Practical Assessment, Research & Evaluation

A peer-reviewed electronic journal.

Copyright is retained by the first or sole author, who grants right of first publication to *Practical Assessment, Research & Evaluation.* Permission is granted to distribute this article for nonprofit, educational purposes if it is copied in its entirety and the journal is credited. PARE has the right to authorize third party reproduction of this article in print, electronic and database forms.

Volume 29 Number 8, April 2024 ISSN 1531-7714

Frequentist and Bayesian Factorial Invariance using R

Teck Kiang Tan, *National University of Singapore*

The procedures of carrying out factorial invariance to validate a construct were well developed to ensure the reliability of the construct that can be used across groups for comparison and analysis, yet mainly restricted to the frequentist approach. This motivates an update to incorporate the growing Bayesian approach for carrying out the Bayesian factorial invariance, as well as the frequentist approach, using the recent add-on R packages to show the procedures systematically for testing measurement equivalence via multigroup confirmatory factor analysis. The practical procedure and guidelines for carrying out factorial invariance under MCFA using a classic empirical example are demonstrated. Comparison between the frequentist and the Bayesian procedures and demonstration using priors are another two nuclei of the paper.

Keywords: Measurement invariance, Multigroup confirmatory factor analysis, Frequentist factorial invariance, Bayesian factorial invariance, R package

Introduction

Self-reported measurement instruments through surveys and questionnaires are often used in research and practice to obtain a latent construct or a group of constructs when the construct cannot be directly observed and measured. Many examples, such as self-esteem and self-efficacy, are usually approximated by a scale using observed items to derive the construct. The main concern of this approach of deriving a latent construct to form a scale is that it needs a process of validation to ensure its reliability and validity which can be used across groups for comparison and analysis. When a measurement instrument can sufficiently maintain its measurement structure across groups, it is referred to as factorial invariant (FI) or measurement invariant. The lack of it indicates the latent construct cannot be interpreted in the same way across groups. Factorial invariance is thus the condition setting for an instrument measure to indicate the level of validity that could be used across population subgroups.

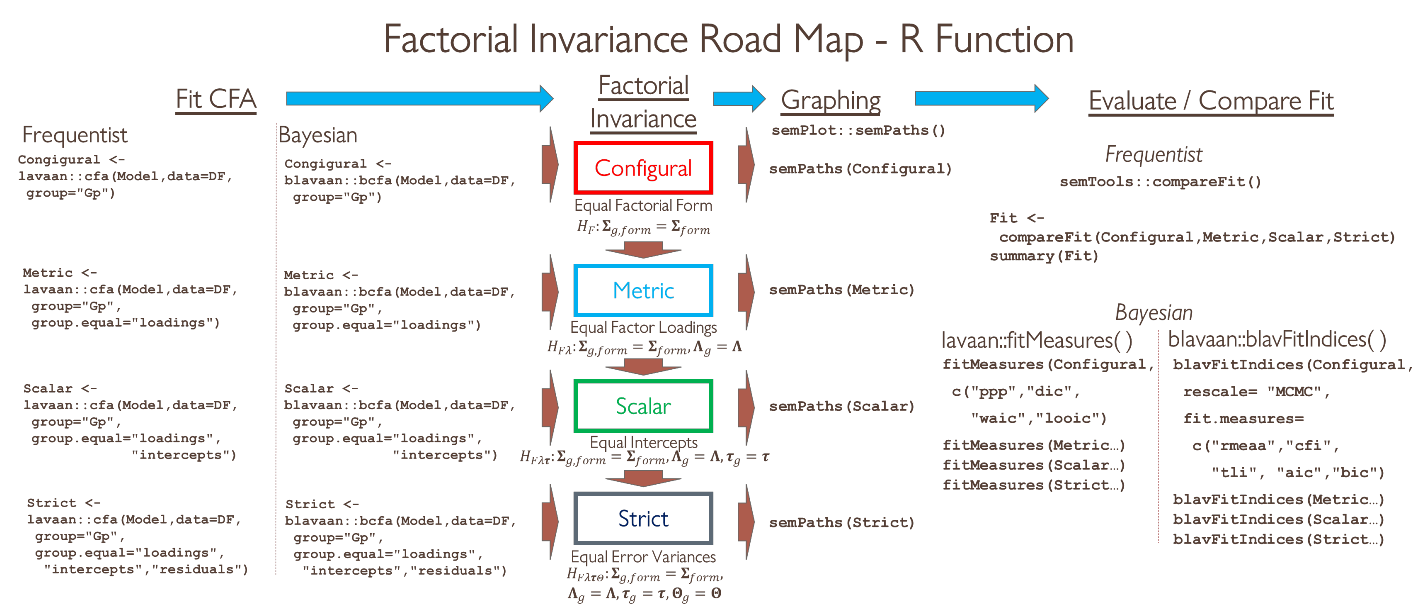
Validation of factorial invariance becomes a common procedure, a pre-requisite practice, and a requirement to be carried out before using the instrument to conduct further analyses. For instance, Tan & Feng (2022) carried out the FI validation process for a medical assessment. Bagheri et al., (2022) validated an instrument concerning life enjoyment & satisfaction instruments using FI. However, not all researchers treat it as a mandatory step. One plausible explanation is that there is a lack of requisite technical skills of the researchers to carry out the procedure and perhaps more importantly the lack of software that directly aims to carry out the procedure using simple syntax specification. The current paper fills this gap by introducing the R packages, using recently developed functions, presenting both the frequentist & Bayesian frameworks, and providing the procedure with straightforward syntax for carrying out FI.

A common analytical method to attain the FI of an instrument is to carry out the multigroup confirmatory factor analysis (MCFA) to ensure comparable differences across groups are achievable. MCFA has been applied in many areas of research and studies such as criminology, cross-cultural psychology, developmental psychology, education, gerontology, medical examination, sports psychology, marketing, and organizational sciences, among others (An et al., 2017; Feldt et al., 2014; Huansuriya et al., 2020; Lau & Yuen, 2015; Moreira et al., 2019; Oppzda-Suder et al., 2021; Scheffers et al., 2017; Tan & Feng, 2022; Yu et al., 2019). The procedures of carrying out FI via MCFA to validate a construct were well developed, yet mainly restricted to the frequentist approach. This motivates an update to incorporate the growing Bayesian approach for carrying out the Bayesian factorial invariance, using the recent add-on R packages to show the procedures systematically for testing measurement equivalence. More importantly, the differences between the Bayesian and frequentist approaches for carrying out FI are seldom discussed in the literature concerning their practical considerations and providing direct help with codes, syntaxes, and references. Also, using priors is seldom discussed. This paper elaborates with an example to show how Bayesian factorial invariance can be effectively applied to determine whether cross-loadings are necessary to confirm the structural form of a CFA model. The main intended readers for this paper are those with little idea of carrying out FI, shown systematically to demonstrate how to carry out FI nonetheless with an understanding of CFA (e.g. Bollen, 1989) is presumed. Another focus of the paper is to update readers that are not familiar with Bayesian FI. Even so, both the frequentist and Bayesian approaches are described. Discussion on the benefits and limitations of the two approaches will be given at the end of the paper. Basic R knowledge is assumed for readers to benefit from the paper.

The main goals of the paper are to (a) formally state the multiple group confirmatory factor analysis to introduce the concept of factorial invariance under this analytical framework; (b) discuss the state-of-the-practice of factorial invariance under the frequentist and Bayesian framework; (c) provide practical procedures and guidelines for carrying out factorial invariance under MCFA with a roadmap of R functions stated at the beginning of the paper to give an overall R functions route, and (d) present an empirical example of factorial invariance using recently developed R packages.

An overall representation that summarizes the R functions for carrying out FI is given in Figure 1. The first set of syntaxes on the left of the roadmap shows the specification of generating the four FI models, namely configural, metric, scalar, and strict models for both the frequentist and Bayesian approaches using the two packages lavaan and blavaan respectively (Merkle et al., 2021; Rosseel, 2012) together with their purposes and hypotheses stated on their right. The syntaxes of graphing of these four models via the package

Figure 1. Factorial Invariance Road Map – R Function



semPlot follow (Epskamp, 2022). The fit functions for both approaches are stated on the utmost right of the roadmap (Jorgensen et al., 2022). The main R package is semTools. The details of these functions will be illustrated in the paper.

Multigroup Confirmatory Factor Analysis

The multigroup confirmatory factor analysis (MCFA) specification for carrying out factorial invariance (FI) is briefly described here. Equation 1 states the formal formula for FI under MCFA for g subpopulation groups. For subject j in group g, y_{jg} represent the observed scores, η_{jg} represents the latent factor scores, τ_g represents the item intercepts, Λ_g represents the loadings, and δ_{jg} represents the error terms,

$$y_{jg} = \tau_g + \Lambda_g \eta_{jg} + \delta_{jg} \tag{1}$$

The implied variance and covariance among the items in the gth subpopulation group, Σ_g, is stated in Equation 2 where the measurement errors of the latent variances and covariances are denoted as Θ_g, and Φ_g respectively.

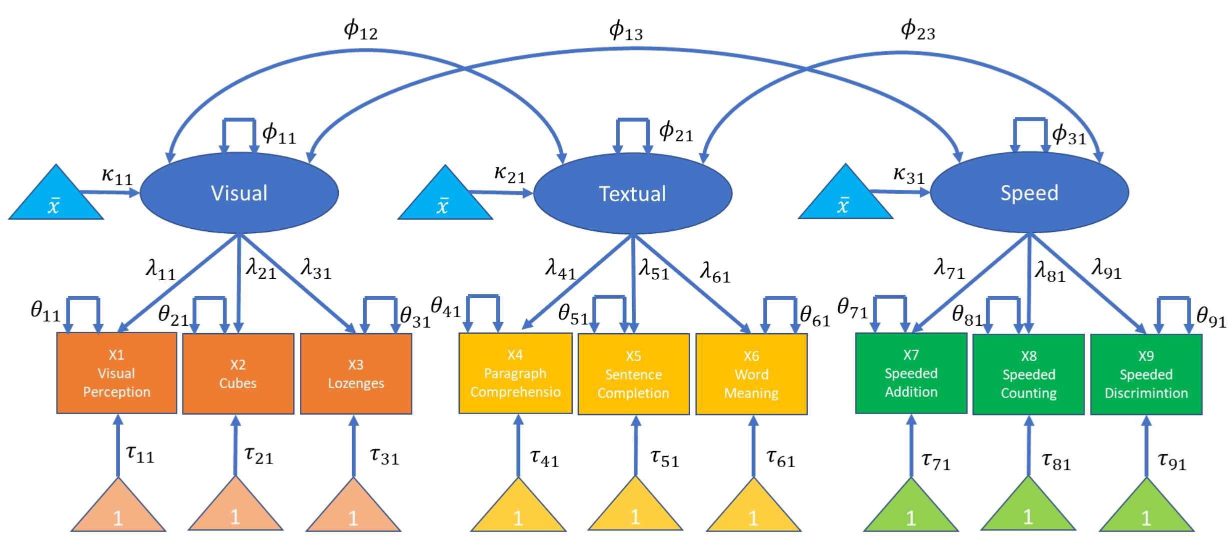
$$\Sigma_g = \Lambda_g \Phi_g \Lambda_g' + \Theta_g \tag{2}$$

The classic example of Holzinger & Swineford (1939) is used for demonstration of how to perform a multigroup three-factor CFA under both frequentist and Bayesian approaches. This widely used dataset consists of mental ability test scores of seventh- and eighth-grade children from two different schools (Pasteur and Grant-White) and is one best example to show the stability of a CFA solution that is invariant across the two schools under the multigroup analysis. Figure 2 displays the path diagram of this 3-factor MCFA model with 3 constructs: Visual, Textural, and Speed, each with three indicators. These 9 indicators are visual perception (x1), cubes (x2), lozenges (x3), paragraph comprehension (x4), sentence completion (x5), word meaning (x6), speeded addition (x7), speeded counting of dots (x8), speeded discrimination straight and curved capitals (x9), for the first three indicators load to Visual, the next three into Textural and the last three into Speed. The ovals represent latent factors, rectangles represent manifest items, triangles with x and 1 inserted represent the means and the intercepts respectively, while the single-headed arrows represent the values of the regression parameters or intercept, and double-headed arrows represent the common factor variances and covariances. The parameters $\lambda_{11}, \dots, \lambda_{91}$ represent factor loadings, $\tau_{11}, \dots, \tau_{91}$ represent item intercepts, $\theta_{11}, \dots, \theta_{91}$ represent item residual variances, $\phi_{11}, \dots, \phi_{31}$ represent factor variances and covariances, and $\kappa_{11}, \dots, \kappa_{31}$ represents factor means.

Reading, Visualizing, and Exploring Data

The first step of analysis is to read the data. The dataset used in the current paper is originally from Holzinger and Swineford (1939). In the original dataset, there were scores for 26 tests. For illustration

Figure 2. Multigroup Three-Factor CFA Path Diagram



purposes, a smaller subset with 9 variables was used with 301 subjects, out of which 145 students from Grant-White and 156 students from Pasteur. The dataset is available in package `lavaan`, named HolzingerSwineford1939, and renamed here as H. It is restricted to the 9 indicators and the group variable school. The R syntax and file structure are printed below.

```
Library(lavaan)
H <- HolzingerSwineford1939
H <- H[,c(5,7:15)]
str(H)
> str(H)
'data.frame':   301 obs. of  10 variables:
 $ school: Factor w/ 2 levels "Grant-White",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ x1    : num  3.33 5.33 4.5 5.33 4.83 ...
 $ x2    : num  7.75 5.25 5.25 7.75 4.75 5 6 6.25 5.75 5.25 ...
 $ x3    : num  0.375 2.125 1.875 3 0.875 ...
 $ x4    : num  2.33 1.67 1 2.67 2.67 ...
 $ x5    : num  5.75 3 1.75 4.5 4 3 6 4.25 5.75 5 ...
 $ x6    : num  1.286 1.286 0.429 2.429 2.571 ...
 $ x7    : num  3.39 3.78 3.26 3 3.7 ...
 $ x8    : num  5.75 6.25 3.9 5.3 6.3 6.65 6.2 5.15 4.65 4.55 ...
 $ x9    : num  6.36 7.92 4.42 4.86 5.92 ...
```

Before carrying out confirmatory factor modeling and factorial invariance, exploring and visualizing the data by graphing a heatmap helps to examine its dimensions. This facilitates the determination of whether the model specified in Figure 2 is in line with the theoretical structural framing. The functions corrplot.mixed, and corrplot from the package corrplot (Wei & Simko, 2021) provide visualization of a correlation heatmap for the former, and with hierarchical clustering inserted it into the heatmap for the latter. By default, the corrplot.mixed function produces a heatmap with printed correlation coefficients on the lower diagonal and circles with colors on the upper diagonal. The color and the size of the circle as well as the magnitude of the estimated coefficient, are associated with the extent of the correlation being the darker the color, the larger the circle, and the higher the printed size of the coefficient, the higher the coefficient of the correlation, and vice versa. The first graph of Figure 3 indicates the correlation coefficients are highest among the indicators x4, x5, and x6 (0.70 to 0.73), moderately high among x7, x8, and x9 (0.34 to 0.49) as well as among x1, x2, and x3 (0.30 to 0.44). The rest of the correlation coefficients are relatively low in magnitude. The corrplot function produces the hierarchical clustering solutions with thicker lines enclosing the variables that belonged to the cluster. The arguments `order="hclust"` indicates hierarchical clustering to be carried out and `addrect=3` states to produce a three-cluster solution. The second graph of Figure 3 indicates a clear three-cluster solution.

