

**MKT 791--Research II**  
**Spring '96**  
**R. Kleine**

**SESSION 6**

**MQA II: Introducing . . . Confirmatory Factor Analysis**

**READINGS**

Steenkamp, Jan-Benedict E. M. and Hans C. M. van Trijp (1991), "The use of LISREL in Validating Marketing Constructs," *International Journal of Research in Marketing*, 8 (November), 283-299.

Hatcher, Larry (1994), *A Step-by-Step Approach to Using the SAS System for Factor Analysis and Structural Equation Modeling*, Cary, NC: The SAS Institute.

- ◆ Chapter 5: Developing Measurement Models with Confirmatory Factor Analysis

Khattree, Ravindra and Dayanand N. Naik (1995), *Applied Multivariate Statistics with SAS Software*, Cary, NC: The SAS Institute, 9-12.

Bagozzi, Richard P. and Youjae Yi (1988), "On the Evaluation of Structural Equation Models," *Journal of the Academy of Marketing Science*, 16 (1), 74-94.

Bollen, Kenneth A. (1989), *Structural Equations with Latent Variables*, New York: Wiley.

- ◆ Chapter 7: Confirmatory Factor Analysis (skim for major themes)

Review the PROC CALIS description in the *SAS/STAT User's Guide* (Version 6).

**Highly recommended**

Tanaka, J. S., A. T. Panter, Wayne C. Winborne, and G.H. Huba (1990), "Theory Testing in Personality and Social Psychology with Structural Equation Models: A Primer in 20 Questions," in Clyde Hendrick and Margaret S. Clark, eds., *Research Methods in Personality and Social Psychology*, Vol. 11, Newbury Park: Sage, 217-242.

## YOUR TURN

Perform a confirmatory (restricted) factor analysis to assess measurement quality. Use the same data on which you ran the common factor analysis last week. Use SAS's PROC CALIS. Sample control lines are provided on a separate page.

- \* I suggest that you first draw a path diagram depicting the analysis. This is useful for checking the accuracy of your equations.
- \* Be prepared for several iterations. CFAs are iterative experiences!
- \* Remember, theory, not theory-blind statistical procedures, must guide your analysis.
- \* Do your best to muddle through. We'll talk about your experiences in class.
- \* Bring your analyses and questions to class.

**Annotated Control Lines for a 6-variable 2-factor CFA Using SAS's PROC CALIS:  
Just Enough to Make You Dangerous**

The following control lines describe a two factor confirmatory factor analysis. Each factor has three indicators.

**PROC CALIS COV RESIDUAL;** 7This line invokes PROC CALIS. COV and RESIDUAL are options I strongly recommend

**VAR VAR1-VAR6;** 7List variables to use in the analysis after VAR

**LINEQS** 7No punctuation here! (LINEQS=LINear EQUations)

**VAR1=L1 F1+E1,** 7Specify the equations that define your measurement model.  
**VAR1** is an observed variable. **L1** is the arbitrary name I gave the factor loading. **F1** is the name I gave the factor. Factor names must begin with an **F**. **E1** is the name I gave the uniqueness or residual. If you use the same parameter name more than once, you specify an equality constraint. Note: a comma separates each equation

**VAR2=L2 F1+E2,**

**VAR3=L3 F1+E3,**

**VAR4= F2+E4,** 7No factor loading estimated for this item. Hence, it's fixed to 1.0.

**VAR5=L5 F2+E5,**

**VAR6=L6 F2+E6;** 7Follow the last equation with a ;

**STD** 7Again, no punctuation here.

**E1-E6=TE1-TE6, F1=1, F2=VARF2;** 7Here I tell the program to estimate the residuals (E1-E6) and to call those estimated residuals TE1-TE6. F1=1 fixes the variance of F1 to 1. F2=VARF2 tells the program to estimate the variance of F2 and call that estimate VARF2. (No mandatory prefixes here, use names that are meaningful to you).

**COV**

**F1 F2=COVF1F2;** 7Tells the program to estimate the covariance between F1 and F2. COVF1F2 is the name I supplied for the estimated covariance. If this statement is removed, the program will assume that F1 and F2 are orthogonal. The conventions described for the STD statement apply here too with this exception: The two variables you want to covary are specified to the left of the =. Your name for their covariance is specified to the right of the =.

Over for important details . . . L

## NOTES

### Naming Conventions:

- \* Each factor's name *must* begin with an **F**. Each factor must have unique name. Something descriptive is helpful.
- \* The name of each uniqueness (residual) must begin with an **E**. Each uniqueness must have a unique name--or it's not unique! I find numbering them to be most efficient.
- \* No special prefix is required for the factor loadings (the Lns in each equation above tell the program to estimate a factor loading); call them anything you want (e.g., george, loui, karen) Each name must be different--unless you want two or more factor loadings to have the same value.

If you do not specify a parameter name for a factor loading (see, e.g., equation for VAR4), the program sets its value to 1--i.e.,  $VAR4=F2+E4$  is equivalent to  $VAR4=1*F2+E4$ .

In some situations you may wish to fix a factor loading to a specific value (other than 1). Do this by inserting the desired value in the equation (e.g.,  $VAR4=.5 F2+E4$ ).

**IMPORTANT:** You must set the scale for each latent construct. There are two ways to do this; you must use one *or* the other for *each* factor, *never* use both on a single factor:

One way is to fix the variance of a latent construct (factor) to 1.0. The scale for Factor 1 was specified in this way. This is specified in the STD section. Note the  $F1=1$  in the STD section. *Use this method for exogenous latent constructs only.*

The second way to fix a latent construct's scale is to fix one factor loading to a constant value--usually 1. The scale for **F2** was set this way. Note that VAR4 has no named factor loading parameter. If you choose this fix-the-factor loading method you must then have CALIS estimate the variance of the factor--note how this is done in the STD section for F2. *This scale setting method is appropriate for both exogenous and endogenous constructs.*

*In closing:* CALIS provides parameters estimates *only* if you request them--it's a really dumb PROC.