

Parsimony in Model Selection: Tools for Assessing Fit Propensity

Carl F. Falk¹ and Michael Muthukrishna²

¹ Department of Psychology, McGill University

² Department of Psychological and Behavioural Science, London School of Economics and Political Science

Abstract

Theories can be represented as statistical models for empirical testing. There is a vast literature on model selection and multimodel inference that focuses on how to assess which statistical model, and therefore which theory, best fits the available data. For example, given some data, one can compare models on various information criterion or other fit statistics. However, what these indices fail to capture is the full range of counterfactuals. That is, some models may fit the given data better not because they represent a more correct theory, but simply because these models have more *fit propensity*—a tendency to fit a wider range of data, even nonsensical data, better. Current approaches fall short in considering the principle of parsimony (Occam's Razor), often equating it with the number of model parameters. Here we offer a toolkit for researchers to better study and understand parsimony through the fit propensity of structural equation models. We provide an R package (*ockhamSEM*) built on the popular *lavaan* package. To illustrate the importance of evaluating fit propensity, we use *ockhamSEM* to investigate the factor structure of the Rosenberg Self-Esteem Scale.

Translational Abstract

How should we assess which theory or model is correct in light of available data? One approach is to use information criteria or other fit statistics to compare which model best fits the available data. But this approach ignores parsimony. Some models may fit the given data better not because they represent a more correct theory, but simply because these models have more fit propensity—a tendency to better fit a wider range of data, even nonsensical data. Here we offer a toolkit for researchers to assess parsimony through the fit propensity of structural equation models. We provide an R package (*ockhamSEM*) built on the popular *lavaan* package. We illustrate the importance of evaluating fit propensity by using *ockhamSEM* to investigate the factor structure of the Rosenberg Self-Esteem Scale.

Keywords: fit indices, parsimony, model fit, structural equation modeling, formal theory

Supplemental materials: <https://doi.org/10.1037/met0000422.supp>

Theories, no matter how beautiful, live and die on the back of data. Structural equation modeling offers a flexible framework for statistically representing complex theories (Bollen & Pearl, 2013; Grace & Bollen, 2008). Given a choice between two or more theoretically plausible structural equation models, the process of model selection and multimodel inference (Burnham & Anderson, 2002) typically involves asking which model is more consistent with the available empirical data. For example, there are many instances in psychological research where a broad (multifaceted) construct is

defined, a test is created, and further psychometric work indicates that a multidimensional model fits better than a model that measures a single dimension. Some examples include the number and configuration of possible method factors on scales that include reverse-worded items (Reise et al., 2016) or whether a random intercept model should be used to model acquiescence bias (Savalei & Falk, 2014), the tradeoff between a correlated factor, hierarchical factor, and bifactor models for constructs such as self-compassion (Neff et al., 2017), alexithymia (Reise et al., 2013), health outcomes (Reise et al., 2007), and so on. In all cases, debates continue over which model is most correct. What is often overlooked is the counterfactual—that a model may not fit the empirical data better because it is a better description of reality, but simply because it has a tendency to fit *any data* better. That is, what is often overlooked is parsimony.

Occam's razor, or the principle of parsimony, is familiar to most scientists. As we teach our students: given the choice between two equally fitting models, all else being equal it is generally preferable to choose the simpler, or more parsimonious, model. What is less well understood is how one might quantify parsimony. One promising approach is the concept of model *fit propensity* (Preacher, 2006) or *complexity* (Myung et al., 2005; Pitt et al., 2002). Here

Carl F. Falk  <https://orcid.org/0000-0002-4788-7206>

Michael Muthukrishna  <https://orcid.org/0000-0002-7079-5166>

We thank Victoria Savalei for her inspiration and early input on this project. We acknowledge the support of the Natural Science and Engineering Research Council of Canada (NSERC), (funding reference RGPIN-2018-05357 and DGEGR-2018-00083). A preprint of an earlier version of the article is available at: <https://arxiv.org/abs/2007.03699>.

The authors contributed equally to this article. Author order is alphabetical.

Correspondence concerning this article should be addressed to Carl F. Falk, Department of Psychology, McGill University, 2001 McGill College, 7th Floor, Montreal, QC H3A 1G1, Canada. Email: carl.falk@mcgill.ca

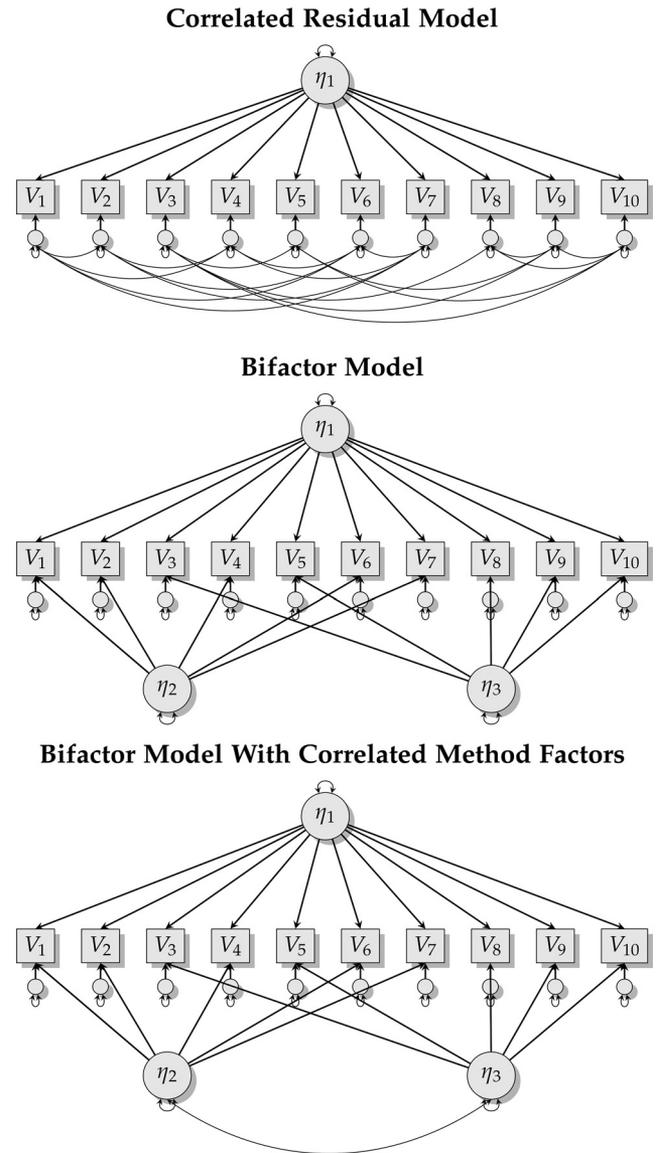
we will use *fit propensity* to avoid confusion with other uses of the term *complex*. Fit propensity is sometimes described as the “complement” of parsimony (Preacher, 2006, p. 230). The basic idea behind fit propensity is that some models will simply do a better job of fitting a wider range of data. These models are less parsimonious. Thus, the process of model selection needs to consider not just model fit, but fit propensity. The ideal theoretically derived model will have both better fit and lower fit propensity than a competing model. But in practice, there is likely to be a tension between fit and fit propensity. In other words, for a model to be both useful and generalizable, a balance must be struck between fitting real data, and parsimony in not also fitting random data (fit propensity can be defined as the propensity to fit random data; Bonifay, 2015; Cudeck & Browne, 1983; Marsh & Balla, 1994; Reise et al., 2013).

Parsimony is sometimes described as a function of degrees of freedom. For example, Marsh and Balla (1994) defined parsimony as “the ratio of degrees of freedom in the model being tested and degrees of freedom in the null model” (James et al., 1982; Mulaik et al., 1989, p. 188). It is thus tempting to equate parsimony with the degrees of freedom of a model such that fewer estimated parameters (and higher *df*) corresponds to more parsimony. However, it is possible to have models with the same number of estimated parameters, but where one has better propensity to fit random data (Bonifay & Cai, 2017; Preacher, 2006). Indeed, a model may even have more estimated parameters than an alternative, but have lower fit propensity and therefore more parsimony (Pearl & Verma, 1995). The configuration of the model (number of latent factors, paths among variables) and functional form of relationships among variables also affects fit propensity. In sum, prior research has identified both the number of estimated parameters and the functional form or configuration of the model as both contributing to fit propensity (Myung et al., 2005; Preacher, 2006). Thus, fit indices that adjust for degrees of freedom, such as Tucker-Lewis index (TLI; Tucker & Lewis, 1973), comparative fit index (CFI; Bentler, 1990), and root mean square error of approximation (RMSEA; Steiger & Lind, 1980) or commonly used information criterion, such as Akaike information criterion (AIC) and Bayesian information criterion (BIC) that have adjustments based on the number of estimated parameters are coarse in how they treat the role of fit propensity in model selection.

