EFA in lavaan

lavaan (LAatent VAriable ANalysis)

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

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).

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.

- Default estimator it maximum likelihood;
- The factor loading of the first indicator of a latent variable is fixed to 1;
- Residual variances are freely estimated;
- All exogenous variables are allowed to covary.

Fixing covariances in lavaan

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

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

Load packages

```
library(lavaan)  # for the loadings
```

```
## This is lavaan 0.6-9
## lavaan is FREE software! Please report any bugs.
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

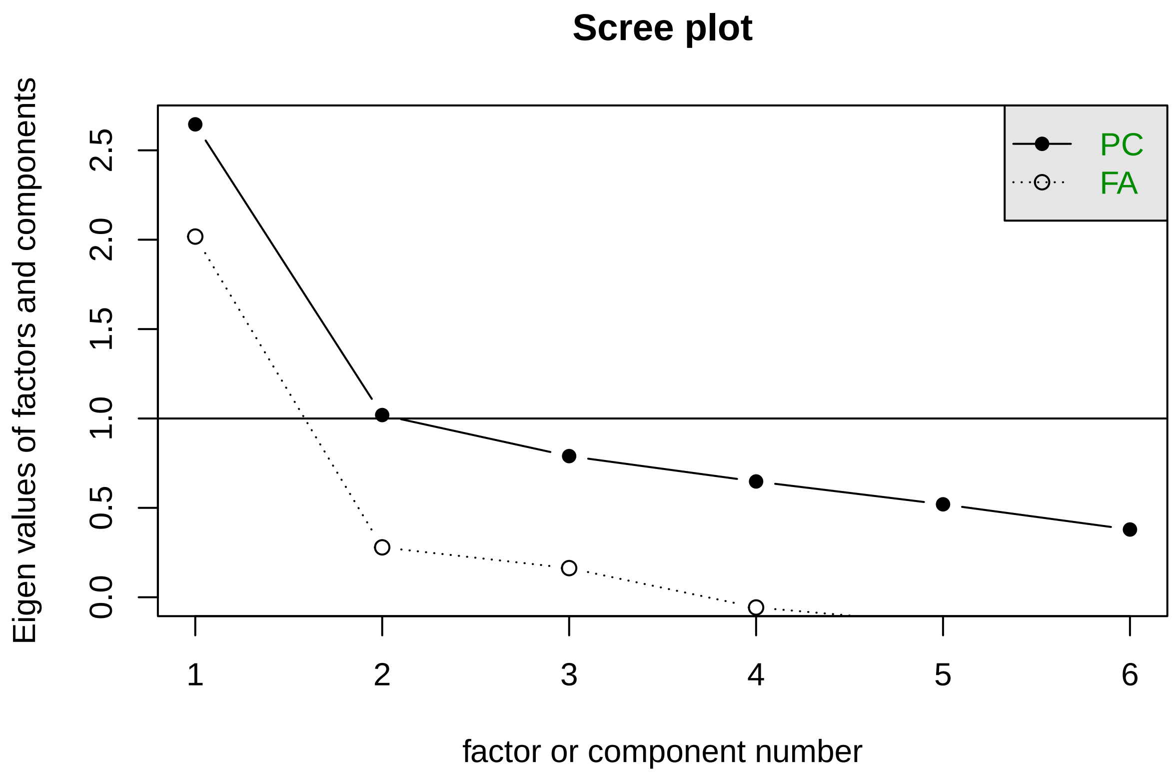
```
## The following object is masked from 'package:lavaan':
##
##     cor2cov
```

```
# psych::alpha()
```

Load file

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

```
scree(efa_data, factors = TRUE)  # get scree plot
```



Specify the model

```
# these are exploratory blocks (you can give them any
# name), you'll need to specify them
efa_model <- "
              efa(\"block1\")*F1 =~ V1 + V2 + V3 + V4 + V5 + V6
              efa(\"block1\")*F2 =~ V1 + V2 + V3 + V4 + V5 + V6
              "
```

Estimate the Model

```
efa_f2 <- sem(model = efa_model, data = efa_data, rotation = "oblimin",
    estimator = "MLR")
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
```

Request the Output

```
summary(efa_f2, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-9 ended normally after 27 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        17
##
##   Rotation method                       OBLIMIN OBLIQUE
##   Oblimin gamma                                      0
##   Rotation algorithm (rstarts)                GPA (100)
##   Standardized metric                             TRUE
##   Row weights                                     None
##
##   Number of observations                           200
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                13.105      14.567
##   Degrees of freedom                                 4           4
##   P-value (Chi-square)                           0.011       0.006
##   Scaling correction factor                                  0.900
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               260.859     203.404
##   Degrees of freedom                                15          15
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.282
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.963       0.944
##   Tucker-Lewis Index (TLI)                       0.861       0.790
##
##   Robust Comparative Fit Index (CFI)                         0.961
##   Robust Tucker-Lewis Index (TLI)                            0.852
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1437.700   -1437.700
##   Scaling correction factor                                  1.473
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -1431.147   -1431.147
##   Scaling correction factor                                  1.364
##       for the MLR correction
##
##   Akaike (AIC)                                2909.400    2909.400
##   Bayesian (BIC)                              2965.471    2965.471
```

