Multiple Group Models & Measurement Invariance

Theory Construction and Statistical Modeling

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Multiple Group

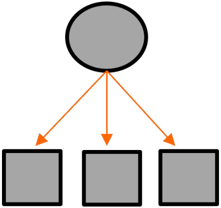
MG regression in R

Measurement invariance (MI)

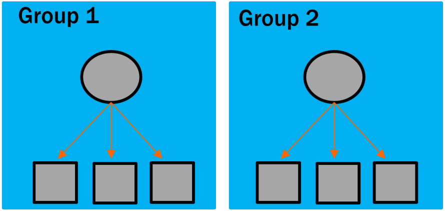
MI in R

From one group to multiple groups

- EFA or CFA for single group:



- EFA or CFA for multiple groups:



General setup of multigroup analysis

1. Each group treated as **separate dataset**.
    - Like “select data” in SPSS: select people who meet a specific condition (e.g., female, Dutch, etc)
    - lavaan does it behind the scenes, after denoting your groups.
2. Model is fitted in each group separately.
3. Apply constraints to test whether parameters are the same / significantly different across different groups.

Example: Multigroup regression

Regression example from lab meeting ‘Intro and regression’:

- Outcome: sw
- Predictors: overt and covert
- Group: gender (males and females)

Multigroup regression - lavaan model

```r
library(dplyr)
library(lavaan)

## Read in the data:
dat1 <- read.table("../data/popular_regr.txt",
                   header = TRUE,
                   na.strings = c("-99", "-999")
) %>%
mutate(sex = factor(gender, labels = c("male", "female"))) %>%
filter(!is.na(sex))

## Define the model syntax:
mod1 <- 'sw ~ 1 + overt + covert'

## Fit the model:
fit1 <- sem(mod1, data = dat1, group = "sex")
```

Multigroup regression - lavaan model

```r
## Define the model syntax with parameter labels:
mod1 <- '
sw ~ c("b0m", "b0f") * 1 +
     c("b1m", "b1f") * overt +
     c("b2m", "b2f") * covert
'

## Fit the model:
fit1 <- sem(mod1, data = dat1, group = "sex")
```

Multigroup regression - lavaan output

```
partSummary(fit1, 1:4)

lavaan 0.6.16 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         8

  Number of observations per group:               Used       Total
    male                                           756         824
    female                                         581         660

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0
  Test statistic for each group:
    male                                         0.000
    female                                       0.000
```

Multigroup regression - lavaan output

```
partSummary(fit1, 8:11)

Group 1 [male]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  sw ~
    overt     (b1m)   -0.278    0.059   -4.719    0.000
    covert    (b2m)   -0.497    0.039  -12.818    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .sw        (b0m)    4.930    0.085   57.938    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .sw                0.336    0.017   19.442    0.000
```

Multigroup regression - lavaan output

```
partSummary(fit1, 13:16)

Group 2 [female]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  sw ~
    overt     (b1f)  -0.232    0.081   -2.871    0.004
    covert    (b2f)  -0.558    0.045  -12.295    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .sw        (b0f)   5.062    0.106   47.703    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .sw                0.318    0.019   17.044    0.000
```

Comparing nested models

To compare fit of two nested models, you need:

1. χ^2 of the constrained model (χ^2_C)
2. df_C of the constrained model
3. χ^2 of the unconstrained model(χ^2_U)
4. df_U of the unconstrained model

Then, subtract: χ^2_C - χ^2_u = χ^2_Δ,
where χ^2_Δ follows a Chi-square distribution with $df = df_C$-df_U.

The function lavTestWald() does this for you!

Comparing nested models Ctd.

Hypotheses
H_0: Unconstrained model fit = constrained model fit
H_A: Unconstrained model fit ≠ constrained model fit

High p-values: No significant difference between the model fits: No evidence that the coefficients differ across groups.

Low p-values: Significant difference between the model fits: Evidence that the coefficients differ across groups.

Multigroup regression: Test equality of coefficients Ctd.

```
## Test equality of regression slopes:
lavTestWald(fit1, constraints = 'b1m == b1f; b2m == b2f')

$stat
[1] 1.035523

$df
[1] 2

$p.value
[1] 0.5958527

$se
[1] "standard"
```

Why do we care?

Construct validity:

- Is the model measuring the same thing in both (all) groups?

