

A flexible Bayesian approach to modeling change over time and across individuals

Timothy Ballard^a, Hector Palada^a, Mark Griffin^b, & Andrew Neal^a

^aThe University of Queensland

^bThe University of Western Australia

Abstract

To do

There has been an increasing emphasis within organizational psychology and organizational behavior on the importance of examining how processes unfold over time (Kozlowski, Chao, Grand, Braun, & Kuljanin, 2013; Lord, Diefendorff, Schmidt, & Hall, 2010; Neal, Ballard, & Vancouver, 2017; M. Wang, Zhou, & Zhang, 2016). This emphasis is born from the observation that organizational phenomena are often dynamic in nature, meaning that they evolve over time and are characterized by continual change. Whether one is considering how performance improves with practice (e.g., Yeo & Neal, 2004), how self-regulatory processes change in response to stressors (e.g., Zhou et al., 2017), or how teams develop shared knowledge (e.g., Grand, Braun, Kuljanin, Kozlowski, & Chao, 2016), many of the questions of interest require explicit consideration of temporal dynamics.

In recognition of this, recent theoretical developments in many areas have focused more strongly on understanding how processes play out over time (e.g., Ballard, Yeo, Vancouver, & Neal, 2017; Fitzsimons, Finkel, & VanDellen, 2015; Morgeson, Mitchell, & Liu, 2015; Rousseau, Hansen, & Tomprou, 2018; Vancouver & Purl, 2017). Researchers have turned to within-subjects, longitudinal designs to test these theories. However, longitudinal designs, whilst necessary, are not sufficient to test dynamic theory. It is not sufficient to simply collect multiple waves of data and compare the levels of outcomes variables across waves using conventional statistics. To make inferences about a dynamic process, the researcher must have a model that explicitly describes the rules that govern how the system evolves from one moment to the next (Taylor et al., 2017; M. Wang et al., 2016). Fortunately, sophisticated approaches to modeling longitudinal data have been developed that can help to answer relatively complex questions regarding dynamic processes (e.g., McArdle, 2009). Of these, the latent change score model (LCSM) is often regarded as the current state of the art (e.g., Howardson, Karim, & Horn, 2017; M. Wang et al., 2017, 2016).

Despite the usefulness of longitudinal models such as the LCSM, there are several limitations to the way these models have typically been implemented. First, these models are typically implemented within a frequentist framework, which means they forgo the advantages of Bayesian analysis such as the ability to deliver evidence for a null hypothesis, quantify uncertainty in a parameter, to incorporate prior beliefs into the analysis, or perhaps most importantly, its status as the mathematically optimal method for scientific inference (see Morey, Hoekstra, Rouder, Lee, & Wagenmakers, 2015; Rouder, Speckman, Sun, Morey, & Iverson, 2009; Vandekerckhove, Rouder, &

Kruschke, 2018; Wagenmakers, Morey, & Lee, 2016). Second, these models are typically implemented as single-level models, which assume participants are independent. This approach prevents the researcher from being able to make inferences about the population from which participants are drawn, and/or the latent subgroups into which participants may cluster.

Third, conventional models usually must be implemented within a structural equation modeling (SEM) framework that requires a unique variable to be assigned for each observation. This makes the models difficult to apply if there are more than a modest number of observations per participant (e.g., 20; M. Wang et al., 2016). Finally, despite the flexibility of models like the LCSM, they have mainly been used to answer a fairly narrow set of questions about the relationships between the levels of certain variables and subsequent changes in those levels (e.g., "is the level of Variable X associated with a change in Variable Y"). We argue that the full potential of these models has yet to be realized.

In this paper, we demonstrate a more flexible approach to modeling longitudinal data that overcomes the challenges described above. This approach takes advantage of recent advances in Bayesian statistical modeling (e.g., Carpenter et al., 2017; Hoffman & Gelman, 2014), which enable straightforward implementation of complex models like the LCSM within a Bayesian framework. The generality of this approach makes it possible to answer nuanced questions about individual variation within a population (e.g., via hierarchical modeling) and/or the clustering of individuals into subgroups (e.g., via mixture modeling). Unlike the SEM framework, this approach can be applied to datasets with a large number of observations per participant. Finally, there is virtually no limit to the model form that can be specified within this framework. This flexibility makes it easy to test relatively specific theoretical assumptions, or even estimate the parameters of mathematical or computational models.

In the next section, we provide an overview of the approaches that have conventionally been used to model longitudinal data. We then describe some of the challenges associated with the approaches that have been used thus far. After that, we demonstrate the alternative approach described above using data from a recent study to facilitate understanding. We begin by showing how this framework can be used to implement an analogue of the standard univariate LCSM. We show the different ways this univariate model can be used to model a single group. In this section, we demonstrate how to estimate parameters at the group level, at the participant level, at the participant and group levels simultaneously via a hierarchical model. We then show how this framework can be extended to model multiple groups. In this section, we demonstrate how to estimate parameters for different, known groups (e.g., clinical sub-populations or levels of an experimental manipulation), and how mixture modeling can be used to examine the clustering of participants into latent subgroups that were unknown a priori. We then demonstrate how the univariate framework can be generalized to implement more sophisticated models such as bivariate models, closed-loop process models, and non-standard models that provide a more precise description of the psychological process being investigated.

Throughout the paper, we have aimed to keep the level of exposition accessible to the researcher who has some experience using methods such as multilevel or structural equation modeling, but who does not necessarily have any experience with platforms such as R or Stan, or with Bayesian analysis more generally. We have kept the mathematical equations to a minimum, using them only where we believed they are required. We strongly believe that for a paper such as this to be useful to a non-methods oriented reader, there must be enough practical content for the reader to apply this framework to his or her own work. We have therefore presented the computer code

required to specify every model that we demonstrate. The models are implemented in Stan, a program with which we expect many readers will be unfamiliar. So we have provided comprehensive descriptions of the code in text. We have also included all the data and code required to implement the models presented in the paper, and many additional models, in the supplementary material.

Conventional Approaches to Modeling Longitudinal Data

Historically, multilevel regression models have commonly been used to analyze longitudinal data (e.g., Bliese & Ployhart, 2002; Hox, Stoel, Everitt, & Howell, 2005). Within this framework, time is modeled as an independent variable at the lowest level (coded as a series of consecutive integers from 0 to T), and the individual is specified at the second level. The intercept in the model indicates the average initial value of the outcome variable. The coefficient associated with the time variable indicates the average rate of change in the outcome over time. Individual differences in the trajectory of the outcome variable over time are modeled as random effects. Non-linear trajectories can be modeled by including higher order polynomials (e.g., quadratic, cubic, etc) in the regression equation.

Another approach that has also been commonly used is latent growth curve modeling (MacCallum & Austin, 2000; Muthen, 1997). The latent growth curve model is implemented within a structural equation modeling framework, where time is modeled by assigning unique variables for each observation of the outcome and estimating latent intercept and slope factors that describe the change in the outcome over time. To do this, factor loadings for the latent intercept are fixed to 1 for all observations. Loadings for the latent slope factor are fixed according to their temporal ordering (e.g., for a linear model, the loading for the first observation equals 0, the loading for the second observation equals 1, etc). Under these constraints, the means for the latent intercept and slope factors indicate the average initial value and rate of change in the outcome respectively. Individual differences in the trajectory are modeled as the variances and covariance of the intercept and slope factors. Non-linear trajectories can be modeled by adding additional latent factors with loadings that specify trends of higher order polynomials.

Multilevel modeling and latent growth curve modeling are both useful tools for modeling the trajectory of an outcome variable over time. They also both adequately address issues inherent in longitudinal data analysis such as non-independence due to nesting of repeated observations within individuals. However, in recent years, there has been a shift in focus from describing patterns of change in variables over time to elucidating the underlying process that gives rise to those changes (Dinh et al., 2014; Kozlowski, 2015; Neal et al., 2017). In other words, the emphasis is not just on understanding *how* variables change over time, but on understanding *why* they change. It is difficult to answer the latter question using multilevel regression or latent growth curve modeling because, although these models can be used to identify factors that covary with the outcome or its trajectory, it is difficult to make causal inferences regarding the nature of these relationships (Liu, Mo, Song, & Wang, 2015).

The latent change score model (LCSM) is a more general framework for modeling change over time that provides a solution to the issue above (McArdle, 2009). Like the latent growth curve model, the LCSM is a class of structural equation model. The LCSM is better suited to examining dynamic processes because it puts the focus not on the level of the variable itself, but on the *change* in level at a given point in time. This makes it easy to disassociate the various drivers of change in a variable of interest. In its simplest form, the model decomposes the change in a single variable at time t (denoted Δy_t) into two components:

$$\Delta y_t = \alpha + \beta y_t. \quad (1)$$

The *constant change* component, represented by α , captures change in variable y that is stable over the entire time series. In other words, the constant change quantifies the amount of change in y that is unaffected by other factors. The *proportional change* component, represented by β , captures change in y that depends on the level of the variable itself at time t . The proportional change quantifies the extent to which the value of y at time t feeds back and influences the change in y . Additional terms can be added to account for the influence of other factors on the change in y .

Because it explicitly models the change in a variable, the LCSM makes it easy to test hypotheses pertaining to dynamic effects (Kievit et al., 2017; M. Wang et al., 2016). It is also flexible with regard to the types of trajectories for which the model can account. It can even be used to model trajectories with unknown functional forms. For these reasons, the LCSM has become widely used for longitudinal data modeling not only within organization science, but also in areas such as developmental psychology (e.g., Keller & El-Sheikh, 2011), neuroscience (e.g., Kievit et al., 2017), health psychology (e.g., Barker, Rancourt, & Jelalian, 2014), cognition (e.g., Mcardle & Prindle, 2008), and social psychology (e.g., Sanford, 2014). Despite this breadth of interest however, we contend that use of the LCSM has thus far been limited to a relatively narrow set of applications. We believe that the fixation on SEM as the predominant framework for analyzing longitudinal data may be leading researchers to overlook opportunities for developing more general models of change. In the next section, we describe ways in which we believe the applicability of longitudinal models can be broadened.

Opportunities for Broadening Applicability

In this section, we highlight avenues that we believe have a great deal of untapped potential to add value to the way researchers model longitudinal data.

Bayesian Inference

Bayesian methods have become more widely used in recent years, in part due to the emergence of Bayesian analogues of standard statistical tests, and open-source software that makes them easy to implement (e.g., JASP; JASP Team, 2018). However, the models required to analyze longitudinal data are typically more complex than these standard tests, and usually cannot be applied in an off-the-shelf manner. This makes it difficult to develop standard Bayesian implementations for models of longitudinal data. As a result, there has been relatively limited uptake in Bayesian methods for modeling longitudinal data within organizational psychology and behavior.

The advantages of Bayesian analysis are numerous, and have been thoroughly discussed elsewhere (e.g., Dienes, 2008; Edwards, Lindman, & Savage, 1963; Kruschke, Aguinis, & Joo, 2012; Wagenmakers et al., 2018, among many others). Here, we briefly address a few of these advantages that are likely to be particularly relevant when modeling longitudinal data. At its core, Bayesian inference is based on probability theory (Jaynes, 2003; Jeffreys, 1939). This connection to probability theory allows one to make probabilistic inferences (e.g., "there is a 99% probability that the constant change in a variable is positive"). Such inferences are not possible within the classical, frequentist framework. This ability to deliver probabilistic inferences also applies to competing sets of hypothesis or models (e.g., "the null hypothesis is more likely than the alternative given the data"). These kinds of probabilistic statements are ultimately the sorts of inferences that researchers seek

to make, which is why many have cited the Bayesian paradigm as the ideal approach to scientific inference (e.g., Etz & Vandekerckhove, 2018; Joyce, 1998; Lindley, 1993)

A well-known feature of the Bayesian approach is the ability to incorporate prior information into the analysis. As we will demonstrate, the consideration of this information provides a natural way of developing hierarchical models and for reducing measurement error (also see Boehm, Marsam, Matzke, & Wagenmakers, n.d.). Another advantage of Bayesian inference is access to information regarding the uncertainty in a parameter estimate. Frequentist parameter estimation methods such as least-squares or maximum likelihood provide only a single point-estimate that represents the 'best guess' of the true parameter value. A Bayesian analysis derives the full posterior distribution on each parameter, which indicates the range of parameter values that are most probable given ones prior information and the observed data.

Another benefit of Bayesian analysis that is perhaps less appreciated is its sophisticated way of defining model complexity. Model complexity generally refers to the flexibility of a model and the diversity of possible observations for which it can account (Myung & Pitt, 1997; Vandekerckhove, Matzke, & Wagenmakers, 2015). It is often the case that a researcher will analyze longitudinal data by testing a series of competing models. The ability of each model to explain the data is typically quantified by evaluating goodness-of-fit relative to model complexity. Complexity is penalized because a more flexible model is less parsimonious, so there is a need to demonstrate that any increase in complexity is justified by an increase in the fit of the model to the data.

In frequentist analysis, it is usually the case that complexity is defined based on the number of parameters. This is the case for commonly used indices such as the AIC (Schwarz, 1978), BIC (Schwarz, 1978), and RMSEA (Steiger, 1990). These methods are problematic because they ignore complexity introduced by the functional form of the model or the size of the space of possible values that the model parameters can take on (see Myung & Pitt, 1997, for a comprehensive treatment of this topic). Failure to account for these additional sources of complexity can result in incorrect conclusions about the relative evidence for each model. A Bayesian analysis naturally accounts for these sources of complexity.

A final advantage of the Bayesian approach is that its validity is not contingent on the adherence to a pre-specified sampling plan. Classical significance tests generally require data to be collected according to a fixed procedure, for example, where the researcher specifies the target sample size ahead of time and stops collecting data when the target size is reached. Failure to adhere to the pre-specified sampling plan invalidates the statistical test (Schönbrodt, Wagenmakers, Zehetleitner, & Perugini, 2017; Wagenmakers, 2007). This requirement poses a serious problem for researchers in organizational behavior or psychology who wish to analyze naturalistic data that may not have been collected for the purpose conducting the statistical test. In these fields, it is often the case that data are obtained after the fact, and therefore usually do not meet the assumptions of traditional tests. This is not a problem in the Bayesian framework. Under a Bayesian approach, the question of how the data were obtained is irrelevant, because the validity of the analysis does not depend on the sampling plan (Lindley, 1993; Wagenmakers et al., 2018, 2016).

