SAS/STAT® 9.3 User's Guide

Introduction to Structural Equation Modeling with Latent Variables

(Chapter)

SAS® Documentation

This document is an individual chapter from *SAS/STAT® 9.3 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2011. *SAS/STAT® 9.3 User's Guide*. Cary, NC: SAS Institute Inc.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, July 2011

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Chapter 17

Introduction to Structural Equation Modeling with Latent Variables

Contents

Overview of Structural Equation Modeling with Latent Variables

Structural equation modeling includes analysis of covariance structures and mean structures, fitting systems of linear structural equations, factor analysis, and path analysis. In terms of the mathematical and statistical techniques involved, these various types of analyses are more or less interchangeable because the underlying methodology is based on analyzing the mean and covariance structures. However, the different analysis types emphasize different aspects of the analysis.

The analysis of covariance structures refers to the formulation of a model for the observed variances and covariances among a set of variables. The model expresses the variances and covariances as functions of some basic parameters. Similarly, the analysis of mean structures refers to the formulation of a model for the observed means. The model expresses the means as functions of some basic parameters. Usually, the covariance structures are of primary interest. However, sometimes the mean structures are analyzed simultaneously with the covariance structures in a model.

Corresponding to this kind of abstract formulation of mean and covariance structure analysis, PROC CALIS offers you two matrix-based modeling languages for specifying your model:

- MSTRUCT: a matrix-based model specification language that enables you to directly specify the parameters in the covariance and mean model matrices
- COSAN: a general matrix-based model specification language that enables you to specify a very wide class of mean and covariance structure models in terms of matrix expressions

Instead of focusing directly on the mean and covariance structures, other generic types of structural equation modeling emphasize more about the functional relationships among variables. Mean and covariance structures are still the means of these analyses, but they are usually implied from the structural relationships, rather than being directly specified as in the COSAN or MSTRUCT modeling languages.

In linear structural equations, the model is formulated as a system of equations that relates several random variables with assumptions about the variances and covariances of the random variables. The variables involved in the system of linear structural equations could be observed (manifest) or latent. Causal relationships between variables are hypothesized in the model.

When all observed variables in the model are hypothesized as indicator measures of underlying latent factors and the main interest is about studying the structural relations among the latent factors, it is a modeling scenario for factor-analysis or LISREL (Keesling 1972; Wiley 1973; Jöreskog 1973). PROC CALIS provides you two modeling languages that are closely related to this type of modeling scenario:

- FACTOR: a non-matrix-based model specification language that supports both exploratory and confirmatory factor analysis, including orthogonal and oblique factor rotations
- LISMOD: a matrix-based model specification language that enables you to specify the parameters in the LISREL model matrices

When causal relationships among observed and latent variables are freely hypothesized so that the observed variables are not limited to the roles of being measured indicators of latent factors, it is a modeling scenario for general path modeling (path analysis). In general path modeling, the model is formulated as a path diagram, in which arrows that connect variables represent variances, covariances, and path coefficients (effects). Depending on the way you represent the path diagram, you can use any of the following three different modeling languages in PROC CALIS:

- PATH: a non-matrix-based language that enables you to specify path-like relationships among variables
- RAM: a matrix-based language that enables you to specify the paths, variances, and covariance parameters in terms of the RAM model matrices (McArdle and McDonald 1984)
- LINEQS: an equation-based language that uses linear equations to specify functional or path relationships among variables (for example, the EQS model by Bentler 1995)

Although various types of analyses are put into distinct classes (with distinct modeling languages), with careful parameterization and model specification, it is possible to apply any of these modeling languages to the same analysis. For example, you can use the PATH modeling language to specify a confirmatory factor-analysis model, or you can use the LISMOD modeling language to specify a general path model. However, for some situations some modeling languages are easier to use than others. See the section "Which Modeling

Language?" on page 997 of Chapter 26, "The CALIS Procedure," for a detailed discussion of the modeling languages supported in PROC CALIS.

Loehlin (1987) provides an excellent introduction to latent variable models by using path diagrams and structural equations. A more advanced treatment of structural equation models with latent variables is given by Bollen (1989). Fuller (1987) provides a highly technical statistical treatment of measurement-error models.

This chapter illustrates applications of PROC CALIS, describes some of the main modeling features of PROC CALIS, and compares the CALIS procedure with the FACTOR and the SYSLIN procedures.

Testing Covariance Patterns

The most basic use of PROC CALIS is testing covariance patterns. Consider a repeated-measures experiment where individuals are tested for their motor skills at three different time points. No treatments are introduced between these tests. The three test scores are denoted as X_1, X_2, and X_3, respectively. These test scores are likely correlated because the same set of individuals has been used. More specifically, the researcher wants to test the following pattern of the population covariance matrix Σ:

$$\Sigma = \begin{pmatrix} \phi & \theta & \theta \\ \theta & \phi & \theta \\ \theta & \theta & \phi \end{pmatrix}$$

Because there are no treatments between the tests, this pattern assumes that the distribution of motor skills stays more or less the same over time, as represented by the same ϕ for the diagonal elements of Σ. The covariances between the test scores for motor skills also stay the same, as represented by the same θ for all the off-diagonal elements of Σ.

Suppose you summarize your data in a covariance matrix, which is stored in the following SAS data set:

```
data motor(type=cov);
   input _type_ $ _name_ $ x1 x2 x3;
   datalines;
COV     x1      3.566   1.342   1.114
COV     x2      1.342   4.012   1.056
COV     x3      1.114   1.056   3.776
N        .         36      36      36
;
```

The diagonal elements are somewhat close to each other but are not the same. The off-diagonal elements are also very close to each other but are not the same. Could these observed differences be due to chance? Given the sample covariance matrix, can you test the hypothesized patterned covariance matrix in the population?

Setting up this patterned covariance model in PROC CALIS is straightforward with the MSTRUCT modeling language:

```
proc calis data=motor;
   mstruct var  = x1-x3;
   matrix _cov_ = phi
                  theta phi
                  theta theta phi;
run;
```

In the VAR= option in the MSTRUCT statement, you specify that x1–x3 are the variables in the covariance matrix. Next, you specify the elements of the patterned covariance matrix in the MATRIX statement with the _COV_ keyword. Because the covariance matrix is symmetric, you need to specify only the lower triangular elements in the MATRIX statement. You use phi for the parameters of all diagonal elements and theta for the parameters of all off-diagonal elements. Matrix elements with the same parameter name are implicitly constrained to be equal. Hence, this is the patterned covariance matrix that you want to test. Some output results from PROC CALIS are shown in Figure 17.1.

Figure 17.1 Fit Summary

Fit Summary

Chi-Square	0.3656
Chi-Square DF	4
Pr > Chi-Square	0.9852

MSTRUCT _COV_ Matrix: Estimate/StdErr/t-value

	x1	x2	x3
x1	3.7847	1.1707	1.1707
	0.5701	0.5099	0.5099
	6.6383	2.2960	2.2960
	[phi]	[theta]	[theta]
x2	1.1707	3.7847	1.1707
	0.5099	0.5701	0.5099
	2.2960	6.6383	2.2960
	[theta]	[phi]	[theta]
x3	1.1707	1.1707	3.7847
	0.5099	0.5099	0.5701
	2.2960	2.2960	6.6383
	[theta]	[theta]	[phi]

First, PROC CALIS shows that the chi-square test for the model fit is 0.3656 (df=4, p=0.9852). Because the chi-square test is not significant, it supports the hypothesized patterned covariance model. Next, PROC CALIS shows the estimates in the covariance matrix under the hypothesized model. The estimates for the diagonal elements are all 3.7847, and the estimates for off-diagonal elements are all 1.1707. Estimates of standard errors and t values for these covariance and variance parameters are also shown.

The MSTRUCT modeling language in PROC CALIS enables you to test various kinds of covariance and mean patterns, including matrices with fixed or constrained values. For example, consider a population covariance model in which correlations among the motor test scores are hypothesized to be zero. In other words, the covariance pattern is:

$$\Sigma = \begin{pmatrix} \phi_1 & 0 & 0 \\ 0 & \phi_2 & 0 \\ 0 & 0 & \phi_3 \end{pmatrix}$$

Essentially, this diagonally-patterned covariance model means that the data are randomly and independently generated for x1–x3 under the multivariate normal distribution. Only the variances of the variables are parameters in the model, and the variables are not correlated at all.

You can use the MSTRUCT modeling language of PROC CALIS to fit this diagonally-patterned covariance matrix to the data for motor skills, as shown in the following statements:

```
proc calis data=motor;
   mstruct var  = x1-x3;
   matrix _cov_ = phi1
                  0.   phi2
                  0.     0.    phi3;
run;
```

Some of the output is shown in Figure 17.2.

Figure 17.2 Fit Summary: Testing Uncorrelatedness

Fit Summary

Chi-Square	9.2939
Chi-Square DF	3
Pr > Chi-Square	0.0256

MSTRUCT _COV_ Matrix: Estimate/StdErr/t-value

	x1	x2	x3
x1	3.5660 0.8524 4.1833 [phi1]	0	0
x2	0	4.0120 0.9591 4.1833 [phi2]	0
x3	0	0	3.7760 0.9026 4.1833 [phi3]

PROC CALIS shows that the chi-square test for the model fit is 9.2939 (df=3, p=0.0256). Because the chi-square test is significant, it does not support the patterned covariance model that postulates zero correlations among the variables. This conclusion is consistent with what is already known—the motor test scores should be somewhat correlated because they are measurements over time for the same group of individuals.

The output also shows the estimates of variances under the model. Each diagonal element of the covariance matrix has a distinct estimate because different parameters have been hypothesized under the patterned covariance model.

Testing Built-In Covariance Patterns in PROC CALIS

Some covariance patterns are well-known in multivariate statistics. For example, testing the diagonal pattern for a covariance matrix in the preceding section is a test of uncorrelatedness between the observed variables. Under the multivariate normal assumption, this test is also a test of independence between the observed variables. This test of independence is routinely applied in maximum likelihood factor analysis for testing the zero common factor hypothesis for the observed variables. For testing such a well-known covariance pattern, PROC CALIS provides an efficient way of specifying a model. With the COVPATTERN= option, you can invoke the built-in covariance patterns in PROC CALIS without the MSTRUCT model specifications, which could become laborious when the number of variables are large.

For example, to test the diagonal pattern (uncorrelatedness) of the motor skills, you can simply use the following specification:

```
proc calis data=motor covpattern=uncorr;
run;
```

The COVPATTERN=UNCORR option in the PROC CALIS statement invokes the diagonally patterned covariance matrix for the motor skills. PROC CALIS then generates the appropriate free parameters for this built-in covariance pattern. As a result, the MATRIX statement is not needed for specifying the free parameters, as it is if you use explicit MSTRUCT model specifications. Some of the output for using the COVPATTERN= option is shown in Figure 17.3.

Figure 17.3 Fit Summary: Testing Uncorrelatedness with the COVPATTERN= Option

Fit Summary

Chi-Square	8.8071
Chi-Square DF	3
Pr > Chi-Square	0.0320

Figure 17.3 *continued*

MSTRUCT _COV_ Matrix: Estimate/StdErr/t-value

	x1	x2	x3
x1	3.5660 0.8524 4.1833 [_varparm_1]	0	0
x2	0	4.0120 0.9591 4.1833 [_varparm_2]	0
x3	0	0	3.7760 0.9026 4.1833 [_varparm_3]

In the second table of Figure 17.3, the estimates of variances and their standard errors are the same as those shown in Figure 17.2. The only difference is that the parameter names (for example, _varparm_1) for the variances in Figure 17.3 are generated by PROC CALIS, instead of being specified as those in Figure 17.2.

However, the current chi-square test for the model fit is 8.8071 (df=3, p=0.0320), which is different from that in Figure 17.2 for testing the same covariance pattern. The reason is that the chi-square correction due to Bartlett (1950) has been applied automatically to the current built-in covariance pattern testing. Theoretically, this corrected chi-square value is more accurate. Therefore, in addition to its efficiency in specification, the built-in covariance pattern with the COVPATTERN= option offers an extra advantage in the automatic chi-square correction.

The COVPATTERN= option supports many other built-in covariance patterns. For details, see the COVPATTERN= option. See also the MEANPATTERN= option for testing built-in mean patterns.

Direct and Implied Covariance Patterns

You have seen how you can use PROC CALIS to test covariance patterns directly. Basically, you can specify the parameters in the covariance and mean matrices directly by using the MSTRUCT modeling language, which is invoked by the MSTRUCT statement. You can also use the COVPATTERN= option to test some built-in covariance patterns in PROC CALIS. To handle more complicated covariance and mean structures that are products of several model matrices, you can use the COSAN modeling language. The COSAN modeling language is too powerful to consider in this introductory chapter, but see the COSAN statement and the section "The COSAN Model" on page 1178 of Chapter 26, "The CALIS Procedure."

This section considers the fitting of patterned covariances matrix directly by using the MSTRUCT and the MATRIX statements or by the COVPATTERN= option. However, in most applications of structural equation modeling, the covariance patterns are not specified directly but are implied from the linear structural relationships among variables. The next few sections show how you can use other modeling languages in PROC CALIS to specify structural equation models with implied mean and covariance structures.

Regression with Measurement Errors

In this section, you start with a linear regression model and learn how the regression equation can be specified in PROC CALIS. The regression model is then extended to include measurement errors in the predictors and in the outcome variables. Problems with model identification are introduced.

Simple Linear Regression

Consider fitting a linear equation to two observed variables, Y and X. Simple linear regression uses the following model form:

$$Y = \alpha + \beta X + E_Y$$

The model makes the following assumption:

$$\mathrm{Cov}(X, E_Y) = 0$$

The parameters α and β are the intercept and regression coefficient, respectively, and E_Y is an error term. If the values of X are fixed, the values of E_Y are assumed to be independent and identically distributed realizations of a normally distributed random variable with mean zero and variance $\mathrm{Var}(E_Y)$. If X is a random variable, X and E_Y are assumed to have a bivariate normal distribution with zero correlation and variances $\mathrm{Var}(X)$ and $\mathrm{Var}(E_Y)$, respectively. Under either set of assumptions, the usual formulas hold for the estimates of the intercept and regression coefficient and their standard errors. (See Chapter 4, "Introduction to Regression Procedures.")

In the REG procedure, you can fit a simple linear regression model with a MODEL statement that lists only the names of the manifest variables, as shown in the following statements:

```
proc reg;
   model Y = X;
run;
```

You can also fit this model with PROC CALIS, but the syntax is different. You can specify the simple linear regression model in PROC CALIS by using the LINEQS modeling language, as shown in the following statements:

```
proc calis;
   lineqs
      Y = beta * X + Ey;
run;
```

LINEQS stands for "LINear EQuationS." You invoke the LINEQS modeling language by using the LINEQS statement in PROC CALIS. In the LINEQS statement, you specify the linear equations of your model. The LINEQS statement syntax is similar to the mathematical equation that you would write for the model. An obvious difference between the LINEQS and the PROC REG model specification is that in LINEQS you can name the parameter involved (for example, beta) and you also specify the error term explicitly. The additional syntax required by the LINEQS statement seems to make the model specification more time-consuming and cumbersome. However, this inconvenience is minor and is offset by the modeling flexibility

of the LINEQS modeling language (and of PROC CALIS, generally). As you proceed to more examples in this chapter, you will find the benefits of specifying parameter names for more complicated models with constraints. You will also find that specifying parameter names for unconstrained parameters is optional. Using parameter names in the current example is for the ease of reference in the current discussion.

You might wonder whether an intercept term is missing in the LINEQS statement and where you should put the intercept term if you want to specify it. The intercept term, which is considered as a mean structure parameter in the context of structural equation modeling, is usually omitted when statistical inferences can be drawn from analyzing the covariance structures alone. However, this does not mean that the regression equation has a default fixed-zero intercept in the LINEQS specification. Rather, it means only that the mean structures are saturated and are not estimated in the covariance structure model. Therefore, in the preceding LINEQS specification, the intercept term α is implicitly assumed in the model. It is not of primary interest and is not estimated.

However, if you want to estimate the intercept, you can specify it in the LINEQS equations, as shown in the following specification:

```
proc calis;
   lineqs
      Y = alpha * Intercept + beta * X + Ey;
run;
```

In this LINEQS statement, alpha represents the intercept parameter α and intercept represents an internal "variable" that has a fixed value of 1 for each observation. With this specification, an estimate of α is displayed in the PROC CALIS output results. However, estimation results for other parameters are the same as those from the specification without the intercept term. For this reason the intercept term is not specified in the examples of this section.

Errors-in-Variables Regression

For ordinary unconstrained regression models, there is no reason to use PROC CALIS instead of PROC REG. But suppose that the predictor variable X is a random variable that is contaminated by errors (especially measurement errors), and you want to estimate the linear relationship between the true, error-free scores. The following model takes this kind of measurement errors into account:

$$\begin{aligned} Y &= \alpha + \beta F_X + E_Y \\ X &= F_X + E_X \end{aligned}$$

The model assumes the following:

$$\mathrm{Cov}(F_X, E_Y) = \mathrm{Cov}(F_X, E_X) = \mathrm{Cov}(E_X, E_Y) = 0$$

There are two equations in the model. The first one is the so-called structural model, which describes the relationships between Y and the true score predictor F_X. This equation is your main interest. However, F_X is a latent variable that has not been observed. Instead, what you have observed for this predictor is X, which is the contaminated version of F_X with measurement error or other errors, denoted by E_X, added. This measurement process is described in the second equation, or the so-called measurement model. By

analyzing the structural and measurement models (or the two linear equations) simultaneously, you want to estimate the true score effect β.

The assumption that the error terms E_X and E_Y and the latent variable F_X are jointly uncorrelated is of critical importance in the model. This assumption must be justified on substantive grounds such as the physical properties of the measurement process. If this assumption is violated, the estimators might be severely biased and inconsistent.

You can express the current errors-in-variables model by the LINEQS modeling language as shown in the following statements:

```
proc calis;
   lineqs
      Y = beta * Fx + Ey,
      X = 1.   * Fx + Ex;
run;
```

In this specification, you need to specify only the equations involved without specifying the assumptions about the correlations among Fx, Ey, and Ex. In the LINEQS modeling language, you should always name latent factors with the 'F' or 'f' prefix (for example, Fx) and error terms with the 'E' or 'e' prefix (for example, Ey and Ex). Given this LINEQS notation, latent factors and error terms, by default, are uncorrelated in the model.

Consider an example of an errors-in-variables regression model. Fuller (1987, pp. 18–19) analyzes a data set from Voss (1969) that involves corn yields (Y) and available soil nitrogen (X) for which there is a prior estimate of the measurement error for soil nitrogen Var(E_X) of 57. The scientific question is: how does nitrogen affect corn yields? The linear prediction of corn yields by nitrogen should be based on a measure of nitrogen that is not contaminated with measurement error. Hence, the errors-in-variables model is applied. F_X in the model represents the "true" nitrogen measure, X represents the observed measure of nitrogen, which has a true score component F_X and an error component E_X. Given that the measurement error for soil nitrogen Var(E_X) is 57, you can specify the errors-in-variables regression model with the following statements in PROC CALIS:

```
data corn(type=cov);
   input _type_ $ _name_ $ y x;
   datalines;
cov    y    87.6727    .
cov    x    104.8818   304.8545
mean   .    97.4545    70.6364
n      .    11         11
;

proc calis data=corn;
   lineqs
      Y  = beta * Fx + Ey,
      X  = 1.   * Fx + Ex;
   variance
      Ex = 57.;
run;
```

In the VARIANCE statement, the variance of Ex (measurement error for X) is given as the constant value 57. PROC CALIS produces the estimates shown in Figure 17.4.

Figure 17.4 Errors-in-Variables Model for Corn Data

Linear Equations

y	=	0.4232*Fx	+	1.0000 Ey	
Std Err		0.1658 beta			
t Value		2.5520			
x	=	1.0000 Fx	+	1.0000 Ex	

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	Ex		57.00000		
Latent	Fx	_Add1	247.85450	136.33508	1.81798
Error	Ey	_Add2	43.29105	23.92488	1.80946

In Figure 17.4, the estimate of beta is 0.4232 with a standard error estimate of 0.1658. The t value is 2.552. It is significant at the 0.05 α-level when compared to the critical value of the standard normal variate (that is, the z table). Also shown in Figure 17.4 are the estimated variances of Fx, Ey, and their estimated standard errors. The names of these parameters have the prefix '_Add'. They are added by PROC CALIS as default parameters. By employing some conventional rules for setting default parameters, PROC CALIS makes your model specification much easier and concise. For example, you do not need to specify each error variance parameter manually if it is not constrained in the model. However, you can specify these parameters explicitly if you desire. Note that in Figure 17.4, the variance of Ex is shown to be 57 without a standard error estimate because it is a fixed constant in the model.

What if you did not model the measurement error in the predictor X? That is, what is the estimate of beta if you use ordinary regression of Y on X, as described by the equation in the section "Simple Linear Regression" on page 290? You can specify such a linear regression model easily by the LINEQS modeling language. Here, you specify this linear regression model as a special case of the errors-in-variables model. That is, you constrain the variance of measurement error E_x to 0 in the preceding LINEQS model specification to form the linear regression model, as shown in the following statements:

```
proc calis data=corn;
   lineqs
      Y  = beta * Fx + Ey,
      X  = 1. *   Fx + Ex;
   variance
      Ex = 0.;
run;
```

Fixing the variance of Ex to zero forces the equality of X and F_X in the measurement model so that this "new" errors-in-variables model is in fact an ordinary regression model. PROC CALIS produces the estimation results in Figure 17.5.

