Simulation Example

Steven Hurwitt

4/29/2020

Basics

Start by including the packages, setting seed, directory and sample size.

```r
library(seminr)
library(mnormt)

getwd()
```

```
## [1] "/media/steven/samsung_t5/gerson"
```

```r
setwd("/media/steven/samsung_t5/gerson")
set.seed(421)
N = 100
```

Data model

We start similar to Lai 2020 by simulating the latent means and covariances. We will use these with the factor loadings to generate the data. Thus, the data is highly variable depending on different choices of means (α) and covariances (ϕ)

$$\begin{bmatrix} y_{0i} \\ y_{1i} \\ y_{2i} \end{bmatrix} = \Lambda \begin{bmatrix} \eta_{1i} \\ \eta_{2i} \\ \eta_{3i} \end{bmatrix} + \begin{bmatrix} e_{0i} \\ e_{1i} \\ e_{2i} \end{bmatrix}$$

```r
opt_lambda = c(.86, .88, .8)
opt_alpha = c(2, 4, 6)
opt_phi = matrix(c(1, .2, .3, .2, .3, 1, .3, .2, 1), nrow = 3)

pt_lambda = c(.78, .8, .64, .8, .75)
pt_alpha = c(4, 3, 5, 3, 4)
pt_phi = matrix(c(1, .2, .3, .4, .5,
                  .2, 1, .4, .5, .3,
                  .3, .4, .5, 1, .2,
                  .4, .5, .3, 1, .2,
                  .5, .4, .3, .2, 1), nrow = 5)

fse_lambda = c(.85, .89, .7)
fse_alpha = c(3, 5, 4)
```

```r
fse_phi = matrix(c(1, .2, .3, .2, .3, 1, .3, .2, 1), nrow = 3)

teamperf_lambda = c(.85, .82, .75)
teamperf_alpha = c(3, 5, 4)
teamperf_phi = matrix(c(1, .2, .3, .2, .3, 1, .3, .2, 1), nrow = 3)
```

Simulating data

In the simulated data, they all follow the normal distribution with given means and covariances.

$$\begin{bmatrix} \eta_{1i} \\ \eta_{2i} \\ \eta_{3i} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \right)$$

The error variances are normally distributed as:

$$\begin{bmatrix} e_{0i} \\ e_{1i} \\ e_{2i} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \theta_{11} & 0 & 0 \\ 0 & \theta_{22} & 0 \\ 0 & 0 & \theta_{33} \end{bmatrix} \right)$$

We will generate data for all of the variables: optimism, perspective taking, self efficacy and team performance.

```r
opt_eta = rmnorm(N, mean = opt_alpha, varcov = opt_phi)

pt_eta = rmnorm(N, mean = pt_alpha, varcov = pt_phi)

fse_eta = rmnorm(N, mean = fse_alpha, varcov = fse_phi)

teamperf_eta = rmnorm(N, mean = teamperf_alpha, varcov = teamperf_phi)
```

To simulate according to composite reliability, we need to know that:

$$CR = \frac{(\sum \lambda_i)^2}{(\sum \lambda_i)^2 + \sum \sigma_i^2}$$

We can then solve this for the sum of the error variances:

$$\sum \sigma_i^2 = \frac{(\sum \lambda_i)^2}{CR} - (\sum \lambda_i)^2$$

We will choose to uniformly distribute the sum of the error variances as the variance in the error term for each generated variable.

```r
lambda = c(opt_lambda, pt_lambda, fse_lambda, teamperf_lambda)
CR = c(.4, .6, .8)

sum_err_var = sum(lambda)^2/CR - sum(lambda)^2
err_var = sum_err_var / 14
```

```
theta1 = diag(err_var[1], nrow = 3)
theta1_pt = diag(err_var[1], nrow = 5)
e1_opt = rmnorm(N, varcov = theta1)
e1_pt = rmnorm(N, varcov = theta1_pt)
e1_fse = rmnorm(N, varcov = theta1)
e1_teamperf = rmnorm(N, varcov = theta1)

theta2 = diag(err_var[2], nrow = 3)
theta2_pt = diag(err_var[2], nrow = 5)
e2_opt = rmnorm(N, varcov = theta1)
e2_pt = rmnorm(N, varcov = theta1_pt)
e2_fse = rmnorm(N, varcov = theta1)
e2_teamperf = rmnorm(N, varcov = theta1)

theta3 = diag(err_var[3], nrow = 3)
theta3_pt = diag(err_var[3], nrow = 5)
e3_opt = rmnorm(N, varcov = theta1)
e3_pt = rmnorm(N, varcov = theta1_pt)
e3_fse = rmnorm(N, varcov = theta1)
e3_teamperf = rmnorm(N, varcov = theta1)
```