Fit Propensity and the Rosenberg Self-Esteem Scale

To understand the importance of fit propensity, consider the Rosenberg Self-Esteem Scale (RSES; Rosenberg, 1965). The RSES is perhaps the most widely used self-report instrument for the measurement of self-esteem. It contains 10 five-point Likert-type items. The RSES is often used by applied researchers to represent a single construct: self-esteem. Higher scores indicate higher self-esteem for five positively keyed items (Items 1, 2, 4, 6, and 7), and five negatively worded or reverse keyed items (Items 3, 5, 8, 9, and 10). The RSES is regularly used, but is the subject of ongoing investigations to examine the confirmatory factor analysis models that may represent it; a single factor model rarely fits RSES data adequately. In a recent example, Donnellan et al. (2016) fit 10 different models to RSES data ($N = 1,127$). Of these, three models stood out as having superior fit: (a) a global factor with correlated residuals among positively and negatively worded items;¹ (b) a bifactor model with method factors for positively and negatively worded items; and (c)

Figure 1
Rosenberg Self-Esteem Models



the same bifactor model, but with correlated method factors (see Figure 1). For illustration, we replicated the original analyses using *lavaan* (Rosseel, 2012), and results for these models and a single factor model are presented in Table 1.²

On the one hand, such well-fitting models may make intuitive sense. All three models account for additional dependencies beyond a single factor, and may be appropriate to the extent that positively worded items share some dependency, as do negatively worded items. On the other hand, one might question whether these models fit for other reasons. Are they parsimonious? The correlated residual

¹ With one residual fixed to zero for identification.

² To account for ordered categorical data, maximum likelihood with robust corrections (i.e., estimator = “MLR”) was employed here and in the original paper.

Table 1
Rosenberg Self-Esteem Model Fit

Model	χ^2	df	TLI	CFI	RMSEA	AIC	BIC	SIC	RMSR
1. Correlated residual	60.59	16	0.97	0.99	0.05	26,343	26,589	26,621	0.02
2. Bifactor	154.88	25	0.94	0.97	0.07	26,437	26,638	26,652	0.03
3. Correlated bifactor	135.09	24	0.95	0.97	0.06	26,413	26,619	26,632	0.02
4. Single factor	872.91	35	0.72	0.78	0.15	27,346	27,497	27,508	0.08

Note. TLI = Tucker-Lewis index; CFI = comparative fit index; RMSEA = root mean square error of approximation; AIC = Akaike information criterion; BIC = Bayesian information criterion; SIC = stochastic information complexity; RMSR = root mean square residuals.

model essentially accomplishes a similar task as the bifactor model in modeling dependencies among similar items, but does so with even more additional model parameters. Does this come with a cost to fit propensity? Indeed, from a traditional standpoint, these models have the most estimated parameters of all 10 models examined: 39, 30, and 31, respectively, with only 20 for the single factor model. The original article primarily considered aforementioned fit indices that make adjustments based on degrees of freedom or the number of estimated parameters: TLI, CFI, RMSEA, AIC, and BIC. The only other information regarding model fit are the chi-square test of fit, and root mean square residuals (RMSR)—a transformation of the difference between sample covariances and recovery of covariances by the model. In all cases, there is either no adjustment for parsimony or only a coarse-grained adjustment for degrees of freedom.

Bonifay et al. (2017) argued that the bifactor model may be good at fitting random noise—that it lacks parsimony. For instance, Bonifay and Cai (2017) found that a bifactor model with two uncorrelated method factors had higher fit propensity in general than two hierarchical models with discrete latent variables with the same number of parameters. In also examining the RSES, Reise et al. (2016) found that the bifactor model with uncorrelated method factors helped explain inconsistent response patterns, but that a single factor model was sufficient for the majority of participants. Note that the fit propensity of the additional models considered by Donnellan et al. (2016) have not been studied. One might then also wonder—how much more fit propensity does a correlated residual model have above and beyond a bifactor model? Or, does adding a single correlation among method factors substantially change fit propensity? How does the fit propensity of such models compare to a single factor model? Does fit propensity correlate with number of parameters? Is it possible that such models tend to fit the data well, not because they are close approximations of reality, but that such models tend to fit any data, even random data, very well? And even more broadly, does relative fit propensity depend on which fit index is examined?

Testing Fit Propensity

The types of questions we ask above provide clues about parsimony that are not easily answered by number of parameters or traditional fit indices. They can, however, be understood through a study of fit propensity.³ A popular method of studying fit propensity requires repeated generation of random data from a data space and fitting the models of interest (Preacher, 2006). Information regarding model fit can then be recorded over a large number of replications and summarized to provide a sense of how well the models fit such random data. Preacher (2006) introduced the concept of fit

propensity to structural equation modeling (SEM) over a decade ago. Further research has been limited, perhaps in part due to a lack of easy to use and efficient software tools for evaluating fit propensity. Preacher's (2003) original code was written in FORTRAN and had a few limitations, such as a Markov chain Monte Carlo (MCMC) algorithm that took a long time to generate random correlation matrices, restriction to only positive correlations, use of a lesser-known software program for fitting models (RAMONA 4.0 for DOS; Browne & Mels, 1990 as cited in Preacher, 2006), and support for few fit indices (only RMSR was studied).⁴

We aim to support further researchers in considering fit propensity of their models by providing an *R* package: *ockhamSEM*. The *ockhamSEM* package offers easy-to-use and highly flexible software built on the popular *lavaan* (Rossee, 2012) package. We hope that *ockhamSEM* will be used for the study of fit propensity by applied researchers investigating models of interest, for classroom demonstrations, or the further study of fit propensity itself and related methodological challenges by quantitative methodologists.

Investigating fit propensity requires generating random correlation matrices, which are computationally intensive. The *ockhamSEM* package provides several innovations in terms of both computational efficiency and the reporting of fit propensity. In particular, random correlation matrices can be generated using the *onion* method by Lewandowski et al. (2009), as well as Preacher's (2006) original MCMC algorithm. The onion method exploits known properties of elliptically contoured distributions applied to a *k*-dimensional hypersphere to provide a space of correlation matrices that can be sampled. This sampling results in the generation of correlation matrices much faster than the MCMC method, which involves iteratively generated random draws, where as with many MCMC methods, most are discarded. Using the onion method, thousands of large correlation matrices can be generated in seconds. We discuss these methods in further detail with additional resources in the Appendix.

Calculations can be performed in parallel using the multiple processing cores common in modern personal computers and computing clusters. Random correlation matrices can be restricted to all positive correlations, or both positive and negative correlations (indeed, other arbitrary restrictions can also be implemented). We also provide support for the full range of fit indices available from

³In our replication, an additional fit index, stochastic information complexity (SIC; Hansen & Yu, 2001), is reported and that could be used for adjustment of model fit that is more in line with fit propensity (Bonifay & Cai, 2017; Preacher, 2006), but as we discuss later in this article, does not immediately provide intuitive information regarding fit propensity.

⁴We thank Kris Preacher for graciously providing us this FORTRAN code, which also appears in his dissertation.

Table 2
Overview of Examples

Example	# Models	Model description	Main purpose/illustrated features
A	2	Two 3-variable models	Basic use of code, algorithms for correlation matrix generation, parallel processing, equal <i>df</i> models, empirical ECDF plots, quantiles
B	2	Factor and simplex models	Positive vs. negative correlations, equal <i>df</i> models, model convergence
C	4	Four RSES models	Other fit indices (CFI, RMSEA, TLI), Euler plots, saving of correlation matrices and fitted models

Note. TLI = Tucker-Lewis index; CFI = comparative fit index; RMSEA = root mean square error of approximation; RSES = Rosenberg Self-Esteem Scale; ECDF = empirical cumulative distribution function.

lavaan. Finally, additional numerical and graphical summaries are provided, going beyond those originally presented by Preacher (2006).

Our work is related to some recent research on fit propensity and model similarity. In particular, Bonifay and Cai (2017) describe methods for studying the fit propensity of item response models with categorical observed variables. Given the unification of item response models and SEM under a unified latent variable modeling framework (e.g., Skrondal & Rabe-Hesketh, 2004), this work is related to the present research. However, it does not address continuous observed variables and none of the underlying code was provided. We take the “ameoba” plots presented by Bonifay and Cai (2017) as inspiration for some visualizations we present later in this article. In addition, Lai et al. (2017) address methods for examining model similarity using mostly scatter plots and line graphs of fit indices, and may be helpful for visualizing whether some models are equivalent or nested (but see Bentler & Satorra, 2010). These authors mention difficulty in generating data from random correlation matrices, and instead opt for data generation from a restricted space that is a mixture of the correlation matrices implied by two competing models. Thus, model similarity rather than fit propensity was the main focus of this previous work.

This article is organized into the following sections. The first section provides a brief description of our implementation of the R code. We then illustrate concepts of fit propensity and basic features of the code in the context of several initial examples from Preacher (2006), and the RSES example. Finally, we conclude with a discussion of additional innovations and alternative ways to compare models.

