Fitting Residual Error Structures for Growth Models in SAS PROC MCMC

Educational and Psychological
Measurement

2017, Vol. 77(4) 587–612

© The Author(s) 2016

Reprints and permissions:

sagepub.com/journalsPermissions.nav

DOI: 10.1177/0013164416652441

journals.sagepub.com/home/epm

Daniel McNeish¹

Abstract

In behavioral sciences broadly, estimating growth models with Bayesian methods is becoming increasingly common, especially to combat small samples common with longitudinal data. Although *Mplus* is becoming an increasingly common program for applied research employing Bayesian methods, the limited selection of prior distributions for the elements of covariance structures makes more general software more advantages under certain conditions. However, as a disadvantage of general software's software flexibility, few preprogrammed commands exist for specifying covariance structures. For instance, PROC MIXED has a few dozen such preprogrammed options, but when researchers divert to a Bayesian framework, software offer no such guidance and requires researchers to manually program these different structures, which is no small task. As such the literature has noted that empirical papers tend to simplify their covariance matrices to circumvent this difficulty, which is not desirable because such a simplification will likely lead to biased estimates of variance components and standard errors. To facilitate wider implementation of Bayesian growth models that properly model covariance structures, this article overviews how to generally program a growth model in SAS PROC MCMC and then demonstrates how to program common residual error structures. Full annotated SAS code and an applied example are provided.

Keywords

MCMC, SAS, latent growth model, Bayes

¹Utrecht University, Utrecht, Netherlands

Corresponding Author:

Daniel McNeish, Utrecht University, PO Box 80140, 3508 TC Utrecht, Netherlands.

Email: d.m.n.mcneish@uu.nl

Research interested in the growth of a construct over time is quite common in behavioral sciences. When modeling this type of data, Bayesian Markov Chain Monte Carlo (MCMC) is becoming increasingly common to estimate growth models, possibly due to advantages such as the ability to incorporate information from prior studies (van de Schoot, Broere, Perryck, Zondervan-Zwijenburg, & Van Loey, 2015; Walker, Gustafson, & Frimer, 2007), to estimate models that are complex or unable to be estimated with maximum likelihood (e.g., Muthén & Asparouhov, 2012; Zhang, Hamagami, Wang, Grimm, & Nesselroade, 2007), and the fact that Bayesian methods in general (of which MCMC is merely one possible estimation method) are less susceptible to bias in the variance components and standard errors with small sample sizes (Hox, van de Schoot, & Matthijsse, 2012; McNeish, 2016a; Meuleman & Billiet, 2009).

As found by van de Schoot (2016), behavioral researchers increasingly prefer to perform MCMC analyses in *Mplus*, presumably due to ease with which such analyses can be implemented. However, as noted by McNeish (2016b), *Mplus* (Version 7) does not currently offer a wide array of prior distributions that are suitable for smaller samples (which are common with longitudinal data) and complex error structures (e.g., autoregressive) must be programmed manually using multiple constraints. Therefore, more general statistical software (e.g., SAS PROC MCMC, WinBUGS, etc.) may be beneficial under such conditions due to a broader selection of prior distributions including the ability to specify the half-Cauchy prior recommended by Gelman (2006) and shown to perform well in a study by McNeish and Stapleton (2016) with sample sizes as low as the mid-single digits.

When resorting to more general statistical software, the covariance structure for the errors is not straightforward to specify. For instance, in SAS PROC MIXED, these structures can be easily programmed via the TYPE = option in the REPEATED statement. However, both in *Mplus* (either Bayes or frequentist modules) and in more general Bayesian software, all but the most basic structures must be programmed manually. Additionally, some of the constraints needed to program certain structures in *Mplus* are not permitted with the BAYES estimator. For instance, the model constraint needed to impose the popular autoregressive structure will result in the following error message:

```
*** ERROR in MODEL CONSTRAINT command
```

```
For analysis with ESTIMATOR=BAYES, NEW and PLOT parameters must  
appear on the left-hand side in at least one MODEL CONSTRAINT equation.  
No equation found containing parameter on the left-hand side: RHC
```

Harring and Blozis (2014) previously noted this difficulty with covariance structures in software when working with nonlinear growth models in a frequentist framework in PROC NL MIXED, a general procedure with few preprogrammed options similar to PROC MCMC. Off of this point, Preacher and Hancock (2015) noted that many studies avoid this essential step in the modeling process by employing overly simple structures “more out of a desire for simplicity, not necessity” (p. 97).

The goal of this article is to facilitate modeling of these residual structures. To accomplish this, first, we briefly overview the mechanics of growth models and show how growth models can be fit in PROC MCMC because the procedure is relatively new and the literature is fairly scant with respect to how one can fit growth models in PROC MCMC. Then, extending the method from Harring and Blozis (2014), we show how commonly employed residual structures beyond the default conditional independence structure can be programmed in PROC MCMC so that researchers can avoid oversimplifying their models when they are estimated with MCMC methods. Ample SAS code and annotation are provided as are example analyses.

Overview of Growth Models

To conceptually explain the idea behind growth models, growth models estimate a mean trajectory for the sample but estimate a unique growth curve for each individual, which is done through the inclusion of random effects. These random effects capture how much the mean trajectory and the individual's trajectory differ and the variance of these random effects is often of interest to determine if curves vary widely from person to person or if curves are more or less the same across people. As with all models, there are also residuals at each measurement occasion, and because repeated measures on the same individual are often serially correlated, it is common to have residuals that are not independently and identically distributed meaning that the covariance structure of the residuals must be explicitly modeled.

Mathematically, the linear growth model for continuous outcomes can be expressed as

$$\mathbf{Y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\varepsilon}_j, \quad (1)$$

where $\mathbf{Y}_j$ is an $m \times 1$ vector of responses for person j where m is the number of measurement occasions, $\mathbf{X}_j$ is an $m \times p$ design matrix for the predictors for person j where p is the number of predictors (which includes the intercept), $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients, $\mathbf{Z}_j$ is an $m \times q$ design matrix for the random effects of person j , $\mathbf{u}_j$ is a $q \times 1$ vector of random effects for person j where q is the number of random effects and $p \geq q$, $E(\mathbf{u}_j) = \mathbf{0}$ and $\text{Cov}(\mathbf{u}_j) = \mathbf{G}$ where $\mathbf{G}$ is $q \times q$, and $\boldsymbol{\varepsilon}_j$ is an $m \times 1$ vector of residuals of the repeated measures for person j where $E(\boldsymbol{\varepsilon}_j) = \mathbf{0}$ and $\text{Cov}(\boldsymbol{\varepsilon}_j) = \mathbf{R}$.

Frequently in applied research studies, the structure for the $\mathbf{R}$ matrix (referred to throughout as the residual structure) is a conditional independence structure such that the variance at each time point is constant, and the covariances between measurement occasions are constrained to 0. In matrix form, this structure is noted as

$$\begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

or more compactly as $\mathbf{R} = (\sigma^2 \mathbf{I})$. However, this assumption is often ill-advised with growth models due to effects of serial correlation and/or measurement error (Davidian & Giltinan, 2003; Harring & Blozis, 2014; Preacher & Hancock, 2015).

