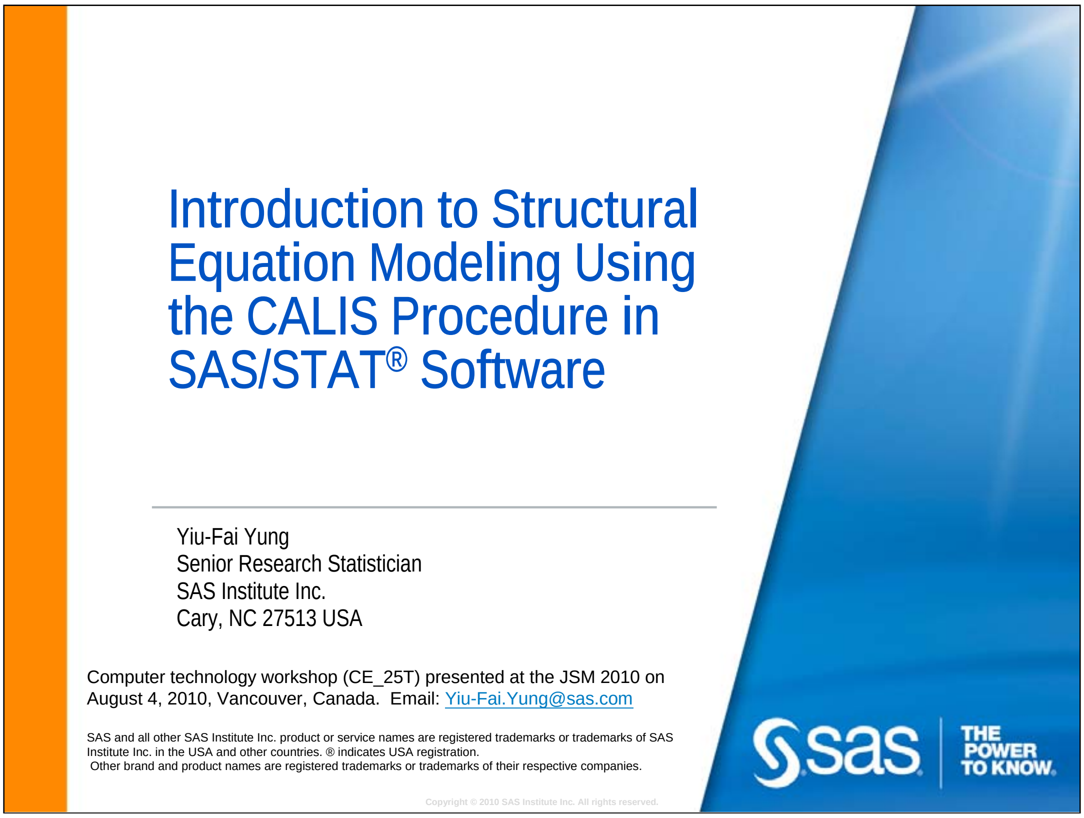


Abstract

The CALIS procedure in SAS/STAT is a general structural equation modeling (SEM) tool. This workshop introduces the general methodology of SEM and the applications of the CALIS procedure. Historical topics such as casual models, path diagram, confirmatory factor-analysis, measurement error model, and linear structural relations (LISREL) are reviewed. Applications of the CALIS procedure to SEM are demonstrated with examples in social, educational, behavioral, and marketing research. Specifically, the following how-to techniques of the CALIS procedure (SAS/STAT 9.22) are covered: (1) Specifying structural equation models with latent variables by using the PATH modeling language; (2) Interpreting the model fit statistics and estimation results; (3) Testing models with multiple groups and multiple models; (4) Analyzing direct and indirect effects; (5) Modifying structural equation models.

This workshop is designed for statisticians and data analysts who want to overview the applications of the SEM by the CALIS procedure. Attendees should have a basic understanding of regression analysis and experience using the SAS language. Previous exposure to SEM is useful, but not required.

Citation of this workshop:
Yung, Y.-F. (2010). Introduction to Structural Equation Modeling Using the CALIS Procedure in SAS/STAT® Software. Computer technology workshop presented at the Joint Statistical Meeting on August 4, 2010, Vancouver, Canada.

SAS/STAT 9.22 or later

is assumed

for this workshop

2

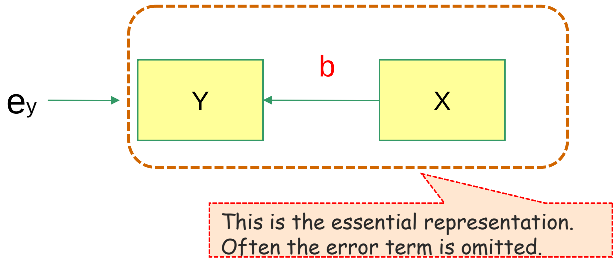
In this workshop, SAS/STAT 9.22 (TS2M3) or later is assumed for the CALIS procedure. Some of the code might work with PROC TCALIS (an experimental procedure) in SAS/STAT 9.2 (TS2M2). However, there is a major syntactical difference between PROC TCALIS and PROC CALIS. In PROC TCALIS, the parameter specification for each path in the PATH statement must **not** be preceded by an equal sign. But this equal sign is required in PROC CALIS when you specify parameters. Also, PROC TCALIS does not support the generalized path specifications (for variances, covariances, means, and intercepts) and multiple-path specifications when you use the PATH modeling language, which is the main focus of today's talk.

Causal Model, Prediction, and Path Diagram

- X causes Y
- X predicts Y
- Linear regression equation

  $Y = b\,X + e_y$

- Path diagram



The central idea of structural equation modeling is the study of causal relationship between variables. For example, you have an X and an Y variable. X is the cause of Y, or doing X results in Y. To give a more realistic example: eating more vegetables (X) brings down your cholesterol level (Y). However, this causal structure is only an idealized framework. In making causal inferences, you must have isolated all other background variables and establish temporal sequence of the variables. Because of the complicated philosophical issues involved in making causal inferences, in general SEM would avoid claiming causal inferences.

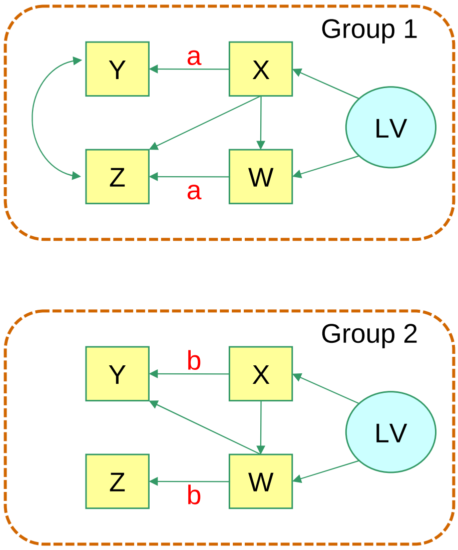
A predictor-outcome framework might be more appropriate philosophically. The semantic is now " X predicts Y". Mathematically and statistically, this idea is represented in the simple linear regression analysis, as shown in the linear regression equation:

$Y = b*X + e.$

The path diagram for this representation is shown in the slide, where b is called the effect , regression coefficient, or path coefficient. Notice that an error term is added to show that the prediction of Y from X is not perfect. But essentially, the predictor-outcome framework is represented by the Y←X path in the path diagram.

Structural Equation Modeling versus Regression Analysis

- More variables
- More equations
- Correlated errors
- Direct and indirect effects
- Latent variables
- Parametric constraints
- Multiple-group analysis



What are the differences between SEM and regression analysis? What more can SEM offer than the linear regression analysis?

You can view SEM as a much more complicated system for multiple predictor-outcome relationships. SEM can handle the following situations where linear regression analysis is of limited usefulness:

1.More variables (not just X and Y, but you can also add W and Z into the path diagram).
2.More equations or functional relationships (not just X→Y, but you can also analyze W→Z simultaneously).
3.Correlated errors, system of equations can have correlated errors . For example, the double-headed arrow between Y and Z.
4.Direct and indirect effects: X has a direct effect on Z and an indirect effect on Z via its effect on W. That is, X→Z and X→W→Z are direct and indirect effects, respectively.
5.Latent variables. For example, LV in the path diagram has effects on X and W.
6.Parametric constraints. For example, the constraints on the path coefficients or effects labeled as 'a' in the upper path diagrams.
7.Multiple-group analysis. For different groups of populations, the overall structure of the model are the same, but the path constraints could be different---while the constrained effect in Group 1 is denoted as 'a,' the constrained effect in Group 2 is denoted as 'b,' which will have a different estimate than that for 'a' in Group 1.

Other Names for Structural Equation Modeling (SEM)

- Path analysis
- LISREL model (Jöreskog 1973, Keesling 1972, Wiley 1973)
- Covariance structures analysis
- Analysis of moment structures
- Confirmatory factor analysis
- Causal modeling
- CALIS: Covariance Analysis of Linear Structural Equations

5

SEM has a lot of synonyms in the field: Path analysis (attributed to Sewall Wright), LISREL model (JKW model), covariance structures analysis, analysis of moment structures, confirmatory factor analysis, causal modeling, and etc. In terms of the statistical methodology involved, all these names are more or less the same.

PROC CALIS, which stands for covariance analysis of linear structural equations, is a software that was designed to handle all these analyses under the umbrella term SEM. Hopefully, one day PROC CALIS would also be remembered as a synonym of SEM.

SEM Software

- AMOS, EQS, LISREL, MPLUS, …
- Why PROC CALIS?
- Which is best?

6

There are several well-known software in the field for doing SEM: AMOS, EQS, LISREL, and MPLUS, and may be more. Why PROC CALIS? Which is best? Although these are very interesting questions, as the current developer of PROC CALIS procedure I am not at liberty to judge other SEM software. This workshop gives you an introductory tour of SEM with the use of PROC CALIS. Therefore, you might compare PROC CALIS with other software on your own after learning some of the features of PROC CALIS.

A Very Brief History of PROC CALIS

- Original developer: Wolfgang Hartmann (80’s)
- Influences
  - o Statistical/mathematical: COSAN (McDonald 1978, 1980)
  - o Syntax: EQS (Bentler 1985, 1995)
- TCALIS (SAS 9.2, 2008): experimental version
- “New” CALIS (SAS 9.22, 2010): PATH modeling language, multiple-group analysis, mean structures, name-free approach to parameter specifications, and much more

7

Let us start with a brief history of PROC CALIS.

In eighties, Wolfgang Hartmann designed and developed the first version of PROC CALIS. The statistical and mathematical model was greatly influenced by the COSAN model proposed by R. P. McDonald. In fact, there was evidence that Cosan, instead of Calis, might have been proposed as the name of the procedure. The most popular syntax in PROC CALIS, however, was under the influence of the EQS program by Peter Bentler. The LINEQS syntax in PROC CALIS for model specification is basically a twin brother of the syntax of the EQS program.

I picked up the development of the software around 2000. I actually rewrote the mathematical foundations of the software. I kept the optimization techniques and initial estimation techniques so that the new CALIS is compatible with the old CALIS.

In 2008, an experimental version called TCALIS was released. Since then, I have modified the syntax a little more and fixed some major bugs.

The new CALIS (SAS 9.22) has been released this year. If you have used PROC CALIS before, you will notice one major change: the emphasis on the PATH modeling language. You can see examples using the PATH statement everywhere in the PROC CALIS documentation. Other noteworthy new features are: multiple-group modeling, redesigned mean structure analysis, and the name-free approach to parameter specifications. Certainly, there are many more new features than these, as you will learn from this workshop and elsewhere.

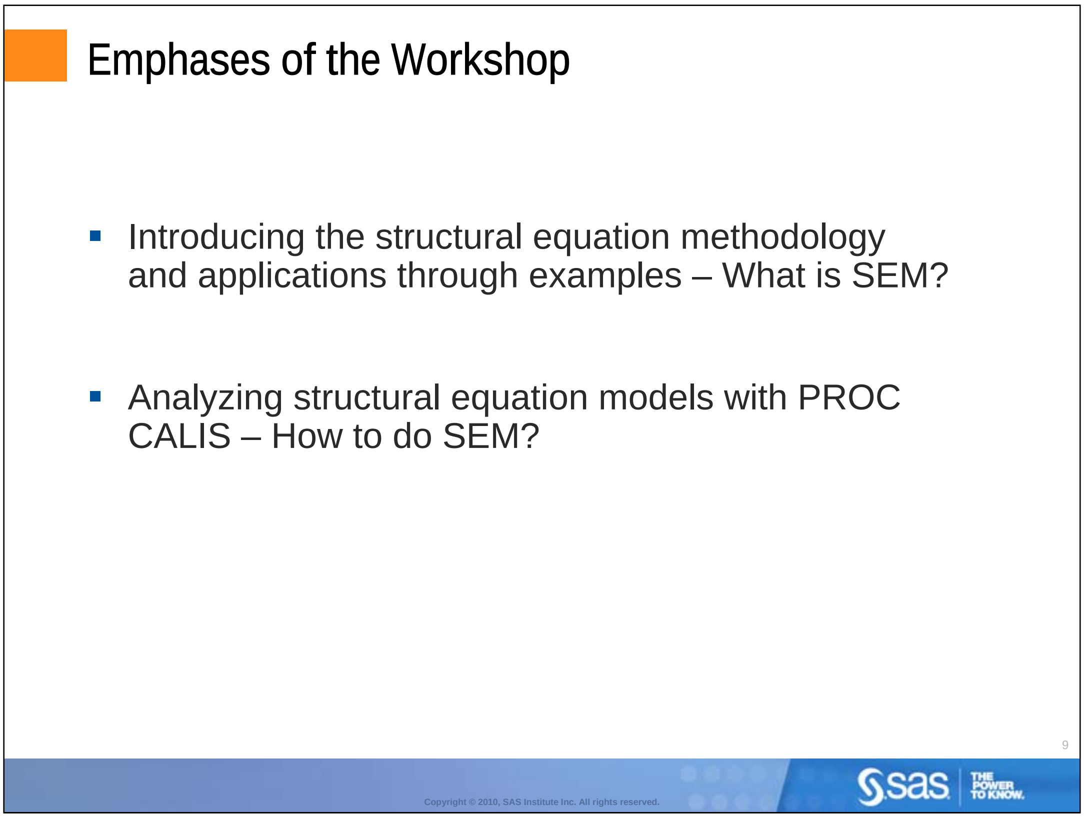
Structure of the Workshop

- First Part: Basic Modeling
  1. A brief description of the process of SEM
  2. The PATH modeling language in PROC CALIS
  3. Specifying models and interpreting results
  4. LISMOD – a language tailored to LISREL users
- Second Part: “Advanced” Modeling
  1. Multiple-group analysis
  2. Analyzing direct and indirect effects
  3. Testing specific hypotheses
  4. Model modifications

8

The first part of the workshop is about the basic SEM modeling using PROC CALIS. I will describe the research process of SEM briefly. Then I will introduce the PATH modeling language in PROC CALIS by using a simple linear regression example. Next, I will move on to more complicated examples that analyze confirmatory-factor models. I will use PROC CALIS in these examples to show how you can specify SEM models by the PATH modeling language, in relation to the path diagram representations. I will show you how to interpret the results generated by PROC CALIS. I will end the first part by showing you how a LISREL model can be specified by the LISMOD statement in PROC CALIS.

The second part of the workshop is about “advanced” modeling---relatively speaking. I will show how multiple-group analysis can be done in PROC CALIS. Other important topics such as direct and indirect effect analysis, testing specific hypotheses, and model modifications are discussed.

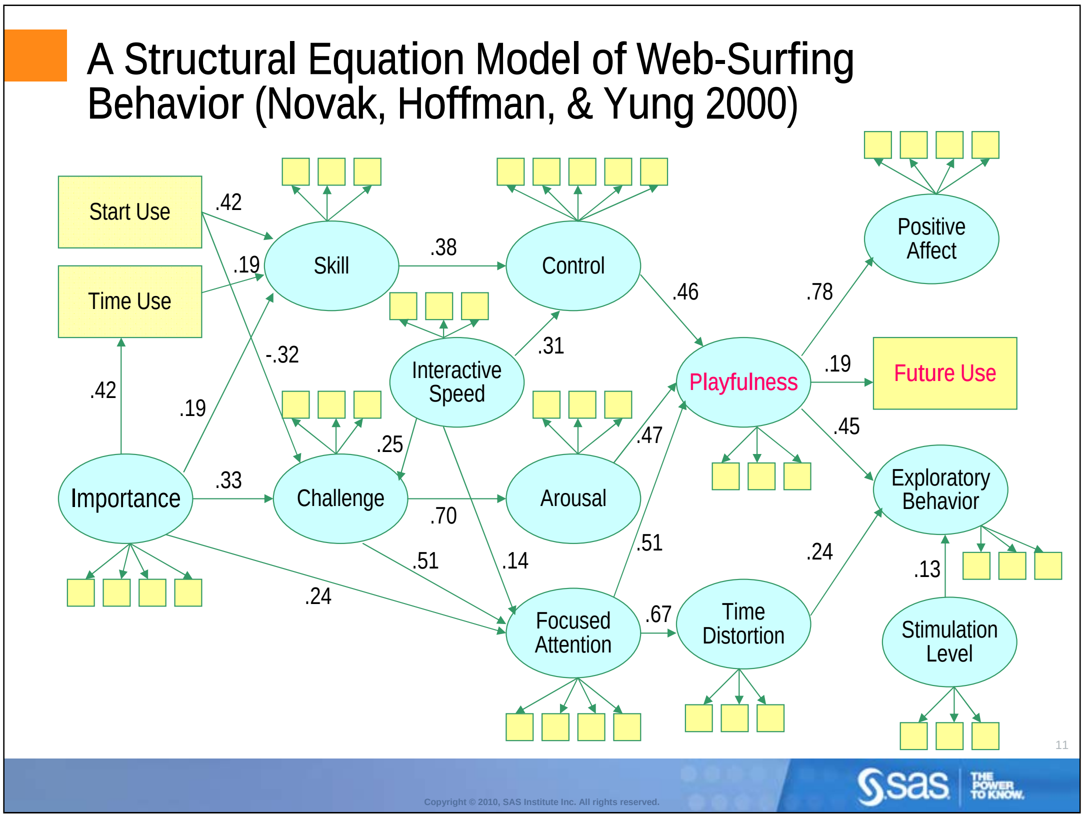


There are two emphases of this talk.

One, I want to show you an overall picture of SEM. This addresses the "what is SEM?" question. I will not give you a technical definition, but I will show you SEM examples so that you will have a "real" feeling about the applications of SEM.

Two, I want to show you how to use PROC CALIS. This addresses the "How to do SEM?" question. I hope that in the end of the workshop, you will find that PROC CALIS is very useful for SEM.

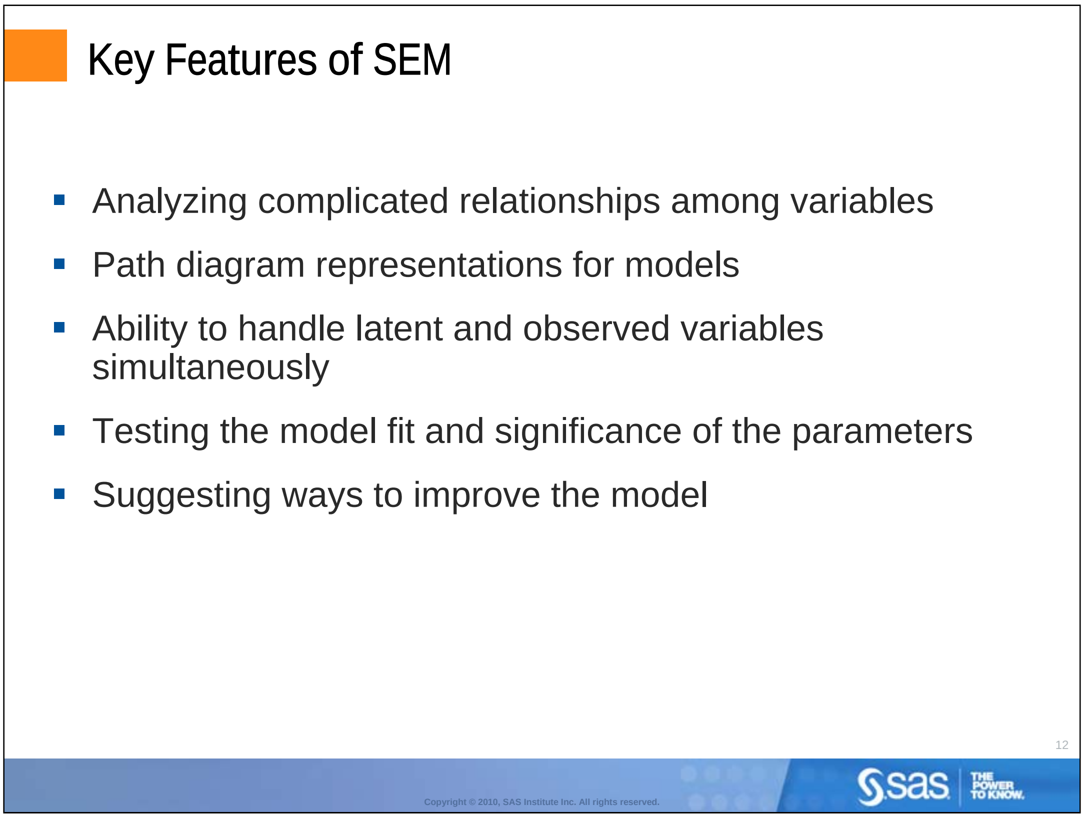
Illustrating the Process of Structural Equation Modeling



This is a structural equation model about web-surfing behavior. The researchers hypothesize that the "Playfulness" of a web-site would enhance the future use ("Future Use") of the same web-site. However, the theory does not end there. The researchers then hypothesize what would make a web-site to be perceived as playful. Three additional constructs are hypothesized in the path diagram: "Control" (of the web-page), "Arousal" (of interest), and "Focused Attention" are determinants of "Playfulness." In fact, the researchers hypothesized even further. For example, they use "Start Use" (when the users started to use computers) and "Time Use" (how often they use computers) as remote "causes" of a lot of latent constructs in the path diagram. In sum, this is a relatively large SEM that theorizes complicated relationships among constructs.

In this path diagram, the oval shapes represent latent variables, which are not measured but serve as useful constructs in the model (e.g., "Playfulness"). The rectangles represent measured or observed variables (e.g., "Start Use", "Time Use", "Future Use"). In order to analyze the latent constructs, some measured variables (or indicators) are needed. In the path diagram, those small unlabeled rectangles are measured indicators for the latent constructs. In this research, these measured indicators are rating responses on a questionnaire. For example, "I lost track of the time when using this web-site" (this is not an exact item from the actual research) could be an item for the "Time Distortion" construct.

Given this path diagram for the theory about web-surfing behavior, an SEM software fits the model based on the observed data and informs you the model fit and the estimates of the effects (path coefficients) in the path diagram. The SEM software also tells you the significance of these estimates. If the model does not fit the data well, the SEM software would suggest ways to improve the model.

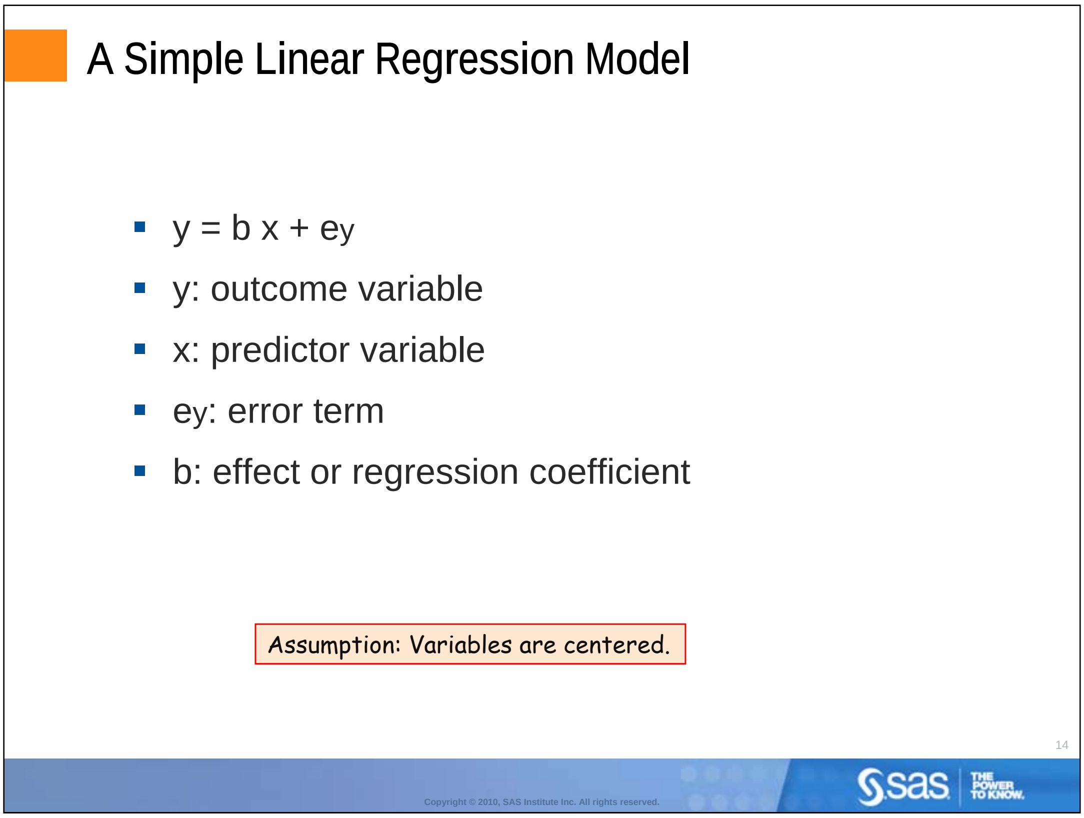


Here is a list of the key features of SEM:

- Analyzing complicated relationships among variables
- Path diagram representations for models
- Ability to handle latent and observed variables simultaneously
- Testing the model fit and significance of the parameters
- Suggesting ways to improve the model

Basics: A Simple Regression Model and the PATH Modeling Language

13



To introduce the PATH modeling language in PROC CALIS, a simple linear regression model is used. In the regression equation, y is the outcome variable, x is the predictor variable, e_y is the error term and b is the effect or regression coefficient. The regression model written in this form assumes that x and y are centered with means zero. But this assumption will not affect the generality to un-centered variables.

Measures of the Number of Hen Pheasants

- Fuller (1987) p.34
- y : average of the number of birds in August
- x : average of the number of birds in Spring (April/May)
- Averages were based on the number of birds sighted by 15 trained observers
- Goal: How many birds will survive 3 months?

15

On p.34 of Fuller's book "Measurement Error Models", he describes a data set about the counting of hen pheasants in April and August. Fifteen trained observers counted the number of birds in the two occasions. Y is the number of birds in August and X is the number of birds in April. The goal of the linear regression is to predict the number of birds in August (Fall) by the number of birds in April (Spring).

Regression Analysis by PROC REG

```
data hens;
   input y x @@;
   datalines;
8     9     6     6.6   9.8  12.3  10.8  11.9  9.7  11.9  9.3  12
9.2   9.6   6.9   7.5   8.1  10.9   8.7  10.4  8.7  10.2  7.4  7.4
10.1  11    10   11.8   7.3   8.2
;
proc reg data=hens;
   model y = x;
run;
```

16

To conduct a linear regression analysis, you can use a SAS procedure called PROC REG. The syntax is quite simple. First, define your data set. Second, call PROC REG with the interested data set specified in the PROC REG statement. Then, the model statement specifies that y = x, which means y is predicted by x. No error term needs to be specified, although PROC REG does assume that prediction is not perfect so that the error does exist with nonzero variance in the regression.

