ID is used to create two additional numeric dummy-variables Dp and Dq. It is obvious that the additional variables are redundant. However, as will be seen below, both numeric and character variables are required by Proc MIXED.

The bivariate mixed-effects model is best understood in terms of matrices the four mixed-effects model matrices (**X**, **β**, **Z**, **G**, & **R**). The following equation presents the bivariate mixed effects model.

$$Y=\begin{bmatrix}11.1\\20.6\\13.3\\16.0\\14.3\\15.1\\18.6\\8.9\\19.2\\7.3\\21.5\\3.6\\10.4\\2.7\\11.1\\3.2\\12.8\\5.0\end{bmatrix}=\begin{bmatrix}1&0\\0&1\\1&0\\0&1\\1&0\\0&1\\1&0\\0&1\\1&0\\0&1\\1&0\\0&1\\1&0\\0&1\\1&0\\0&1\\1&0\\0&1\end{bmatrix}\begin{bmatrix}\beta^P\\\beta^Q\end{bmatrix}+\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}u_1^P\\u_1^Q\\u_2^P\\u_2^Q\\u_3^P\\u_3^P\end{bmatrix}+\begin{bmatrix}e_{11}^P\\e_{11}^Q\\e_{21}^P\\e_{21}^Q\\e_{31}^P\\e_{31}^Q\\e_{12}^P\\e_{12}^Q\\e_{22}^P\\e_{22}^Q\\e_{32}^P\\e_{32}^Q\\e_{13}^P\\e_{13}^Q\\e_{23}^P\\e_{23}^Q\\e_{33}^P\\e_{33}^Q\end{bmatrix}$$

Note that this equation corresponds to the general mixed-effects model: $\mathbf{y}=\mathbf{X\beta}+\mathbf{Zu}+\mathbf{e}$, where **y** is the data vector (18 x 1); **X** and **Z** are fixed- and random-effects design matrices, respectively (described below); **β** (2 x 1) is a fixed-effects coefficient matrix; **u** is the random-effects vector (6 x 1); and **e** is the vector of person and variable specific errors (18 x 1).

The fixed-effects are specified in the MODEL statement. The bivariate random-intercepts model has two fixed-effects (β^P & β^Q) representing the grand means of the two outcome variables. The corresponding model statement is:

```
MODEL Y = Dp Dq / solution noint;
```

where, Dp and Dq are the two dummy variables described above. Note that unlike conventional regression and univariate mixed-effects regression, there is no column of fixed "1s" in the **X** matrix (18 x 2). This is because a common intercept is meaningless with two distinct outcomes. Inclusion of two dummy variables in lieu of a common intercept allows us to estimate separate means for the two variables. Proc Mixed includes a vector of "1" for estimating the fixed-intercept by default. Since, we are explicitly estimating two separate intercepts for the two outcome variables, "noint" option is used in the model statement to suppress the inclusion of the default intercept. The `MODEL` statement internally generates the **X** matrix presented above with the elements for the two columns in each row fixed to the respective values of Dp and Dq.

The random-effects are specified in the RANDOM statement. In this case, the mean level of each outcome is assumed to vary randomly across cluster. This is indicated by specifying the effects of the two dummy variables (D^P & D^Q) as random at the cluster level:

```
RANDOM Dp Dq / subject = cluster_id type = un;
```

As in the case of the univariate random-intercepts model, subject=cluster_ID allows Proc MIXEd to group observations into separate clusters so that it can construct the **Z** matrix which implicitly contains this clustering information. The two numeric dummy variables: D^P and D^Q along with the Cluster_ID instructs Proc Mixed to internally construct **Z** matrix (18 x 6) as shown above. Note that the **Z** matrix serves to link cluster-level means for each outcome with the appropriate observations for individuals within a given cluster. There are six rows in the **Z** matrix, two each for each of the three clusters. Individuals in the first cluster have non-zero values in the $\mathbf{Z}_1$ sub-matrix (first 6 rows and first 2 columns). Each $\mathbf{Z}_j$ sub-matrix, is identical to the corresponding $\mathbf{X}_j$ sub-matrix.