Illustrative Examples

We present three examples to illustrate the basic procedure and concepts used to study fit propensity, including visualization and summaries of results (see Table 2). The first two examples expand upon those initially presented by Preacher (2006). We note that while the general pattern of results remains similar in our implementation, there may be minor discrepancies for a number of reasons.⁵ The final example concentrates on the debate around the RSES and what a study of fit propensity can provide. The first example is fully illustrated in-text with complete R code. The code for additional examples is available in the online supplementary materials.

Example A: Fit Propensity Basics

We will use the two 3-variable models depicted in Figure 2 as our first example (see also Preacher, 2006, p. 228). In Model 1A, V_3 is regressed on V_1 and V_2 , with the covariance among V_1 and V_2 restricted to zero. Model 2A represents a causal chain in which V_2 is regressed on V_3 , which is in turn regressed on V_1 , yet there is no direct path from V_1 to V_2 . These have the same number of estimated parameters (five) and do not represent equivalent models, despite the only difference being the direction of the relationship between V_2 and V_3 . The study of fit propensity is well suited for answering which model has a tendency to yield better fit. Although these models may seem trivially simple, the answer to this question is not so easy to see without the additional work we present below.

General Procedure and Code

The procedure to study fit propensity that we illustrate here follows several steps:

1. Definition of the model(s) of interest.
2. Generation of n random correlation matrices.
3. Fitting the models of interest to the n random correlation matrices.
4. Recording information regarding model fit for each model and correlation matrix.
5. Summaries of model fit using text, graphical displays, and measures of effect size (e.g., Komolgorov-Smirnov).

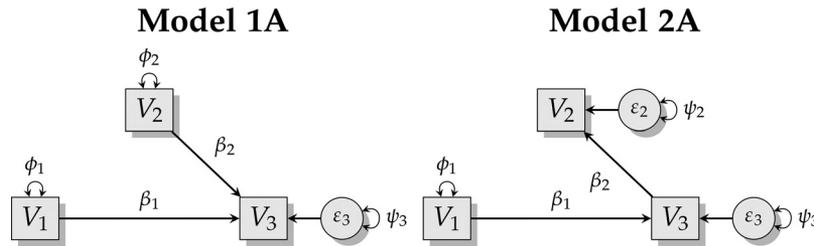
The core custom code used in this paper are included in the *ockhamSEM* package.⁶ Underlying innovations and the methods for generating random correlation matrices are discussed in the Appendix. The package can be loaded with the following R code snippet:

```
library(ockhamSEM)
library(parallel)
```

⁵ Different SEM program with different default estimation options, different handling of nonconverging models (*lavaan* does not allow calculation of some fit indices), etc. Preacher (2006) also used ordinary least squares for estimation, whereas we used maximum likelihood. For this paper, we used *lavaan* version 0.6-5 and R version 3.6.0.

⁶ <https://github.com/falkcarl/ockhamSEM>.

Figure 2
Two 3-Variable Models



Step 1

First use *lavaan* model syntax to define Model 1 and Model 2:

```
mod1a <- 'V3 ~ V1 + V2
          V1 ~ 0*V2'
```

```
mod2a <- 'V3 ~ V1
          V2 ~ V3'
```

Next, two models are fit to data. We expect that this will be the most typical use of studies of fit propensity for applied researchers—two or more substantive models are of interest in particular, because of debates over which is most appropriate for real data. We simply require a fitted *lavaan* model using data that has the same variable names (V1 through V3 in this case) as the above syntax indicates. Alternatively, we may fit the data to some covariance matrix. The following uses the latter strategy in creating an identity matrix:⁷

```
p<-3 # number of variables
temp_mat <- diag(p) # identity matrix

# set row and column names
colnames(temp_mat) <- paste0("V", seq(1, p))
rownames(temp_mat) <- paste0("V", seq(1, p))
```

We then fit the two models using the *sem* function from the *lavaan* package, though note that any function that returns a fitted model of class *lavaan* could be used, such as the *cfa*, *sem*, or *lavaan* functions:

```
mod1a.fit <- sem(mod1a, sample.cov=temp_mat,
                 sample.nobs = 500)
mod2a.fit <- sem(mod2a, sample.cov=temp_mat,
                 sample.nobs = 500)
```

At *this* step, any special options regarding estimation can be passed to *sem*. Our later code will attempt to use these options when fitting models for investigating fit propensity. For instance, here we specify a particular number of observations for this data (*sample.nobs = 500*), although for many fit indices of interest this information is inconsequential. One may ask *lavaan* to mimic a different SEM program, use normal theory or Wishart likelihood, use a different optimizer, change the iteration limit for estimation, scale sample covariance matrices by $(N - 1)/N$, and so on (see *help(lavOptions)*). As long as any of these options are implemented when defining and fitting initial models, they will be

used when the models are fit to randomly generated correlation matrices. However, the ability to do so-called robust corrections or use any estimation approach that requires raw data or a mean structure is not supported; the available options currently must work for model fitting when analyzing only a covariance matrix as input.

Steps 2 Through 4

Generation of random correlation matrices, fitting models to such matrices, and recording model fit are all accomplished by the *run.fitprop* function in the next code snippet:

```
res.on <- run.fitprop(mod1a.fit, mod2a.fit,
                     fit.measure="srmr", rmethod="onion",
                     reps=5000, onlypos=TRUE)
```

The initial arguments to this function are any number of fitted *lavaan* models, such as *mod1a.fit* and *mod2a.fit*. The remaining arguments must be named and are only required for taking explicit control over correlation matrix generation and saving of output. The *fit.measure* argument accepts a character vector that indicates what fit indices will be saved. Anything that matches named output from the *fitMeasures* command from *lavaan* can be used. Users are encouraged to run this command on already fitted models to see what available fit indices are possible (e.g., *fitMeasures(mod1a.fit)*). Here, we save only standardized root-mean-square residual, as indicated by "srmr," which in this case is equivalent to RMSR, because analyzed correlation matrices will already be standardized. RMSR is the fit index primarily studied by Preacher (2006) in his work on fit propensity as it provides a sense of model fit, unadjusted for the number of estimated parameters. We generate random correlation matrices using the onion method (*rmethod="onion"*), requesting 5,000 replications (*reps=5000*), and restricting to only positive correlations (*onlypos=TRUE*).

The result of the *run.fitprop* command in the code above is saved to *res.on* which is an object of class *fitprop* with several options regarding output that will be illustrated shortly. Before we proceed, suppose we wished to see whether results differ if we had instead used the MCMC algorithm to generate random correlation matrices. This latter approach should provide replication of

⁷ Another viable alternative involves generating data from the true models for 1A and 2A, such as with *lavaan*'s *simulateData* function.

Preacher (2006), but may be computationally slow. In this case, we may wish to use parallel processing for faster computations:

```
cl <- makeCluster(8)
res.mcmc <- run.fitprop(mod1a.fit, mod2a.fit,
  fit.measure="srmr", rmethod="mcmc",
  reps=5000, onlypos=TRUE, cluster=cl)
stopCluster(cl)
```

We create a cluster with 8 processing cores with the `makeCluster` command. The result, `cl`, is then passed to the `run.fitprop` function using the cluster argument. The change in `rmethod` to "mcmc" will result in use of the MCMC algorithm for correlation matrix generation. Finally, we shut down the cluster using the `stopCluster` command after we obtain the results.

Step 5

There are multiple different ways to then summarize results. Preacher (2006) primarily focused on empirical cumulative distribution function (ECDF) plots that we also illustrate here and will describe shortly. In particular, both `res.on` and `res.mcmc` are objects of class `fitprop` for which we have defined a default plot function. This allows us to simply use the `plot` command successively in order to generate ECDF plots (see Figure 3) of the requested fit indices and model(s):

```
plot(res.on)
plot(res.mcmc)
```

The argument `savePlot=TRUE` can also be specified and the result will be a list of `ggplot` objects containing ECDF plots corresponding to each fit index. This feature is useful if, for example, the user wishes to modify the legend, title, and so forth, of the resulting plot or otherwise customize output.⁸ For convenience, several additional options can be defined, such as to add custom names (`mod.lab`) for the two (or more) fitted models and the color palette used by `RColorBrewer` (Neuwirth, 2014).

```
plot1 <- plot(res.on, savePlot=TRUE,
  mod.lab=c("model 1A", "model 2A"),
  mod.brewer.pal="Set1")
plot2 <- plot(res.mcmc, savePlot=TRUE,
  mod.lab=c("model 1A", "model 2A"),
  mod.brewer.pal="Set1")
```

To explain ECDFs and Figure 3, suppose we collect all RMSR estimates for the 5,000 fitted models for Model 1A. We then sort these 5,000 estimates in order from lowest to highest. Next, we count the number of RMSR estimates at or below a particular value. For example, "what proportion of fitted models have an RMSR value of .25 or lower? .5 or lower?" Each curve in Figure 3 displays the answer to this question for each model separately and at many values of RMSR along the x -axis such that the lines appear to be continuous. For example, Model 2A had approximately 75% (or .75 as a proportion) of models that had an RMSR (or `srmr`) of .25 or better when correlation matrices were generated using the MCMC algorithm (see where .25 on the x -axis intersects with the dotted blue line on the left-hand panel). Model 1A had a smaller proportion (around .57 or so) of cases with an RMSR of .25 or better. This implies that the higher curve (for Model 2A)

indicates better fit for more models, and therefore more fit propensity when examining RMSR.

