

Seeing the Impossible: Visualizing Latent Variable Models with **flexplavaan**

Dustin Fife¹, Steven Brunwasser¹, & Ed Merkle²

¹ Rowan University

² University of Missouri

Introduction

It is currently an unprecedented time in the social sciences; multiple scientific disciplines are reeling from a “replication crisis” (Camerer et al., 2018; Ioannidis, 2005; Pashler & Wagenmakers, 2012), new norms for credibility are becoming more prevalent (Nelson, Simmons, & Simonsohn, 2018; Nosek, Ebersole, DeHaven, & Mellor, 2018), and the push for open science is accelerating at a rapid pace (Nosek, Ebersole, DeHaven, & Mellor, 2018). Amidst this push for open science practices, some have called for greater use of visualization techniques (Fife, 2020b; Fife & Rodgers, 2019; Tay, Parrigon, Huang, & LeBreton, 2016). As noted by Tay, et al. (2016), “[visualizations]... can strengthen the quality of research by further increasing the transparency of data...” (p. 694). In other words, one of the best, and most efficient ways of making data analysis open and transparent is to display each and every data point through visualization techniques. This is particularly important in research applications where participant-level data cannot be shared.

Not only do visualizations adhere to the principles of openness and transparency, but they offer several additional advantages; they vastly improve encoding of information (Correll, 2015), they highlight model misfit (Healy & Moody, 2014), and they are an essential component in evaluating model assumptions (Levine, 2018; Tay et al., 2016). As such, we (as well as others, e.g., Fife, 2019, 2020b; Wilkinson & Task Force on Statistical Inference, 1999) recommend every statistical model ought to be accompanied by a graphic.

Unfortunately, this suggestion is easier said than done. While visualizing some statistical models is trivial (e.g., regressions, *t*-tests, ANOVAs, multiple regression), visualizing others is not. One particularly troublesome class of models to visualize is latent variable models (LVMs). While researchers routinely visualize LVMs with conceptual models (e.g., via path diagrams), visualizing statistical models is not so easy. The former visualizations are common, while the latter are not (Hallgren, McCabe, King, & Atkins, 2019). The reason statistical visualizations of LVM are not intuitive is because they rely on unobserved variables (Bollen, 1989). If the variables of interest are unobserved, how can we possibly visualize them?

Though it is not, at first glance, intuitive how to visualize unobserved variables, that

does not mean visualizing them is any less important. On the contrary, visualizing latent variables is more important because their presence is unobserved. In the following section, we elaborate on how LVMs are traditionally evaluated and why visualizations are particularly crucial. We then review previous approaches others have used for visualizing LVMs, and note their strengths and weaknesses. We then introduce our approach and the corresponding R package `flexlavaan`, which allows users to visualize both `lavaan` and `blavaan` objects in R. We then conclude with several examples that highlight how visualizations assisted in identifying appropriate statistical models.

Evaluating Model Fit in LVMs

The validity of LVM-based inferences assume structural models closely approximate real-world causal processes (Bollen, 2019; L. Hayduk, 2014). Unfortunately, evaluating model adequacy in LVMs is rife with obstacles. For one, misspecifications in any one part of the model can lead to biases that spread throughout the full systems of equations (Bollen, 2019). To combat this, LVM practitioners generally rely on global fit tests and approximate fit indices to evaluate the model adequacy (Jackson, Gillaspay Jr, & Purc-Stephenson, 2009).

Yet global fit indices themselves represent an obstacle to intelligent model evaluation. Models can yield desirable values (e.g., a non-significant χ^2 test), even when specific aspects of the model are substantially misspecified (Goodboy & Kline, 2017; L. Hayduk, 2014; Tomarken & Waller, 2003). In other words, a nonsensical model can still yield estimates that lead one to believe in their own statistical models. Moreover, in our experience, applied users often lack an intuitive understanding of what global fit statistics tell them about their models. Does a TLI of 0.95 mean we have established a strong theoretical foundation for a model? If RMSEA dips below 0.05, should we consider our model statistically/clinically significant?

While global fit measures may be useful in comparing models and/or identifying problematic model features, users instead rely on conventional cutoffs (e.g. Hu & Bentler, 1998) to determine whether a model is “adequate,” a practice that has received resounding criticism (Barrett, 2007; Chen, Curran, Bollen, Kirby, & Paxton, 2008; L. A. Hayduk, 2014; McIntosh, 2007). This lack of understanding is evident when applied users express confusion about why global fit indices provide conflicting assessments of model fit (Lai & Green, 2016).

A number of scholars have emphasized the importance of supplementing LVM global fit indices with local fit assessment, investigating the tenability of all specific model implications individually (Bollen, 2019; Goodboy & Kline, 2017; L. Hayduk, 2014; Thoemmes, Rosseel, & Textor, 2018; Tomarken & Waller, 2003, 2005). Local fit evaluation procedures (e.g., inspection of residual correlation matrices (Bollen, 1989), confirmatory tetrad analysis (Hipp & Bollen, 2003), and equation-based overidentification tests (Bollen, 2019)) can help identify individual model specifications that are inconsistent with the data and may give clues about effective remedial strategies (Bollen, 1989; Goodboy & Kline, 2017; Tomarken & Waller, 2003, 2005).

While local fit indices will certainly improve model evaluation, they too suffer from a number of problems. First, [steve...care to add some here].

Another problem shared by both global and local fit indices is that they are a highly *compressed* representations of both the data and the model. Many readers may be familiar with Anscombe’s quartet (reproduced in Figure 1). There is a many-to-one relationship between data and indices; very different types of data may yield identical statistics (in this case, correlations, slopes and intercepts). Some of these data patterns may be very poorly represented by the model. This problem is only exacerbated with LVMs simply because traditional algorithms compress the data at multiple levels (e.g., raw data are compressed into means/covariances, which are then compressed into model parameter estimates and/or indices that evaluate model fit). Put differently (and perhaps quite cynically), LVM is the process of compressing hundreds or thousands of datapoints into a handful (or less) of estimates that may or may not represent the data-generating process. When considered from this perspective, the most common practices of evaluating model fit seem primitive, at best.

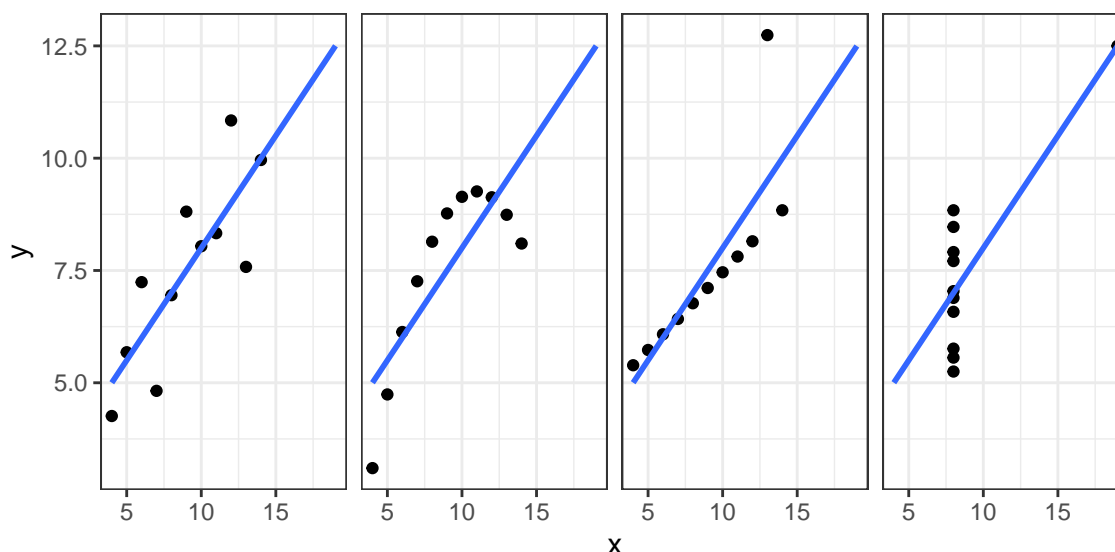


Figure 1. A reproduction of Anscombe’s quartet. Each plot has identical regression lines/correlations, even though the underlying data are vastly different.

The best defense against this compression is to rely on modeling evaluation strategies that evaluate *uncompressed* data. Perhaps the best (if not only) way to evaluate uncompressed data is through visualizations, particularly visuals that display raw data (Fife, 2020b).

Previous Approaches to Visualizing LVMs

There is sparse literature describing visual strategies for evaluating the adequacy of LVMs, at least when compared to the extensive literature discussing the merits of global fit tests and indices (e.g., Barrett, 2007; Chen et al., 2008; L. A. Hayduk, 2014; Hu & Bentler, 1998; McIntosh, 2007; Shi & Maydeu-Olivares, 2020; Smith & McMillan, 2001; Steiger, 2007). Additionally, while reporting of numeric fit indices is ubiquitous in LVM applications (Jackson et al., 2009), plots of data underlying LVMs are rare (Hallgren et

al., 2019). There are, however, some strategies for using visuals to evaluate the tenability of model assumptions, diagnose causal misspecifications, and select the best model from a group of competitors.

A common visual approach for identifying model-data discrepancies is to plot the distribution of the residuals of the covariances/correlation matrix (e.g., using stem-and-leaf plots or histograms; Bollen, 1989). This can aid in identifying specific components of the data that the model struggles to capture (Bollen, 1989; Bollen & Arminger, 1991), which might not be detected with global fit indices (Goodboy & Kline, 2017; Tomarken & Waller, 2003, 2005).

Unfortunately, these plots suffer from two major problems. First, the residuals in this case are themselves *compressed* estimates. As such, we might have a model that is poorly represented by linear correlations (e.g., if the data contain nonlinear relationships), but that problem would never be uncovered by studying residual plots.