Now we generate the data by way of

$$\Lambda \begin{bmatrix}\eta_{1i}\\ \eta_{2i}\\ \eta_{3i}\end{bmatrix} + \begin{bmatrix}e_{0i}\\ e_{1i}\\ e_{2i}\end{bmatrix}$$

```
opt_Lambda = cbind(opt_lambda, opt_lambda, opt_lambda)
pt_Lambda = cbind(pt_lambda, pt_lambda, pt_lambda, pt_lambda, pt_lambda)
fse_Lambda = cbind(fse_lambda, fse_lambda, fse_lambda)
teamperf_Lambda = cbind(teamperf_lambda, teamperf_lambda, teamperf_lambda)

final_opt = tcrossprod(opt_eta, opt_Lambda) + e1_opt
final_opt2 = tcrossprod(opt_eta, opt_Lambda) + e2_opt
final_opt3 = tcrossprod(opt_eta, opt_Lambda) + e3_opt
final_opt = ceiling(final_opt)
final_opt2 = ceiling(final_opt2)
final_opt3 = ceiling(final_opt3)

final_pt = tcrossprod(pt_eta, pt_Lambda) + e1_pt
final_pt2 = tcrossprod(pt_eta, pt_Lambda) + e2_pt
final_pt3 = tcrossprod(pt_eta, pt_Lambda) + e3_pt
final_pt = ceiling(final_pt)
final_pt2 = ceiling(final_pt2)
final_pt3 = ceiling(final_pt3)

final_fse = tcrossprod(fse_eta, fse_Lambda) + e1_fse
final_fse2 = tcrossprod(fse_eta, fse_Lambda) + e2_fse
final_fse3 = tcrossprod(fse_eta, fse_Lambda) + e3_fse
final_fse = ceiling(final_fse)
final_fse2 = ceiling(final_fse2)
final_fse3 = ceiling(final_fse3)
```

```
final_teamperf = tcrossprod(teamperf_eta, teamperf_Lambda) + e1_teamperf
final_teamperf2 = tcrossprod(teamperf_eta, teamperf_Lambda) + e2_teamperf
final_teamperf3 = tcrossprod(teamperf_eta, teamperf_Lambda) + e3_teamperf
final_teamperf = ceiling(final_teamperf)
final_teamperf2 = ceiling(final_teamperf2)
final_teamperf3 = ceiling(final_teamperf3)
```

Then we can combine the simulated data into the final dataset.

```
df1 = cbind(final_opt, final_pt, final_fse, final_teamperf)
df2 = cbind(final_opt2, final_pt2, final_fse2, final_teamperf2)
df3 = cbind(final_opt3, final_pt3, final_fse3, final_teamperf3)
colnames(df1) = c('opt1', 'opt2', 'opt3', 'pt1', 'pt2', 'pt3', 'pt4', 'pt5', 'fse1', 'fse2', 'fse3', 'te
colnames(df2) = c('opt1', 'opt2', 'opt3', 'pt1', 'pt2', 'pt3', 'pt4', 'pt5', 'fse1', 'fse2', 'fse3', 'te
colnames(df3) = c('opt1', 'opt2', 'opt3', 'pt1', 'pt2', 'pt3', 'pt4', 'pt5', 'fse1', 'fse2', 'fse3', 'te
```

