CFA Roulette

Adam Garber

Norwegian University of Science and Technology - A Course in `MplusAutomation`

May 30, 2021

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
- Let the best BIC wins!

A visual way to understand the variance / covariance matrix

***Figure.* Picture adapted from {`OpenMx`} documentation.**

***Figure.* Seeing the forest from the trees.**

Lab 5 - Begin

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

---

loading packages...

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

Change starting location to folder `06-cfa-roulette`

```r
source("rep_functions.R")

change_here(glue("{project_location}/06-cfa-roulette"))

here()
```

```
## [1] "/Users/agarber/github/NTNU-workshop/06-cfa-roulette"
```

```r


lab_data <- read_csv("https://garberadamc.github.io/project-site/data/els_sub4.csv")

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

read in data

```
## [1] "You are totally premium! Super GEE!"
```

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

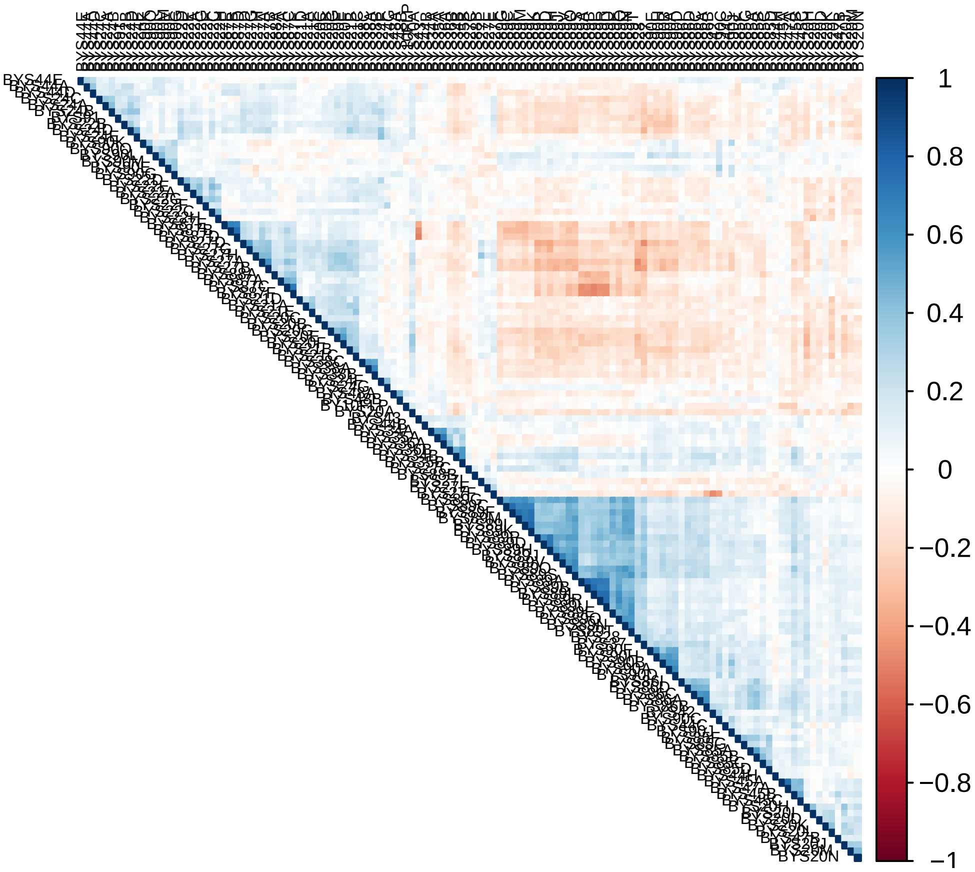
```r


