NJCS - EFA, CFA, Measurement invariance (Finland, Japan)

Anonymous

Exploratory Factor Analysis - Finland

```
jc1_f <- ff %>%
  dplyr::select(JC_Det1_t1, JC_Det2_t1, JC_Det3_t1, JC_Rel1_t1, JC_Rel2_t1, JC_Rel3_t1,
         JC_Aff1_t1, JC_Aff2_t1, JC_Aff3_t1, JC_Mas1_t1, JC_Mas2_t1, JC_Mas3_t1,
         JC_Mea1_t1, JC_Mea2_t1, JC_Mea3_t1, JC_Aut1_t1, JC_Aut2_t1, JC_Aut3_t1)

mat <- cor(jc1_f, method = "pearson", use = "complete.obs")
mat
corrplot(mat, method = "number", type = "upper", tl.col = "black")

## EFA: JC
fa.parallel(jc1_f, fm = "minres", fa = "fa")
ev <- eigen(cor(mat))
ev

efa <- fa(jc1_f, nfactors = 6, rotate = "oblimin",fm = "minres")
efa
print(efa$loadings, cutoff = 0.3)
```

Confirmatory Factor Analysis - Japan

```
## Model specification - 6 factor
model <- '
          # Defining latent variables
          D =~ JC_Det1_t1 + JC_Det2_t1 + JC_Det3_t1
          R =~ JC_Rel1_t1 + JC_Rel2_t1 + JC_Rel3_t1
          Au =~ JC_Aut1_t1 + JC_Aut2_t1 + JC_Aut3_t1
          Ma =~ JC_Mas1_t1 + JC_Mas2_t1 + JC_Mas3_t1
          Me =~ JC_Mea1_t1 + JC_Mea2_t1 + JC_Mea3_t1
          Af =~ JC_Aff1_t1 + JC_Aff2_t1 + JC_Aff3_t1
'

#  Model estimation
fit1 <- cfa(model,
           data = jf,
           estimator = "ML",
           std.lv=TRUE,
           meanstructure = FALSE)

summary(fit1,
        rsquare = TRUE,
        standardized = TRUE,
```

```
        fit.measures = TRUE)

## Model specification - 5 factor
model <- '
          # Defining latent variables
          D.R =~ JC_Det1_t1 + JC_Det2_t1 + JC_Det3_t1 + JC_Rel1_t1 + JC_Rel2_t1 + JC_Rel3_t1
          Au =~ JC_Aut1_t1 + JC_Aut2_t1 + JC_Aut3_t1
          Ma =~ JC_Mas1_t1 + JC_Mas2_t1 + JC_Mas3_t1
          Me =~ JC_Mea1_t1 + JC_Mea2_t1 + JC_Mea3_t1
          Af =~ JC_Aff1_t1 + JC_Aff2_t1 + JC_Aff3_t1
'

#  Model estimation
fit2 <- cfa(model,
            data = ff,
            estimator = "ML",
            std.lv=TRUE,
            meanstructure = FALSE)

summary(fit2,
        rsquare = TRUE,
        standardized = TRUE,
        fit.measures = TRUE)

modindices(fit2, minimum.value = 20, sort = TRUE)

## Model specification - 2 factor
model <- '
          # Defining latent variables
          AV =~ JC_Det1_t1 + JC_Det2_t1 + JC_Det3_t1 + JC_Rel1_t1 + JC_Rel2_t1 + JC_Rel3_t1
          AP =~ JC_Aut1_t1 + JC_Aut2_t1 + JC_Aut3_t1 + JC_Mas1_t1 + JC_Mas2_t1 + JC_Mas3_t1 +
          JC_Mea1_t1 + JC_Mea2_t1 + JC_Mea3_t1 + JC_Aff1_t1 + JC_Aff2_t1 + JC_Aff3_t1
'

#  Model estimation
fit3 <- cfa(model,
            data = jf,
            estimator = "ML",
            std.lv=TRUE,
            meanstructure = FALSE)

summary(fit3,
        rsquare = TRUE,
        standardized = TRUE,
        fit.measures = TRUE)

## Model specification - 1 factor
model <- '
          # Defining latent variables
          OJC =~ JC_Det1_t1 + JC_Det2_t1 + JC_Det3_t1 + JC_Rel1_t1 + JC_Rel2_t1 + JC_Rel3_t1 +
          JC_Aut1_t1 + JC_Aut2_t1 + JC_Aut3_t1 + JC_Mas1_t1 + JC_Mas2_t1 + JC_Mas3_t1 +
          JC_Mea1_t1 + JC_Mea2_t1 + JC_Mea3_t1 + JC_Aff1_t1 + JC_Aff2_t1 + JC_Aff3_t1
          '
```

```
#  Model estimation
fit4 <- cfa(model,
            data = jf,
            estimator = "ML",
            std.lv=TRUE,
            meanstructure = FALSE)

summary(fit4,
        rsquare = TRUE,
        standardized = TRUE,
        fit.measures = TRUE)

## Model comparisons
anova(fit1, fit2)
anova(fit1, fit3)
anova(fit1, fit4)
```