A second problem with plots of correlation/covariance residuals is they are extremely limited in the amounts of misspecification they might reveal. For example, suppose we model a latent variable with three indicators (X_1 , X_2 , and X_3), but the data-generating model actually has X_3 associated with X_2/X_1 , but not an indicator of the latent variable (see Figure 2).¹ These two models will have the same implied variance/covariance matrix.² In other words, a stem and leaf plot will not signal any problems, despite problems existing.

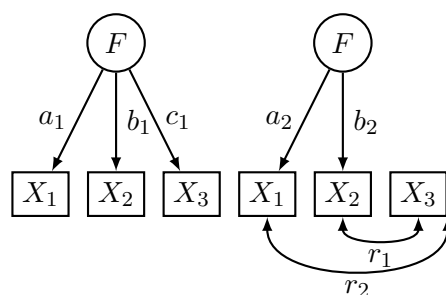


Figure 2. The model on the left is the user-specified model, while the model on the right is the data-generating model. These two models make very different theoretical statements, but have the same implied correlation between the variables.

Rather than visualizing aggregates in LVMs, the raw data themselves ought to be visualized. Bollen and Arminger (1991) developed methods for calculating raw and standard-

¹This problem is similar, but not identical to equivalent models (Lee & Hershberger, 1990; MacCallum, Wegener, Uchino, & Fabrigar, 1993). When two models are equivalent, they have identical variance/covariance matrices *and* identical degrees of freedom (DF). In this case, the two models do not have identical DF. However, the visuals we present do offer an intuitive approach for evaluating equivalent models. Two equivalent models with different latent models will have different visuals. Additionally, these visual will allow one to detect problems with equivalent models, which cannot be detected with traditional model-evaluation statistics.

²In the left model, the standardized relationship between X_1/X_3 is $a_1 \times c_1$, while in the right model it is r_2 . Likewise, the X_1/X_2 relationship is $a_1 \times b_1$ in the left model, while it is r_1 in the right model. The LVM machinery will attempt to make $a_1 \times c_1 = r(X_1, X_3)$, which is the same as setting $a_1 \times c_1 = r_2$ (and it will set $a_1 \times b_1$ to r_1).

ized individual case residuals (ICRs), which represent the difference between observed and model-estimated case values for outcome variables (Bollen & Arminger, 1991). These ICRs are then plotted to help locate outlying and influential observations. Pek and MacCallum (2011) demonstrated how diagnostic procedures commonly used in generalized linear models (e.g., Mahalanobis distance, generalized Cook’s D, and DFBETAs) can be applied to LVMs to detect influential cases with index plots. Flora, LaBrish, and Chalmers (2012) applied these diagnostic procedures and others specifically to factor analysis models, and Yuan and Hayashi (2010) used visualizations of Mahalanobis distance metrics to identify high-leverage cases and outliers. Open-source R packages, including **faoutlier** (Chalmers, 2017) and **influence.SEM** (Pastore & Pastore, 2018), have used these visualization procedures to show case influence on model fit (e.g., likelihood differences) and parameter estimations (e.g., generalized Cook’s D).

Asparouhov and Muthén (2017) proposed a method for extending the diagnosticity of ICRs to detect specific structural misspecifications. They demonstrated that plots of estimated factor scores against observed predictor variables can be used to detect unspecified nonlinear effects of the predictor on the latent outcome. Furthermore, they used ICR scatterplots to detect violations of local independence in a congeneric latent factor model. Finally, they demonstrated in a latent factor model how plotting predicted values for a reflective indicator against the observed indicator values could aid in uncovering unmodeled heterogeneity that could be better captured using a mixture model.

Raykov and Penev (2014) used visualizations of ICRs to aid in model selection in the context of latent growth curve modeling. When comparing linear and quadratic growth curve models for the same data, for example, they showed that a scatterplot of the ICRs for the quadratic model vs. ICRs for a linear model can help identify which model best minimizes model-data discrepancies. In the context of growth mixture modeling, Wang, Hendricks Brown, and Bandeen-Roche (2005) showed how visualization of empirical Bayes residuals (e.g., Q-Q and trajectory plots) can aid in determining the appropriate number of classes, an adequate shape of within-class growth trajectories, and missing confounders.

In short, ICRs have been used to identify high influence/leverage datapoints, nonlinear effects, heterogeneity, and to compare models. While these are certainly a step in the right direction, existing approaches suffer from a few weaknesses. First, ICRs rely on factor score estimates. Individual latent factor scores cannot be uniquely determined (Grice, 2001; Rigdon, Becker, & Sarstedt, 2019; Steiger, 1996). In cases where factors are highly indeterminate – e.g., factors with few indicators only weakly predicted by the latent factor – different factor score estimation methods can yield highly discrepant values, potentially even estimates that are negatively correlated (Grice, 2001). Also, ICRs are computed under the assumption the model is correct. When the model is misspecified, it is unclear how these visual diagnostics will behave. It is possible, of course, that for misspecified models, ICRs will reveal that misfit. (In fact, we show later that, at least under some circumstances, this is indeed the case). A final limitation of existing approaches is that many of their visuals cross platforms; some tools were developed in R (e.g., **influence.SEM** and **faoutlier**), some in Mplus (e.g., Asparouhov & Muthén, 2017), and some approaches provided no software implementation (e.g., xxxxx).

In this paper, we introduce **flexplavaan**, a suite of visualization tools designed to visualize **lavaan** models in R. Before we introduce **flexplavaan** and its core functions, however, we explain the types of graphics used and the rationale behind them.

Our Approach (Linear LVMs)

Example Data

To motivate our discussion/explanation of visualizing LVMs, we begin with a simulated dataset. Suppose the Jedi Council is attempting to identify Padawans who will make good Jedi Knights. To do so, they develop seven indicators (light saber score, fitness score, midichlorian levels, and a Jedi history exam, as well as three written exams completed at the end of Jedi training). Unbeknownst to the Jedi, these indicators measure two latent factors (force and Jedi), according to the relationships specified in Figure 3. Notice that one variable (history) has cross loadings on both factors. Also notice the path from force to Jedi is represented by a curved *one*-headed arrow. This is to indicate there is a curvilinear relationship between the two variables.

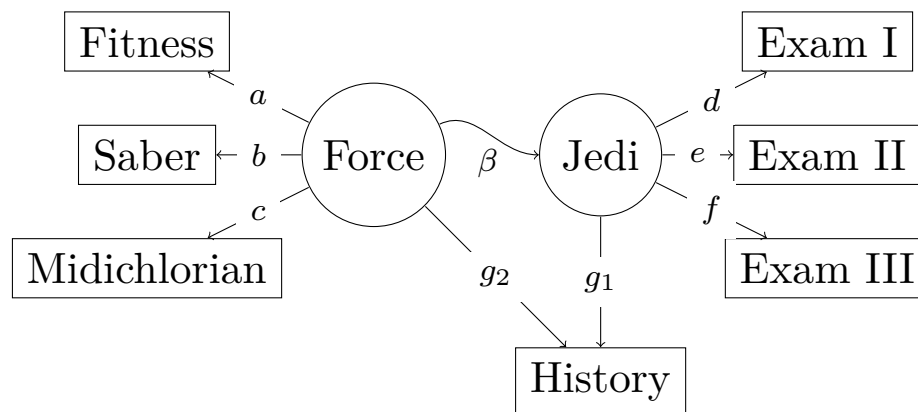


Figure 3. Simulated dataset of Jedi selection and training. This model features a crossloadings on the 'history' indicator, as well as a nonlinear relationship between force and Jedi. This is the data-generating model.

Unfortunately, the Jedi Council posits the model shown in Figure 4. (Jedi are notoriously poor at psychometrics). While most of the important elements are there, the Jedi's (incorrect) model specifies that history loads only onto force, and they posit a linear (rather than nonlinear) path from force to Jedi. According to traditional measures of fit, the χ^2 is not statistically significant. Also, the CFI (1.00), TLI (0.99), RMSEA (0.03), and SRMR (0.03) indicate respectable fit. In the examples that follow, we will use this example both to demonstrate how the visualization algorithms function and how they are able to identify misspecification that traditional fit statistics fail to capture.

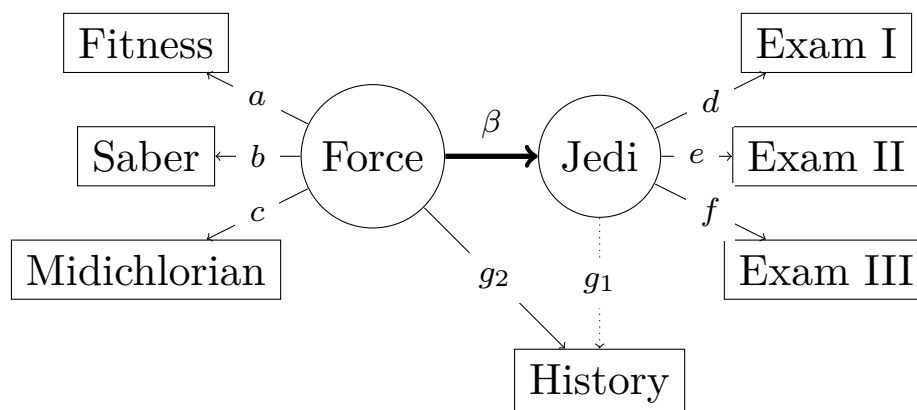


Figure 4. Actual model fitted to the Jedi dataset. This posits a linear (rather than nonlinear) relationship between the latent variables Jedi and force and fails to model the Jedi to history relationship.

