

Introduction to Mixture Models - Enumeration & Plotting

Adam Garber

Norwegian University of Science and Technology - A Course in MplusAutomation

May 31, 2021

Latent Class Analysis (LCA)

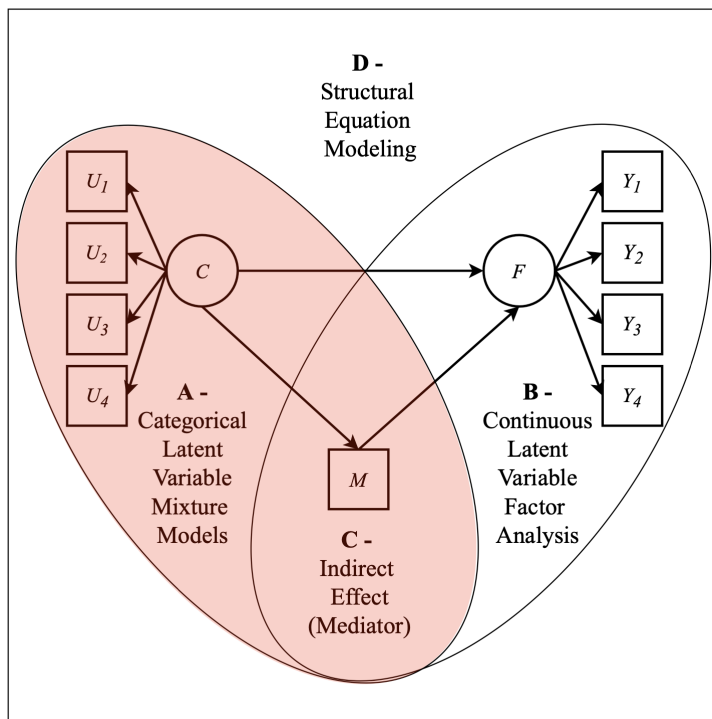


Figure. Picture has been adapted from study by Múthen, 2006.

Lab preparation

Change starting location to folder 19-intro-mixtures

```
source("rep_functions.R")

change_here(glue("{project_location}/19-intro-mixtures"))

here()

## [1] "/Users/agarber/github/NTNU-workshop/19-intro-mixtures"
```

Data source:

1. The first example utilizes a dataset on undergraduate *Cheating* available from the **poLCA** package (Dayton, 1998): [See documentation here](#)
 2. The second examples utilizes the public-use dataset, *The Longitudinal Survey of American Youth* (**LSAY**): [See documentation here](#)
 3. The third examples utilizes the *Kindergarten Student Entrance Profile* (**KSEP**) (Quirk et al., 2011): [See documentation here](#)
-

Load packages

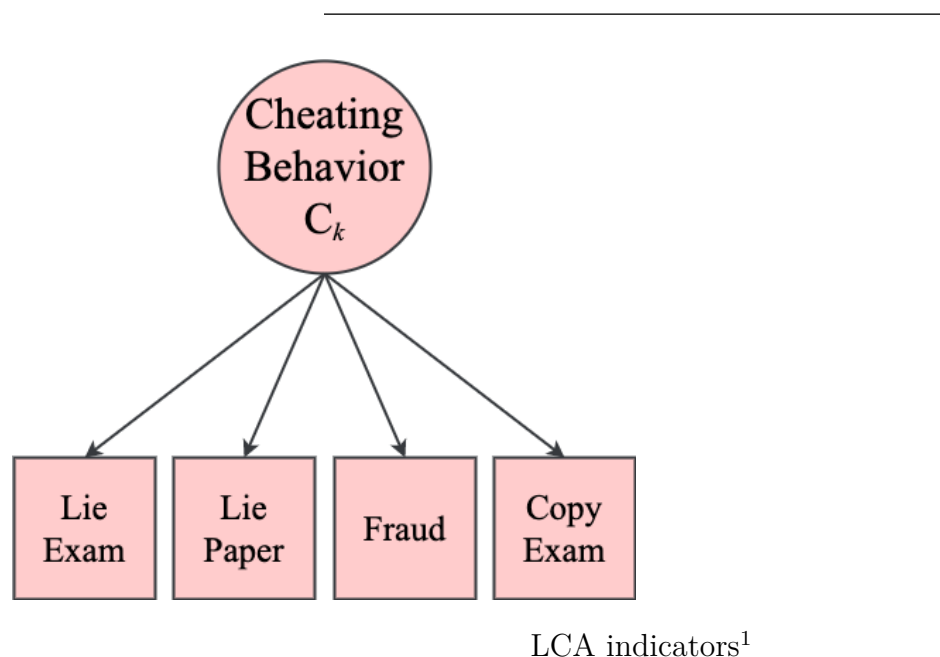
```
library(tidyverse)
library(haven)
library(glue)
library(MplusAutomation)
library(here)
library(janitor)
library(gt)
library(semPlot)
library(reshape2)
library(cowplot)
library(poLCA)
```