Finally, we can run PLS-SEM on the data and fit the measurement and structural models. These are summarized and bootstrapped accordingly for all three composite reliability scores.

```
df1_mm <- constructs(
  composite("COMP",         multi_items("opt", 1:3), weights = mode_A),
  composite("LIKE",   multi_items("pt", 1:5), weights = mode_A),
  composite("CUSA",       multi_items("fse", 1:3), weights = mode_A),
  composite("CUSL",         multi_items("teamperf", 1:3), weights = mode_A)
)

#create structural model
df1_sm <- relationships(
  paths(from = "COMP",        to = c("CUSA", "CUSL")),
  paths(from = "LIKE",  to = c("CUSA", "CUSL")),
  paths(from = "CUSA",      to = "CUSL")
)

#model estimation
df1_pls <- estimate_pls(data = df1,
                         measurement_model = df1_mm,
                         structural_model = df1_sm,
                         inner_weights = path_weighting)
```

```
## Generating the seminr model
## All 100 observations are valid.
```

```
summary(df1_pls)
```

```
##
##  Total Iterations: 45
## Path Coefficients:
##          CUSA   CUSL
## R^2     0.098  0.120
## AdjR^2  0.079  0.092
## COMP   -0.169 -0.315
## LIKE   -0.245 -0.142
## CUSA        . -0.122
##
## Reliability:
##       rhoC   AVE rhoA
## COMP 0.722 0.478    1
## LIKE 0.798 0.449    1
## CUSA 0.680 0.425    1
## CUSL 0.433 0.338    1
```

```
#bootstrap
boot_df1_pls <- bootstrap_model(seminr_model = df1_pls,
                                 nboot = 5000,
                                 cores = 2)
```

```
## Bootstrapping model using seminr...
## SEMinR Model successfully bootstrapped
```

```
summary(boot_df1_pls)
```

```
##
##  Bootstrap resamples: 5000
##
## Bootstrapped Structural Paths:
##                Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## COMP  ->  CUSA        -0.169         -0.166        0.140  -1.213  -0.376
## COMP  ->  CUSL        -0.315         -0.310        0.164  -1.919  -0.504
## LIKE  ->  CUSA        -0.245         -0.209        0.256  -0.957  -0.523
## LIKE  ->  CUSL        -0.142         -0.150        0.230  -0.616  -0.464
## CUSA  ->  CUSL        -0.122         -0.108        0.136  -0.901  -0.352
##                97.5% CI
## COMP  ->  CUSA    0.205
## COMP  ->  CUSL    0.278
## LIKE  ->  CUSA    0.412
## LIKE  ->  CUSL    0.390
## CUSA  ->  CUSL    0.185
##
## Bootstrapped Weights:
##                     Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## opt1  ->  COMP              0.396          0.389        0.271   1.460  -0.283
## opt2  ->  COMP              0.260          0.205        0.305   0.854  -0.610
## opt3  ->  COMP              0.700          0.573        0.300   2.331  -0.530
## pt1  ->  LIKE               0.457          0.318        0.276   1.658  -0.442
## pt2  ->  LIKE               0.348          0.271        0.196   1.772  -0.243
## pt3  ->  LIKE               0.077          0.129        0.172   0.446  -0.275
## pt4  ->  LIKE               0.095          0.150        0.394   0.240  -0.723
## pt5  ->  LIKE               0.396          0.281        0.257   1.543  -0.370
## fse1  ->  CUSA              0.385          0.376        0.263   1.460  -0.312
