Improving Multiple-Group confirmatory factor analysis in R – A tutorial in measurement invariance with continuous and ordinal indicators

Gerrit Hirschfeld
Ruth von Brachel

Follow this and additional works at: https://scholarworks.umass.edu/pare

Recommended Citation
DOI: https://doi.org/10.7275/qazy-2946
Available at: https://scholarworks.umass.edu/pare/vol19/iss1/7

This Article is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Practical Assessment, Research, and Evaluation by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.
Multiple-Group confirmatory factor analysis in R – A tutorial in measurement invariance with continuous and ordinal indicators

Gerrit Hirschfeld, Childrens’ Hospital Datteln, Germany,
Ruth von Brachel, Ruhr-University Bochum, Germany

Multiple-group confirmatory factor analysis (MG-CFA) is among the most productive extensions of structural equation modeling. Many researchers conducting cross-cultural or longitudinal studies are interested in testing for measurement and structural invariance. The aim of the present paper is to provide a tutorial in MG-CFA using the freely available R-packages lavaan, semTools, and semPlot. The combination of these packages enable a highly efficient analysis of the measurement models both for normally distributed as well as ordinal data. Data from two freely available datasets – the first with continuous the second with ordered indicators - will be used to provide a walk-through the individual steps.

Many researchers in psychology and social science are faced with the problem to compare latent constructs (i.e. mathematic ability, extraversion) that are not directly observable between different groups (languages, ethnic-groups), or points in time. Usually these latent constructs are measured by questionnaires, comprised of different scales that reflect different underlying latent variables. Typically differences between groups with regard to these underlying constructs are tested via scale means. Any comparison of means presuppose that the measures function similar in these different groups, i.e. that the response to individual items can be explained by the same latent factors (Byrne, Shavelson, & Muthén, 1989; Cheung & Rensvold, 1999; Vandenberg & Lance, 2000). Multiple-Group Confirmatory Factor Analysis (MG-CFA) has become the de-facto standard to investigate the degree to which measures are invariant across groups (Chen, 2008). Practical applications in educational psychology entail the cross-cultural validation of tests testing for equation of international test (Wu, Li, & Zumbo, 2007) and assessing the invariance of test results across different subgroups, e.g. the validity of somatic complaints in White and African American samples (Kline, 2013). These techniques are also widely-used in medicine where measurement invariance is seen as an important precursor for interpreting patient reported outcomes (Gregorich, 2006).

Although there are several introductions to MG-CFA to test for invariance that are based on commercial programs, e.g. AMOS (Byrne, 2004), several recent additions to the open source software R (R Core Team, 2012) enable researchers to perform such analysis with unprecedented efficiency. In this paper we will describe how the three packages lavaan (Rosseel, 2012), semPlot (Epskamp, 2013) and semTools can be combined to conduct MG-CFA analysis. Before providing a walk-through the analysis a short conceptual introduction is given.

A conceptual introduction to measurement invariance

A scale is said to have measurement invariance (also known as measurement equivalence) across groups if subjects with identical levels of the latent construct have the same expected raw-score on the measure (Drasgow & Kanfer, 1985). As such, the level of measurement invariance a scale exhibits has very important implication for the interpretation of differences. If measurement invariance has been established for a measure, observed mean differences can be attributed to differences in underlying constructs between the groups. If however, one cannot
assume a stable relation between underlying construct and scale score, observed mean differences may be either due to differences in underlying constructs, or due to the different relations between latent constructs and scores. There are currently two approaches to test for invariance; structural equation modeling, and item response theory (Raju, Laffitte, & Byrne, 2002; Reise, Widaman, & Pugh, 1993).

Structural equation modeling (SEM) lends itself naturally to investigate the invariance of the relations between underlying constructs (latent variables) and observed responses (manifest variables), since these relations are explicitly modeled. For example figure 1 is a graphical representation of a measurement model for the classic dataset by Holzinger and Swineford (1939) comprising scores of 300 school children on nine different tests. In this measurement model the performance on the nine different tests is explained by three interrelated latent constructs; speed, textual, and visual. Following usual conventions observed variables are represented by rectangles and latent variables are represented by ovals. The paths indicate which item loads on which factor. The fact that loadings are represented by directed arrows highlights the fact that the measurement model presupposes that the latent variables affect the individual items. In regression terms fitting this model to the data entails estimating six parameters; (1) a regression coefficient (e.g. the loading of test “x1” on factor visual “visual”), (2) a regression intercept, (3) a regression residual variance, (4) the means of the factors, (5) the variances of the underlying factors, and (6) the covariances of the underlying factors.

