Multivariate Analysis for the Behavioral Sciences, Second Edition (Chapman and Hall/CRC, 2019)
Examples of Chapter 16: Confirmatory Factor Analysis and Structural Equation Models

Kimmo Vehkalahti and Brian S. Everitt

14 December 2018

Contents

Examples **2**

Table 16.1: Observed Correlations for the Ability and Aspiration Example **2**
Figure 16.1 3
Table 16.2 5

Figure 16.2: Path diagram for the drug usage model **6**
Table 16.3 9

Figure 16.3: Corrgram for the Systems Intelligence data **11**
Figure 16.4 13

Figure 16.5: Path diagram for the hypothesized structural equation model for job satisfaction data **18**

Examples

Table 16.1: Observed Correlations for the Ability and Aspiration Example

```r
CORR <- matrix(c(1.00,0.00,0.00,0.00,0.00,0.00,
                 0.73,1.00,0.00,0.00,0.00,0.00,
                 0.70,0.68,1.00,0.00,0.00,0.00,
                 0.58,0.61,0.57,1.00,0.00,0.00,
                 0.46,0.43,0.40,0.37,1.00,0.00,
                 0.56,0.52,0.48,0.41,0.72,1.00),
               ncol = 6, byrow = TRUE)
varNames <- c("SCA", "PPE", "PTE", "PFE", "EA", "CP")
dimnames(CORR) <- list(varNames, varNames)
CORR
```

```
##      SCA  PPE  PTE  PFE   EA CP
## SCA 1.00 0.00 0.00 0.00 0.00  0
## PPE 0.73 1.00 0.00 0.00 0.00  0
## PTE 0.70 0.68 1.00 0.00 0.00  0
## PFE 0.58 0.61 0.57 1.00 0.00  0
## EA  0.46 0.43 0.40 0.37 1.00  0
## CP  0.56 0.52 0.48 0.41 0.72  1
```

Figure 16.1

```r
#install.packages("sem")
#install.packages("semPlot")
library(sem)
library(semPlot)

AAmodel <-specifyModel(text = "
   Ability     -> SCA,        lambda[1], NA
   Ability     -> PPE,        lambda[2], NA
   Ability     -> PTE,        lambda[3], NA
   Ability     -> PFE,        lambda[4], NA
   Aspiration  -> EA,         lambda[5], NA
   Aspiration  -> CP,         lambda[6], NA
   Ability    <-> Aspiration, rho,       NA
   SCA        <-> SCA,        u[1],      NA
   PPE        <-> PPE,        u[2],      NA
   PTE        <-> PTE,        u[3],      NA
   PFE        <-> PFE,        u[4],      NA
   EA         <-> EA,         u[5],      NA
   CP         <-> CP,         u[6],      NA
   Ability    <-> Ability,    NA,        1
   Aspiration <-> Aspiration, NA,        1
  ")
```

```
## NOTE: it is generally simpler to use specifyEquations() or cfa()
##       see ?specifyEquations
```

```r
options(fit.indices = c("CFI", "NNFI", "RMSEA", "SRMR")) # (NNFI = TLI)

AAsem <- sem(AAmodel, CORR, N = 556)

semPaths(AAsem, # filetype = "pdf", filename = "AAmodel-KV",
         what = "path", whatLabels = "name", style = "lisrel", layout = "tree2",
         intercepts = FALSE, residuals = TRUE, thresholds = FALSE, reorder = FALSE,
         rotation = 1, nCharNodes = 0, nCharEdges = 0,
         sizeMan = 8, sizeLat = 20,  sizeMan2 = 6, sizeLat2 = 10,
         manifests = varNames, latents = c("Ability", "Aspiration"),
         residScale = 12, as.expression = c("edges", "nodes"), centerLevels = FALSE,
         edge.label.cex = 1.2, esize = 2.5, label.scale = TRUE, curvePivot = TRUE,
         curvePivotShape = 0.67, edge.label.position = 0.67, width = 6, height = 2.5)
```

