A Tutorial on Cognitive Modeling for Cognitive Aging Research

Nathaniel R. Greene[1] & Stephen Rhodes[2]

[1] University of Missouri

[2] Rotman Research Institute

Draft: 06 March 2021

Author Note

Data and code available at https://github.com/stephenrho/cognitive-aging-modeling Both authors contributed equally; author order is alphabetical. This paper was written with the papaja package (Aust & Barth, 2020). The authors thank Alicia Forsberg and John Scofield for helpful feedback.

Correspondence concerning this article should be addressed to Nathaniel R. Greene, 9J McAlester Hall, Department of Psychological Sciences, University of Missouri, Columbia, MO 65211. E-mail: ngreene@mail.missouri.edu

Abstract

Cognitive aging researchers are interested in understanding how cognitive processes change in old age, but the relationship between hypothetical latent cognitive processes and observed behavior is often complex and not fully accounted for in standard analyses (e.g., ANOVA). Cognitive models formalize the relationship between underlying processes and observed behavior and are more suitable for identifying what processes are associated with aging. This article provides a tutorial on how to fit and interpret cognitive models to measure age differences in cognitive processes. We work with an example of a two choice discrimination task and describe how to fit models in the highly flexible modeling software Stan. We describe how to use hierarchical modeling to estimate both group and individual effects simultaneously, and we detail model fitting in a Bayesian statistical framework, which, among other benefits, enables aging researchers to quantify evidence for null effects. We contend that more widespread use of cognitive modeling among cognitive aging researchers may be useful for addressing potential issues of non-replicability in the field, as cognitive modeling is more suitable to addressing questions about what cognitive processes are (or are not) affected by aging.

Keywords: Signal Detection Theory; Cognitive Aging; Mathematical Models

A Tutorial on Cognitive Modeling for Cognitive Aging Research

Cognitive aging researchers aim to understand how cognitive processes change across the adult lifespan, particularly into old age. However, it is impossible to observe a "cognitive process" directly; rather, we must infer cognitive processes from observed behavior (e.g., accuracy, reaction time). One way this is commonly done is by submitting raw data or aggregated data (e.g., mean RT by condition) to a *t*-test, ANOVA, or mixed effects model and inferring process(es) through differences between conditions and their interaction with age. However, this interpretation relies on a simple mapping of cognitive processes onto observed behavior, which is unlikely to be the case and can result in incorrect conclusions about the source of age-differences (see Salthouse, 2000). In contrast to standard analysis, *cognitive models* formalize the relationship between hypothesized underlying processes and observed behavior by way of parameters that reflect the contributions of latent cognitive processes, and are therefore more suitable for identifying what processes are (or are not) associated with aging.

This paper provides a tutorial on fitting and interpreting cognitive models in the context of cognitive aging research. There are *many* types of cognitive models, suitable for different tasks, and it would be impossible for us to exhaustively describe and implement all (or even a substantial subset) of them. Nevertheless, some of the models that may be most suitable for cognitive aging researchers are models of accuracy or reaction time, given that these are two of the most widely reported outcomes in studies of cognitive aging (see section "How Widespread is Cognitive Modeling in Cognitive Aging Research?"). A popular model of reaction time is the drift diffusion model (Ratcliff, 1978), and popular models of accuracy include multinomial processing tree (MPT) models and signal detection theory (SDT) models. There are also many other models, including computational models, which attempt to formalize the underlying nature of a cognitive process (e.g., storage of features in memory as vectorized representations), though these are less common. In this tutorial, we use an example of a two choice discrimination task, given that this kind of task is widely used in cognitive aging research. We use an example of a SDT analysis, and in the supplement we also discuss MPT analyses. We have chosen to focus on highly flexible modeling software (Stan), and the tutorial below is written with the intention of generalizing to other areas, so that readers will be able to start implementing models that are relevant to their particular research area.[1] Finally, the tutorial focuses on implementing hierarchical models under a Bayesian statistical framework. Hierarchical modeling allows one to model group level differences as well as individual differences simultaneously, and Bayesian model comparison allows researchers to go beyond null hypothesis significance testing to quantify the strength of evidence for or against age-related changes to particular cognitive processes (these benefits have been discussed extensively elsewhere; see, e.g., Gelman et al., 2014; Kruschke, 2015; Lee & Wagenmakers, 2014).

Although the following tutorial is intended to be an aid for novice modelers, and a refresher for more expert modelers, we acknowledge upfront that learning cognitive modeling will take time and patience. This tutorial alone may not be sufficient for learning everything about cognitive modeling, given the breadth of research on this topic. To that end, we discuss some of the best practices for researchers to consider as they begin their journey with cognitive modeling (see section “Recommendations on Best Practices and Further Readings”). We also acknowledge that mastering cognitive modeling will likely require learning some rather technical concepts. Although we aim to keep this tutorial as accessible as possible for the novice reader, it would be a disservice to avoid technical details where they are necessary. Nevertheless, we aim to break these technical details down

[1] Indeed there are existing `R` packages that offer user-friendly implementations of hierarchical SDT models, including `bhsdtr` (Paulewicz & Blaut, 2020) and `brms` (Bürkner (2017); see https://vuorre.netlify.app/post/2017/10/09/bayesian-estimation-of-signal-detection-theory-models-part-1/). In covering Stan we want to show readers how to implement models from scratch for situations where a user-friendly implementation does not exist or where more control over the details is desired.

into simpler terms to ease with understanding. Ultimately, it is our hope that this tutorial will serve as a useful guide and a great starting point for cognitive aging researchers who are interested in learning cognitive modeling and applying such models to their own data.

What are the Benefits of Cognitive Modeling?

Perhaps the simplest and most compelling reason to use a model that formalizes the relationship between processes and behavior is that it gets us closer to what we (as cognitive psychologists) are interested in. Consider the following example. The finding that older adults are less likely to recall the source of information and that they are less likely to rely on recollective phenomenology when recalling source information is interesting in and of itself, but thinking through the possible underlying processes and formalizing them in, for example, an MPT model begins to elucidate *why* this is the case. For example, (Boywitt et al., 2012), who used the MPT model of multidimensional source memory from (Meiser & Bröder, 2002), found that older adults were less likely to bind together source dimensions of an episode and did not rely on the retrieval of perceptual source features to distinguish between recollective and familiarity-based responses regarding memory for the source of an item. Whereas younger adults only maybe “remember” responses (corresponding to a recollective experience) when they retrieved perceptual source details, older adults were equally likely to make “remember” or “know” responses (corresponding to a reliance on familiarity), regardless of the extent to which they retrieved perceptual source features, as estimated from the MPT model.

Some may argue that the *why* question is addressed by devising clever manipulations that attempt to isolate particular sources of difference. We agree; a model by itself is not enough, and experimental manipulation is necessary for testing ideas about age differences in cognition. However, a robust and replicable effect does not necessarily translate into insight regarding underlying processes. This is perhaps best understood when it comes to interaction effects, where changing the scale of measurement can change the conclusions that one is able to draw (Loftus, 1978; Rhodes, Cowan, Parra, & Logie, 2019; Wagenmakers, Krypotos, Criss, & Iverson, 2012). A recent example from the cognitive aging literature comes from Archambeau, Forstmann, Van Maanen, and Gevers (2020), who assessed age differences in susceptibility to proactive interference. Specifically, they looked at the recent-probes task in which differences between younger and older participants in errors and reaction time for lures are greater when those lures were studied on the previous trial relative to non-recent lures. These error and RT findings have been interpreted as showing that older adults have a specific difficulty in inhibiting recently studied, but no longer relevant, information. However, Archambeau et al. (2020) applied the drift diffusion model (Ratcliff, 1978) and found evidence that older adults were slower than younger adults in the rate of information accumulation (the "drift rate" parameter) but no evidence that this was especially true for recent lures versus non-recent lures. Thus, what appears as evidence for a greater susceptibility to proactive interference when looking at the raw data can be accounted for without that assumption at the level of cognitive processes (Rotello, Heit, & Dubé, 2015 provide other examples of replicable results that have been reassessed with model-based analyses).

Another advantage of cognitive modeling is that it can provide a more unified measurement approach to reliably characterize age differences across tasks that may vary from study to study but which purport to measure the same underlying processes. As a recent example, consider the work of Loitile and Courtney (2015) who applied a SDT analysis to a behavioral pattern separation paradigm. In these paradigms, participants are tasked with discriminating old items from similar lures and novel items, and the typical finding is that older adults are especially impaired at old-similar discrimination but not at old-new discrimination (Stark, Yassa, Lacy, & Stark, 2013). However, as Loitile and Courtney (2015) noted, old-similar discrimination has been analyzed neither consistently across studies nor in a principled way that allows for an accurate comparison with old-new discrimination. One consequence of a non-standardized measurement approach is that it can lead to somewhat counterintuitive conclusions. In a study by Reagh and Yassa (2014), the researchers found that repeat-encoding of an item (three times versus one time) led to enhanced discrimination of that item from new items (old-new discrimination) but surprisingly *worse* discrimination of that item from similar items (old-similar discrimination), compared with old-similar discrimination for once-encoded items. A close look at the formulas Reagh and Yassa (2014) used reveals that they essentially compared the correct rejection rates for similar lures to once- and thrice-encoded items (i.e., they did not use a cognitive model to measure memory strength). In other words, their formula addressed whether accuracy for *similar lures* was better for lures to once-encoded than thrice-encoded old items, rather than assessing memory for lures *relative* to old items. Loitile and Courtney (2015) applied a SDT analysis to data from a task that was identical to the one used by Reagh and Yassa (2014) and found that repeat encoding actually enhanced old-similar discrimination, a much more expected (i.e., less counterintuitive) finding. As such, the conclusion from Reagh and Yassa (2014) that repeat encoding results in poorer old-similar discrimination was misleading, and such misleading conclusions can affect theories about what processes are affected by repeat encoding and whether repeat encoding may benefit or further harm older adults' old-similar discrimination (though the application to aging was not examined in the study by Reagh and Yassa (2014), but similar paradigms are commonly used in aging research).