Results Obtained from PROC REG

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value
Intercept	1	2.14227	0.84513	2.53
x	1	0.64941	0.08275	7.85

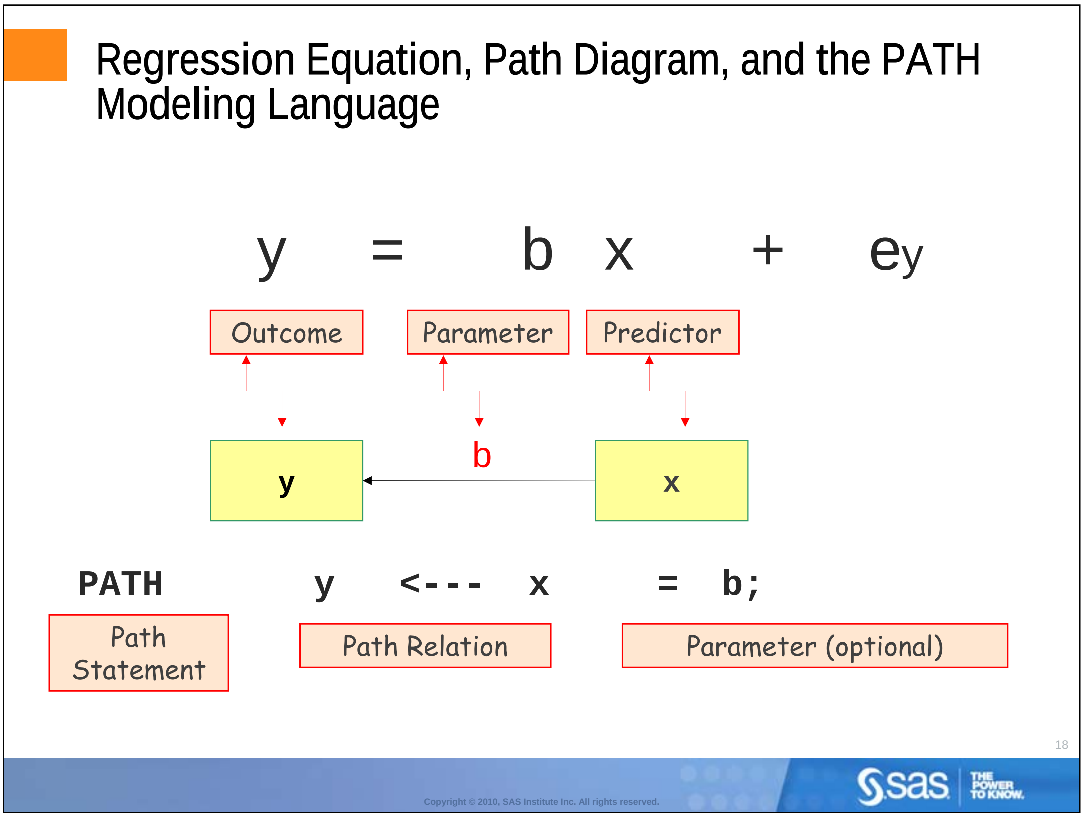
b

Given a base survival of 2.14 birds, every additional bird in Spring predicts a 0.65 bird surviving in August.

17

This table shows the essential results from PROC REG. The output shows an estimate of 0.65 for the regression coefficient b. The intercept estimate is 2.14. PROC REG also shows the standard error estimates and the t values for judging statistical significance. Both estimates are statistically significant.

An interpretation about these regression estimates is this: "Given a base survival of 2.14 birds, every additional bird in Spring predicts a 0.65 bird surviving in August (Fall)."



As shown previously, you can represent the linear regression model by the path diagram, which is also a representation scheme for SEM. Hence, regression models could be specified as SEM by using the path diagram.

Here is what you do to specify a simple linear regression model in PROC CALIS. You use the PATH statement to specify the path in the regression model. In this case, it is just Y<---X in the PATH statement. Optionally, you can denote the corresponding path coefficient parameter. For example, you can put "= b" at the back of the path to denote the parameter label or name.

Regression Model Specified by PROC CALIS

```
proc calis data=hens;
   path
      y <--- x;
run;
```


This is the entire PROC CALIS syntax for the simple linear regression model. Isn't that easy and simple?

Results from PROC CALIS for the Pheasant Data

PATH List

--------Path--------			Parameter	Estimate	Standard Error	t Value
y	<---	x	_Parm1	0.64941	0.07974	8.14445

The same estimate of b by PROC REG

Variance Parameters

Default variance parameters

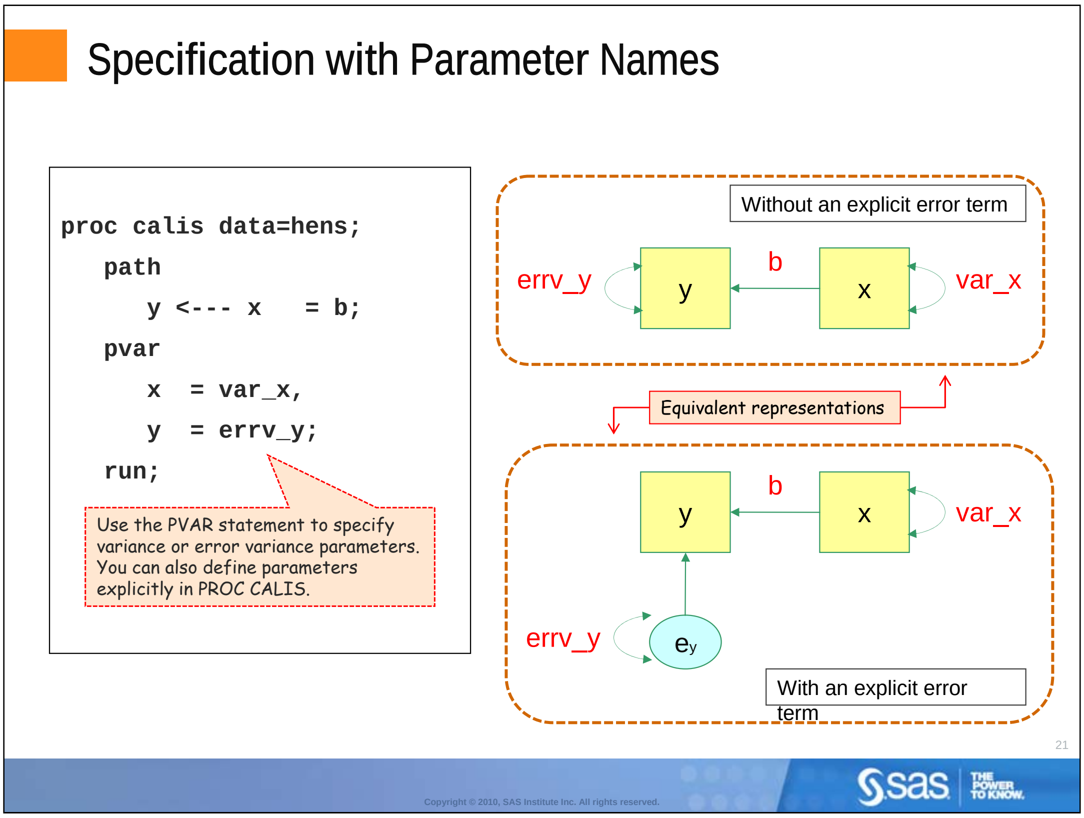
Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	x	_Add1	3.62124	1.36870	2.64575
Error	y	_Add2	0.32233	0.12183	2.64575

20

This slide shows the results from PROC CALIS.

The estimated effect of x on y, denoted as y <--- x in the output, is 0.65, which is the same as that in the PROC REG results. Because you did not name this regression coefficient parameter (but you specify the path nonetheless), PROC CALIS generates a unique parameter name called _Parm1 for it. The standard error estimate and the t value are a little bit different from that of the PROC REG results. This is because different degrees of freedom for computing the standard errors are used in the two approaches.

In PROC CALIS, it also includes results for two more parameters in the SEM. The variance of x and the error variance of y are treated as model parameters. Their estimates are also shown in the PROC CALIS results. Note that PROC CALIS creates default parameter names for these default variances even though you did not specify them. In this example, these variance parameters are named "_Add1" and "_Add2", respectively.



You could name all the parameters in PROC CALIS by putting your preferred names.

In the path diagram at the top right corner, the parameters are shown in red. In the regression model, b is the regression coefficient, var_x is the variance for the predictor variable x, and errv_y is the error variance of y. This path diagram representation is equivalent to the one shown at the bottom right corner, where an explicit error term is attached to Y. The error term is represented by an oval shape because it is treated as a latent variable. This representation has the same set of parameters, only that errv_y is now attached to the error variable directly.

You can specify these parameters explicitly in PROC CALIS. In the left panel of the slide, the parameter b is specified after the y <--- x path, separated by an equal sign. To specify the variances or error variances in the model, you can use the PVAR statement. For example, “x = var_x” means that the variance of x is a parameter called “var_x”.

Notice that naming parameters is entirely optional. For this example, naming parameters appears to serve only as an illustration. Later in this talk, you will find situations where the use of parameter names is not only useful, but also necessary.

Hen Pheasants Results with Parameter Names Specified

PATH List

--------Path--------			Parameter	Estimate	Standard Error	t Value
y	<---	x	b	0.64941	0.07974	8.14445

Parameter name specified

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	x	var_x	3.62124	1.36870	2.64575
Error	y	errv_y	0.32233	0.12183	2.64575

Parameter names specified

22

As shown in this slide, the numerical results from PROC CALIS with explicit parameter names specified are the same as those without using parameter names. The only difference is that now you can use these parameter names to locate the corresponding results directly.

Keys to the PATH Modeling Language

- As easy as drawing a path diagram
- PATH statement specifies the functional relationships – required specification
- PROC CALIS sets variances and error variances by default – optional specification (most of the time)
- Naming free parameters is optional

23

So far, I have shown you that:

1.The PATH modeling language is as easy as drawing a path diagram.
2.You can use the PATH statement to specify the paths in path diagram, with or without specify the parameter names for the path coefficients.
3.You can also specify the variance or error variance parameters explicitly. In most practical applications, variances and error variances have already been set by default and you do not need to worry about specifying them. The essential part of SEM is specified in the PATH statement.
4.Naming parameters is optional in PROC CALIS.

Measurement Errors in Predictors

- Bird counting might involve measurement errors in x
- $x = f_x + e_x$
- f_x : true score, but not observed
- x : observed, but with measurement error e_x

24

Let us make a little step forward to show a special SEM feature that linear regression cannot handle easily.

In the bird counting example, we did not take into account that bird counting could involve measurement errors. In the current context, the measurement error in bird counting could be due to the environment factors in the forest: obstruction from the tree branches, “biased” angles from the bird observers, and etc.

Mathematically, you can hypothesize a variable called fx to represent the “true” counts obtained from the bird observed. The observed number of birds x is the sum of fx, the true score, and ex, an error term.

What you got from the data is x, the observed fallible score. However, ideally, you would want to use fx, the true score in your regression analysis.

A Measurement Error Model for the Pheasant Data

- Structural Equation

  $y = b\ f_x + e_y$

- Measurement equation

  $x = f_x + e_x$

- Can you estimate b?
- Problem: The measurement equation introduces an additional parameter: Var(e_x) (variance of e_x or error variance of x)

The preceding idea is formalized as the following SEM with a latent variable fx.

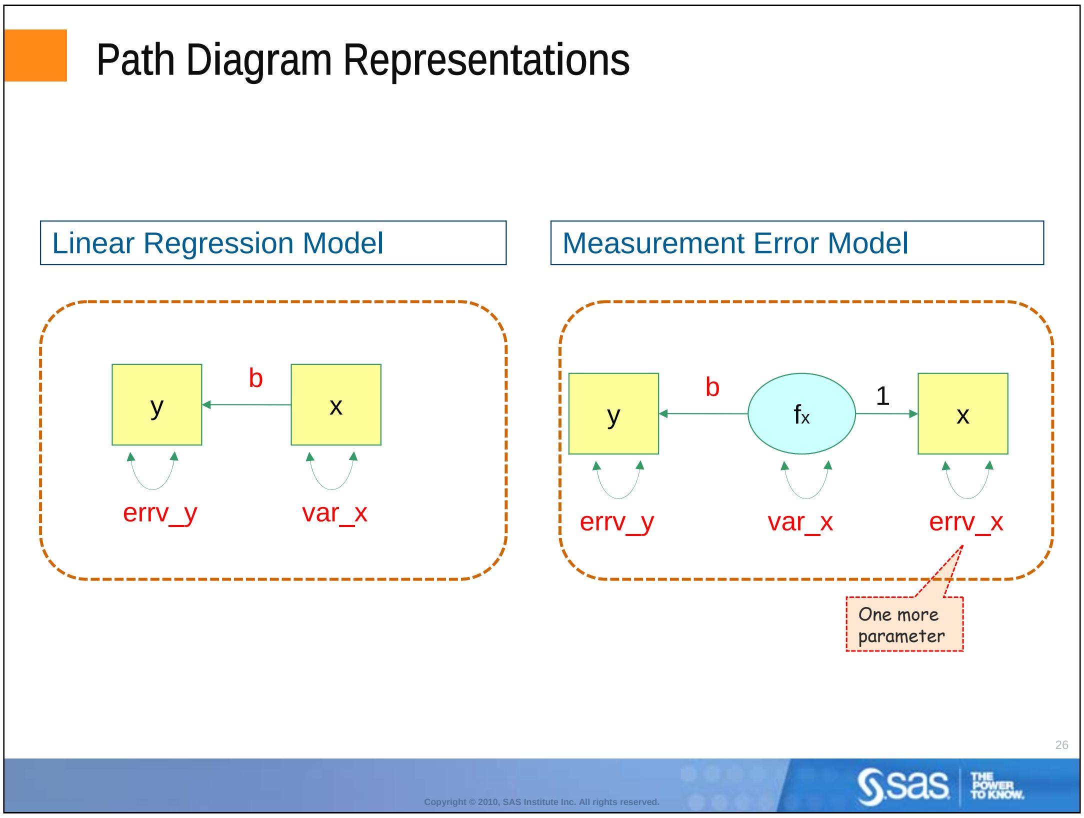
In the so-called structural model, y is predicted from fx, the true score, in the linear regression model. This so-called structural equation takes the role of the original linear regression equation---only now you are supposed to have a better model by using the measurement error-free fx as the predictor.

In the so-called measurement model, you hypothesize that the observed variable x is obtained as the sum of fx and an measurement error term e_x.

Can you estimate b with the latent variable fx in the structural equation?

This answer is yes.

But the technical problem encountered here is that the measurement equation introduces one additional parameters in var(ex)---error variance of x. This problem will make the SEM unidentified. In a very loose sense, this means that your model estimates more parameters than would be allowed by the given information of the data set. Consequently, the parameters in the model are not estimable. I will describe a method to deal with this identification problem later.



This slide compares the linear regression model with the measurement error model by the use of path diagram. It demonstrates why the measurement error model has one more parameter to estimate.

In the left panel, the path diagram for the simple linear regression analysis is shown.

In the right panel for the measurement error model, we still have x and y as the observed variables. But now we have a latent variable fx that takes the role of the predictor of y. Var_x in this model now represents the true variance of the predictor fx. The new parameter in the measurement error model is errv_x (error variance of x). With this additional parameter, we need to make additional assumption to estimate the model parameters.

Constraining the Error Variances

- Bird counting is more accurate in fall (y) than in spring (x)
- In an independent study, error variance (for x) in spring is six times as much as that (for y) in fall
- Fuller’s recommendation: Var(e_x) = 6 Var(e_y)

errv_x = 6* errv_y

Fortunately, we have a reasonable assumption about the relative size of the error variances in the model.

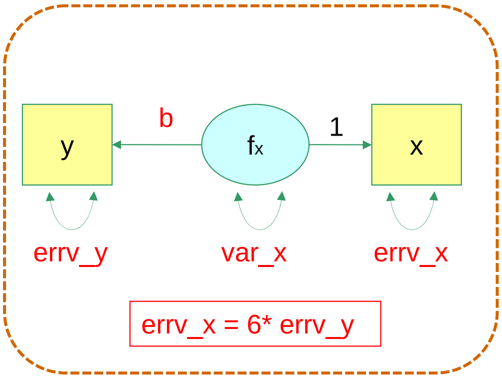
This assumption is based on the fact that bird counting in Fall is more accurate than that in spring. The reason is that the fallen leaves in Fall makes the counting of birds less obstructive.

The assumption we are going to make is based on an independent study about the relative error variances in x and in y. In Fuller’s book, the ratio of these variances is about 6. Mathematically, Var(ex) = 6*Var(ey). That is, error variance for x is six times as much as the error variance of y. Or, in the PROC CALIS specification, you want to state the following parametric constraint in the modeling: errv_x=6*errv_y.

A Measurement Error Model with a Constraint for the Pheasant Data

```
proc calis data=hens;
   path
      y  <--- fx   = b,
      fx ---> x    = 1;
   pvar
      y  = errv_y,
     fx  = var_x,
      x  = errv_x;
   errv_x = 6 * errv_y;
run;
```

The required constraint is specified as a SAS programming statement.



28

It turns out that it is pretty straightforward to specify this parametric constraint in PROC CALIS. You just simply add one more line of code to represent this relationship, as shown in the SAS code of the slide. In the SAS literature, this line of code is called a SAS programming statement, which is used extensively in the DATA step of SAS. You can use as many SAS programming statements as you want to describe the relationships of the parameters in the model.

Measurement Error Model Results for the Pheasant Data

PATH List

--------Path--------			Parameter	Estimate	Standard Error	t Value
y	<---	fx	b	0.75158	0.09228	8.14427
fx	--->	x		1.00000		

A larger estimated effect than the one estimated without taking the measurement error into account (0.649)

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	y	errv_y	0.08205	0.03101	2.64575
Exogenous	fx	var_x	3.12893	1.36180	2.29765
Error	x	errv_x	0.49231	0.18608	2.64575

Six times as much as the estimate for errv_y

29

After you take the measurement error into account, the regression coefficient b is now 0.75, which is a larger effect than 0.649, which you obtained from the linear regression model without taking the measurement error in x into account. Therefore, the previous regression analysis underestimated this effect because it failed to incorporate the measurement error into the model. However, with SEM, you can easily incorporate the measurement errors into the analysis.

Estimates of the variances and error variances are shown in the next table. You can see that the constraint specified in the PROC CALIS is honored in the estimation. The error variance estimate of x is 0.49, which is six times as much as the error variance estimate of y, which is 0.08.

Some Features of the PATH Modeling Language

- Specifying paths in the PATH statement is straightforward
- Deals with latent variables easily – variables are latent if they are not present in the data set
- PVAR statement for specifying variances and error variances
- PCOV statement for specifying covariances and error covariances (to be shown)
- Parameter dependency can be specified by the SAS programming statements. For example,

```
parm1 = 4 * parm2 + exp(parm4) ** parm6;
```


This slide summarizes some features of the PATH modeling language.

1.It is as straightforward as drawing the paths.
2.It can deal with latent variables easily.
3.You can use the PVAR statement to specify variances or error variances (double-headed arrows attached to individual variables in the path diagram).
4.You can use the PCOV statement to specify covariances or error covariances (double-headed arrows attached to pairs of variables in the path diagram).
5.You can specify parameter dependency by using the SAS programming statements directly. Indeed, even very strange and complicated (continuous) parametric functions are supported in PROC CALIS.

A Confirmatory Factor Model

31

We now move on to a more complicated type of structural equation models called confirmatory factor models.

Political Democracy Data

- Bollen (1989) Chapter 7
- Two latent factors: political democracy in 75 developing countries in 1960 and 1965
- Four indicator measures for the latent factors in each year:
  - Freedom of press (Press60, Press65)
  - Freedom of group oppositions (Freop60, Freop65)
  - Fairness of elections (Fair60, Fair65)
  - Elective nature of the legislative body (Legis60, Legis65)
- Purpose of the confirmatory factor analysis: Validate the measurement indicators

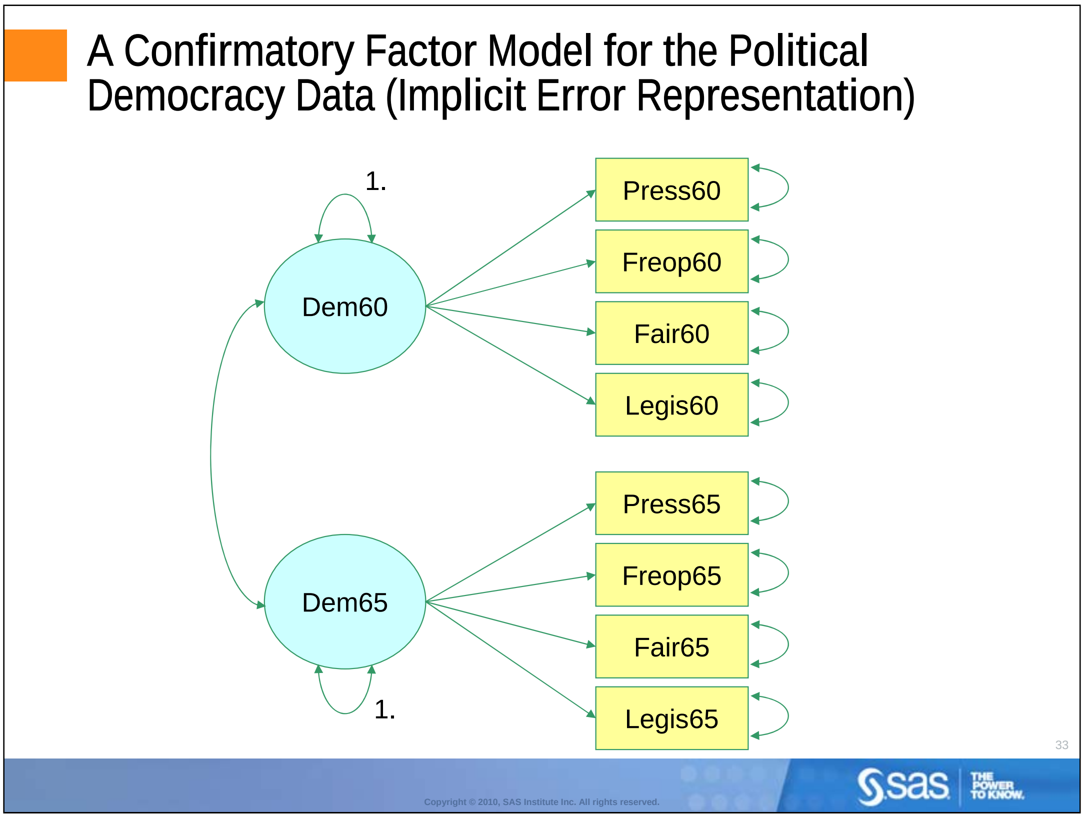
32

This example is based on an example in Chapter 7 of Bollen's classic textbook: Structural Equation Modeling.

In this example, two latent factors for measuring political democracy in 75 developing countries in 1960 and 1965 were hypothesized.

These two latent factors are not observed, but they have some related observed variables that serve as indicators. In each year, you measure four variables to gauge the political democracy: freedom of press, freedom of group oppositions, fairness of elections, and elective nature of the legislative body.

The purpose of the confirmatory factor analysis is to validate these measurement indicators statistically.



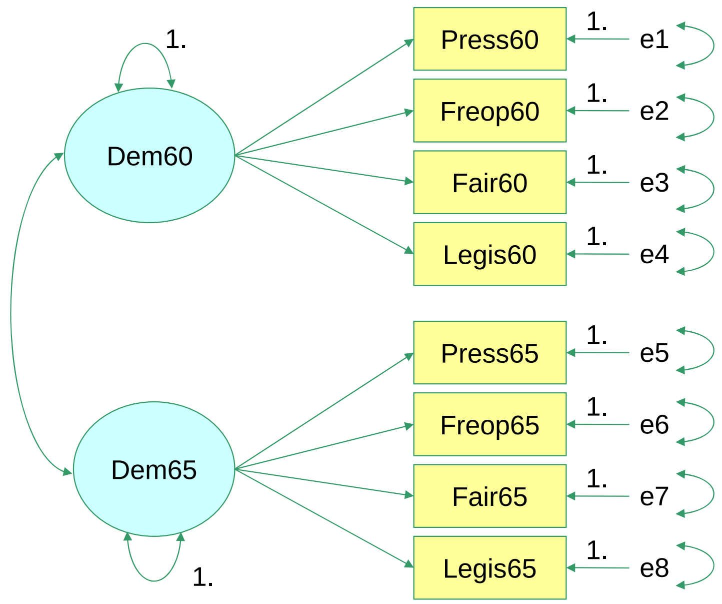
This path diagram shows the hypothesized confirmatory factor model.

In the path diagram, two latent factors are represented by two ovals. Dem60 is the political democracy in 1960 and Dem65 is the political democracy in 1965. They are linked to the respective measured variables, as shown in the path diagram. These single-headed paths represent the typical factor-observed variable relationships.

The double-headed arrow that connects Dem60 and Dem65 represents the covariance parameter between the two factors. It means that the two factors are correlated. Double-headed arrows that are attached to Dem60 and Dem65 individually represent the variance parameters of the two factors. In the model, you fix these variances to 1 so that the scales of the factors are identified. This is conventionally done because the scale of latent factors is arbitrary (you do not measure latent variables directly so that they could be defined on any unit of measurement).

The double-headed arrows that are attached to the observed variables represent the error variances. They signify the fact that the factors in the model do not account for 100% of the variances of the observed variables. The error variances are the unique part of the variances in the variables that are not due to their relationships with the factors in the model.

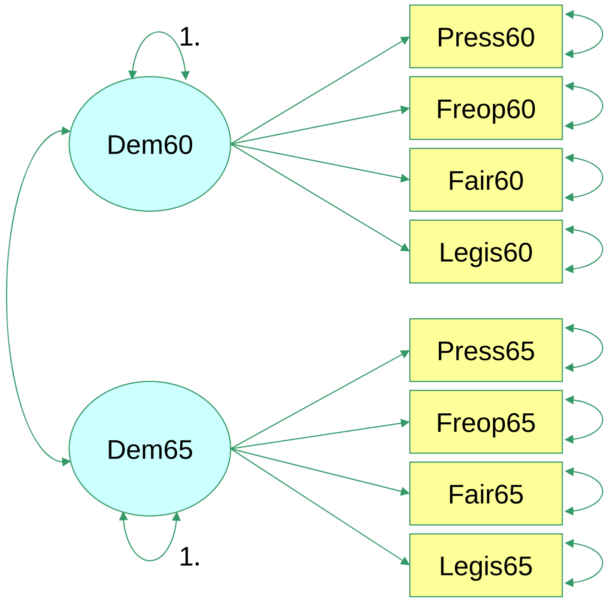
A Confirmatory Factor Model for the Political Democracy Data (with Error Terms)



This is an alternative path diagram representation with the use of explicit error terms. Notice that the double-headed arrows for the observed variables now shift to the error terms. This path diagram representation is shown here only for illustration purposes. In this workshop, I rely on the path diagram representation that does not use explicit error terms.