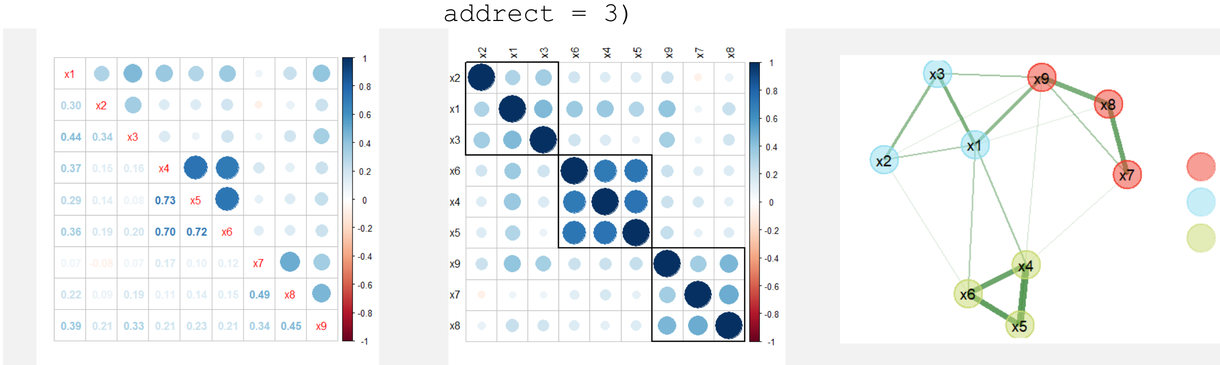
Exploratory graph analysis, a network psychometrics method, is another useful analytical tool to find out the number of dimensions and display the degree of association of indicators by the line thickness and colors to demonstrate the number of dimensions. The ECGnet package (Golino & Christensen, 2022), function ECG, employs the walktrap algorithm (Golino & Epskamp, 2017; Pons & Latapy, 2006) to generate the expolatory graph plot. The third graph, on the right of Figure 3, produces this exploratory graph showing a clear three-clustered classification differentiated by three sets of colors. The strong relationship among x4, x5, and x6 is indicated by the three thick lines enclosing and linking these three indicators. The distance of these three indicators is away from the rest of the indicators with thinner lines showing the strong association of these 3 indicators as a dimension, and they are distanced from the rest. However, the thinner line between x7 and x9 and the thicker line between x1 and x9, show a lower association between x7 and x9, and a moderately higher association between x1 and x9, indicating the two worries about the possible disturbance from the stated structure. Nevertheless, a three-cluster solution is suggested by exploratory graph analysis. In conclusion, the three graphs in Figure 3 indicate a

Figure 3. Heatmap and Exploratory Graph Analysis

```
library(corrplot)
Corr <- round(cor©,2)
library(corrplot)
corrplot.mixed(Corr)
```

```
library(corrplot)
corrplot(Corr,
 order    = "hclust",
 tl.col   = 'black',
 addrect = 3)
```

```
Library(ECGnet)
EGA(data)
```



clear three-factor solution for the 9 indicators, in line with the Figure 2 three-factor CFA specifications.

Factorial Invariance

While many FI procedures were suggested, reported, and used in validity studies, the current paper concentrates on one of the common measurement equivalence practices that arrange an ordered sequence starting from the least restricted configural factorial invariance, moving on to metric invariance, scalar invariance, and ending at the most restrictive strict factorial invariance. Table 1 exhibits the hypotheses for these four invariant tests. The first hypothesis H_F states the factorial forms of group g are all the same, $\Sigma_{g,form} = \Sigma_{form}$. The hypothesis $H_{F\lambda}$ sets the equality of factor loadings for groups g, $\Lambda_g = \Lambda$, adds to the equality of the structural form of the configural, named metric invariance. The next hypothesis $H_{F\lambda\tau}$, the scalar invariance, further qualifies the equality of thresholds/intercepts equality, $\tau_g = \tau$. The strict invariance hypothesis $H_{F\lambda\tau\Theta}$ includes the specification of equal residual across groups hypothesis, $\Theta_g = \Theta$, ended with 4 equality settings.

These four basic factorial models and hypotheses express the level of factorial invariance. Configural invariance model is the most basic factorial invariance that specifies the CFA structures are the same across all the groups, ensuring that the subpopulations are not disjointed, and indicating the same subsets of items within the same construct, often used as the baseline for equivalence testing (Vandenberg & Lance 2000). The metric invariance (Widaman & Reise, 1997), generally referred to as a test of a weak factorial null hypothesis (Horn & McArdle, 1992), is considered an assessment of scaling unit equality (Riordan & Vandenberg, 1994), indicating that the latent factor has the same influence on items across groups. The scalar invariance (Steenkamp & Baumgartner, 1998), also referred to as strong invariance (Meredith, 1993), indicates that the latent factor differences account for mean differences in items across groups (Ghosh et al., 2021). The strict invariance is a test for equality of random errors across groups to address the reliability issue of inconsistent scoring and the validity issue of scalar equivalence. In conclusion, the hierarchy of the four invariance tests reflects the level of factorial invariance from the less restrictive of the configural to the more stringent strict invariance CFA model. In a nutshell, Table 1 summarizes these four factorial invariance tests.

Fit Modeling and Graphing Multigroup Confirmatory Factor Model

The factorial structure of the 3-factor CFA model has to be specified before carrying out the factorial invariance tests. The R syntax of this structural form is stored as a character named HS.Model, as stated below. For instance, variables x1, x2, and x3 are loaded to the latent factor visual (`visual=~x1+x2+x3`), and variables x7, x8, and x9 are loaded to the latent factor speed (`speed=~x7+x8+x9`). The name of the latent construct is specified before the symbol "=~" and the names of the indicators after it.

```
HS.Model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '
```

The functions cfa and bcfa respectively from the package lavaan (Rosseel, 2012) and blavaan (Merkle et al., 2018; Merkle et al., 2021) are the two main functions for generating the frequentist and Bayesian CFA factorial invariances models respectively. Since the syntaxes are the same for these two functions except for the name of the function, only a single description of the syntax is given unless there are differences that need to be highlighted.

The first argument of the function cfa states the name of the structural form of CFA, `HS.Model`, and the dataset to be read in by specifying "`data=H`". The group argument specifies the name of the group variable (`group=school`). Without specifying the group.equal argument, it indicates all the parameters are freely estimated. As such, the configural invariance CFA is specified. The output of CFA is stored in an R object named CFA.Configural, a lavaan class R object which is fundamentally a list structure R output that stores all the frequentist estimates. For generating the output of the Bayesian configural invariance CFA, the output from the function bcfa is a blavaan class, stored in an R object named CFA.Configural.B which is also a list that stores the estimates of Bayesian estimates. For the rest of the specifications of factorial invariance models, the syntax is similar to that of the configural invariance model by adding on the relevant reserved words to state the models. Including the additional text "`loadings`" in the `group.equal argument` specifies the metric model, further adding "`intercepts`", and "`residuals`" specifying the scalar and strict invariance models respectively.

```
# Frequentist Approach
library(lavaan)
CFA.Configural <- cfa(HS.Model, data=H,
    group       = "school")
CFA.Metric <- cfa(HS.Model, data=H,
    group       = "school",
    group.equal = c("loadings"))
CFA.Scalar <- cfa(HS.Model, data=H,
    group       = "school",
    group.equal = c("loadings","intercepts"))
CFA.Strict <- cfa(HS.Model, data=H,
    group       = "school",
group.equal = c("loadings","intercepts","residuals"))

# Bayesian Approach
library(blavaan)
CFA.Configural.B <- bcfa(HS.Model, data=H,
    group       = "school")
CFA.Metric.B <- bcfa(HS.Model, data=H,
    group       = "school",
    group.equal = "loadings")
CFA.Scalar.B <- bcfa(HS.Model, data=H,
    group       = "school",
    group.equal = c("loadings","intercepts"))
CFA.Strict.B <- bcfa(HS.Model, data=H,
    group       = "school",
    group.equal = c("loadings","intercepts","residuals"))
```

Examining Frequentist Approach Results. The four sections of Figure 4 print the frequentist outputs of configural, metric, scalar, and strict invariance models. The estimated factor loadings, intercepts, and residuals of these models clearly show the differences of these four sets of CFA outputs. For the configural invariance model, all these estimates for the two groups varied since this model specifies identical structural forms but their loading, intercept, and residual parameters differ. For instance, the factor loading of x5 is 1.183 under Group 1 Pasteur differs and is higher than x5 of 0.990 under Group 2 Grant-White. The intercept of x6 for Group 1 (1.922) and Group 2 (2.469) also differs substantially. Similarly, the variances of x1 for Group

Table 1. Invariance Test – Configural, Metric, Scalar, and Strict Invariance

Invariance	Hypothesis
Configural	$H_F: \boldsymbol{\Sigma}_{g,form} = \boldsymbol{\Sigma}_{form}$
Metric	$H_{F\lambda}: \boldsymbol{\Sigma}_{g,form} = \boldsymbol{\Sigma}_{form}, \boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$
Scalar	$H_{F\lambda\boldsymbol{\tau}}: \boldsymbol{\Sigma}_{g,form} = \boldsymbol{\Sigma}_{form}, \boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}, \boldsymbol{\tau}_g = \boldsymbol{\tau}$
Strict	$H_{F\lambda\boldsymbol{\tau}\Theta}: \boldsymbol{\Sigma}_{g,form} = \boldsymbol{\Sigma}_{form},, \boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}, \boldsymbol{\tau}_g = \boldsymbol{\tau}, \boldsymbol{\Theta}_g = \boldsymbol{\Theta}$

1 (0.298) and Group 2 (0.715) also have a large discrepancy in their estimates. For the metric factorial invariance model that puts the equality of the factor loadings for both groups, the factor loadings for Group 1 and Group 2 now have the same value for x5 of 1.083, however, the estimates of intercept and variance differ. The scalar invariance model further restricts the equality of the intercepts. For instance, this model produces the same intercept value for x6 of 1.926. Under the strict invariance model, the further equality restriction produces the residuals with the same value. For instance, the residual of x1 for the two groups is with the same value of 0.635.

While these outputs print the estimates, they do not evaluate which model fits best. Having said that, the difference between the two groups could be compared to obtain an overall inkling of which model fits better. For instance, the factor loadings under the configural model do not differ much for the two groups, giving a sense that it is probably wise to proceed from the configural model to the metric invariance model as the latter model that specifies equality of loadings is a better invariance model than the configural model.

Graphing Frequentist CFA. Graphing the path diagram after fixing a CFA model is a beneficial step to take a look at the model pictorially in understanding and examining the relationship of all the estimated parameters. Function semPaths from the package semTools (Epskamp, 2022) provide the plotting facility. The syntaxes of configural and strict CFA models are printed on top of Figure 5 and Figure 6 respectively and the path diagrams below them. The first argument of the function semPaths specifies the fitted CFA model. The second argument, what, states the estimated coefficients to be printed. The `color` and `edge.color` arguments specify the colors of the path diagram and its edges. The word size of the latent and manifest are specified under the argument `sizeLat` and `sizeMan` respectively, and the `edge.label.cex` argument specifies the word size of the label.

Figure 5 produces the path diagrams for both Group 1 and 2 of the configural invariance model showing all the estimated coefficients differ. For instance, by comparing the estimated factor loading coefficients of x2, with an estimated value of 0.39 for Group 1, it is much lower than 0.74 for Group 2. Figure 6 graphs the path diagrams for the strict invariance model showing the factor loading of the x2 now having the same values of 0.59 as well as the intercept of x2 also having the same values of 6.13, and variance value of 1.13. Those coefficients printed in blue are significant estimates and those shown in light grey are non-significant estimates.

