Tools for Thinking about SEM Models

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

April, 2010

Tools for Thinking About SEM Models

Introduction

Structural Equation Modeling (SEM) is a big topic area, and it is impossible to cover more than the basics in a one-term course. So, as we near the end of this semester, and try to carry on in David's absence, I thought about things I wish I had known after *my* first course, but which I didn't (I can't blame the instructor – his name was Steiger).

What Is a Structural Equation Model?

- Modern structural equation modeling software, with its emphasis on the path diagram representation, encourages us to "think diagrammatically" about our models
- However, structural equation models can be thought about in several ways, and at several levels
- To think deeply about the implications of this, we have to go back to basics

What Is a Structural Equation Model?

When we fit a structural equation model with LISREL, how do we proceed? There are minor variations, but many people

1. Start with a diagram representation
2. Translate into LISREL's taxonomy of variable types
3. Send the model commands and data to LISREL
4. LISREL then
   1. Converts the model into algebra for modeling the covariance matrix of the data
   2. Finds the set of model parameters that best fits the observed data, within the confines of the model
   3. Reports back with parameter estimates, standard errors, goodness (or badness) of fit statistics, etc.
5. You try to make sense of it all

What Is a Structural Equation Model?

Once you have the results, the tendency is to interpret them

1. In terms of the diagram's "metaphor"
2. Using currently popular cultural prescriptions

What Is a Structural Equation Model?

1. Sometimes, this way of working and thinking works pretty well. (Especially in the examples you find in textbooks.)
2. Occasionally, it won't do, because certain traps are lurking.

What Is a Structural Equation Model?

The first step in avoiding these traps is to remember what a structural model is doing.

On one side, we have our data. We may choose to model the data in their rawest form, or we may choose to model certain aspects of the data, like their covariances or correlations.

On the other side, we have a set of equations, a set of *free parameters* inside these equations.

The equations are *a prescription for reproducing the data using the parameters*. Once we have our model equations, we vary the parameters until we have done our best to reproduce the data using the equations.

What Is a Structural Equation Model?

Unfortunately, modern textbooks on structural equation modeling fail to point out a fundamental result of this modeling process.

By going back to basics, we can recall what it is, and gain a powerful, fundamental insight into the nature of modeling.

Let's pick a *really simple example* and see what it can teach us.

What Is a Structural Equation Model?

What we want is a really, really simple example of

1. Some data (as little as possible)
2. A "model" in the very basic sense of a set of equations designed to reproduce or fit the data
3. Some *free parameters*, values we are free to manipulate within our fitting efforts.

Ready?

A Really Simple Model for Some Really Simple Data

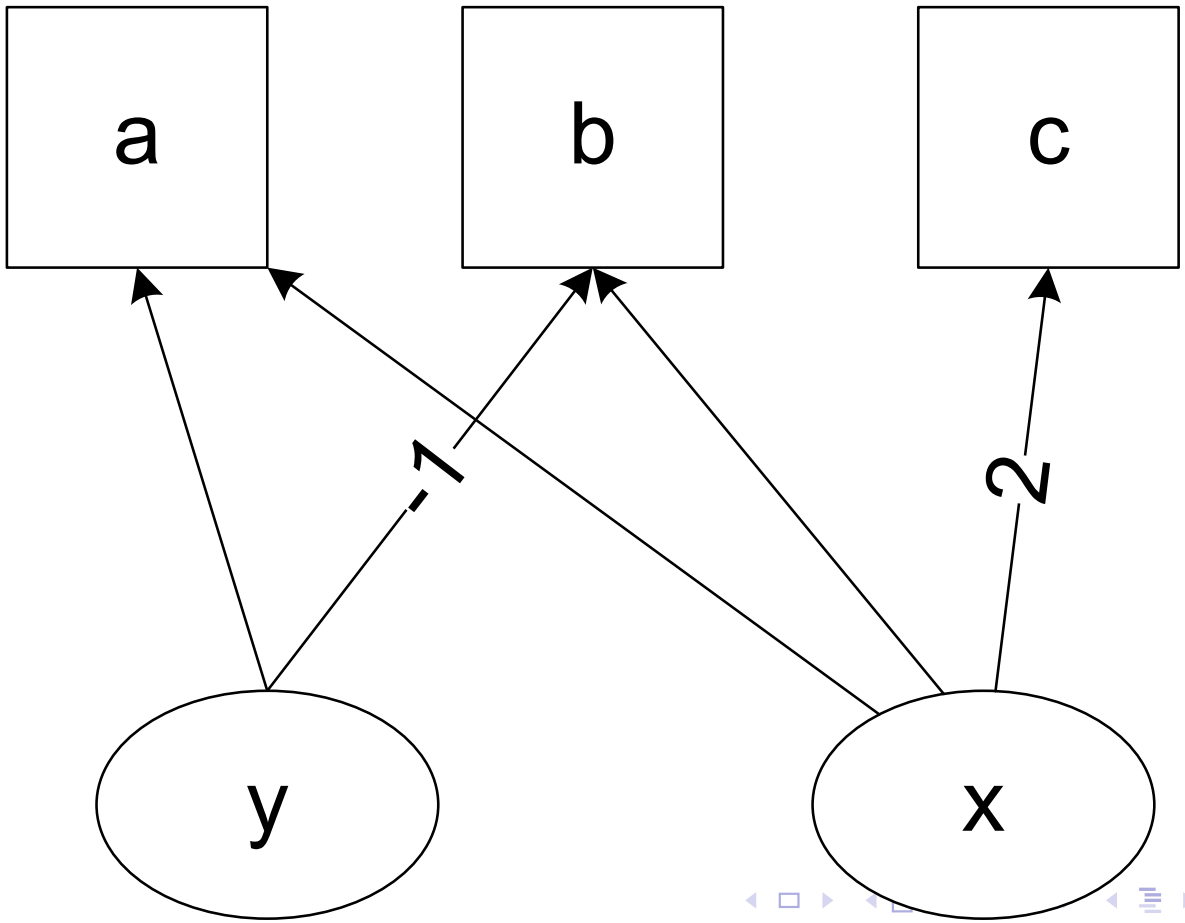
Suppose we have 3 data points, a, b, and c, and we have a "model" that says these data points may be explained in terms of two parameters, x and y. There are 3 model equations, and they are

$$x + y = a, \qquad x - y = b, \qquad 2x = c$$

Some questions:

1. Can you represent this model diagrammatically? (C.P.)
2. Can you fit this model? How can we proceed?

Really Simple Path Model



Analyzing Our Really Simple Model

Before we start analyzing our Really Simple Model, let me tell you that, in all its essentials, this model, and our analysis of it, mimic the essentials of the thought processes of the great Charles Spearman, the actual originator of factor analysis.