None of this is to say that cognitive modeling is a magic key that will inevitably unlock the secrets of cognitive aging. For conclusions to be accurate it is important that the model applied is able to mimic the empirical effects seen in the raw data. Therefore, researchers should refer to the wider literature on the adequacy of particular models in particular settings. Often there will be several competing models that account for behavior with different underlying assumptions, in which case researchers may consider comparing models to see which provides the best account of their data set. We cover how to assess model fit as well as how to compare models in this tutorial.

Why Use Hierarchical Bayesian Estimation?

When fitting a cognitive model typically the researcher must decide whether to fit the model to the data set as a whole, ignoring individual variability, or to each individual separately, in which case often a large number of trials is needed and additional analyses are required to look at group level trends. A hierarchical model is a compromise between these two extremes in which individual level parameters are constrained by a group level distribution [gelman and hill book] and these models are handled especially well under a Bayesian framework [BDA and others]. Hierarchical models are also easily extended to account for sources of variability beyond that attributable to participant differences. For example, the items or stimuli used in a particular experiment can be an additional source of variation in performance and failure to take this into account can increase false discovery (Clark, 1973). This tutorial covers how to include such item effects in a cognitive model.

The ability to simultaneously model group and individual effects, as well as the ability to account for stimulus variability is particularly important in the context of cognitive modeling (see Lee, 2011 for more discussion of the benefits of hierarchical Bayesian methods for cognitive modeling). Specifically, cognitive models are often non-linear in that hypothetical processes are mapped onto observed behavior through non-linear functions. For example, in signal detection theory sensitivity, or d', is mapped onto accuracy through the “S” shaped normal cumulative distribution function. Failure to account for participant and/or item variability can cause systematic bias in parameter estimates that are not correctable via replication (as the bias is systematic; see, e.g., Estes, 1956; Heathcote, Brown, & Mewhort, 2000; Morey, Pratte, & Rouder, 2008; Pratte & Rouder, 2012; Rouder & Lu, 2005).[2]

[2] In the linear case, failure to account for participant or item variability increases the type I error rate but does not produce systematic bias, therefore replication would act as intended: as a check on false discovery. See Rouder and Lu (2005) for a detailed discussion.

How Widespread is Cognitive Modeling in Cognitive Aging Research?

We surveyed all articles published between the years 2011 and 2020 in the journal *Psychology and Aging* to get a sense of the prevalence of cognitive modeling in cognitive aging research. For each year, we searched through all published articles which researched some aspect of cognition (e.g., memory, inhibition, learning, etc.). Any study that included some measure of cognition was counted, and for published papers reporting multiple studies (e.g., papers with two experiments), we counted each study in the paper separately. Although there was some variability in the number of published cognitive aging studies per year, on average there were about 56 (M = 55.70, SD = 20.58).

For each study, we recorded what the dependent variables (DV) were and how these DV's were analyzed. We assume that all cognitive studies were interested in making some claims about cognition, but we only regarded DV's which attempted to measure a latent process as a strict measure of a latent process. For example, many studies reported reaction time as the DV, but most of these studies only considered mean or median differences in reaction time. Mean differences in reaction time are not a measure of a latent process difference between young and older adults. An example of a reaction time DV that satisfies being a measure of a latent process is a drift rate parameter, as this can be interpreted as the rate of accumulation of evidence (a latent process). By far the most commonly reported DV was accuracy, which was defined in many different ways, depending on the task. Examples included proportion correct, proportion of items recalled, d', and proportion hits minus proportion false alarms. We regarded these DV's as measures of latent processes if they attempted to isolate sensitivity from bias. Therefore, while we did not regard proportion correct as a measure of a latent process, we did consider the corrected recognition metric (proportion of hits minus proportion of false alarms) as a measure of a latent process (Snodgrass & Corwin, 1988). We also coded whether each study which measured a latent process also fit a cognitive model. The distinction here is that some DV's which measured a latent process were based on parameters of a cognitive model (such as a SDT, MPT, diffusion, or computational model), whereas others were simply transformations of the data (e.g., proportion hits minus proportion false alarms).

In Figure 1 (panel a) we plotted the proportion of studies per year in *Psychology and Aging* in which the DV was a measure of a latent process. As depicted, this proportion has been relatively low, though it has been rising slightly in more recent years. Nevertheless, it has never been more than 50% of cognitive aging studies in a given year. In other words, more than half of all cognitive aging studies published in one of the field's top journals do not report age differences on an outcome that is a measure of a latent process! In Figure 1 (panel b) we plotted, from those studies for which the DV was a measure of a latent process, the proportion of those studies which fit a cognitive model. The picture here is somewhat less clear, as there have been rises and falls on a year-to-year basis in terms of the number of cognitive aging studies which have used a cognitive model. Also, while it is true that in some years this proportion has been greater than 50%, this is still only 50% (or more) of *less than half* of all cognitive aging studies published in *Psychology and Aging* that year.

Clearly, cognitive modeling remains the exception, rather than the norm, in cognitive aging research. In the next section, we describe how to fit a SDT account of data from a discrimination task (e.g., a recognition memory study) with a focus on testing age differences in underlying parameters. Specifically, we describe how to implement these as hierarchical models that simultaneously model individual participant and group level effects.

Modeling Age Differences in Choice Discrimination

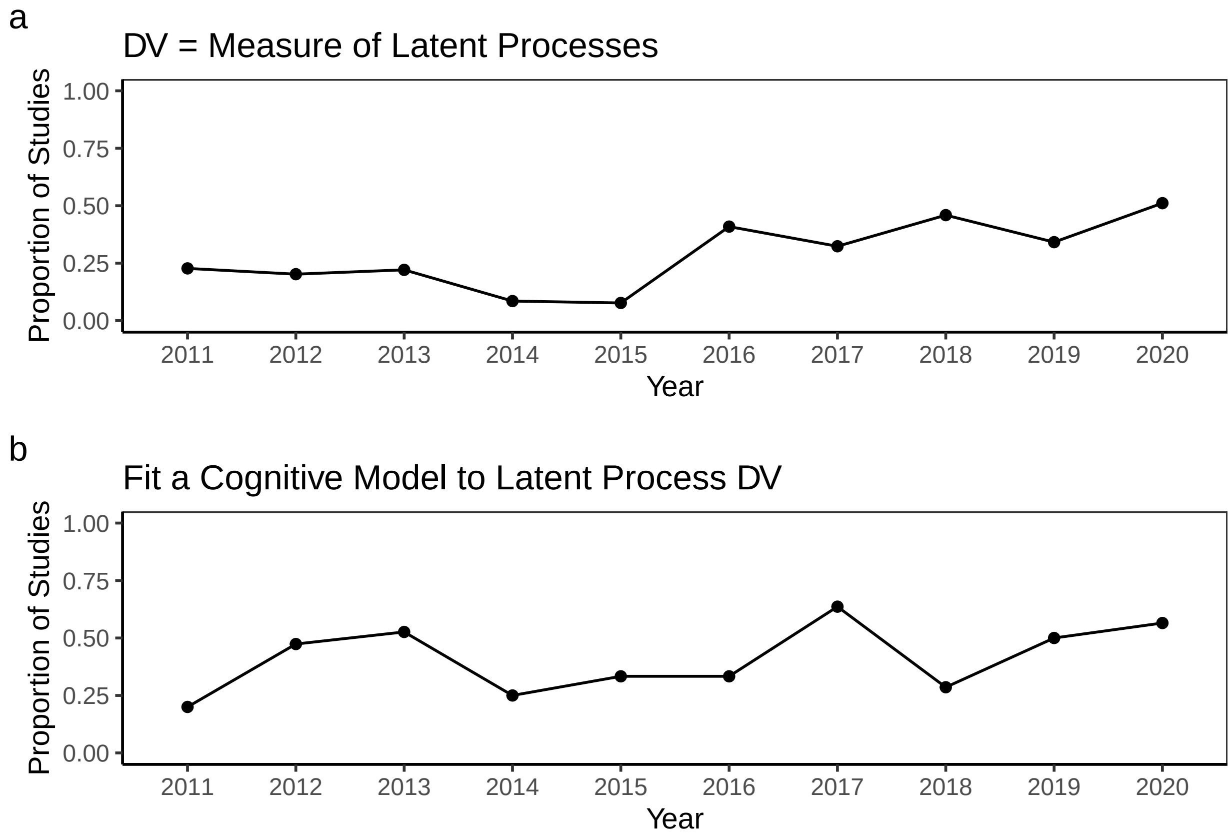

Figure 1. a: Proportion of cognitive aging studies per year in *Psychology and Aging* in which the dependent variable (DV) was a measure of a latent cognitive process. b: Conditional proportion of studies per year which, given that the DV was a measure of a latent cognitive process, used a cognitive model

Description of the model

SDT (Green & Swets, 1966; Tanner Jr & Swets, 1954) provides a framework to understand performance in various situations where the objective is to discriminate signal from noise (see Macmillan & Creelman, 2005 for a comprehensive introduction). One of the most commonly used tasks in cognitive aging studies is the yes-no experiment, in which participants must decide whether a particular trial contains signal (e.g., a previously seen stimulus, the presence of a particular target stimulus) or just noise (e.g., a new stimulus, a non-target).