Hierarchical and/or Mixture Modeling

Multilevel modeling has become commonplace in organizational psychology and behavior, due to the frequent need to account for the nesting of individuals within groups (e.g., organizations). However, most existing applications of LCSM are implemented as single-level analyses (M. Wang et al., 2016). This approach assumes that participants are independent from each other and not

nested within groups. The single-level framework generally restricts the inferences that can be made to those involving group-level effects.

The single-level approach limits the options a researcher has for exploring the higher level structure of the dynamic process under investigation. For example, one might wish to obtain parameter estimates for each individual to clarify the nature of the between-participant variation in different components of the process. This requires the use of a hierarchical model (. One might also wish to examine whether participants cluster into distinct subgroups, which requires the use of a mixture model.

Hierarchical and/or mixture models are straightforward to implement within a Bayesian framework, and offer a number of advantages over single-level models (Kruschke, 2010; Lee, 2011; Rouder & Lu, 2005). Hierarchical Bayesian models are implemented by estimating unique parameters for each individual, but assuming those parameters are drawn from a common population distribution. This framework enables inferences to be made about the population from which participants are sampled, rather than limiting inferences to the sample observed.

Mixture models are implemented by assuming that observations are the result of a combination of data-generating processes (e.g., multiple group membership). This analysis enables one to examine between-group differences in group-level effects and to identify the influence of each group on the individual. This allows inferences to be made about the existence of latent subgroups.

Increased Flexibility

Although the LCSM is currently the most general approach for implementing longitudinal models, applications of this framework thus far have been limited to a relatively narrow set of research questions. These research questions typically center around identifying factors that covary with subsequent change in a given variable (e.g., Sanford, 2014; Taylor et al., 2017). We argue that the restricted range of questions that these models have been used to answer may in part be due to how these models are implemented.

Models such as the LCSM must be constructed within the SEM framework, which can be implemented using programs such as LISREL, Mplus, or in R via packages such as lavaan (Rosseel, 2012), sem (Fox, 2006), and OpenMX (Neale et al., 2016). Although some Bayesian analyses can be implemented in programs such as Mplus (Kaplan & Depaoli, 2012), there are many challenges associated with using SEM for longitudinal data modeling. The SEM framework was not designed for modeling dynamic systems. This framework requires the specification of a unique variable for each observation in the time series, which limits the number of observations that can be analyzed to a relatively small amount (around 20; M. Wang et al., 2016). Even if the number of observations being analyzed is small, the models implemented within the SEM framework can be extremely cumbersome to code. In many cases, applying this type of analysis requires a complex model specification that involves "telling" the program into estimating the model (Rabe-Hesketh, Skrondal, & Zheng, 2007). Multiple parameters must be specified for each observation being analyzed, and many of those parameters must be fixed to particular values in order for the model to be identified. The complexity of implementing these models within the SEM framework increases the likelihood of misspecification, which can invalidate results (Clark, Nuttall, & Bowles, 2018). These practical challenges make it difficult for researchers to venture outside the small set of commonly used models.

Model specification issues aside, the questions that can be answered within the LCSM framework represent only the tip of the iceberg when it comes to theorizing about dynamic processes.

There are a range of interesting questions that cannot easily be answered using this approach. For example, one might test theoretical assumptions that involve conditional relationships (e.g., a salesperson might raise their sales target by some amount if the target is achieved, but otherwise leave it unchanged). Additionally, one might wish to “close the loop” by modeling the process by which changes in the system feed back and influence performance at later time points. These types of questions are often outside the scope of what can be examined using conventional methods.

Until recently, our toolbox for theorizing about such nuanced aspects of a dynamic processes was limited to simulation studies (e.g., Ballard, Yeo, B. Vancouver, & Neal, 2017; Davis, Eisenhardt, & Bingham, 2007; Harrison, Lin, Carroll, & Carley, 2007; Vancouver, More, & Yoder, 2008). A simulation study involves instantiating a theory in the form of a computational model, selecting parameters values for the model, and then simulating the model using the selected parameter values to examine how the theorized process plays out over time. Whilst useful for understanding and developing the predictions of a theory, this approach does not allow for a strong test of the theory. A strong test of the theory requires fitting to data and estimating the parameters of the model. The framework introduced below is flexible enough to implement not only commonly-used longitudinal models, but also estimate the parameters of computational models that make relatively specific assumptions.

Modeling a Single Group

In the sections that follow, we demonstrate an alternative, Bayesian framework for modeling change that provides the opportunities described above. The aim of a Bayesian analysis is to determine the inferences that can be made about a set of parameters in light of some empirical observation (Etz, Gronau, Dablander, Edelsbrunner, & Baribault, 2018; Kruschke et al., 2012; Lee & Wagenmakers, 2013). This involves combining information about the researcher’s a priori beliefs regarding each parameter, referred to as its *prior distribution* (henceforth, prior), with information about the *likelihood* of the model given the data. This analysis results in a *posterior distribution* (henceforth, posterior) on each parameter, which represents the range of parameter values that are credible given the researcher’s prior and the observed data. A highly dispersed posterior, which covers a broad range of possible values, suggests a high degree of uncertainty regarding the true parameter value. A narrow posterior, which covers only a small range of values, suggests more certainty.

Although the Bayesian approach is simple conceptually, the implementation of these methods is often complex. For most models, including the types of models likely to be of interest in OP/OB, there is no closed-form analytic solution for calculating the posterior. The posterior must therefore be approximated using Markov chain Monte Carlo (MCMC) methods. MCMC methods refer to a class of algorithms that can be used to generate a large number of representative samples from the posterior distribution (see Kruschke, 2010; Van Ravenzwaaij, Cassey, & Brown, 2018). Broadly, these methods work by producing sequences, or chains, of sample parameter values that are generated by an algorithm that is known to approximate the posterior given a sufficient number of samples.

There are several open-source platforms for implementing Bayesian models using MCMC methods. These platforms include WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000), JAGS (Plummer, 2003), OpenBUGS (Thomas, O’Hara, Ligges, & Sturtz, 2006), and Stan (Carpenter et al., 2017). The advantage of programs such as these is that they allow the user a great deal of flexibility with regard to model specification, but the MCMC algorithm is implemented under the

hood. This allows the user to focus on the development of the model itself. In this paper, we use the Stan platform. We use Stan because its MCMC algorithm, the No-U-Turn Sampler (Hoffman & Gelman, 2014), is particularly well suited for implementing more complex models with many parameters (e.g., hierarchical models Stan Development Team, 2017).

To facilitate understanding, we use real data from an experiment by Gee, Ballard, Yeo, and Neal (2013). In their study, 60 participants each performed a 70-minute air traffic control simulation task. The task required participants to classify aircraft pairs as a conflict or non-conflict based on their minimum distance of separation. The program recorded the number of correct and incorrect decisions made in each minute of the simulation. Performance was measured as the number of correct decision minus the number of incorrect decisions.

We begin by demonstrating how the above data can be analyzed in a framework analogous to a univariate latent change model, where the purpose of the analysis is to examine the dynamics of a single variable over time. We start with a **group-level** model, which is a single-level model that estimates one set of parameters for the entire sample. We then extend this framework to construct more sophisticated models of inter-individual variation.

Group-level Model

In this section, we show how to implement a simple analogue of the univariate LCSM in Stan. This model has five parameters: a constant change parameter (α), a proportional change parameter (β), an initial level parameter (y_0), and standard deviations associated with the initial value (σ_y) and the change itself (σ_Δ). Each parameter is estimated at the group level, so the number of parameters does not depend on the number of participants.

Figure 1 shows the model depicted as a graphical plate model. Graphical plate models are useful to illustrate the dependencies between the model variables. The variables are represented as nodes. Shaded nodes reflect variables observed from the data, whereas unshaded nodes reflect latent variables and parameters estimated by the model. The rectangular plates represent the sources of variability in the model. The observed performance score and latent change score variables (denoted y_{ij} and Δy_{ij} respectively) are inside both plates, which indicates that they can vary across people and time. The five estimated parameters are outside both plates, which indicates that they do not vary. The arrows drawn between the nodes highlight the dependencies between variables. As can be seen, Δy_{ij} is determined by α , β , and y_{ij} . The initial value of y_{ij} is determined by y_0 and σ_0 , and subsequent values of y_{ij} are determined by Δy_{ij} and σ_Δ .

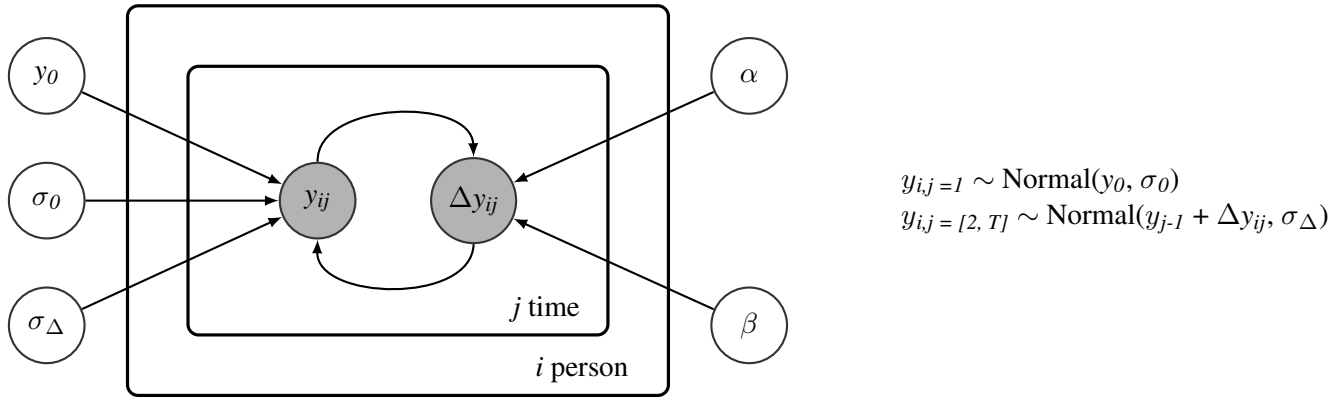


Figure 1. Graphical and mathematical representation of the Bayesian group-level model.

Like programs such as Mplus or Stata, Stan is a syntax-based platform, which means that the model must be expressed via computer code. The code required to specify the group-level model is presented below:

```

1 data {
2   int Nsubj;
3   int Nobs;
4   matrix[Nsubj,Nobs] perf;
5 }
6
7 parameters {
8   real y0;
9   real alpha;
10  real beta;
11  real<lower=0> sigma0;
12  real<lower=0> sigma_change;
13 }
14
15 model {
16   //priors
17   y0 ~ normal(0,5);
18   alpha ~ normal(0,5);
19   beta ~ normal(0,5);
20   sigma0 ~ normal(0,5);
21   sigma_change ~ normal(0,5);
22
23   //likelihood
24   for(i in 1:Nsubj){
25     perf[i,1] ~ normal(y0,sigma0);
26     for(j in 2:Nobs){
27       real change_score = alpha + perf[i,j-1]*beta;
28       perf[i,j] ~ normal(perf[i,j-1] + change_score,sigma_change);
29     }
30   }
31 }

```

As can be seen, the program requires three unique blocks of code. The first is the `data` block. Here, the user must declare the data to be used as inputs to the model. The `Nsubj` and `Nobs` variables are single values that indicate the number of subjects and the number of observations per subject respectively. In this dataset, `Nsubj = 60` and `Nobs = 70`. The `perf` variable is a matrix of performance scores. Each row in this matrix represents a unique subject and each column represents a unique observation for a given subject. Thus, `perf` is a 60×70 matrix. Note that Stan requires the user to declare an object type for each input variable. In the above, the `int` identifier is used to declare `Nsubj` and `Nobs` as integers. The `matrix[i, j]` identifier indicates that `perf` is a matrix with i rows and j columns.

The second block is the `parameters` block. Here, the user must declare the parameters that are to be estimated. In this model, `y0` is the initial performance level, `alpha` and `beta` are the constant and proportional change parameters respectively, and `sigma0` and `sigma_change` are the standard deviations of the initial level of and change in performance respectively. Note that all five parameters are declared as `real` which means they can take on any real value. The two standard deviation parameters are declared with the `<lower=0>` constraint, which imposes a lower bound of 0 on these parameters (which is necessary because a standard deviation, by definition, must be positive).

The third block is the `model` block. The first section of the model block defines the priors. The `~` operator is used to define a variable or parameter that has a distribution, and can be read as *has a distribution of*. When an unknown parameter (as opposed to a known variable) is specified on the left side of `~` operator, the distribution given on the right side becomes the prior for that parameter. Lines 8-10 assign the `y0`, `alpha`, and `beta` parameters normally distributed priors with a mean of 0 and standard deviation of 5. In the context of this dataset, these are uninformative priors that do not place a high degree of prior belief on any particular region of the parameter space. The `sigma0`, and `sigma_change` parameters are also assigned normally distributed priors. However, because we imposed a lower bound of 0 on these parameters in the `parameters` block, the algorithm will only sample positive values from these distributions.

The second section of the `model` block is where the model itself is defined. By “model”, we mean the sequence of operations that are required to determine the likelihood of the data given the sampled parameter values. As can be seen, the model is constructed using a pair of nested for-loops. A for-loop is a programming tool that enables a section of code to be executed repeatedly, with what is known as the *looping variable* taking on a different value in each execution. In the model above, `for(i in 1:Nsubj)` initializes a for-loop that iterates through the series of consecutive integers from 1 to `Nsubj`. The looping variable, `i`, is reassigned with each execution of the loop. In the first execution, `i = 1`; in the second, `i = 2`; and so on. This first loop therefore iterates through each subject, performing a series of operations on each one.

