Introduction to Mixture Models - Latent Class Analysis (LCA)

A Course in `MplusAutomation`

Adam Garber

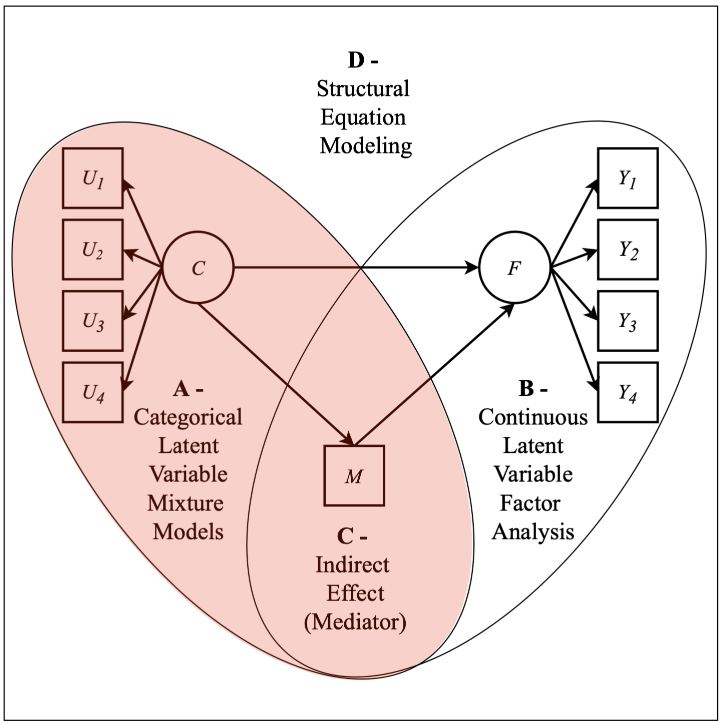


Figure. Picture has been adapted from study by Múthen, 2006.

Preparation

Data source:

1. The first example utilizes data on undergraduate *Cheating* available from the `poLCA` package (Dayton, 1998): See documentation here

2. The second examples utilizes the public-use data, *The Longitudinal Survey of American Youth* (**LSAY**): See documentation here

3. The third examples utilizes data from a study evaluating the *Kindergarten Student Entrance Profile* (**KSEP**) (Quirk et al., 2011): See documentation here

---

Load packages

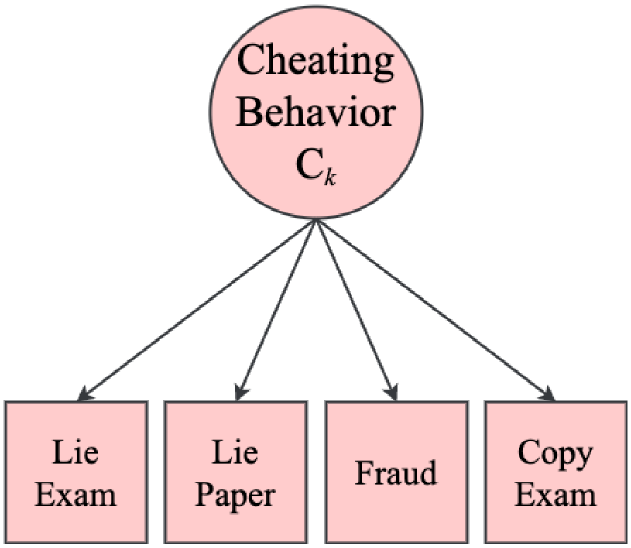
```
library(tidyverse)
library(haven)
library(glue)
library(MplusAutomation)
library(here)
library(janitor)
library(gt)
library(reshape2)
library(cowplot)
library(poLCA)
```

1.1 Enumeration

Estimate K-class models with 1 through 6 classes.

Example 1 - Undergraduate Cheating behavior

"Dichotomous self-report responses by 319 undergraduates to four questions about cheating behavior" (poLCA, 2016).



LCA indicators[1]

Name	Label	Values
LieExam	lied to avoid taking an exam	0 = No, 1 = Yes
LiePaper	lied to avoid handing a term paper in on time	0 = No, 1 = Yes
Fraud	purchased a term paper to hand in as their own or ...	0 = No, 1 = Yes
CopyExam	copied answers during an exam from someone sitting near to them	0 = No, 1 = Yes

[1]Undergraduate Cheating Behavior

Prepare data

```
data(cheating)

cheating <- cheating %>% clean_names()

df_cheat <-  cheating %>%
  dplyr::select(1:4) %>%
  dplyr::mutate_all(funs(.-1))
```

Estimate a quick LCA model

Specification details:

- Within the `lapply()` function `1:4` indicates the number of K-class models to estimate.
- The measurement model indicator names for this example are listed after the `categorical = ...;` & `usevar = ...;` statements.
- Note that in `Mplus` variables ordered as neighboring columns in the `data.frame` can be dplyr::selected by listing the first variable and last variable separated by a dash (e.g., `enjoy-adult`) .

```
lca_k1_4  <- lapply(1:4, function(k) {

  lca_enum  <- mplusObject(

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

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

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

  OUTPUT = "tech11 tech14;",

  PLOT =
    "type = plot3;
    series = lieexam-copyexam(*);",

  usevariables = colnames(df_cheat),
  rdata = df_cheat)

lca_enum_fit <- mplusModeler(lca_enum,
                            dataout=glue(here("12-intro-mixtures", "enum_cheat", "lca_cheat.dat")),
                            modelout=glue(here("12-intro-mixtures", "enum_cheat", "c{k}_lca_cheat.inp")
                            check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

1.2 Generate Model Fit Summary Table

- This syntax can be used to compare model fit from the series of LCA models generated during enumeration
- The code produces a table that is approximately in APA format.

Read in model fit statistics using `readModels()` and `mixtureSummaryTable()` functions

```
output_cheat <- readModels(here("12-intro-mixtures", "enum_cheat"), quiet = TRUE)

enum_summary <- LatexSummaryTable(output_cheat,
                keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC",
                           "BLRT_PValue", "T11_VLMR_PValue","Observations"),
                sortBy = "Title")
```

Calculate relevant fit indices for summary table