Examining Bayesian Output. The syntax of generating the factorial invariance models for the Bayesian is almost the same as that of the frequentist and the output format is also similar in layout but the output contents differ. While the package lavaan uses maximum likelihood estimation to produce point estimates, the package blavaan generates the posterior values producing a set of estimates for each parameter. The default output from the Bayesian using the summary function, as stated below, prints the posterior mean values, standard deviation, lower-density interval value, upper-density interval value, Rhat, and prior specification.

```
Summary(CFA.Configural.B)
summary(CFA.Metric.B)
summary(CFA.Scalar.B)
summary(CFA.Strict.B)
```

Figure 4. Frequentist Output – Configural, Metric, Scalar, and Strict Invariance

<table>
<tr><th>Invariance</th><th>Group 1 – Pasteur</th><th>Group 2 – Grant-White</th></tr>
<tr><td rowspan="2">Configural</td><td>Group 1 [Pasteur]:

Latent Variables:
Estimate Std.Err z-value P(>|z|)
Visual =~
x1 1.000
x2 0.394 0.122 3.220 0.001
x3 0.570 0.140 4.076 0.000
Textual =~
x4 1.000
x5 1.183 0.102 11.613 0.000
x6 0.875 0.077 11.421 0.000
Speed =~
x7 1.000
x8 1.125 0.277 4.057 0.000
x9 0.922 0.225 4.104 0.000</td><td>Group 2 [Grant-White]:

Latent Variables:
Estimate Std.Err z-value P(>|z|)
Visual =~
x1 1.000
x2 0.736 0.155 4.760 0.000
x3 0.925 0.166 5.583 0.000
Textual =~
x4 1.000
x5 0.990 0.087 11.418 0.000
x6 0.963 0.085 11.377 0.000
Speed =~
x7 1.000
x8 1.226 0.187 6.569 0.000
x9 1.058 0.165 6.429 0.000</td></tr>
<tr><td>Intercepts:
Estimate Std.Err z-value P(>|z|)
.x1 4.941 0.095 52.249 0.000
.x2 5.984 0.098 60.949 0.000
.x3 2.487 0.093 26.778 0.000
.x4 2.823 0.092 30.689 0.000
.x5 3.995 0.105 38.183 0.000
.x6 1.922 0.079 24.321 0.000
.x7 4.432 0.087 51.181 0.000
.x8 5.563 0.078 71.214 0.000
.x9 5.418 0.079 68.440 0.000</td><td>Intercepts:
Estimate Std.Err z-value P(>|z|)
.x1 4.930 0.095 51.696 0.000
.x2 6.200 0.092 67.416 0.000
.x3 1.996 0.086 23.195 0.000
.x4 3.317 0.093 35.625 0.000
.x5 4.712 0.096 48.986 0.000
.x6 2.469 0.094 26.277 0.000
.x7 3.921 0.086 45.819 0.000
.x8 5.488 0.087 63.174 0.000
.x9 5.327 0.085 62.571 0.000</td></tr>
</table>

Invariance	Group 1 – Pasteur	Group 2 – Grant-White

Group 1 – Pasteur:

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1                0.298    0.232    1.286    0.198
   .x2                1.334    0.158    8.464    0.000
   .x3                0.989    0.136    7.271    0.000
   .x4                0.425    0.069    6.138    0.000
   .x5                0.456    0.086    5.292    0.000
   .x6                0.290    0.050    5.780    0.000
   .x7                0.820    0.125    6.580    0.000
   .x8                0.510    0.116    4.406    0.000
   .x9                0.680    0.104    6.516    0.000
    Visual            1.097    0.276    3.967    0.000
    Textual           0.894    0.150    5.963    0.000
    Speed             0.350    0.126    2.778    0.005
```

Group 2 – Grant-White:

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1                0.715    0.126    5.676    0.000
   .x2                0.899    0.123    7.339    0.000
   .x3                0.557    0.103    5.409    0.000
   .x4                0.315    0.065    4.870    0.000
   .x5                0.419    0.072    5.812    0.000
   .x6                0.406    0.069    5.880    0.000
   .x7                0.600    0.091    6.584    0.000
   .x8                0.401    0.094    4.249    0.000
   .x9                0.535    0.089    6.010    0.000
    Visual            0.604    0.160    3.762    0.000
    Textual           0.942    0.152    6.177    0.000
    Speed             0.461    0.118    3.910    0.000
```

Metric

Group 1 – Pasteur:

```
Group 1 [Pasteur]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  Visual =~
    x1                1.000
    x2      (.p2.)    0.599    0.100    5.979    0.000
    x3      (.p3.)    0.784    0.108    7.267    0.000
  Textual =~
    x4                1.000
    x5      (.p5.)    1.083    0.067   16.049    0.000
    x6      (.p6.)    0.912    0.058   15.785    0.000
  Speed =~
    x7                1.000
    x8      (.p8.)    1.201    0.155    7.738    0.000
    x9      (.p9.)    1.038    0.136    7.629    0.000
```

```
Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1                4.941    0.093   52.991    0.000
   .x2                5.984    0.100   60.096    0.000
   .x3                2.487    0.094   26.465    0.000
   .x4                2.823    0.093   30.371    0.000
   .x5                3.995    0.101   39.714    0.000
   .x6                1.922    0.081   23.711    0.000
   .x7                4.432    0.086   51.540    0.000
   .x8                5.563    0.078   71.087    0.000
   .x9                5.418    0.079   68.153    0.000
```

Group 2 – Grant-White:

```
Group 2 [Grant-White]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  Visual =~
    x1                1.000
    x2      (.p2.)    0.599    0.100    5.979    0.000
    x3      (.p3.)    0.784    0.108    7.267    0.000
  Textual =~
    x4                1.000
    x5      (.p5.)    1.083    0.067   16.049    0.000
    x6      (.p6.)    0.912    0.058   15.785    0.000
  Speed =~
    x7                1.000
    x8      (.p8.)    1.201    0.155    7.738    0.000
    x9      (.p9.)    1.038    0.136    7.629    0.000
```

```
Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1                4.930    0.097   50.763    0.000
   .x2                6.200    0.091   68.379    0.000
   .x3                1.996    0.085   23.455    0.000
   .x4                3.317    0.092   35.950    0.000
   .x5                4.712    0.100   47.173    0.000
   .x6                2.469    0.091   27.248    0.000
   .x7                3.921    0.086   45.555    0.000
   .x8                5.488    0.087   63.257    0.000
   .x9                5.327    0.085   62.786    0.000
```

Invariance	Group 1 – Pasteur	Group 2 – Grant-White
	Variances: Estimate Std.Err z-value P(>|z|) .x1 0.551 0.137 4.010 0.000 .x2 1.258 0.155 8.117 0.000 .x3 0.882 0.128 6.884 0.000 .x4 0.434 0.070 6.238 0.000 .x5 0.508 0.082 6.229 0.000 .x6 0.266 0.050 5.294 0.000 .x7 0.849 0.114 7.468 0.000 .x8 0.515 0.095 5.409 0.000 .x9 0.658 0.096 6.865 0.000 Visual 0.805 0.171 4.714 0.000 Textual 0.913 0.137 6.651 0.000 Speed 0.305 0.078 3.920 0.000	Variances: Estimate Std.Err z-value P(>|z|) .x1 0.645 0.127 5.084 0.000 .x2 0.933 0.121 7.732 0.000 .x3 0.605 0.096 6.282 0.000 .x4 0.329 0.062 5.279 0.000 .x5 0.384 0.073 5.270 0.000 .x6 0.437 0.067 6.576 0.000 .x7 0.599 0.090 6.651 0.000 .x8 0.406 0.089 4.541 0.000 .x9 0.532 0.086 6.202 0.000 Visual 0.722 0.161 4.490 0.000 Textual 0.906 0.136 6.646 0.000 Speed 0.475 0.109 4.347 0.000
Scalar	Group 1 [Pasteur]: Latent Variables: Estimate Std.Err z-value P(>|z|) Visual =~ x1 1.000 x2 (.p2.) 0.576 0.101 5.713 0.000 x3 (.p3.) 0.798 0.112 7.146 0.000 Textual =~ x4 1.000 x5 (.p5.) 1.120 0.066 16.965 0.000 x6 (.p6.) 0.932 0.056 16.608 0.000 Speed =~ x7 1.000 x8 (.p8.) 1.130 0.145 7.786 0.000 x9 (.p9.) 1.009 0.132 7.667 0.000	Group 2 [Grant-White]: Latent Variables: Estimate Std.Err z-value P(>|z|) Visual =~ x1 1.000 x2 (.p2.) 0.576 0.101 5.713 0.000 x3 (.p3.) 0.798 0.112 7.146 0.000 Textual =~ x4 1.000 x5 (.p5.) 1.120 0.066 16.965 0.000 x6 (.p6.) 0.932 0.056 16.608 0.000 Speed =~ x7 1.000 x8 (.p8.) 1.130 0.145 7.786 0.000 x9 (.p9.) 1.009 0.132 7.667 0.000
	Intercepts: Estimate Std.Err z-value P(>|z|) .x1 (.25.) 5.001 0.090 55.760 0.000 .x2 (.26.) 6.151 0.077 79.905 0.000 .x3 (.27.) 2.271 0.083 27.387 0.000 .x4 (.28.) 2.778 0.087 31.953 0.000 .x5 (.29.) 4.035 0.096 41.858 0.000 .x6 (.30.) 1.926 0.079 24.426 0.000 .x7 (.31.) 4.242 0.073 57.975 0.000 .x8 (.32.) 5.630 0.072 78.531 0.000 .x9 (.33.) 5.465 0.069 79.016 0.000	Intercepts: Estimate Std.Err z-value P(>|z|) .x1 (.25.) 5.001 0.090 55.760 0.000 .x2 (.26.) 6.151 0.077 79.905 0.000 .x3 (.27.) 2.271 0.083 27.387 0.000 .x4 (.28.) 2.778 0.087 31.953 0.000 .x5 (.29.) 4.035 0.096 41.858 0.000 .x6 (.30.) 1.926 0.079 24.426 0.000 .x7 (.31.) 4.242 0.073 57.975 0.000 .x8 (.32.) 5.630 0.072 78.531 0.000 .x9 (.33.) 5.465 0.069 79.016 0.000

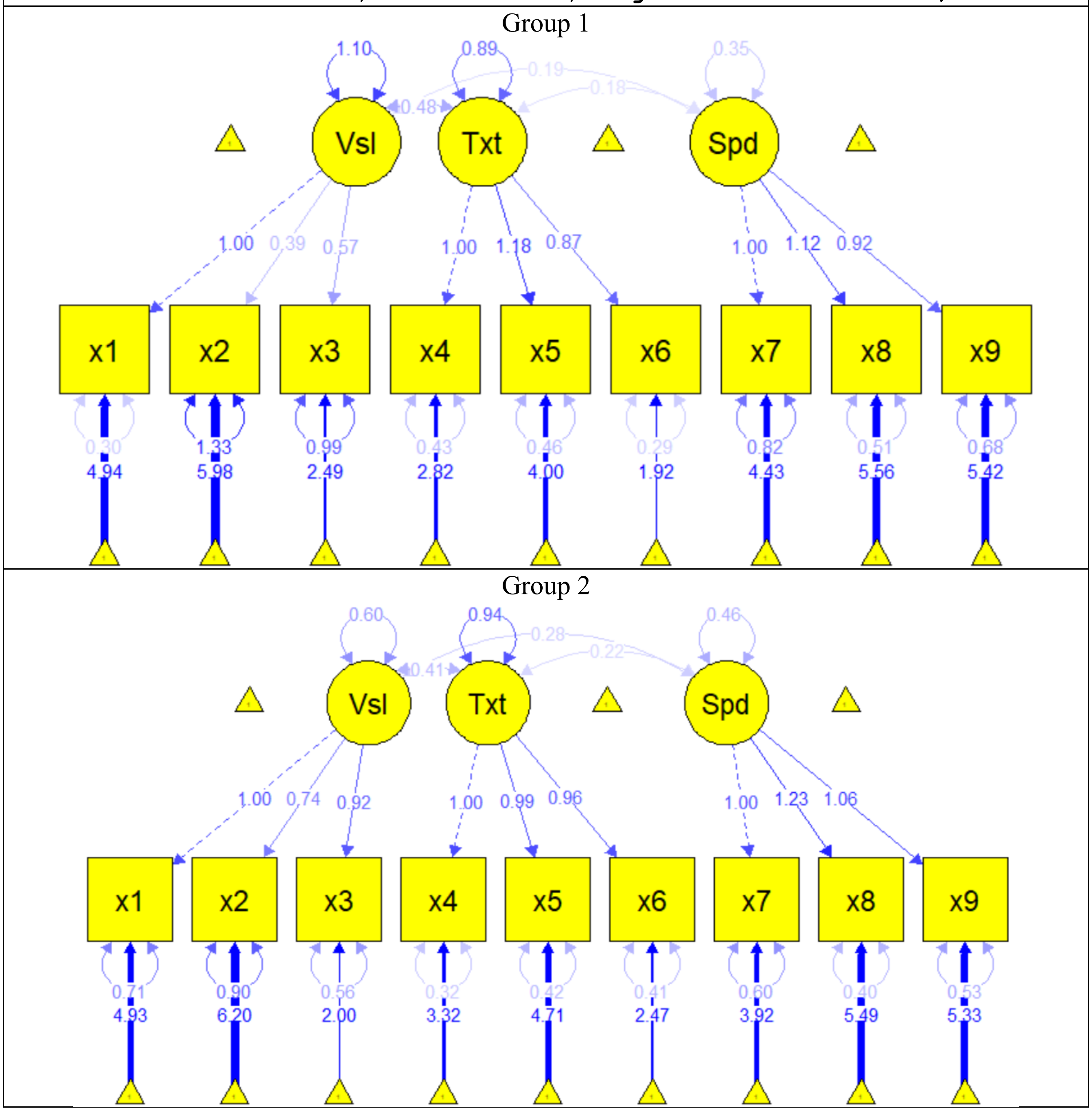
<table>
<tr><th>Invariance</th><th>Group 1 – Pasteur</th><th>Group 2 – Grant-White</th></tr>
<tr><td></td><td>Variances:
Estimate Std.Err z-value P(>|z|)
.x1 0.555 0.139 3.983 0.000
.x2 1.296 0.158 8.186 0.000
.x3 0.944 0.136 6.929 0.000
.x4 0.445 0.069 6.430 0.000
.x5 0.502 0.082 6.136 0.000
.x6 0.263 0.050 5.264 0.000
.x7 0.888 0.120 7.416 0.000
.x8 0.541 0.095 5.706 0.000
.x9 0.654 0.096 6.805 0.000
Visual 0.796 0.172 4.641 0.000
Textual 0.879 0.131 6.694 0.000
Speed 0.322 0.082 3.914 0.000</td><td>Variances:
Estimate Std.Err z-value P(>|z|)
.x1 0.654 0.128 5.094 0.000
.x2 0.964 0.123 7.812 0.000
.x3 0.641 0.101 6.316 0.000
.x4 0.343 0.062 5.534 0.000
.x5 0.376 0.073 5.133 0.000
.x6 0.437 0.067 6.559 0.000
.x7 0.625 0.095 6.574 0.000
.x8 0.434 0.088 4.914 0.000
.x9 0.522 0.086 6.102 0.000
Visual 0.708 0.160 4.417 0.000
Textual 0.870 0.131 6.659 0.000
Speed 0.505 0.115 4.379 0.000</td></tr>
<tr><td rowspan="2">Strict</td><td>Group 1 [Pasteur]:

Latent Variables:
Estimate Std.Err z-value P(>|z|)
Visual =~
x1 1.000
x2 (.p2.) 0.591 0.104 5.691 0.000
x3 (.p3.) 0.837 0.116 7.182 0.000
Textual =~
x4 1.000
x5 (.p5.) 1.125 0.066 17.134 0.000
x6 (.p6.) 0.933 0.056 16.752 0.000
Speed =~
x7 1.000
x8 (.p8.) 1.121 0.151 7.424 0.000
x9 (.p9.) 1.028 0.140 7.356 0.000</td><td>Group 2 [Grant-White]:

