1

Regression, Structural Equation Modeling, `Mplus`, and `lavaan`

HISTORICAL OVERVIEW

Structural equation modeling (SEM) is a comprehensive analytic framework for estimating statistical models and evaluating statistical models against empirical data. The SEM framework is a multivariate framework and has been used to test theories regarding indirect effects, latent variables underlying measured indicators, growth and change of constructs, and the invariance of model parameters across groups and time. Historically, the SEM framework was a *linear* modeling framework building on multiple linear regression models and extending to *linear* latent variable models (e.g., linear confirmatory factor models). Thus, data for SEMs were assumed to be normally distributed. With this assumption, the covariance matrix for the empirical data could be calculated and used to estimate model parameters.

Categorical variables were first considered in the SEM framework in the 1970s. Initial inclusion of categorical variables focused on binary (dichotomous) variables as indicators in a confirmatory factor model. In 1970, Bock and Lieberman (1970) presented a method for estimating parameters of the normal ogive model (cumulative normal distribution) for binary indicators of a common factor using a maximum likelihood estimation routine. This allowed for the expected nonlinear associations between common factors and binary indicators but faced computational challenges. The computational challenges essentially put a limit on the number of indicators (10 to 12 indicators) and the number of common factors (i.e., 1 common factor).

Christoffersson (1975) and Muthén (1978) took a different approach to the estimation of common factor models with binary indicators. They discussed an approach based on first estimating the thresholds (i.e., means, proportions) and the tetrachoric correlations for the sample data, and then using a generalized least squares estimator to estimate

model parameters. This work increased the number of indicators that could be analyzed, increased the number of factors that could be considered, and led to further developments for binary variables within the general SEM framework (Muthén, 1979, 1983; Muthén & Christoffersson, 1981). This work culminated in a general SEM framework for binary, ordered categorical, and quantitative variables (Muthén, 1984) using generalized least squares estimators.

The work by Bock and Lieberman (1970) led to improvements in the estimation of item response models (Lord & Novick, 1968). Item response models, at the time, were a class of latent variable models for binary and ordered categorical indicators with strict assumptions. These assumptions included a single latent variable (i.e., unidimensionality), local independence (i.e., item responses are independent after accounting for the underlying factor), and monotonicity (i.e., monotonic association between the underlying factor and item response). These models were primarily developed in the educational sciences and independent of the growth in the factor analysis of binary and ordered categorical outcomes in the psychological sciences.

Over time, the item response and factor analytic frameworks for binary and ordered categorical outcomes were united with the realization of their equivalence. Now, SEM programs (e.g., `Mplus`, `lavaan`, and `Lisrel`) allow for the specification and estimation of item response models, and item response modeling programs (e.g., `flexMIRT` and `IRTPRO`) allow for the specification of models traditionally fit in the SEM framework. These programs are often referred to as *general latent variable modeling programs* because of their ability to estimate a variety of models.

STRUCTURAL EQUATION MODELING

SEM is a framework for the specification and estimation of statistical models. This general definition of SEM encompasses a large number of statistical models, including, but not limited to, regression models, path models (i.e., multivariate regression models), confirmatory factor models, path models with latent variables, and finite mixture models. SEM is often considered a *theory-driven* framework, where researchers specify models that were developed based on theory, and test these models against empirical data. Thus, the specified statistical model is a representation of the theory, which can include latent variables to represent *constructs* (often indicated by observed variables), as well as direct effects, indirect effects, and symmetric associations between variables.

When approaching a data analysis project with SEM, the following five steps are recommended: (1) *Theory* → *Model*: Form ideas based on theory for how constructs are expected to be related to one another; (2) *Model Formulation*: Determine how measured variables fit into the theory-driven model; (3) *Model Specification and Estimation*: An SEM program is used to specify the model, estimate model parameters and standard errors, and calculate various indices of model fit; (4) *Evaluation and Interpretation*: Examine the *fit*

of the model (potentially compare the fit of the proposed model to *alternative* models), and interpret the model parameters; (5) *Extension*: Explore new ideas and models based on the findings.

