Supplemental Example

Predicting Factor Scores vs. True Latent Variables

Table of contents

```r
library(lavaan)
```

```
This is lavaan 0.6-17
lavaan is FREE software! Please report any bugs.
```

```r
library(ggplot2)
```

The generating model is basically the same as Example 1 in the manuscript, but with a much larger sample size and an additional predictor w for predicting the latent variable eta.

```r
set.seed(1127)
# Simulation condition
num_obs <- 5000  # small sample per group
# Mean of predictor variable w
mean_w <- c(1, 2)
gamma <- 0.5  # coefficient of w on eta
# Simulate scalar invariant data with same mean
lambda <- c(0.9, 0.7, 0.5)
nu <- c(0, 0, 0)
theta <- diag(1 - lambda^2)
# Group 1
psi1 <- 1.6
alpha1 <- 0
# Group 2
psi2 <- 0.4
alpha2 <- 0.8
```

```r
# Function for data generation
gendata <- function(nobs, lambda1, lambda2 = lambda1,
                    nu1, nu2 = nu1, theta1, theta2 = theta1,
                    psi1, psi2, alpha1, alpha2) {
    zero_p <- 0 * lambda  # zero vector of length p (for convenience)
    vw_psi_theta1 <- rbind(
        c(1, 0, zero_p),
        c(0, psi1, zero_p),
        cbind(0, zero_p, theta1)
    )
    w_eta_eps1 <- MASS::mvrnorm(nobs,
        mu = c(mean_w[1], alpha1, 0 * lambda1),
        Sigma = vw_psi_theta1, empirical = TRUE
    )
    w1 <- w_eta_eps1[, 1]
    eta1 <- w1 * gamma + w_eta_eps1[, 2]
    eps1 <- w_eta_eps1[, -(1:2)]
    y1 <- t(nu1 + t(tcrossprod(eta1, lambda1) + eps1))
    vw_psi_theta2 <- rbind(
        c(1, 0, zero_p),
        c(0, psi2, zero_p),
        cbind(0, zero_p, theta2)
    )
    w_eta_eps2 <- MASS::mvrnorm(nobs,
        mu = c(mean_w[2], alpha2, 0 * lambda),
        Sigma = vw_psi_theta2, empirical = TRUE
    )
    w2 <- w_eta_eps2[, 1]
    eta2 <- w2 * gamma + w_eta_eps2[, 2]
    eps2 <- w_eta_eps2[, -(1:2)]
    y2 <- t(nu2 + t(tcrossprod(eta2, lambda2) + eps2))
    out <- rbind(cbind(eta1, y1, group = 1, w = w1),
                 cbind(eta2, y2, group = 2, w = w2))
    out <- data.frame(out)
    out$group <- factor(out$group)
    colnames(out) <- c("eta", paste0("y", seq_along(lambda)), "group", "w")
    out
}
dat_y <- gendata(num_obs, lambda1 = lambda, nu1 = nu, theta1 = theta,
                 psi1 = psi1, psi2 = psi2, alpha1 = alpha1, alpha2 = alpha2)
```