Figure 17.5 Ordinary Regression Model for Corn Data: Zero Measurement Error in X

Linear Equations

```
y       =    0.3440*Fx   +  1.0000 Ey
Std Err      0.1301 beta
t Value      2.6447
x       =    1.0000 Fx   +  1.0000 Ex
```

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	Ex		0		
Latent	Fx	_Add1	304.85450	136.33508	2.23607
Error	Ey	_Add2	51.58928	23.07143	2.23607

The estimate of beta is now 0.3440, which is an underestimate of the effect of nitrogen on corn yields given the presence of nonzero measurement error in X, where the estimate of beta is 0.4232.

Regression with Measurement Errors in X and Y

What if there are also measurement errors in the outcome variable Y? How can you write such an extended model? The following model would take measurement errors in both X and Y into account:

$$\begin{aligned} F_Y &= \alpha + \beta F_X + D_{Fy} \\ Y &= F_Y + E_Y \\ X &= F_X + E_X \end{aligned}$$

with the following assumption:

$$\begin{aligned} \mathrm{Cov}(F_X, D_{Fy}) &= \mathrm{Cov}(F_X, E_Y) = \mathrm{Cov}(F_X, E_X) = \mathrm{Cov}(F_Y, E_Y) \\ &= \mathrm{Cov}(F_Y, E_X) = \mathrm{Cov}(E_X, E_Y) = \mathrm{Cov}(E_X, D_{Fy}) \\ &= \mathrm{Cov}(E_Y, D_{Fy}) = 0 \end{aligned}$$

Again, the first equation, expressing the relationship between two latent true-score variables, defines the structural or causal model. The next two equations express the observed variables in terms of a true score plus error; these two equations define the measurement model. This is essentially the same form as the so-called LISREL model (Keesling 1972; Wiley 1973; Jöreskog 1973), which has been popularized by the LISREL program (Jöreskog and Sörbom 1988). Typically, there are several X and Y variables in a LISREL model. For the moment, however, the focus is on the current regression form in which there is only a single predictor and a single outcome variable. The LISREL model is considered in the section "Fitting LISREL Models by the LISMOD Modeling Language" on page 345.

With the intercept term left out for modeling, you can use the following statements for fitting the regression model with measurement errors in both X and Y:

```
proc calis data=corn;
   lineqs
      Fy  = beta * Fx + DFy,
      Y   = 1.   * Fy + Ey,
      X   = 1.   * Fx + Ex;
run;
```

Again, you do not need to specify the zero-correlation assumptions in the LINEQS model because they are set by default given the latent factors and errors in the LINEQS modeling language. When you run this model, PROC CALIS issues the following warning:

```
WARNING: Estimation problem not identified: More parameters to
         estimate ( 5 ) than the total number of mean and
         covariance elements ( 3 ).
```

The five parameters in the model include beta and the variances for the exogenous variables: Fx, DFy, Ey, and Ex. These variance parameters are treated as free parameters by default in PROC CALIS. You have five parameters to estimate, but the information for estimating these five parameters comes from the three unique elements in the sample covariance matrix for X and Y. Hence, your model is in the so-called underidentification situation. Model identification is discussed in more detail in the section "Model Identification" on page 297.

To make the current model identified, you can put constraints on some parameters. This reduces the number of independent parameters to estimate in the model. In the errors-in-variables model for the corn data, the variance of Ex (measurement error for X) is given as the constant value 57, which was obtained from a previous study. This could still be applied in the current model with measurement errors in both X and Y. In addition, if you are willing to accept the assumption that the structural equation model is (almost) deterministic, then the variance of Dfy could be set to 0. With these two parameter constraints, the current model is just-identified. That is, you can now estimate three free parameters from three distinct covariance elements in the data. The following statements show the LINEQS model specification for this just-identified model:

```
proc calis data=corn;
   lineqs
      Fy  = beta * Fx + Dfy,
      Y   = 1.   * Fy + Ey,
      X   = 1.   * Fx + Ex;
   variance
      Ex  = 57.,
      Dfy = 0.;
run;
```

Figure 17.6 shows the estimation results.

Figure 17.6 Regression Model With Measurement Errors in X and Y for Corn Data

```
                          Linear Equations

                  Fy      =    0.4232*Fx   +  1.0000 Dfy
                  Std Err      0.1658 beta
                  t Value      2.5520
                  y       =    1.0000 Fy   +  1.0000 Ey
                  x       =    1.0000 Fx   +  1.0000 Ex

              Estimates for Variances of Exogenous Variables

Variable                                                  Standard
Type          Variable    Parameter      Estimate          Error       t Value

Error         Ex                          57.00000
Disturbance   Dfy                               0
Latent        Fx          _Add1          247.85450      136.33508       1.81798
Error         Ey          _Add2           43.29105       23.92488       1.80946
```

In Figure 17.6, the estimate of beta is 0.4232, which is basically the same as the estimate for beta in the errors-in-variables model shown in Figure 17.4. The estimated variances for Fx and Ey match for the two models too. In fact, it is not difficult to show mathematically that the current constrained model with measurements errors in both Y and X is equivalent to the errors-in-variables model for the corn data. The numerical results merely confirm this fact.

It is important to emphasize that the equivalence shown here is not a general statement about the current model with measurement errors in X and Y and the errors-in-variables model. Essentially, the equivalence of the two models as applied to the corn data is due to those constraints imposed on the measurement error variances for DFy and Ex. The more important implication from these two analyses is that for the model with measurement errors in both X and Y, you need to set more parameter constraints to make the model identified. Some constraints might be substantively meaningful, while others might need strong or risky assumptions.

For example, setting the variance of Ex to 57 is substantively meaningful because it is based on a prior study. However, setting the variance of Dfy to 0 implies the acceptance of the deterministic structural model, which could be a rather risky assumption in most practical situations. It turns out that using these two constraints together for the model identification of the regression with measurement errors in both X and Y does not give you more substantively important information than what the errors-in-variables model has already given you (compare Figure 17.6 with Figure 17.4). Therefore, the set of identification constraints you use might be important in at least two aspects. First, it might lead to an identified model if you set them properly. Second, given that the model is identified, the meaningfulness of your model depends on how reasonable your identification constraints are.

The two identification constraints set on the regression model with measurement errors in both X and Y make the model identified. But they do not lead to model estimates that are more informative than that of the errors-in-variables regression. Some other sets of identification constraints, if available, might have been more informative. For example, if there were a prior study about the measurement error variance of corn yields (Y), a fixed constant for the variance of Ey could have been set, instead of the unrealistic zero

variance constraint of Dfy. This way the estimation results of the regression model with measurement errors in both X and Y would offer you something different from the errors-in-variables regression.

Setting identification constraints could be based on convention or other arguments. See the section "Illustration of Model Identification: Spleen Data" on page 298 for an example where model identification is attained by setting constant error variances for X and Y in the model. For the corn data, you have seen that fixing the error variance of the predictor variable led to model identification of the errors-in-variables model. In this case, prior knowledge about the measurement error variance is necessary. This necessity is partly due to the fact that each latent true score variable has only one observed variable as its indicator measure. When you have more measurement indicators for the same latent factor, fixing the measurement error variances to constants for model identification would not be necessary. This is the modeling scenario assumed by the LISREL model (see the section "Fitting LISREL Models by the LISMOD Modeling Language" on page 345), of which the confirmatory factor model is a special case. The confirmatory factor model is described and illustrated in the section "The FACTOR and RAM Modeling Languages" on page 320.

Model Identification

As discussed in the preceding section, if you try to fit the errors-in-variables model with measurement errors in both X and Y without applying certain constraints, the model is not identified and you cannot obtain unique estimates of the parameters. For example, the errors-in-variables model with measurement errors in both X and Y has five parameters (one coefficient β and four variances). The covariance matrix of the observed variables Y and X has only three elements that are free to vary, since $\mathrm{Cov}(Y,X)=\mathrm{Cov}(X,Y)$. Therefore, the covariance structure can be expressed as three equations in five unknown parameters. Since there are fewer equations than unknowns, there are many different sets of values for the parameters that provide a solution for the equations. Such a model is said to be underidentified.

If the number of parameters equals the number of free elements in the covariance matrix, then there might exist a unique set of parameter estimates that exactly reproduce the observed covariance matrix. In this case, the model is said to be just-identified or saturated.

If the number of parameters is less than the number of free elements in the covariance matrix, there might exist no set of parameter estimates that reproduces the observed covariance matrix exactly. In this case, the model is said to be overidentified. Various statistical criteria, such as maximum likelihood, can be used to choose parameter estimates that approximately reproduce the observed covariance matrix. If you use ML, FIML, GLS, or WLS estimation, PROC CALIS can perform a statistical test of the goodness of fit of the model under the certain statistical assumptions.

If the model is just-identified or overidentified, it is said to be identified. If you use ML, FIML, GLS, or WLS estimation for an identified model, PROC CALIS can compute approximate standard errors for the parameter estimates. For underidentified models, PROC CALIS obtains approximate standard errors by imposing additional constraints resulting from the use of a generalized inverse of the Hessian matrix.

You cannot guarantee that a model is identified simply by counting the parameters. For example, for any latent variable, you must specify a numeric value for the variance, or for some covariance involving the variable, or for a coefficient of an indicator variable. Otherwise, the scale of the latent variable is indeterminate, and the model is underidentified regardless of the number of parameters and the size of the covariance

matrix. As another example, an exploratory factor analysis with two or more common factors is always underidentified because you can rotate the common factors without affecting the fit of the model.

PROC CALIS can usually detect an underidentified model by computing the approximate covariance matrix of the parameter estimates and checking whether any estimate is linearly related to other estimates (Bollen 1989, pp. 248–250), in which case PROC CALIS displays equations showing the linear relationships among the estimates. Another way to obtain empirical evidence regarding the identification of a model is to run the analysis several times with different initial estimates to see whether the same final estimates are obtained. Bollen (1989) provides detailed discussions of conditions for identification in a variety of models.

Illustration of Model Identification: Spleen Data

When your model involves measurement errors in variables and you need to use latent true scores in the regression or structural equation, you might encounter some model identification problems in estimation if you do not put certain identification constraints in the model. An example is shown in the section "Regression with Measurement Errors in X and Y" on page 294 for the corn data. You "solved" the problem by assuming a deterministic model with perfect prediction in the structural model. However, this assumption could be very risky and does not lead to estimation results that are substantively different from the model with measurement error only in X.

This section shows how you can apply another set of constraints to make the measurement model with errors in both X and Y identified without assuming the deterministic structural model. First, the identification problem is illustrated here again in light of the PROC CALIS diagnostics.

The following example is inspired by Fuller (1987, pp. 40–41). The hypothetical data are counts of two types of cells in spleen samples: cells that form rosettes and nucleated cells. It is reasonable to assume that counts have a Poisson distribution; hence, the square roots of the counts should have a constant error variance of 0.25. You can use PROC CALIS to fit this regression model with measurement errors in X and Y to the data. (See the section "Regression with Measurement Errors in X and Y" on page 294 for model definitions.) However, before fitting this target model, it is illustrative to see what would happen if you do not assume the constant error variance.

The following statements show the LINEQS specification of an errors-in-variables regression model for the square roots of the counts without constraints on the parameters:

```
data spleen;
   input rosette nucleate;
   sqrtrose=sqrt(rosette);
   sqrtnucl=sqrt(nucleate);
   datalines;
4 62
5 87
5 117
6 142
8 212
9 120
12 254
13 179
```

```
15 125
19 182
28 301
51 357
;

proc calis data=spleen;
   lineqs factrose = beta * factnucl + disturb,
          sqrtrose =        factrose + err_rose,
          sqrtnucl =        factnucl + err_nucl;
   variance
          factnucl = v_factnucl,
          disturb  = v_disturb,
          err_rose = v_rose,
          err_nucl = v_nucl;
run;
```

This model is underidentified. You have five parameters to estimate in the model, but the number of distinct covariance elements is only three.

In the LINEQS statement, you specify the structural equation and then two measurement equations. In the structural equation, the variables factrose and factnucl are latent true scores for the corresponding measurements in sqrtrose and sqrtnucl, respectively. The structural equation represents the true variable relationship of interest. You name the regression coefficient parameter as beta and the error term as disturb in the structural model. (For structural equations, you can use names with prefix 'D' or 'd' to denote error terms.) The variance of factnucl and the variance of disturb are also parameters in the model. You name these variance parameters as v_factnucl and v_disturb in the VARIANCE statement. Therefore, you have three parameters in the structural equation.

In the measurement equations, the observed variables sqrtrose and sqrtnucl are specified as the sums of their corresponding true latent scores and error terms, respectively. The error variances are also parameters in the model. You name them as v_rose and v_nucl in the VARIANCE statement. Now, together with the three parameters in the structural equation, you have a total of five parameters in your model.

All variance specifications in the VARIANCE statement are actually optional in PROC CALIS. They are free parameters by default. In this example, it is useful to name these parameters so that explicit references to these parameters can be made in the following discussion.

PROC CALIS displays the following warning when you fit this underidentified model:

```
WARNING: Estimation problem not identified: More parameters to
         estimate ( 5 ) than the total number of mean and
         covariance elements ( 3 ).
```

In this warning, the three covariance elements refer to the sample variances of sqrtrose and sqrtnucl and their covariance. PROC CALIS diagnoses the parameter indeterminacy as follows:

```
   NOTE: Covariance matrix for the estimates is not full rank.
   NOTE: The variance of some parameter estimates is zero or
         some parameter estimates are linearly related to other
         parameter estimates as shown in the following equations:

v_rose       =      -0.147856   +       0.447307   *   v_disturb

v_nucl       =    -110.923690   -       0.374367   *   beta       +
                    10.353896   *   v_factnucl     +   1.536613   *   v_disturb
```

With the warning and the notes, you are now certain that the model is underidentified and you cannot interpret your parameter estimates meaningfully.

Now, to make the model identified, you set the error variances to 0.25 in the VARIANCE statement, as shown in the following specification:

```
proc calis data=spleen residual;
   lineqs factrose = beta * factnucl + disturb,
          sqrtrose =        factrose + err_rose,
          sqrtnucl =        factnucl + err_nucl;
   variance
          factnucl = v_factnucl,
          disturb  = v_disturb,
          err_rose = 0.25,
          err_nucl = 0.25;
run;
```

In the specification, you use the RESIDUAL option in the PROC CALIS statement to request the residual analysis. An annotated fit summary is shown in Figure 17.7.

Figure 17.7 Spleen Data: Annotated Fit Summary for the Just-Identified Model

```
                          Fit Summary

                Chi-Square              0.0000
                Chi-Square DF                0
                Pr > Chi-Square              .
```

You notice that the model fit chi-square is 0 and the corresponding degrees of freedom is also 0. This indicates that your model is "just" identified, or your model is saturated—you have three distinct elements in the sample covariance matrix for the estimation of three parameters in the model. In the PROC CALIS results, you no longer see the warning message about underidentification or any notes about linear dependence in parameters.

For just-identified or saturated models like the current case, you expect to get zero residuals in the covariance matrix, as shown in Figure 17.8:

Figure 17.8 Spleen Data: Residuals for the Just-identified Model

Raw Residual Matrix

	sqrtrose	sqrtnucl
sqrtrose	0.00000	0.00000
sqrtnucl	0.00000	0.00000

Residuals are the differences between the fitted covariance matrix and the sample covariance matrix. When the residuals are all zero, the fitted covariance matrix matches the sample covariance matrix perfectly (the parameter estimates reproduce the sample covariance matrix exactly).

You can now interpret the estimation results of this just-identified model, as shown in Figure 17.9:

Figure 17.9 Spleen Data: Parameter Estimated for the Just-Identified Model

Linear Equations

```
factrose =    0.3907*factnucl +  1.0000 disturb
Std Err       0.0771 beta
t Value       5.0692
sqrtrose =    1.0000 factrose +  1.0000 err_rose
sqrtnucl =    1.0000 factnucl +  1.0000 err_nucl
```

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Latent	factnucl	v_factnucl	10.50458	4.58577	2.29069
Disturbance	disturb	v_disturb	0.38153	0.28556	1.33607
Error	err_rose		0.25000		
	err_nucl		0.25000		

Notice that because the error variance parameters for variables err_rose and err_nucl are fixed constants in the model, there are no standard error estimates for them in Figure 17.9. For the current application, the estimation results of the just-identified model are those you would interpret and report. However, to completely illustrate model identification, an additional constraint is imposed to show an overidentified model. In the section "Regression with Measurement Errors in X and Y" on page 294, you impose a zero-variance constraint on the disturbance variable Dfy for the model identification. Would this constraint be necessary here for the spleen data too? The answer is no because with the two constraints on the variances of err_rose and err_nucl, the model has already been meaningfully specified and identified. Adding more constraints such as a zero variance for disturb would make the model overidentified unnecessarily. The following statements show the specification of such an overidentified model for the spleen data:

```
proc calis data=spleen residual;
   lineqs factrose = beta * factnucl + disturb,
          sqrtrose =        factrose + err_rose,
          sqrtnucl =        factnucl + err_nucl;
   variance
          factnucl = v_factnucl,
          disturb  = 0.,
          err_rose = 0.25,
          err_nucl = 0.25;
run;
```

An annotated fit summary table for the overidentified model is shown in Figure 17.10.

Figure 17.10 Spleen Data: Annotated Fit Summary for the Overidentified Model

Fit Summary

Chi-Square	5.2522
Chi-Square DF	1
Pr > Chi-Square	0.0219
Standardized RMSR (SRMSR)	0.0745
Adjusted GFI (AGFI)	0.1821
RMSEA Estimate	0.6217
Bentler Comparative Fit Index	0.6535

The chi-square is 5.2522 (df=1, p=0.0219). Overall, the model does not provide a good fit. The sample size is so small that the p-value of the chi-square test should not be taken to be accurate, but to get a small p-value with such a small sample indicates that it is possible that the model is seriously deficient.

This same conclusion can be drawn by looking at other fit indices in the table. In Figure 17.10, several fit indices are computed for the model. For example, the standardized root mean square residual (SRMSR) is 0.0745 and the adjusted goodness of fit (AGFI) is 0.1821. By conventions, a good model should have an SRMSR smaller than 0.05 and an AGFI larger than 0.90. The root mean square error of approximation (RMSEA) (Steiger and Lind 1980) is 0.6217, but an RMSEA below 0.05 is recommended for a good model fit (Browne and Cudeck 1993). The comparative fit index (CFI) is 0.6535, which is also low as compared to the acceptable level at 0.90.

When you fit an overidentified model, usually you do not find estimates that match the sample covariance matrix exactly. The discrepancies between the fitted covariance matrix and the sample covariance matrix are shown as residuals in the covariance matrix, as shown in Figure 17.11.

Figure 17.11 Spleen Data: Residuals for the Overidentified Model

Raw Residual Matrix

	sqrtrose	sqrtnucl
sqrtrose	0.28345	-0.11434
sqrtnucl	-0.11434	0.04613

As you can see in Figure 17.11, the residuals are nonzero. This indicates that the parameter estimates do not reproduce the sample covariance matrix exactly. For overidentified models, nonzero residuals would be the norm rather than exception, but the general goal is to find the "best" set of estimates so that the residuals are as small as possible.

The parameter estimates are shown in Figure 17.12.

Figure 17.12 Spleen Data: Parameter Estimated for the Overidentified Model

```
                          Linear Equations

              factrose =    0.4034*factnucl +  1.0000 disturb
              Std Err       0.0508 beta
              t Value       7.9439
              sqrtrose =    1.0000 factrose +  1.0000 err_rose
              sqrtnucl =    1.0000 factnucl +  1.0000 err_nucl


              Estimates for Variances of Exogenous Variables

Variable                                                  Standard
Type          Variable    Parameter       Estimate           Error       t Value

Latent        factnucl    v_factnucl      10.45846         4.56608       2.29047
Disturbance   disturb                            0
Error         err_rose                     0.25000
              err_nucl                     0.25000
```

The estimate of beta in this model is 0.4034. Given that the model fit is bad and the zero variance for the error term disturb is unreasonable, beta could have been overestimated in the current overidentified model, as compared with the just-identified model, where the estimate of beta is only 0.3907. In summary, both the fit summary and the estimation results indicate that the zero variance for disturb in the overidentified model for the spleen data has been imposed unreasonably.

The purpose of the current illustration is not that you should not consider an overidentified model for your data in general. Quite the opposite, in practical structural equation modeling it is usually the overidentified models that are of the paramount interest. You can test or gauge the model fit of overidentified models. Good overidentified models enable you to establish scientific theories that are precise and general. However, most fit indices are not meaningful when applied to just-identified saturated models. Also, even though you always get zero residuals for just-identified saturated models, those models usually are not precise enough to be a scientific theory.

The overidentified model for the spleen data highlights the importance of setting meaningful identification constraints. Whether your resulting model is just-identified or overidentified, it is recommended that you do the following:

- Give priorities to those identification constraints that are derived from prior studies, substantive grounds, or mathematical basis.
- Avoid making unnecessary identification constraints that might bias your model estimation.