Latent Variables:
Estimate Std.Err z-value P(>|z|)
Visual =~
x1 1.000
x2 (.p2.) 0.591 0.104 5.691 0.000
x3 (.p3.) 0.837 0.116 7.182 0.000
Textual =~
x4 1.000
x5 (.p5.) 1.125 0.066 17.134 0.000
x6 (.p6.) 0.933 0.056 16.752 0.000
Speed =~
x7 1.000
x8 (.p8.) 1.121 0.151 7.424 0.000
x9 (.p9.) 1.028 0.140 7.356 0.000</td></tr>
<tr><td>Intercepts:
Estimate Std.Err z-value P(>|z|)
.x1 (.25.) 5.012 0.090 55.461 0.000
.x2 (.26.) 6.133 0.077 79.814 0.000
.x3 (.27.) 2.314 0.083 28.037 0.000
.x4 (.28.) 2.784 0.086 32.193 0.000
.x5 (.29.) 4.029 0.096 41.812 0.000
.x6 (.30.) 1.927 0.081 23.747 0.000
.x7 (.31.) 4.271 0.073 58.428 0.000
.x8 (.32.) 5.622 0.072 78.502 0.000
.x9 (.33.) 5.461 0.070 78.438 0.000</td><td>Intercepts:
Estimate Std.Err z-value P(>|z|)
.x1 (.25.) 5.012 0.090 55.461 0.000
.x2 (.26.) 6.133 0.077 79.814 0.000
.x3 (.27.) 2.314 0.083 28.037 0.000
.x4 (.28.) 2.784 0.086 32.193 0.000
.x5 (.29.) 4.029 0.096 41.812 0.000
.x6 (.30.) 1.927 0.081 23.747 0.000
.x7 (.31.) 4.271 0.073 58.428 0.000
.x8 (.32.) 5.622 0.072 78.502 0.000
.x9 (.33.) 5.461 0.070 78.438 0.000</td></tr>
</table>

Figure 5. Path Diagram – Configural Invariance Model

```
semPaths(CFA.Configural, what="est",
         color="yellow", edge.color="blue",
         sizeLat = 8, sizeMan = 8, edge.label.cex = 1.25)
```



For a detailed output, the blavInspect function from the package blavaan extracts information from a fitted blavaan object to generate the posterior mean estimates, SE of mean, 2.5%, 25%, 50%, 75%, and 97.5% of the posterior estimates. The syntaxes are printed below.

```
Conf.Per  <-
blavInspect(CFA.Configural.B)
Metric.Per <- blavInspect(CFA.Metric.B)
Scalar.Per <- blavInspect(CFA.Scalar.B)
Strict.Per <- blavInspect(CFA.Strict.B)
Conf.Per
Metric.Per
Scalar.Per
Strict.Per
```

Figure 6. Path Diagram – Strict Invariance Model

```
semPaths(CFA.Strict, what="est",
    color="pink", edge.color="blue",
    sizeLat = 8, sizeMan = 8, edge.label.cex = 1.25)
```

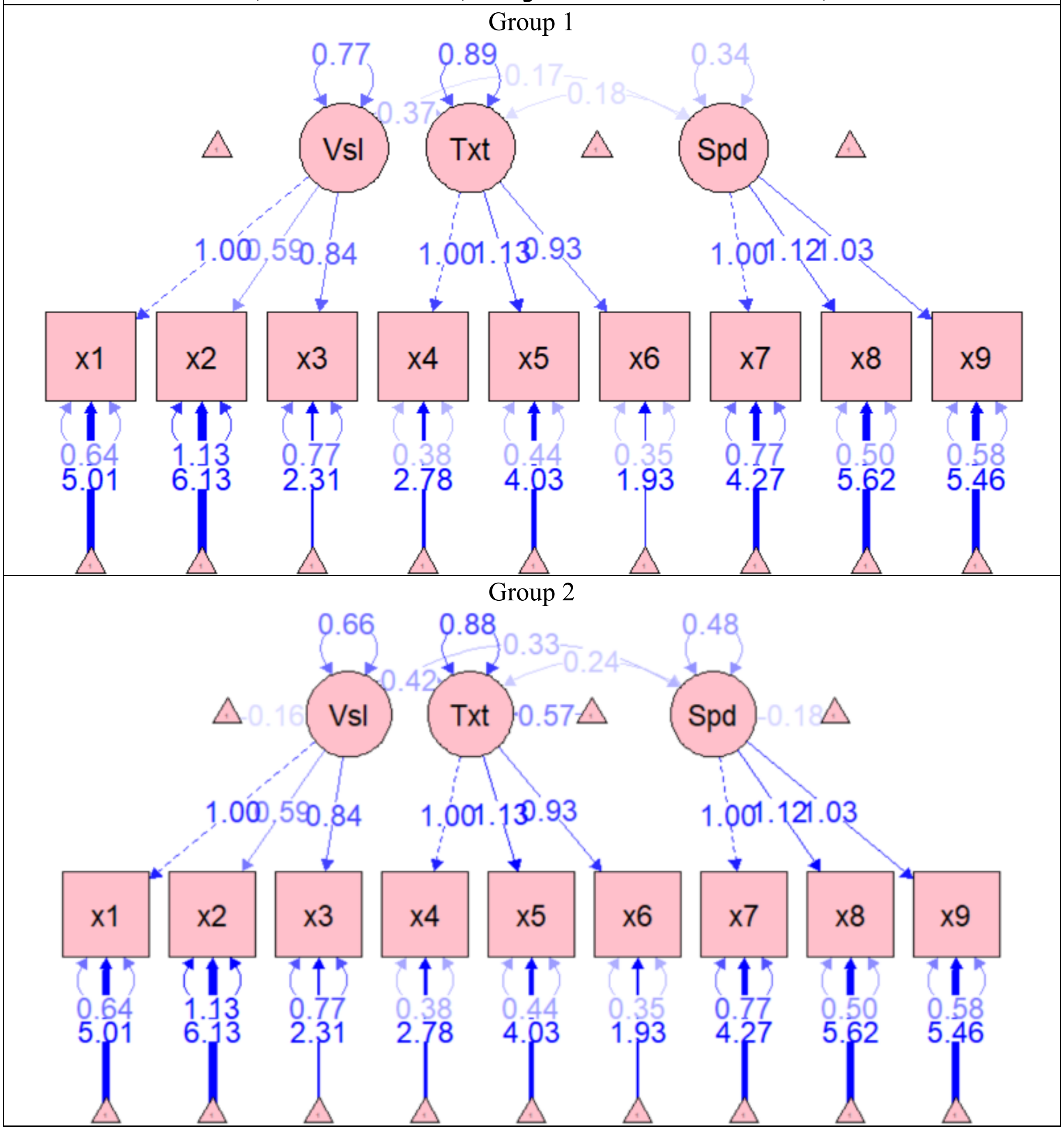


Figure 7 prints the Bayesian CFA output of the strict invariance model. Similar to the frequentist output, the factor loadings, intercepts, and variances are set to equality across the two groups under the strict invariance model. Under the `Estimate` column, the posterior average is printed. The `Post.SD` column prints the posterior standard deviation, `pi.lower` and `pi.upper` columns represent 2.5% and 97.5% of the posterior percentile density interval values respectively, `Rhat` stands for the potential scale reduction factor for assessing chain convergence which values that near 1.00 indicates convergence, Prior column specifies the prior specification. For instance, the posterior factor loading of x2 is 0.869, the posterior SD is 0.159, and the 2.5% and 97.5% density values are 0.385 and 0.851 respectively showing 95% of the range of the factor loadings. The Rhat of 1 indicates convergence and the prior is specified as N(0,10). The output also prints `MargLogLik` which represents the Laplace approximation of the marginal log-likelihood, and PPP represents the posterior predictive p-value. The value of PPP 0.000 indicates a poor fit of the model.

Figure 7. Bayesian Output – Strict Invariance

Group 1 - Pasteur

```
Latent Variables:
                   Estimate  Post.SD pi.lower pi.upper     Rhat    Prior
  Visual =~
    x1                1.000
    x2      (.p2.)    0.607    0.119    0.385    0.851    1.000    normal(0,10)
    x3      (.p3.)    0.869    0.156    0.596    1.210    1.001    normal(0,10)
  Textual =~
    x4                1.000
    x5      (.p5.)    1.132    0.067    1.006    1.265    1.000    normal(0,10)
    x6      (.p6.)    0.937    0.058    0.830    1.056    1.001    normal(0,10)
  Speed =~
    x7                1.000
    x8      (.p8.)    1.145    0.160    0.876    1.492    1.001    normal(0,10)
    x9      (.p9.)    1.069    0.205    0.762    1.533    1.001    normal(0,10)
Intercepts:
                   Estimate  Post.SD pi.lower pi.upper     Rhat    Prior
   .x1      (.25.)    5.009    0.091    4.829    5.189    1.001    normal(0,32)
   .x2      (.26.)    6.133    0.079    5.983    6.284    1.000    normal(0,32)
   .x3      (.27.)    2.317    0.087    2.151    2.488    1.001    normal(0,32)
   .x4      (.28.)    2.785    0.087    2.612    2.953    1.001    normal(0,32)
   .x5      (.29.)    4.029    0.097    3.841    4.217    1.001    normal(0,32)
   .x6      (.30.)    1.929    0.081    1.765    2.087    1.001    normal(0,32)
   .x7      (.31.)    4.268    0.078    4.114    4.424    1.000    normal(0,32)
   .x8      (.32.)    5.621    0.075    5.478    5.771    1.001    normal(0,32)
   .x9      (.33.)    5.460    0.072    5.320    5.604    1.000    normal(0,32)
Variances:
                   Estimate  Post.SD pi.lower pi.upper     Rhat    Prior
   .x1      (.10.)    0.669    0.126    0.407    0.908    0.999 gamma(1,.5)[sd]
   .x2      (.11.)    1.147    0.106    0.959    1.367    0.999 gamma(1,.5)[sd]
   .x3      (.12.)    0.775    0.110    0.556    0.991    1.000 gamma(1,.5)[sd]
   .x4      (.13.)    0.393    0.048    0.302    0.495    1.000 gamma(1,.5)[sd]
   .x5      (.14.)    0.443    0.060    0.335    0.574    1.000 gamma(1,.5)[sd]
   .x6      (.15.)    0.361    0.045    0.276    0.452    1.000 gamma(1,.5)[sd]
   .x7      (.16.)    0.790    0.092    0.625    0.979    1.001 gamma(1,.5)[sd]
   .x8      (.17.)    0.517    0.086    0.353    0.688    0.999 gamma(1,.5)[sd]
   .x9      (.18.)    0.584    0.086    0.410    0.744    1.000 gamma(1,.5)[sd]
    Visual            0.764    0.182    0.454    1.156    1.000 gamma(1,.5)[sd]
    Textual           0.918    0.142    0.672    1.217    1.000 gamma(1,.5)[sd]
    Speed             0.341    0.093    0.180    0.544    1.000 gamma(1,.5)[sd]
```

Group 2 – Grant-White

```
Latent Variables:
                   Estimate  Post.SD pi.lower pi.upper     Rhat
  Visual =~
    x1                1.000
    x2      (.p2.)    0.607    0.119    0.385    0.851    1.000
    x3      (.p3.)    0.869    0.156    0.596    1.210    1.001
  Textual =~
    x4                1.000
    x5      (.p5.)    1.132    0.067    1.006    1.265    1.000
    x6      (.p6.)    0.937    0.058    0.830    1.056    1.001
  Speed =~
    x7                1.000
    x8      (.p8.)    1.145    0.160    0.876    1.492    1.001
    x9      (.p9.)    1.069    0.205    0.762    1.533    1.001
Intercepts:
                   Estimate  Post.SD pi.lower pi.upper     Rhat
   .x1      (.25.)    5.009    0.091    4.829    5.189    1.001
   .x2      (.26.)    6.133    0.079    5.983    6.284    1.000
   .x3      (.27.)    2.317    0.087    2.151    2.488    1.001
   .x4      (.28.)    2.785    0.087    2.612    2.953    1.001
   .x5      (.29.)    4.029    0.097    3.841    4.217    1.001
   .x6      (.30.)    1.929    0.081    1.765    2.087    1.001
   .x7      (.31.)    4.268    0.078    4.114    4.424    1.000
   .x8      (.32.)    5.621    0.075    5.478    5.771    1.001
   .x9      (.33.)    5.460    0.072    5.320    5.604    1.000
Variances:
                   Estimate  Post.SD pi.lower pi.upper     Rhat    Prior
   .x1      (.10.)    0.669    0.126    0.407    0.908    0.999
   .x2      (.11.)    1.147    0.106    0.959    1.367    0.999
   .x3      (.12.)    0.775    0.110    0.556    0.991    1.000
   .x4      (.13.)    0.393    0.048    0.302    0.495    1.000
   .x5      (.14.)    0.443    0.060    0.335    0.574    1.000
   .x6      (.15.)    0.361    0.045    0.276    0.452    1.000
   .x7      (.16.)    0.790    0.092    0.625    0.979    1.001
   .x8      (.17.)    0.517    0.086    0.353    0.688    0.999
   .x9      (.18.)    0.584    0.086    0.410    0.744    1.000
    Visual            0.648    0.174    0.357    1.051    1.000 gamma(1,.5)[sd]
    Textual           0.898    0.140    0.651    1.201    1.000 gamma(1,.5)[sd]
    Speed             0.477    0.123    0.260    0.741    1.002 gamma(1,.5)[sd]
```

```
> Strict.Per
Inference for Stan model: stanmarg.
3 chains, each with iter=1500; warmup=500; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=3000.

               mean se_mean   sd  2.5%   25%   50%   75% 97.5%
ly_sign[1]     0.61    0.00 0.12  0.39  0.52  0.60  0.68  0.85
ly_sign[2]     0.87    0.00 0.16  0.60  0.76  0.86  0.96  1.21
ly_sign[3]     1.13    0.00 0.07  1.01  1.09  1.13  1.18  1.27
ly_sign[4]     0.94    0.00 0.06  0.83  0.90  0.94  0.97  1.06
ly_sign[5]     1.15    0.00 0.16  0.88  1.04  1.13  1.24  1.49
ly_sign[6]     1.07    0.01 0.21  0.76  0.93  1.05  1.18  1.53
Theta_var[1]   0.67    0.00 0.13  0.41  0.59  0.67  0.75  0.91
Theta_var[2]   1.15    0.00 0.11  0.96  1.07  1.14  1.22  1.37
Theta_var[3]   0.77    0.00 0.11  0.56  0.70  0.77  0.85  0.99
Theta_var[4]   0.39    0.00 0.05  0.30  0.36  0.39  0.42  0.49
Theta_var[5]   0.44    0.00 0.06  0.34  0.40  0.44  0.48  0.57
Theta_var[6]   0.36    0.00 0.05  0.28  0.33  0.36  0.39  0.45
Theta_var[7]   0.79    0.00 0.09  0.63  0.72  0.79  0.85  0.98
Theta_var[8]   0.52    0.00 0.09  0.35  0.46  0.52  0.58  0.69
Theta_var[9]   0.58    0.00 0.09  0.41  0.53  0.58  0.64  0.74
Nu_free[1]     5.01    0.00 0.09  4.83  4.95  5.01  5.07  5.19
Nu_free[2]     6.13    0.00 0.08  5.98  6.08  6.13  6.19  6.28
Nu_free[3]     2.32    0.00 0.09  2.15  2.26  2.32  2.37  2.49
Nu_free[4]     2.79    0.00 0.09  2.61  2.73  2.78  2.84  2.95
Nu_free[5]     4.03    0.00 0.10  3.84  3.96  4.03  4.09  4.22
Nu_free[6]     1.93    0.00 0.08  1.77  1.87  1.93  1.98  2.09
Nu_free[7]     4.27    0.00 0.08  4.11  4.22  4.27  4.32  4.42
Nu_free[8]     5.62    0.00 0.07  5.48  5.57  5.62  5.67  5.77
Nu_free[9]     5.46    0.00 0.07  5.32  5.41  5.46  5.51  5.60

                         MargLogLik      PPP
                          -3864.962    0.000
```

