MID - Measurement Model Code

¶ ¶

¶ by Michael Demidenko ¶

January 2023

Contents

The platform used to write and generate this .html document is `cat(platform)` on RStudio, `cat(r_info)`. The required packages to run all necessary code chunks are checked and installed using the pacman R package. `p_load` from the package is used to load and install all listed packages. The details for a number of packages are described before their use. If they are not, the `.rmd` file will provide the list of packages that are installed.

Simulating data

This section is unique only to the Stage 1 registered reported. We are not using any real data, so to ensure that each step is completed correctly, we use simulated data. We can operate under the assumption that all participants will have no missing values given how exclusion criteria are applied and fMRI data are extracted from the BOLD timeseries.

In this section, a population model is fit reflecting the approach and avoidance (measured constructs) that are [presumed] to be associated with stimuli within/across tasks (Manifest variables). As formalized here, in the MID task the contrasts reflect the manifest variables (i.e., reflective model) as opposed of the other way, whereby the items form the variables predict the construct (i.e., formative model).

Specify Population Model

Start off by specifying the **population** model. In this scenario, the individual runs load onto the specific Contrast and ROI combinations. Then, the ROIs are loaded onto the factors `approach` and `avoidance`. The approach and avoidance are specified as *negatively* correlated and the factor variances are fixed to `1`. This is because in the multidimensional circumplex model, when the span of the contrasts are considered, these measures will likely not be orthogonal but instead >90 degree in space.

```
population_model<-'
# By run loadings for bilateral regions
AWin_v_Neut_L_NAcc =~     .7*AWin_v_Neut_L_NAcc_run1    + .7*AWin_v_Neut_L_NAcc_run2
AWin_v_Neut_L_Insula =~   .7*AWin_v_Neut_L_Insula_run1  + .7*AWin_v_Neut_L_Insula_run2
BWin_v_Neut_L_NAcc =~     .7*BWin_v_Neut_L_NAcc_run1    + .7*BWin_v_Neut_L_NAcc_run2
BWin_v_Neut_L_Insula =~   .7*BWin_v_Neut_L_Insula_run1  + .7*BWin_v_Neut_L_Insula_run2
BWin_v_BLose_L_NAcc =~    .7*BWin_v_BLose_L_NAcc_run1   + .7*BWin_v_BLose_L_NAcc_run2
```

```
BWin_v_BLose_L_Insula =~  .7*BWin_v_BLose_L_Insula_run1 + .7*BWin_v_BLose_L_Insula_run2
ALose_v_Neut_L_NAcc =~    .7*ALose_v_Neut_L_NAcc_run1   + .7*ALose_v_Neut_L_NAcc_run2
ALose_v_Neut_L_Insula =~  .7*ALose_v_Neut_L_Insula_run1 + .7*ALose_v_Neut_L_Insula_run2
BLose_v_Neut_L_NAcc =~    .7*BLose_v_Neut_L_NAcc_run1   + .7*BLose_v_Neut_L_NAcc_run2
BLose_v_Neut_L_Insula =~  .7*BLose_v_Neut_L_Insula_run1 + .7*BLose_v_Neut_L_Insula_run2
BLose_v_BWin_L_NAcc =~    .7*BLose_v_BWin_L_NAcc_run1   + .7*BLose_v_BWin_L_NAcc_run2
BLose_v_BWin_L_Insula =~  .7*BLose_v_BWin_L_Insula_run1 + .7*BLose_v_BWin_L_Insula_run2

AWin_v_Neut_R_NAcc =~     .7*AWin_v_Neut_R_NAcc_run1    + .7*AWin_v_Neut_R_NAcc_run2
AWin_v_Neut_R_Insula =~   .7*AWin_v_Neut_R_Insula_run1  + .7*AWin_v_Neut_R_Insula_run2
BWin_v_Neut_R_NAcc =~     .7*BWin_v_Neut_R_NAcc_run1    + .7*BWin_v_Neut_R_NAcc_run2
BWin_v_Neut_R_Insula =~   .7*BWin_v_Neut_R_Insula_run1  + .7*BWin_v_Neut_R_Insula_run2
BWin_v_BLose_R_NAcc =~    .7*BWin_v_BLose_R_NAcc_run1   + .7*BWin_v_BLose_R_NAcc_run2
BWin_v_BLose_R_Insula =~  .7*BWin_v_BLose_R_Insula_run1 + .7*BWin_v_BLose_R_Insula_run2
ALose_v_Neut_R_NAcc =~    .7*ALose_v_Neut_R_NAcc_run1   + .7*ALose_v_Neut_R_NAcc_run2
ALose_v_Neut_R_Insula =~  .7*ALose_v_Neut_R_Insula_run1 + .7*ALose_v_Neut_R_Insula_run2
BLose_v_Neut_R_NAcc =~    .7*BLose_v_Neut_R_NAcc_run1   + .7*BLose_v_Neut_R_NAcc_run2
BLose_v_Neut_R_Insula =~  .7*BLose_v_Neut_R_Insula_run1 + .7*BLose_v_Neut_R_Insula_run2
BLose_v_BWin_R_NAcc =~    .7*BLose_v_BWin_R_NAcc_run1   + .7*BLose_v_BWin_R_NAcc_run2
BLose_v_BWin_R_Insula =~  .7*BLose_v_BWin_R_Insula_run1 + .7*BLose_v_BWin_R_Insula_run2

#Factor item loadings
Approach =~  .8*AWin_v_Neut_L_NAcc + .8*AWin_v_Neut_R_NAcc + .45*AWin_v_Neut_R_Insula +
            .7*BWin_v_Neut_L_NAcc +   .7*BWin_v_Neut_R_NAcc + .4*BWin_v_Neut_R_Insula +
            .8*BWin_v_BLose_L_NAcc +  .8*BWin_v_BLose_R_NAcc

Avoid =~  .8*ALose_v_Neut_L_Insula  + .8*ALose_v_Neut_R_Insula +
          .75*BLose_v_Neut_L_Insula + .75*BLose_v_Neut_R_Insula +
          .8*BLose_v_BWin_L_Insula  + .45*BLose_v_BWin_R_Insula

# Factor Covariances
Approach ~~ -.6*Avoid

# Fixing factor variances
Approach ~~ 1*Approach
Avoid ~~ 1*Avoid

'
```

General samples

Using the population model, simsem is used to create simulated data. This generates a `fake` dataset that is used to pilot the planned Mutligroup CFA, ESEM, EFA and Local SEM models

In this case, 50 repetitions are simulated per model for an *approximate* N sample for each study. Even though the factor variances are specified in the population model as '1', this model fixeds all latent variables using `std.lv = TRUE`.

1. AHRB N = 108
2. MLS N = 159
3. ABCD N = 1000

```
set.seed(25151215)
sim_AHRB <- simsem::sim(nRep = 50, model = "lavaan", n = 108,
          generate = population_model, std.lv = TRUE, lavaanfun = "sem",
           # std.lv ~ ix the variances of all the latent variables
           dataOnly=T, meanstructure = FALSE, seed=123)

sim_MLS <- simsem::sim(nRep = 50, model = "lavaan", n = 159,
                generate = population_model, std.lv = TRUE, lavaanfun = "sem",
                dataOnly=T, meanstructure = FALSE, seed=123)

sim_ABCD <- simsem::sim(nRep = 50, model = "lavaan", n = 1000,
               generate = population_model, std.lv = TRUE, lavaanfun = "sem",
               dataOnly=T, meanstructure = FALSE, seed=123)
```

Average each repetition for each simulated sample. For example, after 50 repetitions of 1000 participants for the population model of ABCD sample, an average estimate is derived using aaply. For each study, the `set` variable is created to differentiate which sample the data is associated with (i.e., grouping variable). In this case, `AHRB = 3`, `MLS = 2`, `ABCD = 1`.

```
sim_AHRB_data <- data.frame(aaply(laply(sim_AHRB, as.matrix), c(2,3), mean))
  sim_AHRB_data$set <-3

sim_MLS_data <- data.frame(aaply(laply(sim_MLS, as.matrix), c(2,3), mean))
  sim_MLS_data$set <-2

sim_ABCD_data <- data.frame(aaply(laply(sim_ABCD, as.matrix), c(2,3), mean))
  sim_ABCD_data$set <-1
```

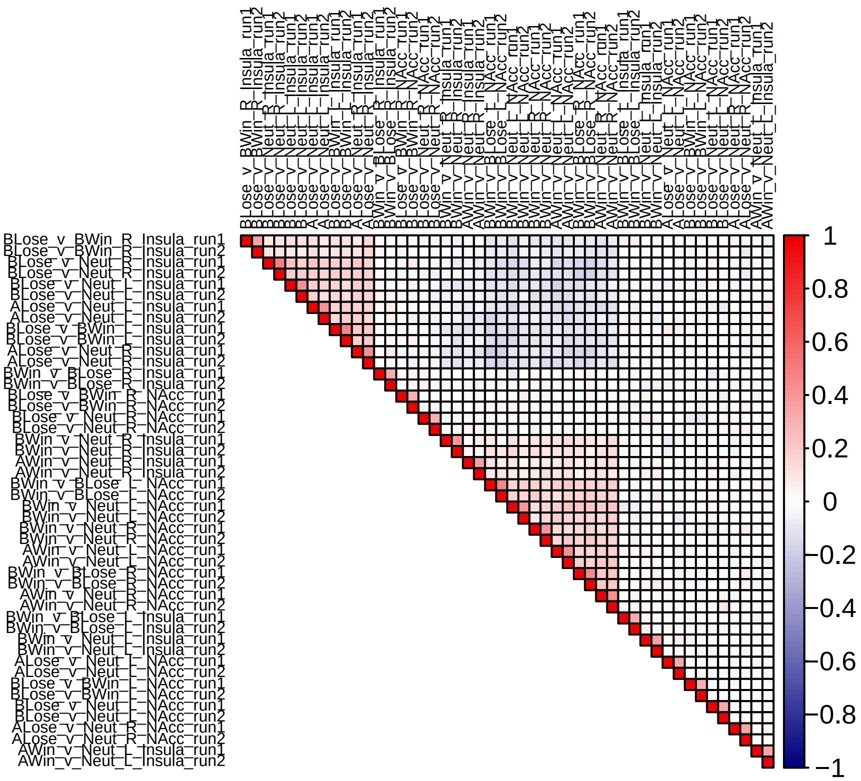
Next, row bind the data sets to form one complete data. This equivalent to stacking the datasets and only retaining a single header row (since all variable names are constant). This creates a 1267 x 49 data matrix (48 brain variables + sample label)

```
brain_set <- rbind(sim_AHRB_data,sim_MLS_data,sim_ABCD_data)
```

Correlation matrix of data

Here, a combination of rcorr and corrplot are used to visualize the associations among the brain variables in the dataset.

```
# Using Hmisc to create a 24x24 matrix for a list (3) that contains: the pearson's r corr, sample size
Brain_corr = rcorr(as.matrix(subset(brain_set,select=-c(set))), # excluding the set of data related to
                   type = "pearson")


# Using corrplot() to create heatmap of the data.
par(mfrow=c(1,1))
corrplot(Brain_corr$r, type = "upper",
         order = 'hclust',
         method =  "color",
         tl.cex = 0.5, tl.col = 'black',
         cl.pos = 'r', tl.pos = 'lt', outline = TRUE,
         col=colorRampPalette(c("navyblue","white","red2"))(100),# colours http://www.stat.columbia.edu
         mar = c(2,.15,.25,.15)#bottom, left, top and right,
         )
```



Running [restricted] Multigroup CFA

Run the CFA multi-group analysis for the three datasets. Multi-group CFA tests the measurement invariance across defined groups to determine whether soft and strict invariance criteria are met and the degree to which the derive estimates for an item in one study can be compared to the same item in another sample. In this case, the focus is on the configural (structure) and metric invariance (loadings). In short, this model evaluates whether factor structure and loadings for the approach and avoidance model are invariant (dont significant differ) across the samples.