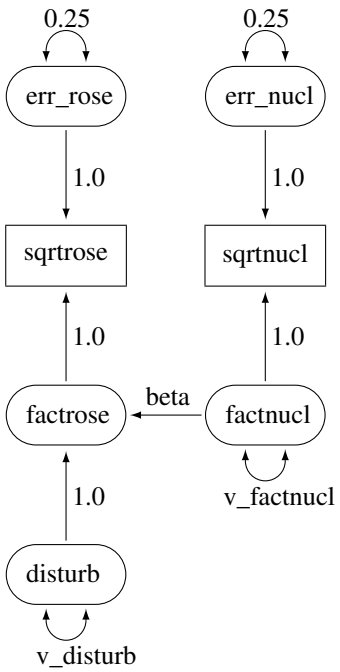
Path Diagrams and Path Analysis

Sections "Errors-in-Variables Regression" on page 291, "Regression with Measurement Errors in X and Y" on page 294, and "Illustration of Model Identification: Spleen Data" on page 298 show how you can specify models by means of equations in the LINEQS modeling language. This section shows you how to specify models that are represented by path diagrams. The PATH modeling language of PROC CALIS is the main tool for this purpose.

Complicated models are often easier to understand when they are expressed as path diagrams. One advantage of path diagrams over equations is that variances and covariances can be shown directly in the path diagram. Loehlin (1987) provides a detailed discussion of path diagrams. Another advantage is that the path diagram can be transcribed easily into the PATH modeling language supported by PROC CALIS.

A path diagram for the spleen data is shown in Figure 17.13. It explicitly shows all latent variables (including error terms) and variances of exogenous variables.

Figure 17.13 Path Diagram: Spleen Data



The path diagram shown in Figure 17.13 is essentially a graphical representation of the same just-identified model for the spleen data that is described in the section "Illustration of Model Identification: Spleen Data" on page 298. In path diagrams, it is customary to write the names of manifest or observed variables in rectangles and the names of latent variables in ovals. For example, sqrtrose and sqrtnucl are observed variables in the path diagram, while all others are latent variables.

The effects (the regression coefficients) in each equation are indicated by drawing arrows from the predictor variables to the outcome variable. For example, the path from factnucl to factrose is labeled with the

regression coefficient beta in the path diagram shown in Figure 17.13. Other paths are labeled with fixed coefficients (or effects) of 1.

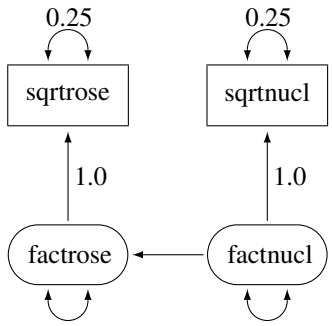
Variances of exogenous variables are drawn as double-headed arrows in Figure 17.13. For example, the variance of disturb is shown as a double-headed arrow pointing to the variable itself and is named v_disturb. Variances of the err_nucl and err_rose are also drawn as double-headed arrows but are labeled with fixed constants 0.25.

The path diagram shown in Figure 17.13 matches the features in the LINEQS model closely. For example, the error terms are depicted explicitly and their paths (regression coefficients) that connect to the associated endogenous variables are marked with fixed constants 1, reflecting the same specification in the equations of the LINEQS model. However, you can simplify the path diagram by using McArdle's RAM (reticular action model) notation (McArdle and McDonald 1984), as described in the following section.

A Simplified Path Diagram for the Spleen Data

The main simplification in the path diagram is to drop all the error terms in the model. Instead, error variances are treated as residual (or partial) variances for the endogenous variables in the model or path diagram. Hence, in the path diagrams for RAM models, error variances are also represented by double-headed arrows directly attached to the endogenous variables, which is the same way you represent variances for the exogenous variables. The RAM model convention leads to a simplified representation of the path diagram for the spleen data, as shown in Figure 17.14.

Figure 17.14 Simplified Path Diagram: Spleen



Another simplification done in Figure 17.14 is the omission of the parameter labeling in the path diagram. This simplification is not a part of the RAM notation. It is just a convention in PROC CALIS that you can omit the unconstrained parameter names without affecting the meaning of the model. Hence, the parameter names beta, v_disturb, and v_factnucl are no longer necessary in the simplified path diagram Figure 17.14. As you can see, this convention makes the task of model specification considerably simpler and easier.

The following statements show the specification of the simplified path diagram in Figure 17.14:

```
proc calis data=spleen;
   path
      sqrtrose <--- factrose   = 1.0,
      sqrtnucl <--- factnucl   = 1.0,
      factrose <--- factnucl   ;
   pvar
      sqrtrose = 0.25,      /* error variance for sqrtrose */
      sqrtnucl = 0.25,      /* error variance for sqrtnucl */
      factrose,             /* disturbance/error variance for factrose */
      factnucl;             /* variance of factnucl */
run;
```

The PATH statement invokes the PATH modeling language of PROC CALIS. In the PATH modeling language, each entry of specification corresponds to either a single- or double- headed arrow specification in the path diagram shown in Figure 17.14, as explained in the following:

- The PATH statement enables you to specify each of the single-headed arrows (paths) as path entries, which are separated by commas. You have three single-headed arrows in the path diagram and therefore you have three path entries in the PATH statement. The path entries "sqrtrose <--- factrose" and "sqrtnucl <--- factnucl" are followed by the constant 1, indicating fixed path coefficients. The path "factrose <--- factnucl" is also specified, but without giving a fixed value or a parameter name. By default, this path entry is associated with a free parameter for the effect or path coefficient.
- The PVAR statement enables you to specify each of the double-headed arrows with both heads pointing to the ***same*** variable, exogenous or endogenous. This type of arrows represents variances or error variances. You have four such double-headed arrows in the path diagram, and therefore there are four corresponding entries under the PVAR statement. Two of them are assigned with fixed constants (0.25), and the remaining two (error variance of factrose and variance of factnucl) are free variance parameters.
- The PCOV statement enables you to specify each of the double-headed arrows with its heads pointing to ***different*** variables, exogenous or endogenous. This type of arrows represents covariances between variables or their error terms. You do not have this type of double-headed arrows in the current path diagram, and therefore you do not need a PCOV statement for the corresponding model specification.

The estimation results are shown in Figure 17.15. Essentially, these are exactly the same estimation results as those that result from the LINEQS modeling language for the just-identified model in section "Illustration of Model Identification: Spleen Data" on page 298.

Figure 17.15 Spleen Data: RAM Model

PATH List

Path			Parameter	Estimate	Standard Error	t Value
sqrtrose	<---	factrose		1.00000		
sqrtnucl	<---	factnucl		1.00000		
factrose	<---	factnucl	_Parm1	0.39074	0.07708	5.06920

Figure 17.15 *continued*

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	sqrtrose		0.25000		
	sqrtnucl		0.25000		
	factrose	_Parm2	0.38153	0.28556	1.33607
Exogenous	factnucl	_Parm3	10.50458	4.58577	2.29069

Notice in Figure 17.15 that the path coefficient for path "factrose <--- factnucl" is given a parameter name _Parm1, which is generated automatically by PROC CALIS. This is the same beta parameter of the LINEQS model in the section "Illustration of Model Identification: Spleen Data" on page 298. Also, the variance parameters _Parm2 and _Parm3 in Figure 17.15 are the same v_disturb and v_factnucl parameters, respectively, in the preceding LINEQS model.

In PROC CALIS, using parameter names to specify free parameters is optional. Parameter names are generated for free parameters by default. Or, if you choose parameter names for your own convenience, you can do so without changing the model specification. For example, you can specify the preceding PATH model equivalently by adding the desired parameter names, as shown in the following statements:

```
proc calis data=spleen;
   path
      sqrtrose <--- factrose   = 1.0,
      sqrtnucl <--- factnucl   = 1.0,
      factrose <--- factnucl   = beta;
   pvar
      sqrtrose = 0.25,       /* error variance for sqrtrose */
      sqrtnucl = 0.25,       /* error variance for sqrtnucl */
      factrose = v_disturb,  /* disturbance/error variance for factrose */
      factnucl = v_factnucl; /* variance of factnucl */
run;
```

A path diagram provides you an easy and conceptual way to represent your model, while the PATH modeling language in PROC CALIS offers you an easy way to input your path diagram in a non-graphical fashion. This is especially useful for models with more complicated path structures. See the section "A Combined Measurement-Structural Model" on page 328 for a more elaborated example of the PATH model application.

The next section provides examples of the PATH model applied to classical test theory.

Some Measurement Models

In the section "Regression with Measurement Errors in X and Y" on page 294, outcome variables and predictor variables are assumed to have been measured with errors. In order to study the true relationships among the true scores variables, models for measurement errors are also incorporated into the estimation. The context of applications is that of regression or econometric analysis.

In the social and behavioral sciences, the same kind of model is developed in the context of test theory or item construction for measuring cognitive abilities, personality traits, or other latent variables. This kind of modeling is better-known as measurement models or confirmatory factor analysis (these two terms are interchangeable) in the psychometric field. Usually, applications in the social and behavioral sciences involve a much larger number of observed variables. This section considers some of these measurement or confirmatory factor-analytic models. For illustration purposes, only a handful of variables are used in the examples. Applications that use the PATH modeling language in PROC CALIS are described.

H4: Full Measurement Model for Lord Data

Psychometric test theory involves many kinds of models that relate scores on psychological and educational tests to latent variables that represent intelligence or various underlying abilities. The following example uses data on four vocabulary tests from Lord (1957). Tests W and X have 15 items each and are administered with very liberal time limits. Tests Y and Z have 75 items and are administered under time pressure. The covariance matrix is read by the following DATA step:

```
data lord(type=cov);
   input _type_ $ _name_ $ W X Y Z;
   datalines;
n    . 649       .       .       .
cov W 86.3979    .       .       .
cov X 57.7751 86.2632    .       .
cov Y 56.8651 59.3177 97.2850    .
cov Z 58.8986 59.6683 73.8201 97.8192
;
```

The psychometric model of interest states that W and X are determined by a single common factor F_1, and Y and Z are determined by a single common factor F_2. The two common factors are expected to have a positive correlation, and it is desired to estimate this correlation. It is convenient to assume that the common factors have unit variance, so their correlation will be equal to their covariance. The error terms for all the manifest variables are assumed to be uncorrelated with each other and with the common factors. The model equations are

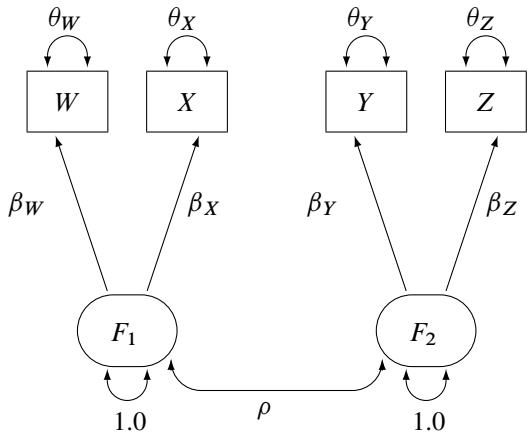
$$
\begin{aligned}
W &= \beta_W F_1 + E_W \\
X &= \beta_X F_1 + E_X \\
Y &= \beta_Y F_2 + E_Y \\
Z &= \beta_Z F_2 + E_Z
\end{aligned}
$$

with the following assumptions:

$$
\begin{aligned}
\text{Var}(F_1) &= \text{Var}(F_2) = 1 \\
\text{Cov}(F_1, F_2) &= \rho \\
\text{Cov}(E_W, E_X) &= \text{Cov}(E_W, E_Y) = \text{Cov}(E_W, E_Z) = \text{Cov}(E_X, E_Y) \\
&= \text{Cov}(E_X, E_Z) = \text{Cov}(E_Y, E_Z) = \text{Cov}(E_W, F_1) \\
&= \text{Cov}(E_W, F_2) = \text{Cov}(E_X, F_1) = \text{Cov}(E_X, F_2) \\
&= \text{Cov}(E_Y, F_1) = \text{Cov}(E_Y, F_2) = \text{Cov}(E_Z, F_1) \\
&= \text{Cov}(E_Z, F_2) = 0
\end{aligned}
$$

The corresponding path diagram is shown in Figure 17.16.

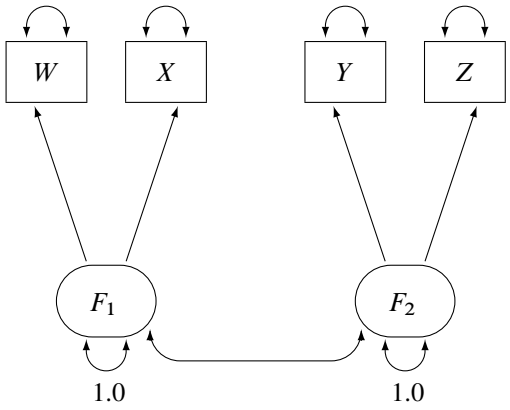
Figure 17.16 Path Diagram: Lord Data



In Figure 17.16, error terms are not explicitly represented, but error variances for the observed variables are represented by double-headed arrows that point to the variables. The error variance parameters in the model are labeled with θ_W, θ_X, θ_Y, and θ_Z, respectively, for the four observed variables. In the terminology of confirmatory factor analysis, these four variables are called indicators of the corresponding latent factors F_1 and F_2.

Figure 17.16 represents the model equations clearly. It includes all the variables and the parameters in the diagram. However, sometimes researchers represent the same model with a simplified path diagram in which unconstrained parameters are not labeled, as shown in Figure 17.17.

Figure 17.17 Simplified Path Diagram: Lord Data



This simplified representation is also compatible with the PATH modeling language of PROC CALIS. In fact, this might be an easier starting point for modelers. With the following rules, the conversion from the path diagram to the PATH model specification is very straightforward:

- Each single-headed arrow in the path diagram is specified in the PATH statement.
- Each double-headed arrow that points to a single variable is specified in the PVAR statement.
- Each double-headed arrow that points to two distinct variables is specified in the PCOV statement.

Hence, you can convert the simplified path diagram in Figure 17.17 easily to the following PATH model specification:

```
proc calis data=lord;
   path
      W <--- F1,
      X <--- F1,
      Y <--- F2,
      Z <--- F2;
   pvar
      F1 = 1.0,
      F2 = 1.0,
      W X Y Z;
   pcov
      F1 F2;
run;
```

In this specification, you do not need to specify the parameter names. However, you do need to specify fixed values specified in the path diagram. For example, the variances of F1 and F2 are both fixed at 1 in the PVAR statement.

These fixed variances are applied solely for the purpose of model identification. Because F1 and F2 are latent variables and their scales are arbitrary, fixing their scales are necessary for model identification. Beyond

these two identification constraints, none of the parameters in the model is constrained. Therefore, this is referred to as the "full" measurement model for the Lord data.

An annotated fit summary is displayed in Figure 17.18.

Figure 17.18 Fit Summary, H4: Full Model With Two Factors for Lord Data

Fit Summary

Chi-Square	0.7030
Chi-Square DF	1
Pr > Chi-Square	0.4018
Standardized RMSR (SRMSR)	0.0030
Adjusted GFI (AGFI)	0.9946
RMSEA Estimate	0.0000
Bentler Comparative Fit Index	1.0000

The chi-square value is 0.7030 (df=1, p=0.4018). This indicates that you cannot reject the hypothesized model. The standardized root mean square error (SRMSR) is 0.003, which is much smaller than the conventional 0.05 value for accepting good model fit. Similarly, the RMSEA value is virtually zero, indicating an excellent fit. The adjusted GFI (AGFI) and Bentler comparative fit index are close to 1, which also indicate an excellent model fit.

The estimation results are displayed in Figure 17.19.

Figure 17.19 Estimation Results, H4: Full Model With Two Factors for Lord Data

PATH List

--------Path--------			Parameter	Estimate	Standard Error	t Value
W	<---	F1	_Parm1	7.50066	0.32339	23.19390
X	<---	F1	_Parm2	7.70266	0.32063	24.02354
Y	<---	F2	_Parm3	8.50947	0.32694	26.02730
Z	<---	F2	_Parm4	8.67505	0.32560	26.64301

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	F1		1.00000		
	F2		1.00000		
Error	W	_Parm5	30.13796	2.47037	12.19979
	X	_Parm6	26.93217	2.43065	11.08021
	Y	_Parm7	24.87396	2.35986	10.54044
	Z	_Parm8	22.56264	2.35028	9.60000

Figure 17.19 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
F1	F2	_Parm9	0.89855	0.01865	48.17998

All estimates are shown with estimates of standard errors in Figure 17.19. They are all statistically significant, supporting nontrivial relationships between the observed variables and the latent factors. Notice that each free parameter in the model has been named automatically in the output. For example, the path coefficient from F1 to W is named _Parm1.

Two results in Figure 17.19 are particularly interesting. First, in the table for estimates of the path coefficients, _Parm1 and _Parm2 values form one cluster, while _Parm3 and _Parm4 values from another cluster. This seems to indicate that the effects from F1 on the indicators W and X could have been the same in the population and the effects from F2 on the indicators Y and Z could also have been the same in the population. Another interesting result is the estimate for the correlation between F1 and F2 (both were set to have variance 1). The correlation estimate (_Parm9 in the Figure 17.19) is 0.8986. It is so close to 1 that you wonder whether F1 and F2 could have been the same factor in the population. These estimation results can be used to motivate additional analyses for testing the suggested constrained models against new data sets. However, for illustration purposes, the same data set is used to demonstrate the additional model fitting in the subsequent sections.

In an analysis of these data by Jöreskog and Sörbom (1979, pp. 54–56) (see also Loehlin 1987, pp. 84–87), four hypotheses are considered:

H_1: One-factor model with parallel tests
$\rho = 1$
$\beta_W = \beta_X$ and $\mathrm{Var}(E_W) = \mathrm{Var}(E_X)$
$\beta_Y = \beta_Z$ and $\mathrm{Var}(E_Y) = \mathrm{Var}(E_Z)$

H_2: Two-factor model with parallel tests
$\beta_W = \beta_X$ and $\mathrm{Var}(E_W) = \mathrm{Var}(E_X)$
$\beta_Y = \beta_Z$ and $\mathrm{Var}(E_Y) = \mathrm{Var}(E_Z)$

H_3: Congeneric model: One factor without assuming parallel tests
$\rho = 1$

H_4: Full model: Two factors without assuming parallel tests

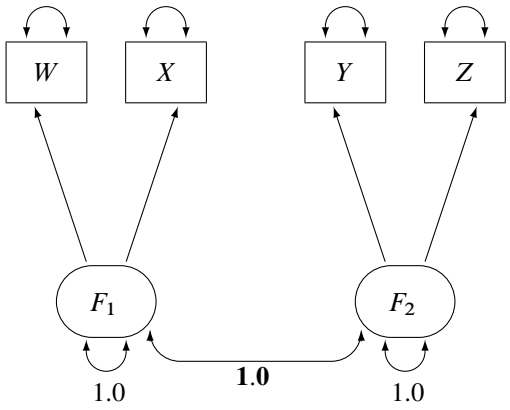
These hypotheses are ordered such that the latter models are less constrained. The hypothesis H_4 is the full model that has been considered in this section. The hypothesis H_3 specifies that there is really just one common factor instead of two; in the terminology of test theory, W, X, Y, and Z are said to be congeneric. Setting the correlation ρ between F1 and F2 to 1 makes the two factors indistinguishable. The hypothesis

H_2 specifies that W and X have the same true scores and have equal error variance; such tests are said to be parallel. The hypothesis H_2 also requires Y and Z to be parallel. Because ρ is not constrained to 1 in H_2, two factors are assumed for this model. The hypothesis H_1 says that W and X are parallel tests, Y and Z are parallel tests, and all four tests are congeneric (with ρ also set to 1).

H3: Congeneric (One-Factor) Model for Lord Data

The path diagram for this congeneric (one-factor) model is shown in Figure 17.20.

Figure 17.20 H3: Congeneric (One-Factor) Model for Lord Data



The only difference between the current path diagram in Figure 17.20 for the congeneric (one-factor) model and the preceding path diagram in Figure 17.17 for the full (two-factor) model is that the double-headed path that connects F1 and F2 is fixed to 1 in the current path diagram. Accordingly, you need to modify only slightly the preceding PROC CALIS specification to form the new model specification, as shown in the following statements:

```
proc calis data=lord;
   path
      W <--- F1,
      X <--- F1,
      Y <--- F2,
      Z <--- F2;
   pvar
      F1 = 1.0,
      F2 = 1.0,
      W X Y Z;
   pcov
      F1 F2 = 1.0;
run;
```

This specification sets the covariance between F1 and F2 to 1.0 in the PCOV statement. An annotated fit summary is displayed in Figure 17.21.

Figure 17.21 Fit Summary, H3: Congeneric (One-Factor) Model for Lord Data

Fit Summary

Chi-Square	36.2095
Chi-Square DF	2
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0277
Adjusted GFI (AGFI)	0.8570
RMSEA Estimate	0.1625
Bentler Comparative Fit Index	0.9766

The chi-square value is 36.2095 ($df = 2$, $p < 0.0001$). This indicates that you can reject the hypothesized model at the 0.01 α-level. The standardized root mean square error (SRMSR) is 0.0277, which indicates a good fit. Bentler's comparative fit index is 0.9766, which is also a good model fit. However, the adjusted GFI (AGFI) is 0.8570, which is not very impressive. Also, the RMSEA value is 0.1625, which is too large to be an acceptable model. Therefore, the congeneric model might not be the one you want to use.

The estimation results are displayed in Figure 17.22. Because the model does not fit well, the corresponding estimation results are not interpreted.