Strict invariance model

The result below confirms that strict invariance is tenable with the simulated data.

```
strict_fit <- cfa(
    "
    f =~ NA * y1 + y2 + y3
    f ~~ c(1, NA) * f
    ",
    data = dat_y,
    # std.lv = TRUE,
    group = "group",
    group.equal = c("loadings", "intercepts", "residuals"),
    likelihood = "wishart"
)
summary(strict_fit)
```

```
lavaan 0.6.17 ended normally after 26 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        20
  Number of equality constraints                     9

  Number of observations per group:
    1                                             5000
    2                                             5000

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 7
  P-value (Chi-square)                           1.000
  Test statistic for each group:
    1                                            0.000
    2                                            0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

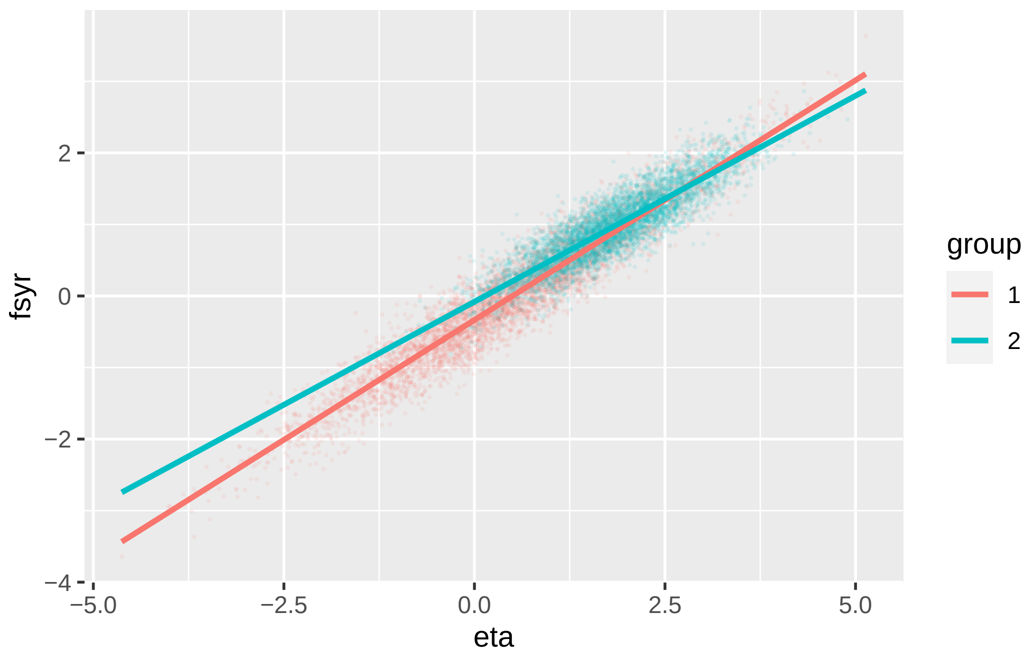
```
Group 1 [1]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1      (.p1.)    1.224    0.014   85.316    0.000
    y2      (.p2.)    0.952    0.013   72.621    0.000
    y3      (.p3.)    0.680    0.012   56.405    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1      (.p8.)    0.450    0.018   24.570    0.000
   .y2      (.p9.)    0.350    0.016   21.744    0.000
   .y3      (.10.)    0.250    0.014   17.945    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    f                 1.000
   .y1      (.p5.)    0.190    0.011   17.042    0.000
   .y2      (.p6.)    0.510    0.010   51.623    0.000
   .y3      (.p7.)    0.750    0.011   65.818    0.000


Group 2 [2]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1      (.p1.)    1.224    0.014   85.316    0.000
    y2      (.p2.)    0.952    0.013   72.621    0.000
    y3      (.p3.)    0.680    0.012   56.405    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1      (.p8.)    0.450    0.018   24.570    0.000
   .y2      (.p9.)    0.350    0.016   21.744    0.000
   .y3      (.10.)    0.250    0.014   17.945    0.000
    f                 0.956    0.021   46.455    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
```

```
 f                  0.351    0.012   28.934    0.000
.y1       (.p5.)    0.190    0.011   17.042    0.000
.y2       (.p6.)    0.510    0.010   51.623    0.000
.y3       (.p7.)    0.750    0.011   65.818    0.000
```

Regression Factor Scores with Group-Specific Latent Distributions

```r
# Regression factor scores
dat_y$fsyr <- lavPredict(strict_fit, assemble = TRUE)$f
dat_y |>
    ggplot(aes(x = eta, y = fsyr, col = group)) +
    geom_point(alpha = 0.1, size = 0.1) +
    geom_smooth(method = "lm", se = FALSE, fullrange = TRUE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Examining Group $\times$ w interaction

The code below shows that using regression factor scores in place of η as the outcome results in a spurious interaction.

```r
# No interaction with true latent variable
lm(eta ~ w * group, data = dat_y) |> summary()
```

```

Call:
lm(formula = eta ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.8370 -0.5796  0.0048  0.5872  4.0928 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.435e-14  2.000e-02    0.00        1    
w            5.000e-01  1.414e-02   35.35   <2e-16 ***
group2       8.000e-01  3.742e-02   21.38   <2e-16 ***
w:group2    -8.150e-16  2.000e-02    0.00        1    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 9996 degrees of freedom
Multiple R-squared:  0.4021,	Adjusted R-squared:  0.4019 
F-statistic:  2241 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```r
# Spurious interaction with regression factor scores
lm(fsyr ~ w * group, data = dat_y) |> summary()
```

```

Call:
lm(formula = fsyr ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2333 -0.4082 -0.0031  0.4119  3.2040 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.335021   0.014083 -23.789  < 2e-16 ***
w            0.335021   0.009959  33.640  < 2e-16 ***
group2       0.714991   0.026348  27.136  < 2e-16 ***
w:group2    -0.047116   0.014084  -3.345 0.000825 ***
```

