HW15

108048110

2022-05-25

BACS HW - Week 15

Prerequisite

```
library(seminr)
```

```
security_score <- read.csv("data/security_data_sem.csv")
```

What is SEM?

- SEM, structural equation modeling, aka covariance structural modeling, casual modeling, analysis of moment structures.
- SEM is suitable when the concepts researcher is interested in are complex or multifaceted, it focuses on indirect or mediated, and performs well on modeling casual system.
- Key to represent SEM:
  $X = t + e$
  - X is the thing that we want to measure.
  - t is the true score, the true underlying level of happiness.
  - e is the error term.
    * Systematic error: Bias, the mean of the individual errors does not cancelled out.
    * Random error: $mean(e) = 0$
- One can thought of SEM as path analysis using latent variables.

Latent Variable

- Most social scientific concepts are not directly observable.
    - e.g. Intellegent, Honesty, Happiness, etc.
- This makes them hypothetical or latent.
    - Latent within people at some level that drive attitude and behavior, but not observable.
    - Unable to measure it directly.
- By using latent variables, we are able to cover the full conceptual map, remove and reduce random errir in `measured construct.`

Path analysis

- Diagrammatic representation of a theoretical model using standardized notion.
- Regression equations are specified between measured variables.
    - **Note.** in standard path analysis, we would not use latent variables, instead, the variables that are directly observed.
- Standardized notion

Table 1: standardized notion

ellipse	measured latent variable
rectangle	observed/manifest variable, indicator
circle	error variance, disturbance term
Curved, double headed arroes	covariance/ non-directional path
Straight arrows point in one direction	regression/ directional path

---

PLS-SEM

Partial least squares SEM

For many years, covariance-based structural equation modeling (CB-SEM) was the dominant method for analyzing complex interrelationships between observed and latent variables. The PLS-SEM is appealing to many researchers as it enables them to estimate complex models with many constructs, indicator variables and structural paths without imposing distributional assumptions (i.e. the dependent variable must be approximately normally distributed, and the relationships among the variables are assumed to be casual, linear and additive) on the data.

- PLS-SEM is a causal-predictive approach to SEM that emphasizes prediction in estimating statistical models, whose structures are designed to provide casual explanations.

When to use PLS-SEM? The most prominent justifications for using PLS-SEM are attributed to…

- Non-normal data

  Data collected for social science research often fails to follow a multivariate normal distribution.

  When attempting to evaluate a path model using CB-SEM, non-normal data can lead to underestimated standard errors and inflated goodness-of-fit measures.

- Small sample sizes

  Different from CB-SEM, PLS-SEM can be utilized with much smaller sample sizes, even when models are highly complex.

  In these situations, PLS-SEM generally achieves higher levels of statistical power and demonstrates much better convergence behavior than CB-SEM.

- Formative Indicators

  Analyzing formative indicators with CB-SEM often leads to identification problems (Jarvis *et al.*, 2003). However, formative indicators should be approached with caution when using PLS-SEM.

  Researchers should be aware that the evaluation of formatively measured constructs relies on a totally different set of criteria compared to their reflective counterparts. Researchers should apply the most recent set of evaluation criteria when examining the validity of formatively measured constructs.

To sum up... Depending on the specific empirical context and objectives of the study, PLS-SEM's distinctive methodological features make it a particularly valuable and potentially better-suited alternative to the more popular CB-SEM approaches in practical applications.

Generally, however, neither method is superior to the other overall. Rather, the selection of the proper method depends on the objective of the study.

Question 1) Composite Path Models using PLS-PM

a. Create a PLS path model

i. Measurement model

- All constructs are measured as composites.

```
sec_mm <- constructs(
  composite("TRUST", multi_items("TRST", 1:4)),
  composite("SEC", multi_items("PSEC", 1:4)),
  composite("REP", multi_items("PREP", 1:4)),
  composite("INV", multi_items("PINV", 1:3)),
  composite("POL", multi_items("PPSS", 1:3)),
  composite("FAML", single_item("FAML1")),
  interaction_term(iv="REP", moderator="POL", method=orthogonal)
  )
```

ii. Structural Model

- Paths between constructs

```
sec_sm <- relationships(
  paths(from=c("REP","INV","POL","FAML", "REP*POL"),
        to = "SEC"),
  paths(from="SEC",
        to = "TRUST")
)
```

b. Showing the result

```r
sec_pls <- estimate_pls(
  data = security_score,
  measurement_model = sec_mm,
  structural_model = sec_sm
)
```