Basic Confirmatory Factor Model for the Political Democracy Data

```
proc calis data=polidem;
path
   Dem60 ---> Press60,
   Dem60 ---> Freop60,
   Dem60 ---> Fair60,
   Dem60 ---> Legis60,
   Dem65 ---> Press65,
   Dem65 ---> Freop65,
   Dem65 ---> Fair65,
   Dem65 ---> Legis65;
pvar
   Dem60 = 1, Dem65 = 1,
   Press60 Freop60 Fair60
   Legis60 Press65 Freop60
   Fair65 Legis65;
pcov
   Dem60 Dem65;
run;
```

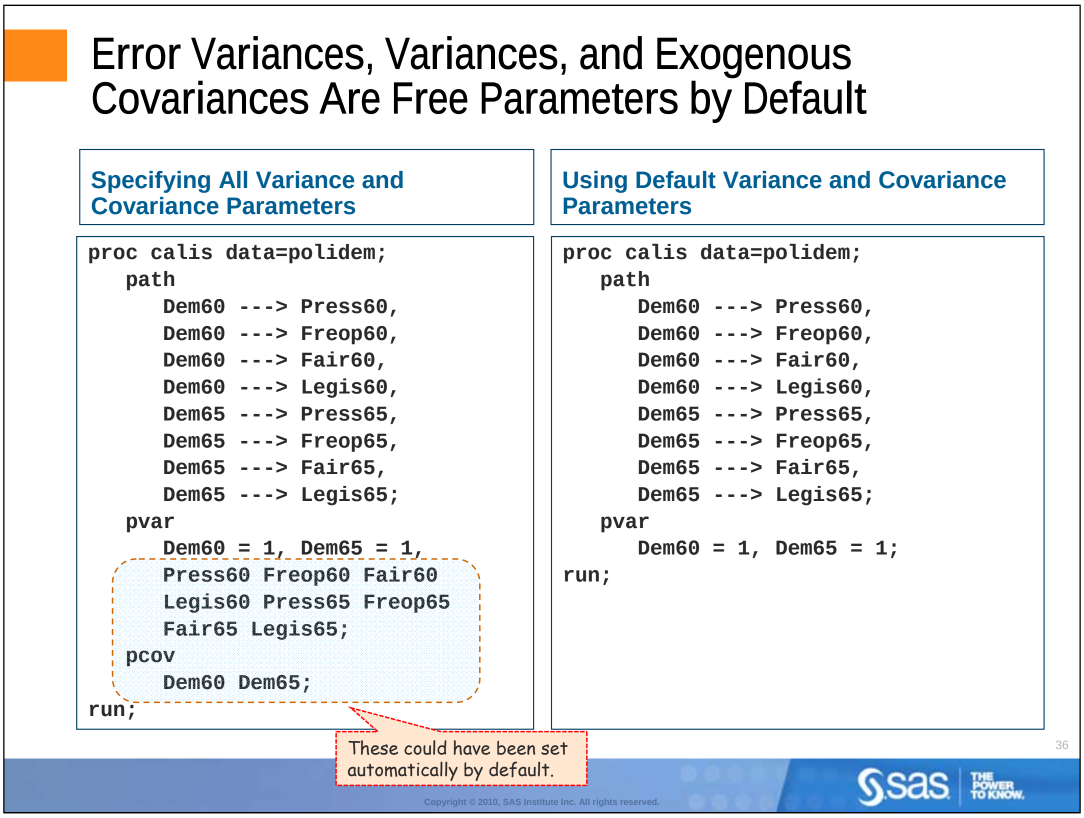


Specifying the CFA model is not much harder than the previous measurement error model. Basically, you only need to specify more paths for the CFA model.

In the PATH statement, you specify all the single-headed paths (arrows) in the path diagram.

In the PVAR statement, you specify all double-headed arrows that are attached to individual variables. PVAR actually stands for partial variance---you can specify the variances and error variances in this statement. "Dem60 =1" means that the variance of Dem60 is fixed to one. Similarly for "Dem65=1". The eight observed variable are specified in the PVAR statement to signify that their error variances are free parameters in the model.

In the PCOV statement, you specify pairs of variables that have covariances or error covariances as parameters in the model. In the current path diagram, Dem60 and Dem65 are correlated.



To make model specification more efficient and error-free, PROC CALIS employs default free parameters in the model. These default free parameters are set because they are commonly employed in practice.

For example, because predictions of outcome variables are usually not perfect, the error variances are free parameters by default. This means that all the PVAR specifications for the observed variables are not necessary because PROC CALIS would have treated them as free parameters by default.

Similarly, the variances of Dem60 and Dem65 and their covariance are default free parameters because they are assumed in most practical applications. In the current example, this means that the PCOV statement specification for the covariance between Dem60 and Dem65 is not necessary.

However, because the variances of Dem60 and Dem65 are fixed to 1 (for identification of the latent variable scales), they must be specified explicitly in the PVAR statement. Otherwise, these variances would have been free parameters by default.

Estimates of Path Coefficients (Loadings) for the Political Democracy Data

PATH List

----------Path----------			Parameter	Estimate	Standard Error	t Value
Dem60	--->	Press60	_Parm1	2.20567	0.25122	8.77998
Dem60	--->	Freop60	_Parm2	3.00132	0.39735	7.55335
Dem60	--->	Fair60	_Parm3	2.31033	0.34026	6.78989
Dem60	--->	Legis60	_Parm4	2.89483	0.31582	9.16619
Dem65	--->	Press65	_Parm5	2.04790	0.25930	7.89771
Dem65	--->	Freop65	_Parm6	2.68003	0.33258	8.05834
Dem65	--->	Fair65	_Parm7	2.70879	0.31804	8.51711
Dem65	--->	Legis65	_Parm8	2.76604	0.30830	8.97190

All path estimates are significant (t > 1.96).

37

This table shows the estimates of path coefficients from PROC CALIS.

In the factor analysis literature, these path coefficients are also called loadings. To validate the relationships between the democracy factors and the observed variables, the t-values must be examined for statistical significance. Using normal approximation, t values with their absolute values bigger than 1.96 are significantly different from zero.

In a typical factor-analysis study, you would want all these t-values to be significant in order to claim nonzero factor-variable relationships . An insignificant t-value means that the corresponding variable is not an indicator for the purported factor. Insignificant t-values for path coefficients would challenge the validity of your factor model.

Estimates of Variances for the Political Democracy Data

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	Dem60		1.00000		
	Dem65		1.00000		
Error	Press60	_Add1	2.01359	0.41048	4.90549
	Freop60	_Add2	6.57189	1.20964	5.43294
	Fair60	_Add3	5.42661	0.96546	5.62076
	Legis60	_Add4	2.83887	0.61417	4.62229
	Press65	_Add5	2.63180	0.49311	5.33709
	Freop65	_Add6	4.19276	0.79422	5.27906
	Fair65	_Add7	3.46180	0.68155	5.07928
	Legis65	_Add8	2.88292	0.59927	4.81068

All error variance estimates are significant ($t > 1.96$).

38

Estimates of variances and error variances are shown in this table. The variances of Dem60 and Dem65 are fixed to 1 and therefore there are no significance tests for these variances. All other error variance estimates are significantly larger than zeros. This also means that the factors do not account for all the variances of the observed variables. This is natural because deterministic relationships between factors and observed variables are rare.

Estimate of Covariance for the Political Democracy Data

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
Dem65	Dem60	_Add9	0.97528	0.02656	36.72321

High and significant correlation between the Democracy factors in 1960 and 1965.

39

This table shows the covariance between Dem60 and Dem65. This estimate is also the estimated correlation between the two latent factors because the variances of the factors are fixed to one. This correlation is extremely high, possibly because the political democracy status do not change much during those 5 years.

How Is the Model Fit?

Fit Summary

Modeling Info	N Observations	75
	N Variables	8
	N Moments	36
	N Parameters	17
	N Active Constraints	0
	Baseline Model Function Value	6.1482
	Baseline Model Chi-Square	454.9633
	Baseline Model Chi-Square DF	28
	Pr > Baseline Model Chi-Square	<.0001
Absolute Index	Fit Function	0.6009
	Chi-Square	44.4686
	Chi-Square DF	19
	Pr > Chi-Square	0.0008
	Z-Test of Wilson & Hilferty	3.1383
	Hoelter Critical N	51
	Root Mean Square Residual (RMSR)	0.5388
	Standardized RMSR (SRMSR)	0.0494
	Goodness of Fit Index (GFI)	0.8658
Parsimony Index	Adjusted GFI (AGFI)	0.7457
	Parsimonious GFI	0.5875
	RMSEA Estimate	0.1346
	RMSEA Lower 90% Confidence Limit	0.0833
	RMSEA Upper 90% Confidence Limit	0.1865
	Probability of Close Fit	0.0062
	ECVI Estimate	1.1240
	ECVI Lower 90% Confidence Limit	0.9065
	ECVI Upper 90% Confidence Limit	1.4608
	Akaike Information Criterion	78.4686
	Bozdogan CAIC	134.8659
	Schwarz Bayesian Criterion	117.8659
	McDonald Centrality	0.8438
Incremental Index	Bentler Comparative Fit Index	0.9403
	Bentler-Bonett NFI	0.9023
	Bentler-Bonett Non-normed Index	0.9121
	Bollen Normed Index Rho1	0.8560
	Bollen Non-normed Index Delta2	0.9416
	James et al. Parsimonious NFI	0.6122

A lot of fit indices, but researchers usually report just a few of them.

40

We have looked at the estimates and concluded that the relationships between the factors and the variables are strong and significant. Those results validated the individual factor-variable relationships.

To gain support for the overall confirmatory factor model, you would want to examine the model fit statistics. This table shows various fit indices computed by PROC CALIS. In the SEM field, a large number of fit indices have been proposed. There is no consensus as to which indices are best to report in the research. But researchers tend to report some of the most popular ones in their respective fields.

Because a large number of indices might be confusing, PROC CALIS provides a way to customize this fit summary table.

Using the FITINDEX Statement to Customize the Fit Summary Output

```
proc calis data=polidem;
   path
      Dem60 ---> Press60,
      Dem60 ---> Freop60,
      Dem60 ---> Fair60,
      Dem60 ---> Legis60,
      Dem65 ---> Press65,
      Dem65 ---> Freop65,
      Dem65 ---> Fair65,
      Dem65 ---> Legis65;
   pvar
      Dem60 = 1, Dem65 = 1;
   fitindex on(only) = [chisq df probchi rmsea cn srmsr
            bentlercfi agfi] noindextype;
run;
```

ON(ONLY)= selects the set of fit indices to display.
NOINDEXTYPE suppresses the printing of index types.

41

You can use the FITINDEX statement to customize your fit summary table.

Use the ON(ONLY)= option to select your "favorite" fit indices.

Use the NOINDEXTYPE option to suppress the printing of the fit index types.

Customized Fit Summary Table

Fit Summary	
Chi-Square	44.4686
Chi-Square DF	19
Pr > Chi-Square	0.0008
Hoelter Critical N	51
Standardized RMSR (SRMSR)	0.0494
Adjusted GFI (AGFI)	0.7457
RMSEA Estimate	0.1346
Bentler Comparative Fit Index	0.9403

"Good" SRMSR and Bentler's CFI. "Bad" chi-square, AGFI, RMSEA.

42

This is the customized fit summary output by using the previous FITINDEX statement specifications.

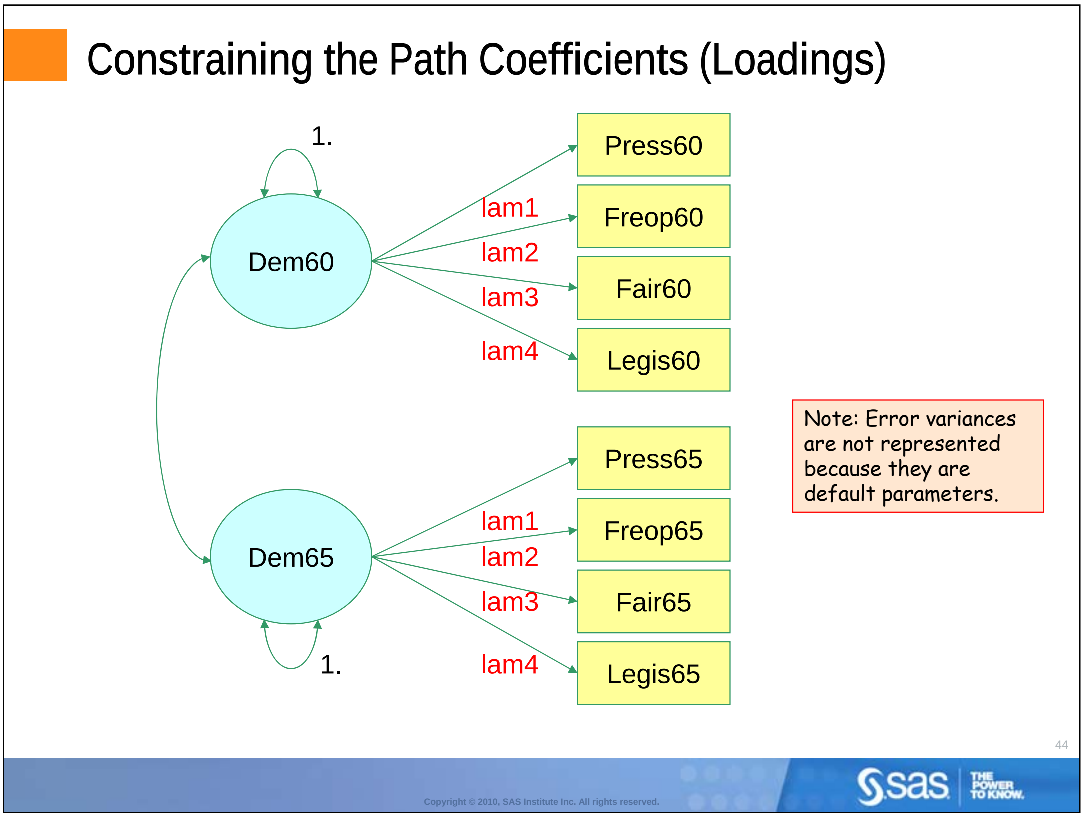
This table contains the more popular fit indices reported in research (as recognized by the author).

In practice, the model fit chi-square model statistic, its df, and the corresponding p-value are routinely reported even though very few researchers in the field would use the model fit chi-square alone to judge model fit. As shown in this table, the p-value is very small so that statistically it means that the hypothesized model should be rejected. However, it is a known issue in SEM that even very useful SEM models with minimum departures from the data would be rejected statistically. Therefore, researchers in the SEM field tend to focus more on other fit indices to judge model fit.

The SRMSR, AGFI, RMSEA, and CFI are four of the most popular fit indices in the SEM field. See the glossary page for the descriptions of these fit indices. For the SRMSR and RMSEA, the smaller the values the better the fit. Usually, values under 0.05 indicate good model fit. Therefore, the SRMSR says that the current model is good, but the RMSEA says that the current model is bad. For the AGFI and Bentler's CFI, the larger the values the better the model fit. Therefore, the AGFI says that the current model is bad, but the CFI says that it is good. Because these indices do not consistently indicate a good model fit, it is safe to say that the current CFA model is promising, but it needs further confirmation.

A Confirmatory Factor Model with Loading Constraints

43



In addition to fitting a basic confirmatory factor model, PROC CALIS enables you to set up parameter constraints easily. The main tool is to use parameter names in the specification.

For the political democracy example, the researcher wants to constraint the factor loadings (path coefficients) across time. The theoretical reason is that basically the measured variables are the same in the two years. In the path diagram, you can represent equality constraints by putting the same parameter names or labels to the pairs of the related paths. For example, lam1 is the loading of Press60 on Dem60. It is also the loading of Press65 on Dem65. Similarly, you can set the other 3 sets of constraints in the path diagram.

Fitting a CFA Model with Constraints on the Loadings

```
proc calis data=polidem;
   path
      Dem60 ---> Press60   = lam1,
      Dem60 ---> Freop60   = lam2,
      Dem60 ---> Fair60    = lam3,
      Dem60 ---> Legis60   = lam4,
      Dem65 ---> Press65   = lam1,
      Dem65 ---> Freop65   = lam2,
      Dem65 ---> Fair65    = lam3,
      Dem65 ---> Legis65   = lam4;
   pvar
      Dem60 = 1, Dem65 = 1;
   fitindex on(only) = [chisq df probchi rmsea cn srmsr
            bentlercfi agfi] noindextype;
run;
```

These constrain the path coefficients.

45

In the PATH modeling language, the constraints could be handled similarly. The code shown in this slide is modified from the previous code by adding the parameter names in the paths. The syntax is to add an equal sign and then the parameter names after the path specifications in the PATH statement. With the same parameter names for the pairs of the related paths, the estimates would be exactly the same.

Fit Summary Table for the Political Democracy Data with Loading Constraints

Fit Summary	
Chi-Square	46.8893
Chi-Square DF	23
Pr > Chi-Square	0.0023
Hoelter Critical N	56
Standardized RMSR (SRMSR)	0.0714
Adjusted GFI (AGFI)	0.7844
RMSEA Estimate	0.1185
Bentler Comparative Fit Index	0.9440

Only Bentler's CFI indicates a good model fit.

46

This table shows the fit summary of the model with the loading constraints. Because of the constraints, this model does not fit as well as the previous model. The SRMSR is larger than 0.05. The AGFI is much smaller than 0.9. The RMSEA is much larger than 0.05. All these show a bad model fit. However, Bentler's CFI (0.94) shows a good model fit.

Estimates of the Constrained Loadings for the Political Democracy Data

PATH List

----------Path----------			Parameter	Estimate	Standard Error	t Value
Dem60	--->	Press60	lam1	2.13970	0.21716	9.85319
Dem60	--->	Freop60	lam2	2.80116	0.29976	9.34473
Dem60	--->	Fair60	lam3	2.54987	0.27316	9.33459
Dem60	--->	Legis60	lam4	2.82969	0.27285	10.37075
Dem65	--->	Press65	lam1	2.13970	0.21716	9.85319
Dem65	--->	Freop65	lam2	2.80116	0.29976	9.34473
Dem65	--->	Fair65	lam3	2.54987	0.27316	9.33459
Dem65	--->	Legis65	lam4	2.82969	0.27285	10.37075

All path coefficients are significant.

As required from the model, paths with the same loading parameter have the same estimates. For example, both Dem60--->Press60 and Dem65--->Press65 have a loading estimate of 2.14 (lam1). All loading estimates, again, are statistically significant. This shows that all the purported factor-variable relationships are supported.

Estimates of Variances and Covariances for the Political Democracy Data with Loading Constraints

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	Dem60		1.00000		
	Dem65		1.00000		
Error	Press60	_Add1	2.01017	0.40312	4.98647
	Freop60	_Add2	6.72037	1.21196	5.54503
	Legis60	_Add3	2.88468	0.60956	4.73243
	Press65	_Add4	2.61966	0.49456	5.29699
	Freop65	_Add5	4.16958	0.79818	5.22383
	Legis65	_Add6	2.85029	0.59556	4.78593
	Fair60	_Add7	5.40824	0.97833	5.52803
	Fair65	_Add8	3.55382	0.67700	5.24935

Covariances Among Exogenous Variables

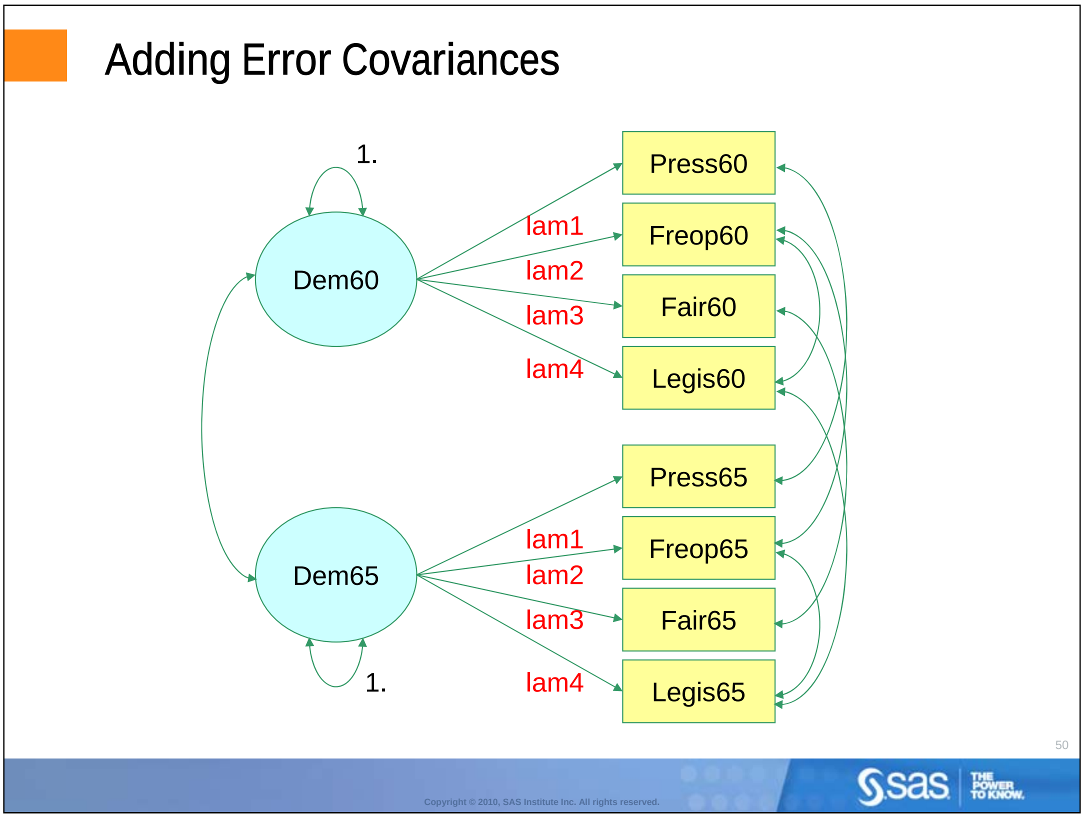
Var1	Var2	Parameter	Estimate	Standard Error	t Value
Dem65	Dem60	_Add9	0.97480	0.02682	36.34662

48

All the error variance estimates are also significant. The correlation between Dem60 and Dem60 is very high and significant.

A Confirmatory Factor Model with Correlated Errors

49



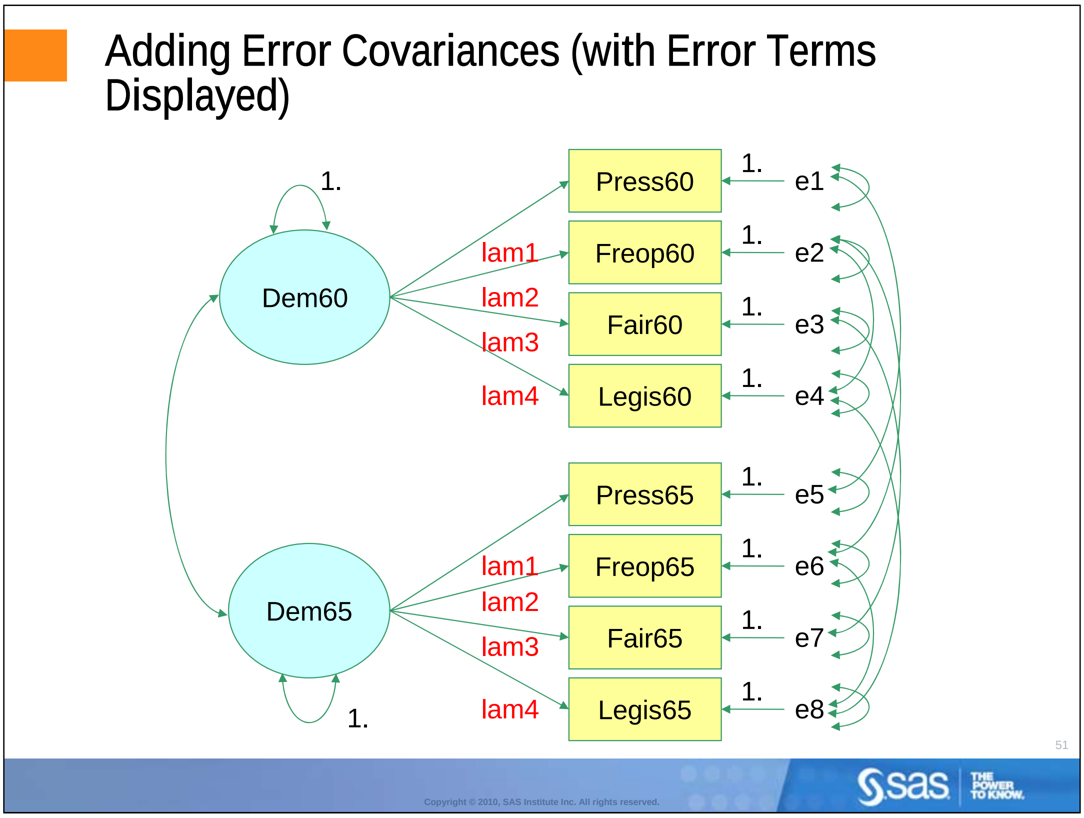
With the loading constraints, you observed a worse model fit.

The four equality constraints on the loadings you basically reduce the number of model parameters by 4. This naturally leads to a worse model fit than if you would allow all the loadings to be freely estimated.

Now you consider an opposite direction. Instead of reducing parameters by putting equality constraints, you want to add more parameters to the model. Adding more parameters to your model would improve the model fit. But the drawback of adding more parameters is that it makes your model more complicated, which is usually judged as an undesirable property for a scientific theory. It does not mean that you cannot add parameters. It only means that you should add only those parameters that could be justified by theoretical or substantive reasons.

In this example, it has been argued that freedom of group opposition and the elective nature of the legislative body have a part of their correlation that is beyond their common latent factors could explain (see Bollen). In SEM, this “extra” correlation is conceptualized as a correlation (or covariance) between the errors of the two variables. In the path diagram, this error covariance is represented by a double-headed arrow connecting the two variables. That is, Freop60 and Legis60 are connected by a double-headed arrow in 1960. By the same argument, Freop65 and Legis65 are also connected by a double-headed arrow to represent error covariance.

In addition, it is argued in Bollen that each of the variable pairs that were of the same nature but were measured at different times have a part of correlation that is beyond their common latent factors could explain. For example, Press60 and Press65 are connected by a double-headed arrow to represent their error covariance, which explains the part of the covariance between the two variables that is beyond the explanation by the covariance between Dem60 and Dem65. Similarly, the Freop-, Fair-, and Legis- pairs are all connected by double-headed arrows to represent error covariances.



The path diagram in this slide is equivalent to the previous representation that does not use explicit error variables.

In this path diagram, error terms for the measured variables are shown. The double-headed arrows are shifted to the error terms. This makes it obvious that those double-headed arrows are covariances between the error variables (but not as partial covariances between the observed variables, as shown in the previous slide).

Therefore, this path diagram representation is conceptually clearer about what are really being correlated in the model. However, the addition of the error terms makes the path diagram more cluttered. In this workshop, I would stick with the path diagram representation that does not use explicit error terms.

Fitting a CFA Model with Loading Constraints and Correlated Errors

```
proc calis data=polidem;
   path
      Dem60 ---> Press60   = lam1,
      Dem60 ---> Freop60   = lam2,
      Dem60 ---> Fair60    = lam3,
      Dem60 ---> Legis60   = lam4,
      Dem65 ---> Press65   = lam1,
      Dem65 ---> Freop65   = lam2,
      Dem65 ---> Fair65    = lam3,
      Dem65 ---> Legis65   = lam4;
   pvar
      Dem60 = 1, Dem65 = 1;
   pcov
      Freop60 Legis60, Freop65 Legis65,
      Press60 Press65, Freop60 Freop65,
      Fair60 Fair65, Legis60 Legis65;
   fitindex on(only) = [chisq df probchi rmsea cn srmsr
         bentlercfi agfi] noindextype;
run;
```

Use the PCOV statement to specify error covariances.

52

With the six additional pairs of correlated errors, you have six more error covariance parameters in the model.

In the PATH modeling language, you can specify these covariance parameters in the PCOV statement. In this example, this means that you enumerate the six pairs of measured variables in the PCOV statement. For example, the first pair is Freop60 and Legis60, which represent a covariance parameter between their error terms.

Fit Summary Table for the CFA Model with Loading Constraints and Correlated Errors

Fit Summary	
Chi-Square	15.1946
Chi-Square DF	17
Pr > Chi-Square	0.5815
Hoelter Critical N	135
Standardized RMSR (SRMSR)	0.0590
Adjusted GFI (AGFI)	0.9043
RMSEA Estimate	0.0000
Bentler Comparative Fit Index	1.0000

All indices indicate a good model fit.

This model is supposed to fit better because of the added parameters for the error covariances.

In fact, the model fit chi-square is not statistically significant. This supports the hypothesized model in the population.

All other fit indices show good or excellent fit. The SRMSR is 0.059, which is only slightly larger than the 0.05 criterion. The AGFI is 0.90, which is an indication of good model fit by convention. The RMSEA is essentially zero, which is the smallest RMSEA you could ever get. The CFI is 1, which is also the largest CFI you could ever get.

Estimates of the Loadings for the CFA Model with Constrained Loadings and Correlated Errors

PATH List

----------Path----------			Parameter	Estimate	Standard Error	t Value
Dem60	--->	Press60	lam1	2.16450	0.23009	9.40738
Dem60	--->	Freop60	lam2	2.61630	0.32500	8.05017
Dem60	--->	Fair60	lam3	2.61693	0.28700	9.11832
Dem60	--->	Legis60	lam4	2.75291	0.28312	9.72356
Dem65	--->	Press65	lam1	2.16450	0.23009	9.40738
Dem65	--->	Freop65	lam2	2.61630	0.32500	8.05017
Dem65	--->	Fair65	lam3	2.61693	0.28700	9.11832
Dem65	--->	Legis65	lam4	2.75291	0.28312	9.72356

All path coefficients are significant.