Code here is based on measurement invariance models from Maasen et al. 2019, Measurement invariance presentation from Kate Xu and Multi-group CFA tutorial from Hirschfeld & Brachel (2014).

The issue of multi-group is invariance what is discussed in Borsboom (2006). In short, (1) Interpretation of group differences on observed scores DEPENDS on the invariance of measurement models & (2) many make conclusions without doing a single test of measurement invariance.

A tutorial on CFA broadly is available from Lizbeth Benzon and Nilam Ram here and tutorial of measurement invariance (in context of longitudinal data) from Nilam Ram is available here

CFA model

The below specified model will be used. The number of estimate parameters are fewer and may be more appropriate for the theoretical model. This model may result in few convergence issues if the number of participants ends up to be few and the coefficients/estimates are lower.(see Kline 2015 book on **Principles and Practice of Structural Equation Modeling**).

```
MID_model <-'

# Factor loadings
Approach =~ AWin_v_Neut_L_NAcc_run1  + AWin_v_Neut_R_NAcc_run1  + AWin_v_Neut_R_Insula_run1 +
            AWin_v_Neut_L_NAcc_run2  + AWin_v_Neut_R_NAcc_run2 + AWin_v_Neut_R_Insula_run2 +
            BWin_v_Neut_L_NAcc_run1  + BWin_v_Neut_R_NAcc_run1  + BWin_v_Neut_R_Insula_run1 +
            BWin_v_Neut_L_NAcc_run2  + BWin_v_Neut_R_NAcc_run2  + BWin_v_Neut_R_Insula_run2 +
            BWin_v_BLose_L_NAcc_run1 + BWin_v_BLose_R_NAcc_run1 +
            BWin_v_BLose_L_NAcc_run2 + BWin_v_BLose_R_NAcc_run2

Avoid =~    ALose_v_Neut_L_Insula_run1 + ALose_v_Neut_R_Insula_run1 +
            ALose_v_Neut_L_Insula_run2 + ALose_v_Neut_R_Insula_run2 +
            BLose_v_Neut_L_Insula_run1 + BLose_v_Neut_R_Insula_run1 +
            BLose_v_Neut_L_Insula_run2 + BLose_v_Neut_R_Insula_run2 +
            BLose_v_BWin_L_Insula_run1 + BLose_v_BWin_R_Insula_run1 +
            BLose_v_BWin_L_Insula_run2 + BLose_v_BWin_R_Insula_run2
'
```

Running CFA: Three Samples

Below is the CFA model that is used to test the proposed restricted model (see Figure 1 in the manuscript). The CFA fitting procedure is consistent with the description here. For each CFA model, the full sample is filtered for each type sample, e.g. AHRB, MLS, ABCD. The `std.lv= = TRUE` constrain the latent factor variances to *1*. The estimator being used is `MLR`, a maximum likelihood robust estimator. A CFA model is estimated for the complete data (i.e., all three datasets).

```
all_sample <- cfa(model = MID_model, data = brain_set,
               estimator = "MLR", std.lv = TRUE, meanstructure = TRUE)
```

Fitting Configural CFA

Here, the `configular multigroup model` is fit. As described in D'Urso et al. (2022) measurement invariance pre-print, the configural model tests:

is the structure of the factors is invariannt across the samples ('set'). In other words, if we *a priori* propose a two-factor structure (FA 1 = approch and FA 2 = Avoidance), does this two factor structure represent the between-person variability in the items that reflect the factors across each sample?

If the variability in one sample suggests a one, three, or four factor structure, this will be degrade the fit statistics.

A pre-specified CFA model is used to evaluate whether the measures/items that reflect the factor are the same across groups. `group= 'set'` is used to define the grouping variable. All loadings and intercepts are free to vary across groups, and the factor variance is set to '1' via `std.lv = TRUE`

```
configural_cfa <- cfa(model = MID_model, data = brain_set, group = 'set',
                      estimator = "MLR", std.lv = TRUE, meanstructure = TRUE)
```

Fitting Metric CFA

After fitting the CFA configurial (factor structure) invariance, if the model fit is not poor, then the next step is to test the metric invariance. Metric invariance tests:

are the loadings are consistent across the groups. In other words,are the phenomena (i.e., approach and avoidance) reflected by the same pattern across the measures/items?

One cause for concern may be that the phenomenon are not invariant across age groups, in that the items/measures (ROIs for a given contrast) do not load in the same manner onto each factor. This 'soft' measure of invariance can determine whether the items functions differ across the items and so cannot be easily compared.

The model is fit using the same procedure as for configurial invariance with one exception: In metric invariance the loadings group equality constraint is added to the model via `group.equal=c("loadings")`. The model fit statistics are used to evaluate whether the fit is poor.

```
metric_cfa <-cfa(model = MID_model, data = brain_set,
                 group = 'set', group.equal=c("loadings"),
                 estimator = "MLR", std.lv = TRUE, meanstructure = TRUE)
```

Extracting Fit Statistics

Once the above models are fit, the following information is pulled out and saved into a `out` data frame:

1. Model name
2. Chi-square statistics
3. Model Degrees of Freedom (df)
4. Model p-value
5. RMSEA
6. CFI
7. SRMR
8. AIC
9. BIC

```
# Below selects specific fit data as described in Maassen et al. 2019 OSF. No comparisons are made to c
out <- matrix(NA, ncol = 10, nrow = 4)
colnames(out) <- c("model","chisq","df","pvalue", "rmsea", "cfi","tli", "srmr",
                   "AIC", "BIC")

# save fit measures from models
out[1,2:8] <- round(data.matrix(fitmeasures(all_sample,
                                            fit.measures = c("chisq","df","pvalue",
                                                             "rmsea", "cfi","tli", "srmr"))),
                    digits=3)

out[2,2:8] <- round(data.matrix(fitmeasures(configural_cfa,
                                            fit.measures = c("chisq","df","pvalue",
                                                             "rmsea", "cfi","tli", "srmr"))),
                            digits=3)

out[3,2:8] <- round(data.matrix(fitmeasures(metric_cfa,
                                            fit.measures = c("chisq","df","pvalue",
                                                             "rmsea", "cfi","tli", "srmr"))),
                    digits=3)


# AIC models
out[1,9] <- round(AIC(all_sample),3)
out[2,9] <- round(AIC(configural_cfa),3)
out[3,9] <- round(AIC(metric_cfa),3)
```

```r
# BIC models
out[1,10] <- round(BIC(all_sample),3)
out[2,10] <- round(BIC(configural_cfa),3)
out[3,10] <- round(BIC(metric_cfa),3)

out[1:3,1] <-  c("Overall CFA", "Configg MG-CFA", "Metric MG-CFA")
```

Model Parameter Summary

Reporting standardized coefficients.

All Sample CFA model

```r
##### Summarizing All Samples CFA model #####
parameters(all_sample, standardize = T)
```

```
## # Loading
## 
## Link                                  | Coefficient |   SE |       95% CI |     z |      p
## -------------------------------------------------------------------------------------------
## Approach =~ AWin_v_Neut_L_NAcc_run1   |        0.44 | 0.03 | [0.38, 0.51] | 14.45 | < .001
## Approach =~ AWin_v_Neut_R_NAcc_run1   |        0.45 | 0.03 | [0.40, 0.51] | 15.78 | < .001
## Approach =~ AWin_v_Neut_R_Insula_run1 |        0.22 | 0.03 | [0.16, 0.29] |  7.08 | < .001
## Approach =~ AWin_v_Neut_L_NAcc_run2   |        0.48 | 0.03 | [0.43, 0.54] | 16.93 | < .001
## Approach =~ AWin_v_Neut_R_NAcc_run2   |        0.48 | 0.03 | [0.43, 0.53] | 17.82 | < .001
## Approach =~ AWin_v_Neut_R_Insula_run2 |        0.24 | 0.03 | [0.17, 0.30] |  7.47 | < .001
## Approach =~ BWin_v_Neut_L_NAcc_run1   |        0.48 | 0.03 | [0.42, 0.53] | 17.17 | < .001
## Approach =~ BWin_v_Neut_R_NAcc_run1   |        0.37 | 0.03 | [0.31, 0.43] | 12.53 | < .001
## Approach =~ BWin_v_Neut_R_Insula_run1 |        0.23 | 0.03 | [0.17, 0.29] |  7.32 | < .001
## Approach =~ BWin_v_Neut_L_NAcc_run2   |        0.40 | 0.03 | [0.35, 0.46] | 13.99 | < .001
## Approach =~ BWin_v_Neut_R_NAcc_run2   |        0.41 | 0.03 | [0.36, 0.47] | 14.73 | < .001
## Approach =~ BWin_v_Neut_R_Insula_run2 |        0.27 | 0.03 | [0.21, 0.33] |  8.30 | < .001
## Approach =~ BWin_v_BLose_L_NAcc_run1  |        0.45 | 0.03 | [0.39, 0.50] | 15.12 | < .001
## Approach =~ BWin_v_BLose_R_NAcc_run1  |        0.47 | 0.03 | [0.41, 0.53] | 16.47 | < .001
## Approach =~ BWin_v_BLose_L_NAcc_run2  |        0.42 | 0.03 | [0.37, 0.48] | 14.19 | < .001
## Approach =~ BWin_v_BLose_R_NAcc_run2  |        0.47 | 0.03 | [0.42, 0.53] | 16.34 | < .001
## Avoid =~ ALose_v_Neut_L_Insula_run1   |        0.46 | 0.03 | [0.40, 0.52] | 15.36 | < .001
## Avoid =~ ALose_v_Neut_R_Insula_run1   |        0.48 | 0.03 | [0.42, 0.53] | 16.58 | < .001
## Avoid =~ ALose_v_Neut_L_Insula_run2   |        0.42 | 0.03 | [0.36, 0.48] | 14.25 | < .001
## Avoid =~ ALose_v_Neut_R_Insula_run2   |        0.47 | 0.03 | [0.41, 0.52] | 16.74 | < .001
## Avoid =~ BLose_v_Neut_L_Insula_run1   |        0.44 | 0.03 | [0.38, 0.50] | 14.51 | < .001
## Avoid =~ BLose_v_Neut_R_Insula_run1   |        0.50 | 0.03 | [0.44, 0.56] | 16.42 | < .001
## Avoid =~ BLose_v_Neut_L_Insula_run2   |        0.41 | 0.03 | [0.35, 0.47] | 13.36 | < .001
## Avoid =~ BLose_v_Neut_R_Insula_run2   |        0.48 | 0.03 | [0.43, 0.54] | 16.06 | < .001
## Avoid =~ BLose_v_BWin_L_Insula_run1   |        0.45 | 0.03 | [0.40, 0.51] | 14.94 | < .001
## Avoid =~ BLose_v_BWin_R_Insula_run1   |        0.27 | 0.03 | [0.21, 0.34] |  8.55 | < .001
## Avoid =~ BLose_v_BWin_L_Insula_run2   |        0.45 | 0.03 | [0.39, 0.51] | 14.91 | < .001
## Avoid =~ BLose_v_BWin_R_Insula_run2   |        0.27 | 0.03 | [0.21, 0.34] |  8.53 | < .001
## 
## # Correlation
## 
## Link              | Coefficient |   SE |         95% CI |      z |      p
## -------------------------------------------------------------------------
## Approach ~~ Avoid |       -0.55 | 0.03 | [-0.61, -0.49] | -18.77 | < .001
```