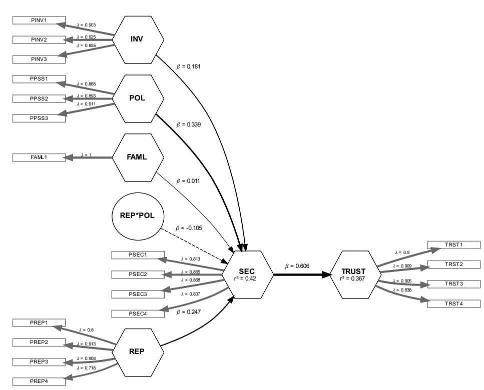
```
## Generating the seminr model

## All 405 observations are valid.
```

```r
sec_pls$path_coef
```

```
##         REP INV POL FAML REP*POL         SEC     TRUST
## REP       0   0   0    0       0  0.24700650 0.0000000
## INV       0   0   0    0       0  0.18065379 0.0000000
## POL       0   0   0    0       0  0.33859098 0.0000000
## FAML      0   0   0    0       0  0.01050482 0.0000000
## REP*POL   0   0   0    0       0 -0.10464613 0.0000000
## SEC       0   0   0    0       0  0.00000000 0.6056388
## TRUST     0   0   0    0       0  0.00000000 0.0000000
```

```r
sec_pls_summary <- summary(sec_pls)
```

```r
plot(sec_pls)
```



***i.* Plot a figure of the estimated model.**

```r
# weights
sec_pls_summary$weights
```

*** ii.* Weights and loadings of composites.**

```
##                REP   INV   POL  FAML REP*POL   SEC TRUST
## TRST1        0.000 0.000 0.000 0.000   0.000 0.000 0.282
## TRST2        0.000 0.000 0.000 0.000   0.000 0.000 0.280
## TRST3        0.000 0.000 0.000 0.000   0.000 0.000 0.286
## TRST4        0.000 0.000 0.000 0.000   0.000 0.000 0.278
## PSEC1        0.000 0.000 0.000 0.000   0.000 0.277 0.000
## PSEC2        0.000 0.000 0.000 0.000   0.000 0.315 0.000
## PSEC3        0.000 0.000 0.000 0.000   0.000 0.307 0.000
## PSEC4        0.000 0.000 0.000 0.000   0.000 0.292 0.000
## PREP1        0.215 0.000 0.000 0.000   0.000 0.000 0.000
## PREP2        0.334 0.000 0.000 0.000   0.000 0.000 0.000
## PREP3        0.349 0.000 0.000 0.000   0.000 0.000 0.000
## PREP4        0.287 0.000 0.000 0.000   0.000 0.000 0.000
## PINV1        0.000 0.363 0.000 0.000   0.000 0.000 0.000
## PINV2        0.000 0.395 0.000 0.000   0.000 0.000 0.000
## PINV3        0.000 0.358 0.000 0.000   0.000 0.000 0.000
## PPSS1        0.000 0.000 0.360 0.000   0.000 0.000 0.000
## PPSS2        0.000 0.000 0.395 0.000   0.000 0.000 0.000
## PPSS3        0.000 0.000 0.367 0.000   0.000 0.000 0.000
## FAML1        0.000 0.000 0.000 1.000   0.000 0.000 0.000
## PREP1*PPSS1 0.000 0.000 0.000 0.000   0.239 0.000 0.000
## PREP1*PPSS2 0.000 0.000 0.000 0.000   0.031 0.000 0.000
## PREP1*PPSS3 0.000 0.000 0.000 0.000   0.021 0.000 0.000
## PREP2*PPSS1 0.000 0.000 0.000 0.000   0.046 0.000 0.000
## PREP2*PPSS2 0.000 0.000 0.000 0.000  -0.104 0.000 0.000
## PREP2*PPSS3 0.000 0.000 0.000 0.000  -0.228 0.000 0.000
## PREP3*PPSS1 0.000 0.000 0.000 0.000  -0.341 0.000 0.000
## PREP3*PPSS2 0.000 0.000 0.000 0.000   0.095 0.000 0.000
## PREP3*PPSS3 0.000 0.000 0.000 0.000   0.108 0.000 0.000
## PREP4*PPSS1 0.000 0.000 0.000 0.000   0.443 0.000 0.000
## PREP4*PPSS2 0.000 0.000 0.000 0.000   0.382 0.000 0.000
## PREP4*PPSS3 0.000 0.000 0.000 0.000   0.271 0.000 0.000
```