The first step is *Theory* $\rightarrow$ *Model*, in which researchers form statistical models relating the constructs of interest. Cooley (1978, 13) noted that "the purpose of statistical procedures is to assist in establishing the plausibility of a theoretical model." The SEM framework is a general statistical framework that allows researchers to be *explicit* about theory and how it is reflected in the model. The goal is to match the model *as closely as possible* to the theory, which then allows for the examination of the plausibility of the model given the observed data. I recommend this step is done with constructs in mind as opposed to measured variables.

The second step is *Model Formulation*, in which measured variables are placed into the theoretical model. The measured variables may be indicators of latent variables representing the constructs from Step 1, they may represent the constructs from Step 1 directly, or they may represent measured explanatory variables or covariates. The third step is *Model Specification and Estimation*, in which the statistical model from Step 2 is specified using an SEM program and the model parameters are estimated. It's important to ensure that the model is identified (i.e., a unique set of parameter estimates optimize the fit of the model). Estimation is often carried out using *maximum likelihood estimation*. The maximum likelihood estimates (assuming multivariate normal data) are those that minimize the maximum likelihood fit function (F_{ML}) contained in Equation 1.1. Calculating maximum likelihood estimates is an iterative process. At each iteration, parameter estimates and the maximum likelihood fit function are updated. The maximum likelihood fit function is minimized when the difference between the model-implied covariance matrix and the covariance matrix for the measured variables is minimized.

EQUATION 1.1.

$$F_{ML} = \log|\boldsymbol{\Sigma}(\hat{\theta})| + \text{tr}\left(\mathbf{S}\boldsymbol{\Sigma}^{-1}(\hat{\theta})\right) - \log|\mathbf{S}| - (p+q)$$

- $\hat{\theta}$ - current set of parameter estimates
- $\boldsymbol{\Sigma}(\hat{\theta})$ - model-implied covariance matrix with current parameter estimates
- $\mathbf{S}$ - covariance matrix for the measured variables
- $p + q$ - number of measured variables
- $||$ - determinant function
- tr () - trace function

The fourth step is *Evaluation and Interpretation*, in which the *fit* of the model is examined to determine whether the model is plausible. Often, model fit is evaluated using various fit indices, including the χ^2 test of model fit, the Root Mean Square Error

of Approximation (RMSEA), the Comparative Fit Index (CFI), the Tucker-Lewis Index (TLI), and the Standardized Root Mean Square Residual (SRMR). The RMSEA and SRMR are absolute fit indices, where lower values indicate better model fit. The CFI and TLI are incremental fit indices, where higher values indicate better model fit. The *residual* covariance matrix should also be examined to understand where the model fits and misfits the observed covariance matrix (i.e., the data). If the model is determined to be plausible (i.e., consistent with the data), the parameter estimates (e.g., factor loadings, direct effects, indirect effects) should be interpreted.

The fifth and final step is *Extension*, in which the results from Step 4 are used to determine next steps. For example, if the model fit was poor, novel models may be proposed (and potentially evaluated with novel data). If the model fit was good, then next steps to evaluate the theory may be proposed.

PATH DIAGRAMS

Path diagrams are a key aspect of SEMs. These diagrams depict the model's specification, which aids in the communication of complex multivariate models. Path diagrams can also be used for model specification in some statistical programs (e.g., AMOS). Path diagrams using reticular action model (RAM) notation (McArdle, 1980, 2005; McArdle & McDonald, 1984) contain all of the model's components and parameters, and this approach to drawing path diagrams is used throughout this book.

In these diagrams, squares represent measured (observed) variables (i.e., variables in the dataset), and circles represent unmeasured (latent) variables (i.e., variables not contained in the dataset). One-headed arrows represent directive associations, such as regression slopes and factor loadings, whereas two-headed arrows represent symmetric associations, such as variances and covariances. A triangle is used to represent the constant, which allows for the inclusion of means and intercepts as one-headed arrows originating from the triangle.