```
allFit <- enum_summary %>%
  mutate(aBIC = -2*LL+Parameters*log((Observations+2)/24)) %>%
  mutate(CIAC = -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)
```

Generate the fit summary table

```
allFit %>%
  mutate(Title = str_remove(Title, " LCA Enumeration - Cheating Behavior Example")) %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md(" ")) %>%
  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* = 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(10,decimals = 2,
             drop_trailing_zeros=TRUE,
             suffixing = TRUE) %>%
  fmt_number(c(3:9,11),
             decimals = 0) %>%
  fmt_missing(1:11,
              missing_text = "--") %>%
  fmt(c(8:9,11),
    fns = function(x)
    ifelse(x<0.001, "<.001",
           scales::number(x, accuracy = 0.01))) %>%
  fmt(10, fns = function(x)
    ifelse(x>100, ">100",
           scales::number(x, accuracy = .1)))
```

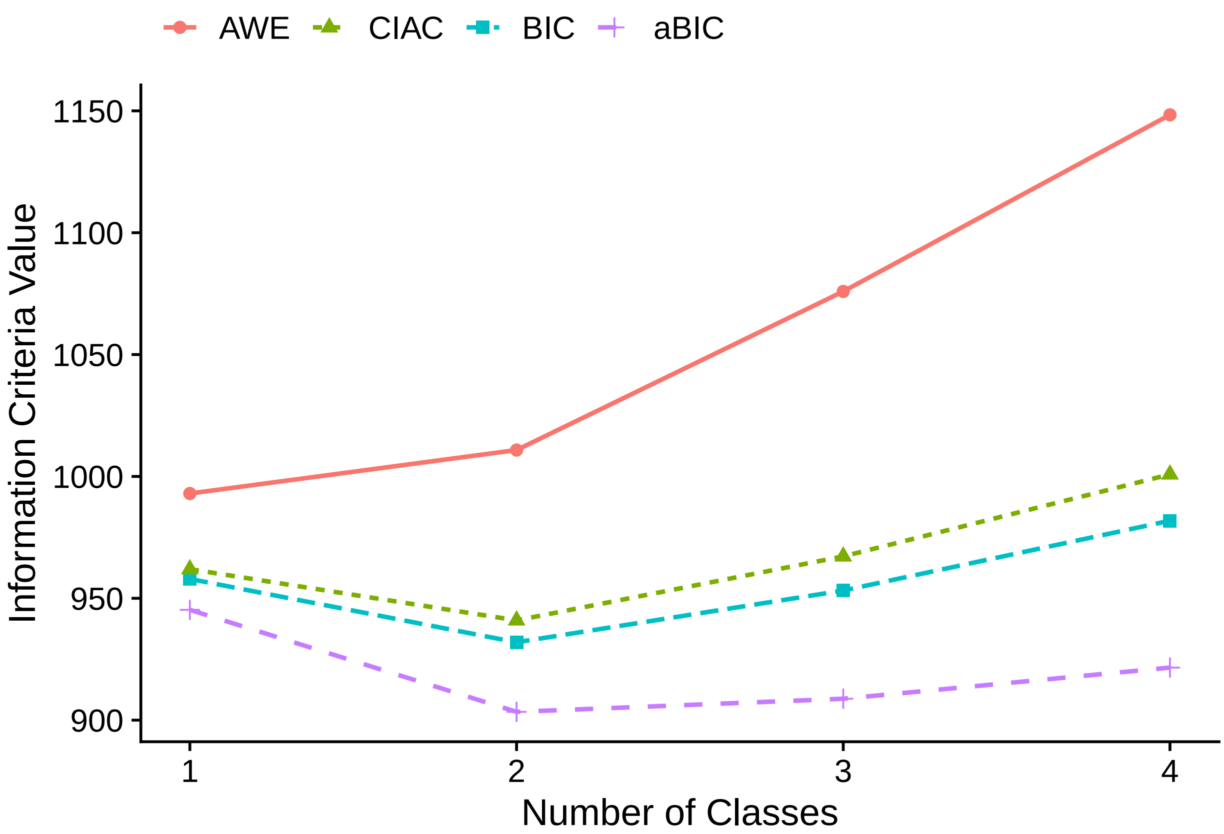
Model Fit Summary Table[1]

Classes	Par	*LL*	BIC	aBIC	CIAC	AWE	BLRT	VLMR	BF	*cmPk*
Class 1	4	−467	958	945	962	993	–	–	0.0	<.001
Class 2	9	−440	932	903	941	1,011	<.001	<.001	>100	1.00
Class 3	14	−436	953	909	967	1,076	0.14	0.17	>100	<.001
Class 4	19	−436	982	922	1,001	1,148	1.00	0.69	–	<.001

[1]Note. Par = parameters; LL = 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.

1.3 Plot Information Criteria

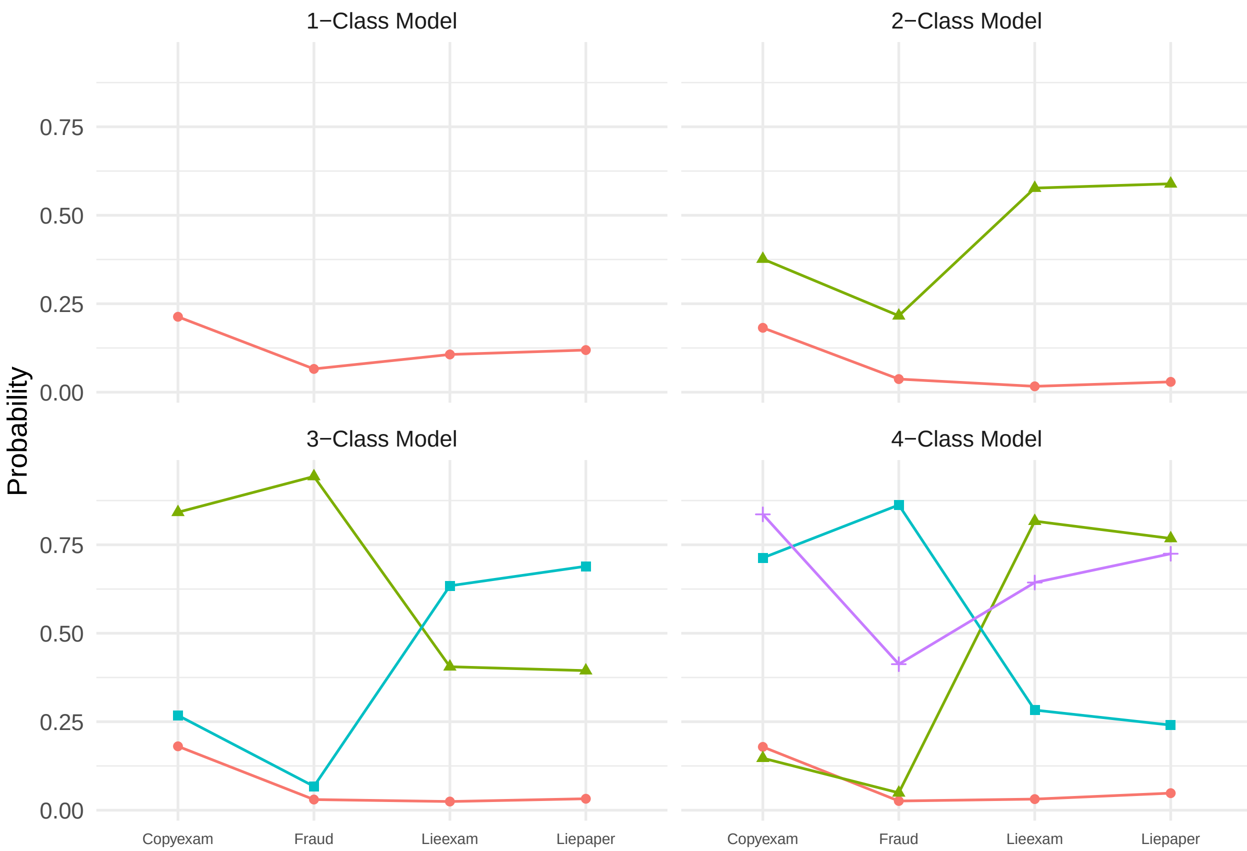
```
allFit %>% dplyr::select(2:7) %>%
  rowid_to_column() %>%
  pivot_longer(`BIC`:`AWE`,
    names_to = "Index",
    values_to = "ic_value") %>%
  mutate(Index = factor(Index,
    levels = c("AWE","CIAC","BIC","aBIC"))) %>%
  ggplot(aes(x = rowid, y = ic_value,
    color = Index, shape = Index,
    group = Index, lty = Index)) +
  geom_point(size = 2.0) + geom_line(size = .8) +
  scale_x_continuous(breaks = 1:6) +
  labs(x = "Number of Classes", y = "Information Criteria Value") +
  theme_cowplot() + theme(legend.title = element_blank(), legend.position = "top")
```




