

A Tutorial on Measurement Invariance Testing for Ordered-Categorical Items

Contents

Helper Function: <code>search_ninv()</code>	2
MI Testing With Ordered-Polytomous Items	3
Test of Configural Invariance	3
Test of Metric Invariance	4
Test of Scalar Invariance	6
Search of Noninvariant Thresholds	7
Test of Partial Scalar Invariance	9
Test of Partial Strict Invariance	10
Factor Mean Comparison	12
MI Testing With Dichotomous Items	12
Test of Configural Invariance	12
Test of Scalar Invariance	13
Test of Strict Invariance	15
Observed Mean Comparison	16
Factor Mean Comparison	16
Reference	17

Helper Function: `search_ninv()`

```
# Helper function for the search of noninvariant items using
# the chi-square difference test
search_ninv <- function(data, model, fit, group,
                        par = c("loadings", "thresholds", "unique.var")) {
  chi_diff <- rep(NA)
  # number of groups
  n_groups <- nrow(unique(subset(parameterestimates(fit), select = group)))
  # extract parameter labels in the syntax
  if (par == "loadings") {
    labels <- (subset(parameterestimates(fit), op == "=" & group == 1))$label
  } else if (par == "thresholds") {
    labels <- (subset(parameterestimates(fit), op == "|" & group == 1))$label
  } else if (par == "unique.var") {
    labels <- (subset(parameterestimates(fit), op == "~~" & group == 1))$label[-1]
  }
  # iteratively free a parameter and compare with the original model fit
  for (i in seq_len(length(labels))) {
    partial_mod <- gsub(
      paste0("c\\(",
        paste0(rep(labels[i], n_groups), collapse = ", "),
        "\\)"),
      "c(NA, NA)", model
    )
    partial_fit <- cfa(model = partial_mod,
                      data = data,
                      estimator = "WLSMV",
                      ordered = TRUE,
                      group = group,
                      parameterization = "theta")
    chi_diff[i] <- lavTestLRT(partial_fit, fit,
                            method = "satorra.2000")["Chisq diff"][[1]][2]
  }
  return(list(chi_diff = chi_diff,
             max_chi_diff = max(chi_diff),
             label = labels[which.max(chi_diff)]))
}
```

In this tutorial, we illustrate the measurement invariance (MI) testing procedure for ordered-polytomous and dichotomous items using *lavaan*. We follow the identification conditions described in Millsap and Yun-Tein (2004) and note that there are alternative identification conditions and test procedures (e.g., Wu and Estabrook 2016).

For replicability, we use the following syntax to import the data provided by Sharman et al. (2019) and select the variables of gender and the seven items in the Helpful subscale.

```
dat <- read.csv("https://osf.io/6gsy8/download")
dat_sub <- subset(dat, select = c(Gender, BACS_38, BACS_31,
                                BACS_29, BACS_30, BACS_1,
                                BACS_26, BACS_4))
```

MI Testing With Ordered-Polytomous Items

Here we provide the complete R codes for the test procedures described in the main text of the paper. For more details, please refer to the main text.

Test of Configural Invariance

We start with a configural model, which assesses the equality of the number and pattern of parameters between groups.

```
configural_mod <- "
## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(NA, NA) * BACS_31 +
     c(NA, NA) * BACS_29 + c(NA, NA) * BACS_30 + c(NA, NA) * BACS_1 +
     c(NA, NA) * BACS_26 + c(NA, NA) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_38 | c(t2, t2) * t2
BACS_38 | c(t3, t3a) * t3
BACS_38 | c(t4, t4a) * t4
BACS_31 | c(t5, t5a) * t1
BACS_31 | c(t6, t6) * t2
BACS_31 | c(t7, t7a) * t3
BACS_31 | c(t8, t8a) * t4
BACS_29 | c(t9, t9a) * t1
BACS_29 | c(t10, t10) * t2
BACS_29 | c(t11, t11a) * t3
BACS_29 | c(t12, t12a) * t4
BACS_30 | c(t13, t13a) * t1
BACS_30 | c(t14, t14) * t2
BACS_30 | c(t15, t15a) * t3
BACS_30 | c(t16, t16a) * t4
BACS_1 | c(t17, t17a) * t1
BACS_1 | c(t18, t18) * t2
BACS_1 | c(t19, t19a) * t3
```

```

BACS_1 | c(t20, t20a) * t4
BACS_26 | c(t21, t21a) * t1
BACS_26 | c(t22, t22) * t2
BACS_26 | c(t23, t23a) * t3
BACS_26 | c(t24, t24a) * t4
BACS_4 | c(t25, t25a) * t1
BACS_4 | c(t26, t26) * t2
BACS_4 | c(t27, t27a) * t3
BACS_4 | c(t28, t28a) * t4

## unique factor variances
BACS_38 ~~ c(1, NA) * BACS_38
BACS_31 ~~ c(1, NA) * BACS_31
BACS_29 ~~ c(1, NA) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1  ~~ c(1, NA) * BACS_1
BACS_26 ~~ c(1, NA) * BACS_26
BACS_4  ~~ c(1, NA) * BACS_4
"

