Confirmatory Factor Analysis (CFA)

Alireza Sheikh-Zadeh, Ph.D.

This module will consider confirmatory factor analysis models in which particular manifest variables can relate to specific factors while other manifest variables are constrained to have zero loadings on some of the factors.

CFA versus EFA

- A confirmatory factor analysis (CFA) model may arise from theoretical considerations or be based on the results of an exploratory factor analysis where the investigator might wish to postulate a specific model for a new set of similar data.
- In exploratory factor analysis (EFA),
    - The study will determine which observed variables are highly correlated with the common factors and how many common factors are needed to adequately describe the data.
    - The loading matrix was nonzero for all factors related to all variables.
    - No constraints are placed on which manifest variables load on which factors.
    - We do not have a theory that says specific variables are related to a particular factor.
- In EFA, factors are assumed to be uncorrelated (before the rotation), while in CFA, we allow that the factors are correlated.
- Both EFA and CFA use maximum likelihood for their estimation. The main difference is that you have a theory in CFA, and you try to confirm it.

What was the EFA model for two factors:

$$X_i = \lambda_{i1} f_1 + \lambda_{i2} f_2 + u_i, \text{ for } i = 1,2, \dots, q$$

Where i is the index of variables.

CFA looks the same. However, in CFA, you are assigning subsets of variables to particular factors. For example, you may say variables X_1, X_2, and X_3 are driven by f_1 and variables X_4 and X_5 are driven by f_2 (Note: both factor and manifest variables are scaled).

$$X_1 = \lambda_1 f_1 + u_1$$

$$X_2 = \lambda_2 f_1 + u_2$$

$$X_3 = \lambda_3 f_1 + u_3$$

$$X_4 = \lambda_4 f_2 + u_4$$

$$X_5 = \lambda_5 f_2 + u_5$$

The model is based on your theory. You may come up with that theory in an exploratory way. You may design a questionnaire, and you say all of these three questions (X_1, X_2, X_3) are measuring the perceived usefulness (f_1) and (X_4, X_5) are measuring the perceived ease-of-use of a software product (f_2).