Within the SEM framework different levels of measurement invariance may be defined; configural, weak, strong, and strict invariance that correspond to the above-mentioned regression parameters. Configural invariance implies that the number of latent variables and the pattern of loadings of latent variables on indicators are similar across the groups. In the above example this implies that in all groups the first three tests “x1”, “x2”, and “x3” are influenced by the same latent variable “visual ability”. Weak invariance (also known as metric invariance) implies that the magnitude of the loadings is similar across the groups. This form of measurement invariance is required in order to meaningfully compare the relationships between latent variables across different groups. Strong invariance (also known as scalar invariance) implies that not only the item loadings but also the item intercepts are similar across the groups. This form of measurement invariance implies that there are no systematic response biases and is required in order to meaningfully compare the means of latent variables across different groups (Chen, 2008). Last, some authors require strict invariance before means can be compared (Wu et al., 2007). Strict invariance implies that in addition to loadings and intercepts, the residual variances are similar across groups. After having established measurement invariance, researchers may go on to test substantial hypotheses about the means and interrelations between latent constructs. For example after having established that the measurement model is invariant across groups one might want to test whether the two groups differ in mean visual ability or whether these latent variables are related to academic achievement as measured by grades in different subjects.

Testing for measurement invariance

Testing for measurement invariance consists of a series of model comparisons that define more and more stringent equality constraints (Byrne, 2009; Cheung & Rensvold, 1999; Raju et al., 2002; Vandenberg & Lance, 2000). First, a baseline model is fit in which the loading pattern is similar in all groups but the magnitude of all parameters — loadings, intercepts, variances, etc. - may vary. Configural invariance exists if this baseline model has a good fit and the same loadings are significant in all groups.
Second, a weak-invariance model in which the factor loadings are constrained to be equal is fit to the data and the fit of this model is compared to the baseline model. Weak invariance exists if the fit of the metric invariance model is not substantially worse than the fit of the baseline model. As described below there exist several statistical alternatives to decide whether the fit is substantially worse. Third, a strong-invariance model in which factor loadings and item intercepts are constrained to be equal is fit to the data and compared against the weak measurement invariance model. Again strong invariance exists if the fit of the scalar invariance model is not substantially worse than the fit of the weak invariance model. Fourth, a strict invariance model in which factor loadings, intercepts, and residual variances are constrained to be equal is fit to the data and compared to the strong measurement invariance model.

A special case pertains to the testing of multiple-group models with ordinal indicators (Millsap & Yun-Tein, 2004; Muthén & Asparouhov, 2002; Temme, 2006). Even though there is some debate about the exact number of categories a likert-scale needs to have in order to be treated as continuous, it is clear that likert-scales with few (probably four) categories are best handled using alternative estimation methods that take into account the ordinal nature of the data (Rhemtulla, Brosseau-Liard, & Savalei, 2012). The approach to dealing with ordered indicators most often employed is modeling thresholds for each indicator that describe at which level of the latent variable a specific category is chosen and using the weighted least squares means and variance adjusted (WLSMV) estimator to estimate parameters. Within the framework of MG-CFA these thresholds are roughly equivalent to the item loadings. That makes testing for weak and strong measurement invariance relatively easy. However, testing for strict invariance, i.e. testing the equality of residual variances, is only possible when theta-parameterization is used to identify model parameters (Muthén & Asparouhov, 2002). Since lavaan currently uses delta-parameterization the residuals are not estimated and one cannot the equality of these parameters across groups.

Importantly, the decision whether or not a measurement model exhibits measurement invariance is not an all-or-none decision. Partial measurement invariance describes scenarios in which only some indicators exhibit a certain level of measurement invariance while the others do not. For example three out of four indicators may exhibit strong invariance while the fourths only exhibits weak invariance (Byrne et al., 1989). This indicator is identified by constraining only those parameters (loadings, intercepts) pertaining to one specific indicator (Cheung & Rensvold, 1999). Whenever indicators show evidence of invariance researchers may drop these indicators from the model, use partial measurement invariance, or omit any interpretation of the scales across the groups. Some authors have argued that only two indicators are needed to be invariant to make meaningful comparisons between groups (Steenkamp & Baumgartner, 1998).