Figure 17.22 Estimation Results, H3: Congeneric (One-Factor) Model for Lord Data

PATH List

--------Path--------			Parameter	Estimate	Standard Error	t Value
W	<---	F1	_Parm1	7.10470	0.32177	22.08012
X	<---	F1	_Parm2	7.26908	0.31826	22.83973
Y	<---	F2	_Parm3	8.37344	0.32542	25.73143
Z	<---	F2	_Parm4	8.51060	0.32409	26.26002

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	F1		1.00000		
	F2		1.00000		
Error	W	_Parm5	35.92111	2.41467	14.87619
	X	_Parm6	33.42373	2.31037	14.46684
	Y	_Parm7	27.17043	2.24621	12.09613
	Z	_Parm8	25.38887	2.20837	11.49664

Figure 17.22 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Estimate	Standard Error	t Value
F1	F2	1.00000		

Perhaps a more natural way to specify the model under hypothesis H_3 is to use only one factor in the PATH model, as shown in the following statements:

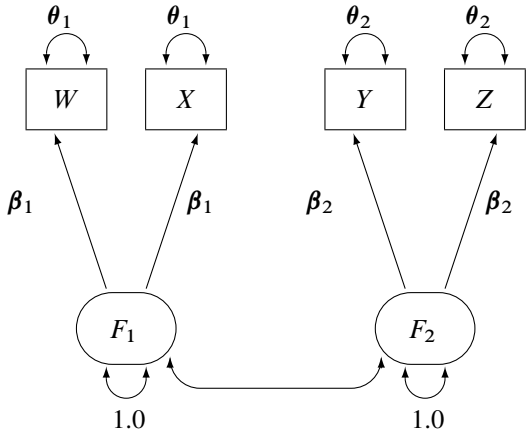
```
proc calis data=lord;
   path
      W <--- F1,
      X <--- F1,
      Y <--- F1,
      Z <--- F1;
   pvar
      F1 = 1.0,
      W X Y Z;
run;
```

This produces essentially the same results as the specification with two factors that have perfect correlation.

H2: Two-Factor Model with Parallel Tests for Lord Data

The path diagram for the two-factor model with parallel tests is shown in Figure 17.23.

Figure 17.23 H2: Two-Factor Model with Parallel Tests for Lord Data



The hypothesis H_2 requires that variables or tests under each factor are "interchangeable." In terms of the measurement model, several pairs of parameters must be constrained to have equal estimates. That is, under the parallel-test model W and X should have the same effect or path coefficient β_1 from their common factor

F1, and they should also have the same measurement error variance θ_1. Similarly, Y and Z should have the same effect or path coefficient β_2 from their common factor F2, and they should also have the same measurement error variance θ_2. These constraints are labeled in Figure 17.23.

You can impose each of these equality constraints by giving the same name for the parameters involved in the PATH model specification. The following statements specify the path diagram in Figure 17.23:

```
proc calis data=lord;
   path
      W <--- F1   = beta1,
      X <--- F1   = beta1,
      Y <--- F2   = beta2,
      Z <--- F2   = beta2;
   pvar
      F1  = 1.0,
      F2  = 1.0,
      W X = 2 * theta1,
      Y Z = 2 * theta2;
   pcov
      F1 F2;
run;
```

Note that the specification **2*theta1** in the PVAR statement means that theta1 is specified twice for the error variances of the two variables W and X. Similarly for the specification **2*theta2**. An annotated fit summary is displayed in Figure 17.24.

Figure 17.24 Fit Summary, H2: Two-Factor Model with Parallel Tests for Lord Data

Fit Summary

Chi-Square	1.9335
Chi-Square DF	5
Pr > Chi-Square	0.8583
Standardized RMSR (SRMSR)	0.0076
Adjusted GFI (AGFI)	0.9970
RMSEA Estimate	0.0000
Bentler Comparative Fit Index	1.0000

The chi-square value is 1.9335 (df=5, p=0.8583). This indicates that you cannot reject the hypothesized model H2. The standardized root mean square error (SRMSR) is 0.0076, which indicates a very good fit. Bentler's comparative fit index is 1.0000. The adjusted GFI (AGFI) is 0.9970, and the RMSEA is close to zero. All results indicate that this is a good model for the data.

The estimation results are displayed in Figure 17.25.

Figure 17.25 Estimation Results, H2: Two-Factor Model with Parallel Tests for Lord Data

PATH List

Path			Parameter	Estimate	Standard Error	t Value
W	<---	F1	beta1	7.60099	0.26844	28.31580
X	<---	F1	beta1	7.60099	0.26844	28.31580
Y	<---	F2	beta2	8.59186	0.27967	30.72146
Z	<---	F2	beta2	8.59186	0.27967	30.72146

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	F1		1.00000		
	F2		1.00000		
Error	W	theta1	28.55545	1.58641	18.00000
	X	theta1	28.55545	1.58641	18.00000
	Y	theta2	23.73200	1.31844	18.00000
	Z	theta2	23.73200	1.31844	18.00000

Covariances Among Exogenous Variables

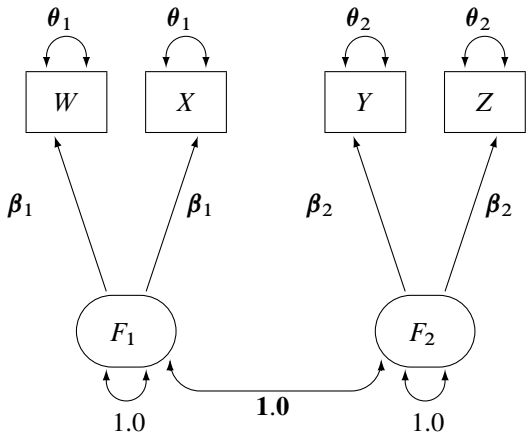
Var1	Var2	Parameter	Estimate	Standard Error	t Value
F1	F2	_Parm1	0.89864	0.01865	48.18011

Notice that because you explicitly specify the parameter names for the path coefficients (that is, beta1 and beta2), they are used in the output shown in Figure 17.25. The correlation between F1 and F2 is 0.8987, which is a very high correlation that suggests F1 and F2 might have been the same factor in the population. The next section sets this value to one so that the current model becomes a one-factor model with parallel tests.

H1: One-Factor Model with Parallel Tests for Lord Data

The path diagram for the one-factor model with parallel tests is shown in Figure 17.26.

Figure 17.26 H1: One-Factor Model with Parallel Tests for Lord Data



The hypothesis H_1 differs from H_2 in that F1 and F2 have a perfect correlation in H_1. This is indicated by the fixed value 1.0 for the double-headed path that connects F1 and F2 in Figure 17.26. Again, you need only minimal modification of the preceding specification for H_2 to specify the path diagram in Figure 17.26, as shown in the following statements:

```
proc calis data=lord;
   path
      W <--- F1   = beta1,
      X <--- F1   = beta1,
      Y <--- F2   = beta2,
      Z <--- F2   = beta2;
   pvar
      F1  = 1.0,
      F2  = 1.0,
      W X = 2 * theta1,
      Y Z = 2 * theta2;
   pcov
      F1 F2 = 1.0;
run;
```

The only modification of the preceding specification is in the PCOV statement, where you put a constant 1 for the covariance between F1 and F2. An annotated fit summary is displayed in Figure 17.27.

Figure 17.27 Fit Summary, H1: One-Factor Model with Parallel Tests for Lord Data

Fit Summary

Chi-Square	37.3337
Chi-Square DF	6
Pr > Chi-Square	<.0001
Standardized RMSR (SRMSR)	0.0286
Adjusted GFI (AGFI)	0.9509
RMSEA Estimate	0.0898
Bentler Comparative Fit Index	0.9785

The chi-square value is 37.3337 (df=6, p<0.0001). This indicates that you can reject the hypothesized model H1 at the 0.01 α-level. The standardized root mean square error (SRMSR) is 0.0286, the adjusted GFI (AGFI) is 0.9509, and Bentler's comparative fit index is 0.9785. All these indicate good model fit. However, the RMSEA is 0.0898, which does not support an acceptable model for the data.

The estimation results are displayed in Figure 17.28.

Figure 17.28 Estimation Results, H1: One-Factor Model with Parallel Tests for Lord Data

PATH List

--------Path--------			Parameter	Estimate	Standard Error	t Value
W	<---	F1	beta1	7.18623	0.26598	27.01802
X	<---	F1	beta1	7.18623	0.26598	27.01802
Y	<---	F2	beta2	8.44198	0.28000	30.14943
Z	<---	F2	beta2	8.44198	0.28000	30.14943

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	F1		1.00000		
	F2		1.00000		
Error	W	theta1	34.68865	1.64634	21.07010
	X	theta1	34.68865	1.64634	21.07010
	Y	theta2	26.28513	1.39955	18.78119
	Z	theta2	26.28513	1.39955	18.78119

Covariances Among Exogenous Variables

Var1	Var2	Estimate	Standard Error	t Value
F1	F2	1.00000		

The goodness-of-fit tests for the four hypotheses are summarized in the following table.

Hypothesis	Number of Parameters	χ^2	Degrees of Freedom	p-value	$\hat{\rho}$
H_1	4	37.33	6	< .0001	1.0
H_2	5	1.93	5	0.8583	0.8986
H_3	8	36.21	2	< .0001	1.0
H_4	9	0.70	1	0.4018	0.8986

Recall that the estimates of ρ for H_2 and H_4 are almost identical, about 0.90, indicating that the speeded and unspeeded tests are measuring almost the same latent variable. However, when ρ was set to 1 in H_1 and H_3 (both one-factor models), both hypotheses were rejected. Hypotheses H_2 and H_4 (both two-factor models) seem to be consistent with the data. Since H_2 is obtained by adding four constraints (for the requirement of parallel tests) to H_4 (the full model), you can test H_2 versus H_4 by computing the differences of the chi-square statistics and their degrees of freedom, yielding a chi-square of 1.23 with four degrees of freedom, which is obviously not significant. In a sense, the chi-square difference test means that representing the data by H_2 would not be significantly worse than representing the data by H_4. In addition, because H_2 offers a more precise description of the data (with the assumption of parallel tests) than H_4, it should be chosen because of its simplicity. In conclusion, the two-factor model with parallel tests provides the best explanation of the data.

The FACTOR and RAM Modeling Languages

In the section "Some Measurement Models" on page 307, you use the path diagram to represent the measurement models for data with cognitive tests and then you use the PATH modeling language to specify the model in PROC CALIS. You could have used other types of modeling languages for specifying the same model. In this section, the FACTOR and the RAM modeling languages are illustrated.

Specifying the Full Measurement Model (H4) by the FACTOR Modeling Language: Lord Data

The measurement models described in the section "Some Measurement Models" on page 307 are also known as confirmatory factor models. PROC CALIS has a specific modeling language, called FACTOR, for confirmatory factor models. You can use this modeling language for both exploratory and confirmatory factor analysis.

For example, the full measurement model H4 in the section "H4: Full Measurement Model for Lord Data" on page 308 can be specified equivalently by the FACTOR modeling language with the following statements:

```
proc calis data=lord;
   factor
      F1  --->  W  X,
      F2  --->  Y  Z;
   pvar
      F1 = 1.0,
      F2 = 1.0,
      W X Y Z;
   cov
      F1 F2;
run;
```

In the specification, you use the FACTOR statement to invoke the FACTOR modeling language. In the FACTOR statement, you specify the paths from the latent factors to the measurement indicators. For example, F1 has two paths to its indicators, W and X. Similarly, F2 has two paths to its indicators, Y and Z. Next, you use the PVAR statement to specify the variances, which is exactly the same way you use the PATH model specification in the section "H4: Full Measurement Model for Lord Data" on page 308. Lastly, you use the COV statement to specify the covariance among the factors, much like you use the PCOV statement to specify the same covariance in the PATH model specification.

Given the same confirmatory factor model, there is a major difference between the paths specified by the PATH statement and the paths specified by the FACTOR statement. In the FACTOR statement, each path must start with a latent factor followed by a right arrow and the variable list. In the PATH statement, each path can start or end with an observed or latent variable, and the direction of the arrow can be left or right.

The fit summary table for the FACTOR model is shown in Figure 17.29:

Figure 17.29 Fit Summary of the Full Confirmatory Factor Model for Lord Data

Fit Summary

Chi-Square	0.7030
Chi-Square DF	1
Pr > Chi-Square	0.4018
Standardized RMSR (SRMSR)	0.0030
Adjusted GFI (AGFI)	0.9946
RMSEA Estimate	0.0000
Bentler Comparative Fit Index	1.0000

This is exactly the same fit summary as shown in Figure 17.18, which is for the PATH model specification. Therefore, this confirms that the same model is being fit by the FACTOR model specification.

The estimation results are shown in Figure 17.30.

Figure 17.30 Estimation Results of Full Confirmatory Factor Model for Lord Data

Factor Loading Matrix: Estimate/StdErr/t-value

	F1	F2
W	7.5007 0.3234 23.1939 [_Parm1]	0
X	7.7027 0.3206 24.0235 [_Parm2]	0
Y	0	8.5095 0.3269 26.0273 [_Parm3]
Z	0	8.6751 0.3256 26.6430 [_Parm4]

Factor Covariance Matrix: Estimate/StdErr/t-value

	F1	F2
F1	1.0000	0.8986 0.0186 48.1800 [_Parm9]
F2	0.8986 0.0186 48.1800 [_Parm9]	1.0000

Error Variances

Variable	Parameter	Estimate	Standard Error	t Value
W	_Parm5	30.13796	2.47037	12.19979
X	_Parm6	26.93217	2.43065	11.08021
Y	_Parm7	24.87396	2.35986	10.54044
Z	_Parm8	22.56264	2.35028	9.60000

Again, these are the same estimates as those shown in Figure 17.19, which is for the PATH model specification. The FACTOR results displayed in Figure 17.30 are arranged differently though. No paths are shown there. The relationships between the latent factors and its indicators are shown in matrix form. The factor variance and covariances are also shown in matrix form.

Specifying the Parallel Tests Model (H2) by the FACTOR Modeling Language: Lord Data

In the section "H2: Two-Factor Model with Parallel Tests for Lord Data" on page 315, you fit a two-factor model with parallel tests for the Lord data by the PATH modeling language in PROC CALIS. Some paths and error variance are constrained under the PATH model. You can also specify this parallel tests model by the FACTOR modeling language, as shown in the following statements:

```
proc calis data=lord;
   factor
      F1  --->  W  X    = 2 * beta1,
      F2  --->  Y  Z    = 2 * beta2;
   pvar
      F1  = 1.0,
      F2  = 1.0,
      W X = 2 * theta1,
      Y Z = 2 * theta2;
   cov
      F1 F2;
run;
```

In this specification, you specify some parameters explicitly. You apply the parameter beta1 to the loadings of both W and X on F1. This means that F1 has the same amount of effect on W and X. Similarly, you apply the parameter beta2 to the loadings of Y and Z on F2. The constraints on the error variances for W, X, Y, and Z in this FACTOR model specification are done in the same way as in the PATH model specification in the section "H2: Two-Factor Model with Parallel Tests for Lord Data" on page 315.

The fit summary table for this parallel tests model is shown in Figure 17.31.

Figure 17.31 Fit Summary of the Confirmatory Factor Model with Parallel Tests for Lord Data

Fit Summary

Chi-Square	1.9335
Chi-Square DF	5
Pr > Chi-Square	0.8583
Standardized RMSR (SRMSR)	0.0076
Adjusted GFI (AGFI)	0.9970
RMSEA Estimate	0.0000
Bentler Comparative Fit Index	1.0000

All the fit indices shown in Figure 17.31 for the FACTOR model match the corresponding PATH model results displayed in Figure 17.24. All the estimation results in Figure 17.32 for the FACTOR model are the same as those for the corresponding PATH model in Figure 17.25.

Figure 17.32 Estimation Results of the Confirmatory Factor Model with Parallel Tests for Lord Data

Factor Loading Matrix: Estimate/StdErr/t-value

	F1	F2
W	7.6010 0.2684 28.3158 [beta1]	0
X	7.6010 0.2684 28.3158 [beta1]	0
Y	0	8.5919 0.2797 30.7215 [beta2]
Z	0	8.5919 0.2797 30.7215 [beta2]

Factor Covariance Matrix: Estimate/StdErr/t-value

	F1	F2
F1	1.0000	0.8986 0.0187 48.1801 [_Parm1]
F2	0.8986 0.0187 48.1801 [_Parm1]	1.0000

Error Variances

Variable	Parameter	Estimate	Standard Error	t Value
W	theta1	28.55545	1.58641	18.00000
X	theta1	28.55545	1.58641	18.00000
Y	theta2	23.73200	1.31844	18.00000
Z	theta2	23.73200	1.31844	18.00000

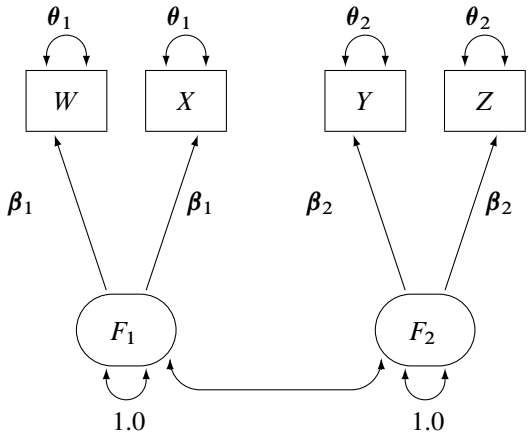
Specifying the Parallel Tests Model (H2) by the RAM Modeling Language: Lord Data

In the preceding section, you use the FACTOR modeling language of PROC CALIS to specify the parallel tests model. This model has also been specified by the PATH modeling language in the section "H2: Two-Factor Model with Parallel Tests for Lord Data" on page 315. The two specifications are equivalent; they lead to the same model fitting and estimation results. The main reason for providing two different types of modeling languages in PROC CALIS is that different researchers come from different fields of applications. Some researchers might be more comfortable with the confirmatory factor tradition, and some might equate structural equation models with path diagrams for variables.

PROC CALIS has still another modeling language that is closely related to the path diagram representation: the RAM model specification. In this section, the parallel tests model (H2) described in "H2: Two-Factor Model with Parallel Tests for Lord Data" on page 315 is used to illustrate the RAM model specification in PROC CALIS.

The path diagram for this model is reproduced in Figure 17.33.

Figure 17.33 H2: Two-Factor Model with Parallel Tests for Lord Data



The path diagram in Figure 17.33 can be readily transcribed into the RAM model specification by following these simple rules:

- Each single- or double- headed path corresponds to an entry in the RAM model specification.
- The single-headed paths are specified with the _A_ path type or matrix keyword.
- The double-headed paths are specified with the _P_ path type or matrix keyword.

At this point, you do not need to define the RAM model matrices _A_ and _P_, as long as you recognize that they are used as keywords to distinguish different path types. There are 11 single- or double- headed paths in Figure 17.33, and therefore you expect to specify these 11 elements in the RAM model, as shown in the following statements:

```
proc calis data=lord;
   ram var = W X Y Z F1 F2, /* W=1, X=2, Y=3, Z=4, F1=5, F2=6*/
       _A_   1  5  beta1,
       _A_   2  5  beta1,
       _A_   3  6  beta2,
       _A_   4  6  beta2,
       _P_   5  5  1.0,
       _P_   6  6  1.0,
       _P_   1  1  theta1,
       _P_   2  2  theta1,
       _P_   3  3  theta2,
       _P_   4  4  theta2,
       _P_   5  6  ;
run;
```

In this specification, the RAM statement invokes the RAM modeling language. The first option is the VAR= option where you specify the variables, observed and latent, in the model. The order in the VAR= variable list represents the order of these variables in the RAM model matrices. For this example, W is 1, X is 2, and so on. Next, you specify 11 RAM entries for the 11 path elements in the path diagram shown in Figure 17.33.

The first four entries are for the single-headed paths. They all begin with the _A_ keyword. In each of these _A_ entries, you specify the variable number of the outcome variable (being pointed at), and then the variable number of the predictor variable. At the end of the entry, you can specify a parameter name, a fixed value, an initial value, or nothing. In this example, all the _A_ entries are specified with parameter names. The first two paths are constrained because they use the same parameter name beta1. The next two paths are constrained because they use the same parameter name beta2.

The rest of the RAM entries in the example are of the _P_ type, which is for the specification of variances or covariances in the RAM model (the double-headed arrows in the path diagram). The _P_ entry with [5,5] is for the variance of the fifth variable, F1, on the VAR= list. This variance is fixed at 1.0 in the model, and so is the variance of the sixth variable, F2, in the next _P_ entry.

The next four _P_ entries are for the specification of error variances of the observed variables W, X, Y, and Z. You use the desired parameter names for constraining these parameters, as required in the parallel test model.

The last _P_ entry in the RAM statement is for the covariance between the fifth variable (F1) and the sixth variable (F2). You specify neither a parameter name nor a fixed value at the end of this entry. By default, this empty parameter specification is treated as a free parameter in the model. A parameter name for this entry is generated by PROC CALIS.

The fit summary for this RAM model is shown in Figure 17.34, and the estimation results are shown in Figure 17.35.

