

# Hierarchical Factor Analysis.

As published in Benchmarks RSS Matters, April 2013

<http://web3.unt.edu/benchmarks/issues/2013/04/rss-matters>

Jon Starkweather, PhD

Jon Starkweather, PhD  
jonathan.starkweather@unt.edu  
Consultant  
**Research and Statistical Support**



<http://www.unt.edu>



<http://www.unt.edu/rss>

RSS hosts a number of “Short Courses”.  
A list of them is available at:  
<http://www.unt.edu/rss/Instructional.htm>

Those interested in learning more about R or how to use it can find information here:  
[http://www.unt.edu/rss/class/Jon/R\\_SC](http://www.unt.edu/rss/class/Jon/R_SC)

# Hierarchical Factor Analysis.

Lately, the issue of how to fit a hierarchical factor model has come up a few times and therefore, we thought it might be useful to do an article on the subject. We have had an **R** script on the **R** short course page for this subject; but perhaps a narrative or article format might be more helpful. First, we should note that hierarchical factor models are also known as higher order factor models - the term 'higher order' is often preferred over 'hierarchical' in this situation because 'higher order' better distinguishes these models from any kind of hierarchical mixed effects models (e.g., hierarchical linear modeling). Second, it is important to note that we are discussing factor analysis in this article, which is to say; modeling latent variable structure. As such, we will not be using a components extraction technique (e.g., principal components extraction) and instead will be using a factor extraction technique (e.g., minimum residual extraction, maximum likelihood extraction, etc.). For more information on the differences between component extraction and factor extraction, see the Special Issue (1990) listed in the Reference/Resources section at the end of this article (all 13 articles of which are available online at the link provided).

## Simulated Data

Using simulated data will allow us to verify we are conducting the analysis correctly. Simply put, we should be able to retrieve the parameters (e.g., factor loadings) we used to construct the data (model). However, given the generation process used for simulating data, the actual data (i.e. individual score values) will be different each time the script below is run unless we specify a seed number. To ensure we are able to replicate the exact numbers below, we use the 'set.seed' function which specifies a Random Number Generator (RNG). Below, we are using the 'psych' (Revelle, 2013) package, which has dedicated functions for simulating this type of data (e.g., hierarchical structure). The psych package requires the 'mvrnorm' function (for creating random deviates from a multivariate normal distribution) which comes from the 'MASS' (Ripley, Venables, Hornik, Gebhardt, & Firth, 2013) package. We will also use the 'GPArotation' (Bernaards, & Jennrich, 2013) package to apply an oblique rotation strategy when we begin to fit the model. Notice below, the input script is printed in red Courier New font and the output is printed in dark blue Courier New font while the text of this article is printed in black Times New Roman font.

First, load the necessary packages and set the seed (which will allow us to replicate the output).

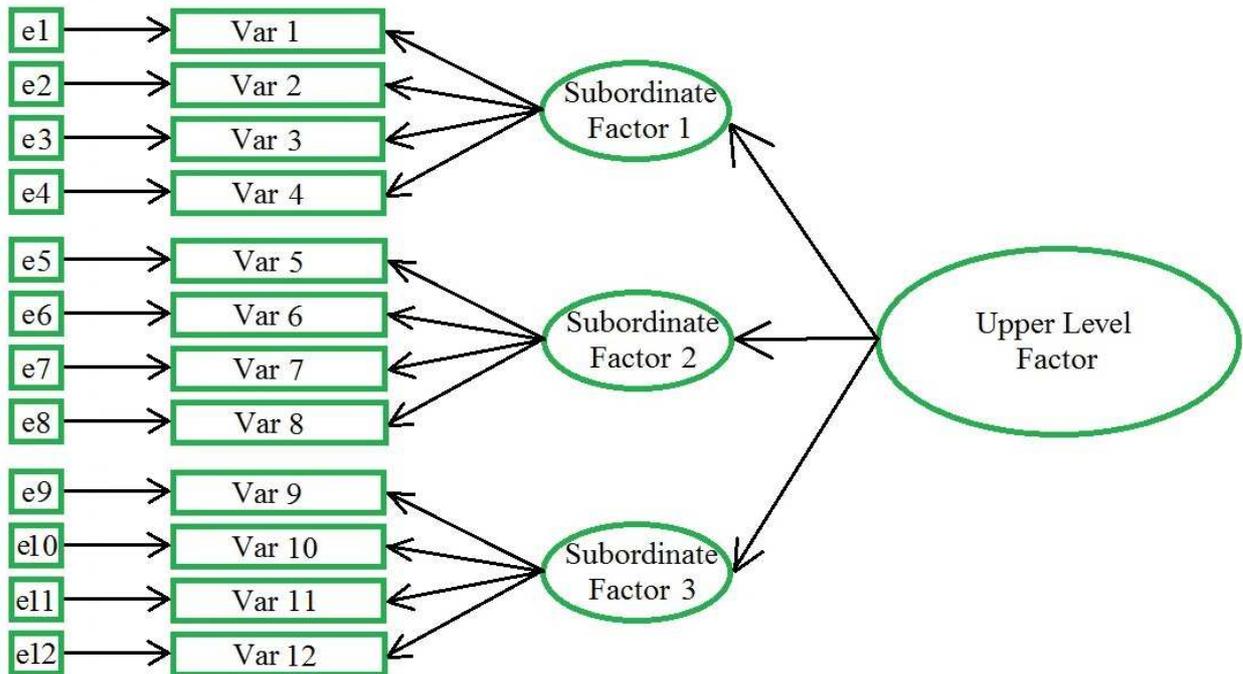
```
library(psych)
library(MASS)
library(GPArotation)
set.seed(20130403)
```

