Discovering Structural Equation Modeling Using Stata

Revised Edition

ALAN C. ACOCK
Oregon State University

A Stata Press Publication
StataCorp LP
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2$_\varepsilon$
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-59718-139-0
ISBN-13: 978-1-59718-139-6

Library of Congress Control Number: 2013946346

(Pages omitted)

Contents

(Pages omitted)

Preface

What is assumed?

There are two ways of learning about structural equation modeling (SEM). The one I have chosen for this book is best described by an old advertising tag for a sport shoe company: “Just do it”. My approach could be called kinetic learning because it is based on the tactile experience of learning about SEM by using Stata to estimate and interpret models. This means you should have Stata open while you read this book; otherwise, this book might help you go to sleep if you try to read it without simultaneously working through it on your computer. By contrast, if you do work through the examples in the book by running the commands as you are reading, I hope you develop the same excitement that I have for SEM.

The alternative approach to learning SEM is to read books that are much more theoretical and may not even illustrate the mechanics of estimating models. These kinds of books are important, and reading them will enrich your understanding of SEM. This book is not meant to replace those books, but simply to get you started. My intent is for you to work your way through this book sequentially, but I recognize that some readers will want to skip around. I am hopeful that after you have been through this book once, you will want to return to specific chapters to reference techniques covered there. To facilitate this, each chapter includes some repetition of the most salient concepts covered in prior chapters. There is also a detailed index at the end of the book.

What background is assumed? A person who has never used Stata will need some help getting started. A big part of Stata’s brilliance is its simplicity, so a few minutes of help will get you up and ready for what you need to know about Stata. If you are new to Stata, have a friend who is familiar with the program show you the basics. If you have read my book *A Gentle Introduction to Stata* (2012a), you are ahead of the game. If you have any experience using Stata, then you are in great shape for this book. If you are a longtime Stata user, you will find that parts of this book explain things you already know.

To get the most out of this book, you need to have some background in statistics with experience in multiple regression. If you know path analysis, you will find the SEM approach to path analysis a big improvement over traditional approaches; however, the material on path analysis has been written for someone who has had very little exposure to path analysis. Even though the first chapter begins by covering how factor analysis has been used traditionally, a background in factor analysis is less important than having had some exposure to multiple regression. The first chapter shows how confirmatory factor analysis adds capabilities to move beyond the traditional approach—you may never want to rely on alpha and principal component factor analysis again for developing a scale. I have covered enough about the traditional applications of factor analysis that you will be okay if you have had little or no prior exposure to factor analysis.

What will I learn?

We will explore many of the most widely used applications of SEM. We will begin with how to estimate a confirmatory factor analysis model—this is the measurement model part of SEM. This chapter includes parceling as a way to handle a large number of items for one or more of your factors. Next, we will cover path models using SEM—this is the structural model part of SEM. This chapter also introduces nonrecursive path models. We then put these two components together to introduce the full structural equation model. This chapter on the full model includes a number of specialized actions, such as equality constraints. With this foundation, we move on to a chapter on growth curves and conclude with a chapter on multiple-group analysis.

The book has two appendixes. Appendix A shows you how to use Stata's graphical user interface (GUI) to draw and estimate models with Stata's SEM Builder. It would be very useful to begin here so that you are familiar with the SEM Builder interface. If you have no background in SEM, you will not understand how to interpret the results you generate in appendix A, but this is not the point. Appendix A is just there to acquaint you with the SEM Builder that Stata introduced in version 12 and enhanced in version 13. How the interface works is the focus of appendix A. In the text, I use this GUI fairly often, but the focus is on understanding why we are estimating models the way we do and how we interpret and present the results. All the figures presented in this book were created using the SEM Builder, which produces publication-quality figures—far better than what you can draw with most other software packages that produce "near" publication-quality figures.

Appendix B shows you how to work with summary data (means, standard deviations, correlations) that are often reported in published works. You will be able to fit most models with these summary statistics even if you do not have the real data. This feature is great when you read an article and would like to explore how alternative models might be more appropriate. Many articles include a correlation matrix along with standard deviations and means. If these are not included, it is easier to request them from the author than it is to request the author's actual data.

In addition to the two main appendixes, the first two chapters each have their own appendix that briefly describes using the SEM Builder for the models estimated in that chapter.

How do I use this book?

The chapters are intended to be read in the order they appear, and you should follow along with your reading on your computer. (It does not matter whether you use a Windows, Mac, or Unix operating system because the Stata commands and results will be identical.) If you have no background in SEM or factor analysis, take your time reading chapter 1. If you are comfortable with SEM and factor analysis, you should still go over chapter 1 enough to get a feel for how Stata works with the measurement model.

Though the chapters are fairly long, they are broken up into more manageable sections. If you are like me, once you know the commands I cover, you will have enough on your plate that you will forget the specifics before you need to fit a particular type of model. The sections in each chapter build on each other but are sufficiently independent that you should find them useful as a reference. Someday you will want to estimate a nonrecursive path model or a mediation model; you can easily find the section covering the appropriate model and come back to it. At the same time, this book does not attempt to compete with Stata's own *Structural Equation Modeling Reference Manual*, or [SEM]; I only cover a widely used subset of the options and postestimation commands available in Stata's SEM package.

What resources are available?

To facilitate the kinetic part of learning, you can download all the data used in this book as well as the Stata programs, called do-files, that fit every model. In the Command window, type the following:

```
. net from http://www.stata-press.com/data/dsemusr/
. net describe dsemusr
. net get dsemusr
```

When you run these three commands, you do not type the initial period and space, called the dot prompt. A convention in all Stata documentation and output in the Results window is to include the dot prompt as a prefix to each command, but you need only type the command itself.

There are several varieties of Stata software, and all of these are able to run the models described in this text. I focus on the Windows and Mac operating systems, and I show when there are slight differences in how they work in the GUI. The Unix GUI is very similar to the Windows GUI. The same Stata do-files run on all operating systems, though the systems differ slightly in how the file structure is organized.

One variety of Stata is called Small Stata. This is full featured and is small only in the sense of being limited in the number of observations (1,200) and variables (99) it can handle. Because a few of the datasets I use have more than 1,200 observations, I have made up smaller datasets that will work using Small Stata. You can obtain these datasets by entering the following in the Command window:

```
. net from http://www.stata-press.com/data/dsemusr/
. net describe dsemusr_small
. net get dsemusr_small
```

Using the Small Stata data, you will get somewhat different results for some models in the book simply because you will be using a smaller dataset. In addition, there are three models in chapter 4 that will not run using Small Stata.