A RAM notation path diagram of a single-factor model is shown in Figure 1.1. The latent variable η_1 is indicated by five measured variables named y_1 through y_5. The factor loading and measurement intercept for y_1 are fixed at 1 and 0, respectively. This allows us to estimate the mean and variance of η_1. The remaining factor loadings are estimated and labeled λ_{21} through λ_{51}, denoting the measured variable number (i.e., 2 through 5) and the latent variable number (i.e., 1). The measurement intercepts for y_2 through y_5 are labeled ν_2 through ν_5 and the unique factor variances for y_1 through y_5 are labeled θ_{11} through θ_{55}. The unique covariance between y_4 and y_5 is denoted θ_{54}. The latent variable's variance is denoted ψ_{11}, and its mean is α_1. The labeling of model parameters here follows the *all-y* notation from Lisrel.

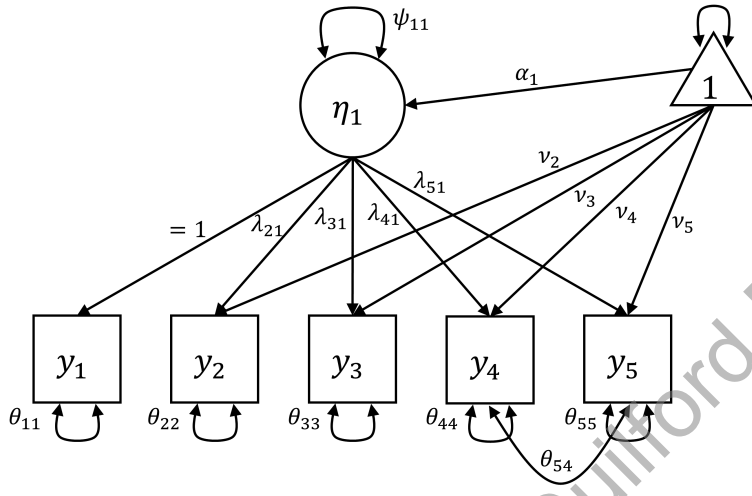

FIGURE 1.1. RAM notation path diagram of a single-factor model.

STATISTICAL PROGRAMS

Throughout the book, I utilize two statistical programs. These programs are R (R Core Team, 2021) and Mplus (Muthén & Muthén, 1998–2017). R is a free, comprehensive, open-source statistical program that can be downloaded from https://cran.r-project.org/. R allows and encourages researchers to modify, extend, and develop additions to the base program. These additions are referred to as *packages*. The packages used throughout the book include the ggplot2 (Wickham, 2016) package for high-quality plotting; the psych (Revelle, 2023) package for descriptive statistics; the VGAM (Yee, 2015), MASS (Venables & Ripley, 2002), and pscl (Jackman, 2020) packages for regression models with categorical outcomes; and the lavaan (Rosseel, 2012) package for SEM. lavaan is a comprehensive SEM package that can handle both continuous variables and ordered categorical variables. lavaan also has a straightforward programming language for model specification and several estimation routines for fitting SEMs with ordered categorical outcomes.

The second program is Mplus, which is the most comprehensive latent variable modeling program available. Mplus can fit SEMs, multilevel models, and mixture models, and it can handle continuous, ordered categorical, unordered categorical, count, zero-inflated, and censored outcome variables. Mplus has efficient estimation routines, especially for categorical outcomes; features a straightforward programming language; and is continually being improved. This makes Mplus the most utilized latent variable modeling program available. The Mplus website (http://statmodel.com/) contains a demonstration

version of the program that is only limited by the number of variables included in the analysis, the user manual, a collection of examples, discussion forums, and a series of papers highlighting the features of the program.

lavaan

Models are specified and estimated using lavaan in two steps. The first step is the model's specification and the second step is the model's estimation using one of lavaan's functions. The model's parameters are specified using a series of symbols to relate variables to one another or to themselves. The model's specification is contained within quotes and assigned to an object.

The symbol ~ is used to specify regression models with the outcome variable on the left-hand side and the explanatory variables on the right-hand side. For example, yi ~ x1i + x2i specifies a regression model with yi as the outcome and x1i and x2i as the explanatory variables. This notation follows the specification of regression models in R using the lm() or glm() functions. The same symbol, ~, is also used to specify means and intercepts when the right-hand side of the equation is 1. For example, y1i ~ 1 specifies the mean of y1i if it is an explanatory variable or its intercept if it is an outcome variable (note that the term *intercept* is simply a conditional mean).