## fse2  ->  CUSA              0.366          0.318        0.243   1.508  -0.265
## fse3  ->  CUSA              0.716          0.626        0.276   2.600  -0.239
## teamperf1  ->  CUSL         0.153          0.223        0.315   0.484  -0.474
## teamperf2  ->  CUSL         0.026          0.134        0.360   0.073  -0.561
## teamperf3  ->  CUSL         0.982          0.722        0.362   2.708  -0.478
##                     97.5% CI
## opt1  ->  COMP         0.865
## opt2  ->  COMP         0.690
## opt3  ->  COMP         0.906
## pt1  ->  LIKE          0.728
## pt2  ->  LIKE          0.582
## pt3  ->  LIKE          0.428
## pt4  ->  LIKE          0.887
## pt5  ->  LIKE          0.675
## fse1  ->  CUSA         0.804
## fse2  ->  CUSA         0.745
## fse3  ->  CUSA         0.963
## teamperf1  ->  CUSL    0.751
## teamperf2  ->  CUSL    0.794
## teamperf3  ->  CUSL    1.000
##
## Bootstrapped Loadings:
##                     Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## opt1  ->  COMP              0.650          0.605        0.259   2.512  -0.137
## opt2  ->  COMP              0.488          0.400        0.346   1.412  -0.627
## opt3  ->  COMP              0.879          0.745        0.320   2.750  -0.511
## pt1  ->  LIKE               0.785          0.625        0.294   2.676  -0.312
## pt2  ->  LIKE               0.712          0.598        0.265   2.686  -0.199
## pt3  ->  LIKE               0.478          0.448        0.236   2.023  -0.220
## pt4  ->  LIKE               0.550          0.499        0.365   1.508  -0.476
## pt5  ->  LIKE               0.769          0.610        0.267   2.875  -0.207
## fse1  ->  CUSA              0.554          0.523        0.272   2.032  -0.217
## fse2  ->  CUSA              0.539          0.477        0.251   2.148  -0.202
## fse3  ->  CUSA              0.823          0.727        0.270   3.052  -0.174
## teamperf1  ->  CUSL         0.192          0.273        0.361   0.532  -0.542
## teamperf2  ->  CUSL         0.052          0.168        0.396   0.131  -0.616
## teamperf3  ->  CUSL         0.988          0.742        0.366   2.701  -0.482
##                     97.5% CI
## opt1  ->  COMP         0.924
## opt2  ->  COMP         0.827
## opt3  ->  COMP         0.965
## pt1  ->  LIKE          0.887
## pt2  ->  LIKE          0.859
## pt3  ->  LIKE          0.741
## pt4  ->  LIKE          0.917
## pt5  ->  LIKE          0.878
## fse1  ->  CUSA         0.888
## fse2  ->  CUSA         0.840
## fse3  ->  CUSA         0.975
## teamperf1  ->  CUSL    0.826
## teamperf2  ->  CUSL    0.851
## teamperf3  ->  CUSL    0.996
##
## Bootstrapped HTMT:
##                Original Est. Bootstrap Mean Bootstrap SD 2.5% CI 97.5% CI
## COMP  ->  LIKE         0.297          0.395        0.100   0.234    0.631
## COMP  ->  CUSA         0.373          0.598        0.184   0.315    1.027
## COMP  ->  CUSL         1.199          1.097        0.399   0.541    2.042
## LIKE  ->  CUSA         0.454          0.571        0.147   0.351    0.914
## LIKE  ->  CUSL         0.598          0.709        0.244   0.383    1.321
## CUSA  ->  CUSL         0.599          0.871        0.356   0.392    1.738
```

