Best Practices in Mixture Modeling using Free Open Source Software

Caspar J. van Lissa[1,2]

[1] Utrecht University, Methodology & Statistics

[2] Open Science Community Utrecht

Author Note

Correspondence concerning this article should be addressed to Caspar J. van Lissa, Padualaan 14, 3584CH Utrecht, The Netherlands. E-mail: c.j.vanlissa@gmail.com

Abstract

Latent class analysis is a popular technique for identifying groups in data based on a parametric model. Examples of this technique are known as mixture models, latent profile analysis, latent class analysis, growth mixture modeling, and latent class growth analysis. Despite the popularity of this technique, there is limited guidance with respect to best practices in estimating and reporting mixture models. Moreover, although user-friendly interfaces for advanced mixture modeling have long been available in commercial software packages, open source alternatives have remained somewhat inaccessible. This tutorial describes best practices for the estimation and reporting of latent class analysis, using free and open source software in R. To this end, this tutorial introduces new functionality for estimating and reporting mixture models in the `tidySEM` R-package. These functions rely on estimation using the `OpenMx` R-package.

Keywords: keywords

Word count: X

Best Practices in Mixture Modeling using Free Open Source Software

Latent class analysis (LCA) is an umbrella term that refers to a number of techniques for estimating unobserved group membership based on a parametric model of one or more observed indicators of group membership. This method has become quite popular across scientific fields, and under a number of different names; most notably (finite Gaussian) mixture modeling and latent profile analysis. Vermunt, J.K. et al. (2004) defined it more generally as virtually any statistical model where "some of the parameters [...] differ across unobserved subgroups".

Despite the popularity of the method, there is a lack of standards for estimating and reporting latent class analyses. While Van De Schoot, Sijbrandij, Winter, Depaoli, and Vermunt (2017) developed reporting guidelines for a specific subcategory of LCA known as latent growth models, general reporting guidelines for latent class analysis are still lacking. This complicates manuscript review and assessment of the quality of published studies, and introduces a risk of misapplications of the technique. The present paper seeks to address this gap in the literature by suggesting updated guidelines for estimation and reporting on latent class analysis, based on current best practice. Importantly, in order to lower the barrier of entry and ensure reproducibility of all examples, this paper exclusively relies on free, open source software for latent class analysis in R. Our goal is to make best-practices in latent class analysis widely accessible.

Defining latent class analysis

Latent class analysis can be understood as a method for estimating unobserved groups based on a parametric model of observed indicators of group membership. The concept of latent class analysis can be understood in different ways. Generally speaking, a mixture model assumes that the study population is composed of K subpopulations or classes. It further assumes that the observed data are a mixture of data generated by class-specific models. The simplest univariate "model" is a normal distribution, which can be described with two parameters: the mean and the variance. Commonly, the same model is estimated across all classes, but with different parameters for each class (i.e., class-specific means and variances). Mixture modeling then estimates both the parameters for each class, and the probability that an individual belongs to each class.

As an illustrative example, imagine that a detective wants to know if it would be possible to use mixture modeling to identify the sex of a suspect, based on footprints found at the crime scene. To test the feasibility of this approach, the detective records the shoe sizes and sex of 100 volunteers. The resulting observed data look like this:

The distribution is evidently bimodal, which bodes well for the intended mixture model. In this case, the number of classes is known a-priori. When estimating a two-class mixture model, the detective observes that the model estimates the mean shoe size of the two groups are equal to 7.25 and 9.22, which is close to the true means of the two groups, namely 9.04 and 6.93. When tabulating estimated group membership against observed (known) group membership, it can be seen that women are classified with a high degree of accuracy, but men are not:

A mixture model is like confirmatory factor analysis, except that the continuous latent variable is substituted with a categorical latent variable. One difference between the two techniques is that factor analysis can be considered as a way to group observed *variables* into latent constructs, with factor loadings indicating which items belong are most indicative of a construct. By contrast, mixture modeling groups *individuals* into classes (see Nylund-Gibson & Choi, 2018). In line with this distinction, latent class analysis is sometimes referred to as a "person-centered" technique, and factor analysis as a "variable-centered" technique.

When the focus is on the model parameters in each group, then LCA can be thought of as similar to a multi-group structural equation model. The main distinction is that group membership is not known a-priori, but is instead estimated - with measurement error - based on the data. Whereas in a multi-group model, the data are split by group and treated as independent samples, in a mixture model, all cases contribute to the estimation of all parameters in all groups. The relative contribution of each case to the parameters of each group is determined by that case's posterior probability of belonging to that group.

When the focus is on each individual's estimated class membership, latent class analysis can be thought of as a type of clustering algorithm. In line with this perspective, mixture modeling is sometimes described as "model-based clustering" (Hennig, Meila, Murtagh, & Rocci, 2015; Scrucca, Fop, Murphy, & Raftery, 2016). Many clustering algorithms apply some recursive splitting algorithm to the data. By contrast "model-based" clustering refers to the fact that LCA estimates cluster membership based on a parametric model. Specifically, the posterior class probability that an individual belongs to a latent class can be computed from the likelihood of that individual's observed data under given the class-specific model.

Finally, in the context of machine learning, LCA can be considered as an *unsupervised classification* problem (Figueiredo & Jain, 2002). The term *unsupervised* refers to the fact that the outcome variable, true class membership, is not known, and the term *classification* refers to the fact that the algorithm is predicting a categorical outcome – class membership.

A taxonomy of latent class analyses

In this paper, we use the term latent class analysis to refer to techniques that estimate latent class membership based on a parametric model of observed indicators. From a historical perspective, the term latent class analysis was initially conceived to refer to analyses with categorical (binary) indicators (Vermunt, J.K. et al., 2004). Nowadays, there are a number of related techniques, known by distinct names, that serve a similar purpose. The term "latent class analysis" seems most appropriate as an umbrella term for this broader class of models, as it only refers to the purpose of the analysis, and does not imply restrictions to the model used, or the level of measurement of the indicators. Given the abundance of terms in use for closely related classes of models, we will provide a rudimentary taxonomy of latent class analyses.