At the end of each chapter, you will find some exercises that illustrate the material covered in the chapter. It is important to fit all the models in the text while you read the book because this reinforces what you are learning, as does typing in the commands yourself. The exercises extend this learning process by having you develop your own set of commands and models using the GUI system.

There is much more to SEM than could possibly be covered in a book this size. This book is intended to complement the material in the Stata manuals (over 11,000 pages of helpful information), which are available as PDF files when you install Stata. One way to access the [SEM] manual is to type `help sem` in the Command window of Stata. This opens a help file. At the top left of the help file, the title ([SEM] `sem and gsem`) is highlighted in blue. Clicking on this blue link will open up the PDF file of the [SEM] manual.

This book contains a fairly detailed index. Although I have explanatory section headings and these are a good place to start searching for how to do something, the index is naturally much more detailed. You may need to find how to place equality constraints on a multiple-group analysis or on a pair of reciprocal paths. These are covered in very different sections of the book, and the index tells you where to find them. The index was written to be useful after you have read this book and are using it as a reference to guide you while fitting your own models on your own data.

Conventions

Typewriter font. I use a `typewriter font` when something would be meaningful to Stata as input. This would be the case for something you type in the Command window or in a do-file. If a command is separated from the main text, as in

```
. sem (compliance <- educ income gender)
```

a dot prompt will precede the command. I also use a typewriter font for all Stata results, variable names, folders, and filenames.

Bold font. I use a **bold font** for menu items and for buttons you click within a menu. The bold font helps distinguish the button from the text; for example, you might be instructed to click the **Adjust Canvas Size** button.

Slant font. I use a *slant font* when referring to keys on our keyboard, such as the *Enter* key.

Italic font. I use an *italic font* when referring to text in a menu that you need to replace with something else, such as an actual variable name.

Capitalization. Stata is case sensitive. The command `sem (compliance <- educ income gender)` will produce a maximum likelihood multiple regression. If you replace `sem` with `Sem`, Stata will report that it has no command called `Sem`. I will use lowercase for all commands and all observed variables. When I refer to latent variables, I will capitalize the first letter of the latent variable. A simple confirmatory factor analysis would be `sem (Alienation -> anomia isolate depress report)`. Only the latent variable, `Alienation`, is capitalized. The arrow indicates that observed variables measure how a person responds on an anomia scale labeled `anomia`, an isolation scale labeled `isolate`, a depression scale labeled `depress`, and a reported score from an observer labeled `report`. All four of these observed variables depend on their level of `Alienation`, the latent variable.

(Pages omitted)

1 Introduction to confirmatory factor analysis

1.1 Introduction

When we are measuring a concept, it is desirable for that concept to be unidimensional. For example, if we are measuring a person's perception of his or her health, the concept is vague in that there are multiple dimensions of health—physical health, mental health, and so on. Let us suppose Chloe is in excellent physical health (a perfect 10), but she is very low on mental health (a score of 0.0). Madison is in excellent mental health (a perfect 10), but has serious problems with her physical health (a score of 0.0). Jaylen is average on both dimensions (a score of 5.0 on both). Do we want to give all three the same score because Chloe, Madison, and Jaylen each average 5.0? Should one dimension be more important than another?

When you have two dimensions (x and y) and try to represent them on a graph, you need two values: one showing a score on the x dimension and one showing a score on the y dimension. When there is more than one dimension, a single score becomes difficult to interpret, and it is often misleading to represent the location of a person on the concept with a single number. Thus there are advantages to narrowly defining our concepts so our measures can tap a single dimension. If we are interested in multiple dimensions, such as distinguishing between physical and mental health, then we need multiple concepts and multiple empirical sets of measures.

On the other hand, we can carry this argument too far. Each item we might pick to measure physical health will represent a slightly different aspect of physical health. We should aim to represent as broad a meaning of physical health as we can without adding distinctly different dimensions. The ideal way to do this is to allow each item to have its own unique variance and develop a scale score that represents the shared meaning of the set of items on a single dimension. This way, our measurement model represents concepts that are neither too broad to have a clear meaning nor too narrow to be of general interest.

This is where we will go with confirmatory factor analysis (CFA). We will first cover the "do not even think about it" approach, followed by the exploratory search for a single dimension using the traditional principal component factor analysis approach. We will then extend this to CFA measurement models where we have first one and then two or more concepts.

1.2 The “do not even think about it” approach

Many studies have brief sections reporting how they measured the variables. The authors simply assume the dimensionality of what they are measuring; all they report is the alpha (α) measure of reliability. (Do not confuse this alpha coefficient with the use of alpha for the conventional level of statistical significance.) If alpha is greater than 0.80 (sometimes a lower value is considered adequate), then these authors say they have a good measure. The alpha coefficient is a measure of internal consistency. It depends on just two parameters, namely, the average correlation/covariance of the items with one another and the number of items. With 20 or more items, the alpha could be 0.80 even if the items are only weakly correlated with one another and even if the items represent several dimensions.

A good alpha value does not ensure that a single dimension is being tapped. Consider the following correlation matrix:

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$
$x1$	1.0					
$x2$	0.6	1.0				
$x3$	0.6	0.6	1.0			
$x4$	0.3	0.3	0.3	1.0		
$x5$	0.3	0.3	0.3	0.6	1.0	
$x6$	0.3	0.3	0.3	0.6	0.6	1.0

We see in this matrix two subsets of items: $x1$–$x3$ and $x4$–$x6$. Items $x1$–$x3$ are all highly correlated with each other (r’s = 0.6) but much less correlated with items $x4$–$x6$ (r’s = 0.3). Similarly, items $x4$–$x6$ are highly correlated with one another (r’s = 0.6) but much less correlated with items $x1$–$x3$ (r’s = 0.3). This indicates that there are two related dimensions, namely, whatever is being measured by $x1$–$x3$ for one dimension and whatever is being measured by $x4$–$x6$ for the other. The alpha for these six items is $\alpha = 0.81$, which is considered good. For example, Kline (2000) indicates that an alpha of 0.70 and above is acceptable. However, the point here is that when we rely on alpha to justify computing a total or mean score for a set of items, we may be forcing together two or more dimensions, that is, trying to represent two (or more) concepts with one number. At the very least, we should routinely combine reports of reliability with some sort of factor analysis to evaluate how many dimensions we are measuring.