```
ggsave(here("12-intro-mixtures", "figures", "Fig1_IC_plot_cheat.png"), dpi=300, height=5, width=7, unit
```

1.4 Compare Conditional Item Probability Plots

```
model_results <- data.frame()
for (i in 1:length(output_cheat)) {
  temp <- data.frame(unclass(output_cheat[[i]]$parameters$unstandardized)) %>%
    mutate(model = paste0(i, "-Class Model"))
  model_results <- rbind(model_results, temp) }

pp_plots <- model_results %>%
  filter(paramHeader == "Thresholds") %>% dplyr::select(est, model, LatentClass, param) %>%
  mutate(prob = (1 / (1 + exp(est))), param = str_to_title(str_remove_all(param, "[$1]")))

ggplot(pp_plots,
       aes(x = param, y = prob, color = LatentClass, shape = LatentClass, group = LatentClass)) +
  geom_point() + geom_line() + facet_wrap(~ model, ncol = 2) + labs(x= "", y = "Probability") +
  theme_minimal() + theme(legend.position = "none", axis.text.x = element_text(size = 6))
```




```
ggsave(here("12-intro-mixtures", "figures","Fig2_compare_Kclass_Cheat.png"), dpi=300, height=4, width=6
```

1.5 Plot Final Model - Conditional Item Probability Plot

This syntax creates a function called `plot_lca_function` that requires 7 arguments (inputs):

- `model_name`: name of Mplus model object (e.g., `model_step1`)
- `item_num`: the number of items in LCA measurement model (e.g., `5`)
- `class_num`: the number of classes (k) in LCA model (e.g., `3`)
- `item_labels`: the item labels for x-axis (e.g., `c("Enjoy","Useful","Logical","Job","Adult")`)
- `class_labels`: the class label names (e.g., `c("Adaptive Coping","Externalizing Behavior","No Coping")`)
- `class_legend_order` = change the order that class names are listed in the plot legend (e.g., `c(2,1,3)`)
- `plot_title`: include the title of the plot here (e.g., `"LCA Posterior Probability Plot"`)

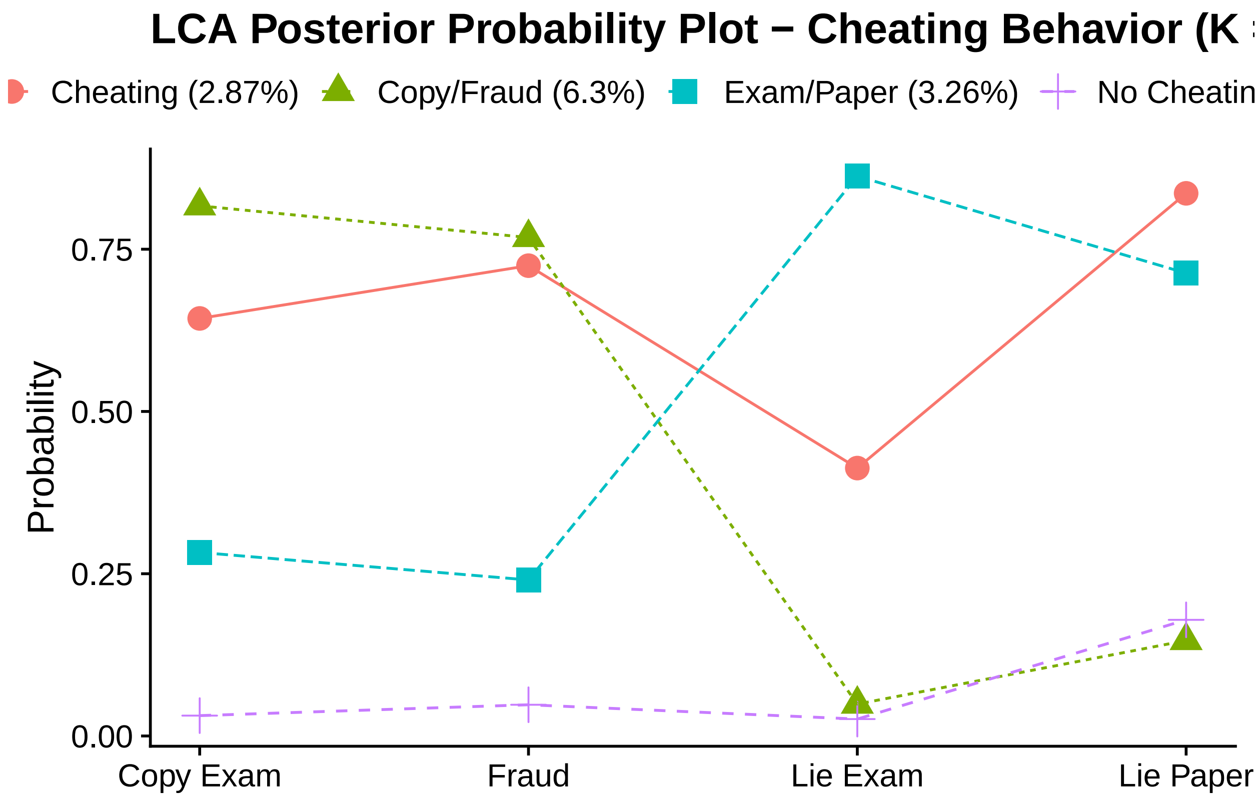
Read in plot data from Mplus output file `c5_lca_enum.out`

