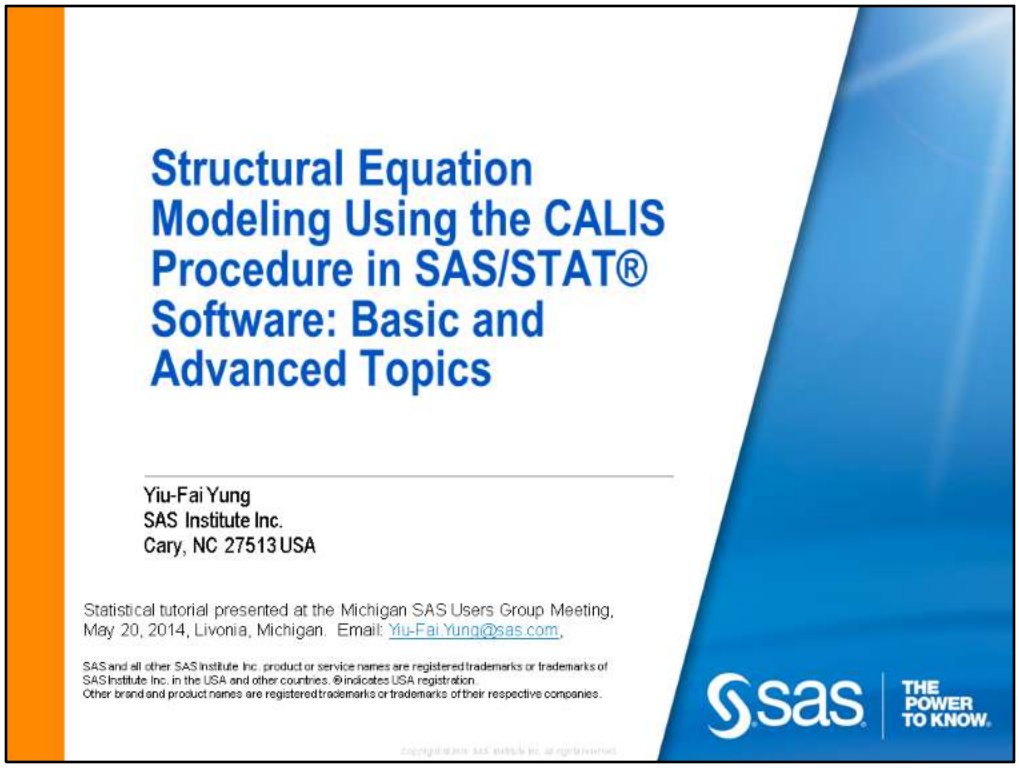
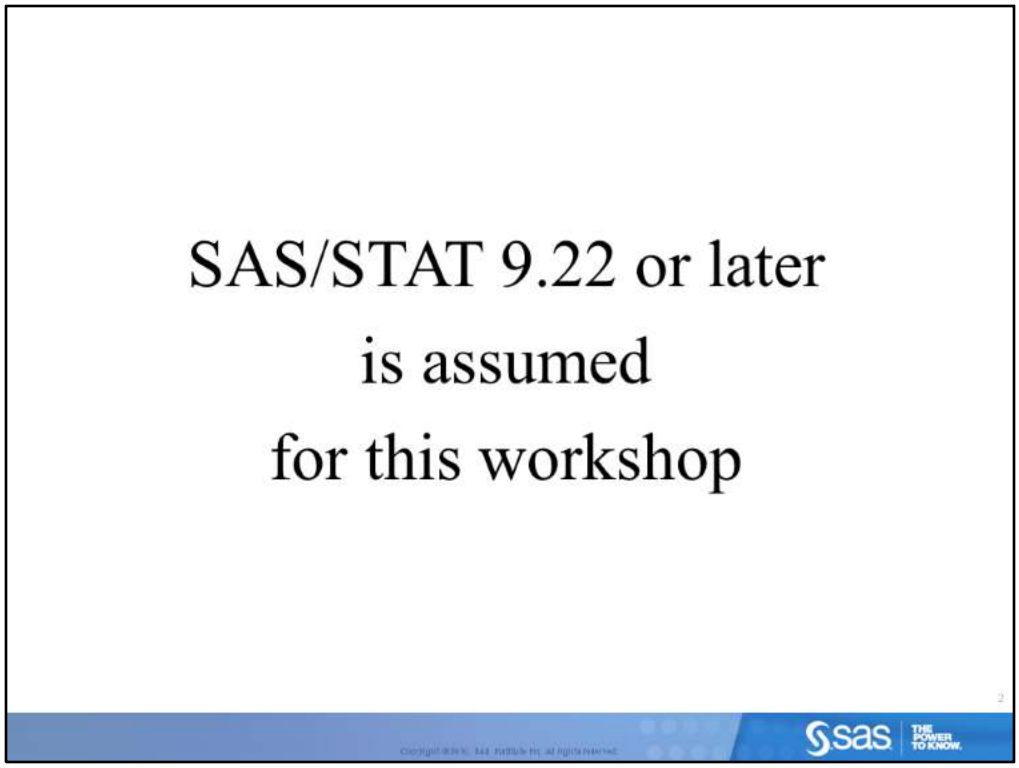


Abstract

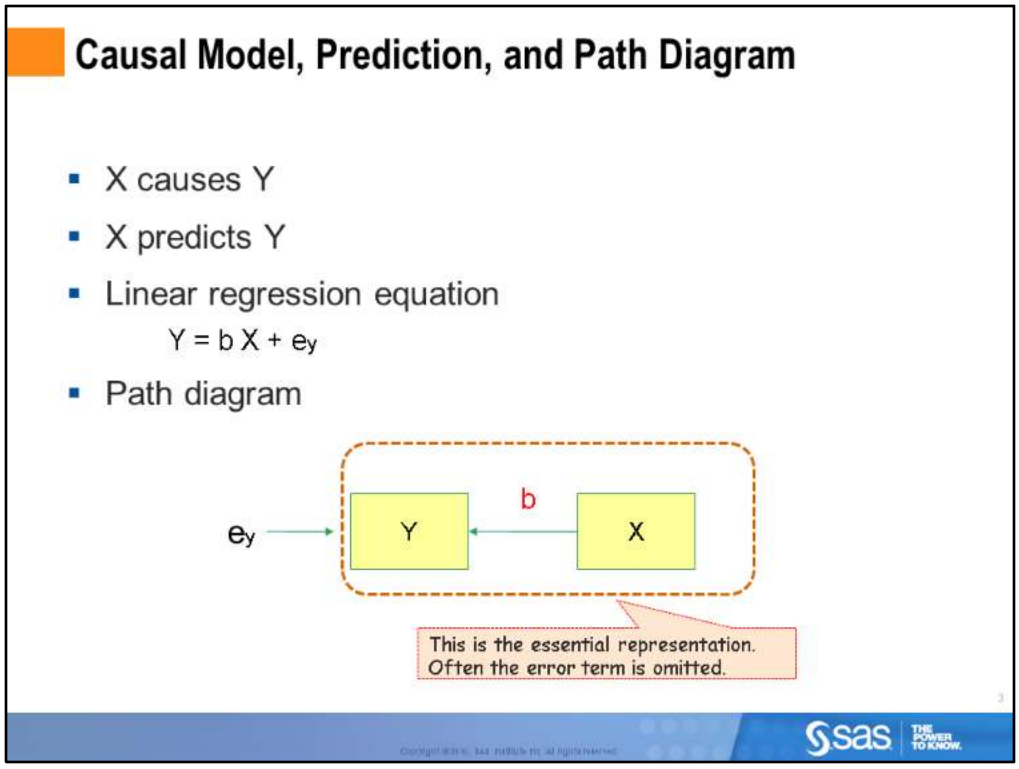
The CALIS procedure in SAS/STAT software is a general structural equation modeling (SEM) tool. This workshop introduces the general methodology of SEM and the applications of PROC CALIS. Background topics such as path analysis, confirmatory factor analysis, measurement error models, and linear structural relations (LISREL) are reviewed. Applications are demonstrated with examples in social, educational, behavioral, and marketing research. More advanced SEM techniques such as the full information maximum likelihood (FIML) method for treating incomplete observations, robust estimation, and diagnostics for outliers and leverage points in the SEM context are also covered.

This workshop is designed for statisticians and data analysts who want an overview of SEM applications using the CALIS procedure in SAS/STAT 9.22 and later releases. Attendees should have a basic understanding of regression analysis and experience using the SAS language. Previous exposure to SEM is useful but not required. Attendees will learn how to use PROC CALIS for (1) specifying structural equation models with latent variables, (2) interpreting model fit statistics and estimation results, (3) using the FIML method for treating incomplete observations, (4) and detecting outliers and leverage points.

Citation of this workshop and notes:
Yung, Y.-F. (2014). Structural Equation Modeling Using the CALIS Procedure in SAS/STAT® Software: Basic and Advanced Topics. Statistical tutorial presented at the Michigan SAS Users Group Meeting, May 20, 2014.



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 extended 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. PROC TCALIS also does not support FIML, residual diagnostics, and robust estimation.

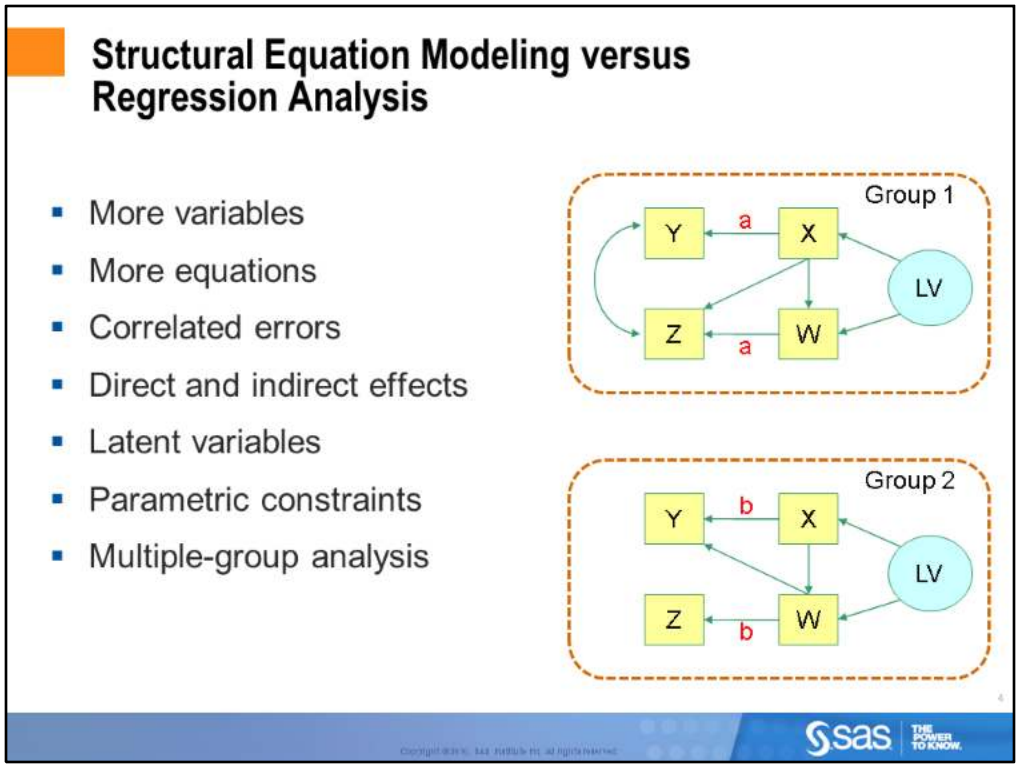


The central idea of structural equation modeling is the study of causal relationship between variables. For example, you have an X and a Y variable. X is the cause of Y, or doing X results in Y. To give a more realistic example: eating more vegetables (X) brings down your cholesterol level (Y). However,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

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 model, as shown in the linear regression equation:

Y = b*X + e.

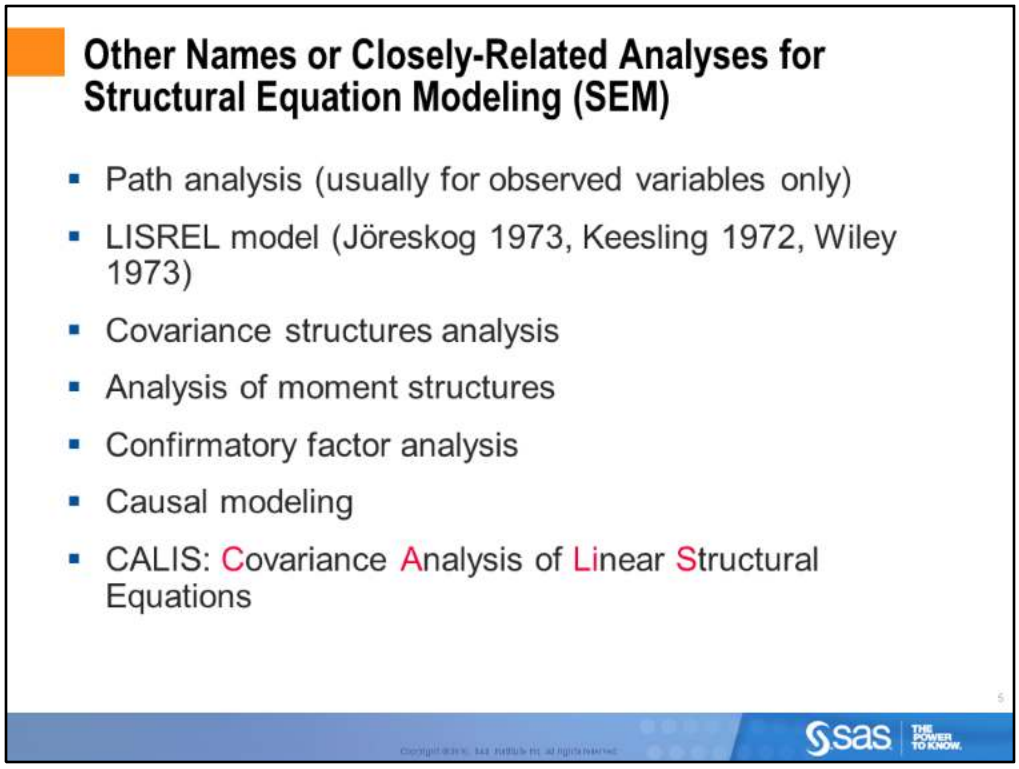
The path diagram for this representation is also shown in the slide, where b is called the effect, regression coefficient, or path coefficient. Notice that an error term is added to show that the prediction of Y from X is not perfect, which is usually true in practice. Essentially, the predictor-outcome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



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:

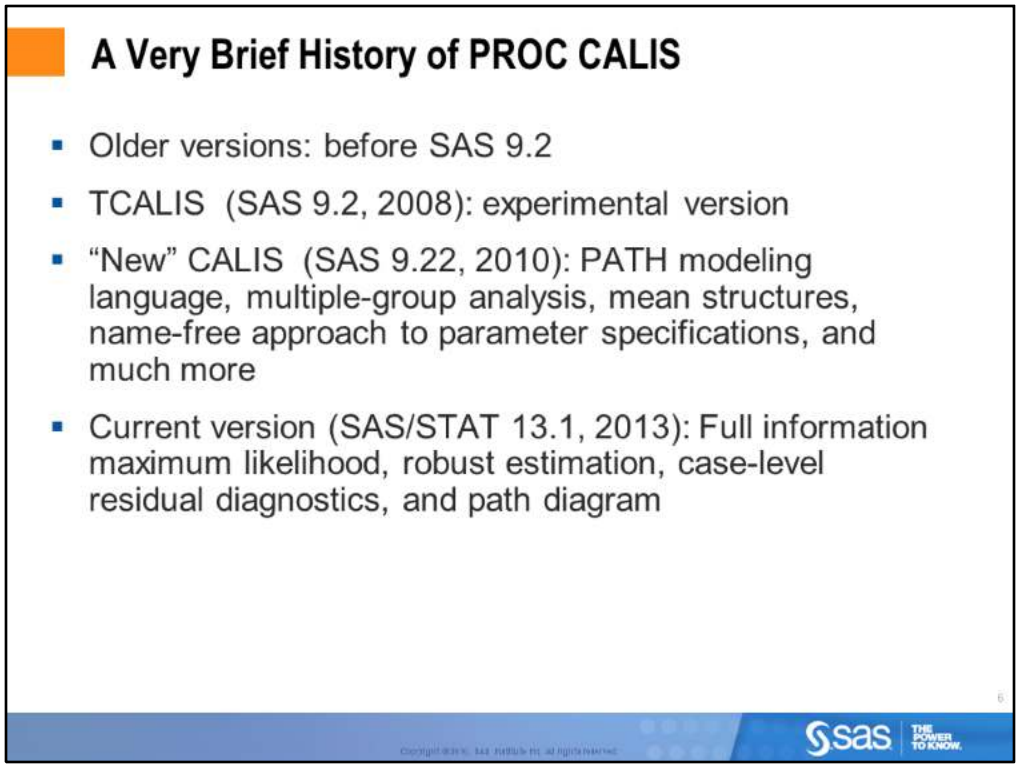
1. More variables (not just X and Y, but you can also add W and Z into the path diagram).
2. More equations or functional relationships (not just X→Y, but you can also analyze W→Z simultaneously).
3. Correlated errors: system of equations can have correlated errors . For example, the double-headed arrow 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 of X on Y, respectively.
5. Latent variables. For example, the latent variable LV in the path diagram has effects on X and W. In SEM, latent variables are represented by ovals or circles, while observed variables are represented by rectangles or squares.
6. Parametric constraints. For example, two path coefficients or effects labeled as 'a' in the upper path diagrams are constrained to be equal.
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.



SEM has a lot of synonyms (or closely-related statistical techniques) 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 techniques 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.

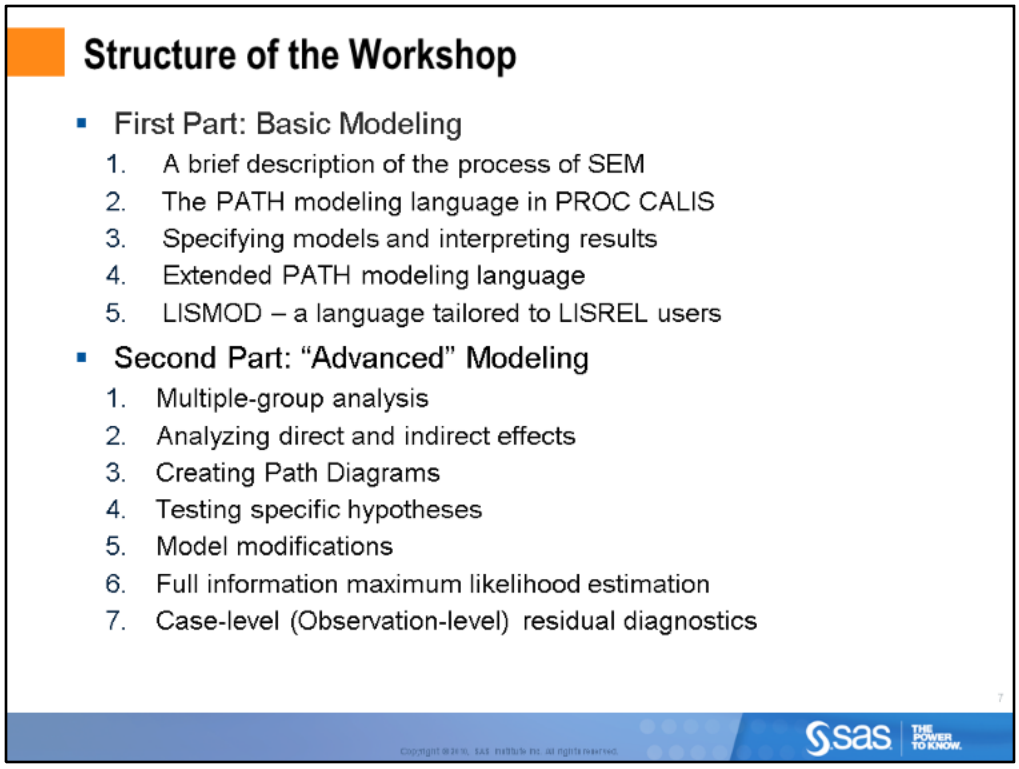
Hopefully, one day PROC CALIS would also be remembered as a synonym of SEM.



Let us start with a brief history of PROC CALIS. In the 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 am emulation of the syntax of the EQS program.

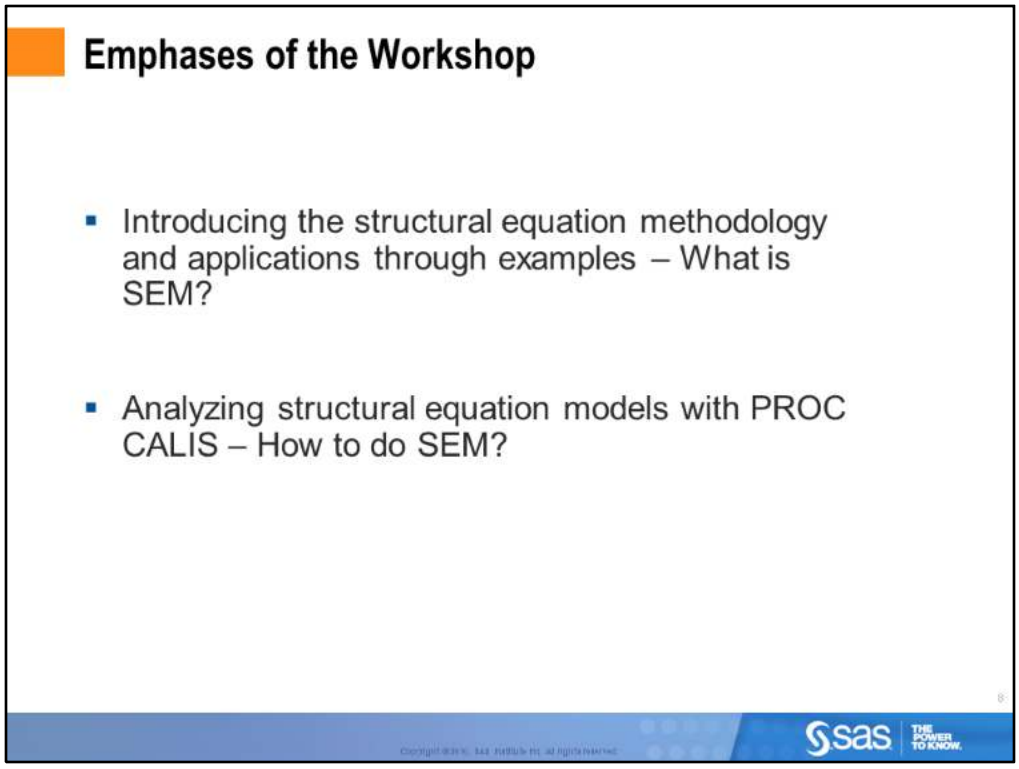
I (Yiu-Fai Yung) picked up the development of the software around 2000. I actually rewrote the mathematical foundations of the software. I kept the optimization techniques and initial estimation techniques so that the estimation results of "new" CALIS is compatible with the "old" CALIS. In 2008, an experimental version called TCALIS was released. Since then, I have modified the syntax a little more, fixed some major bugs, and added some new features.

The SAS 9.22 version of the CALIS procedure was released in 2010. 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.



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.

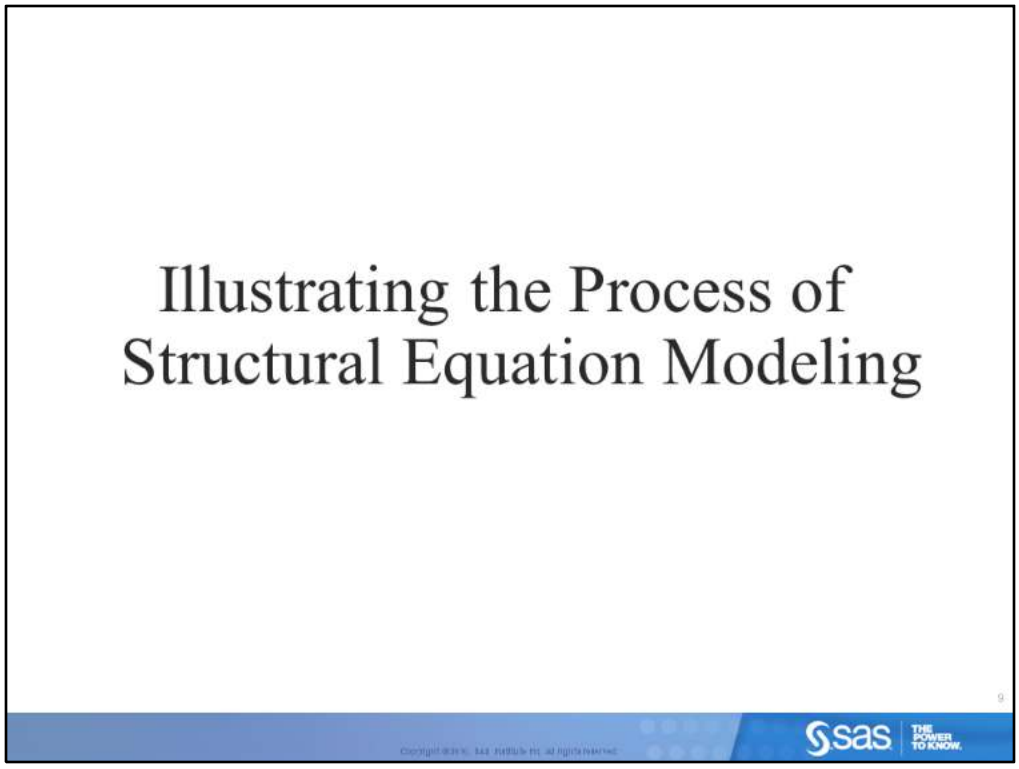
The second part of the workshop is about "advanced" modeling---relatively speaking. I will show how multiple-group analysis can be done in PROC CALIS. Other important topics such as direct and indirect effect analysis, testing specific hypotheses, and model modifications are discussed. Two newest SEM techniques, FIML estimation and case-level residual analysis, are also described.



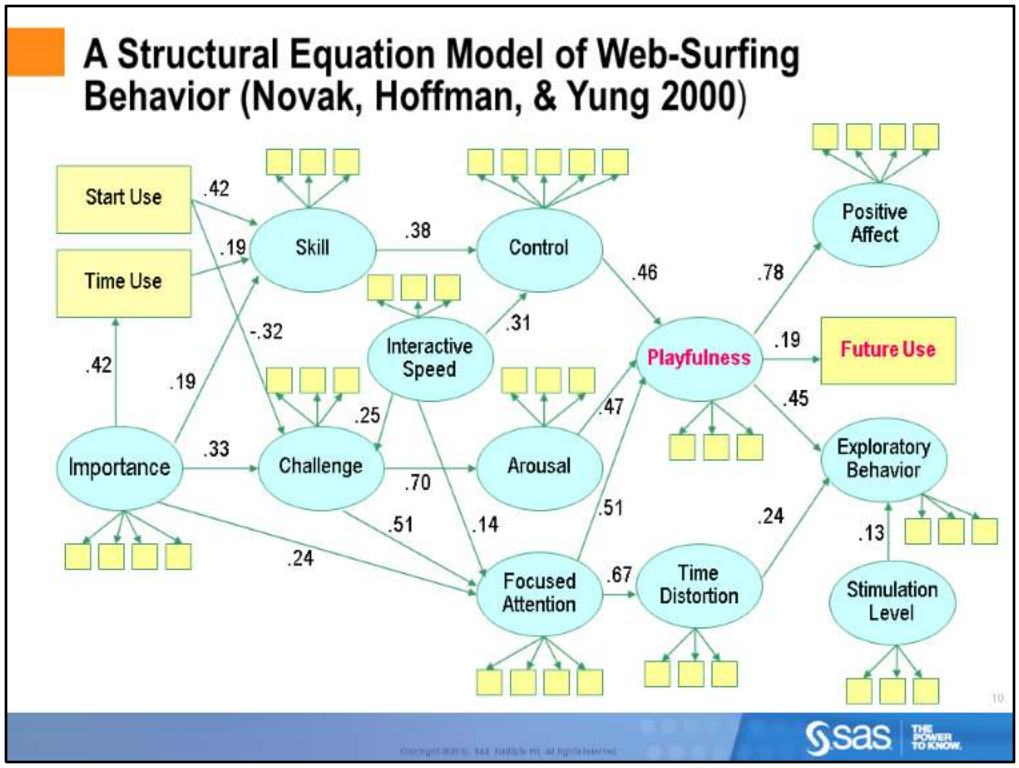
There are two emphases of this talk.

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 modeling structural relationships.



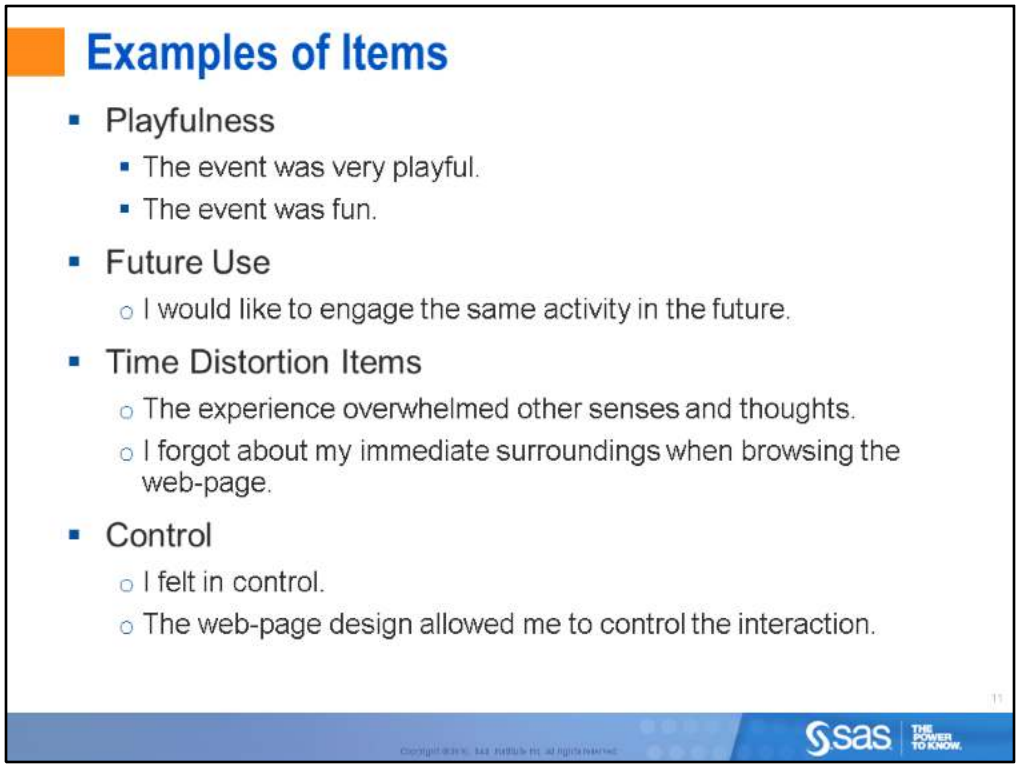
To give you a realistic idea about the scope of application of structural equation modeling, I will first describe an elaborate research example.



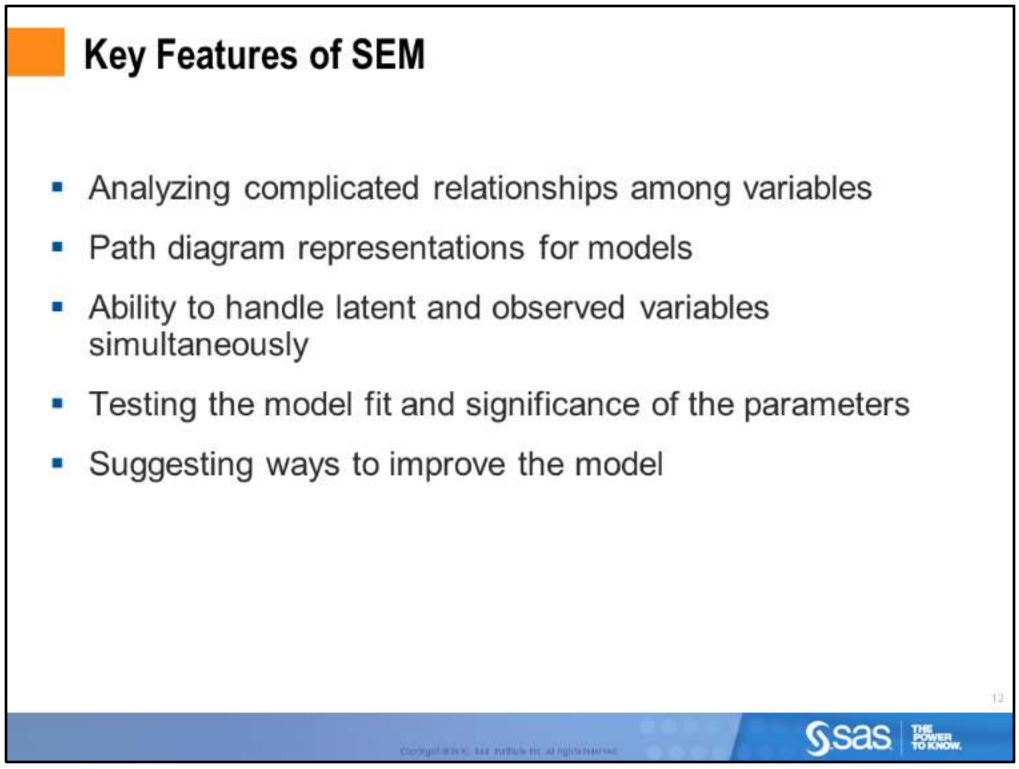
This is a structural equation model about web-surfing behavior. The researchers hypothesize that "Playfulness" of a web-site enhances the future use ("Future Use") of the same web-site. However, the theory does not end there. The researchers also hypothesize what makes a web-site to be perceived as playful. 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 that predict the future use of a web-site.

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) for the latent constructs are needed. In the path diagram, those small unlabeled rectangles are measured indicators for their latent constructs. In this research, these measured indicators are rating responses on a questionnaire. See the next page for examples of these items.

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. All numbers in this path diagram are effect estimates. In addition, the SEM software tells you the significance of these estimates. If the model does not fit the data well, the SEM software suggests ways to improve the model.

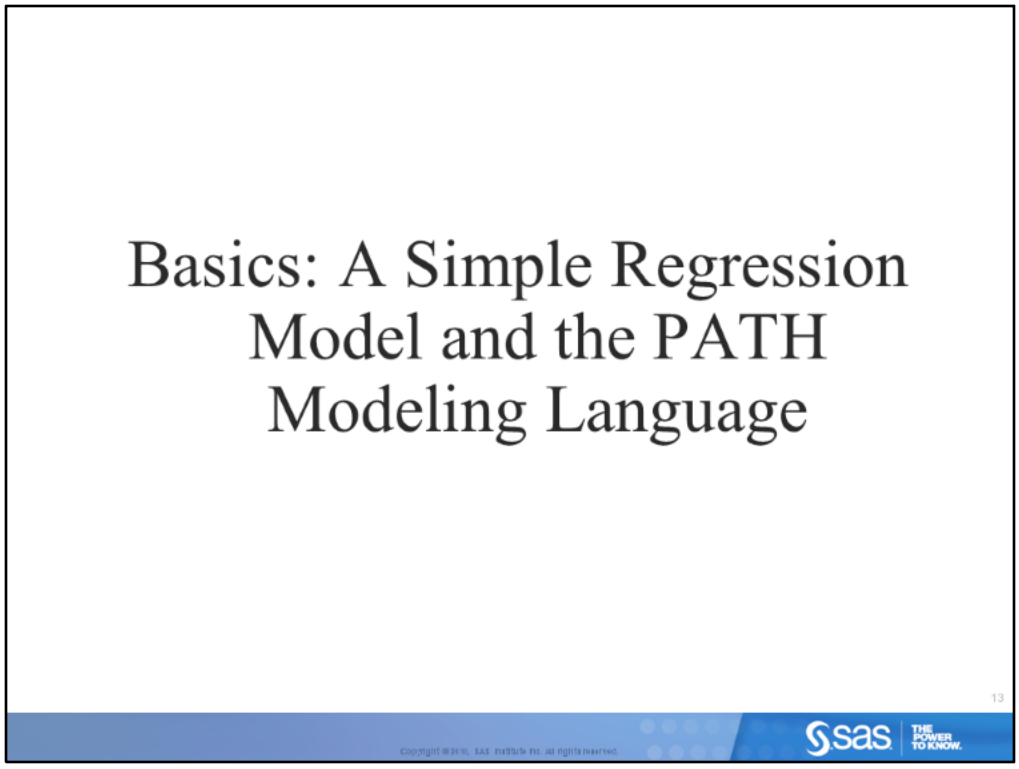


This slide shows the examples of the items used in the research. All these are rating scales. Respondents indicate whether they agree or disagree on a 7- or 5-point scale for each item.

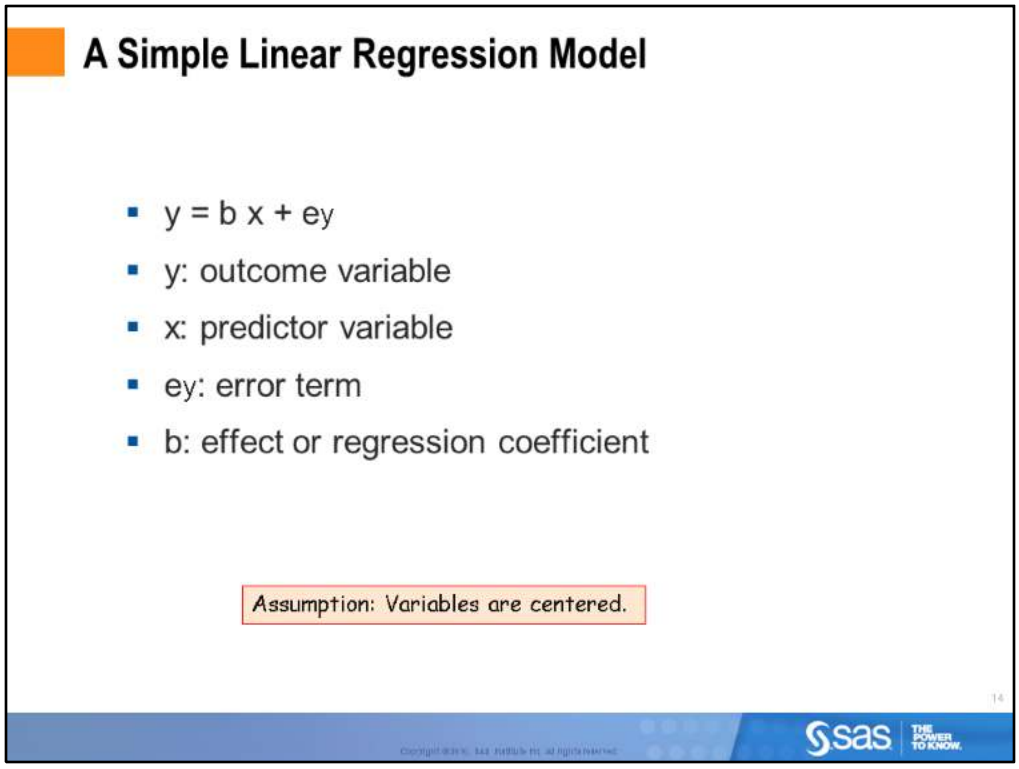


To summarize, here is a list of the key features of SEM:

- 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 effect parameters
- Suggesting ways to improve the model

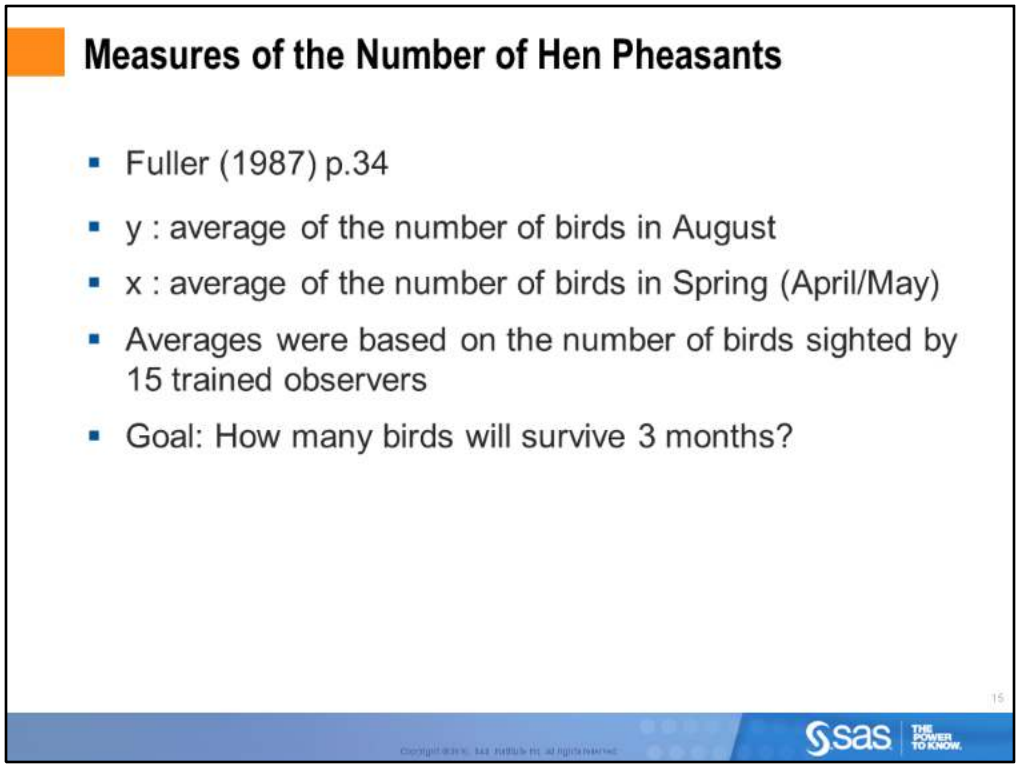


Learning SEM can be a very complicated process. Let us start with a simple example in linear regression.

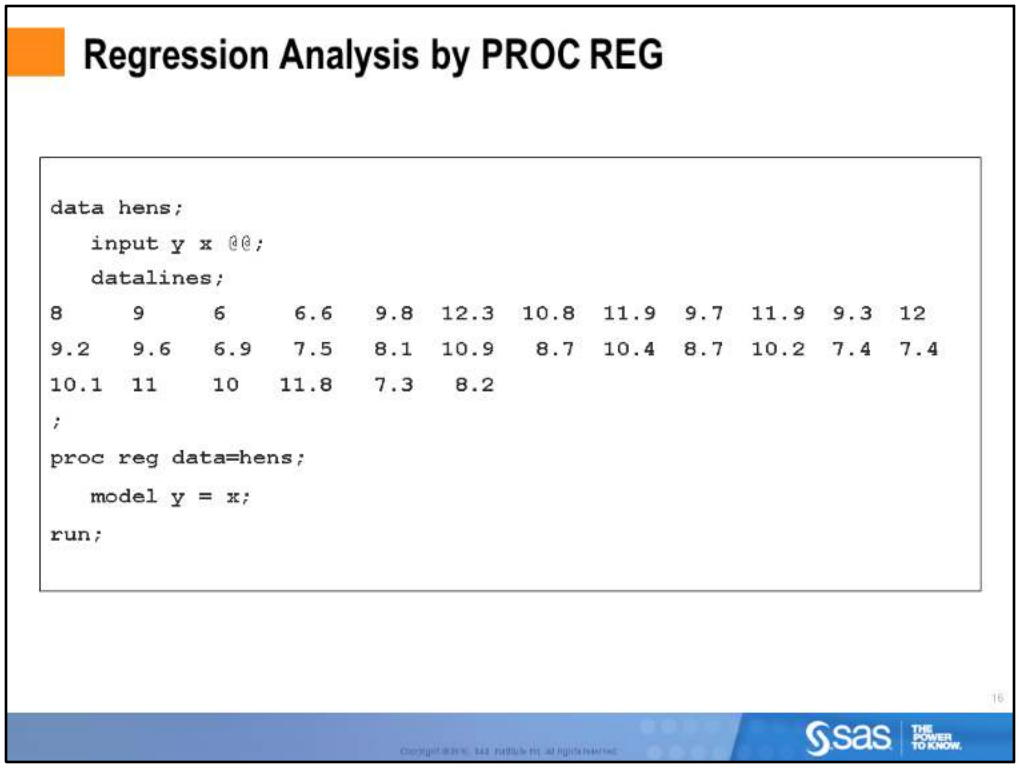


To introduce the PATH modeling language in PROC CALIS, a simple linear regression model is used. In the regression equation, y is the outcome variable, x is the predictor variable, e_y is the error term and b is the effect or regression coefficient. The regression model written in this form assumes that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

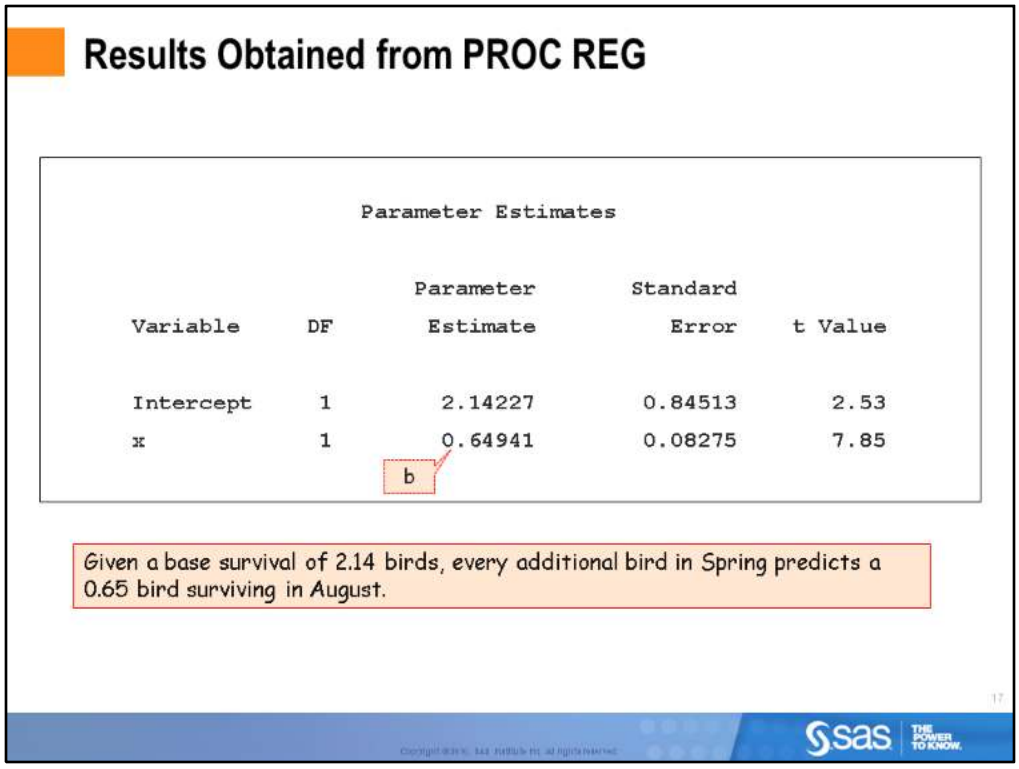
When you use PROC CALIS, you can input raw data or the covariance matrix of the observed variables for analysis. There is no need to center your variables.



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).

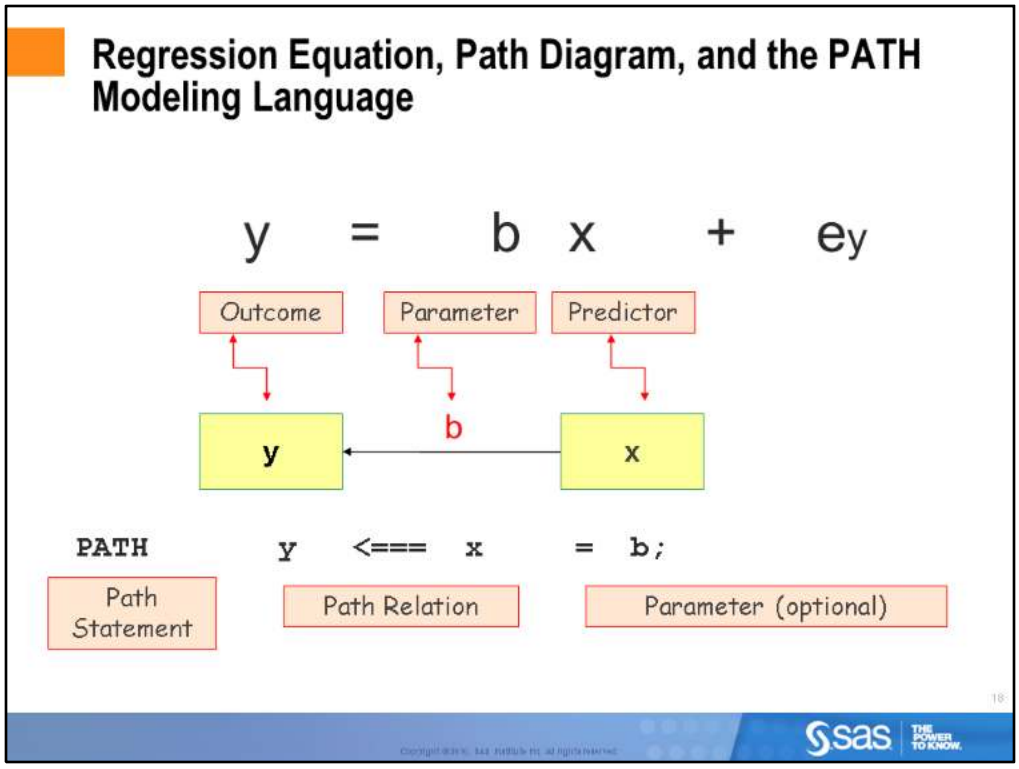


To conduct a linear regression analysis, you can use a SAS procedure called PROC REG. The syntax is quite simple. First, define your data set. Second, call PROC REG with the interested data set specified in the PROC REG statement. Then, the model statement specifies that y = x, which means that y is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



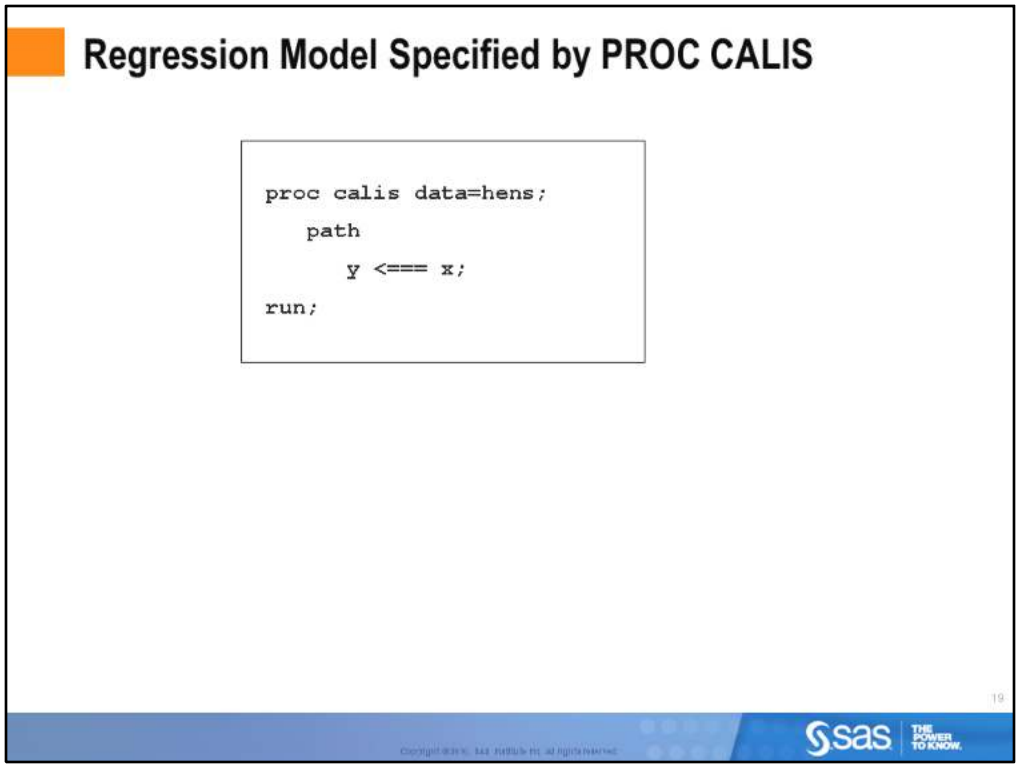
This table shows the essential results from PROC REG. The output shows an estimate of 0.65 for the regression coefficient b. The intercept estimate is 2.14. PROC REG also shows the standard error estimates and the t values for judging statistical significance. Both estimates are statistically significant.

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



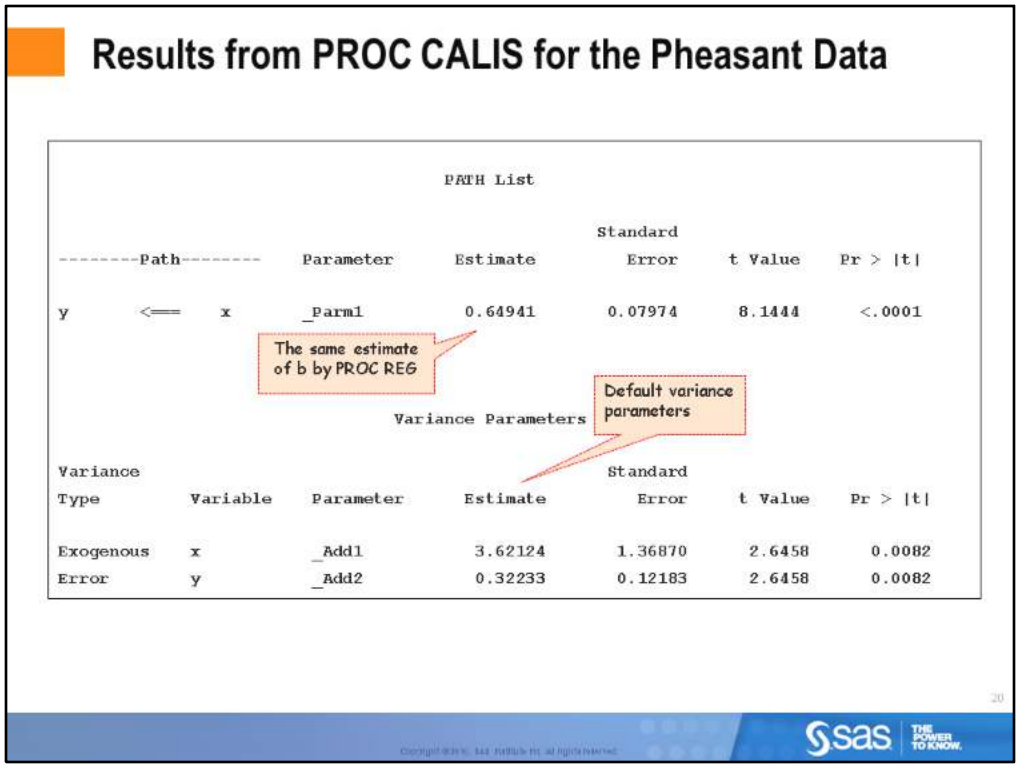
As shown previously, you can represent the linear regression model by the path diagram, which is also a representation scheme for SEM.

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 name for the regression coefficient or effect.



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

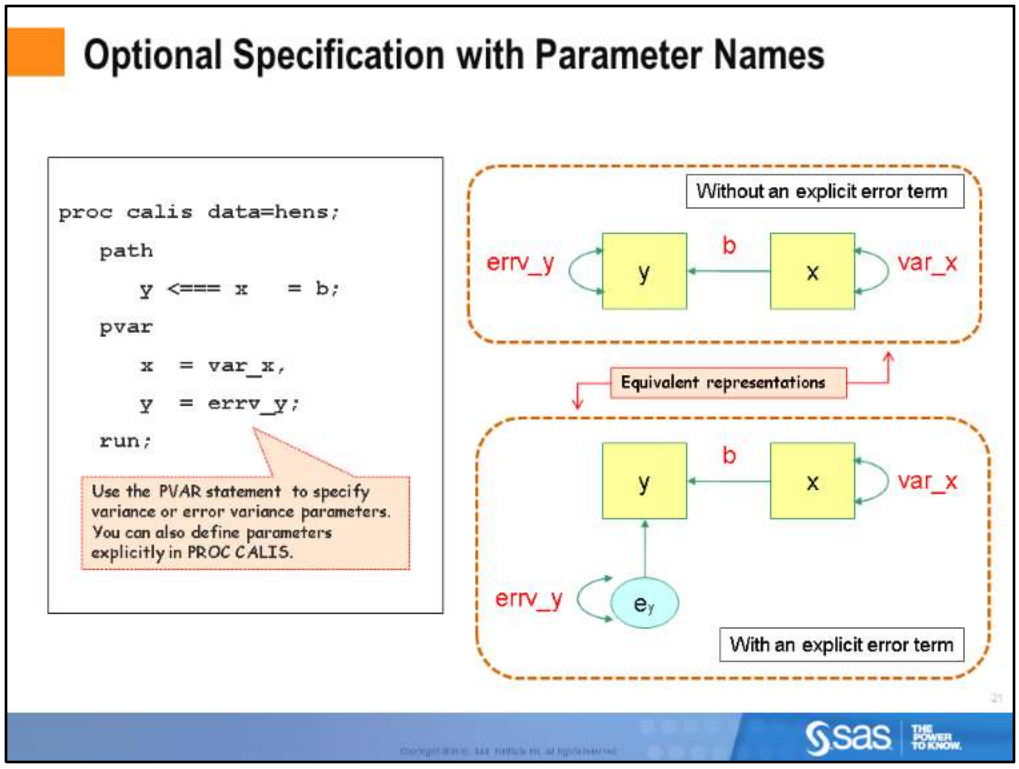
Again, you do not need to specify any error term (and the corresponding error variance) for the regression (or the path) as PROC CALIS assumes the prediction is not perfect by default.



This slide shows the output 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 model. 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. In fact, all default parameters added by PROC CALIS have the prefix "_Add".

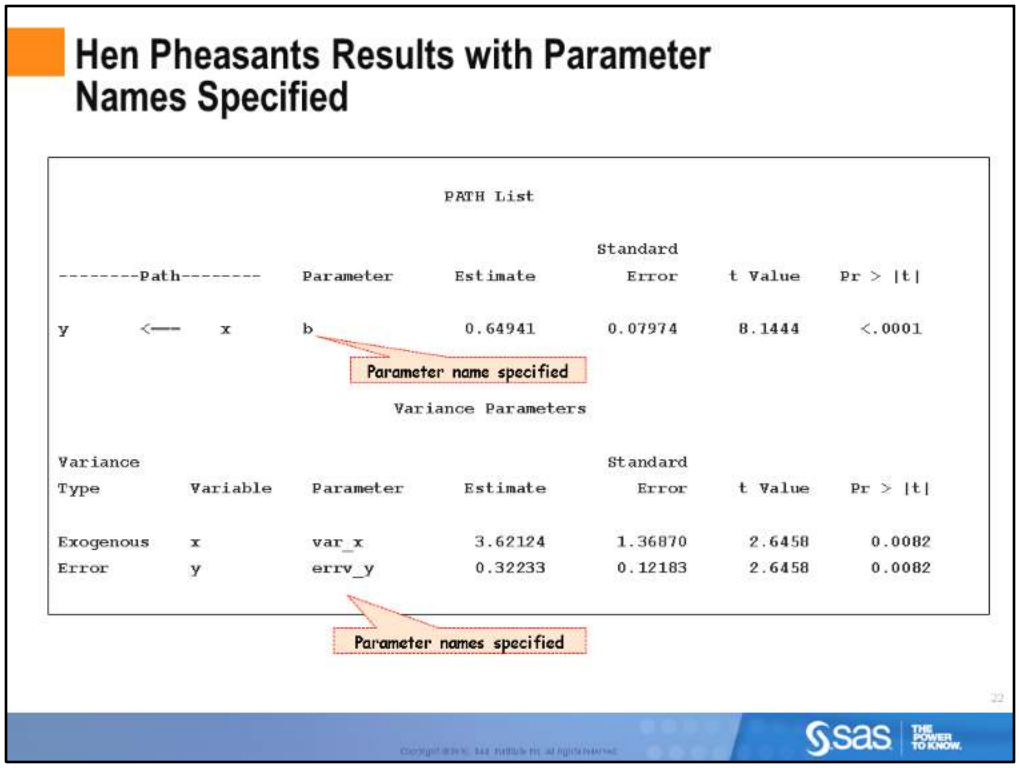


You could name all the parameters in PROC CALIS by putting your preferred names. This slide shows a complete specification of the regression model.