Hopper Plots

Traditionally, conscientious researchers wanting to engage in model evaluation will often produce stem and leaf diagrams of the residual variance/covariance matrix. However, stem and leaf diagrams are somewhat less intuitive, especially for those without experience interpreting them. As an alternative, we suggest using what we call “hopper plots,” which plot the size of the residuals against the rank-ordered correlations (see Figure 5). The dots represent the residual size, while the lines show the absolute value of the residual (on the right) and $-1 \times$ the absolute value of the residual (on the left). These plots end up looking like a funnel, but the name “funnel plot” was already snagged by meta-analytic researchers. Instead, we call them hopper plots. (A hopper is a type of funnel, frequently used to dispense grains). By default, hopper plots only show those variables with residuals larger than 0.01 (in absolute value).

Trace Plots

In order to conceptualize our approach to visualizing LVMs, let us first consider how typical linear models are visualized. In a standard regression, each dot in a scatterplot represents scores on the observed variables. Often, analysts overlay additional symbols to represent the fit of the model (e.g., a line to represent the fitted regression model, or large dots to represent the mean). Sometimes additional symbols are overlaid to represent uncertainty (e.g., confidence bands for a regression line or standard error bars). In either case, the dots represent observed information, while the fitted information is conveyed using other symbols.

Likewise, visualizing LVMs might follow similar conventions; the dots should represent the observed information, as in Bauer (2005). In his visuals, pairwise relationships between observed variables are represented in a scatterplot. However, Bauer’s approach did not overlay a model-implied fit, as we seek to do. When the line represents the model-implied

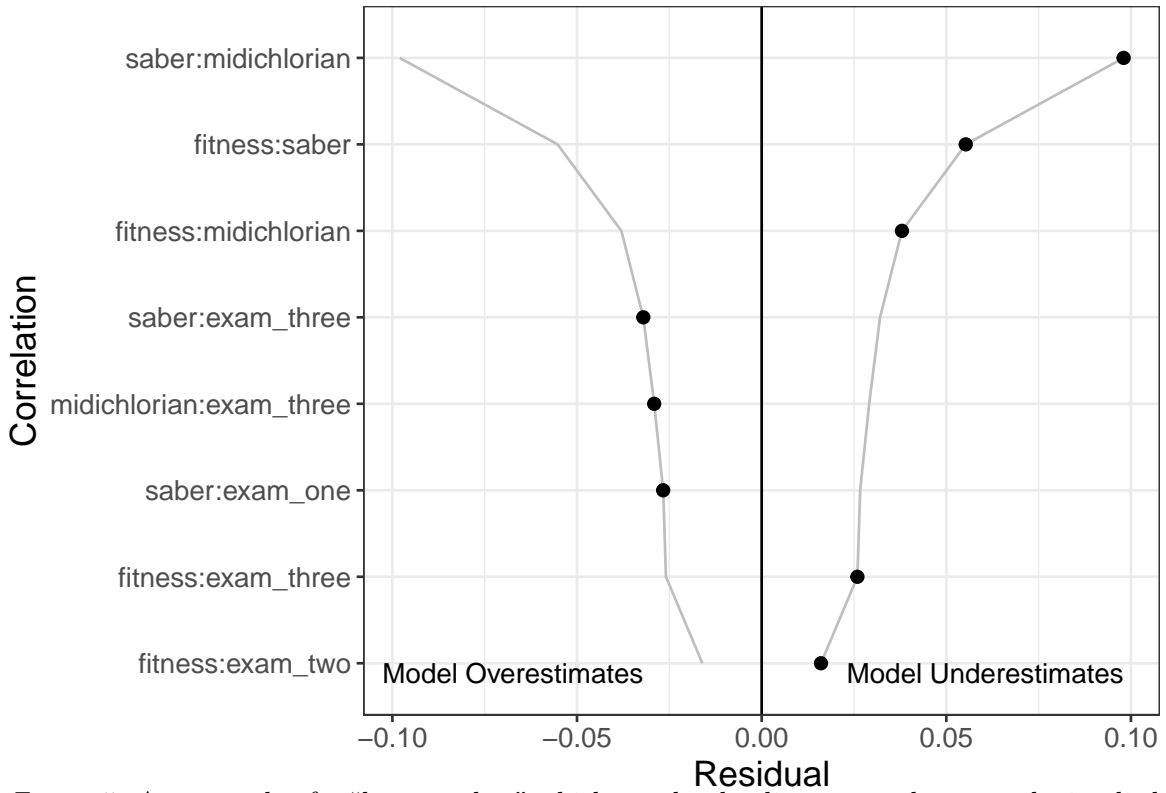


Figure 5. An example of a “hopper plot,” which graphs the discrepancy between the implied and observed correlation matrix. The dots represent observed residuals, while the lines represent the absolute value/negative absolute value.

fit, it denotes the “trace” left behind by the unobserved latent variable. As such, we call these plots “trace plots.”

How then does one identify the slope/intercept of the LVM’s model-implied fit? It is quite easy to do so when standard linear LVMs are used. Recall how our force factor has four indicators (e.g., saber, fitness, midichlorians, and history). To visualize the bivariate relationship between saber and fitness, for example, we can simply utilize the model-implied variance/covariance matrix. Recall the relationship between a covariance and a slope:

$$\beta_{y|x} = \frac{\sigma^2(x, y)}{\sigma_x^2}$$

For our example,

$$b_{S|F} = \frac{\hat{\sigma}_{S,F}^2}{\hat{\sigma}_F^2}$$

where S and F represent “saber” and “fitness,” respectively, and $\hat{\sigma}_F^2$ represents the *residual* variance of fitness. (Put differently, this is the expected slope between saber and fitness,

unadjusted for unreliability). With the slope, one can then estimate the intercept using basic algebra:

$$b_0 = \bar{S} - \beta_{S|F} \times \bar{F}$$

Figure 6 shows the LVM model-implied fit in red with a regression line in blue. Because the regression line minimizes the sum of squared errors, we would hope the LVM fitted line (red) closely approximates the regression line (blue). In this case, the two are similar, but there is a visually detectable difference between the two; the model (red line) underestimates the relationship between the two variables.

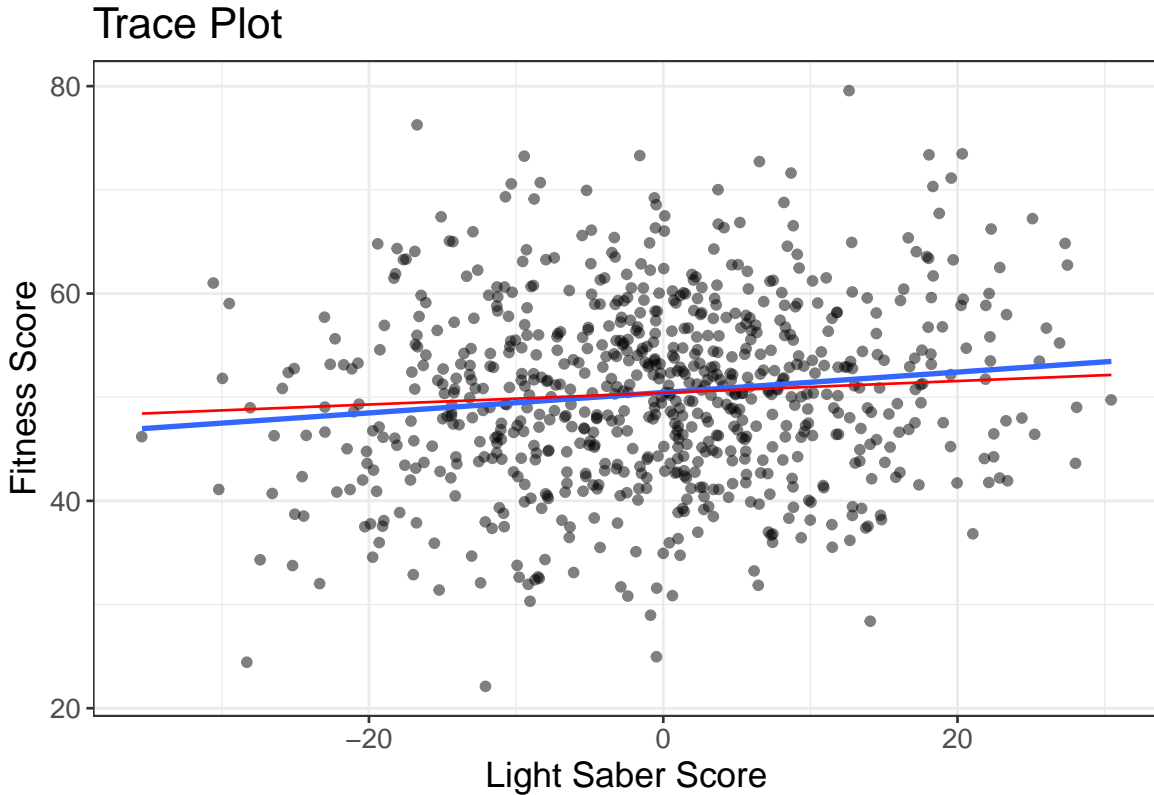


Figure 6. The LVM-implied fit between fitness score and light saber score, shown in red. The blue line represents the regression line between the two variables. The more closely the model-implied fit line resembles the regression line, the better the fit of the LVM.

Of course, Figure 6 only shows one pairwise relationship between variables. If we wished to visualize all the variables in our model, we would have to utilize a scatterplot matrix, as in Figure 7. The diagonal elements show histograms of ICRs, enabling researchers to (somewhat) evaluate the assumption of normality.³ Naturally, this becomes quite cumbersome when users have more than seven or eight variables. In this case, it is best to visualize only a subset of variables. By default, `flexplavaan` sorts the scatterplot matrix using a

³Technically, LVMs assume multivariate normality, while these plots show univariate normality. However, the univariate plots at least suggest when multivariate normality might be violated.

blockmodeling algorithm, then presents the block with the largest residuals first. For our figure (Figure 7), we have examined only the variables associated with the latent force variable, while Figure 8 shows the variables associated with the latent Jedi variable. We have also asked `flexplavaan` to show a loess line instead of a regression line, which will allow us to detect nonlinear patterns. These figures reveals some potential problems, including some nonlinearity, some underestimation (e.g., between saber and midichlorian), and some overestimation (e.g., between saber and history).

