Promoting Data Science

Volume 3 2023 Number 1

Journal of Behavioral Data Science V3N1 (2023)

https://isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

ISBN: 2575-8306 (Print) 2574-1284 (Online)
https://jbds.isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

No Publication Charge and Open Access

jbds@isdsa.org

List of Articles

Journal of Behavioral Data Science, 2023, *3* (1), 1–33.
DOI: https://doi.org/10.35566/jbds/v3n1/marvin

Using Bayesian Piecewise Growth Curve Models to Handle Complex Nonlinear Trajectories

Luca Marvin, Haiyan Liu, and Sarah Depaoli

University of California, Merced, USA
lmarvin@ucmerced.edu

Abstract. Bayesian growth curve modeling is a popular method for studying longitudinal data. In this study, we discuss a flexible extension, the Bayesian piecewise growth curve model (BPGCM), which allows the researcher to break up a trajectory into phases joined at change points called *knots*. By fitting BPGCMs, the researcher can specify three or more phases of growth without concern for model identification. Our goal is to provide substantive researchers with a guide for implementing this important class of models. We present a simple application of Bayesian linear BPGCMs to childrens' math achievement. Our tutorial includes M*plus* code, strategies for specifying knots, and how to interpret model selection and fit indices. Extensions of the model are discussed.

Keywords: Piecewise Growth Curve Models · Bayesian SEM · Model Selection

1 Introduction

Developmental researchers often study within-person change over time to better understand a variety of dynamic processes. For example, Marksxand Coll (2007) contrasted growth in reading and math skills in children across four major ethnic groups from kindergarten through third grade in order to highlight the needs of American Indian and Alaska Native youth. Seiderxet al. (2019) documented the development of Black and Latino high school students' beliefs about poverty and racism to examine the role of schooling and how these beliefs relate to each other. Finally, Shono, Edwards, Ames,xand Stacy (2018) captured change in cannabis use across teen years as a component of validity testing a new cannabis-related word association test. These examples highlight a wide range of topics within developmental research.

For many developmental research questions, choosing an appropriate model to summarize the trajectory of development over time is crucial. Longitudinal methods typically describe within-person change and explain between-person differences in that change. There are many longitudinal models available, and a truly helpful model will guide the researcher to evaluate their research questions and meaningfully communicate their findings. Of the many different model forms that researchers can choose from, the growth curve model (GCM) is perhaps one of the more beneficial for tracking change over multiple time-points. The GCM uses repeated observations to estimate the latent population trajectory. Through GCMs, researchers can summarize change over time or test hypotheses about specific aspects of growth (e.g., the rate of change). In addition to summarizing within-person change, GCMs also allow researchers to examine between-person variability in development.

The GCM has many forms, and the simplest captures linear change over time (called a "linear GCM"). Researchers using a linear GCM can describe change with growth parameters that are straightforward to interpret: a mean intercept and a mean slope. For example, Marksxand Coll (2007) examined differences in reading development by interpreting the initial level of reading (i.e., the intercept) and the average rate of change (i.e., the slope) across ethnic groups. The linear GCM is useful in many research scenarios, but it also has some limitations that applied researchers should consider while selecting a model. The main limitation is that it assumes the true growth trajectory is a straight line, and can not capture nonlinear changes that may be of substantive importance.

In some cases examining more dynamic processes, this linear assumption is too restrictive and will not capture the substantive changes of most interest. Development may follow a curve or other irregular deviations from linearity. For example, Zimmer-Gembeckxet al. (2021) found the development of social anxiety in adolescents was best represented by a quadratic GCM. Vargas Lascano, Galambos, Krahn,xand Lachman (2015) found that a cubic model best fit the shifts in perceived control in adults aged 18 to 43. In aging adults across the last 16 years of life, Schillin, Deeg,xand Huisman (2018) found that the decrease in positive affect was best captured by an exponential GCM. The developmental trajectories in these studies were not linear, and so the researchers used GCMs that assumed a nonlinear growth trajectory.

An alternative to imposing any assumptions about the shape of the overall trajectory (e.g., a quadratic growth model) is to instead capture the trajectory with several linear segments using a linear piecewise growth curve model (PGCM; Meredithx& Tisak, 1990). The word "piecewise" indicates that the linear slope may be different across different "pieces" of the study period, which gives the researcher greater flexibility while maintaining simple parameters. For example, Finkel, Reynolds, McArdle,xand Gatz (2003) used a linear PGCM to capture cognitive decline in adults over 60 years of age, estimating different rates of change for observations before and after age 65. This approach allowed them to show that aging adults under 65 improved each year on certain cognitive measures, but those scores declined after age 65. More recently, Gaudreau, Louvet,xand Kljajic (2018) used a piecewise approach to capture the development of adolescents' performance in gymnastics classes, which decreased for the first three classes before showing consistent improvement in the last three classes of the study period. Taking a piecewise approach allowed these researchers to capture unique shifts in the direction of development over time.

These researchers used the simplest piecewise model: a linear-linear PGCM. This type of PGCM is useful for capturing a nonlinear trajectory with a single change in direction such as the switch from declining to improving performance in gymnastics (as shown in Gaudreauxet al., 2018). A linear-linear PGCM uses two phases of growth, but PGCMs with additional phases are possible with enough measurement occasions. For growth trajectories with more complex nonlinearity (i.e., growth with more than one change in direction), researchers may wish to use additional phases. In the frequentist framework, the number of phases is somewhat restricted in order to maintain model identification. One way to work around this restriction is to estimate PGCMs in the Bayesian estimation framework, an alternative approach that can be used to estimate some non-identified models. For PGCMs, this allows additional phases of growth.

In addition to allowing more phases of growth in PGCMs, Bayesian estimation has been shown to handle complex models with fewer estimation issues (e.g., convergence, biased estimates). Instead of relying solely on observed data and a likelihood function, Bayesian methods also incorporate prior information into estimation using a *prior distribution*. Wangxand McArdle (2008) found that Bayesian estimation fairly accurately captures parameters in nonlinear piecewise growth models, and Depaoli (2013) found that Bayesian growth mixture models estimated using informative priors yielded minimal bias in parameter estimates. Using Bayesian estimation methods with thoughtfully selected prior distributions can help to accurately recover model parameters.

Bayesian PGCMs extend conventionally-taught linear growth models by altering both the functional form of growth and the estimation framework. This is an active area of methodological development, with recent extensions that enable the direct estimation of knot placement (Kohli, Hughes, Wang, Zopluoglu,x& Davison, 2015; Lock, Kohli,x& Bose, 2018), incorporation of covariates (Lamm, 2022), and capturing the interdependent nature of bivariate piecewise trajectories (Peralta, Kohli, Lock,x& Davison, 2022). Our intended scope for the current paper is to provide an introductory, hands-on walkthrough to the novice data scientist or graduate student. That is, our tutorial is written to bridge the knowledge gap between linear growth curve models in the frequentist framework and more complex piecewise models estimated in the Bayesian framework. Given this audience, the specific goals of the current paper are:

- Present readers to Bayesian PGCMs as a flexible way to capture complex nonlinearity.
- Thoroughly illustrate Bayesian PGCMs with an empirical dataset, including how to select priors.
- Provide readers with additional resources to expand on this tutorial.

To achieve these goals, the remaining sections of the paper are structured as follows. First, we describe linear GCMs and how linear PGCMs are a simple extension. We also highlight how to extend PGCMs beyond two phases of growth. Second, we introduce Bayesian estimation. Our explanation describes some benefits of Bayesian estimation, key terminology, how to specify priors, and how the

Bayesian framework allows additional phases of growth. Third, we present an illustration of Bayesian PGCMs applied to nonlinear growth in childrens' math achievement. This demonstration provides the syntax to implement the model in M*plus*, illustrates how to use comparative model indices to select the best model, and shows how to interpret model results. Finally, we discuss the limitations of linear PGCMs and possible extensions.

2 Piecewise Growth Curve Models

The main goal of a growth model is to summarize many repeated within-person observations with a few growth parameters. The general form of a growth model is

$$y_j = g(t_j) + e_j, \tag{1}$$

which says that the jth measurement of the variable y is the sum of some function of time at the jth measurement $g(t_j)$ and timing-specific measurement error e_j. The j subscript indicates that the outcome, time, and error can vary across all $j = 1, 2, ..., J$ measurement occasions. In the following sections, we describe different specifications of $g(t_j)$. Next we describe a linear GCM, how GCMs can be adapted for nonlinearity, a two-phase linear PGCM, and linear PGCMs with three or more phases. Finally, we connect these models to M*plus* syntax.

2.1 Linear GCM

A linear GCM assumes the growth function $g(t_j)$ is a linear function of time t:

$$g(t_j) = \beta_0 + \beta_1 t_j, \tag{2}$$

where β_0 represents the intercept and β_1 represents the expected rate of change for every 1-unit increase in time t_j[1]. We refer to these coefficients as growth parameters. Researchers are typically interested in estimating linear growth parameters using a sample of $i = 1, 2, ..., N$ persons with repeated measurements at J different time points. To clarify that we are interested in estimating person-specific outcomes as a function of person-specific time, we add an i subscript to $g(t_j)$ in Equation (equation2). The linear growth function can be given by

$$g(t_{ij}) = \beta_0 + \beta_1 t_{ij}, \tag{3}$$

[1] The coding and interpretation of t_j is determined by the researcher. For example, t_j may refer to the number of weeks after the study began, or the number of months after an intervention. If the measurements were not spaced consistently, this can be reflected in the observed values of t_j. For example, a study with measurements in January, February, April, and July could code time as the number of months since the first measurement occasion so that $t_1 = 0, t_2 = 1, t_3 = 3$, and $t_4 = 6$. In this case, the intercept is placed at the first measurement occasion, but the researcher may choose a different placement. For example, if an intervention occurred in April, the researcher may choose to place the intercept there by recoding t_j as $t_1 = -3, t_2 = -2, t_3 = 0$, and $t_4 = 3$. Thoughtfully specifying time ensures that the intercept and slope can be interpreted in a meaningful way.

where the growth function of person i's time at measurement occasion j is a linear function with intercept β_0 and slope β_1. Plugging in this growth function and adding an i subscript to Equation (equation1) gives

$$y_{ij} = \beta_0 + \beta_1 t_{ij} + e_{ij}, \tag{4}$$

where y_{ij} refers to the outcome variable for person i at time j, t_{ij} is person i's time measured at time point j, and e_{ij} is unexplained error for person i at time point j. We assume that e_{ij} is normally distributed around zero, or $e_{ij} \sim N(0, \sigma^2_{ej})$. The error variance parameter σ^2_{ej} represents variability in the observed data at time j that is unexplained by the model. The two coefficients in this model, β_0 and β_1, refer to growth parameters that are held constant across persons. However, there is often some between-person fluctuations in the growth parameters. Imposing the same intercept and slope on each participant in the sample can lead to higher measurement error e_{ij}. To prevent this, we introduce a person-specific growth function, $d_i(t_{ij})$. We define $d_i(t_{ij})$ as,

$$d_i(t_{ij}) = \delta_{0i} + \delta_{1i} t_{ij}, \tag{5}$$

where δ_{0i} and δ_{1i} refer to a person-specific intercept and slope, respectively. We assume the values of δ_{0i} are distributed normally with a mean of β_0 and that δ_{1i} is normally distributed around β_1. These assumptions can be summarized in the following way:

$$\begin{bmatrix}\delta_0\\ \delta_1\end{bmatrix} \sim MVN\left(\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}, \mathbf{\Sigma}_\delta\right), \tag{6}$$

where $\mathbf{\Sigma}_\delta$ is a 2×2 covariance matrix. The diagonal elements of this matrix describe the variance of the intercept and variance of the slope. The off-diagonal element describes the covariance of the intercept and slope. These variances can have interesting substantive meaning. For example, if a researcher studied the number of words children learn from age two to five and found the variance of the intercept is smaller than the variance of the slope, this suggests that the number of words children knew at age two varies less than how many new words children learned per year. By replacing $g(t_j)$ with $d_i(t_{ij})$ in Equation (equation1), we can write the full linear GCM,

$$y_{ij} = \delta_{0i} + \delta_{1i} t_{ij} + e_{ij}, \tag{7}$$

which describes the outcome variable y_{ij} as a function of time t_{ij} and person i's growth parameters δ_{0i} and δ_{1i}.

2.2 Capturing Nonlinearity

Linear GCMs assume change over time can be captured with a straight line, but in some cases this linear assumption is too restrictive. Change in a variable over time may follow a curve or have other deviations from linear change. When change is not linear, the researcher's analysis plan must transition to capture nonlinearity. There are many ways to model nonlinearity, but these extensions may have limited applicability. For example, a researcher may add a third term such as "$+\delta_{2i}t_{ij}^2$" to Equation (equation7) to estimate a quadratic coefficient δ_{2i} for trajectories shaped like a parabola. Researchers can also alter linear GCM specifications in more complex ways to capture cyclical growth with a sine function (e.g. Bollenx& Curran, 2006) or S-shaped growth with a Gompertz curve (e.g. Grimmx& Ram, 2009). The parameters estimated by these models are shape-specific and some may be challenging to substantively interpret. When the goal of the model is simply to capture the trajectory, this is not a problem. However, when the researcher wants a simpler interpretation of growth parameters, an alternative method is to break up the trajectory into linear phases as shown in Figure figure1. These phases comprise a "piecewise" approach to modeling nonlinear growth patterns. Using this piecewise approach allows a GCM to capture nonlinear growth while maintaining the simple interpretation of linear slope parameters.

The simplest piecewise model uses two phases to capture growth with a single change in direction. The time when one growth phase switches to another is called a *knot*, denoted k. The knot is placed at a measurement occasion chosen by the researcher. We adapt the growth function in Equation (equation5) to include a change in slope at k:

$$d_i(t_{ij}) = \delta_{0i} + \delta_{1i}t_{ij} + \delta_{2i}(t_{ij} - k)_+. \tag{8}$$

Here, δ_{2i} represents the person-specific change in slope that occurs at values of t_{ij} after the knot. Similar to the other coefficients, δ_{2i} has a mean of β_2 and information about its variance and covariances are contained in a 3×3 covariance matrix $\mathbf{\Sigma}_\delta$. To implement a change in slope for some values of t_{ij} but not others, we introduce a new term, $(t_{ij} - k)_+$, which represents "the positive part of $t_{ij} - k$". This is defined as,

$$(t_{ij} - k)_+ = \begin{cases} 0 & \text{if } t_{ij} \leq k \\ t_{ij} - k & \text{if } t_{ij} > k, \end{cases} \tag{9}$$

which means the term $(t_{ij} - k)_+$ only appears when $t_{ij} - k$ positive. This means in Equation (equation8), person i's linear slope when $t \leq k$ is δ_{1i}, but the slope for $t > k$ is $\delta_{1i} + \delta_{2i}$. Adding this to Equation (equation7) gives the linear PGCM with one knot:

$$y_{ij} = \delta_{0i} + \delta_{1i}t_{ij} + \delta_{2i}(t_{ij} - k)_+ + e_{ij}. \tag{10}$$

Here, the coefficients δ_{0i} and δ_{1i} describe person-specific growth parameters in the first phase of growth. The person-specific change in slope at k is described by δ_{2i}. The last term, e_{ij}, describes leftover error that is not captured by $d_i(t_{ij})$. To illustrate this, consider Figure figure1, which shows a linear GCM in part (a) and a linear PGCM in part (b). In part (b), there are four measurement occasions $t_1 = 0$, $t_2 = 1$, $t_3 = 2$, and $t_4 = 3$, and a knot, $k = 2$. The rate of growth increases at the knot, which appears visually as a steeper slope from $k = 2$ onward.

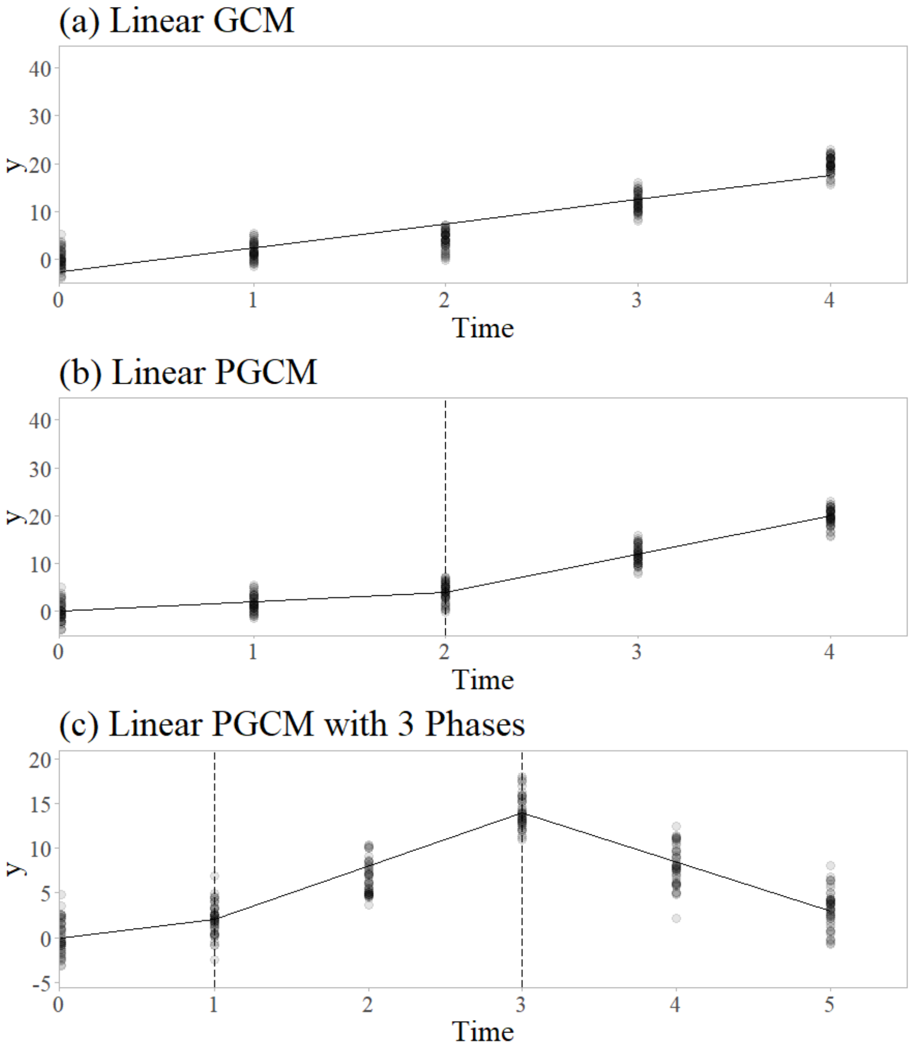

Figure 1. Examples of nonlinear development in a generic outcome *y*. The points represent simulated data and solid lines represent estimated growth trajectories for different models. Panel (a) shows a linear growth curve model (GCM) fitted to nonlinear data; panel (b) shows a linear piecewise growth curve model (PGCM) that divides the trajectory into two phases joined at a single knot indicated by the vertical dashed line; panel (c) shows a longer simulated trajectory with more complex nonlinearity that requires two knots (that is, three phases) to capture.

2.3 Extending PGCMs to Three or More Phases

In the frequentist framework, extending piecewise models beyond two phases of growth requires several measurement occasions. For example, Bollenxand Curran (2006) showed at least five measurement occasions are required to estimate a two-phase PGCM, and Flora (2008) noted that a three-phase PGCM needs at least seven measurements. These restrictions ensure the model is identified, meaning it has enough observed variables to estimate the parameters. A non-identified model cannot be estimated using frequentist methods. In this section we describe PGCMs with three or more phases, which traditionally require many measurement occasions. Later we describe the Bayesian estimation framework, an alternative approach that can estimate non-identified models.

To create more phases, the researcher must specify more knots. To refer to M specific knots, we use $k_1, k_2, ..., k_M$. First, we generalize the person-specific growth function $d_i(t_{ij})$ to address more phases of growth:

$$d_i(t_{ij}) = \delta_{0i} + \delta_{1i}t_{ij} + \sum_{m=1}^{M} \delta_{(1+m)i}(t_{ij} - k_m)_+. \tag{11}$$

The change from δ_{2i} in Equation (equation8) to $\sum_{m=1}^{M} \delta_{(1+m)i}$ here generalizes the growth function to handle more than two phases. Each coefficient next to the summation sign $\delta_{2i}, \delta_{3i}, ..., \delta_{(1+M)i}$ refers to a change in slope that occurs after the first phase. For example, for a model with $M = 5$ knots, the slope in the sixth and final phase of growth would be $\delta_{1i} + \delta_{2i} + ... + \delta_{6i}$, or $\delta_{1i} + \sum_{m=1}^{5} \delta_{(1+m)i}$. Putting the growth function from Equation (equation11) into Equation (equation1), we get the full linear PGCM:

$$y_{ij} = \delta_{0i} + \delta_{1i}t_{ij} + \sum_{m=1}^{M} \delta_{(1+m)i}(t_{ij} - k_m)_+ + e_{ij}. \tag{12}$$

This model is a generalization of the model shown in Equation (equation10) that can address two or more phases. The summation describes how the linear slope of each phase of growth is the sum of multiple coefficients.

To illustrate this concept, see part (c) in Figure figure1. This plot shows change over six measurements with two knots placed at $k_1 = 1$ and $k_2 = 3$. Visually, growth appears slow in the first phase, accelerates in the second phase, then switches to a decline in the third phase. We could specify these knots in Equation (equation12) in the following way:

$$y_{ij} = \delta_{0i} + \delta_{1i}t_{ij} + \delta_{2i}(t_{ij} - 1)_+ + \delta_{3i}(t_{ij} - 3)_+ + e_{ij}. \tag{13}$$

In this model, the general term $\sum_{m=1}^{M} \delta_{(1+m)i}(t_{ij} - k_m)_+$ has been spelled out as $\delta_{2i}(t_{ij} - 1)_+ + \delta_{3i}(t_{ij} - 3)_+$. As before, δ_{0i} and δ_{1i} describe person i's growth trajectory in the first phase of growth, which covers $t_1 = 0$ and $t_2 = 1$. The second phase of growth extends from the second time point to the fourth, or $1 < t \leq 3$. The rate of change in this phase is $\delta_{1i} + \delta_{2i}$. The third phase starts at the second knot $k_2 = 3$ and includes the next two time points. This phase of growth has the slope $\delta_{1i} + \delta_{2i} + \delta_{3i}$. This is equivalent to $\delta_{1i} + \sum_{m=1}^{2} \delta_{(1+m)i}$.

Nonlinear trajectories may show complex nonlinearity that does not have clear phases of growth. In these cases it is not clear how many phases are needed to capture the trend, or where knots should be placed. There may be multiple knot specifications that could capture the trajectory, or developmental theories may disagree on when one phase of growth ends and another begins. In these situations, a model selection approach can be useful.

Model selection is a method where multiple candidate models are estimated and compared before selecting the "best" one. The criteria for this selection is usually one or more model comparison indices, which are often provided by statistical software. These indices may include *model fit* indices or *model comparison* indices. Model fit refers to how well an estimated model minimizes error variance or "fits" the data. Model fit indices are used to evaluate the estimated model on some index-specific scale. For example, values below 0.05 suggest excellent fit according to the root mean square error of approximation (RMSEA; Brownex& Cudeck, 1992; Steigerx& Lind, 1980). Other model fit indices include the Comparative Fit Index (CFI; Bentler, 1990) and Tucker-Lewis Index (TLI; Tuckerx& Lewis, 1973). In contrast, model comparison is the task of comparing two or more models and selecting the model with the best balance of fit and parsimony.

Model comparison indices may be applied to PGCMs to select the best knot specification out of several candidate models. Two commonly-used indices are the Akaike information criterion (AIC; Akaike, 1992) and the Bayesian information criterion (BIC; Schwarz, 1978). These comparison indices describe the fit of a model (measured using the loglikelihood) penalized by model complexity (the number of free parameters in the model). When evaluating candidate models, the model with the smallest AIC (or BIC) is considered the winning model. For further information on these and other model comparison indices, we refer the reader to Nylund, Asparouhov,xand Muthén (2007).

2.4 Notation and M*plus* Syntax

Translating linear PGCMs to syntax is relatively straightforward. We start by showing how to implement the linear model in Equation (equation7) and part (a) of Figure figure1 in M*plus*. In this example, the five variables labelled `y1`, `y2`, `y3`, `y4`, and `y5` refer to observations of our variable of interest at five different measurement occasions:

```
MODEL:
delta_0 delta_1 | y1@0 y2@1 y3@2 y4@3 y5@4;
```

The `MODEL` command indicates to M*plus* that the following lines of code define our model. In the next line, `delta_0` and `delta_1` refer to the growth parameters we want to estimate: δ_{0i} and δ_{1i} from Equation (equation7). The `|` symbol means the intercept and slope on the left should be estimated using the information on the right. On the right side of the vertical line, we see five main elements. Each of these elements contains a y, an `@`, and a number. Each observation of y is paired with a value of t (represented by the number for each element). The `@` symbol means that the value of y occurred at a specific time t. For example, `y1@0` indicates that the first measurement occasion `y1` occurred when $t = 0$, which places the intercept at the beginning of the study period. We extend this syntax to address two phases of growth by adding a third line to estimate the change in slope δ_{2i} in Equation (equation10). We can implement the piecewise model in Figure figure1 part (b), which uses a single knot $k = 2$, in the following way:

```
MODEL:
delta_0 delta_1 | y1@0 y2@1 y3@2 y4@3 y5@4;
delta_0 delta_2 | y1@0 y2@0 y3@0 y4@1 y5@2;
```

The third line of syntax tells M*plus* to estimate a change in slope called `delta_2` by pairing each observation of y with the value of $(t_j - k)_+$. The `delta_0` term is included to tell M*plus* the growth segments are connected, but it does not mean `delta_0` is the intercept of the second segment. As noted in Equation (equation9), the value of $(t_j - k)_+$ is zero when $t_j \leq k$. As shown in part (b) Figure figure1, the first three observations are left of or equal to the knot at $k = 2$, represented as a dotted line in part (b) of Figure figure1. Piecewise models like the one shown in part (c) of Figure figure1 are also possible with additional lines of syntax, and we present examples in the Tutorial section.

3 The Bayesian Estimation Framework

There are multiple reasons for researchers use the Bayesian estimation framework. Bayesian methods allow researchers to incorporate background knowledge in analyses and use an estimator that does not rely on large sample theory. These features allow Bayesian methods to estimate non-identified models, which may allow the researcher to implement more phases of growth than what is possible in the frequentist framework. Bayesian estimation can also improve the accuracy of parameter estimates in nonlinear growth models (e.g., Depaoli, 2013; Wangx& McArdle, 2008). We introduce researchers to the Bayesian estimation framework here by discussing key Bayesian terminology, prior specification, the estimation process, and Bayesian model indices for model selection and evaluation. For more information, we recommend Kruschke (2014) and Depaoli (2021).

3.1 Key Terminology

Bayesian estimation addresses uncertainty about exact parameter values by treating model parameters as random variables with their own probability distributions. The results of a Bayesian analysis include an estimated probability distribution for each parameter called a *posterior distribution*. To obtain posterior distributions, the researcher must provide probability distributions called prior distributions, or *priors*. These priors represent the researcher's background knowledge about the model parameters. The prior distributions are combined with a likelihood function built from the observed data. The general process of Bayesian estimation in developmental research is to specify our background knowledge of change over time (priors), combine this knowledge with new data, and create an updated description of change over time (posterior distributions).

3.2 Prior Specification

The prior is a hugely important component of Bayesian estimation that can provide the researcher with potential influence over final parameter estimates. Prior specification is a process where each parameter in the researcher's model is assigned a probability distribution. Priors may provide more or less information depending on specification. Diffuse priors incorporate uncertainty into the analysis by providing almost no information. In contrast, an informative prior incorporates certainty into the analysis by providing information about likely values for the model parameter.

The level of informativeness of a prior reflects the level of certainty about possible values of the model parameter. As an example, consider the two-phase PGCM shown in Figure figure1, part (b) and described in Equation (equation10). The main parameters in this model are the mean intercept and slope for the first phase, β_0 and β_1, and the average change in slope in the second phase, β_2. These are mean parameters, which are commonly assigned normal distribution priors. Normal distributions are defined by a mean and a standard deviation. One way to assign a prior to β_0 is to give it a normal prior with a mean of zero and an extremely large standard deviation such as $\sigma = 10^5$. We write this formally as $\beta_0 \sim N(0, \sigma = 10^5)$. This suggests a tremendous range of values, including those as extreme as 1,000,000, are all potential values of β_0. This prior is a "diffuse" prior, meaning it does not provide much information about what values of β_0 are likely. Alternatively, the researcher may believe β_0 lies somewhere between zero and 100. To narrow the range of likely values of β_0, the researcher could specify $\beta_0 \sim N(50, \sigma = 20)$. The density of this normal distribution is almost entirely between zero and 100, with values close to 50 more likely than values far away. A similar strategy may be used to assign priors to β_1 and β_2.

The remaining parameters in the model are the coefficient covariance matrix $\mathbf{\Sigma}_\delta$ and measurement error variances $\sigma^2_{e1}, ..., \sigma^2_{e7}$. Variance parameters should not receive normal priors. In M*plus*, the options for variance prior distributions are the inverse gamma distribution or the inverse Wishart distribution. We use the diffuse M*plus* default variance priors (described in detail in the tutorial) to focus our demonstration on mean growth parameters, but interested readers can see Asparouhovxand Muthén (2021b) for guidance on how to construct informative variance priors.

Careful prior specification is always important in Bayesian estimation, but it is especially crucial for PGCMs with many phases. In the frequentist framework, models must be identified to be estimated. In PGCMs specifically, the requirements for model identification restrict the number of growth phases (Bollenx&

Curran, 2006; Flora, 2008). The Bayesian estimation framework offers an alternative to the limitations of model identification. Bayesian estimation of non-identified models (e.g., many phases of growth in piecewise growth curve models) are possible because the addition of prior information aids the estimation process and can make up for a lack of information in the observed dataset. However, careful prior specification may be especially important because the priors compensate for a lack of observed information. Priors placed on the latent covariance matrix in SEMs may be especially important for model estimation when the model is not identified. Other literature (e.g., Liu, Zhang,x& Grimm, 2016) has demonstrated how some prior specifications on this component of a growth curve model can lead to biased estimates in identified models. Some prior specifications can lead to model convergence problems and estimated non-positive definite covariance matrices, so the researcher needs to be mindful to assess the impact of their chosen priors.

In this paper, we use weakly informative priors for mean parameters. Weakly informative priors incorporate a small amount of certainty into the analysis. These priors are based on our scale of measurement and used to demonstrate one option for prior specification, but there are many others. Priors may be derived from a data-splitting technique (e.g., Depaolix& van de Schoot, 2017; Gelman, Meng,x& Stern, 1996), meta-analysis (e.g., Rietbergen, Klugkist, Janssen, Moons,x& Hoijtink, 2011), or expert consultation (e.g., Veen, Stoel, Zondervan-Zwijnenburg,x& van de Schoot, 2017). A researcher may also use data from a previous study to specify informative priors. Once all model parameters have priors specified, the researcher can estimate the model.

3.3 Model Estimation

Posterior distributions are constructed by combining priors with observed data. This combination of observed data and a prior distribution for each parameter leads to a complex, multivariate equation that usually has no simple solution. Statistical software employs iterative algorithms to solve such complex equations regardless of the estimation framework (e.g., frequentist estimation commonly uses maximum likelihood via the expectation-maximization algorithm). Bayesian estimation uses Markov chain Monte Carlo (MCMC), a technique for sampling from a probability distribution, in order to construct posterior distributions.

MCMC sampling uses an iterative process to gather a series of samples from the posterior distribution, which is then used to construct an empirical estimate of the posterior distribution. The "chain" part of MCMC refers to a record of samples from each parameter's posterior distribution. MCMC sampling in M*plus* uses two chains by default, but any number of chains can be specified. To achieve stable and meaningful posterior estimates, the MCMC chains must converge on the posterior distribution. Wildly inconsistent samples from the posterior suggest the chains have not yet converged[2], meaning the posterior distribution estimates

[2] Chains may also be slow to converge due to high autocorrelation, a phenomenon where adjacent samples in a chain are highly dependent on each other. Some re-

are not yet stable. If the posterior estimates are unstable, the researcher cannot draw valid inferences about growth in the population. Therefore, it is crucial for the researcher to assess convergence.

The first several iterations in a chain are usually unstable before the chain "finds" the posterior, and these are referred to as *burn-in* iterations. After estimating the model and discarding the burn-in iterations (M*plus* automatically discards the first half of the MCMC chain), the researcher may check convergence by inspecting plots of parameter estimates in each chain (called *trace plots*) or by using various convergence diagnostics such as the potential scale reduction factor (PSRF; Brooksx& Gelman, 1998). These two diagnostic tools are directly available in M*plus*, but additional diagnostics (such as the Geweke statistic, Geweke, 1991) can also be obtained by exporting chains to other software such as the coda package in R (Plummer, Best, Cowles,x& Vines, 2006).

Trace plots display post-burn-in iterations on the x-axis and parameter estimates on the y-axis. If a chain has converged, the trace plot should display parameter estimates with a consistent mean (i.e., a stable horizontal band) and a consistent variance (i.e., a stable height of the chain). The researcher must check trace plots for each model parameter. Chains that show inconsistent mean and variance suggest a lack of convergence. Diagnostic statistics are additional tools that are helpful for assessing convergence. The PSRF represents the ratio of within-chain to between-chain variability in post burn-in iterations for a given parameter. Ideally, all MCMC chains will converge to the same probability distribution, and the PSRF will be close to 1.0 for all parameters, but values below 1.1 are considered acceptable. M*plus* reports the model's highest PSRF throughout estimation.

3.4 Model Selection Indices

The most common model selection indices used in the Bayesian framework are the deviance information criterion (DIC; Spiegelhalter, Best, Carlin,x& van der Linde, 2002, 2014), and Bayesian information criterion (BIC; Schwarz, 1978). Both add a measure of general model fit to a penalty for model complexity. The goal is to balance good fit with parsimony. Among competing models, the model with lowest DIC (or BIC) is preferred. A third index is the posterior predictive p-value (PPP; Gelmanxet al., 1996; Meng, 1994). Unlike the DIC and BIC, the PPP is a model fit index rather than a model selection index, but it can provide useful information for model selection. These three indices can be used to choose among competing PGCM models.

The PPP is a measure of how well the model explains the observed data by evaluating simulated datasets based on the model. The contrived datasets may fit the model better or worse than the observed data, and the PPP is the proportion of simulated datasets that show more discrepancy from the model

searchers address autocorrelation by thinning the MCMC chain. We use the M*plus* default of no thinning in our tutorial, but we encourage readers who are concerned about autocorrelation in their analyses to check Depaoli (2021) or Kruschke (2014).

than the observed dataset. M*plus* conducts these simulations automatically. If simulated data consistently show worse model fit than the observed data, the model does not have good predictive accuracy. On the other hand, if simulated data based on the model always shows better fit than the real data, this also suggests model misfit. A PPP of 0.5 suggests excellent fit, with values close to zero or one suggesting model misspecification. Recent work by Cainxand Zhang (2019) suggest using a cutoff of 0.15 or lower to identify model misfit.

After using model comparison indices, it is useful to evaluate the preferred model. Recent developments in Bayesian SEM research have lead to new model fit indices including the Bayesian RMSEA (BRMSEA), the Bayesian comparative fit index (BCFI), and the Bayesian Tucker-Lewis index (BTLI). In addition to point estimates for these indices, M*plus* also provides 90% credibility intervals which can provide additional information. In particular, Asparouhovxand Muthén (2021a) suggest three interpretations for the BRMSEA credibility interval. If the full interval is below 0.06, BRMSEA suggests the model fits well, but if the full interval is above 0.06, BRMSEA indicates poor fit. If the credibility interval contains the cutoff value 0.06, the fit index is inconclusive (i.e., it cannot determine whether fit is good or bad). The credibility intervals for the other fit indices BCFI and BTLI have a similar interpretation. If the BCFI's credible interval is above 0.95, it suggests the model is well-fitting. If the interval lies below 0.95, it suggests poor fit. If the credible interval contains 0.95, the fit index is inconclusive. The interpretation of BTLI is the same. Further information on the formulation and use of these fit indices are provided in Asparouhovxand Muthén (2021a) and Garnier-Villarrealxand Jorgensen (2020).

4 Tutorial

Bayesian linear PGCMs provide a flexible approach to handling nonlinear trajectories with easily-interpretable parameters[3]. To illustrate this approach, we applied Bayesian linear PGCMs to math achievement data using the model selection approach to knot specification. There are many statistical programs that can implement Bayesian PGCMs, including Stan (Stan Development Team, 2019) and OpenBUGS (Spiegelhalter, Thomas, Best,x& Lunn, 2007), but we use M*plus* for this tutorial because of its popularity and accessibility.

4.1 Introduction to the ECLS-K Math Application

We used math achievement data from the Early Childhood Longitudinal Study, Kindergarten cohort (ECLS-K; Tourangeau, Nord, Lê, Sorongon,x& Najarian, 2009) to illustrate Bayesian PGCMs. The ECLS-K dataset is a nationally representative sample from the United States with approximately $22,000$ children who started kindergarten in the fall of 1998. The full dataset is larger than many

[3] Readers interested in the performance of the model selection approach we outline here can find a proof of concept simulation in Supplemental Material.

datasets in developmental research, so we used a random subsample of $N = 500$ children to make our demonstration more applicable to common research settings. We also ensured our sample had no missing math measurements to focus our discussion on implemention.

The ECLS-K contains measurements of math achievement from kindergarten through eighth grade. Trained evaluators assessed the children's math ability in the fall and spring of kindergarten, fall and spring of first grade, the spring of third grade, the spring of fifth grade, and the fall of eighth grade. We coded these times as 0.0, 0.5, 1.0, 1.5, 3.5, 5.5, and 8.0. This way, "1.0" corresponds to fall of first grade, "3.5" refers to a spring of third-grade measurement, and so on. The Math item response theory (IRT) scores reported in the dataset are scale scores that represent estimates of the number of items children would have answered correctly if they had taken all 174 items at all seven measurement occasions. The IRT scale provided in the ECLS-K ensures that math scores are comparable across test forms. Further details are provided by Pollack, Najarian, Rock,xand Atkins-Burnett (2005). Figure figure2 shows a scatterplot of the math achievement data across all seven measurement occasions. In the figure, math ability generally increased over time, but some periods of growth were more rapid than others. To estimate a linear PGCM to the nonlinear growth shown in Figure figure2, the first step is to determine knot placement. Unlike the simple examples shown in Figure figure1, the most appropriate knot specification is not clear. Model selection is one way to address this ambiguity.

