Lab 2 - Exploratory Factor Analysis

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

Adam Garber

1/12/2020

EXERCISE 00: MAKE A NEW 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"

IGNORE/SKIP (return here if you recieve an *error* when loading packages)

1. First check if the packages load (run lines 32-39)

2. If an error is returned for package(s)

3. then run the following lines of code

4. AFTER ANY INSTALLATION... MUST RE-LOAD PACKAGES! (run lines 32-39 again)

** **NOTE:** **the following code ONLY needs to be run once per computer!**

```
# to install just the new package "MVN"
install.packages("MVN")

# to install all packages
install.packages(c("MVN", "MplusAutomation", "haven", "tidyverse",
                   "here", "corrplot", "kableExtra"))
```

IF package rhdf5 does not load then run:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("rhdf5")
```

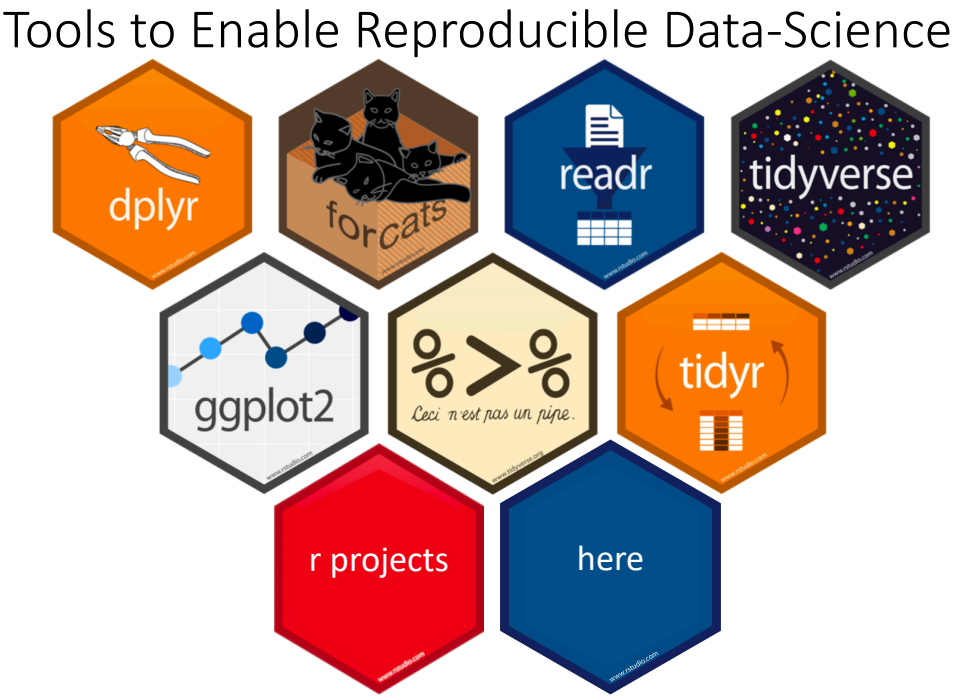

Figure 1: Picture adapted from, **Allison Horst** - Environmental Systems Management (ESM)

LINK: allisonhorst.github

EXERCISE 00: loading packages

```
library(MVN)
library(MplusAutomation)
library(haven)
library(rhdf5)
library(tidyverse)
library(here)
library(corrplot)
library(kableExtra)
```

EXERCISE 1: READ IN DATA TO R ENVIRONMENT

```
lab_data <- read_spss(here("data", "lab_efa_cfa_data.sav"))
```

EXERCISE 2: SUBSET

```
# make a subset of all the student reported variables

by_student <- lab_data %>%
  select(22:145)

# make another subset (just the variables we will use for the EFA)

schl_safe <- lab_data %>%
  select(
    "BYS20A", "BYS20B", "BYS20C", "BYS20D", "BYS20E", "BYS20F", "BYS20G", # F1
    "BYS20H", "BYS20I", "BYS20J", "BYS20K", "BYS20L", "BYS20M", "BYS20N", # F2
    "BYS21A", "BYS21B", "BYS21C", "BYS21D", "BYS21E", # F3
    "BYSEX", "BYRACE", "BYSTLANG" # add some covariates or grouping variables
  )

# subset the first six indicators
inds_6 <- schl_safe %>%
  select(BYS20A:BYS20F)

# subset of the indicators only using the (-) symbol
indicators <- schl_safe %>%
  select(-BYSEX, -BYRACE, -BYSTLANG)

# change class of indicator variables to numeric
indicators %>%
  modify_at(c(1:19), as.numeric) %>%
  str()
```

EXERCISE 3: UNIVARIATE & MULTIVARIATE DIAGNOSTICS

create univariate histograms