Residual Error Structures

The $\mathbf{R}$ matrix is almost never a primary interest of any growth model (e.g., Harring & Blozis, 2014), which stems from the fact that it captures unmodeled variation in the outcome variable. Nonetheless, proper specification of the $\mathbf{R}$ matrix is an explicit assumption of growth modeling and coefficient and standard error estimates throughout the model may be biased when this structure is misspecified (e.g., Ferron, Dailey, & Yi, 2002; Grimm & Widaman, 2010; Kwok, West, & Green, 2007; Litière, Alonso, & Molenberghs, 2008). Therefore, researchers should not ignore this step in the modeling process and should not automatically assume that the common software default of conditional independence is sufficient (e.g., Davidian & Giltinan, 2003; Preacher & Hancock, 2015).

Software greatly facilitates several different residual structures with frequentist methods. For instance, SPSS MIXED has 17 such preprogrammed structures (Peugh & Enders, 2005), SAS PROC MIXED has 37 preprogrammed structures (SAS Institute, 2008), and Stata xtmixed has 8 preprogrammed structures (StataCorp, 2013). Commonly implemented residual structures include conditional independence (the default in most programs), heterogeneous diagonal, compound symmetry, and autoregressive (Blozis & Cudeck, 1999; Cudeck & Harring, 2007), and in PROC MIXED, these structures can be selected from a REPEATED statement such that TYPE = VC, TYPE = UN(1), TYPE = CS, and TYPE = AR(1), respectively. The following subsections will explain the nature of these structures in more detail.

Conditional Independence. The common default conditional independence structure assumes that the variance at each measurement occasion is equal and that residuals at each measurement occasion are not related to one another. In matrix form, this structure can be written as

$$\mathbf{R}_{Cond\ Ind.} = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}. \quad (2)$$

With growth data for younger individuals, this assumption is often tenuous because individuals tend to exhibit larger variances as time progresses. That is, as people become older, there are more unexplained sources of variance (e.g., life experiences) that models are unable to detect.

Heterogeneous Diagonal. As an alternative structure that takes the increasing variance into account, the heterogeneous diagonal structure is a variation of the conditional

independence structure that allows the variance to be different at each measurement occasion but still constrains the off-diagonal terms to 0 such that

$$\mathbf{R}_{Het. Diag.} = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_m^2 \end{bmatrix}. \quad (3)$$

The drawback of this structure is that as measurement occasions increase, several additional parameters must be estimated, which may make the model overly complex (especially with smaller samples) unless the variance at each measurement occasion are quite different. With few measurement occasions though, the heterogeneous diagonal structure requires few additional parameters and can provide much additional information.

Compound Symmetric. In other circumstances, it may not be reasonable to assume that the random effects capture all the covariation between measurement occasions, meaning that some residual covariance remains. The previous two structures do not account for this as the off-diagonal terms are constrained to zero. However, the compound symmetric structure allows for a covariance parameter ρ such that

$$\mathbf{R}_{CS} = \begin{bmatrix} \sigma^2 & \rho & \rho \\ \rho & \ddots & \rho \\ \rho & \rho & \sigma^2 \end{bmatrix}. \quad (4)$$

Each off-diagonal element is equal to ρ —the compound symmetric matrix assumes that the covariance is the same between all measurement occasions (this aids parsimony because only one additional parameter, ρ , is needed compared to the conditional independence structure). The disadvantage of this structure with growth models is that the relation between measurement occasions often decreases as the measurements become further apart meaning that if $Corr(\text{Time 1, Time 2}) = \rho$, then $Corr(\text{Time 1, Time 4})$ is likely less than ρ . Nonetheless, if the error covariances are approximately equal or if measurements are taken close to one another, the compound symmetric structure can maintain a high degree of parsimony.

Autoregressive. For the case where the residual covariances are expected to be different as measurements become more distant, the autoregressive structure may be more appropriate. With this structure, a single parameter for the off-diagonal elements is still estimated, which maintains parsimony. Each successive lag is designated to have a weaker relation such that the covariance between the j th, j' th (where $j \neq j'$) element is defined as $\rho^{|j-j'|}$. With four repeated measures, an autoregressive structure would appear as follows:

$$\mathbf{R}_{CS} = \begin{bmatrix} \sigma^2 & \rho & \rho^2 & \rho^3 \\ \rho & \sigma^2 & \rho & \rho^2 \\ \rho^2 & \rho & \sigma^2 & \rho \\ \rho^3 & \rho^2 & \rho & \sigma^2 \end{bmatrix}. \quad (5)$$

This allows more distant measurements to be less related—for instance, if $\text{Corr}(\text{Time 1, Time 2}) = 0.50$, then $\text{Corr}(\text{Time 1, Time 4})$ would be equal to $0.50^3 = 0.125$ —while also only needing to estimate a single ρ parameter.

Choosing Between Different Structures

Two basic facets can help determine the type of residual structure that may be most appropriate: (1) rank order of people's residuals over time and (2) spread of the residuals over time. Facet 1 relates to whether the residual structure should include off-diagonal elements and Facet 2 relates to whether the diagonal elements should be constrained to the same value. When the rank order of people's residuals over time is thought to be arbitrary, this means that the person with the highest residual at Time 1 is just as likely to have the highest residual at Time 2 as he or she is to have the lowest residual at Time 2—there is no relation between the rank ordering of people's residuals at the various time points. In this case where the rank ordering is thought to be more or less arbitrary, then a structure with off-diagonals constrained to 0 (e.g., conditional independence, heterogeneous diagonal) is appropriate. If the rank ordering is thought to have a nonarbitrary pattern where the subject with the highest residual at Time 1 is more likely to also have the highest residual at Time 2, then a structure with estimated off-diagonal elements (e.g., compound symmetry, autoregressive) is more appropriate.

With regard to Facet 2, if the spread of the residuals is thought to be consistent across time, then a residual structure with constrained diagonal elements (e.g., conditional independence, compound symmetry) is appropriate. If the spread of the residuals is thought to increase or decrease over time, a residual structure with different diagonal elements (heterogeneous diagonal) is more appropriate.

As an illustration of Facet 1, Figure 1 shows residuals for 10 simulated subjects' data drawn from a conditionally independent structure (top panels) and a compound symmetric structure (bottom panels). The left panels show scatterplots for residuals and the two structures look more or less the same. However, the right panels connect the residuals for each subject. In the upper right panel, the rank ordering of subjects' changes at every time point whereas the lower right panel shows lines that are fairly horizontal, indicating that the subjects' rank ordering of the residuals is much less variable.

As an illustration of Facet 2, Figure 2 shows residuals for 50 simulated subjects' drawn for a heterogeneous diagonal structure (left panel) and a conditionally independent structure (right panel). In the right panel, it can be seen that the spread of the residuals is more or less the same at each time point. In the left panel, the spread of the residuals is clearly increasing over time in a classic "fan" pattern. In traditional linear regression analyses, this fan pattern is a cause for concern; however, this type

Figure 1. Comparison of residuals over 4 time points for 10 simulated subjects. The top panels are generated from a conditionally independent residual structure, and the bottom panels are generated from a compound symmetric residual structure. The right panels connect the residuals for the four repeated measures for each subject.