Figure 17.34 Fit Summary of RAM Model with Parallel Tests for Lord Data

Fit Summary

Chi-Square	1.9335
Chi-Square DF	5
Pr > Chi-Square	0.8583
Standardized RMSR (SRMSR)	0.0076
Adjusted GFI (AGFI)	0.9970
RMSEA Estimate	0.0000
Bentler Comparative Fit Index	1.0000

Figure 17.35 Estimation Results of RAM Model with Parallel Tests for Lord Data

RAM Pattern and Estimates

Matrix	--Row--		-Column-		Parameter	Estimate	Standard Error	t Value
_A_ (1)	W	1	F1	5	beta1	7.60099	0.26844	28.31580
	X	2	F1	5	beta1	7.60099	0.26844	28.31580
	Y	3	F2	6	beta2	8.59186	0.27967	30.72146
	Z	4	F2	6	beta2	8.59186	0.27967	30.72146
_P_ (2)	F1	5	F1	5		1.00000		
	F2	6	F2	6		1.00000		
	W	1	W	1	theta1	28.55545	1.58641	18.00000
	X	2	X	2	theta1	28.55545	1.58641	18.00000
	Y	3	Y	3	theta2	23.73200	1.31844	18.00000
	Z	4	Z	4	theta2	23.73200	1.31844	18.00000
	F1	5	F2	6	_Parm1	0.89864	0.01865	48.18011

Again, the model fit and the estimation results match those from the PATH model specification in Figure 17.24 and Figure 17.25, and those from the FACTOR model specification in Figure 17.31 and Figure 17.32.

A Combined Measurement-Structural Model

To illustrate a more complex model, this example uses some well-known data from Haller and Butterworth (1960). Various models and analyses of these data are given by Duncan, Haller, and Portes (1968), Jöreskog and Sörbom (1988), and Loehlin (1987).

The study concerns the career aspirations of high school students and how these aspirations are affected by close friends. The data are collected from 442 seventeen-year-old boys in Michigan. There are 329 boys in the sample who named another boy in the sample as a best friend. The data from these 329 boys paired with the data from their best friends are analyzed.

The method of data collection introduces two statistical problems. First, restricting the analysis to boys whose best friends are in the original sample causes the reduced sample to be biased. Second, since the data from a given boy might appear in two or more observations, the observations are not independent. Therefore, any statistical conclusions should be considered tentative. It is difficult to accurately assess the effects of the dependence of the observations on the analysis, but it could be argued on intuitive grounds that since each observation has data from two boys and since it seems likely that many of the boys appear in the data set at least twice, the effective sample size might be as small as half of the reported 329 observations.

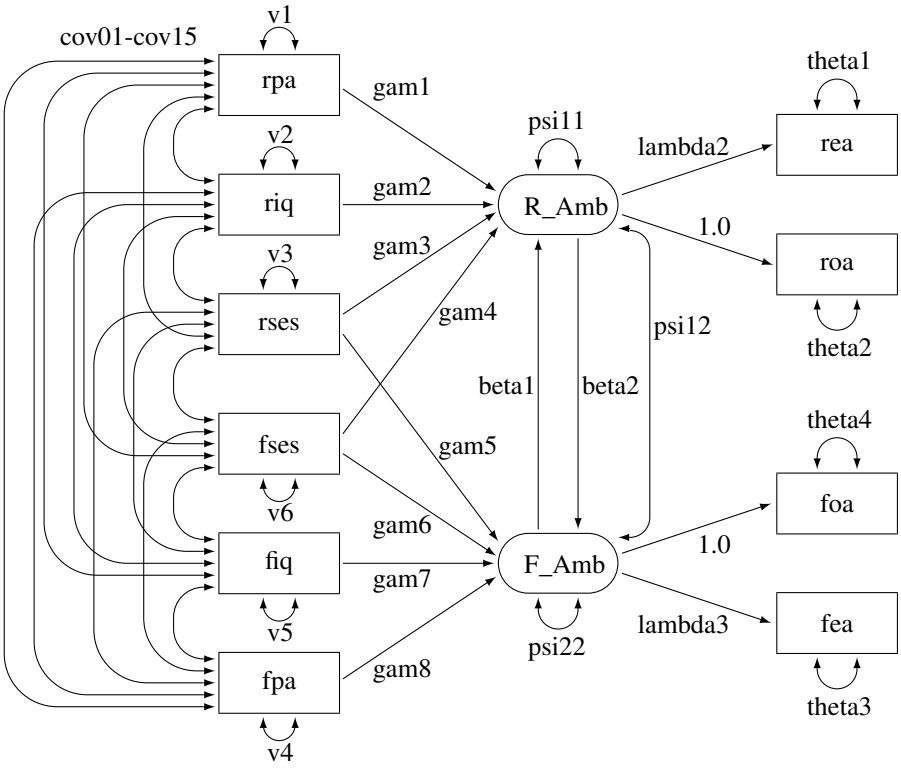
The correlation matrix, taken from Jöreskog and Sörbom (1988), is shown in the following DATA step:

```
title 'Peer Influences on Aspiration: Haller & Butterworth (1960)';
data aspire(type=corr);
   _type_='corr';
   input _name_ $ riq rpa rses roa rea fiq fpa fses foa fea;
   label riq='Respondent: Intelligence'
         rpa='Respondent: Parental Aspiration'
         rses='Respondent: Family SES'
         roa='Respondent: Occupational Aspiration'
         rea='Respondent: Educational Aspiration'
         fiq='Friend: Intelligence'
         fpa='Friend: Parental Aspiration'
         fses='Friend: Family SES'
         foa='Friend: Occupational Aspiration'
         fea='Friend: Educational Aspiration';
   datalines;
riq   1.      .      .      .      .      .       .      .      .      .
rpa   .1839  1.      .      .      .      .       .      .      .      .
rses  .2220  .0489  1.      .      .      .       .      .      .      .
roa   .4105  .2137  .3240  1.      .      .       .      .      .      .
rea   .4043  .2742  .4047  .6247  1.      .       .      .      .      .
fiq   .3355  .0782  .2302  .2995  .2863  1.      .      .      .      .
fpa   .1021  .1147  .0931  .0760  .0702  .2087   1.      .      .      .
fses  .1861  .0186  .2707  .2930  .2407  .2950  -.0438  1.      .      .
foa   .2598  .0839  .2786  .4216  .3275  .5007   .1988  .3607  1.      .
fea   .2903  .1124  .3054  .3269  .3669  .5191   .2784  .4105  .6404  1.
;
```

Career Aspiration: Analysis 1

The model analyzed by Jöreskog and Sörbom (1988) is displayed in the path diagram in Figure 17.36.

Figure 17.36 Path Diagram: Career Aspiration – Jöreskog and Sörbom (1988)



Two latent variables, R_Amb and F_Amb, represent the respondent's level of ambition and his best friend's level of ambition, respectively. The model states that the respondent's ambition (R_Amb) is determined by his intelligence (riq) and socioeconomic status (rses), his perception of his parents' aspiration for him (rpa), and his friend's socioeconomic status (fses) and ambition (F_Amb). It is assumed that his friend's intelligence (fiq) and parental aspiration (fpa) affect the respondent's ambition only indirectly through the friend's ambition (F_Amb). Ambition is indicated by the manifest variables of occupational (roa) and educational aspiration (rea), which are assumed to have uncorrelated residuals. The path coefficient from ambition to occupational aspiration is set to 1.0 to determine the scale of the ambition latent variable.

The path diagram shown in Figure 17.36 appears to be very complicated. Sometimes when researchers draw their path diagram with a lot of variables, they omit the covariances among exogenous variables for overall clarity. For example, the double-headed paths that represent cov01–cov15 in Figure 17.36 can be omitted. In addition, unconstrained variance and error variance parameters in the path diagram could be omitted without losing the pertinent information in the path diagram. For example, variance parameters v1–v6 and error variance parameters theta1–theta4 in Figure 17.36 detract from the main focus on the functional relationships of the model.

These omissions in the path diagram are in fact inconsequential when you transcribe them into the PATH model in PROC CALIS. The reason is that PROC CALIS employs several useful default parameterization rules that make the model specification process much easier and more intuitive. Here are the sets of default covariance structure parameters in the PATH modeling language:

- variances for all exogenous (observed or latent) variables
- error variances of all endogenous (observed or latent) variables
- covariances among all exogenous (observed or latent, excluding error) variables

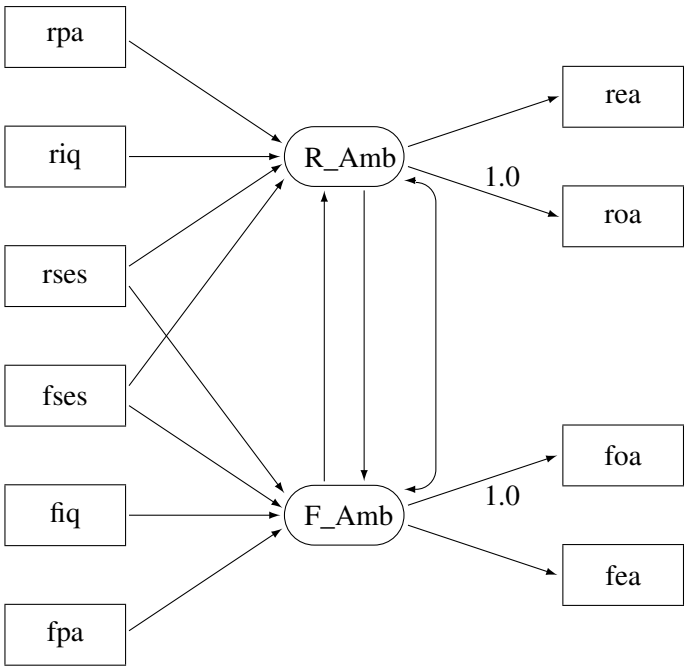
For example, these rules for setting default covariance structure parameters mean that the following sets of parameters in Figure 17.36 are optional in the path diagram representation and in the corresponding PATH model specification:

- v1–v6
- theta1–theta4, psi11, and psi22
- cov01–cov15

Note that the double-headed path labeled with psi12, which is a covariance parameter among error terms for R_Amb and F_Amb, is not a default parameter. As a result, it must be represented in the path diagram and in the PATH model specification.

Another simplification is to omit the unconstrained parameter names in the path diagram. In the PATH model specification, an "unnamed" parameter is a free parameter by default—there is no need to give unique names to denote free parameters. With all the mentioned simplifications, you can depict your path diagram simply as the one in Figure 17.37.

Figure 17.37 Simplified Path Diagram for Career Aspiration : Analysis 1



The simplified path diagram in Figure 17.37 is readily transcribed into the PATH model as shown in the following statements:

```
proc calis data=aspire nobs=329;
   path
      /* structural model of influences */
      R_Amb <--- rpa     ,
      R_Amb <--- riq     ,
      R_Amb <--- rses    ,
      R_Amb <--- fses    ,
      F_Amb <--- rses    ,
      F_Amb <--- fses    ,
      F_Amb <--- fiq     ,
      F_Amb <--- fpa     ,
      R_Amb <--- F_Amb   ,
      F_Amb <--- R_Amb   ,

      /* measurement model for aspiration */
      rea <---  R_Amb      ,
      roa <---  R_Amb  = 1.,
      foa <---  F_Amb  = 1.,
      fea <---  F_Amb    ;
   pcov
      R_Amb  F_Amb;
run;
```

Again, because you have 15 paths (single- or double- headed) in the path diagram, you expect that there are 15 entries in the PATH and the PCOV statements. Essentially, in this PATH model specification you specify all the functional relationships (single-headed arrows) in the path diagram and the covariance of error terms (double-headed arrows) for R_Amb and F_Amb.

Since this TYPE=CORR data set does not contain an observation with _TYPE_=N giving the sample size, it is necessary to specify the NOBS= option in the PROC CALIS statement.

The fit summary is displayed in Figure 17.38, and the estimation results are displayed in Figure 17.39.

Figure 17.38 Career Aspiration Data: Fit Summary for Analysis 1

Fit Summary

Chi-Square	26.6972
Chi-Square DF	15
Pr > Chi-Square	0.0313
Standardized RMSR (SRMSR)	0.0202
Adjusted GFI (AGFI)	0.9428
RMSEA Estimate	0.0488
Akaike Information Criterion	106.6972
Schwarz Bayesian Criterion	258.5395
Bentler Comparative Fit Index	0.9859

The model fit chi-square value is 26.6972 (df=15, p=0.0313). From the hypothesis testing point of view, this result says that this is an extreme sample given the model is true; therefore, the model should be rejected. But in social and behavioral sciences, you rarely abandon a model purely on the ground of chi-square significance test. The main reason is that you might only need to find a model that is approximately true, but the hypothesis testing framework is for testing exact model representation in the population. To determine whether a model is good or bad, you usually consult other fit indices. Several fit indices are shown in Figure 17.38.

The standardized RMSR is 0.0202. The RMSEA value is 0.0488. Both of these indices are smaller than 0.05, which indicate good model fit by convention. The adjusted GFI is 0.9428, and the comparative fit index is 0.9859. Again, values greater than 0.9 for these indices indicate good model fit by convention. Therefore, you can conclude that this is a good model for the data. Akaike's information criterion (AIC) and the Schwarz Bayesian criterion are also shown. You cannot interpret these values directly, but they are useful for model comparison given the same data, as shown in later sections.

Figure 17.39 Career Aspiration Data: Estimation Results for Analysis 1

PATH List

Path			Parameter	Estimate	Standard Error	t Value
R_Amb	<---	rpa	_Parm01	0.16122	0.03879	4.15602
R_Amb	<---	riq	_Parm02	0.24965	0.04398	5.67631
R_Amb	<---	rses	_Parm03	0.21840	0.04420	4.94151
R_Amb	<---	fses	_Parm04	0.07184	0.04971	1.44527
F_Amb	<---	rses	_Parm05	0.05754	0.04812	1.19561
F_Amb	<---	fses	_Parm06	0.21278	0.04169	5.10416
F_Amb	<---	fiq	_Parm07	0.32451	0.04352	7.45618
F_Amb	<---	fpa	_Parm08	0.14832	0.03645	4.06964
R_Amb	<---	F_Amb	_Parm09	0.19816	0.10228	1.93741
F_Amb	<---	R_Amb	_Parm10	0.21893	0.11125	1.96795
rea	<---	R_Amb	_Parm11	1.06268	0.09014	11.78936
roa	<---	R_Amb		1.00000		
foa	<---	F_Amb		1.00000		
fea	<---	F_Amb	_Parm12	1.07558	0.08131	13.22868

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	riq	_Add01	1.00000	0.07809	12.80625
	rpa	_Add02	1.00000	0.07809	12.80625
	rses	_Add03	1.00000	0.07809	12.80625
	fiq	_Add04	1.00000	0.07809	12.80625
	fpa	_Add05	1.00000	0.07809	12.80625
	fses	_Add06	1.00000	0.07809	12.80625
Error	roa	_Add07	0.41215	0.05122	8.04585
	rea	_Add08	0.33614	0.05210	6.45192
	foa	_Add09	0.40460	0.04618	8.76059
	fea	_Add10	0.31120	0.04593	6.77588
	R_Amb	_Add11	0.28099	0.04623	6.07782
	F_Amb	_Add12	0.22806	0.03850	5.92335

Figure 17.39 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
rpa	riq	_Add13	0.18390	0.05614	3.27564
rses	riq	_Add14	0.22200	0.05656	3.92503
rses	rpa	_Add15	0.04890	0.05528	0.88456
fiq	riq	_Add16	0.33550	0.05824	5.76060
fiq	rpa	_Add17	0.07820	0.05538	1.41195
fiq	rses	_Add18	0.23020	0.05666	4.06284
fpa	riq	_Add19	0.10210	0.05550	1.83955
fpa	rpa	_Add20	0.11470	0.05558	2.06377
fpa	rses	_Add21	0.09310	0.05545	1.67885
fpa	fiq	_Add22	0.20870	0.05641	3.70000
fses	riq	_Add23	0.18610	0.05616	3.31352
fses	rpa	_Add24	0.01860	0.05523	0.33680
fses	rses	_Add25	0.27070	0.05720	4.73226
fses	fiq	_Add26	0.29500	0.05757	5.12435
fses	fpa	_Add27	-0.04380	0.05527	-0.79249

In Figure 17.39, some of the paths do not show significance. That is, fses does not seem to be a good indicator of a respondent's ambition R_Amb and rses does not seem to be a good indicator of a friend's ambition F_Amb. The t values are 1.445 and 1.195, respectively, which are much smaller than the nominal 1.96 value at the 0.05 α-level of significance. Other paths are either significant or marginally significant.

You should be very cautious about interpreting the current analysis results for two reasons. First, as mentioned previously the data consist of dependent observations, and it was not certain how the issue could have been addressed beyond setting the sample size to half of the actual size. Second, structural equation modeling methodology is mainly applicable when you analyze covariance structures. When you input a correlation matrix for analysis, there is no guarantee that the statistical tests and standard error estimates are applicable. You should view the interpretations made here just as an exercise of applying structural equation modeling.

In Output 17.39, all parameter names are generated by PROC CALIS. Alternatively, you can also name these parameters in your PATH model specification. The following shows a PATH model specification that corresponds to the complete path diagram shown in Figure 17.36:

```
proc calis data=aspire nobs=329;
   path
      /* structural model of influences */
      rpa    --->  R_Amb   =  gam1,
      riq    --->  R_Amb   =  gam2,
      rses   --->  R_Amb   =  gam3,
      fses   --->  R_Amb   =  gam4,
      rses   --->  F_Amb   =  gam5,
      fses   --->  F_Amb   =  gam6,
      fiq    --->  F_Amb   =  gam7,
      fpa    --->  F_Amb   =  gam8,
      F_Amb  --->  R_Amb   = beta1,
      R_Amb  --->  F_Amb   = beta2,

      /* measurement model for aspiration */
      R_Amb  --->  rea     = lambda2,
      R_Amb  --->  roa     = 1.,
      F_Amb  --->  foa     = 1.,
      F_Amb  --->  fea     = lambda3;
   pvar
      R_Amb = psi11,
      F_Amb = psi22,
      rpa riq rses fpa fiq fses = v1-v6,
      rea roa fea foa = theta1-theta4;
   pcov
      R_Amb F_Amb = psi12,
      rpa riq rses fpa fiq fses = cov01-cov15;
run;
```

In this specification, the names of the parameters correspond to those used by Jöreskog and Sörbom (1988). Compared with the simplified version of the same model specification, you name 27 more parameters in the current specification. You have to be careful with this many parameters. If you inadvertently repeat the use of some parameter names, you will have unexpected constraints in the model.

The results from this analysis are displayed in Figure 17.40.

Figure 17.40 Career Aspiration Data: Estimation Results with Designated Parameter Names (Analysis 1)

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rpa	--->	R_Amb	gam1	0.16122	0.03879	4.15602
riq	--->	R_Amb	gam2	0.24965	0.04398	5.67631
rses	--->	R_Amb	gam3	0.21840	0.04420	4.94151
fses	--->	R_Amb	gam4	0.07184	0.04971	1.44527
rses	--->	F_Amb	gam5	0.05754	0.04812	1.19561
fses	--->	F_Amb	gam6	0.21278	0.04169	5.10416
fiq	--->	F_Amb	gam7	0.32451	0.04352	7.45618
fpa	--->	F_Amb	gam8	0.14832	0.03645	4.06964
F_Amb	--->	R_Amb	beta1	0.19816	0.10228	1.93741
R_Amb	--->	F_Amb	beta2	0.21893	0.11125	1.96795
R_Amb	--->	rea	lambda2	1.06268	0.09014	11.78936
R_Amb	--->	roa		1.00000		
F_Amb	--->	foa		1.00000		
F_Amb	--->	fea	lambda3	1.07558	0.08131	13.22868

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	R_Amb	psi11	0.28099	0.04623	6.07782
	F_Amb	psi22	0.22806	0.03850	5.92335
Exogenous	rpa	v1	1.00000	0.07809	12.80625
	riq	v2	1.00000	0.07809	12.80625
	rses	v3	1.00000	0.07809	12.80625
	fpa	v4	1.00000	0.07809	12.80625
	fiq	v5	1.00000	0.07809	12.80625
	fses	v6	1.00000	0.07809	12.80625
Error	rea	theta1	0.33614	0.05210	6.45192
	roa	theta2	0.41215	0.05122	8.04585
	fea	theta3	0.31120	0.04593	6.77588
	foa	theta4	0.40460	0.04618	8.76059

Figure 17.40 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
rpa	riq	cov01	0.18390	0.05614	3.27564
rpa	rses	cov02	0.04890	0.05528	0.88456
riq	rses	cov03	0.22200	0.05656	3.92503
rpa	fpa	cov04	0.11470	0.05558	2.06377
riq	fpa	cov05	0.10210	0.05550	1.83955
rses	fpa	cov06	0.09310	0.05545	1.67885
rpa	fiq	cov07	0.07820	0.05538	1.41195
riq	fiq	cov08	0.33550	0.05824	5.76060
rses	fiq	cov09	0.23020	0.05666	4.06284
fpa	fiq	cov10	0.20870	0.05641	3.70000
rpa	fses	cov11	0.01860	0.05523	0.33680
riq	fses	cov12	0.18610	0.05616	3.31352
rses	fses	cov13	0.27070	0.05720	4.73226
fpa	fses	cov14	-0.04380	0.05527	-0.79249
fiq	fses	cov15	0.29500	0.05757	5.12435

These are the same results as displayed in Figure 17.39 for the simplified PATH model specification. The only differences are the arrangement of estimation results and the naming of the parameters.