Configural CFA model

```
##### Summarizing Configural MG-CFA model #####
parameters(configural_cfa, standardize = T)
```

```
## # Loading
## 
## Link                                  | Coefficient |   SE |        95% CI |     z |      p | Group
## ----------------------------------------------------------------------------------------------------
## Approach =~ AWin_v_Neut_L_NAcc_run1   |        0.51 | 0.13 | [ 0.26, 0.76] |  4.03 | < .001 |  1.00
## Approach =~ AWin_v_Neut_R_NAcc_run1   |        0.51 | 0.10 | [ 0.31, 0.71] |  5.07 | < .001 |  1.00
## Approach =~ AWin_v_Neut_R_Insula_run1 |        0.19 | 0.13 | [-0.07, 0.45] |  1.43 | 0.154  |  1.00
## Approach =~ AWin_v_Neut_L_NAcc_run2   |        0.47 | 0.13 | [ 0.21, 0.72] |  3.63 | < .001 |  1.00
## Approach =~ AWin_v_Neut_R_NAcc_run2   |        0.38 | 0.12 | [ 0.15, 0.61] |  3.23 | 0.001  |  1.00
## Approach =~ AWin_v_Neut_R_Insula_run2 |        0.18 | 0.14 | [-0.09, 0.45] |  1.29 | 0.198  |  1.00
## Approach =~ BWin_v_Neut_L_NAcc_run1   |        0.33 | 0.11 | [ 0.12, 0.54] |  3.12 | 0.002  |  1.00
## Approach =~ BWin_v_Neut_R_NAcc_run1   |        0.30 | 0.15 | [ 0.01, 0.59] |  2.00 | 0.046  |  1.00
## Approach =~ BWin_v_Neut_R_Insula_run1 |        0.10 | 0.14 | [-0.16, 0.37] |  0.76 | 0.445  |  1.00
## Approach =~ BWin_v_Neut_L_NAcc_run2   |        0.34 | 0.10 | [ 0.14, 0.55] |  3.28 | 0.001  |  1.00
## Approach =~ BWin_v_Neut_R_NAcc_run2   |        0.45 | 0.11 | [ 0.25, 0.66] |  4.30 | < .001 |  1.00
## Approach =~ BWin_v_Neut_R_Insula_run2 |        0.26 | 0.11 | [ 0.04, 0.48] |  2.27 | 0.023  |  1.00
## Approach =~ BWin_v_BLose_L_NAcc_run1  |        0.34 | 0.14 | [ 0.07, 0.61] |  2.45 | 0.014  |  1.00
## Approach =~ BWin_v_BLose_R_NAcc_run1  |        0.48 | 0.10 | [ 0.29, 0.67] |  4.94 | < .001 |  1.00
## Approach =~ BWin_v_BLose_L_NAcc_run2  |        0.44 | 0.15 | [ 0.14, 0.73] |  2.88 | 0.004  |  1.00
## Approach =~ BWin_v_BLose_R_NAcc_run2  |        0.56 | 0.09 | [ 0.38, 0.75] |  6.06 | < .001 |  1.00
## Avoid =~ ALose_v_Neut_L_Insula_run1   |        0.52 | 0.10 | [ 0.32, 0.72] |  5.16 | < .001 |  1.00
## Avoid =~ ALose_v_Neut_R_Insula_run1   |        0.53 | 0.09 | [ 0.34, 0.71] |  5.62 | < .001 |  1.00
## Avoid =~ ALose_v_Neut_L_Insula_run2   |        0.55 | 0.11 | [ 0.33, 0.77] |  4.86 | < .001 |  1.00
## Avoid =~ ALose_v_Neut_R_Insula_run2   |        0.54 | 0.10 | [ 0.34, 0.73] |  5.33 | < .001 |  1.00
## Avoid =~ BLose_v_Neut_L_Insula_run1   |        0.50 | 0.10 | [ 0.30, 0.69] |  4.95 | < .001 |  1.00
## Avoid =~ BLose_v_Neut_R_Insula_run1   |        0.41 | 0.10 | [ 0.21, 0.61] |  3.99 | < .001 |  1.00
## Avoid =~ BLose_v_Neut_L_Insula_run2   |        0.23 | 0.09 | [ 0.05, 0.41] |  2.54 | 0.011  |  1.00
## Avoid =~ BLose_v_Neut_R_Insula_run2   |        0.46 | 0.11 | [ 0.24, 0.68] |  4.11 | < .001 |  1.00
## Avoid =~ BLose_v_BWin_L_Insula_run1   |        0.34 | 0.10 | [ 0.15, 0.53] |  3.52 | < .001 |  1.00
## Avoid =~ BLose_v_BWin_R_Insula_run1   |        0.37 | 0.09 | [ 0.19, 0.56] |  3.98 | < .001 |  1.00
## Avoid =~ BLose_v_BWin_L_Insula_run2   |        0.41 | 0.12 | [ 0.17, 0.64] |  3.38 | < .001 |  1.00
## Avoid =~ BLose_v_BWin_R_Insula_run2   |        0.37 | 0.10 | [ 0.17, 0.57] |  3.61 | < .001 |  1.00
## Approach =~ AWin_v_Neut_L_NAcc_run1   |        0.42 | 0.10 | [ 0.22, 0.61] |  4.23 | < .001 |  2.00
## Approach =~ AWin_v_Neut_R_NAcc_run1   |        0.54 | 0.07 | [ 0.40, 0.69] |  7.43 | < .001 |  2.00
## Approach =~ AWin_v_Neut_R_Insula_run1 |        0.15 | 0.08 | [-0.01, 0.31] |  1.89 | 0.059  |  2.00
## Approach =~ AWin_v_Neut_L_NAcc_run2   |        0.51 | 0.08 | [ 0.36, 0.67] |  6.55 | < .001 |  2.00
## Approach =~ AWin_v_Neut_R_NAcc_run2   |        0.41 | 0.09 | [ 0.23, 0.59] |  4.52 | < .001 |  2.00
## Approach =~ AWin_v_Neut_R_Insula_run2 |    2.85e-03 | 0.09 | [-0.18, 0.18] |  0.03 | 0.975  |  2.00
## Approach =~ BWin_v_Neut_L_NAcc_run1   |        0.40 | 0.11 | [ 0.19, 0.61] |  3.73 | < .001 |  2.00
## Approach =~ BWin_v_Neut_R_NAcc_run1   |        0.32 | 0.10 | [ 0.11, 0.52] |  3.04 | 0.002  |  2.00
## Approach =~ BWin_v_Neut_R_Insula_run1 |        0.23 | 0.10 | [ 0.03, 0.44] |  2.23 | 0.026  |  2.00
## Approach =~ BWin_v_Neut_L_NAcc_run2   |        0.31 | 0.10 | [ 0.12, 0.49] |  3.23 | 0.001  |  2.00
## Approach =~ BWin_v_Neut_R_NAcc_run2   |        0.33 | 0.09 | [ 0.16, 0.50] |  3.80 | < .001 |  2.00
## Approach =~ BWin_v_Neut_R_Insula_run2 |        0.27 | 0.10 | [ 0.08, 0.47] |  2.74 | 0.006  |  2.00
## Approach =~ BWin_v_BLose_L_NAcc_run1  |        0.37 | 0.10 | [ 0.17, 0.56] |  3.71 | < .001 |  2.00
## Approach =~ BWin_v_BLose_R_NAcc_run1  |        0.51 | 0.08 | [ 0.35, 0.67] |  6.29 | < .001 |  2.00
## Approach =~ BWin_v_BLose_L_NAcc_run2  |        0.44 | 0.09 | [ 0.27, 0.61] |  5.10 | < .001 |  2.00
## Approach =~ BWin_v_BLose_R_NAcc_run2  |        0.36 | 0.09 | [ 0.19, 0.54] |  3.99 | < .001 |  2.00
## Avoid =~ ALose_v_Neut_L_Insula_run1  |        0.51 | 0.08 | [ 0.34, 0.67] |  6.14 | < .001 |  2.00
## Avoid =~ ALose_v_Neut_R_Insula_run1  |        0.37 | 0.09 | [ 0.20, 0.53] |  4.24 | < .001 |  2.00
## Avoid =~ ALose_v_Neut_L_Insula_run2  |        0.39 | 0.09 | [ 0.22, 0.56] |  4.38 | < .001 |  2.00
## Avoid =~ ALose_v_Neut_R_Insula_run2  |        0.51 | 0.07 | [ 0.38, 0.64] |  7.65 | < .001 |  2.00
## Avoid =~ BLose_v_Neut_L_Insula_run1  |        0.35 | 0.08 | [ 0.19, 0.51] |  4.24 | < .001 |  2.00
## Avoid =~ BLose_v_Neut_R_Insula_run1  |        0.50 | 0.07 | [ 0.35, 0.64] |  6.64 | < .001 |  2.00
## Avoid =~ BLose_v_Neut_L_Insula_run2  |        0.44 | 0.07 | [ 0.30, 0.58] |  6.24 | < .001 |  2.00
## Avoid =~ BLose_v_Neut_R_Insula_run2  |        0.47 | 0.09 | [ 0.31, 0.64] |  5.55 | < .001 |  2.00
## Avoid =~ BLose_v_BWin_L_Insula_run1  |        0.61 | 0.08 | [ 0.46, 0.76] |  7.97 | < .001 |  2.00
## Avoid =~ BLose_v_BWin_R_Insula_run1  |        0.48 | 0.07 | [ 0.33, 0.62] |  6.40 | < .001 |  2.00
## Avoid =~ BLose_v_BWin_L_Insula_run2  |        0.63 | 0.06 | [ 0.50, 0.75] |  9.85 | < .001 |  2.00
## Avoid =~ BLose_v_BWin_R_Insula_run2  |        0.44 | 0.08 | [ 0.29, 0.59] |  5.74 | < .001 |  2.00
## Approach =~ AWin_v_Neut_L_NAcc_run1   |        0.45 | 0.03 | [ 0.38, 0.51] | 13.11 | < .001 |  3.00
## Approach =~ AWin_v_Neut_R_NAcc_run1   |        0.43 | 0.03 | [ 0.37, 0.50] | 13.35 | < .001 |  3.00
## Approach =~ AWin_v_Neut_R_Insula_run1 |        0.24 | 0.04 | [ 0.17, 0.31] |  6.69 | < .001 |  3.00
## Approach =~ AWin_v_Neut_L_NAcc_run2   |        0.48 | 0.03 | [ 0.42, 0.55] | 14.87 | < .001 |  3.00
## Approach =~ AWin_v_Neut_R_NAcc_run2   |        0.49 | 0.03 | [ 0.44, 0.55] | 17.00 | < .001 |  3.00
## Approach =~ AWin_v_Neut_R_Insula_run2 |        0.27 | 0.03 | [ 0.20, 0.34] |  7.91 | < .001 |  3.00
## Approach =~ BWin_v_Neut_L_NAcc_run1   |        0.51 | 0.03 | [ 0.45, 0.56] | 16.98 | < .001 |  3.00
## Approach =~ BWin_v_Neut_R_NAcc_run1   |        0.39 | 0.03 | [ 0.32, 0.45] | 12.10 | < .001 |  3.00
## Approach =~ BWin_v_Neut_R_Insula_run1 |        0.24 | 0.03 | [ 0.17, 0.31] |  6.99 | < .001 |  3.00
## Approach =~ BWin_v_Neut_L_NAcc_run2   |        0.42 | 0.03 | [ 0.36, 0.48] | 13.20 | < .001 |  3.00
## Approach =~ BWin_v_Neut_R_NAcc_run2   |        0.42 | 0.03 | [ 0.36, 0.48] | 13.55 | < .001 |  3.00
## Approach =~ BWin_v_Neut_R_Insula_run2 |        0.27 | 0.04 | [ 0.20, 0.35] |  7.45 | < .001 |  3.00
## Approach =~ BWin_v_BLose_L_NAcc_run1  |        0.47 | 0.03 | [ 0.41, 0.53] | 15.00 | < .001 |  3.00
## Approach =~ BWin_v_BLose_R_NAcc_run1  |        0.46 | 0.03 | [ 0.40, 0.53] | 14.42 | < .001 |  3.00
## Approach =~ BWin_v_BLose_L_NAcc_run2  |        0.42 | 0.03 | [ 0.36, 0.48] | 12.84 | < .001 |  3.00
## Approach =~ BWin_v_BLose_R_NAcc_run2  |        0.48 | 0.03 | [ 0.42, 0.55] | 14.99 | < .001 |  3.00
## Avoid =~ ALose_v_Neut_L_Insula_run1  |        0.44 | 0.03 | [ 0.38, 0.51] | 12.82 | < .001 |  3.00
## Avoid =~ ALose_v_Neut_R_Insula_run1  |        0.50 | 0.03 | [ 0.43, 0.56] | 15.48 | < .001 |  3.00
## Avoid =~ ALose_v_Neut_L_Insula_run2  |        0.41 | 0.03 | [ 0.34, 0.47] | 12.44 | < .001 |  3.00
## Avoid =~ ALose_v_Neut_R_Insula_run2  |        0.46 | 0.03 | [ 0.39, 0.52] | 14.14 | < .001 |  3.00
## Avoid =~ BLose_v_Neut_L_Insula_run1  |        0.44 | 0.03 | [ 0.38, 0.51] | 12.90 | < .001 |  3.00
## Avoid =~ BLose_v_Neut_R_Insula_run1  |        0.51 | 0.03 | [ 0.44, 0.57] | 14.46 | < .001 |  3.00
## Avoid =~ BLose_v_Neut_L_Insula_run2  |        0.43 | 0.04 | [ 0.36, 0.50] | 11.76 | < .001 |  3.00
## Avoid =~ BLose_v_Neut_R_Insula_run2  |        0.48 | 0.03 | [ 0.42, 0.55] | 14.05 | < .001 |  3.00
## Avoid =~ BLose_v_BWin_L_Insula_run1  |        0.44 | 0.03 | [ 0.37, 0.51] | 12.62 | < .001 |  3.00
## Avoid =~ BLose_v_BWin_R_Insula_run1  |        0.23 | 0.04 | [ 0.16, 0.30] |  6.18 | < .001 |  3.00
## Avoid =~ BLose_v_BWin_L_Insula_run2  |        0.42 | 0.04 | [ 0.35, 0.49] | 11.89 | < .001 |  3.00
## Avoid =~ BLose_v_BWin_R_Insula_run2  |        0.23 | 0.04 | [ 0.16, 0.31] |  6.25 | < .001 |  3.00
## 
## # Correlation
## 
## Link              | Coefficient |   SE |         95% CI |      z |      p | Group
## ---------------------------------------------------------------------------------
## Approach ~~ Avoid |       -0.42 | 0.12 | [-0.65, -0.19] |  -3.63 | < .001 |  1.00
## Approach ~~ Avoid |       -0.54 | 0.09 | [-0.71, -0.36] |  -6.01 | < .001 |  2.00
## Approach ~~ Avoid |       -0.57 | 0.03 | [-0.63, -0.51] | -17.66 | < .001 |  3.00
```