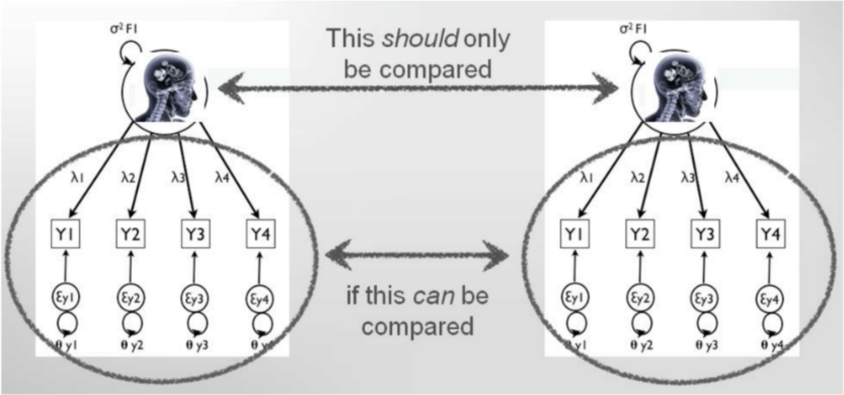
Can we make a fair comparison between groups?

- Did the groups understand the questions in the same way?

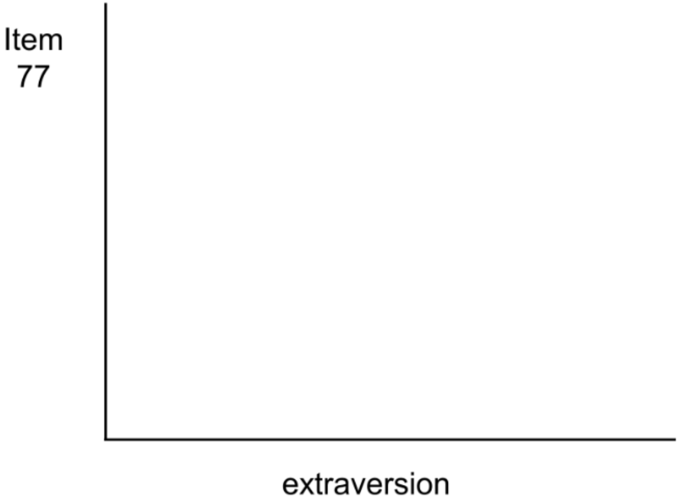
Same latent score should result in the same observed scores.

- Equal slopes (factor loadings)
- Equal intercepts (item means)

