Lab 5 - Path Diagrams & EFA rotation

Adam Garber

Factor Analysis ED 216B - Instructor: Karen Nylund-Gibson

2/3/2020

Lab 5 outline

1. EFA rotations, override the default
2. Specify a confirmatory factor analysis (CFA)
3. Generate a path diagram of your CFA model

Getting started - following the routine...

1. Create an R-Project
2. Install packages
3. Load packages

R-Project instructions:

1. click "NEW PROJECT" (upper right corner of window)
2. choose option "NEW DIRECTORY"
3. choose location of project (on desktop OR in a designated class folder)

Within R-studio under the files pane (bottom right):

1. click "New Folder" and name folder "data"
2. click "New Folder" and name folder "efa_mplus"
3. click "New Folder" and name folder "cfa_mplus"

```
install.packages("semPlot")
install.packages("stargazer")

# Install previously used pacakages (only if your on a new computer!)
install.packages(c("semPlot",
                   "stargazer",
                   "MplusAutomation",
                   "tidyverse",
                   "here"))
```

Lab 5 - Begin

---

loading packages...

```
library(tidyverse)
library(MplusAutomation)
library(rhdf5)
library(here)
library(semPlot)
library(stargazer)
```

read in data

```
trouble_data <- read_csv(here("Lab5_FA", "data", "school_trouble_data.csv"))
```

take a look at the EFA data (same indicators used for lab 4)

```
stargazer(as.data.frame(trouble_data), type="text", digits=1)
```

replicate the SAME random split as last week (do not change the seed)

find the size of half of original sample.

The "floor()" function helps with rounding

```
smp_size <- floor(0.50 * nrow(trouble_data))
```

set the seed to make your partition reproducible

```
set.seed(123)
```

the function "sample()" will pick at random the values of the specified number

```
calibrate_smp <- sample(seq_len(nrow(trouble_data)), size = smp_size)
```

create two samples called "calibrate" & "validate"

```
calibrate <- trouble_data[calibrate_smp, ]
validate <- trouble_data[-calibrate_smp, ]
```

reorder variables & remove column "15" variable

```
# removed BYS24G (due to low loading < .2)
calibrate <-  calibrate %>%
  select(1,3:14,2)

# do the same for the validate sample
validate <-  validate %>%
  select(1,3:14,2)
```

save the calibrate & validate samples using "write_csv"

```
# save calibrate sample
write_csv(calibrate, here("data", "calibrate_trouble_sample.csv"))

# save validate sample
write_csv(validate, here("data", "validate_trouble_sample.csv"))
```

Default rotation: Geomin Oblique

run an EFA with the "calibrate" sample

```
efa_geomin  <- mplusObject(
  TITLE = "Geomin Oblique EFA - LAB 5 DEMO",
  VARIABLE =
    " ! removed BYS24G (due to low loading < .2)
    usevar = BYS22A-BYS22B;",

  ANALYSIS =
    "type = efa 1 3;
     estimator = mlr;
     rotation = geomin; ! this is the default (added to be explicit)
     parallel=50; ! run parallel analysis",

  MODEL = "" ,

  PLOT = "type = plot3;",
  OUTPUT = "sampstat;",

  usevariables = colnames(calibrate),
  rdata = calibrate)

efa_geomin_fit <- mplusModeler(efa_geomin,
                               dataout=here("efa_mplus", "lab5_efa_geomin_oblique.dat"),
                               modelout=here("efa_mplus", "lab5_efa_geomin_oblique.inp"),
                               check=TRUE, run = TRUE, hashfilename = FALSE)
```

Rotation: Varimax Orthogonal

```
efa_varimax  <- mplusObject(
  TITLE = "Varimax Orthogonal EFA - LAB 5 DEMO",
  VARIABLE =
    " ! removed BYS24G (due to low loading < .2)
    usevar = BYS22A-BYS22B;",

  ANALYSIS =
    "type = efa 1 3;
     estimator = mlr;
     rotation = varimax; ! orthogonal (no factor correlations)
     parallel=50; ! run parallel analysis",

  MODEL = "" ,

  PLOT = "type = plot3;",
  OUTPUT = "sampstat;",

  usevariables = colnames(calibrate),
  rdata = calibrate)

efa_varimax_fit <- mplusModeler(efa_varimax,
                            dataout=here("efa_mplus", "lab5_efa_varimax_orthogonal.dat"),
                            modelout=here("efa_mplus", "lab5_efa_varimax_orthogonal.inp"),
                            check=TRUE, run = TRUE, hashfilename = FALSE)
```

Estimate a Confirmatory Factor Analysis (CFA) model

```
cfa_validate  <- mplusObject(
  TITLE = "Geomin Oblique EFA - LAB 5 DEMO",
  VARIABLE =
    "usevar = BYS22A-BYS22B;",

  ANALYSIS =
    "estimator = mlr;",

  MODEL =
    "FACTOR_1 by BYS22A BYS22C BYS22D BYS22E BYS22F BYS22G BYS22H;

     FACTOR_2 BY BYS24A BYS24B BYS24C BYS24D BYS24E BYS24F BYS22B;" ,

  PLOT = "type = plot3;",
  OUTPUT = "sampstat standardized residual modindices (3.84);",

  usevariables = colnames(validate),
  rdata = validate)

cfa_val_fit <- mplusModeler(cfa_validate,
                            dataout=here("cfa_mplus", "lab5_cfa_validate.dat"),
```

```
                              modelout=here("cfa_mplus", "lab5_cfa_validate.inp"),
                              check=TRUE, run = TRUE, hashfilename = FALSE)
```

create a path diagram of the CFA model

```
# Read in the model to R within the "cfa_mplus" folder
cfa_output <- readModels(here("cfa_mplus"))

# Plot model:
semPaths(cfa_output,
         intercepts=FALSE,
         fixedStyle = c(1))
```

play with some formatting

```
# Plot model:
semPaths(cfa_output, "std",
         intercepts=FALSE,
         fixedStyle = c(1),
         color= list(lat = c("light blue"," light green")),
        )
```

End of Lab 5
