How can I conduct exploratory factor analysis (EFA) within a confirmatory factor analysis (CFA) framework in Stata?

Authored by

stats writer

July 1, 2024

Exploratory Factor Analysis (EFA) is a statistical technique used to identify underlying factors and their relationships within a set of observed variables. Confirmatory Factor Analysis (CFA) is a method to test a pre-specified factor structure in a dataset. In Stata, it is possible to conduct EFA within a CFA framework by using the "sem" command. This command allows for the inclusion of exploratory factors in the CFA model, providing a more comprehensive analysis of the data. By incorporating EFA within the CFA framework in Stata, researchers can gain a deeper understanding of the underlying factors influencing their variables and test the validity of their proposed factor structure. This approach can enhance the accuracy and robustness of the results obtained from CFA, making it a valuable tool in research and data analysis.

How can I do EFA within a CFA framework? | Stata FAQ

EFA within a CFA framework, as the name implies, combines aspects of both EFA and CFA.
It produces a factor solution that is close to an EFA solution while providing features
found in CFA, such as standard errors, statistical tests and modification indices.

The trick to doing EFA within a CFA framework is to use the same number of constraints in CFA
as are used in EFA. If m is the number of factors in an EFA model then the number of
constraints is m2. By using m2 constraints in CFA we get a model with
the same fit as an EFA model.

We will demonstrate EFA within a CFA framework using

an artificial dataset with 500 observations on six variables. The data wee constructed to give a two factor solution.

We begin by loading the data and computing a correlation matrix.

```
use https://stats.idre.ucla.edu/stat/data/efa_cfa, clearcorrelate

(obs=500)

| y1 y2 y3 y4 y5 y6
-------------+------------------------------------------------
y1 | 1.0000
y2 | 0.5212 1.0000
y3 | 0.4719 0.5339 1.0000
y4 | -0.0008 -0.0345 -0.0101 1.0000
y5 | -0.0256 -0.0365 -0.0400 0.4297 1.0000
y6 | 0.0176 -0.0159 0.0390 0.3688 0.4268 1.0000
```

Now we are ready to run a maximum likelihood EFA model followed by an oblique rotation using the quartimin normalize option. We will also use estat common to get the correlation

between the oblique facts.

factor y1 y2 y3 y4 y5 y6, ml

(obs=500)
number of factors adjusted to 3
Iteration 0: log likelihood = -31.62904
Iteration 1: log likelihood = -3.1615516
Iteration 2: log likelihood = -.34164269
Iteration 3: log likelihood = -.34066861 (backed up)

Iteration 25: log likelihood = -.05290372

number of factors adjusted to 2
Iteration 0: log likelihood = -35.679836
Iteration 1: log likelihood = -1.103664
Iteration 2: log likelihood = -1.1024515

Factor analysis/correlation Number of obs = 500
Method: maximum likelihood Retained factors = 2
Rotation: (unrotated) Number of params = 11
Schwarz's BIC = 70.5656
Log likelihood = -1.102451 (Akaike's) AIC = 24.2049

