# R Workshop: CFA & Structural Equation Modeling

Brier Gallihugh, M.S.

2023-06-19

## Table of contents



```
set.seed(5212023)
library(tidyverse)
library(lavaan)
library(psych)
library(semTools)
library(semPlot)

data <- psych::bfi[,16:25]                                    ①

cfa_data <- data[sample(nrow(data),300),]                     ②

sem_data <- lavaan::PoliticalDemocracy %>% na.omit()          ③
```

① Create overall data for CFA
② Randomly sample 300 observations from `data` using `sample()` function
③ Create data for SEM using the `PoliticalDemocracy` data set from the `lavaan` package. Omit missing data using the `na.omit()` function

## Confirmatory Factor Analysis

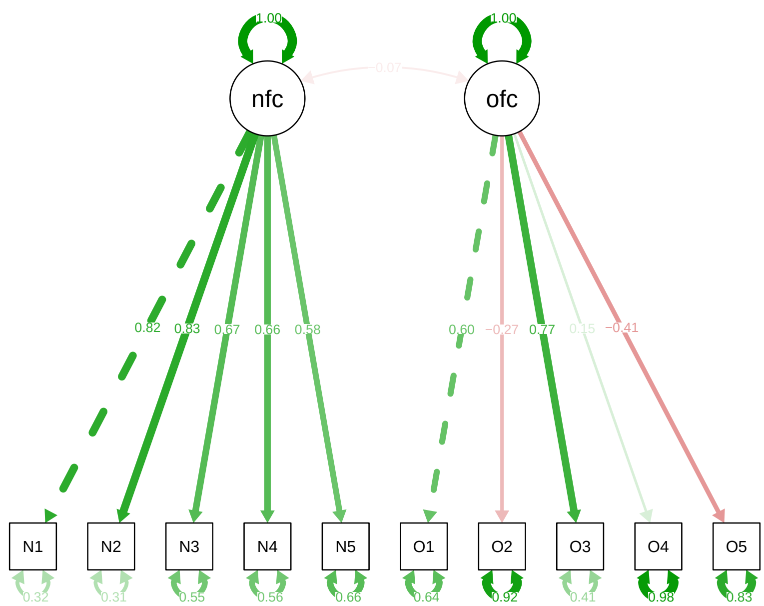
```
# Create CFA Model
cfa_model <- 'nfactor  =~ N1 + N2 + N3 + N4 + N5
              ofactor =~ O1 + O2 + O3 + O4 + O5'

fit_cfa <- cfa(cfa_model, data = cfa_data)                               ①

summary(fit_cfa, fit.measures = TRUE)                                    ②

semPaths(fit_cfa,'std')                                                  ③
```

① Run a CFA on the model above using the `cfa()` function
② Generate CFA output and fit measures using the `summary()` function with the `fit.measures` argument set to `TRUE`
③ Create a basic path diagram of the CFA model using the `semPaths()` function with standardized coefficients using the `std` argument




```
lavaan 0.6.15 ended normally after 39 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
```

```
  Number of model parameters                        21

                                                  Used       Total
  Number of observations                           284         300

Model Test User Model:

  Test statistic                               126.828
  Degrees of freedom                                34
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                               785.605
  Degrees of freedom                                45
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.875
  Tucker-Lewis Index (TLI)                       0.834

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -4737.244
  Loglikelihood unrestricted model (H1)      -4673.830

  Akaike (AIC)                                9516.489
  Bayesian (BIC)                              9593.117
  Sample-size adjusted Bayesian (SABIC)       9526.525

Root Mean Square Error of Approximation:

  RMSEA                                          0.098
  90 Percent confidence interval - lower         0.080
  90 Percent confidence interval - upper         0.117
  P-value H_0: RMSEA <= 0.050                    0.000
  P-value H_0: RMSEA >= 0.080                    0.952

Standardized Root Mean Square Residual:

  SRMR                                           0.084
```

```
Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  nfactor =~
    N1                1.000
    N2                0.979    0.067   14.513    0.000
    N3                0.809    0.071   11.478    0.000
    N4                0.794    0.070   11.382    0.000
    N5                0.746    0.076    9.796    0.000
  ofactor =~
    O1                1.000
    O2               -0.580    0.159   -3.635    0.000
    O3                1.314    0.250    5.249    0.000
    O4                0.266    0.125    2.134    0.033
    O5               -0.799    0.158   -5.051    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  nfactor ~~
    ofactor          -0.070    0.072   -0.967    0.333

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .N1                0.838    0.109    7.720    0.000
   .N2                0.759    0.101    7.491    0.000
   .N3                1.442    0.139   10.387    0.000
   .N4                1.424    0.137   10.427    0.000
   .N5                1.921    0.175   10.953    0.000
   .O1                0.879    0.115    7.630    0.000
   .O2                2.025    0.177   11.468    0.000
   .O3                0.595    0.158    3.769    0.000
   .O4                1.415    0.120   11.787    0.000
   .O5                1.583    0.147   10.733    0.000
    nfactor           1.779    0.224    7.947    0.000
    ofactor           0.491    0.125    3.932    0.000
```

> **Tip**
>
> For SEM and CFA models, the =~ syntax is used. You can interpret it as an "equals" sign more or less