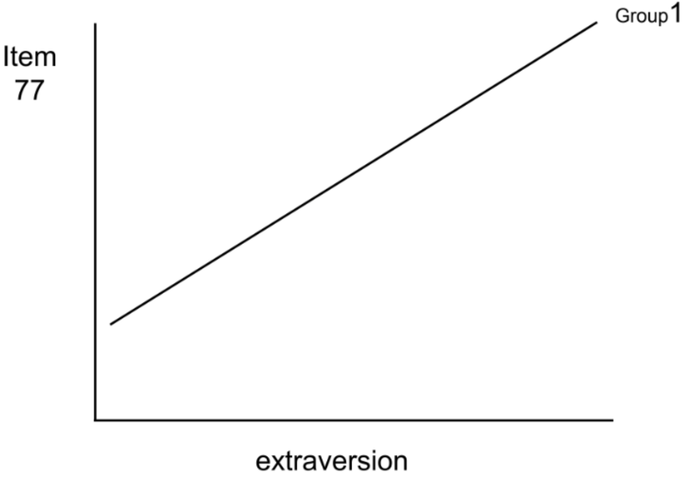
Stated Differently



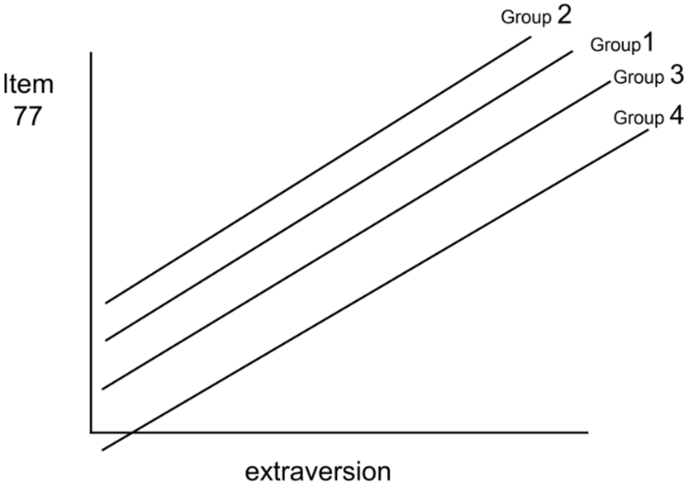
Stated Differently



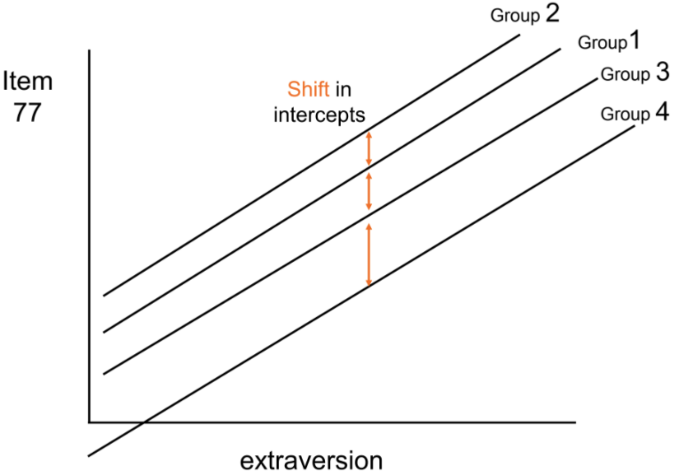
Stated Differently



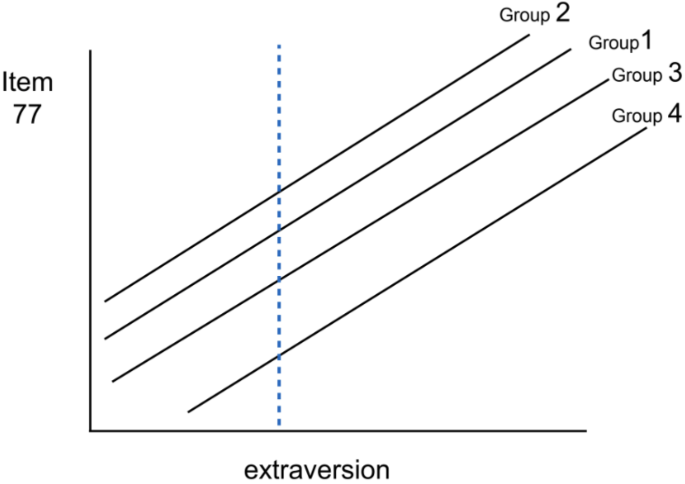
Stated Differently



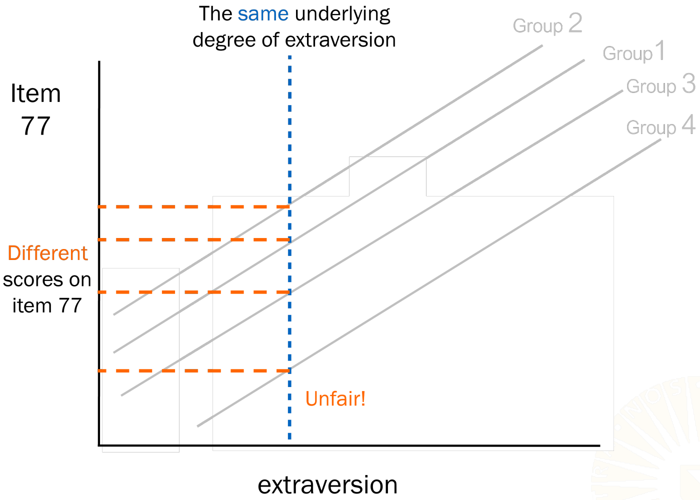
Stated Differently



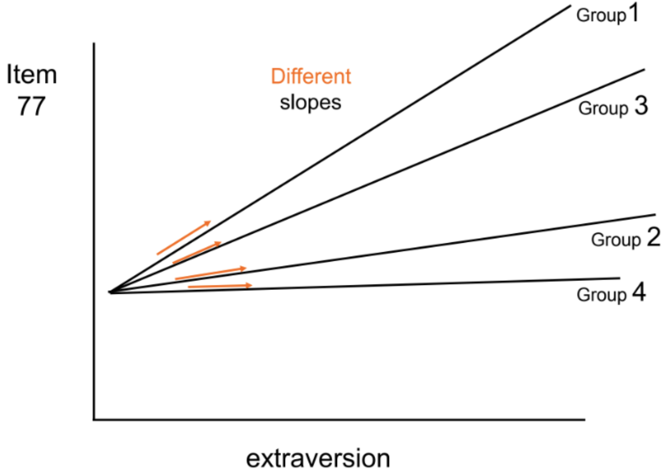
Stated Differently



