

EFA in R

lavaan

Full documentation available at (<https://cran.r-project.org/web/packages/lavaan/lavaan.pdf>)

lavaan Defaults to Know About

As in Mplus, this varies by the model being estimated, we'll go over these as needed but all is noted in the lavaan official documentation. Many are identical to Mplus' defaults, but it's worth checking them first. One to know here is that, just Mplus uses MLR by default, so does lavaan.

Some Useful lavaan Notations

The “=~” operator can be used to define (continuous) latent variables. This is to define a reflexive factor.

The “~~” (‘double tilde’) operator specifies (residual) variances of an observed or latent variable, or a set of covariances.

The “<~” operator can be used to define a formative factor.

The “|” operator can be used to define the thresholds of categorical endogenous variables.

As in Mplus, you can have multiple variables on the left side of the operator. If you do, you will need to add “+” in between them (if you list them without the “+” (as you would in Mplus), you will get an error message).

Fixing covariances in lavaan

Building on the above, you can specify an orthogonal (zero) covariance between two latent or observed variables: `f1 ~~ 0*f2`

EFA in R

EFA in R isn't as straightforward as in Mplus, but it is possible. You will need the “psych” package for this.

```
library(psych)  # for EFA
library(GPArotation)  # for the loadings
```

Load file

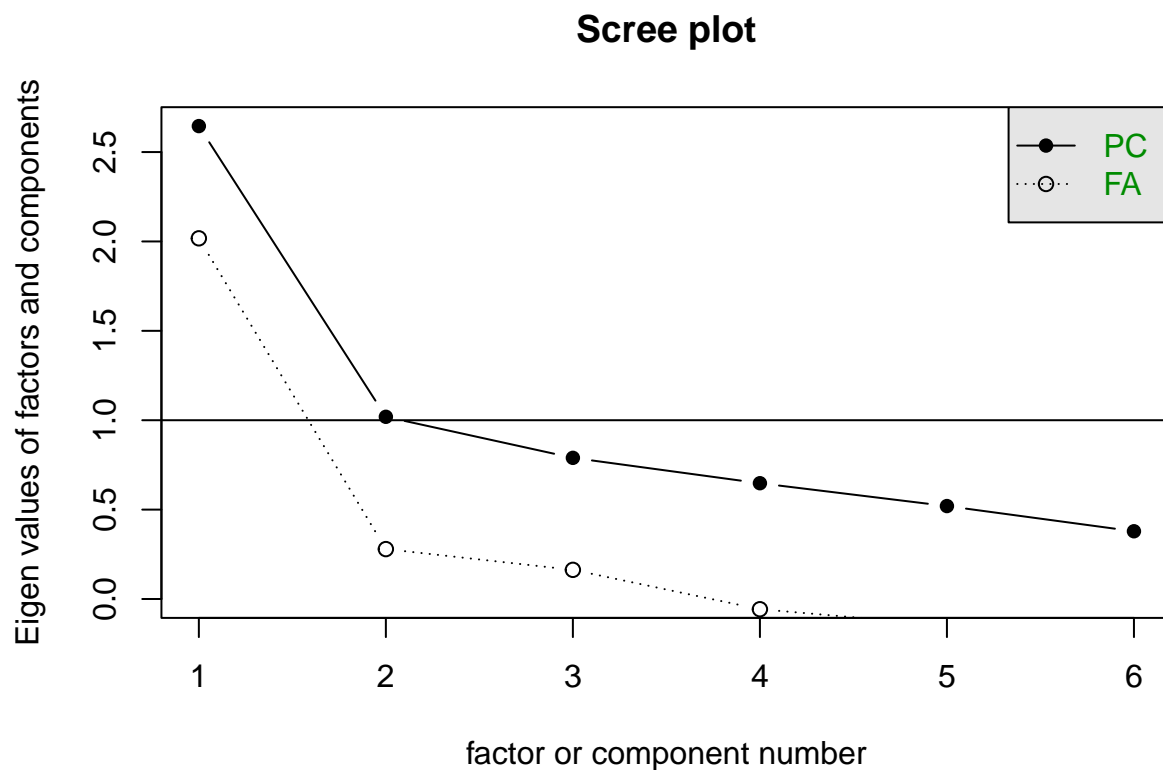
```
efa_data <- read.table("IH validation 1-21.dat")
```