The random statement also instructs the software to construct a random-effects covariance matrix or the **G** matrix (6 x 6):

$$\mathbf{G} = \begin{bmatrix} \Sigma^b & & \mathbf{0} \\ & \Sigma^b & \\ \mathbf{0} & & \Sigma^b \end{bmatrix} = \begin{bmatrix} \sigma_P^{2(b)} & & & & & \mathbf{0} \\ \sigma_{Q,P}^{(b)} & \sigma_Q^{2(b)} & & & & \\ & & \sigma_P^{2(b)} & & & \\ & & \sigma_{Q,P}^{(b)} & \sigma_Q^{2(b)} & & \\ & & & & \sigma_P^{2(b)} & \\ \mathbf{0} & & & & \sigma_{Q,P}^{(b)} & \sigma_Q^{2(b)} \end{bmatrix}.$$

The G matrix is block-diagonal with 3 blocks representing the between-clusters covariance matrix for each of the three clusters. The random-effects are independent across clusters. The covariance between the two random effects is identical for each cluster.

The residual-covariance matrix or the covariance matrix for the individual level error terms is specified in the REPEATED statement. The REPEATED statement models the residual-covariance for the lowest unit of observation i.e., the individual. For each individual, there are multiple rows of observations- each observation representing a different outcome variable. As a result, Proc Mixed must be explicitly instructed about: (a) the id variable reflecting the lowest unit of observation (individual_id); and, (b) the id variable indicating the type of the outcome variable in a given row of observation (variable_id). The input dataset already has this information. This information is communicated to Proc MIXED. Within Proc MIXED, the CLASS statement is used to flag variables containing classification information:

```
CLASS individual_id cluster_id variable_id;
```

The residual covariance matrix (**R**) can now be specified as:

```
REPEATED variable_id / subject = individual_id ( cluster_id) type = un;
```

With this specification, the software constructs a symmetric block (**R**$_{ij}$) for each individual identified by a unique individual_id variable, the variable_id allows construction of separate variances for each separate outcome for each individual. For each individual, the **R**$_{ij}$ sub-matrix contains as many rows and columns as the number of observations (outcome variables) for that individual, in this case (2 x 2). Hence, for a subject that has both observations (Y^P or Y^Q) the data matrix would have two rows of data with variable_id is equal to "*P*" and "*Q*", respectively. Hence, the REPEATED statement would construct a full two by two block containing the residual covariance between e_{ij}^P and e_{ij}^Q . If a given individual only had a single observation (Y^Q) as indicated by variable_id, then the **R** matrix block for that individual would have a single element. For the above example the **R** matrix has 6 identical blocks of within-clusters covariance along the diagonal:

$$\mathbf{R} = \begin{bmatrix} \boldsymbol{\Sigma}^w & & & & & & & & \mathbf{0} \\ & \boldsymbol{\Sigma}^w & & & & & & & \\ & & \boldsymbol{\Sigma}^w & & & & & & \\ & & & \boldsymbol{\Sigma}^w & & & & & \\ & & & & \boldsymbol{\Sigma}^w & & & & \\ & & & & & \boldsymbol{\Sigma}^w & & & \\ & & & & & & \boldsymbol{\Sigma}^w & & \\ & & & & & & & \boldsymbol{\Sigma}^w & \\ \mathbf{0} & & & & & & & & \boldsymbol{\Sigma}^w \end{bmatrix},$$

and

$$\boldsymbol{\Sigma}^w = \begin{bmatrix} \sigma_P^{2(w)} & \\ \sigma_{Q,P}^{2(w)} & \sigma_Q^{2(w)} \end{bmatrix}.$$