```
##   Sample-size adjusted Bayesian (BIC)        2911.613    2911.613
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.107       0.115
##   90 Percent confidence interval - lower        0.046       0.053
##   90 Percent confidence interval - upper        0.173       0.185
##   P-value RMSEA <= 0.05                         0.060       0.045
##
##   Robust RMSEA                                              0.109
##   90 Percent confidence interval - lower                    0.053
##   90 Percent confidence interval - upper                    0.172
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.038       0.038
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 =~ block1
##     V1                1.402    1.997    0.702    0.483    1.402    1.356
##     V2                0.065    0.139    0.467    0.641    0.065    0.084
##     V3               -0.080    0.118   -0.677    0.498   -0.080   -0.099
##     V4                0.168    0.326    0.514    0.607    0.168    0.181
##     V5               -0.063    0.097   -0.648    0.517   -0.063   -0.073
##     V6                0.033    0.147    0.222    0.825    0.033    0.034
##   F2 =~ block1
##     V1                0.001    0.006    0.146    0.884    0.001    0.001
##     V2                0.535    0.101    5.302    0.000    0.535    0.693
##     V3                0.394    0.111    3.551    0.000    0.394    0.491
##     V4                0.434    0.175    2.484    0.013    0.434    0.470
##     V5                0.675    0.112    6.022    0.000    0.675    0.781
##     V6                0.486    0.113    4.311    0.000    0.486    0.503
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 ~~
##     F2                0.273    0.337    0.809    0.418    0.273    0.273
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1               -0.897    5.601   -0.160    0.873   -0.897   -0.839
##    .V2                0.286    0.047    6.053    0.000    0.286    0.481
##    .V3                0.500    0.086    5.810    0.000    0.500    0.776
##    .V4                0.598    0.090    6.627    0.000    0.598    0.700
##    .V5                0.309    0.101    3.075    0.002    0.309    0.415
##    .V6                0.687    0.093    7.370    0.000    0.687    0.737
##     F1                1.000                               1.000    1.000
```

```
##     F2                1.000                              1.000    1.000
```

```r
# lavInspect() is a helpful function to get additional
# informaiton on the model

# Here, you can see the parameter numbers (as in Mplus'
# TECH1)
lavInspect(efa_f2)
```

```
## $lambda
##    F1 F2
## V1  1  0
## V2  2  7
## V3  3  8
## V4  4  9
## V5  5 10
## V6  6 11
##
## $theta
##    V1 V2 V3 V4 V5 V6
## V1 12
## V2  0 13
## V3  0  0 14
## V4  0  0  0 15
## V5  0  0  0  0 16
## V6  0  0  0  0  0 17
##
## $psi
##    F1 F2
## F1 0
## F2 0  0
```

```r
# By asking for 'sampstat' you can see the observed
# var/covar matrix (as in Mplus)
lavInspect(efa_f2, "sampstat")
```

```
## $cov
##    V1    V2    V3    V4    V5    V6
## V1 1.069
## V2 0.292 0.595
## V3 0.043 0.165 0.645
## V4 0.402 0.257 0.211 0.854
## V5 0.167 0.385 0.250 0.272 0.745
## V6 0.247 0.254 0.231 0.344 0.289 0.933
```

```r
# There are MANY other options. To see residual var/covar
# matrix:
lavInspect(efa_f2, "resid")
```

```
## $cov
##    V1     V2     V3     V4     V5     V6
## V1  0.000
```

```
## V2 -0.004  0.000
## V3  0.004 -0.036  0.000
## V4  0.000 -0.018  0.045  0.000
## V5 -0.003  0.026  0.000 -0.034  0.000
## V6  0.015 -0.021  0.049  0.102 -0.034  0.000
```

```
# To see R2 values (note there's an NA for V1 because it's
# fixed to 1 by default)
lavInspect(efa_f2, "rsquare")
```

```
##    V1    V2    V3    V4    V5    V6
##    NA 0.519 0.224 0.300 0.585 0.263
```

```
# ... And lots more
```

You can also run an EFA using the cfa() function of lavaan, by specifying auto.efa = TRUE

```
efa_f2 <- cfa(model = efa_model, data = efa_data, rotation = "oblimin",
    estimator = "MLR", auto.efa = TRUE)
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
```

```
summary(efa_f2, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-9 ended normally after 27 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        17
##
##   Rotation method                      OBLIMIN OBLIQUE
##   Oblimin gamma                                      0
##   Rotation algorithm (rstarts)               GPA (100)
##   Standardized metric                             TRUE
##   Row weights                                     None
##
##   Number of observations                           200
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                13.105      14.567
##   Degrees of freedom                                 4           4
##   P-value (Chi-square)                           0.011       0.006
##   Scaling correction factor                                  0.900
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
```