```
# data_obeject[rows,columns] for example schl_safe[,1:6] is the first 6 columns

mvn(data = schl_safe[,1:6], univariatePlot = "histogram")
```

create univariate qq-plots

```
mvn(data = as.matrix(inds_6), univariatePlot = "qqplot")
```

create multivariate qq-plots

```
# doesn't work
mvn(data = indicators, multivariatePlot = "qq")
```

run diagnostics all at once

```
result = mvn(data = iris[-4],
             subset = "Species",
             univariatePlot = "histogram",
             multivariatePlot = "qq",
             multivariateOutlierMethod = "adj",
             showOutliers = TRUE, showNewData = TRUE)

#### Multivariate Normality Result
result$multivariateNormality
### Univariate Normality Result
result$univariateNormality
### Descriptives
result$Descriptives
### Multivariate Outliers
result$multivariateOutliers
### New data without multivariate outliers
result$newData
```

LINK: RUN SHINY MVC

EXERCISE 4: REVERSE CODE

reverse indicators so scale has consistent meaning for factor interpretation

expected factors based on item wording:

- Factor 1: "school climate", higher values indicate postive school climate
- Factor 2: "safety", higher values indicate safe school conditions
- Factor 3: "clear rules", higher values indicate clear communication of rules

```
# Reverse code the following variables:

cols = c("BYS20A", "BYS20B", "BYS20C", # FACTOR 1: school climate
         "BYS20E", "BYS20F", "BYS20G",
         "BYS21A", "BYS21B", "BYS21C", "BYS21D", "BYS21E") # FACTOR 3: clear rules

# the number "5" will change: Use "number of categories" + 1 (e.g., 4 + 1)
schl_safe[ ,cols] <-  5 - schl_safe[ ,cols]
```

EXERCISE 5: CHECK CORRELATIONS

check correlations to see if coding was correct (all blue, no red)

```
f1_cor <- cor(schl_safe[1:7], use = "pairwise.complete.obs")
f2_cor <- cor(schl_safe[8:14], use = "pairwise.complete.obs")
f3_cor <- cor(schl_safe[15:19], use = "pairwise.complete.obs")
```

```r
corrplot(f1_cor,
         method = "circle",
         type = "upper")

corrplot(f2_cor,
         method = "circle",
         type = "upper")

corrplot(f3_cor,
         method = "circle",
         type = "upper")

# DISCOVERING PATTERNS in large correlation matrices:
# The correlation matrix can be reordered according to the correlation coefficient.
# This is important to identify the hidden structure and pattern in the matrix.
# "hclust" for hierarchical clustering order can be used...

# ADD CODE LINE: order="hclust"
```

EXERCISE 6: PREPARE DATASETS

```r
### prepare datasets, remove SPSS labeling

# write a CSV datafile (preferable format for reading into R, without labels)
write_csv(schl_safe, here("data", "lab_fa_hsls_subset.csv"))

# write a SPSS datafile (preferable format for reading into SPSS, labels are preserved)
write_sav(schl_safe, here("data", "lab_fa_hsls_subset.sav"))

# read the unlabeled data back into R
fa_data <- read_csv(here("data", "lab_fa_hsls_subset.csv"))

# write an Mplus DAT datafile
prepareMplusData(fa_data, here("data", "lab_fa_hsls_subset.dat"))
```

EXERCISE 7: MPLUS AUTOMATION - GET DESCRIPTIVES

```r
## RUN TYPE = BASIC ANALYSIS (indicators: school climate, safety, clear rules )

m_basic  <- mplusObject(
  TITLE = "RUN TYPE = BASIC ANALYSIS - LAB 2 DEMO",
  VARIABLE =
    " ! an mplusObject() will always need a 'usevar' statement
      ! ONLY specify variables to use in analysis
```

```
      ! lines of code in MPLUS ALWAYS end with a semicolon ';'
    usevar =
 BYS20A BYS20B BYS20C BYS20D BYS20E BYS20F BYS20G
 BYS20H BYS20I BYS20J BYS20K BYS20L BYS20M BYS20N
 BYS21A BYS21B BYS21C BYS21D BYS21E;",

  ANALYSIS =
    "type = basic" ,

  MODEL = "" ,

  PLOT = "",

  OUTPUT = "",

  usevariables = colnames(fa_data),   # tell MplusAutomation the column names to use
  rdata = fa_data)                    # this is the data object used (must be un-label)

m_basic_fit <- mplusModeler(m_basic,
                    dataout=here("efa_mplus", "basic_Lab2_DEMO.dat"),
                    modelout=here("efa_mplus", "basic_Lab2_DEMO.inp"),
                    check=TRUE, run = TRUE, hashfilename = FALSE)