Factor | Eigenvalue Difference Proportion Cumulative

----------+----------

Factor1 | 1.54078 0.30931 0.5558 0.5558

Factor2 | 1.23147 . 0.4442 1.0000

LR test: independent vs. saturated: chi2(15) = 593.54 Prob>chi2 = 0.0000

LR test: 2 factors vs. saturated: chi2(4) = 2.19 Prob>chi2 = 0.7015

Factor loadings (pattern matrix) and unique variances

Variable | Factor1 Factor2 | Uniqueness

----------+----------+----------

y1 | 0.6763 0.0631 | 0.5387

y2 | 0.7657 0.0354 | 0.4124

y3 | 0.6944 0.0612 | 0.5141

y4 | -0.0675 0.6060 | 0.6282

y5 | -0.0975 0.6969 | 0.5048

y6 | -0.0290 0.6080 | 0.6295

----

rotate, oblique quartimin normalize

Factor analysis/correlation Number of obs = 500

Method: maximum likelihood Retained factors = 2

Rotation: oblique quartimin (Kaiser on) Number of params = 11

Schwarz's BIC = 70.5656

Log likelihood = -1.102451 (Akaike's) AIC = 24.2049

Factor | Variance Proportion Rotated factors are correlated

----------+----------

Factor1 | 1.53686 0.5544

Factor2 | 1.23729 0.4463

LR test: independent vs. saturated: chi2(15) = 593.54 Prob>chi2 = 0.0000

LR test: 2 factors vs. saturated: chi2(4) = 2.19 Prob>chi2 = 0.7015

Rotated factor loadings (pattern matrix) and unique variances

---

Variable | Factor1 Factor2 | Uniqueness

----------+----------+----------

y1 | 0.6794 0.0117 | 0.5387

y2 | 0.7657 -0.0228 | 0.4124

y3 | 0.6972 0.0084 | 0.5141

y4 | -0.0073 0.6095 | 0.6282

y5 | -0.0281 0.7025 | 0.5048

y6 | 0.0313 0.6086 | 0.6295

Factor rotation matrix

| Factor1 Factor2

---+---

Factor1 | 0.9971 -0.0990

Factor2 | 0.0758 0.9951

---

estat common

Correlation matrix of the quartimin rotated common factors

Factors | Factor1 Factor2

---+---

Factor1 | 1

Factor2 | -.02325 1

The rotated factor solution gives us a rather clean two factor model. We note that the model
fit versus a saturated model has a chi-square of 2.19 with four degrees of freedom.
This is a very good fit for an EFA and reflects the synthetic nature of the data. We also
note the the two factors have a small correlation of -.02325.

Since the EFA solution has two factors we will need to make a total of four constraints
in the CFA model. The easiest ones are to set the factor variances to one (2 factors, 2 constraints). The other constraints
revolve around identifying anchor variables for each factor and constraining the cross
loadings for the anchor variables to be zero (2 cross loadings, 2 more constraints for a total of 4).

An anchor item has a high loading on one factor and low loadings on the remaining factors.

From the rotated solution above we see that y2 has the highest loading on Factor1
and has low loading on Factor2. We will make y2 the anchor item for Factor1 and
will constrain the loading to equal zero in Factor2. We will use y5 as the anchor
item for Factor2.

Here is the sem command for the EFA within a CFA framework.

sem (F1 -> y1 y2 y3 y4 y5@0 y6) ///
(F2 -> y1 y2@0 y3 y4 y5 y6) , ///
variance(F1@1 F2@1) standardized

Endogenous variables

Measurement: y1 y2 y3 y4 y5 y6

Exogenous variables

Latent: F1 F2

Fitting target model:

Iteration 0: log likelihood = -5064.3487 (not concave)

Iteration 1: log likelihood = -5005.6323 (not concave)
Iteration 2: log likelihood = -4997.4943 (not concave)
Iteration 3: log likelihood = -4982.0445 (not concave)
Iteration 4: log likelihood = -4978.5317 (not concave)

Iteration 15: log likelihood = -4975.0886 (not concave)
Iteration 16: log likelihood = -4975.0731
Iteration 17: log likelihood = 2163798 (not concave)

Iteration 40: log likelihood = 2163798 (not concave)
Iteration 41: log likelihood = 2163798 (not concave)
Iteration 42: log likelihood = 2163798 (not concave)
--Break--
r(1);

The sem command would have run forever if we had let it. However, since the log likelihood did not change from the 17th iteration on, we broke out of the program. There are two possible reasons for the endless iterations: 1) Either the model is not identified or 2) the starting values did not allow sem to converge on a solution. I'm betting that it is the second reason: bad starting values.

To investigate this we will run the sem command again but limit the iterations to 16
and see what the results look like.

sem (F1 -> y1 y2 y3 y4 y5@0 y6) ///
(F2 -> y1 y2@0 y3 y4 y5 y6) , ///
variance(F1@1 F2@1) standardized iterate(16)

Endogenous variables

Measurement: y1 y2 y3 y4 y5 y6

Exogenous variables

Latent: F1 F2

Fitting target model:

Iteration 0: log likelihood = -5064.3487 (not concave)
Iteration 1: log likelihood = -5005.6323 (not concave)
Iteration 2: log likelihood = -4997.4943 (not concave)

Iteration 15: log likelihood = -4975.0886 (not concave)
Iteration 16: log likelihood = -4975.0731
convergence not achieved