To model this situation (and others), SDT assumes that each trial elicits a particular value along what is called the “decision variable”; on average, signal trials produce larger values, but, due to internal and/or external noise, the distribution of values for signal and noise trials will often overlap (which makes the discrimination imperfect). The distance between the central tendencies of these two distributions is an index of “discriminability” (or sensitivity).

In the simple yes-no case, a plausible assumption is that participants use a “criterion” to make the decision. If the value on the decision variable on a particular trial is above the criterion the response will be *yes* (i.e., signal), otherwise the response will be *no*. The familiar measures d' and c come from a particular implementation of SDT in which the underlying signal/noise distributions are assumed to be normal with equal width. This “equal-variance” assumption has been a particular source of controversy (Ratcliff, Sheu, & Gronlund, 1992; Swets, 1986a, 1986b); however, in order to test this assumption more information is needed beyond the binary yes-no decision.

One additional source of information is to ask the participant to provide a more fine grained discrimination in which they provide their level of confidence on the presence or absence of signal. This can be done with a likert-type scale ranging from sure-no to sure-yes or by first asking for an yes-no decision before asking for a level of confidence (Yonelinas & Parks, 2007). To model this, we can assume that, instead of one criterion, the observer establishes $K - 1$ criteria, where K is the number of options. Figure 2 (top panel) gives a visual depiction of this model where d is the difference between the means of the noise and signal distributions and s is the standard deviation of the signal distribution. The standard deviation of the noise distribution is fixed to 1 to provide scale (note: the exact values of the decision variable are arbitrary).

Under this model, the probability that the observer gives a particular rating is given by the area under the noise or signal distributions that falls in-between the relevant

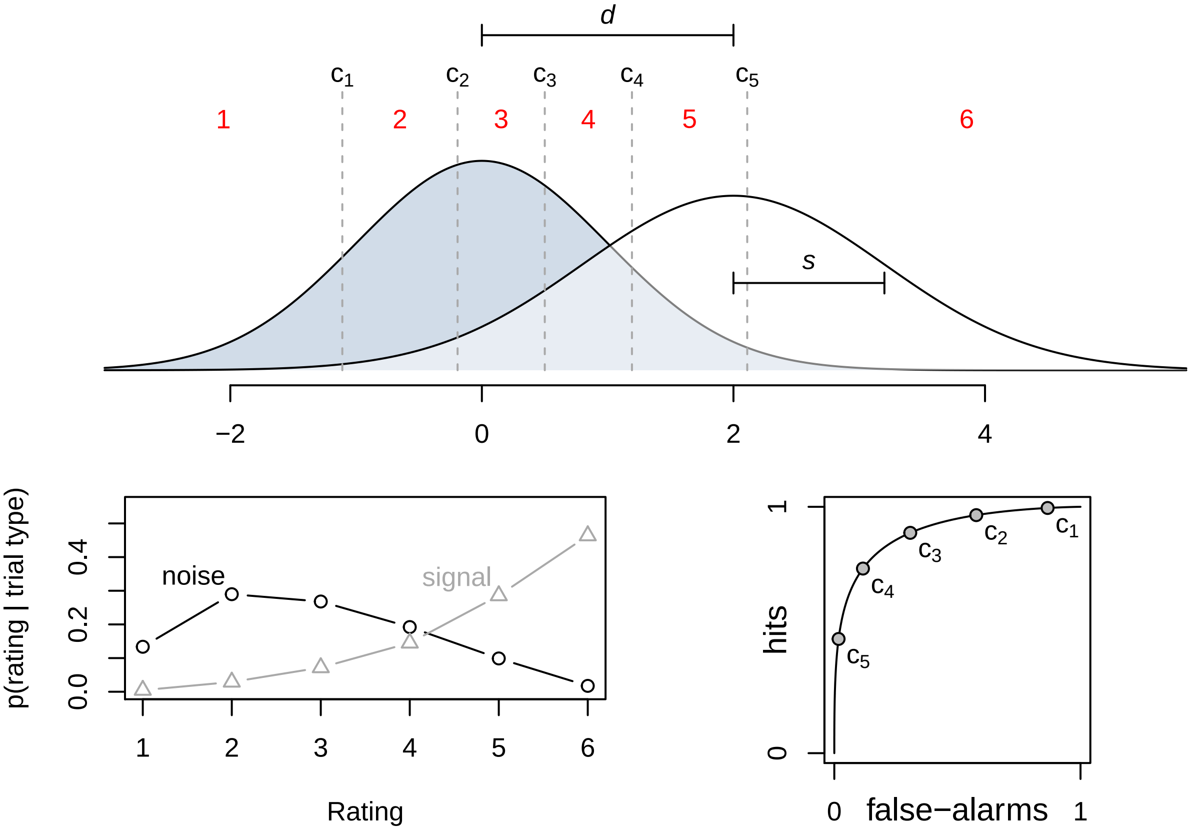

Figure 2. Top: A signal detection model for a rating experiment in which there are 6 options (1 = "sure no signal", 6 = "sure signal"). The observer is assumed to use 5 criteria and the rating chosen on a particular trial is determined by the sampled value on the decision variable (e.g., a rating of 4 would be given if the sampled value falls between thresholds 3 and 4). Bottom left: Expected proportion that each rating would be selected for noise and signal trials under the model depicted in the top panel. The proportions are determined by finding the area under the noise/signal curves between the respective criteria. Bottom right: The reciever operating characteristic (ROC) curve implied by the model. Hits are calculated by taking the cumulative sum of the response proportions on signal trials, starting at "sure signal" (i.e., rating 6). False-alarms are calculated similarly but using noise trials. The furthest left point is the response proportions for ratings of 6, the next point sums the proportions across ratings of 5 and 6, and so on. The last point is not presented as it must be (1,1).

criteria. For example, for rating 3 it is the area between criteria 2 and 3 (c_2 and c_3). Assuming the underlying distributions are normal we can use:

$$p(\text{rating} = k \mid \text{noise}) = \Phi(-c_k) - \Phi(-c_{k-1})$$
$$p(\text{rating} = k \mid \text{signal}) = \Phi\left(\frac{d - c_k}{s}\right) - \Phi\left(\frac{d - c_{k-1}}{s}\right)$$

where $c_0 = -\infty$, $c_K = \infty$, and Φ is the standard normal cumulative distribution function (`pnorm` in `R`). Predicted rating probabilities are given in the bottom left of Figure 2 along with the receiver operating characteristic (ROC) curve in the bottom right.

The parameters of the SDT model shown in Figure 2 are d, s, and the criteria $c_{1,\ldots,K-1}$. This means that using a task with more response categories requires more parameters to fit. Recently, Selker, Bergh, Criss, and Wagenmakers (2019) have proposed a more parsimonious version of the model in which, instead of having separate parameters for each criterion, we start with unbiased criteria (u) that are then scaled (a) and shifted (b) by two free parameters:

$$u_k = \log\left(\frac{k/K}{1 - k/K}\right) = \text{Logit}(k/K)$$
$$c_k = au_k + b.$$

Higher values of a spread the criteria further apart, whereas higher values of b shift the criteria to the right (more conservative).

Implementing the model

Figure 2 shows the situation where $d = 2$, $s = 1.20$, $a = 1$, and $b = 0.50$. However, these parameters likely vary across participants (differing both between young and older adults, but also within each age group) as well as with experimental manipulation. We could fit the model to each individual participant separately (e.g., via maximum likelihood), but, for the reasons outlined above, our preferred approach is to model both group- and individual-level effects simultaneously in a Bayesian hierarchical model. In this tutorial we will outline how to implement the SDT model in the probabilistic programming language, Stan (https://mc-stan.org/; Carpenter et al., 2017), which performs the MCMC sampling often required in fitting models under a Bayesian framework. Documentation and tutorials on the Stan language can be found here: https://mc-stan.org/users/documentation/tutorials. Stan can be used in conjunction with other software, such as Python, Matlab, and Stata (https://mc-stan.org/users/interfaces/), but here we will be using `R` (R Core Team, 2018) and the `rstan` package (Stan Development Team, 2018) as our interface (see the supplementary materials[3] for more information on installing the relevant software). Thus, while familiarity with `R` will be helpful, the tutorial on writing Stan models will hopefully be of use to users of other analysis software.[4] Next we will go through a series of questions that we need to address to set up a working Stan implementation of the (hierarchical) SDT model.

How are the data organized? To write the model we need a clear understanding of the structure of the data. Here we will be working with simulated data from a rating experiment in which young and old adult participants ($N = 24$ each group) rated their confidence in the presence of a signal on a scale from 1 to 6. The following `R` code allows us to see the first 6 rows:

```r
rdat = read.csv("data/example-data.csv")
head(rdat)
```

```
##   id trial group cond item signal rating
```

[3] The supplement is available here: https://github.com/stephenrho/cognitive-aging-modeling/blob/master/paper/cognitive-modeling-supplement.pdf

[4] We encourage readers to download the materials accompanying this article from https://github.com/stephenrho/cognitive-aging-modeling and use the `fit_SDT.R` script to follow the analyses described here. As the models can take a long time to fit, fitted models can also be downloaded by following instructions at the link above.

```
## 1  1     1     0    A    4      0      1
## 2  1     2     0    A    3      0      4
## 3  1     3     0    A   16      0      2
## 4  1     4     0    A   14      0      2
## 5  1     5     0    A    7      0      2
## 6  1     6     0    A   18      0      4
```

The format of these data is referred to as long (or tall), where a single observation is recorded on each row. Each row represents a trial where `id` is a participant identifier, `group` signals whether the participant is younger (Y) or older (O), `signal` codes for whether the trial was noise (0) or signal (1), and `rating` is the response. The variable `cond` codes for an additional factor of condition with two levels (A, B; 40 observations each), which we include here to demonstrate how to test for interactions between age and other variables, which are common in the cognitive aging literature. As a concrete example, this could be thought of as data from a recognition memory experiment in which each participant rated their degree of memory for particular items (from sure new to sure old) under two conditions (e.g., a levels of processing manipulation).