54

All loading (path coefficients) estimates are statistically significant, supporting the relationships between the latent factors and the measured variables.

Estimates of the Variances for the CFA Model with Constrained Loadings and Correlated Errors

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	Dem60		1.00000		
	Dem65		1.00000		
Error	Press60	_Add1	1.91664	0.43982	4.35780
	Freop60	_Add2	7.65544	1.39023	5.50661
	Legis60	_Add3	3.27028	0.73387	4.45621
	Press65	_Add4	2.52969	0.52882	4.78360
	Freop65	_Add5	4.87208	0.94384	5.16199
	Legis65	_Add6	3.25392	0.73319	4.43805
	Fair60	_Add7	5.03798	0.98299	5.12514
	Fair65	_Add8	3.32508	0.71220	4.66875

All error variance estimates are significant.

55

All error variance estimates are significant.

Estimates of the Covariances for the CFA Model with Constrained Loadings and Correlated Errors

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
Dem65	Dem60	_Add9	0.96603	0.02928	32.99044

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
Freop60	Legis60	_Parm1	1.42826	0.69666	2.05017
Freop65	Legis65	_Parm2	1.26677	0.59365	2.13389
Press60	Press65	_Parm3	0.58548	0.37178	1.57478
Freop60	Freop65	_Parm4	2.09624	0.74763	2.80386
Fair60	Fair65	_Parm5	0.74805	0.62336	1.20003
Legis60	Legis65	_Parm6	0.47686	0.46214	1.03186

Bad news: Some error covariance estimates are not significant.

56

The first table shows the correlation between the two latent factors. Again, the correlation is very high and significant.

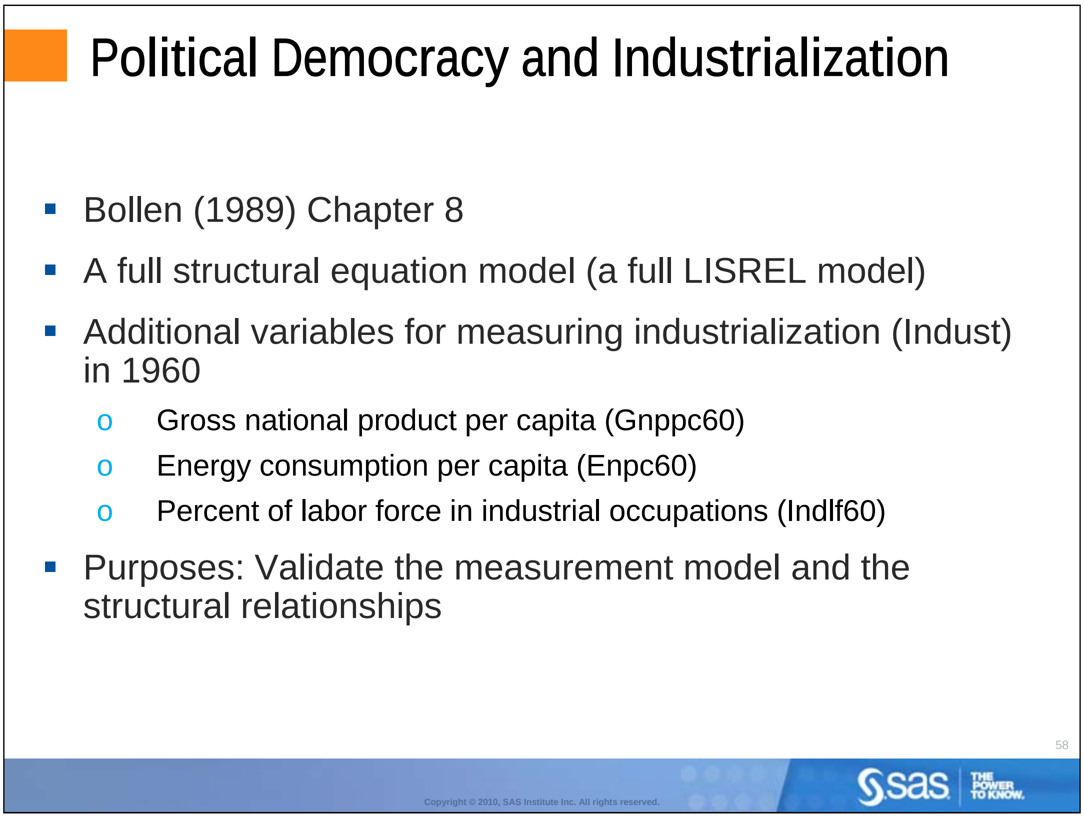
The second table shows the estimates for the newly added covariances between errors. Three of these covariances are significant, while the others are not. For example, Freop60 and Legis60, Freop65 and Legis65, and Freop60 and Freop65 are three error covariances that have t values larger than 1.96. The other three pairs have insignificant t-values. This means that adding these three covariances might be somewhat undesirable because their estimates are actually not significantly different from zero, casting doubts about their presence in the model.

The lesson here is that even though adding error correlations (or covariances) might improve the model fit, you should not routinely add error covariances only to boost the model fit. Adding unjustified error covariances makes your model more complicated and harder to interpret, especially when some error variance estimates turn out to be insignificant.

Political Democracy and Industrialization:

A Full Structural Equation Model

57

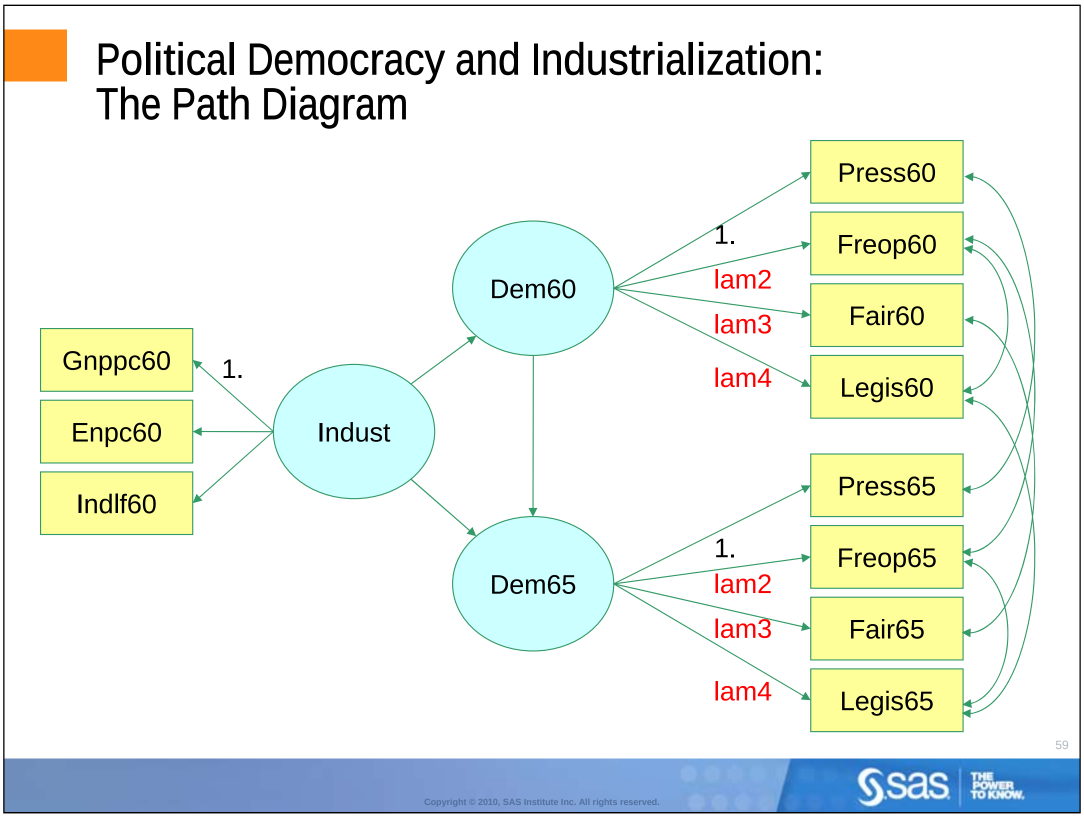


We continue with the previous model and add one more latent factor and its indicators into the model.

This example illustrates a full structural equation model (or a full LISREL) model. Essentially, this means that our focus is not only on validating the relationships between the latent factors and the measured variables (that is, the measurement model), but also on validating the functional relationships among latent variables (that is, the structural model).

For example, you have a latent factor called industrialization (Induct) that is supposed to be reflected by three observed variables: gross national product per capita (Gnppc60), energy consumption per capita (enpc60), and percent of labor force in industrial occupations (Indlf60). All these variables were measured in 1960.

The industrialization (Induct) latent variable serves as a predictor of the two democracy factors (Dem60 and Dem65). This kind of functional relationships between latent variables has not been explored in the confirmatory factor models discussed previously.



The entire SEM model is depicted in the path diagram of the current slide. The most notable addition is the paths from Indust to Dem60 and Dem65--- industrialization in 1960 serves as a predictor of democracy in both 1960 and 1965. Three observed variables serve as indicators of the industrialization: Gnppc60, Enpc60, and Indlf60.

There are two main modifications from the preceding confirmatory factor model.

First, instead of allowing Dem60 and Dem65 to freely covary in the CFA, the current model treats Dem60 as a predictor of Dem65.

Second, a different method for identifying the latent factor scales is used in the current model. In the preceding CFA model, variances of Dem60 and Dem65 are fixed to one. But because they become endogenous in the current model, you can no longer use this type of scale identification method. Instead, one of their observed indicator variables (that is, Press60 and Press65) now has a fixed path coefficient at one. Similarly, the path coefficient from Indust to Gnppc60 is fixed to one for scale identification.

Fitting the Structural Equation Model for the Political Democracy and Industrialization Data

Multiple-path specifications

```
proc calis data=polidem;
   path
      Dem60   ---> Press60 Freop60 Fair60 Legis60   =  1.   lam2   lam3   lam4,
      Dem65   ---> Press65 Freop65 Fair65 Legis65   =  1.   lam2   lam3   lam4,
      Indust  ---> Gnppc60 Enpc60 Indlf60           =  1.,
      Indust  ---> Dem60 Dem65,
      Dem60   ---> Dem65;
   pcov
      Freop60 Legis60, Freop65 Legis65,
      Press60 Press65, Freop60 Freop65,
      Fair60 Fair65, Legis60 Legis65;
   fitindex on(only) = [chisq df probchi rmsea cn srmsr
            bentlercfi agfi] noindextype;
run;
```

60

You can use PROC CALIS to specify this structural equation model easily.

In the PATH statement, I use a multiple-path specification syntax. In the first specification, Dem60 is a predictor of 4 outcome variables: Press60, Freop60, Fair60, and Legis60. This specifies four paths in a single path specification. After using an equal sign, I specify four parameters for the four paths. The first one is a fixed constant 1, which is applied to the Dem60 ---> Press60 path. The second one is a free parameter lam2, which is applied to the Dem60 ---> Freop60 path, and so on.

In the next 3 path specifications, I also use the multiple-path specification syntax. The second multiple-path syntax specifies that Dem65 is a factor with four indicators. The path coefficients (loadings) are also specified explicitly. The third multiple-path syntax specifies that Indust is a factor of three observed indicators, with a fixed one for the effect of Indust on Gnppc60. The path coefficients for the paths Indust--->Enpc60 and Indust--->Indlf60 are unnamed free parameters (with the empty specifications). The fourth multiple-path syntax specifies that Indust is a predictor of both Dem60 and Dem65. The corresponding path coefficients are (unnamed) free parameters in the model.

The last path in the PATH statement specifies Dem60 as a predictor of Dem65. Notice that no PVAR statement is used because fixing the Dem60 and Dem65 variances to one is not used in the current model. The scales of the latent factors are identified by fixing some path coefficients to 1.

Political Democracy and Industrialization: Fit Summary Table

Fit Summary	
Chi-Square	39.6438
Chi-Square DF	38
Pr > Chi-Square	0.3966
Hoelter Critical N	100
Standardized RMSR (SRMSR)	0.0558
Adjusted GFI (AGFI)	0.8606
RMSEA Estimate	0.0242
Bentler Comparative Fit Index	0.9975

Not a bad fit for the data.

61

The fit of the structural model is acceptable, if not exceptionally good.

The model fit chi-square is not significant, supporting the hypothesized model. The SRMSR is close to 0.05. The AGFI is 0.86, which shows a reasonable fit. The RMSEA indicates a very good model fit, as the value (.0242) is much lower than 0.05. The CFI is almost 1, which shows a perfect model fit.

Political Democracy and Industrialization: Estimates of Path Coefficients

PATH List

----------Path----------			Parameter	Estimate	Standard Error	t Value
Dem60	--->	Press60		1.00000		
Dem60	--->	Freop60	lam2	1.19079	0.14020	8.49336
Dem60	--->	Fair60	lam3	1.17454	0.12121	9.68988
Dem60	--->	Legis60	lam4	1.25099	0.11757	10.64006
Dem65	--->	Press65		1.00000		
Dem65	--->	Freop65	lam2	1.19079	0.14020	8.49336
Dem65	--->	Fair65	lam3	1.17454	0.12121	9.68988
Dem65	--->	Legis65	lam4	1.25099	0.11757	10.64006
Indust	--->	Gnppc60		1.00000		
Indust	--->	Enpc60	_Parm01	2.17966	0.13932	15.64530
Indust	--->	Indlf60	_Parm02	1.81821	0.15290	11.89126
Indust	--->	Dem60	_Parm03	1.47133	0.39496	3.72529
Indust	--->	Dem65	_Parm04	0.60046	0.22722	2.64267
Dem60	--->	Dem65	_Parm05	0.86504	0.07538	11.47648

62

All path coefficients are significant---a pretty good sign.

Political Democracy and Industrialization: Estimates of Variances

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous Error	Indust	_Add01	0.45466	0.08846	5.13991
	Press60	_Add02	1.87973	0.44229	4.25001
	Freop60	_Add03	7.68378	1.39404	5.51189
	Legis60	_Add04	3.26801	0.73807	4.42779
	Press65	_Add05	2.34432	0.48851	4.79895
	Freop65	_Add06	5.03534	0.93993	5.35716
	Legis65	_Add07	3.35236	0.71788	4.66983
	Gnppc60	_Add08	0.08249	0.01986	4.15376
	Enpc60	_Add09	0.12206	0.07105	1.71776
	Indlf60	_Add10	0.47297	0.09197	5.14268
	Fair60	_Add11	5.02270	0.97587	5.14691
	Fair65	_Add12	3.60813	0.72394	4.98402
	Dem60	_Add13	3.92767	0.88311	4.44753
	Dem65	_Add14	0.16668	0.23158	0.71975

63

Some error variances are not significant: Enpc60 and Dem65. Enpc60 is an indicator of the Industrialization in 1960. This insignificant error variance means that the Indust factor predict Enpc60 perfectly. However, the corresponding t-value is 1.71, which could be judged as marginally significant.

The error variance for Dem65 is also not significant, as evident by the non-significant t-value of 0.72. This means that given Indust and Dem60, Dem65 can be predicted almost perfectly.

Political Democracy and Industrialization: Estimates of Covariances

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
Freop60	Legis60	_Parm06	1.45956	0.70251	2.07764
Freop65	Legis65	_Parm07	1.39032	0.58859	2.36212
Press60	Press65	_Parm08	0.59042	0.36307	1.62619
Freop60	Freop65	_Parm09	2.21252	0.75242	2.94054
Fair60	Fair65	_Parm10	0.72123	0.62333	1.15706
Legis60	Legis65	_Parm11	0.36769	0.45324	0.81125

64

Again, there are some insignificant error covariances. This result challenges their presence in the model.

Political Democracy and Industrialization: Squared Multiple Correlations

Squared Multiple Correlations

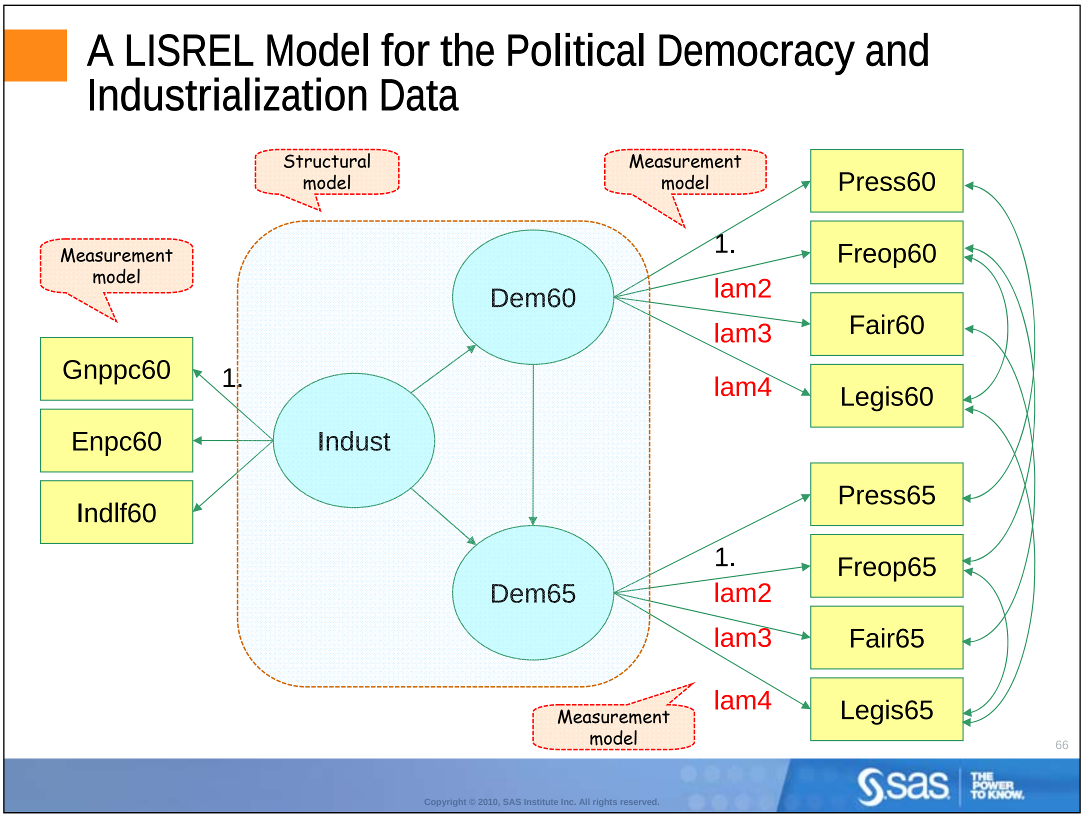
Variable	Error Variance	Total Variance	R-Square
Enpc60	0.12206	2.28211	0.9465
Fair60	5.02270	11.79895	0.5743
Fair65	3.60813	10.09361	0.6425
Freop60	7.68378	14.64875	0.4755
Freop65	5.03534	11.70144	0.5697
Gnppc60	0.08249	0.53715	0.8464
Indlf60	0.47297	1.97602	0.7606
Legis60	3.26801	10.95502	0.7017
Legis65	3.35236	10.70953	0.6870
Press60	1.87973	6.79166	0.7232
Press65	2.34432	7.04548	0.6673
Dem60	3.92767	4.91193	0.2004
Dem65	0.16668	4.70116	0.9645

65

The square multiple correlations are usually used to measure the percentage of overlapping variance between the predictors and the outcome variables. In the current example, R-squares range from 0.2 to extreme high values such as 0.95 and 0.96.

The smallest R-square is the one for predicting Dem60, which is 0.2. This actually is not that small an R-square value for social science data.

But the R-square (0.96) for Dem65 is extremely high. This means that Dem65 is almost perfectly predicted from democracy and industrialization in 1960.



This full SEM model is also a good illustration of the LISREL model.

The path diagram for the preceding model remains unchanged here. In order to call this path diagram a LISREL model, you have to identify the LISREL components in this path diagram. The two main components in LISREL are the measurement models and the structural model.

First, the measurement models are identified. A measurement model is about how observed variables are related to the latent variables or constructs in the model. Specifically, the measurement model involving industrialization is the measurement model of x because Indust serves as an exogenous (independent) variable in the path diagram. The measurement model involving Dem60 and Dem65 is the measurement model of y because Dem60 and Dem65 are endogenous (dependent) variables in the path diagram.

Second, the structural model is identified and highlighted in the center of the path diagram. The structural model describes the functional relationships among the latent variables (constructs) in the path diagram.

Therefore, all the essential component of the LISREL model is identified in the current path diagram.

Fitting the Structural Equation Model for the Political Democracy and Industrialization Data

```
proc calis data=polidem;
   path
     Dem60   ---> Press60 Freop60 Fair60 Legis60   =  1.   lam2   lam3   lam4,
     Dem65   ---> Press65 Freop65 Fair65 Legis65   =  1.   lam2   lam3   lam4,
     Indust  ---> Gnppc60 Enpc60 Indlf60           =  1.,
     Indust  ---> Dem60 Dem65,
     Dem60   ---> Dem65;
   pcov
     Freop60 Legis60, Freop65 Legis65,
     Press60 Press65, Freop60 Freop65,
     Fair60 Fair65, Legis60 Legis65;
   fitindex on(only) = [chisq df probchi rmsea cn srmsr
            bentlercfi agfi] noindextype;
run;
```

Measurement model: Relationships between latent and observed indicators

Structural model: Relationships among latent constructs

Measurement model

67

In the PATH modeling language, you can also identify the code for the measurement models and the structural model. The preceding code is recited here for illustrations.

In the PATH statement, the first three multiple-path specifications are concerned with the measurement of the latent constructs. In addition, all specifications in the PCOV statement are for the covariances of the measurement errors.

The last two specifications in the PATH statement are for the structural model. They describe the functional relationships between Indust, Dem60, and Dem65.

After identifying the LISREL components in the path diagram and in the SAS code, you now have a clue to specify the LISREL model in PROC CALIS. Essentially, it should be clear that the same path diagram is being used by the PATH modeling language and the LISREL model. The only task is to transcribe the code in the PATH modeling language to the language for the LISREL model.

A LISREL Model Specified by the LISMOD Modeling Language of PROC CALIS

```
proc calis data=polidem nose noparmname;
  lismod
     xvar = Gnppc60 Enpc60 Indlf60,
     yvar = Press60 Freop60 Fair60 Legis60 Press65 Freop65 Fair65 Legis65,
     xi   = Indust,
     eta  = Dem60 Dem65;
  matrix
     _LambdaY_ [ 1, @1] = 1.  lam2  lam3  lam4,    /* Paths from Dem60 and Dem65 to yvar */
               [ 5, @2] = 1.  lam2  lam3  lam4;
  matrix
     _ThetaY_  [ 4, 2], [ 8, 6],                   /* pcov statement in the path model */
               [ 5, 1], [ 6, 2], [ 7, 3], [ 8, 4];
  matrix
     _LambdaX_ [ 1, 1]  = 1.,                      /* Path from Indust to xvar*/
               [ 2 to 3, 1];
  matrix
     _Gamma_   [ 1 to 2, 1];
  matrix
     _Beta_    [2,1];
run;
```

NOSE: No standard errors
NOPARMNAME: No parameter names in the output

Measurement model for y

Measurement model for x

Structural model

PROC CALIS supports the so-called LISMOD modeling language. In order to fully understand the PROC CALIS code for the LISREL model, knowledge about matrix algebra is needed. But I will only describe the code in a conceptual way.

In the LISMOD statement, you first classify your variables into one of the four categories:

1.x-variables: observed indicators for the exogenous (independent) latent factors in the model.
2.y-variables: observed indicators for the endogenous (dependent) latent factors in the model.
3.xi-variables: exogenous (independent) latent factors in the model.
4.eta-variables: endogenous (dependent) latent factors in the model.

The LISREL model or LISREL program had been developed as a matrix-based language. Parameters in the models are specified as matrix elements in some specific model matrices with Greek names. PROC CALIS supports the matrix input of these LISREL model matrices. For example, in the measurement model for y, _LambdaY_ is the matrix that relates the y-variables to the eta-variables. Instead of specifying the paths as in the PATH statement, the MATRIX statement for _LambdaY_ serves the same purpose in the LISMOD modeling language. The MATRIX statement for _ThetaY_ specifies the error variances and covariances of the y-variables, much like the specifications of the PCOV statement in the PATH modeling language. In other words, the PATH model specifications are transcribed into the LISMOD model specifications for the y-variables.

Similarly, the MATRIX statement for _LAMBDAX_ specifies the parameters in the measurement model for the x-variables.

Finally, the structural relationships or the path relationships among the latent factors are specified in the MATRIX statements for the _GAMMA_ and _BETA_ matrices.

To simplify the output, I used two options in PROC CALIS statement. The NOSE option suppresses the printing of standard errors and the NOPARMNAME option suppresses the printing of the parameter names.

LISMOD Output for the Political Democracy and Industrialization Data (Measurement for y)

LAMBDAY Matrix

	Dem60	Dem65
Press60	1.0000	0
Freop60	1.1908	0
Fair60	1.1745	0
Legis60	1.2510	0
Press65	0	1.0000
Freop65	0	1.1908
Fair65	0	1.1745
Legis65	0	1.2510

Note: The NOSE and NOPARMNAME options suppress the printing of the standard error estimates and parameter names.

69

The following few slides show the output from PROC CALIS for the LISREL model. All the results are matrix-oriented. Details for these results have been discussed for the PATH model output and will not be repeated here. In general, you can find correspondence between the LISMOD and the PATH results.

This slide shows the measurement model for the y-variables.

LISMOD Output for the Political Democracy and Industrialization Data (Measurement for y)

THETAY Matrix

	Press60	Freop60	Fair60	Legis60	Press65	Freop65	Fair65	Legis65
Press60	1.8797	0	0	0	0.5904	0	0	0
Freop60	0	7.6838	0	1.4596	0	2.2125	0	0
Fair60	0	0	5.0227	0	0	0	0.7212	0
Legis60	0	1.4596	0	3.2680	0	0	0	0.3677
Press65	0.5904	0	0	0	2.3443	0	0	0
Freop65	0	2.2125	0	0	0	5.0353	0	1.3903
Fair65	0	0	0.7212	0	0	0	3.6081	0
Legis65	0	0	0	0.3677	0	1.3903	0	3.3524

Note: Error variances (diagonal elements) were set by default.

This slide shows the measurement error variances and covariances for the y-variables.

LISMOD Output for the Political Democracy and Industrialization Data (Measurement for x)

LAMBDAX Matrix

	Indust
Gnppc60	1.0000
Enpc60	2.1797
Indlf60	1.8182

THETAX Matrix

	Gnppc60	Enpc60	Indlf60
Gnppc60	0.0825	0	0
Enpc60	0	0.1221	0
Indlf60	0	0	0.4730

Note: Error variances (diagonal elements in _THETAX_) were set by default.

71

This slide shows the results of the measurement model for the x-variables, including the path coefficients and the error variances.

LISMOD Output for the Political Democracy and Industrialization Data (Structural Model)

BETA Matrix

	Dem60	Dem65
Dem60	0	0
Dem65	0.8650	0

GAMMA Matrix

	Indust
Dem60	1.4713
Dem65	0.6005

72

This slide shows the functional relationships between latent constructs.

LISMOD Output for the Political Democracy and Industrialization Data (Structural Covariances)

PSI Matrix

	Dem60	Dem65
Dem60	3.9277	0
Dem65	0	0.1667

PHI Matrix

	Indust
Indust	0.4547

Note: All variances (diagonal elements in _PSI_ and _PHI_) were set by default.

73

This slide shows the error variances of the eta-variable and the variance of the xi-variable.

Features of the LISMOD Modeling Language

- Supports the JKW (LISREL) models (not the LISREL program)
- Supports mean structure analysis
- Users input:
  - The ordered lists of x, y, ξ, and η variables
  - MATRIX statements to define free and fixed parameters
  - Names for parameters (not required for free parameters)
- Default covariance structure parameters of the LISMOD language:
  - Diagonal elements of all covariance matrices (all variances)
  - Lower triangular elements of the _PHI_ matrix (covariances of the ξ-variables)

74

In sum, the LISMOD modeling language in PROC CALIS supports the LISREL model by providing syntax to specify the essential components of the LISREL model. However, LISMOD itself does not interpret a LISREL program.