4.2 Choosing Model Candidates

The first step for implementing Bayesian PGCMs is devising a set of model candidates to estimate. The goal is to estimate several models that differ in knot specification and use model selection indices to determine the best model. The only difference between the models should be knot specification. In the Bayesian estimation framework, the researcher may place knots on any measurement occasion except the first and last, and use up to $J - 2$ knots in total. This means for the ECLS-K data, a researcher may specify a PGCM with anywhere from one to five knots. In this section, we describe five competing models we will use to determine knot specification. The knot placement in these models break up the overall trajectory in up to six phases, visualized in Figure figure3. We discuss the rationale behind the knot placement for each model here.

The first knot specification uses a theory-driven approach. According to Piaget's classic theory of cognitive development (Flavell, 1963), children occupy the preoperational stage of development from ages two to seven. The concrete operational stage occurs from ages seven to eleven, and the formal operational stage begins at twelve years old. These stages represent an increase in childrens' ability to think abstractly, and a researcher could argue these stages relate to math development. A researcher may apply these phases of development to the ECLS-K data by placing knots at $k_1 = 1.5$ and $k_2 = 5.5$. For this specification, the first phase of growth corresponds to preoperational development, the second

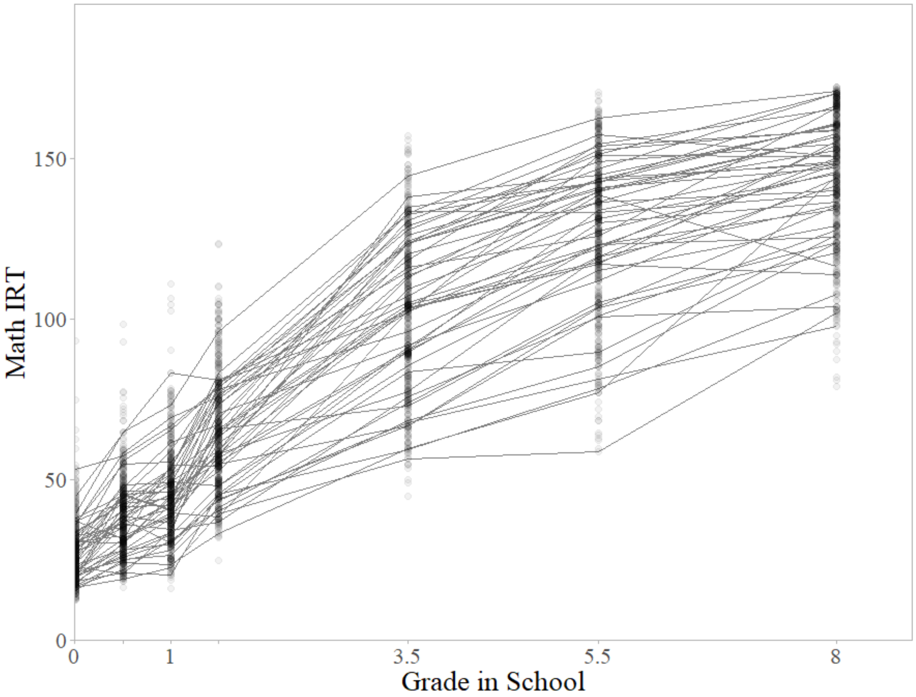

Figure 2. The development of math achievement in the ECLS-K dataset. “Math IRT” refers to the repeated measures outcome variable indicating math achievement in the ECLS-K, dots represent individual children’s scores, and grade in school ranges from zero (representing fall of kindergarten) to eight (representing fall of eighth grade). Lines illustrate the trajectory over time for a random subsample of n=50 children.

to the concrete operational stage of development, and the third to the formal operational stage. Model 1 implements this knot specification.[4]

The remaining four models use a data-driven approach to derive knot placement. Model 2 divides the trajectory into four almost equally-sized segments by placing knots at $k_1 = 1.5, k_2 = 3.5$, and $k_3 = 5.5$. Each phase of growth encompasses 2 years on average, and is the closest to equally-sized segments that is possible with the timing of measurements. The third model places knots at every other measurement occasion: $k_1 = 0.5, k_2 = 1.5$, and $k_3 = 5.5$. The fourth model increases complexity to four knots. The scatterplot in Figure figure2 may be interpreted to show no meaningful change in growth between the first and second measurement compared to the growth between the second and third. To treat the whole of kindergarten as a single phase of growth while allowing unique phases between all other measurements, Model 4 implements four knots $k_1 = 1.0$, $k_2 = 1.5$, $k_3 = 3.5$, and $k_4 = 5.5$. Finally, Model 5 implements all possible knots $k_1 = 0.5$, $k_2 = 1.0$, $k_3 = 1.5$, $k_4 = 3.5$, and $k_5 = 5.5$. This fifth model suggests that the rate of change between every single measurement is meaningfully different. In the following sections, we demonstrate how to implement PGCMs in the Bayesian framework and how to use Bayesian model selection indices to choose the most appropriate model.

4.3 Prior Specification Strategy

Each parameter in a model requires a prior. For linear PGCMs, these parameters include coefficient means (for the intercept, first slope, and changes in slope), variances of the coefficients, covariances of the coefficients, and measurement errors. We employed a combination of weakly informative and diffuse priors.

The intercept mean reflects the mean math achievement score when $t = 0$, or the fall of kindergarten. Visual inspection of the data scatterplot showed no negative values at the first measurement occasion, so the default prior centered on zero did not seem appropriate. Instead, we specified a “weakly informative” prior as Normal($\mu = 25$, $\sigma^2 = 100$). The mean of all scores at the first timepoint appears close to 25 in the scatterplot, and setting the variance to 100 reflects our uncertainty about this exact mean value.

After setting the prior for the intercept, we took a more general approach to the priors for the other coefficient means. Setting priors for PGCMs comes with an additional challenge when using model selection indices to determine knot placement: Priors should be kept as consistent as possible across models to ensure that differences in model selection indices are due to knot placement alone. The ECLSK dataset includes math IRT values ranging from approximately 10 to 174. Because of this range of values, the full width of the default priors did not

[4] We are using this example of Piaget’s stages simply for illustrative purposes. We make no claims about whether these are viable stages of development, nor how they may (or may not) relate to math development. Instead, we wanted to form a concrete example that would be easy for readers to follow in order to highlight aspects of conducting the analysis.

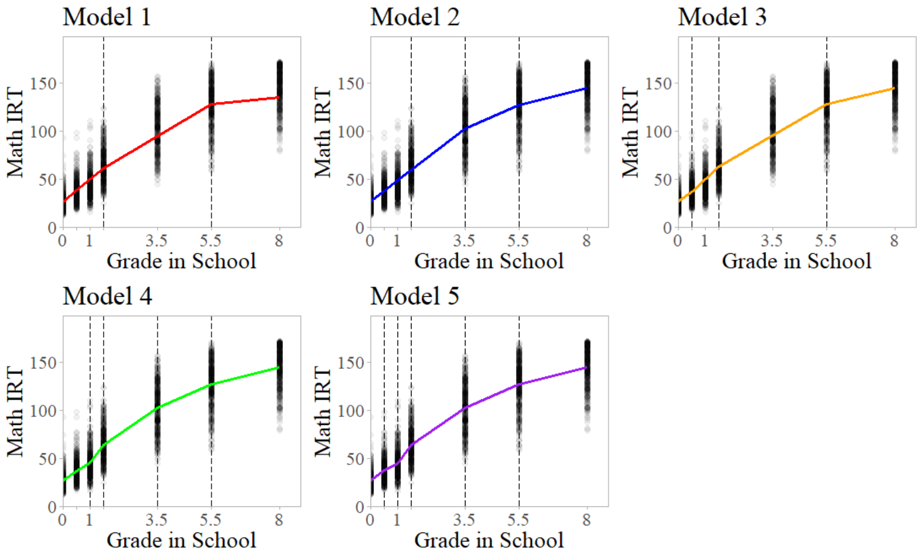

Figure 3. Five candidate knot specifications for the ECLS-K dataset. "Math IRT" refers to the repeated measures outcome variable indicating math achievement in the ECLS-K, dots represent individual children's scores, and grade in school ranges from zero (representing fall of kindergarten) to eight (representing fall of eighth grade). Knot location is indicated by the dashed vertical lines, and each model uses unique phases of growth to capture the development of math achievement. These models range in complexity from the three-phase Model 1 to the six-phase Model 5. Colored lines indicate the estimated mean trajectory according to each model.

seem useful. A visual inspection of the data scatterplot shows that rate of growth changed over time, but did not appear to exceed 30 IRT points within a single year. In order to keep the priors for the slope means consistent, we assigned a normal prior with N($\mu = 0$, $\sigma^2 = 400$) to each one. This reflects our belief that a range of linear slope values from -60 to 60 are possible, with slopes closer to zero more likely. In substantive terms, this means we expected childrens' math IRT score to change by some value in the -60 to 60 range each year for the first segment of growth, and the rate of change itself would never change by more than 60 points.

Coefficient variances, covariances, and residual variances were also estimated. Because we did not have a clear substantive reason to alter the priors for these parameters, we used M*plus* default settings. For the coefficient covariance matrix $\mathbf{\Sigma}_\delta$, M*plus* uses an inverse Wishart prior with a $\mathbf{0}$ scale matrix and $-p-1$ degrees of freedom, where p is the number of latent growth factors. Residual variances receive an inverse gamma prior defined as IG(-1, 0). Next, we describe how to implement these priors in M*plus*.

4.4 Implementing Linear PGCMs in Bayesian Software

Estimating Bayesian PGCMs in M*plus* requires an input file with five sections: data information (including the `DATA` and `VARIABLE` commands), the model itself (under `MODEL`), estimation details (under `ANALYSIS`), prior specification (under `MODEL PRIORS`), and output details (including `PLOT` and `OUTPUT` commands). We present the syntax for Model 5 here, but readers can implement any of the other candidate models with minor edits to the `MODEL` and `MODEL PRIORS` sections. We begin with the `MODEL` section.

The equation for Model 5 can be written,

$$\begin{aligned} y_{ij} = \delta_{0i} + \delta_{1i}t_{ij} + \delta_{2i}(t_{ij} - 0.5)_+ + \delta_{3i}(t_{ij} - 1.0)_+ \\ + \delta_{4i}(t_{ij} - 1.5)_+ + \delta_{5i}(t_{ij} - 3.5)_+ + \delta_{6i}(t_{ij} - 5.5)_+ + e_{ij}, \end{aligned} \tag{14}$$

which translates to the following syntax:

```
MODEL:
delta_0 delta_1 | y1@0 y2@0.5 y3@1.0 y4@1.5 y5@3.5 y6@5.5 y7@8.0;
delta_0 delta_2 | y1-y2@0 y3@0.5 y4@1.0 y5@3.0 y6@5.0 y7@7.5;
delta_0 delta_3 | y1-y3@0 y4@0.5 y5@2.5 y6@4.5 y7@7.0;
delta_0 delta_4 | y1-y4@0 y5@2 y6@4 y7@6.5;
delta_0 delta_5 | y1-y5@0 y6@2 y7@4.5;
delta_0 delta_6 | y1-y6@0 y7@2.5;

[delta_0-delta_6] (beta_0-beta_6);
```

The first line after the `MODEL` command tells M*plus* the timing of all seven measurement occasions, and tells M*plus* to use the timing to estimate the intercept and slope of the first phase. The next line contains `delta_2` and tells M*plus* to estimate the change in slope for the second phase of growth, starting at $t_{ij} = 0.5$. This line of syntax assigns the values of $(t_{ij} - 0.5)_+$ to each measurement occasion. For the first two measurements, the values are zero. The five time points after the knot are $(t_{ij} - 0.5)_+$ for $t_{ij} = 1.0$, 1.5, 3.5, 5.5, 8.0. For example, the fifth measurement `y5` occurs when $t_{ij} = 3.5$. The value of $(3.5 - 0.5)_+ = 3.0$, the value of time assigned to `y5`. The next four lines of syntax repeat this process for the remaining four phases of growth. In the final line, the square brackets refer to the means of the parameters inside and parentheses contain labels for these means. This line of syntax indicates the mean of the growth coefficients $\delta_{0i}, \delta_{1i}, ..., \delta_{6i}$ are labelled $\beta_0, \beta_1, ..., \beta_6$.

The next section of code tells M*plus* how to estimate the PGCM described above:

```
ANALYSIS:
  ESTIMATOR=BAYES;
  FBITERATIONS = 100000;
  BSEED = 1979;
```

The first line under the `ANALYSIS` heading tells M*plus* that we want to use Bayesian estimation. The next command, `FBITERATIONS = 100000`, requests 100,000 MCMC iterations. This number was selected based on the number of iterations required for Model 5 to converge according to PSRF. Next, the `BSEED = 1979` command provides M*plus* a "seed" number to begin implementing the MCMC algorithm. We provide one here so the reader may replicate our results, but M*plus* can generate its own if one is not provided. If the model reaches convergence, the seed number does not influence model results.

Next, priors are specified in the `MODEL PRIORS` section:

```
MODEL PRIORS:
  beta_0 ~ N(25, 100);
  beta_1-beta_6 ~ N(0, 400);
```

The first line under the `MODEL PRIORS` heading tells M*plus* that the mean of the intercept is normally distributed around 25 with a variance of 100, or $\beta_0 \sim N(25, 100)$. The next line assigns a prior to the means of all six slope parameters, β_1, $\beta_2, ..., \beta_6 \sim N(0, 400)$. We do not explicitly assign variance priors here, so M*plus* will use its diffuse defaults. Once each candidate model's input file has been written in M*plus*, we can estimate the models and use the results to conduct model selection.

4.5 Model Selection

The five candidate models provide slightly different descriptions of change in math achievement over time, as illustrated in Figure figure3. The next step of the process is to examine Bayesian model selection indices summarized in Table table1 to choose the best model. For the PPP, values close to 0.500 suggest excellent fit, and values close to zero or one suggest poor fit. For the DIC and BIC, the lowest value suggests the best balance of model fit with model complexity. In this case, the PPP and DIC suggest that Model 5 is the best model. However, the BIC suggests the best model is Model 4. For the purposes of this illustration, we consider Model 5 the optimal model.

Table 1. Model selection indices and approximate model fit indices.

Fit indices for model selection					
Fit Index	Model 1	Model 2	Model 3	Model 4	Model 5
PPP	0.000	0.000	0.000	0.001	0.468
DIC	26533.70	26245.22	26486.75	26023.36	25992.85
BIC	26626.34	26371.32	26615.68	26229.44	26458.46
Approximate fit indices for evaluating Model 5					
Fit Index	Point Estimate	90% Credible Interval			
BRMSEA	0.031	[0.000, 0.158]			
BCFI	1.000	[0.995, 1.000]			
BTLI	0.997	[0.925, 1.000]			

Note. PPP = posterior predictive p-value; DIC = deviance information criterion; BIC = Bayesian information criterion. Each model uses a different knot specification to create unique phases of growth in the development of math achievement. These models range in complexity from the three-phase Model 1 to the six-phase Model 5. BRMSEA = Bayesian root mean square error of approximation; BCFI = Bayesian comparative fit index; BTLI = Bayesian Tucker-Lewis index.

We can evaluate the quality of Model 5 using the BRMSEA, BCFI, and BTLI. The point estimates and 90% credible intervals for these fit indices are reported in Table table1. We focus on the credibility intervals to keep our interpretation consistent with Asparouhovxand Muthén (2021a). For the BRMSEA, values below 0.06 indicate good model fit. The BRMSEA's credible interval ranged from zero to 0.158. Because the credible interval contained 0.06, this fit index is inconclusive. Next, we consider the BCFI and BTLI, where values between 0.95 and 1.00 suggest excellent fit. For the BCFI, the credible interval ranged from 0.995 to 1.000, and the BTLI credible interval ranged from 0.925 to 1.000. The BCFI results suggest good model fit because the credible interval is entirely above 0.95. However, the BTLI credible interval contains the cutoff value and we consider this fit index inconclusive. In summary, one fit index suggested good fit but the other two were inconclusive.

Next, we describe and interpret the parameter estimates for Model 5, which are summarized in Table table2. Recall that β_0 and β_1 refer to the mean intercept and linear slope for the first phase of growth, and each following coefficient from β_2 to β_6 refer to an average change in slope. In other words, the mean rate of change in the second slope is not the estimate of β_2 alone, but the sum of $\beta_1 + \beta_2$. These changes accumulate across the phases of growth. For ease of interpretation, Table table2 contains a "Total Slope" column that reflects the rate of change in all six phases. For example, the total slope in the first phase of growth (from fall of kindergarten to spring of kindergarten) is 21.28. This value means that children's math achievement would increase by an average of 21.28 points in one year if growth remained constant. The rate of growth in the second phase of development (spring of kindergarten to fall of first grade) decreases by -6.89, resulting in a rate of 14.39. In contrast, the third phase of development (from fall to spring of first grade) showed an increase in growth of 22.95, leading to a total slope of 37.34. The fourth phase of growth addresses two years of growth from spring of first grade to the spring of third grade. The average rate of change per year in this phase decreased from the previous phase by -18.02, meaning childrens' math achievement increased by 19.32 per year on average. In the fifth phase, growth slowed again by -6.90, creating a 12.42 increase in math achievement per year from spring of third grade to the spring of fifth grade. In the final phase from spring of fifth grade to fall of eighth grade, growth slowed by -5.61 to a rate of change of 6.81. Overall, growth was most rapid in the third phase, which was also when the most dramatic change in the rate of development occurred. Table table2 also reports measurement error variance at all seven timepoints, which ranged from 10.60 at the first measurement to 39.84 at the fifth measurement.

We present the covariance matrix of the growth coefficients $\mathbf{\Sigma}_\delta$ in the lower portion of Table table2. The individual elements in this matrix are not typically of interest, but we can note that each coefficient covaries with the others. There are particularly strong negative covariances between δ_{1i} and δ_{2i}, δ_{2i} and δ_{3i}, and δ_{3i} and δ_{4i}. In other words, the rate of change in one phase of growth tends to increase when the next phase decreases, and this relationship is particularly strong across the first, second, third, and fourth phases. We can also note that the change in slope for the second, third, and fourth phases show the highest variance of all latent growth coefficients.

4.6 Final Results

In this application, we devised a set of candidate models and used model selection indices to determine the most adequate model. The final model was Model 5, which treats the time between each measurement occasion as a distinct phase of growth with its own unique rate of change. The BCFI suggested good model fit, but other approximate fit indices were inconclusive. We interpreted these results as not suggesting excellent fit, but not suggesting substantial misfit either. According to this model, the most rapid growth occurred in the third phase, from fall to spring of first grade. After this phase, the rate of growth decreased in each subsequent phase.

Table 2. Model 5 parameter estimates for latent coefficient means and measurement errors at each timepoint.

Growth Factor Estimates, Standard Errors, and Phase-Specific Slopes

Coefficient	Estimate(SE)	Total Slope
β_0	27.25(0.43)	
β_1	21.28(0.65)	21.28
β_2	-6.89(1.16)	14.39
β_3	22.95(1.37)	37.34
β_4	-18.02(1.09)	19.32
β_5	-6.90(0.51)	12.42
β_6	-5.61(0.41)	6.81

Error Variances

σ^2_{e1}	10.60(7.22)
σ^2_{e2}	11.78(9.08)
σ^2_{e3}	20.79(11.29)
σ^2_{e4}	32.56(21.58)
σ^2_{e5}	39.84(25.54)
σ^2_{e6}	30.56(20.15)
σ^2_{e7}	39.30(29.67)

Covariance Matrix Σ_δ

Coefficient	δ_0	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6
δ_0	82.76						
δ_1	16.00	116.89					
δ_2	18.311	-159.98	354.63				
δ_3	-15.54	78.13	-239.28	438.00			
δ_4	-3.56	-21.58	38.03	-274.32	299.31		
δ_5	-22.39	-13.93	8.03	0.68	-40.00	68.88	
δ_6	-4.55	-9.14	5.05	-7.25	9.07	-19.54	40.18

Note. β_0 = mean baseline Math IRT score; β_1 = average linear slope of the first phase of growth; β_2 = average change in slope for the second phase of growth; $\beta_3, \beta_4, \beta_5, \beta_6$ refer to cumulative changes in slope for the third through sixth phases of growth. σ^2_{e1} through σ^2_{e7} refer to measurement error variance at the first through seventh measurement occasions. δ_0 refers to the latent intercept; δ_1 refers to latent slope in the first phase; δ_2 through δ_6 refer to cumulative changes in slope across phases of growth.

5 Discussion

Our goal for this paper was to demonstrate how linear PGCMs are a flexible extension of linear GCMs, with models addressing three or more phases of growth possible in the Bayesian estimation framework. This added flexibility can dramatically increase the number of possible models, and we outlined the process of specifying candidate models and using model selection indices to choose the final model. To provide a simple and accessible tutorial to implement Bayesian linear PGCMs, several extensions and technical features were not addressed in detail. We discuss extensions of the presented model and some technical cautions here.

5.1 Potential Extensions of the Current Work

In this tutorial, we focused on Bayesian linear PGCMs due to their simple coefficient interpretations in order to provide an introduction to the field of piecewise growth models. As noted previously, there are several newer extensions of the presented model, which we encourage readers to explore. These extensions include piecewise models that directly estimate knot placements (Kohlixet al., 2015; Lockxet al., 2018), employ covariates (Lamm, 2022), or capture bivariate piecewise trajectories (Peraltaxet al., 2022). Additionally, PGCMs with higher-order polynomials (e.g., cubic) or inherently nonlinear functions (e.g., exponential) are also possible. Harring, Strazzeri,xand Blozis (2021) provide a discussion of these extensions in the context of PGCMs with random knots. Additionally, Rioux, Stickley,xand Little (2021) demonstrate PGCMs with discontinuities (i.e., gaps in the growth trajectory) to address cancelled data collection waves. Piecewise models with inconsistent measurement timing can be easily addressed in the multilevel modeling framework, where they are commonly called splines. Harezlak, Ruppert,xand Wand (2018) provide a thorough introduction, including Bayesian extensions.

5.2 Prior Cautions

Implementing linear PGCMs in the Bayesian estimation framework frees the researcher from model identification requirements inherent in frequentist estimation because prior distributions can compensate for additional measurement occasions. However, implementing non-identified models in the Bayesian framework must be done cautiously. The prior placed on the latent covariance matrix can be especially influential on model results, as shown by Liuxet al. (2016) and Depaoli, Liu,xand Marvin (2021). The specific implementation of the inverse Wishart prior (the M*plus* default) can also impact results in unexpected ways. A key method of assessing how sensitive results are to prior specification is to conduct a prior sensitivity analysis. In a prior sensitivity analysis, the researcher estimates the chosen model under a set of alternative prior conditions, and discusses how robust the model results are. We recommend van Erp, Mulder,xand Oberski (2018), van de Schoot, Veen, Smeets, Winter,xand Depaoli (2020), and

Depaoli, Winter,xand Visser (2020) for thorough demonstrations. When implemented conscientiously, we believe Bayesian linear PGCMs can be a useful class of models because they frame development as phases of growth with simple parameters. This is in contrast to other GCM extensions with parameters that may be challenging to interpret (e.g., a cubic coefficient).

5.3 Concluding Remarks

Developmental researchers study within-person change over time in many settings. A linear GCM easily captures growth that follows a straight line, but may not capture substantively important nonlinear changes. In this paper we presented a tutorial to estimate Bayesian PGCMs, which can handle complex nonlinearity with simple parameter interpretations. The goal of this tutorial was specifically aimed to act as a precursor to more advanced methodological work (e.g., Kohlixet al., 2015), which we recommend the interested reader to explore as a subsequent resource to this work.

Applying this model to math achievement data allowed us to examine when development accelerated or slowed, and highlighted phases of growth with more variability than others. Bayesian PGCMs provide researchers with a useful model that can capture nonlinear growth using parameters that are straightforward to interpret. While research is needed to better understand the impact of different covariance priors, the Bayesian linear PGCM can provide interesting results when implemented thoughtfully.

References

Akaike, H. (1992). Information theory and an extension of the maximum likelihood principle. In *Breakthroughs in statistics* (pp. 610–624). Springer.

Asparouhov, T., & Muthén, B. (2021a). Advances in Bayesian model fit evaluation for structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *28*, 1–14. doi: https://doi.org/10.1080/10705511.2020.1764360

Asparouhov, T., & Muthén, B. (2021b). *Bayesian analysis of latent variable models using mplus (version 5).* m*Plus*. Retrieved from https://www.statmodel.com/download/BayesAdvantages18.pdf

Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological bulletin*, *107*(2), 238.

Bollen, K. A., & Curran, P. J. (2006). *Latent curve models: A structural equation perspective* (Vol. 467). John Wiley & Sons.

Brooks, S. P., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*(4), 434–455.

Browne, M. W., & Cudeck, R. (1992). Alternative ways of assessing model fit. *Sociological methods & research*, *21*(2), 230–258.

Cain, M. K., & Zhang, Z. (2019). Fit for a Bayesian: An evaluation of ppp and dic for structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *26*, 39–50. doi: https://doi.org/10.1080/10705511.2018.1490648

Depaoli, S. (2013). Mixture class recovery in GMM under varying degrees of class separation: Frequentist versus Bayesian estimation. *Psychological Methods*, *18*(2), 186–219. doi: https://doi.org/10.1037/a0031609

Depaoli, S. (2021). *Bayesian structural equation modeling.* The Guilford Press.

Depaoli, S., Liu, H., & Marvin, L. (2021). Parameter specification in Bayesian CFA: An exploration of multivariate and separation strategy priors. *Structural Equation Modeling: A Multidisciplinary Journal*, *28*(5), 1–17. doi: https://doi.org/10.1080/10705511.2021.1894154

Depaoli, S., & van de Schoot, R. (2017). Improving transparency and replication in Bayesian statistics: The WAMBS-checklist. *Psychological methods*, *22*(2), 240. doi: https://doi.org/10.1037/met0000065

Depaoli, S., Winter, S. D., & Visser, M. (2020). The importance of prior sensitivity analysis in Bayesian statistics: Demonstrations using an interactive Shiny app. *Frontiers in Psychology*, *11*. doi: https://doi.org/10.3389/fpsyg.2020.608045

Finkel, D., Reynolds, C. A., McArdle, J. J., & Gatz, N. L., M. Pedersen. (2003). Latent growth curve analyses of accelerating decline in cognitive abilities in late adulthood. *Developmental Psychology*, *39*(3), 535–550. doi: https://doi.org/10.1037/0012-1649.39.3.535

Flavell, J. H. (1963). *The developmental psychology of Jean Piaget.* D van Nostrand.

Flora, D. B. (2008). Specifying piecewise latent trajectory models for longitudinal data. *Structural Equation Modeling: A Multidisciplinary Journal*, *15*(3), 513–533. doi: https://doi.org/10.1080/10705510802154349

Garnier-Villarreal, M., & Jorgensen, T. D. (2020). Adapting fit indices for Bayesian structural equation modeling: Comparison to maximum likelihoods. *Psychological Methods*, *25*(1), 46–70. doi: https://doi.org/10.1037/met0000224

Gaudreau, P., Louvet, B., & Kljajic, K. (2018). The performance trajectory of physical education students differs across subtypes of perfectionism: A piecewise growth curve model of the 2 × 2 model of perfectionism. *Sport, Exercise, and Performance Psychology*, *8*(2), 223–237. doi: https://doi.org/10.1037/spy0000138

Gelman, A., Meng, X.-L., & Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 733–760.

Geweke, J. F. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments* (Tech. Rep.). Federal Reserve Bank of Minneapolis.

Grimm, K. J., & Ram, N. (2009). Nonlinear growth models in Mplus and SAS. *Structural Equation Modeling*, *16*(4), 676–701. doi: https://doi.org/10.1080/10705510903206055

Harezlak, J., Ruppert, D., & Wand, M. P. (2018). *Semiparametric regression with r*. Springer.

Harring, J. R., Strazzeri, M. M., & Blozis, S. A. (2021). Piecewise latent growth models: beyond modeling linear-linear processes. *Behavior Research Methods*, *53*(2), 593–608.

Kohli, N., Hughes, J., Wang, C., Zopluoglu, C., & Davison, M. L. (2015). Fitting a linear–linear piecewise growth mixture model with unknown knots: A comparison of two common approaches to inference. *Psychological methods*, *20*(2), 259.

Kruschke, J. (2014). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press.

Lamm, R. Z. (2022). *Incorporation of covariates in bayesian piecewise growth mixture models* (Unpublished doctoral dissertation). University of Minnesota.

Liu, H., Zhang, Z., & Grimm, K. J. (2016). Comparison of inverse Wishart and separation-strategy priors for Bayesian estimation of covariance parameter matrix in growth curve analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(3), 354–367. doi: https://doi.org/10.1080/10705511.2015.1057285

Lock, E. F., Kohli, N., & Bose, M. (2018). Detecting multiple random changepoints in bayesian piecewise growth mixture models. *Psychometrika*, *83*, 733–750.

Marks, A., & Coll, C. G. (2007). Psychological and demographic correlates of early academic skill development among American Indian and Alaska Native youth: A growth modeling study. *Developmental Psychology*, *43*(3), 663–674. doi: https://doi.org/0.1037/0012-1649.43.3.663

Meng, X.-L. (1994). Posterior predictive p-values. *The annals of statistics*, *22*(3), 1142–1160.

Meredith, W., & Tisak, J. (1990). Latent curve analysis. *Psychometrika*, *55*(1), 107–122.

Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural equation modeling: A multidisciplinary Journal*, *14*(4), 535–569.

Peralta, Y., Kohli, N., Lock, E. F., & Davison, M. L. (2022). Bayesian modeling of associations in bivariate piecewise linear mixed-effects models. *Psychological methods*.

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News*, *6*(1), 7–11. Retrieved from `https://journal.r-project.org/archive/`

Pollack, J. M., Najarian, M., Rock, D. A., & Atkins-Burnett, S. (2005). Early childhood longitudinal study, kindergarten class of 1998-99 (ECLS-K): Psychometric report for the fifth grade. NCES 2006-036. *National Center for Education Statistics*.

Rietbergen, C., Klugkist, I., Janssen, K. J., Moons, K. G., & Hoijtink, H. J.

(2011). Incorporation of historical data in the analysis of randomized therapeutic trials. *Contemporary Clinical Trials*, *32*(6), 848–855. doi: https://doi.org/10.1016/j.cct.2011.06.002

Rioux, C., Stickley, Z. L., & Little, T. D. (2021). Solutions for latent growth modeling following COVID-19-related discontinuities in change and disruptions in longitudinal data collection. *International Journal of Behavioral Development*, *45*(5), 463–473. doi: https://doi.org/10.1177/01650254211031631

Schillin, O. K., Deeg, D. J. H., & Huisman, M. (2018). Affective well-being in the last years of life: The role of health decline. *Psychology and Aging*, *33*(5), 739–753. doi: https://doi.org/10.1037/pag0000279

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 461–464.

Seider, S., Clark, S., Graves, D., Kelly, L. L., Soutter, M., El-Aamin, A., & Jennett, P. (2019). Black and latinx adolescents' developing beliefs about poverty and associations with their awareness of racism. *Developmental Psychology*, *55*(3), 509–524. doi: https://doi.org/10.1037/dev0000585

Shono, Y., Edwards, M. C., Ames, S. L., & Stacy, A. W. (2018). Trajectories of cannabis-related associative memory among vulnerable adolescents: Psychometric and longitudinal evaluations. *Developmental Psychology*, *54*(6), 1148–1158. doi: https://doi.org/10.1037/dev0000510

Spiegelhalter, D., Best, N. G., Carlin, B. P., & van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society*, *64*(4), 583–639. doi: https://doi.org/10.1111/1467-9868.00353

Spiegelhalter, D., Best, N. G., Carlin, B. P., & van der Linde, A. (2014, April). The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*(3), 485–493. doi: https://doi.org/10.1111/rssb.12062

Spiegelhalter, D., Thomas, A., Best, N., & Lunn, D. (2007). OpenBUGS user manual. *Version*, *3*(2), 2007.

Stan Development Team. (2019). *Stan modeling language users guide and reference manual, version 2.28.0.* Retrieved from `http://mc-stan.org/`

Steiger, J. H., & Lind, J. C. (1980). Statistically based tests for the number of common factors. In *The annual meeting of the Psychometric Society. Iowa City, IA.*

Tourangeau, K., Nord, C., Lê, T., Sorongon, A. G., & Najarian, M. (2009). Early childhood longitudinal study, kindergarten class of 1998-99 (ECLS-K): Combined user's manual for the ECLS-K eighth-grade and K-8 full sample data files and electronic codebooks. NCES 2009-004. *National Center for Education Statistics*.

Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika*, *38*(1), 1–10.

van de Schoot, R., Veen, D., Smeets, L., Winter, S. D., & Depaoli, S. (2020). A tutorial on using the wambs checklist to avoid the misuse of bayesian statistics. *Small Sample Size Solutions: A Guide for Applied Researchers* *and Practitioners; van de Schoot, R., Miocevic, M., Eds*, 30–49.

van Erp, S., Mulder, J., & Oberski, D. L. (2018). Prior sensitivity analysis in default Bayesian structural equation modeling. *Psychological Methods*, *23*(2), 363–388. doi: https://doi.org/10.3758/s13428-011-0088-6

Vargas Lascano, D. I., Galambos, N. L., Krahn, H. J., & Lachman, M. E. (2015). Growth in perceived control across 25 years from the late teens to midlife: The role of personal and parents' education. *Developmental Psychology*, *51*(1), 124–135. doi: https://doi.org/10.1037/a0038433

Veen, D., Stoel, D., Zondervan-Zwijnenburg, M., & van de Schoot, R. (2017). Proposal for a five-step method to elicit expert judgment. *Frontiers in psychology*, *8*, 2110. doi: https://doi.org/10.3389/fpsyg.2017.02110

Wang, L., & McArdle, J. J. (2008). A simulation study comparison of Bayesian estimation with conventional methods for estimating unknown change points. *Structural Equation Modeling: A Multidisciplinary Journal*, *15*(1), 52–74. doi: https://doi.org/10.1080/10705510701758265

Zimmer-Gembeck, M., Gardner, A. A., Hawes, T., Maters, M. R., Waters, A. M., & Farrell, L. J. (2021). Rejection sensitivity and the development of social anxiety symptoms during adolescence: A five-year longitudinal study. *International Journal of Behavioral Development*, *45*(3), 204–215. doi: https://doi.org/10.1177/0165025421995921

Appendix A Proof of Concept Simulation

To demonstrate the utility of treating the knot specification problem as a model selection problem, we performed a simulation study. The purpose of this study was twofold. First, we aimed to evaluate the performance of Bayesian model fit indices in selecting the correct knot specification. Second, we assessed the accuracy of the model parameter estimates.

Appendix A.1 Simulation Design

We considered five population models based on the five candidate models used in analyzing the ECLS-K (see Figure figure3). There are seven measurement occasions, coded like the ECLS-K such that $t = 0.0, 0.5, 1.0, 1.5, 3.5, 5.5, 8.0$. For Population Model 1, growth was split into three phases, with knots at $t = 1.5, 5.5$. Both Population Model 2 and Population Model 3 had four phases in the growth trajectory but different knot placements. Population Model 2 used knots at $t = 1.5, 3.5, 5.5$ and Population Model 3 used knots at $t = 0.5, 1.5, 5.5$. Population Model 4 implemented five phases of growth with knots at $t = 1.0, 1.5, 3.5, 5.5$. Population Model 5 split the trajectory into six unique phases of growth, with as many knots as possible at $t = 0.5, 1.0, 1.5, 3.5, 5.5$.

The distribution of the latent growth factors for the five population models were based on the ECLS-K estimates. The distribution of growth factors for Population Model 1 through 5 are described in the following Equations (1) through (5):

$$\begin{bmatrix}\delta_0\\ \delta_1\\ \delta_2\\ \delta_3\end{bmatrix} \sim MVN\left(\begin{bmatrix}26.71\\ 23.10\\ -6.50\\ -9.95\end{bmatrix}, \begin{bmatrix}75.00 & & & \\ 29.96 & 30.65 & & \\ -27.03 & -20.70 & 20.95 & \\ -14.36 & -19.69 & 2.57 & 39.17\end{bmatrix}\right), \tag{15}$$

$$\begin{bmatrix}\delta_0\\ \delta_1\\ \delta_2\\ \delta_3\\ \delta_4\end{bmatrix} \sim MVN\left(\begin{bmatrix}26.93\\ 21.83\\ -0.44\\ -8.96\\ -5.61\end{bmatrix}, \begin{bmatrix}76.71 & & & & \\ 28.20 & 26.45 & & & \\ -14.68 & -5.82 & 17.83 & & \\ -20.18 & -21.62 & -16.97 & 63.61 & \\ -4.46 & -7.83 & 5.58 & -19.30 & 38.86\end{bmatrix}\right), \tag{16}$$

$$\begin{bmatrix}\delta_0\\ \delta_1\\ \delta_2\\ \delta_3\\ \delta_4\end{bmatrix} \sim MVN\left(\begin{bmatrix}27.26\\ 19.34\\ 6.68\\ -9.83\\ -9.55\end{bmatrix}, \begin{bmatrix}85.00 & & & & \\ 10.38 & 62.49 & & & \\ 20.30 & -36.14 & 64.72 & & \\ -27.56 & -16.60 & -30.99 & 52.06 & \\ -15.15 & -15.70 & -2.80 & 0.80 & 38.89\end{bmatrix}\right), \tag{17}$$

$$\begin{bmatrix}\delta_0\\ \delta_1\\ \delta_2\\ \delta_3\\ \delta_4\\ \delta_5\end{bmatrix} \sim MVN\left(\begin{bmatrix}27.57\\ 18.22\\ 17.75\\ -16.63\\ -6.90\\ -5.60\end{bmatrix}, \begin{bmatrix}76.61 & & & & & \\ 29.66 & 29.19 & & & & \\ -8.32 & 9.03 & 171.87 & & & \\ -5.40 & -27.60 & -197.26 & 270.61 & & \\ -22.58 & -10.42 & 9.72 & -41.46 & 64.09 & \\ -4.74 & -6.90 & -3.12 & 6.87 & -16.44 & 38.54\end{bmatrix}\right), \tag{18}$$

$$\begin{bmatrix}\delta_0\\ \delta_1\\ \delta_2\\ \delta_3\\ \delta_4\\ \delta_5\\ \delta_6\end{bmatrix} \sim MVN\left(\begin{bmatrix}27.25\\ 21.29\\ -6.89\\ 22.95\\ -18.02\\ -6.50\\ -5.61\end{bmatrix}, \begin{bmatrix}82.76 & & & & & & \\ 16.00 & 116.89 & & & & & \\ 18.31 & -159.98 & 116.89 & & & & \\ -15.54 & 78.13 & -239.28 & 438.00 & & & \\ -3.56 & -21.58 & 38.03 & -274.32 & 299.31 & & \\ -22.39 & -13.93 & 8.03 & 0.68 & -40.00 & 68.88 & \\ -4.55 & -9.14 & 5.05 & -7.25 & 9.07 & -19.54 & 40.18\end{bmatrix}\right). \tag{19}$$

Measurement error variances were set to 1.0. For each population model, we simulated 500 datasets with sample size $N = 500$. For each generated dataset, we fit all five candidate models, including the true model and the models with misspecified knot placement. We estimated each model using Bayesian estimation methods with the same setup (i.e., prior specification, number of iterations) as that in analyzing the ECLS-K data. For each replication and model fitted, we recorded the model parameter estimates and the Bayesian model fit indices PPP, DIC, and BIC.