One common type of LCA is the *finite Gaussian mixture model*; a univariate analysis where the observed distribution of a single variable is assumed to result from a mixture of a known number of Gaussian (normal) distributions. The parameters of a finite Gaussian mixture model are the means and variances of these underlying normal distributions. The analysis of shoe sizes presented earlier is a canonical example of this type of analysis. In the multivatiate case, with more than one indicator variable, the parameters of a mixture model are the means, variances, and covariances between the indicators (which can be standardized to obtain correlations). These parameters can be estimated freely, or set to be constrained across classes.

The technique known as *latent profile analysis (LPA)* is a special case of the mixture model, which assumes conditional independence of the indicators. Conditional independence means that, after class membership is accounted for, the covariances/correlations between indicators are assumed to be zero. This can be conceived of as a restricted mixture model with covariances fixed to zero. In some cases, such constraints will be inappropriate; for example, when the cohesion between indicators is expected to differ between classes. As an example, a mixture model analysis of ocean plastic particles found two classes of particles based on length and width: a class of smaller particles with a high correlation between length and width, meaning that these particles were approximately round or square in shape, and a class of larger particles with a low correlation between length and width, meaning that these particles were heterogenous in shape. From a theoretical perspective, this makes sense, because the smaller particles have been ground down to a more uniform shape by the elements.

It is also possible to estimate a mixture model based on latent indicators. This means that, within each class, one or more continuous latent variables are estimated based on the observed indicators. Categorical latent variable membership is then estimated based on these continuous latent variables. A common application of this approach is in longitudinal research, where the indicators reflect one construct assessed at different time points. Examples of this approach include *growth mixture models* (GMM) and *latent class growth analyses* (LCGA). These techniques estimate a latent growth model to describe individual trajectories over time. The growth mixture model is a latent class model where the parameters that indicate class membership are the intercepts and variances (and typically covariances) of the latent growth variables, e.g., a latent intercept and slope. This technique assumes that individuals within a class can have heterogenous trajectories. If the variance of the growth parameters is fixed to zero, it is known as a latent class growth analysis. This latter approach assumes that all individuals within a class share the same identical trajectory, and that any variance in the indicators not explained by the class-specific latent trajectories is due to residual error variance.

The term latent class analysis originally referred to cases where the observed indicators were categorical. Nowadays, it is more commonly used as an umbrella term. To prevent ambiguity, the special case where indicators are of binary or ordinal measurement level might be described as *latent class analysis with ordinal indicators*. Latent class models with ordinal indicators are parameterized differently from mixture models. One common parameterization assumes that each categorical variable reflects an underlying standard normal distribution. The parameters are "thresholds" that correspond to quantiles of a standard normal distribution (with $N(\mu = 0, \sigma = 1)$). These thresholds are estimated based on the proportion of individuals in each of the response categories of the indicator variable. For example, a binary indicator has a single threshold that distinguishes the two response categories. If responses are distributed 50/50, then the corresponding threshold would be $t_1 = 0.00$. If the responses are distributed 60/40, then the resulting threshold would be $t_1 = 0.25$.

This paper will primarily focus on mixture models and special cases thereof, although most of the suggested guidelines are applicable to all latent class analyses.

Use cases for latent class analysis

There are several use cases for which latent class analyses are suitable. One example is to test a theory that postulates the existence of a categorical latent variable. For example, *identity status theory* posits that, at any given point in time, adolescents reside in one of four identity statuses. Latent class analysis can be used to identify these four statuses based on observed indicators (e.g., self-reported identity exploration and commitment). If results indicate that the data are better described by a different number of classes, or that the four-class solution does not correspond to the predicted pattern of responses on the indicators, then the theory may be called into question.

Another use case is unsupervised learning; when the goal is to restore unobserved class membership based on observed indicators, or to classify individuals. For example, a mixture model can be used as a diagnostic aid when several clinical indicators can be used to distinguish between a fixed number of physical (Baughman, Bisgard, Lynn, & Meade, 2006) or mental (Wu, Woody, Yang, Pan, & Blazer, 2011) health problems. The example of shoe size is a rudimentary illustration of this type of application.

LCA can be used as a descriptive analysis where a researcher wishes to describe their sample and identify a few prototypes based on many variables. As an example, if a survey among all Dutch academics was carried out with the goal of bringing about a funding reform, LCA could be used to discover the types of publications that get funding.

With LCA, our goal could be to inductively identify the number of classes. For instance, if we believe that a variable represents a group, but don't know how many groups there are, LCA may be an appropriate technique to answer this question. For example, Hopfer, Tan, and Wylie (2014) studied the substance use, sexual behavior, and mental health status of urban population in Winnipeg, Canada. The underlying assumption was that there were different risk profiles, but their number was not known. From a collection of indicators, the LCA provided evidence that there may be four distinct risk profiles in the Winnipeg area.

Another application of LCA is to classify individuals. In a peer harassment study, Giang and Graham (2008) used latent class analysis to classify over 2,000 sixth grade students into aggressor and victim latent classes. The five-class solution comprised of classes of victims, aggressors, and socially adjusted students. For instance, it revealed that there were two types of victims: highly-victimized aggressive-victims and highly-aggressive aggressive-victims.

LCA is also appropriate when we wish to identify indicators that capture classes well. High quality indicators are strongly related to the latent variable and lead to good class separation. This relationship of high quality indicators to the latent variable is reflected in very high or very low conditional response probabilities. For a simulation study exploring the effects of indicator quality on LCA, see Geiser and Wurpts (2014). Therefore, one could use conditional response probabilities for each item to assess its quality with regards to how well it helps separate the latent classes. From this, a theory about the selection of indicators could be informed.

An extension of LCA is that containing covariates which can be used to predict class membership. In this approach, we not only model the latent class variable based on indicators, but we also relate the class membership to other explanatory variables (Vermunt (2017)). An example of using covariates comes from Nozadi et al. (2016) who applied LCA to identify the probability of children's membership to an anxiety class. The authors tested several covariates including children's age, sex, and accuracy scores. Age and sex were not found to be related to the children's latent class membership, hence these covariates were excluded from the analysis. Accuracy scores were related to probabilities of being in anxiety and attention- anxiety classes and therefore this covariate was kept as a valuable predictor.