The LISMOD modeling language in PROC CALIS also supports the mean structure analysis. This is done by providing additional MATRIX statement input for the mean model matrices in the LISREL model.

If you understand the LISREL model, here are three things you input by using the LISMOD language:
1.The ordered lists of x, y, ξ, and η variables
2.MATRIX statements to define free and fixed parameters
3.Names for parameters (not required for free parameters)

(Note: This slide was updated after the printing of the handout)
The Default covariance structure parameters in the LISMOD language are:

1.Diagonal elements of all covariance matrices (all variances)
2.Lower triangular elements of the _PHI_ matrix (covariances of the ξ- variables)

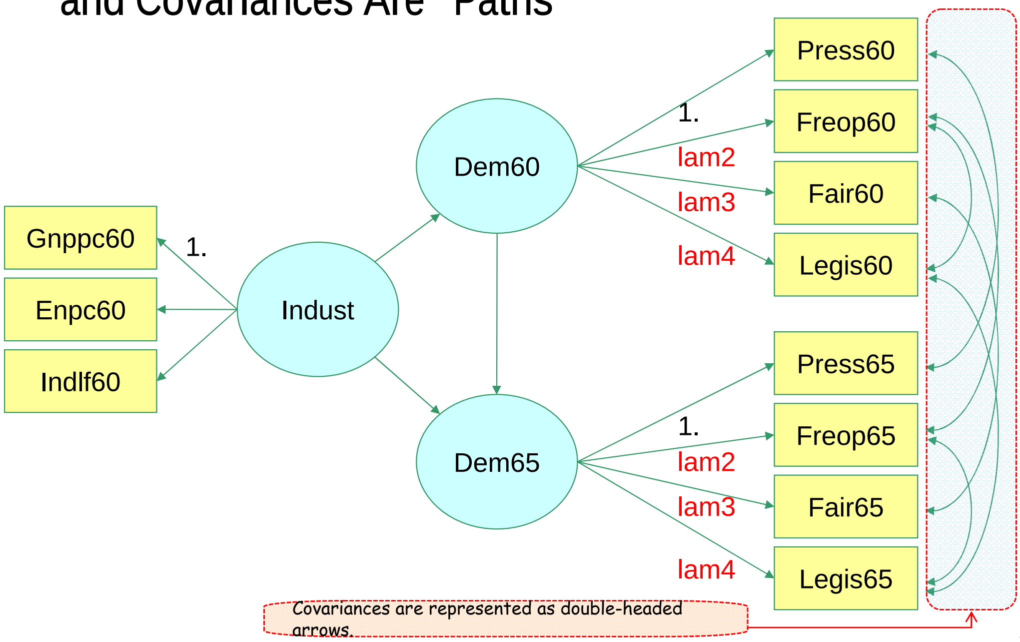
Specifying the default parameters explicitly is certainly allowed, especially when you need to set constraints on these parameters.

In addition, when the mean structures are modeled, the intercepts of the x- and y- variables are default free parameters, while the intercepts of the η- variables and the means of the ξ- variables are fixed zeros by default.

The Generalized PATH Modeling Language

75

Political Democracy and Industrialization: Variances and Covariances Are "Paths"



To generalize the PATH modeling language, error covariances in the path diagram could also be specified as "paths" in PROC CALIS. In fact, covariances in the path diagram are already represented as double-headed arrows, as shown in the political democracy and industrialization example.

An Example of the Generalized PATH Modeling Language

```
PROC CALIS DATA=polidem;
   PATH
      Dem60  ---> Press60 Freop60 Fair60 Legis60   =  1.  lam2  lam3  lam4,
      Dem65  ---> Press65 Freop65 Fair65 Legis65   =  1.  lam2  lam3  lam4,
      Indust ---> Gnppc60 Enpc60 Indlf60           =  1.,
      Indust ---> Dem60 Dem65,
      Dem60  ---> Dem65;
   PCOV
      Freop60 Legis60, Freop65 Legis65,
      Press60 Press65, Freop60 Freop65,
      Fair60 Fair65, Legis60 Legis65;
```

Use the PCOV statement to specify the covariances

```
proc calis data=polidem;
   path
      Dem60  ---> Press60 Freop60 Fair60 Legis60   =  1.  lam2  lam3  lam4,
      Dem65  ---> Press65 Freop65 Fair65 Legis65   =  1.  lam2  lam3  lam4,
      Indust ---> Gnppc60 Enpc60 Indlf60           =  1.,
      Indust ---> Dem60 Dem65,
      Dem60  ---> Dem65,
      Freop60 <--> Legis60, Freop65 <--> Legis65,
      Press60 <--> Press65, Freop60 <--> Freop65,
      Fair60  <--> Fair65,  Legis60 <--> Legis65;
```

Use the generalized path to specify covariances (and variances)

77

The top panel shows the use of PCOV statement to specify the covariances. The bottom panel shows that these covariances are specified as double-headed paths, which resemble their representations in the path diagram.

The two PROC CALIS specifications shown above are equivalent.

Political Democracy and Industrialization: Output with Generalized Paths

PATH List

Path			Parameter	Estimate	Standard Error	t Value
Dem60	--->	Press60		1.00000		
Dem60	--->	Freop60	lam2	1.19079	0.14020	8.49336
Dem60	--->	Fair60	lam3	1.17454	0.12121	9.68988
Dem60	--->	Legis60	lam4	1.25099	0.11757	10.64006
Dem65	--->	Press65		1.00000		
Dem65	--->	Freop65	lam2	1.19079	0.14020	8.49336
Dem65	--->	Fair65	lam3	1.17454	0.12121	9.68988
Dem65	--->	Legis65	lam4	1.25099	0.11757	10.64006
Indust	--->	Gnppc60		1.00000		
Indust	--->	Enpc60	_Parm01	2.17966	0.13932	15.64530
Indust	--->	Indlf60	_Parm02	1.81821	0.15290	11.89126
Indust	--->	Dem60	_Parm03	1.47133	0.39496	3.72529
Indust	--->	Dem65	_Parm04	0.60046	0.22722	2.64267
Dem60	--->	Dem65	_Parm05	0.86504	0.07538	11.47648
Freop60	<-->	Legis60	_Parm06	1.45956	0.70251	2.07764
Freop65	<-->	Legis65	_Parm07	1.39032	0.58859	2.36212
Press60	<-->	Press65	_Parm08	0.59042	0.36307	1.62619
Freop60	<-->	Freop65	_Parm09	2.21252	0.75242	2.94054
Fair60	<-->	Fair65	_Parm10	0.72123	0.62333	1.15706
Legis60	<-->	Legis65	_Parm11	0.36769	0.45324	0.81125

78

The results obtained from PROC CALIS now shows the covariance estimates as "paths" in the PATH list. This table could be used directly in research paper for reporting the SEM estimation results.

Features of the Generalized PATH Modeling Language

- Extension of the PATH modeling language
- Represents all generalized paths in the PATH statement
- Variance-path: Y <--> Y
- Covariance-path: X <--> Y
- Mean or intercept (one-path): **1** ---> Y

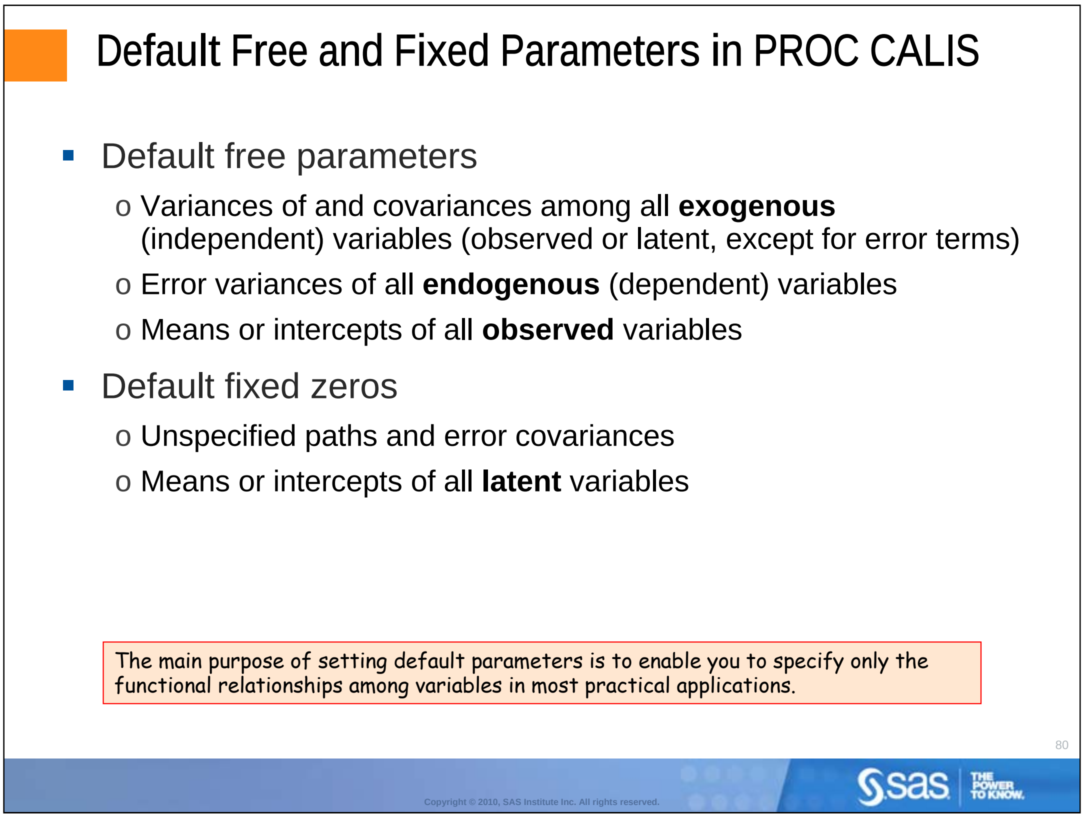
79

In sum, the generalized path modeling language enables you to specify all types of arrows in the path diagram as "paths," including the variance, covariance, intercept, and mean parameters.

Variance of Y is a path like Y <--> Y.

Covariance between X and Y is a path like X <--> Y

Mean or intercept for Y is one-path like 1--->Y.



Knowing the default free and fixed parameters in PROC CALIS are useful because it enhances the coding efficiency and accuracy. Here is a list of default free parameters and fixed zeros used in PROC CALIS:

(This slide has been changed slightly after the printing of the handout)

- Default free parameters
  - Variances of and covariances among all **exogenous** (independent) variables (observed or latent, except for error terms)
  - Error variances of all **endogenous** (dependent) variables
  - Means or intercepts of all **observed** variables
- Default fixed zeros
  - Unspecified paths and error covariances
  - Means or intercepts of all **latent** variables

At the first glance, it might seem to be tedious and demanding that modelers must remember all these default parameter rules to specify an SEM accurately. However, the default parameterization used in PROC CALIS matches that of regression analysis and it is designed with the following main purpose in mind: In most practical applications, you would only need to specify the functional relationships among variables (that is, the single-headed paths in the path diagram) and the fixed variances of the latent variables.

Multiple-Group Analysis

Multiple-group analysis represents an important class of SEM applications.

PROC CALIS Syntax for Specifying a Multiple-Group Analysis

```
proc calis;
   group 1 / data=ds1;    /* ds1: data set for Group 1 */
   group 2 / data=ds2;    /* ds2: data set for Group 2 */
   model 1 / group=1;     /* Group 1 is fitted by Model 1 */

     /* Insert the Model Specification for Group 1 */

   model 2 / group=2;     /* Group 2 is fitted by Model 2 */

     /* Insert the Model Specification for Group 2 */

run;
```

82

The skeleton of the multiple-group syntax of PROC CALIS is shown in this slide.

In this specification, you have two independent groups, which are stored in separate data sets 'ds1' and 'ds2.' You define two models: Model 1 and Model 2, respectively, for fitting to the two groups.

Under each MODEL statement, you specify the model by one of the modeling languages supported by PROC CALIS. For example, a PATH statement or a LISMOD statement.

Group-Model Mapping and Model Referencing

```
proc calis;
   group 1 / data=ds1;
   group 2 / data=ds2;
   group 3 / data=ds3;
   model 1 / group = 1,2;
      path
         x1 ---> x2-x5,
         x2 ---> x3-x5,
         x3 ---> x5     = effect1;
   model 2 / group = 3;
      refmodel 1;
      path
         x3 ---> x5     = effect2;
run;
```

Both Groups 1 and 2 are fitted by Model 1.

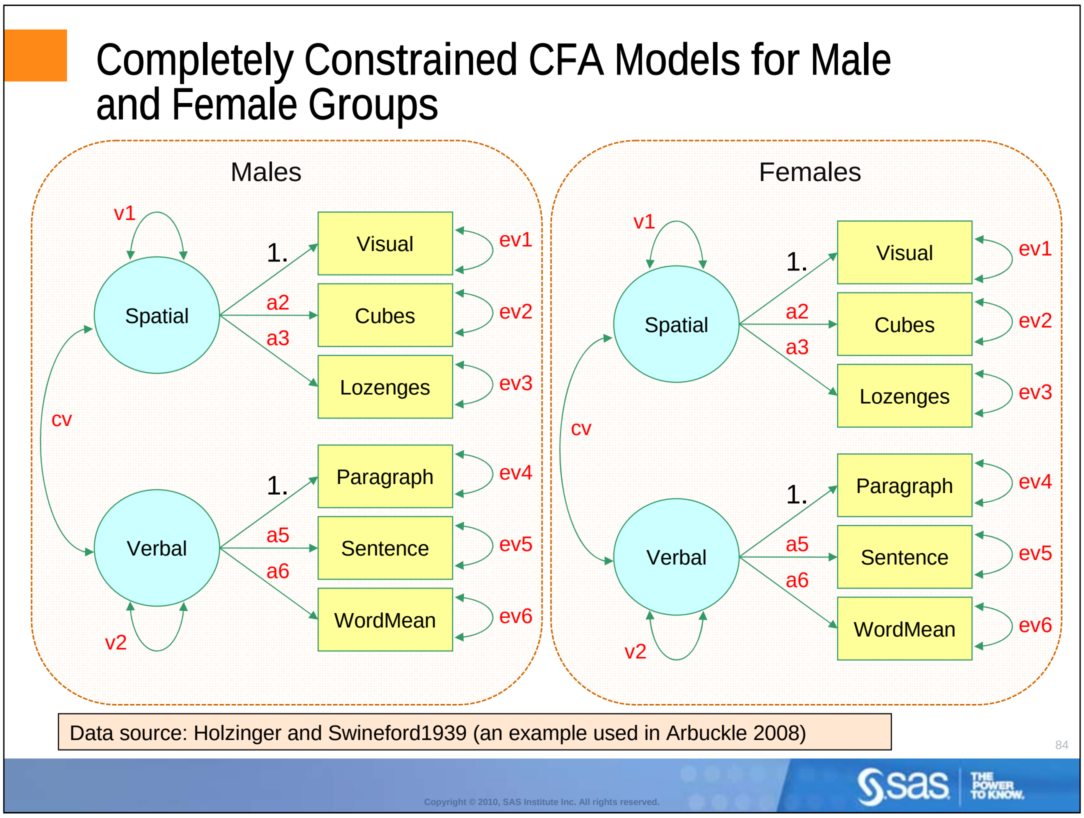
Model 2 makes reference to Model 1, but with a unique parameter for the path from x3 to x5. All other paths are constrained between the two models.

Note: REFMODEL does not constrain default free parameters in the two models. It constrains only those parameters that are specified explicitly.

83

This example shows three groups of data. Model 1 is fitted to Groups 1 and 2 by the PATH model specified immediately after. Model 2 is fitted to Group 3. To define Model 2, a REFMODEL statement is used. REFMODEL 1 means that the specifications in Model 1 is referenced here. This means that all **explicit** specifications in Model 1 are copied into the current model. For example, all the PATH statement specifications in Model 1 are copied into Model 2. However, REFMODEL allows modifications from the reference model. In this example, the x3 --->x5 path is re-specified with a new path coefficient called 'effect2.' This 'effect2' in Model 2 is a different parameter than 'effect1' in Model 1, even though both model have this x3--->x5 path. Except for this x3--->x5 path, Models 1 and 2 share all the remaining specified paths with the same set of path coefficients.

Notice that by using the REMODEL statement to connect models, all (and only) the **explicit** specifications in the models are constrained (unless being modified or re-specified). The constraints, however, will not apply to default parameters (assuming that they are not explicitly specified). To constrain the default parameters, they will have to be specified explicitly in the MODEL definitions.



Let us use an example to illustrate the multiple-group analysis. The Holzinger and Swineford data are sued. This data set is also used in Arbuckle's AMOS manual.

In this research, visual and verbal test scores were observed. Visual, Cubes, and Lozenges are spatial tests. Paragraph, Sentence, and WordMean are verbal tests. CFA models were hypothesized for the two groups: one group is for males and the other for females. The basic factor structures for the groups are the same. Two latent factors 'Spatial' and 'Verbal' are assumed for the measured variables. All parameters (including those default parameters in PROC CALIS) in the two models for the groups are labeled in red.

This is a completely constrained multiple-group model because the two path diagrams for males and females are exactly the same (that is, both have the same path structures and the same set of parameters).

Different Methods to Specify the Completely Constrained Models

1. Male and female groups fit by two models, but with all parameters (including all default parameters) being constrained in the models
2. Male and female groups fit by a single model definition
3. Male and female groups fit by two models that are constrained through the REFMODEL specification

85

In PROC CALIS, you can use one of the following three ways to specify the preceding completely constrained model:

1.Male and female groups fitted by two models, but with all parameters (including all default parameters) being constrained in the models
2.Male and female groups fitted by a single model definition
3.Male and female groups fitted by two models that are constrained through the REFMODEL specification

I will describe each of these methods.

Method 1: Two Groups Fitted by the Two Completely Constrained Models

```
proc calis;
   group 1 / label='Males' data=males;
   group 2 / label='Females' data=females;
   model 1 / group = 1;
      path
         Spatial  ---> Visual Cubes Lozenges       =  1. a2 a3,
         Verbal   ---> Paragraph Sentence Wordmean =  1. a5 a6;
      pvar
         Visual Cubes Lozenges Paragraph Sentence Wordmean = ev1-ev6,
         Spatial = v1, Verbal = v2;
      pcov
         Spatial Verbal = cv;
   model 2 / group = 2;
      path
         Spatial  ---> Visual Cubes Lozenges       =  1. a2 a3,
         Verbal   ---> Paragraph Sentence Wordmean =  1. a5 a6;
      pvar
         Visual Cubes Lozenges Paragraph Sentence Wordmean = ev1-ev6,
         Spatial = v1, Verbal = v2;
      pcov
         Spatial Verbal = cv;
   fitindex on(only) = [chisq df probchi rmsea srmsr aic
         bentlercfi agfi] noindextype;
run;
```

Models are constrained through the use of the same parameter names.

86

The first method is to define two models for the two groups. The model specifications under the two MODEL statements must be exactly the same.

This method is intuitive, but a little clumsy because you need to specify all parameters with matching names in the models (although you can cut-and-paste the model specifications to ensure an exact copy). You also need to specify each parameter in the model, including the default parameters, which you might sometimes miss.

Method 2: Two Groups Fitted by the Same Model

```
proc calis;
   group 1 / label='Males' data=males;
   group 2 / label='Females' data=females;
   model 1 / group = 1, 2;
      path
         Spatial  ---> Visual Cubes Lozenges       =  1. ,
         Verbal   ---> Paragraph Sentence Wordmean =  1. ;
   fitindex on(only) = [chisq df probchi rmsea srmsr aic
         bentlercfi agfi] noindextype;
run;
```

Groups 1 and 2 are fitted by the same model.

87

The second method is very simple and intuitive. You specify one model and fit this model to the two gender groups. This ensures the groups are fitted exactly by the same model.

The advantage of this method is that it is simple, intuitive, and no parameter names are necessary for constraining models. Also, you do not need to specify any of the default parameters explicitly for setting up constraints.

This is an ideal specification method if this model is all what you want to fit. However, if you are going to fit a sequence of multiple-group models to the groups, you might want to consider the next method.

Method 3: Two Groups Fitted by the Two Models That Are Constrained Through the REFMODEL

```
proc calis;
   group 1 / label='Males' data=males;
   group 2 / label='Females' data=females;
   model 1 / group = 1;
      path
         Spatial   ---> Visual Cubes Lozenges       =  1. ,
         Verbal    ---> Paragraph Sentence Wordmean =  1. ;
      pvar
         Visual Cubes Lozenges Paragraph Sentence Wordmean
         Spatial Verbal;
      pcov
         Spatial Verbal;
   model 2 / group = 2;
      refmodel 1;
   fitindex on(only) = [chisq df probchi rmsea srmsr aic
         bentlercfi agfi] noindextype;
run;
```

The REFMODEL statement makes reference to all explicit specifications in Model 1.

Note: This method is used for the completely constrained model and the subsequent models with parameter constraints.

88

The third method constrains the models by the REFMODEL statement. As discussed previously, the REFMODEL copies all the **explicit** specifications from the reference model to the target model. In this slide, all path coefficients, variance parameters, and covariance parameters are specified in Model 1, which is fitted to Group 1 (Males). Model 2, which is fitted to Group 2 (Females), makes reference to Model 1 without any modifications or re-specifications.

Notice that in order to completely constrain the two models for the two groups, all parameters, including those could have been set by default by PROC CALIS (e.g., specifications in the PVAR statement and PCOV statement), must be specified **explicitly** in Model 1. This way, Model 2 will copy all these parameter specifications via the REFMODEL statement specification.

Unlike Method 1, this method does not require the use of parameter names for constraints across models. Constraints are done via the REFMODEL statement. Although not as intuitive as Method 2, this method would be useful if you need to consider fitting a sequence of multiple-group models, which will be illustrated later.

Fit Summary of the Completely Constrained Multiple-Group Model

Fit Summary

Chi-Square	26.0154
Chi-Square DF	29
Pr > Chi-Square	0.6247
Standardized RMSR (SRMSR)	0.0968
Adjusted GFI (AGFI)	0.9235
RMSEA Estimate	0.0000
Akaike Information Criterion	52.0154
Bozdogan CAIC	103.7130
Schwarz Bayesian Criterion	90.7130
Bentler Comparative Fit Index	1.0000

Not a bad fit.

89

The completely constrained model provide a good fit of the data. The model fit chi-square is not significant. The RMSEA is perfect, although the SRMSR is not very good. The AGFI and the CFI are also good. The AIC, the CAIC, and the SBC are also printed. These indices cannot be interpreted by their absolute values, but will be useful when you compare the fits of different multiple-group models. You will use these indices to select the “best” multiple-group model for the data later.

Fitting Less Restrictive Multiple-Group Models

- Completely constrained multiple-group model: Error variances, structural covariances, and loadings are all constrained
- Release the constraints on error variances
- Release the constraints on structural covariances
- Release the constraints on the loadings – Completely unconstrained

90

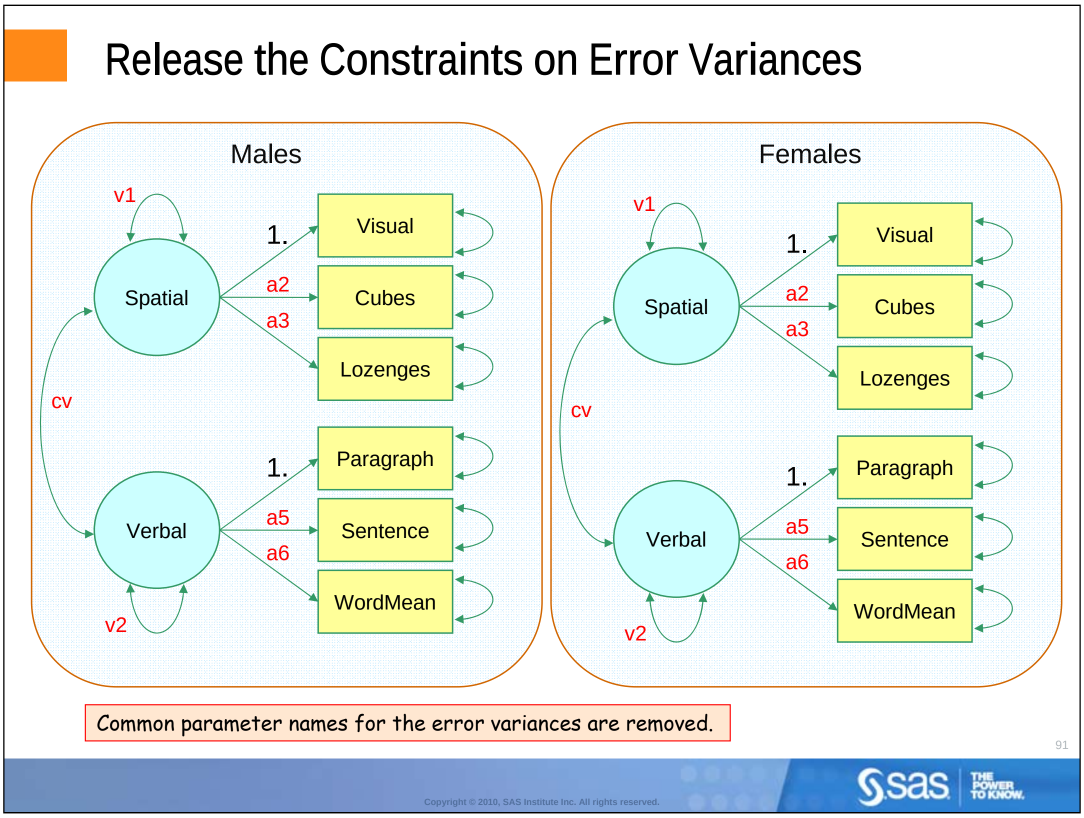
We have fitted the completely constrained multiple-group model by the REFMODEL method. We can further fit less constrained multiple-group model by modifying our PROC CALIS code.

We can release the constraints on error variances.

Then we can release the constraints on the structural covariances (among latent variables).

Finally, we can release the constraints on the path coefficients (or loadings).

I am going to show these step by step.



The path diagrams for the multiple-group model that releases the constraints on the error variances are shown above.

Except for the error variance parameters, all the remaining parameters are labeled. This means that only the error variance parameters are not invariant across the models for the groups.

Releasing the Constraints on the Error Variances

```
proc calis;
   group 1 / label='Males' data=males;
   group 2 / label='Females' data=females;
   model 1 / group = 1;
      path
         Spatial   ---> Visual Cubes Lozenges       =  1. ,
         Verbal    ---> Paragraph Sentence Wordmean =  1. ;
      pvar
         /* Visual Cubes Lozenges Paragraph Sentence Wordmean */
         Spatial Verbal;
      pcov
         Spatial Verbal;
   model 2 / group = 2;
      refmodel 1;
   fitindex on(only) = [chisq df probchi rmsea srmsr aic
         bentlercfi agfi] noindextype;
run;
```

Comment out the error variance specifications in the PVAR statement, and let PROC CALIS set two distinct sets of default error variances for the two models.

92

In terms of PROC CALIS specification, this means that the model for females makes reference to the model for males with regard to those constrained parameters only.