Example: Ability data (page 206 EH textbook). Latent variables are defined based on a theory.

```
# Like PCA and EFA, in CFA, all we need is the correlation (or covariance)
matrix of the data as an input.
# Here, we create the given correlation matrix of ability data for CFA
ab <- c(0.73,
        0.70, 0.68,
        0.58, 0.61, 0.57,
        0.46, 0.43, 0.40, 0.37,
        0.56, 0.52, 0.48, 0.41, 0.72)
cov.ability <- diag(6)/2
cov.ability[upper.tri(cov.ability)] <- ab
cov.ability <- cov.ability + t(cov.ability)
rownames(cov.ability) <- colnames(cov.ability) <-
    c("SCA","PPE","PTE","PFE","EA","CP")
cov.ability # We use this correlation matrix to apply CFA

##      SCA  PPE  PTE  PFE   EA   CP
## SCA 1.00 0.73 0.70 0.58 0.46 0.56
## PPE 0.73 1.00 0.68 0.61 0.43 0.52
## PTE 0.70 0.68 1.00 0.57 0.40 0.48
## PFE 0.58 0.61 0.57 1.00 0.37 0.41
## EA  0.46 0.43 0.40 0.37 1.00 0.72
## CP  0.56 0.52 0.48 0.41 0.72 1.00

# SCA: self-concept of ability;
# PPE: perceived parental evaluation;
# PTE: perceived teacher evaluation;
# PFE: perceived friend's evaluation;
# EA: educational aspiration;
# CP: college plans.

# Calsyn and Kenny (1977) postulated that two underlying latent variables,
ability, and aspiration, generated the relationships between the observed
variables.

# where f1 represents the ability latent variable
# and f2 represents the aspiration latent variable

# The first four of the manifest variables were assumed to be indicators of
ability and the last two indicators of aspiration;
```

```r
# install.packages("sem")
# install.packages("semPlot")
library("sem")
# The model is specified via arrows
# The text consists of three columns.
# The first one corresponds to an arrow
# specification where single-headed or directional arrows correspond to regre
ssion
# coefficients and double-headed arrows correspond to variance parameters. Th
e second column denotes parameter names, and the third one assigns values to
fixed parameters.
# Further details are available from the sem package documentation.
ability_model <- specifyModel(text = "
Ability     -> SCA, lambda1, NA
Ability     -> PPE, lambda2, NA
Ability     -> PTE, lambda3, NA
Ability     -> PFE, lambda4, NA
Aspiration  -> EA, lambda5, NA
Aspiration  -> CP, lambda6, NA
Ability    <-> Aspiration, rho, NA
SCA        <-> SCA, theta1, NA
PPE        <-> PPE, theta2, NA
PTE        <-> PTE, theta3, NA
PFE        <-> PFE, theta4, NA
EA         <-> EA, theta5, NA
CP         <-> CP, theta6, NA
Ability    <-> Ability, NA, 1
Aspiration <-> Aspiration, NA, 1
")

ability_sem <- sem(ability_model, cov.ability, 556)

summary(ability_sem)

##
##  Model Chisquare =  9.255732   Df =  8 Pr(>Chisq) = 0.3211842
##  AIC =  35.25573
##  BIC =  -41.31041
##
##  Normalized Residuals
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.4409685 -0.1870306 -0.0000018 -0.0130992  0.2107128  0.5333068
##
##  R-square for Endogenous Variables
##    SCA    PPE    PTE    PFE     EA     CP
## 0.7451 0.7213 0.6482 0.4834 0.6008 0.8629
##
##  Parameter Estimates
##          Estimate  Std Error  z value   Pr(>|z|)
## lambda1 0.8632049 0.03514508 24.561188 3.284552e-133
```

```
## lambda2 0.8493226 0.03545022 23.958178 7.593661e-127
## lambda3 0.8050861 0.03640470 22.114892 2.272503e-108
## lambda4 0.6952671 0.03863370 17.996387  2.079489e-72
## lambda5 0.7750850 0.04035675 19.205834  3.307658e-82
## lambda6 0.9289304 0.03940959 23.571177 7.615270e-123
## rho     0.6663697 0.03095414 21.527645 8.578257e-103
## theta1  0.2548772 0.02336722 10.907470  1.061704e-27
## theta2  0.2786512 0.02412754 11.549097  7.460043e-31
## theta3  0.3518366 0.02691875 13.070321  4.865973e-39
## theta4  0.5166036 0.03472534 14.876847  4.659431e-50
## theta5  0.3992432 0.03819583 10.452535  1.426604e-25
## theta6  0.1370884 0.04350459  3.151126  1.626425e-03
##
## lambda1 SCA <--- Ability
## lambda2 PPE <--- Ability
## lambda3 PTE <--- Ability
## lambda4 PFE <--- Ability
## lambda5 EA <--- Aspiration
## lambda6 CP <--- Aspiration
## rho     Aspiration <--> Ability
## theta1  SCA <--> SCA
## theta2  PPE <--> PPE
## theta3  PTE <--> PTE
## theta4  PFE <--> PFE
## theta5  EA <--> EA
## theta6  CP <--> CP
##
##  Iterations =  29

# Of particular note amongst the parameter estimates is the correlation betwe
en "true" ability and "true" aspiration; this is known as a 'disattenuated' c
orrelation. In this case, the estimate is rho = 0.666 with a standard error o
f 0.031. An approximate 95% confidence interval for the disattenuated correla
tion is [0.606; 0.727].
```

Estimation and discrepancy function

We have the actual cov (or corr) matrix and an estimate cov (or corr) matrix based on the model. The goodness of fit of the model depends on the discrepancy between these two cov matrices. The more similar the cov matrices the better model. If those matrices are drastically different, it says the model is not right.

We can compare the restricted cov (or corr) matrix versus the non-restricted (or original) cov matrix.

```
# restricted Cor matrix
ability_sem$C
```

```
##            SCA       PPE       PTE       PFE        EA        CP
## SCA 1.0000000 0.7331395 0.6949543 0.6001580 0.4458394 0.5343334
## PPE 0.7331395 1.0000001 0.6837778 0.5905061 0.4386693 0.5257401
## PTE 0.6949543 0.6837778 1.0000002 0.5597499 0.4158214 0.4983572
## PFE 0.6001580 0.5905061 0.5597499 0.9999999 0.3591007 0.4303780
## EA  0.4458394 0.4386693 0.4158214 0.3591007 0.9999999 0.7200000
## CP  0.5343334 0.5257401 0.4983572 0.4303780 0.7200000 1.0000001
```

```
# non-restricted Cor matrix
ability_sem$S # This is the original correlation matrix: ability.
```

```
##      SCA  PPE  PTE  PFE   EA   CP
## SCA 1.00 0.73 0.70 0.58 0.46 0.56
## PPE 0.73 1.00 0.68 0.61 0.43 0.52
## PTE 0.70 0.68 1.00 0.57 0.40 0.48
## PFE 0.58 0.61 0.57 1.00 0.37 0.41
## EA  0.46 0.43 0.40 0.37 1.00 0.72
## CP  0.56 0.52 0.48 0.41 0.72 1.00
```

```
# the root mean square error
sqrt(mean((ability_sem$C-ability_sem$S)^2))
```

```
## [1] 0.01297385
```