When our interest is the prediction of one or more outcomes, LCA can be used to construct latent classes as categorical predictors. Lanza, Tan, and Bray (2013) demonstrated how LCA can be used to classify adolescents into depression classes, and subsequently these classes can be used to predict smoking, grades, and delinquency. The study showed that the outcomes predicted by class membership can be binary (regular smoking), continuous (grades) or count (delinquency).

In addition to these applications, LCA can be used for dimensionality reduction as the resulting groups summarize response patterns on a large number of indicators. For example, MacGregor et al. (2021) investigated symptom profiles among injured U.S. military personnel. They used fifteen dichotomous items from the Post-Deployment Health Assessment survey as LCA indicators. Combinatorics informs us that fifteen dichotomous items have 2^{15} or $32,768$ unique symptom combinations. Perhaps for this reason, MacGregor et al. (2021) incorporated LCA as a method of dimensionality reduction. A five class solution was found to have the best fit according to both statistical criteria and clinical interpretability.

Finally, LCA can be used to deal with data which violate certain assumptions. As discussed in the shoe size example, LCA can deal with violations of normality. In fact, LCA assumes the population distribution in a non-normal mixture of K normal distributions, and it can discover the value of K, i.e. generate a K-class solution.

Best practices

In estimation

The best practices in estimation, as outlined in Table **??**, are rooted in existing recommendations for best practices for estimating specific sub-types of latent class analyses, including latent class growth analysis (Van De Schoot et al., 2017) and latent class analysis with ordinal indicators (e.g., Nylund-Gibson & Choi, 2018). These were generalized to be more relevant to all types of latent class analyses, and updated to current best practices, as explained below.

Examine observed data. Examining observed data is essential for any analysis as it may reveal patterns and violations of assumptions that had not been considered prior to data collection. Special attention should be paid to level of measurement of the indicators. Finite Gaussian mixture models (including LPA) are only suitable for continuous variables. Indicators with an ordinal level of measurement are likely to violate the assumption of within-class normal distributions of mixture models (see Vermunt, 2011). Personal experience consulting on latent class analyses and moderating the `tidyLPA` Google group suggest that the misapplication of mixture models to ordinal (e.g., Likert-type) indicators is the most common source of user error. Whereas it has been argued that some parametric methods are robust when scales with 7+ indicators are treated as continuous (e.g., Norman, 2010), this certainly does not imply that all methods are. It is certainly unlikely that such ordinal variables can be treated as a *mixture* of multiple normal distributions. The problem becomes egregious when the number of classes estimated equals or exceeds the number of categories; in this case, each class-specific mean could describe a single response category, and a class-specific variance component would be nonsensical. In sum, Likert-type scales are rarely suitable for mixture modeling; latent class analysis with ordinal indicators is more appropriate.

Relatedly, a recent publication claimed that an assumption of mixture models is that observed indicators are normally distributed (Spurk, Hirschi, Wang, Valero, & Kauffeld, 2020). This is incorrect. When the number of classes is greater than one, mixture models assume that the observed indicators are a mixture of multiple (multivariate) normal distributions. In our shoe size example, it can be seen that the population distribution comprised of two normal distributions. When examined visually, the population distribution is evidently bimodal. The Shapiro-Wilk normality test ($W = 0.971$, $p < 0.05$) rejects the null hypothesis that the sample comes from a normally distributed population. Yet, this is a prototypical example of a mixed population distribution where LCA can discover latent groups. If the population distribution were instead normal, there would be no classes to extract as the whole population would belong to a single class.

Extensive descriptive statistics (including the number of unique values, variance of categorical variables, and missingness; see next paragraph) can be obtained using the function `tidySEM::descriptives(data)`. Note, however, that sample-level descriptive statistics are of limited value when the goal of a study is to identify sub-samples using latent class analysis. Plots (density plots for continuous variables, and bar charts for categorical ones) may be more diagnostic. Note that density plots can also aid in the choice of the number of classes, as further explained in the section on visualization. Descriptive statistics and plots can be relegated to online supplements, provided that these are readily accessible (consider using a GitHub repository as a comprehensive public research archive, as explained in Van Lissa et al., 2021).

Handling missing data. Previous work has emphasized the importance of examining the pattern of missing data and reporting how missingness was handled (Van De Schoot et al., 2017). Three types of missingness have been distinguished in the literature (Rubin, 1976): Missing completely at random (MCAR), which means that missingness is random; missing at random (MAR), which means that missingness is contingent on the *observed* data (and can thus be accounted for); and finally missing not at random (MNAR), which means that missingness is related to unobserved factors. It is possible to conduct a so-called "MCAR" test, for example the non-parametric MCAR test (Jamshidian & Jalal, 2010). But note that the name "MCAR test" is somewhat misleading, as the null-hypothesis of this test is that the data are not MAR, and a significant test statistic indicates that missingness is related to the observed data (MAR). A non-significant test statistic does not distinguish between MCAR or MNAR. As Little's classic MCAR test relies on the comparison of variances across groups with different patterns of missing data, it assumes normality (Little, 1988). This assumption is tenuous in the context of latent class analysis. A non-parametric MCAR test, as provided by Jamshidian and Jalal, may be more suitable

(Jamshidian & Jalal, 2010). Unfortunately, this test was removed from the central R-repository CRAN due to lack of maintenance. For this tutorial, I have re-implemented it in the `mice` package as `mice::mcar()`, with a fast backend in C++ and new printing and plotting methods.

While we concur that investigating missingness is due dilligence, it is important to emphasize that missingness is adequately handled by default in many software packages for latent class analyses, such as Mplus, and `OpenMx` which is the backend of `tidySEM`. These packages use Full Information Maximum Likelihood (FIML) estimation, which makes use of all available information without imputing missing values. FIML is a best-practice solution for handling missing data; on par with multiple imputation (Lee & Shi, 2021). FIML estimation assumes that missingness is either MCAR or MAR. Thus, one would typically proceed with FIML regardless of the outcome of an MCAR test. Although FIML does not, by default, handle missingness in exogenous variables - all indicator variables in latent class analysis are endogenous, so this is not a concern.