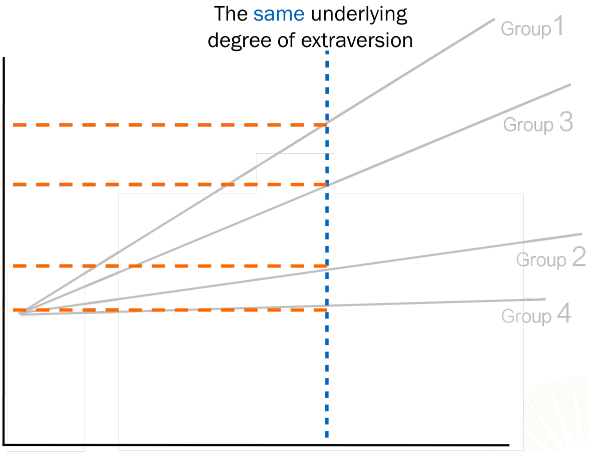
Stated Differently


The same underlying degree of extraversion
Group 2
Group 1
Group 3
Group 4
Item 77
Different scores on item 77
Unfair!
extraversion

Stated Differently



Stated Differently



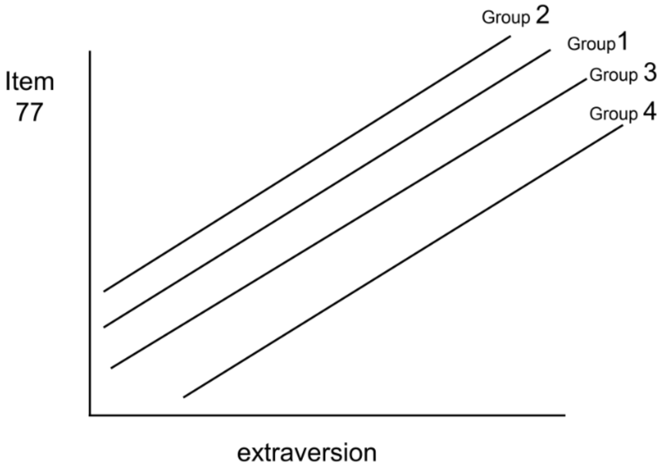
Item 77

Different scores on item 77

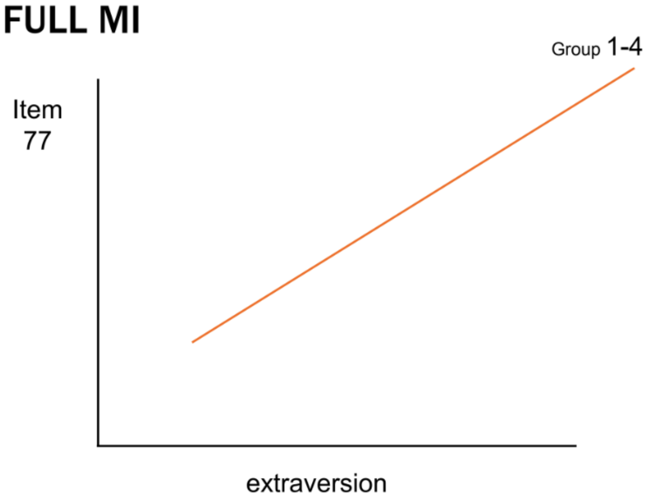
(differences depend on degree of extraversion)