```

```

configural_fit <- cfa(mod = configural_mod,
                      data = dat_sub,
                      estimator = "WLSMV",
                      ordered = TRUE,
                      group = "Gender",
                      parameterization = "theta")
summary(configural_fit)

```

The configural model has an acceptable fit, $\chi^2(28) = 56.56$, $p = .001$, RMSEA = 0.10, 95%CI [0.06, 0.14], CFI = 0.99, SRMR = 0.04.

Test of Metric Invariance

We then move on to assess metric invariance, which has the same identification constraints as the configural model, except that it has additional equality constraints on the loadings across groups.

```

metric_mod <- "
## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(12, 12) * BACS_31 +
      c(13, 13) * BACS_29 + c(14, 14) * BACS_30 + c(15, 15) * BACS_1 +
      c(16, 16) * BACS_26 + c(17, 17) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_38 | c(t2, t2) * t2
BACS_38 | c(t3, t3a) * t3
BACS_38 | c(t4, t4a) * t4
BACS_31 | c(t5, t5a) * t1

```

```

BACS_31 | c(t6, t6) * t2
BACS_31 | c(t7, t7a) * t3
BACS_31 | c(t8, t8a) * t4
BACS_29 | c(t9, t9a) * t1
BACS_29 | c(t10, t10) * t2
BACS_29 | c(t11, t11a) * t3
BACS_29 | c(t12, t12a) * t4
BACS_30 | c(t13, t13a) * t1
BACS_30 | c(t14, t14) * t2
BACS_30 | c(t15, t15a) * t3
BACS_30 | c(t16, t16a) * t4
BACS_1 | c(t17, t17a) * t1
BACS_1 | c(t18, t18) * t2
BACS_1 | c(t19, t19a) * t3
BACS_1 | c(t20, t20a) * t4
BACS_26 | c(t21, t21a) * t1
BACS_26 | c(t22, t22) * t2
BACS_26 | c(t23, t23a) * t3
BACS_26 | c(t24, t24a) * t4
BACS_4 | c(t25, t25a) * t1
BACS_4 | c(t26, t26) * t2
BACS_4 | c(t27, t27a) * t3
BACS_4 | c(t28, t28a) * t4

## unique factor variances
BACS_38 ~~ c(1, NA) * BACS_38
BACS_31 ~~ c(1, NA) * BACS_31
BACS_29 ~~ c(1, NA) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1  ~~ c(1, NA) * BACS_1
BACS_26 ~~ c(1, NA) * BACS_26
BACS_4  ~~ c(1, NA) * BACS_4
"

```

```

metric_fit <- cfa(mod = metric_mod,
  data = dat_sub,
  estimator = "WLSMV",
  ordered = TRUE,
  group = "Gender",
  parameterization = "theta")
summary(metric_fit)

```

To evaluate metric invariance, we compare the model fit of the configural model to the metric model using the chi-square difference test with the method proposed by Satorra and Bentler (2010).

```

lavTestLRT(configural_fit, metric_fit, "satorra.bentler.2010")

```

```

## Scaled Chi-Squared Difference Test (method = "satorra.2000")
##
## 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_fit 28           23.078
## metric_fit     34           30.089         10.4         6         0.1088
```

The metric model has an acceptable fit, $\chi^2(34) = 68.68$, $p < .001$, RMSEA = 0.10, 95%CI [0.06, 0.13], CFI = 0.99, SRMR = 0.04. The chi-square difference test is statistically nonsignificant, scaled $\Delta\chi^2(6) = 12.22$, $p = .057$, suggesting insufficient evidence that the loadings are noninvariant.