- **Adding these weights together gets the scores of the entire construct.**

```r
# loadings
sec_pls_summary$loadings
```

```
##                REP    INV    POL   FAML REP*POL    SEC  TRUST
## TRST1        0.000  0.000  0.000  0.000  -0.000  0.000  0.900
## TRST2        0.000  0.000  0.000  0.000  -0.000  0.000  0.909
## TRST3        0.000  0.000  0.000  0.000  -0.000  0.000  0.905
## TRST4        0.000  0.000  0.000  0.000  -0.000  0.000  0.838
## PSEC1        0.000  0.000  0.000  0.000  -0.000  0.813  0.000
## PSEC2        0.000  0.000  0.000  0.000  -0.000  0.865  0.000
## PSEC3        0.000  0.000  0.000  0.000  -0.000  0.868  0.000
```

```
## PSEC4        0.000  0.000  0.000  0.000  -0.000  0.807  0.000
## PREP1        0.800  0.000  0.000  0.000   0.000  0.000  0.000
## PREP2        0.913  0.000  0.000  0.000   0.000  0.000  0.000
## PREP3        0.908  0.000  0.000  0.000   0.000  0.000  0.000
## PREP4        0.718  0.000  0.000  0.000   0.000  0.000  0.000
## PINV1        0.000  0.903  0.000  0.000  -0.000  0.000  0.000
## PINV2        0.000  0.925  0.000  0.000  -0.000  0.000  0.000
## PINV3        0.000  0.855  0.000  0.000  -0.000  0.000  0.000
## PPSS1        0.000  0.000  0.868  0.000   0.000  0.000  0.000
## PPSS2        0.000  0.000  0.893  0.000   0.000  0.000  0.000
## PPSS3        0.000  0.000  0.911  0.000   0.000  0.000  0.000
## FAML1        0.000  0.000  0.000  1.000  -0.000  0.000  0.000
## PREP1*PPSS1 -0.000 -0.000 -0.000 -0.000   0.581 -0.000 -0.000
## PREP1*PPSS2 -0.000 -0.000  0.000 -0.000   0.510 -0.000 -0.000
## PREP1*PPSS3 -0.000 -0.000 -0.000 -0.000   0.506 -0.000 -0.000
## PREP2*PPSS1 -0.000 -0.000 -0.000 -0.000   0.509 -0.000 -0.000
## PREP2*PPSS2 -0.000 -0.000  0.000 -0.000   0.421  0.000  0.000
## PREP2*PPSS3 -0.000 -0.000 -0.000  0.000   0.336  0.000  0.000
## PREP3*PPSS1 -0.000 -0.000 -0.000  0.000   0.236  0.000  0.000
## PREP3*PPSS2 -0.000 -0.000  0.000 -0.000   0.555 -0.000 -0.000
## PREP3*PPSS3 -0.000 -0.000 -0.000  0.000   0.466 -0.000 -0.000
## PREP4*PPSS1  0.000 -0.000  0.000  0.000   0.900 -0.000 -0.000
## PREP4*PPSS2 -0.000 -0.000 -0.000 -0.000   0.836 -0.000  0.000
## PREP4*PPSS3  0.000 -0.000  0.000  0.000   0.859 -0.000  0.000
```