extraversion

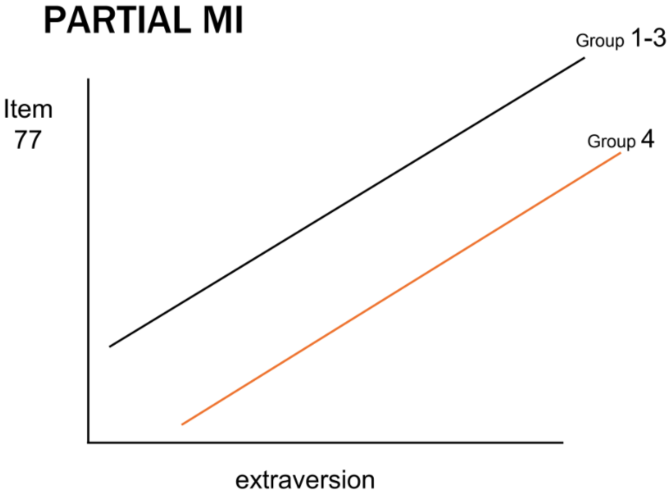
Stated Differently



Stated Differently



Stated Differently

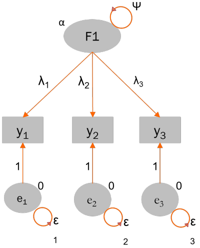


Confirmatory bottom-up approach

1. Test model separately for each group (**configural invariance**).
   - Must fit the data
2. Test equality of loadings across groups (**metric/weak invariance**).
   - Must be equal
3. Test equality of intercepts across groups (**scalar/strong invar.**).
   - Must be equal
4. Test equality of measurement error variances (**strict invariance**).
   - Not essential, often overly restrictive

Testing measurement invariance (MI)

1. Test overall model.
2. Test model for groups separately.
3. Test equal factor loadings (λ_1, λ_2, λ_3,).
4. Test equal intercepts / item means (ν_1, ν_2, ν_3,).
5. Test equal residual/error variances (ϵ_1, ϵ_2, ϵ_3,).
6. Test factor means (α) and/or variances (Ψ).



Note: Steps 3, 4, and 5, that is, testing for configural invariance, metric/weak invariance, and scalar/strong invariance, respectively, are quite easily incorporated in lavaan.

Example: South African Personality Inventory Project (SAPI)

Carin Hill
Leon Jackson
Deon Meiring
J. Aleweyn Nel

Ian Rothmann
Michael Temane
Velichko H. Valchev
Fons J. R. van de Vijver

Nel, J. A., Valchev, V. H., Rothmann, S., van de Vijver, F. J. R., Meiring, D., & de Bruin, G. P. (2012). Exploring the personality struct

SAPI Example: MI models with lavaan

```
## Read in the data:
sapi <- read.table("../data/sapi.txt",
                   header = TRUE,
                   na.strings = "-999") %>%
mutate(sex = factor(Gender, labels = c("male", "female"))) %>%
filter(!is.na(sex))

## Define the model syntax:
cfaMod <- '
having_fun  =~ Q77 + Q84 + Q196
being_liked =~ Q44 + Q63 + Q98
'

## Fit the model:
```

SAPI Example: MI models with lavaan Ctd.

```r
## Configurally invariant model:
configFit <- cfa(cfaMod,
                 data = sapi,
                 std.lv = TRUE,
                 group = "sex",
                 missing = "FIML")

## Weakly invariant model:
weakFit <- cfa(cfaMod,
               data = sapi,
               std.lv = TRUE,
               group = "sex",
               group.equal = "loadings",
               missing = "FIML")

## Strongly invariant model:
strongFit <- cfa(cfaMod,
                 data = sapi,
                 std.lv = TRUE,
                 group = "sex",
                 group.equal = c("loadings", "intercepts"),
                 missing = "FIML")
```

SAPI Example: Testing MI with lavaan

```
library(semTools)
compareFit(configFit, weakFit, strongFit) %>% summary()

################### Nested Model Comparison #########################

Chi-Squared Difference Test

          Df   AIC   BIC  Chisq Chisq diff  RMSEA Df diff Pr(>Chisq)
configFit 16 15354 15540 85.083
weakFit   20 15348 15514 86.483     1.3992 0.0000       4     0.8443
strongFit 24 15347 15494 93.862     7.3791 0.0415       4     0.1172

####################### Model Fit Indices ###########################
            chisq df pvalue rmsea   cfi   tli  srmr        aic        bic
configFit 85.083† 16   .000 .094  .939  .886  .039† 15354.393  15540.159
weakFit   86.483  20   .000 .082  .942† .913  .039  15347.793  15514.004
strongFit 93.862  24   .000 .077† .939  .923† .041  15347.172† 15493.829†

################## Differences in Fit Indices #######################
                    df  rmsea    cfi   tli  srmr    aic     bic
weakFit - configFit  4 -0.011  0.002 0.026 0.000 -6.601 -26.155
strongFit - weakFit  4 -0.005 -0.003 0.011 0.002 -0.621 -20.175
```