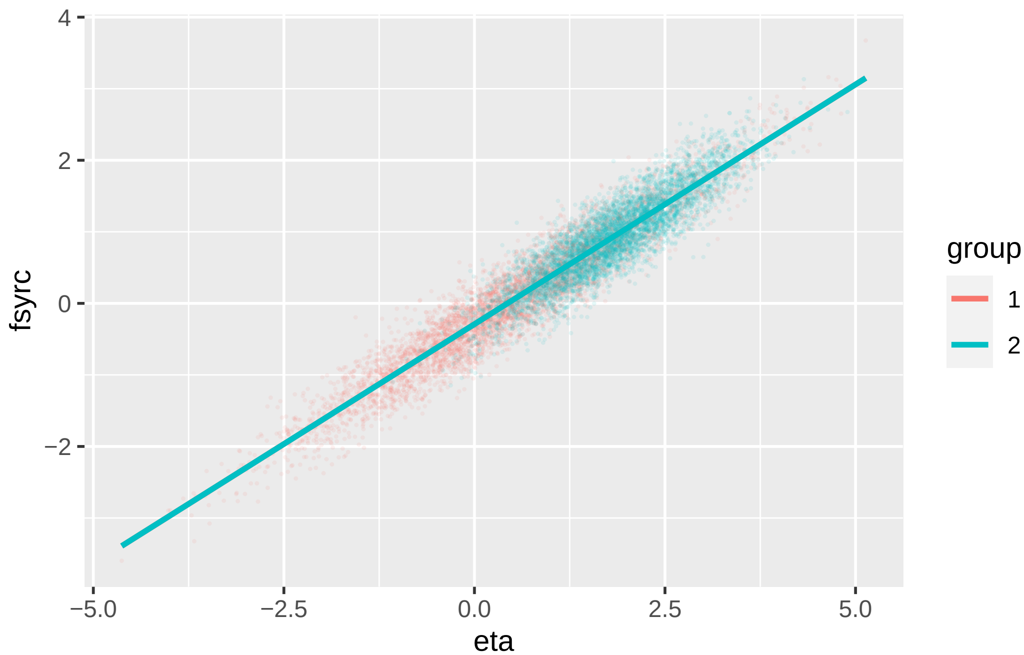
```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7041 on 9996 degrees of freedom
Multiple R-squared:  0.3967,    Adjusted R-squared:  0.3965
F-statistic:  2191 on 3 and 9996 DF,  p-value: < 2.2e-16
```

Regression Factor Scores with Common Latent Distributions

```r
# Function for scoring matrix
compute_a_reg <- function(lambda, theta, psi) {
    covy <- lambda %*% psi %*% t(lambda) + theta
    ginvcovy <- MASS::ginv(covy)
    tlam_invcov <- crossprod(lambda, ginvcovy)
    psi %*% tlam_invcov
}
# Function for computing factor scores with common distributions
compute_fscore <- function(y, lambda, nu, theta,
                           psi, alpha) {
    y1c <- t(as.matrix(y))
    meany <- lambda %*% alpha + nu
    y1c <- y1c - as.vector(meany)
    a_mat <- compute_a_reg(lambda, psi = psi, theta = theta)
    t(a_mat %*% y1c + as.vector(alpha))
}
strict_pars <- lavInspect(strict_fit, what = "est")
fscores_reg3 <- list(
    `1` = compute_fscore(
        dat_y[dat_y$group == 1, 2:4],
        lambda = strict_pars[[1]]$lambda,
        theta = strict_pars[[1]]$theta,
        nu = strict_pars[[1]]$nu,
        psi = 1,
        alpha = 0.5
    ),
    `2` = compute_fscore(
        dat_y[dat_y$group == 2, 2:4],
        lambda = strict_pars[[2]]$lambda,
        theta = strict_pars[[2]]$theta,
        nu = strict_pars[[2]]$nu,
        psi = 1,
```

```r
        alpha = 0.5
    )
)
# Add to data
dat_y$fsyrc <- do.call(rbind, fscores_reg3)
dat_y |>
    ggplot(aes(x = eta, y = fsyrc, col = group)) +
    geom_point(alpha = 0.1, size = 0.1) +
    geom_smooth(method = "lm", se = FALSE, fullrange = TRUE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Examining Group $\times$ w interaction

Using the same latent distribution for computing factor scores avoids the spurious interaction.