The symbol =~ is used to denote factor loadings of measurement models with the latent variable on the left-hand side and the indicators on the right-hand side of the symbol. For example, eta1 =~ y1i + y2i + y3i specifies eta1 as the latent variable with y1i, y2i, and y3i as its indicators. The symbol ~~ is used to specify variances and covariances (i.e., two-headed arrows in the path diagram). Covariances are specified by listing different variables on each side of the symbol, and variances are specified by listing the same variable on both sides of the symbol. For example, y1i ~~ y2i specifies the covariance between these two variables, and y1i ~~ y1i specifies the variance of y1i.

Once the model is specified, the model is estimated using one of lavaan's functions. These include lavaan(), sem(), cfa(), and growth(). Each function takes the object name from the model's specification and the name of the dataset. Additional statements, such as estimator= and missing=, are added to control estimation options. The summary() function is then used to print the parameter estimates and model fit information. An example model specification for the model in Figure 1.1 is contained in Appendix 1.A.

Mplus

An input file for Mplus typically has six statements: (1) TITLE:, (2) DATA:, (3) VARIABLE:, (4) ANALYSIS:, (5) MODEL:, and (6) OUTPUT:. The TITLE: and DATA: statements are where the title for the model and the data file to be analyzed are listed.

The VARIABLE: statement is where the names of the variables are listed along with several options regarding the variable scores. For example, the variables to be included in the model and the missing data indicator are listed in the VARIABLE: statement. The ANALYSIS: statement is used to specify aspects of the model's estimation. This statement isn't necessary, but this is where the type of analysis (TYPE=) and the estimator (ESTIMATOR=) are specified. This statement is only used when default settings are not desired. The MODEL: statement is where the parameters of the SEM are specified, and the OUTPUT: statement is used to specify output options, such as requesting standardized parameter estimates or sample statistics.

The MODEL: statement, where the parameters of the SEM are specified, is the focus of this discussion. Mplus uses keywords to specify parameters connecting variables (e.g., regressions, factor loadings, covariances), and parameters related to the variable itself (e.g., mean) are specified by referring to the variable names in different ways. The keyword ON is used to specify regression models with the outcome variable on the left-hand side and the explanatory variables on the right-hand side. For example, `yi ON x1i x2i;` specifies a regression with yi as the outcome and x1i and x2i as the explanatory variables. The keyword BY is used to denote factor loadings of measurement models with the latent variable on the left-hand side and the indicators on the right-hand side. For example, `eta1 BY y1i y2i y3i;` specifies eta1 as the latent variable with y1i, y2i, and y3i as its indicators. Finally, the keyword WITH is used to specify covariances. For example, `y1i WITH y2i;` specifies the covariance between these two variables.

Univariate parameters are specified by referring to the variable names in different ways. Variances and residual variances are specified by listing the variable names. For example, the code `y1i y2i;` specifies the variances (or residual variances if they are outcome variables) of these two variables. Means and intercepts are specified by writing the variable names in brackets. For example, the code `[y1i y2i];` specifies the means (or intercepts if they are outcome variables) of these two variables. An example Mplus input script for the model in Figure 1.1 is contained in Appendix 1.A.

LINEAR REGRESSION

SEM can be viewed as a multivariate extension of linear regression analysis. Thus, having a solid foundation in multiple linear regression is essential for understanding SEM. Here, I review multiple linear regression. I also generate (simulate) data from a linear regression model and estimate a regression model using the simulated data. When fitting the regression model, the goal is to obtain parameter estimates that align with the population parameters used in data generation. This process of generating data and fitting the appropriate statistical model is helpful in understanding the components and assumptions of statistical models. This, in turn, leads to a deeper understanding of the model and its

interpretation. I use this approach throughout the book as I've found it to be especially beneficial when analyzing categorical outcomes.