The second model.

```r
df2_mm <- constructs(
  composite("COMP",         multi_items("opt", 1:3), weights = mode_A),
  composite("LIKE",   multi_items("pt", 1:5), weights = mode_A),
  composite("CUSA",       multi_items("fse", 1:3), weights = mode_A),
  composite("CUSL",         multi_items("teamperf", 1:3), weights = mode_A)
)

#create structural model
df2_sm <- relationships(
  paths(from = "COMP",        to = c("CUSA", "CUSL")),
  paths(from = "LIKE",  to = c("CUSA", "CUSL")),
  paths(from = "CUSA",      to = "CUSL")
)

#model estimation
df2_pls <- estimate_pls(data = df2,
                        measurement_model = df2_mm,
                        structural_model = df2_sm,
                        inner_weights = path_weighting)
```

```
## Generating the seminr model
## All 100 observations are valid.
```

```r
summary(df2_pls)
```

```
##
##  Total Iterations: 33
## Path Coefficients:
##         CUSA   CUSL
## R^2    0.070  0.099
## AdjR^2 0.051  0.070
## COMP   0.261  0.113
## LIKE   0.045  0.272
## CUSA       . -0.158
##
## Reliability:
##        rhoC   AVE rhoA
## COMP 0.4689 0.337    1
## LIKE 0.7714 0.414    1
## CUSA 0.7209 0.465    1
## CUSL 0.0115 0.303    1
```

```r
#bootstrap
boot_df2_pls <- bootstrap_model(seminr_model = df2_pls,
                                nboot = 5000,
                                cores = 2)
```

```
## Bootstrapping model using seminr...
## SEMinR Model successfully bootstrapped
```

```r
summary(boot_df2_pls)
```

```
##
##  Bootstrap resamples: 5000
##
## Bootstrapped Structural Paths:
##                Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## COMP  ->  CUSA         0.261          0.216        0.229   1.140  -0.381
## COMP  ->  CUSL         0.113         -0.029        0.196   0.577  -0.372
## LIKE  ->  CUSA         0.045          0.063        0.187   0.241  -0.316
## LIKE  ->  CUSL         0.272          0.017        0.309   0.879  -0.434
## CUSA  ->  CUSL        -0.158         -0.014        0.214  -0.739  -0.379
##                97.5% CI
## COMP  ->  CUSA    0.464
## COMP  ->  CUSL    0.340
## LIKE  ->  CUSA    0.377
## LIKE  ->  CUSL    0.448
## CUSA  ->  CUSL    0.377
##
## Bootstrapped Weights:
##                     Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## opt1  ->  COMP             -0.221          0.080        0.447  -0.494  -0.797
## opt2  ->  COMP              0.233          0.250        0.415   0.561  -0.658
## opt3  ->  COMP              0.985          0.523        0.522   1.887  -0.836
## pt1  ->  LIKE               0.043          0.174        0.311   0.140  -0.476
## pt2  ->  LIKE               0.506          0.247        0.329   1.539  -0.592
## pt3  ->  LIKE               0.307          0.209        0.247   1.242  -0.437
## pt4  ->  LIKE               0.410          0.249        0.322   1.274  -0.602
## pt5  ->  LIKE               0.145          0.165        0.272   0.534  -0.487
## fse1  ->  CUSA              0.311          0.335        0.307   1.013  -0.463
## fse2  ->  CUSA              0.496          0.410        0.322   1.541  -0.425
## fse3  ->  CUSA              0.628          0.398        0.420   1.495  -0.773
## teamperf1  ->  CUSL        -0.049          0.302        0.362  -0.135  -0.532
## teamperf2  ->  CUSL        -0.712          0.235        0.552  -1.290  -0.830
## teamperf3  ->  CUSL         0.796          0.202        0.570   1.396  -0.808
##                     97.5% CI
## opt1  ->  COMP         0.916
## opt2  ->  COMP         0.928
## opt3  ->  COMP         1.023
## pt1  ->  LIKE          0.815
## pt2  ->  LIKE          0.721
## pt3  ->  LIKE          0.585
## pt4  ->  LIKE          0.665
## pt5  ->  LIKE          0.635
## fse1  ->  CUSA         0.859
## fse2  ->  CUSA         0.898
## fse3  ->  CUSA         0.960
## teamperf1  ->  CUSL    0.901
## teamperf2  ->  CUSL    0.983
## teamperf3  ->  CUSL    0.988
##
```

```
## Bootstrapped Loadings:
##                     Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## opt1  ->  COMP              0.058          0.243        0.428   0.135  -0.669
## opt2  ->  COMP              0.313          0.322        0.424   0.739  -0.642
## opt3  ->  COMP              0.954          0.578        0.488   1.957  -0.729
## pt1  ->  LIKE               0.445          0.426        0.345   1.290  -0.508
## pt2  ->  LIKE               0.785          0.487        0.418   1.878  -0.778
## pt3  ->  LIKE               0.623          0.463        0.359   1.737  -0.618
## pt4  ->  LIKE               0.771          0.527        0.449   1.715  -0.801
## pt5  ->  LIKE               0.522          0.409        0.386   1.351  -0.631
## fse1  ->  CUSA              0.579          0.544        0.332   1.741  -0.420
## fse2  ->  CUSA              0.712          0.602        0.337   2.115  -0.366
## fse3  ->  CUSA              0.745          0.513        0.422   1.765  -0.773
## teamperf1  ->  CUSL        -0.229          0.418        0.425  -0.538  -0.589
## teamperf2  ->  CUSL        -0.617          0.377        0.549  -1.124  -0.765
## teamperf3  ->  CUSL         0.690          0.274        0.534   1.292  -0.747
##                     97.5% CI
## opt1  ->  COMP         0.903
## opt2  ->  COMP         0.939
## opt3  ->  COMP         0.991
## pt1  ->  LIKE          0.880
## pt2  ->  LIKE          0.889
## pt3  ->  LIKE          0.820
## pt4  ->  LIKE          0.885
## pt5  ->  LIKE          0.836
## fse1  ->  CUSA         0.926
## fse2  ->  CUSA         0.951
## fse3  ->  CUSA         0.971
## teamperf1  ->  CUSL    0.941
## teamperf2  ->  CUSL    0.970
## teamperf3  ->  CUSL    0.970
##
## Bootstrapped HTMT:
##                Original Est. Bootstrap Mean Bootstrap SD 2.5% CI 97.5% CI
## COMP  ->  LIKE         0.417          0.565        0.171   0.320    0.970
## COMP  ->  CUSA         0.478          0.675        0.235   0.338    1.234
## COMP  ->  CUSL         0.249          0.524        0.194   0.254    0.975
## LIKE  ->  CUSA         0.247          0.400        0.123   0.219    0.691
## LIKE  ->  CUSL         0.349          0.470        0.121   0.277    0.741
## CUSA  ->  CUSL         0.314          0.486        0.167   0.236    0.879
```