Figure 2. Comparison of residuals over 4 time points for 50 simulated subjects. The left panel is generated from a heterogeneous diagonal residual structure, and the right panel is generated from a conditionally independent residual structure.

of pattern is a nonissue with growth models so long as the appropriate residual structure is selected.

Implementation in SAS

Before presenting details on programming residual structures in PROC MCMC, we will overview a growth model analysis in PROC MCMC that uses the default conditional independence residual structure. There are many Bayesian estimation software programs available, and although SAS is a very popular general statistical package, its Bayesian module is relatively new, and we want to ensure that readers have a basic familiarity with growth models in PROC MCMC before continuing. Additionally, tutorials on PROC MCMC are rather limited and are typically aimed at psychometric modeling (e.g., Ames & Samonte, 2016; although see Zhang, 2016, for some introductory material on fitting growth models in PROC MCMC). We will assume some familiarity with the SAS programming language and basics of MCMC estimation. For a highly readable introduction to MCMC, readers are referred to van de Schoot et al. (2014).

For the remainder of this article, we use data from Willett (1988) that also become a popular demonstration data set after being including in Singer (1998). The Willett (1988) data set contains the scores from 35 individuals from an opposites-naming task at four equally spaced occasions. We removed 3 observations for being potential outliers leaving a total of 32 individuals in the data. A person-level covariate for each person's general cognitive ability is also included (this variable is grand-mean centered in the data). In Raudenbush and Bryk (2002) notation, the growth model using a conditional independence error structure can be written as

$$\begin{aligned}\text{Task Score}_{ij} &= \beta_{0j} + \beta_{1j}(\text{Time}_i) + e_{ij}. \\ \beta_{0j} &= \gamma_{00} + \gamma_{01}(\text{Cognitive}_j) + u_{0j}. \\ \beta_{1j} &= \gamma_{10} + \gamma_{11}(\text{Cognitive}_j) + u_{1j}.\end{aligned}\tag{6}$$

$$\mathbf{u} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} g_{00} & 0 \\ 0 & g_{11} \end{bmatrix}\right) \quad \text{and} \quad \mathbf{e} \sim MVN(\mathbf{0}, \sigma^2 \mathbf{I}_4).$$

$$\text{for } i = 0, 1, 2, 3 \quad \text{and} \quad j = 1, \dots, 32.$$

We chose these data because they are freely available on the UCLA (University of California, Los Angeles) Institute for Digital Research and Education website (www.ats.ucla.edu/stat/mplus/examples/alda/aldamplusch7.htm), it does not have any missing values, and the sample size is small, which can both demonstrate the benefits of MCMC with small samples and allows interested readers to experiment with the code without much computational burden.

Overview of PROC MCMC

Similar to other SAS procedures for growth models, using standard procedures, PROC MCMC requires the data in a “long” format such that the data from a single individual extends over multiple rows. To demonstrate,

ID	Time	Score
I	0	205
I	1	217
I	2	268
I	3	302

Long formatted data contain index variables for time and subject ID and a single variable captures the repeatedly measured outcome variable values. This is in opposition to the “wide” format, which for the same data would look like,

ID	Score 1	Score 2	Score 3	Score 4
I	205	217	268	302

where all data for a particular subject are contained to a single row of the data.

The full code for a growth model with half-Cauchy priors for the variance components, and conditional independent error structure is shown below in its entirety. We proceed by describing what each section of the code commands SAS to do. A fully annotated version of the code can be found in the appendix on the author’s personal webpage (https://sites.google.com/site/danielmmcneish/academic-work/procmmcmc_code).

```
PROC MCMC DATA = OppLong NMC =250000 NBI =25000 THIN =25 SEED =9718
DIAG = (GEWEKE) DIC STATISTICS (PERCENTAGES=(. 5 2.5 50 97.5 99.5));
```

```
PARMS gam00=0 gam10=0 gam01=0 gam11=0;
PARMS sigma2=5 g00=5 g11=5;
```

```
halfCauchy1=pdf('Cauchy', g00, 0, 1000) + pdf('Cauchy', -g00, 0, 1000);
halfCauchy2=pdf('Cauchy', g11, 0, 100) + pdf('Cauchy', -g11, 0, 100);
```

```
PRIOR gam: ~ normal (0, var=1e6);
PRIOR sigma2 ~ uniform (0, 350);
PRIOR g00 ~ general (halfCauchy1);
PRIOR g11 ~ general (halfCauchy2);
```

```
RANDOM int ~ normal ( gam00 , var=g00 ) SUBJECT = ID ;
RANDOM slo ~ normal ( gam10 , var=g11 ) SUBJECT = ID ;
```

```
mu = (int) + (slo) * time + gam01*ccog+gam11*(ccog*time);
```

```
MODEL y~ normal(mu, var=sigma2) ;
```

```
RUN ;
```

PROC MCMC Statement. In the PROC MCMC statement, users specify the basics of their analysis. With PROC MCMC, this is where one notes the number of burn-in iterations (NBI =), the number of MCMC iterations (NMC =), and how many iterations by which to thin (THIN =). Other relevant options include whether to output the DIC (done by simply specifying DIC), whether to output the posterior summaries to a SAS data set (OUTPOST =), which chain convergence criteria to output (DIAG =), which percentiles of the posterior to output [STATISTICS (PERCENTAGE = ())] and setting a seed for the analysis (SEED =). So, for example, the following code would be used to specify 1,000 burn-in iterations, 5,000 recorded iterations, thinning by 10, and to report DIC, Geweke's test of convergence, and the 2.5%, 50%, and 97.5% values of the posterior.

```
PROC MCMC DATA =OppLong NMC=5000 NBI=1000 THIN=10
DIAG=GEWEKE DIC STATISTICS (PERCENTAGE=(2.5 50 97.5)) ;
```

PARMS Statements. Parameters of the model are first listed along with their starting values in PARMS statements.

```
PARMS gam00=0 gam10=0 gam01=0 gam11=0;
PARMS sigma2=5 g00=5 g11=5;
```

Parameters are named based on Equation 6 such that “gam” refers to fixed effects (short for gamma), “sigma2” refers to the residual variance, and “g00” and “g11” refer to the intercept and slope variances, respectively. Names of the parameters are arbitrary provided that they are used consistently. In the above code, notice that there are two separate PARMS statement. This is not completely necessary and all parameters can be placed in a single PARMS statement or each parameter can be given a separate PARMS statement. The potential downside of including multiple parameters in a single PARM statement stems from PROC MCMC's usage of a Metropolis–Hasting algorithm rather than a Gibbs sampler. With Metropolis–Hastings algorithm, parameter updates can be rejected, which results in parameters not necessarily being updated at each iteration. If all parameters are included in a single PARMS statement, all the parameter updates will be collectively rejected or accepted at each iteration, which can be problematic if parameters are on different scales and can result in poor mixing. Conversely, if all parameters are given their own PARMS statement, the computational burden will increase linearly (i.e., five PARMS statements takes five times as long as one PARMS statement) because the parameter update portion of the algorithm repeats for each PARMS statement. Therefore, a common strategy is to block similar parameter together to maximize efficiency of the parameter update portion of the algorithm. Figure 3 shows an example of chain mixing for the intercept variance (g_{00}) using 1 PARMS statement, 2 PARMS statements (as shown in the code above), and unique PARMS statements for each parameter. The top panel reached convergence in 52 seconds, the middle panel in 1 minute 2 seconds (19% longer than with a single PARM statement), and the bottom panel in 1 minute and 17 seconds

Figure 3. Comparison of chain mixing with all parameters in one PARM statement (top), 2 blocked PARMS statements (middle), and unique PARMS statements (bottom).