Regression analysis is the primary analytic tool for developing explanatory models in the social and behavioral sciences. Regression models are also specified for evaluating causal effects in experimental studies. The linear regression model assumes the outcome is quantitative and is written in Equation 1.2. The key parameters of the linear regression model are the *intercept*, which is the predicted value of the outcome when the explanatory variable scores equal 0, and the *slopes*, which are expected differences in the outcome score for a 1-unit increase in the respective explanatory variable score (holding all other explanatory variable scores constant).

EQUATION 1.2.

$$y_i = b_0 + b_1 \cdot x_{1i} + ... + b_p \cdot x_{pi} + e_i$$

- y_i - outcome variable score for individual i
- x_{1i} through x_{pi} - explanatory variable scores for individual i
- b_0 - intercept parameter
- b_1 through b_p - slope parameters
- e_i - residual term

The residual term in the linear regression is the difference between the expected value of y_i (denoted $\hat{y}_i$ or $\mathbb{E}(y_i|x_i)$) and the observed value of y_i, and is assumed to be normally distributed with a mean of 0 and a standard deviation of σ_e. These residuals are assumed to be independent of (uncorrelated with) the explanatory variable scores.

As mentioned earlier, the components of the linear regression model can be better understood through simulation. To highlight the components of linear regression models, I'll simulate data following the linear regression model in Equation 1.3, where the intercept is 25 and the slope is 0.8. In this simulation, I generate data for $N = 500$ cases.

EQUATION 1.3.

$$y_i = 25 + 0.8 \cdot (x_{1i} - 20) + e_i$$

- x_{1i} - the explanatory variable simulated from a normal distribution with a mean of 20 and a standard deviation of 6 (i.e., $x_{1i} \sim N(20, 6)$)
- e_i - the residual scores simulated from a normal distribution with a mean of 0 and a standard deviation of 8 (i.e., $e_i \sim N(0, 8)$)

The R code to generate these data is

```
# sample size
N = 500
# regression parameters
b0 = 25
b1 = .8
# explanatory variable scores
x1i = rnorm(N,20,6)
# prediction equation
yhati = b0 + b1*(x1i - 20)
# residuals
ei = rnorm(N,0,8)
# observed scores
yi = yhati + ei
# putting the simulated data into a data frame
temp = data.frame(x1i,yi,yhati,ei)
```

where I begin by setting the sample size and population parameters for the regression model. Explanatory variable scores are generated using the `rnorm()` function, which generates random scores from a normal distribution with a given mean and standard deviation. Thus, the code `rnorm(N,20,6)` generates 500 random scores from a normal distribution with a mean of 20 and a standard deviation of 6. From the explanatory variable scores, `x1i`, I generate the predicted values of the outcome. I call these predicted values `yhati`, and these values are generated based on the population regression equation. The residual scores are then generated using the `rnorm()` function and come from a normal distribution with a mean of 0 and a standard deviation of 8. The outcome scores, denoted `yi`, are created as the sum of the predicted scores and the residuals. These simulated data are then put into a data frame named `temp`. A bivariate scatterplot of these data is shown in Figure 1.2, where the positive association between `x1i` and `yi` is visualized. The population regression line is also plotted in this figure and the `yi` scores vary around this regression line.

One check on the correctness of the data generation process is to use linear regression software to estimate the regression parameters for these simulated data. The code to specify and estimate a linear regression model in R follows. First, a centered version of `x1i` is calculated. This variable, named `x1iC20`, is equal to `x1i` minus 20. The linear regression model is specified next using the `lm()` function. The regression model is specified as `yi ~ x1iC20` as `yi` is the outcome and `x1iC20` is the explanatory variable. The dataset is `temp`. The object `linearReg1` holds the output from the linear regression and the `summary()` function is used to print the parameter estimates and model fit information.