Appendix A.2 Results

Table center3 shows the selection rates for the PPP, DIC, and BIC within all five population models. To compute the selection rate of the PPP, we found the proportion of replications where a given model had a PPP value closer to 0.5 than all competing models. The selection rate of the DIC was the proportion of replications where a given model had a lower DIC than all competing models. The BIC selection rate was computed the same way[5].

When the data were generated from Model 1, the selection rate of PPP for the correct model (i.e., Model 1) was around 11%. Similarly, when the data were generated from Model 2, the selection rate of PPP for the correct model was 16%. When data were generated from Model 3, PPP selected Model 3 28% of the time, and when data were generated from Model 4, PPP selected Model 4 28% of the time. For Population Model 5, the PPP selected the correct model (i.e., Model 5) 100% of the time. Based on the simulation results, the PPP tends to select more complex models. However, the DIC and BIC were more effective at selecting the correct model. When the data were generated from Model 1, the DIC selected the correct model (i.e., Model 1) 86% of the time. When the data were generated from Model 2, DIC selected Model 2 92% of the time, and when data were generated from Model 3, DIC selected Model 3 98% of the time. For data generated from Model 4, the DIC selected Model 4 with a 93% selection rate. Lastly, when data were generated from Model 5, the DIC selected Model 5 100% of the time. The BIC showed generally high selection rates for the correct model. When the data were generated from Model 1, the BIC selected Model 1(i.e., the correct model) 100% of the time. The BIC selected the correct model 100% of the time when data was generated from Model 2, Model 3, and Model 4. However, when the data were generated from Model 5, the BIC selected Model 5 only 1.2% of the time.

Table center4 reports the mean relative bias for coefficient estimates for the five estimated models across all five population models. Relative bias was computed as the difference between a parameter estimate and its true value, divided by the true value. The highest relative bias for a correct model was 1.04% for Model 2's β_2. Otherwise, relative bias for the true population model never exceeded 1%.

Overall, these results suggest that the DIC and BIC can effectively select an appropriate knot specification among competing models in most conditions. In general, the BIC selected the correct model more often than the DIC. A major exception to this occurred for data generated from Population Model 5. When data were generated from Model 5, the BIC selected Model 4 (an incorrect and less-complex model) 99% of the time but the DIC selected Model 5 (the correct model) 100% of the time. The PPP does not seem to reliably select the correct model when models with more phases are available. When the correct model is

[5] Ties were extremely rare and only occurred for the PPP. Ties for the winning model according to PPP occurred in 1.2% of Population Model 1 replications, 0.02% of replications in Population Model 3, and 0.02% of replications in Population Model 4. No other ties occurred.

selected, growth factor means are estimated with very little bias. While these results provide evidence that the Bayesian PGCM demonstrated in this tutorial is a useful tool for handling complex nonlinear trajectories, a more thorough simulation study is needed to examine whether this pattern of results holds across different research conditions.

Table 3. Selection rates for model fit indices.

Population	Estimated	PPP	DIC	BIC
Population Model 1	Model 1	**0.11**	**0.86**	**1.00**
	Model 2	0.12	0.06	0.00
	Model 3	0.13	0.07	0.00
	Model 4	0.27	0.01	0.00
	Model 5	0.36	0.00	0.00
Population Model 2	Model 1	0.00	0.00	0.00
	Model 2	**0.16**	**0.92**	**1.00**
	Model 3	0.00	0.00	0.00
	Model 4	0.24	0.06	0.00
	Model 5	0.61	0.02	0.00
Population Model 3	Model 1	0.00	0.00	0.00
	Model 2	0.00	0.00	0.00
	Model 3	**0.23**	**0.98**	**1.00**
	Model 4	0.00	0.00	0.00
	Model 5	0.77	0.02	0.00
Population Model 4	Model 1	0.00	0.00	0.00
	Model 2	0.00	0.00	0.00
	Model 3	0.00	0.00	0.00
	Model 4	**0.28**	**0.93**	**1.00**
	Model 5	0.72	0.07	0.00
Population Model 5	Model 1	0.00	0.00	0.00
	Model 2	0.00	0.00	0.00
	Model 3	0.00	0.00	0.00
	Model 4	0.00	0.00	0.99
	Model 5	**1.00**	**1.00**	**0.01**

Note. PPP = posterior predictive p-value; DIC = deviance information criterion; BIC = Bayesian information criterion.
Selection rates for the true model are bolded.

Table 4. Average relative bias (in %) for growth factor means across the proof of concept simulation.

Population	Estimated	β_0	β_1	β_2	β_3	β_4	β_5	β_6
Population Model 1	*Model 1*	*-0.19*	*-0.07*	*-0.21*	*-0.16*			
	Model 2	-0.19	-0.06	-0.19	-100.03			
	Model 3	-0.18	-0.07	-100.01	-34.80			
	Model 4	-0.20	-0.06	-99.96	-34.83			
	Model 5	-0.20	-0.06	-99.97	-99.96			
Population Model 2	Model 1	-0.07	-0.01	982.00	17.17			
	Model 2	*-0.04*	*-0.08*	*-1.04*	*0.27*	*0.08*		
	Model 3	-0.04	-0.13	-108.01	-46.79	67.24		
	Model 4	-0.06	-0.09	-100.74	-95.11	59.25		
	Model 5	-0.06	-0.09	-99.55	-100.04	-92.18		
Population Model 3	Model 1	-10.54	32.33	-240.20	-2.42			
	Model 2	-10.05	31.58	-237.04	-98.97	-0.10		
	Model 3	*-0.04*	*-0.13*	*-0.00*	*-0.22*	*-0.12*		
	Model 4	-5.04	24.10	-69.86	-0.01	-100.02		
	Model 5	-0.06	-0.12	-0.07	-100.00	2.68		
Population Model 4	Model 1	-0.43	1.80	-102.80	-32.32			
	Model 2	-0.18	0.91	-70.09	-32.40	-18.46		
	Model 3	0.02	-0.04	-89.94	-86.17	58.06		
	Model 4	*0.00*	*0.02*	*-0.23*	*-0.31*	*0.07*	*0.27*	
	Model 5	0.00	0.05	-100.06	-206.53	140.32	23.33	
Population Model 5	Model 1	-1.39	7.79	-6.60	-142.37			
	Model 2	-0.74	0.87	-102.51	-139.94	-69.01		
	Model 3	0.03	-9.35	-202.68	-145.14	-48.82		
	Model 4	0.77	-14.95	-365.47	-174.41	-61.66	-19.19	
	Model 5	*0.02*	*-0.10*	*-0.93*	*-0.38*	*-0.15*	*0.09*	*-0.60*

Note. β_0 = mean baseline math achievement; $\beta_1, ..., \beta_6$ = mean slope parameters. The 'correct' estimated models are italicized. All relative biases are reported to two decimals, such that 0.02 indicates relative bias is 0.02% less than 0.005%.

Journal of Behavioral Data Science, 2023, *3* (1), 34–58.
DOI: https://doi.org/10.35566/jbds/v3n1/ogasawara

On Some Known Derivations and New Ones for the Wishart Distribution: A Didactic

Haruhiko Ogasawara[0000−0002−5029−2086]

Otaru University of Commerce, Otaru, Japan
emt-hogasa@emt.otaru-uc.ac.jp

Abstract. The proofs of the probability density function (pdf) of the Wishart distribution tend to be complicated with geometric viewpoints, tedious Jacobians and not self-contained algebra. In this paper, some known proofs and simple new ones for uncorrelated and correlated cases are provided with didactic explanations. For the new derivation of the uncorrelated case, an elementary direct derivation of the distribution of the Bartlett-decomposed matrix is provided. In the derivation of the correlated case from the uncorrelated one, simple methods including a new one are shown.

Keywords: Jacobian · Multivariate normality · Probability density function (pdf) · Triangular matrix · Bartlett decomposition.

1 Introduction

The Wishart distribution has been often used for the matrix of the squares and cross products of random vectors. In multivariate analysis or more specifically structural equation modeling (SEM), a modified log-likelihood of this distribution (see e.g., Ogasawara, 2016, Equation (2.8)) has been used probably as a gold-standard discrepancy function for estimation even under non-normality though the distribution is given under multivariate normality. In SEM, variations of the distribution are also used as priors for covariance matrices (Liu, Qu, Zhang, & Wu, 2022; Zhang, 2021). The distribution has various extensions e.g., the inverted distribution (Anderson, 2003, Section 7.7), singular cases (Bodnar & Okhrin, 2008; Mathai & Provost, 2022; Srivastava, 2003), complex-valued ones (Srivastava & Khatri, 1979, Section 3.7; Mathai, Provost, & Haubold, 2022, Section 5.5), those with two different degrees of freedom (df's) (Ogasawara, 2023b), the joint distributions of the Wishart matrix and normal vectors (Yonenaga, 2022) and cases under arbitrary distributions (Hsu, 1940; Srivastava & Khatri, 1979, Lemma 3.2.3; Olkin, 2002, Section 2).

Asymptotic results associated with the Wishart distribution are also of practical use. In SEM, the asymptotic standard errors of the Wishart maximum likelihood estimators for structural parameters are often used under normality or non-normality. In this situation, the large df is assumed. When the number of variables is also large under some condition as in high-dimensional data (see e.g., Yao, Zheng, & Bai, 2015), the limiting distribution of the eigenvalues of the Wishart matrix is given by the Marčenko and Pastur (1967, M-P) distribution (the author is indebted to an anonymous reviewer for this point). The M-P distribution gives a tool for the problems of the numbers of factors or components in SEM (Chen & Weng, 2023).

The probability density functions (pdf's) of the Wishart distribution were given by Fisher (1915, p. 510) and Wishart (1928) for the bivariate and general multivariate cases, respectively. The derivations tend to be involved with geometric viewpoints (see e.g., Anderson, 2003, Section 7.2) or not self-contained algebra as criticized by Ghosh and Sinha (2002) (for the references of derivations see Srivastava & Khatri, 1979, p. 73 and Anderson, 2003, pp. 256-257). Khatri (1963) showed a brief derivation using an integral of the unity over the constant quadratic forms having the chi-square density. Ghosh and Sinha (2002) gave a self-contained concise proof of the Wishart density though it is an indirect method. In spite of frequent use of the Wishart density and its variations in SEM, the derivation of the pdf seems to be often intractable for beginning students/researchers. Probably, many of them use the Wishart pdf as if referencing a cook book without understanding the derivation, which is an undesirable situation. A relatively concise derivation is to use the characteristic function and its inversion (Wishart & Bartlett, 1933; Wilks, 1962, Section 18.2). However, this method requires the Fourier integral theorem or Levy's inversion formula, which may be unfamiliar for beginners. In this paper, almost self-contained known proofs and new ones for the uncorrelated and correlated multivariate cases are shown with didactic explanations.

2 Proofs of the Wishart Distributions

2.1 The distribution of a lower-triangular matrix for the Wishart density

Suppose that in the random matrix $\mathbf{X} = \{X_{ij}\}$ $(i = 1, ..., p; j = 1, ..., n; p \leq n)$, each column is multivariate normally distributed as $\mathrm{N}_p(\mathbf{0}, \mathbf{I}_p)$ independent of the other columns with the population mean vector $\mathbf{0}$ and covariance matrix $\mathbf{I}_p$ denoting the $p \times p$ identity matrix. That is, all the elements of $\mathbf{X}$ are mutually independently distributed as standard normal.

Let $\mathbf{S} \equiv \mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{T}\mathbf{T}^{\mathrm{T}}$ be Bartlett-decomposed such that $\mathbf{T}$ is a $p \times p$ lower-triangular matrix whose diagonal elements are positive. Define $\mathbf{s} = (s_{11}, s_{21}, s_{22}, ..., s_{p1}, ..., s_{pp})^{\mathrm{T}}$ and $\mathbf{t} = (t_{11}, t_{21}, t_{22}, ..., t_{p1}, ..., t_{pp})^{\mathrm{T}}$, where $\mathbf{s}$ and $\mathbf{t}$ are the $\{(p^2 + p)/2\} \times 1$ vectors of the non-duplicated elements of $\mathbf{S}$ and the random elements of $\mathbf{T}$, respectively. Let $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_+$ (Srivastava & Khatri, 1979, p. 28) be the absolute value of the determinant of the Jacobian matrix for the transformation $\mathbf{S} \to \mathbf{T}$:

$$\frac{\partial\mathbf{s}}{\partial\mathbf{t}^{\mathrm{T}}} = \left\{\frac{\partial s_{ij}}{\partial t_{kl}}\right\} (p \geq i \geq j \geq 1; p \geq k \geq l \geq 1)$$

using the double subscript notation for the rows of the elements of $\mathbf{S}$ and columns for those of $\mathbf{t}^{\mathrm{T}}$ in $\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}$. Then, the Jacobian of the transformation is given by $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+}$. For the proof of the Wishart distribution, the following lemmas are used.

Lemma 1 *Suppose that each of* $2m$ *variables* X_{ik} *and* X_{jk} $(i \neq j; k = 1, ..., m;$ $m = 1, 2, ...)$ *independently follows* $\mathrm{N}(0, 1) \equiv \mathrm{N}_1(0, 1)$. *Then, the distribution of* $\sum_{k=1}^{m} X_{ik}X_{jk}$ *is the same as that of* $X_{il}\sqrt{\sum_{k=1}^{m} X_{jk}^2}$ $(i \neq j;\ l = 1, ..., m)$.

Proof. When $m = 1$, the equal distribution of $X_{i1}X_{j1}$ and $X_{i1}\sqrt{X_{j1}^2} = X_{i1}|X_{j1}|$ is given by the symmetric distribution of $X_{i1}X_{j1}$ about zero. For general cases, consider the moment generating functions (mgf's). By definition, the mgf of $\sum_{k=1}^{m} X_{ik}X_{jk}$ is

$$\begin{aligned}
&\mathrm{E}\left\{\exp\left(t\textstyle\sum_{k=1}^{m} X_{ik}X_{jk}\right)\right\} = \textstyle\prod_{k=1}^{m} \mathrm{E}\left\{\exp(tX_{ik}X_{jk})\right\} \\
&= \textstyle\prod_{k=1}^{m} \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left(tx_{ik}x_{jk} - \frac{x_{jk}^2}{2}\right)\mathrm{d}x_{jk} \exp\left(-\frac{x_{ik}^2}{2}\right)\mathrm{d}x_{ik} \\
&= \textstyle\prod_{k=1}^{m} \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x_{jk}-tx_{ik})^2}{2}\right\}\mathrm{d}x_{jk} \exp\left\{-\frac{(1-t^2)x_{ik}^2}{2}\right\}\mathrm{d}x_{ik} \\
&= \textstyle\prod_{k=1}^{m} \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \exp\left\{-\frac{(1-t^2)x_{ik}^2}{2}\right\}\mathrm{d}x_{ik} \\
&= (1-t^2)^{-m/2}\ (|t|\ < 1).
\end{aligned}$$

On the other hand, the mgf of $X_{il}\sqrt{\sum_{k=1}^{m} X_{jk}^2}$ is

$$\begin{aligned}
&\mathrm{E}\exp\left(tX_{il}\sqrt{\textstyle\sum_{k=1}^{m} X_{jk}^2}\right) \\
&= \frac{1}{(2\pi)^{(m+1)/2}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \exp\left(tx_{il}\sqrt{\textstyle\sum_{k=1}^{m} x_{jk}^2} - \frac{x_{il}^2}{2} - \frac{\sum_{k=1}^{m} x_{jk}^2}{2}\right) \\
&\qquad \times \mathrm{d}x_{il}\mathrm{d}x_{j1}\cdots\mathrm{d}x_{jm} \\
&= \frac{1}{(2\pi)^{m/2}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \frac{1}{(2\pi)^{1/2}} \exp\left\{-\left(x_{il} - t\sqrt{\textstyle\sum_{k=1}^{m} x_{jk}^2}\right)^2/2\right\}\mathrm{d}x_{il} \\
&\qquad \times \exp\left\{-(1-t^2)\textstyle\sum_{k=1}^{m} x_{jk}^2/2\right\}\mathrm{d}x_{j1}\cdots\mathrm{d}x_{jm} \\
&= \frac{1}{(2\pi)^{m/2}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \exp\left\{-(1-t^2)\textstyle\sum_{k=1}^{m} x_{jk}^2/2\right\}\mathrm{d}x_{j1}\cdots\mathrm{d}x_{jm} \\
&= (1-t^2)^{-m/2}\ (|t|\ < 1).
\end{aligned}$$

It is found that the above two mgf's are the same, which shows the same distribution of $\sum_{k=1}^{m} X_{ik}X_{jk}$ and $X_{il}\sqrt{\sum_{k=1}^{m} X_{jk}^2}$ $(i \neq j;\ l = 1, ..., m)$. □

The second proof using the pdf of the chi-distribution is given in the supplement to this paper (Ogasawara, 2023a).

Lemma 2 (Deemer & Olkin, 1951, Theorem 4.1; Srivastava & Khatri, 1979, Exercise 1.28 (i); Muirhead, 1982, Theorem 2.1.9; Anderson, 2003, p. 255). *The Jacobian of the transformation* $\mathbf{S} \to \mathbf{T}$ *is*

$$|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+} = 2^{p}\prod\nolimits_{i=1}^{p} t_{ii}^{p-i+1}.$$

Proof. Deemer and Olkin (1951) derived the result as a special case of another general theorem. Muirhead (1982) used the exterior product while an essential standard proof was given by Anderson (2003). The derivation is given here by induction. When $p = 1$, $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+} = \mathrm{d}s_{11}/\mathrm{d}t_{11} = \mathrm{d}t_{11}^{2}/\mathrm{d}t_{11} = 2t_{11} > 0$ showing that the above result holds. Assume that the result holds when $p = p^{*}$ i.e., $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+} = 2^{p^{*}}\prod_{i=1}^{p^{*}} t_{ii}^{p^{*}-i+1}(p^{*} \geq 1)$. When $p = p^{*}+1$, the elements $s_{p*+1,1}, s_{p*+1,2}, ..., s_{p*+1,p*+1}$ are added to $\mathbf{s}$ at its end. Similarly, $t_{p*+1,1}, t_{p*+1,2}, ..., t_{p*+1,p*+1}$ are added to $\mathbf{t}^{\mathrm{T}}$. Noting that $s_{ij} = \sum\limits_{k=1}^{j} t_{ik}t_{jk}$ $(p \geq i \geq j \geq 1)$, we find that $\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}$ is a lower-triangular matrix. Consequently, the added factor in $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+}$ when $p = p^{*}+1$ over when $p = p^{*}$ is given by the product of the added diagonal elements:

$$\frac{\partial s_{p*+1,1}}{\partial t_{p*+1,1}}\frac{\partial s_{p*+1,2}}{\partial t_{p*+1,2}}\cdots\frac{\partial s_{p*+1,p*}}{\partial t_{p*+1,p*}}\frac{\partial s_{p*+1,p*+1}}{\partial t_{p*+1,p*+1}} = t_{11}t_{22}\cdots t_{p*p*}2t_{p*+1,p*+1}.$$

That is, $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+}$ becomes

$$2^{p^{*}}\left(\prod\nolimits_{i=1}^{p^{*}} t_{ii}^{p^{*}-i+1}\right)t_{11}t_{22}\cdots t_{p*p*}2t_{p*+1,p*+1} = 2^{p^{*}+1}\prod\nolimits_{i=1}^{p^{*}+1} t_{ii}^{p^{*}+1-i+1},$$

which shows that the formula $|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_{+} = 2^{p}\prod_{i=1}^{p} t_{ii}^{p-i+1}$ holds when $p = p^{*}+1$ indicating the required result. □

In the following theorem for a known Wishart density, we use $\Gamma_{p}(n/2) \equiv \pi^{p(p-1)/4}\prod_{i=1}^{p}\Gamma\{(n-i+1)/2\}$ i.e., the p-variate Gamma function (Anderson, 2003, Definition 7.2.1; Subsection 7.2, Equation (18); see also DLMF, 2021, Section 35.3, https://dlmf.nist.gov/35.3), where $\Gamma(k) = \int_{0}^{\infty} v^{k-1}\exp(-v)\mathrm{d}v$ $(k > 0)$ is the usual gamma function.

Theorem 1 *Under the condition that the n columns of* $\mathbf{X}$ *independently follow* $\mathrm{N}_{p}(\mathbf{0}, \mathbf{I}_{p})$*, the pdf of the Wishart distributed* $\mathbf{S}$ *is given by*

$$w_{p}(\mathbf{S}|\mathbf{I}_{p}, n) = \frac{\exp\{-\mathrm{tr}(\mathbf{S})/2\}|\mathbf{S}|^{(n-p-1)/2}}{2^{np/2}\Gamma_{p}(n/2)}\ (n \geq p).$$

Proof. Consider the case of $t_{ij} = X_{ij}$ and $t_{ii} = \sqrt{\sum_{k=i}^{n} X_{ik}^{2}}\,(i = 1, ..., p; j = 1, ..., i-1)$. Since $X_{ij}(i = 1, ..., p; j = 1, ..., n)$ are mutually independent, t_{ij} $(i = 1, ..., p; j = 1, ..., i)$ are independent. Note that $(\mathbf{TT}^{\mathrm{T}})_{ii} = \sum_{j=1}^{i} t_{ij}^{2} = (\mathbf{XX}^{\mathrm{T}})_{ii}$ $(i = 1, ..., p)$ are independently chi-square distributed with n df, where $(\cdot)_{ij}$ is the (i, j)-th element of a matrix; and t_{ii} is chi-distributed with $n - i + 1$

df. Further, Lemma 1 shows that the distributions of the off-diagonal elements $(\mathbf{TT}^{\mathrm{T}})_{ij} = \sum_{k=1}^{j} t_{ik}t_{jk}$ and $(\mathbf{XX}^{\mathrm{T}})_{ij} (p \geq i > j \geq 1)$ using t_{jj} and $t_{ij}\,(i = 1, ..., p; j = 1, ..., i-1)$ are the same. That is, the distribution of $\mathbf{S} = \mathbf{XX}^{\mathrm{T}}$ and $\mathbf{TT}^{\mathrm{T}}$ are the same when t_{ij} $(i = 1, ..., p; j = 1, ..., i)$ are distributed as above. The pdf of the constructed t_{ij}'s $(p \geq i \geq j \geq 1)$ denoted by $f_p(\mathbf{T})$ becomes

$$
\begin{aligned}
f_p(\mathbf{T}) &= \left[\prod_{i=1}^{p} \frac{t_{ii}^{n-i}\exp(-t_{ii}^2/2)}{2^{\{(n-i+1)/2\}-1}\mathit{\Gamma}\{(n-i+1)/2\}}\right] \\
&\quad \times \frac{1}{(\sqrt{2\pi})^{(p^2-p)/2}} \left\{\prod_{p\geq i>j\geq 1} \exp\left(-t_{ij}^2/2\right)\right\} \\
&= \frac{\left\{\prod_{i=1}^{p} t_{ii}^{n-i}\exp(-t_{ii}^2/2)\right\}\left\{\prod_{p\geq i>j\geq 1} \exp\left(-t_{ij}^2/2\right)\right\}}{2^{\frac{(n+1)p}{2}-\frac{p(p+1)}{4}-p} \times 2^{\frac{p(p-1)}{4}}\pi^{\frac{p(p-1)}{4}} \prod_{i=1}^{p} \mathit{\Gamma}\{(n-i+1)/2\}} \\
&= \frac{\left(\prod_{i=1}^{p} t_{ii}^{n-i}\right)\exp\{-\mathrm{tr}(\mathbf{TT}^{\mathrm{T}})/2\}}{2^{\frac{np}{2}-p}\mathit{\Gamma}_p(n/2)}.
\end{aligned}
$$

In the above expression, the pdf of the chi-distributed t_{ii} with k df denoted by $f_\chi(t_{ii}|k)$ is given by that of the chi-square distributed $u = t_{ii}^2$ with k df i.e., $f_{\chi^2}(u|k) = \dfrac{u^{(k/2)-1}}{2^{k/2}\mathit{\Gamma}(k/2)}\exp(-u/2)$ with the Jacobian $\mathrm{d}u/\mathrm{d}t_{ii} = 2t_{ii}$, yielding

$$
f_\chi(t_{ii}|k) = \frac{u^{(k/2)-1}}{2^{k/2}\mathit{\Gamma}(k/2)}\exp(-u/2)\frac{\mathrm{d}u}{\mathrm{d}t_{ii}} = \frac{t_{ii}^{(n-i+1)-2+1}\exp(-t_{ii}^2/2)}{2^{(n-i+1)/2-1}\mathit{\Gamma}\{(n-i+1)/2\}}
$$

as shown earlier, when $u = t_{ii}^2$ and $k = n - i + 1$.

Consider the transformation $\mathbf{T} \to \mathbf{S}$ in $\mathbf{S} = \mathbf{XX}^{\mathrm{T}} = \mathbf{TT}^{\mathrm{T}}$. The Jacobian $J(\mathbf{T} \to \mathbf{S})$ of this transformation is given by the reciprocal of $J(\mathbf{S} \to \mathbf{T})$ obtained in Lemma 2 as $J(\mathbf{T} \to \mathbf{S}) = 1/|\partial\mathbf{s}/\partial\mathbf{t}^{\mathrm{T}}|_+ = \left(2^p \prod_{i=1}^{p} t_{ii}^{p-i+1}\right)^{-1}$. Consequently, using $|\mathbf{S}|^{1/2} = |\mathbf{T}| = t_{11}\cdots t_{pp}$ the pdf of $\mathbf{S}$ becomes

$$
\begin{aligned}
&w_p(\mathbf{S}|\mathbf{I}_p, n) = f_p(\mathbf{T})J(\mathbf{T} \to \mathbf{S}) \\
&= \frac{\left(\prod_{i=1}^{p} t_{ii}^{n-i}\right)\exp\{-\mathrm{tr}(\mathbf{TT}^{\mathrm{T}})/2\}}{2^{\frac{np}{2}-p}\mathit{\Gamma}_p(n/2)2^p\prod_{i=1}^{p} t_{ii}^{p-i+1}} = \frac{\exp\{-\mathrm{tr}(\mathbf{S})/2\}|\mathbf{S}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)}.
\end{aligned}
$$

□

Remark 1 The pdf of t_{ij}'s $(p \geq i \geq j \geq 1)$ i.e., $f_p(\mathbf{T})$ given above using Lemma 1 is algebraically equal to those of Anderson (2003, Equation (6), p. 253, Corollary 7.2.1), Wijsman (1957, Equation (12)) and Kshirsagar (1959, Remarks). However, a typical derivation by e.g., Anderson is an indirect one using orthogonalization and the conditional normal density. Since Anderson's derivation seems to give some complicated impressions for beginning students/researchers though it is almost self-contained, the corresponding didactic explanation of his derivation is given below. Anderson (2003, Equation (2), p. 252) defined the n-dimensional independent random vectors $\mathbf{v}_i \sim \mathrm{N}_n(\mathbf{0}, \mathbf{I}_n)$ $(i = 1, ..., p)$ with

$$\mathbf{X} = \begin{pmatrix} \mathbf{v}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{v}_p^{\mathrm{T}} \end{pmatrix}.$$

Then, the Gram-Schmidt sequential orthogonalization is employed (Anderson, 2003, Equation (3), p. 253) as

$$\mathbf{w}_i = \mathbf{v}_i - \sum_{j=1}^{i-1} \mathbf{w}_j \frac{\mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i}{\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j}\ (i = 2, ..., p) \text{ and } \mathbf{w}_1 = \mathbf{v}_1,$$

where he used the expression $\mathbf{v}_j^{\mathrm{T}}\mathbf{w}_j$ for the denominator $\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j$. Though $\mathbf{v}_j^{\mathrm{T}}\mathbf{w}_j = \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j$ $(j = 1, ..., i)$ as will become apparent, $\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j$ may be more natural and appropriate. While he included the short derivation of the orthogonality among $\mathbf{w}_i$'s by induction, it is repeated here with some added explanations. When $i = 2$, we have

$$\mathbf{w}_2^{\mathrm{T}}\mathbf{w}_1 = \{\mathbf{v}_2 - \mathbf{w}_1(\mathbf{w}_1^{\mathrm{T}}\mathbf{w}_1)^{-1}\mathbf{w}_1^{\mathrm{T}}\mathbf{v}_2\}^{\mathrm{T}}\mathbf{w}_1 = \mathbf{v}_2{}^{\mathrm{T}}\mathbf{w}_1 - \mathbf{v}_2{}^{\mathrm{T}}\mathbf{w}_1(\mathbf{w}_1^{\mathrm{T}}\mathbf{w}_1)^{-1}\mathbf{w}_1^{\mathrm{T}}\mathbf{w}_1 = 0$$

showing the orthogonality. Suppose that

$$\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_k\ = 0\ (j, k = 1, ..., i-1;\ j \neq k)$$

hold. Then, we have

$$\begin{aligned} \mathbf{w}_k^{\mathrm{T}}\mathbf{w}_i &= \mathbf{w}_k^{\mathrm{T}}\left(\mathbf{v}_i - \sum_{j=1}^{i-1} \mathbf{w}_j \frac{\mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i}{\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j}\right) = \mathbf{w}_k^{\mathrm{T}}\mathbf{v}_i - \sum_{j=1}^{i-1} \mathbf{w}_k^{\mathrm{T}}\mathbf{w}_j \frac{\mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i}{\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j} \\ &= \mathbf{w}_k^{\mathrm{T}}\mathbf{v}_i - \mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k \frac{\mathbf{w}_k^{\mathrm{T}}\mathbf{v}_i}{\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k} = 0\ (i = 2, ..., p;\ k = 1, ..., i-1), \end{aligned}$$

due to the assumption $\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_k\ = 0$ $(j, k = 1, ..., i-1;\ j \neq k)$, showing the required result $\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_k\ = 0$ $(j, k = 1, ..., i;\ j \neq k)$. Recall that $\mathbf{v}_j^{\mathrm{T}}\mathbf{w}_j = \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j$ $(j = 1, ..., i)$ mentioned earlier, which is obtained by $\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_k\ = 0$ $(j, k = 1, ..., i;\ j \neq k)$ and $\mathbf{w}_i = \mathbf{v}_i - \sum_{j=1}^{i-1} \mathbf{w}_j \frac{\mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i}{\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j}$ $(i = 2, ..., p)$.

The orthogonalization procedure is re-expressed by

$$\begin{aligned} \mathbf{w}_i &= \mathbf{v}_i - \sum_{j=1}^{i-1} \mathbf{w}_j \frac{\mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i}{\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j} \\ &= \mathbf{v}_i - (\mathbf{w}_1, ..., \mathbf{w}_{i-1})\mathrm{diag}\{(\mathbf{w}_1^{\mathrm{T}}\mathbf{w}_1)^{-1}, ..., (\mathbf{w}_{i-1}^{\mathrm{T}}\mathbf{w}_{i-1})^{-1}\}(\mathbf{w}_1, ..., \mathbf{w}_{i-1})^{\mathrm{T}}\mathbf{v}_i \\ &\equiv \mathbf{v}_i - \mathbf{P}_{\mathbf{W}_{i-1}}\mathbf{v}_i = (\mathbf{I}_n - \mathbf{P}_{\mathbf{W}_{i-1}})\mathbf{v}_i \equiv \mathbf{Q}_{\mathbf{W}_{i-1}}\mathbf{v}_i\ (i = 2, ..., p), \end{aligned}$$

where $\mathbf{P}_{\mathbf{W}_{i-1}} \equiv \mathbf{W}_{i-1}(\mathbf{W}_{i-1}^{\mathrm{T}}\mathbf{W}_{i-1})^{-1}\mathbf{W}_{i-1}^{\mathrm{T}}$ is the idempotent (i.e., $\mathbf{P}_{\mathbf{W}_{i-1}}^2 = \mathbf{P}_{\mathbf{W}_{i-1}}$) and symmetric projection matrix transforming or projecting $\mathbf{v}_i$ onto the space spanned by the columns of $\mathbf{W}_{i-1} \equiv (\mathbf{w}_1, ..., \mathbf{w}_{i-1})$ of full column rank by assumption; and $\mathbf{Q}_{\mathbf{W}_{i-1}} = \mathbf{I}_n - \mathbf{P}_{\mathbf{W}_{i-1}}$ is also an idempotent and symmetric projection matrix yielding the residual vector $\mathbf{v}_i - \mathbf{P}_{\mathbf{W}_{i-1}}\mathbf{v}_i$ or the projected vector on the space orthogonal to the column space of $\mathbf{W}_{i-1}$ with $\mathbf{v}_i = \mathbf{P}_{\mathbf{W}_{i-1}}\mathbf{v}_i + \mathbf{Q}_{\mathbf{W}_{i-1}}\mathbf{v}_i$. Anderson (2003, p. 252) stated that "$\mathbf{w}_i$ is the vector from $\mathbf{v}_i$ to the projection on $\mathbf{w}_1, ..., \mathbf{w}_{i-1}$" with his Figure 7.1. He repeatedly stressed the equivalence of the column space of $\mathbf{W}_{i-1}$ and that of $\mathbf{v}_1, ..., \mathbf{v}_{i-1}$ in our expression.

Using the constructed $\mathbf{w}_1, ..., \mathbf{w}_{i-1}$ by the Gram-Schmidt orthogonalization or projection, Anderson (2003, p. 252) defined

$$t_{ii} = ||\mathbf{w}_i|| = \sqrt{\mathbf{w}_i^{\mathrm{T}}\mathbf{w}_i} \ (i = 1, ..., p)$$

and

$$t_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j / \, ||\mathbf{w}_j|| \ \ (i = 2, ..., p; \, j = 1, ..., i-1),$$

which may be uniformly expressed by $t_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j / \, ||\mathbf{w}_j|| = \ (i = 2, ..., p; \, j = 1, ..., i)$ due to $\mathbf{v}_j^{\mathrm{T}}\mathbf{w}_j = \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j \ (j = 1, ..., i)$ mentioned earlier. Then, noting that $\mathbf{w}_i = \mathbf{v}_i - \mathbf{P}_{\mathbf{W}_{i-1}}\mathbf{v}_i$, we have

$$\mathbf{v}_i = \mathbf{w}_i + \mathbf{P}_{\mathbf{W}_{i-1}}\mathbf{v}_i = \mathbf{w}_i + \sum_{j=1}^{i-1}\frac{\mathbf{w}_j\mathbf{w}_j^{\mathrm{T}}}{\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j}\mathbf{v}_i = \sum_{j=1}^{i}\frac{\mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i}{||\mathbf{w}_j||^2}\mathbf{w}_j = \sum_{j=1}^{i}\frac{t_{ij}}{||\mathbf{w}_j||}\mathbf{w}_j$$

and

$$\begin{aligned}(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{v}_j &= \left\{\sum_{k=1}^{i}\frac{t_{ik}}{||\mathbf{w}_k||}\mathbf{w}_k^{\mathrm{T}}\right\}\sum_{k=1}^{j}\frac{t_{jk}}{||\mathbf{w}_k||}\mathbf{w}_k \\ &= \sum_{k=1}^{j}\frac{t_{ik}}{||\mathbf{w}_k||}\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\frac{t_{jk}}{||\mathbf{w}_k||} = \sum_{k=1}^{j}t_{ik}t_{jk} \ (p \geq i \geq j \geq 1)\end{aligned}$$

(Anderson, 2003, p. 252). In $\mathbf{v}_i = \sum_{j=1}^{i}\frac{t_{ij}}{||\mathbf{w}_j||}\mathbf{w}_j \ (i = 1, ..., p)$, $\mathbf{w}_j / \, ||\mathbf{w}_j|| \ (j = 1, ..., i-1)$ is seen as the unit-norm vector representing the direction for the j-th coordinate in the $i-1$ coordinates given by $\mathbf{w}_1, ..., \mathbf{w}_{i-1}$. He stated that "$t_{ij}, j = 1, ..., i-1$ are the first $i-1$ coordinates in the coordinate system with $\mathbf{w}_1, ..., \mathbf{w}_{i-1}$ as the first coordinates axes" (p. 252). We also find that t_{ij} is $||\mathbf{w}_j||$ times the regression coefficient b_{ij} for $\mathbf{v}_i$ on $\mathbf{w}_j$ since

$$\begin{aligned}t_{ij} &= \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j / \, ||\mathbf{w}_j|| = (\mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j \, / \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j \,) \, ||\mathbf{w}_j|| \\ &= b_{ij}||\mathbf{w}_j||(i = 2, ..., p; \, j = 1, ..., i-1).\end{aligned}$$

The properties of the normality of $t_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j / \, ||\mathbf{w}_j|| \ (i = 2, ..., p; \, j = 1, ..., i-1)$ and their mutual independence shown by Anderson are based on the normality of the conditional distribution of the multivariate normal when $\mathbf{w}_j(j =$

$1, ..., i-1$) are given and orthogonal transformation in $t_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j / ||\mathbf{w}_j||$ ($i = 2, ..., p; j = 1, ..., i-1$). That is, the standard normally-distributed variables $t_{ij} = \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j / ||\mathbf{w}_j||$ do not depend on $\mathbf{w}_1, ..., \mathbf{w}_{i-1}$ indicating independence with $(\mathbf{w}_j / ||\mathbf{w}_j||)^{\mathrm{T}}\mathbf{w}_k / ||\mathbf{w}_k|| = \delta_{jk}$ ($j, k = 1, ..., i-1$), where δ_{jk} is the Kronecker delta with $\delta_{jj} = 1$ and $\delta_{jk} = 0$ ($j \neq k$) (Anderson, 2003, Theorem 3.3.1). The independent property of t_{ii}'s is given by $t_{ii} = \left\{(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{ii} - \sum_{j=1}^{i-1} t_{ij}^2\right\}^{1/2}$. Although the same result as shown above by the didactic explanation of Anderson's derivation is directly given by Lemma 1, the two methods may be insightful with compensatory properties. [end of Remark 1]

2.2 The Wishart density for general correlated cases

For the correlated cases, four lemmas are provided. Lemma 3 is for three Jacobians in the product of two lower-triangular matrices, where the first Jacobian was used by Anderson (2003, Theorem 7.2.2) to derive the Wishart density for general correlated cases while the remaining two are given for generality with didactic purposes. Lemmas 4 and 5 are provided for the Jacobians in two alternative derivations of the general Wishart density. The proof of Lemma 6 associated with sufficient statistics is based on Ghosh and Sinha (2002).

