Exploratory Factor Analysis (EFA)

A Course in `MplusAutomation`

Adam Garber

Load packages

```
library(MplusAutomation)
library(tidyverse)
library(haven)
library(here)
library(gt)
library(sjPlot)
```

Load data example

Data source. This tutorial utilizes the NCES public-use data called the Education Longitudinal Study of 2002 (Lauff & Ingels, 2014) This data can be found on the NCES website. Note that all examples used are for purposes of illustration only and are not intended to be interpreted substantively.

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

Prepare `data.frame` for analysis (select, reorder, & rename columns)

```
schl_safe <- els_data %>%
  select(
    'stu_tch' ="BYS20A", 'sc_spirt'="BYS20B",
    'tch_good'="BYS20E", 'tch_intr'="BYS20F",
    'tch_prai'="BYS20G", 'stu_dwn' ="BYS20I",
    'not_safe'="BYS20J", 'disr_lrn'="BYS20K",
    'gangs'   ="BYS20M", 'rac_fght'="BYS20N",
    'sch_rule'="BYS21A", 'pun_same'="BYS21C",
    'strict'  ="BYS21D", 'pun_rule'="BYS21E",
    'female'  ="BYSEX",  'stu_race'="BYRACE",
    'eng_natv'="BYSTLANG"
  )
```

View meta-data from labeled SPSS file

```
# This meta-data or codebook can be downloaded as a PDF using the "print" option
sjPlot::view_df(schl_safe)
```

Look at variables for EFA example

Applied Example: School Safety[1]

Name	Variable Description
stu_tch	Students get along well with teachers
sc_spirt	There is real school spirit
tch_good	The teaching is good
tch_intr	Teachers are interested in students
tch_prai	Teachers praise effort
stu_dwn	In class often feels put down by students
not_safe	Does not feel safe at this school
disr_lrn	Disruptions get in way of learning
gangs	There are gangs in school
rac_fght	Racial/ethnic groups often fight
sch_rule	Everyone knows what school rules are
pun_same	Punishment same no matter who you are
strict	School rules are strictly enforced
pun_rule	Students know punishment for broken rules
Covariates	
female	Student reported gender (male/female)
stu_race	Student reported race
eng_natv	Whether English is student's native language

[1]Note. All scale indicators have 4-point Likert response options ranging from Strongly Agree (1) to Strongly Disagree (4).

Reverse code indicators for factor interpretation

Expected factors based on theory and item similarity:

- Factor 1: *School climate*, higher values indicate positive school climate
- Factor 2: *safety*, higher values indicate safe school conditions
- Factor 3: *clear rules*, higher values indicate clear communication of rules