Test of Scalar Invariance

Next, we move on to the scalar model which further constrains thresholds to be equal between groups in addition to the constraints in the metric model.

```
scalar_mod <- "
## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(12, 12) * BACS_31 +
      c(13, 13) * BACS_29 + c(14, 14) * BACS_30 + c(15, 15) * BACS_1 +
      c(16, 16) * BACS_26 + c(17, 17) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_38 | c(t2, t2) * t2
BACS_38 | c(t3, t3) * t3
BACS_38 | c(t4, t4a) * t4
BACS_31 | c(t5, t5) * t1
BACS_31 | c(t6, t6) * t2
BACS_31 | c(t7, t7) * t3
BACS_31 | c(t8, t8) * t4
BACS_29 | c(t9, t9) * t1
BACS_29 | c(t10, t10) * t2
BACS_29 | c(t11, t11) * t3
BACS_29 | c(t12, t12) * t4
BACS_30 | c(t13, t13) * t1
BACS_30 | c(t14, t14) * t2
BACS_30 | c(t15, t15) * t3
BACS_30 | c(t16, t16) * t4
BACS_1 | c(t17, t17) * t1
BACS_1 | c(t18, t18) * t2
BACS_1 | c(t19, t19) * t3
BACS_1 | c(t20, t20) * t4
BACS_26 | c(t21, t21) * t1
BACS_26 | c(t22, t22) * t2
BACS_26 | c(t23, t23) * t3
BACS_26 | c(t24, t24) * t4
BACS_4 | c(t25, t25) * t1
BACS_4 | c(t26, t26) * t2
BACS_4 | c(t27, t27) * t3
BACS_4 | c(t28, t28) * t4
```

```
## unique factor variances
BACS_38 ~~ c(1, NA) * BACS_38
BACS_31 ~~ c(1, NA) * BACS_31
BACS_29 ~~ c(1, NA) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1  ~~ c(1, NA) * BACS_1
BACS_26 ~~ c(1, NA) * BACS_26
BACS_4  ~~ c(1, NA) * BACS_4
"
```

```
scalar_fit <- cfa(mod = scalar_mod,
  data = dat_sub,
  estimator = "WLSMV",
  ordered = TRUE,
  group = "Gender",
  parameterization = "theta")
```

```
lavTestLRT(metric_fit, scalar_fit, "satorra.bentler.2010")
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.2000")
##
## 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)
## metric_fit 34          30.089
## scalar_fit 53          59.380      32.924      19      0.02453 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The scalar model has fit $\chi^2(53) = 106.37$, $p < .001$, RMSEA = 0.10, 95%CI [0.07, 0.13], CFI = 0.98, SRMR = 0.04, and it is significantly different from the metric model, scaled $\Delta\chi^2(19) = 34.07$, $p = .018$, indicating some thresholds are noninvariant.

Search of Noninvariant Thresholds

Since the full threshold invariance fails, the unconstrained thresholds for all items need to be tested sequentially to identify the noninvariant threshold(s). `modindices()` in *lavaan* performs the search of noninvariant items with likelihood ratio test. A parameter is regarded as noninvariant if it has a modification index larger than 3.84, which is the quantile of the chi-square distribution with 1 degree of freedom at the .05 nominal level.