```
model_c4 <- readModels(here("12-intro-mixtures", "enum_cheat", "c4_lca_cheat.out"), quiet = TRUE)
```

```r
plot_lca_function <- function(model_name,item_num,class_num,item_labels,
                              class_labels,class_legend_order,plot_title){

mplus_model <- as.data.frame(model_name$gh5$means_and_variances_data$estimated_probs$values)
plot_data <- mplus_model[seq(2, 2*item_num, 2),]

c_size <- as.data.frame(model_name$class_counts$modelEstimated$proportion)
colnames(c_size) <- paste0("cs")
c_size <- c_size %>% mutate(cs = round(cs*100, 2))
colnames(plot_data) <- paste0(class_labels, glue(" ({c_size[1:class_num,]}%)"))
plot_data <- plot_data %>% relocate(class_legend_order)

plot_data <- cbind(Var = paste0("U", 1:item_num), plot_data)
plot_data$Var <- factor(plot_data$Var,
               labels = item_labels)
plot_data$Var <- fct_inorder(plot_data$Var)

pd_long_data <- melt(plot_data, id.vars = "Var")

# This syntax uses the data.frame created above to produce the plot with `ggplot()`

p <- pd_long_data %>%
  ggplot(aes(x = as.integer(Var), y = value,
  shape = variable, colour = variable, lty = variable)) +
  geom_point(size = 4) + geom_line() +
  scale_x_continuous("", breaks = 1:item_num, labels = plot_data$Var) +
  labs(title = plot_title, y = "Probability") +
  theme_cowplot() +
  theme(legend.title = element_blank(),
        legend.position = "top")

p
return(p)
}
```

Run the `plot_lca_function` by specifying each input (***Figure 1***)

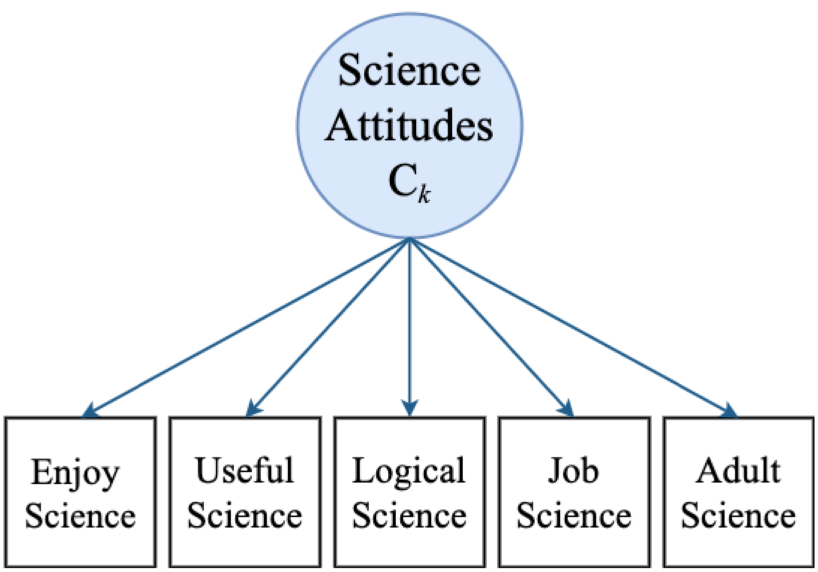
```r
plot_lca_function(
  model_name = model_c4,
  item_num = 4,
  class_num = 4,
  item_labels = c("Copy Exam", "Fraud", "Lie Exam", "Lie Paper"),
  class_labels = c("No Cheating","Copy/Fraud","Exam/Paper","Cheating"),
  class_legend_order = c(4,2,3,1),
  plot_title = "LCA Posterior Probability Plot - Cheating Behavior (K = 4)"
  )
```