Lemma 3 *Suppose that* $\mathbf{A} = \mathbf{BC}$*, where* $\mathbf{A}$*,* $\mathbf{B}$ *and* $\mathbf{C}$ *are* $p \times p$ *lower-triangular matrices. Consider the variable transformation from the non-zero elements of* $\mathbf{C}$ *or* $\mathbf{B}$ *to those of* $\mathbf{A}$*. Then, the Jacobians* $J(\mathbf{C} \to \mathbf{A})$ *and* $J(\mathbf{B} \to \mathbf{A})$ *are* $\left|\prod_{i=1}^{p} b_{ii}^{i}\right|^{-1}$ *and* $\left|\prod_{i=1}^{p} c_{ii}^{p-i+1}\right|^{-1}$*, respectively. When* $\mathbf{B} = \mathbf{C}$*,* $J(\mathbf{B} \to \mathbf{A}) = \left|\prod_{i=1}^{p} \prod_{j=1}^{i} (b_{ii} + b_{jj})\right|^{-1}$*.*

Proof. Note that Anderson (2003, p. 254) gave $J(\mathbf{C} \to \mathbf{A})$. Since $a_{ij} = \sum_{k=j}^{i} b_{ik}c_{kj}$ ($p \geq i \geq j \geq 1$), we have

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{22} \\ \vdots \\ a_{p1} \\ \vdots \\ a_{pm} \end{bmatrix} = \begin{bmatrix} b_{11} & 0 & 0 & \cdots & 0 & \cdots & 0 \\ * & b_{22} & 0 & \cdots & 0 & \cdots & 0 \\ * & * & b_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ * & * & * & \cdots & b_{pp} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ * & * & * & \cdots & * & \cdots & b_{pp} \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{21} \\ c_{22} \\ \vdots \\ c_{p1} \\ \vdots \\ c_{pp} \end{bmatrix},$$

where the diagonal element of the lower-triangular matrix corresponding to the row for a_{ij} and the column for c_{ij} is b_{ii} ($p \geq i \geq j \geq 1$); the asterisks indicate zero or non-zero elements; and

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{22} \\ \vdots \\ a_{p1} \\ \vdots \\ a_{pp} \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 & \cdots & 0 & \cdots & 0 \\ * & c_{11} & 0 & \cdots & 0 & \cdots & 0 \\ * & * & c_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ * & * & * & \cdots & c_{11} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ * & * & * & \cdots & * & \cdots & c_{pp} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \\ b_{22} \\ \vdots \\ b_{p1} \\ \vdots \\ b_{pp} \end{bmatrix},$$

where the corresponding diagonal element for a_{ij} and b_{ij} is c_{jj} $(p \geq i \geq j \geq 1)$. Since the inverses of the Jacobian matrices for $J(\mathbf{C} \to \mathbf{A})$ and $J(\mathbf{B} \to \mathbf{A})$ on the right-hand sides of the above equations are lower-triangular, the Jacobians become the reciprocals of the absolute values of the determinants i.e., $\prod_{i=1}^{p} b_{ii}^{i}$ and $\prod_{i=1}^{p} c_{ii}^{p-i+1}$, respectively. The result when $\mathbf{B} = \mathbf{C}$ is obtained by the reciprocal of the determinant of the sum of the two lower-triangular matrices. □

Lemma 4 *Suppose that* $\mathbf{A} = \mathbf{B}\mathbf{C}\mathbf{B}^{\mathrm{T}}$*, where* $\mathbf{A}$ *and* $\mathbf{C}$ *are* $p \times p$ *symmetric matrices; and* $\mathbf{B}$ *is a lower-triangular matrix. Consider the variable transformation from the non-duplicated elements of* $\mathbf{C}$ *to those of* $\mathbf{A}$*. Then, the Jacobian* $J(\mathbf{C} \to \mathbf{A})$ *is* $|\mathbf{B}|_{+}^{-(p+1)}$.

Proof. Since the non-duplicated elements of $\mathbf{A}$ using its diagonal and infra-diagonal elements are $a_{ij} = \sum_{k=1}^{i} \sum_{l=1}^{j} b_{ik} c_{kl} b_{jl}$ $(p \geq i \geq j \geq 1)$, we have

$$\frac{\partial a_{ij}}{\partial c_{kl}} = b_{ik} b_{jl} \ (p \geq i \geq j \geq 1;\ k = 1, ..., i;\ l = 1, ..., j),$$

which gives

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{22} \\ \vdots \\ a_{p1} \\ \vdots \\ a_{pp} \end{bmatrix} = \begin{bmatrix} b_{11}b_{11} & 0 & 0 & \cdots & 0 & \cdots & 0 \\ * & b_{22}b_{11} & 0 & \cdots & 0 & \cdots & 0 \\ * & * & b_{22}b_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ * & * & * & \cdots & b_{pp}b_{11} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ * & * & * & \cdots & * & \cdots & b_{pp}b_{pp} \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{21} \\ c_{22} \\ \vdots \\ c_{p1} \\ \vdots \\ c_{pp} \end{bmatrix},$$

where the diagonal element of the lower-triangular matrix for a_{ij} and c_{ij} is $\partial a_{ij} / \partial c_{ij} = b_{ii} b_{jj}$ $(p \geq i \geq j \geq 1)$. Since $J(\mathbf{C} \to \mathbf{A})$ is the reciprocal of the absolute value of the determinant of the above lower-triangular matrix, we obtain $J(\mathbf{C} \to \mathbf{A}) = 1 / \left| \prod_{i=1}^{p} b_{ii}^{p+1} \right| = |\mathbf{B}|_{+}^{-(p+1)}$. □

Lemma 5 *Suppose that* $\mathbf{A} = \mathbf{B}\mathbf{C}\mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$*, where* $\mathbf{A}$ *is a* $p \times p$ *symmetric matrix; and* $\mathbf{B}$ *and* $\mathbf{C}$ *are lower-triangular matrices. Consider the variable transformation from the non-zero elements of* $\mathbf{C}$ *to the non-duplicated elements of* $\mathbf{A}$*. Then, the Jacobian* $J(\mathbf{C} \to \mathbf{A})$ *is* $|\mathbf{B}|_{+}^{-(p+1)} / \left| 2^{p} \prod_{i=1}^{p} c_{ii}^{p-i+1} \right|$.

Proof 1 The diagonal and infra-diagonal elements of $\mathbf{A}$ are employed for its non-duplicated ones without loss of generality. Then, define $\mathbf{a} =$ $(a_{11}, a_{21}, a_{22}, ..., a_{p1}, ..., a_{pp})^{\mathrm{T}}$ and $\mathbf{c} = (c_{11}, c_{21}, c_{22}, ..., c_{p1}, ..., c_{pp})^{\mathrm{T}}$ with the elements lexicographically ordered. Since $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{BC}$ are lower-triangular, the Jacobian matrix $\partial\mathbf{a}/\partial\mathbf{c}^{\mathrm{T}} = \{\partial a_{ij}/\partial c_{kl}\}$ $(p \geq i \geq j \geq 1; p \geq k \geq l \geq 1)$ becomes lower-triangular. This can be shown by

$$\begin{aligned}\tfrac{\partial a_{ij}}{\partial c_{kl}} &= \{\mathbf{B}(\mathbf{E}_{kl}\mathbf{C}^{\mathrm{T}} + \mathbf{C}\mathbf{E}_{lk})\mathbf{B}^{\mathrm{T}}\}_{ij} = (\mathbf{B}\mathbf{E}_{kl}\mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}})_{ij} + (\mathbf{B}\mathbf{C}\mathbf{E}_{lk}\mathbf{B}^{\mathrm{T}})_{ij} \\ &= b_{ik}(\mathbf{BC})_{jl} + (\mathbf{BC})_{il}b_{jk} \; (p \geq i \geq j \geq 1;\; p \geq k \geq l \geq 1),\end{aligned}$$

where $\mathbf{E}_{ij}$ is the matrix of an appropriate size, whose (i, j)th element is 1 with the remaining ones being 0. The right-hand side of the last equation in the above expression vanishes when $i < k$ or $\{i = k\} \cap \{j < l\}$. This condition indicates the lower-triangular form of $\partial\mathbf{a}/\partial\mathbf{c}^{\mathrm{T}} = \{\partial a_{ij}/\partial c_{kl}\}$. Then, the diagonal elements are

$$\frac{\partial a_{ij}}{\partial c_{ij}} = \{\mathbf{B}(\mathbf{E}_{ij}\mathbf{C}^{\mathrm{T}} + \mathbf{C}\mathbf{E}_{ji})\mathbf{B}^{\mathrm{T}}\}_{ij} = (\mathbf{B}\mathbf{E}_{ij}\mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}})_{ij} = b_{ii}c_{jj}b_{jj} \; (p \geq i > j \geq 1)$$

and

$$\frac{\partial a_{ii}}{\partial c_{ii}} = \{\mathbf{B}(\mathbf{E}_{ii}\mathbf{C}^{\mathrm{T}} + \mathbf{C}\mathbf{E}_{ii})\mathbf{B}^{\mathrm{T}}\}_{ii} = 2b_{ii}^2 c_{ii} \; (i = 1, ..., p).$$

Since the determinant of the Jacobian matrix for $J(\mathbf{A} \to \mathbf{C})$ is

$$\begin{aligned}&\textstyle\prod_{i=1}^{p}\prod_{j=1}^{i}\frac{\partial a_{ij}}{\partial c_{ij}} = \left(\prod_{i=1}^{p}\prod_{j=1}^{i-1}\frac{\partial a_{ij}}{\partial c_{ij}}\right)\prod_{i=1}^{p}\frac{\partial a_{ii}}{\partial c_{ii}} = 2^p\prod_{i=1}^{p}\prod_{j=1}^{i} b_{ii}c_{jj}b_{jj} \\ &= 2^p\left(\textstyle\prod_{i=1}^{p} b_{ii}^{i}\right)\prod_{j=1}^{p} c_{jj}^{p-j+1}b_{jj}^{p-j+1} = 2^p\prod_{i=1}^{p} b_{ii}^{p+1}c_{ii}^{p-i+1} \\ &= 2^p|\mathbf{B}|^{p+1}\textstyle\prod_{i=1}^{p} c_{ii}^{p-i+1},\end{aligned}$$

the Jacobian $J(\mathbf{C} \to \mathbf{A})$ is the reciprocal of the absolute value of the above quantity:

$$J(\mathbf{C} \to \mathbf{A}) = |\mathbf{B}|_{+}^{-(p+1)} / \left| 2^p \prod\nolimits_{i=1}^{p} c_{ii}^{p-i+1} \right|,$$

which is the required result. □

Proof 2 The transformation $\mathbf{A} = \mathbf{B}\mathbf{C}\mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$ is seen in two steps. In the first step, the transformation $\mathbf{C} \to \mathbf{C}\mathbf{C}^{\mathrm{T}}$ is considered, whose Jacobian is given by Lemma 2 as $J(\mathbf{C} \to \mathbf{C}\mathbf{C}^{\mathrm{T}}) = 1/\left|2^p\prod_{i=1}^{p} c_{ii}^{p-i+1}\right|$. The second step is for the transformation $\mathbf{C}\mathbf{C}^{\mathrm{T}} \to \mathbf{A} = \mathbf{B}\mathbf{C}\mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$ with the Jacobian $J(\mathbf{C}\mathbf{C}^{\mathrm{T}} \to \mathbf{A}) = |\mathbf{B}|_{+}^{-(p+1)}$, which is given by Lemma 4. Then, the Jacobian $J(\mathbf{C} \to \mathbf{A})$ is the product of the two Jacobians due to the chain rule, which gives the required result. □

Suppose that each column of a $p \times n$ matrix $\mathbf{Y}$ follows $\mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$ with positive definite $\mathbf{\Sigma}$ independent of the other columns. Recall $\mathbf{X}$ in Theorem 1. Let $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}^{\mathrm{T}}$ be the Cholesky decomposition, where $\mathbf{B}$ is a fixed lower-triangular matrix whose diagonal elements are positive for identification and convenience. Then, each column of $\mathbf{Y} = \mathbf{BX}$ independently follows $\mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$. Define $\mathbf{S}_{\mathbf{\Sigma}} \equiv \mathbf{YY}^{\mathrm{T}} = \mathbf{BXX}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} = \mathbf{BSB}^{\mathrm{T}}$, where $\mathbf{S} = \mathbf{S}_{\mathbf{I}_p} = \mathbf{XX}^{\mathrm{T}} = \mathbf{TT}^{\mathrm{T}}$, and the $\{p(p+1)/2\} \times 1$ vector $\mathbf{s}_{\mathbf{\Sigma}} \equiv (s_{\mathbf{\Sigma}11},\ s_{\mathbf{\Sigma}21}, s_{\mathbf{\Sigma}22}, ..., s_{\mathbf{\Sigma}p1}, ..., s_{\mathbf{\Sigma}pp})^{\mathrm{T}}$ with $\mathbf{S}_{\mathbf{\Sigma}} = \{s_{\mathbf{\Sigma}ij}\}$ $(i, j = 1, ..., p)$.

Lemma 6 *Define positive definite* $\mathbf{\Sigma}_i = \mathbf{B}_i\mathbf{B}_i^{\mathrm{T}}$ *and* $\mathbf{S}_{\mathbf{\Sigma}_i} = \mathbf{B}_i\mathbf{SB}_i^{\mathrm{T}}$ $(i = 1, 2)$, *where* $\mathbf{S}$ *is as before. Denote the pdf's of* $\mathbf{S}_{\mathbf{\Sigma}_i}$ *at* $\mathbf{S}_{\mathbf{\Sigma}}$ *by* $g_{\mathbf{\Sigma}=\mathbf{\Sigma}_i}(\mathbf{S}_{\mathbf{\Sigma}})$ $(i = 1, 2)$. *Then,*

$$\frac{g_{\mathbf{\Sigma}=\mathbf{\Sigma}_1}(\mathbf{S}_{\mathbf{\Sigma}})}{g_{\mathbf{\Sigma}=\mathbf{\Sigma}_2}(\mathbf{S}_{\mathbf{\Sigma}})} = \frac{\phi_{p,n}(\mathbf{Y}|\mathbf{0},\ \mathbf{\Sigma}_1)}{\phi_{p,n}(\mathbf{Y}|\mathbf{0},\ \mathbf{\Sigma}_2)},$$

where $\phi_{p,n}(\mathbf{Y}|\mathbf{0},\ \mathbf{\Sigma}_i) = \prod_{j=1}^{n} \phi_p\{(\mathbf{Y})_{\cdot j}|\mathbf{0},\ \mathbf{\Sigma}_i\}$; $(\mathbf{Y})_{\cdot j}$ *is the* j*-th column of* $\mathbf{Y}$; *and*

$$\phi_p\{(\mathbf{Y})_{\cdot j}|\mathbf{0},\ \mathbf{\Sigma}_i\} = \frac{\exp\{-(\mathbf{Y})_{\cdot j}^{\mathrm{T}}\mathbf{\Sigma}_i^{-1}(\mathbf{Y})_{\cdot j}/2\}}{(2\pi)^{n/2}|\mathbf{\Sigma}_i|^{1/2}}\ \ (i = 1, 2;\ j = 1, ..., n).$$

Proof. The derivation is given by the factorization theorem for the sufficient statistic corresponding to $\mathbf{S}_{\mathbf{\Sigma}}$ for $\mathbf{\Sigma}$ as used by Ghosh and Sinha (2002, Equation (8)):

$$\phi_{p,n}(\mathbf{Y}|\mathbf{0},\ \mathbf{\Sigma}_i) = g_{\mathbf{\Sigma}=\mathbf{\Sigma}_i}(\mathbf{S}_{\mathbf{\Sigma}})h(\mathbf{Y})\ (i = 1, 2),$$

which gives the required result. □

The Wishart density for general correlated cases (see e.g., Srivastava & Khatri, 1979, Theorem 3.2.1; Anderson, 2003, Theorem 7.2.2) is derived in different ways.

Theorem 2 *Let each column of a* $p \times n$ *matrix* $\mathbf{Y}$ *follows* $\mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$ *with positive definite* $\mathbf{\Sigma}$ *independent of the other columns. Then, the pdf of* $\mathbf{S}_{\mathbf{\Sigma}} = \mathbf{YY}^{\mathrm{T}}$ *is*

$$w_p(\mathbf{S}_{\mathbf{\Sigma}}|\mathbf{\Sigma}, n) = \frac{\exp\{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_{\mathbf{\Sigma}})/2\}|\mathbf{S}_{\mathbf{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\mathbf{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)}.$$

Proof 1 Consider the transformation $\mathbf{T} \to \mathbf{S}_{\mathbf{\Sigma}} = \mathbf{BTT}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$. The Jacobian is given by Lemma 5, when $\mathbf{A} = \mathbf{S}_{\mathbf{\Sigma}},\ \mathbf{B} = \mathbf{B}$ and $\mathbf{C} = \mathbf{T}$ with added restrictions $b_{ii} > 0$ and $t_{ii} > 0$ $(i = 1, ..., p)$ as

$$J(\mathbf{T} \to \mathbf{S}_{\mathbf{\Sigma}}) = |\mathbf{B}|^{-(p+1)} / \left(2^p \prod_{i=1}^{p} t_{ii}^{p-i+1}\right) = |\mathbf{\Sigma}|^{-(p+1)/2} / \left(2^p \prod_{i=1}^{p} t_{ii}^{p-i+1}\right).$$

The pdf of $\mathbf{T}$ denoted by $f_p(\mathbf{T})$ was given by Theorem 1. Then, we have

$$\begin{aligned}
&w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma}, n) = f_p(\mathbf{T})J(\mathbf{T} \to \mathbf{S}_{\boldsymbol{\Sigma}}) \\
&= \frac{\exp\{-\mathrm{tr}(\mathbf{T}\mathbf{T}^{\mathrm{T}})/2\} \prod_{i=1}^{p} t_{ii}^{n-i}}{2^{(np/2)-p}\mathit{\Gamma}_p(n/2)} \frac{|\boldsymbol{\Sigma}|^{-(p+1)/2}}{2^p \prod_{i=1}^{p} t_{ii}^{p-i+1}} \\
&= \frac{\exp\{-\mathrm{tr}(\mathbf{T}\mathbf{T}^{\mathrm{T}})/2\}|\boldsymbol{\Sigma}|^{-(p+1)/2} \prod_{i=1}^{p} t_{ii}^{n-p-1}}{2^{np/2}\mathit{\Gamma}_p(n/2)} \\
&= \frac{\exp\{-\mathrm{tr}(\mathbf{B}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}\mathbf{B}^{\mathrm{T}-1})/2\}|\boldsymbol{\Sigma}|^{-(p+1)/2}|\mathbf{B}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}\mathbf{B}^{\mathrm{T}-1}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)} \\
&= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)},
\end{aligned}$$

where $\mathrm{tr}(\mathbf{B}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}\mathbf{B}^{\mathrm{T}-1}) = \mathrm{tr}(\mathbf{B}^{\mathrm{T}-1}\mathbf{B}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}) = \mathrm{tr}\{(\mathbf{B}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}\} = \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})$ and $|\mathbf{B}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}\mathbf{B}^{\mathrm{T}-1}| = |\mathbf{S}_{\boldsymbol{\Sigma}}||\boldsymbol{\Sigma}|^{-1}$ are used. The last expression gives the required result. □

Proof 2 Employ the two-step transformation $\mathbf{T} \to \mathbf{S} = \mathbf{T}\mathbf{T}^{\mathrm{T}} \to \mathbf{S}_{\boldsymbol{\Sigma}} = \mathbf{B}\mathbf{S}\mathbf{B}^{\mathrm{T}}$. The first step was used by Theorem 1. The Jacobian $J(\mathbf{T} \to \mathbf{S} = \mathbf{T}\mathbf{T}^{\mathrm{T}})$ in the first step is given by Lemma 2 by taking the reciprocal of the last result of the lemma while $J(\mathbf{S} \to \mathbf{S}_{\boldsymbol{\Sigma}} = \mathbf{B}\mathbf{S}\mathbf{B}^{\mathrm{T}})$ is obtained by Lemma 4. That is,

$$\begin{aligned}
&w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma}, n) = f_p(\mathbf{T})J(\mathbf{T} \to \mathbf{S})J(\mathbf{S} \to \mathbf{S}_{\boldsymbol{\Sigma}}) \\
&= \frac{\exp\{-\mathrm{tr}(\mathbf{S})/2\}|\mathbf{S}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)} J(\mathbf{S} \to \mathbf{S}_{\boldsymbol{\Sigma}}) \\
&= \frac{\exp\{-\mathrm{tr}(\mathbf{S})/2\}|\mathbf{S}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)} |\mathbf{B}|^{-(p+1)} \\
&= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}|\boldsymbol{\Sigma}|^{-(p+1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)} \\
&= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)}.
\end{aligned}$$

□

Proof 3 (Anderson, 2003, Theorem 7.2.2) Anderson used an alternative two-step transformation $\mathbf{T} \to \mathbf{T}^* = \mathbf{B}\mathbf{T} \to \mathbf{S}_{\boldsymbol{\Sigma}} = \mathbf{T}^*\mathbf{T}^{*T}$. The Jacobian $J(\mathbf{T} \to \mathbf{T}^*)$ is given by the first result of Lemma 3 while $J(\mathbf{T}^* \to \mathbf{S}_{\boldsymbol{\Sigma}})$ is given by the reciprocal of the last result in Lemma 2 when $\mathbf{T} = \mathbf{T}^*$. That is,

$$\begin{aligned} w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma}, n) &= f_p(\mathbf{T})J(\mathbf{T} \to \mathbf{T}^*)J(\mathbf{T}^* \to \mathbf{S}_{\boldsymbol{\Sigma}}) \\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{T}\mathbf{T}^{\mathrm{T}})/2\} \prod\limits_{i=1}^{p} t_{ii}^{n-i}}{2^{(np/2)-p}\mathit{\Gamma}_p(n/2)} \left(\textstyle\prod_{i=1}^{p} b_{ii}^{i}\right)^{-1} \left(2^p \textstyle\prod_{i=1}^{p} t_{ii}^{*p-i+1}\right)^{-1} \\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{T}\mathbf{T}^{\mathrm{T}})/2\} \prod\limits_{i=1}^{p} (t_{ii}^{*}/b_{ii})^{n-i}}{2^{np/2}\mathit{\Gamma}_p(n/2)\left(\prod_{i=1}^{p} b_{ii}^{i}\right)\prod_{i=1}^{p} t_{ii}^{*p-i+1}} = \frac{\exp\{-\mathrm{tr}(\mathbf{T}\mathbf{T}^{\mathrm{T}})/2\} \prod\limits_{i=1}^{p} t_{ii}^{*n-p-1}}{2^{np/2}\left(\prod_{i=1}^{p} b_{ii}^{n}\right)\mathit{\Gamma}_p(n/2)} \\ &= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)}. \end{aligned}$$

□

Proof 4 Use Theorem 1 and Lemma 6 when $\boldsymbol{\Sigma}_1 = \mathbf{I}_p$ and $\boldsymbol{\Sigma}_2 = \mathbf{B}_2\mathbf{B}_2^{\mathrm{T}} = \boldsymbol{\Sigma}^{1/2}(\boldsymbol{\Sigma}^{1/2})^{\mathrm{T}} = \boldsymbol{\Sigma}$. Then, we have

$$\begin{aligned} w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma}, n) &= w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\mathbf{I}_p, n)\frac{\phi_{p,n}(\mathbf{Y}|\mathbf{0},\ \boldsymbol{\Sigma})}{\phi_{p,n}(\mathbf{Y}|\mathbf{0},\ \mathbf{I}_p)} \\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)} \frac{\exp\{-\mathrm{tr}(\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1})/2\}/\{(2\pi)^{pn/2}|\boldsymbol{\Sigma}|^{n/2}\}}{\exp\{-\mathrm{tr}(\mathbf{Y}\mathbf{Y}^{\mathrm{T}})/2\}/(2\pi)^{pn/2}} \\ &= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)}. \end{aligned}$$

□

3 Remarks and Conclusion

For the general correlated cases, four proofs are shown in Theorem 2. The one-step first proof uses $f_p(\mathbf{T})$ with $J(\mathbf{T} \to \mathbf{S}_{\boldsymbol{\Sigma}})$ given by Lemma 5, where $\mathbf{S}_{\boldsymbol{\Sigma}} = \mathbf{B}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$ with lower-triangular $\mathbf{B}$ and $\mathbf{T}$ is seen as a two-fold Bartlett (Cholesky) decomposition or a usual Bartlett (1933) $\mathbf{S}_{\boldsymbol{\Sigma}} = \mathbf{B}\mathbf{T}(\mathbf{B}\mathbf{T})^{\mathrm{T}}$ in terms of lower-triangular $\mathbf{B}\mathbf{T}$. The two-step second proof uses $f_p(\mathbf{T})$ with $J(\mathbf{T} \to \mathbf{S} = \mathbf{T}\mathbf{T}^{\mathrm{T}})$ and $J(\mathbf{S} \to \mathbf{S}_{\boldsymbol{\Sigma}} = \mathbf{B}\mathbf{S}\mathbf{B}^{\mathrm{T}})$ obtained by Lemmas 2 and 4, respectively. Anderson (2003)'s two-step third proof uses $f_p(\mathbf{T})$ with $J(\mathbf{T} \to \mathbf{T}^* = \mathbf{B}\mathbf{T})$ and $J(\mathbf{T}^* \to \mathbf{S}_{\boldsymbol{\Sigma}})$ given by Lemmas 3 and 2, respectively. Among the four proofs, the first and fourth ones are relatively simple. The remaining two-step proofs seem to be comparable. It is found that in order to derive the final Jacobian by Proofs 2 and 3, Lemma 2 is firstly and secondly used, respectively. When only the pdf of $\mathbf{S}(= \mathbf{S}_{\boldsymbol{\Sigma}=\mathbf{I}_p})$ is focused on, Proof 2 may be the simplest though the same result is immediately obtained from the pdf of $\mathbf{S}_{\boldsymbol{\Sigma}}$ substituting $\boldsymbol{\Sigma} = \mathbf{I}_p$.

In each of the four proofs, $f_p(\mathbf{T})$ is used. Two derivations for $f_p(\mathbf{T})$ were shown. The first method using Lemma 1 is much simpler than that used by Anderson (2003) as detailed in Remark 1. The author believes that this simplification will reduce the difficulties frequently encountered when beginning students/researchers master the Wishart density. Note that when the Wishart density for $w_p(\mathbf{S}|\mathbf{I}_p, n)$ is given, $f_p(\mathbf{T})$ is obtained using $J(\mathbf{S} \to \mathbf{T})$ in Lemma 2 as easily as the transformation $J(\mathbf{T} \to \mathbf{S})$, which is the reversed problem (see Bartlett, 1933; Muirhead, 1982, Theorem 3.2.14).

Remark 2 Lemma 1 gave the justification of $\mathbf{XX}^{\mathrm{T}} = \mathbf{TT}^{\mathrm{T}}$ with mutually independent normal t_{ij} $(p \geq i > j \geq)$ and chi-distributed $t_{ii} (i = 1, ..., p)$. While the chi-square distribution of $(\mathbf{TT}^{\mathrm{T}})_{ii}$ is obvious, the distribution of $(\mathbf{TT}^{\mathrm{T}})_{ij}$ $(i > j)$ is that of the product sum of p pairs of independent normals (the product-sum normal for short). The pdf and mgf of the product-sum normal in the case of a possibly correlated single pair was given by Craig (1936) (see also Ogasawara, 2023a, Remarks S1-S4). For current developments of this issue, see e.g., Seijas-Macías and Oliveira (2012), Seijas-Macías, Oliveira, Oliveira, and Leiva (2020), and Gaunt (2022).

Remark 3 As addressed earlier, the complicated property found in many of the proofs of the Wishart density seems to be due partially to the associated Jacobians in e.g., Srivastava and Khatri (1979, Section 3.2) and Anderson (2003, Section 7.2). The proof of the Wishart density in Theorem 1 is similar to that in Srivastava and Khatri (1979, Section 3.2).Though the Jacobian in Lemma 2 was also used by Srivastava and Khatri, we did not use the Jacobian of $\mathbf{X} \to \{\mathbf{T},\ \mathbf{V}^*\}$ in $\mathbf{X} = \mathbf{TV}^*$, where $\mathbf{V}^*$ is a $p \times n$ semi-orthonormal matrix with $\mathbf{V}^*\mathbf{V}^{*\mathrm{T}} = \mathbf{I}_p$ (see Srivastava & Khatri, 1979, Exercise 1.33). Instead, we used the marginal chi and normal distributions for $\mathbf{T}$ as in Anderson (2003).

As shown earlier, in the three proofs of the Wishart density $w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma}, n)$, the Bartlett-like Cholesky decomposition $\boldsymbol{\Sigma} = \mathbf{BB}^{\mathrm{T}}$ is used for non-stochastic $\boldsymbol{\Sigma}$. Though this factorization gives simple results, other factorizations can also be used with $\boldsymbol{\Sigma} = \mathbf{BGG}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} = \mathbf{BG}(\mathbf{BG})^{\mathrm{T}} = \mathbf{DD}^{\mathrm{T}}$, where $\mathbf{GG}^{\mathrm{T}} = \mathbf{G}^{\mathrm{T}}\mathbf{G} = \mathbf{I}_p$ and $\mathbf{D} = \mathbf{BG}$. For illustration, Proof 5 using $\mathbf{D} = \boldsymbol{\Sigma}^{1/2}$ with $(\boldsymbol{\Sigma}^{1/2})^2 = \boldsymbol{\Sigma}$ will be shown in Appendix A for didactic purposes with associated remarks. The concise derivation of Khatri (1963) will be explained in Appendix B. The Bartlett decomposition $\mathbf{S} = \mathbf{TT}^{\mathrm{T}}$ can also be replaced by other ones with the same number of random variables. The case called the exchanged Bartlett decomposition will be shown in Appendix C.

Conclusion Among Proofs 1 to 4 of the Wishart distribution given earlier and Proofs 5 to 7 to be shown in the appendix for expository purposes, Proof 4 using our Lemma 1 for the equivalence of the distributions of the product-sum normal and the product of the chi and standard normal as well as Lemma 6 for the factorization theorem given by Ghosh and Sinha (2002) is the simplest. Since Proof 4 uses elementary and self-contained methods, the proof may be understood by beginning students/researchers without much difficulty.

References

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis* (3rd ed.). New York: Wiley.

Bartlett, M. S. (1933). On the theory of statistical regression. *Proceedings of the Royal Society of Edinburgh*, *53*, 260–283. doi: https://doi.org/10.1017/s0370164600015637

Bodnar, T., & Okhrin, Y. (2008). Properties of the singular, inverse and generalized inverse partitioned Wishart distributions. *Journal of Multivariate Analysis*, *99*(10), 2389–2405. doi: https://doi.org/10.1016/j.jmva.2008.02.024

Chen, Y.-L., & Weng, L.-J. (2023). On Horn's approximation to the sampling distribution of eigenvalues from random correlation matrices in parallel analysis. *Current Psychology*(online published). doi: https://doi.org/10.1007/s12144-023-04635-9

Craig, C. C. (1936). On the frequency function of xy. *The Annals of Mathematical Statistics*, *7*(1), 1–15. doi: https://doi.org/10.1214/aoms/1177732541

Deemer, W. L., & Olkin, I. (1951). The jacobians of certain matrix transformations useful in multivariate analysis: Based on lectures of P. L. Hsu at the University of North Carolina, 1947. *Biometrika*, *38*(3/4), 345–367. doi: https://doi.org/10.2307/2332581

DLMF. (2021). *NIST Digital Library of Mathematical Functions. National Institutes of Standards and Technology, U. S. Department of Commerce. Release 1.1.3 of 2021-09-15.* Retrieved from http://dlmf.nist.gov/ (F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, M. A. McClain (Eds))

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, *10*(4), 507–521. doi: https://doi.org/10.2307/2331838

Gaunt, R. E. (2022). The basic distributional theory for the product of zero mean correlated normal random variables. *Statistica Neerlandica*, *76*(4), 450–470. doi: https://doi.org/10.1111/stan.12267

Ghosh, M., & Sinha, B. K. (2002). A simple derivation of the Wishart distribution. *The American Statistician*, *56*(2), 100–101. doi: https://doi.org/10.1198/000313002317572754

Hsu, P. L. (1940). An algebraic derivation of the distribution of rectangular coordinates. *Proceedings of the Edinburgh Mathematical Society*, *6*(3), 185–189. doi: https://doi.org/10.1007/978-1-4684-9324-5_14

Khatri, C. G. (1963). (no title). *Journal of the Indian Statistical Association*, *1*(Queries section), 52–54.

Kshirsagar, A. M. (1959). Bartlett decomposition and Wishart distribution. *The Annals of Mathematical Statistics*, *30*(1), 239–241. doi: https://doi.org/10.1214/aoms/1177706379

Kshirsagar, A. M. (1972). *Multivariate analysis.* New York: Marcel Dekker. doi: https://doi.org/10.2307/1267507

Liu, H., Qu, W., Zhang, Z., & Wu, H. (2022). A new Bayesian structural equation modeling approach with priors on the covariance matrix parameter. *Journal of Behavioral Data Science*, *2*(2), 23–46. doi:

https://doi.org/10.35566/jbds/v2n2/p2

Magnus, J. R., & Neudecker, H. (1986). Symmetry, 0-1 matrices and Jacobians: A review. *Econometric Theory*, *2*(2), 157–190. doi: https://doi.org/10.1017/s0266466600011476

Magnus, J. R., & Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics* (Rev. ed.). New York: Wiley. doi: https://doi.org/10.1002/9781119541219

Marčenko, V. A., & Pastur, L. A. (1967). Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, *1*(4), 457-–483. doi: https://doi.org/10.1070/sm1967v001n04abeh001994

Mathai, A. M., & Provost, S. B. (2022). On the singular gamma, Wishart, and beta matrix-variate density functions. *Canadian Journal of Statistics*, *50*(4), 1143–1165. doi: https://doi.org/10.1002/cjs.11710

Mathai, A. M., Provost, S. B., & Haubold, H. J. (2022). *Multivariate statistical analysis in the real and complex domains.* Cham, Switzerland: Springer Nature. doi: https://doi.org/10.1007/978-3-030-95864-0

Muirhead, R. J. (1982). *Aspects of multivariate statistical theory.* New York: Wiley. doi: https://doi.org/10.2307/2288369

Ogasawara, H. (2007). Asymptotic expansions of the distributions of estimators in canonical correlation analysis under nonnormality. *Journal of Multivariate Analysis*, *98*(9), 1726–1750. doi: https://doi.org/10.1016/j.jmva.2006.12.001

Ogasawara, H. (2016). Bias correction of the Akaike information criterion in factor analysis. *Journal of Multivariate Analysis*, *149*, 144–159. doi: https://doi.org/10.1016/j.jmva.2016.04.003

Ogasawara, H. (2022). *A stochastic derivation of the surface area of the (n-1)-sphere.* Preprint at ResearchGate. doi: https://doi.org/10.13140/RG.2.2.28827.95528

Ogasawara, H. (2023a). Supplement to the paper "on some known derivations and new ones for the Wishart distribution: A didactic". To appear in *Economic Review (Otaru University of Commerce)*, *74*(2 & 3, https://www.otaru-uc.ac.jp/˜emt-hogasa/, https://barrel.repo.nii.ac.jp/).

Ogasawara, H. (2023b). The Wishart distribution with two different degrees of freedom. *Statistics and Probability Letters*, *200*(109866, online published). doi: https://doi.org/10.1016/j.spl.2023.109866

Olkin, I. (2002). The 70th anniversary of the distribution of random matrices: A survey. *Linear Algebra and Its Applications*, *354*, 231–243. doi: https://doi.org/10.1016/s0024-3795(01)00314-7

Schuberth, F. (2021). The Henseler-Ogasawara specification of composites in structural equation modeling: A tutorial. *Psychological methods*(online published). doi: https://doi.org/10.1037/met0000432

Seijas-Macías, A., & Oliveira, A. (2012). An approach to distribution of the product of two normal variables. *Discussiones Mathematicae Probability and Statistics*, *32*(1-2), 87–99. doi: https://doi.org/10.7151/dmps.1146

Seijas-Macías, A., Oliveira, A., Oliveira, T. A., & Leiva, V. (2020). Approximating the distribution of the product of two normally distributed random variables. *Symmetry*, *12*(8), 1201. doi: https://doi.org/10.3390/sym12081201

Srivastava, M. S. (2003). Singular Wishart and multivariate beta distributions. *The Annals of Statistics*, *31*(5), 1537–1560. doi: https://doi.org/10.1214/aos/1065705118

Srivastava, M. S., & Khatri, C. G. (1979). *An introduction to multivariate statistics, 350.* Amsterdam: Elsevier.

Wijsman, R. A. (1957). Random orthogonal transformations and their use in some classical distribution problems in multivariate analysis. *The Annals of Mathematical Statistics*, *28*(2), 415–423. doi: https://doi.org/10.1214/aoms/1177706969

Wilks, S. S. (1962). *Mathematical statistics.* New York: Wiley. doi: https://doi.org/10.2307/2311277

Wishart, J. (1928). The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, *20A*, 32–52. doi: https://doi.org/10.2307/2331939

Wishart, J., & Bartlett, M. S. (1933). The generalised product moment distribution in a normal system. *Mathematical Proceedings of the Cambridge Philosophical Society*, *29*(2), 260–270. doi: https://doi.org/10.1017/s0305004100011063

Yao, J., Zheng, S., & Bai, Z. D. (2015). *Sample covariance matrices and high-dimensional data analysis.* New York: Cambridge University Press.