The complete code for fitting the random-intercepts model is:

```
TITLE "bivariate random-intercepts model";
PROC MIXED DATA=bivar COVTEST METHOD = ml;
CLASS cluster_ID individual_id variable_id ;
MODEL y = Dp Dq / SOLUTION NOINT;
RANDOM Dp Dq / SUBJECT = cluster_ID TYPE=UN G GCORR;
REPEATED variable_id / SUBJECT = individual_ID(cluster_ID) TYPE=UN R RCORR;
RUN;
```

E. Bivariate Random-Intercepts Model in Mx

In Mx, the model is fitted at the level of the cluster, with data from all individuals and outcomes within a cluster serving as distinct observations. As in the case of the univariate model, the multivariate multilevel data are also arranged with a single row of observations per cluster. Each row has data for three individuals and two observations per individual.

```
! Data file (bivar.ran.int.cfa.txt ):
! 3 rows with 3 subjects * 2 variables = 6
! observations each:
! Each row represents data for a single cluster
! 11.1 20.6 13.3 16.0 14.3 15.1
! 18.6 8.9 19.2 7.3 21.5 3.6
! 10.4 2.7 11.1 3.2 12.8 5.0
```

Because the model is defined at the cluster level, we only need cluster level submatrices of $\mathbf{X}j$, $\mathbf{Z}_j$, $\mathbf{G}_j$ and $\mathbf{R}_j$. The following matrices are specified in the Mx script:

$$\mathbf{X}_j = \mathbf{Z}_j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$\mathbf{G}_j = \Sigma^b = \begin{bmatrix} \sigma_P^{2(b)} & \\ \sigma_{Q,P}^{(b)} & \sigma_Q^{2(b)} \end{bmatrix}.$$

$$\mathbf{R}_j = \begin{bmatrix} \mathbf{\Sigma}^w & & \\ & \mathbf{\Sigma}^w & \\ & & \mathbf{\Sigma}^w \end{bmatrix}$$

$$\mathbf{R}_{ij} = \mathbf{\Sigma}^w = \begin{bmatrix} \sigma_P^{2(w)} & \\ \sigma_{Q,P}^{2(w)} & \sigma_Q^{2(w)} \end{bmatrix}.$$