Multiple imputation is less suitable to latent class analyses for two reasons. First, because latent class analyses are often computationally expensive, and conducting them on multiple imputed datasets may be unfeasible. Second, because there is no straightforward way to integrate latent class analysis results across multiple datasets. To conclude; our recommendation is to inspect missingness (e.g., using `mice::MCAR()`) and report the proportion of missingness per variable (e.g., using `tidySEM::descriptives()`), before proceeding with FIML. One minor concern is that the K-means algorithm, which `tidySEM` uses for determining starting values, is *not* robust to missing values. When it fails, `tidySEM` automatically switches to hierarchical clustering, unless the user specifies a different clustering algorithm or uses manual starting values.

Alternative model specifications. In order to aid researchers working with latent trajectory models, Van De Schoot et al. (2017) developed a protocol called Guidelines for Reporting on Latent Trajectory Studies (GRoLTS). Two of the GRoLTS guidelines (namely, 6a and 6b) refer to considering alternative model specifications. Both are discussing specific cases in latent trajectory model specification. The first is about whether the variance of the growth parameter is estimated freely or fixed, and the second is about whether conditional independence is assumed as the researcher might also want to free-up the variance-covariance structure. In LCA, there are also many different ways to specify the model. Means, variances, and covariances between the indicators can either be constrained across classes, or estimated freely. Researchers using LCA should transparently report their chosen parametrization as well as discuss different parametrizations that were tested in the process.

Different types of latent class models have different parameters. For example, mixture models and latent profile analyses typically have class-specific means, variances, and covariances. Latent growth analyses have the same parameters, but with respect to the latent growth variables. Latent class analyses with ordinal indicators have thresholds. All of these parameters can be freely estimated across classes, constrained to be equal across classes, or fixed to a certain value (e.g., to zero). The total number of parameters thus scales with the number of estimated classes. Consequently, latent class analyses have a potentially very high number of parameters. As any of these parameters could be misspecified, it is important to consider alternative model specifications. However, alternative model specifications may be approached differently depending on whether an analysis is data driven (exploratory), or theoretically driven (confirmatory). This distinction has remained underemphasized in prior writing.

Prior literature on latent class analysis has emphasized exploratory applications of the method (see Nylund, Asparouhov, & Muthén, 2007). In exploratory analyses, a large number of models are typically estimated in batch, with varying numbers of classes and model specifications. The "correct" model specification is then determined based on a combination of fit indices, significance tests, and interpretability. For latent profile analysis, the function `tidySEM::mx_profiles(classes, variances, covariances)` largely automates this process. The argument `classes` indicates which class solutions should be estimated (e.g., 1 through 6). The argument `variances` specifies whether variances should be `"equal"` or `"varying"` across classes. The argument `covariances` specifies whether covariances should be constrained to `"zero"`, `"equal"` or `"varying"` across classes. The means are free to vary across classes by default, although the more general function `tidySEM::mx_mixture()` could be used to circumvent this. After all models have been estimated, the function `tidySEM::table_fit()` can be used to obtain a model fit table suitable for determining the optimal model according to best practices. Note however that this table does not include the bootstrapped likelihood ratio test (BLRT) by default, because this test is very computationally expensive. It is recommended to use the function `tidySEM::BLRT()` to compare a shortlist of likely candidate models based on other fit indices. More on fit indices can be found in the Model fit indices subsection of this paper.

Confirmatory analyses typically require less comprehensive alternative model specifications. For example, in the context of preregistered analyses, the main models of interest may have been specified a priori. Even in case of confirmatory LCA, the theoretical model could be compared to a few others to contextualize it.

Software

Many software packages are available for the estimation of latent class analyses. Some of these packages have limited functionality, or implement specific innovations. Other packages implement latent class analyses in the context of a more flexible structural equation modeling framework. The most notable examples of the latter are the commercial programs Mplus and Latent GOLD, and the free open source R-package OpenMx. The commercial packages stand out because they offer relatively user-friendly interfaces and implement sensible defaults for complex analyses, including latent class analysis. This lowers the threshold for applied researchers to adopt such methods. Commercial software also has several downsides, however. One such downside is that use of the software is restricted to those individuals and institutions who can afford a license. A second downside is that the source code, being proprietary, cannot be audited, debugged, or enhanced by third parties. This incurs the risk that mistakes in the source code may go unnoticed, and curbs progress as software developers cannot add new functionality.

Conversely, the free open source program OpenMx is very flexible, but not very user-friendly. We directly address this limitation using the `tidySEM` R-package. New functionality in `tidySEM` seeks to lower the threshold for latent class analysis using `OpenMx`. It adheres to best practices in estimation and reporting, as described in this paper. The user interface is simple, making use of the model syntax of the widely used `lavaan` R-package. This syntax offers a human-readable way to specify latent variable models. Minor enhancements are made to simplify the specification of latent class analysis.

Because of the limitations of the aforementioned tools, we set out to develop a free tool that provides sensible defaults and is easy to use, but provides the option to access and modify all of the model inputs (i.e., low barrier, high ceiling). `tidySEM` interfaces with existing tools, and is able to translate between what existing tools are capable of and what researchers and analysts carrying-out person-oriented analyses would like to specify. Furthermore,`tidySEM` facilitates fully-reproducible analyses and contributes to open science.

Best practices in estimation

Algorithm. Mixture model parameters and model fit statistics can be estimated in a variety of ways. The choice of the estimator depends on the presence of missing values, sample size, number of indicators, and available computational resources (Weller, Bowen, & Faubert, 2020). A commonly used technique is maximum likelihood (ML) estimation with the expectation-maximization (EM) algorithm as a local optimizer. Imagine we are estimating two parameters, e.g. the class-specific means μ_c on a continuous indicator (ignoring the variance for now). The EM algorithm will attempt to find a combination of values for these two parameters that maximizes the likelihood (LL) of all observed data. In practice, instead of maximizing LL, often $-2 * LL$ is minimized, as this offers computational advantages. We can think of this optimization problem as a three-dimensional landscape: The X and Y dimensions are determined by the class-specific means, so $X = \mu_1$ and $Y = \mu_2$ - and the Z-dimension is determined by $Z = -2 * LL$. The optimizer must find the deepest "valley" in this landscape, which reflects the combination of μ_1 and μ_2 that maximizes the likelihood of the data. The EM optimizer behaves somewhat like a marble, dropped in this landscape. It is dropped at some random point in space, and will roll into the nearest valley. The problem is that, once EM rolls into a valley, it will settle on the bottom of that valley (this is known as "convergence"). It cannot climb out again. Thus, if their are multiple valleys, the risk is that the optimizer gets stuck in a shallower valley (a "local optimum"), and never discovers the deepest valley (the "global optimum", or best solution). One solution to this problem is to drop many marbles at random places, compare their final $-2 * LL$ values, choose the solution with the lowest $-2 * LL$, and make sure that several marbles replicated this solution. This is the "random starts" approach.