Structural equation model Number of obs = 500

```
Estimation method = ml
Log likelihood = -4975.0731

( 1) _cons = 1
( 2) _cons = 1
------------------------------------------------------------------------------
             | OIM
             | Coef. Std. Err. z P>|z| Standardized
-------------+----------------------------------------------------------------
Measurement |
  y1 <- |
          F1 | .6708773 .0557835 12.03 0.000 .5615436 .7802109
          F2 | .0120947 .042074 0.29 0.774 -.0703689 .0945583
       _cons | -.0157292 .0447242 -0.35 0.725 -.103387 .0719286
-----------+----------------------------------------------------------------
y2 chi2 = 0.0000
Warning: convergence not achieved
```

In the results above we see that all of the loadings are between 0 and 1 except for variable y6.
Also the residual variance for y6 is much higher than any of the other residual variances. I think we
can fix the problem by setting the initial starting value for the loading on Factor1 to 0 and

on Factor2 to 0.5. We do this by including init in the sem command as seen below:

```
sem (F1 -> y1 y2 y3 y4 y5@0 (y6, init(0.0))) ///
(F2 -> y1 y2@0 y3 y4 y5 (y6, init(0.5))) , ///
variance(F1@1 F2@1) standardized
```

```
Endogenous variables

Measurement: y1 y2 y3 y4 y5 y6

Exogenous variables

Latent: F1 F2

Fitting target model:

Iteration 0: log likelihood = -5467.1006 (not concave)
Iteration 1: log likelihood = -5087.7064 (not concave)

Iteration 8: log likelihood = -4905.7634
Iteration 9: log likelihood = -4905.7634

Structural equation model Number of obs = 500
Estimation method = ml
```

Log likelihood = -4905.7634

(1) _cons = 1
(2) _cons = 1

--
| OIM
Standardized | Coef. Std. Err. z P>|z|
-------------+--
Measurement |
y1 chi2 = 0.6981

This time sem converged on a solution. All of the factor loadings and residual variances look reasonable. The model fit versus a saturated model has a chi-square of 2.20 with 4 df.
This value is almost identical to the chi-square of 2.19 with 4 df from the maximum likelihood EFA model we ran at the beginning.

Before comparing the EFA within a CFA framework with the EFA solution, let's look at some of the various indicators of goodness of fit.

estat gof, stat(all)

```
------------------------------------------------------------------------------
Fit statistic | Value Description
---------------------+--------------------------------------------------------
Likelihood ratio |
chi2_ms(4) | 2.205 model vs. saturated
p > chi2 | 0.698
chi2_bs(15) | 596.921 baseline vs. saturated
p > chi2 | 0.000
---------------------+--------------------------------------------------------
Population error |
RMSEA | 0.000 Root mean squared error of approximation
90% CI, lower bound | 0.000
upper bound | 0.051
pclose | 0.947 Probability RMSEA <= 0.05
---------------------+--------------------------------------------------------
Information criteria |
AIC | 9857.527 Akaike's information criterion
BIC | 9954.463 Bayesian information criterion
---------------------+--------------------------------------------------------
Baseline comparison |
CFI | 1.000 Comparative fit index
TLI | 1.012 Tucker-Lewis index
---------------------+--------------------------------------------------------
```

Size of residuals |
SRMR | 0.007 Standardized root mean squared residual
CD | 0.925 Coefficient of determination

It is not surprising that the gof values all look good considering the artificial nature of the data.

Now, let's compare the EFA within CFA solution to the EFA solution.

Summary of results

efa rotated maximum
efa within cfa loadings likelihood loading
F1 F2 Factor1 Factor2
y1 .6814311 .0319918 0.6794 0.0117
y2 .7665428 0 0.7657 -0.0228
y3 .6991976 .0292474 0.6972 0.0084
y4 .0171268 .6110693 -0.0073 0.6095
y5 0 .7036822 -0.0281 0.7025
y6 .0557532 .6112825 0.0313 0.6086

factor: LR test: 2 factors vs. saturated: chi2(4) = 2.19
Prob>chi2 = 0.7015

sem: LR test of model vs. saturated: chi2(4) = 2.20 Prob>chi2 = 0.6981

The EFA within a CFA framework results appear very close to the rotated maximum likelihood EFA results while having the added benefits of standard errors, statistical tests and access to modification indices, if needed.