For each subject, the model treats each observation of `perf` in turn as a dependent variable. When a known variable (as opposed to an estimated parameter) is specified on the left side of the `~` operator, it is treated as an outcome variable, with the arguments on the right side representing the predicted distribution of that variable. On line 25, we specify the first performance observation for subject i (indexed by `perf[i, 1]`) to be normally distributed with a mean of `y0` and standard deviation of `sigma0`. The algorithm will therefore evaluate the likelihood of subject i 's first performance observation given the samples values of `y0` and `sigma0`, and update the posterior accordingly. The model then contains a second for-loop that this time iterates through observations 2 to `Nobs` for subject i . With each execution of the loop, the model calculates the predicted change score (line 27)

based on the sampled values of `alpha`, `beta`, and the previous performance observation for subject i (indexed by `perf[i, j-1]`). It then specifies the current performance observation (`perf[i, j]`, line 28) to be normally distributed with a mean equal to the previous performance observation plus the predicted change score and a standard deviation equal to `sigma_change`. In other words, we are regressing the current performance observation on the sum of the previous performance observation and the change score, with `sigma_change` representing the residual standard deviation of the current performance observation¹.

To run the model, the user must call Stan using R, MATLAB, Python, Julia, or the command line. For these analyses, we run Stan via the RStan package (Stan Development Team, 2016), which provides an R interface to Stan (see supplementary materials for R code used to run the model). Stan returns an object that contains samples from the posterior distribution for each parameter in the model. The RStan package provides a summary of each posterior, which is shown below. The summary contains the posterior mean, the standard error of that mean, and the standard deviation of the posterior. It also displays the 2.5%, 25%, 50%, 75%, and 97.5% quantiles of the distribution.

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
2 y0	0.12	0.00	0.22	-0.32	-0.02	0.12	0.27	0.55	4000	1
3 alpha	0.47	0.00	0.02	0.43	0.46	0.47	0.49	0.52	4000	1
4 beta	-0.90	0.00	0.02	-0.93	-0.91	-0.90	-0.89	-0.87	4000	1
5 sigma0	1.71	0.00	0.16	1.43	1.60	1.70	1.81	2.06	4000	1
6 sigma_change	1.27	0.00	0.01	1.25	1.27	1.28	1.28	1.30	4000	1

The output also provides some information to help assess model convergence. The number of effective samples (`n_eff`) is a measure of the information contained in the posterior samples that accounts for the autocorrelation between neighboring samples in each MCMC chain. Specifically, the number of effective samples represents the number of independent samples that would produce an equivalent amount of information to the set of samples obtained. If there is no autocorrelation and samples are effectively independent, the number of effective samples will be equal to the total number of samples (4000 in this example, which is the RStan default). This is the case for all parameters in the above model. For parameters that are more difficult to estimate, samples may be autocorrelated. Autocorrelation reduces the information value of each individual sample because it creates dependencies between consecutive samples. In this case, the number of effective samples will be fewer than the actual number of samples, indicating the information contained in the posterior is less than it would be if the samples were completely independent. It is important to make sure the effective sample size is large enough that the approximated posterior will be representative of the underlying distribution. Here, the effective sample sizes are sufficiently large that we can be confident in the obtained posteriors. The final piece of information provided is the potential scale reduction factor (`Rhat`; ??). This gives an indication of the extent to which the chains converged on the same region of the parameter space. As the chains approach convergence, `Rhat` approaches one. A common rule of thumb is that `Rhat` should be less than 1.1 before any inferences are made on the posteriors (Kruschke et al., 2012).

The information regarding the parameter posteriors can easily be broken down for more detailed exploration. It is good practice to examine the full posterior distribution on each parameter, as well as the joint posterior distribution between pairs of parameters. To illustrate, the top row of Figure 2 contains the posteriors on the constant change (`alpha`) and proportional change (`beta`) parameters. The first two columns show an approximation of the full posterior distribution for the two

¹To keep the example simple, we have assumed that there is no measurement in error in the performance variable. We have included an example of how to model measurement error within this framework in the supplementary material

parameters. As can be seen in the top-left panel, the most probable values for the constant change parameter lie within the range of 0.4 to 0.55, suggesting that the most probable average increase in performance from one time point to the next is within that range. As can be seen in the top-middle panel, the most probable values for the proportional change parameter are negative, within the range of about -0.96 to -0.84. This suggests that the overall level of performance has a detrimental impact on the change in performance from one trial to the next. In other words, improvements in performance become smaller as performance gets better. This is consistent with the idea that performance should reach an asymptote given sufficient learning time. The top-right panel in Figure 2 shows the joint posterior on the constant and proportional change parameters. Here, the inner most contours represent the most probable parameter values.

Once the researcher is satisfied that the model has converged and that the parameters are well estimated, they will likely wish to systematically determine the most probable values for each parameter. One commonly used way of summarizing the most probable parameter values is by calculating the 95% *credible interval* (CI). The 95% CI spans the 2.5% and 97.5% quantiles of the posterior distribution. In the above model, the CI on `alpha` is 0.43 to 0.52, and the CI on `beta` is -0.93 to -0.87. The 95% *highest density interval* (HDI) is another commonly used interval for summarizing the posterior distribution. The 95% HDI represents the shortest possible interval containing 95% of the posterior density, and will therefore always include the most probable parameter values (Kruschke et al., 2012). The CI and HDI each has advantages and disadvantages (e.g., see Kruschke, 2010), but for symmetrical distributions the two intervals will be very similar.

The CI or HDI is often used to decide whether particular parameter values should be accepted or rejected. For example, intervals that exclude zero are often taken as evidence that the parameter value is reliably different from zero (Kruschke et al., 2012). Another simple way to quantify the evidence for a parameter being either greater than or less than a particular value is to determine the percentage of the posterior density that is above or below that value (Kruschke, 2010). For example, 100% of the posterior `alpha` distribution is above 0, whereas 0% of the posterior `beta` distribution is above 0. This provides strong evidence that `alpha` is positive and that `beta` is negative. It should be noted that some have argued against making inferences based only on a parameter's posterior distribution (e.g., Rouder, Morey, Verhagen, Province, & Wagenmakers, 2016) and instead advocate for inferences based on model comparison. We return to the issue of model comparison later in the paper.

Although estimating parameters at the group-level can be informative, this approach can only be used to examine average effects. In this model, the `alpha`, `beta` parameters represent the *average* rate of constant and proportional change across participants in the sample. These parameters tell us nothing about these components of the process vary across individuals. Moreover, the `sigma_change` parameter conflates within-person and between-person variation in change, making it difficult to interpret. This model therefore cannot be used to examine how these effects may differ between members of a group or population. To do this, we require a model in which unique parameters are estimated for each person. We present such a model in the next section.

Person-level Model

Whilst making inferences purely at the group-level can be useful, there are likely to be instances where one wishes to examine effects at the level of the individual participant. For example, ...XYZ. The model presented above can be easily extended to estimate parameters at the person level. To illustrate, figure shows the graphical representation of a model that estimates unique

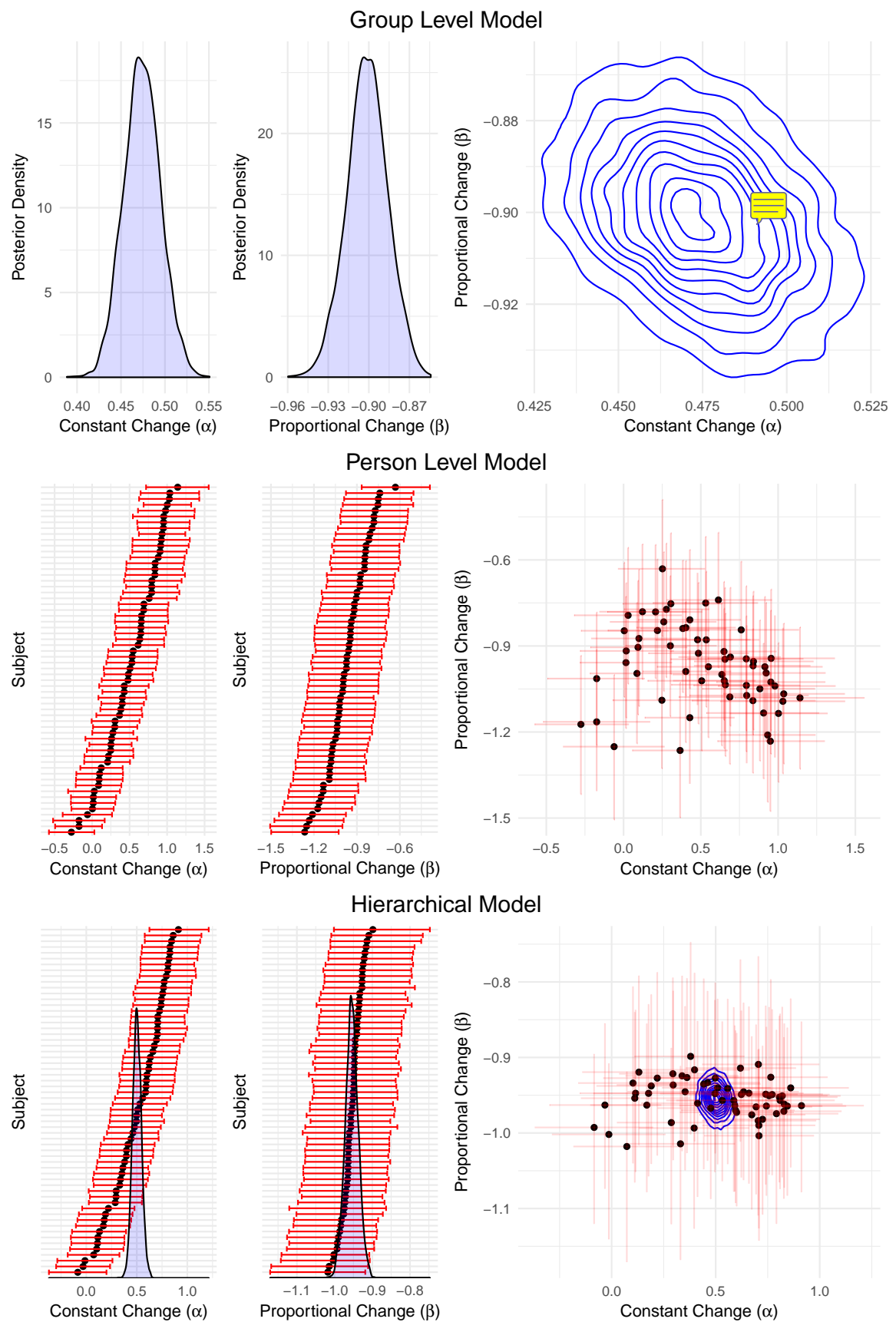


Figure 2. Posterior distributions on the constant and proportional change parameters for the group-level, person-level, and hierarchical models.

`alpha`, `beta`, and `sigma_change` parameters for each participant. As can be seen, the α , β , and σ_{Δ} are presented inside of the person plate and subscripted with i . This indicates that these parameters vary across individuals.

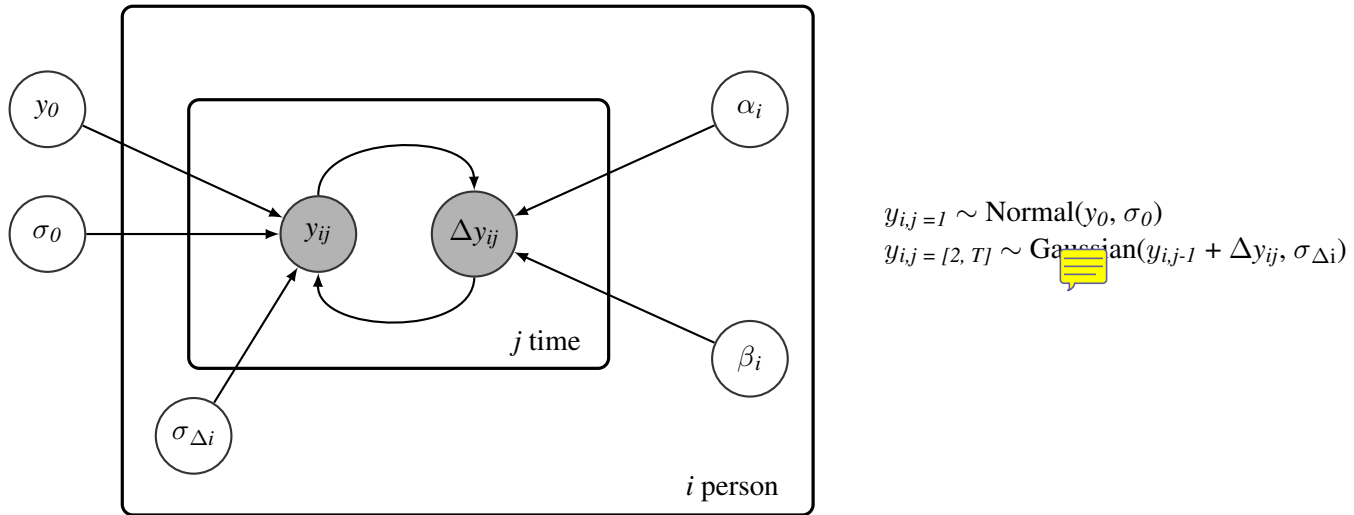


Figure 3. Graphical and mathematical representation of the Bayesian person-level model.

The code below shows the changes to the `parameters` and `model` blocks needed to estimate the person-level model.

```
...
7  parameters {
8    real y0;
9    real alpha[Nsubj];
10   real beta[Nsubj];
11   real<lower=0> sigma0;
12   real<lower=0> sigma_change[Nsubj];
13 }

...

27   real change_score = alpha[i] + perf[i,j-1]*beta[i];
28   perf[i,j] ~ normal(perf[i,j-1] + change_score,sigma_change[i]);

...
```

Only two simple changes are required. First, in the `parameters` block, the clause `[Nsubj]` has been added after the declarations of the `alpha`, `beta`, and `sigma_change` parameters (lines 9, 10, and 12). This indicates that `alpha`, `beta`, and `sigma_change` are now parameter arrays with lengths equal to the number of participants (as opposed to scalar values). Each element of the array corresponds to one participant, so there will be a unique parameter estimated for each one. The second change is on lines 27 and 28. As can be seen, `alpha`, `beta`, and `sigma_change` are now indexed with `[i]`. This index means that change score will be calculated using the `alpha` and `beta` values associated with subject i (see line 27). Furthermore, the standard deviation of that change score will be quantified separately for each person (see line 28).