With this data in mind we can start to write the Stan model, which can be found in the file `SDT_m1.stan` (available at https://github.com/stephenrho/cognitive-aging-modeling). Stan models are made up of "blocks" (https://mc-stan.org/docs/2_24/reference-manual/overview-of-stans-program-blocks.html) and usually start by declaring the data. The Stan code below specifies the data necessary to fit the SDT model (the comments, following "//", describe each line).

```
data {
int<lower=0> N;               // n observations
int y[N];                     // ratings
int<lower=0> J;               // n participants
int<lower=1,upper=J> id[N];   // participant ids
matrix[N, 2] X;               // fixed design matrix
vector[N] sig_trial;          // indicator for signal (1) or noise (0) trial
int<lower=2> K;               // n categories
}
```

In the `data` block one must specify the type (integer, real valued, matrix, vector) and, where appropriate, range for each item (e.g., are values below zero allowed?). Another important thing to note is that there are multiple other ways of structuring the data that could work. For example, here we have chosen to keep the rating response as a single long vector with an indicator vector that tells us whether an observation was a noise or signal trial. Instead we could have specified two vectors, one for noise trials and one for signal trials but the remaining stan code would have to be written differently to account for this.

Next, in fitting the Selker et al. (2019) version of the model, we use the number of categories, K, to create the unbiased criteria that the model will shift and scale. We use a `transformed data` block to do this:

```
transformed data {
vector[K-1] unb_c;
for (k in 1:(K-1)){
unb_c[k] = -log( (1 - (exp(log(k) - log(K))))/(exp(log(k) - log(K))) );
}
}
```

This creates a vector called `unb_c` and populates it with the $K - 1$ unbiased criteria (by cycling around the `for` loop from `1` to `K-1`, as the "`:`" produces a sequence between the specified end points). Note that we specify the type of object `unb_c` is at the beginning of the block and, as it is a vector, we can select particular elements using `unb_c[k]`. This block also highlights an important point about Stan; dividing an integer by an integer produces an integer answer, which is not what we want here. Thus, to calculate each `k/K` we must instead subtract the logarithms then take the exponent to get an non-integer answer. Note the unbiased criteria could have been calculated outside of the stan model and provided in the `data` block (but in this case we only have to supply `K`).

What are the model parameters? The parameters of the signal detection model are d, s, a and b (where the latter two determine the criteria $c_{1,\ldots,K-1}$). However, in fitting the model hierarchically there are additional parameters we need to think of. Specifically, we need to think of the group-level and individual-level effects on the SDT parameters. Taking d as an example, we can model d for each individual, j, as,

$$d_j = \beta^{(d)} + b_j^{(d)}$$
$$b_j^{(d)} \sim \text{Normal}(0,\ \tau^{(d)}),$$

where there are two parts: the group average, $\beta^{(d)}$, and individual deviations from that average, $b_j^{(d)}$, that are assumed to be normally distributed with standard deviation, $\tau^{(d)}$. An alternative way of writing this, which better captures the idea that the ds come from a "population" distribution is, $d_j \sim \text{Normal}(\beta^{(d)},\ \tau^{(d)})$. The $\sim$ (tilde symbol) means "is distributed as" and is also used by Stan to denote variables with a distribution.

As cognitive aging researchers we can extend this to model age differences in d by including an additional group-level effect:

$$d_j = \beta_0^{(d)} + \beta_1^{(d)} x_j + b_j^{(d)},$$

where x_j codes the age group of individual j, and $\beta_1^{(d)}$ is the coefficient giving the difference between groups. Thus, there are 3 parameters related to d that we estimate directly: $\beta_0^{(d)}$, $\beta_1^{(d)}$, and $\tau^{(d)}$. The individual $b_j^{(d)}$s are estimated indirectly and depend on $\tau^{(d)}$. In a highly flexible model we might apply the same approach to the other SDT parameters to account for all possible sources of age difference.

In Stan the parameters are specified in their own block, as follows:

```
parameters {
// d
vector[2] B_d;
real b_d[J];
real<lower=0> tau_d;

// c (shift [b] and scale [a])
vector[2] B_a;
real b_a[J];
real<lower=0> tau_a;

vector[2] B_b;
real b_b[J];
real<lower=0> tau_b;

// s
vector[2] B_s;
real b_s[J];
real<lower=0> tau_s;
}
```

where uppercase `B`s are the βs and lowercase `b`s are the bs and the letter after the underscore refers to the corresponding SDT parameter. The two β parameters are specified as a vector.

Next we can use a `transformed parameters` block to map these parameters onto the trial level values of d, s, a, and b. However, before we do this, we have to address a remaining issue in relating the hierarchical parameters to the parameters of the SDT model. Specifically, the SDT parameters d, s, and a are *constrained* to be positive (see Paulewicz & Blaut, 2020 for rationale on why d should be positive). However, adding the normally distributed individual-level effects to the population means allows for negative values. Thus, for constrained parameters we need a *link function* to map the hierarchical parameters onto the SDT parameters (this will be familiar to users of generalized linear models). For positively constrained parameters a common choice is to have the hierarchical parameters (the βs, bs, and τs) be on the log scale, where the exponential function is the link that maps these to their natural scale. So taking the example of d again we modify the above equation to:

$$d_j = \exp\left(\beta_0^{(d)} + \beta_1^{(d)} x_j + b_j^{(d)}\right).$$

The `transformed parameters` block starts by creating vectors to hold the SDT parameters for each observation, $1, ..., N$ (we will discuss `theta` later). The `for` loop then goes through the N observations and uses the value of the predictors contained in `X` and the participant `id`s to set things for observation i. This requires the use of indexing: `X[i,]` selects row `i` of the design matrix and this is used to calculate the dot produced with the group level parameters (i.e., $\beta_0 + \beta_1 x_i$); `b_d[id[i]]` first finds the participant `id` associated with observation i and uses this to index the individual-level parameter associated with that participant (sometimes written $b_{j[i]}$; e.g., Gelman & Hill, 2007).

```
transformed parameters {
real<lower=0> d[N]; // note that d, a, and s are constrained positive
real<lower=0> a[N];
real b[N];
real<lower=0> s[N];
vector[K-1] c[N];

simplex[K] theta[N];
```

```
for (i in 1:N){
// observation level parameters
d[i] = exp( dot_product(X[i,], B_d) +  b_d[id[i]] );
a[i] = exp( dot_product(X[i,], B_a) + b_a[id[i]] );
b[i] = dot_product(X[i,], B_b) + b_b[id[i]];
s[i] = exp( dot_product(X[i,], B_s) + b_s[id[i]] );

c[i] = a[i]*unb_c + b[i];

... // continued below
```

How do the parameters relate to predictions for each observation? As we have now specified the values of the SDT parameters for each observation, we are now ready to use them to produce predicted ratings, which is done in the second part of the `transformed parameters` block:

```
... // continued from above

// rating probabilities under SDT
if (sig_trial[i] == 1){ // signal trial
theta[i,1] = normal_cdf(c[i,1], d[i], s[i]);
for (k in 2:(K-1)){
theta[i,k] = normal_cdf(c[i,k], d[i], s[i]) - sum(theta[i,1:(k-1)]);
}
}
else { // noise trial
theta[i,1] = normal_cdf(c[i,1], 0, 1);
for (k in 2:(K-1)){
theta[i,k] = normal_cdf(c[i,k], 0, 1) - sum(theta[i,1:(k-1)]);
}
}
theta[i,K] = 1 - sum(theta[i,1:(K-1)]); // last rating probability
}
}
```

While it looks somewhat complicated, this implements the first two equations presented above to produce predicted probabilities for the $1, ..., K$ rating categories. The object `theta` contains the predicted ratings for each observation in the data set (each `theta[i,]` is defined as a unit simplex at the beginning of this block, which means the values must sum to 1).

The line starting `theta[i,1] ...` finds the cumulative probability between $-\infty$ and the first criteria, `c[i,1]`, using the `normal_cdf` function (i.e., the probability of selecting rating 1). The `d[i]*sig_trial[i]` means that if this observation is a signal trial the mean of the normal is d, whereas is it zero otherwise. The `(1 + (-1 + s[i])*sig_trial[i])` means that if the observation is a noise trial the standard deviation of the `normal_cdf` function is 1, whereas if `sig_trial` $= 1$ it will equal s. The `for` loop then works out the probability of selecting other ratings up to $K - 1$ by finding the area up to the relevant criteria and then subtracting that already accounted for by previous criteria. Finally, the probability of selecting the final rating category (the line beginning `theta[i,K] ...`) is what remains up to a total of 1.

