Richard Gonzalez
Psych 614
Version 3.1 (Jan 2023)

LECTURE NOTES #13: Reliability, Structural Equation Modeling, and Hierarchical Modeling

```
And I have felt
A presence that disturbs me with the joy
Of elevated thoughts; a sense sublime
Of something far more deeply interfused,

(William Wordsworth, Lines Composed a Few Miles Above Tintern Abbey)
```

1. Introduction

   These lecture notes present some basic intuitions underlying structural equations modeling (SEM). If you find this technique useful in your research, I suggest that you take a semester-long course on SEM and/or hierarchical linear model (HLM). It turns out that SEM and HLM are essentially the same technique but some things are easier to implement in one versus the other so disciplines tend to favor one over the other. I'll give a general overview in these notes but it won't replace a semester-long course.

2. Elementary Covariance Algebra

   It will be useful to learn how to work with expectations and variances. I introduce some rules for how to manipulate expectation, variance and covariance. These rules will be useful as we develop intuition for structural equations modeling.

   In the following, E refers to expectation, var refers to variance (sometimes just V), and cov refers to covariance (sometimes just C).

   (a) If a is some constant number, then

   $$E(a) = a$$

   "The average of a constant is a constant." Think about a column in a data matrix. If all the numbers in that column are identical, then the mean will be the value of that number.

(b) If a is some constant value real number and X is a random variable with expectation $E(X)$, then

$$E(aX) = aE(X)$$

Multiplying by a constant then averaging is the same as averaging then multiplying the average by the constant.

(c) If a is a constant real number and X is a random variable, then

$$E(X + a) = E(X) + a$$

Adding a constant then averaging is the same as averaging then adding a constant.

(d) If X is a random variable with expectation $E(X)$, and Y is a random variable with expectation $E(Y)$, then

$$E(X + Y) = E(X) + E(Y)$$

The sum of two averages equals the average of two sums.

(e) Given some finite number of random variables, the expectation of the sum of those variables is the sum of their individual expectations. Thus,

$$E(X + Y + Z) = E(X) + E(Y) + E(Z)$$

(f) If a is some constant real number, and if X is a random variable with expectation $E(X)$ and variance σ^2, then the random variable $(X + a)$ has variance σ^2. In symbols,

$$\text{var(X)} = \text{var}(X \ + \ a)$$

(g) If a is some constant real number, and if X is a random variable with variance σ^2, the variance of the random variable aX is

$$\text{var}(aX) = a^2\sigma^2$$

(h) If X and Y are independent random variables, with variances σ^2_{x} and σ^2_{y} respectively, then the variance of the sum $X + Y$ is

$$\sigma^2_{(X+Y)} = \sigma^2_{\text{x}} + \sigma^2_{\text{y}}$$

Similarly, the variance of $X - Y$ when both are independent is

$$\sigma^2_{(X-Y)} \quad = \quad \sigma^2_{\text{x}} + \sigma^2_{\text{y}}$$

(i) Given random variable X with expectation $E(X)$ and the random variable Y with expectation $E(Y)$, then if X and Y are independent,

$$E(XY) \quad = \quad E(X)E(Y)$$

The implication does not go in the other direction. If $E(XY) \quad = \quad E(X)E(Y)$, that does not imply that the two variables are independent.

(j) If $E(XY) \neq E(X)E(Y)$, the variables X and Y are not independent.

This statement is simply the contrapositive of the previous rule.

(k) Given the random variable X with expectation $E(X)$ and the random variable Y with expectation $E(Y)$, then the covariance of X and Y is

$$\text{cov}(X, Y) \quad = \quad E(XY) - E(X)E(Y)$$

(l) Another definition of covariance is

$$\text{cov}(X, Y) \quad = \quad E[(X - \mu_X)(Y - \mu_Y)]$$

(m) If a is a constant and X is a random variable, then the covariance is

$$\text{cov}(X, a) \quad = \quad 0$$

“The covariance of a random variable with a constant is zero.”

(n) If a is a constant and the variables X and Y are random, then

$$\text{cov}(aX, Y) \quad = \quad a\text{cov}(X, Y)$$

(o) The variance of a random variable X is equal to the covariance of X with itself, i.e.,

$$\text{cov}(X, X) \quad = \quad \text{var}(X)$$

(p) The sum operation between random variables is distributive for covariances , i.e.,

$$\text{cov}(X, Y + Z) \;=\; \text{cov}(X, Y) + \text{cov}(X, Z)$$

(q) Now that we know about covariances, we can return to the definition of the sum of two random variables. I will now relax the restriction of independence. If X and Y are random variables, with variances σ^2_{x} and σ^2_{y} respectively, then the variance of the sum $X + Y$ is

$$\sigma^2_{(X+Y)} \;=\; \sigma^2_{\text{x}} + \sigma^2_{\text{y}} + 2\text{cov}(X, Y)$$

Similarly, the variance of $X - Y$ is

$$\sigma^2_{(X-Y)} \;=\; \sigma^2_{\text{x}} + \sigma^2_{\text{y}} - 2\text{cov}(X, Y)$$

(r) Definition of the *correlation*: Given two random variables X and Y, then the covariance divided by the standard deviation of each variable is the correlation ρ,

$$\rho \;=\; \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

Note that the correlation is simply a normalized covariance.

The following results are useful for interactions:

(s) The expectation of the product of two variables is

$$E(XY) \;=\; C(X, Y) + E(X)E(Y)$$

(t) The variance of the product of two variables that are normally distributed is (using C for cov and V for var)

$$\begin{aligned} V(XY) \;=\;& E(X)^2V(Y) + E(Y)^2V(X) + \\ & 2E(X)E(Y)C(X, Y) + \\ & V(X)V(Y) + C(X, Y)^2 \end{aligned}$$

(u) The covariance of two products with all variables distributed multivariate normal is

$$\begin{aligned} C(XY, UV) &= E(X)E(U)C(Y,V) + \\ &\quad E(X)E(V)C(Y,U) + \\ &\quad E(Y)E(U)C(X,V) + \\ &\quad E(Y)E(V)C(X,U) + \\ &\quad C(X,U)C(Y,V) + C(X,V)C(Y,U) \end{aligned}$$

3. Covariance Matrices

Suppose you have three variables. There will be three variances (one for each variable) and three covariances (one for each possible pairwise combination). Making sense of these six numbers is facilitated if they are arranged in a matrix such as

	variable 1	variable 2	variable 3
variable 1	var(1)	cov(1,2)	cov(1,3)
variable 2	cov(2,1)	var(2)	cov(2,3)
variable 3	cov(3,1)	cov(3,2)	var(3)

Because cov(x,y)=cov(y,x) (i.e., covariances are symmetric) the lower triangle of the above matrix mirrors the upper triangle[1]. This property is called symmetry and a matrix that has this property is called symmetric.

4. Correlation Matrices

Because a correlation is defined as

$$\text{cor(x,y)} = \frac{\text{cov(x,y)}}{\sqrt{\text{var(x)var(y)}}} \tag{13-1}$$

all information necessary to compute a correlation matrix is present in the covariance matrix.

The elements along the diagonal of a correlation matrix will equal one. The correlation matrix is also symmetric. Note that while it is easy to convert a covariance matrix into a correlation matrix, the conversion will not be possible going the other way (correlation matrix to covariance matrix) unless the variances of each variable are known.

A convenient way to represent both the covariance matrix and the correlation matrix is to display one matrix that has variances in the diagonal, covariances in the upper triangle, and correlations in the lower triangle. That is,

[1] Triangles are formed on either side of the main diagonal (which starts at the upper left and moves to the lower right).

	variable 1	variable 2	variable 3
variable 1	var(1)	cov(1,2)	cov(1,3)
variable 2	cor(2,1)	var(2)	cov(2,3)
variable 3	cor(3,1)	cor(3,2)	var(3)

Such a combined matrix is only for display (usually in published papers) and not for computation.

As we have seen already, these matrices are the building blocks of most multivariate statistical techniques including principal components, factor analysis, discriminant analysis, multivariate analysis of variance, canonical correlation, and structural equations modeling. Some problems are easier to deal with in the form of a covariance matrix and other problems are easier to deal with in the form of a correlation matrix.

5. Using these covariance rules to create new things: testing two dependent variances

How do you test two variances on variables that come from the same subjects (i.e. are dependent)? This is analogous to the paired-t test for means but applied to variances instead. This test is not easy to find in textbooks or computer programs, but it is straightforward to derive now that we know the covariance algebra rules.

The correlation between the sum and the difference of the two variables is identical to the difference of the two dependent variances. That is,

$$\text{cov}(X - Y, X + Y) = \text{var}(X) - \text{var}(Y)$$

One gets an equation like this by working backwards from the desired term (like the difference between two dependent variances) and seeing if there is a way to get that term through another way. To see this, take the covariance term on the left hand side, apply the distributive rule and apply the definition that the covariance of a variable with itself equals the variance. The difference of two variances automatically falls out.

We can see that testing if the covariance (or the correlation) is zero is equivalent to testing whether the difference in the dependent variances is equal to 0. So, to test whether two dependent variances are equal simply test if the correlation between the sum and difference of X and Y is different from zero (see LN#6 for testing a correlation).

6. Making a connection between a correlation matrix and multiple regression

It is instructive to see how a correlation matrix enters in techniques you already know. Here we will show the connection between a correlation matrix and regression. The "beta coefficients" for each variable in a regression can be computed from the correlation matrix of the variables involved. The “beta coefficients” are the regression weights when all variables (predictors and criterion) are standardized. In such a model there is no intercept. The following discussion is adapted from Edwards (1985).

Let R be the correlation matrix of only the predictors, b be the vector of “standardized beta coefficients”[2], and r is the vector of correlations between each predictor and the criterion. Multiple regression satisfies the following equation:

$$\mathrm{Rb} = \mathrm{r} \tag{13-2}$$

How one multiplies matrices is not important for our purposes (though see the appendix in LN11 for a refresher). All you need to know now is that by “multiplying” R and b we can compute r, the correlation of the criterion with each predictor. But, usually we know r and R and not b; the task is to estimate b. So, think back to middle school. The unknown is b so “divide” both sides of the equation by R to solve for b. Division is computationally a little more difficult for matrices, but the intuition is the same.

The correlation matrix R contains the information of how the predictors correlate with each other and the vector r constrains the information of how each predictor correlates with the dependent variable. We observe both matrix R and vector r but we need to estimate vector b (the standardized beta coefficients) from the data.

Back to the regression problem. Look at Equation 13-2—you’ll see that all we need to do is multiply both sides by the inverse of R.

$$\begin{aligned}
\mathrm{Rb} &= \mathrm{r} && \text{(13-3)}\\
\mathrm{R}^{-1}\mathrm{Rb} &= \mathrm{R}^{-1}\mathrm{r} && \text{(13-4)}\\
\mathrm{b} &= \mathrm{R}^{-1}\mathrm{r} && \text{(13-5)}
\end{aligned}$$

This shows at all we need to compute the standardized beta coefficients from the regression are the correlations between predictors (matrix R) and the correlations between each predictor and the dependent variable (vector r)

Further, with knowledge of the variances we can transform the “standardized beta coefficients” computed through this method to “raw score regression coefficients”, i.e., the regression coefficients that result when the variables are *not* standardized. All one needs to do is multiply the “beta coefficients” by the ratio of the standard deviation of the criterion and the standard deviation of the predictor associated with the slope. That is,

$$\text{raw score reg coef for } \mathrm{x}_i = \text{“beta” for } \mathrm{x}_i \frac{\mathrm{sd(y)}}{\mathrm{sd}(\mathrm{x}_i)} \tag{13-6}$$

[2] For our purposes a vector can be treated as a matrix with one column—these vectors are, not surprisingly, called column vectors.

This section showed that the variance and correlation matrices contain all the information necessary to compute slopes in a regression. This demonstrates that these constructs (means, variances, and covariances) are fundamental to statistics. In other words, once we have the covariance matrix we no longer need the original data to compute regressions. Of course, having the original data is helpful because we can check for outliers, check assumptions, etc., which we cannot do if we only have the means and covariances. The covariance matrix is analogous to the "distance matrix" between cities we studied in multidimensional scaling in LN10.

We now make use of these concepts to develop some intuition to a set of analyses including reliability, mediation and systems of linear equations (aka structural equation modeling).

As in MDS (LN10), we have an observed similarity matrix and a model-implied covariance matrix. A set of regression equations that define regression, factor models, growth curves, etc., is what defines the model-implied covariance matrix. We can extend the diagram I introduced in Lecture Notes 10 in MDS to this setting (the new square bracket object is in the lower right, the one with a list of structural models):

$$\begin{tikzcd}[ampersand replacement=\&]
\text{Observed Distance Matrix} \& \text{Model-Implied Distance Matrix} \\
\left[\begin{matrix} d_{11} \& \dots \& d_{1k} \\ \vdots \& \ddots \& \vdots \\ dk1 \& \& d_{kk} \end{matrix}\right] \arrow[r, leftrightarrow] \& \left[\begin{matrix} \hat{d}_{11} \& \dots \& \hat{d}_{1k} \\ \vdots \& \ddots \& \vdots \\ \hat{d}k1 \& \& \hat{d}_{kk} \end{matrix}\right] \arrow[d, leftrightarrow] \\
 \& \left[\begin{matrix} y^1 \& = \& \beta_0^1 + \beta_1^1 X1 + \beta_2^1 X2 + \epsilon^1 \\ y^2 \& = \& \beta_0^2 + \beta_1^2 X1 + \beta_2^2 X2 + \beta_3^2 X3 + \epsilon^2 \\ y^3 \& = \& \beta_0^3 + \beta_1^3 y^2 + \epsilon^3 \end{matrix}\right]
\end{tikzcd}$$

A set of regression equations defines a model-implied covariance matrix and we compare the observed covariance matrix to the model-implied covariance matrix. The parameters of the regression equations are found by minimizing the discrepancy between the two covariance matrices. In this way, we estimate a system of equations. The technique called Structural Equations Modeling (SEM) is simply a way to run multiple structural models simultaneously, and surprisingly this has PCA, factor analysis, canonical correlation, as well as regression and ANOVA, as special cases.

Let's move on to another topic, reliability, that can easily be modeled in SEM.

7. Reliability

What is reliability? The standard definition uses the simple additive model

$$X = T + \epsilon \qquad (13\text{-}7)$$

where X is the observed score, T is the true score, and ϵ is measurement error. Think of a bathroom scale—you have a true weight T but you observe a reading X that includes both your true weight and any measurement error from the scale. The measurement error could be positive or negative; it behaves in a similar manner to the ϵ in regression/ANOVA but there is a slight difference in interpretation. The ϵ in regression/ANOVA assesses the prediction deviation, that is, the discrepancy between the weighted sum of the predictors and intercept and the observed dependent variable. However, the ϵ in reliability is interpreted as measurement error. We can have both types of error in the same problem. For example, in a regression a predictor might be measured with error (error in predictor) and there is also the discrepancy in prediction (i.e., the usual regression ϵ, which is the difference between the observed dependent variable Y and the predicted Y). We'll come back to this point later in these notes.

The concept of reliability is based on variances. The reliability of variable X is defined as

$$\begin{aligned} \text{reliability of X} &= \frac{\text{var(T)}}{\text{var(T) + var}(\epsilon)} && (13\text{-}8) \\ &= \frac{\text{var(T)}}{\text{var(X)}} && (13\text{-}9) \end{aligned}$$

In words, reliability is defined as a proportion of observed variance that is true variance. Reliability is interpreted as a proportion—reliability cannot be negative.

But in practice we don't know the true score T (nor the error ϵ) so what can we do? How do we separate the true score from the observed score? One approach is a test-retest paradigm[3]. Test the person on the same variable twice. Assuming no changes over time in the true score[4], time 1 is influenced by the same "latent variable" as time 2. The model assumes that the only difference between the two time periods is the error, which I'll denote ϵ_1 and ϵ_2.

So, we have $X_1 = T + \epsilon_1$ and $X_2 = T + \epsilon_2$.[5] We could get a handle on the true score T by examining the covariance between the two observed variables X_1 and X_2. The true component T is what both X_1 and X_2 have in common. We need to assume that the two errors are

[3] There are many other approaches that I will not cover here.

[4] This is an important assumption. There cannot be change in the intervening time period, so if you expect "growth" or change to occur between the first and second administration, then this definition of reliability is not so useful.

[5] Sometimes the role of true score and noise can flip. For example, in some areas of signal processing the T represents the noise and ϵs represent the signal. If one is recording muscle activity using surface EMG, the electrodes also pick up background noise like the "humming" of the fluorescent lights. If there are recordings of two muscle contractions, the humm is the same in both recordings but the signals may be slightly different due to differences in muscle contraction. By subtracting data from recording 1 and recording 2 one can cancel the background hum noise (the part of the signal that is the same at both times) from the part of the signal we care about, which may be different at both times: (hum + sig1) - (hum + sig2) = sig1 - sig2. This contrasts with the true score idea where we correlate in order to "cancel" the noise ϵ and examine the remaining "true" score.

independent from T and are not correlated with each other (you will see how that assumption simplifies a complicated expression).

$$\begin{aligned}
\text{cov}(X_1, X_2) &= \text{cov}(T + \epsilon_1, T + \epsilon_2) \\
&= \text{cov}(T, T) + \text{cov}(T, \epsilon_1) + \text{cov}(T, \epsilon_2) + \text{cov}(\epsilon_1, \epsilon_2) \\
&= \text{cov}(T, T) \ \text{ under independence} \\
&= \text{var}(T)
\end{aligned}$$

Recall that the correlation between two variables is defined as the covariance divided by the product of the standard deviations.

$$\text{cor} \;=\; \frac{\text{cov}(X, Y)}{\text{sd}(X)\text{sd}(Y)} \tag{13-10}$$

To compute the correlation of X_1 and X_2 we just need the covariance (computed above) and the standard deviations.