Alpha can be high even with items that are only minimally related to one another. The formula for a standardized alpha is

$$\alpha = \frac{k\overline{r}}{1 + (k-1)\overline{r}}$$

where k is the number of items in the scale and $\overline{r}$ is the mean correlation among the items. We would not think of an $r = 0.17$, for example, as more than a minimal relationship. After all, if $r = 0.17$ then $r^2 = 0.03$, meaning that 97% of the variance in the two variables is not linearly related. However, if you had a 40-item scale with an average correlation of just 0.17, your alpha would be 0.80. The measure would be reliable in the sense of internal consistency, but the high alpha does not mean we are measuring a single dimension. Simply adding up a series of items (or taking their mean) and reporting an alpha is insufficient to qualify a measure as a good measure.

1.3 The principal component factor analysis approach

Principal component factor analysis (PCFA) is the most traditional approach to factor analysis. Several authors have demonstrated that this is far from the best type of factor analysis (Fabrigar et al. 1999; Costello and Osborne 2005), and some prefer to go so far as to say it is not really factor analysis at all. Stata offers alternative exploratory factor analysis methods, including maximum likelihood factor analysis, that have significant advantages; we are using Stata's PCFA only because of its widespread use. The structural equation modeling approach has advantages over all the traditional approaches to factor analysis and will be the focus of this book.

A major concern with PCFA is that it tries to account for all the variance and covariance of the set of items rather than the portion of the covariance that the items share in common. Thus it assumes there is no unique or error variance in each of the indicator variables. One reason PCFA is so widely used is because it is the default method in other widely used statistical packages, and you need to override this default in those programs to get a truer form of factor analysis. In Stata, PCFA is an option you need to specify and not the default. The Stata command for PCFA is simply `factor` *varlist*`, pcf`, where `pcf` stands for principal component factor analysis. Through the menu system, click on **Statistics** > **Multivariate analysis** > **Factor and principal component analysis** > **Factor analysis**.[1] In that dialog box, you list your variables under the **Model** tab. Under the **Model 2** tab, you pick *Principal-component factor*.

We will illustrate PCFA using actual data from the National Longitudinal Survey of Youth, 1997 (NLSY97). This is a longitudinal study that focuses on the transition from youth to adulthood. In 2006, when the participants were in their 20s, the NLSY97 asked a series of questions about the government being proactive in promoting well-being. The questions covered such topics as providing decent housing, college aid, reducing the income differential, health care, and providing jobs. We are interested in using 10 items to create a measure of conservatism. In the `nlsy97cfa.dta` dataset, these items are named `s8646900`–`s8647800`; for simplicity, we have renamed them `x1` to `x10`. The commands appear in a do-file called `ch1.do`, which you can find at http://www.stata-press.com/data/dsemusr/ch1.do. The dataset is located at

```
. use http://www.stata-press.com/data/dsemusr/nlsy97cfa.dta
```

[1] Warning: When you go to **Statistics** > **Multivariate analysis** > **Factor and principal component analysis**, do not then pick **Principal component analysis (PCA)** from the menu. This is intended to extract principal components, linear combinations of the variables, rather than factors.

Before constructing a scale, we need to examine the items. As you can see below, the responses range from 1 (meaning the task definitely should be the role of government) to 4 (meaning the task definitely should not be the role of government). Higher scores indicate that the person is more conservative.

```
. codebook x1-x10, compact
Variable    Obs Unique      Mean  Min  Max  Label
--------------------------------------------------------------------------------

x1         1833      4  2.331697    1    4  GOVT RESPONSIBILITY - PROVIDE JOB...
x2         1859      4  1.620226    1    4  GOVT RESPNSBLTY - KEEP PRICES UND...
x3         1874      4  1.416222    1    4  GOVT RESPNSBLTY - HLTH CARE FOR S...
x4         1872      4  1.365385    1    4  GOVT RESPNSBLTY -PROV ELD LIV STA...
x5         1815      4  1.773003    1    4  GOVT RESPNSBLTY -PROV IND HELP 2006
x6         1811      4  2.276643    1    4  GOVT RESPNSBLTY -PROV UNEMP LIV S...
x7         1775      4  2.228732    1    4  GOVT RESPNSBLTY -REDUCE INC DIFF ...
x8         1875      4  1.309333    1    4  GOVT RESPNSBLTY -PROV COLL FIN AI...
x9         1847      4  1.705468    1    4  GOVT RESPNSBLTY -PROV DECENT HOUS...
x10        1860      4   1.39086    1    4  GOVT RESPNSBLTY -PROTECT ENVIRONM...
--------------------------------------------------------------------------------
```

A PCFA can be run on these items by using a very simple command:

```
. factor x1-x10, pcf
(obs=1617)

Factor analysis/correlation                        Number of obs    =     1617
    Method: principal-component factors            Retained factors =        2
    Rotation: (unrotated)                          Number of params =       19

    --------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+------------------------------------------------------------
        Factor1  |      3.91523      2.90094            0.3915       0.3915
        Factor2  |      1.01429      0.13285            0.1014       0.4930
        Factor3  |      0.88144      0.11496            0.0881       0.5811
        Factor4  |      0.76648      0.02404            0.0766       0.6577
        Factor5  |      0.74243      0.04889            0.0742       0.7320
        Factor6  |      0.69354      0.08649            0.0694       0.8013
        Factor7  |      0.60705      0.06820            0.0607       0.8620
        Factor8  |      0.53886      0.09140            0.0539       0.9159
        Factor9  |      0.44746      0.05424            0.0447       0.9607
       Factor10  |      0.39322            .            0.0393       1.0000
    --------------------------------------------------------------------------

    LR test: independent vs. saturated:  chi2(45) = 4083.46 Prob>chi2 = 0.0000

Factor loadings (pattern matrix) and unique variances

    -------------------------------------------------
        Variable |  Factor1   Factor2 |   Uniqueness
    -------------+--------------------+--------------
              x1 |   0.6064   -0.3789 |      0.4888
              x2 |   0.5810    0.0438 |      0.6605
              x3 |   0.7221    0.2140 |      0.4328
              x4 |   0.7174    0.3200 |      0.3830
              x5 |   0.5780   -0.0261 |      0.6653
              x6 |   0.6091   -0.4536 |      0.4233
              x7 |   0.6050   -0.3327 |      0.5233
              x8 |   0.5994    0.3252 |      0.5350
              x9 |   0.7330   -0.1621 |      0.4365
             x10 |   0.4543    0.5211 |      0.5221
    -------------------------------------------------
```