Measuring the discrepancy between the estimate Cov matrix and actual Cov matrix, based on different criteria:

- Chi-square test for the model (p-value > 0.05 implies that the actual cov matrix and an estimate cov matrix are almost equal), see page 204 EH textbook.

```
# Degree of Freedom  = q*(q+1)/2  - number of parameters in the model
df = nrow(cov.ability)*(nrow(cov.ability)+1)/2 - length(ability_sem$coeff)
df
```

```
## [1] 8
```

```
summary(ability_sem)
```

```
##
##  Model Chisquare =  9.255732   Df =  8 Pr(>Chisq) = 0.3211842
##  AIC =  35.25573
##  BIC =  -41.31041
##
##  Normalized Residuals
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.4409685 -0.1870306 -0.0000018 -0.0130992  0.2107128  0.5333068
##
##  R-square for Endogenous Variables
##    SCA    PPE    PTE    PFE     EA     CP
## 0.7451 0.7213 0.6482 0.4834 0.6008 0.8629
##
##  Parameter Estimates
```

```
##          Estimate  Std Error  z value   Pr(>|z|)
## lambda1 0.8632049 0.03514508 24.561188 3.284552e-133
## lambda2 0.8493226 0.03545022 23.958178 7.593661e-127
## lambda3 0.8050861 0.03640470 22.114892 2.272503e-108
## lambda4 0.6952671 0.03863370 17.996387  2.079489e-72
## lambda5 0.7750850 0.04035675 19.205834  3.307658e-82
## lambda6 0.9289304 0.03940959 23.571177 7.615270e-123
## rho     0.6663697 0.03095414 21.527645 8.578257e-103
## theta1  0.2548772 0.02336722 10.907470  1.061704e-27
## theta2  0.2786512 0.02412754 11.549097  7.460043e-31
## theta3  0.3518366 0.02691875 13.070321  4.865973e-39
## theta4  0.5166036 0.03472534 14.876847  4.659431e-50
## theta5  0.3992432 0.03819583 10.452535  1.426604e-25
## theta6  0.1370884 0.04350459  3.151126  1.626425e-03
##
## lambda1 SCA <--- Ability
## lambda2 PPE <--- Ability
## lambda3 PTE <--- Ability
## lambda4 PFE <--- Ability
## lambda5 EA <--- Aspiration
## lambda6 CP <--- Aspiration
## rho     Aspiration <--> Ability
## theta1  SCA <--> SCA
## theta2  PPE <--> PPE
## theta3  PTE <--> PTE
## theta4  PFE <--> PFE
## theta5  EA <--> EA
## theta6  CP <--> CP
##
##  Iterations =  29

# P-value > 0.05 implies that the data support the CFA model.
```

- The standard root means square difference (SRMR) (<0.05 is acceptable.), page 205 EH Textbook.

```
dif = ability_sem$C - ability_sem$S
# it measures the root mean square error of the lower or upper triangle of th
e discrepancy matrix.
sqrt(mean(dif[lower.tri(dif, diag = T)]^2))

## [1] 0.01201145
```

- Goodness of fit index (GFI and AGFI) (>0.95 is good), page 205 EH Textbook

```
options(fit.indices = c("GFI", "AGFI", "SRMR")) # Some fit indices
summary(ability_sem)

##
##  Model Chisquare =  9.255732   Df =  8 Pr(>Chisq) = 0.3211842
##  Goodness-of-fit index =  0.9944253
##  Adjusted goodness-of-fit index =  0.9853663
```