Metric CFA model

```
##### Summarizing Metric Multi-group CFA model #####
parameters(metric_cfa, standardize = T)
```

```
## # Loading
##
## Link                                          | Coefficient |   SE |       95% CI |     z |      p |
## ----------------------------------------------------------------------------------------------------
## Approach =~ AWin_v_Neut_L_NAcc_run1 (.p1.)    |        0.43 | 0.05 | [0.33, 0.53] |  8.39 | < .001 |
## Approach =~ AWin_v_Neut_R_NAcc_run1 (.p2.)    |        0.43 | 0.06 | [0.32, 0.54] |  7.56 | < .001 |
## Approach =~ AWin_v_Neut_R_Insula_run1 (.p3.)  |        0.20 | 0.04 | [0.13, 0.27] |  5.63 | < .001 |
## Approach =~ AWin_v_Neut_L_NAcc_run2 (.p4.)    |        0.46 | 0.05 | [0.36, 0.56] |  9.09 | < .001 |
## Approach =~ AWin_v_Neut_R_NAcc_run2 (.p5.)    |        0.46 | 0.05 | [0.36, 0.55] |  9.65 | < .001 |
## Approach =~ AWin_v_Neut_R_Insula_run2 (.p6.)  |        0.23 | 0.04 | [0.14, 0.31] |  5.41 | < .001 |
## Approach =~ BWin_v_Neut_L_NAcc_run1 (.p7.)    |        0.43 | 0.05 | [0.34, 0.53] |  8.94 | < .001 |
## Approach =~ BWin_v_Neut_R_NAcc_run1 (.p8.)    |        0.36 | 0.05 | [0.26, 0.45] |  7.54 | < .001 |
## Approach =~ BWin_v_Neut_R_Insula_run1 (.p9.)  |        0.22 | 0.04 | [0.15, 0.30] |  5.99 | < .001 |
## Approach =~ BWin_v_Neut_L_NAcc_run2 (.p10.)   |        0.39 | 0.05 | [0.30, 0.48] |  8.24 | < .001 |
## Approach =~ BWin_v_Neut_R_NAcc_run2 (.p11.)   |        0.38 | 0.05 | [0.28, 0.47] |  7.77 | < .001 |
## Approach =~ BWin_v_Neut_R_Insula_run2 (.p12.) |        0.29 | 0.04 | [0.20, 0.38] |  6.53 | < .001 |
## Approach =~ BWin_v_BLose_L_NAcc_run1 (.p13.)  |        0.40 | 0.05 | [0.31, 0.49] |  8.48 | < .001 |
## Approach =~ BWin_v_BLose_R_NAcc_run1 (.p14.)  |        0.45 | 0.05 | [0.36, 0.55] |  9.18 | < .001 |
## Approach =~ BWin_v_BLose_L_NAcc_run2 (.p15.)  |        0.36 | 0.05 | [0.27, 0.45] |  7.80 | < .001 |
## Approach =~ BWin_v_BLose_R_NAcc_run2 (.p16.)  |        0.50 | 0.06 | [0.38, 0.61] |  8.40 | < .001 |
## Avoid =~ ALose_v_Neut_L_Insula_run1 (.p17.)   |        0.47 | 0.05 | [0.37, 0.57] |  9.59 | < .001 |
## Avoid =~ ALose_v_Neut_R_Insula_run1 (.p18.)   |        0.48 | 0.05 | [0.38, 0.58] |  9.33 | < .001 |
## Avoid =~ ALose_v_Neut_L_Insula_run2 (.p19.)   |        0.41 | 0.05 | [0.32, 0.51] |  8.48 | < .001 |
## Avoid =~ ALose_v_Neut_R_Insula_run2 (.p20.)   |        0.44 | 0.05 | [0.34, 0.54] |  8.57 | < .001 |
## Avoid =~ BLose_v_Neut_L_Insula_run1 (.p21.)   |        0.46 | 0.05 | [0.36, 0.55] |  9.63 | < .001 |
## Avoid =~ BLose_v_Neut_R_Insula_run1 (.p22.)   |        0.48 | 0.04 | [0.39, 0.57] | 10.63 | < .001 |
## Avoid =~ BLose_v_Neut_L_Insula_run2 (.p23.)   |        0.40 | 0.04 | [0.32, 0.49] |  9.31 | < .001 |
## Avoid =~ BLose_v_Neut_R_Insula_run2 (.p24.)   |        0.51 | 0.05 | [0.41, 0.61] | 10.19 | < .001 |
## Avoid =~ BLose_v_BWin_L_Insula_run1 (.p25.)   |        0.48 | 0.04 | [0.39, 0.56] | 11.05 | < .001 |
## Avoid =~ BLose_v_BWin_R_Insula_run1 (.p26.)   |        0.30 | 0.04 | [0.21, 0.39] |  6.78 | < .001 |
## Avoid =~ BLose_v_BWin_L_Insula_run2 (.p27.)   |        0.47 | 0.05 | [0.37, 0.56] |  9.97 | < .001 |
## Avoid =~ BLose_v_BWin_R_Insula_run2 (.p28.)   |        0.28 | 0.04 | [0.19, 0.36] |  6.44 | < .001 |
## Approach =~ AWin_v_Neut_L_NAcc_run1 (.p1.)    |        0.42 | 0.04 | [0.33, 0.50] |  9.65 | < .001 |
## Approach =~ AWin_v_Neut_R_NAcc_run1 (.p2.)    |        0.44 | 0.05 | [0.34, 0.53] |  9.03 | < .001 |
## Approach =~ AWin_v_Neut_R_Insula_run1 (.p3.)  |        0.22 | 0.03 | [0.15, 0.29] |  6.46 | < .001 |
## Approach =~ AWin_v_Neut_L_NAcc_run2 (.p4.)    |        0.44 | 0.04 | [0.35, 0.52] | 10.49 | < .001 |
## Approach =~ AWin_v_Neut_R_NAcc_run2 (.p5.)    |        0.44 | 0.04 | [0.37, 0.52] | 10.97 | < .001 |
## Approach =~ AWin_v_Neut_R_Insula_run2 (.p6.)  |        0.22 | 0.03 | [0.16, 0.28] |  7.38 | < .001 |
## Approach =~ BWin_v_Neut_L_NAcc_run1 (.p7.)    |        0.42 | 0.04 | [0.34, 0.51] | 10.00 | < .001 |
## Approach =~ BWin_v_Neut_R_NAcc_run1 (.p8.)    |        0.32 | 0.04 | [0.25, 0.40] |  8.44 | < .001 |
## Approach =~ BWin_v_Neut_R_Insula_run1 (.p9.)  |        0.20 | 0.03 | [0.14, 0.27] |  6.20 | < .001 |
## Approach =~ BWin_v_Neut_L_NAcc_run2 (.p10.)   |        0.38 | 0.04 | [0.30, 0.45] |  9.46 | < .001 |
## Approach =~ BWin_v_Neut_R_NAcc_run2 (.p11.)   |        0.36 | 0.04 | [0.28, 0.44] |  9.06 | < .001 |
## Approach =~ BWin_v_Neut_R_Insula_run2 (.p12.) |        0.24 | 0.04 | [0.17, 0.32] |  6.51 | < .001 |
## Approach =~ BWin_v_BLose_L_NAcc_run1 (.p13.)  |        0.41 | 0.04 | [0.33, 0.48] | 10.32 | < .001 |
## Approach =~ BWin_v_BLose_R_NAcc_run1 (.p14.)  |        0.45 | 0.04 | [0.36, 0.54] | 10.14 | < .001 |
## Approach =~ BWin_v_BLose_L_NAcc_run2 (.p15.)  |        0.37 | 0.04 | [0.29, 0.45] |  9.01 | < .001 |
## Approach =~ BWin_v_BLose_R_NAcc_run2 (.p16.)  |        0.41 | 0.04 | [0.33, 0.49] | 10.34 | < .001 |
## Avoid =~ ALose_v_Neut_L_Insula_run1 (.p17.)   |        0.48 | 0.05 | [0.39, 0.57] | 10.43 | < .001 |
## Avoid =~ ALose_v_Neut_R_Insula_run1 (.p18.)   |        0.50 | 0.04 | [0.43, 0.58] | 13.51 | < .001 |
## Avoid =~ ALose_v_Neut_L_Insula_run2 (.p19.)   |        0.44 | 0.04 | [0.36, 0.53] | 10.74 | < .001 |
## Avoid =~ ALose_v_Neut_R_Insula_run2 (.p20.)   |        0.53 | 0.04 | [0.45, 0.61] | 12.91 | < .001 |
## Avoid =~ BLose_v_Neut_L_Insula_run1 (.p21.)   |        0.45 | 0.04 | [0.37, 0.53] | 11.26 | < .001 |
## Avoid =~ BLose_v_Neut_R_Insula_run1 (.p22.)   |        0.56 | 0.04 | [0.48, 0.64] | 13.66 | < .001 |
## Avoid =~ BLose_v_Neut_L_Insula_run2 (.p23.)   |        0.46 | 0.04 | [0.38, 0.55] | 10.64 | < .001 |
## Avoid =~ BLose_v_Neut_R_Insula_run2 (.p24.)   |        0.52 | 0.04 | [0.43, 0.60] | 12.04 | < .001 |
## Avoid =~ BLose_v_BWin_L_Insula_run1 (.p25.)   |        0.51 | 0.05 | [0.42, 0.60] | 10.92 | < .001 |
## Avoid =~ BLose_v_BWin_R_Insula_run1 (.p26.)   |        0.33 | 0.05 | [0.23, 0.43] |  6.59 | < .001 |
## Avoid =~ BLose_v_BWin_L_Insula_run2 (.p27.)   |        0.50 | 0.05 | [0.40, 0.59] | 10.01 | < .001 |
## Avoid =~ BLose_v_BWin_R_Insula_run2 (.p28.)   |        0.28 | 0.04 | [0.20, 0.36] |  6.93 | < .001 |
## Approach =~ AWin_v_Neut_L_NAcc_run1 (.p1.)    |        0.45 | 0.03 | [0.39, 0.51] | 14.42 | < .001 |
## Approach =~ AWin_v_Neut_R_NAcc_run1 (.p2.)    |        0.46 | 0.03 | [0.40, 0.51] | 15.97 | < .001 |
## Approach =~ AWin_v_Neut_R_Insula_run1 (.p3.)  |        0.23 | 0.03 | [0.16, 0.29] |  7.03 | < .001 |
## Approach =~ AWin_v_Neut_L_NAcc_run2 (.p4.)    |        0.49 | 0.03 | [0.43, 0.55] | 16.52 | < .001 |
## Approach =~ AWin_v_Neut_R_NAcc_run2 (.p5.)    |        0.48 | 0.03 | [0.43, 0.54] | 17.30 | < .001 |
## Approach =~ AWin_v_Neut_R_Insula_run2 (.p6.)  |        0.24 | 0.03 | [0.17, 0.30] |  7.16 | < .001 |
## Approach =~ BWin_v_Neut_L_NAcc_run1 (.p7.)    |        0.50 | 0.03 | [0.44, 0.55] | 17.20 | < .001 |
## Approach =~ BWin_v_Neut_R_NAcc_run1 (.p8.)    |        0.38 | 0.03 | [0.32, 0.44] | 12.64 | < .001 |
## Approach =~ BWin_v_Neut_R_Insula_run1 (.p9.)  |        0.24 | 0.03 | [0.17, 0.30] |  7.28 | < .001 |
## Approach =~ BWin_v_Neut_L_NAcc_run2 (.p10.)   |        0.41 | 0.03 | [0.35, 0.47] | 13.61 | < .001 |
## Approach =~ BWin_v_Neut_R_NAcc_run2 (.p11.)   |        0.42 | 0.03 | [0.37, 0.48] | 14.80 | < .001 |
## Approach =~ BWin_v_Neut_R_Insula_run2 (.p12.) |        0.27 | 0.03 | [0.21, 0.34] |  8.27 | < .001 |
## Approach =~ BWin_v_BLose_L_NAcc_run1 (.p13.)  |        0.46 | 0.03 | [0.40, 0.52] | 15.10 | < .001 |
## Approach =~ BWin_v_BLose_R_NAcc_run1 (.p14.)  |        0.47 | 0.03 | [0.42, 0.53] | 16.39 | < .001 |
## Approach =~ BWin_v_BLose_L_NAcc_run2 (.p15.)  |        0.43 | 0.03 | [0.37, 0.49] | 14.26 | < .001 |
## Approach =~ BWin_v_BLose_R_NAcc_run2 (.p16.)  |        0.49 | 0.03 | [0.43, 0.54] | 16.43 | < .001 |
## Avoid =~ ALose_v_Neut_L_Insula_run1 (.p17.)   |        0.46 | 0.03 | [0.40, 0.52] | 15.12 | < .001 |
## Avoid =~ ALose_v_Neut_R_Insula_run1 (.p18.)   |        0.47 | 0.03 | [0.41, 0.53] | 15.57 | < .001 |
## Avoid =~ ALose_v_Neut_L_Insula_run2 (.p19.)   |        0.42 | 0.03 | [0.36, 0.48] | 14.03 | < .001 |
## Avoid =~ ALose_v_Neut_R_Insula_run2 (.p20.)   |        0.46 | 0.03 | [0.40, 0.51] | 16.08 | < .001 |
## Avoid =~ BLose_v_Neut_L_Insula_run1 (.p21.)   |        0.43 | 0.03 | [0.37, 0.49] | 13.74 | < .001 |
## Avoid =~ BLose_v_Neut_R_Insula_run1 (.p22.)   |        0.49 | 0.03 | [0.42, 0.55] | 15.06 | < .001 |
## Avoid =~ BLose_v_Neut_L_Insula_run2 (.p23.)   |        0.40 | 0.03 | [0.34, 0.47] | 12.48 | < .001 |
## Avoid =~ BLose_v_Neut_R_Insula_run2 (.p24.)   |        0.47 | 0.03 | [0.41, 0.53] | 15.11 | < .001 |
## Avoid =~ BLose_v_BWin_L_Insula_run1 (.p25.)   |        0.44 | 0.03 | [0.38, 0.51] | 14.04 | < .001 |
## Avoid =~ BLose_v_BWin_R_Insula_run1 (.p26.)   |        0.27 | 0.03 | [0.21, 0.33] |  8.91 | < .001 |
## Avoid =~ BLose_v_BWin_L_Insula_run2 (.p27.)   |        0.44 | 0.03 | [0.38, 0.50] | 14.73 | < .001 |
## Avoid =~ BLose_v_BWin_R_Insula_run2 (.p28.)   |        0.27 | 0.03 | [0.21, 0.33] |  8.55 | < .001 |
##
## # Correlation
##
## Link              | Coefficient |   SE |         95% CI |      z |      p | Group
## ------------------------------------------------------------------------------
## Approach ~~ Avoid |       -0.40 | 0.11 | [-0.62, -0.18] |  -3.61 | < .001 |  1.00
## Approach ~~ Avoid |       -0.53 | 0.09 | [-0.71, -0.36] |  -6.03 | < .001 |  2.00
## Approach ~~ Avoid |       -0.57 | 0.03 | [-0.64, -0.51] | -17.75 | < .001 |  3.00
```