(24% longer than two PARM statements and 48% longer than one PARM statement), demonstrating that blocking balances chain mixing with computational efficiency.

PRIOR Statements. After giving the program starting values, then prior distributions for each parameter must be given. For regression coefficients, a normal distribution with mean 0 and a large variance is typical in the absence of prior information. There are also a few popular choices for the error variance prior including an inverse gamma distribution with small hyperparameters¹ (e.g., 0.01 or 0.001) or a uniform

distribution with a fairly large range, considering the scale of the outcome (we will use the latter).

```
prior gam:~normal(0,var=1e6);
prior sigma2~uniform(1e-5,350);
```

Note that the “.” is a SAS wildcard character. For instance, “gam:” executes the line of code for any parameter beginning with “gam” (trailing characters are not restricted and can be letters, numbers, or symbols). The priors can also be written out separately without the wildcard operator as well, so this is simply a programmatic shortcut.

With most growth models, there are two random effects (intercept and [linear] slope). If a covariance between these two random effects is desired, a 2×2 covariance matrix for these random effects can be specified either with a multivariate prior (e.g., an inverse Wishart) or with separate priors for the each random effect variance and the covariance between them (if necessary). In accordance with recommendations in Liu, Zhang, and Grimm (2016), we will use the latter approach (code for the inverse Wishart specification is also provided in the online appendix).

The half-Cauchy distribution recommended by Gelman (2006) is not preprogrammed in SAS but can easily be programmed using a GENERAL distribution along with a PDF command that offers the Cauchy distribution. The half-Cauchy distribution is equal to a Cauchy distribution reflected over the input value $x = 0$. The SAS code for this manipulation is

```
halfCauchy1=pdf('Cauchy',g00,0,1000) + pdf('Cauchy',-g00,0,1000);
halfCauchy2=pdf('Cauchy',g11,0,100) + pdf('Cauchy',-g11,0,100);
```

where `halfCauchy1` is the prior for the intercept variance and `halfCauchy2` is the prior for the slope variance. The first argument of the PDF function is the distribution type (Cauchy), the second argument is the input variable (either the intercept variance or the slope variance), the third argument is the location parameter about which the distribution is centered (this should be zero for variance components), and the fourth argument is the scale parameter that may be thought of similarly to the variance of a t distribution (this value will change depending on the scale of the outcome variable). The fact that the random intercepts are expected to vary more than the random slopes is reflected by the larger scale parameter in `halfCauchy1`. The code after the addition sign is needed to reflect the distribution over $x = 0$. Once the half-Cauchy distribution is programmed, then the priors are set by using the GENERAL distribution,

```
prior g00~general(halfCauchy1);
prior g11~general(halfCauchy2);
```

RANDOM Statement. Similar to PROC MIXED, PROC MCMC allows users to specify random effects that is also done through a RANDOM statement. This is done with the following statements:

```
RANDOM int~normal(gam00,var=g00) SUBJECT=ID;
RANDOM slo~normal(gam10,var=g11) SUBJECT=ID;
```

where “int” is the total intercept for the j th person (fixed effect plus random effect) and “slo” is the total slope for the j th person (fixed effect plus random effect) and “ID” is a variable in the data that uniquely identifies each individual.

MODEL Statement. Next, the mean trajectory is specified for the whole sample. This is rather simple and for the Willett data is done with the following code:

```
mu=(int)+(slo)*time+ gam01*ccog+gam11*(ccog*time)
```

Because mu is a linear combination of parameters in the model, it is not included in a PARMS statement. With the mean trajectory, random effects, and error variance specified, the final line of code needed is

```
MODEL y~normal(mu,var=sigma2)
```

which informs SAS that the outcome variable is y and that it is normally distributed with mean mu and (independently and identically distributed error) variance sigma2. Despite the fact that the MODEL statement does not appear to capture the entire model, the preceding statements piece the different components of the model together. The RANDOM statements along with specifying mu and the MODEL statement all together reproduce the full model Equation 6 in PROC MCMC.

PROC MCMC Output

The output tables from a PROC MCMC analysis differ from a typical growth model in SAS, so we will briefly present the appearance of these tables and the information contained within them.

Posterior Summaries. The posterior summaries table in the PROC MCMC output shows the mean and standard deviation of the posterior distribution for each parameter in the model, the number of recorded iterations, and the requested percentiles of the posterior distribution. Figure 4 shows a screenshot from the Willett (1988) growth model with a conditionally independent error structure. The parameters are listed in the order that they appear in the PARMS statement. The values requested in the PERCENTAGES= command of the PROC MCMC statement can be used to determine if the effects are nonnull, similar to how confidence intervals are used in frequentist analyses.

Trace Plots, Autocorrelation, and the Posterior Distribution. For each parameter in the model, PROC MCMC shows a trace plot, autocorrelation plot, and posterior distribution. Figure 5 shows an example of these plots for the intercept variance from the Willett (1988) model. The trace plot seems to show that the chain has converged as the values in the latter half appear to random samples from the target posterior distribution. The autocorrelation plot in the lower left corner shows the correlation between values at each lag (after thinning). In Figure 5, the plot shows that there is

Posterior Summaries								
Parameter	N	Mean	Standard Deviation	Percentiles				
				0.5%	2.5%	50%	97.5%	99.5%
gam01	10000	-0.1494	0.4270	-1.3327	-0.9743	-0.1539	0.6968	0.9962
gam11	10000	0.4486	0.1618	0.0249	0.1273	0.4489	0.7647	0.8853
gam00	10000	171.4	5.2854	157.1	161.1	171.4	181.9	185.7
gam10	10000	25.6776	2.0841	20.1431	21.4450	25.6756	29.8101	31.1653
sigma2	10000	179.3	33.6974	113.5	125.5	175.2	257.5	289.4
g00	10000	798.5	269.2	357.4	417.1	753.7	1439.2	1802.0
g11	10000	104.4	41.0553	34.3782	45.2737	97.4318	204.3	257.5

Figure 4. Posterior summaries table from PROC MCMC output.

very little correlation between iterations. The plot in the lower right corner shows the posterior distribution of the replications. This is the distribution from which point estimates and credible intervals are taken.

Geweke’s Statistics. In addition to the trace plots that are shown in Figure 3, Geweke’s statistic is a common inferential method to determine whether the MCMC chain has converged. Geweke’s test compares the mean from the first 10% of the chain to the mean of the last 50% of the chain with a *z* test for each parameter in the model. If these means are not significantly different, then conclusion is that the chain converged in the first 10% of the chain and all subsequent iterations are drawn from the target distribution. Figure 6 shows a screen shot of the Geweke’s statistics from the model discussed in the previous subsections.