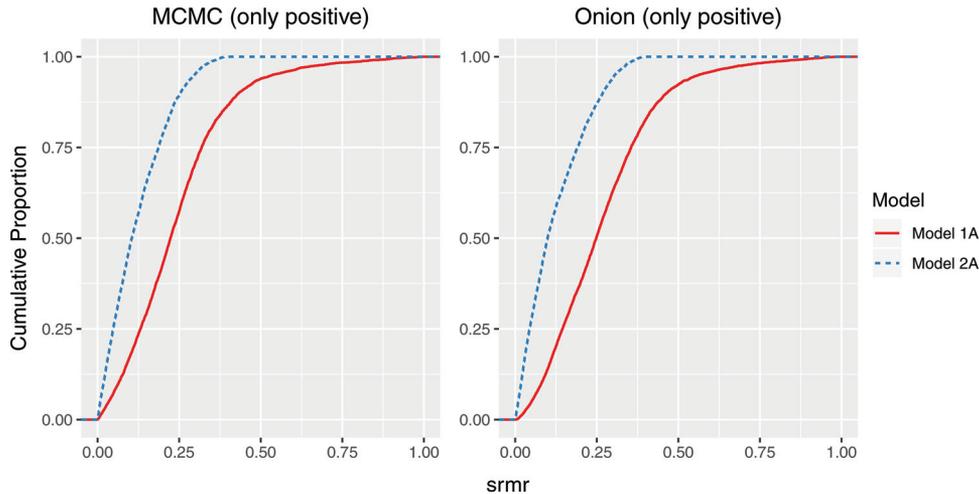
For the most part, the results displayed here replicate those of Preacher (2006): model 2A appears to have more fit propensity in that there is a higher proportion of RMSR values that are relatively small, and this result tends to hold for both MCMC and Onion methods. Had either correlation matrix generating method allowed for both positive and negative correlations, the basic pattern in fit propensity regarding Models 1A and 2A would have been similar and the interested reader is encouraged to verify this assertion.

Default `print` and summary methods are also available for `fitprop` objects. `summary` will provide some diagnostic information regarding whether any nonconvergent models were encountered, selected quantiles of the resulting fit statistics, and effect sizes to help quantify the differences in fit between estimated models. For example, instead of eyeballing Figure 3, we can ask for the value of `srmr` that corresponds to a cumulative proportion of .25, .5, and .75 for both models using the following:

```
summary(res.mcmc, probs=c(.25, .5, .75))
##
## Quantiles for each model and fit measure:
##
## Model 1
## srmr
## 25% 0.134
## 50% 0.229
## 75% 0.330
##
## Model 2
## srmr
## 25% 0.047
## 50% 0.105
## 75% 0.187
##
## Information about replications for each
## model and fit measure:
##
## Model 1
##
## Mean across replications
## srmr
## 0.298
##
## Median across replications
## srmr
## 0.229
##
## Number of finite values
## srmr
## 5000
##
## Number of NA values
## srmr
## 0
##
## Model 2
```

⁸ Which is how we added all of the plot titles, combined multiple plots into a single figure, customized axis dimensions and labels, and so on in the present article.

Figure 3
ECDF Plots Comparing Fit Propensity of Models 1A and 2A



Note. ECDF = empirical cumulative distribution function; MCMC = Markov chain Monte Carlo. See the online article for the color version of this figure.

```
##
## Mean across replications
## srmr
## 0.123
##
## Median across replications
## srmr
## 0.105
##
## Number of finite values
## srmr
## 5000
##
## Number of NA values
## srmr
## 0
##
## Effect sizes for differences in model fit:
##
## srmr
##
## Model 1 versus Model 2
## Cohen's d: 0.607
## Cliff's delta: 0.512
## Komolgorov Smirnov: 0.374
```

These results tell a similar story to the graphical summaries provided. For example, 50% of Model 1A results had an RSMR of .229 or lower, whereas 50% of Model 2A had an RSMR of .105 or lower. Alternatively, one may examine the mean or median values for RSMR. We also see that there were apparently no models where nonconvergence was a problem, because there are no NA values for any srmr estimates.

At the end of the output, differences between all estimated models and all recorded fit indices are quantified using three effect sizes: Cohen's d (Cohen, 1988), Cliff's delta (Cliff, 1996), and a Komolgorov-Smirnov coefficient (K-S). Although additional effect sizes could be easily added, we initially chose these three for several

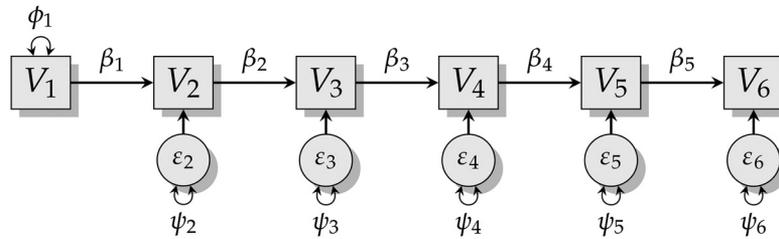
reasons. First, Cohen's d is likely familiar to many researchers in the social sciences. Here, we see that a value of .61 is observed, which is typically considered as between a "medium" and "large" effect size in research settings. Here it is indicating that Model 1A's distribution for RSMR is on average .61 SD higher than Model 2A. Cohen's d has a nice conceptual interpretation as the number of standard deviation units that separate the RSMR distributions for Models 1A and 2A. Because Cohen's d is typically not considered a robust effect size measure (e.g., Wilcox, 2012), we included Cliff's delta, which is robust to outliers and skewness, and K-S as additionally sensitive to variability across two distributions. These may be more unfamiliar to researchers.

Conceptually, Cliff's delta is a difference between two probabilities: (a) the probability that a value in the first distribution is greater than that in the second; and (b) the probability that a value in the second distribution is greater than that in the first. It is computed in part by comparing each observation from one distribution versus all observations in a second distribution. Cliff's delta ranges between -1 and 1 , with values close to zero indicating no difference between two distributions. The K-S coefficient is the maximum difference between two ECDFs over all cumulative probabilities of some measure (in our case, a given fit index). Therefore, as ECDFs also are bound between 0 and 1, K-S ranges between 0 and 1, with 1 indicating a larger discrepancy between two distributions.⁹ To our knowledge, there are no accepted guidelines for Cliff's delta or the K-S statistic equivalent to "small," "medium," and "large" for Cohen's d , in part because the statistics are readily interpretable. For example, in the output above, the difference in probabilities is 51.2% with values tending to be higher in Model 1, and the largest discrepancy in ECDFs is .374.

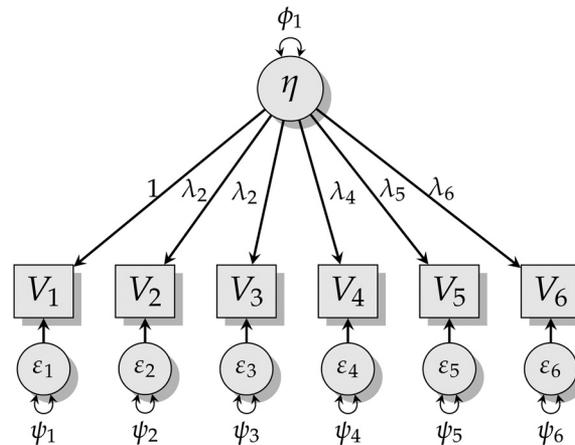
⁹ Although K-S could also yield a p -value, this value would largely be dependent on the number of replications chosen for the fit propensity analysis.

Figure 4
Simplex and Constrained Factor Analysis Model

Simplex Model (1B)



Factor Model (with loading constraint; 2B)



Example B: Simplex Model Versus Factor Model

Next, we extend an example given by Preacher (2006) in which we compare a simplex model (1B) with a single factor model with a loading equality constraint for the second and third loadings (2B; Figure 4).

