Latent Class Analysis Enumeration

IMMERSE Training Team

Updated: June 07, 2023

Contents

IMMERSE Project

The Institute of Mixture Modeling for Equity-Oriented Researchers, Scholars, and Educators (IMMERSE) is an IES funded training grant (R305B220021) to support Education scholars in integrating mixture modeling into their research.

- Please visit our website to learn more and apply for the year-long fellowship.
- Follow us on Twitter!

How to reference this walkthrough: *This work was supported by the IMMERSE Project* (IES - 305B220021)

Visit our GitHub account to download the materials needed for this walkthrough.

Example: Bullying in Schools

To demonstrate mixture modeling in the training program and online resource components of the IES grant we utilize the *Civil Rights Data Collection (CRDC)* (CRDC) data repository. The CRDC is a federally mandated school-level data collection effort that occurs every other year. This public data is currently available for selected latent class indicators across 4 years (2011, 2013, 2015, 2017) and all US states. In this example, we use the Arizona state sample. We utilize six focal indicators which constitute the latent class model in our example; three variables which report on harassment/bullying in schools based on disability, race, or sex, and three variables on full-time equivalent school staff hires (counselor, psychologist, law enforcement). This data source also includes covariates on a variety of subjects and distal outcomes reported in 2018 such as math/reading assessments and graduation rates.

Load packages

```
library(tidyverse)
library(haven)
library(glue)
library(MplusAutomation)
library(here)
library(janitor)
library(gt)
library(cowplot)
library(DiagrammeR)
here::i_am("lca_enum.Rmd")
```

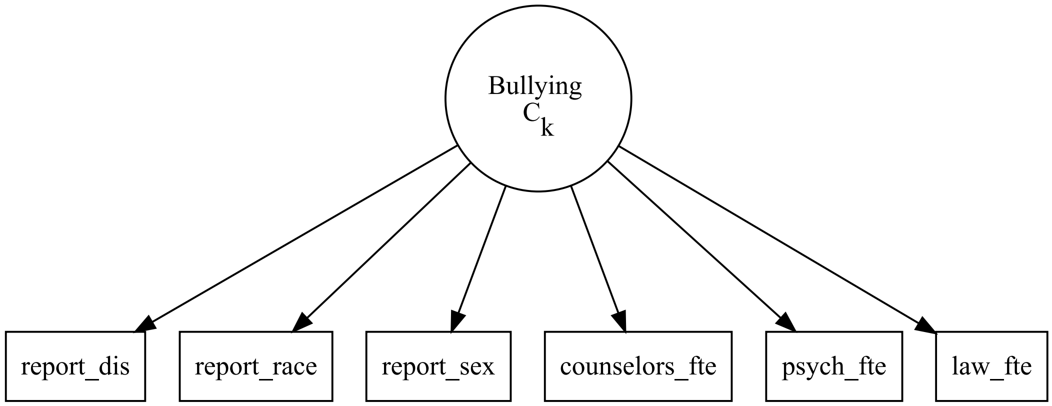
Variable Description

LCA indicators[1]		
Name	Label	Values
leaid	District Identification Code	
ncessch	School Identification Code	
report_dis	Number of students harassed or bullied on the basis of disability	0 = No reported incidents, 1 = At least one reported incident
report_race	Number of students harassed or bullied on the basis of race, color, or national origin	0 = No reported incidents, 1 = At least one reported incident
report_sex	Number of students harassed or bullied on the basis of sex	0 = No reported incidents, 1 = At least one reported incident
counselors_fte	Number of full time equivalent counselors hired as school staff	0 = No staff present, 1 = At least one staff present
report_sex	Number of full time equivalent psychologists hired as school staff	0 = No staff present, 1 = At least one staff present
counselors_fte	Number of full time equivalent law enforcement officers hired as school staff	0 = No staff present, 1 = At least one staff present

[1] Civil Rights Data Collection (CRDC)

Variables have been transformed to be dichotomous indicators using the following coding strategy

Harassment and bullying count variables are recoded `1` if the school reported at least one incident of harassment (`0` indicates no reported incidents). On the original scale reported by the CDRC staff variables for full time equivalent employees (FTE) are represented as `1` and part time employees are represented by values between `1` and `0`. Schools with greater than one staff of the designated type are represented by values greater than 1. All values greater than zero were recorded as `1s` (e.g., `.5`, `1`,`3`) indicating that the school has a staff present on campus at least part time. Schools with no staff of the designated type are indicated as `0` for the dichotomous variable.