This model produces a set of posterior `alpha`, `beta`, `sigma_change` samples for each participant. The middle row of Figure 2 shows the results for the `alpha` and `beta` parameters. The first two plots show the 95% credible intervals on `alpha` and `beta` for each participant. Note that it is possible to access the full posterior for each participant, but the credible interval is more visually compact and makes it easier to compare a larger number of participants. As can be seen, there is heterogeneity between participants in both the constant change and proportional change in performance over time. The right panel shows the constant and proportional change parameters for each participant plotted against each other (with the crosses representing the credible intervals for each parameter).

Hierarchical Model

Whilst parameter estimation at the person level has advantages over the group-level model, one problem with the person-level model is that it can be difficult to make inferences about the group or population from which participants are drawn. Analyzing a person-level model is akin to running a separate, single-level analysis on each participant in a sample. This approach has less power because it only considers data from one participant at a time, and ignores the rest of the sample. It also fails to capture commonalities between individuals that may arise from participants being members of the same population.

It is quite often the case that researchers wish to examine variation between participants, but also to make inferences about the population as a whole. A hierarchical modeling approach is extremely useful for this purpose (Kruschke & Vanpaemel, 2015; Turner et al., 2013; Vincent, 2016). Hierarchical Bayesian models allow researchers to “have their cake and eat it too” by modeling the individual and group levels simultaneously (Lewandowsky & Farrell, 2011). As with the person-level model, the hierarchical model estimates unique parameters for each individual. However, unlike the person-level model, the hierarchical approach also models the distribution of the person-level parameters at the group level. The model explicitly captures both the top-down process by which group membership influences the individual, and the bottom-up process by which individuals influence the group-level distribution of parameters or effects. As a result, person-level parameters are informed not only by the data for that one individual, but also by the group-level distribution on the relevant parameter. This increases the accuracy of the parameter estimates (Boehm et al., n.d.; Rouder & Lu, 2005).

Figure 4 shows the graphical representation of a model that estimates the `alpha`, `beta`, and `sigma_change` parameters hierarchically. As can be seen, this model assumes that α_i , β_i , and $\sigma_{\Delta i}$ are each informed by two higher level parameters— μ and σ —which together describe the nature of the variability in these parameters at the group level.

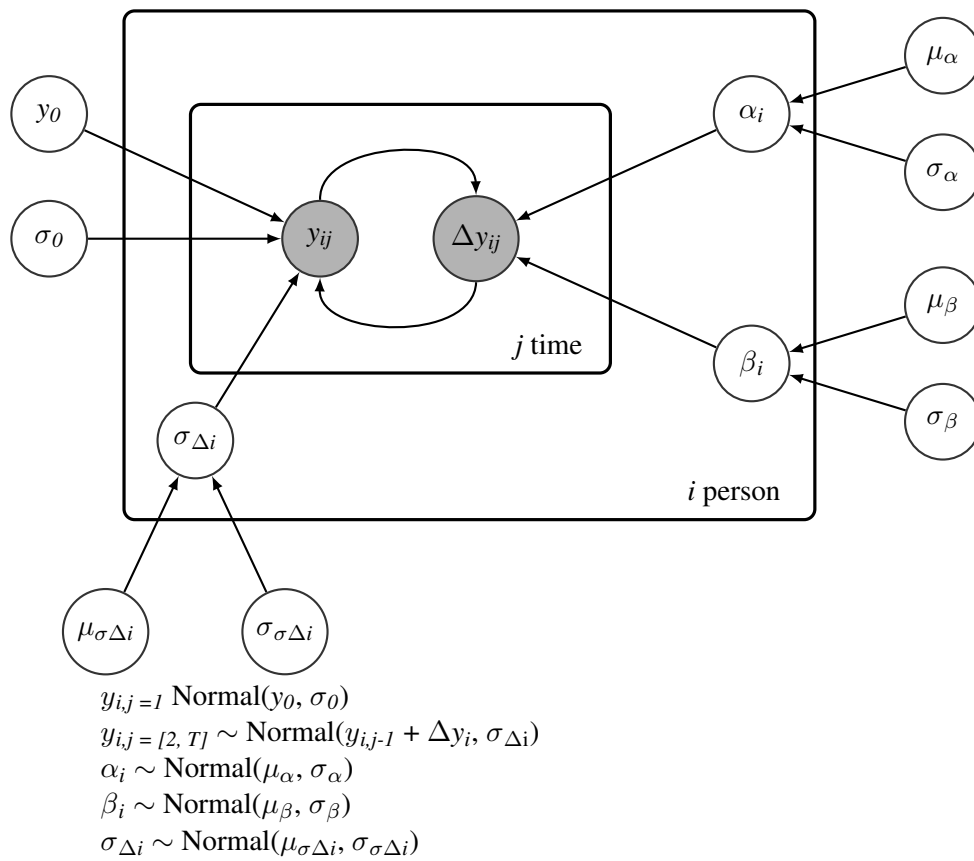


Figure 4. Graphical and mathematical representation of the hierarchical Bayesian model.

The code below shows the `parameters` and `model` blocks required to implement this model (the `data` block does not need to be changed from the original example).

```

7 parameters {
8   real y0;
9   real alpha[Nsubj];
10  real beta[Nsubj];
11  real<lower=0> sigma0;
12  real<lower=0> sigma_change[Nsubj];
13  real alpha_mean;
14  real<lower=0> alpha_sd;
15  real beta_mean;
16  real<lower=0> beta_sd;
17  real<lower=0> sigma_change_mean;
18  real<lower=0> sigma_change_sd;
19 }
20
21 model {
22   //priors
23   y0 ~ normal(0,5);
24   sigma0 ~ normal(0,5);
25   sigma_change ~ normal(sigma_change_mean, sigma_change_sd);
26   alpha ~ normal(alpha_mean, alpha_sd);
27   beta ~ normal(beta_mean, beta_sd);

```

```

28  alpha_mean ~ normal(0,5);
29  alpha_sd ~ normal(0,5);
30  beta_mean ~ normal(0,5);
31  beta_sd ~ normal(0,5);
32  sigma_change_mean ~ normal(0,5);
33  sigma_change_sd ~ normal(0,5);
34
35  //likelihood
36  for(i in 1:Nsubj){
37    perf[i,1] ~ normal(y0,sigma0);
38    for(j in 2:Nobs){
39      real change_score = alpha[i] + perf[i,j-1]*beta[i];
40      perf[i,j] ~ normal(perf[i,j-1] + change_score,sigma_change[i]);
41    }
42  }
43 }

```

As can be seen, the hierarchical model requires six new parameters. The first two are `alpha_mean` and `alpha_sd` (lines 13 and 14), which represent the mean and standard deviation of the person-level distribution of `alpha`. These parameters characterize the distribution of constant change at the group-level. The `beta_mean` and `beta_sd` parameters (lines 15 and 16) represent the mean and standard deviation for the person-level distribution of `beta`. These parameters characterize the distribution of proportional change at the group-level. Finally, the `sigma_change_mean` and `sigma_change_sd` parameters (lines 17 and 18) characterize the mean and standard deviation of the person-level distribution of `sigma_change`.

As can be seen on lines 24-26, the priors on `alpha`, `beta`, and `sigma_change` differ in the hierarchical model, compared to the other models we have thus far demonstrated. Whereas in the group-level and person-level models these priors are set using fixed values that produced broad, uninformative distributions, in the hierarchical model these priors are set using the group-level means and standard deviations. In other words, the hierarchical model uses information about the group-level distribution as the prior for the person-level parameters. The parameters that define the group-level distributions—`alpha_mean`, `alpha_sd`, `beta_mean`, `beta_sd`, `sigma_change_mean`, and `sigma_change_sd`—are then given their own priors on lines 27-32. The priors on the parameters that define higher level distributions are commonly referred to as *hyperprior* or *parent distributions*.

The above hierarchical model produces unique posteriors on `alpha`, `beta`, and `sigma_change` for each subject and also group-level posteriors for `alpha_mean`, `alpha_sd`, `beta_mean`, `beta_sd`, `sigma_change_mean`, `sigma_change_sd`, `y0`, and `sigma0`. The bottom row in Figure 2 shows a breakdown of the posteriors on the constant and proportional change parameters. In the first two columns, the red lines represent the credible intervals on `alpha` and `beta` for each participant. The blue densities represent the full posterior on `alpha_mean` and `beta_mean`.

As can be seen, the posteriors on the group-level means in the hierarchical model occupy a similar region of the parameter space as the posteriors in the group-level model (shown in the top row of the figure). The reader may note however, that the credible intervals on the person-level parameters are less dispersed in the hierarchical model than in the person-level model (shown in the middle row of the figure). This is because in the hierarchical model, the group-level distribution imposes an additional constraint on the person-level parameters, which pulls the person-level parameters closer to the group mean. This process is called *shrinkage*, and it can also be seen in the bivariate plot in the bottom-right panel. Shrinkage reduces measurement error because it means

that parameters that are less reliably estimated become more strongly influenced by the group mean (Boehm et al., n.d.). This also reduces the likelihood of outliers.

Having demonstrated the advantages of a hierarchical model compared to a person-level model, it should be noted that a hierarchical model is not always the more desirable choice. For example, a hierarchical model requires some knowledge regarding the variation in person-level parameters (e.g., whether they are normally distributed). Without any knowledge about this variation, it would be difficult to know whether the model is appropriately specified. In this case, it may be useful to first implement a person-level model in order to ascertain the nature of the variation in the parameters, and then to use these results to inform the specification of a hierarchical model. Another case where a hierarchical model may be less appropriate is when the researcher is interested in individual differences. A hierarchical model imposes a certain degree of homogeneity on the person-level parameters, which may obscure the influence of individual difference variables.

Modeling Multiple Groups

The models we have addressed thus far examine variation within a single group or population. However, many research questions require the consideration of multiple groups. For example, a researcher may want to examine whether the temporal dynamics of some process differ across experimental conditions. In this case, they need to examine whether the parameters of interest differ between two or more predefined groups. Alternatively, a researcher may wish to examine whether participants naturally cluster into distinct subgroups. For example, perhaps some for participants, change in a variable of interest is driven primarily by a constant effect and is insensitive to the level of that variable, whereas for others, change is primarily driven by the level of the variable and there is little constant change.

In this section, we describe two approaches that can help to answer the types of research questions described above. The first is a multiple-group model, in which the aim is to examine whether parameters of interest differ among known groups. The second is a mixture model, in which the aim is to identify naturally emerging latent subgroups into which participants cluster. For simplicity, we demonstrate group level models only, where parameters vary across conditions or subgroups but not across people. However, both of the models we present can also be implemented as hierarchical models, where person-level parameters are drawn from different group level distributions that each represent a unique condition or subgroup. We include code to implement such models in the supplementary material.

Known Group Membership

In the study that generated the example dataset, task framing was manipulated between participants. For half of the participants, the task was framed in approach terms. For these participants, the aim was to maximize the number of correct decisions. For the other half, the task was framed in avoidance terms. In this case, the aim was to minimize the number of incorrect decisions. In the next example model, we examine whether the temporal dynamics of performance differs between these groups by estimating unique constant change (`alpha`) and proportional change (`beta`) parameters for each group and then comparing them.

The code that specifies this model is shown below. As can be seen, there are three changes required to convert the group-level model described above to the multiple-group model. First, two new variables need to be read in to Stan. The `condition` variable (line 5) indicates the condition

for each participant (1 = Approach, 2 = Avoidance). The `Ncond` variable (line 6) is a single integer indicating the number of unique conditions (in this case, 2). Second, the `alpha` and `beta` parameters are declared as arrays with lengths equal to the number of conditions. This means that two `alpha` and `beta` parameters will be estimated, one for each condition. The final change is on line 29. As can be seen, the indexing statement `[condition[i]]` has been added to the `alpha` and `beta` parameters. This means that the change score will be calculated using the parameter values associated with the experimental condition of subject i .

```

1 data {
2   int Nsubj;
3   int Nobs;
4   matrix[Nsubj,Nobs] perf;
5   int condition[Nsubj];
6   int Ncond;
7 }
8
9 parameters {
10  real y0;
11  real alpha[Ncond];
12  real beta[Ncond];
13  real<lower=0> sigma0;
14  real<lower=0> sigma_change;
15 }
16
17 model {
18   //priors
19   y0 ~ normal(0,5);
20   alpha ~ normal(0,5);
21   beta ~ normal(0,5);
22   sigma0 ~ normal(0,5);
23   sigma_change ~ normal(0,5);
24
25   //likelihood
26   for(i in 1:Nsubj){
27     perf[i,1] ~ normal(y0,sigma0);
28     for(j in 2:Nobs){
29       real change_score = alpha[condition[i]] + perf[i,j-1]*beta[condition[i]];
30       perf[i,j] ~ normal(perf[i,j-1] + change_score,sigma_change);
31     }
32   }
33 }

```

The top row of Figure 5 shows the posteriors on the constant change (left panel) and proportional change (middle panel) parameters for the approach and avoidance conditions. As can be seen, the posterior distributions on the proportional change parameter for the two conditions overlap considerably. However, there is some separation between the posteriors on the constant change parameter. Specifically, the constant change in performance over time is generally greater in the approach condition than in the avoidance.

To examine the difference in parameter values between conditions, we need to examine the posterior distribution on this difference. This is done by calculating, for each posterior sample, a variable that is equal to the difference between the sampled parameter value for the approach condition and the sampled value for the avoidance condition. This yields a difference score for each posterior sample, which can be used to approximate the posterior distribution of the difference score.

The posterior on the difference scores for the two constant and proportional change parameters can be seen in the top-right panel.