We'll begin applying Spearman's analytic technique, but instead of using the factor analysis model, we'll use the Really Simple Model and data instead, so we can get a sense of what it is we are doing without being blinded by the complexities.

Analyzing Our Really Simple Model

Here we go!

Here is how Spearman thought about the situation:

Let's look at our data, our model, and our parameters, and ask the key question, "What can the model, if it is true, tell us about the data?"

How can we do that? Well, that is a thought-provoking question. Spearman's answer might not be obvious to us, although some modern mathematicians consider the insight fundamental. The answer is, write our knowledge down, and then eliminate the free parameters (the unknowns) from the knowledge set, so we are left with a set of statements about the data. Once the statements are in terms of data alone, we may discover *what the model implies about the data.*

Analysis Through Elimination

That's quite a mouthful, but it isn't as complicated as it seems.

What we are going to do is systematically eliminate the free parameters from our equations, and see where that leaves us. Ready? Here is our model again:

$$x + y = a, \qquad x - y = b, \qquad 2x = c$$

$2x = c$ implies that $x = c/2$. So let's begin by eliminating x from the other two equations. What do we get? (C.P.)

Analysis Through Elimination

We are now left with only 2 equations in one parameter, i.e.

$$c/2 - y = b, \qquad c/2 + y = a$$

Adding these two equations together, what do we get?

Analysis Through Elimination

That's right! The y values cancel out and we get

$$c = a + b$$

Only data sets satisfying this equation can fit our model!

Confirming Our Analysis

Let's check out what we have learned. Let's pick a data set that "fits the model" first. How about $c = 4$, $a = 2$, $b = 2$.

Look again at the model. Can you find free parameters that fit these data?

$$x + y = a, \qquad x - y = b, \qquad 2x = c$$

OK, it helps many of us to plug in numbers.

$$x + y = 2, \qquad x - y = 2, \qquad 2x = 4$$

Got it? (C.P.)

Confirming Our Analysis

Now let's try some data that don't follow the rule that data *must follow* to fit the Really Simple Model. How about $c = 4$, $a = 3$, $b = 3$? Plugging in, we get

$$x + y = 3, \qquad x - y = 3, \qquad 2x = 4$$

You can try from here to eternity, you will never find an x and a y that fit the really simple model with those data!

Solving for the Parameters

Notice that we started with 2 parameters and 3 data points. Now that we have eliminated the parameters, we can use the fact that $c = a + b$ to derive equations for the parameters in terms of the data points, given that the data fit the model. In this case, our job is really easy.

The unique solution is

$$x = \frac{a+b}{2} = \frac{c}{2}, \qquad y = \frac{a-b}{2}$$

Solving for the Parameters

What have we learned about the Really Simple Model and its relationship to the data set a, b, c. We have learned that

1. The model fits perfectly if and only if $c = a + b$
2. If the model fits, then $x = (a + b)/2$, and $y = (a - b)/2$

Additional Thoughts

Any additional thoughts about what we have learned about the Really Simple Model?

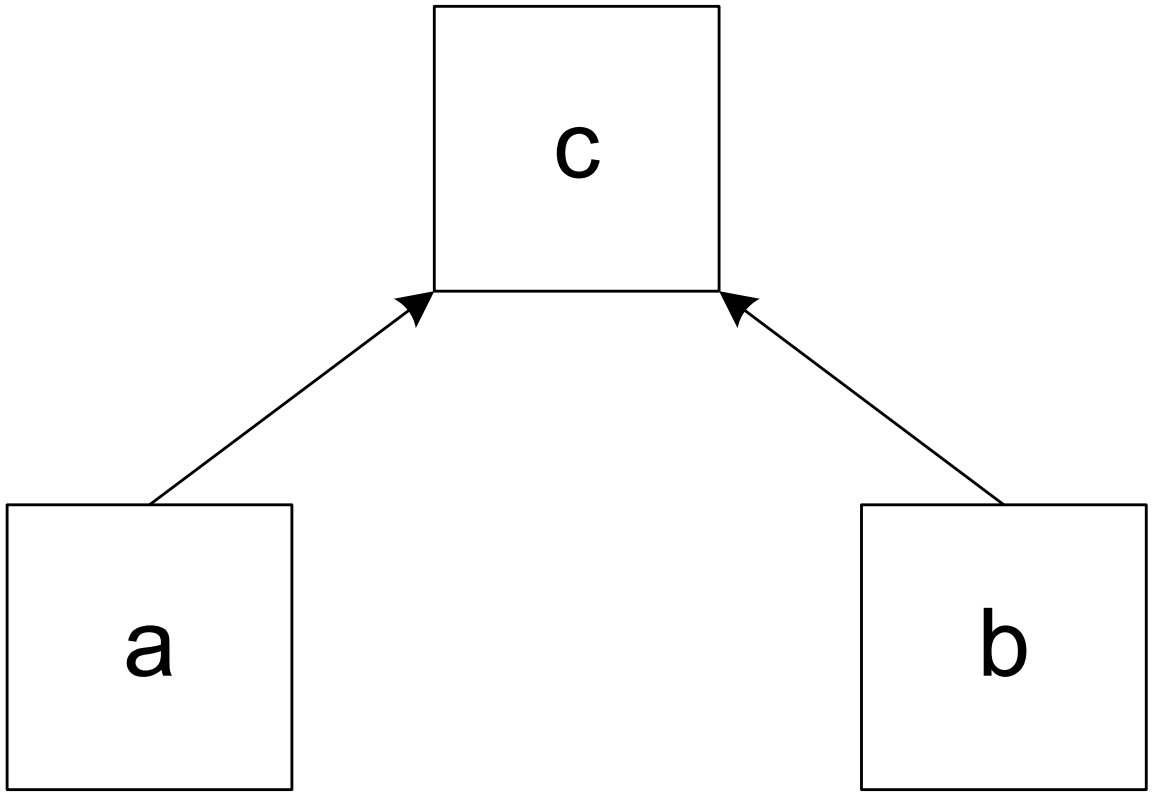
Any additional thoughts about what we have learned about the nature of modeling?

Additional Thoughts

Some more questions. We started with a model for the data, in terms of equations and free parameters.

After elimination, do we not have another model? Can you draw a path diagram for this new model?