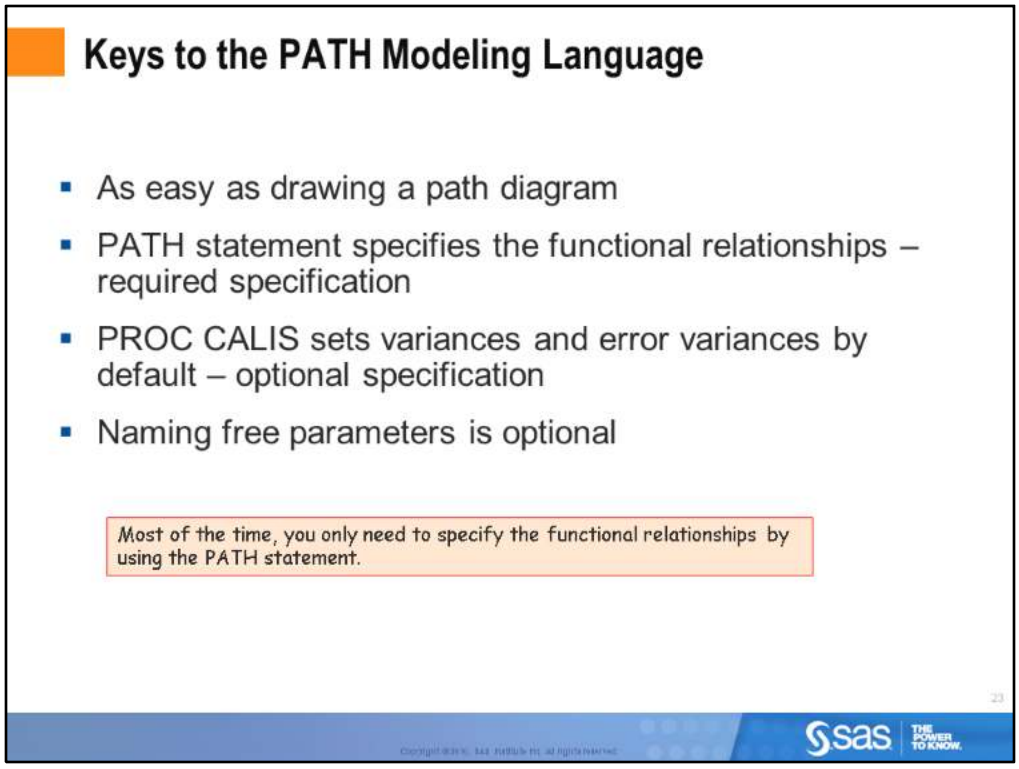
In the path diagram at the top right corner, the parameters are shown in red. In the regression model, b is the regression coefficient, var_x is the variance of the predictor variable x, and errv_y is the error variance of y. This path diagram representation is equivalent to the one shown at the bottom 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 all 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. The capability of naming parameters means that you can have more control on specifying your models.

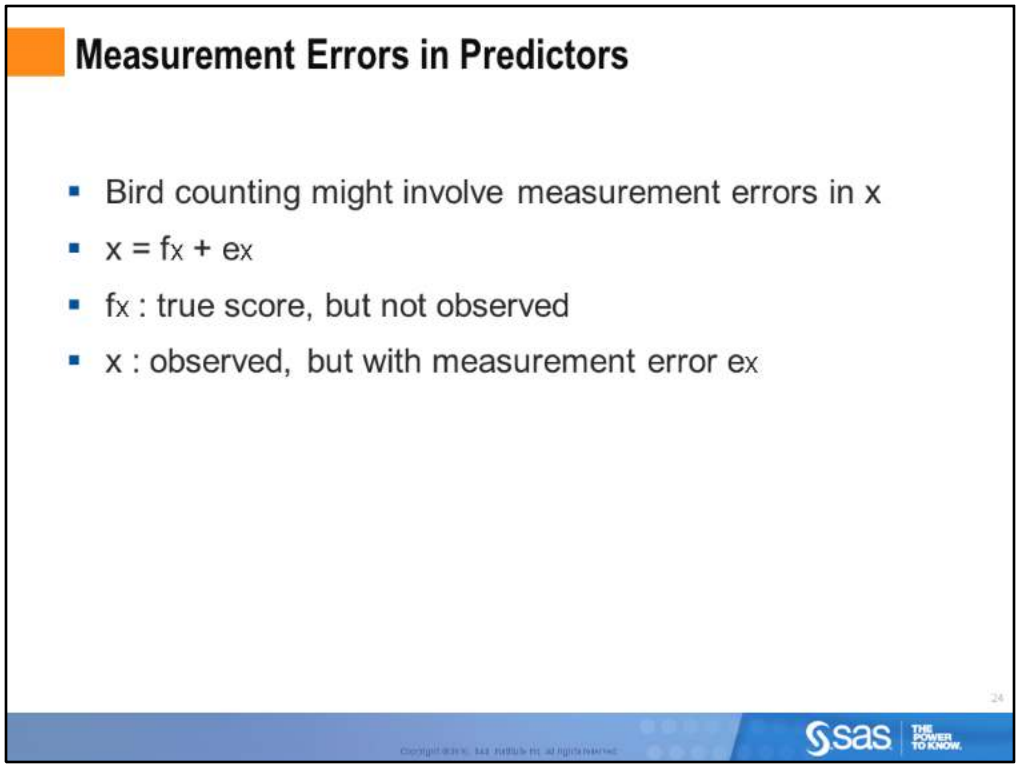


As shown in this slide, the numerical results from PROC CALIS with explicit parameter names specified are the same as those without using parameter names. The only difference is that now you can use these parameter names to locate the corresponding results directly.



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 specifying 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 the CALIS syntax is the paths specified in the PATH statement.**
4. Naming parameters is optional in PROC CALIS.

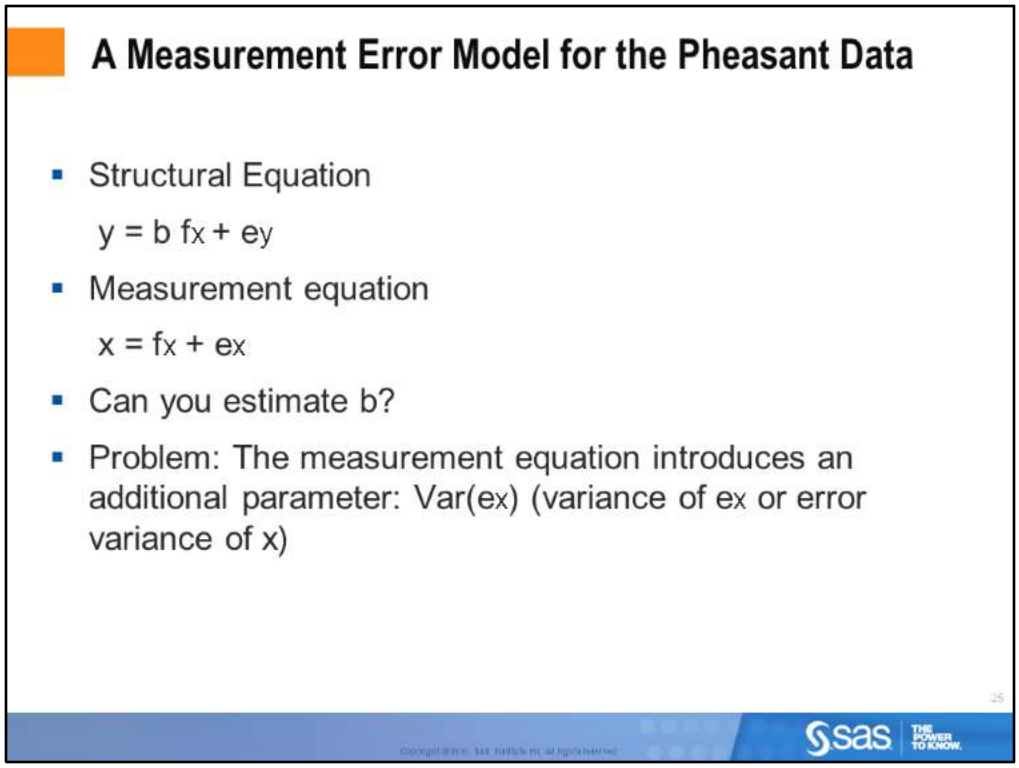


Let us make a little step forward to show a special SEM feature that linear regression cannot handle easily.

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" bird counts. 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.



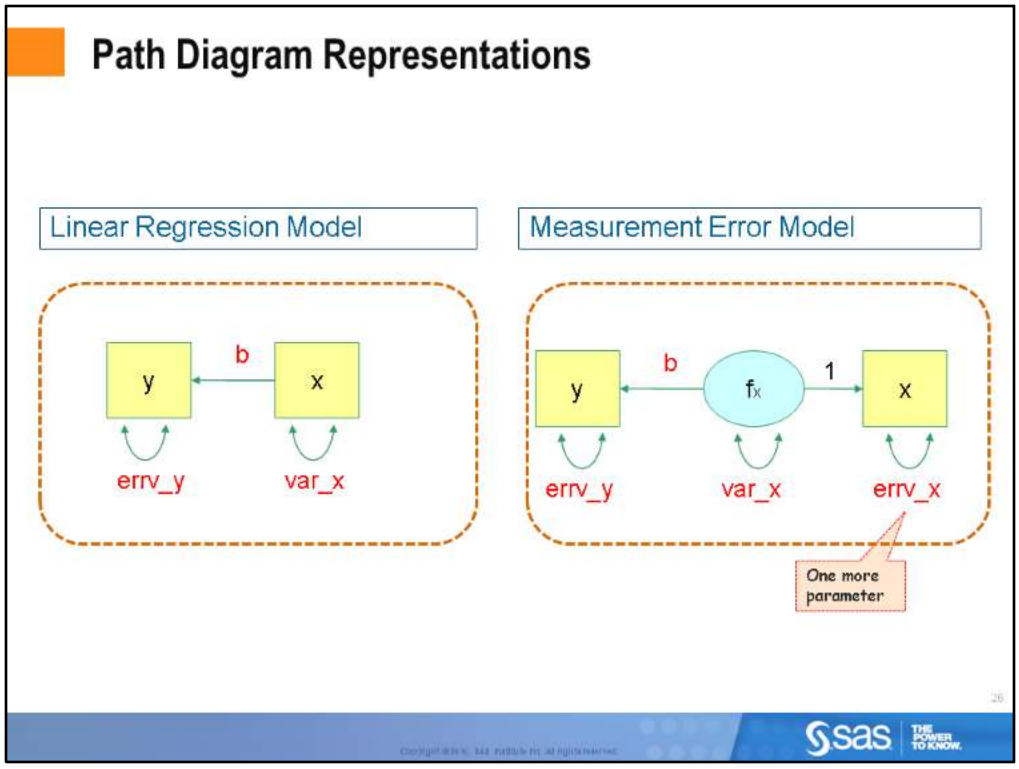
The preceding idea is formalized as the following structural equation model with a latent variable fx.

In the structural model, y is now 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 variable fx as the predictor. In the 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? Will it give you different results than that of the ordinary regression analysis? This answer is yes.

But a technical problem encountered in measurement error model must be dealt with first. That is, the measurement equation introduces one additional parameters in var(ex)---error variance of x. We must have a way to know the approximate amount of this error variance in order to estimate other parameters in the model. More technically, this is an identification problem in the context of SEM. In a 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.

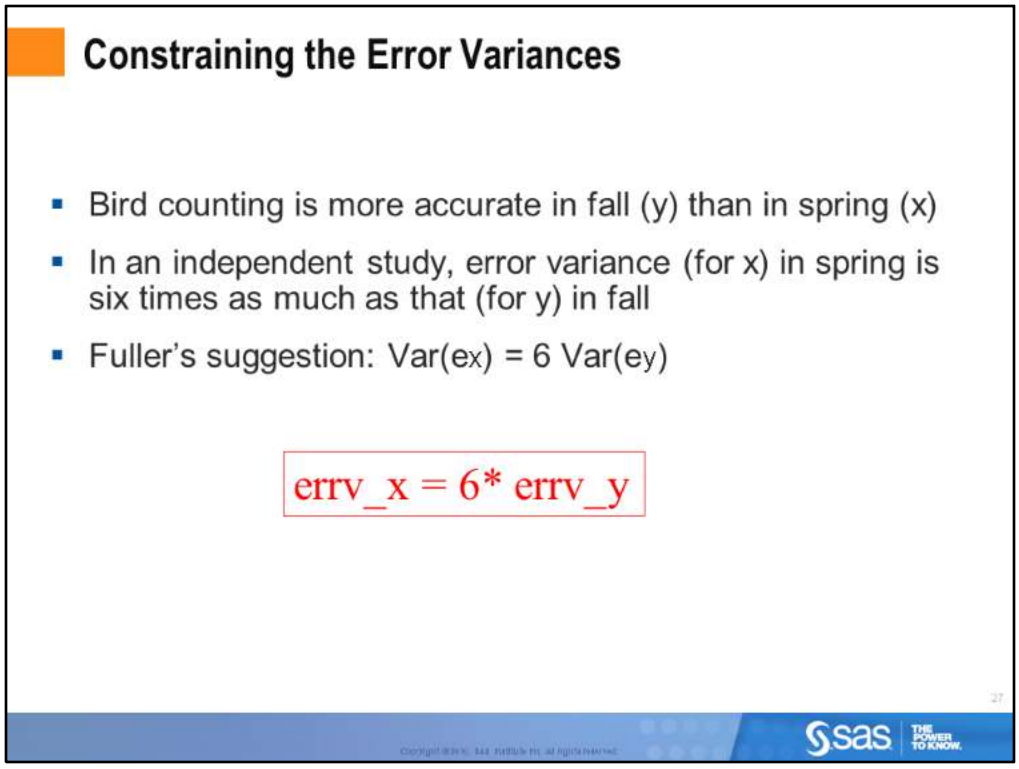
In general SEM, using three or more observed indicators for each latent factor (true score) would generally resolve this kind of identification problem. This will be described later in the context of confirmatory factor analysis. For the current example, Fuller suggests a useful way to access the amount of measurement error in x so that identification problem vanishes.



Before I discuss Fuller's solution, let us compare the linear regression model with the measurement error model by use of path diagram. This 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.

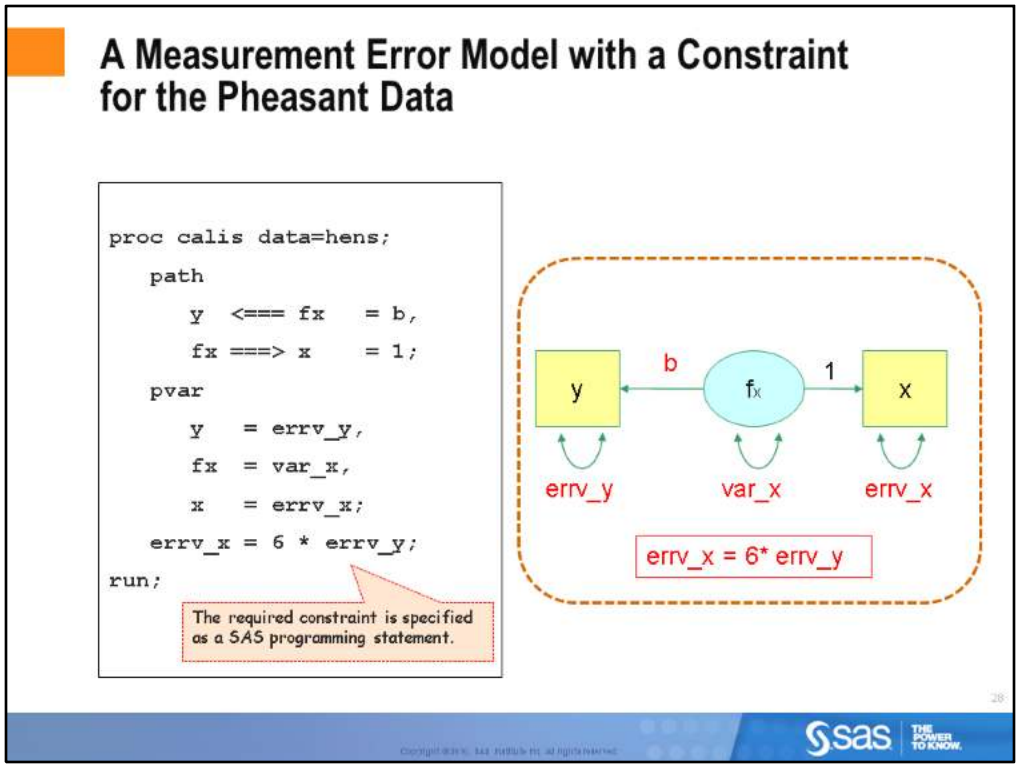
In the right panel for the measurement error model, we still have x and y as the observed variables. But now we have a latent variable fx that takes the role of the predictor of y. Var_x in this model now represents the true variance of the predictor fx. The new parameter in the measurement error model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



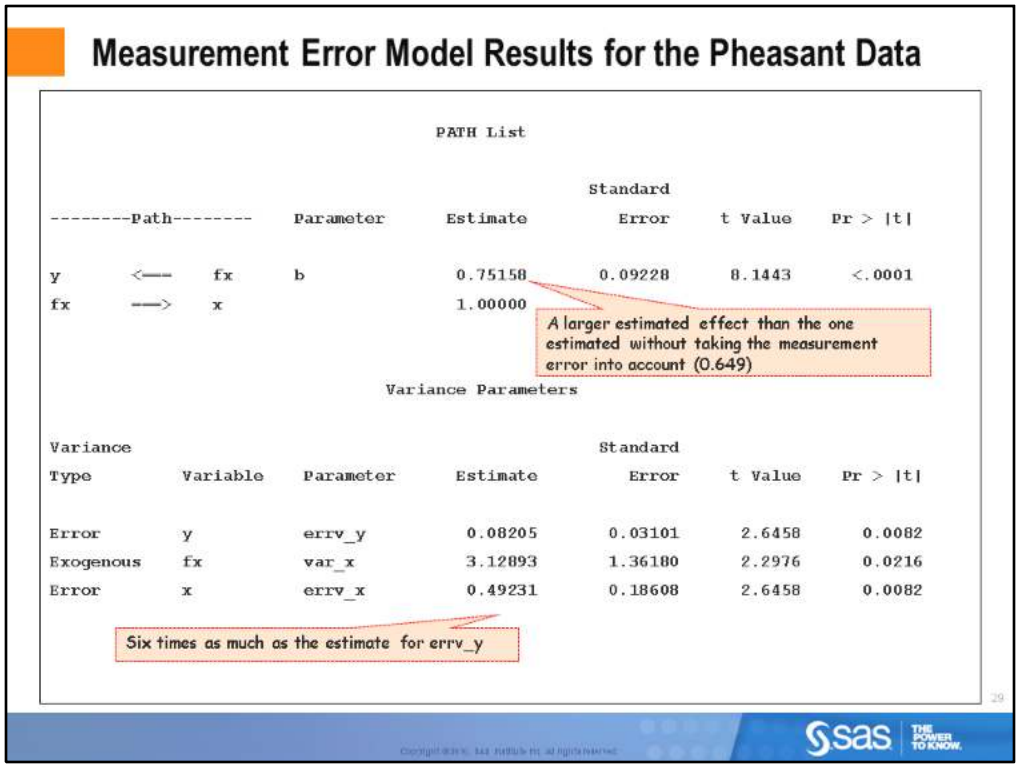
Fortunately, we have a reasonable assumption about the relative size of the measurement error variances of x in the model. This slide follows Fuller's solution in his textbook.

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 was validated by an independent study about the relative error variances in x and in y. In Fuller's textbook, he reported that this ratio is about 6. Mathematically, therefore, we may set Var(ex) = 6*Var(ey). That is, error variance for x is six times as much as the error variance of y. In running PROC CALIS, you need to incorporate the following parametric constraint in the modeling: errv_x=6*errv_y.

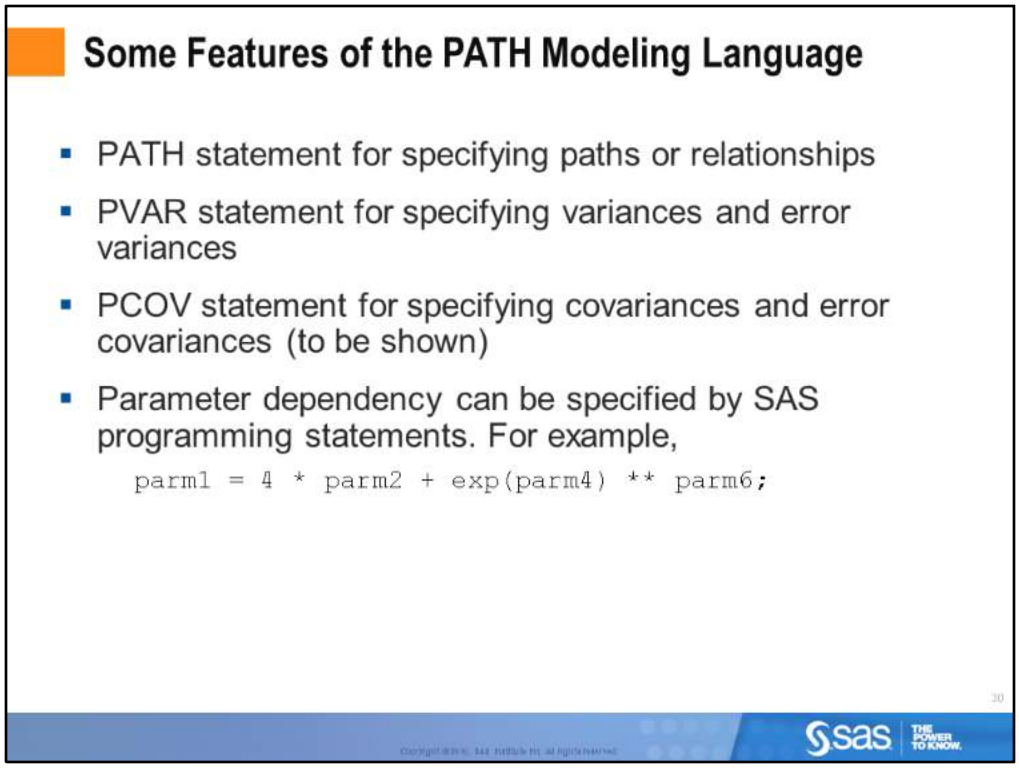


It turns out that it is pretty straightforward to specify this kind of parametric constraint in PROC CALIS. You just simply add one more line of code to represent this relationship, as shown in the SAS code on the slide. In the SAS literature, this line of code is called a SAS programming statement, 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. With this statement for constraining the error variances, your model is identified.



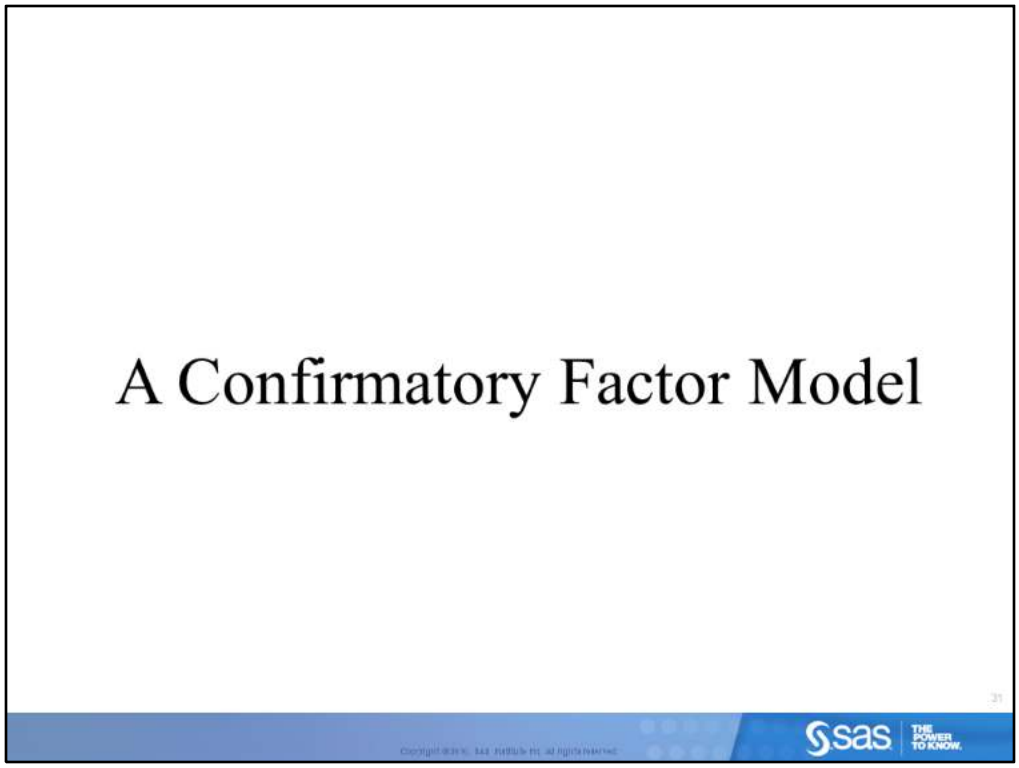
With the measurement error model, the regression coefficient b is now 0.75. The represents a larger effect than 0.649, which you obtained from the linear regression model without taking the measurement error in x into account. Therefore, the previous regression analysis underestimated this effect because 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 syntax is honored in the estimation. The error variance estimate of x is 0.49, which is indeed six times as much as the error variance estimate of y, which is 0.08.

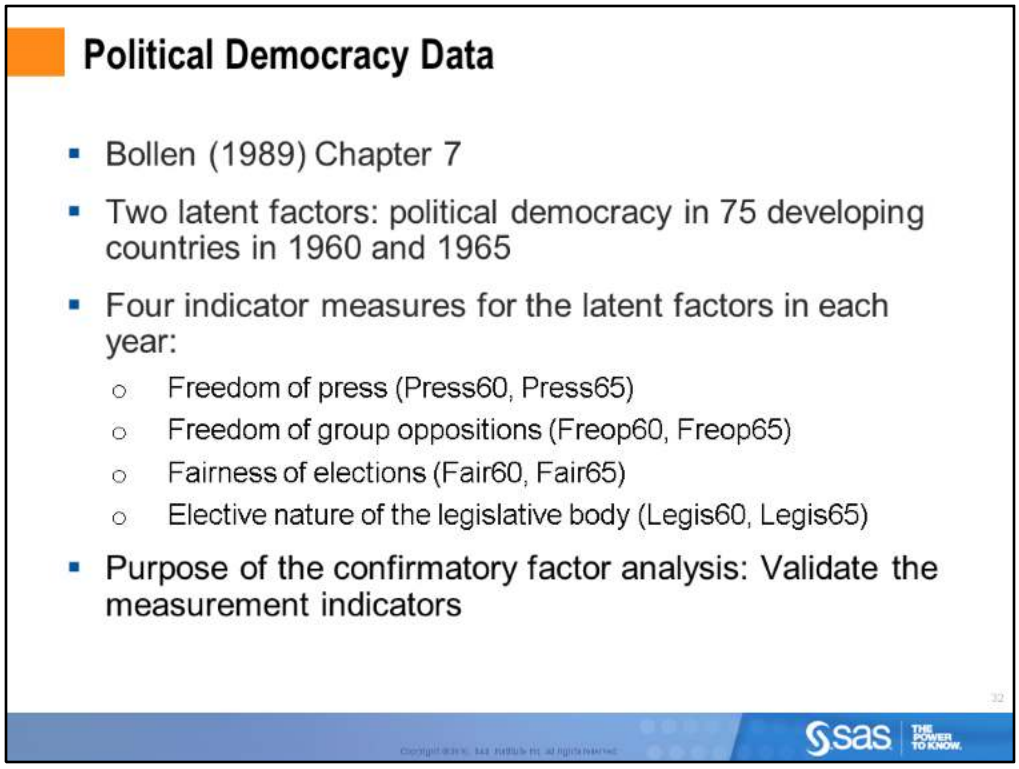


This slide summarizes some features of the PATH modeling language.

1. It is as straightforward as drawing the paths--- you can specify latent and observed variables in the same way.
2. You can use the PVAR statement to specify variances or error variances (double-headed arrows attached to individual variables in the path diagram).
3. You can use the PCOV statement to specify covariances or error covariances (double-headed arrows attached to pairs of distinct variables in the path diagram).
4. 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.



We now move on to a little more complicated class of models called confirmatory factor-analysis (CFA) models.

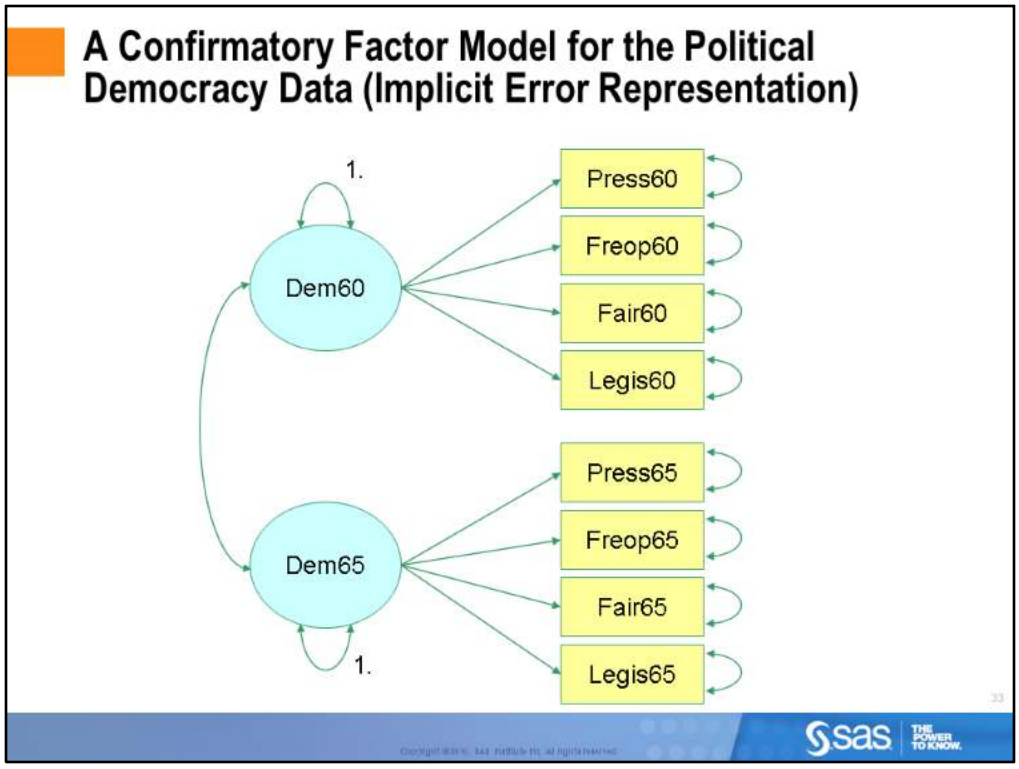


This example is based on an example in Chapter 7 of Bollen's classic textbook: Structural Equation Modeling.

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 (Press), freedom of group oppositions (Freop), fairness of elections (Fair), and elective nature of the legislative body (Legis).

The purpose of the confirmatory factor analysis is to validate these measurement indicators statistically. We will discuss what would be considered to be a validation of these measurement indicators.

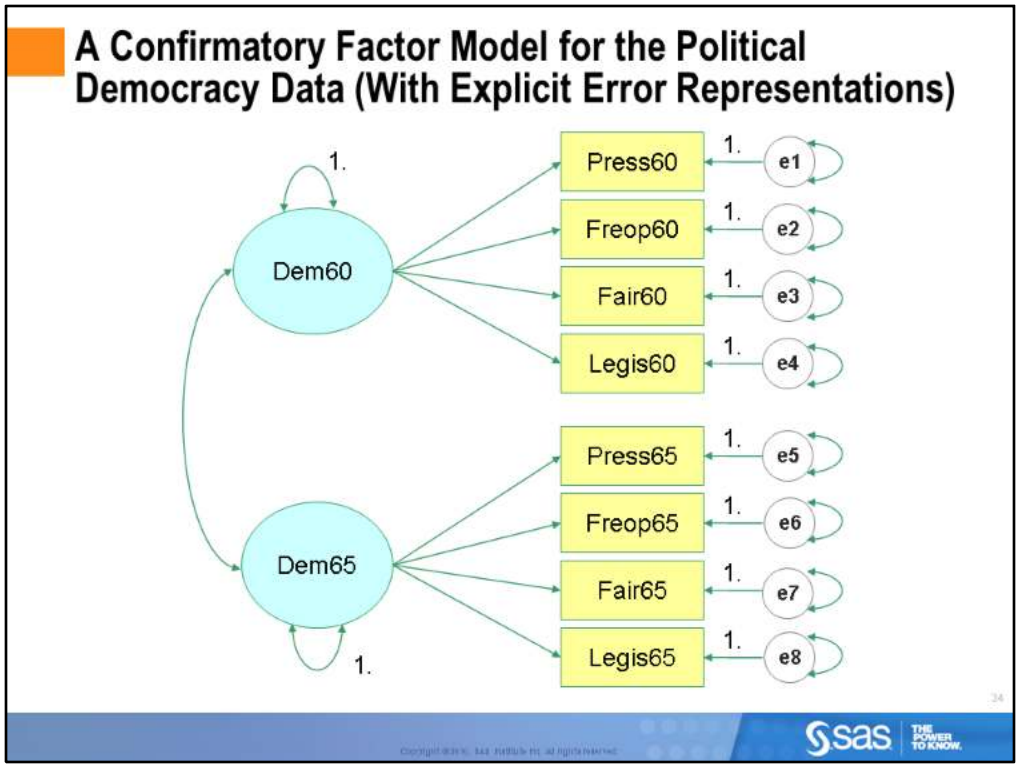


This path diagram shows the hypothesized confirmatory factor model.

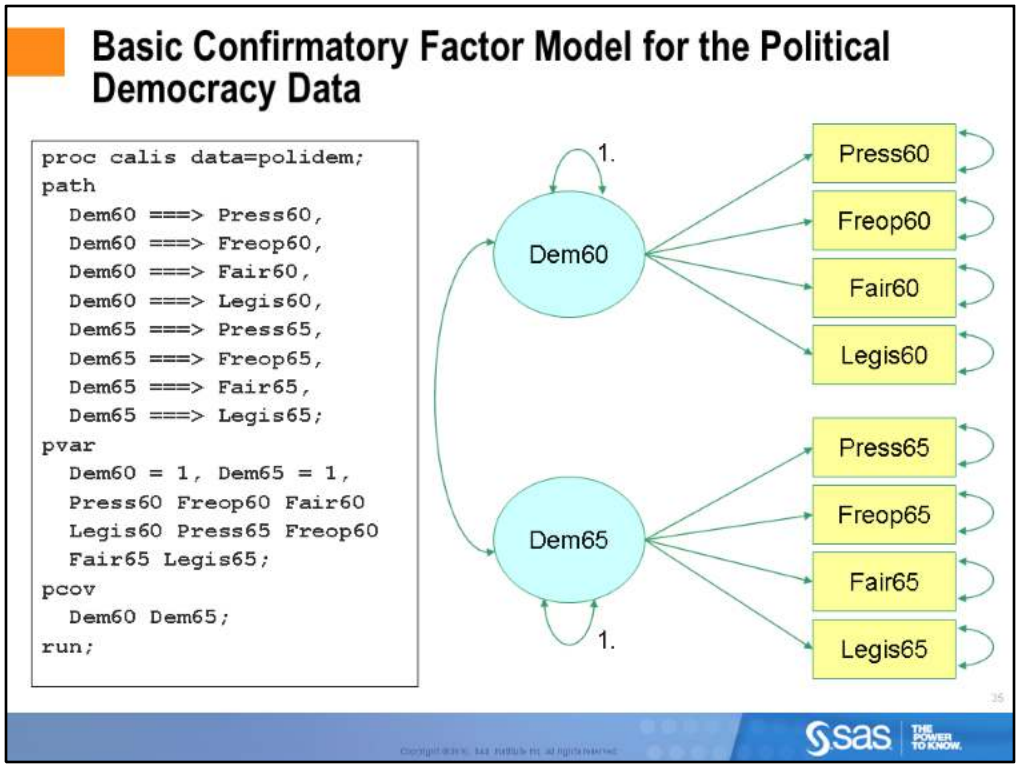
In the path diagram, two latent factors are represented by two ovals. Dem60 is the political democracy in 1960 and Dem65 is the political democracy in 1965. They are linked to the respective measured variables, as shown in the path diagram. These single-headed paths represent the typical factor-observed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

The double-headed arrow that connects Dem60 and Dem65 represents the covariance parameter between the two latent 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 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 observed variables that are not due to their relationships with the factors in the model.



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 mostly rely on the path diagram representation that does not use explicit error terms.

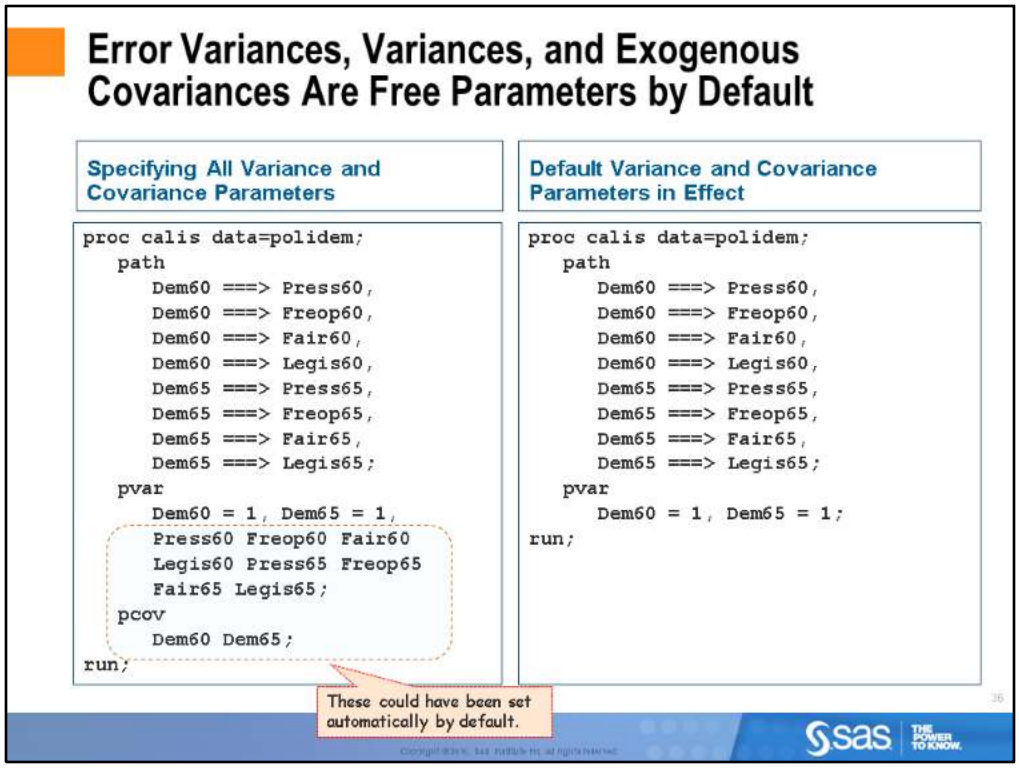


Specifying the CFA model is not much harder than the previous measurement error model. Basically, you only need to specify more paths for the CFA model.

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 free parameters in the model. In the current path diagram, Dem60 and Dem65 are correlated and so they are specified in the PCOV statement.

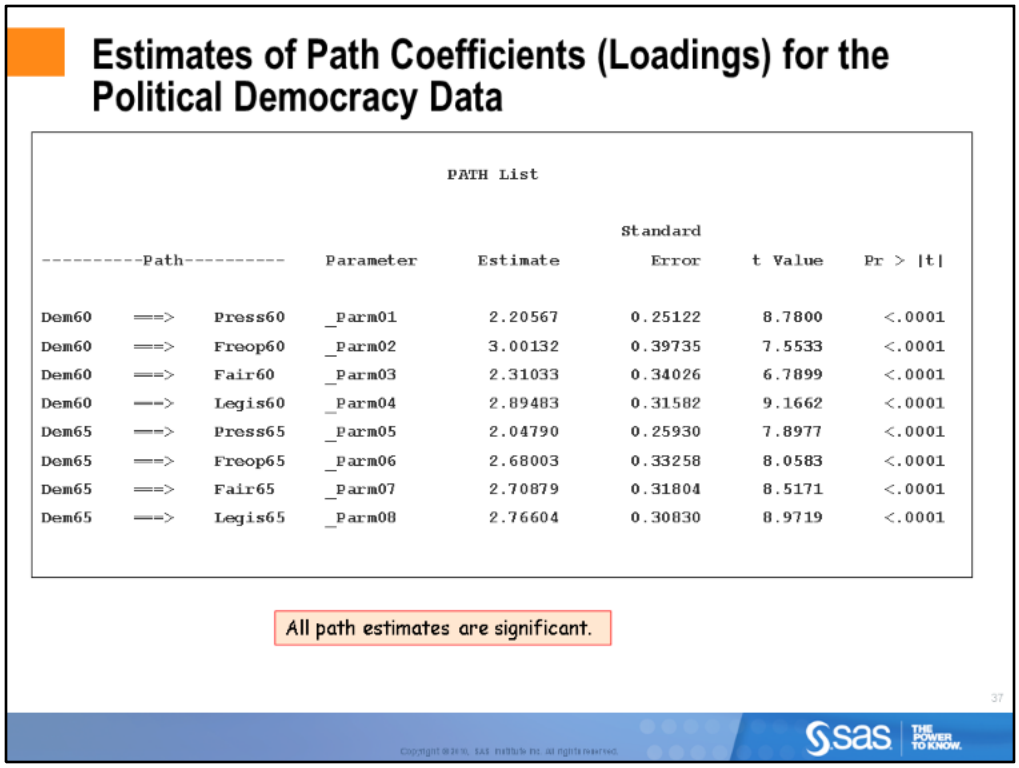


To make model specification more efficient and error-free, PROC CALIS employs default free parameters in the model. These default free parameters are set because they are commonly employed in practice.

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 treated as free parameters by default.



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.

For this example, all path coefficients are statistically significant and so all factor-variable relationships are well-established.

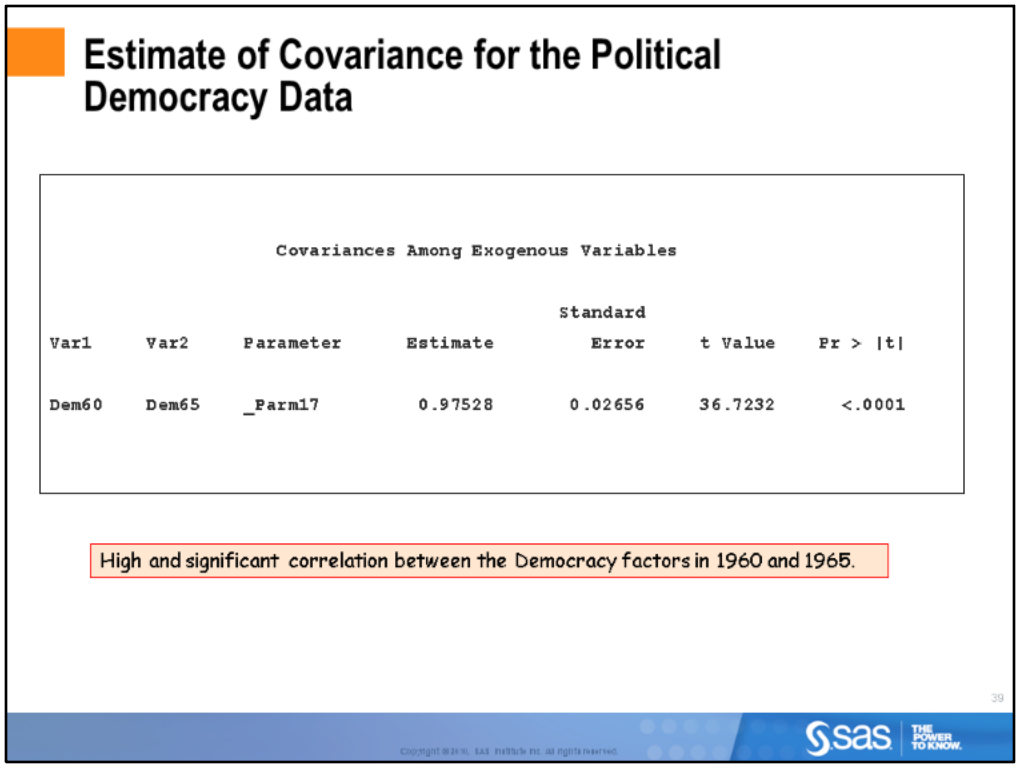
Estimates of Variances for the Political Democracy Data

Variance Parameters

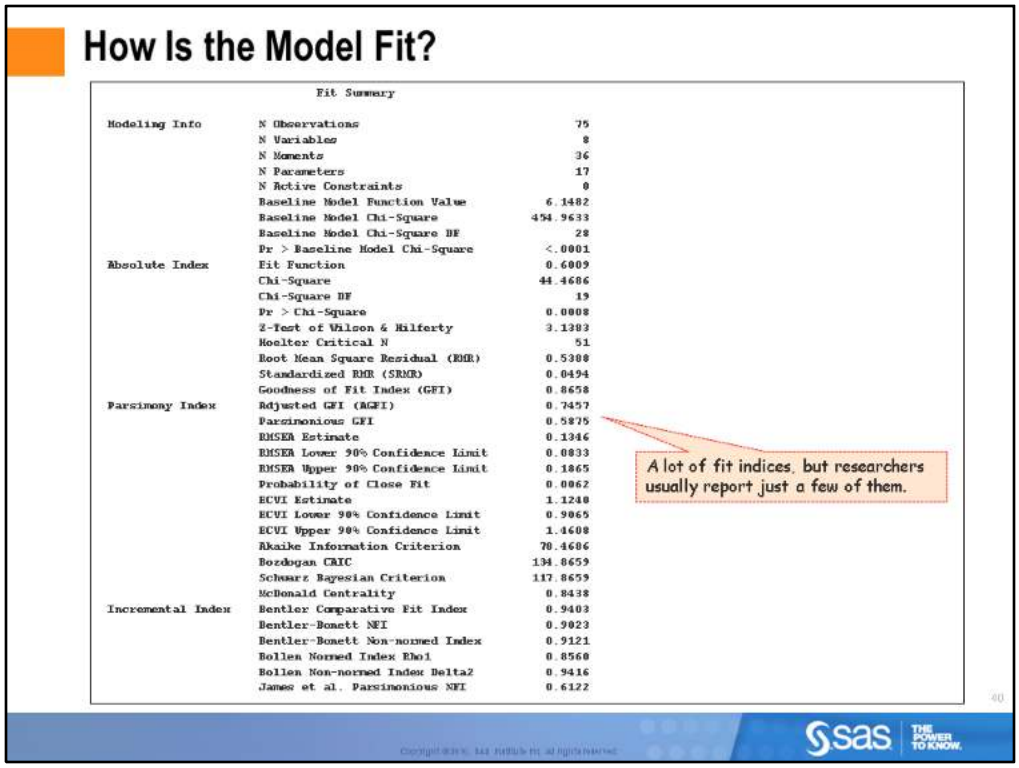
Variance Type	Variable	Parameter	Estimate	Standard Error	t Value	Pr > |t|
Exogenous	Dem60		1.00000			
	Dem65		1.00000			
Error	Press60	_Parm09	2.01359	0.41048	4.9055	<.0001
	Freop60	_Parm10	6.57189	1.20964	5.4329	<.0001
	Fair60	_Parm11	5.42661	0.96546	5.6208	<.0001
	Legis60	_Parm12	2.83887	0.61417	4.6223	<.0001
	Press65	_Parm13	2.63180	0.49311	5.3371	<.0001
	Freop65	_Parm14	4.19276	0.79422	5.2791	<.0001
	Fair65	_Parm15	3.46180	0.68155	5.0793	<.0001
	Legis65	_Parm16	2.88292	0.59927	4.8107	<.0001

All error variance estimates are significant.

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 (indicated by zero error variances) between factors and observed variables are rare.



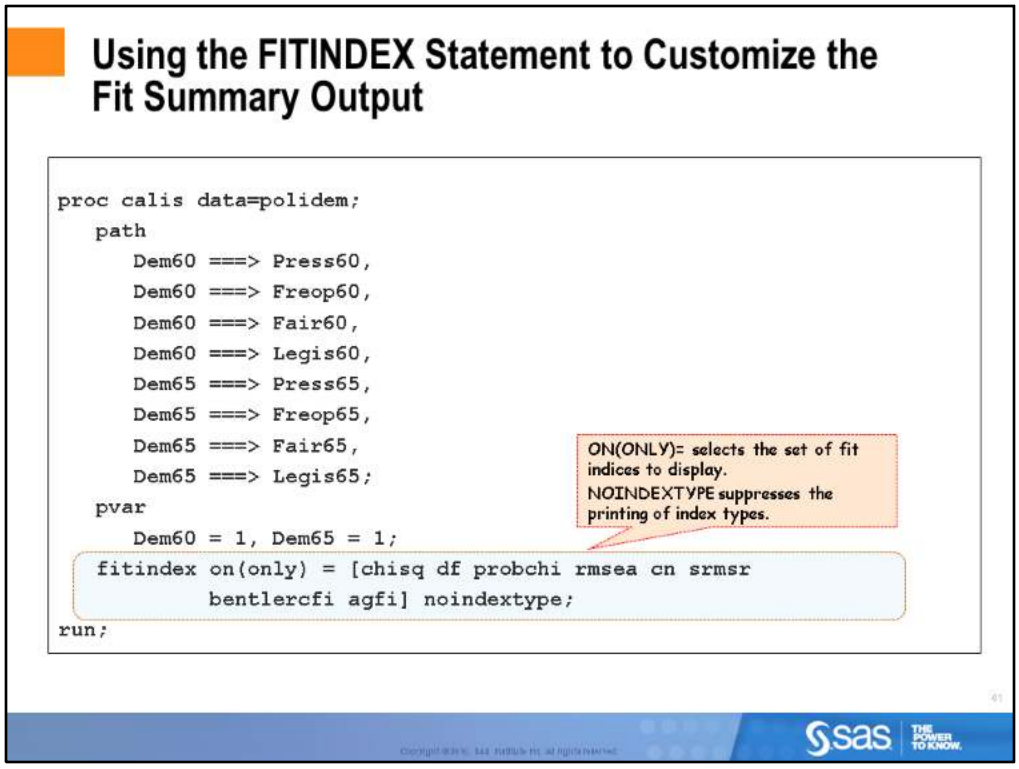
This table shows the covariance between Dem60 and Dem65. This estimate is also the estimated correlation between the two latent factors because the variances of the factors are fixed to one. This correlation is extremely high, possibly because the political democracy status of the countries do not change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



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 also need 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.

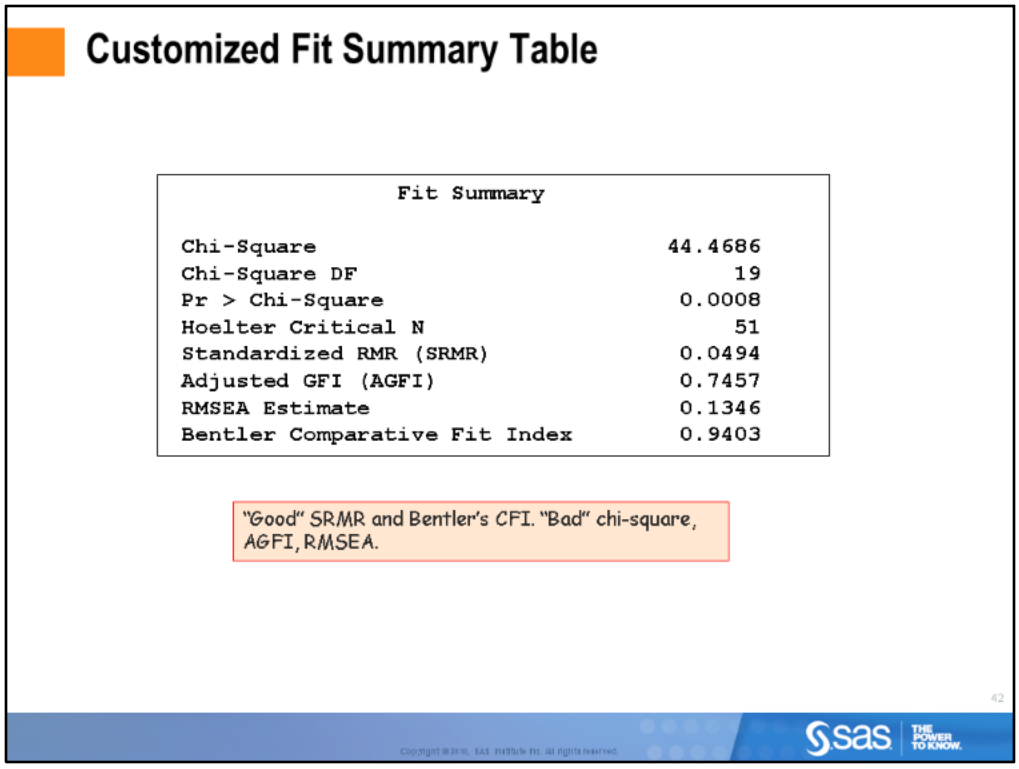


You can use the FITINDEX statement to customize your fit summary table.

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

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

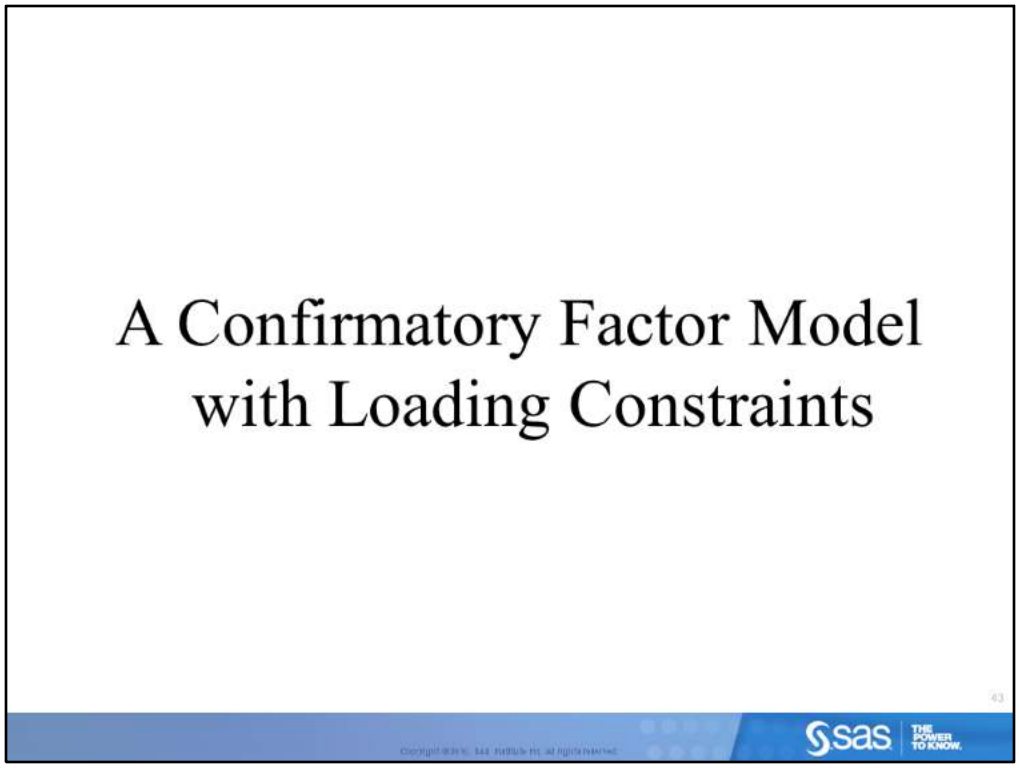
In this slide, I have selected the most “useful” fit indices to report in the field. These indices are the most useful because they are either: (1) theoretically sound; (2) easy to interpret; (3) justified by simulation studies; (4) justified by expert experience; (5) merely popular; or (6)
useful by a combination of the abovementioned reasons.



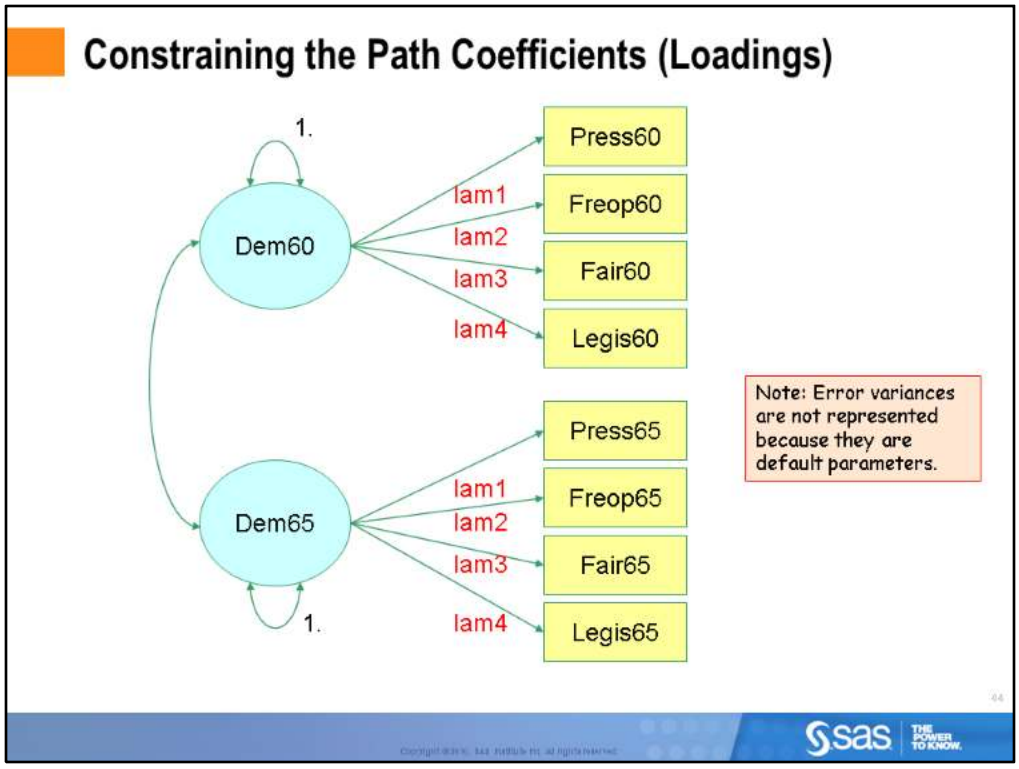
This is the customized fit summary output by using the previous FITINDEX statement specification.

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 nowadays 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 SRMR, 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 SRMR and RMSEA, the smaller the values, the better the fit. Usually, values under 0.05 indicate good model fit. Therefore, the SRMR 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.