- **The correlation between the composite and the construct.**

```
# VIF
sec_pls_summary$validity$vif_items
```

```
## REP :
## PREP1 PREP2 PREP3 PREP4
## 2.106 3.775 3.285 1.391
##
## INV :
## PINV1 PINV2 PINV3
## 2.831 3.112 1.924
##
## POL :
## PPSS1 PPSS2 PPSS3
## 2.091 2.319 2.744
##
## FAML :
## FAML1
##     1
##
## REP*POL :
## PREP1*PPSS1 PREP1*PPSS2 PREP1*PPSS3 PREP2*PPSS1 PREP2*PPSS2 PREP2*PPSS3
##      10.723       3.480       8.948      10.335       9.085      10.892
## PREP3*PPSS1 PREP3*PPSS2 PREP3*PPSS3 PREP4*PPSS1 PREP4*PPSS2 PREP4*PPSS3
##       2.326       8.778       7.969       4.072       2.787       4.306
##
## SEC :
```

```
## PSEC1 PSEC2 PSEC3 PSEC4
## 1.893 2.344 2.362 1.811
##
## TRUST :
## TRST1 TRST2 TRST3 TRST4
## 3.346 3.503 3.117 2.098
```

```r
# where the path fits the R^2
sec_pls_summary$paths
```

***iii.* Regression coefficients of paths between factors.**

```
##            SEC TRUST
## R^2      0.420 0.367
## AdjR^2   0.412 0.365
## REP      0.247     .
## INV      0.181     .
## POL      0.339     .
## FAML     0.011     .
## REP*POL -0.105     .
## SEC          . 0.606
```

```r
sec_pls_summary$descriptives$statistics$constructs
```

```
##           No. Missing   Mean Median    Min   Max Std.Dev. Kurtosis Skewness
## REP     1.000   0.000 -0.000  0.315 -4.049 0.871    1.000    4.748   -1.479
## INV     2.000   0.000 -0.000  0.276 -3.767 1.085    1.000    2.946   -0.760
## POL     3.000   0.000  0.000  0.345 -3.700 1.019    1.000    4.184   -1.057
## FAML    4.000   0.000 -0.000  0.234 -3.346 0.831    1.000    4.573   -1.405
## REP*POL 5.000   0.000  0.000 -0.007 -7.022 6.058    1.042   12.207   -0.370
## SEC     6.000   0.000  0.000 -0.065 -3.164 1.404    1.000    2.566   -0.304
## TRUST   7.000   0.000  0.000  0.234 -3.271 1.151    1.000    3.098   -0.801
```

```r
sec_pls_summary$descriptives$correlations$constructs
```

```
##           REP    INV   POL   FAML REP*POL    SEC  TRUST
## REP     1.000  0.611 0.471  0.553   0.000  0.523  0.616
## INV     0.611  1.000 0.434  0.463  -0.105  0.495  0.503
## POL     0.471  0.434 1.000  0.555   0.000  0.539  0.409
## FAML    0.553  0.463 0.555  1.000  -0.043  0.423  0.450
## REP*POL 0.000 -0.105 0.000 -0.043   1.000 -0.128 -0.056
## SEC     0.523  0.495 0.539  0.423  -0.128  1.000  0.606
## TRUST   0.616  0.503 0.409  0.450  -0.056  0.606  1.000
```

```r
boot_sec_pls <- bootstrap_model(
  seminr_model = sec_pls,
  nboot = 1000,
  seed = 42
)
```