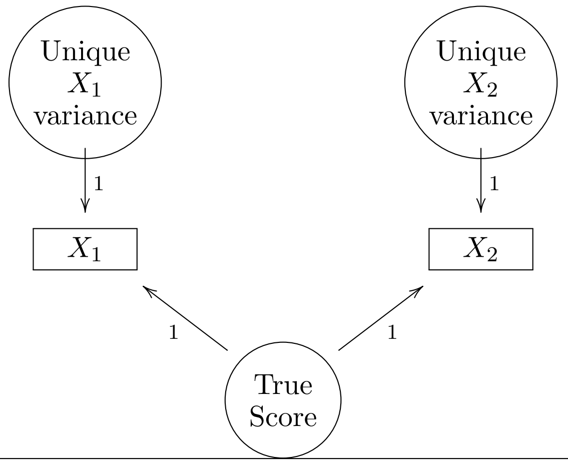
$$\begin{aligned}
\text{cor}(X_1, X_2) &= \frac{\text{cov}(X_1, X_2)}{\sqrt{\text{var}(X_1)\text{var}(X_2)}} \\
&= \frac{\text{var}(T)}{\sqrt{\text{var}(T + \epsilon_1)\text{var}(T + \epsilon_2)}} \\
&= \frac{\text{var}(T)}{\text{var}(T + \epsilon)} \ \text{ assume var of } \epsilon\text{s are equal \& ind} \\
&= \text{reliability}
\end{aligned}$$

Thus, a correlation of a test-retest variable is equivalent to reliability. For this reason, test-retest reliability is usually denoted r_{xx}. This is a case where a raw correlation can be interpreted as a proportion of true variance to observed variance—there is no need to square a correlation in a test-retest situation when it is interpreted as a reliability. In fact, taking the square root of the test-retest reliability leads to an estimate of the correlation between true score and observed score.

This same logic extends to the true correlation between two variables, each measured with their own error. For example, suppose variables X and Y are each measured with error. The correlation between the "true" component of X and the "true" component of Y is equal to:

$$\text{r}_{\text{X}_\text{T}\text{Y}_\text{T}} = \frac{\text{r}_\text{XY}}{\sqrt{\text{r}_\text{XX}\text{r}_\text{YY}}} \tag{13-11}$$

Figure 13-1: Test-retest structural model.



This equation is important in test theory. It shows that a true correlation is equal to the observed correlation (the numerator in the right hand side) divided by the product of the square roots of the reliability coefficients. So, the product of the reliabilities serves to correct the observed correlation between the two variables. As long as both reliabilities are non-zero and at least one is also not equal to 1, then the correction will always be to increase the value of the observed correlation between X and Y. The correlation in Equation 13-11 is called the **"correction for attenuation."**

A few words on latent variables. Latent variables seem to rub people the wrong way at first. It seems we should only base empirical science on what we can directly observe. Actually, there are many examples in science of latent variables (constructs we don't directly observe). One recent example is the black hole in astronomy. A black hole cannot be observed directly, but the presence of a black hole can be observed indirectly, such as the pattern light makes when it travels near a black hole (e.g., see the link black holes. This serves as an analogy for how latent variables are used in structural equation modeling. One has a set of observations, or indicators, and they jointly determine the unobserved, latent variable. The latent variables can be used to make testable predictions. Latent variables are used throughout science; they are not something limited to social science or statistics or, more specifically, to structural equations modeling.

8. Spearman-Brown (SB) Formula

For a complete derivation of the SB formula see Nunnally's *Psychometric Theory*.

The logic of the SB formula is to estimate the reliability of a k item test from knowledge of the reliability of individual items. For example, suppose I have a 5 item questionnaire and I know the intercorrelations of each item with the other items. However, I do not know (without further computation) how a person's score on this 5 item test (i.e., the sum over the 5 items) would correlate with their summed score on another 5 items (where the items are sampled from the same domain), or how the sum of these five items today would correlate with the same five items tomorrow. In other words, we would like to know the reliability of the five item test as a total score, but all we have are the individual reliabilities of each item.

We denote the correlation of an item with another item as r_{11} (some books use r_{ij} to denote item i with item j). The correlation of a sum of k items with another sum of k items is denoted r_{kk}.

Now comes the famous SB formula. To estimate r_{kk} we take

$$r_{kk} = \frac{kr_{11}}{1 + (k-1)r_{11}}$$

Thus, from the inter-item correlations between k items we estimate the reliability of the sum of k items. This form is known as the standardized version because it is based on correlations. There is also an unstandardized version that I present below. SPSS and R compute both.

The limit of SB as k goes to infinity (an infinitely large test) is 1, which represents perfect reliability. This means that as the number of items in a scale grows, the reliability will increase as well.

To study the behavior of SB we can use a simple R function, which produces Table 1. The columns show different values of k (number of items) and the rows show different values of r11 (item correlation). The entries in each cell are the SB values. Note how as the number of items increases, the SB increases as well, so even with very low item-level reliabilities (.1) the SB reaches .74 when there are 25 items.

```
library(xtable)
SB <- function(r11, k) {
   k*r11/(1+(k-1)*r11)
}

rvals <- c(.1,.2,.3,.4,.5)
kvals <- c(2,3,4,5,10,25)
SBvalues <- matrix(0,length(rvals),length(kvals))
for (i in 1:length(rvals))
  for (j in 1:length(kvals))
    SBvalues[i,j] <- SB(rvals[i],kvals[j])
rownames(SBvalues) <- as.character(rvals)
colnames(SBvalues) <- as.character(kvals)
```

```
print(xtable(SBvalues,digits=2, table.placement="!h",
  caption.placement="top", label="tab:sb",
  caption=

"Spearman-Brown for number of items (cols) and correlations (rows)"))
```

	2	3	4	5	10	25
0.1	0.18	0.25	0.31	0.36	0.53	0.74
0.2	0.33	0.43	0.50	0.56	0.71	0.86
0.3	0.46	0.56	0.63	0.68	0.81	0.91
0.4	0.57	0.67	0.73	0.77	0.87	0.94
0.5	0.67	0.75	0.80	0.83	0.91	0.96

Table 1: Spearman-Brown for number of items (cols) and correlations (rows)

9. Cronbach's α

What do you do if you have many different inter-item correlations, so there are several estimates of r$_{11}$? You want to estimate a single r$_{kk}$ using Spearman-Brown, but how do you choose among the different observed r$_{11}$s?

Nothing deep here. Simply take the average correlation, apply S-B to the average, and you get what's known as the standardized Cronbach's α. If you had thought of that, then everyone would be saying YOURNAMEGOESHERE's α.

So, to estimate r$_{11}$ from k items compute all possible correlations and *average* (yes, average) the correlations to estimate a singe r$_{11}$. In symbols,

$$r_{11} = \frac{\sum_{ij} r_{ij}}{\binom{k}{2}}$$

where $\binom{k}{2}$ is "k choose 2" (the total number of correlations that went into the average).

Let's do this example from Nunnally & Bernstein. Imagine we have 3 judges who rate 8 subjects. We want to know the reliability of these three judges. One way to frame this question is how much do these three judges intercorrelate (here just take the average inter-item correlation). Another way to frame this question is to ask what is the reliability of the sum of the three judges (i.e., if these judges were to rate another eight subjects and we summed the three judge's scores, what would be the correlation of the original eight sums with the new eight sums?). This latter question is answered by applying the SB formula on the average inter-item correlation (which is called Cronbach's α).

Example from data in Nunnally & Bernstein.

Items	Judges		
1	1	1	1
2	1	1	1
3	0	0	0
4	0	1	0
5	0	0	0
6	0	1	1
7	0	0	0
8	0	1	1

The SPSS syntax for this problem is

```
data list free / j1 j2 j3.

begin data
1 1 1
1 1 1
0 0 0
0 1 0
0 0 0
0 1 1
0 0 0
0 1 1
end data.

reliabilities  variables = j1 j2 j3
 /statistics all.
```

The intercorrelations of these judges are

	J1	J2	J3
J1	1		
J2	.4472	1	
J3	.5774	.7746	1

If you take the average correlation and apply SB, then you get the standardized alpha. In this example the average correlation r is .5997, apply SB to that and you get .8180. SPSS reports a "standardized item alpha" = .8180, which is the same thing!

Some textbooks report different ways of computing alpha. For example, using the covariance matrix (so this is the nonstandardized form of alpha)

	J1	J2	J3
J1	.2143		
J2	.1071	.2679	
J3	.1429	.2143	.2857

Sum of all elements in the covariance matrix (9 terms) is 1.6965. The sum of the diagonal elements is .7679. Cronbach's alpha is given by (Nunnally & Bernstein 6-26)

$$\begin{aligned}\alpha &= \frac{k}{k-1}\left(1-\frac{\Sigma\sigma_a^2}{\sigma_y^2}\right) && (13\text{-}12)\\ &= \frac{3}{2}\left(\frac{1.6965-.7679}{1.6965}\right) && (13\text{-}13)\\ &= .821 && (13\text{-}14)\end{aligned}$$

This is the unstandardized form of Cronbach's α.

A completely different, but equivalent, way to compute Cronbach's α is through the intraclass correlation and then apply the SB formula. To compute the intraclass, first compute the ANOVA source table using items and judge as two factors (no interaction term is fitted because there is only one observation per cell as in the randomized block design from LN4).

Source table (k=3 judges):

	MS	SS	df
items	.5655	3.958	7
judge	.292	.583	2
residual	.1012	1.417	14
total		5.958	23

The ANOVA intraclass is .6046. This is an index of similarity of one judge to the next; it is not the average inter-judge correlation. The equation for the ANOVA intraclass is

$$\text{intraclass} = \frac{\text{MSB-MSW}}{\text{MSB + (k-1)MSW}} \qquad (13\text{-}15)$$

$$
\begin{aligned}
&= \frac{.5655 - .1012}{.5655 + (2).1012} && \text{(13-16)}\\
&= .6046 && \text{(13-17)}
\end{aligned}
$$

If you apply Spearman-Brown to this intraclass

$$
\begin{aligned}
r_{kk} &= \frac{kr_{11}}{1 + (k-1)r_{11}} && \text{(13-18)}\\
&= \frac{3(.6046)}{1 + 2(.6046)} && \text{(13-19)}\\
&= .821 && \text{(13-20)}
\end{aligned}
$$

This last number is interpreted in terms of how this set of 3 judges will correlate to another set of 3. Again, the same number as before, the unstandardized Cronbach's α.

With a little algebra both steps (the intraclass and the SB formula) can be condensed into a single expression:

$$\frac{\text{MSB-MSW}}{\text{MSB}} \qquad \text{(13-21)}$$

This is what appears on page 277 Eq 7-20 in Nunnally & Bernstein. Cronbach's α is identical to applying the SB formula to the intraclass correlation, and the simple equation in these notes Eq 13-21 shows that the relevant info is embedded in ANOVA. So ANOVA also shows up in reliability theory.

It may seem strange to use the sum of squares (SS) for items rather than SS for judges (aka raters). Judges are conceptualized as a blocking factor so we want to remove from the error term any effect due to mean differences between the judges. So, lets suppose the judges perfectly agree with each other. Then all three scores for each item will be the same, there won't be any within item variance (so MSW goes to 0) and all the remaining variance is between items (MSB). In this extreme case when MSW = 0 the intraclass will be equal to 1. The other extreme is that all variance is within items (i.e., all three judges always disagree with each other), so MSB=0 and the intraclass in Equation 13-15 goes to -1/(k-1). The intraclass correlation is asymmetric as it only approaches 0 when k is very large.

I'll show two different ways to compute Cronbach alpha in R. One way to compute alpha in R is through the alpha command in the psych package. It prints both standardized and unstandardized (called raw in the output) versions and provides additional information such as a 95% CI around alpha, some descriptive statistics and information on missing data.

```
library(psych)

# define rater data
rater.data <- matrix(c(1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
    0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1), ncol = 3, byrow = T)
colnames(rater.data) <- c("j1", "j2", "j3")

# print covariance matrix
round(cov(rater.data), 3)
```

```
##       j1    j2    j3
## j1 0.214 0.107 0.143
## j2 0.107 0.268 0.214
## j3 0.143 0.214 0.286
```

```
# EXAMPLE of alpha using psych package
alpha(rater.data)
```

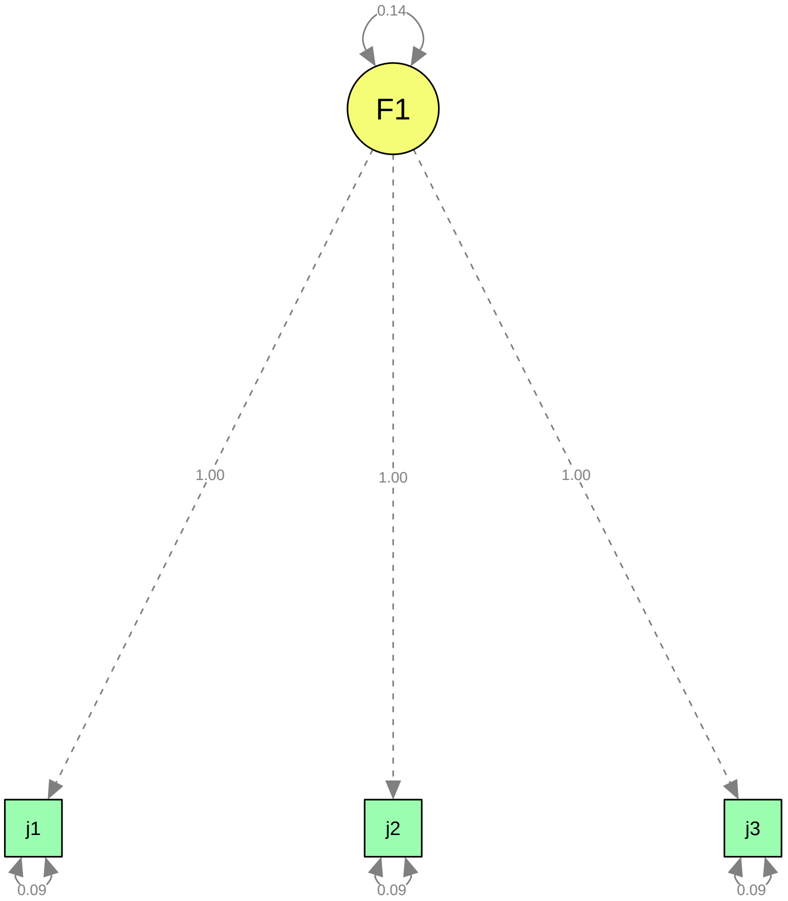
```
##
## Reliability analysis
## Call: alpha(x = rater.data)
##
##   raw_alpha std.alpha G6(smc) average_r S/N  ase mean   sd
##       0.82      0.82    0.79       0.6 4.5 0.11 0.46 0.43
##  median_r
##      0.58
##
##     95% confidence boundaries
##          lower alpha upper
## Feldt     0.40  0.82  0.96
## Duhachek  0.61  0.82  1.03
##
##  Reliability if an item is dropped:
##    raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r
## j1      0.87      0.87    0.77      0.77 6.9     0.09    NA
## j2      0.73      0.73    0.58      0.58 2.7     0.19    NA
## j3      0.62      0.62    0.45      0.45 1.6     0.27    NA
##    med.r
## j1  0.77
## j2  0.58
## j3  0.45
##
##  Item statistics
##    n raw.r std.r r.cor r.drop mean   sd
## j1 8  0.77  0.79  0.60   0.54 0.25 0.46
## j2 8  0.87  0.86  0.80   0.70 0.62 0.52
## j3 8  0.92  0.92  0.89   0.80 0.50 0.53
##
## Non missing response frequency for each item
##       0    1 miss
## j1 0.75 0.25    0
## j2 0.38 0.62    0
## j3 0.50 0.50    0
```

Another way to compute alpha is through an SEM approach using the lavaan package. This requires understanding of SEM, which we will cover later in these notes, but I will go ahead and show it now. This model estimates the factor loadings on a single factor, where those factor loadings are set to be equal to each other, and the variances of the observed variables j are also set equal to each other. This model has 3 variances and 3 covariances as input (the covariance matrix) and estimates two parameters (equal coefficient to create the factor and the factor variance). Think a kind of factor model (from LN11) where the elements of the first eigenvector are equal and the errors (ϵs) are also estimated and set equal to each other for a kind of homogeneity of variance across variables assumption.

The Rsquared from this SEM, where the loadings are forced to be equal and residual variances are forced to be equal, is equivalent to the intraclass correlation. Just apply S-B to this intraclass of .605 and you get Cronbach's alpha.

```
library(lavaan)
library(semPlot)
library(semTools)

# EXAMPLE of alpha using lavaan (SEM framework) factor
# loading is set to 1 & error variances set equal to each
# other
model.rater <- "F1 =~ l1*j1 + l1*j2 + l1*j3
j1 ~~ v1*j1
j2 ~~ v1*j2
j3 ~~ v1*j3
"
out.cfa <- cfa(model.rater, data = rater.data, auto.fix.first = F)
summary(out.cfa, rsquare = T, standardized = T)



## lavaan 0.6.13 ended normally after 9 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                          4
##   Number of equality constraints                      2
##
##   Number of observations                              8
##
## Model Test User Model:
##
##   Test statistic                                  2.245
##   Degrees of freedom                                  4
##   P-value (Chi-square)                            0.691
##
## Parameter Estimates:
##
##   Standard errors                              Standard
##   Information                                  Expected
##   Information saturated (h1) model           Structured
##
## Latent Variables:
##                      Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     j1        (l1)    1.000
##     j2        (l1)    1.000
##     j3        (l1)    1.000
##    Std.lv  Std.all
##
##     0.368    0.778
##     0.368    0.778
##     0.368    0.778
##
## Variances:
##                      Estimate  Std.Err  z-value  P(>|z|)
##    .j1        (v1)    0.089    0.031    2.828    0.005
##    .j2        (v1)    0.089    0.031    2.828    0.005
##    .j3        (v1)    0.089    0.031    2.828    0.005
##     F1                0.135    0.083    1.629    0.103
##    Std.lv  Std.all
##     0.089    0.395
##     0.089    0.395
##     0.089    0.395
##     1.000    1.000
##
## R-Square:
##                      Estimate
##     j1                0.605
##     j2                   0.605
##     j3                   0.605
```

```
semPaths(out.cfa, whatLabels = "est", color = list(lat = rgb(245,
    253, 118, maxColorValue = 255), man = rgb(155, 253, 175,
    maxColorValue = 255)))
```



The R-square printed here is .605, so apply the SB formula and you get Cronbach's α of .82 as we saw before.