Bullying
C_k
report_dis
report_race
report_sex
counselors_fte
psych_fte
law_fte

Prepare Data

```
df_bully <- read_csv(here("data", "crdc_lca_data.csv")) %>%
  clean_names() %>%
  dplyr::select(report_dis, report_race, report_sex, counselors_fte, psych_fte, law_fte)
```

Descriptive Statistics

```
# Set up data to find proportions of binary indicators
ds <- df_bully %>%
  pivot_longer(c(report_dis, report_race, report_sex, counselors_fte, psych_fte, law_fte), names_to = "V

# Create table of variables and counts
tab <- table(ds$Variable, ds$value)

# Find proportions and round to 3 decimal places
prop <- prop.table(tab, margin = 1) %>%
  round(3)

# Combine everything to one table
dframe <- data.frame(Variables=rownames(tab), Proportion=prop[,2], Count=tab[,2])
#remove row names
row.names(dframe) <- NULL

# Make it a gt() table
prop_table <- dframe %>%
  gt()
prop_table
```

Variables	Proportion	Count
counselors_fte	0.460	919
law_fte	0.126	251
psych_fte	0.474	947
report_dis	0.043	85
report_race	0.103	206
report_sex	0.170	340

```
# Save as img
gtsave(prop_table, here("figures", "prop_table.png"))
```

Enumeration

This code uses the `mplusObject` function in the `MplusAutomation` package.

```r
lca_6  <- lapply(1:6, function(k) {
  lca_enum  <- mplusObject(

    TITLE = glue("{k}-Class"),

    VARIABLE = glue(
    "categorical = report_dis-law_fte;
     usevar = report_dis-law_fte;
     classes = c({k}); "),

  ANALYSIS =
   "estimator = mlr;
    type = mixture;
    starts = 200 100;
    processors = 10;",

  OUTPUT = "sampstat residual tech11 tech14;",

  PLOT =
    "type = plot3;
    series = report_dis-law_fte(*);",

  usevariables = colnames(df_bully),
  rdata = df_bully)

lca_enum_fit <- mplusModeler(lca_enum,
                            dataout=glue(here("enum", "bully.dat")),
                            modelout=glue(here("enum", "c{k}_bully.inp")) ,
                            check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

Table of Fit

First, extract data:

```r
#
output_bully <- readModels(here("enum"), filefilter = "bully", quiet = TRUE)

enum_extract <- LatexSummaryTable(
  output_bully,
  keepCols = c(
    "Title",
    "Parameters",
    "LL",
    "BIC",
    "aBIC",
    "BLRT_PValue",
    "T11_VLMR_PValue",
    "Observations"
  ),
```

```
  sortBy = "Title"
)


allFit <- enum_extract %>%
  mutate(CAIC = -2 * LL + Parameters * (log(Observations) + 1)) %>%
  mutate(AWE = -2 * LL + 2 * Parameters * (log(Observations) + 1.5)) %>%
  mutate(SIC = -.5 * BIC) %>%
  mutate(expSIC = exp(SIC - max(SIC))) %>%
  mutate(BF = exp(SIC - lead(SIC))) %>%
  mutate(cmPk = expSIC / sum(expSIC)) %>%
  dplyr::select(1:5, 9:10, 6:7, 13, 14) %>%
  arrange(Parameters)
```

Then, create table:

```
fit_table1 <- allFit %>%
  gt() %>%
  tab_header(title = md("**Model Fit Summary Table**")) %>%
  cols_label(
    Title = "Classes",
    Parameters = md("Par"),
    LL = md("*LL*"),
    T11_VLMR_PValue = "VLMR",
    BLRT_PValue = "BLRT",
    BF = md("BF"),
    cmPk = md("*cmPk*")
  ) %>%
  tab_footnote(
    footnote = md(
      "*Note.* Par = Parameters; *LL* = model log likelihood;
BIC = Bayesian information criterion;
aBIC = sample size adjusted BIC; CAIC = consistent Akaike information criterion;
AWE = approximate weight of evidence criterion;
BLRT = bootstrapped likelihood ratio test p-value;
VLMR = Vuong-Lo-Mendell-Rubin adjusted likelihood ratio test p-value;
*cmPk* = approximate correct model probability."
    ),
locations = cells_title()
  ) %>%
  tab_options(column_labels.font.weight = "bold") %>%
  fmt_number(c(3:7),
             decimals = 2) %>%
  sub_missing(1:11,
              missing_text = "--") %>%
  fmt(
    c(8:9, 11),
    fns = function(x)
      ifelse(x < 0.001, "<.001",
             scales::number(x, accuracy = .01))
  ) %>%
  fmt(
    10,
```