SAPI Example: MI - Model fit

– Cheung, G. W., & Rensvold, R. B. (2002). Evaluating goodness-of-fit indexes for testing measurement invariance. Structural Equation Modeling, 9(2), 233–255. doi:10.1207/S15328007SEM0902_5

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

Testing MI with lavaan: IF significant

- If the test for configural invariance against weak invariance is significant, then there is a lack of metric invariance and, thus, there is no need to test for scalar and strict invariance.

- If tests significant: may try to find source of bias with modification indices.
- Then, aim for partial MI: Continue with MI tests with source of bias freely estimated between groups.

Moderation via Multiple Group SEM

When our moderator is a categorical variable, we can use multiple group CFA/SEM to test for moderation.

- Categorical moderators define groups.
- Significant moderation with categorical moderators implies between-group differences in the focal effect.
- We can directly test these hypotheses with multiple group SEM.

Example

```r
## Read the data and subset to only high school and college graduates:
bfi <- readRDS("../data/bfiData2.rds") %>%
    filter(educ %in% c("highSchool", "college"))

## Specify the (configurally invariance) measurement model:
mod0 <- '
agree =~ A1 + A2 + A3 + A4 + A5
open  =~ O1 + O2 + O3 + O4 + O5
'

## Estimate the unrestricted model:
out0 <- cfa(mod0, data = bfi, std.lv = TRUE, group = "educ")
```

Example

```r
## Define the weakly invariant model:
mod1 <- measEq.syntax(configural.model = out0,
                      group = "educ",
                      group.equal = "loadings") %>%
    as.character()

## Define the strongly invariant model:
mod2 <- measEq.syntax(configural.model = out0,
                      group = "educ",
                      group.equal = c("loadings", "intercepts")
                      ) %>%
    as.character()

## Estimate the models:
out1 <- cfa(mod1, data = bfi, group = "educ")
out2 <- cfa(mod2, data = bfi, group = "educ")

## Test measurement invariance:
compareFit(out0, out1, out2) %>% summary()
```

Example

```
################### Nested Model Comparison #########################

Chi-Squared Difference Test

     Df   AIC   BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
out0 68 75336 75694 418.25
out1 76 75358 75669 455.70     37.451 0.055855       8  9.505e-06 ***
out2 84 75461 75726 575.15    119.446 0.108654       8  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

####################### Model Fit Indices ###########################
        chisq df pvalue rmsea   cfi   tli  srmr        aic        bic
out0 418.253† 68   .000 .066  .906† .875  .046† 75336.292† 75693.810
out1 455.705  76   .000 .065† .898  .879† .050  75357.744  75669.130†
out2 575.151  84   .000 .070  .868  .858  .056  75461.190  75726.445

################## Differences in Fit Indices #######################
            df  rmsea    cfi    tli  srmr     aic     bic
out1 - out0  8 -0.001 -0.008  0.004 0.004  21.451 -24.680
out2 - out1  8  0.005 -0.030 -0.021 0.006 103.446  57.315
```

Example

```r
## Specify a structural model:
mod3 <- '
agree =~ A1 + A2 + A3 + A4 + A5
open  =~ O1 + O2 + O3 + O4 + O5

agree ~ open
'

## Estimate the model with strong invariance constraints:
out3 <- sem(mod3,
            data = bfi,
            std.lv = TRUE,
            group = "educ",
            group.equal = c("loadings", "intercepts")
            )
```

Example

```
## Check the group-specific slopes:
partSummary(out3, c(8, 10, 14, 16))

Group 1 [highSchool]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~
    open             -0.321    0.040   -7.957    0.000

Group 2 [college]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~
    open             -0.203    0.051   -3.972    0.000
```