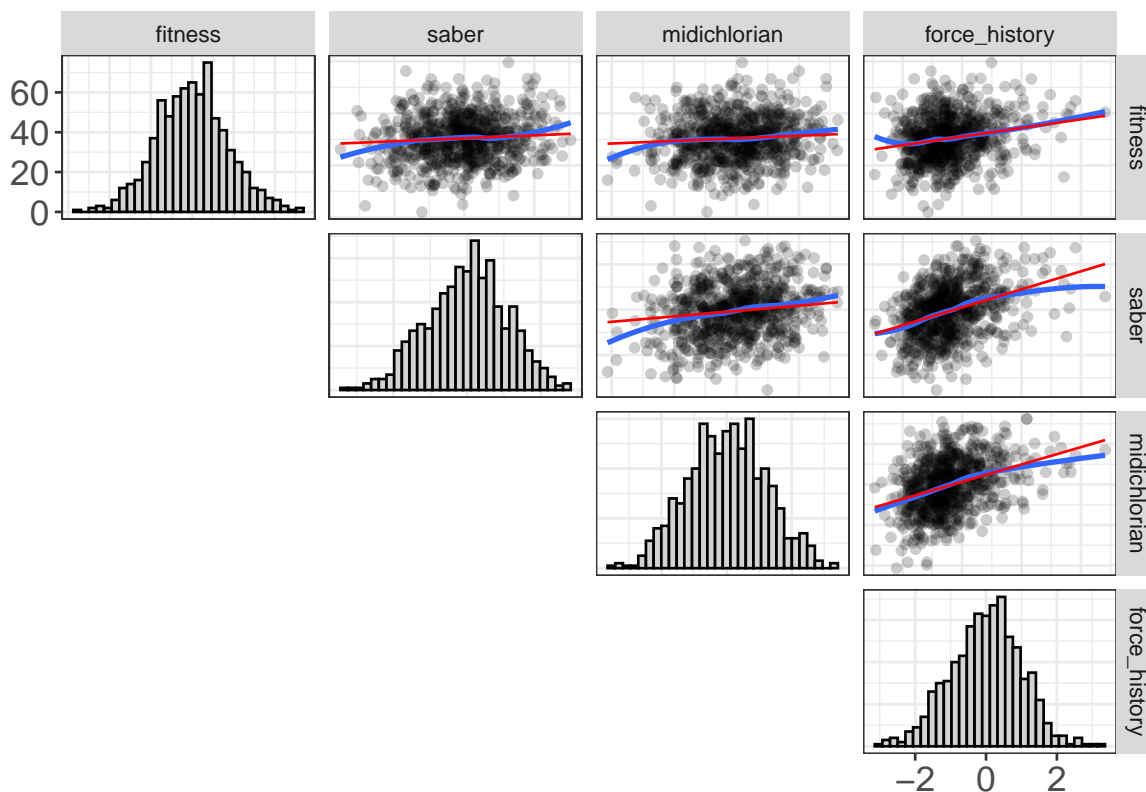


Figure 7. Scatterplot matrix showing the model-implied fit (red) and loess lines (blue) between four simulated indicator variables. These four variables are the indicators for the force latent variable. The diagonals show the histograms of the ICRs.

The primary advantage of trace plots is that they easily show many types of misspecification in LVMs. Another advantage is they visually (and often times strikingly) show how little information a model might capture. Returning to Figure 7, we see that many of the bivariate plots reveal quite weak relationships; many of the slopes are quite near zero. Recall that global fit indices suggested a well-fitting model. The trace plots, however, suggest there's little information to fit from the beginning for at least some of these relationships.

Disturbance-Dependence Plots

One common technique for visualizing the adequacy of statistical models in classic regression is residual-dependence plots. With these graphics, one simply plots the residuals

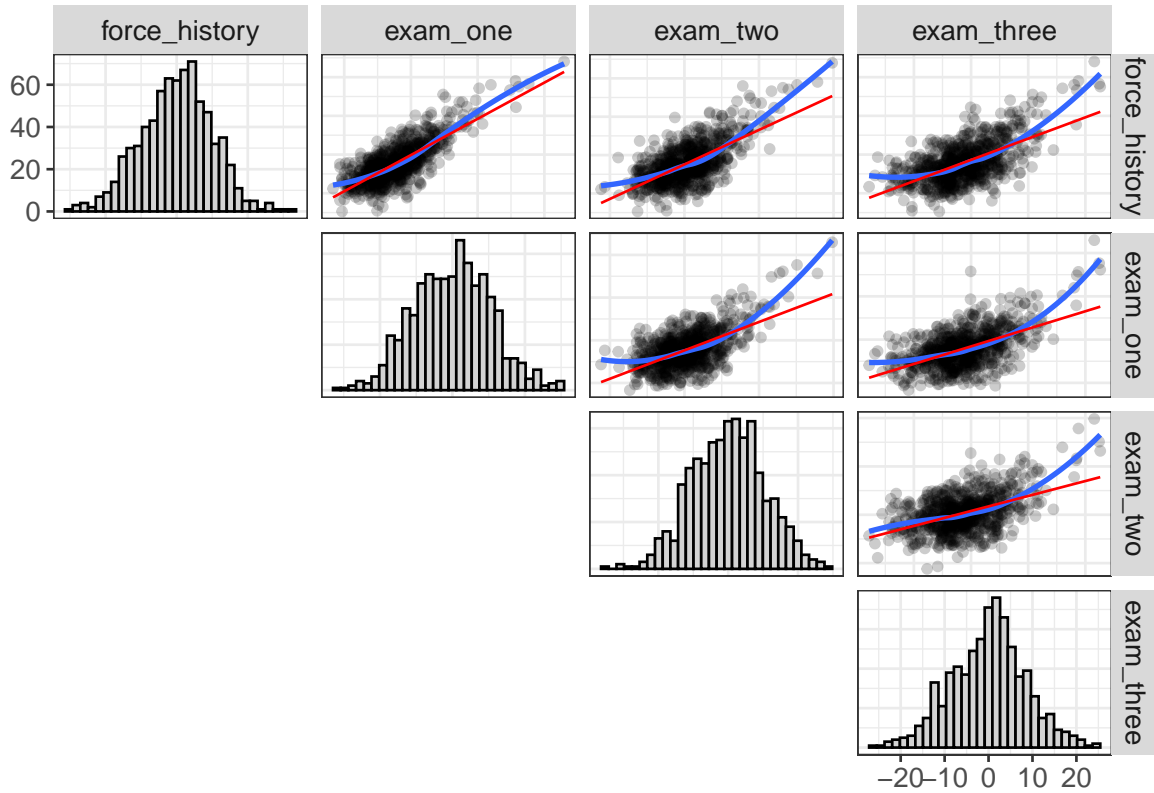


Figure 8. Scatterplot matrix showing the model-implied fit (red) and regression-implied fit (blue) between four simulated indicator variables. These four variables are the indicators for the Jedi latent variable. The diagonals show the histograms.

of the model (Y axis) against the predicted values (X axis). The rationale behind this is simple: the model should have extracted any association between the prediction and the outcome. The residuals represent the remaining information after extracting the signal from the model. If there is a clear trend remaining in the data (e.g., a nonlinear pattern or a “megaphone” shape in the residuals), this indicates the model failed to capture important information.

Likewise, in LVMs, we can apply this same idea to determine whether the fit implied by the LVM has successfully extracted any association between any pair of predictors. However, in LVMs, residuals refer to the discrepancy between the model-implied and the actual variance/covariance matrix (or correlation matrix). As such, naming these plots “residual-dependence plots” would be a misnomer. Rather, misfit at the raw data level is typically called either a disturbance or an individual case residual (ICR), as mentioned previously. In this paper, we call these plots disturbance-dependence plots.

Like trace plots, we visualize disturbance-dependence plots for each pair of observed variables. To do so, `flexplavaan` subtracts the fit implied by the model from the observed scores. For example, a disturbance dependence plot for an X_1/X_2 relationship would subtract the “fit” of X_2 implied by the model from the actual X_2 scores (and vice versa for the X_2/X_1 relationship). If the trace-plot fit actually extracts all association between the pair of

observed variables, we would expect to see a scatterplot that shows no remaining association between the two. If there is a pattern in the scatterplot remaining, we know the fit of the model misses information about that specific relationship.

To aid in interpreting these plots, we can overlay the plot with a flat line (with a slope of zero), as well as a regression (or loess) line. The first line indicates what signal should remain after fitting the model, while the second line shows what actually remains.

Figure 9 shows an example of trace plots in the upper triangle and disturbance-dependence plots in the lower triangle of a scatterplot matrix. These plots are for the same data shown in the right image of Figure 8. Notice how many of the plots have nonlinear patterns showing up in both the DDP and the trace plots.