Enumerate and plot mixtures

Compare k -class models 1 through 6

Example 1: Undergraduate Cheating behavior

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



| 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 |

¹Undergraduate Cheating Behavior

Prepare data

```
data(cheating)

cheating <- cheating %>% clean_names()

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

Run a quick LCA

```
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("enum_cheat", "lca_cheat.dat")),
                             modelout=glue(here("enum_cheat", "c{k}_lca_cheat.inp")),
                             check=TRUE, run = TRUE, hashfilename = FALSE)
})

```

View model fit statistics with `mixtureSummaryTable()`

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

```

## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>

```

```

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

enum_summary %>%
  gt() %>%
  tab_header(
    title = "Fit Indices",
    subtitle = md("&nbsp;")) %>%
  cols_label(
    Title = "Classes",
    LL = md("*LL*"),
    BLRT_PValue = html("BLRT"),
    T11_VLMR_PValue = html("VLMR")) %>%
  tab_options(
    table.width = pct(80)) %>%
  tab_footnote(
    footnote = "Undergraduate Cheating Behavior",
    location = cells_title())

```

Fit Indices¹

| Classes | LL | BIC | aBIC | BLRT | VLMR |
|---------|----------|---------|---------|--------|--------|
| Class 1 | -467.438 | 957.937 | 945.250 | NA | NA |
| Class 2 | -440.027 | 931.941 | 903.395 | 0.0000 | 0.0000 |
| Class 3 | -436.236 | 953.184 | 908.779 | 0.1395 | 0.1656 |
| Class 4 | -436.145 | 981.829 | 921.564 | 1.0000 | 0.6868 |

¹Undergraduate Cheating Behavior

Extract and prepare plot data

```
# extract posterior probabilities
plot1 <- as.data.frame(output_cheat[["c4_lca_cheat.out"]]
                        [["gh5"]][["means_and_variances_data"]]
                        [["estimated_probs"]][["values"]]
                        [seq(2, 8, 2),]) #seq("from", "to", "by")

# extract class size proportions
c_size <- as.data.frame(output_cheat[["c4_lca_cheat.out"]]
                        [["class_counts"]][["modelEstimated"]][["proportion"]])

colnames(c_size) <- paste0("cs")

c_size <- c_size %>% mutate(cs = round(cs*100, 2))

#rename columns (classes) and "Var" (indicator names)
colnames(plot1) <- paste0("C", 1:4, glue(" ({c_size[1:4,]}%))")
plot1 <- cbind(Var = paste0("U", 1:4), plot1)

# choose the order of indicators by changing to ordered factor
plot1$Var <- fct_inorder(plot1$Var)

#change dataframe from wide to long format
pd_long1 <- melt(plot1, id.vars = "Var")
```

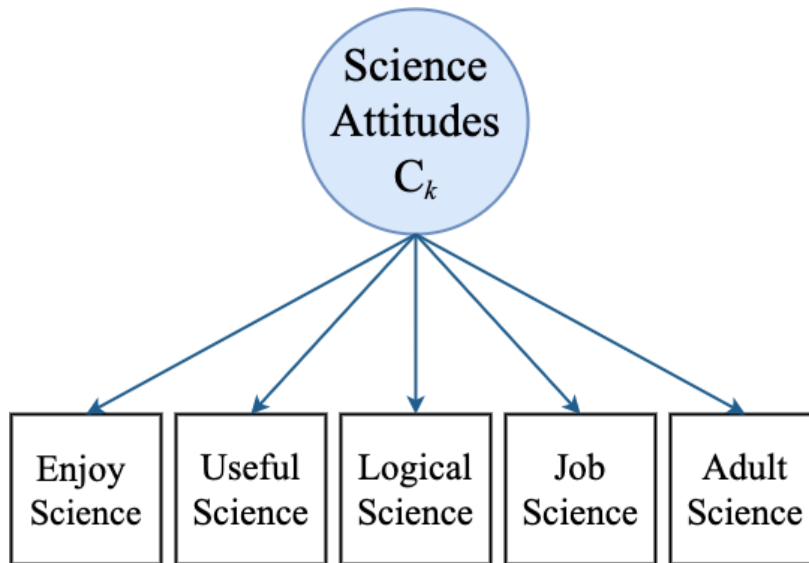
Plot 4-class latent class posterior probability plot