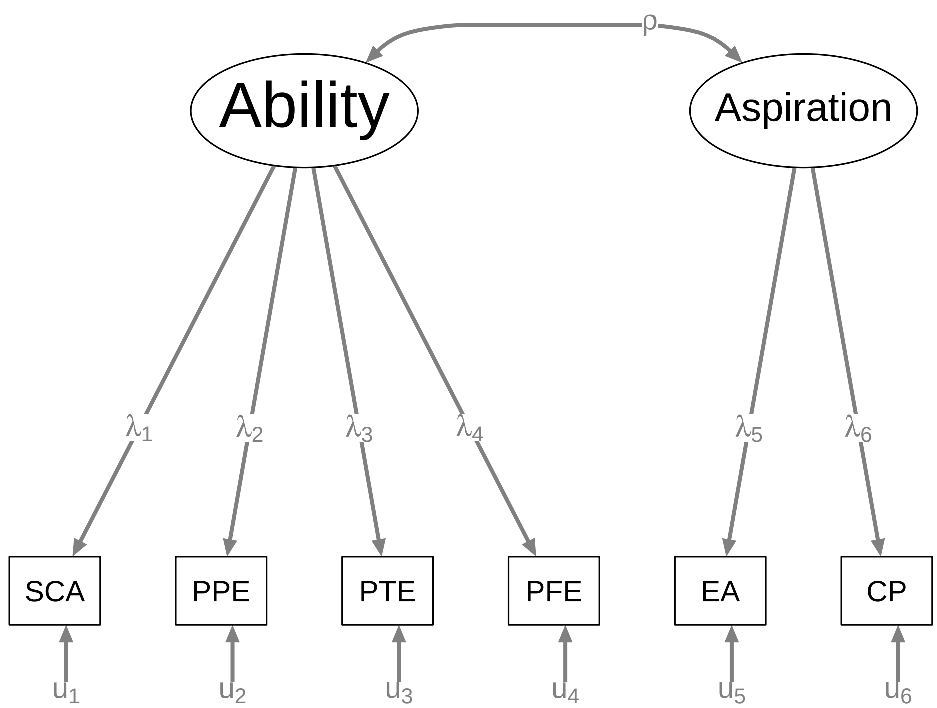

ρ
Ability
Aspiration
λ1
λ2
λ3
λ4
λ5
λ6
SCA
PPE
PTE
PFE
EA
CP
u1
u2
u3
u4
u5
u6

Table 16.2

```r
summary(AAsem)
```

```
##
##  Model Chisquare =  9.255732   Df =  8 Pr(>Chisq) = 0.3211842
##  RMSEA index =  0.01681733   90% CI: (NA, 0.05432054)
##  Tucker-Lewis NNFI =  0.9987042
##  Bentler CFI =  0.9993089
##  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|)
## lambda[1] 0.8632049 0.03514508 24.561188 3.284552e-133
## lambda[2] 0.8493226 0.03545022 23.958178 7.593661e-127
## lambda[3] 0.8050861 0.03640470 22.114892 2.272503e-108
## lambda[4] 0.6952671 0.03863370 17.996387  2.079489e-72
## lambda[5] 0.7750850 0.04035675 19.205834  3.307658e-82
## lambda[6] 0.9289304 0.03940959 23.571177 7.615270e-123
## rho       0.6663697 0.03095414 21.527645 8.578257e-103
## u[1]      0.2548772 0.02336722 10.907470  1.061704e-27
## u[2]      0.2786512 0.02412754 11.549097  7.460043e-31
## u[3]      0.3518366 0.02691875 13.070321  4.865973e-39
## u[4]      0.5166036 0.03472534 14.876847  4.659431e-50
## u[5]      0.3992432 0.03819583 10.452535  1.426604e-25
## u[6]      0.1370884 0.04350459  3.151126  1.626425e-03
##
## lambda[1] SCA <--- Ability
## lambda[2] PPE <--- Ability
## lambda[3] PTE <--- Ability
## lambda[4] PFE <--- Ability
## lambda[5] EA <--- Aspiration
## lambda[6] CP <--- Aspiration
## rho       Aspiration <--> Ability
## u[1]      SCA <--> SCA
## u[2]      PPE <--> PPE
## u[3]      PTE <--> PTE
## u[4]      PFE <--> PFE
## u[5]      EA <--> EA
## u[6]      CP <--> CP
##
##  Iterations =  29
```

Figure 16.2: Path diagram for the drug usage model

Revisiting the drug usage example introduced in **Chapter 13**:

```
DRUGcorr <- matrix(c(
  1.0,rep(0,12),
  0.447,1.0,rep(0,11),
  0.422,0.619,1.0,rep(0,10),
  0.436,0.604,0.583,1.0,rep(0,9),
  0.114,0.068,0.053,0.115,1.0,rep(0,8),
  0.203,0.146,0.139,0.256,0.349,1.0,rep(0,7),
  0.091,0.103,0.110,0.122,0.209,0.221,1.0,rep(0,6),
  0.082,0.063,0.066,0.097,0.321,0.355,0.201,1.0,rep(0,5),
  0.513,0.445,0.365,0.482,0.186,0.316,0.150,0.154,1.0,rep(0,4),
  0.304,0.318,0.240,0.368,0.303,0.377,0.163,0.219,0.534,1.0,rep(0,3),
  0.245,0.203,0.183,0.255,0.272,0.323,0.310,0.288,0.301,0.302,1.0,rep(0,2),
  0.101,0.088,0.074,0.139,0.279,0.367,0.232,0.320,0.204,0.368,0.340,1.0,0.0,
  0.245,0.199,0.184,0.293,0.278,0.545,0.232,0.314,0.394,0.467,0.392,0.511,1.0), ncol=13, byrow=T)
DRUGnames <- c("Cigs", "Beer", "Wine", "Liqr", "Cocn", "Tran", "Drug", "Hern",
               "Marj", "Hash", "Inhl", "Hall", "Amph")
dimnames(DRUGcorr) <- list(DRUGnames, DRUGnames)
DRUGcorr <- DRUGcorr + t(DRUGcorr)
diag(DRUGcorr) <- diag(DRUGcorr) - 1
DRUGcorr
```

```
##       Cigs  Beer  Wine  Liqr  Cocn  Tran  Drug  Hern  Marj  Hash  Inhl
## Cigs 1.000 0.447 0.422 0.436 0.114 0.203 0.091 0.082 0.513 0.304 0.245
## Beer 0.447 1.000 0.619 0.604 0.068 0.146 0.103 0.063 0.445 0.318 0.203
## Wine 0.422 0.619 1.000 0.583 0.053 0.139 0.110 0.066 0.365 0.240 0.183
## Liqr 0.436 0.604 0.583 1.000 0.115 0.256 0.122 0.097 0.482 0.368 0.255
## Cocn 0.114 0.068 0.053 0.115 1.000 0.349 0.209 0.321 0.186 0.303 0.272
## Tran 0.203 0.146 0.139 0.256 0.349 1.000 0.221 0.355 0.316 0.377 0.323
## Drug 0.091 0.103 0.110 0.122 0.209 0.221 1.000 0.201 0.150 0.163 0.310
## Hern 0.082 0.063 0.066 0.097 0.321 0.355 0.201 1.000 0.154 0.219 0.288
## Marj 0.513 0.445 0.365 0.482 0.186 0.316 0.150 0.154 1.000 0.534 0.301
## Hash 0.304 0.318 0.240 0.368 0.303 0.377 0.163 0.219 0.534 1.000 0.302
## Inhl 0.245 0.203 0.183 0.255 0.272 0.323 0.310 0.288 0.301 0.302 1.000
## Hall 0.101 0.088 0.074 0.139 0.279 0.367 0.232 0.320 0.204 0.368 0.340
## Amph 0.245 0.199 0.184 0.293 0.278 0.545 0.232 0.314 0.394 0.467 0.392
##       Hall  Amph
## Cigs 0.101 0.245
## Beer 0.088 0.199
## Wine 0.074 0.184
## Liqr 0.139 0.293
## Cocn 0.279 0.278
## Tran 0.367 0.545
## Drug 0.232 0.232
## Hern 0.320 0.314
## Marj 0.204 0.394
## Hash 0.368 0.467
## Inhl 0.340 0.392
## Hall 1.000 0.511
## Amph 0.511 1.000
```

```r
DRUGmodel <-specifyModel(text = "

Alcohol   -> Cigs,  lambda[1],NA
Cannabis  -> Cigs,  lambda[2],NA
Alcohol   -> Beer,  lambda[3],NA
Alcohol   -> Wine,  lambda[4],NA
Cannabis  -> Wine,  lambda[5],NA
Alcohol   -> Liqr,  lambda[6],NA
HardDrug  -> Liqr,  lambda[7],NA
HardDrug  -> Cocn,  lambda[8],NA
HardDrug  -> Tran,  lambda[9],NA
HardDrug  -> Drug,  lambda[10],NA
HardDrug  -> Hern,  lambda[11],NA
Cannabis  -> Marj,  lambda[12],NA
Cannabis  -> Hash,  lambda[13],NA
HardDrug  -> Hash,  lambda[14],NA
HardDrug  -> Inhl,  lambda[15],NA
HardDrug  -> Hall,  lambda[16],NA
HardDrug  -> Amph,  lambda[17],NA

Cigs     <-> Cigs,  u[1],NA
Beer     <-> Beer,  u[2],NA
Wine     <-> Wine,  u[3],NA
Liqr     <-> Liqr,  u[4],NA
Cocn     <-> Cocn,  u[5],NA
Tran     <-> Tran,  u[6],NA
Drug     <-> Drug,  u[7],NA
Hern     <-> Hern,  u[8],NA
Marj     <-> Marj,  u[9],NA
Hash     <-> Hash,  u[10],NA
Inhl     <-> Inhl,  u[11],NA
Hall     <-> Hall,  u[12],NA
Amph     <-> Amph,  u[13],NA

Alcohol  <-> Cannabis, rho[1],NA
Alcohol  <-> HardDrug, rho[2],NA
Cannabis <-> HardDrug, rho[3],NA

Alcohol  <-> Alcohol, NA,1
Cannabis <-> Cannabis,NA,1
HardDrug <-> HardDrug,NA,1

")
```

```
## NOTE: it is generally simpler to use specifyEquations() or cfa()
##       see ?specifyEquations
```

```r
options(fit.indices = c("CFI", "NNFI", "RMSEA", "SRMR")) # (NNFI = TLI)

DRUGsem <- sem(DRUGmodel, DRUGcorr, N = 1634)
```

```r
semPaths(DRUGsem, # filetype = "pdf", filename = "DRUGmodel-KV",
         what = "path", whatLabels = "hide",
         style = "lisrel", layout = "tree2",
         residuals = TRUE, rotation = 2, nCharNodes = 0,
         sizeMan = 8, sizeLat = 10, sizeMan2 = 4, sizeLat2 = 10,
         esize = 2, label.cex = 0.75, label.scale = FALSE, width = 6, height = 5)
```