Additional Thoughts



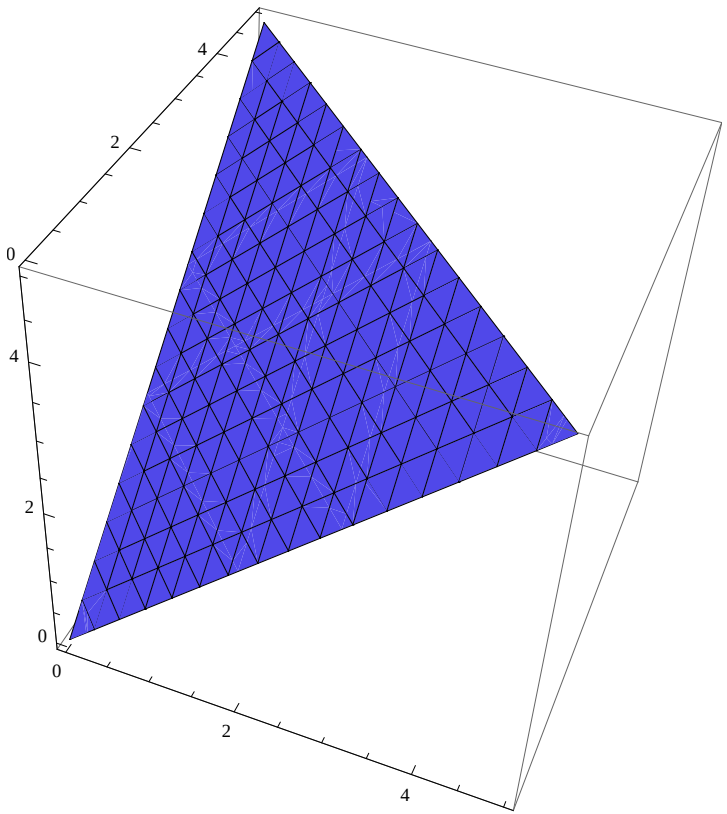
The Nature of a Model

Notice that our new model is shouting out something very important to us. It has defined a *subspace of the data space*. We have computer software that can actually plot this subspace in the case of two or three dimensions.