```
ggplot(pd_long1, aes(Var, value, shape = variable,
                    colour = variable, lty = variable)) +
  geom_point(size = 4) + geom_line(aes(as.integer(Var))) +
  scale_x_discrete(labels = c("Lie Exam", "Lie Paper", "Fraud", "Copy Exam")) +
  scale_y_continuous("Probability") +
  scale_colour_viridis_d(end = .7) +
  theme_cowplot() + labs(x=" ") +
  theme(text=element_text( size=12),
        legend.key.width = unit(.5, "line"),
        legend.text = element_text( size=12),
        legend.title = element_blank(),
        legend.position = "top")
```

save figure

```
ggsave(here("figures", "C4_Cheat_LCA_Plot.png"), dpi="retina", height=5, width=7, units="in")
```

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 Indicators¹

| Name | Label | Values |
|---------|-------------------------------------|-------------------------|
| Enjoy | I enjoy science | 0 = Disagree, 1 = Agree |
| Useful | Science useful in everyday 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 |

¹Longitudinal Study of American Youth

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("enum_lsay", "lca_lsay.dat")),
                             modelout=glue(here("enum_lsay", "c{k}_lca.inp")),
                             check=TRUE, run = TRUE, hashfilename = FALSE)
})

```

Compare model fit for series of enumerated models

```
all_output <- readModels(here("enum_lsay"), quiet = TRUE)
```

```

## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>

```

```

enum_summary <- LatexSummaryTable(all_output,
                                   keepCols=c("Title", "LL", "BIC", "aBIC",
                                                "BLRT_PValue", "T11_VLMR_PValue"),
                                   sortBy = "Title")

gt(enum_summary) %>%
  tab_header(
    title = "Fit Indices",
    subtitle = md("&nbsp;")) %>%
  cols_label(
    Title = "Classes",
    LL = md("*LL*"),
    BLRT_PValue = html("BLRT"),
    T11_VLMR_PValue = html("VLMR")) %>%

```

```

tab_options(
  table.width = pct(80)) %>%
tab_footnote(
  footnote = "Longitudinal Study of American Youth",
  location = cells_title())

```

Fit Indices¹

| Classes | <i>LL</i> | BIC | aBIC | BLRT | VLMR |
|---------|------------|----------|----------|--------|--------|
| Class 1 | -10250.604 | 20541.34 | 20525.45 | NA | NA |
| Class 2 | -8785.317 | 17658.92 | 17623.97 | 0.0000 | 0.0000 |
| Class 3 | -8693.569 | 17523.59 | 17469.57 | 0.0000 | 0.0000 |
| Class 4 | -8664.090 | 17512.79 | 17439.71 | 0.0000 | 0.0000 |
| Class 5 | -8662.386 | 17557.54 | 17465.40 | 1.0000 | 0.6736 |
| Class 6 | -8661.541 | 17604.01 | 17492.80 | 0.6667 | 0.7884 |

¹Longitudinal Study of American Youth

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

```

# for (i in 1:length(all_output)) {
#   temp <- all_output[[i]]$parameters$unstandardized
#   temp <- data.frame(unclass(temp)) %>%
#     mutate(model = paste0(i, "-Class Model"))
#   model_results <- rbind(model_results, temp)
# }
#
# model_results <- model_results %>%
#   filter(paramHeader == "Thresholds") %>%
#   dplyr::select(est, model, LatentClass, param) %>%
#   mutate(prob = (1 / (1 + exp(est))))
#
# ggplot(model_results, aes(x = param, y = prob,
#   color = LatentClass,
#   shape = LatentClass,
#   group = LatentClass,
#   lty = LatentClass)) +
#   geom_point() + geom_line() +
#   facet_wrap(~ model, ncol = 2) +
#   labs(title = "LCA Posterior Probability Plot",
#     x = "Science attitudes", y = "Probability") +
#   theme_minimal()