iv. Bootstrapped path coefficients: t-values, 95% CI.

```
## Bootstrapping model using seminr...

## SEMinR Model successfully bootstrapped
```

```r
# a matrix of the bootstrap path coefficients and standard deviations
boot_sec_pls$paths_descriptives
```

```
##         SEC PLS Est. TRUST PLS Est. SEC Boot Mean TRUST Boot Mean SEC Boot SD
## REP            0.247          0.000         0.242           0.000       0.059
## INV            0.181          0.000         0.188           0.000       0.058
## POL            0.339          0.000         0.342           0.000       0.056
## FAML           0.011          0.000         0.010           0.000       0.058
## REP*POL       -0.105          0.000        -0.025           0.000       0.123
## SEC            0.000          0.606         0.000           0.610       0.000
##         TRUST Boot SD
## REP             0.000
## INV             0.000
## POL             0.000
## FAML            0.000
## REP*POL         0.000
## SEC             0.034
```

```r
boot_sec_pls$total_paths_descriptives
```

```
##         SEC PLS Est. TRUST PLS Est. SEC Boot Mean TRUST Boot Mean SEC Boot SD
## REP            0.247          0.150         0.242           0.148       0.059
## INV            0.181          0.109         0.188           0.115       0.058
## POL            0.339          0.205         0.342           0.209       0.056
## FAML           0.011          0.006         0.010           0.006       0.058
## REP*POL       -0.105         -0.063        -0.025          -0.015       0.123
## SEC            0.000          0.606         0.000           0.610       0.000
##         TRUST Boot SD
## REP             0.038
## INV             0.036
## POL             0.036
## FAML            0.036
## REP*POL         0.075
## SEC             0.034
```

```r
boot_sec_pls_summary <- summary(boot_sec_pls)
#reports a matrix of direct paths and their standard deviation, t_values, and confidence intervals.
boot_sec_pls_summary$bootstrapped_paths
```

```
##                   Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## REP  ->  SEC              0.247          0.242        0.059   4.190   0.120
## INV  ->  SEC              0.181          0.188        0.058   3.109   0.075
## POL  ->  SEC              0.339          0.342        0.056   6.077   0.230
## FAML  ->  SEC             0.011          0.010        0.058   0.181  -0.101
## REP*POL  ->  SEC         -0.105         -0.025        0.123  -0.854  -0.190
```

```
## SEC   ->  TRUST           0.606          0.610        0.034  17.739   0.540
##                  97.5% CI
## REP   ->  SEC        0.353
## INV   ->  SEC        0.303
## POL   ->  SEC        0.455
## FAML   ->  SEC       0.125
## REP*POL  ->  SEC     0.191
## SEC   ->  TRUST      0.680
```

```
#reports a matrix of total paths and their standard deviation, t_values, and confidence intervals.
boot_sec_pls_summary$bootstrapped_total_paths
```

```
##                    Original Est. Bootstrap Mean Bootstrap SD T Stat. 2.5% CI
## REP   ->  SEC              0.247          0.242        0.059   4.190   0.120
## REP   ->  TRUST            0.150          0.148        0.038   3.888   0.071
## INV   ->  SEC              0.181          0.188        0.058   3.109   0.075
## INV   ->  TRUST            0.109          0.115        0.036   3.033   0.047
## POL   ->  SEC              0.339          0.342        0.056   6.077   0.230
## POL   ->  TRUST            0.205          0.209        0.036   5.630   0.140
## FAML   ->  SEC             0.011          0.010        0.058   0.181  -0.101
## FAML   ->  TRUST           0.006          0.006        0.036   0.179  -0.062
## REP*POL  ->  SEC          -0.105         -0.025        0.123  -0.854  -0.190
## REP*POL  ->  TRUST        -0.063         -0.015        0.075  -0.846  -0.116
## SEC   ->  TRUST            0.606          0.610        0.034  17.739   0.540
##                    97.5% CI
## REP   ->  SEC          0.353
## REP   ->  TRUST        0.222
## INV   ->  SEC          0.303
## INV   ->  TRUST        0.187
## POL   ->  SEC          0.455
## POL   ->  TRUST        0.281
## FAML   ->  SEC         0.125
## FAML   ->  TRUST       0.079
## REP*POL  ->  SEC       0.191
## REP*POL  ->  TRUST     0.118
## SEC   ->  TRUST        0.680
```

```
# evaluate the mean estimate, standard deviation, t_value and confidence intervals for specific paths
# from sec to trust
specific_effect_significance(boot_sec_pls,
                             from = "SEC",
                             to = "TRUST",
                             alpha = 0.05
                             )
```

```
##   Original Est. Bootstrap Mean   Bootstrap SD        T Stat.        2.5% CI
##      0.60563883     0.60994262     0.03414127    17.73919930     0.54011550
##        97.5% CI
##      0.67987022
```

Question 2) Common-Factor Models using CB-SEM

a. Create a common factor model

```r
sec_mm_reflective <- as.reflective(sec_mm)
```

i. Convert your earlier measurement model to being entirely reflective.