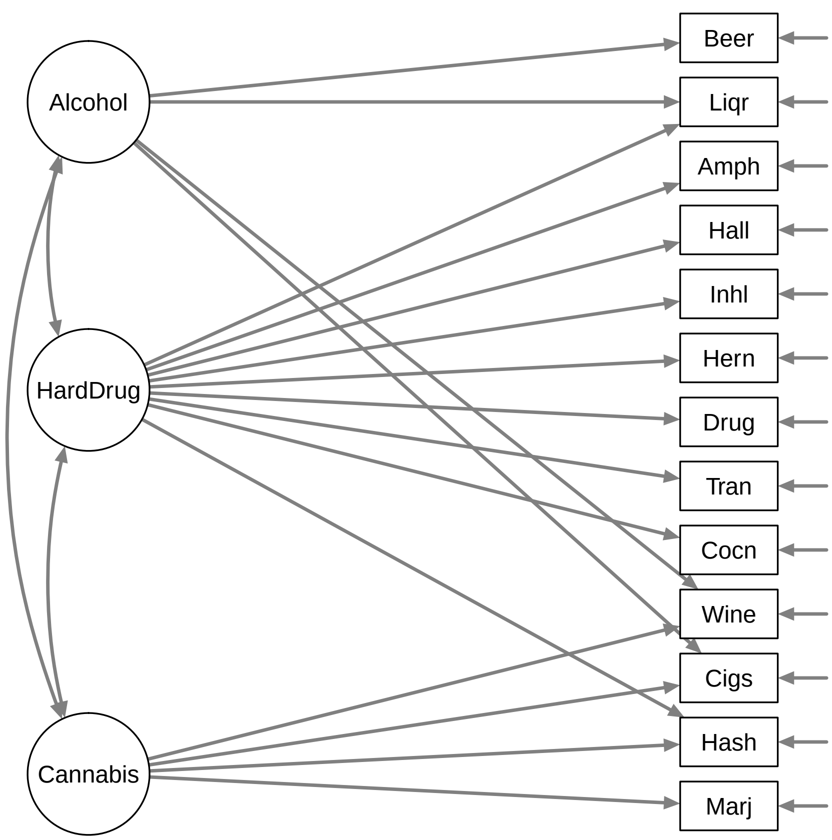


Table 16.3

```r
summary(DRUGsem)
```

```
##
##  Model Chisquare =  323.3541   Df =  58 Pr(>Chisq) = 1.676805e-38
##  RMSEA index =  0.05293048   90% CI: (0.04738096, 0.05863211)
##  Tucker-Lewis NNFI =  0.9453964
##  Bentler CFI =  0.9593974
##  SRMR =  0.03899979
##
##  Normalized Residuals
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.031458 -0.884693 -0.000004 -0.021815  0.998363  4.576743
##
##  R-square for Endogenous Variables
##   Cigs   Beer   Wine   Liqr   Cocn   Tran   Drug   Hern   Marj   Hash
## 0.3888 0.6264 0.6215 0.5922 0.2159 0.4562 0.1285 0.2265 0.8320 0.4531
##   Inhl   Hall   Amph
## 0.2948 0.3823 0.5828
##
##  Parameter Estimates
##            Estimate   Std Error  z value   Pr(>|z|)
## lambda[1]   0.3580610 0.03449894 10.378898  3.093248e-25
## lambda[2]   0.3317505 0.03527140  9.405651  5.170904e-21
## lambda[3]   0.7914844 0.02260160 35.018962 1.157714e-268
## lambda[4]   0.8759572 0.03759170 23.301881 4.242807e-120
## lambda[5]  -0.1522411 0.03659116 -4.160598  3.174149e-05
## lambda[6]   0.7222173 0.02354662 30.671802 1.353388e-206
## lambda[7]   0.1226893 0.02265514  5.415519  6.111133e-08
## lambda[8]   0.4646768 0.02572690 18.061903  6.359149e-73
## lambda[9]   0.6754614 0.02397020 28.179212 1.051469e-174
## lambda[10]  0.3584240 0.02637142 13.591383  4.504847e-42
## lambda[11]  0.4759180 0.02564863 18.555299  7.389575e-77
## lambda[12]  0.9121178 0.03045556 29.949139 4.514959e-197
## lambda[13]  0.3955832 0.02960491 13.362081  1.007085e-40
## lambda[14]  0.3817698 0.02923025 13.060776  5.516293e-39
## lambda[15]  0.5429839 0.02514237 21.596370 1.942958e-103
## lambda[16]  0.6182726 0.02449908 25.236559 1.590899e-140
## lambda[17]  0.7633843 0.02313045 33.003443 7.249129e-239
## u[1]        0.6112369 0.02367015 25.823114 4.878528e-147
## u[2]        0.3735526 0.01992742 18.745656  2.100659e-78
## u[3]        0.3784784 0.02357593 16.053595  5.394692e-58
## u[4]        0.4078314 0.01911408 21.336701 5.182466e-101
## u[5]        0.7840758 0.02920268 26.849444 8.561618e-159
## u[6]        0.5437528 0.02341101 23.226369 2.465982e-119
## u[7]        0.8715324 0.03151476 27.654733 2.447978e-168
## u[8]        0.7735019 0.02892760 26.739240 1.647200e-157
## u[9]        0.1680413 0.04394465  3.823931  1.313404e-04
## u[10]       0.5469320 0.02223445 24.598408 1.313719e-133
## u[11]       0.7051680 0.02718073 25.943676 2.143063e-148
## u[12]       0.6177393 0.02505445 24.655873 3.183322e-134
## u[13]       0.4172447 0.02118572 19.694615  2.398122e-86
## rho[1]      0.6334751 0.02712849 23.350911 1.349094e-120
```

```
## rho[2]      0.3132784 0.02933063 10.680931  1.250110e-26
## rho[3]      0.4993778 0.02711226 18.418889  9.268602e-76
##
## lambda[1]  Cigs <--- Alcohol
## lambda[2]  Cigs <--- Cannabis
## lambda[3]  Beer <--- Alcohol
## lambda[4]  Wine <--- Alcohol
## lambda[5]  Wine <--- Cannabis
## lambda[6]  Liqr <--- Alcohol
## lambda[7]  Liqr <--- HardDrug
## lambda[8]  Cocn <--- HardDrug
## lambda[9]  Tran <--- HardDrug
## lambda[10] Drug <--- HardDrug
## lambda[11] Hern <--- HardDrug
## lambda[12] Marj <--- Cannabis
## lambda[13] Hash <--- Cannabis
## lambda[14] Hash <--- HardDrug
## lambda[15] Inhl <--- HardDrug
## lambda[16] Hall <--- HardDrug
## lambda[17] Amph <--- HardDrug
## u[1]       Cigs <--> Cigs
## u[2]       Beer <--> Beer
## u[3]       Wine <--> Wine
## u[4]       Liqr <--> Liqr
## u[5]       Cocn <--> Cocn
## u[6]       Tran <--> Tran
## u[7]       Drug <--> Drug
## u[8]       Hern <--> Hern
## u[9]       Marj <--> Marj
## u[10]      Hash <--> Hash
## u[11]      Inhl <--> Inhl
## u[12]      Hall <--> Hall
## u[13]      Amph <--> Amph
## rho[1]     Cannabis <--> Alcohol
## rho[2]     HardDrug <--> Alcohol
## rho[3]     HardDrug <--> Cannabis
##
##  Iterations =  34
```