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

beep(1)
praise()
```

Subsetting all ordinal variables

```
## [1] "You are geometric!"
```

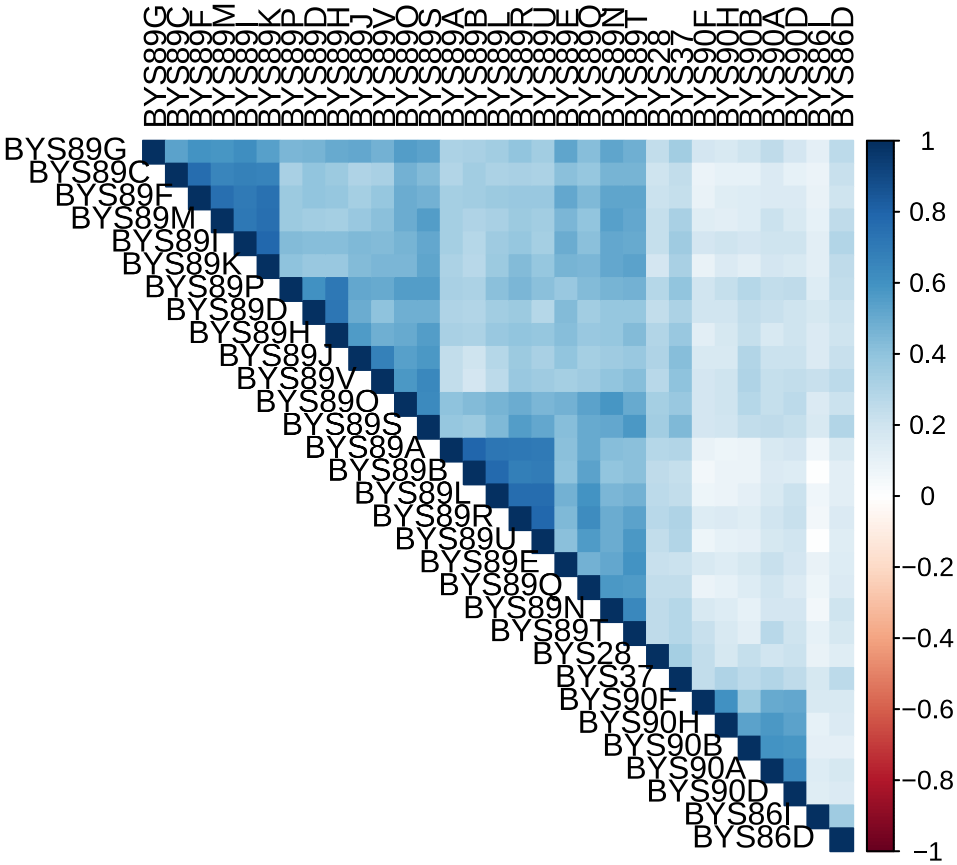
Order variables based on correlations & create 2 cluster item pools to pull from

```
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")
```

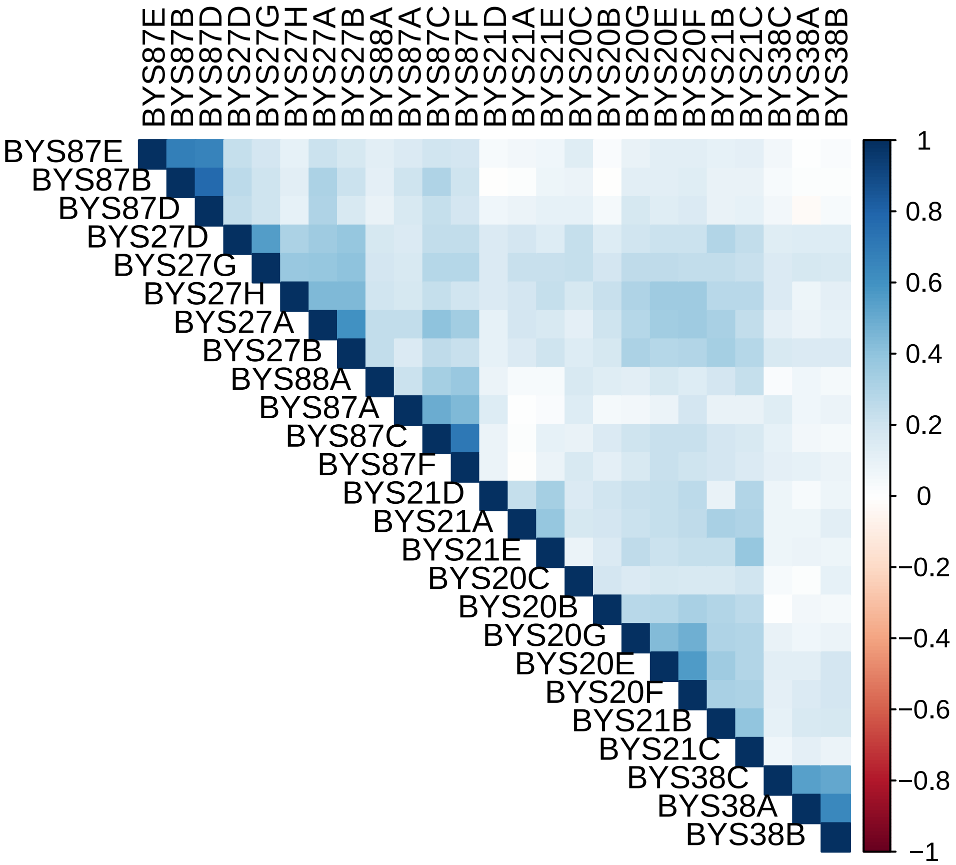