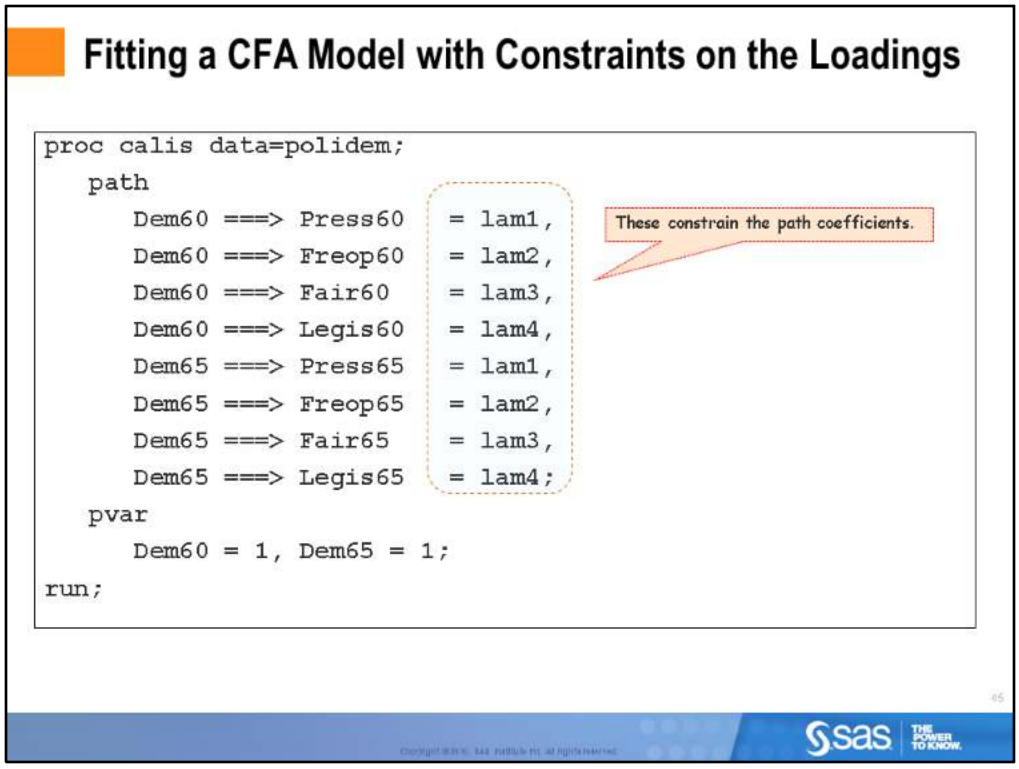


This example show you how to specify equality constraints in your model.

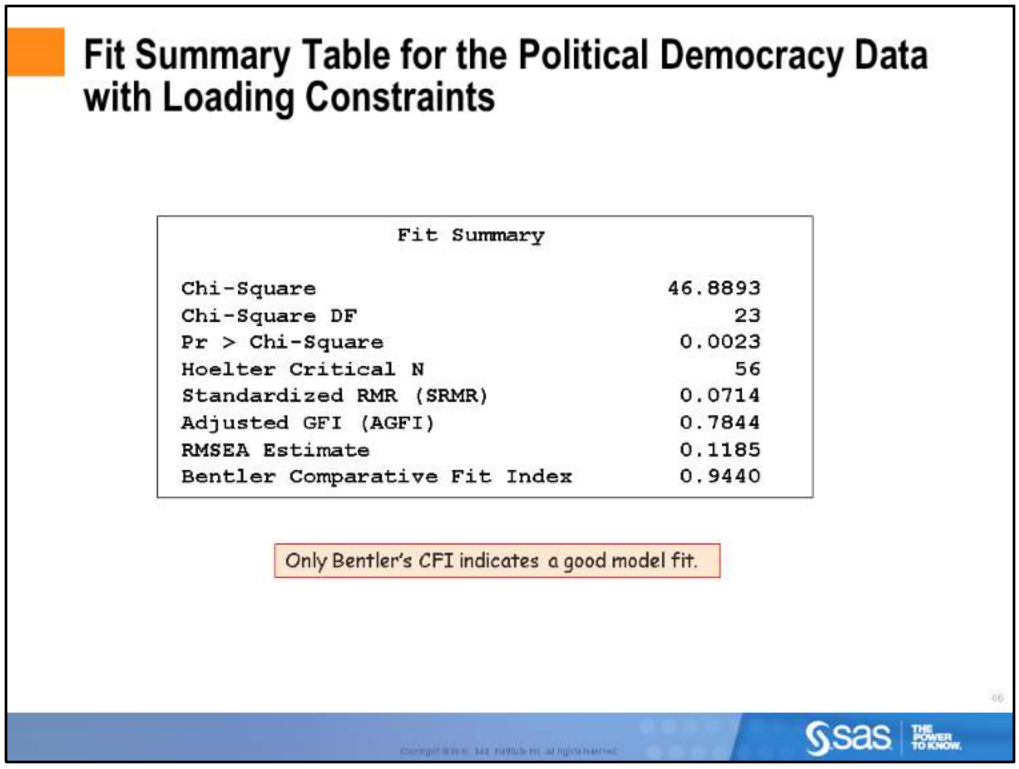


In addition to fitting a basic confirmatory factor model, PROC CALIS enables you to set up parameter constraints easily. The main tool is to use parameter names in the specification. We continue the previous example by imposing more constraints to the model.

For the political democracy example, the researcher wants to constrain the factor loadings (path coefficients) across time. The theoretical reason is that the measured variables are basically 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.



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.



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 SRMR 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) still shows a good model fit.

Certainly, the purpose of the current modeling is to illustrate the use of constraints, we expected a worse fit than the previous unconstrained model.

Estimates of the Constrained Loadings for the Political Democracy Data

PATH List

Path			Parameter	Estimate	Standard Error	t Value	Pr > |t|
Dem60	===>	Press60	lam1	2.13970	0.21716	9.8532	<.0001
Dem60	===>	Freop60	lam2	2.80116	0.29976	9.3447	<.0001
Dem60	===>	Fair60	lam3	2.54987	0.27316	9.3346	<.0001
Dem60	===>	Legis60	lam4	2.82969	0.27285	10.3708	<.0001
Dem65	===>	Press65	lam1	2.13970	0.21716	9.8532	<.0001
Dem65	===>	Freop65	lam2	2.80116	0.29976	9.3447	<.0001
Dem65	===>	Fair65	lam3	2.54987	0.27316	9.3346	<.0001
Dem65	===>	Legis65	lam4	2.82969	0.27285	10.3708	<.0001

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

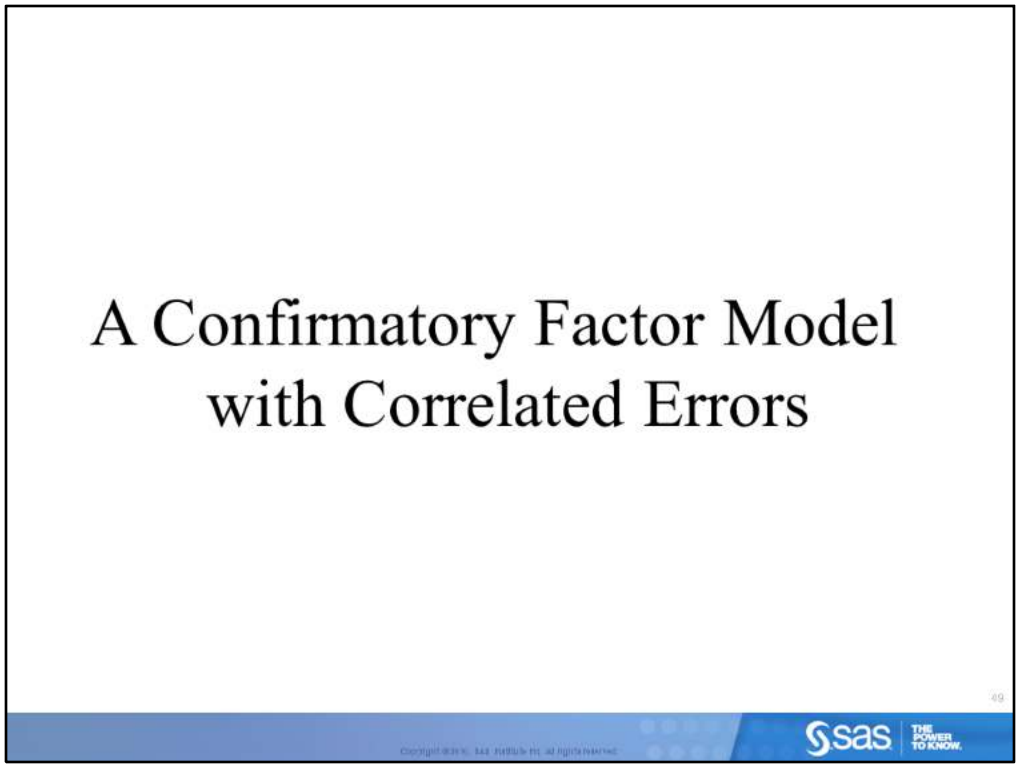
Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value	Pr > |t|
Exogenous	Dem60		1.00000			
	Dem65		1.00000			
Error	Press60	_Add1	2.01017	0.40312	4.9865	<.0001
	Freop60	_Add2	6.72037	1.21196	5.5450	<.0001
	Fair60	_Add3	5.40824	0.97833	5.5280	<.0001
	Legis60	_Add4	2.88468	0.60956	4.7324	<.0001
	Press65	_Add5	2.61966	0.49456	5.2970	<.0001
	Freop65	_Add6	4.16958	0.79818	5.2238	<.0001
	Fair65	_Add7	3.55382	0.67700	5.2494	<.0001
	Legis65	_Add8	2.85029	0.59556	4.7859	<.0001

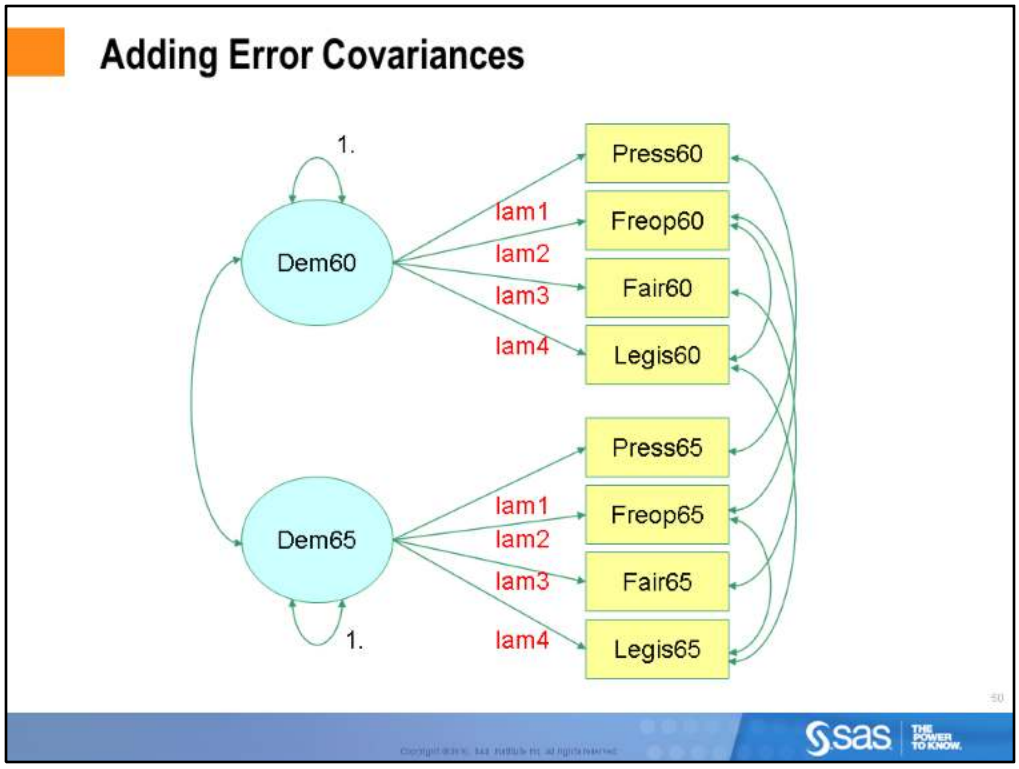
Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value	Pr > |t|
Dem65	Dem60	_Add9	0.97480	0.02682	36.3466	<.0001

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



Constraining parameters in the preceding example led to worse model fit. Now we will modify the model in the opposite way---adding more parameters to improve the model fit.



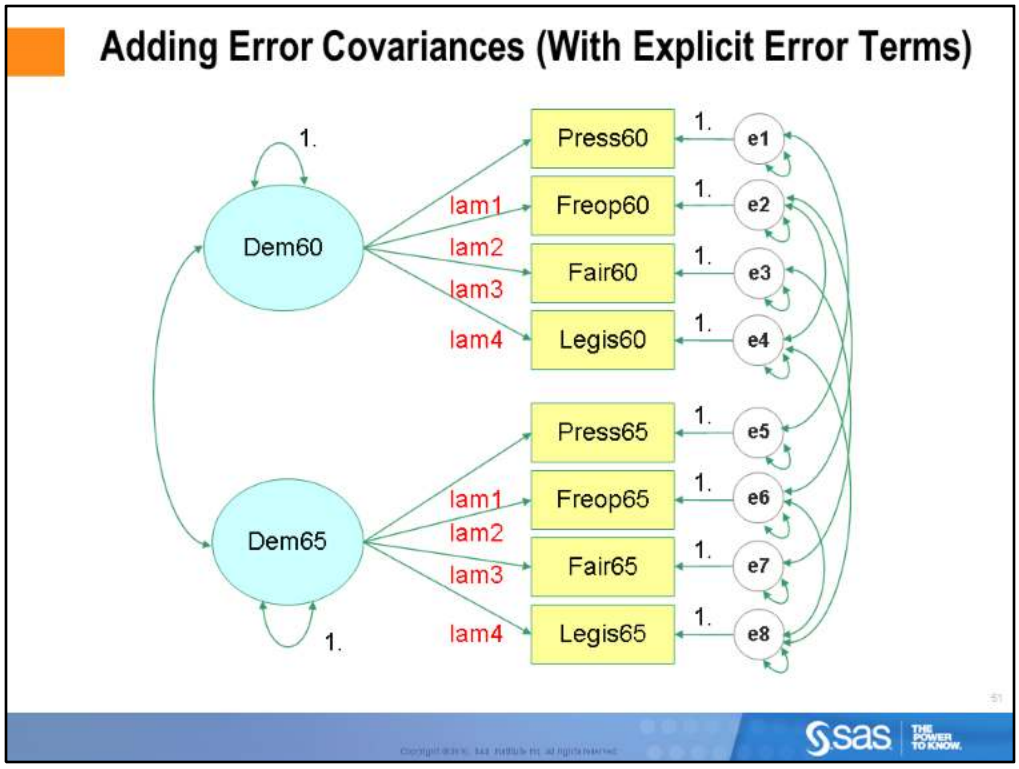
With the loading constraints, you observed a worse model fit.

With 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. Although adding more parameters to your model would improve the model fit, the drawback is that it makes your model more complicated, which is usually judged as an undesirable property of a sound model. 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 (Freop) and the elective nature of the legislative body (Legis) 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 the error covariance.

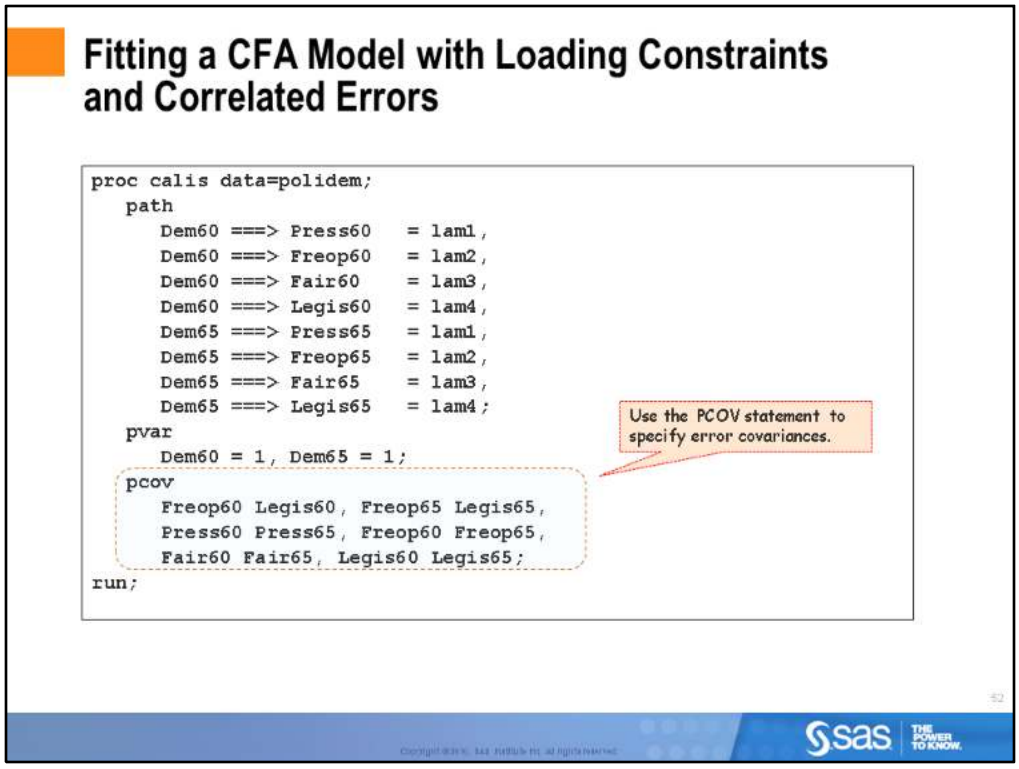
In addition, it is argued in Bollen that each of the variable pairs that were of the same nature but were measured at different times have a part of correlation that is beyond their common latent factors could explain. For example, Press60 and Press65 are connected by a double-headed arrow to represent 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.

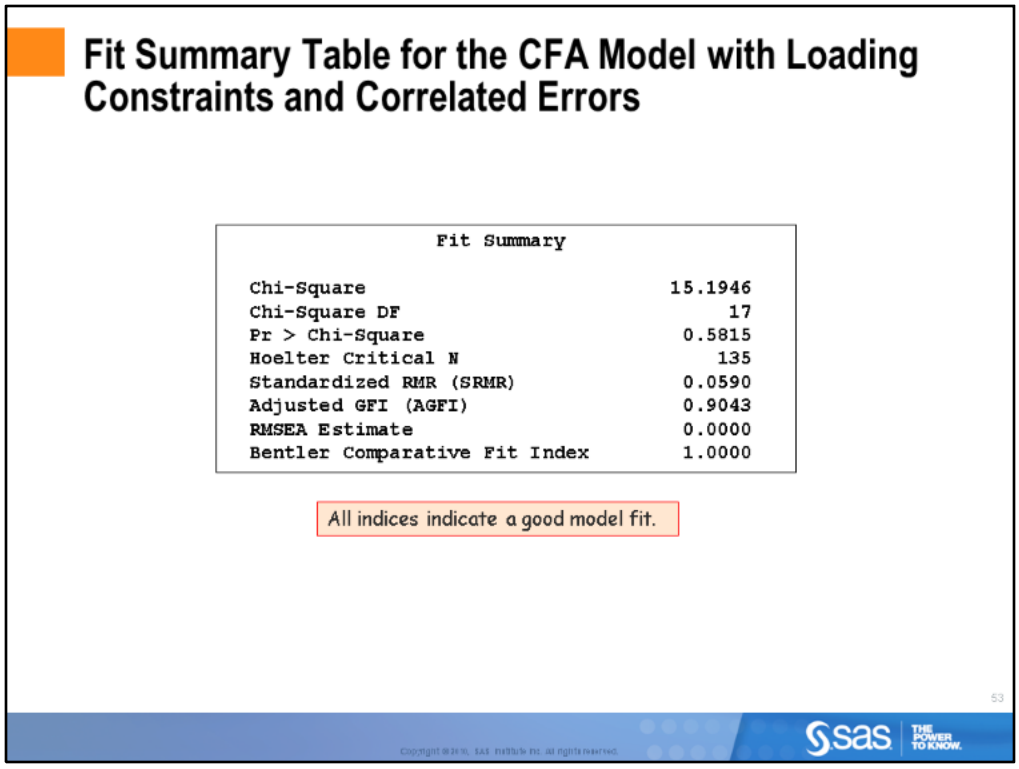
In this path diagram, error terms for the measured variables are shown. The double-headed arrows are shifted to the error terms. This makes it obvious that those double-headed arrows are covariances between the error variables (but not as partial covariances between the observed variables, as shown in 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, most of the time I would stick with the path diagram representation that does not use explicit error terms.



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

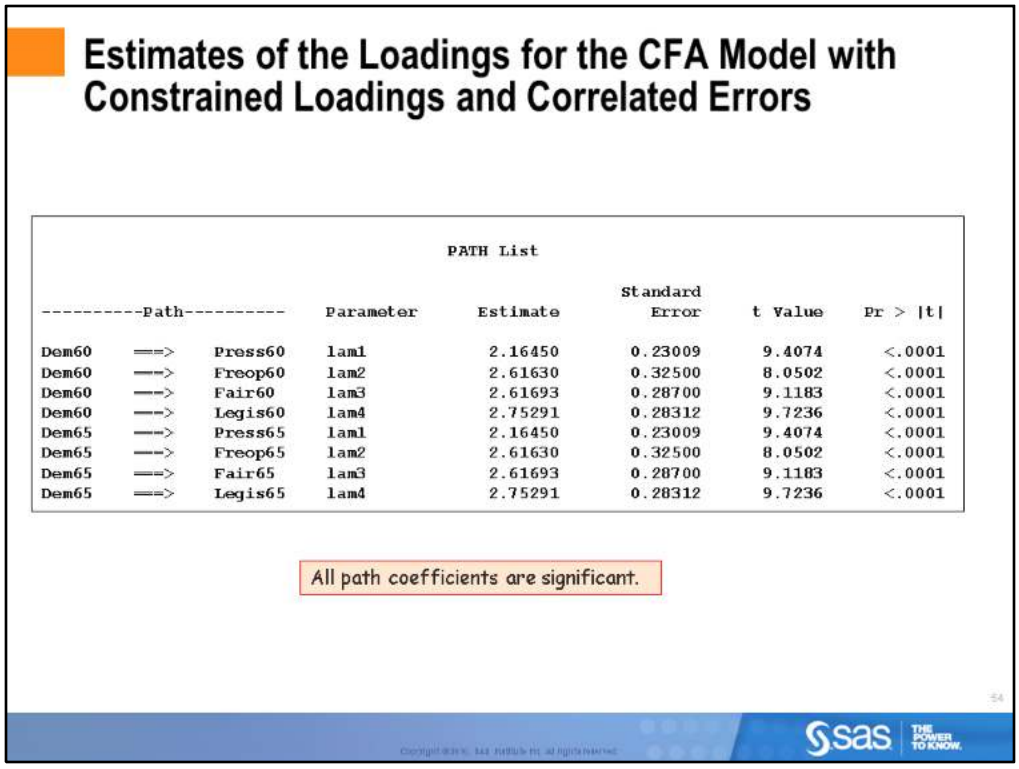
In the PATH modeling language, you can specify these covariance parameters in the PCOV statement. In this example, this means that you enumerate the six pairs of measured variables in the PCOV statement. For example, the first pair is Freop60 and Legis60, which represent a covariance parameter between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



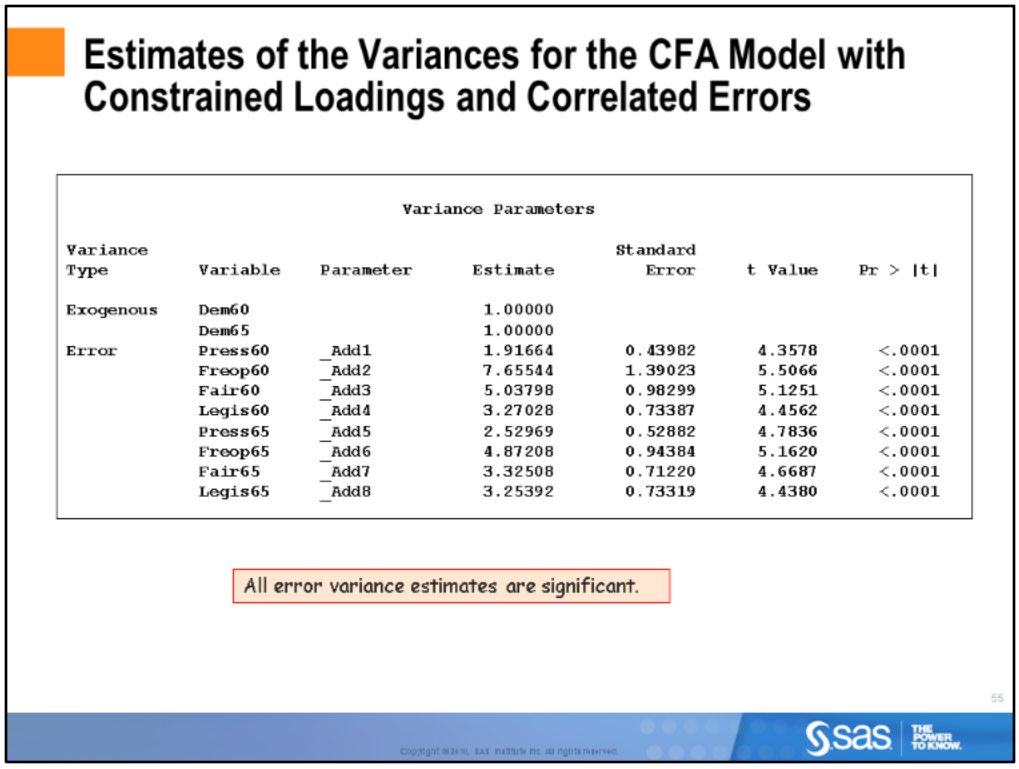
This model is supposed to fit better because of the added parameters for the error covariances.

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

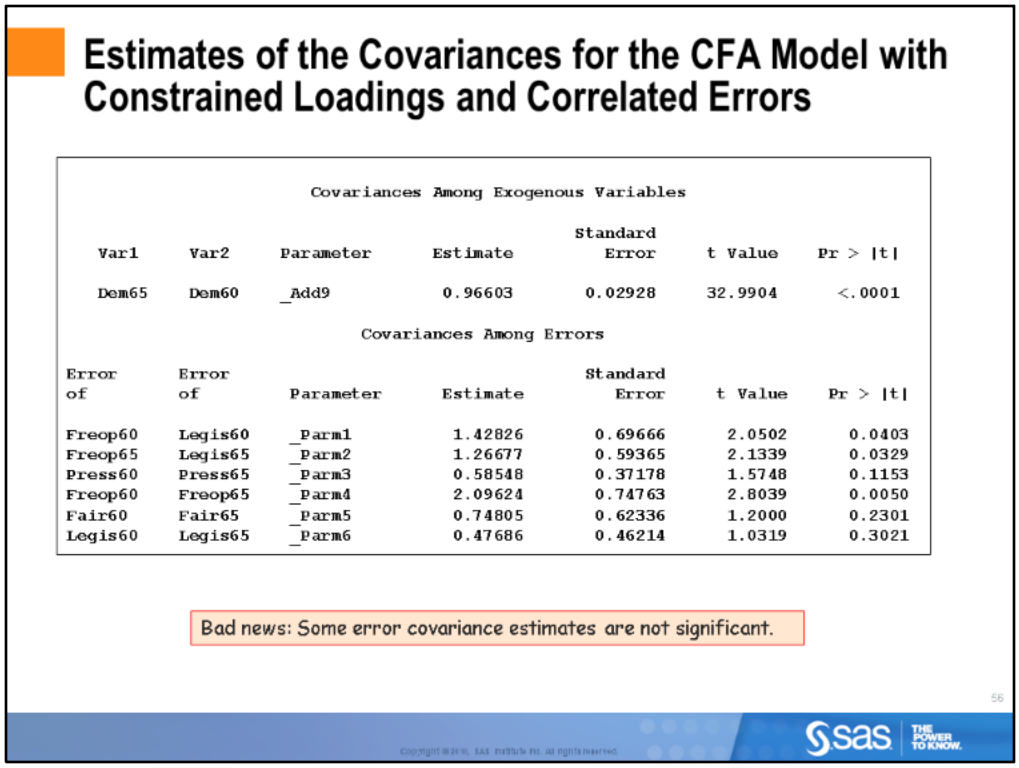
All other fit indices show good or excellent fit. The SRMR is 0.059, which is only slightly larger than the 0.05 criterion. The AGFI is 0.90, which is an indication of good model fit by convention. The RMSEA is essentially zero, which is the smallest RMSEA you could ever get. The CFI is 1, which is also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



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



All error variance estimates are significant.

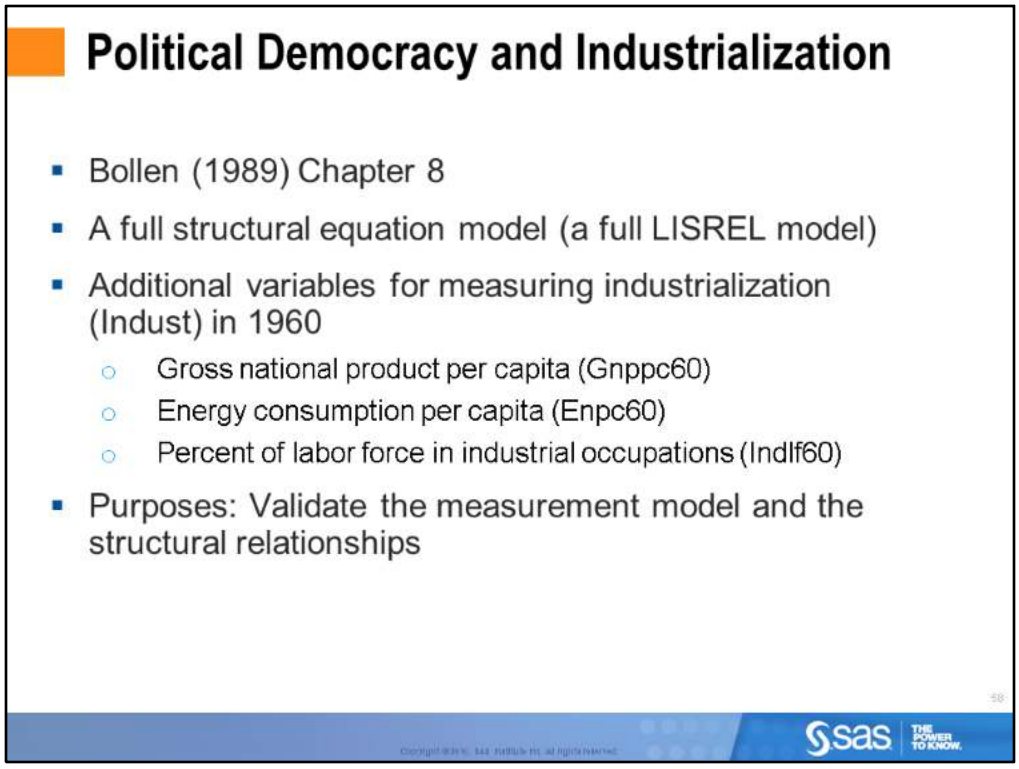


The first table shows the correlation between the two latent factors. Again, the correlation is very high and significant.

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

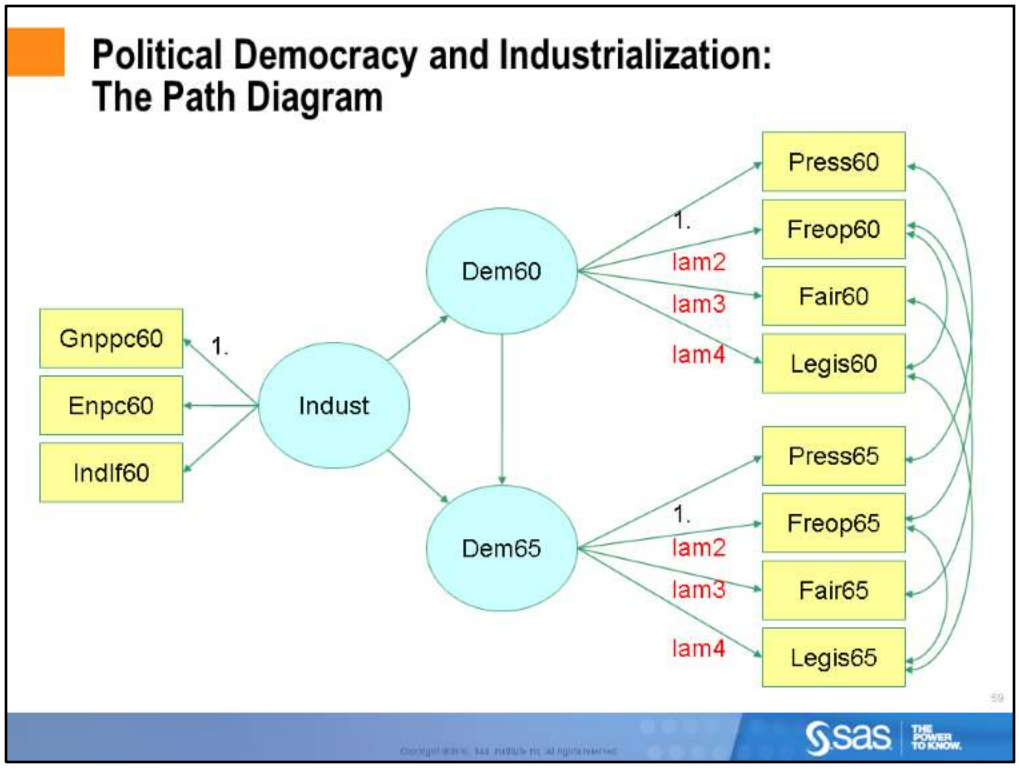


We continue with the previous model and add one more latent factor and its indicators into the model.

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, now 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 previously in the confirmatory factor models.

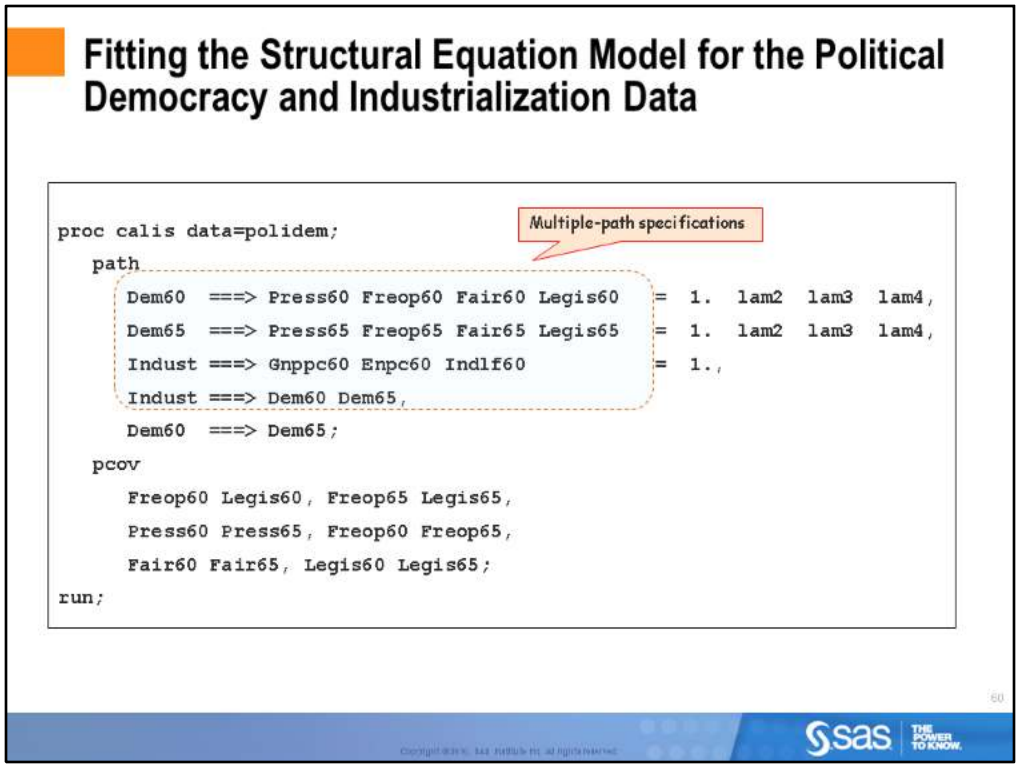


The entire SEM model is depicted in the path diagram of the current slide. The most notable addition is the paths from Indust to Dem60 and Dem65--- industrialization in 1960 serves as a predictor of democracy in both 1960 and 1965. Three observed variables serve as indicators of the industrialization: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

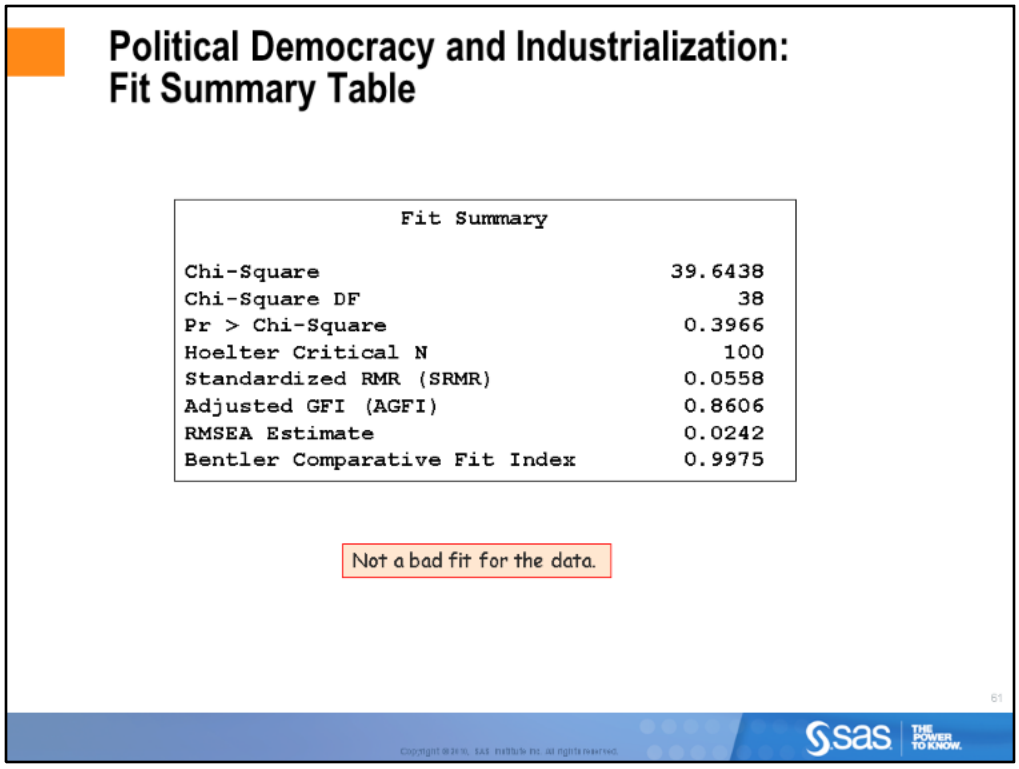


You can use PROC CALIS to specify this structural equation model easily.

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.



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

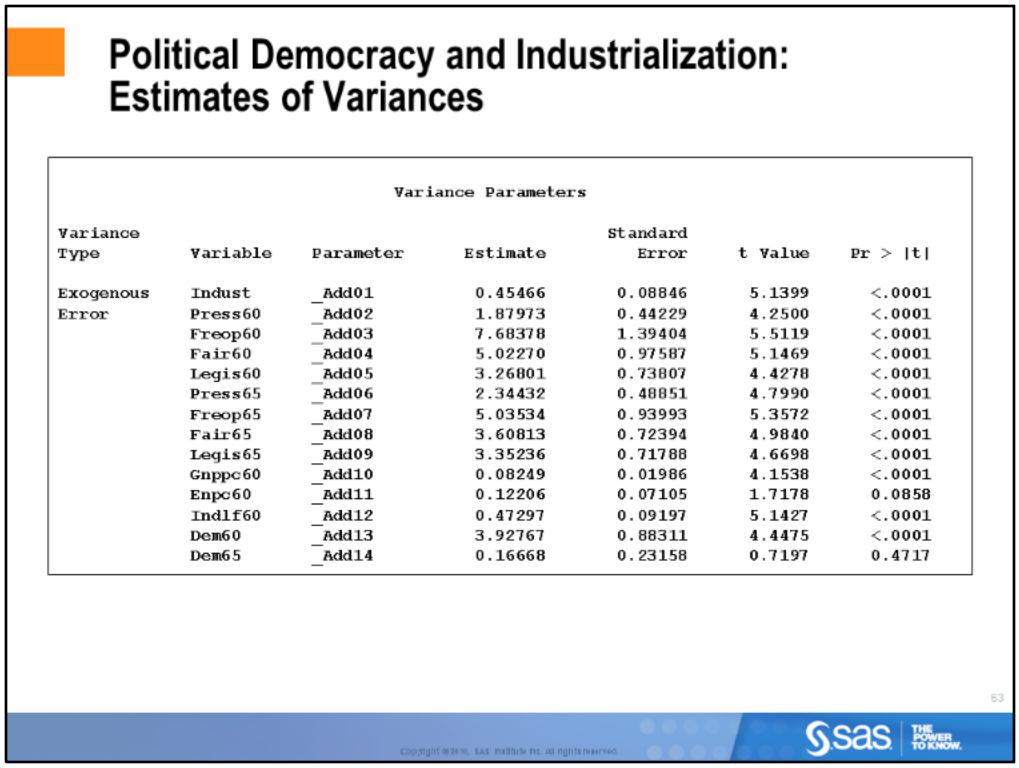
The model fit chi-square is not significant, supporting the hypothesized model. The SRMR is close to 0.05. The AGFI is 0.86, which shows a reasonable fit. The RMSEA indicates a very good model fit, as the value (.0242) is much lower than 0.05. The CFI is almost 1, which shows a perfect model fit.

Political Democracy and Industrialization: Estimates of Path Coefficients

PATH List

Path			Parameter	Estimate	Standard Error	t Value	Pr > |t|
Dem60	===>	Press60		1.00000			
Dem60	===>	Freop60	lam2	1.19079	0.14020	8.4934	<.0001
Dem60	===>	Fair60	lam3	1.17454	0.12121	9.6899	<.0001
Dem60	===>	Legis60	lam4	1.25099	0.11757	10.6401	<.0001
Dem65	===>	Press65		1.00000			
Dem65	===>	Freop65	lam2	1.19079	0.14020	8.4934	<.0001
Dem65	===>	Fair65	lam3	1.17454	0.12121	9.6899	<.0001
Dem65	===>	Legis65	lam4	1.25099	0.11757	10.6401	<.0001
Indust	===>	Gnppc60		1.00000			
Indust	===>	Enpc60	_Parm01	2.17966	0.13932	15.6453	<.0001
Indust	===>	Indlf60	_Parm02	1.81821	0.15290	11.8913	<.0001
Indust	===>	Dem60	_Parm03	1.47133	0.39496	3.7253	0.0002
Indust	===>	Dem65	_Parm04	0.60046	0.22722	2.6427	0.0082
Dem60	===>	Dem65	_Parm05	0.86504	0.07538	11.4765	<.0001

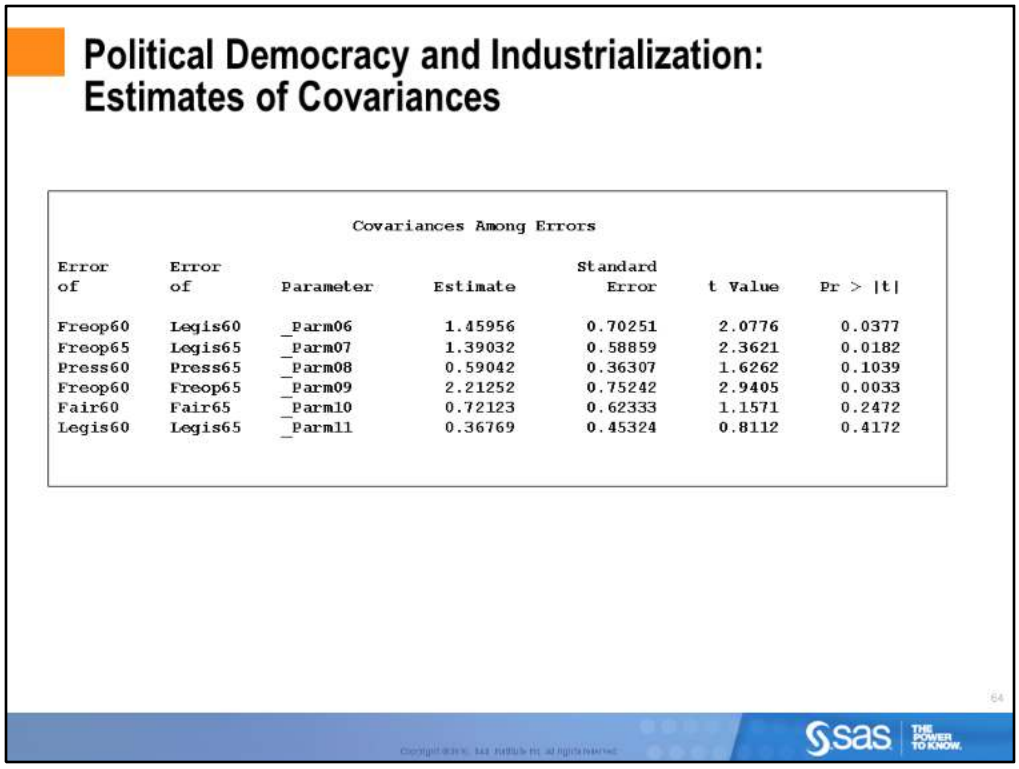
All path coefficients are significant---a pretty good sign.



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 almost 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.

Unlike insignificant error covariances, insignificant error variances are not serious concerns. Insignificant error covariances challenge the proposed model with "wastebasket" parameters to boost model fit, while insignificant error variances only means that predictor and outcome relationships might be nearly perfect.



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

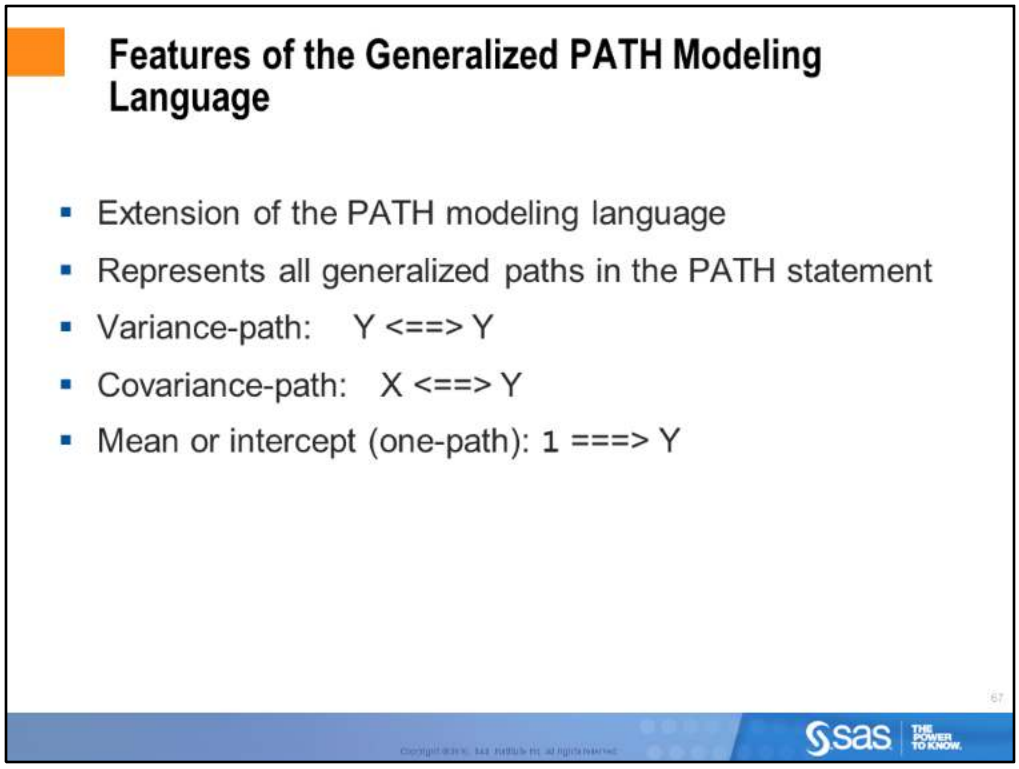
Variable	Error Variance	Total Variance	R-Square
Enpc60	0.12206	2.28211	0.9465
Fair60	5.02270	11.79895	0.5743
Fair65	3.60813	10.09361	0.6425
Freop60	7.68378	14.64875	0.4755
Freop65	5.03534	11.70144	0.5697
Gnppc60	0.08249	0.53715	0.8464
Indlf60	0.47297	1.97602	0.7606
Legis60	3.26801	10.95502	0.7017
Legis65	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

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 a small 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.

The Extended PATH Modeling Language

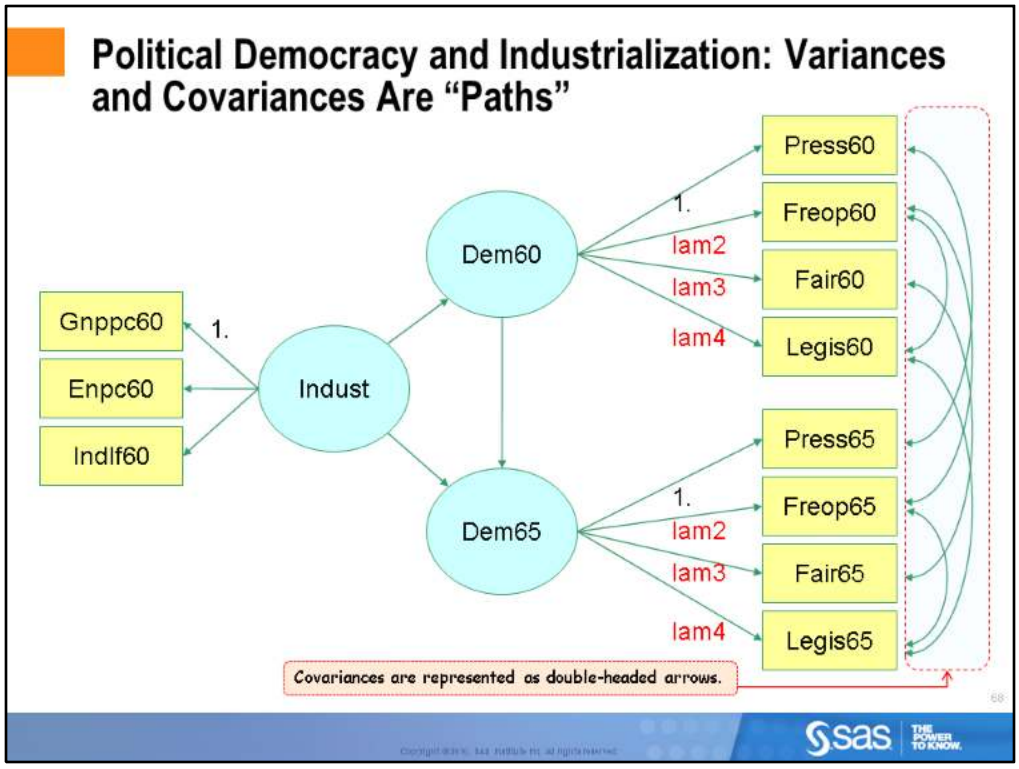


In sum, the generalized path modeling language enables you to specify all types of arrows in the path diagram as "paths," including the variance, covariance, intercept, and mean parameters.

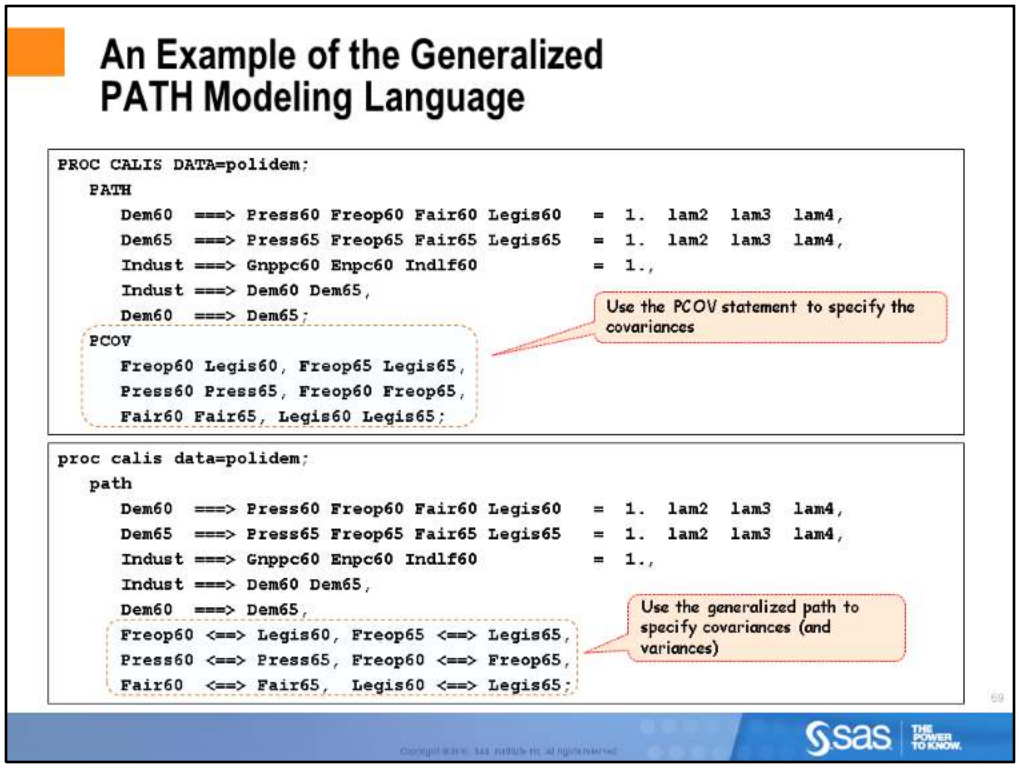
Variance of Y is a path like Y <==> Y.

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

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



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.



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. They will generate the same estimation results.

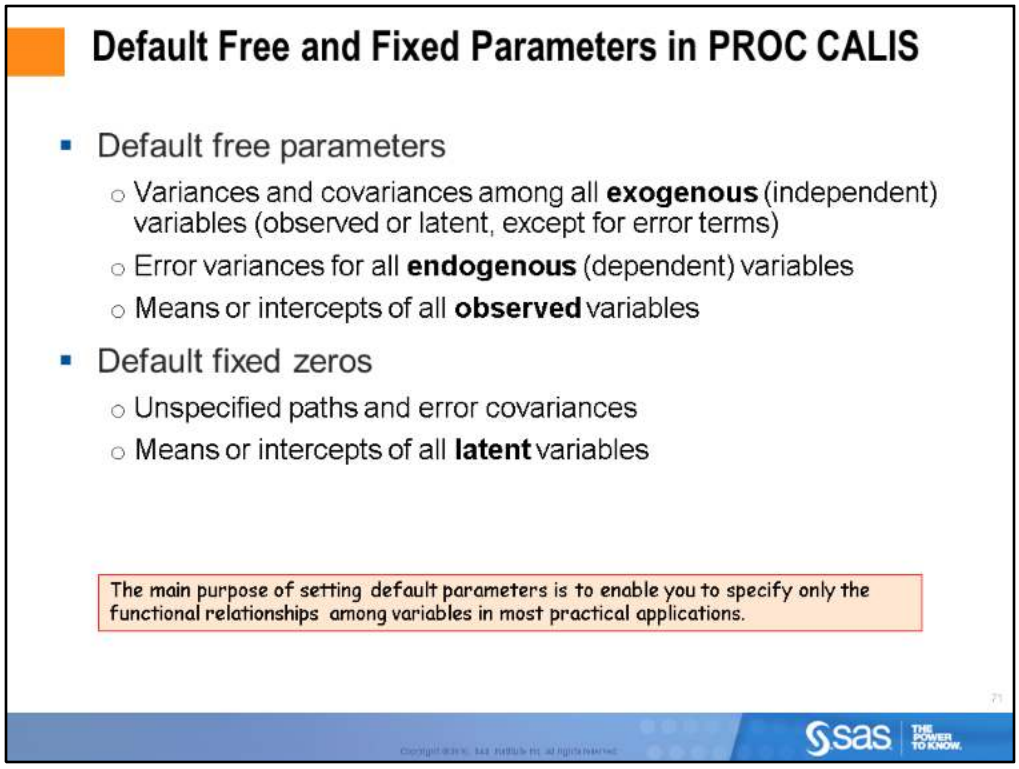
Political Democracy and Industrialization: Output with Generalized Paths

PATH List

Path			Parameter	Estimate	Standard Error	t Value	Pr > |t|
Dem60	===>	Press60		1.00000			
Dem60	===>	Freop60	lam2	1.19079	0.14020	8.4934	<.0001
Dem60	===>	Fair60	lam3	1.17454	0.12121	9.6899	<.0001
Dem60	===>	Legis60	lam4	1.25099	0.11757	10.6401	<.0001
Dem65	===>	Press65		1.00000			
Dem65	===>	Freop65	lam2	1.19079	0.14020	8.4934	<.0001
Dem65	===>	Fair65	lam3	1.17454	0.12121	9.6899	<.0001
Dem65	===>	Legis65	lam4	1.25099	0.11757	10.6401	<.0001
Indust	===>	Gnppc60		1.00000			
Indust	===>	Enpc60	_Parm01	2.17966	0.13932	15.6453	<.0001
Indust	===>	Indlf60	_Parm02	1.81821	0.15290	11.8913	<.0001
Indust	===>	Dem60	_Parm03	1.47133	0.39496	3.7253	0.0002
Indust	===>	Dem65	_Parm04	0.60046	0.22722	2.6427	0.0082
Dem60	===>	Dem65	_Parm05	0.86504	0.07538	11.4765	<.0001
Freop60	<==>	Legis60	_Parm06	1.45956	0.70251	2.0776	0.0377
Freop65	<==>	Legis65	_Parm07	1.39032	0.58859	2.3621	0.0182
Press60	<==>	Press65	_Parm08	0.59042	0.36307	1.6262	0.1039
Freop60	<==>	Freop65	_Parm09	2.21252	0.75242	2.9405	0.0033
Fair60	<==>	Fair65	_Parm10	0.72123	0.62333	1.1571	0.2472
Legis60	<==>	Legis65	_Parm11	0.36769	0.45324	0.8112	0.4172

The results obtained from PROC CALIS now shows the covariance estimates as “paths” in the PATH list.

This is where the extended path modeling language might be very useful---it shows all estimates in the same table so that you can report all the SEM estimates directly in your research paper.



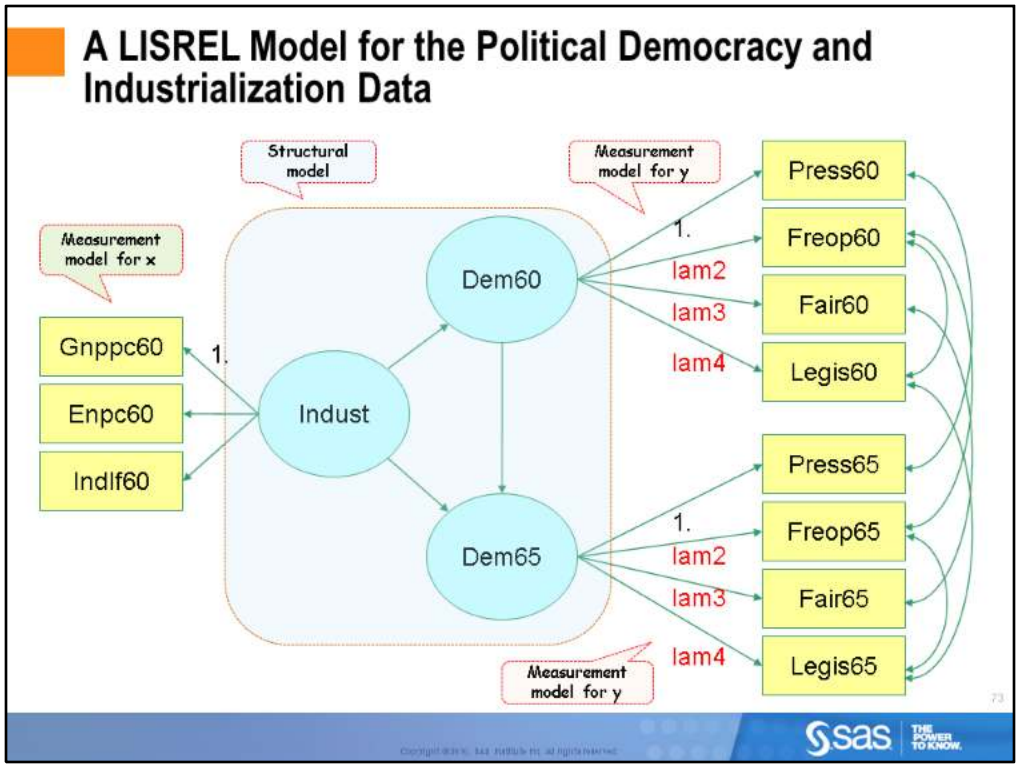
Knowing the default free and fixed parameters in PROC CALIS are useful because it enhances the coding efficiency and accuracy. Here is a list of default free parameters and fixed zeros used in PROC CALIS:

(Note: 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.