Figure 16.3: Corrgram for the Systems Intelligence data

Source of data: Törmänen, J., Hämäläinen, R. P. and Saarinen, E. (2016). Systems Intelligence inventory. *The Learning Organization*, 23, 218–231. (The data and parts of the R code by *Juha Törmänen* are used below with kind permission from the authors.)

See also:

- Saarinen, E. and Hämäläinen, R. P. (2004). Systems intelligence: Connecting engineering thinking with human sensitivity. In *Systems Intelligence—Discovering a Hidden Competence in Human Action and Organizational Life* (eds. R. P. Hämäläinen and E. Saarinen). Helsinki University of Technology, Research Reports A88. http://sal.aalto.fi/publications/pdffiles/systemsintelligence2004.pdf.
- Hämäläinen, R. P., Jones, R. and Saarinen, E. (2014). *Being Better Better: Living with Systems Intelligence.* Aalto University Publications Crossover 4/2014, http://systemsintelligence.aalto.fi/being_better_better/.

```r
# Set seed to remove effect of random generator
set.seed(42)

# Load data (sidata.tsv contains renamed answer columns with names si1-si32)
si_cfa <- list()
si_cfa$data <- read.table("data/sidata.tsv", header = TRUE, sep = '\t')
si_cfa$dataset <- si_cfa$data$set

# SI model answers for the 32 items
si_cfa$answers <- si_cfa$data[, 10:41]

head(si_cfa$answers)
```

```
##   si1 si2 si3 si17 si13 si14 si15 si4 si25 si5 si26 si6 si27 si28 si7 si18
## 1   4   5   4    5    3    4    4   3    4   4    5   4    4    3   4    4
## 2   4   4   4    4    6    5    5   5    3   4    4   4    3    3   4    4
## 3   4   4   5    5    4    5    5   4    4   4    4   6    4    3   4    4
## 4   4   5   4    4    4    4    3   5    3   5    4   5    4    5   5    3
## 5   5   5   5    5    3    5    4   5    5   5    3   5    3    4   5    4
## 6   3   5   4    6    4    4    4   5    5   5    5   6    4    5   4    5
##   si20 si21 si9 si10 si11 si22 si8 si23 si19 si16 si29 si30 si12 si24 si31
## 1    4    5   3    3    4    3   6    4    5    4    4    1    1    6    4
## 2    5    3   3    3   NA    4   3    4    5    5    4    2    3    3    4
## 3    6    5   3    5    4    5   5    3    4    4    4    3    3    4    3
## 4    4    5   1    5    1    5   5    5    5    4    4    1    0    5    4
## 5    5    3   2    5    2    3   4    4    5    4    4    2    1    5    5
## 6    6    4   3    6    3    4   4    4    5    5    5    3    3    5    5
##   si32
## 1    4
## 2    4
## 3    3
## 4    5
## 5    4
## 6    5
```

```r
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
mcor <- cor(si_cfa$answers, use = "complete.obs")
# Add leading zeros to var names for the "alphabet" option of corrplot:
rownames(mcor) <- c("si01", "si02", "si03", "si17", "si13", "si14", "si15", "si04",
                    "si25", "si05", "si26", "si06", "si27", "si28", "si07", "si18",
                    "si20", "si21", "si09", "si10", "si11", "si22", "si08", "si23",
                    "si19", "si16", "si29", "si30", "si12", "si24", "si31", "si32")
colnames(mcor) <- rownames(mcor)
corrplot(mcor, method = "shade", order = "alphabet",
         tl.cex = 0.7, tl.col = "black", tl.srt = 45)
```