Or you can make use of the reliabilities command in the SemTools package and get Cronbach's α directly from the output of the earlier lavaan call.

```
reliability(out.cfa)

##               F1
## alpha  0.8210526
## omega  0.8210526
## omega2 0.8210526
## omega3 0.8210526
## avevar 0.6046512
```

This output also mentions other measures of reliability such as three different forms of a measure called omega (ω). For this model the three omega measures are equivalent to Cronbach's alpha but omega works for more general models that do not impose the equality restrictions on both the loadings and the error variances I used to show alpha. Below is a more general model for which the loadings and the error variances are not assumed to be equal. Note that Cronbach's alpha remains the same because it is estimated relative to the more constrained model we used earlier, but the three omega's are now different from Cronbach's α. For a recent description of omega see Flora (2020) in the journal *Advances in Methods and Practices in Psychological Science*. As we have seen throughout this year, the assumptions you want to make can have major implications on the results of your modeling.

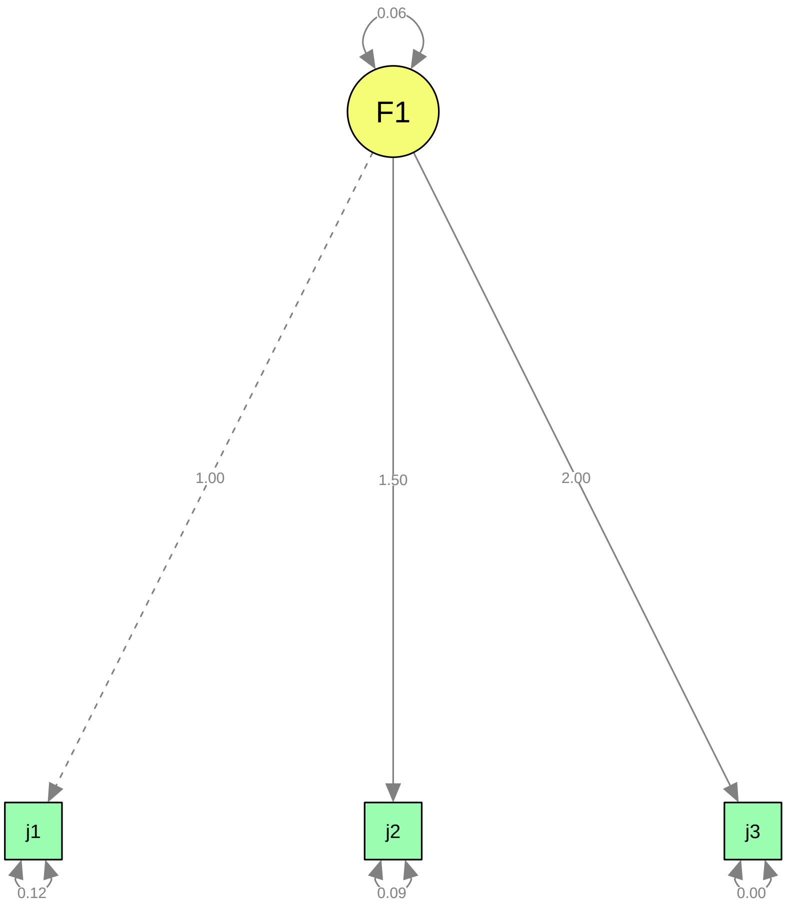
```
model.rater2 <- "F1 =~ j1 + j2 + j3"
out.cfa.free <- cfa(model.rater2, data = rater.data, auto.fix.first = F)
summary(out.cfa.free, rsquare = T, standardized = T)

## lavaan 0.6.13 ended normally after 22 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         6
##
##   Number of observations                             8
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
```

```
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     j1                 1.000
##     j2                 1.500    0.866    1.732    0.083
##     j3                 2.000    1.291    1.549    0.121
##    Std.lv  Std.all
##
##     0.250    0.577
##     0.375    0.775
##     0.500    1.000
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .j1                 0.125    0.068    1.852    0.064
##    .j2                 0.094    0.074    1.265    0.206
##    .j3                 0.000    0.102    0.000    1.000
##     F1                 0.063    0.074    0.840    0.401
##    Std.lv  Std.all
##     0.125    0.667
##     0.094    0.400
##     0.000    0.000
##     1.000    1.000
##
## R-Square:
##                     Estimate
##     j1                 0.333
##     j2                 0.600
##     j3                 1.000
```

```
reliability(out.cfa.free)
```

```
##               F1
## alpha  0.8210526
## omega  0.8526316
## omega2 0.8526316
## omega3 0.8526316
## avevar 0.6744186
```

```
semPaths(out.cfa.free, whatLabels = "est", color = list(lat = rgb(245,
    253, 118, maxColorValue = 255), man = rgb(155, 253, 175,
    maxColorValue = 255)))
```



I'll cover more of the SEM logic later in these lecture notes. For now, note how similar this is to the factor model approach in LN11. We'll see soon that SEM also encompasses regression and ANOVA so it becomes the mother of all statistical methods covered across these two semesters.

10. Reliability & Regression with One Predictor

How does measurement error influence a linear regression with one predictor? When the dependent variable has measurement error there is no serious problem because that is exactly the kind of error regression is designed to deal with (but there will be loss of statistical power). For instance, take the usual

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{13-22}$$

The ϵ term captures both measurement error on Y and the prediction error of the equation. A simple regression cannot distinguish measurement noise in the criterion from noise in the prediction. The "true component" of Y is taken to be a linear transformation of the variable X. Of course, measurement error on Y reduces the power of the tests (e.g., increases the size of the confidence interval around β_1) but does not introduce bias. The requirement is that the error be independent from all the effects in the model. Violations of these assumptions can be dealt with easily (e.g., if errors are correlated across observations, as in a time series, then there exist techniques to take into account the "autocorrelation").

There is no provision in the standard regression model for measurement error on X. More generally, there is no provision in the standard regression model for measurement error on any predictor variable. But frequently regressions involve predictors that contain measurement error (e.g., the total score the self-esteem scale might be used as a predictor, the behavioral coding of a rat's licking behavior may have error in measurement, etc). It turns out that measurement error on X not only reduces power, but it also introduces bias. The problem associated with bias are, in general, more serious than reduction in power. Thus, the inability for regression to handle measurement error in the predictor(s) is a very serious limitation.

We now turn to an examination of the effects of having a "noisy" predictor in a simple regression equation with just one predictor; this discussion is adapted from Neter, Wasserman, and Kutner, (1985). The only machinery we need to reach a better understanding is covariance algebra. Consider a simple linear regression as displayed in Equation 13-22. The kind of regression model we want to study is

$$Y = \beta_0 + \beta_1 X_T + \epsilon \tag{13-23}$$

where X_T denotes the true X score. However, when there is measurement error on X (I'm using X to denote the observed score), then the desired model can be expressed as

$$Y = \beta_0 + \beta_1 (X - \epsilon_x) + \epsilon \tag{13-24}$$

where ϵ_x is the measurement error on X. Note that the previous two equations are identical because X_T = X-ϵ_x.

This last equation shows that a crucial assumption in regression is being violated: independence of error terms. By re-arranging terms we get

$$Y \quad = \quad \beta_0 + \beta_1 X + (\epsilon - \beta_1 \epsilon_x) \qquad (13\text{-}25)$$

and can see that the total error is correlated with X[6].

I have just shown that the independence assumption is violated when there is measurement error on X. Now let's see exactly what happens to the β_1 term in the regression equation when there is measurement error on X. Below I show that the β_1 under measurement error for X will always be less than the true β_1 (i.e., the true beta without measurement error).

We need to define some terms. Y is the dependent variable, X is the observed predictor, X$_T$ is the true predictor, ϵ_{X} is the measurement error on X, and ϵ is the measurement error on Y.

The β_1 for the regression of Y on X is equal to

$$\frac{\text{cov}(X, Y)}{\text{var}(X)} \qquad (13\text{-}26)$$

(this is just the standard definition of the slope). Similarly, the true β_1, denoted β_1^{true}, for the regression of Y on X$_{\mathrm{T}}$ is

$$\frac{\text{cov}(X_T, Y)}{\text{var}(X_T)} \qquad (13\text{-}27)$$

The covariance of X and Y is equal to (under independence of ϵ_x from Y)

$$\begin{aligned} \text{cov}(X_T + \epsilon_x, Y) &= \text{cov}(X_T, Y) && (13\text{-}28) \\ &= \beta_1^{\text{true}}\text{var}(X_T) && (13\text{-}29) \end{aligned}$$

One more definition. The reliability, as discussed above, will be denoted r_{xx}. The xx subscript suggests the interpretation of the correlation of X with itself (as in a test-retest situation). Following the definition of reliability (Equation 13-9)

$$r_{xx} \quad = \quad \frac{\text{var}(X_T)}{\text{var}(X)} \qquad (13\text{-}30)$$

Now the last step, recall that the slope using the observed predictor is

$$\frac{\text{cov}(X, Y)}{\text{var}(X)} \qquad (13\text{-}31)$$

[6] As a do-it-yourself exercise, you might want to show the covariance between X and $(\epsilon + \beta_1 \epsilon_x)$ equals $\beta_1 \text{var}(\epsilon_x)$. This value will not usually be zero and thus, there is a violation of the independence assumption because now the residual will be correlated with the predictor.

But I just showed that the cov(X,Y) = $\beta_1^{\text{true}}\text{var}(X_T)$, so

$$\begin{aligned}
\beta_1 &= \frac{\text{cov(X,Y)}}{\text{var(X)}} && \text{(13-32)}\\
&= \frac{\beta_1^{\text{true}}\text{var}(X_T)}{\text{var(X)}} && \text{(13-33)}\\
&= \beta_1^{\text{true}} r_{xx} && \text{(13-34)}
\end{aligned}$$

Now because r_{xx}, the reliability of X, is always a number between 0 and 1 β_1 will be less than or equal to β_1^{true}. So measurement error introduces systematic bias on the slope parameter. Things get more complicated, of course, with multiple predictors.

What do you do when your predictor has measurement error? There are several things to do to deal with measurement error. First, if you found significance with a "noisy" predictor, then you are probably okay because noise probably made the test less powerful[7]. The problem comes when you fail to find significance—it might be that the true X variable is related, but the measurement error masked that relationship. Second, a technique that some people have used involves the use of "instrumental variables." An instrumental variable is correlated with the true X but not with the measurement error on X. Instrumental variables can then be included in the analyses in a way that helps adjust for the error issue. A practical problem with this technique is that it is difficult to know when a variable is correlated with true X but not with ϵ_x. A third technique involves the use of structural equations models (SEM). These models include measurement error directly in the regression equation. Constructing a model and developing an algorithm for estimating/testing the parameters of the model is, obviously, the best thing to do.

11. Reliability and Multiple Regression

All bets are off when trying to get a handle on the effects of measurement error on predictors in the context of multiple regression. Measurement error can increase, decrease, or even change the sign of correlations and betas at the observed level. As you might suspect, getting a deep understanding of the effects of measurement error in the context of multiple regression is not easy. Following Cohen and Cohen (1983, pp. 406-413) we will build intuition by using the notion of partial correlations.

Recall from Lecture Notes 8 that a partial correlation involves a correlation between two variables of interest where the effects of a third variable (typically a nuisance variable or a covariate, but sometimes a theoretically meaningful variable such as a mediator) have been removed. One way to think about this is to imagine a correlation between Y and X where you want to "partial out" the effects of a variable W. The partial correlation between Y and X

[7] This may not generally be the case because the above result was obtained under the assumption that the measurement error on X is independent from Y. How violations of such assumptions influence both β_1 and the variance of β_1 is very complicated and no simple statement can be made.

(partialling the effects of W) is equivalent to performing two regressions (Y on W and X on W) and correlating the residuals from the two regressions. In other words, the regression of Y on W creates residuals that have the linear effect of W "removed" from Y and the regression of X on W creates residuals that have the linear effect of W "removed" from X. The correlation of these two sets of residuals is the partial correlation.

A relatively simple way to compute the partial correlation is by the formula

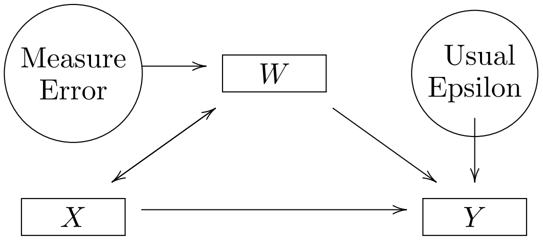
$$r_{yx.w} = \frac{r_{yx} - r_{yw}r_{xw}}{\sqrt{(1-r_{yw}^2)(1-r_{xw}^2)}} \qquad (13\text{-}35)$$

The partial correlation formula (Equation 13-35) provides a convenient way to understand the effects of measurement error on each of the variables. Cohen and Cohen (1983, p406-413) discuss the different effects of having measurement error on Y, X, and W. Here is a table showing the effects of measurement error on the observed partial correlation. We allow measurement error on variable W and assess the effects on the partial correlation r$_{yx.w}$. I only display a few cases (all for reliability .7), but the effects are more dramatic for lower reliability.

example	r$_{yx}$	r$_{yw}$	r$_{xw}$	r$_{ww}$	r$_{yx.w}$	true r$_{yx.w}$
1	.3	.5	.6	.7	0	-.23
2	.5	.7	.5	.7	.24	0
3	.5	.7	.6	.7	.14	-.26
4	.5	.3	.8	.7	.45	.57
5	.5	.3	.6	.7	.42	.37

Look at how discrepant the observed partial is from the true partial. That is, with a reliability of .7, which many people take to be acceptable level of reliability in social science. Things get worse with lower reliability, and extremely more complicated with more than two predictor variables and less than perfect reliability on two or more of those predictors. One nice feature of ANOVA is that the predictors are known (e.g., you assign subjects to groups) so the predictors have perfect reliability (assuming we didn't make any coding or assignment errors), so ANOVA doesn't have to deal with measurement error in the predictors in the same way as the more general regression. Of course, add a measured covariate to the ANOVA and now you have the same problem because likely the measured covariate contains measurement error.

The structural model representing this multiple regression with a predictor having error is depicted below.



As before, squares indicated observed variables and circles indicate unobserved or latent variables. We do not observe measurement error or prediction error directly but estimate it from data. Y has two predictors so two arrows coming into its box. Predictors X and W are allowed to be correlates as with any regression. One of the predictors, W, contains measurement error. You can derive tables like the one below by using the definition of partial correlation and the covariance algebra rules introduced at the beginning of the these lecture notes.

A more general point about measurement error in regression. The above table showing that the observed partial correlation (and hence regression beta) can move around a lot and not necessarily track the true partial correlation makes me wonder about people who make the case that they are using "evidence-based approaches" to justify their arguments. For example, recent arguments about whether or not GREs should be included in graduate school admissions has proponents of both sides making claims they have evidence-based recommendations. typically based on analyses involving multiple regressions. In what I have seen, most of these arguments are flawed because, among many reasons, they don't take into account the measurement error of the variables in their regression equations. They say "Look at the evidence. This is what the regression shows." But, typically, the regressions are not trustworthy because they didn't properly account for measurement error, selection effects, etc. We can't trust the regressions because measurement error can completely change the sign of a regression coefficient as the table above showed. My point is that the material reviewed in this section should make us suspicious of hiding behind the "evidence-based" mantra if that just means we blindly run a regression without deeper thinking. A meta-analysis won't help with this issue either because these measurement error issues (and other issues like selection effects) are sprinkled throughout the literature making it extremely difficult to aggregate results in a meta-analysis.

See Horn (1963, Acta Psychologica, 21, 184-271) for additional relations between covariance matrices and a discussion of methodological issues such as correlations with difference scores.

12. Instrumental Variables

Instrumental variables is a popular technique in some of the social sciences. It allows one to exploit a randomization-like structure to get better causal estimates even in the context of a correlational design. A simple example, taken from Gennetian et al, 2008, *Developmental*

Psychology, 44, 381-, involves the relation between mother's education attainment (ME) and child's performance (CP). At best the relation between those two variables is a correlation and without additional information or assumptions one could not move toward "causal inferences" of the effect of ME on CP. Instrumental variables serve the role of that additional information allowing one to do a little more with correlational information; random assignment is another way to move toward causal inferences, but random assignment is not always possible.

Suppose that moms are randomly assigned to a treatment condition (T), such as an intervention designed to influence her educational attainment ME (but not influence in a direct way the child's educational performance). Mom's educational attainment (ME) is assessed after this intervention. One can run a regression with ME = β_0 +β_1T + ϵ_1, which if T is a dummy code, then as we saw in an earlier set of lecture notes showing the connection between regression and the two-sample t test, the slope represents the difference between treatment and control. That part of the design can establish the "causal effect" of the treatment T on ME. But we still don't know about the outcome of interest CP. Do changes in ME "cause" subsequent changes in CP? Is it possible to influence CP by influencing ME?

We can make use of some of the structure in this setup and make a few assumptions. We will use what is called a "two stage least squares estimation", which is just a fancy way of saying run one regression, take relevant output from that regression and use it in a second regression (so two stages[8]). The first regression is the one I mentioned in the previous paragraph: take the predicted values $\widehat{\text{ME}}$. This represents the predicted educational attainment of mom given her random assignment to treatment T; those predicted scores do not include the residuals (i.e., noise) and hence any correlation between the residuals ϵ_1 and other variables have been removed. In other words, ME and CP could be correlated because they share a common variable, but by using the predicted values $\widehat{\text{ME}}$ that are associated with the treatment T we reduce the impact of such issues like common factor and correlated residuals.