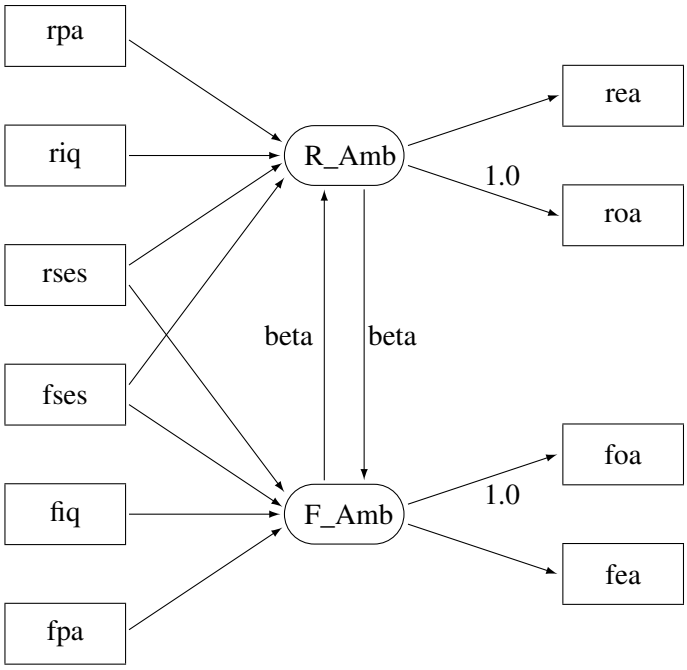
Career Aspiration: Analysis 2

Jöreskog and Sörbom (1988) present more detailed results from a second analysis in which two constraints are imposed:

- The coefficients that connect the latent ambition variables are equal.
- The covariance of the disturbances of the ambition variables is zero.

Applying these constraints to Figure 17.37, you get the path diagram displayed in Figure 17.41.

Figure 17.41 Path Diagram for Career Aspiration : Analysis 2



In Figure 17.41, the double-headed path that connected R_Amb and F_Amb no longer exists. Also, the single-headed paths between R_Amb and F_Amb are both labeled with beta, indicating the required constrained effects in the model. The path diagram in Figure 17.41 is transcribed into the PATH model in the following statements:

```
proc calis data=aspire nobs=329;
   path
      /* structural model of influences */
      rpa    ---> R_Amb         ,
      riq    ---> R_Amb         ,
      rses   ---> R_Amb         ,
      fses   ---> R_Amb         ,
      rses   ---> F_Amb         ,
      fses   ---> F_Amb         ,
      fiq    ---> F_Amb         ,
      fpa    ---> F_Amb         ,
      F_Amb  ---> R_Amb   = beta,
      R_Amb  ---> F_Amb   = beta,

      /* measurement model for aspiration */
      R_Amb  ---> rea         ,
      R_Amb  ---> roa     = 1.,
      F_Amb  ---> foa     = 1.,
      F_Amb  ---> fea     ;
run;
```

The only differences between the current specification and the preceding specification for Analysis 1 are the labeling of two paths with the same parameter beta and the deletion of PCOV statement where the covariance of R_Amb and F_Amb was specified in Analysis 1. The fit summary of the current model is displayed in Figure 17.42, and the estimation results are displayed in Figure 17.43.

Figure 17.42 Career Aspiration Data: Fit Summary for Analysis 2

Fit Summary

Chi-Square	26.8987
Chi-Square DF	17
Pr > Chi-Square	0.0596
Standardized RMSR (SRMSR)	0.0203
Adjusted GFI (AGFI)	0.9492
RMSEA Estimate	0.0421
Akaike Information Criterion	102.8987
Schwarz Bayesian Criterion	247.1489
Bentler Comparative Fit Index	0.9880

The model fit chi-square value is 26.8987 (df=17, p=0.0596). The standardized RMSR and the RMSEA are both less than 0.05, while the adjusted GFI and comparative fit index are both bigger than 0.9. All these indicate a good model fit, but how does this model (Analysis 2) compare with that in Analysis 1?

The difference between the chi-square values for Analyses 1 and 2 is $26.8987 - 26.6972 = 0.2015$ with two degrees of freedom, which is far from significant. This indicates that the restricted model (Analysis 2) fits as well as the unrestricted model (Analysis 1). The AIC is 102.8987, and the SBC is 247.149. Both of these values are smaller than that of Analysis 1 (106.697 for AIC and 258.540 for SBC), and hence they indicate that the current model is a better one.

Figure 17.43 Career Aspiration Data: Estimation Results for Analysis 2

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rpa	--->	R_Amb	_Parm01	0.16367	0.03872	4.22740
riq	--->	R_Amb	_Parm02	0.25395	0.04186	6.06726
rses	--->	R_Amb	_Parm03	0.22115	0.04187	5.28218
fses	--->	R_Amb	_Parm04	0.07728	0.04149	1.86264
rses	--->	F_Amb	_Parm05	0.06840	0.03868	1.76809
fses	--->	F_Amb	_Parm06	0.21839	0.03948	5.53198
fiq	--->	F_Amb	_Parm07	0.33063	0.04116	8.03314
fpa	--->	F_Amb	_Parm08	0.15204	0.03636	4.18169
F_Amb	--->	R_Amb	beta	0.18007	0.03912	4.60305
R_Amb	--->	F_Amb	beta	0.18007	0.03912	4.60305
R_Amb	--->	rea	_Parm09	1.06097	0.08921	11.89233
R_Amb	--->	roa		1.00000		
F_Amb	--->	foa		1.00000		
F_Amb	--->	fea	_Parm10	1.07359	0.08063	13.31498

Figure 17.43 *continued*

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	riq	_Add01	1.00000	0.07809	12.80625
	rpa	_Add02	1.00000	0.07809	12.80625
	rses	_Add03	1.00000	0.07809	12.80625
	fiq	_Add04	1.00000	0.07809	12.80625
	fpa	_Add05	1.00000	0.07809	12.80625
	fses	_Add06	1.00000	0.07809	12.80625
Error	roa	_Add07	0.41205	0.05103	8.07403
	rea	_Add08	0.33764	0.05178	6.52039
	foa	_Add09	0.40381	0.04608	8.76427
	fea	_Add10	0.31337	0.04574	6.85166
	R_Amb	_Add11	0.28113	0.04640	6.05867
	F_Amb	_Add12	0.22924	0.03889	5.89393

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
rpa	riq	_Add13	0.18390	0.05614	3.27564
rses	riq	_Add14	0.22200	0.05656	3.92503
rses	rpa	_Add15	0.04890	0.05528	0.88456
fiq	riq	_Add16	0.33550	0.05824	5.76060
fiq	rpa	_Add17	0.07820	0.05538	1.41195
fiq	rses	_Add18	0.23020	0.05666	4.06284
fpa	riq	_Add19	0.10210	0.05550	1.83955
fpa	rpa	_Add20	0.11470	0.05558	2.06377
fpa	rses	_Add21	0.09310	0.05545	1.67885
fpa	fiq	_Add22	0.20870	0.05641	3.70000
fses	riq	_Add23	0.18610	0.05616	3.31352
fses	rpa	_Add24	0.01860	0.05523	0.33680
fses	rses	_Add25	0.27070	0.05720	4.73226
fses	fiq	_Add26	0.29500	0.05757	5.12435
fses	fpa	_Add27	-0.04380	0.05527	-0.79249

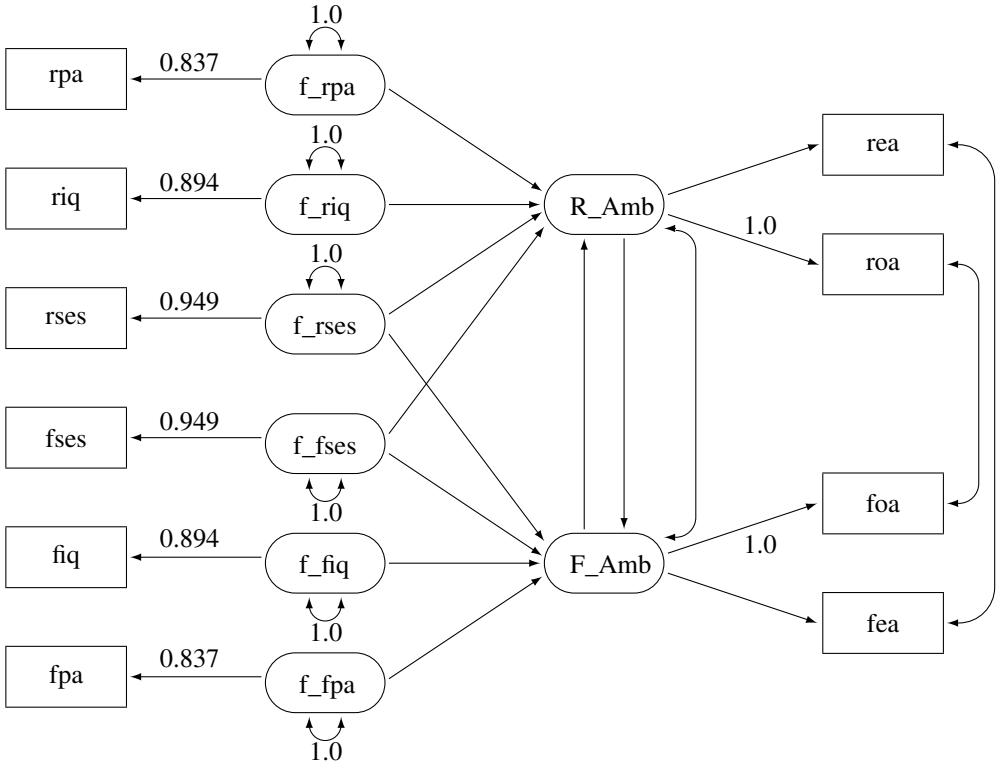
Like Analysis 1, the same two paths in the current analysis are not significant. That is, fses does not seem to be a good indicator of a respondent's ambition R_Amb, and rses does not seem to be a good indicator of a friend's ambition F_Amb. The t values are 1.862 and 1.768, respectively.

Career Aspiration: Analysis 3

Loehlin (1987) points out that the models considered are unrealistic in at least two respects. First, the variables of parental aspiration, intelligence, and socioeconomic status are assumed to be measured without error. Loehlin adds uncorrelated measurement errors to the model and assumes, for illustrative purposes, that the reliabilities of these variables are known to be 0.7, 0.8, and 0.9, respectively. In practice, these reliabilities would need to be obtained from a separate study of the same or a very similar population. If these constraints are omitted, the model is not identified. However, constraining parameters to a constant

in an analysis of a correlation matrix might make the chi-square goodness-of-fit test inaccurate, so there is more reason to be skeptical of the p-values. Second, the error terms for the respondent's aspiration are assumed to be uncorrelated with the corresponding terms for his friend. Loehlin introduces a correlation between the two educational aspiration error terms and between the two occupational aspiration error terms. These additions produce the path diagram for Loehlin's model shown in Figure 17.44.

Figure 17.44 Path Diagram for Career Aspiration: Analysis 3



In Figure 17.44, the observed variables rpa, riq, rses, fses, fiq, and fpa are all measured with measurement errors. Their true scores counterparts f_rpa, f_riq, f_rses, f_fses, f_fiq, and f_fpa are latent variables in the model. Path coefficients from these latent variables to the observed variables are fixed coefficients, indicating the square roots of the theoretical reliabilities in the model. These latent variables, rather than the observed counterparts, serve as predictors of the ambition factors R_Amb and F_Amb in the current model (Analysis 3). The error terms for these two latent factors are correlated, as indicated by a double-headed path (arrow) that connects the two factors. Correlated errors for the occupational aspiration variables (roa and foa) and the educational aspiration variables (rea and fea) are also shown in Figure 17.44. Again, these correlated errors are represented by two double-headed paths (arrows) in the path diagram.

You use the following statements to specify the path model for Analysis 3:

```
proc calis data=aspire nobs=329;
   path
      /* measurement model for intelligence and environment */
      rpa     <---  f_rpa    = 0.837,
      riq     <---  f_riq    = 0.894,
      rses    <---  f_rses   = 0.949,
      fses    <---  f_fses   = 0.949,
      fiq     <---  f_fiq    = 0.894,
      fpa     <---  f_fpa    = 0.837,

      /* structural model of influences */
      f_rpa   --->  R_Amb,
      f_riq   --->  R_Amb,
      f_rses  --->  R_Amb,
      f_fses  --->  R_Amb,
      f_rses  --->  F_Amb,
      f_fses  --->  F_Amb,
      f_fiq   --->  F_Amb,
      f_fpa   --->  F_Amb,
      F_Amb   --->  R_Amb,
      R_Amb   --->  F_Amb,

      /* measurement model for aspiration */
      R_Amb   --->  rea        ,
      R_Amb   --->  roa    = 1.,
      F_Amb   --->  foa    = 1.,
      F_Amb   --->  fea    ;
   pvar
      f_rpa f_riq f_rses f_fses f_fiq f_fpa = 6 * 1.0;
   pcov
      R_Amb F_Amb  ,
      rea  fea     ,
      roa  foa     ;
run;
```

In this specification, the measurement model for the six intelligence and environment variables are added. They are the first six paths in the PATH statement. Fixed constants are set for these path coefficients so as to make the measurement model identified and to set the required reliabilities of these measurement indicators. The structural model of influences and the measurement model for aspiration are the same as specified in Analysis 1. (See the section "Career Aspiration: Analysis 1" on page 329.) All the correlated errors are specified in the PCOV statement.

The fit summary of the current model is displayed in Figure 17.45.

Figure 17.45 Career Aspiration Data: Fit Summary for Analysis 3

Fit Summary

Chi-Square	12.0132
Chi-Square DF	13
Pr > Chi-Square	0.5266
Standardized RMSR (SRMSR)	0.0149
Adjusted GFI (AGFI)	0.9692
RMSEA Estimate	0.0000
Akaike Information Criterion	96.0132
Schwarz Bayesian Criterion	255.4476
Bentler Comparative Fit Index	1.0000

Since the p-value for the chi-square test is 0.5266, this model clearly cannot be rejected. Both the standardized RMSR and the RMSEA are very small, and both the adjusted GFI and the comparative fit index are high. All these point to an excellent model fit. However, Schwarz's Bayesian criterion for this model (SBC = 255.4476) is somewhat larger than for Jöreskog and Sörbom (1988) Analysis 2 in Figure 17.42 (SBC = 247.1489), suggesting that a more parsimonious model would be desirable.

The estimation results are displayed in Figure 17.46.

Figure 17.46 Career Aspiration Data: Estimation Results for Analysis 3

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rpa	<---	f_rpa		0.83700		
riq	<---	f_riq		0.89400		
rses	<---	f_rses		0.94900		
fses	<---	f_fses		0.94900		
fiq	<---	f_fiq		0.89400		
fpa	<---	f_fpa		0.83700		
f_rpa	--->	R_Amb	_Parm01	0.18370	0.05044	3.64197
f_riq	--->	R_Amb	_Parm02	0.28004	0.06139	4.56182
f_rses	--->	R_Amb	_Parm03	0.22616	0.05223	4.32999
f_fses	--->	R_Amb	_Parm04	0.08698	0.05476	1.58829
f_rses	--->	F_Amb	_Parm05	0.06327	0.05219	1.21242
f_fses	--->	F_Amb	_Parm06	0.21539	0.05121	4.20597
f_fiq	--->	F_Amb	_Parm07	0.35387	0.06741	5.24970
f_fpa	--->	F_Amb	_Parm08	0.16876	0.04934	3.42048
F_Amb	--->	R_Amb	_Parm09	0.11898	0.11396	1.04412
R_Amb	--->	F_Amb	_Parm10	0.13022	0.12067	1.07912
R_Amb	--->	rea	_Parm11	1.08399	0.09417	11.51051
R_Amb	--->	roa		1.00000		
F_Amb	--->	foa		1.00000		
F_Amb	--->	fea	_Parm12	1.11630	0.08627	12.93945

Figure 17.46 *continued*

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	f_rpa		1.00000		
	f_riq		1.00000		
	f_rses		1.00000		
	f_fses		1.00000		
	f_fiq		1.00000		
	f_fpa		1.00000		
Error	riq	_Add01	0.20874	0.07832	2.66518
	rpa	_Add02	0.29584	0.07774	3.80572
	rses	_Add03	0.09887	0.07803	1.26712
	roa	_Add04	0.42307	0.05243	8.06949
	rea	_Add05	0.32707	0.05452	5.99881
	fiq	_Add06	0.19989	0.07674	2.60483
	fpa	_Add07	0.29988	0.07807	3.84092
	fses	_Add08	0.10324	0.07824	1.31952
	foa	_Add09	0.42240	0.04730	8.93099
	fea	_Add10	0.28716	0.04804	5.97756
	R_Amb	_Add11	0.25418	0.04469	5.68740
	F_Amb	_Add12	0.19698	0.03814	5.16528

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
f_riq	f_rpa	_Add13	0.24677	0.07519	3.28202
f_rses	f_rpa	_Add14	0.06183	0.06945	0.89030
f_rses	f_riq	_Add15	0.26351	0.06687	3.94078
f_fses	f_rpa	_Add16	0.02382	0.06952	0.34267
f_fses	f_riq	_Add17	0.22136	0.06648	3.32983
f_fses	f_rses	_Add18	0.30156	0.06359	4.74210
f_fiq	f_rpa	_Add19	0.10853	0.07362	1.47416
f_fiq	f_riq	_Add20	0.42476	0.07219	5.88372
f_fiq	f_rses	_Add21	0.27250	0.06660	4.09143
f_fiq	f_fses	_Add22	0.34922	0.06771	5.15762
f_fpa	f_rpa	_Add23	0.15789	0.07873	2.00555
f_fpa	f_riq	_Add24	0.13084	0.07418	1.76387
f_fpa	f_rses	_Add25	0.11516	0.06978	1.65050
f_fpa	f_fses	_Add26	-0.05622	0.06971	-0.80648
f_fpa	f_fiq	_Add27	0.27867	0.07530	3.70082

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
R_Amb	F_Amb	_Parm13	-0.00936	0.05010	-0.18673
rea	fea	_Parm14	0.02308	0.03139	0.73545
roa	foa	_Parm15	0.11206	0.03258	3.43988

Like Analyses 1 and 2, two paths that concern the validity of the indicators in the current analysis do not show significance. That is, f_fses does not seem to be a good indicator of a respondent's ambition R_Amb, and f_rses does not seem to be a good indicator of a friend's ambition F_Amb. The t values are 1.588 and 1.212, respectively. In addition, in the current model (Analysis 3), the structural relationships between the ambition factors do not show significance. The t value for the path from the friend's ambition factor F_Amb on the respondent's ambition factor R_Amb is only 1.044, while the t value for the path from the respondent's ambition factor R_Amb on the friend's ambition factor F_Amb is only 1.079. These cast doubts on the validity of the structural model and perhaps even the entire model.

Fitting LISREL Models by the LISMOD Modeling Language

The model described in the section "Career Aspiration: Analysis 3" on page 340 provides a good example of the LISREL model. In PROC CALIS, the LISREL model specifications are supported by a matrix-based language called LISMOD (LISREL model). In this section, the path diagram in Figure 17.44 is specified by the LISMOD modeling language of PROC CALIS. See the section "Career Aspiration: Analysis 3" on page 340 for detailed descriptions of the model.

In order to understand the LISMOD modeling language of PROC CALIS, some basic understanding of the LISREL model is necessary. In a LISREL model, variables are classified into four distinct classes:

- ξ is a vector of exogenous (independent) latent variables in the model. They are specified in XI= variable list in the LISMOD statement.
- η is a vector of endogenous (dependent) latent variables in the model. They are specified in ETA= variable list in the LISMOD statement.
- x is a vector of observed indicator variables for ξ in the model. They are specified in XVAR= variable list in the LISMOD statement.
- y is a vector of observed indicator variables for η in the model. They are specified in YVAR= variable list in the LISMOD statement.

For detailed descriptions of the LISMOD modeling language, see the LISMOD statement and the section "The LISMOD Model and Submodels" on page 1197. To successfully set up a LISMOD model in PROC CALIS, you first need to recognize these classes of variables in your model. For the path diagram in Figure 17.44, it is not difficult to see the following:

- ξ is the vector of the intelligence and environmental factors: f_rpa, f_riq, f_rses, f_fses, f_fiq, and f_fpa. These variables are exogenous because no single-headed arrows point to them.
- η is the vector of the ambition factors: R_Amb, and F_Amb. They are endogenous because each of them has at least one single-headed arrow pointing to it.
- x is the vector of the observed indicator variables for the intelligence and environmental factors ξ. These indicators are rpa, riq, rses, fses, fiq, and fpa.

- y is the vector of observed indicator variables for the ambition factors η. These indicators are rea, roa, foa, and fea.

In LISMOD, you do not need to define error terms explicitly as latent variables. The parameters in LISMOD are defined as entries in various model matrices. The following statements specify the LISMOD model for the diagram in Figure 17.44:

```
proc calis data=aspire nobs=329;
   lismod
      xi   = f_rpa f_riq f_rses f_fses f_fiq f_fpa,
      eta  = R_Amb F_Amb,
      xvar = rpa riq rses fses fiq fpa,
      yvar = rea roa foa fea;

   /* measurement model for aspiration */
   matrix _lambday_ [1,1], [2,1] = 1.0, [3,2] = 1.0, [4,2];
   matrix _thetay_  [4,1], [3,2];

   /* measurement model for intelligence and environment */
   matrix _lambdax_ [1,1] = 0.837 0.894 0.949 0.949 0.894 0.837;

   /* structural model of influences */
   matrix _beta_ [2,1],[1,2];
   matrix _gamma_ [1,1 to 4], [2,3 to 6];

   /* Covariances among Eta-variables */
   matrix _psi_ [2,1];

   /* Fixed variances for Xi-variables */
   matrix _phi_ [1,1] = 6 * 1.0;
run;
```

The LISMOD statement invokes the LISMOD modeling language of PROC CALIS. In the LISMOD statement, you list the four classes of variables in the model in the XI=, ETA=, XVAR=, and YVAR= variable lists, respectively. After you define the four classes of variables, you use several MATRIX statements to specify the model matrices and the parameters in the model.

