

Evaluating SEM predictions

Marjolein Fokkema

2023-09-26

Fit a SEM

```
library("lavaan")
```

```
## This is lavaan 0.6-15  
## lavaan is FREE software! Please report any bugs.
```

```
library("semPlot")  
data(PoliticalDemocracy)  
head(PoliticalDemocracy)
```

```
##      y1      y2      y3      y4      y5      y6      y7      y8      x1  
## 1  2.50 0.000000 3.333333 0.000000 1.250000 0.000000 3.726360 3.333333 4.442651  
## 2  1.25 0.000000 3.333333 0.000000 6.250000 1.100000 6.666666 0.736999 5.384495  
## 3  7.50 8.800000 9.999998 9.199991 8.750000 8.094061 9.999998 8.211809 5.961005  
## 4  8.90 8.800000 9.999998 9.199991 8.907948 8.127979 9.999998 4.615086 6.285998  
## 5 10.00 3.333333 9.999998 6.666666 7.500000 3.333333 9.999998 6.666666 5.863631  
## 6  7.50 3.333333 6.666666 6.666666 6.250000 1.100000 6.666666 0.368500 5.533389  
##      x2      x3  
## 1 3.637586 2.557615  
## 2 5.062595 3.568079  
## 3 6.255750 5.224433  
## 4 7.567863 6.267495  
## 5 6.818924 4.573679  
## 6 5.135798 3.892270
```

```
colnames(PoliticalDemocracy) <- c("z1", "z2", "z3", "z4",  
                                   "y1", "y2", "y3", "y4",  
                                   "x1", "x2", "x3")  
  
mod <- '  
  
  # latent variable definitions  
  ind60 =~ x1 + x2 + x3  
  dem60 =~ z1 + z2 + z3 + z4  
  dem65 =~ y1 + y2 + y3 + y4  
  
  # regressions  
  dem60 ~ ind60
```

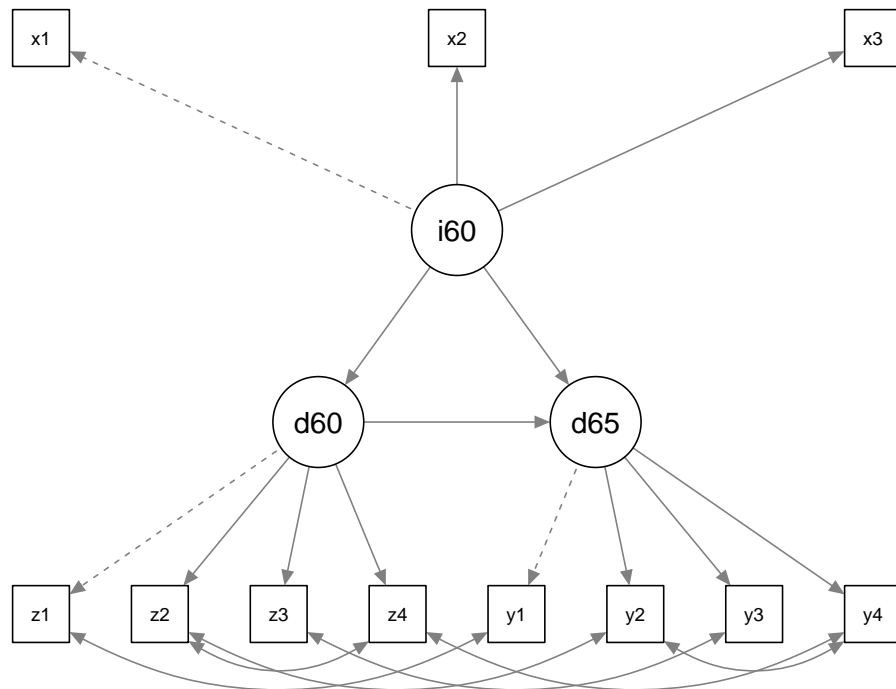
```

dem65 ~ ind60 + dem60

# residual correlations
z1 ~~ y1
z2 ~~ z4 + y2
z3 ~~ y3
z4 ~~ y4
y2 ~~ y4
,

fit <- sem(mod, data = PoliticalDemocracy, meanstructure = TRUE, warn = FALSE)
semPaths(fit, title = FALSE, intercepts = FALSE, residuals = FALSE)

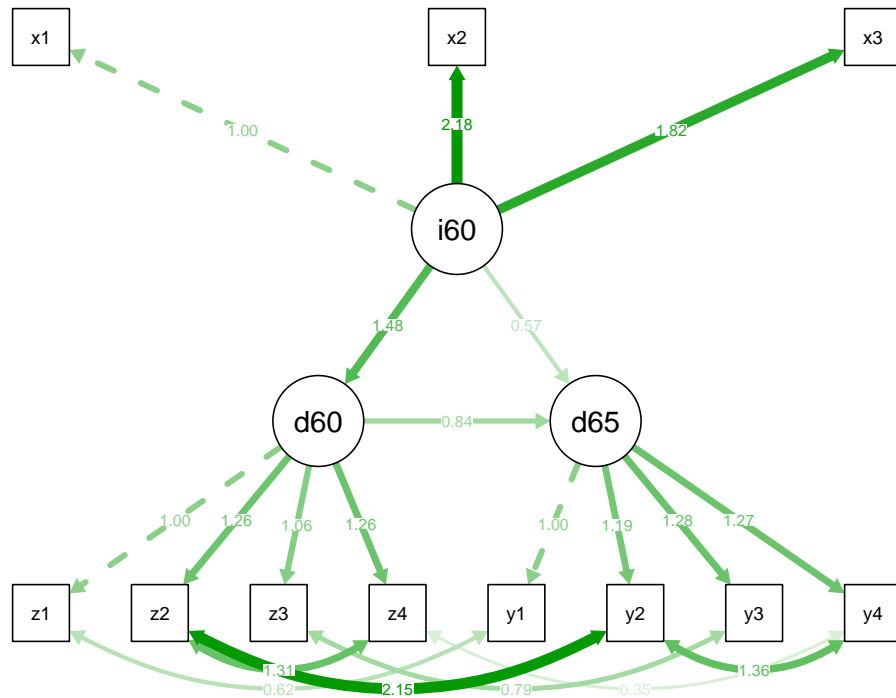
```



```

semPaths(fit, what = "est", intercepts = FALSE, residuals = FALSE)