```r
lm(eta ~ w * group, data = dat_y) |> summary()
```

```
Call:
lm(formula = eta ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.8370 -0.5796  0.0048  0.5872  4.0928 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.435e-14  2.000e-02    0.00        1    
w            5.000e-01  1.414e-02   35.35   <2e-16 ***
group2       8.000e-01  3.742e-02   21.38   <2e-16 ***
w:group2    -8.150e-16  2.000e-02    0.00        1    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 9996 degrees of freedom
Multiple R-squared:  0.4021,	Adjusted R-squared:  0.4019 
F-statistic:  2241 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```r
lm(fsyrc ~ w * group, data = dat_y) |> summary()
```

```
Call:
lm(formula = fsyrc ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2333 -0.4464 -0.0035  0.4454  3.2040 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.907e-01  1.456e-02  -19.97   <2e-16 ***
w            3.350e-01  1.030e-02   32.54   <2e-16 ***
group2       5.360e-01  2.724e-02   19.68   <2e-16 ***
w:group2    -3.509e-16  1.456e-02    0.00        1    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7279 on 9996 degrees of freedom
Multiple R-squared:  0.3631,	Adjusted R-squared:  0.3629 
F-statistic:  1899 on 3 and 9996 DF,  p-value: < 2.2e-16
```

Including w When Computing Regression Factor Scores

Another possible option is to use covariate-informed factor scores [see @xxx] by including w in the strict invariance model.

```
strict_w_fit <- cfa(
    "
    f =~ NA * y1 + y2 + y3
    f ~~ c(1, NA) * f
    f ~ w
    ",
    data = dat_y,
    # std.lv = TRUE,
    group = "group",
    group.equal = c("loadings", "intercepts", "residuals"),
    likelihood = "wishart"
)
summary(strict_w_fit)
```

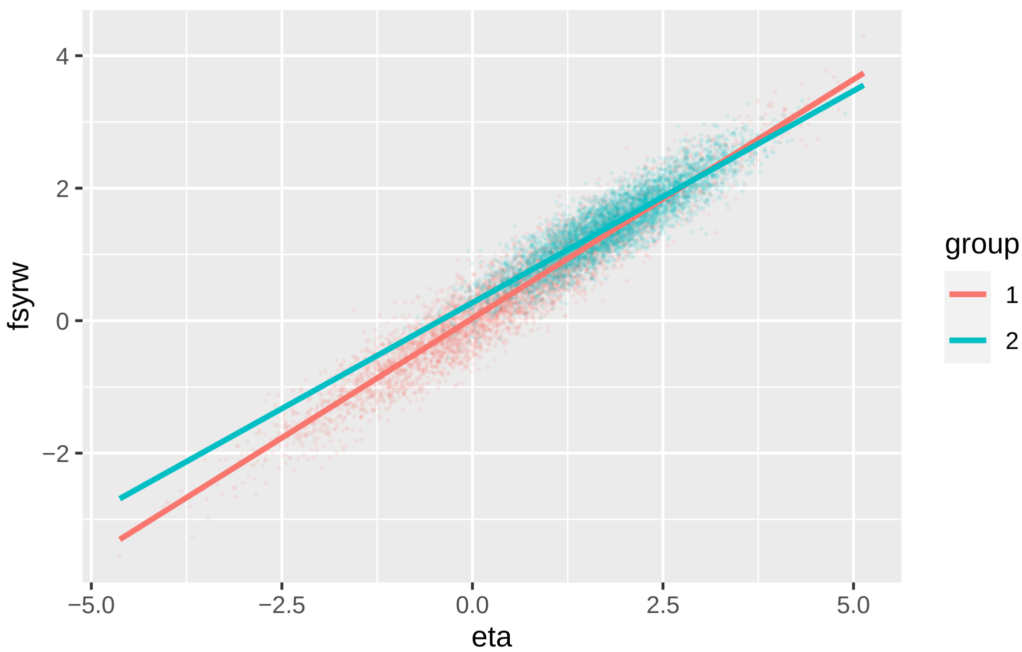
```
lavaan 0.6.17 ended normally after 29 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        22
  Number of equality constraints                     9