Before we begin simulating the data from a specified model, it may be helpful to create a diagram of the model we want to specify (and use to generate the data). Of course, a latent factor model assumes the Classical Test Theory (CTT) model of measurement. The CTT states that each observed variable ( $O$ ; or item response, or question response from a survey) is the result of the combination of the true scores ( $T$ ; the actual latent amount of the thing being measured) and some errors ( $E$ ; e.g., measurement error, model specification error, sampling bias, etc.). In simple equation form,

the CTT can be stated as:

$$O = T + E \quad (1)$$

where the  $O$  represents the observed score, which equals the combination of  $T$  (true score) and  $E$  (error). A figure below contains the factor model we will be using for this example (note: the CTT is reinforced by this style of diagram because the arrows accurately reflect the direction of causality - observed variables, or scores, are a result of true scores [latent factor] and errors).



Next, we create a matrix object which contains the loadings between our three subordinate factors and our single higher order, or upper level, factor. Here, we are specifying what those loadings are going to be (0.60, 0.80, & 0.40).

```
sub.factor.loads <- matrix(c(.6,.8,.4), nrow = 3)
sub.factor.loads
```

```
      [,1]
[1,]  0.6
[2,]  0.8
[3,]  0.4
```

Next, we create a matrix object which contains the loadings between our 12 observed variables (often referred to as items) and our three subordinate factors. The following matrix can be thought of as the Pattern Coefficient matrix or simply Pattern matrix. Each column represents one of the three (subordinate) factors and each row represents one of the twelve observed variables.

```
obs.factor.loads <- matrix(c(.8,0,0,
                             .7,0,0,
                             .6,0,0,
```

```

                                .5,0,0,
                                0,.7,0,
                                0,.6,0,
                                0,.5,0,
                                0,.6,0,
                                0,0,.6,
                                0,0,.5,
                                0,0,.7,
                                0,0,.4), ncol = 3, byrow = TRUE)
obs.factor.loads

```

```

      [,1] [,2] [,3]
[1,] 0.8  0.0  0.0
[2,] 0.7  0.0  0.0
[3,] 0.6  0.0  0.0
[4,] 0.5  0.0  0.0
[5,] 0.0  0.7  0.0
[6,] 0.0  0.6  0.0
[7,] 0.0  0.5  0.0
[8,] 0.0  0.6  0.0
[9,] 0.0  0.0  0.6
[10,] 0.0  0.0  0.5
[11,] 0.0  0.0  0.7
[12,] 0.0  0.0  0.4

```

Next, we specify the number of observations and the mean for each of the observed variables. Here we are specifying  $n = 1000$  observations (i.e. rows of data) and we are using 10 as the mean of each of the observed variable scores, simply replicating ‘10’ twelve times for the twelve observed variables.

```

n.obs <- 1000
means <- rep(10, 12)

```

Next, we can use the ‘sim.hierarchical’ function from package ‘psych’ to simulate the observed data based on the model we specified above. Here, we give the data the name ‘data.df.1’. Notice below, we are requesting raw scores (raw = TRUE) and we are having the function return those observed raw scores (\$observed).

```

data.df.1 <- sim.hierarchical(gload = sub.factor.loads,
                             fload = obs.factor.loads,
                             n = n.obs,
                             raw = TRUE,
                             mu = means)$observed