### Decision rules for invariance tests

An open issue pertains the use of different decision rules for invariance (Wu et al., 2007). The problem is that imposing equality constraints will always result in a decrease in fit because less degrees of freedom are available. Consider testing for weak invariance by comparing the baseline model with the weak-invariance model. In the baseline model the loadings of the items on the factors are allowed to be different between the group. In the weak-invariance model these loadings are constrained to be equal. Since the baseline-model has more free parameters than the weak-invariance model the baseline-model’s overall model fit will be better. This raises the question whether a specific decrease in fit observed during the model comparisons is substantial or not. Initial studies used chi-square tests to decide whether or not the increase in fit is substantial (Byrne et al., 1989). Following studies have however identified several problems with this approach and proposed using a difference in fit indices to define invariance (Cheung & Rensvold, 2002). Other authors have adopted a hybrid approach arguing that chi-square should be used to determine invariance at the measurement level (i.e. configural, weak, strong, and strict invariance), and fit-indices should be used at the structural level (Little, 1997). At present the inspection of changes in fit-indices, specifically the difference in comparative fit index (CFI) ($\Delta$CFI), seems the most widely used and empirically best supported criterion to define invariance (Chen, 2007; Cheung & Rensvold, 2002). Most often a cutpoint of $\Delta$CFI < .01 is chosen to decide whether a more constrained model, e.g. the weak-invariance model, shows a substantial decrease in model fit compared to a less constrained model, e.g. the baseline model. Some authors have however shown that the optimal cutpoints for differences in chi-square or CFI...
strongly depend on model complexity and have provided tables for cutpoints that result in higher power and sensitivity to detect invariance than global decision rules (Meade, Johnson, & Braddy, 2008). Unfortunately previous systematic simulation studies into the performance of cut-off values have used maximum likelihood estimation (Chen, 2007; Cheung & Rensvold, 2002; Meade et al., 2008). As a result it is unknown whether or not the standard cutoff points for differences in CFI are also applicable to models estimated with WLSMV. As a result very few studies into measurement invariance have (Chungkham, Ingre, Karasek, Westerlund, & Theorell, 2013) taken into account ordinal indicators and instead have used ML estimation to fit the data. A recent simulation study (Koh & Zumbo, 2008) has shown that this practice does not lead to inflated type-I error rates, i.e. claiming non-invariance when models are in fact invariant. We compare the outcome of the analysis in the second example presented below.

**Example I: Continuous indicators**

Our first example will analyze a dataset that only included continuous indicators. The packages lavaan, semTools and semPlot contain all functions needed to efficiently run MG-CFA analysis in R. Running a MG-CFA analysis comprises six steps; (1) Install/load packages; (2) Loading data; (3) specifying a baseline model; (4) defining equality constraints; (5) comparing the models; (6) visualizing results. Table 1 gives an overview of the functions used and their most important parameters. The first three steps have already been described in more detail in a previous article in this journal (Beaujean, 2013) so that they are only summarized here.

**Install/load Packages**

Before the functions can be used they have to be installed once and loaded at the beginning of the script. In the following lines beginning with “>” denote code that has to be entered by the user and the output that is generated by R is printed in bold, “[…]” is used to denote that the output was truncated. The following commands install and load the packages:

```
> install.packages(c('lavaan', 'semTools', 'semPlot'))
> library(lavaan)
> library(semPlot)
> library(semTools)
```

**Loading data**

R has many functions to load data in various formats, ranging from simple tabular data such as comma-separated files to more specialized data files such as SPSS or SAS-data files (Beaujean, 2013). Also some packages already include datasets. In our first example we will use the Holzinger-Swineford data that is part of the lavaan package. As such it can be loaded into memory using the function data(), after the package is installed and the package loaded, as described in the previous section.

```
> data(HolzingerSwineford1939)
> str(HolzingerSwineford1939)
'data.frame': 301 obs. of 15 variables:
$ id : int 1 2 3 4 5 6 7 8 9 11 ...
$ sex : int 1 2 2 1 2 2 1 2 2 2 ...
$ ageyr : int 13 13 13 13 12 14 12 12 13 12 ...
$ agemo : int 1 7 1 2 2 1 1 2 0 5 ...
$ school: Factor w/ 2 levels "Grant-White",..: 2 2 2 2 2 2 2 2 2 2 ...
$ grade : int 7 7 7 7 7 7 7 7 7 7 ...
$ x1   : num 3.33 5.33 4.5 5.33 4.83 ...
$ x2   : num 7.75 5.25 5.25 7.75 4.75 ...
$ x3   : num 6.25 5.75 5.25 ...
$ x4   : num 3.33 3.33 4.5 5.33 4.83 ...
$ x5   : num 7.75 5.25 5.25 7.75 4.75 5 6
$ x6   : num 1.286 1.286 0.429 2.429 2.571 ...
$ x7   : num 3.39 3.78 3.26 3.7 ...
$ x8   : num 5.75 6.25 3.9 6.3 6.65 ...
$ x9   : num 6.36 7.92 4.42 4.86 5.92 ...
```