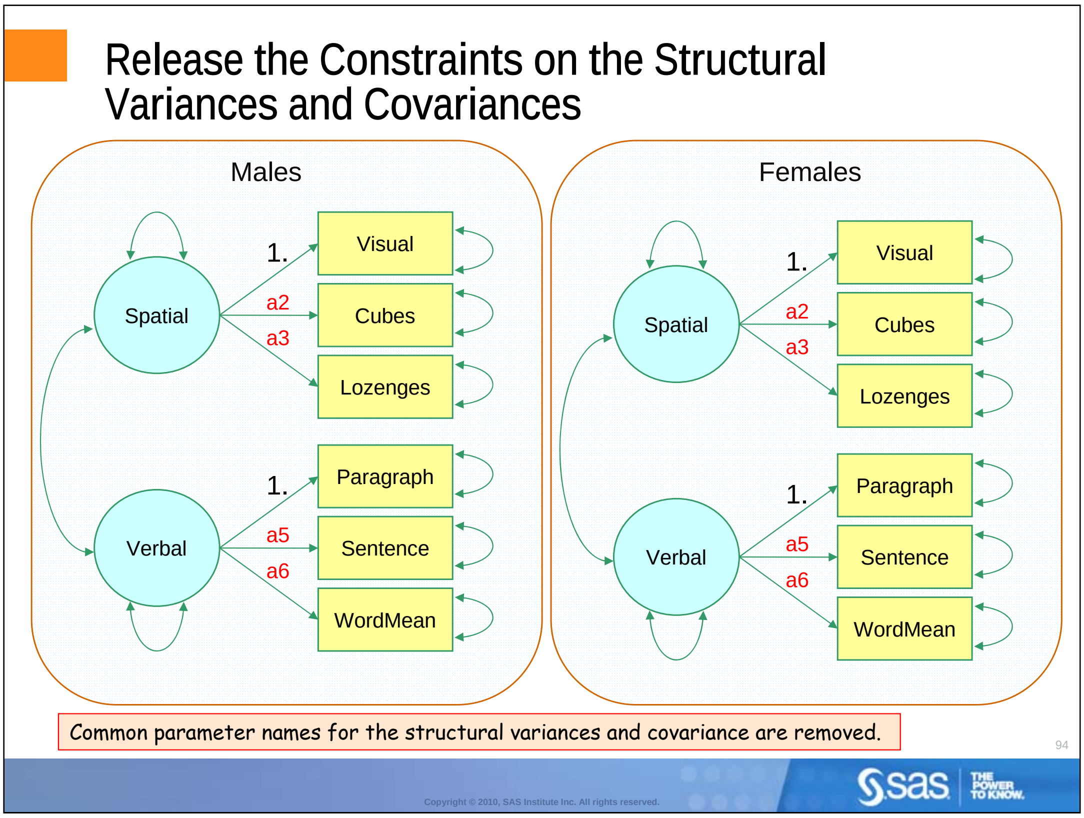
This could be done very easily from modifying the completely constrained multiple-group model. All you need to do is to comment out the PVAR statement specifications for the observed variables.

When Model 2 makes reference to Model 1, only those explicit specifications would be constrained between the two models. Because the error variances are not specified in both models (that is, commented out from the previous code), PROC CALIS would generate different sets of default error variance parameters for the two models. In other words, the error variance constraints are released in this PROC CALIS specification.

Fit Summary of the Multiple-Group Model with Constraints on Loadings and Structural Covariances

Fit Summary	
Chi-Square	22.0334
Chi-Square DF	23
Pr > Chi-Square	0.5182
Standardized RMSR (SRMSR)	0.0903
Adjusted GFI (AGFI)	0.9163
RMSEA Estimate	0.0000
Akaike Information Criterion	60.0334
Bozdogan CAIC	135.5913
Schwarz Bayesian Criterion	116.5913
Bentler Comparative Fit Index	1.0000

The model fit chi-square still is not significant, indicating a good model fit. The RMSEA, the AGFI, and the CFI are all good. However, the SRMSR does not indicate a good model fit.



How about releasing the constraints on the structural covariances?

In the path diagram, only the path coefficients are constrained now (by using the same set of parameter names). This means that only the path effects are invariant across the models for the groups.

Releasing the Constraints on the Error Variances and Structural Covariances

```
proc calis;
   group 1 / label='Males' data=males;
   group 2 / label='Females' data=females;
   model 1 / group = 1;
      path
         Spatial  ---> Visual Cubes Lozenges       =  1. ,
         Verbal   ---> Paragraph Sentence Wordmean =  1. ;
      /*
      pvar
         Visual Cubes Lozenges Paragraph Sentence Wordmean
         Spatial Verbal;
      pcov
         Spatial Verbal;
      */
   model 2 / group = 2;
      refmodel 1;
   fitindex on(only) = [chisq df probchi rmsea srmsr aic
         bentlercfi agfi] noindextype;
run;
```

Comment out the PVAR and PCOV statements, and let the PROC CALIS set two distinct sets of default variances and covariances for the two models.

95

This new multiple-group model can be specifying by commented out the explicit specifications of the structural covariances (variances and covariances among latent variables) in Model 1.

When Model 2 makes reference to Model 1, it copies the explicit specifications in the PATH statement of Model 1. Error variances, structural variances and covariances in the two models are now set by default and are unconstrained between the two models.

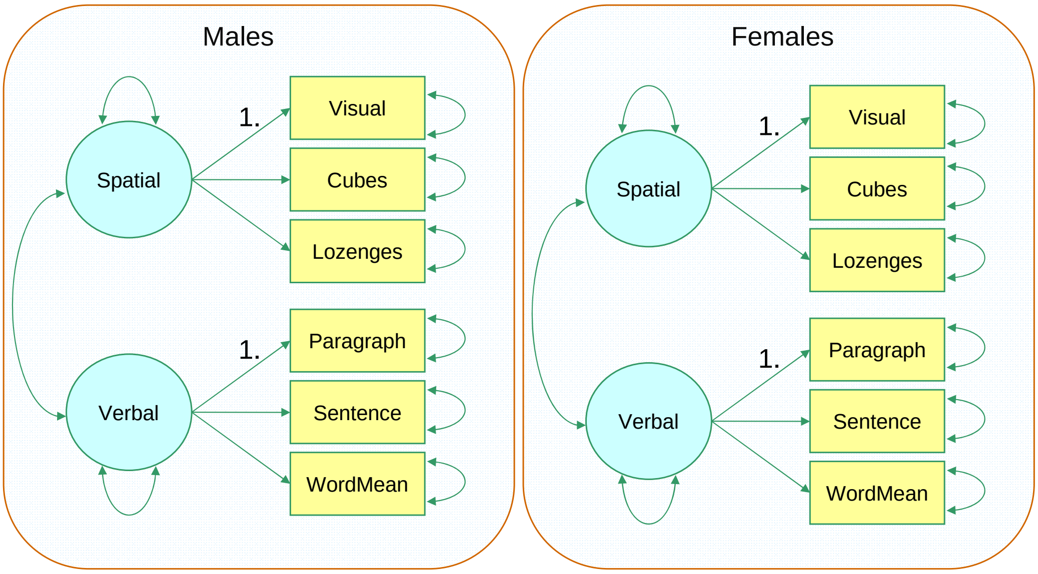
Fit Summary of the Multiple-Group Model with Loading Constraints

Fit Summary	
Chi-Square	18.2915
Chi-Square DF	20
Pr > Chi-Square	0.5682
Standardized RMSR (SRMSR)	0.0539
Adjusted GFI (AGFI)	0.9179
RMSEA Estimate	0.0000
Akaike Information Criterion	62.2915
Bozdogan CAIC	149.7796
Schwarz Bayesian Criterion	127.7796
Bentler Comparative Fit Index	1.0000

96

The model fit chi-square is not significant. Now, the SRMSR is acceptable. The AGFI, the RMSEA, and the CFI continue to be very good.

Release the Constraints on the Loadings



Common parameter names for the loadings are removed.

97

Finally, for the completely unconstrained multiple-group the path diagrams for the two groups are the same, but no parameter names (except for fixed values of 1) are used to denote constraints.

Completely Unconstrained Multiple-Group Model

```
proc calis;
   group 1 / label='Males' data=males;
   group 2 / label='Females' data=females;
   model 1 / group = 1;
      path
         Spatial   ---> Visual Cubes Lozenges       =  1. ,
         Verbal    ---> Paragraph Sentence Wordmean =  1. ;
   model 2 / group = 2;
      path
         Spatial   ---> Visual Cubes Lozenges       =  1. ,
         Verbal    ---> Paragraph Sentence Wordmean =  1. ;
   fitindex on(only) = [chisq df probchi rmsea srmsr aic
         bentlercfi agfi] noindextype;
run;
```

The REFMODEL statement is not used here because the two models are not constrained with each other.

98

Because the two models for the groups are totally unrelated, you should not need to use the REFMODEL statement any more. Instead, the two models are defined exactly by the same PATH statement specifications. However, because no common parameter names are used for the path coefficients, the two models are not constrained.

Fit Summary of the Completely Unconstrained Multiple-Group Model

Fit Summary	
Chi-Square	16.4795
Chi-Square DF	16
Pr > Chi-Square	0.4200
Standardized RMSR (SRMSR)	0.0449
Adjusted GFI (AGFI)	0.9077
RMSEA Estimate	0.0205
Akaike Information Criterion	68.4795
Bozdogan CAIC	171.8746
Schwarz Bayesian Criterion	145.8746
Bentler Comparative Fit Index	0.9984

The unconstrained SEM model for the groups gives you the best fit, but it is also the least interesting multiple-group model.

99

All fit indices indicate very good fit of the completely unconstrained multiple-group model.

Chi-Square Difference Tests for the Nested Multiple-Group Models

	Completely Constrained	Constrained Loadings and Struct. Cov.	Constrained Loadings
Constrained Loadings and Structural Covariances	3.892 (p=0.32)		
Constrained Loadings	7.724 (p=0.44)	3.742 (p=0.71)	
Completely Unconstrained	9.536 (p=.27)	5.539 (p=.41)	1.182 (p=.23)

There are no significant differences between the multiple-group models.

100

Which model is the best for the data?

Chi-square difference tests provide a statistical method to see if models are significantly different from each other. This slides shows the chi-square difference tests for comparing the four multiple-group models.

As all p-values are bigger than 0.05, it means all these multiple-group models are not significantly different from each other.

Comparing Model Fits by Using Various Fit Indices

	Completely Constrained	Constrained Loadings and Struct. Cov.	Constrained Loadings	Completely Unconstrained
Chi-Square	26.0154	22.0334	18.2915	16.4795
Chi-Square DF	29	23	20	16
Pr > Chi-Square	0.6247	0.5182	0.5682	0.4200
Standardized RMSR (SRMSR)	0.0968	0.0903	0.0539	0.0449
Adjusted GFI (AGFI)	0.9235	0.9163	0.9179	0.9077
RMSEA Estimate	0.0000	0.0000	0.0000	0.0205
Akaike Information Criterion	52.0154	60.0334	62.2915	68.4795
Bozdogan CAIC	103.7130	135.5913	149.7796	171.8746
Schwarz Bayesian Criterion	90.7130	116.5913	127.7796	145.8746
Bentler Comparative Fit Index	1.0000	1.0000	1.0000	0.9984

Absolute indices: Chi-square, SRMSR (smaller is better)
Parsimonious indices: AGFI (larger is better),
RMSEA, AIC, CAIC, SBC (smaller is better)
Incremental indices: Bentler CFI (larger is better)

101

We can also compare the four models by means of the fit index values.

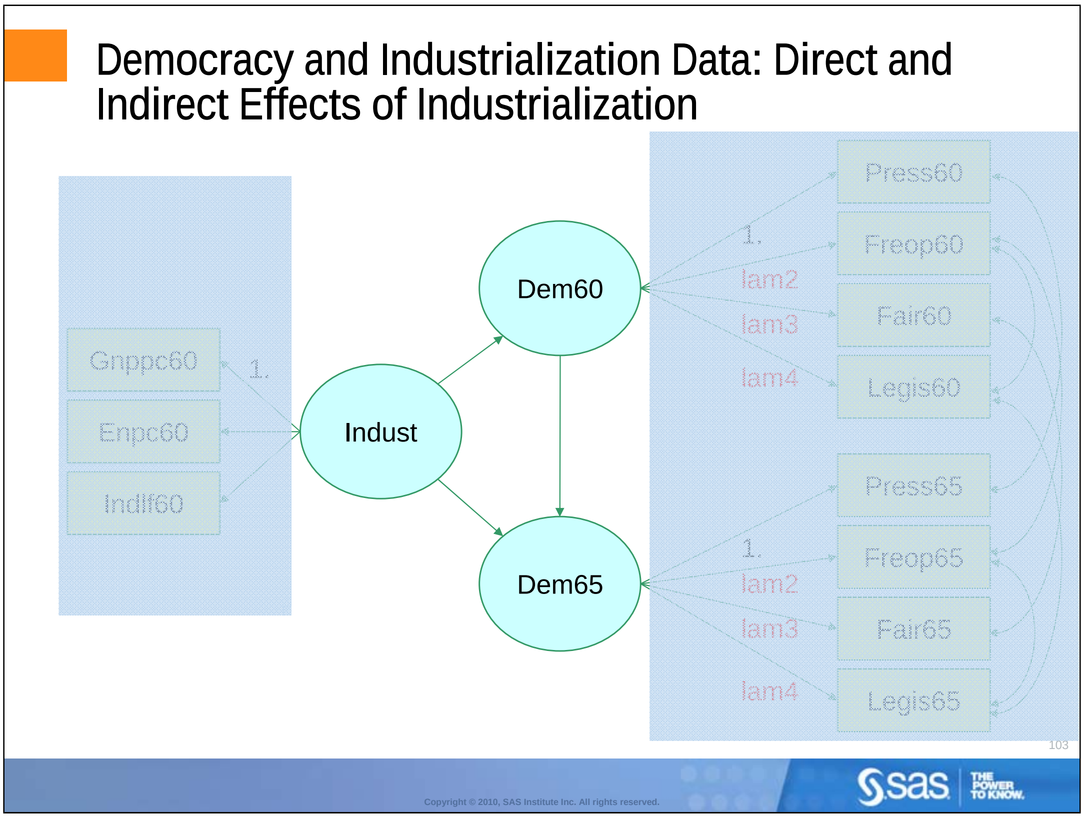
The model fit chi-square value always favors the model with the largest number of parameters. So, according to the model fit chi-square, the completely unconstrained model is the best model. The SRMSR also favors the completely unconstrained model simply because it can be viewed as a monotone transformation of the chi-square value. However, you should not select your best model based on the absolute indices such as model-fit chi-square value or the SRMSR because these indices do not take model parsimony into account. Complicated models might have perfect model fit chi-square and SRMSR values (i.e., 0). But these complex models should not be selected as the best models because they have very little scientific value.

The AGFI, the RMSEA, the AIC, the CAIC, and the SBC all takes model parsimony into account. For the AGFI, the larger the better. For other indices, the smaller the better. All these parsimonious indices point to the completely constrained model as the best multiple-group model for the data.

Lastly, the incremental fit index Bentler CFI favors the completely constrained model too. However, virtually all multiple-group model in this comparison are equally good according to the CFI. Notice that incremental indices such as the CFI measures how a target model measures better than a so-called baseline model. They do not take model parsimony into account. In addition, they depend on how good the baseline model is used in the computing formula. If the baseline model is very bad (such as the commonly-used uncorrelatedness model), all competing models would have good incremental fit only because the baseline model is much worse. For this reason, incremental fit indices might not serve as good criteria for model selection.

Analyzing Direct and Indirect Effects

102



Analyzing direct and indirect effects is something unique to SEM.

Let us look at the model for the democracy and industrialization data. Only the structural part of the SEM is shown to illustrate the idea.

Industrialization Effects on Democracy in 1960 and 1965

- On democracy in 1960

  A direct effect:

  Indust --->Dem60

- On democracy in 1965

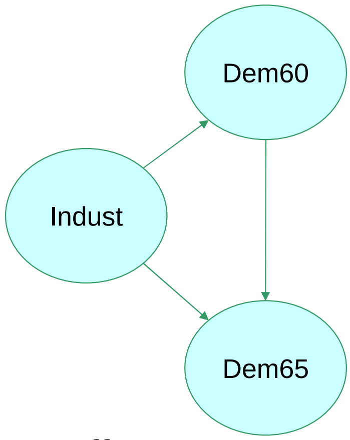
  A direct effect:

  Indust --->Dem65

  An indirect effect:

  Indust --->Dem60 --->Dem65

- Total effect = direct effect + indirect effect



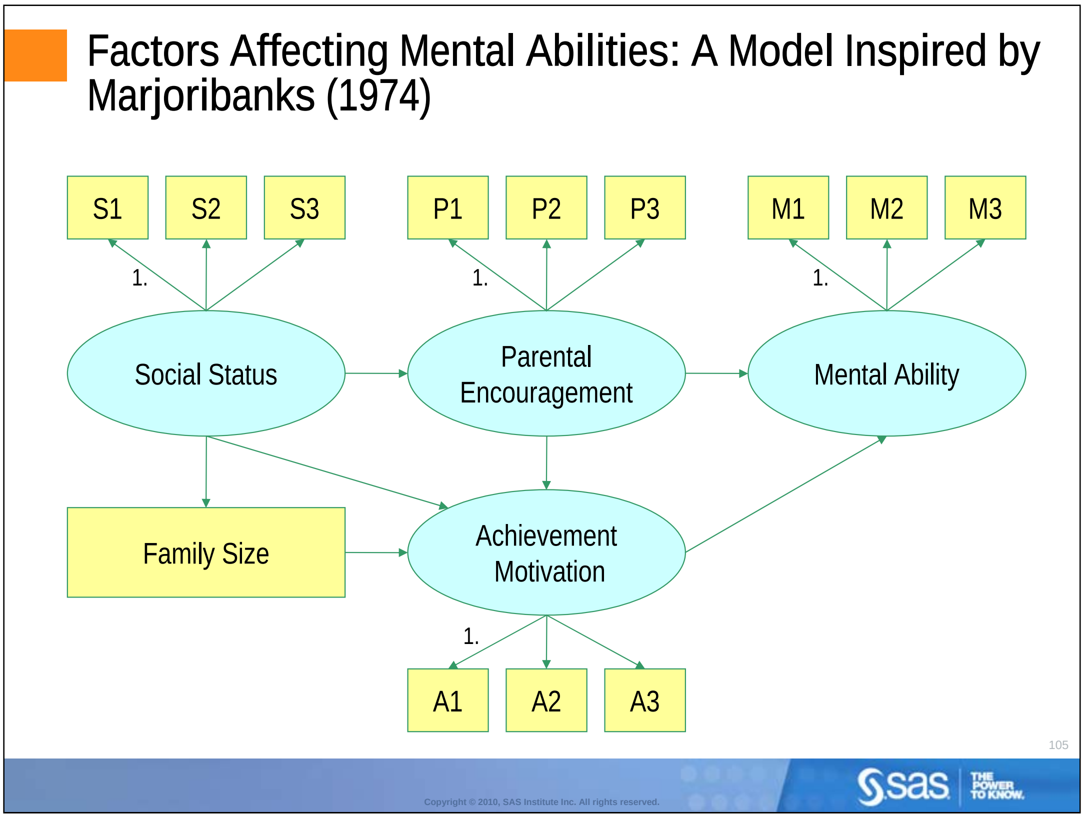
104

First, let us look at the effect of industrialization on the democracy measure in 1960. The direct effect of Indust on Dem60 refers to the path Indust ---> Dem60. This effect can be estimated directly from any SEM software.

On the democracy measure in 1965, industrialization has a direct and indirect effect. The direct effect refers to the path Indust ---> Dem65. The indirect effect is indicated by the track Indust ---> Dem60 ---> Dem65.

When you add up the direct and indirect effect, it gives you the total effect.

In SEM, the direct effects are estimated as the path coefficients. Indirect effects and total effects are functions of the parameter estimates. Fortunately, PROC CALIS can compute these functions efficiently and it can also provide standard error estimates for these effects.



This slide shows a more interesting example about analyzing direct, indirect, and total effects.

The example is inspired by a model of Marjoribanks (1974). The current model is a simplification and the data are generated. The results here do not represent the original study, but would serve well for illustration purposes.

The main idea of the study is to model the mental ability of students. The mental ability is a latent construct, which is supposed to be determined (predicted) by parental encouragement and achievement motivation, both of which are formulated as latent construct in the model. Two remote causes (predictors), social status and family size, have direct effects on parental encouragement and achievement motivation. However, these two remote causes affect the mental ability only indirectly. Social status is also formulated as a latent variable, while family size is an observed variable. For all the latent variables, observed indicators are used and they are represented by small rectangles in the path diagram.

There are some motivating questions about this path diagram regarding the direct and indirect effects. For example,

1.Even though social status does not affect the mental ability, it does have an indirect effect on the mental ability via parental encouragement and achievement motivation. One would like the SEM software to compute this this indirect effect and its significance.
2.Parental encouragement has a direct and an indirect effects on the mental ability. What is the overall total effect of parental encouragement on the mental ability. One would also like the SEM software to compute all these effects and their significance.

Factors Affecting Mental Abilities: PROC CALIS Code

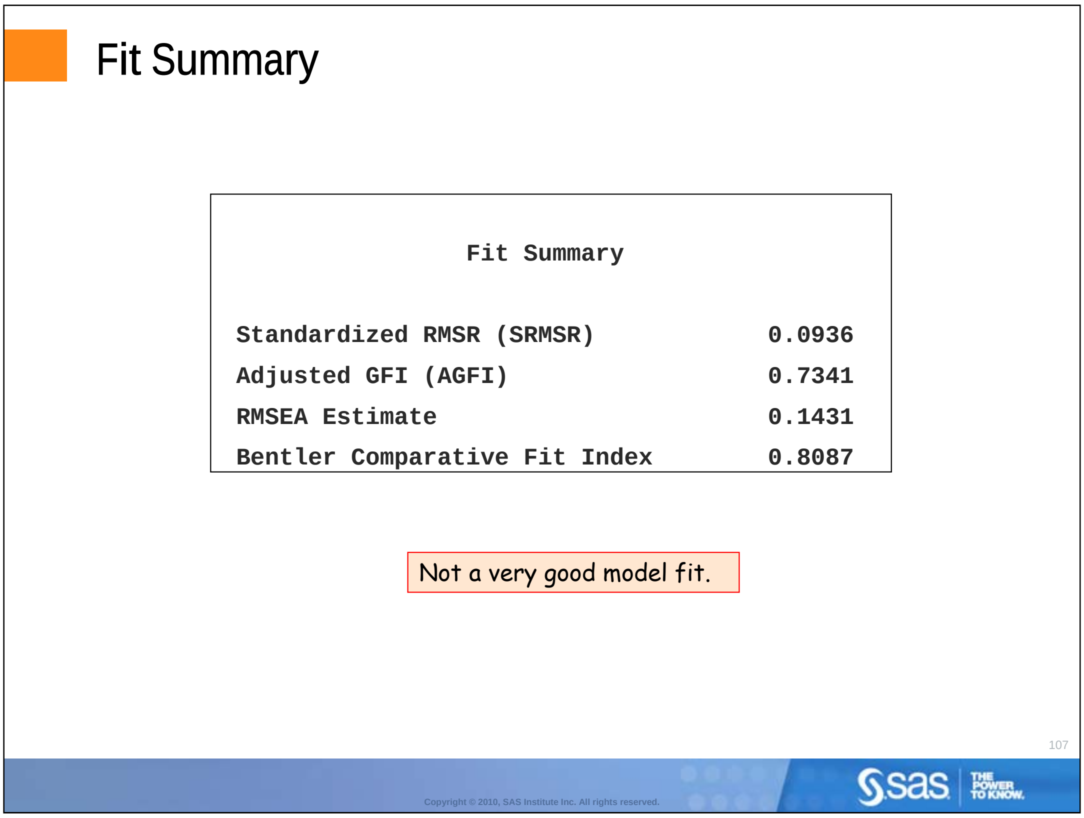
```
proc calis data=mental nobs=115 effpart;
   path
      /* Structural Model */
      SocialStatus --->  ParentalEncouragement FamilySize
                         AchievementMotivation,
      FamilySize   --->  AchievementMotivation,
      ParentalEncouragement  --->  AchievementMotivation MentalAbility,
      AchievementMotivation  --->  MentalAbility,

      /* Measurement Model */
      SocialStatus          ---> S1 S2 S3   = 1.,
      ParentalEncouragement ---> P1 P2 P3   = 1.,
      AchievementMotivation ---> A1 A2 A3   = 1.,
      MentalAbility         ---> M1 M2 M3   = 1.;
   fitindex on(only)=[agfi srmsr rmsea bentlercfi] noindextype;
run;
```

The EFFPART option analyzes the effect partitioning in the model.

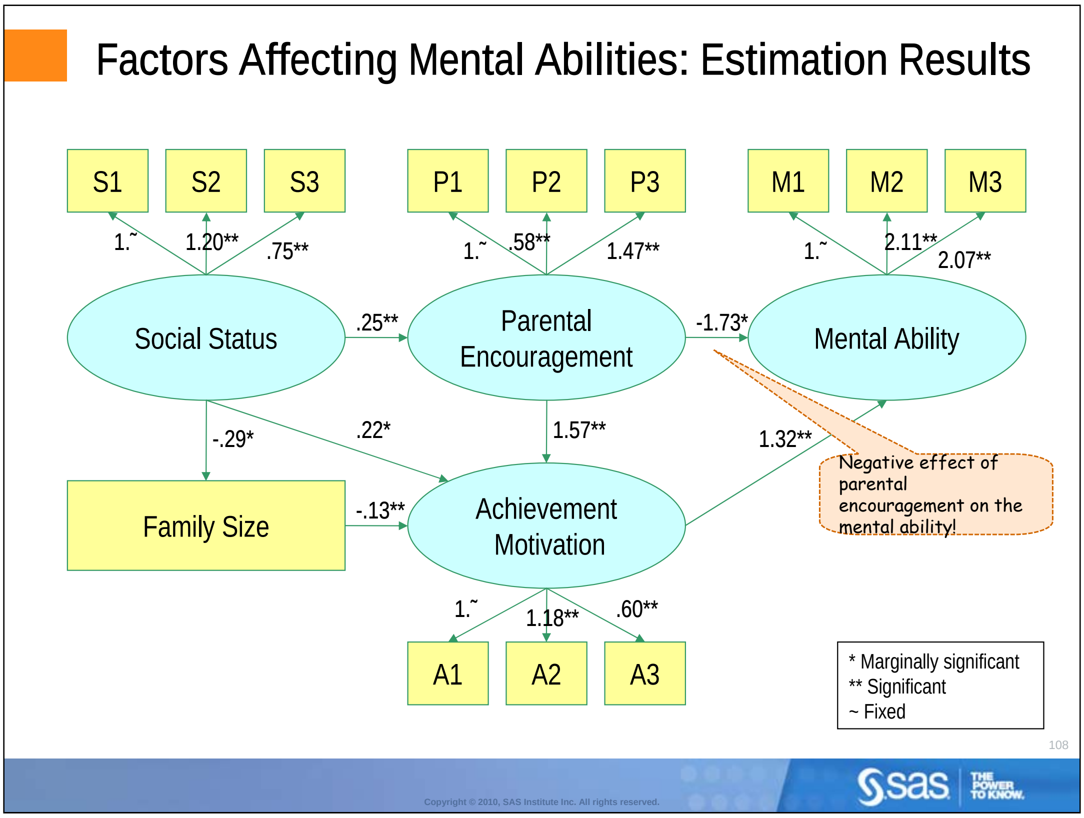
106

Now, the PATH specification for the target model should be easy for you. You can specify the measurement model and the structural model by the multiple-path syntax. You can look at the path diagram and write down the paths in the PATH statement. Notice that each path in the path diagram represents a direct effect of one variable on another variable.

The only new option introduced here is the EFFPART option in the PROC CALIS statement. EFFPART stands for effect partitioning. In other words, it partitions the total effects of any variable on any other variable into direct and indirect effects. PROC CALIS will compute these effects and the standardized version---all with standard error estimates provided.



The model fit actually does not look too good for this simulated data. But this is not the concern here. We want to study the effect partitioning, assuming that we are satisfied with the model fit.



Before diving into the results for the effect partitioning, I want to look at the estimates shown in the path diagram. I want to throw in one more motivation to study direct and indirect effects in a structural equation model.

In this path diagram, estimates are shown with their significance marked. Two asterisks after an estimate means the estimate is statistically significant. One asterisk after an estimate means that the estimate is marginally significant.

I want to focus on the effects of parental encouragement on mental ability. The direct effect is -1.73. This means that parental encouragement has a negative effect on mental ability. This sounds a little strange at the first glance. But if we look at the bigger picture in the path diagram, we can understand why that is so. Notice that parental encouragement has a positive effect on achievement motivation, which in turns has a positive effect on mental ability. The whole picture suggests that purely parental encouragement do not necessarily affect mental ability in a positive way. Sometimes, the more encouragement would only add more pressure to the individual's mental performance---hence the negative direct effect on mental ability observed in the path diagram result. However, when the parental encouragement can affect something more internal of the individuals---namely, the individual's achievement motivation, then it will result in a higher mental ability score. Hence, there is a positive indirect effect of parental encouragement on the mental ability.