```

Now we have our simulated, observed, data. If we run a summary of our data frame, we will see

the mean of each variable is approximately 10 (as we specified above).

```
summary(data.df.1)
```

```
      V1          V2          V3          V4
Min.   : 7.008   Min.   : 6.994   Min.   : 6.980   Min.   : 6.746
1st Qu.: 9.392   1st Qu.: 9.377   1st Qu.: 9.388   1st Qu.: 9.287
Median :10.086   Median :10.014   Median :10.014   Median :10.012
Mean   :10.076   Mean   :10.039   Mean   :10.026   Mean   : 9.993
3rd Qu.:10.749   3rd Qu.:10.708   3rd Qu.:10.658   3rd Qu.:10.640
Max.   :13.258   Max.   :13.469   Max.   :13.063   Max.   :13.024

      V5          V6          V7          V8
Min.   : 7.135   Min.   : 6.773   Min.   : 7.234   Min.   : 6.989
1st Qu.: 9.363   1st Qu.: 9.289   1st Qu.: 9.328   1st Qu.: 9.418
Median : 9.999   Median : 9.999   Median : 9.950   Median :10.025
Mean   :10.008   Mean   :10.001   Mean   : 9.990   Mean   :10.059
3rd Qu.:10.644   3rd Qu.:10.707   3rd Qu.:10.668   3rd Qu.:10.689
Max.   :13.476   Max.   :13.231   Max.   :12.765   Max.   :13.651

      V9          V10         V11         V12
Min.   : 7.438   Min.   : 7.011   Min.   : 6.493   Min.   : 6.884
1st Qu.: 9.334   1st Qu.: 9.319   1st Qu.: 9.389   1st Qu.: 9.314
Median : 9.989   Median :10.011   Median :10.090   Median : 9.976
Mean   :10.014   Mean   : 9.995   Mean   :10.073   Mean   :10.022
3rd Qu.:10.670   3rd Qu.:10.666   3rd Qu.:10.744   3rd Qu.:10.734
Max.   :13.135   Max.   :12.747   Max.   :13.230   Max.   :12.758
```

## Hierarchical Factor Analysis

Next, we can begin fitting our data to the model by applying a factor analysis at the subordinate level; extracting three factors with an oblique rotation from the observed (simulated) data. It is important to realize that an oblique rotation is called for — if the factors are not or were not related; there would be no reason to suspect a higher order factor or higher order factors. We start by obtaining a correlation matrix of the observed variables and assigning that matrix to an object ('FA.L1.mat' for factor analysis level 1 matrix). Then we can pass that matrix to our factor analysis function ('fa'), also supplying the number of factors (to extract), the factor method (i.e. extraction technique; 'minres'), the number of observations ( $n = 1000$ ), and the rotation strategy (oblimin).

```
cor.mat <- cor(data.df.1)
FA.L1 <- fa(r = cor.mat, nfactors = 3, fm = "minres", n.obs = 1000,
           rotate = "oblimin")
FA.L1
```

```
Factor Analysis using method = minres
Call: fa(r = cor.mat, nfactors = 3, n.obs = 1000, rotate = "oblimin",
```

```

fm = "minres")
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1   MR3   MR2   h2   u2
V1    0.76  0.05  0.01  0.62  0.38
V2    0.73 -0.05 -0.01  0.50  0.50
V3    0.60 -0.01  0.01  0.35  0.65
V4    0.46  0.02 -0.03  0.21  0.79
V5    0.04  0.65 -0.01  0.45  0.55
V6   -0.02  0.52  0.08  0.30  0.70
V7    0.00  0.55 -0.06  0.28  0.72
V8   -0.03  0.58  0.03  0.33  0.67
V9    0.04  0.01  0.62  0.40  0.60
V10  -0.01  0.01  0.49  0.25  0.75
V11  -0.02  0.00  0.68  0.45  0.55
V12   0.00 -0.02  0.43  0.18  0.82

```

```

              MR1  MR3  MR2
SS loadings      1.68 1.36 1.28
Proportion Var   0.14 0.11 0.11
Cumulative Var   0.14 0.25 0.36
Proportion Explained 0.39 0.31 0.30
Cumulative Proportion 0.39 0.70 1.00