How do we relate the predictions to the data? The `model` block brings everything together and serves two main purposes: (1) to specify the prior distribution for the model parameters and (2) to specify the likelihood function that expresses the likelihood of the observed data given the model probabilities. These things form the basis of Bayesian estimation, where the posterior distribution is proportional to the prior × the likelihood. Prior distributions reflect the degree of belief in particular parameter values before seeing new data, and further detail on the particular settings used below is given in the supplement.

```
model {
// priors
B_d[1] ~ normal(0, 1);
B_d[2] ~ normal(0, 0.5);
B_a[1] ~ normal(0, 1);
B_a[2] ~ normal(0, 0.5);
B_b[1] ~ normal(0, 2);
B_b[2] ~ normal(0, 1);
B_s[1] ~ normal(0, 0.5);
B_s[2] ~ normal(0, 0.25);

tau_d ~ cauchy(0, 1);
tau_a ~ cauchy(0, 1);
tau_b ~ cauchy(0, 2);
tau_s ~ cauchy(0, 0.5);

// individual-level deviations
b_d ~ normal(0, tau_d);
b_a ~ normal(0, tau_a);
b_b ~ normal(0, tau_b);
b_s ~ normal(0, tau_s);

// likelihood
for (i in 1:N){
y[i] ~ categorical(theta[i]);
}
}
```

The line `b_d ~ normal(0, tau_d);` captures the assumption that individual deviations from the group mean follow a normal distribution centered on zero with a standard deviation of `tau_d` (same for the other SDT parameters). Finally, the line `y[i] ~ categorical(theta[i]);` expresses the assumption that the response on trial *i* comes from (or is distributed as) a categorical distribution with the probabilities of the *K* categories set by `theta`, which we created in the block above.

Fitting the model

With the Stan model written and saved in a `.stan` file we can now switch to `R` to fit the model. The `rstan` package takes data in list form so we need to extract the relevant information from the data set that we read earlier on (saved in the object, `rdat`). Here we need to recall what names we use in the `data` block of the Stan model and match these to the list items in `R`:

```r
data_list = list(
  N = nrow(rdat), # number of observations (trials)
  y = rdat$rating, # response
  J = length(unique(rdat$id)), # number of participants
  id = rdat$id, # participant ids
  X = model.matrix(~ 1 + group, data = rdat), # predictors
  sig_trial = rdat$signal, # was this a signal trial? (1 = yes, 0 = no)
  K = 6 # number of rating categories
)
```

`data_list` now contains everything we need. The `stan` function does the work of fitting the model:

```r
library(rstan)


SDT_m1_fit <- stan(
  file = "models/SDT_m1.stan", # the stan model (from a separate file)
  data = data_list, # the list created above
  chains = 4, # run 4 separate chains to assess convergence
  warmup = 1000, # these are used to tune the sampler and ’burn in’
  iter = 2000, # number of iterations (#kept = chains*(iter - warmup))
  cores = 4 # chains can be run in parallel on separate cores (if possible)
)
```

Once the sampling is complete the `fit` object contains posterior samples of the model parameters. To assess whether the sampler has converged on a stable posterior distribution we can compare variability within chains to that between (using the $\hat{R}$ statistic discussed in Gelman et al., 2014, pp. 284–286). Discussion of other warning messages that may be produced by Stan and ways to resolve them is beyond the present scope (see https://mc-stan.org/misc/warnings.html).

Does the model do a good job? It is important to assess whether the fitted model provides an accurate representation of the observed data. One way of doing this is by plotting data simulated from the fitted model (posterior predictions) against the observed data. This requires that we add an extra block, `generated quantities`, to our Stan model, and this is covered in the supplementary material.

Assessing age differences in model parameters. We can extract samples for parameters of interest. In particular we are interested in $\beta_1^{(d)}$ which is the coefficient associated with age differences in sensitivity. The code below extracts the samples for this parameter and plots a histogram.

```r
age_d = extract(SDT_m1_fit, pars="B_d[2]")[[1]] # extract the age effect on d
```

```r
# plot a histogram
hist(age_d, breaks=20, xlab="", main=bquote(Beta[1]^'(d)'), probability = T)
```

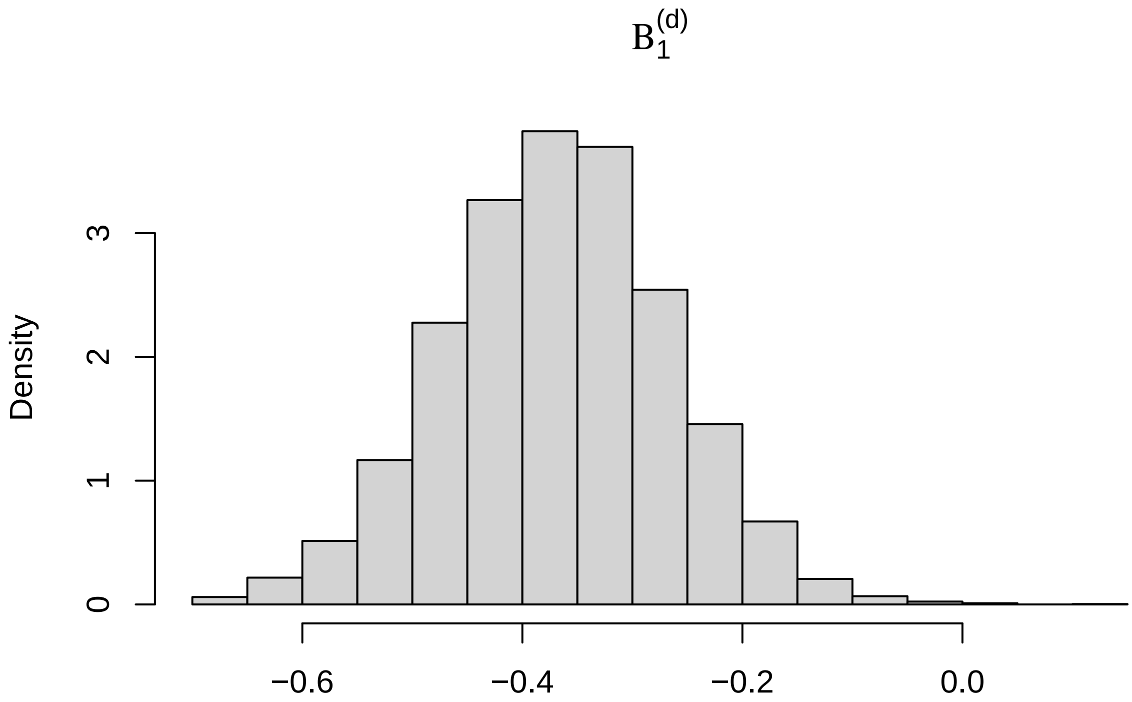

Figure 3. Histogram of posterior samples for the hiearchical parameter estimating age differences in d (on the transformed/log scale).

Do the groups differ in sensitivity? Assuming we have converged onto a stable distribution, parameter values will appear in the samples in proportion to their density under the posterior distribution. This means we can make statements like, "there is a 95% chance that the true difference between groups falls in this interval" (which is not the case for standard confidence intervals; Hoekstra, Morey, Rouder, & Wagenmakers, 2014).

In the `R` code and output below we extract the posterior mean and median as well as measures on uncertainty: 95% credible (CI) and highest density (HDI) intervals. The CI is based on quantiles of the posterior samples whereas the HDI is the shortest interval that contains X% of the posterior samples (see Kruschke, 2015 for details on interpretation).

```
# posterior mean
mean(age_d)
```

```
## [1] -0.367988
```

```
# median and 95% credible interval
quantile(age_d, probs = c(0.025, .5, 0.975))
```

```
##       2.5%        50%      97.5%
## -0.5716423 -0.3674097 -0.1672130
```

```
# 95% highest density interval
library(HDInterval)
hdi(age_d, credMass = .95)
```

```
##      lower      upper
## -0.5809965 -0.1784062
## attr(,"credMass")
## [1] 0.95
```

Thus, our best estimate of the group difference in log(d) is -0.37 and we cannot confidently rule out values of between -0.58 and -0.18 (95% HDI). It may be more intuitive to convert difference back to the natural scale of d. This is done in the code below:

```
# extract the group-level effects for d
B_d = extract(SDT_m1_fit, pars="B_d")[[1]]
# column 1 of B_d is the intercept and column 2 is the age difference
# the younger group is the intercept as they were coded zero

d_young = exp( B_d[,1] ) # transform back to natural scale
d_old = exp( B_d[,1] + B_d[,2] )

hdi(d_young)
```

```
##    lower    upper
## 1.866045 2.475373
## attr(,"credMass")
## [1] 0.95
```

```
hdi(d_old)
```

```
##    lower    upper
## 1.268319 1.713491
## attr(,"credMass")
## [1] 0.95
```

```
hdi(d_young - d_old) # HDI for the group difference
```

```
##     lower     upper
## 0.3017166 1.0416610
## attr(,"credMass")
## [1] 0.95
```

The 95% HDI for the difference in d between the younger and older groups is [0.30, 1.04].

Extending the model

The first model can be extended in multiple ways. For example, if we wanted to evaluate the weight of evidence in favor an age difference in d we could construct a second model in which this parameter is fixed across groups (this model is written in the `SDT_m2.stan` file) to compare to the first model. The supplemental material covers model comparison using the `bridgesampling` package (Gronau & Singmann, 2017) to calculate Bayes' factors. The supplement also covers the inclusion of item/stimulus effects, which are often an important source of variability in performance, and how to model differences in between-participant variability between different age groups (Shammi, Bosman, & Stuss, 1998).