These results indicate that the first factor is very strong with an eigenvalue of 3.92. The eigenvalue is how much of the total variance over all the items is explained by the first factor. By using the display command, we can compute the first eigenvalue as the sum of the squared factor loadings for the first factor:

```
. display .6064^2+.5810^2+.7221^2+.7174^2+.5780^2+.6091^2+.6050^2 +.5994^2+
> .7330^2+.4543^2
3.9154428
```

The PCFA analyzes the correlation matrix where each item is standardized to have a variance of 1.0. Therefore, with 10 items, the eigenvalues combined will add up to 10. With 3.92 out of 10 being explained by the first factor, we say the first factor explains 39.2% of the variance in the set of items. Any factor with an eigenvalue of less than 1.0 can usually be ignored.

The second factor has an eigenvalue of 1.01, which is very weak though it does not strictly fall below the 1.0 cutoff. We decide that the first factor, explaining 39.2% of the variance in the 10 items, is the only strong factor. This is reasonably consistent with our intention to pick items that tap a single dimension. We do not have an explicit test of a single-factor solution, but the eigenvalue of 3.92 is large enough to be reasonably confident that all the items are tapping a single dimension. Notice that all the loadings of the items of Factor1 are substantial, varying from 0.45 to 0.73. This range is also good when compared to conventions of the loadings being 0.4 or above. Some authors feel a loading of at least 0.30 is the minimum criterion for an item (Costello and Osborne 2005). You may recall that with the PCFA, the loadings are the correlation between how people respond to each item and the underlying, latent dimension.

Even though the last item has a loading over 0.40, its loading is considerably weaker than the rest of the items. The last item is about the environment, which can be a personal concern of anyone, whether conservative or not. By contrast, the other nine items involve government response to needs people have because of their limited personal resources. Because there is a second factor with an eigenvalue greater than 1.0 and because the loading of the tenth item on the first factor is the weakest, we will drop that item and rerun our analysis to see if we can obtain a clearer result.

```
. factor x1-x9, pcf
(obs=1625)
Factor analysis/correlation                        Number of obs    =     1625
    Method: principal-component factors            Retained factors =        1
    Rotation: (unrotated)                          Number of params =        9

    --------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+------------------------------------------------------------
        Factor1  |      3.76124      2.80650            0.4179       0.4179
        Factor2  |      0.95473      0.10627            0.1061       0.5240
        Factor3  |      0.84847      0.10176            0.0943       0.6183
        Factor4  |      0.74671      0.05561            0.0830       0.7012
        Factor5  |      0.69110      0.07429            0.0768       0.7780
        Factor6  |      0.61681      0.07780            0.0685       0.8466
        Factor7  |      0.53900      0.09177            0.0599       0.9065
        Factor8  |      0.44723      0.05252            0.0497       0.9561
        Factor9  |      0.39471            .            0.0439       1.0000
    --------------------------------------------------------------------------
    LR test: independent vs. saturated:  chi2(36) = 3863.18 Prob>chi2 = 0.0000
Factor loadings (pattern matrix) and unique variances

    ---------------------------------------
        Variable |  Factor1 |   Uniqueness
    -------------+----------+--------------
              x1 |   0.6243 |       0.6103
              x2 |   0.5883 |       0.6539
              x3 |   0.7222 |       0.4785
              x4 |   0.7131 |       0.4915
              x5 |   0.5818 |       0.6615
              x6 |   0.6197 |       0.6160
              x7 |   0.6085 |       0.6297
              x8 |   0.5968 |       0.6439
              x9 |   0.7392 |       0.4535
    ---------------------------------------
```

Most researchers would be quite happy with these results. Only one factor has an eigenvalue greater than 1.0, and all nine items load over 0.5 on that factor.

Many researchers ignore the results shown in the column labeled `Uniqueness`. These values represent the unique variance or error variance. For example, 61% of the variance in indicator variable `x1` is not accounted for by the factor solution. The principal component factor method assumes that these uniquenesses are 0. The uniquenesses are sufficiently large that we should consider using a different method for performing exploratory factor analysis, such as the default principle factor method, which does not assume that the uniquenesses are 0. Had we instead typed `factor x1-x9, pf`, the results would have been similar, having only one factor with an eigenvalue greater than 1 and with factor loadings ranging from 0.51 to 0.69 on that factor.

Let us proceed with just the first nine items.

1.4 Alpha reliability for our nine-item scale

The next step is to assess the reliability of our nine-item scale of conservatism. We will use the alpha command with three options: the item option gives us item analysis, the label option includes labels of our variables (which can make output look messy if you have long labels), and the asis option (which stands for "as is") does not let Stata reverse-code items to get them to fit better. If you have an item that is coded in the opposite direction, you should reverse-code it yourself before running the analysis.

Here is the alpha command with results. Stata can estimate alpha using the variance and covariances (unstandardized, the default) or the correlations (standardized). Because we are going to generate mean or total scores, we will estimate the unstandardized value. The unstandardized version is recommended when generating a scale score using unstandardized variables.

```
. alpha x1-x9, item label asis

Test scale = mean(unstandardized items)

Items        |  S  it-cor  ir-cor  ii-cov   alpha  label
-------------+--------------------------------------------------------
x1           |  +   0.664   0.505  .19857   0.789  GOVT RESPONSIBILITY -
             |                                       PROVIDE JOBS 2006
x2           |  +   0.589   0.454  .21848   0.793  GOVT RESPNSBLTY - KEEP
             |                                       PRICES UND CTRL 2006
x3           |  +   0.669   0.573  .21577   0.781  GOVT RESPNSBLTY - HLTH CARE
             |                                       FOR SICK 2006
x4           |  +   0.658   0.568  .21954   0.783  GOVT RESPNSBLTY -PROV ELD
             |                                       LIV STAND 2006
x5           |  +   0.582   0.441  .21865   0.795  GOVT RESPNSBLTY -PROV IND
             |                                       HELP 2006
x6           |  +   0.650   0.503  .20456   0.788  GOVT RESPNSBLTY -PROV UNEMP
             |                                       LIV STAND 2006
x7           |  +   0.656   0.487  .19844   0.793  GOVT RESPNSBLTY -REDUCE INC
             |                                       DIFF 2006
x8           |  +   0.540   0.441   .2348   0.797  GOVT RESPNSBLTY -PROV COLL
             |                                       FIN AID 2006
x9           |  +   0.717   0.622  .20509   0.774  GOVT RESPNSBLTY -PROV DECENT
             |                                       HOUSING 2006
-------------+--------------------------------------------------------
Test scale   |                     .21262   0.807  mean(unstandardized items)
--------------------------------------------------------------------
```