```
    fns = function (x)
      ifelse(x > 100, ">100",
             scales::number(x, accuracy = .01))
  ) %>%
  tab_style(
    style = list(
      cell_text(weight = "bold")
      ),
    locations = list(cells_body(
     columns = BIC,
     row = BIC == min(BIC[c(1:6)]) # Change this to the number of classes you estimated
    ),
    cells_body(
     columns = aBIC,
     row = aBIC == min(aBIC[1:6])
    ),
    cells_body(
     columns = CAIC,
     row = CAIC == min(CAIC[1:6])
    ),
    cells_body(
     columns = AWE,
     row = AWE == min(AWE[1:6])
    ),
    cells_body(
     columns = cmPk,
     row =  cmPk == max(cmPk[1:6])
     ),
    cells_body(
     columns = BF,
     row =  BF > 10),
    cells_body(
     columns =  T11_VLMR_PValue,
     row =  T11_VLMR_PValue < .001),
    cells_body(
     columns =  BLRT_PValue,
     row =  BLRT_PValue < .001)
  )
)

fit_table1
```

Model Fit Summary Table[1]

Classes	Par	*LL*	BIC	aBIC	CAIC	AWE	BLRT	VLMR	BF	*cmPk*
1-Class	6	$-5,443.41$	$10,932.50$	$10,913.44$	$10,938.50$	$10,996.19$	–	–	0.00	<.001
2-Class	13	$-5,194.14$	$10,487.26$	$10,445.96$	$10,500.26$	$10,625.24$	<.001	<.001	0.00	<.001
3-Class	20	$-5,122.48$	$10,397.24$	$10,333.70$	$10,417.24$	$10,609.53$	<.001	<.001	>100	1.00
4-Class	27	$-5,111.76$	$10,429.10$	$10,343.32$	$10,456.10$	$10,715.69$	<.001	0.01	>100	<.001
5-Class	34	$-5,105.59$	$10,470.07$	$10,362.04$	$10,504.06$	$10,830.95$	0.29	0.18	>100	<.001
6-Class	41	$-5,099.88$	$10,511.95$	$10,381.69$	$10,552.95$	$10,947.14$	0.38	0.18	–	<.001

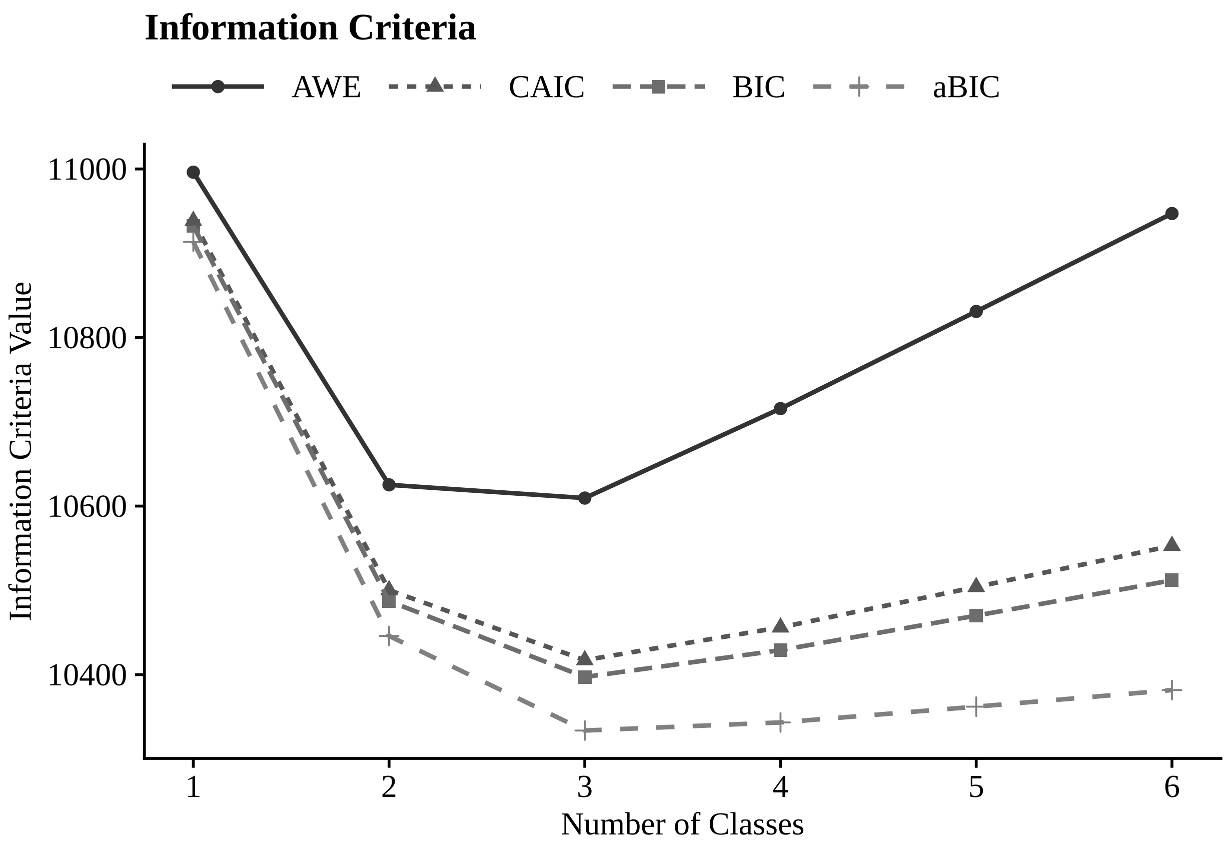
[1] *Note.* Par = Parameters; *LL* = model log likelihood; BIC = Bayesian information criterion; aBIC = sample size adjusted BIC; CAIC = consistent Akaike information criterion; AWE = approximate weight of evidence criterion; BLRT = bootstrapped likelihood ratio test p-value; VLMR = Vuong-Lo-Mendell-Rubin adjusted likelihood ratio test p-value; *cmPk* = approximate correct model probability.

Save table:

```
gtsave(fit_table1, here("figures", "fit_table1.png"))
```

Information Criteria Plot

```
allFit %>%
  dplyr::select(2:7) %>%
  rowid_to_column() %>%
  pivot_longer(`BIC`:`AWE`,
               names_to = "Index",
               values_to = "ic_value") %>%
  mutate(Index = factor(Index,
                        levels = c ("AWE", "CAIC", "BIC", "aBIC"))) %>%
  ggplot(aes(
    x = rowid,
    y = ic_value,
    color = Index,
    shape = Index,
    group = Index,
    lty = Index
  )) +
  geom_point(size = 2.0) + geom_line(size = .8) +
  scale_x_continuous(breaks = 1:nrow(allFit)) +
  scale_colour_grey(end = .5) +
  theme_cowplot() +
  labs(x = "Number of Classes", y = "Information Criteria Value", title = "Information Criteria") +
  theme(
    text = element_text(family = "serif", size = 12),
    legend.text = element_text(family="serif", size=12),
    legend.key.width = unit(3, "line"),
    legend.title = element_blank(),
    legend.position = "top"
  )
```



Save figure:

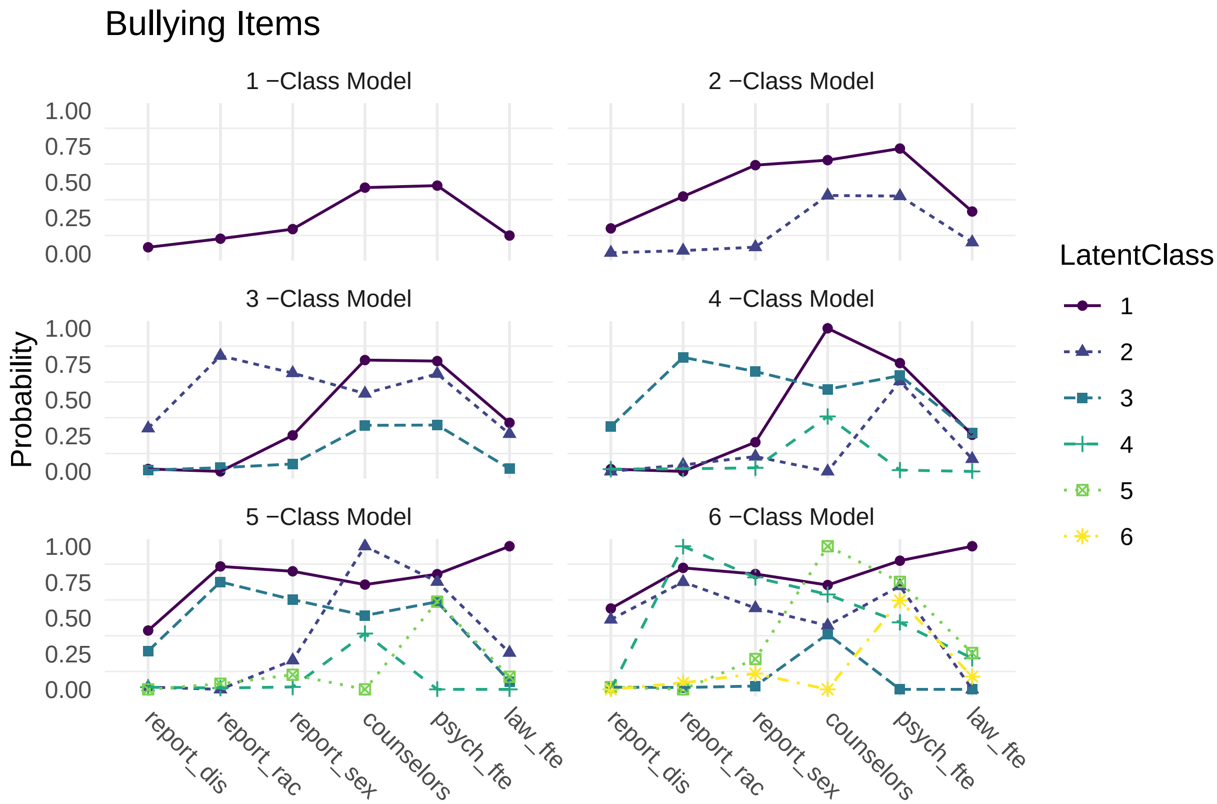
```
ggsave(here("figures", "info_criteria.png"), dpi=300, height=5, width=7, units="in")
```

Compare Class Solutions

Compare probability plots for $K = 1:6$ class solutions