```
modindices(scalar_fit, free.remove = FALSE, op = "|", sort = TRUE)
```

```
##           lhs op rhs block group level      mi      epc sepc.lv sepc.all sepc.nox
## 80 BACS_30 | t1      2      2      1 5.860 -1.508 -1.508 -0.603 -0.603
## 93 BACS_4  | t2      2      2      1 3.199 0.427 0.427 0.289 0.289
## 91 BACS_26 | t4      2      2      1 2.970 0.774 0.774 0.429 0.429
```

## 89	BACS_26		t2	2	2	1	2.547	0.466	0.466	0.258	0.258
## 70	BACS_38		t3	2	2	1	1.915	-0.433	-0.433	-0.226	-0.226
## 87	BACS_1		t4	2	2	1	1.627	-0.317	-0.317	-0.224	-0.224
## 35	BACS_4		t2	1	1	1	1.599	-0.214	-0.214	-0.153	-0.153
## 22	BACS_30		t1	1	1	1	1.444	0.372	0.372	0.184	0.184
## 85	BACS_1		t2	2	2	1	1.407	0.291	0.291	0.206	0.206
## 27	BACS_1		t2	1	1	1	1.186	-0.245	-0.245	-0.162	-0.162
## 74	BACS_31		t3	2	2	1	1.093	-0.319	-0.319	-0.169	-0.169
## 31	BACS_26		t2	1	1	1	0.914	-0.167	-0.167	-0.113	-0.113
## 12	BACS_38		t3	1	1	1	0.807	0.182	0.182	0.092	0.092
## 79	BACS_29		t4	2	2	1	0.754	0.479	0.479	0.184	0.184
## 29	BACS_1		t4	1	1	1	0.684	0.133	0.133	0.088	0.088
## 76	BACS_29		t1	2	2	1	0.656	-0.417	-0.417	-0.160	-0.160
## 72	BACS_31		t1	2	2	1	0.584	-0.336	-0.336	-0.178	-0.178
## 73	BACS_31		t2	2	2	1	0.576	0.238	0.238	0.126	0.126
## 16	BACS_31		t3	1	1	1	0.467	0.136	0.136	0.071	0.071
## 33	BACS_26		t4	1	1	1	0.418	-0.109	-0.109	-0.074	-0.074
## 77	BACS_29		t2	2	2	1	0.404	0.269	0.269	0.103	0.103
## 82	BACS_30		t3	2	2	1	0.397	-0.256	-0.256	-0.102	-0.102
## 15	BACS_31		t2	1	1	1	0.359	-0.148	-0.148	-0.077	-0.077
## 14	BACS_31		t1	1	1	1	0.319	0.183	0.183	0.095	0.095
## 18	BACS_29		t1	1	1	1	0.313	0.199	0.199	0.086	0.086
## 75	BACS_31		t4	2	2	1	0.229	-0.174	-0.174	-0.092	-0.092
## 78	BACS_29		t3	2	2	1	0.199	-0.191	-0.191	-0.073	-0.073
## 86	BACS_1		t3	2	2	1	0.185	0.098	0.098	0.070	0.070
## 90	BACS_26		t3	2	2	1	0.176	0.124	0.124	0.069	0.069
## 19	BACS_29		t2	1	1	1	0.169	-0.112	-0.112	-0.048	-0.048
## 84	BACS_1		t1	2	2	1	0.164	-0.228	-0.228	-0.162	-0.162
## 21	BACS_29		t4	1	1	1	0.156	-0.099	-0.099	-0.042	-0.042
## 88	BACS_26		t1	2	2	1	0.104	0.104	0.104	0.058	0.058
## 24	BACS_30		t3	1	1	1	0.104	0.067	0.067	0.033	0.033
## 28	BACS_1		t3	1	1	1	0.094	-0.050	-0.050	-0.033	-0.033
## 26	BACS_1		t1	1	1	1	0.089	0.123	0.123	0.081	0.081
## 17	BACS_31		t4	1	1	1	0.075	0.057	0.057	0.030	0.030
## 83	BACS_30		t4	2	2	1	0.072	0.133	0.133	0.053	0.053
## 81	BACS_30		t2	2	2	1	0.069	-0.109	-0.109	-0.044	-0.044
## 20	BACS_29		t3	1	1	1	0.062	0.059	0.059	0.025	0.025
## 30	BACS_26		t1	1	1	1	0.051	-0.051	-0.051	-0.035	-0.035
## 32	BACS_26		t3	1	1	1	0.047	-0.033	-0.033	-0.022	-0.022
## 94	BACS_4		t3	2	2	1	0.043	0.049	0.049	0.034	0.034
## 95	BACS_4		t4	2	2	1	0.027	0.046	0.046	0.031	0.031
## 23	BACS_30		t2	1	1	1	0.024	0.038	0.038	0.019	0.019
## 69	BACS_38		t2	2	2	1	0.018	0.043	0.043	0.022	0.022
## 36	BACS_4		t3	1	1	1	0.016	-0.019	-0.019	-0.013	-0.013
## 25	BACS_30		t4	1	1	1	0.015	-0.027	-0.027	-0.014	-0.014
## 11	BACS_38		t2	1	1	1	0.010	-0.024	-0.024	-0.012	-0.012
## 37	BACS_4		t4	1	1	1	0.008	-0.013	-0.013	-0.009	-0.009
## 92	BACS_4		t1	2	2	1	0.001	-0.010	-0.010	-0.007	-0.007
## 34	BACS_4		t1	1	1	1	0.001	0.006	0.006	0.004	0.004
## 68	BACS_38		t1	2	2	1	0.001	-0.009	-0.009	-0.005	-0.005
## 10	BACS_38		t1	1	1	1	0.000	0.007	0.007	0.004	0.004
## 71	BACS_38		t4	2	2	1	0.000	0.000	0.000	0.000	0.000
## 13	BACS_38		t4	1	1	1	0.000	0.000	0.000	0.000	0.000

The modification index suggests that the first threshold of BACS_30 is noninvariant.

Another way to search for noninvariant parameters is with the chi-square difference test, which accounts for the ordered-categorical nature of the data (Svetina, Rutkowski, and Rutkowski 2019). However, this method is not directly applied in *lavaan*. We wrote a helper function, `search_ninv()` to iteratively search for noninvariant parameters. To search for noninvariant thresholds, we specify “par = thresholds” and provide the scalar model syntax and fit of the scalar model. This may take a few minutes to run as it fits a series of models that free a threshold at a time.