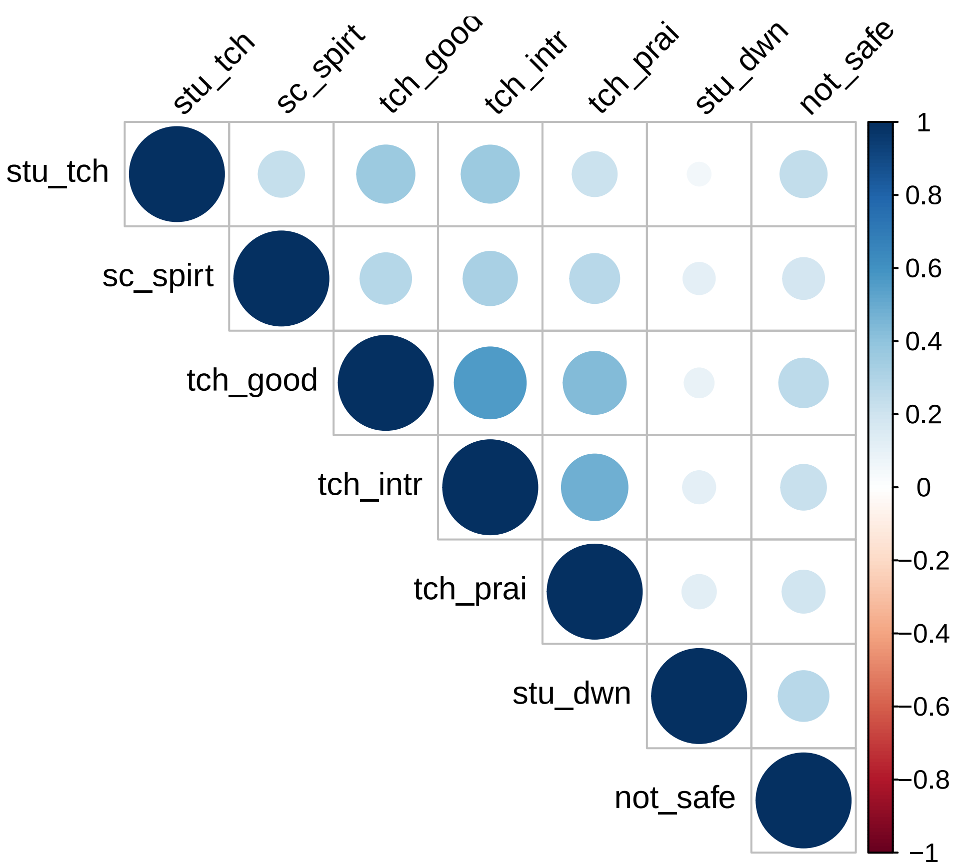
```
# select items to reverse code
cols = c("stu_tch", "sc_spirt",                          # Factor 1: school climate
         "tch_good", "tch_intr", "tch_prai",             # Factor 2: safety
         "sch_rule", "pun_same", "strict", "pun_rule") # Factor 3: clear rules
```

Use formula: `number of response categories` + `1` (e.g., `4 + 1` = `5`)

```
 ### The number `5` in syntax below will change with applied context ###

schl_safe[ ,cols] <-  5 - schl_safe[ ,cols]
```

Check correlations to see if coding was correct (i.e., correlation is consistent within factor)

```
library(corrplot)

f1_cor <- cor(schl_safe[1:7], use = "pairwise.complete.obs")

corrplot(f1_cor,
         method = "circle",
         type = "upper",
         tl.col="black",
         tl.srt=45)
```



Create sub-folders for project organization:

1. create folder named `data`
2. create folder named `figures`
3. create folder named `efa_mplus`
4. create folder named `wls_efa`

Prepare datasets

Prepare dataset for `mplusObject()` by removing SPSS labels

```
# write a CSV datafile (preferable format for reading into R, without labels)
write_csv(schl_safe, here("03-efa", "data", "els_efa_ready.csv"))

# read the unlabeled data back into R
efa_data <- read_csv(here("03-efa", "data", "els_efa_ready.csv"))
```

Estimate Exploratory Factor Analysis Model (EFA)

Model 1: Default rotation

```
efa_1  <- mplusObject(

  TITLE = "EFA",

  VARIABLE =
 "usevar =
  stu_tch sc_spirt tch_good tch_intr
  tch_prai stu_dwn not_safe disr_lrn
  gangs rac_fght sch_rule pun_same strict pun_rule;",

  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 elbow plot
  ",

  MODEL = "" ,

  PLOT = "type = plot3;",

  OUTPUT = "sampstat;",

  usevariables = colnames(efa_data),
  rdata = efa_data)

efa_1_fit <- mplusModeler(efa_1,
             dataout=here("03-efa","efa_mplus","efa_els.dat"),
             modelout=here("03-efa","efa_mplus","efa_els.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

Create table summarizing model fit