```
model_results <- data.frame()

for (i in 1:length(output_bully)) {

  temp <- output_bully[[i]]$parameters$probability.scale %>%
    mutate(model = paste(i,"-Class Model"))

  model_results <- rbind(model_results, temp)
}

rm(temp)
```

```r
compare_plot <-
  model_results %>%
  filter(category == 2) %>%
  dplyr::select(est, model, LatentClass, param) %>%
  mutate(param = as.factor(str_to_lower(param)))

compare_plot$param <- fct_inorder(compare_plot$param)

ggplot(
  compare_plot,
  aes(
    x = param,
    y = est,
    color = LatentClass,
    shape = LatentClass,
    group = LatentClass,
    lty = LatentClass
  )
) +
  geom_point() +
  geom_line() +
  scale_colour_viridis_d() +
  facet_wrap( ~ model, ncol = 2) +
  labs(title = "Bullying Items",
       x = " ", y = "Probability") +
  theme_minimal() +
  theme(panel.grid.major.y = element_blank(),
                          axis.text.x = element_text(angle = -45, hjust = -.1))
```

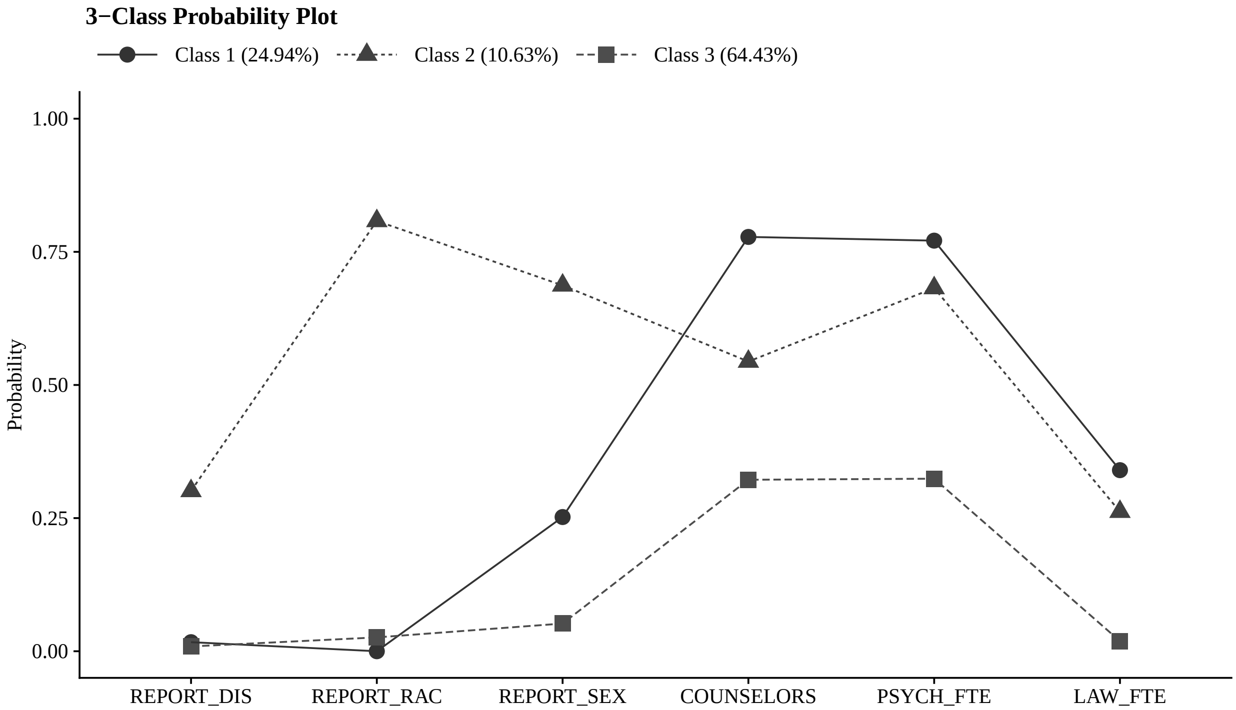


Save figure:

```
ggsave(here("figures", "compare_kclass_plot.png"), dpi=300, height=5, width=7, units="in")
```

3-Class Probability Plot

```
source("plot_lca_function.txt")

plot_lca_function(model_name = output_bully$c3_bully.out)
```



Save figure:

```
ggsave(here("figures", "C3_bully_LCA_Plot.png"), dpi="retina", height=5, width=7, units="in")
```

Observed Response Patterns

Save response frequencies for the 3-class model from the previous lab with `response is _____.dat` under `SAVEDATA`.

```
patterns  <- mplusObject(

  TITLE = "C3 LCA - Save response patterns",

  VARIABLE =
  "categorical = report_dis-law_fte;
   usevar =  report_dis-law_fte;
   classes = c(3);",

  ANALYSIS =
   "estimator = mlr;
    type = mixture;
    starts = 200 100;
    processors = 10;",
```

```
  SAVEDATA =
   "File=savedata.dat;
    Save=cprob;

    ! Code to save response frequency data

    response is resp_patterns.dat;",

  OUTPUT = "residual patterns tech10 tech11 tech14",

  usevariables = colnames(df_bully),
  rdata = df_bully)

patterns_fit <- mplusModeler(patterns,
                dataout=here("enum", "bully.dat"),
                modelout=here("enum", "patterns.inp") ,
                check=TRUE, run = TRUE, hashfilename = FALSE)
```

```
## <simpleError in bivarFitData[mPos, ] <- c(vars, values): number of items to replace is not a multiple
```

Note: You may see an error that says `<simpleError in bivarFitData[mPos, ] <- c(vars, values): number of items to replace is not a multiple of replacement length>`, the developers are aware of this and are working to fix it.

Read in observed response pattern data and relabel the columns

```
# Read in response frequency data that we just created:
patterns <- read_table(here("enum", "resp_patterns.dat"),
                        col_names=FALSE, na = "*")

# Extract the column names
names <- names(readModels(here("enum", "patterns.out"))[['savedata']])
```

```
## <simpleError in bivarFitData[mPos, ] <- c(vars, values): number of items to replace is not a multiple
```

```
# Add the names back to the dataset
colnames(patterns) <- c("Frequency", names)
```

Create a table with the top 5 unconditional response pattern, then top of conditional response pattern for each modal class assignment

```
# Order responses by highest frequency
order_highest <- patterns %>%
  arrange(desc(Frequency))

# Loop `patterns` data to list top 5 conditional response patterns for each class
loop_cond  <- lapply(1:max(patterns$C), function(k) {
order_cond <- patterns %>%
```

```r
  filter(C == k) %>%
  arrange(desc(Frequency)) %>%
  head(5)
  })

