# Calibration / Validation

A Course in `MplusAutomation`

*Adam Garber*

---

**Outline**

1. Randomly split data into 2 equal parts (calibration & validation samples)
2. Introduction to using iterators or loops with `mplusObject()`

---

**Getting started - following the routine...**

1. Create an R-Project
2. Install packages (ONLY IF NEEDED)
3. Load packages

**Folder structure:**

Parent folder:

- `04-splits-iterators`

Nested folders:

- `data`
- `efa_mplus`
- `efa_mplus2`
- `figures`

---

## Begin

---

DATA SOURCE: This exercise utilizes the NCES public-use dataset: Education Longitudinal Study of 2002 (Lauff & Ingels, 2014) See website: nces.ed.gov

---

**loading packages...**

```
library(janitor)
library(tidyverse)
library(haven)
library(MplusAutomation)
library(here)
library(corrplot)
library(glue)
```

**read in the raw dataset**

```
raw_data <- read_spss("https://garberadamc.github.io/project-site/data/els_sub1_spss.sav")
```

**create a subset of the dataset called `school_trouble`**

```
school_trouble <- raw_data %>%
  select(41:55)
```

**make a new codebook from the `school_trouble` subset**

```
sjPlot::view_df(school_trouble)
```

**write a `CSV` datafile**

```
write_csv(school_trouble, here("04-splits-iterators", "data", "school_trouble_data.csv"))
```

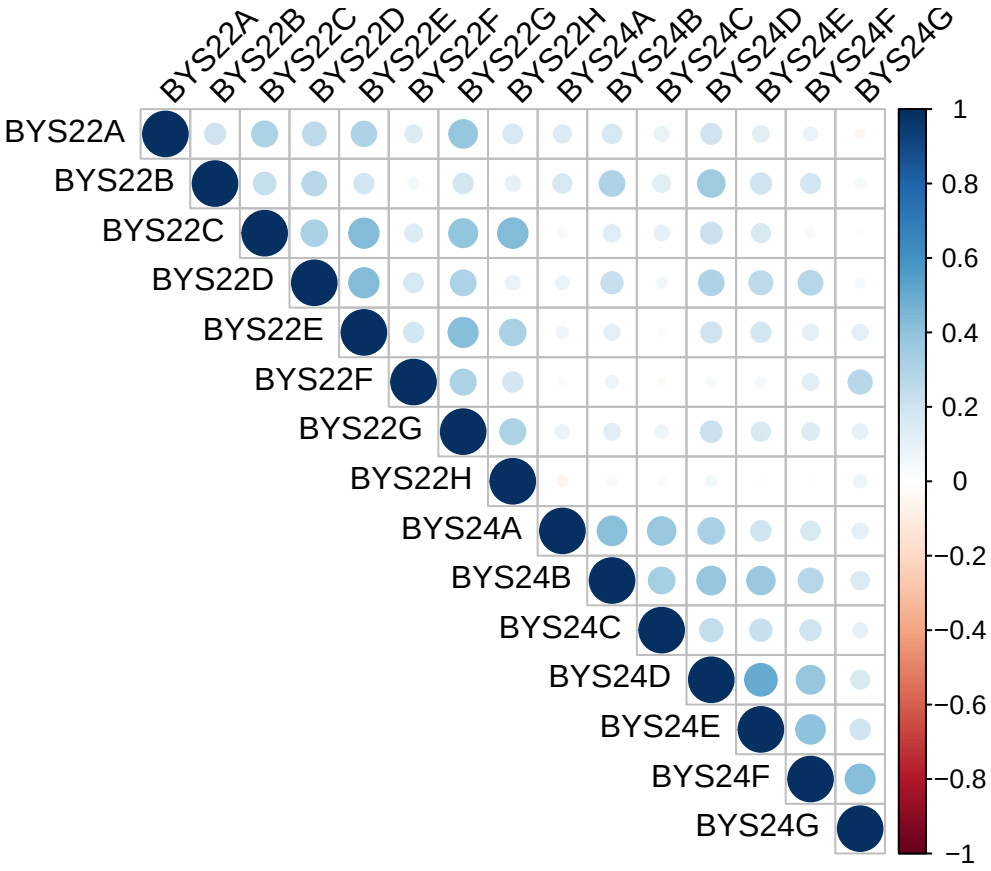
**read the unlabeled data back into R**

```
mplus_ready_data <- read_csv(here("04-splits-iterators", "data", "school_trouble_data.csv"))
```

**check items to see if reverse coding is needed**

```
cor_matrix <- cor(mplus_ready_data, use = "pairwise.complete.obs")

corrplot(cor_matrix, method="circle",
         type = "upper",
         tl.col="black",
         tl.srt=45)
```




---

## Randomly split a sample into 2 equal parts

---

- Get $n$-size of half of original sample using `nrow()`
- The `floor()` function helps with rounding

```
smp_size <- floor(0.50 * nrow(mplus_ready_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(mplus_ready_data)), size = smp_size)
```

**create two samples called “calibrate” & “validate”**

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

**Run EFA with the `calibrate` sample**

```
m_efa_1  <- mplusObject(
  TITLE = "School Trouble EFA",
  VARIABLE =
    "usevar = BYS22A-BYS24G;",

  ANALYSIS =
    "type = efa 1 5;
     estimator = mlr;
     parallel=50; ! run parallel analysis",

  MODEL = "" ,

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

  usevariables = colnames(calibrate),
  rdata = calibrate)

m_efa_1_fit <- mplusModeler(m_efa_1,
               dataout=here("04-splits-iterators", "efa_mplus", "efa1_trouble.dat"),
               modelout=here("04-splits-iterators", "efa_mplus", "efa1_trouble.inp"),
               check=TRUE, run = TRUE, hashfilename = FALSE)
```

**read into R an Mplus output file using the `readModels()` function**

```
efa_summary <- readModels(here("04-splits-iterators", "efa_mplus", "efa1_trouble.out"), quiet = TRUE)
```

---

## Introduction to MplusAutomation with iterators (`validate` sample)

---

**Alternate way to run an EFA with the `validate` sample**

```
m_efa_k15  <- lapply(1:5, function(k) {

  m_efa  <- mplusObject(
```

```
    TITLE =
      "School Trouble EFA",

    VARIABLE =
      "usevar = BYS22A-BYS24G;",

    ANALYSIS =
      paste("type=efa", k, k),

    PLOT =
      "type = plot3;",

    OUTPUT =
      "sampstat;",

    usevariables = colnames(validate),
    rdata = validate)

  m_efa_fit <- mplusModeler(m_efa,
                dataout=sprintf(here("04-splits-iterators", "efa_mplus2", "efa_trouble.dat"), k),
                modelout=sprintf(here("04-splits-iterators", "efa_mplus2", "efa_%d_trouble.inp"), k),
                check=TRUE, run = TRUE, hashfilename = FALSE)

})
```

## References


Hallquist, M. N., & Wiley, J. F. (2018). MplusAutomation: An R Package for Facilitating Large-Scale Latent Variable Analyses in Mplus. Structural equation modeling: a multidisciplinary journal, 25(4), 621-638.

Horst, A. (2020). Course & Workshop Materials. GitHub Repositories, https://https://allisonhorst.github.io/

Muthén, L.K. and Muthén, B.O. (1998-2017). Mplus User's Guide. Eighth Edition. Los Angeles, CA: Muthén & Muthén

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686


---

## Further resources & examples here:

https://garberadamc.github.io/project-site/

https://www.adam-garber.com/

---