In sum, an interesting question in this path diagram result is that what is the overall total effect of parental encouragement on mental ability, given that it has a negative direct effect and a positive indirect effect?

Partitioning of the Effects: A Prerequisite

```
Stability Coefficient of Reciprocal Causation = 0

          Stability Coefficient < 1

       Total and Indirect Effects Converge
```

```
NOTE: The stability coefficient is 0, which is less
      than one. The condition for converged total and
      indirect effects is satisfied.
```


Before you can analyze direct and indirect effects, you should check whether a prerequisite is satisfied. In order to study the effect partitioning legitimately, the so-called stability coefficient must be less then 1. PROC CALIS provides such a check. When you see these messages from the PROC CALIS output, you could proceed to examine your effect partitioning results.

Partitioning of the Effects: Total Effects

Total Effects
Effect / Std Error / t Value / p Value

	FamilySize	Achievement Motivation	Mental Ability	Parental Encouragement	SocialStatus
A1	-0.1287 0.0360 -3.5769 0.000348	1.0000	0	1.5701 0.5176 3.0337 0.002416	0.6523 0.0942 6.9258 <.0001
A2	-0.1522 0.0420 -3.6274 0.000286	1.1821 0.1084 10.9012 <.0001	0	1.8560 0.6051 3.0672 0.002161	0.7710 0.1036 7.4398 <.0001
.	.	.	.	.	.
MentalAbility	-0.1699 0.0572 -2.9696 0.002982	1.3196 0.4124 3.1995 0.001377	0	0.3376 0.4045 0.8346 0.4039	0.4244 0.1280 3.3159 0.000914
ParentalEncouragement	0	0	0	0	0.2516 0.0692 3.6356 0.000277

Details omitted.

110

With the EFFPART option, PROC CALIS produces tables for total, direct, and indirect effects separately. These tables could be large. I just annotate these results here. Some results are not shown.

This table is about the estimates of the total effects, their standard errors, t-values, and significance levels.

Partitioning of the Effects: Direct Effects

Direct Effects
Effect / Std Error / t Value / p Value

	FamilySize	Achievement Motivation	Mental Ability	Parental Encouragement	SocialStatus
A1	0	1.0000	0	0	0
A2	0	1.1821 0.1084 10.9012 <.0001	0	0	0
.	.	.	.	.	.
MentalAbility	0	1.3196 0.4124 3.1995 0.001377	0	-1.7343 0.9047 -1.9169 0.0553	0
ParentalEncouragement	0	0	0	0	0.2516 0.0692 3.6356 0.000277

Details omitted.

111

This table is about the direct effects , their standard errors, t-values, and significance levels.

Partitioning of the Effects: Indirect Effects

Indirect Effects
Effect / Std Error / t Value / p Value

	FamilySize	Achievement Motivation	Mental Ability	Parental Encouragement	SocialStatus
A1	-0.1287	0	0	1.5701	0.6523
	0.0360			0.5176	0.0942
	-3.5769			3.0337	6.9258
	0.000348			0.002416	<.0001
A2	-0.1522	0	0	1.8560	0.7710
	0.0420			0.6051	0.1036
	-3.6274			3.0672	7.4398
	0.000286			0.002161	<.0001
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
MentalAbility	-0.1699	0	0	2.0719	0.4244
	0.0572			1.0483	0.1280
	-2.9696			1.9763	3.3159
	0.002982			0.0481	0.000914
ParentalEncouragement	0	0	0	0	0

Details omitted.

112

This table is about the indirect effects , their standard errors, t-values, and significance levels.

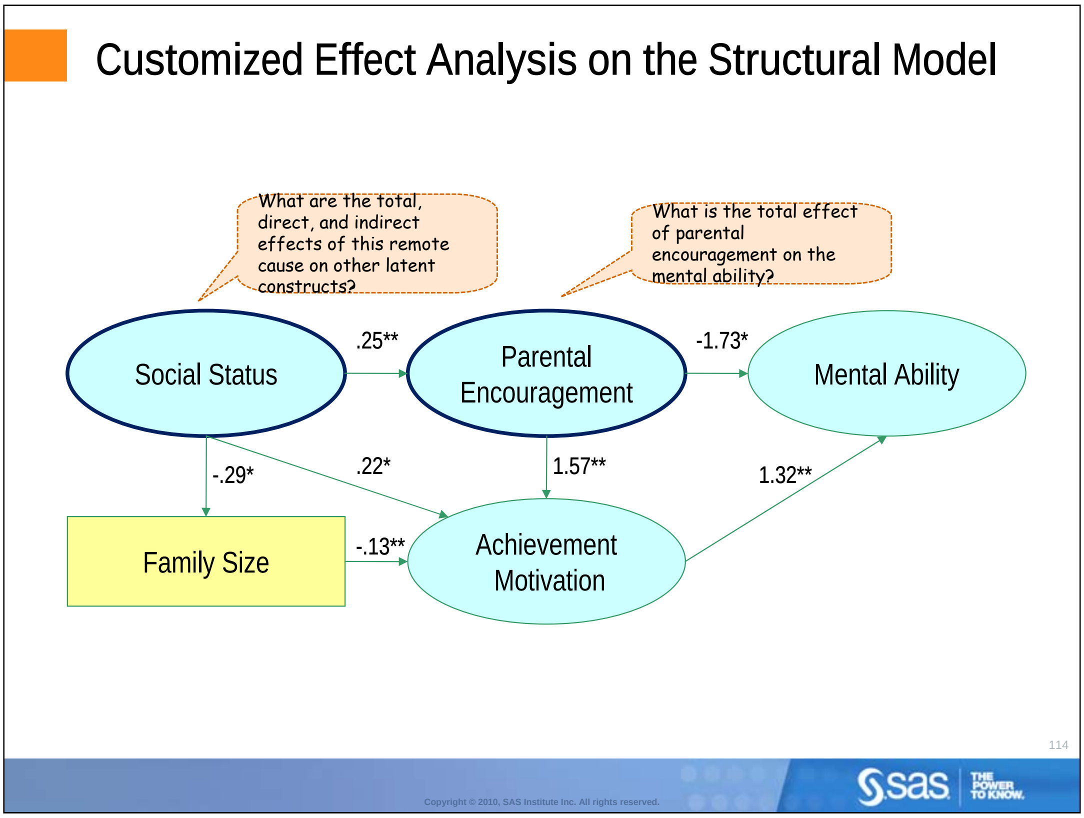
Customized Effect Analysis

- The EFFPART option displays all logical possible effects of the variables
- Columns: Five variables, each of which serves as a predictor at least once:
  - o FamilySize
  - o AchivementMotivation
  - o MentalAbility
  - o ParentalEncouragement
  - o SocialStatus
- Rows: Sixteen variables, each of which serves as an outcome variable at least once (all variables except for SocialStatus)

113

When you have large tables like those shown in previous slides, you are likely to be doing exploratory analysis without specific questions in your mind. The effect tables could get very large and you might have a difficult time to look for the particular results that you are interested in. For example, the columns of the effect tables consist of five variables, each of which serves as a predictor at least once in the path diagram. These five variables have direct or indirect effects on the row variables. The rows consist of sixteen variables, each of which serves as an outcome variable at least once. In the current path diagram, it includes all variables except for the SocialStatus variable.

However, if you have specific research questions in your mind, you are recommended to do customized effect analysis that is supported by PROC CALIS.



In the beginning, we already have these motivating questions.

1.For the social status variable, What are the total, direct, and indirect effects of this remote cause on other latent constructs, especially on the mental ability?
2.What is the total effect of parental encouragement on the mental ability?

Factors Affecting Mental Abilities: Customized Effect Analysis

```
proc calis data=mental nobs=115;
   path
      /* Structural Model */
      SocialStatus -->  ParentalEncouragement FamilySize AchievementMotivation,
      FamilySize   --->  AchievementMotivation,
      ParentalEncouragement  --->  AchievementMotivation MentalAbility,
      AchievementMotivation  --->  MentalAbility,

      /* Measurement Model */
      SocialStatus          ---> S1 S2 S3   = 1.,
      ParentalEncouragement ---> P1 P2 P3   = 1.,
      AchievementMotivation ---> A1 A2 A3   = 1.,
      MentalAbility         ---> M1 M2 M3   = 1.;
   effpart
      SocialStatus          -> ParentalEncouragement AchievementMotivation
                               MentalAbility,
      ParentalEncouragement -> MentalAbility;
run;
```

The EFFPART statement defines the customized effects of interest.

115

PROC CALIS supports the customized effect analysis. This can be done by the EFFPART statement, as shown in the PROC CALIS code in this slide.

First, you want to study the effect partitioning of social status on these three variables: parental encouragement, achievement motivation, and mental ability. Hence, you use the following code in the EFFPART statement:

```
SocialStatus -> ParentalEncouragement AchievementMotivation  MentalAbility,
```

Second, you want to study the effect partitioning of parental encouragement on mental ability. Hence, you use the following code in the EFFPART statement:

```
ParentalEncouragement -> MentalAbility;
```

Effects of Social Status

Effects of SocialStatus
Effect / Std Error / t Value / p Value

	Total	Direct	Indirect
ParentalEncouragement	0.2516	0.2516	0
	0.0692	0.0692	
	3.6356	3.6356	
	0.000277	0.000277	
AchievementMotivation	0.6523	0.2193	0.4330
	0.0942	0.1147	0.1203
	6.9258	1.9125	3.5985
	<.0001	0.0558	0.000320
MentalAbility	0.4244	0	0.4244
	0.1280		0.1280
	3.3159		3.3159
	0.000914		0.000914

Direct effect only (ParentalEncouragement)

Direct and indirect effects (AchievementMotivation)

Indirect effect only (MentalAbility)

116

The effect partitioning results from PROC CALIS are shown in this slide and the next one.

The effects of social status on the three specific latent variables are shown in this table.

On the parental encouragement, social status has a direct effect only, which is positive and significant.

On the achievement motivation, social status has both a direct and an indirect effects. Both of these effects are significant. The total effect is the sum of the direct and indirect effect. The total effect is also significant.

On the mental ability, social status has only an indirect effect, which is also significant.

Effects of Parental Encouragement on Mental Ability

Effects of ParentalEncouragement
Effect / Std Error / t Value / p Value

	Total	Direct	Indirect
MentalAbility	0.3376	-1.7343	2.0719
	0.4045	0.9047	1.0483
	0.8346	-1.9169	1.9763
	0.4039	0.0553	0.0481

Parental encouragement has a positive total effect on the mental ability, although the total effect is not significant.

117

This slide shows the effect partitioning of parental encouragement on mental ability.

The direct effect is negative, as shown previously in the path diagram. This direct effect is marginally significant.

The indirect effect is positive and is statistical significant. This is a piece of "comforting" information---parental encouragement does affect the mental ability positively, but only through its effect on achievement motivation.

The total effect, which is the sum of direct and indirect effect, however, is not significant.

This example shows that SEM effect analysis can show some effect patterns that simply cannot be analyzed by linear regression analysis adequately. The SEM effect analysis provides something more detailed and refined regarding the totality of the theory. In this regard, the customized effect analysis supported by PROC CALIS is very useful.

Standardized Effects of Social Status

Standardized Effects of SocialStatus
Effect / Std Error / t Value / p Value

	Total	Direct	Indirect
ParentalEncouragement	0.6675	0.6675	0
	0.0833	0.0833	
	8.0141	8.0141	
	<.0001	<.0001	
AchievementMotivation	0.6990	0.2350	0.4640
	0.0597	0.1207	0.1153
	11.7105	1.9478	4.0231
	<.0001	0.0514	<.0001
MentalAbility	0.4960	0	0.4960
	0.0850		0.0850
	5.8374		5.8374
	<.0001		<.0001

PROC CALIS also provides the standardized results for effect analysis. Standard errors, t-values, and p-values are also computed for the standardized effect estimates.

Standardized Effects of Parental Encouragement on Mental Ability

Standardized Effects of ParentalEncouragement
Effect / Std Error / t Value / p Value

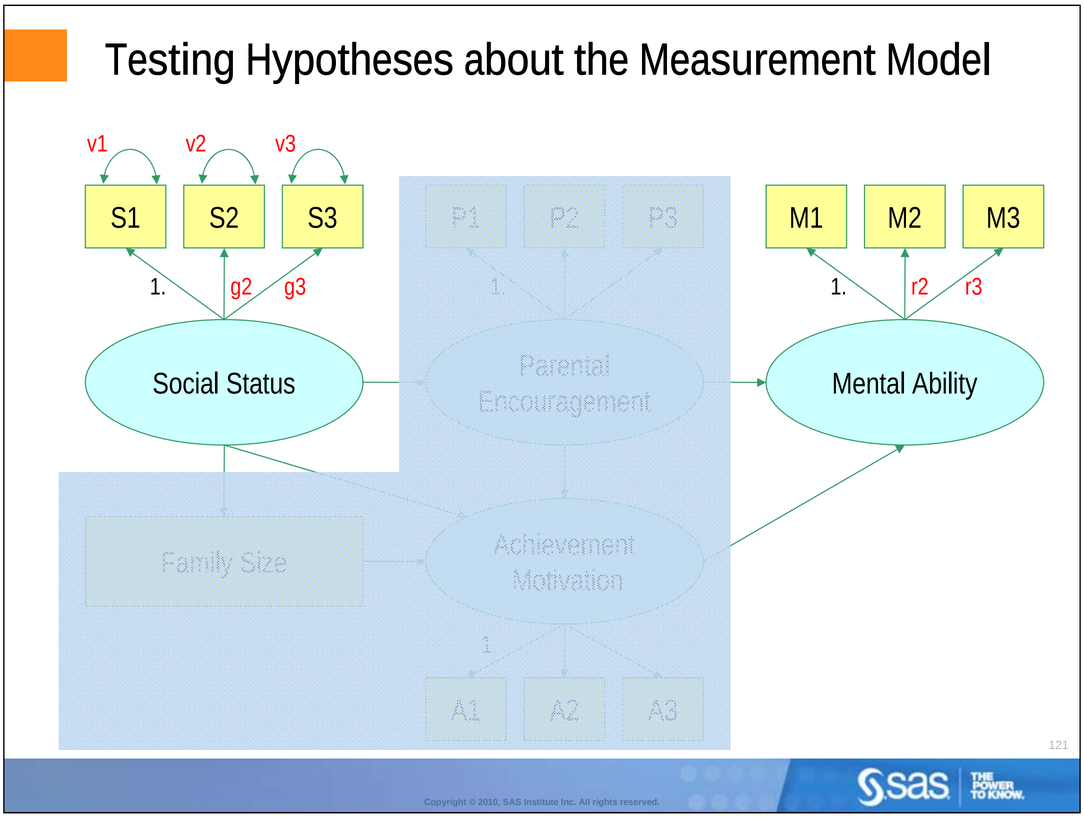
	Total	Direct	Indirect
MentalAbility	0.1487	-0.7639	0.9126
	0.1705	0.3096	0.3487
	0.8722	-2.4675	2.6171
	0.3831	0.0136	0.008869

Parental encouragement has a positive standardized total effect on the mental ability, although the standardized total effect is not significant.

119

This slide shows the standardized effects of parental encouragement on mental ability. The pattern is quite similar to the unstandardized version.

Testing Specific Hypotheses

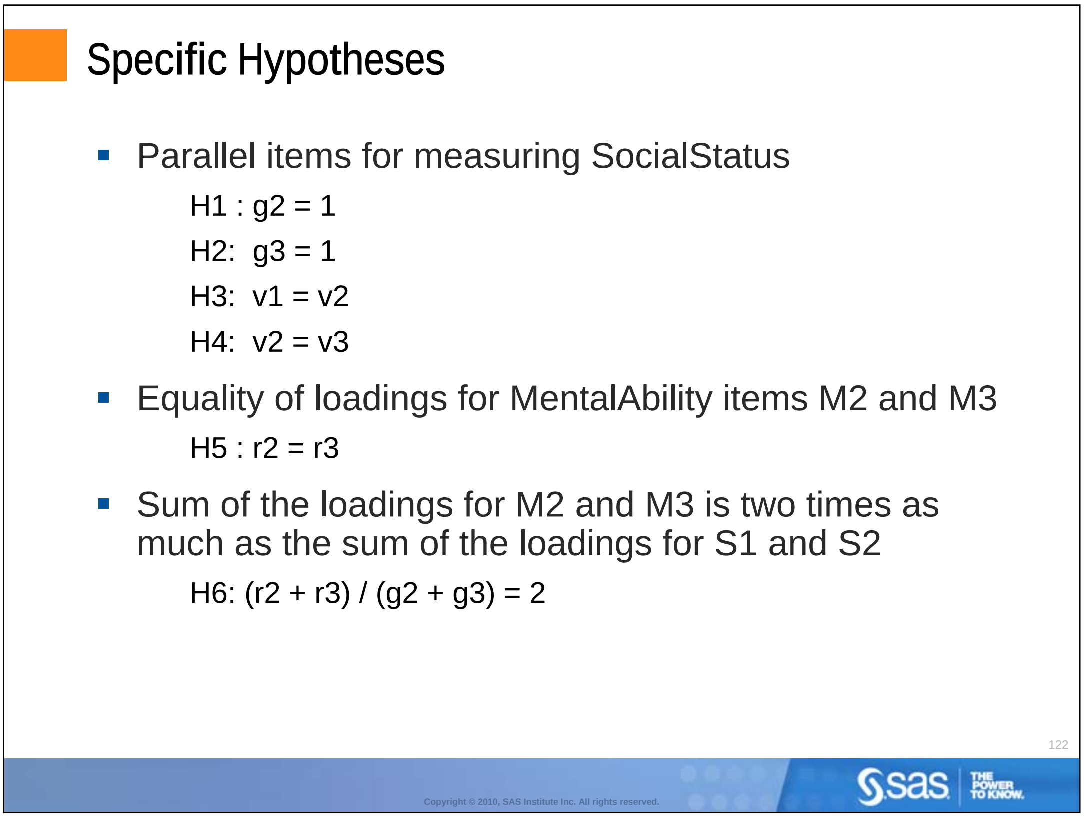


Testing specific hypotheses is an interesting topic. Here we look at some examples.

In the mental ability model, you have some indicator variables for the latent variables. Two latent variables are selected to illustrate the testing of specific hypotheses.

For the mental ability factor, one might want to test the hypothesis that the loadings (path coefficients) are the same for the M2 and M3 indicators. In the path diagram, r2 and r3 are labeled as the path coefficients. You want to test whether r2 and r3 are equal within the model.

For the social status factor, you not only want to test the hypothesis that the loadings (path coefficients) are the same for the three indicators, but you also want to see if their corresponding error variances are the same in the population. In the path diagram, g2, g3, v1, v2, and v3 are parameters of interest. You want to test simultaneously whether g2, g3 are equal to 1 and v1, v2, and v3 are the same in the population.



The test of equal loadings and equal error variances for the social status items is a test of parallel items. This could be stated more formally as the following four component hypotheses H1, H2, H3, and H4, as shown in the slide. These four hypotheses need to be tested simultaneously. Rejection of the simultaneous test means the items are not parallel.

The test of equal loadings for the measurement indicators of the mental ability factor is simpler. It is stated in H5. Rejection of H5 means that r2 and r3 are not equal in the population.

Finally, you can invent any strange hypothesis that can be expressed as a continuous function of the model parameters. For example, H6 states that the ratio of the sum of r2 and r3 to the sum of g2 and g3 is 2. This hypothesis may or may not make sense. But it is included here to demonstrate the flexibility of PROC CALIS.

PROC CALIS Hypotheses Testing: h(θ) = 0

- Parallel items for measuring SocialStatus:

  H1 : h1 = g2 — 1 = 0

  H2: h2 = g3 — 1 = 0

  H3 : h3 = v1 — v2 = 0

  H4: h4 = v2 — v3 = 0

- Equality of loadings for MentalAbility items M2 and M3

  H5 : h5 = r2 — r3 = 0

- Sum of the loadings for M2 and M3 is two times as much as the sum of the loadings for S1 and S2

  H6: h6 =2(g2 + g3) — (r2 + r3) = 0

123

Before I show you the PROC CALIS code, it is useful to reformulate the hypotheses into the forms that match the PROC CALIS input.

PROC CALIS tests hypotheses of the form h(θ)=0, where h(θ) is any continuous function of the model parameters (for example, the error variances and the path coefficients in the model).

The hypotheses in the previous slide could all be rewritten in this required form, as shown in this slide. With these forms, you are ready to specify those hypotheses in PROC CALIS.

Testing Specific Hypotheses about the Measurement Model Using PROC CALIS

```
proc calis data=mental nobs=115;
   path
      SocialStatus --->  ParentalEncouragement FamilySize AchievementMotivation,
      FamilySize   --->  AchievementMotivation,
      ParentalEncouragement  --->  AchievementMotivation MentalAbility,
      AchievementMotivation  --->  MentalAbility,
      SocialStatus           ---> S1 S2 S3   = 1. g2 g3,
      ParentalEncouragement  ---> P1 P2 P3   = 1.,
      AchievementMotivation  ---> A1 A2 A3   = 1.,
      MentalAbility          ---> M1 M2 M3   = 1. r2 r3;
   pvar S1-S3 = v1-v3;
   simtest parallel_social_items=[h1 h2 h3 h4];
   testfunc h5_equal_load_m2_m3 h6_proportional_sum;
   h1 = g2 - 1;
   h2 = g3 - 1;
   h3 = v1 - v2;
   h4 = v2 - v3;
   h5_equal_load_m2_m3 = r2 - r3;
   h6_proportional_sum = 2*(g2 + g3) - (r2 + r3);
run;
```

Specify g2, g3, r2, r3, v1, v2, and v3 explicitly.

Use the SIMTEST statement to test simultaneously hypotheses. Use the TESTFUNC statement to test individual hypotheses.

Use the SAS programming statements to define the parametric functions in the tests.

124

First, you have to label or name the parameters in the correct locations of the model specification. For example, g2 and g3 are the path coefficients for S2 and S3, respectively; and r2 and r3 are the path coefficients for M2 and M3, respectively. Notice that you did not name these parameters in the preceding model specifications. Naming these parameters were optional because you did not reference them. However, because you are going to refer to these parameters in the hypothesis testing, you must name or label them in the respective locations now. Similarly, the error variances for S1-S3 are named as v1-v3, as shown in the PVAR statement.

The main tools for testing specific hypothesis in PROC CALIS are the SIMTESTS and the TESTFUNC statements.

The SIMTEST statement enables you to test simultaneous hypotheses like the parallel hypothesis with four component hypotheses. Here we have h1, h2, h3, and h4, all of which are treated just as the names of the hypotheses that are defined later.

The TESTFUNC statement enables you to test individual hypotheses like the equality of loadings and the proportionality hypotheses described previously. Here I use long names such as h5_equal_load_m2_m3 and h6_proportional_sum to remind me of the nature of the target hypotheses.

Now I use the so-called SAS programming statements to define the hypotheses: h1-h4, h5_equal_load_m2_m3, and h6_proportional_sum. The SAS programming statements are just like common mathematical equations. These six SAS programming statements define the parametric functions in the target hypotheses. PROC CALIS tests all parametric functions equaling zero.

Individual Tests of Parametric Functions: TESTFUNC Results

Tests for Parametric Functions

Parametric Function	Value	Standard Error	t Value	p Value
h5_equal_load_m2_m3	0.04147	0.24290	0.17072	0.8644
h6_proportional_sum	-0.27995	1.02816	-0.27228	0.7854

Both individual hypotheses are supported.

125

The TESTFUNC specification produces the results shown in this table.

You fail to reject the equality of loadings for M2 and M3 because the p-value is bigger than 0.05. So, the equality of the loadings is supported.

You also fail to reject the proportional sum hypothesis (p-value=0.79).

Tests for Parallel Social Status Items: SIMTESTS Results

Simultaneous Tests

Simultaneous Test	Parametric Function	Function Value	DF	Chi-Square	p Value
parallel_social_items			4	24.23862	<.0001
	h1	0.19873	1	3.57013	0.0588
	h2	-0.25004	1	8.37521	0.0038
	h3	-0.24067	1	0.17774	0.6733
	h4	-0.64108	1	1.49312	0.2217

Overall parallelism hypothesis is not supported for the SocialStatus items, although the equality of error variances is supported.

126

The SIMTESTS statement specification produces the output shown in this table.

For the parallel hypothesis, the simultaneous test is rejected (p <.0001). The parallel item hypothesis is not supported.

PROC CALIS also provides individual tests for the component hypotheses. This would be useful for doing an ad hoc analysis to probe what fails the simultaneous hypothesis. For example, both h1 and h2 are at least marginally significant. But h3 and h4 are not significant. Recall that h1 and h2 are about the equality of the loadings (path coefficients) while h3 and h4 are about the equality of error variances. The current results show that the items might have the same error variances but not the same loadings in the population.

Model Modifications

127

When the Model Does Not Fit Well ...

Fit Summary

Chi-Square	196.7455
Chi-Square DF	59
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0936
Adjusted GFI (AGFI)	0.7341
RMSEA Estimate	0.1431
Bentler Comparative Fit Index	0.8087

- Large SRMSR and RMSEA
- Small AGFI and CFI
- Model modification: suggests ways to improve the model fit
- Lagrange multiplier (LM) tests: which parameters you can add to significantly decrease the model fit chi-square value

128

The mental ability model did not fit well. The SRMSR and the RMSEA are large, while the AGFI and the CFI are small. When you encounter a bad model fit, it would jeopardize your interpretations of the model parameters, effect analysis, hypothesis testing, and etc.

Model modification is a statistical technique that suggests ways to improve your model fit. The most common model modification technique is done through the so-called Lagrange multiplier (LM) tests. Essentially, the LM tests suggest which parameters you could add to the model to significantly lower the model fit chi-square value. When the model fit chi-square is lowered, most other fit indices (but not all, especially those parsimonious indices that take model complexity into account) might also improve.

Using the MODIFICATION Option

```
proc calis data=mental nobs=115 modification;
   path
      SocialStatus --->  ParentalEncouragement FamilySize
                         AchievementMotivation,
      FamilySize   --->  AchievementMotivation,
      ParentalEncouragement  --->  AchievementMotivation MentalAbility,
      AchievementMotivation  --->  MentalAbility,
      SocialStatus           ---> S1 S2 S3   = 1. ,
      ParentalEncouragement  ---> P1 P2 P3   = 1.,
      AchievementMotivation  ---> A1 A2 A3   = 1.,
      MentalAbility          ---> M1 M2 M3   = 1.;
run;
```

129

The option you can use to do model modification in PROC CALIS is the MODIFICATION option in the PROC CALIS statement. You can simply add this option to the PROC CALIS statement when you run your model. This example shows that the LM tests for model modification is requested for the original mental ability model, which does not have a very good model fit.

LM Tests for Paths

Rank Order of the 10 Largest LM Stat for Path Relations

To	From	LM Stat	Pr > ChiSq	Parm Change
P2	P1	56.19414	<.0001	-0.73639
P1	P2	56.19349	<.0001	-0.72904
A2	M2	19.17647	<.0001	0.22842
A2	ParentalEncouragement	18.57947	<.0001	-2.31463
A2	MentalAbility	17.20340	<.0001	0.95581
ParentalEncouragement	A1	17.04464	<.0001	0.27042
A1	ParentalEncouragement	15.86099	<.0001	1.88904
FamilySize	A2	14.43548	0.0001	-1.14590
A1	MentalAbility	13.88705	0.0002	-0.75314
A2	P3	12.96818	0.0003	-0.57151

Adding the P2 <--- P1 (or P1 <--- P2) path reduces your model fit chi-square by 56 approximately.
Adding the A2 <--- M2 path reduces your model fit chi-square by 19 approximately.

130

PROC CALIS output several tables for the LM tests. The results are shown in different tables, according the type of the parameters. This table shows the ranking of LM statistics for adding the (single-headed) paths into the mental ability model. It gives you the ten paths that can improve the model fit chi-square statistic the most.