Basically, there are three model components in the LISMOD specification: two measurement models and one structural model. The first measurement model specifies the functional relationships between observed variables y (YVAR= variables) and the endogenous (dependent) latent factors η (ETA= variables). The second measurement model specifies the functional relationships between observed variables x and (XVAR= variables) and the exogenous (independent) latent factors ξ (XI= variables). The structural model specifies the relationships between the endogenous and exogenous latent variables η and ξ. To facilitate the discussion of these model components and the corresponding LISMOD model specification, some initial model output from PROC CALIS are shown.

The Measurement Model for y

The first component of the LISMOD specification is the measurement model for y, as shown in the following equation:

$$y = \mathbf{\Lambda_y}\eta + \epsilon$$

In the context of covariance structure analysis, without loss of generality, it is assumed that y and η are centered so that there is no intercept term in the equation. This equation essentially states that y is a function of the true scores vector η plus the error term ϵ, which is independent of η. The model matrices involved in this measurement model are $\mathbf{\Lambda_y}$ (effects of η on y) and $\mathbf{\Theta_y}$, which is the covariance matrix of ϵ.

For the career aspiration data, you specify the following two MATRIX statements for this measurement model:

```
matrix _lambday_ [1,1], [2,1] = 1.0, [3,2] = 1.0, [4,2];
matrix _thetay_  [4,1], [3,2];
```

The first matrix statement is for matrix $\mathbf{\Lambda_y}$. You specify four parameters in this matrix. The [1,1] and [4,2] elements are free parameters, and the [2,1] and [3,2] elements have fixed values of 1. You do not specify other elements in this matrix. By default, unspecified elements in the $\mathbf{\Lambda_y}$ matrix are fixed zeros. You can check your initial model specification of this matrix, as shown in the Figure 17.47.

Figure 17.47 Career Aspiration Analysis 3: Initial Measurement Model for y

Initial _LAMBDAY_ Matrix

	R_Amb	F_Amb
rea	.	0
	[_Parm01]	
roa	1.0000	0
foa	0	1.0000
fea	0	.
		[_Parm02]

Figure 17.47 *continued*

Initial _THETAY_ Matrix

	rea	roa	foa	fea
rea	. [_Add07]	0	0	. [_Parm13]
roa	0	. [_Add08]	. [_Parm14]	0
foa	0	. [_Parm14]	. [_Add09]	0
fea	. [_Parm13]	0	0	. [_Add10]

NOTE: Parameters with prefix '_Add' are added by PROC CALIS.

In Figure 17.47, the initial _LAMBDAY_ matrix is a 4×2 matrix. The _LAMBDAY_ matrix contains information about the relationships between the row indicator variables y (YVAR= variables) and the column factors η (ETA= variables). As specified in the MATRIX statement for _LAMBDAY_, the [1,1] and [4,2] are free parameters named _Parm01 and _Parm02, respectively. These parameter names are generated by PROC CALIS. Fixed values 1.0 appear in the [2,1] and [3,2] elements. These fixed values are used to identify the scales of the latent variables R_Amb and F_Amb.

The _THETAY_ matrix in Figure 17.47 is the covariance matrix among the error terms for the y-variables (YVAR= variables). This is a 4×4 matrix for the four measured indicators. As specified in the MATRIX statement for _THETAY_, the [4,1] and [3,2] elements are free parameters named _Parm13 and _Parm14, respectively. Because _THETAY_ is a symmetric matrix, elements [1,4] and [2,3] are also implicitly specified as parameters in this model matrix.

As shown in Figure 17.47, PROC CALIS adds four default free parameters to the _THETAY_ matrix. On the diagonal of the _THETAY_ matrix, parameters _Add07, _Add08, _Add09, and _Add10 are added as default free parameters by PROC CALIS automatically. In general, error variances are default free parameters in PROC CALIS. You do not have to specify them but you can specify them if you want to, especially when you need to set fixed values or other constraints on them.

The Measurement Model for x

The second component of the LISMOD specification is the measurement model for x, as shown in the following equation:

$$x = \mathbf{\Lambda_x}\xi + \delta$$

The measurement model for x is similar to that for y. Assuming that x and ξ are centered, this equation states that x is a function of the true scores vector ξ plus the error term δ, which is independent of ξ. The model matrices involved in this measurement model are $\mathbf{\Lambda_x}$ (effects of ξ on x) and $\mathbf{\Theta_x}$, which is the covariance matrix of δ.

For the career aspiration data, you specify the following MATRIX statement for this measurement model:

```
matrix _lambdax_ [1,1] = 0.837 0.894 0.949 0.949 0.894 0.837;
```

Figure 17.48 shows the output related to the specification of the measurement model for x.

Figure 17.48 Career Aspiration Analysis 3: Initial Measurement Model for x

Initial _LAMBDAX_ Matrix

	f_rpa	f_riq	f_rses	f_fses	f_fiq	f_fpa
rpa	0.8370	0	0	0	0	0
riq	0	0.8940	0	0	0	0
rses	0	0	0.9490	0	0	0
fses	0	0	0	0.9490	0	0
fiq	0	0	0	0	0.8940	0
fpa	0	0	0	0	0	0.8370

Initial _THETAX_ Matrix

	rpa	riq	rses	fses	fiq	fpa
rpa	. [_Add01]	0	0	0	0	0
riq	0	. [_Add02]	0	0	0	0
rses	0	0	. [_Add03]	0	0	0
fses	0	0	0	. [_Add04]	0	0
fiq	0	0	0	0	. [_Add05]	0
fpa	0	0	0	0	0	. [_Add06]

NOTE: Parameters with prefix '_Add' are added by PROC CALIS.

In Figure 17.48, the initial _LAMBDAX_ matrix is a 6×6 matrix. The _LAMBDAX_ matrix contains information about the relationships between the row indicator variables x (XVAR= variables) and the column factors ξ (XI= variables). As specified in the MATRIX statement for _LAMBDAX_, the diagonal elements are filled with the fixed values provided. The [1,1] specification in the MATRIX statement for _LAMBDAX_ provides the starting element for the subsequent parameter list to fill in. In this case, the list contains six fixed values, and PROC CALIS proceeds from [1,1] to [2,2], [3,3] and so on until the entire list of parameters is consumed. This kind of notation is a shortcut of the following equivalent specification:

```
matrix _lambdax_ [1,1]=0.837, [2,2]=0.894, [3,3]=0.949,
                 [4,4]=0.949, [5,5]=0.894, [6,6]=0.837;
```

PROC CALIS provides many different kinds of shortcuts in specifying matrix elements. See the MATRIX statement of Chapter 26, "The CALIS Procedure," for details.

At the bottom of Figure 17.48, the initial _THETAX_ matrix is shown. Even though you did not specify any elements of this matrix in any MATRIX statements, the diagonal elements of this matrix are set as default parameters by PROC CALIS. Default parameters added by PROC CALIS are all denoted by names with the prefix '_Add'.

The Structural Model

The last component of the LISMOD specification is the structural model that describes the relationship between η and ξ, as shown in the following equation:

$$\eta = \boldsymbol{\beta}\eta + \boldsymbol{\Gamma}\xi + \zeta$$

In this equation, η is endogenous (dependent) and ξ is exogenous (independent). Variables in η can have effects among themselves. Their effects are specified in the $\boldsymbol{\beta}$ matrix. The effects of ξ on η are specified in the $\boldsymbol{\Gamma}$ matrix. Finally, the error term for the structural relationships is denoted by ζ, which is independent of ξ.

There are four model matrices assumed in the structural model. $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$ are matrices for the effects of variables. In addition, matrix $\boldsymbol{\Psi}$ denotes the covariance matrix for the error term ζ, and matrix $\boldsymbol{\Phi}$ denotes the covariance matrix of ξ.

For the career aspiration data, you use the following MATRIX statements for the structural model:

```
matrix _beta_ [2,1],[1,2];
matrix _gamma_ [1,1 to 4], [2,3 to 6];
matrix _psi_ [2,1];
matrix _phi_ [1,1] = 6 * 1.0;
```

In Figure 17.49, initial _BETA_ and _GAMMA_ matrices are shown.

Figure 17.49 Career Aspiration Analysis 3: Initial Structural Equations

Initial _BETA_ Matrix

	R_Amb	F_Amb
R_Amb	0	. [_Parm12]
F_Amb	. [_Parm11]	0

Initial _GAMMA_ Matrix

	f_rpa	f_riq	f_rses	f_fses	f_fiq	f_fpa
R_Amb	. [_Parm03]	. [_Parm04]	. [_Parm05]	. [_Parm06]	0	0
F_Amb	0	0	. [_Parm07]	. [_Parm08]	. [_Parm09]	. [_Parm10]

In Figure 17.49, the _BETA_ matrix contains information about the relationships among the η-variables (ETA= variables). Both the row and column variables of the _BETA_ matrix refer to the list of η-variables. The row variables receive effects from the column variables. You specify two parameters in the _BETA_ matrix: element [2,1] is the effect of R_Amb on F_Amb, and element [2,1] is the effect of F_Amb on R_Amb. Other effects are fixed zeros in this matrix.

The _GAMMA_ matrix contains information about the relationships between the η-variables (ETA= variables) and the ξ-variables (XI= variables). The row variables are the η-variables, which receive effects from the column ξ-variables. You specify eight free parameters in this matrix. These eight parameters represent the eight path coefficients from ξ (the intelligence and environment factors) to the η variables (the ambition factors), as shown in the path diagram in Figure 17.44. A shortcut in the MATRIX statement syntax for the _GAMMA_ matrix has been used. That is, `[1, 1 to 4]` means the [1,1], [1,2], [1,3], and [1,4] elements, and `[2, 3 to 6]` means the [2,3], [2,4], [2,5], and [2,6] elements. All these elements are free parameters in the model and free parameter names are generated for these elements.

Figure 17.50 shows the initial _PSI_ and _PHI_ matrices.

Figure 17.50 Career Aspiration Analysis 3: Initial Variances and Covariances

Initial _PSI_ Matrix

	R_Amb	F_Amb
R_Amb	. [_Add11]	. [_Parm15]
F_Amb	. [_Parm15]	. [_Add12]

NOTE: Parameters with prefix '_Add' are added by PROC CALIS.

Initial _PHI_ Matrix

	f_rpa	f_riq	f_rses	f_fses	f_fiq	f_fpa
f_rpa	1.0000	. [_Add13]	. [_Add14]	. [_Add16]	. [_Add19]	. [_Add23]
f_riq	. [_Add13]	1.0000	. [_Add15]	. [_Add17]	. [_Add20]	. [_Add24]
f_rses	. [_Add14]	. [_Add15]	1.0000	. [_Add18]	. [_Add21]	. [_Add25]
f_fses	. [_Add16]	. [_Add17]	. [_Add18]	1.0000	. [_Add22]	. [_Add26]
f_fiq	. [_Add19]	. [_Add20]	. [_Add21]	. [_Add22]	1.0000	. [_Add27]
f_fpa	. [_Add23]	. [_Add24]	. [_Add25]	. [_Add26]	. [_Add27]	1.0000

NOTE: Parameters with prefix '_Add' are added by PROC CALIS.

The _PSI_ matrix contains information about the covariances of error terms for the η-variables, which are endogenous in the structural model. There are two η-variables in the model—the two ambition factors R_Amb and F_Amb. You specify the [2,1] element as a free parameter in the MATRIX statement for _PSI_. This means that the error covariance between R_Amb and F_Amb is a free parameter to estimate in the model. In Figure 17.50, both [2,1] and [1,2] elements are named as _Parm15 because _PSI_ is a symmetric matrix. Again, the diagonal elements of this covariance matrix, which are for the error variances of the ambition factors, are default free parameters in PROC CALIS. These parameters are named with the prefix _Add.

Finally, the _PHI_ matrix contains information about the covariances among the exogenous latent factors in the structural model. For the _PHI_ matrix, you fix all the diagonal elements to 1 in the MATRIX statement for _PHI_. This makes the latent variable scales identified. These fixed values are echoed in the output of the initial _PHI_ matrix shown in Figure 17.50. In addition, all covariances among latent exogenous variables are set to be free parameters by default.

Fit Summary of the LISMOD Model for Career Aspiration Analysis 3

Figure 17.51 shows the fit summary of the LISMOD model. All these fit index values match those from using the PATH model specification of the same model, as shown in Figure 17.45. Therefore, you are confident that the current LISMOD model specification is equivalent to the PATH model specification shown in the section "Career Aspiration: Analysis 3" on page 340.

Figure 17.51 Career Aspiration Analysis 3: Fit Summary of the LISMOD Model

Fit Summary

Chi-Square	12.0132
Chi-Square DF	13
Pr > Chi-Square	0.5266
Standardized RMSR (SRMSR)	0.0149
Adjusted GFI (AGFI)	0.9692
RMSEA Estimate	0.0000
Akaike Information Criterion	96.0132
Schwarz Bayesian Criterion	255.4476
Bentler Comparative Fit Index	1.0000

Estimation results are shown in Figure 17.52, Figure 17.53, and Figure 17.54, respectively for the measurement model for y, measurement model for x, and the structural model. These are the same estimation results as those from the equivalent PATH model specification in Figure 17.46. However, estimates in the LISMOD model are now arranged in the matrix form, with standard error estimates and t values shown.

Figure 17.52 Career Aspiration Analysis 3: Estimation of Measurement Model for y

LAMBDAY Matrix: Estimate/StdErr/t-value

	R_Amb	F_Amb
rea	1.0840 0.0942 11.5105 [_Parm01]	0
roa	1.0000	0
foa	0	1.0000
fea	0	1.1163 0.0863 12.9394 [_Parm02]

THETAY Matrix: Estimate/StdErr/t-value

	rea	roa	foa	fea
rea	0.3271 0.0545 5.9988 [_Add07]	0	0	0.0231 0.0314 0.7355 [_Parm13]
roa	0	0.4231 0.0524 8.0695 [_Add08]	0.1121 0.0326 3.4399 [_Parm14]	0
foa	0	0.1121 0.0326 3.4399 [_Parm14]	0.4224 0.0473 8.9310 [_Add09]	0
fea	0.0231 0.0314 0.7355 [_Parm13]	0	0	0.2872 0.0480 5.9776 [_Add10]

Figure 17.53 Career Aspiration Analysis 3: Estimation of Measurement Model for x

LAMBDAX Matrix: Estimate/StdErr/t-value

	f_rpa	f_riq	f_rses	f_fses	f_fiq	f_fpa
rpa	0.8370	0	0	0	0	0
riq	0	0.8940	0	0	0	0
rses	0	0	0.9490	0	0	0
fses	0	0	0	0.9490	0	0
fiq	0	0	0	0	0.8940	0
fpa	0	0	0	0	0	0.8370

Figure 17.53 *continued*

THETAX Matrix: Estimate/StdErr/t-value

	rpa	riq	rses	fses	fiq	fpa
rpa	0.2958 0.0777 3.8057 [_Add01]	0	0	0	0	0
riq	0	0.2087 0.0783 2.6652 [_Add02]	0	0	0	0
rses	0	0	0.0989 0.0780 1.2671 [_Add03]	0	0	0
fses	0	0	0	0.1032 0.0782 1.3195 [_Add04]	0	0
fiq	0	0	0	0	0.1999 0.0767 2.6048 [_Add05]	0
fpa	0	0	0	0	0	0.2999 0.0781 3.8409 [_Add06]

Figure 17.54 Career Aspiration Analysis 3: Estimation of Structural Model

BETA Matrix: Estimate/StdErr/t-value

	R_Amb	F_Amb
R_Amb	0	0.1190
		0.1140
		1.0441
		[_Parm12]
F_Amb	0.1302	0
	0.1207	
	1.0791	
	[_Parm11]	

GAMMA Matrix: Estimate/StdErr/t-value

	f_rpa	f_riq	f_rses	f_fses	f_fiq	f_fpa
R_Amb	0.1837	0.2800	0.2262	0.0870	0	0
	0.0504	0.0614	0.0522	0.0548		
	3.6420	4.5618	4.3300	1.5883		
	[_Parm03]	[_Parm04]	[_Parm05]	[_Parm06]		
F_Amb	0	0	0.0633	0.2154	0.3539	0.1688
			0.0522	0.0512	0.0674	0.0493
			1.2124	4.2060	5.2497	3.4205
			[_Parm07]	[_Parm08]	[_Parm09]	[_Parm10]

PSI Matrix: Estimate/StdErr/t-value

	R_Amb	F_Amb
R_Amb	0.2542	-0.009355
	0.0447	0.0501
	5.6874	-0.1867
	[_Add11]	[_Parm15]
F_Amb	-0.009355	0.1970
	0.0501	0.0381
	-0.1867	5.1653
	[_Parm15]	[_Add12]

Figure 17.54 *continued*

PHI Matrix: Estimate/StdErr/t-value

	f_rpa	f_riq	f_rses	f_fses	f_fiq	f_fpa
f_rpa	1.0000	0.2468	0.0618	0.0238	0.1085	0.1579
		0.0752	0.0695	0.0695	0.0736	0.0787
		3.2820	0.8903	0.3427	1.4742	2.0056
		[_Add13]	[_Add14]	[_Add16]	[_Add19]	[_Add23]
f_riq	0.2468	1.0000	0.2635	0.2214	0.4248	0.1308
	0.0752		0.0669	0.0665	0.0722	0.0742
	3.2820		3.9408	3.3298	5.8837	1.7639
	[_Add13]		[_Add15]	[_Add17]	[_Add20]	[_Add24]
f_rses	0.0618	0.2635	1.0000	0.3016	0.2725	0.1152
	0.0695	0.0669		0.0636	0.0666	0.0698
	0.8903	3.9408		4.7421	4.0914	1.6505
	[_Add14]	[_Add15]		[_Add18]	[_Add21]	[_Add25]
f_fses	0.0238	0.2214	0.3016	1.0000	0.3492	-0.0562
	0.0695	0.0665	0.0636		0.0677	0.0697
	0.3427	3.3298	4.7421		5.1576	-0.8065
	[_Add16]	[_Add17]	[_Add18]		[_Add22]	[_Add26]
f_fiq	0.1085	0.4248	0.2725	0.3492	1.0000	0.2787
	0.0736	0.0722	0.0666	0.0677		0.0753
	1.4742	5.8837	4.0914	5.1576		3.7008
	[_Add19]	[_Add20]	[_Add21]	[_Add22]		[_Add27]
f_fpa	0.1579	0.1308	0.1152	-0.0562	0.2787	1.0000
	0.0787	0.0742	0.0698	0.0697	0.0753	
	2.0056	1.7639	1.6505	-0.8065	3.7008	
	[_Add23]	[_Add24]	[_Add25]	[_Add26]	[_Add27]	

Some Important PROC CALIS Features

In this section, some of the main features of PROC CALIS are introduced. Emphasis is placed on showing how these features are useful in practical structural equation modeling.

Modeling Languages for Specifying Models

PROC CALIS provides several modeling languages to specify a model. Different modeling languages in PROC CALIS are signified by the main model specification statement used. In this chapter, you have seen examples of the FACTOR, LINEQS, LISMOD, MSTRUCT, PATH, and RAM modeling languages. Depending on your modeling philosophy and the type of the model, you can choose a modeling language that is most suitable for your application. For example, models specified using structural equations can be transcribed directly into the LINEQS statement. Models that are hypothesized using path diagrams can be

described easily in the PATH or RAM statement. First-order confirmatory or exploratory factor models are most conveniently specified by using the FACTOR and MATRIX statements. Traditional LISREL models are supported through the LISMOD and MATRIX statements. Finally, patterned covariance and mean models can be specified directly by the MSTRUCT and MATRIX statements, or by the COVPATTERN= and MEANPATTERN= options.

For most applications in structural equation modeling, the PATH and LINEQS statements are the easiest to use. For testing the built-in covariance and mean patterns of PROC CALIS, the use of the COVPATTERN= and the MEANPATTERN= options are the most efficient. In other cases, the FACTOR, LISMOD, MSTRUCT, or RAM statement might be more suitable. For very general matrix model specifications, you can use the COSAN modeling language. See the COSAN statement and the section "The COSAN Model" on page 1178 of Chapter 26, "The CALIS Procedure," for details about the COSAN modeling language. See also the section "Which Modeling Language?" on page 997 in Chapter 26, "The CALIS Procedure," for a more detailed discussion about the use of different modeling languages.

Estimation Methods

The CALIS procedure provides six methods of estimation specified by the METHOD= option:

DWLS	diagonally weighted least squares
FIML	full-information maximum likelihood
GLS	normal theory generalized least squares
ML	maximum likelihood for multivariate normal distributions
ULS	unweighted least squares
WLS	weighted least squares for arbitrary distributions

Each estimation method is based on finding parameter estimates that minimize a discrepancy (badness-of-fit) function, which measures the difference between the observed sample covariance matrix and the fitted (predicted) covariance matrix, given the model and the parameter estimates. The difference between the observed sample mean vector and the fitted (predicted) mean vector is also taken into account when the mean structures are modeled. See the section "Estimation Criteria" on page 1231 in Chapter 26, "The CALIS Procedure," for formulas, or refer to Loehlin (1987, pp. 54–62) and Bollen (1989, pp. 104–123) for further discussion.