Table 1. Important functions and parameters

<table>
<thead>
<tr>
<th>Function</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cfa()</code></td>
<td>Fits a model to data. The parameters <code>group.equal</code> and <code>group.partial</code> allow defining and relaxing constraints.</td>
</tr>
<tr>
<td><code>moreFitIndices()</code></td>
<td>Gives several additional fit indices.</td>
</tr>
<tr>
<td><code>semPaths()</code></td>
<td>Plots structural models and estimates.</td>
</tr>
<tr>
<td><code>Measurement</code></td>
<td>Performs a series of model comparisons for which chi-square and ΔCFI are reported. Allows relaxing constraints via the parameter <code>group.partial</code>.</td>
</tr>
<tr>
<td><code>Invariance()</code></td>
<td>Visualizes data.</td>
</tr>
<tr>
<td><code>ggplot()</code></td>
<td>Gives only part of the model summary so that these can be stored.</td>
</tr>
<tr>
<td><code>inspect()</code></td>
<td>Performs a permutation test to estimate the distribution of ΔCFI for random groups.</td>
</tr>
</tbody>
</table>
The output of the function \texttt{str()} describes the variables in the dataset “HolzingerSwineford1939”. Each line represents one variable, in which the name (e.g. sex), format (int = integer, or num = numeric, or Factor) and first few datapoints are given.

**Specifying and inspecting the baseline model**

We will fit a simple three-factor model to the data. This entails specifying the model using lavaan’s model-synt x; fitting the model to the data using the function \texttt{cfa()}, and inspecting the model with the functions \texttt{summary()}, \texttt{moreFitIndices()} and \texttt{semPaths()}.

```r
> model <- ' visual =~ x1 + x2 + x3; 
    textual =~ x4 + x5 + x6; speed =~ x7 + x8 + x9 '

Lavaan’s model-synt x was designed to enable researchers to quickly set up models with useful default parameters in mind. As such covariances between all latent variables (“visual” and “textual”) are added automatically. All defaults can be overridden as described by Rosseel (2012).

```r
> fit <- cfa(model, data=HolzingerSwineford1939)
> summary(fit, standardized = TRUE, fit.measures = TRUE)
```

```
lavaan (0.5-16) converged normally after 35 iterations
Number of observations                  301
Estimator                                ML
Minimum Function Test Statistic      85.306
Degrees of freedom                       24
P-value (Chi-square)                  0.000
Model test baseline model:
Minimum Function Test Statistic     918.852
Degrees of freedom                       36
P-value                                  0.000
User model versus baseline model:
Comparative Fit Index (CFI)           0.931
Tucker-Lewis Index (TLI)              0.896
Loglikelihood and Information Criteria:
Loglikelihood user model (H0)     -3737.745
Loglikelihood unrestricted model (H1)     -3695.092
Number of free parameters            21
Akaike (AIC)                        7517.490
Bayesian (BIC)                      7595.339
Sample-size adjusted Bayesian (BIC)  7528.739
Root Mean Square Error of Approximation:
RMSEA                                  0.092
90 Percent Confidence Interval 0.071  0.114
P-value RMSEA <= 0.05                 0.001
```

The output of the function \texttt{summary()} already provides the user with data pertaining to the model fit, e.g. RMSEA. Two additional functions provide more fit indices and a graphical representation of the model.

```r
> moreFitIndices(fit)
> semPaths(fit, "std")
```

```
gammaHat       adjGammaHat   baseline.rmsea
0.9565611      0.9185521     0.2854364
aic.smallN     bic.priorN    hqc
7476.5731866   7544.0149775  7517.2909607
sic
3794.0917641
```

```
Figure 2. Measurement model for the Holzinger and Swinford Data including parameter estimates

Overall, inspection of the output shows that this model only has a very weak fit to the data ($\chi^2 = 85.306; DF = 24; CFI = .93; gamma hat = .96; RMSEA = .092; SRMR = 0.065$). Normally, researchers would have to improve the model fit, as this is also important to assess configural invariance. Since our main aim is to describe the analysis, we continue to work with this model and focus on testing hypotheses about weak, strong, and strict invariance.