```
model_fit <- LatexSummaryTable(efa_1_fit,
  keepCols=c("Title","Parameters", "LL",
             "ChiSqM_Value","ChiSqM_DF", "ChiSqM_PValue",
             "RMSEA_Estimate","RMSEA_90CI_LB","RMSEA_90CI_UB",
             "CFI", "TLI","SRMR")) %>%
  mutate(Title = c("1-Factor","2-Factor","3-Factor","4-Factor","5-Factor")) %>%
  mutate_at(vars(contains("RMSEA")), ~format(., nsmall = 3)) %>%
  unite(CI, RMSEA_90CI_LB:RMSEA_90CI_UB, sep=", ", remove = TRUE) %>%
  mutate(CI = paste0("(", CI, ")")) %>%
  unite(RMSEA, RMSEA_Estimate:CI, sep=" ", remove = TRUE)

model_fit %>%
  gt() %>%
  tab_header(
    title = md("**Table 1**"),
    subtitle = md("*Summary of Model Fit Indices*")) %>%
  cols_label(
    Title = "Model",
    Parameters = md("Par"),
    LL = md("*LL*"),
    ChiSqM_Value = md("Chi^2"),
    ChiSqM_PValue = md("*p-value*"),
    ChiSqM_DF = md("*df*"),
    RMSEA = "RMSEA (90% CI)" ) %>%
  tab_options(column_labels.font.weight = "bold") %>%
  fmt(c(6), fns = function(x) ifelse(x<0.001,"<.001",
                                     scales::number(x, accuracy = 0.01)))
```

Table 1

Summary of Model Fit Indices

Model	Par	*LL*	Chi^2	*df*	*p-value*	RMSEA (90% CI)	CFI	TLI	SRMR
1-Factor	42	-10460.82	483.023	77	<.001	0.086 (0.078, 0.093)	0.714	0.662	0.078
2-Factor	55	-10310.32	250.673	64	<.001	0.064 (0.055, 0.072)	0.868	0.813	0.048
3-Factor	67	-10215.92	92.392	52	<.001	0.033 (0.022, 0.044)	0.972	0.950	0.027
4-Factor	78	-10180.40	32.369	41	0.83	0.000 (0.000, 0.016)	1.000	1.000	0.014
5-Factor	88	-10172.91	21.900	31	0.89	0.000 (0.000, 0.014)	1.000	1.000	0.010

Plot Parallel Analysis & Eigenvalues

Extract relevant data & prepare data.frame for plot

```
x <- list(EFA=efa_1_fit$results$gh5$efa$eigenvalues,
          Parallel=efa_1_fit$results$gh5$efa$parallel_average)

plot_data <- as_data_frame(x)
plot_data <- cbind(Factor = paste0(1:nrow(plot_data)), plot_data)