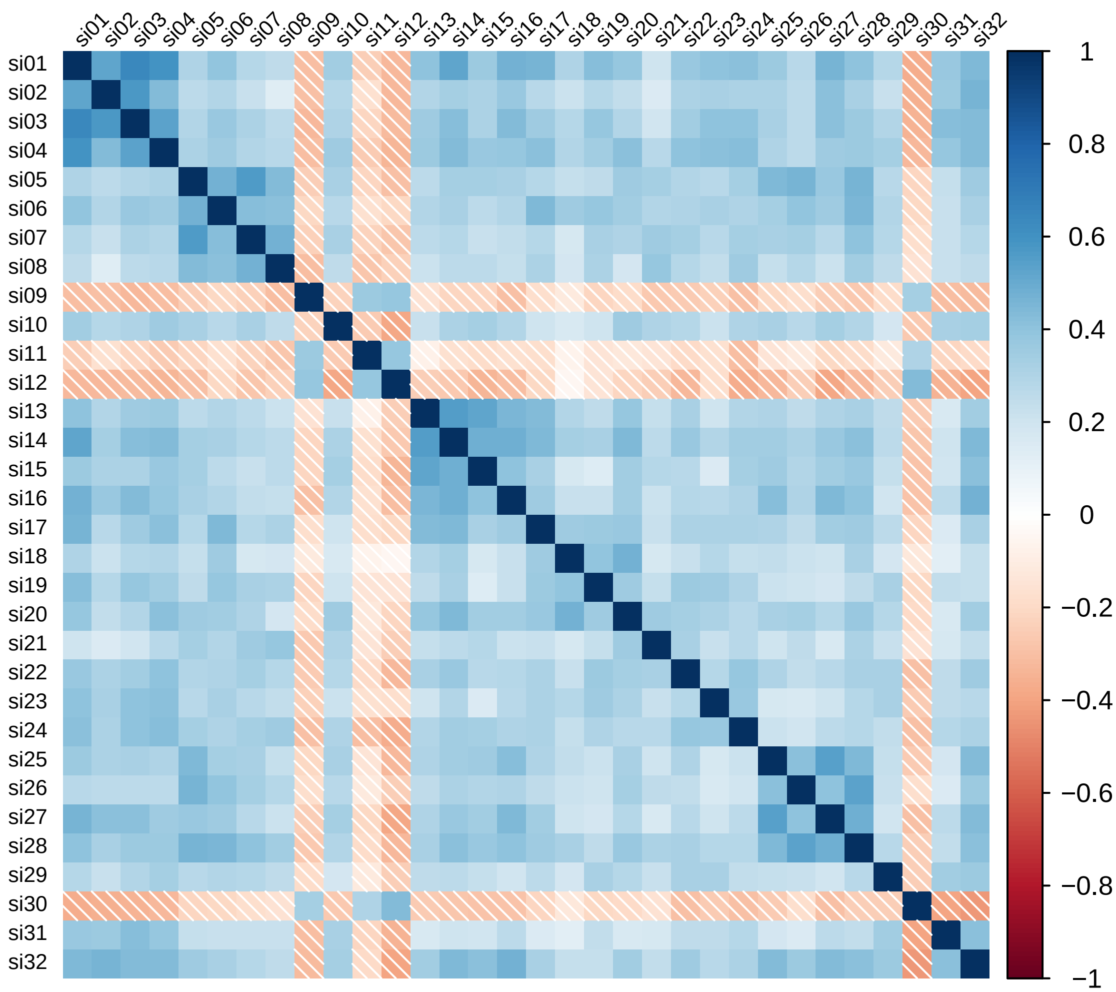


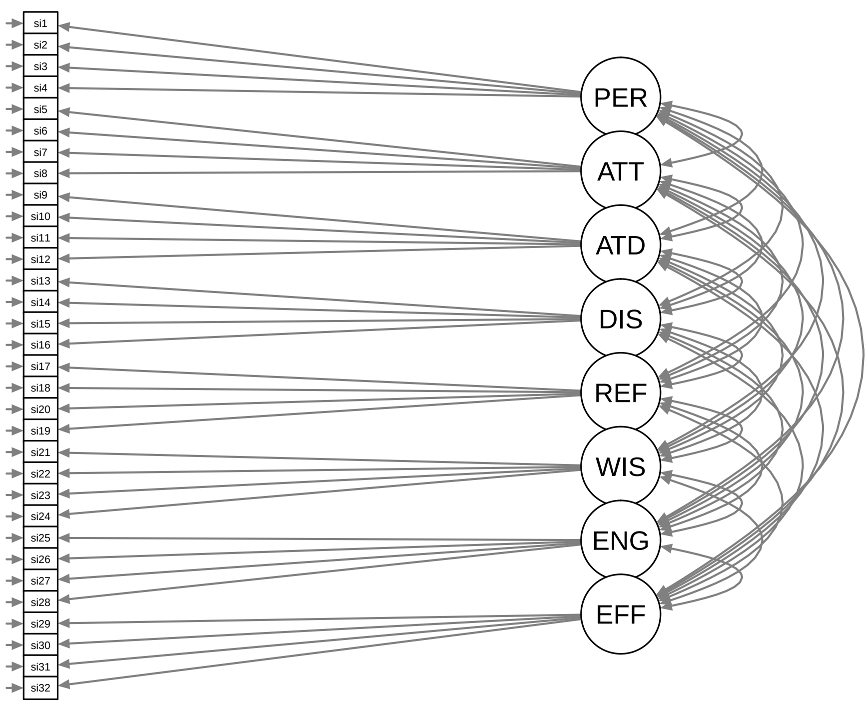
Figure 16.4

```
# DATA: Validation dataset rows with no missing data (N=815):
si_cfa$data.validation <- si_cfa$data[rowSums(is.na(si_cfa$answers)) == 0 &
                                      si_cfa$dataset == 'validation', ]

# MODEL: Eight factors, free loadings:
# Free weight model: Each item is explained by one factor, factors have unit variance
# In total 8*4 loadings + 7*8/2 factor covariances + 8*4 item variances = 92 parameters

model.8free <- specifyEquations(text = "

si1  =  a1(1)*PER
si2  =  a2(1)*PER
si3  =  a3(1)*PER
si4  =  a4(1)*PER

si5  =  a5(1)*ATT
si6  =  a6(1)*ATT
si7  =  a7(1)*ATT
si8  =  a8(1)*ATT

# ATD factor starting values coded negatively here; othersize sem has trouble converging
si9  =  a9(1)*ATD
si10 = a10(-1)*ATD
si11 = a11(1)*ATD
si12 = a12(1)*ATD

si13 = a13(1)*DIS
si14 = a14(1)*DIS
si15 = a15(1)*DIS
si16 = a16(1)*DIS

si17 = a17(1)*REF
si18 = a18(1)*REF
si19 = a19(1)*REF
si20 = a20(1)*REF

si21 = a21(1)*WIS
si22 = a22(1)*WIS
si23 = a23(1)*WIS
si24 = a24(1)*WIS

si25 = a25(1)*ENG
si26 = a26(1)*ENG
si27 = a27(1)*ENG
si28 = a28(1)*ENG

si29 = a29(1)*EFF
si30 = a30(-1)*EFF
si31 = a31(1)*EFF
si32 = a32(1)*EFF
```

```
V(PER) = 1
V(ATT) = 1
V(ATD) = 1
V(DIS) = 1
V(REF) = 1
V(WIS) = 1
V(ENG) = 1
V(EFF) = 1

", covs = "PER,ATT,ATD,DIS,REF,WIS,ENG,EFF")
```

```
## NOTE: adding 32 variances to the model
```

```
options(fit.indices = c("SRMR", "CFI", "RMSEA"))

res.8free <- sem(model = model.8free, data = si_cfa$data.validation, objective = objectiveGLS)
# summary(res.8free) outputs CFI = 0.951, RMSEA = 0.048, SRMR = 0.068

semPaths(res.8free, # filetype = "pdf", filename = "SImodel-KV",
         what = "path", whatLabels = "hide", style = "lisrel",
         residuals = TRUE, curve = 3.2, rotation = 4, nCharNodes = 0,
         sizeLat = 7, sizeMan = 3, sizeMan2 = 2,  esize = 1.3, mar = c(3, 8, 3, 15))
```



```r
summary(res.8free)
```

```
##
##  Model Chisquare =  1256.666   Df =  436 Pr(>Chisq) = 1.484832e-80
##  RMSEA index =  0.04808702   90% CI: (NA, NA)
##  Bentler CFI =  0.9507248
##  SRMR =  0.06815243
##
##  Normalized Residuals
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.9692 -0.5851  0.3119  0.4389  1.4955  5.0806
##
##  R-square for Endogenous Variables
##    si1    si2    si3    si4    si5    si6    si7    si8    si9   si10
## 0.6853 0.4683 0.6063 0.5770 0.5986 0.4679 0.5216 0.4154 0.3048 0.3812
##   si11   si12   si13   si14   si15   si16   si17   si18   si19   si20
## 0.2329 0.4948 0.5226 0.5683 0.4666 0.4705 0.4399 0.3281 0.3864 0.5134
##   si21   si22   si23   si24   si25   si26   si27   si28   si29   si30
## 0.3039 0.3875 0.3102 0.4414 0.4525 0.4340 0.5544 0.5727 0.3177 0.4029
##   si31   si32
## 0.3903 0.5908
##
##  Parameter Estimates
##           Estimate   Std Error  z value    Pr(>|z|)
## a1         0.6685600 0.03242847  20.616450  1.953814e-94 si1 <--- PER
## a2         0.5176584 0.03255797  15.899592  6.377848e-57 si2 <--- PER
## a3         0.5503226 0.02912131  18.897592  1.193777e-79 si3 <--- PER
## a4         0.6639999 0.03216349  20.644522  1.093385e-94 si4 <--- PER
## a5         0.7199893 0.03731261  19.296137  5.787753e-83 si5 <--- ATT
## a6         0.5830731 0.03756700  15.520887  2.505649e-54 si6 <--- ATT
## a7         0.6052763 0.03284806  18.426547  8.045792e-76 si7 <--- ATT
## a8         0.4726956 0.03702206  12.767945  2.476041e-37 si8 <--- ATT
## a9         0.5539789 0.04804567  11.530257  9.286654e-31 si9 <--- ATD
## a10       -0.5988248 0.04408796 -13.582504  5.085730e-42 si10 <--- ATD
## a11        0.5207716 0.05559112   9.367892  7.399543e-21 si11 <--- ATD
## a12        0.7809641 0.05014166  15.575156  1.073899e-54 si12 <--- ATD
## a13        0.7076282 0.04249851  16.650660  2.992851e-62 si13 <--- DIS
## a14        0.7079806 0.03510499  20.167519  1.888795e-90 si14 <--- DIS
## a15        0.6562514 0.04315793  15.205813  3.235861e-52 si15 <--- DIS
## a16        0.6307662 0.04019476  15.692747  1.695563e-55 si16 <--- DIS
## a17        0.5489697 0.03795019  14.465532  2.000525e-47 si17 <--- REF
## a18        0.5322472 0.04663205  11.413762  3.569486e-30 si18 <--- REF
## a19        0.4752247 0.03977588  11.947560  6.685971e-33 si19 <--- REF
## a20        0.7397153 0.04971787  14.878259  4.562140e-50 si20 <--- REF
## a21        0.5196336 0.04772932  10.887095  1.328092e-27 si21 <--- WIS
## a22        0.5626853 0.03632400  15.490729  4.007202e-54 si22 <--- WIS
## a23        0.4487358 0.03665690  12.241511  1.865266e-34 si23 <--- WIS
## a24        0.6262280 0.03834655  16.330752  5.964490e-60 si24 <--- WIS
## a25        0.6894555 0.04596920  14.998207  7.542925e-51 si25 <--- ENG
## a26        0.6236446 0.03980360  15.668046  2.501507e-55 si26 <--- ENG
## a27        0.7692516 0.04325956  17.782232  9.703919e-71 si27 <--- ENG
## a28        0.6896095 0.03461536  19.922065  2.619637e-88 si28 <--- ENG
## a29        0.4938007 0.04166490  11.851720  2.108191e-32 si29 <--- EFF
## a30       -0.6660330 0.04612172 -14.440766  2.866367e-47 si30 <--- EFF
```