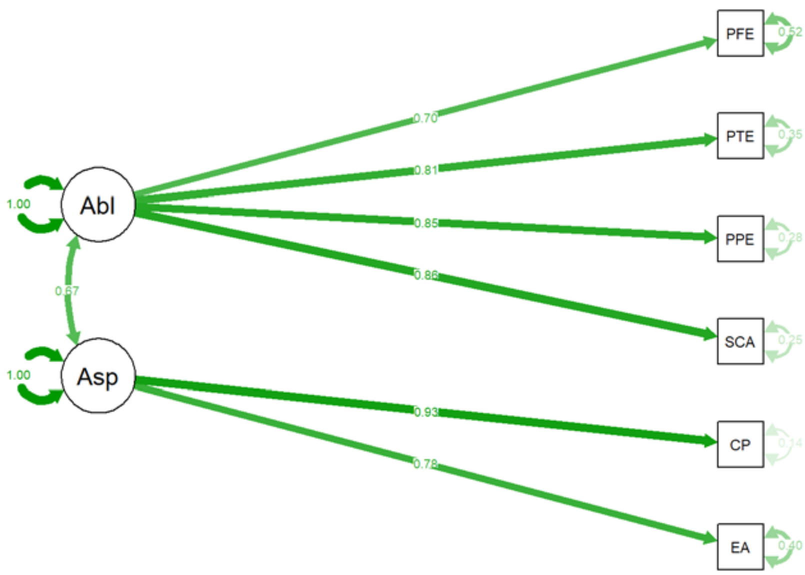
```
##  SRMR =  0.01201145
##
##  Normalized Residuals
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.4409685 -0.1870306 -0.0000018 -0.0130992  0.2107128  0.5333068
##
##  R-square for Endogenous Variables
##    SCA    PPE    PTE    PFE     EA     CP
## 0.7451 0.7213 0.6482 0.4834 0.6008 0.8629
##
##  Parameter Estimates
##         Estimate  Std Error  z value   Pr(>|z|)
## lambda1 0.8632049 0.03514508 24.561188 3.284552e-133
## lambda2 0.8493226 0.03545022 23.958178 7.593661e-127
## lambda3 0.8050861 0.03640470 22.114892 2.272503e-108
## lambda4 0.6952671 0.03863370 17.996387  2.079489e-72
## lambda5 0.7750850 0.04035675 19.205834  3.307658e-82
## lambda6 0.9289304 0.03940959 23.571177 7.615270e-123
## rho     0.6663697 0.03095414 21.527645 8.578257e-103
## theta1  0.2548772 0.02336722 10.907470  1.061704e-27
## theta2  0.2786512 0.02412754 11.549097  7.460043e-31
## theta3  0.3518366 0.02691875 13.070321  4.865973e-39
## theta4  0.5166036 0.03472534 14.876847  4.659431e-50
## theta5  0.3992432 0.03819583 10.452535  1.426604e-25
## theta6  0.1370884 0.04350459  3.151126  1.626425e-03
##
## lambda1 SCA <--- Ability
## lambda2 PPE <--- Ability
## lambda3 PTE <--- Ability
## lambda4 PFE <--- Ability
## lambda5 EA <--- Aspiration
## lambda6 CP <--- Aspiration
## rho     Aspiration <--> Ability
## theta1  SCA <--> SCA
## theta2  PPE <--> PPE
## theta3  PTE <--> PTE
## theta4  PFE <--> PFE
## theta5  EA <--> EA
## theta6  CP <--> CP
##
##  Iterations =  29

# SRMR < 0.05 implies that the data support the CFA model.
```

Lets create **path diagram**

```
library(semPlot)

## Warning: package 'semPlot' was built under R version 3.6.1
```

```
## Registered S3 methods overwritten by 'huge':
##   method    from
##   plot.sim  BDgraph
##   print.sim BDgraph
```

```
semPaths(ability_sem, rotation = 2, 'est')
```



Example: Crime Data. Perform an exploratory factor analysis (EFA) for finding two latent variables. Then we use those variables for doing CFA.

```
crime <- read.csv("https://rb.gy/wu8kvo", header=TRUE, row.names=1)
efa <- factanal(crime, 2)
print(efa$loadings, cut = 0.5)
```

```
##
## Loadings:
##          Factor1 Factor2
## MURDER            0.896
## RAPE      0.548   0.671
## ROBBERY           0.527
## ASSAULT           0.745
## BURGLARY  0.833
## LARCENY   0.881
## AUTO      0.574
##
##                Factor1 Factor2
## SS loadings      2.472   2.294
```

```
## Proportion Var   0.353   0.328
## Cumulative Var   0.353   0.681

# we go with 2 factors: personal crime and property crime
```