Trace/DDP Plots

Red=Implied, Blue=Observed

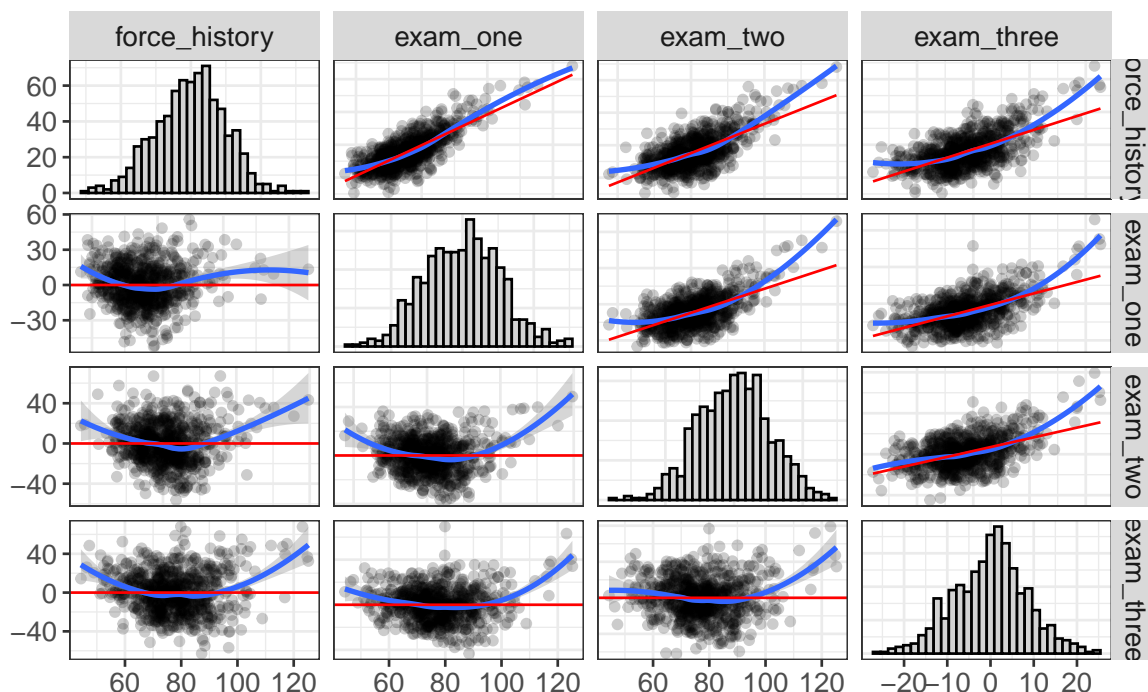


Figure 9. This plot shows a disturbance-dependence plot for the same data visualized in Figure 8 in the lower triangle.

Together, both of these plots (trace plots and DDPs) serve as a critical diagnostic check. Both these plots will signal certain types of misfit both in the measurement and structural components of the model. However, these plots suffer from a major weakness. Recall how earlier we referenced Figure 2 and noted that sometimes severe misspecification will go undetected simply because an incorrect model will often yield a model-implied covariance matrix that well approximates the actual covariance matrix. However, this sort of misspecification may show up in measurement plots, which we address next.

Measurement Plots

Earlier we mentioned how Asparouhov and Muthén (2017) utilized factor score estimates to visualize ICRs as a diagnostic tool. One of their plots showed the latent variable on the Y axis and the observed (indicator) variable on the X axis. While these may be capable of revealing nonlinearities, they cannot reveal other sorts of misspecification (e.g., many types of cross-loadings and residual correlations) without a simple modification. The modification we propose is similar to the traceplots: overlay the model-implied slope and a regression (or loess) line.

Fortunately, similar to the trace plots, we can use the model-implied variance/covariance matrix of the observed/latent variables to determine the model-implied slope. The advantage of these plots is they are far more sensitive to many types of misspecification than trace plots. This is because trace plots are unable to pick up misspecification unless that misspecification introduces bias in estimating the observed variance/correlation matrix. However, as shown in Figure 2, not all misspecification manifests itself as bias in estimating correlations between observed variables. While the two models in Figure 2 will not yield different observed covariances, they will yield different latent variable estimates (and thus, different covariances between latent/observed).

Figure 10 plots several graphs of the relationship between the observed variables and the latent variables. To do so, **flexplavaan** puts all variables on a common scale. Additionally, **flexplavaan** defaults to displaying only four observed variables at a time. Which four are chosen is determined by the degree of discrepancy between the observed (blue) and model-implied (red) slope, such that the four observed variables with the largest discrepancy are chosen. From Figure 10, we see that the model consistently underestimates the relationship between the indicators and the latent variables, and that this underestimation is stronger for the Jedi latent variable. It is also interesting to note that the history indicator is nearly synonymous with the force latent variable.

Structural (Cross-Hair) Plots

When modeling latent variables, often the visuals of interest are not the observed variables, but the latent variables. In other words, the measurement model is ancillary to the substantive model. Naturally, we might wish to visualize the relationship between the latent variables.

However, our latent variables are merely estimates. As such, we ought to have visuals that reflect uncertainty in our estimates of the latent scores. In **flexplavaan**, this uncertainty is represented as crosshairs. The widths of each line of the crosshair (for both the X axis and the Y axis) are obtained from prediction intervals for **lavaan** objects (using the **plausibleValues** function of the **semTools** package). Figure 11 shows these plots, which we call “structural plots,” or “cross-hair plots”. Interestingly, the relationship between the two, though simulated to be significantly nonlinear, shows up as fairly linear. This seems to suggest that, by the time a model’s factor scores are estimated, any nonlinearity that exists in the data has been discarded. For this reason, we disagree with the approach recommended

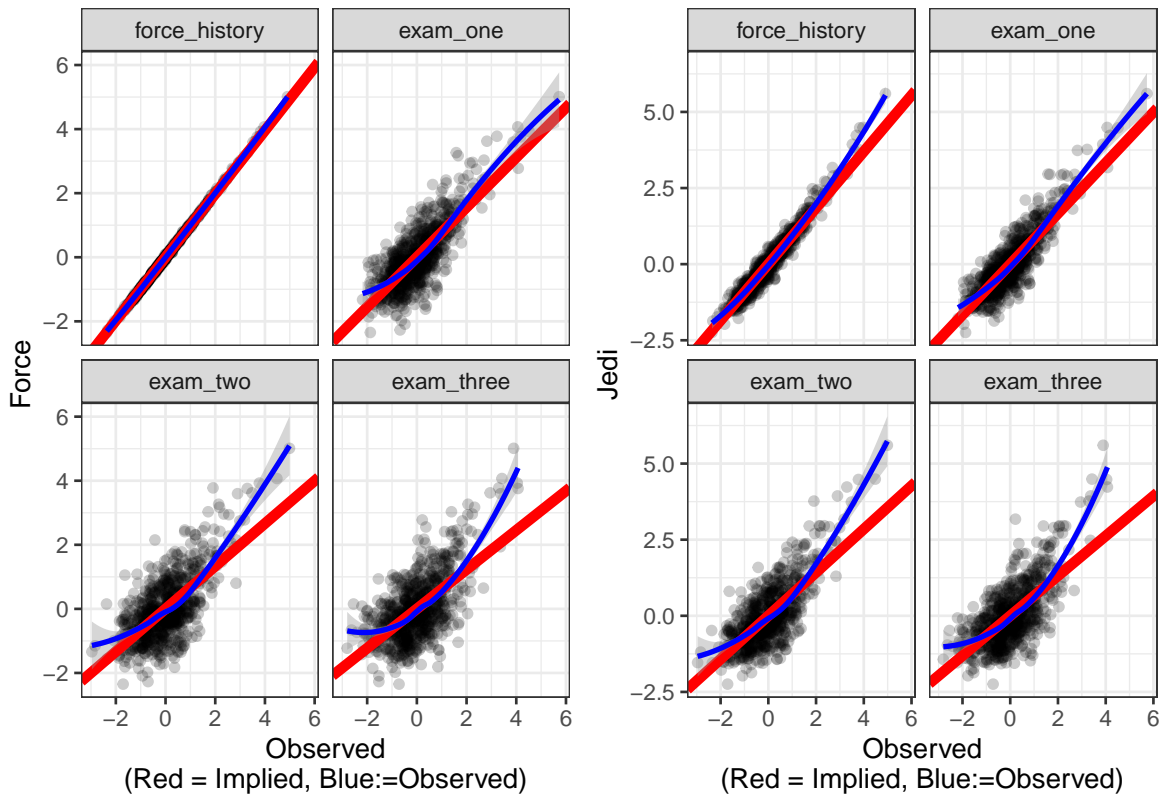


Figure 10. This image, called a measurement plot, shows the relationship between the latent variables (Y axis) and each standardized indicator. The blue lines are loess lines, while the red lines are the model-implied fits of the model.

by Hallgren et al. (2019), who suggest researchers plot a structural plot as evidence for model fit; simply plotting structural plots without the previous graphics will yield a very misleading picture of data/model fit. This final plot is little more than a summary of the model, provided the model diagnostics check out. Showing these is no more meaningful than reporting a p-value for a model that may or may not meet statistical assumptions.

When there are multiple latent variables, there is a great deal of flexibility in how one visualizes the structural model. `flexplavaan` makes a best guess at how to visualize this relationship using the model specified by the user. However, the user can always specify how to plot the structural model using a `flexplot` equation (Fife, 2020a). In our example, we only had two variables to visualize, so a simple bivariate plot was most natural. When more variables are included, we might utilize paneling, added variable plots, beeswarm plots, etc. For a review of the types of plots possible, see Fife (2020a).

Model Comparisons

It is quite common to compare two different models when using LVMs. Indeed, model comparisons are a recommended strategy for building LVMs (Rodgers, 2010). Not only can we use statistics to do model comparisons, but we can also compare them visually.