```



Compute predicted values on the response variables:

```
xnames <- colnames(PoliticalDemocracy)[-c(5, 6, 7, 8)]
ynames <- colnames(PoliticalDemocracy)[c(5, 6, 7, 8)]
lavPredictY(fit, xnames = xnames, ynames = ynames,
            newdata = PoliticalDemocracy[1:10, ])
```

```
##          y1      y2      y3      y4
## [1,]  2.414 -0.302  2.734  0.565
## [2,]  2.253 -0.046  2.989  0.807
## [3,]  7.584  6.324  9.694  7.547
## [4,]  8.642  7.126 10.608  8.475
## [5,]  8.428  4.943  9.698  7.362
## [6,]  6.535  3.448  7.445  5.594
## [7,]  6.466  3.363  7.352  5.501
## [8,]  6.059  3.084  7.148  4.156
## [9,]  3.456  1.733  4.364  2.552
## [10,] 8.494  5.879  9.942  7.824
```

Evaluate predictive accuracy using 10-fold cross validation

```
k <- 10
fold_id <- rep(1:k, length.out = nrow(PoliticalDemocracy))
```

```

pred_y <- matrix(0, nrow = nrow(PoliticalDemocracy), ncol = 4)
for (i in 1:k) {

  ## Fit SEM on training data
  fit <- sem(mod, data = PoliticalDemocracy[fold_id != i, ],
            meanstructure = TRUE, warn = FALSE)

  ## Predict on test data
  pred_y[fold_id == i, ] <- lavPredictY(fit, xnames = xnames, ynames = ynames,
                                         newdata = PoliticalDemocracy[fold_id == i, ])

}

```

```

## Warning in lav_object_post_check(object): lavaan WARNING: some estimated lv
## variances are negative

```

```

## Evaluate performance pooled RMSE
sqrt(colMeans((PoliticalDemocracy[ , ynames] - pred_y)^2))

```

```

##          y1          y2          y3          y4
## 1.616693 2.237663 2.223714 2.190521

```