

Lab 6 - CFA *Roulette*

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

Adam Garber

2/8/2020



Outline lab 6 - Factor Analysis Content

1. Unit loading identification (ULI)
 2. Unit Variance identification (UVI)
 3. Interpreting Residuals
 4. Modification Indices
-

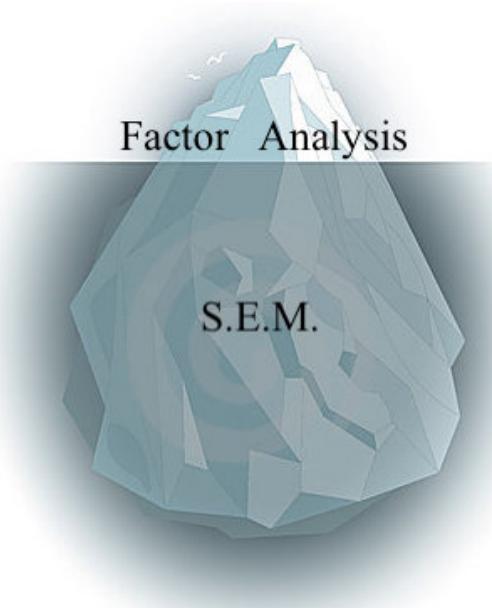
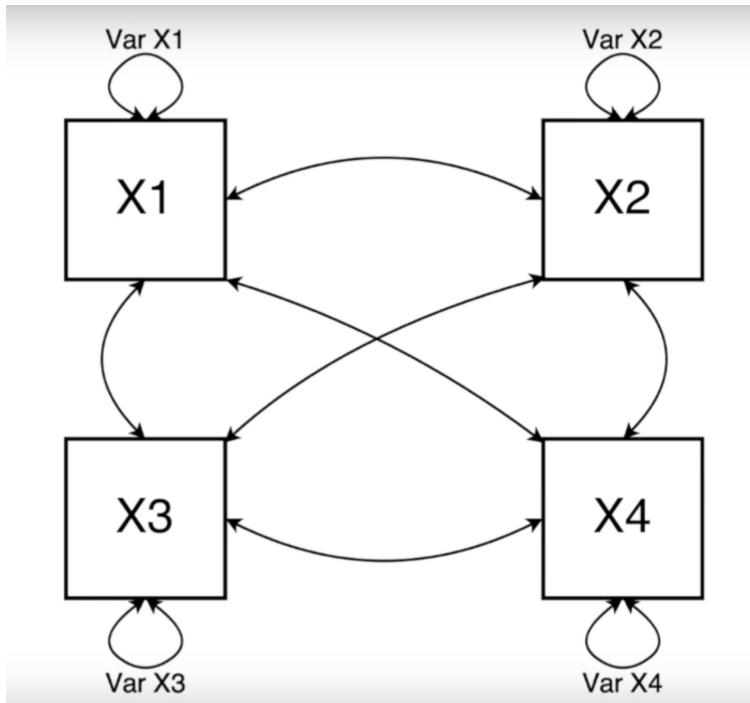
CFA *Roulette* - rules of the game:

- Create a pool of items ordered based on similarity of relationship (correlation) using the hclust algorithm
- Split into 2 pools or clusters of items
- Spin the wheel: randomly choose 5 items from each pool (each of our CFA's will be different!)
- Use these 2 sets of items as the indicators in a 2 factor CFA
- Choose between 1 - 2 modifications from the mod indices to "improve" your model
- Upload your **final** model output file (.out) to GS portal. Use the following naming convention:

firstname_lastname_cfa.out e.g., adam_g_cfa.out

- Lastly, we will upload all models into a table and see who had the “best” model
 - Let the best BIC wins!
-

A visual way to understand the variance / covariance matrix



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 “class_mplus”
3. click “New Folder” and name folder “cfa_mplus”

install a couple NEW packages for this weeks lab

```
install.packages("glue")
install.packages("beepR")
install.packages("praise")
install.packages("") 
install.packages("") 

# Alternatively, install ALL packages (only if your on a new computer!)
install.packages(c("tidyverse",
                  "MplusAutomation",
                  "rhdf5",
                  "here",
                  "semPlot",
                  "stargazer",
                  "corrplot",
                  "glue",
                  "kableExtra",
                  "beepR",
                  "praise"))
```