Our scale looks great by conventional standards. At the bottom of the table in the row labeled Test scale, we have the alpha for our scale. The alpha is 0.81, which is over the 0.70 minimum value standard. Under the column labeled alpha, we see what would happen if we dropped any single item from our scale; in each case, the alpha would go down. If dropping an item (one at a time) would substantially raise the alpha, we might look carefully at the item to make sure it was measuring the same concept as the other items. Most likely, the PCFA would have spotted such a problematic item as not fitting the first factor.

To obtain our scale score for each person in our sample, we would simply compute the total or mean score for the nine items. I usually prefer the mean score of the items, because it will be on the same scale as the original items (for these items, between 1 to 4). Given this, a mean of 3.0 would denote that a person is conservative and does not support a proactive government. A mean of 1.5 would denote that the person is fairly liberal, between definitely and probably supporting a proactive government.

By contrast, a total score would range from 9 to 36, and it would be much harder to interpret a total score of, say, 24.0 (instead of 3.0) or 12.0 (instead of 1.5). Another problem with the total score arises if there are missing values for some items. An item with a missing value would contribute nothing to the total, as if we had assigned that item a value of 0.0. If a person skips an item, giving them a score of 0 for that item is ridiculous because that would indicate more definite support of a proactive government than the most favorable available response that is coded as 1.0.

To obtain the mean score for each person, we generate our scale score as the mean of the items the person answered. This egen (extended generation) command gives you the mean of however many of the nine items the person answered:

```
. egen conserve = rowmean(x1-x9)
(7097 missing values generated)
```

The egen command shows that there are 7,097 missing values on our generated conserve variable. This is not a problem because the item was only asked for a subset of the overall dataset. The summarize command below tells us that the mean is 1.78, the standard deviation is 0.51, and this is based on 1,888 observations. These 1,888 observations include anybody who answered at least one of the items (see box 2.1 for alternative treatments of missing values). The histogram with a normal distribution overlay (figure 1.1) shows that our score is pretty skewed to the right with a concentration of people favoring a proactive government.

```
. summarize conserve, detail
                          conserve
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              1
10%     1.111111              1       Obs                1888
25%     1.354167              1       Sum of Wgt.        1888
50%     1.690476                      Mean           1.775299
                        Largest       Std. Dev.      .5132186
75%     2.111111       3.888889
90%     2.444444       3.888889       Variance       .2633934
95%     2.666667              4       Skewness       .7200074
99%     3.222222              4       Kurtosis       3.537959
```

```
. histogram conserve, norm freq
(bin=32, start=1, width=.09375)
```

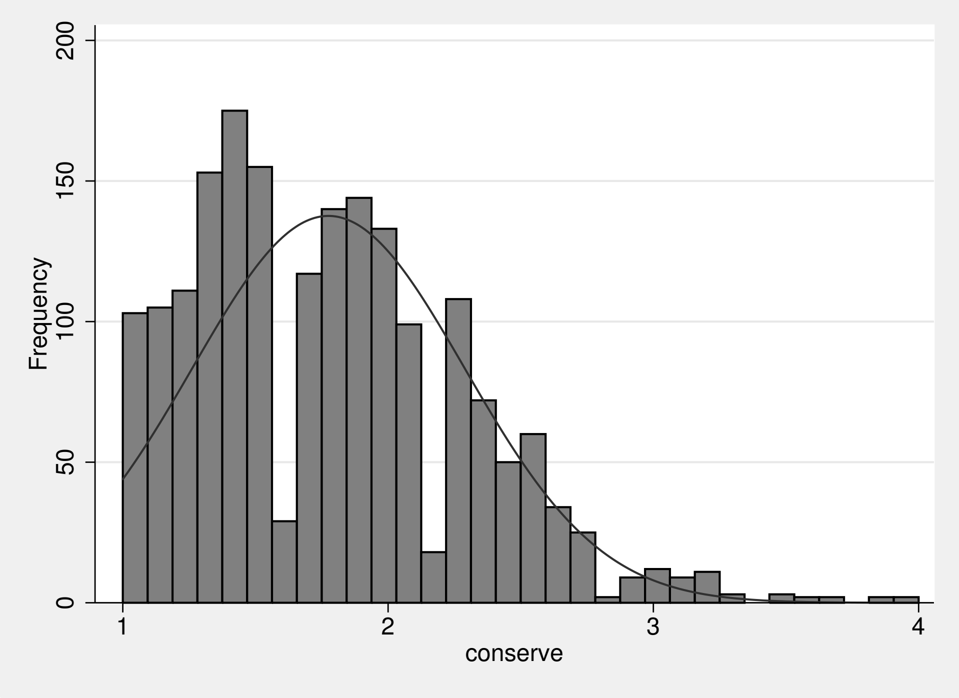


Figure 1.1. Histogram of generated mean score on conservatism

1.5 Generating a factor score rather than a mean or summative score

When we generated our `conserve` scale using the traditional approach, we simply got the mean of the nine items. This method counts each item as equally relevant to the concept being measured. If all the items are equally important, we say that the items are τ ("tau") equivalent. If this were true, then each item would have an identical loading; rarely is this the case. An item that has a loading of 0.90 on a factor is more salient than one that has a loading of 0.30. Therefore, the item with the larger loading should be given a greater weight when we generate the scale score.

You can generate a factor score that weights each item according to how salient it is to the concept being measured. Factor scores will be extremely highly correlated with the simple mean or summative score whenever the loadings are all fairly similar. If the loadings vary widely, the factor score will be a better score to use because factor scores weight items by their salience (loadings and correlations with the other items), but the advantage is only substantial when some items have much weaker loadings than others. The factor score will be scaled to have a mean of 0.0 and a variance of 1.0; in other words, it will be the standardized score for the concept.