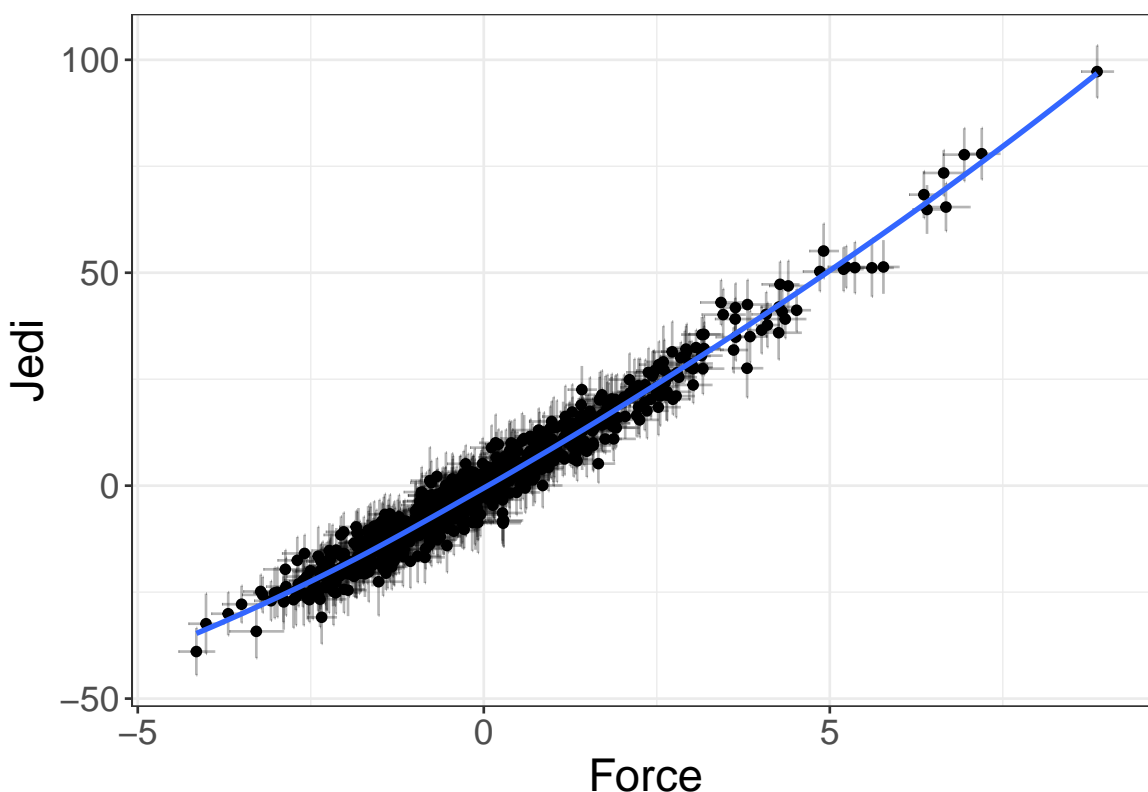


Figure 11. Structural or “cross-hair” plot of the relationship between the latent variables force and Jedi. The crosshairs represent the prediction intervals for the factor scores of the latent variables.

All plots previously mentioned (hopper plots, trace plots, DDPs, and structural plots) can visualize two models at the same time. While visualizing two models side-by-side makes differences more visually detectable, we also recommend visualizing each model individually since certain aspects of misfit in a model might be masked by the comparison.

For illustration purposes, we’re going to fit a second model that attempts to address both the nonlinearity and the missing cross-loading. The second model is shown in Figure 12. To model the nonlinearity, we have created three nonlinear indicators (saber^2 , fitness^2 , and midichlorian^2), as well as added the cross-loading from Jedi to history.

Figure 13 shows the residual plots, for all pairwise relationships. The new model seems to have reduced the largest residuals (though it has, in some instances, slightly increased the size of some residuals).

Figure 14 shows the trace/DDPs for the two models. As before, the red line is the original model, while the blue line shows the nonlinear model for the variables saber, fitness, midichlorian, and history. There are some minor discrepancies (e.g., the saber/midichlorian relationship), with the nonlinear model suggesting stronger relationships between the variables. However, as before, these plots are not terribly sensitive to misspecification. (In fact, the trace/DDPs for the other factor showed hardly any discrepancies).

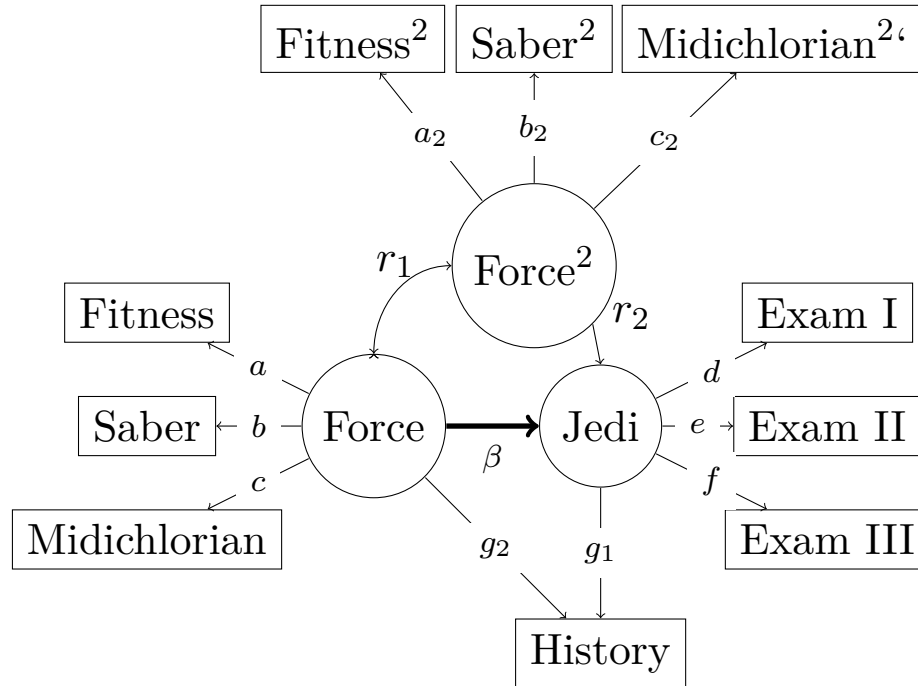


Figure 12. Nonlinear model for the Jedi dataset. This model proposes a new latent variable (Force^2) that has saber, fitness, and midichlorian as squared indicators. This model also allows history to load onto both force and Jedi.

On the other hand, the measurement plots (Figure 15) do show differences between the two models, at least for the force latent variable. The nonlinear model shows consistently weaker relationships between the force variable and the indicators shown (force history, exam one, exam two, and exam three).

Finally, the structural plot, shown in Figure 16 shows the relationship between the force/Jedi variable. The data from the left plot are for the original model, while the right plot are for the nonlinear model. Not surprisingly, the right model captures the nonlinearity between the two much better than does the original model.

Taken together, these results suggest the original model failed to capture important nonlinear relationship, overestimated some relationships, and underestimated others. The nonlinear model, however, has smaller residuals, seems to better approximate the actual data, and suggests the latent variables are associated nonlinearly. Table 1 shows fit statistics for each model. We display these *not* to validate the efficacy of the visuals (we trust the visuals more than we trust the fit statistics). Rather, we show these to highlight the convergent validity of the statistics and the graphics.

References

Asparouhov, T., & Muthén, B. (2017). *Using Mplus individual residual plots for diagnostics and model evaluation in SEM.*

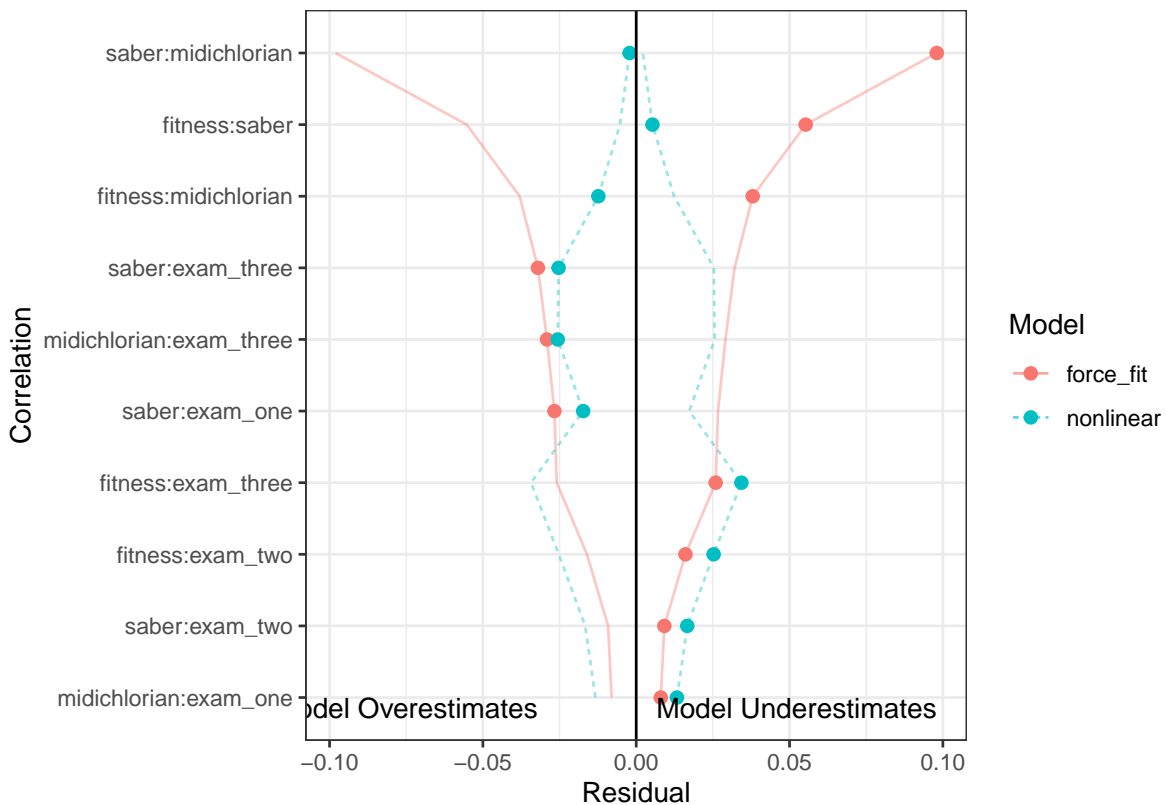


Figure 13. A hopper plot, but this time two models are being compared. The red line shows the residuals for the original model, while the blue lines show the residuals for the polynomial model.