**Running multiple-group tests**

Multiple-group CFAs are implemented in lavaan by calling the function \texttt{cfa()} with additional parameters (group, group.equal, and group.partial) that specify equality constraints between the different groups. One can specify these different models manually and compare them using the function \texttt{anova()}.

```r
> config <- cfa(model, data=HolzingerSwineford1939, group="school")
```

```
Standardized Root Mean Square Residual:
   0.065
Parameter estimates:
[...]
```

```
```
The model comparisons to test for weak, strong and strict invariance are found under the headings [Model 1 versus model 2], [Model 2 versus model 3], [Model 3 versus model 4], respectively. The first three entries give the difference in chi-square, the corresponding degrees of freedom and significance test. The last entry gives the difference in CFI between the two models. The output of this function prints all data needed including CFI needed to construct a typical table (tab. 2).

### Table 2. Series of model comparisons

<table>
<thead>
<tr>
<th>Model</th>
<th>$\chi^2$ ($Df$)</th>
<th>$p$ ($\Delta p$)</th>
<th>CFI ($\Delta CFI$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 Configural</td>
<td>115.851</td>
<td>48</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>M2 Weak invariance (loadings)</td>
<td>(8.192)</td>
<td>(6)</td>
<td>(0.224)</td>
</tr>
<tr>
<td>M3 Strong invariance (loadings, and intercepts)</td>
<td>(40.059)</td>
<td>(6)</td>
<td>(&lt;.001)</td>
</tr>
<tr>
<td>M3b. Partial strong invariance (except item #x3)</td>
<td>(32.322)</td>
<td>(5)</td>
<td>(&lt;.001)</td>
</tr>
<tr>
<td>M3c. Partial strong invariance (except items #x3 and #7)</td>
<td>(5.379)</td>
<td>(4)</td>
<td>(.251)</td>
</tr>
<tr>
<td>M4 Partial strict (M3c plus residual variances)</td>
<td>(11.585)</td>
<td>(7)</td>
<td>(0.115)</td>
</tr>
</tbody>
</table>

Note. According to Cheung and Rensvold (2002) $\Delta$CFI < 0.01 implies that the invariance assumption still holds.

The series of model comparisons indicate that the factor loadings can be assumed to be equal, since the chi-square test is not significant and $\Delta$CFI is smaller than the proposed cutpoint of .01 (Cheung & Rensvold, 2002). When constraining the intercepts to be equal across groups a significant increase in chi-square and a large increase in CFI highlights that the strong invariance assumption cannot be met. In order to test for partial invariance we will inspect the modification indices for individual parameters in the
more constrained model – here the strong-invariance model. Specifically, we will first use the function 
modificationIndices() to extract the modification indices and inspect modification indices that pertain to 
intercepts.

```r
> mod_strong<-modificationIndices(strong)
> mod_strong[mod_strong$op == "~1",]
  lhs op rhs group mi  epc sepc.lv sepc.all sepc.nox
1  x1 ~1  1 4.485 0.133 -0.133 -0.114 -0.114
2  x2 ~1  1 6.634 -0.165 -0.165 -0.132 -0.132
3  x3 ~1  1 17.717 0.248 0.248 0.206 0.206
4  x4 ~1  1 1.816 0.058 0.058 0.050 0.050
5  x5 ~1  1 1.316 -0.054 -0.054 -0.042 -0.042
6  x6 ~1  1 0.028 -0.007 -0.007 -0.007 -0.007
7  x7 ~1  1 13.681 0.205 0.205 0.186 0.186
8  x8 ~1  1 3.864 -0.099 -0.099 -0.102 -0.102
9  x9 ~1  1 1.322 -0.058 -0.058 -0.059 -0.059
10 visual ~1  1 0.000 0.000 0.000 0.000 0.000
11 textual ~1  1 0.000 0.000 0.000 0.000 0.000
12 speed ~1  1 0.000 0.000 0.000 0.000 0.000
13 x1 ~1  2 4.485 0.133 0.133 0.114 0.114
14 x2 ~1  2 6.634 0.165 0.165 0.132 0.132
15 x3 ~1  2 17.717 0.248 0.248 0.206 0.206
16 x4 ~1  2 1.816 0.058 0.058 0.050 0.050
17 x5 ~1  2 1.316 -0.054 -0.054 -0.042 -0.042
18 x6 ~1  2 0.028 -0.007 -0.007 -0.007 -0.007
19 x7 ~1  2 13.681 -0.205 -0.205 -0.186 -0.186
20 x8 ~1  2 3.864 -0.099 -0.099 -0.102 -0.102
21 x9 ~1  2 1.322 -0.058 -0.058 -0.059 -0.059
22 visual ~1  2 0.000 0.000 0.000 0.000 0.000
23 textual ~1  2 0.000 0.000 0.000 0.000 0.000
24 speed ~1  2 0.000 0.000 0.000 0.000 0.000
```