To generate a factor score on the first factor, we run the postestimation command `predict` immediately after the `factor` command. The `predict` command we use includes only those participants who answered all nine items. This means we may have a substantially smaller N if several participants have skipped at least one of the items. This casewise deletion can be a serious limitation because we normally want to use all available data.

```
. factor x1-x9, pcf
  (output omitted)
. predict conservf1
(regression scoring assumed)
Scoring coefficients (method = regression)

    ----------------------
        Variable | Factor1
    -------------+--------
              x1 | 0.16598
              x2 | 0.15641
              x3 | 0.19200
              x4 | 0.18958
              x5 | 0.15468
              x6 | 0.16475
              x7 | 0.16179
              x8 | 0.15866
              x9 | 0.19654
    ----------------------
```

We do not need any options on the `predict` command because the default is to generate the factor score for the first factor. By contrast, the `egen conserve = rowmean(x1-x9)` we used in the previous section computed the mean of however many items a person answered so long as they answered at least one item. We have 1,625 people who answered all nine items and 1,888 people who answered at least one of the nine items; therefore, the `egen` command retains more observations.[2]

The results above show us the factor scoring coefficients, which are like standardized beta weights. Notice that the ninth item has a scoring coefficient of 0.20 and the second item has a scoring coefficient of 0.16. This means the ninth item counts slightly more in generation of the factor score, which makes sense because the ninth item had a bigger loading than the second item (0.74 > 0.59).

The default for the `predict` command is to predict the factor score as the weighted sum of the items using the scoring coefficient as the weight for each item. The factor score should be more reliable than the summative or mean score because it more optimally weights the items.

[2] To get the mean for only those who answered all nine items, we would have used `egen conservm = rowmean(x1-x9) if !missing(x1, x2, x3, x4, x5, x6, x7, x8, x9)`. Stata reads `!missing` as "not missing". Notice the items are all listed and separated by a comma; the commas are necessary for this command.

How much does it matter whether you compute a mean/summative score or a factor score? The correlation between the average score on conservatism and the factor score is $r = 0.99$. Thus it does not matter which approach you use in this example, except for the different ways of handling missing values on skipped items. Here is a graph comparing the distributions of the two variables. The factor score would be more reliable when the items varied substantially in their loadings and, hence, their factor scoring coefficients. The mean or summative score would use more information if there were a lot of missing values.

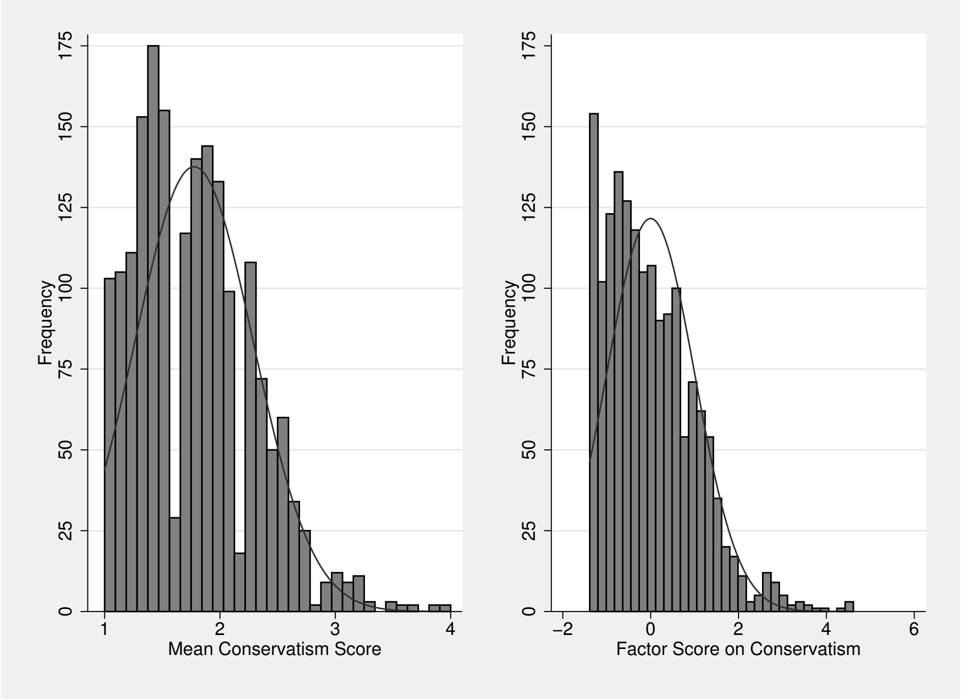


Figure 1.2. Generated mean score on conservatism versus factor score on conservatism

1.6 What can CFA add?

One thing that both CFA and factor analysis methods other than PCFA do is to allow each item to have its own unique variance. This is illustrated in figure 1.3, where each observed question (x_1–x_9) has a corresponding error term, ϵ_1–ϵ_9. These error terms allow for variance in the responses to the question that are unique to the item and do not reflecting the shared variance of the nine items. The latent variable, `Conservative`, appears in the oval and is what the nine items share; the ϵ's are what is unique about each item. It is usual to assume that the error terms are normally distributed and uncorrelated (this assumption will be relaxed in later sections of the book).

CFA assumes that the latent variable accounts for how people respond to all nine individual questions, which is what the nine items share in common. Notice the direction of the arrows from `Conservative` to each of the nine items; the arrows take us from the latent variable to the observed items. This is because how people respond to a question is the dependent variable; that is, a person's response depends on how conservative he or she is, the independent variable. Because all the items seem to tap conservatism, we will posit that a single factor is all we need, and so we draw the single-factor model seen in figure 1.3.