EFA with “psych” package

Documentation: <https://www.rdocumentation.org/packages/psych/versions/2.1.3/topics/fa>

```
# The data can be a correlation matrix or a data matrix.  
# If data, then correlations use pairwise deletions. See  
# below for options with missing data.
```

```
screes(efa_data, factors = TRUE)
```



```
# pc = principal components - can remove by adding 'pc = F'
```

```
efa_model1 <-  
  fa(  
    efa_data, # raw data, corr or cov matrix  
    nfactors = 1, # default is 1  
    rotate = "oblimin", # default is "oblimin"  
    fm = 'uls'  
  )  
  
efa_model1$loadings
```

```
##
## Loadings:
##   ULS1
## V1 0.440
## V2 0.710
## V3 0.433
## V4 0.613
## V5 0.670
## V6 0.555
##
##           ULS1
## SS loadings 2.017
## Proportion Var 0.336
```

```
efa_model2 <-
  fa(
    efa_data, # raw data, corr or cov matrix
    nfactors = 2, # default is 1
    rotate = "oblimin", # default is "oblimin"
    fm = 'uls'
  )
```

```
efa_model2$loadings
```

```
##
## Loadings:
##   ULS1  ULS2
## V1      0.998
## V2 0.634 0.132
## V3 0.531 -0.123
## V4 0.455 0.262
## V5 0.775
## V6 0.507
##
##           ULS1  ULS2
## SS loadings 1.750 1.110
## Proportion Var 0.292 0.185
## Cumulative Var 0.292 0.477
```

```
efa_model2$values # get eigen values
```

```
## [1] 2.15170293 0.79095576 0.19390606 -0.02501665 -0.07425232 -0.09634264
```

```
residuals(efa_model2) # the diagonal is the variances
```

```
##   V1   V2   V3   V4   V5   V6
## V1 0.00
## V2 0.01 0.52
## V3 -0.01 -0.05 0.75
## V4 0.00 -0.04 0.05 0.64
## V5 0.00 0.08 -0.01 -0.05 0.44
## V6 -0.01 -0.03 0.04 0.07 -0.05 0.71
```

```
# structure(efa_model2)
# with structure(), can get all oblimin loadings,
# u2 (uniquenesses), h2 (communalities),
# com (complexity).
```

If you have missing data

psych uses pairwise deletion if data is missing. If you have missing data, you have two options to address it. One is to input the mean or median using psych package syntax. Another is to use FIML to estimate missing data. To do this, you first need to create a covariance or correlation table using FIML, then you use this matrix for the EFA with the psych package.

Note that missing data is tricky with EFA because the solution is data-driven, which means the values you use can alter the results.

The option below allows to create the matrix with FIML

```
efa_matrix <- corFiml(efa_data, covar = FALSE, show = FALSE)

# show = F means that you will do FIML,
# show = true means only showing missingness patterns
# (also useful) but not doing FIML.

# You can specify with columns to use without creating
# a new dataframe
# (e.g., corFiml(psychTools::efa_data[1:3], show = FALSE)).

efa_model <-
  fa(
    efa_matrix,
    nfactors = 2, # default is 1
    rotate = "oblimin", # default is "oblimin"
    fm = 'uls'
  )

efa_model$loadings
```

```
##
## Loadings:
##      ULS1   ULS2
## V1      0.998
## V2  0.634  0.132
## V3  0.531 -0.123
## V4  0.455  0.262
## V5  0.775
## V6  0.507
##
##              ULS1  ULS2
## SS loadings   1.750 1.110
## Proportion Var 0.292 0.185
## Cumulative Var 0.292 0.477
```

If you have categorical indicators

Muthen & Muthen recommend weighted least squares (WLS) when you have many factors and not so many factor indicators. They recommend maximum likelihood (ML, MLR) when you have few factors and many factor indicators. Both MLR and WLS can deal with categorical and continuous outcomes.

Mplus resources for EFA

Not just code, the stats too

<http://www.statmodel.com/discussion/messages/8/8.html>