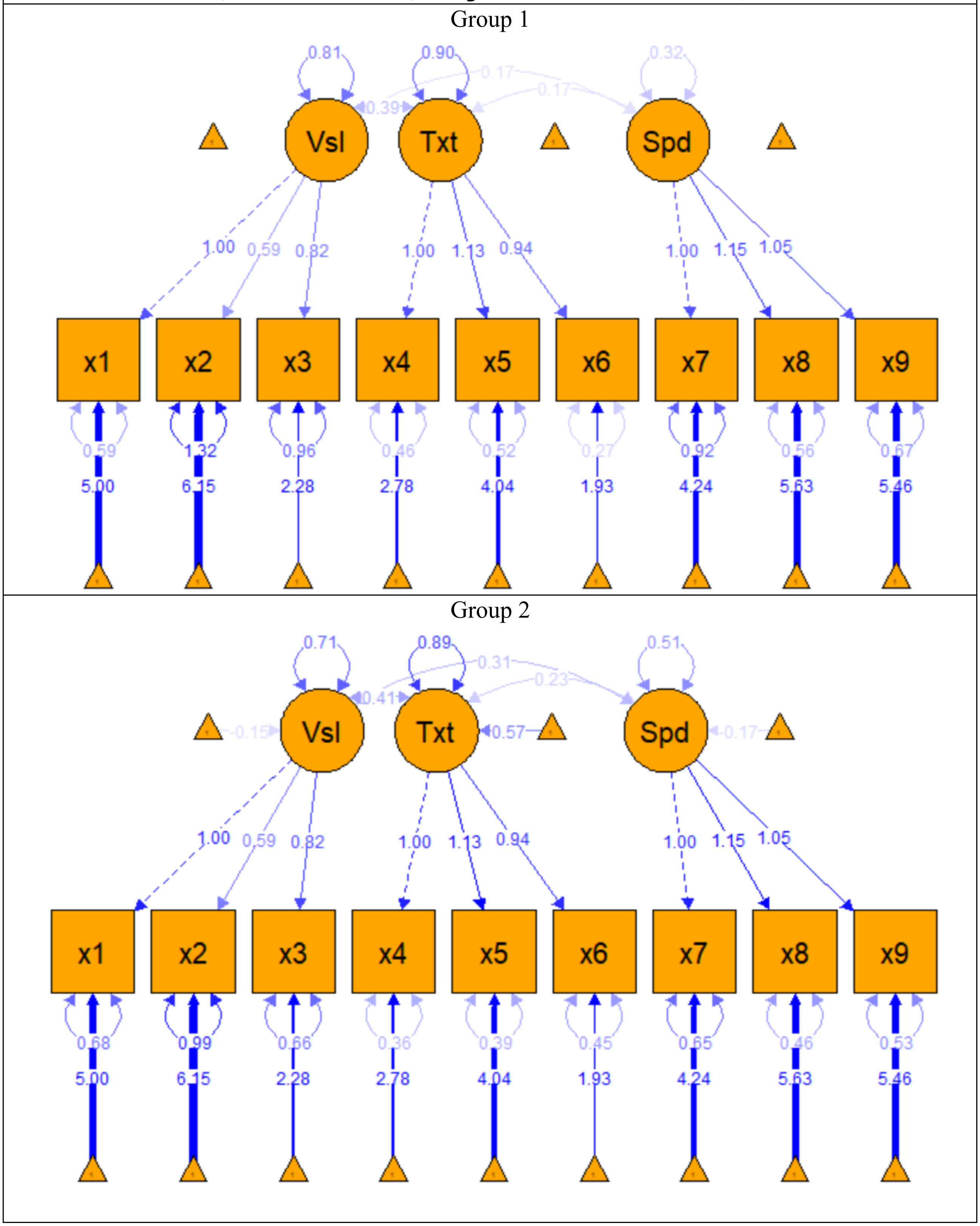
The object `Strict.Per` prints 8 posterior statistics and 3 sets of variables. The statistics include the posterior mean (`mean`), posterior mean se (`se_mean`), posterior sd (`sd`), 2.5%. 25%, 50%, 75%, and 97.5% of the posterior estimates. The variables whose names start with `ly_sign` are the posterior factor loadings, `Nu_free` are the posterior intercepts, and `Thera_var` are the posterior variances. The first six rows print the factor loadings of x2, x3, x5, x6, x8, and x9 under the names initial `ly_sign` followed by a bracket [ ] that within it starts from 1 to 6. Since the factor loadings of x1, x4, and x7 are by default set to one, only 6 factor loadings are printed. For instance, the mean factor loading of x2 (`ly_sign[1]`) is 0.61, with sd 0.12, factor loading estimates range from 0.39 at 2.5% to 0.85 at 97.5%. An examination of these estimated posterior values gives a sense of the spread of these parameters. For instance, a comparison of the posterior loadings of x2 and x6 gives the contrast of their loading distributions. X2 has a wider spread for the percentile posterior loadings ranging from 0.39 (2.5%) to 0.85 (97.5%) while the narrower spread of x6 ranges from 0.83 (2.5%) to 1.06 (97.5%). The posterior intercepts and variances have the same interpretation. For instance, the mean value of the posterior intercept of x1 (`Nu-free[1]`) is 5.01 with estimates ranging from 4.83 at 2.5% to 5.19 at 97.5%, the mean value of the posterior variance of x2 (`Theta_var[2]`) is 1.15 with estimates ranging from 0.11 at 2.5% to 1.37 at 97.5%.

` Similar to producing the frequentist CFA model graphically, function semPaths could also be used to graph the path diagram of the Bayesian CFA model. The Bayesian scalar CFA invariance is printed in Figure 8 with the syntaxes given on top of the path diagrams.

Varying Priors Settings of Factor Loadings – An Example. Setting priors is one crucial procedure for carrying out Bayesian factorial invariance analysis as specifying prior information can lead to quite different estimation outcomes (Van de Schoot et al., 2014). The default priors settings in blavaan (Merkle & Rosseel,

Figure 8. Path Diagram – Bayesian Scalar Invariance Model

```
semPaths(CFA.Scalar.B, what="est",
   color="orange", edge.color="blue",
   sizeLat = 8, sizeMan = 8, edge.label.cex = 1.25)
```



2018) for a Bayesian model places $N(0, 10)$ as priors for factor loadings & indicators; N(0, 32) for the intercepts; and gamma(1, 0.5) for indicator residual standard deviations. These default prior settings can be printed using the dpriors function as shown below. The abbreviations of the function dpriors are listed in Appendix C, Table C1.

```
> dpriors()
                nu                alpha               lambda                 beta                theta
   "normal(0,32)"       "normal(0,10)"       "normal(0,10)"       "normal(0,10)" "gamma(1,.5)[sd]"
               psi                  rho                ibpsi                  tau
"gamma(1,.5)[sd]"          "beta(1,1)"  "wishart(3,iden)"      "normal(0,1.5)"
```

An example is illustrated by changing the prior setting of factor loadings from the default specified in package blavaan to examine whether the factorial invariance model is misspecified. Muthén and Asparouhov (2012) suggested estimating all possible cross-loadings using priors by relaxing their estimates serves as a means as if it is another way to carry out "model modification" via the Bayesian approach. While the main aim of model modification is to improve model fit if there are parameters omitted by adding them (MacCallum et al., 1992), creating all possible cross-loadings using Bayesian confirmatory factor analysis with the specification of small variance priors with mean zero is another way to examine whether there are possible significant and high cross-loadings that differ from the theoretical specification (Jorgensen et al., 2019). For instance, while the indicators x4 to x9 do not expect to load to the visual construct, by specifying restrictive low values for these factor loadings with the specification of $N(0, \sigma = 0.06)$, the estimated Bayesian loadings serve as a check for possible misspecification if any of the indicators x4 to x9 are heavily loaded to the construct visual. The same argument goes for the textual and speed constructs. The syntax of specifying all possible cross-factor loadings to the visual, textual, and speed constructs is stated below. The prior function states the priors to use are from a normal distribution with zero mean and a low standard deviation of 0.06. Figure 9 prints the output of the specified model.

```
HS.Model.CL <-' visual   =~ x1 + x2 + x3
                textual =~ x4 + x5 + x6
                speed   =~ x7 + x8 + x9
  # Specify Prior Cross-loadings
visual  =~ prior("normal(0,.06)")*x4 +
           prior("normal(0,.06)")*x5 +
           prior("normal(0,.06)")*x6 +
           prior("normal(0,.06)")*x7 +
           prior("normal(0,.06)")*x8 +
           prior("normal(0,.06)")*x9
textual =~ prior("normal(0,.06)")*x1 +
           prior("normal(0,.06)")*x2 +
           prior("normal(0,.06)")*x3 +
           prior("normal(0,.06)")*x7 +
           prior("normal(0,.06)")*x8 +
           prior("normal(0,.06)")*x9
speed   =~ prior("normal(0,.06)")*x1 +
           prior("normal(0,.06)")*x2 +
           prior("normal(0,.06)")*x3 +
           prior("normal(0,.06)")*x4 +
           prior("normal(0,.06)")*x5 +
           prior("normal(0,.06)")*x6'
CFA.Configural.B.CL <- bcfa(HS.Model.CL, data=H,
                            group = "school")
summary(CFA.Configural.B.CL)
```

Figure 9. Bayesian Output – Metric Invariance with All Cross Loadings

MargLogLik	PPP
-3906.086	0.037

Group 1 – Pasteur

	Estimate	Post.SD	pi.lower	pi.upper	Rhat	Prior
visual =~						
x4	0.034	0.051	-0.065	0.136	1.000	normal(0,.06)
x5	-0.058	0.055	-0.166	0.047	0.999	normal(0,.06)
x6	0.062	0.050	-0.039	0.157	1.000	normal(0,.06)
x7	-0.046	0.056	-0.156	0.065	1.001	normal(0,.06)
x8	-0.012	0.055	-0.120	0.095	0.999	normal(0,.06)
x9	0.078	0.055	-0.033	0.187	1.000	normal(0,.06)
textual =~						
x1	0.054	0.060	-0.063	0.169	1.000	normal(0,.06)
x2	-0.013	0.056	-0.125	0.100	1.001	normal(0,.06)
x3	-0.047	0.054	-0.149	0.058	1.000	normal(0,.06)
x7	0.031	0.053	-0.074	0.135	1.000	normal(0,.06)
x8	-0.014	0.056	-0.124	0.095	0.999	normal(0,.06)
x9	0.005	0.052	-0.095	0.105	1.000	normal(0,.06)
speed =~						
x1	0.002	0.059	-0.110	0.118	1.000	normal(0,.06)
x2	-0.011	0.058	-0.125	0.100	0.999	normal(0,.06)
x3	0.009	0.057	-0.103	0.124	1.000	normal(0,.06)
x4	0.007	0.057	-0.104	0.118	0.999	normal(0,.06)
x5	-0.026	0.057	-0.133	0.087	1.000	normal(0,.06)
x6	0.034	0.053	-0.067	0.141	1.000	normal(0,.06)

Group 2 – Grant-White

	Estimate	Post.SD	pi.lower	pi.upper	Rhat	Prior
visual =~						
x4	0.009	0.054	-0.096	0.117	1.000	normal(0,.06)
x5	-0.014	0.056	-0.123	0.095	0.999	normal(0,.06)
x6	0.009	0.056	-0.099	0.119	1.000	normal(0,.06)
x7	-0.054	0.057	-0.168	0.055	0.999	normal(0,.06)
x8	-0.011	0.056	-0.122	0.098	1.000	normal(0,.06)
x9	0.087	0.059	-0.033	0.201	1.000	normal(0,.06)
textual =~						
x1	0.015	0.058	-0.100	0.126	1.000	normal(0,.06)
x2	-0.015	0.057	-0.125	0.097	0.999	normal(0,.06)
x3	0.006	0.055	-0.105	0.111	1.000	normal(0,.06)
x7	0.027	0.051	-0.074	0.127	1.000	normal(0,.06)
x8	-0.062	0.052	-0.162	0.040	1.000	normal(0,.06)
x9	0.070	0.053	-0.036	0.172	1.001	normal(0,.06)
speed =~						
x1	0.028	0.057	-0.087	0.140	1.000	normal(0,.06)
x2	-0.009	0.056	-0.119	0.101	1.000	normal(0,.06)
x3	-0.009	0.056	-0.119	0.104	0.999	normal(0,.06)
x4	-0.020	0.053	-0.124	0.084	1.000	normal(0,.06)
x5	0.037	0.055	-0.070	0.146	0.999	normal(0,.06)
x6	-0.015	0.052	-0.118	0.085	0.999	normal(0,.06)

Compared to the default model without the cross-loadings with a PPP value of 0.000, the HS.model.CL model for the cross-loading model with a PPP value of 0.036 (Figure 9) indicating it is a better-fit model. However, this is an overfitted model due to the inclusion of 18 nuisance cross-loading parameters which are purposely inserted to exhibit their triviality. As such, the fit should not lead to the conclusion of using the cross-loading model. The outcomes indicate the estimates of these cross-loadings are relatively small, the values of pi.lower are all negative, and the values of pi.upper are all positive. These outcomes demonstrate the irrelevancy of these loadings that cross the zero estimates from negative to positive, signifying their inappropriateness of including these loadings. In short, this example shows the benefit of using the Bayesian CFA for examining the factor loading with the specification of priors for exploring model misspecification.