Comparing models w/ BIC/AIC (anova)

The below compares whether the complete data (across all three samples) in the `all_cfa` model is significantly improved by the configural invariance model. A significant value indicates that the configural model is significantly better than the full sample cfa.

```
anova(all_sample, configural_cfa)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                   Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## all_sample      349 -21672 -21235 2200.6
## configural_cfa 1047 -21482 -20170 2961.4     777.17     698    0.01964 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, anova is used to compare the model improve in AIC/BIC by between the configural and metric invariance. A significantly result in the anova would indicate a significant improvement of the metric model over the configural model.

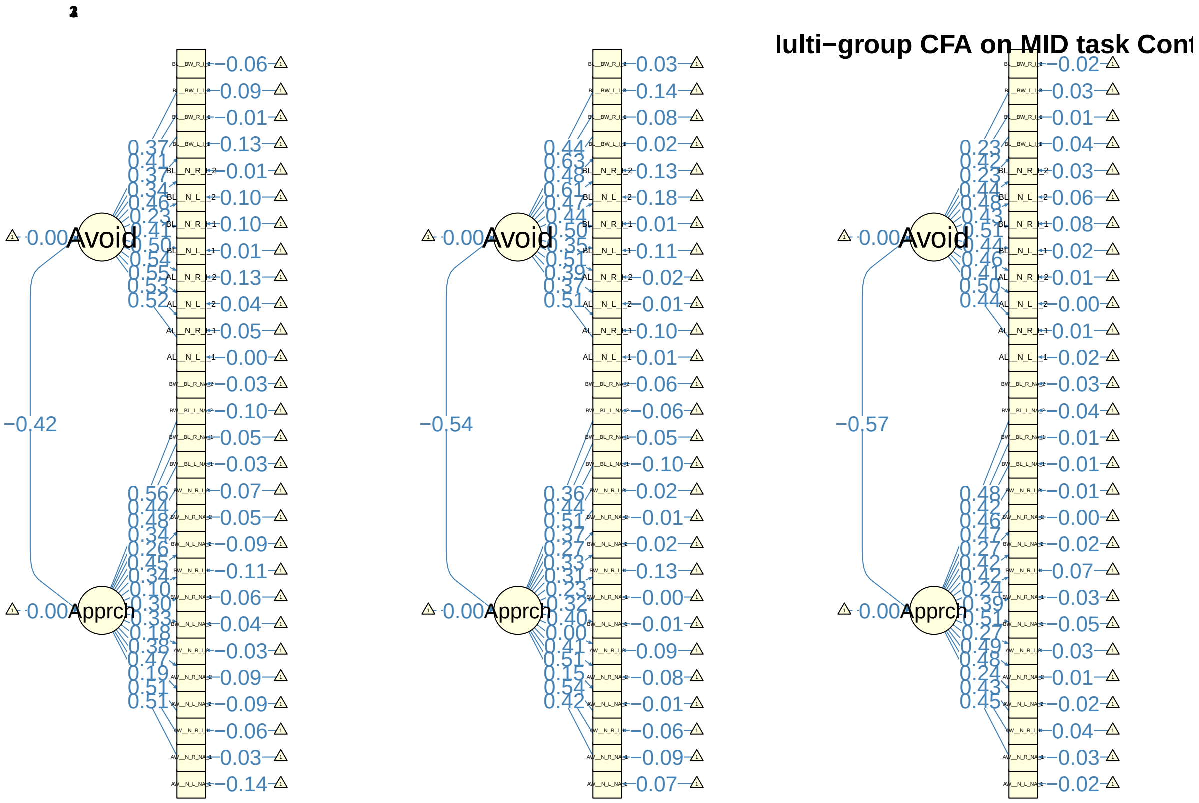
```
anova(configural_cfa, metric_cfa)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                   Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## configural_cfa 1047 -21482 -20170 2961.4
## metric_cfa     1099 -21523 -20479 3024.3     53.862      52      0.403
```

Plotting multi-group config. CFA

Use semPaths to plot the configural invariance CFA multigroup model

```
# this plottinig is not function with runs loading onto ROIs

layout(t(1:3))
semPaths(configural_cfa,
         color = "lightyellow",
         theme="colorblind",
         whatLabels = "std",
         style = "lisrel",
         sizeLat = 10,
         sizeLat2 = 10,
         sizeMan = 6,
         edge.color = "steelblue",
         edge.label.cex = 2,
         label.cex = 2,
         rotation = 2,
         layout = "tree2",
         intercepts = TRUE,
         residuals = FALSE,
         #residScale = 10,
         curve = 2,
         title = T,
         title.color = "black",
         cardinal = "lat cov",
         curvePivot = T,
         nCharNodes = 6,
         #nodeLabels = label,
         mar = c(2,5,2,6))
# Title
title("Multi-group CFA on MID task Contrasts")
```



Running [semi-restricted] ESEM Model

As described in the manuscript, the restricted CFA may incorrectly account for some measurement error in the items. This may degrade the fit statistics. See Marsh et al. (2014) for an in-depth discussion.

In this case, Exploratory Structural Equation Modeling (ESEM) is used to fit a CFA pre-specified model that allows for non-zero loadings. The technique and application of ESEM is available through the psych esem and esemcomp package. Here, the `esemcomp` package is used to fit a model using the steps described by Mateus Silvestrin here and by Guàrdia-Olmos et al.. The github code for esemcomp is available here. Below can be used to download the esemComp package – which worked with R version 4.2.1 on x86_64-apply-dawin17.0 during September 2022.

```
devtools::install_github("MateusPsi/esemComp", build_vignettes = TRUE)
```

Selected items for ESEM

First, select the items that are consistent with those in the CFA model

```
# ordering so can specify numerically
esem_data = brain_set[,c("AWin_v_Neut_L_NAcc_run1"  ,"AWin_v_Neut_L_NAcc_run2" ,
                         "BWin_v_Neut_L_NAcc_run1"  ,"BWin_v_Neut_L_NAcc_run2" ,
                         "BWin_v_BLose_L_NAcc_run1" ,"BWin_v_BLose_L_NAcc_run2",
                          "AWin_v_Neut_R_NAcc_run1" , "AWin_v_Neut_R_NAcc_run2",
                          "BWin_v_Neut_R_NAcc_run1" , "BWin_v_Neut_R_NAcc_run2",
                          "BWin_v_BLose_R_NAcc_run1", "BWin_v_BLose_R_NAcc_run2",
                         # insula values apprach
                         "AWin_v_Neut_R_Insula_run1","AWin_v_Neut_R_Insula_run2",
                         "BWin_v_Neut_R_Insula_run1","BWin_v_Neut_R_Insula_run2",
                         # avoidance
                         "ALose_v_Neut_L_Insula_run1","ALose_v_Neut_L_Insula_run2",
                         "BLose_v_Neut_L_Insula_run1","BLose_v_Neut_L_Insula_run2",
                         "BLose_v_BWin_L_Insula_run1","BLose_v_BWin_L_Insula_run2",
                         "ALose_v_Neut_R_Insula_run1","ALose_v_Neut_R_Insula_run2",
                         "BLose_v_Neut_R_Insula_run1","BLose_v_Neut_R_Insula_run2",
                         "BLose_v_BWin_R_Insula_run1","BLose_v_BWin_R_Insula_run2",
                         "set")]
```

Specify EFA Model

As described in March et al. (2014), create a target rotation for items onto factors. In this case two factors are specified by the CFA model, so factor 1 and factor 2 are specified in `make_target`.

```
target_rot <- make_target(28,mainloadings = list(f1 = 1:16, f2 = 17:28))
esem.efa <- esem_efa(data = esem_data[,1:28], nfactors = 2,
                     target = target_rot, fm = "ml")
```

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

```
esem.efa$loadings
```

```
##
## Loadings:
##                            ML1    ML2
## AWin_v_Neut_L_NAcc_run1     0.421
## AWin_v_Neut_L_NAcc_run2     0.459
## BWin_v_Neut_L_NAcc_run1     0.455
## BWin_v_Neut_L_NAcc_run2     0.414
## BWin_v_BLose_L_NAcc_run1    0.398
## BWin_v_BLose_L_NAcc_run2    0.364
## AWin_v_Neut_R_NAcc_run1     0.516
## AWin_v_Neut_R_NAcc_run2     0.564  0.112
## BWin_v_Neut_R_NAcc_run1     0.391
## BWin_v_Neut_R_NAcc_run2     0.419
## BWin_v_BLose_R_NAcc_run1    0.450
## BWin_v_BLose_R_NAcc_run2    0.461
## AWin_v_Neut_R_Insula_run1   0.204
## AWin_v_Neut_R_Insula_run2   0.226
## BWin_v_Neut_R_Insula_run1   0.245
## BWin_v_Neut_R_Insula_run2   0.255
## ALose_v_Neut_L_Insula_run1         0.470
## ALose_v_Neut_L_Insula_run2         0.435
## BLose_v_Neut_L_Insula_run1         0.450
## BLose_v_Neut_L_Insula_run2         0.431
## BLose_v_BWin_L_Insula_run1         0.462
## BLose_v_BWin_L_Insula_run2         0.451
## ALose_v_Neut_R_Insula_run1         0.466
## ALose_v_Neut_R_Insula_run2         0.427
## BLose_v_Neut_R_Insula_run1         0.502
## BLose_v_Neut_R_Insula_run2         0.489
## BLose_v_BWin_R_Insula_run1         0.228
## BLose_v_BWin_R_Insula_run2         0.215
##
##                  ML1   ML2
## SS loadings    2.616 2.243
## Proportion Var 0.093 0.080
## Cumulative Var 0.093 0.174
```

Using item that loads highest on factor 1 and lowest on factor 2 and vice versa, and define as anchor using `find_referents`

```
# per the example from Mateus Silverstrin, need to define anchor for each factor (value to loads higher
anchor <- find_referents(efa_object = esem.efa,factor_names = c("f1","f2"))
```

Once the esem efa and anchors are defined, use `syntax_composer` to specied the esem model. This will produce a lavaan specified model that references starting values that will be used in the cfa model

```
# Pull starting parameters
esem_mid_model <- syntax_composer(efa_object = esem.efa, referents = anchor)
```

Run ESEM model

Specified Model

The starting values are printed below to provide reference for how starting values differ from a strict CFA model. Notice, how some values that were original not fit onto the Approach factor (f1), such as big lose contrasts, they are now specified with loading values that are between .05 to -.05.