Lab 5 - Begin

loading packages...

```
library(tidyverse)
library(MplusAutomation)
library(rhdf5)
library(here)
library(semPlot)
library(stargazer)
library(corrplot)
library(glue)
library(kableExtra)
library(beep)
library(praise)
beep(2)
```

read in data

```
lab_data <- read_csv(here("data", "lab_ef_a_cfa_data.csv"))

beep(1)
praise("You are totally ${adjective}! Super ${EXCLAMATION}!")

# praise(0) # picks a random sound
```

Subsetting all Ordinal type variables

```
ordinal_data <- lab_data %>%
  select(21:145)

beep(1)
praise()
```

Order variables based on correlations & create 2 cluster item pools to pull from

```
big_matrix <- cor(ordinal_data, use = "pairwise.complete.obs")

corrplot(big_matrix,
  method = "color",
  type = "upper",
  order = "hclust",
  addrect = 2,
```

```

tl.cex = .5, tl.col = "black")

order <- corrMatOrder(big_matrix, order="hclust")

order_data <- ordinal_data %>%
  select(order)

clust1 <- order_data %>%
  select(BYS89G:BYS86D)

clust2 <- order_data %>%
  select(BYS87E:BYS38B)

cor_c1 <- cor(clust1, use = "pairwise.complete.obs")

corrplot(cor_c1,
  method = "color",
  type = "upper",
  tl.col = "black")

cor_c2 <- cor(clust2, use = "pairwise.complete.obs")

corrplot(cor_c2,
  method = "color",
  type = "upper",
  tl.col = "black")

beep(1)
praise()

```

Try your luck! (select columns at random)

```

# select 5 columns at random for factor1

# set.seed(*****) # setting a seed is optional, use to replicate same solution

roulette_1 <- clust1 %>%
  select(sample(ncol(clust1), 5))

f1_vars <- colnames(roulette_1)

beep(1)
praise()

```

```
# select 5 columns at random for factor2
roulette_2 <- clust2 %>%
  select(sample(ncol(clust2), 5))

f2_vars <- colnames(roulette_2)

beep(1)
praise()
```

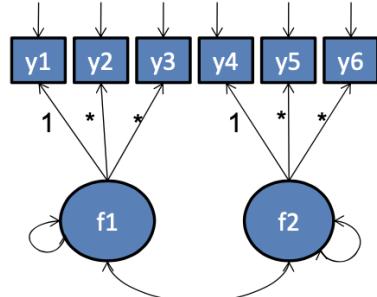
take a look at the items in roulette 1

```
stargazer(as.data.frame(roulette_1), type="text", digits=1)
```

take a look at the items in roulette 2

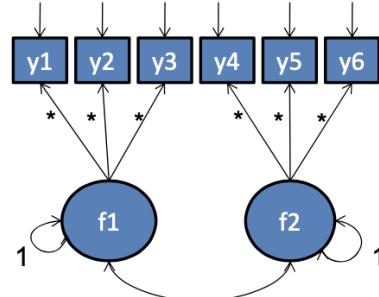
```
stargazer(as.data.frame(roulette_2), type="text", digits=1)
```

Unit Loading Identification (ULI)



Unstandardized Factors

Unit Variance Identification (UVI)



Standardized Factors

CFA Roulette

```
# DEFAULT: Unit Loading Identification (ULI)
cfa_ULI <- mplusObject(
  TITLE = "CFA - ULI - LAB 6 DEMO",
  VARIABLE =
    glue(
      "usevar =\n      {noquote(f1_vars[1])}\n      {noquote(f1_vars[2])}")
```

```

{noquote(f1_vars[3])}
{noquote(f1_vars[4])}
{noquote(f1_vars[5])}
{noquote(f2_vars[1])}
{noquote(f2_vars[2])}
{noquote(f2_vars[3])}
{noquote(f2_vars[4])}
{noquote(f2_vars[5])}
;"),
ANALYSIS =
"estimator = mlr;" , 

MODEL =
glue(
"FACTOR_1 by
{noquote(f1_vars[1])}
{noquote(f1_vars[2])}
{noquote(f1_vars[3])}
{noquote(f1_vars[4])}
{noquote(f1_vars[5])}; 

FACTOR_2 by
{noquote(f2_vars[1])}
{noquote(f2_vars[2])}
{noquote(f2_vars[3])}
{noquote(f2_vars[4])}
{noquote(f2_vars[5])};") , 

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

usevariables = colnames(order_data),
rdata = order_data)

cfa_ULI_fit <- mplusModeler(cfa_ULI,
                               dataout=here("cfa_mplus", "lab6_cfa_ULI.dat"),
                               modelout=here("cfa_mplus", "lab6_cfa_ULI.inp"),
                               check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()