This list shows that the modification indices are largest for the intercept belonging to the item x3. So this will be the first item for which we relax the equality constraint, i.e. we allow the intercept for this item to differ between groups. For this the function measurementInvariance() is used together with the parameter group.partial to specify the intercepts for which we relax the constraints.

```r
> measurementInvariance(model, 
data=HolzingerSwineford1939, 
group="school", group.partial = 
c("x3 ~1"))
```

The line corresponding to the partial strong invariance test now shows a smaller difference in chi-square and the correct degrees of freedom (5, was 6). However, both chi-square significance test and ΔCFI still indicate a lack of strong invariance. Revisiting the modification indices (see above) indicated item x7 as a second potential source for invariance. So next we allow the intercept corresponding to this item to differ between the groups.

```r
> measurementInvariance(model, 
data=HolzingerSwineford1939, 
group="school", group.partial = 
c("x3 ~1", "x7 ~1"))
```

The line corresponding to the partial strong invariance test now shows a non-significant chi-square test and a ΔCFI that is below the cutpoint of .01. Furthermore, even though the test for strict invariance yields a significant chi-square test, the ΔCFI is not larger than the cutpoint indicating that with the exception of items x3 and x7 the scale exhibits partial strict measurement invariance. Based on these results researchers may thus interpret differences between schools in the means between those two groups as reflecting real differences in the underlying latent trait (i.e. intelligence) rather than the measure.

**Example II: Ordinal indicators**

Our second example will use data from 3376 participants who took part in an online survey that administered the sexual compulsivity scale (Kalichman & Rompa, 1995). The data is made available on the website http://personality-testing.info/_rawdata/. This scale consists of ten items consisting of descriptions about sexual behaviour, e.g “I think about sex more than I would like to”. Participants respond to each item on a four-category likert scale ranging from “not at all like me” to “very much like me”. Even though this issue could be investigated using ML estimation (Koh & Zumbo, 2008), we will also use the “correct” way by declaring these variables as ordinal.

**Load Packages**

As before, we need to load the previously installed packages lavaan, semPlot, and semTools before we can assess the functions.

```r
> library(lavaan)
> library(semPlot)
> library(semTools)
```

**Loading data**

Since we want to use data that is stored as a zip-file on a website, we need to first download this file and unzip it before we can load it into R. You may download and unzip the file (“http://personality-testing.info/_rawdata/SCS.zip”) manually or use the function download.file() and unzip() as described below. The file is loaded with read.csv(). Since we want to compare men to women, we use the subset of the data in which participants responded either male or female (subset(tmp, gender == "1" | gender == "2").
Practical Assessment, Research, & Evaluation, Vol 19, No 7
Hirschfeld & von Brachel, Multiple-group confirmatory factor analysis

Specifying and inspecting the baseline model

Next we will fit a one-factor model taking into account the ordered nature of the indicators. This is done by declaring the indicators as ordinal using the parameter ordered. By declaring the items q1 to q10 as ordered lavaan automatically switches to a different estimation method. The output of the function summary() provides both parameter estimates and indices for overall model fit.

> scs_model_fit<- cfa(scs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), data=scs)
> summary(scs_model_fit, standardized = TRUE, fit.measures = TRUE)

lavaan (0.5-16) converged normally after 23 iterations

Number of observations 3348
Estimator             DWLS Robust
Minimum Function Test Statistic 1083.730 2356.933
Degrees of freedom 35 35
P-value (Chi-square 0.000 0.000
Scaling correction factor 0.461
Shift parameter for simple second-order correction (Mplus variant) 4.613

Model test baseline model:

Minimum Function Test Statistic 95130.457 35638.844
Degrees of freedom 45 45
P-value 0.000 0.000

User model versus baseline model:

Comparative Fit Index (CFI) 0.989 0.935
Tucker-Lewis Index (TLI) 0.986 0.916

Root Mean Square Error of Approximation:

RMSEA 0.095 0.141
90 Percent Confidence Interval 0.090-0.100 0.136-0.146
P-value RMSEA <= 0.05 0.000 0.000

Weighted Root Mean Square Residual:

WRMR 3.571 3.571

Parameter estimates:

> semPaths(scs_model_fit, "std", curvePivot = TRUE, thresholds = FALSE)

Running multiple-group tests

Performing the multiple-group CFAs is slightly different because the function measurementInvariance() will try to constrain “loadings”, “intercepts” and “residuals”. Since residuals are not parameters in the delta-parameterization lavaan uses (see section 1.2 above), the function will produce meaningless output, i.e. comparing models that have identical constraints. So the individual models and comparisons to test for configural, weak, and strong invariance have to be specified by hand. In order to compare models the function semTools::difftest() will be used.