plot_data <- plot_data %>%
```

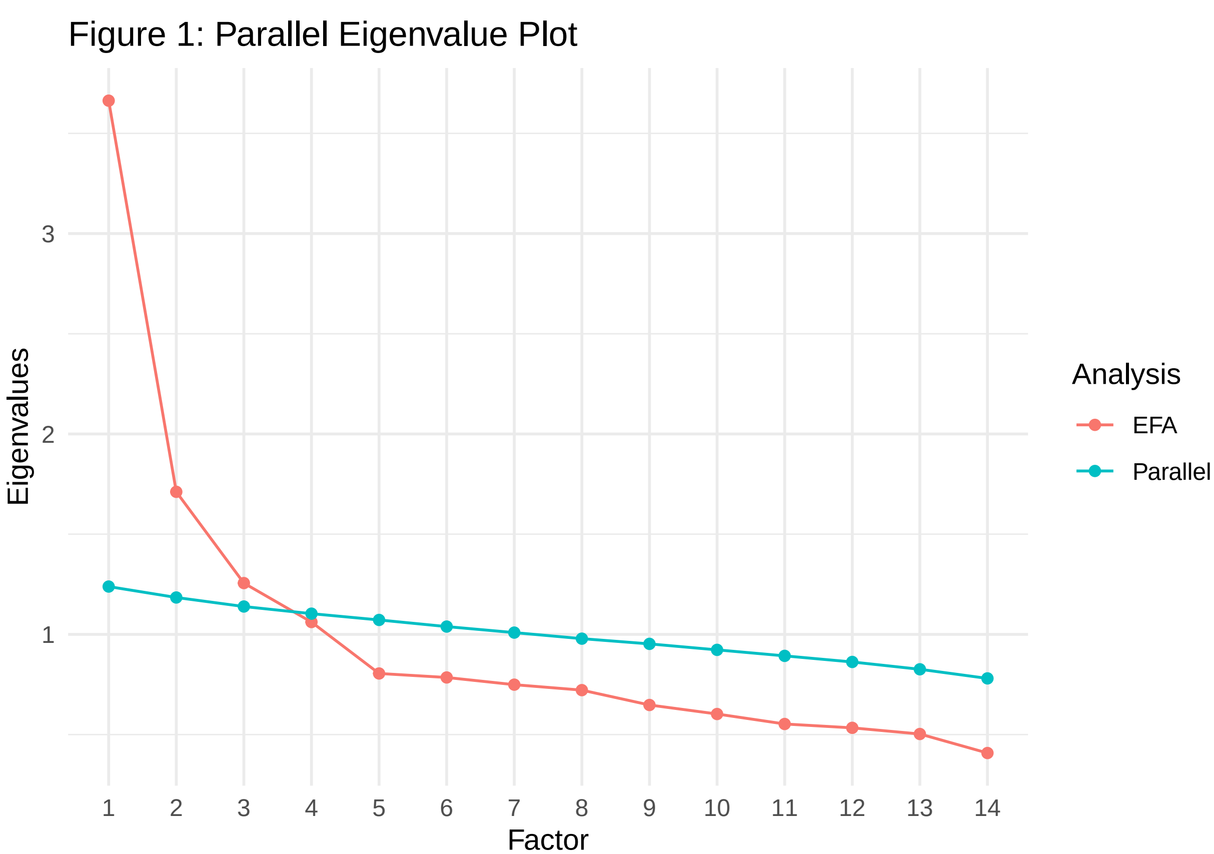
```
  mutate(Factor = fct_inorder(Factor))
```

Pivot the dataframe to “long” format

```
plot_data_long <- plot_data %>%
  pivot_longer(EFA:Parallel,                    # The columns I'm gathering together
               names_to = "Analysis",       # new column name for existing names
               values_to = "Eigenvalues")   # new column name to store values
```

Plot using ggplot

```
plot_data_long %>%
ggplot(aes(y=Eigenvalues,
           x=Factor,
           group=Analysis,
           color=Analysis)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(title = "Figure 1: Parallel Eigenvalue Plot")
```



save figure to the designated folder

```r
ggsave(here("03-efa","figures","eigenvalue_elbow_rplot.png"),
       dpi=300, height=5, width=7, units="in")
```

Create table of EFA loading estimates

```r
loadings_stdyx <- efa_1_fit$results$parameters$efa$f3$loadings$estimates %>%
   as.data.frame() %>%
  rownames_to_column("Names") %>%
  mutate(Names = str_to_lower(Names))