Yonenaga, K. (2022). *Functionals of a Wishart matrix and a normal vector and its application to linear discriminant analysis.* Doctoral dissertation, Graduate School of Economics and Business, Hokkaido University, Japan.

Yu, X., Schuberth, F., & Henseler, J. (2023). Specifying composites in structural equation modeling: A refinement of the Henseler–Ogasawara specification. *Statistical Analysis and Data Mining: The ASA Data Science Journal*(online published). doi: https://doi.org/10.1002/sam.11608

Zhang, Z. (2021). A note on Wishart and inverse Wishart priors for covariance matrix. *Journal of Behavioral Data Science*, *1*(2), 119–126. doi: https://doi.org/10.35566/jbds/v1n2/p2

Appendix A An alternative proof of the Wishart density for correlated cases

Let $\mathbf{\Sigma}^{1/2}$ be a symmetric matrix-square-root of $\mathbf{\Sigma}$ satisfying $(\mathbf{\Sigma}^{1/2})^2 = \mathbf{\Sigma}$. Then, we have $(\mathbf{Y})_{\cdot j} = (\mathbf{\Sigma}^{1/2}\mathbf{X})_{\cdot j} \sim \mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$ as $(\mathbf{BX})_{\cdot j} \sim \mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$ $(j = 1, ..., n)$, which gives $\mathbf{S}_{\mathbf{\Sigma}} \equiv \mathbf{YY}^{\mathrm{T}} = \mathbf{\Sigma}^{1/2}\mathbf{XX}^{\mathrm{T}}\mathbf{\Sigma}^{1/2} = \mathbf{\Sigma}^{1/2}\mathbf{S}\mathbf{\Sigma}^{1/2}$, where $\mathbf{S}_{\mathbf{\Sigma}}$ is redefined using $\mathbf{\Sigma}^{1/2}$. Let $\mathbf{s}_{\mathbf{\Sigma}} = (s_{\mathbf{\Sigma}11}, s_{\mathbf{\Sigma}21}, s_{\mathbf{\Sigma}22}, ..., s_{\mathbf{\Sigma}p1}, ..., s_{\mathbf{\Sigma}pp})^{\mathrm{T}}$ using redefined $\mathbf{S}_{\mathbf{\Sigma}} = \{s_{\mathbf{\Sigma}ij}\}$ $(i, j = 1, ..., p)$. Then, $\mathbf{D}_p\mathbf{s}_{\mathbf{\Sigma}} = \mathrm{vec}(\mathbf{S}_{\mathbf{\Sigma}})$ follows, where $\mathbf{D}_p$ of full column rank is the $p^2 \times \{p(p+1)/2\}$ duplication matrix consisting of 0's and 1's

(Magnus & Neudecker, 1999, Chapter 3, Section 8); and vec$(\cdot)$ is the vectorizing operator stacking the columns of a matrix in parentheses sequentially with the first column on the top. Using the formula $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})\mathrm{vec}(\mathbf{B})$ (see Magnus & Neudecker, 1999, Chapter 2, Theorem 2), where $\otimes$ denotes the direct or Kronecker product, we obtain

$$\mathbf{D}_p\mathbf{s}_{\boldsymbol{\Sigma}} = \mathrm{vec}(\mathbf{S}_{\boldsymbol{\Sigma}}) = \mathrm{vec}(\boldsymbol{\Sigma}^{1/2}\mathbf{S}\boldsymbol{\Sigma}^{1/2}) = (\boldsymbol{\Sigma}^{1/2}\otimes\boldsymbol{\Sigma}^{1/2})\mathrm{vec}(\mathbf{S}) = (\boldsymbol{\Sigma}^{1/2}\otimes\boldsymbol{\Sigma}^{1/2})\mathbf{D}_p\mathbf{s}.$$

Pre-multiplying the above equation by $(\mathbf{D}_p^{\mathrm{T}}\mathbf{D}_p)^{-1}\mathbf{D}_p^{\mathrm{T}} \equiv \mathbf{D}_p^{+}$, which is the left- or Moore-Penrose generalized inverse of $\mathbf{D}_p$ with $\mathbf{D}_p^{+}\mathbf{D}_p = \mathbf{I}_{p(p+1)/2}$ (see Magnus & Neudecker, 1999, Chapter 3, Section 8), we have

$$\mathbf{s}_{\boldsymbol{\Sigma}} = \mathbf{D}_p^{+}(\boldsymbol{\Sigma}^{1/2} \otimes \boldsymbol{\Sigma}^{1/2})\mathbf{D}_p\mathbf{s}.$$

The Jacobian of the transformation $\mathbf{S}_{\boldsymbol{\Sigma}} \to \mathbf{S}$ or equivalently $\mathbf{s}_{\boldsymbol{\Sigma}} \to \mathbf{s}$ is given by $|\mathbf{D}_p^{+}(\boldsymbol{\Sigma}^{1/2} \otimes \boldsymbol{\Sigma}^{1/2})\mathbf{D}_p|_{+} = |\boldsymbol{\Sigma}|^{(p+1)/2}$, which is derived using the following lemma.

Lemma 7 (Magnus & Neudecker, 1986, Equation (7.11)). *Let* $\mathbf{A}$ *be a* $p \times p$ *positive definite matrix with distinct eigenvalues. Then,* $|\mathbf{D}_p^{+}(\mathbf{A} \otimes \mathbf{A})\mathbf{D}_p| = |\mathbf{A}|^{p+1}$.

Proof. While Magnus and Neudecker (1986) used Shur's theorem for the existence of a non-singular matrix $\mathbf{V}$ satisfying $\mathbf{V}^{-1}\mathbf{AV} = \mathbf{M}$, where $\mathbf{M}$ is an upper-triangular matrix for a general square matrix $\mathbf{A}$, we use a familiar special case of the theorem as $\mathbf{L}^{\mathrm{T}}\mathbf{AL} = \boldsymbol{\Lambda}$ when $\mathbf{A} = \mathbf{L}\boldsymbol{\Lambda}\mathbf{L}^{\mathrm{T}}$ with $\mathbf{LL}^{\mathrm{T}} = \mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{I}_p$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_p)$ $(\lambda_1 > ... > \lambda_p > 0)$, where the columns of $\mathbf{L}$ and $\lambda_i (i = 1, ..., p)$ are the eigenvectors and eigenvalues of $\mathbf{A}$, respectively. Note that

$$\begin{aligned}
&\mathbf{D}_p^{+}(\mathbf{L}^{\mathrm{T}} \otimes \mathbf{L}^{\mathrm{T}})\mathbf{D}_p\mathbf{D}_p^{+}(\mathbf{A} \otimes \mathbf{A})\mathbf{D}_p\mathbf{D}_p^{+}(\mathbf{L} \otimes \mathbf{L})\mathbf{D}_p \\
&= \mathbf{D}_p^{+}(\mathbf{L}^{\mathrm{T}} \otimes \mathbf{L}^{\mathrm{T}})(\mathbf{A} \otimes \mathbf{A})(\mathbf{L} \otimes \mathbf{L})\mathbf{D}_p \\
&= \mathbf{D}_p^{+}\{(\mathbf{L}^{\mathrm{T}}\mathbf{AL}) \otimes (\mathbf{L}^{\mathrm{T}}\mathbf{AL})\}\mathbf{D}_p = \mathbf{D}_p^{+}(\boldsymbol{\Lambda} \otimes \boldsymbol{\Lambda})\mathbf{D}_p,
\end{aligned}$$

where $\mathbf{D}_p\mathbf{D}_p^{+}(\mathbf{A} \otimes \mathbf{A}) = (\mathbf{A} \otimes \mathbf{A})\mathbf{D}_p\mathbf{D}_p^{+}$ and $\mathbf{D}_p\mathbf{D}_p^{+}\mathbf{D}_p = \mathbf{D}_p$ (Magnus & Neudecker, 1999, Chapter 3, Theorem 13) are used, followed by the transformation given by $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$ when multiplications are defined.

Note that $\mathbf{D}_p^{+}(\mathbf{L}^{\mathrm{T}} \otimes \mathbf{L}^{\mathrm{T}})\mathbf{D}_p = \{\mathbf{D}_p^{+}(\mathbf{L} \otimes \mathbf{L})\mathbf{D}_p\}^{-1}$ since

$$\mathbf{D}_p^{+}(\mathbf{L}^{\mathrm{T}}\otimes\mathbf{L}^{\mathrm{T}})\mathbf{D}_p\mathbf{D}_p^{+}(\mathbf{L}\otimes\mathbf{L})\mathbf{D}_p = \mathbf{D}_p^{+}(\mathbf{L}^{\mathrm{T}}\otimes\mathbf{L}^{\mathrm{T}})(\mathbf{L}\otimes\mathbf{L})\mathbf{D}_p = \mathbf{D}_p^{+}\mathbf{D}_p = \mathbf{I}_{p(p+1)/2}.$$

Consequently, we can write as

$$\begin{aligned}
&\mathbf{D}_p^{+}(\mathbf{L}^{\mathrm{T}} \otimes \mathbf{L}^{\mathrm{T}})\mathbf{D}_p\mathbf{D}_p^{+}(\mathbf{A} \otimes \mathbf{A})\mathbf{D}_p\mathbf{D}_p^{+}(\mathbf{L} \otimes \mathbf{L})\mathbf{D}_p \\
&\equiv \mathbf{B}^{-1}\mathbf{D}_p^{+}(\mathbf{A} \otimes \mathbf{A})\mathbf{D}_p\mathbf{B} = \mathbf{D}_p^{+}(\boldsymbol{\Lambda} \otimes \boldsymbol{\Lambda})\mathbf{D}_p,
\end{aligned}$$

which shows that the eigenvalues of $\mathbf{D}_p^+(\mathbf{A}\otimes\mathbf{A})\mathbf{D}_p$ are the same as those of $\mathbf{D}_p^+(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}$ (see e.g., Magnus & Neudecker, 1999, Chapter 1, Theorem 5). Employ the double subscript notation as used earlier for the row numbers i and j ($p \geq i \geq j \geq 1$) and column numbers k and l ($p \geq k \geq l \geq 1$) of the $\{p(p+1)/2\} \times \{p(p+1)/2\}$ matrix $\mathbf{D}_p^+(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p$. These numbers correspond to the subscripts of the elements of e.g., the $\{p(p+1)/2\} \times 1$ vector $\mathbf{s} = (s_{11}, s_{21}, s_{22}, ..., s_{p1}, ..., , s_{pp})^{\mathrm{T}}$.

Consider $(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p$, where the (k,k)th columns of $(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p$ $(k = 1,\ldots,p)$ are unchanged from the corresponding ones of $\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda}$ while the (k,l)th columns $(p \geq k > l \geq 1)$ of $(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p$ are combined ones as the sum of the (k,l)- and (l,k)-th columns of $\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda}$ such that e.g.,

$$(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_2 = \mathrm{diag}(\lambda_1^2,\ \lambda_1\,\lambda_2,\ \lambda_2\,\lambda_1, \lambda_2^2)\begin{pmatrix}1&0&0\\0&1&0\\0&1&0\\0&0&1\end{pmatrix} = \begin{pmatrix}\lambda_1^2&0&0\\0&\lambda_1\,\lambda_2&0\\0&\lambda_2\,\lambda_1&0\\0&0&\lambda_2^2\end{pmatrix}$$

when $p = 2$. For the second transformation $\mathbf{D}_p^+(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p$, noting that $\mathbf{D}_p^+ = (\mathbf{D}_p^{\mathrm{T}}\mathbf{D}_p)^{-1}\mathbf{D}_p^{\mathrm{T}}$ consists of 1's, 1/2's and 0's as $\mathbf{D}_2^+ = \begin{pmatrix}1&0&0&0\\0&1/2&1/2&0\\0&0&0&1\end{pmatrix}$, we find that $\mathbf{D}_p^+(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p$ is the $\{p(p+1)/2\} \times \{p(p+1)/2\}$ diagonal matrix whose diagonal elements are λ_i^2 $(i = 1,...,p)$ and $\lambda_i\lambda_j$ $(p \geq i > j \geq 1)$ as $\mathbf{D}_2^+(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_2 = \mathrm{diag}(\lambda_1^2,\ \lambda_2\,\lambda_1, \lambda_2^2)$. Then, we have

$$\begin{aligned}|\mathbf{D}_p^+(\mathbf{A}\otimes\mathbf{A})\mathbf{D}_p| &= |\mathbf{D}_p^+(\boldsymbol{\Lambda}\otimes\boldsymbol{\Lambda})\mathbf{D}_p|\\ = \left(\prod_{i=1}^{p}\lambda_i^2\right)\prod_{p\geq i>j\geq 1}\lambda_i\lambda_j = \left(\prod_{i=1}^{p}\lambda_i\right)^{p+1} &= |\mathbf{A}|^{p+1}.\end{aligned}$$

□

Proof 5 of the Wishart density in Theorem 2 The Jacobian of the transformation $\mathbf{S}_{\boldsymbol{\Sigma}} \to \mathbf{S}$ or equivalently $\mathbf{s}_{\boldsymbol{\Sigma}} \to \mathbf{s}$ is given by Lemma 7 as $|\mathbf{D}_p^+(\boldsymbol{\Sigma}^{1/2}\otimes\boldsymbol{\Sigma}^{1/2})\mathbf{D}_p|_+ = |\boldsymbol{\Sigma}|^{(p+1)/2}$. Consequently, $J(\mathbf{s} \to \mathbf{s}_{\boldsymbol{\Sigma}})$ becomes $|\boldsymbol{\Sigma}|^{-(p+1)/2}$. Then, the pdf of $\mathbf{S}_{\boldsymbol{\Sigma}}$ is obtained by that of $\mathbf{S} = \boldsymbol{\Sigma}^{-1/2}\mathbf{S}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$ in Theorem 1 and $J(\mathbf{s} \to \mathbf{s}_{\boldsymbol{\Sigma}}) = |\boldsymbol{\Sigma}|^{-(p+1)/2}$ as

$$\begin{aligned}w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma}, n) &= \frac{\exp\{-\mathrm{tr}(\mathbf{S})/2\}|\mathbf{S}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)}|\boldsymbol{\Sigma}|^{-(p+1)/2}\\ &= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1/2}\mathbf{S}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2})/2\}|\boldsymbol{\Sigma}^{-1/2}\mathbf{S}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}|^{(n-p-1)/2}|\boldsymbol{\Sigma}|^{-(p+1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)}\\ &= \frac{\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\boldsymbol{\Sigma}})/2\}|\mathbf{S}_{\boldsymbol{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)}.\end{aligned}$$

□

Remark 4 When Lemma 7 for the Jacobian of $\mathbf{S}_{\boldsymbol{\Sigma}} \to \mathbf{S}$ is given, Theorem 2 for the Wishart density for general correlated cases was immediately obtained. Conversely, when the Wishart densities for $\mathbf{S}$ and $\mathbf{S}_{\boldsymbol{\Sigma}}$ are available, the Jacobian is easily given by comparing two densities using $\mathbf{S} = \boldsymbol{\Sigma}^{-1/2}\mathbf{S}_{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$, which was employed by Anderson (2003, Theorem 7.3.3).

Appendix B On Khatri (1963)'s self-contained concise derivation

Khatri (1963) is referred to only by Kshirsagar (1972, p. 59) and, Srivastava and Khatri (1979, p. 76) to the author's knowledge. The derivation depends on the integral $\pi^{k/2}q^{(k/2)-1}/\mathit{\Gamma}(k/2) = \int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k$, where q is a positive constant and x_i's with $\mathbf{x} = (x_1, ..., x_k)^{\mathrm{T}}$ independently follow the standard normal. This integral is typically obtained in a proof of the chi-square distribution with k df using the surface area $S_{k-1} = 2\pi^{k/2}r^{k-1}/\mathit{\Gamma}(k/2)$ of the $(k-1)$-sphere with the radius $r = q^{1/2}$ in the k-dimensional Euclidian space and $\mathrm{d}r = \{1/(2q^{1/2})\}\mathrm{d}q$:

$$\begin{aligned}&\left\{\textstyle\prod_{i=1}^{k} (1/\sqrt{2\pi}) \exp(-x_i^2/2)|_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q}\right\} \textstyle\int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k \\ &= \frac{1}{(2\pi)^{k/2}} \exp\left(-\tfrac{q}{2}\right) \frac{2\pi^{k/2}r^{k-1}}{\mathit{\Gamma}(k/2)} \frac{\mathrm{d}r}{\mathrm{d}q} = \frac{1}{(2\pi)^{k/2}} \exp\left(-\tfrac{q}{2}\right) \frac{2\pi^{k/2}q^{(k-1)/2}}{\mathit{\Gamma}(k/2)} \frac{1}{2q^{1/2}} \\ &= \frac{1}{2^{k/2}\mathit{\Gamma}(k/2)} q^{(k/2)-1} \exp\left(-\tfrac{q}{2}\right),\end{aligned}$$

yielding

$$\int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k = \frac{2\pi^{k/2}q^{(k-1)/2}}{\mathit{\Gamma}(k/2)} \frac{1}{2q^{1/2}} = \frac{\pi^{k/2}q^{(k/2)-1}}{\mathit{\Gamma}(k/2)}.$$

Khatri (1963, p. 53) stated that $\int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k = \pi^{k/2}q^{k/2}/\mathit{\Gamma}(k/2)$ using our notation, where $q^{k/2}$ rather than $q^{(k/2)-1}$ is probably a typo since otherwise the correct factor $|\mathbf{S}|^{(n-p-1)/2}$ corresponding to $q^{(k/2)-1}$ when $k = n - p + 1$ in his subsequent expression of the Wishart density does not follow. An alternative short derivation of $\int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k$ was given by Ogasawara (2022) as follows. Suppose that the pdf of the chi-square with k df, which is equal to that of the gamma with the shape parameter $k/2$ and the scale parameter 2, is obtained by a different method using e.g., the property of the distribution that the sum of the independent gamma distributed variables with the same scale parameter becomes the gamma with the shape parameter being the sum of those of the gammas and the same scale. Note that the beta integral or the moment generating function can be used for the derivation of this property. Then, we have

$$\left\{\prod_{i=1}^{k} (1/\sqrt{2\pi}) \exp(-x_i^2/2)|_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q}\right\} \int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k = \frac{q^{(k/2)-1}\exp(-q/2)}{2^{k/2}\mathit{\Gamma}(k/2)},$$

which gives

$$\int_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q} \mathrm{d}x_1 \cdots \mathrm{d}x_k = \frac{q^{(k/2)-1}\exp(-q/2)/\{2^{k/2}\Gamma(k/2)\}}{\prod_{i=1}^{k}(1/\sqrt{2\pi})\exp(-x_i^2/2)|_{\mathbf{x}^{\mathrm{T}}\mathbf{x}=q}} = \frac{\pi^{k/2}q^{(k/2)-1}}{\Gamma(k/2)}.$$

We find that this derivation without using the area of the $(k-1)$-sphere is similar to that by Anderson (2003) mentioned in Remark 4.

Proof 6 of the Wishart density in Theorem 2 (Khatri, 1963) Khatri's 1.5-page short derivation is due partially to his concise description. Since the article is less well documented with no title, the citations mentioned earlier using the same incorrect page numbers and several possible typos including the above one for important points and other minor errors, the corrected proof is provided with some added explanations. The derivation consists of a p-step variable transformation with p Jacobians canceling most of them after multiplication.

Define the $p\times n$ matrix $\mathbf{X}_\Sigma$, where each column independently follows $\mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$. Partition $\mathbf{S}_\Sigma = \mathbf{X}_\Sigma\mathbf{X}_\Sigma^{\mathrm{T}} = \begin{bmatrix} \mathbf{S}_{p-1} & \mathbf{s}_{p-1} \\ \mathbf{s}_{p-1}^{\mathrm{T}} & s_{pp} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{p-1}\mathbf{X}_{p-1}^{\mathrm{T}} & \mathbf{X}_{p-1}\mathbf{x}_p \\ \mathbf{x}_p^{\mathrm{T}}\mathbf{X}_{p-1}^{\mathrm{T}} & \mathbf{x}_p^{\mathrm{T}}\mathbf{x}_p \end{bmatrix}$, where e.g., s_{pp} is temporarily used in place of $s_{\Sigma pp}$ for simplicity. Define the $n \times n$ matrix $\mathbf{P}_n = \begin{bmatrix} \mathbf{X}_{p-1} \\ \mathbf{Y}_{n-p+1} \end{bmatrix}$, where the $(n-p+1)\times n$ submatrix $\mathbf{Y}_{n-p+1}$ is chosen such that $\mathbf{Y}_{n-p+1}\mathbf{X}_{p-1}^{\mathrm{T}} = \mathbf{O}$ and $\mathbf{Y}_{n-p+1}\mathbf{Y}_{n-p+1}^{\mathrm{T}} = \mathbf{I}_{n-p+1}$. Then, we have $\mathbf{P}_n\mathbf{P}_n^{\mathrm{T}} = \begin{bmatrix} \mathbf{S}_{p-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{n-p+1} \end{bmatrix}$, which gives $|\mathbf{P}_n|_+ = |\mathbf{P}_n\mathbf{P}_n^{\mathrm{T}}|^{1/2} = |\mathbf{S}_{p-1}|^{1/2}$. Consider the variable transformation from $\mathbf{x}_p$ to $\mathbf{P}_n\mathbf{x}_p$ with $(\mathbf{s}_{p-1}^{\mathrm{T}},\ \mathbf{z}_{n-p+1}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{P}_n\mathbf{x}_p$, where $\mathbf{z}_{n-p+1} \equiv \mathbf{Y}_{n-p+1}\mathbf{x}_p$ and $J(\mathbf{x}_p \to \mathbf{P}_n\mathbf{x}_p) = |\mathbf{P}_n|_+^{-1} = |\mathbf{S}_{p-1}|^{-1/2}$. Since

$$\begin{aligned} s_{pp} &= \mathbf{x}_p^{\mathrm{T}}\mathbf{x}_p = (\mathbf{s}_{p-1}^{\mathrm{T}},\ \mathbf{z}_{n-p+1}^{\mathrm{T}})\mathbf{P}_p^{\mathrm{T}-1}\mathbf{P}_p^{-1}(\mathbf{s}_{p-1}^{\mathrm{T}},\ \mathbf{z}_{n-p+1}^{\mathrm{T}})^{\mathrm{T}} \\ &= (\mathbf{s}_{p-1}^{\mathrm{T}},\ \mathbf{z}_{n-p+1}^{\mathrm{T}}) \begin{bmatrix} \mathbf{S}_{p-1}^{-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{n-p+1} \end{bmatrix} \begin{bmatrix} \mathbf{s}_{p-1} \\ \mathbf{z}_{n-p+1} \end{bmatrix} = \mathbf{s}_{p-1}^{\mathrm{T}}\mathbf{S}_{p-1}^{-1}\mathbf{s}_{p-1} + \mathbf{z}_{n-p+1}^{\mathrm{T}}\mathbf{z}_{n-p+1}, \end{aligned}$$

we have $\mathbf{z}_{n-p+1}^{\mathrm{T}}\mathbf{z}_{n-p+1} = s_{pp} - \mathbf{s}_{p-1}^{\mathrm{T}}\mathbf{S}_{p-1}^{-1}\mathbf{s}_{p-1} = |\mathbf{S}_\Sigma|/|\mathbf{S}_{p-1}|$.

Using the multivariate normal density, the joint marginal density of $\mathbf{X}_{p-1}$, when a random matrix $\mathbf{S}_\Sigma$ at $\mathbf{S}_\Sigma$ is a fixed one, becomes

$$\begin{aligned} & f_{\mathbf{X}_{p-1}}(\mathbf{X}_{p-1}) \equiv f_{\mathbf{X}_{p-1}} \\ &= \textstyle\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (2\pi)^{-np/2}|\mathbf{\Sigma}|^{-n/2}\exp\{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_\Sigma)/2\} \\ &\qquad \times J\{\mathbf{x}_p \to (\mathbf{s}_{p-1}^{\mathrm{T}},\ \mathbf{z}_{n-p+1}^{\mathrm{T}})^{\mathrm{T}}\}\mathrm{d}z_1 \cdots \mathrm{d}z_{n-p+1} \\ &= \textstyle\int_{-\infty}^{\infty} (2\pi)^{-np/2}|\mathbf{\Sigma}|^{-n/2}\exp\{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_\Sigma)/2\}\,|\mathbf{S}_{p-1}|^{-1/2}\mathrm{d}\mathbf{z}_{n-p+1}, \end{aligned}$$

where the integrand does not include $\mathbf{z}_{n-p+1}$. Then, the above integral becomes

$$
\begin{aligned}
f_{\mathbf{X}_{p-1}} &= (2\pi)^{-np/2}|\boldsymbol{\Sigma}|^{-n/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}\,|\mathbf{S}_{p-1}|^{-1/2} \\
&\quad\times \int_{\mathbf{z}_{n-p+1}^{\mathrm{T}}\mathbf{z}_{n-p+1}=\,|\mathbf{S}_{\Sigma}|/|\mathbf{S}_{p-1}|} \mathrm{d}\mathbf{z}_{n-p+1} \\
&= (2\pi)^{-np/2}|\boldsymbol{\Sigma}|^{-n/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}\,|\mathbf{S}_{p-1}|^{-1/2} \\
&\quad\times\frac{\pi^{(n-p+1)/2}\,(|\mathbf{S}_{\Sigma}|/|\mathbf{S}_{p-1}|)^{\{(n-p+1)/2\}-1}}{\Gamma\{(n-p+1)/2\}} \\
&= \frac{\pi^{(n-p+1)/2}|\mathbf{S}_{\Sigma}|^{(n-p-1)/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}\Gamma\{(n-p+1)/2\}}\frac{1}{|\mathbf{S}_{p-1}|^{(n-p)/2}},
\end{aligned}
$$

where Khatri's (p. 54) expression $|\mathbf{S}_{p-1}|(n-p-2)/2$ using our notation in place of $|\mathbf{S}_{p-1}|^{(n-p)/2}$ is incorrect. Define $\mathbf{X}_{p-i}\{(p-i)\times n\}$, $\mathbf{Y}_{n-p+i}\{(n-p+i)\times n\}$, $\mathbf{S}_{p-i}\{(p-i)\times(p-i)\}$, $\mathbf{s}_{p-i}\{(p-i)\times 1\}$ and $\mathbf{z}_{n-p+i}\{(n-p+i)\times 1\}$ $(i=2,...,p-1)$ similarly to those when $i=1$, respectively. Then, using these matrices and vectors in similar manners, we have the successive transformations as

$$
\begin{aligned}
f_{\mathbf{X}_1} &= \frac{\pi^{(n-p+1)/2}|\mathbf{S}_{\Sigma}|^{(n-p-1)/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}\Gamma\{(n-p+1)/2\}}\frac{1}{|\mathbf{S}_{p-1}|^{(n-p)/2}} \\
&\quad\times \textstyle\prod_{i=2}^{p-1}\int_{\mathbf{z}_{n-p+i}^{\mathrm{T}}\mathbf{z}_{n-p+i}=\,|\mathbf{S}_{p-i+1}/|\mathbf{S}_{p-i}|} \mathrm{d}\mathbf{z}_{n-p+i} \\
&= \frac{\pi^{(n-p+1)/2}|\mathbf{S}_{\Sigma}|^{(n-p-1)/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}\Gamma\{(n-p+1)/2\}}\frac{1}{|\mathbf{S}_{p-1}|^{(n-p)/2}} \\
&\quad\times\prod_{i=2}^{p-1}\frac{\pi^{(n-p+i)/2}\ \ |\mathbf{S}_{p-i+1}|^{(n-p+i-2)/2}/|\mathbf{S}_{p-i}|^{\{(n-p+i-1)/2}}{\Gamma\{(n-p+i)/2\}} \\
&= \frac{\pi^{[(n-p)(p-1)+\{p(p-1)/2\}-np]/2}|\mathbf{S}_{\Sigma}|^{(n-p-1)/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\prod_{i=1}^{p-1}\Gamma\{(n-p+i)/2\}}|\mathbf{S}_1|^{-(n-2)/2} \\
&= \frac{|\mathbf{S}_{\Sigma}|^{(n-p-1)/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\pi^{p(p-1)/4}\prod_{i=1}^{p}\Gamma\{(n-p+i)/2\}}\times\frac{(\mathbf{X}_1\mathbf{X}_1^{\mathrm{T}})^{-(n-2)/2}}{\pi^{n/2}/\Gamma(n/2)}.
\end{aligned}
$$

Noting that $(\mathbf{X}_1\mathbf{X}_1^{\mathrm{T}})^{-(n-2)/2}=|\mathbf{S}_1|^{-(n-2)/2}=s_{11}^{-(n-2)/2}$ is a fixed quantity, the last step is the integral with respect to the row vector $\mathbf{X}_1$:

$$
\begin{aligned}
&w_p(\mathbf{S}_{\boldsymbol{\Sigma}}|\boldsymbol{\Sigma},n) = f_{\mathbf{X}_1}\textstyle\int_{\mathbf{X}_1\mathbf{X}_1^{\mathrm{T}}=s_{11}}\mathrm{d}\mathbf{X}_1 = f_{\mathbf{X}_1}\pi^{n/2}s_{11}^{(n/2)-1}/\Gamma(n/2) \\
&= \frac{|\mathbf{S}_{\Sigma}|^{(n-p-1)/2}\exp\{-\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\Sigma})/2\}}{2^{np/2}|\boldsymbol{\Sigma}|^{n/2}\pi^{p(p-1)/4}\prod_{i=1}^{p}\Gamma\{(n-p+i)/2\}}.
\end{aligned}
$$

□

Appendix C The exchanged Bartlett decomposition

The Bartlett decomposition $\mathbf{S}=\mathbf{T}\mathbf{T}^{\mathrm{T}}$ has been used in this paper as well as in literatures. Let $\mathbf{S}=\mathbf{U}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}(\neq\mathbf{T}^{\mathrm{T}})$ is the upper-triangular matrix whose non-zero elements are random variables. Note that $\mathbf{U}$ can be obtained by rotating $\mathbf{T}$ as $\mathbf{U} = \mathbf{TV}$ using an orthonormal matrix $\mathbf{V}$. Define the upper-triangular matrix $\mathbf{C}$ satisfying $\mathbf{\Sigma} = \mathbf{CC}^{\mathrm{T}}$ with $c_{ii} > 0$ $(i = 1, ..., p)$, where $\mathbf{C}$ is obtained by $\mathbf{C} = \mathbf{BV}^*$ and $\mathbf{V}^*$ is another orthonormal matrix. Recall that the Cholesky decomposition $\mathbf{\Sigma} = \mathbf{BB}^{\mathrm{T}}$ was used earlier. The form $\mathbf{\Sigma} = \mathbf{CC}^{\mathrm{T}}$ is also called the exchanged (reversed) Cholesky or upper-lower (UL) decomposition in this paper.

Remark 5 Consider the distribution of $u_{ij}(i = 1, ..., p; j = i, ..., p)$, which are assumed to be mutually independent. As in the case of the usual Bartlett, Lemma 1 shows that when u_{ii} is chi-distributed with $n - p + i$ df $(i = 1, ..., p)$ and u_{ij} is standard normal $(i = 1, ..., p; j = i + 1, ..., p)$, the distribution of $\mathbf{S} = \mathbf{XX}^{\mathrm{T}}(= \mathbf{TT}^{\mathrm{T}})$ is the same as that of $\mathbf{UU}^{\mathrm{T}}$. Note that t_{ii} is chi-distributed with $n-i+1$ df rather than $n - p + i$. The joint pdf of $\mathbf{U}$ denoted by $f_p(\mathbf{U})$ becomes

$$\begin{aligned} f_p(\mathbf{U}) &= \left[\prod_{i=1}^{p} \frac{u_{ii}^{n-p+i-1}\exp(-u_{ii}^2/2)}{2^{\{(n-p+i)/2\}-1}\mathit{\Gamma}\{(n-p+i)/2\}} \right] \\ &\quad \times \frac{1}{(\sqrt{2\pi})^{(p^2-p)/2}} \left\{ \prod_{1 \le i<j \le p} \exp\left(-u_{ij}^2/2\right) \right\} \\ &= \frac{\left\{ \prod_{i=1}^{p} u_{ii}^{n-p+i-1}\exp(-u_{ii}^2/2) \right\} \left\{ \prod_{1 \le i<j \le p} \exp\left(-u_{ij}^2/2\right) \right\}}{2^{\frac{(n-p)p}{2}+\frac{p(p+1)}{4}-p} \times 2^{\frac{p(p-1)}{4}} \pi^{\frac{p(p-1)}{4}} \prod_{i=1}^{p} \mathit{\Gamma}\{(n-p+i)/2\}} \\ &= \frac{\left(\prod_{i=1}^{p} u_{ii}^{n-p+i-1} \right) \exp\{-\mathrm{tr}(\mathbf{UU}^{\mathrm{T}})/2\}}{2^{\frac{np}{2}-p}\mathit{\Gamma}_p(n/2)}. \end{aligned}$$

Proof 7 of the Wishart density in Theorem 2 Consider the one-step transformation from $\mathbf{U}$ to $\mathbf{S}_{\Sigma} = \mathbf{CXX}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} = \mathbf{CSC}^{\mathrm{T}} = \mathbf{CUU}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}$, where it is found that $\mathbf{C}(\mathbf{X})_{\cdot j} \overset{\text{i.i.d.}}{\sim} \mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$ $(j = 1, ..., n)$. Redefine the vector of the non-duplicated elements in $\mathbf{S}_{\Sigma}$ as $\mathbf{s}_{\Sigma} = (s_{\Sigma 11}, ..., s_{\Sigma 1p}, s_{\Sigma 22}, ..., s_{\Sigma 2p}, ..., s_{\Sigma pp})^{\mathrm{T}}$ whose elements are lexicographically ordered Similarly, define the $\{p(p+1)/2\} \times 1$ vectors $\mathbf{c}$ and $\mathbf{u}$ using the corresponding elements of $\mathbf{C}$ and $\mathbf{U}$, respectively.

The proof is similar to Proof 1 of Lemma 5. Since $\mathbf{C}$, $\mathbf{U}$ and $\mathbf{CU}$ are upper-triangular, the Jacobian matrix $\partial \mathbf{s}_{\Sigma}/\partial \mathbf{u}^{\mathrm{T}} = \{\partial s_{\Sigma ij}/\partial u_{kl}\}$ $(1 \le i \le j \le p; 1 \le k \le l \le p)$ becomes upper-triangular, whose diagonal elements are

$$\frac{\partial s_{\Sigma ij}}{\partial u_{ij}} = \{\mathbf{C}(\mathbf{E}_{ij}\mathbf{U}^{\mathrm{T}} + \mathbf{UE}_{ji})\mathbf{C}^{\mathrm{T}}\}_{ij} = (\mathbf{CE}_{ij}\mathbf{U}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}})_{ij} = c_{ii}u_{jj}c_{jj} \ (1 \le i < j \le p)$$

and

$$\frac{\partial s_{\Sigma ii}}{\partial u_{ii}} = \{\mathbf{C}(\mathbf{E}_{ii}\mathbf{U}^{\mathrm{T}} + \mathbf{UE}_{ii})\mathbf{C}^{\mathrm{T}}\}_{ii} = 2c_{ii}^2 u_{ii} \ (i = 1, ..., p).$$

Since the determinant of the Jacobian matrix or $J(\mathbf{S}_{\Sigma} \to \mathbf{U})$ becomes

$$\begin{aligned}&\textstyle\prod_{i=1}^{p}\prod_{j=i}^{p}\frac{\partial s_{\Sigma ij}}{\partial u_{ij}} = \left(\prod_{i=1}^{p}\prod_{j=i+1}^{p}\frac{\partial s_{\Sigma ij}}{\partial u_{ij}}\right)\prod_{i=1}^{p}\frac{\partial s_{\Sigma ii}}{\partial u_{ii}} = 2^p\prod_{i=1}^{p}\prod_{j=i}^{p}c_{ii}u_{jj}c_{jj}\\ &\textstyle= 2^p\left(\prod_{i=1}^{p}c_{ii}^{p-i+1}\right)\prod_{j=1}^{p}u_{jj}^{j}c_{jj}^{j} = 2^p\prod_{i=1}^{p}c_{ii}^{p+1}u_{ii}^{i} = 2^p|\mathbf{C}|^{p+1}\prod_{i=1}^{p}u_{ii}^{i}\\ &\textstyle= 2^p|\mathbf{\Sigma}|^{(p+1)/2}\prod_{i=1}^{p}u_{ii}^{i},\end{aligned}$$

$J(\mathbf{U} \to \mathbf{S}_{\Sigma})$ is given by the reciprocal of the above quantity.