One problem with the random starts approach is that it is computationally expensive to run this many replications. Moreover, because the algorithm begins with random starting values, many of the marbles are likely to be very far away from a "good enough" solution. Two innovations may improve the estimation procedure. The first is that, instead of picking random starting values, a "reasonable solution" may be used for the starting values. For example, if we assume that the different classes are likely to have different mean values on the indicators, then the K-means clustering algorithm can be used to determine these cluster centroids. We can compute the expected values of all model parameters by treating the K-means solution as a known class solution, and use these as starting values for a mixture model. One remaining concern is that this approach may result in starting values close to a local optimum, and that the EM algorithm will thus never find the global optimum. A second innovation addresses this concern.

Instead of using EM, it is possible to use an optimizer that can climb out of a valley.

Simulated annealing iteratively considers some “destination” in the landscape, and compares its likelihood to the current one. If the destination likelihood is higher, the estimator moves there. If the destination likelihood is *lower*, the estimator still moves there occasionally, based on probability. This latter property allows it to escape local optima, and find the global optimum.

By default, `tidySEM` employs this solution of deriving starting values using K-means clustering, and identifying the global optimum solution using simulated annealing. Once a solution has been found, simulated annealing is followed up with a short run of the EM algorithm, as EM inherently produces an asymptotic covariance matrix for the parameters that can be used to compute standard errors. Note that these defaults can be manually overridden.

One recent paper suggested maximum likelihood with robust standard errors should be used when the observed indicators are not normally distributed (Spurk et al., 2020). This statement is incorrect, and may lead readers to believe that they must use commercial software, as robust maximum likelihood is currently only implemented in Mplus and latentGOLD. As explained before, mixture modeling assumes that observed data are a mixture of (multivariate) normal distributions; thus, the observed indicators will likely not be normally distributed.

Class enumeration. LCA can be done in an exploratory or in a confirmatory fashion. In exploratory LCA, a sequence of models is fitted to the data with each additional model estimating one more class than the previous model. These models are then compared and the best solution is selected as the final class solution. In some cases, prior theory can inform the researcher about the number of classes to expect. Even in such confirmatory LCA cases, it is nonetheless useful to know if the theoretical model is markedly better than those with differing numbers of classes. Therefore, it may always be useful to compare different class solutions.

From a sequence of models, the final class solution is chosen based on both theoretical and statistical criteria. Theory should drive the selection of indicator variables, inform the expectations and reflect on the findings. In addition to this, there are several statistical criteria to consider in model selection. These include but are not limited to likelihood ratio tests, information criteria, and the Bayes factor (Weller et al., 2020).

Relative model fit can be examined using the likelihood ratio test. This is only appropriate when the two models we wish to compare are nested. The likelihood ratio test statistic is computed as the difference in maximum log likelihoods of the two models, with the new degrees of freedom being the difference in their degrees of freedom. This statistic also follows the χ^2 distribution. Similar to the LR χ^2 goodness-of-fit test, we want the test statistic to be non-significant in order to give support to the simpler model. The likelihood ratio test can only compare two nested models at a time (Lanza, Bray, & Collins, 2003).

Model Fit Indices. Fit indices typically used for determining the optimal number of classes include the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both information criteria are based on the -2*log likelihood (which is lower for better fitting models), and add a penalty for the number of parameters (thus incentivizing simpler models). This helps balance fit and model complexity. The lower the value of an information criterion, the better the overall fit of the model. The BIC applies a stronger penalty for model complexity that scales logarithmically with the sample size. The literature suggests the BIC may be the most appropriate information criterion to use for model comparison (Nylund-Gibson & Choi, 2018). Both the AIC and the BIC are available in the `tidySEM` output.

Information criteria may occasionally contradict each other, so it is important to identify a suitable strategy to reconcile them. One option is to select a specific fit index before analyzing the data. Another option is to always prefer the most parsimonious model that has best fit according to any of the available fit indices. Yet another option is to incorporate information from multiple fit indices using the analytic hierarchy process

(Akogul & Erisoglu, 2016). Finally, one might make an elbow plot and compare multiple information criteria (for an example see Nylund-Gibson & Choi, 2018).

Another common test of model fit is the likelihood ratio χ^2 goodness-of-fit test. However, this test is not implemented in `tidySEM`.

LCA studies commonly report -2*log likelihood of the final class solution. This is a basic fit measure used to compute most information criteria. However, since log likelihood is not penalized for model complexity, it will continuously fall with the addition of more classes.

An alternative is using the bootstrapped likelihood ratio test which can be run using `tidySEM::BLRT()`. Currently, this test is computationally expensive and can be slow on most computers. A faster version of this test, namely an implementation of *the lazy bootstrap* (Kollenburg, Mulder, & Vermunt, 2018) to `tidySEM` is being developed.

Classification Diagnostics. Best models will divide the sample into subgroups which are internally homogeneous and externally distinct. Classification diagnostics give us a way to assess the degree to which this is the case. They are separate from model fit indices as a model can fit the data well but show poor latent class separation (Masyn, 2013). Classification diagnostics should not be used for model selection, but they can be used to disqualify certain solutions because they are uninterpretable. Interpretability should always be a consideration when considering different class solutions (Nylund-Gibson & Choi, 2018).

Three important classification diagnostics provided by `tidySEM` are are *the minimum* and *maximum percentage of the sample assigned to a particular class*, *the range of the posterior class probabilities by most likely class membership*, and *entropy*. All three are based on posterior class probabilities.