The Nature of a Model

In[23]:=
```
ContourPlot3D[c == a + b, {a, 0, 5}, {b, 0, 5}, {c, 0, 5}]
```

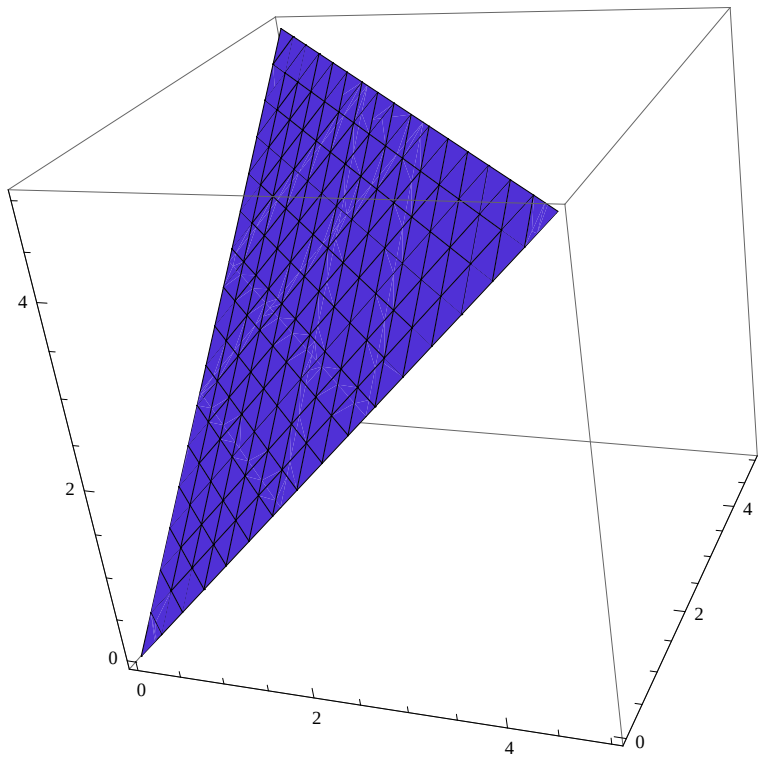
Out[23]=



The Nature of a Model

In[23]:= `ContourPlot3D[c == a + b, {a, 0, 5}, {b, 0, 5}, {c, 0, 5}]`

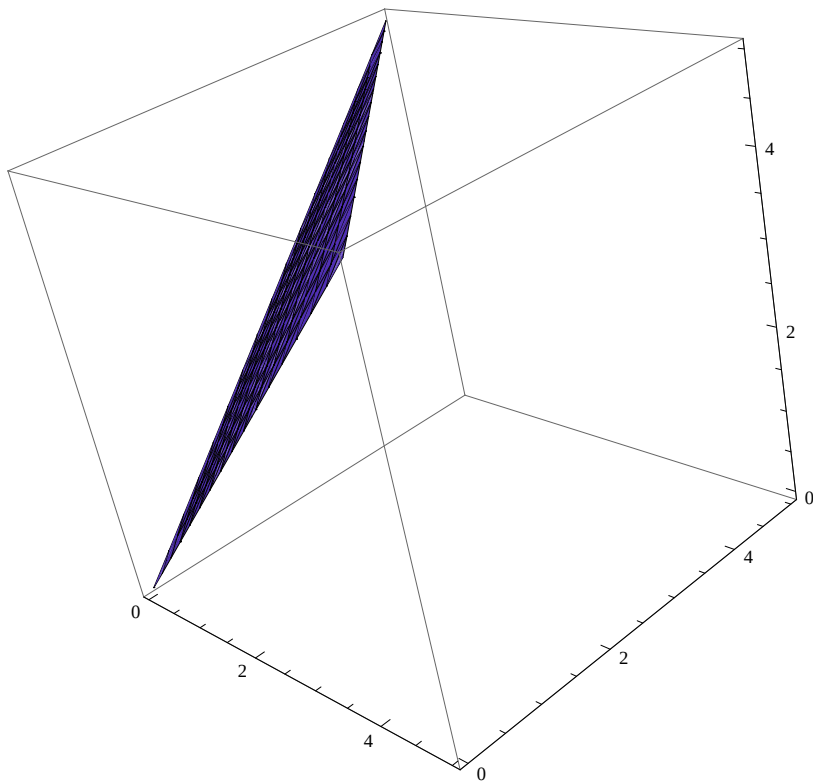
Out[23]=



The Nature of a Model

In[23]:=
```
ContourPlot3D[c == a + b, {a, 0, 5}, {b, 0, 5}, {c, 0, 5}]
```

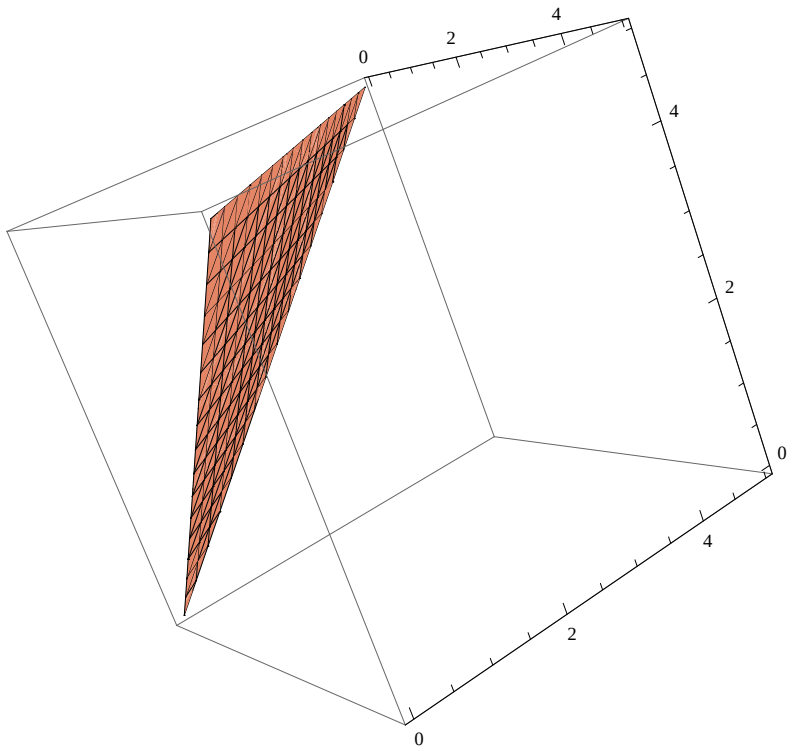
Out[23]=



The Nature of a Model

In[23]:=
```
ContourPlot3D[c == a + b, {a, 0, 5}, {b, 0, 5}, {c, 0, 5}]
```

Out[23]=



The Nature of a Model

Notice how our single model equation has stated that data sets fitting the model are located in a subspace whose dimensionality is reduced by 1. That is, data fitting our model cannot be just any data. They are data that are confined to a two-dimensional subspace of the full three-dimensional data space.

It is *that* notion of a model that statisticians sometimes talk about. They say perplexing things like "a model is a statement that the data are confined to a subspace of reduced dimensionality."

Additional Thoughts

Our second model seems quite different from the first, in that we can impose all kinds of layers of abstraction on the first model, whereas the second seems quite mundane. But notice – in practice we might often approach the first kind of model as an attempt to "capture what is going on" in our data, and our new, Even Simpler Model seems rather impoverished by comparison.

Which is correct? (C.P.) (or perhaps I should have asked, Which is "correct"?)

Empirically Equivalent Models

Our two models, the Really Simple Model and the Even Simpler Model, are *empirically indistinguishable*. That is, any set of data that fits one fits the other.

Extending the Lessons Learned

The above simple example showed how we can analyze a model by elimination. You might see some connections between (a) what we just did and (b) structural equation modeling, but you might (justifiably) remain skeptical about the implications for practice.

So, let's continue by first pointing out that this identical approach was employed (laboriously!) by Charles Spearman. Spearman invented factor analysis, and the notion that we could learn about the "general factor" of intelligence without ever measuring it directly stunned him.

He set about analyzing his factor analysis model in exactly the same way we just analyzed the Really Simple Model.

Let's briefly follow what Spearman did, except let's look at it through a slightly different, slightly more complex model that is more relevant to modern structural equation modeling, i.e., a two factor measurement model.

Extending the Lessons Learned

Before I continue, let me say that we are very fortunate these days to have "computer algebra systems" that can do in a second what it took Spearman days, weeks, or even months to accomplish. Perhaps the best-known of such systems is *Mathematica*. The next slide shows *Mathematica* code for analyzing the Really Simple Model.

Mathematica Analysis

In[3]:=
```
model = {x + y == a, x - y == b, 2 x == c}
```

Out[3]=
$$\{x + y == a,\ x - y == b,\ 2\,x == c\}$$

In[5]:=
```
implication = Eliminate[model, {x, y}]
```

Out[5]=
$$c == a + b$$

In[9]:=
```
WhatWeKnow = Append[model, implication]
```

Out[9]=
$$\{x + y == a,\ x - y == b,\ 2\,x == c,\ c == a + b\}$$

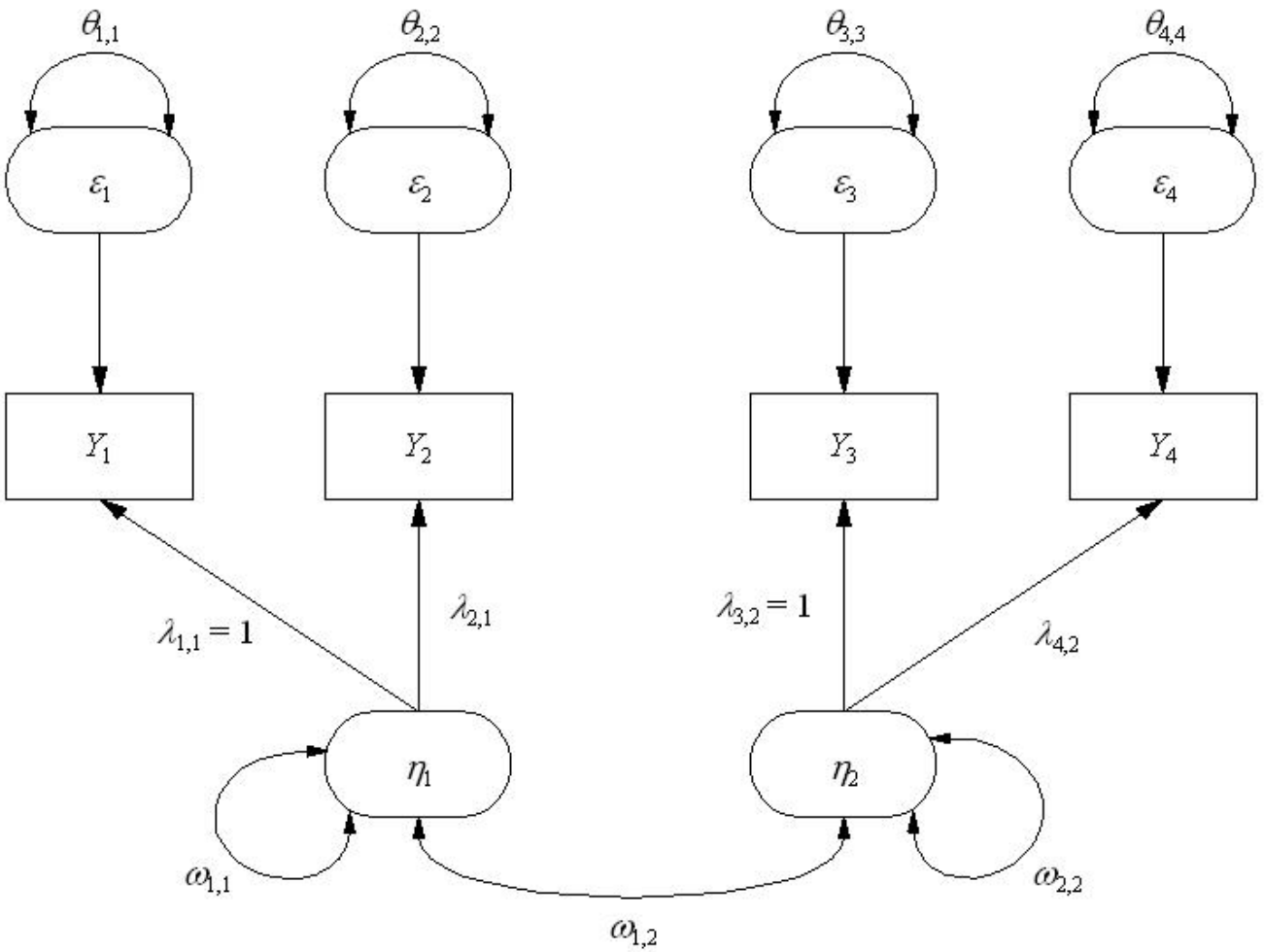
In[10]:=
```
Solve[WhatWeKnow, {x, y}]
```