Then use that predicted value of ME (denoted $\widehat{\text{ME}}$) in a second equation to predict the child's educational performance, i.e., the regression CP = β_0' +$\beta_1'\widehat{\text{ME}}$ + ϵ_2 where primes are used to distinguish the slope and intercept from the previous regression. The two sets of residuals are denoted by 1 and 2 for first and second regression. If we make some assumptions, such as the correlation between the treatment and the residuals in the second regression is 0, we can derive an estimate of the direct effect (some would say the causal link) between mother's educational attainment and child's performance. This example used a random assignment, but econometricians have shown that even when there isn't random assignment, as long as some additional assumptions are met, one can use this logic to derive causal estimates even from a correlation design.

The two step regression I outlined above yields the right parameter estimate, but the standard

[8] There are now more efficient and better approaches than two-stage least squares but the logic of the procedure is the same. The improvements have been more on the computational side of how the regression weights and error terms are computed.

error for the regression slope β'_1 will be off because the second regression does not properly model the correlation between the epsilons. There are specialized programs, including one in SPSS, that run instrumental variable analyses and yield the proper error term. Here is some example syntax using the variables in the example I've been discussing.

```
2SLS CP WITH ME
  /INSTRUMENTS T
  /CONSTANT.
```

You specify the DV and critical predictor in the first line and define the instrument in the second line. The output looks like this:

```
Model Summary
Equation 1 Multiple R .432
R Square .187
Adjusted R Square .142
Std. Error of the Estimate 6.206

ANOVA
Sum of Squares df Mean Square F Sig.
Equation 1 Regression 159.454 1 159.454 4.140 .057
Residual 693.291 18 38.516
Total 852.745 19

Coefficients
                      B       SE    beta    t     p
Equation 1 (Constant) 14.597 42.438         .344 .735
M                       .863   .424 1.361 2.035 .057
```

In R the two stage least squares function is ivreg in the AER package, with the same results.

```
library(AER)
summary(ivreg(CP ~ ME | T, data = data))


##
## Call:
## ivreg(formula = CP ~ ME | T, data = data)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.856 -4.586 -0.844  2.609 14.290
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.5966    42.4379   0.344   0.7349
## ME            0.8629     0.4241   2.035   0.0569 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.206 on 18 degrees of freedom
## Multiple R-Squared: -1.586,Adjusted R-squared: -1.729
## Wald test:  4.14 on 1 and 18 DF,  p-value: 0.05688
```

We can use our new knowledge in covariance algebra to figure out what all the fuss is about with instrumental variables. We know that a slope is equal to the ratio of the covariance between the DV and the predictor and the variance of the predictor (LN#6). Thus, in the second regression just described, the slope β_1' can be rewritten as

$$\text{cov}(\text{CP},\widehat{\text{ME}})/\text{var}(\widehat{\text{ME}}) \tag{13-36}$$

Substituting in the regression equation for $\widehat{\text{ME}}$ leads to the familiar equation for the instrumental slope after making some assumptions about correlated error

$$\beta_1' \;=\; \frac{\text{cov(CP, T)}}{\text{cov(ME, T)}} \tag{13-37}$$

This slope amounts to a ratio of two differences: the difference in treatment for the dependent variable CP over the difference in treatment for the predictor variable mother's educational attainment.

This equation can be derived from taking Equation 13-36 and using covariance algebra to re-express the terms to simplify to Equation 13-37. That is,

$$\begin{aligned}
\beta_1' &= \frac{\text{cov}(\text{CP}, \widehat{\text{ME}})}{\text{var}(\widehat{\text{ME}})} \\
&= \frac{\text{cov}(\text{CP}, \beta_0 + \beta_1 T)}{\text{var}(\beta_0 + \beta_1 T)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\beta_1 \text{cov(CP,T)}}{\beta_1^2 \text{var(T)}} \\
&= \frac{\text{cov(CP,T)}}{\text{cov(ME,T)}}
\end{aligned}
$$

Note that $\widehat{\text{ME}}$ (predicted ME from the first regression in the two stage model) in this setup does not have a residual variance so the numerator and denominator in the second line don't include an ϵ_1. The last line follows because $\beta_1 = \frac{\text{cov(ME,T)}}{\text{var(T)}}$. Equivalent equations that you may see in the literature are (1) a ratio of two betas (e.g., ratio of slopes from the regression of CP on T and regression of ME on T) and, (2) if T is a binary variable indicating membership in one of two groups, the IV estimator defined as the ratio of the difference of the means on the DV over the difference of the means on the predictor variable (e.g., $\frac{\mu_{Y1}-\mu_{Y2}}{\mu_{P1}-\mu_{P2}}$ where Y is the DV (CP in our running example), P stands for the predictor variable (ME in our running example) and numbers 1,2 refer to groups 1 and 2).

I introduce this example here to illustrate how covariance algebra can illuminate some complicated statistical ideas. Instrumental variables are not easy to understand on their own, but by using covariance algebra the idea and the importance of its assumptions fall out naturally.

Another way to understand what is going on is to reproduce the results as a sequence of two regressions. Run one regression with the instrument predicting the predictor, save the fitted values, and use the fitted values to predict the dependent variable. See below. Note that the betas are the same as ivreg() and SPSS but the standard errors are off; do not trust the standard errors out of lm() when using two-stage least squares. One needs special programs like ivreg() to compute the proper standard errors. Still this example illustrates the same betas emerge so we verify our understanding.[9]

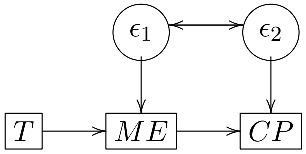
```
reg1 <- lm(ME ~ T, data = data)
fitted.vals <- predict(reg1)
reg2 <- lm(CP ~ fitted.vals, data = data)
summary(reg2)



##
## Call:
## lm(formula = CP ~ fitted.vals, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.5118 -2.0608  0.0495  1.6410  4.0363
##
## Coefficients:
```

[9] One would not get the same betas if residuals are used instead of the fitted values or if both predictor and instrument are included in the same regression with two predictors.

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.5966    16.8009   0.869    0.396
## fitted.vals   0.8629     0.1679   5.139 6.87e-05 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.457 on 18 degrees of freedom
## Multiple R-squared:  0.5947,Adjusted R-squared:  0.5722
## F-statistic: 26.41 on 1 and 18 DF,  p-value: 6.872e-05
```

The instrumental variables logic can be applied in a structural equation modeling (SEM) context as well. One needs to be careful because SEMs are usually estimated in a maximum likelihood context whereas instrumental variables are traditionally estimated in a two stage least squares context, so the estimates and their standard errors may differ slightly if one compares the output from SEM to the output from a traditional instrumental variables program.

The SEM representation of the instrumental variables approach is given in this Figure. Note that the underlying logic is that there is correlation between the two key variables, ME and CP in this case, and the point of using an instrument is to get around that correlation. The assumption is that the instrument correlates with one of the variables but not the other, so we can use the betas from regressions to control for the possible correlation between the two key variables we can't otherwise control.



This is similar to the mediation model (presented later in these notes) but the focus in instrumental variables is on the intermediate path between ME and CP (slope β_1' above) rather than whether there is a direct path between T and CP (as in the case of mediation). One of the key assumptions of an instrument is that it NOT have a direct path to the final variable, so the instrument should not directly influence CP except through the mediator. In the mediation literature this is called perfect mediation.

The IV literature has correctly addressed the issue of correlated residuals. However, related issues of measurement error have been pretty much ignored in the mediation literature (though mentioned in the original Baron & Kenney article). As we saw earlier in these lecture notes, measurement error in the context of a multiple regression can create serious problems.

An interesting observation is that if one views the instrument as another measurement of the predictor variable (as in test-retest), then the covariance between the instrument and the predictor is the numerator in the test-retest correlation (see earlier in these lecture notes; it is equal to the variance of the "true" score). The beta of the instrumental variable approach is closely related to the disattenuated correlation in test theory, which shows that one can account for the bias in a correlation due to less than perfect reliability through the formula

$$\text{cor(Tx, Ty)} \quad = \quad \frac{\text{cor(x, y)}}{\sqrt{\text{rel(x)*rel(y)}}}$$

where x and y are observed variables, Tx represents "true" x (i.e., x without measurement error) and rel(x) represents the reliability of x (see the relevant definitions and results presented earlier in these lecture notes).

To see this, let T1 and T2 be the X and the instrument, respectively, and Y be the dependent variable. We know that the instrumental variable estimate is cov(Y,T2)/cov(T1,T2). Because cov(T1,T2) estimates V(T) and cov(Y,T2) is equal to r(Y,T2)sqrt(V(Y)*V(T1)), we can rewrite the instrumental variable estimate as

$$\frac{\text{r(Y,T2)}}{\sqrt{\text{rel(Y) * rel(T2)}}}\sqrt{\frac{\text{V(true Y)}}{\text{V(true T)}}}.$$

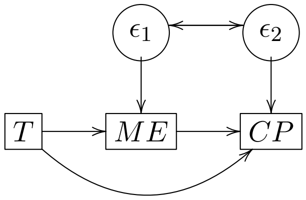
This is simply the classic disattenuated correlation from test theory times a factor of the ratio of the square root of the true score variance for Y over the true score variance for T. The factor of the ratio of square root of the true score variances converts the estimate of the true score correlation (a concept from test theory) to the estimate of the true score regression beta (a concept from the instrumental variable approach). This is an unusual interpretation of the instrumental variable approach but it falls out completely from the covariance algebra; it shows that there is a deep connection between classic test theory and modern approaches of instrumental variables. Most proponents of instrumental variables believe they are doing something newer than classic test theory but now you know better (☺).

It may appear that a natural use of instrumental variables in behavioral work is when one has a manipulation check and wants to examine the effect of the manipulation check variable on the dependent variable. In this case the variable representing the manipulation is the instrument (e.g., group 1 vs group 2). The manipulation check becomes the predictor of interest such that the researcher wants to examine how the variability in the manipulation played out (i.e., some people responded more to the manipulation than others), so the key question becomes how the manipulation check predicts the dependent variable instead of how the intended manipulation predicts the dependent variable. The latter acts as though everyone responded the same way to the treatment, the former uses the "actual" response to the manipulation as measured by the manipulation check . In other words, if one uses the actual manipulation variable as the only predictor, then one examines, in a sense, the "intent to treat" because the analysis focuses on the condition the subjects are in not how they responded to the treatment. However, if one uses the manipulation variable as an instrument, runs a regression with the manipulation check as the dependent variable, uses the predicted values from that regression as predictors in the second stage regression, makes several assumptions, then it may appear they are examining the effect of the manipulation check (how the manipulation actually played out) on the dependent variable. The problem is that the predicted scores in the first regression are two values (i.e., the $\hat{y}$ for group 1 and the $\hat{y}$ for group 2), so when entered in the second regression as a predictor the t test is identical to the test from the regression using the actual manipulation check variable. So in the end using a manipulation check as the predictor and the actual manipulation values as the instrument leads to the same result and running the IV approach ends up being a waste of time. An analogous critique can be made about using the mediation check as a mediator between the actual manipulation and the dependent variable, though that is more subtle because it is the residuals of the mediator that play a key role in determining the path between the mediator and the outcome variable.

The potential use of instrumental variables in correlation studies is compelling. One still needs to make a set of assumptions that cannot easily be empirically validated. The pessimistic take is that if one doesn't use instrumental variables, then in order to interpret the correlation as "causal" one must make lots of handy wavy arguments (i.e., assumptions) about there not being a third variable, that causation goes in one direction, etc. In order to use instrumental variables, one's hands are waving just as much, but in the end one has a formula that leads to an estimate of the "causal relation." While I recognize that randomized studies cannot always be carried out, I can't break away from my traditional upbringing that "cause" should be reserved for randomized experiments. I can't distinguish whether the pair of waving hands from the person who uses instrumental variables is more or less convincing than the pair of waving hands from the person who doesn't and tries to convince me why the reported correlation should be interpreted in a causal manner. In both cases people are waving their hands. I'm happy using correlations in my research because they provide important information about associations and also using randomized designs when possible to address causality. I try to be mindful about when I do and don't use causal language in my thinking.

13. Mediation

The basic mediation analysis uses a similar structure to instrumental variables but the emphasis is on a different part of the path diagram. Whereas in instrumental variables the focus is on examining the path between M and Y, in mediation analysis the question is whether X influences Y directly (the curved path) and/or whether X influences Y indirectly through M. So, mediation boils down to a question about partitioning possible pathways: a direct path from X to Y, a mediated path through M, both, or neither.



A colleague gave me a nice way to conceptualize the meaning of an indirect path. Suppose you tell me a secret and I turn around and tell your advisor. One wouldn't say that you told your advisor (i.e., there was no direct communication link between you and your advisor). I was the middleman who took input from you and spewed it to your advisor. I'm the "mediator" in that example—there are two communication links, one from you to me and another from me to your advisor (the direct link between you and your advisor did not occur).

Much of behavioral science research begins along the lines of simple stimulus-response descriptions (questions such as does X influence Y). Those research questions are black box questions because there isn't much interest in understanding how the relation works, rather the interest in merely in describing the relation. A mechanistic or process view focuses on an intermediate process of how X influences Y. If I pay my son to spend more time studying, and then he gets a high grade on a test, the stimulus-response operational language is that payment led to high grade. But the payment directly lead to a higher grade? Does the test "know" how much my son got paid and is reflecting that variable? Most likely there is a more nuanced account. Payment for studying led my son to spend more time studying, which meant he learned the material better, which led to better comprehension, which led to a higher performance on the exam. This can be compared to the counterfactual where I didn't pay him to study more thereby breaking the chain. You see these kinds of process theories in many behavioral science literatures. The treatment affected behavior, not because it directly changed behavior but because it changed a psychological, or internal state, and in turn that new psychological state contributed to the change in behavior. Different literatures vary on how much value they place on breaking open the black box and testing the intervening variables versus being content with just the stimulus-response association ("I pay my son to study and he gets better grades"). Under this viewpoint, I don't care why it happens, I think my money is well-spent if he gets better grades. As long at it works, I'll continue applying the intervention though there may be unintended consequences of this intervention such as potentially undermining his interest in the material because, after all, he is studying "for the money".

Testing mediation is tricky. The modern method is to run an SEM model and test the product of two βs (the beta in the path from X to M times the beta in the path from M to Y). Testing products of two betas is not easy, and has led to specialized methods using bootstrap and Bayesian approaches.

Even if one can conduct a randomized experiment by manipulating X, it isn't clear that the randomized experimental approach really in yielding causal explanations because the relation between M and Y isn't manipulated if one only manipulates X. If one manipulates both X and M, then that isn't directly testing mediation—one could test, for example, for main effects and interaction of X and M on Y. But that isn't testing the intervening variable proposition that X influences M (you tell me a secret) and in turn M influences Y (I then tell the secret to your advisor). So manipulation doesn't necessarily help with these kind of process models.

We should offer an entire course on the variants of mediation analysis, and there are complete textbooks just on mediation (a solid one is by David Mackinnon). Here, I just want to introduce the concept of mediation so you have some familiarity if you encounter it in your research literature. There are many details one needs to consider when running mediation-like tests, such as there are several models (ways of drawing arrows) that lead to different interpretations yet they are indistinguishable because they have the identical fit criteria. That is, different models imply the same covariance matrix, so data cannot distinguish different representations without additional assumptions or structure. An example is three variables X, M and Y were collected at the same time, then it is difficult to disentangle directions of arrows making it difficult to interpret the results along causal mechanisms without making assumptions about causal ordering. However, if X was collected at time 1, M was collected at time 2 and Y was collected at time 3, then Y could not have been a direct cause of M and consequently we can rule out that possible relation.

Here is one approach to testing mediation in R using the lavaan package. I'm also using lavaan here as a way to gently introduce you to SEM modeling. We first define the structural model, which can be a set of several structural models. Here we have two structural submodels: one model for the mediator and a second model for the dependent variable. I'm using the letters a, b and c as labels for the betas so that I can then reuse the later to define new estimates like the product of a and b.[10] The analogous operation can be done in SPSS through the AMOS package but I prefer not to use AMOS because the syntax aspect is too cumbersome and the pull-down menu system to build a model is too clunky and can easily lead to errors in model specification. I'll show an example of AMOS syntax later in this notes and you'll see what I mean.

Here are the two structural models needed for the lavaan package.

```
library(lavaan)
model <- "
#structural model for the mediator
ME ~ a*T

#structural model for the dv
CP ~ b*ME + c*T
```

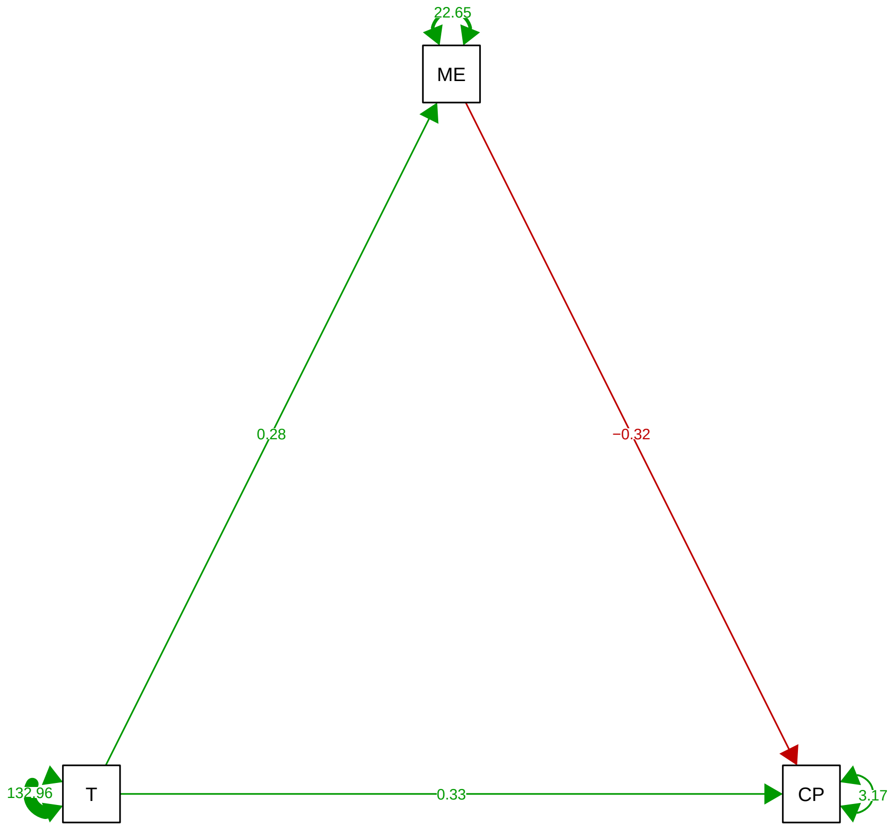
[10] Another way to perform mediation in R is through the mediation package which has some addition features specific to mediation, including Bayesian methods, dealing with binary, count and nominal data, and computing various effect estimates.