```
ggsave(here("12-intro-mixtures", "figures", "Fig3_C4_LCA_cheat.png"), dpi=300, height=5, width=7, units=
```

Example 2: Longitudinal Study of American Youth, Science Attitudes

---



Load data

```
lsay_data <- read_csv("https://garberadamc.github.io/project-site/data/lca_lsay_sci.csv",
                      na = c("9999", "9999.00")) %>%
  clean_names() %>%
  dplyr::select(1:5, Enjoy = ab39m, Useful = ab39t,
                     Logical = ab39u, Job = ab39w, Adult = ab39x)
```

View LCA indicators

LCA Indicators[1]

Name	Label	Values
Enjoy	I enjoy science	0 = Disagree, 1 = Agree
Useful	Science useful in everday problems	0 = Disagree, 1 = Agree
Logical	Science helps logical thinkng	0 = Disagree, 1 = Agree
Job	Need science for a good job	0 = Disagree, 1 = Agree
Adult	Will use science often as an adult	0 = Disagree, 1 = Agree

[1]Longitudinal Study of American Youth

2.1 Enumeration:

Estimate K-class models with 1 through 6 classes.

Run enumeration using `mplusObject` method

```
lca_k1_6  <- lapply(1:6, function(k) {

  lca_enum  <- mplusObject(

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

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

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

  OUTPUT = "sampstat residual tech11 tech14;",
```

```
  PLOT =
    "type = plot3;
    series = Enjoy-Adult(*);",

  usevariables = colnames(lsay_data),
  rdata = lsay_data)

lca_enum_fit <- mplusModeler(lca_enum,
                            dataout=glue(here("12-intro-mixtures", "enum_lsay", "lca_lsay.dat")),
                            modelout=glue(here("12-intro-mixtures", "enum_lsay", "c{k}_lca.inp")) ,
                            check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

2.2 Generate Model Fit Summary Table

```
output_lsay <- readModels(here("12-intro-mixtures", "enum_lsay"), quiet = TRUE)

enum_lsay <- LatexSummaryTable(output_lsay,
                keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC",
                           "BLRT_PValue", "T11_VLMR_PValue","Observations"),
                sortBy = "Title")
```

Calculate relevant fit indices for summary table

```
allFit <- enum_lsay %>%
  mutate(aBIC = -2*LL+Parameters*log((Observations+2)/24)) %>%
  mutate(CIAC = -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)
```

Generate the fit summary table

```
allFit %>%
  mutate(Title = str_remove(Title, " LCA Enumeration - LSAY Example")) %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md(" ")) %>%
  cols_label(
    Title = "Classes",
```

```r
    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* = 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(10,decimals = 2,
             drop_trailing_zeros=TRUE,
             suffixing = TRUE) %>%
  fmt_number(c(3:9,11),
             decimals = 0) %>%
  fmt_missing(1:11,
              missing_text = "--") %>%
  fmt(c(8:9,11),
    fns = function(x)
    ifelse(x<0.001, "<.001",
           scales::number(x, accuracy = 0.01))) %>%
  fmt(10, fns = function(x)
    ifelse(x>100, ">100",
           scales::number(x, accuracy = .1)))
```

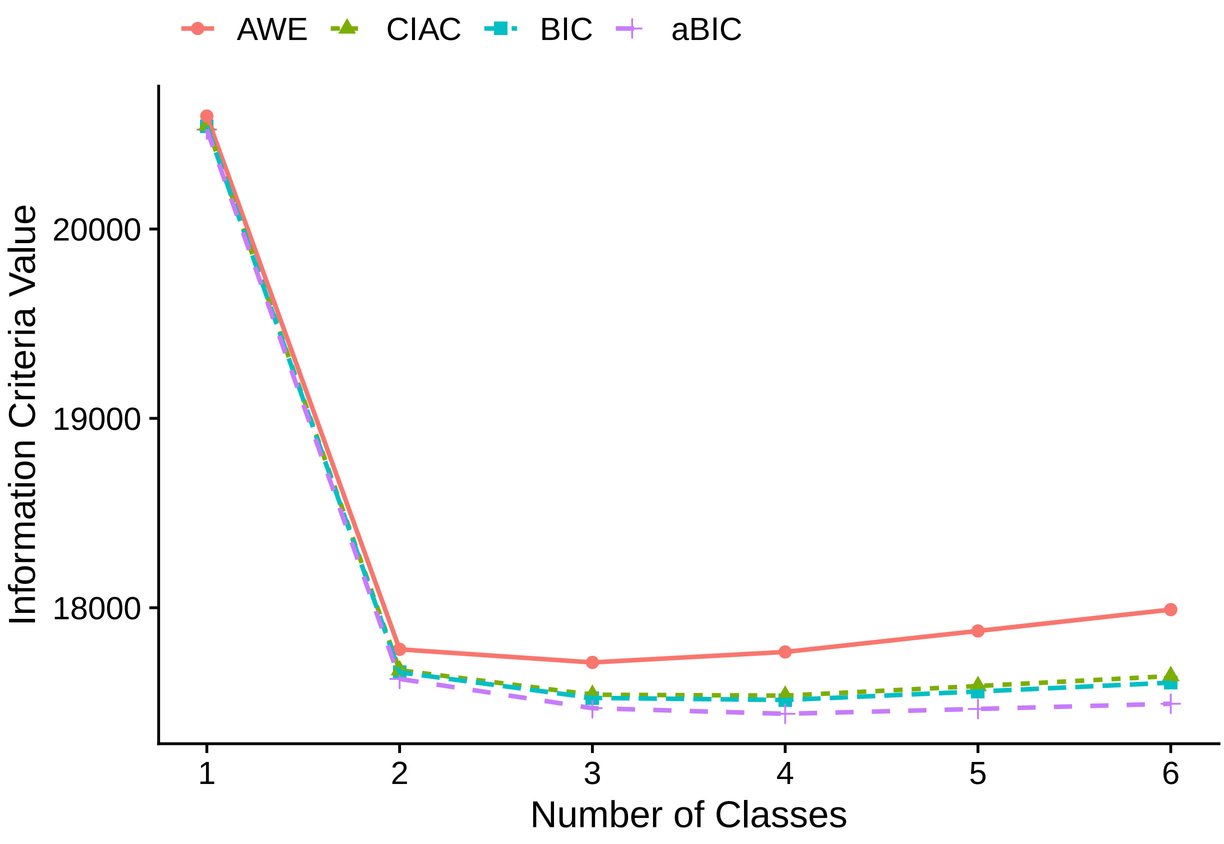
Model Fit Summary Table[1]