```

```

# OVERRIDE DEFAULT: Unit Variance Identification
cfa_UVI <- mplusObject(
  TITLE = "CFA - UVI - LAB 6 DEMO",
  VARIABLE =
  glue(
    "usevar =
{noquote(f1_vars[1])}
{noquote(f1_vars[2])}
{noquote(f1_vars[3])}
{noquote(f1_vars[4])}")

```

```

{noquote(f1_vars[5])}
{noquote(f2_vars[1])}
{noquote(f2_vars[2])}
{noquote(f2_vars[3])}
{noquote(f2_vars[4])}
{noquote(f2_vars[5])};" ),
ANALYSIS =
"estimator = mlr;",

MODEL =
glue(
"FACTOR_1 by
{noquote(f1_vars[1])}* !estimate first variable loading
{noquote(f1_vars[2])}
{noquote(f1_vars[3])}
{noquote(f1_vars[4])}
{noquote(f1_vars[5])};"

FACTOR_1@1; !fix variance of factor to 1

FACTOR_2 by
{noquote(f2_vars[1])}*
{noquote(f2_vars[2])}
{noquote(f2_vars[3])}
{noquote(f2_vars[4])}
{noquote(f2_vars[5])};

FACTOR_2@1;" ) ,

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

usevariables = colnames(order_data),
rdata = order_data)

cfa_UVI_fit <- mplusModeler(cfa_UVI,
                               dataout=here("cfa_mplus", "lab6_cfa_UVI.dat"),
                               modelout=here("cfa_mplus", "lab6_cfa_UVI.inp"),
                               check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()

```

Residual Output:

- The “output = residual;” option is used to request residuals for the observed variables in the analysis.

- Residuals are computed for the model estimated means/intercepts/thresholds and the model estimated covariances/correlations/residual correlations.
- Residuals are computed as the difference between the value of the observed sample statistic and its model estimated value.
- Standardized and normalized residuals are available for continuous outcomes with TYPE=GENERAL and maximum likelihood estimation.
- Standardized residuals are computed as the difference between the value of the observed sample statistic and its model estimated value divided by the standard deviation of the difference between the value of the observed sample statistic and its model estimated value. Standardized residuals are approximate z-scores.
- Normalized residuals are computed as the difference between the value of the observed sample statistic and its model estimated value divided by the standard deviation of the value of the observed sample statistic (Mplus 6 User's Guide, p. 644).

#	<i>Standardized Residuals (z-scores) for Covariances</i>				
#	<i>BYS86D</i>	<i>BYS89M</i>	<i>BYS89B</i>	<i>BYS89H</i>	<i>BYS90A</i>
#	-----	-----	-----	-----	-----
# <i>BYS86D</i>	999.000				
# <i>BYS89M</i>	3.191	999.000			
# <i>BYS89B</i>	-2.207	0.157	999.000		
# <i>BYS89H</i>	0.462	21.983	-0.213	999.000	
# <i>BYS90A</i>	1.244	0.520	-1.780	-1.448	0.315
# <i>BYS87E</i>	-2.402	-8.142	3.679	-4.084	-2.417
# <i>BYS87C</i>	2.602	6.951	999.000	999.000	0.143
# <i>BYS21D</i>	-1.319	2.332	1.374	0.811	-0.302
# <i>BYS20B</i>	-0.440	2.480	1.689	1.533	-1.888
# <i>BYS20F</i>	1.784	1.677	0.370	1.931	-1.238
#					
#	<i>BYS87E</i>	<i>BYS87C</i>	<i>BYS21D</i>	<i>BYS20B</i>	<i>BYS20F</i>
#	-----	-----	-----	-----	-----
# <i>BYS87E</i>	0.320				
# <i>BYS87C</i>	3.282	999.000			
# <i>BYS21D</i>	-1.831	-1.951	0.043		
# <i>BYS20B</i>	-3.563	-1.255	2.936	999.000	
# <i>BYS20F</i>	-1.944	-1.790	999.000	999.000	0.365

Each value can be mapped to a z-score distribution in which they can be interpreted as standard deviations from an ideal zero residual that signifying perfect reproduction of the variance-covariance matrix. Values above 1.96 or 2.00 indicates statistically significantly over- or under- estimation at a $p < .05$ level.