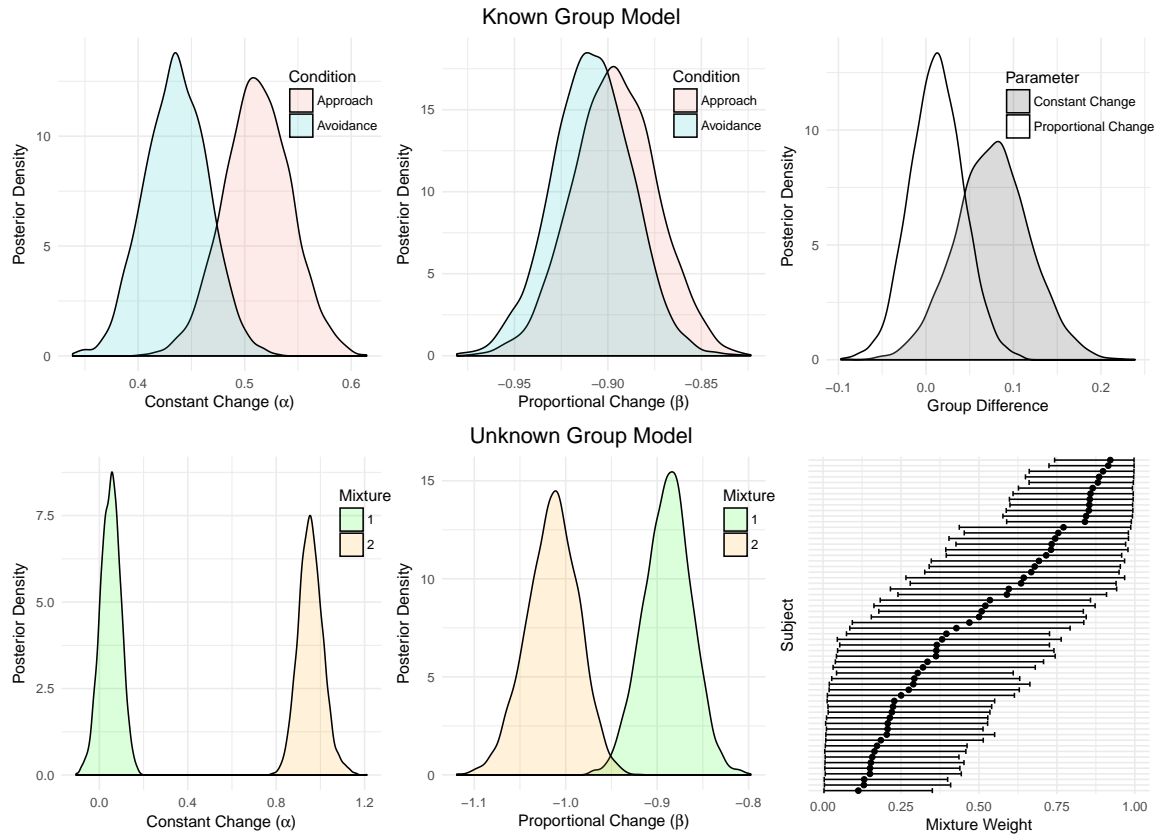


Figure 5. Posterior distributions on the constant and proportional change parameters for the known group and unknown group models.

Unknown Group Membership

In the above example, the researcher knew beforehand the group to which each participant belonged. However, this is not always the case. Sometimes, the researcher may wish to identify subgroups of participants who behave similarly, without any prior knowledge regarding group membership. This can be achieved using a mixture model. A mixture model is used to capture behavior that results from different processes (Bartlema, Lee, Wetzels, & Vanpaemel, 2014). For example, there may be subgroup of participants who behave according to one process, and another subgroup whose behavior is governed by a different process. These different processes are referred to as 'mixtures'. The goal of the analysis is to determine the parameters that best characterize each mixture, and to make inferences about the relative influence of each mixture on each participant's behavior.

The `parameters` and `model` blocks below specify a mixture model in which mixtures differ in their constant change and proportional change parameters (the `data` block does not need to change from the original example). For simplicity, we model only two mixtures in this example. The supplementary materials contain an example of how this model can be generalized to any number mixtures. There are two changes required to the `parameters` block. As with the multiple-group

model, this mixture model estimates two unique `alpha` and `beta` parameters. However, in the mixture model, one of these parameters must be declared as an ordered vector (see line 9). The ordered vector is a Stan object type that sorts the values within it in ascending order. This is needed for model identifiability.

The mixture model also includes an additional parameter (line 13). This parameter is the mixture weight (`mix_weight`), which indicates the relative influence of each mixture on the behavior of each participant. This weight ranges from 0 to 1, where higher values indicate a stronger influence of mixture 1. The model estimates a unique mixing proportion for each participant. Here, the mixture weight represents the probability of a participant belonging to a particular group.

```

7 parameters {
8   real y0;
9   ordered[2] alpha;
10  real beta[2];
11  real<lower=0> sigma0;
12  real<lower=0> sigma_change;
13  real<lower=0,upper=1> mix_weight[Nsubjj];
14 }
15
16 model {
17   //priors
18   y0 ~ normal(0,5);
19   alpha ~ normal(0,5);
20   beta ~ normal(0,5);
21   sigma0 ~ normal(0,5);
22   sigma_change ~ normal(0,5);
23
24   //likelihood
25   for(i in 1:Nsubjj){
26     perf[i,1] ~ normal(y0,sigma0);
27     for(j in 2:Nobs){
28       real change_score_mix1 = alpha[1] + perf[i,j-1]*beta[1];
29       real change_score_mix2 = alpha[2] + perf[i,j-1]*beta[2];
30       target += log_mix(mix_weight[i],
31         normal_lpdf(perf[i,j] | perf[i,j-1] + change_score_mix1, sigma_change),
32         normal_lpdf(perf[i,j] | perf[i,j-1] + change_score_mix2, sigma_change));
33     }
34   }
35 }

```

As can be seen in the `model` block, no changes to the priors are required to implement the mixture model. The reader may note that we have not explicitly assigned a prior to `mix_weight`. This is because by default Stan imposes a uniform prior, which is appropriate for `mix_weight` because the parameter is bounded on both ends. As can be seen, there are some key differences in the likelihood component of the `model` block compared to the previous models we have demonstrated. First, the model now calculates two separate predicted change scores. The first is calculated based on `alpha[1]` and `beta[1]`, which are the parameters associated with mixture 1 (line 28). The second is calculated based on the parameters associated with mixture 2 (line 29).

As can be seen on lines 30-32, the expression of the likelihood itself has also changed. The Stan syntax required to define the likelihood for a mixture model is more complex than the syntax used in the previously demonstrated models. To implement a mixture model in Stan, the `mix_weight` parameter must be marginalized out of the likelihood (Stan Development Team, 2017).



This means that the likelihood of the observation given the model must be calculated based on the weighted sum of the likelihood of the data under each separate mixture. Formally, the likelihood is expressed as follows:

$$p(y|\lambda, \mu, \sigma) = \sum_{k=1}^K \lambda_k \times \text{Normal}(y|\mu_k, \sigma_k) \quad (2)$$

where, in the above model, λ for mixture 1 is `mix_weight`, λ for mixture 2 is `1-mix_weight`, μ for mixtures 1 and 2 is `perf[i,j-1] + change_score_mix1` and `perf[i,j-1] + change_score_mix2` respectively, and σ for both mixtures is `sigma_change`. Lines 30-32 implement this likelihood function automatically using Stan's built-in `log_mix()` function².

The bottom row of Figure 5 displays the results for the above mixture model. The left and middle panels show the differences in the constant change and proportional change parameters between the two mixtures identified by the model. As can be seen, mixture 1 has weaker change components overall. The posteriors for mixture 1 on both parameters are closer to 0 than the posteriors for mixture 2. The right panel shows the credible intervals on the mixture weight for each participant. These results suggest that almost half of the participants clearly belong in mixture 1, and about a quarter clearly belong in mixture 2. For the remaining participants, the evidence for group membership is relatively balanced between the two mixtures.

Before making inferences based on a mixture model such as the above, it is advisable to test it against alternative models that specify different numbers of mixtures (Bartlema et al., 2014). For example, we might compare the two-group model above to one-group and three-group models based on which provides the best description of the data. The goal of the model comparison is to determine the number of mixtures that is most strongly favored by the evidence. We address the issue of model comparison in more detail later in the paper.

Further Extensions

The models presented thus far provide a useful foundation for developing more complex models. In this section, we provide a few examples that show how this framework can be extended to capture more complex processes.

Bivariate Model

For simplicity, we have focused thus far on modeling a single variable. However, it is common the case that a researcher will be interested in examining whether change in one variable is influenced by other variables. To do so requires a bivariate change model. A bivariate model enables one to model the change in two variables simultaneously. Similar to the univariate model, the bivariate model captures the constant and proportional change in each variable of interest. However, the bivariate model also captures the change in each variable that is attributable to the other variable.

²In this model, the likelihood cannot be evaluated using a statement of the form `y ~ normal(mean, sd)`, because the existence of the separate mixtures means that `perf[i, j]` will not be normally distributed. Instead, the likelihood must be calculated directly and the posterior must be updated using the `target+=` statement. See Stan user's guide for further information (<http://mc-stan.org/users/documentation/>).

Figure 6 shows the graphical representation of a group-level bivariate model³.

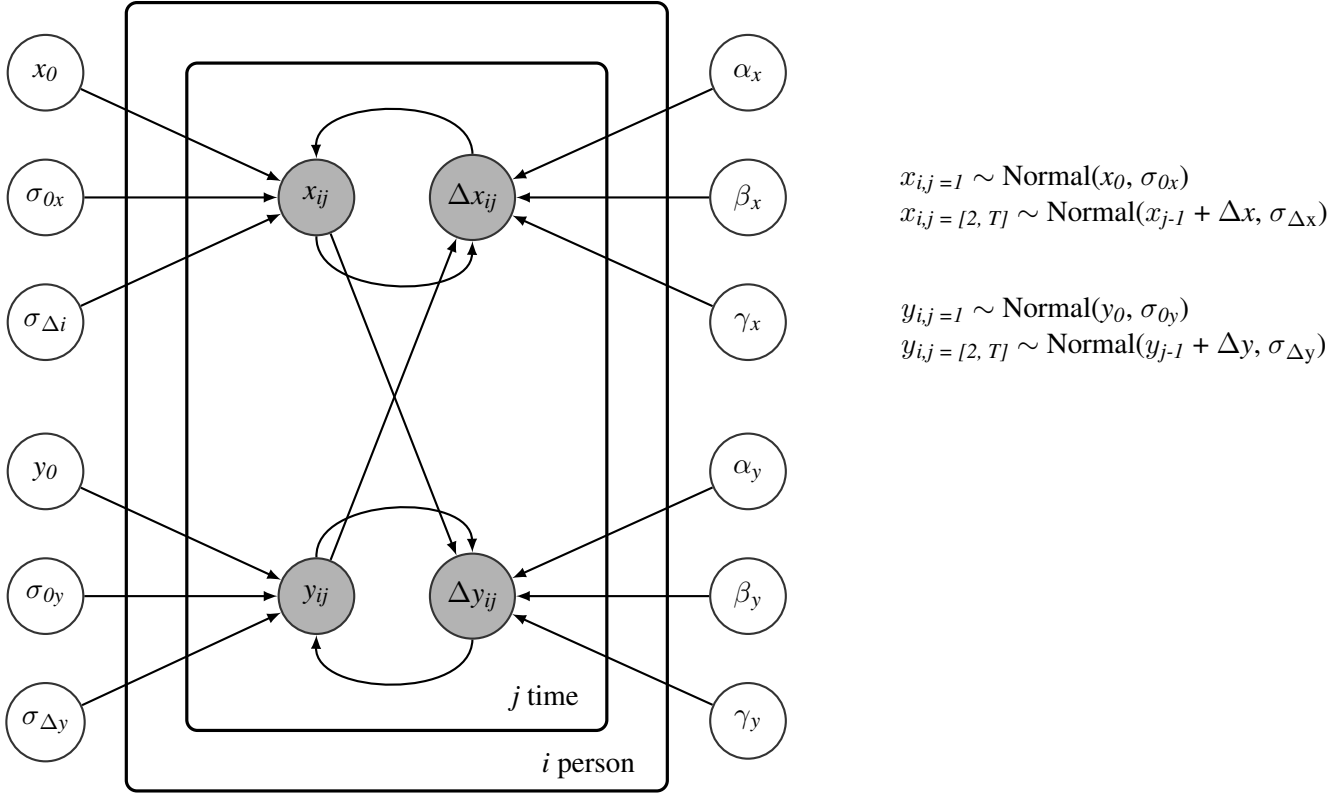


Figure 6. Graphical and mathematical representation of the Bayesian bivariate model.

The example code below implements a bivariate change model of performance and perceived difficulty. The latter was measured at the end of every 1-minute time window by asking participants to respond to the question "How hard did you find the task over the previous 1 minute?". Participants responded on a scale from 0 (Extremely Easy) to 10 (Extremely Hard).

As can be seen, the `data` block includes a new variable, `diff`, which represents the perceived difficulty rating. The bivariate model shown below has 12 parameters. As can be seen in the `parameters` block, the model includes parameters that represent initial value, constant change, proportional change, and the variability in the initial value and change scores for both performance (lines 9-13) and difficulty (lines 15-19). It also includes two new parameters that represent the effects of each variable on the change in the other variable. These parameters are referred to as *coupling* parameters and are denoted by γ . In the model, `gamma_perf` represents the change in performance that is attribute to the level of perceived difficulty at time $t - 1$, and `gamma_diff` represents the change in perceived difficulty that is attributable to the level of performance at $t - 1$ ⁴.

³As with the models presented above, the bivariate model shown here can easily be extended to implement person-level or hierarchical models. We include example code for specifying person-level and hierarchical bivariate models in the supplementary material.