```
search_ninv(data = dat_sub,
            model = scalar_mod,
            fit = scalar_fit,
            group = "Gender",
            par = "thresholds")

## Warning in lavTestLRT(partial_fit, fit, method = "satorra.2000"): lavaan
## WARNING: some models have the same degrees of freedom

## $chi_diff
## [1] 0.2274904 0.2646986 5.4661144 0.0000000 1.6305240 2.0857183 3.4711318
## [8] 0.7717238 1.5236591 1.1388046 0.6611144 1.6315006 5.9639198 0.4140140
## [15] 0.9108997 0.4992836 0.4101160 3.7529060 0.5872871 3.2286381 0.4025511
## [22] 4.6106774 0.5704069 2.8111951 0.2156444 7.6042889 0.3288168 0.3706585
##
## $max_chi_diff
## [1] 7.604289
##
## $label
## [1] "t26"
```

The chi-square difference test suggests that the threshold with the label `t26`, the first threshold of BACS_30, is noninvariant. The result agrees with `modindices()`, but yields a different chi-square value, as also observed in Svetina, Rutkowski, and Rutkowski (2019).

Test of Partial Scalar Invariance

Next, we evaluate the fit of the partial scalar model that freely estimates the first threshold of BACS_30.

```
partial_scalar_mod <- "
## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(12, 12) * BACS_31 +
      c(13, 13) * BACS_29 + c(14, 14) * BACS_30 + c(15, 15) * BACS_1 +
      c(16, 16) * BACS_26 + c(17, 17) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_38 | c(t2, t2) * t2
BACS_38 | c(t3, t3) * t3
BACS_38 | c(t4, t4) * t4
```

```

BACS_31 | c(t5, t5) * t1
BACS_31 | c(t6, t6) * t2
BACS_31 | c(t7, t7) * t3
BACS_31 | c(t8, t8) * t4
BACS_29 | c(t9, t9) * t1
BACS_29 | c(t10, t10) * t2
BACS_29 | c(t11, t11) * t3
BACS_29 | c(t12, t12) * t4
BACS_30 | c(t13, t13a) * t1
BACS_30 | c(t14, t14) * t2
BACS_30 | c(t15, t15) * t3
BACS_30 | c(t16, t16) * t4
BACS_1 | c(t17, t17) * t1
BACS_1 | c(t18, t18) * t2
BACS_1 | c(t19, t19) * t3
BACS_1 | c(t20, t20) * t4
BACS_26 | c(t21, t21) * t1
BACS_26 | c(t22, t22) * t2
BACS_26 | c(t23, t23) * t3
BACS_26 | c(t24, t24) * t4
BACS_4 | c(t25, t25) * t1
BACS_4 | c(t26, t26) * t2
BACS_4 | c(t27, t27) * t3
BACS_4 | c(t28, t28) * t4

## unique factor variances
BACS_38 ~~ c(1, NA) * BACS_38
BACS_31 ~~ c(1, NA) * BACS_31
BACS_29 ~~ c(1, NA) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1  ~~ c(1, NA) * BACS_1
BACS_26 ~~ c(1, NA) * BACS_26
BACS_4  ~~ c(1, NA) * BACS_4
"

```

```

partial_scalar_fit <- cfa(model = partial_scalar_mod,
  data = dat_sub,
  estimator = "WLSMV",
  ordered = TRUE,
  group = "Gender",
  parameterization = "theta")

```

The partial scalar model has a fit $\chi^2(53) = 95.71$, $p < .001$, RMSEA = 0.09, 95%CI [0.06, 0.12], CFI = 0.99, SRMR = 0.04 and does not fit worse than the metric model, scaled $\Delta\chi^2(19) = 26.42$, $p = .119$. As such, we proceed to the next stage of invariance testing with this partial scalar model.

Test of Partial Strict Invariance

Finally, we evaluate the partial strict invariance model, which constrains the unique factor variances to be equal in the items that have invariant thresholds in the partial scalar model.

```

partial_strict_mod <- "
## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(12, 12) * BACS_31 +
      c(13, 13) * BACS_29 + c(14, 14) * BACS_30 + c(15, 15) * BACS_1 +
      c(16, 16) * BACS_26 + c(17, 17) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_38 | c(t2, t2) * t2
BACS_38 | c(t3, t3) * t3
BACS_38 | c(t4, t4) * t4
BACS_31 | c(t5, t5) * t1
BACS_31 | c(t6, t6) * t2
BACS_31 | c(t7, t7) * t3
BACS_31 | c(t8, t8) * t4
BACS_29 | c(t9, t9) * t1
BACS_29 | c(t10, t10) * t2
BACS_29 | c(t11, t11) * t3
BACS_29 | c(t12, t12) * t4
BACS_30 | c(t13, t13a) * t1
BACS_30 | c(t14, t14) * t2
BACS_30 | c(t15, t15) * t3
BACS_30 | c(t16, t16) * t4
BACS_1 | c(t17, t17) * t1
BACS_1 | c(t18, t18) * t2
BACS_1 | c(t19, t19) * t3
BACS_1 | c(t20, t20) * t4
BACS_26 | c(t21, t21) * t1
BACS_26 | c(t22, t22) * t2
BACS_26 | c(t23, t23) * t3
BACS_26 | c(t24, t24) * t4
BACS_4 | c(t25, t25) * t1
BACS_4 | c(t26, t26) * t2
BACS_4 | c(t27, t27) * t3
BACS_4 | c(t28, t28) * t4