Apply CFA based on two factors that are derived from crime EFA

```
# for CFA, we first need a model
library(sem)

# Another way to read the text:
crime_model <- specifyModel(text = "
Personal     -> MURDER, lambda1, NA
Personal     -> RAPE, lambda2, NA
Personal     -> ROBBERY, lambda3, NA
Personal     -> ASSAULT, lambda4, NA
Property     -> BURGLARY, lambda5, NA
Property     -> LARCENY, lambda6, NA
Property     -> AUTO, lambda7, NA
Personal    <-> Property, rho, NA
MURDER      <-> MURDER, theta1, NA
RAPE        <-> RAPE, theta2, NA
ROBBERY     <-> ROBBERY, theta3, NA
ASSAULT     <-> ASSAULT, theta4, NA
BURGLARY    <-> BURGLARY, theta5, NA
LARCENY     <-> LARCENY, theta6, NA
AUTO        <-> AUTO, theta7, NA
Personal <-> Personal, NA, 1
Property <-> Property, NA, 1")

crime_sem <- sem(crime_model, cor(crime), nrow(crime))

#summary(crime_sem)
```

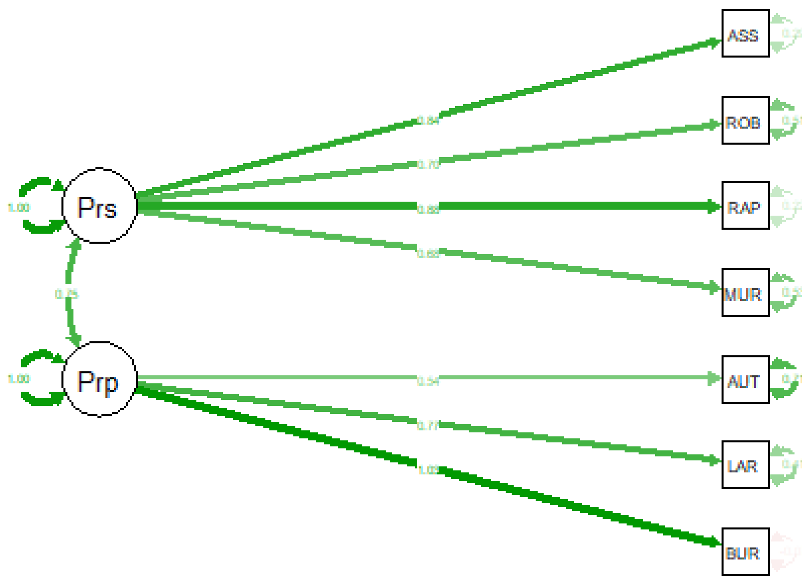
Test the hypothesis that the restricted cov matrix is equal to the non-restricted cov matrix.

```
options(fit.indices = c("GFI", "AGFI", "SRMR")) # Some fit indices
summary(crime_sem)

##
##  Model Chisquare =  39.26441   Df =  13 Pr(>Chisq) = 0.000181408
##  Goodness-of-fit index =  0.8335498
##  Adjusted goodness-of-fit index =  0.641492
##  SRMR =  0.1012327
##
##  Normalized Residuals
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.8941398 -0.1746991 -0.0000004 -0.0450387  0.1774130  2.0889445
##
##  R-square for Endogenous Variables
##   MURDER     RAPE  ROBBERY  ASSAULT BURGLARY  LARCENY     AUTO
```

```
##   0.4669   0.7760   0.4865   0.7119   1.0662   0.5882   0.2886
##
##  Parameter Estimates
##          Estimate    Std Error  z value   Pr(>|z|)
## lambda1  0.68329419 0.13054318  5.234239 1.656660e-07
## lambda2  0.88088432 0.11671842  7.547089 4.450935e-14
## lambda3  0.69748807 0.12965902  5.379403 7.473342e-08
## lambda4  0.84372263 0.11950271  7.060280 1.661673e-12
## lambda5  1.03254996 0.10630637  9.712964 2.655025e-22
## lambda6  0.76692741 0.12363043  6.203387 5.526070e-10
## lambda7  0.53719398 0.13165738  4.080242 4.498883e-05
## rho      0.74922176 0.07725770  9.697696 3.083821e-22
## theta1   0.53310906 0.11920667  4.472141 7.744028e-06
## theta2   0.22404288 0.07460082  3.003223 2.671368e-03
## theta3   0.51351079 0.11586213  4.432085 9.332622e-06
## theta4   0.28813226 0.08139578  3.539892 4.002910e-04
## theta5  -0.06615948 0.08795370 -0.752208 4.519260e-01
## theta6   0.41182239 0.09480688  4.343803 1.400372e-05
## theta7   0.71142245 0.14320676  4.967799 6.771700e-07
##
## lambda1 MURDER <--- Personal
## lambda2 RAPE <--- Personal
## lambda3 ROBBERY <--- Personal
## lambda4 ASSAULT <--- Personal
## lambda5 BURGLARY <--- Property
## lambda6 LARCENY <--- Property
## lambda7 AUTO <--- Property
## rho     Property <--> Personal
## theta1  MURDER <--> MURDER
## theta2  RAPE <--> RAPE
## theta3  ROBBERY <--> ROBBERY
## theta4  ASSAULT <--> ASSAULT
## theta5  BURGLARY <--> BURGLARY
## theta6  LARCENY <--> LARCENY
## theta7  AUTO <--> AUTO
##
##  Iterations =  21

# null hypothesis: the restricted cov matrix (of CFA) is equal to the non-res
tricted cov matrix (of original data)

# p-value = 0.00018 < 0.05 --> conclusion: reject the null hypothesis, so the
re is not enough evidence to say that the restricted cov matrix is equal to t
he non-restricted cov matrix.

# GFI and AGFI are also low. And SRMR is higher than 0.05.

# Data does not support the designed CFA model. MODEL IS NOT CONFIRMED!
```