- **But we often wish to conduct a CFA of our measurement model prior to CBSEM**.

```r
sec_cfa <- estimate_cfa(data=security_score,
                        sec_mm_reflective)
```

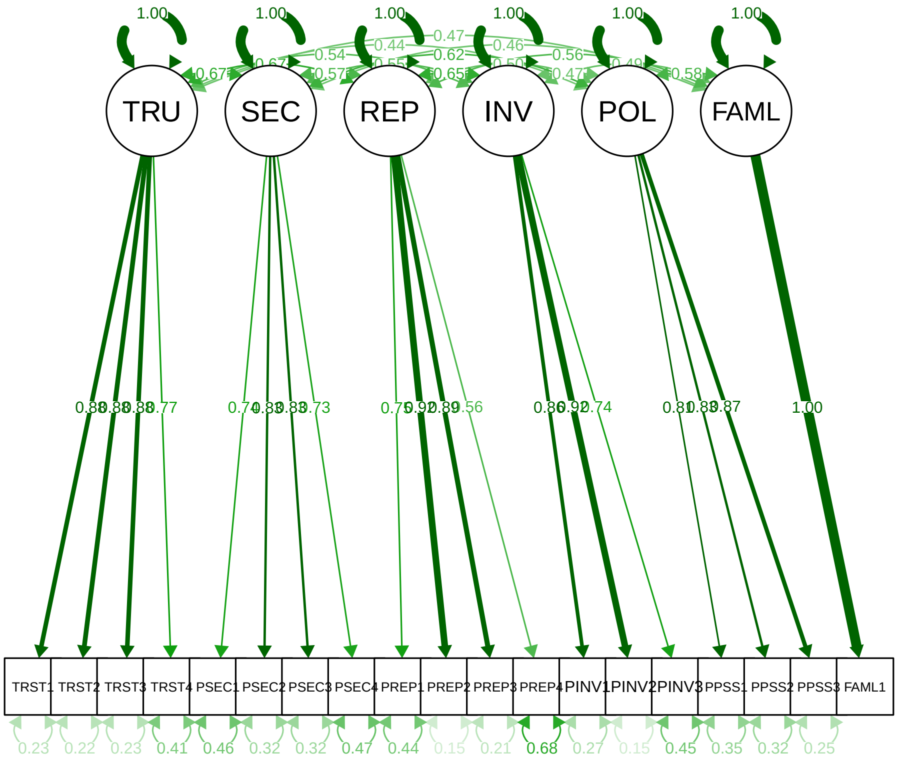
```
## Generating the seminr model for CFA
```

```r
sec_cfa_summary <- summary(sec_cfa)
sec_cfa_summary$descriptives$correlations$constructs
```

```
##       TRUST SEC   REP   INV   POL   FAML
## TRUST 1.000
## SEC   0.667 1.000
## REP   0.668 0.572 1.000
## INV   0.542 0.554 0.652 1.000
## POL   0.442 0.615 0.496 0.475 1.000
## FAML  0.466 0.457 0.561 0.493 0.584 1.000
```

```r
plot(sec_cfa)
```

```
## Plotting of lavaan models using semPlot.
```




```
## NULL
```

```
sec_cbsem <- estimate_cbsem(data=security_score,
                            sec_mm_reflective,
                            sec_sm)
```

***ii.* Use the same structural model as before.**

```
## Generating the seminr model for CBSEM
```

```
sec_cbsem_summary <- summary(sec_cbsem)
sec_cbsem_summary$descriptives$correlations$constructs
```