```
order <- corrMatOrder(big_matrix, order="hclust")

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")
```



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
set.seed(123)

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

f1_vars <- colnames(roulette_1)

beep(1)
praise()
```

```
## [1] "You are great!"
```

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

```
f2_vars <- colnames(roulette_2)

beep(1)
praise()
```

```
## [1] "You are cat's meow!"
```

```


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

take a look at the items in roulette 1

```
##
## =====================================================
## Statistic  N  Mean St. Dev. Min Pctl(25) Pctl(75) Max
## -----------------------------------------------------
## BYS86D    605 2.4    0.6    1.0   2.0      3.0    3.0
## BYS89B    545 2.4    0.9    1.0   2.0      3.0    4.0
## BYS89E    532 2.9    0.9    1.0   2.0      4.0    4.0
## BYS89A    549 2.5    0.9    1.0   2.0      3.0    4.0
## BYS89F    538 2.7    0.9    1.0   2.0      3.0    4.0
## -----------------------------------------------------
```

```


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

take a look at the items in roulette 2

```
##
## =====================================================
## Statistic  N  Mean St. Dev. Min Pctl(25) Pctl(75) Max
## -----------------------------------------------------
## BYS20G    711 2.3    0.8    1.0   2.0      3.0    4.0
## BYS21C    715 2.4    0.9    1.0   2.0      3.0    4.0
## BYS87C    563 2.8    0.8    1.0   2.0      3.0    4.0
## BYS27G    714 1.8    0.8    1.0   1.0      2.0    4.0
## BYS20F    710 2.1    0.7    1.0   2.0      2.8    4.0
## -----------------------------------------------------
```

***Figure*: Picture adapted from slide by Dr. Karen Nylund-Gibson**

CFA *Roulette*

```
# DEFAULT: Unit Loading Identification (ULI)

cfa_ULI  <- mplusObject(
  TITLE = "CFA - ULI - 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])}
    {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", "cfa_ULI.dat"),
               modelout=here("cfa_mplus", "cfa_ULI.inp"),
               check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()
```

```
# OVERRIDE DEFAULT: Unit Varianvce 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", "cfa_UVI.dat"),
               modelout=here("cfa_mplus", "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).

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

```
#                                    M.I.     E.P.C.  Std E.P.C.  StdYX E.P.C.
#
# BY Statements
#
# 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

```
cfa_mod1  <- mplusObject(
  TITLE = "CFA UVI - 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 MODIFICATIONS**** !!!
    ![XXXXXX] WITH [XXXXXX]; ! estimate residual correlation mod indice
    BYS89F   WITH BYS89E;
    ") ,

  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", "cfa_mod1.dat"),
                            modelout=here("cfa_mplus", "cfa_mod1.inp"),
                            check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()
```

Note: alter the modification following modification statement for your model

```
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**** !!!
    ![XXXXXX] WITH [XXXXXX]; !estimate residual correlation mod indice
    ![XXXXXX] WITH [XXXXXX];
    BYS89F   WITH BYS89E;
    BYS20F   WITH BYS20G;
    ") ,

  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", "cfa_mod2.dat"),
                            modelout=here("cfa_mplus", "cfa_mod2.inp"),
                            check=TRUE, run = TRUE, hashfilename = FALSE)

beep(1)
praise()
```

Add a second modification from the mod indices statements

Collect class output files - upload to GauchoSpace portal

- Download 1 class .out file per person using the naming convention
- Read in all files & create a table
- The best BIC wins!

```
best_models <- readModels(here("cfa_mplus"), quiet = 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")
```

	Filename	BIC
2	cfa_mod2.out	13933.85
1	cfa_mod1.out	13962.15
3	cfa_ULI.out	14050.40
4	cfa_UVI.out	14050.40

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

```
## [1] "OH!"
```

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-m

- 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

```r
# Identifying all the necessary variables
cfa_models <-readModels(here("cfa_mplus"), quiet = TRUE)

SB0 <- cfa_models[["cfa_UVI.out"]][["summaries"]][["ChiSqM_Value"]]
SB1 <- cfa_models[["cfa_mod1.out"]][["summaries"]][["ChiSqM_Value"]]
c0  <- cfa_models[["cfa_UVI.out"]][["summaries"]][["ChiSqM_ScalingCorrection"]]
c1  <- cfa_models[["cfa_mod1.out"]][["summaries"]][["ChiSqM_ScalingCorrection"]]
d0  <- cfa_models[["cfa_UVI.out"]][["summaries"]][["ChiSqM_DF"]]
d1  <- cfa_models[["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
```

```
## [1] 71.05706
```

```r
df
```

```
## [1] 1
```

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

```
## [1] 3.470413e-17
```

END

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