Although it seems unlikely that researchers would consider these two alternative models for the same dataset, they have the same degrees of freedom and will yield different fit propensity. Furthermore, these examples are useful for demonstrating the impact of a restriction on the data space. In particular, we compared the fit propensity using `smmr` for these two models by crossing two conditions: Correlation matrix generation (MCMC vs. onion) and positivity of correlations (all positive vs. positive and negative). The option for obtaining both positive and negative correlations can be achieved by setting `onlypos=FALSE` when using the `run.fitprop` command.¹⁰

Here, we see that the restriction of the data space to only positive correlations makes a substantial difference, yet also note that Preacher (2006) only examined positive correlations. The factor model (2B) generally has more fit propensity than the simplex model (1B) if only positive correlations are considered as shown in the two plots on the left-hand side of Figure 5. However, the fit propensity for the factor model is lower if we consider the possibility that the data space may include both positive and negative correlations as shown on the right-hand side of Figure 5. In retrospect, this makes intuitive sense given that the second and third

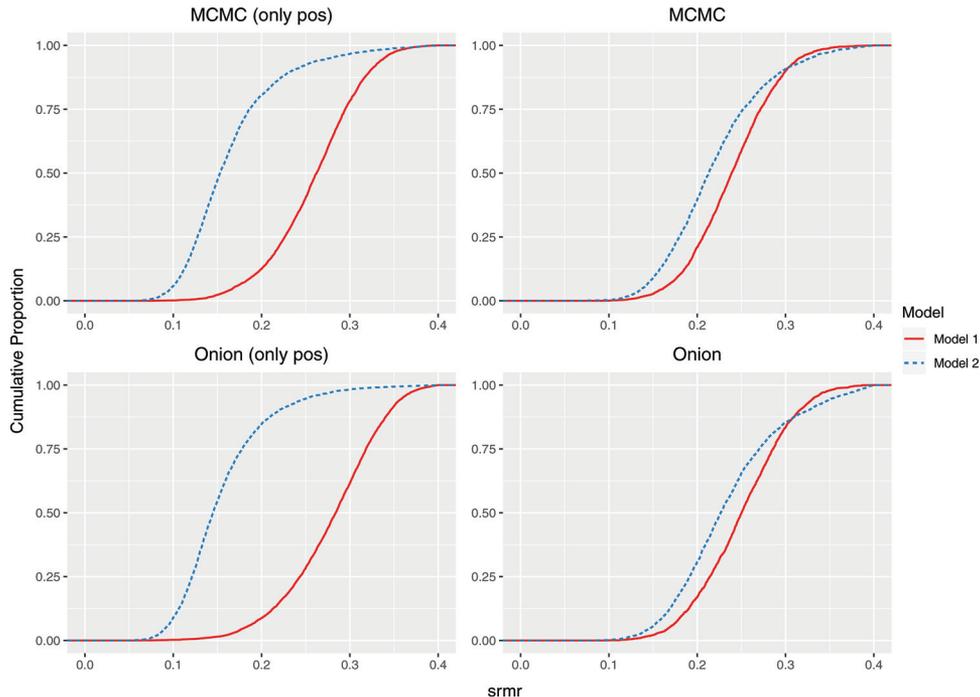
loadings of the factor model are constrained equal. At minimum, such a restriction would likely only make sense if the correlations among these two items have the same sign. For example, if it were the case that Item 2 tended to have negative correlations with other items, but Item 3 had positive correlations, the two loadings would seem to have opposite signs. Model fit would then deteriorate due in part to constraining these loading estimates to be equal.

We also note that substantial estimation problems were encountered when the correlation matrices included both positive and negative values. In particular, 2,462 models failed to converge for the factor model when negative correlations were allowed for the onion method, but zero and 16 failed to converge for the MCMC and onion (only positive) conditions, respectively. Such convergence failures are indicated as NA values for particular fit indices. Such information regarding the number of valid replications is available via the `summary` command.

The above plots and summary information are based on replications where *both* models converged as this is the default behavior. We can, however, change output so that results are based on the available number of replications for either model, regardless of whether the replications are the same by setting `samereps=FALSE` when using the `plot` command (For example, `summary(res.mcmc, samereps=FALSE)`). Thus, the quantiles and plots for Model

¹⁰R code for this and for all following examples appears in the online supplementary materials.

Figure 5
ECDF Plots Comparing Fit Propensity of Models 1B and 2B



Note. ECDF = empirical cumulative distribution function; MCMC = Markov chain Monte Carlo. See the online article for the color version of this figure.

1B could be based on a slightly different subset of random correlation matrices than Model 2B. However, this does not appear to substantially change the resulting plots or summary information, and the interested reader is encouraged to verify this observation.

Models for the Rosenberg Self-Esteem Scale

In our final example, we return to the debate regarding an appropriate model for the RSES. In particular, we investigated the fit propensity of the four models considered in Table 1. For reference, we label these as follows: correlated residuals (Model 1C), bifactor (Model 2C), bifactor with correlated method factors (Model 3C), and a single factor model (Model 4C). Due to more models and the possibility of nonconvergence, we increased the number of replications to 10,000. As the RSES items typically intercorrelate positively (after reverse-coding), we used the onion method with only positive correlations. For fit indices, we additionally demonstrate the performance of TLI, CFI, RMSEA, and RMSR. The model chi-square is also saved to later check model nesting.

Since information is saved from four fit indices (TLI, RMSR, CFI, RMSEA) and the model chi-square test of fit, we illustrate two additional options for aiding in generating plots. First, we can either obtain plots for all fit indices (the default behavior), or request plots just for a particular fit index by passing the `whichfit` argument to the `plot` function. `whichfit` accepts a character vector in the same fashion as `fit.measure` (resulting in plots for selected fit indices being displayed successively), or may be left at

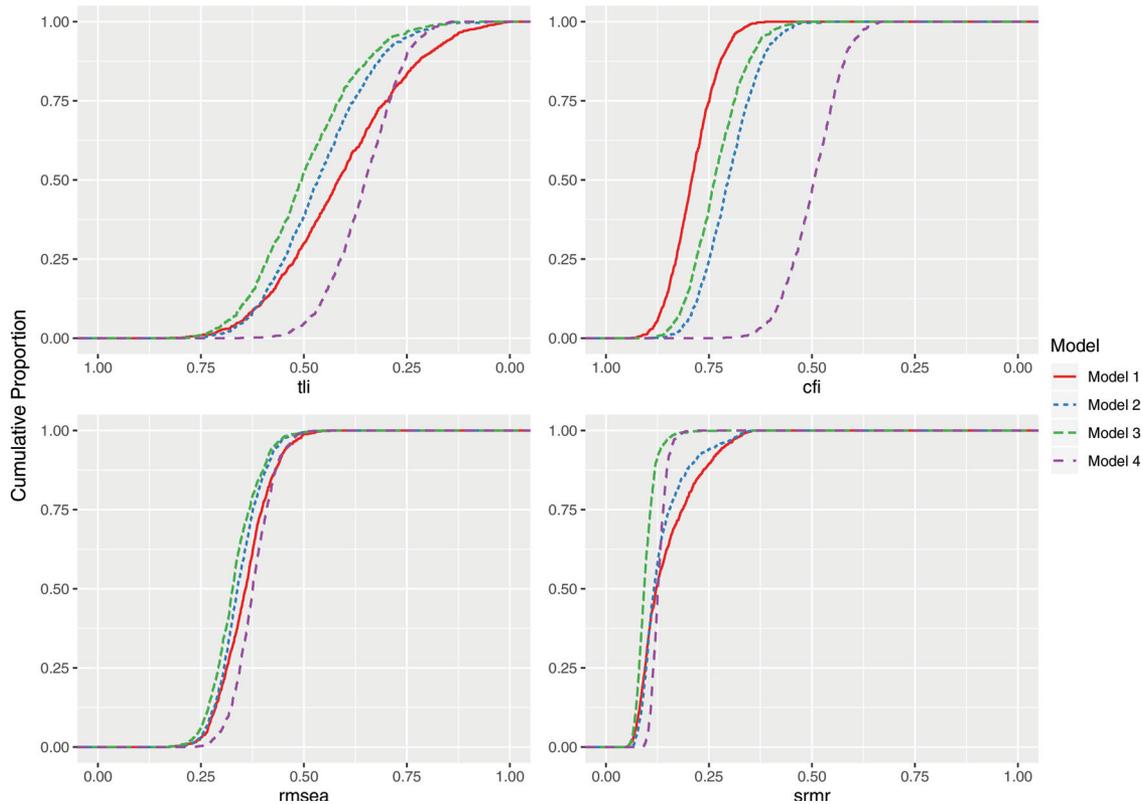
its default setting (successively showing plots for all saved fit indices).