Evaluation of Factorial Invariance Results – Model Fit and Comparison

Fit Indices. The factorial invariance models specify the level of factorial invariance that could be attained. Examining at which level that could have been achieved, model fit indices and comparison methods are often used for evaluation. This section recommends five frequentist fit indices and their corresponding Bayesian fit indices together with the more recently developed Bayesian fit indices. The frequentist fit indices include the Root Mean Square Error of Approximation (RMSEA; Steiger and Lind, 1980), Comparative Fit Index (CFI; Bentler, 1990), Tuker-Lewis Index (TLI; Bentler & Bonett, 1980; Tuker & Lewis, 1973), Akaike Information Criteria (AIC; Akaike, 1974), and Bayesian Information Criteria (BIC; Schwarz, 1978). The equivalent in Bayesian indices include Bayesian RMSEA (BRMSEA; Hoofs et al., 2018; Garnier-Villarreal & Jorgensen, 2020), Bayesian TLI (BTLI), and Bayesian CFI (BCFI), and Deviance Information Criterion (DIC; Spiegelhalter, Best, Carlin, & Linde, 2014; Spiegelhalter, Best, Carlin, & Van Der Linde, 2002) which are conceptually related to their frequentist counterparts. The fit indices from the Bayesian framework include the Widely Applicable Information Criterion (WAIC, Watanabe, 2010), Leave-one-out cross-validation (LOO, Gelfand, 1996), and the prior posterior predictive p-value (PPP; Gelman, Carlin, Stern, & Rubin, 2014). For examining between two CFA factorial models, the frequentist approaches include five comparative fit indices (ΔCI), ΔCFI, namely the ΔRMSEA, ΔCFI, ΔTLI, ΔAIC, and ΔBIC.

Practical Guidelines. To evaluate the fit of a CFA model, the literature generally recommended using a cutoff to provide a dichotomous indicator of model fit or a series of cutoffs to signal the acceptable level of model fit. Common interpretations of cutoffs for RMSEA are less than 0.05 for a good fit, 0.05 indicates a close fit, more than 0.10 indicates a poor fit, and various proposal cutoff intervals not far away from this (e.g. Browne & Cudeck, 1993; Hu & Bentler, 1999; MacCallum, Browne, & Sugawara, 1996). For the CFI and TLI, values between .95 and .97 suggest a good fit and a value above .97 suggests an excellent fit.

For changes in comparative fit indices (ΔCI), Chen (2007) suggested that the value of ΔCFI equal to or greater than −.010 supplemented by RMSEA less than or equal to .05 are indicative of non-invariance when sample sizes are equal across groups and larger than 300 in each group. Cheung & Rensvold (2002) suggested ΔCFI ≤ .01 of non-invariance across models. Meade et al., (2008) suggested a smaller ΔCFI ≤ .002 is more appropriate for assessing invariance.

Similar to the principle of providing the cutoff guideline under the frequentist approach, Bayesian cutoffs also aim to provide the outcomes of model fit. The simulation study of Hoofs et al., (2018) recommended the guideline for BRMSEA with cutoff values for the lower and upper limits <0.05 and <0.08 respectively. A low PPP (< 0.05) indicates poor model fit and PPP values of around 0.50 indicate very good fit (Muthén & Asparouhov, 2012).

Evaluation of Factorial Invariance using R Packages. Function compareFit from the package semTools (Epskamp, 2022) provides model fit indices and comparative fit indices. Specifying the name of the four factorial-invariance CFA models in this function produces the model fit of the models. The summary function with the specification of the fit.measures argument states the fit indices to be outputted. The syntax below specifies ten fit statistics to be generated.

```
CompareAll <- compareFit(CFA.Configural,
                         CFA.Metric,
                         CFA.Scalar,
                         CFA.Strict)
summary(CompareAll,
  fit.measures = c("chisq","df","pvalue",
                   "rmsea","rmsea.ci.lower","rmsea.ci.upper",
                   "cfi","tli","aic","bic")
```

Figure 10 prints the outputs of the model fit indices and the difference in fit indices separately into two portions. In the first section, the symbol "†" inserted after the values of the fit index indicates the best CFA factorial invariance model. The lowest values of RMSEA (0.097), & AIC (7480.587), and the highest value of TLI (0.895) indicate the best model is the metric CFA model. The configural CFA is suggested as the best model with the highest CFI (0.923) whereas strict CFA is indicated by the results of the lowest BIC value (7652.632). According to Cheung & Rensvold's (2002) criterion of ΔCFI value of .01 criterion, the configural model is suggested as the ΔCI for both CFI and TLI met this criterion. Overall, counting the number of fit statistics, the metric model is the preferred choice.

Figure 10. Model Comparison and Differences in Fit Indices - Frequentist

```
###################### Model Fit Indices ###########################
                  chisq df pvalue rmsea rmsea.ci.lower rmsea.ci.upper  cfi   tli      aic       bic
CFA.Configural 115.851† 48    .000 .097           .075           .120 .923† .885  7484.395  7706.822
CFA.Metric     124.044  54    .000 .093†          .071†          .114† .921 .895† 7480.587† 7680.771
CFA.Scalar     164.103  60    .000 .107           .088           .127 .882  .859  7508.647  7686.588
CFA.Strict     181.511  69    .000 .104           .086           .123 .873  .867  7508.055  7652.632†

################# Differences in Fit Indices ######################
                          df  rmsea rmsea.ci.lower rmsea.ci.upper    cfi    tli    aic     bic
CFA.Metric - CFA.Configural  6 -0.004        -0.003         -0.005 -0.002  0.009 -3.808 -26.050
CFA.Scalar - CFA.Metric      6  0.015         0.017          0.013 -0.038 -0.036 28.059   5.817
CFA.Strict - CFA.Scalar      9 -0.003        -0.002         -0.005 -0.009  0.008 -0.591 -33.955
```

While the function compareFit provides the results with the specification of the CFA invariance models, it is worth mentioning a more recent package equaltestMI (Jiang & Mai, 2021) provides functions for carrying out measurement invariance with a comprehensive 11 CFA models without specifying the CFA models (see Appendix B Table B1). This function is useful for readers already familiar with the process of FI to produce a thorough list.

Similar to the frequentist approach, the function fitMeasures from the package lavaan provides four Bayesian fit indices PPP, DIC, WAIC, and LOOIC. The syntax is similar to that of the frequentist in that the name of the model has to be specified, followed by the list of the names of the fit indices. The results for these four factorial invariance models are tabulated in Table 2. All the PPP values are less than 0.05 showing all the fits are poor which is in line with the results of the frequentist fit value such as the value of TLI for the metric model is 0.895, less than the threshold of 0.95. The DIC shows the configural model fits best while WAIC and LOOIC prefer the metric model with the lowest values. Similar to the frequentist, the Bayesian counterpart also suggests the metric invariance model.

```
fitmeasures(CFA.Configural.B,c("ppp","dic","waic","looic"))
fitmeasures(CFA.Metric.B,c("ppp","dic","waic","looic"))
fitmeasures(CFA.Scalar.B,c("ppp","dic","waic","looic"))
fitmeasures(CFA.Strict.B,c("ppp","dic","waic","looic"))
```

Table 2. Bayesian Fit Statistics

Factorial Invariance	PPP	DIC	WAIC	LOOIC
Configural	0.00000	7020	7540	7538
Metric	0.00033	7481	7488	7489
Scalar	0.00000	7509	7516	7516
Strict	0.00000	7507	7512	7513

The posterior frequentist fit indices are often generated and analyzed under the Bayesian framework as this is the core benefit of using this approach in that the parameter is a distribution, not a fixed estimate. The function blavFitIndices from the package blavaan provides this facility to store the fit indices into an R object. The first argument of this function states the names of the fitted Bayesian model. The second rescale argument states the option MCMC to use the Markov chain based on the model-implied moments to output the posterior values of the five fit indices, RMSEA, CFI, TLI, AIC, and BIC.

```
set.seed(1234567)
Conf.B.FitIndex <- blavFitIndices(CFA.Configural.B,
    Rescale      = "MCMC",
fit.measures = c("rmsea","cfi","tli","aic","bic"))
Metric.B.FitIndex <- blavFitIndices(CFA.Metric.B,
    Rescale      = "MCMC",
    fit.measures = c("rmsea","cfi","tli","aic","bic"))
Scalar.B.FitIndex   <- blavFitIndices(CFA.Scalar.B,
    rescale      = "MCMC",
    fit.measures = c("rmsea","cfi","tli","aic","bic"))
Strict.B.FitIndex   <- blavFitIndices(CFA.Strict.B,
    rescale      = "MCMC",
    fit.measures = c("rmsea","cfi","tli","aic","bic"))
```

Figure 11 plots the posterior distributions of the four sets of Bayesian factorial invariance models for the five fit indices: CFI, TLI, RMSEA, AIC, and BIC. As each of the posterior parameters contains 3,000 observations, it is useful to examine these posterior distributions using a histogram. The mean, median, and standard deviation of these five statistics are tabulated in Table 3.

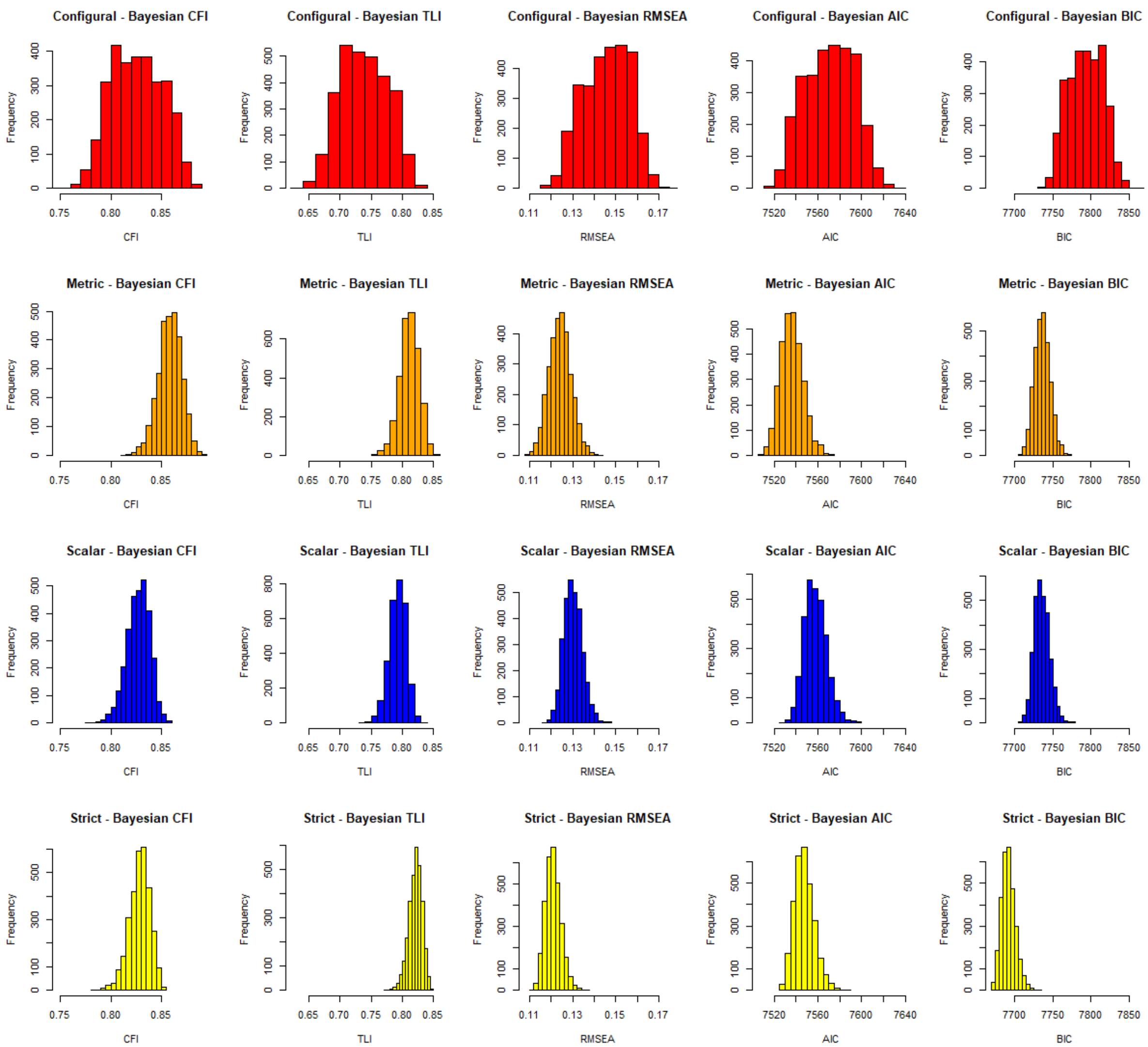
The histograms indicate the characteristics of the fit indices. The distribution of CFI is positioned at a higher value range with a narrow distribution for the metric invariance model (Figure 11), indicating this model is a better fit, in comparison to other invariance models. A low standard deviation with the highest mean and median values (Table 3) also shows similar findings. In contrast, the same set of indicators for TLI, RMSEA, and BIC show in favor of the strict invariance model while the results of AIC and BIC do not indicate a clear solution.

Summary and Conclusion

The procedure of carrying out factorial invariance using multigroup confirmatory factor analysis is systematically described in the current paper by first introducing the concepts of factorial invariance, presenting the commonly adopted four steps of the factorial procedure (configural, weak, strong, and strict invariance), recapitulating the various frequentist and Bayesian fit indices and working on a classical example using more recently developed R packages and functions to show the steps and interpreting the results.

While the frequentist approach was intensively discussed in the factorial invariance literature and there were many reviews and practical applications, a short discussion is useful for researchers who intend to proceed with the Bayesian factorial invariance framework. The Bayesian framework generally complements the frequentist approach in validation and reporting, shifting from a point estimate to a posterior distribution. The main advantage of using this framework is that it provides an alternative range of informative tests of the CFA invariance model. Although the application of Bayesian CFA is on the rise, and many issues still warrant further research, noting the usage needs careful attention. For instance, while DIC is a generalization of AIC being it tries to find the simplest model that fits the data well

Figure 11. Posterior Distribution: CFI, TLI, RMSEA, AIC, and BIC



(Plummer, 2006), and the simulation on BRMSEA fit indices by Hoofs et al., (2018) provide the practical guidelines, more research still needed to be carried out on other Bayesian fit indices to provide more evidence to stipulate practical and useful guidelines.

Since the prior setting is one crucial factor for Bayesian CFA, and an example is demonstrated to show how to specify the syntax and demonstrate its application for examining misspecification, a short discussion on the prior specification for carrying out Bayesian factorial invariance is helpful to readers not familiar with the consequences of prior settings. While the current paper uses the default prior setting of function bcfa to carry out Bayesian factorial invariance, researchers can specify prior information (e.g., Van de Schoot et al., 2014) to produce expected estimations. In frequentist CFA, the correlation matrix Ω needs to be a sparse matrix to ensure identifiability. While it is typically difficult to foresee the correlation of the residuals, Bayesian CFA provides a neat solution to this problem by specifying a very large value of η using the LKJ prior (Lewandowski et al., 2009) on Ω, forces all residual correlations to be low, the default specifications of package lavaan. However, specifying the prior should be carried out carefully when the data set is small. The main reason is that the prior of intercepts assumes univariate normal priors and prior factor loading could be considered too informative when a given item does not load on a given factor leading to assigning a small standard deviation prior.