```
#extra parameters to compute for est, se(est), t and CI
indirect := a*b
direct := c
total := direct + indirect
"
```

The structural model is submitted to the sem() program in lavaan and summarized. The botto of the summary() output includes the estimates and tests of the indirect, direct and total effects. This R code chunk also includes a plot of the model with estimates and some additional summary using the bootstrap for confidence intervals.

```
# estimate model, could include se='bootstrap'
fit <- sem(model, data = data, estimator = "ML")
summary(fit, standardized = T, fit.measures = T, rsq = T)



## lavaan 0.6.13 ended normally after 1 iteration
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         5
##
##   Number of observations                            20
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Model Test Baseline Model:
##
##   Test statistic                                36.603
##   Degrees of freedom                                 3
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)                -99.487
##   Loglikelihood unrestricted model (H1)        -99.487
##
##   Akaike (AIC)                                 208.975
##   Bayesian (BIC)                               213.953
##   Sample-size adjusted Bayesian (SABIC)        198.540
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.000
##   P-value H_0: RMSEA <= 0.050                       NA
##   P-value H_0: RMSEA >= 0.080                       NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   ME ~
##     T           (a)    0.284    0.092    3.075    0.002
##   CP ~
##     ME          (b)   -0.316    0.084   -3.783    0.000
##     T           (c)    0.335    0.042    7.991    0.000
##    Std.lv  Std.all
##
##     0.284    0.567
##
##    -0.316   -0.499
##     0.335    1.054
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ME               22.654    7.164    3.162    0.002
##    .CP                3.167    1.001    3.162    0.002
##    Std.lv  Std.all
##    22.654    0.679
##     3.167    0.236
##
## R-Square:
##                    Estimate
##     ME                0.321
##     CP                0.764
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     indirect         -0.090    0.038   -2.386    0.017
##     direct            0.335    0.042    7.991    0.000
##     total             0.245    0.045    5.417    0.000
##    Std.lv  Std.all
##    -0.090   -0.283
##     0.335    1.054
##     0.245    0.771
```

```
semPaths(fit, what = "par", fade = FALSE, layout = "spring")
```



```r
# print bootstrap CIs for each parameter
parameterEstimates(fit, boot.ci.type = "bca.simple", level = 0.95,
    ci = TRUE, standardized = FALSE)
```

```
##        lhs op             rhs    label     est    se      z
## 1       ME  ~               T        a   0.284 0.092  3.075
## 2       CP  ~              ME        b  -0.316 0.084 -3.783
## 3       CP  ~               T        c   0.335 0.042  7.991
## 4       ME ~~              ME           22.654 7.164  3.162
## 5       CP ~~              CP            3.167 1.001  3.162
## 6        T ~~               T          132.962 0.000     NA
## 7 indirect :=             a*b indirect  -0.090 0.038 -2.386
## 8   direct :=               c   direct   0.335 0.042  7.991
## 9    total := direct+indirect    total   0.245 0.045  5.417
##   pvalue ci.lower ci.upper
## 1  0.002    0.103    0.465
## 2  0.000   -0.480   -0.152
## 3  0.000    0.253    0.417
## 4  0.002    8.613   36.695
## 5  0.002    1.204    5.129
## 6     NA  132.962  132.962
## 7  0.017   -0.163   -0.016
## 8  0.000    0.253    0.417
## 9  0.000    0.156    0.333
```

The indirect parameter estimate is defined as the product of a and b, the direct parameter estimate is the regression slope for the independent variable controlling for the mediator, and the total effect is the sum of the direct and indirect effects. This example is somewhat unusual in that the indirect effect is close to zero (slightly negative and significantly different from zero) but the two paths a and b are of different sign. The treatment T is associated higher scores in ME and CP, even though the relation between ME and CP is negative. Worth thinking about what this could mean, implications that can be derived and additional data that can test some of those predictions.

You can use covariance algebra to prove that the indirect effect is the product of a and b, and the total is the sum of direct and indirect. The direct effect is equivalent to the regression slope of CP (dv) on T (key predictor), as shown below. The idea of mediation is to decompose the regression slope into two parts (direct and indirect) based on a model in order to assess the proportion of the effect that is directly related to the key independent variable versus the proportion of the effect that is indirectly related to the key independent variable through the mediator (the logic that the iv influences the mediator and then the mediator influences the dv).

We can verify that the simple regression of T predicting CP estimates a slope that is equal to the total effect (direct plus indirect) in the mediation model. The parameter estimate for the slope out of the linear regression is equal to the total effect estimated in the SEM presented earlier.

```
coef(lm(CP ~ T, data = data))


## (Intercept)           T
##   77.014613    0.244872
```

It is instructive to compare mediation to the instrumental variable approach. These are different models even though the diagrams look quite similar. I'll write a model following the instrumental variables logic where T is a predictor of ME, and we estimate another regression with T as a predictor of CP. Dividing these two betas yields the estimate of the instrumental variable approach as per earlier in the lecture notes. Note that this a and b are not the same as the a and b from the mediation, and if we multiply the a and b we don't get the same estimate as the instrumental variables approach.

```r
model2 <- "
#structural model for ME
ME ~ a*T

#structural model for CP
CP ~ b*T

#show that this a*b is not equal to full mediator
not.e.mediator := a*b

#inst variable is b/a as per lecture notes
iv := b/a
"

# estimate model, could include se='bootstrap'
fit2 <- sem(model2, data = data, estimator = "ML")
summary(fit2, standardized = T, fit.measures = T, rsq = T)
```

```
## lavaan 0.6.13 ended normally after 16 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         5
##
##   Number of observations                            20
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Model Test Baseline Model:
##
##   Test statistic                                36.603
##   Degrees of freedom                                 3
##   P-value                                        0.000
##
```

```
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)                -99.487
##   Loglikelihood unrestricted model (H1)        -99.487
##
##   Akaike (AIC)                                 208.975
##   Bayesian (BIC)                               213.953
##   Sample-size adjusted Bayesian (SABIC)        198.540
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.000
##   P-value H_0: RMSEA <= 0.050                       NA
##   P-value H_0: RMSEA >= 0.080                       NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   ME ~
##     T          (a)    0.284    0.092    3.075    0.002
##   CP ~
##     T          (b)    0.245    0.045    5.417    0.000
##    Std.lv  Std.all
##
##     0.284    0.567
##
##     0.245    0.771
##
```

```
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   .ME ~~
##     .CP              -7.165    2.953   -2.426    0.015
##     Std.lv  Std.all
##
##     -7.165   -0.646
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     .ME              22.654    7.164    3.162    0.002
##     .CP               5.433    1.718    3.162    0.002
##     Std.lv  Std.all
##     22.654    0.679
##      5.433    0.405
##
## R-Square:
##                    Estimate
##      ME               0.321
##      CP               0.595
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##      not.e.mediator    0.069    0.017    4.006    0.000
##      iv                0.863    0.402    2.145    0.032
##     Std.lv  Std.all
##      0.069    0.437
##      0.863    1.361
```

The reason the standard error and z test are slightly different than the ivreg output from both SPSS and R is that lavaan uses maximum likelihood (ML) rather than restricted maximum likelihood (REML). The implication is that SEM doesn't address degrees of freedom in the same way that regression approaches do, but as sample size increases these differences become very small. In the limit, REML approaches ML.

FYI: An early paper by Smith (1982, JPSP, Beliefs, Attributions, and Evaluations: Nonhierarchical Models of Mediation in Social Cognition) combined both the logic of instrumental variables and the logic of mediation to be able to estimate models with reciprocal causality (A causes B and B causes A). He combined experimental manipulation in the context of an instrumental variable approach and the logic of mediation of measured variables. This method hasn't received much attention but is an interesting way to use both instrumental and mediation methods simultaneously.

14. Structural Equations Modeling

SEM is a general procedure that can do all ANOVA, regression, PCA, MANOVA, and canonical correlation. The framework also allows for measurement error. It is possible to mix and match these techniques, such as perform a PCA with measurement error within a regression model.

Words of Caution: Structural equations programs such as LISREL, EQS, CALIS, EZPATH, Amos, Mplus, lavaan and the like can be very tricky. It is easy for an inexperienced user (and sometimes even an experienced user) to make a mistake and not be aware that he or she has made one. It is tempting to play with structural/measurement models (add a path here, remove a path over there, correlate an error, free a variance, etc). One little change can have a dramatic impact on the subtle features of the model (e.g., identification). You should develop a deep distrust of structural equations models that appear in the literature. I hope we can instill an appreciation for the elegance and necessity of this type of modeling as well as the motivation for the need to seek further training in how to use them. My goal here is merely to introduce the basic framework so you can develop the necessary intuition and understand the published papers you read.

I already showed how to use this SEM approach to estimate mediation models. I'll now show an application of SEM with canonical correlation (LN#12) but extended to include measurement error so that one can simultaneously assess the reliability of each observed variable along with the factor structure. So, this application of SEM extends a well-known technique in a new direction to account for reliability of the observed data.

15. Factor Models in SEM

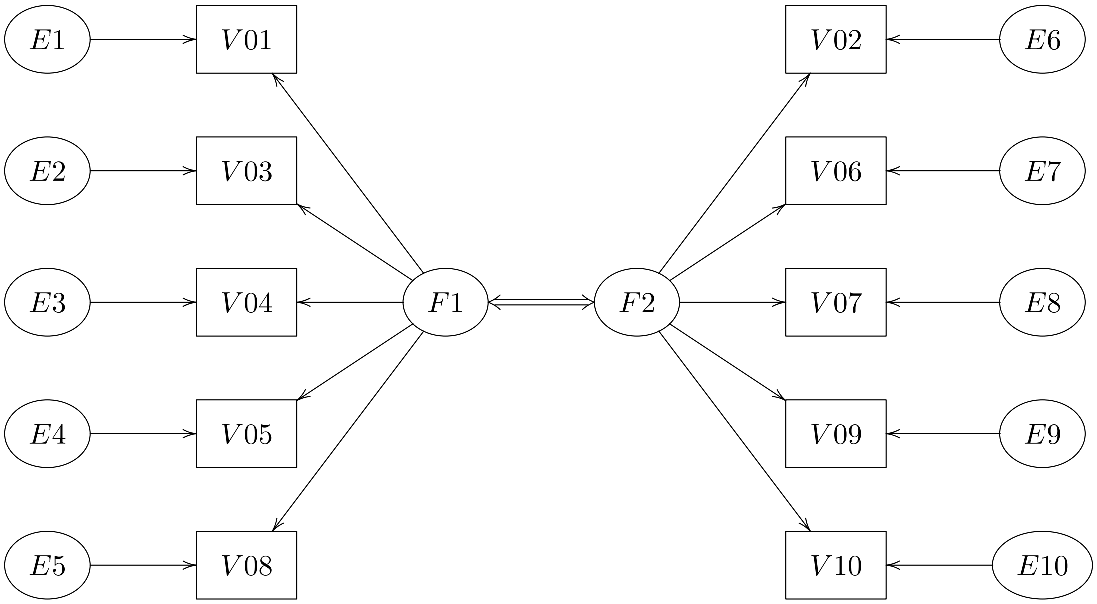
Structural equations modeling provides a way to estimate correlations between latent variables, or factors, in a manner that can automatically take into account measurement error (i.e., one does not need to correct for attenuation). The analysis automatically provides factor loadings and reliabilities along with the relation between the factors.

Recall the self-esteem example from Lecture Notes #12 where I illustrated canonical correlation. Figure 13-2 illustrates the generalization in graphical form of canonical correlation to allow for measurement error. The addition is that there are now 10 circles labelled E1 to E10 pointing to their respective observed variables. For example, this shows that observed variable V01 has two arrows point to it: the unobserved factor F1 and the unobserved measurement error E1 thus representing the regression equation

$$\mathrm{V01} \quad = \quad \beta \mathrm{F1} + \mathrm{E1}$$

There are nine such equations for each of the remaining nine observed variables V02 to V10. Each of these equations separately assesses the measurement error of each variable (the unob-

Figure 13-2: Graphical illustration of a structural equation model on the five positive and five negative items for the self-esteem data.



served E) and the common component, the unobserved factor F. The parameters of the model include 10 βs, the variances of the two factors, and the variances of the unobserved errors (i.e., the variances of the 10 errors). There are a few details to address such as the identification of parameters as we will soon see. Below I present the logic of goodness of fit measures and tests, which compare the observed covariance matrix to the model implied covariance matrix, providing the analogue to residual and sum of squared residuals. In this case the

16. Models

Recall that in regression models we assess the fit between the observed score y and the score predicted by the model, ŷ. A residual is the difference between the observed score and the predicted score:

$$\text{residual} = \text{observed score} - \text{predicted score} \qquad (13\text{-}38)$$

In multiple regression the score predicted by the model is completely determined by the regression coefficients. One way to think about regression coefficients is that they are the weights that minimize the sum of squared residuals (where the sum is taken over cases); that is, for each subject

$$\text{residual} = \text{observed score} - (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots) \qquad (13\text{-}39)$$

and the coefficients are selected so $\sum(\text{residual}^2)$ (sum of squared residuals over subjects) is minimized. The residual for a case is the estimated ϵ for that case.

This concept of difference between observed and predicted can be extended to covariance and correlation matrices.

$$\text{residual matrix} \quad = \quad \text{observed matrix} - \text{predicted matrix} \qquad (13\text{-}40)$$

The difference is taken element by element. For example, assume the observed covariance matrix

	x	w	y
x	0.25	0.13	0.05
w	0.13	3.24	0.87
y	0.05	0.87	0.95

and the expected covariance matrix from some model

	x	w	y
x	0.25	0.30	0.15
w	0.30	3.56	1.46
y	0.15	1.46	1.37

The matrix of residuals is

	x	w	y
x	0.00	-0.17	-0.10
w	-0.17	-0.327	-0.59
y	-0.10	-0.59	-0.75

A matrix of residuals is not easy to interpret as is because the metric is not well-specified (e.g., is -.75 residual something to worry about?). So we need a metric that can be interpreted as goodness of fit of the model. Such an index is usually abbreviated GFI (goodness of fit index).

17. GFI

The material in the next two sections is adapted from Bollen (1989) and Everitt (1984).

First, let's make a connection between GFI and the familiar concept of R^2. Recall that

$$\begin{aligned}\mathrm{R}^2 &= \frac{\mathrm{SSR}}{\mathrm{SST}} && \text{(13-41)}\\ &= \frac{\text{SST-SSE}}{\mathrm{SST}} && \text{(13-42)}\\ &= 1-\frac{\mathrm{SSE}}{\mathrm{SST}} && \text{(13-43)}\end{aligned}$$

The last equation re-expresses R^2 as 1 minus the ratio of observed error over total error. The observed total error can be interpreted as a "baseline" to compare the observed error. Note that the baseline (the denominator) is based on observed scores not expected scores.

The sum of the squared elements in the residual matrix is a nice candidate for the analog to SSE. Square every element in the residual matrix, then sum all the squared elements. The analog to SST is just the sum of the squared elements in the observed covariance matrix (yes, squared variances). This leads to

$$\mathrm{GFI} = 1-\frac{\sum \text{squared terms from residual matrix}}{\sum \text{squared terms from observed cov matrix}} \qquad \text{(13-44)}$$

How is this index interpreted? When the model is perfect, then the sum of squared residuals will be 0 and GFI = 1. When the model is "terrible" in the sense that the model predicts the same number for every subject, the sum of squared residuals will equal the sum of squared terms from the covariance matrix. Note that this "terrible" model could arise in cases where you know nothing and the best prediction for each subject on each variable is the mean of that variable. It is possible for this index to be negative.

NOTE: this particular GFI index is now outdated and has been replaced with better indices. The new indices are computationally intensive and are not as illuminating as the one presented here. See Bollen (1989) for newer treatments. Most GFI measures are based on the same underlying intuition. The reason I presented this form is because it illustrates the ideas in the simplest way, even though it is may not be the optimal way to define the concept.

18. Testing Fit

A simple Chi-square test for testing the fit of the model is

$$X^2 = \frac{(\mathrm{N}-1)}{2}\text{sum of squared terms from residual matrix} \qquad \text{(13-45)}$$

The computed X^2 can be compared to the tabled Chi-square value. The degrees of freedom are

$$\frac{k(k+1)}{2} - p \qquad \text{(13-46)}$$

where k is the total number of variables (i.e., number of rows or columns in the covariance matrix) and p is the number of parameters that are being estimated (e.g., the total number of regression coefficients that are used to compute the expected covariance matrix, not counting any parameter that is associated with error). The total number of parameters is simply the total number of slopes present in the model. Note that the intercepts don't enter here because adding or subtracting a constant has no effect on the variance or the covariance. A modification is needed to add intercepts and means to the mix of structural equations modeling, sometimes called mean-structure SEM.

There are peculiar things about this Chi-square test. First, the number of subjects does not enter into the computation of degrees of freedom. The reason is that what is being taken as data is the covariance matrix not the individual data points. The parameter k, the number of variables, captures the "size" of the covariance matrix which is a k x k matrix. Second, the null hypothesis here is the model you are trying to fit. So you want to accept the null hypothesis. A significant Chi-square means that the model you are testing does *not* fit the data, where by "fitting the data" we mean does the covariance matrix implied by the model match the covariance matrix of the data. Third, the sample size *does* enter into the computation of the actual Chi-square statistic, even though it did not enter into the GFI measure. So, with enough subjects (i.e., power) you will usually be able to reject the null hypothesis. There is an unusual tradeoff here: in order to be sure about accepting the null hypothesis you need lots of power but too much power might lead to erroneous rejection of a reasonable model.

NOTE: this particular Chi-square test is now outdated and has been replaced with better tests. The newer tests are computationally intensive and are not as illuminating as the one presented here. The newer tests get around some of the problems of this test. Also, some of the newer tests count parameters a little differently. We will discuss these newer tests later in the context of structural equations modeling.

19. Theoretical v. Empirical Models

If you know the true parameters, you can compute the expected covariance matrix and compare it to the observed covariance matrix using the GFI and the Chi-square test. However, typically one does not know the true parameters of the models. To what matrix is the observed covariance matrix compared when the true model is not known?

The answer is easy. Assume a model, and run the regressions to estimate the relevant path coefficients. Now you have a model with numerical estimates. You can use the slopes estimated from these regression equations to create the model-implied covariance matrix. For example, suppose I was testing the model that w mediates the relation between x and y. Following Baron & Kenny I would run two regressions: regress y on both w and x and regress x on w. Suppose the two regressions resulted in the following slopes and intercepts

$$\begin{aligned} y &= 3.84 + .46w + .27x && \text{(13-47)} \\ w &= -2.22 + 1.62x && \text{(13-48)} \end{aligned}$$

With this information one can compute the model-implied covariance matrix and compare it to the observed covariance. That is, the regression output provides estimates that are used to compute the “expected” covariance matrix. This “expected” covariance matrix can be compared to the observed covariance matrix to see how well the model is doing in capturing the observed data. The GFI index and X^2 test can also be calculated. This is how the program works. Instead of known the regression betas, they are estimated in order to maximize GFI. That is, the program searches the optimal combination of betas that produce the expected covariance matrix that is closest to the observed covariance matrix.

In this way we can simultaneously assess the fit of several regression models. This is useful for more realistic testing where there may be several dependent variables and multiple hypotheses over these variables.

In this way, one use of SEM is to estimate several regression equations simultaneously (i.e., a system of regression equations).

20. SEM example

Let’s compute the earlier example with two factors on six observed variables with measurement error on each observed variable. I will use the six equations to set up a model that predicts the observed 6x6 covariance matrix. This model implied covariance matrix will be compared to the observed covariance matrix to estimate parameters (βs, factor variances, error variances, factor covariances). The six equations are listed here:

$$\begin{aligned}
\text{x1A} &= \beta\text{1F1} + \text{E1} \qquad (13\text{-}49)\\
\text{x1B} &= \beta\text{2F1} + \text{E2} \qquad (13\text{-}50)\\
\text{x1C} &= \beta\text{2F1} + \text{E3} \qquad (13\text{-}51)\\
\text{x2A} &= \beta\text{3F2} + \text{E4} \qquad (13\text{-}52)\\
\text{x2B} &= \beta\text{4F2} + \text{E5} \qquad (13\text{-}53)\\
\text{x2C} &= \beta\text{4F2} + \text{E6} \qquad (13\text{-}54)
\end{aligned}$$

Here we have six observed variables, x1A to x2C. Each variable has its own error (denoted E1 to E6). There are two factors denoted F1 and F2. This can be interpreted as a set of six regression equations, where the goal is to estimate the six βs and the six error variances using two unknown factors. In this simple model we will set the variances of the unknown factors to 1 and force the two factors to be independent (i.e., have a covariance equal to 0). We will relax the second assumption (uncorrelated factors) later.

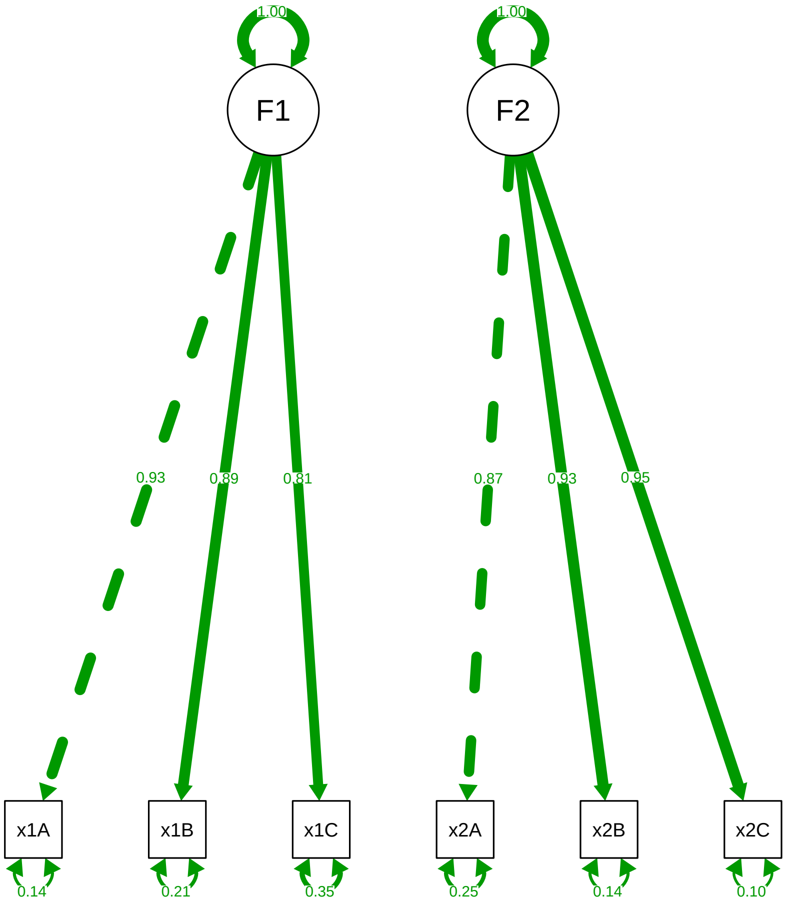
Here I’ll use the R package lavaan. It is similar to the syntax used in MPlus making it easy to go back and forth (one can even run MPlus within R). I prefer to deviate from the traditional maximum likelihood estimate which uses biased variances/covariances (i.e., divide by N) rather than unbiased variances/covariances (i.e., divide by N - 1). Different programs can have different defaults so you should check the program you use so you understand the default settings. Mplus and R's lavaan both use the biased ML approach by default; below I specify likelihood="wishart" to yield the N - 1 version. Later, I also show AMOS output from SPSS.