LISREL Models



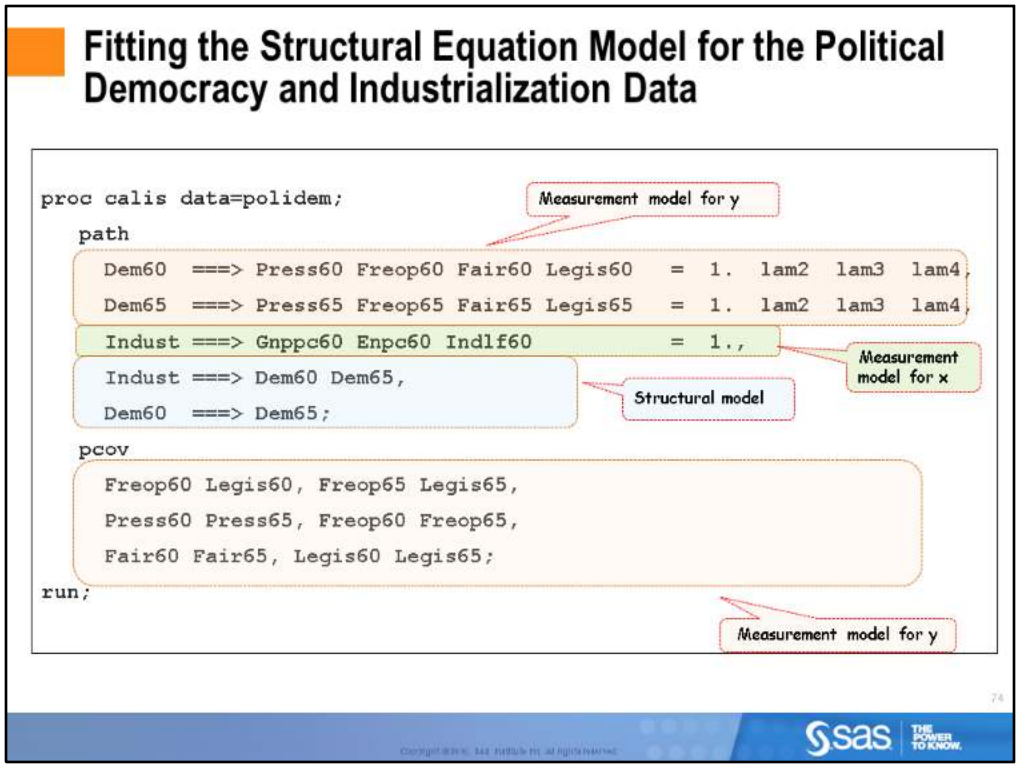
The preceding full SEM model is also a good illustration of the LISREL model.

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 that involves industrialization is the measurement model of x because Indust serves as an exogenous (independent) factor in the path diagram. The measurement model that involves Dem60 and Dem65 is the measurement model of y because Dem60 and Dem65 are endogenous (dependent) factors 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 components of the LISREL model are identified in the current path diagram.

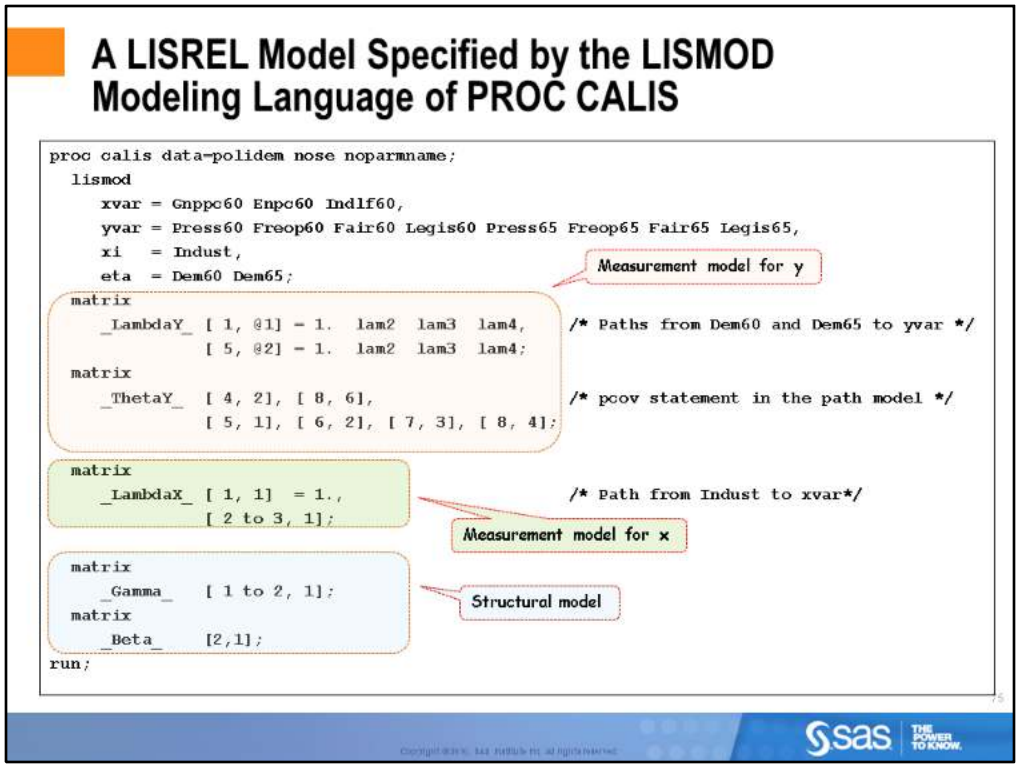


In the PATH modeling language, you can also identify the code for the measurement models and the structural model. The preceding code is recited here for illustrations.

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 path 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, now you might have a clue to specify the LISREL model in PROC CALIS. Because the same path diagram is being used by the PATH modeling language and the LISREL model, the only task here is to transcribe the code in the PATH modeling language to the language for the LISREL model---that is, you need to specify the measurement and structural models in terms of LISREL model matrices.



PROC CALIS supports the so-called LISMOD modeling language for specifying LISREL models. In order to fully understand the PROC CALIS code for specifying the LISREL model, knowledge about matrix algebra is needed. But here I 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: a list of observed indicators for the exogenous (independent) latent factors in the model.
2. y-variables: a list of observed indicators for the endogenous (dependent) latent factors in the model.
3. xi-variables: a list of exogenous (independent) latent factors in the model.
4. eta-variables: a list of endogenous (dependent) latent factors in the model.

Traditionally, 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.

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.

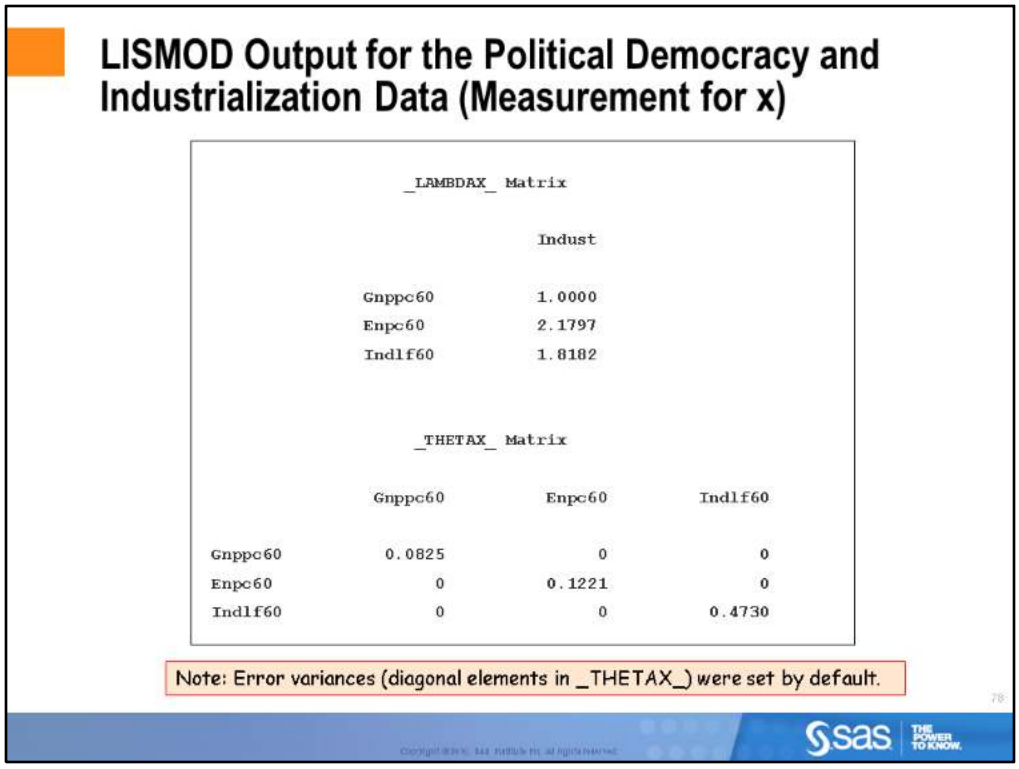
LISMOD Output for the Political Democracy and Industrialization Data (Measurement for y)

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.



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

This slide shows the functional relationships between latent constructs.

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

PSI Matrix

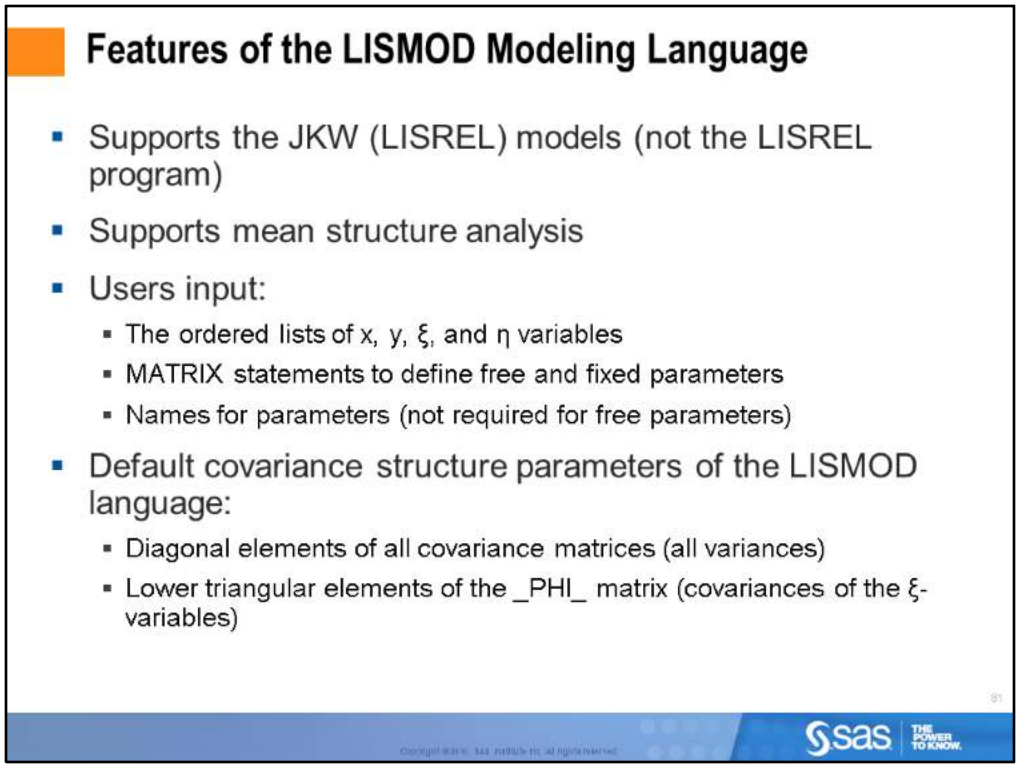
	Dem60	Dem65
Dem60	3.9277	0
Dem65	0	0.1667

PHI Matrix

	Indust
Indust	0.4547

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

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



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 statements for the mean model matrices in the LISREL model.

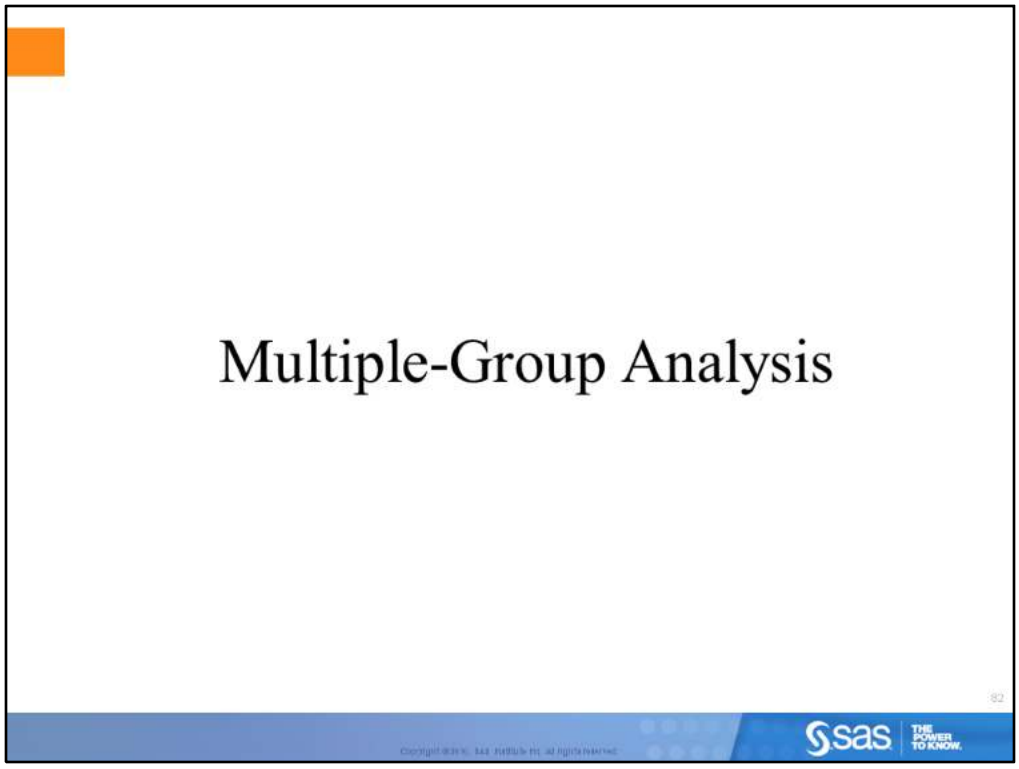
If you understand the LISREL model, here are three things you input by using the LISMOD language:
1. The ordered lists of x, y, ξ, and η variables
2. MATRIX statements to define free and fixed parameters
3. Names for parameters (not required for free parameters)

The Default covariance structure parameters in the LISMOD language are:

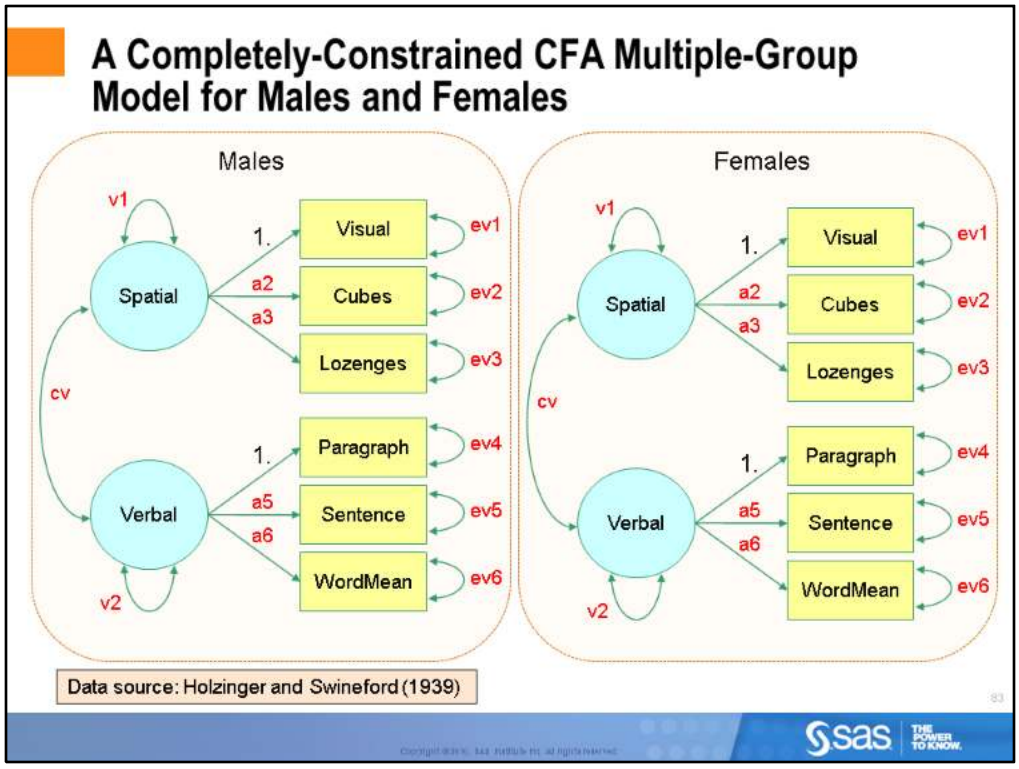
1. Diagonal elements of all covariance matrices (all variances)
2. Lower triangular elements of the _PHI_ matrix (covariances of the ξ- variables)

Specifying the default parameters explicitly is not necessary but 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.



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

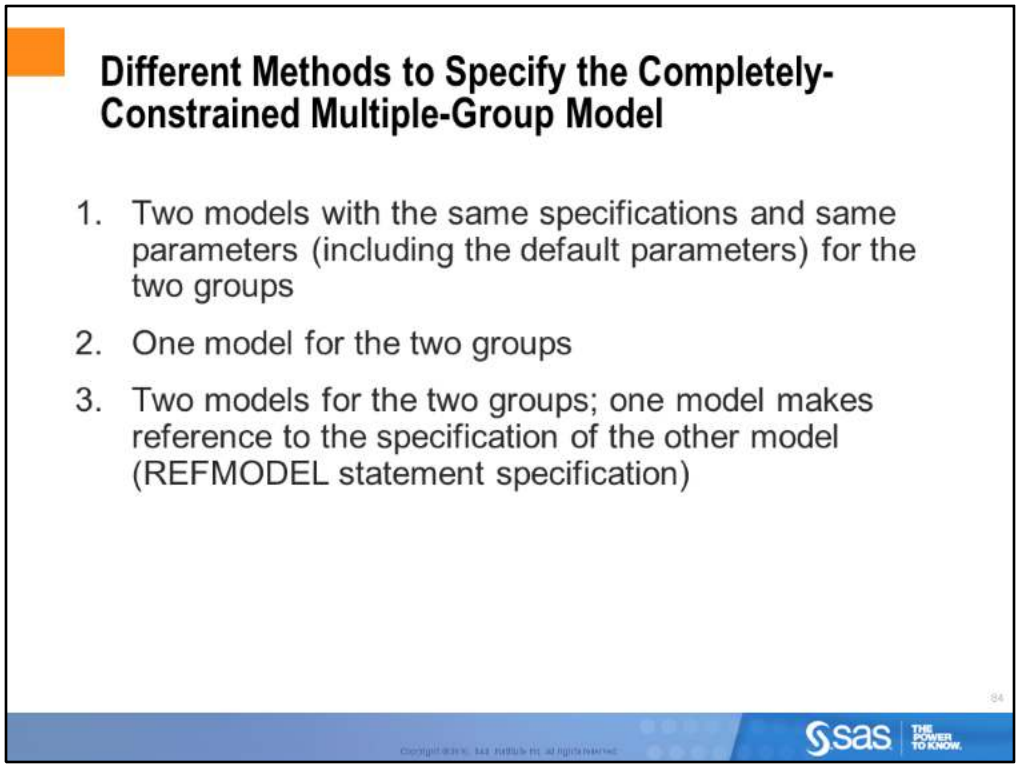


Let us use an example to illustrate the multiple-group analysis. The Holzinger and Swineford (1939) data are used. An application to this data set is also demonstrated in Arbuckle's AMOS manual (2008).

In this research, visual and verbal test scores were observed. Visual, Cubes, and Lozenges are spatial tests that measure spatial ability. Paragraph, Sentence, and WordMean are verbal tests that measure verbal ability. CFA models were hypothesized for the two groups: one group is for males and the other for females.

The diagrams in this slide show that the models for the two gender groups are exactly the same. That is, the factor structures for the groups are the same and all parameters in the two models for the groups are the same. The parameters are labeled in red in the path diagram. The set of all parameters includes factor loadings a2, a3, a5, and a6; error variances ev1-ev6; and factor variances and covariances v1, v2, and cv.

Analyzing the models for the two groups together form a multiple-group analysis where the two groups are fitted exactly by the same model. This will be called a completely constrained multiple-group model. How do we test the fit of this completely constrained multiple-group model to the data? And, if this model does not fit, how do we test multiple-group model with partial constraints?

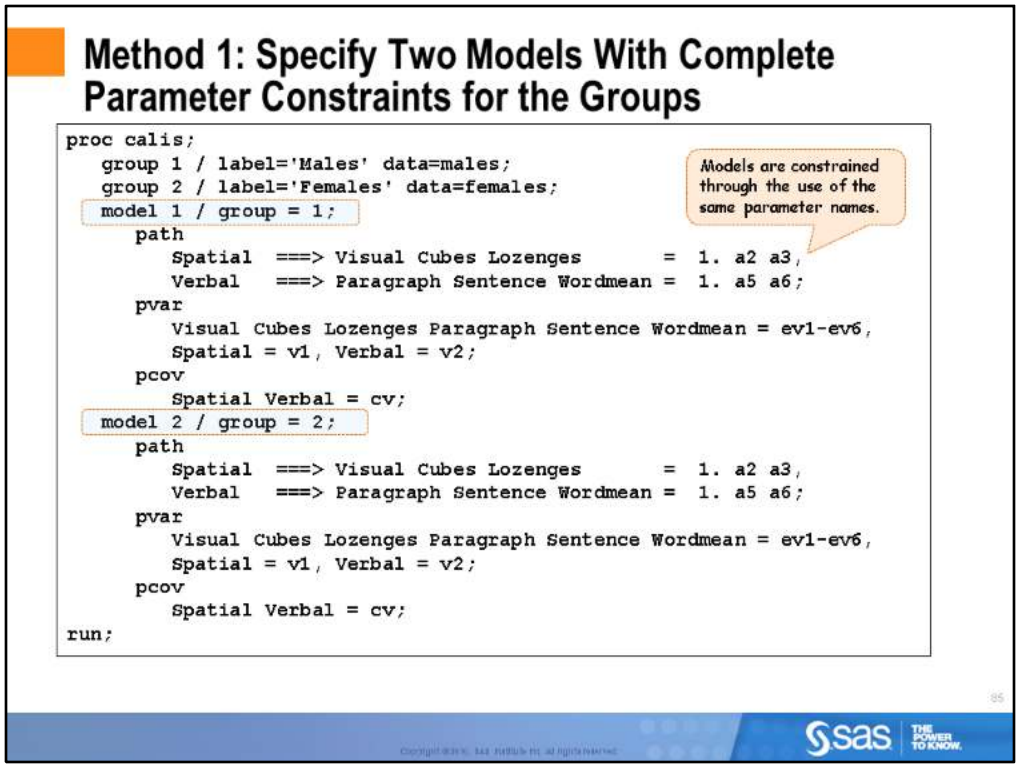


Let us focus on the completely-constrained multiple-group model first.

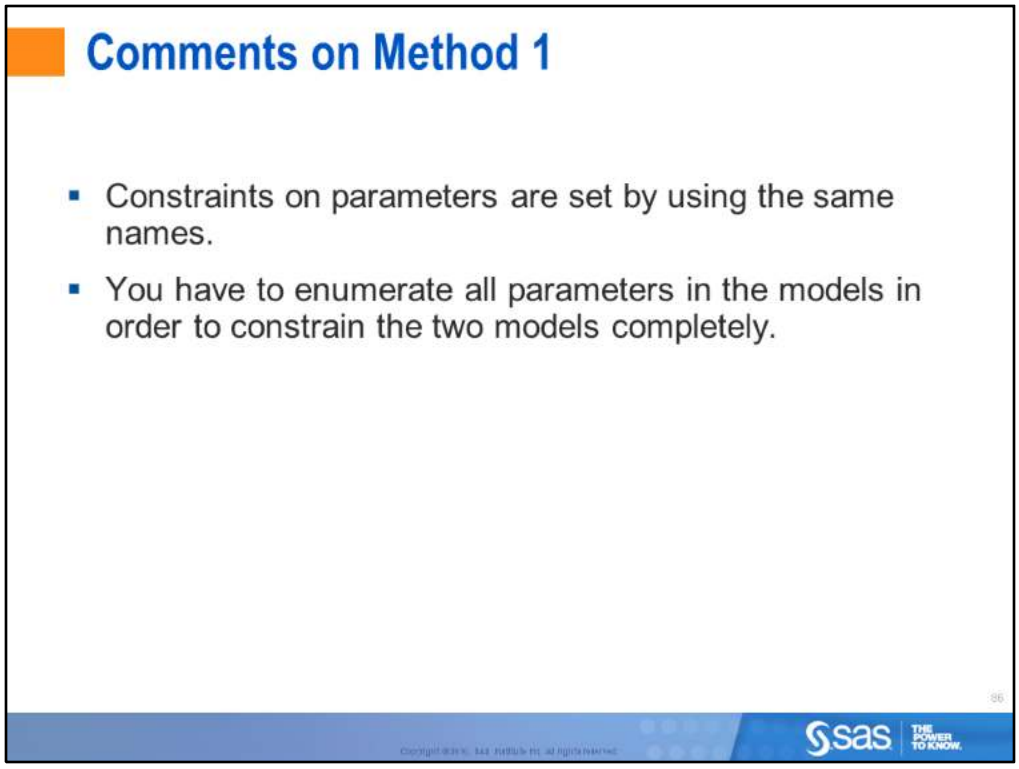
In PROC CALIS, you can use one of the following three ways to specify the preceding completely-constrained model:

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

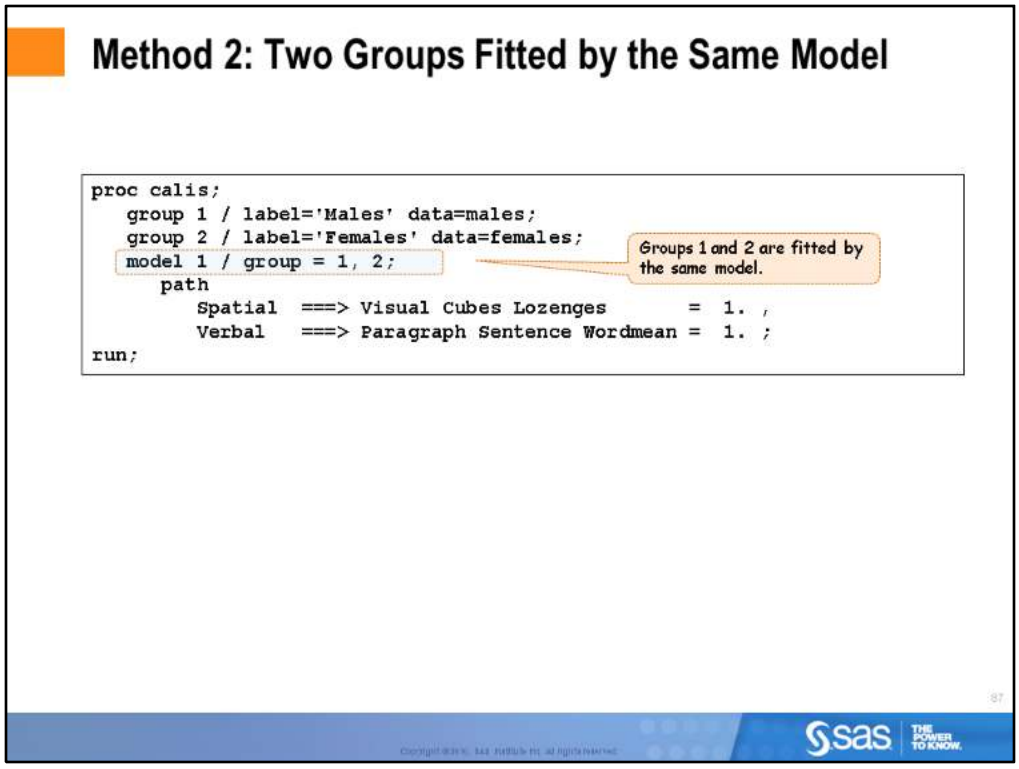
I will describe each of these methods. All will give you the same estimation results.



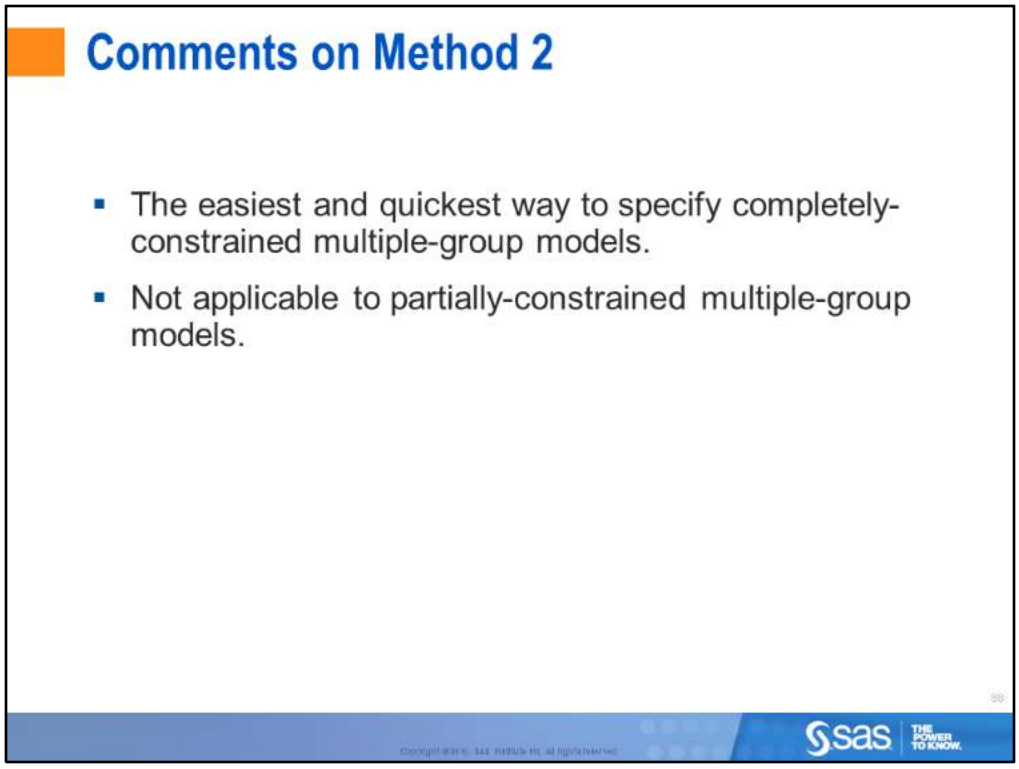
The first method is to define two models for the two groups. The model specifications under the two MODEL statements must be exactly the same.



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.

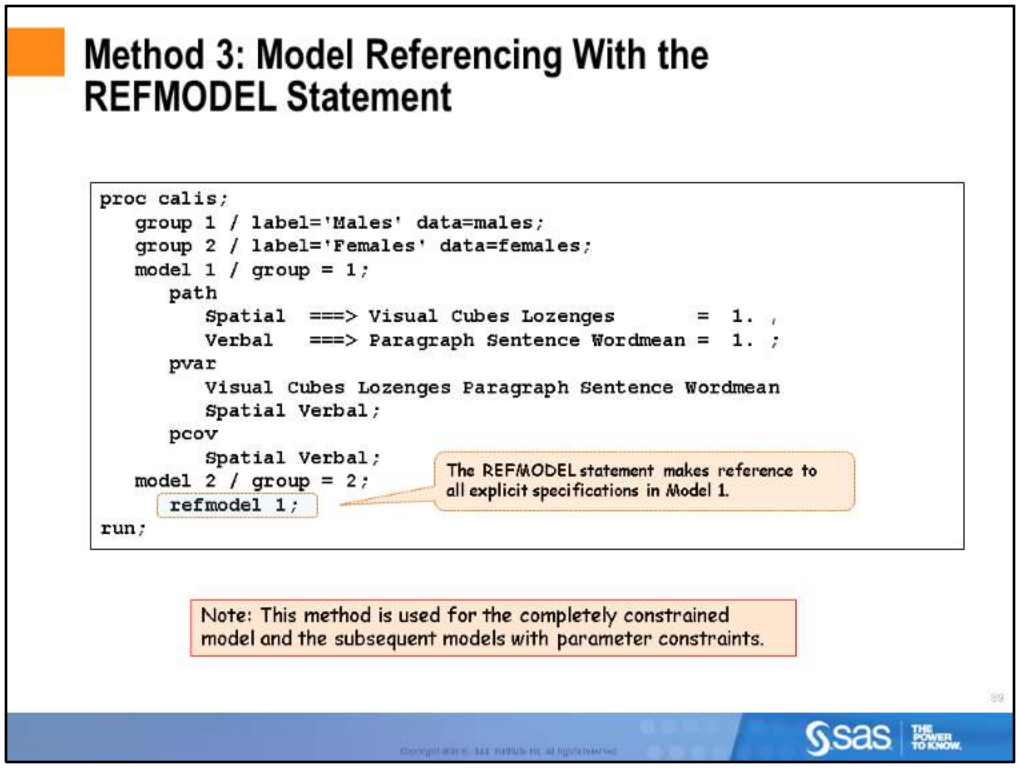


The second method is very simple and intuitive. You specify one model and fit this model to the two gender groups. This ensures the groups are fitted exactly by the same model.



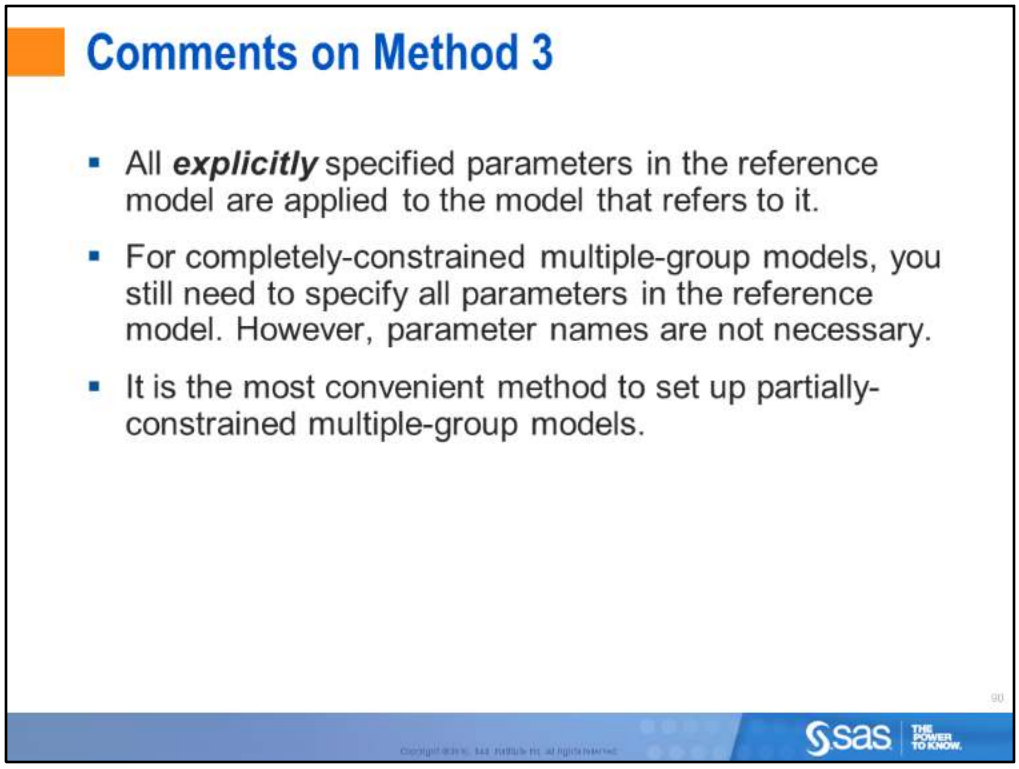
The advantage of this method is that it is simple, intuitive, and no parameter names are necessary for constraining models. Also, you do not need to specify any of the default parameters explicitly for setting up constraints.

This is an ideal specification method if the completely-constrained multiple-group model is all you want to fit. However, if you are going to fit a sequence of multiple-group models (including partially constrained models), you might want to consider the next method.



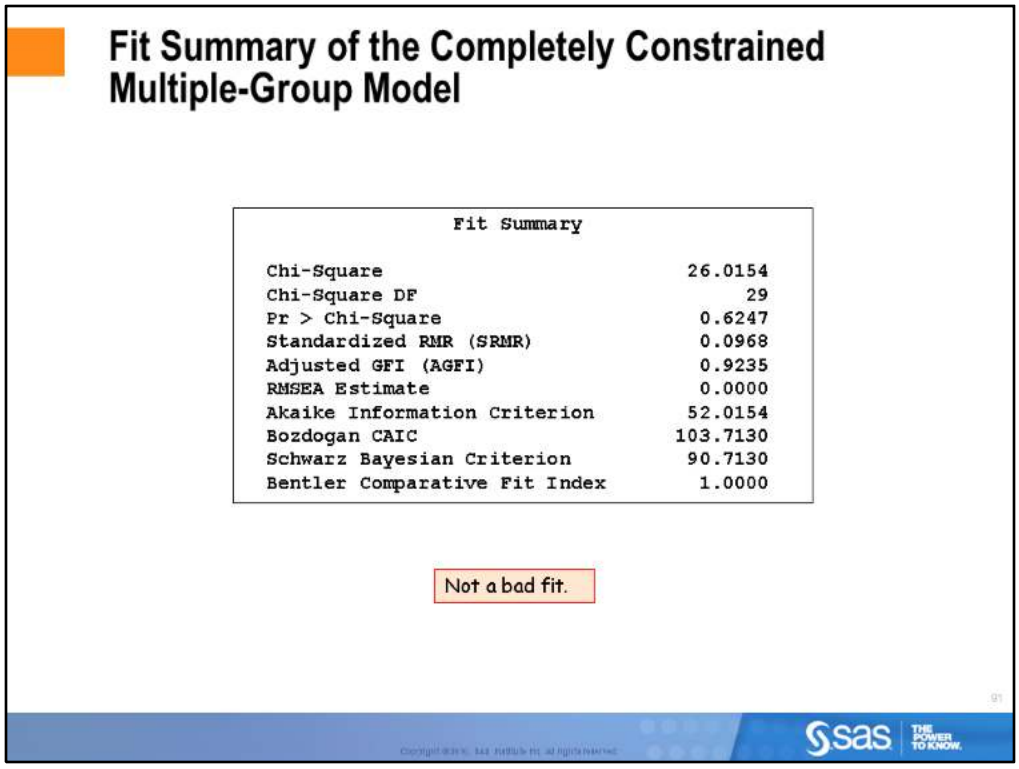
The third method constrains the models by the REFMODEL statement. The REFMODEL makes reference to all the **explicit** specifications in the reference model. This means that all explicit specifications in the reference model are duplicated to the current model.

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. Therefore, Model 1 and Model 2 are exactly the same---in other words, they are completely constrained.

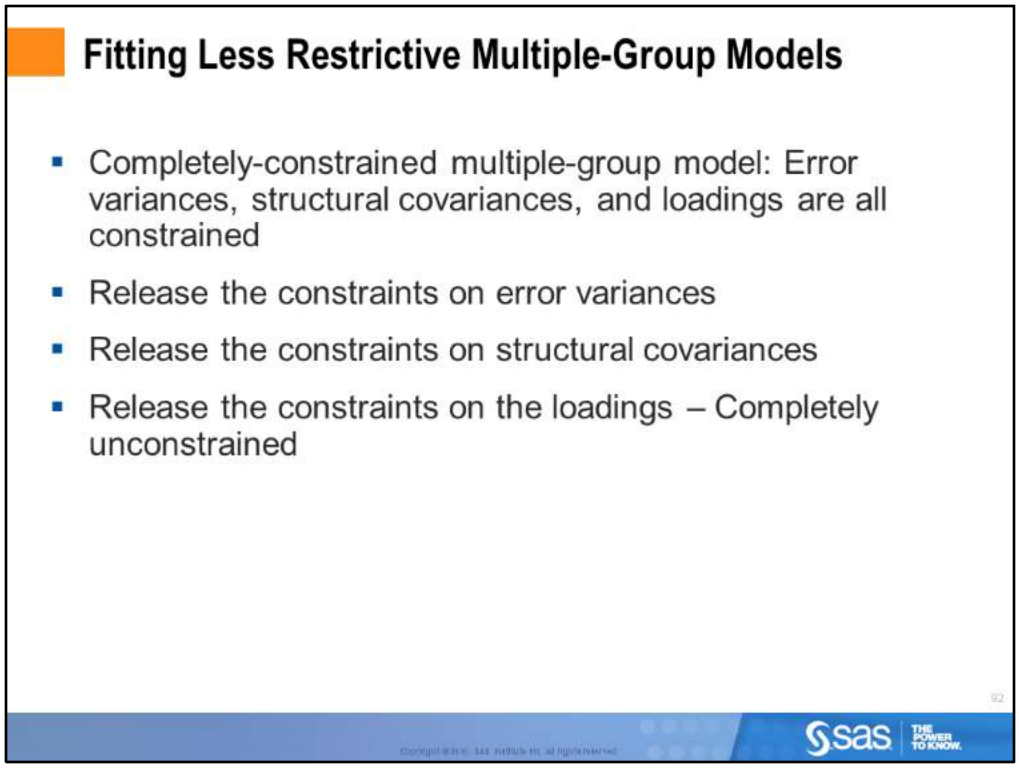


It is important to recognize that in order to completely constrain the two models for the two groups, all parameters, including those could have been set by default by PROC CALIS (e.g., specifications in the PVAR statement and PCOV statement), must be specified **explicitly** in Model 1. That way Model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

Like Method 1, the use of REFMODEL statement in Method 3 for specifying completely constrained models requires the enumeration of all parameters. However, Unlike Method 1, method 3 does not require the use of parameter names for setting constraints across models. Constraints are done via the REFMODEL statement. Although not as intuitive as Method 2, this method would be more useful if you need to fit a sequence of multiple-group models, which will be illustrated later.



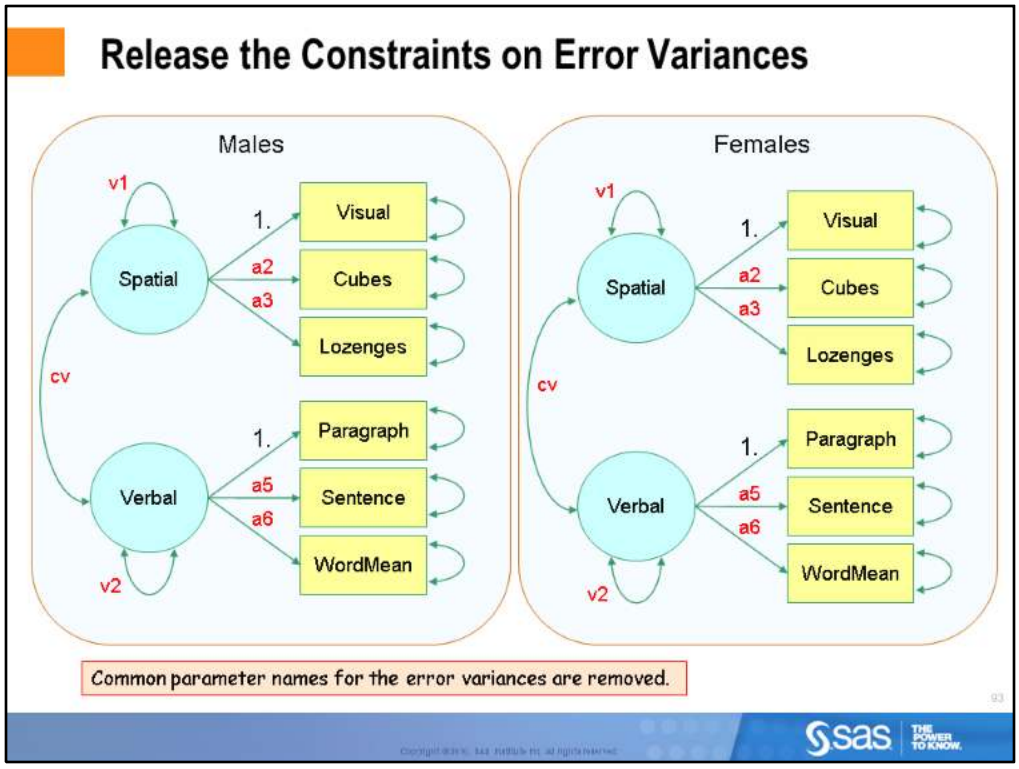
The completely-constrained multiple-group model provide a good fit of the data. The model fit chi-square is not significant. The RMSEA is perfect, although the SRMR is not very good. The AGFI and the CFI are also good. The AIC, the CAIC, and the SBC are also included in this table. These indices cannot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



We have fitted the completely constrained multiple-group model by the REFMODEL method (Method 3). We can fit less constrained multiple-group model by modifying our PROC CALIS code.

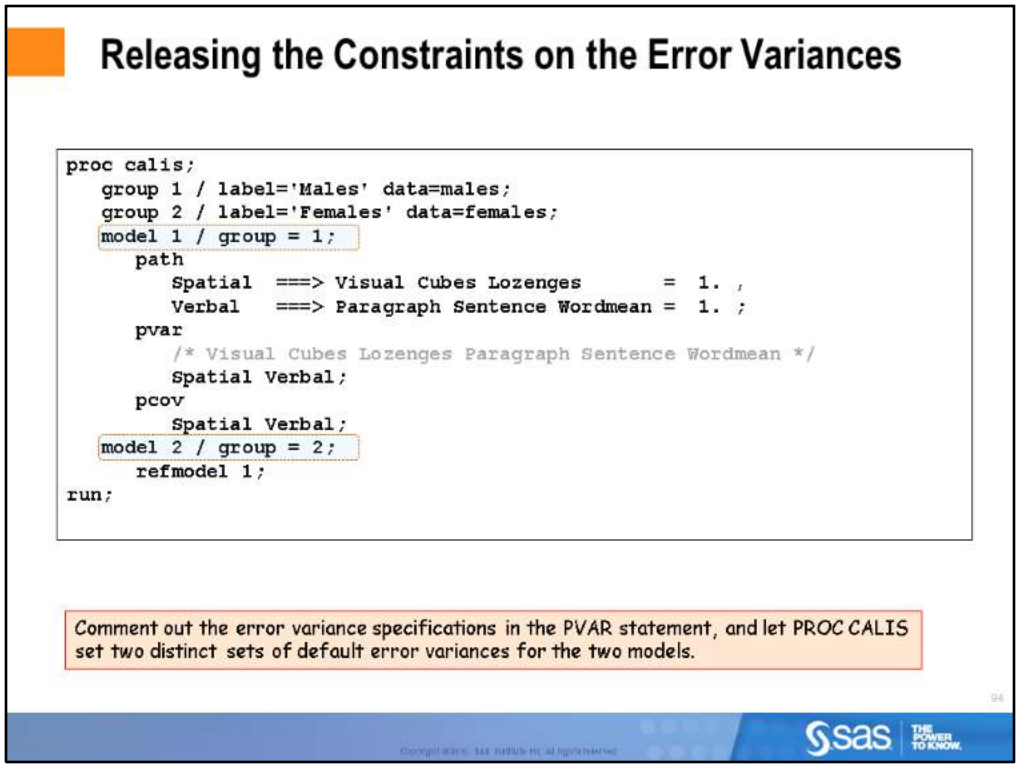
We can release the constraints on error variances. Then we can release the constraints on the structural covariances (among latent variables). Finally, we can release the constraints on the path coefficients (or loadings).

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.



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.

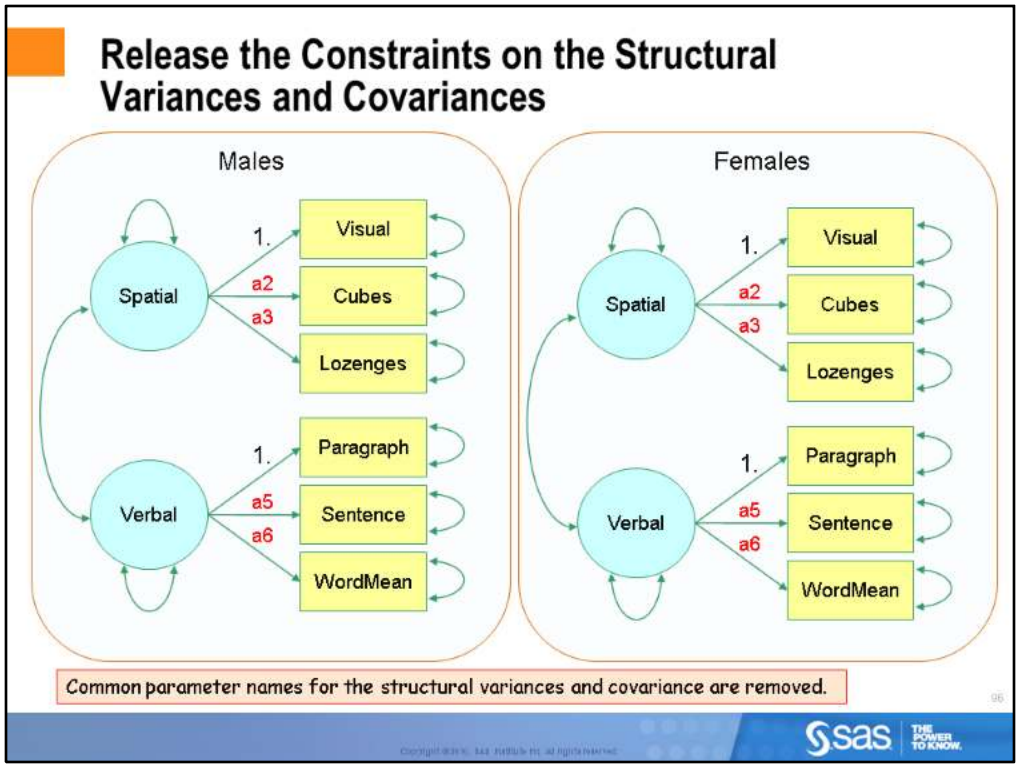
This could be done very easily by modifying from the completely constrained multiple-group model. All you need to do is to comment out the PVAR statement specifications for the observed variables.

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, they are 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

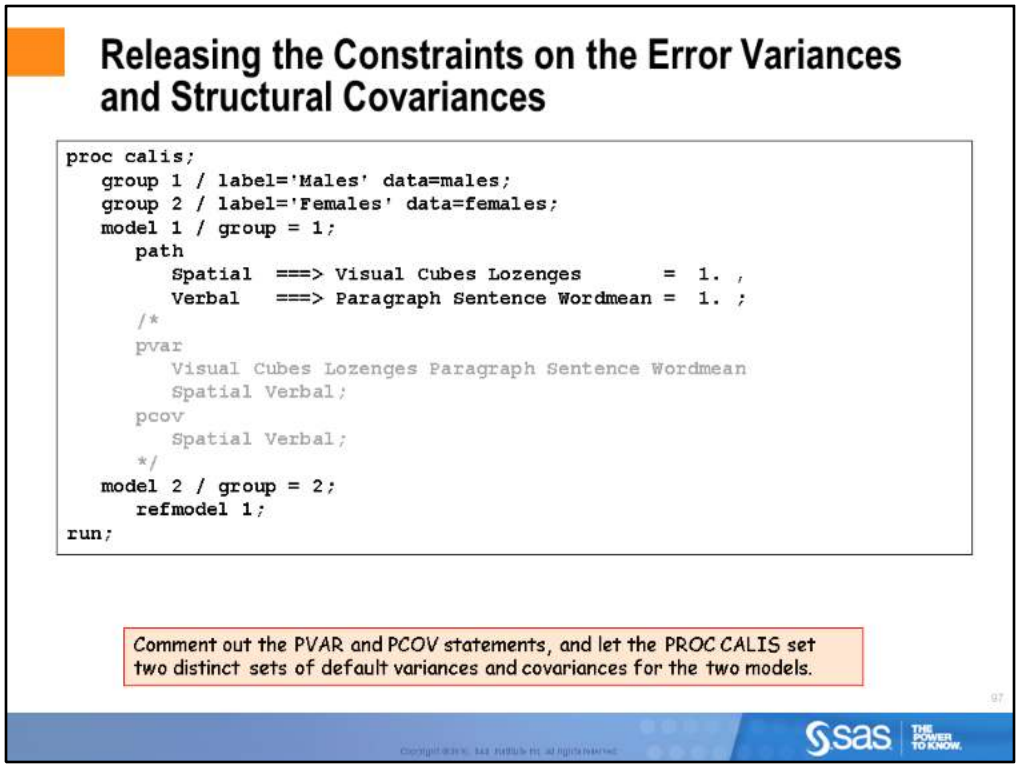
Fit Summary	
Chi-Square	22.0334
Chi-Square DF	23
Pr > Chi-Square	0.5182
Standardized RMR (SRMR)	0.0903
Adjusted GFI (AGFI)	0.9163
RMSEA Estimate	0.0000
Akaike Information Criterion	60.0334
Bozdogan CAIC	135.5913
Schwarz Bayesian Criterion	116.5913
Bentler Comparative Fit Index	1.0000

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



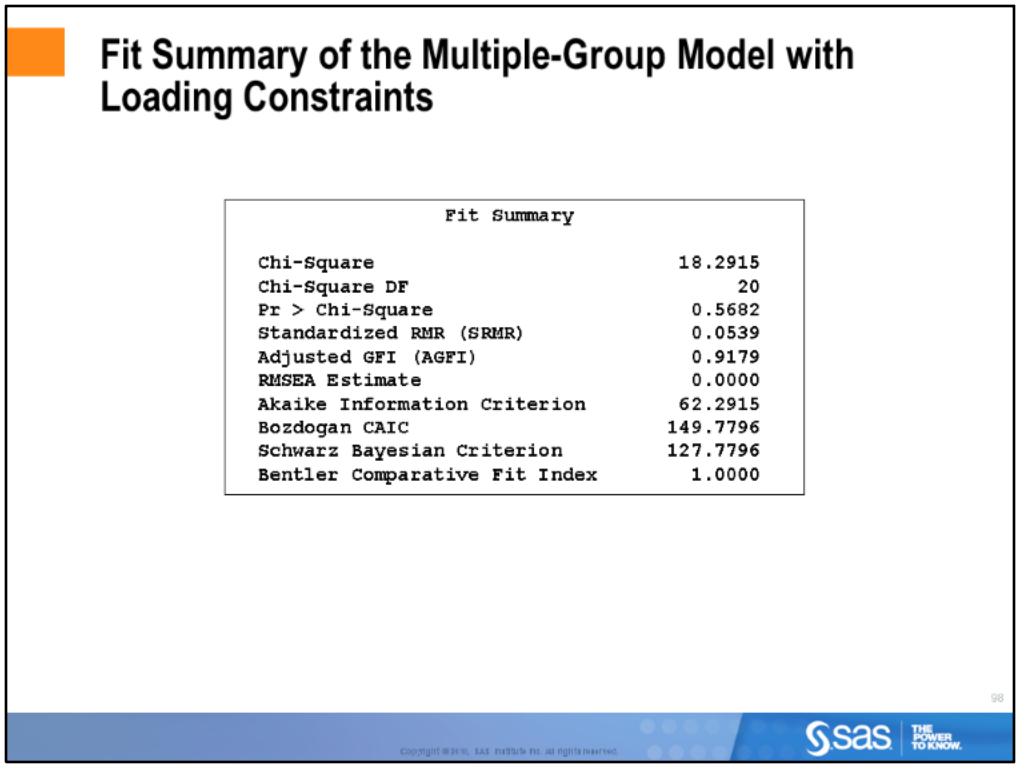
How about releasing the constraints on the structural covariances?

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

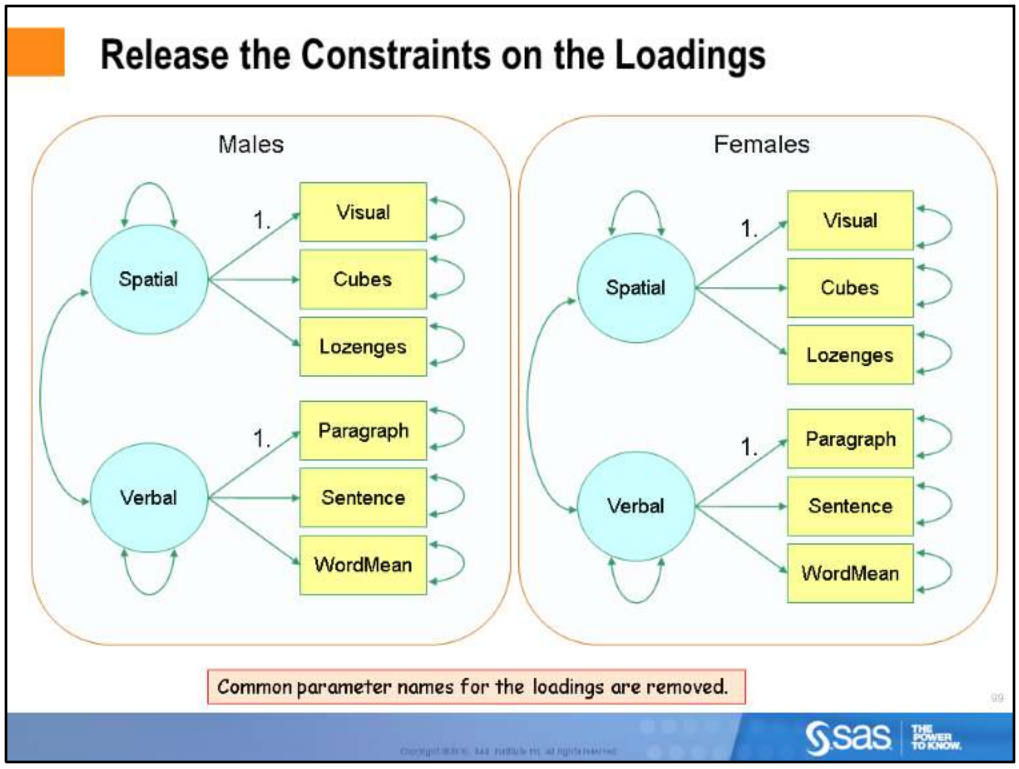


This new multiple-group model can be specifying by commented out the explicit specifications of the structural covariances (variances and covariances among latent variables) in Model 1.