```

```

With factor correlations of
      MR1  MR3  MR2
MR1  1.00  0.48  0.25
MR3  0.48  1.00  0.37
MR2  0.25  0.37  1.00

```

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 66 and the objective function value is 21.88  
The degrees of freedom for the model are 33 and the objective function value is 0.007

The root mean square of the residuals (RMSR) is 0.01  
The df corrected root mean square of the residuals is 0.02

The harmonic number of observations is 1000 with the empirical chi square = 21.88  
The total number of observations was 1000 with MLE Chi Square = 21.88

Tucker Lewis Index of factoring reliability = 1.011  
RMSEA index = 0 and the 90 % confidence intervals are NA 0.007  
BIC = -206.08  
Fit based upon off diagonal values = 1  
Measures of factor score adequacy

	MR1	MR3	MR2
Correlation of scores with factors	0.88	0.84	0.83
Multiple R square of scores with factors	0.78	0.71	0.68
Minimum correlation of possible factor scores	0.56	0.42	0.37

Notice in the above output, the 'fa' function used the Minimum Residual ("minres") factor method (extraction); this is the default. Given the simulated nature of the data, we would get the same substantive results had we applied the Maximum Likelihood ("ml") factor method (extraction). The second thing to take note of in the output is that the pattern matrix (i.e. loadings; e.g., V1 = 0.76, V2 = 0.73, V3 = 0.60, V4 = 0.46 for Subordinate factor 1) is very similar to what we specified when we simulated the data (e.g., V1 = 0.80, V2 = 0.70, V3 = 0.60, V4 = 0.50 for Subordinate factor 1). These loadings will not be exactly as specified due to the 'random' nature of the scores produced by the 'mvrnorm' function (i.e. produces multivariate random normal deviates - deviation scores); but the loadings will be very close. The next part of the output to pay particular attention to is the factor correlation matrix. Be advised, this correlation matrix is not (and will not be) the same as the correlation matrix resulting from saving factor scores and correlating them. The factor correlation matrix is commonly referred to as the 'Phi' matrix and can be extracted from the output by name using the \$ operator.

```
FA.L1.mat <- FA.L1$Phi
FA.L1.mat
```

```
      [,1]      [,2]      [,3]
[1,] 1.0000000 0.4817366 0.2451246
[2,] 0.4817366 1.0000000 0.3673734
[3,] 0.2451246 0.3673734 1.0000000
```

Next, we can apply the upper level factor analysis to the Phi matrix from above; specifying 1 factor (to be extracted) using the minimum residual (minres) factor method (i.e. extraction), specifying the number of observations ( $n = 1000$ ), and no rotation (rotate = "NULL").

```
FA.L2 <- fa(r = FA.L1.mat, nfactors = 1, fm = "minres", n.obs = 1000,
           rotate="NULL")
FA.L2
```

```
Factor Analysis using method = minres
Call: fa(r = FA.L1.mat, nfactors = 1, n.obs = 1000, rotate = "NULL",
        fm = "minres")
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1    h2    u2
1 0.57 0.32 0.68
2 0.85 0.72 0.28
3 0.43 0.19 0.81
```

```
MR1
```

```
SS loadings      1.23
Proportion Var  0.41
```

Test of the hypothesis that 1 factor is sufficient.

```
The degrees of freedom for the null model are 3 and the objective function value is 1.23
The degrees of freedom for the model are 0 and the objective function value is 0.41
```

```
The root mean square of the residuals (RMSR) is 0
The df corrected root mean square of the residuals is NA
```

```
The harmonic number of observations is 1000 with the empirical chi square value of 1.23
The total number of observations was 1000 with MLE Chi Square = 0 with 0 df
```

```
Tucker Lewis Index of factoring reliability = -Inf
Fit based upon off diagonal values = 1
Measures of factor score adequacy
```

```
Correlation of scores with factors      MR1      0.88
Multiple R square of scores with factors MR1      0.77
Minimum correlation of possible factor scores MR1 0.53
```

Notice in the output above, the loadings matrix, or pattern matrix, (0.57, 0.85, & 0.43) very closely resembles the coefficients we specified in creating the data (0.60, 0.80, & 0.40). These are the loadings between the subordinate factors and the upper level factor.