Example

```
## Specify the restricted model:
mod4 <- '
agree =~ A1 + A2 + A3 + A4 + A5
open  =~ O1 + O2 + O3 + O4 + O5

agree ~ c(beta, beta) * open
'

## Estimate the model:
out4 <- sem(mod4,
            data = bfi,
            std.lv = TRUE,
            group = "educ",
            group.equal = c("loadings", "intercepts")
            )
```

Example

```
## Check the slopes:
partSummary(out4, c(8, 10, 14, 16))

Group 1 [highSchool]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~
    open    (beta)   -0.278    0.032   -8.621    0.000

Group 2 [college]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~
    open    (beta)   -0.278    0.032   -8.621    0.000
```

Example

```
## Do a chi-squared difference test for moderation:
anova(out3, out4)


Chi-Squared Difference Test

     Df   AIC   BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
out3 84 75461 75726 575.15
out4 85 75463 75722 578.59      3.435 0.045426       1    0.06383 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example

```
## Do a similar test via OLS regression:
readRDS("../data/bfiData1.rds") %>%
    filter(educ %in% c("highSchool", "college")) %$%
    lm(agree ~ open * educ) %>%
    partSummary(-(1:2))

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       3.24965    0.12321  26.376   <2e-16
open              0.27115    0.03134   8.652   <2e-16
educcollege       0.03975    0.22849   0.174    0.862
open:educcollege -0.05654    0.05856  -0.965    0.334

Residual standard error: 0.6972 on 2356 degrees of freedom
Multiple R-squared:  0.05314,Adjusted R-squared:  0.05194
F-statistic: 44.08 on 3 and 2356 DF,  p-value: < 2.2e-16
```

Probing Multiple Group Moderation

Testing moderation with multiple group SEM has several advantages.

- Remove measurement error from the estimates
- Test for factorial invariance
- All simple effects are directly estimated in the unrestricted model

Simple Slopes & Intercepts

```
Group 1 [highSchool]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~
    open             -0.321    0.040   -7.957    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .agree             0.000

Group 2 [college]:

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~
    open             -0.203    0.051   -3.972    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .agree             0.170    0.056    3.058    0.002
```

Simple Slopes Visualized

We can visualize the simple slopes by plotting the factor scores.

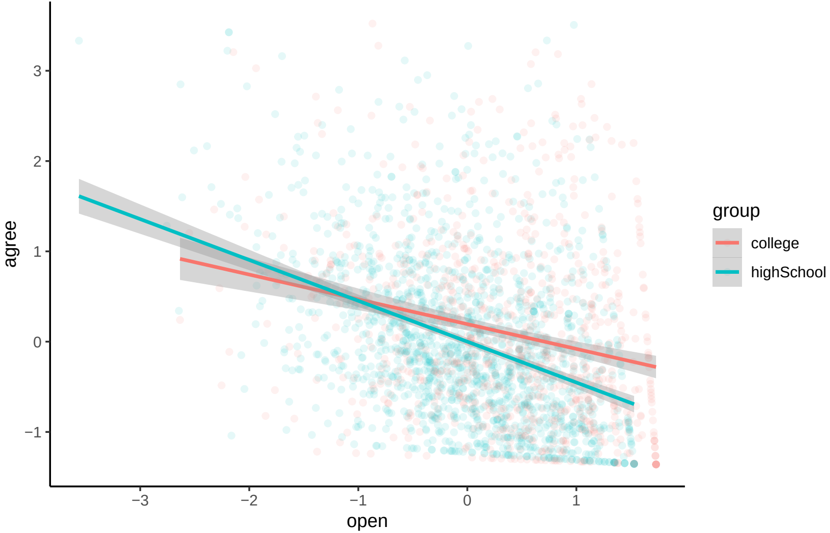
```
library(ggplot2)

## Generate factor scores:
tmp <- predict(out3)

## Stack factor scores into a "tidy" dataset:
pData <- data.frame(do.call(rbind, tmp),
                    group = rep(names(tmp), sapply(tmp, nrow))
                    )

## Create a simple slopes plot:
ssPlot <- ggplot(pData, aes(open, agree, color = group)) +
    geom_point(alpha = 0.1) +
    geom_smooth(method = "lm") +
    theme_classic()
```

Simple Slopes Visualized


agree
open
group
college
highSchool