Modification Indices:

#	<i>M.I.</i>	<i>E.P.C.</i>	<i>Std E.P.C.</i>	<i>StdYX E.P.C.</i>
#				
# <i>BY Statements</i>				
#				

# FACTOR_1 BY BYS87E	18.781	-2.612	-0.546	-0.603
# FACTOR_1 BY BYS87C	10.943	-1.988	-0.416	-0.500
# FACTOR_1 BY BYS20B	5.174	1.199	0.251	0.327
# FACTOR_1 BY BYS20F	7.830	1.676	0.351	0.497
# FACTOR_2 BY BYS89M	11.748	1.582	0.463	0.518
# FACTOR_2 BY BYS89B	7.505	-1.359	-0.398	-0.419
# FACTOR_2 BY BYS90A	4.788	-0.650	-0.190	-0.315
#				
# WITH Statements				
#				
# BYS89M WITH BYS86D	4.507	0.050	0.050	0.115
# BYS89B WITH BYS86D	4.497	-0.053	-0.053	-0.115
# BYS87E WITH BYS89M	26.032	-0.169	-0.169	-0.261
# BYS87E WITH BYS89B	7.894	0.096	0.096	0.143
# BYS87C WITH BYS86D	6.548	0.053	0.053	0.124
# BYS87C WITH BYS89M	13.201	0.108	0.108	0.196
# BYS87C WITH BYS89B	60.860	-0.242	-0.242	-0.419
# BYS21D WITH BYS89M	4.395	0.053	0.053	0.107
# BYS21D WITH BYS87E	4.022	-0.054	-0.054	-0.095
# BYS21D WITH BYS87C	5.370	-0.057	-0.057	-0.117
# BYS20B WITH BYS89M	4.366	0.058	0.058	0.110
# BYS20B WITH BYS87E	11.120	-0.097	-0.097	-0.163
# BYS20F WITH BYS87E	4.211	-0.057	-0.057	-0.117
# BYS20F WITH BYS21D	10.269	0.066	0.066	0.175
# BYS20F WITH BYS20B	15.315	0.093	0.093	0.235

Add a modification indice

Note: alter the modification following modification statement for your model

```
cfa_mod1 <- mplusObject(
  TITLE = "CFA - mod1 - LAB 6 DEMO",
  VARIABLE =
    glue(
      "usevar =
        {noquote(f1_vars[1])}
        {noquote(f1_vars[2])}
        {noquote(f1_vars[3])}
        {noquote(f1_vars[4])}
        {noquote(f1_vars[5])}
        {noquote(f2_vars[1])}
        {noquote(f2_vars[2])}
        {noquote(f2_vars[3])}
        {noquote(f2_vars[4])}
        {noquote(f2_vars[5])};" ),
  ANALYSIS =
    "estimator = mlr;" ,
  MODEL =
    glue(
```

```

"FACTOR_1 by
{noquote(f1_vars[1])}* !estimate first variable loading
{noquote(f1_vars[2])}
{noquote(f1_vars[3])}
{noquote(f1_vars[4])}
{noquote(f1_vars[5])};

FACTOR_1@1; !fix variance of factor to 1

FACTOR_2 by
{noquote(f2_vars[1])}*
{noquote(f2_vars[2])}
{noquote(f2_vars[3])}
{noquote(f2_vars[4])}
{noquote(f2_vars[5])};

FACTOR_2@1;

! ****CHANGE TO REFLECT YOUR MODS****
BYS89V WITH BYS89J; !estimate residual correlation mod indice" ) ,

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

usevariables = colnames(order_data),
rdata = order_data

cfa_mod1_fit <- mplusModeler(cfa_mod1,
                               dataout=here("cfa_mplus", "lab6_cfa_mod1.dat"),
                               modelout=here("cfa_mplus", "lab6_cfa_mod1.inp"),
                               check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()