```
cat(esem_mid_model)
```

```
## f1 =~ start(0.421)*AWin_v_Neut_L_NAcc_run1+
## start(0.459)*AWin_v_Neut_L_NAcc_run2+
## start(0.455)*BWin_v_Neut_L_NAcc_run1+
## start(0.414)*BWin_v_Neut_L_NAcc_run2+
## start(0.398)*BWin_v_BLose_L_NAcc_run1+
## start(0.364)*BWin_v_BLose_L_NAcc_run2+
## start(0.516)*AWin_v_Neut_R_NAcc_run1+
## start(0.564)*AWin_v_Neut_R_NAcc_run2+
## start(0.391)*BWin_v_Neut_R_NAcc_run1+
## start(0.419)*BWin_v_Neut_R_NAcc_run2+
## start(0.45)*BWin_v_BLose_R_NAcc_run1+
## start(0.461)*BWin_v_BLose_R_NAcc_run2+
## start(0.204)*AWin_v_Neut_R_Insula_run1+
## start(0.226)*AWin_v_Neut_R_Insula_run2+
## start(0.245)*BWin_v_Neut_R_Insula_run1+
## start(0.255)*BWin_v_Neut_R_Insula_run2+
## start(0.012)*ALose_v_Neut_L_Insula_run1+
## start(0.016)*ALose_v_Neut_L_Insula_run2+
## start(0.012)*BLose_v_Neut_L_Insula_run1+
## start(0.025)*BLose_v_Neut_L_Insula_run2+
## start(0.01)*BLose_v_BWin_L_Insula_run1+
## start(0)*BLose_v_BWin_L_Insula_run2+
## start(-0.019)*ALose_v_Neut_R_Insula_run1+
## start(-0.054)*ALose_v_Neut_R_Insula_run2+
## 0.004*BLose_v_Neut_R_Insula_run1+
## start(0.009)*BLose_v_Neut_R_Insula_run2+
## start(-0.058)*BLose_v_BWin_R_Insula_run1+
## start(-0.073)*BLose_v_BWin_R_Insula_run2
##
## f2 =~ start(-0.034)*AWin_v_Neut_L_NAcc_run1+
## start(-0.034)*AWin_v_Neut_L_NAcc_run2+
## start(-0.037)*BWin_v_Neut_L_NAcc_run1+
## start(0.019)*BWin_v_Neut_L_NAcc_run2+
## start(-0.063)*BWin_v_BLose_L_NAcc_run1+
## start(-0.084)*BWin_v_BLose_L_NAcc_run2+
## start(0.079)*AWin_v_Neut_R_NAcc_run1+
## 0.112*AWin_v_Neut_R_NAcc_run2+
## start(0.03)*BWin_v_Neut_R_NAcc_run1+
## start(0.011)*BWin_v_Neut_R_NAcc_run2+
## start(-0.029)*BWin_v_BLose_R_NAcc_run1+
## start(-0.02)*BWin_v_BLose_R_NAcc_run2+
## start(-0.03)*AWin_v_Neut_R_Insula_run1+
## start(-0.012)*AWin_v_Neut_R_Insula_run2+
## start(0.021)*BWin_v_Neut_R_Insula_run1+
## start(-0.026)*BWin_v_Neut_R_Insula_run2+
## start(0.47)*ALose_v_Neut_L_Insula_run1+
## start(0.435)*ALose_v_Neut_L_Insula_run2+
## start(0.45)*BLose_v_Neut_L_Insula_run1+
## start(0.431)*BLose_v_Neut_L_Insula_run2+
## start(0.462)*BLose_v_BWin_L_Insula_run1+
## start(0.451)*BLose_v_BWin_L_Insula_run2+
## start(0.466)*ALose_v_Neut_R_Insula_run1+
## start(0.427)*ALose_v_Neut_R_Insula_run2+
## start(0.502)*BLose_v_Neut_R_Insula_run1+
## start(0.489)*BLose_v_Neut_R_Insula_run2+
## start(0.228)*BLose_v_BWin_R_Insula_run1+
## start(0.215)*BLose_v_BWin_R_Insula_run2
```

Running full ESEM model

After the EFA loadings are extracted using a target rotation, starting values are now available. These are now used to specify a less restrictive CFA model

```r
esem_mid_fit<- cfa(esem_mid_model, esem_data[,1:28], std.lv=TRUE, meanstructure = TRUE,
                   estimator = "MLR")
```

Pull and add fit statistics to the out dataframe and print results to see decreases in AIC/BIC

```r
# adding values to the CFA model fit indices
out[4,2:8] <- round(data.matrix(fitmeasures(esem_mid_fit,
                                            fit.measures = c("chisq","df","pvalue",
                                                             "rmsea", "cfi","tli", "srmr"))),
                    digits=3)
out[4,9] <- round(AIC(esem_mid_fit),3)
out[4,10] <- round(BIC(esem_mid_fit),3)
out[4,1] <-  c("Overall ESEM")

out <- as.data.frame(out)

out %>%
  knitr::kable(
    col.names = c("Model", "Chi-sq", "DF", "p value", "RMSEA", "CFI", "TLI","SRMR", "AIC", "BIC"),
    caption = "Fit statistics from MG-CFA and ESEM models",
    booktabs = TRUE
    )
```

Table 1: Fit statistics from MG-CFA and ESEM models

Model	Chi-sq	DF	p value	RMSEA	CFI	TLI	SRMR	AIC	BIC
Overall CFA	2200.622	349	0	0.065	0.657	0.628	0.053	-21672.085	-21234.81
Configg MG-CFA	2961.409	1047	0	0.066	0.65	0.621	0.06	-21482.229	-20170.405
Metric MG-CFA	3024.264	1099	0	0.064	0.648	0.637	0.062	-21523.374	-20479.059
Overall ESEM	2174.511	323	0	0.067	0.657	0.598	0.052	-21646.197	-21075.167

Aim: 2. Running EFA [Unrestricted] model

Here, a data-driven exploratory factor analysis is performed as implemented using the (https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/factanal)[https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/factanal] in the stats package. The same variables as in the CFA and ESEM dataset are used. A tutorial from Nilam Ram on EFA is also available here

By Sample EFA

Used the (factanal)[https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/factanal] to run EFA model. Specifying the number of factors and using the `promax` (non-orthogonal) rotation.

```
abcd_efa_df <- esem_data %>% filter(set == 1)
mls_efa_df <- esem_data %>% filter(set == 2)
ahrb_efa_df <- esem_data %>% filter(set == 3)
```

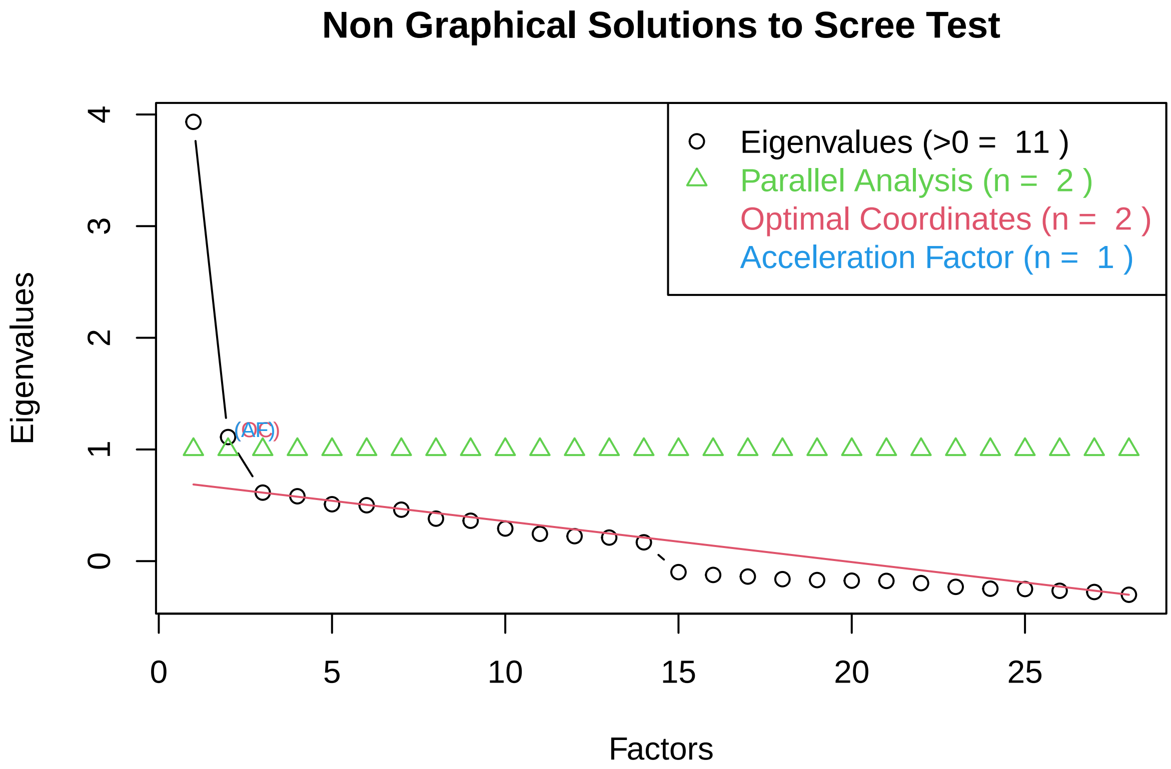
Sample: *ABCD*

A number of methods can be used to estimate the recommended data driven factors in the data. There are nuanced differences in calculations between packages and methods. Therefore, a number of packages are used to acquire consistent evidence to acquire the most parsimonious model.

Using nFactors package see the recommended factors for the EFA model using a number of models.