The extension we want to cover here goes beyond asking whether there are age differences in certain parameters (we will focus again on d) to ask whether group differences are modulated by experimental manipulation (tests of group × condition interaction). In the example data set there is the additional factor of `cond` and each participant provides observations in both conditions (i.e., repeated measures). Therefore, we can model individual differences in the effect of condition. To extend the model (see `SDT_m3.stan`) we also must expand the `data` block of the model to change the design matrix, `X`, for the population-level effects and introduce a design matrix, `Z`, for individual-level effects:

```
matrix[N, 4] X;                 // fixed design matrix
matrix[N, 2] Z;                 // random design matrix
```

In `R` we also modify the data list so that `X` codes for main effects of group and condition plus their interaction and `Z` codes for the main effect of condition (including an individual-level effect of group here would not make sense as each individual can only belong to one group).

```
data_list$X = model.matrix(~ 1 + group + cond + group:cond, data = rdat)
data_list$Z = model.matrix(~ 1 + cond, data = rdat)
```

The `parameters` block must also be modified to reflect that there are now 4 group level effects and 2 individual effects, for which we are also modeling the correlation. Estimating the correlation allows us to assess whether participants with greater discriminability overall (captured by the intercept) exhibit a larger or smaller condition effect.

```
// d
vector[4] B_d; // 4 population effects (intercept, group, condition, interaction)
vector[2] b_d[J]; // 2 individual effects (intercept, condition)
corr_matrix[2] Sigma_d; // correlation of individual effects
vector<lower=0>[2] tau_d; // SD of individual effects
```

In the `transformed parameters` block the main modification is to the line determining d to reflect the combination of the group- and individual-level parameters. In addition the lines for the other parameters are modified so that only the first two columns of the design matrix (`X[i,1:2]`) are used as we are only modeling the main effect of age group for these parameters:

```
d[i] = exp( dot_product(X[i,], B_d) + dot_product(Z[i,], b_d[id[i]]) );
a[i] = exp( dot_product(X[i,1:2], B_a) + b_a[id[i]] );
b[i] = dot_product(X[i,1:2], B_b) + b_b[id[i]];
s[i] = exp( dot_product(X[i,1:2], B_s) + b_s[id[i]] );
```

Finally, in the model block we must modify the way in which individual-level parameters for d are determined. The first line specifies the prior distribution for the correlation matrix Sigma_d and the subsequent lines loop through the participants and sample their parameters from a multivariate normal distribution (the quad_form_diag function creates a covariance matrix).

```
Sigma_d ~ lkj_corr(1.0);


// sample individual coefficients
for (j in 1:J){
b_d[j] ~ multi_normal([0,0], quad_form_diag(Sigma_d, tau_d));
}
```

The line Sigma_d ~ lkj_corr(1.0); specifies an “LKJ” prior (Lewandowski, Kurowicka, & Joe, 2009) on the correlation matrix for the individual-level d effects (i.e., the participant intercept and effect of condition). The setting of 1 means that all correlations (-1 to 1) are equally likely before seeing the data. More information on specifying multivariate priors can be found at https://mc-stan.org/docs/2_24/stan-users-guide/multivariate-hierarchical-priors-section.html.

With these modifications we are ready to fit this model with rstan:

```
SDT_m3_fit <- stan(
  file = "models/SDT_m3.stan",
  data = data_list,
  chains = 4,
  warmup = 1000,
  iter = 2000,
  cores = 4
)
```

The supplementary material shows how to use this fitted model to calculate differences between groups and conditions.

Discussion

In this tutorial, we have laid out how to model age differences in the cognitive processes that are theorized to underlie task performance. We have done so with an example of a choice discrimination task, given that this is one of the most commonly encountered paradigms in the study of cognitive aging, and with models inspired by signal detection theory, as these represent some of the most popular cognitive models. Nevertheless, the approach we have outlined here can be applied broadly to other models and other tasks of cognition. For example, researchers interested in measuring age differences in working memory capacity can easily amend the model code to include standard formulas for calculating working memory capacity (e.g., Cowan, 2001; Pashler, 1988; and see Rhodes, Cowan, Hardman, & Logie, 2018; Greene, Naveh-Benjamin, & Cowan, 2020 for applications of hierarchical Bayesian working memory capacity models with age comparisons). We provide extensions of the modeling techniques reported here in the supplement, where we also discuss how to use MPT models with an example of Bröder, Kellen, Schütz, and Rohrmeier (2013)'s ratings model.

As cognitive aging researchers are primarily interested in understanding how cognitive processes change across the adult lifespan, cognitive models are better suited to measuring these theorized processes than are traditional models (e.g., ANOVA) applied to the observed data (e.g., accuracy). Of course, as with any statistical model, a cognitive model is merely an approximation of reality. Researchers must be aware of limitations of cognitive models before using them. For example, although cognitive models are useful for deriving estimates for parameters corresponding to theorized processes, these estimates cannot, strictly speaking, indicate whether a theorized process truly exists, as all cognitive processes are unobservable by nature. Also, many cognitive models have been designed and validated for specific tasks, and such models are not suitable for measuring some other phenomena. This is true of MPT models (Batchelder & Riefer, 1999), for example.

There are multiple classes of cognitive models, many suitable for different tasks or outcomes. In addition, we can distinguish between *computational models* and *measurement models*. Computational models aim to formulate process theories to adequately describe cognitive processes and representations underlying tasks like the encoding, storage, and retrieval from memory. Common examples are global memory models like REM (Shiffrin & Steyvers, 1997) and SAM (Raaijmakers & Shiffrin, 1981), which make specific predictions about the structure and mechanics of memory. Such models make specific predictions about the archicture of a cognitive process, such as how memory traces are created (e.g., as vectors of event features). Measurement models, on the other hand, aim to explain how individuals arrive at responses on a task by some combination of cognitive processes and response strategies, which entails that these models are more akin to general decision theories. The SDT models covered in this tutorial are one popular example of a measurement model because they explain discrimination on a two-choice task as a combination of memory strength and response bias, but besides conceptualizing the output of the memory retrieval process as continuous (as opposed to discrete, which is implied by MPT models), SDT models do not make specific processing predictions about the encoding or retrieval from memory.

Widespread use of cognitive modeling in cognitive aging research can be useful for addressing potential issues of non-replicability in the field. Because many studies in cognitive aging only rely on analyses applied to the raw data (as our survey of the last ten years of articles in *Psychology and Aging* indicates), these studies risk conflating the contributions of cognitive processes with response bias or other non-cognitive processes

(Rhodes et al., 2019; Rotello, Masson, & Verde, 2008). Accordingly, these studies risk overestimating the amount of age-related change that occurs to specific cognitive processes (e.g., memory strength). If age differences in cognitive processes are actually quite small, then the true effect size will be small as well, such that the probability of replication will be reduced, especially with under-powered studies. Relatedly, such studies may lead to a mistaken literature of age differences in cognitive processes that are based on analyses that conflate different processes or some processes with participant bias (in this case replication would only serve to compound the error; Rotello et al., 2015). In addition, cognitive models are better suited to ascertaining whether there is evidence for or against an age difference, as age comparisons in cognitive models are based on differences in parameters, which enables equivalence testing. As evidence for null effects may of particular interest to cognitive aging researchers, the ability to quantify such evidence, as with Bayesian implementations of cognitive models, should be a powerful asset for cognitive aging researchers.

Recommendations for Best Practices and Further Readings

This tutorial is likely to serve as one important learning tool for cognitive aging researchers as they begin to embark on their own cognitive modeling journeys. However, there are certainly other resources to consult, and learning cognitive modeling requires time, patience, and dedication. For a lengthier read about cognitive modeling in general (though not with specific applications in aging, per se), we recommend the great book by Farrell and Lewandowsky (2018), which also includes references to R code and Bayesian modeling. There are many great books for learning Bayesian statistics more generally, including Kruschke (2015) and McElreath (2016) (and see the article by Etz, Gronau, Dablander, Edelsbrunner, and Baribault (2018) for an annoted reading list). Finally, many great articles on the topic of cognitive modeling were published in the December 2019 issue of *Computational Brain & Behavior*, which featured a lengthy discussion among several top cognitive modelers on the topic of robust modeling in cognitive science (Lee et al., 2019) and other best practices in cognitive modeling, including whether cognitive models should be pre-registered (MacEachern & Van Zandt, 2019).

Cognitive modeling is an ever-evolving field. Therefore, it is important that cognitive modelers keep as up-to-date as possible with new advances and recommended practices. In this tutorial, we focused on hierarchical Bayesian estimation of cognitive models, as estimation under a hierarchical Bayesian framework is suitable for addressing both and individual- and group-level effects simultaneously and is a powerful technique for non-linear modeling, which is often not possible with frequentist methods. Bayesian analyses are more robust than frequentist equivalents, as they can regularize inference in low-power situations (Morey, Romeijn, & Rouder, 2016) and are suitable for quantifiying uncertainty about cognitive parameters, and models in general (Wagenmakers, Morey, & Lee, 2016). When adopting a Bayesian analytical approach for cognitive modeling, researchers should consider the following practices regarding choice of prior specification of model parameters; diagnostic checks of model adequacy, convergence, and fit; and posterior predictive checks of the model, using simulations from the model's posterior distribution (for a more detailed discussion, including a recommended work-flow, see Schad, Betancourt, and Vasishth (2021)). In addition, it is often necessary to compare multiple models, and such comparisons can be made in a Bayesian framework with a Bayes factor approach, though researchers should be mindful of the sensitivity of Bayes factors to priors. We discuss each of these points in more depth below.