```
## a31          0.4733875 0.03523098  13.436683  3.685926e-41 si31 <--- EFF
## a32          0.7197076 0.03831885  18.782075  1.058611e-78 si32 <--- EFF
## C[PER,ATT]   0.6762624 0.03215036  21.034367  3.179857e-98 ATT <--> PER
## C[PER,ATD]  -0.7810532 0.03194689 -24.448492 5.221765e-132 ATD <--> PER
## C[PER,DIS]   0.8358883 0.02263808  36.923990 1.904974e-298 DIS <--> PER
## C[PER,REF]   0.8386651 0.02640626  31.760091 2.303946e-221 REF <--> PER
## C[PER,WIS]   0.8864440 0.02456816  36.081004 4.504133e-285 WIS <--> PER
## C[PER,ENG]   0.7638040 0.02683644  28.461447 3.516906e-178 ENG <--> PER
## C[PER,EFF]   0.8161309 0.02419588  33.730154 2.089629e-249 EFF <--> PER
## C[ATT,ATD]  -0.7987056 0.03383097 -23.608712 3.136551e-123 ATD <--> ATT
## C[ATT,DIS]   0.7021611 0.03129715  22.435302 1.780974e-111 DIS <--> ATT
## C[ATT,REF]   0.7505247 0.03420019  21.945047 9.656894e-107 REF <--> ATT
## C[ATT,WIS]   0.8241651 0.03052199  27.002334 1.387521e-160 WIS <--> ATT
## C[ATT,ENG]   0.8679997 0.02244717  38.668562  0.000000e+00 ENG <--> ATT
## C[ATT,EFF]   0.7144656 0.03309641  21.587405 2.358938e-103 EFF <--> ATT
## C[ATD,DIS]  -0.7679736 0.03658449 -20.991782  7.796705e-98 DIS <--> ATD
## C[ATD,REF]  -0.7656555 0.04034199 -18.979119  2.538162e-80 REF <--> ATD
## C[ATD,WIS]  -0.8709160 0.03342931 -26.052468 1.261453e-149 WIS <--> ATD
## C[ATD,ENG]  -0.8200374 0.03157837 -25.968327 1.129124e-148 ENG <--> ATD
## C[ATD,EFF]  -0.8454915 0.03057594 -27.652180 2.627332e-168 EFF <--> ATD
## C[DIS,REF]   0.8317475 0.02898892  28.691912 4.813321e-181 REF <--> DIS
## C[DIS,WIS]   0.7894968 0.03083435  25.604454 1.360900e-144 WIS <--> DIS
## C[DIS,ENG]   0.7614736 0.02868166  26.549150 2.626927e-155 ENG <--> DIS
## C[DIS,EFF]   0.7731367 0.03071951  25.167615 9.066424e-140 EFF <--> DIS
## C[REF,WIS]   0.8840439 0.02962020  29.845978 9.899816e-196 WIS <--> REF
## C[REF,ENG]   0.7464520 0.03291526  22.677990 7.388787e-114 ENG <--> REF
## C[REF,EFF]   0.7541803 0.03521238  21.418043 9.072026e-102 EFF <--> REF
## C[WIS,ENG]   0.7958373 0.03200873  24.863133 1.864692e-136 ENG <--> WIS
## C[WIS,EFF]   0.8044019 0.03138961  25.626374 7.755361e-145 EFF <--> WIS
## C[ENG,EFF]   0.7464017 0.03060912  24.384941 2.470795e-131 EFF <--> ENG
## V[si1]       0.2052307 0.01510102  13.590525  4.557998e-42 si1 <--> si1
## V[si2]       0.3042285 0.01850548  16.439912  9.906310e-61 si2 <--> si2
## V[si3]       0.1966617 0.01289154  15.255091  1.522818e-52 si3 <--> si3
## V[si4]       0.3232391 0.02100144  15.391286  1.872739e-53 si4 <--> si4
## V[si5]       0.3475479 0.02503211  13.884081  7.910845e-44 si5 <--> si5
## V[si6]       0.3865663 0.02473449  15.628634  4.646611e-55 si6 <--> si6
## V[si7]       0.3359755 0.02189731  15.343234  3.931117e-53 si7 <--> si7
## V[si8]       0.3145093 0.02064421  15.234745  2.079377e-52 si8 <--> si8
## V[si9]       0.6999811 0.04189340  16.708624  1.134231e-62 si9 <--> si9
## V[si10]      0.5820615 0.03774915  15.419196  1.216088e-53 si10 <--> si10
## V[si11]      0.8933729 0.05247718  17.024026  5.449204e-65 si11 <--> si11
## V[si12]      0.6227930 0.04302111  14.476450  1.706870e-47 si12 <--> si12
## V[si13]      0.4573467 0.03065546  14.918931  2.482176e-50 si13 <--> si13
## V[si14]      0.3807060 0.02525753  15.072966  2.439107e-51 si14 <--> si14
## V[si15]      0.4924139 0.03225694  15.265361  1.301064e-52 si15 <--> si15
## V[si16]      0.4477171 0.02885971  15.513568  2.808320e-54 si16 <--> si16
## V[si17]      0.3836756 0.02518243  15.235849  2.044538e-52 si17 <--> si17
## V[si18]      0.5800615 0.03521116  16.473796  5.660387e-61 si18 <--> si18
## V[si19]      0.3585894 0.02326245  15.414943  1.298846e-53 si19 <--> si19
## V[si20]      0.5187093 0.03803427  13.637944  2.381815e-42 si20 <--> si20
## V[si21]      0.6185608 0.03814391  16.216502  3.855328e-59 si21 <--> si21
## V[si22]      0.5003786 0.02999815  16.680314  1.822588e-62 si22 <--> si22
## V[si23]      0.4477391 0.02610443  17.151840  6.089339e-66 si23 <--> si23
## V[si24]      0.4962187 0.03089210  16.062965  4.638357e-58 si24 <--> si24
```

```
## V[si25]     0.5751351 0.03580959  16.060923  4.793616e-58 si25 <--> si25
## V[si26]     0.5072077 0.03030312  16.737804  6.950558e-63 si26 <--> si26
## V[si27]     0.4756047 0.03209640  14.818008  1.120579e-49 si27 <--> si27
## V[si28]     0.3548095 0.02422488  14.646492  1.418532e-48 si28 <--> si28
## V[si29]     0.5235896 0.03232428  16.198028  5.207018e-59 si29 <--> si29
## V[si30]     0.6575266 0.04082600  16.105584  2.330882e-58 si30 <--> si30
## V[si31]     0.3500031 0.02195291  15.943356  3.168708e-57 si31 <--> si31
## V[si32]     0.3587271 0.02777209  12.916819  3.617910e-38 si32 <--> si32
##
##  Iterations =  84
```