```
library(lavaan)
data <- read.table(
#/users/gonzo/rich/Teach/multstat/unixfiles/lectnotes/sem/
  "rg1.data")
colnames(data) <- c("id", "x1A", "x1B", "x1C", "x2A","x2B","x2C")

model1 <- '
#linear equations
F1 =~ x1A + x1B + x1C
F2 =~ x2A + x2B + x2C

#set variances and covariances
F1 ~~ F1
F2 ~~ F2
F1 ~~ 0*F2
'

out.sem <- sem(model1, data=data,likelihood="wishart")
summary(out.sem,fit.measures=T)
```

```
## lavaan 0.6.13 ended normally after 24 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##
##   Number of observations                           120
##
## Model Test User Model:
##
##   Test statistic                                13.724
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.132
##
## Model Test Baseline Model:
##
##   Test statistic                               576.533
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.992
##   Tucker-Lewis Index (TLI)                       0.986
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)               -880.980
##   Loglikelihood unrestricted model (H1)       -874.061
##
##   Akaike (AIC)                                1785.961
##   Bayesian (BIC)                              1819.411
##   Sample-size adjusted Bayesian (SABIC)       1781.472
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.066
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.133
##   P-value H_0: RMSEA <= 0.050                    0.306
##   P-value H_0: RMSEA >= 0.080                    0.422
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.106
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     x1A               1.000
##     x1B               1.220    0.090   13.569    0.000
##     x1C               0.843    0.073   11.626    0.000
##   F2 =~
##     x2A               1.000
##     x2B               1.051    0.072   14.660    0.000
##     x2C               1.422    0.094   15.145    0.000
```

```
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 ~~
##     F2                0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     F1                1.089    0.170    6.389    0.000
##     F2                0.971    0.165    5.888    0.000
##    .x1A               0.176    0.057    3.121    0.002
##    .x1B               0.419    0.094    4.449    0.000
##    .x1C               0.416    0.065    6.377    0.000
##    .x2A               0.320    0.051    6.269    0.000
##    .x2B               0.179    0.040    4.444    0.000
##    .x2C               0.225    0.067    3.363    0.001
```

```
library(semPlot)
semPaths(out.sem,what="std",fade=FALSE)
```



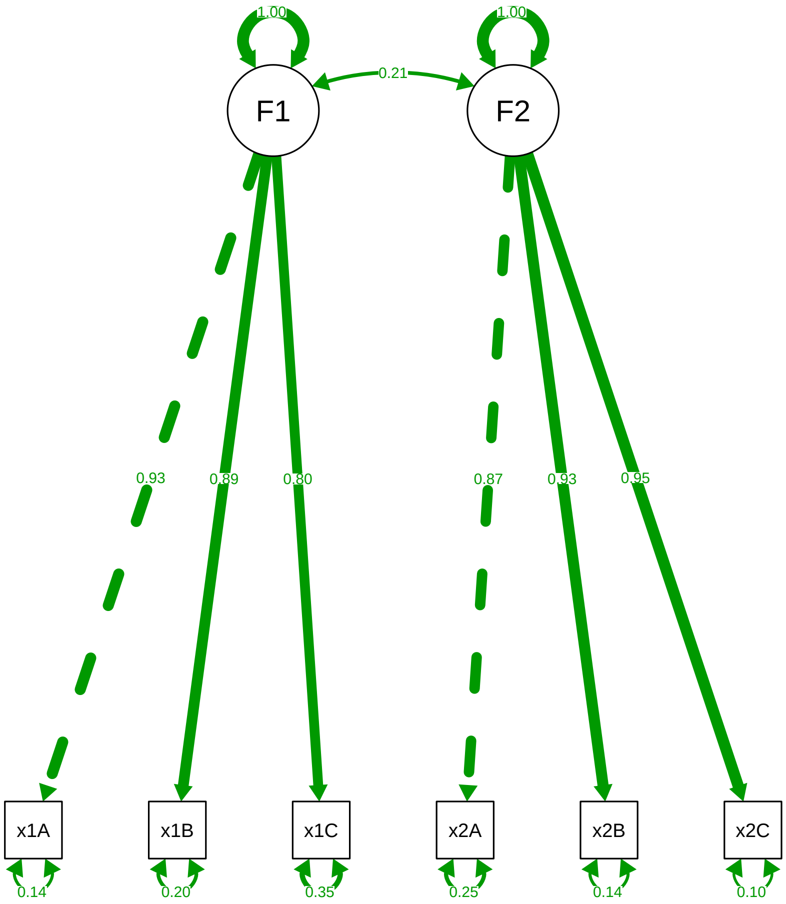
The model contains the six regression equations and specifications for the variances and covariances. For each observed variable we define a factor as a predictor and set up measurement error. The covariance between the two factors is fixed to 0 in this first run to yield two independent factors; a two-headed arrow does not appear between the two latent factors. The output reports goodness of fit (CFI and TLI) measures for this model that are related but not equavilent to the GFI presented earlier; the output also presents a Chi-square test that is a variant of the one presented earlier in these notes.

The output shows the estimates of the paths (one path in each factor is fixed to 1 for identification), the estimated factor variances and the six error variances.

One may be interested in testing the correlation between the two factors. You run the same syntax except change the one line that fixes the correlation to 0 so that the program estimates the correlation.

```
model2 <- "
#linear equations
F1 =~ x1A + x1B + x1C
F2 =~ x2A + x2B + x2C

#variances and covariances
F1 ~~ F1
F2 ~~ F2
F1 ~~ F2
"
out2.sem <- sem(model2, data = data, likelihood = "wishart")
summary(out2.sem, fit.measures = T)
```

```
## lavaan 0.6.13 ended normally after 25 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           120
##
## Model Test User Model:
##
##   Test statistic                                 9.255
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           0.321
##
## Model Test Baseline Model:
##
##   Test statistic                               576.533
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.998
##   Tucker-Lewis Index (TLI)                       0.996
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)               -878.727
##   Loglikelihood unrestricted model (H1)       -874.061
##
##   Akaike (AIC)                                1783.454
##   Bayesian (BIC)                              1819.691
##   Sample-size adjusted Bayesian (SABIC)       1778.591
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.036
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.117
##   P-value H_0: RMSEA <= 0.050                    0.528
##   P-value H_0: RMSEA >= 0.080                    0.238
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.040
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     x1A               1.000
##     x1B               1.226    0.090   13.663    0.000
##     x1C               0.842    0.073   11.585    0.000
##   F2 =~
##     x2A               1.000
##     x2B               1.052    0.072   14.677    0.000
##     x2C               1.421    0.094   15.117    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 ~~
##     F2                0.211    0.103    2.046    0.041
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     F1                1.086    0.170    6.381    0.000
##     F2                0.970    0.165    5.884    0.000
##    .x1A              0.179    0.056    3.195    0.001
##    .x1B              0.409    0.093    4.385    0.000
##    .x1C              0.420    0.066    6.413    0.000
##    .x2A              0.321    0.051    6.282    0.000
##    .x2B              0.176    0.040    4.403    0.000
##    .x2C              0.229    0.067    3.429    0.001


semPaths(out2.sem, what = "std", fade = FALSE)
```



The covariance between the two factors is .206 and is statistically significant. The figure now draws a two-headed arrow between the two latent factors.

It turns out that the test presented in most SEM programs for individual variables is incorrectly defined (I can supply a paper to those who are interested). The best way to test parameters such as the covariance or correlation between factors is to compare the Chi square tests across the two runs, the full and reduced model where the full model is when the parameter is free and the reduced model is when the parameter is set to the value of the null hypothesis (which is usually 0). Looking at these two outputs we have a Chisq of 9.25 with 8 degrees of freedom for the full model and a Chisq of 13.724 with 9 degrees of freedom for the reduced model. Obviously, the reduced model fits worse (a greater Chisq) because it has one fewer parameter to fit the data. There is one degree of freedom difference because the only change from the reduced to the full model is that we estimate one more parameter (the value of the covariance between the two factors). The difference between those two Chisq values is itself a Chisq distribution, so we can take 13.724 - 9.25 = 4.47, with one degree of freedom (9-8=1). You then look up the p-value for a Chisq of 4.47 with 1 d.f., which is about 0.035.

Under some conditions, the anova() command in R will correctly compare two lavaan models. You can see that the computation is equivalent to what I show by hand in the earlier paragraph.

```
anova(out.sem, out2.sem)



##
## Chi-Squared Difference Test
##
##          Df    AIC    BIC   Chisq Chisq diff   RMSEA
## out2.sem  8 1783.5 1819.7  9.2547
## out.sem   9 1786.0 1819.4 13.7240     4.4693 0.17003
##          Df diff Pr(>Chisq)
## out2.sem
## out.sem        1    0.03451 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

CFA, EFA & PCA

It is possible to estimate exploratory factor analysis (EFA) using a confirmatory factor analysis (CFA) program like lavaan. There needs to be some constraints added in order to identify the model as all paths from each factor to each indicator are estimated in EFA, but the results are the same once the constraints are properly taken into account. Likewise, CFA can reproduce PCA if one omits the error variances on each indicator (eliminate the ϵs, as I showed in LN12); I've seen that some programs complain if all the error variances are set to zero but a simple workaround is to set the variances to very small numbers like .0001 (effectively zero).

Amos

Redo example with AMOS. Can be skipped if you don't use AMOS.

Here is the AMOS syntax. AMOS has a nice feature that it is possible to draw the SEM diagram on the screen to build the model (rather than entering syntax) but you really need to know what you are doing since there are constraints that need to be set and such. Best to start with syntax and then when you understand the program, work your way over to the graphical interface. The syntax though is not easy to read; it is possible that the syntax has been simplified and I missed the memo.

```
#Region "Header"
Imports System
Imports System.Diagnostics
Imports Microsoft.VisualBasic
Imports AmosEngineLib
Imports AmosGraphics
Imports AmosEngineLib.AmosEngine.TMatrixID
Imports PBayes
#End Region

Module MainModule
Public Sub Main()
Dim Sem As AmosEngineLib.AmosEngine=New AmosEngineLib.AmosEngine
Sem.TextOutput()
Sem.Standardized()
Sem.Smc()

Sem.BeginGroup ("f:\rich\teach\multstat\unixfiles\lectnotes\sem\rg1.sav")

     Sem.AStructure ("x1A   = (1) F1 + (1) err_x1A")
     Sem.AStructure ("x1B   =     F1 + (1) err_x1B")
     Sem.AStructure ("x1C   =     F1 + (1) err_x1C")

     Sem.AStructure ("x2A   = (1) F2 + (1) err_x2A")
     Sem.AStructure ("x2B   =     F2 + (1) err_x2B")
     Sem.AStructure ("x2C   =     F2 + (1) err_x2C")

     Sem.AStructure ("err_x1A (ex1A)")
     Sem.AStructure ("err_x1B (ex1B)")
     Sem.AStructure ("err_x1C (ex1C)")
     Sem.AStructure ("err_x2A (ex2A)")
     Sem.AStructure ("err_x2B (ex2B)")
     Sem.AStructure ("err_x2C (ex2C)")
     Sem.AStructure ("F1 (VF1)")
     Sem.AStructure ("F2 (VF2)")
Sem.Dispose()
```

```
End Sub
End Module
```

Let's walk through this syntax. The first few lines determine some of the output, such as setting text output and printing the standardized solution as well as the raw solution. The Smc piece adds the reliabilities to the output. The file containing the SPSS data is "rg1.sav". The next block of output sets up the six regression equations. For each observed variable we define a factor as a predictor and set up measurement error. Then we define the six errors of measurement and the variances of the two factors are set to 1. The correlation between the two factors is set to 0.