## unique factor variances
BACS_38 ~~ c(1, 1) * BACS_38
BACS_31 ~~ c(1, 1) * BACS_31
BACS_29 ~~ c(1, 1) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1  ~~ c(1, 1) * BACS_1
BACS_26 ~~ c(1, 1) * BACS_26
BACS_4  ~~ c(1, 1) * BACS_4
"

partial_strict_fit <- cfa(model = partial_strict_mod,
                          data = dat_sub,
                          estimator = "WLSMV",

```

```
ordered = TRUE,
group = "Gender",
parameterization = "theta")
```

The partial strict model has similar fit as the partial scalar model, scaled $\Delta\chi^2(6) = 11.06$, $p = .087$. Thus, the final model is a partial strict invariant model with the first threshold and unique factor variance of the item BACS_30 freed.

Factor Mean Comparison

We can obtain the parameter estimates in the partial scalar and partial strict models using `parameterestimates()`. Below we subset the details of the estimated factor mean difference (i.e., the estimated factor mean of the focal group as the factor mean of the reference group is set to 0).

```
subset(parameterestimates(partial_scalar_fit),
  group == 2 & lhs == "f" & op == "~1")
```

##	lhs	op	rhs	block	group	label	est	se	z	pvalue	ci.lower	ci.upper
## 66	f	~1	2	2			-0.634	0.181	-3.5	0	-0.988	-0.279

```
subset(parameterestimates(partial_strict_fit),
  group == 2 & lhs == "f" & op == "~1")
```

##	lhs	op	rhs	block	group	label	est	se	z	pvalue	ci.lower	ci.upper
## 66	f	~1	2	2			-0.64	0.18	-3.546	0	-0.993	-0.286

The factor mean difference is statistically significant in both the partial scalar model, $-.63$, 95% CI $[-0.99, -0.29]$, and the partial strict model, $-.64$, 95% CI $[-0.99, -0.28]$. Note that for ordered-polytomous items, factor mean comparisons are valid in both the scalar/partial scalar and strict/partial strict models.

MI Testing With Dichotomous Items

In the following, we illustrate the MI testing procedure for dichotomous items using *lavaan*. We note that, with binary outcomes, past research recommended skipping metric model and going directly from configural to scalar model (see Muthén 2002; Putnick and Bornstein 2016; Wu and Estabrook 2016 for more details), and hence only three models should be tested, including the strict model.

To create a dichotomous version of the Helpful subscale, we dichotomize the ordered-polytomous items by relabeling the original responses less than 3 as 0, and the responses equal to or higher than 3 as 1.

```
dat_dich <- dat_sub %>%
  mutate_at(vars(BACS_38:BACS_4), ~ if_else(. < 3, 0, 1))
items <- c("BACS_38", "BACS_31", "BACS_29", "BACS_30", "BACS_1", "BACS_26", "BACS_4")
```

Test of Configural Invariance

We begin with choosing item BACS_38 as the marker variable and constrain its loading to be equal between groups. Note we fix the latent factor variance of the first group to be 1 and allow the loading of the marker variable to freely vary. Moreover, all thresholds are constrained to be equal for identification purposes. As

such, for dichotomous items, the configural model evaluates the equality of the number and patterns of parameters, as well as threshold invariance. An inadequate fit of the configural model would indicate a violation of configural invariance and threshold invariance. We refer readers to Millsap and Yun-Tein (2004) for details on identification conditions for dichotomous items.