Figure 16.5: Path diagram for the hypothesized structural equation model for job satisfaction data

Source of data: Kline, R. B. (2016). *Principles and Practice of Structural Equation Modeling*, 4th edition. Guilford Press, New York.

```
# Modified from https://www.guilford.com/add/kline/houghton-lavaan.r

# Fully latent model of thought strategies and job satisfaction

#install.packages("lavaan")
library(lavaan)
```

```
## This is lavaan 0.6-3

## lavaan is BETA software! Please report any bugs.

##
## Attaching package: 'lavaan'

## The following objects are masked from 'package:sem':
##
##     cfa, sem
```

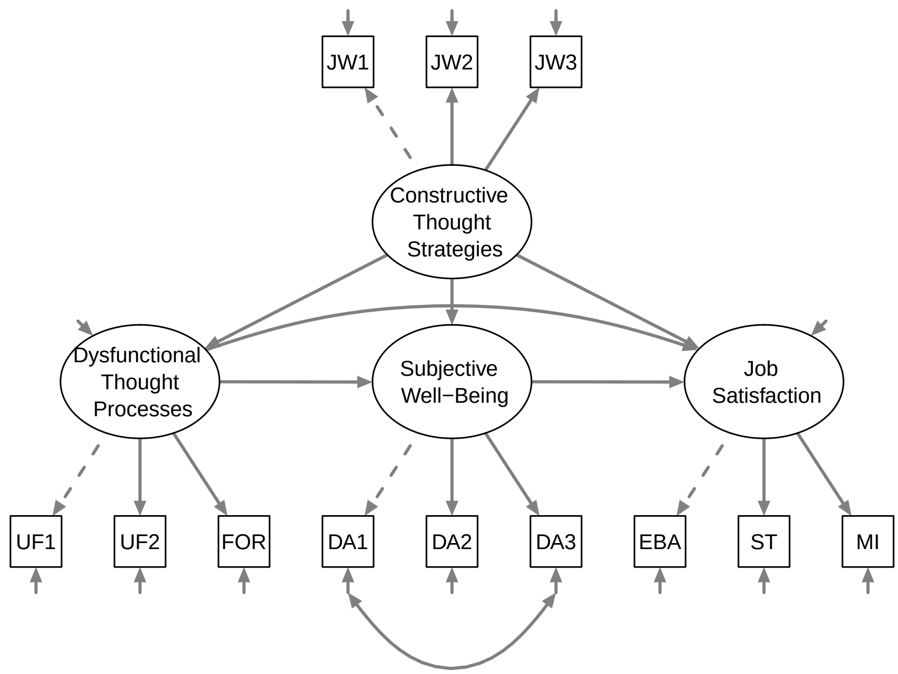
```
# Input the correlations in lower diagonal form
houghtonLower.cor <- '
1.000
 .668 1.000
 .635  .599 1.000
 .263  .261  .164 1.000
 .290  .315  .247  .486 1.000
 .207  .245  .231  .251  .449 1.000
-.206 -.182 -.195 -.309 -.266 -.142 1.000
-.280 -.241 -.238 -.344 -.305 -.230  .753 1.000
-.258 -.244 -.185 -.255 -.255 -.215  .554  .587 1.000
 .080  .096  .094 -.017  .151  .141 -.074 -.111  .016 1.000
 .061  .028 -.035 -.058 -.051 -.003 -.040 -.040 -.018  .284 1.000
 .113  .174  .059  .063  .138  .044 -.119 -.073 -.084  .563  .379 1.000 '

# Name the variables and convert to full correlation matrix
# KV: change names of manifest and factor variables (more according to the article diagram):

houghtonFull.cor <- getCov(houghtonLower.cor,
        names = c("JW1","JW2","JW3", "UF1","UF2","FOR", "DA1","DA2","DA3", "EBA","ST","MI"))

# add the standard deviations and convert to covariances
houghtonFull.cov <- cor2cov(houghtonFull.cor,
        sds = c(0.939, 1.017, 0.937, 0.562, 0.760, 0.524, 0.585, 0.609, 0.731, 0.711, 1.124, 1.001))
houghtonFull.cov
```

```
##            JW1        JW2        JW3         UF1        UF2
## JW1  0.88172100 0.63791528 0.55870031 0.138789834 0.20695560
## JW2  0.63791528 1.03428900 0.57080447 0.149175594 0.24346980
## JW3  0.55870031 0.57080447 0.87796900 0.086361416 0.17589364
## UF1  0.13878983 0.14917559 0.08636142 0.315844000 0.20758032
## UF2  0.20695560 0.24346980 0.17589364 0.207580320 0.57760000
```

```
## FOR  0.10185145  0.13056246  0.11341823  0.073916488  0.17880976
## DA1 -0.11315889 -0.10827999 -0.10688828 -0.101589930 -0.11826360
## DA2 -0.16011828 -0.14926407 -0.13581065 -0.117736752 -0.14116620
## DA3 -0.17709352 -0.18139619 -0.12671520 -0.104759610 -0.14166780
## EBA  0.05341032  0.06941635  0.06262346 -0.006792894  0.08159436
## ST   0.06438160  0.03200702 -0.03686158 -0.036637904 -0.04356624
## MI   0.10621311  0.17713496  0.05533828  0.035441406  0.10498488
##              FOR         DA1         DA2          DA3          EBA
## JW1  0.101851452 -0.11315889 -0.16011828 -0.177093522  0.053410320
## JW2  0.130562460 -0.10827999 -0.14926407 -0.181396188  0.069416352
## JW3  0.113418228 -0.10688828 -0.13581065 -0.126715195  0.062623458
## UF1  0.073916488 -0.10158993 -0.11773675 -0.104759610 -0.006792894
## UF2  0.178809760 -0.11826360 -0.14116620 -0.141667800  0.081594360
## FOR  0.274576000 -0.04352868 -0.07339668 -0.082354460  0.052531524
## DA1 -0.043528680  0.34222500  0.26826754  0.236909790 -0.030779190
## DA2 -0.073396680  0.26826754  0.37088100  0.261320073 -0.048062889
## DA3 -0.082354460  0.23690979  0.26132007  0.534361000  0.008315856
## EBA  0.052531524 -0.03077919 -0.04806289  0.008315856  0.505521000
## ST  -0.001766928 -0.02630160 -0.02738064 -0.014789592  0.226962576
## MI   0.023079056 -0.06968461 -0.04450146 -0.061465404  0.400693293
##               ST          MI
## JW1  0.064381596  0.10621311
## JW2  0.032007024  0.17713496
## JW3 -0.036861580  0.05533828
## UF1 -0.036637904  0.03544141
## UF2 -0.043566240  0.10498488
## FOR -0.001766928  0.02307906
## DA1 -0.026301600 -0.06968461
## DA2 -0.027380640 -0.04450146
## DA3 -0.014789592 -0.06146540
## EBA  0.226962576  0.40069329
## ST   1.263376000  0.42642200
## MI   0.426421996  1.00200100
```

```
# Specify SEM (with Lavaan notation):

JSmodel <- '

# measurement part
CTS =~ EBA + ST + MI
DTP =~ DA1 + DA2 + DA3
SWB =~ UF1 + UF2 + FOR
JS =~ JW1 + JW2 + JW3

# error covariance
UF1 ~~ FOR

# structural part
DTP ~ CTS
SWB ~ CTS + DTP
JS  ~ CTS + DTP + SWB
'

options(fit.indices = c("CFI", "NNFI", "RMSEA", "SRMR")) # (NNFI = TLI)
```

```r
# fit JSmodel to data
JSsem <- sem(JSmodel, sample.cov = houghtonFull.cov, sample.nobs = 263)

longLabels = c("JW1","JW2","JW3", "UF1","UF2","FOR", "DA1","DA2","DA3", "EBA","ST","MI",
               "Constructive \n Thought \n Strategies",
               "Dysfunctional \n Thought \n Processes",
               "Subjective \n Well-Being",
               "Job \n Satisfaction")

semPaths(JSsem, # filetype = "pdf", filename = "JSmodel-KV",
         what = "path", whatLabels = "hide", style = "lisrel", layout = "tree2",
         residuals = TRUE, nodeLabels = longLabels, nCharNodes = 0,