The posterior class probability is a measure of classification uncertainty which can be computed for each individual, or averaged for each latent class. When the posterior class probability is computed for each individual in the dataset, it represents each person's probability of belonging to each latent class. For each person, the highest posterior class probability is then determined and the individual is assigned to the corresponding class. We want each individual's posterior class probabilities to be high for one and low for the remaining latent classes. This is considered a high classification accuracy and means that the classes are distinct. To obtain posterior class probabilities, run `tidySEM::class_prob()`. This function produces an output comprised of several elements. Namely:

`$sum.posterior` is a summary table of the posterior class probabilities indicating what proportion of the data contributes to each class.

`$sum.mostlikely` is a summary table of the most likely class membership based on the highest posterior class probability. From this table, we compute the minimum and maximum percentage of the sample assigned to a particular class, , i.e. **n_min** (the smallest class proportion based on the posterior class probabilities) and **n_max** (the largest class proportion based on the posterior class probabilities). We are especially interested in **n_min** as if it is very small and comprised of few observations, the model for that group might not be locally identified. Estimating LCA parameters on small subsamples might lead to bias in the results. Therefore, we advise caution when dealing with small classes.

`$mostlikely.class` is a table with rows representing the class the person was assigned to, and the columns indicating the average posterior probability. The diagonal represents the probability that observations in each class will be correctly classified. If any of the values on the diagonal of this table is low, we might consider not to interpret that solution. In `tidySEM` we use the diagonal to compute the range of the posterior class probabilities by most likely class membership which consists of the lowest class posterior probability (**prob_min**), and the highest posterior probability (**prob_max**). Both **prob_min** and **prob_max** can be used to (dis)qualify certain class solutions, and are a convenient way to summarize class separation in LCA. We want both **prob_min** and **prob_max** to be high as that means that for all classes the people who were assigned to that class have a high probability of being there. **prob_min** is especially important as it can diagnose if there is a class with low posterior probabilities which could make one reconsider that class solution.

`$avg.mostlikely` contains the average posterior probabilities for each class, for the subset of observations with most likely class of 1:k, where k is the number of classes.

`$individual` is the individual posterior probability matrix, with dimensions n (number of cases in the data) x k (number of classes). Individual class probabilities are often useful for researchers who wish to do follow up analyses.

Entropy is a summary measure of posterior class probabilities across classes and individuals. It ranges from 0 (model classification no better than random chance) to 1 (perfect classification). As a rule of thumb, values above .80 are deemed acceptable and those approaching 1 are considered ideal. An appropriate use of entropy is that it can disqualify certain solutions if class separability is too low or if one of the latent classes is too small to be meaningful or to calculate descriptive statistics. Entropy was not built for nor should it be used for model selection during class enumeration (Masyn, 2013).

n_min, **n_max**, **prob_min**, **prob_max**, and **entropy** and can be obtained using `tidySEM::table_fit()`.

Interpreting Class Solution

An important outcome of LCA are conditional item probabilities, also known as class-specific item probabilities (Masyn, 2013), conditional response or conditional solution probabilities (Geiser, 2012). They indicate the probability of an item being endorsed given that the observation belongs to a particular latent class. Conditional item probabilities can be obtained using `tidySEM::table_prob()` If a particular item is endorsed by two or more classes at markedly different rates, it is said to discriminate well between the classes and is consequently considered a good indicator. Classes are considered highly homogeneous with respect to an item when for a particular item there is a distinct difference in conditional item probabilities for two or more classes. For instance, if an item is endorsed below 30% for one class and above 70% for another class, the classes have high homogeneity with respect to this item (Masyn, 2013). Conditional item probabilities are the analogue of mean and standard deviation when the indicators are binary or ordinal.

A problem which can occur is that of inadmissible solutions. With binary indicators, LCA is modelling a cross-table with all the predictors. The problem with such cross-tables is that they will often contain empty cells, i.e. combinations of responses that never occur together. This problem is reflected by extreme conditional item probabilities (as in exactly 0 or 1). Such boundary parameter estimates could indicate that the solution is invalid (Geiser, 2012). Boundary parameter estimates can also happen with continuous indicators. For instance, if we have a zero-inflated normal distribution and a two class solution, one class might have the mean of zero and its standard deviation cannot be determined since there is little variance. This too could be a sign of an invalid solution, warn us that too many classes were extracted, or indicate a local optimum (Geiser, 2012).

Label switching. The final class solution will usually discover and enumerate several classes. The class ordering however is completely arbitrary. The class labeled as Class 1 in one solution may become Class 2 or Class 3 in another model, even when the only difference between the models is in their starting values. Label switching is something to be mindful of when comparing different LCA models (Masyn, 2013).

The order of clusters is nondeterministic when using K-means in `tidySEM`. Therefore label switching is still a consideration. A simple solution to this is setting a random seed number one line prior to fitting the model. We advise `tidySEM` users to always do so in order to circumvent label switching.

Class names should be chosen to accurately reflect group membership. Overly simplified and generalized class names may prove misleading to both audiences and researches alike leading to what is known as a naming fallacy (Weller et al., 2020).

Best Practices in Reporting

Among studies using LCA, reporting practices vary significantly (Weller et al., 2020). Various authors have tried to better and standardize ways of reporting LCA (e.g. Masyn, 2013; Weller et al., 2020), but more work is needed. Van Lissa et al. (2020) developed WORCS, a workflow for open reproducible code in science. WORCS consists of step-by-step guidelines for research projects based on the TOP-guidelines developed by Nosek et al. (2015). WORCS workflow can be easily implemented in R in form of an R package which facilitates preregistration, article drafting, version control, citation and formatting, among others (Van Lissa et al., 2020).

TOP-guidelines emphasise the use of comprehensive citation (including referencing the software used in the analysis), as well as code and data sharing wherever possible (Nosek et al., 2015). Van Lissa et al. (2020) suggest sharing synthetic data in case the original data cannot be shared, and provide functions to generate such synthetic data. Ideally, the entire research project is made reproducible so that others may download it and reproduce it with just one click; for guidance, see Van Lissa et al. (2020).