The top one is the p1 ---> p2 path. The LM statistic 56.19 means that if you include this path into the model, you can expect to reduce the model fit chi-square by about 56. This is a substantial improvement because you can get this big improvement by just losing one degree of freedom. The second one is the p2 ---> p1 path. Essentially, this will give the same amount of model improvement as the first path. The third one is not that dramatic, but still give you a substantial improvement. Adding the M2 ---> A2 path reduces the model fit chi-square by 19.

Do you want to add these paths into your model? Let us discuss this after we examine more results about the LM tests.

LM Tests for Error Variances and Covariances

Rank Order of the 10 Largest LM Stat for Error Variances and Covariances

Error of	Error of	LM Stat	Pr > ChiSq	Parm Change
P2	P1	56.19312	<.0001	-1.96473
ParentalEncouragement	A1	12.26622	0.0005	0.46050
ParentalEncouragement	A2	12.08031	0.0005	-0.48351
FamilySize	A2	11.22650	0.0008	-1.88205
M2	A2	10.26408	0.0014	1.55895
S2	S1	7.78117	0.0053	1.55314
MentalAbility	A2	7.48800	0.0062	0.78161
AchievementMotivation	A1	6.95709	0.0083	-0.52904
P2	A3	6.54315	0.0105	0.76007
A3	A2	6.21429	0.0127	-0.67173

Adding the error covariance (P2 <---> P1) reduces your model fit chi-square by 56 approximately.

131

This table shows the LM tests (statistics) for the error variances and covariances. On the top of the list is the covariance between the errors of P2 and P1. Adding the covariance between the errors of these two variables reduces the model fit chi-square statistic by 56. This is actually the same improvement that we have seen for adding either the P2 ---> P1 or
P1 ---> P2 path. The next one in the list has a much less improvement. The LM statistic is only 12.26.

For this particularly model, these two tables are all that PROC CALIS produces for the LM statistics. The question now is which parameter or parameters you want to add to the model. This could not be answered by just looking at the LM statistics. But it might also involve some judgment about how reasonable the added parameters are. Do these added parameters render your model un-interpretable, or even contradictory to your theoretical claims, despite the fact that they improve your model fit substantially?

Notes on the LM Statistics

- Chi-square reductions are linear approximations
- Chi-square reductions are not additive
- Modifications suggested might not be substantively meaningful

132

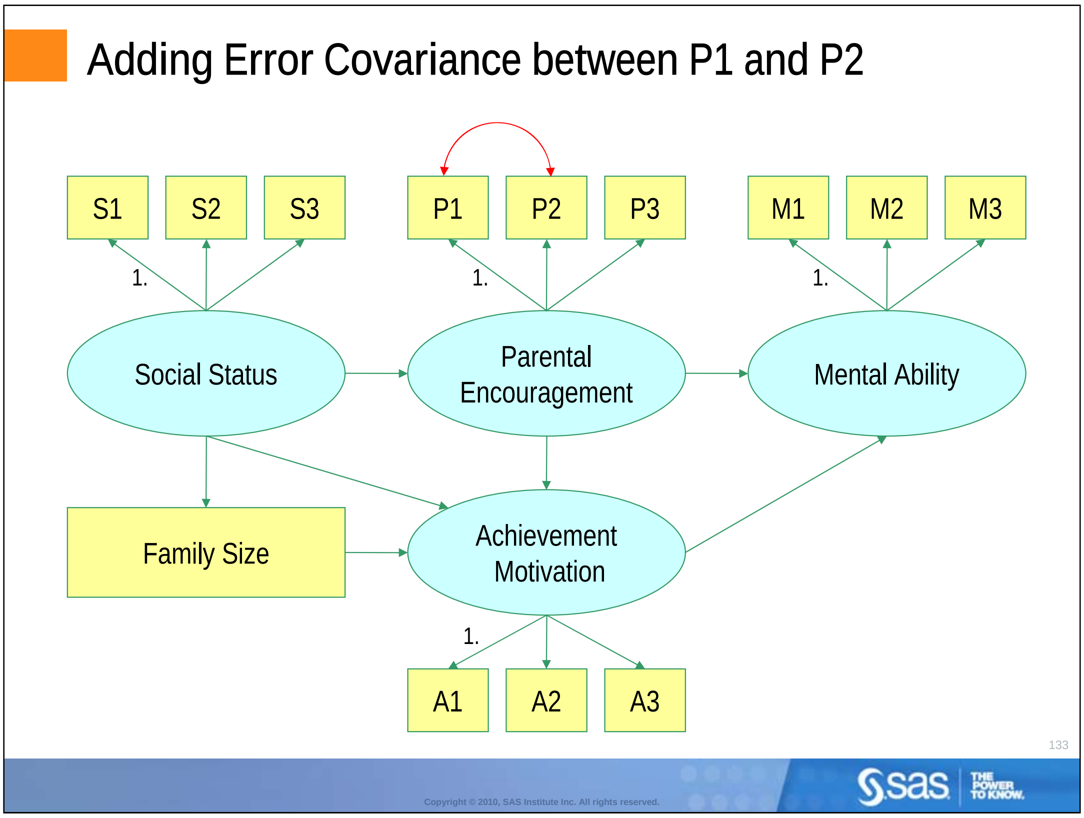
Before giving an answer to the current model modification analysis, some important general points about the LM statistics are discussed.

First, the model fit chi-square reductions as indicated by the LM test statistics are only linear approximations. This means that if you actually refit the model by adding the suggested parameter, the actual chi-square reduction might be more or less.

Second, the chi-square reductions as suggested by the LM test statistics are not additive. That means that you cannot add two or more parameters into the model and expect the actual reduction in the new model is exactly the sum of the corresponding LM statistics. Usually, the actual reduction would be smaller (although it could be larger).

Last but not least, modification suggested by the LM statistics might not be substantively meaningful.

All these three points are important in deciding which parameter you want to add to the current mental ability model for improving the model fit.



Considering the top suggestions from the results of the LM test statistics, I would add the covariance between P1 and P2. The added parameter is shown in the path diagram.

Basically, all the top LM suggestions --- the P2 ---> P1 and P1 ---> P2 paths, and the error covariance between P1 and P2 are just different manifestations of the same lack of fit about a covariance element in the original model. That is, the covariance between P1 and P2 was not well-explained by the original model. Adding either of these will lead to a better fitting of the covariance between P1 and P2. In addition, adding either of these will give you an approximate model fit chi-square improvement of 56. But, you would not get three times of this amount by adding all these three. In fact, if you were to add all these three parameters, it is very likely that your model is not identified, meaning that you would not get unique estimates.

Among the top three choices, the error covariance is chosen because the interpretation of added error covariance is a little “cleaner.” P1 and P2 are measurement indicators of the same factor (Parental Encouragement). The error covariance interpretation is that these two indicators have some sort of correlation that is unexplained by their common factor. The added error covariance represents the covariance explained by some unknown sources. However, if I were to add either the P1 ---> P2 or P2 ---> P1 paths, it would create some conflicts with purported common factor structure for the two indicator variables.

Note that the current conclusion is based on a very general argument that aims at preserving the original factor-variable structure. It is not a universal principle. In practice, you have to also consider the substantive grounds of the added parameters.

Adding Covariance between the Errors of P1 and P2

```
proc calis data=mental nobs=115 modification;
   path
      SocialStatus ---> ParentalEncouragement FamilySize
                        AchievementMotivation,
      FamilySize   ---> AchievementMotivation,
      ParentalEncouragement  ---> AchievementMotivation MentalAbility,
      AchievementMotivation  ---> MentalAbility,
      SocialStatus           ---> S1 S2 S3   = 1. ,
      ParentalEncouragement  ---> P1 P2 P3   = 1.,
      AchievementMotivation  ---> A1 A2 A3   = 1.,
      MentalAbility          ---> M1 M2 M3   = 1.;
   pcov P1 P2;
run;
```

134

Now that I have decided to add the covariance between P1 and P2, I refit the model by adding the PCOV statement specification for the two variables, as shown in the SAS code in this slide. I also use the MODIFICATION option one more time to see if there could be any further suggested improvements.

Before and After Adding the Error Covariance between P1 and P2

Before …

Fit Summary

Chi-Square	196.7455
Chi-Square DF	59
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0936
Adjusted GFI (AGFI)	0.7341
RMSEA Estimate	0.1431
Bentler Comparative Fit Index	0.8087

After…

Fit Summary

Chi-Square	110.6388
Chi-Square DF	58
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0661
Adjusted GFI (AGFI)	0.8062
RMSEA Estimate	0.0892
Bentler Comparative Fit Index	0.9269

Improve the model fit chi-square a lot more than 56.

135

Before you add the error covariance between P1 and P2, your model fit chi-square was about 197. After adding the covariance, the model fit chi-square is about 111. This improvement is actually larger than what the LM statistic suggested, which was 56.

Other fit indices also improve. The SRMSR and the RMSEA are now close to be acceptable. The AGFI and the CFI are boosted to higher levels.

A New Set of LM Tests for Paths

Rank Order of the 10 Largest LM Stat for Path Relations

To	From	LM Stat	Pr > ChiSq	Parm Change
A2	M2	23.93741	<.0001	0.23830
A2	ParentalEncouragement	22.22260	<.0001	-2.30398
A2	MentalAbility	21.70998	<.0001	0.90790
A1	ParentalEncouragement	19.34281	<.0001	1.96513
A1	MentalAbility	18.10545	<.0001	-0.75752
ParentalEncouragement	A1	17.30632	<.0001	0.32504
A1	M2	15.06992	0.0001	-0.17617
A2	FamilySize	15.06786	0.0001	-0.17675
FamilySize	A2	14.29265	0.0002	-0.89658
AchievementMotivation	A1	11.75569	0.0006	-0.35962

Adding the A2 <--- M2 path now reduces your model fit chi-square by 24 (was 19 before adding the error covariance).

136

The new set of LM tests for paths suggests the addition of the M2 ---> A2 path. The LM statistic is about 24. If you compare this result with the first LM results regarding the same path, you notice that the LM statistics changes as the fitted model changes. Previously, the same path had an LM statistic of 19. This illustrates the nonlinearity and non-additivity of the LM statistics.

A New Set of LM Tests for Error Variances and Covariances

Rank Order of the 10 Largest LM Stat for Error Variances and Covariances

Error of	Error of	LM Stat	Pr > ChiSq	Parm Change
ParentalEncouragement	A2	14.11717	0.0002	-0.53966
ParentalEncouragement	A1	14.00823	0.0002	0.51242
FamilySize	A2	13.96827	0.0002	-1.98162
M2	A2	11.86015	0.0006	1.65462
AchievementMotivation	A1	11.75570	0.0006	-0.57402
MentalAbility	A2	10.30441	0.0013	0.86542
P1	A1	9.02987	0.0027	0.53357
S2	S1	8.94243	0.0028	1.66876
MentalAbility	FamilySize	8.05365	0.0045	2.49300
MentalAbility	AchievementMotivation	6.94045	0.0084	0.98947

137

There is also a new set of LM tests for adding error covariances.

You might want to improve your model further by adding some parameters from these two LM tables, although I will not attempt to do more here.

Customized LM Tests

- Principled modification process
- Restrict the set of parameters of interest for the LM tests

Model modification by using the MODIFICATION option is kind of “blind-search” procedure that you try to improve your model without any definite directions. As discussed before, the LM test statistics might not give you suggestions that are substantively meaningful.

However, in some occasions you might want to restrict your attention to certain set of potential paths or parameters in your model, rather than all possible parameter space searched by the MODIFICATION option.

If you want to do such a principled modification process, you can use the customized LM tests supported in PROC CALIS.

Customized LM Tests by Using the LMTESTS Statement

```
proc calis data=mental nobs=115;
   path
      SocialStatus ---> ParentalEncouragement FamilySize AchievementMotivation,
      FamilySize   ---> AchievementMotivation,
      ParentalEncouragement  ---> AchievementMotivation MentalAbility,
      AchievementMotivation  ---> MentalAbility,
      SocialStatus           ---> S1 S2 S3   = 1. ,
      ParentalEncouragement  ---> P1 P2 P3   = 1.,
      AchievementMotivation  ---> A1 A2 A3   = 1.,
      MentalAbility          ---> M1 M2 M3   = 1.;
   lmtests corr_err=[coverr] path=[LV->LV LV->MV];
run;
```

Explore the set of LM tests called "corr_err," which contains all the potential error covariance parameters (COVERR) to be freed.

Explore the set of LM tests called "path," which contains all potential latent variable paths (LV-> LV) and measurement paths (LV -> MV) to be freed.

139

The customized LM tests define sets of parameters of interest so that your model modification process (or LM statistics output) would be limited to those sets of parameters. PROC CALIS provides the LMTESTS statement syntax to achieve the customized LM tests.

The mental ability model is used again. This time I define two sets of parameters of interest. The first set of LM tests is called "corr_err" (it is just a name you assign). This set of parameters contains the parameter region COVERR, which is a keyword that denotes all error covariances in the model. The second set of LM tests is called "path"—a name you assign. This set of parameters do not exhaust all paths in the model. It contains the parameter regions LV->LV and LV->MV, which are keywords that denotes the latent variable (LV) to latent variable (LV) paths and the latent variable (LV) to manifest variable (MV) paths, respectively. Therefore, this customized set "path" excludes paths from observed variables to observed variables, or from observed variables to latent variables so that the factor structures of the model could not be potentially destroyed by adding these paths. The LM tests for these paths are simply not included in the results for the "path" set.

Customized LM Tests for Error Covariances

Rank Order of the 10 Largest LM Stat for Set corr_err

Type	Var1	Var2	LM Stat	Pr > ChiSq	Parm Change
COVERR	P2	P1	56.19312	<.0001	-1.96473
COVERR	ParentalEncouragement	A1	12.26622	0.0005	0.46050
COVERR	ParentalEncouragement	A2	12.08031	0.0005	-0.48351
COVERR	FamilySize	A2	11.22650	0.0008	-1.88205
COVERR	M2	A2	10.26408	0.0014	1.55895
COVERR	S2	S1	7.78117	0.0053	1.55314
COVERR	MentalAbility	A2	7.48800	0.0062	0.78161
COVERR	AchievementMotivation	A1	6.95709	0.0083	-0.52904
COVERR	P2	A3	6.54315	0.0105	0.76007
COVERR	A3	A2	6.21429	0.0127	-0.67173

140

This table shows the customized LM tests of the “CORR_ERR” set. Essentially, this table is the same as one of the standard tables produced with the MODIFICATION option because both tables have the same parameter region “COVERR.”

Customized LM Tests for Paths

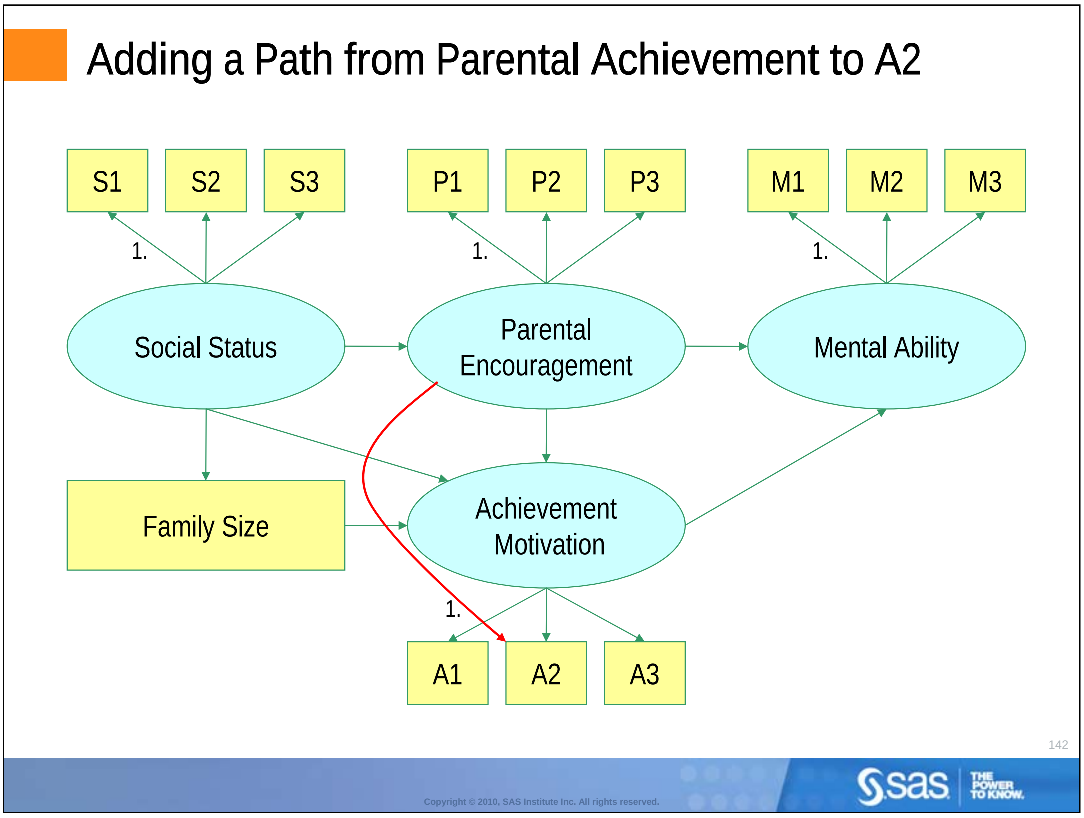
Rank Order of the 10 Largest LM Stat for Set path

Type	Var1	Var2	LM Stat	Pr > ChiSq	Parm Change
DV_DV	A2	ParentalEncouragement	18.57947	<.0001	-2.31463
DV_DV	A2	MentalAbility	17.20340	<.0001	0.95581
DV_DV	A1	ParentalEncouragement	15.86099	<.0001	1.88904
DV_DV	A1	MentalAbility	13.88705	0.0002	-0.75314
DV_DV	S2	MentalAbility	8.97262	0.0027	-0.38910
DV_DV	S3	AchievementMotivation	6.27192	0.0123	0.32436
DV_DV	S2	AchievementMotivation	6.19233	0.0128	-0.40928
DV_DV	ParentalEncouragement	MentalAbility	5.57439	0.0182	0.29376
DV_DV	ParentalEncouragement	AchievementMotivation	5.33788	0.0209	0.39218
DV_DV	FamilySize	AchievementMotivation	5.33730	0.0209	-1.20671

The measurement path A2 <--- ParentalEncouragement reduces the model fit chi-square by 19.

141

The second customized set of LM tests suggests that adding the dependent variable (DV) to dependent variable (DV) path A2 <--- ParentalEncouragement improves the model fit the most amongst all paths in the "path" set. The chi-square improvement is about 19, which is statistically significant.



Adding the first path suggested by the second set of customized set is represented by the path diagram shown above. The path in red shows that A2, which is an indicator of Achievement Motivation, is now also an indicator of Parental Encouragement. Although the factor-variable functional relationship is preserved in this suggested path diagram, A2 becomes factorially-complex. The also implies that A2 might not have been a good (unique) measure of achievement motivation.

Adding the ParentalEncouragement ---> A2 path

```
proc calis data=mental nobs=115;
   path
      SocialStatus ---> ParentalEncouragement FamilySize AchievementMotivation,
      FamilySize   ---> AchievementMotivation,
      ParentalEncouragement  ---> AchievementMotivation MentalAbility,
      AchievementMotivation  ---> MentalAbility,
      SocialStatus           ---> S1 S2 S3     = 1. ,
      ParentalEncouragement  ---> P1 P2 P3 A2  = 1.,
      AchievementMotivation  ---> A1 A2 A3     = 1.,
      MentalAbility          ---> M1 M2 M3     = 1.;
run;
```

143

Nonetheless, you add this new path for A2, as shown in the above PROC CALIS code. All you need to do is to add A2 as one of the observed indicators of the ParentalEncouragement factor.

Before and After Adding the ParentalEncouragement ---> A2 Path

Before …

Fit Summary

Chi-Square	196.7455
Chi-Square DF	59
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0936
Adjusted GFI (AGFI)	0.7341
RMSEA Estimate	0.1431
Bentler Comparative Fit Index	0.8087

After…

Fit Summary

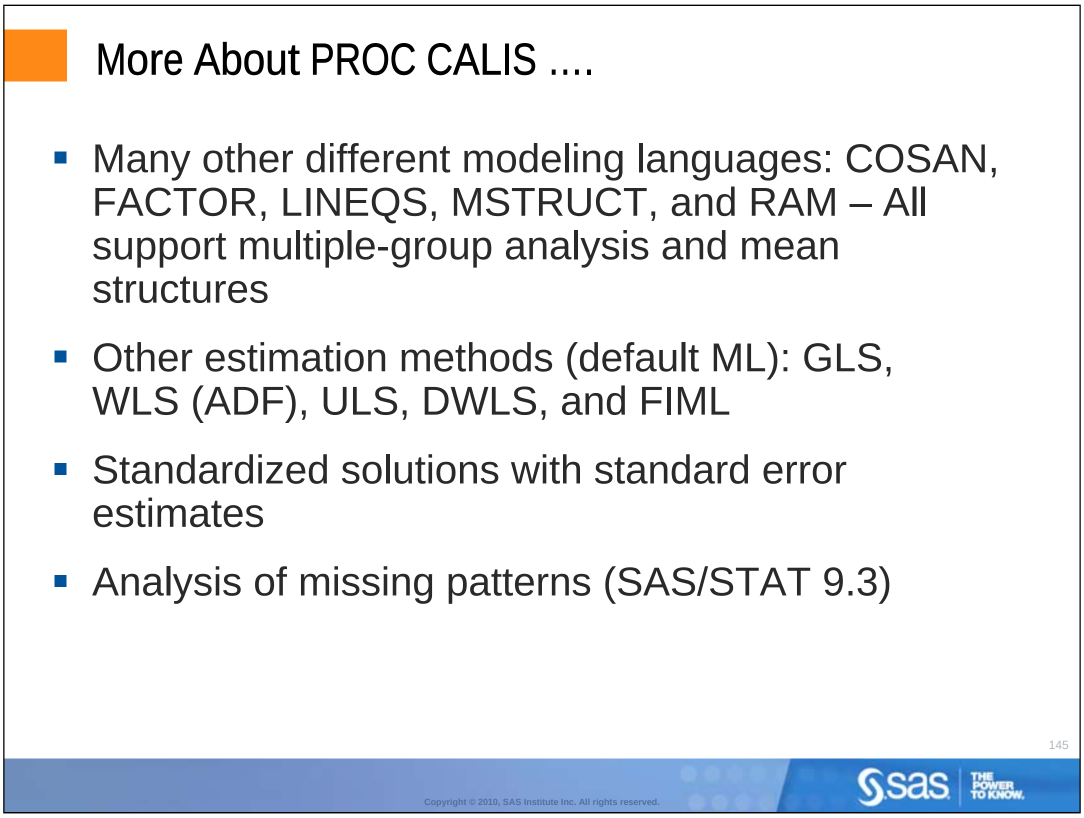
Chi-Square	168.2618
Chi-Square DF	58
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0896
Adjusted GFI (AGFI)	0.7675
RMSEA Estimate	0.1291
Bentler Comparative Fit Index	0.8468

The model fit improves quite a bit.

144

These two tables compare the fit indices before and after adding the ParentalEncouragement ---> A2 path. The model fit chi-square actually drops more than 19, which was suggested by the LM statistic in the preceding results. All other fit indices improve quite a bit too.

Finally, a caution about all model modification: you should validate your newly-established model by new data. The reason is that the model modification process is subject to the capitalization on chance. Using a principled modification process by the customized LM tests might not avoid the chance problem completely. Confirmation from new data is always recommended.



In this workshop, I mostly use the PATH modeling language to fit SEM. I also briefly mentioned the LISMOD as an interface for the LISREL model. There are actually quite a few more modeling language in PROC CALIS: COSAN, FACTOR, LINEQS, MSTRUCT, and RAM. All of these languages support multiple-group analysis and mean structure analysis.

I have also used the default ML (maximum likelihood) estimation method in this workshop, but PROC CALIS supports many other estimation methods as well: GLS (generalize least squares), WLS (weighted least squares), ULS (unweighted least squares), DWLS (diagonally-weighted least squares), and FIML (full information maximum likelihood).

I have only used unstandardized results in most examples, but PROC CALIS also provide standardized solutions with standard error estimates.

Finally, analysis of missing patterns will be available with the FIML method in SAS/STAT 9.3. I hope to add more functionalities to PROC CALIS in the future.

Glossary

Manifest – Observed variables (measured variables) in the data set.
Latent – Unobserved variables.
Endogenous – Dependent /mediating variables; at least one single-headed arrow points to it; used as an outcome variable in an equation; can also be a predictor variable in other equations.
Exogenous – Independent variables; no single-headed arrows point to it; never used as an outcome variable in the model; used only as a predictor in the model.
Factor – A latent (unmeasured) variable that is treated as a hypothetical construct (systematic source) in the model.
Error – An exogenous term for uncertainty (unsystematic source) associated with an endogenous *manifest* variable (or any endogenous variable, in a more general definition).
Disturbance – An exogenous term for uncertainty (unsystematic source) associated with an endogenous *latent* variable.
Path diagram representation
– Rectangles: Observed / manifest variables.
– Ovals / circles : Latent variables (factors, errors, and disturbances). Errors and disturbances are not necessarily put into ovals/circles.

146

Glossary

– Single-headed arrows: Directed paths, direct effects, path coefficients; specified in the PATH statement.
– Double-headed arrows that point to individual variables: Variance parameters of exogenous variables or error variance parameters of endogenous variables; specified in the PVAR statement.
– Double-headed arrows that point to two distinct variables: Covariance parameters between exogenous variables or error covariance parameters between endogenous variables; specified in the PCOV statement.

Fit assessment

– model fit chi-square statistic: Nonsignificance means that the theoretical model is supported; not a very practical index because it almost always rejects all approximating models that are practically useful.
– AGFI (adjusted goodness-of-fit index) and Bentler's CFI (comparative fit index): Two popular fit indices that indicate good model fit when their values are above 0.9.
– SRMR (standardized root mean square residual) and RMSEA (root mean squared error approximation): Two popular fit indices that indicate good model fit when their values are below 0.05.
– AIC, CAIC, and SBC: Information criteria for comparing competing models. The smaller the better.

References

Arbuckle, J. (2008), *AMOS 17.0 User's Guide*, Crawfordville, FL: Amos Development Corporation.

Bentler, P. M. (1985), *Theory and Implementation of EQS: A Structural Equations Program*, Manual for Program Version 2.0, Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1995), *EQS, Structural Equations Program Manual*, Program Version 5.0, Encino, CA: Multivariate Software.

Bollen, K. A. (1989), *Structural Equations with Latent Variables*, New York: John Wiley & Sons.

Fuller, W. A. (1987), *Measurement Error Models*, New York: John Wiley and Sons.

Holzinger, K. J. and Swineford, F. A. (1939), "A Study in Factor Analysis: The Stability of a Bi-Factor Solution," *Supplementary Educational Monographs, No. 48. Chicago: University of Chicago*,Dept. of Education.

Jöreskog, K. G. (1973), "A General Method for Estimating a Linear Structural Equation System," in A. S. Goldberger and O. D. Duncan, eds., *Structural Equation Models in the Social Sciences*, New York: Academic Press.

References

Keesling, J. W. (1972), *Maximum Likelihood Approaches to Causal Analysis*, Ph.D. thesis, University of Chicago.

Marjoribanks, K., ed. (1974), Environments for Learning, London: National Foundation for Educational Research Publications.

McDonald, R. P. (1978), "A Simple Comprehensive Model for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 31, 59–72.

McDonald, R. P. (1980), "A Simple Comprehensive Model for the Analysis of Covariance Structures: Some Remarks on Applications," *British Journal of Mathematical and Statistical Psychology*, 33, 161–183.

Nowak, T. P., Hoffman, D. L., and Yung, Y. F. (2000), "Measuring the Customer Experience in Online Environments: A Structural Modeling Approach," *Marketing Science*, *19(1)*, 22–42.

Wiley, D. E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in A. S. Goldberger and O. D. Duncan, eds., *Structural Equation Models in the Social Sciences*, New York: Academic Press.