## Factor Scores

Next, we can demonstrate how the correlations among the factor scores will be different than the factor correlations (Phi matrix). Here, we again apply the 'fa' function but in order to get the factor scores, we must supply the raw data (data frame) matrix instead of the correlation matrix — of course, we also need to specify what type of scores we want. Here we specify standard regression scores, however; other methods are available (e.g., Bartlett's, Anderson's, etc.).

```
fa.ll <- fa(r = data.df.1, nfactors = 3, fm = "minres", n.obs = 1000,
           rotate = "oblimin", scores = "regression")
```

```
fa.ll
```

```
Factor Analysis using method = minres
Call: fa(r = data.df.1, nfactors = 3, n.obs = 1000, rotate = "oblimin",
        scores = "regression", fm = "minres")
```

```
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1  MR3  MR2  h2  u2
V1    0.76  0.05  0.01  0.62  0.38
```

V2	0.73	-0.05	-0.01	0.50	0.50
V3	0.60	-0.01	0.01	0.35	0.65
V4	0.46	0.02	-0.03	0.21	0.79
V5	0.04	0.65	-0.01	0.45	0.55
V6	-0.02	0.52	0.08	0.30	0.70
V7	0.00	0.55	-0.06	0.28	0.72
V8	-0.03	0.58	0.03	0.33	0.67
V9	0.04	0.01	0.62	0.40	0.60
V10	-0.01	0.01	0.49	0.25	0.75
V11	-0.02	0.00	0.68	0.45	0.55
V12	0.00	-0.02	0.43	0.18	0.82

	MR1	MR3	MR2
SS loadings	1.68	1.36	1.28
Proportion Var	0.14	0.11	0.11
Cumulative Var	0.14	0.25	0.36
Proportion Explained	0.39	0.31	0.30
Cumulative Proportion	0.39	0.70	1.00

With factor correlations of

	MR1	MR3	MR2
MR1	1.00	0.48	0.25
MR3	0.48	1.00	0.37
MR2	0.25	0.37	1.00

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 66 and the objective function value is 21.88  
The degrees of freedom for the model are 33 and the objective function value is 0.01

The root mean square of the residuals (RMSR) is 0.01  
The df corrected root mean square of the residuals is 0.02

The harmonic number of observations is 1000 with the empirical chi square value is 21.88  
The total number of observations was 1000 with MLE Chi Square = 21.88

Tucker Lewis Index of factoring reliability = 1.011  
RMSEA index = 0 and the 90 % confidence intervals are NA 0.007  
BIC = -206.08  
Fit based upon off diagonal values = 1  
Measures of factor score adequacy

	MR1	MR3	MR2
Correlation of scores with factors	0.88	0.84	0.83
Multiple R square of scores with factors	0.78	0.71	0.68
Minimum correlation of possible factor scores	0.56	0.42	0.37

Next, we extract the factor scores from the (lower case) lower level factor analysis output (fa.l1) using the \$ operator and assign them to the 'fa.l1.sc' object. We can also correlate those scores and notice how different those correlations are from the factor correlations matrix (Phi matrix).

```
fa.l1.sc <- data.frame(fa.l1$scores)
cor(fa.l1.sc)
```

```
      X1      X2      X3
X1 1.0000000 0.6004052 0.3217752
X2 0.6004052 1.0000000 0.4805090
X3 0.3217752 0.4805090 1.0000000
```

As a result of these differences (between the factor score correlations and the factor correlations), the upper level factor loadings are going to be more biased (i.e. quite different than what we specified when generating the data and when we did the upper level factor analysis with the correct [Phi] matrix of association). To verify this, we apply the factor analysis to the factor scores themselves.

```
fa.l2 <- fa(r = fa.l1.sc, nfactors = 1, fm = "minres", n.obs = 1000,
           rotate = "oblimin", scores = "regression")
```

```
fa.l2
```

```
Factor Analysis using method = minres
Call: fa(r = fa.l1.sc, nfactors = 1, n.obs = 1000, rotate = "oblimin",
        scores = "regression", fm = "minres")
```

```
Standardized loadings (pattern matrix) based upon correlation matrix
```

```
      MR1    h2    u2
X1 0.63 0.40 0.60
X2 0.95 0.90 0.10
X3 0.51 0.26 0.74
```

```
      MR1
SS loadings    1.56
Proportion Var 0.52
```

```
Test of the hypothesis that 1 factor is sufficient.
```

```
The degrees of freedom for the null model are 3 and the objective function value is 1000
The degrees of freedom for the model are 0 and the objective function value is 0
```

```
The root mean square of the residuals (RMSR) is 0
The df corrected root mean square of the residuals is NA
```

```
The harmonic number of observations is 1000 with the empirical chi square value of 1000
The total number of observations was 1000 with MLE Chi Square = 0 with df = 0
```

```

Tucker Lewis Index of factoring reliability = -Inf
Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                         MR1
Correlation of scores with factors      0.95
Multiple R square of scores with factors 0.91
Minimum correlation of possible factor scores 0.81