Barrett, P. (2007). Structural equation modelling: Adjudging model fit. *Personality and Individual Differences*, 42(5), 815–824.

Bauer, D. J. (2005). The role of nonlinear factor-to-indicator relationships in tests of measurement equivalence. *Psychological Methods*, 10(3), 305–316. <https://doi.org/10.1037/1082-989X.10.3.305>

Bollen, K. A. (1989). *Structural equations with latent variables*. John Wiley & Sons.

Bollen, K. A. (2019). Model implied instrumental variables (miivs): An alternative orientation to structural equation modeling. *Multivariate Behavioral Research*, 54(1), 31–46.

Table 1

Global Fit Indices for the Original and Nonlinear Model for the Jedi Dataset.

	χ^2	df	p	CFI	TLI	BIC	RMSEA	SRMR
Original	19.4	13	0.110	0.996	0.994	42,616.2	0.026	0.025
Nonlinear	26.3	31	0.705	1.000	1.004	50,388.6	0.000	0.022

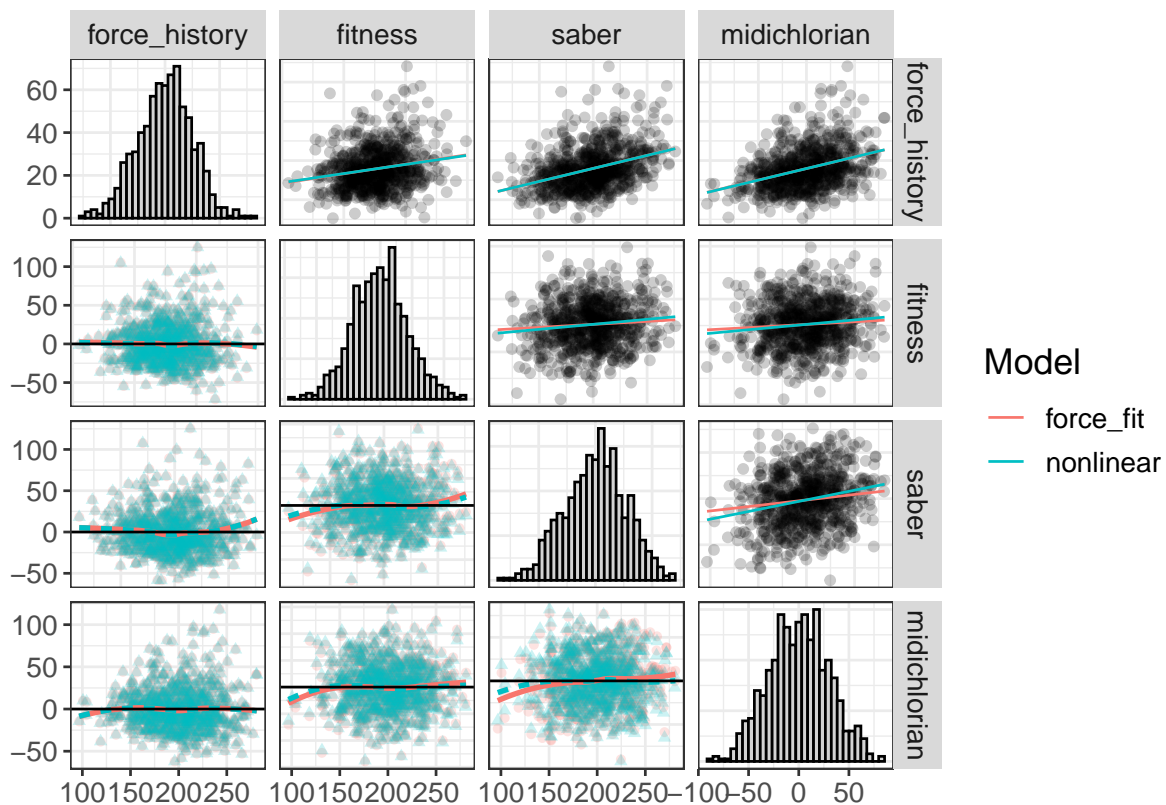


Figure 14. Trace plots and DDPs for the two models. The red shows the original model, while the blue shows the fit for the nonlinear model.

Bollen, K. A., & Arminger, G. (1991). Observational residuals in factor analysis and structural equation models. *Sociological Methodology*, 235–262.

Camerer, C. F., Dreber, A., Holzmeister, F., Ho, T. H., Huber, J., Johannesson, M., ... Wu, H. (2018). Evaluating the replicability of social science experiments in Nature and Science between 2010 and 2015. <https://doi.org/10.1038/s41562-018-0399-z>

Chalmers, P. (2017). Package “faoutlier”.

Chen, F., Curran, P. J., Bollen, K. A., Kirby, J., & Paxton, P. (2008). An empirical evaluation of the use of fixed cutoff points in RMSEA test statistic in structural equation models. *Sociological Methods and Research*, 36(4), 462–494. <https://doi.org/10.1177/0049124108314720>

Correll, M. A. (2015). *Visual Statistics* (Doctoral Dissertation). University of Wisconsin-Madison.

Fife, D. A. (2019). A Graphic is Worth a Thousand Test Statistics: Mapping Visuals onto Common Analyses. Retrieved from <http://rpubs.com/dustinfife/528244>

Fife, D. A. (2020a). Flexplot: Graphical-Based Data Analysis. *PsyArxiv*. <https://doi.org/10.31234/osf.io/kh9c3>

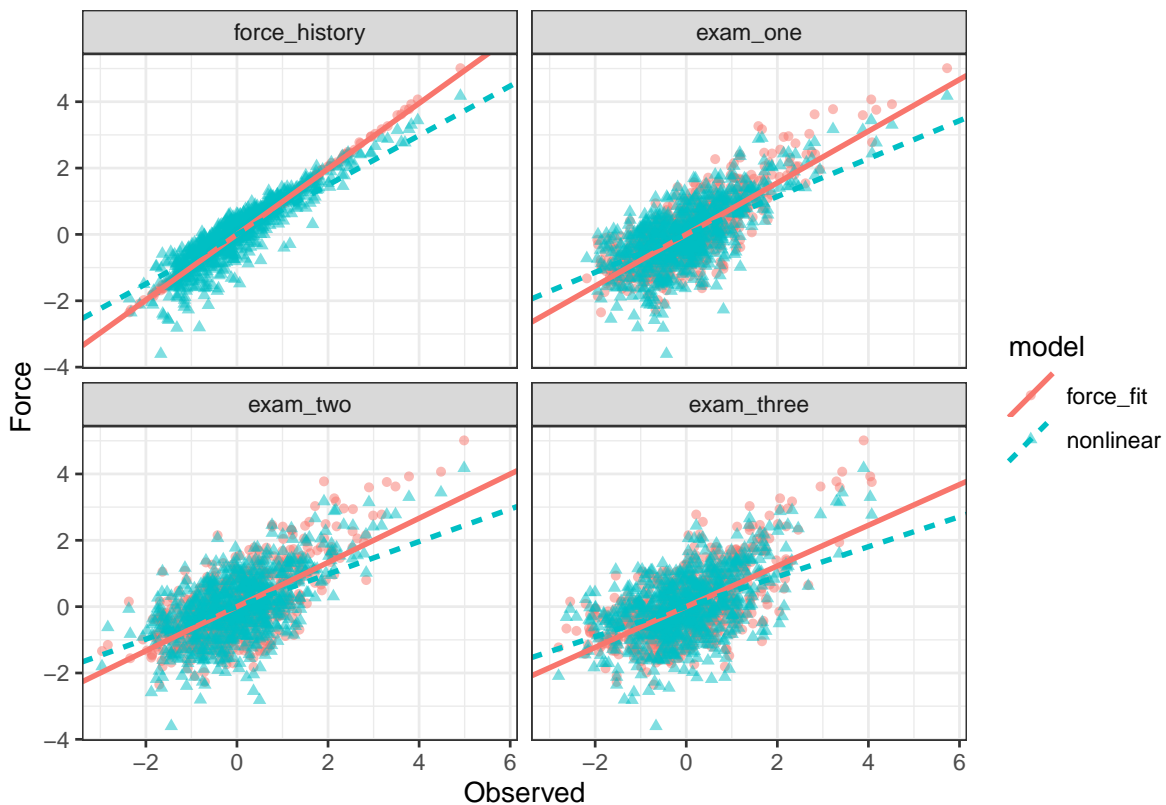


Figure 15. Measurement plots for the force latent variable, with both the nonlinear, red, model and original, blue, model.

Fife, D. A. (2020b). The Eight Steps of Data Analysis: A Graphical Framework to Promote Sound Statistical Analysis. *Perspectives on Psychological Science*, 15(4), 1054–1075. <https://doi.org/10.1177/1745691620917333>

Fife, D. A., & Rodgers, J. L. (2019). Exonerating EDA: Addressing the Replication Crisis By Expanding the EDA/CDA Continuum. *Unpublished Manuscript*. Retrieved from <http://quantpsych.net/fife-exonerating-eda-draft-oct2019-df-edits/>

Flora, D. B., LaBrish, C., & Chalmers, R. P. (2012). Old and new ideas for data screening and assumption testing for exploratory and confirmatory factor analysis. *Frontiers in Psychology*, 3, 55.