The default estimation is METHOD=ML, which is the most popular method for applications. The option METHOD=GLS usually produces very similar results to those produced by METHOD=ML. If your data contain random missing values and it is important to use the information from those incomplete observations, you might want to use the FIML method, which provides a sound treatment of missing values in data. METHOD=ML and METHOD=FIML are essentially the same method when you do not have missing values (see Example 26.15 of Chapter 26, "The CALIS Procedure,"). Asymptotically, ML and GLS are the same. Both methods assume a multivariate normal distribution in the population. The WLS method with the default weight matrix is equivalent to the asymptotically distribution free (ADF) method, which yields asymptotically normal estimates regardless of the distribution in the population. When the multivariate normal assumption is in doubt, especially if the variables have high kurtosis, you should seriously consider the WLS method. When a correlation matrix is analyzed, only WLS can produce correct standard error estimates. However, in order to use the WLS method with the expected statistical properties, the sample size must be large. Several thousand might be a minimum requirement.

The ULS and DWLS methods yield reasonable estimates under less restrictive assumptions. You can apply these methods to normal or nonnormal situations or to covariance or correlation matrices. The drawback is that the statistical qualities of the estimates seem to be unknown. For this reason, PROC CALIS does not provide standard errors or test statistics with these two methods.

You cannot use METHOD=ML or METHOD=GLS if the observed covariance matrix is singular. You can either remove variables involved in the linear dependencies or use less restrictive estimation methods such as ULS. Specifying METHOD=ML assumes that the predicted covariance matrix is nonsingular. If ML fails because of a singular predicted covariance matrix, you need to examine whether the model specification leads to the singularity. If so, modify the model specification to eliminate the problem. If not, you probably need to use other estimation methods.

You should remove outliers and try to transform variables that are skewed or heavy-tailed. This applies to all estimation methods, since all the estimation methods depend on the sample covariance matrix, and the sample covariance matrix is a poor estimator for distributions with high kurtosis (Bollen 1989, pp. 415–418; Huber 1981; Hampel et al. 1986). PROC CALIS displays estimates of univariate and multivariate kurtosis (Bollen 1989, pp. 418–425) if you specify the KURTOSIS option in the PROC CALIS statement.

See the section "Estimation Methods" on page 359 for the general use of these methods. See the section "Estimation Criteria" on page 1231 of Chapter 26, "The CALIS Procedure," for details about these estimation criteria.

Statistical Inference

When you specify the ML, FIML, GLS, or WLS estimation with appropriate models, PROC CALIS can compute the following:

- a chi-square goodness-of-fit test of the specified model versus the alternative that the data are from a population with unconstrained covariance matrix (Loehlin 1987, pp. 62–64; Bollen 1989, pp. 110, 115, 263–269)
- approximate standard errors of the parameter estimates (Bollen 1989, pp. 109, 114, 286), displayed with the STDERR option
- various modification indices, requested via the MODIFICATION or MOD option, that give the approximate change in the chi-square statistic that would result from removing constraints on the parameters or constraining additional parameters to zero (Bollen 1989, pp. 293–303)

If you have two models such that one model results from imposing constraints on the parameters of the other, you can test the constrained model against the more general model by fitting both models with PROC CALIS. If the constrained model is correct, the difference between the chi-square goodness of fit statistics for the two models has an approximate chi-square distribution with degrees of freedom equal to the difference between the degrees of freedom for the two models (Loehlin 1987, pp. 62–67; Bollen 1989, pp. 291–292).

All of the test statistics and standard errors computed under ML and GLS depend on the assumption of multivariate normality. Normality is a much more important requirement for data with random independent variables than it is for fixed independent variables. If the independent variables are random, distributions with high kurtosis tend to give liberal tests and excessively small standard errors, while low kurtosis tends to produce the opposite effects (Bollen 1989, pp. 266–267, 415–432).

All test statistics and standard errors computed by PROC CALIS are based on asymptotic theory and should not be trusted in small samples. There are no firm guidelines on how large a sample must be for the asymptotic theory to apply with reasonable accuracy. Some simulation studies have indicated that problems are likely to occur with sample sizes less than 100 (Loehlin 1987, pp. 60–61; Bollen 1989, pp. 267–268). Extrapolating from experience with multiple regression would suggest that the sample size should be at least 5 to 20 times the number of parameters to be estimated in order to get reliable and interpretable results. The WLS method might even require that the sample size be over several thousand.

The asymptotic theory requires the parameter estimates to be in the interior of the parameter space. If you do an analysis with inequality constraints and one or more constraints are active at the solution (for example, if you constrain a variance to be nonnegative and the estimate turns out to be zero), the chi-square test and standard errors might not provide good approximations to the actual sampling distributions.

For modeling correlation structures, the only theoretically correct method is the WLS method with the default ASYCOV=CORR option. For other methods, standard error estimates for modeling correlation structures might be inaccurate even for sample sizes as large as 400. The chi-square statistic is generally the same regardless of which matrix is analyzed, provided that the model involves no scale-dependent constraints. However, if the purpose is to obtain reasonable parameter estimates for the correlation structures only, then you might also find other estimation methods useful.

If you fit a model to a correlation matrix and the model constrains one or more elements of the predicted matrix to equal 1.0, the degrees of freedom of the chi-square statistic must be reduced by the number of such constraints. PROC CALIS attempts to determine which diagonal elements of the predicted correlation matrix are constrained to a constant, but it might fail to detect such constraints in complicated models, particularly when programming statements are used. If this happens, you should add parameters to the model to release the constraints on the diagonal elements.

Multiple-Group Analysis

PROC CALIS supports multiple-group multiple-model analysis. You can fit the same covariance (and mean) structure model to several independent groups (data sets). Or, you can fit several different but constrained models to the independent groups (data sets). In PROC CALIS, you can use the GROUP statements to define several independent groups and the MODEL statements to define several different models. For example, the following statements show a multiple-group analysis by PROC CALIS:

```
proc calis;
   group 1 / data=set1;
   group 2 / data=set2;
   group 3 / data=set3;
   model 1 / group=1,2;
      path
         y <--- x   = beta ,
         x <--- z   = gamma;
   model 2 / group=3;
      path
         y <--- x   = beta,
         x <--- z   = alpha;
run;
```

In this specification, you conduct a three-group analysis. You define two PATH models. You fit Model 1 to Groups 1 and 2 and Model 2 to Group 3. The two models are constrained for the y<---x path because they use the same path coefficient parameter beta. Other parameters in the models are not constrained.

To facilitate model specification by model referencing, you can use the REFMODEL statement to specify models based on model referencing. For example, the previous example can be specified equivalently as the following statements:

```
proc calis;
   group 1 / data=set1;
   group 2 / data=set2;
   group 3 / data=set3;
   model 1 / group=1,2;
      path
         y <--- x   = beta ,
         x <--- z   = gamma;
   model 2 / group=3;
      refmodel 1;
      renameparm gamma=alpha;
run;
```

The current specification differs from the preceding specification in the definition of Model 2. In the current specification, Model 2 is making reference to Model 1. Basically, this means that the *explicit* specification in Model 1 is transferred to Model 2. However, the RENAMEPARM statement requests a name change for gamma, which becomes a new parameter named alpha in Model 2. Hence, Model 2 and Model 1 are not the same. They are constrained by the same path coefficient beta for the y<---x path, but they have different path coefficients for the x<---z path.

Model referencing by the REFMODEL statement offers you an efficient and concise way to define models based on the similarities and differences between models. The advantages become more obvious when you have several large models in multiple-group analysis and each model differs just a little bit from each other.

Goodness-of-Fit Statistics

In addition to the chi-square test, there are many other statistics for assessing the goodness of fit of the predicted correlation or covariance matrix to the observed matrix.

Akaike's information criterion (AIC, Akaike 1987) and Schwarz's Bayesian criterion (SBC, Schwarz 1978) are useful for comparing models with different numbers of parameters—the model with the smallest value of AIC or SBC is considered best. Based on both theoretical considerations and various simulation studies, SBC seems to work better, since AIC tends to select models with too many parameters when the sample size is large.

There are many descriptive measures of goodness of fit that are scaled to range approximately from zero to one: the goodness-of-fit index (GFI) and GFI adjusted for degrees of freedom (AGFI) (Jöreskog and Sörbom 1988), centrality (McDonald 1989), and the parsimonious fit index (James, Mulaik, and Brett 1982). Bentler and Bonett (1980) and Bollen (1986) have proposed measures for comparing the goodness of fit of one model with another in a descriptive rather than inferential sense.

The root mean squared error approximation (RMSEA) proposed by Steiger and Lind (1980) does not assume a true model being fitted to the data. It measures the discrepancy between the fitted model and the covariance matrix in the population. For samples, RMSEA and confidence intervals can be estimated. Statistical tests for determining whether the population RMSEAs fall below certain specified values are available (Browne and Cudeck 1993). In the same vein, Browne and Cudeck (1993) propose the expected cross validation index (ECVI), which measures how good a model is for predicting future sample covariances. Point estimate and confidence intervals for ECVI are also developed.

None of these measures of goodness of fit are related to the goodness of prediction of the structural equations. Goodness of fit is assessed by comparing the observed correlation or covariance and mean matrices with the matrices computed from the model and parameter estimates. Goodness of prediction is assessed by comparing the actual values of the endogenous variables with their predicted values, usually in terms of root mean squared error or proportion of variance accounted for (R square). For latent endogenous variables, root mean squared error and R square can be estimated from the fitted model.

Customizable Fit Summary Table

Because there are so many fit indices that PROC CALIS can display and researchers prefer certain sets of fit indices, PROC CALIS enables you to customize the set of fit indices to display. For example, you can use the following statement to limit the set of fit indices to display:

```
fitindex on(only) = [chisq SRMSR RMSEA AIC];
```

With this statement, only the model-fit chi-square, standardized root mean square residual, root mean square error of approximation, and Akaiki's information criterion are displayed in your output. You can also save all your fit index values in an output data file by adding the OUTFIT= option in the FITINDEX statement. This output data file contains all available fit index values even if you have limited the set of fit indices to display in the listing output.

Standardized Solution

In many applications in social and behavioral sciences, measurement scales of variables are arbitrary. Although it should not be viewed as a universal solution, some researchers resort to the standardized solution for interpreting estimation results. PROC CALIS computes the standardized solutions for all models (except for COSAN) automatically. Standard error estimates are also produced for standardized solutions so that you can examine the statistical significance of the standardized estimates too.

However, equality or linear constraints on parameters are almost always set on the unstandardized variables. These parameter constraints are not preserved when the estimation solution is standardized. This would add difficulties in interpreting standardized estimates when your model is defined meaningfully with constraints on the unstandardized variables.

A general recommendation is to make sure your variables are measured on "comparable" scales (it does not necessarily mean that they are mean- and variance-standardized) for the analysis. But what makes different kinds of variables "comparable" is an ongoing philosophical issue.

Some researchers might totally abandon the concept of standardized solutions in structural equation modeling. If you prefer to turn off the standardized solutions in PROC CALIS, you can use the NOSTAND option in the PROC CALIS statement.

Testing Parametric Functions

Oftentimes, researchers might have a priori hypotheses about the parameters in their models. After knowing the model fit is satisfactory, they want to test those hypotheses under the model. PROC CALIS provides two statements for testing these kinds of hypotheses. The TESTFUNC statement enables you to test each parametric function separately, and the SIMTESTS statement enables you to test parametric functions jointly (and separately). For example, assuming that effect1, effect2, effect3, and effect4 are parameters in your model, the following SIMTESTS statement tests the joint hypothesis test1, which consists of two component hypotheses diff_effect and sum_effect:

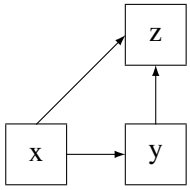
```
SIMTESTS  test1 = (diff_effect sum_ffect);
diff_effect = effect1 - effect2;
sum_effect  = effect3 + effect4;
```

To make test1 well-defined, each of the component hypotheses diff_effect and sum_effect is assumed to be defined as a parametric function by some SAS programming statements. In the specification, diff_effect represents the difference between effect1 and effect2, and sum_effect represents the sum of effect3 and effect4. Hence, the component hypotheses being tested are:

$$H_1: \quad \text{diff\_effect} = \text{effect1} - \text{effect2} = 0$$
$$H_2: \quad \text{sum\_effect} = \text{effect3} + \text{effect4} = 0$$

Effect Analysis

In structural equation modeling, effects from one variable to other variables can be direct or indirect. For example, in the following path diagram x has a direct effect on z in addition to an indirect effect on z via y:



However, y has only a direct effect (but no indirect effect) on z. In cases like this, researchers are interested in computing the total, direct, and indirect effects from x and y to z. You can use the EFFPART option in the PROC CALIS statement to request this kind of effect partitioning in your model. Total, direct, and indirect effects are displayed, together with their standard error estimates. If your output contains standardized results (default), the standardized total, direct, and indirect effects and their standard error estimates are also displayed. With the EFFPART option, effects analysis is applied to all variables (excluding error terms) in your model.

In large models with many variables, researchers might want to analyze the effects only for a handful of variables. In this regard, PROC CALIS provides you a way to do a customized version of effect analysis. For example, the following EFFPART statement requests the effect partitioning of x1 and x2 on y1 and y2, even though there might be many more variables in the model:

```
effpart   x1 x2 ---> y1 y2;
```

See the EFFPART statement of Chapter 26, "The CALIS Procedure," for details.

Model Modifications

When you fit a model and the model fit is not satisfactory, you might want to know what you could do to improve the model. The LM (Lagrange multiplier) tests in PROC CALIS can help you improve the model fit by testing the potential free parameters in the model. To request the LM tests, you can use the MODIFICATION option in the PROC CALIS statement.

The LM test results contain lists of parameters, organized according to their types. In each list, the potential parameter with the greatest model improvement is shown first. Adding these new parameters improves the model fit approximately by the amount of the corresponding LM statistic.

Sometimes, researchers might have a target set of parameters they want to test in the LM tests. PROC CALIS offers a flexible way that you can customize the set the parameters for the LM tests. See the LMTESTS statement for details.

In addition, the Wald statistics produced by PROC CALIS suggest whether any parameters in your model can be dropped (or fixed to zero) without significantly affecting the model fit. You can request the Wald statistics with the MODIFICATION option in the PROC CALIS statement.

Optimization Methods

PROC CALIS uses a variety of nonlinear optimization algorithms for computing parameter estimates. These algorithms are very complicated and do not always work for every data set. PROC CALIS generally informs you when the computations fail, usually by displaying an error message about the iteration limit being exceeded. When this happens, you might be able to correct the problem simply by increasing the iteration limit (MAXITER= and MAXFUNC=). However, it is often more effective to change the optimization method (OMETHOD=) or initial values. For more details, see the section "Use of Optimization Techniques" on page 1268 in Chapter 26, "The CALIS Procedure," and refer to Bollen (1989, pp. 254–256).

PROC CALIS might sometimes converge to a local optimum rather than the global optimum. To gain some protection against local optima, you can run the analysis several times with different initial estimates. The RANDOM= option in the PROC CALIS statement is useful for generating a variety of initial estimates.

Other Commonly Used Options

Other commonly used options in the PROC CALIS statement include the following:

- INMODEL= to input model specification from a data set, usually created by the OUTMODEL= option
- MEANSTR to analyze the mean structures
- NOBS to specify the number of observations
- NOPARMNAME to suppress the printing of parameter names
- NOSE to suppress the display of approximate standard errors
- OUTMODEL= to output model specification and estimation results to an external file for later use (for example, fitting the same model to other data sets)
- RESIDUAL to display residual correlations or covariances

Comparison of the CALIS and FACTOR Procedures for Exploratory Factor Analysis

Both the CALIS and the FACTOR procedures can fit exploratory factor models. However, there are several notable differences:

- By default, PROC FACTOR analyzes the correlation matrix, while PROC CALIS analyzes the covariance matrix.

- PROC FACTOR and PROC CALIS use different parameterizations in the initial factor solution. PROC CALIS uses a lower triangle pattern on the factor loading matrix (a confirmatory factor pattern) in the initial unrotated solution, while PROC FACTOR use certain matrix constraints in the initial unrotated solution. All other things being equal, PROC CALIS and PROC FACTOR might give the same solution after the same factor rotation.
- Because of the way it parameterizes, PROC FACTOR is usually more efficient computationally. PROC CALIS uses a more general algorithm that might not be computationally optimal for exploratory factor analysis.

Comparison of the CALIS and SYSLIN Procedures

The SYSLIN procedure in SAS/ETS software can fit certain kinds of path models and linear structural equation models. PROC CALIS differs from PROC SYSLIN in that PROC CALIS is more general in the use of latent variables in the models. Latent variables are unobserved, hypothetical variables, as distinct from manifest variables, which are the observed data. PROC SYSLIN allows at most one latent variable, the error term, in each equation. PROC CALIS allows several latent variables to appear in an equation—in fact, all the variables in an equation can be latent as long as there are other equations that relate the latent variables to manifest variables.

Both the CALIS and SYSLIN procedures enable you to specify a model as a system of linear equations. When there are several equations, a given variable might be a dependent variable in one equation and an independent variable in other equations. Therefore, additional terminology is needed to describe unambiguously the roles of variables in the system. Variables with values that are determined jointly and simultaneously by the system of equations are called *endogenous variables*. Variables with values that are determined outside the system—that is, in a manner separate from the process described by the system of equations—are called exogenous variables. The purpose of the system of equations is to explain the variation of each endogenous variable in terms of exogenous variables or other endogenous variables or both. Refer to Loehlin (1987, p. 4) for further discussion of endogenous and exogenous variables. In the econometric literature, error and disturbance terms are usually distinguished from exogenous variables, but in systems with more than one latent variable in an equation, the distinction is not always clear.

In PROC SYSLIN, endogenous variables are identified by the ENDOGENOUS statement. In PROC CALIS, endogenous variables are identified by the procedure automatically after you specify the model. With different modeling languages, the identification of endogenous variables by PROC CALIS is done by different sets of rules. For example, when you specify structural equations by using the LINEQS modeling language in PROC CALIS, endogenous variables are assumed to be those that appear on the left-hand sides of the equations; a given variable can appear on the left-hand side of at most one equation. When you specify your model by using the PATH modeling language in PROC CALIS, endogenous variables are those variables pointed to by arrows at least once in the path specifications.

PROC SYSLIN provides many methods of estimation, some of which are applicable only in special cases. For example, ordinary least squares estimates are suitable in certain kinds of systems but might be statistically biased and inconsistent in other kinds. PROC CALIS provides three major methods of estimation that can be used with most models. Both the CALIS and SYSLIN procedures can do maximum likelihood estimation, which PROC CALIS calls ML and PROC SYSLIN calls FIML. PROC SYSLIN can be much faster

than PROC CALIS in those special cases for which it provides computationally efficient estimation methods. However, PROC CALIS has a variety of sophisticated algorithms for maximum likelihood estimation that might be much faster than FIML in PROC SYSLIN.

PROC CALIS can impose a wider variety of constraints on the parameters, including nonlinear constraints, than can PROC SYSLIN. For example, PROC CALIS can constrain error variances or covariances to equal specified constants, or it can constrain two error variances to have a specified ratio.

References

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Bartlett, M. S. (1950), "Tests of Significance in Factor Analysis," *British Journal of Psychology*, 3, 77–85.

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

Bentler, P. M. and Bonett, D. G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.

Bollen, K. A. (1986), "Sample Size and Bentler and Bonett's Nonnormed Fit Index," *Psychometrika*, 51, 375–377.

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

Browne, M. W. and Cudeck, R. (1993), "Alternative Ways of Assessing Model Fit," in K. A. Bollen and S. Long, eds., *Testing Structural Equation Models*, Newbury Park, CA: Sage Publications.

Duncan, O. D., Haller, A. O., and Portes, A. (1968), "Peer Influences on Aspirations: A Reinterpretation," *American Journal of Sociology*, 74, 119–137.

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

Haller, A. O. and Butterworth, C. E. (1960), "Peer Influences on Levels of Occupational and Educational Aspiration," *Social Forces*, 38, 289–295.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986), *Robust Statistics, The Approach Based on Influence Functions*, New York: John Wiley & Sons.

Huber, P. J. (1981), *Robust Statistics*, New York: John Wiley & Sons.

James, L. R., Mulaik, S. A., and Brett, J. M. (1982), *Causal Analysis*, Beverly Hills: Sage Publications.

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.

Jöreskog, K. G. and Sörbom, D. (1979), *Advances in Factor Analysis and Structural Equation Models*, Cambridge, MA: Abt Books.

Jöreskog, K. G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*, Chicago: SPSS.

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

Loehlin, J. C. (1987), *Latent Variable Models*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Lord, F. M. (1957), "A Significance Test for the Hypothesis That Two Variables Measure the Same Trait Except for Errors of Measurement," *Psychometrika*, 22, 207–220.

McArdle, J. J. and McDonald, R. P. (1984), "Some Algebraic Properties of the Reticular Action Model," *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.

McDonald, R. P. (1989), "An Index of Goodness-of-Fit Based on Noncentrality," *Journal of Classification*, 6, 97–103.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Steiger, J. H. and Lind, J. C. (1980), "Statistically Based Tests for the Number of Common Factors," Paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Voss, R. E. (1969), *Response by Corn to NPK Fertilization on Marshall and Monona Soils as Influenced by Management and Meteorological FactorResponse be*, Ph.D. thesis, Iowa Stat University, Ames, IA.

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.

Index

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.