Out[10]=
$$\left\{\left\{y \to \frac{a - b}{2},\ x \to \frac{a + b}{2}\right\}\right\}$$

Two Factor Confirmatory Factor Model

Here is a confirmatory factor model. Actually, it is a measurement model I pulled out of a more complicated path diagram.

Two Factor Confirmatory Factor Model



Two Factor Confirmatory Factor Model

Notice that, in the diagram, I have $\lambda_{1,1}$ fixed to 1. Why?

Two Factor Confirmatory Factor Model

For our confirmatory model, the LISREL model equation is simplified to

$$\mathbf{\Sigma}_{yy} = \mathbf{\Lambda}_y \mathbf{\Omega} \mathbf{\Lambda}_y' + \mathbf{\Theta}_\epsilon \tag{1}$$

with

$$\mathbf{\Lambda_y} = \begin{bmatrix} 1 & 0 \\ \lambda_{2,1} & 0 \\ 0 & 1 \\ 0 & \lambda_{4,2} \end{bmatrix} \tag{2}$$

$$\mathbf{\Omega} = \begin{bmatrix} \omega_{1,1} & \omega_{2,1} \\ \omega_{2,1} & \omega_{2,2} \end{bmatrix} \tag{3}$$

$$\mathbf{\Theta_\epsilon} = \begin{bmatrix} \theta_{1,1} & 0 & 0 & 0 \\ 0 & \theta_{2,2} & 0 & 0 \\ 0 & 0 & \theta_{3,3} & 0 \\ 0 & 0 & 0 & \theta_{4,4} \end{bmatrix} \tag{4}$$

Two Factor Confirmatory Factor Model

Computing Equation 1 we find that the model equations (with redundant elements above the diagonal not shown) are

$$
\begin{aligned}
\sigma_{1,1} &= \theta_{1,1} + \omega_{1,1} \\
\sigma_{2,1} &= \lambda_{2,1}\omega_{1,1} \\
\sigma_{2,2} &= \theta_{2,2} + \lambda_{2,1}^2\omega_{1,1} \\
\sigma_{3,1} &= \omega_{2,1} \\
\sigma_{3,2} &= \lambda_{2,1}\omega_{2,1} \\
\sigma_{3,3} &= \theta_{3,3} + \omega_{2,2} \\
\sigma_{4,1} &= \lambda_{4,2}\omega_{2,1} \\
\sigma_{4,2} &= \lambda_{2,1}\lambda_{4,2}\omega_{2,1} \\
\sigma_{4,3} &= \lambda_{4,2}\omega_{2,2} \\
\sigma_{4,4} &= \theta_{4,4} + \lambda_{4,2}^2\omega_{2,2}
\end{aligned}
$$

Two Factor Confirmatory Factor Model

Mathematica can do the elimination for the above set of equations pretty easily. You end up with the following result:

$$\sigma_{3,1}\sigma_{4,2} = \sigma_{3,2}\sigma_{4,1} \tag{5}$$

In general, I refer to such an equation, derived this way, as a "$\boldsymbol{\Sigma}$-constraint." It is a constraint that the data must follow in order to satisfy the model. Clearly, there are infinitely many covariance matrices that do not satisfy this constraint. But any other model that has the same $\boldsymbol{\Sigma}$-constraint is empirically indistinguishable from our confirmatory factor model.

Two Factor Confirmatory Factor Model

Examination of a model's $\mathbf{\Sigma}$-constraints can reveal interesting aspects of the model. For example we see that the variances of the 4 observed variables are not present in the $\mathbf{\Sigma}$-constraint for this model, and all subscript values occur equally often on both sides of the constraint equation. This implies that any change of scale of the 4 observed variables cannot affect whether the model fits $\mathbf{\Sigma}$, and so in this case an equivalent constraint is $\rho_{3,1}\rho_{4,2} - \rho_{3,2}\rho_{4,1} = 0$.

Two Factor Confirmatory Factor Model

This equation has an interesting form. It is the difference of two products of correlations, each of which involve the same four variables but in different permutations. Spearman(1904) showed that all $\mathbf{\Sigma}$-constraints for an unrestricted single factor model could be expressed in this form. He called such a constraint a *tetrad equation*, and the left side of the equation a *tetrad difference*.

Two Factor Confirmatory Factor Model

Adding Equation 5 to the original system, one can show that, if the data fit the model, and certain degenerate conditions (e.g., $\sigma_{4,1} = 0$) do not hold, then closed form solutions for all model parameters are available. For example,

$$\lambda_{4,2} = \frac{\sigma_{4,2}}{\sigma_{3,2}}$$

$$\lambda_{2,1} = \frac{\sigma_{4,2}}{\sigma_{4,1}}$$

Why Should We Care?

Why should we care about this?

1. We might discover that models we thought are the same are actually different
2. We might discover that models we thought are different are actually empirically indistinguishable
3. We might discover that the ability of some models to fit data is invariant under changes of scale
4. We can prove that a model is identified (when it is identified!)
5. We might discover helpful clues as to why models are not identified
6. We might discover situations in which LISREL gives wrong answers

Let's continue. It might be worth it.

Unit Loading Identification Constraints

Earlier I asked why some of the λ values in our confirmatory factor analysis model are constrained to be 1.

The answer is that they are required to identify the parameters of the model. Without some constraint on the model parameters, it turns out that infinitely many sets of parameter values will fit equally well.

Unit Loading Identification Constraints

A simple and easy constraint that can be applied in many situations without causing too many problems is what Steiger (2002) referred to as a Unit Loading Identification (ULI) constraint.

Many textbooks will say something like "each endogenous factor needs one loading fixed to unity to establish identification" or "each endogenous factor needs one loading fixed to unity to fix the scale of the factor."

The clear implication of the advice is that ULI constraints can, and should, be used almost automatically.