loadings_stdyx %>%
  gt() %>%
  tab_header(
    title = md("**Table 2**"),
    subtitle = md("*Summary of Factor Loadings: 3-Factor EFA Model*")) %>%
  cols_label('1' = "F1", '2' = "F2", '3' = "F3") %>%
  tab_row_group(group = "Factor 1: School Climate", rows = 1:5) %>%
  tab_row_group(group = "Factor 2: Safety", rows = 6:10) %>%
  tab_row_group(group = "Factor 3: Clear Rules", rows = 11:14) %>%
  row_group_order(groups = c("Factor 1: School Climate","Factor 2: Safety","Factor 3: Clear Rules")) %>%
  tab_style(style = list(cell_text(weight = "bold")),
            locations = cells_body(columns = "1",rows = 1:5)) %>%
  tab_style(style = list(cell_text(weight = "bold")),
            locations = cells_body(columns = "2",rows = 6:10)) %>%
  tab_style(style = list(cell_text(weight = "bold")),
            locations = cells_body(columns = "3",rows = 11:14)) %>%
  tab_options(column_labels.font.weight = "bold")
```

Table 2

Summary of Factor Loadings: 3-Factor EFA Model

Names	F1	F2	F3
Factor 1: School Climate			
stu_tch	0.476	0.142	-0.095
sc_spirt	0.358	0.083	0.103
tch_good	0.744	-0.004	-0.011
tch_intr	0.771	0.004	0.013
tch_prai	0.563	-0.045	0.125
Factor 2: Safety			
stu_dwn	0.012	0.256	0.132
not_safe	0.169	0.526	0.010
disr_lrn	-0.025	0.390	0.089
gangs	0.043	0.607	-0.004
rac_fght	-0.040	0.649	-0.012
Factor 3: Clear Rules			
sch_rule	0.098	0.099	0.460
pun_same	0.213	0.023	0.450
strict	0.179	-0.015	0.376
pun_rule	-0.011	-0.090	0.762

EFA with non-correlated factors

Model 2: Varimax orthoginal rotation

```
efa_2  <- mplusObject(
  TITLE = "EFA",
  VARIABLE =
 "usevar =
  stu_tch sc_spirt tch_good tch_intr
  tch_prai stu_dwn not_safe disr_lrn
  gangs rac_fght sch_rule pun_same strict pun_rule;",

  ANALYSIS =
 "type = efa 1 5;
  estimator = MLR;
  rotation = varimax;  !!! orthogonal (no factor correlations) !!!
  ",

  MODEL = "" ,

  PLOT = "type = plot3;",

  OUTPUT = "sampstat;",

  usevariables = colnames(efa_data),
  rdata = efa_data)

efa_2_fit <- mplusModeler(efa_2,
             dataout=here("03-efa","efa_mplus","m2_efa_els.dat"),
             modelout=here("03-efa","efa_mplus","m2_efa_els.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

Alternate syntax: use `update()` to alter the `mplusObject()` named `efa_1`

- `tilde` (~) will replace everything in that section of the input.
- `tilde-dot-plus` (~.+) will update the section by adding the specified code into that section

```
efa_2  <- update(efa_1,
                 ANALYSIS = ~.+ "rotation = varimax;")

efa_2_fit <- mplusModeler(efa_2,
             dataout=here("03-efa","efa_mplus","m2_efa_els.dat"),
             modelout=here("03-efa","efa_mplus","m2_efa_els.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

Applied example 2: EFA with Categorical indicators

Weighted Least Spuares Estimator (WLS)

DATA SOURCE: This lab exercise utilizes a subset of the HSLS public-use dataset: High School Longitudinal Study of 2009 (Ingels et al., 2011) See website: nces.ed.gov

```
data_raw <- read_csv("https://garberadamc.github.io/project-site/data/hsls_fa_data_subset.csv")
```

Reverse code for factor interpretation

```
data_raw1 <- data_raw

cols = c("S1MPERS1", "S1MPERS2", "S1MUSELI", "S1MUSECL", "S1MUSEJO",
         "S1MTESTS", "S1MTEXTB", "S1MSKILL", "S1MASSEX", "S1MENJNG",
         "S1SPERS1", "S1SPERS2", "S1SUSELI", "S1SUSECL", "S1SUSEJO",
         "S1STESTS", "S1STEXTB", "S1SSKILL", "S1SASSEX", "S1SENJNG")

data_raw1[ ,cols] <-  5 - data_raw1[ ,cols]
```

Prepare `data.frame` for analysis (select & rename columns)