Classes	Par	*LL*	BIC	aBIC	CIAC	AWE	BLRT	VLMR	BF	*cmPk*
Class 1	5	−10,251	20,541	20,525	20,546	20,596	–	–	0.0	<.001
Class 2	11	−8,785	17,659	17,624	17,670	17,780	<.001	<.001	0.0	<.001
Class 3	17	−8,694	17,524	17,470	17,541	17,711	<.001	<.001	0.0	0.00
Class 4	23	−8,664	17,513	17,440	17,536	17,766	<.001	<.001	>100	1.00
Class 5	29	−8,662	17,558	17,465	17,587	17,877	1.00	0.67	>100	<.001
Class 6	35	−8,662	17,604	17,493	17,639	17,990	0.67	0.79	–	<.001

[1]Note. Par = parameters; LL = 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.

2.3 Plot Information Criteria

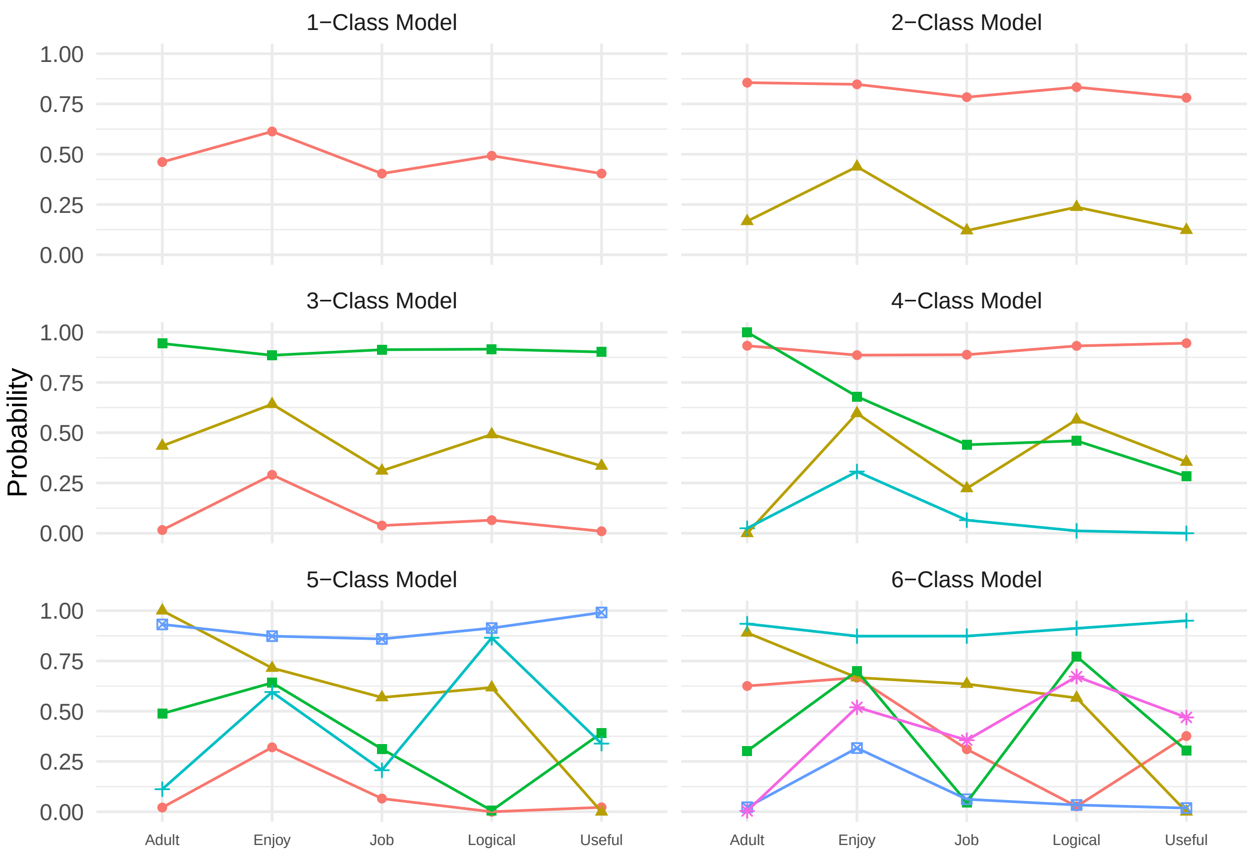
```r
allFit %>% dplyr::select(2:7) %>%
  rowid_to_column() %>%
  pivot_longer(`BIC`:`AWE`,
    names_to = "Index",
    values_to = "ic_value") %>%
  mutate(Index = factor(Index,
    levels = c("AWE","CIAC","BIC","aBIC"))) %>%
  ggplot(aes(x = rowid, y = ic_value,
    color = Index, shape = Index,
    group = Index, lty = Index)) +
  geom_point(size = 2.0) + geom_line(size = .8) +
  scale_x_continuous(breaks = 1:6) +
  labs(x = "Number of Classes", y = "Information Criteria Value") +
  theme_cowplot() + theme(legend.title = element_blank(), legend.position = "top")
```




```r
ggsave(here("12-intro-mixtures", "figures", "Fig1_IC_plot_LSAY.png"), dpi=300, height=5, width=7, units
```