Choice of Priors. An important ingredient in a Bayesian model is the prior distribution of the model's parameters. The prior conveys our beliefs about the values a parameter should take before we see the data; although this may seem "subjective," a good prior is chosen based on a careful consideration of the literature. There is no one perfect prior that fits every model. Instead, priors need to be chosen based on reasonable assumptions about what values are plausible for a parameter. For example, if a researcher is interested in comparing reaction times between young and older adults with a drift diffusion model, specifying a prior on the drift rate parameter that places most of its weight on extreme reaction times of, for instance, less than 50 ms would be a bad prior, as too much prior weight is given to implausible reaction times (for a discussion of why such fast reaction times are implausible, see, for example Rouder, Tuerlinckx, Speckamn, Lu, and Gomez (2008)).

We recommend that researchers select priors that are sensible for their experimental design and the research question at hand. Priors can be chosen to be weakly informative, such that they do not weigh as heavily as the data in determining the values of the posterior distribution. However, researchers should be careful when choosing diffuse priors (which cast a wide net overly an extremely large range of plausible values for a parameter), as these can result in extremely large and unreliable posterior estimates if the data is underpowered. For non-linear models, which are common for cognitive models, priors need to be informative enough to ensure the posterior sampling algorithm (e.g., MCMC) can get started and that posterior chains can converge. It is also important to consider what scale the priors are being modeled on; a Normal(0,10) prior may be weakly informative when working with parameters that are modeled on the raw scale of the data, but could be extremely diffuse if the model parameters are on a log-normal scale.

Researchers should evaluate how influential their priors were by conducting prior sensitivty analyses, in which they compare the posterior distribution of the model parameters, based on the same data set, with different prior specifications. These may include a weakly informative prior and a highly informative prior. Subtle changes in the posterior distribution are likely to occur, but if substantial changes occur, this may call into question how reliable the data were, or whether the chosen prior was appropriate.

Sometimes a prior may be grossly inadequate for a particular research question. Researchers may consider conducting a prior predictive check to test whether their chosen priors make reasonable predictions about the possible values a parameter may take or whether these priors result in overestimation extreme values (for a detailed discussion on prior predictive checks, including a worked example, see Schad et al. (2021)).

Diagnosing Model Convergence. Cognitive models are too complex for their posterior distributions to be solved analytically. Instead, estimation of the posterior distribution relies on sampling from it with an algorithm, usually an MCMC algorithm. A detailed discussion of MCMC is beyond the scope of this article. However, it is important to ensure that the algorithm produced reliable posterior estimates. When sampling from the model's posterior distribution, it is recommended that the analyst run multiple chains (usually at least 3 or 4). From each chain, the analyst specifies a set number of iterations, as we demonstrated in our worked example in this tutorial. The first several hundred (or, in some cases, thousands) of these iterations should be designated as burn-in samples, which are discarded from the actual posterior summaries of the model's parameters. This burn-in period is necessary to ensure that the chains converge to the same values of the posterior distribution. Convergence can be diagnosed in multiple ways, as with an autocorrelation plot, which shows how correlated successive iterations of the sampling algorithm are across various lags. For example, an autocorrelation plot can demonstrate whether two samples drawn from the posterior distribution at 50 iterations apart are very highly correlated; if they are, this likely indicates that the MCMC algorithm is getting stuck in some region of the posterior distribution. The analyst can correct for high autocorrelation by running a higher number of iterations and setting a thinning rate to ensure that only every, say, 10 iterations are retained. Convergence can also be diagnosed with the Gelman-Rubin statistic described in Gelman et al. (2014).

Assessing Model Fit: Posterior Predictive Checks. After fitting a model, it is the researcher's responsibility to ensure the model actually fits the data and to assess the predictive power of the model to fit unseen data. One way to do this is with posterior predictive checks, which we described in our worked example earlier. The basic idea of a posterior predictive check is that, based on the posterior distribution of the model, random samples of “new” data are simulated. Next, these new samples are compared with the observed data, and a good match (e.g., in terms of some summary statistic, like a mean) indicates that the model fits well because it can recover the summary statistics of the data in its simulations.

Conclusions

To conclude, aging researchers seeking to understand age differences in cognitive processes should consider incorporating cognitive modeling, either in lieu of standard statistical analyses applied to the raw data, or to supplement those analyses with models that parameterize the latent cognitive processes of theoretical or empirical interest.

References

Archambeau, K., Forstmann, B., Van Maanen, L., & Gevers, W. (2020). Proactive interference in aging: A model-based study. *Psychonomic Bulletin & Review*, *27*(1), 130–138.

Aust, F., & Barth, M. (2020). *papaja: Create APA manuscripts with R Markdown*. Retrieved from https://github.com/crsh/papaja

Batchelder, W. H., & Riefer, D. M. (1999). Theoretical and empirical review of multinomial process tree modeling. *Psychonomic Bulletin & Review*, *6*, 57–86.

Bayen, U. J., Murnane, K., & Erdfelder, E. (1996). Source discrimination, item detection, and multinomial models of source monitoring. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(1), 197.

Bays, P. M., Catalao, R. F., & Husain, M. (2009). The precision of visual working memory is set by allocation of a shared resource. *Journal of Vision*, *9*(10), 7–7.

Boywitt, C. D., Kuhlmann, B. G., & Meiser, T. (2012). The role of source memory in older adults' recollective experience. *Psychology and Aging*, *35*(6), 866–880.

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, *57*(3), 153–178.

Bröder, A., Kellen, D., Schütz, J., & Rohrmeier, C. (2013). Validating a two-high-threshold measurement model for confidence rating data in recognition. *Memory*, *21*(8), 916–944.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1).

Clark, H. H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior*, *12*(4), 335–359.

Cowan, N. (2001). The magical number 4 in short-term memory: A recosnideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87–114.

Estes, W. K. (1956). The problem of inference from curves based on group data. *Psychological Bulletin*, *53*(2), 134.

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 & Review*, *25*, 219–234.

Farrell, S., & Lewandowsky, S. (2018). *Computational modeling of cognition and behavior*. Cambridge, UK: Cambridge University Press.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014).

Bayesian data analysis (Vol. 3). Chapman & Hall/CRC Boca Raton, FL, USA.

Gelman, A., & Hill, J. (2007). *Data analysis using regression and hierarchical/multilevel models*. Cambridge, UK: Cambridge University Press.

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics*. New York: Wiley.

Greene, N. R., Naveh-Benjamin, M., & Cowan, N. (2020). Adult age differences in working memory capacity: Spared central storage but deficits in ability to maximize peripheral storage. *Psychology and Aging*, *35*(6), 866–880.

Gronau, Q. F., & Singmann, H. (2017). *Bridgesampling: Bridge sampling for marginal likelihoods and bayes factors*. Retrieved from https://CRAN.R-project.org/package=bridgesampling

Heathcote, A., Brown, S., & Mewhort, D. J. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, *7*(2), 185–207.

Hoekstra, R., Morey, R. D., Rouder, J. N., & Wagenmakers, E.-J. (2014). Robust misinterpretation of confidence intervals. *Psychonomic Bulletin & Review*, *21*(5), 1157–1164.

Kruschke, J. K. (2015). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press.

Lee, M. D. (2011). How cognitive modeling can benefit from hierarchical bayesian models. *Journal of Mathematical Psychology*, *55*(1), 1–7.

Lee, M. D., Criss, A. H., Devezer, B., Donkin, C., Etz, A., Leite, F. P., … Vandekerckhove, J. (2019). Robust modeling in cognitive science. *Computational Brain & Behavior*, *2*(3), 141–153.

Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press.

Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, *100*(9), 1989–2001.

Loftus, G. R. (1978). On interpretation of interactions. *Memory & Cognition*, *6*(3), 312–319.

Loitile, R. E., & Courtney, S. M. (2015). A signal detection theory analysis of behavioral pattern separation paradigms. *Learning & Memory*, *22*(8), 364–369.

MacEachern, S. N., & Van Zandt, T. (2019). Robust modeling in cognitive science. *Computational Brain & Behavior*, *2*(3), 179–182.

Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A user's guide* (2nd ed.). Mahwah, New Jersey: Lawrence Erlbaum Associates.

McElreath, R. (2016). *Statistical rethinking: A bayesian course with examples in R and Stan*. Taylor & Francis.

Meiser, T., & Bröder, A. (2002). Memory for multidimensional source information. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *28*(1), 116–137.

Morey, R. D., Pratte, M. S., & Rouder, J. N. (2008). Problematic effects of aggregation in z roc analysis and a hierarchical modeling solution. *Journal of Mathematical Psychology*, *52*(6), 376–388.

Morey, R. D., Romeijn, J.-W., & Rouder, J. N. (2016). The philosophy of Bayes factors and the quantification of statistical evidence. *Journal of Mathematical Psychology*, *72*, 6–18.

Oberauer, K., & Lin, H.-Y. (2017). An interference model of visual working memory. *Psychological Review*, *124*(1), 21.

Pashler, H. (1988). Familiarity and visual change detection. *Perception & Psychophysics*, *44*, 369–378.

Paulewicz, B., & Blaut, A. (2020). The bhsdtr package: A general-purpose method of bayesian inference for signal detection theory models. *Behavior Research Methods*, 1–20.

Pratte, M. S., & Rouder, J. N. (2012). Assessing the dissociability of recollection and familiarity in recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *38*(6), 1591.

Raaijmakers, J. G., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review*, *88*(2), 93–134.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, *20*(4), 873–922.