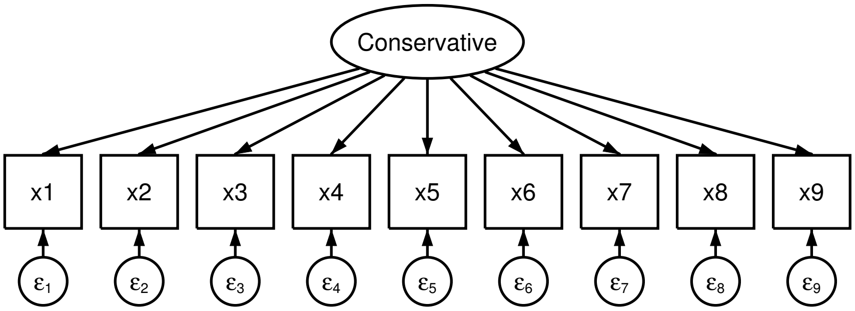


Figure 1.3. CFA for nine-item conservatism scale

This is a confirmatory model because we have specified the factor that underlies the responses to these nine items; that is, all the items are indicators of conservatism. When we ran PCFA, we hoped there would be a single dominant factor. With CFA, we specify the number of factors. In this example, we specified that the covariance of the nine items is fully explained by the single latent variable plus the unique variance of each item. Notice that we are estimating the unique variance or error variance for each of the nine observed indicator variables (items). In the PCFA, we assumed conservatism had to explain all the variance among the nine items. Here we acknowledge that each item may have some unique variance that we are treating as random error. We assume the error variables are normally distributed with a mean of 0.

There are real advantages to CFA. By isolating the shared variance of the nine questions from their unique variances, we are able to obtain a better measure of the latent variable. We are also likely to get stronger results by removing measurement error if the latent variable is subsequently used as an independent or dependent variable in a structural equation model. This is because measurement error, by its nature, only adds noise to our measurement; it has no explanatory power.

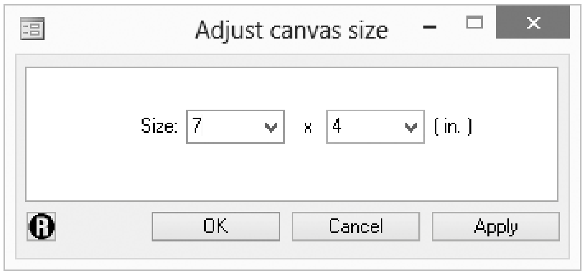
Box 1.1. Using the SEM Builder to draw a model

Appendix A provides an introduction to using the extremely capable drawing package that Stata offers, the SEM Builder. Here I will just show how we created figure 1.3. In the Command window, type in `sembuilder` to open the drawing program.

Select the **Add Measurement Component** tool, , and then click within the SEM Builder wherever you want the latent variable to appear (for our purposes, you will want it to appear in the middle horizontally and a bit high vertically).

In the box labeled *Latent variable name*, type `Conservative` (remember that our convention is to capitalize the first letter of a latent variable). In the box labeled *Measurement variables*, choose the variables `x1` through `x9` from the drop-down menu (assuming you have the `nlsy97cfa.dta` dataset open). Make sure the *Measurement direction* is `Down`. Click **OK**.

With nine indicator variables, the default size for observed variable boxes will cause the diagram to be wider than the default size of the canvas. To see the full diagram, click on the **Adjust Canvas Size** button, , and set the canvas size to `7` × `4`.


Adjust canvas size

Size: 7 x 4 (in.)

This new canvas size will be large enough to accommodate the full diagram. However, you may not yet be able to see the full canvas. Click on the **Fit in Window** button, , to see the full canvas in the Builder window. If a portion of the diagram is not on the canvas, click on the **Select** tool, , and drag it over the model so that all objects are highlighted. Then move the diagram until you see the entire diagram on the canvas.

Box 1.1. (*continued*)

Because `Conservative` is so long, it does not fit in the default size oval for a latent variable. To make the oval larger, select **Settings** > **Variables** > **All Latent...** from the menu (on a Mac, click the **Tools** button, ≡ Tools, in the upper right to find the **Settings** menu.) In the dialog box that opens, change the size to 0.75 × 0.38. You can also change the size of the boxes for observed variables through the **Settings** > **Variables** menu if you like.

Should you need to copy this diagram to another document, such as a Word document, you can do this with the standard copy and paste commands. You can use the **Adjust Canvas Size** button if you want to specify the exact width and height of the object that is copied. Then click on the **Copy Diagram** button, ⎘. Now the diagram is ready to paste into another document. More detail appears in the chapter 1 appendix.

With so many indicators, it should be clear now why you want short names for your variables. I used the `clonevar` command to rename the variables because their original names in the dataset were long and unclear, for example, `clonevar x1 = s8332500`.

1.7 Fitting a CFA model

We can fit a CFA model by using the Stata command language directly or by using the SEM Builder. Here I will show how to do this with the commands, to ensure that you understand them. The chapter 1 appendix then replicates selected results with the SEM Builder.

The Stata command to fit our CFA model is simple. We do need to run a set of four commands, but each of them is quite simple. First, to fit the model, we run

```
. sem (Conservative -> x1-x9)
```

By running this command, we have the name of our latent variable, `Conservative`, and the `->` points from the latent variable to its indicators, `x1-x9`, just like in figure 1.3.[3] The direction of the arrow is sometimes difficult for beginners to grasp. The idea is that a person's response to each item is caused by how conservative he or she is. That is, your response to an item does not cause you to be conservative; rather, your level of conservatism causes your response. The latent variable here is the independent variable, and the indicators are the dependent variables. We have not specified any options. We have four possible estimation methods:

[3] Note that the name of the latent variable should be capitalized to help us distinguish indicators, which should be all lowercase, from latent variables.

1. The default is `method(ml)` which means that we fit the model using maximum likelihood estimation. By default, when using `method(ml)`, the variance–covariance matrix of the estimators (and therefore the standard errors) is computed using an observed information matrix. Where you assume normality, `method(ml)` is often the best option and is fairly robust even with some violation of normality. This uses listwise deletion.

2. When option `method(ml)` is combined with option `vce(robust)`, `sem` performs quasi maximum likelihood estimation, and the standard errors are estimated in a manner that does not assume normality. This uses the Huber–White sandwich estimator of the variance–covariance matrix of the estimators. Because several of our items are clearly not normally distributed, this might be a good option to use. The robust standard errors are less efficient than the observed information matrix standard errors if the assumptions of maximum likelihood estimation are met. This uses listwise deletion.

3. The option `method(adf)` is asymptotically distribution free. This method makes no normality assumptions and is a form of weighted least squares. It is also less efficient than maximum likelihood where that is appropriate, but more efficient than the quasi maximum likelihood estimation. Because it does not assume normality and is asymptotically equivalent (in a large sample) to maximum likelihood, this may be the best option for our data. This uses listwise deletion.

4. The option `method(mlmv)` is appropriate when you want to use all the information available in the presence of missing values on one or more variables. This method assumes joint normality and that the missing values are missing at random. This does not use listwise deletion. In our example, we would have an $N = 1888$ using the `method(mlmv)` option, whereas with any of the other three estimators our $N = 1665$.