⁴This model assumes that the change in performance and perceived difficulty are independent. It is also possible to

```

1 data {
2   int<lower=0> Nsubj;
3   int<lower=0> Nobs;
4   matrix[Nsubj,Nobs] perf;
5   matrix[Nsubj,Nobs] diff;
6 }
7
8 parameters {
9   real y0_perf;
10  real alpha_perf;
11  real beta_perf;
12  real<lower=0> sigma0_perf;
13  real<lower=0> sigma_change_perf;
14
15  real y0_diff;
16  real alpha_diff;
17  real beta_diff;
18  real<lower=0> sigma0_diff;
19  real<lower=0> sigma_change_diff;
20
21  real gamma_perf;
22  real gamma_diff;
23 }
24
25 model {
26   //priors
27   y0_perf ~ normal(0,5);
28   alpha_perf ~ normal(0,5);
29   beta_perf ~ normal(0,5);
30   sigma0_perf ~ normal(0,5);
31   sigma_change_perf ~ normal(0,5);
32
33   y0_diff ~ normal(0,5);
34   alpha_diff ~ normal(0,5);
35   beta_diff ~ normal(0,5);
36   sigma0_diff ~ normal(0,5);
37   sigma_change_diff ~ normal(0,5);
38
39   gamma_perf ~ normal(0,5);
40   gamma_diff ~ normal(0,5);
41
42   //likelihood
43   for(i in 1:Nsubj){
44
45     perf[i,1] ~ normal(y0_perf,sigma0_perf);
46     diff[i,1] ~ normal(y0_diff,sigma0_diff);
47
48     for(j in 2:Nobs){
49
50       real perf_change_score = alpha_perf + perf[i,j-1]*beta_perf + diff[i,j-1]*gamma_perf;
51       real diff_change_score = alpha_diff + diff[i,j-1]*beta_diff + perf[i,j-1]*gamma_diff;
52
53       perf[i,j] ~ normal(perf[i,j-1] + perf_change_score,sigma_change_perf);
54       diff[i,j] ~ normal(diff[i,j-1] + diff_change_score,sigma_change_diff);

```

model the covariance in these change scores. We include an example model that does so in the supplementary materials.

```

55
56     }
57   }
58 }

```

As can be seen in the `model` block, the bivariate model treats `diff` in the same way as `perf`. That is, the model calculates a unique change score for each variable (lines 50-51), which is used to evaluate the likelihood of each observed value (lines 53-54). The major change in the bivariate model is that there is an additional component that determines the change score. As can be seen, the predicted performance change score is determined by the previous level of perceived difficulty weighted by `gamma_perf`. The predicted difficulty change score is determined by the previous level of performance weighted by `gamma_diff`. This is how the bivariate model captures the extent to which the change in one variable is attributable to another.

Closed Loop Model

Theoretically, dynamic phenomena are *closed loop* processes (e.g., Vancouver, Weinhardt, & Schmidt, 2010; Vancouver, Weinhardt, & Vigo, 2014). A closed loop process, which is also referred to as a *feedback* process, is one in which the change incurred at time t feeds back and directly influences the state of the process at time $t + 1$. For example, a person might take some action in response to what they perceive as an unsafe workplace environment, which has the effect of bringing about new safe work practices. The presence of these practices in turn reduces the subsequent need for the person to advocate for these changes to be made.

Predictions about closed loop processes can often only be generated by simulating the entire time series. These predictions are often generated using systems dynamics models, which are computational models that are used for understanding how a system evolves over time (M. Wang et al., 2017). Systems dynamics modeling involves a) defining a set of rules that govern how the components of the system interact, b) setting each model parameter beforehand to values the researcher deems plausible, c) and simulating the model so that the researcher can observe how the dynamic process unfolds. Systems dynamics models are becoming increasingly used for theory development, because they provide a formal link between the theory and its predictions, and enable coherent theories to be developed of highly complex processes (Weinhardt & Vancouver, 2012).

Despite the increasing use of systems dynamics modeling for theorizing about closed-loop processes, it can be difficult to analyze empirical data using these models. For example, in many cases, a researcher will not only wish to generate predictions from a model based on known parameter values, but will also wish to infer the value of model parameters based on observed data. The latter is difficult using standard statistical tools. As a result, research must typically resort to open-loop models for data analysis, which do not require simulation. In an open-loop model, the predictions regarding the output of the system (e.g., the change in a variable) at one point in the time series have no direct effect on predictions later in the time series. Almost all off-the-shelf statistical models, including the latent change score model, and all of the models presented so far in this paper fall into this category.

Whereas in some cases open-loop models are appropriate for making inferences about closed-loop processes, in many cases they are not (CITE). In such cases, closed-loop models are required to make inferences about model parameters. We provide an example of a closed-loop model be-

low⁵. This model is an analogue of the group-level model presented above (i.e., the first model introduced), but is implemented as a closed-loop process model. Note that the closed-loop and open-loop implementations of this particular model do not yield differences in the constant and proportional change parameters. However, we keep to the familiar model for ease of exposition.

The difference between this model and its open-loop counterpart is that the closed-loop model simulates the entire time series, calculating predicted change scores based on previous *predicted* performance scores. By contrast, in the previous models we have demonstrated, predicted change scores are determined based on previous *observed* performance scores. This model "closes the loop" by allowing the predicted performance scores to feed back and produce changes at later stages of the process, rather than constraining predictions regarding change so that they depend on the observed data.

```

15 model {
16   matrix[Nsubj,Nobs] predicted_perf;
17
18   //priors
19   y0 ~ normal(0,5);
20   alpha ~ normal(0,5);
21   beta ~ normal(0,5);
22   sigma0 ~ normal(0,5);
23   sigma_change ~ normal(0,5);
24
25   //likelihood
26   for(i in 1:Nsubj){
27     predicted_perf[i,1] = y0;
28     perf[i,1] ~ normal(predicted_perf[i,1],sigma0);
29     for(j in 2:Nobs){
30       real change_score = alpha + predicted_perf[i,j-1]*beta;
31       predicted_perf[i,j] = predicted_perf[i,j-1] + change_score;
32       perf[i,j] ~ normal(predicted_perf[i,j] ,sigma_change);
33     }
34   }
35 }
```

In this model, the `predicted_perf` object (declared on line 16) stores the predicted performance scores for each observation in the time series, for each participant. As can be seen on line 27, the initial predicted performance score is assumed to be equal to `y0`. The predicted change score is determined by the previous predicted performance score (line 30), as opposed to the previous observed score. The predicted performance score at the current point in the time series is equal to the sum of the previously predicted score and the change score (line 31). Finally, in the closed-loop model, the observed performance is assumed to have a mean equal to predicted score and a standard deviation equal to `sigma_change`.

The closed-loop form of the model yields the same estimates of `y0`, `alpha`, `beta`, and `sigma0`. However, in this model, the `sigma_change` value will be larger, because it conflates variability in the change score itself with error in the previous predicted performance score.

Psychological Model

An important feature of this Bayesian modeling framework is its flexibility. The models we have presented thus far highlight some features that make this a versatile approach for modeling

⁵The `data` and `parameters` blocks do not need to be changed from the first example.

variation in the change process among individuals within a group, between different groups, and between different outcome variables. However, a distinctive feature of this framework that makes it even more flexible is its ability to instantiate models of virtually any functional form. This opens the door for researchers to develop customized models that provide a more accurate representation of the psychological phenomenon being investigated than would be provided by generic, off-the-shelf models.

As an example, consider the learning literature, where decades of research has focused on understanding the precise nature of the relationship between practice and skill acquisition (e.g., Estes, 1994; Thurston, 1919). Skill acquisition does not follow a linear trajectory. It is typically characterized by rapid initial improvement in performance, with changes in performance becoming smaller over time as the learner becomes more proficient. Simple linear models are not appropriate for characterizing this relationship. One common way of representing the improvement in performance with practice is the following exponential model:

$$P_t = P_\infty - (P_\infty - P_0) \cdot e^{-\alpha t}, \quad (3)$$

where t represents time, P_0 is the initial level of performance, P_∞ is the performance asymptote (i.e., P_t as t approaches infinity), and α is the learning rate (Heathcote, Brown, & Mewhort, 2000).

Non-standard models like the one above can easily be implemented in this framework. In the example model below, we use the above equation to characterize the change over time in performance and perceived difficulty. The example model has seven parameters. First, there are parameters representing the initial level of performance and the performance asymptote (lines 9-10). There are also parameters representing the initial level of perceived difficulty and the perceived difficulty asymptote (lines 11-12). There is a single learning rate parameter (line 13), which means that changes in performance and perceived difficulty are governed by the same process. Finally, there are residual standard deviation parameters for the performance and perceived difficulty variables (lines 14-15).

```

8 parameters {
9   real y0_perf;
10  real yinf_perf;
11  real y0_diff;
12  real yinf_diff;
13  real<lower=0> rate;
14  real<lower=0> sigma_perf;
15  real<lower=0> sigma_diff;
16 }
17
18 model {
19   //priors
20   y0_perf ~ normal(0,5);
21   yinf_perf ~ normal(0,5);
22   y0_diff ~ normal(0,5);
23   yinf_diff ~ normal(0,5);
24   rate ~ normal(0,5);
25
26   sigma_perf ~ normal(0,5);
27   sigma_diff ~ normal(0,5);
28
29   //likelihood
30   for(i in 1:Nsubj){

```

```

31   for(j in 1:Nobs){
32     real predicted_perf = yinf_perf-(yinf_perf*y0_perf)*exp(-rate*j);
33     real predicted_diff = yinf_diff-(yinf_diff*y0_diff)*exp(-rate*j);
34
35     perf[i,j] ~ normal(predicted_perf,sigma_perf);
36     diff[i,j] ~ normal(predicted_diff,sigma_diff);
37
38   }
39 }
40 }

```

As can be seen, the implementation of this model is quite simple. Equations are specified that calculate the predicted values for performance and perceived difficulty (lines 32-33). Finally, the model evaluates the likelihood of each observed value, given the relevant predicted value and residual standard deviation parameter (lines 35-36).

Model Evaluation

Once the researcher has specified an appropriate model and confirmed that the model has converged, he or she will need to evaluate whether the model provides a satisfactory account of the data. Parameter estimates are only meaningful to the extent that the model is a good description of the phenomenon being investigated. If the model is a poor approximation of the data, then information contained in the parameter estimates will not be representative of the process the researcher is trying to model. In such cases, the researcher may need to respecify the model in order to improve its ability to account for the empirical observations.

In this section, we discuss tools that researchers can use to help determine whether a model provides a satisfactory description of the data. We must emphasize that there is no one-size-fits-all approach to model evaluation. There are often strong theoretical reasons to prefer one model over another that need to be considered. Here, we simply address a few tools that researchers have at their disposal to facilitate the evaluation process.

Visual Inspection of Model Fit

One of the simplest, yet most powerful methods of evaluating the extent to which a model adequately describes the empirical trends is by visualizing the predictions of the model in the context of the data (Heathcote, Brown, & Wagenmakers, 2015). This approach is particularly useful for ruling out poorly performing models, because mismatches between the model and the data are often very obvious.

In a Bayesian model, the model predictions are generated by sampling from the posterior distribution on the model parameters and, for each sample generated, calculating the predicted value of the outcome variable(s). This results in a set of samples that form a distribution on each outcome variable. This distribution is commonly referred to as the *posterior predictive* distribution, because it represents the predicted distribution on the outcome variable(s) that is implied by the posterior distribution on the model parameters.

For the exponential learning model described above, the posterior predictive distribution can be obtained by randomly generating a large set of samples from the posterior distribution on the seven model parameters. Note that the distributions on the individual parameters are not independent, so the same sample must be used for all parameters. For example, if the seventh sample in the

MCMC chain is drawn for one parameter, the seventh sample in the chain must be used for all parameters⁶. For each sample generated, the predicted performance and perceived difficulty for each observation in the time series are calculated based on the sampled parameters. Once this process has been repeated for every sample in the set, the posterior predictive distributions can be examined.

Figure 7 shows summary data from the example dataset superimposed over the posterior predictive distributions from the exponential learning model. This visualization makes it easy to examine the fit of the model predictions to the empirical trends. As can be seen in the left panel, the model reproduces the positive, decelerating trend in performance over time. However, the overlap between the posterior predictive distributions and the observed data is far from perfect. There are many cases where the standard error of the observed mean is outside of the model's 95% credible interval, which is indicated by the gray ribbon. These cases represent instances where the model provides a poorer account of the empirical observations. As can be seen in the right panel of Figure 7, the model does a somewhat better job of accounting for perceived difficulty, though there are still a few instances where the model prediction does not quite correspond with the empirical observation.

⁶Technically, the posterior distribution obtained is a joint distribution in which the number of dimensions is equal to the number of parameters in the model. The posteriors on the individual parameters that we have been considering thus far represent the marginal distribution on each parameter.

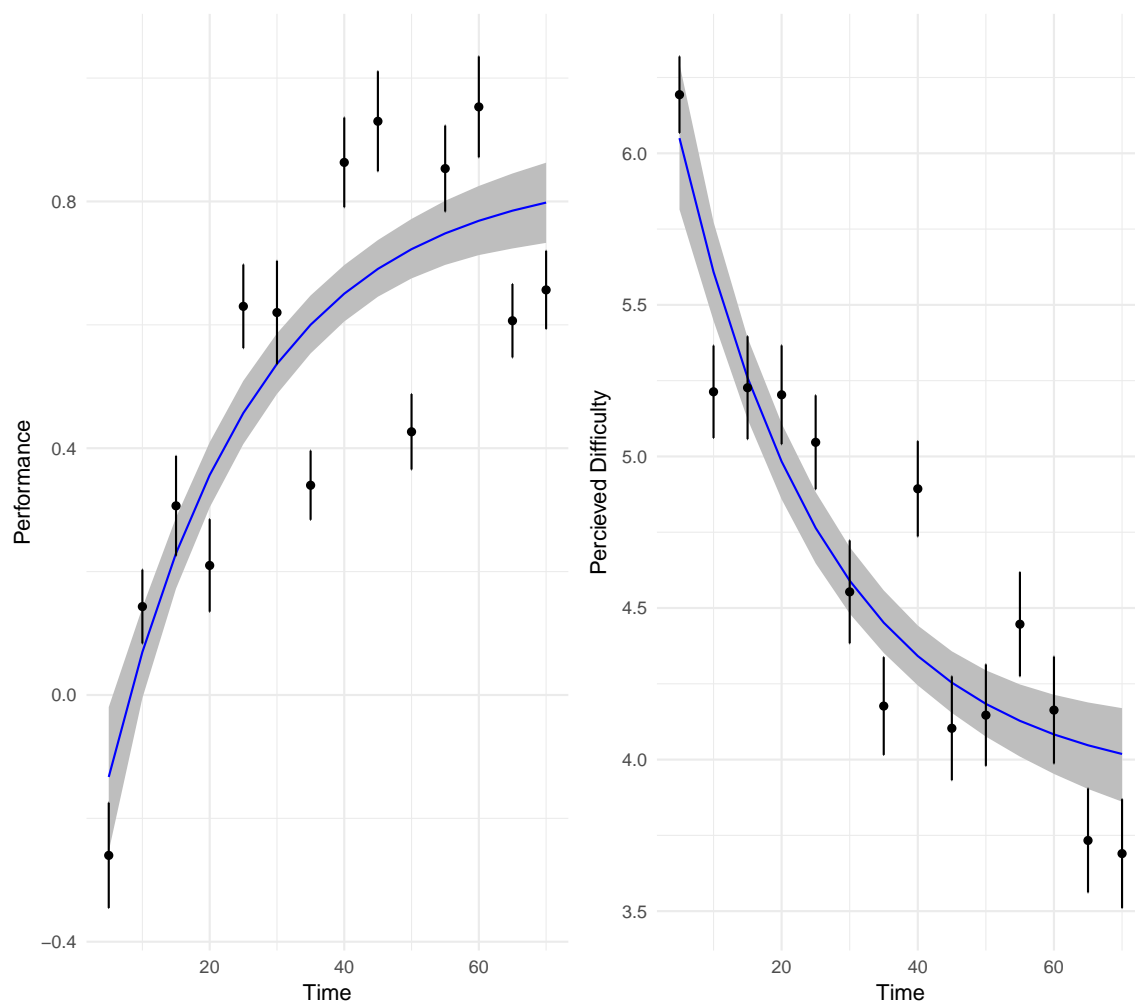


Figure 7. Summary data from the example dataset and posterior predictive distributions from the exponential learning model. The black dots and bars represent the observed means and standard errors respectively. The blue line and gray ribbon represent the mean and 95% credible interval of the posterior predictive distributions respectively. Data and model predictions were summarized for each 5-minute time window.