2.4 Compare Conditional Item Probability Plots

```r
model_results <- data.frame()
for (i in 1:length(output_lsay)) {
  temp <- data.frame(unclass(output_lsay[[i]]$parameters$unstandardized)) %>%
    mutate(model = paste0(i, "-Class Model"))
  model_results <- rbind(model_results, temp) }

pp_plots <- model_results %>%
  filter(paramHeader == "Thresholds") %>% dplyr::select(est, model, LatentClass, param) %>%
  mutate(prob = (1 / (1 + exp(est))), param = str_to_title(str_remove_all(param, "[$1]")))

ggplot(pp_plots,
       aes(x = param, y = prob, color = LatentClass, shape = LatentClass, group = LatentClass)) +
  geom_point() + geom_line() + facet_wrap(~ model, ncol = 2) + labs(x= "", y = "Probability") +
  theme_minimal() + theme(legend.position = "none", axis.text.x = element_text(size = 6))
```



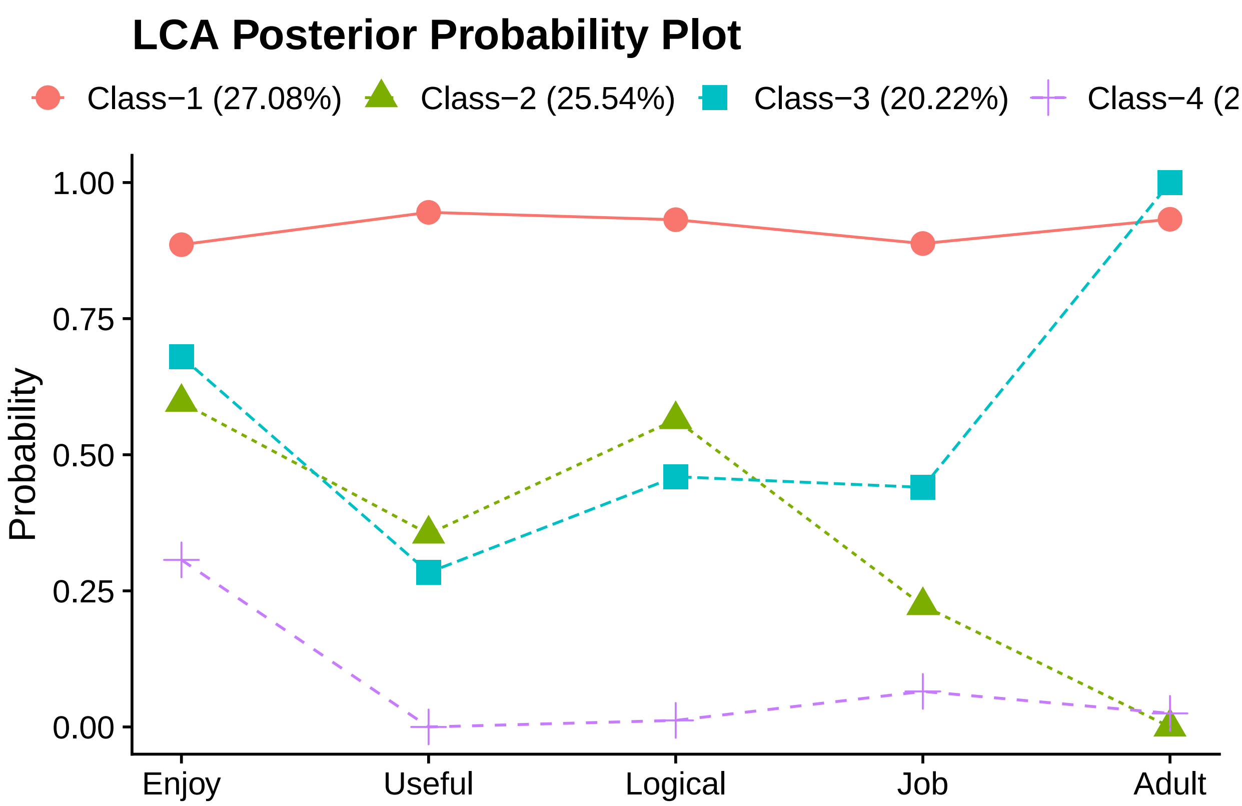

```r
ggsave(here("12-intro-mixtures", "figures","Fig2_compare_Kclass_LSAY.png"), dpi=300, height=4, width=6,
```

2.5 Plot Final Model - Conditional Item Probability Plot

```r
model_c4 <- readModels(here("12-intro-mixtures", "enum_lsay", "c4_lca.out"), quiet = TRUE)
```

Run the `plot_lca_function` by specifying each input (*Figure 1*)

```r
plot_lca_function(
  model_name = model_c4,
  item_num = 5,
  class_num = 4,
  item_labels = c("Enjoy", "Useful", "Logical", "Job", "Adult"),
  class_labels = c("Class-1","Class-2","Class-3","Class-4"),
  class_legend_order = c(1,2,3,4),
  plot_title = "LCA Posterior Probability Plot"
  )
```



Example 3 - Kindergarten Student Entrance Profile (KSEP; Quirk et al., 2011)

```r
ksep <- read_csv("https://garberadamc.github.io/project-site/data/ksep_sub_18.csv")
```

LCA Indicators - KSEP Example[1]

Name	Label	Values
seek_hlp	Seeks adult help when appropriate	0 = Not Mastered, 1 = Mastered
cooperat	Engages in cooperative play activities with peers	0 = Not Mastered, 1 = Mastered
imp_cntr	Exhibits impulse control and self-regulation	0 = Not Mastered, 1 = Mastered
repeats	Stays with or repeats a task	0 = Not Mastered, 1 = Mastered
separate	Separates appropriately from caregiver most days	0 = Not Mastered, 1 = Mastered
new_activ	Is enthusiastic and curious in approaching new activities	0 = Not Mastered, 1 = Mastered
folw_rul	Follows rules when participating in routine activities	0 = Not Mastered, 1 = Mastered
name	Recognizes own name	0 = Not Mastered, 1 = Mastered
writes	Writes own name	0 = Not Mastered, 1 = Mastered
express	Demonstrates expressive abilities	0 = Not Mastered, 1 = Mastered
quantity	Understands that numbers represent quantity	0 = Not Mastered, 1 = Mastered
colors	Recognizes Colors	0 = Not Mastered, 1 = Mastered
shapes	Recognizes primary shapes	0 = Not Mastered, 1 = Mastered