The Wishart density is given by $f_p(\mathbf{U})$ and $J(\mathbf{U} \to \mathbf{S}_{\Sigma})$:

$$\begin{aligned}&w_p(\mathbf{S}_{\mathbf{\Sigma}}|\mathbf{\Sigma}, n) = f_p(\mathbf{U})J(\mathbf{U} \to \mathbf{S}_{\mathbf{\Sigma}})\\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{U}\mathbf{U}^{\mathrm{T}})/2\}\prod\limits_{i=1}^{p}u_{ii}^{n-p+i-1}}{2^{(np/2)-p}\mathit{\Gamma}_p(n/2)}\frac{|\mathbf{\Sigma}|^{-(p+1)/2}}{2^p\prod_{i=1}^{p}u_{ii}^{i}}\\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{U}\mathbf{U}^{\mathrm{T}})/2\}|\mathbf{\Sigma}|^{-(p+1)/2}\prod\limits_{i=1}^{p}u_{ii}^{n-p-1}}{2^{np/2}\mathit{\Gamma}_p(n/2)}\\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{C}^{-1}\mathbf{S}_{\mathbf{\Sigma}}\mathbf{C}^{\mathrm{T}-1})/2\}|\mathbf{\Sigma}|^{-(p+1)/2}|\mathbf{C}^{-1}\mathbf{S}_{\mathbf{\Sigma}}\mathbf{C}^{\mathrm{T}-1}|^{(n-p-1)/2}}{2^{np/2}\mathit{\Gamma}_p(n/2)}\\ &= \frac{\exp\{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_{\mathbf{\Sigma}})/2\}|\mathbf{S}_{\mathbf{\Sigma}}|^{(n-p-1)/2}}{2^{np/2}|\mathbf{\Sigma}|^{n/2}\mathit{\Gamma}_p(n/2)}\end{aligned}$$

as expected. □

Remark 6 Though $\mathbf{U} \neq \mathbf{T}^{\mathrm{T}}$ as noted earlier, $\mathbf{U}$ is obtained by reversing the row indexes of $\mathbf{T}$ followed by the similar reversal of the column ones. When $p = 3$, this transformation proceeds as

$$\mathbf{T} = \begin{bmatrix} t_{11} & 0 & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \to \begin{bmatrix} t_{31} & t_{32} & t_{33} \\ t_{21} & t_{22} & 0 \\ t_{11} & 0 & 0 \end{bmatrix} \to \begin{bmatrix} t_{33} & t_{32} & t_{31} \\ 0 & t_{22} & t_{21} \\ 0 & 0 & t_{11} \end{bmatrix} \equiv \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \mathbf{U}.$$

The above example indicates other decompositions $\mathbf{S} = \mathbf{T}^*\mathbf{T}^{*\mathrm{T}} = \mathbf{U}^*\mathbf{U}^{*\mathrm{T}}$ with the unchanged distribution of $\mathbf{S} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$, where $\mathbf{T}^*(\mathbf{U}^*)$ is a lower (upper)-triangular matrix defined with the non-zero elements on and below (above) the minor diagonals. Note that $\mathbf{T}^*$ and $\mathbf{U}^*$ are obtained by $\mathbf{T}$ and $\mathbf{U}$ by reversing the row or column indexes. When $p = 3$, $\mathbf{T}^*$ and $\mathbf{U}^*$ are $\begin{bmatrix} 0 & 0 & t_{11} \\ 0 & t_{22} & t_{21} \\ t_{33} & t_{32} & t_{31} \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 & t_{33}^* \\ 0 & t_{22}^* & t_{23}^* \\ t_{31}^* & t_{32}^* & t_{33}^* \end{bmatrix}$ and $\begin{bmatrix} u_{13} & u_{12} & u_{11} \\ u_{23} & u_{22} & 0 \\ u_{33} & 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} u_{11}^* & u_{12}^* & u_{13}^* \\ u_{21}^* & u_{22}^* & 0 \\ u_{31}^* & 0 & 0 \end{bmatrix}$, respectively.

Actually, we have infinitely many transformations with the unchanged distribution of $\mathbf{S}$, including the above ones, using various orthonormal $p \times p$ matrices denoted by $\mathbf{V}$'s since each column of $\mathbf{VX}$ independently follows $\mathrm{N}_p(\mathbf{0}, \mathbf{I}_p)$ (see e.g., Anderson, 2003, Theorem 3.3.1). In other words, the distributions of $\mathbf{VX}$ and $\mathbf{X}$ are the same. Then, $\mathbf{S} = \mathbf{XX}^{\mathrm{T}}$ can be replaced by $\mathbf{S} = \mathbf{VXX}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}$. Note that one of the decomposed matrices e.g., $\mathbf{T}$, $\mathbf{T}^{*}$, $\mathbf{U}$ and $\mathbf{U}^{*}$ are given by other ones using $\mathbf{V}$ as $\mathbf{T} = \mathbf{VU}^{*}$. This indeterminacy of transformation is similar to the rotational indeterminacy in orthogonal rotation in factor analysis and canonical correlation analysis or more generally transformations in structural equation modeling (Ogasawara, 2007; Schuberth, 2021; Yu, Schuberth, & Henseler, 2023).

Journal of Behavioral Data Science, 2023, *3* (1), 59–69.
DOI: `https://doi.org/10.35566/jbds/v3n1/wyman`

API Face Value: Evaluating the Current Status and Potential of Emotion Detection Software in Emotional Deficit Interventions

Austin T. Wyman and Zhiyong Zhang

University of Notre Dame, Notre Dame, USA
`awyman@nd.edu`

Abstract. Emotion recognition application programming interface (API) is a recent advancement in computing technology that synthesizes computer vision, machine-learning algorithms, deep-learning neural networks, and other information to detect and label human emotions. The strongest iterations of this technology are produced by technology giants with large, cloud infrastructure (i.e., Google, and Microsoft), bolstering high true positive rates. We review the current status of applications of emotion recognition API in psychological research and find that, despite evidence of spatial, age, and race bias effects, API is improving the accessibility of clinical and educational research. Specifically, emotion detection software can assist individuals with emotion-related deficits (e.g., Autism Spectrum Disorder, Attention Deficit-Hyperactivity Disorder, Alexithymia). API has been incorporated in various computer-assisted interventions for Autism, where it has been used to diagnose, train, and monitor emotional responses to one's environment. We identify AP's potential to enhance interventions in other emotional dysfunction populations and to address various professional needs. Future work should aim to address the bias limitations of API software and expand its utility in subfields of clinical, educational, neurocognitive, and industrial-organizational psychology.

Keywords: API · Emotion Recognition · Machine Learning · ASD · ADHD · Alexithymia

Emotions, their expression, and understanding are often described as unique characteristics of human life and development; however, with the growing sophistication of computer vision and machine learning, computing technology is rapidly shrinking the disparity between human and artificial intelligence. This evolution is particularly marked by a redefinition of artificial intelligence (Lisetti & Schiano, 2000). While originally referring to computers' ability to perform cognitive tasks, artificial intelligence has now expanded to include a variety of subfields, including artificial wisdom (Jeste et al., 2020), and emotional intelligence (Erol et al., 2020; Poria, Majumder, Mihalcea, & Hovy, 2019). These advancements represent the latest hurdles computing technology must jump to match human intelligence capabilities (Schuller & Schuller, 2018), which has the potential to enhance psychological research. The expansion of emotion detection software is highly relevant to the improvement of measurement in psychological research and practice.

1 Introduction to Emotion Recognition API

Application programming interface (API) is a broad term that describes any means of communication between two or more computer programs. In particular, emotion recognition APIs allow the synthesis of computer vision, machine learning algorithms, deep learning neural networks, and other components in order to accurately detect and label human emotions (Deshmukh & Jagtap, 2017). The emotion API performance is further enhanced by cloud-based support, which continuously supplies learning algorithms with severs full of facial and emotional data (Khanal, Barroso, Lopes, Sampaio, & Filipe, 2018). Naturally, technology giants with the largest cloud infrastructure (e.g., Amazon, Microsoft, Google), are the most equipped to construct accurate emotion recognition programs. While specific expressions that can be detected vary from program to program, most algorithms are minimally equipped to identify the six basic human emotions: disgust, contempt, anger, fear, surprise, and sadness (Deshmukh & Jagtap, 2017).

The leading iterations of this technology are Microsoft Azure and Google Cloud Vision, which offer distinct advantages over one another in emotion recognition (Khanal et al., 2018). Microsoft's API triumphs in overall accuracy, reporting high true positive (TP) rates for straight-facing and partially-straight facing profiles (Half Left Face TP = 60%; Straight Face TP = 74.9%; Half Right TP = 57.4%). Google's API, however, can detect a wider range of facial profiles, particularly side-facing, but with reduced accuracy (Full Left Face TP = 7.3%; Half Left TP = 42.9%; Straight TP = 45.2%; Half Right TP = 43.2%; Full Right TP = 10.4%). The lack of non-frontal facial recognition is a significant limitation, but the implementation of new machine learning frameworks is gradually improving detection accuracy (Lin, Ma, Gong, & Wang, 2022).

It is important to mention that programming limitations are often readily addressed in future software updates, but sampling limitations require more targeted attention. Emotion APIs are typically trained with large samples of facial and corresponding emotion data, but a lack of diverse data often makes it difficult to account for physiological differences in emotion expression among different groups (Hernandez et al., 2021). Due to convenience sampling, training samples predominantly consist of white, young adults in America. This produces significant racial and age bias effects, which further confound previous accuracy estimates. For example, Microsoft and Amazon's APIs are more likely to label Black participants' neutral faces as angry or contemptuous compared to white participants exhibiting the same emotion (Kyriakou, Kleanthous, Otterbacher, & Papadopoulos, 2020; Rhue, 2018). Additionally, these programs demonstrate reduced accuracy with middle aged and older adults compared to young adult participants (Kim, Bryant, Srikanth, & Howard, 2021). Gender bias used to be a serious concern in previous API iterations (Klare, Burge, Klontz, Vorder-Bruegge, & Jain, 2012), but current research suggests that this disparity was addressed in recent updates (Kim et al., 2021). Whether through sampling adjustment or algorithmic improvement (Howard, Zhang, & Horvitz, 2017), it is likely the emotion recognition API will become more accurate with respect to racial and age bias, but progress in this area requires selective attention to improving representation.

2 Current Applications of Emotion Recognition API

The present review searched electronic databases (i.e., Google Scholar and Psyinfo) using five term categories: "emotion API", "emotion detection", "psychology", "intervention", and "emotion deficit." Studies published since 2017 were included in the review if they (a) were published in English, (b) developed a new intervention using emotion recognition API, and (c) targeted individuals with emotion-related deficits. Current publications on emotion recognition API have seldom reached the mainstream of psychology research, as most studies exploring this technology have focused more on the computer science and algorithmic strength of software than its applications in measuring psychological constructs. Nonetheless, key methodologies have emerged in clinical, neurodevelopmental, and educational psychology research (See Table 1).

Table 1. Current Applications of Emotion Recognition API in Psychology

Author (year)	Area	Software
Alharbi and Huang (2020)	Clinical	Microsoft
Bharatharaj, Huang, Mohan, Al-Jumaily, and Krägeloh (2017)	Clinical	Oxford
Grossard et al. (2017)	Clinical	-
Jiang et al. (2019)	Clinical	-
Liu, Wu, Zhao, and Luo (2017)	Clinical	-
Manfredonia et al. (2018)	Clinical	FACET
Chu, Tsai, Liao, and Chen (2017)	Educational	FACEAPI
Chu, Tsai, Liao, Chen, and Chen (2020)	Educational	Face Tracking API 3.2
Borsos, Jakab, Stefanik, Bogdán, and Gyori (2022)	Quantitative	FR8
Flynn et al. (2020)	Quantitative	iMotions

In clinical and neurodevelopmental areas, emotion detection software has assisted with the monitoring, treatment, and education of various individuals with emotion-related deficits. Byrne, Bogue, Egan, and Lonergan (2016) writes that "psychological mindedness," the process of identifying and describing emotions, is an "explicit mentalizing capacity that is needed to engage effectively in psychotherapy." Many talk-therapy techniques rely upon a baseline level of emotional intelligence, requiring that individuals are able to understand their own and others' emotions. However, clients with emotional deficits struggle with emotion recognition, and, therefore, may not benefit from talk-therapy. Thus, emotion-related interventions are an important gateway step to other substantive areas of mental health treatment.

An expansive body of literature has investigated interventions for improving psychological mindedness in neurodevelopmental disorders, particularly Autism Spectrum Disorder (ASD) and Attention Deficit-Hyperactivity Disorder (ADHD). These populations often struggle with reduced empathy (Baron-Cohen & Wheelwright, 2004; Da Fonseca, Seguier, Santos, Poinso, & Deruelle, 2009; Uekermann et al., 2010) and emotion self-regulation (Braaten & Rosen, 2000), producing significant behavioral problems (Milton, 2012). The prime window to treat emotional deficits is during early childhood, but many individuals are diagnosed later in life. As the brain matures and patterns of dysfunctional social cognition become fixed, it is incredibly difficult to teach fundamental skills like empathy (Baron-Cohen, 2009). Given the behavioral consequences of ASD and ADHD in adolescents and adults, it is important that current emotional-deficit interventions are expanded to include populations in late-stage treatment.

Emotion recognition API, thus, is valuable because it can be readily incorporated in a variety of intervention settings and stages, from diagnosis to late-stage treatment (Liu et al., 2017). Regarding diagnosis, Manfredonia et al. (2018) used facial expression analysis software to measure differences in emotion expression and replicated diagnoses for ASD participants, ranging from 9-years-old to 54-years-old. Similarly, Jiang et al. (2019) synthesized emotion recognition and eye-tracking software to achieve a diagnosis accuracy rate that was competitive with those by professional psychologists. Post diagnosis, emotion API has been used to provide engaging education for ASD participants to build emotion-related skills. For example, Bharatharaj et al. (2017) developed a semi-autonomous robot presented as a toy parrot, which used Oxford API to monitor emotion regulation and practice social interaction with ASD children. Alharbi and Huang (2020) designed computer games that reward ASD children for accurately matching facial expressions in order to train empathy and communication skills. Many other popular games have been adapted using emotion API and computer-assisted instruction (Grossard et al., 2017), which improves both the accessibility and entertainment of diagnostic and intervention strategies for children with neurodevelopmental disorders.

The concern of emotion regulation in children also emerges within educational psychology literature, with multiple studies demonstrating that students with better emotion regulation ability perform better in the classroom and have higher levels of academic achievement (Gumora & Arsenio, 2002; Howse, Calkins, Anastopoulos, Keane, & Shelton, 2003). Naturally, students with emotional deficits, such as ASD, report much lower academic achievement rates than typically developing students (Ashburner, Ziviani, & Rodger, 2010). In E-Learning environments, emotion recognition API has been used to detect emotion changes in students with ASD during assessments (Chu et al., 2017), for targeted intervention strategies. This intervention was followed up by (Chu et al., 2020), which designed an emotion API-based intervention that utilized computer adaptive testing to identify and address learning stress in students with ASD. The result of this intervention significantly improved students' math performance compared with baseline scores.

From a measurement perspective, some studies have raised concern about the reliability of the software's emotion estimates. Borsos et al. (2022) evaluated the test-retest reliability of emotion API and found small but significant differences in the ratings. Flynn et al. (2020) observed group differences in the accuracy of emotion estimates between children and adults. However, it is important to note that both of these studies used emotion detection software (FR8 and iMotions respectively) that is meagerly discussed in the literature compared to the API produced by tech giants (e.g., Google Cloud and Microsoft Azure). These limitations are likely not representative of the method as a whole because these studies are operating on less-than-standard measurement tools. Nonetheless, inconsistency in emotion API responses are to be expected to some extent, which highlights the imperfect nature of emotion estimates. However, the adaptability of emotion detection software is a critical strength of this measurement approach, and as the software is incrementally improved over time, the accuracy of emotion estimates will also improve.

3 Potential Applications of Emotion Recognition API

Beyond neurodevelopmental disorders, clinical literature expresses a need for interventions to address a wide-range of psychopathology exhibiting emotion-related deficits. Alexithymia and empathy-related concerns are present in many other disorder classifications, particularly personality disorders (De Panfilis, Ossola, Tonna, Catania, & Marchesi, 2015; Thoma, Friedmann, & Suchan, 2013), and often lead to interpersonal dysfunction (Cook, Brewer, Shah, & Bird, 2013; Vanheule, Desmet, Meganck, & Bogaerts, 2007), internalizing and externalizing behavior (Aldao et al., 2016). Current personality pathology interventions often rely on self-report instruments, which have various validity concerns (Haeffel & Howard, 2010). Thus, the increased availability of emotion detection software has the potential to expand the range of options in how emotion-related experiments are designed. Emotion API has demonstrated its effectiveness in predicting Big Five personality traits and risk-taking behavior (Gloor et al., 2022), which is a significant facet of pathological personality (Watson & Clark, 2020). Detection software could be readily incorporated into studies interested in examining operational ways of measuring emotion dysregulation and psychopathological traits.

Regarding interventions, API strategies for neurodevelopmental disorders have not been tested on other psychopathology, but these interventions could generalize well with disorders that exhibit similar transdiagnostic traits. For example, Antisocial Personality Disorder and Narcissistic Personality Disorders often overlap with ASD and ADHD (Matthies & Philipsen, 2016). Emotion API could be an incredibly valuable tool in the measurement and design of pathological personality interventions beyond the scope of its current self-report methodology, which could benefit researchers and practitioners alike.

Emotion API interventions could also generalize to the broader, industrial-organizational need for better emotional intelligence trainings. Emotional intelligence is frequently measured in industrial-organizational contexts and is associated with multiple occupational outcomes, including job performance, retention, and interpersonal relations (Prentice, Lopes, & Wang, 2020). Thus, many industries declare a strong vested interest in screening for candidates with high emotional intelligence, or enhancing the emotional intelligence of their current employees. Facial expression is often described as a basic facet of emotional intelligence (Hildebrandt, Sommer, Schacht, & Wilhelm, 2015), and is often a targeted topic in emotional intelligence training programs. Employers and industrial-organizational researchers could capitalize off the automated and adaptive features of emotion recognition API to quickly improve employees' emotional intelligence ability. API-based programs in emotion regulation could be inserted as a complement to existing modules on effective nonverbal communication and empathy.

As mentioned previously, emotion regulation is a critical component of students' success in the classroom (Gumora & Arsenio, 2002; Howse et al., 2003), but other aspects of emotional functioning are relevant as well. Despite a common avoidance to express negative emotions, literature shows that negative emotions are a way to elicit support and build stronger relationships (Graham et al., 2008). Students who less openly express their emotions are less likely to receive help when struggling because they are often unable to call attention to signs of distress. That said, similar emotion expression interventions to the ones currently used for ASD could be helpful to acclimate these types of students to the importance of emotional intelligence. Alternatively, emotion API could be integrated into research focusing on instructors. Literature suggests that the emotion regulation ability of instructors also impacts student engagement and success in the classroom (Sutton et al., 2009; Wang & Ye, 2021). Detection software could complement classroom observation studies, generating ecological momentary assessments of instructors and their emotion regulation ability over the course of a lecture, which may be more accurate and reliable than current self-report or interview assessment strategies.

Broadly speaking, the integration of computational research methods would greatly benefit all areas of psychology, and this can especially be seen with emotion recognition API. The software allows researchers to easily collect and assign quantitative values to emotion-related data (Yannakakis, Cowie, & Busso, 2021), which increases the feasibility of collecting larger data without compromising quality of data. Emotion recognition API triumphs in efficiency over traditional measurement attempts, which are often long, unreliable, and cumbersome. Neurocognitive research could especially benefit from an increased efficiency in data collection, which is a contributing factor to concerns of low statistical power in current research (Button et al., 2013). An upgrade in statistical power is highly important and has the potential to increase the frequency and reproducibility of emotion-related research in clinical trials and neurocognitive work.

4 Conclusion

Although the integration of emotion recognition API is very much in its infancy in psychology, several subfields would benefit from an expansion of this highly adaptive area of measurement. Clinical research could enhance current intervention, develop new models of treatment, and establish new methods of measuring emotional functioning domains. Industrial-organizational research could develop new emotional intelligence indexes and training programs. Educational research could identify new ways of identifying and supporting students in the classroom. And neurocognitive research could generate more power and enhance the precision of neural mechanisms behind emotional expression. As this technology becomes more accessible, future studies should investigate API in all of these important disciplines and other, unidentified yet equally important areas. Although there are significant concerns of reliability and bias in the software currently, the incremental improvement of cloud-based programs confidently suggests that API is becoming a more reliable tool. Understanding emotions is a fundamental facet of human life experiences and emotion recognition API will allow psychologists to understand this phenomenon even further.

Acknowledgement

This study is supported by a grant from the Department of Education (R305D210023). However, the contents of the study do not necessarily represent the policy of the Department of Education, and you should not assume endorsement by the Federal Government.

References

Alharbi, M., & Huang, S. (2020). An augmentative system with facial and emotion recognition for improving social skills of children with autism spectrum disorders. In *2020 IEEE International Systems Conference (SysCon)* (pp. 1–6). doi: https://doi.org/10.1109/SysCon47679.2020.9275659

Ashburner, J., Ziviani, J., & Rodger, S. (2010). Surviving in the mainstream: Capacity of children with autism spectrum disorders to perform academically and regulate their emotions and behavior at school. *Research in Autism Spectrum Disorders*, *4*(1), 18–27. doi: https://doi.org/10.1016/j.rasd.2009.07.002

Baron-Cohen, S. (2009). Autism: The empathizing-systemizing (E-S) theory. *Annals of the New York Academy of Sciences*, *1156*(1), 68–80. doi: https://doi.org/10.1111/j.1749-6632.2009.04467.x

Baron-Cohen, S., & Wheelwright, S. (2004). The empathy quotient: An investigation of adults with asperger syndrome or high functioning autism, and normal sex differences. *Journal of Autism and Developmental Disorders*, *34*(2), 163–175. doi: https://doi.org/10.1023/B:JADD.0000022607.19833.00

Bharatharaj, J., Huang, L., Mohan, R. E., Al-Jumaily, A., & Krägeloh, C. (2017). Robot-assisted therapy for learning and social interaction of children with autism spectrum disorder. *Robotics*, *6*(1), 4. doi: https://doi.org/10.3390/robotics6010004

Borsos, Z., Jakab, Z., Stefanik, K., Bogdán, B., & Gyori, M. (2022). Test–retest reliability in automated emotional facial expression analysis: Exploring facereader 8.0 on data from typically developing children and children with autism. *Applied Sciences*, *12*(15), 7759. doi: https://doi.org/10.3390/app12157759

Braaten, E. B., & Rosen, L. A. (2000). Self-regulation of affect in attention deficit-hyperactivity disorder (adhd) and non-adhd boys: Differences in empathic responding. *Journal of Consulting and Clinical Psychology*, *68*(2), 313–321. doi: https://doi.org/10.1037/0022-006X.68.2.313

Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., & Munafò, M. R. (2013). Power failure: Why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, *14*, 365–376. doi: https://doi.org/10.1038/nrn3475

Byrne, G., Bogue, J., Egan, R., & Lonergan, E. (2016). "identifying and describing emotions": Measuring the effectiveness of a brief, alexithymia-specific intervention for a sex offender population. *Sexual Abuse: A Journal of Research and Treatment*, *28*(7), 599–619. doi: https://doi.org/10.1177/1079063214528822

Chu, H.-C., Tsai, W.-H., Liao, M.-J., & Chen, Y.-M. (2017). Facial emotion recognition with transition detection for students with high-functioning autism in adaptive e-learning. *Soft Computing*, *22*, 2973–2999. doi: https://doi.org/10.1007/s00500-017-2549-z

Chu, H.-C., Tsai, W.-H., Liao, M.-J., Chen, Y.-M., & Chen, J.-Y. (2020). Supporting e-learning with emotion regulation for students with autism spectrum disorder. *Educational Technology & Society*, *23*(4), 124–146. Retrieved from https://www.jstor.org/stable/26981748

Cook, R., Brewer, R., Shah, P., & Bird, G. (2013). Alexithymia, not autism, predicts poor recognition of emotional facial expressions. *Psychological Science*, *24*(5), 723–732. doi: https://doi.org/10.1177/0956797612463582

Da Fonseca, D., Seguier, V., Santos, A., Poinso, F., & Deruelle, C. (2009). Emotion understanding in children with adhd. *Child Psychiatry and Human Development*, *40*(1), 111–121. doi: https://doi.org/10.1007/s10578-008-0114-9

De Panfilis, C., Ossola, P., Tonna, M., Catania, L., & Marchesi, C. (2015). Finding words for feelings: The relationship between personality disorders and alexithymia. *Personality and Individual Differences*, *74*, 285–291. doi: https://doi.org/10.1016/j.paid.2014.10.050

Deshmukh, R. S., & Jagtap, V. (2017). A survey: Software api and database for emotion recognition. In *2017 international conference on intelligent computing and control systems (iciccs)* (pp. 284–289). doi: https://doi.org/10.1109/ICCONS.2017.8250727

Erol, B. A., Majumdar, A., Benavidez, P., Rad, P., Choo, K.-S., & Jamshidi, M. (2020). Toward artificial emotional intelligence for cooperative social human-machine interaction. *IEEE Transactions on Computational Social Systems*, *7*(1), 234–246. doi: https://doi.org/10.1109/TCSS.2019.2922593

Flynn, M., Effraimidis, D., Angelopoulou, A., Kapetanios, E., Williams, D., Hemanth, J., & Towell, T. (2020). Assessing the effectiveness of automated emotion recognition in adults and children for clinical investigation. *Frontiers in Human Neuroscience*, *14*. doi: https://doi.org/10.3389/fnhum.2020.00070

Gloor, P. A., Colladon, A. F., Altuntas, E., Cetinkaya, C., Kaiser, M. F., Ripperger, L., & Schaefer, T. (2022). Your face mirrors your deepest beliefs—predicting personality and morals through facial emotion recognition. *Future Internet*, *14*(1), 5. doi: https://doi.org/10.3390/fi14010005

Grossard, C., Grynspan, O., Serret, S., Jouen, A.-L., Bailly, K., & Cohen, D. (2017). Serious games to teach social interactions and emotions to individuals with autism spectrum disorders (asd). *Computers & Education*, *113*, 195–211. doi: https://doi.org/10.1016/j.compedu.2017.05.002

Gumora, G., & Arsenio, W. F. (2002). Emotionality, emotion regulation, and school performance in middle school children. *Journal of School Psychology*, *40*(5), 395–413. doi: https://doi.org/10.1016/S0022-4405(02)00108-5

Haeffel, G. J., & Howard, G. S. (2010). Self-report: Psychology's four-letter word. *The American Journal of Psychology*, *123*(2), 181–188. doi: https://doi.org/10.2307/40827643

Hernandez, J., Lovejoy, J., McDuff, D., Suh, J., O'Brien, T., Sethumadhavan, A., ... Czerwinski, M. (2021). Guidelines for assessing and minimizing risks of emotion recognition applications. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)* (pp. 1–8). doi: https://doi.org/10.1109/ACII52823.2021.9597452

Hildebrandt, A., Sommer, W., Schacht, A., & Wilhelm, O. (2015). Perceiving and remembering emotional facial expressions–A basic facet of emotional intelligence. *Intelligence*, *50*, 52–67. doi: https://doi.org/10.1016/j.intell.2015.02.003

Howard, A., Zhang, C., & Horvitz, E. (2017). Addressing bias in machine learning algorithms: A pilot study on emotion recognition for intelligent systems. In *2017 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)* (pp. 1–7). doi: https://doi.org/10.1109/ARSO.2017.8025197

Howse, R. B., Calkins, S. D., Anastopoulos, A. D., Keane, S. P., & Shelton, T. L. (2003). Regulatory contributors to children's kindergarten achievement. *Early Education and Development*, *14*(1), 101–120. doi: https://doi.org/10.1207/s15566935eed1401_7

Jeste, D. V., Graham, S. A., Nguyen, T. T., Depp, C. A., Lee, E. E., & Kim, H. (2020). Beyond artificial intelligence: Exploring artificial wisdom. *International Psychogeriatrics*, *32*(8), 993–1001. doi: https://doi.org/10.1017/S1041610220000927

Jiang, M., Francis, S. M., Srishyla, D., Conelea, C., Zhao, Q., & Jacob, S. (2019). Classifying individuals with asd through facial emotion recognition and eye-tracking. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 6063–6068). doi: https://doi.org/10.1109/EMBC.2019.8857005

Khanal, S. R., Barroso, J., Lopes, N., Sampaio, J., & Filipe, V. (2018). Performance analysis of microsoft's and google's emotion recognition api using pose-invariant faces. In *Proceedings of the 8th international conference on software development and technologies for enhancing accessibility and fighting info-exclusion* (pp. 172–178). doi: https://doi.org/10.1145/3218585.3224223

Kim, E., Bryant, D., Srikanth, D., & Howard, A. (2021). Age bias in emotion detection: An analysis of facial emotion recognition performance on young, middle-aged, and older adults. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 638–644). doi: https://doi.org/10.1145/3461702.3462609

Klare, B. F., Burge, M. J., Klontz, J. C., Vorder-Bruegge, R. W., & Jain, A. K. (2012). Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, *7*(6), 1789–1801. doi: https://doi.org/10.1109/TIFS.2012.2214212

Kyriakou, K., Kleanthous, S., Otterbacher, J., & Papadopoulos, G. A. (2020). Emotion-based stereotypes in image analysis services. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (pp. 252–259). doi: https://doi.org/10.1145/3386392.3399567

Lin, H., Ma, H., Gong, W., & Wang, C. (2022). Non-frontal face recognition method with a side-face-correction generative adversarial networks. In *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)* (pp. 563–567). doi: https://doi.org/10.1109/CVIDLICCEA56201.2022.9825237

Lisetti, C. L., & Schiano, D. J. (2000). Automatic facial expression interpretation: Where human-computer interaction, artificial intelligence and cognitive science intersect. *Pragmatics and Cognition*, *8*(1), 185–235. doi: https://doi.org/10.1075/pc.8.1.09lis

Liu, X., Wu, Q. J., Zhao, W., & Luo, X. (2017). Technology-facilitated diagnosis and treatment of individuals with autism spectrum disorder: An engineering perspective. *Applied Sciences*, *7*(10), 1051. doi: https://doi.org/10.3390/app7101051

Manfredonia, J., Bangerter, A., Manyakov, N. V., Ness, S., Lewin, D., Skalkin, A., … others (2018). Automatic recognition of posed facial expression of emotion in individuals with autism spectrum disorder. *Journal of Autism and Developmental Disorders*, *49*, 279–293. doi: https://doi.org/10.1007/s10803-018-3757-9

Matthies, S., & Philipsen, A. (2016). Comorbidity of personality disorders and adult attention deficit hyperactivity disorders (adhd)—review of recent findings. *Current Psychiatry Reports*, *18*(4), 1–7. doi: https://doi.org/10.1007/s11920-016-0675-4

Milton, D. E. M. (2012). On the ontological status of autism: The 'double empathy problem.'. *Disability & Society*, *27*(6), 883–887. doi: https://doi.org/10.1080/09687599.2012.710008

Poria, S., Majumder, N., Mihalcea, R., & Hovy, E. (2019). Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, *7*, 100943–100953. doi: https://doi.org/10.1109/ACCESS.2019.2929050

Prentice, C., Lopes, S. D., & Wang, X. (2020). Emotional intelligence or artificial intelligence—an employee perspective. *Journal of Hospitality Marketing & Management*, *29*(4), 377–403. doi: https://doi.org/10.1080/19368623.2019.1647124

Rhue, L. (2018). Racial influence on automated perceptions of emotions. *SSRN Electronic Journal*. doi: https://doi.org/10.2139/ssrn.3216634

Schuller, D., & Schuller, B. W. (2018). Computer. *Computer*, *51*(9), 38–46. doi: https://doi.org/10.1109/MC.2018.3620963

Thoma, P., Friedmann, C., & Suchan, B. (2013). Empathy and social problem solving in alcohol dependence, mood disorders and selected personality disorders. *Neuroscience & Biobehavioral Reviews*, *37*(3), 448–470. doi: https://doi.org/10.1016/j.neubiorev.2013.01.024

Uekermann, J., Kraemer, M., Abde-Hamid, M., Schimmelmann, B. G., Hebebrand, J., Daum, I., … Kis, B. (2010). Social cognition in attention-deficit hyperactivity disorders (adhd). *Neuroscience & Biobehavioral Reviews*, *34*(5), 734–743. doi: https://doi.org/10.1016/j.neubiorev.2009.10.009

Vanheule, S., Desmet, M., Meganck, R., & Bogaerts, S. (2007). Alexithymia and interpersonal problems. *Journal of Clinical Psychology*, *63*(1), 109–117. doi: https://doi.org/10.1002/jclp.20324

Watson, D., & Clark, L. A. (2020). Personality traits as an organizing framework for personality pathology. *Personality and Mental Health*, *14*, 51–75. doi: https://doi.org/10.1002/pmh.1458

Yannakakis, G. N., Cowie, R., & Busso, C. (2021). The ordinal nature of emotions: An emerging approach. *IEEE Transactions on Affective Computing*, *12*(1), 16–35. doi: https://doi.org/10.1109/TAFFC.2018.2879512

Journal of Behavioral Data Science, 2023, *3* (1), 70–83.
DOI:https://doi.org/10.35566/jbds/v3n1/s

Predicting Dyslexia with Machine Learning: A Comprehensive Review of Feature Selection, Algorithms, and Evaluation Metrics

Velmurugan S[0000−0003−1956−3674]

Department of Electrical Engineering, Indian Institute of technology Madras
Chennai, India
ee21s131@smail.iitm.ac.in

Abstract. This literature review explores the use of machine learning-based approaches for the diagnosis and treatment of dyslexia, a learning disorder that affects reading and spelling skills. Various machine learning models, such as artificial neural networks (ANNs), support vector machines (SVMs), and decision trees, have been used to classify individuals as either dyslexic or non-dyslexic based on functional magnetic resonance imaging (fMRI) and electroencephalography (EEG) data. These models have shown promising results for early detection and personalized treatment plans. However, further research is needed to validate these approaches and identify optimal features and models for dyslexia diagnosis and treatment.

Keywords: SVM · EEG · Dyslexia

1 Introduction

Dyslexia is a learning disorder that affects reading and spelling skills. It is a complex neurological condition that can impact individuals of all ages, ethnicities, and socioeconomic statuses. Early detection and intervention are crucial for managing dyslexia, and machine learning-based approaches have emerged as a promising tool for achieving this (Kaisar, 2020). Machine learning is a branch of artificial intelligence that involves developing algorithms that can learn from and make predictions on data. Machine learning models can be trained on large datasets of dyslexia-related information, such as functional magnetic resonance imaging (fMRI) and electroencephalography (EEG) data, to extract features and patterns that are associated with dyslexia. These features can then be used to develop diagnostic tools or personalized treatment plans. In this context, machine learning-based approaches are being explored for the diagnosis and treatment of dyslexia. These approaches involve the use of different machine learning algorithms, such as artificial neural networks (ANNs), support vector machines (SVMs), decision trees, and Bayesian networks, to classify individuals as either dyslexic or non-dyslexic based on specific features extracted from the data(Chakraborty, Vani,x& Sundaram, 2021). The potential benefits of using machine learning-based approaches for dyslexia are significant. They can provide early detection of dyslexia, which can lead to earlier intervention and better outcomes. Additionally, personalized treatment plans can be developed, which take into account individual characteristics such as age, gender, and severity of dyslexia, and can increase the likelihood of treatment success (Prabhax& Bhargavi, 2022; Rellox& Ballesteros, 2015). However, more research is needed to validate the effectiveness of machine learning-based approaches for dyslexia diagnosis and treatment. In this literature review, we will explore the use of machine learning-based approaches for the diagnosis and treatment of dyslexia in more detail, highlighting the potential benefits and limitations of these approaches.

2 Data Collection and Preprocessing

2.1 Datasets

The data collection process for dyslexia prediction involves obtaining samples from both dyslexic and non-dyslexic individuals. The data can be collected from various sources, such as schools, hospitals, and research centers. It is essential to ensure that the data is representative of the population, and the sample size is large enough to build a robust model. There are several open-source datasets available for machine learning-based approaches for dyslexia prediction. Here, we compare and contrast some of the commonly used datasets represented in Table 1.

2.2 Preprocessing Techniques

Preprocessing techniques are crucial in dyslexia prediction as they help in improving the accuracy and reliability of the data. Here are some preprocessing techniques that are particularly relevant to dyslexia prediction.

Data Cleaning Data cleaning is an essential step in preparing datasets for machine learning models. Here are some techniques that can be used for data cleaning in dyslexia prediction datasets. Outliers are data points that are significantly different from other data points in the dataset. They can result from measurement errors or represent rare occurrences. Outliers can significantly impact the accuracy of a predictive model, and therefore it is essential to detect and handle them appropriately (Ahmad, Rehman, Hassan, Ahmad,x& Rashid, 2022). Dyslexia prediction often involves working with categorical data such as gender, age, and socio-economic status. Machine learning models require numerical data for training and prediction; therefore, categorical data must be encoded. One common approach is label encoding, where each category is assigned a unique numerical value (Chakrabortyx& Sundaram, 2020).

Table 1. Score

Dataset	Sample Size	Age Range	Features	Limitations
Dyslexia Data	50	7-16	Brain wave patterns (EEG)	Small sample size, limited age range, limited features
Coimbra Dyslexia Database	289	7-14	EEG, behavioral, neuropsychological measures	Limited age range, limited geographic distribution
Haskins Dyslexia Corpus	45	7-18	Behavioral and brain imaging measures	Limited sample size, limited features, limited geographic distribution
Dyslexia EEG Dataset	19	10-14	EEG	Extremely small sample size, limited age range, limited features
Dunedin Study	1,037	Birth to 38	Cognitive, behavioral, and neurological tests	Limited to one geographic location, limited age range, may not have been specifically designed for dyslexia
German longitudinal study	365	5-6 at entry	Behavioral and neuropsychological measures	Limited age range, limited geographic distribution
Large-scale Dyslexia Dataset	3,920	6-21	Behavioral and neuropsychological measures	Limited EEG data, limited geographic distribution

Feature Extraction Feature extraction is a technique used to select relevant features from the raw data to improve the performance of the model. Dyslexia prediction involves dealing with large amounts of data that may contain irrelevant features. Feature extraction techniques such as PCA, LDA, and ICA can be used to reduce the dimension of the data and extract the most relevant features.

Normalization Normalization is a technique used to scale the data to a common range. Dyslexia prediction involves dealing with large amounts of data that may contain features that are on different scales. Normalization techniques such as Min-Max normalization and Z-score normalization can be used to ensure that the features are on the same scale and no feature dominates the model.

Feature Selection Feature selection is a technique used to select the most important features from the data. Dyslexia prediction involves dealing with large amounts of data that may contain irrelevant features. Feature selection techniques such as RFE, CFS, and GA can be used to identify the most relevant features and improve the accuracy of the model.

Data Augmentation Dyslexia prediction involves dealing with class-imbalanced data where there may be more non-dyslexic samples than dyslexic samples. Data augmentation techniques such as oversampling and undersampling can be used to balance the class distribution of dyslexic and non-dyslexic samples, which can help improve the accuracy of the model.

In summary, preprocessing techniques play a crucial role in Dyslexia prediction. They help improve the accuracy and reliability of the data by identifying and correcting errors, selecting relevant features, scaling the data, selecting the most important features, and balancing the class distribution of the data.