*** Measurement invariance across time***

```
# Create a tidy dataset
jc_f <- ff %>%
  dplyr::select(JC_Det1_t1, JC_Det2_t1, JC_Det3_t1, JC_Rel1_t1, JC_Rel2_t1, JC_Rel3_t1,
                JC_Aff1_t1, JC_Aff2_t1, JC_Aff3_t1, JC_Mas1_t1, JC_Mas2_t1, JC_Mas3_t1,
                JC_Mea1_t1, JC_Mea2_t1, JC_Mea3_t1, JC_Aut1_t1, JC_Aut2_t1, JC_Aut3_t1,
                JC_Det1_t2, JC_Det2_t2, JC_Det3_t2, JC_Rel1_t2, JC_Rel2_t2, JC_Rel3_t2,
                JC_Aff1_t2, JC_Aff2_t2, JC_Aff3_t2, JC_Mas1_t2, JC_Mas2_t2, JC_Mas3_t2,
                JC_Mea1_t2, JC_Mea2_t2, JC_Mea3_t2, JC_Aut1_t2, JC_Aut2_t2, JC_Aut3_t2,
                JC_Det1_t3, JC_Det2_t3, JC_Det3_t3, JC_Rel1_t3, JC_Rel2_t3, JC_Rel3_t3,
                JC_Aff1_t3, JC_Aff2_t3, JC_Aff3_t3, JC_Mas1_t3, JC_Mas2_t3, JC_Mas3_t3,
                JC_Mea1_t3, JC_Mea2_t3, JC_Mea3_t3, JC_Aut1_t3, JC_Aut2_t3, JC_Aut3_t3)

jc_f <- jc_f %>%
  rename(
    a1 = JC_Det1_t1, a2 = JC_Det1_t2, a3 = JC_Det1_t3,
    b1 = JC_Det2_t1, b2 = JC_Det2_t2, b3 = JC_Det2_t3,
    c1 = JC_Det3_t1, c2 = JC_Det3_t2, c3 = JC_Det3_t3,
    d1 = JC_Rel1_t1, d2 = JC_Rel1_t2, d3 = JC_Rel1_t3,
    e1 = JC_Rel2_t1, e2 = JC_Rel2_t2, e3 = JC_Rel2_t3,
    f1 = JC_Rel3_t1, f2 = JC_Rel3_t2, f3 = JC_Rel3_t3,
    g1 = JC_Aff1_t1, g2 = JC_Aff1_t2, g3 = JC_Aff1_t3,
    h1 = JC_Aff2_t1, h2 = JC_Aff2_t2, h3 = JC_Aff2_t3,
    i1 = JC_Aff3_t1, i2 = JC_Aff3_t2, i3 = JC_Aff3_t3,
    j1 = JC_Mas1_t1, j2 = JC_Mas1_t2, j3 = JC_Mas1_t3,
    k1 = JC_Mas2_t1, k2 = JC_Mas2_t2, k3 = JC_Mas2_t3,
    l1 = JC_Mas3_t1, l2 = JC_Mas3_t2, l3 = JC_Mas3_t3,
    m1 = JC_Mea1_t1, m2 = JC_Mea1_t2, m3 = JC_Mea1_t3,
    n1 = JC_Mea2_t1, n2 = JC_Mea2_t2, n3 = JC_Mea2_t3,
    o1 = JC_Mea3_t1, o2 = JC_Mea3_t2, o3 = JC_Mea3_t3,
    p1 = JC_Aut1_t1, p2 = JC_Aut1_t2, p3 = JC_Aut1_t3,
    q1 = JC_Aut2_t1, q2 = JC_Aut2_t2, q3 = JC_Aut2_t3,
    r1 = JC_Aut3_t1, r2 = JC_Aut3_t2, r3 = JC_Aut3_t3)

jc_tidy <- pivot_longer(jc_f,
```

```
                        cols = a1:r3,
                        names_to = c(".value", "time"),
                        names_pattern = "(.)(.)")

## Longitudinal CFA
# Measurement invariance - configural, metric, scalar
model <- '
          # Defining latent variables
          D =~ a + b + c
          R =~ d + e + f
          Af =~ g + h + i
          Ma =~ j + k + l
          Me =~ m + n + o
          Au =~ p + q + r
'

##  Model estimation
# Configural invariance
fit1 <- cfa(model,
            data = jc_tidy,
            estimator = "ML",
            group = "time")

# Metric invariance
fit2 <- cfa(model,
            data = jc_tidy,
            estimator = "ML",
            group = "time",
            group.equal = "loadings")

# Scalar invariance
fit3 <- cfa(model,
            data = jc_tidy,
            estimator = "ML",
            group = "time",
            group.equal = c("intercepts", "loadings"))

summary(fit3,
        rsquare = TRUE,
        standardized = TRUE,
        fit.measures = TRUE)

# Model comparison
anova(fit1, fit2, fit3)
```