As the open science movement is gaining momentum, researchers are becoming increasingly aware how important it is that analyses can be reproduced and audited. In line with open science principles, one of the suggested reporting standards relates to reproducible code. In this context, it is important to note that user-friendly methods for estimating latent class analyses have predominantly been available in commercial software packages (e.g., *Mplus* and *Latent GOLD*). A potential downside of commercial software is that it restricts the ability to reproduce analyses to license holders, and prevents auditing research because the underlying source code is proprietary. To overcome these limitations, the present paper introduces new user-friendly functions in the `tidySEM` R-package that can be used to estimate a wide range of latent class analysis models using the free, open-source R-package `OpenMx`. The reporting guidelines described in this paper are adopted in `tidySEM` by default.

The `tidySEM` R-package thus makes advanced mixture modeling based on best practices widely accessible, and facilitates the adoption of the estimation and reporting guidelines described in this paper.

Best Practices in Visualization

Tutorial

References

Akogul, S., & Erisoglu, M. (2016). A Comparison of Information Criteria in Clustering Based on Mixture of Multivariate Normal Distributions. *Mathematical and Computational Applications*, *21*(3), 34. https://doi.org/10.3390/mca21030034

Baughman, A. L., Bisgard, K. M., Lynn, F., & Meade, B. D. (2006). Mixture model analysis for establishing a diagnostic cut-off point for pertussis antibody levels. *Statistics in Medicine*, *25*(17), 2994–3010. https://doi.org/10.1002/sim.2442

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 381–396. https://doi.org/10.1109/34.990138

Geiser, C. (2012). *Data Analysis with Mplus*. Guilford Press.

Geiser, C., & Wurpts, I. (2014). Is adding more indicators to a latent class analysis beneficial or detrimental? Results of a Monte Carlo study. *Frontiers in Psychology: Quantitative Psychology and Measurement*, *5*. https://doi.org/https://doi.org/10.3389/fpsyg.2014.00920

Giang, M. T., & Graham, S. (2008). Using latent class analysis to identify aggressors and victims of peer harassment. *Aggressive Behavior*, *34*(2), 203–213. https://doi.org/10.1002/ab.20233

Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (2015). *Handbook of Cluster Analysis*. 28.

Hopfer, S., Tan, X., & Wylie, J. L. (2014). A Social Network–Informed Latent Class Analysis of Patterns of Substance Use, Sexual Behavior, and Mental Health: Social Network Study III, Winnipeg, Manitoba, Canada. *American Journal of Public Health*, *104*(5), 834–839. https://doi.org/10.2105/AJPH.2013.301833

Jamshidian, M., & Jalal, S. (2010). Tests of Homoscedasticity, Normality, and Missing Completely at Random for Incomplete Multivariate Data. *Psychometrika*, *75*(4), 649–674. https://doi.org/10.1007/s11336-010-9175-3

Kollenburg, G. H. van, Mulder, J., & Vermunt, J. K. (2018). *The Lazy Bootstrap. A Fast*

Resampling Method for Evaluating Latent Class Model Fit. 23.

Lanza, S. T., Bray, B. C., & Collins, L. M. (2003). An Introduction to Latent Class and Latent Transition Analysis. In *Handbook of Psychology: Research Methods in Psychology* (2nd ed., Vol. 2, pp. 690–712). John Wiley & Sons.

Lanza, S. T., Tan, X., & Bray, B. C. (2013). Latent Class Analysis With Distal Outcomes: A Flexible Model-Based Approach. *Structural Equation Modeling : A Multidisciplinary Journal*, *20*(1), 1–26. https://doi.org/10.1080/10705511.2013.742377

Lee, T., & Shi, D. (2021). A comparison of full information maximum likelihood and multiple imputation in structural equation modeling with missing data. *Psychological Methods*, No Pagination Specified–No Pagination Specified. https://doi.org/10.1037/met0000381

Little, R. J. A. (1988). A Test of Missing Completely at Random for Multivariate Data with Missing Values. *Journal of the American Statistical Association*, *83*(404), pp. 1198–1202. https://doi.org/10.2307/2290157

MacGregor, A. J., Dougherty, A. L., D'Souza, E. W., McCabe, C. T., Crouch, D. J., Zouris, J. M., … Fraser, J. J. (2021). Symptom profiles following combat injury and long-term quality of life: A latent class analysis. *Quality of Life Research*, *30*(9), 2531–2540. https://doi.org/10.1007/s11136-021-02836-y

Masyn, K. E. (2013). Latent Class Analysis and Finite Mixture Modeling. In *The Oxford Handbook of Quantitative Methods*: *Vol. 2: Statistical Analysis* (p. 551). Oxford University Press.

Norman, G. (2010). Likert scales, levels of measurement and the "laws" of statistics. *Advances in Health Sciences Education*, *15*(5), 625–632. https://doi.org/10.1007/s10459-010-9222-y

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., … Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425. https://doi.org/10.1126/science.aab2374

Nozadi, S. S., Troller-Renfree, S., White, L. K., Frenkel, T., Degnan, K. A., Bar-Haim, Y.,

. . . Fox, N. A. (2016). The Moderating Role of Attention Biases in understanding the link between Behavioral Inhibition and Anxiety. *Journal of Experimental Psychopathology*, *7*(3), 451–465. https://doi.org/10.5127/jep.052515

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. https://doi.org/10.1080/10705510701575396

Nylund-Gibson, K., & Choi, A. Y. (2018). Ten frequently asked questions about latent class analysis. *Translational Issues in Psychological Science*, *4*(4), 440–461. https://doi.org/10.1037/tps0000176

Rubin, D. B. (1976). Inference and Missing Data. *Biometrika*, *63*(3), 581–592. https://doi.org/10.2307/2335739

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). Mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal*, *8*(1), 289–317.