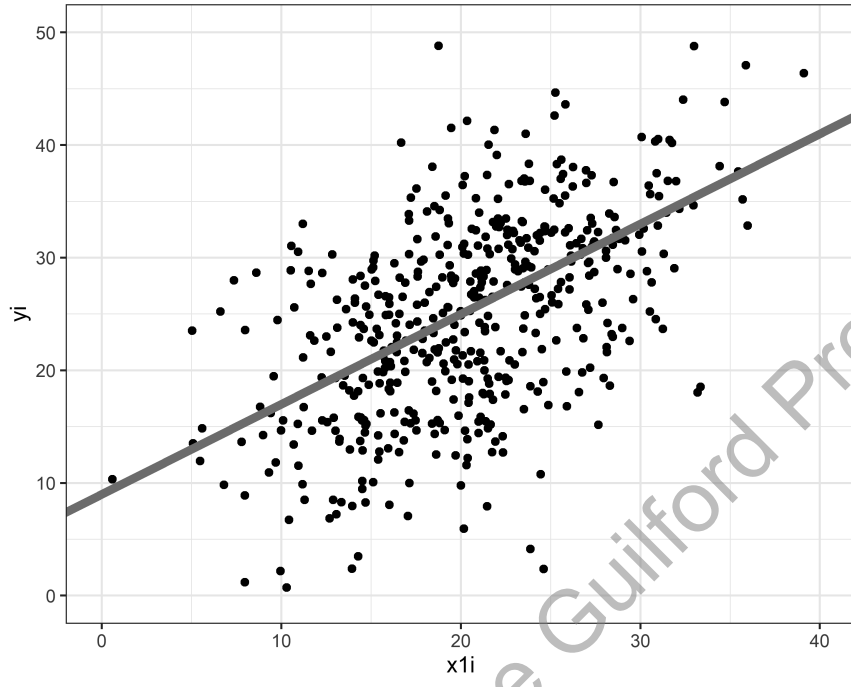

FIGURE 1.2. Bivariate scatterplot with regression line for the simulated data.

```
temp$x1iC20 = temp$x1i - 20

linearReg1 = lm(yi~x1iC20, temp)
summary(linearReg1)

##
## Call:
## lm(formula = yi ~ x1iC20, data = temp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.295  -5.293   0.294 5.133 34.665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.57244    0.34932   70.34   <2e-16 ***
## x1iC20       0.78202    0.05567   14.05   <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.787 on 498 degrees of freedom
## Multiple R-squared: 0.2838, Adjusted R-squared: 0.2823
## F-statistic: 197.3 on 1 and 498 DF, p-value: < 2.2e-16
```

The estimated intercept was 24.57 and the estimated slope for x1iC20 was 0.78, and these values map onto the population parameters used to generate the data, which gives me confidence that the data were correctly generated. The explained variance for the model was 0.28, which is the amount of variance in yi accounted for by x1iC20. The value in the population was 0.26 (26%) and this population value is calculated as the variance in yi associated with x1iC20, which is equal to $b_1 \cdot \sigma^2_{x1} \cdot b_1$, divided by the total variance in yi, which is equal to $b_1 \cdot \sigma^2_{x1} \cdot b_1 + \sigma^2_e$ $\left(\text{i.e., } \frac{b_1 \cdot \sigma^2_{x1} \cdot b_1}{b_1 \cdot \sigma^2_{x1} \cdot b_1 + \sigma^2_e}\right)$.

Explained variance (i.e., R^2) in the linear regression model can be thought about in a variety of ways. For example, explained variance is equal to the squared correlation between the observed and predicted outcome scores $\left(\text{i.e., } \left(\text{cor}\left[y_i, \hat{y}_i\right]\right)^2\right)$, the proportional reduction in the residual variance when fitting the hypothesized model compared to when fitting an intercept-only model $\left(\text{i.e., } \frac{\text{var}(y_i - \bar{y}_i) - \text{var}(y_i - \hat{y}_i)}{\text{var}(y_i - \bar{y}_i)}\right)$, and the proportion of the variance in the outcome attributed to the explanatory variables $\left(\text{i.e., } \frac{\text{var}(\hat{y}_i)}{\text{var}(\hat{y}_i) + \text{var}(e_i)}\right)$. These different ways to think about explained variance are calculated in the following R script, and all lead to the same estimate, which is 0.28. The consistency of these different approaches to calculating explained variance changes when analyzing categorical outcomes.

```
# squared correlation
cor(yi,fitted.values(linearReg1))^2

## [1] 0.2837724
# proportional reduction in residual variance
linearReg0 = lm(yi ~ 1, temp)
1 - var(residuals(linearReg1))/var(residuals(linearReg0))

## [1] 0.2837724
# variance attributable to explanatory variables
var(fitted.values(linearReg1))/var(yi)

## [1] 0.2837724
```