```
hsls_data <- data_raw1 %>%
  select(
    'mth_pers'="S1MPERS1", 'mth_othr'="S1MPERS2", 'mth_life'="S1MUSELI",
    'mth_cllg'="S1MUSECL", 'mth_futr'="S1MUSEJO", 'mth_tsts'="S1MTESTS",
    'mth_text'="S1MTEXTB", 'mth_mstr'="S1MSKILL", 'mth_asgn'="S1MASSEX",
    'mth_enjy'="S1MENJNG", 'sci_pers'="S1SPERS1", 'sci_othr'="S1SPERS2",
    'sci_life'="S1SUSELI", 'sci_cllg'="S1SUSECL", 'sci_futr'="S1SUSEJO",
    'sci_tsts'="S1STESTS", 'sci_text'="S1STEXTB", 'sci_mstr'="S1SSKILL",
    'sci_asgn'="S1SASSEX", 'sci_enjy'="S1SENJNG")
```

Look at variables for EFA example with categorical indicators

Applied Example: Math & Science Utility[1]

Name	Variable Description
Math Indicators	
mth_pers	9th grader sees himself/herself as a math person
mth_othr	Others see 9th grader as a math person
mth_life	9th grader thinks fall 2009 math course is useful for everyday life
mth_cllg	9th grader thinks fall 2009 math course will be useful for college
mth_futr	9th grader thinks fall 2009 math course is useful for future career
mth_tsts	9th grader confident can do excellent job on fall 2009 math tests
mth_text	9th grader certain can understand fall 2009 math textbook

mth_mstr	9th grader certain can master skills in fall 2009 math course
mth_asgn	9th grader confident can do excellent job on fall 2009 math assignments
mth_enjy	9th grader is enjoying fall 2009 math course very much
Science Indicators	
sci_pers	9th grader sees himself/herself as a science person
sci_othr	Others see 9th grader as a science person
sci_life	9th grader thinks fall 2009 science course is useful for everyday life
sci_cllg	9th grader thinks fall 2009 science course will be useful for college
sci_futr	9th grader thinks fall 2009 science course is useful for future career
sci_tsts	9th grader confident can do excellent job on fall 2009 science tests
sci_text	9th grader certain can understand fall 2009 science textbook
sci_mstr	9th grader certain can master skills in fall 2009 science course
sci_asgn	9th grader confident can do excellent job on fall 2009 science assignments
sci_enjy	9th grader is enjoying fall 2009 science course very much

[1]Note. All scale indicators have 4-point Likert response options ranging from Strongly Agree (1) to Strongly Disagree (4).

Model 0 - Exploratory Factor Analysis (EFA) with WLS Estimator

---

```
efa_wls  <- mplusObject(

  TITLE =
    "EFA with Categorical Indicators - HSLS",

  VARIABLE =
    "usevar = mth_pers-sci_enjy;

     categorical = mth_pers-sci_enjy;",

  ANALYSIS =
    "type = efa 1 7;
     estimator=wlsmv;",

  MODEL = "" ,

  PLOT = "type = plot3;",

  OUTPUT = "sampstat;",

  usevariables = colnames(hsls_data),
  rdata = hsls_data)

efa_wls_fit <- mplusModeler(efa_wls,
                dataout=here("03-efa", "wls_efa", "efa_sci_HSLS_wls.dat"),
                modelout=here("03-efa", "wls_efa", "efa_sci_HSLS_wls.inp"),
                check=TRUE, run = TRUE, hashfilename = FALSE)
```

Create table summarizing model fit

```
model_fit <- LatexSummaryTable(efa_wls_fit,
  keepCols=c("Title","Parameters", "LL",
             "ChiSqM_Value","ChiSqM_DF", "ChiSqM_PValue",
             "RMSEA_Estimate","RMSEA_90CI_LB","RMSEA_90CI_UB",
             "CFI", "TLI","SRMR")) %>%
  mutate(Title = c("1-Factor","2-Factor","3-Factor",
                   "4-Factor","5-Factor","6-Factor","7-Factor")) %>%
  mutate_at(vars(contains("RMSEA")), ~format(., nsmall = 3)) %>%
  unite(CI, RMSEA_90CI_LB:RMSEA_90CI_UB, sep=", ", remove = TRUE) %>%
  mutate(CI = paste0("(", CI, ")")) %>%
  unite(RMSEA, RMSEA_Estimate:CI, sep=" ", remove = TRUE)