Second, an additional argument can be changed to aid in interpreting TLI and CFI. Note that lower values of RMSR typically indicate better fit, but *higher* values of TLI and CFI correspond to better fit. The `lower.tail` argument accepts a logical vector of the same length as the number of stored fit indices (i.e., the same length as `fit.measure`). By default, the vector sets all elements to `TRUE`. Setting the corresponding element to `FALSE` for CFI ensures that the ECDF plots start with higher numbers on the left-hand side of the *x*-axis, making interpretation of such plots visually similar to those for RMSR. In the following code snippet, assuming results are stored in `res.rses` we only plot results for TLI, CFI, RMSEA and RMSR, and we make the above corresponding changes for the ECDF plot for TLI and CFI.

```
plot(res.rses, whichfit=c("tli", "cfi",
                        "rmsea", "srmr"),
     lower.tail=c(FALSE, FALSE, TRUE, TRUE))
```

We next examine each fit index separately (see Figure 6). CFI is perhaps the easiest to interpret, yet also raises the most concerns about its use as a measure of model fit. In particular, very clearly the four models are ordered in terms of the number of estimated parameters, with models with fewer degrees of freedom (more estimated parameters) having more fit propensity according to CFI. This is intuitive to the extent that more parameters leads to better fit. However, this is concerning since the data are random

Figure 6
ECDF Plots Comparing Fit Propensity of Models 1C, 2C, 3C, and 4C



Note. ECDF = empirical cumulative distribution function; TLI = Tucker-Lewis index; CFI = comparative fit index; RMSEA = root mean square error of approximation. See the online article for the color version of this figure.

and CFI should include an adjustment for parsimony. That is, the adjustment included in CFI does not appear to equate the fit of the resulting models. Even for some nontrivial percentage of replications, CFI even exceeds .80 or so (e.g., CFI averages around .79 for the correlated residual model). This large percentage of models that have good fit for random data should be concerning to researchers using CFI.

TLI and RMSEA tend to have similar patterns of results to each other. Almost no replications are in the range of what is usually considered acceptable fit, and models are ordered in a nonintuitive way: The bifactor model with correlated method factors had the highest fit propensity, followed by the bifactor model, correlated residual model, and finally the one factor model. Thus, the ordering of fit propensity is not in line with the *df* or number of estimated parameters in that the correlated residual model did not have the highest fit propensity.

Differences among the models are also more difficult to detect visually. Additionally, effect sizes for the differences across these distributions also tends to be smaller. For example, K-S yields values between .11 and .40 for all pairwise differences for RMSEA. For RMSR, the bifactor model with correlated method factors yielded the highest fit propensity with the other models mixed. For instance, the difference between the bifactor model and the correlated residual is negligible (Cliff's $\delta = .05$, K-S = .11). However,

we may also raise some concerns about the utility of RMSR for these studied models. For instance, many RMSR estimates were below .1, with a full 60% of the correlated bifactor models below this threshold.

Euler plots can further aid in helping visualize relative fit of the models for the same randomly generated correlation matrices and are similar to Venn diagrams. These are inspired by the “ameoba” plots of Bonifay and Cai (2017). In Euler plots, the area of each circle (or ellipse) is proportional to the number of cases that meet some criteria. Overlap among circles indicates overlap in the sets of correlation matrices that meet this criteria.¹¹

Take for example, TLI. The ECDF plot seems to indicate that the correlated bifactor model had about 50% of replications with a TLI of .5 or better, whereas the single factor model had only about 8% of replications with a TLI of .5 or better. But, does the correlated bifactor model *always* fit better according to TLI? In other words, consider the entire set of correlation matrices that had TLI of .5 or better for the correlated bifactor model and the set of correlation matrices that had TLI of .5 or better for the single factor model. Are there some replications where the single factor model had a TLI better than .5, but the correlated bifactor model did not?

¹¹ Code using nVennR for generating plots is also currently in development.

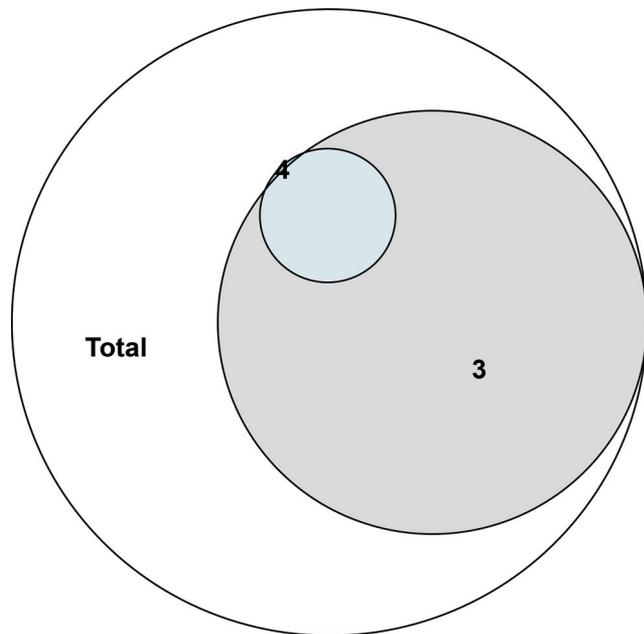
This information regarding overlapping sets of replications that meet this TLI criterion is depicted in a Euler plot:

```
plot(res.rses, type="euler", whichfit="tli",
     whichmod=c(3,4), cutoff=.5,
     lower.tail=FALSE)
```

Figure 7 was generated with the above code. Larger ellipses are conceptually similar to occupation of a wider area of the data space, or more fit propensity, and directly represent the proportion of correlation matrices for each model where TLI was better than .5. Here we see that the correlated bifactor model takes up the most space relative to the single factor model, indicating that it had a higher proportion of cases with TLI of .5 or better. If one model *always* fits better than another, we would expect that the worse fitting model would be completely contained within the better fitting model's ellipse. Such a case would indicate that the entire set of replications that met the TLI $\geq .5$ criterion for the lesser fitting model was also met by the better fitting model. That each model has some part of the ellipse that is not covered by the other suggests that there are some replications where each model fits better than the others. Here we see that the single factor model is *not* completely contained within the correlated bifactor's space. This means that there are some correlation matrices for which the single factor model has TLI better than .5, but the bifactor model does *not* have TLI better than .5. In other words, there are some correlation matrices for which the single factor model has better TLI than the correlated bifactor model.

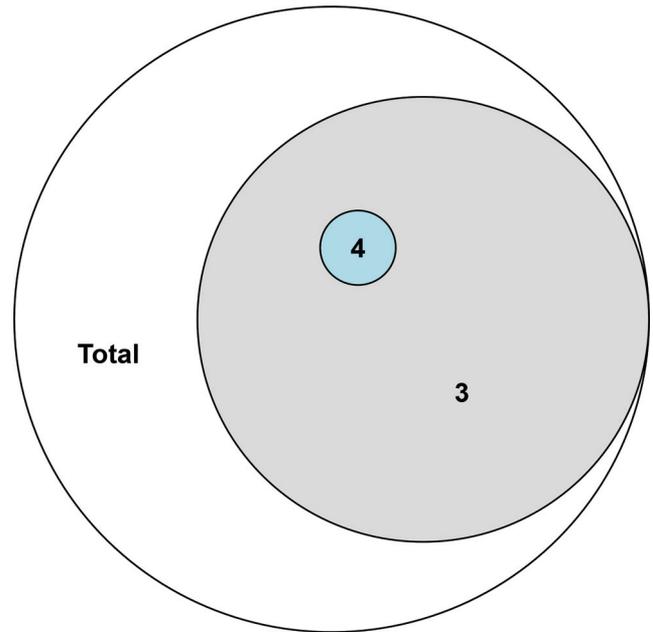
Although one might be under the impression that models with more factors should always fit better than a single factor model, this result typically applies to the chi-square test of fit and to the case of nested models. We can, however, directly check to see if

Figure 7
Euler Plot for TLI at Cutoff = .5



Note. TLI = Tucker-Lewis index. See the online article for the color version of this figure.

Figure 8
Euler Plot for Chi-Square Test at Cutoff = 3,000



Note. See the online article for the color version of this figure.

the 1-factor model is nested with a correlated bifactor model by also examining Euler plots for the chi-square test of model fit (see Figure 8). Here, we used a cutoff value of $\chi^2 = 3,000$. The single factor model has no unique set that is not completely encompassed by the correlated bifactor model. In other words, there do not appear to be any single factor models that are better than $\chi^2 = 3,000$ but for which a correlated bifactor model is not as good. These results are what we would expect with a unidimensional model being nested with the correlated bifactor model.¹²

Finally, SIC as reported in Table 1 has been suggested by others (Bonifay & Cai, 2017; see also Hansen & Yu, 2001; Preacher, 2006) as a computationally feasible index that is similar to AIC and BIC, but may make a more fine-grained adjustment than that based on the number of estimated parameters.¹³ In this particular case, SIC suggests the same ordering of model fit, and therefore the same selected model, as these other more traditional indices. While *sem-Tools* (Jorgensen et al., 2019) now provides computation of SIC from fitted *lavaan* models, we note that SIC does not immediately provide us with the useful information regarding fit propensity and other fit indices that may be of interest. The method of studying fit

¹² As a small aside, proper examination of model nesting is contingent upon good starting values and successful estimation. For example, *lavaan*'s defaults for starting values can sometimes converge on a local optimum and thereby yield a higher chi-square value even for a more general model. Should the researcher wish to examine particular replications, randomly generated correlation matrices and fitted models can be saved as lists to the `res.rses` object.