The reader should compare these cluster level matrices specified in Mx with the corresponding full matrices presented above for the mixed-effects model. Also note that the $\mathbf{X}_j$ and $\mathbf{Z}_j$ are identical and are fixed by design as in the mixed-effects model ($\mathbf{\Lambda}_j = \mathbf{X}_j = \mathbf{Z}_j$). Although the following script uses separate $\mathbf{X}_j$ and $\mathbf{Z}_j$ matrices, a single factor-loading matrix could have been used to represent both. In the general case, the factor-loading matrix may be specified as: $\mathbf{\Lambda}_j = \mathbf{1}_{N_j} \otimes \mathbf{I}_{n_y}$. The $\mathbf{G}_j$ matrix corresponds to the factor-covariance matrix: ($\mathbf{G}_j = \mathbf{\Psi}$; n_y x n_y). The $\mathbf{R}_j$ matrix corresponds to the individual level residual-covariance matrix ($\mathbf{R}_j = \mathbf{\Theta}$). With three individuals in each cluster, the $\mathbf{R}_j$ matrix has three blocks of individual level residual-covariances, $\mathbf{R}_{ij}$ (n_y x n_y) along the diagonal. In the general case, the residual covariance may be specified as: $\mathbf{R}_j = \mathbf{\Theta}_j = \mathbf{I}_{N_j} \otimes \mathbf{\Sigma}^w$. The following Mx script uses such a specification. Finally, the fixed-effects coefficients vector ($\boldsymbol{\beta}$) corresponds to the means of the between-level latent factors. The following Mx script specifies these matrices.

```
! Data file (bivar.ran.int.cfa.txt ):
! 3 rows with 3 subjects * 2 variables = 6
! observations each:
! Each row represents data for a single cluster
! 11.1 20.6 13.3 16.0 14.3 15.1
! 18.6 8.9 19.2 7.3 21.5 3.6
! 10.4 2.7 11.1 3.2 12.8 5.0

#Ngroups 1

TITLE: Bivariate random-intercepts model, CFA (parallel-test)
formulation
DATA Ninput=6 Nobservations=3
!3 clusters with a maximum of 3 observations per cluster

RECT FILE=bivar.ran.int.mixed.txt
!raw-data file is bivar.ran.int.cfa.txt

Labels
YP1 YQ1 YP2 YQ2 YP3 YQ3
! Variables Yp and Yq for three subjects in cluster j
! YP1 and YQ1 are outcomes P and Q for person 1
! YP2 and YQ2 are outcomes P and Q for person 2
! YP3 and YQ3 are outcomes P and Q for person 3

Begin Matrices;
! In Mx, the model is specified at the cluster level.
! Hence, only the cluster level sub-matrices of
! $X_j$, $Z_j$, $G_j$ and $R_j$ need to be specified.
G  SYMM 2 2 Free
!Gj: Between-cluster covariance

W  SYMM 2 2 Free
!Wij: within-cluster residual variance

Z full 6 2
!Zj: random-effects design matrix

X full 6 2
!Xj: fixed-effects  design matrix

B full 2 1 Free
!B:  vector of fixed-effects coefficients.

I Iden 3 3  ! I is an identity matrix (size = Nj)
End Matrices;
```

```
!Gj matrix specification:
! Gj matrix start-values: most values will work
MATRIX G
2.1
1 3.1

!Wij matrix start-values: most start values will work
MATRIX W
02
00 02

! Zj Random-effects design matrix:
! number of rows = ny*Nj
! number of columns = ny
MATRIX Z
1 0
0 1
1 0
0 1
1 0
0 1

! Xj Fixed-effects design matrix.
! number of rows = ny*Nj,
! number of columns = ny
MATRIX X
1 0
0 1
1 0
0 1
1 0
0 1

! In the general case, both Xj and Zj may be specified as U@C,
! U is (Nj x 1)a unit vector; and
! C is (ny x ny) an identity matrix.

!B start-value of the grand-mean of P and Q
MATRIX B
6 10

BEGIN ALGEBRA;
! Rj: Wij is the expected within-cluster
! covariance for a single individual i
! in cluster j.
! Rj is the covariance matrix for the whole cluster j:
```

```
R = I@W;
!Vj: computing model implied covariance matrix (for cluster j)
V = Z*G*Z' + R;
!Mj: computing model implied means (for cluster j)
M = X*B;
END ALGEBRA;
Mean M/       ! Mj is the matrix of model-implied means
Covariance V/ ! Vj is the matrix of model-implied covariance
End
! NOTE: The formulae for specifying model
! implied means and covariance was identical for all Mx scripts.
! and reflects the general CFA template used for conceptualizing
! multivariate mixed-effects models.
```

F. ML-CFA Model With Across-Level Measurement Invariance in Mplus

The following Mplus script demonstrates a multilevel CFA model with invariance constraints across levels.

DATA: Data are read in with one row of observations per person. It is assumed that a variable is included indicating the *cluster-ID* for each individual. In the following script, TID (teacher-ID) is flagged as the variable containing "CLUSTER" identification.

MODEL: Within- and between-clusters models are specified separately within blocks identified by the label *%WITHIN%* and *%BETWEEN%*, respectively. The specification of a measurement model at either level is just like any other factor model.

WITHIN-MODEL: Single-factor model.