What, precisely, do they mean by that? We can understand better by employing a useful tool for understanding some of the vagaries of path diagrams. I call this tool the "pipeline metaphor."

The Pipeline Metaphor

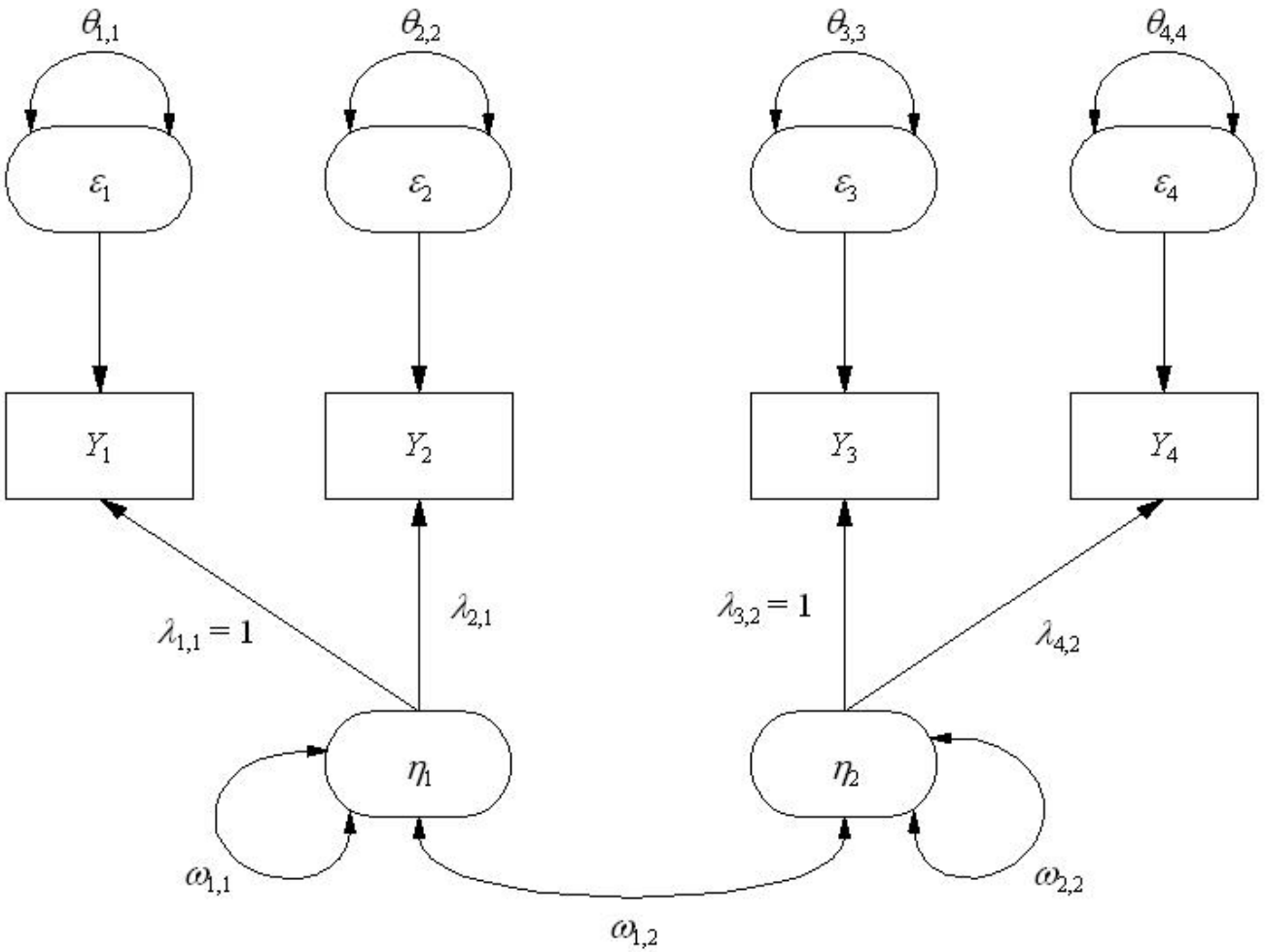
Let's look back, again, at our confirmatory factor analysis model. Imagine that standing at any point in the diagram and monitor the numbers being "piped" through the paths.

Doubling the standard deviation (or quadrupling the variance) of a variable simply doubles the magnitude of every number coming out of it.

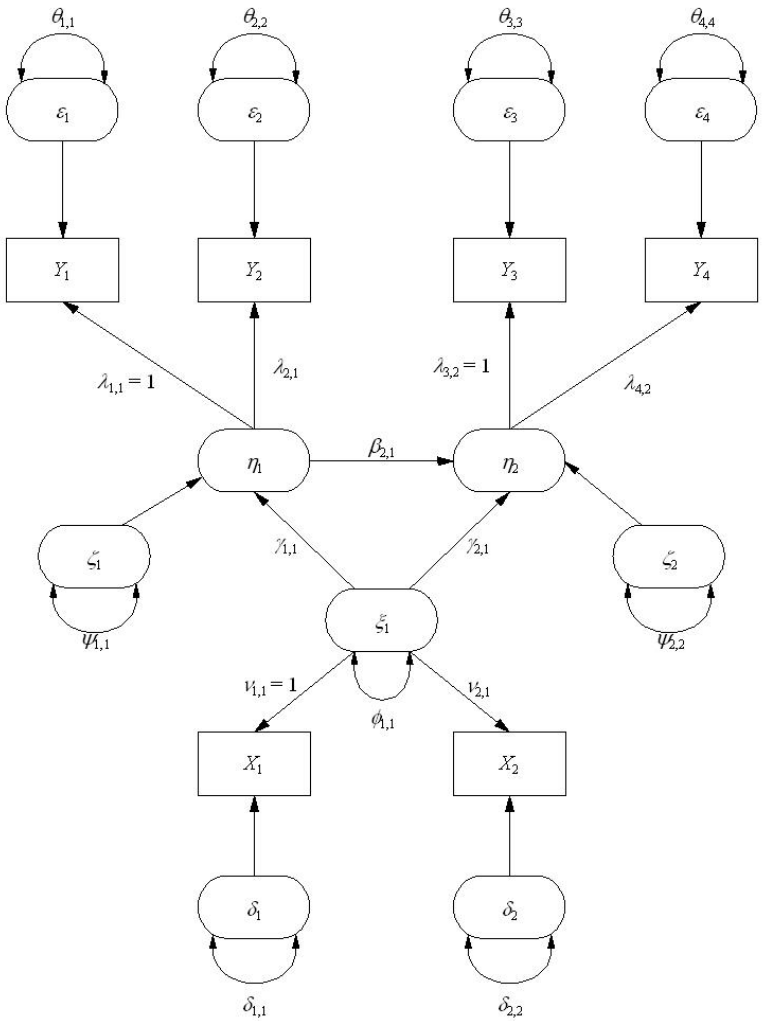
Path coefficients in such diagrams act like multipliers, so any number is multiplied by a path coefficient it passes through. Because every number passing through a path is multiplied by its path coefficient, the standard deviation of the number is multiplied by the absolute value of the coefficient, and the variance by the square of the coefficient.