2.3 Issues with imbalanced datasets

Dyslexia is a relatively rare condition, and datasets used for dyslexia prediction are often imbalanced, meaning that there are fewer positive (dyslexic) cases than negative (non-dyslexic) cases. Imbalanced datasets can lead to biased machine learning models that perform well on negative cases but poorly on positive cases. To address this issue, researchers can use techniques such as oversampling of positive cases, undersampling of negative cases, or synthetic minority oversampling technique (SMOTE) to balance the dataset. Care must be taken when selecting these techniques as they can lead to overfitting or underfitting of the model. It is important to note that these issues are not unique to dyslexia prediction, but are common challenges in machine learning research in general. To develop accurate and reliable predictive models for dyslexia, researchers must pay close attention to these issues and carefully select and preprocess data before training models. Furthermore, the development of ethical guidelines for the use of predictive models for dyslexia is necessary to ensure that such models are not used in discriminatory or harmful ways (Prabhax& Bhargavi, 2022).

3 Materials and Methods

There have been several machine learning approaches used for dyslexia prediction. Some of the most commonly used approaches are discussed below.

3.1 Logistic Regression

In the context of dyslexia, logistic regression has been employed to analyze various features and identify key predictors. Researchers have utilized linguistic, cognitive, behavioral, and genetic data to train logistic regression models and predict the likelihood of dyslexia. A study conducted by Martin, Kronbichler,xand Richlan (2016) used logistic regression to analyze linguistic features and achieved an accuracy of 85% in predicting dyslexia. Similarly, Plantexet al. (2015) utilized logistic regression to classify behavioral and genetic data, achieving an accuracy of 81% in dyslexia prediction. Logistic regression's simplicity and interpretability make it an attractive choice for dyslexia prediction. It allows researchers to understand the contribution of different features and provides a clear understanding of the relationship between predictors and the likelihood of dyslexia (Tamboer, Vorst,x& Oort, 2014).

3.2 Decision Trees

Decision Trees have been employed as a predictive tool for dyslexia. A decision tree is a flowchart-like structure where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the outcome or class label. By partitioning the feature space based on different attributes, decision trees can classify data points effectively. Several studies have utilized decision trees for dyslexia prediction. For example, a study conducted by Prabha, Bhargavi,xand Ragala (2019) employed decision trees to analyze behavioral and cognitive data of dyslexic and non-dyslexic individuals and achieved an accuracy of 79%. Additionally, a study by Vanithaxand Kasthuri (2021) utilized decision trees to classify genetic and environmental data of dyslexic and non-dyslexic adults and achieved an accuracy of 83%.

3.3 Random Forest

Random Forest is an ensemble learning algorithm that uses multiple decision trees to classify data. It has been used in dyslexia prediction by classifying fMRI data of dyslexic and non-dyslexic individuals (Prabhaxet al., 2019)

3.4 Support Vector Machines (SVM)

Support Vector Machines (SVM) have proven to be highly efficient in predicting dyslexia with remarkable accuracy. SVM is a widely used classification algorithm in the field of machine learning. It operates by finding an optimal hyperplane that effectively separates different classes in the data. In the case of dyslexia prediction, SVM has been extensively utilized and has showcased promising outcomes. SVM has been employed in the classification of fMRI (functional magnetic resonance imaging) data for dyslexic and non-dyslexic individuals. A study conducted by Martinxet al. (2016) focused on using SVM to classify fMRI data from dyslexic and non-dyslexic children. They achieved an accuracy of 87.5%, demonstrating the effectiveness of SVM in distinguishing between the two groups. Similarly, Plantexet al. (2015) employed SVM to classify fMRI data of dyslexic and non-dyslexic adults and obtained an accuracy of 80%, further emphasizing the utility of SVM in dyslexia prediction.

3.5 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is another classification algorithm that has been explored for dyslexia prediction. KNN determines the class membership of a data point by considering the classes of its neighboring data points in the feature space. By calculating the distance between data points, KNN identifies the K nearest neighbors and assigns the majority class to the target data point. In the context of dyslexia prediction, KNN has shown promising results. A study conducted by (Kaisar, 2020) utilized KNN to classify linguistic and behavioral features of dyslexic and non-dyslexic individuals and achieved an accuracy of 82% . Similarly, a study by Thompson et al. (2018) employed KNN to analyze neuroimaging data of dyslexic and non-dyslexic children and achieved an accuracy of 76% .

3.6 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) have been employed in predicting dyslexia with remarkable accuracy. ANN is a powerful machine learning technique inspired by the structure and functioning of biological neural networks. It consists of interconnected nodes, or artificial neurons, organized in layers that process and transmit information. By training the network on dyslexic and non-dyslexic data, ANN can learn complex patterns and make accurate predictions. Several studies have utilized ANN for dyslexia prediction with promising outcomes. For example, a study conducted by Martinxet al. (2016) utilized ANN to analyze linguistic and cognitive features of dyslexic and non-dyslexic individuals and achieved an accuracy of 91%. Another study by Plantexet al. (2015) employed ANN to classify behavioral and genetic data of dyslexic and non-dyslexic children and achieved an accuracy of 85%.

3.7 Convolutional Neural Networks (CNN)

In dyslexia research, Convolutional Neural Networks (CNNs) have been applied to classify brain scans, such as MRI or fMRI, of dyslexic and non-dyslexic individuals. By utilizing convolutional layers to detect local patterns and pooling layers to aggregate information, CNNs can automatically learn discriminative features that differentiate between the two groups. A study conducted by Zahia, Garcia-Zapirain, Saralegui,xand Fernandez-Ruanova (2020) utilized CNNs to classify brain activation patterns from fMRI data, achieving an accuracy of 88% in distinguishing dyslexic and non-dyslexic individuals . Similarly, a study by Alqahtani, Alzahrani,xand Ramzan (2023)) employed CNNs to analyze structural brain data, obtaining an accuracy of 82% in dyslexia prediction. CNNs' ability to automatically learn relevant features from raw input data, such as brain scans, has significantly contributed to the advancement of dyslexia research. Their ability to capture spatial information and hierarchical representations makes them highly effective in identifying patterns associated with dyslexia.

Overall, the choice of machine learning approach depends on the type of data available and the research question being addressed need to carefully consider the trade-offs between accuracy and interpretability when selecting a machine learning approach for dyslexia prediction.

4 Case Studies and Experiments Proposed by Researchers

Asvestopoulouxet al. (2019) present a screening tool for dyslexia based on machine learning techniques. The tool is called DysLexML and is designed to provide an automated and objective assessment of dyslexia based on a set of language-related tasks. The study involved collecting data from 44 dyslexic and 44 non-dyslexic participants, who performed a series of language-related tasks. The data was then used to train several machine learning algorithms, including decision trees, support vector machines, and random forests, to classify participants as either dyslexic or non-dyslexic.The results showed that DysLexML achieved an accuracy of 89.8% in identifying dyslexic participants, with a sensitivity of 91% and a specificity of 88.6%. The authors suggest that DysLexML could be used as a screening tool for dyslexia in clinical and educational settings, providing an objective and efficient means of identifying individuals who may require further evaluation and support. Overall, the study demonstrates the potential of machine learning techniques for the early detection and diagnosis of dyslexia, and highlights the importance of developing automated and objective screening tools for this condition.

Vajs, Kovic, Papic, Savic,xand Jankovic (2022) studied the use of machine learning and eye-tracking measures to detect readers with dyslexia. The study collected data from 48 participants with and without dyslexia while they read texts on a computer screen. Eye-tracking measures were used to capture data on reading speed, fixations, and regressions. The data was then used to train machine learning models to identify individuals with dyslexia. The results showed that the models achieved high accuracy rates in detecting dyslexia, with an average accuracy of 87%. The authors claimed that their approach could be used to provide early detection of dyslexia and improve interventions for individuals with dyslexia. They also suggested that their method could be used to develop personalized reading interventions for individuals with dyslexia based on their specific reading patterns.

The work by Relloxet al. (2018) proposes a new method for screening dyslexia in English using human-computer interaction (HCI) measures and machine learning. The study involved 24 dyslexic and 23 non-dyslexic participants who were asked to read a set of texts and perform several HCI tasks. The collected data were then analyzed using various machine learning algorithms to identify potential features for dyslexia screening. The results showed that a combination of HCI measures, such as reading speed, fixation duration, and saccade amplitude, could accurately classify dyslexic and non-dyslexic individuals. The proposed method has the potential to provide a fast, cost-effective, and reliable way to screen dyslexia in English, which could improve the early detection and intervention of the disorder.

The work by Relloxet al. (2016) presents a screening tool for dyslexia called Dytective that uses a game-based approach to assess reading skills. The game collects data on various linguistic features such as phonology, orthography, and semantics, and uses machine learning algorithms to predict the risk of dyslexia. The study suggests that the game-based approach is engaging and effective in identifying individuals at risk of dyslexia, with a reported accuracy of 90

The work by Khan, Cheng,xand Bee (2018) proposes a diagnostic and classification system (DCS) for identifying dyslexia in children using machine learning techniques. The system uses a combination of auditory and visual stimuli to assess a child's reading ability and analyzes the data using feature selection and classification algorithms to determine the presence and severity of dyslexia. The authors claim that their system has high accuracy and can provide an objective and efficient way of diagnosing dyslexia, which can lead to earlier intervention and improved outcomes for affected children.

The paper by Chakrabortyxand Sundaram (2020) presents a machine learning algorithm for predicting dyslexia using eye movement data. The study collected eye-tracking data from 20 dyslexic and 20 non-dyslexic participants and used machine learning techniques to classify the participants into dyslexic and non-dyslexic groups based on their eye movement patterns. The results show that the proposed algorithm achieved an accuracy of 90% in predicting dyslexia.

The paper by Kariyawasam, Nadeeshani, Hamid, Subasinghe,xand Ratnayake (2019) proposes a gamified approach for screening and intervention of dyslexia, dysgraphia, and dyscalculia. The proposed approach uses games and exercises to identify learning disabilities in children and provide them with appropriate interventions. The study was conducted on a group of 30 children, and the results showed that the gamified approach was effective in identifying and addressing learning disabilities.

The paper by MMTxand Sangamithra (2019) proposes an intelligent system for predicting learning disabilities in school-going children using fuzzy logic and K-means clustering in machine learning. The study collected data from 100 students and used fuzzy logic and K-means clustering to classify the students into normal and learning-disabled groups. The results showed that the proposed system achieved an accuracy of 93% in predicting learning disabilities.

The paper by Jothi Prabhaxand Bhargavi (2019) presents a predictive model for dyslexia using eye fixation events. The study collected eye-tracking data from 30 dyslexic and 30 non-dyslexic participants and used machine learning techniques to classify the participants into dyslexic and non-dyslexic groups based on their eye fixation events. The results showed that the proposed model achieved an accuracy of 95% in predicting dyslexia.

5 Evaluation Metrics

Many evaluation metrics are used to measure the performance of dyslexia prediction models. The commonly used evaluation metrics for classification models are accuracy, precision, recall, F1 score, area under the receiver operating characteristic curve (AUC-ROC), and confusion matrix (Fawcett, 2006; Powers, 2020; Saitox& Rehmsmeier, 2015)].

- Accuracy: It is the proportion of correct predictions out of the total predictions made by the model. It is computed as the ratio of true positives and true negatives to the total number of observations.
- Precision: It is the proportion of true positive predictions out of the total positive predictions made by the model. It is computed as the ratio of true positives to the sum of true positives and false positives.
- Recall: It is the proportion of true positive predictions out of the total actual positive observations in the data. It is computed as the ratio of true positives to the sum of true positives and false negatives.
- F1 score: It is the harmonic mean of precision and recall, which balances both the measures. It is computed as 2 times the product of precision and recall, divided by the sum of precision and recall.
- AUC-ROC: It is a performance metric that measures the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity) at different classification thresholds. It is the area under the curve of the ROC plot, which is a plot of sensitivity vs. 1-specificity at different threshold values.
- Confusion matrix: It is a table that summarizes the performance of a classification model by showing the number of true positives, true negatives, false positives, and false negatives.

These evaluation metrics are important to assess the performance of dyslexia prediction models and to compare the performance of different models. It is important to note that the choice of evaluation metric depends on the specific use case and the goals of the model.

6 Limitations and Challenges

Although machine learning techniques have shown great promise in predicting dyslexia disease, there are some limitations and challenges that must be considered.

Data availability. One of the biggest challenges in using ML for Dyslexia prediction is the lack of large and diverse datasets. Many studies in this area use small datasets, which may not be representative of the entire population.

Generalization. Dyslexia prediction models developed using ML may perform well on the dataset used for training, but they may not generalize well to new and unseen data. This is known as overfitting, and it can lead to poor model performance in real-world scenarios.

Complexity of algorithms. Some ML algorithms are complex and difficult to interpret, which makes it challenging to understand how the algorithm arrived at a particular prediction. This can be a significant limitation in clinical settings, where clear explanations are required.

Class imbalance. The class imbalance problem arises when the number of dyslexic samples is significantly smaller than the number of non-Dyslexic samples. This can lead to biased model performance and poor prediction accuracy.

Feature selection. The selection of relevant features is crucial for developing accurate dyslexia prediction models. However, identifying the most important features can be challenging and may require expert knowledge of the disease.

Preprocessing. The selection of appropriate preprocessing techniques, such as data cleaning, normalization, and feature extraction, can impact the performance of the ML model.

Ethical concerns. There are ethical concerns related to the use of ML in predicting dyslexia disease. For example, there is a risk that the predictions may be used to stigmatize individuals or limit their opportunities. See also the section below.

7 Ethical Considerations

Ethical considerations should be emphasized regarding the use of sensitive personal data and potential stigmatization. In particular, the use of eye-tracking measures and other behavioral data for dyslexia prediction raises privacy concerns. Researchers must ensure that they have obtained informed consent from participants and protect their privacy by using secure data storage and appropriate data sharing policies. Additionally, the use of machine learning algorithms in dyslexia prediction can lead to potential biases, especially if the training data is biased. Therefore, researchers must take steps to ensure that their models are unbiased and do not perpetuate existing biases or stereotypes. Moreover, dyslexia prediction using machine learning should not be used as a basis for exclusion or discrimination against individuals with dyslexia. It is crucial to ensure that the results of dyslexia prediction are used only to support early intervention and support for individuals with dyslexia and not for labeling or stigmatizing them (Chakrabortyx& Sundaram, 2020; Kariyawasamxet al., 2019; Relloxet al., 2016).

8 Future Directions and Potential Areas for Improvement

Dyslexia prediction using machine learning is a promising area of research with potential for significant impact on early identification and intervention for children with dyslexia. However, there are several areas for improvement and future directions that researchers can focus on.

Larger datasets. One of the main challenges in dyslexia prediction using machine learning is the availability of large and diverse datasets. Future research should focus on collecting and sharing larger datasets that include data from different populations, languages, and cultures.

Better feature engineering. Feature engineering is the process of selecting and extracting relevant features from data that can be used for machine learning. Future research should focus on developing better feature engineering methods that can capture more relevant features from data, including features related to cognitive processes and linguistic features.

Model interpretability. Machine learning models used for dyslexia prediction should be interpretable, meaning that it should be possible to understand how the model arrived at its prediction. This is important for clinicians and educators who need to make decisions based on the model's output. Future research should focus on developing machine learning models that are more interpretable and transparent.

Validation and replication. Dyslexia prediction models should be validated on independent datasets to ensure that they are robust and generalizable. Future research should focus on replicating existing models on independent datasets and comparing their performance to identify the most effective models.

Integration with clinical practice. Dyslexia prediction models should be integrated with clinical practice to ensure that they are useful in real-world settings. Future research should focus on developing user-friendly interfaces for dyslexia prediction models and testing their effectiveness in clinical practice.

Overall, dyslexia prediction using machine learning has the potential to make a significant impact on early identification and intervention for children with dyslexia. By addressing the above areas for improvement, researchers can develop more accurate, reliable, and clinically useful dyslexia prediction models.

9 Conclusion

The field of predicting dyslexia with machine learning is rapidly evolving with advancements in feature selection, algorithm development, and evaluation metrics. Through our comprehensive review of the existing literature, we have provided an overview of the state-of-the-art techniques and highlighted their strengths and weaknesses. We found that a combination of behavioral and neuroimaging data is essential for accurate dyslexia prediction. In addition, the use of advanced algorithms such as deep learning has shown promising results. However, there are still some challenges that need to be addressed, such as small sample sizes and the need for validation in diverse populations. We recommend that future research focuses on addressing these challenges and developing more robust models that can be applied in clinical settings. Overall, the use of machine learning for dyslexia prediction has the potential to greatly improve early identification and intervention, leading to better outcomes for individuals with dyslexia.

References

Ahmad, N., Rehman, M. B., Hassan, H. M. E., Ahmad, I., & Rashid, M. (2022, jul). An efficient machine learning-based feature optimization model for the detection of dyslexia. *Computational Intelligence and Neuroscience*, *2022*, 1–7. doi: https://doi.org/10.1155/2022/8491753

Alqahtani, N. D., Alzahrani, B., & Ramzan, M. S. (2023). Deep learning applications for dyslexia prediction. *Applied Sciences*, *13*(5), 2804. doi: https://doi.org/10.3390/app13052804

Asvestopoulou, T., Manousaki, V., Psistakis, A., Smyrnakis, I., Andreadakis, V., Aslanides, I. M., & Papadopouli, M. (2019). *Dyslexml: Screening tool for dyslexia using machine learning.* arXiv. doi: https://doi.org/10.48550/arXiv.1903.06274

Chakraborty, V., & Sundaram, M. (2020). Machine learning algorithms for prediction of dyslexia using eye movement. In *Journal of physics: Conference series* (Vol. 1427, p. 012029). doi: https://doi.org/10.1088/1742-6596/1427/1/012012

Chakraborty, V., Vani, & Sundaram, M. (2021). An efficient smote-based model for dyslexia prediction. *International Journal of Information Engineering & Electronic Business*, *13*(6), 13-21. doi: https://doi.org/10.5815/ijieeb.2021.06.02

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, *27*(8), 861–874. doi: https://doi.org/10.1016/j.patrec.2005.10.010

Jothi Prabha, A., & Bhargavi, R. (2019). Predictive model for dyslexia using machine learning—a research travelogue. In *Proceedings of the third international conference on microelectronics, computing and communication systems: Mccs 2018* (pp. 1–6).

Kaisar, S. (2020). Developmental dyslexia detection using machine learning techniques: A survey. *ICT Express*, *6*(3), 181–184. doi: https://doi.org/10.1016/j.icte.2020.05.006

Kariyawasam, R., Nadeeshani, M., Hamid, T., Subasinghe, I., & Ratnayake, P. (2019, dec). A gamified approach for screening and intervention of dyslexia, dysgraphia and dyscalculia. In *2019 international conference on advancements in computing (icac)* (pp. 1–6). IEEE. doi: https://doi.org/10.1109/icac49085.2019.9103336

Khan, R. U., Cheng, J. L. A., & Bee, O. Y. (2018). Machine learning and dyslexia: Diagnostic and classification system (dcs) for kids with learning disabilities. *International Journal of Engineering & Technology*, *7*(3.18), 97–100.

Martin, A., Kronbichler, M., & Richlan, F. (2016). Dyslexic brain activation abnormalities in deep and shallow orthographies: A meta-analysis of 28 functional neuroimaging studies. *Human brain mapping*, *37*(7), 2676–2699. doi: https://doi.org/10.1002/hbm.23202

MMT, M. H., & Sangamithra, A. (2019). Intelligent predicting learning disabilities in school going children using fuzzy logic k mean clustering in machine learning. *Int. J. Recent Technol. Eng*, *8*(4), 1694–1698. doi: https://doi.org/10.35940/ijrte.c5620.118419

Plante, E., Patterson, D., Gomez, R., Almryde, K. R., White, M. G., & Asbjørnsen, A. E. (2015). The nature of the language input affects brain activation during learning from a natural language. *Journal of Neurolinguistics*, *36*, 17–34. doi: https://doi.org/10.1016/j.jneuroling.2015.02.002

Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*. doi: https://doi.org/10.48550/arXiv.2010.16061

Prabha, A. J., & Bhargavi, R. (2022). Prediction of dyslexia from eye movements using machine learning. *IETE Journal of Research*, *68*(2), 814–823. doi: https://doi.org/10.1080/03772063.2019.1622461

Prabha, A. J., Bhargavi, R., & Ragala, R. (2019). Predictive model for dyslexia from eye fixation events. *International Journal of Engineering and Advanced Technology (IJEAT)*, *9*, 235–240. doi: https://doi.org/10.35940/ijeat.a1045.1291s319

Rello, L., & Ballesteros, M. (2015). Detecting readers with dyslexia using machine learning with eye tracking measures. In *Proceedings of the 12th international web for all conference.* doi: https://doi.org/10.1145/2745555.2746644

Rello, L., Ballesteros, M., Ali, A., Serra, M., Sanchez, D. A., & Bigham, J. P. (2016). Dytective: diagnosing risk of dyslexia with a game. In *Pervasivehealth.* ACM. doi: https://doi.org/10.4108/eai.16-5-2016.2263338

Rello, L., Romero, E., Rauschenberger, M., Ali, A., Williams, K., Bigham, J. P., & White, N. C. (2018, apr). Screening dyslexia for english using hci measures and machine learning. In *Proceedings of the 2018 international conference on digital health.* ACM. doi: https://doi.org/10.1145/3194658.3194675

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432. doi: https://doi.org/10.1371/journal.pone.0118432

Tamboer, P., Vorst, H. C., & Oort, F. J. (2014). Identifying dyslexia in adults: an iterative method using the predictive value of item scores and self-report questions. *Annals of dyslexia*, *64*, 34–56. doi: https://doi.org/10.1007/s11881-013-0085-9

Vajs, I., Kovic, V., Papic, T., Savic, A. M., & Jankovic, M. M. (2022, aug). Dyslexia detection in children using eye tracking data based on vgg16 network. In *2022 30th european signal processing conference (eusipco).*

IEEE. doi: https://doi.org/10.23919/eusipco55093.2022.9909817

Vanitha, G., & Kasthuri, M. (2021). Dyslexia prediction using machine learning algorithms–a review. *International Journal of Aquatic Science*, *12*(2), 3372–3380.

Zahia, S., Garcia-Zapirain, B., Saralegui, I., & Fernandez-Ruanova, B. (2020, dec). Dyslexia detection using 3d convolutional neural networks and functional magnetic resonance imaging. *Computer methods and programs in biomedicine*, *197*, 105726. doi: https://doi.org/10.1016/j.cmpb.2020.105726

Journal of Behavioral Data Science, 2023, *3* (1), 84–107.
DOI: `https://doi.org/10.35566/jbds/v3n1/mccure`

Bayesian IRT in JAGS: A Tutorial

Kenneth McClure[1]

Department of Psychology, University of Notre Dame, Notre Dame, USA
`kmcclur5@nd.edu`

Abstract. Item response modeling is common throughout psychology and education in assessments of intelligence, psychopathology, and ability. The current paper provides a tutorial on estimating the two-parameter logistic and graded response models in a Bayesian framework as well as provide an introduction on evaluating convergence and model fit in this framework. Example data are drawn from depression items in the 2017 Wave of the National Longitudinal Survey of Youth and example code is provided for `JAGS` and implemented through `R` using the `runjags` package. The aim of this paper is to provide readers with the necessary information to conduct Bayesian IRT in `JAGS`.

Keywords: Logistic Response Model · Item Response Theory · Bayesian Method · JAGS Tutorial

1 Introduction

Item response theory (IRT) is a psychometric framework for modeling relationships between observed responses, often in the form of test or survey data, and latent abilities or traits (Birnbaum, 1968; Embretson & Reise, 2000). IRT models consist of two sets of parameters namely ability parameters θ_i, $i = 1, 2, ...N$, and item parameters $\boldsymbol{\omega_j}$, $j = 1, 2, ...J$ where i indexes the number of respondents and j indexes the test items. Thus, the sample size is N and test length is J. IRT models are natural fit for Bayesian estimation (Baker & Kim, 2004; Fox, 2010; Lord, 1986; Patz & Junker, 1999) and provide a natural way to obtain ability and item parameter estimates simultaneously.

While ability may be multidimensional or non-normally distributed (Reckase, 2009), it is assumed that θ_i is unidimensional and

$$\theta_i \sim N(0, 1) \tag{1}$$

in this tutorial, for simplicity, as is common in practice. Other common assumptions for IRT models include local independence of responses

$$P(\boldsymbol{x_i}|\theta_i) = \prod_{j=1}^{J} p(x_{ij}|\theta_i) \tag{2}$$

where the probability of observing a given response pattern $\boldsymbol{x_i}$ is given by $\boldsymbol{x_i} = (X_{i1} = x_{i1}, ..., X_{iJ} = x_{ij})$ and monotonicity of the latent trait

$$\theta_1 > \theta_2 \rightarrow p(x = 1|\theta_1) \geq p(x = 1|\theta_2). \tag{3}$$

Monotonicity implies that higher values of the latent trait increase the probability of endorsing the item. Thus, item scores are typically coded such that all inter-item correlations are nonnegative.

While the distribution of θ is often informed by theory underlying constructs of interest or computational convenience, the nature of $\boldsymbol{\omega}$ depends on item characteristics (e.g., number of response options) and the specified item response model. A common model for binary data is the logistic model (Birnbaum, 1968) which includes a family of models spanning from a single item parameter to four item parameters (Barton & Lord, 1981). These item parameters can accommodate item difficulty, discrimination, and response asymptotes (e.g., guessing). In addition to binary data, many psychological measures contain ordered categorical response options (e.g., Likert-type scales). Polytomous IRT models, such as the graded response model (GRM), are better suited for these instruments (Samejima, 1969). This paper focuses on the two-parameter logistic (2PL) and GRM for binary and ordered categorical responses respectively.

The organization of the rest of the paper is as follows: first, the 2PL and GRM are detailed along with a discussion on priors for item parameters. Then, demonstrations of the 2PL and the GRM in `JAGS` using the package `runjags` (Denwood, 2016) are provided using data from the National Longitudinal Survey of Youth (Bureau of Labor Statistics, 2017). Convergence analysis, model fit, and item curves are also demonstrated. We conclude with a brief discussion.

2 Two-Parameter Logistic Model

The 2PL model is given by

$$P_{ij}(x_j = 1|\theta_i, \boldsymbol{\omega}_j) = \frac{\exp(D\alpha_j\theta_i - \beta_K)}{1 + \exp(D\alpha_j(\theta_i - \beta_j))} \tag{4}$$

or alternatively

$$= \frac{1}{1 + \exp(-D\alpha_j(\theta_i - \beta_j))} \tag{5}$$

where $\boldsymbol{\omega}_j = \{\alpha_j, \beta_j\}$. Here α_j is the discrimination parameter, β_j is the difficulty parameter and D is a scaling constant. The 2PL can also be written as

$$logit(P_{ij}) = D\alpha_j(\theta_i - \beta_j). \tag{6}$$

Since scores can be coded to ensure positive inter-item correlation, which is necessary to preserve the assumption of monotonicity, αs are constrained greater than 0 and are typically between 0.5 - 2 in practice. The Rasch model can be obtained by constraining $\alpha_j = 1$ (i.e., all items are equally discriminant). No strict constraints are necessary to impose on β, however, values of β should overlap with the distribution of θ in practice to ensure sufficient variability in item responses. D is a scaling factor and setting $D = 1.702$ produces essentially the same scaling as the normal ogive model (Camilli, 1994).

2.1 α_j Priors

Priors for the discrimination parameter α_j must accommodate the constraint that $\alpha_j > 0$. Common choices include the truncated normal (i.e., N_+, Curtis (2010)) and the lognormal (Patz & Junker, 1999) distributions. We use the truncated normal distribution in the demonstration of the 2PL

$$\alpha_j \sim N_+(\mu_{\alpha_j}, \sigma^2_{\alpha_j}). \tag{7}$$

Researchers wishing to use a log-normal prior for α_j should note that that both μ_{α_j} and $\sigma^2_{\alpha_j}$ impact the mean and variance of the log-normal distribution making prior specification challenging (Curtis, 2010). We fix $\phi_{\alpha_j} = 1/\sigma^2_{\alpha_j} = .00001$ and draw $\mu_{\alpha_j} \sim U[0.5, 2]$ in the demonstration below.

2.2 β_j Priors

The difficulty parameter prior can be specified as a normal distribution

$$\beta_j \sim N(\mu_{\beta_j}, \sigma^2_{\beta_j}) \tag{8}$$

allowing the mean (μ_{β_j}) and variance ($\sigma^2_{\beta_j}$) to vary across items. These parameters can be fixed or treated as hyper-parameters drawn from hyper-priors. For demonstration, we draw $\mu_{\beta_j} \sim U[-2, 2]$ but fix $\sigma^2_{\beta_j}$ in the example. We fix $\sigma^2_{\beta_j} = 10^6$ by fixing the precision of the difficulty parameters $\phi_{\beta_j} = .000001$. Precision is commonly used in Bayesian analysis and is the inverse of the variance (i.e., $\phi = 1/\sigma^2_{\beta_j}$).

2.3 Bayesian 2PL in JAGS

Multiple software programs for Bayesian analysis are openly available (Lunn, Spiegelhalter, Thomas, & Best, 2009; Plummer, 2003; Stan Development Team, 2023). This paper focuses on `JAGS` implemented in `R` (R Team Core, 2022) via the `runjags` package (Denwood, 2016). Alternative packages for running `JAGS` through `R` are also available (Plummer, 2022). Specifying models in `JAGS` consists of three primary components: 1) model specification, 2) initial values, and 3) data. Once all of the components have been compiled, the `runjags` function can conduct Markov Chain Monte Carlo (MCMC) sampling.

Example Data Data for this tutorial consists of 4 items assessing depression in the 2017 wave of the NLSY. For the demonstration of the logistic model, a dichotomized version of the depression items are examined where responses of 1 are recoded as 0 and responses larger than 1 are recoded as a 1. Note that we do not advocate dichotomozing polytomous responses in practice and do this only for pedagogical purposes. Data are provided in the supplementary material.

2PL Model Specification First we specify `twoPL` as the 2PL model to run in `JAGS`. In the code, `i` indexes the `N` respondents and `j` indexes the `J` items. The item response for person `i` on item `j` is represented as $X[i, j]$ and are drawn from a Bernoulli distribution based on a probability determined by the underlying 2PL.

```
twoPL<- "
model{
  for (i in 1:N){
    for (j in 1:J){
      X[i, j] ~ dbern(p[i,j])
      logit(p[i,j]) <- D*alpha[j]*(theta[i] - beta[j]) #2PL
    }
    theta[i] ~ dnorm(0, 1)
  }
  #Priors for model parameters
  for (j in 1:J){
    beta[j] ~ dnorm(mu.beta[j], pre.beta)
    alpha[j] ~ dnorm(mu.alpha[j],pre.alpha)T(0,)
  }
  #Hyper Prior for mu.beta and mu.alpha
  for(j in 1:J){
    mu.beta[j] ~ dunif(-1,1)
    mu.alpha[j] ~ dunif(.75,1)
  }

  for(i in 1:N){
    for(j in 1:J){
      X.rep[i,j] ~ dbern(p[i,j]) #Model implied data
    }
  }
  for(j in 1:J){
    ppp[j] <-step(sum(X.rep[,j])-sum(X[,j])) # ppp for item fit
  }
  D=1.702 #scaling constant
}
"
```

Here θ_i is assumed to follow a standard normal distribution. A normal prior is chosen for difficulty parameters $\beta_j \sim N(\mu_{\beta_j}, \phi_{\beta_j})$, where ϕ is the precision. We draw $\mu_{\beta_j} \sim U[-2,2]$ and choose $\phi_\beta = .000001$. For the discrimination parameter, a truncated normal (i.e., N_+) distribution is chosen $\alpha_j \sim N_+(\mu_{\alpha_j}, \phi_{\alpha_j})$ with $\mu_{alpha} \sim U[.75,1]$ and $\phi_{alpha} = .000001$. This prior ensures that α_j are non-negative. In `JAGS`, truncation of the normal distribution below at zero is specified using `T(0,)`. In addition to ability and item parameters, `X.rep` and `ppp` (i.e., posterior predictive p-values) are specified to obtain posterior predictive checks. `X.rep` are draws from the implied model to be used in posterior predictive checks via `ppp` (Gelman, Meng, & Stern, 1996). `step(x)` is a function which return 1 if $x \geq 0$ and 0 otherwise.

To calculate the PPP, a new set of data $\boldsymbol{y^m}$ is generated based on parameter estimate θ^m at MCMC iteration m. The statistic of interest (e.g., expectation) is calculated for both this generated posterior predictive distribution and the sample data $\boldsymbol{x}$ using θ^m. The PPP is the proportion of generated statistics that are greater than the statistics of the data. If T is the statistics of interest, the PPP can be defined as

$$PPP = P(T(\boldsymbol{x}) < T(\boldsymbol{y})). \tag{9}$$

PPP values less than 0.10 (i.e., or greater than 0.90) indicate poor fit while models which fit exceptionally well have PPPs near 0.5 (Cain & Zhang, 2019). We choose $T_j = \sum_{i=1}^{N} x_{ij}$ for the 2PL and obtain a PPP for each item.

2PL Initial Values In addition to model specification, it is also necessary to specify initial values for item parameters. When selecting initial parameters, it is crucial to select values of α and β which are valid for the model (i.e., $\alpha > 0$). To specify initial values in `JAGS` named lists are given for each desired chain. For multiple chains, a list of named lists is used. Below, initial values for 2 chains are specified. Note that for certain convergence metrics, such as the potential scale reduction factor (psrf), multiple chains are needed (Gelman and Rubin (1992)). Additionally, seeds for the Markov chains (i.e., `.RNG.seed`), as well as the random number generation method (i.e., `.RNG.name`), can be supplied in the initial values object to make the chains reproducible. `JAGS` possesses a number of random number generators, we use the Mersenne-Twister method.

```
inits.2PL <- list(list(beta=rep(-.25, ncol(dep2017.binary)),
                       alpha=rep(.25, ncol(dep2017.binary)),
                 .RNG.seed=1, .RNG.name="base::Mersenne-Twister"),
                   list(beta=rep(.25, ncol(dep2017.binary)),
                       alpha=rep(.5, ncol(dep2017.binary)),
                 .RNG.seed=2, .RNG.name="base::Mersenne-Twister"))
```

2PL Model Data It is also necessary to specify data for `JAGS` in the form of a named list. This data file includes the item response data as well as other necessary constant values for the model script such as `N` and `J`. In the model data list, additional information about the hyperparameters (e.g., precision of α and β) or model constants (e.g., D) can be provided if they are not explicitly defined in the model specification. For demonstration, hyperparameter precision is provided as data and the scaling constant D is defined in the model specification.

```
data.2PL <- list(N=nrow(dep2017.binary), J=ncol(dep2017.binary),
                 X=dep2017.binary, pre.alpha=1E-6, pre.beta=1E-6)
```

Monte Carlo Sampling The `run.jags` function can be used to translate the model, initial values, and data into `JAGS` and conduct Gibbs sampling. This function also allows users to specify which model parameters should be monitored for convergence using the `monitor` argument. In addition to parameters of interest, we are also able to specify other values, such as posterior predictive p-values (PPP), or log-likelihood values to be returned in our output. Users are also able to specify the burnin and chain length using the `burnin` and `sample` arguments respectively. Below we specify a `burnin` period of 1000 samples and a chain length of 3000 samples. For readers new to Bayesian analysis, “burnin” samples are thought to not be sampled prior to Markov Chains to reaching stationarity and are discarded from analysis. `JAGS` also allows for multiple sampling methods for MCMC via the `method` argument. We use the `parallel` method which conducts MCMC sampling for each chain simultaneously on separate cores. The code below conducts sampling in `JAGS` and returns Markov chains for θ_i, α_j, β_j, and ppp_j. Convergence is evaluated and discussed in a later section.

```
out.2PL <- run.jags(twoPL,monitor=c("theta","beta","alpha","ppp"),
                    data=data.2PL, n.chains=2, method="parallel",
                    inits=inits.2PL,adapt=500, burnin=1000,
                    sample=3000)
```

3 Graded Response Model

The GRM (Samejima, 1969) is appropriate for items with ordered categorical responses $(1, ..., K_j)$. Note that the number of item response options is allowed to vary by item. It is assumed, however, that response categories are monotonically increasing in difficulty/severity. Then the cumulative probability P_{ijk} of endorsing up to category k is

$$P_{ijk} = P(X_{ij} \leq k|\theta_i) \tag{10}$$

and the probability p_{ijk} of endorsing category k is given by

$$p_{ijk} = P_{ijk} - P_{ijk-1}, \quad k = 2, ..., K_j \tag{11}$$

with $p_{ij1} = P_{ij1}$ and $P_{ijK_j} = 1$. Thus, there are $K_j - 1$ boundaries between response categories governed by item thresholds $\kappa_1 < ... < \kappa_{K_j-1}$. Given this,

P_{ijk} can be written as

$$P_{ijk}(x_{ij} \leq k|\theta_i, \boldsymbol{\omega}_j) = \frac{1}{1 + \exp(\kappa_{jk} - \alpha_j \theta_i)} \tag{12}$$

Readers will note that (11) is the cumulative distribution of the logistic function similar to the 2PL.

3.1 α_j Priors

Priors for α_j can be obtained using the same methods as the 2PL. Again we use the truncated normal distributions

$$\alpha_j \sim N_+(\mu_{\alpha_j}, \phi_{\alpha_j}). \tag{13}$$

3.2 κ_j Priors

Distributions for κ_j need to accommodate the ordering constraint $\kappa_1 < ... < \kappa_{K_j-1}$ but otherwise can be conceptualized similar to the β_j parameters in the 2PL. To account for ordering, we recommend using unconstrained auxiliary parameters $\kappa^*_{j1}, ..., \kappa^*_{jK_j-1}$ following Curtis (2010). These auxiliary parameters can be drawn from

$$\kappa^*_{jk} \sim N(\mu_\kappa, \sigma^2_\kappa) \tag{14}$$

and sorted in increasing order. Following this rank ordering, κ_{jk} is assigned the kth ordered κ^*_{jk}.

3.3 GRM in JAGS

For the GRM, the original Likert-type depression items are analyzed. Responses on the original measure ranged from 1 to 5; however, not all categories were endorsed on each item. Item 1 only has responses in categories $k = 1, 2, 3, 5$ but items 2-4 have responses to all five categories. While the GRM can easily accommodate different K_j, it is necessary for these categories to be adjacent and start at 1 in `JAGS`. Thus, responses of 5 on item 1 are recoded as 4.