Table 3. Mean, and Median – RMSEA, CFI, TLI, AIC, and BIC: Bayesian

Factorial Invariance	Mean				
	CFI	TLI	RMSEA	AIC	BIC
Configural	0.8256	0.7384	0.1458	7571	7793
Metric	0.8583	0.8111	0.1242	7536	7736
Scalar	0.8271	0.7925	0.1302	7558	7736
Strict	0.8284	0.8209	0.1209	7548	7692
Factorial Invariance	Median				
	CFI	TLI	RMSEA	AIC	BIC
Configural	0.8248	0.7371	0.1466	7572	7794
Metric	0.8587	0.8115	0.1241	7536	7736
Scalar	0.8280	0.7936	0.1299	7557	7735
Strict	0.8292	0.8217	0.1207	7547	7691
Factorial Invariance	Standard Deviation				
	CFI	TLI	RMSEA	AIC	BIC
Configural	0.0251	0.0376	0.0106	22.20	22.20
Metric	0.0116	0.0155	0.0051	10.29	10.29
Scalar	0.0113	0.0136	0.0042	10.03	10.03
Strict	0.0101	0.0105	0.0035	8.95	8.95

When the prior knowledge is available and known, informative priors can be chosen, however, it can also turn to biased results in certain directions. Thus, all informative priors need to have a clear justification. The recommendation for reporting Bayesian invariance results is the disclosure of the specification of priors when using the informative priors and the reason and justification are clearly stated such that readers read the results with the consideration of the informed prior information. For instance, stating the reason for restricting the use of uniform priors when the parameter is restricted, using weak priors with normal distribution specified for exploratory factorial invariance and diagnosing purposes, and using a student's t distribution to replace normal distribution to solve fat tails distribution. In summary, the cost of uninformative priors is that you are putting too much weight on your actual data; the cost of too strong a prior is that you are letting assumptions rather than data do most of the work. The researcher may have to decide how to take a balance.

In conclusion, the advantages of adopting the Bayesian framework in factorial invariance are that it offers a flexible approach that allows the incorporation of prior knowledge for the estimation of under-identified models, that it is a natural means of constraining parameters, and better for small-sample performance (Scheines, Hoijtink, & Boomsma, 1999). Another main benefit lies in the Bayesian factorial invariance provides full posterior distributions for each parameter and latent variable so that researchers can learn about the model as a whole. Not forgetting that there is also the benefit of the frequentist of the ML point estimate, the current paper offers a start on the journey to Bayesian factorial invariance, providing the syntax of both the frequentist and Bayesian approaches and serves as a reference towards this journey.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions Automatic Control, 19(6)*, 716-723.

An, M., Kusurkar, R. A., Xiao, Y., Zhrng, C., Hu, J. & Chen, M. (2017). Measurement invariance of the strength of motivation for medical school: A multigroup confirmatory factor analysis. *BMC Medical Education, 17,* 116. doi:10.1186/s12909-017-0958-4.

Bagheri, Z., Chamanpara, P., Jafari, P., Balhara, Y. P. S., Arya, S., Ransing, R., Doric, A., Knez, R., et al. (2022). Cross-cultural measurement invariance of the quality of life enjoyment and satisfaction questionnaire-short form across ten countries: The application of Bayesian approximate measurement invariance. *BMC Psychology, 10(160),* 1-10. https://doi.org/10.1186/s40359-022-00864-y

Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin, 107*, 238–246.

Bentler, P. M., & Bonett, D. G. (1980). Significance tests and goodness of fit in the analysis of covariance structures. *Psychological Bulletin, 88*, 588–606.

Bollen, K. A. (1989). *Structural Equations with Latent Variables*. John Wiley and Sons, Inc., New York. Wiley. doi:10.1002/9781118619179

Cain, M. K. & Zhang, Z. (2019). Fit for a Bayesian: An evaluation of PPP and DIC for structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal, 26*, 39–50. https://doi.org/10.1080/10705511.2018.1490648.

Chen, F. F. (2007). Sensitivity of goodness of fit indexes to lack of measurement invariance. *Structural Equation Modeling: A Multidisciplinary Journal*, *14*(3), 464–504. https://doi.org/10.1080/10705510701301834

Cheung, G. W., & Rensvold, R. B. (2002). Evaluating goodness-of-fit indexes for testing measurement invariance. *Structural Equation Modeling, 9*, 233-255.

Cheung, G. W., & Rensvold, R. B. (2002). Evaluating goodness-of-fit indexes for testing measurement invariance. *Structural Equation Modeling, 9*, 233-255.

Deng, L., & Yuan, K. H. (2016). Comparing latent means without mean structure models: A projection-based approach. *Psychometrika, 81(3),* 802-829. https://doi.org/10.1007/s11336-015-9491-8.

Dimitrov, D. M. (2010). Testing for invariance in the context of construct validation. *Measurement and Evaluation in Counseling and Development, 43*, 121. doi:10.1177/0748175610373459.

Epskamp, S. (2022). semPlot: Path diagrams and visual analysis of various SEM Packages' output. R package version 1.1.5. https://CRAN.R-project.org/package=semPlot.

Feldt, T., Rantanen, J., Hyvonen, K., Makikangas, A., Huhtala, M., Pihlajasaari, P., & Kinnunen, U. (2014). The 9-item Bergen Burnout Inventory: Factorial validity across organizations and measurements of longitudinal data. *Industrial Health, 52*, 102-11

Garnier-Villarreal, M., & Jorgensen, T. D. (2020). Adapting fit indices for Bayesian structural equation modeling: Comparison to maximum likelihood. *Psychological Methods, 25 (1),* 46-70. https://doi.org/10.1037/met0000224.

Gelfand, A. E. (1996). Model determination using sampling-based methods. In W. R. Gilks, S. Richardson, D. J. Spiegelhalter (eds.), *Markov Chain Monte Carlo in Practice*, pp. 145-162. London: Chapman & Hall.

Gelman, A., Meng, X.-L., & Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica, 6(4),* 733–760.

Golino, H., & Christensen, A. P. (2022). *EGAnet: Exploratory Graph Analysis – A framework for estimating the number of dimensions in multivariate data using network psychometrics*. R package version 1.1.1.

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research, 2017, *PLOS ONE, 12(6)*, e0174035.

Ghosh, A., Niilekselaa, C. R., Parham, A. & Janis, R. A. (2021). Investigating factorial invariance of the counseling center Assessment of Psychological Symptoms 34 CCAPS 34 with military and nonmilitary students. *Measurement and Evaluation in Counseling and Development, 54(1),* 42–55. https://doi.org/10.1080/07481756.2020.1745649.

Holzinger, K., & Swineford, F. (1939). A study in factor analysis: The stability of a bifactor solution. *Supplementary Educational Monograph, no. 48*. Chicago: University of Chicago Press.

Hoofs, H. van de Schoot, R., Jansen, N. W. H., & Kant, I. (2018). Evaluating model fit in Bayesian confirmatory factor analysis with large samples: Simulation study introducing the BRMSEA. *Educational and Psychological Measurement, 78(4)*, 537–568. doi:10.1177/0013164417709314.

Holzinger, K., and Swineford, F. (1939). A study in factor analysis: The stability of a bifactor solution.

Supplementary Educational Monograph, no. 48. Chicago: University of Chicago Press.

Horn, J. L., & McArdle, J. J. (1992). A practical and theoretical guide to measurement invariance in aging research. *Experimental Aging Research, 18*, 117–144.

Huansuriya, T., Kruahiran, P., Sriutaisuk, S, & Musa, R. (2020). Construct validity and measurement invariance of the Asian family characteristics scale in the Thai population. *Journal of Health Research, 36(4)*, 615-628. doi:10.1108/JHR-06-2020-0219.

Jiang, G., Mai, Y., & Yuan, K. H. (2017). Advances in Measurement Invariance and Mean Comparison of Latent Variables: Equivalence Testing and A Projection-Based Approach. *Frontiers in Psychology, 8*, 1823.

Jiang, G., & Mai, Y. (2021). equaltestMI: Examine measurement invariance via equivalence testing and projection method. R package version 0.6.1, https://CRAN.Rproject.org/package=equaltestMI.

Jorgensen, T. D., Garnier-Villarreal, M., Pornprasertmanit, S. & Lee, D. (2019). Small-variance priors can prevent detecting important misspecifications in Bayesian confirmatory factor analysis. In *Quantitative Psychology: The 83rd Annual Meeting of the Psychometric Society, New York, NY, 2018*, edited by Wiberg, M., Culpepper, S., Janssen, R., González, J., and Molenaar, D., 265:255–63. Springer Proceedings in Mathematics & Statistics. New York, NY, US: Springer. https://doi.org/10.1007/978-3-030-01310-3_23.

Jorgensen, T. D., Pornprasertmanit, S., Schoemann, A. M., & Rosseel, Y. (2022). semTools: Useful tools for structural equation modeling. R package version 0.5-6. Retrieved from https://CRAN.R-project.org/package=semTools

Kaplan, D., & Depaoli, S. (2012). Bayesian structural equation modeling. In R. Hoyle (Ed.), *Handbook of structural equation modeling (pp. 650-673)*. New York, NY: Guilford Press

Lau, W. W. F. & Yuen, A., H. K. (2015). Factorial invariance across gender of a perceived ICT literacy scale. *Learning and Individual Differences, 41*, 79-85. http://dx.doi.org/10.1016/j.lindif.2015.06.001.

Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis, 100*, 1989–2001. doi:10.1016/j.jmva.2009.04.008

Liau, A. K., Tan, T. K., Li, D., & Khoo, A. (2012). Factorial invariance of the Personal Strengths Inventory-2 for children and adolescents across school level and gender. *European Journal of Psychology of Education*, *27*, 451-465. doi:10.1007/s10212-011-0088-z.

Kline, R. B. (2015). *Principles and Practice of Structural Equation Modeling*, Fourth Edition. Guilford Publications.

Liang, X. & Luo, Y. (2020). A comprehensive comparison of model selection methods for Testing Factorial Invariance. *Structural Equation Modeling: A Multidisciplinary Journal, 27(3)*, 380-395, doi:10.1080/10705511.2019.1649983.

MacCallum, R. C., Roznowski, M., & Necowitz, L. B. (1992). Model modifications in covariance structure analysis: The problem of capitalization on chance. *Psychological Bulletin, 111*, 490-504.

Martin, A. J. (2006). Personal Bests (PBs): A proposed multidimensional model and empirical analysis. *The British Psychological Society, 76*, 803-825.

Meade, A. W., Johnson, E. C., & Braddy, P. W. (2008). Power and sensitivity of alternative fit indices in tests of measurement invariance. *Journal of Applied Psychology, 93*, 568-592.

Meng, X.-L. (1994). Posterior predictive p-values. *The Annals of Statistics, 22*, 1142–1160. doi:10.1214/aos/1176325622.

Meredith, W. (1993). Measurement invariance, factor analysis and factorial invariance. *Psychometrika, 58*, 525–543. doi:10.1007/BF02294825.

Merkle, E. C., & Rosseel, Y. (2018). blavaan: Bayesian structural equation models via parameter expansion. *Journal of Statistical Software, 85(4),* 1-30. doi:10.18637/jss.v085.i04.

Merkle, E. C., Fitzsimmons, E., Uanhoro, J., Goodrich, B. (2021). Efficient Bayesian structural equation modeling in Stan. *Journal of Statistical Software, 100(6),* 1-22. doi:10.18637/jss.v100.i06.

Millsap, R. E. (2011). *Statistical Approaches to Measurement Invariance*. New York: Routledge.

Moreira, P., Cunha, D., & Inman, R. A. (2019). Achievement Emotions Questionnaire-Mathematics (AEQ-M) in adolescents: Factorial structure measurement invariance and convergent validity with personality. *European Journal of Developmental Psychology, 16(6)*, 750-762, doi:10.1080/17405629.2018.1548349.

Mullen, M. R. (1995). Diagnosing measurement equivalence in cross-national research. *Journal of International Business Studies, 26*, 573–596. https://doi.org/fjgsqq.

Muthén, B., & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological Methods*, *17*(3), 313–335. https://doi.org/10.1037/a0026802

Myers, M. B., Calantone, R. J., Page, T. J., Jr., & Taylor, C. R. (2000). An application of multiple-group causal models in assessing cross-cultural measurement equivalence. *Journal of International Marketing, 8*, 108–121. https://doi.org/drz4f6

Nye, C. & Drasgow, F. (2011). Effect size indices for analyses of measurement equivalence: Understanding the practical importance of differences between groups. *Journal of Applied Psychology, 96(5)*, 966-980.

Oppzda-Suder, S., Karteczka-Świętek, K., & Piasecka, M. (2021). Psychometric properties and longitudinal measurement invariance of the drug craving scale Modification of the Polish version of the Penn Alchohol Craving Scale (PACS). *PLOS ONE, 6(9),* e0256018. https://doi.org/10.1371/journal.pone.025601.

Pavlov, G., Maydeu-Oliv, A., & Shi, D. (2021). Using the standardized root mean squared residual (SRMR) to assess exact fit in structural equation models. *Educational and Psychological Measurement, 81(1),* 100-130. doi:10.1177/0013164420926231.

Plummer, M. (2006). Comment on Article by Celeux et al. *Bayesian Analysis, 1(4),* 681–686.

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, *10*, 191-218.

Putnick, D. L. & Bornstein, M. H. (2016). Measurement invariance conventions and reporting: The state of the art and future directions for psychological research. *Developmental Review, 41*, 71–90. doi:10.1016/j.dr.2016.06.004.

Riordan, C. M., & Vandenberg, R. J. (1994). A central question in cross-cultural research: Do employees of different cultures interpret work-related measures in an equivalent manner? *Journal of Management, 20*, 643–671.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48(2)*, 1-36. https://doi.org/10.18637/jss.v048.i02.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics, 12*, 1151–1172. doi:10.1214/aos/1176346785.

Satorra, A., & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika, 66(4),* 507-514.

Satorra, A., & Bentler, P. M. (2010). Ensuring postiveness of the scaled difference chi-square test statistic. *Psychometrika, 75(2),* 243-248.

Scheffers, M., van Duijn, M. A. J., Bosscher, R. J., Wiersma, D., Schoevers, R. A., & van Busschbach, J. T. (2017). Psychometric properties of the Dresden body image questionnaire: A multiple-group confirmatory factor analysis across sex and age in a 45 Dutch non-clinical sample. *PLOS ONE, 12(7),* e0181908. https://doi.org/10.1371/journal.pone.0181908.