This is true not only in the confirmatory model when it is presented in isolation, but also when it is embedded in a larger structural equation model as a "measurement model."

The Pipeline Metaphor



The Pipeline Metaphor



The Pipeline Metaphor

With these simple notions in tow, we note first that latent variable η_1 is never observed, and so its variability may only be inferred from two sources:

1. the variances and covariances of the variables with paths leading to η_1,
2. the values of the path coefficients leading to η_1.

The Pipeline Metaphor

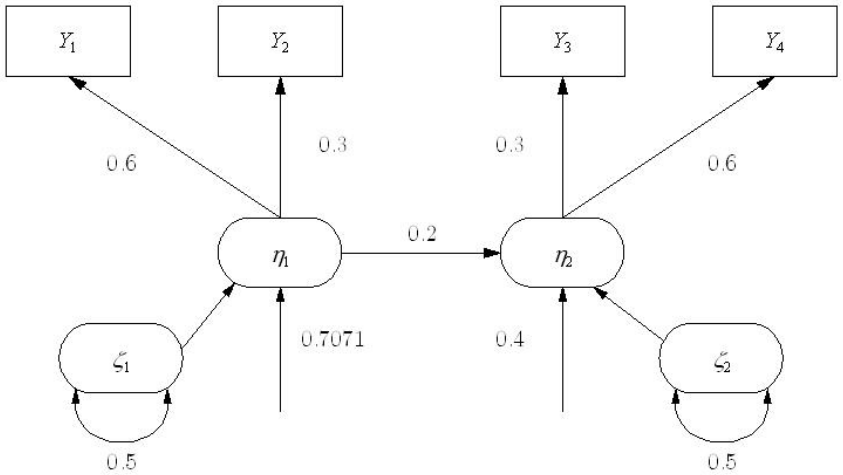
The variances of η_1 and η_2 are not uniquely defined, and are free to vary unless some constraints are imposed on the free parameters in our confirmatory model.

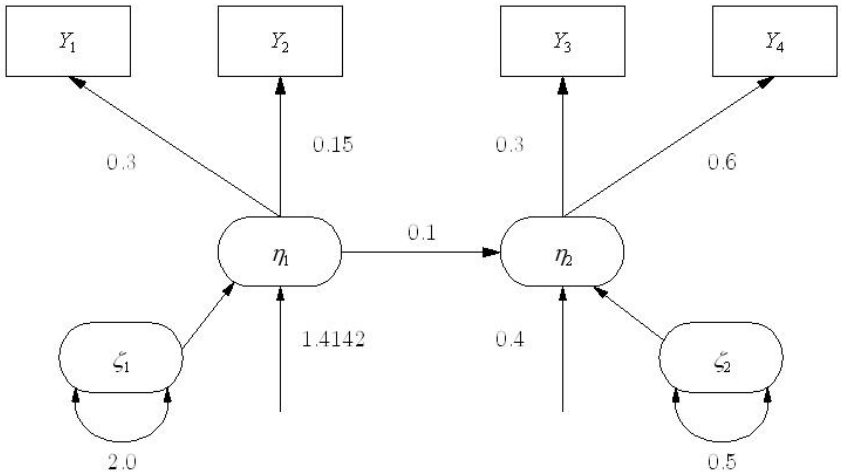
To see why, suppose that the ULI constraints *were removed* from $\lambda_{1,1}$ and $\lambda_{2,1}$, and that, by some combination of circumstances, the paths leading to η_1 and η_2 had values that caused η_1 to have a variance of 1.

Suppose further that under these circumstances, the values .6, .3, .6, and .3 for parameters $\lambda_{1,1}$,$\lambda_{2,1}$,$\lambda_{3,2}$, and $\lambda_{4,2}$ lead to an optimal fit of the model to the data.

The upper diagram in the figure on the next slide shows this situation.

The Pipeline Metaphor





The Pipeline Metaphor

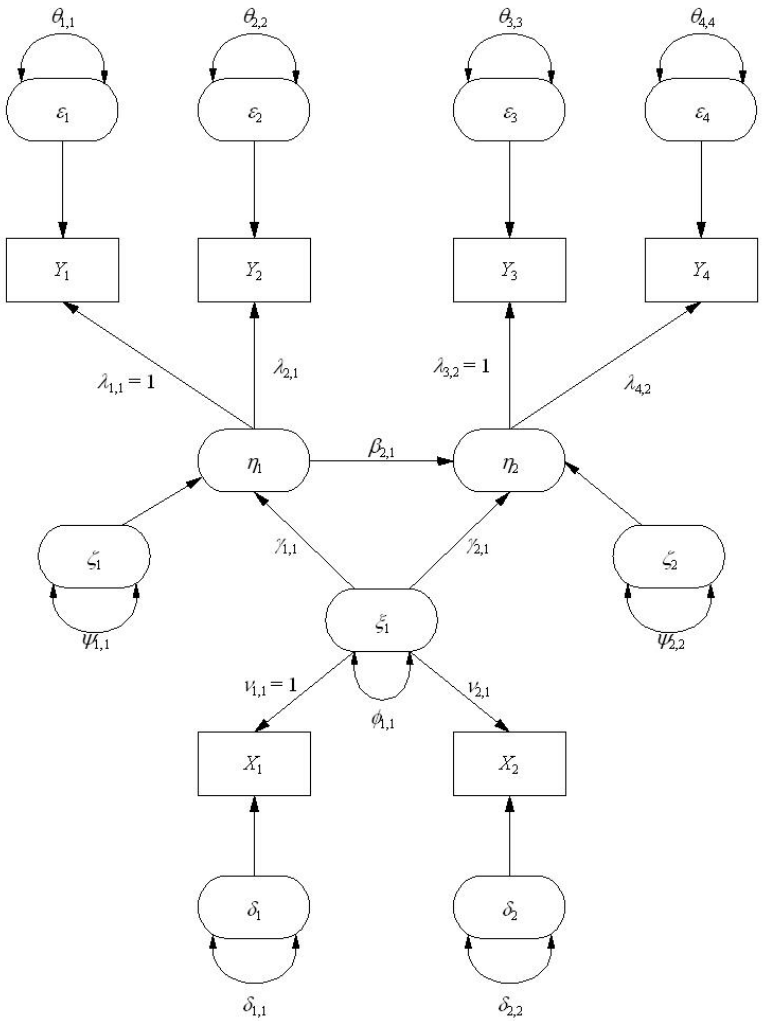
Next, imagine we wished the variance of η_1 to be some value other than 1, say, 4.