[1]Kindergarten Student Entrance Profile

3.1 Enumeration:

Estimate K-class models with 1 through 6 classes.

```r
lca_k1_6  <- lapply(1:6, function(k) {

  lca_enum  <- mplusObject(

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

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

  ANALYSIS =
   "estimator = mlr;
    type = mixture;
    !stseed = 5212020;
    starts = 200 100;
```

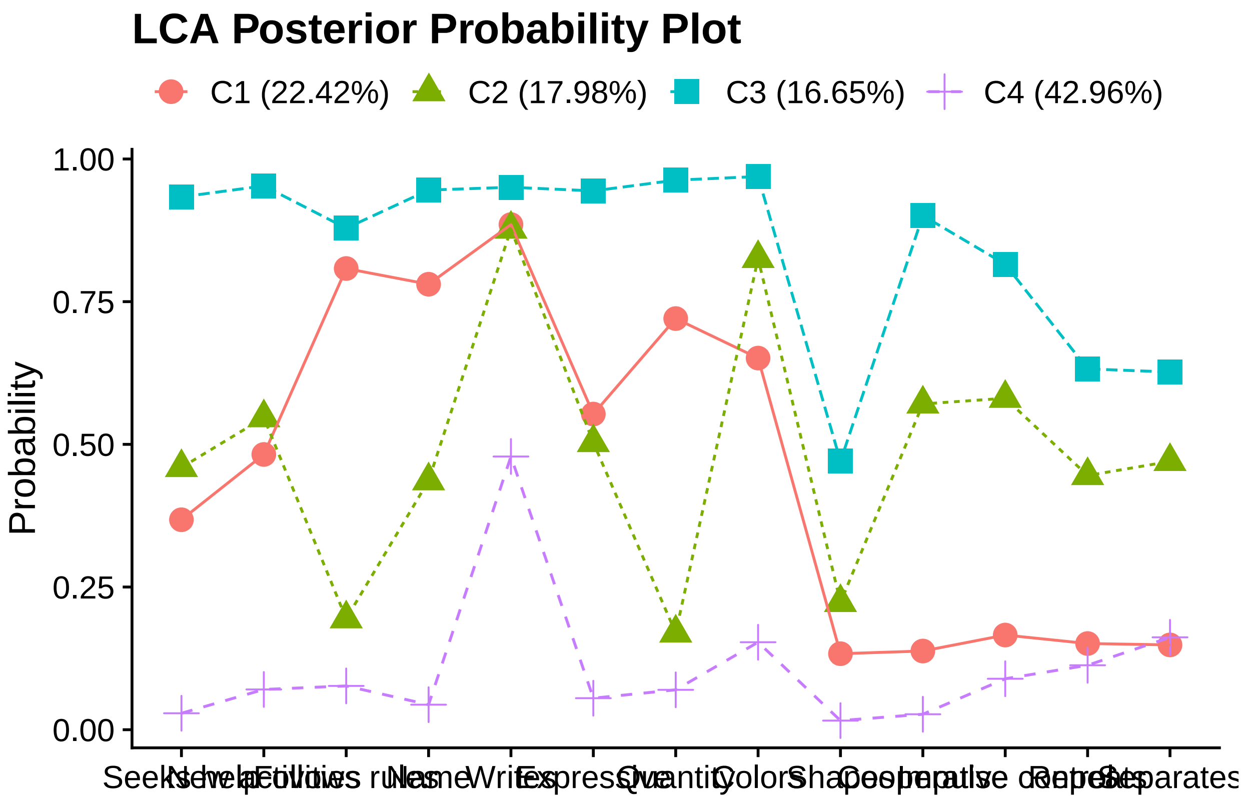
```
    processors = 10;",

  OUTPUT = "sampstat residual tech11 tech14;",

  PLOT =
    "type = plot3;
    series = seek_hlp-shapes(*);",

  usevariables = colnames(ksep),
  rdata = ksep)

lca_enum_fit <- mplusModeler(lca_enum,
                            dataout=glue(here("12-intro-mixtures", "enum_ksep", "lca_ksep.dat")),
                            modelout=glue(here("12-intro-mixtures", "enum_ksep", "c{k}_lca_ksep.inp"))
                            check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

3.5 Plot Final Model - Conditional Item Probability Plot

```
model_c4 <- readModels(here("12-intro-mixtures", "enum_ksep", "c4_lca_ksep.out"), quiet = TRUE)
```

Run the `plot_lca_function` by specifying each input (***Figure 1***)

```
plot_lca_function(
  model_name = model_c4,
  item_num = 13,
  class_num = 4,
  item_labels = c("Seeks help", "Cooperative", "Impulse control", "Repeats",
                  "Separates", "New activities", "Follows rules", "Name",
                  "Writes", "Expressive", "Quantity", "Colors", "Shapes"),
  class_labels = c("C2","C1","C3","C4"),
  class_legend_order = c(2,1,3,4),
  plot_title = "LCA Posterior Probability Plot"
  )
```




```
ggsave(here("12-intro-mixtures", "figures","C4_KSEP_LCA_plot.png"),
       dpi=300, height=4, width=6, units="in")
```

References

Drew A. Linzer, Jeffrey B. Lewis (2011). poLCA: An R Package for Polytomous Variable Latent Class Analysis. Journal of Statistical Software, 42(10), 1-29. URL http://www.jstatsoft.org/v42/i10/.

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.

Miller, J. D., Hoffer, T., Suchner, R., Brown, K., & Nelson, C. (1992). LSAY codebook. Northern Illinois University.

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

Quirk, M., Furlong, M., Lilles, E., Felix, E., & Chin, J. (2011). Preliminary development of a kindergarten school readiness assessment for Latino students. Journal of Applied School Psychology, 27(1), 77-102.

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/