It is important to note that the standards for model-data correspondence are likely to be highly context-dependent. In basic laboratory research, where there is a high degree of experimental control, it is often expected that model predictions provide an extremely close fit to the data. In such contexts, even minor discrepancies between the posterior predictive distributions and the experimental data may be a sign that the model is not an adequate explanation of the phenomenon being investigated. By contrast, in observational or field studies, where there is more noise, discrepancies between the model and the data may be more tolerable. In general however, researchers should be particularly wary of cases where the model produces a qualitatively different trend than the one observed in the data. For example, if the experiment showed that improvements in performance *increased* over time, this would be a very strong indication that the model does not provide a good explanation, and should therefore be rejected.


The final point we wish to make regarding model visualization is that this tool can be used for evaluating any model that makes predictions, not just Bayesian models. In fact, we argue that visually inspecting the fit of a model to the data should be an essential practice when modeling longitudinal data. Without knowing whether a model adequately characterizes the process that is being investigated, it is impossible to know whether parameters from that model can be interpreted meaningfully. Failure to conduct a visual inspection may therefore lead to inappropriate conclusions.

Quantitative Model Comparison

Even if a visual inspection reveals that the predictions of one's model correspond closely with empirical data, there may still be another model that provides an even better description. It is therefore important to directly compare plausible alternative models in order to rule out competing accounts. For example, although the exponential model described above is a popular method for describing the learning process, other models have been proposed (e.g., the power model; Heathcote et al., 2000). Only through directly comparing the ability of the different models to account for empirical observations can the evidence in favor of each model be quantified.

Although visual inspection of model-data fit can be helpful for comparing models, it is usually not sufficient for discriminating between them. If more than one model produces a close fit to the data, the differences in model-data fit may be too subtle to be picked up visually. Moreover, evaluation based on visual fit alone ignores the other key question that needs to be considered: parsimony. If two models fit the data to a similar extent, but differ in terms of complexity, the simpler model should be preferred on the grounds of parsimony (Myung & Pitt, 1997; Vandekerckhove et al., 2015).

Several approaches to Bayesian model comparison have been proposed that quantify the tradeoff between fit and parsimony. The standard solution for Bayesian model comparison is the Bayes factor (Jeffreys, 1935; Kass & Raftery, 1995). The Bayes factor refers to the ratio of the probabilities of the data under each model, which provides an index of the relative evidence delivered by the data for one model against an alternative that takes into account both fit and model complexity (Kruschke & Liddell, 2018).

There are some challenges associated with the use of the Bayes factor however (Vandekerckhove et al., 2015). First, the Bayes factor can be difficult to obtain when comparing complex models. Calculating the Bayes factor requires the likelihood of the data under the model to be integrated across the entire parameter space, which can be computationally demanding for models with many parameters. This is especially true for models that are estimated using  MCMC methods. In recent years, methods have been introduced for approximating Bayes factor for models estimated via MCMC methods (e.g., Evans & Brown, 2018; Gronau, 2017; L. Wang & Meng, 2016). However, these methods are very computationally demanding.

Another challenge associated with use of the Bayes factor is prior sensitivity. The Bayes factor is often influenced by the researcher's choice of priors (Rouder et al., 2009). This is less problematic for standard, off-the-shelf, statistical tests, where considerable thought has gone into the recommended "default" priors (e.g., see JASP Team, 2018). However, when working with models that are novel and/or more complex, the researcher may be less confident in their choice of priors because a default specification will likely not exist. In such cases, one can assess the robustness of the results by systematically examining the impact of different priors on the Bayes factor obtained. This process is referred to as prior sensitivity analysis (e.g., Wagenmakers, Wetzels, Borsboom, & van der Maas, 2011). It is worth noting that prior sensitivity is not necessarily a bad thing. It has

been argued that priors are often an important aspect of the underlying theory and are therefore an important consideration when evaluating a model (Vanpaemel, 2010).

Several other approaches to quantitative model comparison have been proposed (Gelman, Hwang, & Vehtari, 2014; Piironen, Vehtari, Piironen, & Fi, 2017). Two examples are leave one out cross validation (LOO-VC; Geisser & Eddy, 1979) and the Watanabe-Akaike Information Criterion (also known as the 'Widely applicable information criterion'; Watanabe, 2010). These indices quantify the ability of a model to predict new data, whereas the Bayes factor addresses the evidence for each model given by the obtained data (and the priors). The LOO-CV index and WAIC are interpreted in a similar manner to AIC and BIC, where a lower value indicates a better trade off between fit and parsimony. It should be noted that these methods have not escaped criticism (e.g., Etz & Vandekerckhove, 2018)

Discussion

References

- Ballard, T., Yeo, G., B. Vancouver, J., & Neal, A. (2017). The dynamics of avoidance goal regulation. *Motivation and Emotion*, 41, 1–10. doi: 10.1007/s11031-017-9640-8
- Ballard, T., Yeo, G., Vancouver, J. B., & Neal, A. (2017). The dynamics of avoidance goal regulation. *Motivation and Emotion*. doi: 10.1007/s11031-017-9640-8
- Barker, D. H., Rancourt, D., & Jelalian, E. (2014). Flexible models of change: Using structural equations to match statistical and theoretical models of multiple change processes. *Journal of Pediatric Psychology*, 39, 233–245. doi: 10.1093/jpepsy/jst082
- Bartlema, A., Lee, M., Wetzels, R., & Vanpaemel, W. (2014). A Bayesian hierarchical mixture approach to individual differences: Case studies in selective attention and representation in category learning. *Journal of Mathematical Psychology*, 59(1), 132–150. Retrieved from <http://dx.doi.org/10.1016/j.jmp.2013.12.002> doi: 10.1016/j.jmp.2013.12.002
- Bliese, P. D., & Ployhart, R. E. (2002). Growth modeling using random coefficient models: Model building, testing, and illustrations. *Organizational Research Methods*, 5, 362–387. Retrieved from <http://orm.sagepub.com/cgi/doi/10.1177/109442802237116> doi: 10.1177/109442802237116
- Boehm, U., Marsam, M., Matzke, D., & Wagenmakers, E.-J. (n.d.). On the importance of avoiding shortcuts in applying cognitive models to hierarchical data. *Behavior Research Methods*.
- Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal Of Statistical Software*, 76, 1–32. Retrieved from <http://mc-stan.org/users/documentation/>
- Clark, D. A., Nuttall, A. K., & Bowles, R. P. (2018). Misspecification in latent change score models: Consequences for parameter estimation, model evaluation, and predicting change. *Multivariate Behavioral Research*, 53, 172–189. doi: 10.1080/00273171.2017.1409612
- Davis, J. P., Eisenhardt, K. M., & Bingham, C. B. (2007, apr). Developing theory through simulation methods. *Academy of Management Review*, 32, 480–499. Retrieved from <http://amr.aom.org/cgi/doi/10.5465/AMR.2007.24351453> doi: 10.5465/AMR.2007.24351453
- Dienes, Z. (2008). Bayesian Versus Orthodox Statistics: Which Side Are You On? *Perspectives on Psychological Science*, 6, 274–290. Retrieved from <http://journals.sagepub.com.ezproxy.library.uq.edu.au/doi/pdf/10.1177/1745691611406920> doi: 10.1177/1745691611406920
- Dinh, J. E., Lord, R. G., Gardner, W. L., Meuser, J. D., Liden, R. C., & Hu, J. (2014). Leadership theory and research in the new millennium: Current theoretical trends and changing perspectives. *Leadership Quarterly*, 25(1), 36–62. Retrieved from <http://dx.doi.org/10.1016/j.leaqua.2013.11.005> doi: 10.1016/j.leaqua.2013.11.005

- Edwards, W., Lindman, H., & Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychological Review*, 70, 193–242.
- Estes, W. K. (1994). Toward a statistical theory of learning. *Psychological Review*, 101, 282–289.
- Etz, A., Gronau, Q. F., Dablander, F., Edelsbrunner, P. A., & Baribault, B. (2018). How to become a Bayesian in eight easy steps: An annotated reading list. *Psychonomic Bulletin and Review*, 25, 219–234. doi: 10.3758/s13423-017-1317-5
- Etz, A., & Vandekerckhove, J. (2018). Introduction to Bayesian inference for psychology. *Psychonomic Bulletin & Review*, 25, 5–34. doi: 10.3758/s13423-017-1262-3
- Evans, N. J., & Brown, S. D. (2018). Bayes factors for the linear ballistic accumulator model of decision-making. *Behavior Research Methods*, 50, 589–603. doi: 10.3758/s13428-017-0887-5
- Fitzsimons, G. M., Finkel, E. J., & VanDellen, M. R. (2015). Transactive goal dynamics. *Psychological Review*, 122, 648–673. doi: <adata-auto="ep_link"href="http://dx.doi.org.ezproxy.library.uvic.ca/10.1037/a0039654"target="_blank" id="linkhttp:dx.doi.org10.1037/a0039654" title="http://dx.doi.org.ezproxy.library.uvic.ca/10.1037/a0039654" data-title="http://dx.doi.org.ezproxy.library.uvic.ca/10.1037/a0039654">http://dx.doi.org.ezproxy.library.uvic.ca/10.1037/a0039654
- Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling*, 13, 465–486. Retrieved from <https://socialsciences.mcmaster.ca/jfox/Misc/sem/SEM-paper.pdf>
- Gee, P., Ballard, T., Yeo, G., & Neal, A. (2013). *Antecedents of affect during approach and avoidance goal striving*. Perth, Australia.
- Geisser, S., & Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74, 153–160. Retrieved from <https://www.tandfonline.com/doi/abs/10.1080/01621459.1979.10481632> doi: 10.1080/01621459.1979.10481632
- Gelman, A., Hwang, J., & Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24, 997–1016. doi: 10.1007/s11222-013-9416-2
- Grand, J. A., Braun, M. T., Kuljanin, G., Kozlowski, S. W. J., & Chao, G. T. (2016). The dynamics of team cognition: A process-oriented theory of knowledge emergence in teams. *Journal of Applied Psychology*, 101, 1353–1385.
- Gronau, Q. F. (2017). A tutorial on bridge sampling. *Journal of Mathematical Psychology*. Retrieved from https://ac.els-cdn.com/S0022249617300640/1-s2.0-S0022249617300640-main.pdf?{_}tid=7bcd01ae-c17a-11e7-9224-00000aacb35e{\&}acdnat=1509811926{_}457d241aa3a178deec34467ffe1874f0 doi: 10.1016/j.jmp.2017.09.005
- Harrison, J. R., Lin, Z., Carroll, G. R., & Carley, K. M. (2007). Simulation modeling in organizational and management research. *Academy of Management Review*, 32, 1229–1245.
- Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7, 185–207.
- Heathcote, A., Brown, S. D., & Wagenmakers, E.-J. (2015). An Introduction to good practices in cognitive modeling. In B. U. Forstmann & E. J. Wagenmakers (Eds.), *An introduction to model-based cognitive neuroscience* (pp. 25–48). New York, US: Springer.
- Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 1593–1623.
- Howardson, G. N., Karim, M. N., & Horn, R. G. (2017). The latent change score model: A more flexible approach to modeling time in self-regulated learning. *Journal of Business and Psychology*, 32, 317–334. doi: 10.1007/s10869-016-9475-4
- Hox, J., Stoel, R. D., Everitt, B. S., & Howell, D. C. (2005). Multilevel and SEM approaches to growth curve modeling. In B. S. Everitt & D. C. Howell (Eds.), *Encyclopedia of statistics in behavioral science* (pp. 1296–1305). Chichester, UK: John Wiley & Sons, Ltd. Retrieved from <http://joophox.net/publist/ebs05.pdf>
- Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge, UK: Cambridge University Press.
- Jeffreys, H. (1935). Some tests of significance, treated by the theory of probability. *Mathematical Proceed-*