The third.

```
#create measurement model matrix
df3_mm <- constructs(
  composite("COMP",         multi_items("opt", 1:3), weights = mode_A),
  composite("LIKE",   multi_items("pt", 1:5), weights = mode_A),
  composite("CUSA",       multi_items("fse", 1:3), weights = mode_A),
  composite("CUSL",         multi_items("teamperf", 1:3), weights = mode_A)
)

#create structural model
df3_sm <- relationships(
  paths(from = "COMP",        to = c("CUSA", "CUSL")),
  paths(from = "LIKE",  to = c("CUSA", "CUSL")),
  paths(from = "CUSA",      to = "CUSL")
)

#model estimation
df3_pls <- estimate_pls(data = df3,
                        measurement_model = df3_mm,
                        structural_model = df3_sm,
                        inner_weights = path_weighting)
```

```
## Generating the seminr model
## All 100 observations are valid.
```

```
summary(df3_pls)
```

```
##
##  Total Iterations: 74
## Path Coefficients:
##          CUSA   CUSL
## R^2     0.162  0.114
## AdjR^2  0.145  0.086
## COMP    0.067  0.304
## LIKE   -0.398  0.044
## CUSA        . -0.138
##
## Reliability:
##          rhoC   AVE rhoA
## COMP 2.05e-01 0.273    1
## LIKE 5.54e-05 0.177    1
## CUSA 5.37e-02 0.326    1
## CUSL 5.43e-01 0.364    1
```

```
#bootstrap
boot_df3_pls <- bootstrap_model(seminr_model = df3_pls,
                                nboot = 5000,
                                cores = 2)
```

```
## Bootstrapping model using seminr...
## SEMinR Model successfully bootstrapped
```

```
summary(boot_df3_pls)
```

```
##
##  Bootstrap resamples: 5000
##
## Bootstrapped Structural Paths:
##                 Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## COMP  ->  CUSA          0.067         -0.041        0.199   0.334  -0.381
## COMP  ->  CUSL          0.304          0.056        0.290   1.048  -0.453
## LIKE  ->  CUSA         -0.398         -0.067        0.339  -1.174  -0.520
## LIKE  ->  CUSL          0.044          0.003        0.216   0.204  -0.387
## CUSA  ->  CUSL        -0.138         -0.128        0.243  -0.570  -0.478
##                97.5% CI
## COMP  ->  CUSA    0.332
## COMP  ->  CUSL    0.454
## LIKE  ->  CUSA    0.499
## LIKE  ->  CUSL    0.399
## CUSA  ->  CUSL    0.389
##
## Bootstrapped Weights:
##                     Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## opt1  ->  COMP             -0.708          0.143        0.566  -1.251  -0.905
## opt2  ->  COMP              0.805          0.239        0.577   1.394  -0.915
## opt3  ->  COMP              0.396          0.244        0.500   0.792  -0.843
## pt1  ->  LIKE               0.306          0.175        0.298   1.028  -0.501
## pt2  ->  LIKE               0.321          0.178        0.301   1.065  -0.509
## pt3  ->  LIKE              -0.693          0.072        0.471  -1.473  -0.756
## pt4  ->  LIKE              -0.533          0.120        0.356  -1.499  -0.636
## pt5  ->  LIKE               0.508          0.207        0.378   1.345  -0.647
## fse1  ->  CUSA             -0.154          0.062        0.512  -0.301  -0.846
## fse2  ->  CUSA              0.910          0.414        