BETWEEN-MODEL: Single-factor model. Latent factor mean is fixed to 0.0, observed-variable intercepts are freely estimated. The second-level residuals for the observed variables are fixed to 0.0, however, residual standard-deviations are estimated by creating dummy latent-variables that are defined by each level-2 (between-clusters) random-intercept. The variances of these dummy-variables are fixed to 1.0 and their factor-loadings are freely estimated. These factor loadings actually reflect the standard-deviation of level-2 (between-clusters) residual variances.

```
    TITLE: ML-CFA, People are Variables Too, Grade 3

    DATA: FILE IS C:\projects\PAVT_revised\mplus\PAVT_mlcfa.txt;

    VARIABLE:
       NAMES ARE
          write
          wordread
          phoaware
          tid
          spell
          picvoc;
       USEVARIABLES =
          write
          wordread
          phoaware
          spell
          picvoc;
! TID is the variable containing CLUSTER IDs.
       CLUSTER = TID;
! Missing-data code is 999
       MISSING ARE ALL (999);

    ANALYSIS:
! A two level model is being fitted and data contain missing values.
! H1 requests Mplus to estimate the base multivariate mixed-effects model
! This allows model comparisons with the more restricted fitted model (H0)
        TYPE = TWOLEVEL MISSING H1;
! Here we use the FIML estimator. Mplus also allows robust estimators.
        ESTIMATOR = ML;
    MODEL:

    %WITHIN%
! Within-Literacy factor (W_L) is "measured BY" the five observed variables.
! the scale of the W_L factor is defined by fixing the loading of "wordread"
! to 1.0. The integers in parentheses are used for imposing equality
! constraints across levels. The use of the same integer any where in the
! script constrains the corresponding parameters to be equal.

    W_L by spell*13 (110) ;
    W_L by wordread@1;
    W_L by phoaware(111);
    W_L by write (112);
    W_L by picvoc(113);
! freely estimate all five residual-variances
    write wordread phoaware spell picvoc;
! freely estimate variance of the within-literacy factor.
    W_L*1;

    %BETWEEN%
! Between-Literacy factor (B_L) is "measured BY" the level-2 random-
! intercepts of all five observed variables.
! The scale of the B_L factor is defined by fixing the loading of "wordread"
! to 1.0. The integers in parenthesis are used for imposing equality
! constraints across levels. The use of the same integer anywhere in the
! script constrains the corresponding parameters to be equal.
! In this case, the factor-loadings are constrained to be equal across
! levels.

    b_L by spell*13 (110) ;
    b_L by wordread@1;
    b_L by phoaware (111);
    b_L by write (112);
    b_L by picvoc (113);

! The variance of the level-2 between-literacy factor,
! (B_L) is freely estimated:
    B_L*1;
! The mean of the level-2 between-literacy factor,
! (B_L) is fixed to 0.0:
    [B_L@0 ];

! The observed measurement-intercepts for all five variables are
! freely estimated.
    [write wordread phoaware spell picvoc];

! The next section is a "trick" for estimating standard deviations of
! level-2 residuals instead of residual variances.
! We define five latent variables, each of these level-2 latent variables
! are measured BY corresponding five level-2 residuals of the
! random-intercepts.

! The factor-loadings of all five dummy latent variables are freely
! estimated. These represent the standard deviations of the
! corresponding level-2 residuals of the random-intercepts:
    s_sp by spell*;
    s_wr by wordread*;
    s_pa by phoaware*;
    s_write by write*;
    s_pv by picvoc*;
! Variances of all five newly created level-2 latent variables
! are FIXED to 1.0:
    s_sp@1 s_wr@1 s_pa@1 s_write@1  s_pv@1;

! ALL FIVE dummy latent variables are uncorrelated amongst themselves:
    s_sp with    s_wr@0 s_pa@0 s_write@0 s_pv@0;
    s_wr with    s_pa@0 s_write@0 s_pv@0;
    s_pa with    s_write@0 s_pv@0;
    s_write with s_pv@0;

! ALL FIVE dummy latent variables are uncorrelated with
! the between-literacy factor:
    b_L with s_sp@0 s_wr@0 s_pa@0 s_write@0 s_pv@0;
! Residual variances of the original five level-2 residuals
! are FIXED to 0.0:
    write@0 wordread@0 phoaware@0 spell@0 picvoc@0;

    OUTPUT: residual sampstat standardized;
```

G. Univariate Random-Slopes Model in Mx

A univariate random-slopes model is fitted using the same CFA template as a conventional ML-SEM model. The model includes a random slope for a *strictly* exogenous predictor, in addition to a random intercept. As discussed in the papers random-slopes models are very similar to latent growth curve models (also see Mehta & West, 2000 for an example of fitting a latent growth curve with continuous time).