> config <- cfa(model, data=HolzingerSwineford1939, group="school")
> scs_model_weak <- cfa(scs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), group = "gender", group.equal = c("loadings"), data=scs)
> semTools::difftest(scs_model_config, scs_model_weak)

delta.chisq delta.df delta.p.value delta.cfi
175.840 9.000 0.000 0.002

The test for weak invariance results in a significant scaled chi-square test but a delta CFI that is below the cutpoint of .01. Since chi-square is sensitive to sample size, we assume that the scale still exhibits weak invariance and proceed to testing for strong invariance.

Inspection of the output shows that this model only has a very weak fit to the data (χ² = 2356.933; DF = 35; CFI = 0.94; RMSEA = .14). As in the first example, model fit is far from acceptable, but we proceed with the testing hypothesis about weak, strong, and strict invariance.

> config <- cfa(model, data=HolzingerSwineford1939, group="school")
> scs_model_weak <- cfa(scs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), group = "gender", group.equal = c("loadings"), data=scs)
> semTools::difftest(scs_model_config, scs_model_weak)

delta.chisq delta.df delta.p.value delta.cfi
175.840 9.000 0.000 0.002

The test for weak invariance results in a significant scaled chi-square test but a delta CFI that is below the cutpoint of .01. Since chi-square is sensitive to sample size, we assume that the scale still exhibits weak invariance and proceed to testing for strong invariance.
> scs_model_strong <- cfa(scs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), group = "gender", group.equal = c("loadings", "thresholds"), data=scs)
> semTools:::difftest(scs_model_weak, scs_model_strong)

<table>
<thead>
<tr>
<th>delta.chisq</th>
<th>delta.df</th>
<th>delta.p.value</th>
<th>delta.cfi</th>
</tr>
</thead>
<tbody>
<tr>
<td>-31.027</td>
<td>29.000</td>
<td>1.000</td>
<td>-0.001</td>
</tr>
</tbody>
</table>

Due to different scaling parameters in the models, the differences in chi-square and CFI may also be negative. These indicate however, that the strong invariance assumption still holds. As a comparison we also repeat the analysis without declaring the variables as ordered.

> measurementInvariance(scs_model, data=scs, group="gender", strict=TRUE)

Measurement invariance tests:

<table>
<thead>
<tr>
<th>Model 1 versus model 2</th>
<th>delta.chisq</th>
<th>delta.df</th>
<th>delta.p.value</th>
<th>delta.cfi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24.317</td>
<td>9.000</td>
<td>0.004</td>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model 2 versus model 3</th>
<th>delta.chisq</th>
<th>delta.df</th>
<th>delta.p.value</th>
<th>delta.cfi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>74.056</td>
<td>9.000</td>
<td>0.000</td>
<td>0.004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model 3 versus model 4</th>
<th>delta.chisq</th>
<th>delta.df</th>
<th>delta.p.value</th>
<th>delta.cfi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11.996</td>
<td>10.000</td>
<td>0.285</td>
<td>0.000</td>
</tr>
</tbody>
</table>

The output of this analysis gives very similar results. Specifically, the tests for weak and strong invariance yield a significant chi-square test but a small ΔCFI. The test for strict invariance yields both an insignificant chi-square test and a negligible ΔCFI. Both methods – WLSMV estimation and ML estimation – suggest that researchers may interpret differences in the means between those two groups as reflecting differences in the underlying latent trait rather than the measure.

Conclusions

Testing for measurement invariance is a central aspect of assessment and evaluation (Byrne et al., 1989; Chen, 2008; Cheung & Rensvold, 1999; Vandenberg & Lance, 2000; Wu et al., 2007). Even though item response theory can also be used to test for invariance (Raju et al., 2002; Reise et al., 1993), multiple group confirmatory factor analysis is the most widely used method to establish invariant measurements across groups.