```

The output above shows the higher order factor loadings (0.62, 0.95, & 0.51) are substantively different (i.e. more biased) than what we would expect (0.60, 0.80, & 0.40); and what we found farther above using the factor correlation, or Phi, matrix (0.57, 0.85, & 0.43).

### Upper Level Factor Scores

Unfortunately, the ‘fa’ function cannot return the factor scores when supplied a correlation matrix as was done originally (i.e. FA.L2). So, in order to calculate the upper level factor scores and save them for use in further analyses (e.g., Structural Equation Modeling), we must use the factor correlation matrix (Phi matrix) to generate some lower level (3 factor) raw data on which we can then apply the upper level (1 factor) analysis - which will allow us to save the higher level factor scores for future use. Below, we demonstrate this by first using the ‘mvrnorm’ function; supplying it with the number of cases/rows the data should have (equal to the number of rows of the original data [nrow(data.df.1)]), the mean we want each of the three factors to have [three means of zero: rep(0,3)], and the all-important matrix to which the data should conform [FA.L1.mat (which is the Phi matrix from above)]. The ‘empirical = TRUE’ argument simply requests that the data conform exactly to the matrix we supply. Once the data has been created, we can do a summary of the data frame and check the correlations to see if they indeed match the Phi matrix from above; and they do.

```

L2.df <- data.frame(mvrnorm(n = nrow(data.df.1), rep(0,3),
                           FA.L1.mat, empirical = TRUE))
summary(L2.df)

```

```

      X1                X2                X3
Min.   :-4.375742   Min.   :-3.54389   Min.   :-2.83910
1st Qu.:-0.688568   1st Qu.:-0.68134   1st Qu.:-0.74282
Median :-0.002373   Median  : 0.01529   Median :-0.01382
Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.00000
3rd Qu.: 0.691442   3rd Qu.: 0.69003   3rd Qu.: 0.68826
Max.   : 3.776186   Max.   : 2.76209   Max.   : 3.65736

```

```
cor(L2.df)
```

```

      X1      X2      X3

```

```
X1 1.0000000 0.4817366 0.2451246
X2 0.4817366 1.0000000 0.3673734
X3 0.2451246 0.3673734 1.0000000
```

Next, we can then use the newly created 'raw' data (L2.df [data frame]) to apply the upper level (1 factor) model. The key point here is; we can save the upper level factor scores because we are supplying 'raw' data we just created - and that raw data exactly conforms to the Phi matrix.

```
FA.L2.2 <- fa(r = L2.df, nfactors = 1, fm = "minres", n.obs = 1000,
             rotate="NULL", scores = "regression")
```

```
FA.L2.2
```

```
Factor Analysis using method = minres
Call: fa(r = L2.df, nfactors = 1, n.obs = 1000, rotate = "NULL", scores =
       fm = "minres")
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1   h2   u2
X1 0.57 0.32 0.68
X2 0.85 0.72 0.28
X3 0.43 0.19 0.81
```

```

              MR1
SS loadings   1.23
Proportion Var 0.41
```

Test of the hypothesis that 1 factor is sufficient.

The degrees of freedom for the null model are 3 and the objective function value is 1.23  
The degrees of freedom for the model are 0 and the objective function value is 0

The root mean square of the residuals (RMSR) is 0  
The df corrected root mean square of the residuals is NA

The harmonic number of observations is 1000 with the empirical chi square value of 1.23  
The total number of observations was 1000 with MLE Chi Square = 0 with df = 0

```
Tucker Lewis Index of factoring reliability = -Inf
Fit based upon off diagonal values = 1
Measures of factor score adequacy
```

```

              MR1
Correlation of scores with factors      0.88
Multiple R square of scores with factors 0.77
Minimum correlation of possible factor scores 0.53
```