Report the path diagram that shows coefficients.

```r
library(semPlot)
semPaths(crime_sem, rotation = 2, 'std', 'est')
```



Report SRMR, GFI and AGFI. What do you conclude?

```r
options(fit.indices = c("GFI", "AGFI", "SRMR")) # Some fit indices
criteria = summary(crime_sem)
criteria$SRMR

## [1] 0.1012327

criteria$GFI

## [1] 0.8335498

criteria$AGFI

## [1] 0.641492

criteria$SRMR < 0.05

## [1] FALSE

criteria$GFI > 0.95

## [1] FALSE

criteria$AGFI > 0.95
```

```
## [1] FALSE

# Data does not support the designed CFA model. MODEL IS NOT CONFIRMED!
```

Find the 95% confidence interval for the disattenuated correlation between personal and property crimes.

```
parameters = summary(crime_sem)
parameters$coeff

##            Estimate  Std Error   z value     Pr(>|z|)
## lambda1  0.68329419 0.13054318  5.234239 1.656660e-07
## lambda2  0.88088432 0.11671842  7.547089 4.450935e-14
## lambda3  0.69748807 0.12965902  5.379403 7.473342e-08
## lambda4  0.84372263 0.11950271  7.060280 1.661673e-12
## lambda5  1.03254996 0.10630637  9.712964 2.655025e-22
## lambda6  0.76692741 0.12363043  6.203387 5.526070e-10
## lambda7  0.53719398 0.13165738  4.080242 4.498883e-05
## rho      0.74922176 0.07725770  9.697696 3.083821e-22
## theta1   0.53310906 0.11920667  4.472141 7.744028e-06
## theta2   0.22404288 0.07460082  3.003223 2.671368e-03
## theta3   0.51351079 0.11586213  4.432085 9.332622e-06
## theta4   0.28813226 0.08139578  3.539892 4.002910e-04
## theta5  -0.06615948 0.08795370 -0.752208 4.519260e-01
## theta6   0.41182239 0.09480688  4.343803 1.400372e-05
## theta7   0.71142245 0.14320676  4.967799 6.771700e-07
##
## lambda1   MURDER <--- Personal
## lambda2     RAPE <--- Personal
## lambda3  ROBBERY <--- Personal
## lambda4  ASSAULT <--- Personal
## lambda5 BURGLARY <--- Property
## lambda6  LARCENY <--- Property
## lambda7     AUTO <--- Property
## rho     Property <--> Personal
## theta1      MURDER <--> MURDER
## theta2          RAPE <--> RAPE
## theta3    ROBBERY <--> ROBBERY
## theta4    ASSAULT <--> ASSAULT
## theta5  BURGLARY <--> BURGLARY
## theta6    LARCENY <--> LARCENY
## theta7          AUTO <--> AUTO

# lets focus on Rho, the correlation between the factors
parameters$coeff[8,]$Estimate

## [1] 0.7492218

conf.L = parameters$coeff[8,]$Estimate - 1.96 * parameters$coeff[8,]$`Std Err
or`
conf.U = parameters$coeff[8,]$Estimate + 1.96 * parameters$coeff[8,]$`Std Err
```

```
or`
conf.L

## [1] 0.5977967

conf.U

## [1] 0.9006469
```