```

Example 3 - Kindergarten Student Entrance Profile (KSEP)

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


LCA Indicators¹

| 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 |

¹Kindergarten Student Entrance Profile

Enumeration: Compare k -class models 1-6

```
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("enum_ksep", "lca_ksep.dat")),
                              modelout=glue(here("enum_ksep", "c{k}_lca_ksep.inp")) ,
                              check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

Compare model fit for series of enumerated models

```
all_output <- readModels(here("enum_ksep"), quiet = TRUE)
```

```
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
```

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

```
gt(enum_summary) %>%
  tab_header(
    title = "Fit Indices",
    subtitle = md("&nbsp;")) %>%
  cols_label(
    Title = "Classes",
    LL = md("*LL*"),
    BLRT_PValue = html("BLRT"),
    T11_VLMR_PValue = html("VLMR")) %>%
  tab_options(
    table.width = pct(80)) %>%
  tab_footnote(
    footnote = "Kindergarten Student Entrance Profile",
    location = cells_title())
```

Fit Indices¹

| Classes | <i>LL</i> | BIC | aBIC | BLRT | VLMR |
|---------|------------|----------|----------|------|--------|
| Class 1 | -11844.461 | 23783.62 | 23742.32 | NA | NA |
| Class 2 | -9712.793 | 19622.26 | 19536.49 | 0 | 0.0000 |
| Class 3 | -9372.043 | 19042.74 | 18912.49 | 0 | 0.0000 |
| Class 4 | -9215.475 | 18831.58 | 18656.86 | 0 | 0.1240 |
| Class 5 | -9137.866 | 18778.34 | 18559.15 | 0 | 0.0722 |
| Class 6 | -9075.565 | 18755.71 | 18492.05 | 0 | 0.0394 |

¹Kindergarten Student Entrance Profile

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

```
# for (i in 1:length(all_output)) {
#   temp <- all_output[[i]]$parameters$unstandardized
#   temp <- data.frame(unclass(temp)) %>%
#     mutate(model = paste0(i, "-Class Model"))
#   model_results <- rbind(model_results, temp)
# }
#
# model_results <- model_results %>%
#   filter(paramHeader == "Thresholds") %>%
#   dplyr::select(est, model, LatentClass, param) %>%
```

```
# mutate(prob = (1 / (1 + exp(est))))
#
# ggplot(model_results, aes(x = param, y = prob,
#                           color = LatentClass, shape = LatentClass,
#                           group = LatentClass, lty = LatentClass)) +
#   geom_point() + geom_line() +
#   scale_colour_viridis_d() +
#   facet_wrap(~ model, ncol = 2) +
#   labs(title = "Kindergarten Student Entrance Profile (KSEP)",
#        x = " ", y = "Probability") +
#   scale_x_discrete(labels =
#     c("Seeks help", "Cooperative", "Impulse control", "Repeats", "Separates",
#       "New activities", "Follows rules", "Name", "Writes", "Expressive", "Quantity",
#       "Colors", "Shapes")) +
#   theme_minimal() + theme(panel.grid.major.y = element_blank(),
#                           axis.text.x = element_text(angle = -45, hjust = -.1))
```

Extract and prepare plot data

```
# extract posterior probabilities
plot1 <- as.data.frame(all_output[["c4_lca_ksep.out"]]
                        [["gh5"]][["means_and_variances_data"]]
                        [["estimated_probs"]][["values"]]
                        [seq(2, 26, 2),]) #seq("from", "to", "by")

# extract class size proportions
c_size <- as.data.frame(all_output[["c4_lca_ksep.out"]]
                        [["class_counts"]][["modelEstimated"]][["proportion"]])

colnames(c_size) <- paste0("cs")

c_size <- c_size %>% mutate(cs = round(cs*100, 2))

#rename columns (classes) and "Var" (indicator names)
colnames(plot1) <- paste0("C", 1:4, glue(" ({c_size[1:4,]}%)"))
plot1 <- cbind(Var = paste0("U", 1:13), plot1)