Data: Each row contains observations (dependent and independent variables) for all individuals from a single cluster. The total number of variables is therefore $2*maxN_j$, where $maxN_j$ is the largest cluster size. In addition, the exogenous predictor whose effect is modeled as random at the higher level is flagged as a *definition-variable*. The following five lines of data should be placed in a file named: "input.txt"

```
10 7 12 4          3 1 3 1
11 2 8  9          5 2 4 6
4  5 1  3          1 2 1 1
7  7 8  9          2 3 2 4
10 12 14 11        6 5 7 8
```

Model: A random-slopes model is simply a factor model in which factor loadings for the first latent factor (intercept) are fixed to 1.0 for all individuals and factor loadings for the second latent factor (slope) are fixed to the person-specific values of the exogenous predictor flagged as a definition-variable. So, we need a method of mapping score for each person y_{ij} with the corresponding value of the exogenous predictor (x_{ij}). This is handled by the variable code assigned to each definition-variable. Mx assigns codes of -1 to $–maxN_j$ to each of the $maxN_j$ definition-variables. Hence, the factor loadings ($\lambda_{i,2}$) for the dependent variable scores of person-1 to person-$maxN_j$ are fixed to values -1 to $–maxN_j$. (see the specification of the factor-loading matrix L below). The rest of the model specification is just like a conventional factor model.

```
! Mx code for fitting a random-slopes model.
TITLE: Random-slopes
#maxNj  4   ! define maximum number of individuals in a cluster
#nFac 2     ! define number of latent factors (intercept and slope)
! Here we assume that maxNj = 4.
DATA NI=8 NO=3   ! Input Data
RECT file = input.txt

! For missing data, the file should be read in as a variable
! length file.

Labels Y1 Y2 Y3 Y4      X1 X2 X3 X4  ! Variable Labels:
! Y1-Y4: Outcome scores for 4 subjects. These are modeled.
! X1-X4: Predictor scores for the same four subjects.
! These are flagged as definition-variables:
DEFINITION_VARIABLES X1 X2 X3 X4;
!Note X1-X4: Individual scores on X

Begin Matrices;
L FULL maxNj nFac
! LAMBDA-Y: Factor-loading matrix
! Four observations (maxNj = 4) by 2 factors (intercept & slope)

P SYMM  nFac nFac Free
! PSI: Latent factor covariance matrix

T DIAG maxNj maxNj Free
! THETA: residual covariance matrix
! Diagonal matrix containing residual variances.

A FULL  nFac 1 Free
! ALPHA: latent factor covariance matrix.

End Matrices;

SPECIFY L
0 -1
0 -2
0 -3
0 -4
! First factor: All factor loadings are fixed (not estimated)
! Second-factor: All factor loadings are fixed to the values
! of the corresponding definition-
! variable. Hence, the factor-loading for person-1 is fixed
! to X1; factor-loading for person-2 is
! fixed to X2, so on and so forth.
MATRIX L
1 0
1 0
1 0
1 0
! First factor: All factor loadings are fixed to value = 1.0
! Second-factor: NO NEED TO SPECIFY ANY VALUES,
! as these were already fixed in the
! LY matrix specification.
```

```
MATRIX P
2
0 1.5
! start values for the PSI matrix

SPECIFY T
33 33 33 33
! The use of identical positive-integer in Theta matrix
! specification indicates that the variances
! are constrained to be equal for all individuals.
! Note, that the matrix is diagonal containing four
! residual-variances,
! reflecting the assumption of independent and identically
! distributed residuals.
MATRIX T
2 2 2 2
! start values for the Theta matrix. Only, diagonal values
! need to be specified.

! The following section is identical to the
! random-intercepts model.
Begin Algebra;
M = L*A;
S = L*P*L' + T;
End Algebra;
Means M;           !Means
Covariance  S;   !Covariances
End
```