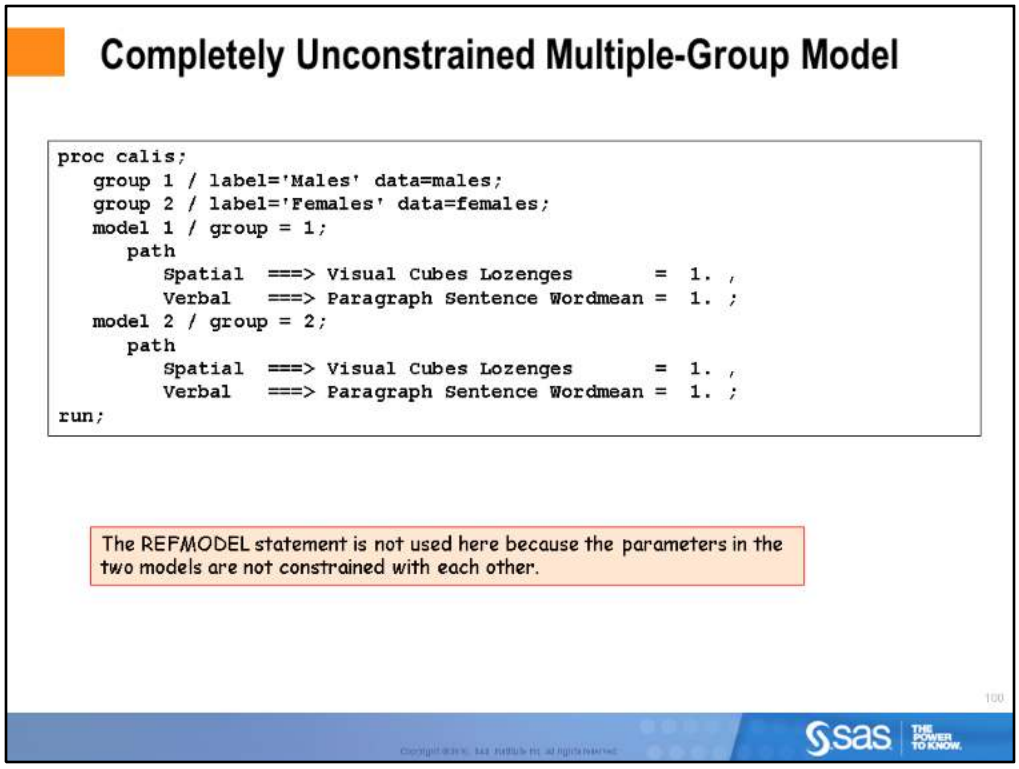
When Model 2 makes reference to Model 1, it copies the explicit specifications in the PATH statement of Model 1. Error variances, structural variances and covariances in the two models are now set by default and are unconstrained between the two models.



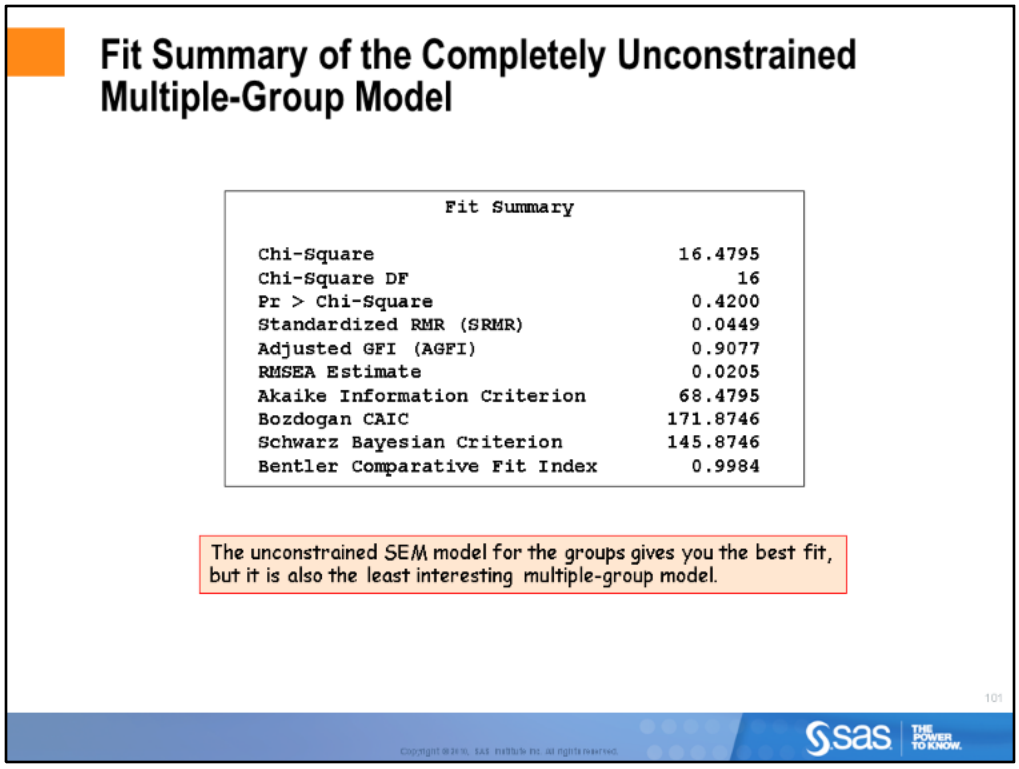
The model fit chi-square is not significant. Now, the SRMR is more acceptable. The AGFI, the RMSEA, and the CFI continue to be very good.



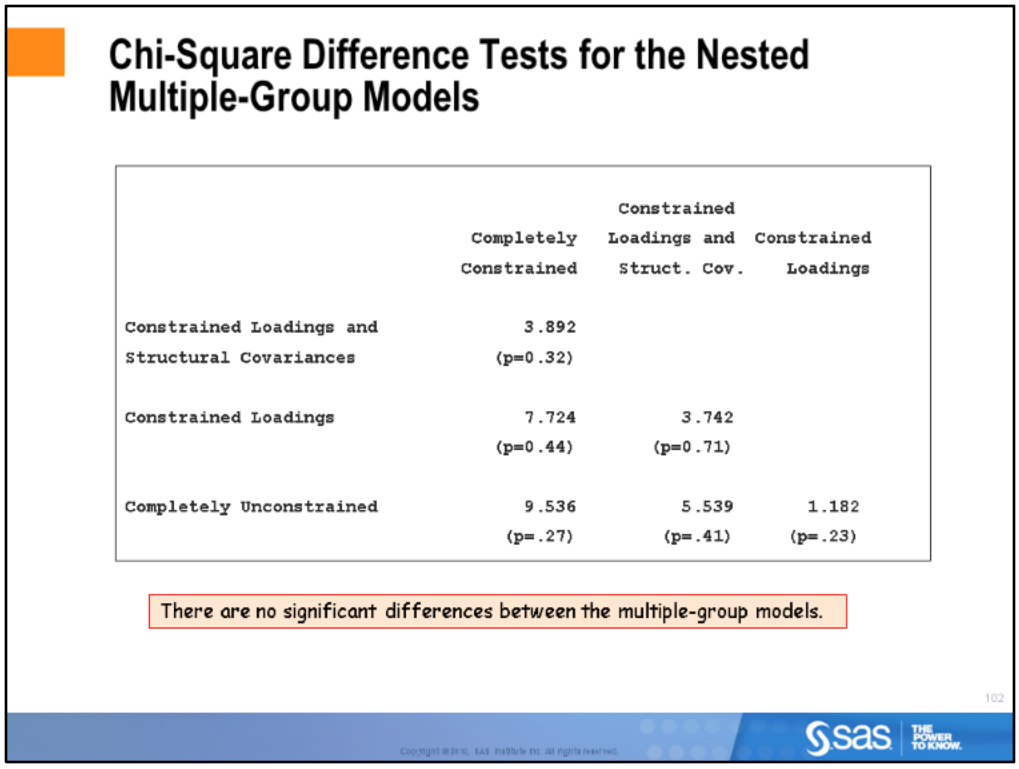
Finally, for the completely unconstrained multiple-group model the path diagrams for the two groups are the same, but no parameter names (except for fixed values of 1) are used to denote constraints.



Because the two models for the groups are totally unrelated, you do 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 (except for the same set of identification constraints with fixed 1).



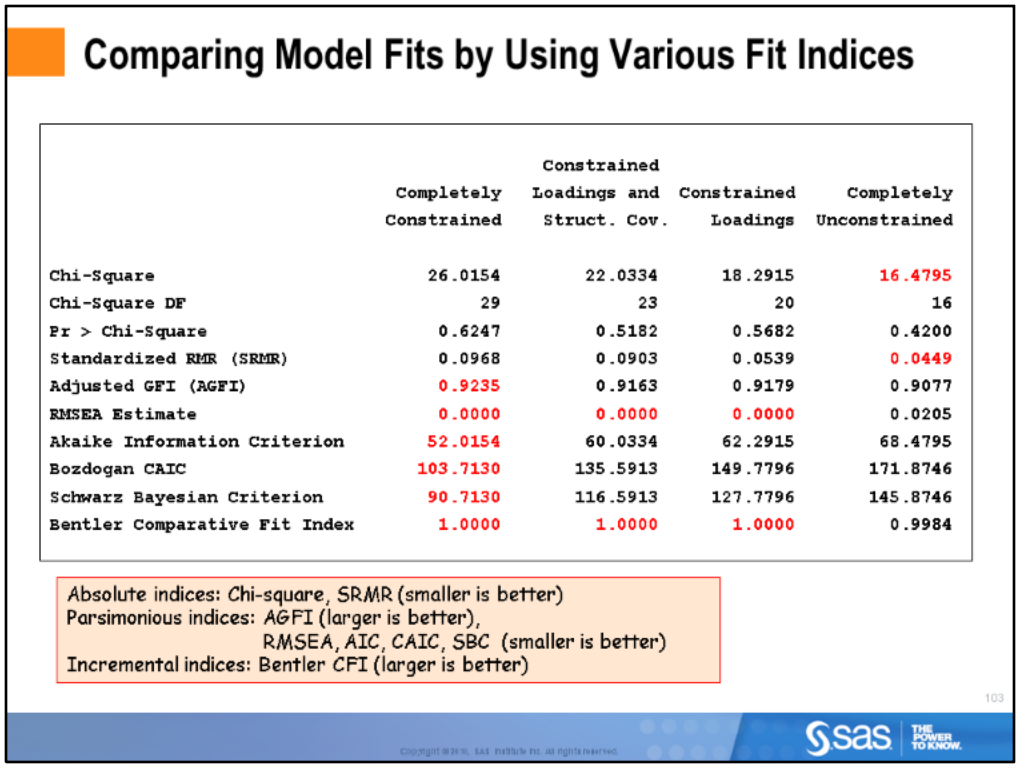
All fit indices indicate very good fit of the unconstrained multiple-group model.



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. (Note: this table is not a part of the SAS output.)

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

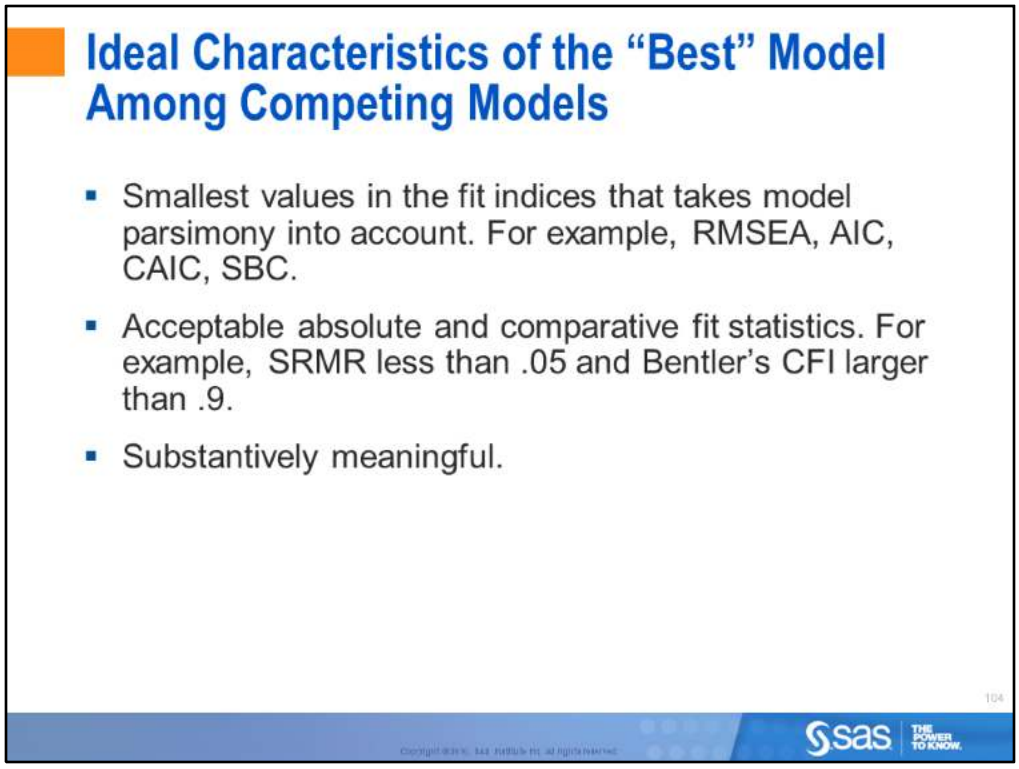


We can also compare the four models by means of the fit index values.

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 SRMR 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 SRMR value because these indices do not take model parsimony into account. Complicated models might have perfect model fit chi-square and SRMR values (that is, 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.

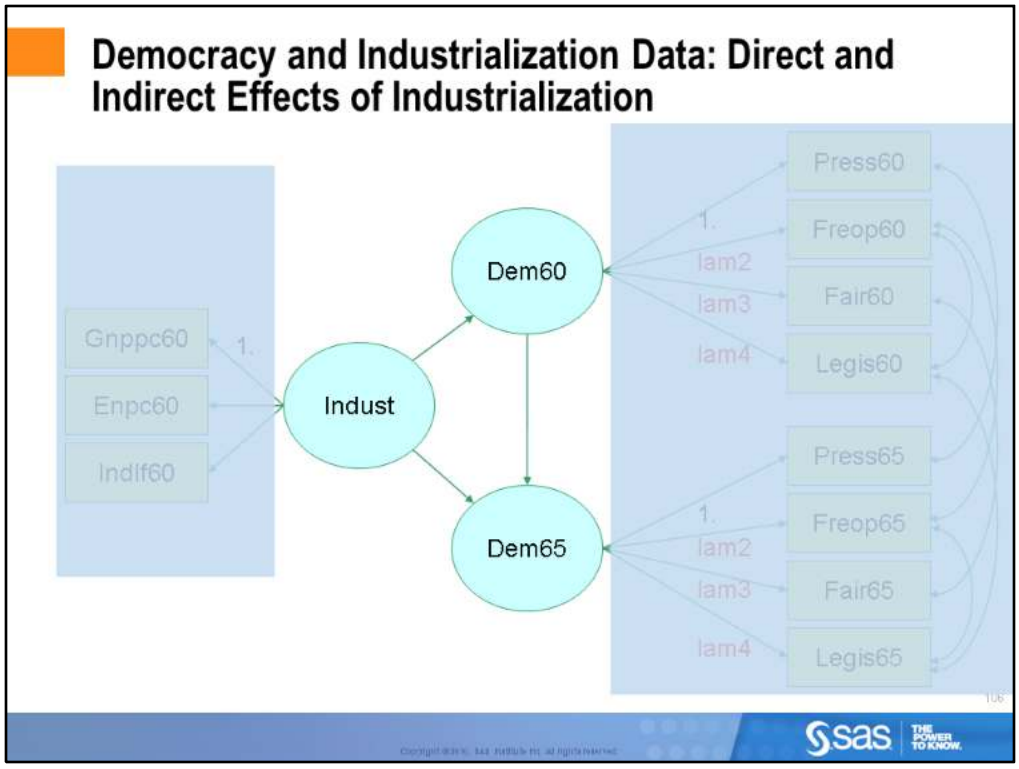
Lastly, the incremental fit index Bentler CFI favors the completely constrained model too. However, virtually all multiple-group model in this comparison are equally good according to the CFI. Notice that incremental indices such as the CFI measures how a target model measures better than a so-called 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.



When you fit a set of competing models for your data, you should select your models based on the fit indices that take model complexity into account. Parsimonious fit indices such as RMSEA, AIC, CAIC, and SBC could be used. These indices might not point to the same "best" model. If they do point to same model pretty consistently, then you might also need to check if the absolute fit indices or other fit indices of the best models are good enough---much like how you judge the fit of an individual model. Simply being the best competing model does not necessarily imply that the model fits the data well. The RMSEA, SRMR, CFI, and etc. of the best competing model must also be acceptable.

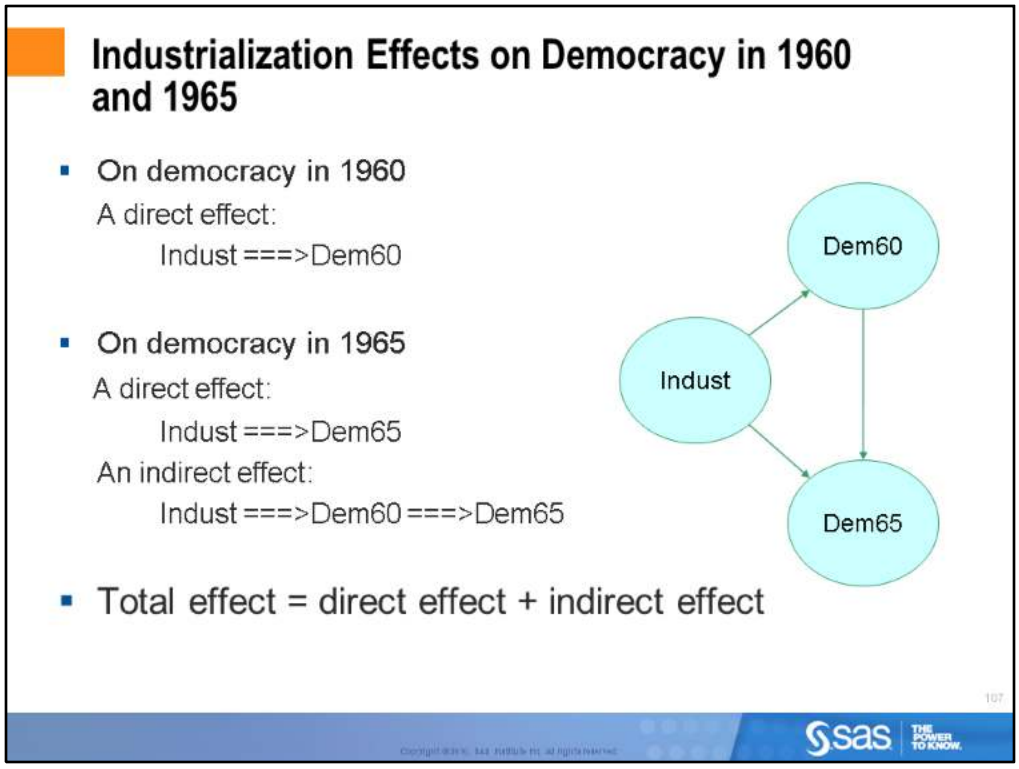
Finally, substantively meaningful models with reasonable fit are preferred to complex models with very good fit that are due to ad-hoc modifications.

Analyzing Direct and Indirect Effects



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.

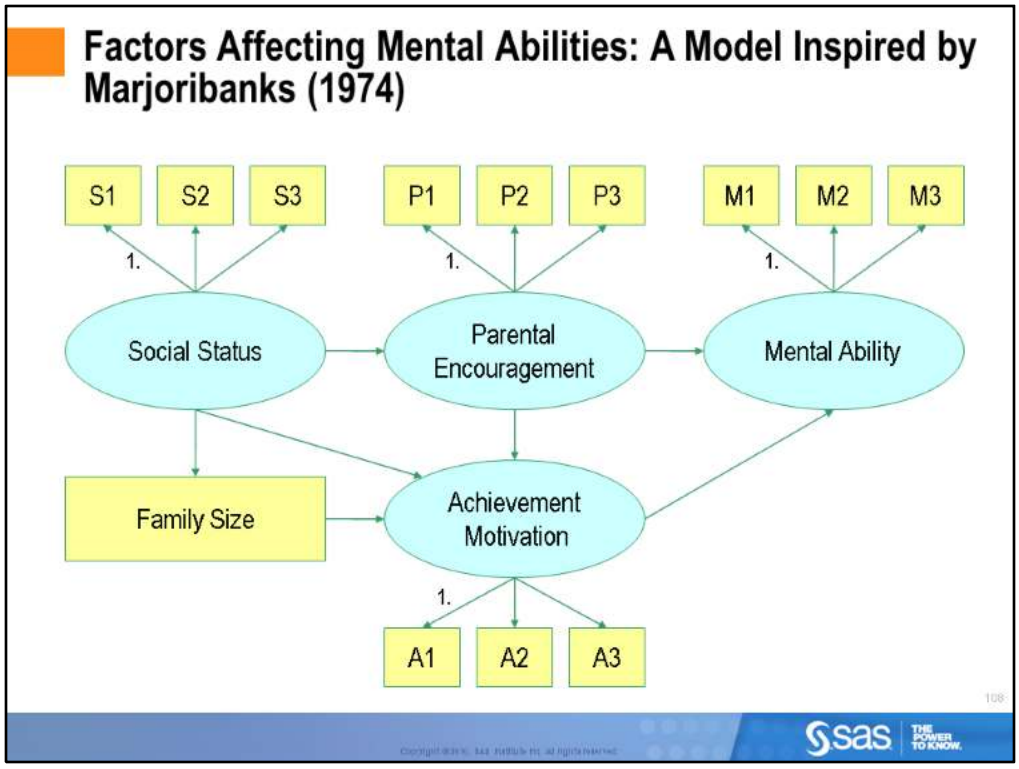


First, let us look at the effect of industrialization on the democracy measure in 1960 (Dem60). The direct effect of Indust on Dem60 refers to the path Indust ===> Dem60. This effect can be estimated directly from any SEM software.

On the democracy measure in 1965 (Dem65), 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 effects, 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.



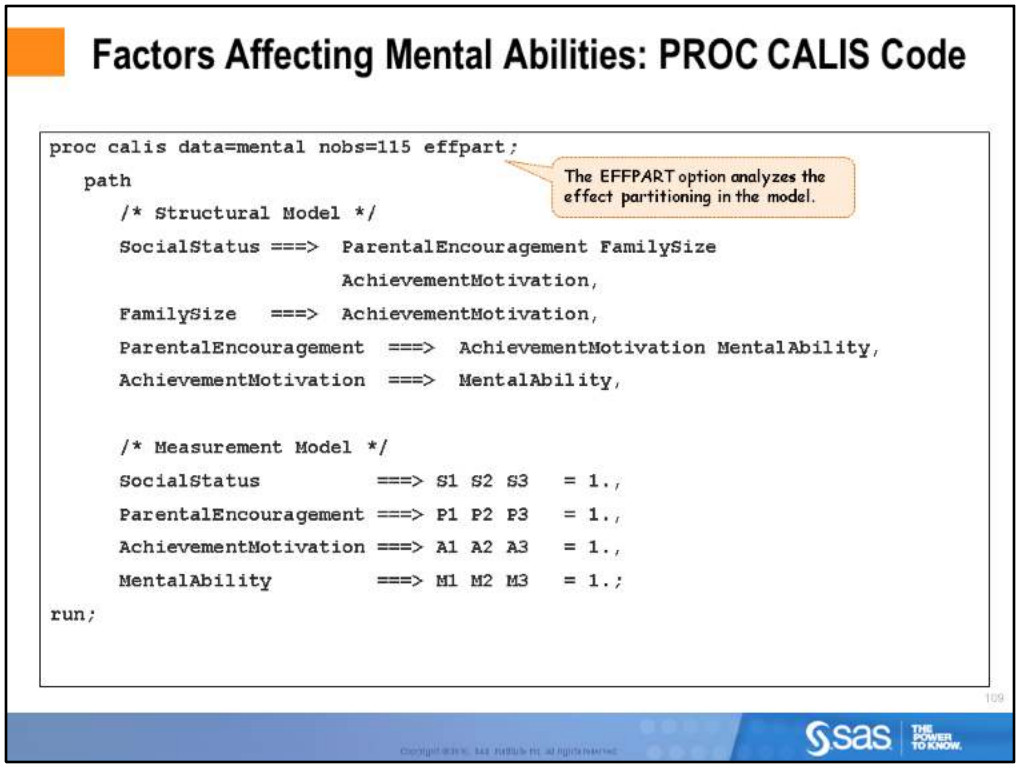
This slide shows a more interesting example about analyzing direct, indirect, and total effects.

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 our purpose.

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.

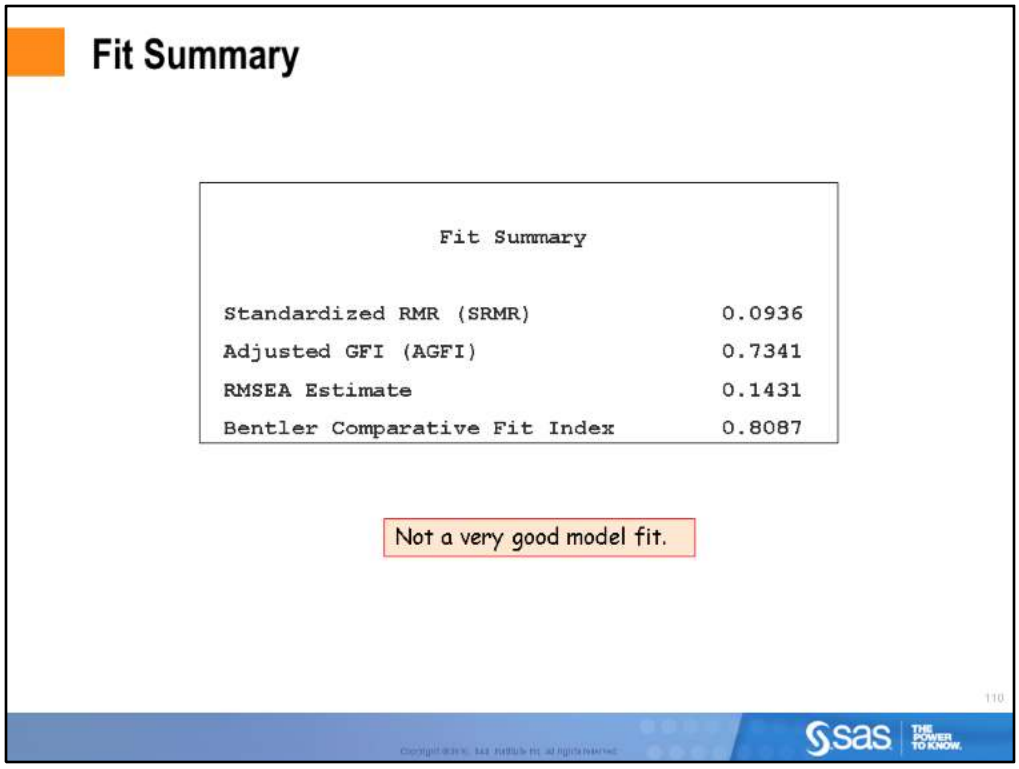
There are some motivating questions about this path diagram regarding the direct and indirect effects. For example,

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 to test 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 to test their significance.

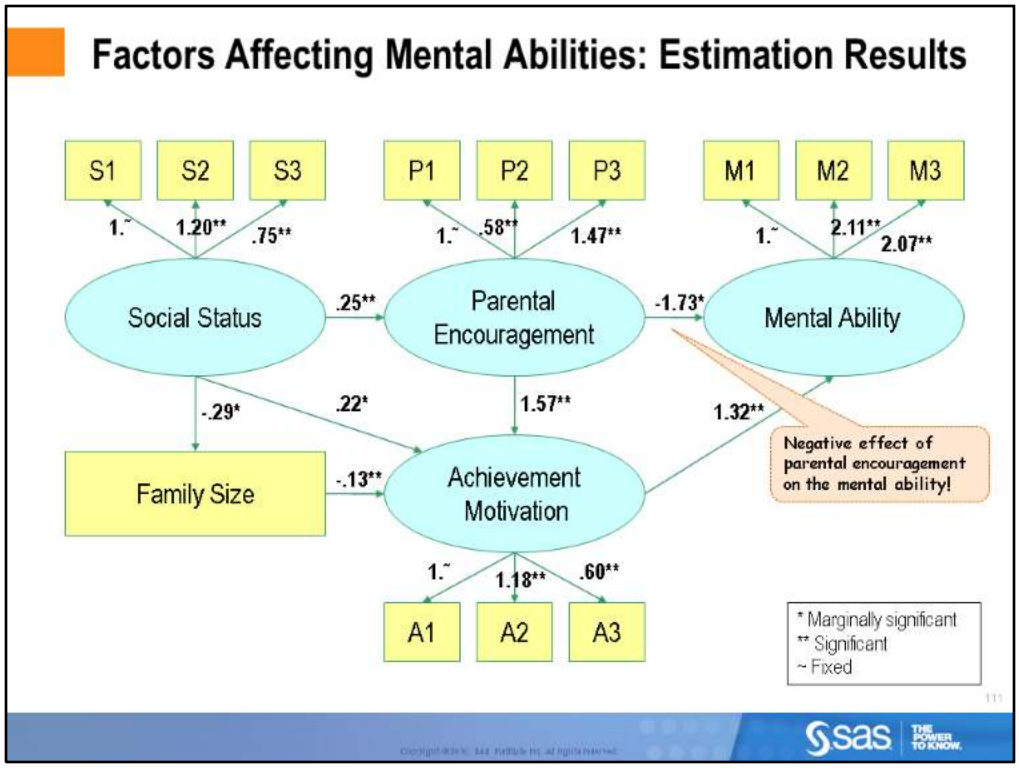


Now, the PATH specification for the target model should be easy for you. You can specify the measurement model and the structural model by the multiple-path syntax. You can look at the path diagram and write down the paths in the PATH statement. Notice that each path in the path diagram represents direct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

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 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 with the current example. In the subsequent discussion, I assume that we are satisfied with the model fit so that the discussion of the effects would be meaningful.

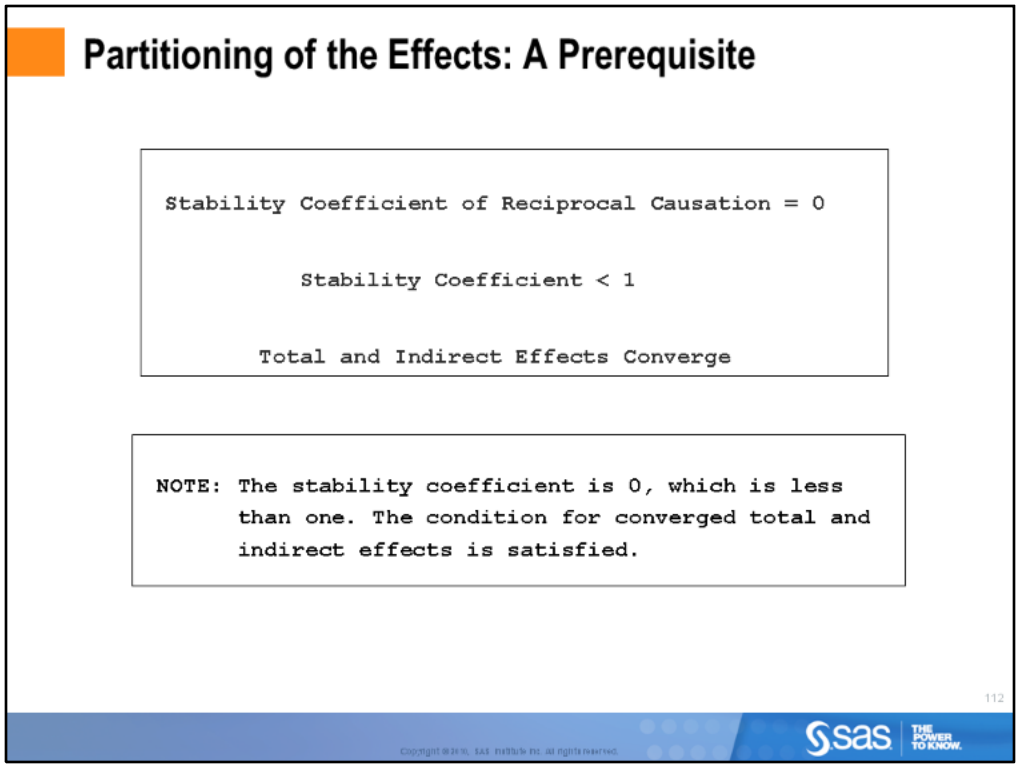


Before diving into the results for effect partitioning, I want to look at the estimates shown in the path diagram. I want to throw in one more motivation to study direct and indirect effects in this structural equation model.

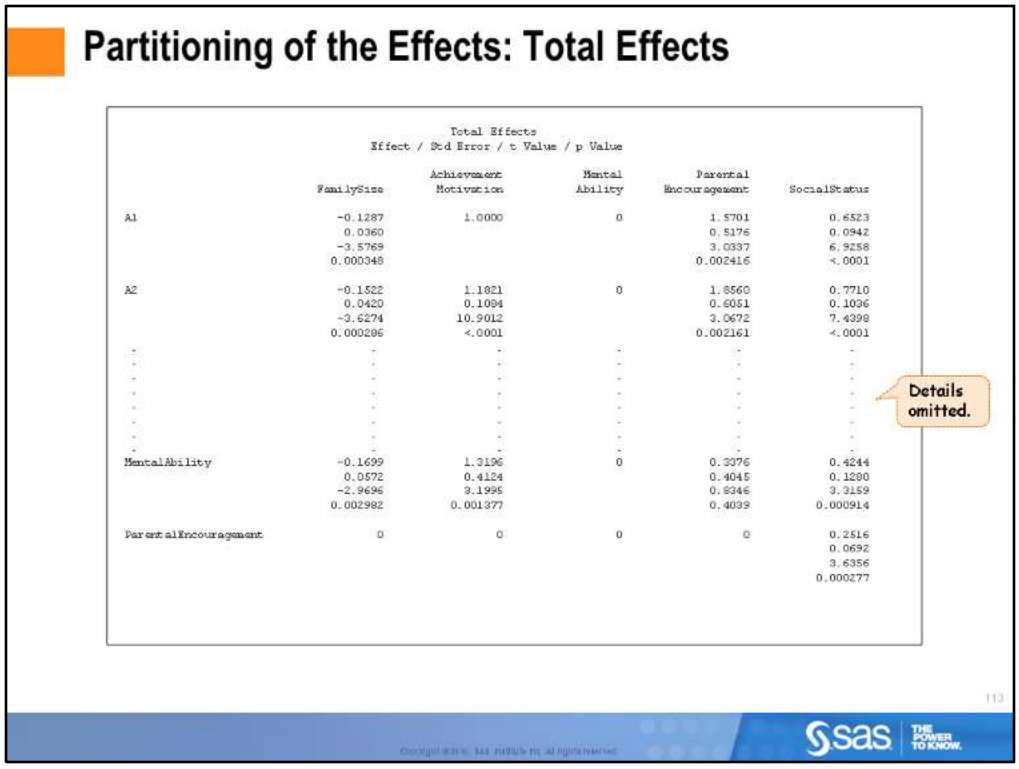
In this path diagram, estimates are shown with their significance marked. Two asterisks after an estimate means the estimate is statistically significant. One asterisk after an estimate means that the estimate is marginally significant.

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 direct 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?

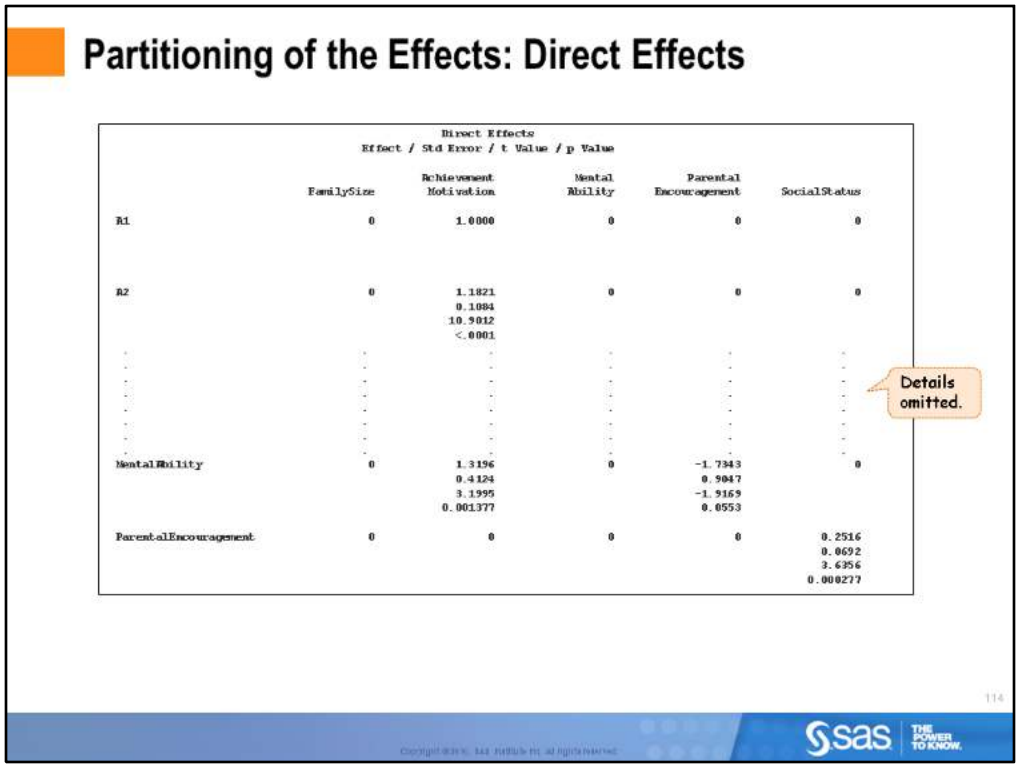


Before you can analyze direct and indirect effects, you should check whether a prerequisite is satisfied. In order to study the effect partitioning legitimately, the so-called stability coefficient must be less then 1. PROC CALIS provides such a check. The checking of this stability coefficient is important. When you see the messages in this slide from the PROC CALIS output, you could proceed to examine your effect partitioning results. Otherwise, if the stability coefficient is not less than 1, you cannot interpret the indirect and total effects.

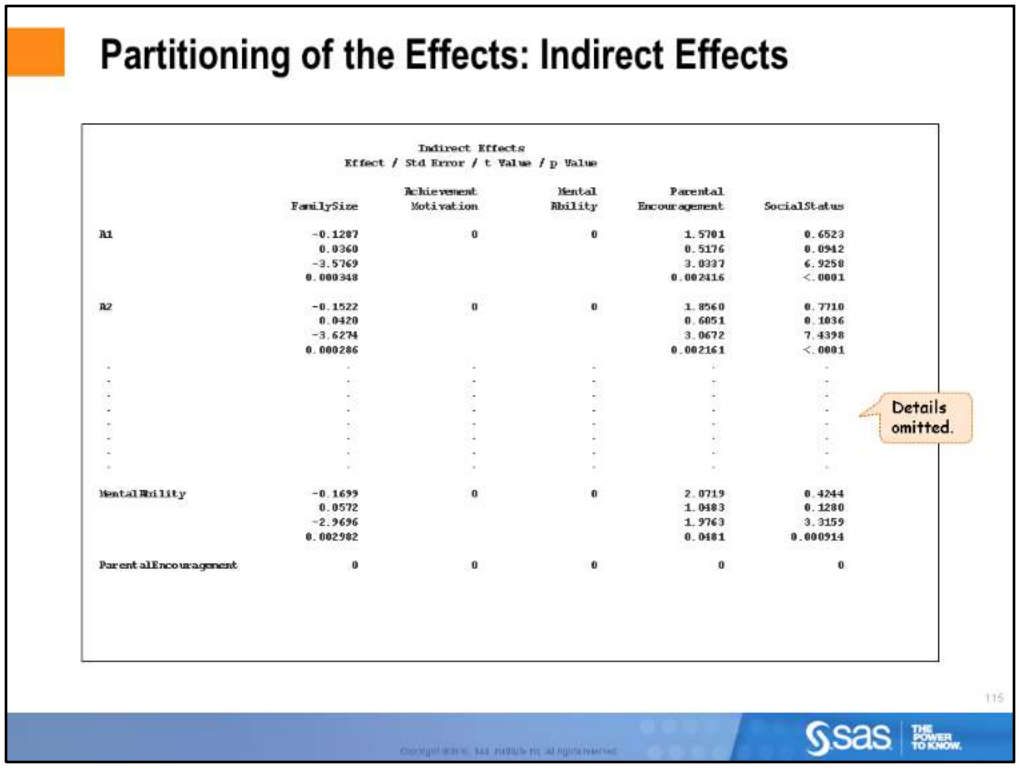


With the EFFPART option, PROC CALIS produces tables for total, direct, and indirect effects separately. These tables could be large. I just annotate these results here. Some results are not shown.

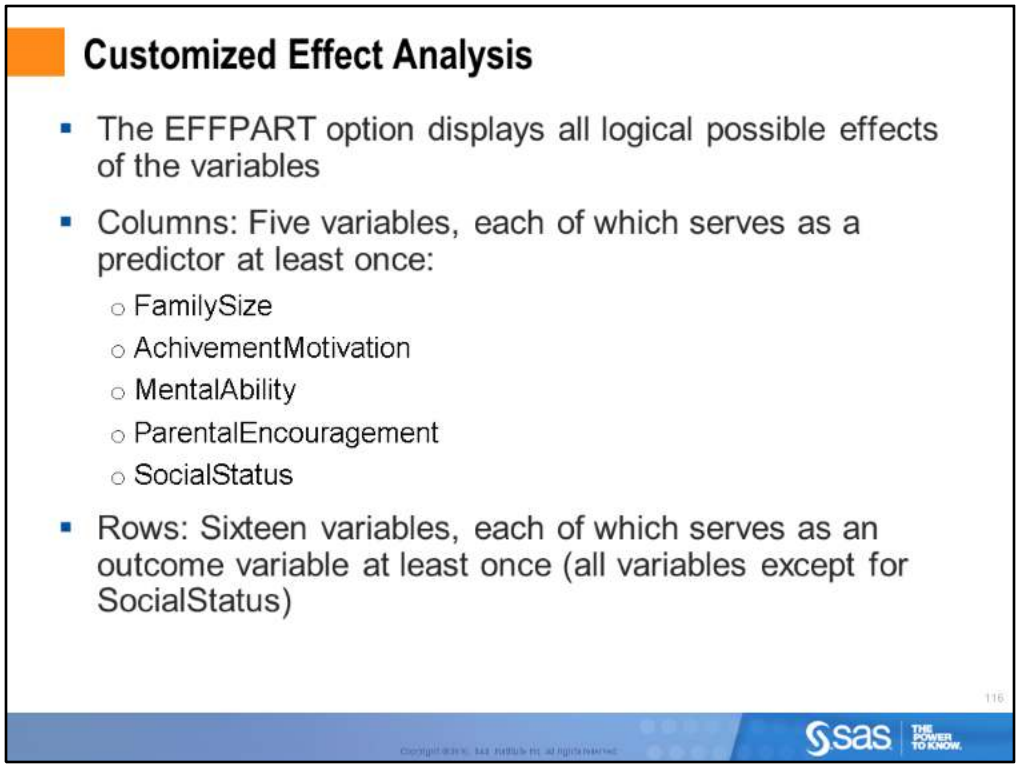
This table is about the estimates of the total effects, their standard errors, t-values, and significance levels.



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

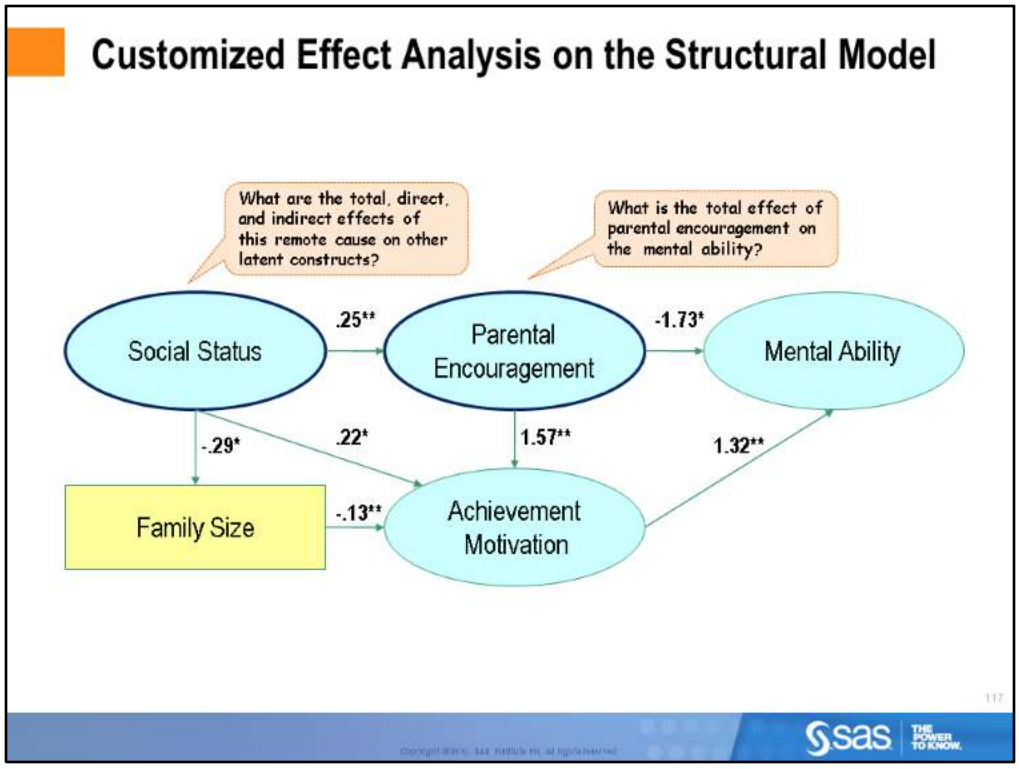


This table is about the indirect effects , their standard errors, t-values, and significance levels.



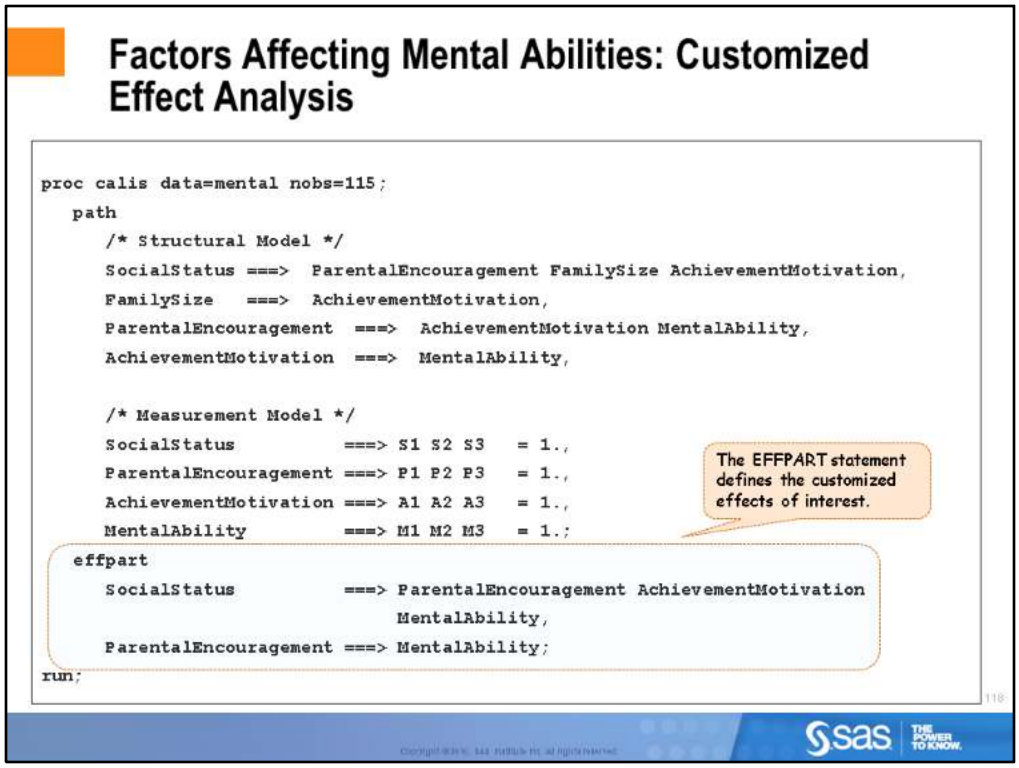
When you have large tables like those shown in previous slides, you are likely to be doing exploratory analysis without specific questions in your mind. The effect tables could get very large and you might have a difficult time to look for the particular results that you are interested in. For example, 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 the customized effect analysis, which 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 mental ability?
2. What is the total effect of parental encouragement on the mental ability?



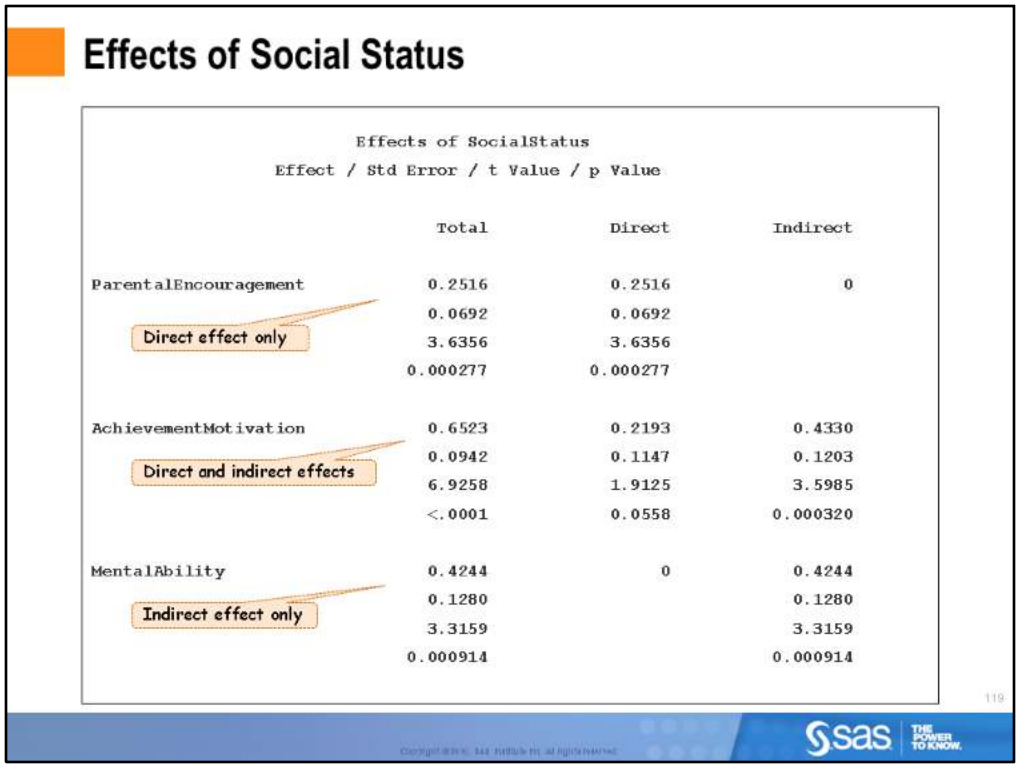
PROC CALIS supports the customized effect analysis. This can be done by the EFFPART statement, as shown in the PROC CALIS code in this slide.