  Number of observations per group:
    1                                             5000
    2                                             5000

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                11
  P-value (Chi-square)                           1.000
  Test statistic for each group:
    1                                            0.000
    2                                            0.000

Parameter Estimates:

  Standard errors                             Standard
```

```
  Information                                      Expected
  Information saturated (h1) model               Structured


Group 1 [1]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1      (.p1.)    1.138    0.013   85.322    0.000
    y2      (.p2.)    0.885    0.012   73.342    0.000
    y3      (.p3.)    0.632    0.011   56.589    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  f ~
    w                 0.395    0.016   25.383    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1      (.10.)    0.000    0.024    0.000    1.000
   .y2      (.11.)    0.000    0.021    0.000    1.000
   .y3      (.12.)    0.000    0.018    0.000    1.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .f                 1.000
   .y1      (.p6.)    0.190    0.009   20.825    0.000
   .y2      (.p7.)    0.510    0.009   55.966    0.000
   .y3      (.p8.)    0.750    0.011   66.555    0.000


Group 2 [2]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1      (.p1.)    1.138    0.013   85.322    0.000
    y2      (.p2.)    0.885    0.012   73.342    0.000
    y3      (.p3.)    0.632    0.011   56.589    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
```

```
  f ~
    w                 0.395    0.010   41.104    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1      (.10.)    0.000    0.024    0.000    1.000
   .y2      (.11.)    0.000    0.021    0.000    1.000
   .y3      (.12.)    0.000    0.018    0.000    1.000
   .f                 0.632    0.029   21.582    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .f                 0.250    0.010   25.980    0.000
   .y1      (.p6.)    0.190    0.009   20.825    0.000
   .y2      (.p7.)    0.510    0.009   55.966    0.000
   .y3      (.p8.)    0.750    0.011   66.555    0.000
```

```r
# Regression factor scores
dat_y$fsyrw <- lavPredict(strict_w_fit, assemble = TRUE)$f
dat_y |>
    ggplot(aes(x = eta, y = fsyrw, col = group)) +
    geom_point(alpha = 0.1, size = 0.1) +
    geom_smooth(method = "lm", se = FALSE, fullrange = TRUE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Examining Group $\times$ w interaction

The code below shows that using regression factor scores in place of η as the outcome results in a spurious interaction.

```r
# No interaction with true latent variable
lm(eta ~ w * group, data = dat_y) |> summary()
```

```

Call:
lm(formula = eta ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.8370 -0.5796  0.0048  0.5872  4.0928 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.435e-14  2.000e-02    0.00        1    
w            5.000e-01  1.414e-02   35.35   <2e-16 ***
group2       8.000e-01  3.742e-02   21.38   <2e-16 ***
w:group2    -8.150e-16  2.000e-02    0.00        1    
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 9996 degrees of freedom
Multiple R-squared:  0.4021,	Adjusted R-squared:  0.4019 
F-statistic:  2241 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```r
# Spurious interaction with regression factor scores
lm(fsyrw ~ w * group, data = dat_y) |> summary()
```

```

Call:
lm(formula = fsyrw ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4292 -0.4048 -0.0031  0.4059  3.3982 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.517e-09  1.464e-02    0.00        1    
w            3.953e-01  1.035e-02   38.18   <2e-16 ***
group2       6.325e-01  2.739e-02   23.09   <2e-16 ***
w:group2     3.371e-09  1.464e-02    0.00        1    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.732 on 9996 degrees of freedom
Multiple R-squared:  0.4397,	Adjusted R-squared:  0.4395 
F-statistic:  2615 on 3 and 9996 DF,  p-value: < 2.2e-16
```