You can also use the `vce(bootstrap)` option to estimate the standard errors with the bootstrap procedure. This method will resample your observations with replacement and fit the model however many times you specify. It will then use the distribution of the parameter estimates across these replications to estimate your standard error. This will be especially useful when you are concerned about violating the normality assumption of the maximum likelihood options. For example, you might run the following command:

```
. sem (Conservative -> x1-x9), vce(bootstrap, reps(1000) seed(111))
```

We are using the `vce(bootstrap)` option and specifying `reps(1000)`, which means that we are drawing 1,000 samples for our replications. The `seed(111)` option is used so that we can replicate our results; you will get different results each time you run the command unless you set a seed.

For now, we will just use the default version of the command. Here are the results:

```
. sem (Conservative -> x1-x9)
(7360 observations with missing values excluded)

Endogenous variables

Measurement:  x1 x2 x3 x4 x5 x6 x7 x8 x9

Exogenous variables

Latent:       Conservative

Fitting target model:

Iteration 0:   log likelihood = -15604.985
Iteration 1:   log likelihood = -15594.134
Iteration 2:   log likelihood =  -15593.73
Iteration 3:   log likelihood = -15593.729

Structural equation model                       Number of obs      =      1625
Estimation method  = ml
Log likelihood     = -15593.729

 ( 1)  [x1]Conservative = 1
------------------------------------------------------------------------------
              |                 OIM
              |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
Measurement   |
  x1 <-       |
  Conservat~e |          1  (constrained)
        _cons |   2.329846   .0253521    91.90   0.000     2.280157    2.379535
  ------------+----------------------------------------------------------------
  x2 <-       |
  Conservat~e |   .7377011   .0451423    16.34   0.000     .6492237    .8261784
        _cons |   1.617231   .0198829    81.34   0.000     1.578261    1.656201
  ------------+----------------------------------------------------------------
  x3 <-       |
  Conservat~e |   .8267157   .0432635    19.11   0.000     .7419209    .9115105
        _cons |   1.414154   .0167434    84.46   0.000     1.381337     1.44697
  ------------+----------------------------------------------------------------
  x4 <-       |
  Conservat~e |   .7555335   .0403806    18.71   0.000      .676389     .834678
        _cons |   1.362462   .0155865    87.41   0.000     1.331913     1.39301
  ------------+----------------------------------------------------------------
  x5 <-       |
  Conservat~e |   .7380149   .0462134    15.97   0.000     .6474383    .8285914
        _cons |   1.769846   .0202603    87.36   0.000     1.730137    1.809556
  ------------+----------------------------------------------------------------
  x6 <-       |
  Conservat~e |   .9146378    .053406    17.13   0.000     .8099639    1.019312
        _cons |   2.259692   .0229301    98.55   0.000      2.21475    2.304634
  ------------+----------------------------------------------------------------
  x7 <-       |
  Conservat~e |   1.028027   .0614681    16.72   0.000     .9075522    1.148503
        _cons |   2.219692   .0266439    83.31   0.000     2.167471    2.271913
  ------------+----------------------------------------------------------------
  x8 <-       |
  Conservat~e |   .5486913    .033463    16.40   0.000      .483105    .6142775
        _cons |   1.307077   .0141374    92.46   0.000     1.279368    1.334786
  ------------+----------------------------------------------------------------
 x9 <-          |
 Conservat~e    |   .9278118   .0479147    19.36   0.000     .8339008    1.021723
       _cons    |   1.705231   .0187041    91.17   0.000     1.668571     1.74189
----------------+----------------------------------------------------------------
     var(e.x1)  |   .7287257   .0280851                      .6757076    .7859038
     var(e.x2)  |   .4706031   .0178489                      .4368885    .5069195
     var(e.x3)  |   .2397812   .0104761                      .2201029    .2612188
     var(e.x4)  |   .2145611    .009255                      .1971672    .2334895
     var(e.x5)  |   .4950753   .0186802                      .4597838    .5330757
     var(e.x6)  |    .590299   .0229507                      .5469876    .6370399
     var(e.x7)  |   .8199315   .0314634                      .7605262    .8839769
     var(e.x8)  |   .2297334   .0087974                       .213122    .2476396
     var(e.x9)  |   .2967257   .0129788                      .2723476    .3232858
var(Conserv~e)  |   .3157048   .0287081                       .264167    .3772973
---------------------------------------------------------------------------------
LR test of model vs. saturated: chi2(27)  =    419.01, Prob > chi2 = 0.0000
```

1.8 Interpreting and presenting CFA results

At the top of the results, we see that we have 7,360 observations with missing values excluded. The default estimation method, maximum likelihood, uses listwise deletion and drops any observations that do not have a response for all nine of our items.[4] The results next report our endogenous (dependent) variables. All of our observed items, `x1` to `x9`, are endogenous; that is, these measurement variables depend on the latent variable. We next have a list of exogenous variables. Stata reports just one latent exogenous variable, `Conservative`; Stata does not list the measurement-error terms ϵ_1 to ϵ_9 here even though these are also latent exogenous variables.

The maximum likelihood estimator maximizes the log-likelihood function. Stata converges quite quickly, taking just three iterations. We do not use the log-likelihood function directly. Notice that with listwise deletion, we only have 1,625 observations that have no missing values.

The results above include a section labeled `Measurement` and a section reporting variances. The measurement section gives estimates of the unstandardized measurement coefficients (factor loadings), their standard errors, and a z test for each estimate along with a 95% confidence interval. By contrast, our PCFA estimates only included factor loadings. The variance section shows the estimates of the variances of the error terms, ϵ_1 to ϵ_9. In the column labeled `Coef.` appears the unstandardized solution. To identify the variance of the latent variable, `Conservative`, Stata fixes the loading of the first indicator at 1.0. The indicator that has its loading fixed at 1.0 is called the reference indicator. All the unstandardized estimates will change if you change the reference indicator. It is a good idea to have one of the stronger indicators be the reference indicator. If you want the second indicator to be the reference indicator, you would simply list the variables with that indicator appearing first: `sem (Conservative -> x2 x1 x3-x9)`.

[4] If we had wanted a full information approach that utilized all available information, we would have specified `sem (Conservative -> x1-x9), method(mlmv)`.