```
##
##   Test statistic                               260.859     203.404
##   Degrees of freedom                                15          15
##   P-value                                        0.000       0.000
##   Scaling correction factor                                   1.282
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.963       0.944
##   Tucker-Lewis Index (TLI)                       0.861       0.790
##
##   Robust Comparative Fit Index (CFI)                         0.961
##   Robust Tucker-Lewis Index (TLI)                            0.852
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1437.700   -1437.700
##   Scaling correction factor                                   1.473
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -1431.147   -1431.147
##   Scaling correction factor                                   1.364
##       for the MLR correction
##
##   Akaike (AIC)                                2909.400    2909.400
##   Bayesian (BIC)                              2965.471    2965.471
##   Sample-size adjusted Bayesian (BIC)         2911.613    2911.613
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.107       0.115
##   90 Percent confidence interval - lower         0.046       0.053
##   90 Percent confidence interval - upper         0.173       0.185
##   P-value RMSEA <= 0.05                          0.060       0.045
##
##   Robust RMSEA                                               0.109
##   90 Percent confidence interval - lower                     0.053
##   90 Percent confidence interval - upper                     0.172
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.038       0.038
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 =~ block1
##     V1                1.402    1.997    0.702    0.483    1.402    1.356
##     V2                0.065    0.139    0.467    0.641    0.065    0.084
##     V3               -0.080    0.118   -0.677    0.498   -0.080   -0.099
```

```
##     V4                0.168    0.326    0.514    0.607    0.168    0.181
##     V5               -0.063    0.097   -0.648    0.517   -0.063   -0.073
##     V6                0.033    0.147    0.222    0.825    0.033    0.034
##   F2 =~ block1
##     V1                0.001    0.006    0.146    0.884    0.001    0.001
##     V2                0.535    0.101    5.302    0.000    0.535    0.693
##     V3                0.394    0.111    3.551    0.000    0.394    0.491
##     V4                0.434    0.175    2.484    0.013    0.434    0.470
##     V5                0.675    0.112    6.022    0.000    0.675    0.781
##     V6                0.486    0.113    4.311    0.000    0.486    0.503
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 ~~
##     F2                0.273    0.337    0.809    0.418    0.273    0.273
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1               -0.897    5.601   -0.160    0.873   -0.897   -0.839
##    .V2                0.286    0.047    6.053    0.000    0.286    0.481
##    .V3                0.500    0.086    5.810    0.000    0.500    0.776
##    .V4                0.598    0.090    6.627    0.000    0.598    0.700
##    .V5                0.309    0.101    3.075    0.002    0.309    0.415
##    .V6                0.687    0.093    7.370    0.000    0.687    0.737
##     F1                1.000                               1.000    1.000
##     F2                1.000                               1.000    1.000
```

Finally, (among other options) you can use the "psych" package to do your EFA

```r
library(psych)  # for EFA
```

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

```
## Loading required namespace: GPArotation
```

```r
summary(efa_model)
```

```
##
## Factor analysis with Call: fa(r = efa_data, nfactors = 2, rotate = "oblimin", fm = "uls")
##
## Test of the hypothesis that 2 factors are sufficient.
## The degrees of freedom for the model is 4  and the objective function was  0.07
## The number of observations was  200  with Chi Square =  14.25  with prob <  0.0065
##
## The root mean square of the residuals (RMSA) is  0.04
```

```
## The df corrected root mean square of the residuals is  0.08
##
## Tucker Lewis Index of factoring reliability =  0.839
## RMSEA index =  0.113  and the 10 % confidence intervals are  0.054 0.18
## BIC =  -6.94
##  With factor correlations of
##      ULS1 ULS2
## ULS1 1.00 0.35
## ULS2 0.35 1.00
```

```
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
```

```
efa_model$value # eigen values
```

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

```
efa_model$communality # communalities for items
```

```
##        V1        V2        V3        V4        V5        V6
## 0.9968896 0.4776948 0.2519209 0.3581770 0.5622910 0.2939798
```

```
# needs to be a matrix for this one so we'll practice
# making a matrix in which we use FIML to account for
# missing data

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

# show = F means that you will do FIML, show = true means
# only showing missingess patterns (also useful) but not
# doing FIML. covar = FALSE means we're getting a
# correlation (and not a covariance) matrix

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

```r
pca_model <- principal(r = efa_matrix, nfactors = 2, rotate = "oblimin")

pca_model$values  # Eigen values as we saw in Mplus
```

```
## [1] 2.6452188 1.0191197 0.7895048 0.6475670 0.5198274 0.3787623
```

```r
pca_model$communality
```

```
##        V1        V2        V3        V4        V5        V6
## 0.7999020 0.5771754 0.6491529 0.5707061 0.6297183 0.4376838
```

“‘