The output is presented in Figure 13-3. The output includes the six regression equations equations, with betas and their individual tests, the standarized betas, the six error variances, tests and goodness of fit measures for the entire model (all six regressions simultaneously fit and tested to the covariance matrix), and finally the reliabilities of each variable.

One may be interested in testing the correlation between the two factors. You run the same syntax except change the one line that fixes the correlation to 0 so that the program estimates the correlation. The one line of syntax is changed to the following where "corr" is an arbitrary label I assign to that parameter:

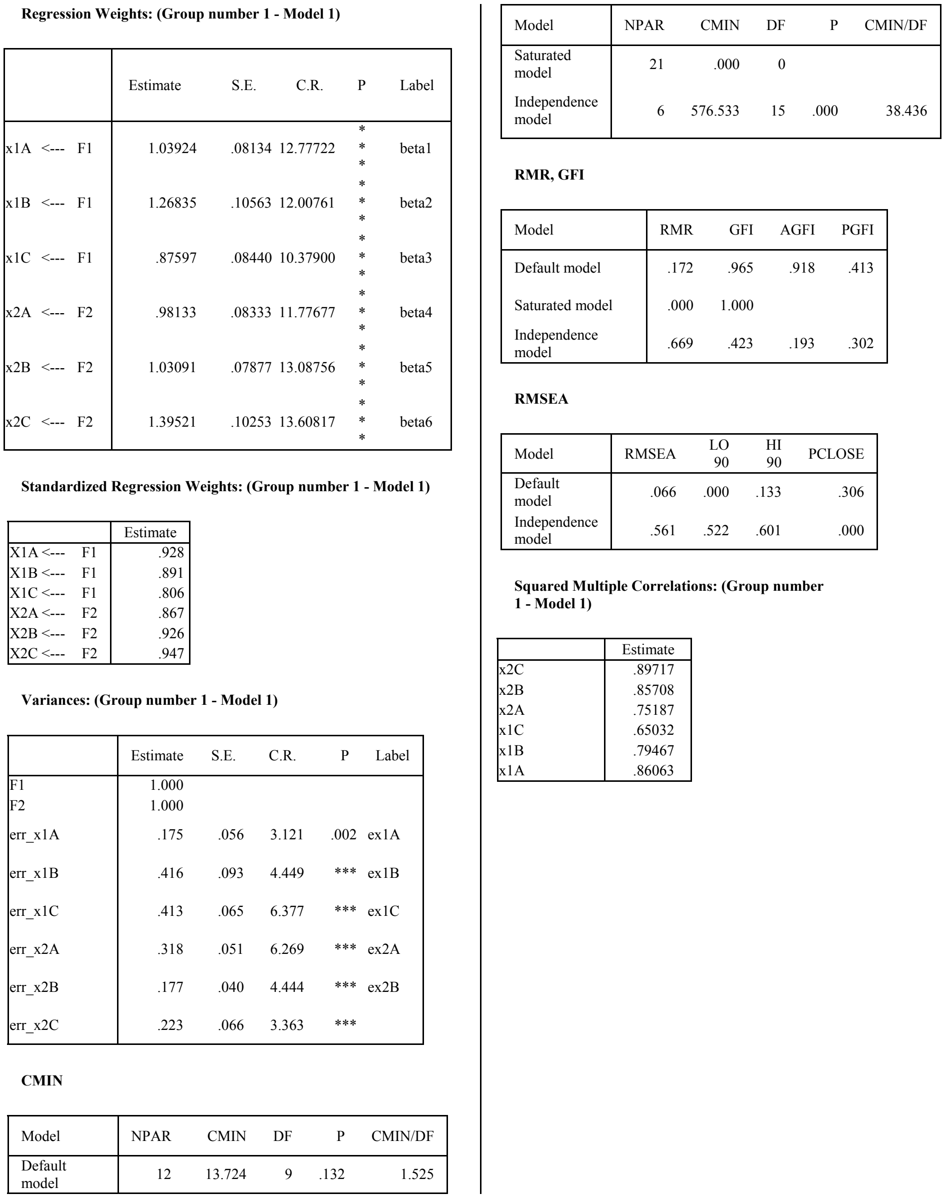
```
    Sem.AStructure ("F1 <> F2 (corr)")
```

After running this I get new output which I present only snippets in Figure 13-4. The covariance between the two factors is .206 and is statistically significant.

21. What to Report from an SEM Run?

What do you do after you run all these models? You would like to decide which model "fits" the data better so you need to supply the necessary information. You can organize the output in a way that facilitates comparison. For example, here is one suggestion (you might prefer a different way). You could also list the residual matrix of each model and eye-ball to see where each of them fails. Once you decide on a best model, then you present the relevant parameter estimates and interpret the parameters of the best fitting model.

Figure 13-3: Two factor model with six observed variables. Factor variances set to 1.



Regression Weights: (Group number 1 - Model 1)

	Estimate	S.E.	C.R.	P	Label
x1A <--- F1	1.03924	.08134	12.77722	***	beta1
x1B <--- F1	1.26835	.10563	12.00761	***	beta2
x1C <--- F1	.87597	.08440	10.37900	***	beta3
x2A <--- F2	.98133	.08333	11.77677	***	beta4
x2B <--- F2	1.03091	.07877	13.08756	***	beta5
x2C <--- F2	1.39521	.10253	13.60817	***	beta6

Standardized Regression Weights: (Group number 1 - Model 1)

	Estimate
X1A <--- F1	.928
X1B <--- F1	.891
X1C <--- F1	.806
X2A <--- F2	.867
X2B <--- F2	.926
X2C <--- F2	.947

Variances: (Group number 1 - Model 1)

	Estimate	S.E.	C.R.	P	Label
F1	1.000				
F2	1.000				
err_x1A	.175	.056	3.121	.002	ex1A
err_x1B	.416	.093	4.449	***	ex1B
err_x1C	.413	.065	6.377	***	ex1C
err_x2A	.318	.051	6.269	***	ex2A
err_x2B	.177	.040	4.444	***	ex2B
err_x2C	.223	.066	3.363	***	

CMIN

Model	NPAR	CMIN	DF	P	CMIN/DF
Default model	12	13.724	9	.132	1.525
Saturated model	21	.000	0		
Independence model	6	576.533	15	.000	38.436

RMR, GFI

Model	RMR	GFI	AGFI	PGFI
Default model	.172	.965	.918	.413
Saturated model	.000	1.000		
Independence model	.669	.423	.193	.302

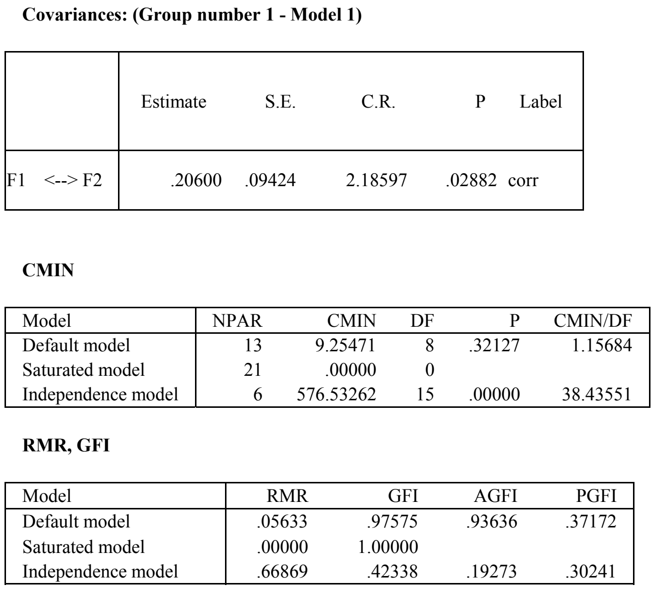
RMSEA

Model	RMSEA	LO 90	HI 90	PCLOSE
Default model	.066	.000	.133	.306
Independence model	.561	.522	.601	.000

Squared Multiple Correlations: (Group number 1 - Model 1)

	Estimate
x2C	.89717
x2B	.85708
x2A	.75187
x1C	.65032
x1B	.79467
x1A	.86063

Figure 13-4: Two factor model with six observed variables. Factor variances set to 1 and are allowed to be correlated.



Covariances: (Group number 1 - Model 1)

	Estimate	S.E.	C.R.	P	Label
F1 <--> F2	.20600	.09424	2.18597	.02882	corr

CMIN

Model	NPAR	CMIN	DF	P	CMIN/DF
Default model	13	9.25471	8	.32127	1.15684
Saturated model	21	.00000	0		
Independence model	6	576.53262	15	.00000	38.43551

RMR, GFI

Model	RMR	GFI	AGFI	PGFI
Default model	.05633	.97575	.93636	.37172
Saturated model	.00000	1.00000		
Independence model	.66869	.42338	.19273	.30241

Model	# parameters	GFI	CFI	RMSEA	Chi-square	df
1						
2						
3						
4						

Other measures that have risen in popularity are AIC and BIC.

Keep in mind that there is a trade-off between number of parameters and how well a model fits. Are there some models that clearly stick out as not being good in comparison to the rest?

Recall that it is possible to compare two models by taking the difference between the two Chi-square values and subtracting the two degrees of freedom (always taking the difference in the order "reduced minus full"). This works when one model is nested within the other (i.e., the two models are identical except that one or more paths in the full model are forced to zero in the reduced model).

Whenever you have a standard error for a parameter, you can create a confidence interval around that parameter by using the usual "plus or minus 1.96 times the standard error" rule. But as I hinted above, the tests as defined in most SEM programs give poor approximations to the standard error of parameters. It is best to test parameters using the "reduced minus full" approach I showed above; that is, set the parameter to the value of the null hypothesis (reduced) and rerun the model with the parameter as a free parameter (full model). See Gonzalez & Griffin (2001) for an explanation for why this approach should be used over the other possible approaches.

22. SEM and Hierarchical Linear Models

There is a simple connection between SEM and another kind of model that is popular these days called multilevel models, or hierarchical linear models. We used this model in LN5 for repeated-measures designs and also to address fixed and random factors in ANOVA.

In terms of the SEM path diagram, we can think of the observed "squares" as nested within the unobserved "circles." Think back to ANOVA designs when we talked about random effects and nested effects. We discussed the repeated measures design as having a subjects factor that is treated as a random effect and repeated observations being connected to a given subject. For example, if there are three observations per subject we would write an ANOVA design as

$$Y_{ij} = \mu + \alpha_i + \pi_j + \epsilon_{ij} \qquad (13\text{-}55)$$

with π a random effect representing subjects.

We saw this same design in regression where we treated each subjects as a factor and created N-1 dummy codes to properly account for the subject variance.

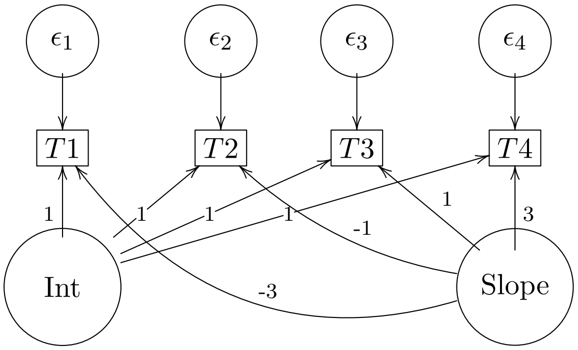
We can represent this as a structural equation model with latent variable representing the intercept term and slope terms.

In LN5 we represented this as a hierarchical linear model. In SPSS we used the MIXED command and in R we used lme (though the appendix to LN5 shows the complication within R in getting the right error term and degrees of freedom). The general syntax of the MIXED command looks like this

```
MIXED DV with X1 X2 etc XK
 /print = solution
 /method=ml
 /fixed = X1 X2 etc XK
 /random = intercept X1 | subject(ID) covtype(UN).
```

This syntax has a dependent variable DV where the intercept and the slope for X1 are random and nested within the variable code ID with covtype(UN) meaning unstructured, or free, variance and covariance between intercept and random slope. The fixed part has a common slope for each subject. If the DV is a long column of data with all the time 1 data, then all the time 2 data, etc., and the Xs contain time contrasts (like the linear contrast -3 -1 1 3 for, say, four times), then we can extend the simple repeated measures ANOVA by allowing each subject to have a unique slope and unique intercept. In this case, the simple $Y_{ij} = \mu + \alpha_i + \pi_j + \epsilon_{ij}$ model adds one more term to allow for a random slope per subject as well as the random intercept. We can estimate the variability of, say, the slope. In more complex models we can predict the variability or use the variability to predict other data. This allows us to model the heterogeneity in the data as we would model any other statistical process.

Now that the reminders are in place I switch to the SEM representation. The latent growth model can be represented in a standard SEM diagram like this:



with the unit contrast (1, 1, 1, 1) defining the latent variable called intercept and the contrast (-3,-1,1,3) defining the latent variable called slope. This example has four observed times on the same variable, labeled T1 to T4. By setting the factor paths to specific contrast values, the latent variables are defined just like contrast values in ANOVA. Each subject is assigned a contrast score (we called them "I hat" last semester) separately for intercept and for slope. Thus, each subject has his/her own slope and intercept. This SEM representation will accomplish the same thing as the MIXED model above. There are subtle differences though. The traditional latent growth curve model sets all error variances across time to be equal, but the SEM framework can estimate a more general model with separate error variances over time and more complicated error structures. Though, it is possible to set the error variances to be equal in SEM (or in the mixed model formulation to generalize the error variances to be unequal).

The lavaan syntax for a latent growth curve model is listed below. In this example we have a random intercept and a random slope (meaning each subjects is assigned their own intercept and slope). These are treated in lavaan by defining two latent factors, denoted i and s for intercept and slope, with the four observed times as the indicators and the contrast weights as fixed loadings. The intercept is defined by the unit contrast and the slope is defined by the linear contrast (-3,-1,1,3). Further, the example includes two predictors of those random intercepts and slopes (x1 and x2), something that is not easy to implement in traditional regression-based approaches and highlights the benefits of using an SEM approach that can be so flexible. We use the special growth() command in lavaan to estimate the growth curve model, which has some additional constraints imposed relative to a completely general SEM; the growth() command also sets up the mean-structure version of SEM because recall that earlier in these lecture notes I pointed out that means and intercepts need to be treated in a special way. There is an analogous package in R called blavaan for computing Bayesian versions of SEM models. The MPLUS program, available through UM's site license program, can also run general SEMs and their Bayesian analogous.

```
# define growth curve model
model.syntax <- "
  # intercept and slope with fixed coefficients
```

```
    i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
    s =~ -3*t1 + -1*t2 + 1*t3 + 3*t4

  # regressions
    i ~ x1 + x2
    s ~ x1 + x2
"

# fit model and print summary
fit1 <- growth(model.syntax, data = Demo.growth)
summary(fit1, fit.measures = TRUE)

# to get printout of random effects
lavPredict(fit1, type = "lv")
```

[output omitted]

With four times it is possible to also have a quadratic and a cubic contrast (recall number of contrasts is constrained by number of times minus 1). But, one can't necessarily make all of those terms random effects because in typical repeated measures there isn't sufficient data to estimate all the required variances. So, while the fixed effect terms can be estimated for linear, quadratic and cubic contrasts, it is likely that only the linear (and depending on the data, possibly the quadratic) can be random. If one requests random effects on all terms, one can easily run into the problem that there are more terms to estimate than there data points. We saw this idea pop up at various stages in earlier Lecture Notes. There have been papers sugesting that all parameters be treated as random effects, but those papers are easily critiques because often those problems cannot be estimated. What can be scary is that within a Bayesian framework one needs to do careful diagnostics to spot such problems of over-specifying the number of random effects in a model. The Bayesian program will produce out even when the problem technically cannot be solved; hints of problems will appear in the diagnostics and other parts of the model, you just have to bother to look and, like assumption testing, most people don't do the extra wor.

All of these formulations—repeated measures, regression, SEM, multilevel model—will yield the identical results if they are each run in a particular way and there are no missing data. But the reason for the more complicated models is that they provide more flexibility, such as each subject has his or her own slope or better ways of handling missing data such as full-information maximum likelihood (FIML) and imputation. All specialized topics that would be part of a third semester course following 613/614.

There are more general formulations of these latent growth curve models that allow one to estimate subgroups of subjects that exhibit different values on the parameters of their trajectories. You can think of this as a way to cluster subjects based on their commonality on the parameter values (e.g., slope and intercept). This idea is similar to k-means and other clustering methods we reviewed in Lecture Notes 10. These more general models that allow subject clustering in the context of SEM are called "mixture models." The mixture model idea can be applied to any type of SEM, so we can have, for example, mixture models in mediation to assess whether there are subgroups of subject that exhibit different patterns of mediation, mixture models of factor analysis to assess whether subgroups of subjects exhibit different factor structures, and, as relevant to this subsection, mixture models of growth curve models to assess whether subjects cluster on different properties of growth (such as different intercepts and slopes). This requires using a specialized program Mplus as neither SPSS nor R can handle all the variations of mixture models. If I taught a third semester, it would include these mixture models along with some machine learning models, some more Bayesian analyses and some examples of "process models" (such as computational models, agent-based models).

23. Item Response Theory (IRT)

Earlier in these lecture notes I presented reliability on binary data (e.g., there judges each rate 8 subjects). I glossed over the fact that in those examples the data were binary, yet the key measures of reliability, like Cronbach's alpha, just plugged along as though the data were continuous and normally distributed.

There are newer extensions of the concept of reliability that are collectively known as Item Response Theory (IRT). These models can be fit through SEM as well so I'll discuss them briefly.