### Structural Equation Modeling

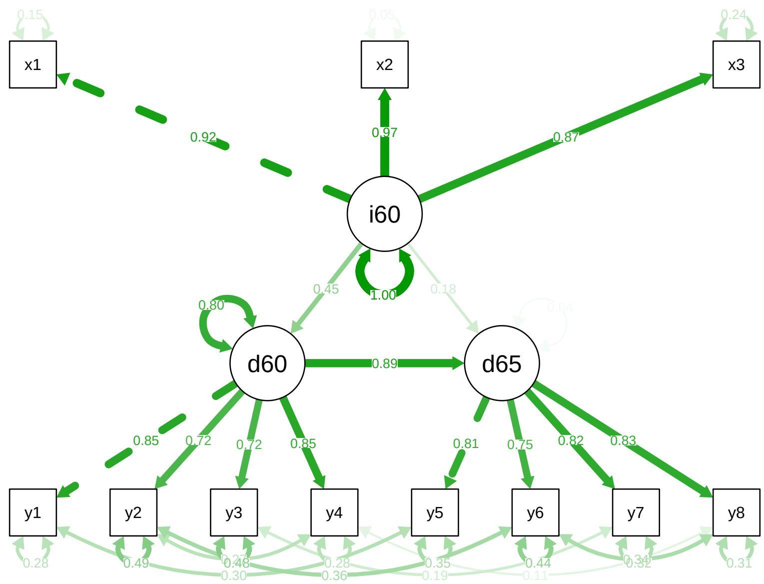
```
# Create SEM Model
sem_model <- 'ind60 =~ x1 + x2 + x3
    dem60 =~ y1 + y2 + y3 + y4
    dem65 =~ y5 + y6 + y7 + y8
    dem60 ~ ind60
    dem65 ~ ind60 + dem60
    y1 ~~ y5
    y2 ~~ y4 + y6
    y3 ~~ y7
    y4 ~~ y8
    y6 ~~ y8'

fit_sem <- sem(sem_model, data = sem_data)                             ①
summary(fit_sem, standardized = TRUE, fit.measures = TRUE)             ②
semPaths(fit_sem,'std')                                                ③
```

① Run an SEM model using the `sem()` function

② Generate a summary of the SEM model with standardized results and fit measures using the `summary()` function with the `standardized` and `fit.measures()` arguments set to `TRUE`

③ Generate a basic path diagram of the SEM model usign the `semPaths()` function with standardized coefficients using the `std` argument.




```
lavaan 0.6.15 ended normally after 68 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        31

  Number of observations                            75

Model Test User Model:

  Test statistic                                38.125
  Degrees of freedom                                35
  P-value (Chi-square)                           0.329

Model Test Baseline Model:

  Test statistic                               730.654
  Degrees of freedom                                55
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.995
```

```
  Tucker-Lewis Index (TLI)                       0.993

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -1547.791
  Loglikelihood unrestricted model (H1)      -1528.728

  Akaike (AIC)                                3157.582
  Bayesian (BIC)                              3229.424
  Sample-size adjusted Bayesian (SABIC)       3131.720

Root Mean Square Error of Approximation:

  RMSEA                                          0.035
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.092
  P-value H_0: RMSEA <= 0.050                    0.611
  P-value H_0: RMSEA >= 0.080                    0.114

Standardized Root Mean Square Residual:

  SRMR                                           0.044

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  ind60 =~
    x1                1.000                               0.670    0.920
    x2                2.180    0.139   15.742    0.000    1.460    0.973
    x3                1.819    0.152   11.967    0.000    1.218    0.872
  dem60 =~
    y1                1.000                               2.223    0.850
    y2                1.257    0.182    6.889    0.000    2.794    0.717
    y3                1.058    0.151    6.987    0.000    2.351    0.722
    y4                1.265    0.145    8.722    0.000    2.812    0.846
  dem65 =~
    y5                1.000                               2.103    0.808
    y6                1.186    0.169    7.024    0.000    2.493    0.746
```

```
    y7                1.280    0.160    8.002    0.000    2.691    0.824
    y8                1.266    0.158    8.007    0.000    2.662    0.828

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  dem60 ~
    ind60             1.483    0.399    3.715    0.000    0.447    0.447
  dem65 ~
    ind60             0.572    0.221    2.586    0.010    0.182    0.182
    dem60             0.837    0.098    8.514    0.000    0.885    0.885

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
 .y1 ~~
   .y5                0.624    0.358    1.741    0.082    0.624    0.296
 .y2 ~~
   .y4                1.313    0.702    1.871    0.061    1.313    0.273
   .y6                2.153    0.734    2.934    0.003    2.153    0.356
 .y3 ~~
   .y7                0.795    0.608    1.308    0.191    0.795    0.191
 .y4 ~~
   .y8                0.348    0.442    0.787    0.431    0.348    0.109
 .y6 ~~
   .y8                1.356    0.568    2.386    0.017    1.356    0.338

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.082    0.019    4.184    0.000    0.082    0.154
   .x2                0.120    0.070    1.718    0.086    0.120    0.053
   .x3                0.467    0.090    5.177    0.000    0.467    0.239
   .y1                1.891    0.444    4.256    0.000    1.891    0.277
   .y2                7.373    1.374    5.366    0.000    7.373    0.486
   .y3                5.067    0.952    5.324    0.000    5.067    0.478
   .y4                3.148    0.739    4.261    0.000    3.148    0.285
   .y5                2.351    0.480    4.895    0.000    2.351    0.347
   .y6                4.954    0.914    5.419    0.000    4.954    0.443
   .y7                3.431    0.713    4.814    0.000    3.431    0.322
   .y8                3.254    0.695    4.685    0.000    3.254    0.315
    ind60             0.448    0.087    5.173    0.000    1.000    1.000
   .dem60             3.956    0.921    4.295    0.000    0.800    0.800
   .dem65             0.172    0.215    0.803    0.422    0.039    0.039
```

Tip

As stated above, for SEM models we want the =~ syntax. For reference, a regression syntax is simply ~ while residuals syntax are ~~. Each of these can as with SEM, be interpreted as an "equals" sign.