```
fa_abcd <- subset(abcd_efa_df[,1:28])
```

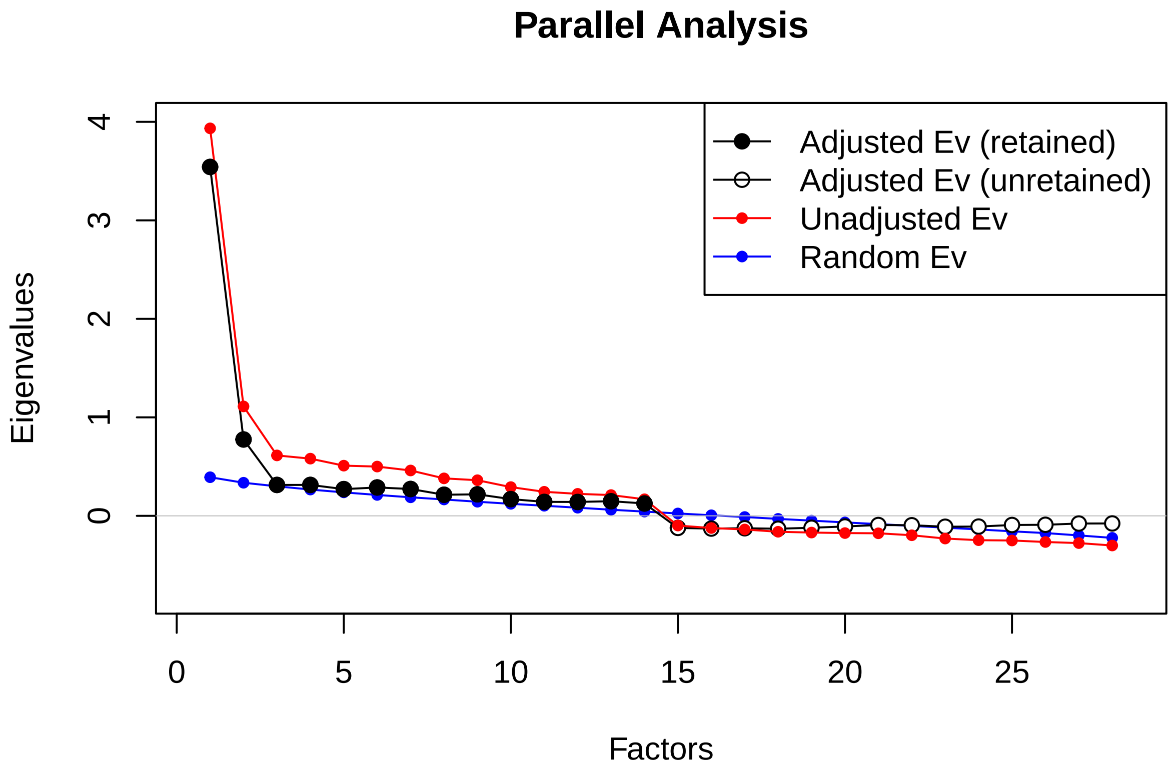
```
plot(nScree(x=fa_abcd,model="factors"))
```



To avoid biasing of packages in different calculations of recommendation factors that depend on strong correlations between bilateral regions by using parallel analysis. Parallel analysis is also implemented using the paran package.

```
paran(x = fa_abcd,
      iterations = 1000, quietly = FALSE, centile = 95,
      status = FALSE, all = TRUE, cfa = TRUE, graph = TRUE, color = TRUE,
      col = c("black", "red", "blue"), lty = c(1, 2, 3), lwd = 1, legend = TRUE,
      seed = 100)
```

```
##
## Using eigendecomposition of correlation matrix.
##
## Results of Horn's Parallel Analysis for factor retention
## 1000 iterations, using the 95 centile estimate
##
## --------------------------------------------------
## Factor      Adjusted    Unadjusted    Estimated
##             Eigenvalue  Eigenvalue    Bias
## --------------------------------------------------
## No components passed.
## --------------------------------------------------
## 1           3.542178    3.934015      0.391836
## 2           0.775202    1.111071      0.335869
## 3           0.313670    0.613506      0.299835
## 4           0.313864    0.580679      0.266815
## 5           0.271159    0.509467      0.238307
## 6           0.287997    0.500248      0.212250
## 7           0.272560    0.460549      0.187988
## 8           0.214122    0.380351      0.166228
## 9           0.218417    0.361595      0.143178
## 10          0.169955    0.291872      0.121916
## 11          0.141768    0.244225      0.102457
## 12          0.140901    0.223414      0.082512
## 13          0.148606    0.210788      0.062181
## 14          0.125450    0.168313      0.042863
## 15         -0.122634   -0.09868      0.023946
## 16         -0.129530   -0.12376      0.005761
## 17         -0.126362   -0.13831     -0.01194
## 18         -0.130790   -0.16166     -0.03087
## 19         -0.121125   -0.16974     -0.04861
## 20         -0.108212   -0.17507     -0.06685
## 21         -0.093136   -0.17720     -0.08406
## 22         -0.095347   -0.19714     -0.10179
## 23         -0.110512   -0.23053     -0.12002
## 24         -0.109017   -0.24716     -0.13814
## 25         -0.092890   -0.25014     -0.15725
## 26         -0.090723   -0.26576     -0.17504
## 27         -0.077950   -0.27668     -0.19873
## 28         -0.077263   -0.30109     -0.22383
## --------------------------------------------------
##
## Adjusted eigenvalues > 0 indicate dimensions to retain.
## (14 factors    retained)
```



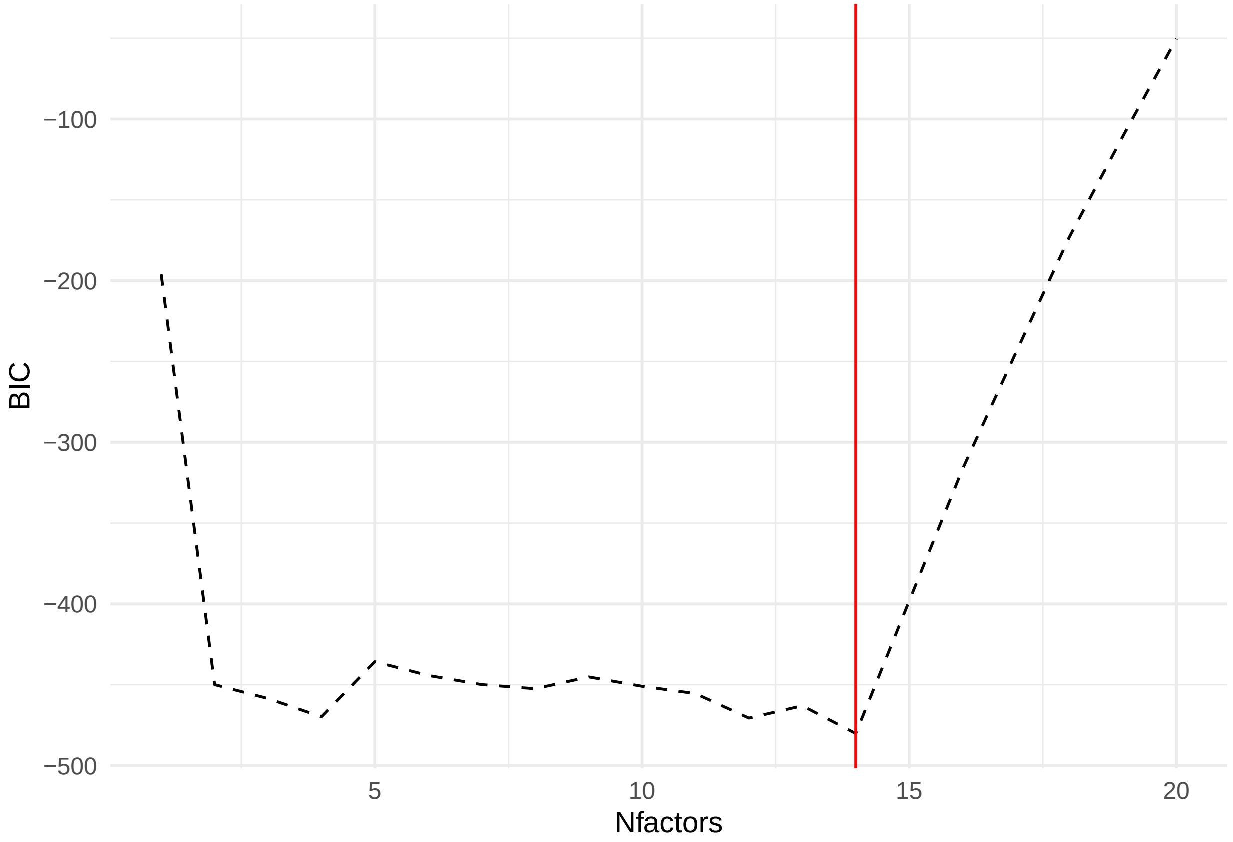
Comparing the above with the BIC comparison of an EFA model to determine the best fitting model based on fit statistics. Factor Analysis is submitted across a range of factors, e.g., 1-5, and the BIC is extracted from the model to determine the optimal number of factors

```
abcd_rec_factors <- matrix(NA, ncol = 2, nrow = 20)
colnames(abcd_rec_factors) <- c("Nfactors","BIC")

for (f in 1:20) {
  test_fac <- fa(r = fa_abcd,  #raw data
            nfactors = f, fm = 'minres',
            rotate = "oblimin")
  abcd_rec_factors[f,1] <- f
  abcd_rec_factors[f,2] <-test_fac$BIC
}
```

```
## Warning in GPFoblq(L, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.

## Warning in GPFoblq(L, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.

## Warning in GPFoblq(L, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.

## Warning in GPFoblq(L, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.
```

```
abcd_bic_fact = as.data.frame(abcd_rec_factors)
```

```
abcd_lowest_bic <- which.min(abcd_bic_fact$BIC)

abcd_bic_fact %>%
  ggplot(aes(x = Nfactors, y = BIC)) +
  geom_line(colour = 'black', linetype = 'dashed') +
  geom_vline(xintercept = abcd_bic_fact$Nfactors[abcd_lowest_bic], colour = 'red')+
  theme_minimal()
```



Running EFA

```
abcd_efa <- factanal(x = fa_abcd,  #raw data
              factors = 2, fm ='minres', rotation = "promax" # oblique rotation allow for non-orthogona
              )
```

```
heatmaply(round(abcd_efa$loadings[,1:2],2) %>% print(sort = T),
          scale_fill_gradient_fun = ggplot2::scale_fill_gradient2(
                 low = "blue",
                 high = "darkred",
                 space = "Lab",
                 midpoint = 0,
                 limits = c(-1, 1)
               ),
               dendrogram = "none",
               xlab = "", ylab = "",
               main = "",
```