3.4 GRM Specification

The GRM can be specified in `JAGS` in multiple ways. The first utilizes the categorical distribution for polytomous responses and auxiliary parameters κ^* to obtain item threshold parameters κ (Curtis, 2010). This approach is similar to the 2PL and applies the `logit` function to each $p(x_{i,j} = k|\theta_i, \kappa_{j,k}, \alpha_j)$ to obtain the probability of responding to each response category. This specification, including a demonstration of the truncated normal distribution for the α_j prior is provided below.

```
GRM <- "
model{
for(i in 1:N){
  for(j in 1:J){
    X[i,j] ~ dcat(prob[i,j,1:K[j]]) #categorical distribution
  }
  theta[i]~dnorm(0,1)

  for(j in 1:J){
    for(k in 1:(K[j]-1)){
      logit(P[i,j,k])<- kappa[j,k]-alpha[j]*theta[i]
        #kappa is the threshold
    }
    P[i,j,K[j]]<-1
  }

  for(j in 1:J){
    prob[i,j,1] <- P[i,j,1]
    for(k in 2:K[j]){
      prob[i,j,k] <- P[i,j,k]-P[i,j,k-1]
    }
  }
}
  for(j in 1:J){
    #truncated normal prior
    alpha[j] ~ dnorm(mu.alpha,pre.alpha)T(0,)
  }

  for(j in 1:J){
    for(k in 1:(K[j]-1)){
      #sample auxiliary parameters
      kappa.star[j,k] ~ dnorm(mu.kappa,pre.kappa)
    }
    #Need to sort kappa.star in increasing order
    kappa[j,1:(K[j]-1)] <- sort(kappa.star[j,1:(K[j]-1)])
  }
  pre.alpha = 1E-06 #alpha precision
  pre.kappa = 1E-06 #kappa.star precision
  mu.alpha = 0.5 #alpha mean
  mu.kappa = 0 #kappa.star mean
}
"
```

This specification also requires a dummy coded data matrix of κ when items possess different number of response categories. This matrix is also J by $K-1$

with NA entries where κ_j will be estimated and a dummy value of 0 for entries where no κ_j is to be estimated.

```
K = apply(dep2017,2,max) #nummber of response categories per item
J = ncol(dep2017) #number of items
N = nrow(dep2017) #number of respondents

kappa.dat = matrix(c(NA,NA,NA,0,
                      NA,NA,NA,NA,
                      NA,NA,NA,NA,
                     NA,NA,NA,NA),
                   nrow=J, ncol=(max(K)-1), byrow=T)
```

An alternative specification of the GRM uses the ordered logit distribution from the glm module in JAGS. This allows for direct sampling given a location parameter μ and sequence of $K-1$ response categories. For the GRM, the location parameter is given by

$$\mu_{i,j} = \alpha_j \theta_i. \tag{15}$$

Readers will note that this specification does not require iteration through the K_j-1 response boundaries. For this reason, we recommend this implementation of the GRM and focus on it for the remainder of this paper.

```
GRM2 <-"
model{
for(i in 1:N){
  for(j in 1:J){
    X[i,j]~dordered.logit(mu[i,j],c[j,1:(K[j]-1)])
    mu[i,j] <- alpha[j]*theta[i]
  }
  theta[i]~dnorm(0,1)
}

for(j in 1:J){
  for(k in 1:(K[j]-1)){
    c[j,k]~dnorm(0,.0001) #prior for thresholds/boundary
  }
  alpha[j] ~ dnorm(mu.alpha,pre.alpha) #prior for alpha
}
pre.alpha=1E-6
mu.alpha=0
}
"
```

3.5 GRM Initial Values

Specifying initial values of κ^* requires the specification of a J by $K-1$ matrix of values. Initial values should be monotonically increasing within each row. Further, for items with less than K response categories `NA` should be included as place holder in this matrix. As with the 2PL, initial values for α_j can be provided in a vector of length J. Both this vector and the matrix for κ^* should be entered into a named list.

```
kappa.star.init = matrix(c(0,1,2,NA,
              -1,0,1,2,
              0,1,2,3,
              0,1,2,3),
              nrow=J, ncol=max(K)-1, byrow=T)

inits.grm = list(list(alpha=rep(1,J), c=kappa.star.init,
        .RNG.seed=2, .RNG.name="base::Mersenne-Twister"),
    list(alpha=rep(.5,J), c=kappa.star.init,
        .RNG.seed=3, .RNG.name="base::Mersenne-Twister"))
```

3.6 GRM Model Data

In addition to the data directly used in the model, when K_j differs across items, a matrix for κ (i.e., `kappa.dat` above) is required for the first implementation of the GRM discussed above. This matrix is not required for the ordered logit approach used here.

```
data.grm = list(N=N,K=K,J=J, X=as.matrix(dep2017))
```

3.7 Monte Carlo Sampling

MCMC sampling for the GRM is nearly identical to the 2PL. The `monitor` argument is altered to reflect the new model parameters. The GRM is a more complex model than the 2PL and thus may require more iterations for chains to reach convergence.

```
out.grm2 <- run.jags(GRM2, monitor=c("c","theta","alpha"),
                     data=data.grm2, n.chains=2, method="parallel",
                     inits=inits.grm2, adapt=1000, burnin=10000,
                     sample=300000, modules="glm")
```

4 Convergence Diagnostics

Following MCMC sampling, it is critical to evaluate if the MCMC procedures converged to a stable posterior distribution that well approximates the underlying process of interest. Convergence analyses, both graphical and statistical, are required to justify the use of resulting chains for inferential purposes. In general it is crucial to determine the stability of the Markov Chains (i.e., convergence to stationarity), the sensitivity of the results to starting values, and the dependence of Monte Carlo samples (i.e., auto-correlation). Below we examine the convergence of the 2PL results obtained above using the `coda` package (Plummer, Best, Cowles, & Vines, 2006). The same process can be applied to the GRM.

Convergence can be assessed graphically using trace plots and numerically via diagnostic statistics. Multiple diagnostic statistics are available in the `coda` package including the Geweke Statistic (Geweke, 1992), the Heidelberger and Welch Test (Heidelberger & Welch, 1983), the Raftery and Lewis test (Raftery & Lewis, 1992), and the psrf (Gelman & Rubin, 1992). A review of diagnostic statistics is beyond the scope of this paper and readers are referred to Roy (2020). Below we demonstrate how to examine convergence on a subset of the model parameters; in practice, all parameters for the analysis of interest should be assessed for convergence prior to interpretation and inferential testing.

4.1 Graphical Methods for Convergence

Multiple plots are helpful in evaluating convergence of posterior distributions. The `plot` function, when applied to an output from the `run.jags` function will automatically produce four plots for each paramter monitored during sampling. The plots include the 1) trace plot (i.e. history plot), 2) empirical CDF of the parameter, 3) empirical pdf of the parameter (i.e., historgram), and 4) auto-correlation plot of MCMC samples. Trace plots depict sampled parameter values across the MCMC samples and are useful in cursory evaluation of chain mixing and convergence. Visual evidence of chain convergence is provided when chains appear to stabilize around a single parameter value. Mixed chains demonstrate significant overlap in the trace of each chain. The auto-correlation plot provides insight into the mixing speed of the chains; chains which quickly mix demonstrate small auto-correlation while slower mixing chains possess higher auto-correlation. Empirical cdf and pdf plots allow for direct examination of the posterior distributions itself and allow researchers to check whether posteriors are of the intended form.

Example Plots Below plots are provided for a single ability θ_1 (Figure 1) and difficulty parameter β_3 (Figure 2). By default, the `plot` function will attempt to plot all monitored parameters. To ensure brevity, we specify parameters to plot using the `var` argument. A brief discussion of each parameter plot is provided below.

The trace plot for θ_1 is provided in the upper left pane with different colors representing different Markov chains. We see that the chains are largely overlapping and appear to oscillate around a value of $\theta_1 = 1$ suggesting that the chains have converged to a stationarity posterior distribution. The bottom right pane depicts the auto-correlation plot which shows fast mixing of the two chains. The empirical cdf and pdf of θ_1, in the top right and bottom left panes respectively,

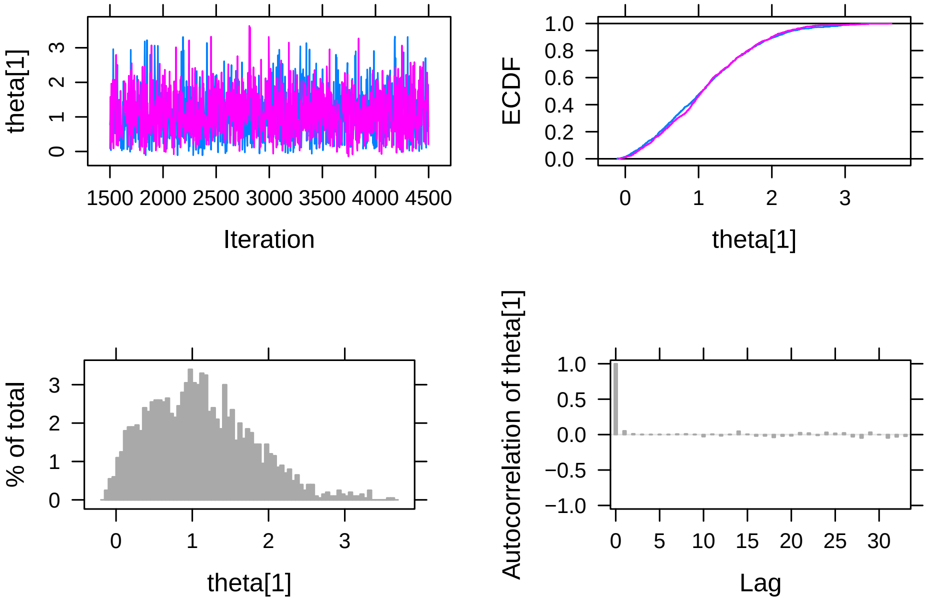

Figure 1. Graphical Convergence Plots for a Single Ability Parameter

appear to be approximately normal as expected based on assumptions on θ and setting $D = 1.702$. Further the empirical cdf is overlapping for both chains.

Conversely, the trace plot for β_3 shows that chains have neither converged nor mixed well. The auto-correlation plot demonstrates high correlation between samples suggesting a very slow mixing process. Chains do not appear to converge and are mixing very slowly. As a result, it is necessary to increase the chain length and re-run analyses and obtain additional samples.

4.2 Diagnostic Statistics

Although graphical methods of evaluating convergence are useful and intuitive, they are subjective and become impractical when many parameters must be assessed. Thus, it is recommended to evaluate MCMC convergence using numeric metrics as well. We demonstrate how to obtain the Geweke and Gelman Rubin statistics from the `coda` package.

Geweke Statistics The Geweke convergence diagnostic tests the equality of means of two segments of a Markov chain with the null hypothesis that the mean of a preliminary segment of the chain (e.g., first 10%) is equal to the latter segment (e.g., last 50%). The Geweke statistic should be applied to individual Markov chains and can be obtained using the `geweke.diag` function from the

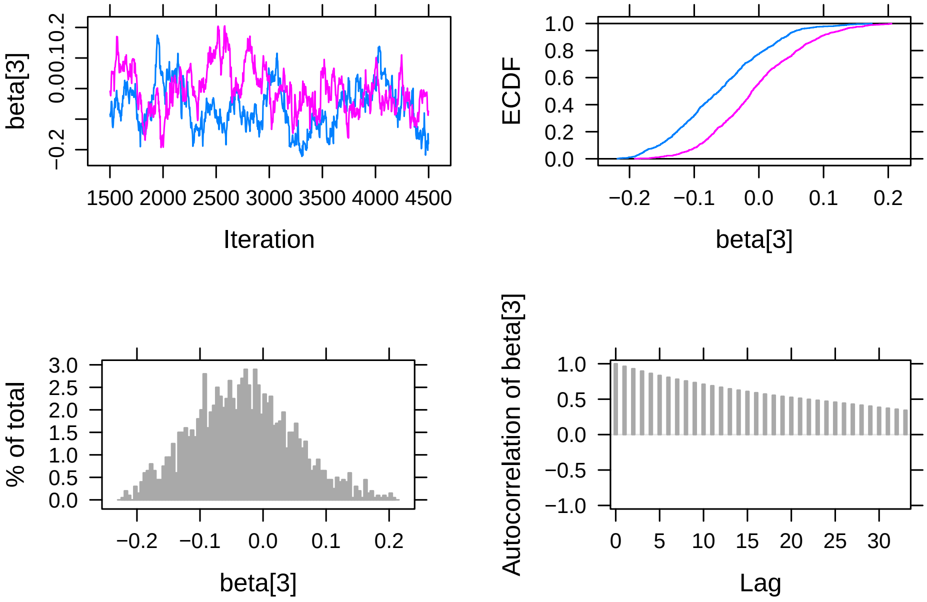

Figure 2. Graphical Convergence Plots for β_3 Suggesting Nonconvergence

`coda` package. Geweke statistics are Z-scores; values larger than ±1.96 suggest a lack of convergence.

```
chain1 = out.2PL[["mcmc"]][[1]]
geweke1 = geweke.diag(chain1)
```

We show Geweke statistics for the first 4 respondents and all item parameters in the first chain below (Table 1). Here, the Geweke statistics suggest a lack of convergence for θ_1, α_1 and β_4 in chain 1. These results suggest that for these chains there is a significant difference between the initial samples in this chain and the later samples. Readers are encouraged to examine the Geweke statistics for all chains as individual chains may reach convergence faster than others.

Table 1. Geweke Statistics

theta	alpha	beta
-2.446	2.031	-1.797
1.229	0.947	0.997
-1.224	-1.114	0.426
-0.254	1.394	2.165

Gelman and Rubin Statistic The Gelman and Rubin convergence diagnostic, also denoted as the psrf or rhat, requires multiple Markov chains and can be intuitively understood as a ratio of between chain variance to within chain variance. Values near 1 are preferred and values less than 1.1 are typically used as evidence of chain convergence (Gelman et al., 2015). The `gelman.diag` function from the `coda` package calculates point estimates and upper confidence limits of the psrf for each parameter in the chain.

```
psrf = gelman.diag(out.2PL)
```

Again, we show psrf diagnostics for the first 4 person parameters as well as the item parameters below (Table 2). Following the pattern from the graphical examination of convergence, θ_i appears to show convergence. Convergence for item parameters, however, is less consistent with β_3 demonstrating $psrf > 1.1$. Thus, both graphical and statistical methods suggest that chains are yet to converge.

Table 2. Gelman and Rubin Statistics

theta	alpha	beta
1.006	1.000	1.024
1.003	1.000	1.075
1.008	1.011	1.502
1.010	1.005	1.018

Successful chain convergence is necessary for all model parameters prior to subsequent analysis steps. Without convergence, there is insufficient evidence to support the assumptions that MCMC has reached the stationary posterior distribution needed for inference. Thus, descriptions or inferences drawn from non-convergent Markov Chains are largely invalid. To obtain convergence in the 2PL example, the number of iterations was increased to ensure all $psrf < 1.10$. The `extend.jags` function can be used to continue MCMC sampling from an exiting `runjags` object. Below, we extend the `out.2PL` object by 1,000,000 iterations. Following this, we confirm that convergence for all parameters has been achieved using the Gelman-Rubin statistic.

```
out.2PL.ext = extend.jags(out.2PL, method="parallel",
                          sample=1000000, adapt=3000)
```

Examination of the psrf from the `coda` package for the longer 2PL chains show convergence in both person and item parameters using $psrf < 1.1$ as the criteria for convergence. The 5 largest psrf values after extending the chain are provided below.

```
out.2PL.ext.psrf = gelman.diag(out.2PL.ext)
out.2PL.ext.psrf[["psrf"]][order(out.2PL.ext.psrf[["psrf"]][,1],
                                 decreasing=TRUE)[1:5],1]
```

```
##  beta[1] alpha[2]  beta[4]  beta[3]  beta[2]
##    1.010    1.008    1.007    1.002    1.000
```

For demonstration, we again provide plots to assess convergence of β_3 graphically (Figure 3). Note the overlap of chains in both the trace and empirical CDF plots (i.e., upper left and upper right panes). This is in contrast to the preliminary assessment of convergence above. Additionally, the autocorrelation plot suggests MCMC samples are much closer to independent samples when contrasted with the original convergence plots.

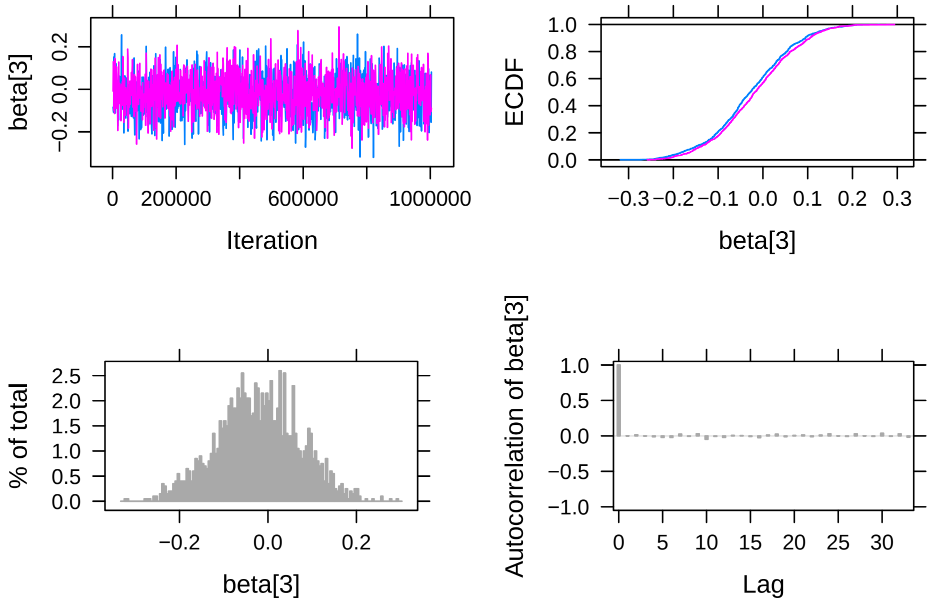

Figure 3. Graphical Convergence Plots for β_3 Suggesting Convergence

Assessing convergence for the GRM follows the same procedure. Below we observe that the original MCMC did not reach convergence for chain lengths of 300,000 samples.

```
grm.gelman=gelman.diag(out.grm2)
max(grm.gelman[["psrf"]])
```

For demonstration, we extend the GRM chains using the `autoextend.jags` function which automatically extends MCMC chains until a target psrf is obtained (e.g., `psrf.target=1.10`). We see below that our psrf threshold was met. The `autoextend.jags` function may not work well for complicated models (Denwood, 2016); furthermore, we recommend assessing convergence via the `coda` package following chain extension to ensure the validity of any subsequent conclusions.

```
grm.auto = autoextend.jags(out.grm2, psrf.target=1.10,
                           method="parallel")
grm.auto.psrf = gelman.diag(grm.auto)
max(grm.auto.psrf[["psrf"]][,1]) # < 1.10
```

```
## [1] 1.008
```

5 Summarize Posterior Samples

Posterior distributions which successfully pass convergence checks can be summarized and, if desired, used for inferential analyses. The `runjags` package contains a `summary` function which provides Highest Posterior Density (HPD) intervals, measures of central tendency, and other useful information such as effective sample size (i.e., `SSeff`). Effective sample size (`SSeff`) provides a metric of information present in a MCMC accounting for auto-correlation among samples. Recall, however, that samples are correlated and do not provide independent information about the parameter. Effective sample sizes of at least 400 are recommended (Gelman et al., 2015). HPD interval confidence level can be specified using the `confidence` argument.

For convenience, we split the person parameter (i.e., $\hat{\theta_i}$) and item parameter estimates (i.e., $\hat{\alpha_j}$, $\hat{\beta_j}\}$) as well as the PPP into separate summary objects. Below we provide example summary output of the `summary` function for the first 3 θ_i parameters.

```
thetas = summary.2PL[startsWith(rownames(summary.2PL),"theta["),]
betas = summary.2PL[startsWith(rownames(summary.2PL),"beta["),]
alphas = summary.2PL[startsWith(rownames(summary.2PL),"alpha["),]
item.params = rbind(betas,alphas)
ppps = summary.2PL[startsWith(rownames(summary.2PL),"ppp["),]
```

```
##          Lower95 Median Upper95   Mean    SD MCerr MC%ofSD SSeff AC.10 psrf
## theta[1]  -0.045  1.032   2.414  1.108 0.686 0.003     0.5 40000 0.004    1
## theta[2]  -2.366 -1.046   0.000 -1.119 0.658 0.003     0.5 39310 0.004    1
## theta[3]  -0.093  0.718   1.975  0.812 0.586 0.003     0.5 40000 0.010    1
```

Posterior distributions of θ_i and HPD intervals can be used to compare ability/severity across individual respondents. Below (Figure 4), 95% HPD intervals of $\hat{\theta_i}$ are plotted for θ_i. It is readily seen that although individuals vary in their point estimates of depression, the interval estimates largely overlap. HPDs are gplotted for both the 2PL and the converged GRM (code provided in supplemental material). Notice that HDIs for the 2PL are much larger in this example which is partially attributable to dichotomizing ordinal response options. Additionally, note that θ_i HPD intervals in the GRM centered above $\theta_i = 1$ posses narrower intervals which is a function of item information discussed in a subsequent section.

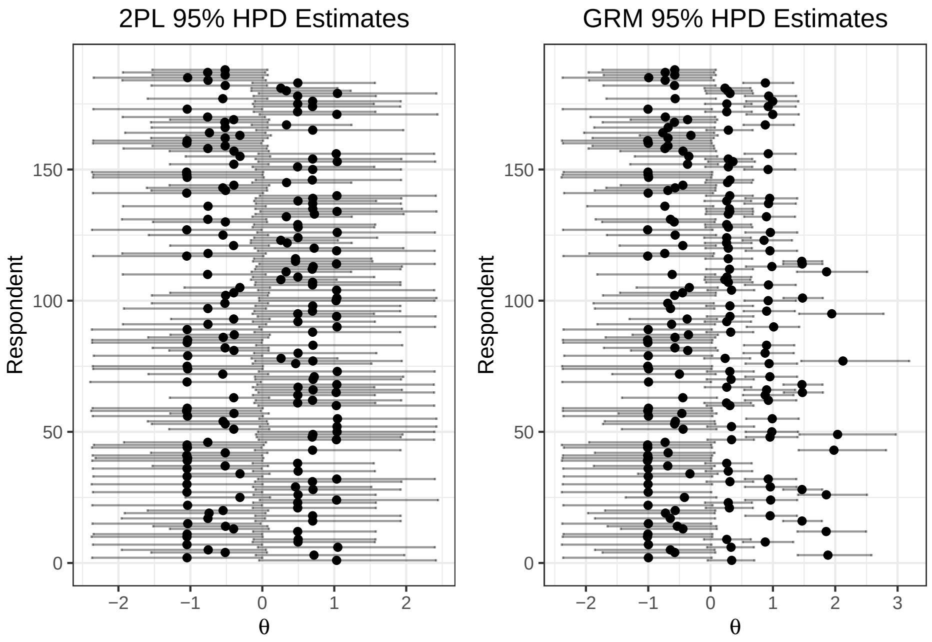

Figure 4. 95% HPD Intervals for 2PL and GRM θ Estimates

Here we see that the 2PL and the GRM yield similar $\hat{\theta}$ estimates; however, the intervals for the GRM are typically much narrower. This reflects the increased precision in estimates that arises from preserving the original ordinal scale of the items rather than dichotomizing it as was done for 2PL demonstration. Further, it is worth noting that the width of the HPD varies based on the $\hat{\theta}$ with estimates larger than zero demonstrating relatively more precision. This suggests that these particular items may be better at distinguishing among respondents with mild to moderate levels of depression than their non-depressed counterparts.

6 Model Fit

Posterior predictive checks can be used to determine if the proposed models fit the observed data. We use the PPP (Gelman et al., 1996). We observe that $PPP \approx 0.50$ for items 1,2 and 4 but $ppp_3 = 0.9$ suggesting that the the 2PL is a reasonable model for items 1,2, and 4 does not perform well for item 3. PPPs could also be obtained for each respondent to detect potential outlying response patterns by altering the `JAGS` model specification to include PPPs for each i. We can use the posterior mean of the PPP_j to examine model fit for each item.

```
## ppp[1] ppp[2] ppp[3] ppp[4]
##  0.516  0.532  0.903  0.524
```

7 Item Curves

It is often of interest when conducting IRT analyses to evaluate how well test items perform across a range of θs. For example, certain items may be more informative for respondents with high levels of θ while other perform better at lower levels. In this section we demonstrate how to plot Item Characteristic Curves (ICCs), Item Information Curves (IICs), and test information for the 2PL. Given the poor model fit for item 3, we only examine curves for items 1, 2, and 4. We use the posterior means of all item and person parameters to compute p_{ij}.

```
alpha_hat = alphas[c(1,2,4),"Mean"]
beta_hat = betas[c(1,2,4),"Mean"]
theta_hat = thetas[,"Mean"]
p = calcP(thet=theta_hat,a=alpha_hat,b=beta_hat,D=1.702)
colnames(p) <- paste0("Item",c(1,2,4))
```

7.1 Item Characteristic Curves

Researchers often wish to examine how the probability of endorsing (or correctly answering) an item varies as a function of θ. ICCs plot p_j across the range of θ providing a useful description of item functioning. Figure 5 plots p_{ij} over $\hat{\theta}$ for items 1, 2, and 4.

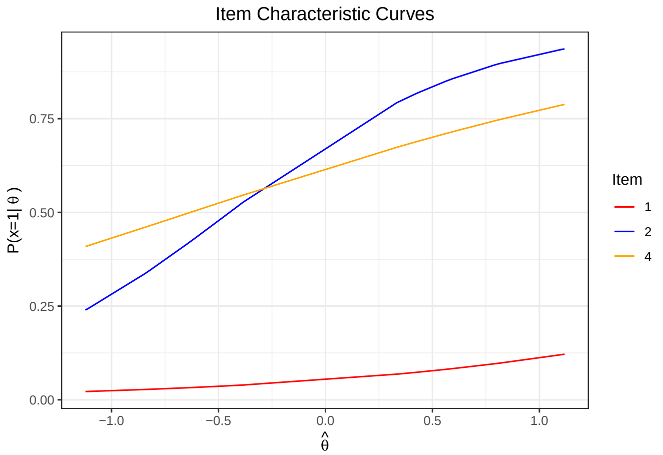

Figure 5. Item Characteristic Curves of Items 1, 2, and 4

The ICC demonstrates that item 1 is rarely ever endorsed across the θ range. The remaining items are endorsed more frequently as θ increases (i.e., more severe depression).

7.2 Item Information Curves

It is also often helpful to plot the item information curves (IIC) which depict how informative responses to an item are for a given level of ability. Items are often evaluated using Fisher information which is defined for the 2PL (Lord, 1980)

$$I(\theta, x_j) = \alpha_j^2 p_{ij}(1 - p_{ij}). \tag{16}$$

Fixing α_j to be the posterior mean as above, we can obtain item information curves. Figure 6 displays the item information for Items 1, 2, and 4.

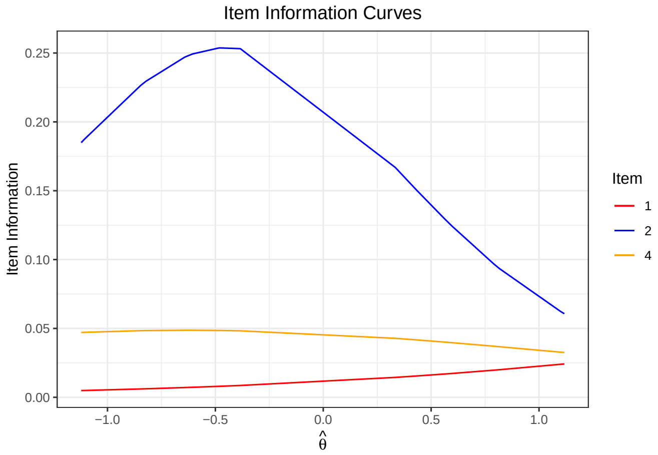

Figure 6. Item Information Curves for Items 1, 2, and 4

7.3 Test Information

Item information provides a metric for how well an item performs across values of θ. The test information provides a similar metric for the entire test. Given the assumption of local independence, calculating test information is a straightforward sum of the item information. Below we demonstrate how to plot the overall test information again omitting item 3 (Figure 7).

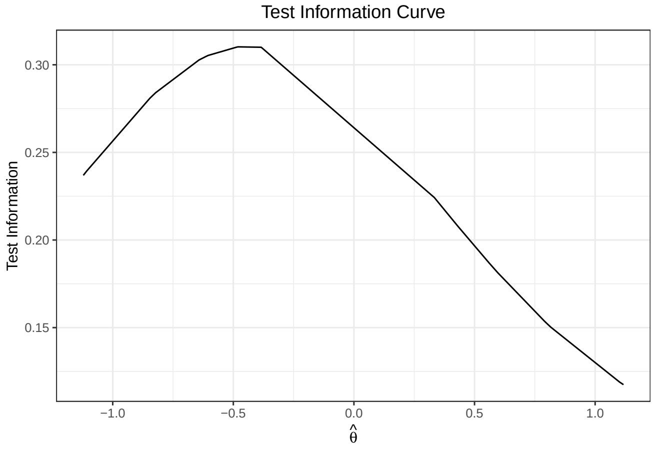

Figure 7. Test Information Curve for Items 1, 2, and 4

8 Summary

This paper provides a demonstration of Bayesian IRT models, specifically the 2PL and GRM, in `JAGS` using the `runjags` package. The general procedure for conducting Bayesian analyses can be summarized in 7 overarching steps:

1. Model Specification
   - Step 1a: Specify the desired model
   - Step 1b: Specify priors and hyper-priors
2. Specify Initial Values in a named list
   - If multiple chains are to be run, this list should be a list of named lists.
3. Specify Data for Analysis in a named list
   - This list should contain data (e.g., item responses) as well as variables such as N or J which are used in model specification.
4. Conduct MCMC sampling using `run.jags`
5. Convergence Analysis
   - Successful convergence in all parameters is necessary to proceed to later steps.
   - If convergence is not met, increase the length of the MCMC.
   - Convergence can be assessed graphically and statistically.
6. Summarize Posterior Distributions
7. Assess Model Fit
8. Conduct Desired Inferential Analyses

Details of each step clearly varies based on models and analytic but this general template provides a heuristic for conducting Bayesian IRT analyses using `JAGS`. Code to implement the 2PL and GRM as well as conduct convergence diagnostics and summarize posterior MCMC is provided.

9 Discussion

This paper provided a demonstration of Bayesian IRT models via the 2PL and GRM in JAGS using data from the NLSY. In general, implementing models in JAGS requires 1) Model specification including priors, 2) specification of initial values, and 3) specifying the data needed to run the model. Depending on idiosyncracies of item response models, dummy coded data for certain parameters, such as the item intercepts in the GRM examined here, may be necessary in JAGS. The models demonstrated here are by no means exhaustive of item response models that can be analyzed in JAGS but provide a foundation for readers to understand the general process of implementing IRT models in JAGS and evaluating model convergence.

References

Baker, F. B., & Kim, S.-H. (2004). *Item response theory: Parameter estimation techniques* (Second ed.). Boca Raton: CRC Press. doi: https://doi.org/10.1201/9781482276725

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. *ETS Research Report Series*, *1981*(1), i–8. doi: https://doi.org/10.1002/j.2333-8504.1981.tb01255.x

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. Lord & M. Novick (Eds.), *Statistical theories of mental test scores* (pp. 397–479). Reading: Addison-Wesley.

Bureau of Labor Statistics. (2017). *National Longitudinal Survey of Youth 1997 cohort, 1997-2017 (rounds 1-18).*

Cain, M. K., & Zhang, Z. (2019). Fit for a bayesian: An evaluation of ppp and dic for structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *26*(1), 39–50. doi: https://doi.org/10.1080/10705511.2018.1490648

Camilli, G. (1994). Teacher's corner: Origin of the scaling constant d = 1.7 in item response theory. *Journal of Educational Statistics*, *19*(3), 293–295. doi: https://doi.org/10.3102/10769986019003293

Curtis, S. M. (2010, August). BUGS code for item response theory. *Journal of Statistical Software*, *36*, 1–34. doi: https://doi.org/10.18637/jss.v036.c01

Denwood, M. J. (2016). Runjags: An R package providing interface utilities, model templates, parallel computing methods and additional distributions for MCMC models in JAGS. *Journal of statistical software*, *71*, 1–25. doi: https://doi.org/10.18637/jss.v071.i09

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists.* Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Fox, J.-P. (2010). *Bayesian item response modeling: Theory and applications.* New York, NY: Springer. doi: https://doi.org/10.1007/978-1-4419-0742-4

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2015). *Bayesian Data Analysis* (Third ed.). New York: Chapman and Hall/CRC. doi: https://doi.org/10.1201/b16018

Gelman, A., Meng, X.-L., & Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, *6*(4), 733–760.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*(4), 457–472.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments. *Bayesian Statistics*, *4*, 641–649. doi: https://doi.org/10.21034/sr.148

Heidelberger, P., & Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, *31*(6), 1109–1144. doi: https://doi.org/10.1287/opre.31.6.1109

Lord, F. M. (1980). *Applications of item response theory to practical testing problems.* New York: Routledge. doi: https://doi.org/10.4324/9780203056615

Lord, F. M. (1986). Maximum likelihood and bayesian parameter estimation in item response theory. *Journal of Educational Measurement*, *23*(2), 157–162. doi: https://doi.org/10.1111/j.1745-3984.1986.tb00241.x

Lunn, D., Spiegelhalter, D., Thomas, A., & Best, N. (2009). The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, *28*(25), 3049–3067. doi: https://doi.org/10.1002/sim.3680

Patz, R. J., & Junker, B. W. (1999). A straightforward approach to Markov Chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics*, *24*(2), 146–178. doi: https://doi.org/10.2307/1165199

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.*

Plummer, M. (2022). *Rjags: Bayesian graphical models using MCMC* [Manual].

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006, March). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, *6*(1), 7–11.

R Team Core. (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.

Raftery, A. E., & Lewis, S. M. (1992). Practical Markov Chain Monte Carlo: Comment: One long run with diagnostics: Implementation strategies for Markov Chain Monte Carlo. *Statistical Science*, *7*(4), 493–497. doi: https://doi.org/10.1214/ss/1177011143

Reckase, M. (2009). *Multidimensional item response theory.* New York, NY: Springer. doi: https://doi.org/10.1007/978-0-387-89976-3

Roy, V. (2020). Convergence diagnostics for Markov chain Monte Carlo. *Annual Review of Statistics and Its Application*, *7*, 387–412. doi: https://doi.org/10.1146/annurev-statistics-031219-041300

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*, *34*(4), 100–100. doi: https://doi.org/10.1007/bf03372160

Stan Development Team. (2023). *RStan: the R interface to Stan.* Retrieved from `https://mc-stan.org/` (R package version 2.21.8)

Appendix A NLSY Depression Items

NLSY Depression Items:

1. How often have you been a nervous person?
2. How often have you been calm/peaceful in the past month? (R)
3. How often have you felt down or blue?
4. How often have you been depressed in the last month?

(R) = reverse coded

Appendix B Supplemental Code

```
dep2017.binary <- apply(dep2017,2,function(x){ifelse(x>1,1,0)})
dep2017[dep2017[,1]==5,1]<-4

summary.theta.df = data.frame(thetas)
summary.theta.df$id <- 1:nrow(thetas)
hpd1 <- ggplot(data=summary.theta.df,aes(y=id))+
  geom_point(aes(x=Median))+
  geom_errorbar(aes(xmin=Lower95,xmax=Upper95),alpha=.4)+
  ylab("Respondent")+
  xlab(expression(theta))+
  ggtitle("2PL 95% HPD Estimates")+
  theme_bw()+theme(plot.title=element_text(hjust=.5))

summary.thetas.grm.df = data.frame(thetas.grm)
summary.thetas.grm.df$id <- 1:nrow(thetas.grm)
hpd2 <- ggplot(data=summary.thetas.grm.df,aes(y=id))+
  geom_point(aes(x=Median))+
  geom_errorbar(aes(xmin=Lower95,xmax=Upper95),alpha=.4)+
  ylab("Respondent")+xlab(expression(theta))+
  ggtitle("GRM 95% HPD Estimates")+theme_bw()+
  theme(plot.title=element_text(hjust=.5))

# Calculate probability of x = 1 given theta, alpha, beta
calcP <- function(theta,a,b,D=1.702){ #D is scaling constant
  logitP = D*(a%*%t(theta)-b)
  p = exp(logitP)/(1+exp(logitP))
  return(t(p))
}


# Item Characteristic Curves
ICC.df = data.frame(theta=theta_hat,p)
ICC.plot <- ggplot(ICC.df,aes(x=theta))+
  geom_line(aes(y=Item1,color=’1’))+
  geom_line(aes(y=Item2,color=’2’))+
  geom_line(aes(y=Item4,color=’3’))+
  xlab(expression(hat(theta)))+
  ylab(expression("P(x=1|"~theta~")"))+ggtitle("ICC Plot")+
  scale_color_manual(name="Item",breaks=c("1","2","3"),
       values=c("1"="red","2"="blue","3"="orange"),
                     labels=c("1","2","4"))+
  theme_bw()+theme(plot.title=element_text(hjust=.5))

# Item Information Curve
IIC.df = data.frame(I.item1 = alpha_hat[1]^2*(p[,1]*(1-p[,1])),
                I.item2 = alpha_hat[2]^2*(p[,2]*(1-p[,2])),
                I.item4 = alpha_hat[3]^2*(p[,3]*(1-p[,3])),
                theta_hat)
IIC.plot <- ggplot(IIC.df,aes(x=theta_hat))+
  geom_line(aes(y=I.item1,color=’1’))+
  geom_line(aes(y=I.item2,color=’2’))+
  geom_line(aes(y=I.item4,color=’3’))+
  xlab(expression(hat(theta)))+
  ylab("Item Information")+ggtitle("Item Information Plot")+
  scale_color_manual(name="Item",breaks=c("1","2","3"),
            values=c("1"="red","2"="blue","3"="orange"),
                     labels=c("1","2","4"))+
  theme_bw()+theme(plot.title=element_text(hjust=.5))

# Test Information Curve
test.i.df = data.frame(theta_hat,
             testInfo=apply(IIC.df[,1:3],1,sum))
test.i.plot <- ggplot(test.i.df,aes(x=theta_hat))+
  geom_line(aes(y=testInfo))+
  ylab("Test Information")+xlab(expression(theta))+
  xlab(expression(hat(theta)))+
  ggtitle("Test Information Curve")+
  theme_bw()+theme(plot.title=element_text(hjust=.5))
```