¹³ Like AIC and BIC, SIC is computed in part from the negative log-likelihood, but makes an adjustment based on the log of the determinant of the information matrix for the item parameters. Thus, more parameters may lead to a greater adjustment, but to the extent that parameter estimates are asymptotically correlated the adjustment may be less.

propensity that we have discussed thus far also allows restriction of the data space, which is also not possible with SIC.

Discussion and Conclusion

SEMs are a flexible method for representing complex theories, a useful tool in moving toward a formal theoretical approach to the psychological sciences (Muthukrishna & Henrich, 2019). But in evaluating competing theories instantiated as different SEMs, researchers need to consider not only the fit of their models to the data, but the parsimony of that fit. Fit propensity is therefore an important consideration in evaluating the relative merit of competing theories.

We introduce and investigate fit propensity using several examples, including investigation of the Rosenberg Self Esteem Scale. These investigations reveal important caveats in evaluating model fit and multimodel inference. First, they make clear that model fit alone is not sufficient to select between competing models—fit propensity also needs to be considered. Second, they reveal that fit indices, particularly RMSR and CFI, evaluated in isolation can be misleading. For example, in the RSES example, CFI was found to have decent for fit for a large percentage of random correlation matrices—as high as 80% in some cases. Finally, this investigation provides a replication of Preacher’s (2006) original findings regarding fit propensity, extending these beyond positive correlation matrices. These examples also illustrate the usefulness of the *ockhamSEM* package. The package is intended to support researchers in considering the fit propensity of their SEMs, but also support applied researchers and methodologists in further investigating fit propensity itself and refining methods for evaluating fit propensity.

Several challenges remain that we hope will be addressed by methodological researchers. One practical avenue for refinement is computational complexity. In Preacher’s (2006) original code generating random correlation matrices was a significant computational bottleneck. We solved this problem by using the onion method (Joe, 2006; Lewandowski et al., 2009). Model fitting, however, remains computationally intensive, and this problem increases with more variables. Models with more variables may also encounter more estimation problems. Some avenues for solving this problem include integrating our R code with a structural equation modeling program that might fit models faster (e.g., Bentler, 2006; Muthén & Muthén, 1998–2010; Neale et al., 2016). OpenMx may be the most promising alternative as it also allows access to optimizers that better overcome local optimum or automate attempts at different starting values.

Another important area of further research is constraints on the space of data. Example B reveals how restricting the data space to only positive correlations versus allowing positive and negative correlations can affect fit propensity. Ideally the data space could be restricted based on the universe of possible data for a particular research question. In future research, we hope to investigate the generation of random correlation matrices from specific data spaces for a fixed number of factors (where the number of factors is less than the number of items), or from the entire data space under a particular theoretical model of interest. Lai et al. (2017) provided an initial attempt at something similar, but their approach is not guaranteed to generate uniformly from the entire data space under any given model. Additional tuning parameters are available under Lewandowski et al. (2009), but are challenging to translate for the applied researcher.

As our illustrations reveal, fit propensity and number of parameters should be interpreted independently when assessing parsimony.

Researchers can only attribute the difference in fit propensity to model specification, or make general statements about types of model specifications and their relative fit propensity when number of parameters are the same. For instance, Examples A and B illustrate cases where fit propensity is different even though competing models have the same number of estimated parameters. This will of course not always be the case in the real world, for example, when assessing two theoretically derived model specifications, as is the case with the final RSES example. Assessing fit propensity gives us another tool for claiming parsimony, even making the case for a model not only having fewer parameters, but also higher fit propensity (Bonifay & Cai, 2017; Pearl & Verma, 1995). For example, for some indices, the most unrestricted RSES model did not have the highest fit propensity. It is also not easy to separate the effects on fit propensity due to the number of estimated parameters from the configural (or functional) form of the model. For instance, even though TLI, CFI, and RMSEA make some adjustment for *df*, they yielded different conclusions in the RSES example.

Beyond this tutorial, some researchers may desire additional reading on the background literature that gave rise to fit propensity and its relation to other fields. To briefly elaborate, the minimum description length (MDL; Rissanen, 1978) principle is one key concept in information theoretic perspectives regarding model fit (e.g., see Bonifay, 2015) and is often used in cognitive, computer science, and machine learning. MDL considers the data to be composed of both noisy and systematic components. The goal is often to find a way to encode the systematic part in such a way that both systematic (i.e., model) and noisy parts (i.e., remaining error) can be most concisely encoded. Thus, MDL formalizes Occam’s razor and the tradeoff between a complicated model (i.e., not concisely describing the systematic part) and poor model fit (i.e., too much remaining error to describe). Importantly, the MDL perspective does not require that there is a “true” population model. However, it is often used for inductive inference as the model that allows description of the data in the most concise manner is often one that allows us to learn much from the data (it is useful) and should be considered the best model. Researchers wishing to have a better understanding of these concepts or indices more directly derived from MDL (such as normalized maximum likelihood) are referred to resources that we believe are accessible in part by focusing on such concepts within SEM and IRT frameworks (Bonifay, 2015; Bonifay & Cai, 2017; Preacher, 2003, 2006).

While there are practical considerations in evaluating fit propensity for model selection, it is clear that fit propensity cannot simply be ignored—parsimony is a crucial aspect of evaluating theories. That a model better fits the right data is insufficient evidence if the model also fits the wrong data. We hope to motivate researchers to always consider parsimony and fit propensity and provide them with the tools to do so.

References

- Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, *107*(2), 238–246. <https://doi.org/10.1037/0033-2909.107.2.238>
- Bentler, P. M. (2006). *EQS 6 structural equations program manual*. Multivariate Software Inc.