OVERVIEW OF THE BOOK

This book is written to lead the reader from regression analysis with categorical outcomes through complex SEMs with latent variables for categorical indicators. The book is broken down into four sections. The first section, *Regression Analysis with Categorical Outcomes in R*, discusses the specification, estimation, and interpretation of multiple regression models for binary, ordinal (ordered categorical), nominal (unordered categorical), and count outcomes. Empirical data are analyzed in each chapter using R

(comparable code for SAS is available on the companion website). This section sets the stage for the interpretation of model parameters when working with categorical outcomes.

In the second section, *Regression Analysis with Categorical Outcomes in Structural Equation Modeling Programs*, I first review the specification of categorical outcomes in Mplus and lavaan and then discuss the specification, estimation, and interpretation of multiple regression models for binary, ordinal, nominal, and count outcomes using these programs. The same empirical data analyzed in Section 1 are reanalyzed in Section 2, so the comparability of the results is examined. This section highlights how univariate models for categorical outcomes can be estimated with SEM programs, which bridges the gap between regression models for categorical outcomes and more complex SEMs for categorical outcomes.

The third section, *Structural Equation Models with Categorical Outcomes*, discusses path models, confirmatory factor models, and latent variable path models with categorical outcomes. In the first chapter of this section, I describe the specification, estimation, and interpretation of path models with multiple binary and ordinal outcomes as well as path models with binary and ordinal mediators. In the second chapter, I discuss the factor-analytic and item response model specifications of latent variable measurement models for binary and ordinal variables. In the third chapter, I discuss how factor-analytic models can be incorporated into path models. Here, I describe a two-step approach to first examine the fit of the measurement model prior to examining the fit of the full SEM. Empirical data are analyzed in each chapter using Mplus and lavaan.

The fourth and final section, *Advanced Structural Equation Models with Categorical Outcomes*, covers five topics. First, I describe growth models to examine between-person differences in within-person change when the outcome is binary or ordinal. Second, I discuss multiple group confirmatory factor models with binary and ordinal outcomes to evaluate measurement invariance or the invariance of measurement parameters over measured groups of individuals. Third, I present latent class models, which are a type of finite mixture model for binary and ordinal outcomes, to examine whether there are different response patterns for different unobserved groups of participants. Fourth, I discuss count outcomes that have a high preponderance of zero responses. I review zero-inflated and hurdle models for these outcomes, but I limit the discussion to regression models even though more complicated models (e.g., growth models, measurement models) can be fit with these types of outcomes. Finally, I discuss time-to-event data as another form of categorical data and describe discrete- and continuous-time survival models for these data. Empirical data are analyzed in each chapter using Mplus and lavaan, where possible.

RECOMMENDED USES OF THE BOOK

This book is designed based on a categorical SEM workshop I taught through Statistical Horizons and a subsequent graduate course I developed titled *Advanced Categorical Data Analysis*. Thus, this book can be used as a stand-alone text for similar graduate courses in psychology, education, human development, family studies, and sociology. Additionally, this book can be used as a companion for excellent books on categorical data analysis (e.g., *Categorical Data Analysis* by Agresti [2013]; *An Introduction to Categorical Data Analysis* by Agresti [2018]; *Applied Categorical and Count Data Analysis* by Tang et al. [2023]; and *Categorical Data Analysis and Multilevel Modeling Using R* by Liu [2023]) that do not discuss SEM. The book can also be used as a companion for excellent SEM books (e.g., *Principles and Practice of Structural Equation Modeling* by Kline [2011]; *Structural Equations with Latent Variables* by Bollen [1989]; and *Latent Variable Models: An Introduction to Factor, Path, and Structural Analysis* by Loehlin [1998]) that do not cover categorical data SEMs.