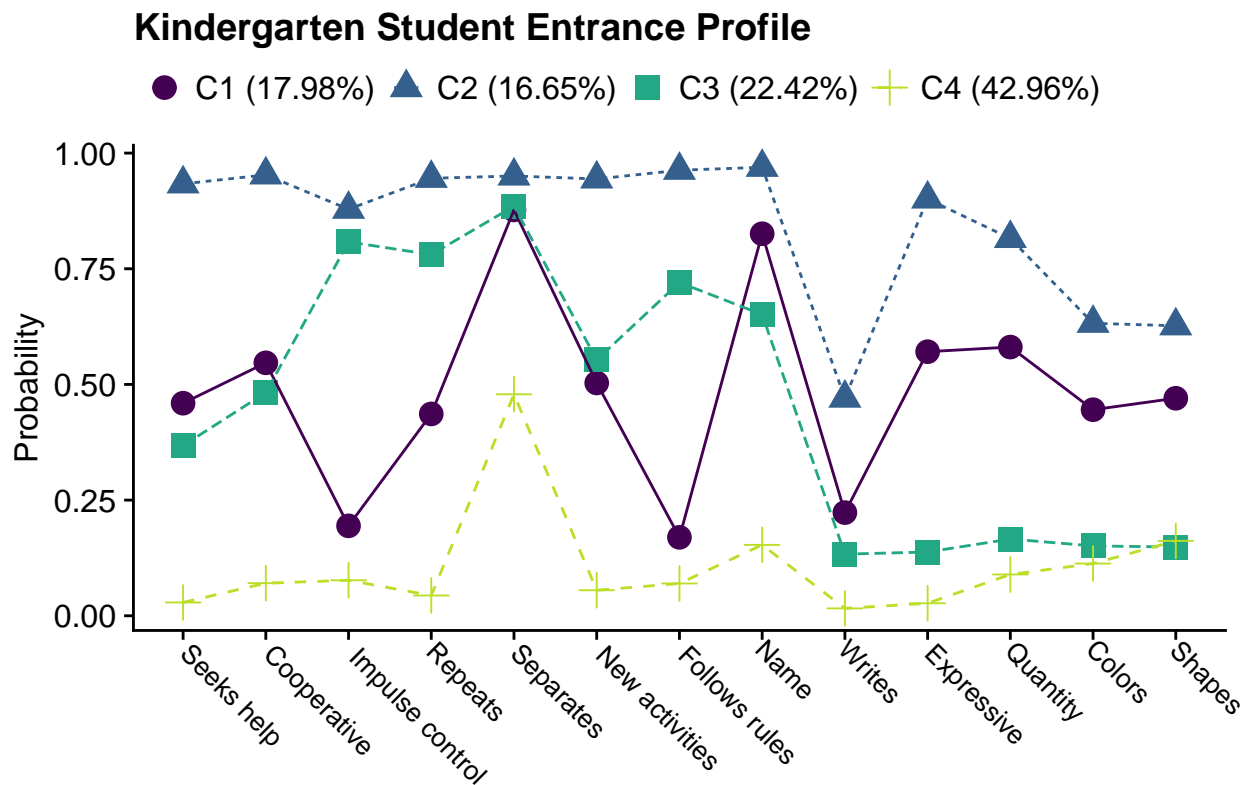
# choose the order of indicators by changing to ordered factor
plot1$Var <- fct_inorder(plot1$Var)

#change dataframe from wide to long format
pd_long1 <- melt(plot1, id.vars = "Var")
```

Plot 4-class mixture

```
ggplot(pd_long1, aes(Var, value, shape = variable,
                    colour = variable, lty = variable)) +
  geom_point(size = 4) + geom_line(aes(as.integer(Var))) +
  scale_x_discrete(labels = c("Seeks help", "Cooperative", "Impulse control", "Repeats",
                              "Separates", "New activities", "Follows rules", "Name",
                              "Writes", "Expressive", "Quantity", "Colors", "Shapes")) +
  scale_y_continuous("Probability") +
  scale_colour_viridis_d(end = .9) +
```

```
labs(title="Kindergarten Student Entrance Profile", x=" ") +
theme_cowplot() +
theme(text=element_text( size=12),
      legend.key.width = unit(.5, "line"),
      legend.text = element_text( size=12),
      legend.title = element_blank(),
      axis.text.x = element_text(angle = -45, hjust = -.1, size=10),
      legend.position = "top")
```



```
ggsave(here("figures", "Class4_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>