```
configural_mod_d <-
"## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(NA, NA) * BACS_31 +
      c(NA, NA) * BACS_29 + c(NA, NA) * BACS_30 + c(NA, NA) * BACS_1 +
      c(NA, NA) * BACS_26 + c(NA, NA) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_31 | c(t2, t2) * t1
BACS_29 | c(t3, t3) * t1
BACS_30 | c(t4, t4) * t1
BACS_1 | c(t5, t5) * t1
BACS_26 | c(t6, t6) * t1
BACS_4 | c(t7, t7) * t1

## unique factor variances
BACS_38 ~~ c(1, 1) * BACS_38
BACS_31 ~~ c(1, NA) * BACS_31
BACS_29 ~~ c(1, NA) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1  ~~ c(1, NA) * BACS_1
BACS_26 ~~ c(1, NA) * BACS_26
BACS_4  ~~ c(1, NA) * BACS_4"
```

```
configural_fit_d <- cfa(model = configural_mod_d,
                        data = dat_dich,
                        estimator = "WLSMV",
                        ordered = items,
                        group = "Gender",
                        parameterization = "theta")
summary(configural_fit_d, fit.measures = TRUE)
```

The configural model has an acceptable fit, $\chi^2(28) = 36.47$, $p = .131$, RMSEA = 0.05, 95%CI [0.00, 0.10], CFI = 0.99, SRMR = 0.08, supporting configural and threshold invariance.

Test of Scalar Invariance

Next, the scalar invariance model is fit to data by setting equality constraints on all loadings across groups, in addition to the identification constraints set in the configural model.

```
scalar_mod_d <- "
## loadings
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(12, 12) * BACS_31 +
      c(13, 13) * BACS_29 + c(14, 14) * BACS_30 + c(15, 15) * BACS_1 +
```

```

      c(16, 16) * BACS_26 + c(17, 17) * BACS_4

## common factor
f ~ c(0, NA) * 1
f ~~ c(1, NA) * f

## thresholds
BACS_38 | c(t1, t1) * t1
BACS_31 | c(t2, t2) * t1
BACS_29 | c(t3, t3) * t1
BACS_30 | c(t4, t4) * t1
BACS_1 | c(t5, t5) * t1
BACS_26 | c(t6, t6) * t1
BACS_4 | c(t7, t7) * t1

## unique factor variances
BACS_38 ~~ c(1, 1) * BACS_38
BACS_31 ~~ c(1, NA) * BACS_31
BACS_29 ~~ c(1, NA) * BACS_29
BACS_30 ~~ c(1, NA) * BACS_30
BACS_1 ~~ c(1, NA) * BACS_1
BACS_26 ~~ c(1, NA) * BACS_26
BACS_4 ~~ c(1, NA) * BACS_4
"

scalar_fit_d <- cfa(model = scalar_mod_d,
                    data = dat_dich,
                    estimator = "WLSMV",
                    ordered = TRUE,
                    group = "Gender",
                    parameterization = "theta")
summary(scalar_fit_d, fit.measures = TRUE)

```

To evaluate scalar invariance, we compare the model fit of the configural model to the scalar model using the chi-square difference test with the method proposed by Satorra and Bentler (2010).

```

lavTestLRT(configural_fit_d, scalar_fit_d,
            method = "satorra.bentler.2010")

## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2010")
##
## 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_fit_d 28      22.231
## scalar_fit_d     34      27.627      8.0499      6      0.2345

```

The scalar invariance model does not fit worse than the configural model, scaled $\Delta\chi^2(6) = 8.05$, $p = .234$. It has acceptable fit $\chi^2(34) = 44.82$, $p = .102$, RMSEA = 0.06, 95%CI [0.00, 0.10], CFI = 0.99, SRMR = 0.08.

Test of Strict Invariance

Lastly, we fit a strict invariance model which further constrains the unique factor variance for all items across groups.

```
strict_mod_d <- "  
## loadings  
f =~ c(NA, NA) * BACS_38 + c(11, 11) * BACS_38 + c(12, 12) * BACS_31 +  
      c(13, 13) * BACS_29 + c(14, 14) * BACS_30 + c(15, 15) * BACS_1 +  
      c(16, 16) * BACS_26 + c(17, 17) * BACS_4  
  
## common factor  
f ~ c(0, NA) * 1  
f ~~ c(1, NA) * f  
  
## thresholds  
BACS_38 | c(t1, t1) * t1  
BACS_31 | c(t2, t2) * t1  
BACS_29 | c(t3, t3) * t1  
BACS_30 | c(t4, t4) * t1  
BACS_1 | c(t5, t5) * t1  
BACS_26 | c(t6, t6) * t1  
BACS_4 | c(t7, t7) * t1  
  
## unique factor variances  
BACS_38 ~~ c(1, 1) * BACS_38  
BACS_31 ~~ c(1, 1) * BACS_31  
BACS_29 ~~ c(1, 1) * BACS_29  
BACS_30 ~~ c(1, 1) * BACS_30  
BACS_1  ~~ c(1, 1) * BACS_1  
BACS_26 ~~ c(1, 1) * BACS_26  
BACS_4  ~~ c(1, 1) * BACS_4  
"
```

```
strict_fit_d <- cfa(model = strict_mod_d,  
                    data = dat_dich,  
                    estimator = "WLSMV",  
                    ordered = TRUE,  
                    group = "Gender",  
                    parameterization = "theta")  
summary(strict_fit_d, fit.measures = TRUE)
```

Similarly, we compare the strict model with the scalar model using chi-square difference test to evaluate whether strict invariance holds.