- Bentler, P. M., & Satorra, A. (2010). Testing model nesting and equivalence. *Psychological Methods, 15*(2), 111–123. <https://doi.org/10.1037/a0019625>
- Bollen, K. A., & Pearl, J. (2013). Eight myths about causality and structural equation models. In S. L. Morgan (Ed.), *Handbook of causal analysis for social research* (pp. 301–328). Springer Netherlands. https://doi.org/10.1007/978-94-007-6094-3_15
- Bonifay, W. E. (2015). *An integrative framework of model evaluation* [PhD thesis, Department of Psychology, University of California, Los Angeles].
- Bonifay, W. E., & Cai, L. (2017). On the complexity of item response theory models. *Multivariate Behavioral Research, 52*(4), 465–484. <https://doi.org/10.1080/00273171.2017.1309262>
- Bonifay, W. E., Lane, S. P., & Reise, S. P. (2017). Three concerns with applying a bifactor model as a structure of psychopathology. *Clinical Psychological Science, 5*(1), 184–186. <https://doi.org/10.1177/2167702616657069>
- Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach* (2nd ed.). Springer.
- Cliff, N. (1996). Answering ordinal questions with ordinal data using ordinal statistics. *Multivariate Behavioral Research, 31*(3), 331–350. https://doi.org/10.1207/s15327906mbr3103_4
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Erlbaum.
- Cudeck, R., & Browne, M. W. (1983). Cross-validation of covariance structures. *Multivariate Behavioral Research, 18*(2), 147–167. https://doi.org/10.1207/s15327906mbr1802_2
- Donnellan, M. B., Ackerman, R. A., & Brecheen, C. (2016). Extending structural analyses of the Rosenberg self-esteem scale to consider criterion-related validity: Can composite self-esteem scores be good enough? *Journal of Personality Assessment, 98*(2), 169–177. <https://doi.org/10.1080/00223891.2015.1058268>
- Grace, J. B., & Bollen, K. A. (2008). Representing general theoretical concepts in structural equation models: The role of composite variables. *Environmental and Ecological Statistics, 15*(2), 191–213. <https://doi.org/10.1007/s10651-007-0047-7>
- Hansen, M. H., & Yu, B. (2001). Model selection and the principle of minimum description length. *Journal of the American Statistical Association, 96*(454), 746–774. <https://doi.org/10.1198/016214501753168398>
- James L. R., Mulaik, S. A., & Brett, J. M. (1982). *Causal analysis, assumptions, models, and data*. Sage.
- Joe, H. (2006). Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis, 97*(10), 2177–2189. <https://doi.org/10.1016/j.jmva.2005.05.010>
- Jorgensen, T. D., Pornprasertmanit, S., Schoemann, A. M., & Rosseel, Y. (2019). *semTools: Useful tools for structural equation modeling*. <https://CRAN.R-project.org/package=semTools>
- Lai, K., Green, S. B., & Levy, R. (2017). Graphical displays for understanding SEM model similarity. *Structural Equation Modeling: A Multidisciplinary Journal, 24*(6), 803–818. <https://doi.org/10.1080/10705511.2017.1334206>
- Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis, 100*(9), 1989–2001. <https://doi.org/10.1016/j.jmva.2009.04.008>
- Marsh, H. W., & Balla, J. (1994). Goodness of fit in confirmatory factor analysis: The effects of sample size and model parsimony. *Quality and Quantity, 28*(2), 185–217. <https://doi.org/10.1007/BF01102761>
- Mulaik, S. A., James, L. R., Van Alstine, J., Bennett, N., Lind, S., & Stilwell, C. D. (1989). Evaluation of goodness-of-fit indices for structural equation models. *Psychological Bulletin, 105*(3), 430.
- Muthén, L. K., & Muthén, B. O. (1998–2010). *Mplus user's guide* (6th). Muthukrishna, M., & Henrich, J. (2019). A problem in theory. *Nature Human Behaviour, 3*(3), 221–229. <https://doi.org/10.1038/s41562-018-0522-1>
- Myung, I. J., Pitt, M. A., & Kim, W. (2005). Model evaluation, testing and selection. In K. Lamberts & R. Goldstone (Eds.), *Handbook of cognition* (pp. 422–436). Sage.
- Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H., & Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika, 81*(2), 535–549. <https://doi.org/10.1007/s11336-014-9435-8>
- Neff, K. D., Whittaker, T. A., & Karl, A. (2017). Examining the factor structure of the self-compassion scale in four distinct populations: Is the use of a total scale score justified? *Journal of Personality Assessment, 99*(6), 596–607. <https://doi.org/10.1080/00223891.2016.1269334>
- Neuwirth, E. (2014). *RColorBrewer: ColorBrewer palettes*. <https://CRAN.R-project.org/package=RColorBrewer>
- Pearl, J., & Verma, T. S. (1995). A theory of inferred causation. In D. Prawitz, B. Skyrms, & D. Westerstahl (Eds.), *Logic, methodology and philosophy of science ix* (Vol. 134, pp. 789–811). Elsevier. [https://doi.org/10.1016/S0049-237X\(06\)80074-1](https://doi.org/10.1016/S0049-237X(06)80074-1)
- Pitt, M. A., Myung, I. J., & Zhang, S. (2002). Toward a method of selecting among computational models of cognition. *Psychological Review, 109*(3), 472–491. <https://doi.org/10.1037/0033-295x.109.3.472>
- Preacher, K. J. (2003). *The role of model complexity in the evaluation of structural equation models* [PhD thesis, The Ohio State University].
- Preacher, K. J. (2006). Quantifying parsimony in structural equation modeling. *Multivariate Behavioral Research, 41*(3), 227–259. https://doi.org/10.1207/s15327906mbr4103_1
- Qiu, W., & Joe, H. (2015). *ClusterGeneration: Random cluster generation (with specified degree of separation)*. <https://CRAN.R-project.org/package=clusterGeneration>
- Reise, S. P., Bonifay, W. E., & Haviland, M. G. (2013). Scoring and modeling psychological measures in the presence of multidimensionality. *Journal of Personality Assessment, 95*(2), 129–140. <https://doi.org/10.1080/00223891.2012.725437>
- Reise, S. P., Kim, D. S., Mansolf, M., & Widaman, K. F. (2016). Is the bifactor model a better model or is it just better at modeling implausible responses? Application of iteratively reweighted least squares to the Rosenberg self-esteem scale. *Multivariate Behavioral Research, 51*(6), 818–838. <https://doi.org/10.1080/00273171.2016.1243461>
- Reise, S. P., Morizot, J., & Hays, R. D. (2007). The role of the bifactor model in resolving dimensionality issues in health outcomes measures. *Quality of Life Research, 16*(S1), 19–31. <https://doi.org/10.1007/s11136-007-9183-7>
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*(5), 465–471. [https://doi.org/10.1016/0005-1098\(78\)90005-5](https://doi.org/10.1016/0005-1098(78)90005-5)
- Rosenberg, M. (1965). *Society and the adolescent self-image*. Princeton University Press.
- Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. <https://doi.org/10.18637/jss.v048.i02>
- Savalei, V., & Falk, C. F. (2014). Recovering substantive factor loadings in the presence of acquiescence bias: A comparison of three approaches. *Multivariate Behavioral Research, 49*(5), 407–424. <https://doi.org/10.1080/00273171.2014.931800>
- Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Hall/CRC.
- Steiger, J. H., & Lind, J. C. (1980, May). *Statistically based tests for the number of common factors* [Paper presentation]. Annual Spring Meeting of the Psychometric Society, Iowa City, IA, United States.
- Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika, 38*(1), 1–10. <https://doi.org/10.1007/BF02291170>
- Wilcox, R. R. (2012). *Introduction to robust estimation and hypothesis testing*. Academic Press.

Appendix

Generation of Random Correlation Matrices

The original FORTRAN code for the MCMC algorithm was provided by Kristopher Preacher (2003) and ported to R and modified slightly by the authors. Let \mathbf{r}_k be a $p(p-1)/2$ vector of correlations at iteration k , where p is the number of observed variables. In brief, this approach begins the MCMC chain with the correlation matrix set to an identity matrix, (i.e., $\mathbf{r}_0 = 0$).¹⁴ Candidate draws, are computed by $\mathbf{r}_{k+1} = \mathbf{r}_k + \gamma\mathbf{z}$, where γ is a step size, and $\mathbf{z} = t^{1/p} \frac{\mathbf{x}}{\sqrt{\mathbf{x}\mathbf{x}'}}$, with \mathbf{x} and t randomly drawn at each iteration from an independent normal distribution, $\mathbf{x} \sim \mathcal{N}_p(0, \mathbf{I})$, and uniform distribution, $t \sim \text{unif}(0, 1)$. Candidate draws are only rejected if they result in a nonpositive-definite matrix, or if correlation values exceed allowable values (i.e., within ± 1). In order to reduce these possibilities with large correlation matrices, smaller step sizes are required, which in turn then requires more iterations between draws to reduce autocorrelations. That is, as in typical MCMC applications a number determines *thinning*—or the number of iterations between saving random correlation matrices—and only a subset of iterations (e.g., $n = 10,000$ out of 200,000) are saved. The step size (from .56 to .1) and number of iterations (from 200,000 to 10 million) are preset under the original algorithm for a range of observed variables from three to approximately 16.

Our modifications of the code included an increase to a default of 5 million iterations for the MCMC algorithm. Parallel processing can be used by creating m independent chains for generating correlation matrices. The total number of iterations is held constant where possible by dividing the total number of iterations and draws equally among the m chains. In the case of many processing cores, this could lead to very few iterations per chain. To prevent this, a minimum number of iterations per chain is set at 10,000. In both cases, these options are modifiable by passing a list to an additional argument, `mcmc.args`, and documentation on possible options is provided in the *ockhamSEM* package.

Lewandowski et al. (2009) introduced the ability to generate correlation matrices with the *vine* and *onion* methods, which are faster than the MCMC algorithm. The MCMC algorithm may still generate many correlation matrices that must be discarded due to lack of positive definiteness. The vine method is based on work by Joe (2006), in which partial correlations are generated from a linearly transformed Beta distribution and transformed into product moment correlations. The computations involved

can be illustrated using *C-vines*, which define the dependency structure among the variables using a graphical model. We do not pursue the vine method further due to the need to further study involved tuning parameters that may affect the space for the randomly generated matrices. The method we pursue in the current article is the onion method which allows uniform sampling “over the space of correlation matrices” (Lewandowski et al., 2009, p. 1998). The onion method constructs random correlation matrices recursively, starting with a single dimension and adding additional dimensions in later steps. Lewandowski et al. (2009) provide a detailed description of the method as it relates to elliptical distributions. Either approach is computationally fast, and these authors report generation of many (5,000) large correlation matrices (e.g., 80×80), in only a few seconds (using compiled C code) or a minute or two (using Matlab). In the present application, we use the *clusterGeneration* package (Qiu & Joe, 2015). As neither method requires iterations as does MCMC, parallel processing can be easily conducted by dividing the number of generated correlation matrices among m processing cores.

While we expect that most applications using the onion method will use correlation matrices generated as-is, we have also implemented the following ad-hoc procedure for restricting correlation matrix generation to the space of all positive correlations. Let \mathbf{R} be a generated correlation matrix. To ensure all positive correlations, we compute a new correlation matrix by $\mathbf{R} = .5(\mathbf{R} + \mathbf{1})$, where $\mathbf{1}$ is a matrix of 1’s of conformable dimensions. We offer no proof at the moment that this results in uniform sampling from this data space, however, we note that this approach in many cases results in similar conclusions regarding fit propensity as the MCMC algorithm.

¹⁴Or a matrix with .5 correlations if only positive correlations are desired, as in the original code.

Received July 9, 2020
 Revision received January 10, 2021
 Accepted June 2, 2021 ■