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;
```



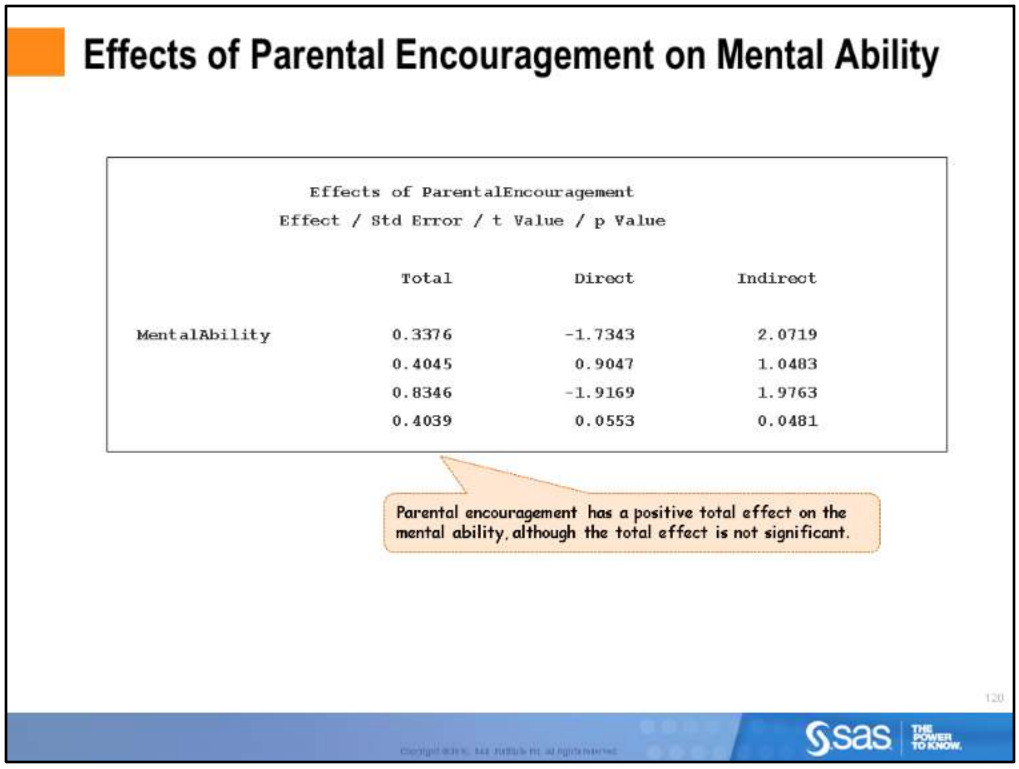
The effect partitioning results from PROC CALIS are shown in this slide and the next one.

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.



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.

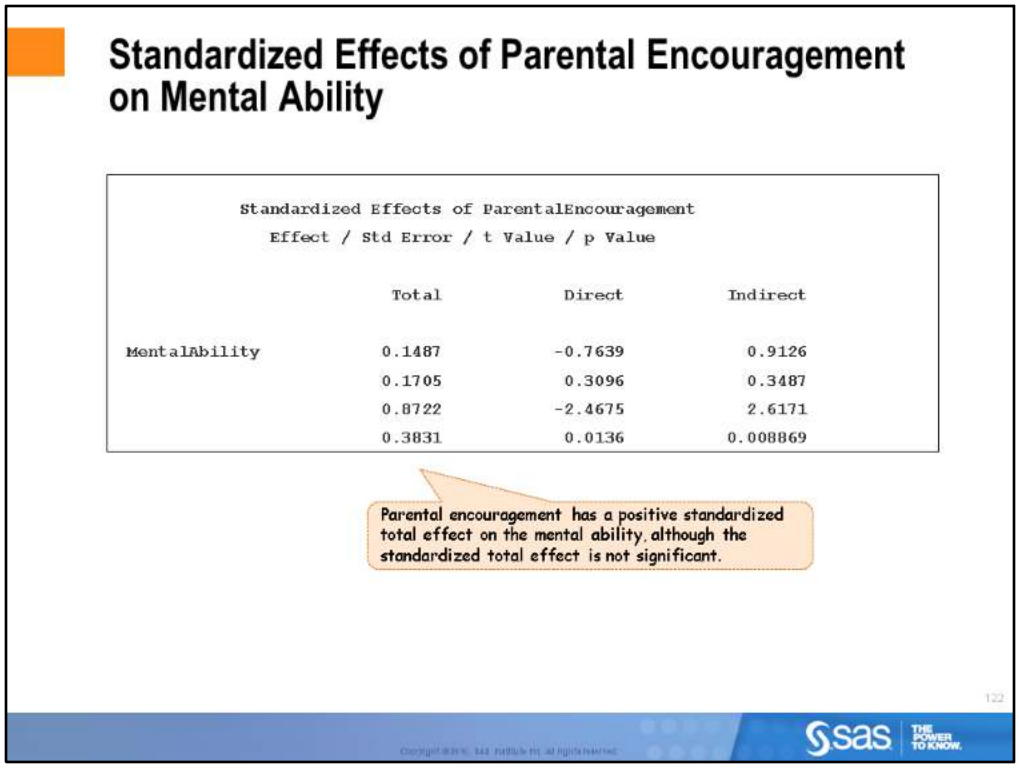
This example shows that SEM effect analysis can show some effect patterns that simply cannot be analyzed by linear regression analysis adequately. The SEM effect analysis provides something more detailed and refined regarding the totality of the theory. In this regard, the customized effect analysis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

Standardized Effects of Social Status

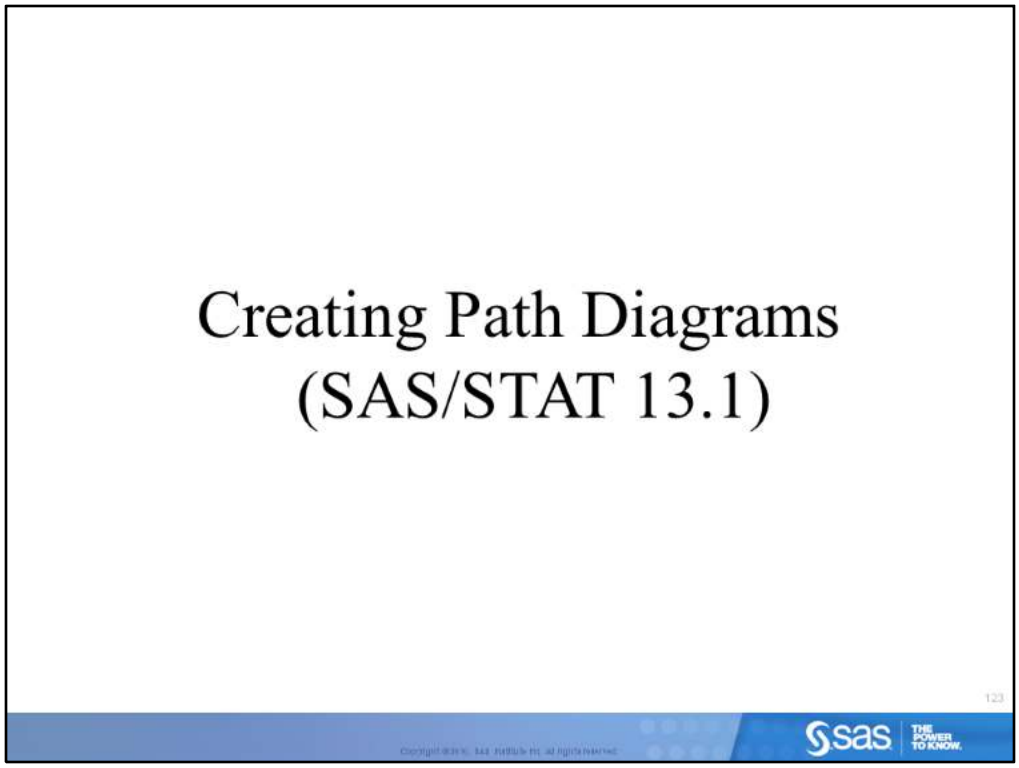
Standardized Effects of SocialStatus
Effect / Std Error / t Value / p Value

	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

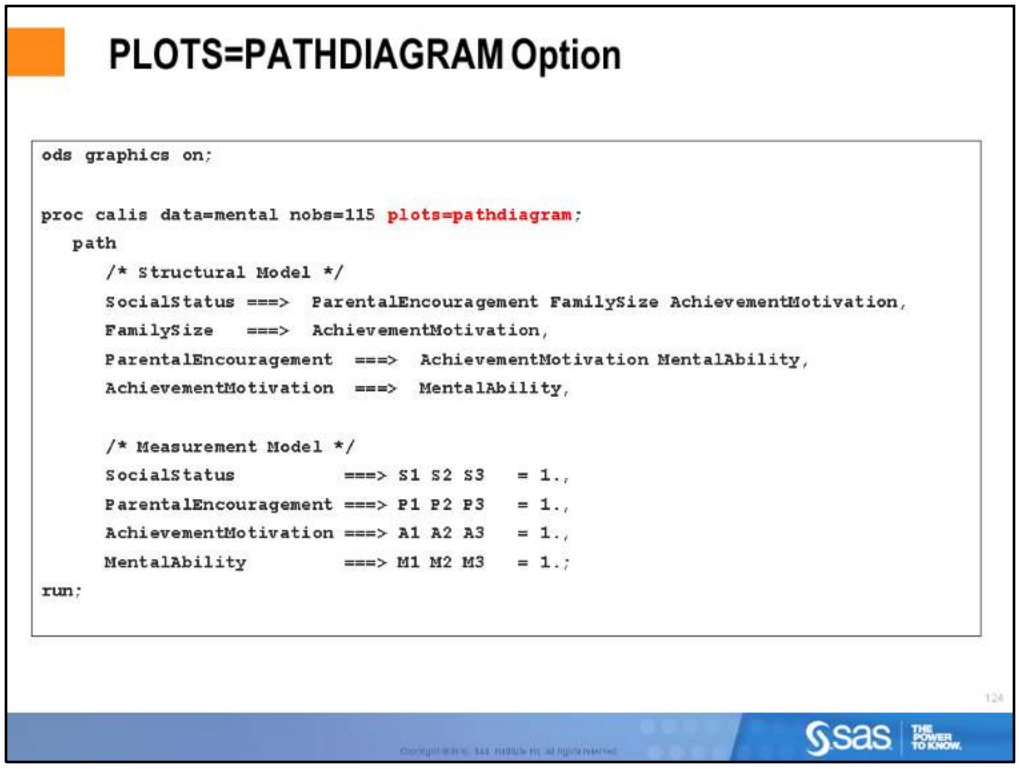
PROC CALIS also provides the standardized results for effect analysis. Standard errors, t-values, and p-values are also computed for the standardized effect estimates.



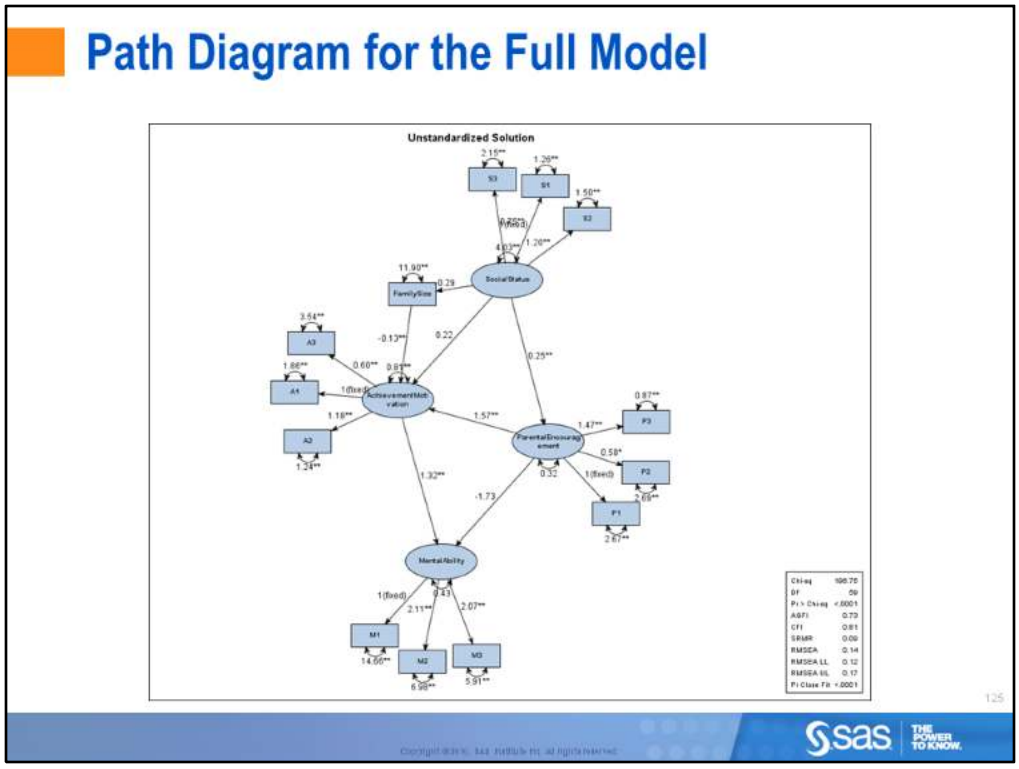
This slide shows the standardized effects of parental encouragement on mental ability. The pattern is quite similar to the unstandardized version.



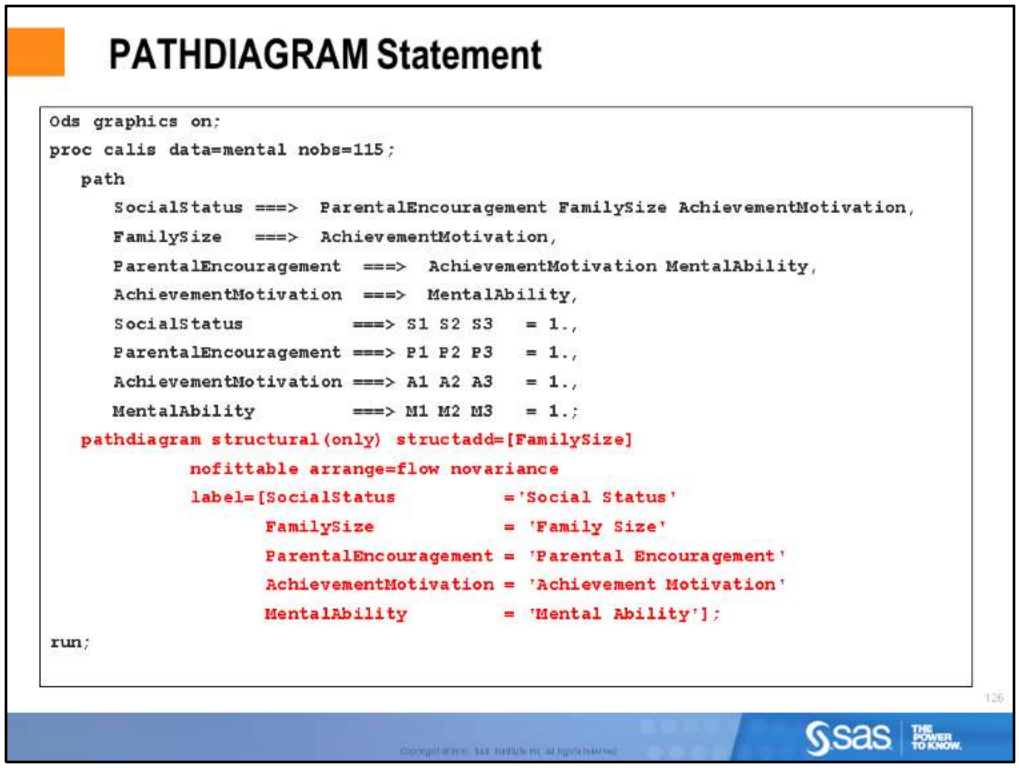
Creating path diagrams from PROC CALIS is a new capability in SAS/STAT 13.1.



PROC CALIS can create path diagrams automatically from model input. This example uses the same data set about achievement motivation. The simplest way to create path diagrams from PROC CALIS is to use the PLOTS=PATHDIAGRAM option in the PROC CALIS statement.



This is the default path diagram for the unstandardized solution. Estimates that are flagged with "**" are significant at the 0.01 alpha level.
Estimates that are flagged with "*" are significant at the 0.05 alpha level. A fit summary table is also shown.

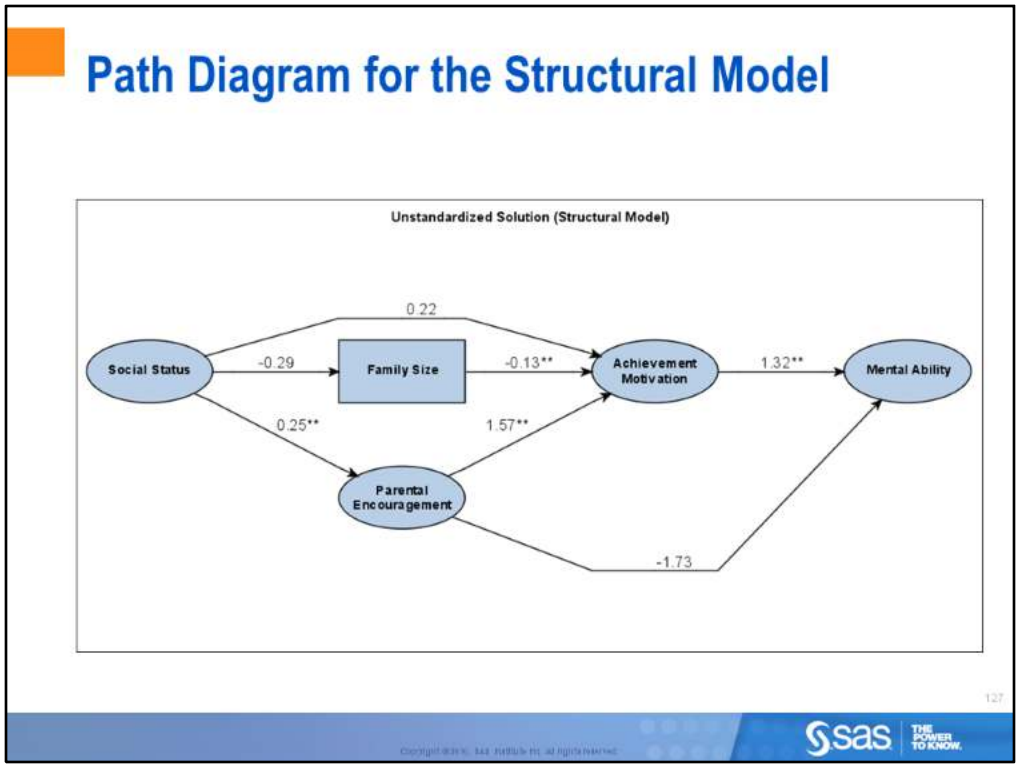


Showing the full model might not be the best way to present the main ideas of your fitted model. In structural equation modeling, researchers sometimes focus on the structural component of the model only. The reason is that most of the causal interpretations apply to the structural relationships only. You can customized path diagrams that show only the structural components of the models. The customization is done by using options in the PATHDIAGRAM statement. For example, the STRUCTURAL(ONLY) option requests to creation of the path diagram for the structural component only.

Traditionally, the structural component refers to the part of the model that include only the latent factors and their corresponding functional relationships. This seems to be a good definition for the LISREL-type models only. In general, variables that are supposed by be measured without measured errors could be included in the structural component. In this example, FamilySize is an exogenous variable in the model and it is measured without errors. To include this variable in the path diagram the STRUCTADD= option is used. Otherwise, the FamilySize variable will not be included in the path diagram.

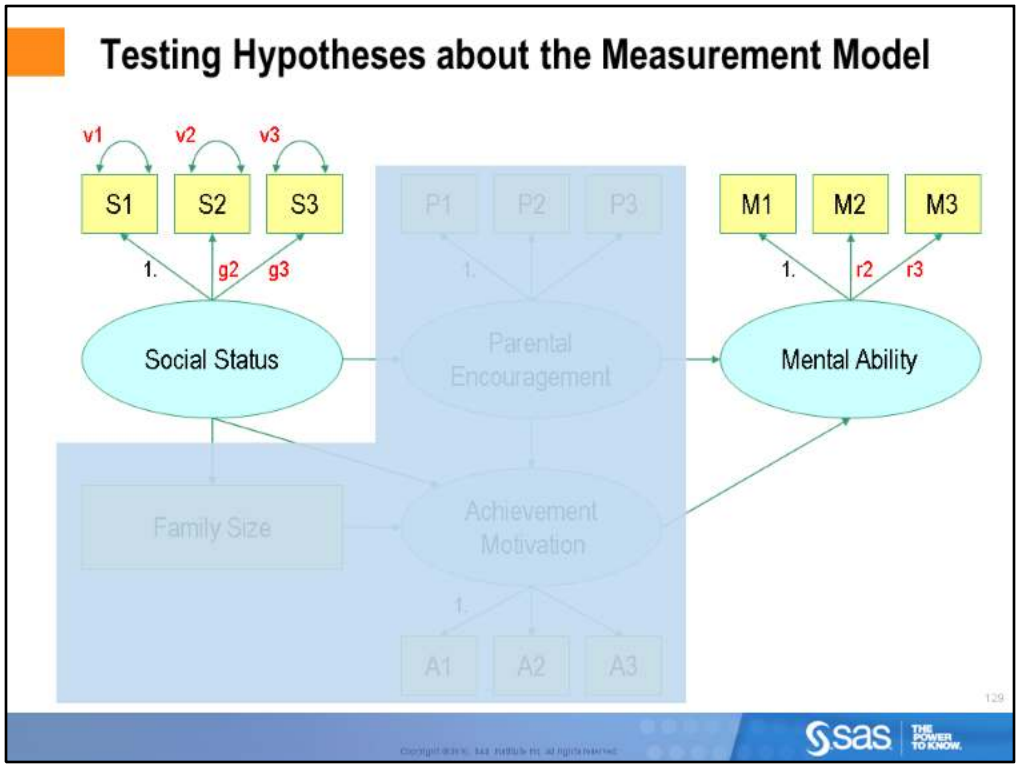
Other options in the PATHDIAGRAM statement include:

1. The NOFITATBLE option suppresses the display of the fit summary table.
2. The ARRANGE=FLOW option requests the use of the FLOW layout algorithm so that the causal ordering of the effects is emphasized. If you do not use this option, the layout algorithm is automatically determined.
3. The NOVARIANCE option suppresses the display of all variance estimates so the path diagram will have a cleaner look.
4. The LABEL= option specifies the labels be used in the path diagram. In this example, the labels used are actually similar to the original names---only that appropriate spaces are added in the labels. This would help the layout algorithm find proper breaks of character strings when encountering long texts.



The structural model shows a much cleaner picture for presentation.

Testing Specific Hypotheses

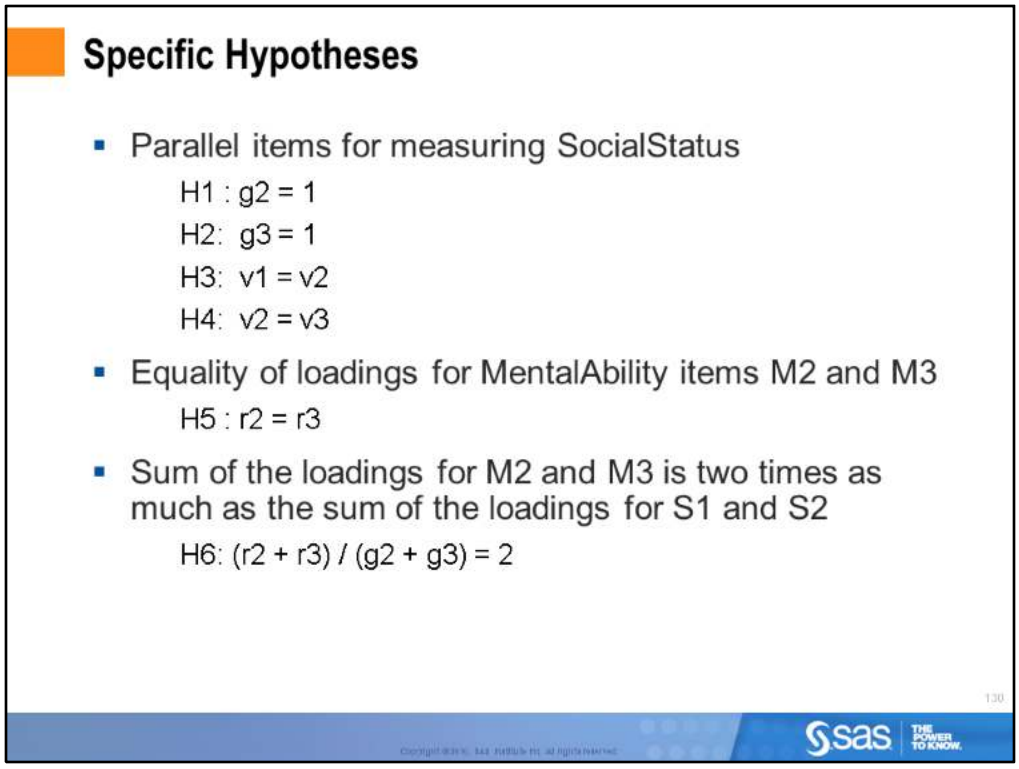


Testing specific hypotheses is an interesting topic. Here we look at some examples.

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.

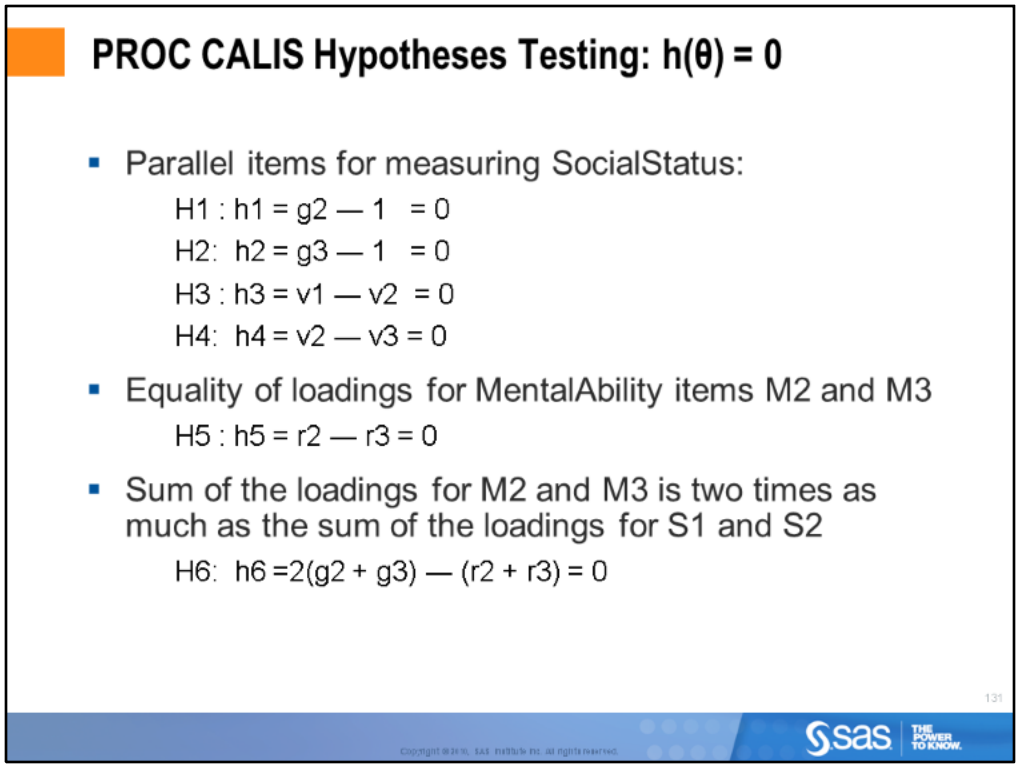
For the social status factor, you not only want to test the hypothesis that the loadings (path coefficients) are the same for the three indicators, but you also want to see if their corresponding error variances are the same in the population. In the path diagram, g2, g3, v1, v2, and v3 are parameters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

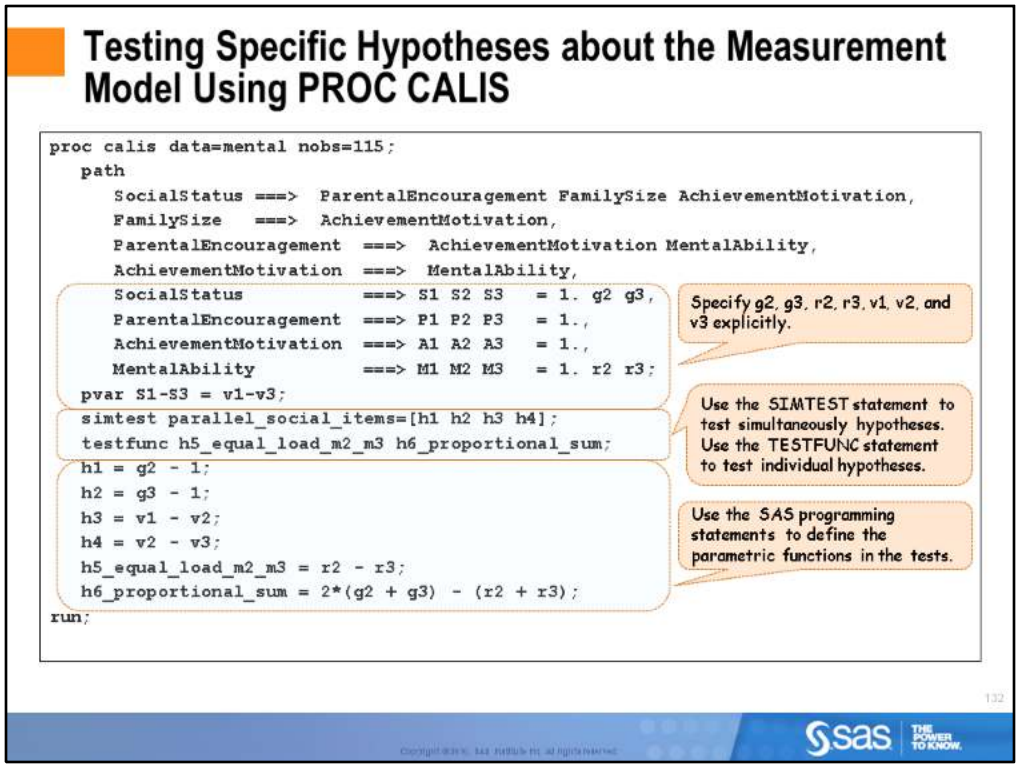
Finally, you can invent any strange hypothesis that can be expressed as a continuous function of the model parameters. For example, H6 states that the ratio of the sum of r2 and r3 to the sum of g2 and g3 is 2. This hypothesis may or may not make sense. But it is included here to demonstrate the flexibility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



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.



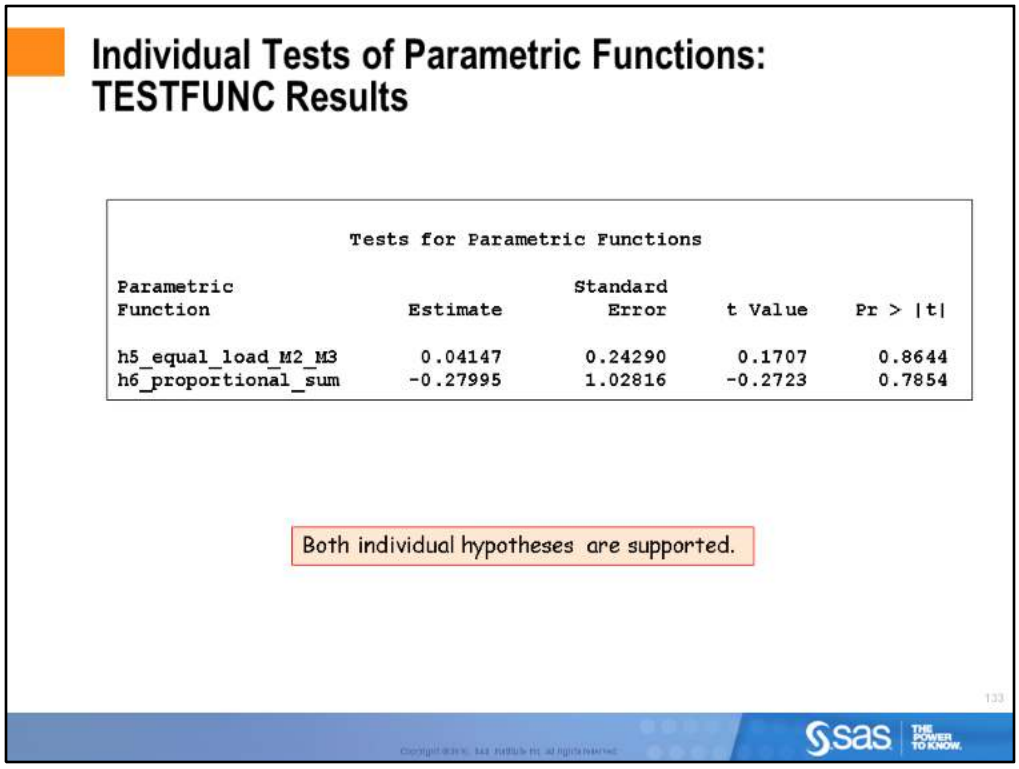
First, you have to label or name the parameters in the correct locations of the model specification. For example, g2 and g3 are the path coefficients for S2 and S3, respectively; and r2 and r3 are the path coefficients for M2 and M3, respectively. Notice that you did not name these parameters in the preceding model specifications. Naming these parameters were optional because you did not need to make reference to them. However, because you are going to refer to these parameters in the hypothesis tests, you must name or label them in the respective locations in this example. 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.

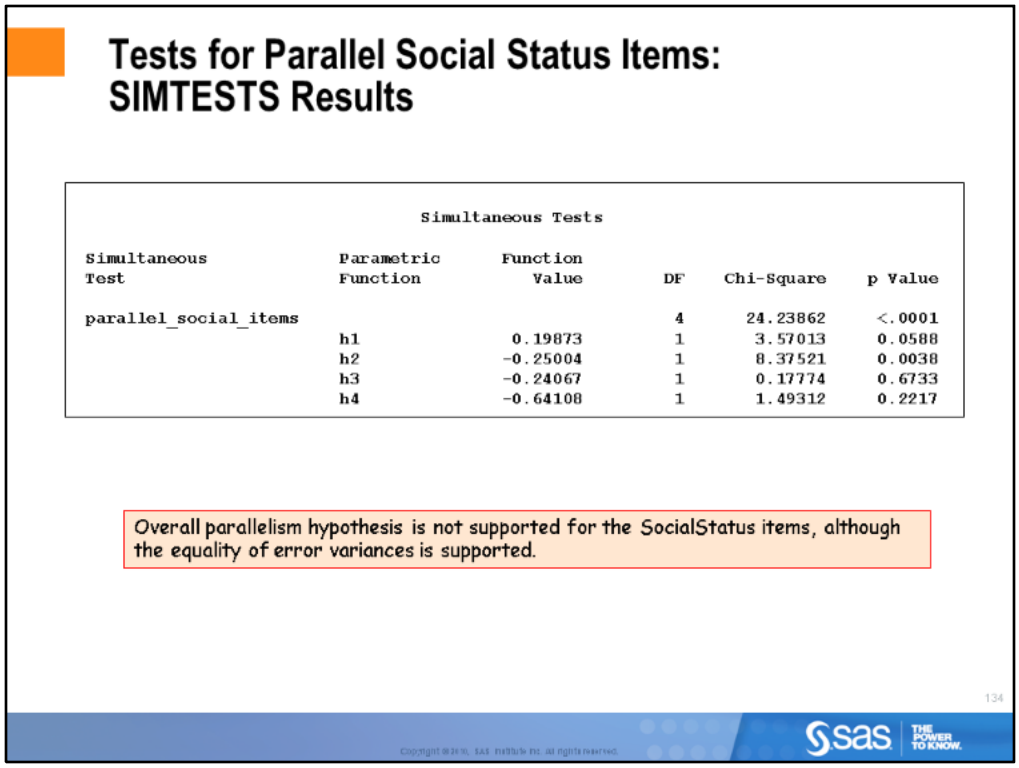
Now I use the so-called SAS programming statements to define the hypotheses: h1-h4, h5_equal_load_m2_m3, and h6_proportional_sum. The SAS programming statements are just like common mathematical equations. These six SAS programming statements define the parametric functions in the target hypotheses.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



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).

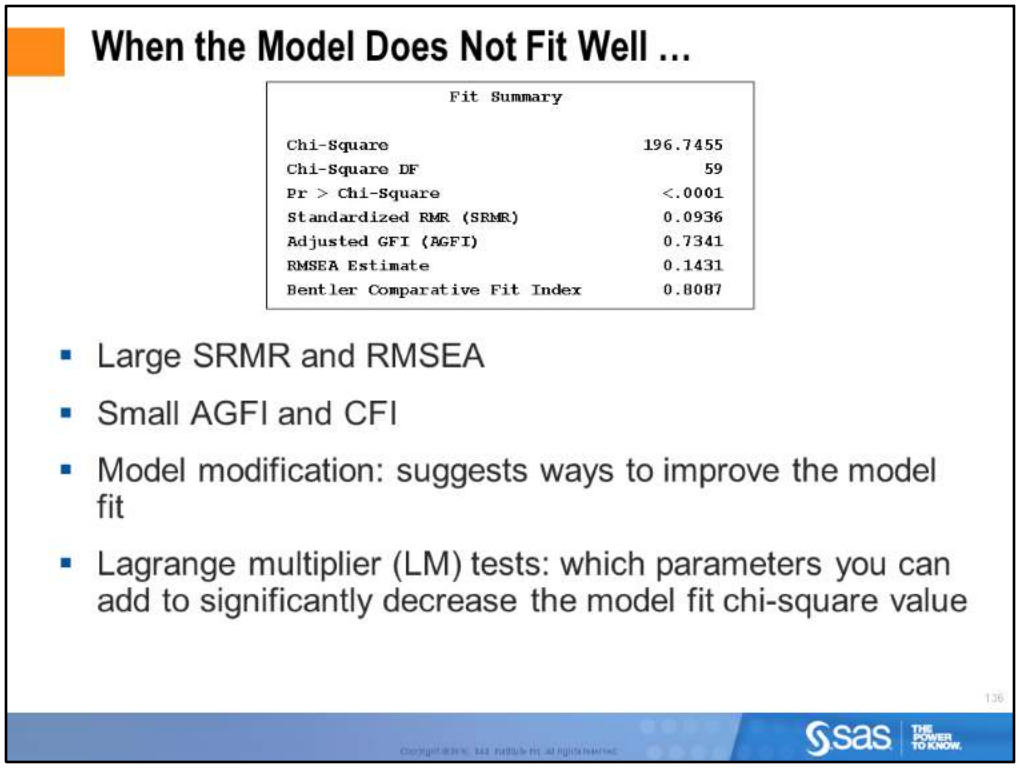


The SIMTESTS statement specification produces the output shown in this table.

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

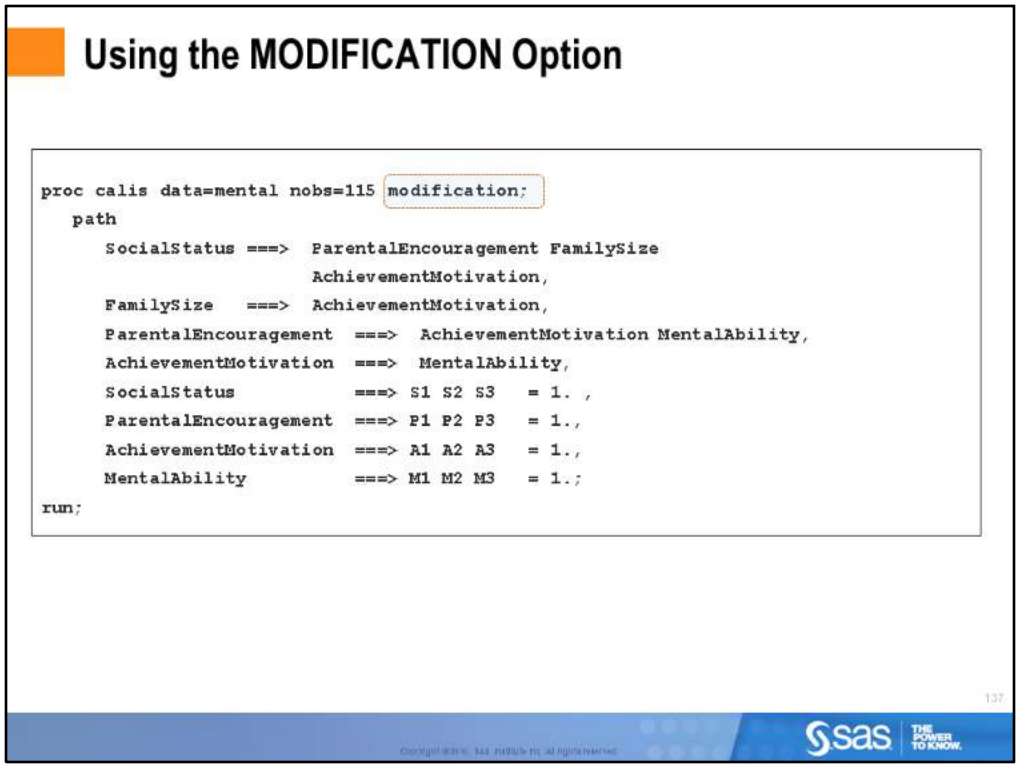
PROC CALIS also provides individual tests for the component hypotheses. This would be useful for doing an ad-hoc analysis to probe what fails the simultaneous hypothesis. For example, both h1 and h2 are at least marginally significant. But h3 and h4 are not significant. Recall that h1 and h2 are about 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

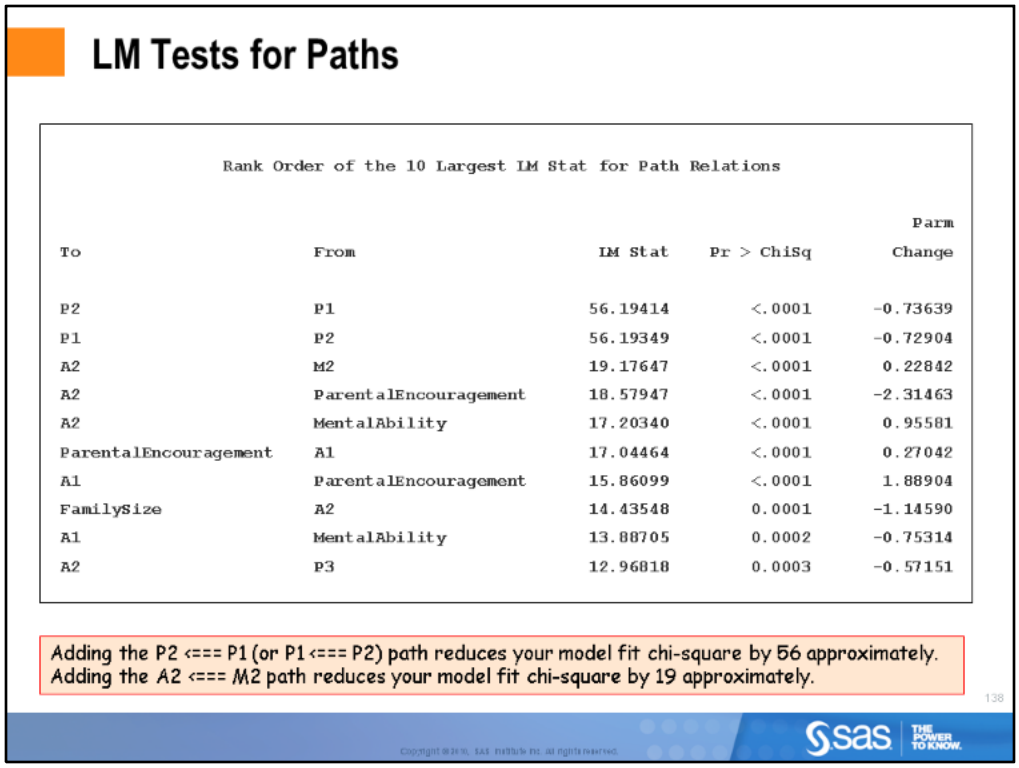


The mental ability model did not fit well. The SRMR and the RMSEA are large, while the AGFI and the CFI are small. When you encounter a bad model fit, it would jeopardize your interpretations of the model parameters, effect analysis, hypothesis testing, and etc.

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.



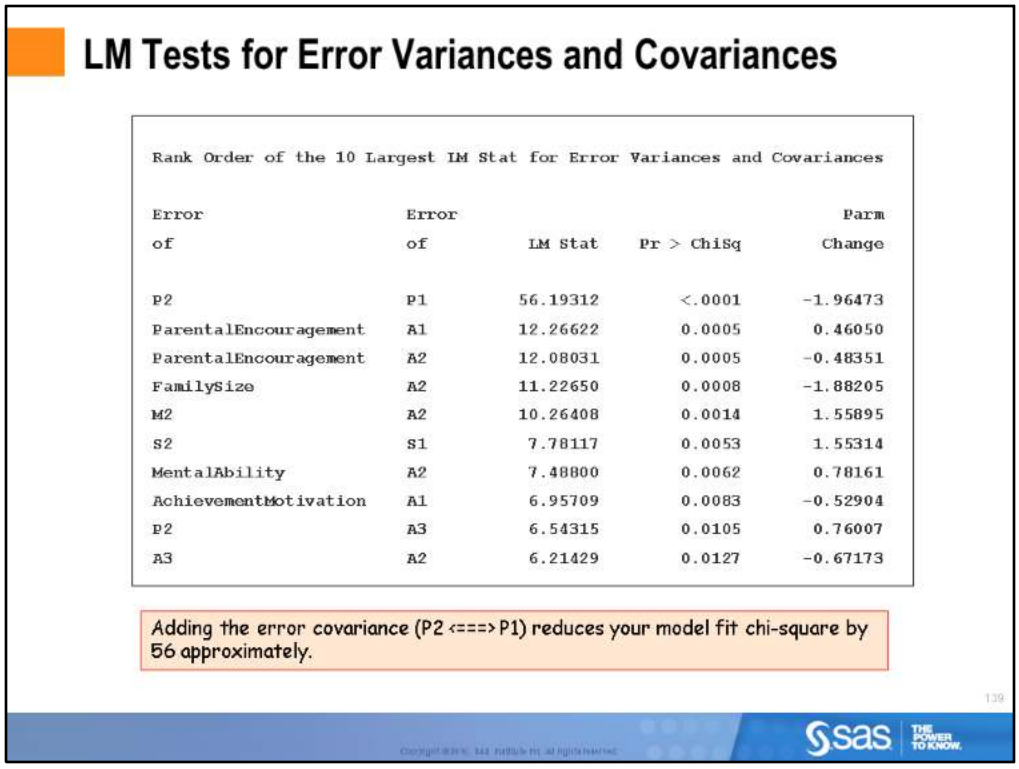
The option you can use to do model modification in PROC CALIS is the MODIFICATION option in the PROC CALIS statement. You can simply add this option to the PROC CALIS statement when you run your model. This example shows that the LM tests for model modification is requested for the original mental ability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



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.

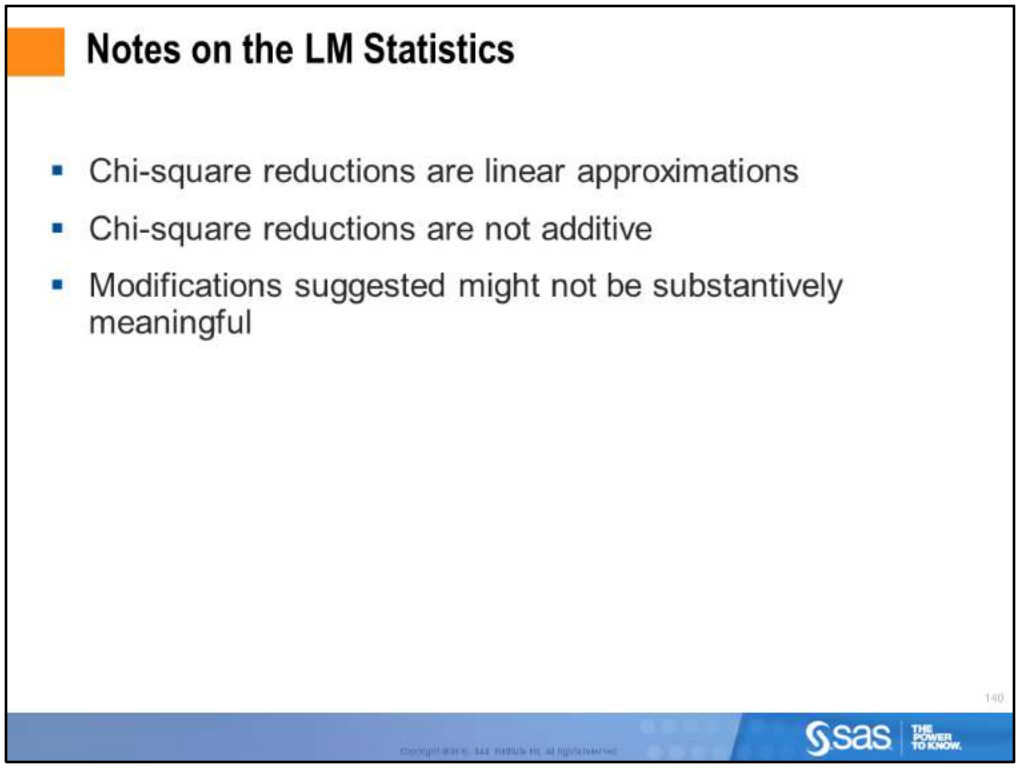
The top one is the p1 ===> p2 path. The LM statistic 56.19 means that if you include this path into the model, you can expect to reduce the model fit chi-square by about 56. This is a substantial improvement because you can get this big improvement by just losing one degree of freedom. The second one 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.



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?



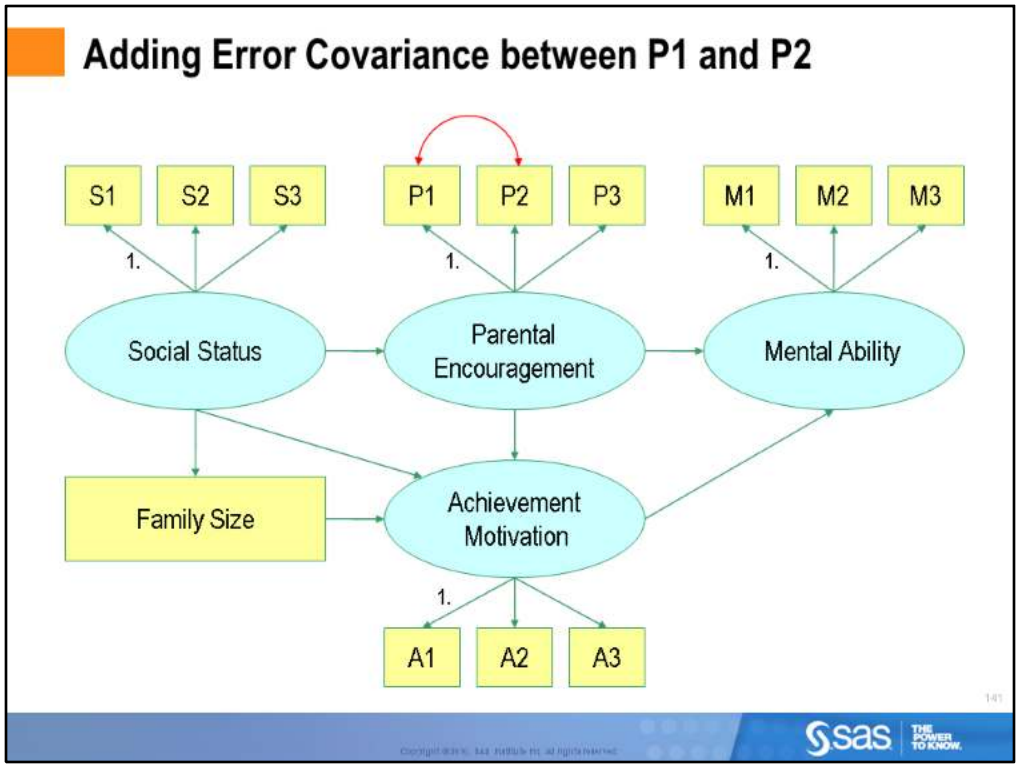
Before giving an answer to the current model modification analysis, some important general points about the LM statistics are discussed.

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.

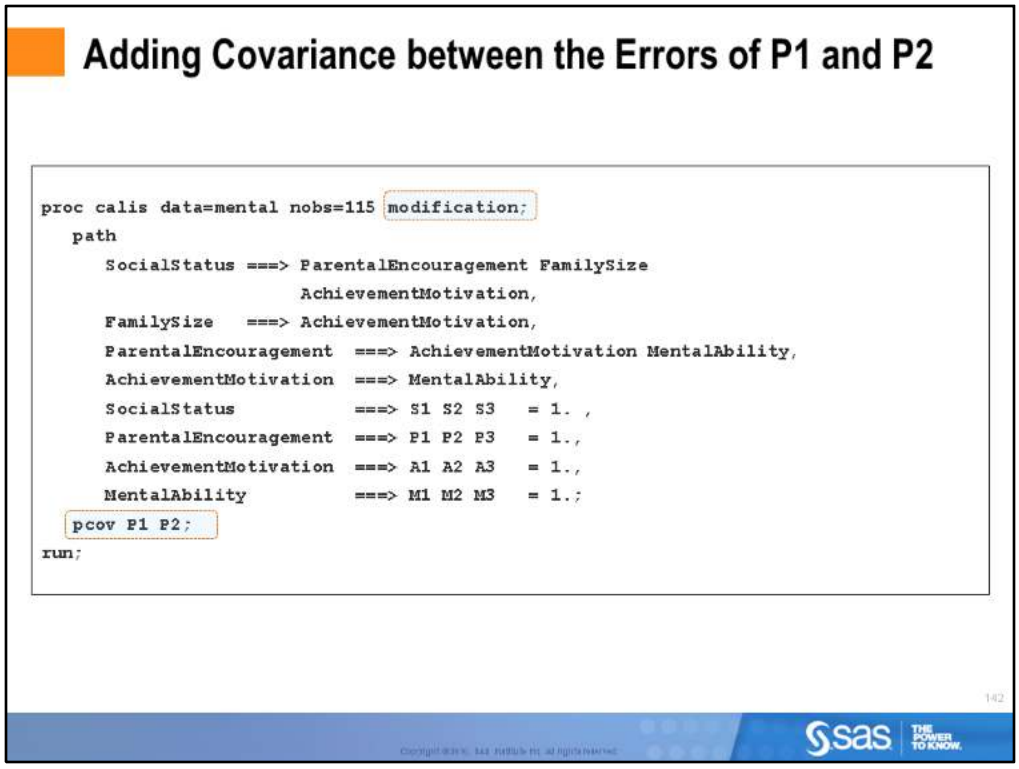


Considering the top suggestions from the results of the LM test statistics, I would add the covariance between P1 and P2. The added parameter is shown in the path diagram.

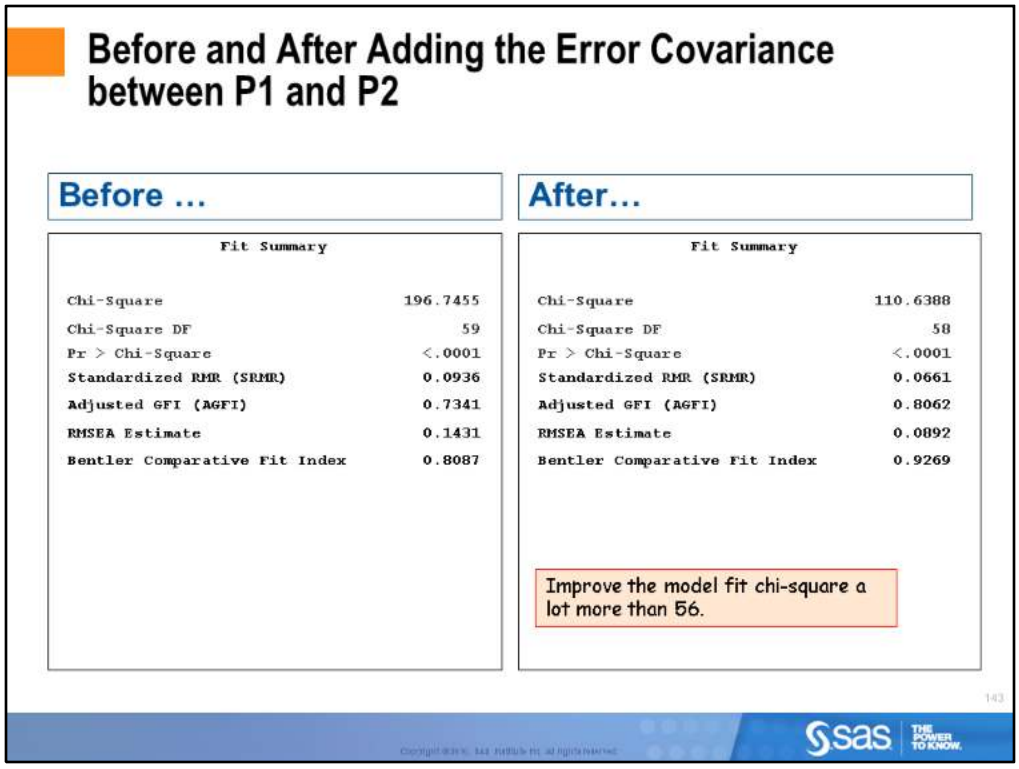
Basically, all the top LM suggestions --- the P2 ===> P1 and P1 ===> P2 paths, and the error covariance between P1 and P2 are just different manifestations of the same lack of fit about a covariance element in the original model. That is, the covariance between P1 and P2 was not well-explained by the 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.

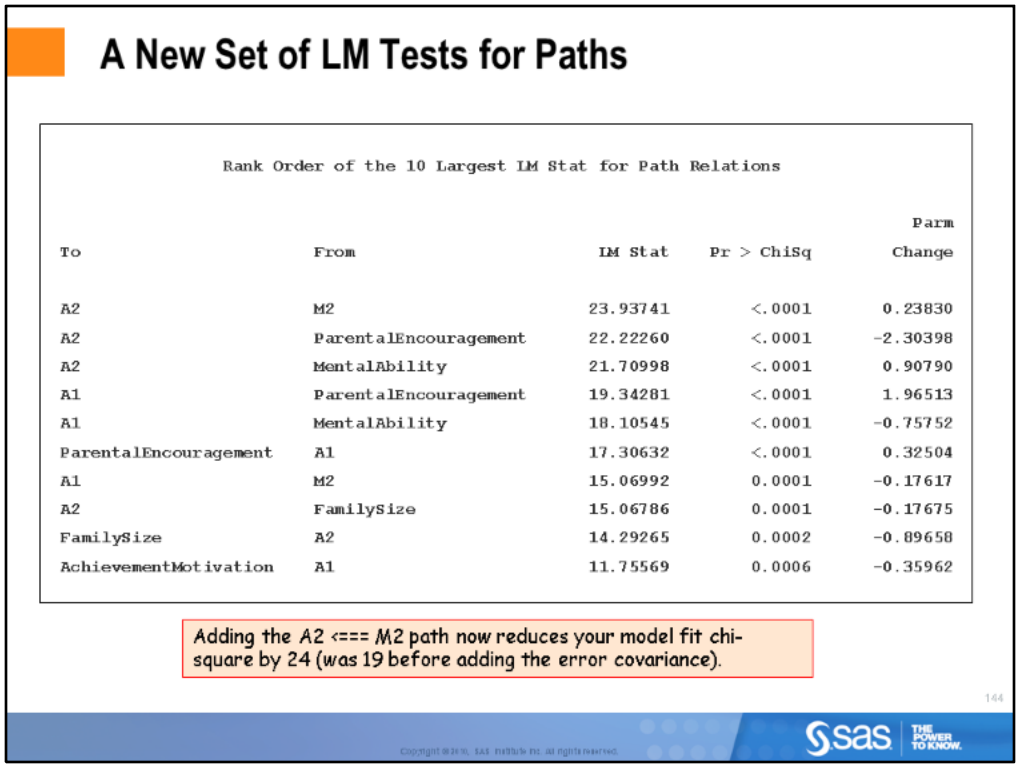


Now that I have decided to add the covariance between P1 and P2, I refit the model by adding the PCOV statement specification for the two variables, as shown in the SAS code in this slide. I also use the MODIFICATION option one more time to see if there could be any further suggested improvements.



Before 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 SRMR and the RMSEA are now close to be acceptable. The AGFI and the CFI are boosted to higher levels.



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

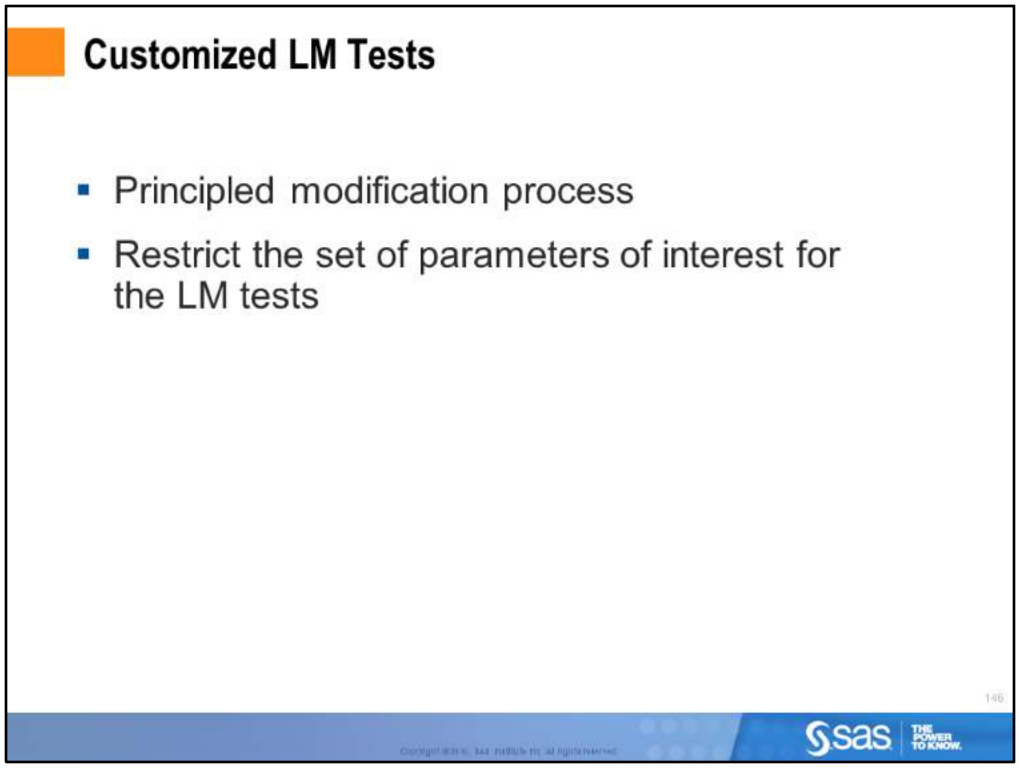
Rank Order of the 10 Largest LM Stat for Error Variances and Covariances

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

145

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

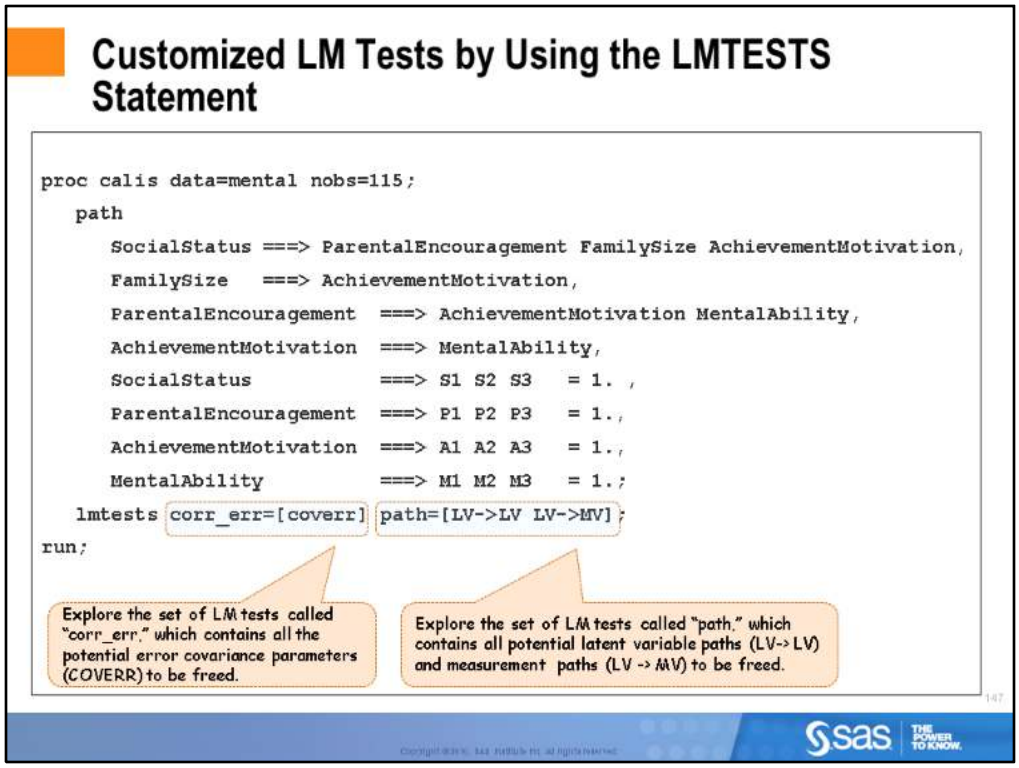
You might want to improve your model further by adding some parameters from these two LM tables, although I will not attempt to do more here.



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.

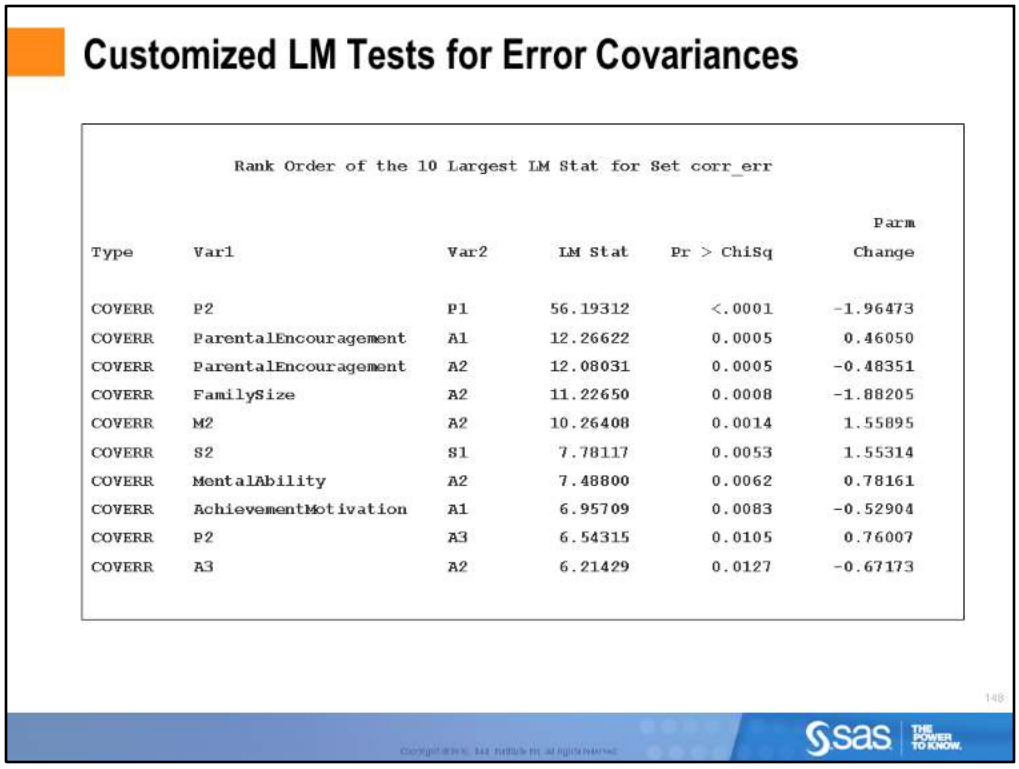
However, in some occasions you might want to restrict your attention to certain set of potential paths or parameters in your model, rather than all possible parameter space searched by the MODIFICATION option.

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

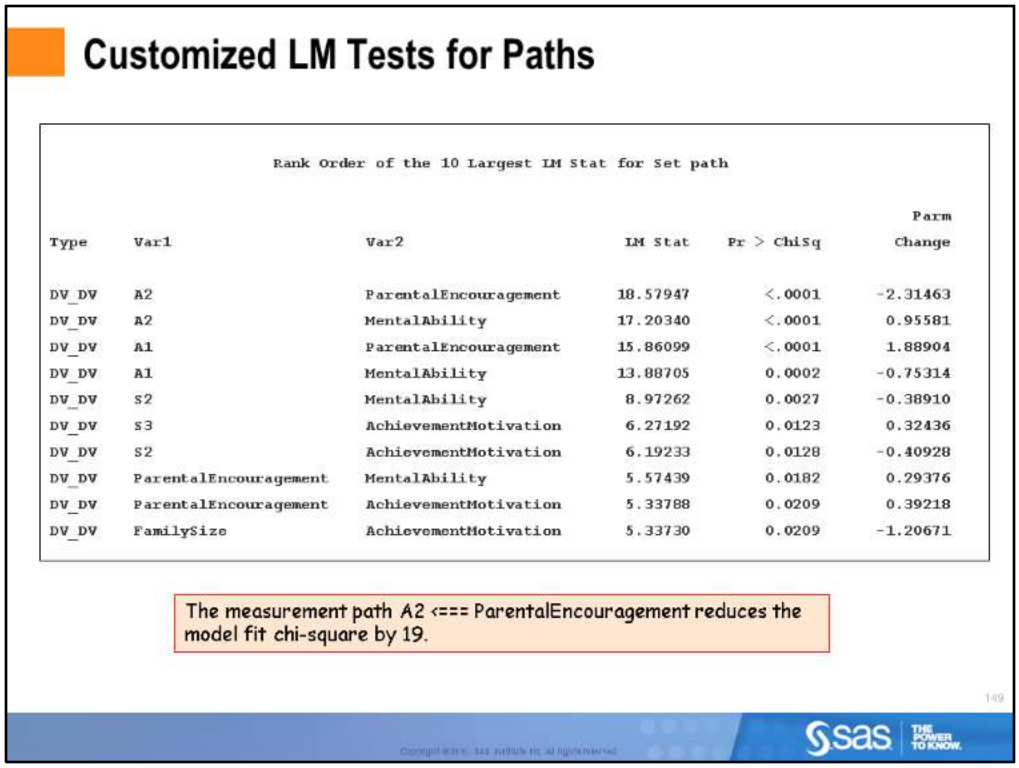


The customized LM tests define sets of parameters of interest so that your model modification process (or LM statistics output) would be limited to those sets of parameters. PROC CALIS provides the LMTESTS statement syntax to achieve the customized LM tests.

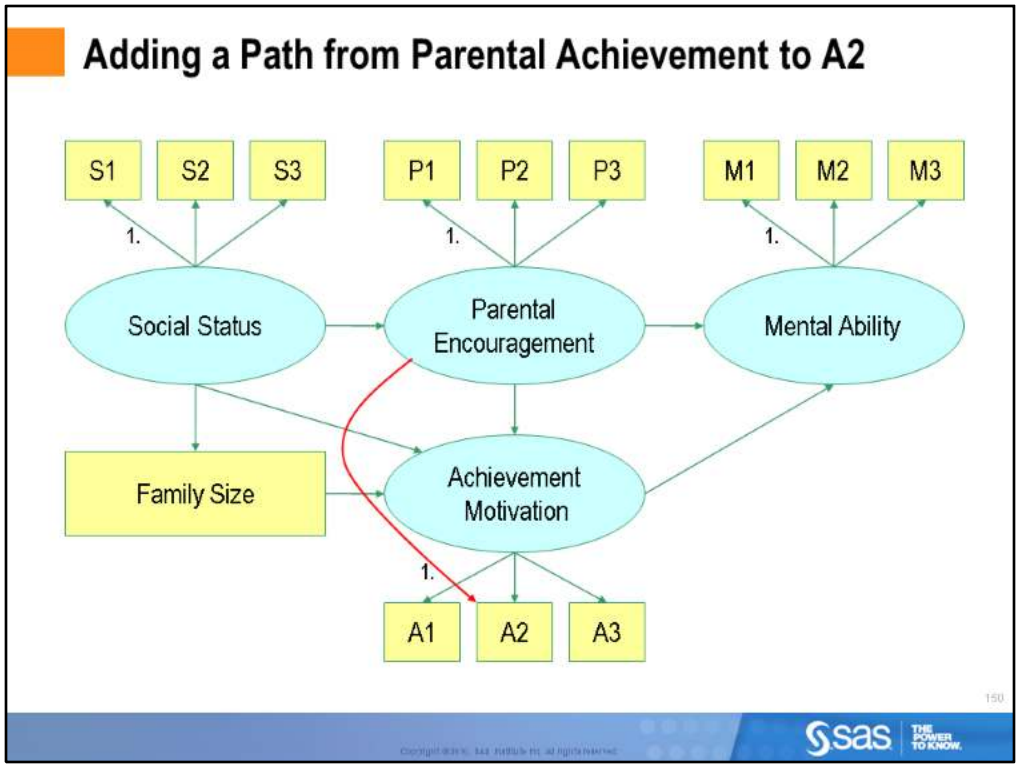
The mental ability model is used again. This time I define two sets of parameters of interest. The first set of LM tests is called “corr_err” (it is just a name you assign). This set of parameters contains the parameter region COVERR, which is a keyword that denotes all error covariances in the model. 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.



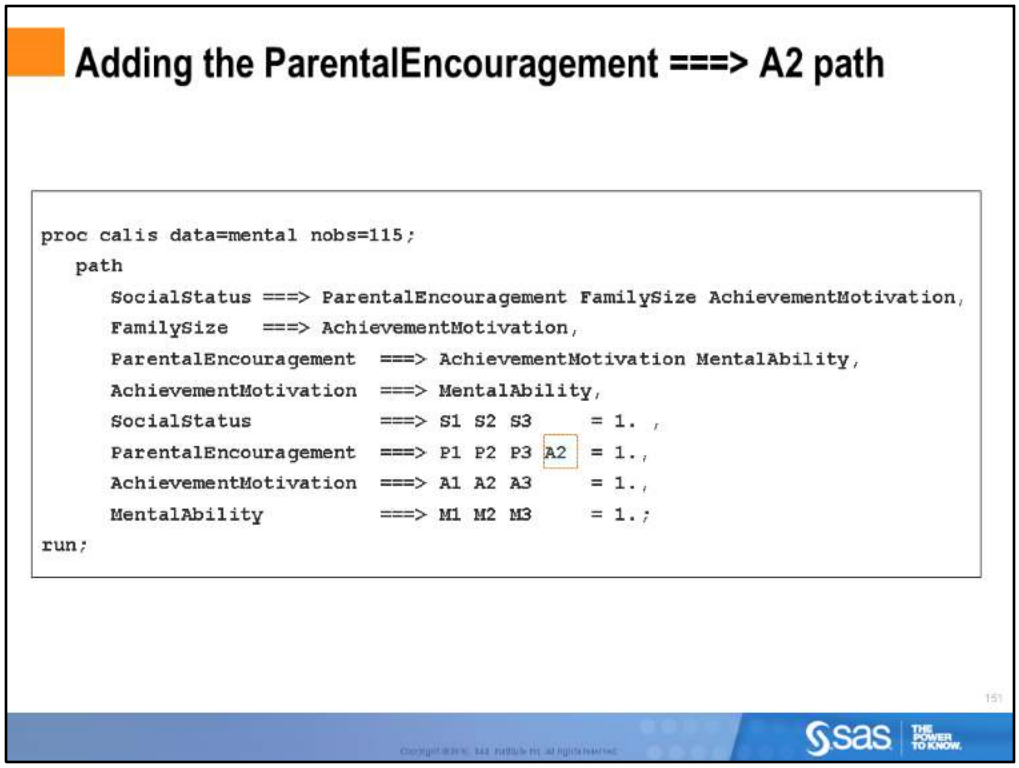
This table shows the customized LM tests of the "CORR_ERR" set. Essentially, this table is the same as one of the standard tables produced with the MODIFICATION option because both tables have the same parameter region "COVERR."



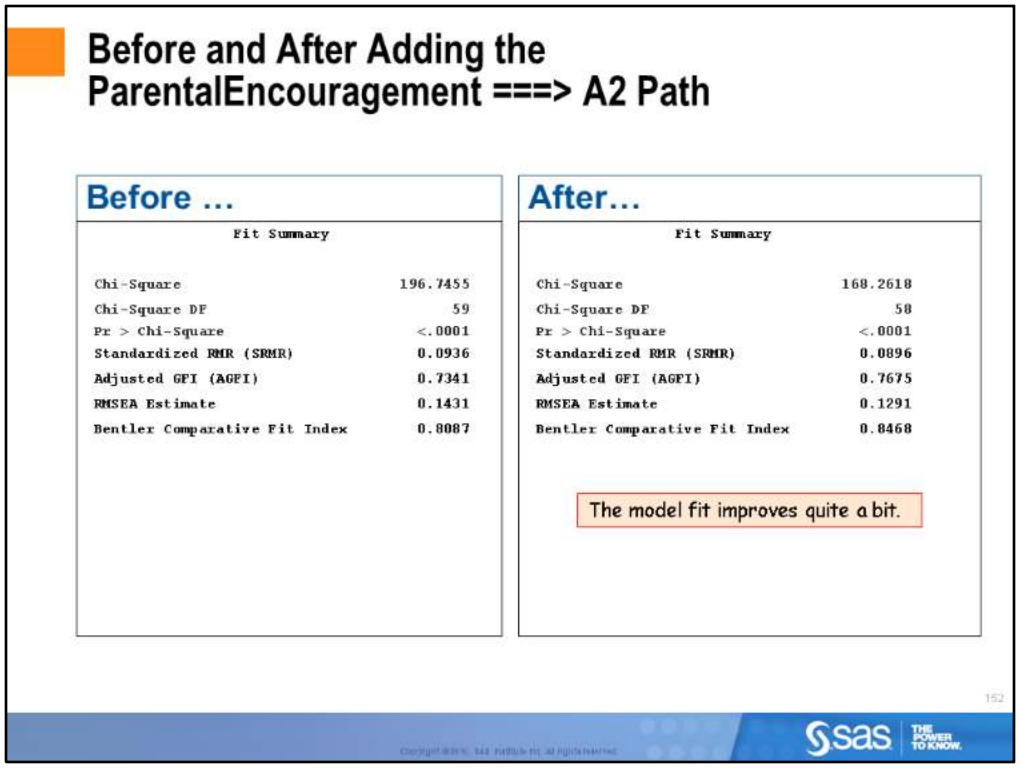
The second customized set of LM tests suggests that adding the dependent variable (DV) to dependent variable (DV) path A2 <=== ParentalEncouragement improves the model fit the most amongst all paths in the "path" set. The chi-square improvement is about 19, which is statistically significant.



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.

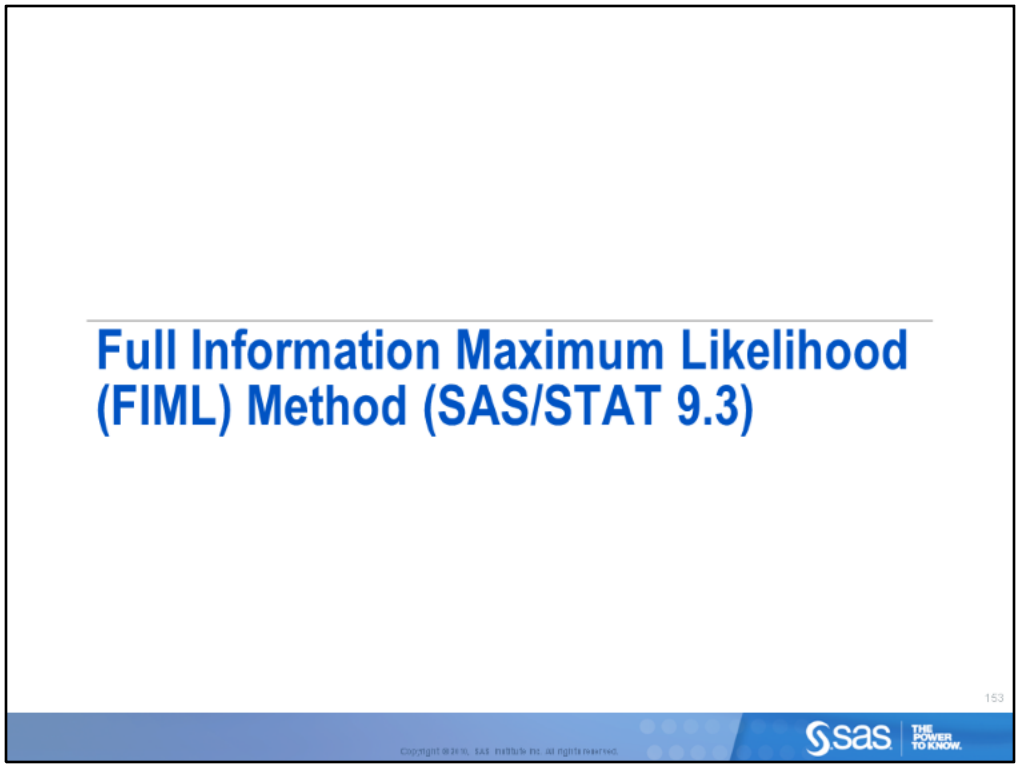


Nonetheless, you add this new path for A2, as shown in the above PROC CALIS code. All you need to do is to add A2 as one of the observed indicators of the ParentalEncouragement factor.



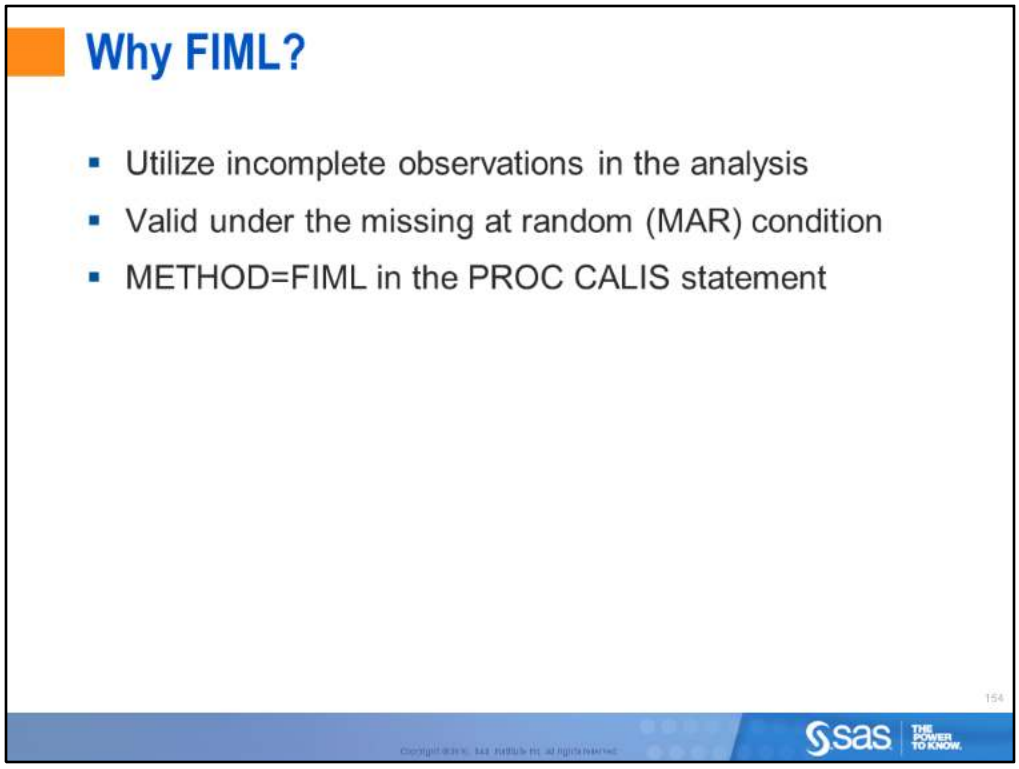
These two tables compare the fit indices before and after adding the ParentalEncouragement ===> A2 path. The model fit chi-square actually drops more than 19, which was suggested by the LM statistic in the preceding results. All other fit indices improve quite a bit too.