Scheines, R., Hoijtink, H., & Boomsma, A. (1999). Bayesian estimation and testing of structural equation models. *Psychometrika, 64(1),* 37–52. doi:10.1007/BF02294318.

Schwarz, G. (1978). Estimation the dimension of a model. *Annals of Statistics, 6*, 461-464.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Linde, A. (2014). The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 76(3)*, 485–493. doi:10.1111/rssb.2014.76.issue-3.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 64(4),* 583–639. doi:10.1111/rssb.2002.64.issue-4.

Steenkamp, J.-B. E., & Baumgartner, H. (1998). Assessing measurement invariance in cross-national consumer research. *Journal of Consumer Research, 25,* 78–90. https://doi.org/d7rr8k.

Steiger, J. H., & Lind, J. C. (1980). *Statistically based tests for the number of common factors.* Paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Svetina, D., Rutkowski, L., & Ruthkowski, D. (2020). Multiple-group invariance with categorical outcomes using updated guidelines: An illustration using Mplus and the lavaan semTools packages. *Structural Equation Modeling: A Multidisciplinary Journal, 27(1),* 111-130, doi:10.1080/10705511.2019.1602776.

Tan, T. K., & Feng, Q. (2022). Validity and reliability of Min-Mental State Examination in Older Adults in China: Inline Mini-Mental State Examination with cognitive functions. *International Journal of Population Studies, 8(2),* 15-30. doi:10.18063/ijps.v8i2.1285.

Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika, 38 (1)*, 1–10.

Van de Schoot, R., Kaplan, D., Denissen, J., Asendorpf, J. B., Neyer, F. J., & van Aken, M. A. G. (2014). A gentle introduction to Bayesian analysis: Applications to developmental research. *Child Development, 85*, 842-860. doi:10.1111/cdev.12169

Vandenberg, R. J. (2002). Toward a further understanding of and improvement in measurement invariance methods and procedures. *Organizational Research Methods, 5*, 139. doi:10.1177/1094428102005002001.

Vandenberg, R. J., & Lance, C. E. (2000). A review and synthesis of the measurement invariance literature: suggestions, practices, and recommendations for organizational research. *Organizational Research Methods, 3 (1)*, 4–70. doi:10.1177/109442810031002.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing, 27*, 1413–1432. doi:10.1007/s11222-016-9696-4.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research, 11*, 3571-3594.

Wei, T., & Simko, V. (2021). *R package 'corrplot': Visualization of a correlation matrix* (Version 0.92). Available from https://github.com/taiyun/corrplot.

Widaman, K. F. & Reise, S. P. (1997). Exploring the measurement invariance of psychological instruments: Applications in the substance use domain. In: Bryant, KJ.; Windle, ME.; West, SG., editors. *The Science of Prevention: Methodological Advances from Alcohol and Substance Abuse Research.* Washington, DC: American Psychological Association, 281-324.

Winter, S. D. & Depaoli, S. (2022). Performance of model fit and selection indices for Bayesian structural equation modeling with missing data. *Structural Equation Modeling: A Multidisciplinary Journal, 29(4),* 531-549, https://doi.org/10.1080/10705511.2021.2018656.

Yang, Y, (2008). *Partial invariance in loadings and intercepts - Their interplays and implications for latent mean comparisons.* [PhD Thesis, University of Nebraska]. University of Nebraska.

Yu, J.-P., Choi, H.-J. & Kim, J.-H. (2019). Multigroup analysis and measurement equivalence: Korean and Chinese consumers of Korean cosmetics. *Social Behavior and Personality, 47(3),* e7499. https://doi.org/10.2224/sbp.7499.

Yuan, K. H., & Chan, W. (2016). Measurement invariance via multigroup SEM: Issues and solutions with chi-square-difference tests. *Psychological Methods, 21(3),* 405-426. https://doi.org/10.1037/met0000080.

Citation:
Tan. T. K.. (2024). Frequentist and Bayesian Factorial Invariance using R. *Practical Assessment, Research, & Evaluation*, 29(8). Available online: DOI TO BE ADDED BY EDITORS.

Corresponding Author:
Teck Kiang Tan
National University of Singapore

Email: alsttk [at] nus.edu.sg

Appendix A. R Packages Functions and Syntax

Table A1. Functions for Graphing EFA and CFA Model - Package EGAnet and semPlot

<table>
<tr><th>Package EGAnet</th><th>Description</th></tr>
<tr><td><pre>ega.HSQ <- EGA(DF,
            uni.method = "LE",
            corr   = "cor_auto",
            model   = "glasso",
            algorithm = "walktrap",
            plot.EGA   = TRUE,
            plot.type = "qgraph")</pre></td><td>Exploratory Graph Analysis</td></tr>
<tr><td>Package semPlot</td><td></td></tr>
<tr><td><pre>semPaths(model,
     what   = "est",
     color   = "yellow",
     edge.color   = "blue",
     sizeLat   = 8,
     sizeMan   = 8,
     edge.label.cex = .75)</pre></td><td>Plot Path Diagram – Frequentist and Bayesian</td></tr>
</table>

Table A2. Functions for Fitting Confirmatory Factor Model and Factorial Invariance – Package lavaan and blavaan

Package / Function	Description
Model Specification `Model <- ' L1 =~ x1 + x2 + x3` `          L2 =~ x4 + x5 + x6` `          L3 =~ x7 + x8 + x9 '`	Specify a CFA Model
Package lavaan – Factorial Invariance	
`Configural<-cfa(Model,data=DF,group="Gp")`	Configural Invariance
`Metric<-cfa(Model,data=DF,group="Gp",` `group.equal="loadings")`	Metric Invariance
`Scalar<-cfa(Model,data=DF,group="Gp",` `group.equal="loadings","intercepts")`	Scalar Invariance
`Strict<-cfa(Model,data=DF,group="Gp",` `group.equal="loadings","intercepts","residuals")`	Strict Invariance
Package blavaan – Factorial Invariance	
`Configural.B<-bcfa(Model,data=DF,group="Gp")`	Bayesian Configural Invariance
`Metric.B<-bcfa(Model,data=DF,group="Gp",` `group.equal="loadings")`	Bayesian Metric Invariance
`Scalar.B<-bcfa(Model,data=DF,group="Gp",` `group.equal="loadings","intercepts")`	Bayesian Scalar Invariance
`Strict.B<-bcfa(Model,data=DF,group="Gp",` `group.equal="loadings","intercepts","residuals")`	Bayesian Strict Invariance

Table A3. Functions for Generate Fit Statistics and Carry out Difference Test – Package lavaan, blavaan, semTools, and equaltestMI

Fit Statistics and Difference Test	Description
Package lavaan – Global Fit Measures	
(1) Frequentist Fit Indices	
`fitMeasures(Configural)`	Fit Measures - Configural Invariance
`fitMeasures(Metric)`	Fit Measures - Metric Invariance
`fitMeasures(Scalar)`	Fit Measures - Scalar Invariance
`fitMeasures(Strict)`	Fit Measures - Strict Invariance
(2) Bayesian – PPP,DIC,WAIC, and LOOIC	
`fitMeasures(Configural.B)`	Fit Measures - Bayesian Configural Invariance
`fitMeasures(Metric.B)`	Fit Measures - Bayesian Metric Invariance
`fitMeasures(Scalar.B)`	Fit Measures - Bayesian Scalar Invariance
`fitMeasures(Strict.B)`	Fit Measures - Bayesian Strict Invariance
Package blavaan	
Bayesian – MCMC Posterior	
`blavFitIndices(Configural.B,rescale="MCMC",` `fit.measures=c("rmsea","cfi","tli","aic","bic"))`	Posterior - Bayesian Configural Invariance
`blavFitIndices(Metric.B,rescale="MCMC",` `fit.measures=c("rmsea","cfi","tli","aic","bic"))`	Posterior - Bayesian Metric Invariance
`blavFitIndices(Scalar.B,rescale="MCMC",` `fit.measures=c("rmsea","cfi","tli","aic","bic"))`	Posterior - Bayesian Scalar Invariance
`blavFitIndices(Sgtrict.B,rescale="MCMC",` `fit.measures=c("rmsea","cfi","tli","aic","bic"))`	Posterior - Bayesian Strict Invariance
Package semTools – Compare Model Fit	
Frequentist	
`compareFit(Configural,Metric)`	Compare Fit between Configural and Metric
`compareFit(Metric,Scalar)`	Compare Fit between Metric and Scalar
`compareFit(Scalar,Strict)`	Compare Fit between Scalar and Strict
`compareFit(Configural,Metric,Scalar,Strict)`	Compare Fit for All
`compareFit(Configural,Metric,Scalar,Strict,` ` fit.measures =` `  c("rmsea","cfi","tli","aci","bci")`	Restrict to Selected Statistics
Package equaltestMI **Comprehensive Factorial Invariance**	
`library(equaltestMI)` `eqMI.main(model = Model,` `  data             = DF,` `  group            = "school",` `  meanstructure    = TRUE,` `  equivalence.test = TRUE,` `  adjRMSEA         = TRUE)`	Comprehensive Factorial Invariance

Appendix B. Package equaltestMI, Function eqMI.main

```
library(equaltestMI)
MI_EQ <- eqMI.main(model = HS.Model,
  data              = H,
  group             = "Gp",
  meanstructure     = TRUE,
  equivalence.test = TRUE,
  adjRMSEA          = TRUE)

MI_EQ
```

```
$AnnotatedOutput$`Equality of Population Covariance Matrices under NHT`
              Chisq Df     pvalue
fit.pop.cov 59.50675 45 0.07231226

$AnnotatedOutput$`Chi-Square and Chi-Square-Difference Test under NHT`
                      Chisq Df   pvalue Chisq.diff Df.diff   pvalue
fit.pop.cov           59.507 45    0.072
fit.configural.g1     51.542 24    0.001
fit.configural.g2     64.309 24    0.000
fit.combine.groups   115.851 48
fit.metric           124.044 54    0.000      8.192       6    0.224
fit.residuals        141.994 63    0.000     17.951       9    0.036
fit.varfactor        147.949 69    0.000      5.955       6    0.428
fit.scalar           164.103 60    0.000     40.059       6    0.000
fit.strong.means     204.605 63    0.000     40.502       3    0.000
fit.strict.residuals 181.511 69    0.000     17.409       9    0.043
fit.strict.means     221.335 72    0.000     39.824       3    0.000
$AnnotatedOutput$`T-size epsilon, RMSEA, and Adjusted Cutoff Values under ET`
                   epsilon_t  RMESA_t   cut.01   cut.05   cut.08   cut.10 goodness-of-fit
fit.pop.cov            0.129    0.076    0.056    0.079    0.105    0.123           close
fit.configural.g1      0.175    0.121    0.067    0.089    0.114    0.133        mediocre
fit.configural.g2      0.228    0.138    0.067    0.089    0.114    0.133            poor
fit.metric             0.047    0.125    0.106    0.124    0.150    0.168            fair
fit.residuals          0.084    0.137    0.092    0.112    0.137    0.155            fair
fit.varfactor          0.034    0.106    0.106    0.124    0.150    0.168       excellent
fit.scalar             0.194    0.254    0.106    0.124    0.150    0.168            poor
fit.strong.means       0.207    0.371    0.138    0.153    0.177    0.195            poor
fit.strict.residuals   0.081    0.134    0.092    0.112    0.137    0.155            fair
fit.strict.means       0.204    0.369    0.138    0.153    0.177    0.195            poor
$eqMI.stat
                          Chisq Df       pvalue  epsilon_t   RMESA_t     cut.01     cut.05     cut.08    cut.10 goodness-of-fit
fit.pop.cov             59.506752 45 7.231226e-02 0.12926395 0.0757962 0.05565624 0.07902077 0.10461050 0.1230937           close
fit.configural.g1       51.542237 24 8.973209e-04 0.17460895 0.1206265 0.06708755 0.08900601 0.11443055 0.1328721        mediocre
fit.configural.g2       64.309106 24 1.536263e-05 0.22753012 0.1376984 0.06708755 0.08900601 0.11443055 0.1328721            poor
fit.combine.groups     115.851344 48 1.545283e-07 0.34549694 0.1199821 0.05464250 0.07815196 0.10376664 0.1222601        mediocre
fit.metric             124.043544 54 1.962798e-07 0.35579138 0.1147931 0.05286000 0.07663364 0.10229874 0.1208143        mediocre
fit.residuals          141.994425 63 4.988611e-08 0.39299355 0.1116960 0.05065246 0.07477231 0.10051365 0.1190654        mediocre
fit.varfactor          147.949069 69 1.092643e-07 0.39440854 0.1069211 0.04941282 0.07373781 0.09953001 0.1181072        mediocre
fit.scalar             164.102831 60 1.296141e-11 0.48859894 0.1276191 0.05133631 0.07534644 0.10106233 0.1196018            poor
fit.strong.means       204.605279 63 1.110223e-16 0.63127475 0.1415644 0.05065246 0.07477231 0.10051365 0.1190654            poor
fit.strict.residuals   181.511345 69 4.646394e-12 0.52272303 0.1230910 0.04941282 0.07373781 0.09953001 0.1181072            poor
fit.strict.means       221.335082 72 0.000000e+00 0.66216914 0.1356230 0.04884828 0.07326957 0.09908715 0.1176773            poor
fit.metric.diff          8.192200  6 2.243577e-01 0.04656763 0.1245895 0.10645397 0.12441675 0.14950179 0.1679588            fair
fit.residuals.diff      17.950881  9 3.574771e-02 0.08386272 0.1365143 0.09235012 0.11164801 0.13689658 0.1553715            fair
fit.varfactor.diff       5.954644  6 4.282900e-01 0.03356515 0.1057752 0.10645397 0.12441675 0.14950179 0.1679588       excellent
fit.scalar.diff         40.059287  6 4.434566e-07 0.19384344 0.2541938 0.10645397 0.12441675 0.14950179 0.1679588            poor
fit.strong.means.diff   40.502448  3 8.337521e-09 0.20701267 0.3714949 0.13758478 0.15270698 0.17711958 0.1953541            poor
fit.strict.residuals.diff 17.408514  9 4.269001e-02 0.08125776 0.1343774 0.09235012 0.11164801 0.13689658 0.1553715         fair
fit.strict.means.diff   39.823737  3 1.161229e-08 0.20414679 0.3689144 0.13758478 0.15270698 0.17711958 0.1953541            poor
```

Appendix C. Abbreviation Used in Function dpriors

Table C1. Abbreviation Used in Function dpriors

Abbreviation	Description
nu	Observed variable intercept parameters
alpha	Latent variable intercept parameters
lambda	Loading parameters
beta	Regression parameters
itheta	Observed variable precision parameters
ipsi	Latent variable precision parameters
rho	Correlation parameters (associated with covariance parameters)