Building the Likelihood and the General Distribution

Using preprogrammed distributions in the MODEL statement in PROC MCMC for linear models restricts users to the conditional independence structure by only allowing the VAR = option to take a single argument. This shortcoming is not unique to PROC MCMC as Harring and Blozis (2014) have noted this same restriction in PROC NL MIXED, another highly flexible computing environment that uses maximum likelihood rather than MCMC. Also similar to PROC NL MIXED, PROC MCMC allows users to specify a model through a user-defined likelihood (or log-likelihood) function using the GENERAL distribution in the MODEL statement of PROC MCMC. It is through this option that we will demonstrate how alternative error structures can be specified.

Because this will require specifying the log-likelihood for the model, we will briefly discuss the log-likelihood function for linear growth models. The MODEL

Figure 5. Trace plot, autocorrelation plot, and posterior distribution for intercept variance parameter.

Geweke Diagnostics		
Parameter	z	Pr > z
gam01	-1.4213	0.1552
gam11	-0.4239	0.6717
gam00	-1.6265	0.1039
gam10	2.0404	0.0413
sigma2	-0.5403	0.5890
g00	0.8076	0.4193
g11	0.9228	0.3561

Figure 6. Geweke’s statistic table from PROC MCMC output.

statement in SAS is conditional on the random effect parameters (Harring & Blozis, 2014); the likelihood can therefore be written as follows:

$$\ln[\ell(\mathbf{R}|\mathbf{b}, \mathbf{G})] = -\frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln|\mathbf{R}| - \frac{1}{2} \left\{ \sum_{i=1}^M (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{R}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}) \right\}. \quad (7)$$

This likelihood is composed of three components—the constant term ($-\frac{M}{2} \ln 2\pi$), the log of the determinant of the residual covariance matrix, $\frac{1}{2} \ln|\mathbf{R}|$, and the Mahalanobis' distance portion, $-\frac{1}{2} \left\{ \sum_{i=1}^M (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{R}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}) \right\}$.

As outlined in Harring and Blozis (2014), common structures for $\mathbf{R}$ can be rewritten as a function of additional parameters to facilitate programming. Namely,

$$\mathbf{R} = \boldsymbol{\Omega}^{1/2}(\boldsymbol{\omega}) \mathbf{P}(\boldsymbol{\rho}) \boldsymbol{\Omega}^{1/2}(\boldsymbol{\omega}), \quad (8)$$

where $\boldsymbol{\Omega}$ is a scalar or an $m \times m$ diagonal matrix dependent on a set of parameters $\boldsymbol{\omega}$ and $\mathbf{P}$ is an $m \times m$ correlation matrix dependent on parameters $\boldsymbol{\rho}$, and m is the number of measurements per individual.

The following subsections will outline the parameterizations necessary to specify a conditionally independent, heterogeneous diagonal, compound symmetric, and autoregressive structure.

Conditionally Independent. Although this structure can be specified in SAS exclusively using preprogrammed statements, we will begin with this structure to ease readers into how to build the likelihood manually and how to use the GENERAL distribution to estimate the model. With this structure, $\boldsymbol{\Omega}$ and $\mathbf{P}$ take fairly simple forms such that $\boldsymbol{\Omega} = \sigma^2 \mathbf{I}_m$ and $\mathbf{P} = \mathbf{I}_m$.

In SAS, all the previous code from the PARMS statements to the RANDOM statement can be recycled. When writing out the likelihood manually, it will be much easier to consider data in the “wide” format where each row consists of all the measurements for a single person (as is required in structural equation modeling software) rather than the “long” format (as required by PROC MIXED). The reason for this change is because the likelihood contains a summation sign, and it is simpler to write a looping function over a fixed set of columns in a data set (variables) than over a fluctuating set of rows (people). Therefore, we will add three ARRAY statements at the beginning of the code—one for the outcome variable, one for the time variable, and one for the residual at each measurement occasion. With the Willet (1988) data that have four measurement occasions, these statements are written as

```
array y[4] y1-y4;
array x[4] (0 1 2 3);
array e[4];
```

(0 1 2 3) appears after the x array because these represent the time variable for measurement occasions—parentheses indicate elements are fixed to particular values. y1-

y4 appear after the y array to indicate that the elements are equal to the variables in the data set.

Next, consider the three components of the likelihood function that need to be built—the constant term ($-\frac{M}{2}\ln 2\pi$), the log of the determinant of the error covariance matrix, $\frac{1}{2}\ln|\mathbf{R}(\omega, \rho)|$, and the Mahalanobis' distance portion, $-\frac{1}{2}\left\{\sum_{i=1}^M (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})^T \mathbf{R}_i(\omega, \rho)^{-1} (\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})\right\}$. First, note that each portion is preceded by $-\frac{1}{2}$, which can be factored out. The constant term is also easy to program in SAS: `ln2pi=1.837877`. The second component is also straightforward to compute with the conditional independence structure—the determinant of a diagonal matrix is equal to the product of the diagonal entries. For the conditional independence structure, $|\mathbf{R}|=(\sigma^2)^m$, making the log of the determinant simply equal to $\log|\mathbf{R}|=m\sigma^2$ or `lndet=m*log(errvar)`. The third component is slightly more complex to program. First, the mean trajectory must be programmed, which changes as a function of time. The residuals $(\mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta})$ must then be calculated, squared, divided by the single error variance, and summed. Altogether for the Willett (1988) data, this can be input into SAS as

```
w1=0;m=4;
do j=1 to m;
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
e[j]=y[j]-mu;
w1=w1+e[j]*e[j];end;

ln2pi=1.837877;
lndet=m*log(sigma2);
mah=w1/sigma2;
```

Once all three components are programmed, then the log-likelihood can be written as an addition of the three components, `llike=-.5*(m*ln2pi+lndet+mah)` and the model can be specified with the GENERAL distribution, `MODEL general(llike)`. The full annotated SAS script for this example is included in the appendix on the author's personal webpage.

Heterogeneous Diagonal Structure. Fitting the heterogeneous diagonal structure differs from the conditional independence structure in that each diagonal element of the $\mathbf{R}$ matrix is freely estimated, which changes how some of the components of the likelihood function are formed. The constant term is unaffected, but the log determinant will be affected because it will no longer be equal to $m\sigma^2$. Instead, $\log|\mathbf{R}|=\sigma_1^2 \times \sigma_2^2 \times \dots \times \sigma_m^2$ —this is still the product of the diagonal elements except now each diagonal element is uniquely estimated. The $\mathbf{R}_i(\omega, \rho)^{-1}$ term in the Mahalanobis' distance term will also change as the index i changes from 1 to M .

In SAS, the heterogeneous diagonal structure requires an additional array (array sigma[4]) compared to the conditional independent structure now that each repeated measure requires a unique residual variance estimate.

Each element of the new “sigma” array also needs its own prior—we will use the same prior that was used for the error variance except that we will make use of the “:” wildcard operator again to apply the prior to multiple elements, prior sigma:~uniform(1e-5, 350).