# Convert loop into data frame
table_data <- as.data.frame(bind_rows(loop_cond))

# Combine unconditional and conditional responses patterns
response_patterns <-  rbind(order_highest[1:5,], table_data)
```

Finally, use {gt} to make a nicely formatted table

```r
response_patterns %>%
  gt() %>%
    tab_header(
    title = "Observed Response Patterns",
    subtitle = html("Response patterns, estimated frequencies, estimated posterior class probabilities a
    tab_source_note(
    source_note = md("Data Source: **Civil Rights Data Collection (CRDC)**")) %>%
    cols_label(
      Frequency = html("<i>f</i><sub>r</sub>"),
    REPORT_D = "Harrassment: Disability",
    REPORT_R = "Harrassment: Race",
    REPORT_S = "Harrassment: Sex",
    COUNSELO = "Staff: Counselor",
    PSYCH_FT = "Staff: Psychologist",
    LAW_FTE = "Staff: Law Enforcement",
    CPROB1 = html("P<sub><i>k</i></sub>=1"),
    CPROB2 = html("P<sub><i>k</i></sub>=2"),
    CPROB3 = html("P<sub><i>k</i></sub>=3"),
    C = md("*k*")) %>%
  tab_row_group(
    label = "Unconditional response patterns",
    rows = 1:5) %>%
  tab_row_group(
    label = md("*k* = 1 Conditional response patterns"),
    rows = 6:10) %>% #EDIT THESE VALUES BASED ON THE LAST COLUMN
  tab_row_group(
    label = md("*k* = 2 Conditional response patterns"),
    rows = 11:15)  %>% #EDIT THESE VALUES BASED ON THE LAST COLUMN
  tab_row_group(
    label = md("*k* = 3 Conditional response patterns"),
    rows = 16:20)  %>% #EDIT THESE VALUES BASED ON THE LAST COLUMN
    row_group_order(
      groups = c("Unconditional response patterns",
                 md("*k* = 1 Conditional response patterns"),
                 md("*k* = 2 Conditional response patterns"),
                 md("*k* = 3 Conditional response patterns"))) %>%
    tab_footnote(
    footnote = html(
      "<i>Note.</i> <i>f</i><sub>r</sub> = response pattern frequency; P<sub><i>k</i></sub> = posterior
    )
```

```
) %>%
cols_align(align = "center") %>%
opt_align_table_header(align = "left") %>%
gt::tab_options(table.font.names = "Times New Roman") %>%
gtsave("figures/response_patterns.png")
```

Observed Response Patterns

Response patterns, estimated frequencies, estimated posterior class probabilities and modal assignments

f_r	Harrassment: Disability	Harrassment: Race	Harrassment: Sex	Staff: Counselor	Staff: Psychologist	Staff: Law Enforcement	P_k=1	P_k=2	P_k=3
Unconditional response patterns									
525	0	0	0	0	0	0	0.023	0.002	0.976
299	0	0	0	0	1	0	0.139	0.007	0.854
293	0	0	0	1	0	0	0.146	0.004	0.850
251	0	0	0	1	1	0	0.541	0.009	0.449
75	0	0	0	1	1	1	0.959	0.011	0.030
k = 1 Conditional response patterns									
251	0	0	0	1	1	0	0.541	0.009	0.449
75	0	0	0	1	1	1	0.959	0.011	0.030
72	0	0	1	1	1	0	0.803	0.088	0.108
38	0	0	1	0	1	0	0.431	0.139	0.430
34	0	0	0	0	1	1	0.789	0.027	0.184
k = 2 Conditional response patterns									
24	0	1	0	0	1	0	0.000	0.561	0.439
20	0	1	1	0	1	0	0.000	0.981	0.019
19	0	1	1	1	1	0	0.000	0.992	0.008
18	0	1	1	1	0	0	0.000	0.967	0.033
12	0	1	1	1	1	1	0.000	1.000	0.000
k = 3 Conditional response patterns									
525	0	0	0	0	0	0	0.023	0.002	0.976
299	0	0	0	0	1	0	0.139	0.007	0.854
293	0	0	0	1	0	0	0.146	0.004	0.850
36	0	0	1	0	0	0	0.117	0.060	0.823
27	0	0	0	NA	NA	NA	0.236	0.006	0.758

Note. f_r = response pattern frequency; P_k = posterior class probabilities

Data Source: **Civil Rights Data Collection (CRDC)**

Classification Diagnostics

Use Mplus to calculate k-class confidence intervals (Note: Change the synax to make your chosen *k*-class model):