Our description made apparent several areas where systematic simulation studies and software development is necessary. First, systematic simulation studies need to compare the relative utility of different decision rules for invariance tests. Studies using categorical data and WLSMV estimation would be especially useful to close the gap between single-group CFA where these estimation methods are widely used and multiple-group CFA for which most researchers still use ML estimation irrespective of the nature of the data (Koh & Zumbo, 2008). Second, further software development is also needed. We applaud the goal of the developers of the lavaan package to implement techniques available in commercial package. We believe that functions that are missing in the present version (0.5-16), e.g. theta-parameterization, will further increase the utility and adoption of this package.

We hope that the present manuscript showing how measurement invariance studies can be implemented in the open-source software R, will be useful to other researchers working with latent variables who want to performing such analysis.

References


Epskamp, S. (2013). semPlot: Path diagrams and visual analysis of various SEM packages’ output. R package version 0.3.2.


Appendix R Script

```r
# 1. Install / load packages
#install.packages(c("lavaan", "semTools", "semPlot", "ggplot2"))

library(lavaan)
library(semPlot)
library(semTools)
options(width = 22)

setwd("/Users/gerrit/Documents/Forschung/31_MG-CFA/1_Intro_paper/1_analysis")

data(HolzingerSwineford1939)
str(HolzingerSwineford1939)

model <- ' 
visual =~ x1 + x2 + x3 
textual =~ x4 + x5 + x6 
speed =~ x7 + x8 + x9 
'

fit <- cfa(model, data=HolzingerSwineford1939)
summary(fit, standardized = FALSE, fit.measures = TRUE)
moreFitIndices(fit)

semPaths(fit, rotation = 2, layout = "tree2", nCharNodes = 0, sizeLat = 15,
sizeLat2 = 7, label.norm = "OOOOO", mar=c(2,6,2,4), curvePivot = TRUE,
edge.label.cex=1.2, residuals = F)
dev.print(png, "fig_1_measurement.png", width=6, height=4, res=300,
units="in")

semPaths(fit, "std", rotation = 2, layout = "tree2", nCharNodes = 0, sizeLat =
15, sizeLat2 = 7, label.norm = "OOOOO", mar=c(2,6,2,4), curvePivot = TRUE,
edge.label.cex=1.2, residuals = F)
dev.print(png, "fig_2_cfa.png", width=8, height=4, res=300, units="in")

#Multiple Group CFA
config <- cfa(model, data=HolzingerSwineford1939, group="school")
weak <- cfa(model, data=HolzingerSwineford1939, group="school",
group.equal="loadings")
strong<- cfa(model, data=HolzingerSwineford1939, group="school", group.equal =
c("loadings", "intercepts"))
strict<- cfa(model, data=HolzingerSwineford1939, group="school", group.equal =
c("loadings", "intercepts", "residuals"))
anova(config, weak, strong, strict)
measurementInvariance(model, data=HolzingerSwineford1939, group="school",
strict=TRUE)

mod_strong<-modificationIndices(strong)
mod_strong[mod_strong$op == "~1",]

measurementInvariance(model, data=HolzingerSwineford1939, group="school",
group.partial = c("x3 ~1"))
measurementInvariance(model, data=HolzingerSwineford1939, group="school",)
```

Published by ScholarWorks@UMass Amherst, 2014
# Example 2: Categorical indicators

```r
# Download file
download.file("http://personality-testing.info/_rawdata/SCS.zip","SCS.zip")
unzip("SCS.zip")
schs <- read.csv("SCS/data.csv")
schs <- subset(schs, gender == "1" | gender == "2")

# SCs model
scs_model <- 'scs =~ Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7 + Q8 + Q9 + Q10'
schs_model_fit <- cfa(schs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), data=schs)
summary(schs_model_fit, fit.measures = TRUE)

# SCs model config
scs_model_config <- cfa(schs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), group = "gender", data=schs)

# SCs model weak
scs_model_weak <- cfa(schs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), group = "gender", group.equal = c("loadings"), data=schs)

# SCs model strong
scs_model_strong <- cfa(schs_model, ordered = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9", "Q10"), group = "gender", group.equal = c("loadings", "thresholds"), data=schs)

measurementInvariance(schs_model, data=schs, group="gender", strict=TRUE)
```

**Citation:**


**Authors:**

Gerrit Hirschfeld (corresponding author)  
German Paediatric Pain Centre  
Children’s Hospital Datteln  
Dr.-Friedrich-Steiner Str. 5  
45711 Datteln, Germany  
eMail: g.hirschfeld [at] deutsches-kinderschmerzzentrum.de

Ruth von Brachel  
Department of Clinical Psychology and Psychotherapy  
Ruhr-University Bochum, Bochum, Germany