0.576   1.580  -0.829
## fse3  ->  CUSA             -0.414          0.067        0.450  -0.919  -0.686
## teamperf1  ->  CUSL         0.947          0.399        0.570   1.662  -0.926
## teamperf2  ->  CUSL         0.157          0.308        0.374   0.419  -0.461
## teamperf3  ->  CUSL         0.138          0.232        0.344   0.402  -0.566
##                     97.5% CI
## opt1  ->  COMP         1.021
## opt2  ->  COMP         1.026
## opt3  ->  COMP         0.973
## pt1  ->  LIKE          0.606
## pt2  ->  LIKE          0.667
## pt3  ->  LIKE          0.814
## pt4  ->  LIKE          0.694
## pt5  ->  LIKE          0.759
## fse1  ->  CUSA         0.957
## fse2  ->  CUSA         0.996
## fse3  ->  CUSA         0.865
## teamperf1  ->  CUSL    0.993
## teamperf2  ->  CUSL    0.928
## teamperf3  ->  CUSL    0.807
##
## Bootstrapped Loadings:
##                     Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## opt1  ->  COMP             -0.389          0.264        0.496  -0.784  -0.758
## opt2  ->  COMP              0.670          0.345        0.534   1.256  -0.794
## opt3  ->  COMP              0.468          0.329        0.517   0.907  -0.833
## pt1  ->  LIKE               0.368          0.393        0.441   0.835  -0.680
## pt2  ->  LIKE               0.211          0.394        0.407   0.517  -0.615
## pt3  ->  LIKE              -0.693          0.192        0.525  -1.319  -0.836
## pt4  ->  LIKE              -0.263          0.385        0.430  -0.612  -0.692
## pt5  ->  LIKE               0.392          0.398        0.441   0.889  -0.717
## fse1  ->  CUSA             -0.123          0.124        0.526  -0.233  -0.810
## fse2  ->  CUSA              0.885          0.440        0.576   1.537  -0.827
## fse3  ->  CUSA             -0.424          0.094        0.486  -0.872  -0.725
## teamperf1  ->  CUSL         0.974          0.457        0.598   1.629  -0.937
## teamperf2  ->  CUSL         0.226          0.390        0.405   0.558  -0.504
## teamperf3  ->  CUSL         0.305          0.371        0.391   0.781  -0.581
##                     97.5% CI
## opt1  ->  COMP         0.967
## opt2  ->  COMP         0.977
## opt3  ->  COMP         0.966
## pt1  ->  LIKE          0.863
## pt2  ->  LIKE          0.848
## pt3  ->  LIKE          0.900
## pt4  ->  LIKE          0.879
## pt5  ->  LIKE          0.872
## fse1  ->  CUSA         0.963
## fse2  ->  CUSA         0.986
## fse3  ->  CUSA         0.892
## teamperf1  ->  CUSL    0.988
## teamperf2  ->  CUSL    0.953
## teamperf3  ->  CUSL    0.887
##
## Bootstrapped HTMT:
##                Original Est. Bootstrap Mean Bootstrap SD 2.5% CI 97.5% CI
## COMP  ->  LIKE         0.216          0.368        0.110   0.201    0.626
## COMP  ->  CUSA         0.340          0.589        0.222   0.277    1.147
## COMP  ->  CUSL         0.449          0.636        0.235   0.311    1.212
## LIKE  ->  CUSA         0.452          0.612        0.183   0.357    1.026
## LIKE  ->  CUSL         0.267          0.451        0.150   0.233    0.815
## CUSA  ->  CUSL         0.805          0.913        0.310   0.474    1.643
```