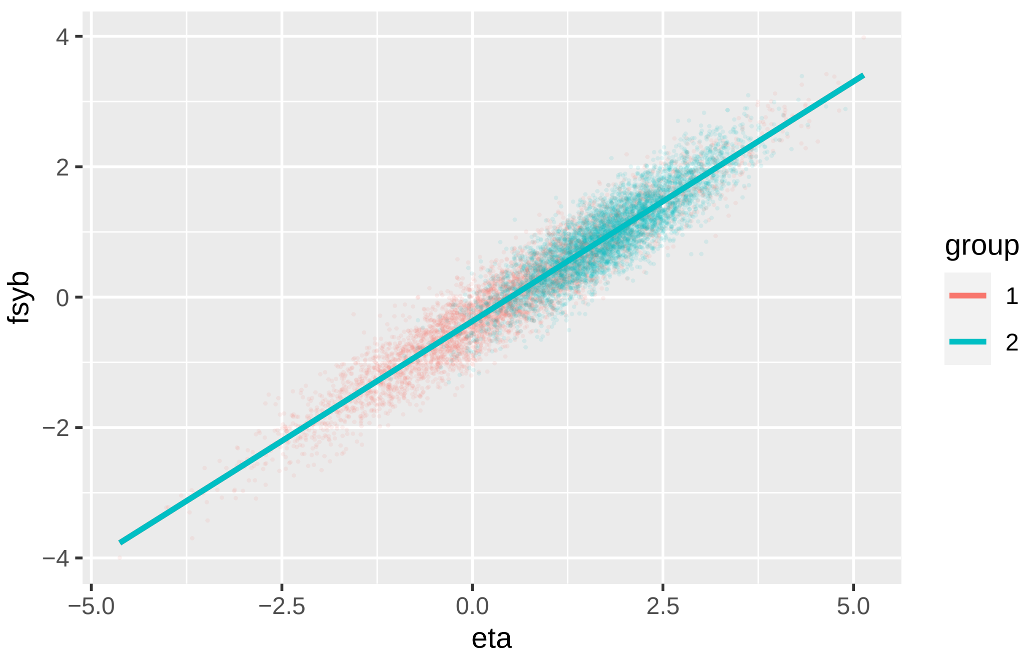
The caveat is one may need to include all potential predictors in the measurement and scoring model.

Bartlett Factor Scores

Bartlett factor scores are scalar invariant, so it does not lead to spurious interactions.

```
# Regression factor scores
dat_y$fsyb <- lavPredict(strict_fit, method = "Bartlett", assemble = TRUE)$f
dat_y |>
    ggplot(aes(x = eta, y = fsyb, col = group)) +
    geom_point(alpha = 0.1, size = 0.1) +
    geom_smooth(method = "lm", se = FALSE, fullrange = TRUE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Examining Group $\times$ w interaction

The code below shows that using regression factor scores in place of η as the outcome results in a spurious interaction.

```
# No interaction with true latent variable
lm(eta ~ w * group, data = dat_y) |> summary()
```

```
Call:
lm(formula = eta ~ w * group, data = dat_y)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-4.8370 -0.5796  0.0048  0.5872  4.0928 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.435e-14  2.000e-02    0.00        1    
w            5.000e-01  1.414e-02   35.35   <2e-16 ***
group2       8.000e-01  3.742e-02   21.38   <2e-16 ***
w:group2    -8.150e-16  2.000e-02    0.00        1    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 9996 degrees of freedom
Multiple R-squared:  0.4021,	Adjusted R-squared:  0.4019 
F-statistic:  2241 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```r
# Spurious interaction with regression factor scores
lm(fsyb ~ w * group, data = dat_y) |> summary()
```

```
Call:
lm(formula = fsyb ~ w * group, data = dat_y)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.5478 -0.4898 -0.0038  0.4887  3.5156 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.676e-01  1.597e-02  -23.01   <2e-16 ***
w            3.676e-01  1.130e-02   32.54   <2e-16 ***
group2       5.882e-01  2.989e-02   19.68   <2e-16 ***
w:group2    -1.521e-16  1.598e-02    0.00        1    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7987 on 9996 degrees of freedom
Multiple R-squared:  0.3631,	Adjusted R-squared:  0.3629 
F-statistic:  1899 on 3 and 9996 DF,  p-value: < 2.2e-16
```