Spurk, D., Hirschi, A., Wang, M., Valero, D., & Kauffeld, S. (2020). Latent profile analysis: A review and "how to" guide of its application within vocational behavior research. *Journal of Vocational Behavior*, *120*, 103445. https://doi.org/10.1016/j.jvb.2020.103445

Van De Schoot, R., Sijbrandij, M., Winter, S. D., Depaoli, S., & Vermunt, J. K. (2017). The GRoLTS-Checklist: Guidelines for Reporting on Latent Trajectory Studies. *Structural Equation Modeling: A Multidisciplinary Journal*, *24*(3), 451–467. https://doi.org/10.1080/10705511.2016.1247646

Van Lissa, C. J., Brandmaier, A. M., Brinkman, L., Lamprecht, A.-L., Peikert, A., Struiksma, M. E., & Vreede, B. (2020). *WORCS: A Workflow for Open Reproducible Code in Science*. https://doi.org/10.17605/OSF.IO/ZCVBS

Van Lissa, C. J., Brandmaier, A. M., Brinkman, L., Lamprecht, A.-L., Peikert, A., Struiksma, M. E., & Vreede, B. M. I. (2021). WORCS: A workflow for open reproducible code in science. *Data Science*, *4* (1), 29–49. https://doi.org/10.3233/DS-210031

Vermunt, J. K. (2011). K-means may perform as well as mixture model clustering but may also be much worse: Comment on Steinley and Brusco (2011). *Psychological Methods*, *16* (1), 82–88. https://doi.org/10.1037/a0020144

Vermunt, J. K. (2017). Latent Class Modeling with Covariates: Two Improved Three-Step Approaches. *Political Analysis*, *18* (4), 450–469. https://doi.org/10.1093/pan/mpq025

Vermunt, J.K., Magidson, J., Lewis-Beck, M., Bryman, A., Liao, T.F., & Department of Methodology and Statistics. (2004). Latent class analysis. In *The Sage encyclopedia of social sciences research methods* (pp. 549–553). Sage. Retrieved from https://research.tilburguniversity.edu/en/publications/0caedd00-27c1-42bd-bb4e-d1dcb0864956

Weller, B. E., Bowen, N. K., & Faubert, S. J. (2020). Latent Class Analysis: A Guide to Best Practice. *Journal of Black Psychology*, *46* (4), 287–311. https://doi.org/10.1177/0095798420930932

Wu, L.-T., Woody, G. E., Yang, C., Pan, J.-J., & Blazer, D. G. (2011). Abuse and dependence on prescription opioids in adults: A mixture categorical and dimensional approach to diagnostic classification. *Psychological Medicine*, *41* (3), 653–664. https://doi.org/10.1017/S0033291710000954

Table 1

Observed group membership by estimated class membership.

Observed	Class 1	Class 2
Man	21	28
Woman	51	0

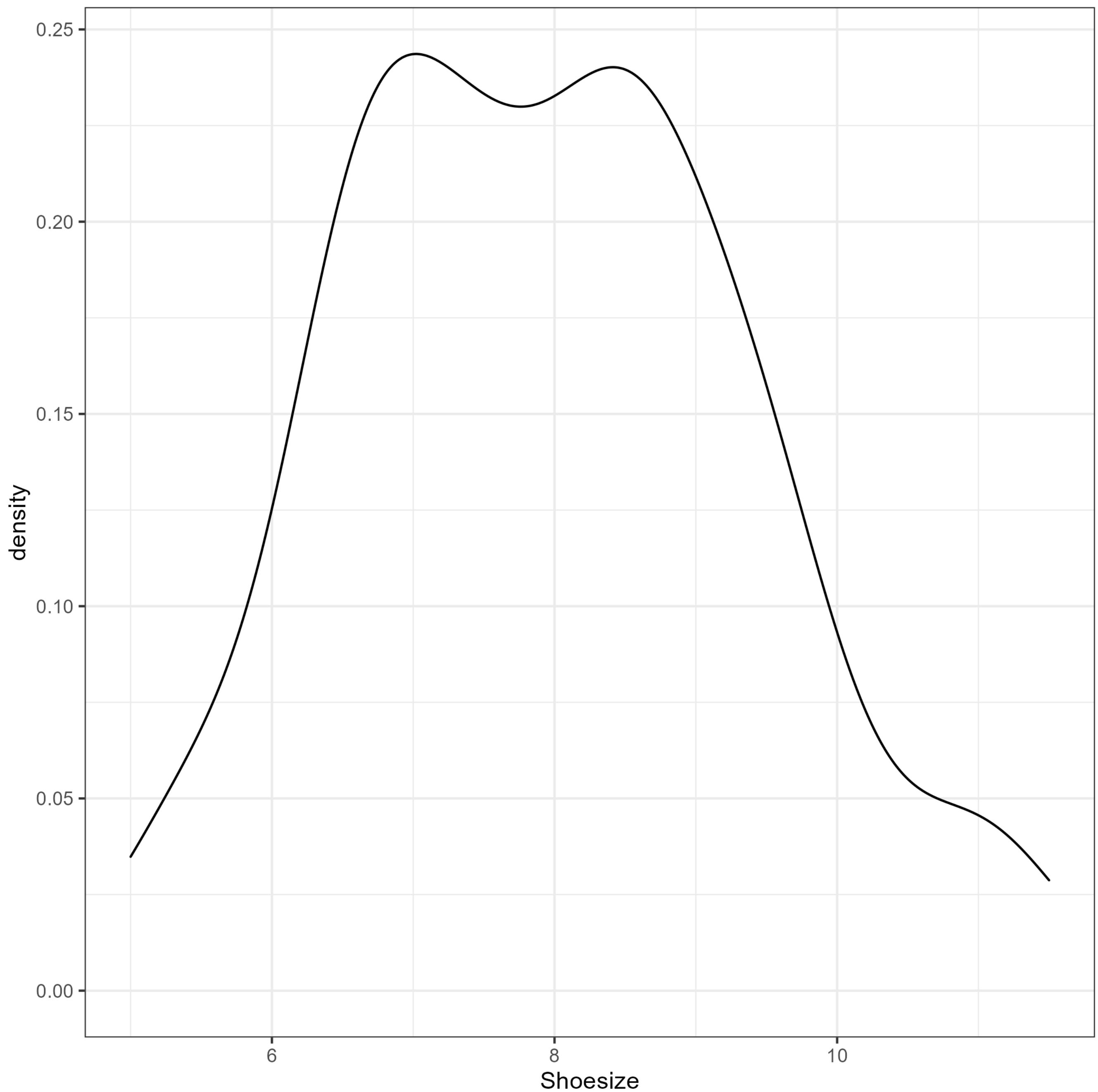

Figure 1. Kernel density plot of shoe sizes.