Ratcliff, R., Sheu, C.-F., & Gronlund, S. D. (1992). Testing global memory models using ROC curves. *Psychological Review*, *99*(3), 518–535.

R Core Team. (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Reagh, Z. M., & Yassa, M. A. (2014). Repetition strengthens target recognition but impairs similar lure discrimination: Evidence for trace competition. *Learning & Memory*, *21*(7), 342–346.

Rhodes, S., Abbene, E. E., Meierhofer, A. M., & Naveh-Benjamin, M. (2020). Age differences in the precision of memory at short and long delays. *Psychology and Aging*, *35*(8), 1073.

Rhodes, S., Cowan, N., Hardman, K. O., & Logie, R. H. (2018). Informed guessing in change detection. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *44*(7), 1023–1035.

Rhodes, S., Cowan, N., Parra, M. A., & Logie, R. H. (2019). Interaction effects on common measures of sensitivity: Choice of measure, type i error, and power. *Behavior Research Methods*, *51*(5), 2209–2227.

Rotello, C. M., Heit, E., & Dubé, C. (2015). When more data steer us wrong: Replications with the wrong dependent measure perpetuate erroneous conclusions. *Psychonomic Bulletin & Review*, *22*, 944–954.

Rotello, C. M., Masson, M. E., & Verde, M. F. (2008). Type I error rates and power analyses for single-point sensitivity measures. *Perception & Psychophysics*, *70*(2), 389–401.

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*(4), 573–604.

Rouder, J. N., Morey, R. D., Morey, C. C., & Cowan, N. (2011). How to measure working memory capacity in the change detection paradigm. *Psychonomic Bulletin and Review*, *18*, 324–330.

Rouder, J. N., Tuerlinckx, F., Speckamn, P., Lu, J., & Gomez, P. (2008). A hierarchical approach for fitting curves to response time measurements. *Psychonomic Bulletin & Review*, *15*(6), 1201–1208.

Salthouse, T. A. (2000). Methodological assumptions in cognitive aging research. In F. I. M. Craik & T. A. Salthouse (Eds.), *The handbook of cognitive aging* (2nd ed., pp. 467–498). Mahwah,NJ: Erlbaum.

Schad, D. J., Betancourt, M., & Vasishth, S. (2021). Toward a principled bayesian workflow in cognitive science. *Psychological Methods*, (1), 103–126.

Selker, R., Bergh, D. van den, Criss, A. H., & Wagenmakers, E.-J. (2019). Parsimonious estimation of signal detection models from confidence ratings. *Behavior Research*

Methods, *51*(5), 1953–1967.

Shammi, P., Bosman, E., & Stuss, D. T. (1998). Aging and variability in performance. *Aging, Neuropsychology, and Cognition*, *5*(1), 1–13.

Shiffrin, R. M., & Steyvers, M. (1997). Model for recognition memory: REM—retrieving effective from memory. *Psychonomic Bulletin & Review*, *4*, 145–166.

Snodgrass, J. G., & Corwin, J. (1988). Pragmatics of measuring recognition memory: Applications to dementia and amnesia. *Journal of Experimental Psychology: General*, *117*(1), 34–50.

Souza, A. S. (2016). No age deficits in the ability to use attention to improve visual working memory. *Psychology and Aging*, *31*(5), 456.

Stan Development Team. (2018). RStan: The R interface to Stan. Retrieved from http://mc-stan.org/

Stark, S. M., Yassa, M. A., Lacy, Joyce W, & Stark, C. E. L. (2013). A task to assess behavioral pattern separation (BPS) in humans: Data from healthy aging and mild cognitive impairment. *Neuropsychologia*, *51*(12), 2442–2449.

Swets, J. A. (1986a). Form of empirical ROCs in discrimination and diagnostic tasks: Implications for theory and measurement of performance. *Psychological Bulletin*, *99*(2), 181–198.

Swets, J. A. (1986b). Indices of discrimination or diagnostic accuracy: Their ROCs and implied models. *Psychological Bulletin*, *99*(1), 100–117.

Tanner Jr, W. P., & Swets, J. A. (1954). A decision-making theory of visual detection. *Psychological Review*, *61*(6), 401–409.

Van den Berg, R., Shin, H., Chou, W.-C., George, R., & Ma, W. J. (2012). Variability in encoding precision accounts for visual short-term memory limitations. *Proceedings of the National Academy of Sciences*, *109*(22), 8780–8785.

Wagenmakers, E.-J., Krypotos, A.-M., Criss, A. H., & Iverson, G. (2012). On the interpretation of removable interactions: A survey of the field 33 years after Loftus. *Memory & Cognition*, *40*(2), 145–160.

Wagenmakers, E.-J., Morey, R. D., & Lee, M. D. (2016). Bayesian benefits for the pragmatic researcher. *Current Directions in Psychological Science*, *25*(3), 169–176.

Wagenmakers, E.-J., Van Der Maas, H. L., & Grasman, R. P. (2007). An ez-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, *14*(1), 3–22.

Yonelinas, A. P. (1994). Receiver-operating characteristics in recognition memory: Evidence for a dual-process model. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*(6), 1341.

Yonelinas, A. P., & Parks, C. M. (2007). Receiver operating characteristics (ROCs) in recognition memory: a review. *Psychological Bulletin*, *133*(5), 800–832.

Zhang, W., & Luck, S. J. (2008). Discrete fixed-resolution representations in visual working memory. *Nature*, *453*(7192), 233–235.

Table 1

Selected models that were also used in papers identified by our review of Psychology and Aging between 2011 and 2020. The references and related/competing models are not exhaustive.

Model	Brief description	Selected references	Selected related/competing models
Multinomial processing tree models of source monitoring	To judge the source of previously studied information or whether a test probe is a new lure participants are assumed to enter distinct cognitive states in sequence. For example, they may detect that a particular probe is old but then fail to detect the source of the probe in which case they would have to guess between sources. This is in contrast to SDT in which participants are assumed to monitor a continuous decision variable (e.g., familiarity) and not enter a discrete detect/not-detect state.	Bayen, Murnane and Erdfelder (1996); Meiser and Bröder (2002); Boywitt, Kuhlmann, and Meiser (2012)	Threshold models of signal detection (Snodgrass & Corwin, 1988). Dual process model of recollection and familiarity (Yonelinas, 1994). Models of the capacity of working memory (Rouder, Morey, Morey, & Cowan, 2011).
Diffusion model of choice reaction time and accuracy	In two-choice decision tasks participants are assumed to accumulate information (corrupted by noise) and respond when there is strong enough evidence in favor of a particular response. There are parameters that control the rate of accumulation, bias towards particular responses, and time spent not making the decision (encoding, output). This model simultaneously fits reaction time and accuracy data.	Ratcliff (1978); Ratcliff and McKoon (2008)	Linear ballistic accumulator model (Brown & Heathcote, 2008). EZ-diffusion (Wagenmakers, Van Der Maas, & Grasman, 2007)
Mixture model of recall error	Participants are cued to recall a particular feature (e.g., color, orientation, location) as precisely as possible. Recall error, or the distance between the studied and recalled feature, is assumed to reflect a mixture of guesses and true recall from memory with some precision. Some models also assume that participants occasionally erroneously recall a feature associated with another cue.	Zhang and Luck (2008); Bays, Catalao, and Husain (2009); Rhodes, Abbene, Meierhofer, and Naveh-Benjamin (2020); Souza (2016)	Variable precision model (Van den Berg, Shin, Chou, George, & Ma, 2012). Interference model (Oberauer & Lin, 2017)
Prototype models of category learning			

Table 2

Selected parts of the Stan syntax used in the present example with a brief explanation

Syntax	Explanation
`int<lower=1> N;`	`N` is an integer (int) and cannot be lower than 1
`int y[N];`	`y` contains `N` integers and `y` is an array (arrays are different from vectors and matrices, which are limited to one and two dimensions, respectively)
`int<lower=1,upper=J> id[N];`	`id` contains `N` integers that must be between 1 and `J` (where `J` is also an integer)
`1:(K-1)`	produces a sequence of integers between 1 and `K-1` (e.g., if `K` is 6 the result would be `[1,2,3,4,5]`)
`matrix[N, 2] X;`	`X` is a matrix with `N` rows and 2 columns
`vector[N] sig_trial;`	`sig_trial` is a vector of `N` numbers
`real b_d[J];`	`b_d` is an array of `J` numbers that can take any value
`real<lower=0> tau_d;`	`tau_d` is a single number that can take any value of 0 or greater
`for (i in 1:N){ ...}`	produce a sequence of integers between 1 and `N` and for each integer (`i`) in this sequence run the code between the braces (note `i` is often used by convention but doesn't have to be)
`x = 1;`	`x` is assigned the value of 1
`x ~ normal(0, 1);`	`x` is assumed to be normally distributed with a mean of 0 and a standard deviation of 1
`b[i] = dot_product(X[i,], B_b)`	the element of `b` in position `i` is equal to the dot product of row `i` of matrix `X` and `B_d`
`b_b[id[i]];`	find the `id` of the participant that observation `i` comes from (`id[i]`) and use this to select the value of `b_b` corresponding to that `id`
`y[i] ~ categorical(theta[i]);`	element `i` of `y` is sampled from a categorical distribution with probabilities determined by element `i` of `theta`
`exp(log(k) - log(K))`	if `k` and `K` are integers the result of `kK` will also be an integer, which is not what we want when setting the unbiased criteria of the SDT model. Subtracting the logarithms and taking the exponent produces the non-integer result we desire