[Need to add this subsection]

24. Coda

We end the course with the most difficult topic of the year: SEM. I outlined how we can use SEM to run (almost) all the designs we've covered this year. SEM can even be used to create new analyses such as PCA with error, or regression with latent variables, or ANOVA on PCA-like factors, regression on factors with clustering, etc. I ended these notes by showing how SEM connects to the relatively complicated technique of multilevel modeling and latent growth modeling, which ties back to the most elementary of all statistical tests—the paired t test, which is just the basic ANOVA model with a grand mean, a fixed effect α for the two times, a random effect π for subject variability treated as a blocking factor, and an error term ϵ.

Hope you enjoyed the journey. May all your results be statistically significant, may your studies be sufficiently powered and, most importantly, may you have significant hypotheses to test!

The End

Appendix 1: Amos

Amos is a program that runs SEM. It is common at the University of Michigan because there is a site license for it. SPSS bought AMOS a few years ago, so the interface between AMOS and SPSS is relatively seamless.

There are plenty of tutorials for AMOS on the web. Most rely on the graphical interface where the user enters circles, squares, arrows and double headed arrows. That is fine for the simplest of models, but for more realistic models the syntax is more efficient. The syntax is a little cumbersome (written in visual basic style), but it can do most things. Unfortunately, some of Amos' bells and whistles are not available through syntax file.

Appendix 2: R

Cronbach's α

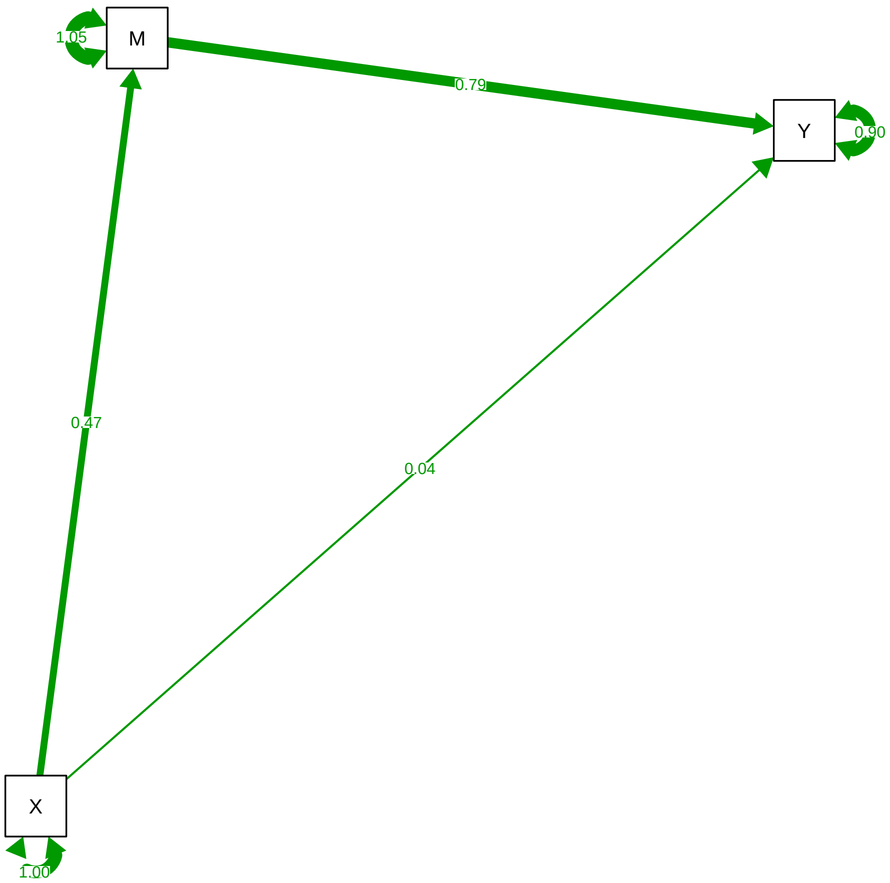
There are two packages (coefficientalpha and psy) that offer versions of Cronbach's α.

SEM

My preferred package for running SEM in R is lavaan. The syntax is very similar to Mplus so it is easy to go back an forth. There is an older package called sem that is pretty good too. For those who want to get very involved in SEM possibly even doing your own simulations and writing more advanced code to develop new methods, then the package openMX would be relevant.

Mediation tests can be worked into SEM programs. The lavaan package has a nice way of testing mediation including bootstrapping the indirect effect as a product term. Here is an example from the lavaan package documentation. One defines the linear equations, and then defines new parameters, say, 'ab' and 'total' that are functions of the estimated parameters. The default in lavaan is to estimate the standard errors of these derived parameters using approximations (see Gonzalez and Griffin, 2001, for an explanation about Wald tests), but one can specify a bootstrap version as well, which is illustrated below.

```
set.seed(1234)
X <- rnorm(100)
M <- 0.5 * X + rnorm(100)
Y <- 0.7 * M + rnorm(100)
Data <- data.frame(X = X, Y = Y, M = M)
model <- " # direct effect
             Y ~ c*X
           # mediator
             M ~ a*X
             Y ~ b*M
           # indirect effect (a*b)
             ab := a*b
           # total effect
             total := c + (a*b)
         "
fit <- sem(model, data = Data)
summary(fit)


## lavaan 0.6.13 ended normally after 1 iteration
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         5
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Y ~
##     X          (c)    0.036    0.104    0.348    0.728
##   M ~
##     X          (a)    0.474    0.103    4.613    0.000
##   Y ~
##     M          (b)    0.788    0.092    8.539    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Y                 0.898    0.127    7.071    0.000
##    .M                 1.054    0.149    7.071    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     ab                0.374    0.092    4.059    0.000
##     total             0.410    0.125    3.287    0.001


semPaths(fit, what = "par", fade = FALSE, layout = "spring")
```



Lavaan can even get bootstrap standard errors; the implementation is relatively slow however. The standard errors are different from the previous output, hence the different z-values and the corresponding p-values. Because mediation is testing the product of two parameters (the path "a times b") and standard errors for products of random variables are more complicated, the convention is to use bootstrapping to estimate the standard errors in mediation.

```
fit <- sem(model, data = Data, se = "bootstrap")
summary(fit)


## lavaan 0.6.13 ended normally after 1 iteration
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         5
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                            Bootstrap
##   Number of requested bootstrap draws             1000
##   Number of successful bootstrap draws            1000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Y ~
##     X          (c)    0.036    0.112    0.325    0.745
##   M ~
##     X          (a)    0.474    0.099    4.807    0.000
##   Y ~
##     M          (b)    0.788    0.089    8.874    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Y                 0.898    0.152    5.912    0.000
##    .M                 1.054    0.162    6.525    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     ab                0.374    0.084    4.437    0.000
##     total             0.410    0.131    3.131    0.002
```

To see confidence intervals of parameters use this command

```
parameterEstimates(fit)
```

```
##     lhs op     rhs label   est    se     z pvalue ci.lower
## 1     Y  ~       X     c 0.036 0.112 0.325  0.745   -0.156
## 2     M  ~       X     a 0.474 0.099 4.807  0.000    0.283
## 3     Y  ~       M     b 0.788 0.089 8.874  0.000    0.602
## 4     Y ~~       Y       0.898 0.152 5.912  0.000    0.605
## 5     M ~~       M       1.054 0.162 6.525  0.000     0.733
## 6     X ~~       X       0.999 0.000    NA     NA     0.999
## 7    ab :=     a*b    ab 0.374 0.084 4.437  0.000     0.218
## 8 total := c+(a*b) total 0.410 0.131 3.131  0.002     0.168
##   ci.upper
## 1    0.275
## 2    0.666
## 3    0.945
## 4    1.200
## 5    1.381
## 6    0.999
## 7    0.548
## 8    0.675
```

One can combine a factor model to assess measurement error and also a mediation model. Here is a simplified version of the classic example from Bollen.

```
model <- "
  # latent variable definitions
     ind60 =~ x1 + x2 + x3
     dem60 =~ y1 + a*y2 + b*y3 + c*y4
     dem65 =~ y5 + a*y6 + b*y7 + c*y8

  # regressions
    dem60 ~ apath* ind60
    dem65 ~ cpath*ind60 + bpath*dem60

 # define indirect effect (a*b)
             ab := apath*bpath

 # define total effect
             total := cpath + (apath*bpath)
"

fit <- sem(model, data = PoliticalDemocracy)
summary(fit, fit.measures = TRUE)
```

```
## lavaan 0.6.13 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        25
##   Number of equality constraints                     3
##
```

```
##   Number of observations                            75
##
## Model Test User Model:
##
##   Test statistic                                74.618
##   Degrees of freedom                                44
##   P-value (Chi-square)                           0.003
##
## Model Test Baseline Model:
##
##   Test statistic                               730.654
##   Degrees of freedom                                55
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.955
##   Tucker-Lewis Index (TLI)                       0.943
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1566.037
##   Loglikelihood unrestricted model (H1)      -1528.728
##
##   Akaike (AIC)                                3176.075
##   Bayesian (BIC)                              3227.059
##   Sample-size adjusted Bayesian (SABIC)       3157.721
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.096
##   90 Percent confidence interval - lower         0.057
##   90 Percent confidence interval - upper         0.133
##   P-value H_0: RMSEA <= 0.050                    0.031
##   P-value H_0: RMSEA >= 0.080                    0.775
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   ind60 =~
##     x1                1.000
##     x2                2.181    0.139   15.716    0.000
##     x3                1.819    0.152   11.958    0.000
##   dem60 =~
##     y1                1.000
##     y2         (a)    1.287    0.118   10.875    0.000
##     y3         (b)    1.174    0.107   10.930    0.000
##     y4         (c)    1.299    0.102   12.729    0.000
##   dem65 =~
##     y5                1.000
##     y6         (a)    1.287    0.118   10.875    0.000
##     y7         (b)    1.174    0.107   10.930    0.000
##     y8         (c)    1.299    0.102   12.729    0.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dem60 ~
##     ind60   (apth)    1.463    0.383    3.822    0.000
##   dem65 ~
##     ind60   (cpth)    0.464    0.223    2.082    0.037
##     dem60   (bpth)    0.887    0.074   12.053    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.082    0.020    4.179    0.000
##    .x2                0.119    0.070    1.693    0.090
##    .x3                0.467    0.090    5.174    0.000
##    .y1                1.921    0.389    4.944    0.000
##    .y2                6.620    1.186    5.582    0.000
##    .y3                5.321    0.957    5.562    0.000
##    .y4                2.917    0.607    4.803    0.000
##    .y5                2.379    0.446    5.335    0.000
##    .y6                4.343    0.803    5.408    0.000
##    .y7                3.604    0.667    5.406    0.000
##    .y8                2.936    0.586    5.013    0.000
##     ind60             0.448    0.087    5.170    0.000
##    .dem60             3.794    0.811    4.676    0.000
##    .dem65             0.120    0.208    0.577    0.564
##
```

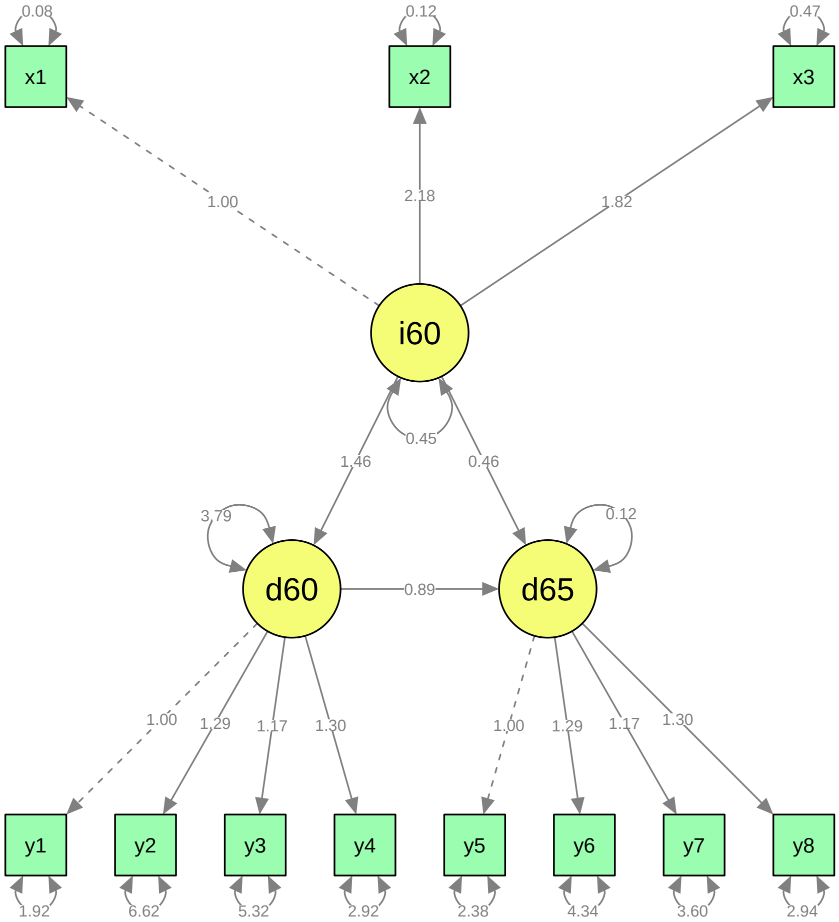
```
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     ab                1.298    0.357    3.637    0.000
##     total             1.761    0.361    4.885    0.000
```

```
parameterEstimates(fit)
```

```
##      lhs op                 rhs label   est    se      z
## 1  ind60 =~                  x1       1.000 0.000     NA
## 2  ind60 =~                  x2       2.181 0.139 15.716
## 3  ind60 =~                  x3       1.819 0.152 11.958
## 4  dem60 =~                  y1       1.000 0.000     NA
## 5  dem60 =~                  y2     a 1.287 0.118 10.875
## 6  dem60 =~                  y3     b 1.174 0.107 10.930
## 7  dem60 =~                  y4     c 1.299 0.102 12.729
## 8  dem65 =~                  y5       1.000 0.000     NA
## 9  dem65 =~                  y6     a 1.287 0.118 10.875
## 10 dem65 =~                  y7     b 1.174 0.107 10.930
## 11 dem65 =~                  y8     c 1.299 0.102 12.729
## 12 dem60  ~               ind60 apath 1.463 0.383  3.822
## 13 dem65  ~               ind60 cpath 0.464 0.223  2.082
## 14 dem65  ~               dem60 bpath 0.887 0.074 12.053
## 15    x1 ~~                  x1       0.082 0.020  4.179
## 16    x2 ~~                  x2       0.119 0.070  1.693
## 17    x3 ~~                  x3       0.467 0.090  5.174
## 18    y1 ~~                  y1       1.921 0.389  4.944
## 19    y2 ~~                  y2       6.620 1.186  5.582
## 20    y3 ~~                  y3       5.321 0.957  5.562
## 21    y4 ~~                  y4       2.917 0.607  4.803
## 22    y5 ~~                  y5       2.379 0.446  5.335
## 23    y6 ~~                  y6       4.343 0.803  5.408
## 24    y7 ~~                  y7       3.604 0.667  5.406
## 25    y8 ~~                  y8       2.936 0.586  5.013
## 26 ind60 ~~               ind60       0.448 0.087  5.170
## 27 dem60 ~~               dem60       3.794 0.811  4.676
## 28 dem65 ~~               dem65       0.120 0.208  0.577
## 29    ab :=         apath*bpath    ab 1.298 0.357  3.637
## 30 total := cpath+(apath*bpath) total 1.761 0.361  4.885
##    pvalue ci.lower ci.upper
## 1      NA    1.000    1.000
## 2   0.000    1.909    2.453
## 3   0.000    1.521    2.117
## 4      NA    1.000    1.000
## 5   0.000    1.055    1.520
## 6   0.000    0.964    1.385
## 7   0.000    1.099    1.499
## 8      NA    1.000    1.000
## 9   0.000    1.055    1.520
## 10  0.000    0.964    1.385
## 11  0.000    1.099    1.499
## 12  0.000    0.713    2.213
## 13  0.037    0.027    0.900
## 14  0.000    0.743    1.031
## 15  0.000    0.043    0.120
## 16  0.090   -0.019    0.256
## 17  0.000    0.290    0.644
## 18  0.000    1.159    2.682
## 19  0.000    4.296    8.944
## 20  0.000    3.446    7.196
## 21  0.000    1.727    4.107
## 22  0.000    1.505    3.252
## 23  0.000    2.769    5.917
## 24  0.000    2.297    4.910
## 25  0.000    1.788    4.084
## 26  0.000    0.278    0.618
## 27  0.000    2.203    5.384
## 28  0.564   -0.287    0.527
## 29  0.000    0.598    1.997
## 30  0.000    1.055    2.468
```

There are also a few packages specifically for mediation, including the mediation package that has a Bayesian implementation and the semMediation package that adds functionality to a lavaan model. You can use the semPlot package as well.

```
semPaths(fit, whatLabels = "est", color = list(lat = rgb(245,
    253, 118, maxColorValue = 255), man = rgb(155, 253, 175,
    maxColorValue = 255)))
```



Instrumental variables

The sem package has a function to compute instrumental variables analysis. Econometricians are now beginning to use R so there are more instrumental variable tools available (STATA is the economists' SPSS).

Also, check out the package MIIVsem that takes SEM models (like lavaan syntax) and expresses them in terms of instrumental variables. Here is an example using the mediation model and data described above under SEM.

```
library(MIIVsem)
model <- " # direct effect
             Y ~ c*X
           # mediator
             M ~ a*X
             Y ~ b*M
         "

miivs(model)
miive(model, Data)
```

```
> miivs(model)
Model Equation Information

 LHS RHS  Composite Disturbance MIIVs
 Y   X, M e.Y                   X, M
 M   X    e.M                   X


> miive(model,Data)

MIIVsem results

  DV  EV   Estimate  StdErr  Z     P(|Z|)  Sargan  df  P(Chi)
  Y   Int  0.164     0.096   1.70  0.09
      X    0.036     0.104   0.35  0.73
      M    0.788     0.092   8.54  0.00
  M   Int  0.037     0.104   0.36  0.72
      X    0.474     0.103   4.61  0.00
```