Goodboy, A. K., & Kline, R. B. (2017). Statistical and Practical Concerns With Published Communication Research Featuring Structural Equation Modeling. *Communication Research Reports*, 34(1), 68–77. <https://doi.org/10.1080/08824096.2016.1214121>

Grice, J. W. (2001). Computing and evaluating factor scores. *Psychological Methods*, 6(4), 430.

Hallgren, K. A., McCabe, C. J., King, K. M., & Atkins, D. C. (2019). Beyond path diagrams: Enhancing applied structural equation modeling research through data visualization. *Addictive Behaviors*, 94(March 2018), 74–82. <https://doi.org/10.1016/j.addbeh.2018.08>

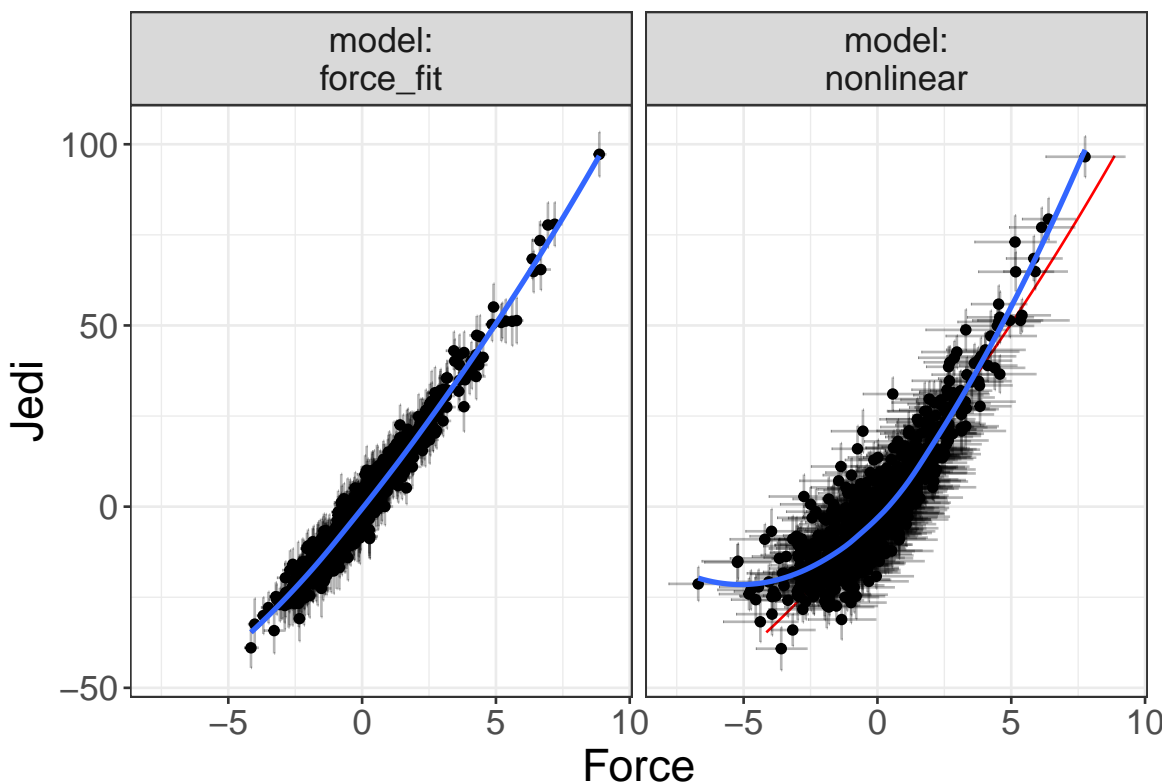


Figure 16. Relationship between the two latent variables for the original model (left) and the nonlinear model (right). The red line is a ‘ghost line,’ which simply repeats the line from the left plot to the right plot.

030

- Hayduk, L. (2014). Seeing Perfectly Fitting Factor Models That Are Causally Misspecified: Understanding That Close-Fitting Models Can Be Worse. *Educational and Psychological Measurement*, 74(6), 905–926. <https://doi.org/10.1177/0013164414527449>
- Hayduk, L. A. (2014). Shame for disrespecting evidence: The personal consequences of insufficient respect for structural equation model testing. *BMC Medical Research Methodology*, 14(1), 124.
- Healy, K., & Moody, J. (2014). Data Visualization in Sociology. *Annual Review of Sociology*, 40(1), 105–128. <https://doi.org/10.1146/ANNUREV-SOC-071312-145551>
- Hipp, J. R., & Bollen, K. A. (2003). Model fit in structural equation models with censored, ordinal, and dichotomous variables: Testing vanishing tetrads. *Sociological Methodology*, 33(1), 267–305.
- Hu, L.-t., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to underparameterized model misspecification. *Psychological Methods*, 3(4), 424.
- Ioannidis, J. P. A. (2005). Why Most Published Research Findings Are False. *PLoS Medicine*, 2(8), e124. <https://doi.org/10.1371/journal.pmed.0020124>

- Jackson, D. L., Gillaspay Jr, J. A., & Purc-Stephenson, R. (2009). Reporting practices in confirmatory factor analysis: An overview and some recommendations. *Psychological Methods*, 14(1), 6.
- Lai, K., & Green, S. B. (2016). The problem with having two watches: Assessment of fit when rmsea and cfi disagree. *Multivariate Behavioral Research*, 51(2-3), 220–239.
- Lee, S., & Hershberger, S. (1990). A simple rule for generating equivalent models in covariance structure modeling. *Multivariate Behavioral Research*, 25(3), 313–334.
- Levine, S. S. (2018). Show us your data: Connect the dots, improve science. *Management and Organization Review*, 14(2), 433–437. <https://doi.org/10.1017/mor.2018.19>
- MacCallum, R. C., Wegener, D. T., Uchino, B. N., & Fabrigar, L. R. (1993). The problem of equivalent models in applications of covariance structure analysis. *Psychological Bulletin*, 114(1), 185–199.
- McIntosh, C. N. (2007). Rethinking fit assessment in structural equation modelling: A commentary and elaboration on Barrett (2007). *Personality and Individual Differences*, 42(5), 859–867. <https://doi.org/10.1016/j.paid.2006.09.020>
- Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences*. <https://doi.org/10.1073/pnas.1708274114>
- Pashler, H., & Wagenmakers, E.-J. (2012). Editors' Introduction to the Special Section on Replicability in Psychological Science. *Perspectives on Psychological Science*, 7(6), 528–530. <https://doi.org/10.1177/1745691612465253>
- Pastore, M., & Pastore, M. M. (2018). Package “influence. SEM”.
- Pek, J., & MacCallum, R. C. (2011). Sensitivity analysis in structural equation models: Cases and their influence. *Multivariate Behavioral Research*, 46(2), 202–228.
- Raykov, T., & Penev, S. (2014). Latent growth curve model selection: The potential of individual case residuals. *Structural Equation Modeling: A Multidisciplinary Journal*, 21(1), 20–30.
- Rigdon, E. E., Becker, J.-M., & Sarstedt, M. (2019). Factor indeterminacy as metrological uncertainty: Implications for advancing psychological measurement. *Multivariate Behavioral Research*, 54(3), 429–443.
- Rodgers, J. L. (2010). The epistemology of mathematical and statistical modeling: a quiet methodological revolution. *The American Psychologist*, 65(1), 1–12. <https://doi.org/10.1037/a0018326>
- Shi, D., & Maydeu-Olivares, A. (2020). The effect of estimation methods on sem fit indices. *Educational and Psychological Measurement*, 80(3), 421–445.
- Smith, T. D., & McMillan, B. F. (2001). A primer of model fit indices in structural equation modeling.

- Steiger, J. H. (1996). Dispelling some myths about factor indeterminacy. *Multivariate Behavioral Research*, 31(4), 539–550.
- Steiger, J. H. (2007). Understanding the limitations of global fit assessment in structural equation modeling. *Personality and Individual Differences*, 42(5), 893–898. <https://doi.org/10.1016/j.paid.2006.09.017>
- Tay, L., Parrigon, S., Huang, Q., & LeBreton, J. M. (2016). Graphical Descriptives: A Way to Improve Data Transparency and Methodological Rigor in Psychology. *Perspectives on Psychological Science*, 11(5), 692–701. <https://doi.org/10.1177/1745691616663875>
- Thoemmes, F., Rosseel, Y., & Textor, J. (2018). Local fit evaluation of structural equation models using graphical criteria. *Psychological Methods*, 23(1), 27–41. <https://doi.org/10.1037/met0000147>
- Tomarken, A. J., & Waller, N. G. (2003). Potential problems with "well fitting" models. *Journal of Abnormal Psychology*, 112(4), 578.
- Tomarken, A. J., & Waller, N. G. (2005). Structural Equation Modeling: Strengths, Limitations, and Misconceptions. *Annual Review of Clinical Psychology*, 1(1), 31–65. <https://doi.org/10.1146/annurev.clinpsy.1.102803.144239>
- Wang, C.-P., Hendricks Brown, C., & Bandeen-Roche, K. (2005). Residual diagnostics for growth mixture models: Examining the impact of a preventive intervention on multiple trajectories of aggressive behavior. *Journal of the American Statistical Association*, 100(471), 1054–1076.
- Wilkinson, L., & Task Force on Statistical Inference. (1999). Statistical Methods in Psychology Journals: Guidelines and Explanations. *American Psychologist*, 54(8), 594–601.
- Yuan, K.-H., & Hayashi, K. (2010). Fitting data to model: Structural equation modeling diagnosis using two scatter plots. *Psychological Methods*, 15(4), 335.