## END: TYPE = BASIC ANALYSIS
```

EXERCISE 8: EXPLORATORY FACTOR ANALYSIS (EFA)

```
## EXPLORATORY FACTOR ANALYSIS: (indicators: school climate, safety, clear rules)

m_efa_1  <- mplusObject(
  TITLE = "FACTOR ANALYSIS EFA - LAB 2 DEMO",
  VARIABLE =
    "usevar =
 BYS20A BYS20B BYS20C BYS20D BYS20E BYS20F BYS20G
 BYS20H BYS20I BYS20J BYS20K BYS20L BYS20M BYS20N
 BYS21A BYS21B BYS21C BYS21D BYS21E;",

  ANALYSIS =
 "type = efa 1 5;    ! run efa of 1 through 5 factor models
  estimator = MLR;   ! using the ROBUST ML Estimator
  parallel=50;       ! run the parallel analysis for viewing in elbow plotå
  ",

  MODEL = "" ,

  PLOT = "type = plot3;",
```

```
  OUTPUT = "sampstat standardized residual modindices (3.84);",

  usevariables = colnames(fa_data),
  rdata = fa_data)

m_efa_1_fit <- mplusModeler(m_efa_1,
                            dataout=here("efa_mplus", "EFA1_Lab2_DEMO.dat"),
                            modelout=here("efa_mplus", "EFA1_Lab2_DEMO.inp"),
                            check=TRUE, run = TRUE, hashfilename = FALSE)

## END: EXPLORATORY FACTOR ANALYSIS
```

EXERCISE 9: EFA REDUCED INDICATOR SET

Removed items: (loadings <.5 and/or cross-loadings)

How to make a tribble table?

```
lab_tools <- tribble(
  ~"Items", ~"Factor 1", ~"Factor 2",  ~"Factor 3",
 #----------/-------------/------------/-----------/,
  "BYS20C"  ,  " 0.149 "  ,  "0.168*"  ,  "0.120 "  ,
  "BYS20D"  ,  " 0.075 "  ,  "0.338*"  ,  "0.082 "  ,
  "BYS20H"  ,  " 0.345*"  ,  "0.307*"  ,  "0.061 "  ,
  "BYS20I"  ,  "-0.032 "  ,  "0.386*"  ,  "0.167 "  ,
  "BYS20L"  ,  " 0.004 "  ,  "0.400*"  ,  "0.377*"  ,
  "BYS21B"  ,  " 0.418*"  ,  "0.024 "  ,  "0.187*"  ,
)

lab_tools %>%
  kable("latex", booktabs = T, linesep = "") %>%
  kable_styling(latex_options = c("striped"),
                full_width = F,
                position = "left")
```

Items	Factor 1	Factor 2	Factor 3
BYS20C	0.149	0.168*	0.120
BYS20D	0.075	0.338*	0.082
BYS20H	0.345*	0.307*	0.061
BYS20I	-0.032	0.386*	0.167
BYS20L	0.004	0.400*	0.377*
BYS21B	0.418*	0.024	0.187*

```
## EXPLORATORY FACTOR ANALYSIS - REDUCED SET

m.step1  <- mplusObject(
  TITLE = "FACTOR ANALYSIS EFA - REDUCED SET - LAB 2 DEMO",
  VARIABLE =
    "usevar =
     BYS20A BYS20B BYS20E BYS20F BYS20G
     ! removed: BYS20C BYS20D
     BYS20J BYS20K BYS20M BYS20N
     ! removed:BYS20H BYS20I BYS20L
     BYS21A BYS21C BYS21D BYS21E
     ! removed: BYS21B
     ;",

  ANALYSIS =
    "type = efa 1 5;    ! run efa of 1 through 5 factor models
     estimator = MLR;   ! using the ROBUST ML Estimator
     parallel=50;       ! run the parallel analysis for viewing in elbow plot
    ",

  MODEL = "" ,

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

  usevariables = colnames(fa_data),
  rdata = fa_data)

m.step1.fit <- mplusModeler(m.step1,
                            dataout=here("efa_mplus", "EFA2_Lab1_DEMO.dat"),
                            modelout=here("efa_mplus", "EFA2_Lab1_DEMO.inp"),
                            check=TRUE, run = TRUE, hashfilename = FALSE)

## END: EXPLORATORY FACTOR ANALYSIS OF - REDUCED SET
```

References:

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