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.



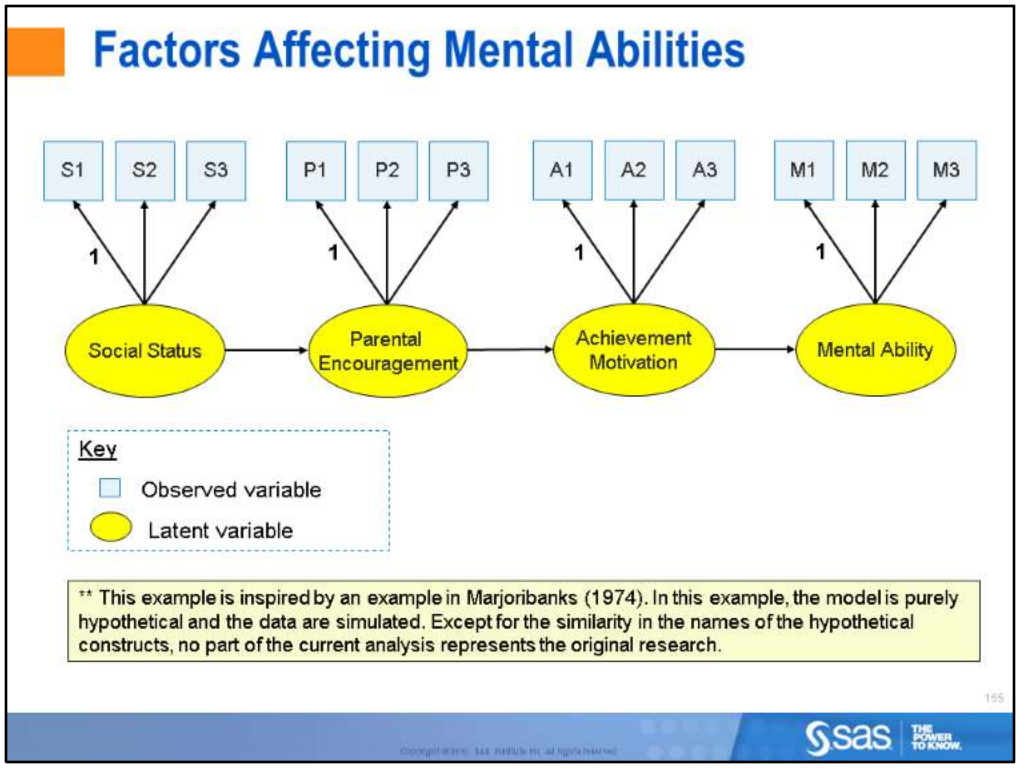
The full information maximum likelihood method in PROC CALIS is a likelihood-based method for estimating model parameters with the presence of missing values. Many SAS statistical procedures, including the CALIS procedure, delete all incomplete observations (observations with at least one missing values) from analysis by default. Therefore, even if the missing values are due to "ignorable" reasons, default estimation methods lose valuable information from the incomplete observations. By using the full information maximum likelihood method, all available values in the incomplete and complete observations are used in the estimation.

The full information maximum likelihood method is available in SAS/STAT 9.3 and later.



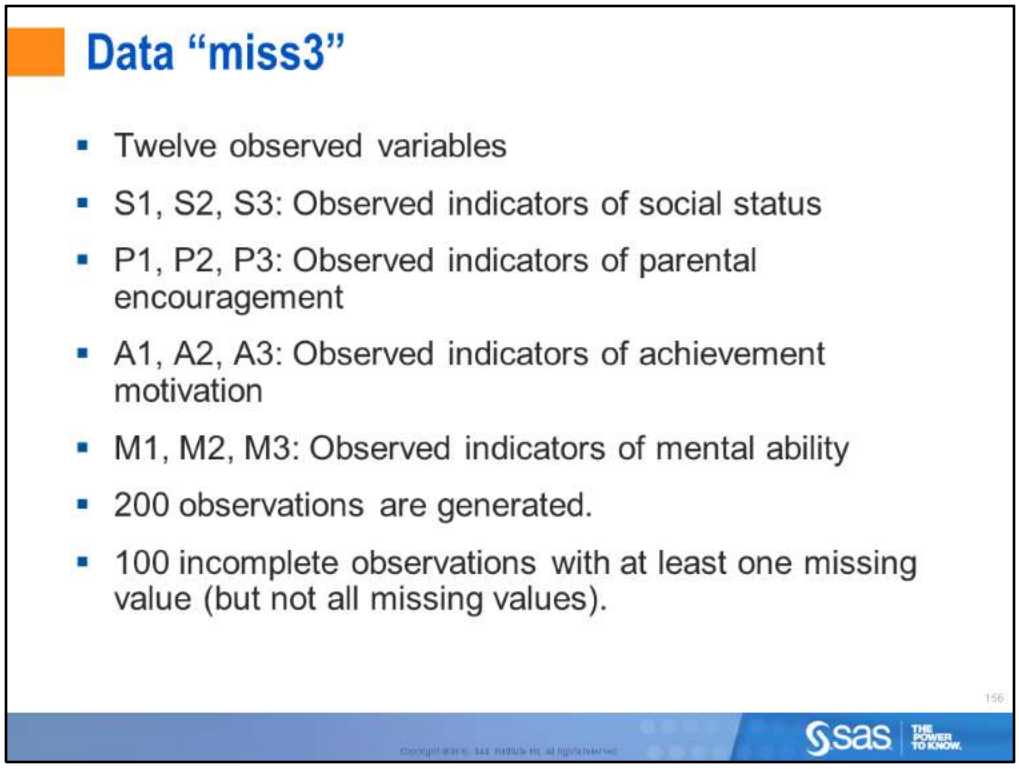
FIML ensures the maximum use of the data values. However, it assumes the missing at random condition (the MAR condition, following Rubin's definition). Simply put, the FIML estimation assumes that the missingness is not dependent on the missing values, although the missingness could be related to other variables. This assumption cannot be tested, however.

In order to use the FIML estimation, you can specify the METHOD=FIML option in the PROC CALIS statement, instead of the default ML method.



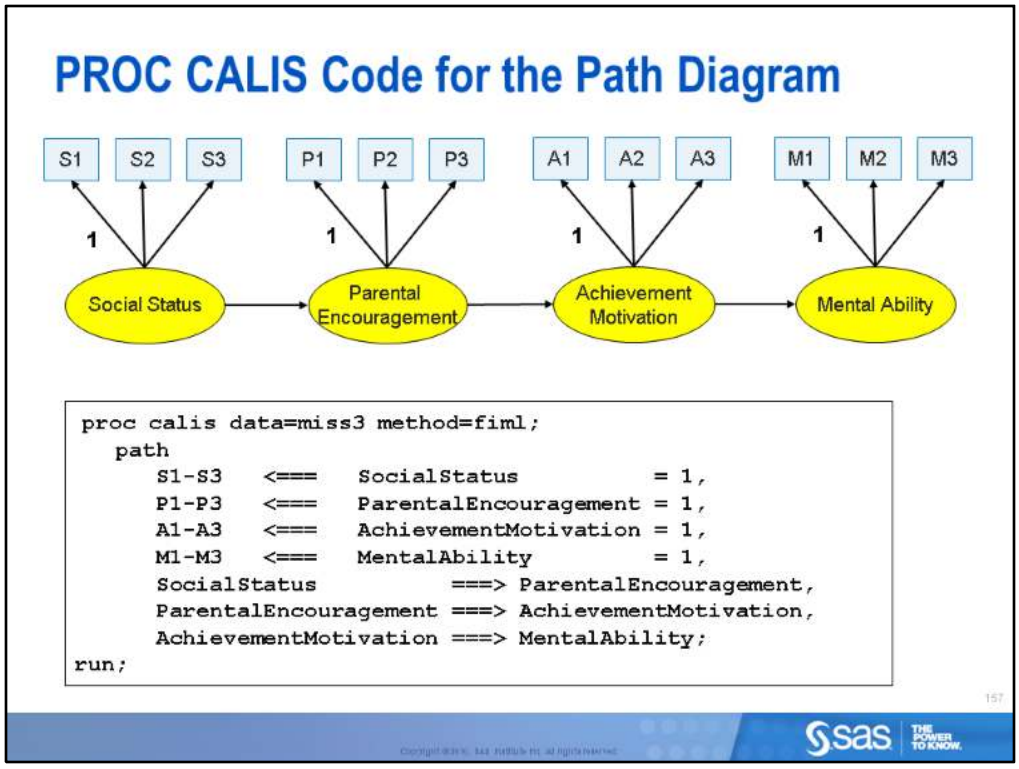
This is a model that assumes a sequential order of causal effects of Social Status on Mental Ability. The effect of Social Status on Mantel Ability is mediated by Parental Encouragement and Achievement Motivation. These variables are all formulated as latent variables in the model. Each of them has some measured indicators (S1-S3, P1-P3, and so on).

Indeed, Marjoribanks (1974) uses these variables in a research. In this presentation, I use these constructs only for demonstration purposes. Except for the similarity in the names of the hypothetical constructs, no part of the current analysis represents the original research. Both the model and the data are made up for the demonstration.



As mentioned, there are twelve variables as reflective indicators of the latent constructs in the model.

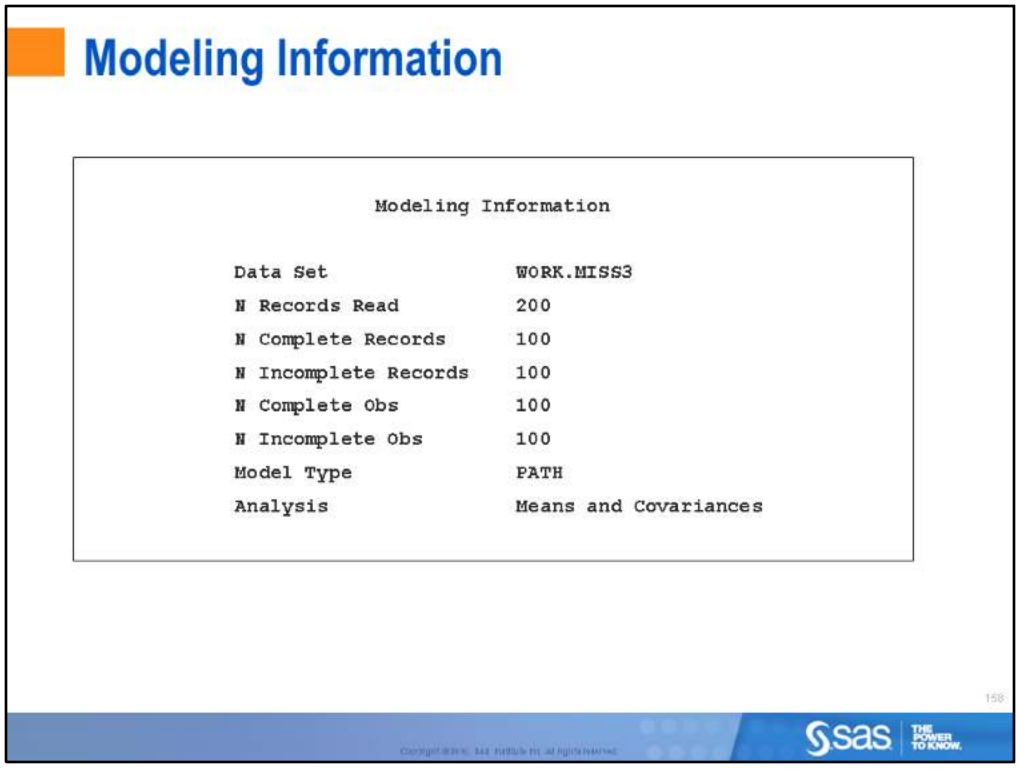
I simulated a data set of 200 observations. One hundred of these observations contain at least one missing values. The data set is called "missed3."



The PROC CALIS code for this model is very simple.

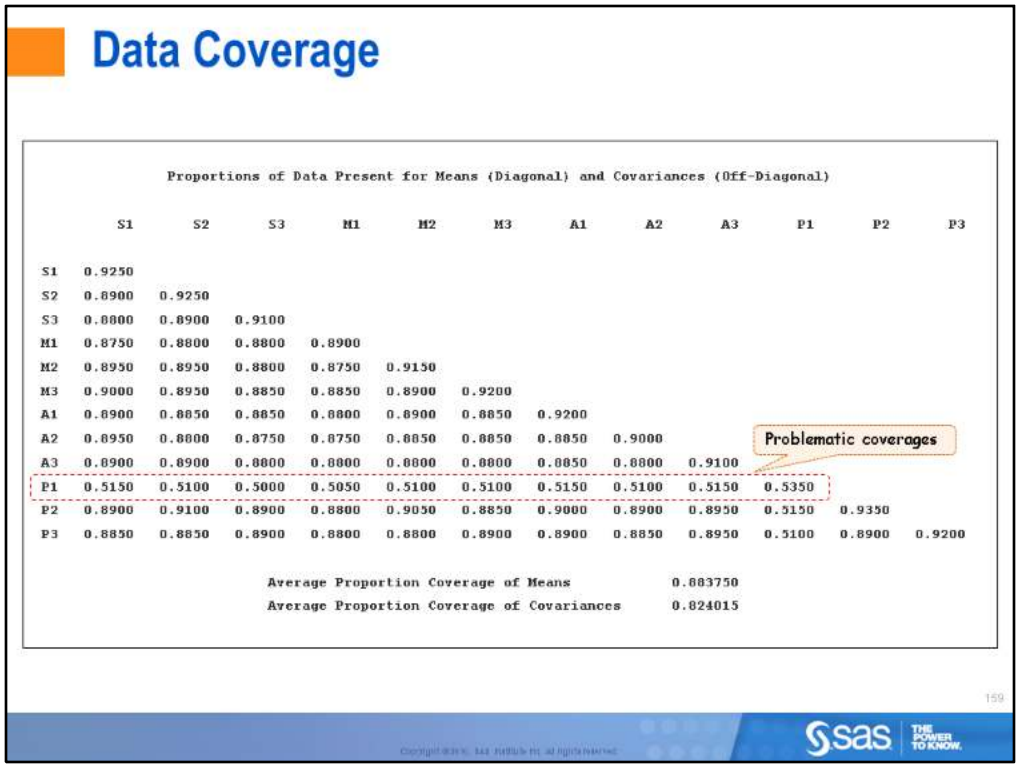
In the PROC CALIS statement, the DATA= option specifies the data set and the METHOD= option specifies the FIML method for estimation.

In the PATH statement, the first four entries specify the reflective indicators (observed variables) for the four latent constructs. I set a fixed loading of 1 to the first indicator of each latent construct. All other loadings are free parameters. In the next three entries, I specify the functional relations among the latent constructs, reflecting exactly what is hypothesized in the path diagram.



This slide summarizes the modeling information. It shows that 200 data records were read--100 of them are complete records and 100 of them are incomplete records. Because no frequency variable is used, these numbers are also for the numbers of complete observations and incomplete observations, respectively.

With the FIML method, the means of the variables are also modeled. Because no mean structures are specified in the model, saturated mean structures are assumed.



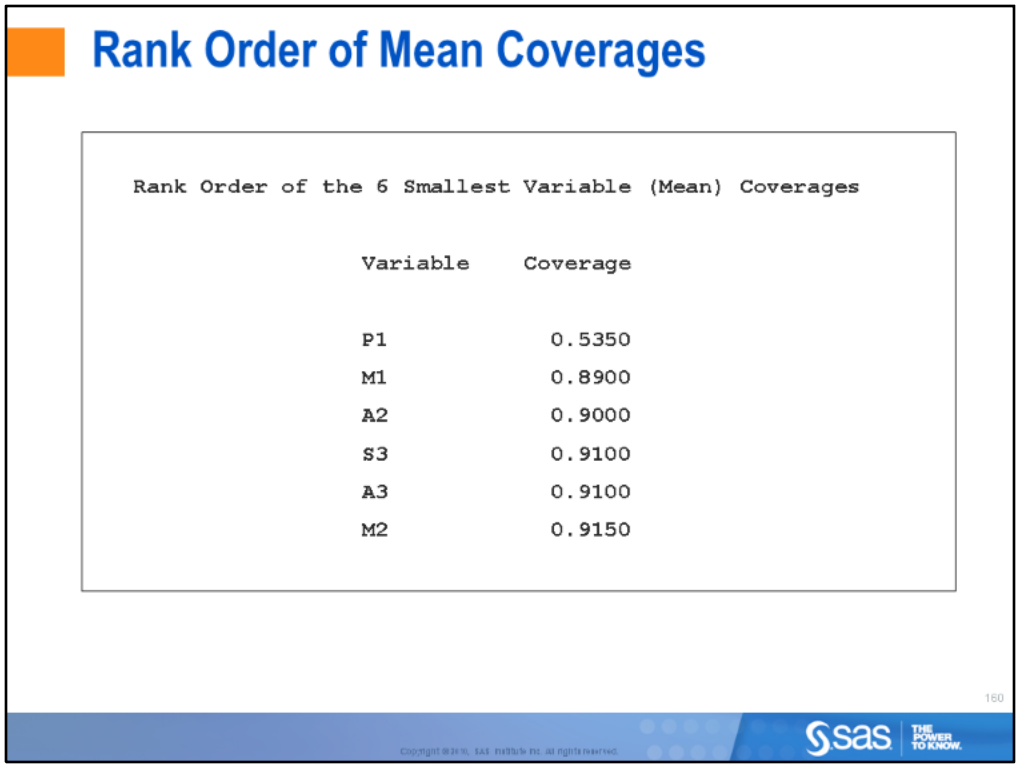
When the FIML method is requested, PROC CALIS also displays some information about the missing patterns in the data.

This table shows the proportion of data coverages in the covariance matrix. For example, for computing the covariance element (S1,S1), you only need the values of S1. This table shows that 92.5% of the S1 values are available or non-missing. So, you have a high proportion of data coverage for computing this covariance element. Certainly, this number is also the proportion of data coverage for computing the mean of S1.

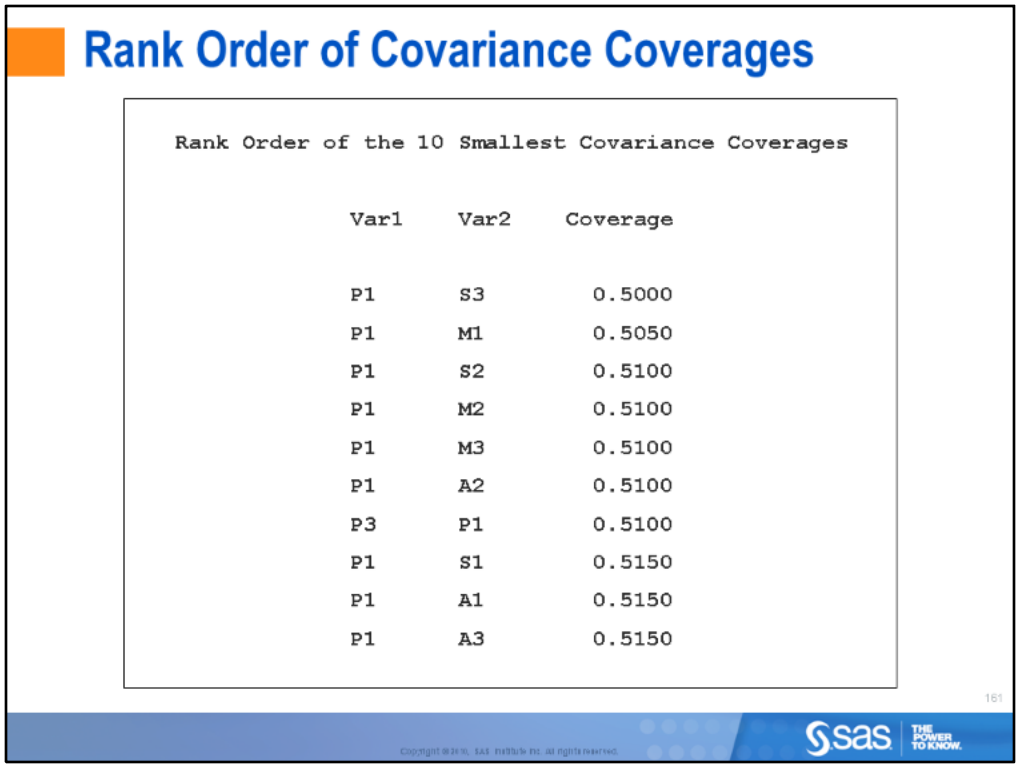
For off-diagonal elements, this table shows the proportions of joint coverage of variable-pairs. For example, to compute the covariance element (S1,S2), you need both the values of S1 and S2. The table shows that 89% of the observations have both non-missing S1 and S2 values. So, the proportion coverage for computing this covariance is still high.

Using this table, you might be able to spot the problematic coverages. For example, the proportion coverages related to P1 are all about 50%, which is much lower than other coverage values. This might tell you that something is wrong about the P1 variable.

All covariances have high proportions of data coverages, except for those with the P1 variables.

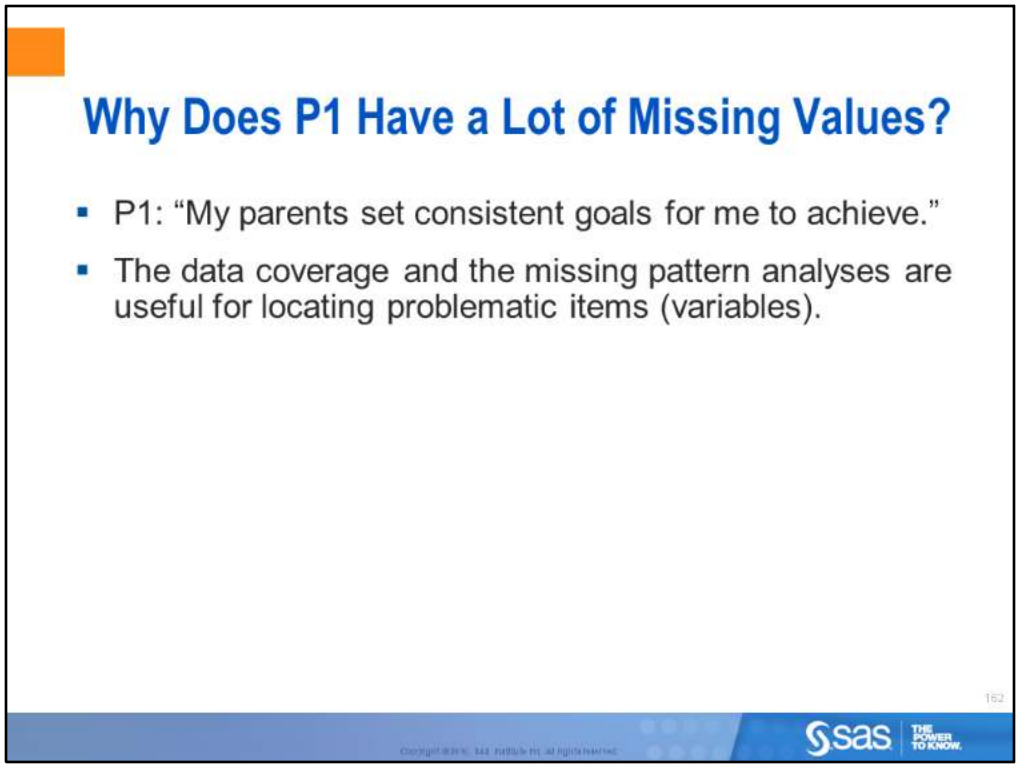


When you have a large covariance matrix, it might be more difficult to locate the problematic data coverages. To help you make a more efficient examination of the data coverages, PROC CALIS ranks the coverages and shows the smallest data coverages. This table shows the smallest coverage in the variables (that is, the smallest coverages among the diagonal elements in the covariance matrix). Clearly, variable P1 has the most serious problem in terms of data coverage.

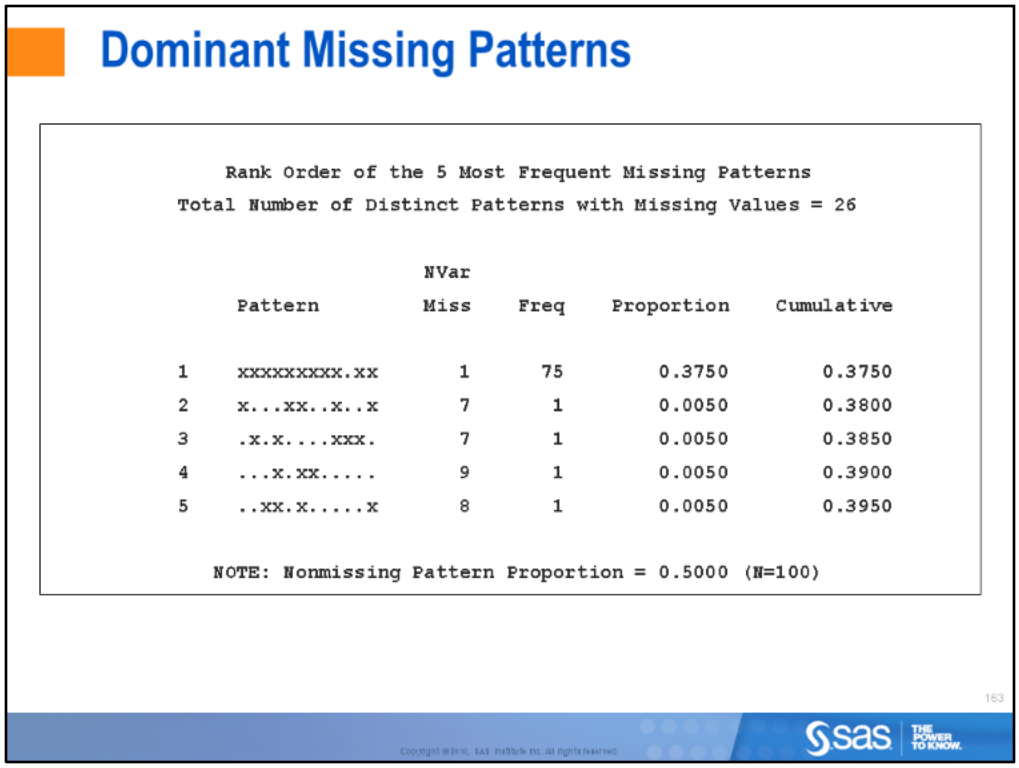


This table shows the ten lowest proportions of joint coverages of variable-pairs. The results here clearly point to the problematic nature of P1. All these lowest proportion coverages are related to variable P1.

So, what might have happened to P1 during the data collection process? This is something that practical researchers would like to find out. To complete my illustration, let me just make up the P1 variable for a possible explanation.



Suppose that the P1 item is "My parents set consistent goals for me to achieve." This item is appropriate to respondents who live with their parents. But what about the respondents who might have been living with single parents? Should they answer "Strongly Agree" to this question only because the goals must be consistent with a single parent? Or should they just choose not to respond to the question? Either way, the low data coverage of the P1 variable exposes the problematic nature of this item. The researchers might need to replace this item by a better one to avoid missing values.



Another useful output from PROC CALIS is the display of the most dominant missing patterns. For the current example, it is evident that one missing pattern is dominating. Pattern #1 has one missing variable (represented by a dot) and all other variables do not have missing values. This pattern has a distinctively high frequency of occurrence: 75. It accounts for about 38% of the observations. In the note, it shows that the proportion of nonmissing pattern (with complete data) is 50%.

So, what is the missing variable in this pattern? I think you can guess now that it is P1.

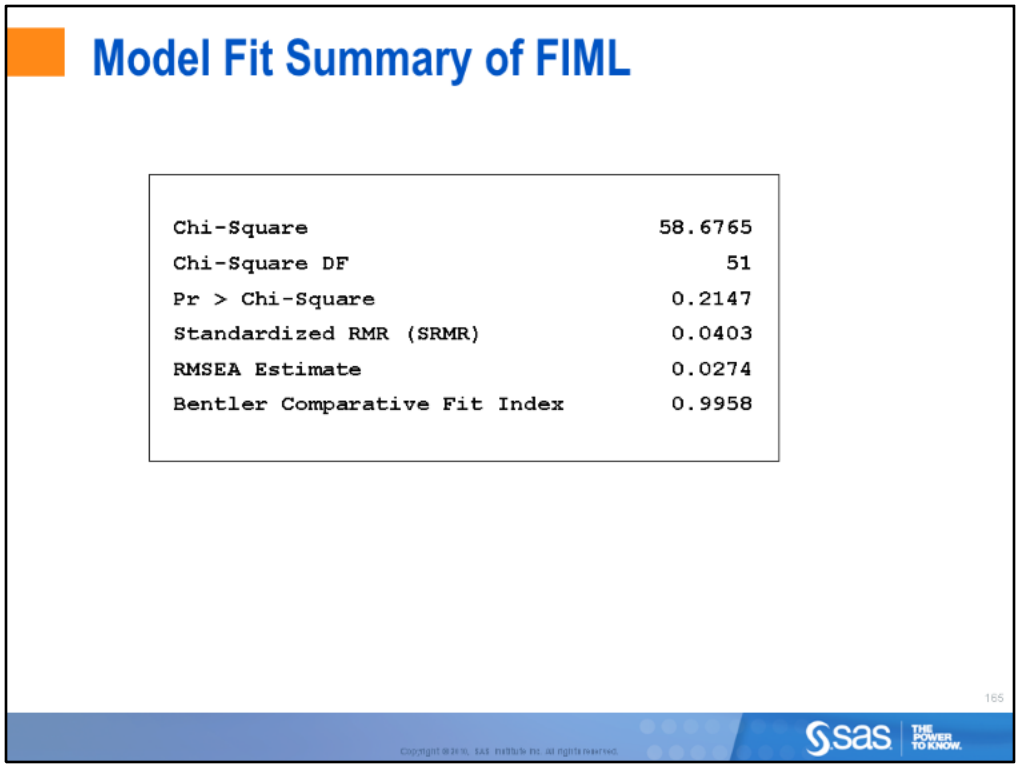
Mean Profiles of the Dominant Missing Patterns

Means of the Nonmissing and the Most Frequent Missing Patterns

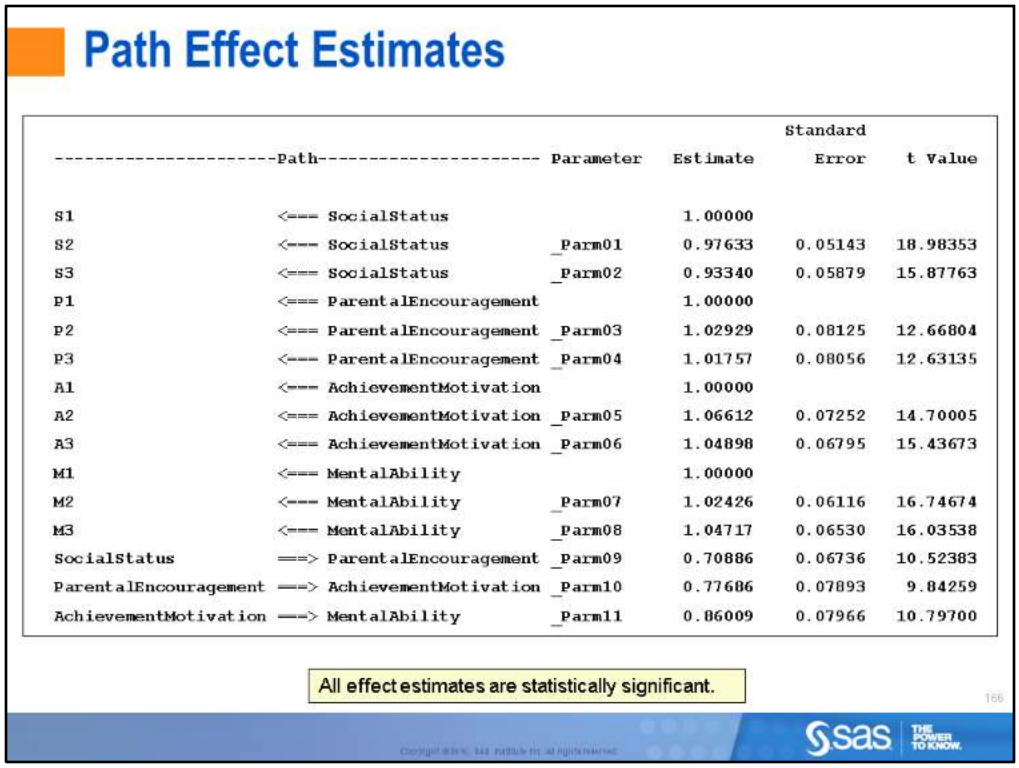
		Missing Pattern				
	Nonmissing	1	2	3	4	5
Variable	(N=100)	(N=75)	(N=1)	(N=1)	(N=1)	(N=1)
S1	4.04000	3.58667	4.00000	.	.	.
S2	3.91000	3.65333	.	3.00000	.	.
S3	4.00000	3.52000	.	.	.	7.00000
M1	4.02000	3.56000	.	1.00000	2.00000	6.00000
M2	4.04000	3.56000	6.00000	.	.	.
M3	4.01000	3.42667	3.00000	.	4.00000	6.00000
A1	4.18000	3.66667	.	.	4.00000	.
A2	4.29000	3.50667	.	.	.	.
A3	4.30000	3.46667	4.00000	2.00000	.	.
P1	4.08000	.	.	3.00000	.	.
P2	4.15000	3.73333	.	3.00000	.	.
P3	4.06000	3.70667	6.00000	.	.	6.00000

This output supplements the previous one with the display of the means of the nonmissing variables in the dominant missing patterns and in the nonmissing pattern. You can also use this table to locate the missing variables in the missing patterns.

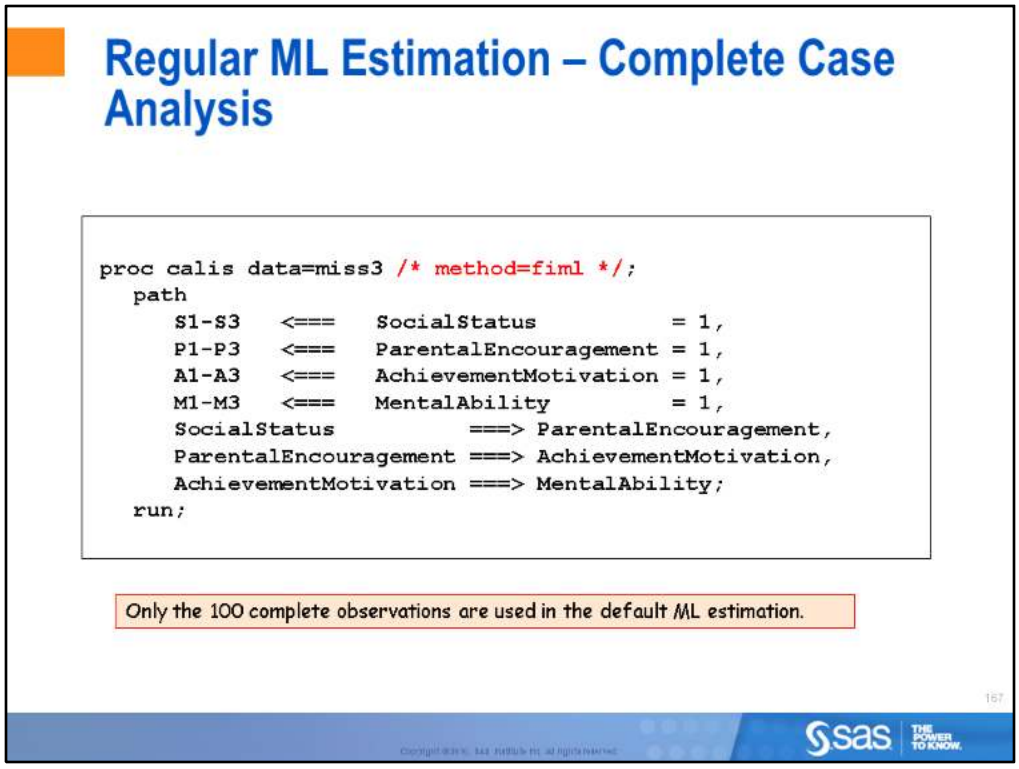
A dot in this table represents the corresponding missing variable in the patterns. For example, in missing pattern #1 (the most dominant missing pattern), P1 is the missing variable. Comparing the means of this missing pattern with the nonmissing patterns shows that all the means in this pattern are lower. Is this just a coincidence? Or, is this missing pattern represent a meaningful sub-population that has a lower mean profile? I do not know the answer here. But empirical researchers might be interested in following up this kind of questions after examining the mean profiles of the missing patterns.



The FIML method also displays model fit summary. In this example, the model seems to be very good. Model fit chi-square in not significant. SRMR and RMSEA are both less than 0.05. Bentler's CFI is well above .90.

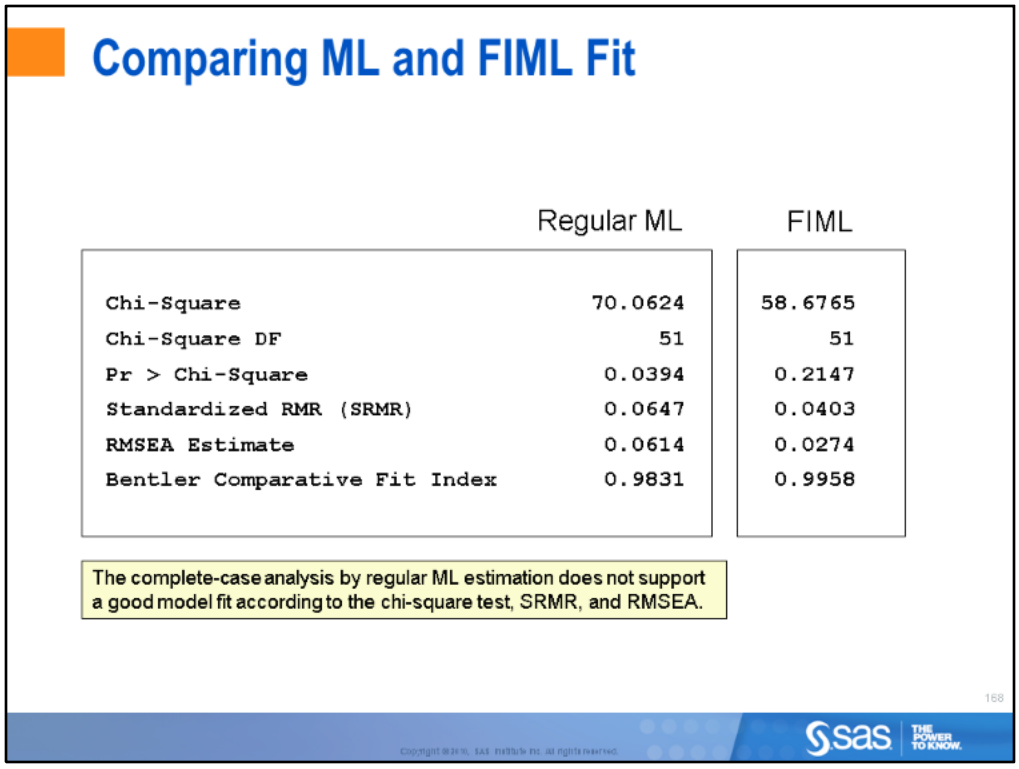


All the path coefficients or effects are statistically significant. This is a good sign for the model---you did not put useless paths in the model.



What if you do not use the FIML method for estimation? Will it give you the same estimation results?

To use the default ML method, let's just comment out the METHOD=FIML option.



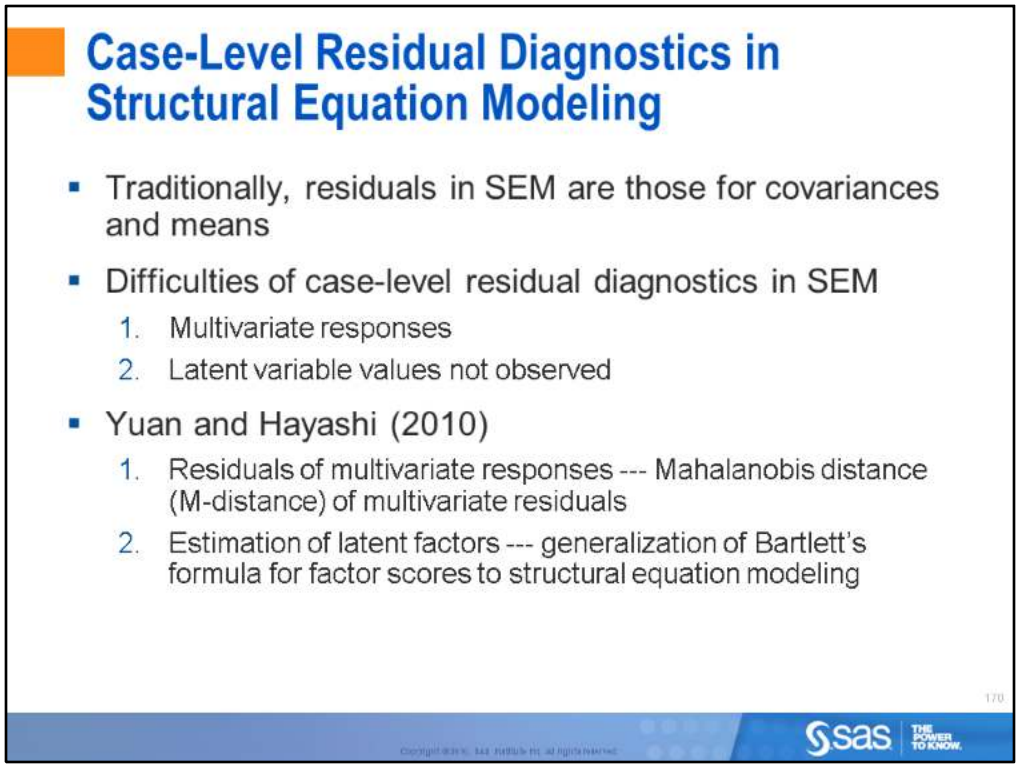
With the regular ML method, the chi-square statistic is 70.0624 (df=51, p =.0394), which is significant. Both the SRMR and RMSEA are larger than .05. These indicate bad model fit, even though Bentler's CFI is still showing a good model fit. For this particular example, it appears that the ML method fails to obtain better evidence for a good model fit from the incomplete observations. The FIML estimation does show a much favorable picture of model fit.

Certainly, this example does not mean that you will always get better model fit with the FIML method, as compared with the ML method. The most important idea is that if you have a large proportion of incomplete observations, it might be better to use the FIML estimation so that your statistical decisions can be based on as much information as available.

Case-level residual diagnostics is an old topic in regression analysis, but it is relatively new in structural equation modeling.

Some popular topics of case-level residual diagnostics include the detections of outliers and leverage points, studying the linearity of the case-level residuals, and so on.

The case-level residual diagnostic capability will be available in PROC CALIS in SAS/STAT 12.1, scheduled to be released in Summer/Fall 2012l.

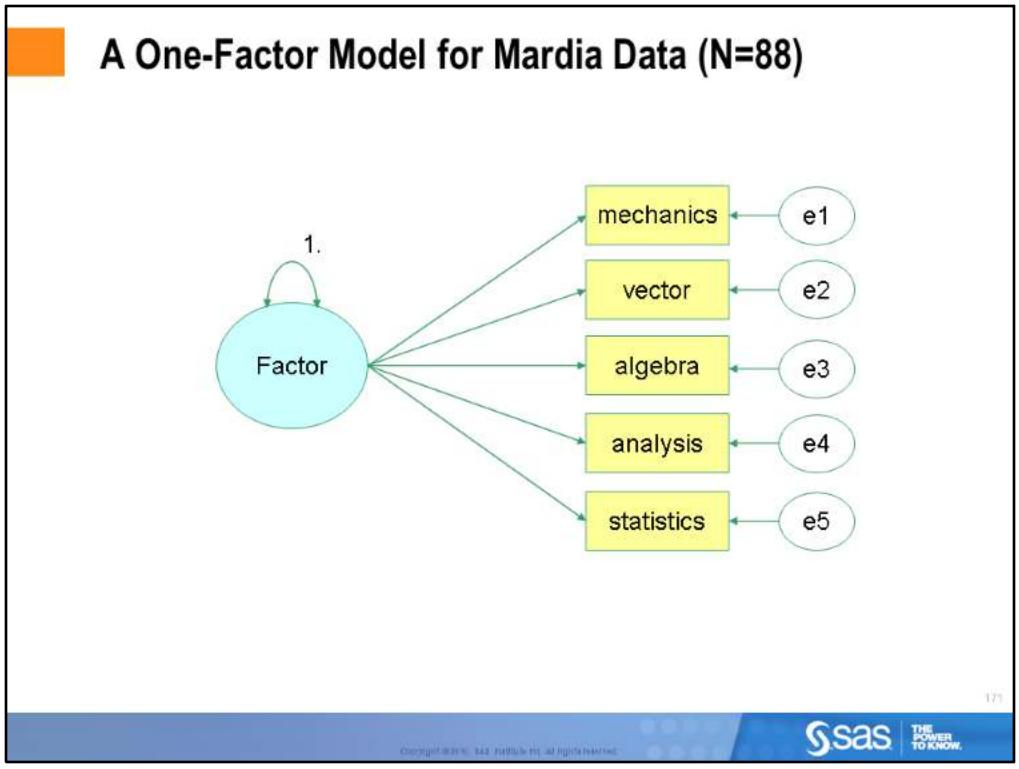


Traditionally, residuals in structural equation modeling refers to those of the covariance and mean elements. Residual analysis has not been well explored at the observation level in SEM. The difficulties of case-level or observation-level residual analysis in SEM are due to the involvements of multivariate responses and the latent variables.

In regression analysis, there is only one response variable in a regression equation. This makes it easy to define outliers by the magnitudes of the residuals. However, in SEM you usually have multivariate responses. The way to define outliers has to be done with an overall measure of residuals.

In regression analysis, both response and predictor variables are observed. There would be no difficulty in defining leverages and residuals using the observed and predicted values of the variables. However, in SEM latent variables are not observed and must be estimated. To compute leverages and residuals, you must also estimate the factor scores first.

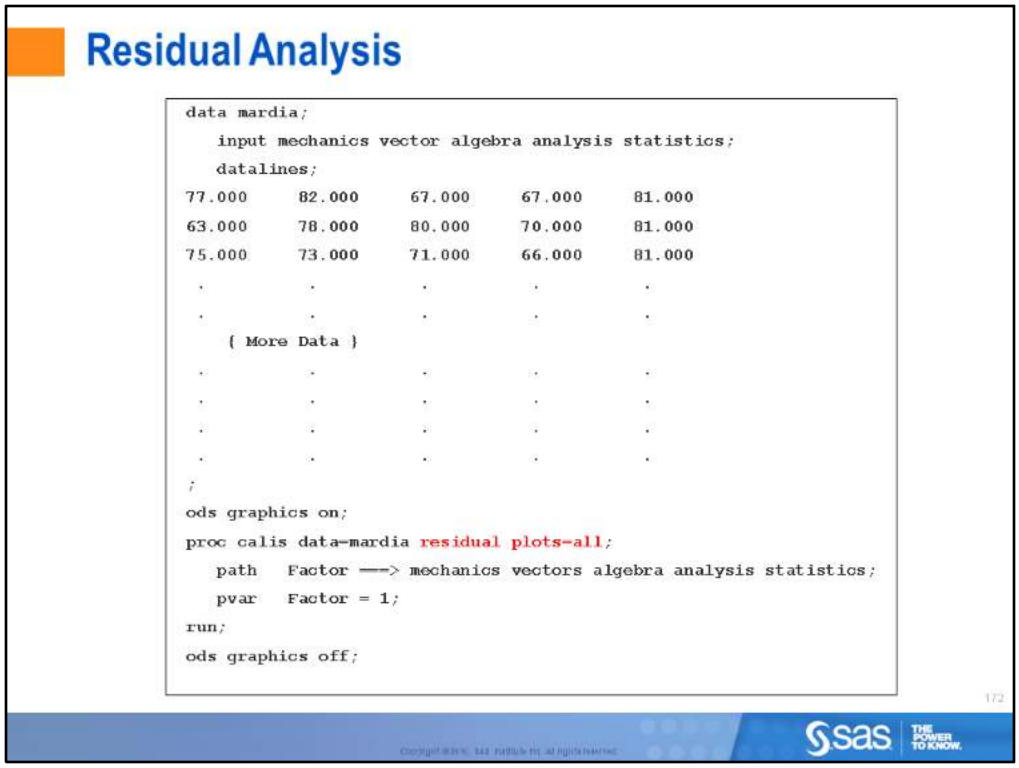
These two issues have been dealt with in Yuan and Hayashi (2010) paper. Basically, multivariate residuals are reduced to a single measure called residual M-distance for each observation and the estimation of factor scores by Bartlett's formula has been generalized to structural equation modeling.



Let us use an example to demonstrate this newer residual diagnostic technique in structural equation modeling.