```

Add a second modification from the mod indices statements

```

cfa_mod2 <- mplusObject(
  TITLE = "CFA - mod1 - LAB 6 DEMO",
  VARIABLE =
    glue(
      "usevar =
{noquote(f1_vars[1])}
{noquote(f1_vars[2])}
{noquote(f1_vars[3])}
{noquote(f1_vars[4])}
{noquote(f1_vars[5])}
{noquote(f2_vars[1])}
{noquote(f2_vars[2])}
{noquote(f2_vars[3])}
{noquote(f2_vars[4])}
{noquote(f2_vars[5])};"
  ),

```

```

ANALYSIS =
  "estimator = mlr;",

MODEL =
  glue(
    "FACTOR_1 by
    {noquote(f1_vars[1])}* !estimate first variable loading
    {noquote(f1_vars[2])}
    {noquote(f1_vars[3])}
    {noquote(f1_vars[4])}
    {noquote(f1_vars[5])};"

    FACTOR_1@1; !fix variance of factor to 1

    FACTOR_2 by
    {noquote(f2_vars[1])}*
    {noquote(f2_vars[2])}
    {noquote(f2_vars[3])}
    {noquote(f2_vars[4])}
    {noquote(f2_vars[5])};

    FACTOR_2@1;

  ! ****CHANGE TO REFLECT YOUR MODS****
  BYS89V WITH BYS89J; !estimate residual correlation mod indice
  BYS89D WITH BYS89H ; ") ,

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

usevariables = colnames(order_data),
rdata = order_data)

cfa_mod2_fit <- mplusModeler(cfa_mod2,
                               dataout=here("cfa_mplus", "lab6_cfa_mod2.dat"),
                               modelout=here("cfa_mplus", "lab6_cfa_mod2.inp"),
                               check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()

```

Collect class output files - upload to GauchoSpace portal

- Download 1 class .out file per person using the naming convention (**firstname_lastname_cfa.out** e.g., **adam_g_cfa.out**)
- Read in all files & create a table
- The best BIC wins!

```

best_models <- readModels(here("class_mplus"), recursive = TRUE)

best_table <- LatexSummaryTable(best_models,
                                keepCols=c("Filename",
                                           "BIC"),
                                sortBy = "BIC")

best_table %>%
  kable(booktabs = T, linesep = "") %>%
  kable_styling(c("striped"),
                full_width = F,
                position = "left")

beep(1)
praise("${EXCLAMATION}!")

```

Calculate Satora-Bentler scaled Chi-square difference test (use with MLR estimator)

See website: stats.idre.ucla.edu-mplus-faq-how-can-i-compute-a-chi-square-test-for-nested-models-with-the-r

- SB0 = null model Chi-square value
- SB1 = alternate model Chi-square value
- c0 = null model scaling correction factor
- c1 = alternate model scaling correction factor
- d0 = null model degrees of freedom
- d1 = alternate model degrees of freedom
- df = Chi-square test degrees of freedom

```

# Identifying all the necessary variables
cfa_models <-readModels(here("cfa_mplus"))

SB0 <- cfa_models[["lab6_cfa_UVI.out"]][["summaries"]][["ChiSqM_Value"]]
SB1 <- cfa_models[["lab6_cfa_mod1.out"]][["summaries"]][["ChiSqM_Value"]]
c0  <- cfa_models[["lab6_cfa_UVI.out"]][["summaries"]][["ChiSqM_ScalingCorrection"]]
c1  <- cfa_models[["lab6_cfa_mod1.out"]][["summaries"]][["ChiSqM_ScalingCorrection"]]
d0  <- cfa_models[["lab6_cfa_UVI.out"]][["summaries"]][["ChiSqM_DF"]]
d1  <- cfa_models[["lab6_cfa_mod1.out"]][["summaries"]][["ChiSqM_DF"]]
df  <- d0-d1

# Satora-Bentler scaled Difference test equations
cd <- (((d0*c0)-(d1*c1))/(d0-d1))
t  <- (((SB0*c0)-(SB1*c1))/cd)

# Chi-square and degrees of freedom
t
df

```

```
# Significance test  
pchisq(t, df, lower.tail=FALSE)
```

End of Lab 6