The changes to the determinant and Mahalanobis’ distance components of the likelihood are made by including sigma[j] into the looping code to actively multiply the diagonal elements of the matrix and also dividing the w term by sigma[j] to reflect the changing inverse term in the Mahalanobis distance formula. Specifically,

```
w1=0;dt=1;m=4;
```

```
do j=1 to m;
```

```
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
```

```
e[j]=y[j]-mu;
```

```
w1=w1+((e[j]*e[j])/sigma[j]);
```

```
dt=dt*sigma[j];end;
```

```
ln2pi=1.837877;
```

```
lndet=log(dt);
```

```
mah=w1;
```

The likelihood is then composed of the same three components, and the model is specified with the GENERAL distribution,

```
llike=-.5*(m*ln2pi+lndet+mah);
```

```
MODEL general(llike);.
```

Compound Symmetric Structure. When off-diagonal terms are introduced, the determinant component and the inverse in the Mahalanobis’ distance component become more complex. Using the notation in Harring and Blozis (2014), with the compound symmetric structure $\Omega = \sigma^2 \mathbf{I}_m$ and $\mathbf{P} = (1 - \rho) \mathbf{I}_m + \rho \mathbf{J}_m$, where $\mathbf{J}$ is a matrix of ones. The inverse of $\mathbf{R}$ is equal to the product of Ω^{-1} and $\mathbf{P}^{-1}$. From Rao (1973), $\mathbf{P}^{-1} = (a - b) \mathbf{I}_m + b \mathbf{J}_m$ where $a = \frac{1 + \rho(m-2)}{(1-\rho)[1 + (m-1)\rho]}$ and $\frac{\rho}{(1-\rho)[1 + (m-1)\rho]}$. Regarding the determinants, Sutradhar and Kumar (2003) showed that $|\mathbf{R}_{CS}| = |\Omega| |\mathbf{P}|$, which can more concretely be expressed as $|\Omega| |\mathbf{P}| = (\sigma^2)^m (1 - \rho)^m (1 + \rho(m - 1))$.

Although the addition of the off-diagonal term appears complex, given that the formulas have been derived, these quantities are fairly simple to program in SAS. The “sigma” array is no longer needed because the compound symmetric structure assumes equal variances at each repeated measure. Instead, a “p” array is used that is equal to the value of the time points plus one, array p [4] (1 2 3 4). To elucidate, if the time points were 0, 2, 4, and 6 instead of 0, 1, 2, and 3, then the p array would be

(1 3 5 7). The PARMS statement and prior statement are also needed for the ρ parameter that was not present with the previous structures. After this information is included, then the likelihood can be built. First, the constant, values of a and b , and m w terms are programmed,

```
m=4;
na=1+((m-2)*rho);
nb=-rho;
denom=(1-rho)*(1+(m-1)*rho);
nad=na/denom;
ndb=nb/denom;
w1=0;w2=0;w3=0;w4=0;
```

Then, m separate loops are created to compute the Mahalanobis' distance component,

```
do j=1 to m;
q=p[j];
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
e[j]=y[j]-mu;
w1=w1+e[j]*e[j];end;
```

```
do j=2 to m;
q2=p[j];q1=p[j-1];
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
e[j]=y[j]-mu;
if (q2-q1)=1 then
w2=w2+e[j]*e[j-1];
else w2=w2+0;end;
```

```
do j=2 to m;
q2=p[j];q1=p[j-1];
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
e[j]=y[j]-mu;
if (q2-q1)=2 then
w3=w3+e[j]*e[j-1];
else w3=w3+0;end;
```

```
do j=2 to m;
q2=p[j];q1=p[j-1];
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
e[j]=y[j]-mu;
if (q2-q1)=3 then
w4=w4+e[j]*e[j-1];
else w4=w4+0;end;
```

```
ln2pi=1.837877;
mah=( (2*ndb*(sum(of w2-w4))+(nad*w1))/sigma2);
lndet=m*log(sigma2)+m*log(1-rho)+log(1+(m-1)*rho);
```

Once the constant, determinant, and Mahalanobis' distance component are programmed, then the likelihood can be built, `llike=-.5*(m*ln2pi+lndet+mah)`, and the model can be run using the GENERAL distribution, `MODEL general(llike)`.

Autoregressive Structure. Similar to the compound symmetric structure, the autoregressive structure also features nonzero off-diagonal elements that makes computation of the determinant and the inverse somewhat complex. Similar to the compound symmetric structure, Graybill (1983) showed that $\mathbf{R}^{-1} = (\mathbf{\Omega}^{-1})(\mathbf{P}^{-1})$. Specifically, as noted in Harring and Blozis (2014), Grenander and Szegö (1958) showed that $\mathbf{P}^{-1}$ is a symmetric matrix with half-bandwidth equal to 1 (aka, tridiagonal) meaning that all nonzero entries are confined to the main diagonal, the first superdiagonal, and the first subdiagonal. More specifically, $\mathbf{P}^{-1}$ is equal to

$$\frac{1}{(1-\rho^2)} \begin{bmatrix} 1 & -\rho & 0 & \cdots & 0 & 0 \\ -\rho & 1+\rho^2 & -\rho & \cdots & 0 & 0 \\ 0 & -\rho & 1+\rho^2 & \cdots & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & -\rho \\ 0 & 0 & 0 & \cdots & -\rho & 1 \end{bmatrix}$$

the main diagonal with the exception of the upper-leftmost and lower-rightmost elements are equal to $(1+\rho^2)/(1-\rho^2)$ and the first subdiagonal and first superdiagonal is equal to $-\rho/(1-\rho^2)$. The determinant of a matrix with an autoregressive structure is fairly straightforward and Graybill (1983) showed it to be $|\mathbf{R}_{AR}| = (\sigma^2)^m(1-\rho^2)^{m-1}$.

To implement this structure in SAS, first the three components of $\mathbf{P}^{-1}$ must be coded—the multiplication term in front of the matrix (`ka`), the superdiagonal and subdiagonal term (`kb`), and the main diagonal term (`kc`). This is done with the following code:

```
m=4;
rho2=rho*rho;
ka=1/(1-rho2);
kb=-ka*rho;
kc=ka*(1+rho2);
```

The residual variance at the first measurement occasion is first calculated by itself,

```
w1=0; w2=0; w3=0; q=p[1];
mu=b1+b2*x[1] +b3*ccog+b4*(ccog*x[1]);
e[1]=y[1]-mu;
```

and then a loop is created for the residual sum of squares for the first lag which is used to estimate ρ .

```
do j=2 to m;
mu=(int)+(slo)*x[j]+gam01*ccog+gam11*(ccog*x[j]);
e[j]=y[j]-mu;
w2=w2+e[j]*e[j-1];
w3=w3+e[j]*e[j];end;

w1=(e[1]*e[1])+(e[m]*e[m]);
w3=w3-(e[m]*e[m]);
```

No additional loops are needed for any lag greater than 1 because the autoregressive error structure calculates these by $\rho^{|j-j'|}$. Then, the components of the log-likelihood are built

```
ln2pi=1.837877;
lndet=m*log(sigma2)+(m-1)*log(1-rho2);
mah=(ka*w1+2*kb*w2+kc*w3)/sigma2;
llike=-.5*(m*ln2pi+lndet+mah);
```

and then the model is estimated with, MODEL general(llike).