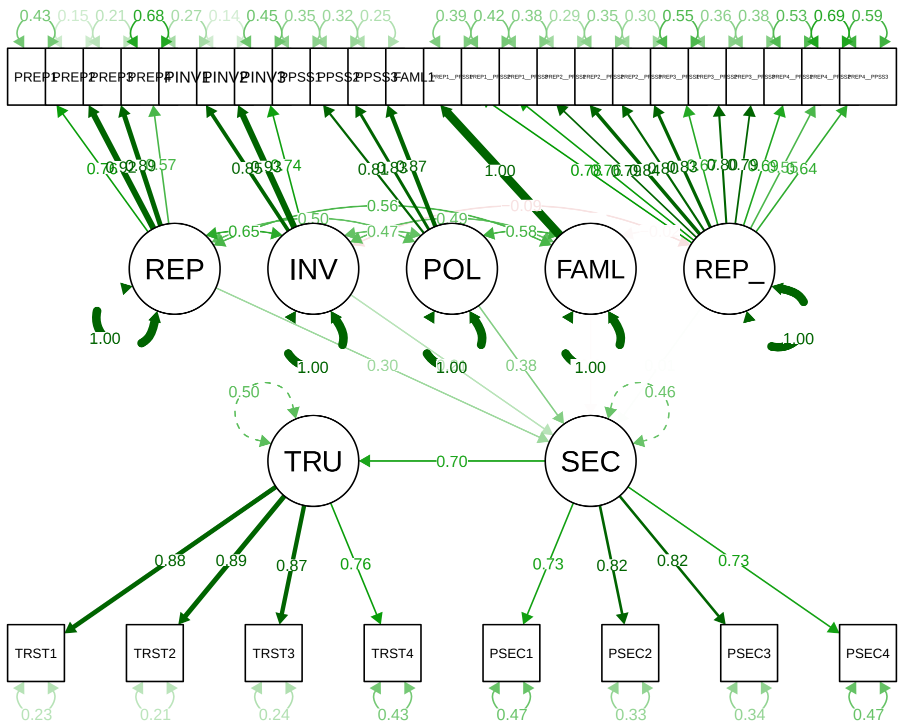
```
##             TRUST   SEC    REP    INV    POL    FAML   REP__P
## TRUST       1.000
## SEC         0.704  1.000
## REP         0.437  0.621  1.000
## INV         0.410  0.583  0.653  1.000
## POL         0.437  0.621  0.496  0.473  1.000
## FAML        0.341  0.485  0.562  0.493  0.584  1.000
## REP_x_POL -0.008 -0.011  0.000 -0.093  0.000 -0.022  1.000
```

b. Showing the result

```r
# CB-SEM
plot(sec_cbsem)
```

***i.* Plot a figure of the estimated model.**

```
## Plotting of lavaan models using semPlot.
```




```
## NULL
```

```r
sec_cbsem_summary$loadings$significance
```