H. Bivariate Random-Slopes Model in Mplus

```
TITLE:     This is an example of a two-level
           regression analysis for a continuous
           dependent variable
DATA:      FILE IS bpa_m.txt;
VARIABLE:
      !Variables in the dataset (Assigning labels)
      NAMES ARE  id tid tidu age female prog immer project
   texas urban distno HO AU BR LA
   BWS BNS SWS PES readS WRS LWS
   BWE BNE SWE PEE readE WRE LWE
   tottime TRROBS tteeng ttespa ttemix tsteng tstspa tstmix
   totEPA totSPA totMPA
   totEAP totSAP totMAP OBSYN;
      ! missing data code
      MISSING are .;
      !variables used in the current analysis (subsetiing)
      USEVARIABLES ARE
         tidu readE readS pes;
      ! pes is an exogenous predictor at the within-level:
      WITHIN =   pes;
      ! Cluster-id
      CLUSTER = tidu;
      ! Centering: predictor is centered at the grand-mean
      ! See Mehta & West (2000) for a discussion of centering.
      CENTERING = GRANDMEAN(pes);

! ANALYSIS:
! (a) two-level model for clustered data
! (b) random-effect of a predictor on an outcome.
ANALYSIS:  TYPE = TWOLEVEL RANDOM;

! MODEL specification:
! Model is specified in two parts.
! WITHIN: Model specification for the individuals
! BETWEEN: Model specification for the clusters

  MODEL:
             %WITHIN%
! (a) individual level outcome,
! readE is regressed on individual level predictor: PES;
! (b) The effect of readE on PES is RANDOM "|" at level-2.
! (c) The random-effect at level 2 or the latent slope-factor will be called
! readE_S
! (d) The random-intercept at level 2 will be implictly called readE
             readE_S| readE ON pes;

! (a) individual level outcome,
! readS is regressed on individual level predictor: PES;
! (b) The effect of readE on PES is RANDOM "|" at level-2.
! (c) The random-effect at level 2 or the latent slope-factor will be called
! readS_S
! (d) The random-intercept at level 2 will be implictly called readS

             readS_S| readS ON pes;
```

```
! Individual level residuals of readE and readS are allowed to covary.
            readE with readS;

! At the between-level, there are four latent variables:
! Intercept and Slope for readE:
! These are referred to by labels readE and readE_S
! Intercept and Slope for readS:
! These are referred to by labels readS and readS_S
! ALL FOUR LATENT VARIABLES ARE ALLOWED TO COVARY
! AND THEIR MEANS ARE ESTIMTAED BY DEFAULT.
            %BETWEEN%
            readE with readE_S;
            readS with readS_S;
            readS with readE_S;
            readE with readS_S;
            readE with readS;
            readE_S with readS_S;

  OUTPUT:;
```

Filename:	ML-SEM.annotated.code.doc
Directory:	C:\projects\PAVT_final\web
Template:	C:\Documents and Settings\pmehta\Application Data\Microsoft\Templates\Normal.dot
Title:	Multilevel SEM Models
Subject:	
Author:	Paras Mehta
Keywords:	
Comments:	
Creation Date:	5/9/2005 10:08:00 AM
Change Number:	55
Last Saved On:	6/1/2005 6:53:00 AM
Last Saved By:	Paras Mehta
Total Editing Time:	141 Minutes
Last Printed On:	6/1/2005 8:50:00 AM
As of Last Complete Printing	
Number of Pages:	27
Number of Words:	5,191 (approx.)
Number of Characters:	29,589 (approx.)