It has been said that Latent Variable Models (LVMs) are “a dangerously conjectural technique for asking essential research questions which otherwise are impossible to consider” (McDonald, 1999, p. 367). LVMs are frequently misused and abused. For example, users often utilize rules of thumb that have no empirical basis (Chen, Curran, Bollen, Kirby, & Paxton, 2008), which invites users to view fit as a dichotomous evaluation. Another problem with LVMs is there is no general consensus on how to evaluate fit. The chi square fit index is overly sensitive to sample size, often flagging well-fitting models as problematic while letting poor-fitting models slip (Jiang & Yuan, 2017; Steiger, 2004). Approximate fit indices are often utilized instead, but these have their own sets of problems; fit indices are sensitive to factors they should not, such as sample size, reliability of items, number of variables, number of latent variables, etc. (West, Taylor, & Wu, 2012). A final problem with LVMs is that they, like most other statistical models, make several assumptions, such as linearity, homoscedasticity, and normality. While other statistical models (e.g., regression) have standard procedures for evaluating the efficacy of assumptions, with LVMs this is less straightforward. Very often the model is fitted on a variance/covariance matrix, rather than the raw data. As such, it becomes impossible to evaluate the viability of assumptions.

Most, if not all, of these problems could be addressed if LVMs could be visualized. Graphics allow one to determine, at a glance, if the assumptions have been met, if the model fits the data, and the fit is displayed continuously, which can prevent users from making binary evaluations of model-fit (Correll & Gleicher, 2014). Unfortunately, the variables modeled are unobserved. How does one visualize a model that is, by definition, unobserved? In this paper we seek to introduce both an approach and statistical software for visualizing LVMs. This approach/software, which we call flexplavaan, aims to provide essential tools for model evaluation. The software, available in R, combines the powerful graphing capabilities of flexplot (Fife, 2020) with the modeling capabilities of lavaan/blavaan (Merkle & Rosseel, 2018). In the next section, we introduce the general approach to visualizing LVMs, then demonstrate with a few notable examples.

**Trace Plots**

To understand how flexplavaan visualizes LVMs, let us consider how standard regression models are visualized. The graphic of choice for regression is a scatterplot. Each dot in a scatterplot represents the score on a pair of *observed* variables. A regression line is often superimposed on the scatterplot. The line is the *model.*

Flexplavaan takes a similar approach. Each pairwise relationship (e.g., between two latent indicator variables) can be visualized as a scatterplot. Multiple relationships can be visualized simultaneously through a scatterplot matrix, as in Figure 1. As before, we can overlay a line that is a visual representation of the model, which is shown as blue lines. These lines are the regression lines that are implied by the latent variable model. They are derived from the model-implied correlation between each pair of variables. As such, they show the “trace” left by the latent variable, hence the name trace plots. The green lines are simple loess lines that make it easier to see how well the LVM fits the data; if the green and blue lines look very similar, the model fits well. If they do not (as in the top row of plots), there is misfit in the data.

These plots can be used to detect model misfit (as in Figure 1) as well as violated assumptions. Consider Figure 2, which shows very clear nonlinear patterns between various observed variables.

**Disturbance Dependence Plots: Removing the Effect of the Latent Variable**

It may also be interesting to utilize “disturbance dependence plots,” or to plot the relationship between the variables on interest, once the effect of the latent variable has been removed. This is similar to a residual dependence plot in regression models, but we use different terminology to avoid confusion (since residuals in SEM often refer to the residual correlation matrix).

To generate these disturbance terms, we need only to generate predicted factor scores, then subtract those from the observed factor scores. We can then plot the relationship between two variables after removing the effect of the latent variable. If the data are locally independent, we should observe no relationship. We could then show the expected relationship as a line centered on zero with no slope. We could then add this plot to the scatterplot matrix in, say, the lower diagonals, as in Figure 4.

**Model Plots**

Once one has iterated through the diagnostic plots, they can be more confident the factor scores are accurately estimated. At that point, the user may choose to plot the relationship between each indicator and the latent variable(s), or they may choose to plot the relationship between multiple latent variables. However, representing these as a scatterplot can be misleading because these are estimated scores, not observed scores. As such, one should provide a visual representation of uncertainty. In flexplavaan, each point is an ellipse whose size is an indicator of uncertainty.

**Discussion**

In this paper we have introduced the logic and approach used to visualize LVMs in flexplavaan. These visualizations diagnose misfit, identify violated assumptions, and help users to better conceptualize statistical models. It is our hope that this software package will assist in helping users become better informed consumers of LVMs.

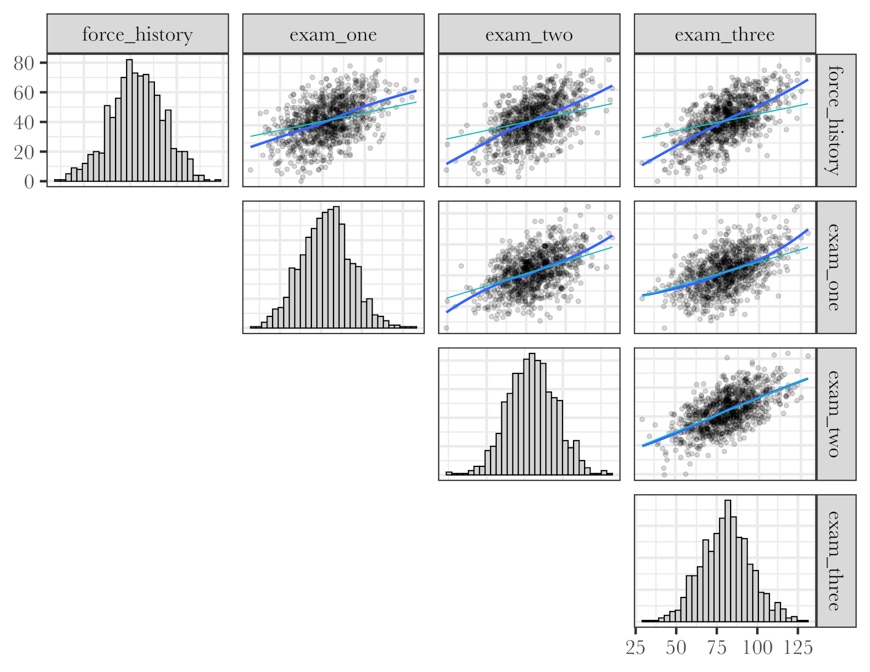


Figure : Scatterplot matrix of four observed variables. The blue line is the model-implied fit, while the green line is the loess line.



Figure : Flexplavaan scatterplot matrix that shows nonlinear relationships for the Hogwarts dataset.



Figure : Flexplavaan scatterplot matrix with disturbance-dependence plots that show nonlinear relationships for the Hogwarts dataset.



Figure : A model plot that shows the estimated relationship between two latent variables.