Getting Started in Factor Analysis
(v. 1.5)

Oscar Torres-Reyna
otorres@princeton.edu

August 2008
http://www.princeton.edu/~otorres/
Factor analysis: intro

Factor analysis is used mostly for data reduction purposes:

- To get a small set of variables (preferably uncorrelated) from a large set of variables (most of which are correlated to each other)
- To create indexes with variables that measure similar things (conceptually).

**Two types of factor analysis**

**Exploratory**

It is exploratory when you do not have a pre-defined idea of the structure or how many dimensions are in a set of variables.

**Confirmatory.**

It is confirmatory when you want to test specific hypothesis about the structure or the number of dimensions underlying a set of variables (i.e. in your data you may think there are two dimensions and you want to verify that).
Factor analysis: step 1

To run factor analysis use the command `factor` (type `help factor` for more details).

Total variance accounted by each factor. The sum of all eigenvalues = total number of variables. When negative, the sum of eigenvalues = total number of factors (variables) with positive eigenvalues. Kaiser criterion suggests to retain those factors with eigenvalues equal or higher than 1.

Difference between one eigenvalue and the next.

Since the sum of eigenvalues = total number of variables. Proportion indicate the relative weight of each factor in the total variance. For example, 1.54525/5=0.3090. The first factor explains 30.9% of the total variance.

Cumulative shows the amount of variance explained by n+(n-1) factors. For example, factor 1 and factor 2 account for 57.55% of the total variance.

Uniqueness is the variance that is ‘unique’ to the variable and not shared with other variables. It is equal to 1 – communality (variance that is shared with other variables). For example, 61.57% of the variance in ‘ideol’ is not shared with other variables in the overall factor model. On the contrary ‘owner’ has low variance not accounted by other variables (28.61%). Notice that the greater ‘uniqueness’ the lower the relevance of the variable in the factor model.

Factor loadings are the weights and correlations between each variable and the factor. The higher the load the more relevant in defining the factor’s dimensionality. A negative value indicates an inverse impact on the factor. Here, two factors are retained because both have eigenvalues over 1. It seems that ‘owner’ and ‘competition’ define factor1, and ‘equality’, ‘respon’ and ‘ideol’ define factor2.

Uniqueness is the variance that is ‘unique’ to the variable and not shared with other variables. It is equal to 1 – communality (variance that is shared with other variables). For example, 61.57% of the variance in ‘ideol’ is not shared with other variables in the overall factor model. On the contrary ‘owner’ has low variance not accounted by other variables (28.61%). Notice that the greater ‘uniqueness’ the lower the relevance of the variable in the factor model.

Factor loadings are the weights and correlations between each variable and the factor. The higher the load the more relevant in defining the factor’s dimensionality. A negative value indicates an inverse impact on the factor. Here, two factors are retained because both have eigenvalues over 1. It seems that ‘owner’ and ‘competition’ define factor1, and ‘equality’, ‘respon’ and ‘ideol’ define factor2.
After running `factor` you need to rotate the factor loads to get a clearer pattern, just type `rotate` to get a final solution.

By default the rotation is varimax which produces orthogonal factors. This means that factors are not correlated to each other. This setting is recommended when you want to identify variables to create indexes or new variables without inter-correlated components.

Same description as in the previous slide with new composition between the two factors. Still both factors explain 57.55% of the total variance observed.

The pattern matrix here offers a clearer picture of the relevance of each variable in the factor. Factor1 is mostly defined by ‘owner’ and ‘competition’ and factor2 by ‘equality’, ‘respon’ and ‘ideol’.

This is a conversion matrix to estimate the rotated factor loadings (RFL):

\[ \text{RFL} = \text{Factor loadings} \times \text{Factor rotation} \]

**NOTE**: If you want the factors to be correlated (oblique rotation) you need to use the option `promax` after `rotate`: `rotate, promax`

Type `help rotate` for details. See [http://www.ats.ucla.edu/stat/stata/output/fa_output.htm](http://www.ats.ucla.edu/stat/stata/output/fa_output.htm) for more info.

Thank you to Jeannie-Marie S. Leoutsakos for useful feedback.
To create the new variables, after `factor`, rotate you type `predict`.

```
predict factor1 factor2 /*or whatever name you prefer to identify the factors*/
```

Another option (called *naïve* by some) could be to create indexes out of each cluster of variables. For example, ‘owner’ and ‘competition’ define one factor. You could aggregate these two to create a new variable to measure ‘market oriented attitudes’. On the other hand you could aggregate ‘ideol’, ‘equality’ and ‘respon’ to create an index to measure ‘egalitarian attitudes’. Since all variables are in the same valence (liberal for small values, capitalist for larger values), we can create the two new variables as

```
gen market = (owner + competition)/2
ngen egalitatariran = (ideol + equality + respon)/3
```
The main sources/references for this section are:

**Books**


**Online**

- StatNotes: [http://faculty.chass.ncsu.edu/garson/PA765/factor.htm](http://faculty.chass.ncsu.edu/garson/PA765/factor.htm)


- UCLA Resources: [http://www.ats.ucla.edu/stat/stata/output/fa_output.htm](http://www.ats.ucla.edu/stat/stata/output/fa_output.htm)