Comparison Using Willett (1988) Data

Using the same Willett (1988) data, we will compare the estimates from SAS' frequentist procedure PROC MIXED to PROC MCMC for the aforementioned error structures. PROC MIXED was used because it features restricted maximum likelihood (REML) estimation, which is known to be more appropriate for smaller samples compared to full maximum likelihood that would be used if the growth model were fit in another software program such as *Mplus*. Code for these models is provided in the appendix on the author's personal webpage. A spaghetti plot for all 32 individuals is shown in Figure 7 and model results are shown in Table 1. For the MCMC models, 25,000 burn-in iterations were used and 250,000 Monte Carlo replications were conducted that were then thinned by 25 to reduce potential autocorrelation.² With these settings, the Geweke statistics for each parameter were not significant at the 0.01 level, providing evidence that the number of iterations was sufficient for the MCMC chain to reach convergence. The total variance of the outcome variable at each time point was between 834 and 1,456 and prior distributions for the elements of the covariance structure were selected accordingly. Specifically, the scale of the half-Cauchy prior for the intercept was set to 1,000, the scale of the half-Cauchy prior for the slope was set to 100, and priors for error variances were set to be uniform over the range [1E-5, 350] in order to be reasonably noninformative while also being plausible based on the scale of the data. Of course, with such a small sample, the prior distributions

Figure 7. Spaghetti plot for opposites-naming score across time for 32 individuals.

will noticeably affect the posterior distribution, so using different hyperparameter values will certainly affect the posterior results to some degree (McNeish, 2016a).

From Table 1, the parameters of the mean trajectory are nearly the same between the MCMC and frequentist estimates across each of the error structures. However, note that the estimates of the covariance structure (random effect variances and error variances) are noticeably larger in the MCMC models. Studies investigating these types of models with small samples have consistently noted that these parameters are most at risk to be biased when estimated with frequentist methods, even with state-of-the-art corrections (e.g., Ferron, Bell, Hess, Rendina-Gobioff, & Hibbard, 2009; Maas & Hox, 2005; McNeish, 2016b). However, studies have noted that MCMC methods provide better estimates with small sample sizes if priors are carefully chosen (e.g., van de Schoot et al., 2015). McNeish and Stapleton (2016) showed that the half-Cauchy prior performed well for these parameters with a sample size as small as 8 and routinely outperformed REML estimation (which tends to be downwardly biased with smaller samples). This pattern is quite apparent with this data and highlights a potential advantage of reverting to MCMC methods for growth models with small samples, even if the model can be equivalently fit within a frequentist context. Also note that the deviance information criterion (DIC) and Bayesian information criterion (BIC) would select different residual structures if one strictly adheres to this criteria—the DIC is the lowest for the heterogeneous diagonal structure, while the BIC is the lowest for the conditional independence structure.

Table 1. Comparison of Estimates in PROC MCMC and PROC MIXED for Various Error Structures.

	Conditional independence		Heterogeneous diagonal		Compound symmetry		Autoregressive	
	MCMC	MIXED	MCMC	MIXED	MCMC	MIXED	MCMC	MIXED
Mean structure								
Intercept	171.4 (5.29)	171.4 (4.99)	171.4 (5.45)	171.4 (5.02)	171.4 (5.38)	171.4 (4.99)	171.4 (5.35)	171.4 (4.96)
Time	25.68** (2.08)	25.68** (1.95)	25.64** (2.13)	25.62** (1.93)	25.67** (2.07)	25.67** (1.95)	25.70** (2.11)	25.71** (1.96)
Cognitive	-0.15 (.43)	-0.16 (.39)	-0.19 (.42)	-0.18 (.39)	-0.16 (.40)	-0.16 (.39)	-0.15 (.41)	-0.14 (.39)
Time × Cognitive	0.45** (.16)	0.44** (.14)	0.45** (.16)	0.45** (.15)	0.44** (.16)	0.44** (.15)	0.44** (.16)	.44** (.15)
Covariance structure								
Intercept variance	753.7	678.1	775.3	667.5	705.7	544.8	651.3	590.8
Time variance	97.4	87.6	96.9	76.2	100.6	87.6	85.6	73.9
σ^2	175.2	168.8	—	—	174.0	168.8	259.0	233.6
ρ	—	—	—	—	0.123	0.068	0.253	0.258
σ_1^2	—	—	217.6	223.9	—	—	—	—
σ_2^2	—	—	170.2	146.8	—	—	—	—
σ_3^2	—	—	174.4	145.3	—	—	—	—
σ_4^2	—	—	209.8	226.9	—	—	—	—
DIC/BIC	1076.7	1154.0	1069.2	1163.9	1074.9	1281.1	1098.09	1157.2

Note. MCMC = Markov Chain Monte Carlo; DIC = deviance information criterion; BIC = Bayesian information criterion. MCMC estimates were based on the median of the posterior distribution as prescribed for the half-Cauchy prior by Gelman (2006). Standard errors/standard deviations of the posterior are included in parenthesis for mean structure parameters.

** $p < .01$ or 0 not included in the 99% credible interval.

Concluding Remarks

Despite noted advantages of MCMC methods, specifying Bayesian growth models in software with appropriate residual structures is no small task (the user-friendly *Mplus* included) and presents an obstacle for applied researchers to effectively model their data. The corresponding *Mplus* code for the residual structures fit in this article are provided in the appendix on the author's personal webpage should readers be interested in fitting alternative structures in *Mplus*. When comparing PROC MCMC and *Mplus* code in the appendix, the *Mplus* code is notably simplified. However, keep in mind that Bayesian analysis in *Mplus* are limited by the options *Mplus* offers (e.g., limited choice of prior distributions, some types of model constraints needed for residual structures not permissible with BAYES estimation option) that can limit the utility of *Mplus* for some types of analysis (e.g., small samples where one needs more flexible prior distributions).

Although not discussed in the preceding sections, missing data can also be accommodated easily within PROC MCMC. Beginning in Version 9.4, SAS can accommodate missing at random data with fully Bayesian estimation. This method is conceptually similar the full information maximum likelihood method for frequentist models in that it does not require researchers to specify an imputation model or pool estimates across multiple imputed data sets. Instead, each missing value becomes an extra parameter in the model that the MCMC algorithm updates at each iteration. PROC MCMC will use the same model for the outcome for missing data by default although researchers can specify a different model for missing values as well. This procedure is the default in new versions (SAS/STAT 12.1 user's guide; SAS Institute, 2012, p. 11). For more information on the implementation of this procedure in SAS, readers are referred to pages 4518 to 4521 of SAS/STAT 12.1 user's guide.