Note in the output above the loadings between the lower level (3) factors and the upper level (1)

factor are identical to what was produced when the factor analysis was done on the (lower level 3) factor correlation, or Phi, matrix. Now we can extract the factor scores from the output object using the \$ operator and the 'scores' name. Below we assign those factor scores to the object 'L2.scores' and create a summary of those scores. Notice, the factor scores are centered on zero (i.e. standardized) because the latent factor has an unknown metric.

```
L2.scores <- FA.L2.2$scores
summary(L2.scores)
```

```
MR1
Min.   :-2.95558
1st Qu.:-0.57108
Median :-0.01473
Mean   : 0.00000
3rd Qu.: 0.57839
Max.   : 2.56140
```

Of course, if our goal from the start was to get these (upper level, 1 factor) factor scores, then we would have simply created the 'raw' data after the first factor analysis and supplied it to the second factor analysis while requesting the scores. The script below accomplishes this goal without all the comments, explanation, and output.

```
fa1 <- fa(r = data.df.1, nfactors = 3, fm = "minres", n.obs = 1000
         rotate = "oblimin")
fa1.df <- data.frame(mvrnorm(n = nrow(data.df.1), rep(0,3),
                           fa1$Phi, empirical = TRUE))
fa2 <- fa(r = fa1.df, nfactors = 1, fm = "minres", n.obs = 1000,
         rotate = "none", scores = "regression")
fa2.sc <- fa2$scores
```

## Conclusions

It is important to note that the examples above involve a relatively simple factor structure, however; the general procedures would be the same for more complex structures (i.e. multiple upper level factors, or multiple upper levels - levels 1, 2, 3, 4, etc.). The importance of having knowledge of these procedures in one's repertoire should be apparent, given the frequency of hierarchical measurement structures in the social sciences. Next month, we will show how to use the upper level factor scores in a larger (Structural Equation) model by incorporating them into a structural model with other latent variables.

Until next time, happy computing...

## References & Resources

- Bernaards, C. A., & Jennrich, R. I. 2013. Package ‘GPARotation’: Gradient Projection Algorithm Rotation for Factor Analysis. *Educational and Psychological Measurement*, 65, 676 – 696.
- Gorsuch, R. L. (1983). *Factor analysis* (2nd. ed.) Hillsdale, NJ: Lawrence Earlbaum Associates.
- McDonald, R. P. (1985). *Factor analysis and related methods*. Hillsdale, NJ: Lawrence Earlbaum Associates.
- McDonald, R. P. (1999). *Test theory: A unified treatment*. Mahwah, NJ: Lawrence Earlbaum Associates.
- Mulaik, S. A., & Quartelli, D. A. (1997). First order or higher order general factor? *Structural Equation Modeling*, 4(3), 193 - 211.
- Pyryt, M. C. (2000). Finding “g”: Easy viewing through higher order factor analysis. *Gifted Child Quarterly*, 44(3), 190 - 193.
- Revelle, W. (2013). Package ‘psych’: Procedures for Personality and Psychological Research.
- Revelle, W. (in prep.). *Psychometric theory with applications in R*. Available (draft) at:  
<http://personality-project.org/r/r.guide.html>  
and  
<http://personality-project.org/r/book/>  
And specifically, here:  
<http://personality-project.org/r/book/Chapter6.pdf>
- Ripley, B., Venables, B., Hornik, K., Gebhardt, A., & Firth, D. (2013). Package ‘MASS’: Functions and datasets to support Venables and Ripley, ‘*Modern Applied Statistics with S*’ (4th ed.), 2002.
- Schmid, J., & Leiman, J. M. (1957). The development of hierarchical factor solutions. *Psychometrika*, 22, 83 - 90.
- Special Issue. (1990). The Journal of the Society of Multivariate Experimental Psychology. *Multivariate Behavioral Research*, 25(1), 1 - 136. All 13 Articles available at:  
<http://www.tandfonline.com/toc/hmbr20/25/1>
- Wherry, R. J. (1959). Hierarchical factor solutions without rotation. *Psychometrika*, 24, 45 - 51.
- Wolff, H., & Preising, K. (2005). Exploring item and higher order factor structure with the Schmid-Leiman solution: Syntax codes for SPSS and SAS. *Behavior Research Methods*, 37(1), 48 - 58.

Yung, Y., Thissen, D., & McLeod, L. D. (1999). On the relationship between the higher-order factor model and the hierarchical factor model. *Psychometrika*, *64*(2), 113 - 128.

This article was last updated on April 10, 2013.

This document was created using  $\text{\LaTeX}$