***ii.* Show the loadings of composites.**

```
##                               Std Estimate         SE      t-Value   2.5% CI
## TRUST -> TRST1                   0.8800240 0.02272091 0.000000e+00 0.8354919
## TRUST -> TRST2                   0.8886342 0.03330783 0.000000e+00 0.8233521
## TRUST -> TRST3                   0.8690644 0.03749444 0.000000e+00 0.7955767
```

```
## TRUST -> TRST4                 0.7575988 0.04846748 0.000000e+00 0.6626042
## SEC -> PSEC1                   0.7308766 0.03679205 0.000000e+00 0.6587655
## SEC -> PSEC2                   0.8173481 0.04480183 0.000000e+00 0.7295381
## SEC -> PSEC3                   0.8151708 0.03728082 0.000000e+00 0.7421017
## SEC -> PSEC4                   0.7260444 0.03811841 0.000000e+00 0.6513337
## REP -> PREP1                   0.7551328 0.04464916 0.000000e+00 0.6676220
## REP -> PREP2                   0.9199208 0.02635333 0.000000e+00 0.8682692
## REP -> PREP3                   0.8871362 0.04015103 0.000000e+00 0.8084416
## REP -> PREP4                   0.5650059 0.04585583 0.000000e+00 0.4751302
## INV -> PINV1                   0.8520004 0.04489927 0.000000e+00 0.7639994
## INV -> PINV2                   0.9257476 0.04556425 0.000000e+00 0.8364433
## INV -> PINV3                   0.7388750 0.04511601 0.000000e+00 0.6504492
## POL -> PPSS1                   0.8051533 0.04355300 0.000000e+00 0.7197910
## POL -> PPSS2                   0.8272576 0.02807169 0.000000e+00 0.7722381
## POL -> PPSS3                   0.8674335 0.03273664 0.000000e+00 0.8032708
## FAML -> FAML1                  1.0000000 0.00000000           NA 1.0000000
## REP_x_POL -> PREP1_x_PPSS1    0.7781584 0.05799871 0.000000e+00 0.6644831
## REP_x_POL -> PREP1_x_PPSS2    0.7597768 0.05931838 0.000000e+00 0.6435149
## REP_x_POL -> PREP1_x_PPSS3    0.7879106 0.05013554 0.000000e+00 0.6896467
## REP_x_POL -> PREP2_x_PPSS1    0.8447368 0.03649041 0.000000e+00 0.7732169
## REP_x_POL -> PREP2_x_PPSS2    0.8034561 0.03639411 0.000000e+00 0.7321250
## REP_x_POL -> PREP2_x_PPSS3    0.8342444 0.03536430 0.000000e+00 0.7649317
## REP_x_POL -> PREP3_x_PPSS1    0.6736451 0.12948899 1.967998e-07 0.4198514
## REP_x_POL -> PREP3_x_PPSS2    0.8011944 0.03780427 0.000000e+00 0.7270994
## REP_x_POL -> PREP3_x_PPSS3    0.7902063 0.06416741 0.000000e+00 0.6644405
## REP_x_POL -> PREP4_x_PPSS1    0.6854770 0.06906812 0.000000e+00 0.5501059
## REP_x_POL -> PREP4_x_PPSS2    0.5531922 0.06212434 0.000000e+00 0.4314307
## REP_x_POL -> PREP4_x_PPSS3    0.6405843 0.05794028 0.000000e+00 0.5270235
##                             97.5% CI
## TRUST -> TRST1              0.9245562
## TRUST -> TRST2              0.9539164
## TRUST -> TRST3              0.9425522
## TRUST -> TRST4              0.8525933
## SEC -> PSEC1                0.8029877
## SEC -> PSEC2                0.9051581
## SEC -> PSEC3                0.8882399
## SEC -> PSEC4                0.8007551
## REP -> PREP1                0.8426435
## REP -> PREP2                0.9715724
## REP -> PREP3                0.9658307
## REP -> PREP4                0.6548817
## INV -> PINV1                0.9400013
## INV -> PINV2                1.0150518
## INV -> PINV3                0.8273007
## POL -> PPSS1                0.8905156
## POL -> PPSS2                0.8822771
## POL -> PPSS3                0.9315961
## FAML -> FAML1               1.0000000
## REP_x_POL -> PREP1_x_PPSS1 0.8918338
## REP_x_POL -> PREP1_x_PPSS2 0.8760387
## REP_x_POL -> PREP1_x_PPSS3 0.8861744
## REP_x_POL -> PREP2_x_PPSS1 0.9162567
## REP_x_POL -> PREP2_x_PPSS2 0.8747873
## REP_x_POL -> PREP2_x_PPSS3 0.9035572
```

```
## REP_x_POL -> PREP3_x_PPSS1 0.9274389
## REP_x_POL -> PREP3_x_PPSS2 0.8752894
## REP_x_POL -> PREP3_x_PPSS3 0.9159721
## REP_x_POL -> PREP4_x_PPSS1 0.8208480
## REP_x_POL -> PREP4_x_PPSS2 0.6749536
## REP_x_POL -> PREP4_x_PPSS3 0.7541452
```

```
sec_cbsem_summary$paths$significance
```

iii. **Show the regression coefficients of paths between factors, and their p-values.**

```
##                   Std Estimate         SE      t-Value     2.5% CI   97.5% CI
## SEC -> REP         0.299536782 0.07273355 3.817182e-05  0.15698165 0.44209191
## SEC -> INV         0.214253245 0.07345058 3.534482e-03  0.07029275 0.35821374
## SEC -> POL         0.376401499 0.06413246 4.380975e-09  0.25070419 0.50209881
## SEC -> FAML       -0.008837653 0.07010617 8.996836e-01 -0.14624321 0.12856791
## SEC -> REP_x_POL   0.008355287 0.04468802 8.516847e-01 -0.07923162 0.09594219
## TRUST -> SEC       0.703639369 0.03721629 0.000000e+00  0.63069677 0.77658197
```