model_fit %>%
  gt() %>%
  tab_header(
    title = md("**Table 1**"),
    subtitle = md("*Summary of Model Fit Indices*")) %>%
  cols_label(
    Title = "Model",
    Parameters = md("Par"),
    #LL = md("*LL*"),
    ChiSqM_Value = md("Chi^2"),
    ChiSqM_PValue = md("*p-value*"),
    ChiSqM_DF = md("*df*"),
    RMSEA = "RMSEA (90% CI)" ) %>%
  tab_options(column_labels.font.weight = "bold") %>%
  fmt(c(5), fns = function(x) ifelse(x<0.001,"<.001",
                                     scales::number(x, accuracy = 0.01)))
```

Table 1

Summary of Model Fit Indices

Model	Par	Chi^2	*df*	*p-value*	RMSEA (90% CI)	CFI	TLI	SRMR
1-Factor	20	17511.054	170	<.001	0.193 (0.190, 0.195)	0.714	0.681	0.201
2-Factor	39	11510.679	151	<.001	0.166 (0.163, 0.168)	0.813	0.764	0.118
3-Factor	57	6878.199	133	<.001	0.136 (0.133, 0.139)	0.889	0.841	0.077
4-Factor	74	2858.925	116	<.001	0.093 (0.090, 0.096)	0.955	0.926	0.046
5-Factor	90	2014.418	100	<.001	0.084 (0.080, 0.087)	0.968	0.940	0.035
6-Factor	105	1282.979	85	<.001	0.072 (0.068, 0.075)	0.980	0.956	0.025
7-Factor	119	875.376	71	<.001	0.064 (0.060, 0.068)	0.987	0.965	0.019

Create table of EFA loading estimates

```
loadings_stdyx <- efa_wls_fit$results$parameters$efa$f2$loadings$estimates %>%
   as.data.frame() %>%
  rownames_to_column("Names") %>%
  mutate(Names = str_to_lower(Names))
```

```r
loadings_stdyx %>%
  gt() %>%
  tab_header(
    title = md("**Table 2**"),
    subtitle = md("*Summary of Factor Loadings: 2-Factor EFA Model*")) %>%
  cols_label('1' = "F1", '2' = "F2") %>%
  tab_row_group(group = "Factor 1: Math Indicators", rows = 1:10) %>%
  tab_row_group(group = "Factor 2: Science Indicators", rows = 11:20) %>%
  row_group_order(groups = c("Factor 1: Math Indicators",
                             "Factor 2: Science Indicators")) %>%
  tab_style(style = list(cell_text(weight = "bold")),
            locations = cells_body(columns = "1",rows = 1:10)) %>%
  tab_style(style = list(cell_text(weight = "bold")),
            locations = cells_body(columns = "2",rows = 11:20)) %>%
  tab_options(column_labels.font.weight = "bold")
```

Table 2

Summary of Factor Loadings: 2-Factor EFA Model

Names	F1	F2
Factor 1: Math Indicators		
mth_pers	0.806	-0.011
mth_othr	0.739	0.079
mth_life	0.502	0.041
mth_cllg	0.548	0.145
mth_futr	0.551	0.141
mth_tsts	0.888	-0.045
mth_text	0.793	0.003
mth_mstr	0.847	0.038
mth_asgn	0.878	-0.008
mth_enjy	0.635	-0.004
Factor 2: Science Indicators		
sci_pers	-0.204	0.895
sci_othr	-0.134	0.823
sci_life	0.081	0.479
sci_cllg	0.146	0.535
sci_futr	0.072	0.557
sci_tsts	-0.032	0.854
sci_text	0.025	0.764
sci_mstr	0.040	0.837
sci_asgn	0.003	0.859
sci_enjy	0.009	0.617

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.

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/