- ings of the Cambridge Philosophy Society, 31, 203–222. Retrieved from <http://www.uvm.edu/pdodds/files/papers/others/everything/jeffreys1935a.pdf>
- Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.
- Joyce, J. M. (1998). A nonpragmatic vindication of probabilism. *Philosophy of Science*, 65, 575–603.
- Kaplan, D., & Depaoli, S. (2012). Bayesian structural equation modeling. In R. H. Hoyle (Ed.), *Handbook of structural equation modeling* (pp. 650–673). The Guilford Press. Retrieved from <https://www.statmodel.com/download/Kaplan{ }Depaoli.pdf>
- Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90, 773–795.
- Keller, P. S., & El-Sheikh, M. (2011). Latent change score modeling of psychophysiological data: An empirical instantiation using electrodermal responding. *Psychophysiology*, 48, 1578–1587. doi: 10.1111/j.1469-8986.2011.01225.x
- Kievit, R. A., Brandmaier, A. M., Ziegler, G., Van Harmelen, A.-L., De Mooij, S. M. M., Moutoussis, M., ... Dolan, R. J. (2017). Developmental cognitive neuroscience using latent change score models: A tutorial and applications. *Developmental Cognitive Neuroscience*. Retrieved from <https://ac.els-cdn.com/S187892931730021X/1-s2.0-S187892931730021X-main.pdf?{ }tid=53c85cf8-33ab-4a08-8818-f76a193520d9{ }&acdnat=1520383641{ }68470a2ab4fd8749f404a57add58d0cf> doi: 10.1016/j.dcn.2017.11.007
- Kozlowski, S. W. J. (2015). Advancing research on team process dynamics. *Organizational Psychology Review*, 5, 270–299. Retrieved from <http://journals.sagepub.com/doi/10.1177/2041386614533586> doi: 10.1177/2041386614533586
- Kozlowski, S. W. J., Chao, G. T., Grand, J. A., Braun, M. T., & Kuljanin, G. (2013). Advancing multilevel research design: Capturing the dynamics of emergence. *Organizational Research Methods*, 16, 581–615. Retrieved from <http://orm.sagepub.com/content/16/4/581.abstract> doi: 10.1177/1094428113493119
- Kruschke, J. K. (2010). *Doing bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press/Elsevier Science.
- Kruschke, J. K., Aguinis, H., & Joo, H. (2012). The time has come: Bayesian methods for data analysis in the organizational sciences. *Organizational Research Methods*, 15, 722–752. doi: 10.1177/1094428112457829
- Kruschke, J. K., & Liddell, T. M. (2018). Bayesian data analysis for newcomers. *Psychonomic Bulletin & Review*, 25, 155–177. doi: 10.3758/s13423-017-1272-1
- Kruschke, J. K., & Vanpaemel, W. (2015). Bayesian estimation in hierarchical models. In J. R. Busemeyer, Z. Wang, J. T. Townsend, & A. Eidels (Eds.), *The oxford handbook of computational and mathematical psychology* (pp. 279–299). Oxford, UK: Oxford University Press. Retrieved from <http://www.indiana.edu/{ }kruschke/articles/KruschkeVanpaemel2015.pdf>
- Lee, M. D. (2011). How cognitive modeling can benefit from hierarchical Bayesian models. *Journal of Mathematical Psychology*, 55, 1–7. Retrieved from <https://ac.els-cdn.com/S0022249610001148/1-s2.0-S0022249610001148-main.pdf?{ }tid=76bcc504-2f07-49cf-a13e-a597ff5193b5{ }&acdnat=1527816359{ }b6c2544e907bffffdf40f0ade4890e96f> doi: 10.1016/j.jmp.2010.08.013
- Lee, M. D., & Wagenmakers, E.-J. (2013). *Bayesian cognitive modeling: A practical course*. New York, NY: Cambridge University Press.
- Lewandowsky, S., & Farrell, S. (2011). *Computational modeling in cognition: Principles and practice*. Thousand Oaks, CA: Sage.
- Lindley, D. V. (1993). The analysis of experimental data: The appreciation of tea and wine. *Teaching Statistics*, 15, 22–25.
- Liu, Y., Mo, S., Song, Y., & Wang, M. (2015). Longitudinal analysis in occupational health psychology: A review and tutorial of three longitudinal modeling techniques. *Applied Psychology*. Retrieved from <http://doi.wiley.com/10.1111/apps.12055> doi: 10.1111/apps.12055
- Lord, R. G., Diefendorff, J. M., Schmidt, A. M., & Hall, R. J. (2010). Self-regulation at work. *Annual Review*

- of *Psychology*, 61, 543–568. doi: 10.1146/annurev.psych.093008.100314
- Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS – A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10, 325–337.
- MacCallum, R. C., & Austin, J. T. (2000). Applications of structural equation modeling in psychological research. *Annual Review of Psychology*, 51, 201–226. Retrieved from <http://www.annualreviews.org/doi/10.1146/annurev.psych.51.1.201> doi: 10.1146/annurev.psych.51.1.201
- McArdle, J. J. (2009). Latent Variable Modeling of Differences and Changes with Longitudinal Data. *Annual Review of Psychology*, 60(1), 577–605. Retrieved from <http://www.annualreviews.org/doi/10.1146/annurev.psych.60.110707.163612> doi: 10.1146/annurev.psych.60.110707.163612
- McArdle, J. J., & Prindle, J. J. (2008). A latent change score analysis of a randomized clinical trial in reasoning training. *Psychology and Aging*, 23, 702–719.
- Morey, R. D., Hoekstra, R., Rouder, J. N., Lee, M. D., & Wagenmakers, E.-J. (2015). The fallacy of placing confidence in confidence intervals. *Psychonomic Bulletin & Review*, 23, 103–123. doi: 10.3758/s13423-015-0947-8
- Morgeson, F. P., Mitchell, T. R., & Liu, D. (2015). Event system theory: An event-oriented approach to the organizational sciences. *Academy of Management Review*, 40, 515–537.
- Muthen, B. (1997). Latent variable modeling of longitudinal and multilevel data. *Sociological Methodology*, 27, 453–480. Retrieved from http://hbanaszak.mjr.uw.edu.pl/TempTxt/Muthen{_}1997{_}LatentVariableModelingOfLongitudinalAndMultilevelData.pdf doi: 10.1111/1467-9531.271034
- Myung, I. J., & Pitt, M. A. (1997). Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin & Review*, 4, 79–95.
- Neal, A., Ballard, T., & Vancouver, J. B. (2017). Dynamic self-regulation and multiple-goal pursuit. *Annual Review of Organizational Psychology and Organizational Behavior*, 4, 410–423. doi: <https://doi.org/10.1146/annurev-orgpsych-032516-113156>
- Neale, M. C., Hunter, M. D., Pritkin, J., Zahery, M., Brick, T. R., Kirkpatrick, R. M., ... Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, 81, 535–549. doi: 10.1007/s11336-014-9435-8
- Piironen, J., Vehtari, A., Piironen, B. J., & Fi, A. V. (2017). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27, 711–735. doi: 10.1007/s11222-016-9649-y
- Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In K. Hornik, F. Leisch, & A. Zeileis (Eds.), *Proceedings of the 3rd international workshop on distributed statistical computing (dsc 2003)*. Technische Universität Wien, Vienna, Austria. doi: 10.1.1.13.3406
- Rabe-Hesketh, S., Skrondal, A., & Zheng, X. (2007). Multilevel structural equation modeling. In S.-Y. Lee (Ed.), *Handbook of latent variable and related models*. Amsterdam, NL: Elsevier. Retrieved from http://www.gllamm.org/msem{_}chap{_}06.pdf
- Rosseel, Y. (2012). lavaan : An R package for structural equation modeling. *Journal Of Statistical Software*, 48, 1–20.
- Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, 12, 573–604.
- Rouder, J. N., Morey, R. D., Verhagen, J., Province, J. M., & Wagenmakers, E. J. (2016). Is there a free lunch in inference? *Topics in Cognitive Science*, 8, 520–547. doi: 10.1111/tops.12214
- Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, 16, 225–237. doi: 10.3758/PBR.16.2.225
- Rousseau, D. M., Hansen, S. D., & Tomprou, M. (2018). A dynamic phase model of psychological contract processes. *Journal of Organizational Behavior*, 1–18. doi: 10.1002/job.2284
- Sanford, K. (2014). A latent change score model of conflict resolution in couples: Are negative behaviors

- bad, benign, or beneficial. *Journal of Social and Personal Relationships*, 31, 1068–1088. doi: 10.1177/0265407513518156
- Schönbrodt, F. D., Wagenmakers, E.-J., Zehetleitner, M., & Perugini, M. (2017). Sequential hypothesis testing with Bayes Factors: Efficiently testing mean differences. *Psychological Methods*, 22, 322–339.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6, 461–464.
- Stan Development Team. (2016). *RStan: the R interface to Stan, Version 2.10.1*. Retrieved from <http://mc-stan.org/>
- Stan Development Team. (2017). *Stan Modeling Language Users Guide and Reference Manual, Version 2.17.0*. Retrieved from <http://mc-stan.org/users/documentation/index.html>{\% }0D
- Steiger, J. H. (1990). Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research*, 25, 173–180. Retrieved from https://doi.org/10.1207/s15327906mbr2502_4 doi: 10.1207/s15327906mbr2502_4
- Taylor, S. G., Bedeian, A. G., Cole, M. S., Zhang, Z., Boswell, W. R., Chandler, T. D., ... Yang, J. (2017). Developing and testing a dynamic model of workplace incivility change. *Journal of Management*, 43, 645–670. doi: 10.1177/0149206314535432
- Team, J. (2018). *JASP (Version 0.8.6)*. Retrieved from <https://jasp-stats.org/>
- Thomas, A., O'Hara, B., Ligges, U., & Sturtz, S. (2006). Making BUGS Open. *R News*, 6, 12–17.
- Thurston, L. L. (1919). The learning curve equation. *Psychological Monographs*(26), 1–51.
- Turner, B. M., Forstmann, B. U., Wagenmakers, E.-J., Brown, S. D., Sederberg, P. B., & Steyvers, M. (2013). A Bayesian framework for simultaneously modeling neural and behavioral data. *NeuroImage*, 72, 193–206. Retrieved from <http://www.sciencedirect.com/science/article/pii/S1053811913000955> doi: 10.1016/j.neuroimage.2013.01.048
- Van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to Markov Chain Monte-Carlo sampling. *Psychonomic Bulletin & Review*, 25, 143–154. doi: 10.3758/s13423-016-1015-8
- Vancouver, J. B., More, K. M., & Yoder, R. J. (2008, jan). Self-efficacy and resource allocation: support for a nonmonotonic, discontinuous model. *Journal of Applied Psychology*, 93, 35–47. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/18211133> doi: 10.1037/0021-9010.93.1.35
- Vancouver, J. B., & Purl, J. D. (2017). A computational model of self-efficacy's various effects on performance: Moving the debate forward. *Journal of Applied Psychology*, 102, 599–616.
- Vancouver, J. B., Weinhardt, J. M., & Schmidt, A. M. (2010). A formal, computational theory of multiple-goal pursuit: Integrating goal-choice and goal-striving processes. *Journal of Applied Psychology*, 95, 985–1008. doi: 10.1037/a0020628
- Vancouver, J. B., Weinhardt, J. M., & Vigo, R. (2014). Change one can believe in: Adding learning to computational models of self-regulation. *Organizational Behavior and Human Decision Processes*, 124, 56–74. doi: 10.1016/j.obhdp.2013.12.002
- Vandekerckhove, J., Matzke, D., & Wagenmakers, E.-J. (2015). Model comparison and the principle of parsimony. In J. Busemeyer et al. (Eds.), *Oxford handbook of computational and mathematical psychology* (pp. 300–317). Oxford, UK: Oxford University Press.
- Vandekerckhove, J., Rouder, J. N., & Kruschke, J. K. (2018). Editorial: Bayesian methods for advancing psychological science. *Psychonomic Bulletin & Review*, 25, 1–4. doi: 10.3758/s13423-018-1443-8
- Vanpaemel, W. (2010). Prior sensitivity in theory testing: An apologia for the Bayes factor. *Journal of Mathematical Psychology*, 54, 491–498. doi: 10.1016/j.jmp.2010.07.003
- Vincent, B. T. (2016). Hierarchical Bayesian estimation and hypothesis testing for delay discounting tasks. *Behavior Research Methods*, 48, 1608–1620. doi: 10.3758/s13428-015-0672-2
- Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems of p values. *Psychonomic Bulletin & Review*, 14, 779–804.
- Wagenmakers, E.-J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Love, J., ... Morey, R. D. (2018). Bayesian inference for psychology. Part I: Theoretical advantages and practical ramifications. *Psychonomic*

- Bulletin & Review*, 25, 35–57. doi: 10.3758/s13423-017-1343-3
- Wagenmakers, E.-J., Morey, R. D., & Lee, M. D. (2016). Bayesian benefits for the pragmatic researcher. *Current Directions in Psychological Science*, 25, 169–176. doi: 10.1177/0963721416643289
- Wagenmakers, E.-J., Wetzels, R., Borsboom, D., & van der Maas, H. L. J. (2011). Why psychologists must change the way they analyze their data: The case of psi: comment on Bem (2011). *Journal of Personality and Social Psychology*, 100, 426–432. doi: 10.1037/a0022790
- Wang, L., & Meng, X.-L. (2016). Warp bridge sampling: The next generation. *arXiv preprint*. Retrieved from <https://arxiv.org/pdf/1609.07690.pdf>
- Wang, M., Beal, D. J., Chan, D., Newman, D. A., Vancouver, J. B., & Vandenberg, R. J. (2017). Longitudinal Research: A Panel Discussion on Conceptual Issues, Research Design, and Statistical Techniques. *Work, Aging and Retirement*, 3(1), 1–24. Retrieved from <https://academic.oup.com/workar/article-lookup/doi/10.1093/workar/waw033> doi: 10.1093/workar/waw033
- Wang, M., Zhou, L., & Zhang, Z. (2016). Dynamic Modeling. *Annual Review of Organizational Psychology and Organizational Behavior*, 3(1), 241–266. Retrieved from <http://www.annualreviews.org/doi/10.1146/annurev-orgpsych-041015-062553> doi: 10.1146/annurev-orgpsych-041015-062553
- Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11, 3571–3594.
- Weinhardt, J. M., & Vancouver, J. B. (2012, oct). Computational models and organizational psychology: Opportunities abound. *Organizational Psychology Review*, 2, 267–292. Retrieved from <http://opr.sagepub.com/lookup/doi/10.1177/2041386612450455> doi: 10.1177/2041386612450455
- Yeo, G. B., & Neal, A. (2004, apr). A multilevel analysis of effort, practice, and performance: effects of ability, conscientiousness, and goal orientation. *The Journal of applied psychology*, 89(2), 231–47. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/15065972> doi: 10.1037/0021-9010.89.2.231
- Zhou, L., Wang, M., Chang, C. H., Liu, S., Zhan, Y., & Shi, J. (2017). Commuting stress process and self-regulation at work: Moderating roles of daily task significance, family interference with work, and commuting means efficacy. *Personnel Psychology*, 70, 891–922. doi: 10.1111/peps.12219