As shown in this article, with some programming, models with the desired residual structure can be fit PROC MCMC, and we hope that researchers may more easily be able to capitalize on MCMC methods when sample sizes are diminished, when certain software cannot accommodate the desired model, or when otherwise advantageous.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Notes

1. Hyperparameters are the arguments of a prior distribution. If a prior is set to be $N(0, 1)$, then 0 and 1 are the hyperparameters for that particular prior. Hyperparameters need not be constants and can themselves have prior distributions. The prior distribution of a

hyperparameters is called a *hyperprior*. As an example, a prior distribution could be $N(\alpha, \beta)$, where $\alpha \sim N(0, 10)$ and $\beta \sim \Gamma^{-1}(.01, .01)$. α and β would be the hyperparameters, $N(0,10)$ and $\Gamma^{-1}(.01, .01)$ would be the hyperpriors. In this case, the hyperparameters are unknown and will be sampled from a normal and inverse gamma distributions, respectively.

2. The number of iterations may seem rather high, but SAS uses Metropolis–Hastings, which is typically slower to converge than the Gibbs sampler commonly implemented in other programs.

References

- Ames, A. J., & Samonte, K. (2015). Using SAS PROC MCMC for item response theory models. *Educational and Psychological Measurement*, 75, 585-609. doi: 10.1177/0013164414551411
- Blozis, S. A., & Cudeck, R. (1999). Conditionally linear mixed-effects models with latent variable covariates. *Journal of Educational and Behavioral Statistics*, 24, 245-270.
- Cudeck, R., & Harring, J. R. (2007). Analysis of nonlinear patterns of change with random coefficient models. *Annual Review of Psychology*, 58, 615-637.
- Davidian, M., & Giltinan, D. M. (2003). Nonlinear models for repeated measurement data: An overview and update. *Journal of Agricultural, Biological, and Environmental Statistics*, 8, 387-419.
- Ferron, J. M., Bell, B. A., Hess, M. R., Rendina-Gobioff, G., & Hibbard, S. T. (2009). Making treatment effect inferences from multiple-baseline data: The utility of multilevel modeling approaches. *Behavior Research Methods*, 41, 372-384.
- Ferron, J., Dailey, R., & Yi, Q. (2002). Effects of misspecifying the first-level error structure in two-level models of change. *Multivariate Behavioral Research*, 37, 379-403.
- Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1, 515-534.
- Graybill, F. A. (1983). *Matrices with applications in statistics*. Belmont, CA: Wadsworth.
- Grenander, U., & Szegő, G. (1958). *Toeplitz forms and their applications*. Los Angeles: University of California Press.
- Grimm, K. J., & Widaman, K. F. (2010). Residual structures in latent growth curve modeling. *Structural Equation Modeling*, 17, 424-442.
- Harring, J. R., & Blozis, S. A. (2014). Fitting correlated residual error structures in nonlinear mixed-effects models using SAS PROC NLMIXED. *Behavior Research Methods*, 46, 372-384.
- Hox, J., van de Schoot, R., & Matthijsse, S. (2012). How few countries will do? Comparative survey analysis from a Bayesian perspective. *Survey Research Methods*, 6, 87-93.
- Kwok, O. M., West, S. G., & Green, S. B. (2007). The impact of misspecifying the within-subject covariance structure in multiwave longitudinal multilevel models: A Monte Carlo study. *Multivariate Behavioral Research*, 42, 557-592.
- Litière, S., Alonso, A., & Molenberghs, G. (2008). The impact of a misspecified random-effects distribution on the estimation and the performance of inferential procedures in generalized linear mixed models. *Statistics in Medicine*, 27, 3125-3144.
- Liu, H., Zhang, Z., & Grimm, K. J. (2016). Comparison of inverse Wishart and separation-strategy priors for Bayesian estimation of covariance parameter matrix in growth curve analysis. *Structural Equation Modeling*, 23, 354-367.

- Maas, C. J., & Hox, J. J. (2005). Sufficient sample sizes for multilevel modeling. *Methodology*, 1, 86-92.
- McNeish, D. (2016a). On using Bayesian methods to address small sample problems. *Structural Equation Modeling*. Advance online publication. doi:10.1080/10705511.2016.1186549
- McNeish, D., & Stapleton, L. M. (2016). Modeling clustered data with few clusters. *Multivariate Behavioral Research*. Advance online publication. doi:10.1080/00273171.2016.11670
- McNeish, D. M. (2016b). Using data-dependent priors to mitigate small sample bias in latent growth models a discussion and illustration using Mplus. *Journal of Educational and Behavioral Statistics*, 41, 27-56. doi:10.3102/1076998615621299
- Meuleman, B., & Billiet, J. (2009). A Monte Carlo sample size study: How many countries are needed for accurate multilevel SEM? *Survey Research Methods*, 3, 45-58.
- Muthén, B., & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological Methods*, 17, 313-335.
- Peugh, J. L., & Enders, C. K. (2005). Using the SPSS mixed procedure to fit cross-sectional and longitudinal multilevel models. *Educational and Psychological Measurement*, 65, 717-741.
- Preacher, K. J., & Hancock, G. R. (2015). Meaningful aspects of change as novel random coefficients: A general method for reparameterizing longitudinal models. *Psychological Methods*, 20, 84-101.
- Rao, C.R. (1973). *Linear statistical inference and its applications*. New York: Wiley.
- Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods*. Thousand Oaks, CA: Sage.
- SAS Institute. (2008). *SAS/STAT 9.2 user's guide*. Cary, NC: Author.
- SAS Institute. (2012). *SAS/STAT 12.1 user's guide*. Cary, NC: Author.
- Singer, J. D. (1998). Using SAS PROC MIXED to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioral Statistics*, 23, 323-355.
- StataCorp. (2013). *Stata 13 base reference manual*. College Station, TX: Stata Press.
- Sutradhar, B. C., & Kumar, P. (2003). The inversion of correlation matrix for MA (1) process. *Applied Mathematics Letters*, 16, 317-321.
- van de Schoot, R. (2016, January). *25 years of Bayes in psychology*. Paper presented at the 7th Mplus Users' Meeting, Utrecht, Netherlands. Retrieved from <http://mplus.fss.uu.nl/wp-content/uploads/sites/24/2012/07/opening-review-short.pptx>
- van de Schoot, R., Broere, J. J., Perryck, K. H., Zondervan-Zwijnenburg, M., & Van Loey, N. E. (2015). Analyzing small data sets using Bayesian estimation: the case of posttraumatic stress symptoms following mechanical ventilation in burn survivors. *European Journal of Psychotraumatology*, 6, 25216.
- van de Schoot, R., Kaplan, D., Denissen, J., Asendorpf, J. B., Neyer, F. J., & Aken, M. A. (2014). A gentle introduction to Bayesian analysis: Applications to developmental research. *Child Development*, 85, 842-860.
- Walker, L. J., Gustafson, P., & Frimer, J. A. (2007). The application of Bayesian analysis to issues in developmental research. *International Journal of Behavioral Development*, 31, 366-373.
- Willett, J. B. (1988). Questions and answers in the measurement of change. *Review of Research in Education*, 15, 345-422.
- Zhang, Z. (2016). Modeling error distributions of growth curve models through Bayesian methods. *Behavior Research Methods*, 48, 427-444.
- Zhang, Z., Hamagami, F., Wang, L., Grimm, K. J., & Nesselroade, J. R. (2007). Bayesian analysis of longitudinal data using growth curve models. *International Journal of Behavioral Development*, 31, 374-383.