```
lavTestLRT(scalar_fit_d, strict_fit_d,  
           method = "satorra.bentler.2010")
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2010")  
##  
## 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)
## scalar_fit_d 34      27.627
## strict_fit_d 40      29.872      1.6314      6      0.9503
```

The strict invariance model has a similar fit as the scalar invariance model, scaled $\Delta\chi^2(6) = 1.63$, $p = .950$. The fit of the strict model is good, $\chi^2(40) = 43.34$, $p = .331$, RMSEA = 0.03, 95%CI [0.00, 0.08], CFI = 1.00, SRMR = 0.08. Therefore, we conclude that the dichotomous version of the Helpful subscale of BACS is strict invariant.

Observed Mean Comparison

As strict invariance holds, we can perform valid group comparisons with the observed scores.

```
# create sum scores of the seven items
reference_sum <- rowSums(subset(dat_dich, Gender == 2,
                               select = - Gender))
focal_sum <- rowSums(subset(dat_dich, Gender == 1,
                            select = - Gender))
```

```
(t <- t.test(reference_sum, focal_sum))
```

```
##
## Welch Two Sample t-test
##
## data: reference_sum and focal_sum
## t = 4.5997, df = 86.217, p-value = 1.442e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.9293401 2.3439933
## sample estimates:
## mean of x mean of y
##  5.886667  4.250000
```

The Welch independent sample t-test suggests that the Helpful beliefs about crying significantly differ between gender, -1.64, 95% CI [0.93, 2.34].

Factor Mean Comparison

We can obtain the estimate of factor mean difference using the following syntax.

```
subset(parameterestimates(scalar_fit_d),
       group == 2 & lhs == "f" & op == "~1")
```

```
##   lhs op rhs block group label    est    se      z pvalue ci.lower ci.upper
## 45   f ~1      2      2    -0.849 0.201 -4.222      0   -1.243   -0.455
```

```
subset(parameterestimates(strict_fit_d),
       group == 2 & lhs == "f" & op == "~1")
```


##	lhs	op	rhs	block	group	label	est	se	z	pvalue	ci.lower	ci.upper
## 45	f	~1		2	2		-0.868	0.162	-5.345	0	-1.186	-0.55

As shown above, the factor mean difference was similar in the scalar model, -0.85 , 95% CI $[-1.24 - 0.45]$, and in the strict model, -0.87 , 95% CI $[-1.19 - 0.55]$. However, the standard error is larger in the scalar model, $SE = .20$, than in the strict model, $SE = .16$, resulting in a wider confidence interval in the scalar model. Although the statistical conclusion is the same between the two models in this example, as shown in the simulation results, statistical inference can be inaccurate in the scalar model but not in the strict model. In general, we recommend researchers to use the strict/partial strict model for factor mean comparison with dichotomous items, regardless of whether strict invariance holds.

Reference

- Millsap, Roger E., and Jenn Yun-Tein. 2004. "Assessing Factorial Invariance in Ordered-Categorical Measures." *Multivariate Behavioral Research* 39 (3): 479–515. https://doi.org/10.1207/S15327906MBR3903_4.
- Muthén, Bengt O. 2002. "Latent Variable Analysis with Categorical Outcomes: Multiple-group and Growth Modeling in Mplus." *Version 5. Technical Report*, 23.
- Putnick, Diane L., and Marc H. Bornstein. 2016. "Measurement Invariance Conventions and Reporting: The State of the Art and Future Directions for Psychological Research." *Developmental Review* 41 (September): 71–90. <https://doi.org/10.1016/j.dr.2016.06.004>.
- Satorra, Albert, and Peter M. Bentler. 2010. "Ensuring Positiveness of the Scaled Difference Chi-Square Test Statistic." *Psychometrika* 75 (2): 243–48. <https://doi.org/10.1007/s11336-009-9135-y>.
- Svetina, Dubravka, Leslie Rutkowski, and David Rutkowski. 2019. "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): 111–30. <https://doi.org/10.1080/10705511.2019.1602776>.
- Wu, Hao, and Ryne Estabrook. 2016. "Identification of Confirmatory Factor Analysis Models of Different Levels of Invariance for Ordered Categorical Outcomes." *Psychometrika* 81 (4): 1014–45. <https://doi.org/10.1007/s11336-016-9506-0>.