```
classification  <- mplusObject(

  TITLE = "C3 LCA - Calculated k-Class 95% CI",

  VARIABLE =
    "categorical = report_dis-law_fte;
   usevar =  report_dis-law_fte;
   classes = c(3);",

  ANALYSIS =
    "estimator = ml;
    type = mixture;
    starts = 200 100;
    processors = 10;
    stseed = 802779;
    bootstrap = 1000;",

  MODEL =
    "
  !CHANGE THIS SECTION TO YOUR CHOSEN k-CLASS MODEL

  %OVERALL%
  [C#1](c1);

  [C#2](C2)

  Model Constraint:
  New(p1 p2 p3);

  p1 = exp(c1)/(1+exp(c1)+exp(c2));
  p2 = exp(c2)/(1+exp(c1)+exp(c2));
  p3 = 1/(1+exp(c1)+exp(c2));",


  OUTPUT = "sampstat tech11 tech14 cinterval(bcbootstrap)",

  usevariables = colnames(df_bully),
  rdata = df_bully)

classification_fit <- mplusModeler(classification,
                dataout=here("enum", "bully.dat"),
                modelout=here("enum", "class.inp") ,
                check=TRUE, run = TRUE, hashfilename = FALSE)
```

Read in the 3-class model:

```r
# Read in the 3-class model and extract information needed
output_bully <- readModels(here("enum", "class.out"))

# Entropy
entropy <- c(output_bully$summaries$Entropy, rep(NA, output_bully$summaries$NLatentClasses-1))

# 95% k-Class and k-class 95% Confidence Intervals
k_ci <- output_bully$parameters$ci.unstandardized %>%
  filter(paramHeader == "New.Additional.Parameters") %>%
  unite(CI, c(low2.5,up2.5), sep=", ", remove = TRUE) %>%
  mutate(CI = paste0("[", CI, "]")) %>%
  rename(kclass=est) %>%
  dplyr::select(kclass, CI)

# AvePPk = Average Latent Class Probabilities for Most Likely Latent Class Membership (Row) by Latent C
avePPk <- tibble(avePPk = diag(output_bully$class_counts$avgProbs.mostLikely))

# mcaPk = modal class assignment proportion
mcaPk <- round(output_bully$class_counts$mostLikely,3) %>%
  mutate(model = paste0("Class ", class)) %>%
  add_column(avePPk, k_ci) %>%
  rename(mcaPk = proportion) %>%
  dplyr::select(model, kclass, CI, mcaPk, avePPk)

# OCCk = odds of correct classification
OCCk <- mcaPk %>%
  mutate(OCCk = round((avePPk/(1-avePPk))/(kclass/(1-kclass)),3))

# Put everything together
class_table <- data.frame(OCCk, entropy)
```

Now, use {gt} to make a nicely formatted table

```r
class_table %>%
  gt() %>%
    tab_header(
    title = "Model Classification Diagnostics for the 3-Class Solution") %>%
    cols_label(
      model = md("*k*-Class"),
      kclass = md("*k*-Class Proportions"),
      CI = "95% CI",
      mcaPk = md("*mcaPk*"),
      avePPk = md("*AvePPk*"),
      OCCk = md("*OCCk*"),
      entropy = "Entropy") %>%
    sub_missing(7,
              missing_text = "") %>%
    tab_footnote(
    footnote = html(
      "<i>Note.</i> <i>f</i><sub>r</sub> = response pattern frequency; P<sub><i>k</i></sub> = posterior
    )
  ) %>%
  cols_align(align = "center") %>%
```

```
opt_align_table_header(align = "left") %>%
gt::tab_options(table.font.names = "Times New Roman")%>%
gtsave("figures/class_table.png")
```

Model Classification Diagnostics for the 3-Class Solution

k-Class	*k*-Class Proportions	95% CI	*mcaPk*	*AvePPk*	*OCCk*	Entropy
Class 1	0.249	[0.166, 0.329]	0.282	0.675	6.264	0.635
Class 2	0.106	[0.083, 0.136]	0.095	0.904	79.420	
Class 3	0.644	[0.561, 0.731]	0.623	0.893	4.614	

Note. f_r = response pattern frequency; P_k = posterior class probabilities

---

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, B. O., Muthén, L. K., & Asparouhov, T. (2017). Regression and mediation analysis using Mplus. Los Angeles, CA: Muthén & Muthén.

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

UC SANTA BARBARA