         sizeMan = 4.5, sizeLat = 14, sizeLat2 = 10,
         esize = 2, label.cex = 0.7, label.scale = FALSE, width = 6, height = 5)
```




```r
summary(JSsem, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-3 ended normally after 48 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         31
##
##   Number of observations                           263
##
##   Estimator                                         ML
##   Model Fit Test Statistic                      56.662
```

```
##   Degrees of freedom                                47
##   P-value (Chi-square)                           0.158
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             1087.490
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.991
##   Tucker-Lewis Index (TLI)                       0.987
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3121.267
##   Loglikelihood unrestricted model (H1)      -3092.936
##
##   Number of free parameters                         31
##   Akaike (AIC)                                6304.534
##   Bayesian (BIC)                              6415.270
##   Sample-size adjusted Bayesian (BIC)         6316.985
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.028
##   90 Percent Confidence Interval          0.000  0.052
##   P-value RMSEA <= 0.05                          0.936
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.037
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   CTS =~
##     EBA               1.000                               0.460    0.648
##     ST                1.056    0.178    5.922    0.000    0.486    0.433
##     MI                1.890    0.331    5.717    0.000    0.870    0.870
##   DTP =~
##     DA1               1.000                               0.485    0.830
##     DA2               1.133    0.080   14.105    0.000    0.549    0.904
##     DA3               0.993    0.089   11.175    0.000    0.481    0.660
##   SWB =~
##     UF1               1.000                               0.376    0.671
##     UF2               1.490    0.219    6.799    0.000    0.561    0.739
##     FOR               0.821    0.126    6.535    0.000    0.309    0.591
```

```
##   JS =~
##     JW1              1.000                               0.786    0.839
##     JW2              1.035    0.081   12.770    0.000    0.814    0.802
##     JW3              0.891    0.073   12.145    0.000    0.700    0.749
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   DTP ~
##     CTS             -0.131    0.078   -1.681    0.093   -0.124   -0.124
##   SWB ~
##     CTS              0.067    0.061    1.096    0.273    0.082    0.082
##     DTP             -0.365    0.064   -5.664    0.000   -0.470   -0.470
##   JS ~
##     CTS              0.160    0.120    1.331    0.183    0.093    0.093
##     DTP             -0.242    0.130   -1.863    0.063   -0.149   -0.149
##     SWB              0.797    0.202    3.946    0.000    0.382    0.382
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .UF1 ~~
##    .FOR             -0.043    0.018   -2.390    0.017   -0.043   -0.243
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .EBA              0.292    0.043    6.862    0.000    0.292    0.580
##    .ST               1.022    0.097   10.496    0.000    1.022    0.812
##    .MI               0.242    0.123    1.965    0.049    0.242    0.242
##    .DA1              0.106    0.016    6.783    0.000    0.106    0.311
##    .DA2              0.068    0.017    3.975    0.000    0.068    0.183
##    .DA3              0.300    0.029   10.186    0.000    0.300    0.564
##    .UF1              0.173    0.025    6.877    0.000    0.173    0.550
##    .UF2              0.261    0.044    5.970    0.000    0.261    0.453
##    .FOR              0.178    0.022    8.133    0.000    0.178    0.651
##    .JW1              0.260    0.042    6.231    0.000    0.260    0.297
##    .JW2              0.368    0.050    7.394    0.000    0.368    0.357
##    .JW3              0.384    0.044    8.692    0.000    0.384    0.439
##     CTS              0.212    0.049    4.285    0.000    1.000    1.000
##    .DTP              0.231    0.031    7.579    0.000    0.985    0.985
##    .SWB              0.108    0.025    4.398    0.000    0.763    0.763
##    .JS               0.467    0.066    7.063    0.000    0.755    0.755
##
## R-Square:
##                    Estimate
##     EBA              0.420
##     ST               0.188
##     MI               0.758
##     DA1              0.689
##     DA2              0.817
##     DA3              0.436
##     UF1              0.450
##     UF2              0.547
##     FOR              0.349
##     JW1              0.703
##     JW2              0.643
```

```
##     JW3                 0.561
##     DTP                 0.015
##     SWB                 0.237
##     JS                  0.245
```