In the path diagram, a one-factor confirmatory factor model is fitted to the data set in Mardia's book. The measured variables are test scores in five subjects: mechanics, vector, algebra, analysis, and statistics. There are 88 observations in the data set.

The current path diagram shows the error variables explicitly. The primary purpose is to illustrate the role of these error variables in the model. These error variables are the parts of the measured variables that are not predicted by the common factor. You can treat these error variables as latent variables---that is, they are not observed. After the factor scores are estimated for the individuals, the residuals are computed as the differences between the observed values and predicted values of the variables, much like the way that is done in regression analysis.

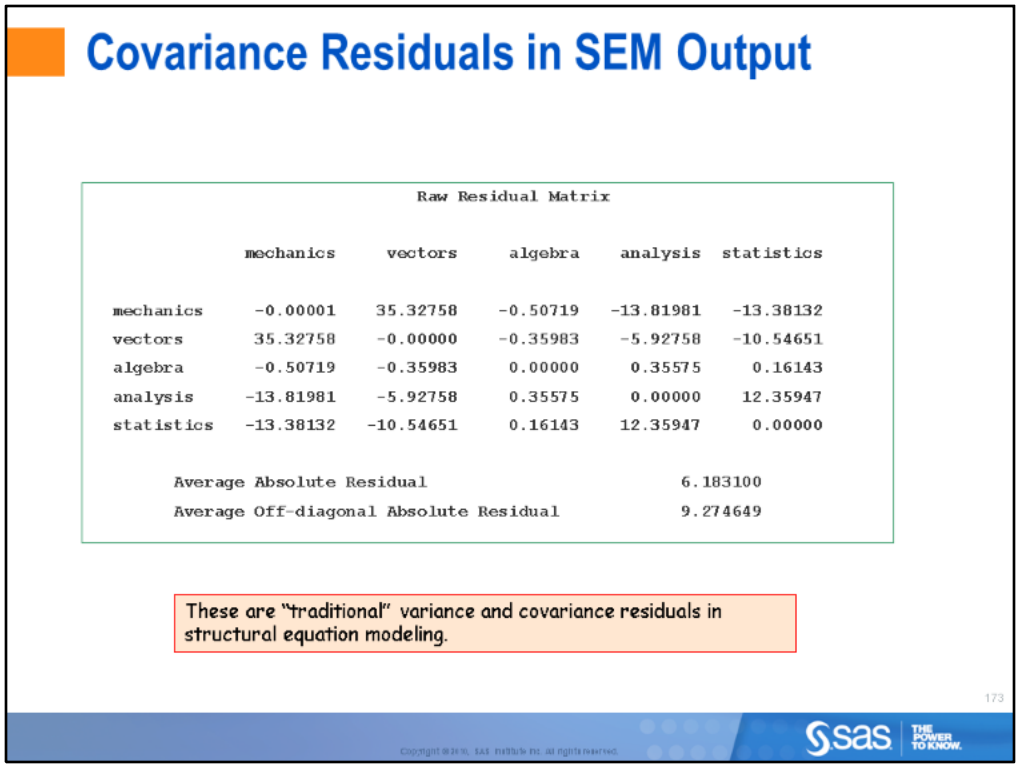


To do case-level residual analysis, you can use the RESIDUAL option in the PROC CALIS statement. The PLOTS=ALL option plots all available graphics in PROC CALIS. This will include several plots for case-level residual analysis.

The PATH statement specifies the relationships between the latent variable and the observed variables.

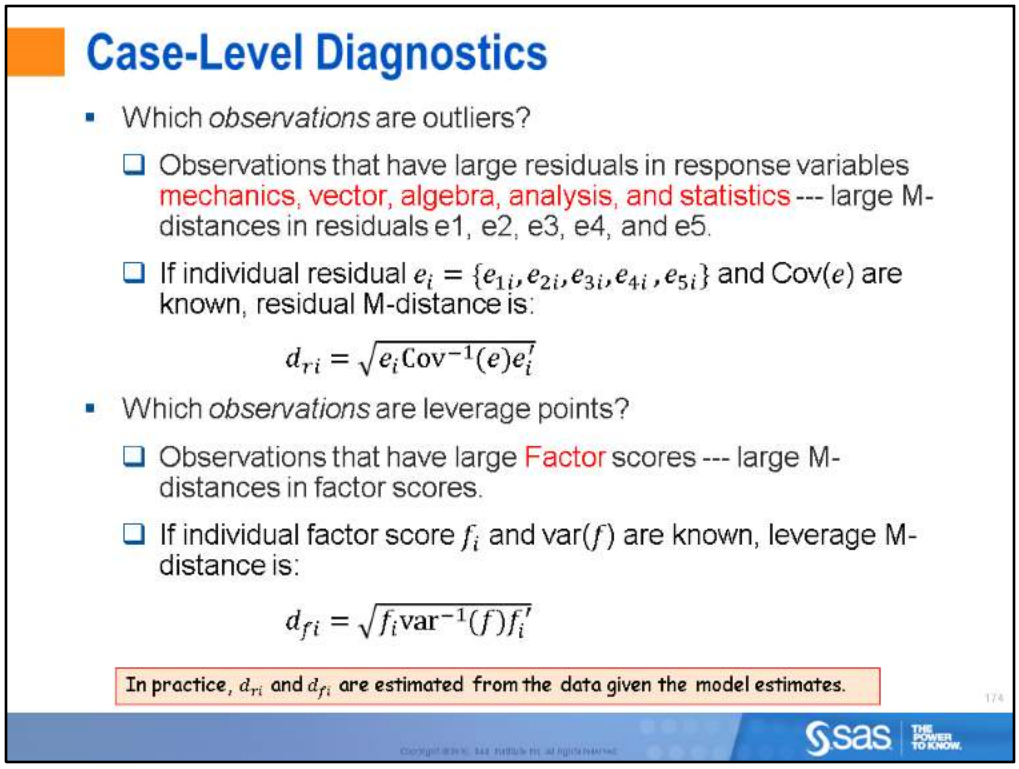
The PVAR statement fixes the variance of the factor to 1 for the identification of factor scale.

The ODS GRAPHICS statement is used to request quality graphics.



First, let us look at some more traditional residual analysis in SEM output. This slide shows the residuals of the covariances. The confirmatory factor model fits the diagonal elements of the covariance matrix perfectly, while some covariance residuals are large compared to others. For example, the residual for the covariance between mechanics and vectors is 35. This reflects the difference between the fitted covariance and the observed covariance.

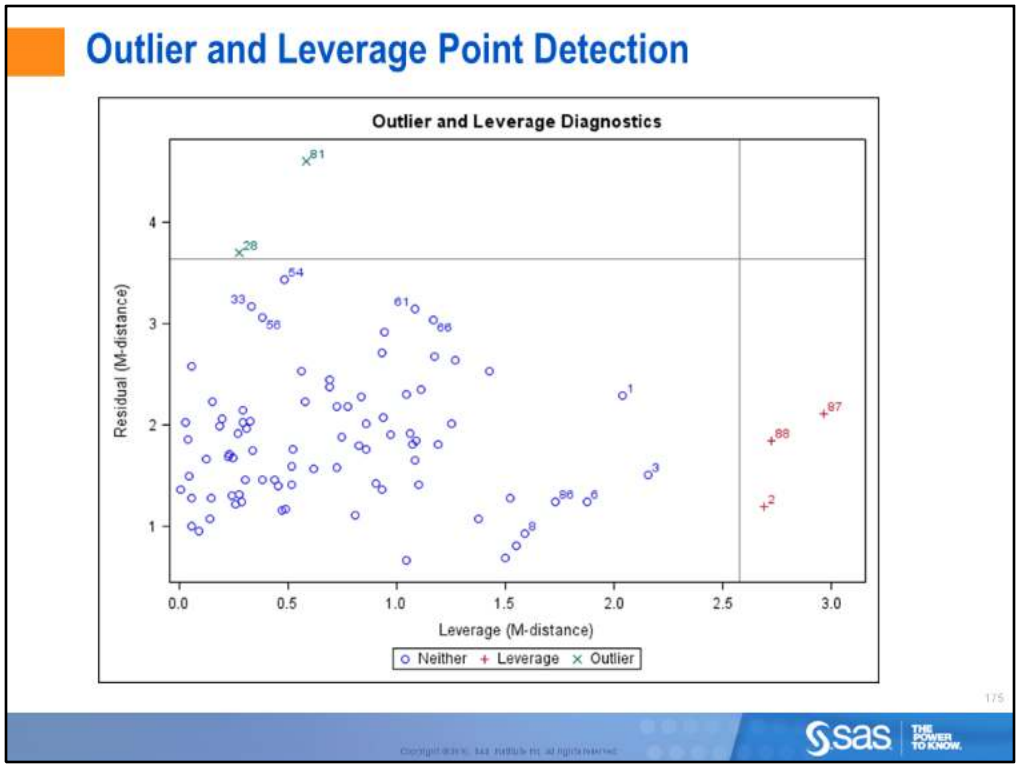
These residuals are not about individual observations. Case-level residual diagnostics use residuals computed for each individual in the raw data.



Two most popular questions in case-level residual diagnostics are: Which observations are outliers? Which observations are and leverage points?

To detect the presence outliers, a measure called residual M-distance is used. Applying to the current example, residual e_i in the five dependent observed variables is computed for each individual. Then, the Mahalanobis distance d_{ri} is computed for each multivariate residual by using the formula in the slide. In practice, residuals and their covariance matrix are estimated from the sample. Outliers would be those observations that have exceedingly large residual M-distances.

To identify leverage points, a measure called leverage M-distance is used. Applying to the current example, the predictor variable f_i and its variance var(f) are used in the formula for computing leverage M-distance d_{fi}. In practice, the factor score f_i and var(f) are estimated from the sample. Leverage points would be those observations that have exceedingly large leverage M-distances.

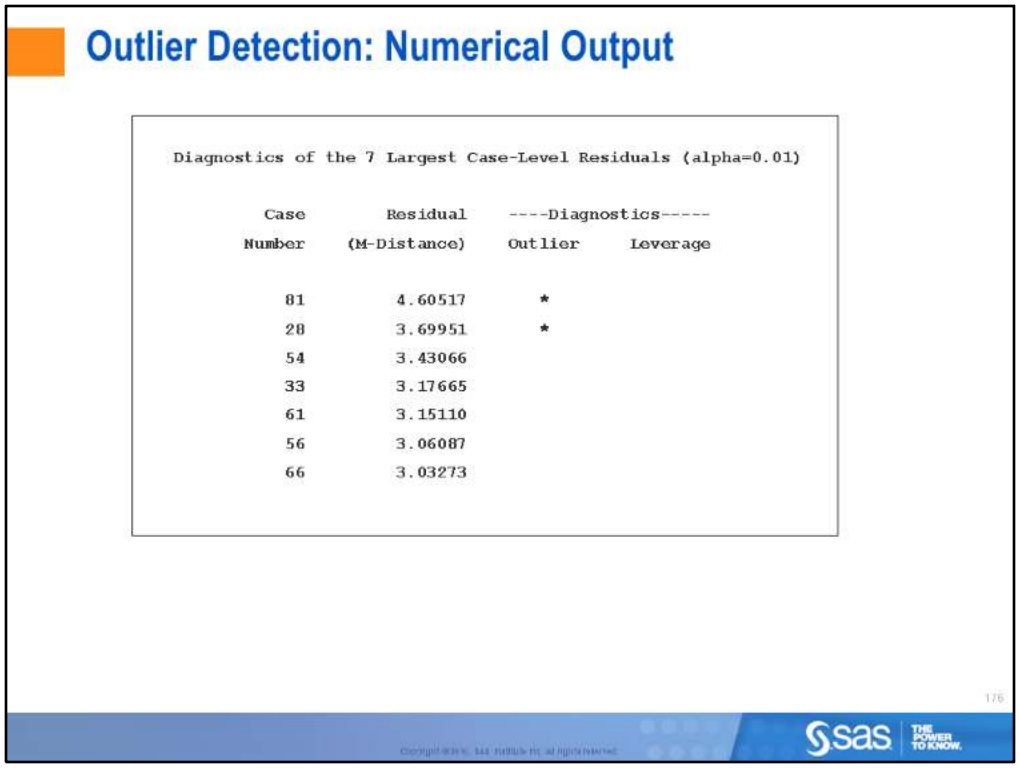


After computing the two types of M-distances for all observations, PROC CALIS plots the observations in a two-dimensional space. The Y-axis represents the residual M-distances and the X-axis represents the leverage M-distances.

To define outliers, a criterion based on certain alpha-level has to be used. In the current example, the alpha-level is 0.05, which is the default value. The horizontal line at about y=3.6 in this plot represents the criterion for outlier detection. Points above this horizontal line are outliers. Observations 28 and 81 fall into the region of outliers.

Similarly, the vertical line at about x=2.6 in this plot represents the criterion for detecting leverage points. Points to the right of this vertical line are leverage points. Observations 2, 87, and 88 fall into the region of leverage points.

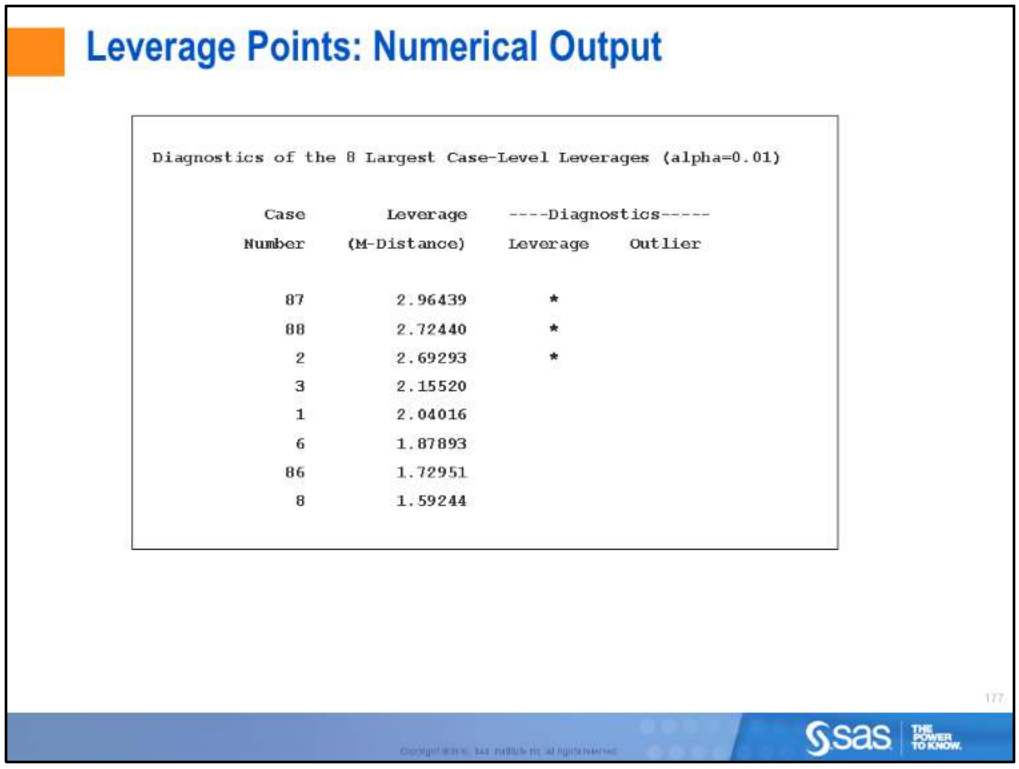
Notice that the upper right region is for observations that are both outliers and leverage points. However, this example does not have any observations in this region.



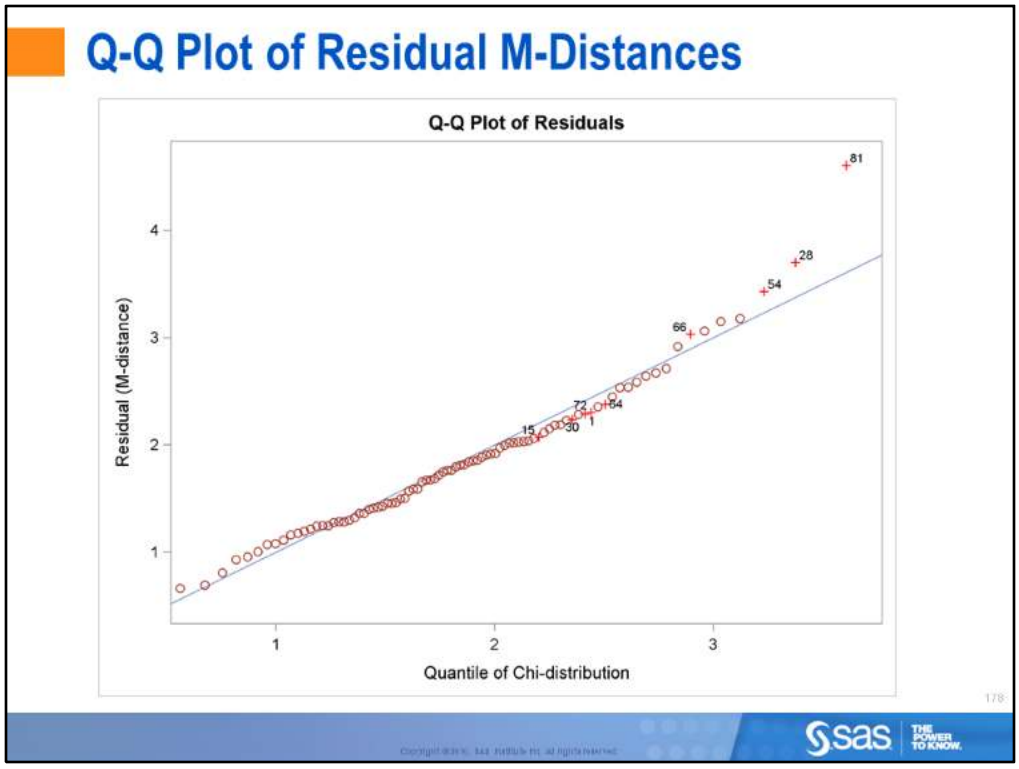
It is useful to supplement the plot of outliers and leverage points with some numerical results.

PROC CALIS outputs the observations with the largest residual M-distances and the largest leverage M-distances in two separate tables.

This slide shows the 7 observations with the largest residual M-distances. Only the first two are classified as outliers. None of them are leverage points.



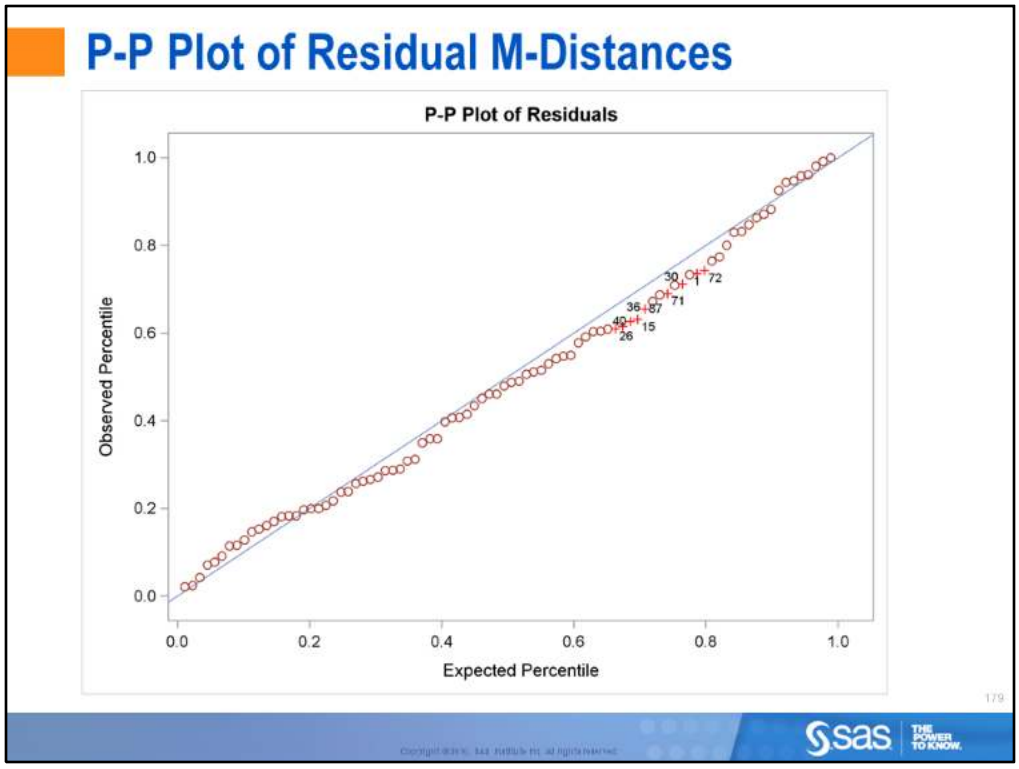
This slides shows the eight observations with the largest leverage M-distances. Only three of them are leverage points. None of them are outliers.



Once residual M-distances are computed, PROC CALIS can plot them against the theoretical quantiles. This plot is known as the Q-Q plot in regression diagnostics. In SEM, you can do the same kind of Q-Q plot to see if the residuals distribute similarly to that of the theoretical distribution. If the residuals are distributed exactly like the theoretical distribution, all observations should fall on the straight line with slope=1 in the Q-Q plot.

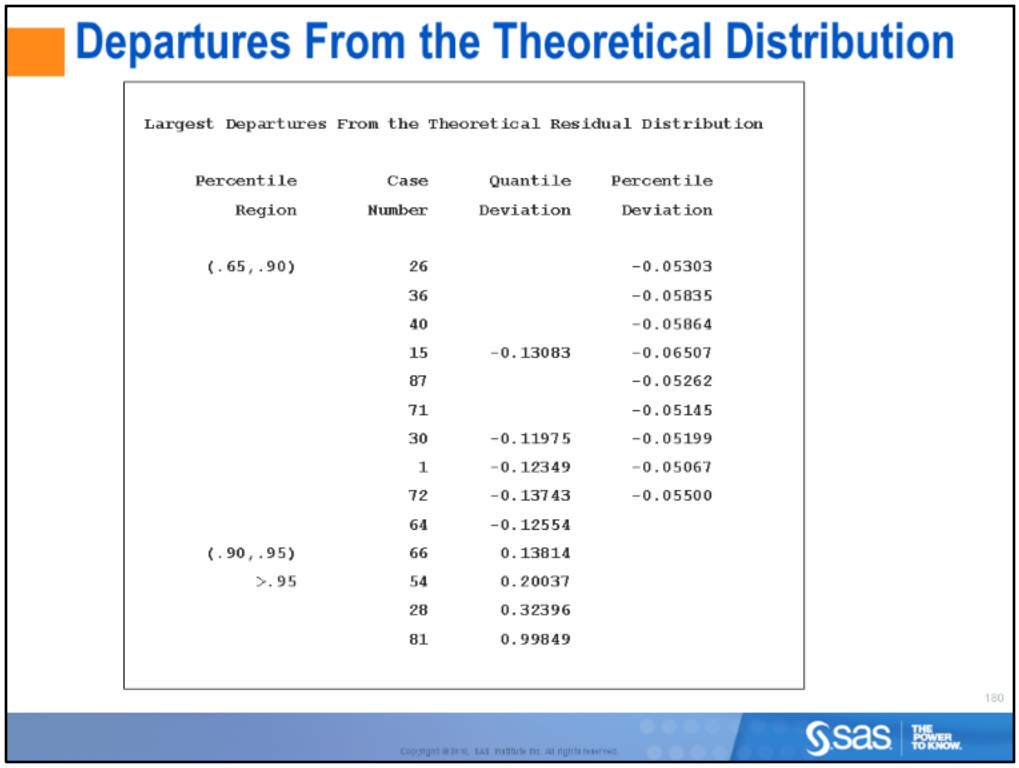
The major difference between SEM and regression Q-Q plots is that residual M-distances are always positive in SEM Q-Q plots. The reference distribution in SEM Q-Q plots is that of a chi-variate (instead of the normal distribution for regression analysis).

This Q-Q plot shows that the two outliers (Observations 28 and 81) are also much deviated from the theoretical distribution in terms of quantiles.



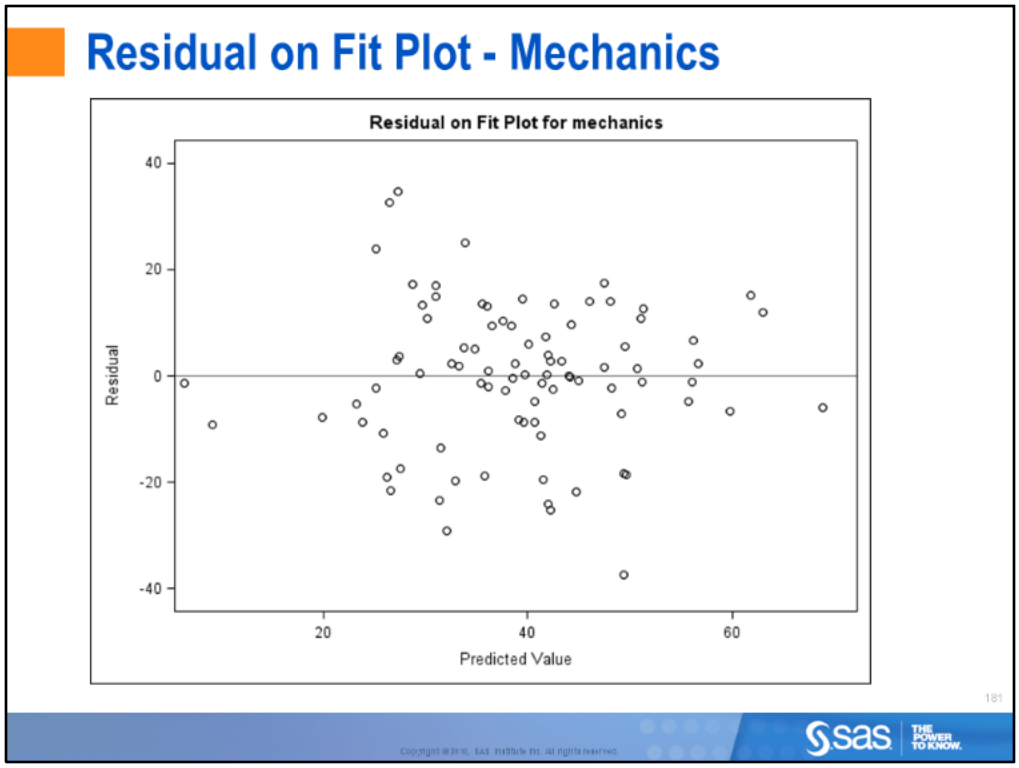
PROC CALIS can also plot the observed percentiles for the residual M-distances against the expected percentiles. This is known as the P-P plot in regression analysis. If the residuals are distributed exactly like that of the theoretical distribution, all observations should fall on the straight line with slope=1.

The P-P plot for the current example shows that several observations have large deviations in the middle of distribution.



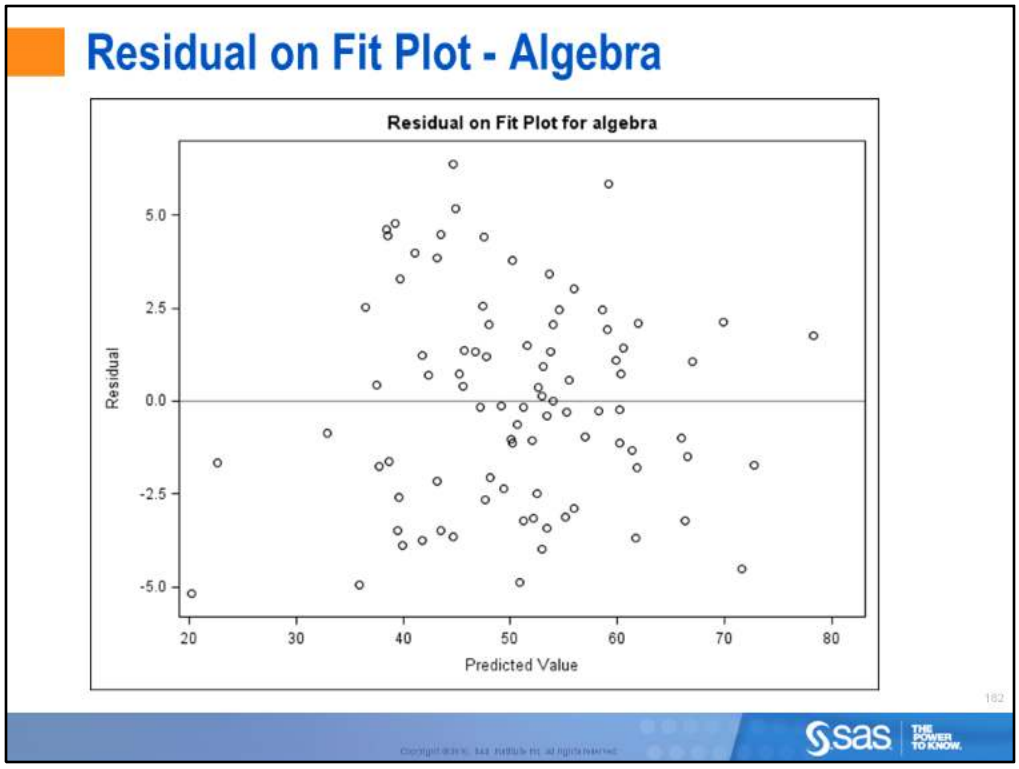
PROC CALIS produces numerical output that shows the departures of the observed residuals from the theoretical distributions.

This slide shows approximately 10 percent of the observations that have the largest departures in terms of quantile and percentile. Some observations, such as 15, 30, 1, and 72, have residual M-distances that are considered to be deviated from the theoretical distributions in terms of both quantile and percentile.

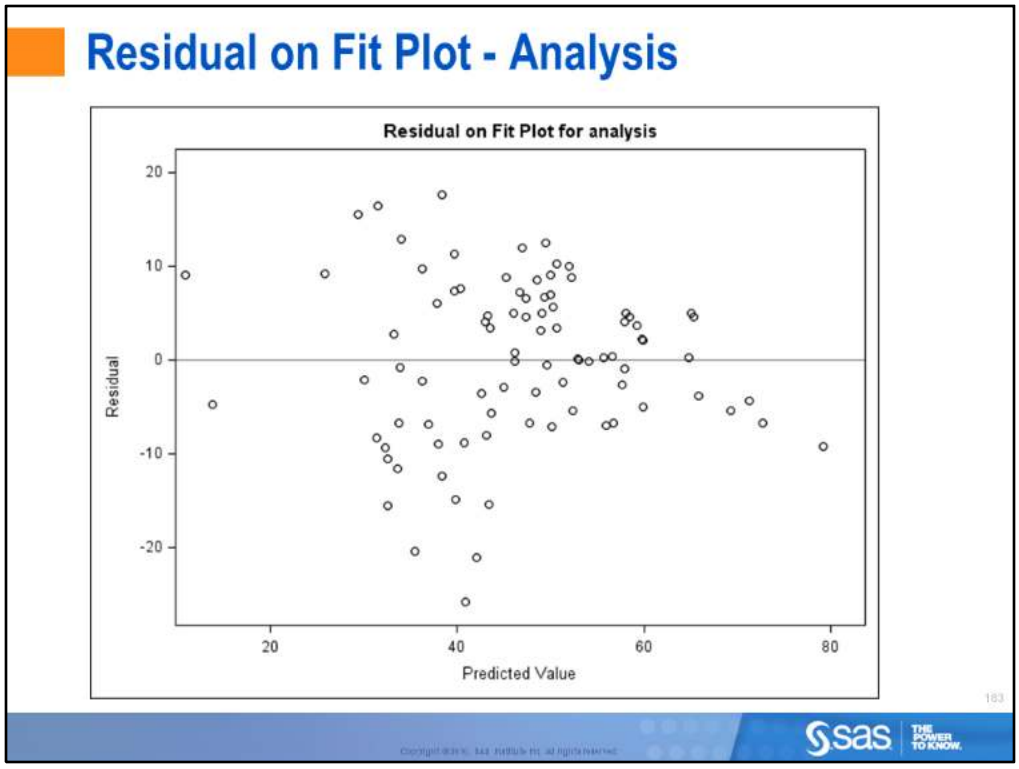


Another type of plots for residual diagnostics is the residual on fit plot. PROC CALIS can produce the residual on fit plots for all endogenous observed variables in the model.

Usually, in the residual on fit plots, you expect residuals are distributed randomly and homogenously along the predicted values if the linear model is true. For example, this slide shows that residuals of mechanics are distributed more-or-less “evenly” at all levels of predicted values.

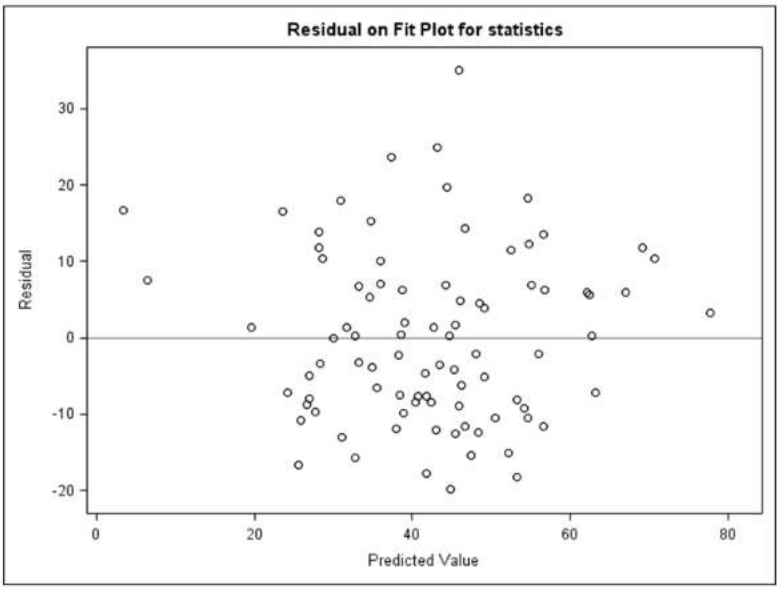


For algebra, the residuals also do not show systematic changes at different levels of the predicted values.

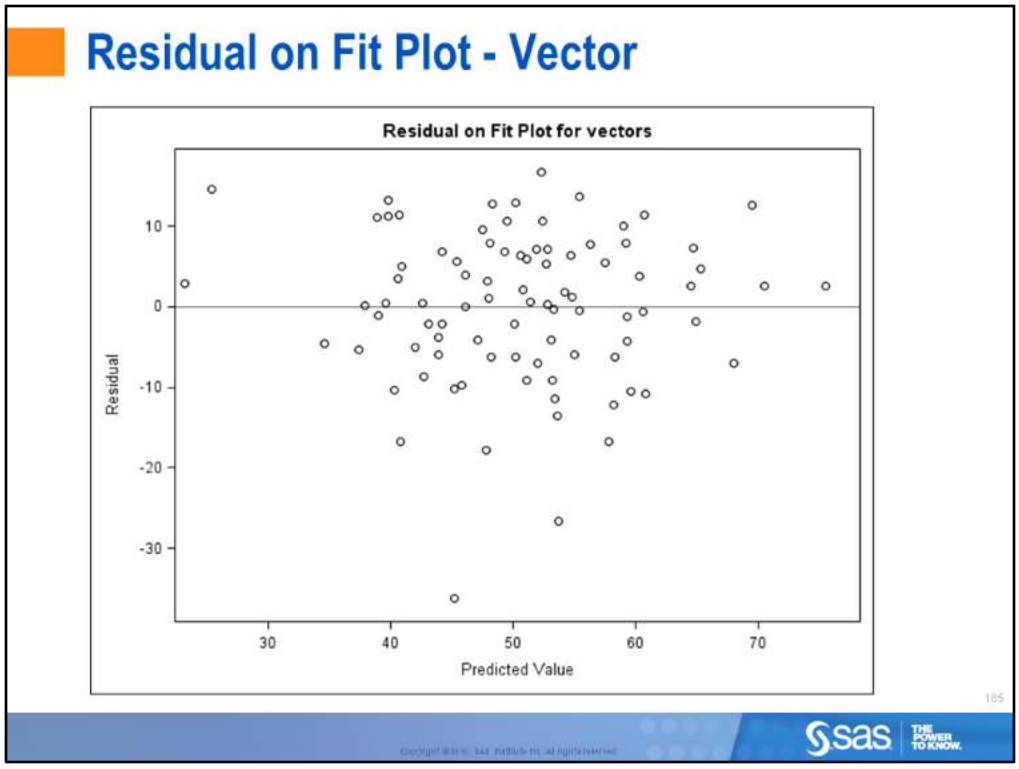


For analysis, however, it seems that the residual variances are getting smaller with higher predicted values.

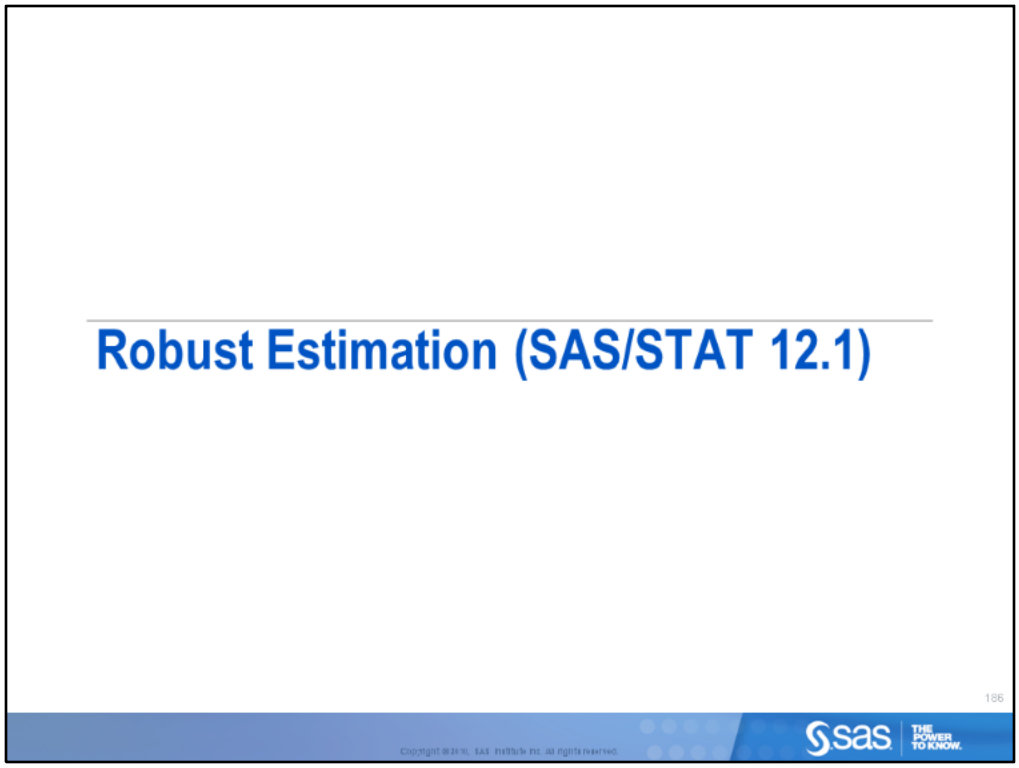
Residual on Fit Plot - Statistics



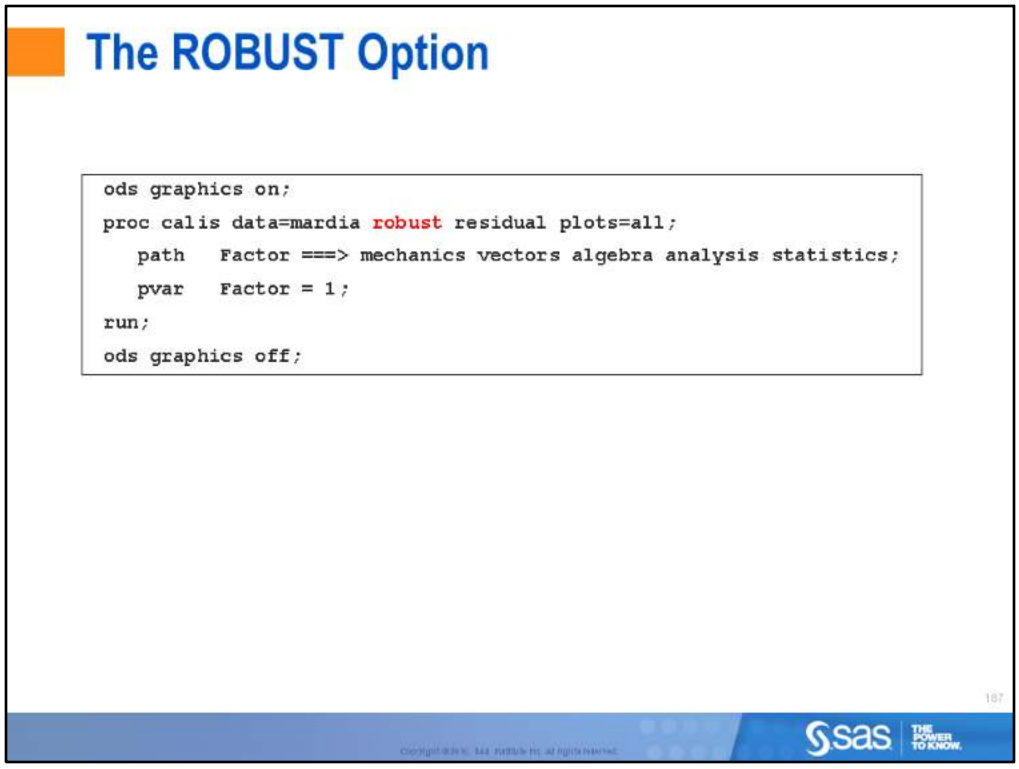
For statistics, the residual distribution looks okay.



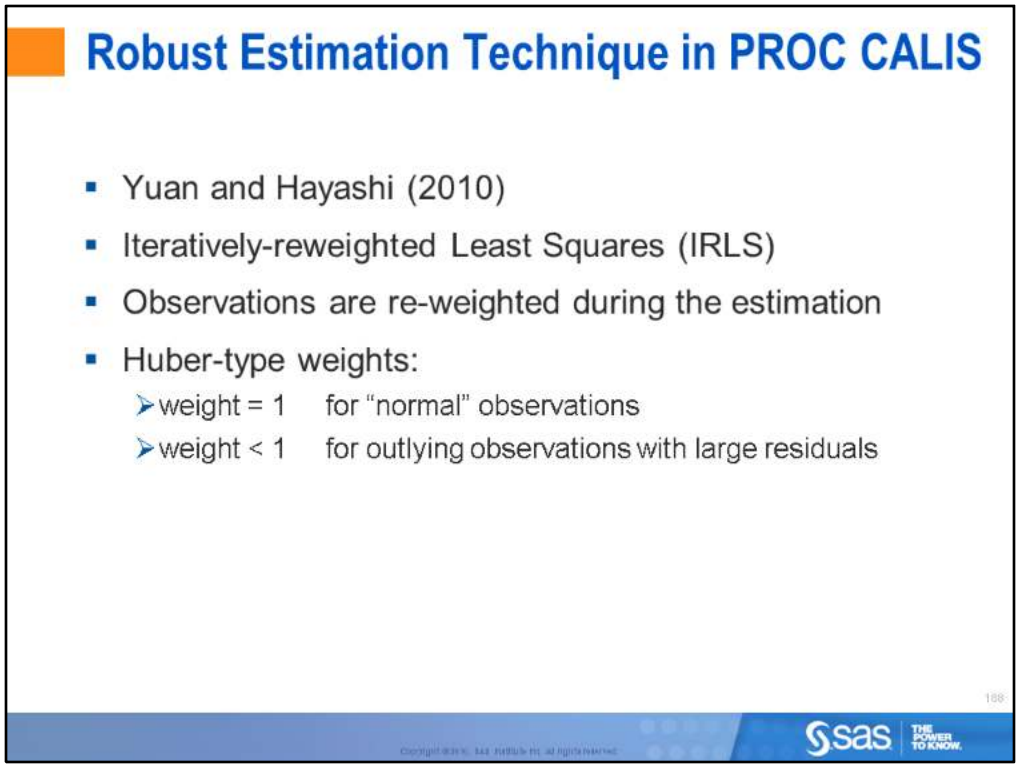
For vector, if you take away the two most extreme negative residuals, the residual distribution does look fine. But with the two extreme negative residuals, somehow the picture seems to suggest that residual variances are getting smaller at higher predicted values. So, although graphical displays are very useful as residual diagnostic tools, they are not always unambiguous --- sometimes we still need to use our judgments for proper interpretations.



The robust estimation of PROC CALIS can be viewed as an extension of the robust regression techniques. The difference is that PROC CALIS can handle a system of linear equations involving latent variables, instead of a single regression equation in regression analysis.



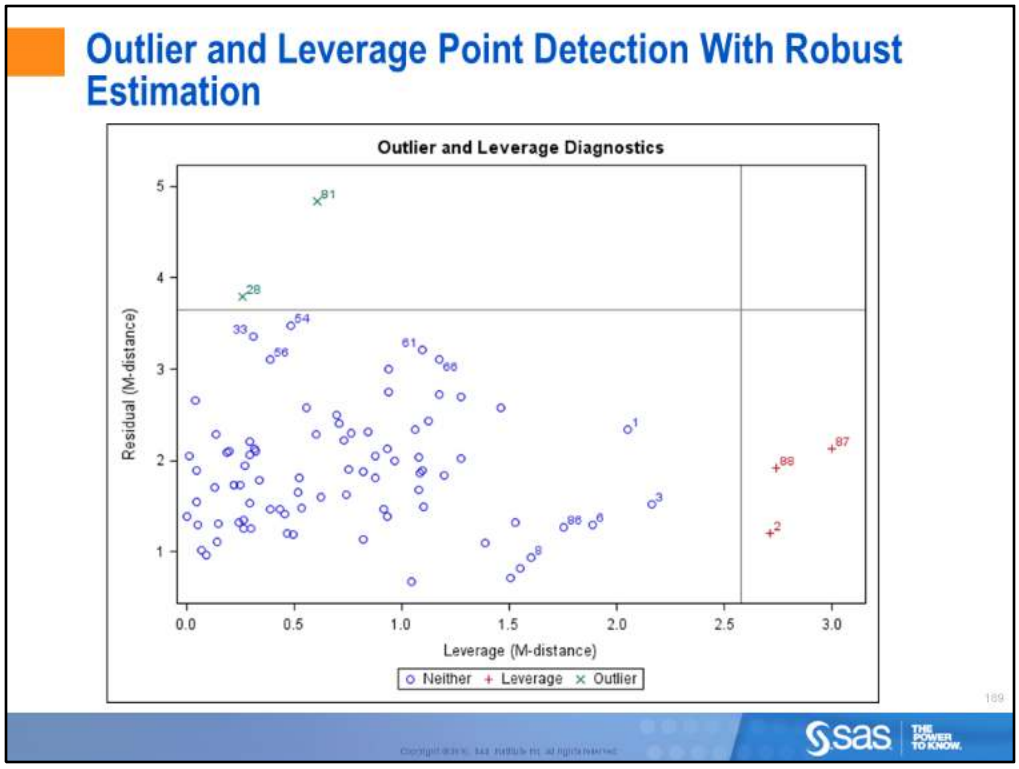
Using robust estimation with PROC CALIS is very easy. All you need to do is to use the ROBUST option in the PROC CALIS statement. The preceding example is now used to demonstrate the robust estimation.



The robust estimation technique of PROC CALIS is based on Yuan and Hayashi (2010) paper.

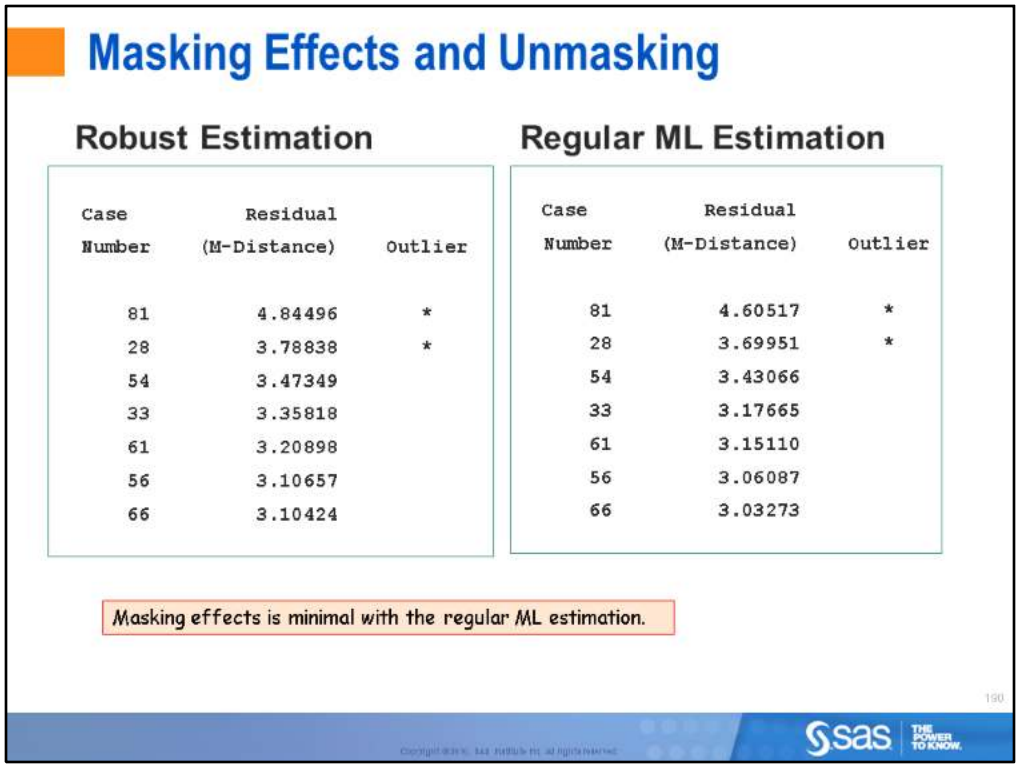
Essentially, observations are weighted and re-weighted during the estimation by the Huber-type weights. Non-outlying (normal) observations will have weights 1 and the outlying observations will have weights less than 1 during the estimation, which is carried out iteratively, as suggested by the name of algorithm---IRLS.

The idea of robust estimation is simple. Outlying observations are down-weighted so that they cannot skew the estimation.

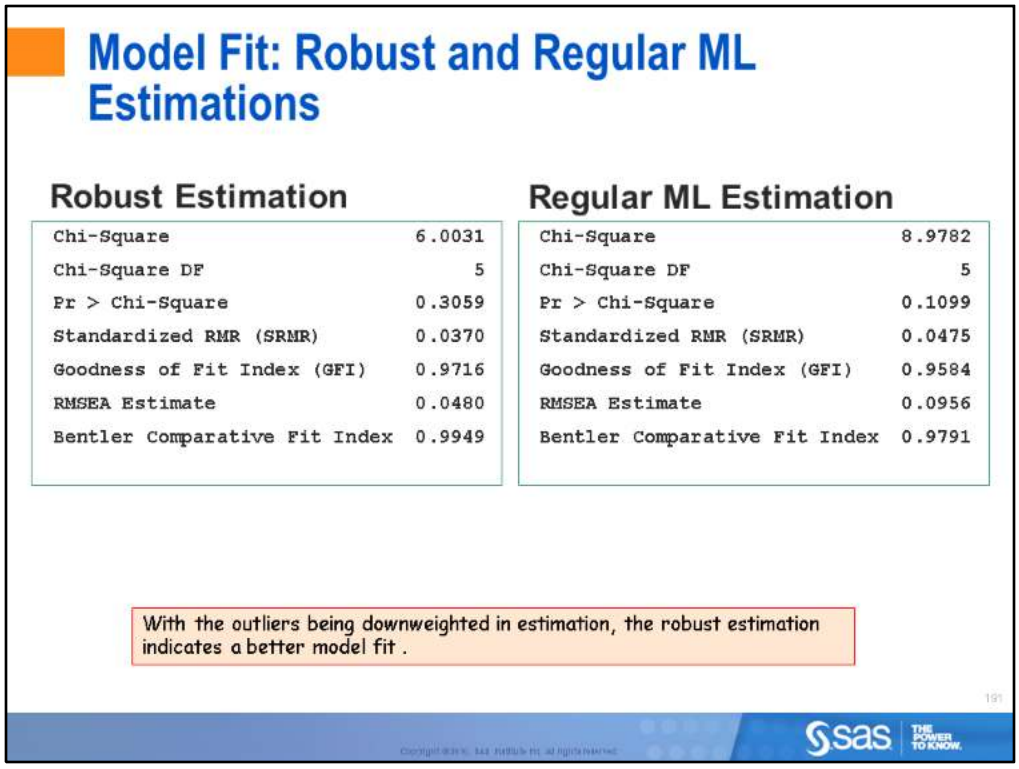


With the robust estimation, outlier and leverage points would be free from the so-called masking effect. The masking effect refers to the phenomenon that the presence of some prominent outliers might skew the estimation so that the less prominent outliers could not be identified. Because robust estimation has already downweighted the outliers during the estimation, residual diagnostics would not be skewed by the outliers and thus the masking effect could be unmasked.

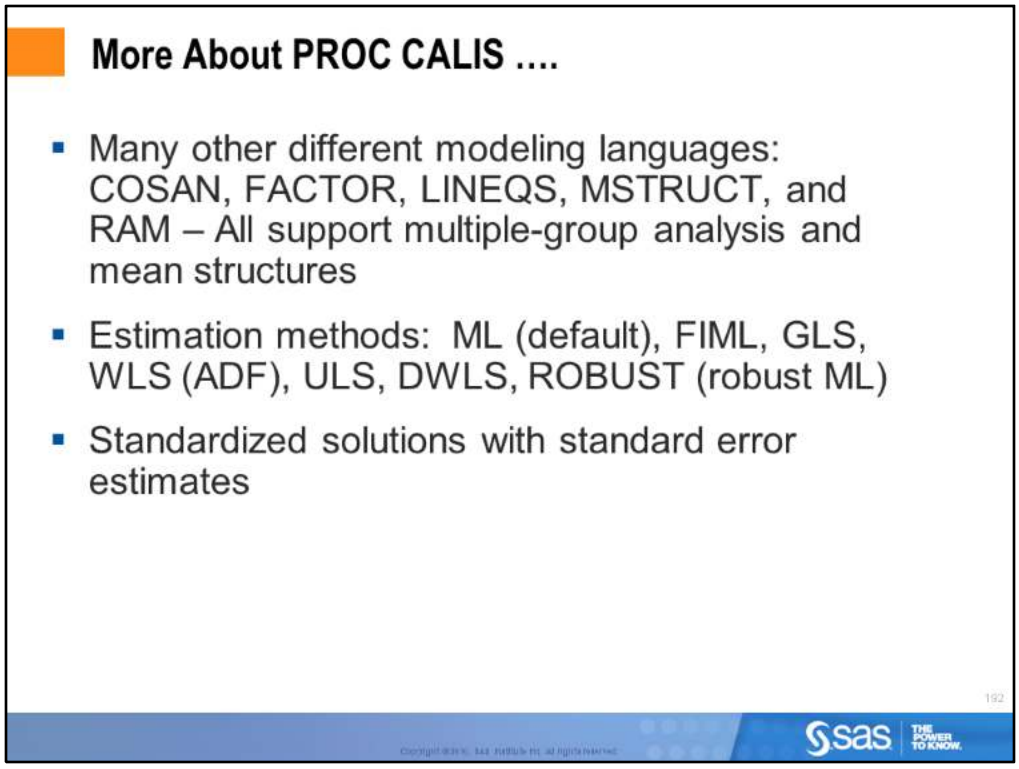
This slide show the residual against leverage plot for the identifications of outliers and leverage points. Because this picture is very similar to the corresponding picture with the regular ML estimation, we might conclude that no masking effect was present in the original ML estimation.



To further substantiate the previous assertion, you can look at the numerical results for outlier detections. The seven largest residual M-distances are more-or-less the same in the robust and the regular ML estimations, although the residual M-distances in the robust estimation are always larger due to the downweighting scheme in estimation. Also, both estimations identify exactly the same set of outliers.



For this particular example, the robust estimation yields a better model fit---the model fit chi-square is smaller in the robust estimation. The SRMR and the RMSEA are much smaller with the robust estimation. The GFI and Bentler CFI are slight better/higher with the robust estimation.



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, 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.

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., & 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.

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

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). *Multivariate Analysis*. London: Academic Press.

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.

References

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., & 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.

Yung, Y.-F. (2014). Creating Path Diagrams That Impress: A New Graphical Capability of the CALIS Procedure. http://support.sas.com/rnd/app/stat/papers/2014/yungpd2014.pdf

Yung, Y.-F. and Zhang, W. (2011). Making Use of Incomplete Observations in the Analysis of Structural Equation Models: The CALIS Procedure's Full Information Maximum Likelihood Method in SAS/STAT 9.3. In *Proceedings of the SAS Global Forum 2011 Conference*, Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings11/333-2011.pdf

Yuan, K.-H., & Hayashi, K. (2010). Fitting Data to Model: Structural Equation Modeling Diagnosis Using Two Scatter Plots. *Psychological Methods*, 15, 335–351.

Zhang, W. and Yung, Y.-F. (2011). A Tutorial on Structural Equation Modeling with Incomplete Observations: Multiple Imputation and FIML Methods Using SAS. http://support.sas.com/rnd/app/stat/papers/imps2011_FIML.pdf