Quadrupling a variable's variance can be accomplished by doubling its standard deviation, or doubling every value of the variable. To achieve this, while maintaining the identical numbers arriving at Y_1, Y_2 and η_2 from η_1, we need only double all values on paths leading to η_1, while simultaneously halving all values ($\lambda_{1,1}$, $\lambda_{2,1}$, and $\beta_{2,1}$) on paths leading away from η_1.

The Pipeline Metaphor

Every number emerging from η_1 is doubled, but is "passed through" coefficients that are now exactly half what they were. So the numbers emerging at Y_1, Y_2, and η_2 are the same as they were. Because $\psi_{1,1}$ and $\gamma_{1,1}$ are free parameters that are attached to unidirectional paths, we can alter them (to halve the values of all numbers arriving at η_1) without affecting anything in the paths leading into our measurement model when it is embedded in a larger model, as shown on the next slide.

The Pipeline Metaphor



The Pipeline Metaphor

The pipeline metaphor has helped us to see why ULI constraints are necessary to identify the coefficients in a path model. Without the application of such a constraint, we can see that the variances of our latent variables could be any positive value.

Now, the choice of a constraint to apply is, in a sense, arbitrary. Why do you think the "powers that be" chose to set one loading to 1, rather than to, say, 2? Is there some *other*, more natural constraint that people might have chosen?

The Pipeline Metaphor

How about constraining the endogenous latent variables to have variances of 1?

The Pipeline Metaphor

When we change the particular constraint we employ to achieve identification, what things (if any) remain constant?

Characteristics of ULI Constraints

By using the pipeline metaphor, we've deduced some things.

1. When a ULI constraint is applied to a parameter, the primary goal is simply to establish identification, and the precise value that the parameter is fixed to will not affect the fit of the model. Specifically, one could use the value 2.0 instead of 1.0, and the test statistic for the model would remain the same, because the fit of the model is *invariant under change of scale of its latent variables.*
2. The particular manifest variable chosen for the ULI constraint for any latent variable should not affect model fit. In the present example, fit will be the same if we constrain either $\lambda_{1,1}$ or $\lambda_{2,1}$ (but not both).
3. Path coefficients leading from a latent variable have the same relative magnitude regardless of the fixed value used in a ULI. Their absolute magnitude will go up or down depending on the fixed value used in the ULI. So, for example, if one changes the 1.0 to a fixed value of 2.0, all path coefficients leading from the latent variable will double.
4. Any multiplicative change in the ULI constraint applied to a path coefficient will be mirrored by a corresponding division of the standard deviation of the latent variable the path leads from, and a corresponding division of path coefficients leading to the latent variable.

Characteristics of ULI Constraints

The above properties reflect the way ULI constraints are supposed to work in practice. *The constraints are intended to be essentially arbitrary values imposed solely to achieve identification, and are not intended to have any substantive impact on model fit or model interpretation.*

There seems to be some confusion in the literature about the latter point.

Numerous sources make a statement to the effect that a ULI constraint for the loading of a particular manifest variable fixes the scale of the latent variable to be "the same as the manifest variable."

Characteristics of ULI Constraints

This misconception has led to the use of the term *reference variable* to refer to the manifest variable with the ULI attached. This view is wrong — if a value of unity is employed, the variance of the latent variable is fixed to the variance of the *common part* of the manifest variable which has the ULI constraint. Moreover, as we have already seen, all other loadings emanating from the latent variable move up or down in concert with the value selected for the ULI constraint, and the variance of the common part is itself determined by the choice of variables in the measurement model. The key issue here is that residual variance includes error variance and unique variance, so fixing the metric of the latent variable to an observed variable's common variance has dubious value.

Invariance of Hypotheses

With these goals in mind, it seems reasonable to ask *which hypotheses are invariant* under choice of ULI constraints (or equivalently, under a choice of the scale of the latent variable), and *which are not*. Unless a particular choice of constraint (or latent variable variance) has a specific substantive meaning, a hypothesis that is not invariant under a choice of constraints will be difficult if not impossible to interpret.

For example, is the hypothesis that $\lambda_{1,1}$ equals $\lambda_{2,1}$ in the the general model invariant under a change of scale of the latent variables? (C.P.)

Invariance of Hypotheses

From the preceding analysis, it would seem that the answer is yes, since any change in the ULI constraint would be reflected proportionally in coefficients $\lambda_{1,1}$ and $\lambda_{2,1}$.

The choice of the particular value employed in the identifying constraint has no effect on this hypothesis.

Another way of putting it is that the particular value of the variance of η_1 has no effect on the truth or falsity of the hypothesis. Similarly, the hypothesis that $\lambda_{3,2}$ and $\lambda_{4,2}$ are equal is invariant under choice of the fixed value employed in an identifying constraint on the variance of η_2.

Invariance of Hypotheses

We have established there are hypotheses about the model coefficients that are invariant under the choice of value we fix latent variable variances to, so long as the constraints are only to achieve identification. It seems reasonable to suggest that, if a hypothesis is invariant under the choice of the fixed value used in the identifying constraint, then the hypothesis might be considered meaningful when the value of 1.0 typically used in the ULI is used.

Invariance of Hypotheses

Consider the hypothesis

$$H_0 : \lambda_{2,1} = \lambda_{4,2}$$

Is the truth status of this hypothesis invariant under a choice of identification constraint?

Invariance of Hypotheses

The answer is "no." This hypothesis is not invariant under the choice of fixed value employed in the identifying constraint on $\lambda_{1,1}$. Doubling the fixed value of $\lambda_{1,1}$ doubles the value of $\lambda_{2,1}$ while leaving $\lambda_{4,2}$ unchanged. In this case, the hypothesis is not *invariant under change of scale of the latent variables.*

Invariance of Hypotheses

So we see that some hypotheses might make sense when ULI constraints (or other arbitrary identification constraints) are employed, while others might not make sense.

The impression given by many textbooks is that ULI constraints are automatic and, in a sense, arbitrary. They might be within the simple context of a few textbook examples, but in the larger framework of structural equation modeling in full generality, they might not be.

Some Questions to Ask

In analyzing whether a ULI constraint (or set of constraints) is truly arbitrary, we should ask questions like these:

1. Does the goodness-of-fit statistic remain invariant under the choice of fixed value employed in the identifying constraint? That is, if we change the 1.0 to some other number, does the value remain constant?
2. Does the goodness-of-fit statistic remain invariant under the choice of which manifest variable is the reference variable?
3. Do the relative sizes of path coefficients leading to the latent variable remain invariant under the choice of the fixed value employed in the identifying constraint?
4. Do the relative sizes of path coefficients leading from the latent variable remain invariant under the choice of the reference variable?